

UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL



FACULTAD DE INGENIERÍAS

CARRERA DE INGENIERÍA ELECTRÓNICA

Trabajo de titulación previa a la obtención del título de:
INGENIERO ELECTRÓNICO

Tema:

**DISEÑO Y CONSTRUCCIÓN DE CINCO ENTRENADORES DIDÁCTICOS
CON SISTEMAS MICROPROCESADOS Y DESARROLLO DE UNA
APLICACIÓN DE CONTROL DE VELOCIDAD PARA UN MOTOR DE
CORRIENTE ALTERNA**

AUTORES:

**RAFAEL CHRISTIAN FRANCO REINA
ISRAEL MONTESDEOCA PALADINES**

DIRECTOR:

ING. LUIS CÓRDOVA RIVADENEIRA

Guayaquil, Septiembre del 2014

DECLARATORIA DE RESPONSABILIDAD

Guayaquil, 15 septiembre de 2014

Mediante este escrito, Rafael Franco Reina e Israel Montesdeoca Paladines declaramos bajo juramento, que la presente tesis aquí descrita es de nuestra autoría; misma que no ha sido previamente presentada para ningún grado o calificación profesional; y que hemos consultado las referencias bibliográficas correspondientes que se incluyen al final del documento.

Así, mediante esta declaración escrita, cedemos nuestro derecho de propiedad intelectual del presente trabajo, a la Universidad Politécnica Salesiana, conforme a lo establecido por la Ley de Propiedad Intelectual, por su reglamento y normativa en vigencia.

Además, declaramos que los conceptos, análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad de los autores.

Rafael Franco Reina

C.I.: 0923328629

Israel Montesdeoca Paladines

CI.: 0922516505

DEDICATORIA

A los docentes y estudiantes de la Carrera de Ingeniería Electrónica de la Universidad Politécnica Salesiana, quienes mediante la implementación del presente trabajo han obtenido mayor eficiencia en el proceso de enseñanza-aprendizaje en la materia de Microprocesados.

AGRADECIMIENTO

A Dios que nos ha iluminado para alcanzar grandes metas como ésta.

Con gran afecto agradecemos a nuestro director de tesis Ing. Luis Córdova Rivadeneira, quien siempre dispuesto a colaborarnos y orientarnos en el desarrollo de este trabajo, especialmente en los momentos de incertidumbre y resolución de problemas del mismo.

A nuestros amigos: Luis Orejuela, Ing. Gary Burgos, Ing. Roberto López, Ing. Henry Cruz, Christopher McCarthy, quienes en el transcurso de estos meses han aportado muchísimo con sus conocimientos y experiencias.

A nuestros padres, quienes han sido motores de motivación, han creído en nuestras capacidades académicas y en nuestras actitudes.

A nuestros profesores quienes con esfuerzo y dedicación a través de sus conocimientos han contribuido para la formación de nuestra vida profesional, como salesianos de corazón y personas de bien para la sociedad.

Rafael Franco Reina

Israel Montesdeoca Paladines

ÍNDICE GENERAL

CARÁTULA.....	I
DECLARATORIA DE RESPONSABILIDAD.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE GENERAL.....	V
ÍNDICE DE FIGURAS.....	XII
ÍNDICE DE TABLAS.....	XVI
ABSTRACT.....	XIX
CAPÍTULO I.....	1
PROBLEMÁTICA	
1.1. PLANTEAMIENTO DEL PROBLEMA.....	<i>1</i>
1.2. DELIMITACIONES.....	<i>1</i>
1.3 JUSTIFICACIONES:.....	<i>3</i>
1.4. OBJETIVOS:	<i>4</i>
1.5. HIPÓTESIS	<i>5</i>
1.6. VARIABLES E INDICADORES.....	<i>6</i>
1.7. POBLACIÓN Y MUESTRA	<i>6</i>
1.8. ALCANCE DE LA PROPUESTA.	<i>7</i>
CAPÍTULO II	8
FUNDAMENTOS TEÓRICOS	
2.1 PICs DE MICROCHIP	8
2.1.1 Características de los PICs.	9
2.1.2 Gammas de PICs	11
2.1.3 PIC 18F4550	12
2.1.3.1 Distribución de pines PIC 18F4550	13
2.1.3.2 Arquitectura y estructura interna.....	14
2.1.4 Herramientas de desarrollo	15

2.1.5 Programación con MicroCode Studio y PICBasic	15
2.1.5.1 Proceso de instalación	16
2.1.5.2 Estructura básica del programa	20
2.1.5.3 Subrutinas de programa	21
2.1.5.3. Operadores y componentes de PICBasic	22
2.2. BLUETOOTH.....	26
2.2.1. Clasificación del bluetooth.....	27
2.2.2. Operación de los dispositivos Bluetooth.....	27
2.2.3. Enlace físico.....	29
2.2.4. Perfiles	30
2.2.5. Módulo Bluetooth Inalámbrico Serial - Transceiver RF	31
2.2.5.1. Características:	31
2.2.5.2. Distribución de pines:	31
2.2.5.3. Modo de operación, comandos AT	32
2.2.5.4. Clasificación de los comandos AT:.....	33
2.3. TECNOLOGÍAS GSM/GPRS Y GPS, MÓDULO GSM/GPRS/GPS V3.0 ...	44
2.3.1. Sistema GSM	44
2.3.1.1. Estructura de una red GSM	44
2.3.1.2. Elementos del sistema GSM	46
2.3.1.3. Módulo de identificación del usuario (SIM):.....	47
2.3.2. La evolución al GPRS.....	49
2.3.2.1. Arquitectura de red GPRS.....	49
2.3.3. TECNOLOGIA GPS.....	51
2.3.3.1. Funcionamiento y estructura	51
2.3.4. Módulo GSM/GPRS/GPS V3.0-SKU: TEL0051	53
2.3.4.1. Especificaciones técnicas	53
2.4. TECNOLOGIA X-BEE.....	54
2.4.1. Características técnicas:	57
2.4.2 Modos de operación	58
2.4.2.1. Modo recibir/transmitir	58
2.4.2.3. Modo de comando.....	59
2.4.2.4. Modo transparente.....	61
2.4.2.5. Modo de operación API	62
2.4.2.5. Idle	63
2.4.4. Pasos de instalación del software X-CTU:.....	64

2.5 DATA LOGGER	68
2.5.1 Características.....	68
2.5.2. Aplicaciones.....	69
2.5.3. Operación y set de comandos.....	70
2.5.4. Diagrama de pines, modo UART, modo SPI.....	74
2.5.5. Datos técnicos e indicadores del Datalogger	75
2.6 ETHERNET	76
2.6.1. Tecnología y velocidad de Ethernet.....	76
2.6.2 Versiones.....	77
2.6.3 Formato de la trama Ethernet.....	79
2.6.4 Módulo Arduino Ethernet	80
2.6.4.1 Características del módulo:	80
2.6.4.2. Sistema de alimentación, potencia, memoria, entrada y salida.....	81
2.7 GLCD (PANTALLA GRÁFICA DE CRISTAL LÍQUIDA)	84
2.7.1. Características de una GLCD 128x64.....	85
2.7.2. Descripción de los pines de conexión de una GLCD 128x64.....	85
2.8. TECLADO MATRICIAL 4X3	86
2.8.1. Esquema eléctrico	87
2.9. LCD 20X4	88
2.9.1. Diagrama de bloques y distribución de pines	89
2.10. SOFTWARE DE DISEÑO Y SIMULACIÓN ELECTRÓNICA PROTEUS	90
2.10.1. Estructura del programa.	90
2.11 MOTORES DE CORRIENTE ALTERNA	93
2.11.1 Motor trifásico.....	93
2.11.2. Partes principales de un motor trifásico	93
2.11.3. Sistema trifásico, tensión de servicio y conexión de motores trifásicos	94
2.11.4 Motor trifásico Siemens 1LA7-070-4YA60, 1800rpm, 0.5 hp.....	96
2.12. CONTROLADORES PID	98
2.13. VARIADOR DE VELOCIDAD ALTIVAR 312	99
2.13.1. Aplicaciones generales y funciones básicas.....	99
2.13.2. Características del variador de velocidad ATV132	101
2.13.3 Descripción del panel de control.....	104
2.13.4. Visualización de parámetros iniciales y estructura de los menús.	105

2.14 LABVIEW (ACRÓNIMO DE LABORATORY VIRTUAL INSTRUMENTATION ENGINEERING WORKBENCH).....	107
2.14.1. Características	108
2.14.2. Interfaces de comunicaciones:	108
2.14.3 Programa en LabVIEW	109
2.14.4. Ventajas de LabVIEW al adquirir datos y procesar señales.	110
2.14.5. Tarjeta NI_USB6009	112
CAPÍTULO III.....	117
DISEÑO, CONSTRUCCIÓN Y MONTAJE DE ENTRENADORES DIDÁCTICOS CON SISTEMA DE MICROPROCESADO Y APLICACIÓN DE CONTROL DE VELOCIDAD DE MOTOR	
3.1. INTRODUCCIÓN.....	117
3.2. DIAGRAMA DE BLOQUES DEL ENTRENADOR DIDÁCTICO.....	117
3.3. DISEÑO, ESQUEMA Y CIRCUITERÍA ELECTRÓNICA DE INTERFAZ ENTRADA.....	120
3.3.1. Diagrama esquemático en Proteus-ISIS	120
3.3.2. Descripción de elementos utilizados en la interfaz de entrada.	121
3.3.3. Diseño del circuito impreso en Proteus-ARES	122
3.3.4. Montaje de elementos electrónicos en la PCB.	123
3.4. DISEÑO, ESQUEMA Y CIRCUITERÍA ELECTRÓNICA DE INTERFAZ SALIDA.	124
3.4.1. Descripción de elementos utilizados en la interfaz de salida.....	124
3.4.2. Diagrama esquemático en Proteus-ISIS	124
3.4.3. Diseño del circuito impreso en Proteus-ARES	126
3.4.4. Montaje de elementos electrónicos en la PCB.	127
3.5. DISEÑO, ESQUEMA Y CIRCUITERÍA ELECTRÓNICA DEL MÓDULO DE CONTROL	128
3.5.1. Diagrama esquemático en Proteus-ISIS.....	128
3.5.2. Descripción de elementos utilizados en el módulo de control.	130
3.5.3. Diseño del circuito impreso en Proteus-ARES	131
3.5.4. Montaje de elementos electrónicos en la PCB.	132
3.6. DISEÑO, ESQUEMA Y CIRCUITERÍA ELECTRÓNICA DEL MÓDULO BLUETOOTH.	133
3.6.1. Diagrama esquemático en Proteus-ISIS	133
3.6.2. Descripción de elementos utilizados en el módulo Bluetooth.	134
3.6.3. Diseño del circuito impreso en Proteus-ARES	134
3.6.4. Montaje de elementos electrónicos en la PCB.	135

3.7. DISEÑO, ESQUEMA Y CIRCUITERÍA ELECTRÓNICA DEL MÓDULO XBEE.....	136
3.7.1. Diagrama esquemático en Proteus-ISIS.....	136
3.7.2. Descripción de elementos utilizados en el módulo Xbee.....	137
3.7.3. Diseño del circuito impreso en Proteus-ARES	137
3.7.4. Montaje de elementos electrónicos en la PCB.....	138
3.8. DISEÑO, ESQUEMA Y CIRCUITERÍA ELECTRÓNICA DEL MÓDULO DATALOOPER.....	139
3.8.1. Diagrama esquemático en Proteus-ISIS.....	139
3.8.2. Descripción de elementos utilizados en el módulo Datalogger.	140
3.8.3. Diseño del circuito impreso en Proteus-ARES	140
3.8.4. Montaje de elementos electrónicos en la PCB.....	141
3.9. DISEÑO, ESQUEMA Y CIRCUITERÍA ELECTRÓNICA DEL MÓDULO ETHERNET..	142
3.9.1. Diagrama esquemático en Proteus-ISIS.....	142
3.9.2. Descripción de elementos utilizados en el módulo Ethernet.	143
3.9.3. Diseño del circuito impreso en Proteus-ARES	143
3.9.4. Montaje de elementos electrónicos en la PCB.....	144
3.10. DISEÑO, ESQUEMA Y CIRCUITERÍA ELECTRÓNICA DEL MÓDULO GSM/GPRS/GPS.....	145
3.10.1. Diagrama esquemático en Proteus-ISIS.....	145
3.10.2. Descripción de elementos utilizados en el módulo GSM/GPRS/GPS.	146
3.10.4. Montaje de elementos electrónicos en la PCB.	147
3.11. DISEÑO, ESQUEMA Y CIRCUITERÍA ELECTRÓNICA DEL MÓDULO DE CONTROL DE VELOCIDAD DE CORRIENTE ALTERNA.....	148
3.11.1. Diagrama esquemático en Proteus-ISIS.....	148
3.11.2. Descripción de elementos utilizados en el módulo de control de velocidad de corriente alterna.	149
3.11.3. Diseño del circuito impreso en Proteus-ARES	150
3.11.4. Montaje de elementos electrónicos en la PCB.....	151
3.12. ESQUEMA Y ACOPLAMIENTO DE LOS DIFERENTES MÓDULOS INALÁMBRICOS Y EL MÓDULO DE CONTROL DE VELOCIDAD DE CORRIENTE ALTERNA.....	152

CAPÍTULO IV	155
--------------------------	------------

MANTENIMIENTO PARA EL ENTRENADOR DIDÁCTICO

4.1. ORGANIZACIÓN DEL MANTENIMIENTO PARA LOS MÓDULOS.....	155
4.2. DETERMINACIÓN DE LAS TAREAS ÓPTIMAS PARA LA CONSERVACIÓN Y PRESERVACIÓN DEL ENTRENADOR CON SISTEMAS MICROPROCESADOS Y APLICACIÓN DE CONTROL DE VELOCIDAD DE MOTOR DE CORRIENTE ALTERNA.	155
4.3. PROGRAMACIÓN DEL MANTENIMIENTO PARA EL ENTRENADOR.....	157

CAPÍTULO V.....	159
------------------------	------------

PRÁCTICAS DE LABORATORIO

5.1. PRÁCTICA 1.....	160
REGISTRO DE TEMPERATURA DE UN HORNO MEDIANTE UN DATA LOGGER.	160
5.2. PRÁCTICA 2.....	175
DESARROLLO DE UN CIRCUITO CONTROLADOR DE TRÁNSITO.....	175
5.3. PRÁCTICA 3.....	189
MEDIANTE UN SENSOR TOMAR MUESTRAS DE TEMPERATURA Y ENVIARLAS INALÁMBRICAMENTE USANDO X-BEE.....	189
5.4. PRÁCTICA 4.....	200
REALIZAR UN CIRCUITO MICRO CONTROLADO CAPAZ DE DETECTAR LA PRESENCIA DE ALGÚN INTRUSO EN UNA VIVIENDA.	200
5.5. PRÁCTICA 5.....	216
REALIZAR UN RELOJ ATÓMICO DIGITAL A TRAVÉS DE UN GPS Y MOSTRAR LA HORA EN UN GLCD	216
5.6. PRÁCTICA 6.....	227
MEDIANTE UN CIRCUITO DIGITAL Y A TRAVÉS DEL MÓDULO GPS MOSTRAR LA UBICACIÓN EN COORDENADAS GEOGRÁFICAS A TRAVÉS DE LA PANTALLA GLCD.....	227
5.7. PRÁCTICA 7.....	238
REALIZAR UN CIRCUITO PARA EL MANEJO DE MATRIZ DE LED USANDO EL PIC 18F4550	238
5.8. PRÁCTICA 8.....	252
MOSTRAR GRÁFICAMENTE EN LA GLCD LA GRÁFICA DE UNA ECUACIÓN DE PRIMER ORDEN DADO DOS PUNTOS DE UNA RECTA INGRESADOS POR TECLADO.	252
5.9. PRÁCTICA 9.....	266

DISEÑAR UN CIRCUITO ELECTRÓNICO DE CONTROL DE VELOCIDAD DE UN MOTOR DE CORRIENTE ALTERNA CON SET POINT.....	266
5.10. PRÁCTICA 10.....	276
DISEÑAR UN CIRCUITO ELECTRÓNICO CON CONTROL PID PARA UN MOTOR DE CORRIENTE ALTERNA	276
CAPÍTULO VI.....	317
COSTO DEL PROYECTO	
6.1. COSTO DE MÓDULO DATA LOGGER.....	317
6.2. COSTO DE MÓDULO XBEE	318
6.3. COSTO DE MÓDULO BLUETOOTH.....	318
6.4. COSTO DE MÓDULO GSM - GPRS - GPS.....	319
6.5. COSTO DE MÓDULO ARDUINO ETHERNET	320
6.6. COSTO DE MÓDULO INTERFACE DE ENTRADA.....	320
6.7. COSTO DE MÓDULO INTERFACE PARA MOTOR.....	321
6.8. COSTO DE MÓDULO INTERFACE DE SALIDA.....	321
6.9. COSTO DE MÓDULO FUENTE	322
6.10. COSTO DE MÓDULO PRINCIPAL O CONTROL.....	323
6.11. COSTO DE ELEMENTOS VARIOS	324
CRONOGRAMA DE EJECUCIÓN	326
BIBLIOGRAFÍA.....	328
CONCLUSIONES.....	330

ÍNDICE DE FIGURAS

	Pág.	
Figura 1	Arquitectura del procesador-modelo Harvard	9
Figura 2	Técnica de segmentación “pipe-line”	10
Figura 3	Distribución de pines del PIC 18F4550	13
Figura 4	Diagrama de bloques PIC 18F4550	14
Figura 5	Link de archivo MicroCode Studio	15
Figura 6	Proceso de instalación inicial	16
Figura 7	Aceptación de licencia de MicroCode	16
Figura 8	Carpeta de enrutamiento de MicroCode	17
Figura 9	Proceso final de instalación del MicroCode	17
Figura 10	Ubicación de la carpeta para la librería de compilación	18
Figura 11	Configuración en la opción de compilación y programación	18
Figura 12	Configuración manual de la opción de compilación	19
Figura 13	Esquema de estructura de programa	20
Figura 14	Esquema de estructura de programa	21
Figura 15	Tecnología Bluetooth	28
Figura 16	Perfil de acceso general	30
Figura 17	Distribución de pines Bluetooth	31
Figura 18	Pulsos de LED parpadeando en modo AT	43
Figura 19	Estructura de la red GSM	44
Figura 20	Descripción de elementos de la red GSM	46
Figura 21	Tipos de SIM en la actualidad	47
Figura 22	Disposición de arquitectura GPRS	49
Figura 23	Funcionamiento del GPS	52
Figura 24	Estructura de Arduino GPS/GSM/GPRS	53
Figura 25	Diagrama esquemático del módulo Xbee	54
Figura 26	Circuito básico del Xbee	55
Figura 27	Modos de operación del módulo Xbee	58
Figura 28	Estructura del comando AT	60
Figura 29	Escritura y uso de parámetro MY	60
Figura 30	Configuración del módulo en modo transparente	61

Figura 31	Pasos de instalación X-CTU, ventana de bienvenida	64
Figura 32	Pasos de instalación X-CTU, selección de la partición.	64
Figura 33	Pasos de instalación X-CTU, proceso de instalación.	65
Figura 34	Pasos de instalación X-CTU, ventana final de la instalación.	65
Figura 35	Búsqueda del programa instalado en el PC	66
Figura 36	Ventana principal software X-CTU: PC Settings, Range Test	66
Figura 37	Ventana principal software X-CTU: Terminal, Modem Conf	67
Figura 38	Estructura física de la memory stick datalogger	68
Figura 39	Diagrama VNC1L / Microcontrolador.	70
Figura 40	Módulo Arduino Ethernet	80
Figura 41	Módulo GLCD	84
Figura 42	Pines de Conexión de GLCD 128x64	86
Figura 43	Teclado matricial 4X3	86
Figura 44	Esquema eléctrico teclado matricial 4X3	87
Figura 45	LCD 20X4	88
Figura 46	Diagrama de bloques LCD 20X4	89
Figura 47	Proteus-Entorno ISIS	91
Figura 48	Proteus-Entorno ARES	92
Figura 49	Partes generales de un motor trifásico	93
Figura 50	Descripción gráfica de tensión de servicio	94
Figura 51	Diagrama esquemático del motor trifásico	95
Figura 52	Tamaño constructivo 070 –motor trifásico	96
Figura 53	Diagrama de bloques controlador PID.	98
Figura 54	Variador de velocidad Altivar 312.	99
Figura 55	Descripción de partes variador de velocidad Altivar 312	104
Figura 56	Variador de velocidad Altivar 312	106
Figura 57	Logotipo LabVIEW	107
Figura 58	Diagrama de bloques en LabVIEW	110
Figura 59	Tarjeta NI USB6009	112
Figura 60	Diagrama de bloques NI USB-6009	113
Figura 61	Diagrama de bloques principal del entrenador didáctico	117
Figura 62	Diagrama de bloques módulo de control de velocidad del motor	118
Figura 63	Diagrama Proteus ISIS interfaz de entrada	119

Figura 64	Diseño circuito impreso Proteus-ARES, interfaz de entrada	122
Figura 65	Tarjeta PCB y elementos electrónicos, interfaz de entrada	123
Figura 66	Diseño electrónico Proteus ISIS, interfaz de salida	125
Figura 67	Circuito impreso Proteus ARES, interfaz de salida	126
Figura 68	Tarjeta PCB y elementos electrónicos, interfaz de salida	127
Figura 69	Relay DC5V, diagrama esquemático	127
Figura 70	Diagrama esquemático Proteus ISIS, módulo de control	129
Figura 71	Diagrama de circuito impreso Proteus ARES, módulo de control	131
Figura 72	Tarjeta PCB y elementos electrónicos, módulo de control	132
Figura 73	Diagrama esquemático Proteus ISIS, módulo Bluetooth	133
Figura 74	Diagrama de circuito impreso Proteus ARES, módulo Bluetooth	134
Figura 75	Tarjeta PCB y elementos electrónicos, módulo Bluetooth	135
Figura 76	Diagrama esquemático Proteus ISIS, módulo Xbee	136
Figura 77	Diseño circuito impreso Proteus ARES, módulo Xbee	137
Figura 78	Tarjeta PCB y elementos electrónicos, módulo Xbee	138
Figura 79	Diseño esquemático Proteus ISIS, módulo Datalogger	139
Figura 80	Diseño de circuito impreso Proteus ARES, módulo Datalogger	140
Figura 81	Diseño de circuito impreso Proteus ARES, módulo Datalogger	141
Figura 82	Diseño esquemático Proteus ISIS, módulo Ethernet	142
Figura 83	Diseño del circuito impreso Proteus ARES, módulo Ethernet	143
Figura 84	Tarjeta PCB y elementos electrónicos, módulo Ethernet	144
Figura 85	Diseño esquemático Proteus ISIS, módulo GSM/GPRS/GPS	145
Figura 86	Diseño de circuito en Proteus ARES, módulo GSM/GPRS/GPS	146
Figura 87	Tarjeta PCB e instalación de elementos, módulo GSM/GPRS/GPS	147
Figura 88	Diseño esquemático Proteus ISIS, módulo de control de velocidad de motor	148
Figura 89	Diseño de circuito impreso Proteus ARES, módulo de control de velocidad de motor	150
Figura 90	Tarjeta PCB y elementos ensamblados, módulo de control de velocidad de motor	151
Figura 91	Diseño de chasis metálico del entrenador didáctico.	152
Figura 92	Prototipo chasis de entrenador didáctico	153

Figura 93 Entrenador didáctico terminado	154
Figura 94 Cronograma de ejecución	326

ÍNDICE DE TABLAS

	Pág.	
Tabla 1	Análisis de la hipótesis del tema de tesis	5
Tabla 2	Tipos de variables	22
Tabla 3	Operadores aritméticos básicos	23
Tabla 4	Operadores binarios básicos	24
Tabla 5	Operadores lógicos básicos	24
Tabla 6	Operadores de comparación básicos	25
Tabla 7	Clasificación de bluetooth según potencia de transmisión	27
Tabla 8	Clasificación de bluetooth según ancho de banda	27
Tabla 9	Descripción de pines Bluetooth	32
Tabla 10	Comandos de pruebas en AT	33
Tabla 11	Comando de pregunta de datos	34
Tabla 12	Comando de obtención de dirección de bluetooth	34
Tabla 13	Comando para obtener información del dispositivo	35
Tabla 14	Comando para obtener versión del módulo Bluetooth	36
Tabla 15	Establecer/preguntar nombre del dispositivo Bluetooth	36
Tabla 16	Establecer/preguntar clave	37
Tabla 17	Eliminar dispositivos apareados	37
Tabla 18	Estados de trabajo del módulo Bluetooth	38
Tabla 19	Comando para eliminar dispositivo vinculado en lista	38
Tabla 20	Establecer/preguntar tipo de dispositivo	39
Tabla 21	Búsqueda del dispositivo Bluetooth	39
Tabla 22	Comando de restauración a estado predeterminado	39
Tabla 23	Parámetros de estado predeterminado del módulo Bluetooth	40
Tabla 24	Establecer/consultar rol del módulo Bluetooth	40
Tabla 25	Establecer/consultar - parámetros serial	41
Tabla 26	Establecer/consultar - modo de conexión	42
Tabla 27	Comando para establecer paridad	42
Tabla 28	Comando de conexión del dispositivo	43
Tabla 29	Especificaciones técnicas generales de los módulos Xbee	57
Tabla 30	Tabla modo sleep y consumo de corriente	59

Tabla 31	Comandos para controlar Memory Datalogger	73
Tabla 32	Errores, comandos, resultados	73
Tabla 33	Diagrama de pines Modo UART	74
Tabla 34	Estados del LED indicador en el Datalogger	75
Tabla 35	Especificaciones técnicas del Datalogger	75
Tabla 36	Versiones Ethernet 802.3	78
Tabla 37	Estructura de la trama Ethernet 802.3	79
Tabla 38	Características generales del Arduino Ethernet	81
Tabla 39	Distribución de pines LCD 20X4	89
Tabla 40	Tensiones normalizadas para redes de corriente trifásica	94
Tabla 41	Conexiones y potencia nominal.	95
Tabla 42	Motores trifásicos Eficiencia IE1	97
Tabla 43	Características de variador de velocidad ATV132	103
Tabla 44	Estructura general de los menús Altivar132	105
Tabla 45	Características básicas de la NI USB-6009	112
Tabla 46	Terminales análogos NI USB-6009	114
Tabla 47	Terminales digitales NI USB-6009	115
Tabla 48	Descripción de señales en la NI USB-6009	116
Tabla 49	Elementos interfaz de entrada	121
Tabla 50	Elementos interfaz de salida	124
Tabla 51	Elementos de módulo de control	130
Tabla 52	Elementos de módulo Bluetooth	134
Tabla 53	Elementos de módulo Xbee	137
Tabla 54	Elementos de módulo Datalogger	140
Tabla 55	Elementos de módulo Ethernet	143
Tabla 56	Elementos de módulo GSM/GPRS/GPS	146
Tabla 57	Elementos de módulo de control de velocidad de motor	149
Tabla 58	Costo de elementos de Módulo DATA LOGGER	317
Tabla 59	Costo de elementos de Módulo XBEE	318
Tabla 60	Costo de elementos de Módulo BLUETOOTH	319
Tabla 61	Costo de elementos de Módulo GSM - GPRS – GPS	319
Tabla 62	Costo de elementos de Módulo ARDUINO ETHERNET	320
Tabla 63	Costo de elementos de Módulo INTERFACE DE ENTRADA	320

Tabla 64	Costo de elementos de Módulo Interface para Motor	321
Tabla 65	Costo de elementos de Módulo Interface de salida	322
Tabla 66	Costo de elementos de Módulo Fuente	323
Tabla 67	Costo de elementos de Módulo Principal o Control	324
Tabla 68	Costo de elementos varios	325

RESUMEN

AÑO	TÍTULO	ALUMNO/S	DIRECTOR DE TESIS	TEMA DE TESIS
2014	INGENIERO ELECTRÓNICO	FRANCO REINA RAFAEL CHRISTIAN MONTESDEOCA PALADINES ISRAEL EFRAIN	CÓRDOVA RIVADENEIRA LUIS	“DISEÑO Y CONSTRUCCIÓN DE CINCO ENTRENADORES DIDÁCTICOS CON SISTEMAS MICROPROCESADOS Y DESARROLLO DE UNA APLICACIÓN DE CONTROL DE VELOCIDAD PAA UN MOTOR DE CORRIENTE ALTERNA”

Este proyecto surgió como necesidad de mejoramiento del nivel de enseñanza aprendizaje en la materia de microprocesados de la carrera de Ingeniería Electrónica en la Universidad Politécnica Salesiana sede Guayaquil, se logró implementar cinco entrenadores didácticos con sistemas microprocesados, que favorecieron la interiorización eficiente de los conocimientos teóricos impartidos por los docentes. La metodología fue de carácter descriptivo y experimental, basada en: investigación de campo, observación áulica y una extensa investigación bibliográfica - experimental; por ende la población muestra se determinó por estudiantes de Microprocesados y sus docentes. Es así, como este proyecto se estructuró en seis capítulos que partieron desde el análisis de la situación problema con los parámetros pertinentes; los fundamentos teóricos para la construcción de los entrenadores didácticos; el diseño, construcción y montaje de entrenadores didácticos con sistemas de microprocesados y aplicación de control de velocidad de motor; el mantenimiento técnico correspondiente, las prácticas de laboratorio hasta el costo generado. Se demostró que si los docentes de la materia de microprocesados utilizan los entrenadores didácticos en la aplicación teórico-práctica de sus clases, se obtendrían beneficios inmediatos en el proceso enseñanza-aprendizaje; lo cual reveló que de la correcta utilización técnica del entrenador didáctico dependió el éxito del aprendizaje. En conclusión se ha contribuido a la innovación y prestigio institucional de la Carrera de Ingeniería Electrónica de la Universidad Politécnica Salesiana.

PALABRAS CLAVES

Entrenadores didácticos/ motor AC trifásico/Bluetooth/X-Bee/ aplicación/ GSM/ Ethernet/ GPS/ módulos/ Datalogger/ PIC/ LabVIEW/NI USB-6009

ABSTRACT

YEAR	TITLE	STUDENTS	THESIS DIRECTOR	THEME OF THESIS
2014	ELECTRONIC ENGINEER	FRANCO REINA RAFAEL CHRISTIAN MONTESDEOCA PALADINES ISRAEL EFRAIN	CÓRDOVA RIVADENEIRA LUIS	“DESIGN AND CONSTRUCTION OF FIVE LEARNING AIDS WITH MICROPROCESSOR SYSTEMS AND DEVELOPMENT OF AN APPLICATION TO CONTROL THE SPEED OF AN ALTERNATIVE CURRENT MOTOR”

This project came about as a need to improve the standard of teaching and learning in the career field of microprocessors Electronic Engineering at the Salesian Polytechnic University, allowing the implementation of five microprocessor learning aids, which helped the efficiently inner of the knowledge of theories taught by the professors. The methodology was both descriptive and experimental, based on: field research, class observation and an extensive bibliography research of published experiment results; therefore the sample population was determined by microprocessors students and their teachers. Thus, as this project was structured into six chapters that started from the analysis of the problematic situation with the parameters relevant; the theoretical basis for the construction of learning aids; the design, construction and installation of microprocessor system learning aid and their application of controlling engine speed; and technical maintenance, corresponding to generated cost of the project. It was shown that if the teachers in the program of microprocessor systems use the learning aids in the theoretical and practical application in their classes, immediate benefits would be obtained in the process of teaching and learning, thus revealing the correct use of the learning aids leads to a success in learning. In conclusion, it has contributed to the innovation and institutional prestige in the study program of Electronic Engineering of the Salesian Polytechnic University.

KEY WORDS:

Learning Aids/ AC Motor/Bluetooth/X-Bee/ Application/ GSM/ Ethernet/ GPS/ modules/ Datalogger/ PIC/ LabVIEW/NI USB-6009

CAPÍTULO I

PROBLEMÁTICA

1.1. Planteamiento del problema.

En la actualidad una de las problemáticas al realizar diferentes prácticas referente a tecnologías inalámbricas o control de ciertos parámetros mediante un microcontrolador dentro de la materia de *microprocesados* en general, es la carencia de los mismos, además de la pérdida de tiempo que esto conlleva o sus costos muy elevados; es de ahí, que se puede deducir que para una óptima funcionalidad de las clases: *¿Es necesaria la construcción de cinco entrenadores con sistema microprocesados y una aplicación de control moderno en los estudiantes de la materia de Microprocesados de la Universidad Politécnica Salesiana?*

1.2. Delimitaciones

El entrenador didáctico mencionado se implementó en el Laboratorio de Electrónica Digital de la Carrera de Ingeniería Electrónica de la UPS, en las cuales se dicta la materia de microprocesados. A continuación se detalla las delimitaciones del proyecto:

De los módulos inalámbricos:

- Solo brinda una solución para las tecnologías inalámbricas: Bluetooth, GSM, X-Bee, Data Logger, Ethernet y GPS.
- Los transeptores de tecnologías inalámbricas tienen rangos limitados, ejemplo: Bluetooth.
- El módulo de control de velocidad tendrá un rango determinado solo para pequeñas prácticas.
- Con nuevas tecnologías a futuro estos quedarán obsoletos.

De la programación PIC:

- Cada tipo de microcontrolador tiene su propio conjunto de instrucciones que un programador tiene que conocer para escribir un programa
- Un programador tiene que conocer el hardware del microcontrolador para escribir un programa
- Los programas al compilarlos pueden resultar un poco extensos y pesados por ello debe tenerse en cuenta la capacidad de memoria de programa del PIC a utilizar.
- Con este lenguaje tampoco se puede controlar del todo los tiempos y los registros bit a bit.

De Microcontroladores:

- En los microcontroladores de Arquitectura Harvard se debe poseer instrucciones especiales para acceder a las tablas de valores constantes que son necesarias incluir en los programas.
- Los microcontroladores de gama baja solo tiene dos niveles en la pila y no admiten interrupciones

De motores de corriente alterna

- Necesidad de alimentación.
- Sensibilidad a las vibraciones.
- Incapacidad para funcionar a bajas velocidades.
- Control de posicionamiento.

De LabVIEW

- No presenta polimorfismo sobre VIs creados, por ejemplo no podría tenerse un módulo que por una misma entrada permita algunas veces entrar un número y otras un arreglo.
- No permite programación orientada a objetos.
- No permite recursividad.
- Tarjetas de adquisición de datos muy costosas.

1.3 Justificaciones:

Los justificativos que han llevado a la propuesta de esta tentativa son:

Evolución: Los constantes descubrimientos de nuevas tecnologías hacen evolucionar inevitablemente la mentalidad de las personas, es por esto que basándonos en la electrónica, los protocolos para la comunicación y el acceso de información se ha optado por desarrollar este proyecto que ofrece un entorno de ejecución en tiempo real.

Innovación: Diseñar, desarrollar y construir cinco entrenadores para microcontroladores MICROCHIP y ATMEL para el control de periféricos de última tecnología con la finalidad de explotar al máximo las herramientas tecnológicas ya que como futuros ingenieros electrónicos tenemos como objetivo crear propuestas innovadoras.

Necesidad: Uno de los mayores justificativos es brindar a los usuarios involucrados a través de esta aplicación y sus entrenadores didácticos la comodidad y flexibilidad de poder realizar todo un seguimiento de los diferentes usos de las tecnologías: Bluetooth, GSM, X-Bee, Data Logger, Ethernet y GPS llegando a realizar pruebas específicas con datos reales y la finalidad de realizar nuevas aplicaciones partiendo de un manual de prácticas didácticas para operar los módulos entrenadores en las clases de microprocesados.

Economía: Realmente constituye un ahorro justificable para el entorno donde se vaya aplicar, por ejemplo en los laboratorios de Electrónica Digital de la Universidad. El uso que se dará a los entrenadores electrónicos en conjunto con la aplicación de control moderno representará costos mínimos en su utilización, que se caracterizan por su rápido desempeño, alta confiabilidad y facilidad de empleo.

1.4. Objetivos:

Objetivo general:

- Diseñar y construir cinco entrenadores didácticos con sistemas microprocesados y una aplicación de control de velocidad para un motor de corriente alterna, que se implementará en el Laboratorio de Electrónica Digital para mayor funcionalidad en las clases de la materia de Microprocesados de la Universidad Politécnica Salesiana.

Objetivos específicos:

- Describir las diferentes tecnologías innovadoras dentro de los esquemas electrónicos y de las telecomunicaciones.
- Diseñar y construir cinco entrenadores bajo las tecnologías inalámbricas: Bluetooth, GSM, X-Bee, Data Logger, Ethernet y GPS; mediante protocolos de comunicación, parámetros electrónicos, y la gama de PIC que existen en el mercado. Los entrenadores controlarán mediante tecnología de comunicación inalámbrica tanto datos, consultas y prácticas de la materia de sistemas de microprocesados II.
- Implementar un módulo de comunicación entre los entrenadores didácticos y la interfaz de control de velocidad del motor de corriente alterna dando una mayor accesibilidad a los datos, resultados específicos a prácticas electrónicas de micro procesados y reportes reales y confiables.
- Proponer una solución a dificultades presentadas al realizar 10 prácticas con las tecnologías expuestas en la materia de Sistemas Microprocesados II, además de la interfaz de control de velocidad en LabVIEW de un motor de corriente alterna y uso de una DAQ, siendo estable, eficaz y segura, con la finalidad de implementarla sin mayores complicaciones en cualquier ambiente donde se necesite este servicio.

1.5. Hipótesis

Si se implementan los entrenadores didácticos con sistema microprocesados y la aplicación de control de velocidad de un motor de corriente alterna, se fortalecerá un proceso de aprendizaje más funcional en las prácticas de las tecnologías inalámbricas: Bluetooth, GSM, X-Bee, Data Logger, Ethernet y GPS, además del uso de la electrónica en aplicaciones con motores de corriente alterna junto a LabVIEW y la tarjeta NI MYDAQ.

Favorecerá la investigación y el uso de los conocimientos adquiridos en la Universidad, además de las experiencias profesionales desarrolladas en el trayecto de la formación de Ingenieros Electrónicos.

Necesidad	Método	Solución
Envío de paquete de datos, códigos de acceso en forma inalámbrica	<i>Sistema de entrenadores didácticos</i> usuario- entrenador didáctico bluetooth, interfaz moderna -usuario	Posibilidad de conexión inalámbrica de corto alcance de datos y voz entre dispositivos/periféricos.
Medición de temperatura, tanque de control de nivel, registro de datos	<i>Sistema de entrenadores didácticos</i> usuario- entrenador didáctico, Data Logger, interfaz moderna -usuario	Obtención de valores reales mediante un archivo txt, xls.
Transmisión de datos por medio de una red de área local con CSMA/CD	<i>Sistema de entrenadores didácticos</i> usuario- entrenador didáctico, Ethernet, interfaz moderna -usuario	Adquisición de conceptos como velocidad de transmisión, tipos de cable, longitud máxima y topologías.
Realizar comunicaciones peer-to-peer, unicast o broadcast, uso de comandos AT	<i>Sistema de entrenadores didácticos</i> usuario- entrenador didáctico, X-Bee, interfaz moderna -usuario	Posibilidad de conectar un microcontrolador directamente al módulo con lo cual se dota a las aplicaciones de comunicación inalámbrica
Pruebas de envío de datos, SMS mediante plataforma GSM	<i>Sistema de entrenadores didácticos</i> usuario- entrenador didáctico, GSM, interfaz moderna -usuario	Adquisición de conceptos con respecto a servicios de red, transmisión de datos, cobertura de red en GSM.
Control de velocidad de un motor mediante módulo electrónico-analógico	<i>Sistema de entrenadores didácticos</i> usuario- entrenador didáctico, interfaz de control de velocidad de motor -usuario	Pruebas de funcionamiento y control de velocidad de un motor mediante teclado, lenguaje de programación de PIC e interfaz usando LabVIEW.

Tabla 1.: Análisis de la hipótesis del tema de tesis.

Fuente: Autores, 2013

1.6. Variables e indicadores

- Diseño y construcción de un entrenador didáctico con sistemas microprocesados que permite el uso de tecnología inalámbrica en las clases de microprocesados.
- Implementación de entrenadores didácticos en los laboratorios de Ingeniería Electrónica Digital de la UPS, que han constituido una herramienta eficaz en la utilización del docente en las prácticas de la materia de microprocesados.
- La adecuación técnica en los laboratorios de Electrónica Digital permitirá en funcionamiento correcto de los entrenadores didácticos.

1.7. Población y muestra

POBLACIÓN - es el conjunto total de individuos, objetos o medidas que poseen algunas características comunes observables en un lugar y en un momento determinado. Por tanto la población son los estudiantes de la Universidad Politécnica Salesiana y los docentes e ingenieros electrónicos de la misma.

MUESTRA - la muestra es un subconjunto fielmente representativo de la población, que serían los estudiantes de Ingeniería Electrónica de la materia de Sistemas Microprocesados II y sus docentes asignados, esta muestra puede ser seleccionada de forma:

- **ALEATORIA** – Si se selecciona al azar y cada estudiante tiene igual oportunidad de ser incluido.
- **ESTRATIFICADA** - cuando se subdivide en estratos o subgrupos según las variables o características que se pretenden investigar y es asignada por el docente mediante prácticas.

- **SISTEMÁTICA** - cuando se establece un patrón o criterio al seleccionar una práctica. Es decir al seleccionar una muestra lo que se hace es estudiar una parte o un subconjunto de la población, pero que la misma sea lo suficientemente representativa de ésta para que luego pueda generalizarse con seguridad de ellas a la población.

1.8. Alcance de la propuesta.

La propuesta de este plan de Tesis está enfocada en la automatización y mejora de procesos prácticos con tecnologías inalámbricas e interfaces de control moderno para evitar el uso de estándares ambiguos u obsoletos en los laboratorios electrónicos con el fin de generar a futuro buenas propuestas innovadoras en la mayoría de microempresas o de grandes empresas en el marco competitivo. Como parte de esta Tesis, los entrenadores didácticos con su respectiva interfaz se enmarcarán en la utilización de las tecnologías Bluetooth, X-Bee, GSM, Data Logger, Ethernet, GPS y la interfaz de control de velocidad de un motor de corriente alterna usando LabVIEW y la NI MyDAQ-USB-6009. El aporte investigativo estará básicamente en las diferentes plataformas, protocolos inalámbricos, lenguaje de programación de PIC, circuitería electrónica-analógica y su soporte para realizar diferentes prácticas en tiempo real.

Esta Tesis tiene como fin reunir todas las soluciones posibles a los requerimientos de los estudiantes, la creación de un manual de prácticas para el manejo de los módulos entrenadores, además de agregar futuras interfaces, métodos, etc., que exige en ámbito tecnológico.

CAPÍTULO II

FUNDAMENTOS TEÓRICOS

2.1 PICs DE MICROCHIP

Gonzáles (1998)

“Un microcontrolador, es un dispositivo electrónico encapsulado en un chip, capaz de ejecutar un programa, este reúne un solo integrado: microprocesador, memoria de programa, memoria de datos y puertos de entrada/salida. En algunos casos dispone de otras características especiales como: puertos serie, comparadores, convertidores analógico-digitales, etc.”¹

Un microcontrolador ejecuta instrucciones. El conjunto de instrucciones es lo que llamamos programa. Las instrucciones son leídas de la memoria de programa para ejecutarlas una detrás de otra. La memoria de programa contiene las instrucciones que queremos que el microcontrolador ejecute.

Programar un microcontrolador consiste en introducir el programa en la memoria del microcontrolador. Las instrucciones son operaciones simples como sumar, restar, escribir en un puerto, activar un bit de un dato, etc.

Mediante estas instrucciones básicas podemos realizar operaciones más complejas y así llegar al objetivo de la aplicación.

Esta TESIS de Grado se centrará en los microprocesadores de la casa Microchip Technology, es decir los PICs. Este tipo de microprocesadores están muy extendidos actualmente en el mercado gracias a su gran variedad y bajo coste. Otra razón del éxito de los PICs es su utilización, ya que una vez que se aprendió a utilizar uno, conociendo su arquitectura y juego de instrucciones, es muy fácil emplear otro modelo diferente.

¹ **Aplicaciones de los microcontroladores PIC de Microchip**, José Adolfo González, Editorial McGraw Hill, 1998.

2.1.1 Características de los PICs.

Las características más destacadas de los PICs se las enumera de la siguiente manera:

1. La arquitectura del procesador sigue el modelo Harvard.

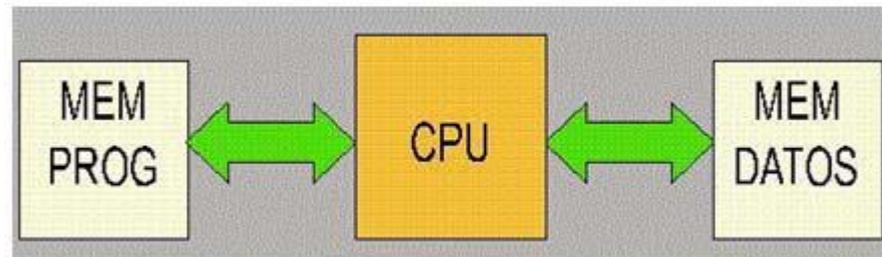


Figura 1.: Arquitectura del procesador –modelo Harvard
Fuente: Libro Microcontroladores PIC, Editorial McGraw Hill, 2013

Inicialmente, las computadoras y microprocesadores siguen el modelo propuesto por John Von Neumann, en el cual la unidad central de proceso, o CPU, está conectada a una memoria única que contiene las instrucciones del programa y los datos. El tamaño de la unidad de datos o instrucciones está fijado por el ancho del bus de la memoria. Esto limita la velocidad de operación del microprocesador, ya que no se puede buscar en la memoria una nueva instrucción, antes de que finalicen las transferencias de datos que pudieran resultar de la instrucción anterior.

En los microprocesadores PIC se utiliza el modelo Harvard. Este tipo de arquitectura conecta de forma independiente y con dos buses distintos la memoria de instrucciones y la de datos:

2. Técnica de segmentación ("pipe-line") en la ejecución de las instrucciones:
La segmentación permite al procesador realizar al mismo tiempo la ejecución de una instrucción y la búsqueda del código de tal manera que se puede ejecutar cada instrucción en un ciclo (un ciclo de instrucción equivale a cuatro ciclos de reloj).

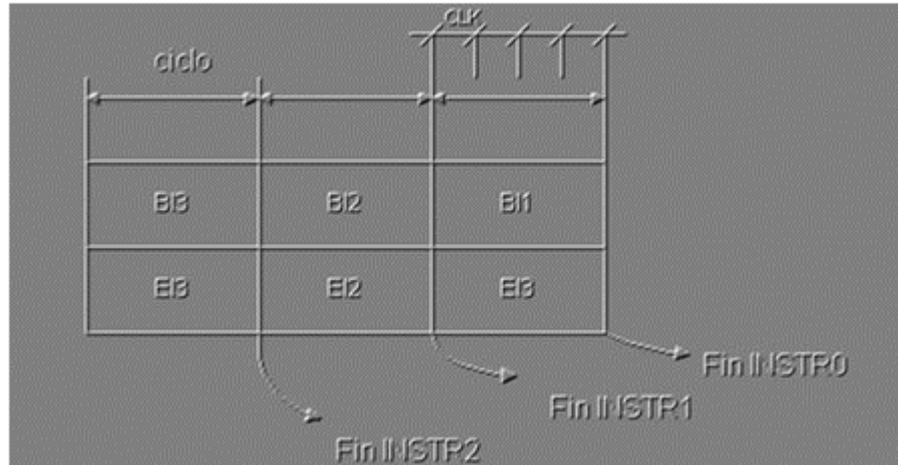


Figura 2.: Técnica de segmentación “pipe-line”

Fuente: Libro Microcontroladores PIC, Editorial McGraw Hill, 2013

La segmentación permite al procesador ejecutar cada instrucción en un ciclo de instrucción equivalente a cuatro ciclos de reloj. En cada ciclo se realiza la búsqueda de una instrucción y la ejecución de la anterior.

Las instrucciones de salto ocupan dos ciclos al no conocer la dirección de la siguiente instrucción hasta que no se haya completado la de bifurcación.

3. El formato de todas las instrucciones tiene la misma longitud. Todas las instrucciones de los microcontroladores de la gama baja tienen una longitud de 12 bits. Gama media 14 bits y más las de la gama alta. Esta característica es muy ventajosa en la optimización de la memoria de instrucciones y facilita enormemente la construcción de ensambladores y compiladores.
4. Procesador RISC (Computador de Juego de Instrucciones Reducido) Dependiendo de la gama del procesador (baja, media o alta) tienen más o menos número de instrucciones. Los modelos de la gama baja disponen de un repertorio de 33 instrucciones, 35 los de la gama media y unas 76 los de la alta.
5. Todas las instrucciones son ortogonales
6. Cualquier instrucción puede manejar cualquier elemento de la arquitectura como fuente o como destino.

7. Arquitectura basada en un banco de registros. Esto significa que todos los objetos del sistema (puertos de E/S, temporizadores, posiciones de memoria, etc.) están implementados físicamente como registros.

2.1.2 Gamas de PICs

Existen actualmente grandes cantidades de aplicaciones que se puede realizar con PICs, aplicaciones sencillas en las cuales no necesitamos muchos recursos y aplicaciones más complejas en las cuales necesitamos microcontroladores muy potentes, por ello y siguiendo esta filosofía, la empresa Microchip fabrica tres tipos de gamas de microcontroladores PIC para atender todas las aplicaciones, microcontroladores de gama baja, gama media y gama alta. Así, hay disponibles microcontroladores sencillos y baratos para atender las aplicaciones simples y otros complejos y más costosos aplicaciones complejas y de mayor ámbito de construcción. Existen dos arquitecturas utilizadas en la fabricación de microcontroladores:

1ª. Microcontroladores de arquitectura cerrada

En este tipo de arquitectura el microcontrolador tiene unos recursos específicos los cuales no permiten ningún tipo de modificación, es decir, no admiten ningún tipo de variaciones ni de ampliaciones. La aplicación a la que se destina debe encontrar en su estructura todo lo que precisa y, en caso contrario, hay que desecharlo. Microchip ha elegido principalmente este modelo de arquitectura.

2ª. Microcontroladores de arquitectura abierta

Este tipo de microcontroladores aparte de tener una estructura interna determinada, permiten ampliación emplear sus líneas de E/S para sacar al exterior los buses de datos, direcciones y control, con lo que se posibilita la ampliación de la memoria y las E/S con circuitos integrados externos. Microchip dispone de modelos PIC con arquitectura abierta, sin embargo, esta alternativa se escapa de la idea de un microcontrolador incrustado y se asemeja a la solución que emplean los clásicos microprocesadores.

2.1.3 PIC 18F4550

Este es el PIC que será utilizado en la realización de cada una de las prácticas con el Entrenador de didáctico, entre las características del 18F4550 tenemos:

- Tecnología nanoWatt, funciones de bajo consumo y ahorro de energía
- Voltaje de operación 4.2V a 5.5V
- Microcontrolador con módulo USB 2.0. Soporta Low speed 1.5 Mb/s y full speeds 12Mb/s.
- 1kB de memoria de doble acceso vía USB
- 35 pines I/O disponibles
- Memoria de programa flash de 32 kB
- RAM de 2048 Bytes
- EEPROM de datos de 256 Bytes
- Velocidad de la CPU 12 MIPS
- Oscilador externo de dos modos hasta 48 MHz
- Oscilador interno seleccionable entre 8 frecuencias desde 31kHz hasta 8MHz
- Oscilador secundario con Timer 1 de hasta 32kHz
- Opciones de oscilador dual permiten que la velocidad de la CPU y del módulo USB sean diferentes
- ADC de 10 bits y 13 canales
- 4 Timer (desde Timer0 a Timer3). Uno de 8 bits y 3 de 16 bits
- 2 módulos de captura/comparación/PWM
- EUSART, SPP, SPI, I²C.
- 20 fuentes de interrupciones (3 externas)
- Resistencias de pull-ups en el puerto B programables
- Función del pin MCLR opcional
- Brown-out Reset de valor programable
- Power-on Reset, Power-up Timer y Oscillator Start-up Timer
- Soporta 100,000 ciclos de borrado/escritura en memoria flash
- Soporta 1,000,000 ciclos de borrado/escritura en memoria EEPROM
- Retención de datos mayor a 40 años
- Protección de código y datos programable
- Encapsulado DIP de 40 pines

2.1.3.1 Distribución de pines PIC 18F4550

Mediante la Fig. 3 se muestra la distribución de pines:

40-Pin PDIP

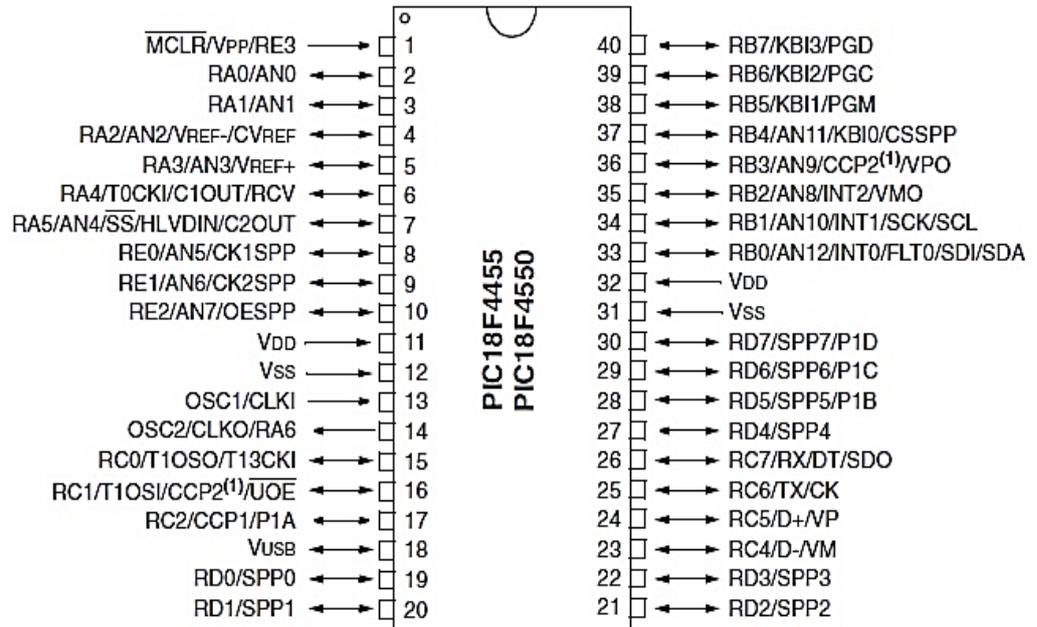


Figura 3. Distribución de pines del PIC 18F4550

Fuente: Mis primeros pasos con el PIC18F4455, http://picmania.garcia-cuervo.net/invitados_primer18f4550.php, 2013

2.1.3.2 Arquitectura y estructura interna

El PIC 18F4550 posee arquitectura tipo Harvard, así dispone de diferentes buses para acceder a la memoria de programa o memoria de datos con la finalidad de ejecutar una instrucción, mientras se lee de la memoria de programa la siguiente instrucción, es decir se realiza de manera simultánea.

Su bus de memoria de programa está distribuido por: 21 líneas de dirección, 16 líneas para instrucciones y 8 para datos; mientras que el bus de memoria de datos posee: 12 líneas de dirección y 8 líneas de datos.

La estructura interna del dispositivo se encuentra distribuida de la siguiente manera mediante este diagrama de bloques de la Fig. 4

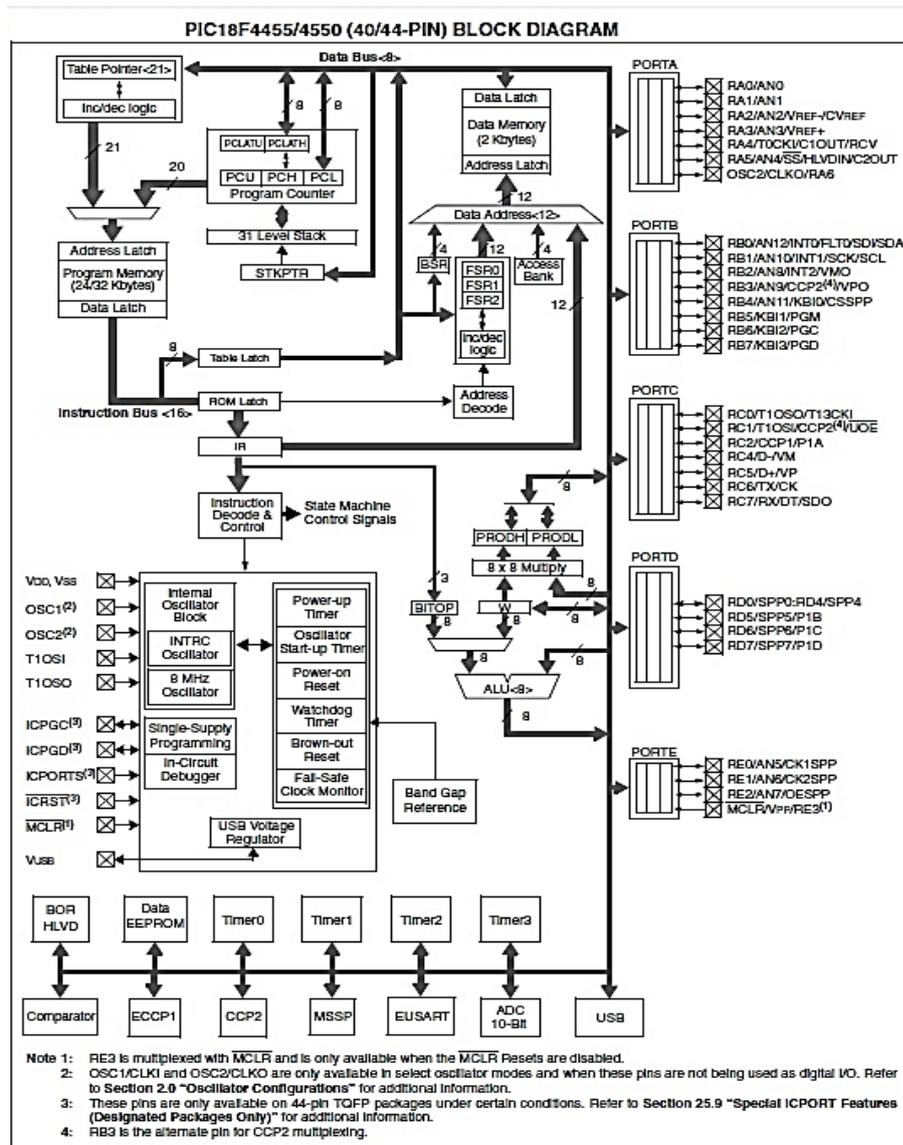


Figura 4. Diagrama de bloques PIC 18F4550

Fuente: PIC18F4455, <http://todoelectrodo.blogspot.com/2013/02/pic-18f4550.html>, 2013

2.1.4 Herramientas de desarrollo

Existen una serie de herramientas de desarrollo y totalmente gratuitas que se pueden descargar desde su página web². Con esta herramienta se procede a programar los microcontroladores PIC y así realizar los proyectos con estos pequeños chips.

Para el desarrollo de la Tesis de ingeniería se utilizará la herramienta de programación MICROCODE STUDIO.

2.1.5 Programación con MicroCode Studio y PICBasic

MicroCode Studio es una aplicación muy potente de desarrollo integrado (IDE), diseñado por MicroEngineering Labs³. Al ingresar a la página(http://melabs.com/resources/win_ide.htm) permite descargar de manera gratuita el archivo instalador



Figura 5: Link de archivo MicroCode Studio

Fuente: microEngineering Labs, Inc., http://melabs.com/resources/win_ide.htm, 2013

Adicional se debe instalar el compilador PICBasic que es muy necesario para toda la programación de las prácticas de la Tesis de los Entrenadores Didácticos.

² Microchip, página web: <http://www.microchip.com>

³ MicroEngineering, <http://www.microengineeringlabs.com>

A continuación se describe el proceso de instalación de MicroCode Studio y PICBasic:

2.1.5.1 Proceso de instalación

El primer paso para la instalación es ejecutar el archivo mcsinstall.exe, el cual inicia el proceso de instalación.



Figura 6.: Proceso de instalación inicial
Fuente: Microcode_PICbasic.pdf, 2013

Clic en siguiente para aceptar el dialogo acerca de la licencia,

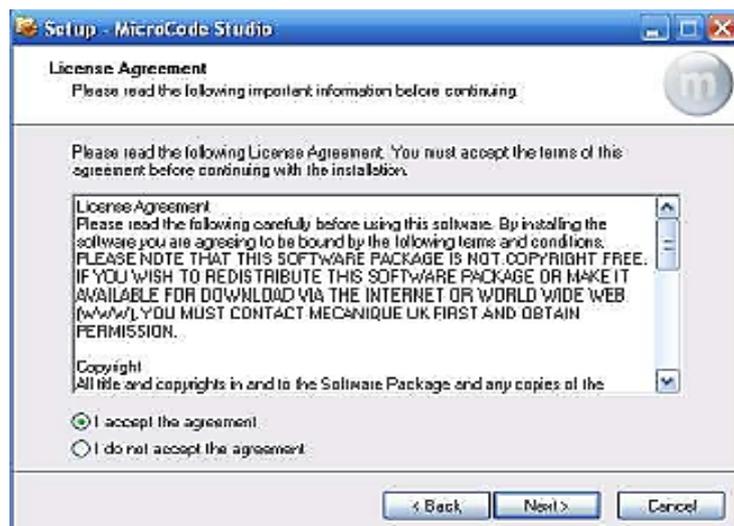


Figura 7.: Aceptación de licencia de MicroCode
Fuente: Microcode_PICbasic.pdf, 2013

Luego se observa la ruta por defecto del instalador de Micro Code Studio, por lo general se instala en la unidad C: y se crea una carpeta llamada Mecanica/MCS.



Figura 8.: Carpeta de enrutamiento de MicroCode
Fuente: Microcode_PICbasic.pdf, 2013

Por último una vez más damos clic en “Next” para que el proceso de instalación sea completado.

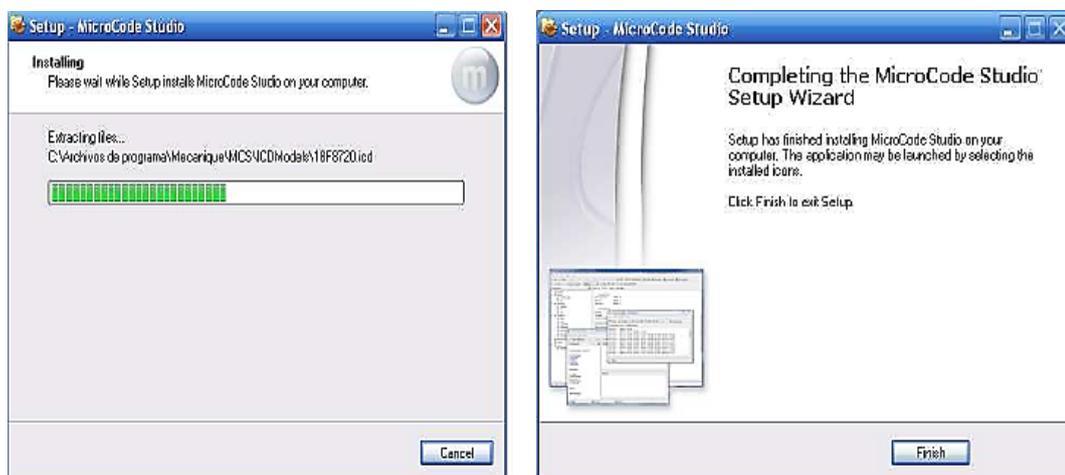


Figura 9.: Proceso final de instalación del MicroCode
Fuente: Microcode_PICbasic.pdf, 2013

Para poder trabajar correctamente con el Micro Code se debe instalar el compilador PIC Basic⁴ o a su vez la carpeta que contiene la librería de los microcontroladores como en la figura la carpeta PBP247

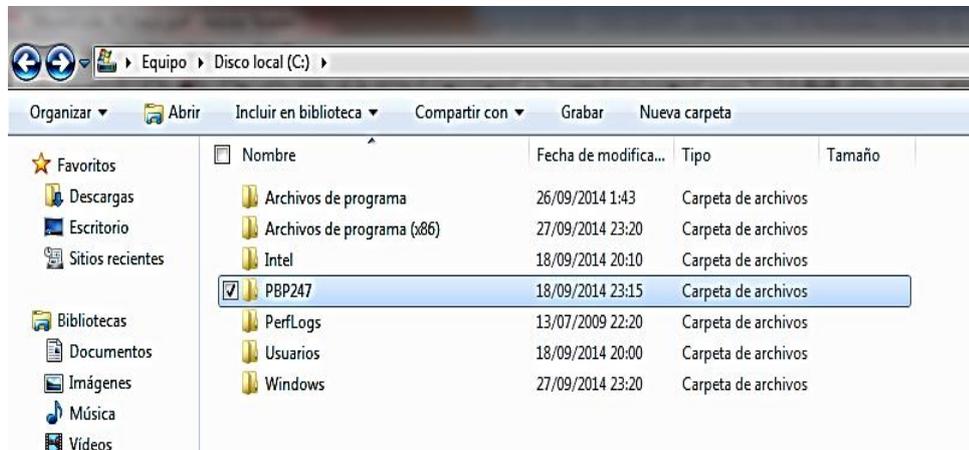


Figura 10.: Ubicación de la carpeta para la librería de compilación
Fuente: Autores, 2014

Se debe abrir el programa y hacer referencia a la carpeta donde se encuentran las librerías.

Abrir el programa MicroCode Studio, dar clic en “View” y luego en “Compile and Program Options...”

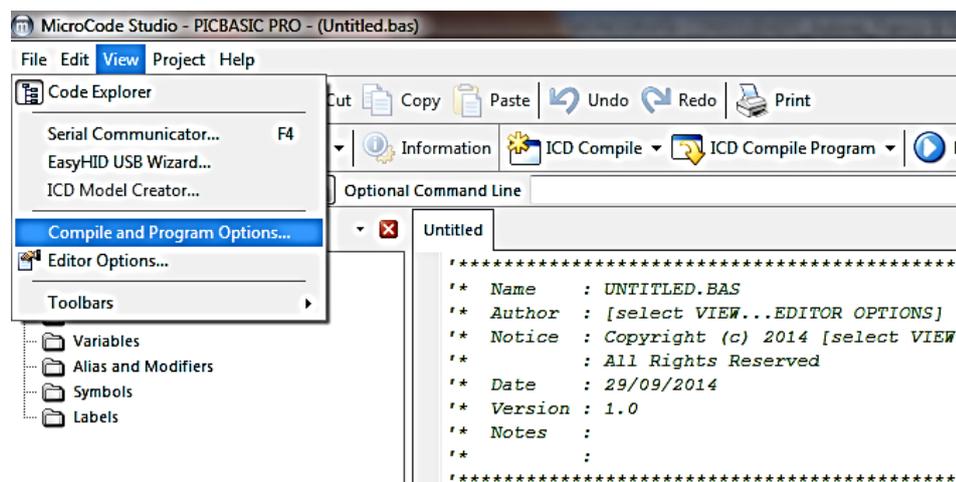


Figura 11.: Configuración en la opción de compilación y programación
Fuente: Autores, 2014

⁴ PIC Basic: Compilador que permite la generación del código que será cargador en el microcontrolador PIC

Al abrirse la ventana “Compile and Program Options” se debe buscar la carpeta manualmente elegirla y dar clic en “OK”. De esta manera se da por terminada la configuración y a trabajar.

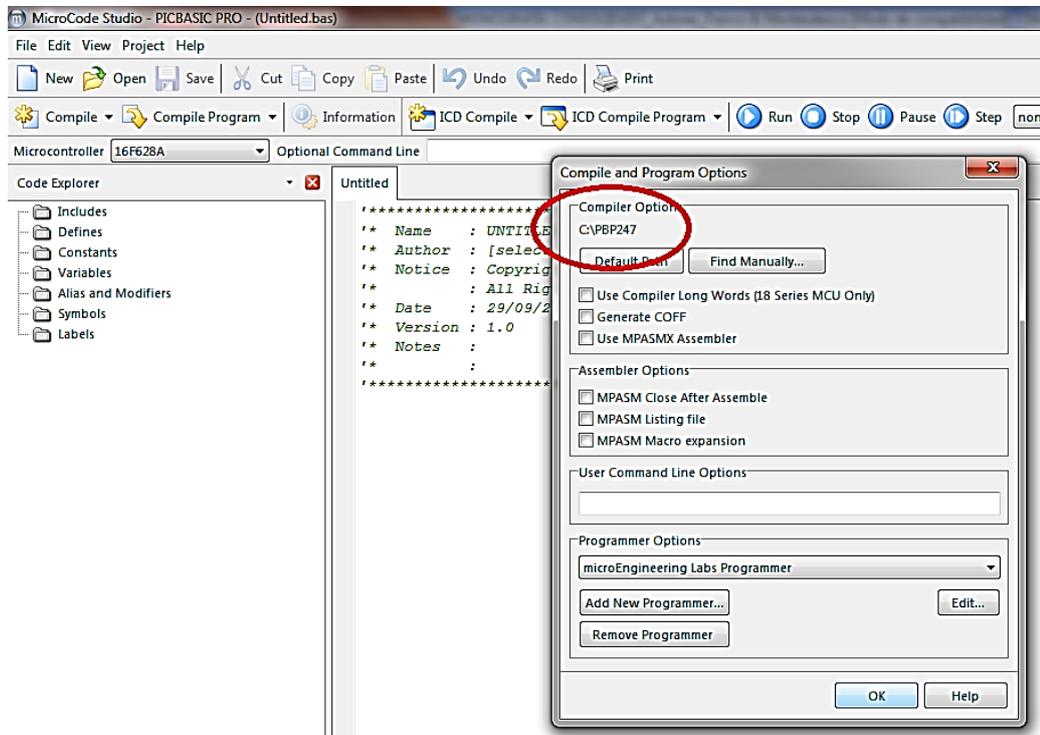


Figura 12.: Configuración manual de la opción de compilación
Fuente: Autores, 2014

2.1.5.2 Estructura básica del programa

En la figura siguiente se detalla una estructura básica de un programa en Microcode Studio con la finalidad de crear programas bien estructurados en las prácticas de esta Tesis así como también encontrar fácilmente errores de programación.

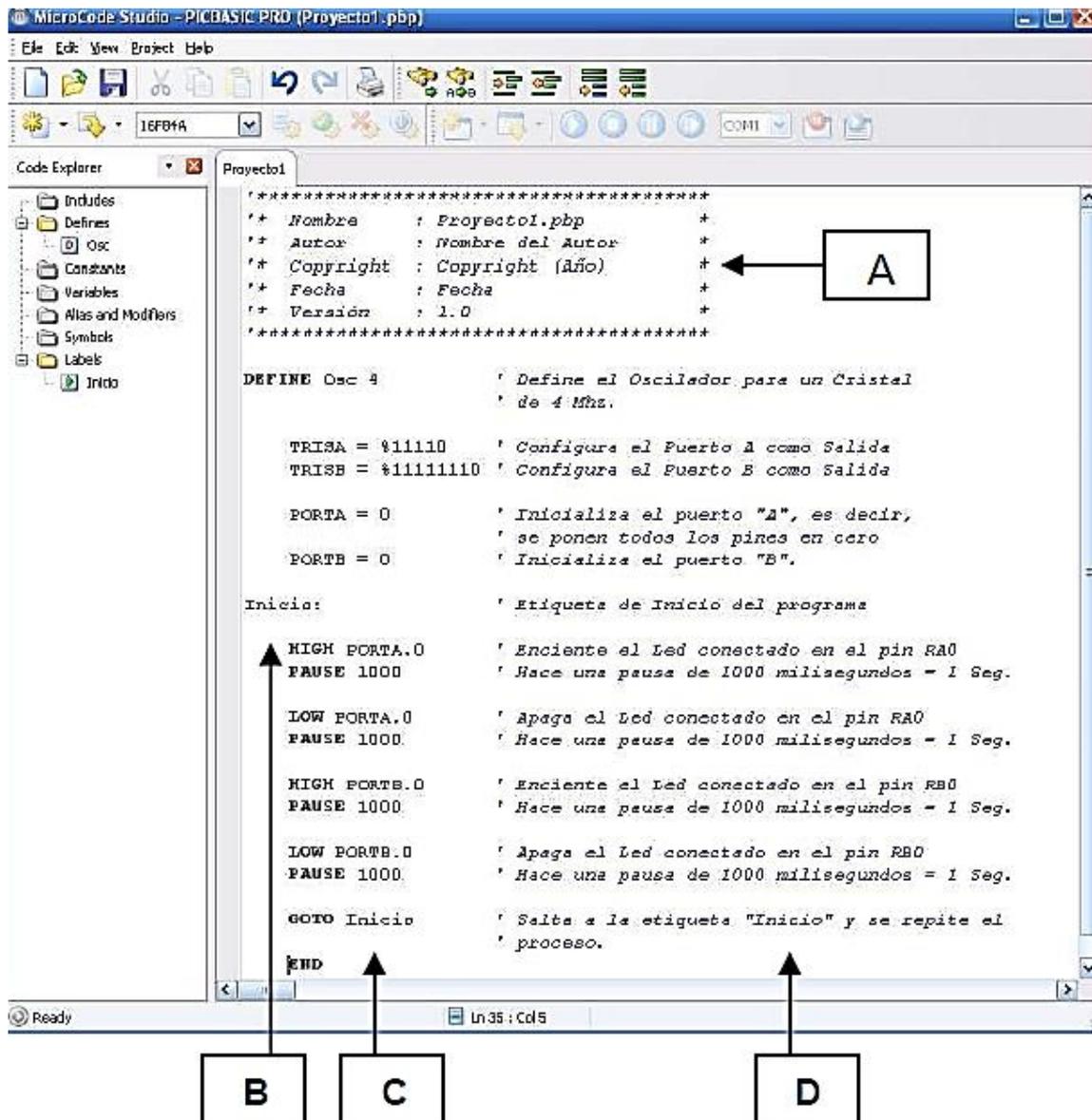


Figura 13.: Esquema de estructura de programa
Fuente: Autores, 2014

Sección A: Encabezado del programa, información importante, autores, fecha de elaboración, últimos cambios, objetivo breve del programa o la sección de programación. Al momento de comentar debemos utilizar una comilla:

Ej.: ' Define el Oscilador para un Cristal

Sección B: Comienza en la columna cero del editor de texto, por lo general aquí se declaran definiciones, etiquetas de subrutinas las cuales pueden corresponder a puntos específicos en el programa, deben tener al final de cada una de ellas el símbolo “:” que define el final de la subrutina.

Sección C: En este ejemplo sencillo esta sección es destinada para la programación es sí, es decir las instrucciones de programa, las cuales están separadas de la columna cero a través de un espacio mínimo que debe ser mayor al que se ha dejado entre la sección A y B.

Sección D: Usada para toda clase de comentario que deseemos realizar acerca de alguna función específica en el programa, siempre cada comentario debe empezar con una “comilla simple”.

2.1.5.3 Subrutinas de programa

El papel desempeñado por una subrutina es muy importante, ya que facilita realizar un papel específico dentro del programa y también pueden ser llamadas cada vez que sean necesarias haciendo referencia a tu etiqueta, esta debe ir siempre al inicio de la subrutina tal como se muestra en la figura:

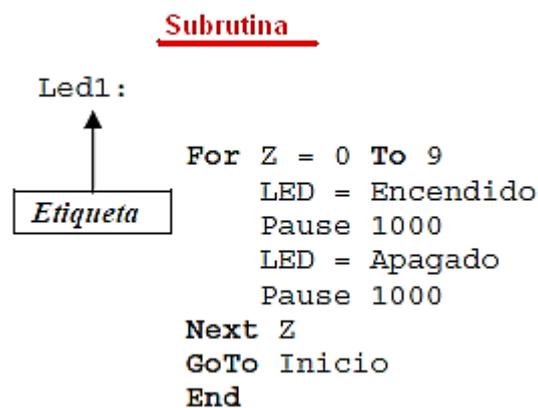


Figura 14.: Esquema de estructura de programa
Fuente: Autores, 2014

2.1.5.3. Operadores y componentes de PICBasic

Antes de programar en PICBasic se debe conocer las herramientas de programación que facilitaran el desarrollo de las prácticas de la Tesis de Ingeniería en el momento de la programación, entre ellas se mencionan las etiquetas, variables, constantes, símbolos, signos aritméticos, etc. A continuación se describen las más importantes, básicas y de mayor uso:

- **Define:** Es una de las directivas más importantes en PICBasic, permite establecer parámetros que permitirán que las prácticas ya definidas y las que se quiera generar sean sencillas en la programación, estos parámetros están vinculados directamente con los dispositivos externos al microcontrolador, por ejemplo: el parámetro para definir el uso de una pantalla LCD se deberán definir los puertos de conexión para el bus de dato y bus de control. Ejemplos:

Define LCD_DREG {puerto} ‘puerto de datos del LDC

Define LCD_RSREG {puerto} ‘puerto para RS (Register Select)

Define LCD_DBIT {bit} ‘bit inicial de puerto de datos

Define LCD_RWREG {puerto} ‘puerto para RW (Read/Write)

Define LCD_LINES {líneas} ‘Números de líneas de la LCD (1,2, 3...)

- **Variables:** Permiten el almacenamiento de datos temporales los cuales pueden ser consultados o verificados cuando así se lo amerite. Se utiliza la palabra **VAR** seguida del tipo de variable, el nombre de la misma lo elige el programador y el tipo se define según el dato que deseemos almacenar.

Nombre de variable	Var	Tipo de variable	Descripción
B4	Var	Bit	Valores de 0 y 1 únicamente
Temp	Var	Byte	Valores de 0 y 255 (8 bits)
Dig8	Var	Word	Valores entre 0 y 65535 (16 bits)

Tabla 2: Tipos de variables

Fuente: MicroCode_PICbasic.pdf, 2014

- **Arrays:** Son consideradas variables pero definidas por el número de elementos, por ejemplo un Array tipo Bit almacena 256, otro tipo Byte almacena hasta 96 y los tipo Word hasta 48 elementos, accedidos desde cualquier caso a través de un índice que se especifica entre corchetes:

Dato Var Byte [5]

- **Constantes y operadores aritméticos:** Las constantes definen un valor constante en la programación facilitando la búsqueda de errores en la funcionalidad, mientras que los operadores aritméticos nos ayudan a sumar, restar, multiplicar, etc., las variables o constantes entre otras funciones.

Ejemplo al declarar una constante:

Temp Max **CON** 300

Descripción de operadores aritméticos:

Operador	Descripción
+	Suma
-	Resta
*	Multiplicación
/	División
//	Residuo
=	Igual, asignar valores
>>	Desplazar a la derecha
<<	Desplazar a la izquierda
SIN	Seno del ángulo
COS	Coseno del ángulo
MIN	Mínimo de un número
MAX	Máximo de un número
DIG	Valor de un dígito para un número decimal
REV	Invertir Bit

Tabla 3: Operadores aritméticos básicos

Fuente: Autores, 2014

- **Operadores binarios, lógicos, de comparación:** Se describe a través de tablas con el ejemplo respectivo:

Operador	Descripción
&	AND lógico
	OR lógico
^	XOR lógico
~	NOT lógico

Tabla 4: Operadores binarios básicos
Fuente: Autores, 2014

Ejemplo:

Var5=%00101001

Entonces,

Var5 = Var5 & %00000001

En esta operación su resultado será Var5= %00000001

Operador	Descripción
AND	AND lógico
OR	OR lógico
XOR	XOR lógico
NOT	NOT lógico
NOT AND	NAND lógico
NOT OR	NOR lógico

Tabla 5: Operadores lógicos básicos
Fuente: Autores, 2014

Ejemplo:

IF Var4 = 2 **AND** Var6=4 **AND** Var8=10 **Then Goto** INICIO

Si se cumplen las tres condiciones, solo así entonces se realiza el proceso de la etiqueta INICIO

Operador	Descripción
=	Igual
<>	Diferente
>	Mayor que
<	Menor que
>=	Mayor o igual que
<=	Menor o igual que

Tabla 6: Operadores de comparación básicos
Fuente: Autores, 2014

2.2. BLUETOOTH

El estándar Bluetooth es una norma abierta que posibilita la conexión inalámbrica de corto alcance de voz y datos entre ordenadores, portátiles, agendas digitales personales, teléfonos móviles, impresoras, escáneres, cámaras digitales e incluso dispositivos de casa, a través de una banda disponible a nivel global (2,4 GHz). Para el uso de la Tesis de ingeniería la comunicación se hará entre dos entrenadores didácticos. Dicho de otro modo la tecnología Bluetooth es el sistema de comunicaciones sin hilos, la cual elimina el engorroso lío de cables de comunicación entre los diferentes dispositivos electrónicos. Principales objetivos:

- Facilitar las comunicaciones entre equipos móviles y fijos.
- Eliminar cables y conectores entre éstos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre nuestros equipos personales.

Bluetooth es el protocolo de comunicaciones diseñado especialmente para dispositivos de bajo consumo, con una cobertura baja y basada en transceptores de bajo coste.

Sin duda este protocolo permite comunicarse entre dispositivos siempre y cuando estén dentro de un determinado rango de alcance. Las comunicaciones se realizan por radiofrecuencia de forma que los dispositivos no tienen por qué estar alineados, pueden incluso estar en habitaciones separadas si la potencia de transmisión lo permite.

2.2.1. Clasificación del bluetooth.

La clasificación de los dispositivos Bluetooth según la potencia de transmisión y siendo compatibles los dispositivos de una clase con la otra son: "Clase 1", "Clase 2" o "Clase 3".

Clase	Potencia máxima permitida (mW)	Potencia máxima permitida (dBm)	Rango (aproximado)
Clase 1	100 mW	20 dBm	~100 metros
Clase 2	2.5 mW	4 dBm	~10 metros
Clase 3	1 mW	0 dBm	~1 metro

Tabla 7: Clasificación de bluetooth según potencia de transmisión

Fuente: Tecnología Bluetooth, <http://es.wikipedia.org/wiki/Bluetooth>, 2013

En cuanto al ancho de banda:

Versión	Ancho de banda
Versión 1.2	1 Mbit/s
Versión 2.0 + EDR	3 Mbit/s
UWB Bluetooth (propuesto)	53 - 480 Mbit/s

Tabla 8: Clasificación de bluetooth según ancho de banda

Fuente: Tecnología Bluetooth, <http://es.wikipedia.org/wiki/Bluetooth>, 2013

2.2.2. Operación de los dispositivos Bluetooth

Cuando un equipo bluetooth está dentro del radio de cobertura de otro, estos pueden crear un enlace entre ellos, ocho unidades bluetooth pueden comunicarse entre ellas y

forman lo que se denomina una Piconet. La unión de varias piconets se denomina Scatternet.



Figura 15: Tecnología Bluetooth

Fuente: <http://es.wikipedia.org/wiki/Bluetooth>, 2013

En todas las piconets sólo puede haber una unidad maestra que normalmente es quien inicia la conexión, el resto de unidades bluetooth se denominan esclavas.

“Cada unidad de la piconet utiliza su identidad maestra y reloj nativo para seguir en el canal de salto. Cuando se establece la conexión, se añade un ajuste de reloj a la propia frecuencia de reloj nativa de la unidad esclava para poder sincronizarse con el reloj nativo del maestro. El reloj nativo mantiene siempre constante su frecuencia, sin embargo, los ajustes producidos por las unidades esclavas para sincronizarse con el maestro, sólo son válidos mientras dura la conexión.”⁵

⁵ **Tecnología bluetooth**, [http:// masadelante.com](http://masadelante.com)

Dentro de la misma área pueden coexistir diversas piconets ya que cada piconet tiene una unidad maestra distinta con su propia secuencia de saltos de canales y de fase. A medida que tenemos más piconets en la misma área de cobertura, la probabilidad de colisión aumenta produciendo una degradación del espectro y reduciendo el rendimiento del sistema.

Una unidad bluetooth puede participar secuencialmente en varias piconets gracias al sistema TDM (división de tiempo multiplexada). Esto es posible siempre y cuando la unidad solo esté activa en una piconet a la vez. Para realizar este proceso, la unidad cuando se incorpora a la nueva piconet debe ajustar el offset de su reloj nativo y realizar los ajustes de configuración correspondientes a la nueva piconet. Cuando una unidad abandona una piconet, la esclava informa el maestro actual que ésta no estará disponible por un determinado periodo, que será en el que estará activa en otra piconet.

Durante su ausencia, el tráfico en la piconet entre el maestro y otros esclavos continúan igualmente.

Una unidad maestra también puede cambiar de piconet, pero en este caso el tráfico de la piconet en la cual está activa deja de tener tráfico hasta la vuelta de la unidad maestra. La maestra que entra en una nueva piconet, en principio, lo hace como esclava, a no ser que posteriormente ésta solicite actuar como maestra.

2.2.3. Enlace físico.

En la especificación Bluetooth se definen dos tipos de enlace:

- **Enlace de sincronización de conexión orientada (SCO)**
 - Conexión simétrica punto a punto entre maestro y esclavo.
 - Los paquetes de configuración nunca son retransmitidos.
 - El maestro utiliza slots de tiempo reservados a intervalos regulares.
 - El maestro puede soportar más de 3 enlaces simultáneos mientras que los esclavos 3 como máximo.

- Este tipo de enlaces es utilizado principalmente para transmitir información de voz con un ratio de transmisión de 64kB/s.

- **Enlace asíncrono de baja conexión (ACL)**

- Se aplica retransmisión de paquetes.
- Conexiones simétricas o asimétricas punto-multipunto entre maestro y esclavo.
- Conexión utilizada para la transmisión de datos.
- Se definen para este tipo de conexión los slots 1,3 y 5.
- La máxima velocidad de envío es de 721 kbit/s en una dirección y 57.6 kbit/s en la otra.

2.2.4. Perfiles

En el estándar bluetooth se han definido una serie de perfiles.

Los perfiles definen los roles y capacidades para aplicaciones específicas. Diferentes perfiles pueden abarcar diferentes capas y protocolos para diferentes grados de seguridad.

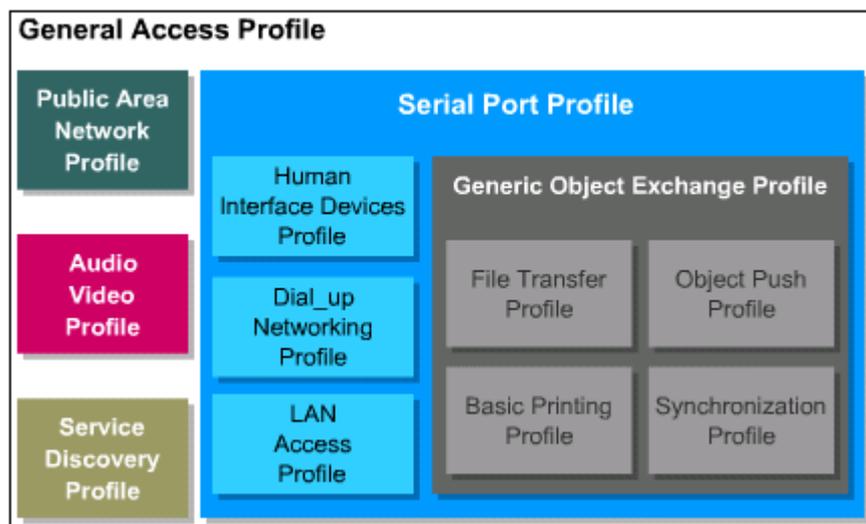


Figura 16: Perfil de acceso general

Fuente: Tecnología Bluetooth, <http://es.wikipedia.org/wiki/Bluetooth>, 2013

2.2.5. Módulo Bluetooth Inalámbrico Serial - Transceiver RF

Es un módulo tipo serie que permite trabajar como maestro o esclavo. Formado por el chip Bluetooth BC417 colocado en un PCB⁶ que permite optimizar el trabajo en placa Arduino o con cualquier microcontrolador tal como el PIC 18F4550 mediante la comunicación serial, su velocidad de comunicación es configurable a través de comandos especiales enviados al módulo.

2.2.5.1. Características:

- Puerto serie de funcionamiento 3.3V
- Durante su emparejamiento o conexión llega a los 30mA, durante la conexión y transmisión de datos o comunicación es de 8mA.
- Velocidad de transmisión fija de 9600, modificable mediante comando AT.
- Usa el chip BC417, protocolo bluetooth estándar V2.0, capa MAC IC con comunicaciones seriales.
- Vinculación sin problemas con una laptop o con un adaptador bluetooth portátil.

2.2.5.2. Distribución de pines:

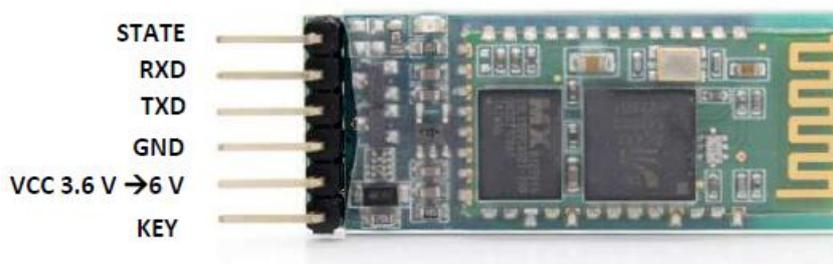


Fig. 17: Distribución de pines Bluetooth

Fuente: Bluetooth Transceiver RF, Bluetooth_TRx_Module_New.pdf, 2013

PIN	Descripción
STATE	No usar
RXD	Pin para la configuración como receptor.
TXD	Pin para la configuración como

⁶ PCB de sus siglas en inglés *Printed Circuit Board*, no es más que una es placa de circuito impreso.

	transmisor.
GND	Conexión a tierra
VCC 3.6V A 6V	PIN de conexión de voltaje de alimentación, Chip BC417 funciona con 3.3v, todo el módulo PCB se encuentra en un rango de 3.6V a 6V regulables.
KEY	Conectado a Vcc antes del swicheo del módulo

Tabla. 9: Descripción de pines Bluetooth

Fuente: Bluetooth Transceiver RF, Bluetooth_TRx_Module_New.pdf, 2013

2.2.5.3. Modo de operación, comandos AT

El módulo tiene dos formas de operación: modo de comando AT y modo de conexión. El modo de operación en que sea escogido puede ser distinguido por el parpadeo del LED que posee la tarjeta del módulo. A continuación se describe el modo de operación que será empleado en el desarrollo de la Tesis de Ingeniería.

Modo de comando AT

Es un modo del módulo donde se utiliza un conjunto de comandos (comandos AT) para instalar y configurar el módulo.

El módulo no puede ser detectado por otros dispositivos Bluetooth, todos los comandos se envían al módulo en serie como una cadena expresada de esta manera:

- Mayúsculas (mayúsculas y minúsculas) y cada comando debe ser terminado mediante "\r\n".
- Si el comando se entiende con éxito por el módulo, el módulo le responderá con una cadena de respuesta "OK".
- Si el comando presenta una incoherencia o no es correcto, el Módulo devuelve "ERROR ()" error con un código específico que define el tipo de este error.

2.2.5.4. Clasificación de los comandos AT:

- Comandos de prueba
- Comandos de preguntas de datos de dispositivos cercanos
- Comandos de indagación de datos de módulo
- Comandos de control y configuración de módulo

Comandos de prueba

Como su nombre lo indica, estos comandos permiten verificar si el módulo está respondiendo correctamente:

Comando	Respuesta	Parámetro
AT	OK	Ninguno
AT+RESET	OK	Ninguno

Tabla. 10: Comandos de pruebas en AT

Fuente: Bluetooth Transceiver RF, Bluetooth_TRx_Module_New.pdf, 2013

Ejemplo 1:

Respuesta del módulo

Cadena de comando enviada:

“AT\r\n”

Respuesta a la cadena:

“OK”

Ejemplo 2:

Reset del módulo

Cadena de comando enviada:

“AT+RESET\r\n”

Respuesta de la cadena:

“OK”

Comandos de preguntas de datos de dispositivos cercanos

- Recibe el nombre del dispositivo bluetooth remoto:

Comando	Respuesta	Parámetros
AT+RNAME?<parámetro1>	1. NAME:<Parámetro2> OK--- success 2. FAIL ---failure	Parametro1: Remote Bluetooth device address Parámetro2: Remote bluetooth device address

Tabla. 11: Comando de pregunta de datos

Fuente: Bluetooth Transceiver RF, Bluetooth_TRx_Module_New.pdf, 2013

Ejemplo:

Dirección del dispositivo: 12:23:45:89 Nombre: PRUEBA

Cadena enviada:

“AT+RNAME?12,23, 45,89\r\n”

Cadena de respuesta:

“+RNAME: PRUEBA

OK”

- Obtiene la dirección Bluetooth del dispositivo emparejado utilizada más recientemente

Comando	Respuesta	Parámetros
AT+MRAD?	+ MRAD : <Parámetro> OK	Parámetro: Dirección de Bluetooth del dispositivo recientemente autenticado

Tabla. 12: Comando de obtención de dirección de bluetooth

Fuente: Bluetooth Transceiver RF, Bluetooth_TRx_Module_New.pdf, 2013

Ejemplo:

Cadena enviada:

“AT+MRAD?\r\n”

Cadena de respuesta:

“+MRAD: 12:23:45:89
OK”

- Pregunta y emite información del dispositivo Bluetooth

Comando	Respuesta	Parámetros
AT+INQ	+ INQ: <Parámetro1> , <Parámetro2>, <Parámetro3> OK	Parámetro1: Dirección del dispositivo Parámetro2: Tipo de dispositivo Parámetro3: Intensidad de la señal RSSI

Tabla. 13: Comando para obtener información del dispositivo

Fuente: Bluetooth Transceiver RF, Bluetooth_TRx_Module_New.pdf, 2013

Ejemplo:

Cadena enviada:

“AT+INQ\r\n”

Cadena de respuesta:

“+INQ: 12:23:45:89
+INQ: bb: cc: dd: ee
+INQ: 43:44:25:55
OK”

Comandos de indagación de datos de módulo.

- Obtiene la versión del módulo

Comando	Respuesta	Parámetros
AT+VERSION?	+ VERSION : <Parámetro> OK	Parámetro: Número de la versión del dispositivo/módulo

Tabla. 14: Comando para obtener versión del módulo Bluetooth

Fuente: Bluetooth Transceiver RF, Bluetooth_TRx_Module_New.pdf, 2013

Ejemplo:

Cadena enviada:

“AT+VERSION?\r\n”

Cadena de respuesta:

“+VERSION: 3.1.235562
OK”

- Establecer/preguntar nombre del dispositivo

Comando	Respuesta	Parámetros
AT+NAME=<Parámetro>	OK	Parámetro: Nombre del dispositivo.
AT+NAME?	1.+NAME: <Parámetro> OK ---- Exitoso 2. FAIL ----Fallido	Definido: “MÓDULO1”

Tabla. 15: Establecer/preguntar nombre del dispositivo Bluetooth

Fuente: Bluetooth Transceiver RF, Bluetooth_TRx_Module_New.pdf, 2013

Ejemplo:

Para establecer el nombre:

Cadena enviada:

“AT+NAME=MÓDULOBI\r\n”

Cadena de respuesta:

“OK”

Para preguntar el nombre:

Cadena enviada:

“AT+NAME?\r\n”

Cadena de respuesta:

“+NAME: MÓDULOBI
OK”

➤ Establecer/preguntar clave

Comando	Respuesta	Parámetros
AT+PSWD=<Parámetro>	OK	Parámetro: Clave
AT+PSWD?	+PSWD: <Parámetro> OK ----- Exitoso	Definido: “1234”

Tabla. 16: Establecer/preguntar clave

Fuente: Bluetooth Transceiver RF, Bluetooth_TRx_Module_New.pdf, 2013

➤ Eliminar todos los dispositivos apareados de la lista

Comando	Respuesta	Parámetros
AT+RMAAD	OK	Ninguno

Tabla. 17: Eliminar dispositivos apareados

Fuente: Bluetooth Transceiver RF, Bluetooth_TRx_Module_New.pdf, 2013

- Obtener el estado de trabajo del módulo Bluetooth

Comando	Respuesta	Parámetros
AT+STATE?	+STATE:<Parámetro> OK	Parámetro: Evalúa y retorna el estado de trabajo del módulo: “INITIALIZED” ----Inicializado “READY” ----Listo/preparado “PAIRABLE” ----Emparejable “PAIRED” ----Emparejado “INQUIRING” ----preguntando/consultando “CONNECTING” ----Conectando “CONNECTED” ----Conectado “DISCONNECTED” ----Desconectado “UNKNOW” ----Desconocido

Tabla. 18: Estados de trabajo del módulo Bluetooth

Fuente: Bluetooth Transceiver RF, Bluetooth_TRx_Module_New.pdf, 2013

- Eliminar dispositivo vinculado en la lista de apareamiento

Comando	Respuesta	Parámetros
AT+RMSAD=<Parámetro>	OK	Parámetro: Dirección del dispositivo Bluetooth

Tabla. 19: Comando para eliminar dispositivo vinculado en lista

Fuente: Bluetooth Transceiver RF, Bluetooth_TRx_Module_New.pdf, 2013

- Establecer/consultar el tipo de dispositivo.

Cada clase de un módulo bluetooth determina si este módulo es parte de un celular, auriculares o una laptop, ya que tienen un valor definido para identificarlas.

Comando	Respuesta	Parámetros
AT+CLASS=<Parámetro>	OK	Parámetro: tipo de dispositivo
AT+CLASS?	1. +CLASS: <Parámetro> OK ----Exitoso 2.FAIL ----Fallido	Tipo de dispositivo Bluetooth es un parámetro 32 byte indica el tipo de dispositivo y qué tipo puede ser soportado. Default: 0

Tabla. 20: Establecer/preguntar tipo de dispositivo

Fuente: Bluetooth Transceiver RF, Bluetooth_TRx_Module_New.pdf, 2013

- Búsqueda del dispositivo vinculado a la lista

Comando	Respuesta	Parámetros
AT+FSAD=<Parámetro>	1. OK ---Exitoso 2.FAIL ---Fallido	Parámetro: Dirección del dispositivo Bluetooth

Tabla. 21: Búsqueda del dispositivo Bluetooth

Fuente: Bluetooth Transceiver RF, Bluetooth_TRx_Module_New.pdf, 2013

Comandos de control y configuración del módulo.

- Restauración estado predeterminado

Comando	Respuesta	Parámetros
AT+ORGL	OK	Ninguno

Tabla. 22: comando de restauración a estado predeterminado

Fuente: Bluetooth Transceiver RF, Bluetooth_TRx_Module_New.pdf, 2013

Device type	0
Inquire code	0x009e8b33
Module work mode	Slave Mode
Connection mode	Connect to the Bluetooth device specified
Serial parameter	Baud rate: 38400 bits/s; Stop bit: 1; Parity bit: None
Passkey	1234
Device name	H-C-2010-06-01

Tabla. 23: Parámetros de estado predeterminado del módulo Bluetooth

Fuente: Bluetooth Transceiver RF, Bluetooth_TRx_Module_New.pdf, 2013

- Establecer/consultar rol del módulo Bluetooth.

Comando	Respuesta	Parámetros
AT+ROLE=<Parámetro>	OK	Parámetro: 0 ----Esclavo
AT+ROLE?	+ROLE: <Parámetro> OK	1 ----Maestro 2 ----Esclavo-bucle Predeterminado: 0

Tabla. 24: Establecer/consultar rol del módulo Bluetooth

Fuente: Bluetooth Transceiver RF, Bluetooth_TRx_Module_New.pdf, 2013

- Establecer/consultar - parámetros serial.

Comando	Respuesta	Parámetros
<p>AT+UART=<Parámetro1>, <Parámetro2>, <Parámetro3></p>	<p>OK</p>	<p>Parámetro1: velocidad de transmisión (bits/s)</p> <p>El valor(decimal) puede ser uno de los siguientes:</p> <p>4800 9600 19200 38400 57600 115200 23400 460800 921600 1382400</p> <p>Parámetro2: bit de parada:</p> <p>0 ----1 bit 1 ----2 bit</p> <p>Parámetro3: Bit de paridad</p> <p>0 ----Nada 1 ----Paridad impar 2 ----Paridad par</p>

Tabla. 25: Establecer/consultar - parámetros serial

Fuente: Bluetooth Transceiver RF, Bluetooth_TRx_Module_New.pdf, 2013

- Establecer/consultar – modo de conexión.

Comando	Respuesta	Parámetros
AT+CMODE=<Parámetro>	OK	Parámetro1: velocidad de transmisión (bits/s) 0 -----Conecta el módulo a una dirección Bluetooth específica. 1 -----Conecta el módulo a cualquier dirección 2 -----Conecta de modo: Slave-Loop Módulo de conexión predeterminada: 0

Tabla. 26: Establecer/consultar - modo de conexión

Fuente: Bluetooth Transceiver RF, Bluetooth_TRx_Module_New.pdf, 2013

- Establecer paridad

Permite al módulo ingresar al estado “PAIRED” con otro dispositivo Bluetooth.

Comando	Respuesta	Parámetros
AT+PAIR=<Parámetro1>, <Parámetro2>	1. OK ----Éxito 2. FAIL ----Fallido	Parámetro1: Dirección Bluetooth del dispositivo remoto. Parámetro2: Límite de tiempo de conexión(segundos)

Tabla. 27: Comando para establecer paridad

Fuente: Bluetooth Transceiver RF, Bluetooth_TRx_Module_New.pdf, 2013

➤ Conexión del dispositivo

Permite al módulo ingresar al estado “CONNECTED”, donde los datos seriales impuestos en el PIN transmisor son transmitidos inmediatamente.

Comando	Respuesta	Parámetros
AT+LINK=<Parámetro1>	1. OK ----Éxito 2. FAIL ----Fallido	Parámetro1: Dirección Bluetooth del dispositivo remoto.

Tabla. 28 Comando de conexión del dispositivo

Fuente: Bluetooth Transceiver RF, Bluetooth_TRx_Module_New.pdf, 2013

2.2.5.5. Detección del modo AT de acuerdo a la tasa de parpadeo del LED

En la siguiente figura se puede notar al dispositivo en modo AT a través del LED el cual parpadea lentamente y se mantiene durante largos intervalos de tiempo.

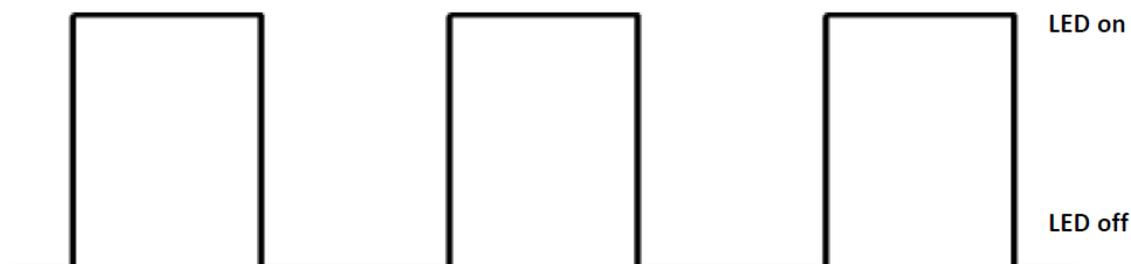


Fig. 18. Pulsos de LED parpadeando en modo AT

Fuente: Bluetooth Transceiver RF, Bluetooth_TRx_Module_New.pdf, 2013

2.3. TECNOLOGÍAS GSM/GPRS y GPS, MÓDULO GSM/GPRS/GPS V3.0

2.3.1. Sistema GSM

GSM (Global System for Mobile Communications), en español es Sistema Global para Comunicaciones Móviles. Ya en el nombre está implícita la clara intención de comunicar a las personas incluso en distintos países, crear algo mundial. Lo que se ha logrado con esta tecnología es crear a lo que se llama roaming internacional, algo más global, que no sólo abarque un país o ciertas zonas específicas del mismo. Es algo así como tener el mismo número para más de 150 países ya que es una tecnología satelital. Tal como se muestra en la figura es una estructura muy bien puesta.

2.3.1.1. Estructura de una red GSM

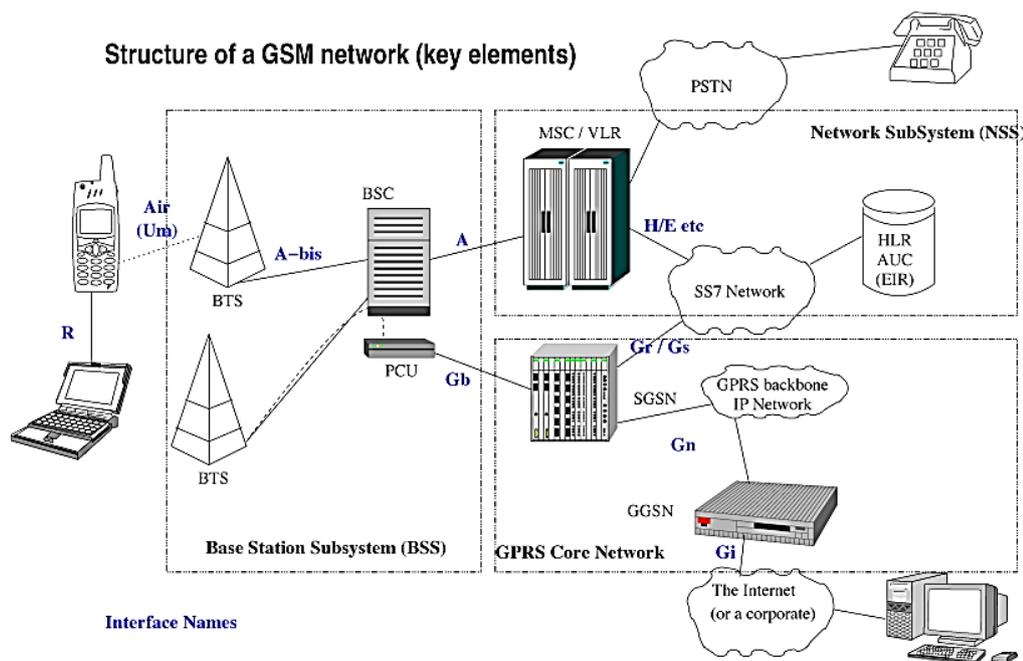


Figura 19: Estructura de la red GSM

Fuente: Sistema global para las comunicaciones móviles.

Una estructura GSM consiste en una red de células radioeléctricas continuas unas de otra que proporcionan cobertura completa a toda una red de servicio. Entre los elementos más importantes de una estructura GSM se destacan:

Estación base (BTS), a la que cada célula radioeléctrica pertenece y opera en un conjunto de canales de radio diferentes a los usados en las células adyacentes y que

se encuentran distribuidas según un plan celular. Un grupo de BTS's se encuentran conectado a un controlador de estaciones base (BSC), encargado de aspectos como el handover (traspaso del móvil de una célula a otra) o el control de potencia de las BTS's y de los móviles.

Controlador de estaciones base (BSC), es el encargado del manejo de toda la red de radio y supone una auténtica novedad respecto a los anteriores sistemas celulares.

Central de conmutación de móviles (MSC), varias BSC's se conectan una MSC, de esta manera se considera a una MSC como el corazón de la estructura GSM, ya que es aquí donde se realiza la inicialización, enrutamiento, control y finalización de las llamadas, así como también información de valores tarifarios definidos por el operador. Sirve además como interface entre varias redes GSM o entre una de ellas y las redes públicas de telefonía o datos.

Registro de posiciones base (HLR), Encargada del análisis de los niveles de suscripción, servicios suplementarios y localización actual, o más reciente de los móviles que pertenecen a la red local. En conjunto con la HRL trabaja el *centro de autenticación (AUC)*, que contiene la información por la que se comprueba la autenticidad de las llamadas para evitar operaciones fraudulentas, uso de tarjetas SIM robadas, etc.

Registro de posiciones de visitantes (VLR), se encarga del contenido de la información sobre los niveles de suscripción, servicios suplementarios y red de localización para un abonado que se encuentra o al menos se encontraba recientemente en otra zona visitada. Esta base de datos dispone también de información relativa a si el abonado se encuentra activo o no, lo que evita el uso improductivo de la red (envío de señales a una localización que se encuentra desconectada).

Registro de identidad de los equipos (EIR), almacena información sobre el tipo de estación móvil en uso y puede eludir que se realice una llamada cuando se detecte que ha sido robada, pertenece a algún modelo no homologado o sufre de algún fallo susceptible de efectuar negativamente a la red.

2.3.1.2. Elementos del sistema GSM

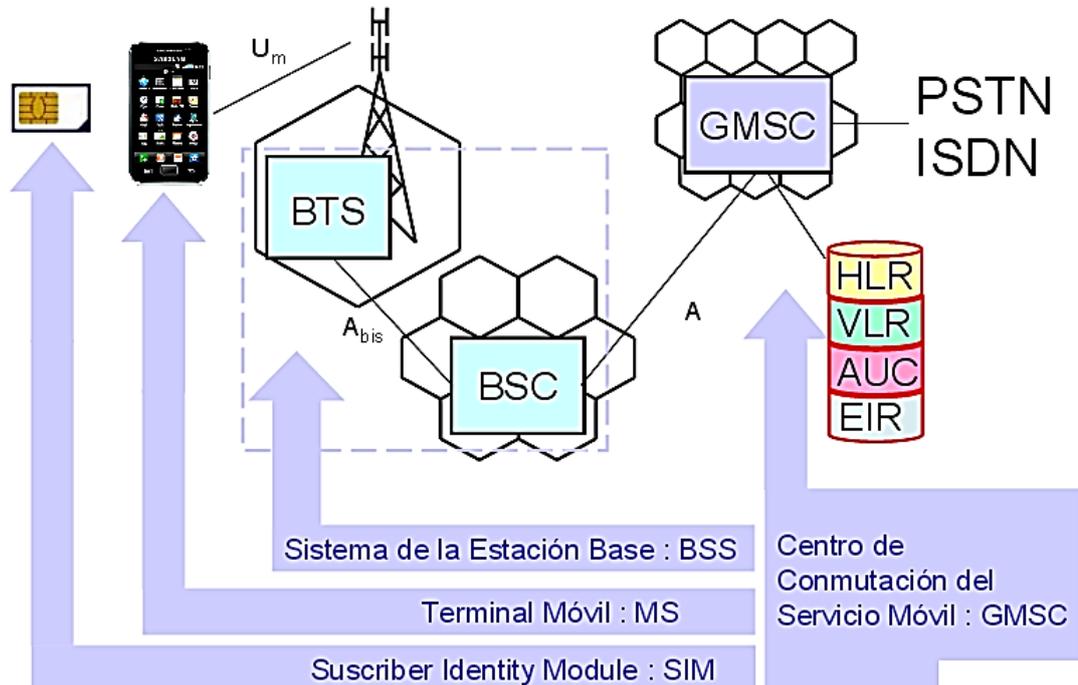


Figura 20: Descripción de elementos de la red GSM

Fuente: Todo sobre GSM Sistema global para las comunicaciones móviles.

Una estación móvil (MS) se compone funcionalmente de dos partes:

- El equipo terminal (ET): realiza funciones semejantes a la de una red RDSI y las siguientes funciones:
 - Transmisión radio
 - Gestión de canales de transmisión radio
 - Capacidad del terminal, incluyendo la interfaz hombre-máquina
 - Codificación de voz
 - Protección de errores
 - Control del flujo de datos de usuario
 - Adaptación de velocidad de datos de usuario y velocidad del canal
 - Soporte de terminales múltiples
 - Gestión de movilidad
- La terminación móvil (TM)
 - Hay tres tipos de TM
 - TMO Realiza las funciones anteriormente mencionadas, sin incluir ningún interfaz

-TM1 Incluye además una interfaz RDSI

-TM2 Incluye además interfaces CCITT series X y V

Utilizando estos tres tipos de TM se pueden establecer las configuraciones necesarias para acceder al sistema GSM.

Una estación móvil puede además clasificarse en distintos tipos según varias características:

-Por su utilización

-Equipo móvil

-Equipo portátil

-Equipo transportable

-Por la potencia de salida

-Clase 1 20 w - Móvil y transportable

-Clase 2 8 w - Vehículo y transportable

-Clase 3 5 w - Portátil

-Clase 4 2 w - Portátil

-Clase 5 0.8 W – Portátil

2.3.1.3. Módulo de identificación del usuario (SIM):



Figura 21: Tipos de SIM en la actualidad

Fuente: GSM, <http://html.rincondelvago.com/gsm.html>, 2014

Este módulo es el que contiene toda la información necesaria para realizar la función de autenticación del usuario, además de otras informaciones necesarias para el sistema.

El SIM debe contener la siguiente información:

-Número de serie

-Estado del SIM (bloqueado o desbloqueado)

-Clave del algoritmo de autenticación

- Algoritmo de Autenticación (A3)
- Identificación internacional del usuario móvil (MSI)
- Identificación temporal del usuario móvil (TMSI)
- Algoritmo de generación de claves de cifrado (A8)
- Clave del algoritmo de cifrado de señalización y datos (A5)
- Número de secuencia de la clave del algoritmo de cifrado
- Clase de control de acceso del usuario

Existen dos tipos distintos de módulo de identificación del usuario:

- Tarjeta inteligente o CHIP que puede ser retirada de la estación móvil cuando el usuario termina de utilizarla, es la más común.
- Módulo incorporado dentro de la estación móvil, con el fin de estar instalado permanentemente, aunque siempre sería posible retirarlo abriendo la carcasa de la estación móvil.

2.3.2. La evolución al GPRS

La Tecnología GPRS (General Packet Radio Service), es la evolución de la tecnología GSM (Global System for Mobile Communication), este estándar a diferencia del GSM permite la comunicación inalámbrica por paquetes. La estructura principal de esta tecnología es el SGSN (Serving GPRS Support Node), este nodo permite una conexión continua haciendo más rápido el acceso a la información.

El GPRS se caracteriza por utilizar la tecnología de "Multiplexación", es decir el canal de transferencia es dividido en subcanales llamados Slot o Time-slot, pudiendo ser dividido hasta en 8 canales. Cada Slot alcanzaría una velocidad aproximada de 13,4 KB/s, siendo utilizados en la actualidad 2 canales de bajada y 1 de subida, por lo que la velocidad máxima alcanzada sería de 26,8 KB/s para la bajada de datos. GPRS no es facturando por tiempo de conexión, como se realiza con GSM, sino por volumen de datos transferidos (enviados + recibidos). Puesto que con la tecnología GPRS se dispone de un acceso continuo, estando conectado permanentemente en el SGSN. Los canales de comunicación (time-slots) se comparten entre los distintos usuarios dinámicamente en función de sus necesidades y son asignados, únicamente, cuando se está transmitiendo datos. Así una vez que el paquete de datos ha sido transmitido a través de la interfaz aérea, los recursos radio pueden ser liberados para el uso por parte de otros usuarios.

2.3.2.1. Arquitectura de red GPRS

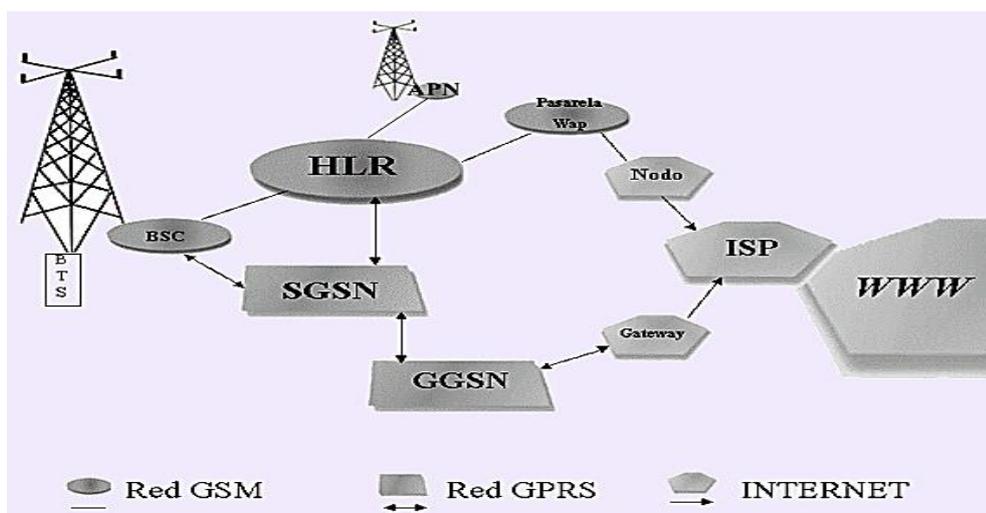


Figura 22: Disposición de arquitectura GPRS

Fuente: GPRS, <http://www.internetmania.net/int0/int67.htm>, 2014

La tecnología GPRS se pueden utilizar servicios como WAP(Wireless Application Protocol), SMS(servicio de mensajes cortos), MMS (servicio de mensajería multimedia), Internet y para los servicios de comunicación, como el correo electrónico y la Web (WWW). Para fijar una conexión de GPRS para un módem inalámbrico, un usuario debe especificar un APN, opcionalmente un nombre y contraseña de usuario, y muy raramente una dirección IP, todo proporcionado por el operador de red.

Descripción de la red:

- **BSS** (Base Station Subsystem) Base de red móvil compuesta por BTS y BSC.
- **SGSN** (Serving GPRS support Node) Nodo GPRS.
- **BTS** (Base Transceiver Station) Base transceptora (repetidor de ondas).
- **BSC** (Base Station Controller) Controlador de BTS.
- **HLR** (Home Location Register) Base de Datos de Usuarios (Comprueba los privilegios de los usuarios).
- **APN** (Access Point Name) Nombre de punto de acceso.
- **GGSN** (Gateway GPRS Support Node) Puerta de enlace o Pasarela de GPRS.
- **WAP** (Wireless Application Protocol) Protocolo de Transmisión Inalámbrica.
- **ISP** (Internet Service Provider) Proveedor de Servicios de Internet.

La red GPRS se integra dentro de la red GSM convencional; el primer paso es el **registro (Attach)** en la red por parte del terminal móvil. Luego en el **HLR**, se comprueba que tenga activado el servicio de **GPRS**, y dentro del HLR, en el **APN** se mide los servicios a los que tiene acceso el usuario de GPRS, entre estos servicios pueden ser Internet, Wap, SMS. Finalmente se da lugar a la activación del contexto del **protocolo de paquetes de datos (PDP)**.

Una vez activado este contexto, ya puede comenzar la comunicación basada en IP entre el usuario móvil y la red externa de datos de paquetes, ya sea a través de una conexión **WAP** o a través de una **puerta de enlace o Gateway** de conexión estándar a Internet.

2.3.3. TECNOLOGIA GPS

El GPS o sistema de posicionamiento global permite determinar en cualquier parte del mundo la posición de un objeto, persona vehículo, etc.

2.3.3.1. Funcionamiento y estructura

Este sistema de navegación está estructurado por 24 satélites en órbita alrededor del planeta tierra, a 20.200 km con trayectorias sincronizadas para cubrir toda la superficie de la Tierra, y estaciones en tierra firme.

Para que el módulo o dispositivo GPS determine una posición en la superficie terrestre, depende de que cada satélite en la constelación transmita su posición exacta; para ello cada satélite emite su propia efeméride⁷. El mismo trabaja con un software moderno que genera las posiciones de los **satélites**. También depende de una señal de tiempo extremadamente precisa a los recibidores en la tierra. Dada esta información, los receptores GPS pueden calcular su distancia al satélite, y combinando esta información (de cuatro satélites) el receptor puede calcular su posición exacta usando un proceso llamado trilateración.

La trilateración es un método matemático para determinar las posiciones relativas de objetos usando la geometría de triángulos de forma análoga la triangulación (método de trilateración inversa) usa las localizaciones conocidas de dos o más puntos de referencia, y la distancia medida entre el sujeto y cada punto de referencia para determinar de forma única y precisa la localización relativa de un punto en un plano bidimensional.

⁷ **Efeméride**, es una tabla de valores que da las posiciones de los objetos astronómicos en el cielo en un momento dado.

Cómo funciona el sistema de posicionamiento global

El sistema realiza una cuadrangulación respecto de la posición de cuatro satélites y calcula la ubicación del usuario.

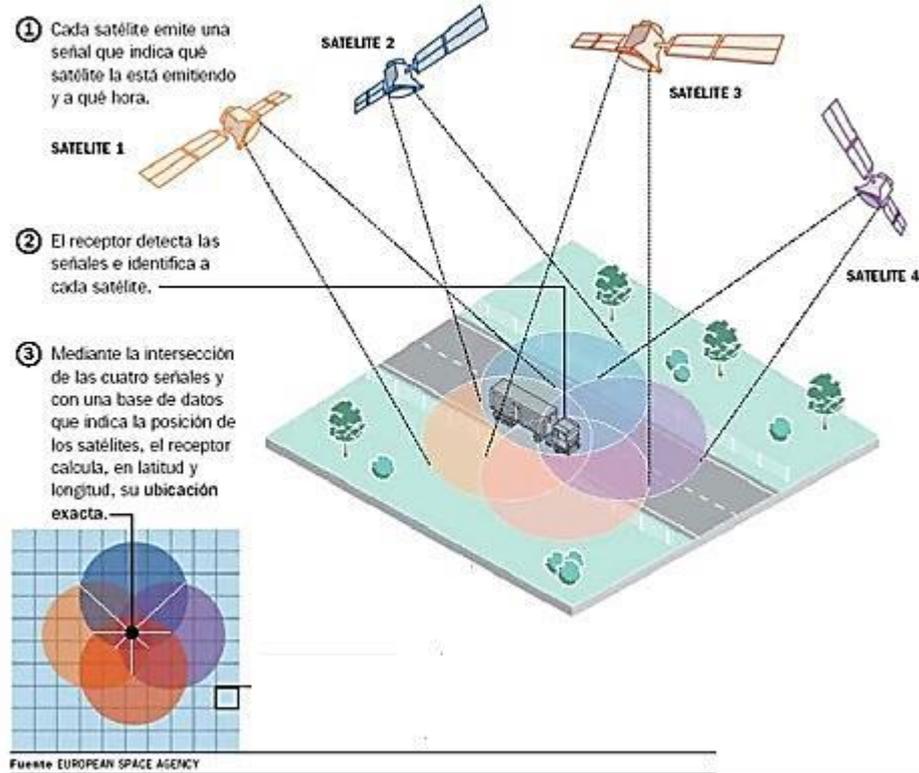


Figura 23: Funcionamiento del GPS

Fuente: GPS, <http://dasilvafernando.blogspot.com/2013/05/gps-o-spg.html>, 2014

Para determinar una posición exacta se necesitan cuatro satélites:

- Cada satélite indica que el receptor se encuentra en un punto en la superficie de la esfera, con centro en el propio satélite y de radio la distancia total hasta el receptor.
- Obteniendo información de dos satélites queda determinada una circunferencia que resulta cuando se intersecan las dos esferas en algún punto de la cual se encuentra el receptor.
- Teniendo información de un cuarto satélite, se elimina el inconveniente de la falta de sincronización entre los relojes de los receptores GPS y los relojes de los satélites. Y es en este momento cuando el receptor GPS puede determinar una posición 3D exacta (latitud, longitud y altitud).

2.3.4. Módulo GSM/GPRS/GPS V3.0-SKU: TEL0051

Este módulo GSM/GPRS consta de 4 bandas que funciona en las frecuencias EGSM 900MHz/ DCS 1800MHz y GSM 850MHz/PCS 1900 MHz, además de incluir la compatibilidad con GPS para navegación por satélite. Para el control y operación del módulo se utiliza comandos AT (GSM07.07, 07.05 y SIMCOM).

2.3.4.1. Especificaciones técnicas

- ✓ Fuente de alimentación: 6 – 12V
- ✓ Quad-Band 850/900/1800/1900 MHz
- ✓ Consumo de energía bajo (100 mA 7V – modo GSM)
- ✓ Incluye tecnología GPS
- ✓ GPRS clase 10 multi-slot
- ✓ Interruptor de control de USB/Arduino
- ✓ Superficie de tablero: Oro de la inmersión
- ✓ Dimensiones: 81 x 70mm
- ✓ 4x4 botón de teclado

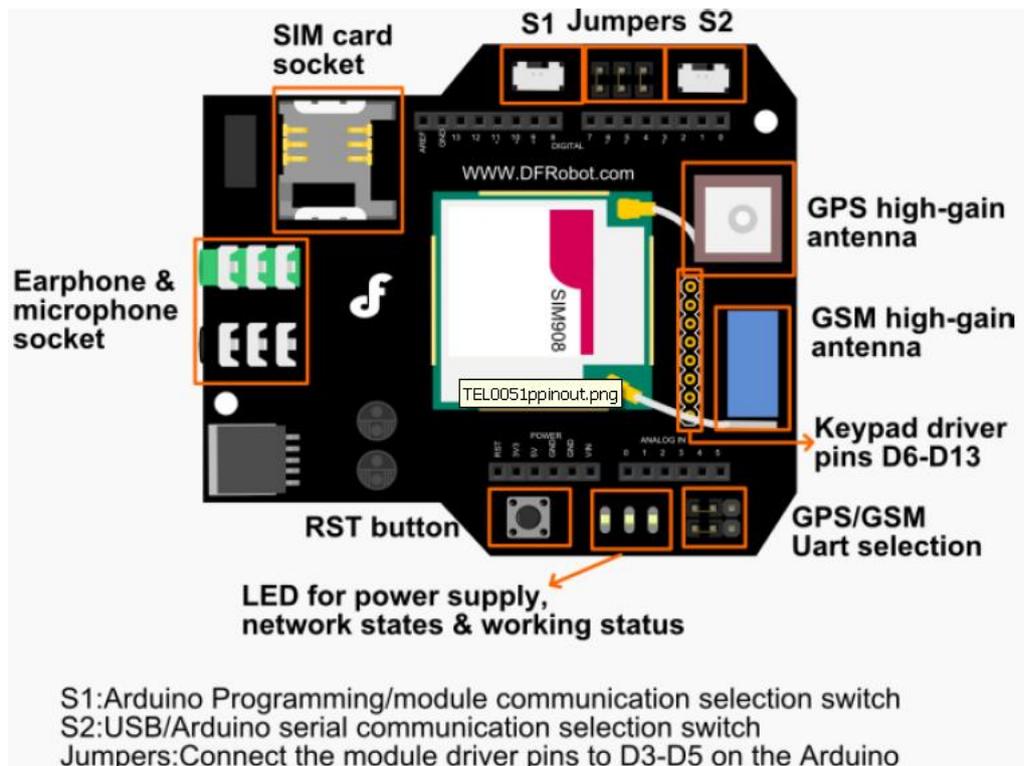


Figura 24: Estructura de Arduino GPS/GSM/GPRS

Fuente: Módulo V3.0, GPS / GPRS / GSM Módulo V3.0 (SKU: TEL0051).PDF, 2014

Interruptor S1: programa de carga PC a bordo de Arduino / comunica con GPS / GPRS / GSM Módulo (Arduino programación / módulo de comunicación) PC.

Interruptor S2: / GPRS Módulo GPS / GSM se conecta directamente con la PC a través del puerto USB o el módulo se comunica con la placa Arduino, que se comunica con el PC (USB / Arduino comunicación serie).

2.4. TECNOLOGIA X-BEE

Los **módulos Xbee** son dispositivos que integran un transmisor - receptor de ZigBee y un procesador en un mismo módulo, lo que le permite a los usuarios desarrollar aplicaciones de manera rápida y sencilla.

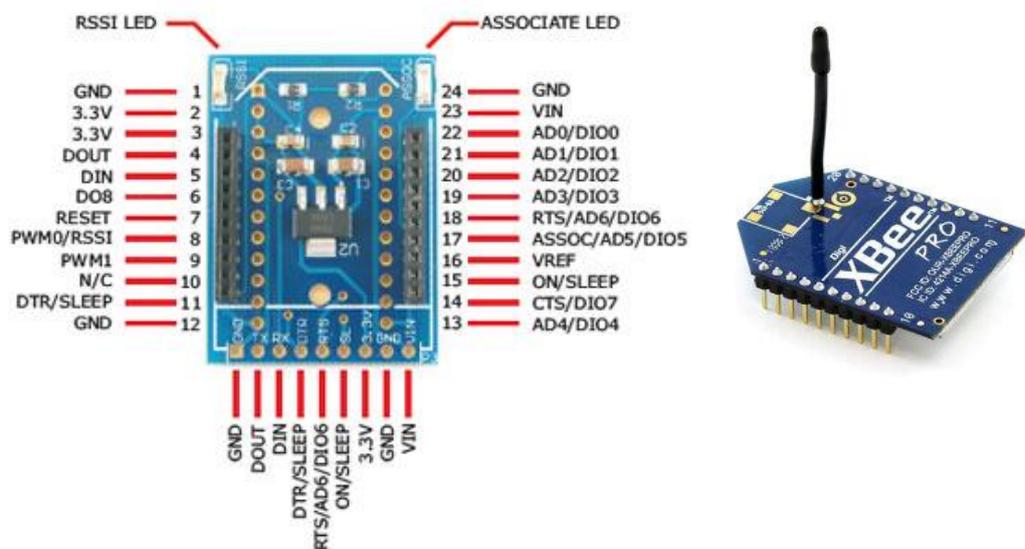


Figura 25: Diagrama esquemático del módulo Xbee

Fuente: Módulo Xbee, <http://www.droids.it/cmsvb4/content.php?143-990.001-XBee-Simple-Board>, 2013

Zigbee es un protocolo de comunicaciones inalámbrico basado en el estándar de comunicaciones para redes inalámbricas IEEE_802.15.4. Creado por Zigbee Alliance, una organización, teóricamente sin ánimo de lucro, de más de 200 grandes empresas (destacan Mitsubishi, Honeywell, Philips, Motorola, Invensys), muchas de ellas fabricantes de semiconductores.

Zigbee permite que dispositivos electrónicos de bajo consumo puedan realizar sus comunicaciones inalámbricas. Es especialmente útil para redes de sensores en entornos industriales, médicos y, sobre todo, **domóticos**.

“Las comunicaciones Zigbee se realizan en la banda libre de 2.4GHz. A diferencia de bluetooth no utiliza FHSS (Frequency hopping), sino que realiza las comunicaciones a través de una única frecuencia, es decir, de un canal. Normalmente puede escogerse un canal de entre 16 posibles. El alcance depende de la potencia de emisión del dispositivo así como el tipo de antenas utilizadas (cerámicas, dipolos,...) El alcance normal con antena dipolo en visión directa suele ser aproximadamente (tomando como ejemplo el caso de MaxStream, en la versión de 1mW de potencia) de 100m y en interiores de unos 30m. La velocidad de transmisión de datos de una red Zigbee es de hasta 256kbps. Por último decir que una red Zigbee la pueden formar, teóricamente, hasta 65535 equipos, es decir, el protocolo está preparado para poder controlar en la misma red esta cantidad enorme de dispositivos. La realidad es menor, siendo, de todas formas, de miles de equipos.”⁸

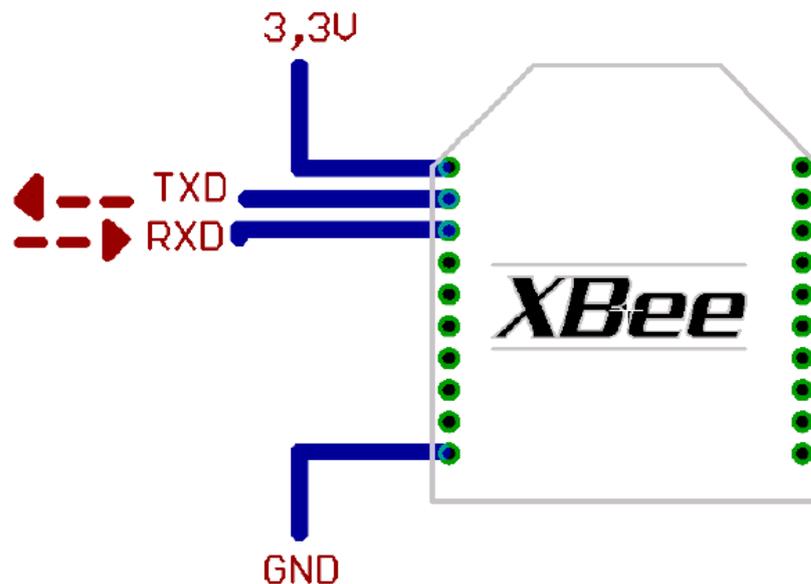


Figura 26: Circuito básico del Xbee
Fuente: Xbee Guía de usuario.pdf; 2008

⁸ **Módulo Xbee**, <http://www.droids.it/cmsvb4/content.php?143-990.001-XBee-Simple-Board>

El módulo Xbee requiere una alimentación desde 2.8 a 3.4 V, la conexión a tierra y las líneas de transmisión de datos por medio del UART (TXD y RXD) para comunicarse con un microcontrolador, o directamente a un puerto serial utilizando un conversor adecuado para los niveles de voltaje. Esta configuración, no permite el uso de Control de Flujo (RTS & CTS), por lo que ésta opción debe estar desactivada en el terminal y en el módulo XBEE.

Los módulos Xbee son económicos, poderosos y fáciles de utilizar. Algunas sus principales características son:

- Buen Alcance: hasta 300ft (100 mts) en línea vista para los módulos Xbee y hasta 1 milla (1.6 Km) para los módulos Xbee Pro.
- 9 entradas/salidas con entradas analógicas y digitales.
- Bajo consumo <50mA cuando están en funcionamiento y <10uA cuando están en modo sleep.
- Interfaz serial.
- 65,000 direcciones para cada uno de los 16 canales disponibles. Se pueden tener muchos de estos dispositivos en una misma red.
- Fáciles de integrar.

Existen 2 series de estos módulos. La serie 1 y la serie 2 o también conocida como 2.5. Los módulos de la Serie 1 y la Serie 2 tienen el mismo pin-out, sin embargo, NO son compatibles entre sí ya que utilizan distintos chipset y trabajan con protocolos diferentes.

La serie 1 está basada en el chipset Freescale y está pensado para ser utilizado en redes punto a punto y punto a multipunto. Los módulos de la serie 2 están basados en el chipset de Ember y están diseñados para ser utilizados en aplicaciones que requieren repetidores o una red mesh⁹. Ambos módulos pueden ser utilizados en los modos AT y API.

⁹ Red MESH: Es la topología más simple para configurar las redes de malla inalámbrica. Los sitios son uniformemente distribuidos y cada nodo puede ver todos los otros nodos.

2.4.1. Características técnicas:

- * ISM 2.4 GHz operating frequency
- * 1 mW (0 dBm) power output (up to 100m range)
- * RPSMA Connector, U.FL. Connector, Chip or Whip antenna options
- * Industrial temperature rating (-40° C to 85° C)
- * Approved for use in the United States, Canada, Australia and Europe
- * Advanced networking & low-power modes supported

Platform	XBee® 802.15.4 (Series 1)	XBee-PRO® 802.15.4 (Series 1)	XBee-PRO® XSC
Performance			
RF Data Rate	250 kbps	250 kbps	10 kbps / 9.6 kbps
Indoor/Urban Range	100 ft (30 m)	300 ft (100 m)	Up to 1200 ft (370 m)
Outdoor/RF Line-of-Sight Range	300 ft (100 m)	1 mi (1.6 km)	Up to 6 mi (9.6 km)
Transmit Power	1 mW (+0 dBm)	60 mW (+18 dBm)*	100 mW (+20 dBm)
Receiver Sensitivity (1% PER)	-92 dBm	-100 dBm	-106 dBm
Features			
Serial Data Interface	3.3V CMOS UART	3.3V CMOS UART	3.3V CMOS UART (5V Tolerant)
Configuration Method	API or AT Commands, local or over-the-air	API or AT Commands, local or over-the-air	AT Commands
Frequency Band	2.4 GHz	2.4 GHz	902 MHz to 928 MHz
Interference Immunity	DSSS (Direct Sequence Spread Spectrum)	DSSS (Direct Sequence Spread Spectrum)	FHSS (Frequency Hopping Spread Spectrum)
Serial Data Rate	1200 bps - 250 kbps	1200 bps - 250 kbps	1200 bps - 57.6 kbps
ADC Inputs	(6) 10-bit ADC inputs	(6) 10-bit ADC inputs	None
Digital I/O	8	8	None
Antenna Options	Chip, Wire Whip, U.FL, & RPSMA	Chip, Wire Whip, U.FL, & RPSMA	Wire Whip, U.FL, RPSMA
Networking & Security			
Encryption	128-bit AES	128-bit AES	No
Reliable Packet Delivery	Retries/Acknowledgments	Retries/Acknowledgments	Retries/Acknowledgements
IDs and Channels	PAN ID, 64-bit IEEE MAC, 16 Channels	PAN ID, 64-bit IEEE MAC, 12 Channels	PAN ID, 32-bit Address, 7 Channels
Power Requirements			
Supply Voltage	2.8 - 3.4VDC	2.8 - 3.4VDC	3.0 - 3.6VDC
Transmit Current	45 mA @ 3.3VDC	215 mA @ 3.3VDC	265 mA typical
Receive Current	50 mA @ 3.3VDC	55 mA @ 3.3VDC	65 mA typical
Power-Down Current	<10 uA @ 25° C	<10 uA @ 25° C	45 uA pin Sleep
Regulatory Approvals			
FCC (USA)	OUR-XBEE	OUR-XBEEPRO	MCO-XBEE XSC
IC (Canada)	4214A-XBEE	4214A-XBEEPRO	1846A-XBEE XSC
ETSI (Europe)	Yes	Yes* Max TX 10 mW	No
C-TICK Australia	Yes	Yes	No
Telec (Japan)	Yes	Yes*	No

Tabla 29: Especificaciones técnicas generales de los módulos Xbee
Fuente: Módulo Xbee, <http://www.digi.com/xbee>, 2014

2.4.2 Modos de operación

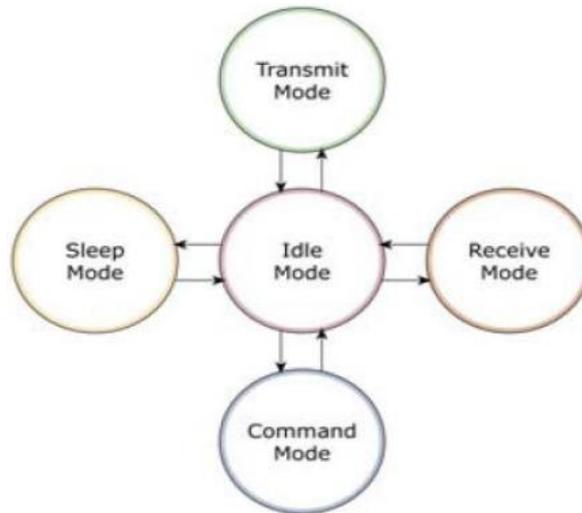


Figura 27: Modos de operación del módulo Xbee
Fuente: Xbee Guía de usuario.pdf; 2008

2.4.2.1. Modo recibir/transmitir

El módulo Xbee trabaja en este modo cuando le llega un paquete RF a través de la antena (modo Receive) o cuando se manda información serial al buffer del pin 3 (UART Data in) que luego será transmitida (modo Transmit).

Esta información puede ser directa o indirecta. En el modo directo la información se envía inmediatamente a la dirección de destino. En el modo Indirecto la información es retenida durante un período de tiempo y es enviada sólo cuando la dirección de destino la solicita. Es posible enviar información por dos modos:

Unicast, en el cual la comunicación es desde un punto a otro, y es el único modo que permite respuesta de quien recibe el paquete RF, es decir, quien recibe debe enviar un **ACK** (paquete llamado así, y que indica que recibió el paquete, el usuario no puede verlo, es interno de los módulos) a la dirección de origen. Quien envió el paquete, espera recibir un **ACK**, en caso de que no le llegue, reenviará el paquete hasta 3 veces o hasta que reciba el **ACK**.

Broadcast, en este, la comunicación es entre un nodo y a todos los nodos de la red, aquí no hay confirmación por **ACK**.

2.4.2.2. Sleep Mode – Modo de bajo consumo

Hace referencia a que se puede colocar al módulo RF en un modo de bajo consumo cuando no se está usando. Con la finalidad de colocar al módulo en modo de sueño, este debe cumplir las siguientes condiciones:

Sleep_RQ (pin 9) está en **alto** y el módulo está en **pin sleep mode** (SM= 1,2 o 5)

El módulo está en reposo (no hay transmisión ni recepción de datos) por la cantidad de tiempo definido por ST (Time before Sleep). [ST sólo está activado cuando SM=4,5]

La configuración de los ciclos de sueño se realiza principalmente con el comando SM. Por defecto, los modos de sueños están deshabilitados (SM=0), permaneciendo el módulo en estado de reposo/recepción. En este estado el módulo está siempre preparado para responder a un comando, ya sea, por el puerto serial o la interfaz RF, a continuación en la siguiente tabla se detalla los modos de operación, consumos, voltajes y condiciones para entrar en modo de bajo consumo:

Modo	Consumo alimentación			Modo Sleep	Modo Wake -up
	2.8 – 3 V	3.2 V	3.4V		
SM=1	<3 uA	32 uA	255 uA	Sleep_RQ	Sleep_RQ
SM=2	<35 uA	48 uA	170 uA	Sleep_RQ	Sleep_RQ
SM=3	(reservado)			(reservado)	(reservado)
SM=4	<34 uA	49 uA	240 uA	Comando ST	Comando SP
SM=5	<34 uA	49 uA	240 uA	Comando ST	Sleep RQ

Tabla 30: Tabla modo sleep y consumo de corriente

Fuente: Xbee Guía de usuario.pdf; 2008

2.4.2.3. Modo de comando

Como su nombre lo indica, modo de comando, permite programar comandos AT al módulo Xbee, para configurar, ajustar o modificar parámetros. Permite ajustar parámetros como la dirección propia o la de destino, así como su modo de operación entre otras cosas. Para poder ingresar los comandos AT es necesario utilizar el Hyperterminal de Windows, el programa X-CTU o algún microcontrolador que maneje UART y tenga los comandos guardados en memoria o los adquiera de alguna otra forma. Para ingresar a este modo se debe esperar un tiempo dado por el

comando GT (Guard Time, por defecto $ATGT=0x3E8^{10}$ que equivalen a 1000ms) luego ingresar +++ y luego esperar otro tiempo GT. Como respuesta el módulo entregará un **OK**. El módulo Xbee viene por defecto con una velocidad de 9600bps.

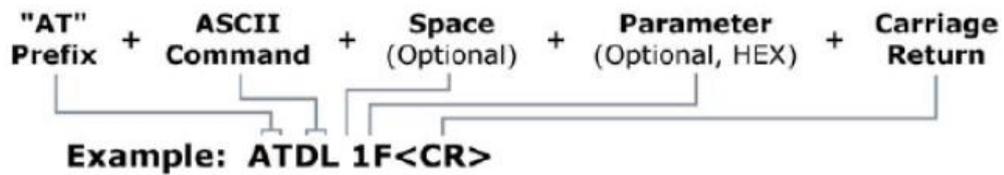


Figura 28: Estructura del comando AT
Fuente: Xbee Guía de usuario.pdf; 2008

La configuración del módulo puede ser por Hyperterminal o mediante el programa X-CTU que se describe más adelante. A continuación se describen algunos comandos básicos utilizados para la configuración el Xbee.

Si deseamos modificar la dirección del módulo de 16-bit se debe realizar con el comando **MY**, tal como el ejemplo siguiente:

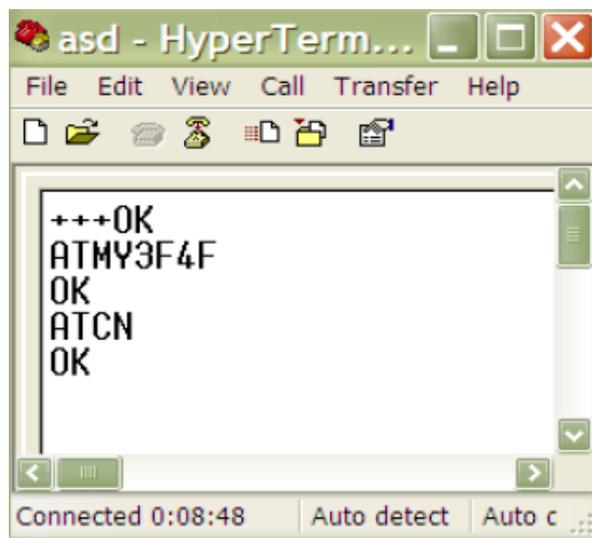


Figura 29: Escritura y uso de parámetro MY
Fuente: Xbee Guía de usuario.pdf; 2008

Tal como muestra la figura, en este ejemplo primero se ingresa al modo de comandos AT, recibiendo un **OK** de respuesta, luego se ingresa el comando **ATMY3F4F** y se presiona **ENTER** o caracter <CR><LF> (Carrier Return y Line Feed) si se maneja desde un microcontrolador. Con ello se recibe un <CR><LF>**OK**<CR><LF> como

¹⁰ Los valores de los parámetros están dado en formato hexadecimal

respuesta. Para salir del modo de Comandos se ingresa **ATCN** y se presiona **ENTER**. En caso de que no se ingrese ningún comando AT válido durante el tiempo determinado por CT (Command Mode Timeout), el módulo se saldrá automáticamente. Para que los cambios realizados tengan efecto se debe ingresar **ATCN** (sale del modo de comandos) o **ATAC** (aplica los cambios inmediatamente). Con el comando **ATWR**, se guardan los cambios en la memoria no volátil del módulo, pero sólo tendrán efecto una vez ingresado el comando AC o CN.

2.4.2.4. Modo transparente

Consiste en que todo lo que ingresa por el pin 3 (Data in), es guardado en el buffer de entrada y luego transmitido y todo lo que ingresa como paquete RF, es guardado en el buffer de salida y luego enviado por el pin 2 (Data out), este módulo viene por defecto en los módulos Xbee. Este tipo de comunicación es utilizado en punto a punto, ya que no necesita ningún tipo de control, otro de los uso es en el reemplazo de conexiones seriales por cable.

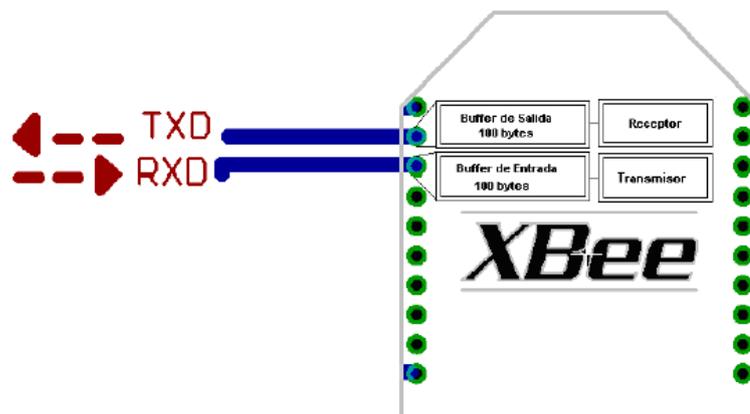


Figura 30: Configuración del módulo en modo transparente
Fuente: Xbee Guía de usuario.pdf; 2008

En este modo, la información es recibida por el pin 3 del módulo Xbee, y guardada en el buffer de entrada. Dependiendo de cómo se configure el comando RO, se puede transmitir la información apenas llegue un carácter (RO=0) o después de un tiempo dado sin recibir ningún carácter serial por el pin 3. En ese momento, se toma lo que se tenga en el buffer de entrada, se empaqueta, es decir, se integra a un paquete RF, y

se transmite. Otra condición que puede darse es cuando el buffer de entrada se llena, esto es, más de 100 bytes de información.

2.4.2.5. Modo de operación API

Es uno de los modos más complejo, pero permite el uso de frames¹¹ con cabeceras que aseguran la entrega de los datos, al estilo TCP. Extiende el nivel en el cual la aplicación del cliente, puede interactuar con las capacidades de red del módulo.

Cuando el módulo XBEE se encuentra en este modo, toda la información que entra y sale, es empaquetada en frames, que definen operaciones y eventos dentro del módulo.

Un Frame de Transmisión de Información (información recibida por el pin 3 o DIN) contiene:

- Frame de información RF transmitida.
- Frame de comandos (equivalente a comandos AT).

Un Frame de Recepción de Información contiene:

- Frame de información RF recibida.
- Comando de respuesta.
- Notificaciones de eventos como Reset, Associate, Disassociate, etc.

De esta manera se provee alternativas para la configuración del módulo y ruteo de la información en la capa de aplicación del cliente. Esto quiere decir que si un cliente envía información al módulo Xbee, estos datos serán contenidos en un frame cuya cabecera tendrá información útil referente el módulo. Además de que esta información se podrá configurar, esto es, en vez de estar usando el modo de comandos para modificar las direcciones, la API¹² lo realiza automáticamente. El módulo así enviará paquetes de datos contenidos en frames a otros módulos de

¹¹ Frame (o marco o trama): es un paquete de datos de longitud fija o variable, que ha sido codificado por un protocolo de comunicaciones en la capa de enlace de datos.

¹² API, Interfaz de Programación de Aplicaciones, es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas.

destino, con información a sus respectivas aplicaciones, conteniendo paquetes de estado, así como el origen, RSSI (potencia de la señal de recepción) e información de la carga útil de los paquetes recibidos.

2.4.2.5. Idle

Si el módulo Xbee no se encuentra en ningún modo de los citados anteriormente, pues no está ni transmitiendo ni recibiendo, ni ahorrando energía ni en el modo de comandos, entonces se dice que se encuentra en un estado al que se le llama IDLE.

2.4.3. Software de programación:

No es necesaria la configuración de comunicaciones de RF. El módulo Xbee posee configuración por defecto que es compatible con una amplia gama de aplicaciones.

Configuraciones avanzadas se pueden aplicar usando comandos simples AT. Dentro de las características avanzadas se encuentran la configuración del conversor integrado A/D de 6 canales con 10 bits de resolución o entradas y salidas digitales.

Los módulos Xbee pueden ser programados a través de una hyperterminal y una interface serial con un MAX3232 y una serie de comandos llamados AT, el fabricante de los módulos también facilita un software de programación llamado **X-CTU**.

Existen dos tipos de interfaces, serial y USB que pueden ser utilizadas para programar los módulos Xbee con el software propietario; con este software podemos definir de una forma rápida todos los parámetros que queramos modificar de los módulos.

2.4.4. Pasos de instalación del software X-CTU:

Se procede a reconocer el icono para la instalación como en la figura 35 el archivo ejecutable X-CTU ver. 5.2.7.5, a continuación nos mostrará una ventana de bienvenida a la cual se da clic en *siguiente*.

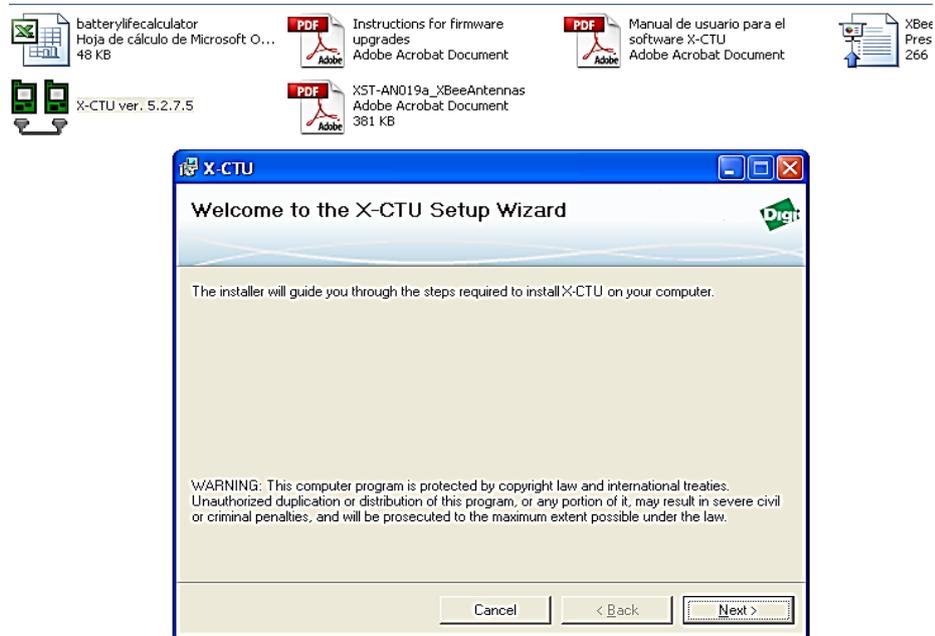


Figura 31: Pasos de instalación X-CTU, ventana de bienvenida
Fuente: Autores, 2014

De preferencia la instalación siempre en la partición C:\ y damos clic en *siguiente*.

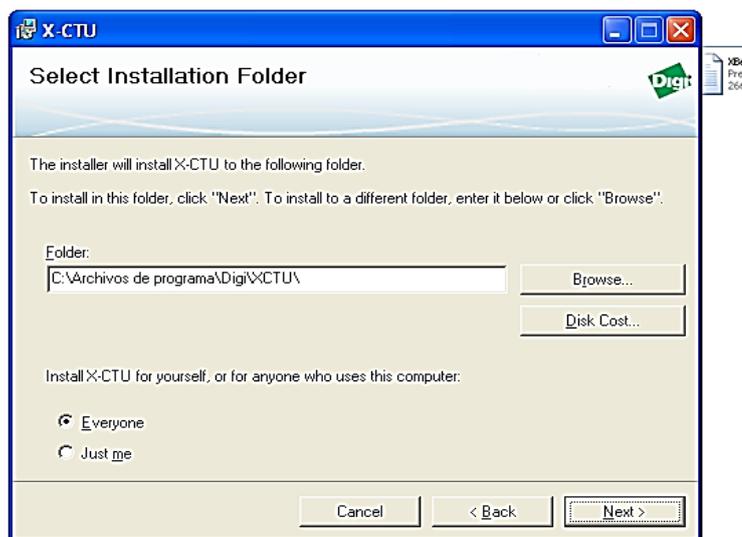


Figura 32: Pasos de instalación X-CTU, selección de la partición.
Fuente: Autores, 2014

Automáticamente el proceso de instalación continúa, puede tomar un par de minutos, depende de las características del PC.

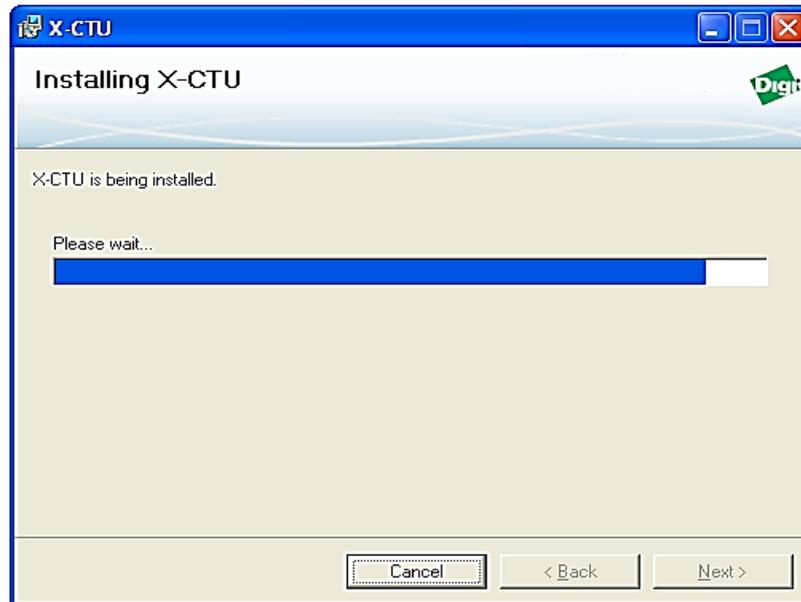


Figura 33: Pasos de instalación X-CTU, proceso de instalación.
Fuente: Autores, 2014

Finalmente se plasmará la ventana de finalización de la instalación tal como en la figura 38.

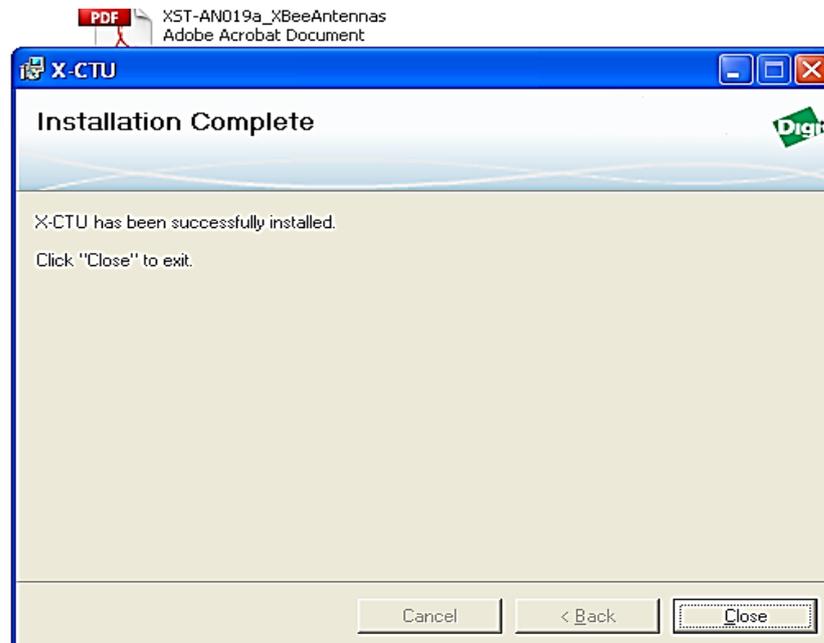


Figura 34: Pasos de instalación X-CTU, ventana final de la instalación.
Fuente: Autores, 2014

Se busca la aplicación como Digi → X-CTU

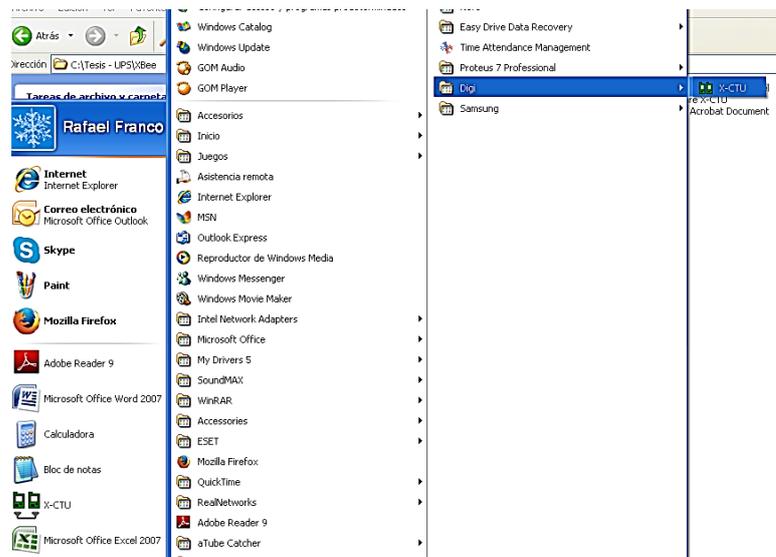


Figura 35: Búsqueda del programa instalado en el PC.
Fuente: Autores, 2014

Al abrir el programa se obtendrá la ventana siguiente donde habrá las pestañas:
 PC Settings, Range Test, Terminal, Modem Configuration

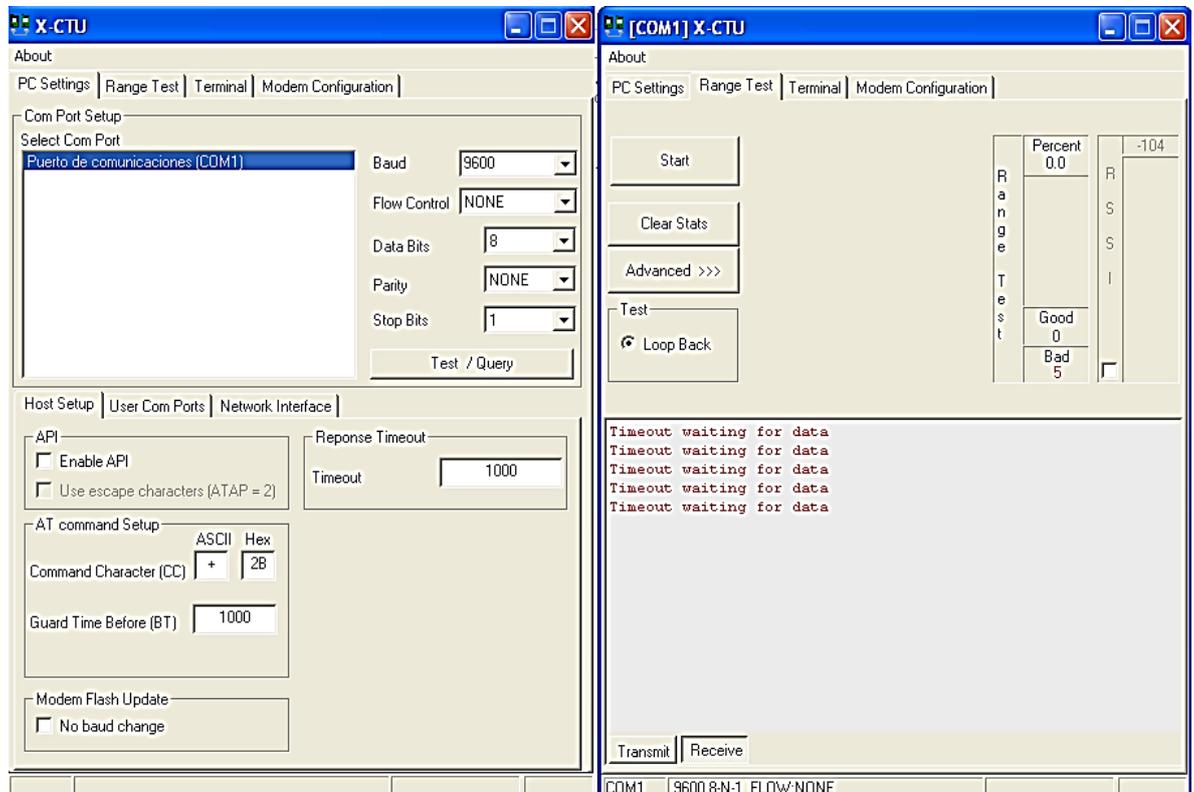


Figura 36: Ventana principal Del software X-CTU: PC Settings, Range Test
Fuente: Autores, 2014

PC Settings: Permite seleccionar el puerto COM deseado y configurar ese puerto para adaptarse a la configuración de las radios.

Range Test: Permite realizar una prueba de rango entre dos radios.

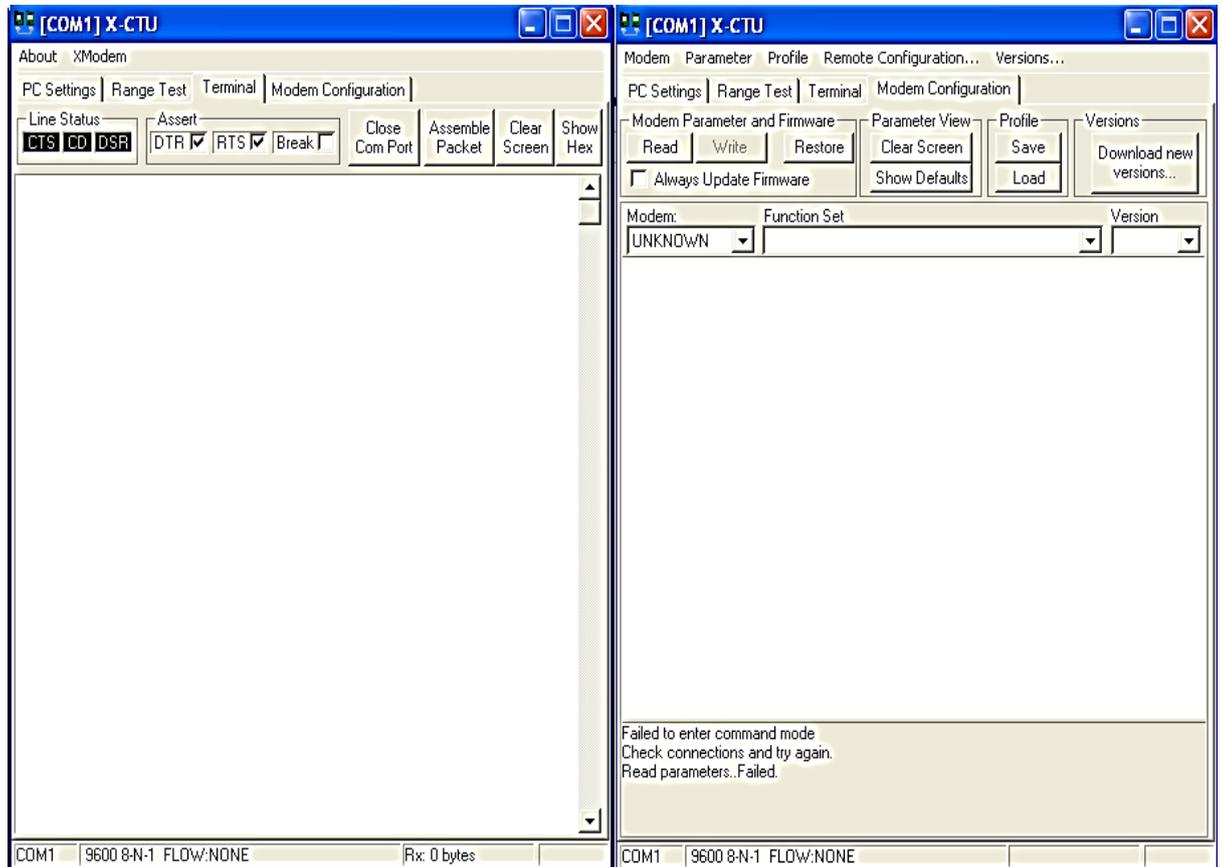


Figura 37: Ventana principal del software X-CTU: Terminal, Modem Configuration
Fuente: Autores, 2014

Terminal: Permite el acceso al puerto COM con un programa de emulación de terminal. Esta opción también permite la posibilidad de acceder al firmware utilizando los comandos AT.

Modem Configuration: Permite la capacidad de programar la configuración del firmware de las radios a través de una interfaz gráfica de usuario. Esta opción también permite cambiar las versiones de firmware.

2.5 DATA LOGGER

Este dispositivo permite registrar datos USB con la finalidad de conectar dispositivos de almacenamiento masivo, en este caso de proyecto de Tesis para una de las prácticas con el microcontrolador. Por lo general se necesita de una Memory Stick (MMC) para poder llevar los archivos a una computadora/laptop. El módulo de registro de datos maneja el sistema de archivos mediante el vínculo IC/Firmware,



todas estas tareas se realizan mediante simples comandos seriales. La utilidad de este dispositivo es muy variada, puede partir desde una simple aplicación educativa así como complicadas aplicaciones para grandes proyectos donde se requiere de una electrónica inteligente.

Figura 38: Estructura física de la memory stick datalogger

Fuente: Parallax, <http://www.parallax.com/product/27937>, Año 2014

2.5.1 Características.

Entre las principales características del Datalogger tenemos:

- Fiabilidad, estabilidad del sistema mediante chip W5100 hardware.
- Conexión directa con dispositivos serie (función de red rápida y sencilla, personalización de Firmware).
- Soporta la conexión PPPoE
- Configuración de serie(comando simple y fácil)
- Se permite contraseña de seguridad
- Configuración de herramientas.
- Permite interfaz Ethernet 10/100 – máximo 230 Kbps(Interfaz serie)
- Trabaja bajo los parámetros de la directiva RoHS ¹³

¹³ Directiva RoHS (Restriction of use of Hazardous substances – Restricción en el uso de Sustancias Peligrosas) regula el uso de plomo y otros componentes potencialmente peligrosos en aparatos eléctricos y electrónicos con la finalidad de proporcionar un nivel de protección para la salud humana y el medio ambiente.

2.5.2. Aplicaciones.

Aplicaciones en procesos:

- Empleado para monitorear las condiciones ambientales de temperatura y humedad relativa de las áreas de producción. Ej. Áreas de producción de comprimidos efervescentes donde la humedad resulta crítica.
- Monitoreo y registro digital de temperaturas de esterilización en autoclaves de calor seco y húmedo.
- Registro de encendido y apagado de equipos y bombas
- Evaluación de temperaturas alcanzadas por el producto en los hornos de termocontraibles

Aplicaciones en logística de distribución

- Control de cadena de frío local y de exportación.
- Monitoreo de las condiciones ambientales durante la cadena de distribución pero también de la temperatura real del producto.
- Determinación de puntos críticos: transporte, almacenamientos temporarios, destino, etc.

Aplicaciones en laboratorios de control de calidad y desarrollo

Los Data Loggers pueden ser muy fácilmente utilizados para el monitoreo de:

- Estufas de cultivo
- Cámaras de Estabilidad
- Heladeras y cámaras de frío utilizadas para conservación de standards químicos y/o cepas microbiológicas de referencia.

2.5.3. Operación y set de comandos.

El corazón del datalogger es el USB Host Controller IC que en sus siglas en inglés corresponde al “Circuito integrado controlador Host USB”, así es posible controlar por el usuario con ayuda de comandos en serie, siempre y cuando los archivos o datos estén en formato FAT¹⁴, además esto permite la fácil lectura o escritura en un computador. La finalidad de estos archivos es importarlos como hojas de cálculo para que el usuario final pueda acceder a los datos sin problemas y realizar las operaciones que sean pertinentes según su la aplicación o uso.

La manera de controlar el Módulo del Datalogger es a través de un dispositivo controlador USB llamado Vinculum VNC1L que pertenece a la familia FTDI Vinculum encapsulado a un dispositivo controlador, el cual nos permite implementar un sin número de aplicaciones relacionadas con dispositivos USB que hacen la función de host.

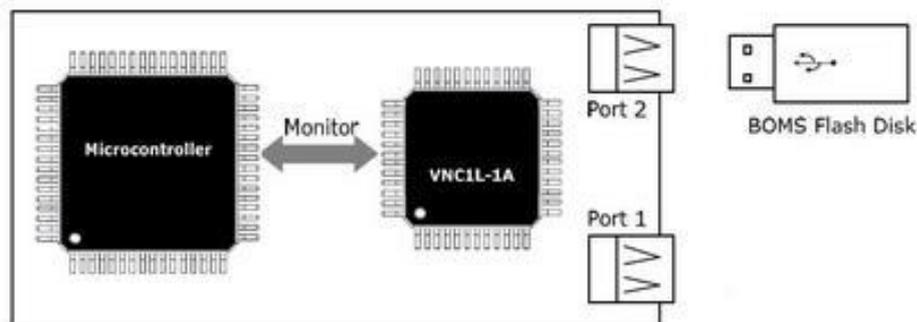


Figura 39: Diagrama VNC1L / Microcontrolador.

Fuente: Backtotheflux, <http://backtotheflux.com/articulos/host-usb-vinculum-vnc1l>, 2014

¹⁴ FAT es el acrónimo de "File Allocation Table" - tabla de localización de archivos, en inglés. Y es una especie de índice, que tu sistema operativo utiliza para guardar la localización real de cada archivo individual.

Descripción de comandos y respuestas del módulo que se usan comúnmente para controlar el Datalogger:

Extendend ASCII Command for Terminal Mode	Shortened Hexadecimal Command for microprocessor mode	Función	Respuesta
Switching between Shortened and Extended Command sets			
'SCS' <cr>	\$10,\$0D	Switches to the shortened command set	This will return the prompt '>',\$0D to indicate that the device is in shortened command set mode
'ECS' <cr>	\$11,\$0D	Switches to the extended command set	This will return the prompt 'D: \>',\$0D to indicate that the device is in extended command set mode
Responses to indicate if disk is online			
<cr>	\$0D	Check if online	This will return the appropriate prompt or 'no disk' message for the current command set
Response to Check if online for Extended Command Mode		If no valid disk is found	No Disk', \$0D
		If a valid disk is found	'D: \>',\$0D
Response to Check if online for Short Command Mode		If no valid disk is found	'ND', \$0D
		If a valid disk is found	'>',\$0D
Directory operations			
'DIR' <cr>	\$01,\$0D	Lists the current directory	A list of the names and directory names are returned. Each entry is terminated by \$0D. A directory entry has <sp>'DIR' after the name and before the \$0D.
'DIR' <sp><name><cr>	\$01,\$20,<name>,\$0D	Lists the name followed by the size. Use this before doing a the read to know how many bytes to expect.	\$0D,<name><sp><size in hex (4 bytes) LSB frst> \$0D
'DLD' <sp><name><cr>	\$05,\$20,<name>,\$0D	Delete directory	Deletes the directory <name> from the current directory <prompt> \$0D

'MKD' <sp><name><cr>	\$07,\$20,<name>,\$0D	Make directory	Creates a new directory <name> in the current directory <prompt>\$0D
'CD' <sp><name><cr>	\$02,\$20,<name>,\$0D	The current directory is changed to the new directory <name>	<prompt>\$0D
'CD' <sp>'..'<cr>	\$02,\$20,\$2E,\$2E,\$0D	Move up one directory level	<prompt>\$0D
File operations			
'RD' <sp><name><cr>	\$04,\$20,<name>,\$0D	Read the <name>	This will send back the entire the in binary to the monitor. The size should first be found by using the 'RD' <sp><name><cr> command so that the expected number of bytes is known. <prompt>\$0D
'RDF' <sp><size in hex (4 bytes)><cr>	\$0B,\$20, size in hex (4 bytes),\$0D	Read the data of <size in hex (4 bytes)> from the current open file.	This will send back the requested amount of data to the monitor. <prompt>\$0D
'DLF' <sp><name><cr>	\$07,\$20,<name>,\$0D	Delete file <name>	This will delete the file from the current directory and free up the FAT sectors. <prompt>\$0D
'WRF' <sp><size in hex (4 bytes)><cr><data bytes of size><cr>	\$08,\$20, size in hex (4 bytes), \$0D, \$data, \$0D	Writes the data of <size in hex (4 bytes)> to the end of the current open file.	<prompt>\$0D
'OPW' <sp><name><cr>	\$09,\$20,<name>,\$0D	Open a file for writing to with 'WRF'	<prompt>\$0D
'OPR' <sp><name><cr>	\$0E,\$20,<name>,\$0D	Open a file for reading to with 'RDF'	<prompt>\$0D
'CLF' <sp><name><cr>	\$0A,\$20,<name>,\$0D	Closes a file for writing	<prompt>\$0D
'REN' <sp><orig name><sp><new name><cr>	\$0C,\$20,<orig name>,\$20,<new name><cr>	Rename a file or directory	<prompt>\$0D
'FS'<cr>	\$12,\$0D	Returns free space in bytes on disk	<free space in hex (4 bytes) LSB first>\$0D
Commands for UART monitor mode only			
'\$BD' <sp><divisor (3 bytes)LSB first><cr>	\$14,\$20, divisor (3bytes) LSB first>,\$0D	Set Baud Rate (See Baud Rate Table)	<prompt>\$0D
Power Management Commands			
'\$UD'<cr>	\$15,\$0D	Suspend the disk when not in use to conserve power. The disk will be woken up automatically the	<prompt>\$0D

		next time a disk command is sent to it.	
'WKD'<cr>	\$16,\$0D	Wake Disk and do not put it into suspend when not in use.	<prompt>\$0D
'SUM'<cr>	\$17,\$0D	Suspend Monitor and stop clocks	<prompt>\$0D
Debug commands			
'SD'<sp><sector number in ASCII hex><cr>	\$0,\$20,... \$0D	Sector dump. This is used for debug purposes and may be removed. e.g. 'SD 0000 <cr>' will dump sector 0000.'SD 0010'<cr> will dump sector 16 decimal.	Send back 512 bytes from the sector specified in HEX converted to ASCII. Every 16 bytes is followed by a \$0D. <prompt>\$0D
'IDD'<cr>	\$0F,\$0D	Identify disk drive. This will display information about the attached disk.	Send IDD data block and then <prompt>\$0D
'FWV'<cr>	\$01,\$0D	Get firmware versions	Display the version number of the main firmware and the reprogramming firmware in the VNCL1 'MAIN x.xx'\$0D 'RPRG x.xx'\$0D Then <prompt>\$0D

Tabla 31: Comandos para controlar Memory Dataloger

Fuente: Parallax,

<http://www.parallax.com/Portals/0/Downloads/docs/prod/comm/VDAPFirmwareSpec.pdf>, 2014

Tabla de errores básica obtenida cuando se manipula el módulo memory stick Datalogger:

Error	Command Mode	Result
If command is unrecognised	Extended Command set	'Bad Command ', \$0D
	Shortened Command set	'BC ', \$0D
If command fails	Extended Command set	'Command Failed ', \$0D
	Shortened Command set	'CF ', \$0D

Tabla 32. Errores, comandos, resultados

Fuente: Parallax,

<http://www.parallax.com/Portals/0/Downloads/docs/prod/comm/VDAPFirmwareSpec.pdf>, 2014

2.5.4. Diagrama de pines, modo UART, modo SPI

El dispositivo Memory Stick Datalogger puede manejarse siguiendo los siguientes protocolos:

UART: Transmisor – Receptor Asíncrono Universal

SPI: Interfaz serial periférica.

El modo UART es el que se utilizará en el desarrollo de la tesis de ingeniería ya que la gama de PIC cuenta con este módulo incorporado.

Pin	Nombre	Descripción
1	VSS	Conectar el sistema a negativo (tierra).
2	RTS#	Respuesta de envío (Conectar a MCU CTS).
3	VDD	Conectar a +5V.
4	RXD	Recepción de datos (Conectar a MCU TXD).
5	TXD	Transmisión de datos (Conectar a MCU RXD).
6	CTS#	Clear to Send (Conectar a MCU RTS).
7	NC	No Conectado.
8	RI#	Indicador Ring (Making this input low resumes from suspends).

Tabla 33. Diagrama de pines Modo UART

Fuente: www.parallax.com/Portals/Downloads/prod/comm/MemoryStickDataloggerV1.pdf, 2014

Como referencia el jumper se debe colocar del lado derecho, además de conocer los datos técnicos de voltajes que se necesita tal como se describe en el siguiente punto.

2.5.5. Datos técnicos e indicadores del Datalogger

El Datalogger posee un LED que permite conocer los diferentes estados del módulo en los procesos de envío de datos o respuestas al mismo:

Observación	Respuesta
Power ON	LED Green y LED Red flash alternadamente por 2 segundos. Esto se repite hasta que se actualice.
Inicialización del disco USB	LED Green ON y LED Red OFF.
Lectura del disco USB	LED Green OFF y LED Red ON.
Remover el disco USB	LED Green OFF y LED Red OFF.
Lectura del disco USB	LED Green OFF y LED Red ON.
Envío de comandos desde software hasta el Puerto del disco USB	LED Green OFF y LED Red Flash.
Envío de comandos desde software hasta el Puerto del disco USB removido	LED Green OFF y LED Red OFF.

Tabla 34. Estados del LED indicador en el Datalogger

Fuente: www.parallax.com/Portals/Downloads/prod/comm/MemoryStickDataloggerV1.pdf, 2014

Con el fin de evitar daños del Datalogger se debe conocer las especificaciones técnicas del mismo tal como se detalla en la siguiente tabla:

Símbolo	Significado	Mínimo	Típico	Máximo	Unidad
V _{DD}	Voltaje de poder	4.75	5.0	5.25	V
-	Temperatura	- 65°	-	- 65°	C
-	Temperatura de operación	0°	-	- 70°	C
I _{VDD}	Corriente de la fuente	-	25	-	mA
I _{VDD}	Corriente en standby	1	2	2	mA

Tabla 35. Especificaciones técnicas del Datalogger

Fuente: www.parallax.com/Portals/Downloads/prod/comm/MemoryStickDataloggerV1.pdf, 2014

2.6 ETHERNET

Ethernet, estándar de redes de área local para computadoras con acceso al medio por contienda CSMA/CD (Acceso múltiple por Detención de Potadora con Detención de colisiones), es una técnica usada en redes, define las características de cableado y señalización de nivel físico y los formatos de tramas de datos del nivel de enlace de datos del modelo OSI.

2.6.1. Tecnología y velocidad de Ethernet

“Hace ya mucho tiempo que Ethernet consiguió situarse como el principal protocolo del nivel de enlace. Ethernet 10Base2 consiguió, ya en la década de los 90s, una gran aceptación en el sector. Hoy por hoy, 10Base2 se considera como una “tecnología de legado” respecto a 100BaseT. Hoy los fabricantes ya han desarrollado adaptadores capaces de trabajar tanto con la tecnología 10BaseT como la tecnología 100BaseT y esto ayuda a una mejor adaptación y transición.”¹⁵

La tecnología Ethernet que existe se diferencia en estos conceptos:

Velocidad de transmisión: -Velocidad a la que transmite la tecnología.

Tipo de cable: -Tecnología de nivel físico que usa la tecnología.

Longitud máxima: -Distancia máxima que puede haber entre dos nodos adyacentes (sin estaciones repetidoras)

Topología: -Determina la forma física de la red. Bus si se usan conectores T (hoy solo usados con las tecnologías más antiguas) y estrella si se usan hubs (estrella de difusión) o switches (estrella conmutada)

¹⁵ Ethernet, <http://es.wikipedia.org/wiki/Ethernet>, 17-02-2014

2.6.2 Versiones

En la siguiente tabla se describe las versiones 802.3:

Estándar Ethernet	Fecha	Descripción
Ethernet experimental	1972 (patentado en 1978)	2,85 Mbit/s sobre cable coaxial en topología de bus.
Ethernet II (DIX v2.0)	1982	10 Mbit/s sobre coaxial fino (thinnet) - La trama tiene un campo de tipo de paquete. El protocolo IP usa este formato de trama sobre cualquier medio.
IEEE 802.3	1983	10BASE5 10 Mbit/s sobre coaxial grueso (thicknet). Longitud máxima del segmento 500 metros - Igual que DIX salvo que el campo de Tipo se substituye por la longitud.
802.3a	1985	10BASE2 10 Mbit/s sobre coaxial fino (thinnet o cheapernet). Longitud máxima del segmento 185 metros
802.3b	1985	10BROAD36
802.3c	1985	Especificación de repetidores de 10 Mbit/s
802.3d	1987	FOIRL (Fiber-Optic Inter-Repeater Link) enlace de fibra óptica entre repetidores.
802.3e	1987	1BASE5 o StarLAN
802.3i	1990	10BASE-T 10 Mbit/s sobre par trenzado no blindado (UTP). Longitud máxima del segmento 150 metros.
802.3j	1993	10BASE-F 10 Mbit/s sobre fibra óptica. Longitud máxima del segmento 1000 metros.
802.3u	1995	100BASE-TX , 100BASE-T4 , 100BASE-FX Fast Ethernet a 100 Mbit/s con auto-negociación de velocidad.
802.3x	1997	Full Duplex (Transmisión y recepción simultáneos) y control de flujo.

802.3y	1998	100BASE-T2 100 Mbit/s sobre par trenzado no blindado(UTP). Longitud máxima del segmento 100 metros
802.3z	1998	1000BASE-X Ethernet de 1 Gbit/s sobre fibra óptica.
802.3ab	1999	1000BASE-T Ethernet de 1 Gbit/s sobre par trenzado no blindado
802.3ac	1998	Extensión de la trama máxima a 1522 bytes (para permitir las "Q-tag") Las Q-tag incluyen información para 802.1Q VLAN y manejan prioridades según el estandar 802.1p.
802.3ad	2000	Agregación de enlaces paralelos.
802.3ae	2003	Ethernet a 10 Gbit/s ; 10GBASE-SR, 10GBASE-LR
IEEE 802.3af	2003	Alimentación sobre Ethernet (PoE).
802.3ah	2004	Ethernet en la última milla.
802.3ak	2004	10GBASE-CX4 Ethernet a 10 Gbit/s sobre cable bi-axial.
802.3an	2006	10GBASE-T Ethernet a 10 Gbit/s sobre par trenzado no blindado (UTP)
802.3ap	en proceso (draft)	Ethernet de 1 y 10 Gbit/s sobre circuito impreso .
802.3aq	en proceso (draft)	10GBASE-LRM Ethernet a 10 Gbit/s sobre fibra óptica multimodo.
802.3ar	en proceso (draft)	Gestión de Congestión
802.3as	en proceso (draft)	Extensión de la trama

Tabla 36: Versiones Ethernet 802.3

Fuente: Ethernet, <http://es.wikipedia.org/wiki/Ethernet>, 2014

2.6.3 Formato de la trama Ethernet

En Ethernet la trama tiene el nombre de frame. A continuación se describe la estructura de Payload dada en Ethernet en la cual se hace uso de TCP e IP.

El primer campo es el preámbulo que indica el inicio de la trama y tienen el objeto de que el dispositivo que lo recibe detecte una nueva trama y se sincronice.

El delimitador de inicio de trama indica que el frame empieza a partir de él.

Los campos de MAC (o dirección) de destino y origen indican las direcciones físicas del dispositivo al que van dirigidos los datos y del dispositivo origen de los datos, respectivamente. La etiqueta es un campo opcional que indica la pertenencia a una VLAN o prioridad en IEEE P802.1p

Ethernettype indica con que protocolo están encapsulados los datos que contiene la Payload, en caso de que se usase un protocolo de capa superior.

La Payload es donde van todos los datos y, en el caso correspondiente, cabeceras de otros protocolos de capas superiores (Según Modelo OSI, véase Protocolos en informática) que pudieran formatear a los datos que se tramiten (IP, TCP, etc.). Tiene un mínimo de 46 Bytes (o 42 si es la versión 802.1Q) hasta un máximo de 1500 Bytes. La secuencia de comprobación es un campo de 4 bytes que contiene un valor de verificación CRC (Control de redundancia cíclica). El emisor calcula el CRC de toda la trama, desde el campo destino al campo CRC suponiendo que vale 0. El receptor lo recalcula, si el valor calculado es 0 la trama es válida. El gap de final de trama son 12 bytes vacíos con el objetivo de espaciado entre tramas.

Estructura de la trama de 802.3 Ethernet								
Preámbulo	Delimitador de inicio de trama	MAC de destino	MAC de origen	802.1Q Etiqueta (opcional)	Ethertype (Ethernet II) o longitud (IEEE 802.3)	Payload	Secuencia de comprobación (32-bit CRC)	Gap entre frames
7 Bytes	1 Byte	6 Byte	6 Bytes	(4 Bytes)	2 Bytes	De 46 (o 42) hasta 1500 Bytes	4 Bytes	12 Bytes
					64–1522 Bytes			
					72–1530 Bytes			
					84–1542 Bytes			

Tabla 37: Estructura de la trama Ethernet 802.3

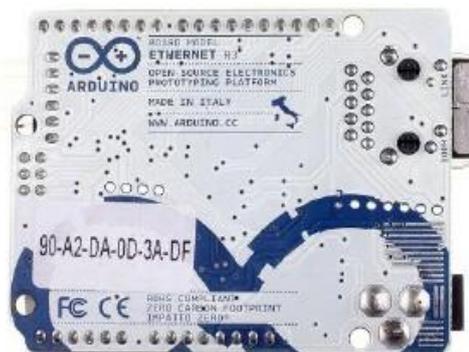
Fuente: Ethernet, <http://es.wikipedia.org/wiki/Ethernet>, 2014

2.6.4 Módulo Arduino Ethernet

Este módulo electrónico está compuesto e un microprocesador Atmega328 (ficha técnica en anexos). Posee 14 pines digitales de entrada / salida, 6 entradas analógicas, un 16 MHz oscilador de cristal, un RJ45 de conexión, un conector de alimentación, una cabecera ICSP, y botón de reinicio.



Arduino Ethernet Rev. vista frontal 3 bordo



Arduino Ethernet Rev. visión trasera 3 bordo

Figura 40: Módulo Arduino Ethernet
Fuente: Arduino Ethernet.pdf, 2014

2.6.4.1 Características del módulo:

Descripción	Observación
Microcontroladores	ATmega328
Tensión de funcionamiento	5V
Plug Voltaje de entrada (recomendado)	7-12V
Plug Voltaje de entrada (límites)	6-20V
Voltaje de entrada PoE (límites)	36-57V
Digital I / O Pins	14 (de los cuales 4 para salidas PWM)
Arduino Pasadores reservados:	
10 a 13 se utiliza para SPI	
4 utilizado para la tarjeta SD	
2 W5100 de interrupción (cuando puenteado)	
Pines de entrada analógica	6
Corriente continua para las E / S Pin	40 mA
Corriente de la CC para Pin 3.3V	50 mA
Memoria Flash	32 KB (ATmega328) de los cuales 0,5 KB utilizado por el gestor de arranque
SRAM	2 KB (ATmega328)

EEPROM	1 KB (ATmega328)
Velocidad del reloj	16 MHz
Controlador Ethernet W5100 TCP / IP Embedded	
Power Over Ethernet listo magnética Jack	
Tarjeta Micro SD con adaptadores activos	

Tabla 38: Características generales del Arduino Ethernet
Fuente: Arduino Ethernet.pdf, 2014

2.6.4.2. Sistema de alimentación, potencia, memoria, entrada y salida.

El módulo Ethernet Arduino permite la conexión de una fuente de alimentación externa a través de un conector serial FTDI o cable USB.

La fuente puede venir con un adaptador de AC-DC o batería, este adaptador puede ser conectado al centro positivo enchufe de 2.1mm en el módulo. Mientras que la batería se puede insertar en los pines GND y Vin.

El suministro para la misma puede ser de 6 a 20 voltios. Si se proporcionan menos de 7V, no obstante, el pin de 5V puede suministrar menos de cinco voltios y la junta puede ser inestable. Si se utiliza más de 12V, el regulador de voltaje se puede sobrecalentar y dañar la placa. Rango recomendado es de 7 a 12 voltios.

A continuación se describe de manera general los pines de alimentación:

Vin. Alimentación de entrada del módulo Ethernet, cuando se trata de utilizar una fuente de alimentación externa (en contraposición a 5 voltios de la conexión USB o de otra fuente de alimentación regulada). Se puede suministrar voltaje a través de este pin, o, si el suministro de voltaje a través de la toma de poder, acceder a ella a través de este pin.

5V. Este pin como salida 5V regulado por el regulador en el tablero. El tablero puede ser alimentado ya sea desde el conector de alimentación de CC (7 - 12), el conector USB (5V) o el pasador de Vin del tablero (7-12V). El suministro de tensión a través de los 5V o 3.3V pins no pasa por el regulador, y puede dañar su módulo Ethernet.

3V3. Suministro de 3,3 voltios generados por el regulador a bordo. El drenaje actual máximo es de 50 mA.

GND. Dirección a tierra.

IOREF. Este pin del módulo Ethernet proporciona la referencia de tensión con la que opera el microcontrolador.

- *Ondulación baja de la salida y el ruido (100mVpp)*
- *Rango de tensión de entrada de 36V a 57V*
- *Protección contra sobrecarga y cortocircuito*
- *9V de salida*
- *Alta eficiencia del convertidor DC / DC: typ 75% @ 50% de carga*
- *Aislamiento 1500 V (entrada a salida)*

Memoria

El ATmega328 tiene 32 KB (con 0,5 KB utilizado por el gestor de arranque). También dispone de 2 KB de SRAM y 1 KB de EEPROM (es accesible para ser leído y escrito con la librería EEPROM).

Entrada y salida

Cada uno de los 14 pines digitales en el módulo Ethernet se puede utilizar como una entrada o salida, con la ayuda de las funciones:

utilizando `pinMode ()`, `digitalWrite ()` y `digitalRead ()`.

Operan a 5 voltios. Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia de pull-up (desconectado por defecto) de 20-50 kOhms.

Además, se ha especializado ciertas funciones como:

De serie: 0 (RX) y 1 (TX). Se utiliza para recibir (RX) y de transmisión (TX) TTL datos en serie.

Interrupciones externas: 2 y 3, estos pines pueden ser configurados para activar una interrupción en un valor bajo, un flanco ascendente o descendente, o un cambio en el valor.

PWM¹⁶: 3, 5, 6, 9, y 10 proporcionan PWM de 8 bits con el `() analogWrite` función.

SPI¹⁷: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK), estos pines soportan comunicación SPI utilizando la biblioteca de SPI.

LED: Hay un built-in LED conectado al pin digital 9, cuando el pin es de alto valor, el LED está encendido, cuando es bajo, es apagado. Mientras el pin 13 se utiliza como parte de la conexión SPI.

El módulo Ethernet tiene **6 entradas analógicas**, etiquetados A0 a A5, cada uno de los cuales proporcionan 10 bits de resolución (es decir, 1.024 valores diferentes). Por defecto se miden desde el suelo a 5 voltios, aunque es posible cambiar el extremo superior de su rango con el pin AREF y la `analogReference ()` función.

AREF. Voltaje de referencia para las entradas analógicas.

Restablecer (RST): Sirve para reiniciar el microcontrolador. Normalmente se utiliza para agregar un botón de reinicio para escudos que bloquean el uno en el tablero.

Comunicación

Este módulo Ethernet tiene una serie de instalaciones para la comunicación con un ordenador, otro mismo módulo, u otros microcontroladores.

También se puede conectar a una red con cable a través de Ethernet. Cuando se conecta a una red, tendrá que proporcionar una dirección IP y una dirección MAC.

El lector de tarjetas microSD, se puede acceder a través de la Biblioteca SD . Cuando se trabaja con esta biblioteca, SS es el Pin 4.

¹⁶ **PWM**, Modulación por ancho de pulsos, es una técnica que permite modificar el ciclo de trabajo de una señal periódica, ya sea para transmitir información la cantidad de energía que se envía a una carga.

¹⁷ **Bus de Interfaz de Periféricos Serie** (SPI), es un estándar para controlar casi cualquier dispositivo electrónico que acepte un flujo de bits serie regulado por reloj.

Programación

Es posible programar el módulo Ethernet de dos maneras: a través de la cabecera de la programación serial de 6 pines o con un programador ISP externo.

El encabezado de la programación serial de 6 pines es compatible con los cables USB FTDI y las juntas Sparkfun y Adafruit estilo FTDI¹⁸ básicos de USB a serial de ruptura como el conector USB.

Serial Arduino, cuenta con soporte para rearme automático, permitiendo bocetos que se cargan sin necesidad de pulsar el botón de reinicio en el tablero. Cuando se conecta a un adaptador USB-estilo FTDI, el Arduino Ethernet está apagado el adaptador.

2.7 GLCD (Pantalla Gráfica de Cristal Líquida)



Figura 41: Módulo GLCD
Fuente: Autores, 2014

Una Pantalla Gráfica de Cristal Líquida o GLCD es una pantalla plana formada por una matriz de píxeles monocromos colocados delante de una fuente de luz o reflectora. Es utilizada en proyectos pequeños ya que su consumo de energía es muy bajo, hay versiones de pantallas con diferentes controladores embebidos, como el Samsung KS0107, Samsung KS0108 o el Toshiba T6963.

Las GLCD disponen de una memoria RAM interna del mismo tamaño de la capacidad que dispone la pantalla, por ejemplo si una pantalla tiene un tamaño de

¹⁸ FTDI (**Future Technology Devices International**): Es la conversión de periféricos preexistentes a la interfaz USB.

128 pixeles de largo por 64 pixeles de alto (128x64) tiene una memoria RAM interna de la misma capacidad (128x64).

2.7.1. Características de una GLCD 128x64

- ✓ Conformado por una matriz de puntos de visualización de 128 pixeles de largo por 64 pixeles de alto.
- ✓ Su iluminación de fondo está entre verde-amarillo cuando se enciende.
- ✓ Fácil manejo con microprocesadores de 8-Bits.
- ✓ Bajo poder de consumo.
- ✓ Contiene dos controladores internos un KS0108B y KS0107B.

2.7.2. Descripción de los pines de conexión de una GLCD 128x64

- ✓ PIN 1: VSS (Conexión a tierra)
- ✓ PIN 2: VDD (Conexión de alimentación - +5V)
- ✓ PIN 3: V0 (Voltaje de salida negativa, por lo general es usada con un potenciómetro con el PIN 18 para el ajuste del contraste de los pixeles)
- ✓ PIN 4: D/I (Datos de E/S para el cambio de registro)
- ✓ PIN 5: R/W (Determinas si los datos se van a leer o escribir)
- ✓ PIN 6: E (Habilita la comunicación con la GLCD)
- ✓ PIN 7 - 14 (Especifica un dato de 8-Bits de información)
- ✓ PIN15: CS1 (Indica si se selecciona la primera mitad de la pantalla, pixeles 0-63)
- ✓ PIN16: CS2 (Indica si se selecciona la segunda mitad de la pantalla, pixeles 64-127)
- ✓ PIN17: RESETB (Señal de reinicio, funciona de varias forma dependiendo de la ocasión)
- ✓ PIN18: VEE (Conexión de ajuste de contraste de los pixeles)
- ✓ PIN19: A (Conexión positiva de la luz de fondo, por lo general son +5V)

- ✓ PIN20: K (Conexión negativa de la luz de fondo, por lo general es tierra)

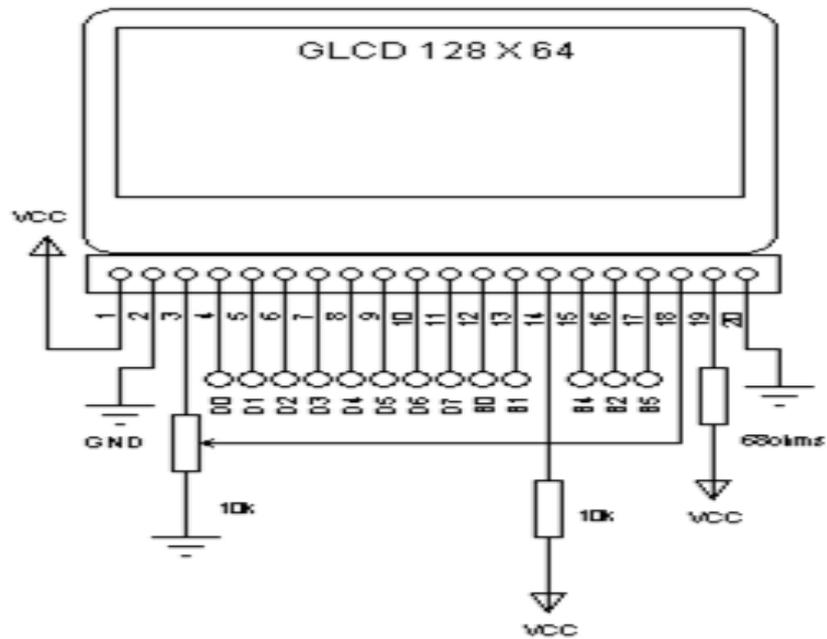


Figura 42: Pines de Conexión de GLCD 128x64
Fuente: <http://www.disca.upv.es/aperles/web51/módulos/glcd/glcd.html>, 2013

2.8. Teclado matricial 4X3

Un teclado matricial en general se caracteriza por contar con filas y columnas, cuando se presiona un botón, se activa la salida correspondiente a la fila y la columna en la que se encuentra dicho botón, al leer este dato se puede saber dónde ha sido presionado.



Figura 43: Teclado matricial 4X3
Fuente: Autores, 2013

2.8.1. Esquema eléctrico

En la siguiente figura se describe el esquema eléctrico del teclado matricial:

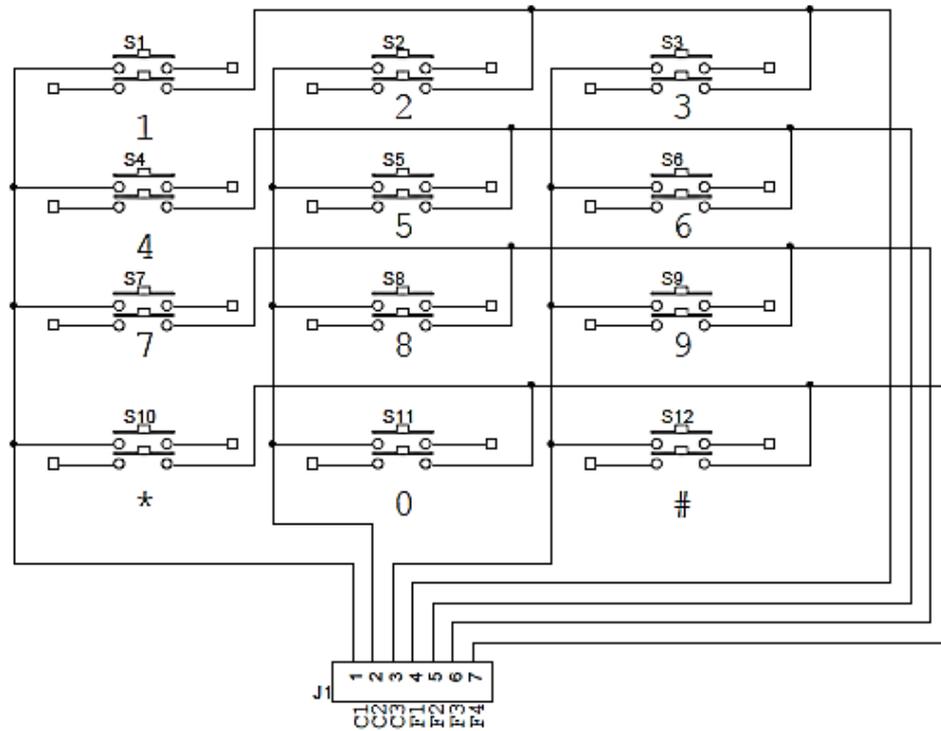


Figura 44: Esquema eléctrico teclado matricial 4X3
Fuente: DatachetTECLADO 4X3.pdf, 2013

2.9. LCD 20X4



Figura 45: LCD 20X4
Fuente: Autores, 2013

El módulo es un módulo del LCD del carácter bajo consumo de energía con un controlador integrado.

El módulo puede ser fácilmente integrado con un microcontrolador.

Formato de presentación: 20 caracteres x 4 líneas.

Tensión de alimentación: 5V.

Totalmente montada y probada LCD Serial 20 × 4 Módulo.

Interfaz analógica.

Backlight Color: azul/negro

Tamaño del artículo: 98 * 60 * 12mm

Peso neto: 78g

2.9.1. Diagrama de bloques y distribución de pines

En la figura se muestra un diagrama de bloques básico:

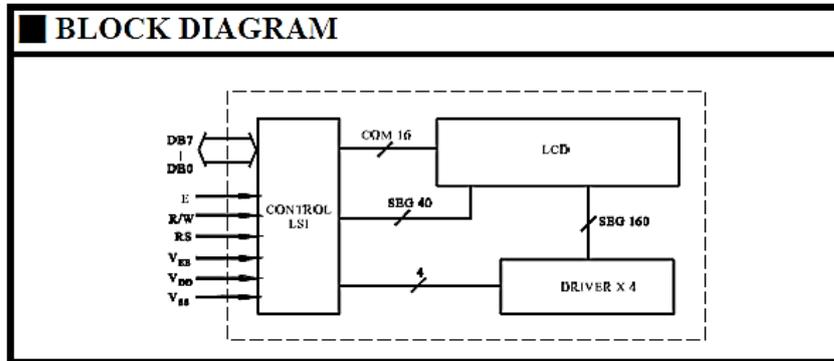


Figura 46: Diagrama de bloques LCD 20X4
Fuente: Systronix_20x4_LCD, 2013

Distribución de pines, simbología y función:

Pin	Symbol	Level	Function
1	VSS	0V	Power GND
2	VDD	+5V	Power supply for logic
3	V0	—	Operation voltage for LCD
4	RS	H/L	H:Data L:Instruction code
5	R/W	H/L	H:Read L:Write
6	E	H/L	Enable signal
7	DB0	H/L	Data bus line
8	DB1	H/L	
9	DB2	H/L	
10	DB3	H/L	
11	DB4	H/L	
12	DB5	H/L	
13	DB6	H/L	
14	DB7	H/L	
15	BLK	0V	Power supply for LED backlight
16	BLA	+5V	

Tabla 39: Distribución de pines LCD 20X4
Fuente: <http://www.patagoniatecnology.com/>, 2013

2.10. Software de diseño y simulación electrónica PROTEUS

“Proteus es una compilación de programas de diseño y simulación electrónica, desarrollado por Labcenter Electronics que consta de los dos programas principales: Ares e Isis, y los módulos VSM y Electra”¹⁹

- Permite diseñar circuitos impresos de mayor calidad.
- Diseño y simulación de PCB (Printed Circuit Board).
- Entorno completamente funcional.
- Visualización de la PCB en 3D.
- Está diseñado según las normas industriales actuales, así los diseños de los módulos del trabajo de grado son legítimos y seguros.

2.10.1. Estructura del programa.

Como se indica en el enunciado pues PROTEUS consta de dos herramientas que se detalla a continuación:

Entorno ISIS: ISIS (Intelligent Schematic Input System) que en español significa Sistema de Enrutado de Esquemas inteligentes, es una potencial herramienta para esta tesis de ingeniería ya que permite diseñar todo el plano eléctrico de cada uno de los módulos o interfaces electrónicas, entre sus herramientas integra desde las resistencias hasta fuentes de alimentación, una gama de PIC muy completa y permite crear propios elementos que no se encuentren en el entorno de ISIS. Además a través del módulo VSM se simula el circuito en tiempo real, así se corrige errores o fallas durante la implementación.

¹⁹ Proteus(electrónica), Wikipedia La Enciclopedia Libre, http://es.wikipedia.org/wiki/Proteus_%28electr%C3%B3nica%29, 2014

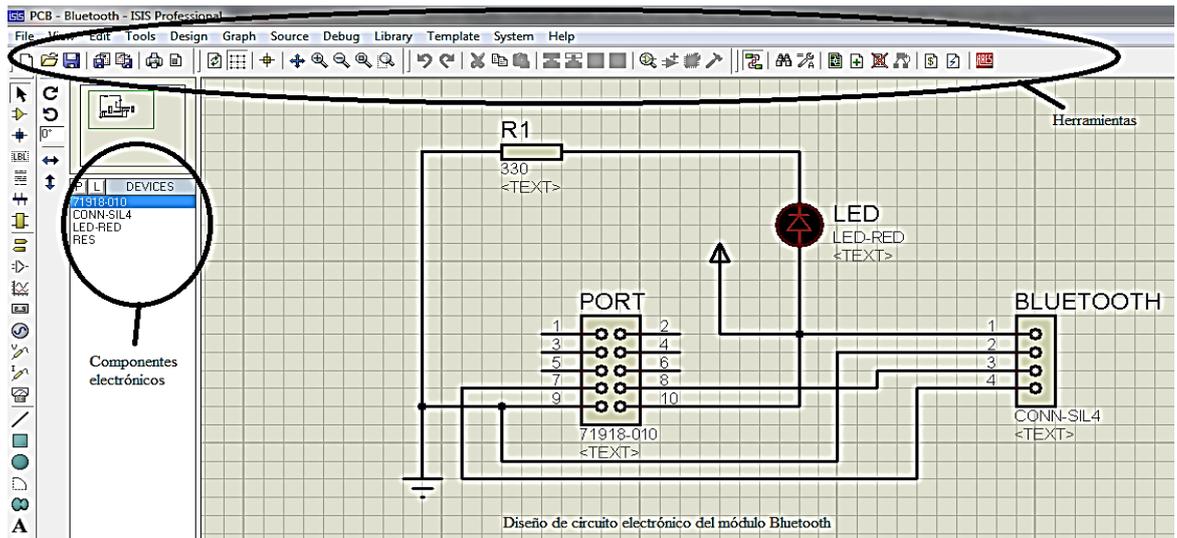


Figura 47: Proteus-Entorno ISIS
Fuente: Autores, 2014

Módulo VSM

El módulo VSM del que se hace mención en el párrafo anterior es un *Sistema Virtual de Modelado* integrada al ISIS, de esta manera se simula en tiempo real y se puede detectar errores para las correcciones respectivas, permite simular una gran variedad de microcontroladores y cada una de sus salidas, a su vez de conectar a estos a diferentes dispositivos como teclados matriciales, LCD, motores, etc.

Entorno ARES

Con sus siglas en inglés “**A**dvanced **R**outing and **E**ding **S**oftware” que significa Software de Edición y Ruteo Avanzado, este entorno permite el enrutamiento, la ubicación y edición de cada uno de los componentes, así se crea el prototipo perfecto para la fabricación de las placas de circuito impresa también conocidas como PCB.

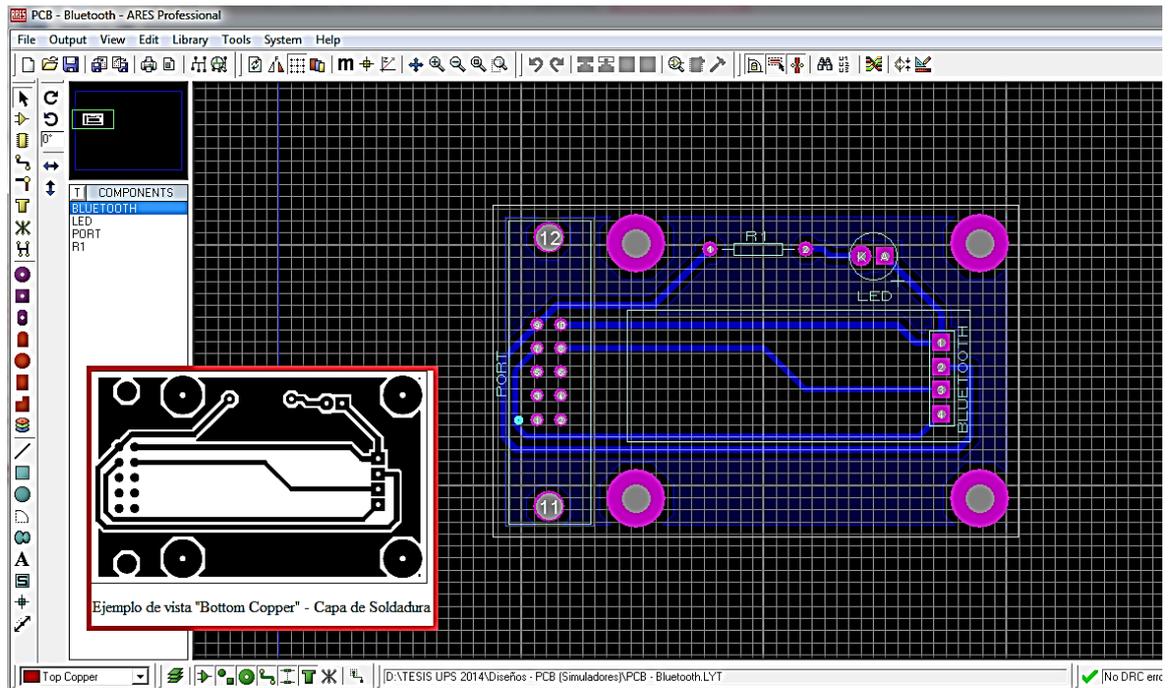


Figura 48: Proteus-Entorno ARES
Fuente: Autores, 2014

Lo interesante de este entorno es que permite la edición del “Top Copper” que es la capa superficial del diseño y también la edición del “Bottom Copper” que es la capa de soldadura.

2.11 Motores de corriente alterna

Se denomina motor de corriente alterna a aquellos motores eléctricos que funcionan con corriente alterna ya sea de 120v o 210v. Un motor es una máquina motriz, esto es, un aparato que convierte una forma determinada de energía en energía mecánica de rotación o par. Un motor eléctrico convierte la energía eléctrica en fuerzas de giro por medio de la acción mutua de los campos magnéticos.

2.11.1 Motor trifásico.

Los motores trifásicos llevan su nombre por el diseño con el cual funcionan, en tres fases de corriente alterna (AC), son utilizados por lo general en aplicaciones industriales. La electricidad de corriente alterna cambia de dirección de negativo a positivo y vuelve muchas veces por segundo. La AC que hay en las viviendas, por ejemplo, pasa de negativo a positivo y viceversa 60 veces por segundo. Cambia de potencia en una onda continua fluida llamada onda sinusoidal. La corriente alterna trifásica tiene tres fuentes de alimentación de AC, todos fuera de fase entre sí. Eso significa que nunca hay dos ondas de AC en el mismo punto al mismo.

2.11.2. Partes principales de un motor trifásico

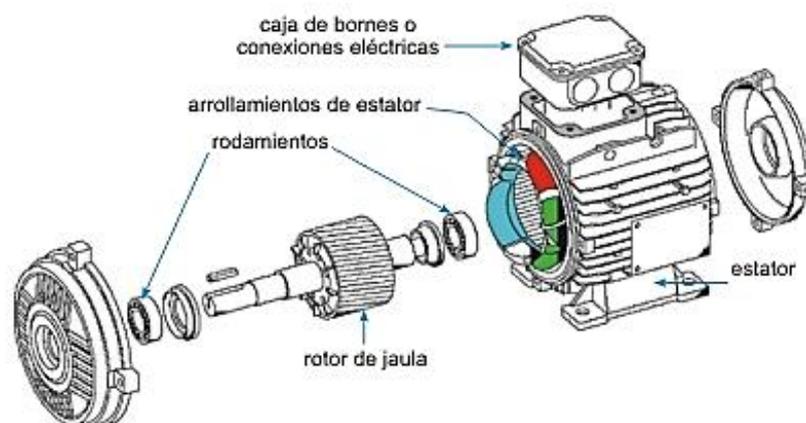


Figura 49: Partes generales de un motor trifásico
Fuente: siemens motores trifásicos libre.pdf, 2013

Un motor de corriente trifásica tiene dos partes principales:

El rotor que gira, también denominado jaula de ardilla, ya que consiste en una red circular de barras y los anillos que tiene la similitud a una jaula conectados a un eje.

El estator que lo gira. Formado por un anillo con tres pares de bobinas, espaciadas uniformemente alrededor del rotor.

2.11.3. Sistema trifásico, tensión de servicio y conexión de motores trifásicos

Una red trifásica de baja tensión está formada por los tres conductores activos R, S y T, y pueden ejecutarse con o sin conductor neutro. Los conductores neutros están unidos al centro de la estrella del generador o del transformador correspondiente al lado de baja tensión. Dos conductores activos, o uno de ellos y el neutro, constituyen un sistema de corriente alterna monofásica.

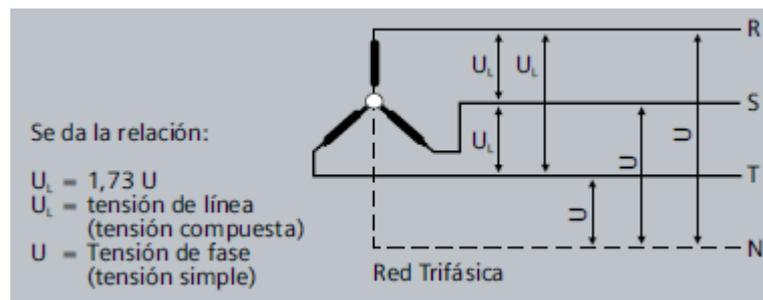


Figura 50: Descripción gráfica de tensión de servicio

Fuente: Siemens Motores Trifásicos libre.pdf, 2013

Ya en el campo de la tensión, hay dos tipos; la tensión de línea²⁰ que es la existente entre dos conductores activos (R, S, T). Y la tensión simple, que es la que hay entre un conductor activo y el neutro es la tensión de la fase. Frecuencia de servicio de 60Hz.

Tensión de línea (V)	Tensión de fase (V)	Denominación usual de la red (V)
208	120	208/120
220	127	220/127
260	150	260/150
380	220	380/220
440	254	440/254

Tabla 40: Tensiones normalizadas para redes de corriente trifásica

Fuente: Siemens Motores Trifásicos libre.pdf, 2013

²⁰ Tensión de línea, también llamada tensión compuesta o tensión de red

Los motores trifásicos se conectan los tres conductores R, S, T. La tensión nominal del motor en la conexión de servicio coincide con la tensión de línea de la red (tensión de servicio).

A continuación se muestra algunas conexiones de servicio de los motores trifásicos y sus potencias nominales:

Ejecución del devanado (V)	Tensión de la red (V)	Devanado en	% Potencia nominal de placa	Tipo de arranque permitido
220-260Δ/440Y ¹⁾ Tamaños 71-160	220	Δ	80	Directo/Y-Δ
	260	Δ	100	Directo/Y-Δ
	380	Δ	100	Directo
	440	Δ	100	Directo
208 - 220 YY/ 440 Y Tamaños 71-112	208	YY	90	Directo
	220	YY	100	Directo
	440	Y	100	Directo
208-220 ΔΔ / 440 Δ Tamaños 132-280	208	ΔΔ	90	Directo/Y-Δ
	220	ΔΔ	100	Directo/Y-Δ
	380	YY		Directo
	440	Δ	100	Directo/Y-Δ

Los motores que se arranquen en estrella-triángulo, la conexión de servicio será en triángulo.

Tabla 41: Conexiones y potencia nominal.

Fuente: Siemens Motores Trifásicos libre.pdf, 2013

El motor que se utilizará para el proyecto de tesis corresponde a la serie Siemens 1LA7-070-4YA60, por ende a continuación la figura de su diagrama esquemático:

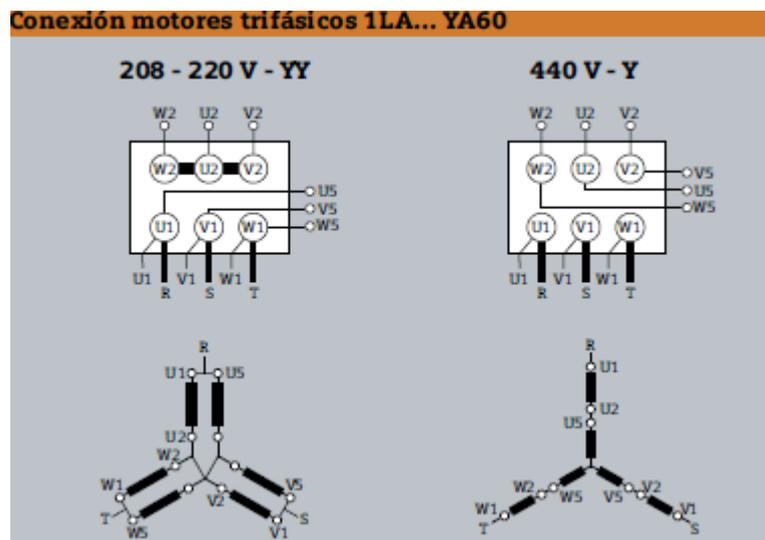


Figura 51: Diagrama esquemático del motor trifásico

Fuente: siemens motores trifásicos libre.pdf, 2013

2.11.4 Motor trifásico Siemens 1LA7-070-4YA60, 1800rpm, 0.5 hp



Figura 52: Tamaño constructivo 070 –motor trifásico
Fuente: siemens motores trifásicos libre.pdf, 2013

Características generales eléctricas:

- Eficiencia IE1 hasta tamaño constructivo 225 (Norma IEC 60034-30²¹)
- Tensiones conmutables 220 / 440 V a 60 Hz.
- Posibilidad de arranque directo en todos los tamaños constructivos.
- Adicionalmente arranque Y - Δ a partir del tamaño 132.
- Clase térmica F - Impregnación al vacío, tecnología VPI
- Tipo de servicio: S1
- Factor de servicio (FS): entre 1.05 a 1.15.
- Temperatura de operación: -15/40oC
- Aptos para ser accionados con variador de velocidad

Características generales mecánicas:

- Carcasa en Aluminio que asegura su bajo peso y excelente conductividad térmica.
- Rodamientos tipo rígidos de bola, de doble sello y con juego interno totalmente cerrados y libres de mantenimiento.
- Vida útil de hasta 20.000 horas de servicio continuo.
- Ejecución IMB²²³ / IMB35 / IMB5
- Con retenedor CD Ring en el platillo AS

²¹ Norma IEC 60034-30: Define las clases de eficiencia IE(International Efficiency) de los motores AC, Electromagazine No35.PDF, 2007

²² IMB: Interfaz Mecánica Básica en los motores, Motores síncronos de imanes.pdf, 2013

- Protección Mecánica IP55
- Para la serie 1LA7 platillos en aluminio y 1LA5 con platillos en fundición.

Tabla de motores trifásicos IEC de propósito general

Referencia	Nivel de eficiencia	Tamaño Constructivo	Potencia		Eficiencia η %	FS	Corriente (A)		Peso Kg
			HP	KW			220V	440V	
VELOCIDAD 1.800 RPM (4 polos)									
1LA7 070-4YA60	-	71	0,5	0,4	63,6	1,15	1,9	0,9	5
1LA7 073-4YA60	-	71	0,75	0,6	64,0	1,15	2,9	1,4	6
1LA7 080-4YA60	IE1	80	1,0	0,7	64,2	1,15	3,5	1,7	8
1LA7 083-4YA60	IE1	80	1,5	1,1	65,9	1,15	5,0	2,5	9
1LA7 096-4YB60	IE1	90 L	2,0	1,5	82,3	1,15	6,6	3,3	15
1LA7 096-4YC60	IE1	90 L	2,4	1,8	81,5	1,15	7,4	3,7	15
1LA7 097-4YB60	IE1	90 L	3,0	2,2	83,0	1,15	8,6	4,3	16
1LA7 112-4YB60	IE1	112 M	4	3,0	85,5	1,15	12,2	6,1	25
1LA7 112-4YC60	IE1	112 M	5,0	3,7	85,3	1,15	15,8	7,9	25
1LA7 114-4YB60	IE1	112 M	6	4,5	85,7	1,05	18,0	9,0	28
1LA7 113-4YA60	IE0	112 M	6,6	4,9	78,0	1,05	19,6	9,8	29
1LA7 115-4YB60	IE1	112 M	7,5	5,6	87,5	1,15	20,0	10,0	30
1LA7 132-4YB70	IE1	132 S/M	10,0	7,5	88,3	1,15	24,4	12,2	52
1LA7 132-4YC70	IE1	132 S/M	12	9,0	88,1	1,15	34,0	17,0	52
1LA7 135-4YB70	IE1	132 S/M	15	11,2	89,8	1,15	43,0	21,5	61
1LA7 164-4YB70	IE1	160 M/L	20	14,9	90,9	1,15	55,6	27,8	79
1LA7 167-4YB70	IE1	160 M/L	25	18,7	91,8	1,15	66,0	33,0	89
1LA5 183-4YB80	IE1	180 M/L	30	22,4	91,8	1,05	78,0	39,0	130
1LA5 187-4YB80	IE1	180 M/L	40	29,8	92,5	1,05	104,0	52,0	140
1LA5 207-4YB80	IE1	200 L	50	37,3	92,5	1,15	126,0	63,0	191
1LA5 220-4YB80	IE1	225 S/M	60	44,8	93,0	1,15	144,0	72,0	226
1LA5 223-4YB80	IE1	225 S/M	75	56,0	94,1	1,15	200,0	100,0	242

Tabla 42: Motores trifásicos Eficiencia IE1
Fuente: Productos eléctricos industriales Siemens.pdf, 2014

2.12. Controladores PID

Los controladores PID como su nombre lo indica tienen una parte proporcional, una derivativa y una integral, esto se puede observar en el siguiente diagrama de bloques:

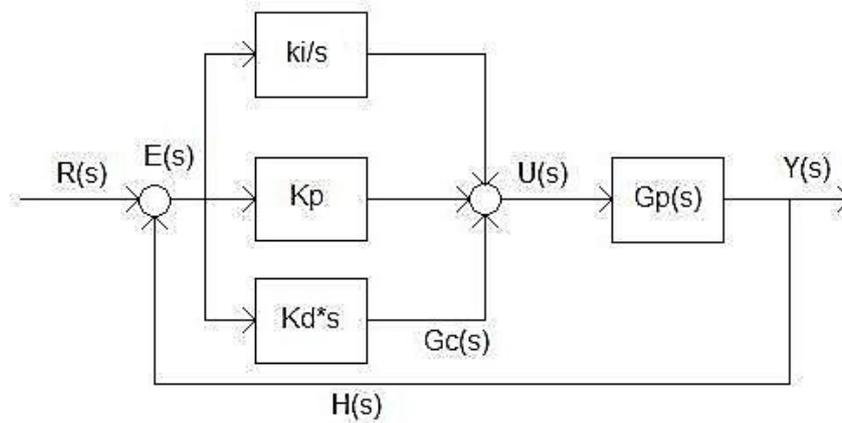


Figura 53: Diagrama de bloques controlador PID.

Fuente: Fuente: Control automático,

http://www.sapiensman.com/control_automatico/control_automatico5.htm

Por lo tanto la ecuación de lazo abierto es:

$$G_p(s) [(K_D + s) + (K_P) + (k_i/s)] = Y(s)/E(s)$$

El propósito de este controlador es el de reducir la señal de error "E(s)".

2.13. Variador de velocidad Altivar 312



Figura 54: Variador de velocidad Altivar 312.
Fuente: Parametros_ATV312, 2012

Este equipo es un convertidor de frecuencia para motores asíncronos trifásicos 200•••600 V de 0,18 a 15 Kw. Entre sus características está su resistencia, fácil uso, transportable debido a sus dimensiones pequeñas y es de fácil instalación.

Este modelo consta con funciones integradas las cuales están especialmente adaptadas para responder a las aplicaciones de máquinas industriales sencillas.

Consta de diferentes tarjetas de comunicación disponibles de forma opcional, así el variador Altivar 312 se integra a la perfección en las principales arquitecturas de automatismo.

2.13.1. Aplicaciones generales y funciones básicas

Entre las aplicaciones más comunes en las cuales es muy útil el variador Altivar 312, se puede enumerar:

- Transporte (cintas transportadoras pequeñas, elevadores, etc.),
- Máquinas de envase y embalaje (envasadoras, etiquetadoras, etc.),

- Maquinas especiales (mezcladores, trituradores, maquinaria textil, etc.),
- Bombas, compresores, ventiladores.

El variador Altivar 312 dispone de dos salidas de relé, seis entradas lógicas, tres entradas analógicas y una salida lógica/analógica. Entre sus funciones básicas tenemos:

- Protecciones para motor y variador,
- Rampas de aceleración y desaceleración, lineales, en S, en U o personalizadas,
- Control local de la referencia velocidad con el botón de navegación,
- +/- velocidad,
- 16 velocidades preseleccionadas,
- Consignas y regulador PI,
- Mando 2 hilos/3 hilos,
- Lógica de freno,
- Recuperación automática con búsqueda de velocidad y re arranque automático.
- Configuración de fallos y de tipos de paradas.
- memorización de la configuración en el variador.
- Se pueden asignar varias funciones a una misma entrada lógica.

2.13.2. Características del variador de velocidad ATV132

A través de las siguientes tablas se resume las características del ATV132.

Características del accionamiento			
Rango de frecuencia de salida	Hz	0...500	
Frecuencia de corte	KHz	Frecuencia de corte nominal: 4 kHz sin desclasificación en régimen permanente. Ajustable en funcionamiento de 2...16 kHz Superados los 4 kHz, debe aplicarse una desclasificación en la corriente nominal del variador y la corriente nominal del motor no deberá superar este valor. Ver las curvas de desclasificación en página 52	
Gama de velocidad		1...50	
Sobrepasar transitorio		170...200% del par nominal del motor (valor típico)	
Par de frenado	Con resistencia de frenado	ATV 312H●●●●●	100% del par nominal del motor permanentemente y hasta 150% durante 60 s
	Sin resistencia de frenado	ATV 312HD18M2	150% del par nominal del motor (valor típico)
		ATV 312H037M2...HD75M2 ATV312H018M3...HD75M3 ATV 312H037N4...HD75N4 ATV 312H075S6	100% del par nominal del motor (valor típico)
		ATV 312HU11M2, HU15M2 ATV 312HU11M3, HU15M3 ATV 312HU11N4, HU15N4 ATV 312HU15S6	50% del par nominal del motor (valor típico)
		ATV 312HU22M2 ATV 312HU22M3 ...HD15M3 ATV 312HU22N4...HD15N4 ATV 312HU22S6...HD15S6	30% del par nominal del motor (valor típico)
	Corriente transitoria máxima		150% de la corriente nominal del variador durante 60 segundos (valor típico)
	Leyes de control motor		Ley estándar (tensión/frecuencia) Ley de altas prestaciones (control vectorial de flujo sin captador) Ley bomba/ventilador (ley cuadrática Kn2) Ley ahorro de energía (dedicada ventilación)
Ganancias del bucle de frecuencia		Ajuste de fábrica con la estabilidad y la ganancia del bucle de velocidad Ajustes posibles para máquinas de par elevado resistente o gran inercia, o para máquinas de ciclos rápidos	
Compensación de deslizamiento		Automática con cualquier carga. Posibilidad de eliminación o ajuste	

Características eléctricas de potencia			
Alimentación	Tensión	V	200 - 15% ... 240 + 10% monofásica para ATV 312●●●●M2 200 - 15% ... 240 + 10% trifásica para ATV 312●●●●M3 380 - 15% ... 500 + 10% trifásica para ATV 312●●●●N4 525 - 15% ... 600 + 10% trifásica para ATV 312●●●●S6
	Frecuencia	Hz	50... 60 + 5%
Corriente de cortocircuito presunta ICC	ATV 312●●●●M2	A	≤ 1000 (ICC en el punto de conexión) para alimentación monofásica
	ATV 312HD18M3...HU40M3 ATV 312H037N4...HU40N4 ATV 312H075S6...HU40S6	A	≤ 5000 (ICC en el punto de conexión) para alimentación trifásica
	ATV 312HU55M3...HD15M3 ATV 312HU55N4...HD15N4 ATV 312HU55S6...HD15S6	A	≤ 22000 (ICC en el punto de conexión) para alimentación trifásica
Tensiones de alimentación y salida del variador			
			Tensión de alimentación del variador
			Tensión de salida del variador para motor
	ATV 312H●●●●M2	V	200...240 monofásica
	ATV 312H●●●●M3	V	200...240 trifásica
	ATV 312H●●●●N4	V	380 ... 500 trifásica
	ATV 312H●●●●S6	V	525...600 trifásica

Características de conexión (bornes del variador para la alimentación de la red, la salida del motor, el bus ccy y la resistencia de frenado)			
Bornes del variador		L1, L2, L3, U, V, W, PC/+, PA/+, PB	
Capacidad máxima de conexión y par de apriete	ATV 312HD18M2...HD75M2 ATV 312HD18M3...HU15M3		2,5 mm ² (AWG 14) 0,8 Nm
	ATV 312HU11M2...HU22M2 ATV 312HU22M3...HU40M3 ATV 312H037N4...HU40N4 ATV 312H075S6...HU40S6		5 mm ² (AWG 10) 1,2 Nm
	ATV 312HU55M3, HU75M3 ATV 312HU55N4, HU75N4 ATV 312HU55S6, HU75S6		16 mm ² (AWG 6) 2,5 Nm
	ATV 312HD11M3, HD15M3 ATV 312HD11N4, HD15N4 ATV 312HD11S6, HD15S6		25 mm ² (AWG 3) 4,5 Nm
Aislación galvánica			Aislación galvánica entre potencia y control (entradas, salidas, alimentaciones)

Características eléctricas de control	
Fuentes internas de alimentación disponibles	<p>Protegidas contra los cortocircuitos y las sobrecargas:</p> <ul style="list-style-type: none"> • 1 alimentación \approx 10 V (0/+ 8%) para el potenciómetro de consigna (2,2 a 10 kΩ), corriente máxima 10 mA, • 1 alimentación \approx 24 V (mín. 19 V, máx. 30 V) para las entradas lógicas de control, corriente máxima 100 mA.
Entradas analógicas	<p>Tiempo de muestreo < 8 ms Resolución: 10 bits Precisión: \pm 4,3% Linealidad: \pm 0,2% del valor máximo de la escala Utilización: <ul style="list-style-type: none"> • 100 m como máximo con cable blindado • 25 m como máximo con cable no blindado </p>
AI1	1 entrada analógica en tensión c 0...10 V, impedancia 30 kW, tensión máxima de no destrucción 30 V
AI2	1 entrada analógica en tensión bipolar \pm 10 V, impedancia 30 kW, tensión máxima de no destrucción 30 V
AI3	1 entrada analógica en corriente X-Y mA programando X e Y de 0 a 20 mA, con impedancia 250 Ω
Salidas analógicas en tensión o en corriente configurable como salida lógica	<p>2 salidas analógicas: <ul style="list-style-type: none"> • 1 salida analógica en tensión (AOV), • 1 salida analógica en corriente (AOC) configurable en salida lógica. Estas 2 salidas analógicas no pueden utilizarse al mismo tiempo</p>
AOV	Salida analógica en tensión 0...10 V, impedancia de carga mín. 470 Ω Resolución 8 bits, precisión \pm 1%, linealidad \pm 0,2% del valor máximo de la escala
AOC	Salida analógica en corriente 0...20 mA, impedancia de carga máx. 800 Ω Resolución 8 bits, precisión \pm 1%, linealidad \pm 0,2% Salida analógica AOC configurable como salida lógica 24 V, 20 mA máx., impedancia de carga mín. 1,2 k Ω Tiempo de muestreo < 8 ms
Salidas de relés	
R1A, R1B, R1C	1 salida lógica de relé, un contacto "NC" y un contacto "NA" con punto común. Poder de conmutación mínimo: 10 mA para 5 V Poder de conmutación máximo: <ul style="list-style-type: none"> • en carga resistiva ($\cos \phi = 1$ y L/R = 0 ms): 5 A para \sim 250 V o \approx 30 V, • en carga inductiva ($\cos \phi = 0,4$ y L/R = 7 ms): 2 A para \sim 250 V o \approx 30 V Tiempo de muestreo < 8 ms Conmutación: 100.000 maniobras
R2A, R2B	1 salida lógica de relé, un contacto "NC", contacto abierto en fallo. Poder de conmutación mínimo: 10 mA para 5 V Poder de conmutación máximo: <ul style="list-style-type: none"> • en carga resistiva ($\cos \phi = 1$ y L/R = 0 ms): 5 A para \sim 250 V o \approx 30 V, • en carga inductiva ($\cos \phi = 0,4$ y L/R = 7 ms): 2 A para \sim 250 V o \approx 30 V Tiempo de muestreo < 8 ms
Entradas lógicas LI	
LI1...LI6	6 entradas lógicas programables, compatibles con autómatas nivel 1, norma IEC/EN 61131-2 Impedancia 3,5 k Ω Alimentación \approx 24 V interna o \approx 24 V externa (mín. 19 V, máx. 30 V) Intensidad máxima: 100 mA Tiempo de muestreo < 4 ms La multiasignación permite configurar varias funciones en una misma entrada (ejemplo: LI1 asignada a marcha adelante y velocidad preseleccionada 2, LI3 asignada a marcha atrás y velocidad preseleccionada 3)
Lógica positiva (Source)	Estado 0 si < 5 V o entrada lógica no cableada Estado 1 si > 11 V
Lógica negativa (Sink)	Estado 0 si > 19 V o entrada lógica no cableada Estado 1 si < 13 V
Posición CLI	Conexión con salida de autómatas programables (ver esquema en página 50)
Capacidad máxima de conexión y par de apriete de las entradas/salidas	2,5 mm ² (AWG 14) 0,6 Nm

Características del puerto de comunicación		
Protocolos disponibles		Protocolos Modbus y CANopen integrados en el variador. A estos dos protocolos puede accederse mediante un solo conector RJ45 situado bajo el variador.
Protocolo Modbus		
Estructura	Conector	Tipo RJ45
	Interfaz física	RS 485
	Modo de transmisión	RTU
	Velocidad de transmisión	Puede configurarse mediante la interfaz hombre-máquina, las terminales remotas o el software de puesta en marcha SoMove : 4800, 9600 ó 19200 bits/s
	Número máximo de equipos	31
	Dirección	1 a 247, puede configurarse mediante la interfaz hombre-máquina, las terminales remotas o el software de puesta en marcha SoMove
Servicios	Perfiles funcionales	CiA 402
	Mensajería	Read Holding Registers (03) Write Single Register (06) Write Multiple Registers (16) Read Device Identification (43)
	Vigilancia de la comunicación	Configurable
Protocolo CANopen		
Estructura	Conector	Tipo RJ45
	Gestión de red	Esclavo
	Velocidad de transmisión	Puede configurarse mediante la interfaz hombre-máquina, las terminales remotas o el software de puesta en marcha SoMove : 10, 20, 50, 125, 250, 500 kbit/s ó 1 Mbit/s
	Número máximo de equipos	127
	Dirección (Node ID)	1 a 127, puede configurarse mediante la interfaz hombre-máquina, las terminales remotas o el software de puesta en marcha SoMove
Servicios	Número de PDO (Process Data Objects/objetos de datos de proceso)	2 PDO: PDO 1: no configurable PDO 6: configurable
	Modos de los PDO	PDO 1: asincrónico PDO 6: asincrónico, sincrónico, asincrónico cíclico
	Número de SDO (Service Data Objects/objetos de datos de servicio)	1 SDO en recepción y 1 SDO en emisión
	Perfiles funcionales	CiA 402
	Vigilancia de la comunicación	Node guarding y Heartbeat
Diagnóstico	Mediante LED	En interfaz hombre-máquina
Archivo de descripción		Hay disponible un archivo eds en nuestro sitio web www.schneider-electric.com.ar

Tabla 43: Características de variador de velocidad ATV132
Fuente: Parametros_ATV312, 2012

2.13.3 Descripción del panel de control

Mediante la siguiente figura se describe cada una de las partes de la interfaz de control del variador de velocidad:

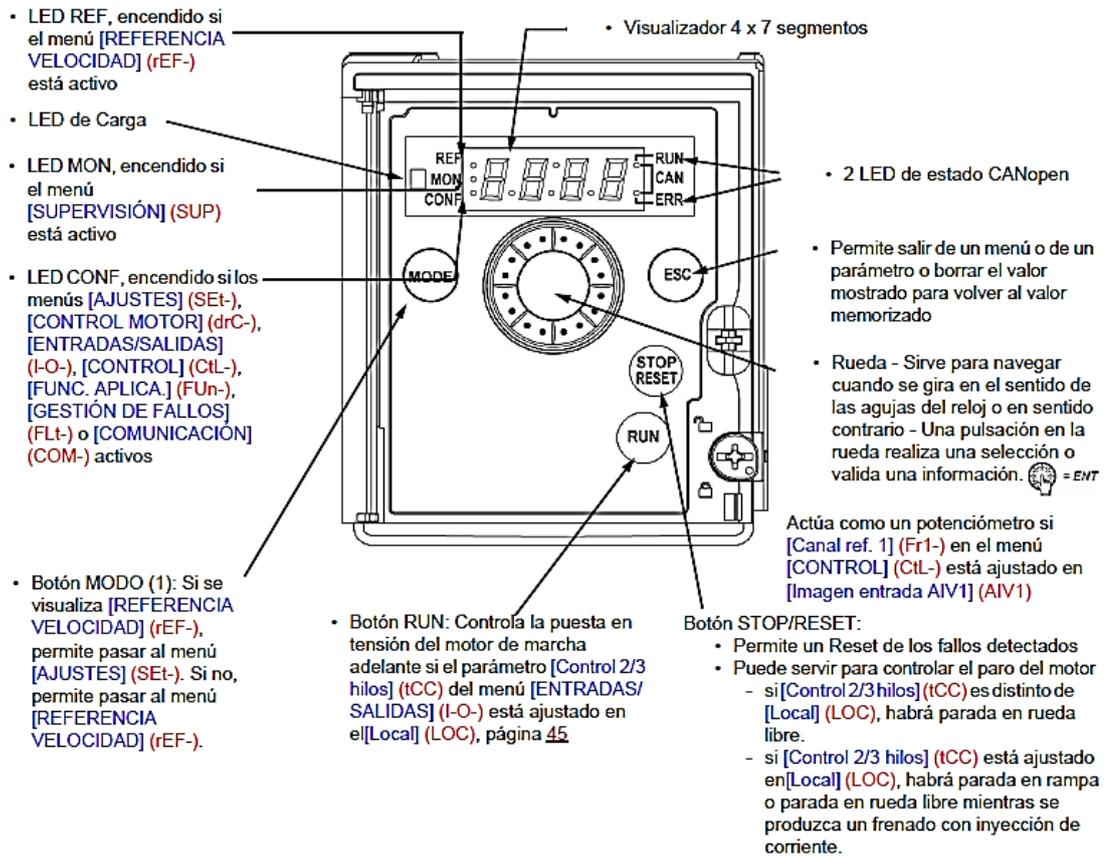


Figura 55: Descripción de partes variador de velocidad Altivar 312.
Fuente: Parametros_ATV312, 2012

2.13.4. Visualización de parámetros iniciales y estructura de los menús.

En esta figura se puede describir los parámetros básicos que aparecerán en el variador de velocidad ATV312²³.

Estructura general de los menús:

- 4 3.0** : Visualización del parámetro seleccionado en el menú [SUPERVISIÓN] (SUP-) (por defecto: frecuencia del motor). En caso de limitación de corriente, el parámetro visualizado parpadea. En este caso, si un terminal gráfico ATV61/ATV71 está conectado al variador, se visualizará CLI en la parte superior izquierda.
- 1 n 1 b** : Secuencia de inicialización.
- r d y** : Variador listo
- d C b** : Frenado por inyección de corriente continua en curso
- n 5 t** : Parada en rueda libre
- F 5 t** : Parada rápida
- t U n** : Autoajuste en curso

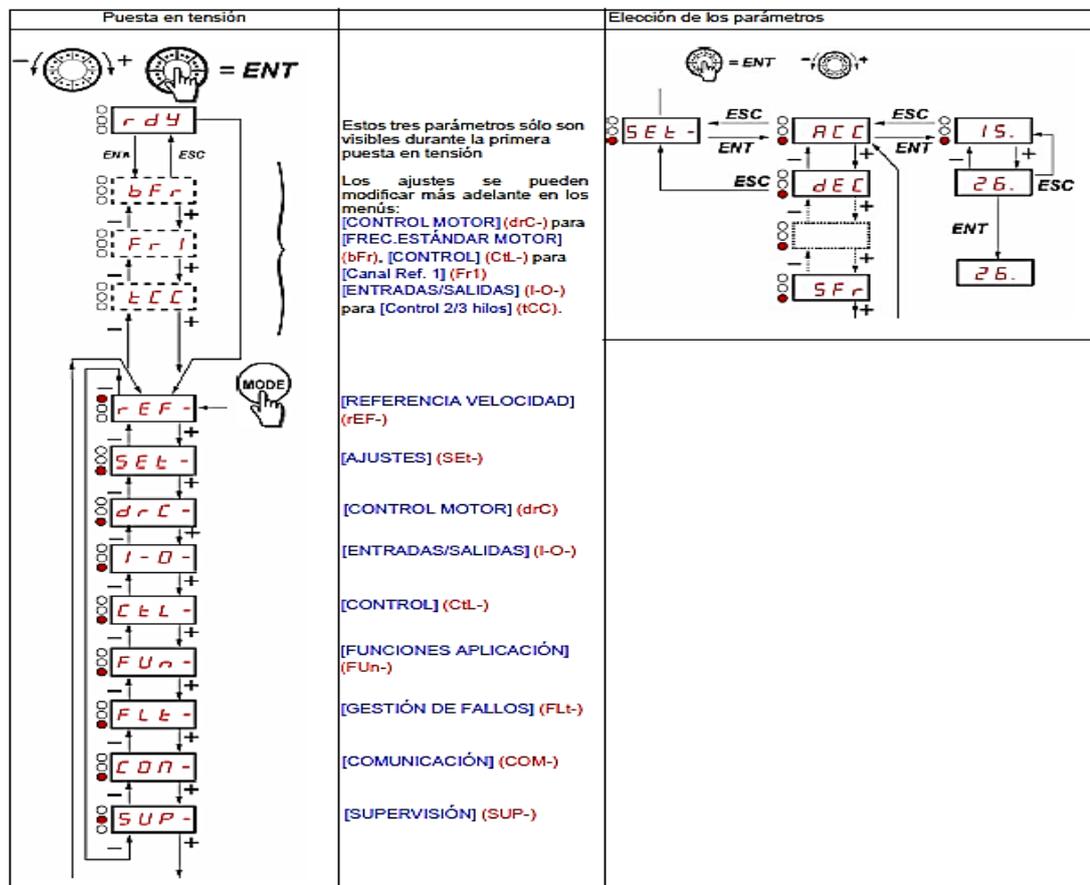


Tabla 44: Estructura general de los menús Altivar132
Fuente: Parametros_ATV312.pdf, 2012

²³ Se consulta mayor información de configuración en Parametros_ATV312.pdf en la sección ANEXOS.

Conexión básica para arranque del motor:

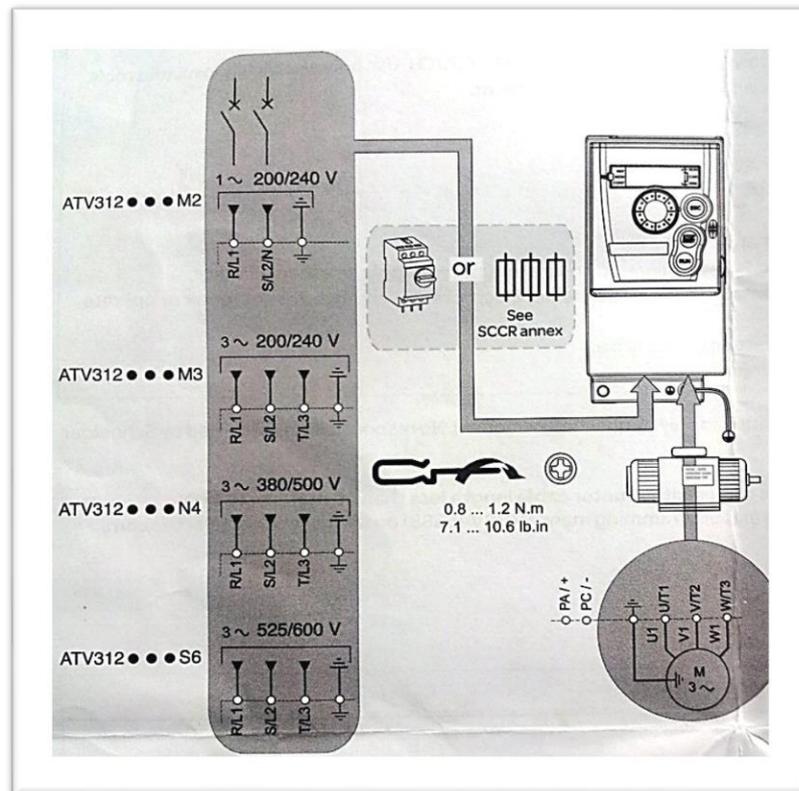


Figura 56: Variador de velocidad Altivar 312.
Fuente: Manual de guía rápida, 2014

2.14 LABVIEW (acrónimo de Laboratory Virtual Instrumentation Engineering Workbench)



Figura 57: Logotipo LabVIEW

Fuente: Programación en LabVIEW,

http://www.inele.ufro.cl/apuntes/LabView/Manuales/Curso%20_Labview.pdf, 2013

Plataforma y entorno de desarrollo para diseñar sistemas, con un lenguaje de programación visual totalmente gráfico. Recomendado para sistemas hardware y software de pruebas, control y diseño, simulado o real, pues acelera la productividad. El lenguaje que usa se llama lenguaje G, donde la G simboliza que es lenguaje Gráfico.

“Este programa fue creado por National Instruments (1976) para funcionar sobre máquinas MAC, salió al mercado por primera vez en 1986. Ahora está disponible para las plataformas Windows, UNIX, MAC y GNU/Linux. La última versión es la 2012, con la increíble demostración de poderse usar simultáneamente para el diseño del firmware de un instrumento RF de última generación, a la programación de alto nivel del mismo instrumento, todo ello con código abierto.”²⁴

Los programas desarrollados con LabVIEW se llaman Instrumentos Virtuales, o VIs, y su origen provenía del control de instrumentos, aunque actualmente abarca otras áreas como las comunicaciones, matemáticas, programación embebida.

²⁴ Programación en Labview, http://www.inele.ufro.cl/apuntes/LabView/Manuales/Curso%20_Labview.pdf, 2013

Dentro de sus objetivos pretende reducir el tiempo de desarrollo de aplicaciones de todo tipo (no sólo en ámbitos de Pruebas, Control y Diseño) y el permitir la entrada a la informática a profesionales de cualquier otro campo. LabVIEW consigue combinarse con todo tipo de software y hardware, tanto del propio fabricante - tarjetas de adquisición de datos, PAC, he incluso la que se utilizará en esta tesis de grado que es la tarjeta de adquisición de datos NI-USB609, instrumentos y otro Hardware-de otros fabricantes.

2.14.1. Características

Su principal característica es la facilidad de uso, válido para programadores profesionales como para personas con pocos conocimientos en programación pueden hacer programas relativamente complejos, imposibles para ellos de hacer con lenguajes tradicionales. Cada programa realizado en el software se lo conoce como instrumentos virtuales (VIs) LabVIEW pueden crearse programas de miles de VIs (equivalente a millones de páginas de código texto) para aplicaciones complejas, programas de automatizaciones de decenas de miles de puntos de entradas/salidas, proyectos para combinar nuevos VIs con VIs ya creados, etc.

2.14.2. Interfaces de comunicaciones:

A continuación se muestra las interfaces con las cuales puede trabar LabVIEW:

- Puerto serie
- Puerto paralelo
- GPIB
- PXI
- VXI
- TCP/IP, UDP, DataSocket
- Irda
- Bluetooth
- USB
- OPC...
- Capacidad de interactuar con otros lenguajes y aplicaciones:
- DLL: librerías de funciones

- .NET
- ActiveX
- Multisim
- Matlab/Simulink
- AutoCAD, SolidWorks, etc.
- Herramientas gráficas y textuales para el procesado digital de señales.
- Visualización y manejo de gráficas con datos dinámicos.
- Adquisición y tratamiento de imágenes.
- Control de movimiento (combinado incluso con todo lo anterior).
- Tiempo Real estrictamente hablando.
- Programación de FPGAs para control o validación.
- Sincronización entre dispositivos.

2.14.3 Programa en LabVIEW

Debido a que LabVIEW es una herramienta gráfica de programación, sus programas no se escriben, sino que se dibujan, debido a esto se facilita mucho al programador. Al tener ya pre-diseñados una gran cantidad de bloques, se le da una gran utilidad al programa a la hora de la creación del proyecto, con lo cual en vez de estar una gran cantidad de tiempo en programar un dispositivo/bloque, se le permite invertir mucho menos tiempo y dedicarse un poco más en la interfaz gráfica y la interacción con el usuario final. Cada VI consta de dos partes diferenciadas:

Panel Frontal: El Panel Frontal es la interfaz con el usuario, utilizada para interactuar cuando el programa se está ejecutando. Se observará los datos del programa actualizados en tiempo real (como van fluyendo los datos, un ejemplo sería una calculadora, donde se coloca las entradas, y la salida sería el resultado). En esta interfaz se definen los controles (los usamos como entradas, pueden ser botones, marcadores etc.) e indicadores (los usamos como salidas, pueden ser gráficas como por ejemplo una LDC.).

Diagrama de Bloques: Es el programa propiamente dicho, donde se define su funcionalidad, aquí se colocan íconos que realizan una determinada función y se interconectan a través del icono/conector que es el medio utilizado para conectar un VI con otros VIs.

En la siguiente figura se muestra un Diagrama de Bloques de un programa en el que se genera un array de 100 elementos aleatorios, a continuación se hace la FFT de este array y se muestra en una gráfica:

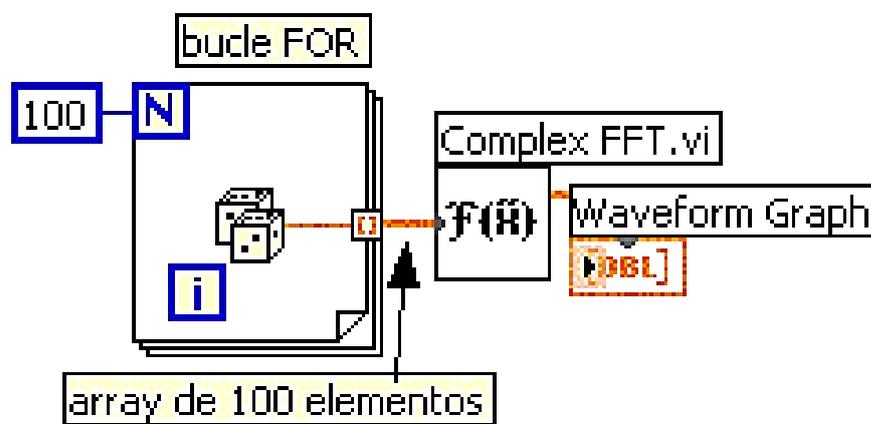


Figura 58: Diagrama de bloques en LabVIEW

Fuente: LabVIEW-Wikipedia, <http://es.wikipedia.org/wiki/LabVIEW>, 2013

2.14.4. Ventajas de LabVIEW al adquirir datos y procesar señales.

1. Potencia y Flexibilidad a través de Software

Uno de los beneficios más grandes de usar un dispositivo DAQ basado en PC es que se puede usar software para personalizar la funcionalidad y visualización del sistema de medidas para cumplir con las necesidades de una aplicación. Realizar medidas manualmente es costoso, lento y propenso a errores en tanto que los sistemas definidos por software las realizan rápidamente y consistentemente. El software NI LabVIEW ofrece una sola interfaz de programación para los dispositivos DAQ, resultando en una estrecha integración de hardware y software.

2. Acelerar la Productividad

LabVIEW permite ser más productivo al enfocarse en datos y las operaciones realizadas en esos datos, así como al abstraer gran parte de la complejidad administrativa de la programación de PC como asignación de memoria y sintaxis del lenguaje. Las bibliotecas integradas de software específico para ingeniería permiten fácil adquisición, análisis, control y compartir datos, para poder enfocarse en datos y no en la programación.

3. Basado en Innovación

National Instruments ha sido uno de los primeros desarrolladores de hardware y software de instrumentación virtual por más de 25 años.

4. Distribuye Aplicaciones Autónomas

Crea aplicaciones autónomas para programas que necesitan ser desplegados a otros sistemas con el Constructor de Aplicaciones de LabVIEW.

5. Transformar Datos sin Procesar en Resultados

LabVIEW incluye miles de funciones avanzadas de análisis creadas especialmente para ingenieros y científicos, todo con archivos y documentación de ayuda detallada. Con estas potentes herramientas, se puede realizar procesamiento de señales avanzado; análisis de frecuencia; probabilidad y estadísticas; ajuste de curvas; interpolación; procesamiento digital de señales y mucho más. También se puede extender LabVIEW con juegos de herramientas para sonido y vibración; visión artificial; comunicaciones RF y análisis transitorio de señales o de corta duración.

2.14.5. Tarjeta NI_USB6009

La NI USB-6008/6009 está constituida por ocho entradas analógicas (AI), dos (AO) canales de salida analógica, 12 entradas / salidas digitales (DIO), y un contador de 32 bits con una interfaz USB de alta velocidad.

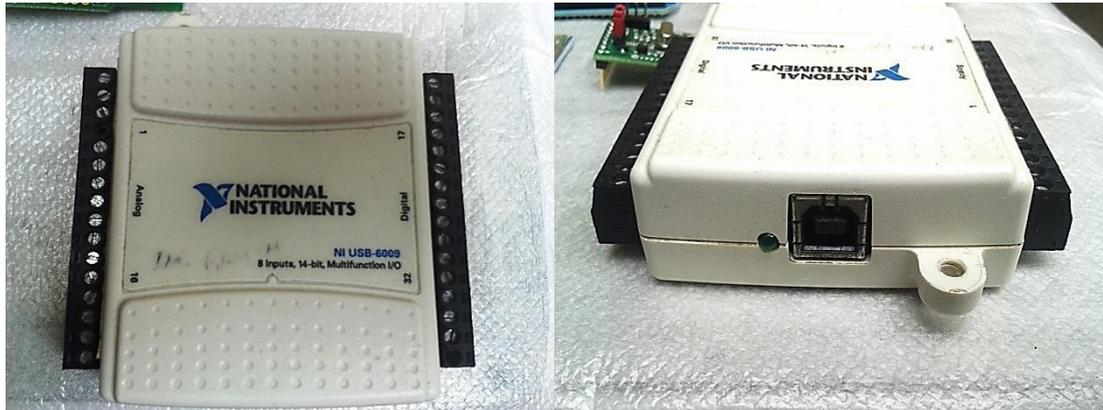


Figura 59: Tarjeta NI USB6009
Fuente: Autores, 2014

Características básicas:

Feature	USB-6009
AI Resolution	14 bits differential, 13 bits single-ended
Maximum AI Sample Rate, Single Channel*	48 kS/s
Maximum AI Sample Rate, Multiple Channels (Aggregate)*	42 kS/s
DIO Configuration	Open collector or active drive
* Might be system dependent.	

Tabla 45: Características básicas de la NI USB-6009
Fuente: NI_6008/6009_ADC_manual.pdf, 2012

Diagrama de bloques:

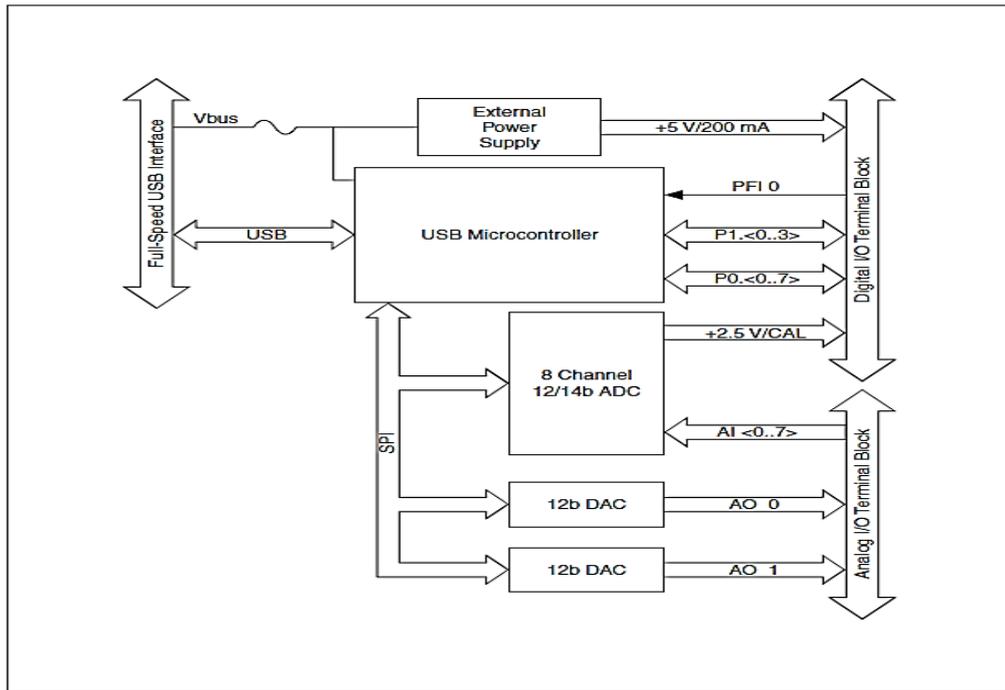


Figura 60: Diagrama de bloques NI USB-6009
Fuente: NI_6008/6009_ADC_manual.pdf, 2012

A continuación se describe los terminales análogos:

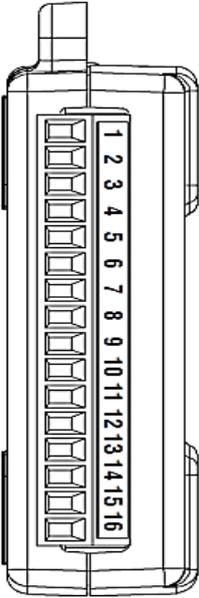
Module	Terminal	Signal, Single-Ended Mode	Signal, Differential Mode
	1	GND	GND
	2	AI 0	AI 0+
	3	AI 4	AI 0-
	4	GND	GND
	5	AI 1	AI 1+
	6	AI 5	AI 1-
	7	GND	GND
	8	AI 2	AI 2+
	9	AI 6	AI 2-
	10	GND	GND
	11	AI 3	AI 3+
	12	AI 7	AI 3-
	13	GND	GND
	14	AO 0	AO 0
	15	AO 1	AO 1
	16	GND	GND

Tabla 46: Terminales análogos NI USB-6009
Fuente: NI_6008/6009_ADC_manual.pdf, 2012

Descripción de terminales digitales:

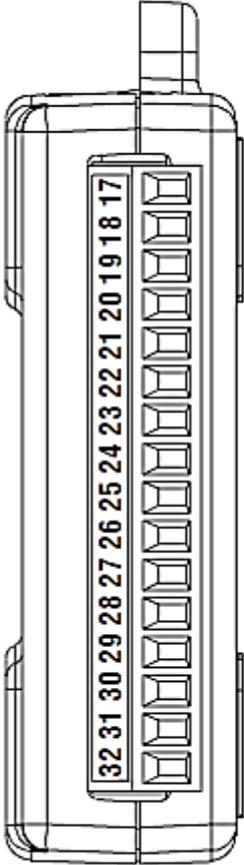
Module	Terminal	Signal
	17	P0.0
	18	P0.1
	19	P0.2
	20	P0.3
	21	P0.4
	22	P0.5
	23	P0.6
	24	P0.7
	25	P1.0
	26	P1.1
	27	P1.2
	28	P1.3
	29	PFI 0
	30	+2.5 V
	31	+5 V
	32	GND

Tabla 47: Terminales digitales NI USB-6009
Fuente: NI_6008/6009_ADC_manual.pdf, 2012

Se describe las señales evaluadas en los conectores I/O:

Signal Name	Reference	Direction	Description
GND	—	—	Ground —The reference point for the single-ended AI measurements, bias current return point for differential mode measurements, AO voltages, digital signals at the I/O connector, +5 VDC supply, and the +2.5 VDC reference.
AI <0..7>	Varies	Input	Analog Input Channels 0 to 7 —For single-ended measurements, each signal is an analog input voltage channel. For differential measurements, AI 0 and AI 4 are the positive and negative inputs of differential analog input channel 0. The following signal pairs also form differential input channels: <AI 1, AI 5>, <AI 2, AI 6>, and <AI 3, AI 7>.
AO 0	GND	Output	Analog Channel 0 Output —Supplies the voltage output of AO channel 0.
AO 1	GND	Output	Analog Channel 1 Output —Supplies the voltage output of AO channel 1.
P1.<0..3> P0.<0..7>	GND	Input or Output	Digital I/O Signals —You can individually configure each signal as an input or output.
+2.5 V	GND	Output	+2.5 V External Reference —Provides a reference for wrap-back testing.
+5 V	GND	Output	+5 V Power Source —Provides +5 V power up to 200 mA.
PFI 0	GND	Input	PFI 0 —This pin is configurable as either a digital trigger or an event counter input.

Tabla 48. Descripción de señales en la NI USB-6009
Fuente: NI_6008/6009_ADC_manual.pdf, 2012

CAPÍTULO III

DISEÑO, CONSTRUCCIÓN Y MONTAJE DE ENTRENADORES DIDÁCTICOS CON SISTEMA DE MICROPROCESADO Y APLICACIÓN DE CONTROL DE VELOCIDAD DE MOTOR

3.1. Introducción

En este capítulo se describe el diseño, esquema y circuitería electrónica de cada módulo, interfaces de salida, en general, así como la descripción básica de su funcionamiento y el diseño final que reúne a los módulos y las interfaces en soporte de chasis que da vida al entrenador didáctico. El cual servirá de gran ayuda para el desarrollo de un sin número de prácticas y proyectos a realizar en los laboratorio de electrónica o afines de la UPS.

3.2. Diagrama de bloques del entrenador didáctico

A continuación se describe detalladamente el entrenador didáctico mediante bloques:

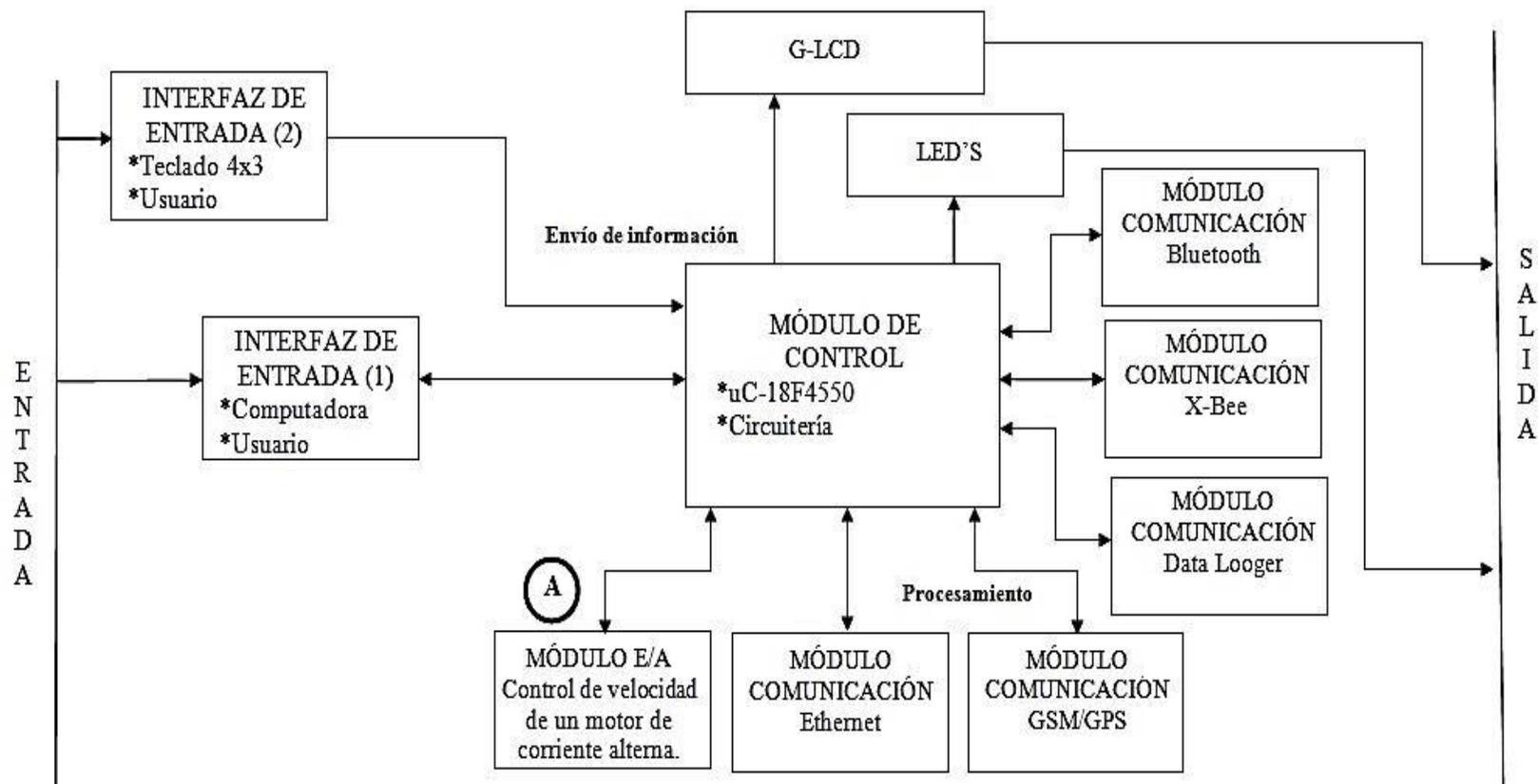


Figura 61: Diagrama de bloques principal del entrenador didáctico
Fuente: NI_6008/6009_ADC_manual.pdf, 2012

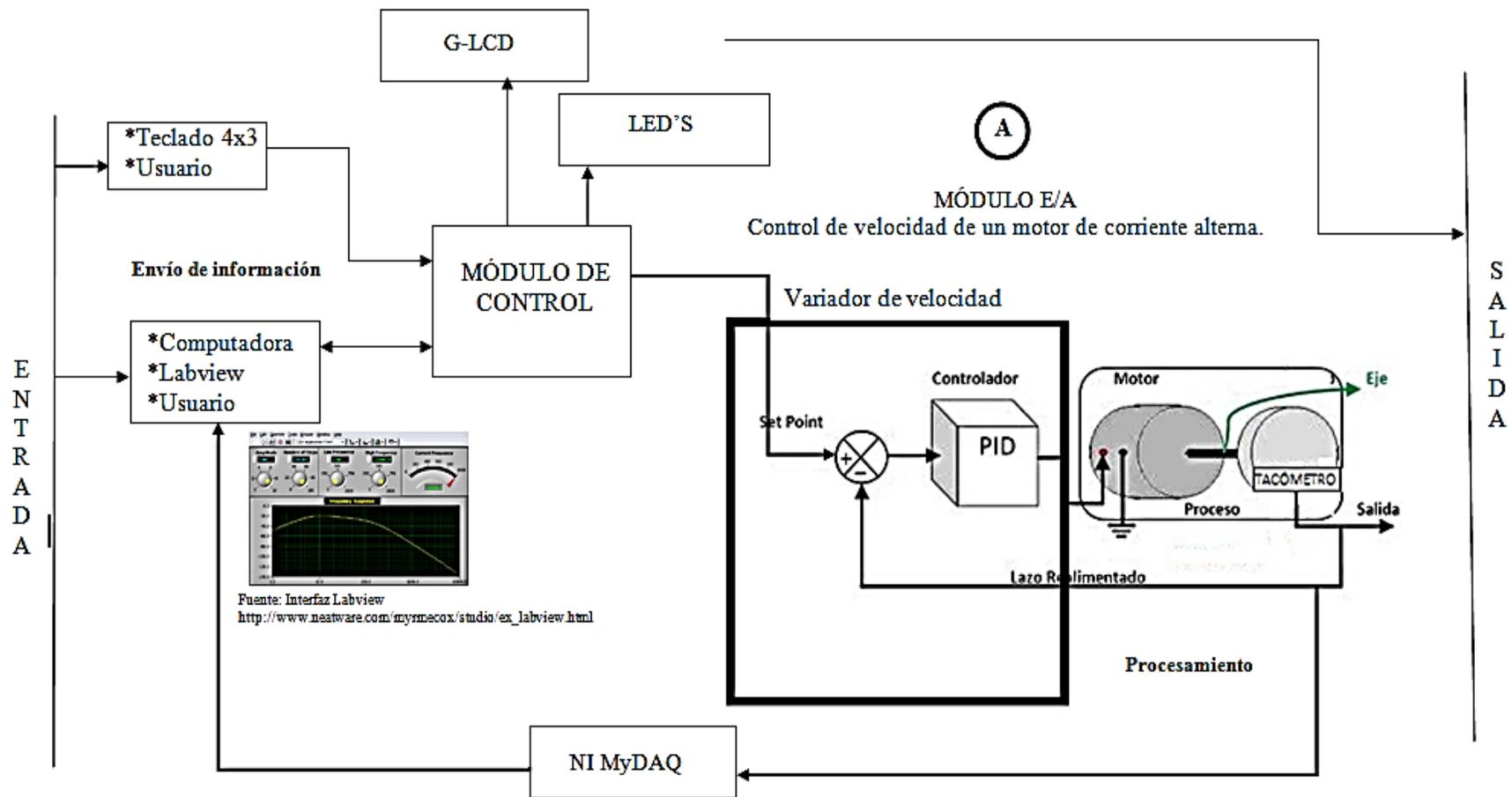


Figura 62: Diagrama de bloques enfoque módulo de control de velocidad del motor
Fuente: Autores, 2012

3.3. Diseño, esquema y circuitería electrónica de interfaz entrada.

La interfaz de entrada permite la comunicación de datos digitales de entrada al PIC por uno de sus puertos configurados previamente para tal uso

3.3.1. Diagrama esquemático en Proteus-ISIS

Se realiza un diagrama esquemático de la interfaz y sus componentes en el software Proteus-ISIS:

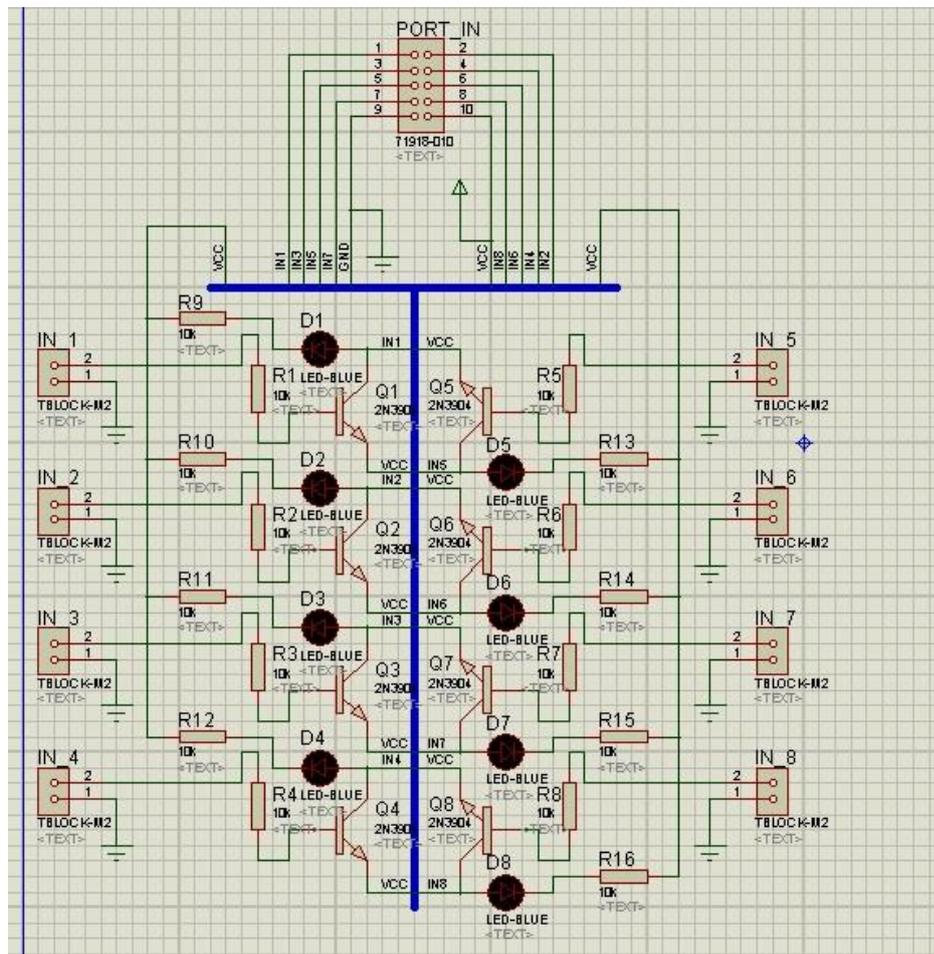


Figura 63: Diagrama Proteus ISIS interfaz de entrada
Fuente: Autores, 2013

Este diagrama está basado en la estructura de transistores NPN los cuales tienen como bondad de ser semiconductores, permitiendo el paso de corriente a través de sus terminales para de esta forma obtener un pulso en la entrada del microcontrolador.

Estos transistores tienen como característica principal 3 terminales los cuales son: emisor, base y colector, siendo el emisor el canal que porta la carga, y su base se encarga de activar mediante una resistencia de 4.7 k Ω conectada a una bornera para que mediante ella pueda cerrar el circuito.

El conector PORT_IN (IDC10 MACHO) se encargará de conectar la interfaz hacia el módulo de control principal y el mismo llegará hacia uno de los puertos del microcontrolador.

3.3.2. Descripción de elementos utilizados en la interfaz de entrada.

Esta interfaz de entrada está formada por los siguientes elementos:

INTERFAZ DE ENTRADA	ELEMENTOS
	Transistor 2N3904 (NPC)
	Tarjeta de circuito impreso (PCB)
	Diodo LED 3 mm (VERDE)
	Resistencia de 330 ohmios
	Resistencia de 4,7 k ohmios
	Bornera de dos polos
	Conector IDC-10 (Macho)

Tabla 48: Elementos interfaz de entrada

Fuente: Autores, 2012

3.3.3. Diseño del circuito impreso en Proteus-ARES

La siguiente figura muestra el diseño del circuito impreso de la interfaz de entrada, la cual tiene una característica de impresión en doble cara para optimizar el espacio y la funcionalidad en el entrenador didáctico.

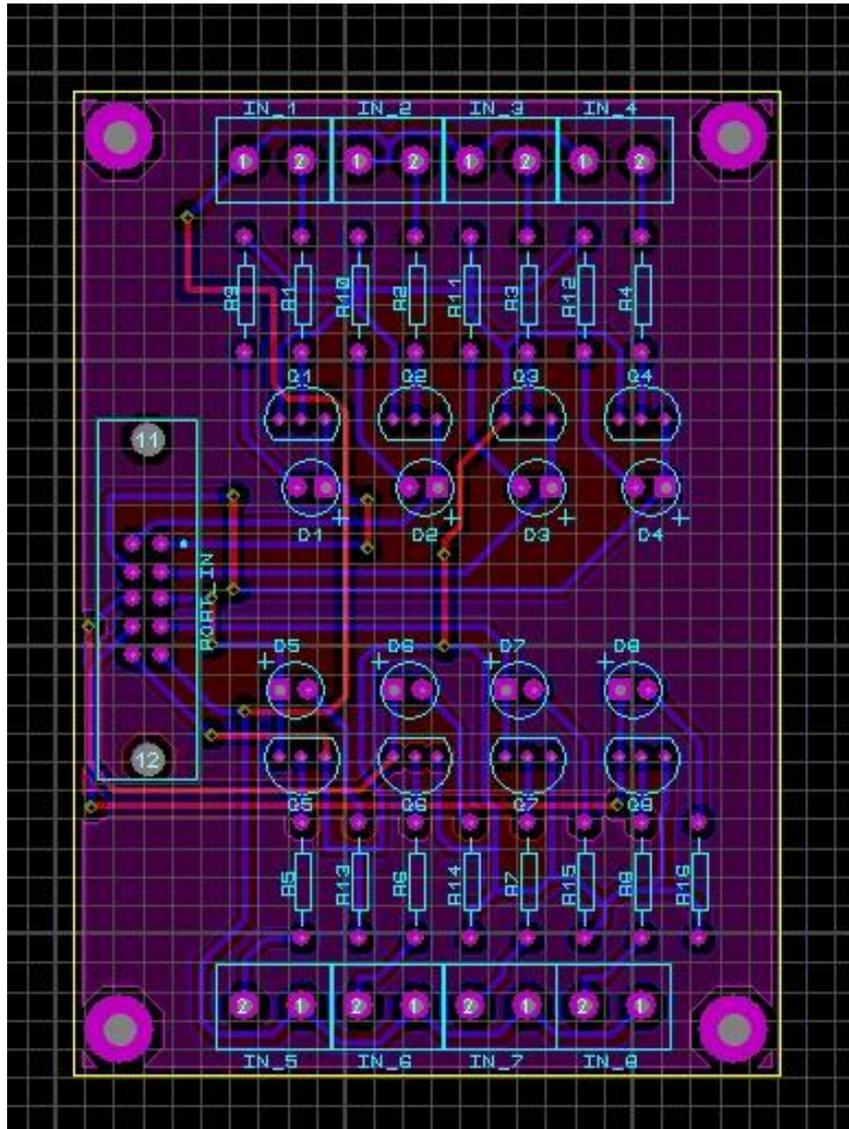


Figura 64: Diseño circuito impreso Proteus-ARES, interfaz de entrada
Fuente: Autores, 2014

3.3.4. Montaje de elementos electrónicos en la PCB.

Se realiza el montaje de los elementos en la PCB ya diseñada, así como la soldadura de cada uno de ellos.

Previamente según el diseño del circuito se obtiene el listado de los elementos a ser montados en la PCB y la correcta ubicación de los mismos, entre ellos la polarización de los diodos LED y los transistores 2N3904.

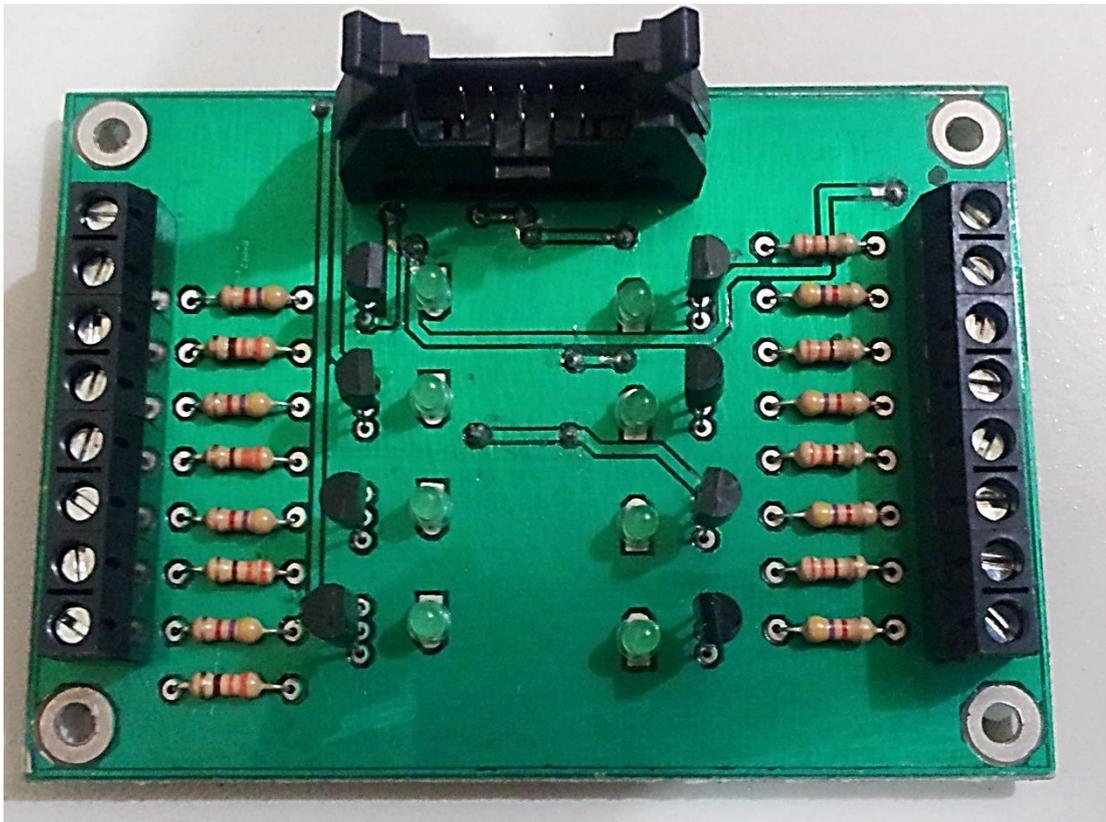


Figura 65: Tarjeta PCB y elementos electrónicos, interfaz de entrada
Fuente: Autores, 2014

3.4. Diseño, esquema y circuitería electrónica de interfaz salida.

Se describe el diseño de la interfaz de salida con cada uno de sus componentes:

3.4.1. Descripción de elementos utilizados en la interfaz de salida

Lista de elementos utilizados en el diseño y montaje de la PCB

Módulo Interface de salida	ELEMENTOS
	Relay DC5V (5 pines)
	Transistor 2N3904
	Tarjeta de circuito impreso (PCB)
	Diodo LED 3 mm (AMARILLO)
	Resistencia de 330 ohmios
	Resistencia de 4,7 k ohmios
	Diodo Rectificador 1N4007
	Bornera de tres polos
	Conector IDC-10 (Macho)

Tabla 50: Elementos interfaz de salida

Fuente: Autores, 2014

3.4.2. Diagrama esquemático en Proteus-ISIS

En la interfaz de salida se utiliza los transistores NPN para de manera inversa cerrar el circuito desde el microcontrolador hacia los relays DC5V (5 pines).

Los relays permiten manejar tensiones y corrientes superiores a las del circuito permitiendo así activar desde interruptores crepusculares hasta motores eléctricos dependiendo de su intensidad máxima de trabajo.

En la siguiente figura se bosqueja el diseño electrónico hecho en Proteus, entorno ISIS:

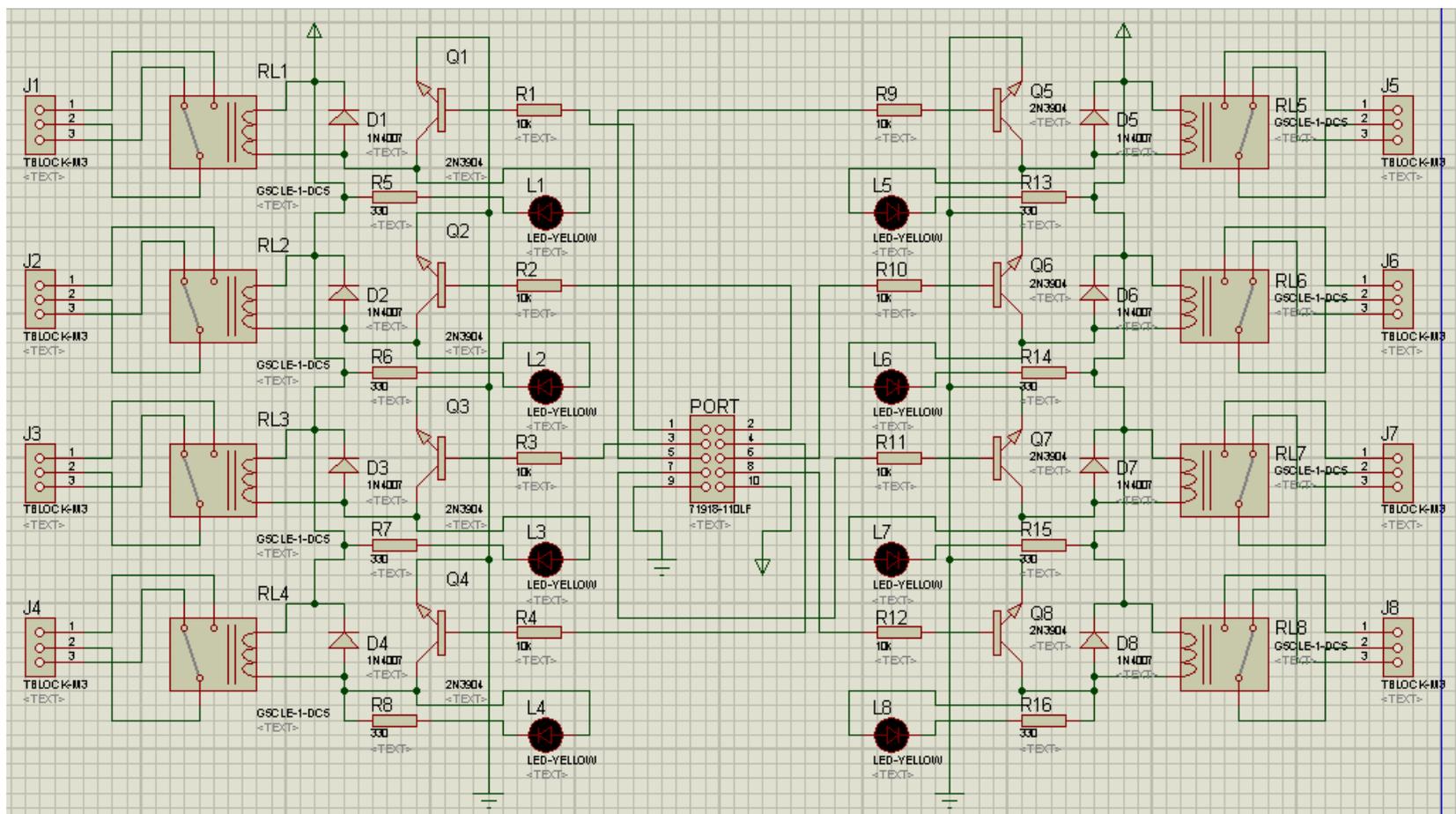


Figura 66: Diseño electrónico Proteus ISIS, interfaz de salida
Fuente: Autores, 2013

El diodo regula el flujo de la corriente en el circuito hacia la bobina del relay para que el mismo pueda ser activado mediante una corriente de excitación y este a su vez cambie de un estado normalmente abierto a cerrado, o de un estado normalmente cerrado a abierto.

En este caso está diseñado este circuito con un relay que posee ambos estados con la finalidad de que sea más flexible en un uso futuro.

3.4.3. Diseño del circuito impreso en Proteus-ARES

Con ayuda del módulo de ARES en Proteus se procede a realizar el diseño para la elaboración de la PCB.

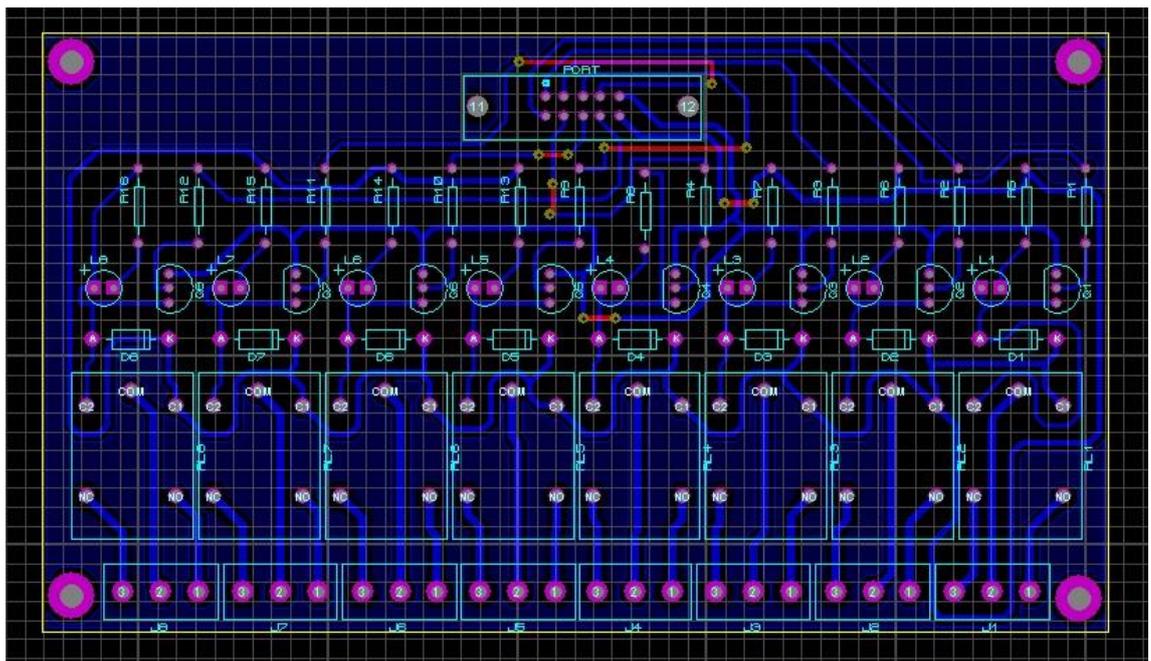


Figura 67: Circuito impreso Proteus ARES, interfaz de salida
Fuente: Autores, 2014

Esta tarjeta está diseñada en una sola cara con la particularidad de sus 8 salidas transistorizadas y al mismo tiempo aisladas por los relays electromecánicos de 10 amperios, dejando para casa uno de ellos su bornera de 3 polos para sus pines de normalmente cerrado, común y normalmente abierto.

3.4.4. Montaje de elementos electrónicos en la PCB.

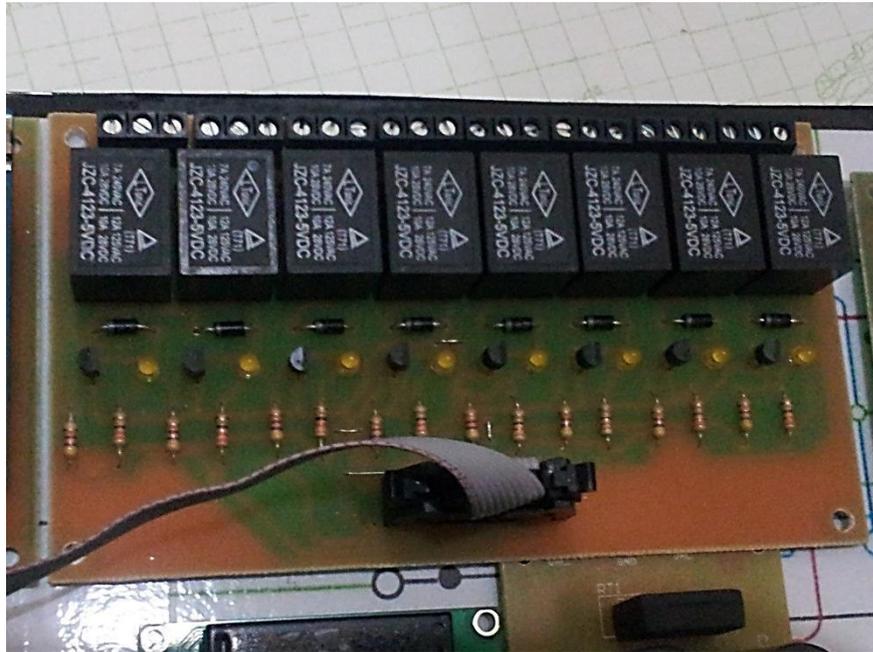


Figura 68: Tarjeta PCB y elementos electrónicos, interfaz de salida
Fuente: Autores, 2014

Se obtiene el listado de los elementos a ser montados en la PCB y la correcta ubicación de los mismos, principalmente de los diodos rectificadores de corriente 1N4007, los diodos LED y los transistores 2N3904.

Los relays DC5V (5 pines) de la PCB deberán estar ubicados de tal forma que correspondan el B1 y el B2 como las terminales que activaran la bobina con voltaje de 5V suministrado por el transistor 2N3904 y rectificado por el diodo 1N4007.

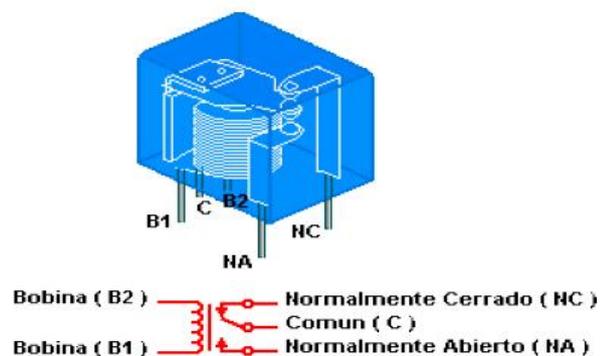


Figura 69: Relay DC5V, diagrama esquemático
Fuente: Relé, <http://es.wikipedia.org/wiki/Relé>, 2014

Así mismo las terminales de normalmente abierto (NA), común (C) y normalmente cerrado (NC) según corresponda.

3.5. Diseño, esquema y circuitería electrónica del módulo de control

Es el turno de describir el módulo principal del entrenador didáctico, es una de las tarjetas que requiere mayor énfasis en su diseño y elaboración ya que este módulo es el corazón del entrenador didáctico donde va alojado el PIC (18F4550) programado previamente para cada práctica que se desee realizar.

3.5.1. Diagrama esquemático en Proteus-ISIS

En el módulo de control tenemos como principal elemento el microcontrolador PIC18F4550, el mismo que fue escogido para la elaboración de este proyecto de tesis por su versatilidad, su gran espacio de memoria interna; por ejemplo su bus de memoria de programa distribuido en 21 líneas de dirección, 16 líneas para instrucciones y 8 para datos y su bus de memoria de datos constituido de 12 líneas de dirección y 8 líneas de datos.

Además el módulo cuenta con el oscilador/cristal de cuarzo de 20Mhz, el mismo que le proporcionará al microcontrolador una señal de ciclo de reloj.

Los Dip Swich permiten habilitar o deshabilitar los puertos del microcontrolador con la finalidad de evitar conflictos en las interfaces a utilizarse.

Se adjunta al diseño un Zip Socket que permite la manipulación del microcontrolador para su fácil montaje y desmontaje lo cual alarga su vida útil.

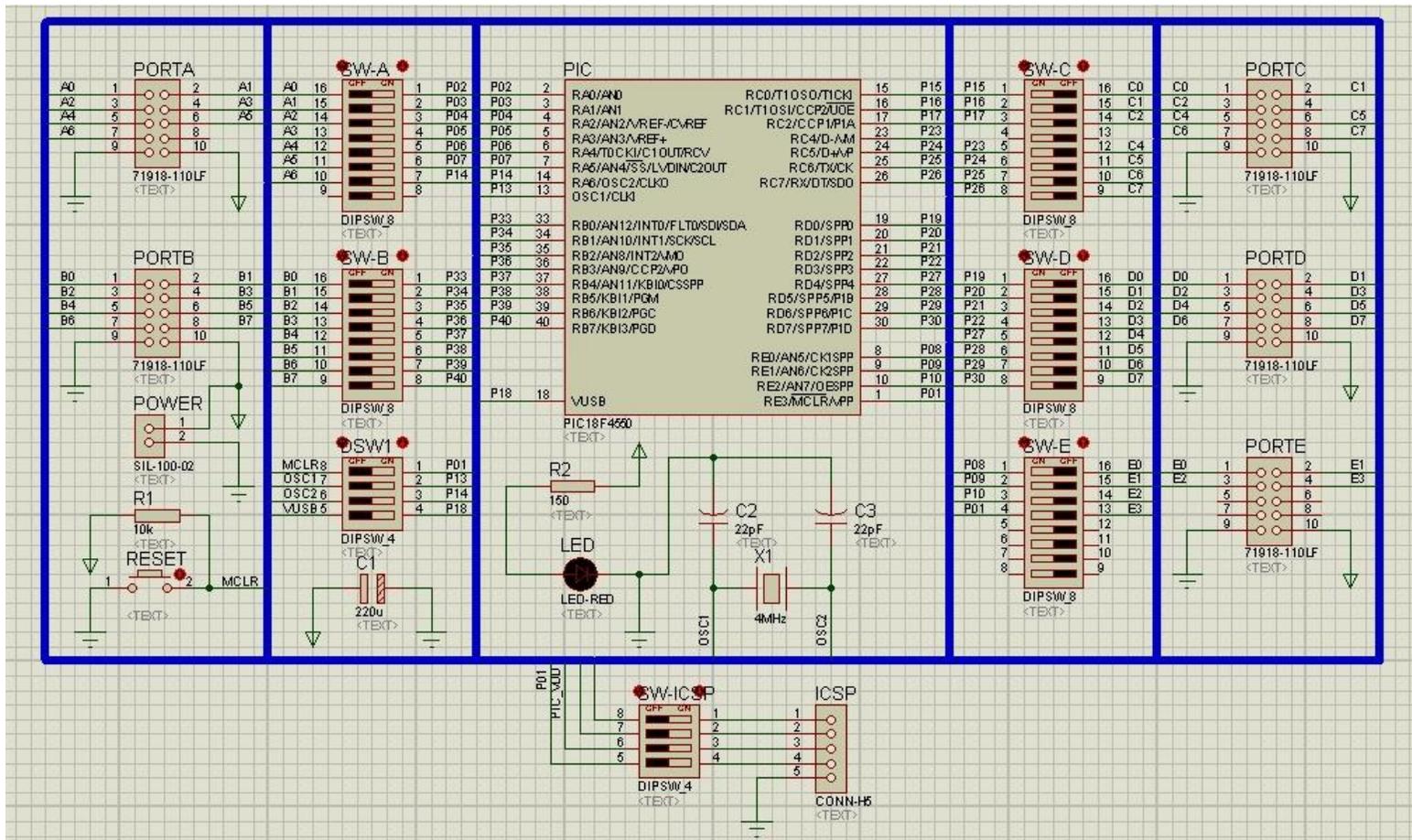


Figura 70: Diagrama esquemático Proteus ISIS, módulo de control
Fuente: Autores, 2014

3.5.2. Descripción de elementos utilizados en el módulo de control.

Módulo Principal o Control	ELEMENTOS
	Microcontrolador PIC 18F4550
	Dip Swicth 8 pines
	Dip Swicth 4 pines
	Cristal de Cuarzo 20Mhz
	Capacitor Electrolítico 1000 uf / 16v
	Tarjeta de circuito impreso (PCB)
	Conector IDC-10 (Macho Simple)
	Conector Molex 1x2
	Resistencia de 330 ohmios
	Resistencia de 10 k ohmios
	Zip Socket
	regleta de 45 pines Hembra
	Diodo LED 3 mm (ROJO)
	Pulsante

Tabla 51: Elementos de módulo de control
Fuente: Autores, 2014

3.5.3. Diseño del circuito impreso en Proteus-ARES

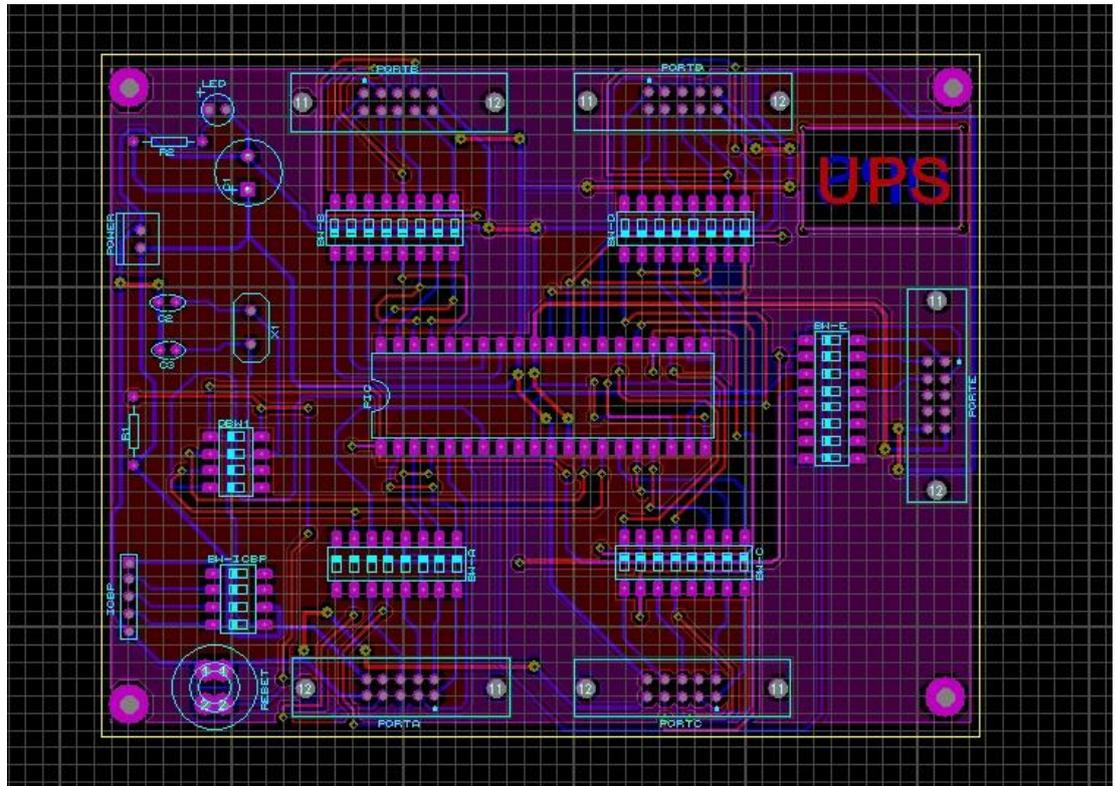


Figura 71: Diagrama de circuito impreso Proteus ARES, módulo de control
Fuente: Autores, 2014

El diseño de la tarjeta PCB tiene una impresión de doble cara que permite la optimización del espacio y la distribución ordenada de los elementos electrónicos.

3.5.4. Montaje de elementos electrónicos en la PCB.

Placa PCB con el resto de elementos ensamblados según diseño:

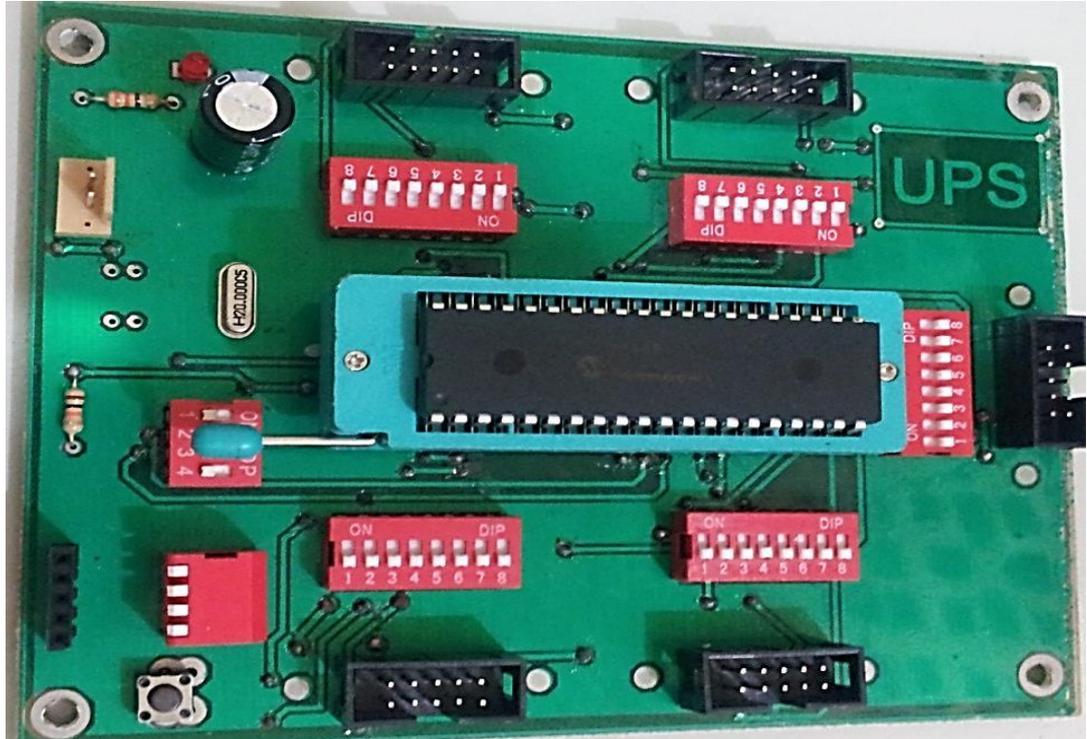


Figura 72: Tarjeta PCB y elementos electrónicos ensamblados, módulo de control
Fuente: Autores, 2014

Como se muestra en la figura adicional a los elementos descritos inciso 3.6.1. el módulo principal consta de un capacitor electrolítico 1000 uf / 16v el mismo que mantiene un nivel equilibrado de voltaje para el circuito evitando la caída de tensión.

Este módulo principal fue diseñado con un conector de 5 pines hembra que permite la conexión directa con un programador de PIC por el puerto ICSP, de igual forma se encuentra swicheado por el Dip Switch de 4 pines por lado (SW-ICSP) hacia el microcontrolador mediante un Dip Switch de 4 pines por lado.

Los conectores IDC10 MACHO permiten la conexión directa hacia los diferentes puertos mediante los Dip Switch de 8 pines por lado haciendo versátiles la utilización de los mismos para ser utilizados según se disponga.

El botón MASTER CLEAR con que consta el módulo principal permite el reset del microcontrolador, teniendo en cuenta su utilización mediante el Dip Switch de 4 pines por lado (DSW1, pin 1).

3.6. Diseño, esquema y circuitería electrónica del módulo Bluetooth.

3.6.1. Diagrama esquemático en Proteus-ISIS

Este diagrama es uno de los más simples, consta de una resistencia de 300Ω , el bluetooth, led y un conector IDC-10.

El pin 9 del conector de microcontrolador con el pin 2 del conector bluetooth conectados a GND y el pin 10 del micro en conjunto con el pin 1 del conector bluetooth a +5Vcc por estándar establecido, el pin 7 al Tx del microcontrolador hacia el Rx del conector bluetooth que es el pin 4.

El pin 3 que es el Tx del conector del bluetooth hacia el Rx del conector del microcontrolador que es el pin 8.

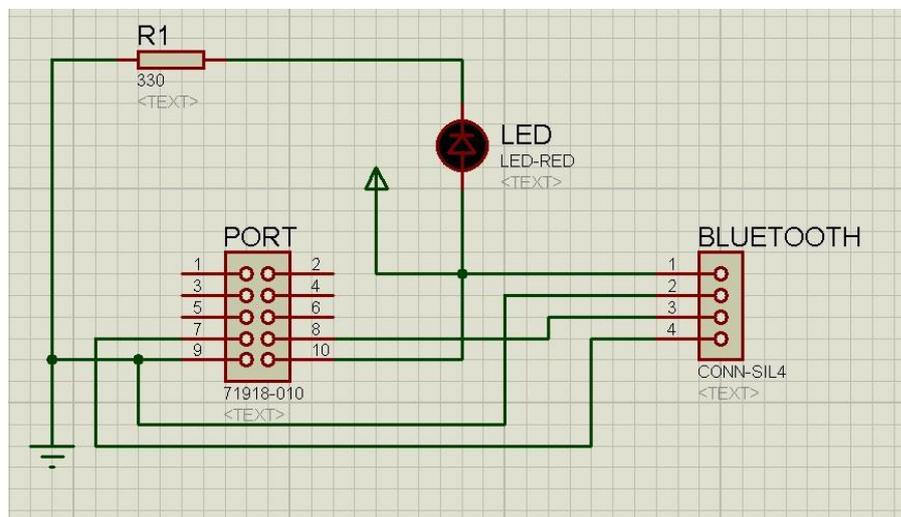


Figura 73: Diagrama esquemático Proteus ISIS, módulo Bluetooth
Fuente: Autores, 2014

3.6.2. Descripción de elementos utilizados en el módulo Bluetooth.

Módulo BLUETOOTH	ELEMENTOS
	Bluetooth DF
	Tarjeta de circuito impreso (PCB)
	Diodo LED 3 mm (ROJO)
	Resistencia de 330 ohmios
	Conector IDC-10 (Macho)

Tabla 52: Elementos de módulo Bluetooth
Fuente: Autores, 2014

3.6.3. Diseño del circuito impreso en Proteus-ARES

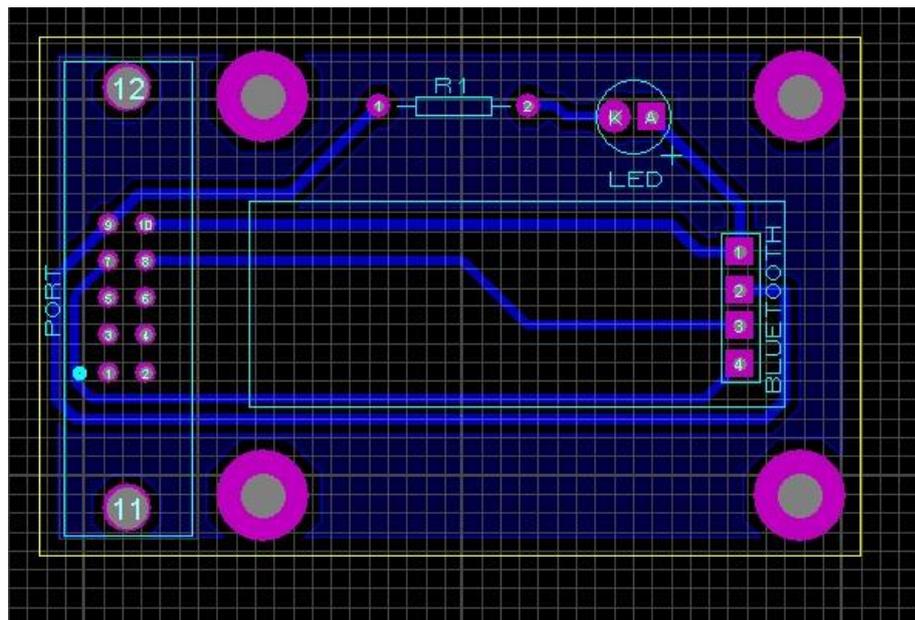


Figura 74: Diagrama de circuito impreso Proteus ARES, módulo Bluetooth
Fuente: Autores, 2014

El circuito impreso está diseñado en una sola cara debido a que no hay mucha complejidad a la hora de realizar las pistas.

3.6.4. Montaje de elementos electrónicos en la PCB.

Se procede a ensamblar la tarjeta PCB con cada uno de sus elementos.

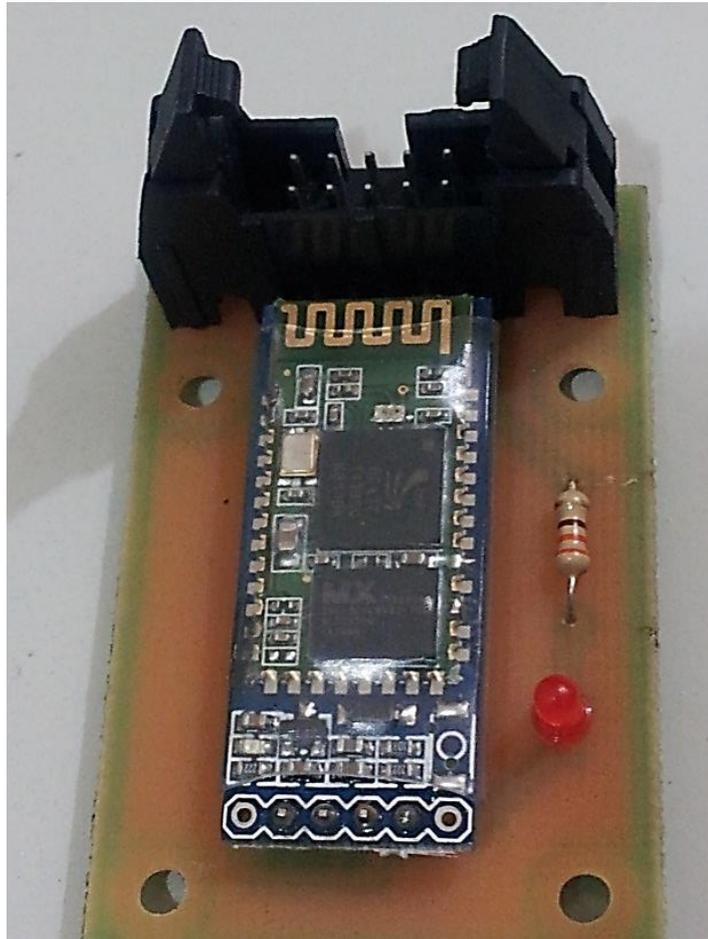


Figura 75: Tarjeta PCB y elementos electrónicos ensamblados, módulo Bluetooth
Fuente: Autores, 2014

El módulo consta del conector IDC10 que permite la comunicación con el módulo principal para la realización de prácticas, también consta de un diodo LED indicador de fuente de poder activo.

3.7. Diseño, esquema y circuitería electrónica del módulo Xbee.

El módulo Xbee nos permitirá desarrollar numerosas prácticas interesantes haciendo referencia a la gran utilidad que tiene el Xbee en el campo inalámbrico.

3.7.1. Diagrama esquemático en Proteus-ISIS

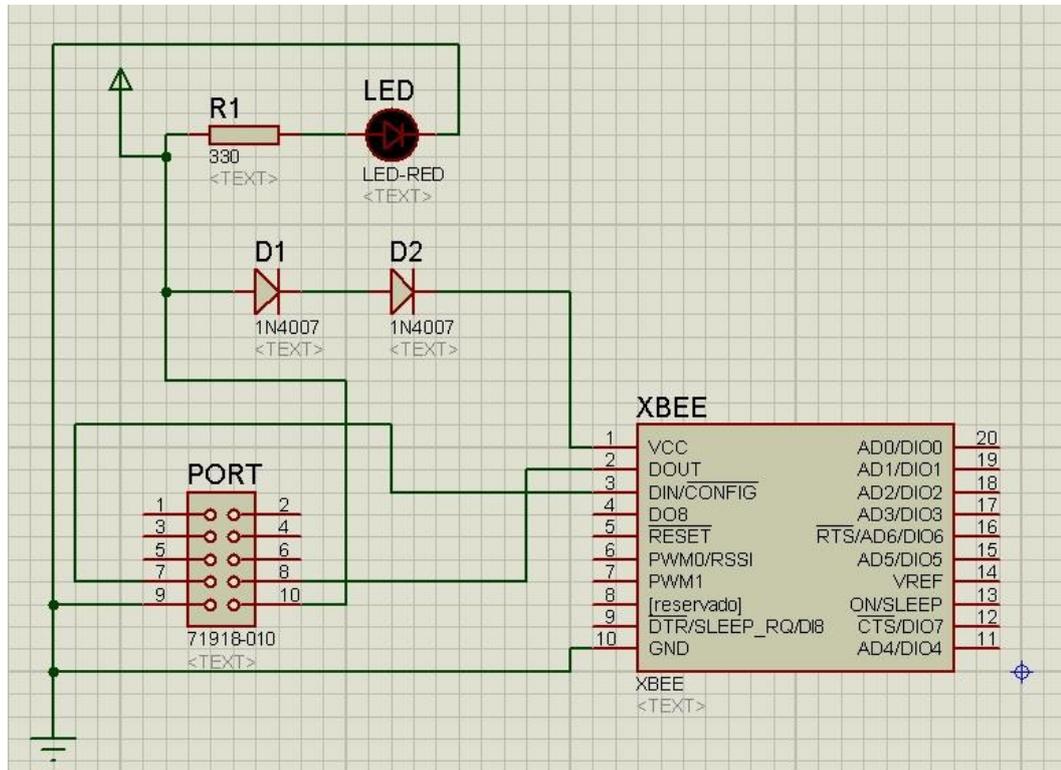


Figura 76: Diagrama esquemático Proteus ISIS, módulo Xbee

Fuente: Autores, 2014

Diseñado con un LED indicador de voltaje en el registro, los D1 y D2 son diodos rectificadores puestos en serie para hacer caer voltaje, ya que el dispositivo Xbee trabaja con 3.3v aproximadamente y el voltaje inicial suministrado es de +5Vcc.

3.7.2. Descripción de elementos utilizados en el módulo Xbee.

Módulo XBEE	ELEMENTOS
	Xbee 1MW Wire Ant
	Tarjeta de circuito impreso (PCB)
	Diodo LED 3 mm (ROJO)
	Diodo Rectificador 1N4007
	Resistencia de 330 ohmios
	Conector IDC-10 (Macho)

Tabla 53: Elementos de módulo Xbee

Fuente: Autores, 2014

3.7.3. Diseño del circuito impreso en Proteus-ARES

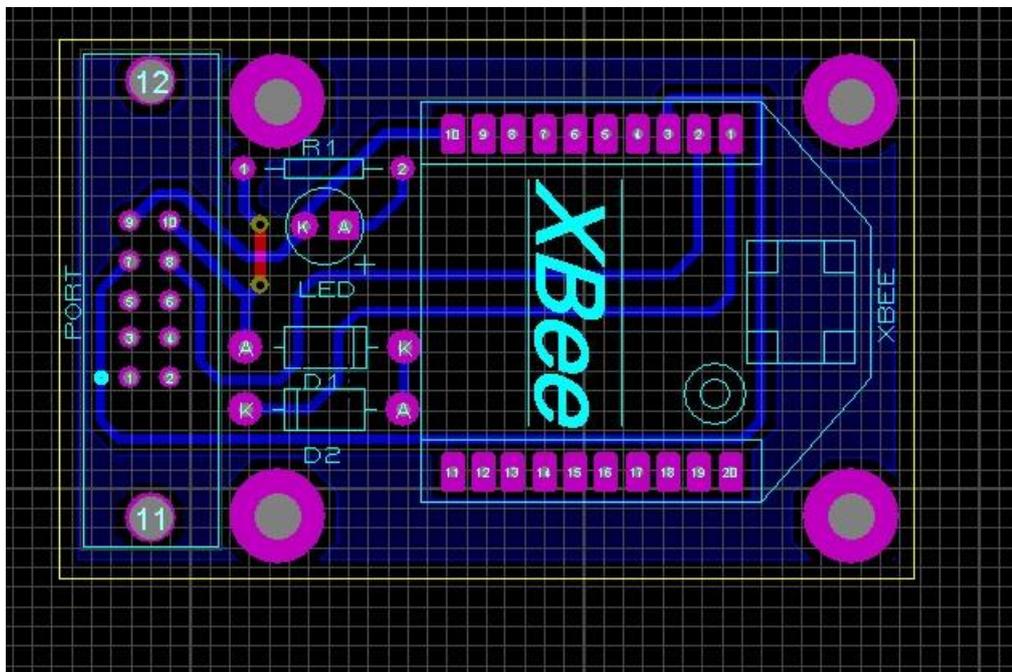


Figura 77: Diseño circuito impreso Proteus ARES, módulo Xbee

Fuente: Autores, 2014

Este circuito está impreso en una sola cara ya que las conexiones son sencillas y fáciles de rootear, así como el montaje de sus elementos electrónicos.

3.7.4. Montaje de elementos electrónicos en la PCB.

Figura con la tarjeta PCB y el montaje de los elementos mencionados anteriormente.

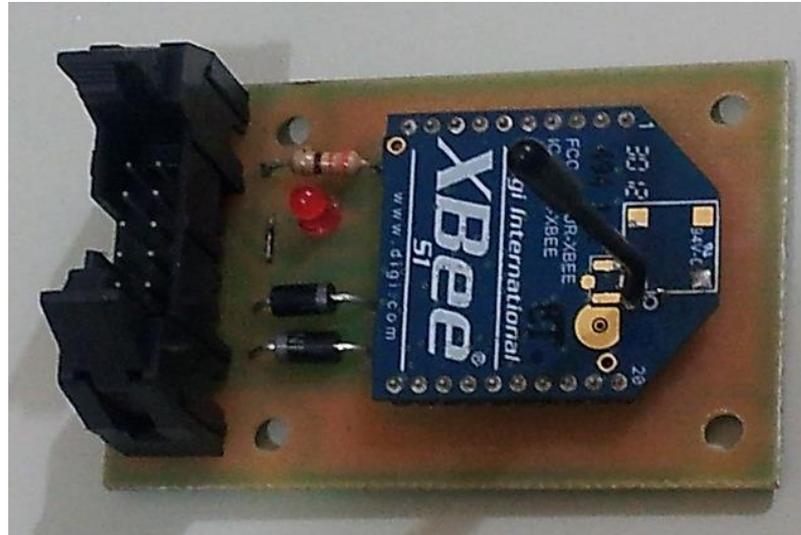


Figura 78: Tarjeta PCB y elementos electrónicos, módulo Xbee
Fuente: Autores, 2014

3.8. Diseño, esquema y circuitería electrónica del módulo Datalogger.

Para la elaboración de este diseño en ISIS hubo que crear el elemento DATALOGGER tal como se muestra en la figura del diagrama esquemático.

3.8.1. Diagrama esquemático en Proteus-ISIS

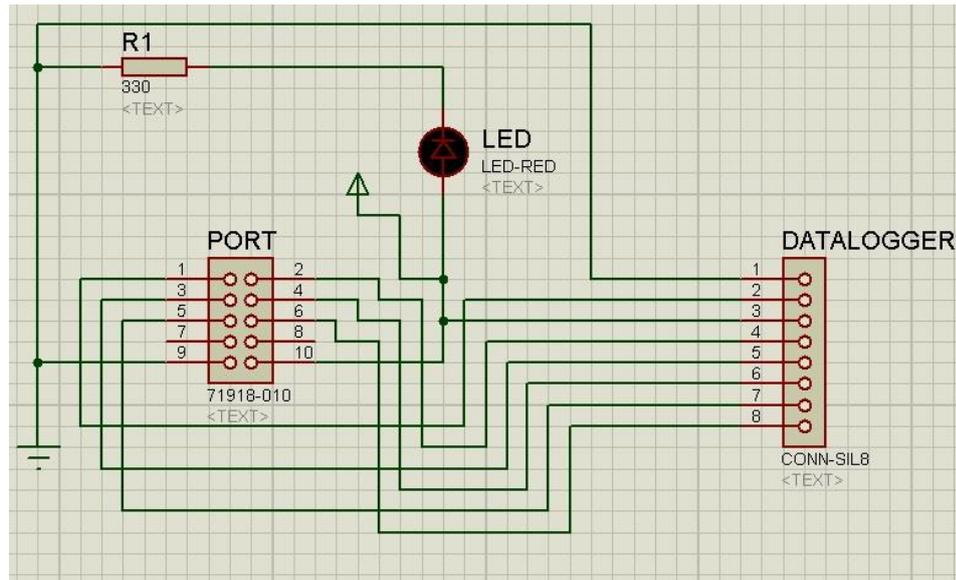


Figura 79: Diseño esquemático Proteus ISIS, módulo Datalogger
Fuente: Autores, 2014

Los pines 9 a GND y el 10 a +5Vcc del conector PORT, de igual forma los pines 1 y 3 del conector DATALOGGER respectivamente.

- PORT, Pin1 → DATALOGGER, Pin 2 (RTS)
- PORT, Pin2 → DATALOGGER, Pin4 (RXD)
- PORT, Pin3 → DATALOGGER, Pin5 (TXD)
- PORT, Pin4 → DATALOGGER, Pin6 (CTS)
- PORT, Pin5 → DATALOGGER, Pin7 (NC)
- PORT, Pin6 → DATALOGGER, Pin8 (RI)

3.8.2. Descripción de elementos utilizados en el módulo Datalogger.

Módulo DATA LOGGER	ELEMENTOS
	Memory Stick Datalogger
	Tarjeta de circuito impreso (PCB)
	Diodo LED 3 mm (ROJO)
	Resistencia de 330 ohmios
	Conector IDC-10 (Macho)

Tabla 53: Elementos de módulo Datalogger
Fuente: Autores, 2014

3.8.3. Diseño del circuito impreso en Proteus-ARES

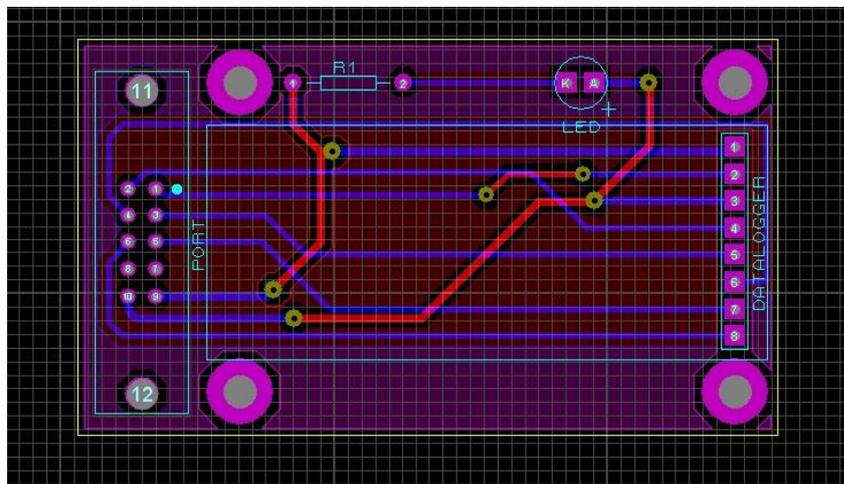


Figura 80: Diseño de circuito impreso Proteus ARES, módulo Datalogger
Fuente: Autores, 2014

Este diseño está impreso en doble cara debido a un mayor número de pistas trazadas que se encuentran en una misma dirección.

3.8.4. Montaje de elementos electrónicos en la PCB.

Aquí se muestra el módulo con el Datalogger y sus elementos ensamblados, el montaje de estos elementos es muy sencillo y poco complejo.

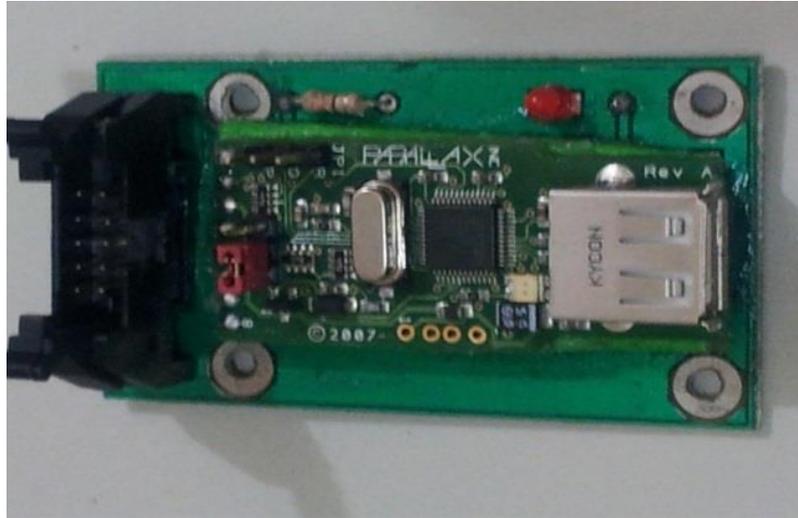


Figura 81: Diseño de circuito impreso Proteus ARES, módulo Datalogger
Fuente: Autores, 2014

3.9. Diseño, esquema y circuitería electrónica del módulo Ethernet.

Descripción del diseño y construcción del módulo Ethernet.

3.9.1. Diagrama esquemático en Proteus-ISIS

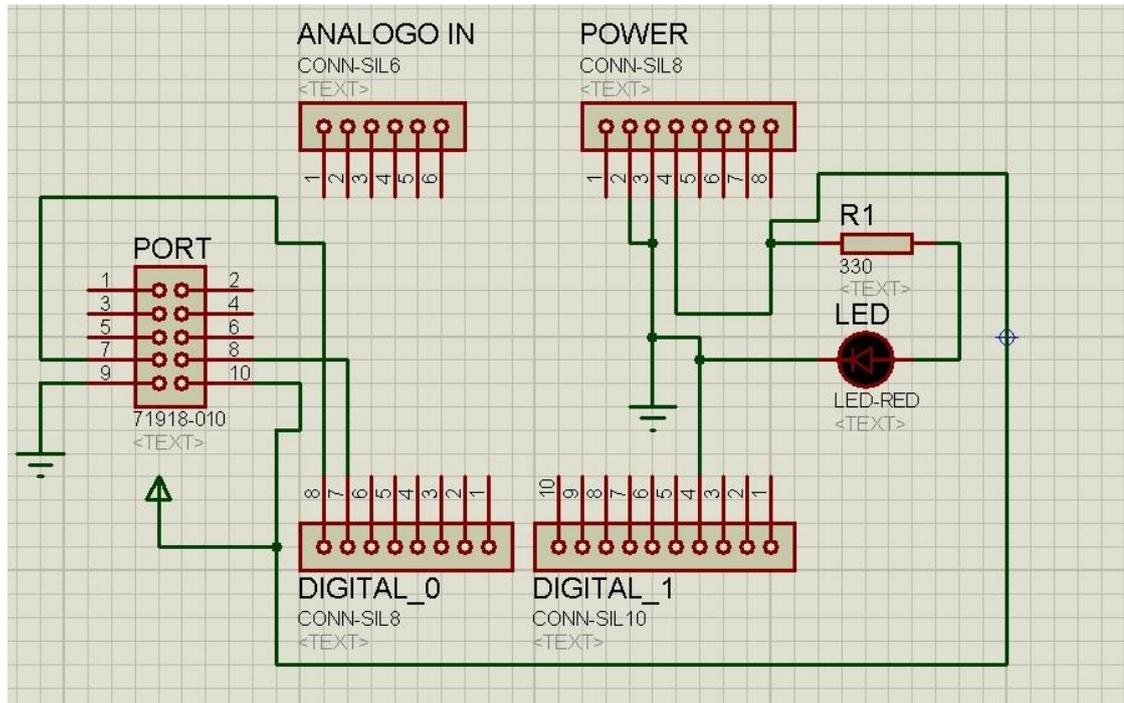


Figura 82: Diseño esquemático Proteus ISIS, módulo Ethernet
Fuente: Autores, 2014

En el diagrama realizado en ISIS se construyó esquemáticamente el módulo Ethernet tomando en cuenta los pines necesarios para anclar el dispositivo Arduino Ethernet.

- PORT, Pin7 → Arduino Ethernet: DIGITAL_0, Pin 8 (RX)
- PORT, Pin8 → Arduino Ethernet: DIGITAL_0, Pin 7 (TX)
- PORT, Pin9(GND) → Arduino Ethernet: POWER, Pin 2 y 3(GND)
DIGITAL_1, Pin 4(GND)
- PORT, Pin10(+5Vcc) → Arduino Ethernet: POWER, Pin 4 (+5Vcc)

3.9.2. Descripción de elementos utilizados en el módulo Ethernet.

Módulo ARDUINO ETHERNET	ELEMENTOS
	ARDUINO ETHERNET
	Tarjeta de circuito impreso (PCB)
	Diodo LED 3 mm (ROJO)
	Resistencia de 330 ohmios
	Conector IDC-10 (Macho)

Tabla 55: Elementos de módulo Ethernet

Fuente: Autores, 2014

3.9.3. Diseño del circuito impreso en Proteus-ARES

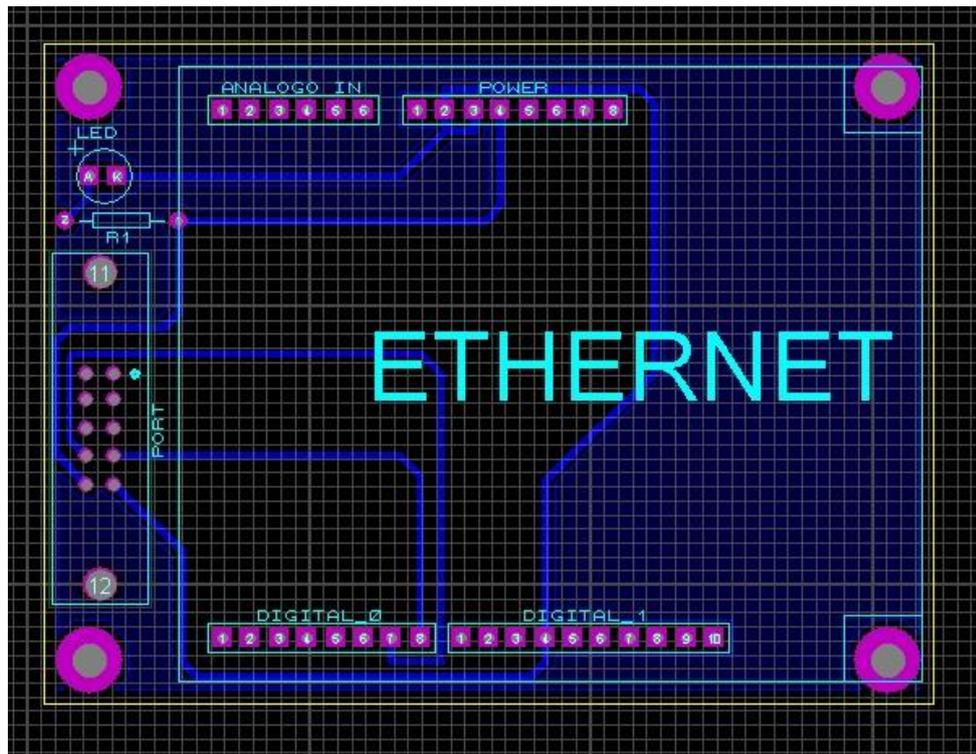


Figura 83: Diseño del circuito impreso Proteus ARES, módulo Ethernet

Fuente: Autores, 2014

Está diseñado en una cara simple debido a que las pistas trazadas son poco complejas, fácil de diseñar.

3.9.4. Montaje de elementos electrónicos en la PCB.

Tarjeta PCB y montaje de elementos.

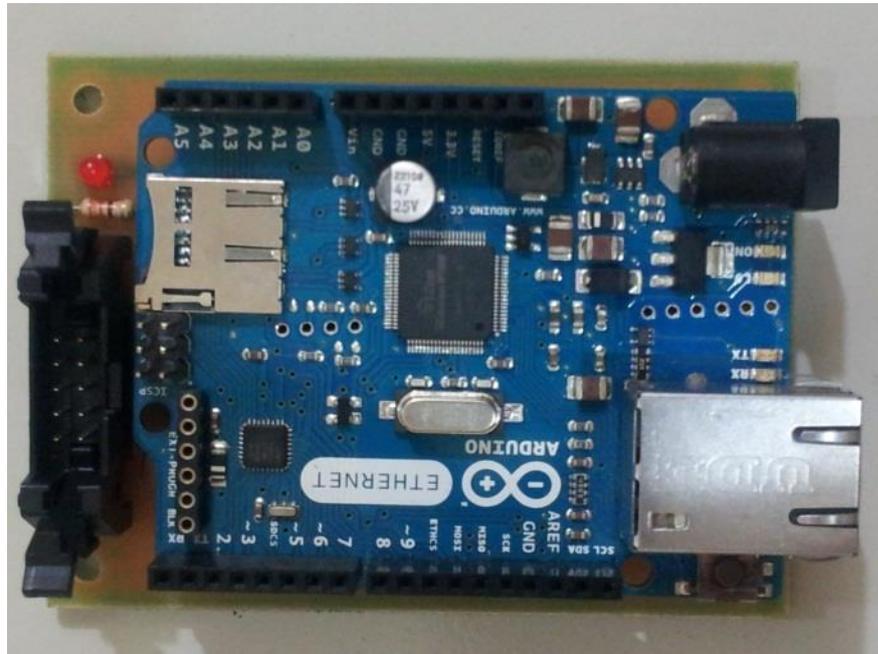


Figura 84: Tarjeta PCB y elementos electrónicos ensamblados, módulo Ethernet
Fuente: Autores, 2014

3.10. Diseño, esquema y circuitería electrónica del módulo GSM/GPRS/GPS.

Uno de los módulos más interesantes es el que contiene el Arduino GSM/GPRS/GPS, ya que se puede realizar un sin número de prácticas de gran utilidad.

3.10.1. Diagrama esquemático en Proteus-ISIS

En este diseño al igual que en anteriores también se realizó el esquemático del Arduino GSM/GPRS/GPS.

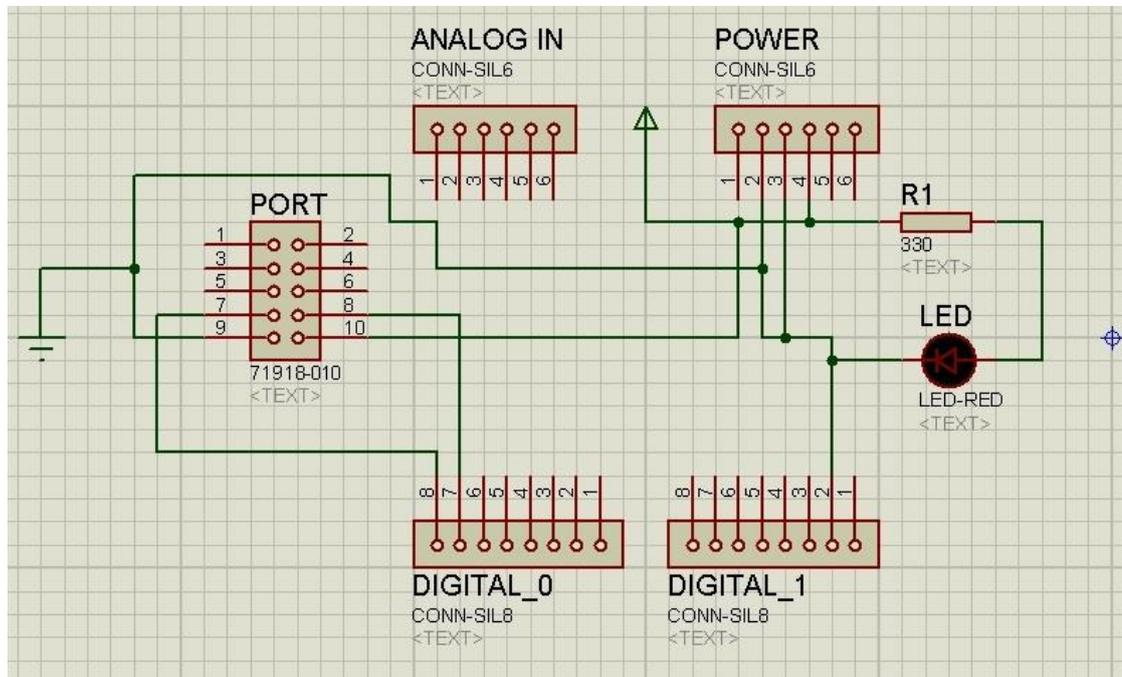


Figura 85: Diseño esquemático Proteus ISIS, módulo GSM/GPRS/GPS
Fuente: Autores, 2014

Las conexiones del módulo diseñado se describen de la siguiente manera:

- PORT, Pin7 → Arduino GSM/GPRS/GPS: DIGITAL_0, Pin 8 (RX)
- PORT, Pin8 → Arduino GSM/GPRS/GPS: DIGITAL_0, Pin 7 (TX)
- PORT, Pin9(GND) → GSM/GPRS/GPS: POWER, Pin 2 y 3(GND)
DIGITAL_1, Pin 2(GND)
- PORT, Pin10(+5Vcc) → GSM/GPRS/GPS: POWER, Pin 4 (+5Vcc)

3.10.2. Descripción de elementos utilizados en el módulo GSM/GPRS/GPS.

Módulo GSM - GPRS - GPS	ELEMENTOS
	GSM - GPRS - GPS
	Tarjeta de circuito impreso (PCB)
	Diodo LED 3 mm (ROJO)
	Resistencia de 330 ohmios
	Conector IDC-10 (Macho)

Tabla 56: Elementos de módulo GSM/GPRS/GPS
Fuente: Autores, 2014

3.10.3. Diseño del circuito impreso en Proteus-ARES

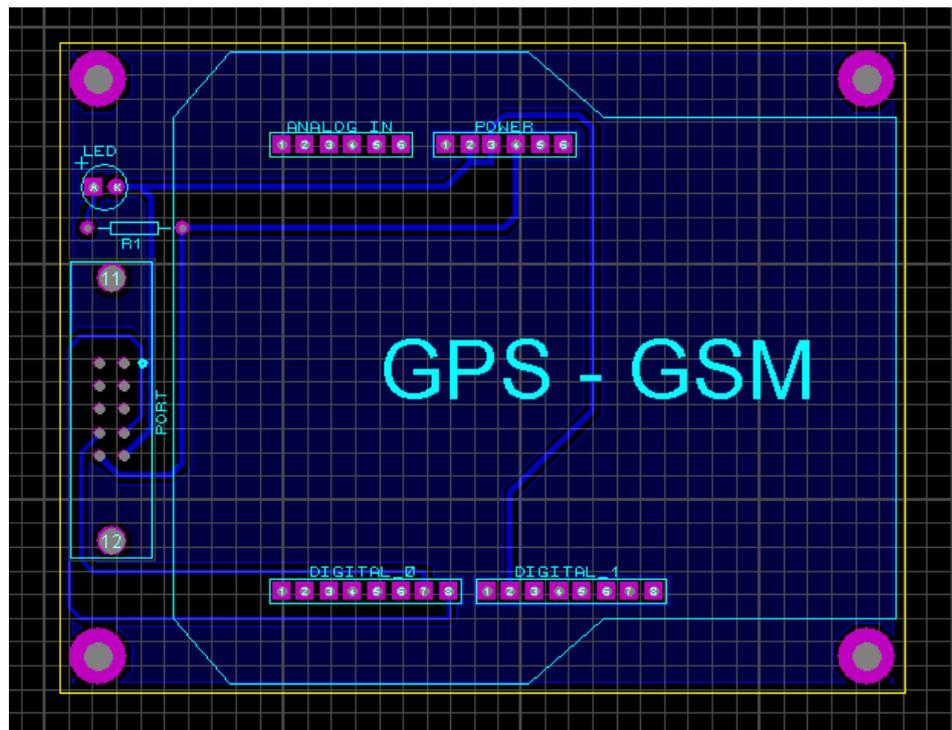


Figura 86: Diseño de circuito impreso Proteus ARES, módulo GSM/GPRS/GPS
Fuente: Autores, 2014

Al igual que el diseño del módulo anterior esta tarjeta está impresa en una sola cara ya que su implementación es simplificada.

3.10.4. Montaje de elementos electrónicos en la PCB.



Figura 87: Tarjeta PCB e instalación de elementos, módulo GSM/GPRS/GPS
Fuente: Autores, 2014

En esta figura se muestra el dispositivo Arduino GSM/GPRS/GPS montado en la PCB junto al resto de elementos electrónicos para así formar el módulo propiamente dicho

3.11. Diseño, esquema y circuitería electrónica del módulo de control de velocidad de corriente alterna.

3.11.1. Diagrama esquemático en Proteus-ISIS

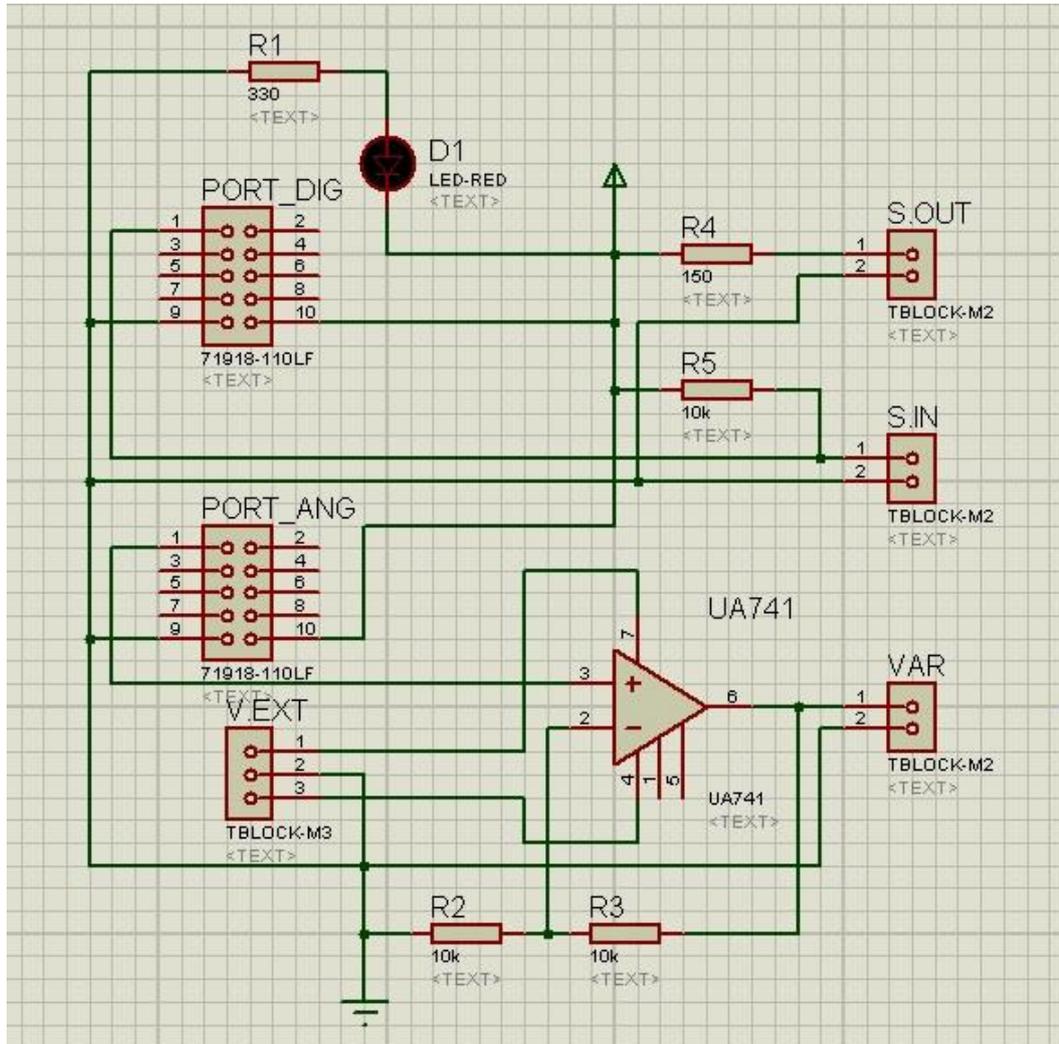


Figura 88: Diseño esquemático Proteus ISIS, módulo de control de velocidad de motor
Fuente: Autores, 2014

Este circuito está diseñado para interactuar con el variador de velocidad Altivar 312 para de esta forma controlar el motor de corriente alterna trifásica y obtener los rangos de velocidad.

Cuenta con un integrado UA741 el mismo que se conectará a una fuente externa (V.EXT) de $\pm 12V_{cc}$ para producir el rango de variación de voltaje de entre 0V a +10V para la entrada remota del variador de velocidad Altivar312.

En la bornera S.OUT se conectará el diodo infrarrojo para que sea censado mediante un fototransistor infrarrojo hacia la bornera S.IN conectada a un sensor infrarrojo para así obtener las revoluciones por minutos del motor de corriente alterno trifásico.

3.11.2. Descripción de elementos utilizados en el módulo de control de velocidad de corriente alterna.

Módulo Interface para Motor	ELEMENTOS
	Integrado UA741
	Socket 2 x 4
	Tarjeta de circuito impreso (PCB)
	Diodo LED 3 mm (ROJO)
	Resistencia de 330 ohmios
	Resistencia de 10 k ohmios
	Bornera de dos polos
	Bornera de tres polos
	Conector IDC-10 (Macho)

Tabla 57: Elementos de módulo de control de velocidad de motor
Fuente: Autores, 2014

3.11.3. Diseño del circuito impreso en Proteus-ARES

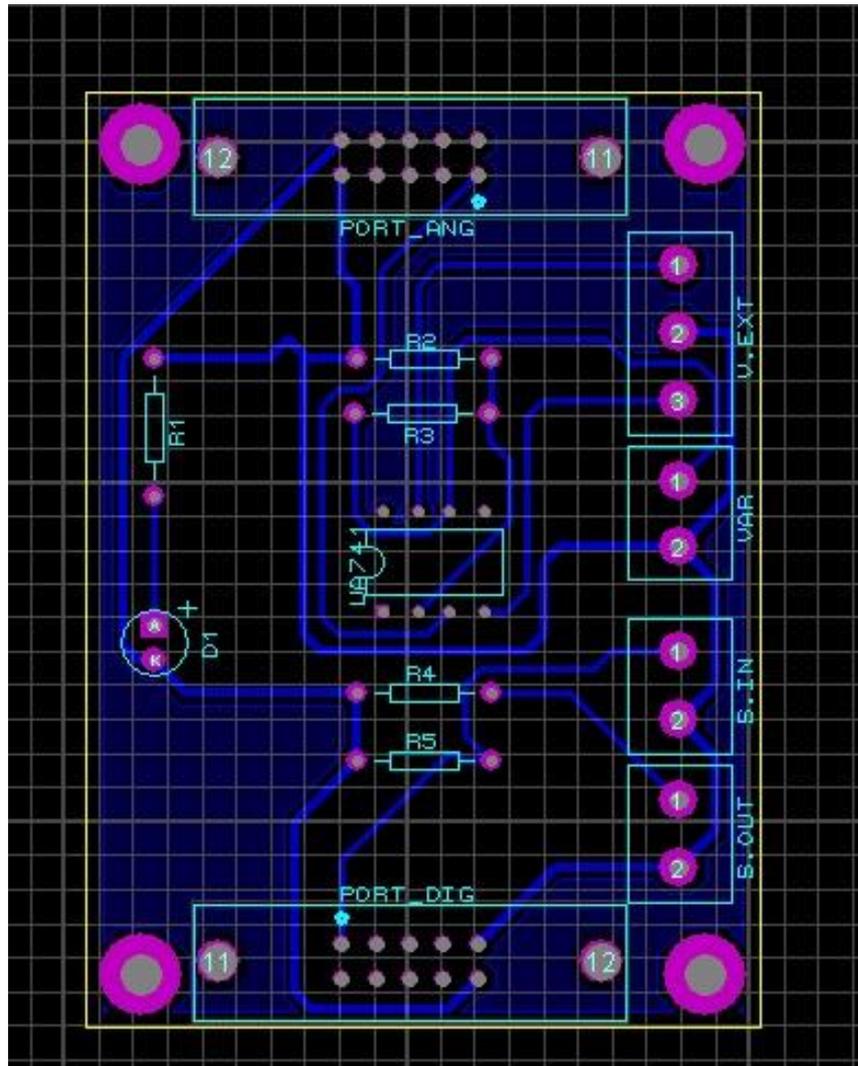


Figura 89: Diseño de circuito Proteus ARES, módulo de control de velocidad de motor
Fuente: Autores, 2014

Este diseño de tarjeta PCB es de una sola cara también, sus elementos electrónicos son básicos y poco complejos.

3.11.4. Montaje de elementos electrónicos en la PCB.

Los elementos distribuidos de la mejor manera, se debe tener precaución en la correcta ubicación de los pines del integrado UA741

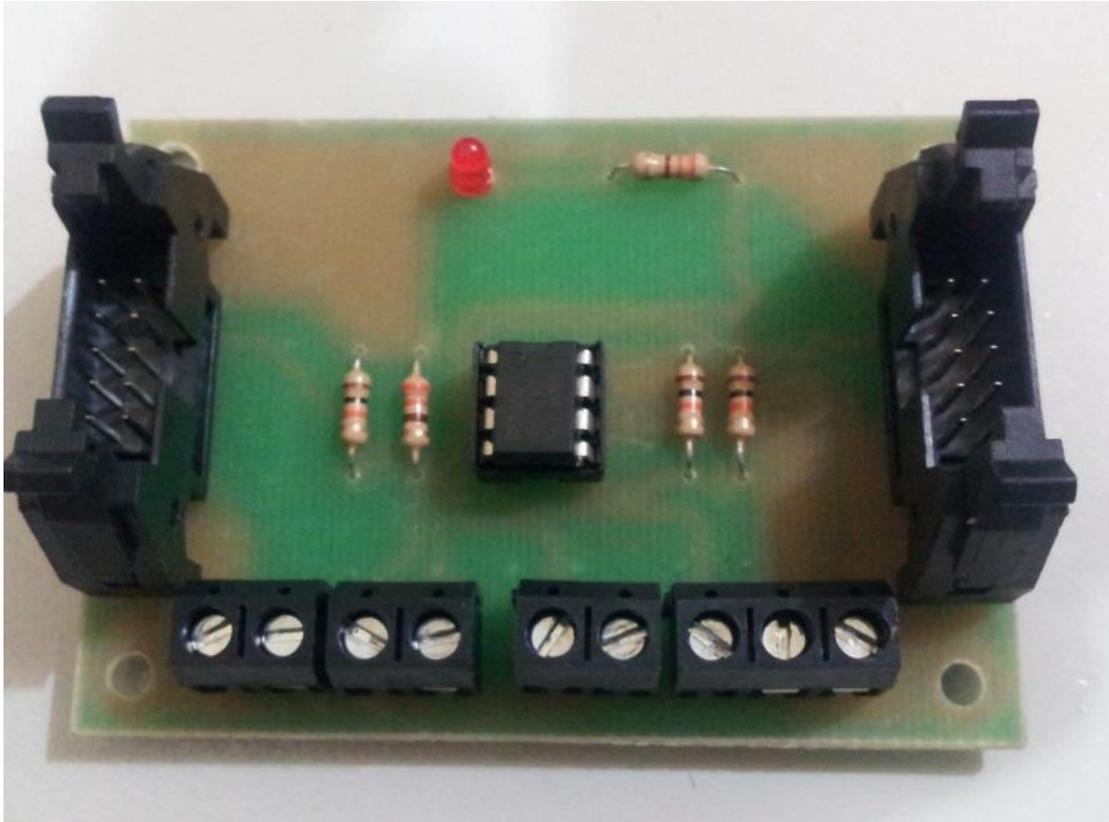


Figura 90: Tarjeta PCB y elementos ensamblados, módulo de control de velocidad de motor
Fuente: Autores, 2014

3.12. Esquema y acoplamiento de los diferentes módulos inalámbricos y el módulo de control de velocidad de corriente alterna.

Una vez realizado cada uno de los módulos se procede a darle forma al entrenador didáctico a través de un chasis metálico tal como se detalla en la siguiente figura se realiza el esquema del mismo y sus medidas.

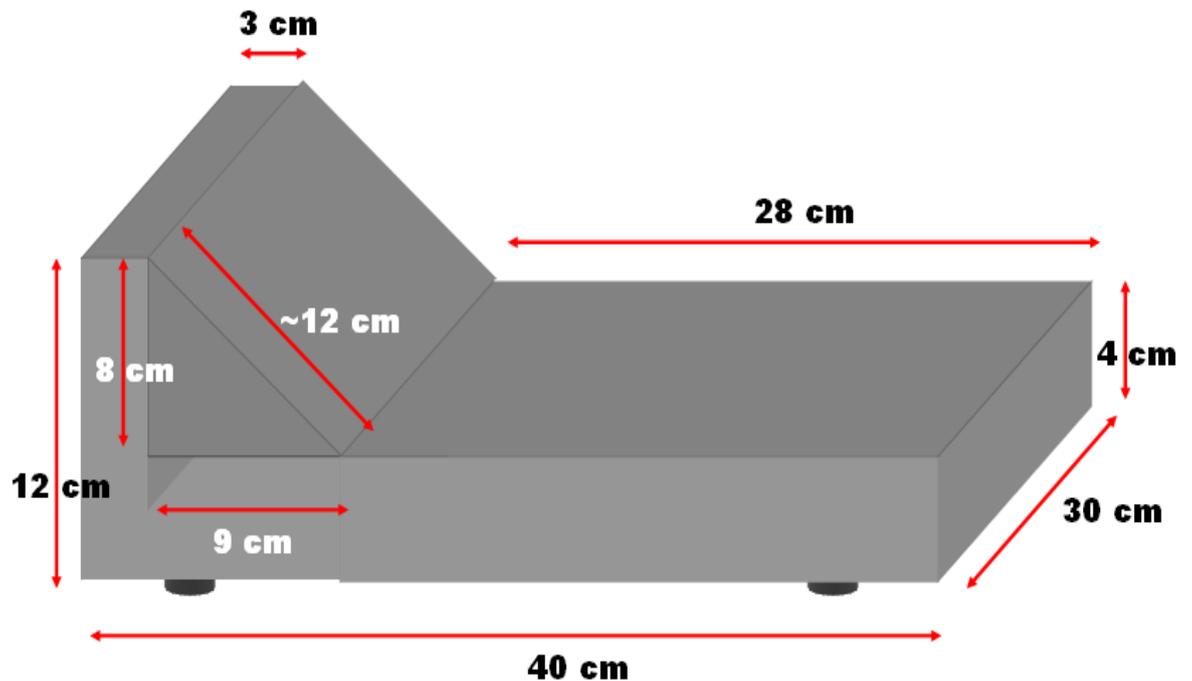


Figura 91: Diseño de chasis metálico del entrenador didáctico.
Fuente: Autores, 2014

Mediante la siguiente figura se muestra el chasis del entrenador didáctico ya terminado, con un diseño acorde a las exigencias de calidad y elegancia que exige el medio, y listo para el ensamble de los módulos.

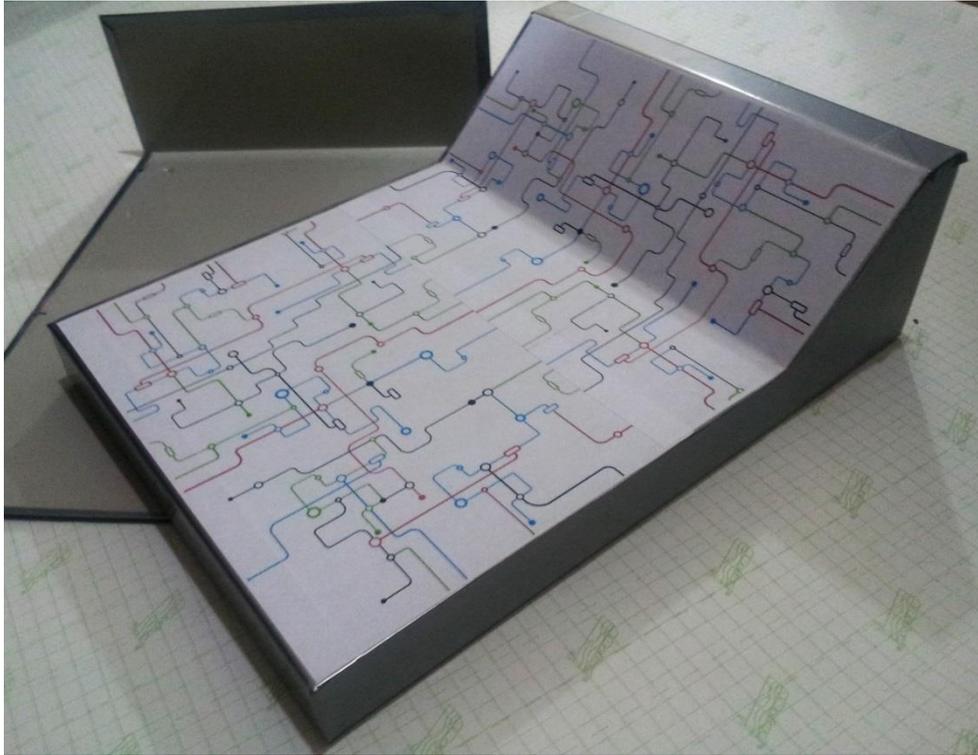


Figura 92: Prototipo chasis de entrenador didáctico.
Fuente: Autores, 2014

Los módulos son colocados de manera estratégica en el chasis del entrenador para el correcto uso del mismo, además en la parte inferior contará con un protoboard para facilitar el desarrollo de las prácticas de laboratorio.

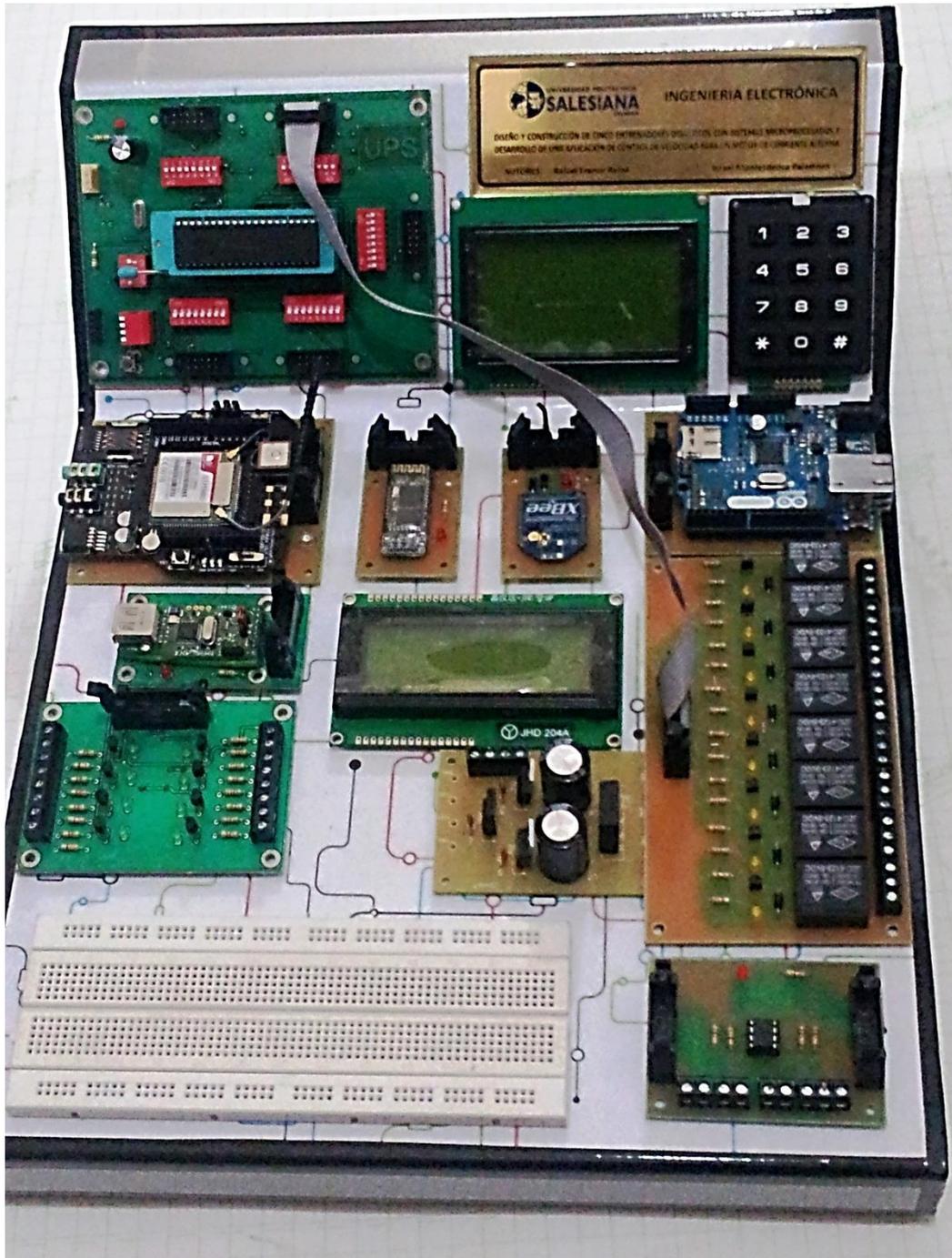


Figura 93: Entrenador didáctico terminado.
Fuente: Autores, 2014

CAPÍTULO IV

MANTENIMIENTO PARA EL ENTRENADOR DIDÁCTICO

La localización de averías también va acompañado de un buen razonamiento deductivo, es así que depende de una observación cuidadosa y evaluación física en busca de evidencias. Sin embargo en la electrónica se dificulta en muchos casos la localización de averías en el modo tradicional es decir buscando evidencias físicas, ya que por ejemplo en los entrenadores dado el caso de haber microcontroladores, resistencias o conectores averiados, estos no gotean, hacen ruido o emiten algún sonido. Por tanto se lineará el mantenimiento de los entrenadores didácticos de la siguiente forma:

4.1. Organización del mantenimiento para los módulos.

El mantenimiento será constituido en el conjunto de actividades que se realizan a un sistema, módulo, equipo o componente con la finalidad de asegurar que continúe desempeñando las funciones deseadas dentro de un contexto operacional determinado.

En la mayoría de libros o manuales se indica que gran parte de dispositivos electrónicos tienen una sola vida y cuando esta concluye deben ser reemplazados y configurados para que todo el sistema vuelva a funcionar correctamente.

4.2. Determinación de las tareas óptimas para la conservación y preservación del entrenador con sistemas microprocesados y aplicación de control de velocidad de motor de corriente alterna.

El conjunto de actividades que se realizará para mantener la funcionalidad del entrenador didáctico con sistemas microprocesados y control de velocidad de un motor de corriente alterna consistirá en tres tareas básicas del mantenimiento preventivo:

Limpieza: Es la actividad más sencilla, económica y a la vez la más efectiva. Conlleva a quitar el polvo, la suciedad residuos u otro tipo de material extraño que se adhiere a los módulos del entrenador. En el desarrollo de esta actividad también se buscará defectos o daños ocultos para luego remediarlos. La limpieza también significa tocar y mirar cada módulo ya sea el de X-Bee, GPS, Ethernet, para detectar defectos o anomalías ocultas, sino se realizara de esta manera se perderá todo significado con respecto a la limpieza.

Inspección: Se averiguará el estado real del entrenador, examinando que todas las partes que componen el entrenador en este caso cada uno de sus módulos estén en buen estado y su funcionalidad sea correcta. Esta inspección la podemos realizar de dos formas:

- Sensorial: como su nombre lo indica utilizaremos nuestros sentidos: oyendo, viendo, oliendo, palpando.
- Instrumental, la más acertada para un ingeniero ya que se empleará instrumentos electrónicos de medición y herramientas con la finalidad de comparar voltajes, corrientes, continuidad, etc.

Ajuste: Lo denominaremos como la calibración y reajuste en los módulos del entrenador para validar condiciones de funcionamientos óptimos del mismo, ejemplo: fijar tornillos flojos, soldar elementos, etc., así se previene fallas graves al entrenador.

4.3. Programación del mantenimiento para el entrenador.

Tarea: Limpieza

Tiempo estimado: 5 minutos

Frecuencia: Antes de usar

Herramientas:

- Pulsera antiestática.
- Pinza extractora de circuitos integrados, microcontroladores.
- Brocha de cerdas finas
- Paño antiestático

Procedimiento:

- Apagar el módulo
- Desconectar de la toma de corriente
- Usar pulsera antiestática.
- Retirar el polvo usando la brocha antiestática.
- Limpiar suciedad con franela antiestática.
- Verificar estado de conectores o zócalos.
- Revisar físicamente cada módulo.
- Probar nuevamente energizando el entrenador.

Tarea: Inspección

Tiempo estimado: 5 minutos

Frecuencia: Cada semana

Herramientas:

- Pulsera antiestática.
- Multímetro

Procedimiento:

- Encender el entrenador didáctico.
- Verificar encendido de diodos led indicadores.
- Usar pulsera antiestática.
- Medir voltajes de corriente directa y alterna, fijos y variables.
- Inspección sensorial tales como oír, oler, ver y tocar.

Tarea: Ajuste

Tiempo estimado: 5 minutos

Frecuencia: Según condición

Herramientas:

- Pulsera antiestática.
- Multímetro
- Cautín
- Estaño
- Destornillador de estrella y plano.

Procedimiento:

- Apagar el entrenador didáctico.
- Desconectar eléctricamente todos los módulos.
- Usar pulsera antiestática.
- Ajustar tornillos flojos o reemplazar alguno en mal estado.
- Soldar elementos.
- Reemplazar elementos electrónicos en casos de averías y de ser necesario.
 - Realizar pruebas y verificar corrección de anomalías en el entrenador.

CAPÍTULO V

PRÁCTICAS DE LABORATORIO

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

5.1. PRÁCTICA 1

Registro de temperatura de un horno mediante un Data Logger.

DATOS INFORMATIVOS

- ✓ MATERIA / CÁTEDRA RELACIONADA: Microcontroladores
- ✓ No. DE PRÁCTICA: 1
- ✓ NÚMERO DE ESTUDIANTES: 2
- ✓ NOMBRE DOCENTE: Ing. Luis Córdova
- ✓ TIEMPO ESTIMADO: 2 Horas

Prácticas a realizarse:

Registro de temperatura de un horno mediante un Data Logger

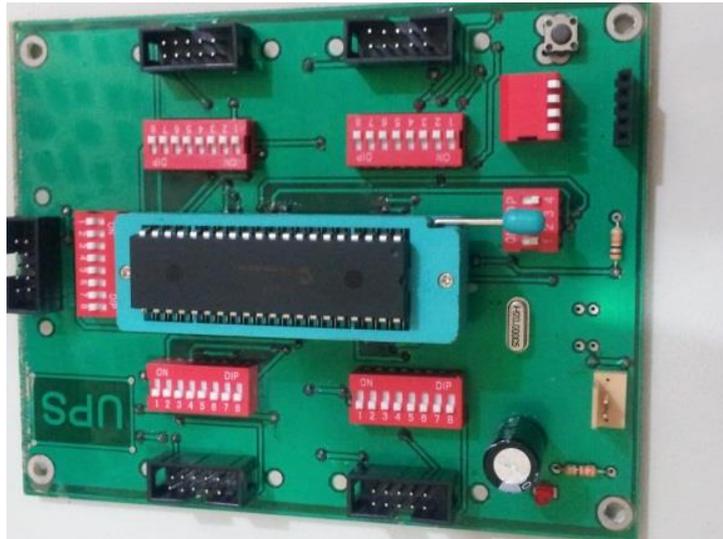
1. Memory Stick Data Logger



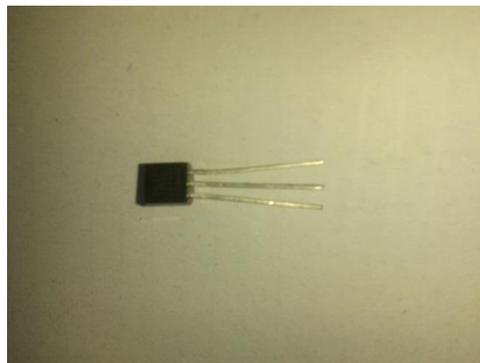
Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

2. Entrenador Didáctico (Módulo de Control)



3. Sensor de Temperatura DS18B20



Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

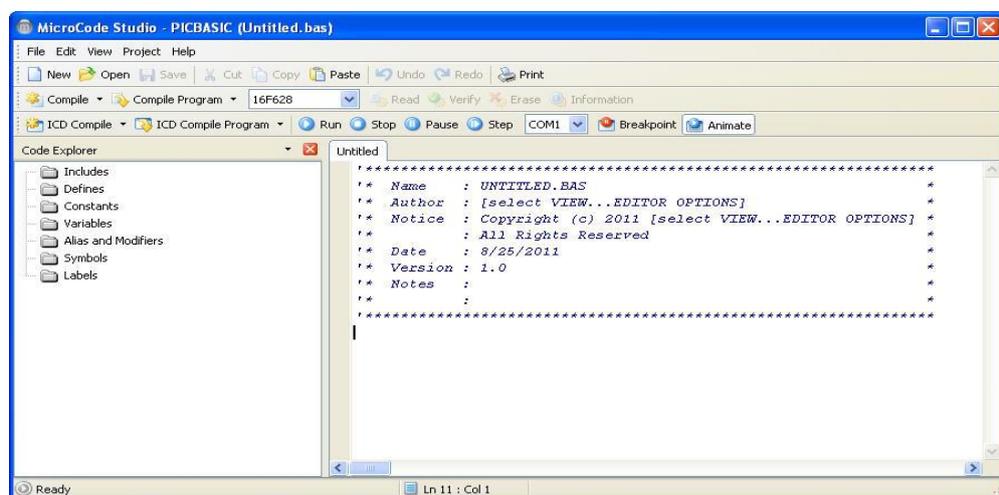
4. Teclado Matricial



5. Display 20x4



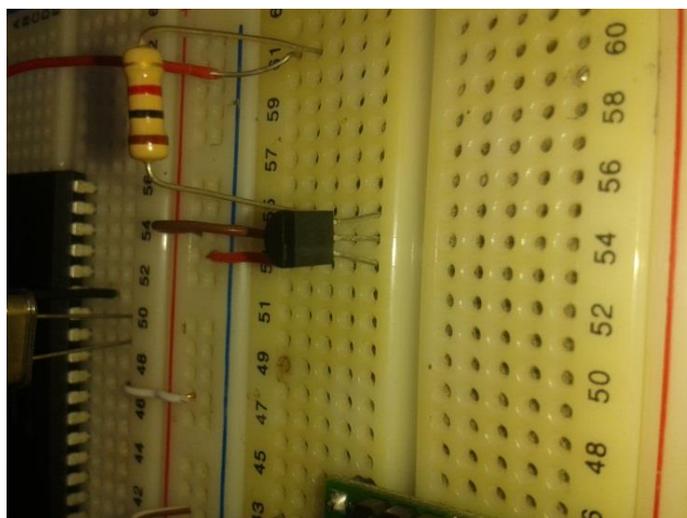
6. Software Micro Code Studio



Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

7. Protoboard



Materiales:

EQUIPO NECESARIO

- ✓ Entrenador Didáctico (Módulo de Control, módulo Datalogger)
- ✓ Sensor de Temperatura DS18B20
- ✓ Teclado Matricial
- ✓ Display 20x4
- ✓ Software Micro Code Studio
- ✓ Proto board

MATERIAL DE APOYO

- ✓ Proyector
- ✓ Pizarra líquida

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Marco Teórico:

La información requerida en esta práctica se puede encontrar revisando el CAPITULO 2, BIBLIOGRAFÍA.

Procedimientos:

Práctica No 1

Registrar la temperatura de un horno por el lapso de 5 Min con intervalo de tiempo de 10seg. Cada temperatura deberá ser guardada con su respectiva hora en un Data Logger. Al final de los 5 Min y con un teclado matricial y un display de 20x4, se deberá seleccionar y visualizar la siguiente información:

- ✓ Cada una de las temperaturas guardadas,
- ✓ La temperatura Máxima
- ✓ La temperatura Mínima

Para esta práctica utilizamos del entrenador didáctico el módulo principal en el cual ingresamos el programa al microcontrolador, el mismo que ejecutará las funciones lectura, almacenamiento, proceso y visualización de los datos obtenidos del sensor de temperatura DS18B20 el cual tiene un rango que va desde -55°C hasta 125°C.

Para ello conectamos un cable de bus de dato suministrado en el entrenador didáctico, en el conector del PUERTO C del módulo de control principal hacia el conector del módulo Datalogger, de igual forma desde el PUERTO B del módulo principal hacia un dispositivo externo que contiene un DS1317 (CLOCK_CALENDAR) necesario para el desarrollo de esta práctica.

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Adicionalmente utilizamos un bus de datos desde el PUERTO_D del módulo principal hacia el Protoboard con su debida adaptación de pines.

En el Protoboard implementamos el circuito para el funcionamiento de sensor de temperatura DS18B20 todo esto mediante un cable de 3 hilos independientes de aproximadamente 2 metros soldado en el sensor de temperatura tomando en cuenta el Pin 1 de dicho conector hacia el Pin_GND, el Pin 2 hacia el Pin PORTD.0 y el Pin 3 hacia +5Vcc, de igual forma puenteamos con una resistencia PULLUP de 4.7KΩ desde +5Vcc hacia el Pin 2 de nuestro DS18B20.

Energizamos la tarjeta principal con la fuente de voltaje +5Vcc incluida en el entrenador didáctico para de esta forma mediante los demás módulos que se encuentran conectado a ella puedan ser energizados.

Ahora procedemos a swichear los pines a utilizar en esta práctica los cuales:

- PUERTO_B: Pin 1 (B0), Pin 2 (B1)
- PUERTO_C: Pin 7 (C6), Pin 8 (C7)
- PUERTO_D: Pin 1 (D0)

Así procedimos a realizar las pruebas de funcionamiento de la práctica y sus requerimientos introduciendo al sensor dentro del horno eléctrico el cual fue conectado previamente a un suministro de energía eléctrica alterna de 110V y ha sido regulada su temperatura a aproximadamente 75°C. Y esperamos un lapso aproximado mayor a las 5 minutos en el cual el display se visualizó el menú con las tres opciones requeridas:

- 1.- TEMPERATURAS ALMACENADAS
- 2.- TEMPERATURA MÁXIMA
- 3.- TEMPERATURA MÍNIMA

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Escogimos mediante teclado matricial la opción 1(TEMPERATURAS ALMACENADAS).

17	10:40:28					
18	10:40:38					
19	10:40:48					
20	10:40:58					
21	10:41:08					
22	10:41:18					
23	10:41:28					
24	10:41:38					
25	10:41:48					
26	10:41:58					

Después de obtenidos estos valores con ayuda del teclado presionamos la tecla # para salir al menú principal, luego se seleccionó la opción 2 (TEMPERATURA MÁXIMA).

De igual forma presionando la tecla # fuimos al menú principal para seleccionar la opción 3 (TEMPERATURA MÍNIMA).

```

*****
'* Name   : Práctica Nro 01.BAS           *
'* Author : Franco Rafael-Montesdeoca Israel *
'* Notice : Copyright (c) 2013 UPS-Guayaquil(Tesis de Grado) *
'*       : All Rights Reserved           *
'* Date   : 11/11/2013                   *
'* Version : 1.0                          *
'* Notes  :                               *
'*       :                               *
*****

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

‘*****DECLARACIÓN DE LIBRERIAS*****’

```
include "modedefs.bas"
$regfile = "m32def.dat"
$crystal = 4000000
$hwstack = 32
$swstack = 10
$framesize = 40
```

‘*****DECLARACIÓN DE PUERTO DE COMUNICACION*****’

```
Declare Sub Lect_mast
  Declare Sub Escr_mast(byvalTx_m As String)
  Declare Sub Lect_rfid
  Declare Sub Escr_rfid(byvalTx_r As String)
  Declare Sub Lect_temp
```

```
'ConfigPorta = &B00000001
'ConfigPortb = &B00001100
'ConfigPortc = &B11000000
'ConfigPortd = &B00001010
```

```
'Config Porta.0 = Input
```

‘*****CONFIGURACION DE PUERTOS*****’

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Config Portb.2 = Input

Config Portb.3 = Input

Config Portc.0 = Output

Config Portc.1 = Output

Config Portc.2 = Output

Config Portc.3 = Output

Config Portc.4 = Output

Config Portc.5 = Output

Config Portc.6 = Input

Config Portc.7 = Input

Config Portd.0 = Input

Config Portd.1 = Output

Config Portd.2 = Input

Config Portd.3 = Output

Config Portd.4 = Input

Config Portd.5 = Input

Config Portd.6 = Input

ConfigLcd = 16 * 2

ConfigLcdpin = Pin , Rs = Portb.0 , E = Portb.1 , Db4 = Portb.4 , Db5 = Portb.5 ,

Db6 = Portb.6 , Db7 = Portb.7

ConfigAdc = Single , Prescaler = Auto

Start Adc

Mem_scl Alias Porta.1

Mem_sda Alias Porta.2

K1 Alias Portc.0

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

K2 Alias Portc.1
 K3 Alias Portc.2
 K4 Alias Portc.3
 K5 Alias Portc.4
 Alarma Alias Portc.5
 S_magn Alias Portc.6
 S_movi Alias Portc.7
 Clock_sda Alias Portd.4
 Clock_scl Alias Portd.5
 Clock_sqw Alias Portd.6

Dim Rx_mast As String * 20
 Dim Tx_mast As String * 20
 Dim Rx_rfid As String * 20
 Dim Tx_rfid As String * 20
 Dim Tarj As String * 20
 Dim Vadc As Byte
 Dim Temp As Word

K1 = 0
 K2 = 0
 K3 = 0
 K4 = 0
 K5 = 0
 Alarma = 0
 S_magn = 0
 S_movi = 0

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Cls

‘*****CALCULOS DE TEMPERATURA*****’

Do

Call Lect_temp

Waitms 100

Temp = Vadc * 150

Temp = Temp / 255

Lcd " UPS "

Lowerline

Lcd "A. K1 - T.(" ; Temp ; Chr(167) ; ")"

K1 = 1

K2 = 0

K3 = 0

K4 = 0

K5 = 0

Waitms 5000

Call Lect_temp

Waitms 100

Temp = Vadc * 150

Temp = Temp / 255

Lcd " UPS "

Lowerline

Lcd "A. K2 - T.(" ; Temp ; Chr(167) ; ")"

K1 = 0

K2 = 1

K3 = 0

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

K4 = 0

K5 = 0

Waitms 5000

Call Lect_temp

Waitms 100

Temp = Vadc * 150

Temp = Temp / 255

Lcd " UPS "

Lowerline

Lcd "A. K3 - T.(" ; Temp ; Chr(167) ; ")"

K1 = 0

K2 = 0

K3 = 1

K4 = 0

K5 = 0

Waitms 5000

Call Lect_temp

Waitms 100

Temp = Vadc * 150

Temp = Temp / 255

Lcd " UPS "

Lowerline

Lcd "A. K4 - T.(" ; Temp ; Chr(167) ; ")"

K1 = 0

K2 = 0

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

K3 = 0

K4 = 1

K5 = 0

Waitms 5000

Call Lect_temp

Waitms 100

Temp = Vadc * 150

Temp = Temp / 255

Lcd " UPS "

Lowerline

Lcd "A. K5 - T.(" ; Temp ; Chr(167) ; ")"

K1 = 0

K2 = 0

K3 = 0

K4 = 0

K5 = 1

Waitms 5000

Loop

End

Sub Lect_mast

‘***** LECTURA DE TEMPERATURA ALMACENADA *****’

SerInRx_mast , 0 , D , 0 , 9600 , 0 , 8 , 1

End Sub

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Sub Escr_mast(byvalTx_m As String)

‘*****RESGISTRO DE TEMPERATURA*****’

SeroutTx_mast , 0 , D , 1 , 9600 , 0 , 8 , 1

End Sub

Sub Lect_rfid

‘*****’

SerInRx_rfid , 0 , D , 2 , 9600 , 0 , 8 , 1

Tarj = Rx_rfid

If Tarj <> "" Then

Cls

Lcd " TEMPERATURA "

Lowerline

Lcd "TMP", TMP ‘***** IMPRIME TEMPERATURA EN PANTALLA*****’

Set K1

Wait 500

Reset K1

Waitms 5000

End If

End Sub

Sub Escr_rfid(byvalTx_r As String)

‘*****ESCRIBIR EN EL PUERTO SERIAL*****’

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

SeroutTx_rfid , 0 , D , 1 , 9600 , 0 , 8 , 1

End Sub

Sub Lect_temp

‘*****CONSULTAR LA TEMPAERATURA*****

Vadc = Getadc(0)

End Sub

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

5.2. PRÁCTICA 2

Desarrollo de un circuito controlador de tránsito

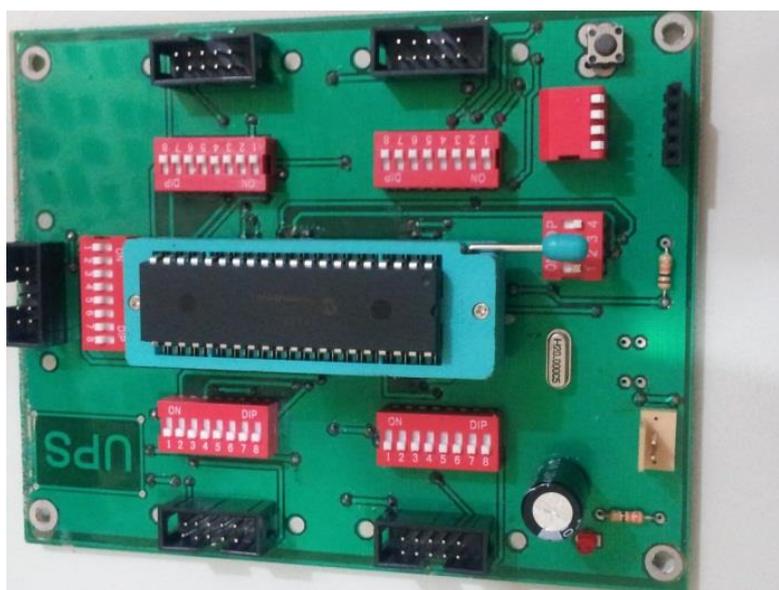
DATOS INFORMATIVOS

- ✓ MATERIA / CÁTEDRA RELACIONADA: Microcontroladores
- ✓ No. DE PRÁCTICA: 2
- ✓ NÚMERO DE ESTUDIANTES: 2
- ✓ NOMBRE DOCENTE: Ing. Luis Córdova
- ✓ TIEMPO ESTIMADO: 2 Horas

Prácticas a realizarse:

Desarrollo de un circuito controlador de tránsito

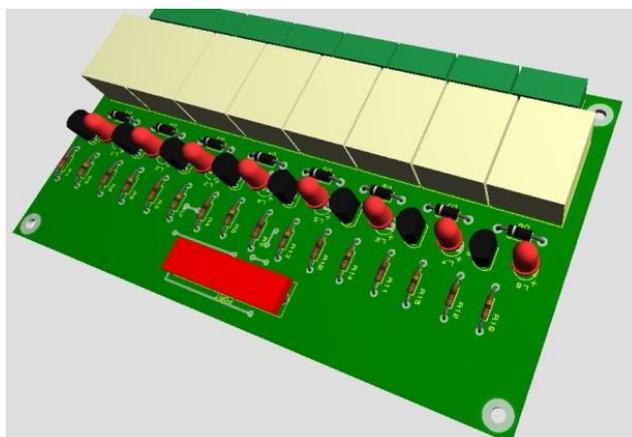
1. Entrenador Didáctico (Módulo de Control)



Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

2. Módulo de interfaz de salida



3. Módulo Arduino Ethernet



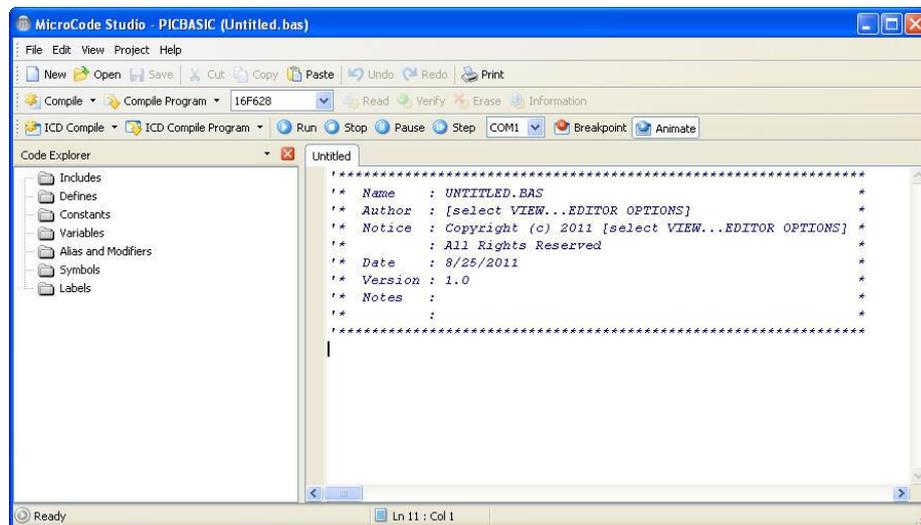
Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

4. Focos indicadores



5. Software Micro Code Studio



Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1	
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	Laboratorio de Electrónica Digital		
CARRERA	Ingeniería Electrónica		
SEDE	Guayaquil		

Materiales:

EQUIPO NECESARIO	MATERIAL DE APOYO
------------------	-------------------

- | | |
|--|----------------------|
| ✓ Entrenador Didáctico (Módulo de Control) | ✓ Proyector |
| ✓ Módulo de interfaz de salida | ✓ Pizarra líquida |
| ✓ Módulo Arduino Ethernet | ✓ Cronómetro digital |
| ✓ Focos indicadores | |
| ✓ Software Micro Code Studio | |

Marco Teórico:

A información requerida en esta práctica se puede encontrar revisando el CAPITULO 2, BIBLIOGRAFÍA.

Procedimientos:

Práctica No 2

Realizar un circuito controlador de tránsito (semáforo), para una intersección de 2 avenidas. Los tiempos deberán ser ingresados por una PC y transferidos al controlador por medio de la red Ethernet.

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Para esta práctica utilizamos del entrenador didáctico, el módulo principal, en el cual ingresamos el programa al microcontrolador, para ello conectamos un cable de bus de dato suministrado en el entrenador didáctico, en el conector del PUERTO C del módulo de control principal hacia el conector del módulo ethernet, de igual forma desde el PUERTO B del módulo principal hacia la interfaz de salida para controlar los relays que accionaran las luces de los semáforos, los cuales deberán estar conectados en las borneras de salidas para cada lámpara con su respectivo relay siendo esta la conexión desde la interfaz de salida hacia los semáforos mediante los cables concéntricos de cada uno de los semáforos didácticos :

- RELAY_1, NA → semáforo 1, lámpara ROJO (cable rojo)
- RELAY_2, NA → semáforo 1, lámpara AMARILLO (cable negro)
- RELAY_3, NA → semáforo 1, lámpara VERDE (cable verde)

Fuente externa 110VAC, NEUTRO → semáforo 1, común (cable blanco)

- RELAY_4, NA → semáforo 2, lámpara ROJO (cable rojo)
- RELAY_5, NA → semáforo 2, lámpara AMARILLO (cable negro)
- RELAY_6, NA → semáforo 2, lámpara VERDE (cable verde)

Fuente externa 110VAC, NEUTRO → semáforo 2, común (cable blanco)

Fuente externa 110VAC, LINEA → interfaz de salida puentada en las borneras a cada pin común (COMMON) de los terminales de los RELAYS utilizados

Conectamos el módulo Ethernet hacia el computador mediante cable de red cruzado por el JACK RJ45 del Arduino Ethernet que cuenta el módulo hacia la tarjeta de red LAN del computador.

Energizamos la tarjeta principal con la fuente de voltaje +5Vcc incluida en el entrenador didáctico para de esta forma mediante los cables de bus de datos (pin 9, GND; pin 10, +5Vcc) conectados hacia los demás módulos utilizados puedan ser energizados.

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Ahora procedemos a swichear los pines a utilizar en esta práctica encontrados en el módulo de control principal que permitirán establecer la conexión desde el microcontrolador hacia los conectores de los diferentes puertos:

- PUERTO_B: Pin 1 (B0), Pin 2 (B1), Pin 3 (B2), Pin 4 (B3), Pin 5 (B4), Pin 6 (B5)
- PUERTO_C: Pin 7 (C6), Pin 8 (C7)

De esta forma procedimos a poner en funcionamiento el circuito tomando como valores de tiempo inicial (DEFAULT) almacenados previamente en la memoria EPROM del microcontrolador: *“eprom 50, [30,22,3,1,5,5]”* , de los cuales los valores iniciales para la variable fueron:

- Semaforo_1, Verde → 30 segundos
- Semaforo_2, Verde → 22 segundos
- Semaforo_1,2, Amarillo → 3 segundos
- Semaforo_1,2, Rojo → 1 segundos

Previamente en el computador se ingresó los tiempos de acción para cada semáforo mediante una aplicación didáctica de apoyo diseñada en Visual Basic 6.0 la cual se encarga de enviar automáticamente la trama con los tiempos que seteados después de haber transcurrido el primer ciclo semafórico.

Luego de este procedimiento verificamos con la ayuda de un cronómetro digital cada tiempo de acción de las lámparas de los semáforos notando su efectividad.

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

*****
'* Name   : Práctica Nro 02.BAS                *
'* Author : Franco Rafael - Motesdeoca Israel  *
'* Notice : Copyright (c) 2013 UPS-Guayaqui (Tesis de Grado) *
'*       : All Rights Reserved                 *
'* Date   : 04/11/2013                         *
'* Version : 1.0                               *
'* Notes  :                                     *
'*       :                                     *
*****

```

***** DECLARACION DE LIBRERIAS*****

```
include "modedefs.bas" 'LIBRERIA DE COMUNICACIÓN
```

```
DEFINE OSC 20      'OSCILADOR DE 20Mhz
```

```

*****
***** ASIGNACION DE PUERTOS *****
*****

```

```
;=====DEFINICION DE SALIDAS=====
```

```

ROJO1  VAR PORTB.0
AMA1   VAR PORTB.1
VERDE1 VAR PORTB.2
ROJO2  VAR PORTB.3
AMA2   VAR PORTB.4

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

VERDE2 VAR PORTB.5

;=====DEFINICION DE ENTRADA =====

CONF VAR PORTD.0

;=====PUERTO DE COMUNICACIÓN =====

TX VAR PORTC.6

RX VAR PORTC.7

;***** DECLARACIÓN DE VARIABLES*****

V1N VAR BYTE

V2N VAR BYTE

AN VAR BYTE

RN VAR BYTE

TIR VAR BYTE

TIF VAR BYTE

TV1 VAR BYTE

TV2 VAR BYTE

TAM VAR BYTE

TRO VAR BYTE

TCO VAR BYTE

X VAR WORD

NUMERO VAR BYTE

;*****PROGRAMACION DE PUERTOS*****

TRISB = 0

TRISC = %10000000

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

TRISD = 255

; AQUI COMIENZA EL PROGRAMA

eeprom 50,[30,22,3,1,5,5]

INICIO:

PORTB = 0

PORTC = 0

PANTALLA1:

FOR X = 0 TO 300

PAUSE 10

NEXT X

INI:

read 50,V1N

read 51,V2N

read 52,AN

read 53,RN

READ 54,TIR

READ 55,TIF

goto ENTRADA

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

FASE1:

PORTB = 0

HIGH VERDE1: HIGH ROJO2

TV1 = V1N

TV2 = V2N

TAM = AN

TRO = RN

HIGH VERDE1: HIGH ROJO2

tco = 1

if tv1 > 10 then TCO = TV1 - 10

FOR X = TV1 TO TCO STEP -1

 PAUSE 1000

 IF CONF = 0 THEN

 GOTO CONFIGURA

 ENDIF

NEXT X

FOR X = tco - 1 TO 1 STEP -1

 PAUSE 1000

NEXT X

FASEAMAR1:

LOW VERDE1

HIGH AMA1: HIGH ROJO2

FOR X = TAM TO 1 STEP -1

 PAUSE 1000

NEXT X

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

gosub ROJOS

FASE2:

PORTB = 0

HIGH ROJO1: HIGH VERDE2

FOR X = TV2 TO 1 STEP -1

PAUSE 1000

NEXT X

FASEAMAR2:

LOW VERDE2

HIGH ROJO1: HIGH AMA2

FOR X = TAM TO 1 STEP -1

PAUSE 1000

NEXT X

gosub ROJOS

goto FASE1

ROJOS:

LOW AMA1: LOW AMA2

HIGH ROJO1: HIGH ROJO2

FOR X = TRO TO 1 STEP -1

PAUSE 1000

NEXT X

return

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

***** CONFIGURACIÓN DE TIEMPOS *****

CONFIGURA:

NUMERO = 0

serout2 tx, 84, ["TMP", 10, 13] 'PETICION DE CONFIGURACIÓN

SERIN2 RX, 84, [WAIT("TMP="), NUMERO] 'RECEPCIÓN DE

CONFIGURACIÓN LISTO

IF NUMERO = 1 THEN GOTO ING_TMP

IF CONF = 1 THEN GOTO INI

GOTO CONFIGURA

PAUSE 500

ING_TMP:

***** INGRESO TMP VERDE PRINCIPAL *****

NUMERO = 0

serout2 tx, 84, ["V1", 10, 13]

SERIN2 RX, 84, [WAIT("V1="), DEC NUMERO]

V1N = NUMERO

PAUSE 500

write 50,V1N

***** INGRESO TMP VERDE SECUNDARIO *****

NUMERO = 0

serout2 tx, 84, ["V2", 10, 13]

SERIN2 RX, 84, [WAIT("V2="), DEC NUMERO]

V2N = NUMERO

PAUSE 500

write 51,V2N

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

***** INGRESO TMP AMARILLO *****

NUMERO = 0

serout2 tx, 84, ["AN", 10, 13]

SERIN2 RX, 84, [WAIT("AN="), DEC NUMERO]

AN = NUMERO

PAUSE 500

write 52,AN

***** INGRESO TMP ROJO *****

NUMERO = 0

serout2 tx, 84, ["RN", 10, 13]

SERIN2 RX, 84, [WAIT("RN="), DEC NUMERO]

RN = NUMERO

PAUSE 500

write 53,RN

***** INGRESO TMP ENTRADA ROJO *****

NUMERO = 0

serout2 tx, 84, ["TIR", 10, 13]

SERIN2 RX, 84, [WAIT("TIR="), DEC NUMERO]

TIR = NUMERO

PAUSE 500

write 54,TIR

***** INGRESO TMP ENTRADA FLASHEO *****

NUMERO = 0

serout2 tx, 84, ["TIF", 10, 13]

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

SERIN2 RX, 84, [WAIT("TIF="), DEC NUMERO]

TIF = NUMERO

PAUSE 500

write 55,TIF

goto INI

ENTRADA:

PORTB = 0

FOR X = 1 TO TIF

HIGH AMA1 : low ROJO2

PAUSE 500

LOW AMA1 : high ROJO2

PAUSE 500

NEXT X

high ROJO1: HIGH ROJO2

FOR X = 1 TO TIR

PAUSE 1000

NEXT X

GOTO FASE1

end

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

5.3. PRÁCTICA 3

Mediante un sensor tomar muestras de temperatura y enviarlas inalámbricamente usando X-Bee

DATOS INFORMATIVOS

- ✓ MATERIA / CÁTEDRA RELACIONADA: Microcontroladores
- ✓ No. DE PRÁCTICA: 3
- ✓ NÚMERO DE ESTUDIANTES: 2
- ✓ NOMBRE DOCENTE: Ing. Luis Córdova
- ✓ TIEMPO ESTIMADO: 2 Horas

Prácticas a realizarse:

Mediante un sensor tomar muestras de temperatura y enviarlas inalámbricamente usando X-Bee

1. Entrenador Didáctico (Módulo de Control)



Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

2. Módulo de XBEE



3. Display GLCD



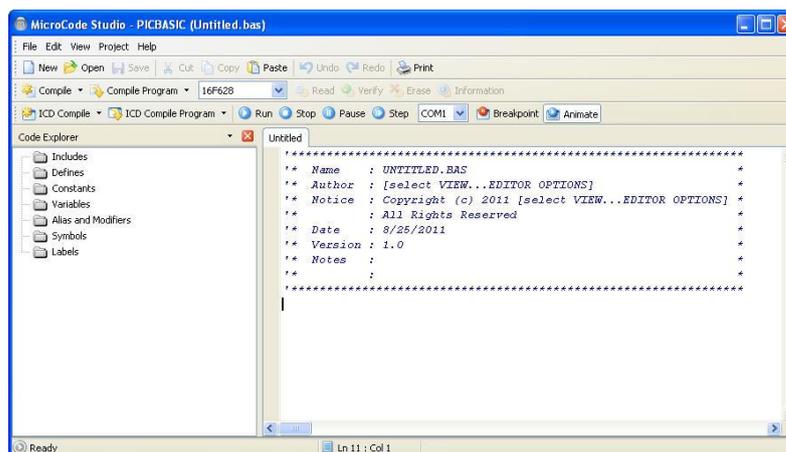
4. Sensor DS18B20



Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

5. Software Micro Code Studio



Materiales:

EQUIPO NECESARIO	MATERIAL DE APOYO
------------------	-------------------

- | | |
|--|--|
| <ul style="list-style-type: none"> ✓ Entrenador Didáctico (Módulo de Control) ✓ Módulo de XBEE ✓ Display GLCD ✓ Sensor DS18B20 ✓ Software Micro Code Studio | <ul style="list-style-type: none"> ✓ Proyector ✓ Pizarra líquida |
|--|--|

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Marco Teórico:

A información requerida en esta práctica se puede encontrar revisando el CAPITULO 2, BIBLIOGRAFÍA.

Procedimientos:

Práctica No 3

Mediante el sensor DS18B20 tomar muestras de temperatura y enviarlas inalámbricamente usando X-Bee a un punto remoto (200mt) para ser visualizadas en un GLCD.

En esta práctica utilizamos 2 módulos entrenadores didácticos:

- Entrenador A → EMISOR
- Entrenador B → RECEPTOR

ENTRENADOR A:

A este entrenador se le asignó la tarea de obtener una temperatura mediante el sensor DS18B20 para luego ser enviadas inalámbricamente mediante el módulo Xbee siguiendo el siguiente proceso:

El módulo principal en el cual ingresamos el programa al microcontrolador, el mismo que ejecutará las funciones lectura, proceso y envío de los datos obtenidos del sensor de temperatura DS18B20 el cual tiene un rango que va desde -55°C hasta 125°C.

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Utilizando un cable de bus de dato suministrado en el entrenador didáctico conectamos desde el conector del PUERTO C del módulo de control principal hacia el conector del módulo Xbee, de igual forma desde el PUERTO_D del módulo principal hacia el Protoboard con su debida adaptación de pines.

En el Protoboard implementamos el circuito para el funcionamiento de sensor de temperatura DS18B20 todo esto mediante un cable de 3 hilos independientes de aproximadamente 2 metros soldado en el sensor de temperatura tomando en cuenta el Pin 1 de dicho conector hacia el Pin_GND, el Pin 2 hacia el Pin PORTD.0 y el Pin 3 hacia +5Vcc, de igual forma puenteamos con una resistencia PULLUP de 4.7K Ω desde +5Vcc hacia el Pin 2 de nuestro DS18B20.

Energizamos la tarjeta principal con la fuente de voltaje +5Vcc incluida en el entrenador didáctico para de esta forma mediante los cables de bus de datos (pin 9, GND; pin 10, +5Vcc) conectados hacia los demás módulos utilizados puedan ser energizados.

Ahora procedemos a swichear los pines a utilizar en esta práctica los cuales:

- PUERTO_C: Pin 7 (C6), Pin 8 (C7)
- PUERTO_D: Pin 1 (D0)

ENTRENADOR B:

A este entrenador se le asignó la tarea de receptar inalámbricamente los datos de las temperaturas obtenidas mediante el sensor DS18B20 en el ENTRENADOR A.

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

En este entrenador didáctico programamos al microcontrolador en el módulo principal para realizar las funciones lectura, proceso y visualización de las temperaturas en la GLCD_128X64.

Utilizando un cable de bus de dato suministrado en el entrenador didáctico conectamos desde el conector del PUERTO C del módulo de control principal hacia el conector del módulo Xbee. Así mismo conectamos desde los conectores del PUERTO_B y PUERTO_D hacia los pines de la GLCD.

Energizamos la tarjeta principal con la fuente de voltaje +5Vcc incluida en el entrenador didáctico para de esta forma mediante los cables de bus de datos (pin 9, GND; pin 10, +5Vcc) conectados hacia los demás módulos utilizados puedan ser energizados.

Ahora procedemos a swichear los pines de la siguiente manera:

- PUERTO_B:
 - Pin 1 (B0) → GLCD_128X64, Pin 4 (D/I)
 - Pin 2 (B1) → GLCD_128X64, Pin 5 (R/W)
 - Pin 3 (B2) → GLCD_128X64, Pin 6 (E)
 - Pin 4 (B3) → GLCD_128X64, Pin 15 (CS1)
 - Pin 5 (B4) → GLCD_128X64, Pin 16 (CS2)
 - Pin 6 (B5) → GLCD_128X64, Pin 17 (RESET)
- PUERTO_C:
 - Pin 7 (C6) → MÓDULO_XBEE, Pin 3 (DIN)
 - Pin 8 (C7) → MÓDULO_XBEE, Pin 2 (DOUT)
- PUERTO_D:
 - Pin 1 (D0) → GLCD_128X64, Pin 7 (B0)

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

- Pin 2 (D1) → GLCD_128X64, Pin 8 (B1)
- Pin 3 (D2) → GLCD_128X64, Pin 9 (B2)
- Pin 4 (D3) → GLCD_128X64, Pin 10 (B3)
- Pin 5 (D4) → GLCD_128X64, Pin 11 (B4)
- Pin 6 (D5) → GLCD_128X64, Pin 12 (B5)
- Pin 7 (D6) → GLCD_128X64, Pin 13 (B6)
- Pin 8 (D7) → GLCD_128X64, Pin 14 (B7)

Así procedimos a realizar las pruebas de funcionamiento de la práctica y sus requerimientos, exponiendo al sensor a varias fuentes de temperatura externas (encendedor de bolsillo, cubeta de hielo) en periodos de aproximadamente 60 segundos que son los ciclos de lectura del sensor DS18B20, programados en el microcontrolador 18F4550 del módulo de control del ENTRENADOR A, el mismo que se encargó de enviar los datos de temperatura inalámbricamente mediante el módulo Xbee (A) hacia el módulo Xbee (B) que los recibe y transmite al microcontrolador 18F4559 del ENTRENADOR B y este a su vez se encarga de mostrarlos en la GLCD_128X64

	TEMPERATURAS			
FUENTE EXTERNA	T1	T2	T3	T4
Encendedor de bolsillo	97°C	81°C	113°C	128°C
Cubeta de hielo	6°C	-14°C	-27°C	-23°C

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```
* Name   : Práctica Nro 03.BAS                *
* Author : Franco Rafael - Motesdeoca Israel  *
* Notice : Copyright (c) 2014 UPS-Guayaqui (Tesis de Grado) *
*       : All Rights Reserved                  *
* Date   : 05/01/2014                          *
* Version : 1.0                                *
* Notes  :                                     *
*       :                                     *
```

```
include "modedefs.bas"
```

```
*****DEFINICION DE PULSANTES*****
```

```
tx   var portb.0
rx   var portb.1
tx1  var portb.2
np   var portc.0
```

```
*****VARIABLES DE SERIAL*****
```

```
SIO   VAR PORTC.3
```

```
v     var word
vx    var word
```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

vy var word
 vm var word
 vn var word
 vx1 var word
 vy1 var word
 vm1 var word
 vn1 var word
 x1 var byte
 k var byte
 kx var byte
 ky var byte
 km var byte
 kn var byte
 np1 var byte
 dato_osa VAR BYTE [2]
 Ia var byte
 Ja var byte
 Xa var byte

*****VARIABLES DE LCD*****

```

DEFINE LCD_DREG  PORTC
DEFINE LCD_DBIT  4
DEFINE LCD_RSREG  PORTA
DEFINE LCD_RSBIT  0
DEFINE LCD_EREG  PORTA
DEFINE LCD_EBIT  1
  
```

*****VARIABLES DE SALIDA DEL PIC*****

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

HIGH SIO

ADCON1 = 7

OPTION_REG.6 = 0

OPTION_REG.7 = 0

PAUSE 1000

,*****

high RX

high tx1

high tx

PAUSE 100

low portb.3

low portb.4

low portb.5

IF X = 0 THEN : X = "I"

IF X = 1 THEN : X = "O" : x1 = "-"

pause 1000

if np=1 then goto mens1

xx = xx + 1

next

serout2 tx,84,["XBEE",10,13]

PAUSE 100

high portb.4

pause 500

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

low portb.4

goto INI3

mens1:

xx = 0

for xx = 0 to 2

high portb.5

pause 300

low portb.5

pause 300

SEROUT2 SIO,188,["TMP.: ", TMP,8]

pause 350

goto INI3

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

5.4. PRÁCTICA 4

Realizar un circuito micro controlado capaz de detectar la presencia de algún intruso en una vivienda.

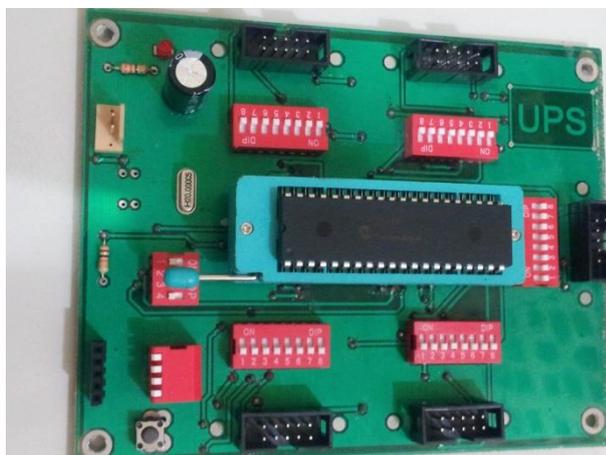
DATOS INFORMATIVOS

- ✓ MATERIA / CÁTEDRA RELACIONADA: Microcontroladores
- ✓ No. DE PRÁCTICA: 4
- ✓ NÚMERO DE ESTUDIANTES: 2
- ✓ NOMBRE DOCENTE: Ing. Luis Córdova
- ✓ TIEMPO ESTIMADO: 2 Horas

Prácticas a realizarse:

Realizar un circuito micro controlado capaz de detectar la presencia de algún intruso en una vivienda.

1. Entrenador Didáctico (Módulo de Control)



Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

2. Módulo de GSM – GPRS – GPS



3. Sensor PirMotion



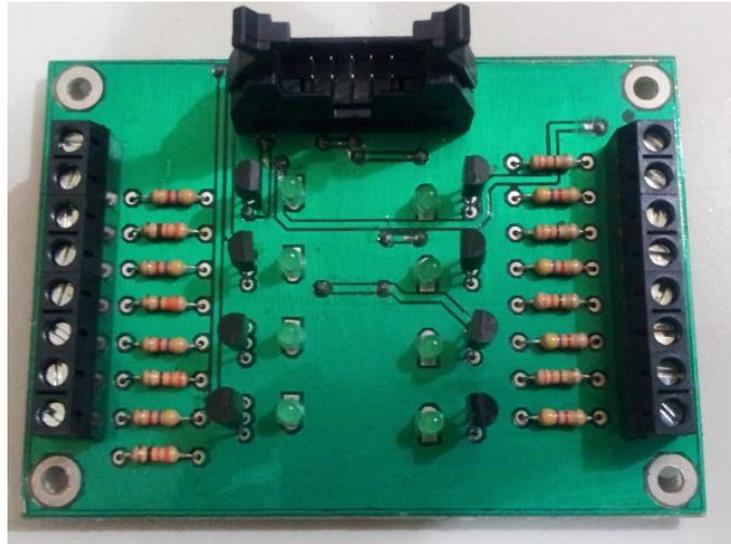
4. Maqueta Didáctica (casa)



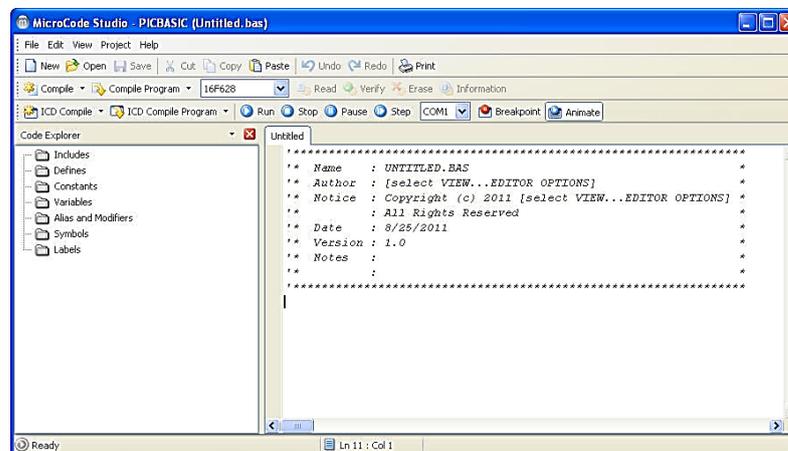
Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

5. Interfaz de Entrada



6. Software Micro Code Studio



Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1	
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	Laboratorio de Electrónica Digital		
CARRERA	Ingeniería Electrónica		
SEDE	Guayaquil		

Materiales:

EQUIPO NECESARIO	MATERIAL DE APOYO
------------------	-------------------

- | | |
|---|--|
| <ul style="list-style-type: none"> ✓ Entrenador Didáctico (Módulo de Control) ✓ Módulo de GSM – GPRS - GPS ✓ Sensor PirMotion ✓ Maqueta Didactica (casa) ✓ Interfaz de Entrada ✓ Software Micro Code Studio | <ul style="list-style-type: none"> ✓ Proyector ✓ Pizarra líquida |
|---|--|

Marco Teórico:

A información requerida en esta práctica se puede encontrar revisando el CAPITULO 2, BIBLIOGRAFÍA.

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Procedimientos:

Práctica 4

Realizar un circuito micro controlado capaz de detectar la presencia de algún intruso en alguna de las 5 zonas de una vivienda. Si detecta la presencia de algún desconocido se deberá enviar un mensaje GSM al celular del propietario.

- ✓ Para esta práctica utilizamos del entrenador didáctico el módulo principal en el cual ingresamos el programa al microcontrolador, el mismo que ejecutará las funciones lectura, proceso y envío de los datos obtenidos de los sensores de movimiento PirMotion

Para ello conectamos un cable de bus de dato suministrado en el entrenador didáctico, en el conector del PUERTO C del módulo de control principal hacia el conector del módulo GSM – GPRS - GPS, de igual forma desde el PUERTO B del módulo principal hacia el conector de la interfaz de entrada.

Adicionalmente conectaremos los cinco sensores PirMotion hacia la interfaz de entrada, ocupando las cinco primeras entradas de la siguiente manera:

- Sensor PirMotion A → Interfaz de Entrada, IN-1 (TERMINAL 2)
- Sensor PirMotion B → Interfaz de Entrada, IN-2 (TERMINAL 2)
- Sensor PirMotion C → Interfaz de Entrada, IN-3 (TERMINAL 2)
- Sensor PirMotion D → Interfaz de Entrada, IN-4 (TERMINAL 2)
- Sensor PirMotion E → Interfaz de Entrada, IN-5 (TERMINAL 2)

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Cada sensor fue conectado con un cable de 3 hilos de aproximadamente 2 metros de distancia, de la siguiente manera para su funcionamiento:

- Sensor PirMotion, PIN 1 → Fuente, +5Vcc
- Sensor PirMotion, PIN 2 → Interfaz de Entrada, IN _ # (TERMINAL 2)
- Sensor PirMotion, PIN 3 → Fuente, GND

En el Módulo GSM – GPRS – GPS, instalamos en la ranura SIM una tarjeta SIM de nuestra operadora de telefonía celular que en este caso fue la empresa de comunicación CLARO (CONECEL) con la cual se conto al moneto de realizar las pruebas, la tarjeta debe estar activada con su respectivo paquete de mensajes de texto o en su defecto con saldo a favor para realizar la prueba.

Energizamos la tarjeta principal con la fuente de voltaje +5Vcc incluida en el entrenador didáctico para de esta forma mediante los demás módulos que se encuentran conectado a ella puedan ser energizados.

Ahora procedemos a swichear los pines a utilizar en esta práctica los cuales:

- PUERTO_B:
 - Pin 1 (B0) → Interfaz de Entrada, Pin1 (IN_1)
 - Pin 2 (B1) → Interfaz de Entrada, Pin2 (IN_2)
 - Pin 3 (B2) → Interfaz de Entrada, Pin3 (IN_3)
 - Pin 4 (B3) → Interfaz de Entrada, Pin4 (IN_4)
 - Pin 5 (B4) → Interfaz de Entrada, Pin5 (IN_5)
- PUERTO_C:
 - Pin 7 (C6, TX) → Módulo GSM – GPRS – GPS, Pin7 (RX)
 - Pin 8 (C7, RX) → Módulo GSM – GPRS – GPS, Pin8 (TX)

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Así procedimos a realizar las pruebas de funcionamiento de la práctica y sus requerimientos, ubicando los sensores en la maqueta didáctica, especificando la ubicación estratégica para cada uno de ellos.

Cuando se comenzo con las pruebas para cada uno de los sensores tomamos en cuenta que para uno de ellos al momento de ser activados se encendia el LEd indicando de cada una de las entradas y a si mismo el microcontrolador de la tarjeta principal de control recepto estas señales y se encargo de iniciar el proceso de envio de los mensajes SMS mediante el Módulo GSM – GPRS – GPS el mismo que se encargo de realizar el envio exitosamente hacia el numero de celular registrado previamente en el código de programa y almacenado en el microcontrolador.

Se Mostro el Siguiete mensaje en el teléfono celular remoto:

ALERTA DE INTRUSO

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```
* Name : Práctica Nro 04.BAS *
```

```
* Author : Franco Rafael - Motesdeoca Israel *
```

```
* Notice : Copyright (c) 2014 UPS-Guayaqui (Tesis de Grado) *
```

```
* : All Rights Reserved *
```

```
* Date : 23/01/2014 *
```

```
* Version : 1.0 *
```

```
* Notes :
```

```
* :
```

```
include "modedefs.bas"
```

```
*****DEFINICION DE PULSANTES*****
```

```
tx var portb.0
rx var portb.1
tx1 var portb.2
np var portc.0
```

```
*****VARIABLES DE SERIAL*****
```

```
SIO VAR PORTC.3

MINUTO VAR BYTE
DHORA VAR BYTE
UHORA VAR BYTE
DMINU VAR BYTE
UMINU VAR BYTE
```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

DH VAR BYTE
 UH VAR BYTE
 DM VAR BYTE
 UM VAR BYTE
 I VAR word
 Ix var word
 J VAR word
 X VAR BYTE
 xx var byte
 GRADOS VAR BYTE
 MINUTOS VAR BYTE
 GRA VAR BYTE
 MINU VAR BYTE
 v var word
 vx var word
 vy var word
 vm var word
 vn var word
 vx1 var word
 vy1 var word
 vm1 var word
 vn1 var word
 x1 var byte
 k var byte
 kx var byte
 ky var byte
 km var byte
 kn var byte

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

np1 var byte

dato_osa VAR BYTE [2]

Ia var byte

Ja var byte

Xa var byte

*****VARIABLES DE LCD*****

DEFINE LCD_DREG PORTC

DEFINE LCD_DBIT 4

DEFINE LCD_RSREG PORTA

DEFINE LCD_RSBIT 0

DEFINE LCD_EREG PORTA

DEFINE LCD_EBIT 1

*****VARIABLES DE SALIDA DEL PIC*****

HIGH SIO

ADCON1 = 7

OPTION_REG.6 = 0

OPTION_REG.7 = 0

PAUSE 1000

high RX

high tx1

high tx

PAUSE 100

low portb.3

low portb.4

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

low portb.5

INI3:

,*****VELOCIDAD*****

```

if np=1 then goto mens1
xx = 0
for xx = 0 to 10
if np=1 then goto mens1
high portb.5
pause 300
low portb.5
pause 300
if np=1 then goto mens1
SEROUT2 SIO,188,["!GPS",8]
SERIN2 SIO,188,[I,J]
k = I * 256
k = I + J
k = k / 100
k = k * 185
k = k / 100
kx = (k dig 3) + "0"
ky = (k dig 2) + "0"
km = (k dig 1) + "0"
kn = (k dig 0) + "0"
pause 350
if np=1 then goto mens1

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

*****LATITUD N - S*****

```

SEROUT2 SIO,188,["!GPS",5]
SERIN2 SIO,188,[GRADOS,MINUTOS,I,J,X]
if np=1 then goto mens1
Ix= I*256
Ix = Ix /100
Ix = Ix * 60
vx = (Ix dig 3)
vy = (Ix dig 2)
vm = (Ix dig 1)
vn = (Ix dig 0)
GRA = GRADOS
MINU = MINUTOS
vx1 = vx
vy1 = vy
vm1 = vm
vn1 = vn
IF X = 0 THEN : X = "N"
IF X = 1 THEN : X = "S" : x1 = "-"
pause 1000
if np=1 then goto mens1

```

*****LONGITUD E - O *****

```

SEROUT2 SIO,188,["!GPS",6]
SERIN2 SIO,188,[GRADOS,MINUTOS,Ia,Ja,Xa]
Ix= Ia*256

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

$I_x = I_x / 100$

$I_x = I_x * 60$

$v_x = (I_x \text{ dig } 3)$

$v_y = (I_x \text{ dig } 2)$

$v_m = (I_x \text{ dig } 1)$

$v_n = (I_x \text{ dig } 0)$

IF X = 0 THEN : X = "E"

IF X = 1 THEN : X = "O" : x1 = "-"

pause 1000

if np=1 then goto mens1

xx = xx + 1

next

serout2 tx,84,["0001_", "n", "_", x1, dec GRA, "°", dec MINU, "", dec vx1, dec vy1, ".", dec vm1, dec vn1, "_", x1, dec GRADOS, "°", dec MINUTOS, "", dec vx, dec vy, ".", Dec vm, dec vn, "_", kx, ky, ".", km, kn, 26, 10, 13]

PAUSE 100

high portb.4

pause 500

low portb.4

goto INI3

;;;PANICO;;;

mens1:

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

xx = 0

for xx = 0 to 2

high portb.5

pause 300

low portb.5

pause 300

SEROUT2 SIO,188,["!GPS",8]

SERIN2 SIO,188,[I,J]

k = I * 256

k = I + J

k = k / 100

k = k * 185

k = k / 100

kx = (k dig 3) + "0"

ky = (k dig 2) + "0"

km = (k dig 1) + "0"

kn = (k dig 0) + "0"

pause 350

*****LATITUD N - S*****

SEROUT2 SIO,188,["!GPS",5]

SERIN2 SIO,188,[GRADOS,MINUTOS,I,J,X]

Ix= I*256

Ix = Ix /100

Ix = Ix * 60

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

vx = (Ix dig 3)

vy = (Ix dig 2)

vm = (Ix dig 1)

vn = (Ix dig 0)

GRA = GRADOS

MINU = MINUTOS

vx1 = vx

vy1 = vy

vm1 = vm

vn1 = vn

IF X = 0 THEN : X = "N"

IF X = 1 THEN : X = "S" : x1 = "-"

pause 350

,*****LONGITUD E - O *****

SEROUT2 SIO,188,["!GPS",6]

SERIN2 SIO,188,[GRADOS,MINUTOS,Ia,Ja,Xa]

Ix= Ia*256

Ix = Ix /100

Ix = Ix * 60

vx = (Ix dig 3)

vy = (Ix dig 2)

vm = (Ix dig 1)

vn = (Ix dig 0)

IF X = 0 THEN : X = "E"

IF X = 1 THEN : X = "O" : x1 = "-"

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

pause 350

xx = xx + 1

next

```
serout2 tx,84,["0001_", "p", "_", x1, dec GRA, "°", dec MINU, "", dec vx1, dec
vy1, ".", dec vm1, dec vn1, "_", x1, dec GRADOS, "°", dec MINUTOS, "", dec vx, dec
vy, ".", Dec vm, dec vn, "_", kx, ky, ".", km, kn, 26, 10, 13]
```

PAUSE 100

goto INI3

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

5.5. PRÁCTICA 5

Realizar un reloj atómico digital a través de un GPS y mostrar la hora en un GLCD

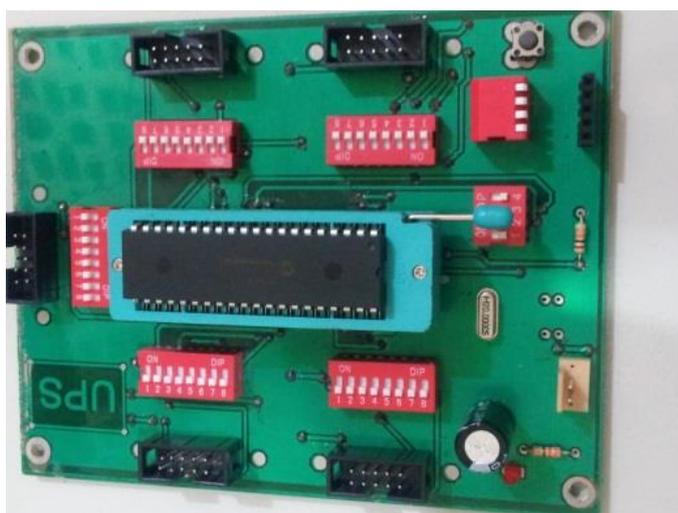
DATOS INFORMATIVOS

- ✓ MATERIA / CÁTEDRA RELACIONADA: Microcontroladores
- ✓ No. DE PRÁCTICA: 5
- ✓ NÚMERO DE ESTUDIANTES: 2
- ✓ NOMBRE DOCENTE: Ing. Luis Córdova
- ✓ TIEMPO ESTIMADO: 2 Horas

Prácticas a realizarse

Realizar un reloj atómico digital a través de un GPS y mostrar la hora en un GLCD

1. Entrenador Didáctico (Módulo de Control)



Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

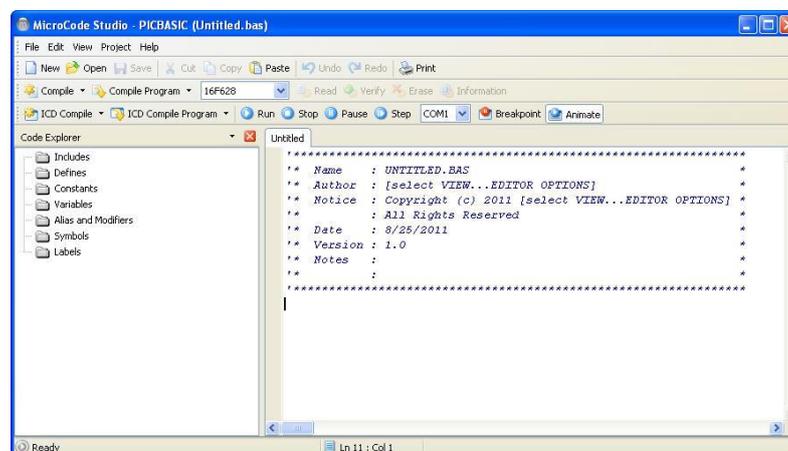
2. Módulo de GSM – GPRS – GPS



3. Display GLCD



4. Software Micro Code Studio



Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1	
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	Laboratorio de Electrónica Digital		
CARRERA	Ingeniería Electrónica		
SEDE	Guayaquil		

Materiales:

EQUIPO NECESARIO

- ✓ Entrenador Didáctico (Módulo de Control)
- ✓ Módulo de GSM – GPRS – GPS
- ✓ Display GLCD
- ✓ Software Micro Code Studio

MATERIAL DE APOYO

- ✓ Proyector
- ✓ Pizarra líquida

Marco Teórico:

A información requerida en esta práctica se puede encontrar revisando el CAPITULO 2, BIBLIOGRAFÍA.

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Procedimientos:

Práctica 5

Realizar un reloj atómico digital a través de un GPS y mostrar la hora en un GLCD

Para esta práctica utilizamos del entrenador didáctico, el módulo principal, en el cual ingresamos el programa al microcontrolador, para ello conectamos un cable de bus de dato suministrado en el entrenador didáctico, en el conector del PUERTO C del módulo de control principal hacia el conector del módulo GSM/GPRS/GPS, de igual forma desde el PUERTO B y PUERTO_D del módulo principal hacia los pines de conexión de la GLCD.

Energizamos la tarjeta principal con la fuente de voltaje +5Vcc incluida en el entrenador didáctico para de esta forma mediante los cables de bus de datos (pin 9, GND; pin 10, +5Vcc) conectados hacia los demás módulos utilizados puedan ser energizados.

Procedimos a swichear los pines de la siguiente manera:

- PUERTO_B:
 - Pin 1 (B0) → GLCD_128X64, Pin 4 (D/I)
 - Pin 2 (B1) → GLCD_128X64, Pin 5 (R/W)
 - Pin 3 (B2) → GLCD_128X64, Pin 6 (E)
 - Pin 4 (B3) → GLCD_128X64, Pin 15 (CS1)
 - Pin 5 (B4) → GLCD_128X64, Pin 16 (CS2)
 - Pin 6 (B5) → GLCD_128X64, Pin 17 (RESET)

- PUERTO_C:

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

- Pin 7 (C6, TX) → Módulo GSM/GPRS/GPS, Pin7 (RX)
- Pin 8 (C7, RX) → Módulo GSM/GPRS/GPS, Pin8 (TX)
- PUERTO_D:
 - Pin 1 (D0) → GLCD_128X64, Pin 7 (B0)
 - Pin 2 (D1) → GLCD_128X64, Pin 8 (B1)
 - Pin 3 (D2) → GLCD_128X64, Pin 9 (B2)
 - Pin 4 (D3) → GLCD_128X64, Pin 10 (B3)
 - Pin 5 (D4) → GLCD_128X64, Pin 11 (B4)
 - Pin 6 (D5) → GLCD_128X64, Pin 12 (B5)
 - Pin 7 (D6) → GLCD_128X64, Pin 13 (B6)
 - Pin 8 (D7) → GLCD_128X64, Pin 14 (B7)

Una vez configuradas las conexiones electrónicas, y energizado el circuito, se dio un lapsus de tiempo de 3 minutos, en cuanto se vió que el GPS captó la señal de más de 3 satélites (led indicador de señal estable del GPS) automáticamente el microcontrolador procesó la información obtenida de los satélites y capturó la hora universal UTC (que sus siglas en español significan Tiempo Universal Coordinado) para luego ser procesado y recalcular la hora local con nuestra zona horaria la cual es GMT -5:00, hora de Ecuador (GMT: Tiempo medio de Greenwich) y de esta forma se visualizó en la GLCD los siguientes datos:

HORA UTC:

20:42:25

HORA GTM -5:00:

15:42:25

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```
* Name      : Práctica Nro 05.BAS                *
* Author    : Franco Rafael - Motesdeoca Israel *
* Notice    : Copyright (c) 2014 UPS-Guayaqui (Tesis de Grado) *
*           : All Rights Reserved                ** Date   : 16/02/2014
*
* Version   : 1.0                               *
* Notes    :                                     *
*           :                                     *
```

```
include "modedefs.bas"
```

```
DEVICE 18F4550
```

```
XTAL=20
```

```
***** configuracion puerto serie fisico *****
```

```
HSERIAL_BAUD=9600
```

```
HSERIAL_RCSTA=%10010000
```

```
HSERIAL_TXSTA=%00100000
```

```
HSERIAL_CLEAR=ON
```

```
***** configuracion puerto analogico/digital ***
```

```
DECLARE ADIN_RES 8
```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```
DECLARE ADIN_TAD frc
DECLARE ADIN_DELAY 50
TRISA=%00000111
ADCON1=%00000000
```

```
***** CONFIGURACION RESTO DE PUERTOS *****
```

```
INPUT PORTB.2
INPUT PORTB.1
INPUT PORTB.0
```

```
***** DS1620 control pins *****
```

```
SYMBOL DQ= PORTB.5 ' DS1620 DQ pin #1
SYMBOL CLK0=PORTB.4 ' DS1620 CLK pin #2
SYMBOL RST=PORTB.3 ' DS1620 RST pin #3
DIM decimal AS BYTE
DIM Temp AS WORD
```

```
'===== variables temperatura =====
```

```
DIM lecturaAn0 AS BYTE
DIM mv AS FLOAT
DIM temperatura AS FLOAT
temperatura=0
mv=0
```

```
'===== variables rtc =====
```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

DIM SEGUNDOS AS BYTE

DIM MINUTOS AS BYTE

DIM HORA AS BYTE

DIM dia AS BYTE

DIM fecha AS BYTE

DIM mes AS BYTE

DIM ano AS BYTE

Init:

HBUSOUT % 11010000,7,[%00010000] 'int externa 1hz

HBUSOUT % 11010000,0,[0] 'inicialisa ds1

DELAYMS 40

nopres:

HIGH PORTC.0

IF PORTB.2=0 THEN

Loadtime:

DELAYMS 40

SEGUNDOS=HRSIN ' Read the Seconds serially

MINUTOS=HRSIN ' Read the Minutes serially

HORA=HRSIN ' Read the Hours serially

fecha=HRSIN ' Read the Date serially

mes=HRSIN ' Read the Month serially

ano=HRSIN ' Read the Year serially

SetTime:

HBUSOUT \$D0,\$00,[SEGUNDOS,MINUTOS,HORA,dia,fecha,mes,ano] ' write
time

GOTO main

ENDIF

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

GOTO nopres

'===== rutina principal ====='

main:

LOW PORTC.0

WHILE 1=1

GOSUB leerMsp700

GOSUB Read_1620

GOSUB GetTime

HRSOUT "time", " ", HEX2 HORA, ":", HEX2 MINUTOS, ":", HEX2 SEGUNDOS,

","data", " ", HEX2, fecha, "/", HEX2, mes, "/", HEX2 ano, 10,13

HRSOUT "*****", 0,10,13

HRSOUT "temperatura_msp700=", " ", DEC2 temperatura, "c", 0,10,13

HRSOUT "*****", 0,10,13

HRSOUT "temperatura_ds1620=", " ", DEC Temp, ".", DEC decimal, 0,10,13

HRSOUT "*****", 0,10,13

GOSUB piloto

mv=0

temperatura=0

WEND

' ***** mcp9700 *****'

leerMsp700:

lecturaAn0=ADIN 0

mv=(lecturaAn0 * .0195)-.400

temperatura=mv/.0195

RETURN

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Read_1620:

```

RST = 1
SHOUT DQ,CLK0,LSBFIRST,[$0C,$02] ' Continuous convert, CPU mode
RST = 0
DELAYMS 10 ' Minimum wait time after write
HIGH RST ' Enable 1620
SHOUT DQ, CLK0, LSBFIRST, [$EE] ' Send start temp convert command
LOW RST ' Disable 1620
DELAYMS 1000 ' Wait for conversion to complete

HIGH RST ' Enable 1620
SHOUT DQ, CLK0, LSBFIRST, [AA] ' Send read temp command
SHIN DQ, CLK0, LSBPRE, [Temp\9] ' Read 9 bit temperature
LOW RST
IF Temp.7=0 THEN
decimal=5
ELSE
decimal=0
ENDIF ' Disable 1620
Temp = Temp/2 ' Scale reading to whole degrees C.
RETURN

```

piloto:

```

HIGH PORTC.2
DELAYMS 100
LOW PORTC.2

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

DELAYMS 100

RETURN

'===== leer la hora actual ====='

GetTime:

HBUSIN \$D1,\$00,[SEGUNDOS,MINUTOS,HORA,dia,fecha,mes,ano]

RETURN

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

5.6. PRÁCTICA 6

Mediante un circuito digital y a través del módulo GPS mostrar la ubicación en coordenadas geográficas a través de la pantalla GLCD.

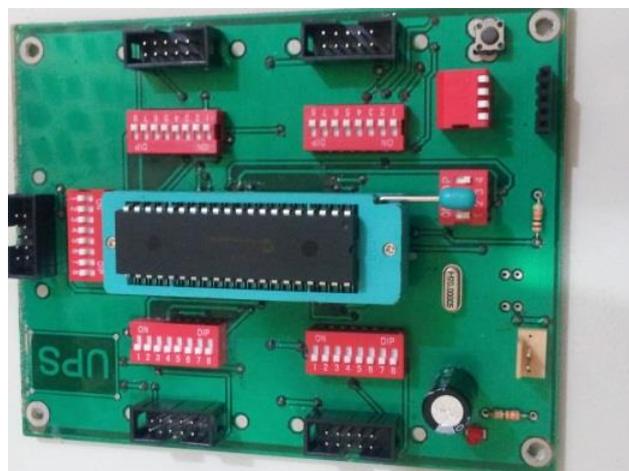
DATOS INFORMATIVOS

- ✓ MATERIA / CÁTEDRA RELACIONADA: Microcontroladores
- ✓ No. DE PRÁCTICA: 6
- ✓ NÚMERO DE ESTUDIANTES: 2
- ✓ NOMBRE DOCENTE: Ing. Luis Córdova
- ✓ TIEMPO ESTIMADO: 2 Horas

Prácticas a realizarse:

Mediante un circuito digital y a través del módulo GPS mostrar la ubicación en coordenadas geográficas a través de la pantalla GLCD.

1. Entrenador Didáctico (Módulo de Control)



Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

2. Módulo de GSM – GPRS – GPS



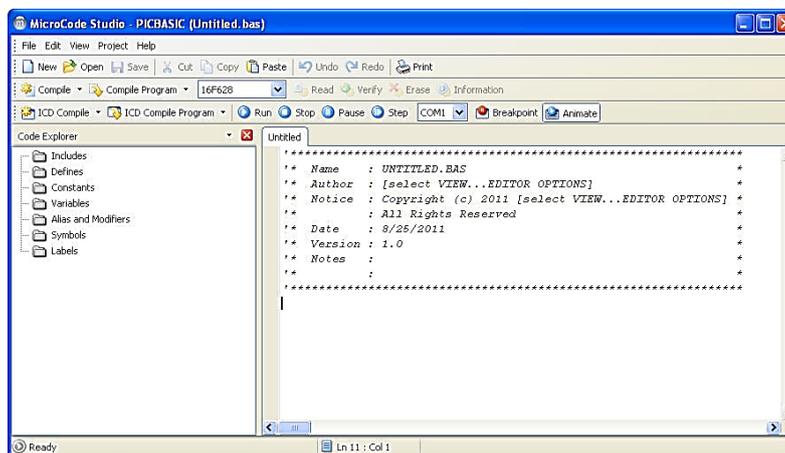
3. Display GLCD



Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

4. Software Micro Code Studio



Materiales:

EQUIPO NECESARIO

- ✓ Entrenador Didáctico (Módulo de Control)
- ✓ Módulo de GSM – GPRS – GPS
- ✓ Display GLCD
- ✓ Software Micro Code Studio

MATERIAL DE APOYO

- ✓ Proyector
- ✓ Pizarra líquida

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Marco Teórico:

A información requerida en esta práctica se puede encontrar revisando el CAPITULO 2, BIBLIOGRAFÍA.

Procedimientos:

Práctica 6

Mediante un circuito digital y a través del módulo GPS mostrar la ubicación en coordenadas geográficas a través de la pantalla GLCD

Se utilizó el entrenador didáctico, el módulo principal, en el cual ingresamos el programa al microcontrolador, para ello conectamos un cable de bus de dato en el entrenador didáctico, en el conector del PUERTO C del módulo de control principal hacia el conector del módulo GSM/GPRS/GPS, de igual forma desde el PUERTO B y PUERTO_D del módulo principal hacia los pines de conexión de la GLCD.

Energizamos la tarjeta principal con la fuente de voltaje +5Vcc incluida en el entrenador didáctico para de esta forma mediante los cables de bus de datos (pin 9, GND; pin 10, +5Vcc) conectados hacia los demás módulos utilizados puedan ser energizados.

Procedimos a swichear los pines:

- PUERTO_B:
 - Pin 1 (B0) → GLCD_128X64, Pin 4 (D/I)
 - Pin 2 (B1) → GLCD_128X64, Pin 5 (R/W)

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

- Pin 3 (B2) → GLCD_128X64, Pin 6 (E)
- Pin 4 (B3) → GLCD_128X64, Pin 15 (CS1)
- Pin 5 (B4) → GLCD_128X64, Pin 16 (CS2)
- Pin 6 (B5) → GLCD_128X64, Pin 17 (RESET)

- PUERTO_C:

- Pin 7 (C6, TX) → Módulo GSM/GPRS/GPS, Pin7 (RX)
- Pin 8 (C7, RX) → Módulo GSM/GPRS/GPS, Pin8 (TX)

- PUERTO_D:

- Pin 1 (D0) → GLCD_128X64, Pin 7 (B0)
- Pin 2 (D1) → GLCD_128X64, Pin 8 (B1)
- Pin 3 (D2) → GLCD_128X64, Pin 9 (B2)
- Pin 4 (D3) → GLCD_128X64, Pin 10 (B3)
- Pin 5 (D4) → GLCD_128X64, Pin 11 (B4)
- Pin 6 (D5) → GLCD_128X64, Pin 12 (B5)
- Pin 7 (D6) → GLCD_128X64, Pin 13 (B6)
- Pin 8 (D7) → GLCD_128X64, Pin 14 (B7)

Una vez configuradas las conexiones electrónicas, y energizado el circuito, se dio un intervalo de tiempo de 10 minutos aproximadamente, en cuanto se vió que el GPS captó la señal de más de 3 satélites (led indicador de señal estable del GPS) utilizando el método de triangulación que consiste en el uso de la trigonometría para determinar posiciones de puntos o medidas de distancias, como en este caso el GPS obtiene las distancias de cada una de las 3 señales de los satélites respecto al punto de medición, conocidas estas tres distancias fácilmente se conoce la posición relativa respecto a los 3 satélites, así como la coordenada y posición de cada uno de ellos y por consiguiente se obtiene la posición absoluta o coordenadas reales del punto de medición con un rango de error de ± 8 metros a la redonda.

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Automáticamente el microcontrolador procesó la información obtenida de los satélites y capturó las coordenadas, de esta forma se visualizó en la GLCD:

LATITUD:

2° 08' 07.4''S

LONGITUD:

79° 54' 15.4'' W

```
* Name   : Práctica Nro 06.BAS           *
* Author : Franco Rafael - Motesdeoca Israel      *
* Notice : Copyright (c) 2014 UPS-Guayaqui (Tesis de Grado) *
*       : All Rights Reserved                *
* Date   : 17/05/2014                       *
* Version : 1.0                             *
* Notes  :                                  *
*       :                                    *
```

```
include "modedefs.bas"
```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

*****DEFINICION DE COMUNICACION*****

```
tx      var PORTB.2
rx      var PORTB.1
SIO     VAR PORTA.0
np      var PORTA.2
LED     VAR PORTA.4
```

*****VARIABLES DE SERIAL*****

```
v       var word
vx      var word
vy      var word
vm      var word
vn      var word
vx1     var word
vy1     var word
vm1     var word
vn1     var word
x1      var byte
k       var byte
kx      var byte
ky      var byte
km      var byte
kn      var byte
np1     var byte
dato_osa VAR BYTE [2]
Ia      var byte
Ja      var byte
```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Xa var byte

;*****VARIABLES DE LCD*****

```

DEFINE LCD_DREG  PORTB
DEFINE LCD_DBIT  4
DEFINE LCD_RSREG PORTB
DEFINE LCD_RSBIT 0
DEFINE LCD_EREG  PORTB
DEFINE LCD_EBIT  3

```

TRISB=%00000010

TRISA=%01100

;*****VARIABLES DE SALIDA DEL PIC*****

```

HIGH  SIO
HIGH  LED
CMCON = 7
OPTION_REG.6 = 0
OPTION_REG.7 = 0
PAUSE 1000

```

;*****

```

high RX
high tx
PAUSE 100
low LED
low portb.4
low portb.5

```

lcdout \$FE,1 ; PRESENTACIÓN DE MENSAJES

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

lcdout \$FE,\$80, " BIENVENIDOS... "

lcdout \$FE,\$C0, " GPS - ROUTER "

PAUSE 2000

HIGH LED

INI3:

if np=1 then goto mens1

xx = 0

for xx = 0 to 10

if np=1 then goto mens1

high portb.5

pause 300

low portb.5

pause 300

if np=1 then goto mens1

SERIN2 SIO,188,[I,J]

k = I * 256

k = I + J

k = k / 100

k = k * 185

k = k / 100

kx = (k dig 3) + "0"

ky = (k dig 2) + "0"

km = (k dig 1) + "0"

kn = (k dig 0) + "0"

pause 350

if np=1 then goto mens1

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

pause 1000
if np=1 then goto mens1

```

```

pause 1000
if np=1 then goto mens1
xx = xx + 1
next

```

```

serout2 tx,84,["0001_", "n", "_", x1, dec GRA, "°", dec MINU, "", dec vx1, dec
vy1, ".", dec vm1, dec vn1, "_", x1, dec GRADOS, "°", dec MINUTOS, "", dec vx, dec
vy, ".", Dec vm, dec vn, "_", kx, ky, ".", km, kn, 26, 10, 13]

```

```

PAUSE 100
high portb.4
pause 500
low portb.4

```

```
goto INI3
```

```
mens1:
```

```

xx = 0
for xx = 0 to 2
high portb.5
pause 300
low portb.5
pause 300

```

```
SEROUT2 SIO,188,["",8]
```

```
SERIN2 SIO,188,[I, J]
```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1	
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	Laboratorio de Electrónica Digital		
CARRERA	Ingeniería Electrónica		
SEDE	Guayaquil		

pause 350

SEROUT2 SIO,188,["",5]

SERIN2 SIO,188,[x, Y]

pause 350

goto INI3

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

5.7. PRÁCTICA 7

Realizar un circuito para el manejo de matriz de led usando el PIC 18F4550

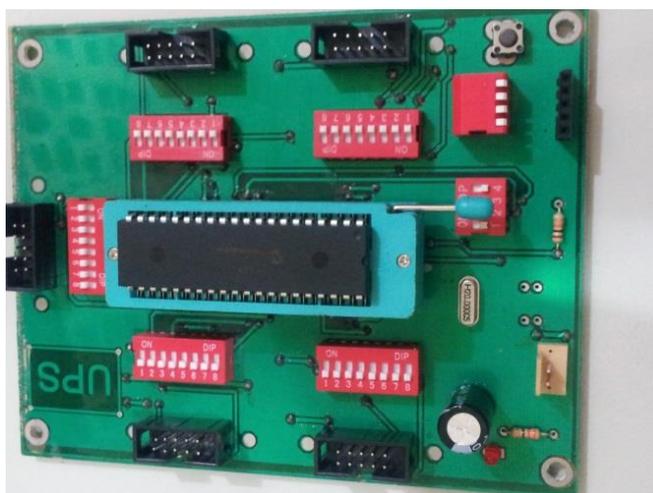
DATOS INFORMATIVOS

- ✓ MATERIA / CÁTEDRA RELACIONADA: Microcontroladores
- ✓ No. DE PRÁCTICA: 7
- ✓ NÚMERO DE ESTUDIANTES: 2
- ✓ NOMBRE DOCENTE: Ing. Luis Córdova
- ✓ TIEMPO ESTIMADO: 2 Horas

Prácticas a realizarse:

Realizar un circuito para el manejo de matriz de led usando el PIC 18F4550

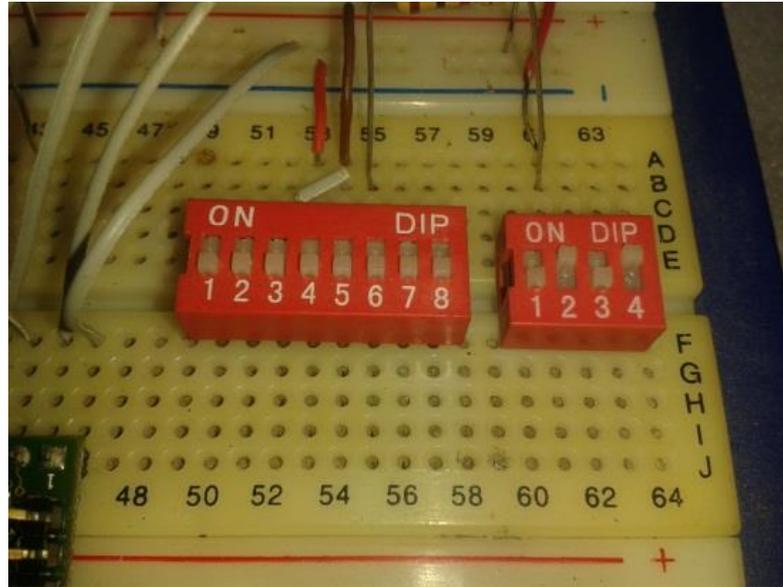
1. Entrenador Didáctico (Módulo de Control)



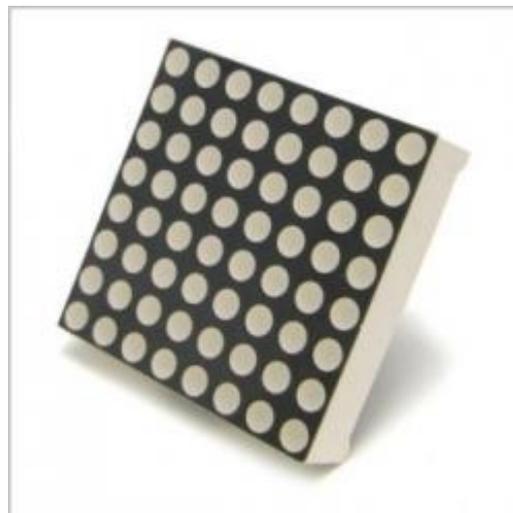
Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

2. Protoboard



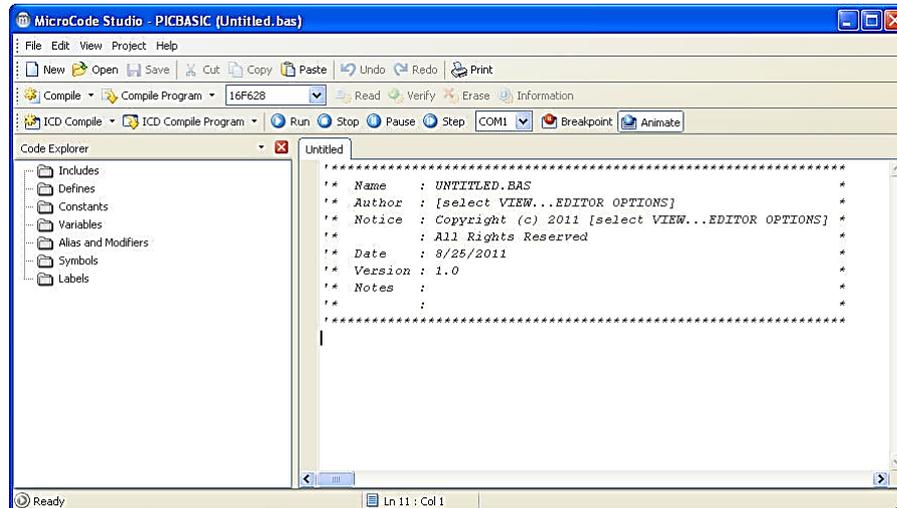
3. Matriz de LED



Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

4. Software Micro Code Studio



Materiales:

EQUIPO NECESARIO

- ✓ Entrenador Didáctico (Módulo de Control)
- ✓ Protoboard
- ✓ Matriz de LED
- ✓ Software Micro Code Studio

MATERIAL DE APOYO

- ✓ Proyector
- ✓ Pizarra líquida

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Marco Teórico:

A información requerida en esta práctica se puede encontrar revisando el CAPITULO 2, BIBLIOGRAFÍA.

Procedimientos:

Práctica 7

Realizar un circuito para el manejo de matriz de led usando el PIC 18F4550

Se utilizó el entrenador didáctico, el módulo principal, en el cual ingresamos el programa al microcontrolador, para ello conectamos un cable de bus de dato en el entrenador didáctico, en el conector del PUERTO_A hacia el panel de matrices de LED previamente construido el cual incluye los drivers multiplexores 74LS137 (para cada una de las matrices) para esta práctica, de igual forma el PUERTO_B.

Cabe recalcar que el PUERTO_A sirve para activar las columnas del panel de LEDs y el PUERTO_B las filas del mismo

Energizamos el módulo principal con la fuente de voltaje +5Vcc incluida en el entrenador didáctico para de esta forma mediante los cables de bus de datos (pin 9, GND; pin 10, +5Vcc) conectados hacia el panel de matrices de LED pueda ser energizado.

Procedimos a swichear los pines de la siguiente manera:

- PUERTO_A:
 - Pin 1 (A0) → PANEL MATRICES, PORT_A, Pin1(74LS137, A)

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

- Pin 1 (A1) → PANEL MATRICES, PORT_A, Pin2(74LS137, B)
- Pin 1 (A1) → PANEL MATRICES, PORT_A, Pin3(74LS137, C)

- PUERTO_B:

- Pin 1 (B0) → PANEL MATRICES, PORT_B, Pin1(A)
- Pin 2 (B1) → PANEL MATRICES, PORT_B, Pin2(B)
- Pin 3 (B2) → PANEL MATRICES, PORT_B, Pin3(C)
- Pin 4 (B3) → PANEL MATRICES, PORT_B, Pin4(D)
- Pin 5 (B4) → PANEL MATRICES, PORT_B, Pin5(E)
- Pin 6 (B5) → PANEL MATRICES, PORT_B, Pin6(F)
- Pin 7 (B6) → PANEL MATRICES, PORT_B, Pin7(G)
- Pin 8 (B7) → PANEL MATRICES, PORT_B, Pin8(H)

Una vez configuradas las conexiones electrónicas, y energizado el circuito, se verificó que se muestra en el panel de la matriz de LED el siguiente mensaje:

UNIVERSIDAD POLITÉCNICA SALESIANA

Este mensaje se mostró en forma de barrido en el panel de LEDs.

```

*****
* Name   : Práctica Nro 07.BAS                *
* Author : Franco Rafael - Motesdeoca Israel  *
* Notice : Copyright (c) 2014 UPS-Guayaqui (Tesis de Grado) *
*       : All Rights Reserved                *
* Date   : 17/05/2014                        *

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

* Version : 1.0 *

* Notes : *

* : *

DEFINE OSC 4

LINEA var byte[32]

CHAR VAR BYTE[7]

CHAR_ACT VAR BYTE ' puntero de caracter

CHAR_LENGTH_ACT VAR BYTE 'puntero de la longitud del caracter

CHAR_LENGTH var byte ' longitud de la longitud

Caracter var byte

Counter1 var byte

Counter2 var byte

Counter3 var byte

Counter4 var byte

'declaracion de alias

DATOS var PORTA.0

CLOCK var PORTA.1

'Inicializacion de los datos

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

TRISB = 0

TRISA = 0

linea[1]=\$FF

linea[2]=\$FF

linea[3]=\$FF

linea[4]=\$FF

linea[5]=\$FF

linea[6]=\$FF

linea[7]=\$FF

linea[8]=\$FF

linea[9]=\$FF

linea[10]=\$FF

linea[11]=\$FF

linea[12]=\$FF

linea[13]=\$FF

linea[14]=\$FF

linea[15]=\$FF

linea[16]=\$FF

linea[17]=\$FF

linea[18]=\$FF

linea[19]=\$FF

linea[20]=\$FF

linea[21]=\$FF

linea[22]=\$FF

linea[23]=\$FF

linea[24]=\$FF

linea[25]=\$FF

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

linea[26]=\$FF

linea[27]=\$FF

linea[28]=\$FF

linea[29]=\$FF

linea[30]=\$FF

linea[31]=\$FF

linea[32]=\$FF

char_act=0

counter3=0

counter4=0

counter2=1

char_length_act = 1

low clock

low datos

DATA @0,"UNIVERSIDAD POLITECNICA SALESIANA"

read char_act,caracter

gosub tabla_char

'inicio del programa

Inicio:

MENSAJE:

linea[1]=char[char_length_act]

IF CHAR_LENGTH_act = char_length then

char_act = char_act + 1

Mens_Salto1:

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

read char_act,caracter
if caracter=$FF then
    char_act=0
    goto mens_salto1
endif
gosub tabla_char
char_length_act=0
endif

```

Img:

```
counter1=0
```

LineH:

```

PORTB = $FF
counter1 = counter1 + 1    'envia LA ACTIVACION del registro
if counter1 = counter2 then 'serial paralelo
    gosub send_data0      'para activar la linea vertical
else
    gosub send_data1
endif
if counter1 = 32 then goto LineV
goto lineh

```

LineV:

```

PORTB = LINEA[COUNTER2]  'determina la posicion activa horizontal
pauseus 500             'y genera la activacion de los leds verticales
if counter2 = 32 then
    counter2 = 1
    gosub send_data0
ELSE

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

counter2 = counter2 + 1
gosub send_data1
endif

```

Shift_Left:

```

if counter3 = 35 then
counter3 = 0
COUNTER4 = COUNTER4 + 1
if counter4 = 2 then ' 250 del counter3 *4 del

linea[32]=linea[31]'
linea[31]=linea[30]'
linea[30]=linea[29]'counter4 es aprox 0.5seg
linea[29]=linea[28]'si hay mas lineas
linea[28]=linea[27]'hay q agrandar losvectores
linea[27]=linea[26]'y cambiar por un for
linea[26]=linea[25]
linea[25]=linea[24]
linea[24]=linea[23]
linea[23]=linea[22]'counter4 es aprox 0.5seg
linea[22]=linea[21]'si hay mas lineas
linea[21]=linea[20]
linea[20]=linea[19]'y cambiar por un for
linea[19]=linea[18]
linea[18]=linea[17]
linea[17]=linea[16]

linea[16]=linea[15]'

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

linea[15]=linea[14]'counter4 es aprox 0.5seg
linea[14]=linea[13]'si hay mas lineas
linea[13]=linea[12]'hay q agrandar losvectores
linea[12]=linea[11]'y cambiar por un for
linea[11]=linea[10]
linea[10]=linea[9]
linea[9]=linea[8]
linea[8]=linea[7]'counter4 es aprox 0.5seg
linea[7]=linea[6]'si hay mas lineas
linea[6]=linea[5]'hay q agrandar los vectores
linea[5]=linea[4]'y cambiar por un for
linea[4]=linea[3]
linea[3]=linea[2]
linea[2]=linea[1]
counter4 = 0
char_length_act = char_length_act+1
GOTO INICIO

```

```
endif
```

```
ELSE
```

```
counter3 = counter3 + 1
```

```
endif
```

```
goto IMG
```

```
SEND_DATA0:
```

```
low datos
```

```
pauseus 2
```

```
low clock
```

```
pauseus 2
```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

high clock

RETURN

SEND_DATA1:

HIGH DATOS

pauseus 2

low clock

pauseus 2

high clock

RETURN

'tabla de seleccion de caracter

TABLA_CHAR:

select case caracter

case " "

char_LENGTH=5

char[5]=\$FF

char[4]=\$FF

char[3]=\$FF

char[2]=\$FF

char[1]=\$FF

case "C"

char_LENGTH=5

char[5]=\$FF

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

char[4]=% 1011101

char[3]=%0111110

char[2]=%0111110

char[1]=% 1000001

case "R"

char_LENGTH=5

char[5]=\$FF

char[4]=% 1001000

char[3]=%0110111

char[2]=%0110111

char[1]=% 0000000

case "I"

char_LENGTH=4

char[4]=\$FF

char[3]=%0111110

char[2]=% 0000000

char[1]=%0111110

case "T"

char_LENGTH=5

char[5]=\$FF

char[4]=%0111111

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

char[3]=%0000000

char[2]=%0000000

char[1]=%0111111

case "O"

char_LENGTH=6

char[6]=\$FF

char[5]=%1000001

char[4]=%0111110

char[3]=%0111110

char[2]=%0111110

char[1]=%1000001

.....caracteres del A a la Z + numeros

case else

char_LENGTH=5

CHAR[1]=\$FF

CHAR[2]=\$FF

CHAR[3]=\$FF

CHAR[4]=\$FF

CHAR[5]=\$FF

end select

return

END

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

5.8. PRÁCTICA 8

Mostrar gráficamente en la GLCD la gráfica de una ecuación de primer orden dado dos puntos de una recta ingresados por teclado.

DATOS INFORMATIVOS

- ✓ MATERIA / CÁTEDRA RELACIONADA: Microcontroladores
- ✓ No. DE PRÁCTICA: 8
- ✓ NÚMERO DE ESTUDIANTES: 2
- ✓ NOMBRE DOCENTE: Ing. Luis Córdova
- ✓ TIEMPO ESTIMADO: 2 Horas

Prácticas a realizarse:

Mostrar gráficamente en la GLCD la gráfica de una ecuación de primer orden dado dos puntos de una recta ingresados por teclado.

1. Entrenador Didáctico (Módulo de Control)



Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

2. Teclado 4x3



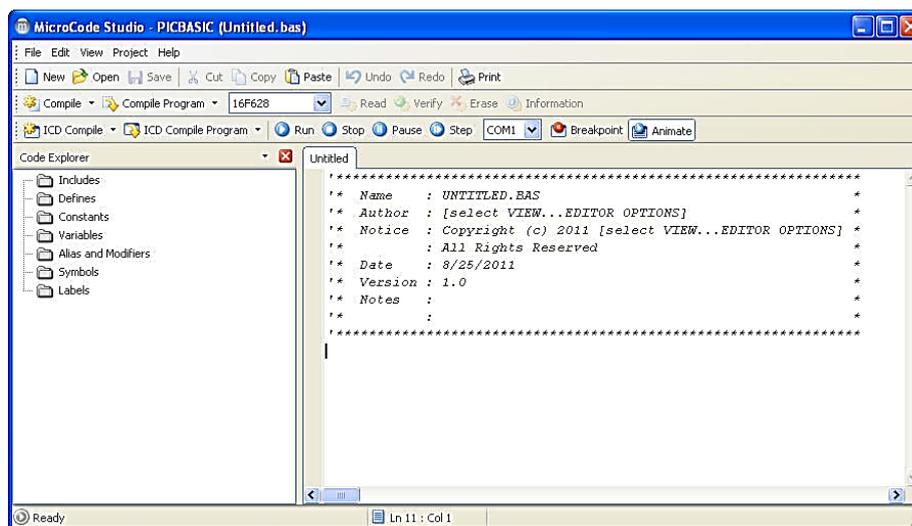
3. Display GLCD



Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

4. Software Micro Code Studio



Materiales:

EQUIPO NECESARIO	MATERIAL DE APOYO
------------------	-------------------

- | | |
|---|--|
| <ul style="list-style-type: none"> ✓ Entrenador Didáctico (Módulo de Control) ✓ Teclado 4x3 ✓ Display GLCD ✓ Software Micro Code Studio | <ul style="list-style-type: none"> ✓ Proyector ✓ Pizarra líquida |
|---|--|

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Marco Teórico:

A información requerida en esta práctica se puede encontrar revisando el CAPITULO 2, BIBLIOGRAFÍA.

Procedimientos:

Práctica 8

Mostrar gráficamente en la GLCD la gráfica de una ecuación de primer orden dados dos puntos de una recta ingresados por teclado

Para esta práctica utilizamos del entrenador didáctico, el módulo principal, en el cual ingresamos el programa al microcontrolador, para ello conectamos un cable de bus de dato suministrado en el entrenador didáctico, en el conector del PUERTO_C del módulo de control principal hacia los pines del teclado matricial, de igual forma desde el PUERTO B y PUERTO_D del módulo principal hacia los pines de conexión de la GLCD.

Energizamos la tarjeta principal con la fuente de voltaje +5Vcc incluida en el entrenador didáctico para de esta forma mediante los cables de bus de datos (pin 9, GND; pin 10, +5Vcc) conectados hacia los pines respectivos del teclado y la GLCD..

Procedimos a swichear los pines de la siguiente manera:

- PUERTO_B:
 - Pin 1 (B0) → GLCD_128X64, Pin 4 (D/I)

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

- Pin 2 (B1) → GLCD_128X64, Pin 5 (R/W)
- Pin 3 (B2) → GLCD_128X64, Pin 6 (E)
- Pin 4 (B3) → GLCD_128X64, Pin 15 (CS1)
- Pin 5 (B4) → GLCD_128X64, Pin 16 (CS2)
- Pin 6 (B5) → GLCD_128X64, Pin 17 (RESET)

- PUERTO_C:

- Pin 1(C0) → TECLADO4X3, Pin 1(Columna 1)
- Pin 2(C1) → TECLADO4X3, Pin 2(Columna 2)
- Pin 3(C2) → TECLADO4X3, Pin 2(Columna 3)
- Pin 5(C4) → TECLADO4X3, Pin A(Fila A)
- Pin 6(C5) → TECLADO4X3, Pin B(Fila B)
- Pin 7(C6) → TECLADO4X3, Pin C(Fila C)
- Pin 8(C7) → TECLADO4X3, Pin D(Fila D)

- PUERTO_D:

- Pin 1 (D0) → GLCD_128X64, Pin 7 (B0)
- Pin 2 (D1) → GLCD_128X64, Pin 8 (B1)
- Pin 3 (D2) → GLCD_128X64, Pin 9 (B2)
- Pin 4 (D3) → GLCD_128X64, Pin 10 (B3)
- Pin 5 (D4) → GLCD_128X64, Pin 11 (B4)
- Pin 6 (D5) → GLCD_128X64, Pin 12 (B5)
- Pin 7 (D6) → GLCD_128X64, Pin 13 (B6)
- Pin 8 (D7) → GLCD_128X64, Pin 14 (B7)

Una vez configuradas las conexiones electrónicas, y energizado el circuito, se procedió a ingresar los datos de las variables para la ecuación de primer orden mediante el teclado matricial ingresando los siguientes datos.

$$Ax+By+C = 0$$

Donde:

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

- A = 2
- B = 3
- C = 1

La GLCD muestra la siguiente gráfica

```

*****
'* Name   : Práctica Nro 08.BAS           *
'* Author : Franco Rafael - Motesdeoca Israel *
'* Notice : Copyright (c) 2014 UPS-Guayaqui (Tesis de Grado) *
'*       : All Rights Reserved           *
'* Date   : 03/08/2014                   *
'* Version : 1.0                         *
'* Notes  :                               *
'*       :                               *
*****

```

```

/*****

```

```

    libreria para control de glcd con controlador

```

```

    st7920 de 128x64

```

```

*****/

```

```

#define MIPS 30

```

```

//definimos los bits de control

```

```

#define rw LATDbits.LATD2

```

```

#define di LATCbits.LATC14

```

```

#define e LATCbits.LATC13

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1	
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	Laboratorio de Electrónica Digital		
CARRERA	Ingeniería Electrónica		
SEDE	Guayaquil		

```
#define rst LATFbits.LATF5
```

```
//definimos los bits de datos
```

```
#define b0 LATEbits.LATE0
```

```
#define b1 LATEbits.LATE1
```

```
#define b2 LATEbits.LATE2
```

```
#define b3 LATEbits.LATE3
```

```
#define b4 LATEbits.LATE4
```

```
#define b5 LATEbits.LATE5
```

```
#define b6 LATEbits.LATE8
```

```
#define b7 LATFbits.LATF0
```

```
//variables de sistema
```

```
#define DI_FUNCTION 0
```

```
#define DI_DATA 1
```

```
#define RW_WRITE 0
```

```
#define RW_READ 1
```

```
#define nop()
```

```
typedef union data {
```

```
    int reg;
```

```
    struct {
```

```
        unsigned d0 :1;
```

```
        unsigned d1 :1;
```

```
        unsigned d2 :1;
```

```
        unsigned d3 :1;
```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

unsigned d4 :1;
unsigned d5 :1;
unsigned d6 :1;
unsigned d7 :1;
};

```

```

} data ;

```

```

data DATA;

```

```

typedef union
{
int word;
char byte[2];
} Dots;

```

```

typedef struct
{
int refrescar :1;
Dots dots[32][12]; // Dimensiones máximas del display (x,y) = (191,31).
} GDRAM; // El punto (0,0) corresponde a la esquina superior izquierda.

```

```

GDRAM gdram;

```

```

void set_output(){
TRISE=0x00;
TRISF=0x00;
TRISC=0x00;

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

    TRISD=0x00;
}

void set_input()
{
    TRISE=0xff;
    TRISFbits.TRISF0=0x01;
}

/*****/
/*   DelayUs Microsegundos      */
/*****/

void DelayUs(int v) // 40 MIPS
{
    asm ("REPEAT #v\n NOP" :: "i" (MIPS-16));
    while ((v--)!=1) {
        asm ("REPEAT #v\n NOP" :: "i" (MIPS-7));
    };
}

/*****/
/*   DelayMs MiliSegundos      */
/*****/

void DelayMs(int v)
{
    //while (v != 0){ DelayUs(1000); v--;}
    asm ("REPEAT #v\n NOP" :: "i" (MIPS*250-5));
    asm ("REPEAT #v\n NOP" :: "i" (MIPS*250-5));
}

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```
asm ("REPEAT #0\n NOP" :: "i" (MIPS*250-6));
asm ("REPEAT #0\n NOP" :: "i" (MIPS*250-6));
while ((v--)!=1) {
    asm ("REPEAT #0\n NOP" :: "i" (MIPS*250-4));
    asm ("REPEAT #0\n NOP" :: "i" (MIPS*250-3));
    asm ("REPEAT #0\n NOP" :: "i" (MIPS*250-3));
    asm ("REPEAT #0\n NOP" :: "i" (MIPS*250-3));
};
}
```

```
void put_data(data Data)
```

```
{
    b0=Data.d0;
    b1=Data.d1;
    b2=Data.d2;
    b3=Data.d3;
    b4=Data.d4;
    b5=Data.d5;
    b6=Data.d6;
    b7=Data.d7;
}
```

```
void glcd_readByte (char address,data Data)
```

```
{
    set_input(); // Set PORTB to input
    di = address;
    nop();
    rw = RW_READ; // Set for reading
```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

nop();
e = 1;    // Pulse the enable pin
nop();
Data.d0 = b0;    // Get the data from the display's output register
Data.d1 = b1;    // Get the data from the display's output register
Data.d2 = b2;    // Get the data from the display's output register
Data.d3 = b3;    // Get the data from the display's output register
Data.d4 = b4;    // Get the data from the display's output register
Data.d5 = b5;    // Get the data from the display's output register
Data.d6 = b6;    // Get the data from the display's output register
Data.d7 = b7;    // Get the data from the display's output register
e = 0;
}

void glcd_writeByte (char address, data Data)
{
    //while ( bit_test (glcd_readByte(RS_FUNCTION), 7) ); // Wait Busy Flag =
FALSE!
    set_output(); // Set PORTB to output
    di = address;
    DelayUs(100);
    rw = RW_WRITE;    // Set for writing
    DelayUs(100);
    e = 0;
    put_data(Data);    // Put the data on the port
    DelayUs(100);
    e = 1;    // Pulse the enable pin
    DelayUs(100);

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

DelayMs(10);
e = 0;
}

void glcd_update ()
{
int v, h;

if (gdram.refrescar)
{
for (v=0; v<32; v++)
{
DATA.reg=0x80 | v;
glcd_writeByte (DI_FUNCTION,DATA); // Set Vertical Address.
DATA.reg=0x80 | 0;
glcd_writeByte (DI_FUNCTION,DATA ); // Set Horizontal Address.

for (h=0; h<12; h++)
{
DATA.reg=gdram.dots[v][h].byte[1];
glcd_writeByte (DI_DATA,DATA ); // Write High Byte.
DATA.reg=gdram.dots[v][h].byte[0];
glcd_writeByte (DI_DATA, DATA); // Write Low Byte.
}
}
gdram.refrescar = 0;
}
}

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

void glcd_fillScreen (char color)
{
    int v, h;
    int d;
    d = (color == 1 ? 0xFFFFL : 0x0000L);
    for (v=0; v<32; v++)
    {
        for (h=0; h<12; h++)
        {
            gdram.dots[v][h].word = d;
        }
    }
    gdram.refrescar = 1;
}

void GLCD_Init(){
    set_output();
    rst=1;
    di=DI_FUNCTION;
    rw=RW_WRITE;
    e=0;
    rst=0;
    DelayUs(10);
    rst=1;
    DelayMs(50);
    DATA.reg=0x30;
    glcd_writeByte (DI_FUNCTION, DATA); // Specify 8 bit interface and basic
instruction set.
    DelayMs (115);

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

    glcd_writeByte (DI_FUNCTION, DATA); // Specify 8 bit interface and basic
instruction set.

    DelayMs (100);
    DATA.reg=0x0F; //0c
    glcd_writeByte (DI_FUNCTION, DATA); // Specify Display on, Cursor off and
Blink off.

    DelayMs (100);
    DATA.reg=0x01;
    glcd_writeByte (DI_FUNCTION, DATA); // Display clear.
    DelayMs (100);
    DATA.reg=0x06;
    glcd_writeByte (DI_FUNCTION, DATA); // AC Increase (cursor move right),
don't shift the display.

    DelayMs (100);
    /*DATA.reg=0x34;
    glcd_writeByte (DI_FUNCTION, DATA); // Select extended instruction set.
    DelayUs (72);
    DATA.reg=0x36;
    glcd_writeByte (DI_FUNCTION, DATA); // Graphic display ON.*/

    /*glcd_fillScreen (1);
    glcd_update (); */

    DATA.reg=0x48;
    glcd_writeByte(DI_DATA,DATA);
    DelayMs(100);
}

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

5.9. PRÁCTICA 9

Diseñar un circuito electrónico de control de velocidad de un motor de corriente alterna con set point

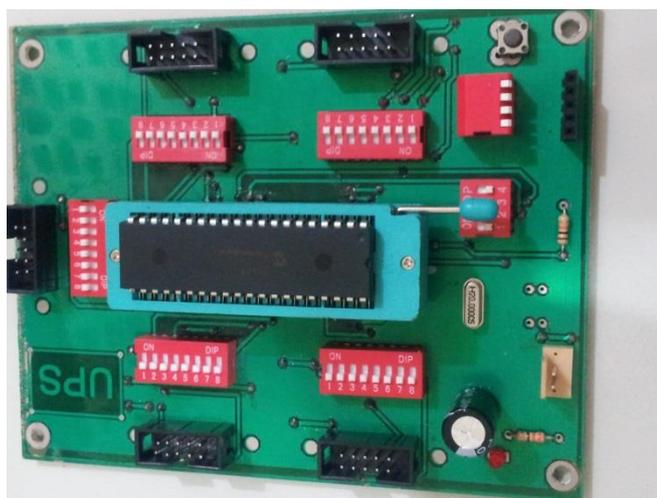
DATOS INFORMATIVOS

- ✓ MATERIA / CÁTEDRA RELACIONADA: Microcontroladores
- ✓ No. DE PRÁCTICA: 9
- ✓ NÚMERO DE ESTUDIANTES: 2
- ✓ NOMBRE DOCENTE: Ing. Luis Córdova
- ✓ TIEMPO ESTIMADO: 2 Horas

Prácticas a realizarse:

Diseñar un circuito electrónico de control de velocidad de un motor de corriente alterna con set point

1. Entrenador Didáctico (Módulo de Control)



Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

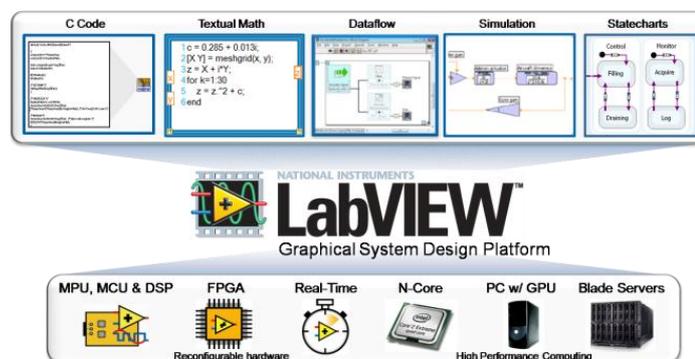
2. Motor Trifasico



3. Tarjeta DAQ



4. LAB VIEW



Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
 UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

5. Display LCD



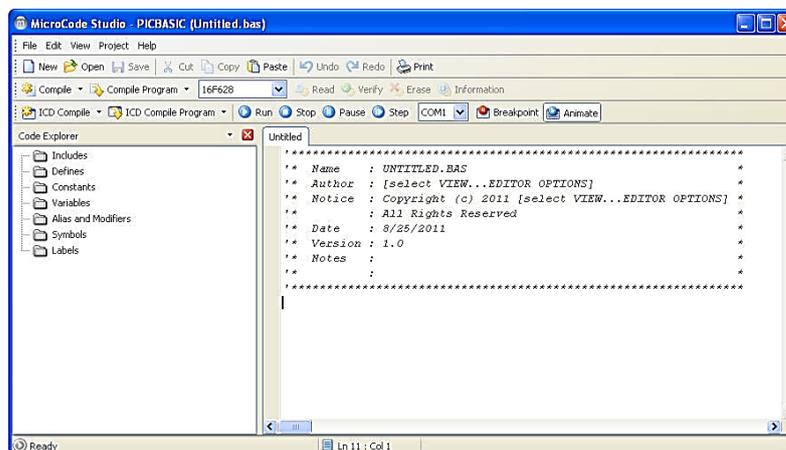
6. Variador de velocidad Altivar 312



Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

7. Software Micro Code Studio



Materiales:

EQUIPO NECESARIO

MATERIAL DE APOYO

- | | |
|--|-------------------|
| ✓ Entrenador Didáctico (Módulo de Control) | ✓ Proyector |
| ✓ Motor trifasico | ✓ Pizarra líquida |
| ✓ Tarjeta DAQ | |
| ✓ Computador, LAB VIEW | |
| ✓ Display LCD | |
| ✓ Software Micro Code Studio | |
| ✓ Variador de velocidad Altivar 132 | |

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Marco Teórico:

A información requerida en esta práctica se puede encontrar revisando el CAPITULO 2, BIBLIOGRAFÍA.

Procedimientos:

Práctica 9

Diseñar un circuito electrónico de control de velocidad de un motor de corriente alterna con set point ingresado desde una interfaz gráfica usando Labview al entrenador didáctico.

- Aplicar PID
- Usar pantalla LCD para mostrar velocidad inicial y velocidad actual

Utilizaremos del entrenador didáctico la tarjeta principal de control, el módulo del motor, además de la interfaz DAQ, cable usb con una computadora en la cual esta previamente diseñado en el entorno LABview en un pequeño sistema de control PID para el motor AC.

De igual forma necesitaremos el motor AC Siemens 1LA7-070-4YA60 y el variador de velocidad Altivar 312, un sensor infrarojo y un diodo infrarojo.

Conectaremos el módulo principal de control mediante un cable bus de datos a la interfaz de motor Siemens 1LA7-070-4YA60

Puerto A,1 → conector analogico 1

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Puerto A,9 (GND) → conector analogico 9 (GND)

Puerto A,10 (Vcc) → conector analogico 10 (Vcc)

Puerto B,1 → conector Digital, 1

Puerto B,9 (GND) → conector analogico 9 (GND)

Puerto B,10 (Vcc) → conector analogico 10 (Vcc)

Puerto C,1 → conector DAQ

Puerto B,9 (GND) → conector analogico 9 (GND)

Puerto B,10 (Vcc) → conector analogico 10 (Vcc)

Conectamos la tarjeta DAQ hacia el computador mediante un cable USB, en uno de sus conectores que se encontraba sin uso.

Conectamos el módulo del motor en las borneras de entrada digital, el pin de dato del sensor infrarojo de igual manera en la bornera que corresponde conectamos el diodo infrarojo, además en las borneras analogicas conectamos la salida al conector de entrada analogica del variador de velocidad Altivar 312.

Seguidamente conetamos de la fuente los voltajes – 12 Vcc + 12 Vcc y GND hacia la tarjeta de interfaz de motor en los conectores que corresponde para que de esta manera pudiera variar de 0 a + 10 Vcc con la ayuda del operacional.

Adicional conectamos la salida de fuerza del variador de velocidad al cableado necesario desde el motor y en la entrada de voltaje conectaremos un flujo externo de 220 voltios AC para poder energizar el motor Siemens 1LA7-070-4YA60 mediante el variador de velocidad.

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Desde el computador con la ayuda de un pequeño programa desarrollado en el entorno LABView seteamos el valor inicial de velocidad que tendra el motor en el momento de arranque para que este valor pueda ser trasmitido mediante la interfaz DAQ haciaa el microcontrolador el cual se encargo de la operaciones lógicas y envio del valor analógico hacia la tarjeta de control interfaz del motor y esta a su vez transformó ese valor analógico de entre cero a cinco voltios a cero a diez voltios y de esta forma el variador de velocidad pudo estabilizar el motor a la velocidad requerida de igual forma se hicieron varias mediciones de velocidad tomando en cuenta los valores obtenidos mediante el sensor infrarojo

```
*****
'* Name   : Práctica Nro 09.BAS           *
'* Author : Franco Rafael - Motesdeoca Israel *
'* Notice : Copyright (c) 2014 UPS-Guayaqui (Tesis de Grado) *
'*       : All Rights Reserved           *
'* Date   : 11/08/2014                   *
'* Version : 1.0                         *
'* Notes  :                               *
'*       :                               *
```

```
*****
#task(rate=20ms,max=1ms)
void ControlSistema(void){
    Medida=read_adc();
    TempReal=Medida;
    TempReal=(TempReal*500)/1024;

    eT=TempRef-TempReal;           //Cálculo error
```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

if(SistControl==1){
    uT=q0*eT - q1*eT_1 + q2*eT_2 + uT_1;    //Cálculo de la salida PID
(uT=[Volt])
    rT=(uT*1000)/5;                        //duty cycle maximo 1000 -> 5V.-
    /* <<<<< AntiWindup >>>>>*/
    if (rT>1000){                          //Salida PID si es mayor que el 1000.-
        uT=(uT + 5*p0)*p00;
        rT=1000;
    }
    if (rT<0n){                            //Salida PID si es menor que el 0
        uT=uT*p00;
        rT=0;
    }
    /* <<<<< Transferencia de salida PID a señal PWM >>>>>*/
    ControlPWM=rT;
    set_pwm1_duty(ControlPWM);
    /* <<<<< Guardar variables para proximo estado >>>>>*/
    eT_2=eT_1;
    eT_1=eT;
    uT_1=uT;
}else{                                     // Control On-Off
    if(eT>0){output_high(PIN_C2);}
    else{output_low(PIN_C2);}
}
}

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```
#task(rate=1ms,max=1ms)
void ImprimirLCD(void){
int j;

if(Print==1){           // Borramos display?
    lcd_comand(1);
    Print=0;
    rtos_yield();       // Se "duerme" hasta el proximo ciclo.-
}
for(j=0;j<LengthMsj1;j++){ // Si hay caracteres en Primer linea envia al LCD.-
    lcd_putc(BufferLCD1[j]);
    rtos_yield();
}
if(LengthMsj2!=0){      // Si hay caracteres a imprimir en la segunda linea..
    lcd_gotoxy(1,2);    // ..salta a la siguiente a la linea 2.
    rtos_yield();
}
for(j=0;j<LengthMsj2;j++){ // Si hay caracteres en Segunda linea envia al LCD.-
    lcd_putc(BufferLCD2[j]);
    rtos_yield();
}
if(PosMod!=0){          // Utilizado para correr el cursor cuando se modifican
                        // los parametros.-
    lcd_gotoxy(PosMod,1); // Vuelve cursor para modificar valores.-
    rtos_yield();
}
rtos_disable(ImprimirLCD);
}
```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

#int_RB
void RB_isr(void)
{
    Tecla=TestTeclado();           //Testea la Tecla Pulsada.-
    if(Tecla!=16){                 //Tecla y no falsa alarma?.-
        rtos_msg_send(VisualizacionLCD,Tecla);
        rtos_enable(VisualizacionLCD);
    }
}

void VisualizacionLCD(void){
    if(rtos_msg_poll(>0){
        SeleccionMenu(rtos_msg_read()); //Dependiendo la Tecla recibida se ubica
            //dentro del Menu.-
        if(Panel==1){               //Si no se a pedido apagar...
            if(Enter==0){           //Y si no se ha presionado Enter...
                ImpresionMenu();    //Cargamos los buffer segun la posicion
                //en el menu para impresion en LCD.-
            }
            rtos_enable(ImprimirLCD); //Habilitamos tarea para impresion en LCD.-
        }
    }
    rtos_disable(VisualizacionLCD);
}

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

5.10. PRÁCTICA 10

Diseñar un circuito electrónico con control PID para un motor de corriente alterna

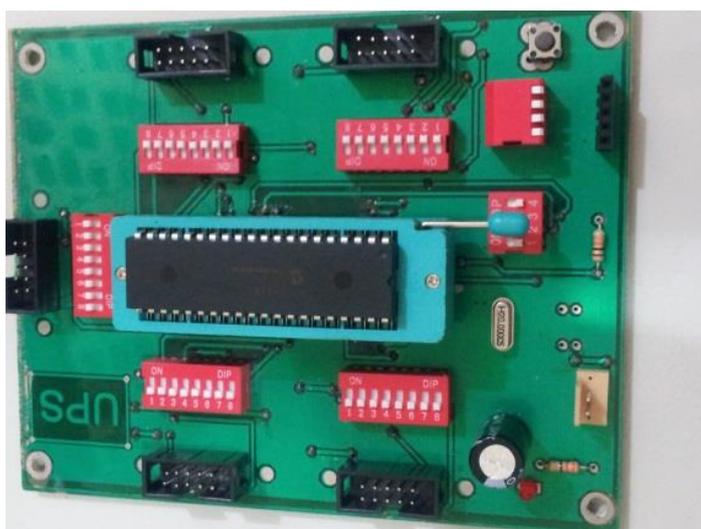
DATOS INFORMATIVOS

- ✓ MATERIA / CÁTEDRA RELACIONADA: Microcontroladores
- ✓ No. DE PRÁCTICA: 10
- ✓ NÚMERO DE ESTUDIANTES: 2
- ✓ NOMBRE DOCENTE: Ing. Luis Córdova
- ✓ TIEMPO ESTIMADO: 2 Horas

Prácticas a realizarse:

Diseñar un circuito electrónico con control PID para un motor de corriente alterna

1. Entrenador Didáctico (Módulo de Control)



Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

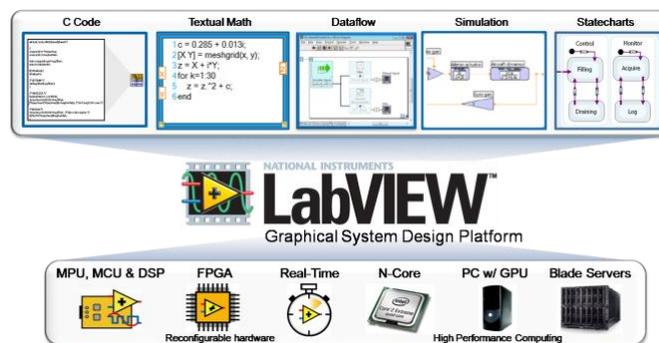
2. Motor Trifasico



3. Tarjeta DAQ



4. LAB VIEW



Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

5. Display LCD



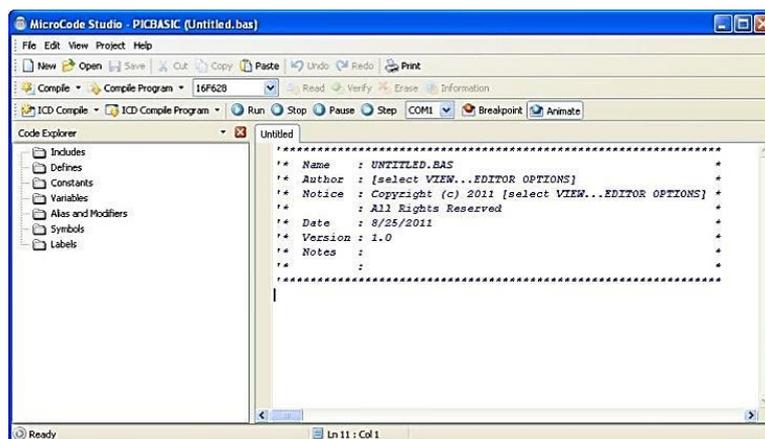
6. Variador de Velocidad Altivar 132



Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

7. Software Micro Code Studio



Materiales:

EQUIPO NECESARIO

- ✓ Entrenador Didáctico (Módulo de Control)
- ✓ Motor trifasico
- ✓ Tarjeta DAQ
- ✓ Computador, LAB VIEW
- ✓ Display LCD
- ✓ Software Micro Code Studio
- ✓ Variador de velocidad Altivar 132

MATERIAL DE APOYO

- ✓ Proyector
- ✓ Pizarra líquida

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Marco Teórico:

A información requerida en esta práctica se puede encontrar revisando el CAPITULO 2, BIBLIOGRAFÍA

Procedimientos:

Práctica 10

Usando el entrenador didáctico diseñar un circuito electrónico con control PID para un motor de corriente alterna y mostrar gráficamente la variación de frecuencia en la computadora usando una interfaz gráfica en LabVIEW.

- Tarjeta DAQ-series NI MYDAQ
- Set point ingresado por teclado.
- Usar LabVIEW para mostrar una interfaz gráfica

Utilizaremos del entrenador didáctico la tarjeta principal de control, el módulo del motor Siemens 1LA7-070-4YA60, además la interfaz DAQ, cable USB con una computadora en la cual esta previamente diseñado en el entorno LABView en un pequeño sistema de control PID para el motor AC.

De igual forma necesitaremos el motor AC Siemens 1LA7-070-4YA60 y el variador de velocidad Altivar 312, un sensor inflarajo y un diodo inflarajo.

Conectaremos el módulo principal de control mediante un cable bus de datos la interfaz del motor.

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Puerto A,1 → conector analogico 1
 Puerto A,9 (GND) → conector analogico 9 (GND)
 Puerto A,10 (Vcc) → conector analogico 10 (Vcc)

Puerto B,1 → conector Digital, 1
 Puerto B,9 (GND) → conector analogico 9 (GND)
 Puerto B,10 (Vcc) → conector analogico 10 (Vcc)

Puerto C,1 → conector DAQ
 Puerto B,9 (GND) → conector analogico 9 (GND)
 Puerto B,10 (Vcc) → conector analogico 10 (Vcc)

Se conectó la tarjeta DAQ hacia el computador mediante un cable USB en uno de sus conectores que se encontraba sin uso.

Conectamos el módulo del motor en las borneras de entrada digital, el pin de dato del sensor infrarojo de igual manera en la bornera que correspondía conectamos el diodo infrarojo además en las borneras analógicas conectamos la salida al conector de entrada analógica del variador de velocidad Altivar 312.

Procedimos a conectar de la fuente los voltajes – 12 Vcc + 12 Vcc y GND hacia la tarjeta de interfaz de motor en los conectores que corresponde para que de esta manera pudiera variar de 0 a +10 Vcc con la ayuda del operacional

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

Conectamos la salida de fuerza del variador de velocidad al cableado necesario desde el motor y en la entrada de voltaje conectaremos un flujo externo de 220 voltios AC, para poder energizar el motor mediante el variador de velocidad

Desde el computador con la ayuda de un pequeño programa desarrollado en el entorno LABView seteamos el valor inicial de velocidad que tendrá el motor en el momento de arranque para que este valor pueda ser transmitido mediante la interfaz DAQ hacia el microcontrolador el cual se encargó de las operaciones lógicas y envió del valor analógico hacia la tarjeta de control interfaz del motor y esta a su vez transformó ese valor analógico de entre cero a cinco voltios a cero a diez voltios y de esta forma el variador de velocidad pudo estabilizar el motor a la velocidad requerida de igual forma se hicieron varias mediciones de velocidad tomando en cuenta los valores obtenidos mediante el sensor infrarrojo.

```

*****
'* Name   : Práctica Nro 10.BAS           *
'* Author : Franco Rafael - Motesdeoca Israel      *
'* Notice : Copyright (c) 2014 UPS-Guayaquil (Tesis de Grado) *
'*       : All Rights Reserved                *
'* Date   : 29/08/2014                       *
'* Version: 1.0                               *
'* Notes  :                                   *
'*       :                                   *
*****

```

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```
#include <pic18.h>
```

```
#include "serial.h"
```

```
#define PWM_TICKS 1
```

```
#define INTEGRAL_CONSTANT (i_gain);
```

```
#define PROPORTIONAL_CONSTANT (p_gain);
```

```
#define DIFFERENTIAL_CONSTANT (d_gain);
```

```
#define JOINT_CONSTANT (j_gain);
```

```
#define ARC_CONSTANT (a_gain);
```

```
#define STOP_DEADBAND 1
```

```
#define PCONTROL_DEADBAND 5
```

```
#define MIN_SPEED 1
```

```
#define MAX_STEP 1
```

```
#define RIGHT_DIRECTION RA4
```

```
#define LEFT_DIRECTION RB5
```

```
#define RIGHT_PWM_H CCPR1L
```

```
#define RIGHT_PWM_M DC1B1
```

```
#define RIGHT_PWM_L DC1B0
```

```
#define LEFT_PWM_H CCPR2L
```

```
#define LEFT_PWM_M DC2B1
```

```
#define LEFT_PWM_L DC2B0
```

```
#define FORWARD_LEFT 1
```

```
#define BACKWARD_LEFT 0
```

```
#define FORWARD_RIGHT 0
```

```
#define BACKWARD_RIGHT 1
```

```
#define RIGHT_BRAKE RA2
```

```
#define LEFT_BRAKE RA3
```

```
#define DEBUG 0
```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

#define DEBUG_ALL 0
#define USE_SPI
#define USE_EEPROM
#define EEPROM_KP_ADDRESS 0
#define EEPROM_KI_ADDRESS 0x08
#define EEPROM_KD_ADDRESS 0x10
#define EEPROM_KJ_ADDRESS 0x18
#define EEPROM_KA_ADDRESS 0x20
#define SPEED_SET 5
#define DIST_SET 10
#define LEFT_SPEED_SET 15
#define RIGHT_SPEED_SET 20
#define LEFT_DIST_SET 25
#define RIGHT_DIST_SET 30
#define GENERAL_READ 35
#define SHIFTED_DIST 40
#define RESET_DIST 45

```

```

void LoopTime(void);
void Initialize(void);
void PwmInitialize(void);
void BlinkAlive(void);
void UpdatePWM(void);
void ManageAccel(void);
void puthex(unsigned int c);
void puthexc(unsigned char c);
void get_params(void);
void write_pwm_left(unsigned int duty_cycle);

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

void write_pwm_right(unsigned int duty_cycle);
void position_control(void);
void parse_SPI(void);
void set_left_position_target(long new_target);
void set_right_position_target(long new_target);
void write_float_to_eeprom( float value, unsigned int address );
void read_float_from_eeprom( float *value, unsigned int address );

```

```

static long last_count_left = 0;
static long last_count_right = 0;
static long error_integral_left = 0;
static long error_integral_right = 0;
static long last_error_left = 0;
static long last_error_right = 0;
static float joint_integral_error = 0;

```

```

volatile long count_left = 0;
volatile long count_right = 0;
unsigned char portb_copy = 0;
volatile long count_left_reference = 0;
volatile long count_right_reference = 0;
volatile long count_left_reference_report = 0;
volatile long count_right_reference_report = 0;
float applied_right = 0;
float applied_left = 0;
long left_tick_target;
long right_tick_target;

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```
char controlling_position = 0;
```

```
char position_slowdown = 0;
```

```
char newmessage = 0;
```

```
char setspeed = 0;
```

```
char setdist = 0;
```

```
int interrupt_speed_left;
```

```
int interrupt_speed_right;
```

```
int interrupt_dist_left;
```

```
int interrupt_dist_right;
```

```
char printmessage = 0;
```

```
char spi_temp = 0;
```

```
char sawthis = 0;
```

```
int desired_target_right = 0;
```

```
int desired_target_left = 0;
```

```
long target_right = 0;
```

```
long target_left = 0;
```

```
float p_gain = 0.01;
```

```
float i_gain = 0.01;
```

```
float d_gain = 0.00;
```

```
float j_gain = 0.00;
```

```
float a_gain = 0.00;
```

```
volatile char rx_message[8];
```

```
char tx_message1[8];
```

```
char tx_message2[8];
```

```
char *current_tx_message = tx_message1;
```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```
char *newest_tx_message = tx_message1;
```

```
void main(void)
```

```
{
```

```
    unsigned char PWM_counter = 0;
```

```
    char c;
```

```
    TRISA = 0xFE;
```

```
    Initialize();
```

```
    PwmInitialize();
```

```
    serial_init();
```

```
    while(1)
```

```
    {
```

```
        if(RA1 == 0 && kbhit())
```

```
        {
```

```
            if(kbhit())
```

```
            {
```

```
                c = getch();
```

```
            }
```

```
                if(c == 'q')
```

```
                {
```

```
                    printf("QuickBug\n\r");
```

```
                    asm("goto 0x6002");
```

```
                }
```

```
                else if(c == 'r')
```

```
            {
```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

desired_target_left = -desired_target_left;
desired_target_right = -desired_target_right;
}

else if(c == 'a')
{
    desired_target_left = -25;
    desired_target_right = 25;
}
else if(c == 'w')
{
    desired_target_left = 25;
    desired_target_right = 25;
}
else if( c == 'd')
{
    desired_target_left = 25;
    desired_target_right = -25;
}
else if(c == 'x')
{
    desired_target_left = -25;
    desired_target_right = -25;
}
else if(c == 's')
{
    desired_target_left = 0;
    desired_target_right = 0;
}

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

else if(c == 'b')
{
    RA2 ^= 1;

    RA3 ^= 1;
    printf("Brakes: %d, %d", RA2, RA3);
}

else if(c == 'p')
{
    set_left_position_target(0);
    set_right_position_target(3375);
}
else if(c == '[')
{
    set_left_position_target(1687);
    set_right_position_target(-1687);
}
else if(c == ']')
{
    set_left_position_target(-1687);
    set_right_position_target(1687);
}

else
{
    write_pwm_left(0);
    write_pwm_right(0);
    get_params();
    count_left = 0;
    count_right = 0;
}

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

        last_count_left = 0;
        last_count_right = 0;
        error_integral_left = 0;
        error_integral_right = 0;
        last_error_left = 0;
        last_error_right = 0;
    }
}
BlinkAlive();
PWM_counter++;
if(PWM_counter == PWM_TICKS)
{
    PWM_counter = 0;
        if(sawthis)
        {
            putchar('T');
            sawthis = 0;
        }
        if(printmessage)
        {
            printmessage = 0;
#if 0
            putchar('.');
            putchar('\n');
            putchar('\r');
            putchar('[');
            for( i = 0; i < 8; i++ )
            {

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

                                puthexc( rx_message[i] );

                                putchar( ' ' );
                                }
                                putchar(']');

                                putchar(' ');
#endif

                                if( WCOL )
                                {
                                        putchar( 'C' );
                                        WCOL = 0;
                                }
                                if( SSPOV )
                                {
                                        putchar( 'O' );
                                        SSPOV = 0;
                                }
                                }
                                parse_SPI();
                                position_control();
                                UpdatePWM();
                                ManageAccel();
                                }
                                LoopTime();
                                }
                                }

```

void parse_SPI(void)

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

{
  if (newmessage)
  {
    putch(newmessage);
    newmessage = 0;
    if(setspeed)
    {
      putch('s');
      setspeed = 0;
      GIEL = 0;
      desired_target_left = interrupt_speed_left;
      desired_target_right = interrupt_speed_right;
      GIEL = 1;
      if(((desired_target_left < 0) && (desired_target_right < 0))
      {
        putch('-');
      }
    }
    if(setdist)
    {
      putch('d');
      setdist = 0;
      GIEL = 0;

      set_left_position_target(interrupt_dist_left);
      set_right_position_target(interrupt_dist_right);
      GIEL = 1;
    }
  }
}

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

}

```
void set_left_position_target(long new_target)
```

```
{
```

```
    GIEH = 0;
```

```
    count_left_reference = count_left;
```

```
    GIEH = 1;
```

```
    left_tick_target = new_target;
```

```
    controlling_position |= 0x01;
```

```
    if(((new_target > 0) && (desired_target_left < 0)) ||
```

```
        ((new_target < 0) && (desired_target_left > 0)))
```

```
    {
```

```
        desired_target_left = -desired_target_left;
```

```
    }
```

```
}
```

```
void set_right_position_target(long new_target)
```

```
{
```

```
    GIEH = 0;
```

```
    count_right_reference = count_right;
```

```
    GIEH = 1;
```

```
    right_tick_target = new_target;
```

```
    controlling_position |= 0x02;
```

```
    if(((new_target > 0) && (desired_target_right < 0)) ||
```

```
        ((new_target < 0) && (desired_target_right > 0)))
```

```
    {
```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

        desired_target_right = -desired_target_right;
    }
}

void position_control(void)
{
    static int posn_error_left;
    static int posn_error_right;

    if(controlling_position)
    {
        GIEH = 0;
        posn_error_left = left_tick_target - ( count_left - count_left_reference );
        posn_error_right = right_tick_target - ( count_right - count_right_reference );
        GIEH = 1;

        putchar('\n');
        putchar('\r');
        puthex(posn_error_left);
        putchar(' ');
        puthex(posn_error_right);
        putchar('|');
        puthex(count_left);
        putchar(' ');
        puthex(count_right);
        putchar('|');
        puthex(desired_target_left);
        putchar(' ');

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

puthex(desired_target_right);
putch('|');

if(controlling_position & 0x01)
{
    if((posn_error_left > 0 && desired_target_left < 0) ||
        (posn_error_left < 0 && desired_target_left > 0))
    {
        putch('<');
        desired_target_left = -desired_target_left;
    }

    if((posn_error_left < (desired_target_left)) &&
        (posn_error_left > -(desired_target_left)))
    {
        desired_target_left = 0;
        controlling_position &= 0xFE;
    }
}
else
{
    desired_target_left = 0;
}

if(controlling_position & 0x02)
{
    if((posn_error_right > 0 && desired_target_right < 0) ||
        (posn_error_right < 0 && desired_target_right > 0))

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

    {
        putchar('>');
        desired_target_right = -desired_target_right;
    }
    if((posn_error_right < (desired_target_right)) &&
        (posn_error_right > -(desired_target_right)))
    {
        desired_target_right = 0;
        controlling_position &= 0xFD;
    }
}
else
{
    desired_target_right = 0;
}
}
}

```

```

void UpdatePWM(void)
{
    long saved_left;
    long saved_right;
    long diff_left;
    long diff_right;
    long error_left;
    long error_right;
    long distance_left;
    long distance_right;
}

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

long report_distance_left;
long report_distance_right;

```

```

GIEH = 0;
saved_left = count_left;
saved_right = count_right;
GIEH = 1;

```

```

distance_left = saved_left - count_left_reference;
distance_right = saved_right - count_right_reference;

```

```

if(1)
{
    if((desired_target_left - target_left) > MAX_STEP)
    {
        target_left += MAX_STEP;
    }
    else if ((desired_target_left - target_left) < -MAX_STEP)
    {
        target_left -= MAX_STEP;
    }
    else
    {
        target_left = desired_target_left;
    }
}

```

```

if(1)

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

{
    if((desired_target_right - target_right) > MAX_STEP)
    {
        target_right += MAX_STEP;
    }
    else if ((desired_target_right - target_right) < -MAX_STEP)
    {
        target_right -= MAX_STEP;
    }
    else
    {
        target_right = desired_target_right;
    }
}

```

```

diff_left = saved_left - last_count_left;
diff_right = saved_right - last_count_right;

```

```

#if 1
    if(controlling_position){
        puthex(diff_left >> 16);
        puthex(diff_left);
        putchar(' ');
        puthex(diff_right >> 16);
        puthex(diff_right);
    }
#endif

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

error_left = target_left - diff_left;
error_right = target_right - diff_right;
#if DEBUG_ALL
    putch('|');
    puthex(error_left);
    putch(' ');
    puthex(error_right);
#endif

error_integral_left += error_left;
error_integral_right += error_right;

    if(desired_target_right == desired_target_left && desired_target_right != 0
&& desired_target_left != 0)
    {
        joint_integral_error += diff_left - diff_right + ARC_CONSTANT;
    }

    if( target_left == 0 && desired_target_left == 0 &&
        ((diff_left >= -STOP_DEADBAND) && (diff_left <=
STOP_DEADBAND)))
    {
        applied_left = error_integral_left * INTEGRAL_CONSTANT;
    }
else
    {
        applied_left = error_integral_left * INTEGRAL_CONSTANT;
        applied_left += error_left * PROPORTIONAL_CONSTANT;

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

applied_left += (error_left - last_error_left) *

DIFFERENTIAL_CONSTANT;

if(desired_target_left == desired_target_right)

{ /*We will only add the straightness gain if we want to go straight*/

applied_left -= joint_integral_error * JOINT_CONSTANT;

}

}

if(applied_left > 1023)

{

applied_left = 1023;

}

else if(applied_left < -1023)

{

applied_left = -1023;

}

if(target_right == 0 && desired_target_right == 0 &&

((diff_right >= -STOP_DEADBAND) && (diff_right <=

STOP_DEADBAND)))

{

applied_right = error_integral_right * INTEGRAL_CONSTANT;

}

else

{

applied_right = error_integral_right * INTEGRAL_CONSTANT;

applied_right += error_right * PROPORTIONAL_CONSTANT;

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

applied_right += (error_right - last_error_right) *

DIFFERENTIAL_CONSTANT;

if(desired_target_left == desired_target_right)

{ /*We will only add the straightness gain if we want to go straight*/

applied_right += joint_integral_error * JOINT_CONSTANT;

}

}

if(applied_right > 1023)

{

applied_right = 1023;

}

else if(applied_right < -1023)

{

applied_right = -1023;

}

if(current_tx_message == tx_message1)

{

newest_tx_message = tx_message2;

}

else

{

newest_tx_message = tx_message1;

}

#if 0

newest_tx_message[0] = (char)((diff_left & 0xFF00) >> 8);

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

newest_tx_message[1] = (char)(diff_left & 0x00FF);
newest_tx_message[2] = (char)((diff_right & 0xFF00) >> 8);
newest_tx_message[3] = (char)(diff_right & 0x00FF);
newest_tx_message[4] = (char)((distance_left & 0xFF00) >> 8);
newest_tx_message[5] = (char)(distance_left & 0x00FF);
newest_tx_message[6] = (char)((distance_right & 0xFF00) >> 8);
newest_tx_message[7] = (char)(distance_right & 0x00FF);
#else
    report_distance_left = (saved_left - count_left_reference_report) / 426;
    report_distance_right = (saved_right - count_right_reference_report) / 426;

newest_tx_message[0] = (char)(diff_left);
newest_tx_message[1] = (char)(diff_right);
newest_tx_message[2] = (char)((report_distance_left & 0xFF00) >> 8);
newest_tx_message[3] = (char)(report_distance_left & 0x00FF);
newest_tx_message[4] = (char)((report_distance_right & 0xFF00) >> 8);
newest_tx_message[5] = (char)(report_distance_right & 0x00FF);
newest_tx_message[6] = (char)(controlling_position);
newest_tx_message[7] = (char)(0xFF);

#endif

    if( current_tx_message == NULL )
    {
        SSPBUF = newest_tx_message[0];
    }

last_count_left = saved_left;
last_count_right = saved_right;

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

last_error_left = error_left;
last_error_right = error_right;
}

void ManageAccel(void)
{
    unsigned int duty_cycle;

    if(applied_left >= 0)
    {
        duty_cycle = (unsigned int)applied_left;
        LEFT_DIRECTION = FORWARD_LEFT;
#ifdef DEBUG_ALL
        putchar('|');
        putchar('+');
        puthex(duty_cycle);
#endif
    }
    else
    {
        duty_cycle = (unsigned int)(-applied_left);
        LEFT_DIRECTION = BACKWARD_LEFT;
#ifdef DEBUG_ALL
        putchar('|');
        putchar('-');
        puthex(duty_cycle);
#endif
    }
}

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

write_pwm_left(duty_cycle);

if(applied_right >= 0)
{
    duty_cycle = (unsigned int)applied_right;
    RIGHT_DIRECTION = FORWARD_RIGHT;
#if DEBUG_ALL
    putchar(' ');
    putchar('+');
    puthex(duty_cycle);
#endif
}
else
{
    duty_cycle = (unsigned int)(-applied_right);
    RIGHT_DIRECTION = BACKWARD_RIGHT;
#if DEBUG_ALL
    putchar(' ');
    putchar('-');
    puthex(duty_cycle);
#endif
}
write_pwm_right(duty_cycle);
#if DEBUG_ALL
    putchar('|');
    puthex(desired_target_left);
    putchar(' ');
    puthex(desired_target_right);

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```
#endif
```

```
}
```

```
void Initialize(void)
```

```
{
```

```
    ADCON1 = 0b10001110;
```

```
    TRISA = 0b11100010;
```

```
    TRISB = 0b11011111;
```

```
    TRISC = 0b11011000;
```

```
    TRISD = 0b00011111;
```

```
    TRISE = 0b00000100;
```

```
    T0CON = 0b10001000;
```

```
    T3CON = 0b10001001;
```

```
    CCP2CON = 0b00000101;
```

```
    IPEN = 1;
```

```
    INTEDG0 = 0;
```

```
    INTEDG1 = 1;
```

```
    INTEDG2 = 1;
```

```
    INT1IP = 1;
```

```
    INT2IP = 1;
```

```
        INTOIE = 1;
```

```
    INT1IE = 1;
```

```
    INT2IE = 1;
```

```
    CCP2IP = 1;
```

```
    RA2 = 1;
```

```
    RA3 = 1;
```

```
    GIE = 1;
```

```
    GIEH = 1;
```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

GIEL = 1;
controlling_position = 0;

`#ifdef USE_SPI`

```

SSPSTAT = 0x00;
SSPCON1 = 0x14;
SSPEN = 1;
CKP = 0;
CKE = 1;
SSPIE = 1;
SSPIP = 0;
    INTEDG0 = 1;
    INTOIE = 0;

T1CKPS1 = 1;
T1CKPS0 = 1;
TMR1CS = 0;
T1OSCEN = 0;
TMR1H = 0xB6;
TMR1L = 0xC2;
TMR1IP = 0;
TMR1IE = 1;
TMR1ON = 0;

```

`#endif`

`#ifdef USE_EEPROM`

```

read_float_from_eeprom( &p_gain, EEPROM_KP_ADDRESS );
read_float_from_eeprom( &i_gain, EEPROM_KI_ADDRESS );
read_float_from_eeprom( &d_gain, EEPROM_KD_ADDRESS );

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

read_float_from_eeprom( &j_gain, EEPROM_KJ_ADDRESS );
read_float_from_eeprom( &a_gain, EEPROM_KA_ADDRESS );

#endif
}

```

```

void PwmInitialize(void)
{
    PR2 = 0xFF;
    T2CON = 0b01111111;
    CCPR1L = 0;
    TRISC &= ~(1 << 2);
    CCP1CON = 0b00001111;

    CCPR2L = 0x00;
    TRISC &= ~(1 << 1);
    CCP2CON = 0b00001111;
}

```

```

void LoopTime(void)
{
    while(TMROIF != 1);
    TMROIF = 0;
    WRITETIMER0(40536);
}

```

```

void interrupt low_priority LowPriISR(void)
{
    int i;

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

static int msg_index = 0;
if(TXIF == 1 && TXIE == 1)
{
    serial_tx_isr();
    TXIF = 0;
}

if(SSPIF == 1)
{
    RA0 = 1;
    if(msg_index == 0)
    {
        if( current_tx_message == NULL )
        {
            current_tx_message = newest_tx_message;
        }
        TMR1ON = 1;
    }
    rx_message[msg_index] = SSPBUF;
    msg_index++;
    SSPIF = 0;
    TMR1H = 0xB6;
    TMR1L = 0xC2;
    if(msg_index > 7)
    {
        msg_index = 0;
        TMR1ON = 0;
        current_tx_message = NULL;
    }
}

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

SSPBUF = newest_tx_message[0];

switch(rx_message[0]){
case SPEED_SET:
interrupt_speed_left = rx_message[2] + (rx_message[1] << 8);
interrupt_speed_right = rx_message[4] + (rx_message[3] <<
8);

setspeed = 1;
newmessage = 'S';

break;
case DIST_SET:
interrupt_dist_left = ((long)(rx_message[2] + (rx_message[1]
<< 8)));
interrupt_dist_right = ((long)(rx_message[4] + (rx_message[3]
<< 8)));

setdist = 1;
newmessage = 'D';

break;
case GENERAL_READ:
newmessage = 'R';
break;
case RESET_DIST:
GIEH = 0;
count_left_reference_report = count_left;
count_right_reference_report = count_right;
GIEH = 1;

default:
newmessage = '!';

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

        break;
    }

    INTOIF = 0;
    INTOIE = 1;
}

else
{
    SSPBUF = current_tx_message[msg_index];
}

RA0 = 0;
}

if(TMR1IF == 1)
{
    TMR1ON = 0;
    TMR1IF = 0;

    msg_index = 0;
    for(i = 0; i<8; i++)
    {
        rx_message[i] = 0;
    }

    TMR1H = 0xB6;
    TMR1L = 0xC2;

    sawthis = 1;
}

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

    }

}

void interrupt HiPriISR(void)
{
    portb_copy = PORTB;

    if( INT2IF == 1 )
    {
        INT2IF = 0;
        if( portb_copy & 0x10 )
        {
            count_left--;
        }
        else
        {
            count_left++;
        }
    }

    if( INT1IF == 1 )
    {
        INT1IF = 0;
        if( portb_copy & 0x08 )
        {
            count_right++;
        }
    }
}

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

else
{
    count_right--;
}
}

if( INTOIF == 1 && INTOIE == 1 )
{
    INTOIE = 0;
    current_tx_message = newest_tx_message;
}
}

#ifdef USE_EEPROM
write_float_to_eeprom( p_gain, EEPROM_KP_ADDRESS );
write_float_to_eeprom( i_gain, EEPROM_KI_ADDRESS );
write_float_to_eeprom( d_gain, EEPROM_KD_ADDRESS );
write_float_to_eeprom( j_gain, EEPROM_KJ_ADDRESS );
write_float_to_eeprom( a_gain, EEPROM_KA_ADDRESS );
#endif

void BlinkAlive(void)
{
#ifdef 0
    static unsigned char alive_count = 250;
    RA0 = 1;
    if(--alive_count == 0)
    {

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

    alive_count = 250;
    RA0 = 0;
}
#endif
}

void puthex(unsigned int c)
{
    char digit;
    int i;
    char buffer[5];
    buffer[4] = 0;
    for(i= 0; i < 4; i++)
    {
        digit = c & 0x0F;
        if(digit > 9)
        {
            buffer[3-i] = 'A' + digit - 10;
        }
        else
        {
            buffer[3-i] = '0' + digit;
        }
        c >>= 4;
    }
    putchar(buffer[0]);
    putchar(buffer[1]);
    putchar(buffer[2]);
}

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```
    putchar(buffer[3]);
}
```

```
void puthexc(unsigned char c)
```

```
{
    char digit;
    int i;
    char buffer[3];
    buffer[2] = 0;
    for(i= 0; i < 2; i++)
    {
        digit = c & 0x0F;
        if(digit > 9)
        {
            buffer[1-i] = 'A' + digit - 10;
        }
        else
        {
            buffer[1-i] = '0' + digit;
        }
        c >>= 4;
    }
    putchar(buffer[0]);
    putchar(buffer[1]);
}
```

```
void write_pwm_left(unsigned int duty_cycle)
```

```
{
```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

LEFT_PWM_H = duty_cycle >> 2;
LEFT_PWM_M = (duty_cycle >> 1) & 0x01;
LEFT_PWM_L = duty_cycle & 0x01;

```

```

}

```

```

void write_pwm_right(unsigned int duty_cycle)

```

```

{
    RIGHT_PWM_H = duty_cycle >> 2;
    RIGHT_PWM_M = (duty_cycle >> 1) & 0x01;
    RIGHT_PWM_L = duty_cycle & 0x01;
}

```

```

#ifdef USE_EEPROM

```

```

void write_float_to_eeprom( float value, unsigned int address )

```

```

{
    char *pointer;
    unsigned int i;
    char byte;

    pointer = (char *)&value;
    for( i = 0; i < sizeof(value); i++ )
    {
        byte = pointer[i];
        EEPROM_WRITE( address + i, byte );
    }
}

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

		REVISION 1/1
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	Laboratorio de Electrónica Digital	
CARRERA	Ingeniería Electrónica	
SEDE	Guayaquil	

```

void read_float_from_eeprom( float *value, unsigned int address )
{
    float temp;
    char *pointer;
    unsigned int i;

    pointer = (char *)&temp;
    for(i = 0; i < sizeof(temp); i++)
    {
        pointer[i] = EEPROM_READ( address + i );
    }
}
#endif

```

Elaborado por: Rafael Franco Israel Montesdeoca	Revisado por: Ing. Luis Córdova	Aprobado por: MSc. Víctor Huilcapi
---	------------------------------------	---------------------------------------

CAPÍTULO VI

COSTO DEL PROYECTO

En este capítulo se detalla los costos de los diferentes elementos que se usaron en el proyecto.

6.1. Costo de Módulo DATA LOGGER

	ELEMENTOS	CANTIDAD POR MÓDULO	COSTO UNITARIO	COSTO TOTAL	
				POR MÓDULO	POR CINCO MÓDULOS
Módulo DATA LOGGER	Memory Stick Datalogger	1	\$ 61,17	\$ 61,17	\$ 305,85
	Tarjeta de circuito impreso (PCB)	1	\$ 4,90	\$ 4,90	\$ 24,50
	Diodo LED 3 mm (ROJO)	1	\$ 0,15	\$ 0,15	\$ 0,75
	Resistencia de 330 ohmios	1	\$ 0,05	\$ 0,05	\$ 0,25
	Conector IDC-10 (Macho)	1	\$ 0,50	\$ 0,50	\$ 2,50
	SUB TOTALES			\$ 66,77	\$ 333,85
	IVA 12%			\$ 8,01	\$ 40,06
	TOTALES			\$ 74,78	\$ 373,91

Tabla 58: Costo de elementos de Módulo DATA LOGGER

Fuente: Autores, 2014

6.2. Costo de Módulo XBEE

Módulo XBEE	ELEMENTOS	CANTIDAD POR MÓDULO	COSTO UNITARIO	COSTO TOTAL	
				POR MÓDULO	POR CINCO MÓDULOS
	Xbee 1MW Wire Ant	1	\$ 32,14	\$ 32,14	\$ 160,70
	Tarjeta de circuito impreso (PCB)	1	\$ 3,92	\$ 3,92	\$ 19,60
	Diodo LED 3 mm (ROJO)	1	\$ 0,15	\$ 0,15	\$ 0,75
	Diodo Rectificador 1N4007	1	\$ 0,05	\$ 0,05	\$ 0,25
	Resistencia de 330 ohmios	1	\$ 0,05	\$ 0,05	\$ 0,25
	Conector IDC-10 (Macho)	1	\$ 0,50	\$ 0,50	\$ 2,50
	SUB TOTALES			\$ 36,81	\$ 184,05
	IVA 12%			\$ 4,42	\$ 22,09
	TOTALES			\$ 41,23	\$ 206,14

Tabla 59: Costo de elementos de Módulo XBEE

Fuente: Autores, 2014

6.3. Costo de Módulo BLUETOOTH

Módulo BLUETOOTH	ELEMENTOS	CANTIDAD POR MÓDULO	COSTO UNITARIO	COSTO TOTAL	
				POR MÓDULO	POR CINCO MÓDULOS
	Bluetooth DF	1	\$ 31,25	\$ 31,25	\$ 156,25
	Tarjeta de circuito impreso (PCB)	1	\$ 3,92	\$ 3,92	\$ 19,60
	Diodo LED 3 mm (ROJO)	1	\$ 0,15	\$ 0,15	\$ 0,75
	Resistencia de 330 ohmios	1	\$ 0,05	\$ 0,05	\$ 0,25

	Conector IDC-10 (Macho)	1	\$ 0,50	\$ 0,50	\$ 2,50
	SUB TOTALES			\$ 35,87	\$ 179,35
	IVA 12%			\$ 4,30	\$ 21,52
	TOTALES			\$ 40,17	\$ 200,87

Tabla 60: Costo de elementos de Módulo BLUETOOTH
Fuente: Autores, 2014

6.4. Costo de Módulo GSM - GPRS - GPS

	ELEMENTOS	CANTIDAD POR MÓDULO	COSTO UNITARIO	COSTO TOTAL	
				POR MÓDULO	POR CINCO MÓDULOS
Módulo GSM - GPRS - GPS	GSM - GPRS – GPS	1	\$ 133,83	\$ 133,83	\$ 669,15
	Tarjeta de circuito impreso (PCB)	1	\$ 10,00	\$ 10,00	\$ 50,00
	Diodo LED 3 mm (ROJO)	1	\$ 0,15	\$ 0,15	\$ 0,75
	Resistencia de 330 ohmios	1	\$ 0,05	\$ 0,05	\$ 0,25
	Conector IDC-10 (Macho)	1	\$ 0,50	\$ 0,50	\$ 2,50
	SUB TOTALES			\$ 144,53	\$ 722,65
	IVA 12%			\$ 17,34	\$ 86,72
	TOTALES			\$ 161,87	\$ 809,37

Tabla 61: Costo de elementos de Módulo GSM - GPRS – GPS
Fuente: Autores, 2014

6.5. Costo de Módulo ARDUINO ETHERNET

Módulo ARDUINO ETHERNET	ELEMENTOS	CANTIDAD POR MÓDULO	COSTO UNITARIO	COSTO TOTAL	
				POR MÓDULO	CINCO MÓDULOS
	ARDUINO ETHERNET	1	\$ 78,35	\$ 78,35	\$ 391,75
	Tarjeta de circuito impreso (PCB)	1	\$ 7,60	\$ 7,60	\$ 38,00
	Diodo LED 3 mm (ROJO)	1	\$ 0,15	\$ 0,15	\$ 0,75
	Resistencia de 330 ohmios	1	\$ 0,05	\$ 0,05	\$ 0,25
	Conector IDC-10 (Macho)	1	\$ 0,50	\$ 0,50	\$ 2,50
	SUB TOTALES			\$ 86,65	\$ 433,25
	IVA 12%			\$ 10,40	\$ 51,99
	TOTALES			\$ 97,05	\$ 485,24

Tabla 62: Costo de elementos de Módulo ARDUINO ETHERNET
Fuente: Autores, 2014

6.6. Costo de Módulo INTERFACE DE ENTRADA

Módulo INTERFACE DE ENTRADA	ELEMENTOS	CANTIDAD POR MÓDULO	COSTO UNITARIO	COSTO TOTAL	
				POR MÓDULO	CINCO MÓDULOS
	Transistor 2N3904	8	\$ 0,15	\$ 1,20	\$ 6,00
	Tarjeta de circuito impreso (PCB)	1	\$ 12,75	\$ 12,75	\$ 63,75
	Diodo LED 3 mm (VERDE)	8	\$ 0,15	\$ 1,20	\$ 6,00
	Resistencia de 330 ohmios	8	\$ 0,05	\$ 0,40	\$ 2,00
	Resistencia de 4,7 k ohmios	8	\$ 0,05	\$ 0,40	\$ 2,00
	Bornera de dos polos	8	\$ 0,25	\$ 2,00	\$ 10,00
	Conector IDC-10 (Macho)	1	\$ 0,50	\$ 0,50	\$ 2,50
	SUB TOTALES			\$ 18,45	\$ 92,25
	IVA 12%			\$ 2,21	\$ 11,07
	TOTALES			\$ 20,66	\$ 103,32

Tabla 63: Costo de elementos de Módulo INTERFACE DE ENTRADA
Fuente: Autores, 2014

6.7. Costo de Módulo Interface para Motor

	ELEMENTOS	CANTIDAD POR MÓDULO	COSTO UNITARIO	COSTO TOTAL		
				POR MÓDULO	POR CINCO MÓDULOS	
Módulo Interface para Motor	Integrado UA741	1	\$ 0,50	\$ 0,50	\$ 2,50	
	Socket 2 x 4	1	\$ 0,05	\$ 0,05	\$ 0,25	
	Tarjeta de circuito impreso (PCB)	1	\$ 5,95	\$ 5,95	\$ 29,75	
	Diodo LED 3 mm (ROJO)	1	\$ 0,15	\$ 0,15	\$ 0,75	
	Resistencia de 330 ohmios	2	\$ 0,05	\$ 0,10	\$ 0,50	
	Resistencia de 10 k ohmios	3	\$ 0,05	\$ 0,15	\$ 0,75	
	Bornera de dos polos	3	\$ 0,25	\$ 0,75	\$ 3,75	
	Bornera de tres polos	1	\$ 0,35	\$ 0,35	\$ 1,75	
	Conector IDC-10 (Macho)	2	\$ 0,50	\$ 1,00	\$ 5,00	
	SUB TOTALES				\$ 9,00	\$ 45,00
	IVA 12%				\$ 1,08	\$ 5,40
	TOTALES				\$ 10,08	\$ 50,40

Tabla 64: Costo de elementos de Módulo Interface para Motor

Fuente: Autores, 2014

6.8. Costo de Módulo Interface de salida

	ELEMENTOS	CANTIDAD POR MÓDULO	COSTO UNITARIO	COSTO TOTAL	
				POR MÓDULO	POR CINCO MÓDULOS
Módulo Interface de salida	Relay DC5V (5 pines)	8	\$ 1,50	\$ 12,00	\$ 60,00
	Transistor 2N3904	8	\$ 0,15	\$ 1,20	\$ 6,00
	Tarjeta de circuito impreso (PCB)	1	\$ 20,00	\$ 20,00	\$ 100,00

Diodo LED 3 mm (AMARILLO)	8	\$ 0,15	\$ 1,20	\$ 6,00
Resistencia de 330 ohmios	8	\$ 0,05	\$ 0,40	\$ 2,00
Resistencia de 4,7 k ohmios	8	\$ 0,05	\$ 0,40	\$ 2,00
Diodo Rectificador 1N4007	8	\$ 0,05	\$ 0,40	\$ 2,00
Bornera de tres polos	8	\$ 0,35	\$ 2,80	\$ 14,00
Conector IDC-10 (Macho)	1	\$ 0,50	\$ 0,50	\$ 2,50
SUB TOTALES			\$ 38,90	\$ 194,50
IVA 12%			\$ 4,67	\$ 23,34
TOTALES			\$ 43,57	\$ 217,84

Tabla 65: Costo de elementos de Módulo Interface de salida
Fuente: Autores, 2014

6.9. Costo de Módulo Fuente

	ELEMENTOS	CANTIDAD POR MÓDULO	COSTO UNITARIO	COSTO TOTAL	
				POR MÓDULO	POR CINCO MÓDULOS
Módulo Fuente	Capacitor Electrolítico 2200 uf / 25v	2	\$ 1,50	\$ 3,00	\$ 15,00
	Regulador de voltaje 7805	1	\$ 0,60	\$ 0,60	\$ 3,00
	Regulador de voltaje 7812	1	\$ 0,60	\$ 0,60	\$ 3,00
	Regulador de voltaje 7912	1	\$ 0,75	\$ 0,75	\$ 3,75
	Capacitor Cerámico de 0,01 uf	2	\$ 0,05	\$ 0,10	\$ 0,50
	Tarjeta de circuito impreso (PCB)	1	\$ 5,00	\$ 5,00	\$ 25,00
	Puente Rectificador GBU3A	1	\$ 1,00	\$ 1,00	\$ 5,00

Bornera de dos polos	2	\$ 0,25	\$ 0,50	\$ 2,50
SUB TOTALES			\$ 11,55	\$ 57,75
IVA 12%			\$ 1,39	\$ 6,93
TOTALES			\$ 12,94	\$ 64,68

Tabla 66: Costo de elementos de Módulo Fuente
Fuente: Autores, 2014

6.10. Costo de Módulo Principal o Control

	ELEMENTOS	CANTIDAD POR MÓDULO	COSTO UNITARIO	COSTO TOTAL	
				POR MÓDULO	POR CINCO MÓDULOS
Módulo Principal o Control	Microcontrolador PIC 18F4550	1	\$ 9,60	\$ 9,60	\$ 48,00
	Dip Swith 8 pines	5	\$ 0,75	\$ 3,75	\$ 18,75
	Dip Swith 4 pines	2	\$ 0,50	\$ 1,00	\$ 5,00
	Cristal de Cuarzo 20Mhz	1	\$ 1,50	\$ 1,50	\$ 7,50
	Capacitor Electrolítico 1000 uf / 16v	1	\$ 1,00	\$ 1,00	\$ 5,00
	Tarjeta de circuito impreso (PCB)	1	\$ 28,60	\$ 28,60	\$ 143,00
	Conector IDC-10 (Macho Simple)	5	\$ 0,31	\$ 1,55	\$ 7,75
	Conector Molex 1x2	1	\$ 0,50	\$ 0,50	\$ 2,50
	Resistencia de 330 ohmios	1	\$ 0,05	\$ 0,05	\$ 0,25
	Resistencia de 10 k ohmios	1	\$ 0,05	\$ 0,05	\$ 0,25
	Zip Socket	1	\$ 3,50	\$ 3,50	\$ 17,50
	regleta de 45 pines	1	\$ 3,50	\$ 3,50	\$ 17,50

Hembra				
Diodo LED 3 mm (ROJO)	1	\$ 0,15	\$ 0,15	\$ 0,75
Pulsante	1	\$ 0,05	\$ 0,05	\$ 0,25
SUB TOTALES			\$ 54,80	\$ 274,00
IVA 12%			\$ 6,58	\$ 32,88
TOTALES			\$ 61,38	\$ 306,88

Tabla 67: Costo de elementos de Módulo Principal o Control
Fuente: Autores, 2014

6.11. Costo de elementos varios

	ELEMENTOS	CANTIDAD POR MÓDULO	COSTO UNITARIO	COSTO TOTAL	
				POR MÓDULO	POR CINCO MÓDULOS
ELEMENTOS	Protoboard	1	\$ 7,50	\$ 7,50	\$ 37,50
	GLCD Display de 128 x 64	1	\$ 23,66	\$ 23,66	\$ 118,30
	Chasis de entrenador didáctico	1	\$ 80,00	\$ 80,00	\$ 400,00
	LCD Display 20 x 4	1	\$ 16,40	\$ 16,40	\$ 82,00
	Cable Plano 10 pines (bus de datos)	1	\$ 1,00	\$ 1,00	\$ 5,00
	Teclado Matricial	1	\$ 6,38	\$ 6,38	\$ 31,90
	Sensor de temperatura DS18B20	1	\$ 4,00	\$ 4,00	\$ 20,00
	Clock Calendar DS1307	1	\$ 4,00	\$ 4,00	\$ 20,00
	Cristal de Cuarzo 32768	1	\$ 1,20	\$ 1,20	\$ 6,00
	Luz Piloto	1	\$ 0,50	\$ 0,50	\$ 2,50
	Switch de 3 pines 10 ^a	1	\$ 0,80	\$ 0,80	\$ 4,00
	Portafusibles	1	\$ 1,50	\$ 1,50	\$ 7,50
	Bornera para toma de Voltaje	1	\$ 0,60	\$ 0,60	\$ 3,00

Elevadores Plásticos	20	\$ 1,15	\$ 23,00	\$ 115,00
Transformadores 12V-0V-12V	1	\$ 12,00	\$ 12,00	\$ 60,00
Placas con Título de Tesis	1	\$ 12,00	\$ 12,00	\$ 60,00
Papel Para lata	1	\$ 3,50	\$ 3,50	\$ 17,50
Impresiones en papel de cada Módulo	1	\$ 3,00	\$ 3,00	\$ 15,00
IDC - 10P	20	\$ 0,36	\$ 7,20	\$ 36,00
SUB TOTALES			\$ 208,24	\$ 1.041,20
IVA 12%			\$ 24,99	\$ 124,94
TOTALES			\$ 233,23	\$ 1.166,14

Tabla 68: Costo de elementos varios

Fuente: Autores, 2014

BIBLIOGRAFÍA

- Adolfo Gonzáles José, (1998). Aplicaciones de los microcontroladores PIC de Microchip, editorial McGraw Hill.
- Ethernet, recuperado de <http://es.wikipedia.org/wiki/Ethernet>, tomado el 2014-Abril
- GLCD Pantalla táctil, recuperado de <http://www.freewebs.com/glafebre/lcdgrafico.htm>, tomado el 2012, Septiembre
- Introducción a PID, recuperado de http://www.sapiensman.com/control_automatico/control_automatico5.htm, tomado el 2013-Noviembre
- LabVIEW, recuperado de <http://es.wikipedia.org/wiki/LabVIEW>, tomado el 2014-Julio
- Lantronix, recuperado de <http://translate.google.com.ec/translate?hl=es-419&sl=en&tl=es&u=http://www.lantronix.com/device-networking/embedded-device-servers/xport.html>, tomado el 2012, Septiembre
- Microchip Technologic, recuperado de <http://www.microchip.com>, tomado el 2012, Agosto
- Motores de corriente alterna, recuperado de <http://www.unicrom.com>, tomado el 2012, Septiembre
- National Instrument, recuperado de <http://latam.ni.com/>, tomado el 2012-Diciembre

- Parallax Datalogger, recuperado de <http://www.parallax.com/product/27937>, tomado el 2014-Junio
- PIC 18F4550, recuperado de http://usuarios.multimania.es/electronicapic/manual/primeros_pasos_con_el_18F4550.pdf, http://picmania.garcia-cuervo.net/invitados_primer18f4550.php, tomado el 2012, Agosto
- Programación de PIC en lenguaje C, recuperado de <http://www.mikroe.com>, tomado el 2012, Septiembre
- Proteus Profesional, recuperado de http://es.wikipedia.org/wiki/Proteus_%28electr%C3%B3nica%29, <http://www.intercambiosvirtuales.org/software/proteus-professional-v8-0-151417>, tomado el 2014-Julio
- Tarjeta de adquisición de datos MyDAQ, recuperado de <http://www.datalights.com.ec>, tomado el 2014-Mayo
- Tecnología Bluetooth, recuperado de [http:// masadelante.com/Bluetooth](http://masadelante.com/Bluetooth), tomado el 2012, Septiembre
- X-bee información, recuperado de <http://tecnologicobj12.blogspot.com/2011/09/que-es-xbee.html>, tomado el 2012, Septiembre

CONCLUSIONES

Mediante esta Tesis de Grado se amplió el aprendizaje y práctica de las tecnologías inalámbricas, uso de la electrónica en conjunto con las interfaces de programación de aplicaciones tales como LabVIEW y la tarjeta NI USB-6009 para el control de velocidad de un motor de corriente alterna.

Se afianzaron los conocimientos adquiridos en la Universidad, además de la investigación de diferentes temas de carácter electrónico y de control de motores trifásicos.

El entrenador didáctico integra módulos que permite procesar datos, consultas y a la vez mostrarlos a través de interfaces de salida como la GLCD o una LCD, esto permite que el docente pueda afianzar los conocimientos impartidos de una manera mucha más práctica en la materia de microprocesados.

Por medio de prácticas y la comprensión del uso correcto de cada uno de los módulos de comunicación en los entrenadores didácticos fabricados para el Laboratorio de Electrónica Digital, el estudiante logrará afianzar los conocimientos teóricos impartidos por el docente, cumpliéndose así el objetivo principal de esta tesis de grado que consiste en crear herramientas innovadoras para interiorizar temas del campo de la ingeniería electrónica.