

UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE CUENCA

CARRERA DE INGENIERIA ELECTRICA – ELECTRÓNICA

TÍTULO:

**“DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE INTERACCIÓN CON EL
COMPUTADOR PARA NIÑOS CON PARÁLISIS CEREBRAL”**

AUTORES:

JOSÉ ISRAEL QUINDE CERCADO.

Tesis previa a la obtención del título

de Tecnólogo Electrónico

CHRISTIAN GEOVANNY ASTUDILLO ASTUDILLO.

Tesis previa a la obtención del título

de Tecnólogo Eléctrico.

DIRECTOR:

ING. LUIS ABAD

CUENCA – ECUADOR

2012

DECLARACIÓN

Nosotros, Christian Geovanny Astudillo Astudillo y Jose Israel Quinde Cercado, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Politécnica Salesiana según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



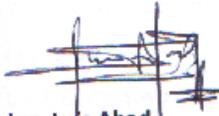
Christian Geovanny Astudillo



José Israel Quinde Cercado

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Christian Geovanny Astudillo Astudillo y José Israel Quinde Cercado, bajo mi supervisión.



Ing. Luis Abad
DIRECTOR DE TESIS

AGRADECIMIENTOS

Primeramente quiero agradecer a Dios, por darme su bendición en los momentos más difíciles de mi vida y también en el desarrollo del presente proyecto.

También quiero agradecer a mis padres, por darme la oportunidad de estudiar y de esta manera adquirir una profesión.

Y otro agradecimiento muy especial al Ing. Luis Abad por guiar el presente trabajo y por sus sabios consejos.

Christian Geovanny Astudillo

AGRADECIMIENTOS

Son muchas las personas a las que debo agradecer por todo el apoyo y enseñanzas que me brindaron para concluir mi proyecto:

DIOS:

Gracias, por acompañarme y guiarme todos los días y por darme la fuerza necesaria para alcanzar otra meta más de mi vida.

FAMILIA:

Gracias, por brindarme siempre su paciencia y comprensión, guiando mi camino hacia un mejor porvenir.

José Israel Quinde Cercado

Índice general

1. 1. ANALISIS DE REQUERIMIENTOS	6
1.1. Importancia de la interacción con niños con parálisis cerebral . . .	6
1.2. Principales problemas de niños con deficiencia auditiva con problemas en la interacción sensorial.	6
1.3. CLASIFICACIÓN	8
1.4. POSIBILIDADES TERAPEUTICAS	8
1.5. TERAPIA DE INTEGRACIÓN SENSORIAL	9
2. 2. FUNDAMENTOS TEÓRICOS	10
2.1. Introducción	10
2.2. Acelerómetro de doble eje	10
2.3. INTRODUCCIÓN A LOS MICROCONTROLADORES	11
2.3.1. UNIDADES BÁSICAS DE LOS MICROCONTROLADORES	11
2.3.2. PROCESADOR	11
2.3.3. MEMORIA	12
2.3.3.1. VENTAJAS DE LOS MICROCONTROLADORES	14
2.3.4. APLICACIONES DE LOS MICROCONTROLADORES	14
2.4. MICROCONTROLADOR AVR	14
2.4.1. Generalidades de los micro controladores AVR	15
2.4.2. Características del Microcontrolador AVR	15
2.4.2.1. Tiny	16
2.4.2.2. Mega	16
2.4.2.3. Xmega	16
2.4.2.4. AVR's	16
2.5. Características del Microcontrolador AVR ATMEL de 8 bits . . .	16
2.5.0.5. CARACTERÍSTICAS GENERALES DEL AVR ATMEGA328	17
2.5.0.6. DESCRIPCIÓN DE PINES DEL ATMEGA328:	19
2.5.0.7. FUNCIONES SECUNDARIAS	20
2.5.0.8. INTERFAZ HOMBRE MAQUINA (HMI)	21

<i>ÍNDICE GENERAL</i>	2
3. Configuración Software Complementario	24
3.1. Configuración de Arduino como grabador ISP	24
3.1.1. Instalación software complementario:	26
3.1.2. Programador Arduino ISP 8Mhz:	28
3.1.3. Descarga Bootloader:	33
3.1.4. Descarga de sketches en el microcontrolador	35
4. Configuración del Mouse	37
4.1. Esquema general del sistema:	37
4.2. Comunicación:	37
4.2.1. Comunicación Serial:	37
4.2.2. Comunicación Serial Cableada:	38
4.2.2.1. Lectura Acelerómetro:	41
4.2.3. Comunicación Serial Inalámbrica:	49
4.3. Comunicación USB:	57
5. Diseño y fabricación de circuitos impresos	60
5.1. Diseño en Fritzing	60
5.2. Elaboración del prototipo PCB	62
5.2.1. Transferencia a la baquelita	62
6. Diseño y elaboración de los contenedores	66
6.1. Diseño del contenedor del Transmisor y Receptor	66
6.2. Elaboración del contenedor del Transmisor y Receptor	68
7. Aprendizaje del sistema por un niño:	72
7.1. Resumen, Conclusiones y Recomendaciones	73
7.1.1. Resumen	73
7.1.2. Conclusiones	74
7.1.3. Recomendaciones	74
I Anexo 1: Imagenes del proceso de construcción de la maqueta	78
II Anexo 2: Hoja de datos del Acelerómetro	84
III Anexo 3: Hoja de datos del Modulo RF	88
IV Anexo 4: Hoja de datos del Microcontrolador Atmega328	93

Índice de figuras

2.1. Arquitectura interna de AVR Atmega328	17
2.2. Disposición de pines del microcontrolador Atmega328	20
2.3. Puerto USB[10]	22
2.4. Tipos de conectores USB	23
3.1. Comunicación ISP	25
3.2. Aplicaciones/Utilidades/Terminal	26
3.3. echo \$PATH	27
3.4. Verificación de compatibilidad	27
3.5. Instalación de Xcode Tools	28
3.6. Descarga pagina oficial	28
3.7. Arduino/Mostrar contenido del paquete	29
3.8. Dirección del archivo boards.txt	29
3.9. Configuraciones del archivo boards.txt	30
3.10. Edición de las configuraciones del archivo boards.txt	30
3.11. Leds grabador ISP	31
3.12. Sketch ArduinoISP	32
3.13. Tarjeta Arduino Uno	32
3.14. Puerto de Comunicación	33
3.15. Verificación y Carga del Sketch	33
3.16. Arduino Grabador ISP	34
3.17. Selección del Microcontrolador Atmega328	34
3.18. Grabar Secuencia de Inicio	35
3.19. Descarga de Sketch en microcontrolador Atmega328	35
3.20. Tarjeta Atmega328p 8Mhz	36
4.1. Esquema General del Prototipo	37
4.2. Comunicación Serial Cableada, Modulo Transmisor	38
4.3. Sketch receptor Arduino Uno	39
4.4. Hardware Comunicación Serial Cableada	40
4.5. Monitor Serial	40
4.6. Datos recibidos por el Arduino	41
4.7. Acelerómetro 2ejes	41
4.8. Principio de funcionamiento del Acelerómetro	42

4.9. Pulso emitido por el acelerómetro	42
4.10. Conexión Acelerómetro al Arduino	43
4.11. Código lectura acelerómetro	44
4.12. Lectura de datos de los ejes del acelerómetro	44
4.13. Acelerómetro en Atmega328	45
4.14. Transmisor con lectura del Acelerómetro	46
4.15. Receptor	47
4.16. Comunicación Transmisor con Arduino UNO	48
4.17. Monitor Serial Lectura de Datos	49
4.18. Modulo Transmisión RF	49
4.19. Configuración de Puerto lectura Modulo RF	50
4.20. Lectura de Módulos RF	51
4.21. Lectura de parámetros de fabrica del Modulo RF	51
4.22. Configuración de Paridad	52
4.23. Configuración de Canal	52
4.24. Configuración Final del Modulo RF	53
4.25. Escritura de la Configuración en el Modulo RF	53
4.26. Conexión de los módulos Transmisor y Receptor.	54
4.27. Receptor Antena	55
4.28. Transmisor Antena	56
4.29. Recepción de Datos en la computadora	56
4.30. Receptor Arduino Leonardo	57
4.31. Transmisor, Receptor (Arduino Leonardo)	58
4.32. Configuración Arduino Leonardo	58
4.33. Configuración Desplazamiento del Mouse	59
5.1. Selección de Ventana del Proyecto	60
5.2. Esquema General del Proyecto	61
5.3. Prototipo del receptor	61
5.4. Prototipo del receptor	62
5.5. Corte de baquelita para el Receptor	63
5.6. Corte de la baquelita para el Transmisor	63
5.7. Elementos para transferir la impresión (Plancha, Baquelita, Circuito impreso)	63
5.8. Sujeción del circuito impreso a la baquelita	64
5.9. Placas obtenidas después de la transferencia Termica	64
5.10. Recipiente de plástico para atacar el exceso de cobre	65
5.11. Suelta de los elementos del receptor	65
6.1. Diseño del Contenedor del Receptor	67
6.2. Diseño del Contenedor del Transmisor	68
6.3. Piezas utilizadas para el ensamblaje del transmisor y el receptor	69
6.4. Dobleces en los contenedores a):Transmisor, b): Receptor	69
6.5. Cable del Receptor + Receptor	70
6.6. Prototipo ensamblado	70
6.7. Cable del Transmisor + Transmisor + Pulsante click Izquierdo	71

6.8. Transmisor	71
7.1. Utilización del Click Izquierdo del mouse	72
7.2. Utilización del mouse sin click	73
7.3. Utilización del mouse sin click	73
7.4. Pruebas con Arduino y microcontrolador Atmega328	79
7.5. Programación Microcontrolador Atmega328	79
7.6. Corte de las tarjetas de Transmisor y Receptor del proyecto	80
7.7. Eliminación del Cobre Sobrante de las placas	80
7.8. Corte de las plantillas de las carcazas	81
7.9. Doble de los contenedores Acrílicos	81
7.10. Perforaciones necesarias para los contenedores	82
7.11. Pruebas finales	82
7.12. Especificaciones Acelerómetro	85
7.13. Especificaciones Acelerómetro	86
7.14. Especificaciones Acelerómetro	87
7.15. Especificaciones Modulos RF	89
7.16. Especificaciones Modulos RF	90
7.17. Especificaciones Modulos RF	91
7.18. Especificaciones Modulos RF	92
7.19. Especificaciones Microcontrolador Atmega328	94
7.20. Especificaciones Microcontrolador Atmega328	95
7.21. Especificaciones Microcontrolador Atmega328	96
7.22. Especificaciones Microcontrolador Atmega328	97

Capítulo 1

1. ANALISIS DE REQUERIMIENTOS

En este capítulo trataremos sobre los problemas a analizar y de la importancia de la interacción sensorial en personas con discapacidades, específicamente en niños con parálisis cerebral.

1.1. Importancia de la interacción con niños con parálisis cerebral.

Parálisis cerebral es un trastorno neuro motor que comprende todas las lesiones cerebrales como consecuencia de enfermedades o acciones externas en el momento del parto. Es un trastorno no progresivo de la movilidad y la postura que se debe a una lesión o anomalía del desarrollo del cerebro inmaduro.

La integración sensorial es la organización de las sensaciones para producir conductas adaptativas y aprendizajes. Permite el adecuado funcionamiento del cerebro y del cuerpo. Es el más importante de los procesos sensoriales realizados por el cerebro.

1.2. Principales problemas de niños con deficiencia auditiva con problemas en la interacción sensorial.

Los sistemas sensoriales de nuestro organismo son:

1. Sistema Vestibular.
2. Sistema Propioceptivo.
3. Sistema Táctil.

4. Sistema Auditivo
5. Sistema Visual
6. Sistema Olfativo
7. Sistema Gustatorio

Una adecuada Integración Sensorial, es decir, una adecuada integración de los sistemas sensoriales entres sí, posibilita un eficiente desarrollo motor, socio-emocional, y de aprendizaje.

Cuando se observan algunas de las siguientes características en los niños, podemos pensar que existen dificultades de procesamiento sensorial en algunos de los sistemas sensoriales. No tienen que estar presentes todas estas características para determinar un problema de integración sensorial; puede que sólo existan algunas de ellas. [2]

Cuando alguno de los sistemas sensoriales no funciona correctamente, podemos encontrarnos con algunos de los siguientes problemas: - Problemas de aprendizaje académico en el colegio: los niños que no siguen el mismo ritmo de aprendizaje, que tienen dificultades en el razonamiento lógico, en la secuenciación, en la planificación motora, etc. A menudo pueden confundirse con trastornos de la lateralidad, déficit de atención, retraso mental, dislexia, retraso madurativo, etc. Pero en realidad se trata de problemas en el procesamiento sensorial. Cuando el niño no alcanza el mismo nivel de lectura ni de escritura, dificultades en las tareas matemáticas, en la memorización de conceptos, etc. [2]

- Hiperactividad – Hipoactividad (alteración de los niveles de actividad): también pueden aparecer alteraciones en los niveles de actividad. El niño puede necesitar gran cantidad de estímulos para estar tranquilo y contento (niños hiperactivos), o por el contrario puede necesitar una cantidad muy pequeña de estímulos (hipoactivos), y son niños que siempre suelen estar en un rincón, sin moverse demasiado ni practicar las mismas actividades de juego que los otros niños de su edad.[2]
- Retrasos psicomotrices en la motricidad fina o gruesa, o en ambas: son niños considerados como patosos, descoordinados, con dificultad para practicar deportes o realizar los mismos juegos motrices que los otros niños. Otros niños presentan dificultades sólo en motricidad fina, con un agarre incorrecto del lápiz, dificultad para ensartar elementos, para pegar, recortar, abotonar, etc.[2]
- Retraso en la adquisición del lenguaje, con problemas de fluidez, de pronunciación de expresividad o de comprensión: son niños que presentan niveles de lenguaje inferiores al nivel correspondiente para su edad cronológica, sin tener alteraciones auditivas ni orales. Algunos niños no encuentran las palabras adecuadas, presentando dificultades de expresión y falta de fluidez, otros niños tardan mucho en empezar a hablar (a los dos años todavía no tienen verbalización), o en otros casos podemos encontrar dificultades en la pronunciación de determinados fonemas.[2]

- Comportamientos problemáticos: el niño con problemas en el procesamiento sensorial puede tener problemas conductuales, que pueden ser explicados por las dificultades sensoriales. Los niños pueden ser explosivos, poco flexibles a los cambios (horarios, distribución. . .), o pueden tener dificultades en las transiciones (cambios de una actividad a otra, de un lugar a otro. . .). El niño puede mostrar una irritabilidad inexplicable, o puede llorar repentinamente sin motivo aparente. . . Pero en realidad la causa suele ser un desajuste sensorial, debido a cambios bruscos del tipo de estimulación, que el niño percibe como una agresión. [2]

1.3. CLASIFICACIÓN

- Tipo
 - Espástica: Daño en la corteza motora o en vías subcorticales intracerebrales (vía piramidal), se caracteriza por la hipertonía rígida o espástica, control limitado de la musculatura del cuello, falta de coordinación entre la musculatura intraoral, perioral y masticatoria, con alteración de masticación, deglución y fonación. Es la forma más habitual (70 %).
- Tipo
 - Disquinética o distónica: daño en los núcleos de la base. Alteración del tono muscular con fluctuaciones y cambios bruscos y movimientos involuntarios. Movimientos mandibulares frecuentes e incontrolados con cierre abrupto de los maxilares y bruxismo. Alteraciones de masticación deglución y fonación. (15 %)
- Tipo
 - Atáxica: Daño a nivel cerebeloso. Hipotonía, incoordinación de movimientos y trastornos del equilibrio. Lentos movimientos incontrolados y bajo tono muscular
 - Formas mixtas.

1.4. POSIBILIDADES TERAPEUTICAS

- La terapia temprana es de vital importancia. No se ha encontrado la forma de reparar las células dañadas.
 - Terapia física: para lograr caminar, sentarse, mantener el equilibrio, Etc.
 - Terapia ocupacional: Destrezas como vestirse, comer, escribir, motricidad fina, etc.
 - Terapia del habla-lenguaje: destrezas para la comunicación
 - Tecnología asistencial: aparatos para la comunicación y tecnología computacional.

1.5. TERAPIA DE INTEGRACIÓN SENSORIAL

Los terapeutas ocupacionales con formación específica en Integración Sensorial, pueden proporcionar terapia de integración sensorial para todos aquellos niños que presenten dificultades en el procesamiento sensorial, y como consecuencia directa presenten dificultades en el aprendizaje y en el desarrollo en general. Los niños con dificultades de integración sensorial pueden jugar en el entorno y recibir gran cantidad y variedad de estímulos, pero no lo hacen de una forma organizada y por lo tanto no integran correctamente la información que reciben. Por este motivo necesitan un entorno con elementos de suspensión, diseñados especialmente para proporcionar estimulación vestibular, propioceptiva, táctil... El terapeuta se encarga de modificar y adaptar el entorno para que el niño pueda interactuar de forma efectiva. El terapeuta ayuda al niño para que seleccione aquella estimulación que necesita en cada momento, y aprenda a procesarla e integrarla de forma adecuada .[2]

Cuando el niño recibe la estimulación que necesita y aprende a procesarle, se le exigen actividades de simples a complejas que permitan al niño generar respuestas adaptadas al mismo tiempo que reciben la estimulación sensorial. No se trata de un juego libre como el que pueden hacer en el parque jugando con los columpios. Se trata de un juego dirigido, que facilita la estimulación adecuada para cada niño, de tal modo que este aprende a interpretarla de forma correcta, y a medida que va integrando la información, sus respuestas son más adaptadas y sus niveles de desarrollo y de aprendizaje mejoran de forma evidente. [2]

Capítulo 2

2. FUNDAMENTOS TEÓRICOS

2.1. Introducción

2.2. Acelerómetro de doble eje

Se denomina acelerómetro a un instrumento destinado a medir aceleraciones, el acelerómetro es uno de los transductores más versátiles, los acelerómetros son utilizados para medir el desplazamiento, la velocidad y la aceleración de los sistemas mecánicos. En el caso del Memsic2125, el acelerómetro es un dispositivo térmico de bajo costo, capaz de medir la inclinación, colisión, estática y aceleración dinámica, rotación y vibración con un rango de ± 3 g en dos ejes. Las aplicaciones que se les puede dar a los acelerómetros son múltiples, hoy en día se ha implementado el uso de estos dispositivos en teléfonos celulares en todo el mundo.

Las aplicaciones son diferentes y únicas, en donde los resultados nos dejan realmente sorprendidos. Siendo un caso típico de aplicación el iPhone.

Este dispositivo junto con los circuitos eléctricos asociados se puede usar para la medida de velocidad y desplazamiento además de la determinación de formas de onda y frecuencia. Una de las ventajas principales de este tipo de transductor es que se puede hacer tan pequeño que su influencia sea despreciable sobre el dispositivo vibrador. El intervalo de frecuencia típica es de 2 Hz a 10 KHz.

Se utiliza en instrumentación, para detectar defectos en máquinas rotativas, prediciendo por ejemplo el mal estado de un rodamiento o cojinete antes de que se provoque la avería. En bombas impulsoras de líquidos detectan los fenómenos de cavitación que pulsan a unas frecuencias características.

Los acelerómetros electrónicos son fabricados para medir la aceleración en una, dos o tres dimensiones. De manera que sea posible medir la aceleración en cada eje. Esta característica permite medir la inclinación de un cuerpo, puesto que es posible determinar con el acelerómetro la componente de la aceleración

provocada por la gravedad que actúa sobre el cuerpo.

2.3. INTRODUCCIÓN A LOS MICROCONTROLADORES

Desde la invención de los semiconductores, el desarrollo de la tecnología digital ha dado lugar a dispositivos cada vez más complejos y rápidos, entre ellos se encuentra los microprocesadores y los microcontroladores. Los microcontroladores se encuentran en nuestro trabajo, en nuestra casa y en nuestra vida, estos controlan el funcionamiento de los teclados de las computadoras, están en los teléfonos celulares, en los hornos de microondas y, en general, en todo aparato electrónico que posea un grado de automatismo. Se dice que un controlador es un dispositivo que se emplea para manejar uno o varios procesos. Hasta hace unos 35 años, los controladores se construían con componentes electrónicos de lógica discreta; posteriormente se emplearon los microprocesadores, apoyados con chips de memoria y dispositivos de E/S sobre una tarjeta de circuito impreso. Desde comienzos de los 90 todos los elementos del controlador se han podido incluir en un solo circuito integrado, el cual recibe el nombre de microcontrolador. Es decir, un microcontrolador es un chip que posee en su interior a un microprocesador, memoria de programa, memoria de datos y puertos para comunicarse con el exterior.

2.3.1. UNIDADES BÁSICAS DE LOS MICROCONTROLADORES

De acuerdo a lo anterior, los microcontroladores poseen unidades básicas importantes los mismos que se describen a continuación: _ Procesador o CPU (Unidad Central de Proceso). _ Memoria RAM para contener los datos. . _ Memoria para el programa tipo ROM/EPROM/EEPROM/Flash. _ Líneas de E/S para comunicarse con el exterior. También puede poseer otros bloques de apoyo que flexibilizan aún más su uso, tales como: _ Módulos para el control de periféricos: temporizadores, puertos serie y paralelo, CAD: Conversores Analógico/Digital, CDA: Conversores Digital/Analógico, etc. Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema. _ Sistemas de protección de sobre corriente o cortocircuito.

2.3.2. PROCESADOR

Es un elemento importante en el microcontrolador ya que se encarga de direccionar la memoria de instrucciones, recibir el código de operación de la instrucción en curso, su decodificación y la ejecución de la operación que implica la instrucción, así como la búsqueda de los operandos y el almacenamiento del resultado.

2.3.3. MEMORIA

En los microcontroladores la memoria de instrucciones y datos se encuentra integrada en el mismo chip. La memoria de instrucciones debe ser no-volátil, tipo ROM/PROM/EPROM, la misma que se encarga de contener el programa de instrucciones el mismo que gobierna la aplicación. La memoria de datos es de tipo RAM, volátil, y se destina a guardar las variables y los datos. La RAM es de poca capacidad, pues sólo debe contener las variables y los cambios de información que se produzcan en el transcurso del programa. Por otra parte, como sólo existe un programa activo, no se requiere guardar una copia del mismo en la RAM pues se ejecuta directamente desde la ROM.

Las siguientes son las tecnologías de memoria no volátil de mayor utilización:

- ROM con máscara: memoria de solo lectura pues se graba durante la fabricación del chip.
- OTP: programable una sola vez por el usuario.
- EPROM: puede borrarse y grabarse muchas veces usando un grabador gobernado desde un PC. El borrado se hace mediante rayos ultravioleta.
- EEPROM: pueden grabarse y borrarse eléctricamente desde el grabador y a través del PC.
- FLASH: memoria no volátil de bajo consumo que se puede escribir y borrar. Funciona como una ROM y densidad que la EEPROM. EEPROM y FLASH pueden ser reprogramados en el mismo circuito.

LÍNEAS DE ENTRADA Y SALIDA

Son aquellas líneas las cuales permiten al microcontrolador comunicarse con el exterior.

RELOJ PRINCIPAL

Es un circuito de reloj incorporado al microcontrolador el cual permite sincronizar la frecuencia de trabajo del mismo.

MÓDULOS PARA EL CONTROL DE PERIFÉRICOS

Cada fabricante oferta numerosas versiones de una arquitectura básica de microcontrolador.

En algunas amplía las capacidades de las memorias, en otras incorpora nuevos recursos, reduce las prestaciones al mínimo para aplicaciones muy simples, etc.

La labor del diseñador es encontrar el modelo mínimo que satisfaga todos los requerimientos de su aplicación.

De esta forma, minimizará el coste, el hardware y el software. Los principales recursos específicos que incorporan los microcontroladores son:

- TEMPORIZADORES Básicamente se emplea para controlar períodos de tiempo y llevar la cuenta de acontecimientos que suceden en el exterior.
 - PERRO GUARDIÁN Consiste en un temporizador que, cuando se desborda y pasa por 0, provoca un reset automáticamente en el sistema.

- **PROTECCIÓN ANTE FALLO DE ALIMENTACIÓN** Se trata de un circuito que resetea al microcontrolador cuando el voltaje de alimentación (VDD) es inferior a un voltaje mínimo ("Brown-out"). Mientras el voltaje de alimentación sea inferior al de Brown-out el dispositivo se mantiene reseteado, comenzando a funcionar normalmente cuando sobrepasa dicho valor.
- **ESTADO DE REPOSO O BAJO CONSUMO** Los microcontroladores poseen una instrucción especial (SLEEP) la cual les permita esperar sin hacer nada, es decir estado de reposo o bajo consumo de energía hasta que se produzca algún acontecimiento en el exterior el cual le permita ponerse en funcionamiento nuevamente.
- **CONVERSIONOR A/D** Los microcontroladores poseen estos conversores con la finalidad de procesar señales analógicas y después convertirlas en señales digitales para su interpretación.
- **CONVERSIONOR D/A** Este tipo de conversores hacen que los datos digitales obtenidos después del procesamiento se transformen en señal analógica la cual se saca a través de las líneas de salida del microcontrolador.
- **COMPARADOR ANALÓGICO** La mayoría de los microcontroladores poseen un amplificador operacional el cual funciona como un comparador entre una señal fija de referencia y una señal variable la cual se aplica a un pin del microcontrolador.
- **MODULADOR PWM** Son circuitos que proporcionan en su salida la modulación por ancho de pulsos, que se ofrecen al exterior a través de los pines del microcontrolador.
- **PUERTOS DE E/S DIGITALES** Todos los microcontroladores destinan algunos de sus pines a soportar líneas de E/S digitales. Por lo general, estas líneas se agrupan de a ocho formando puertos. Las líneas digitales de los puertos pueden configurarse como Entrada o como Salida cargando un 1 ó un 0 en el bit correspondiente de un registro destinado a su configuración.
- **PUERTOS DE COMUNICACIÓN** Para ofrecer al microcontrolador la posibilidad de comunicarse con otros dispositivos externos, y poder adaptarlos con otros elementos bajo normas y protocolos, existen algunos modelos que disponen de recursos que permiten directamente esta tarea, entre los que se destacan a continuación: UART, adaptador de comunicación serie asíncrona. USART, adaptador de comunicación serie síncrona y asíncrona. Puerto paralelo esclavo para poder conectarse con los buses de otros microprocesadores. USB (Universal Serial Bus), que es un moderno bus serie para los PC. Bus I2C, que es un interfaz serie de dos hilos desarrollado por Philips. CAN (Controller Area Network), para permitir la adaptación con redes de conexionado multiplexado desarrollado conjuntamente por Bosch e Intel para el cableado de dispositivos en automóviles. En EE.UU. se usa el J1850.

2.3.3.1. VENTAJAS DE LOS MICROCONTROLADORES

La utilización de los microcontroladores permite tener algunas ventajas que se describen a continuación:

- Aumento de prestaciones: un mayor control sobre un determinado elemento representa una mejora considerable en el mismo.
- Aumento de la fiabilidad: al reemplazar el microcontrolador por un elevado número de elementos disminuye el riesgo de averías y se precisan menos ajustes.
- Reducción del tamaño en el producto acabado: La integración del microcontrolador en un chip disminuye el volumen, la mano de obra y los stocks.
- Mayor flexibilidad: las características de control están programadas por lo que su modificación solo necesita cambios en el programa de instrucciones. El microcontrolador es en definitiva un circuito integrado que incluye todos los componentes de un computador. Debido a su reducido tamaño es posible montar el controlador en el propio dispositivo al que gobierna. En este caso el controlador recibe el nombre de controlador empotrado (embedded controller).

2.3.4. APLICACIONES DE LOS MICROCONTROLADORES

Algunas aplicaciones de los microcontroladores son:

- En sistemas de comunicación: centrales telefónicas, transmisores, receptores, teléfonos fijos, celulares, fax, etc.
- En electrodomésticos: lavadoras, hornos de microondas, heladeras, televisores, reproductores de DVD, microcomponentes, controles remotos, etc.
- Industria informática: se encuentra en casi todos los periféricos: mouse, teclados, escáneres, impresoras, etc.
- Domótica: sistemas de alarma y seguridad, control de procesos hogareños a distancia, etc. · Automatización: climatización, seguridad, etc.
- Industrias: autómatas, control de procesos, etc. · Otros: instrumentación, electro medicina, ascensores, calefacción, aire acondicionado, sistemas de navegación, etc.

2.4. MICROCONTROLADOR AVR

Los AVR son una familia de microcontroladores RISC (Reduced Instruction Set Computing) de ATMEL. La arquitectura de los AVR fue concebida por dos

estudiantes en el Norwegian Institute of Technology, y posteriormente refinada y desarrollada en Atmel Norway, la empresa subsidiaria de Atmel, fundada por los dos arquitectos del chip. Esta tecnología RISC es totalmente diferente con la tecnología CISC (Complex Instruction Set Computing), ya que posee instrucciones con complejidad reducida y memoria flash eléctricamente reprogramable.

Con el pasar del tiempo estos dispositivos han ido formando parte de la mayoría de los equipos electrónicos, desde los más sencillos hasta llegar a formar parte de los más complejos, hoy en día la mayoría de estos dispositivos los utiliza por la facilidad de interactuar con los periféricos y por otras ventajas que estos presentan como son costo, consumo de energía y la facilidad que éstos tienen al ser programados.

El AVR es una CPU de arquitectura Harvard, es decir que las instrucciones y los datos se almacenan en espacios de memoria separados para mejorar el rendimiento. Tiene 32 registros de 8 bits los cuales trabajan como acumuladores. Algunas instrucciones sólo operan en un subconjunto de estos registros. La concatenación de los 32 registros, los registros de entrada/salida y la memoria de datos conforman un espacio de direcciones unificado, al cual se accede a través de operaciones de carga/almacenamiento. A diferencia de los microcontroladores PIC, el stack se ubica en este espacio de memoria unificado, y no está limitado a un tamaño fijo.

2.4.1. Generalidades de los micro controladores AVR

Los microcontroladores AVR están basados en una nueva arquitectura RISC que incorpora memoria Flash para el programa y memoria EEPROM para los datos. Esta arquitectura fue diseñada con el objetivo de que sea compatible con lenguaje C, con esto nos permite desarrollar nuestros proyectos y trabajar en alto nivel.

Esta tecnología RISC es totalmente diferente con la tecnología CISC (Complex Instruction Set Computing), ya que posee instrucciones con complejidad reducida y memoria flash eléctricamente reprogramable.

2.4.2. Características del Microcontrolador AVR

- Alto desempeño, baja potencia.
- Arquitectura RISC avanzada:
- 120 instrucciones poderosas, la mayoría con ejecución de un solo ciclo de reloj.
- 32x8 registros de trabajo de propósito general.
- Operación totalmente estática.
- Programa y Memoria de Datos no volátiles:

- 2/4/8 kbytes de Memoria Flash Programable en el sistema, con duración: 10000 ciclos de escritura/borrado.
- 128/256/512 bytes de EEPROM programable en el sistema, con duración: 100000 ciclos de escritura/borrado.
- 128/256/512 bytes de SRAM interna.
- Cerrojo de programación para auto programar la Memoria Flash y Seguridad de Datos de EEPROM.

2.4 Clasificación del Microcontrolador AVR

Los AVRs se clasifican generalmente en cuatro grupos:

2.4.2.1. Tiny

Posee de 1-8KB de memoria de programa, con sistema periférico limitado.

2.4.2.2. Mega

Posee de 4-256KB de memoria de programa con sistema de instrucción extendida (multiplica las instrucciones para manejar memorias de programa más grandes), es decir posee un sistema periférico extenso

2.4.2.3. Xmega

Posee 16-256KB de memoria de programa. Además ofrece funcionamiento extendido, por ejemplo el acceso directo a memoria, el "sistema del acontecimiento", y ayuda de la criptografía. Periférico extenso fijado con DACs

2.4.2.4. AVRs

Microcontroladores de uso específico, con características especiales no encontradas en los otros miembros de la familia del AVR, tales como regulador del LCD, regulador del USB, PWM avanzado, PODER etc. FPSLIC (Circuito integrado programable del nivel de sistema del campo), una base del AVR con FPGA. El FPSLIC utiliza SRAM para el código del programa del AVR, diferente del resto del AVRs. En parte debido a la diferencia relativa de la velocidad entre SRAM y el flash, la base del AVR en el FPSLIC puede funcionar en hasta 50MHz.

2.5. Características del Microcontrolador AVR AT-MEL de 8 bits

Análogamente expondremos las familias de 8 bits de Atmel.

Device	Flash	EEPROM	RAM	Interrupt Vector Size
ATmega48PA	4K Bytes	256 Bytes	512 Bytes	1 instruction word/vector
ATmega88PA	8K Bytes	512 Bytes	1K Bytes	1 instruction word/vector
ATmega168PA	16K Bytes	512 Bytes	1K Bytes	2 instruction words/vector
ATmega328P	32K Bytes	1K Bytes	2K Bytes	2 instruction words/vector

Cuadro 2.1: Descripción de dispositivos AVR

2.5.0.5. CARACTERÍSTICAS GENERALES DEL AVR ATMEGA328

El microcontrolador Atmega328; es un microcontrolador de 8 bits que se encuentra basado en una y arquitectura Harvard y características avanzadas de tipo RISC. Se caracteriza por su alto rendimiento y su bajo consumo de potencia.

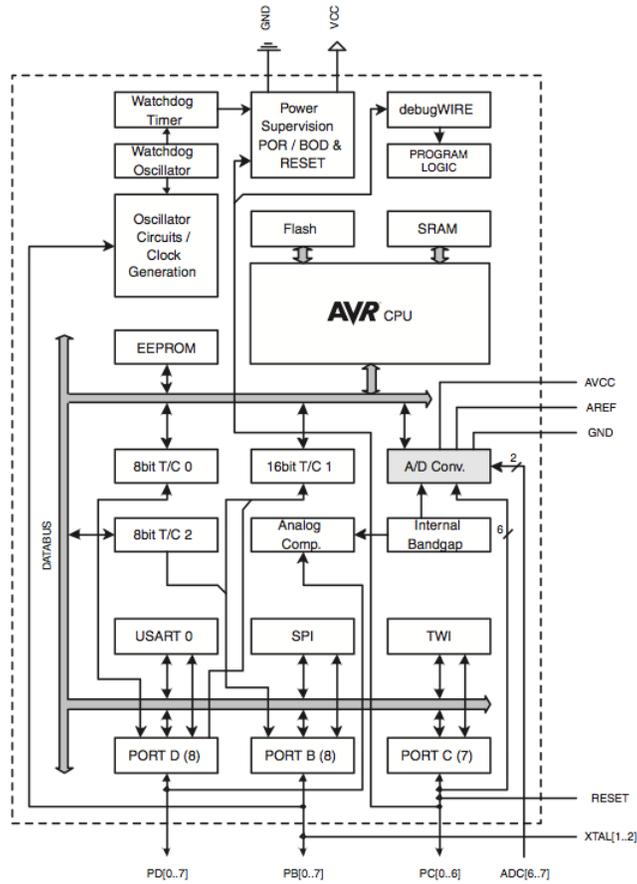


Figura 2.1: Arquitectura interna de AVR Atmega328

La tecnología RISC (Reduced Instruction Set Computing), presenta instrucciones con complejidad reducida, a diferencia de otros que tienen tecnología CISC (Complex Instrucción Set Computing), permite una rápida ejecución de las instrucciones que se ejecutan en un solo ciclo de reloj, el ATMEGA 8L consigue obtener 1 MIPS por Mhz, permitiendo al diseñador del sistema optimizar su consumo de energía versus la velocidad de procesamiento). Las instrucciones en la memoria de programas son ejecutados con estructura segmentada (pipelining), al mismo tiempo que una instrucción es ejecutada, se realiza la búsqueda de la próxima instrucción. Este concepto permite habilitar instrucciones para ser ejecutadas con cada ciclo de reloj.

De acuerdo al diagrama de bloques de la figura 1.3, que se presentó tenemos las siguientes características del microcontrolador a estudiar en este proyecto:

- Posee 32 registros de 8 bits de propósito general como acumuladores. Todos estos registros están conectados a la unidad aritmética lógica (ALU) para un rápido acceso, una instrucción es ejecutada con uno solo ciclo de reloj.
- Memorias de programa y de datos no volátiles.
- Memoria FLASH de 8 K bytes auto programable en sistema.
- Resistencia: 1000 ciclos de escritura / borrado.
- Sección de código añadida opcional con bits de bloqueo independientes.
- Programación en sistema con el programa añadido On-Chip.
- Operación de lectura durante la escritura.
- Memoria EEPROM de 512 bytes.
- Resistencia: 100000 ciclos de escritura / borrado.
- Memoria interna SRAM de 1K bytes
- Bloqueo programable para seguridad del software.
- Características de los periféricos Dos Timers / contador de 8 bits con pre-escaler separado y modo comparación.
- Un Timer / contador de 16 bits con pre-escaler separado, modo comparación y modo captura.
- Comparador analógico ON-CHIP.
- Timer watchdog programable con oscilador separado On-Chip.
- Interface serie SPI maestro / esclavo.
- USART serie programable.
- Contador en tiempo real con oscilador separado.

- ADC de 8 canales en los encapsulados TQFP y MLF. 6 canales de 10 bits de precisión. 2 canales de 8 bits de precisión.
- ADC de 6 canales en el encapsulado PDIP: 4 canales de 10 bits de precisión. 2 canales de 8 bits de precisión.
- 3 canales de PWM. - Interface serie de dos hilos orientada a byte.

Características especiales del microcontrolador.

- Reset de Power-on y detección de Brown-out programable.
- Oscilador RC interno calibrado.
- Fuentes de interrupción externas e internas
- 5 modos de descanso: idle, reducción de ruido ADC, Power-save, Power-down y Stanby.
- PDIP de 28 pines, TQFP y MLF de 32 pines.
- Tensiones de funcionamiento: 4.5 V – 5.5 V (ATmega328).
- Niveles de velocidad: 0 – 8 MHz (ATmega328).
- Consumo de energía a 4 MHz: 3 V, 250C.
- Activo: 3.6 mA.
- Modo Idle: 1.0 mA.
- Modo power-down: 0,5 μ A.
- Oscilador interno de 1, 2, 4, 8 MHz

2.5.0.6. DESCRIPCIÓN DE PINES DEL ATMEGA328:

Algunos pines del microcontrolador tienen más de una función, por tanto, tendrán funciones principales y funciones secundarias como se muestra en la figura

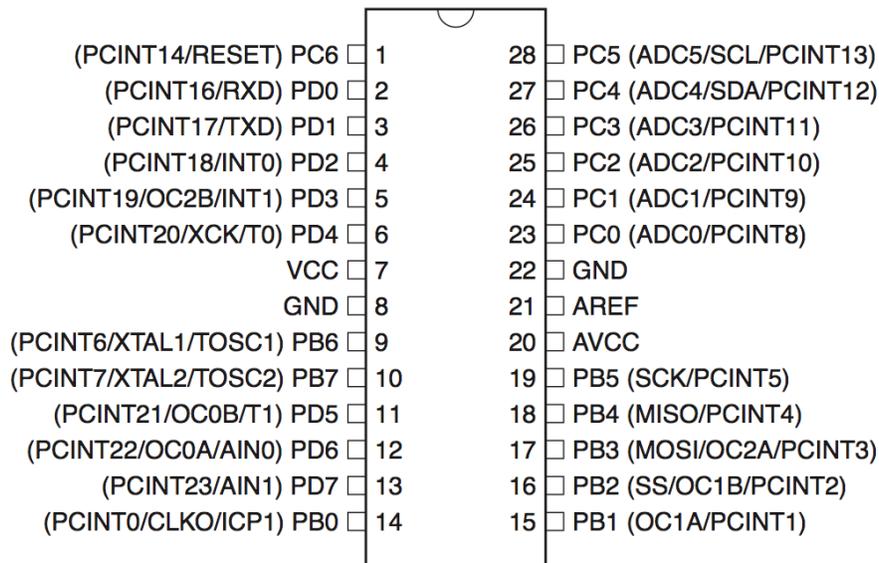


Figura 2.2: Disposición de pines del microcontrolador Atmega328

FUNCIONES PRINCIPALES

- Puerto D: PD0-PD7 (pines 2-6, 11-13)
- Puerto B: PB0-PB7 (pines 14-19, 9-10)
- Puerto C: PC0-PC6 (pines 23-28,1)
- Vcc, GND: (pines 7,8) pines de alimentación CD.
- El voltaje de alimentación es 5V y la corriente de 300 mA.
- ARef (pin 21), referencia analógica para el convertor A/D.
- AVcc (pin 20), voltaje de alimentación para el convertor A/D.

2.5.0.7. FUNCIONES SECUNDARIAS

- Reset (pin 1): si el fusible RSTDISBL esta programado, el pin PC6 se configura como entrada de la señal de reset externo.
- Aplicando un nivel bajo (0V) durante 1,5 μ s en este pin, provoca la re inicialización del microcontrolador; además este pin recibe un voltaje de 13.0V durante la programación del microcontrolador.
- RXD, TXD (pines 2, 3): estos pines sirven para la comunicación serial RS232.
- INT0, INT1 (pines 4, 5): pines de interrupciones externas.

- XCK/T0, T1 (pines 6, 11): pulsos externos para el Timer 0 y 1.
- XTAL1/TOSC1, XTAL2/TOSC2 (pines 9, 10): pines de conexión del oscilador externo. Si se emplea el oscilador interno, el XT externo sirve para el conteo asíncrono del Timer/Counter.
- AIN0, AIN1 (pines 12, 13): pines de comparadores análogos.
- ICP1 (pin 14): entrada para el módulo de captura.
- OC1A, OC1B (pines 15, 16): pines de salida PWM.
- MOSI/OC2, MISO, SCK (pines 17, 18, 19): pines empleados en la grabación del microcontrolador. OC2 pin de salida del comparador.
- ADC0-ADC5 (pines 23-28): pines del módulo A/D.
- SDA, SCL (pines 27, 28): señales de datos y reloj para la comunicación I2C.

2.5.0.8. INTERFAZ HOMBRE MAQUINA (HMI)

La interfaz Hombre-Maquina o HMI por sus siglas en ingles Human Machine Interface. Se podría decir que es el medio por el cual se puede realizar la comunicación o entendimiento entre una maquina y una persona. Esta comunicación entre el computador y el hombre se puede dar mediante dispositivos de entrada y salida en el computador, PLCs, RTU, etc. Todos estos dispositivos deben tener medio sencillo que facilite la interacción y que simplifique las tareas de la actividad que este realiza. Su objetivo principal es la interacción con el usuario, realizando procesos de intercambio.

1. La HMI tiene como finalidad estudiar: [13]
2. El hardware, el software ambos en función de la interacción
3. Como lo establecen Lewis y Reiman (1993): el diseño está en relación al usuario y no a la máquina (user-centered design).
4. Los modelos mentales de los usuarios con relación al modelo del sistema.
5. Las tareas realizables por el sistema y su adaptación a las necesidades del usuario.

El impacto en las organizaciones. No hay mejor forma de entender algo que con un ejemplo sencillo que todos hemos usado alguna vez, el ejemplo más claro de un pantalla táctil (HMI) es un cajero automático ya que con solo tocarlo nos permite realizar varias acciones como obtener dinero.

Puertos E/S:
Puertos USB



Figura 2.3: Puerto USB[10]

El USB en sus principios fue utilizado por Intel, aunque luego fueron siguiendo esta tecnología otros los grandes fabricantes, de forma que se ha establecido como un estándar importante.

El primer ordenador que incluyó un puerto USB de forma estándar fue el iMac de Apple, presentado en Marzo de 1998, que utilizaba esta conexión para el teclado y el ratón. Por su parte el mundo del PC solo comenzó a utilizarlo cuando Microsoft introdujo los controladores correspondientes en la versión OSR 2.1 de Windows 95. Fue a partir de Windows 95C cuando los sistemas de MS incorporan de forma estándar soporte para este bus. En el ámbito de servidores la incorporación se produjo en Windows 2000.

Los primeros dispositivos que empezaron a utilizar este tipo de conexión fueron las cámaras de video conferencia, aunque hoy en día se puede encontrar una gran variedad de aplicaciones dentro de las PCs, ya que la conexión USB se ha convertido en el método universal de conexión de periféricos, incluyendo dispositivos de almacenamiento y los denominados HMI ("Interfaz Hombre Maquina"), principalmente ratones y teclados.

Topología

En un bus USB existen dos tipos de elementos: Anfitrión ("host") y dispositivos; a su vez, los dispositivos pueden ser de dos tipos: concentradores y funciones.[2]

Cables y conectores

El cable de bus USB es de 4 hilos, y comprende líneas de señal (datos) y alimentación, con lo que las funciones pueden utilizar un único cable.

Pin	Nombre	Descripción	Color
1	VBUS	+ 5 V. CC	rojo
2	D-	Data -	azul
3	D+	Data +	amarillo
4	GND	Tierra	verde

Cuadro 2.2: Código de colores y disposición de pines del puerto USB

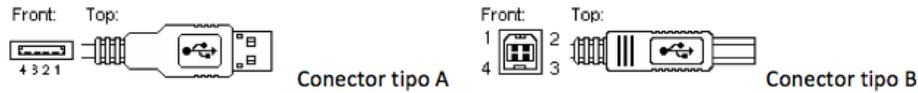


Figura 2.4: Tipos de conectores USB

Existen dos tipos de cable: apantallado y sin apantallar. En el primer caso el par de hilos de señal es trenzado; los de tierra y alimentación son rectos, y la cubierta de protección (pantalla) solo puede conectarse a tierra en el anfitrión. En el cable sin apantallar todos los hilos son rectos. Las conexiones a 15 Mbps y superiores exigen cable apantallado.

Nota: Una forma de identificar que los concentradores (hubs) USB están realmente adaptados a la norma 2.0 es por su cable apantallado.

AWG	mm \varnothing	long. máx.
28	0.321	0.81 m
26	0.405	1.31 m
24	0.511	2.08 m
22	0.644	3.33 m
20	0.812	5.00 m

Cuadro 2.3: Tabla de diámetros de hilo para puerto USB

estándar para los hilos de alimentación del bus. Para cada sección se autoriza una longitud máxima del segmento. En la tabla izquierda se muestran estas distancias [3]; a la derecha se muestran la disposición de pines y colores de identificación.

Se usan dos tipos de conectores, A y B. Ambos son polarizados (solo pueden insertarse en una posición) y utilizan sistemas de presión para sujetarse. Los de tipo A utilizan la hembra en el sistema anfitrión, y suelen usarse en dispositivos en los que la conexión es permanente (por ejemplo, ratones y teclados). Los de tipo B utilizan la hembra en el dispositivo USB (función), y se utilizan en sistemas móviles (por ejemplo, cámaras fotográficas o altavoces). En general podemos afirmar que la hembra de los conectores A están en el lado del host (PC) o de los concentradores (hubs), mientras las de tipo B están del lado de los periféricos.

Capítulo 3

Configuración Software Complementario

3.1. Configuración de Arduino como grabador ISP

Al momento de trabajar con el microcontrolador, en microcontroladores de la firma Atmel, se puede elegir entre dos formas para trabajar, ya sea utilizando osciladores internos o ya sea osciladores externos, al momento de elegir osciladores internos se pueden utilizar velocidades desde 1Mhz hasta 8Mhz, y utilizando osciladores externos se pueden utilizar velocidades superiores a los 16Mhz fácilmente.

Nosotros utilizaremos la primera opción, utilizar el oscilador interno del microcontrolador, si bien no podremos realizar 16 millones de operaciones por segundo, pero en nuestro caso bastara con utilizar velocidades de 8Mhz, es así que emplearemos un oscilador RC interno, además se tiene una grandes ventajas al momento de emplear el oscilador interno, entre ellas las mas destacables son:

- Permite valores de tensión mucho mas pequeñas, permitiendo que su funcionamiento sea inclusive con baterías de 3.3V (baterías empleadas en relojes).
- Consume menos energía para ejecutar sus procesos (con un ahorro de al menos 20 % de energía).
- Procesos dos veces mas lentos (si se compara con el oscilador externo con el que los microcontroladores Atmel funciona por defecto 16Mhz) pero casi no muchos proyectos ameritan realizar más de 8 millones de instrucciones por segundo.

Para poder utilizar el microcontrolador con velocidades de 8Mhz es necesario realizar unos pequeños cambios en el las librerías de Arduino.

Cuando se desea emplear el microcontrolador fuera de la placa Arduino, es necesario grabar el Bootloader en el microcontrolador, pero para ello es nece-

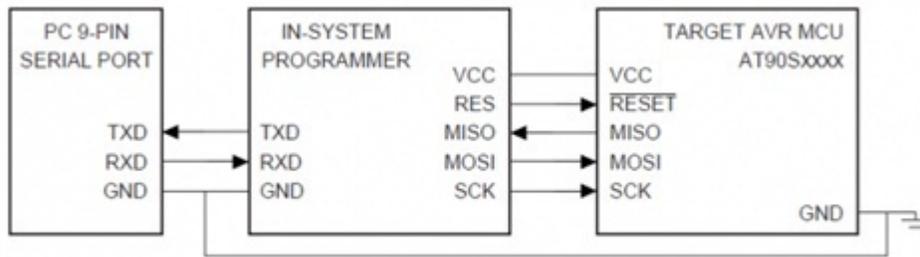


Figura 3.1: Comunicación ISP

sario convertir al Arduino, en un programador ISP, este tipo de programador (In-System Programming) es un método utilizado por Atmel para programar los microcontroladores una vez que ya han sido implementados en un sistema completo, es decir, permite programar los núcleos sin la necesidad de retirarlos de la placa de circuito.

El método de programación ISP (In-System Programming) utiliza el protocolo de comunicación SPI (Serial Peripheral Interface) estándar de comunicaciones empleado en transferencia de información empleando un flujo de bits serie regulado por un señal de reloj.

En los microcontroladores Atmel este bus SPI consiste en tres líneas de datos

- SCK (Serial Clock): Es la señal de reloj.
- MOSI (MasterOut-SlaveIn): Línea de transmisión unidireccional Maestro-Esclavo
- MISO (MasterIn-SlaveOut): Línea de transmisión unidireccional Esclavo-Maestro.

En la siguiente figura se puede observar las líneas de comunicación Maestro-Esclavo:

El primer bloque del esquema corresponde a la computadora que se comunica con el programador mediante comunicación serial virtualizando un puerto USB conectado al segundo bloque que corresponde al grabador ISP (Arduino Uno R2) mientras el tercer y ultimo bloque corresponde al microcontrolador que se desea programar, en este caso un microcontrolador Atmel328p.

Durante la grabación, el Arduino funciona como dispositivo “Maestro” y el circuito integrado será el dispositivo “Esclavo”. Durante cada pulso de reloj, un bit es enviado desde el maestro al esclavo mediante la línea MOSI al igual que es enviado un bit desde el esclavo hacia el maestro mediante la línea MISO, generándose de este modo la comunicación



Figura 3.2: Aplicaciones/Utilidades/Terminal

3.1.1. Instalación software complementario:

Antes de poder utilizar el Arduino como programador ISP, se debe primero realizar una configuración de permisos previa. El computador debe también contar con librerías de AVR-GCC, así como también librerías de AVR-DUDE independiente del sistema operativo que maneje el programador.

Para ello es necesario instalar un paquete propio de AVR para desarrolladores Macintosh, pero primero es necesario ubicar la ruta “`/usr/local/bin`” dentro del directorio principal, debido a que cuando se desea grabar microcontroladores el software de AVR-DUDE busca en este directorio específico y si se instala este programa pero no se agrega esta ruta, no serviría de nada el programa. Para ello es necesario realizar estos de carpeta en el Terminal para poder agregar la ruta “`/usr/local/bin`” en el directorio principal.

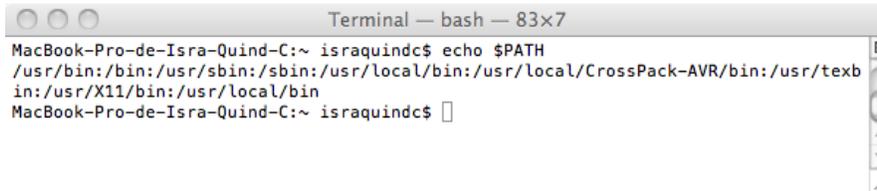
El **Terminal** de Macintosh se lo puede hallar en la carpeta: Aplicaciones/Utilidades, como se puede observar en la figura:

Una vez hallado el terminal se escribe `echo $SHELL` se presiona enter. Si se obtiene `/bin/bash` es necesario escribir el siguiente comando:

`echo 'PATH=$PATH:/usr/local/bin' >> ~/.bash_profile` en una sola línea y presionar enter. Si se obtiene `/bin/csh` o `/bin/tcsh` entonces es necesario el siguiente comando:

`echo 'set path = ($path /usr/local/bin)' >> ~/.cshrc` en una sola línea y presionar enter. Y con ello ha sido agregada la ruta en el directorio principal; el siguiente paso es cerrar el Terminal y abrirlo nuevamente para asegurar que `.bash_profile` o `.cshrc` ha sido recargado. Ahora se escribe `echo $PATH` (para bash) o `echo $path` (para t/csh). Obteniendo una respuesta como la de la siguiente figura:

La información relevante de la líneas es que al final termine con `/usr/local/bin`. De esta manera ya se pueden agregar el software de Atmel para poder utilizarlo.



```
Terminal — bash — 83x7
MacBook-Pro-de-Isra-Quind-C:~ israqindc$ echo $PATH
/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin:/usr/local/CrossPack-AVR/bin:/usr/textbin:/usr/X11/bin:/usr/local/bin
MacBook-Pro-de-Isra-Quind-C:~ israqindc$
```

Figura 3.3: echo \$PATH

Después de ello se tienen dos opciones para instalar el software de Atmel para poder utilizar el grabador ISP.

- OPCION 1

Instalar “CrossPack” entorno de desarrollo para AVR que incluye una suite de compiladores GNU para AVR, una biblioteca C, el cargador AVR-DUDE y otras utilidades de similares características. Este paquete se halla alojado en:

<http://www.obdev.at/products/crosspack/index.html>.

Este software no se utilizara directamente, sino que el Arduino necesita estas librerías para comunicarse con la computadora. Instalando este paquete no se necesitaría mas.

- OPCION 2

Cuando se producen complicaciones con el primer método, es decir, no se puede utilizar el grabador aun, se tiene una segunda opción que garantiza su utilización, para ello el primer paso es realizar la instalación de XCode y posteriormente OSX-AVR. Primero se debe realizar la descarga de las herramientas de desarrollo para Macintosh (XCode). Para ello es necesario descargar los paquetes oficiales de XCode que se los puede encontrar en:

<https://developer.apple.com/technologies/tools/>

El archivo de descarga tiene un peso cercano a los 4.4Gb. básicamente es necesario tener las herramientas de compilador nativo de Mac OS X para que se puedan generar las herramientas del compilador de AVR.

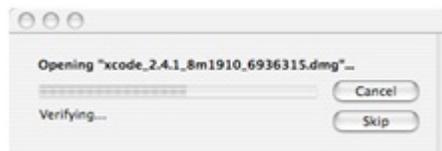


Figura 3.4: Verificación de compatibilidad



Figura 3.5: Instalación de Xcode Tools

El ultimo paso es realizar la instalación de OSX-AVR que se lo puede descargar directamente desde la pagina oficial del autor:

<http://sourceforge.net/projects/osxavr/files/osxavr/OSX-AVR-JAN07/>.

Home / osxavr / OSX-AVR-JAN07

Name	Modified	Size	
↑ Parent folder			
OSX-AVR-JAN07-X86.dmg	2007-02-25	19.6 MB	ⓘ 📄
OSX-AVR-JAN07-PPC.dmg	2007-02-25	26.9 MB	ⓘ 📄
Totals: 2 Items		46.5 MB	

Figura 3.6: Descarga pagina oficial

Una vez descargado el paquete, se lo instala y la configuración ha sido terminada.

Independiente de cual de los dos métodos se emplee, los privilegios para manejar el puerto USB para grabar los microcontroladores Atmel han sido habilitados.

3.1.2. Programador Arduino ISP 8Mhz:

Una vez instalado el software es necesario realizar cambios en las librerías del programa Arduino IDE para poder utilizar el microcontrolador Atmega328p con oscilador interno RC de 8Mhz.

Para agregar esta función es necesario modificar el gestor de arranque (bootloader) del Atmega328p.

Para poder modificar este archivo es necesario ir a: Aplicaciones/Arduino y dar clic derecho en la aplicación y seleccionar la opción “Mostrar el contenido del paquete” como se observa en la figura:



Figura 3.7: Arduino/Mostrar contenido del paquete

Dentro de esta dirección es necesario realizar cambios en el archivo boards.txt ubicado en:

Arduino/Contents/Resources/Java/hardware/Arduino/boards.txt Como se puede observar en la figura:

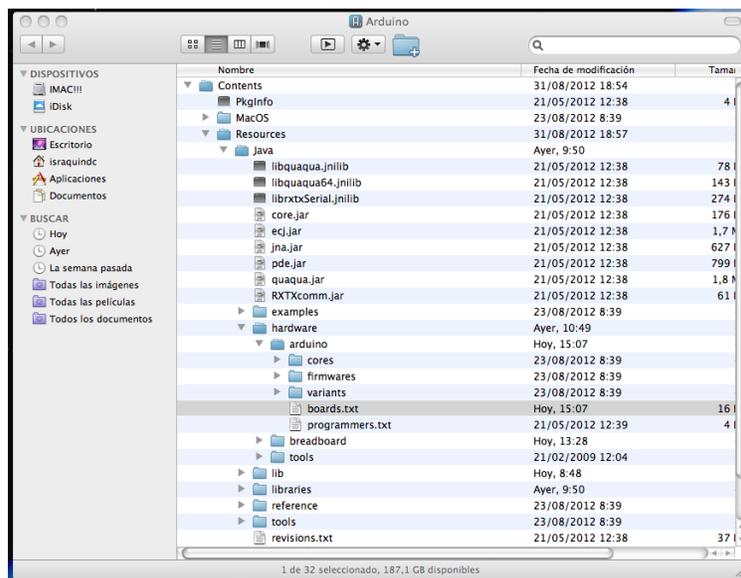


Figura 3.8: Dirección del archivo boards.txt

Este archivo mantiene las configuraciones necesarias de los microcontroladores soportados por Arduino IDE, como se puede observar en la figura.

```
boards.txt
# See: http://code.google.com/p/arduino/wiki/Platforms

#####

uno.name=Arduino Uno
uno.upload.protocol=arduino
uno.upload.maximum_size=32256
uno.upload.speed=115200
uno.bootloader.low_fuses=0xff
uno.bootloader.high_fuses=0xde
uno.bootloader.extended_fuses=0x05
uno.bootloader.path=optiboot
uno.bootloader.file=optiboot_atmega328.hex
uno.bootloader.unlock_bits=0x3F
uno.bootloader.lock_bits=0x0F
uno.build.mcu=atmega328p
uno.build.f_cpu=1600000L
uno.build.core=arduino
uno.build.variant=standard

#####

atmega328.name=Arduino Duemilanove w/ ATmega328

atmega328.upload.protocol=arduino
atmega328.upload.maximum_size=30720
atmega328.upload.speed=57600

atmega328.bootloader.low_fuses=0xFF
atmega328.bootloader.high_fuses=0xDA
```

Figura 3.9: Configuraciones del archivo boards.txt

Al final de este archivo solo es necesario agregar la configuración del microcontrolador para que el grabador reconozca la nueva velocidad de grabación para el microcontrolador Atmega328 para permitirle funcionar con oscilador de 8Mhz:

```
boards.txt

atmega8.bootloader.path=atmega8
atmega8.bootloader.file=ATmegaBOOT.hex
atmega8.bootloader.unlock_bits=0x3F
atmega8.bootloader.lock_bits=0x0F

atmega8.build.mcu=atmega8
atmega8.build.f_cpu=1600000L
atmega8.build.core=arduino
atmega8.build.variant=standard

#####

atmega328bb.name=ATmega328p en Proto!!! (8 Mhz clock interno)

atmega328bb.upload.protocol=arduino
atmega328bb.upload.maximum_size=30720
atmega328bb.upload.speed=57600

atmega328bb.bootloader.low_fuses=0xE2
atmega328bb.bootloader.high_fuses=0xD8
atmega328bb.bootloader.extended_fuses=0x07
atmega328bb.bootloader.path=atmega
atmega328bb.bootloader.file=ATmegaBOOT_168_ATMEGA328_pro_8Mhz.hex
atmega328bb.bootloader.unlock_bits=0x3F
atmega328bb.bootloader.lock_bits=0x0F

atmega328bb.build.mcu=atmega328p
atmega328bb.build.f_cpu=8000000L
atmega328bb.build.core=arduino:arduino
atmega328bb.build.variant=standard
```

Figura 3.10: Edición de las configuraciones del archivo boards.txt

Al final de este archivo solo es necesario agregar la configuración del microcontrolador Atmega328 para que soporte velocidades de oscilador de 8Mhz:

De esta manera se han realizado todas las configuraciones necesarias para grabar el microcontrolador mediante el Arduino, para la configuración del hardware lo primero que debe realizarse es ubicar tres resistencias de 220Ω y tres leds para poder conocer el estado del grabador ISP en todo momento a los pines digitales 7, 8, 9 como se observa en la figura.

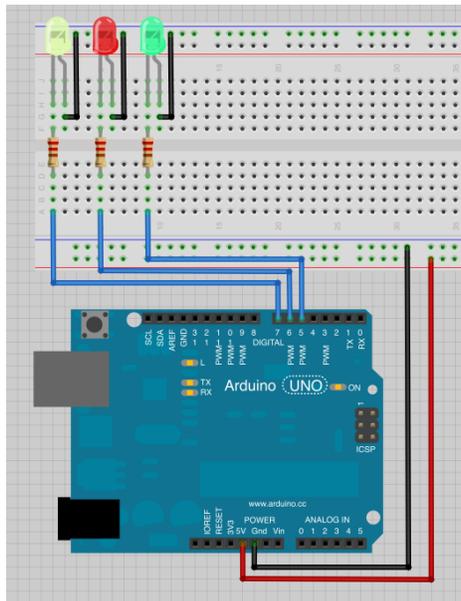


Figura 3.11: Leds grabador ISP

Ahora es necesario descargar el Sketch, para ello se abre **Arduino IDE** y a continuación el sketch “Ejemplos/ArduinoISP” que permite convertir al Arduino UNO en un grabador ISP, tal como se puede apreciar en la siguiente figura:

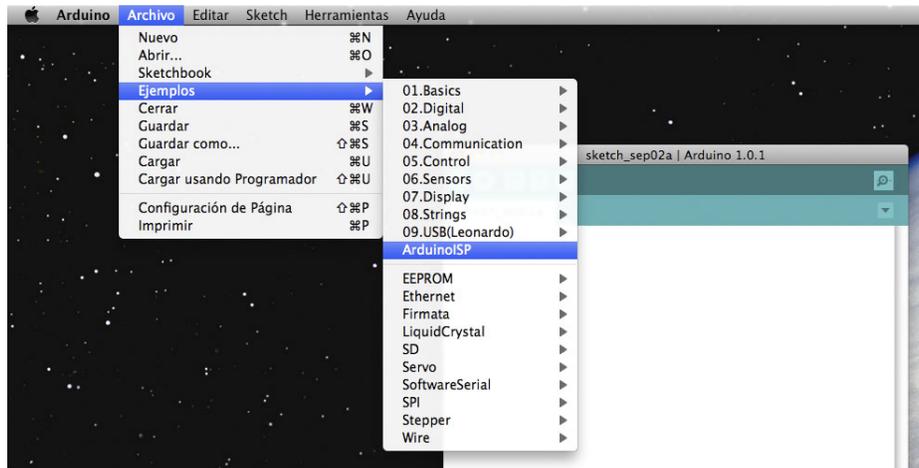


Figura 3.12: Sketch ArduinoISP

Una vez abierto el Sketch de Arduino ISP es necesario configurar la tarjeta, para ello en la barra de Herramientas se selecciona la opción Tarjeta/ArduinoUno como se observa en la figura:

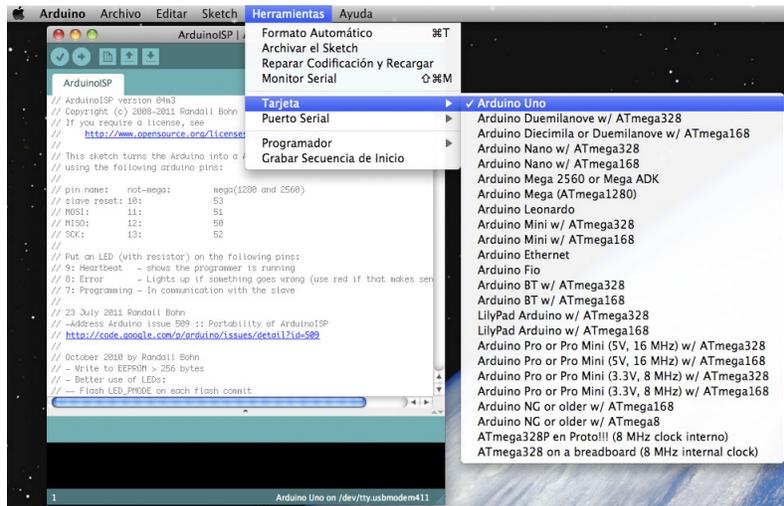


Figura 3.13: Tarjeta Arduino Uno

Luego se procede a seleccionar el puerto USB para poder realizar una correcta comunicación con el grabador Arduino UNO, para ello en la barra de Herramientas se selecciona la opción Puerto Serial/dev/tty.usbmodem411 como se observa en la figura:

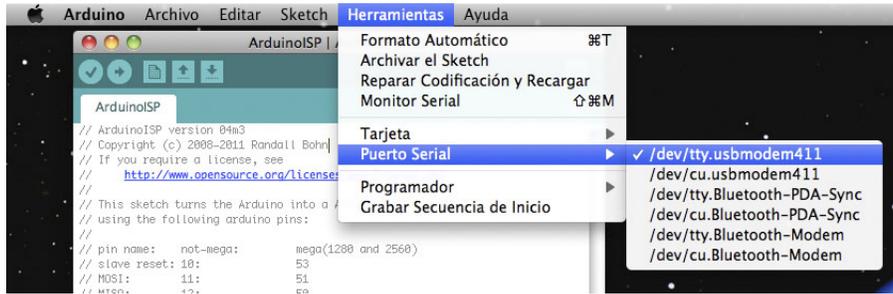


Figura 3.14: Puerto de Comunicación

A continuación es necesario compilar el archivo para verificar que no existen errores y finalmente descargar el sketch en el microcontrolador del Arduino, quedando de esta manera el Arduino configurado como Grabador ISP.



Figura 3.15: Verificación y Carga del Sketch

En la parte izquierda de la figura puede observarse el botón de color blanco se utiliza para la Verificación del Sketch, mientras en la parte derecha puede observarse el botón de color blanco, pertenece el botón para la Carga del Sketch.

3.1.3. Descarga Bootloader:

Una vez configurado el Arduino como grabador ISP, se procede a conectar el microcontrolador Atmega328 en el protoboard utilizando los pines para la comunicación ISP (MISO, MOSI, SCK, RESET) y un oscilador externo de 16Mhz, cabe recalcar que este oscilador solamente se utiliza para poder descargar el sketch, una vez cargado el Bootloader se prescinde del clock.

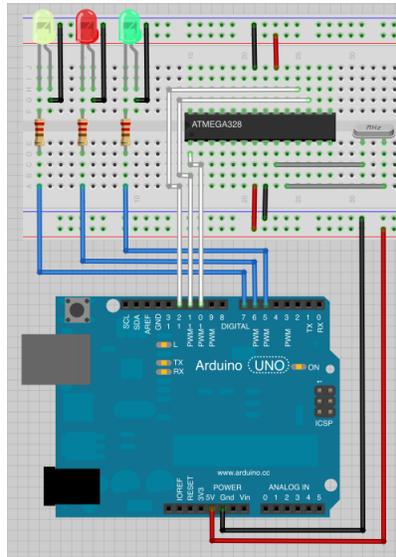


Figura 3.16: Arduino Grabador ISP

Para poder descargar el bootloader en el microcontrolador Atmega328, se debe cambiar la tarjeta del programador; ahora se selecciona al Atmega328P, para ello en la barra de herramientas se selecciona la opción Tarjeta/ATmega328P en Proto!!!(8 Mhz clock interno).

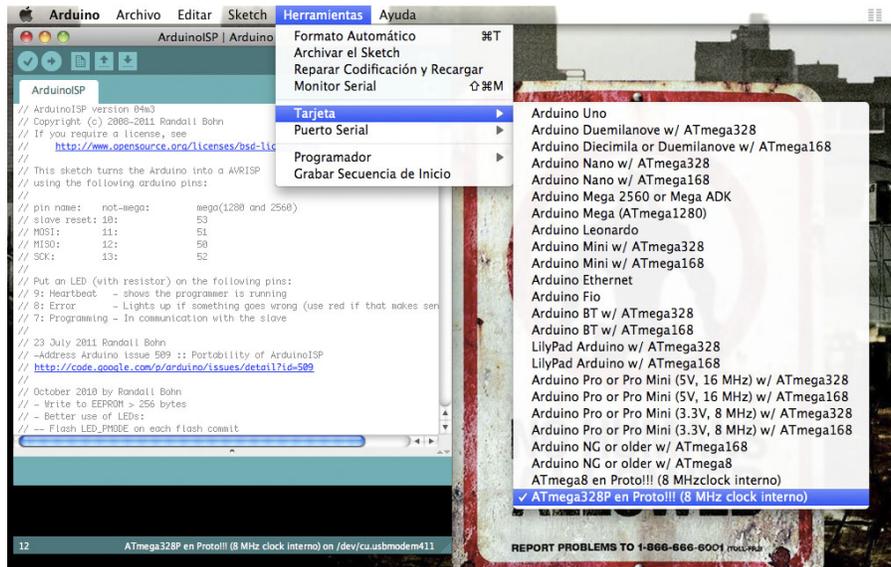


Figura 3.17: Selección del Microcontrolador Atmega328



Figura 3.18: Grabar Secuencia de Inicio

De esta manera ya solo resta descargar el bootloader en el microcontrolador, para ello en la barra de herramientas se selecciona “Grabar Secuencia de inicio”

3.1.4. Descarga de sketches en el microcontrolador

Una vez descargado el Bootloader en el microcontrolador, se pueden descargar los sketches en el mismo, para ello la forma de descarga cambia, así como los pines para descargar los sketches.

Los pines a emplear son los necesarios para la comunicación serial (TX, RX y RESET) además de la alimentación de 5V, y finalmente se remueve el microcontrolador de la tarjeta arduino, tal como se observa en la siguiente figura:

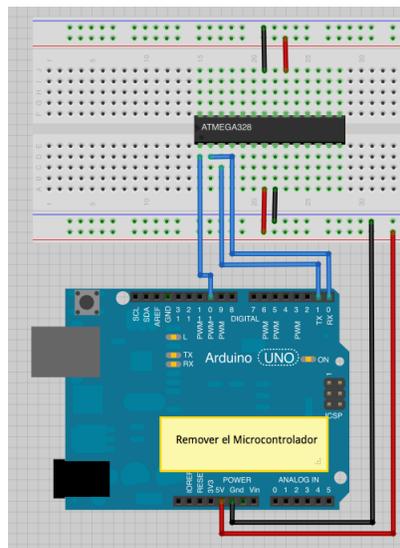


Figura 3.19: Descarga de Sketch en microcontrolador Atmega328

Cuando se ha transmitido el programa, se puede prescindir del oscilador de

16Mhz y utilizar exclusivamente el oscilador interno RC.

Una vez preparado el Hardware se procede a cargar los sketch en el microcontrolador. Para ello en el programa Arduino IDE se verifica que la tarjeta de selección sea el microcontrolador Atmega328 (8Mhz) en la barra de Herramientas/Tarjeta/Atmega328P como se observa en la figura:

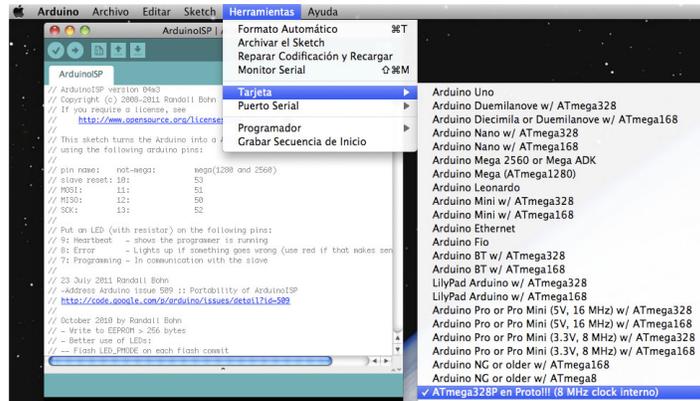


Figura 3.20: Tarjeta Atmega328p 8Mhz

De ahora en adelante cada vez que se desee grabar el microcontrolador Atmega328 se debe revisar que esta configuración sea la adecuada.

Capítulo 4

Configuración del Mouse

4.1. Esquema general del sistema:

A continuación se presenta el prototipo del proyecto que está compuesto por dos elementos:

- El transmisor recibe los datos de un acelerómetro y los transmite al receptor que se halla conectado a la computadora.
- El receptor convierte las variaciones del acelerómetro en coordenadas para modificar la posición del puntero.

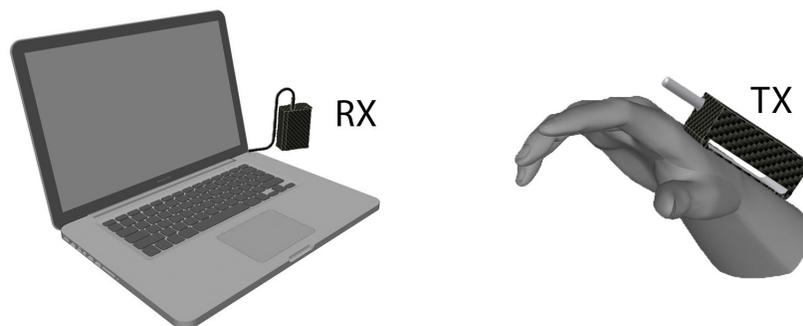


Figura 4.1: Esquema General del Prototipo

4.2. Comunicación:

4.2.1. Comunicación Serial:

Para lograr la comunicación serial entre dispositivos soportados por Arduino, es necesario implementar comunicación UART o USAR “Serial”, comunicación que se lleva a cabo por medio de los pines digitales 0 (RX) y 1 (TX).

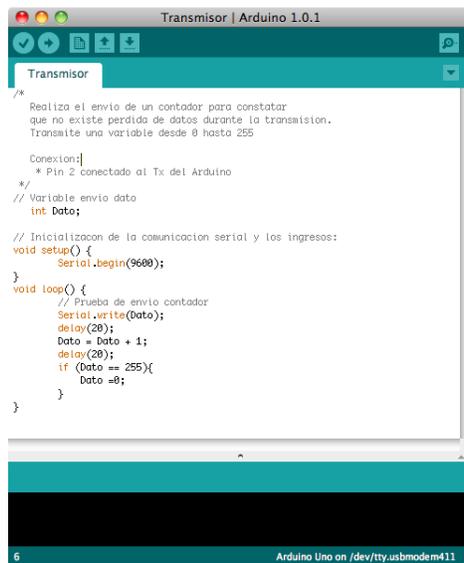
4.2.2. Comunicación Serial Cableada:

Este protocolo de comunicación se implementa para realizar la transmisión de datos entre los microcontroladores de forma cableada, para constatar que no exista pérdida alguna de datos y para que no lleguen datos basura de ningún otro lado, y posteriormente se implementa módulos RF que soportan el mismo protocolo.

Para comprobar la correcta transmisión de información desde el microcontrolador ATmega328 hacia la tarjeta Arduino, se desarrollan dos sketch.

En el primero se configura la transmisión de datos desde el microcontrolador ATmega328 utilizando una variable que cada vez que se envía se incrementa en uno, con inicio en el valor “0” y termina en el valor “255”.

Para ello es necesario configurar las opciones de la tarjeta, para que reconozca al Arduino Uno como receptor y al Arduino en protoboard como transmisor. Teniendo ello en mente se descarga el sketch correspondiente a la transmisión de datos en el Arduino en protoboard como se puede observar en la figura:

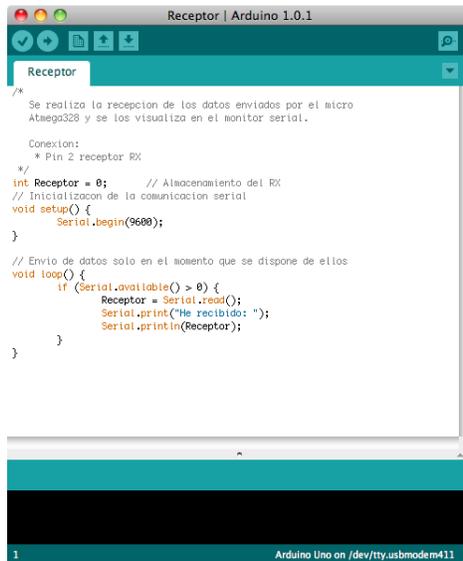


```
/*
 * Realiza el envío de un contador para constatar
 * que no existe pérdida de datos durante la transmisión.
 * Transmite una variable desde 0 hasta 255
 */
Conexion:|
 * Pin 2 conectado al Tx del Arduino
*/
// Variable envío dato
int Dato;

// Inicialización de la comunicación serial y los ingresos:
void setup() {
  Serial.begin(9600);
}
void loop() {
  // Prueba de envío contador
  Serial.write(Dato);
  delay(20);
  Dato = Dato + 1;
  delay(20);
  if (Dato == 255){
    Dato =0;
  }
}
```

Figura 4.2: Comunicación Serial Cableada, Modulo Transmisor

En el segundo sketch se configura la recepción de la información en la tarjeta Arduino UNO y posteriormente se la visualiza en el Monitor Serial como se observa en la siguiente figura:



```
Receptor | Arduino 1.0.1
Receptor
/*
 * Se realiza la recepcion de los datos enviados por el micro
 * Atmega328 y se los visualiza en el monitor serial.
 *
 * Conexion:
 * * Pin 2 receptor RX
 */
int Receptor = 0; // Almacenamiento del RX
// Inicializacion de la comunicacion serial
void setup() {
  Serial.begin(9600);
}

// Envio de datos solo en el momento que se dispone de ellos
void loop() {
  if (Serial.available() > 0) {
    Receptor = Serial.read();
    Serial.print("He recibido: ");
    Serial.println(Receptor);
  }
}
```

1 Arduino Uno on /dev/tty.usbmodem411

Figura 4.3: Sketch receptor Arduino Uno

Una vez establecidos los requerimientos para la comunicación se procede a descargar el sketch del Receptor en la tarjeta Arduino Uno y el sketch del transmisor en el microcontrolador Atmega328 y se arma el hardware de la siguiente figura:

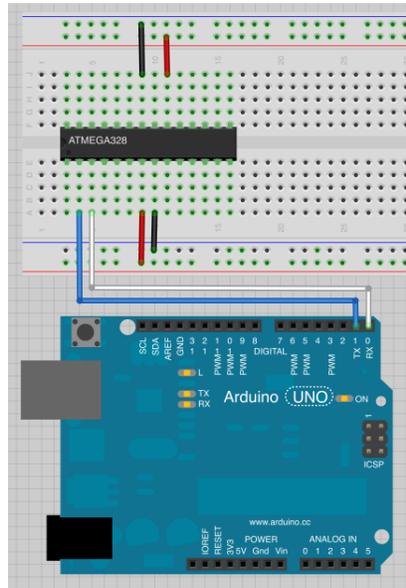


Figura 4.4: Hardware Comunicación Serial Cableada

Manteniendo este esquema se procede a conectar la tarjeta Arduino UNO con el computador y activar el “Monitor Serial” en la interfaz de Arduino IDE, como se aprecia en la gráfica:



Figura 4.5: Monitor Serial

Este terminal debe ser configurado para que acepte comunicación serial a una velocidad de 9600 baudios. En él se puede observar los datos recibidos por el Arduino Uno, procedentes del Atmega328.

En el Monitor Serial, inmediatamente se puede visualizar los datos que han sido enviados desde el microcontrolador hacia la tarjeta, tal como se puede observar en la siguiente figura.

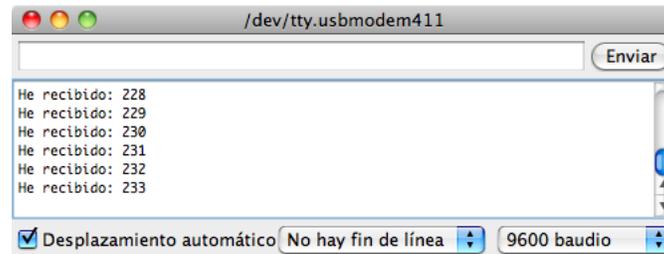


Figura 4.6: Datos recibidos por el Arduino

4.2.2.1. Lectura Acelerómetro:

El sensor Memsic2125 es un acelerómetro de dos ejes, de tipo térmico, capaz de medir la inclinación, la colisión, la aceleración estática y dinámica, la rotación y vibración con un rango de ± 3 g. De iguales características al esquema de la imagen.

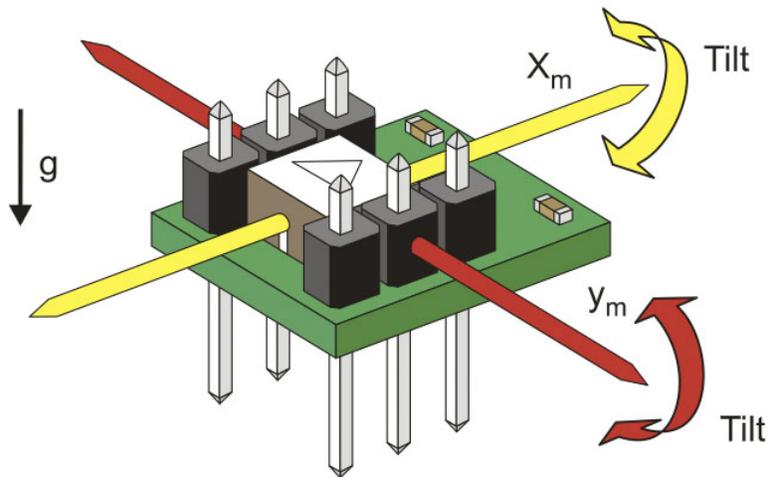


Figura 4.7: Acelerómetro 2ejes

Este sensor dispone de una cámara de gas con un elemento de calentamiento en el centro y cuatro sensores de temperatura alrededor de su borde. Si el acelerómetro se halla nivelado, la burbuja de gas caliente se eleva a la parte superior central de la cámara y de esta forma los sensores medirán la misma temperatura, como se puede observar en la figura (a); si el acelerómetro no está nivelado, la burbuja de aire será desplazada hacia una esquina específica, causando de esta manera que uno o más sensores midan un menor nivel de temperatura en

la esquina que se halla a menor altura, como se puede observar en la figura (b).

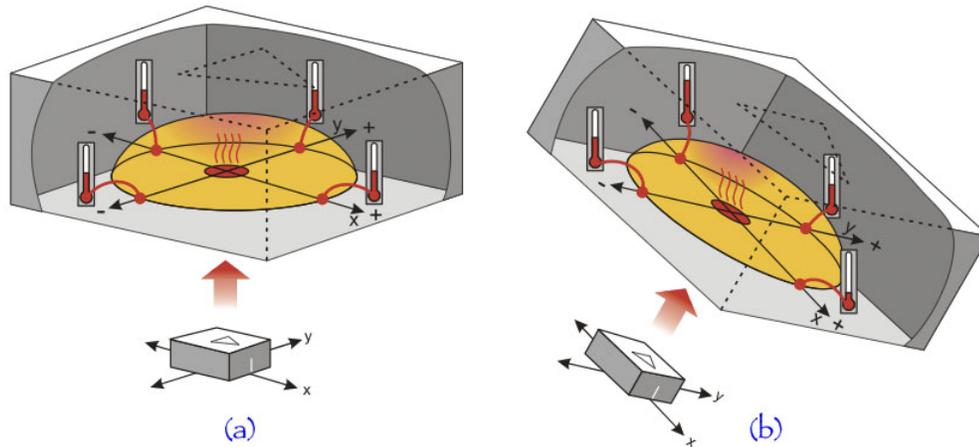


Figura 4.8: Principio de funcionamiento del Acelerómetro

Mediante la comparación de las temperaturas de los sensores de aceleración, tanto estática (gravedad e inclinación) y la aceleración dinámica (movimiento lineal) puede ser detectado, convirtiendo las mediciones de temperatura en señales pulsantes (duración del pulso PWM).

Las lecturas de cada eje, tienen una frecuencia de trabajo de 100Hz, en el cual la aceleración es proporcional a la relación t_{Hx}/T_x , manteniendo un comportamiento de PWM como se observa en la siguiente figura:

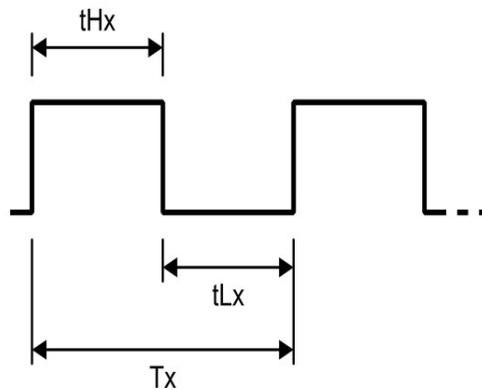


Figura 4.9: Pulso emitido por el acelerómetro

Con $V_{dd} = 5V$, 50% del ciclo de trabajo corresponde a una fuerza g de 0, que puede variar con cada unidad individual dentro de un intervalo de 48,7% a

51,3%; este offset del origen puede ser diferente cuando se utiliza un $V_{dd} = 3,3$ V.

El siguiente paso para realizar pruebas de medición del sensor es armar el siguiente esquema de conexión, en el cual se ha conectado el acelerómetro a los pines digitales 3 y 4 del Arduino UNO, como se observa en el esquema de siguiente figura:

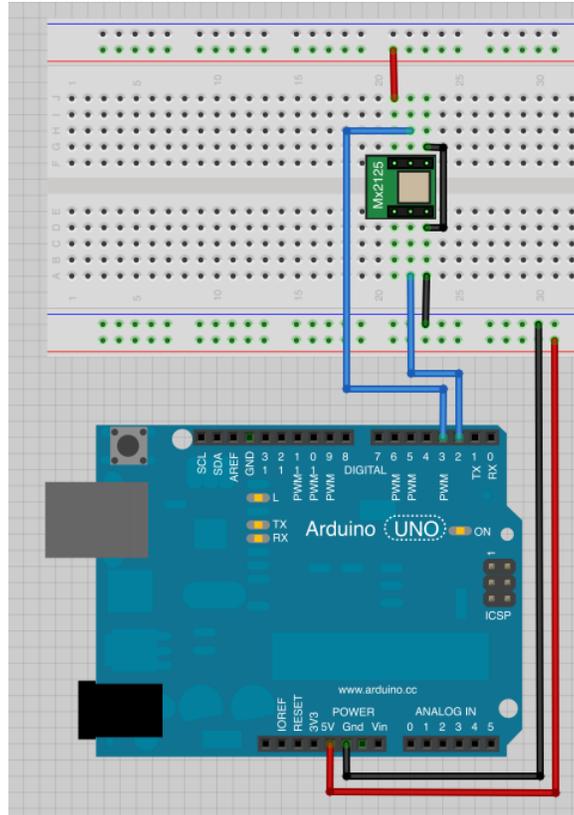


Figura 4.10: Conexión Acelerómetro al Arduino

Una vez armado el circuito, es hora de implementar el Sketch que se observa en la siguiente figura:



```

Acelerometro_Memsic2125 | Arduino 1.0.1
Acelerometro_Memsic2125 $
const int xPin = 2; // Lector eje X pin digital
const int yPin = 3; // Lector eje X pin digital

// Inicializacion de la comunicacion serial y los ingresos:
void setup() {
  Serial.begin(9600);
  pinMode(xPin, INPUT);
  pinMode(yPin, INPUT);
}

// Variables para la lectura del pulso, contenedores de los
// resultados y lectura del pulso de los ejes "x" y "y"
void loop() {
  int pulseX, pulseY;
  int AccX, AccY;
  pulseX = pulseIn(xPin,HIGH);
  pulseY = pulseIn(yPin,HIGH);

  // Conversion del ancho de pulso en aceleracion
  // Conversion en milisimas de atraccion de la tierra "1g"
  // Tratamiento de la señal para tener terminos distinguibles
  AccX = ((pulseX / 10) - 500) * 8;
  AccY = ((pulseY / 10) - 500) * 8;

  // Muestra la aceleracion en el eje "X" y en el eje "Y"
  Serial.print(AccX);
  Serial.print("\t");
  Serial.print(AccY);
  Serial.println();
  delay(100);
}

Guardado Terminado.
Tamaño binario del Sketch: 3.230 bytes (de un máximo de 32.256 bytes)

1 Arduino Uno on /dev/tty.usbmodem411

```

Figura 4.11: Código lectura acelerómetro

Los valores leídos por los puertos digitales son almacenados en las variables pulseX y pulseY, luego son tratados y la señal PWM es convertida en valores distinguibles para poder diferenciar los desplazamientos en ambos ejes, obteniendo valores legibles entre -1500 y 1500 dependiendo si se esta con una inclinación hacia la izquierda o hacia la derecha, dependiendo del eje ("x" o "y"), finalmente estos datos son enviados por el puerto USB y visualizados en la computadora mediante el monitor serial de Arduino IDE, como se observa en la figura:



```

/dev/tty.usbmodem411
Enviar
720 712
728 712
728 712
720 720
720 720
720 712

 Desplazamiento automático No hay fin de línea 9600 baudio

```

Figura 4.12: Lectura de datos de los ejes del acelerómetro

En la figura anterior los valores del lado izquierdo corresponden a los datos de lectura del eje “x”, mientras los valores del lado derecho corresponden a la lectura del eje “y”.

Una vez comprobado el funcionamiento y el correcto envío de los datos leídos por el arduino es hora de implementar el mismo programa en un circuito que trabaja a una frecuencia de 8Mhz y comunica los datos a la computadora utilizando comunicación serial.

Para ello primero se arma el hardware de la figura:

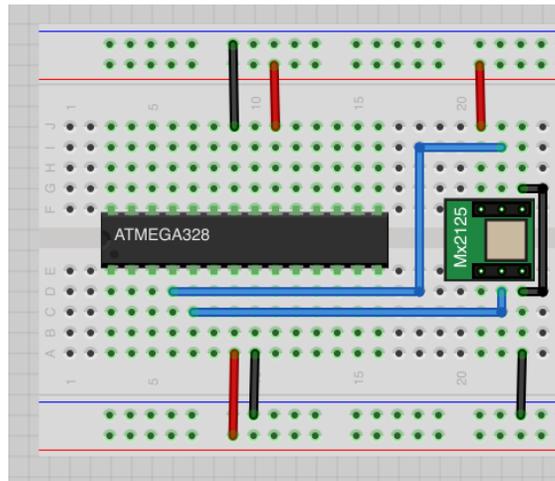
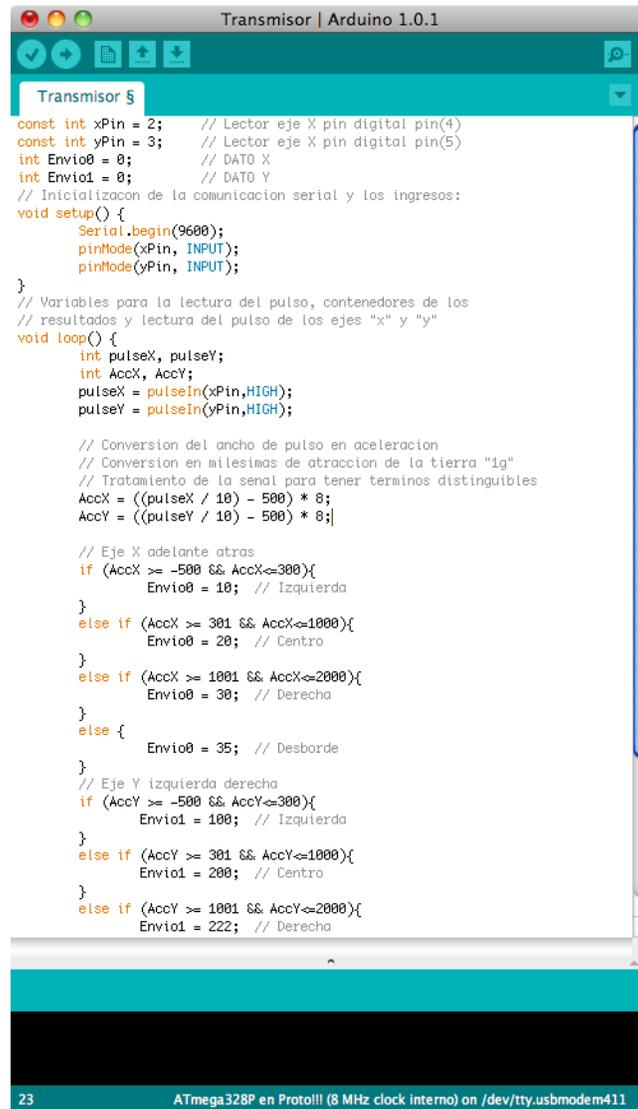


Figura 4.13: Acelerómetro en Atmega328

En el microcontrolador Atmega328 se descarga un sketch de lectura del acelerómetro y se incrementa la secuencia de la comunicación de datos serial, como se observa en la siguiente figura.



```

Transmisor 5
const int xPin = 2; // Lector eje X pin digital pin(4)
const int yPin = 3; // Lector eje X pin digital pin(5)
int Envio0 = 0; // DATO X
int Envio1 = 0; // DATO Y
// Inicializaco de la comunicacion serial y los ingresos:
void setup() {
  Serial.begin(9600);
  pinMode(xPin, INPUT);
  pinMode(yPin, INPUT);
}
// Variables para la lectura del pulso, contenedores de los
// resultados y lectura del pulso de los ejes "x" y "y"
void loop() {
  int pulseX, pulseY;
  int AccX, AccY;
  pulseX = pulseIn(xPin,HIGH);
  pulseY = pulseIn(yPin,HIGH);

  // Conversion del ancho de pulso en aceleracion
  // Conversion en milisimas de atraccion de la tierra "1g"
  // Tratamiento de la senal para tener terminos distinguibles
  AccX = ((pulseX / 10) - 500) * 8;
  AccY = ((pulseY / 10) - 500) * 8;

  // Eje X adelante atras
  if (AccX >= -500 && AccX<=300){
    Envio0 = 10; // Izquierda
  }
  else if (AccX >= 301 && AccX<=1000){
    Envio0 = 20; // Centro
  }
  else if (AccX >= 1001 && AccX<=2000){
    Envio0 = 30; // Derecha
  }
  else {
    Envio0 = 35; // Desborde
  }
  // Eje Y izquierda derecha
  if (AccY >= -500 && AccY<=300){
    Envio1 = 100; // Izquierda
  }
  else if (AccY >= 301 && AccY<=1000){
    Envio1 = 200; // Centro
  }
  else if (AccY >= 1001 && AccY<=2000){
    Envio1 = 222; // Derecha
  }
}
23 ATmega328P en Protoll (8 MHz clock interno) on /dev/tty.usbmodem411

```

Figura 4.14: Transmisor con lectura del Acelerómetro

El Sketch del transmisor se configura para recibir los datos del acelerómetro, discriminarlos por rangos para determinar la posición del puntero y finalmente enviar uno de ocho posibles valores para la variable.

Se determinan las variables globales y posteriormente se habilita la comunicación serial a una velocidad de 9600 baudios para que tenga compatibilidad con la placa Arduino Uno y con la computadora. Posteriormente se realiza la lectura digital del ancho de pulso para el eje “x” y para el eje “y”, luego se trata el dato para obtener valores dentro de un rango de -1500 a 1500 almacenando

estos valores en variables tipo “int”, de esta manera es mas sencillo saber si existe movimiento o no en cada eje para hacer consultas puntuales de los rangos de trabajo para determinar el desplazamiento (izquierda, derecha, arriba, abajo) y finalmente transmitir el dato que ha ocurrido en ese instante tanto para el eje “x” como para el eje “y”, teniendo la precaución de disponer de un tiempo prudente de 20 milisegundos entre cada transmisión, para que no exista colisión de datos ni datos basura lleguen al receptor.

Para el receptor se implementa un sketch, con el cual se reciben los datos mediante la comunicación serial y se los reenvía a la computadora para visualizarlos en el Monitor serial como se puede apreciar en al siguiente figura:



```

Receptor | Arduino 1.0.1

Receptor §
int Receptor = 0; // Almacenamiento del RX
int Envio = 0; // Muestra dato recibido

// Inicializa de la comunicacion serial
void setup() {
  Serial.begin(9600);
}

// Envio de datos solo en el momento que se dispone de ellos
void loop() {
  if (Serial.available() > 0) {
    Receptor = Serial.read();
    // Serial.print("He recibido: ");
    // Serial.println(Receptor);

    // Eje X adelante atras
    if (Receptor == 10){
      Envio = 10; // Izquierda
    }
    else if (Receptor == 20){
      Envio = 20; // Centro
    }
    else if (Receptor == 30){
      Envio = 30; // Derecha
    }
    else if (Receptor == 100){
      Envio = 100; // Izquierda
    }
    else if (Receptor == 200){
      Envio = 200; // Centro
    }
    else if (Receptor == 222){
      Envio = 222; // Derecha
    }
    else {
      Envio = 35; // incompatible
    }
    Serial.println(Envio); //movimiento en eje x
  }
}
1
Arduino Leonardo on /dev/tty.usbmodem411

```

Figura 4.15: Receptor

En el Sketch del Receptor primero se realiza la configuración de la velocidad de la comunicación serial, estableciéndola en 9600 baudios, luego se consulta

la disponibilidad de datos en el puerto para ejecutar las acciones solo cuando existan datos disponibles y no en todo momento para de esta manera no saturar el puerto.

Cuando llegan los datos al puerto, se realiza una consulta para determinar cual será el comportamiento del puntero, existen ocho valores posibles que determinan el movimiento y están clasificados de la siguiente forma:

- Tres valores de variables que determinan si existe movimiento en el eje “X” (desplazamiento izquierda, desplazamiento derecha o no existe cambio)
- Tres valores de variables para determinar si existe movimiento en el eje “Y”,(desplazamiento arriba, desplazamiento abajo o no existe cambio)
- Una variable en la cual se almacenarían si existen datos basura
- Una variable para determinar si se ha producido un click izquierdo en el mouse.

Una vez descargados los sketches correspondientes, se tiene el siguiente esquema de conexión entre el Arduino uno y el Arduino en protoboard:

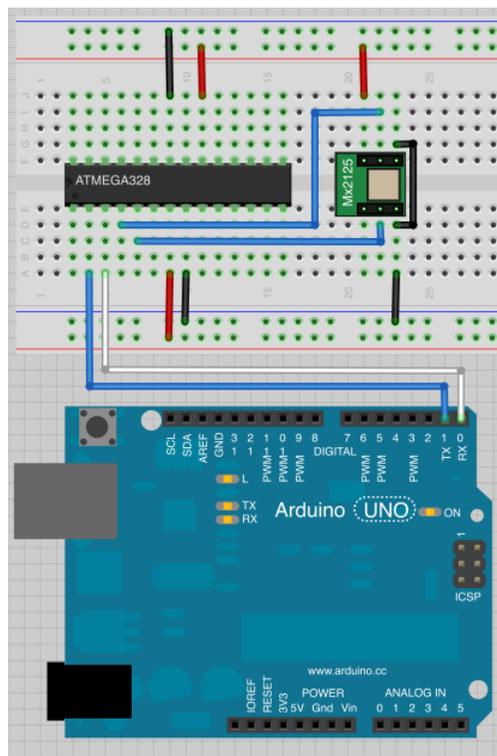


Figura 4.16: Comunicación Transmisor con Arduino UNO

Activando el Monitor Serial de Arduino IDE, se puede observar los datos recibidos por el Arduino Uno, procedentes del Arduino en protoboard, como se aprecia en la siguiente figura:

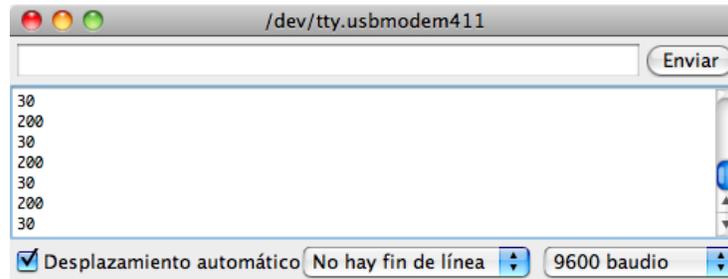


Figura 4.17: Monitor Serial Lectura de Datos

En la imagen puede observarse que cada 20ms, la computadora recibe un dato y lo muestra en el Monitor Serial, por ejemplo en este momento se obtuvieron los valores de “30” y “200”. Interpretando estos valores, se conoce que “30” indica que existe un desplazamiento en el eje “x” hacia la derecha, mientras que el valor de “200” representa un estado de no desplazamiento para el eje “y”.

4.2.3. Comunicación Serial Inalámbrica:

La comunicación de los microcontroladores con las antenas emplea el protocolo serial, para el proyecto se emplearan módulos full duplex “Micropower Data RF Module”, como se puede apreciar en la figura:



Figura 4.18: Modulo Transmisión RF

Estos módulos RF tienen la ventaja de ser configurables, permitiendo que se pueda realizar la configuración pertinente, dependiendo de las necesidades del hardware; permite configurar la velocidad de comunicación del módulo RF con el controlador, la velocidad de comunicación entre módulos RF, la paridad (No empleada) además permite la elección entre once canales posibles, si además a esto adicionamos la variedad de comunicación entre antenas con respecto a su velocidad (air Baud), conseguimos la posibilidad de implementar cuarenta frecuencias distintas, sin que ninguna de ellas se cause interferencia. Para la configuración de los módulos RF es necesario un software especial (Software of FC201S), que permite cambiar los parámetros de comunicación serial, para ello es necesario especificar el puerto serial en el cual ha sido conectado el módulo RF tal cual se observa en la siguiente figura:

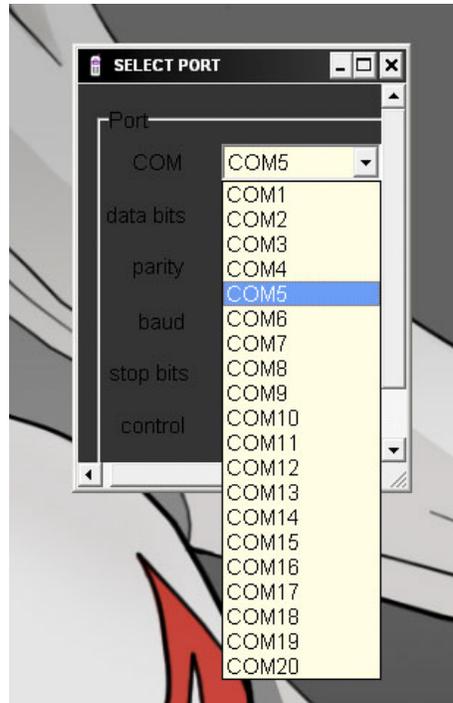


Figura 4.19: Configuración de Puerto lectura Modulo RF

Acto siguiente, de seleccionar el puerto serial de comunicación, se procede a realizar la lectura del módulo RF, para constatar que los módulos a emplear tengan la misma configuración, y que de esta manera pueda enlazarse la comunicación entre ellas y sus microcontroladores. En la siguiente figura se puede observar como se realiza la lectura del módulo RF; para ello se va a “menú principal/program/read”

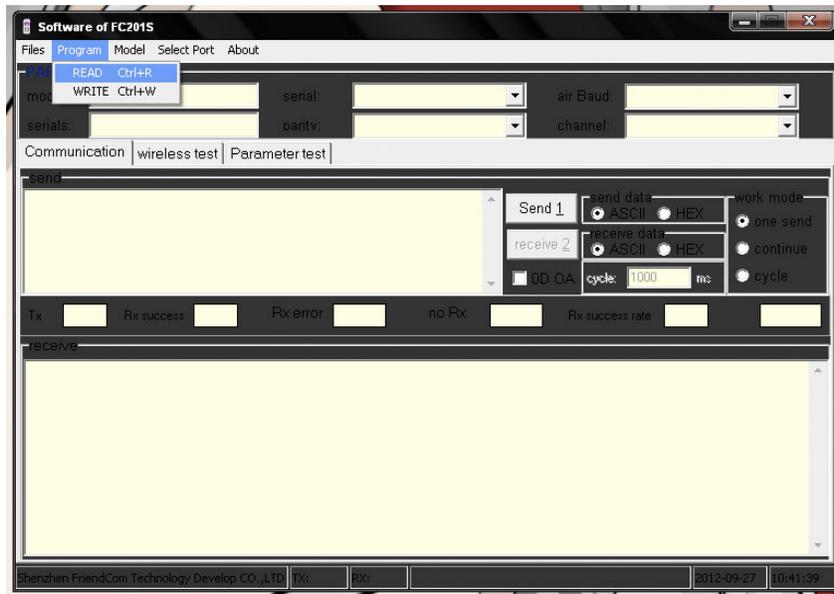


Figura 4.20: Lectura de Módulos RF

En la gráfica se puede observar los parámetros con los que están configurados los módulos RF:

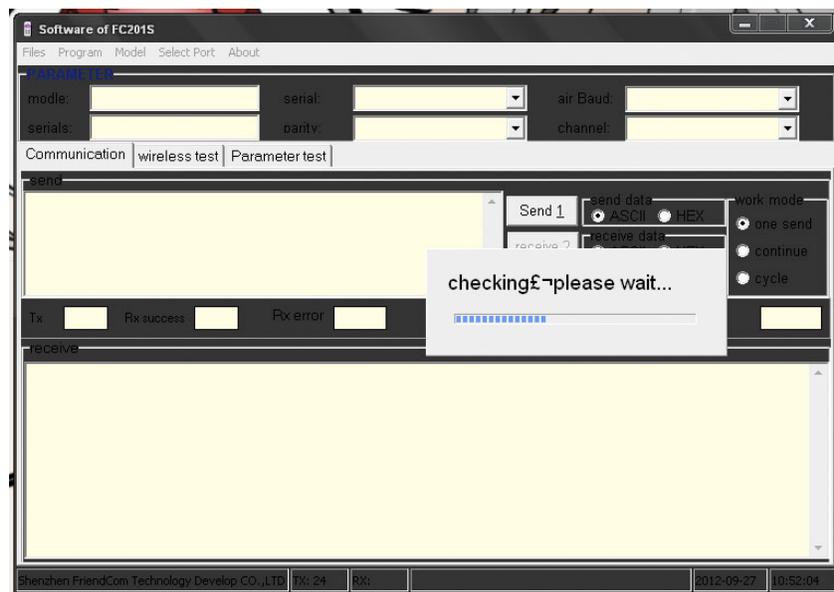


Figura 4.21: Lectura de parámetros de fabrica del Modulo RF

Para el proyecto no se utilizara paridad, y el canal a emplear será el canal 5, en este gráfico se puede observar cómo se cambia la paridad de toda a sin paridad:

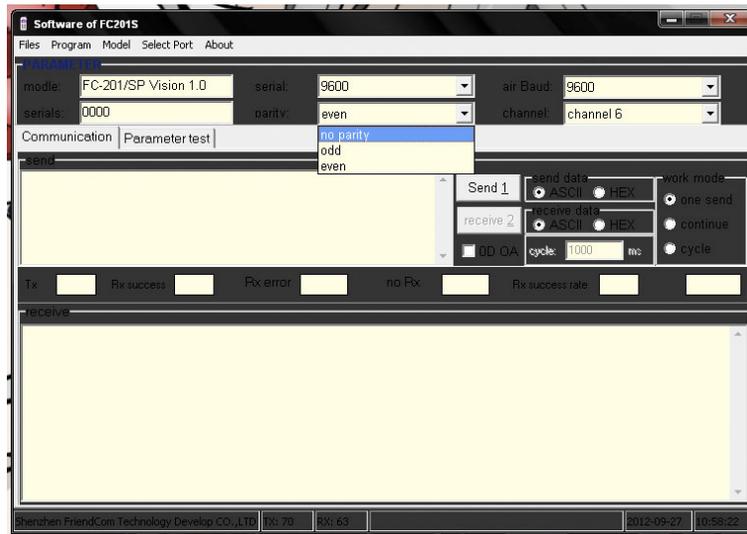


Figura 4.22: Configuración de Paridad

En la siguiente gráfica se puede ver la configuración del canal de envío de datos:

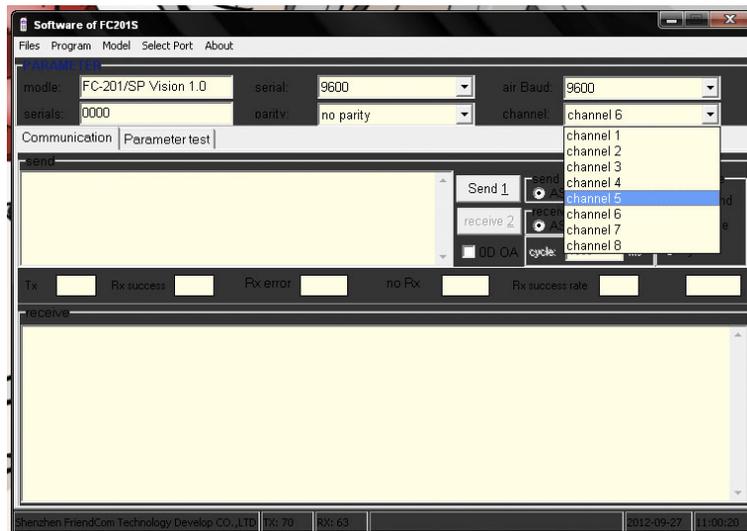


Figura 4.23: Configuración de Canal

En la siguiente figura se puede constatar la configuración empleada para la comunicación de los datos, y la plantilla a considerar para los módulos RF:

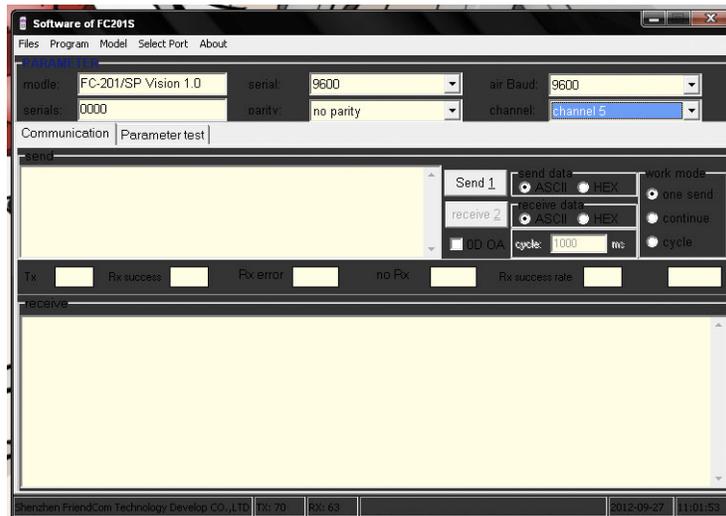


Figura 4.24: Configuración Final del Modulo RF

Una vez realizada la configuración de las antenas, su empleo es sencillo.

Realizada y constatada la correcta configuración del Modulo RF, se procede a descargar la configuración, para ello se va a “menú principal/program/write” como se observa en la siguiente figura:

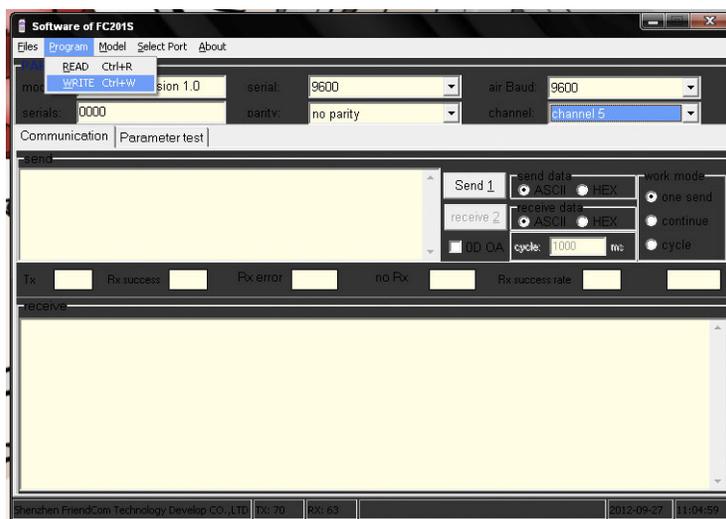


Figura 4.25: Escritura de la Configuración en el Modulo RF

El mismo procedimiento se realiza para descargar la configuración en los dos Módulos RF que se emplearan y de esta manera ha sido configurada la comunicación para el envío de datos entre los módulos RF.

Respetando los pines de comunicación, se realiza la conexión de ambos módulos RF. Un modulo se conecta con el microcontrolador Atmega328 y sera el modulo transmisor, el segundo modulo se conecta con el Arduino UNO para ser el modulo receptor, quedando el hardware como se observa en la siguiente figura:

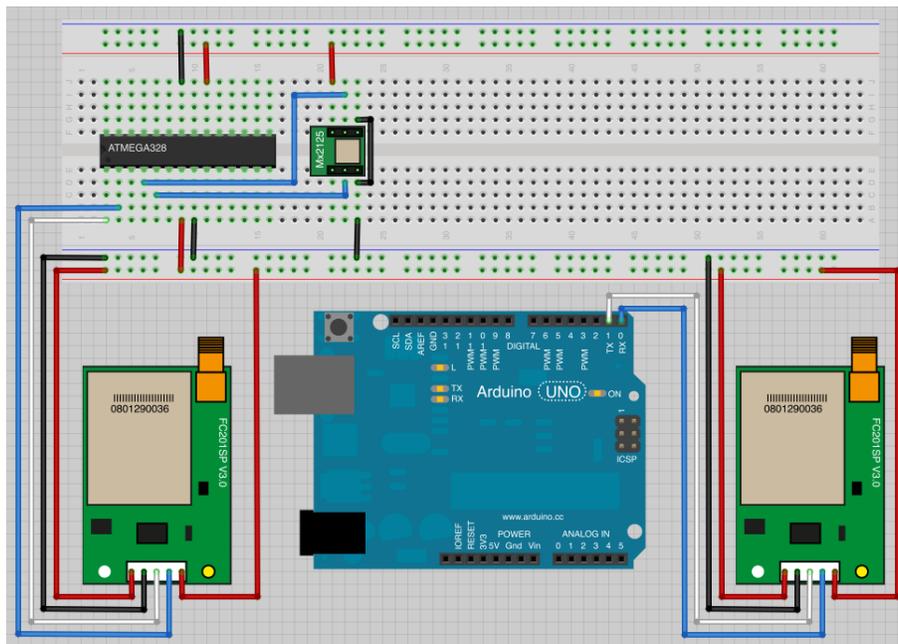
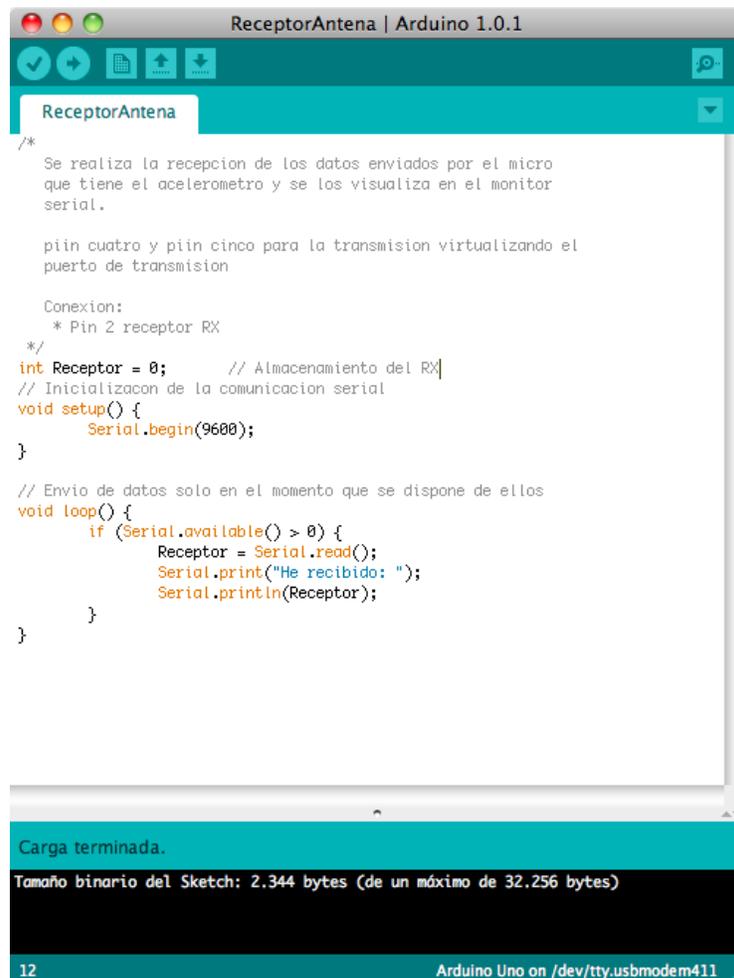


Figura 4.26: Conexión de los módulos Transmisor y Receptor.

Como se puede ver en la figura del receptor, se crea un sketch simple, pero útil para comprobar que no exista perdidas durante la transmisión o durante la recepción de los datos:



```
ReceptorAntena | Arduino 1.0.1
ReceptorAntena
/*
 Se realiza la recepcion de los datos enviados por el micro
 que tiene el acelerometro y se los visualiza en el monitor
 serial.

 piin cuatro y piin cinco para la transmision virtualizando el
 puerto de transmision

 Conexion:
 * Pin 2 receptor RX
 */
int Receptor = 0; // Almacenamiento del RX
// Inicializacion de la comunicacion serial
void setup() {
  Serial.begin(9600);
}

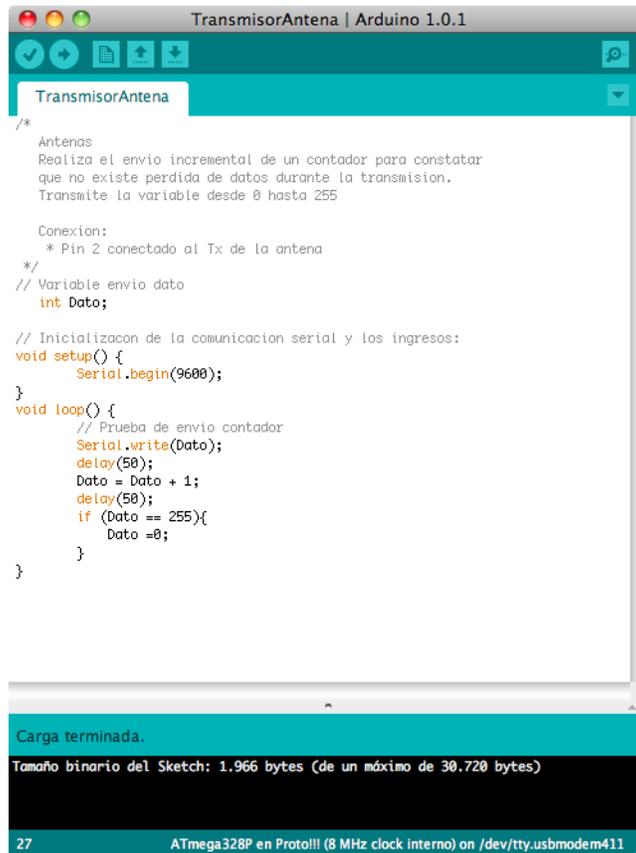
// Envio de datos solo en el momento que se dispone de ellos
void loop() {
  if (Serial.available() > 0) {
    Receptor = Serial.read();
    Serial.print("He recibido: ");
    Serial.println(Receptor);
  }
}

Carga terminada.
Tamaño binario del Sketch: 2.344 bytes (de un máximo de 32.256 bytes)
12 Arduino Uno on /dev/tty.usbmodem411
```

Figura 4.27: Receptor Antena

En este sketch se realiza la recepción de los datos del puerto serial y se los visualiza en el monitor serial

En el sketch del transmisor se realiza una transmisión cada 50 milisegundos de un dato entero entre 0-255, para de esta manera poder saber si existe o no perdida durante la comunicación de los microcontroladores, tal como se aprecia en la siguiente figura:



```
TransmisorAntena | Arduino 1.0.1
TransmisorAntena
/*
  Antenas
  Realiza el envío incremental de un contador para constatar
  que no existe perdida de datos durante la transmisión.
  Transmite la variable desde 0 hasta 255

  Conexion:
  * Pin 2 conectado al Tx de la antena
*/
// Variable envío dato
int Dato;

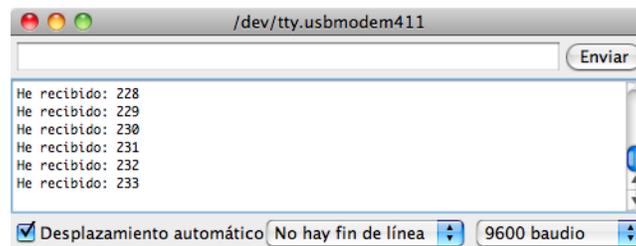
// Inicialización de la comunicación serial y los ingresos:
void setup() {
  Serial.begin(9600);
}
void loop() {
  // Prueba de envío contador
  Serial.write(Dato);
  delay(50);
  Dato = Dato + 1;
  delay(50);
  if (Dato == 255){
    Dato = 0;
  }
}

Carga terminada.
Tamaño binario del Sketch: 1.966 bytes (de un máximo de 30.720 bytes)
27 ATmega328P en Protoll! (8 MHz clock interno) on /dev/tty.usbmodem411
```

Figura 4.28: Transmisor Antena

Una vez desarrollados los sketch se los descarga respectivamente en el Arduino Uno y en el Atmega328 del protoboard para proceder a realizar las pruebas respectivas.

En la gráfica se puede constatar un extracto de la comunicación de los módulos RF.



```
/dev/tty.usbmodem411
Enviar
He recibido: 228
He recibido: 229
He recibido: 230
He recibido: 231
He recibido: 232
He recibido: 233
 Desplazamiento automático No hay fin de línea 9600 baudio
```

Figura 4.29: Recepción de Datos en la computadora

4.3. Comunicación USB:

Mediante el Arduino Leonardo se logra implementar la comunicación USB gracias a la utilización de un microcontrolador ATmega32U4 incorporado a la placa, que le permite mantener un diseño mucho mas sencillo y permite realizar la comunicación USB directa y no por segundos microcontroladores para virtualizar el puerto, como es el caso de la tarjeta Arduino UNO.

Para el modulo transmisor se mantendrá el mismo hardware y software que se utilizo en la comunicación serial, mientras que para el transmisor, la tarjeta Arduino UNO sera reemplazada por la tarjeta Arduino Leonardo, como se observa en la siguiente figura:

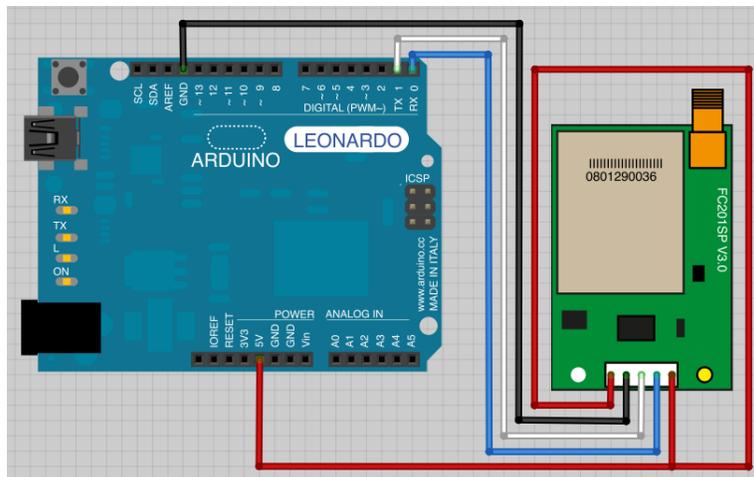


Figura 4.30: Receptor Arduino Leonardo

Finalmente el hardware a emplear debe quedar como el presentado en la siguiente figura:

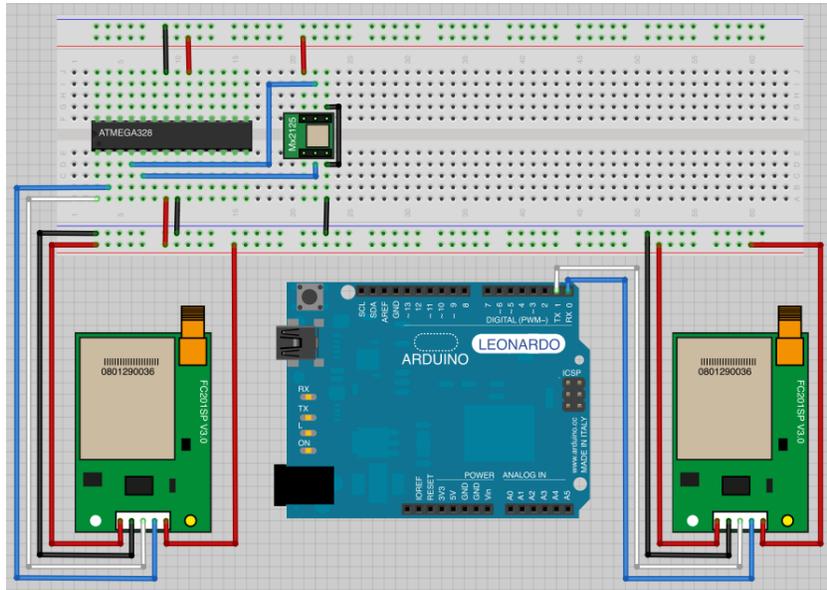


Figura 4.31: Transmisor, Receptor (Arduino Leonardo)

Para poder descargar el Sketch en el hardware, la tarjeta de selección debe ser el Arduino Leonardo, el puerto de comunicación se mantiene y el programador se mantiene también en AVRISP mkll. En la siguiente figura puede observarse el código de programación del Arduino:

```

Mouse | Arduino 1.0.1

Mouse

/*
 * Mouse Arduino Leonardo por lectura datos seriales
 * Control del mouse por la lectura de codigos recibidos serialmente.
 * Hardware:
 *   * Comunicacion Serial "RX"
 */

int Rango = 5;           // Velocidad para el desplazamiento
int TiempoEspera = 10;  // Tiempo de respuesta del mouse (milisegundos)
int Receptor = 0;       // Almacenamiento del RX
int Arriba = LOW;
int Abajo = LOW;
int Izquierda = LOW;
int Derecha = LOW;
int ClickIz = LOW;

void setup() {
  // initialize mouse control:
  Mouse.begin();
  Serial1.begin(9600);
}
    
```

Figura 4.32: Configuración Arduino Leonardo

En la configuración de la imagen anterior, puede observarse que primero se calibra la sensibilidad para determinar el desplazamiento del mouse, en este caso ha sido configurado para que mantenga un desplazamiento suave, posteriormente son declaradas las variables involucradas en la interacción con el mouse y por ultimo se realiza la configuración del puerto serial para recibir los datos de coordenadas que llegan por el modulo RF.

En la siguiente figura se puede observar las consultas que se realizan para el control sobre el mouse:

```

/*
  Mouse Arduino Leonardo por lectura datos seriales
  Control del mouse por la lectura de códigos recibidos serialmente.
  Hardware:
    * Comunicación Serial "RX"
*/

int Rango = 5; // Velocidad para el desplazamiento
int TiempoEspera = 10; // Tiempo de respuesta del mouse (milisegundos)
int Receptor = 0; // Almacenamiento del RX

int Arriba = LOW;
int Abajo = LOW;
int Izquierda = LOW;
int Derecha = LOW;
int ClickIz = LOW;

void setup() {
  // initialize mouse control:
  Mouse.begin();
  Serial1.begin(9600);
}

void loop() {
  if (Serial1.available() > 0) {
    Receptor = Serial1.read();
    switch (Receptor) {
      case 19:
        // Mueve mouse Arriba
        Abajo = LOW;
        Arriba = HIGH;
        ClickIz = LOW;
        break;
      case 20:
        // Niega movimiento
        Arriba = LOW;
        Abajo = LOW;
        ClickIz = LOW;
        break;
      case 30:
        // Mueve mouse Abajo
        Arriba = LOW;
        Abajo = HIGH;
        ClickIz = LOW;
        break;
    }
  }
}

```

```

Izquierdo = HIGH;
Derecha = LOW;
ClickIz = LOW;
break;
case 200:
  // Niega movimiento
  Derecha = LOW;
  Izquierdo = LOW;
  ClickIz = LOW;
  break;
case 222:
  // Mueve mouse derecha
  Izquierdo = LOW;
  Derecha = HIGH;
  ClickIz = LOW;
  break;
case 255:
  // Click izquierdo
  ClickIz = HIGH;
  break;
}
// Calcula la distancia a mover dependiendo de las variables:
int DistanciaX = (Izquierdo - Derecha)*Rango;
int DistanciaY = (Arriba - Abajo)*Rango;
// Si "X" o "Y" no es cero, mueve:
if ((DistanciaX != 0) || (DistanciaY != 0)) {
  Mouse.move(DistanciaX, DistanciaY, 0);
}
// Si se presiona el boton izquierdo:
if (ClickIz == HIGH) {
  // if the mouse is not pressed, press it:
  if (!Mouse.isPressed(MOUSE_LEFT)) {
    Mouse.press(MOUSE_LEFT);
  }
}
// Si el boton izquierdo no ha sido presionado:
else {
  // Si el mouse ha sido presionado:
  if (Mouse.isPressed(MOUSE_LEFT)) {
    Mouse.release(MOUSE_LEFT);
  }
}
// Un tiempo de espera para que el mouse no se mueva muy rapido:
delay(TiempoEspera);
}
}

```

Compilación terminada Guardado Terminado.

Tamaño binario del Sketch: 5.944 bytes (de un máximo de 28.672 bytes) Tamaño binario del Sketch: 5.944 bytes (de un máximo de 28.672 bytes)

17 Arduino Leonardo on /dev/tty.usbmodem411 94 Arduino Leonardo on /dev/tty.usbmodem411

Figura 4.33: Configuración Desplazamiento del Mouse

En la imagen de la izquierda en el lazo principal se consulta el puerto para determinar si hay algún dato disponible, de ser así, se desarrolla una estructura **case**, en la cual se verifican los posibles códigos que envía el microcontrolador Atmega328 y dependiendo de ello activa o desactiva las variables de desplazamiento, luego se determina el tiempo que estuvo activada dicha variable y su valor se multiplica por la velocidad (variable Rango) y de esta manera se obtiene el desplazamiento en ambos ejes (eje "X", eje "Y"), además se realiza la consulta de pulsación del botón izquierdo.

Capítulo 5

Diseño y fabricación de circuitos impresos

5.1. Diseño en Fritzing

Al momento de diseñar las placas PCB, se emplea el software fritzing, debido a que es un programa multiplataforma orientado principalmente a desarrolladores de arduino, además posee una interfaz amigable con el usuario, haciendo que su empleo sea sencillo.

Este software posee tres ventanas que permiten el desarrollo completo de cualquier tipo de proyectos. Dependiendo de en cual ventana se posicione, se podrá cambiar la vista del esquema del proyecto.

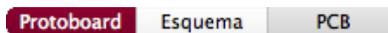


Figura 5.1: Selección de Ventana del Proyecto

En la primera ventana se realiza el diseño real del proyecto, tal cual se lo construye en un Protoboard, al seleccionar la segunda ventana se puede desarrollar el esquemático de conexión del proyecto y finalmente en la tercera ventana se construye el diseño PCB del proyecto.

A lo largo del proyecto hemos presentado esquemas del Protoboard que han sido precisamente construidos en fritzing. En la siguiente gráfica tenemos el esquema final del proyecto presentado en la ventana de Protoboard:

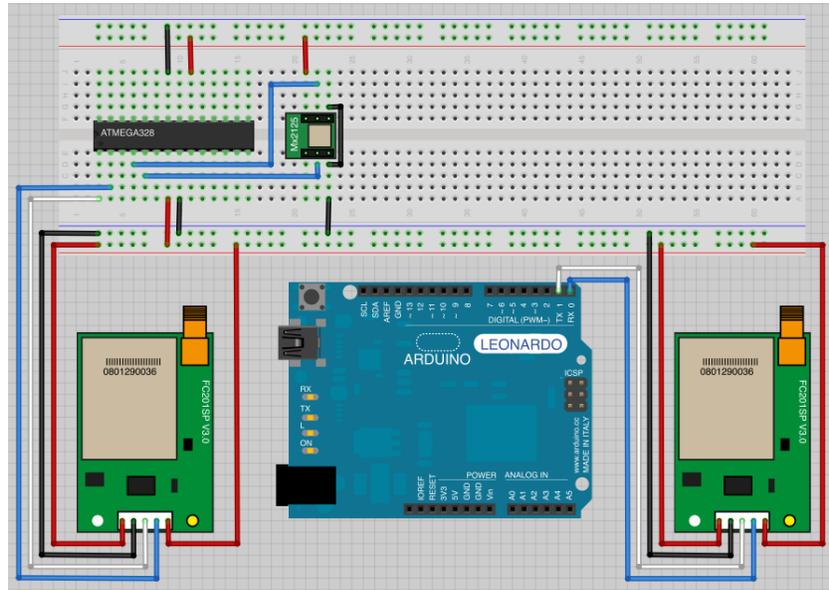


Figura 5.2: Esquema General del Proyecto

Al cambiar a la ventana de PCB se hace el ruteo del proyecto. Teniendo un diseño para el emisor y otro diseño para el receptor.

En la siguiente figura se presenta el esquema del receptor del del prototipo:

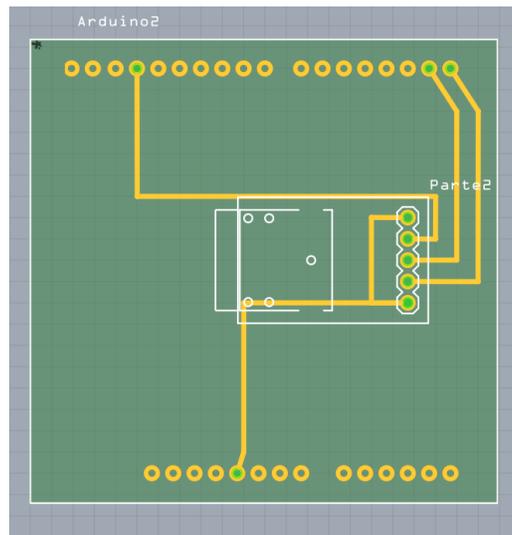


Figura 5.3: Prototipo del receptor

En la siguiente figura se presenta el esquema del transmisor del prototipo:

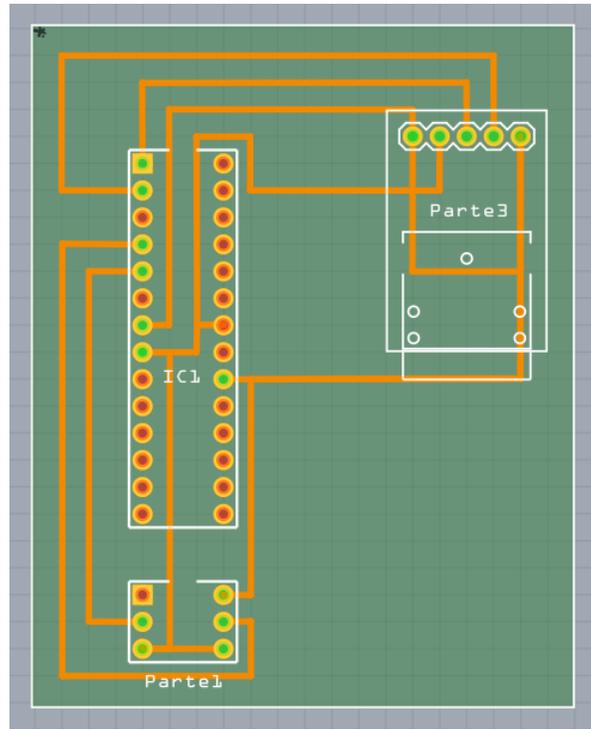


Figura 5.4: Prototipo del receptor

5.2. Elaboración del prototipo PCB

5.2.1. Transferencia a la baquelita

Una vez realizado el diseño del proyecto se utiliza una impresora laser y papel de transferencia termica (papel fotografico tipo Glossy), debido a que la impresión aprovecha un efecto químico al unirse el toner de la impresora con la hoja Glossy.

Realizada la impresión de los circuitos, se corta la baquelita con las dimensiones necesarias para transferir la placa del Transmisor y del Receptor.

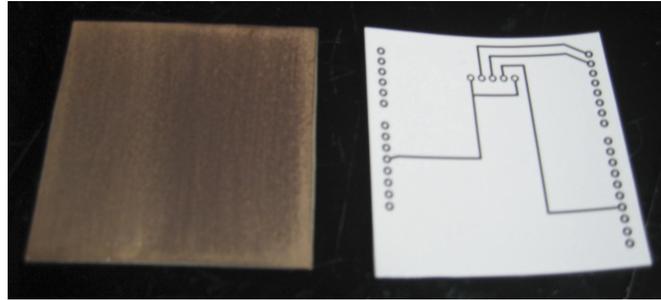


Figura 5.5: Corte de baquelita para el Receptor

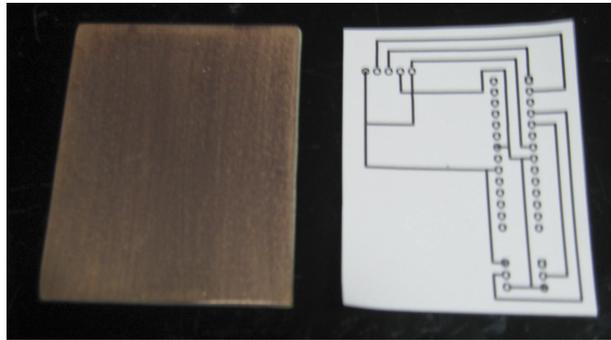


Figura 5.6: Corte de la baquelita para el Transmisor

Cuando se tienen ambos pedazos de baquelita cortados junto con las impresiones, es necesaria una plancha que ayudara a recrear el ambiente cuando se imprimieron los circuitos en las hojas y de esta manera poder transferir el circuito a las baquelitas



Figura 5.7: Elementos para transferir la impresión (Plancha, Baquelita, Circuito impreso)

Para poder comenzar con la transferencia, se sujeta el papel a la baquelita con un pequeño pedazo de cinta masking, la plancha se regula al máximo nivel de temperatura y se presiona ligeramente la plancha durante un minuto para calentar homogéneamente la baquelita, luego durante otro minuto se presiona con la plancha la baquelita en todas direcciones para finalmente transferir el circuito impreso.



Figura 5.8: Sujeción del circuito impreso a la baquelita

Posterior al planchado, es necesario retirar el papel, para ello en una toma de agua junto con un cepillo se retira todo el papel



Figura 5.9: Placas obtenidas después de la transferencia Termica

Para retirar de la baquelita el exceso de cobre (aquellas partes que no están protegidas por la impresión), gracias al Cloruro férrico se lo retira mediante un proceso de atacado lento, en un recipiente de plástico primero se vierte el polvo del ácido férrico y posteriormente se vierte un baso de agua hirviendo y finalmente se colocan las tarjetas. Después de un tiempo aproximado a 30 minutos el exceso de cobre sera eliminado de ambas baquelitas.



Figura 5.10: Recipiente de plástico para atacar el exceso de cobre

Finalmente mediante el empleo de un taladro se hacen los huecos necesarios de las tarjetas y se posicionan los elementos para sujetarlos con la ayuda de pasta y estaño.



Figura 5.11: Suelda de los elementos del receptor

Capítulo 6

Diseño y elaboración de los contenedores

6.1. Diseño del contenedor del Transmisor y Receptor

Para poder hacer el contenedor fue necesario tener las placas, para de esta manera utilizar el espacio mínimo. El diseño en su totalidad se lo realizo en el programa inventor.

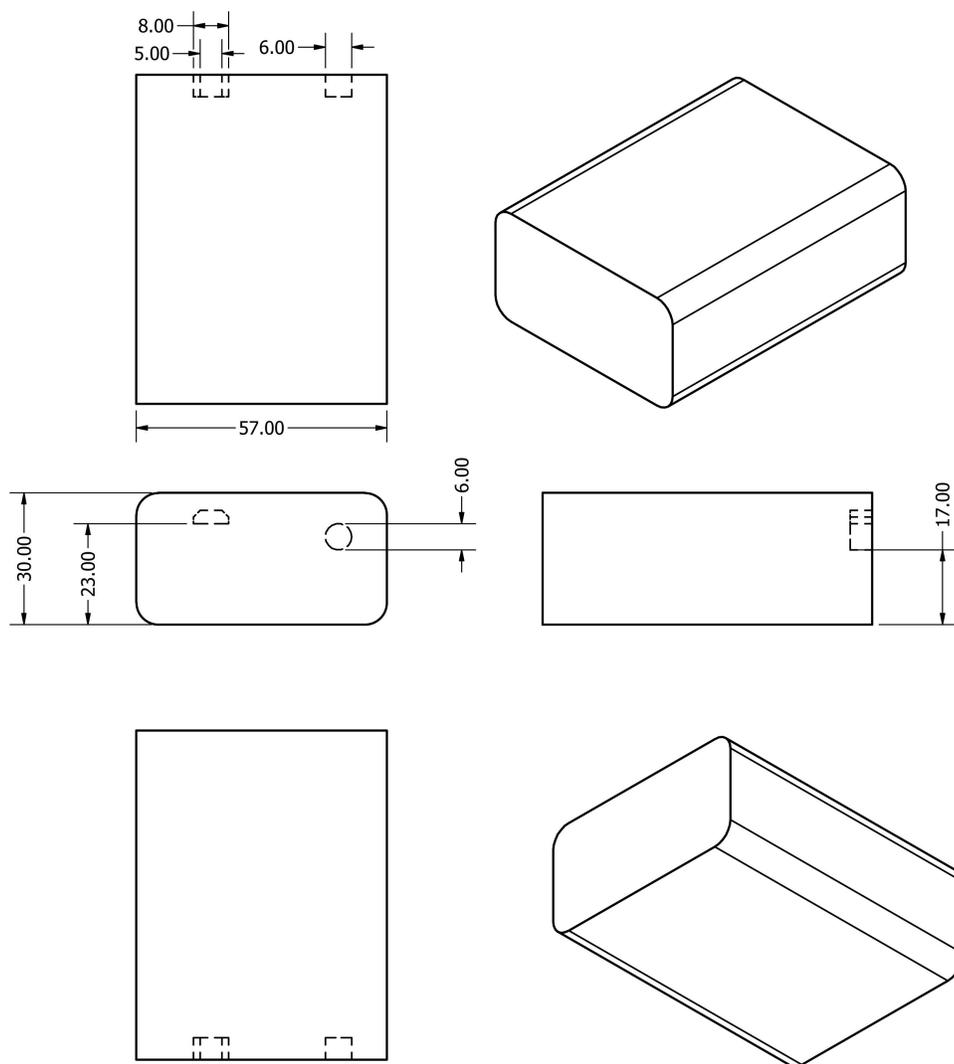


Figura 6.1: Diseño del Contenedor del Receptor

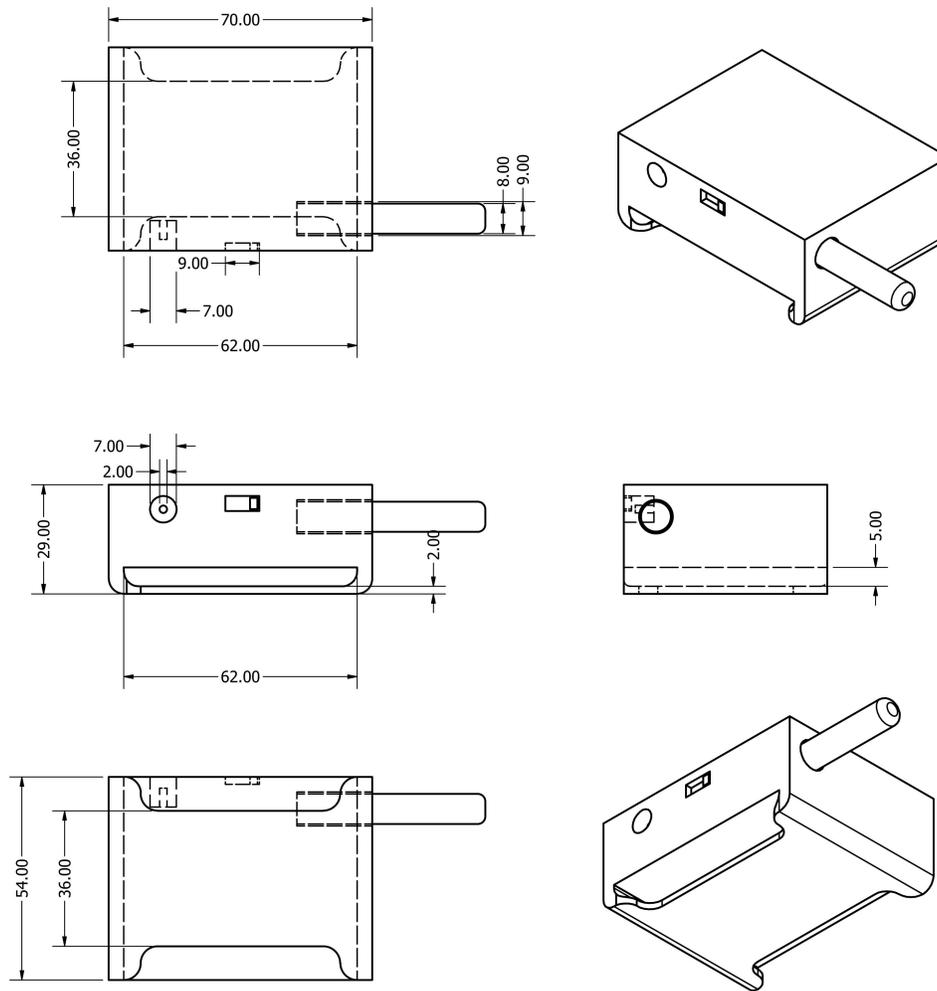


Figura 6.2: Diseño del Contenedor del Transmisor

6.2. Elaboración del contenedor del Transmisor y Receptor

Una vez se tiene realizado el diseño de los contenedores, las piezas se cortan en una lamina de acrílico, mismas que se pueden ensamblar junto con pega adhesiva instantanea universal (Brujita).

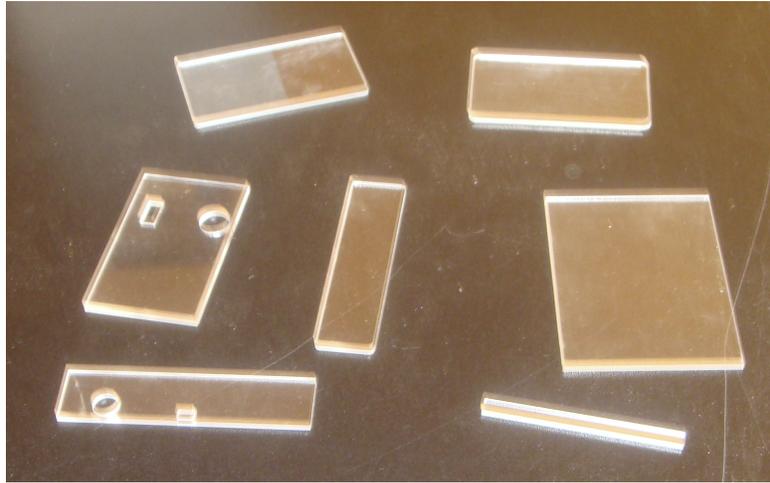


Figura 6.3: Piezas utilizadas para el ensamblaje del transmisor y el receptor

Para realizar el doblado de las piezas, se necesita calentar los filos del acrílico, de esta forma el acrílico es maleable para trabajarlo.

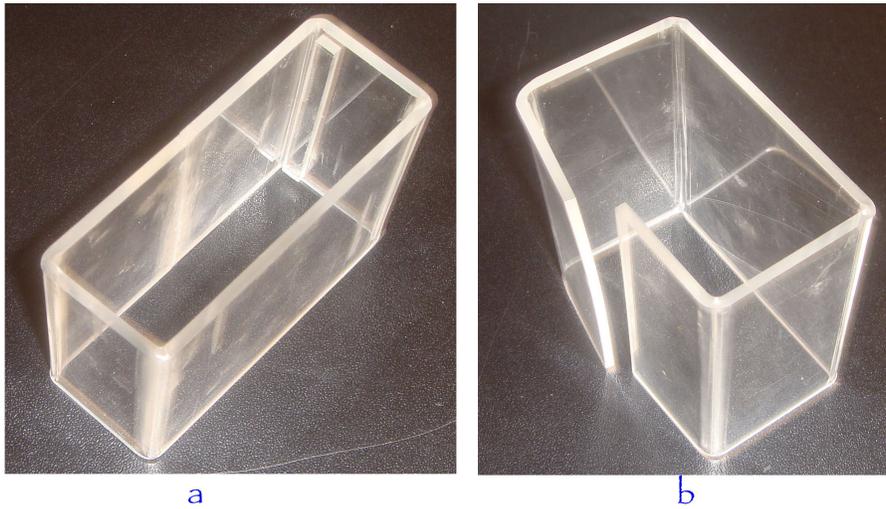


Figura 6.4: Dobleces en los contenedores a): Transmisor, b): Receptor

Cuando ya se tienen ensamblados los contenedores, se le agrega adhesivo de fibra de carbono para resguardar la estructura y que de esta manera tenga una vida útil mas larga



Figura 6.5: Cable del Receptor + Receptor

Finalmente se tiene ensamblado el receptor que se conectara al computador para ejecutar las ordenes del transmisor.



Figura 6.6: Prototipo ensamblado

A continuacion se muestra los elementos que conforman el transmisor.



Figura 6.7: Cable del Transmisor + Transmisor + Pulsante click Izquierdo
Ensamblaje del transmisor que envia los datos al computador.



Figura 6.8: Transmisor

Capítulo 7

Aprendizaje del sistema por un niño:

Durante el proyecto, se tuvo la oportunidad de trabajar con un niño de cuatro años para realizar pruebas utilizando software educativo (Aprende con Pipo), quien después de un tutorial de 10 minutos logro ubicar el cursor exitosamente para aprender para aprender



Figura 7.1: Utilización del Click Izquierdo del mouse

También se tuvo la oportunidad de trabajar con dos niños más, uno de tres años y otro de dos años que les llamo mucho la atención el sistema y el juego de aprendizaje ensí.

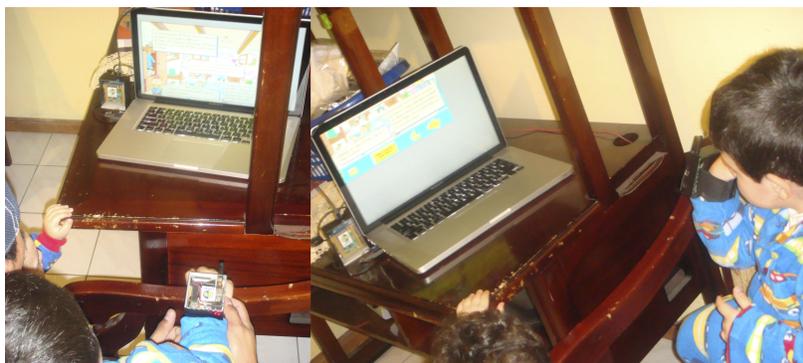


Figura 7.2: Utilización del mouse sin click

Finalmente le llamo la atención del sistema a un niño de dos años que tuvo la paciencia para aprender a jugar con el sistema en la computadora.



Figura 7.3: Utilización del mouse sin click

7.1. Resumen, Conclusiones y Recomendaciones

7.1.1. Resumen

La temática de nuestra tesis se basa en brindar un servicio a la comunidad mediante lo aprendido a lo largo de la carrera, realizando un trabajo académico que cumpla con la visión social, en nuestro caso el IPCA nos abrió las puertas para brindar nuestro granito de arena dentro de dicha institución, en la misma que nos supieron indicar las áreas en las que se podía ayudar a los niños que asisten a este centro, y lo que nos llamo la atención es que no tenían suficiente material didáctico para tener una mejor enseñanza, ya que estos servía para el desarrollo de los niños.

Por ende nuestra idea empezó, con la finalidad de lograr que los niños que no podían tener acceso a una enseñanza computarizada, se puedan vincular

con esta, de otra manera de manera didáctica. Con esto se logro tener una manera de aprender mediante un juego y de igual manera podría servir como un sistema de entrenamiento, para mejorar sus reflejos, etc. Desarrollando de una mejor manera los sentidos de niños con parálisis cerebral. En este proyecto se desarrollo el software y hardware necesario para controlar el mouse de una computadora mediante un acelerómetro, empleando nueva tecnología basada en software libre, para de esta manera brindar un servicio a la comunidad, ayudando a personas con problemas psico-sensometricos, en nuestro caso de forma puntual a un niño de la fundación IPCA El sistema funciona mediante el movimiento de la muñeca (arriba, abajo, derecha, izquierda) por medio de un acelerómetro, con el cual controlamos el mouse de una computadora mediante comunicación RF, la computadora realiza la adquisición de datos por el puerto USB, y con la ayuda del sketch descargado en el arduino se setean las coordenadas rectangulares de ubicación del puntero.

7.1.2. Conclusiones

Al empezar nuestro proyecto partimos con las siguientes preguntas ¿cómo se iba a ayudar a los niños? ¿Qué hacer? ¿Qué beneficios aportaría el proyecto?, es decir vimos como ayudar a niños con deficiencia motriz, ya que será un sistema implementado en la motricidad de la mano. Por ende nuestra idea empezó, con la finalidad de lograr que los niños que no podían tener acceso a una enseñanza computarizada, se puedan vincular con esta, de una manera didáctica.

Con ello logramos la finalidad de que aprenda didacticamente y no quede resagado del grupo ni de la tecnología, mediante un juego y de igual manera podría servir como un sistema de entrenamiento, para mejorar sus reflejos, etc.

Desarrollando de una nueva manera de aprender entreteniendose para los niños.

La principal ventaja tecnológica del proyecto es que es una herramienta multiplataforma, es decir sin necesidad de instalar ningun software adicional puede funcionar bajo los sistemas operativos de Linux, Windows o Macintosh.

7.1.3. Recomendaciones

- Al momento de grabar las placas utilizar el acido en proporciones adecuadas, y debido a que todo es prueba y error tener paciencia para obtener resultados gratificantes.
- Al finalizar de hacer las placas revisar que no exista contacto entre pistas, para evitar que haya cortocircuitos, o pérdida de datos.
- Al hacer los dobleces del acrílico, tener en cuenta una temperatura adecuada (entre 50°C y 60°C), para lograr que esta no se trise o destruya debido a la deformación del mismo.
- En la programación realizar el envío de los datos desde el microcontrolador a la computadora paulatinamente para de esta manera no saturar el puerto

USB.

- Considerar el consumo de la carga para los dispositivos remotos, para de esta manera encontrar la batería adecuada que se ajuste a las necesidades del proyecto.
- Al elegir las antenas, considerar el rango de distancia que funciona el sistema, y la frecuencia a ser utilizada.
- Para la correa que sujeta al transmisor elegir un velcro adecuado, para tener la posibilidad de un ajuste correcto a la muñeca, de una forma cómoda.

Bibliografía

- [1] TERAPIAS ALTERNATIVAS PARA EL DESARROLLO INFANTIL (Recuperado: 22/02/2012) <http://alternativasterapias.blogspot.com>
- [2] INTEGRACION SENSORIAL AYOLA CUESTA PALACIOS Terapeuta Ocupacional Universidad del Rosario (Recuperado: 23/02/1012) <http://www.agapasm.com.br/Artigos/Integracion%20sensorial.pdf>
- [3] ASPACE / PRINCIPALES TIPOS DE PARÁLISIS CEREBRAL (Recuperado 23/02/2012) <http://www.aspace.org/paralisis-cerebral/tipos-de-paralisis-cerebral>
- [4] PARÁLISIS CEREBRAL Y DISCAPACIDAD INTELECTUAL (Recuperado: 25/02/2012) http://www.feaps.org/biblioteca/sindromes_y_apoyos/capitulo14.pdf
- [5] PARÁLISIS CEREBRAL Dra Mercedes Castro Neuropediatra (Recuperado: 25/02/2012) http://ipes.anep.edu.uy/documentos/2011/disc_motriz/materiales/paraliscerebral.pdf
- [6] PARALISIS IS CEREBRAL (Recuperado: 25/02/2012) <http://www.slideshare.net/olmedo50/paralisis-cerebral-5253122>
- [7] PARÁLISIS CEREBRAL / DISCAPANET (Recuperado 25/02/2012) <http://salud.discapnet.es/Castellano/Salud/Discapacidades/Desarrollo%20Motor/Paralisis%20cerebral/Paginas/Descripcion.aspx>
- [8] PARÁLISIS CEREBRAL INFANTIL Pilar Póo Argüelles Servicio de Neurología. Hospital Sant Joan de Dèu, Barcelona (Recuperado 25/02/2012) <http://www.aeped.es/sites/default/files/documentos/36-pci.pdf>
- [9] HID Information (Recuperado: 25/02/1012) <http://www.usb.org/developers/hidpage/>
- [10] PUERTOS E/S: PUERTOS USB (Recuperado: 25/02/2012) http://www.zator.com/Hardware/H2_5_3.htm
- [11] USB (Bus de serie universal) (Recuperado: 26/02/2012) <http://es.kioskea.net/contents/pc/usb.php3>

- [12] ACELEROMETRO DOS EJES (Recuperado: 26/02/2012)
<http://www.msebilbao.com/notas/downloads/Acelerometro%20de%20%20ejes%2028017.pdf>
- [13] Definicion de Interaccion Humano Máquina (Interaction-Interface) (Recuperado 27/02/2012)
[http://interfacemindbraincomputer.wetpaint.com/page/2.A.1.-+Definicion+de+Interaccion+Humano+Máquina+\(Interaction-Interface\)](http://interfacemindbraincomputer.wetpaint.com/page/2.A.1.-+Definicion+de+Interaccion+Humano+Máquina+(Interaction-Interface))
- [14] Arduino Home Page (Recuperado 27/02/2012) <http://arduino.cc/>

Parte I

Anexo 1: Imagenes del proceso de construcción de la maqueta

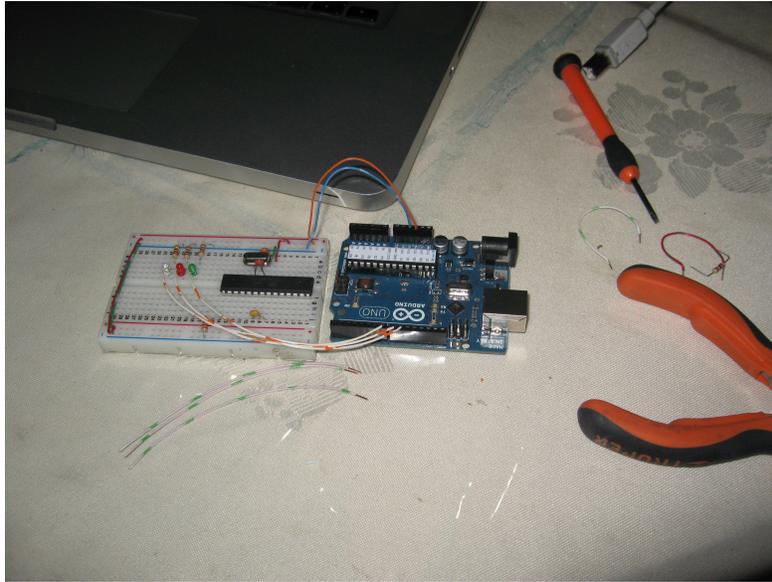


Figura 7.4: Pruebas con Arduino y microcontrolador Atmega328

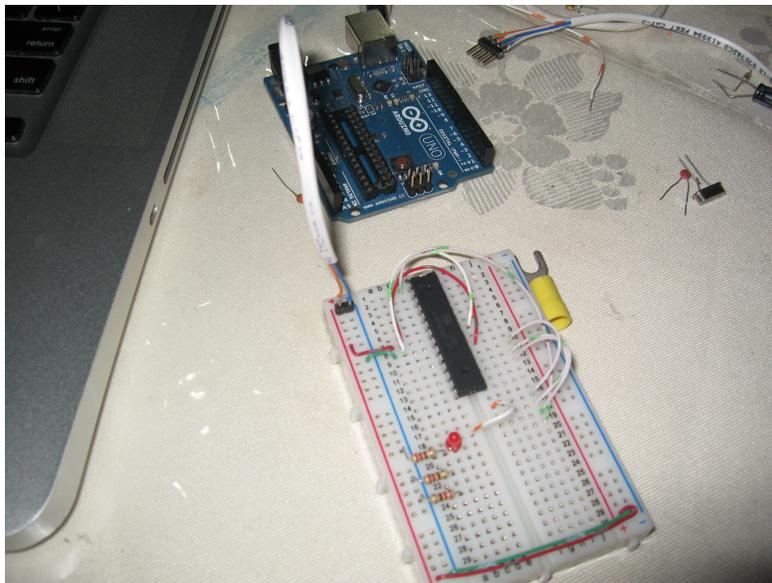


Figura 7.5: Programación Microcontrolador Atmega328

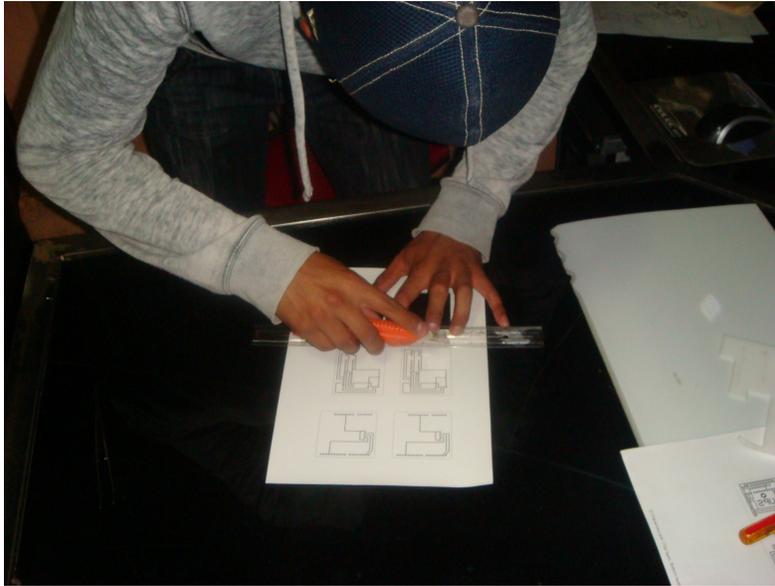


Figura 7.6: Corte de las tarjetas de Transmisor y Receptor del proyecto



Figura 7.7: Eliminación del Cobre Sobrante de las placas



Figura 7.8: Corte de las plantillas de las carcazas



Figura 7.9: Doblez de los contenedores Acrílicos



Figura 7.10: Perforaciones necesarias para los contenedores



Figura 7.11: Pruebas finales

Parte II

Anexo 2: Hoja de datos del Acelerómetro



Web Site: www.parallax.com
 Forums: forums.parallax.com
 Sales: sales@parallax.com
 Technical: support@parallax.com

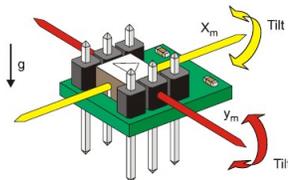
Office: (916) 624-8333
 Fax: (916) 624-8003
 Sales: (888) 512-1024
 Tech Support: (888) 997-8267

Memsic 2125 Dual-Axis Accelerometer (#28017)

The Memsic 2125 is a low-cost thermal accelerometer capable of measuring tilt, collision, static and dynamic acceleration, rotation, and vibration with a range of ± 3 g on two axes. Memsic provides the 2125 IC in a surface-mount format. Parallax mounts the circuit on a tiny PCB providing all I/O connections so it can easily be inserted on a breadboard or through-hole prototype area.

Features

- Measures ± 3 g on each axis
- Simple pulse output of g-force for each axis
- Convenient 6-pin 0.1" spacing DIP module
- Analog output of temperature (TOut pin)
- Fully temperature compensated over 0 to 70 °C operating temperature range



Key Specifications

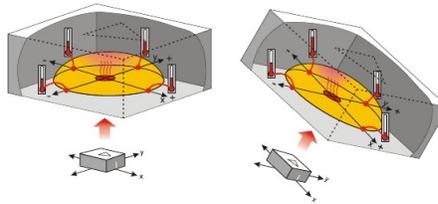
- Power Requirements: 3.3 to 5 VDC; < 5 mA supply current
- Communication: TTL/CMOS compatible 100 Hz PWM output signal with duty cycle proportional to acceleration
- Dimensions: 0.42 x 0.42 x 0.45 in (10.7 x 10.7 x 11.8 mm)
- Operating temperature: 32 to 158 °F (0 to 70 °C)

Application Ideas

- Dual-axis tilt and acceleration sensing for autonomous robot navigation
- R/C tilt controller or autopilot
- Tilt-sensing Human Interface Device
- Motion/lack-of-motion sensor for alarm system
- Single-axis rotational angle and position sensing

Theory of Operation

The MX2125 has a chamber of gas with a heating element in the center and four temperature sensors around its edge. When the accelerometer is level, the hot gas pocket rises to the top-center of the chamber, and all the sensors will measure the same temperature.



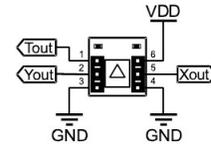
By tilting the accelerometer, the hot gas will collect closer to some of temperature sensors. By comparing the sensor temperatures, both static acceleration (gravity and tilt) and dynamic acceleration (like taking a ride in a car) can be detected. The MX2125 converts the temperature measurements into signals (pulse durations) that are easy for microcontrollers to measure and decipher.

Figura 7.12: Especificaciones Acelerómetro

Pin Definitions

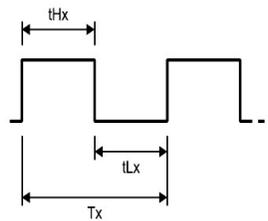
For Memsic MXD2125GL pin ratings, see the manufacturer's datasheet posted on the 28017 product page at www.parallax.com.

Pin	Name	Function
1	Tout	Temperature Out
2	Yout	Y-axis PWM output
3	GND	Ground -> 0 V
4	GND	Ground -> 0 V
5	Xout	X-axis PWM output
6	VDD	Input voltage: +3.3 to +5 VDC



Communication Protocol

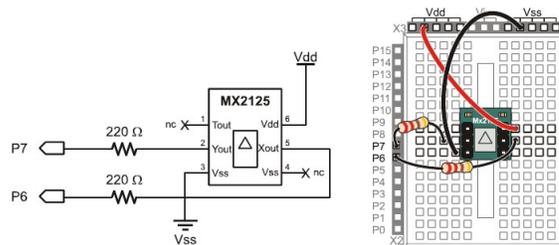
Each axis has a 100 Hz PWM duty cycle output in which acceleration is proportional to the ratio t_{Hx}/T_x . In practice, we have found that T_x is consistent so reliable results can be achieved by measuring only the duration of t_{Hx} . This is easy to accomplish with the BASIC Stamp PULSIN command or with the Propeller chip's counter modules.



With $V_{dd} = 5V$, 50% duty cycle corresponds to 0 g, but this will vary with each individual unit within a range of 48.7% to 51.3%. This zero offset may be different when using $V_{dd} = 3.3 V$. See the manufacturer's datasheet for details.

Example Circuit

The example schematic and wiring diagram below are for the BASIC Stamp and Board of Education.



The program below, SimpleTilt.bs2, simply measures the pulse width, that is, the duration of t_{Hx} , for each axis. The raw values are displayed in the BASIC Stamp Editor's Debug Terminal. If you run the program, then tilt the accelerometer, you should see the values for each axis change.

Figura 7.13: Especificaciones Acelerómetro

```

' Smart Sensors and Applications - SimpleTilt.bs2
' Measure room temperature tilt.

'({$STAMP BS2})
'({$FBASIC 2.5})

x          VAR      Word
y          VAR      Word

DEBUG CLS

DO

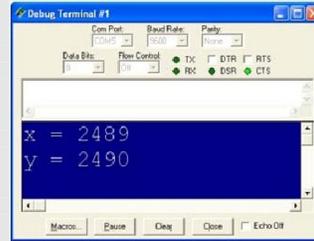
  PULSIN 6, 1, x
  PULSIN 7, 1, y

  DEBUG HOME, DEC4 ? X, DEC4 ? Y

  PAUSE 100

LOOP

```



Programming Resources and Downloads

BASIC Stamp

- Smart Sensors and Applications — The BASIC Stamp example above is taken from the Stamps in Class text *Smart Sensors and Applications*, which features several chapters specific to the Memsic Dual-Axis Accelerometer. Topics include output scaling and offset, measuring vertical rotation, tilt-controlled video gaming basics, data logging g-force during a skateboard trick, and data logging acceleration on an RC car. The book and sample code can be downloaded from the 28029 product page at <http://www.parallax.com>
- Boe-Bot Robot Projects with the Memsic 2125 Accelerometer — The following projects with source code are posted under the Stamps in Class Mini-Projects sticky-thread in the Stamps in Class Forum at <http://forums.parallax.com>:
 - Boe-Bot Robot Navigation with Accelerometer Incline Sensing
 - A Tilt Radio Controller for Your Boe-Bot
- The Memsic 2125 Demo Kit BASIC Stamp Source Code — this source code contains conditional compile directives that allow it to be used with the BS2, BS2e, BS2sx, BS2p, and BS2pe.

Propeller Objects

Several Memsic 2125 Accelerometer code objects and applications for the Propeller chip are available in the Propeller Object Exchange (<http://obex.parallax.com>).

Below is a photograph of the high-speed Memsic MXD2125 Accelerometer Demo in action. This application “provides a high speed assembly driver, and separate-cog and same-cog Spin versions of the MXD2125 Dual Axis Accelerometer. The high speed version displays the data on a television as a 3D wireframe plane with normal vector.



Figura 7.14: Especificaciones Acelerómetro

Parte III

Anexo 3: Hoja de datos del Modulo RF

Micropower Data RF Module

User's Manual



Figura 7.15: Especificaciones Modulos RF



Features of the Micropower Data RF Module

1. ISM frequency band, requiring on applying of frequency point.

Carrier frequency is 433MHz, also capable of providing 868MHz/915MHz carrier frequency.

2. High anti-interference and low BER(Bit error Rate)

Based on the MSK modulation mode, the high-efficiency FEC channel encoding technology is used to enhance data's resistance to both burst interference and random interference

3. Excellent transmission performance

Within the range of visibility and the antenna height >3m, the reliable transmission distance >300m(BER= 10^{-6} /2400Bps).

4.Three interface modes , convenient for setting and use

It can provide three transparent interfaces : TTL,/RS232/ RS485 ,but the user should customise only one of the above before making an order. The interface data rate is 1200/2400/4800/9600/19200Bps optional and its format is 8N1/ 8E1/ 8o1 user-defined.

5. Multi-channel

The standard configuration provides 4 channels. If the user needs, it can be extended to more channels, meeting the multiple communication combination mode of the user.

6.Large data buffer

It can transmit more than 150 bytes long data frames at one time .if the air data rate is greater than the interface rate ,It can transmit more than 200K bytes data frames.

7 . Intelligent data control and transparent data transmission.

The user doesn't need to prepare excessive programs but receive/transmit the data from the interface. The transceiver will automatically complete the other operations such as transmission/receiving conversion in the air, control, etc.

8. Low power consumption

Receiving current is <13mA, transmitting current is <80mA, and sleep current is <0.2mA.(TTL interface used)

Dirección: José Peralta 1077 y Adriano Cobo - Quito -Ecuador - Tlfs: 02-3112514/ 02-2664047
Cel: 099711379/ 096167683 ; Email: info@importronic.net ; Web: www.importronic.net

Figura 7.16: Especificaciones Modulos RF



9. High reliability, small and light convenient embedding.

Single chip RF integrated circuit and single chip MCU are used for lessened peripheral circuits, high reliability, and low failure rate.

Application of Micropower Data RF Module

The Micropower Data RF Module is suitable for:

Short distance wireless data transmission.

Wireless meter reading

Automatic data collecting system

Industrial remote control and remote measurement

Building automation, safety and security, powerhouse equipment wireless monitor, entrance control system

How to use the Micropower Data RF Module

1. Interface sketch map

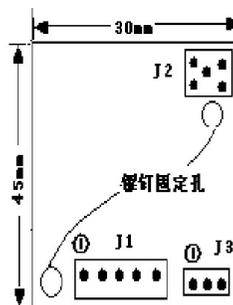


Fig. 1 Interface sketch map

2. Dimension and weight

Size: 45mmX30mmX10mm ; Weight :10g (antenna excluded)

Dirección: José Peralta 1077 y Adriano Cobo - Quito - Ecuador - Tlfs: 02-3112514 / 02-2664047
Cel: 099711379 / 096167683 ; Email: info@importronic.net ; Web: www.importronic.net

Figura 7.17: Especificaciones Modulos RF

3. Definition of Interface.

J1: User interface

Pin 1: $5 \pm 0.5V$

Pin 2 GND

Pin 3:RXD: serial data receiving end (RS485 A)

Pin 4:TXD: serial data transmitting end (RS485 B)

Pin 5 IDLE: power-saving mode selection (IDLE mode), Low level takes effect, hung or high level (3.5V) is for active mode.

J2: Antenna interface

J3: User interface

Pin1: Green light

Pin 2:Power supply

Pin 3: Red light

4. Connection with the terminal

It can provide TTL/RS232/RS485 interface. Please select one interface mode before making an order.

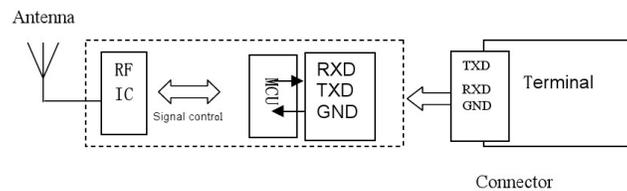


Fig 2. Application principle map

5. Setting of channel , interface and data format

1) Setting and reading of parameters

User can set or read the serial port data rate , air data rate , channel number and address code .

Parte IV

Anexo 4: Hoja de datos del Microcontrolador Atmega328

ATmega48PA/88PA/168PA/328P

1. Pin Configurations

Figure 1-1. Pinout ATmega48PA/88PA/168PA/328P

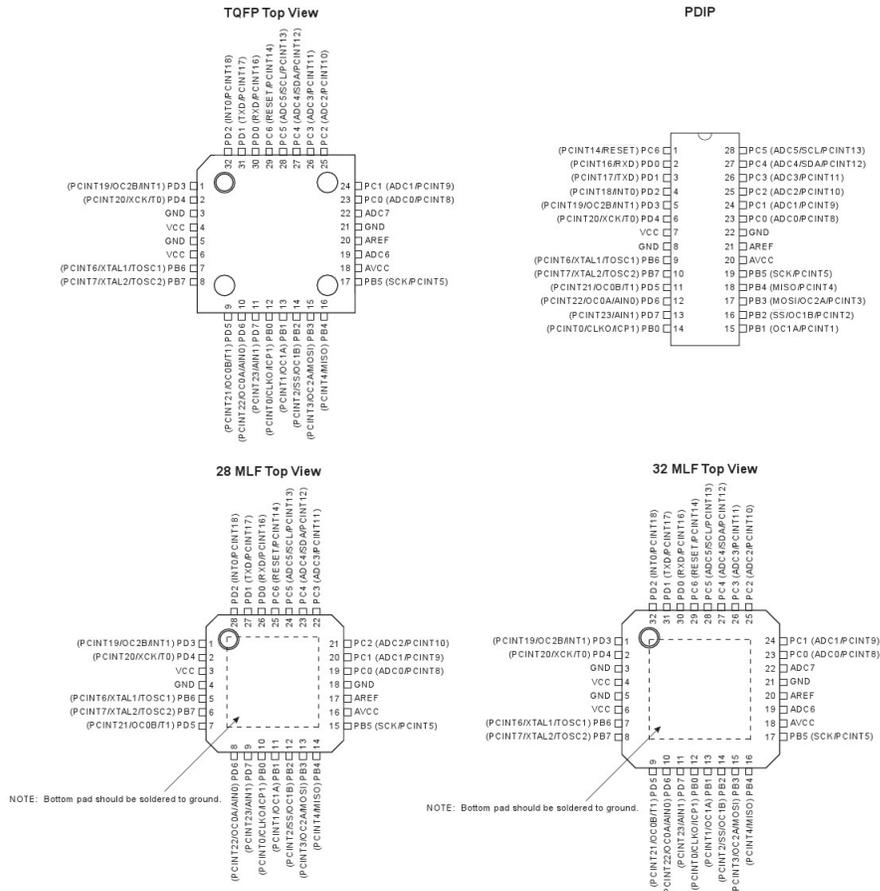


Figura 7.19: Especificaciones Microcontrolador Atmega328

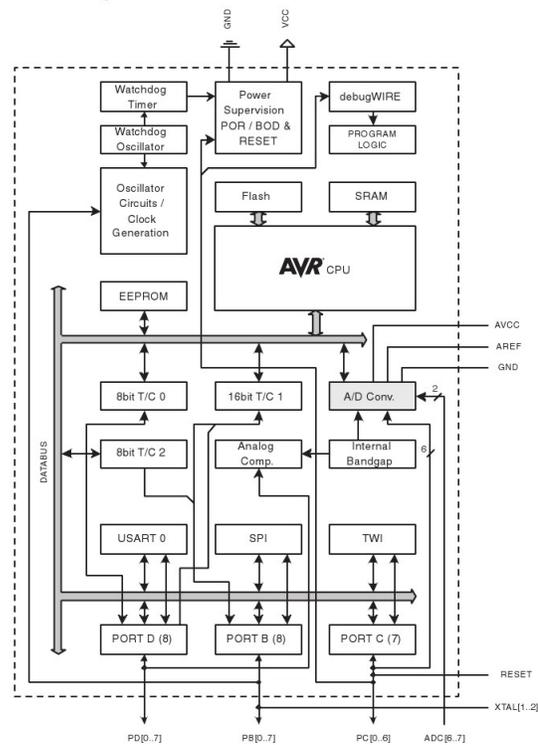
ATmega48PA/88PA/168PA/328P

2. Overview

The ATmega48PA/88PA/168PA/328P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega48PA/88PA/168PA/328P achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

Figure 2-1. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting

Figura 7.20: Especificaciones Microcontrolador Atmega328

ATmega48PA/88PA/168PA/328P

13.4 Register Description

13.4.1 MCUCR – MCU Control Register

Bit	7	6	5	4	3	2	1	0	
0x35 (0x55)	–	BODS	BODSE	PUD	–	–	IVSEL	IVCE	MCUCR
Read/Write	R	R	R	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 4 – PUD: Pull-up Disable**

When this bit is written to one, the pull-ups in the I/O ports are disabled even if the DDxn and PORTxn Registers are configured to enable the pull-ups ((DDxn, PORTxn) = 0b01). See "Configuring the Pin" on page 76 for more details about this feature.

13.4.2 PORTB – The Port B Data Register

Bit	7	6	5	4	3	2	1	0	
0x05 (0x25)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

13.4.3 DDRB – The Port B Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x04 (0x24)	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

13.4.4 PINB – The Port B Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x03 (0x23)	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A								

13.4.5 PORTC – The Port C Data Register

Bit	7	6	5	4	3	2	1	0	
0x08 (0x28)	–	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	PORTC
Read/Write	R	R/W							
Initial Value	0	0	0	0	0	0	0	0	

13.4.6 DDRC – The Port C Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x07 (0x27)	–	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
Read/Write	R	R/W							
Initial Value	0	0	0	0	0	0	0	0	

13.4.7 PINC – The Port C Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x06 (0x26)	–	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	PINC
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	N/A							

Figura 7.21: Especificaciones Microcontrolador Atmega328

ATmega48PA/88PA/168PA/328P

13.4.8 PORTD – The Port D Data Register

Bit	7	6	5	4	3	2	1	0	
0x0B (0x2B)	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

13.4.9 DDRD – The Port D Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x0A (0x2A)	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

13.4.10 PIND – The Port D Input Pins Address

Bit	7	6	5	4	3	2	1	0	
0x09 (0x29)	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A								

Figura 7.22: Especificaciones Microcontrolador Atmega328