



**UNIVERSIDAD POLITÉCNICA SALESIANA SEDE
GUAYAQUIL**

CARRERA: INGENIERÍA ELECTRÓNICA

Trabajo de Titulación previo a la obtención del título de:

INGENIERO ELECTRÓNICO

TEMA:

**DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO DIDÁCTICO DE
DOMÓTICA POR MEDIO DE LÁMINAS CON LA TARJETA RASPBERRY
PI B+ Y EL PROGRAMA QT EN LA UNIVERSIDAD POLITÉCNICA
SALESIANA SEDE GUAYAQUIL**

AUTORES:

Bryan Raúl Gordon Sánchez

Edgar Gustavo Tacuri Romero

TUTOR:

Msc. Ing. Luis Silvio Córdova Rivadeneira

Guayaquil, Octubre 2016

CERTIFICADO DE RESPONSABILIDAD

Nosotros, Gordon Sánchez Bryan Raúl con cédula de identidad N° 0924780588 y Tacuri Romero Edgar Gustavo con cédula de identidad N° 0931043186, declaramos que este trabajo de titulación “DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO DIDÁCTICO DE DOMÓTICA POR MEDIO DE LÁMINAS CON LA TARJETA RASPBERRY PI B+ Y EL PROGRAMA QT EN LA UNIVERSIDAD POLITÉCNICA SALESIANA SEDE GUAYAQUIL” ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Guayaquil, Octubre del 2016

Bryan Raúl Gordon S.

C.I.: 0924780588

Edgar Gustavo Tacuri R.

C.I.: 0931043186

CERTIFICADO DE CESIÓN DE DERECHOS

Nosotros, Gordon Sánchez Bryan Raúl con cédula de identidad N° 0924780588 y Tacuri Romero Edgar Gustavo con cédula de identidad N° 0931043186, autorizamos a la Universidad Politécnica Salesiana publicar en la biblioteca virtual de la institución el presente trabajo de titulación “DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO DIDÁCTICO DE DOMÓTICA POR MEDIO DE LÁMINAS CON LA TARJETA RASPBERRY PI B+ Y EL PROGRAMA QT EN LA UNIVERSIDAD POLITÉCNICA SALESIANA SEDE GUAYAQUIL” cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Guayaquil, Octubre del 2016

Bryan Raúl Gordon S.

C.I.: 0924780588

Edgar Gustavo Tacuri R.

C.I.: 0931043186

**CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN SUSCRITA
POR EL TUTOR**

Por medio de la presente quiero darle a conocer que el Proyecto de Titulación : “DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO DIDÁCTICO DE DOMÓTICA POR MEDIO DE LÁMINAS CON LA TARJETA RASPBERRY PI B+ Y EL PROGRAMA QT EN LA UNIVERSIDAD POLITÉCNICA SALESIANA SEDE GUAYAQUIL” realizado por los Sres. Gordon Sánchez Bryan Raúl y Tacuri Romero Edgar Gustavo, estudiantes de la carrera de Ingeniería Electrónica, ha sido concluida, y revisada en su totalidad, tanto el trabajo ESCRITO como el trabajo PRÁCTICO.

Particular que informo para los fines pertinentes.

Atentamente,

Ing. Luis Córdova Rivadeneira MSc.

DIRECTOR DE PROYECTO DE TITULACIÓN

DEDICATORIA

Dedico este proyecto técnico a todas las personas que me formaron hasta ser quien soy hoy.

A mis padres quienes me han apoyado todo el tiempo con mis metas, proyectos, ideas y aspiraciones a lo largo de toda mi vida, por ellos que me apoyaron aun cuando falle que siempre me aconsejaron y gracias a su formación y ayuda he terminado este proyecto.

A mis compañeros y amigos que gracias a ellos a lo largo de mi vida aprendí varias cosas y me ayudaron de a poco para formar el carácter que tengo hoy en día.

Para ellos es esta dedicatoria, pues es a ellos a quienes se las debo por su apoyo incondicional.

Bryan Gordon

DEDICATORIA

Este trabajo práctico va dedicado para las personas que han formado parte de mi vida, que han estado en los buenos y malos momentos.

Aquellos que me han dado su mano en los momentos en los que nadie creía y en este instante puedo demostrar que siempre fui capaz de cumplir mis metas, a ellos dedico este paso importante en mi vida profesional y personal.

A mis padres que gracias a su apoyo y resistencia puedo llegar a cumplir esta meta que no es solo mía sino que ellos forman un pilar fundamental en esto, a mi hermana que con sus palabras de aliento no me dejaba caer jamás y siempre estuvo ahí.

Gustavo Tacuri

AGRADECIMIENTO

El presente trabajo primero me gustaría agradecerle a mis padres, las personas que estuvieron conmigo en todo momento, con los que crecí como persona, con los que pase momentos buenos y malos, con los que juntos pasamos momentos difíciles en familia y personal, con los que en esos momentos supieron ayudarme y formar mi carácter y convertirme la persona que soy hoy en día, ellos que han esperado por este momento en el que me convierta en un ingeniero, ellos que son las personas que siempre confían y confiaran en mí, ellos son los primeros que quiero agradecer.

A mi tutor de tesis el Ing. Luis Córdova que nos dio la guía y ayuda en varios momentos de este proyecto y cuando fue profesor nuestro en varias materias que son la base esencial de nuestro proyecto.

Quiero agradecerle a mi amigos, compañeros de estudio de la escuela, colegio, universidad, seminarios, cursos, a todos ellos, directamente o indirectamente me ayudaron en mi formación, a ellos que siguen en los recuerdos de las etapas de mi vida les agradezco por ser parte de ellos.

Al final mi agradecimiento es a la vida, cada momento, experiencia, persona, recuerdo, risa, viaje, absolutamente todo hizo que este aquí hoy en día agradeciendo a la vida, por llegar al momento cúspide de mi carrera académica, gracias a todos.

Bryan Gordon

AGRADECIMIENTO

Un agradecimiento total a Dios, por darme la paciencia y fortaleza para avanzar en el camino largo de la vida no solo con este proyecto técnico sino en aspectos personales en los cuales he tenido que ser muy paciente y gracias a ÉL y a sus bendiciones he salido adelante.

A mis padres Esperanza y Vicente que más allá de un apoyo económico me han ayudado moralmente para siempre crecer como persona y como profesional.

A la Universidad Politécnica Salesiana por permitir capacitarnos en sus instalaciones como profesionales, llegando a tener el conocimiento para implementarlo en el ambiente laboral.

Gustavo Tacuri

RESUMEN

DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO DIDÁCTICO DE DOMÓTICA POR MEDIO DE LÁMINAS CON LA TARJETA RASPBERRY PI B+ Y EL PROGRAMA QT EN LA UNIVERSIDAD POLITÉCNICA SALESIANA SEDE GUAYAQUIL

El proyecto de investigación está orientado a un módulo de domótica específicamente con un sistema embebido, el cual tiene alcance a sistemas de seguridad y simulación de una práctica industrial todo esto mediante el uso de la tarjeta Raspberry Pi B+ tipo microprocesador, permitiendo el control de luces piloto, cerradura eléctrica, sensor PIR, sensor magnético, cámaras de seguridad, motor, válvula selenoide, sensor de temperatura, alarma auditiva y teclado numérico.

Debido al avance de la tecnología es necesario diseñar e implementar un módulo didáctico que permita el conocimiento de nuevas tendencias en el caso específico la domótica y la utilización de un nuevo software como Qt Creator. Para ello se planteó la propuesta de un módulo didáctico con láminas móviles el cual cuenta con elementos importantes para un innovador método de estudio.

El módulo está compuesto de varias etapas de control una de ellas es el manejo de luces piloto que serán controladas desde una Laptop a través del software QT Creator donde se declaran los pines GPIO de la Raspberry Pi B+ y la interfaz gráfica de simulación, adicional a las luces piloto se controlarán elementos tales como cerradura eléctrica, ventilador y válvula selenoide, tomando en cuenta que estos elementos trabajan con 120V. Para la comunicación de estos equipos y la Raspberry Pi B+ son necesarias las tarjetas de interfaz de potencia las cuales tendrán una salida de 3.3V lo cual evitará problemas de conexión y de comunicación en las prácticas a realizar.

El circuito cerrado de cámaras de seguridad tiene la misión de monitorear y captar las imágenes una vez que el operador (estudiantes) emita la orden de transmitir desde la interfaz gráfica del software QT Creator.

Las láminas que contienen el teclado matricial y sensor de temperatura tienen una alimentación de 5V de entrada y 3.3V de salida para poder mantener los rangos de operación de la Raspberry Pi B+, los sensores magnéticos, el sensor PIR y la alarma audible trabajan directamente con el valor de los pines GPIO de la Raspberry Pi B+, es decir 3.3V

ABSTRACT

DESIGN AND IMPLEMENTATION OF A HOME AUTOMATION TRAINING MODULE THROUGH BY PLATES WITH CARD RASPBERRY PI AND QT PROGRAM IN SALESIAN POLYTECHNIC UNIVERSITY HEADQUARTERS GUAYAQUIL

The research project is aimed at a module home automation specifically with an embedded system, which has scope to security systems and simulation of an industrial all this practice by using the Raspberry Pi B + type microprocessor card, allowing control of lights pilot, electric lock, PIR sensor, magnetic sensor, security cameras, motor, solenoid valve, temperature, and keypad audible alarm sensor.

Due to the advancement of technology it is necessary to design and implement a training module that enables knowledge of new trends in the specific case of home automation and the use of new software as Qt Creator. To this end the proposal of a training module with moving blades which has important elements for an innovative method of study was proposed.

The module is comprised of various control stages one of which is the management of pilot lights will be controlled from a laptop through the QT Creator software where the GPIO pins on the Raspberry Pi B + and graphical simulation interface are declared, additional pilot lights elements such as electric lock, fan and solenoid valve will be controlled, taking into account that these elements work with 120V. For the communication of these teams and the Raspberry Pi B + interface cards power which will have an output of 3.3V which avoid problems of connection and communication practices are necessary to perform.

Closed circuit security cameras has the mission to monitor and capture images once the operator (students) to issue the order to transmit the GUI QT Creator software.

The sheets containing the matrix keyboard and temperature sensor have a 5V input and 3.3V output to maintain operating ranges of the Raspberry Pi B +, magnetic sensors, the PIR sensor and audible alarm work directly with the value of the GPIO pins Raspberry Pi + B, ie 3.3V

Índice General

INTRODUCCIÓN.....	1
1. Descripción del problema.....	2
1.1 Importancia	3
1.2 Alcance	3
2. Objetivos	5
2.1 Objetivo General	5
2.2 Objetivos Específicos.....	5
3. Fundamento Teórico.....	6
3.1 Introducción a la Domótica.....	6
3.1.1 Ventajas de la Domótica	6
3.1.2 Desventajas de la Domótica.....	7
3.2 Descripción De Los Sistemas.	7
3.2.1 Puertas y ventanas.....	7
3.2.2 Seguridad.....	7
3.3 Raspberry Pi	8
3.4 Raspberry Pi B+	8
3.5 Sistema Operativo.....	12
3.6 Consumo de una Raspberry Pi b+	12
3.7 Qt Creator.....	13
3.8 Características	14
3.9 Características útiles.....	14
3.10 Integración de Qt Designer	14

3.10.1	Teclado de navegación.....	14
3.10.2	Librerías.....	15
3.10.3	Instalación utilizando GIT.	15
3.11	Motion.....	15
3.12	Escritorio Remoto.....	17
3.13	Luz piloto	18
3.14	Chapa eléctrica.....	18
3.15	Bocina	19
3.16	Sensor infrarrojo pasivo.....	19
3.17	Sensor magnético.....	20
3.18	Sensor de temperatura.....	20
3.18.1	Sensor de temperatura lm35	21
3.19	Teclado matricial	21
3.20	Display alfanumérico	22
3.21	Válvula solenoide	22
3.22	Interfaces.....	23
3.22.1	Board APM 4550 PIC 18F4550	23
3.23	Desarrollo de láminas móviles.....	24
3.23.1	Alarma.....	25
3.23.2	Cámaras 1 y 2.....	25
3.23.3	Cerradura Eléctrica.....	26
3.23.4	Electroválvula.....	26
3.23.5	Banco de Luces.....	27

3.23.6 Ventilador.....	27
3.23.7 Sensor P.I.R.....	28
3.23.8 Sensor Magnético 1 y 2.....	28
3.23.9 Sensor de temperatura.....	29
3.23.10 Ingreso de Clave.....	29
3.24 Presupuesto.....	30
4 Marco Metodológico.....	33
5 Innovación.....	34
6 Resultados	35
6.1 Funcionabilidad.....	35
6.2 Tarjeta Interfaz de Potencia de Dos Canales	37
6.3 Esquema en Proteus de la Tarjeta Interfaz de Potencia	38
6.4 Tarjeta de divisor de voltaje de 5V a 3.3V.....	39
6.5 Tarjeta para conexión de bus de datos	40
6.6 Módulo didáctico de domótica con la tarjeta Raspberry Pi B+.....	41
6.7 Prácticas.....	43
Práctica #1.- Simulación e implementación de un sistema de control de luces	43
Práctica #2.- Simulación e implementación de un sistema cerrado con cámaras de seguridad.....	49
Práctica #3.- Simulación e implementación de un circuito con cerradura, sensores magnéticos y sensores de proximidad.....	54
Práctica #4.- Simulación e implementación de un sistema de seguridad residencial	60

Práctica #5.- Simulación e implementación de un sistema industrial con válvulas, motor DC.....	66
7 Conclusiones.....	72
8 Recomendaciones.....	73
9. Referencias Bibliográficas.....	74
10. Anexos.....	76
10.1 Equipos para programación.....	76

Índice de Figuras

Figura 1. Ventajas de la domótica (Domoactualidad, 2014)	7
Figura 2. Puertos GPIO de Raspberry PI b. (Muthu, 2014).....	8
Figura 3. Puertos GPIO de Raspberry PI b+. (Raspberrypi, 2015).....	10
Figura 4. Elementos de la Tarjeta Raspberry PI B+. (Raspberrypi, 2015)	12
Figura 5. Ventana principal de Qt Creator	13
Figura 6. Archivo Motion	16
Figura 7. Escritorio Remoto	16
Figura 8. Entorno de Ingreso XRDP	18
Figura 9. Luces Piloto 110V (Harok, 2015)	18
Figura 10. Chapa eléctrica (Yalelatinoamerica, 2015).....	19
Figura 11. Alarma auditiva (Tvc, 2000)	19
Figura 12. Sensor P.I.R. (Electronics, 2016).....	20
Figura 13. Sensor Magnético (Directlink, 2015).....	20
Figura 14. Sensor de Temperatura (Mileti, 2012).....	21
Figura 15. DataSheet de Teclado Matricial (Microcontroladores, 2009).....	21
Figura 16. Display de Dimensiones 2x16 (Marcelo, 2006)	22
Figura 17. Válvula Selenoide	22
Figura 18. Tarjeta Módulo Interfaz de Potencia de 4 canales.....	23
Figura 19. DataSheet PIC18F4550 (Nextiafenix, 3012)	23
Figura 20. PIC 18F4550	24
Figura 21. Lámina de Alarma.....	25
Figura 22. Láminas Cámaras 1 y 2.....	25
Figura 23. Lámina Cerradura Eléctrica.....	26
Figura 24. Lámina Electroválvula.....	26
Figura 25. Lámina Banco de Luces.....	27
Figura 26. Lámina Ventilador.....	27
Figura 27. Lámina Sensor P.I.R.....	28
Figura 28. Lámina Sensor Magnético 1 y 2.....	28
Figura 29. Lámina Sensor de Temperatura.....	29
Figura 30. Lámina Ingreso de Clave.....	29
Figura 31. Ingreso Mediante Clave	31
Figura 32. Verificación de conexión a Internet	32
Figura 33. Interfaz Raspberry PI B+	35
Figura 34. Ingreso al software Qt Creator	36

Figura 35. Buscar el Proyecto desde de Qt Creator	37
Figura 36. Seleccionar el proyecto en las Carpetas creadas por defecto	37
Figura 37. Tarjeta Interfaz de Potencia de Dos Canales	38
Figura 38. Esquema en Proteus de la Tarjeta Interfaz de Potencia	38
Figura 39. Esquema en Ares de la Tarjeta de Interfaz de Potencia	39
Figura 40. Esquema en Proteus del divisor de voltaje	39
Figura 41. Esquema en Ares de la Tarjeta del divisor de voltaje	40
Figura 42. Esquema en Proteus de la conexión Raspberry Pi B+ / Borneras.....	40
Figura 43. Esquema en Ares de la conexión Raspberry Pi B+ / Borneras	41
Figura 44. Módulo didáctico de domótica con la tarjeta Raspberry Pi B+ y el programa Qt Creator.....	42
Figura 45. Módulo de domótica	42
Figura 46. Interfaz de un sistema de control de luces QtCreator	44
Figura 47. Esquema de Control de luces	45
Figura 48. Funcionamiento de la Practica 1	45
Figura 49. Interfaz del programa de la Practica 2	51
Figura 50. Esquema de un sistema cerrado con cámaras de seguridad	51
Figura 51. Prueba del manejo de las cámaras en el laboratorio.....	51
Figura 52. Interfaz del programa de la Practica 3	55
Figura 53. Esquema de un circuito con cerradura, sensores magnéticos y sensores de proximidad.....	56
Figura 54. Interfaz del programa en funcionamiento con los sensores activados.....	56
Figura 55. Interfaz del programa de la Practica 4	62
Figura 56. Esquema de un Sistema de seguridad residencial	62
Figura 57. Interfaz de la Practica 4 funcionando con los sensores activados	63
Figura 58. Interfaz del programa de la Practica 4	68
Figura 59. Esquema de un sistema industrial con válvulas, motor DC.....	68
Figura 60. Lámina con Motor a 110v.....	76
Figura 61. Lámina con Cerradura Eléctrica a 110V	76
Figura 62. Lámina de sensor magnético 1	77
Figura 63. Lámina de la Electroválvula	77
Figura 64. Lámina de Banco de Luces	78
Figura 65. Lámina de Sensor PIR	78
Figura 66. Lámina de Ingreso de Clave	79
Figura 67. Lámina con cámara de seguridad 2	79

Índice de Tablas

Tabla 1. Comparación de tarjetas Raspberry	9
Tabla 2. Presupuesto de Equipos	30

INTRODUCCIÓN

En la actualidad el uso de la domótica es una tendencia en el mercado ya que existe la necesidad de protección en el hogar y a nivel empresarial de contar con una conexión segura y eficaz cuando no se esté presente en el sitio.

Implementar y dar a conocer un mini-procesador como es la Rapsberry Pi B+ es el inicio para proyectos a futuro de domótica teniendo en cuenta que es una rama de la Ingeniería moderna, uno de los objetivos es levantar el interés y la curiosidad en los estudiantes por aprender algo nuevo para ellos, que sean capaces de resolver un problema y demostrar que no existe dificultad con los conocimientos adecuados.

La importancia de este proyecto es que se pondría en práctica los conocimientos aprendidos en la Universidad de manera más real, utilizando nueva tecnología de comunicación y aplicándola a prácticas ya antes realizadas pero desde un punto diferentes ya que entran en el campo de materias como sensores, microprocesadores, y demostraciones en instalaciones civiles residenciales, teniendo como ayuda el software de programación QT Creator y las Raspberry Pi B+.

1. Descripción del problema

La Universidad Politécnica Salesiana Sede Guayaquil, y específicamente la Carrera Ingeniería Electrónica con el afán de adquirir nuevas herramientas, se plantea la propuesta de diseñar e implementar un módulo de domótica para el uso académico.

Se llevó a cabo el análisis de crear un módulo didáctico, en vista de que existe un incremento de oportunidades en el campo de la domótica y se puede observar que actualmente se necesita de un módulo donde los estudiantes accedan a un entrenamiento didáctico sobre programación, diseño y análisis del funcionamiento de un sistema embebido.

Antecedentes

Se determinó que la carrera de Ingeniería Electrónica actualmente no cuenta con un módulo didáctico con enfoque a la tecnología de los sistemas domóticos, cuenta actualmente con maletines en los cuales se encuentran elementos orientados a este tipo de sistemas, con proyectos basados en Domótica implementados en la industria local que no permiten el acceso a su estudio.

Basado en el Proyecto de Tesis “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE SEGURIDAD DE CONTROL LOCAL Y REMOTO CON DISPOSITIVOS DE VIGILANCIA, DESARROLLADO EN EL SOFTWARE PYTHON, CENTRALIZÁNDOSE EN UNA TARJETA RASPBERRY PI”, implementado fuera de la Universidad Politécnica Salesiana se llegó a la conclusión de que las propuestas sobre Domótica han sido eficientes y demostradas correctamente pero no llega a cumplir un alcance dentro de la institución, por esta razón se escucha del tema y de proyectos probables a desarrollar pero no existe una herramienta o módulo en el cual todos estos avances tecnológicos puedan ser demostrados y puestos en práctica por los estudiantes ya sea en un seminario académico o dentro de una materia de estudio.

1.1 Importancia

Con los nuevos cambios en avances tecnológicos y la rama de los sistemas embebidos la Universidad Politécnica Salesiana Sede Guayaquil - Carrera Ingeniería Electrónica, está involucrándose en mayor grado en este tipo de sistemas.

Esto puede significar para la Universidad Politécnica Salesiana Sede Guayaquil una estrategia en la enseñanza de tecnologías brindando a los estudiantes posibilidades donde enfocar su ingeniería una vez culminados sus estudios.

Por otra parte los sistemas embebidos representan un considerable avance en ordenadores de bajo costo y múltiples aplicaciones, en este caso específico la Rapsberry Pi B+ es una herramienta de gran alcance.

La Rapsberry Pi B+ como una herramienta de estudio dará inicio a un desarrollo de aplicaciones dentro del campo de la robótica y microcontroladores II ya que contiene aplicaciones y plataformas adecuadas para un uso académico.

1.2 Alcance

Los principales o directos beneficiarios serán los estudiantes de la carrera de Ingeniería Electrónica ya que mediante este módulo didáctico podrán analizar futuras herramientas o posibilidades laborales en vista de que es un sistema moderno y de gran impacto por la innovación de tecnología, esto implica que es agradable a la vista de un consumidor final.

En lo académico el campo de la domótica es un nuevo espacio para los estudiantes que se encuentran cursando materias, tales como Sensores, Comunicación y Sistemas Microprocesados II.

Decir también los profesores serán beneficiados, ya que por medio de este proyecto pueden hacer una clase más didáctica y practica con sus estudiantes.

Delimitación

Delimitación Temporal

El tiempo a desarrollar el proyecto es de 6 meses a raíz de la aceptación por la dirección de carrera.

Delimitación Espacial

Este módulo se usará en el laboratorio de automatización industrial de la Universidad Politécnica Salesiana Sede Guayaquil, carrera de Ingeniería Electrónica.

Delimitación Académica

- Se desarrollarán 5 prácticas de amplio alcance para la implementación del panel por completo.
- El módulo tiene alcance para materias de la carrera Ingeniería Electrónica como por ejemplo, microcontroladores II, robótica y demostraciones en instalaciones civiles residenciales.
- Los elementos a usar serán:
 - 6 Luces pilotos de 120V
 - 1 Cerradura Eléctrica 120VAC – 12VDC
 - 1 Sensor PIR
 - 2 Cámaras de Seguridad USB
 - 1 Rapsberry Pi B+
 - 2 Sensores Magnéticos
 - 1 Teclado Matricial
 - 1 Sensor de Temperatura LM35
 - 1 Alarma Audible
 - 1 Ventilador 120V
 - 1 Válvula eléctrica 120V
 - 2 LCD's 2x16

2. Objetivos

2.1 Objetivo General

Desarrollar e implementar un módulo didáctico enfocado a la domótica con un sistema embebido, que será para el beneficio académico de los estudiantes de la Universidad Politécnica Salesiana Sede Guayaquil.

2.2 Objetivos Específicos

- Desarrollar 5 prácticas con Rapsberry Pi B+ como elemento de control.
- Elaborar láminas móviles de tal manera que los elementos estén en sitios estratégicos para dar facilidad al momento de implementar las prácticas.
- Emplear a la tarjeta Rapsberry Pi B+ y el software QT como un conjunto de trabajo teniendo así otra vía de solución para futuras prácticas en los laboratorios de la Universidad Politécnica Salesiana Sede Guayaquil.
- Aplicar en el campo de la domótica los sistemas embebidos apuntando a la relación futura con materias de la carrera Ingeniería Electrónica

3. Fundamento Teórico

3.1 Introducción a la Domótica

Un concepto directo sobre la domótica es que se basa en un uso simultáneo de campos como la electrónica, electricidad y sistemas informáticos todas estas son aplicadas a gestiones técnicas en una residencia inteligente.

- Seguridad: custodia y vigilancia frente a la intrusión, la inundación, el fuego, los escapes de gas, etc.
- Confort: programaciones horarias, escenarios luminosos, riego automático, etc.

Para ello, la domótica usa multitud de dispositivos que pueden ser distribuidos por toda la vivienda en función de las necesidades de los propietarios.

Básicamente estos dispositivos se pueden dividir en sensores y actuadores. Además, si la arquitectura es centralizada, se deben tener en cuenta los controladores. (Domótica)

Inicialmente, la única manera de construir una instalación domótica era con el uso de sensores y actuadores que se unían, con una arquitectura centralizada, a un autómatas o controlador que tenía embarcada toda la inteligencia que se exigía a la vivienda.

Casi siempre eran sistemas propietarios, muy pocos flexibles y que hacían muy difícil y costoso el aumento de las prestaciones. (Domótica)

Pero desde hace pocos años, gracias a la drástica bajada de los precios del hardware electrónico, es posible construir sensores y actuadores con inteligencia suficiente como para implementar “una red de área local” de control distribuido. (Domótica)

3.1.1 Ventajas de la Domótica.

La domótica ofrece varias ventajas. Se detallan entre las siguientes como las importantes:

- Ahorro de energía lo cual permite que tenga un funcionamiento óptimo y ahorro de consumo.
- En el aspecto económico ahorro de insumos.
- Máxima seguridad de vigilancia residencial.
- Acceso remoto con equipos doméstico.
- Debido a las ventajas de la domótica podemos decir que se alcanza un mayor confort muy superior

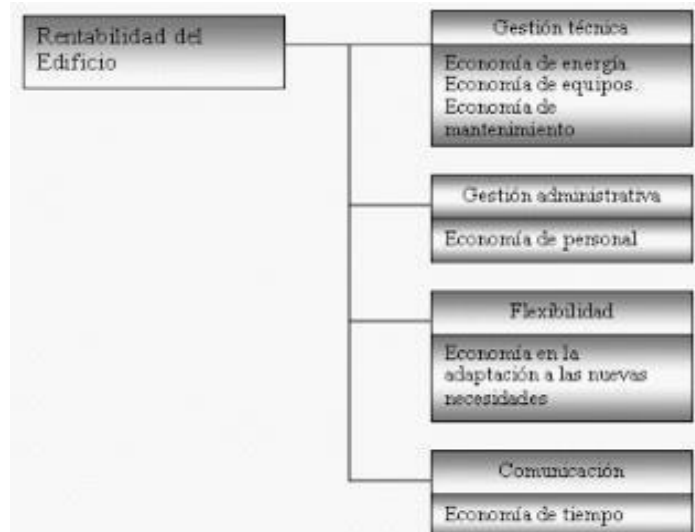


Figura 1. Ventajas de la domótica (Domoactualidad, 2014)

3.1.2 Desventajas de la Domótica.

Las desventajas de la domótica son menores respecto a las ventajas que podemos encontrar:

- Al no tener un amplio conocimiento pueden producirse errores en la programación.
- El uso de la Internet es una gran limitante si de enviar datos se trata y cuando se realice la comunicación con dispositivos remotos.

3.2 Descripción De Los Sistemas.

3.2.1 Puertas y ventanas.

En el caso de tener Puertas y Ventanas estas pueden ser integradas con el sistema de domótica.

El tener una residencia inteligente puede ser proyectado a mucho más allá de facilitar la vida de un usuario por comodidad, a una persona de capacidades físicas limitadas puede ayudar al usuario tanto abrir como cerrar las puertas y ventanas a través del sistema de domótica.

En caso de detectar fuego, gas o humo pueden ser programables para abrirse automáticamente y ser así una vía de escape para el usuario.

3.2.2 Seguridad.

Al ser un sistema con distintas clases de sensores y dispositivos, y con este medio de control por programación nos da la ventaja de poder configurar los sensores a nuestra disposición, así como una mayor configuración de las cámaras, como cuando se activaran y en qué momento grabaran, los sistemas de domótica tienen la ventaja de poder programar a la misma unidad para distintas funciones.

3.3 Raspberry Pi

Es un es un ordenador de menor tamaño que una computadora normal, para su respectivo uso se conecta a un televisor mediante puerto HDMI y mediante puerto USB se conecta un teclado y un mouse.

Aunque su capacidad y procesador es menor a la de una computadora de escritorio puede ser utilizado por muchas de las cosas que su PC de escritorio hace, como hojas de cálculo, procesadores de texto, etc.

La ventaja que encontramos en la ingeniería con la Raspberry Pi es que podemos usar la GPIO, esto nos permite varias funciones y proyectos orientado al control y comunicación entre tarjetas.

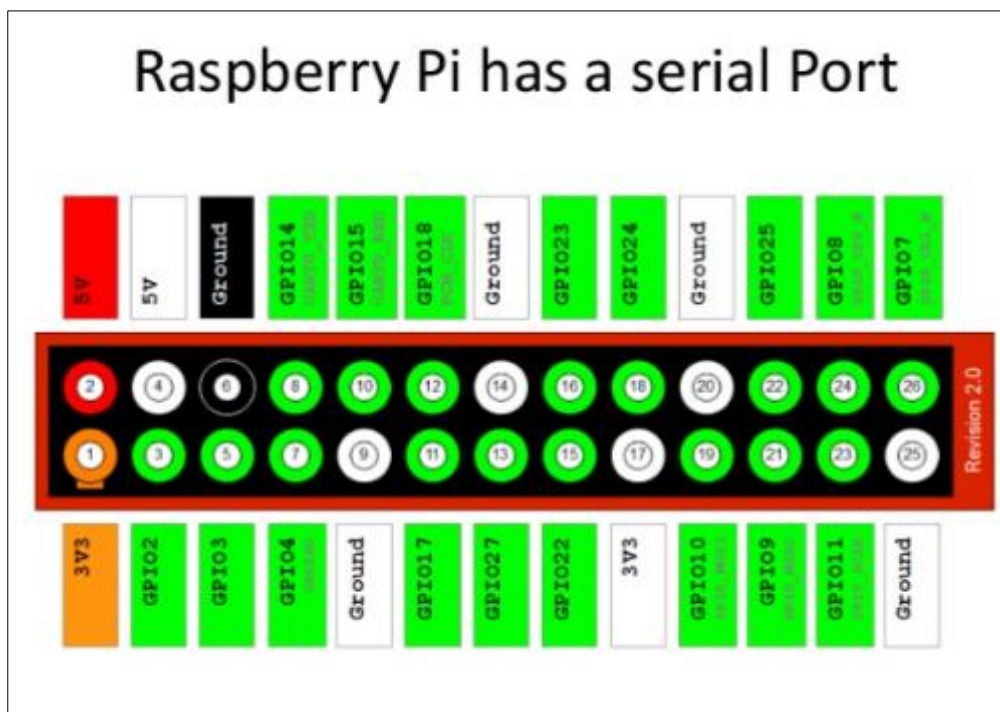


Figura 2. Puertos GPIO de Raspberry Pi b. (Muthu, 2014)

3.4 Raspberry Pi B+

La versión tercera de la Raspberry Pi (Raspberry Pi B+) tiene una mayor capacidad, el procesador es más rápido y tiene una mayor memoria RAM en comparación a sus predecesores. (Foundation, 2011 - 2016)

Especificaciones técnicas de la Raspberry PI B+ en comparación a las versiones anteriores.

	Raspberry / Model A	Raspberry / Model A+	Raspberry / Model B	Raspberry / Model B+
SoC	BROADCOM BCM2835	BROADCOM BCM2835	BROADCOM BCM2835	BROADCOM BCM2835
CPU	ARM 11 ARMV6 700 MHz	ARM 11 ARMV6 700 MHz	ARM 11 ARMV6 700 MHz	ARM 11 ARMV6 700 MHz
GPU	BROADCOM VIDEOCORE IV 250Mhz OPENGL ES 2.0	BROADCOM VIDEOCORE IV 250Mhz OPENGL ES 2.0	BROADCOM VIDEOCORE IV 250Mhz OPENGL ES 2.0	BROADCOM VIDEOCORE IV 250Mhz OPENGL ES 2.0
Memoria RAM	256 MB LPDDR SDRAM 400Mhz	256 MB LPDDR SDRAM 400Mhz	512 MB LPDDR SDRAM 400Mhz	512 MB LPDDR SDRAM 400Mhz
Puertos USB	1	1	2	4
GPIO	26 PINES	40 PINES	26 PINES	40 PINES
Video	HDMI 1.4 1920x1200	HDMI 1.4 1920x1200	HDMI 1.4 1920x1200	HDMI 1.4 1920x1200
Almacenamiento	SD	Micro SD	SD	Micro SD
Ethernet 10/100MBPS	NO	NO	SI	SI
Tamaño	85.60 X 56.5 MM	65 X 56.5 MM	85.60 X 56.5 MM	85.60 X 56.5 MM
Peso (g)	45	23	45	45

Tabla 1. Comparación de tarjetas Raspberry. (PE, 2014)


La mejor definición de la Raspberry Pi sería la de ordenador de placa única, ya que tiene casi todos los componentes de un PC en una única placa electrónica. (Foundation, 2011 - 2016)

Los principales componentes de la Raspberry Pi son:

MicroUSB: Es el sistema de alimentación de la Raspberry, con un cargador de microUSB común podemos darle corriente. Es recomendable que sea de 2ª que es el amperaje con el que trabaja regularmente la Raspberry Pi B+. (Foundation, 2011 - 2016)

GPIO: El puerto GPIO es lo que diferencia a la Raspberry Pi B+ de otros sistemas embebidos, nos permite hacer varias funciones con este puerto. Mediante estas entradas y salidas de propósito general podremos hacer que nuestra raspberry pi interactue con el exterior abriendo y cerrando contactos, encendiendo LEDs, conociendo el estado de un interruptor, etc. (Foundation, 2011 - 2016)

Raspberry Pi B+ J8 Header

Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I2C)		DC Power 5v	04
05	GPIO03 (SCL1 , I2C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I2C ID EEPROM)		(I2C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Rev. 1.1
16/07/2014

<http://www.element14.com>

Figura 3. Puertos GPIO de Raspberry PI b+. (Raspberrypi, 2015)

La tensión con la que trabaja es de 3.3V para alto, y 0V para bajo, la corriente máxima que puede suministrar es de 16mA. (Foundation, 2011 - 2016)

Inter-Integrated Circuit (I2C): I2C es un bus de comunicaciones serie en el que se pueden conectar múltiples periféricos I2C esclavos. (Foundation, 2011 - 2016)

RasPi se configura como maestro del bus. Pines del conector GPIO:

-3 SDA Data

-5 SCL Clock

Serial Peripheral Interface (SPI)

SPI es un bus de comunicaciones serie síncrono (con reloj) en el que se pueden conectar múltiples periféricos SPI esclavos. (Foundation, 2011 - 2016)

RasPi se configura como maestro del bus.

Pines del conector GPIO:

-19 MOSI Master Out, Slave In

-21 MISO Master In, Slave Out

-23 SCLK Serial Clock

-24 CE0 Channel Enable 0. Also known as Slave Select (SS)

-26 CE1 Channel Enable 1. Also known as Slave Select (SS)

Universal Asynchronous Receiver/Transmitter (UART)

UART es un bus de conexión serie asíncrono compatible con el estándar RS232 (atención al circuito adaptador ya que el estándar RS232 trabaja a 12v y no a 3,3v)

Pines del conector GPIO:

-8 TX Transmit

-10 RX Receive

USB: La versión B+ cuenta con 4 puertos USB para diferentes propósitos, el más común es para el teclado, raton y el adaptador WiFi. (Nergiza, 2012)

MICRO SD: También tenemos un slot de tarjeta micro SD que hará el uso de disco duro, donde se guardará la configuración del sistema operativo y todos los archivos que estén en la computadora. (Nergiza, 2012)

HDMI: podemos usar un televisor con conexión HDMI para interactuar con la Raspberry Pi, aunque no es necesario ya que podemos interactuar con una computadora mediante diferente tipo de conexiones como escritorio remoto. (Nergiza, 2012)

ETHERNET: La Raspberry Pi cuenta con conexión a internet mediante un puerto Ethernet

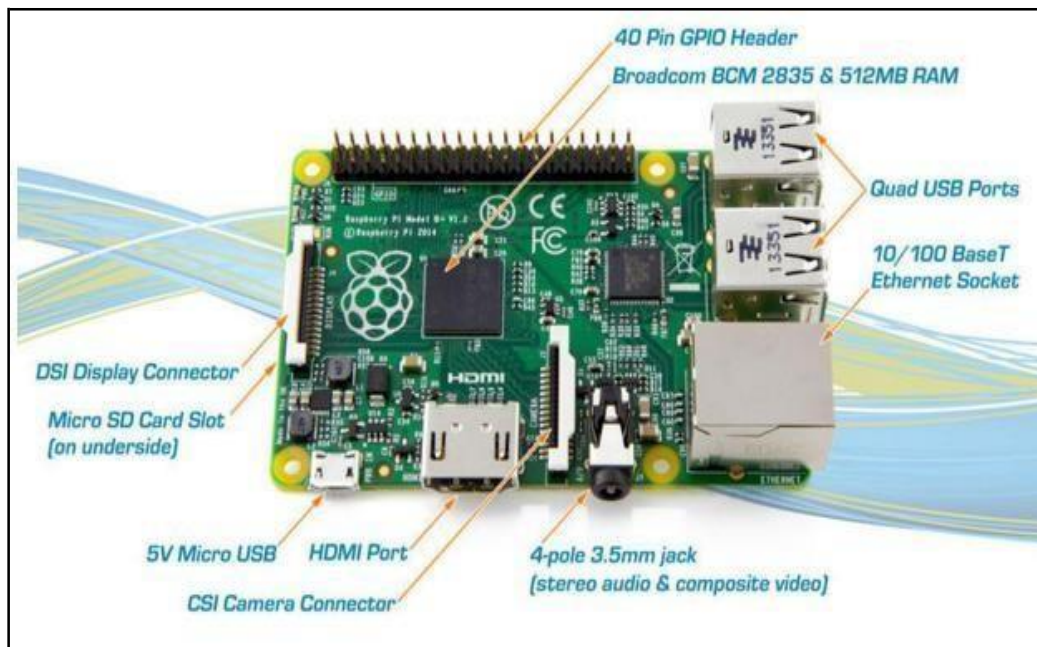


Figura 4. Elementos de la Tarjeta Raspberry Pi B+. (Raspberrypi, 2015)

3.5 Sistema Operativo

La Raspberry Pi está diseñada para trabajar con sistemas operativos basado en Linux, para el propósito de nuestro proyecto usamos la Raspbian debido a que está orientado para usos generales. (Nergiza, 2012)

RASPBIAN: Es un Debian Linux adaptado para la Raspberry, es el más versátil y el que nos permite instalar y modificar más cosas. (Nergiza, 2012)

3.6 Consumo de una Raspberry Pi b+

La Raspberry Pi es energizada con 5V, consume de corriente dependiendo de que periféricos se conecten a la tarjeta, el modelo B+ usa entre 700-1000mA dependiendo de que periféricos se conecten. (FUNDACIÓN, 2011)

Los pines del puerto GPIO pueden distribuir 50mA moralmente distribuido a través de todos los pines, individualmente cada pin puede entregar hasta 16mA.

El puerto HDMI usa 50mA, el módulo de la cámara de Raspberry requiere 250mA, un teclado y un mouse normal usar 100mA. (FUNDACIÓN, 2011)

Esto nos indica que para nuestro proyecto no son necesario elementos que resistan gran corriente. (FUNDACIÓN, 2011)

En cuanto la temperatura que puede alcanzar, al tener un consumo tan bajo parece que no disipa mucho calor, en concreto en las pruebas que hemos hecho no se han superado los 45°C de temperatura de la placa, aun habiendo hecho las pruebas con la carcasa de metacrilato (cerrada) que aparece en las fotos. (FUNDACIÓN, 2011)

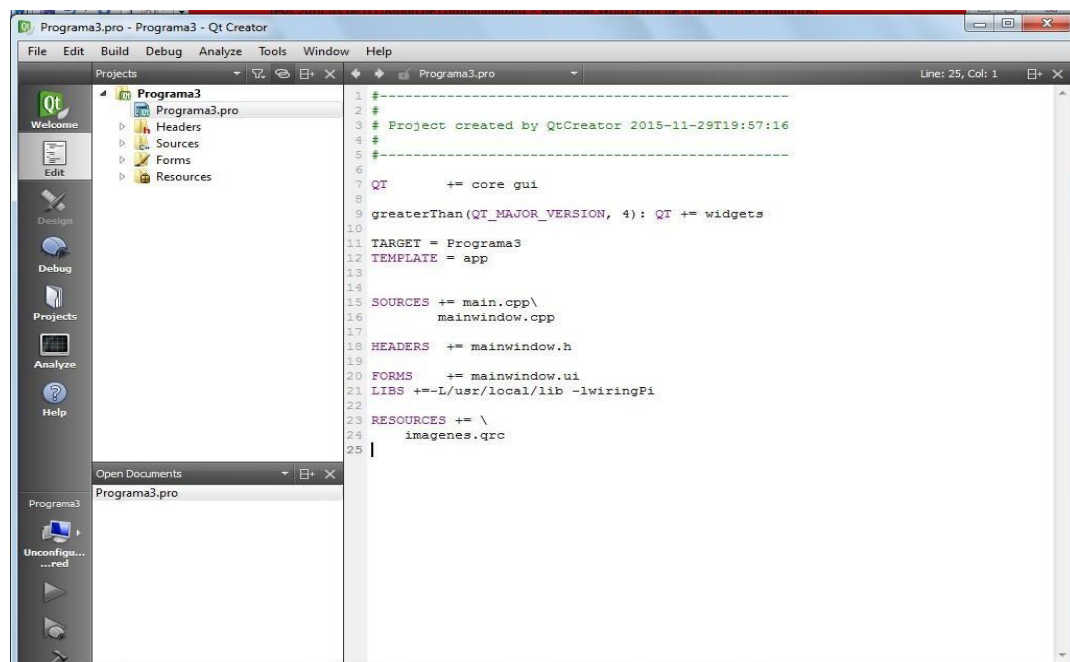
3.7 Qt Creator

Qt Creator es un excelente IDE multiplataforma para desarrollar aplicaciones en C++ de manera sencilla y rápida. Como su nombre lo indica, está basado en la librería Qt y cuenta con las siguientes características principales: (Glatelier, 2009)

- Editor avanzado para C++.
- Diseñador de formularios (GUI) integrado.
- Herramientas para la administración y construcción de proyectos.
- Completado automático.
- Depurador visual.

Qt Creator es un IDE creado por Trolltech para el desarrollo de aplicaciones con las bibliotecas Qt, requiriendo su versión 4.x. Está disponible para los sistemas operativos Linux, Max y Windows, permitiendo al desarrollador crear aplicaciones para múltiples sistemas o plataformas móviles. La versión actual es la 2.3.0. (Glatelier, 2009)

Figura 5. Ventana principal de Qt Creator



Fuente: Los Autores

3.8 Características

Principales características de Qt Creator:

- Posee un avanzado editor de código C++.
- Además soporta los lenguajes: C#.NET Languages (Mono), Python:PyQt y PySide, Ada, Pascal, Perl, PHP y Ruby.
- Posee también una GUI integrada y diseñador de formularios.
- Herramienta para proyectos y administración.
- Ayuda sensible al contexto integrado.
- Depurador visual.
- Resaltado y auto-completado de código.

3.9 Características útiles

Qt Creator incluye una amplia gama de características útiles. Entre ellos se encuentran:

- Smart Code Editor: El editor de código ofrece resaltado de sintaxis, así como de código.
- Qt4 Project Generating Wizard : Este asistente permite al usuario generar un proyecto para una aplicación de consola, una aplicación GUI, o una librería de C++.
- Qt Help Integration : Toda la documentación se puede acceder fácilmente haciendo clic en el botón de Ayuda(Help).
- Qt Designer Integration : En la interfaz de usuario se pueden diseñar formas en Qt Creator.
- Locator : Una potente herramienta de navegación que permite al usuario localizar los archivos y el uso mínimo de clases de teclado.
- Support for qmake's .pro file format : El archivo .Pro se usa como descripción del proyecto.
- Debugging Interface to GDB : Las solicitudes pueden ser depuradas en Qt Creator utilizando una interfaz gráfico para el depurador GNU simbólico.

3.10 Integración de Qt Designer

Qt Creator está totalmente integrado con Qt Designer para ayudarle a diseñar la interfaz de usuario las formas como lo haría con la versión independiente. Qt Designer también incluye la integración y gestión de proyectos de código.

3.10.1 Teclado de navegación.

Qt Creator abastece no sólo a los desarrolladores que están acostumbrados a utilizar el ratón, sino también para los desarrolladores que se sienten más cómodos con el teclado.

Una amplia gama de atajos en el teclado y navegación están disponibles para ayudar a acelerar el proceso de desarrollo de su aplicación. (EcuRed, 2016)

3.10.2 Librerías

WiringPi es una librería escrita en C y liberada bajo licencia GNU LGPLv3, que puede ser empleada en varios lenguajes de programación, además de C y C++, con alguna pequeña modificación en forma de adaptación.

Su principal uso es en la programación de periféricos a través de los pines de General Purpose Input Output (GPIO).

WiringPi, además, ofrece un comando que permite programar y configurar los pines de la GPIO, pudiendo efectuar la lectura y escritura de los pines desde la línea de comandos.

3.10.3 Instalación utilizando GIT.

Git es un software de control de versiones, libre y de código abierto, diseñado para manejar todo tipo de proyectos, desde pequeños a grandes con rapidez y eficiencia. (Rpiplus, 2013)

En caso de no estar instalado GIT se debería instalar. Para ello en primer lugar hay que actualizar los repositorios y posteriormente instalarlo. (Rpiplus, 2013)

- `sudo apt-get update` `sudo apt-get install git-core` Posteriormente hay que obtener la librería mediante GIT
- `git clone git://git.drogon.net/wiringPi`

Una vez descargado, hay que ir al directorio donde se encuentra y obtener una versión actual.

- `cd wiringPi`
- `git pull origin`

A continuación, se debe ejecutar el script de creación e instalación en el directorio de WiringPi.

- `./build`

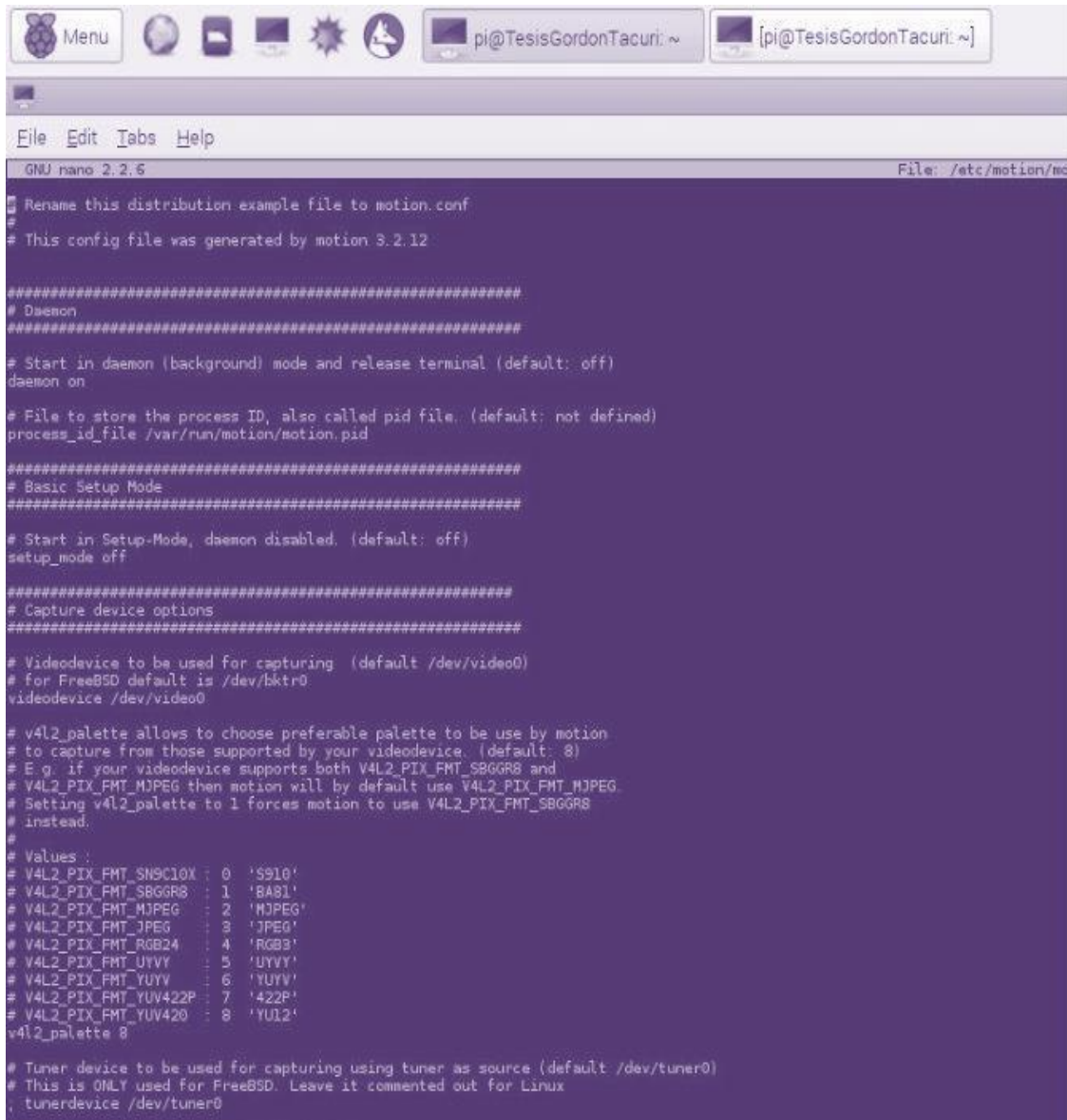
El nuevo script se encarga de compilar y ejecutar todo, sin que sea necesaria la intervención del usuario. (Rpiplus, 2013)

3.11 Motion

Motion es un programa que supervisa la señal de vídeo desde una o más cámaras y es capaz de detectar si una parte significativa de la imagen ha cambiado.

Este programa fue diseñado para trabajar con computadoras que tengan un servidor de Linux, el programa está escrito en C. No tiene absolutamente ninguna interfaz gráfica de usuario. (Motion, 2016)

Figura 6. Archivo Motion



```
GNU nano 2.2.5 File: /etc/motion/mo
# Rename this distribution example file to motion.conf
# This config file was generated by motion 3.2.12

#####
# Daemon
#####

# Start in daemon (background) mode and release terminal (default: off)
daemon on

# File to store the process ID, also called pid file. (default: not defined)
process_id_file /var/run/motion/motion.pid

#####
# Basic Setup Mode
#####

# Start in Setup-Mode, daemon disabled. (default: off)
setup_mode off

#####
# Capture device options
#####

# Videodevice to be used for capturing. (default /dev/video0)
# for FreeBSD default is /dev/bktr0.
videodevice /dev/video0

# v4l2_palette allows to choose preferable palette to be use by motion
# to capture from those supported by your videodevice. (default: 8)
# E.g. if your videodevice supports both V4L2_PIX_FMT_SBGGR8 and
# V4L2_PIX_FMT_MJPEG then motion will by default use V4L2_PIX_FMT_MJPEG.
# Setting v4l2_palette to 1 forces motion to use V4L2_PIX_FMT_SBGGR8
# instead.
#
# Values :
# V4L2_PIX_FMT_SN9C10X : 0 'S910'
# V4L2_PIX_FMT_SBGGR8  : 1 'BA81'
# V4L2_PIX_FMT_MJPEG   : 2 'MJPEG'
# V4L2_PIX_FMT_JPEG    : 3 'JPEG'
# V4L2_PIX_FMT_RGB24   : 4 'RGB3'
# V4L2_PIX_FMT_UYVY    : 5 'UYVY'
# V4L2_PIX_FMT_YUYV    : 6 'YUYV'
# V4L2_PIX_FMT_YUV422P : 7 '422P'
# V4L2_PIX_FMT_YUV420  : 8 'YU12'
v4l2_palette 8

# Tuner device to be used for capturing using tuner as source (default /dev/tuner0)
# This is ONLY used for FreeBSD. Leave it commented out for Linux
, tunerdevice /dev/tuner0
```

Fuente: Los Autores

Instalar Motion

En el terminal ingresaremos los siguientes códigos

- **sudo apt-get update**
- **sudo apt-get upgrade**

Estos códigos sirven para poder actualizar nuestro servidor, cosa q es necesaria antes de instalar cualquier programa (Motion, 2016)

Luego ingresamos el siguiente comando

- **sudo apt-get install motion**

Una vez instalado el Motion necesitamos hacer ciertas configuraciones en el programa, dependerá más del uso que le demos en el programa, para acceder al programa usaremos el siguiente código (Motion, 2016)

- **sudo nano /etc/motion/motion.conf**

Los cambios que haremos serán: Daemon ON

- **v4l2_palette 6**
- **image width 320**
- **image height 240**
- **snapshot_interval 5**
- **webcam_port 8001 webcam_localhost off**
- **control_port 8888 control_localhost off**

3.12 Escritorio Remoto.

El escritorio remoto nos permitirá una conexión remota con la Raspberry Pi B+ mediante una computadora cuando este en la misma red de internet.

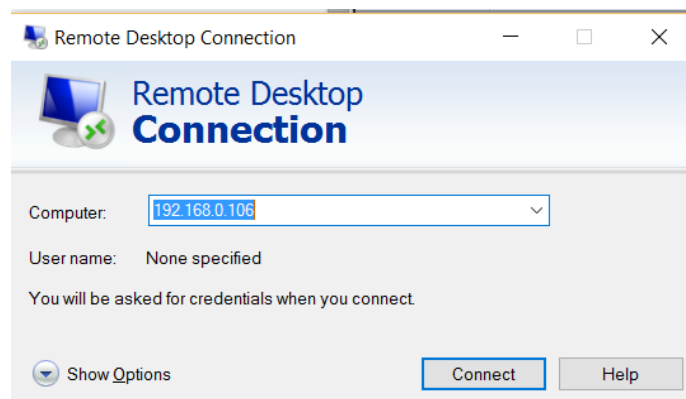
Esto funcionara para que podamos usar la Raspberry pi con una computadora, y usaremos el teclado, el mouse y monitor de esta computadora para controlar el de la Raspberry.

Para poder usar el escritorio remoto necesitamos instalar en la raspberry una librería En el terminal ingresamos el siguiente código (Electroensaimada, 2014)

- **sudo apt-get install xrdp**

Con esto la Raspberry estará disponible para poder usar la conexión de escritorio remoto En el PC abriremos el programa de Conexión a Escritorio Remoto y ahí ingresaremos la IP de la Raspberry PI (Electroensaimada, 2014)

Figura 7. Escritorio Remoto



Fuente: Los Autores

Luego ingresamos la clave y la contraseña y podremos ingresar a la Raspberry Pi B+ desde una PC. (Electroensaimada, 2014)

Figura 8. Entorno de Ingreso XRDP



Fuente: Los Autores

3.13 Luz piloto

Es una luz que indica cual número o condiciones normales de un sistema o dispositivo existe. Una luz piloto es también conocida como una luz monitor o de monitor. (Harok, 2015)

En nuestro proyecto la luz piloto representa un foco de 110V de una vivienda normal. (Harok, 2015)



Figura 9. Luces Piloto 110V (Harok, 2015)

3.14 Chapa eléctrica

La cerradura eléctrica es muy común para dar un acceso de ingreso o paso a un punto donde se use el sistema de domótica. Estas cerraduras eléctricas trabajan con 12VCA y es regulado mediante un transformador de 110V a 12V (Yalelatinoamerica, 2015)



Figura 10. Chapa eléctrica (Yalelatinoamerica, 2015)

3.15 Bocina

Una sirena del tipo electrónico está compuesta básicamente de la unidad de control que en su interior almacena una secuencia de tonos, y posiblemente altavoces conectados a esta unidad. Las sirenas de tipo electrónico tienen un uso extendido, adecuadas para un correcto operar seguido (continuo), es de muy bajo consumo de energía y su fuerza en el tono es proporcional al voltaje que ingrese. (Tvc, 2000)



Figura 11. Alarma auditiva (Tvc, 2000)

3.16 Sensor infrarrojo pasivo

Un sensor infrarrojo pasivo (sensor PIR) es un sensor electrónico que mide la luz infrarroja (IR) radiada de los objetos situados en su campo de visión.

Todos los objetos con una temperatura por encima del cero absoluto emiten calor la energía en forma de radiación. Por lo general, esta radiación es invisible para el ojo humano, ya que irradia en longitudes de onda infrarrojas, pero puede ser detectado por dispositivos electrónicos diseñados para tal propósito. (Electronics, s.f.)

El término pasivo, en este caso, se refiere al hecho de que los dispositivos PIR no generan o irradian cualquier energía para fines de detección. (Electronics, s.f.)

Trabajan en su totalidad para la detección de la energía emitida por otros objetos. Es importante tener en cuenta que los sensores PIR no detectan o miden “calor” pero se, sino que detectan la radiación infrarroja emitida por un objeto, que es diferente pero que a menudo está asociado/correlacionado con la temperatura del objeto (por ejemplo, un detector de rayos X o rayos gamma no sería considerado un detector de calor, a pesar de las altas temperaturas que pueden causar la emisión de la radiación X o gamma) (Electronics, s.f.)



Figura 12. Sensor P.I.R. (Electronics, 2016)

3.17 Sensor magnético

Un sensor magnético tiene la capacidad de detectar campos magnéticos provocados por imanes o corrientes de tipo eléctrica. (Directlink, 2015)

El interruptor Reed es el principal que generalmente no es el móvil, el funcionamiento es con dos láminas metálicas de material ferromagnético incrustadas en el interior de unas pequeñas bases que son atraídas en presencia de un campo magnético teniendo así un circuito cerrado. (Directlink, 2015)



Figura 13. Sensor Magnético (Directlink, 2015)

3.18 Sensor de temperatura

Un sensor de temperatura es un dispositivo capaz de interpretar el cambio de temperatura en el ambiente, o líquidos y lo transforma en señales eléctricas. (Mileti,2012)

3.18.1 Sensor de temperatura lm35

El **LM35** es un sensor de temperatura con una precisión calibrada de 1 °C. Su rango de medición abarca desde -55 °C hasta 150 °C. La salida es lineal y cada grado Celsius equivale a 10 mV. (Mileti, 2012)

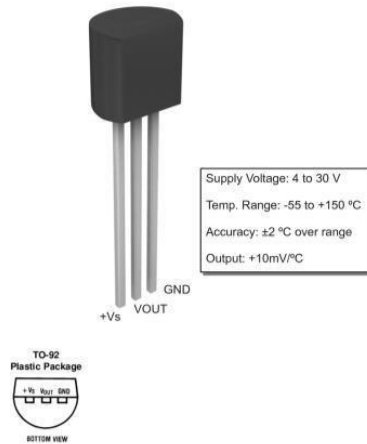


Figura 14. Sensor de Temperatura (Mileti, 2012)

Sus características más relevantes son:

- Está calibrado directamente en grados Celsius.
- La tensión de salida es proporcional a la temperatura.
- Tiene una precisión garantizada de 0.5 °C a 25 °C.
- Baja impedancia de salida.
- Baja corriente de alimentación (60 µA).
- Bajo coste.

3.19 Teclado matricial

El teclado matricial en base es una unión de varios pulsantes los cuales en un extremo estarán conectado a vcc y en el otro será la señal que receptara el PIC o cualquier microcontrolador que se usó. (Microcontroladores, 2009)

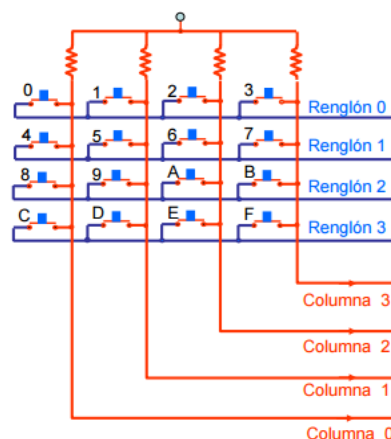


Figura 15. DataSheet de Teclado Matricial (Microcontroladores, 2009)

3.20 Display alfanumérico

Un display alfanumérico de matriz de puntos (dot-matrix) es un dispositivo de interfaz humano, formado por una pantalla de cristal líquido o LCD (Liquid Crystal Display) sobre la que se pueden mostrar mensajes formados por distintos caracteres: letras, números, símbolos (Marcelo, 2006)



Figura 16. Display de Dimensiones 2x16 (Marcelo, 2006)

3.21 Válvula solenoide

Una válvula solenoide es una válvula eléctrica utilizada para controlar el paso de gas (sistemas neumáticos) o fluidos (sistemas hidráulicos).

La apertura o cierre de la válvula se basa en impulsos electromagnéticos de un solenoide que trabaja junto a un muelle diseñado para devolver a la válvula a su posición neutral cuándo el solenoide se desactiva. (Curiosoando, 2011)

Figura 17. Válvula Solenoide



Fuente: Los autores

3.22 Interfaces.

Figura 18. Tarjeta Módulo Interfaz de Potencia de 4 canales



Fuente: Los Autores

Para que la Raspberry Pi B+ sea compatible con un elemento que trabaja a 110 V, teniendo en cuenta que la Raspberry Pi B+ trabaja máximo con 5V es necesario usar una tarjeta de interfaz.

Se adquirió las Tarjetas Módulo Interfaz de Potencia de 4 canales de salida a relé, con entradas optoacopladas para tener así una protección a la tarjeta de desarrollo, en este caso la Raspberry Pi B+. Con estas tarjetas podemos controlar cargas de 110V/220V AC o a su vez cargas DC.

3.22.1 Board APM 4550 PIC 18F4550

El PIC18F4550 es uno de los más populares microcontroladores cuando de conectividad USB se trata, además es un buen paso para dejar a la familia PIC16F, y pasar al siguiente nivel porque no solo incluye conectividad USB, sino que además tiene más memoria RAM EEPROM, y FLASH que el común de la familia PIC16F.

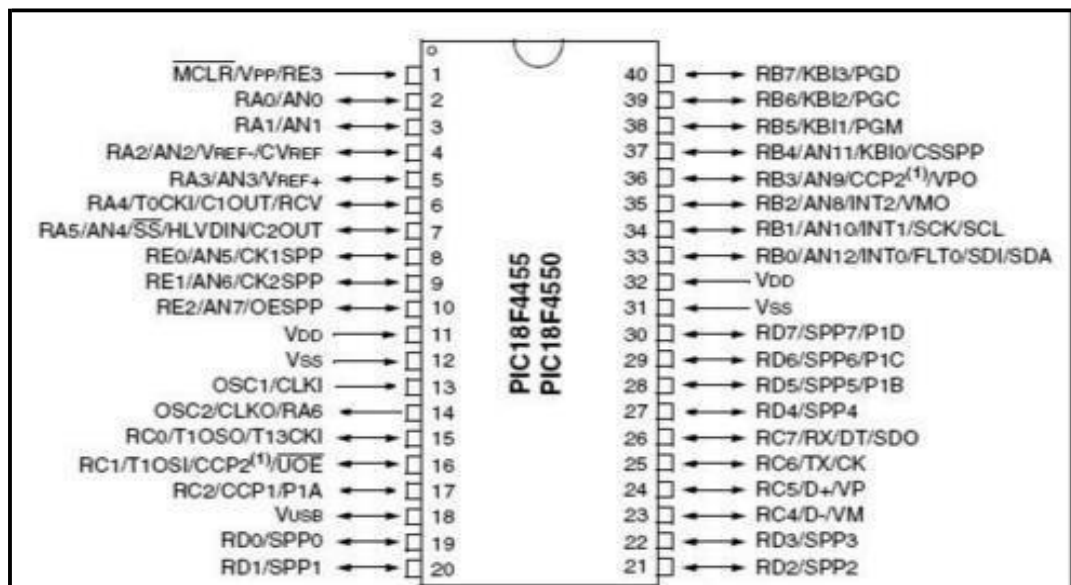


Figura 19. DataSheet PIC18F4550 (Nextiafenix, 3012)

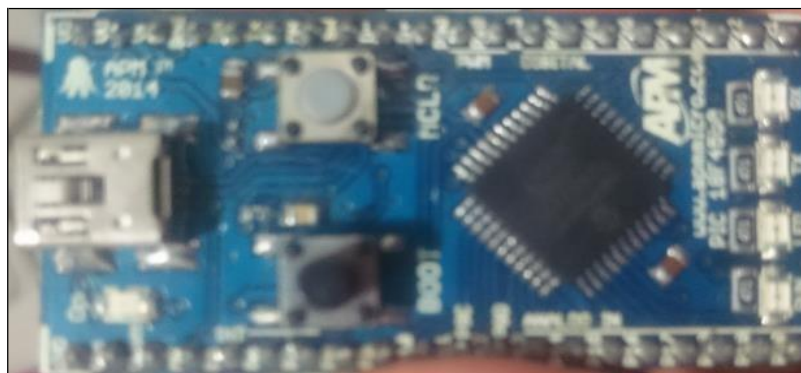
Ideal para pequeñas potencias y aplicaciones de conectividad que benefician de la disponibilidad de los tres puertos seriales:

FS-USB (12 Mbit / s), I²C y SPI (hasta 10 Mbit / s). Su gran capacidad de memoria RAM para almacenamiento temporal y su memoria FLASH de programa mejoradas hacen que sea ideal para el control integrado y aplicaciones de monitoreo que requieren conexión periódica con un ordenador personal a través de USB para la carga / descarga de datos y / o actualizaciones de firmware. (Nextiafenix, 3012)

Principales Características

- Puerto USB V2.0 que puede trabajar en modo low speed (1.5 Mb/s) y Full speed (12 Mb/s).
- RAM 1-Kbyte accesible por USB.
- Reloj externo hasta de 48 MHz.
- Oscilador interno de 31 KHz – 8 MHz configurable por software.
- Pines con salida de alta corriente de hasta 25 mA.
- timers de los cuales: hasta 2 pueden ser utilizados como modulos de captura/comparación/PWM.
- Puerto USART con soporte para comunicaciones MSSP, SPI e I²C.
- Memoria FLASH con 100,000 ciclos de lecturas escritura típicos.
- Memoria EEPROM con 1, 000,000 ciclos de lectura escritura típicos y retención de datos de hasta 40 años.
- Programación con código de protección.

Figura 20. PIC 18F4550



Fuente: Los Autores

3.23 Desarrollo de láminas móviles

Las láminas tienen como base una plancha de Alucobond donde se llevaron a cabo las perforaciones de borneras y bases para los elementos.

Para las perforaciones de las borneras se realizaron a 1.2cm de diámetro y las prensaestopa varían según las necesidades $\frac{3}{4}$ y $\frac{1}{4}$ de pulgada.

Para la impresión se usó planchas que permiten la impresión a laser y con relieve, a continuación se detalla el formato de las láminas en Autocad 2015 versión estudiante.

3.23.1 Alarma

Figura 21. Lámina de Alarma



Fuente: Los Autores

En esta lámina contiene una alarma auditiva la cual trabaja a 3.3V y consta con su respectiva simbología.

La lámina consta con dos borneras (Positivo y negativo) que alimentan directamente a la bocina y de una prensaestopa que permite el paso de cable del elemento a las borneras.

3.23.2 Cámaras 1 y 2

Figura 22. Láminas Cámaras 1 y 2

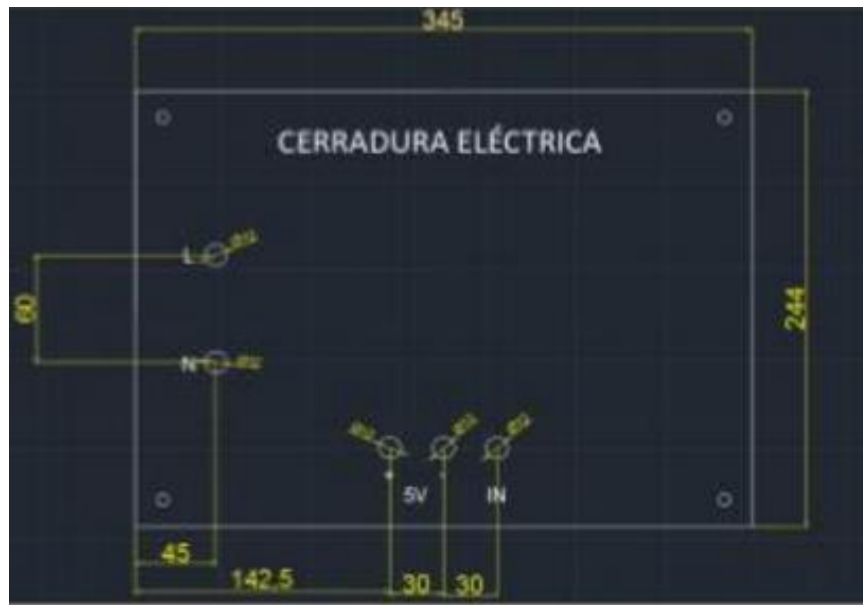


Fuente: Los Autores

Para el caso de las cámaras son colocadas una por lámina, son cámaras web de 5Mp con conexión de USB, una prensaestopa que permite el paso del cable USB a través de la lámina y de un conector USB hembra colocado en la parte posterior.

3.23.3 Cerradura Eléctrica

Figura 23. Lámina Cerradura Eléctrica

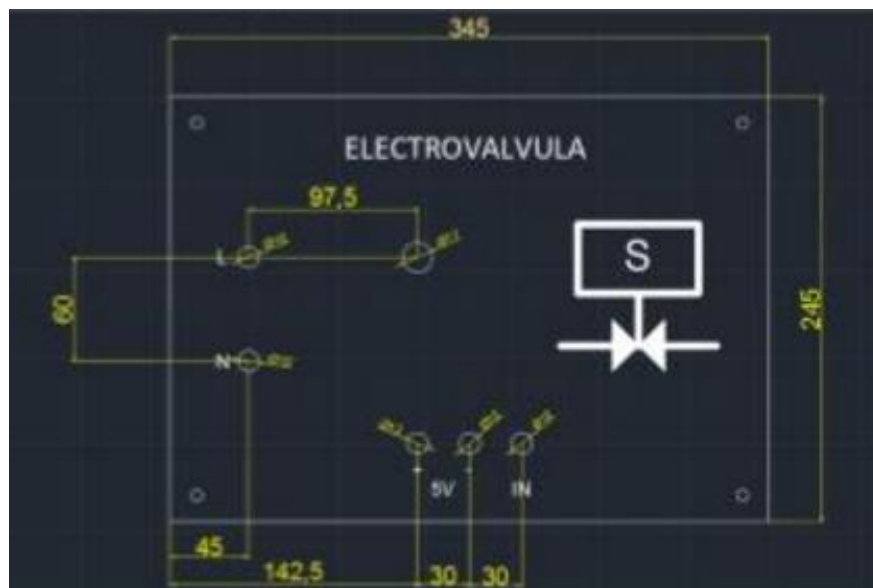


Fuente: Los Autores

En la lámina de cerradura eléctrica se encontrará en la parte inferior dos borneras de alimentación de 5VDC y una de Señal, la cual activará la cerradura eléctrica mediante un puerto GPIO de la Raspberry Pi B+, en la parte izquierda de la lámina se encuentran las borneras de alimentación de 110V.

3.23.4 Electroválvula

Figura 24. Lámina Electroválvula



Fuente: Los Autores

En la lámina de la electroválvula se encuentra su respectiva simbología, en la parte izquierda de la lámina se ubican dos borneras de alimentación de 110 V, y en la parte inferior se encuentran las borneras de alimentación de 5VDC y el de ingreso de señal que se conecta al puerto GPIO de la Raspberry Pi B+.

3.23.5 Banco de Luces

Figura 25. Lámina Banco de Luces

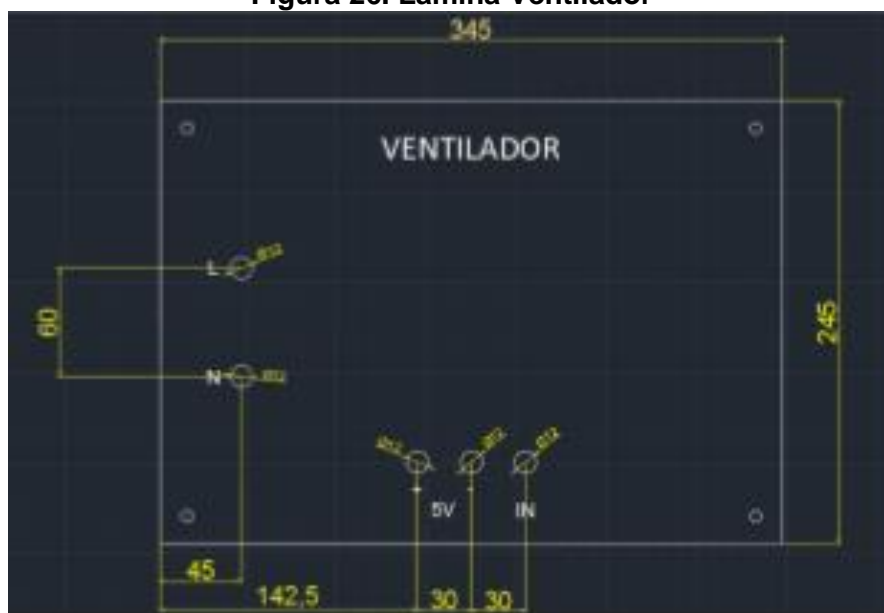


Fuente: Los Autores

El banco de luces constara de 6 luces pilotos, en la parte izquierda de la lámina se encuentran dos borneras de alimentación de 110V y 5VDC, en la parte inferior de cada luz piloto se encuentra el punto que realiza la comunicación al puerto GPIO de la Raspberry Pi B+, las perforaciones en las lamina que corresponden a las lucen piloto son de 2.2 cm de diámetro.

3.23.6 Ventilador

Figura 26. Lámina Ventilador

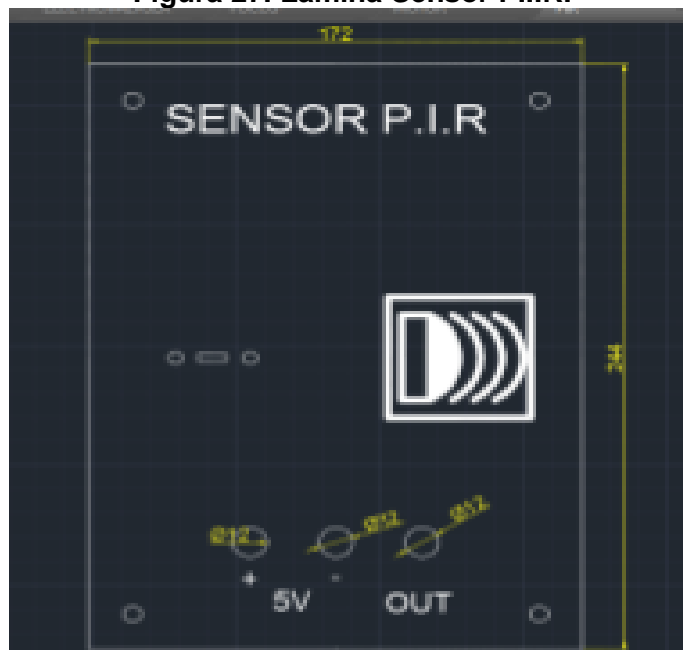


Fuente: Los Autores

En la lámina del ventilador se encuentra colocado en izquierdo la alimentación de 110V, y en la parte inferior se encuentra la alimentación de 5VDC y el ingreso de señal al puerto GPIO de la Raspberry Pi B+

3.23.7 Sensor P.I.R

Figura 27. Lámina Sensor P.I.R.

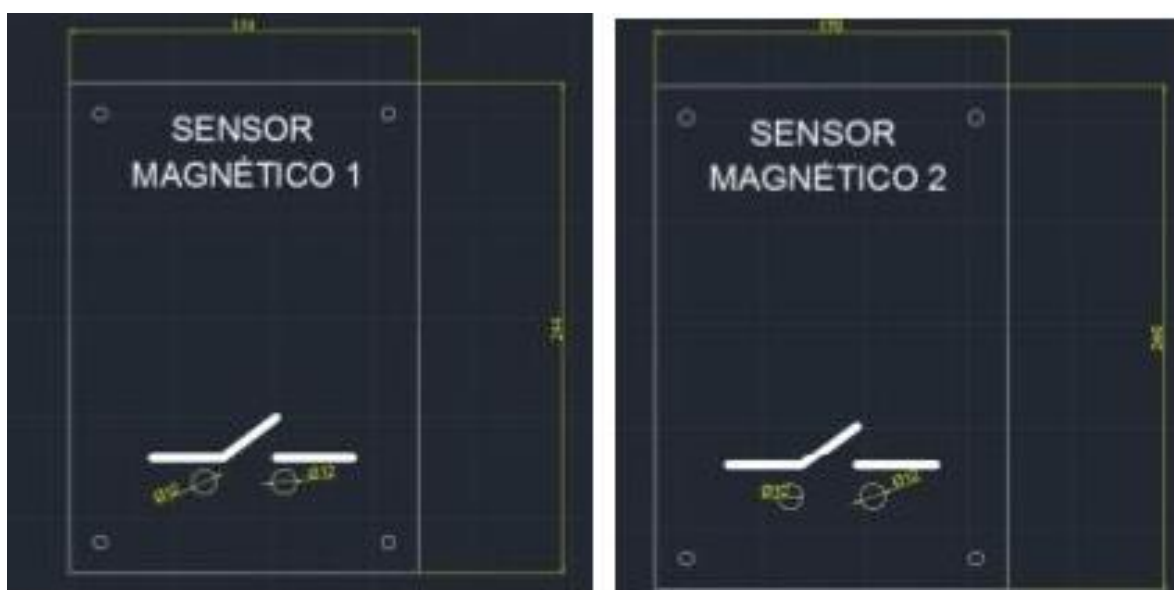


Fuente: Los Autores

La lámina del sensor PIR se encuentra con su respectiva simbología en la parte derecha, y en la parte inferior se encuentran 3 borneras, dos de alimentación de fuente de 5VDC y una de la salida del sensor hacia el puerto GPIO de la Raspberry Pi B+.

3.23.8 Sensor Magnético 1 y 2

Figura 28. Lámina Sensor Magnético 1 y 2



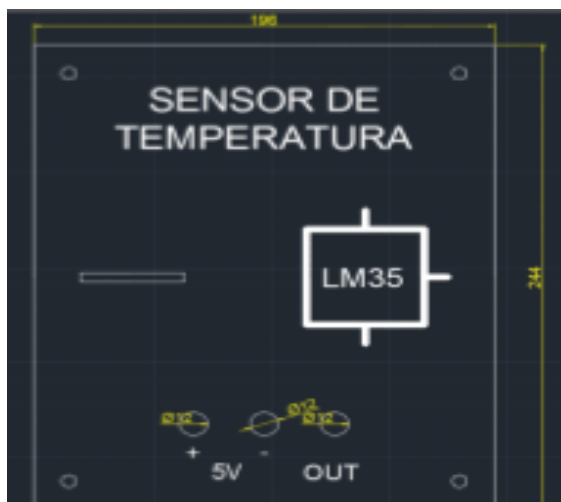
Fuente: Los Autores

Las láminas de los sensores magnéticos constan con la nomenclatura de un contacto

abierto ya que tiene un funcionamiento similar, consta de dos borneras las cuales se activan al cerrar sus contactos magnéticos y envían una señal al puerto GPIO de la Raspberry Pi B+

3.23.9 Sensor de temperatura

Figura 29. Lámina Sensor de Temperatura



Fuente: Los Autores

La lamina del sensor de temperatura consta en su parte derecha la simbología del LM 35, y en la parte inferior las dos borneras de alimentación de 5VDC y de la señal que ofrece el sensor. Además en su parte frontal constara de una pantalla LCD que indicara la temperatura que vaya recibiendo el sensor de Temperatura.

3.23.10 Ingreso de clave

Figura 30. Lámina Ingreso de Clave



Fuente: Los Autores

Para la lámina del ingreso de clave en su parte frontal de un teclado de 4 x 4, una pantalla LCD, y tres borneras, dos para alimentar con 5VDC la lámina y una que será la señal que de la lámina cuando se abra correctamente.

3.24 Presupuesto

Tabla 2. Presupuesto de Equipos

ELEMENTO	VALOR UNITARIO	CANT.	VALOR
Rapsberry Pi B+	\$ 200.00	1	\$ 200.00
Luces piloto	\$ 10.00	6	\$ 60.00
Sensor de proximidad	\$ 25.00	1	\$ 25.00
Sensores magnéticos	\$ 25.00	2	\$ 50.00
Cámara	\$ 50.00	1	\$ 50.00
Cerradura eléctrica	\$ 50.00	1	\$ 50.00
Teclado Numérico	\$ 25.00	1	\$ 25.00
Alarma	\$ 35.00	1	\$ 35.00
Válvulas eléctricas	\$ 75.00	1	\$ 75.00
Motor	\$ 55.00	1	\$ 55.00
Circuitos de fuerza y Tarjeta de protección	\$ 75.00	1	\$ 75.00
Conectores y borneras	\$300.00	-	\$ 300.00
Alucubone	\$75.00	1	\$75.00
Serigrafía	\$25.00	14	\$350.00
Cables de poder	\$15.00	6	\$90.00
Tuercas y tornillos	\$0.75	24	\$18.00
Tarjetas de Interfaz 5V a110V de 1 Relay	\$25.00	3	\$75.00
Tarjetas de Interfaz 5V a 110V de 4 Relay	\$55.00	2	\$110.00
Viáticos	\$125.00	-	\$125.00
Horas de Trabajo (\$20 / hora)	\$20.00	50(x2)	\$2.000
TOTAL			\$ 3843.00

Fuente: Los Autores

Pasos para llevar a cabo la comunicación entre Raspberry Pi B+ y una conexión Wi-Fi

Paso 1

Una vez que se tenga encendida la Raspberry Pi B+, ingresar la siguiente línea de comandos en el Terminal

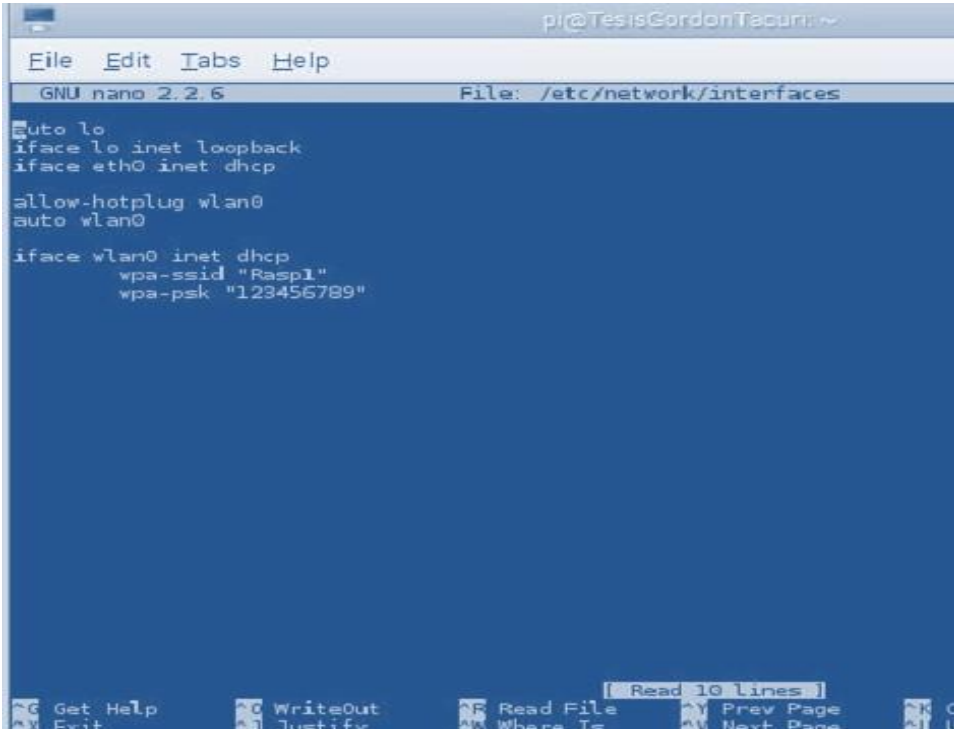
- **sudo nano / etc/network/interfaces**

Paso2

Se coloca el archivo de configuración de la interfaz de red a usar

- **auto lo**
- **iface lo inet loopback**
- **iface eth0 inet dhcp**
- **allow-hotplug wlan0**
- **auto wlan0**
- **iface wlan0 inet dhcp**
- **wpa-ssid "Rasp1"**
- **wpa-psk "123456789"**

Figura 31. Ingreso Mediante Clave



```
pi@TesisGordonTacuri:~  
File Edit Tabs Help  
GNU nano 2.2.6 File: /etc/network/interfaces  
auto lo  
iface lo inet loopback  
iface eth0 inet dhcp  
  
allow-hotplug wlan0  
auto wlan0  
  
iface wlan0 inet dhcp  
    wpa-ssid "Rasp1"  
    wpa-psk "123456789"  
  
Get Help WriteOut Read File Prev Page  
Exit Justify Where Is Next Page
```

Fuente: Los Autores

Paso 3

Se reinicia la Raspberry Pi B+ para guardar cambios, con la siguiente línea de comandos

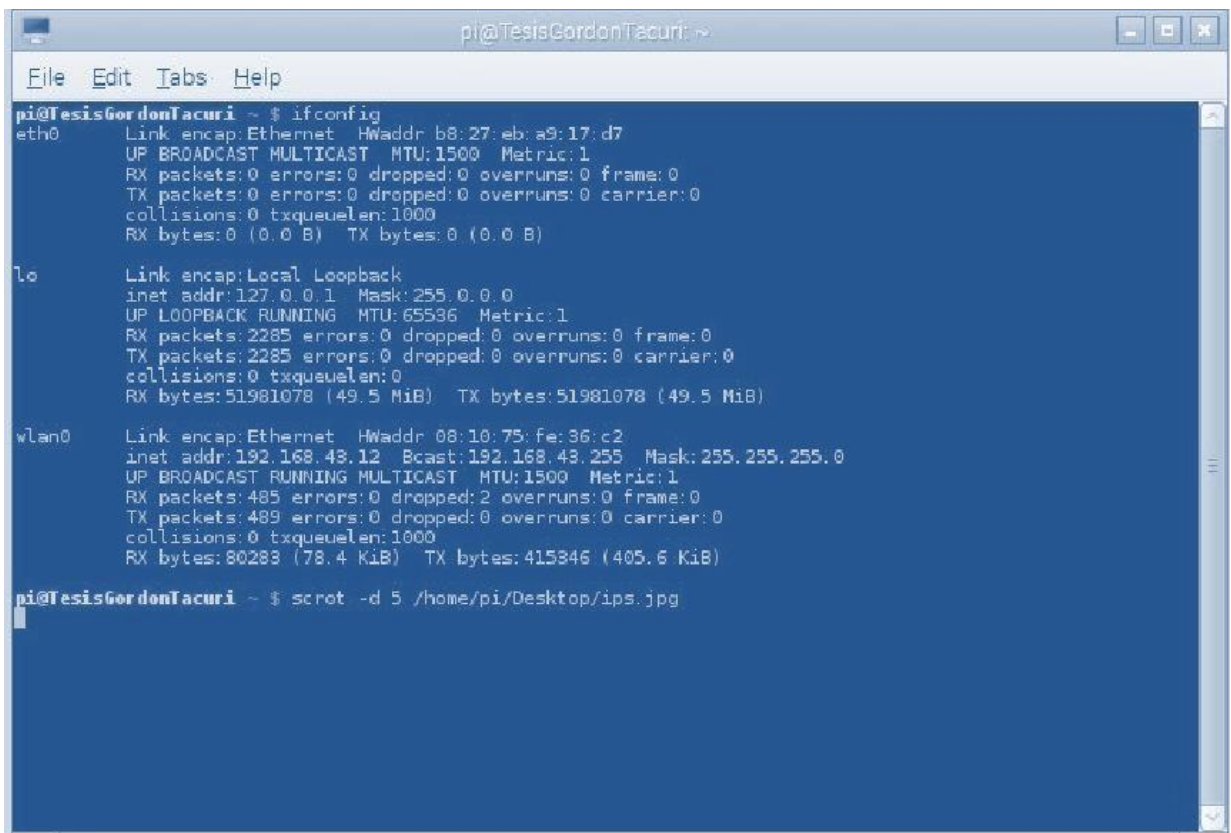
- `sudo shutdown -r now` Paso

Paso 4

Comprobar que el dispositivo esté conectado a Internet

Ifconfig

Figura 32. Verificación de conexión a Internet



```
pi@TesisGordonTacuri: ~  
File Edit Tabs Help  
pi@TesisGordonTacuri ~ $ ifconfig  
eth0      Link encap:Ethernet  HWaddr b8:27:eb:a9:17:d7  
          UP BROADCAST MULTICAST  MTU:1500  Metric:1  
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          UP LOOPBACK RUNNING  MTU:65536  Metric:1  
          RX packets:2285 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:2285 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:0  
          RX bytes:51981078 (49.5 MiB)  TX bytes:51981078 (49.5 MiB)  
  
wlan0     Link encap:Ethernet  HWaddr 08:10:75:fe:86:c2  
          inet addr:192.168.48.12  Bcast:192.168.48.255  Mask:255.255.255.0  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:485 errors:0 dropped:2 overruns:0 frame:0  
          TX packets:489 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:80288 (78.4 KiB)  TX bytes:415846 (405.6 KiB)  
  
pi@TesisGordonTacuri ~ $ scrot -d 5 /home/pi/Desktop/ips.jpg
```

Fuente: Los Autores

4 Marco Metodológico

Se analizaron varias tecnologías actuales que no están como asignatura de estudio para la universidad y entre esos temas estaba la domótica, y los diferentes sistemas que se pueden trabajar en este campo entre ellos los sistemas embebidos.

Mediante el método deductivo directo logramos llegar a la conclusión de que la Universidad no cuenta con una herramienta que le permita mostrar a los nuevos estudiantes sobre esta nueva tecnología específicamente la domótica

Mediante el método histórico se observó que los anteriores proyectos de domótica que fueron enfocados para un uso comercial o práctico en casa u oficinas, pero ninguno fue enfocado para un uso educativo, y nuestro proyecto es para la enseñanza donde los nuevos estudiantes puedan aprender a usar los elementos de un sistema de domótica y programarlo.

Haciendo uso del método analítico procedimos a investigar todos los métodos de configuración de un sistema domótica y se eligió los sistemas embebidos por su facilidad de programación e interacción con el usuario, además de q ofrece un sistema más amplio y amigable para el estudiante.

Mediante el método sistémico se pudo denotar que tipo de elementos tiene compatibilidad con la Rapsberry, entre ellos elegimos elementos digitales y analógicos. Para el caso de elementos de mayor voltaje investigamos la posibilidad de usar tarjeta de interfaces para la interacción entre la Rapsberry con elementos de media tensión.

5 Innovación

Nuestro proyecto innova en una tecnología que si ya había sido explorada nunca había sido enfocada para un sistema educativo.

En proyectos anteriores encontramos sistemas de domótica pero se han limitado en el uso de elementos y aplicaciones, en nuestro módulo hicimos uso de varias luces piloto, cámaras de seguridad y sensores varios.

Otros proyectos de domótica con Rapsberry Pi se encargan en manejar a distancia los elementos conectados de los cuales por lo general son solo focos, en nuestro proyecto realizamos una conexión de escritorio remoto para un uso a distancia mediante una computadora portátil y poder manejar todos los elementos conectados inclusive los elementos de mayor potencia.

El módulo cumple los objetivos al 100% de su funcionabilidad no existen errores eléctricos ni de programación, las condiciones en las cuales están establecidas las prácticas están ajustadas a los objetivos propuestos.

En nuestro Proyecto de Titulación para probar el alcance de la Rapsberry Pi B+, que no había tenido antes en otros proyectos, decidimos hacer una práctica con un enfoque industrial, en el cual colocamos una válvula que se cerrará y se abrirá automáticamente y también contamos de un ventilador que simula un motor dentro de un proceso.

El punto centro de impacto del proyecto es dirigido a los estudiantes de la Universidad Politécnica Salesiana, debido a que nuestro módulo está orientado para ser académico, las prácticas, la forma física del módulo y la programación se han realizado con componentes que sean reconocidos por los estudiantes.

Los estudiantes de 5 Nivel y superiores serán los beneficiados ya que los conocimientos previamente adquiridos son necesarios para poder dar uso del módulo.

Para los estudiantes que estén dispuestos a aprender una nueva tecnología en la cual pueden usar sensores y un nuevo lenguaje de programación, serán los interesados en nuestro proyecto y es en ellos donde se causará mayor impacto.

6 Resultados

El Módulo didáctico de domótica con la tarjeta Raspberry Pi B+ y el programa Qt Creator cumplió los objetivos dispuestos desde el inicio, ya que el microprocesador implementado fue un pilar fundamental para el correcto funcionamiento del proyecto.

Para la instalación del software Qt Creator en la Raspberry Pi B+ se debe dirigir al Terminal ya que es desde este punto donde se procede a la descarga de programas y aplicaciones, cabe indicar que para todo tipo de descarga es necesario estar conectados a Internet.

Para la lectura y escritura de datos es necesaria la librería Wiring Pi la cual es descargada del Terminal, esta librería permite a la Raspberry Pi B+ y Qt Creator una comunicación con los elementos del módulo ya sean análogos o digitales.

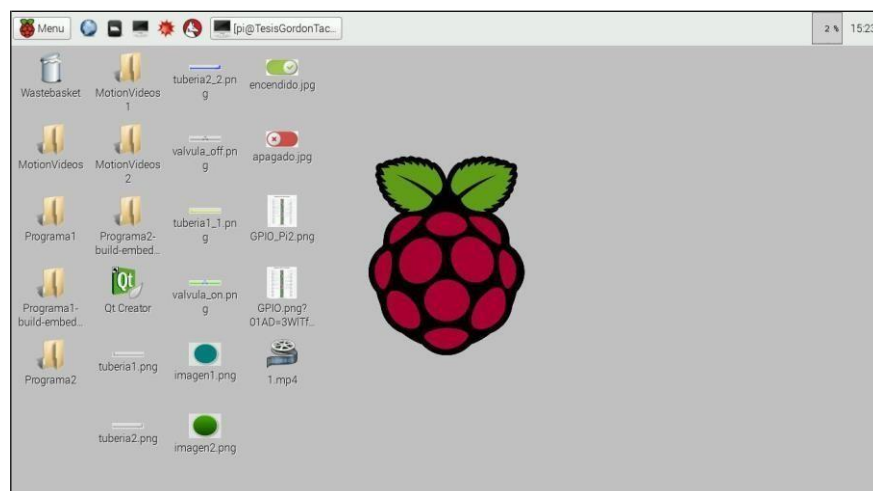
Para usar las Raspberry Pi B+ usamos una tarjeta de interfaz para que pueda ser más fácil su interacción con las borneras del modulo

Una vez encendida la Raspberry Pi B+ se necesitara abrir el programa QT creator, para este debemos seguir los siguientes pasos:

6.1 Funcionabilidad

El módulo cumple los objetivos al 100% de su funcionabilidad no existen errores eléctricos ni de programación, las condiciones en las cuales están establecidas las prácticas están ajustadas a los objetivos propuestos.

Figura 33. Interfaz Raspberry PI B+



Fuente: Los Autores

Primero debemos abrir el terminal y luego seguir los siguientes pasos:

Ingresar el comando `sudo -i` para ingresar en modo súper usuario. Luego, ingresar el comando `sudo QtCreator`.

Este proceso hicimos para poder abrir el programa en modo súper usuario, esto es necesario ya que ciertas librerías usadas en los programas necesitan permiso de súper usuario para poder interactuar correctamente con las entradas del puerto GPIO.

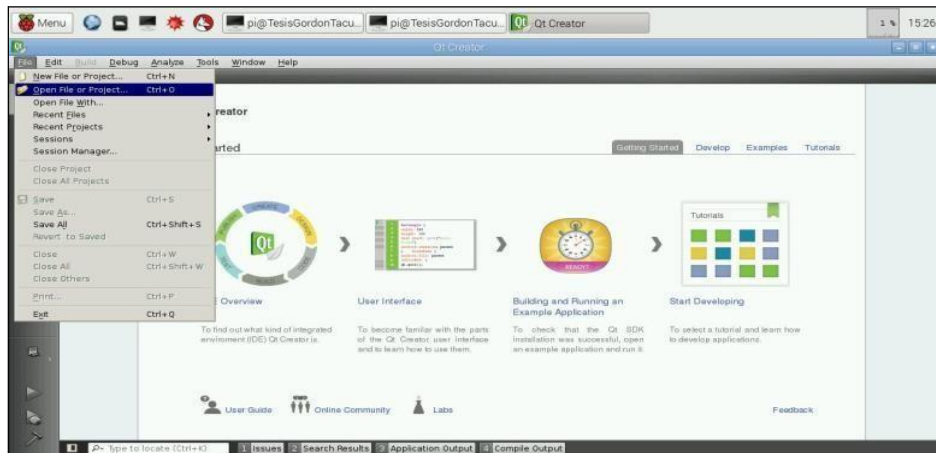
Una vez que abramos el programa Qt Creator para abrir vamos a `File/Open/Dirección del archivo`.

Figura 34. Ingreso al software Qt Creator



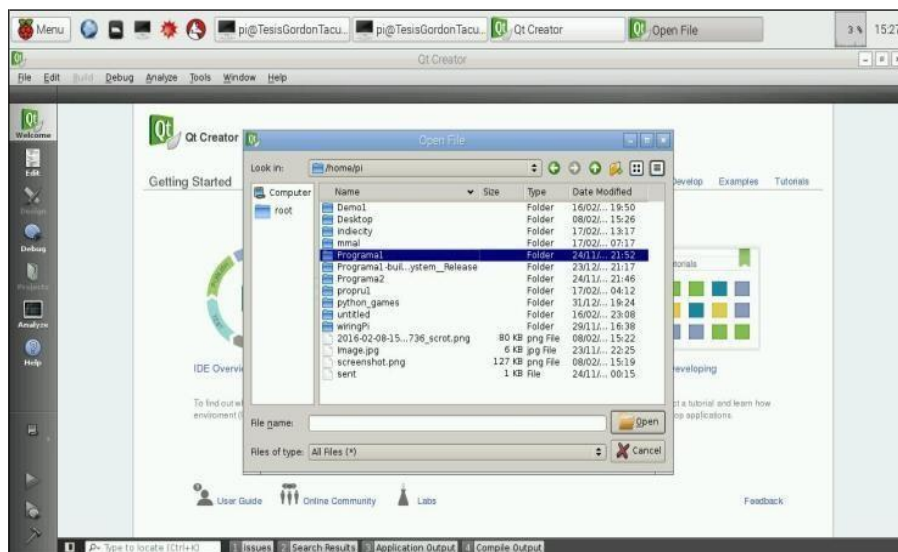
Fuente: Los Autores

Figura 35. Buscar el Proyecto desde de Qt Creator



Fuente: Los Autores

Figura 36. Seleccionar el proyecto en las Carpetas creadas por defecto



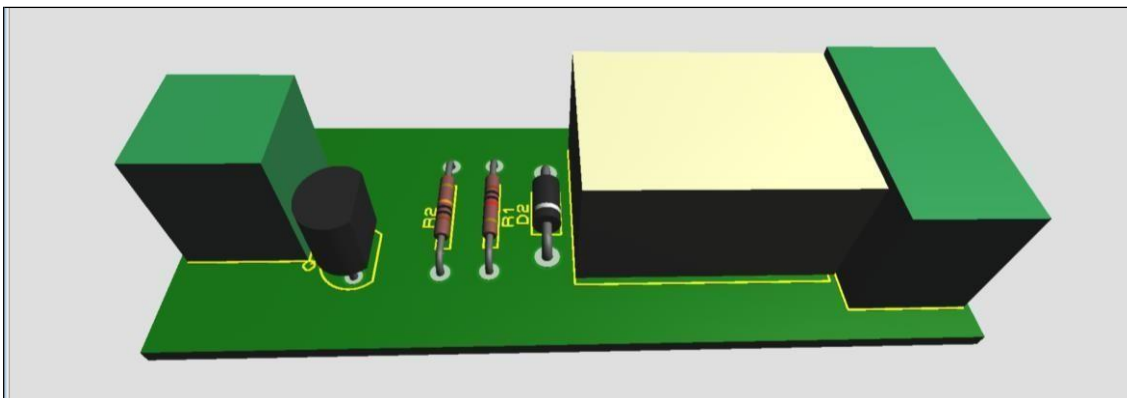
Fuente: Los Autores

6.2 Tarjeta Interfaz de Potencia de Dos Canales

Se llevó a cabo el diseño e implementación de Tarjetas Interfaz de Potencia de 2 canales de salida a relé, estas tarjetas fueron realizadas por Los Autores para llevar a cabo la comunicación con la tarjeta Raspberry Pi B+.

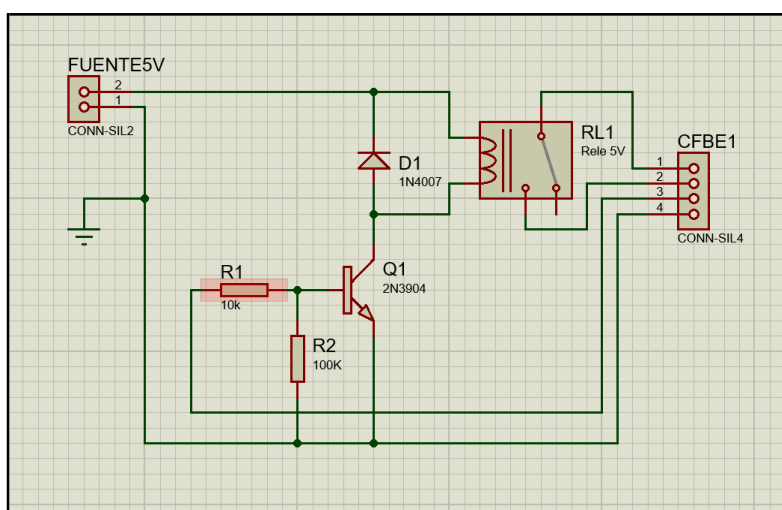
Con estas tarjetas podemos controlar cargas de 110V/ 220V AC o a su vez cargas DC.

Figura 37. Tarjeta Interfaz de Potencia de Dos Canales



Fuente: Los Autores

Figura 38. Esquema en Proteus de la Tarjeta Interfaz de Potencia



Fuente: Los Autores

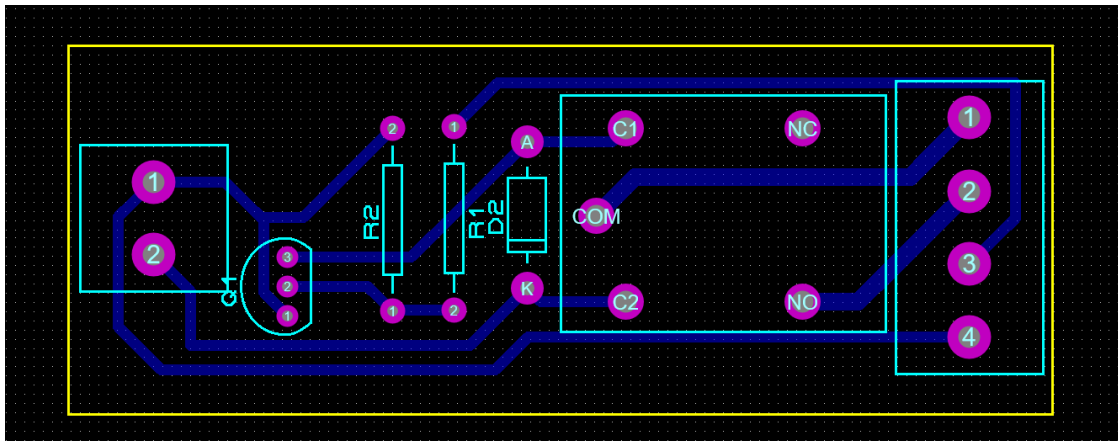
6.3 Esquema en Proteus de la Tarjeta Interfaz de Potencia

Se realizó el esquema en Proteus 8.1 para el diseño de una tarjeta de Interfaz de Potencia, la cual será de utilidad para las cargas de 110v. El diseño se llevó a cabo en este software en vista de que es el más dominado por los estudiantes para el diseño de tarjetas impresas a través de computadora.

Se usó Proteus 8.1 ya que sigue los reglamentos estrictamente industriales, con lo que los trabajos que se realicen en esta plataforma son muy seguros. A modo de ejemplo, el programa opera con distancias estandarizadas entre los puntos de circuitos integrados, y lo mismo se aplica a las dimensiones de los componentes y factores similares.

Este software a más de ofrecer una simulación nos brinda el esquemático previo a la impresión donde se puede modificar el ángulo y el grosor de la pista.

Figura 39. Esquema en Ares de la Tarjeta de Interfaz de Potencia



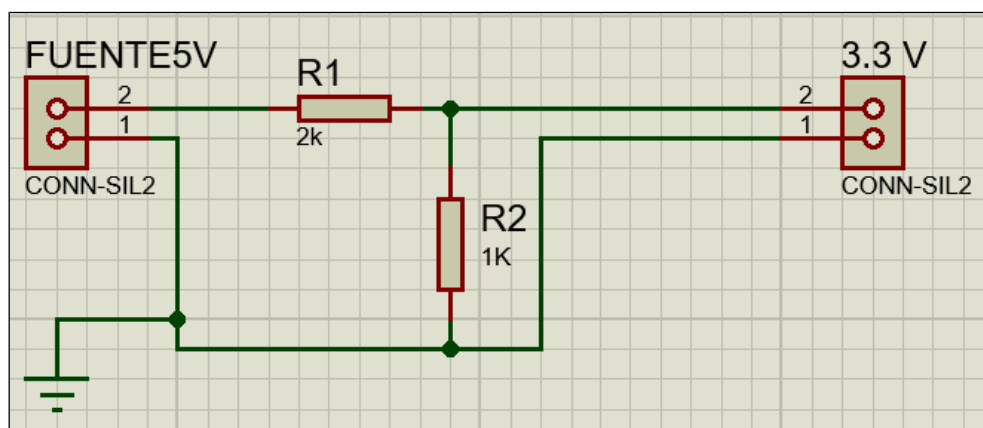
Fuente: Los Autores

6.4 Tarjeta de divisor de voltaje de 5V a 3.3V

Se usó un circuito divisor de voltaje para pasar de 5V a 3.3V, valor con el cual trabaja la tarjeta Raspberry Pi B+

El divisor de tensión tiene que ver con colocar dos resistencias: Una en serie con el elemento que se desea usar y la segunda en paralelo a tierra.

Figura 40. Esquema en Proteus del divisor de voltaje

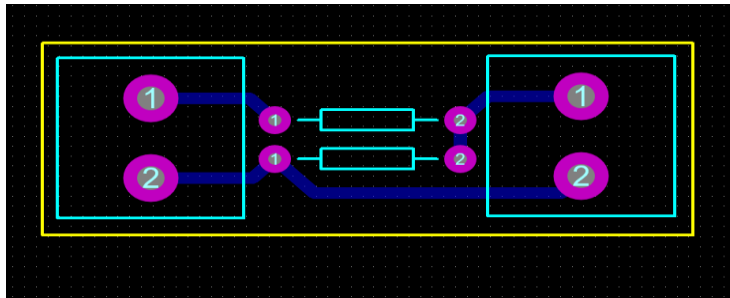


Fuente: Los Autores

La ventaja que se tiene en su eso es la versatilidad, ya que variando los valores de las resistencias se puede conseguir un rango de tensiones muy amplio.

Lo único necesario para conseguir 3.3V es que la resistencia que se esté en paralelo tenga el doble de valor que la resistencia que se tenga conectada al elemento.

Figura 41. Esquema en Ares de la Tarjeta del divisor de voltaje

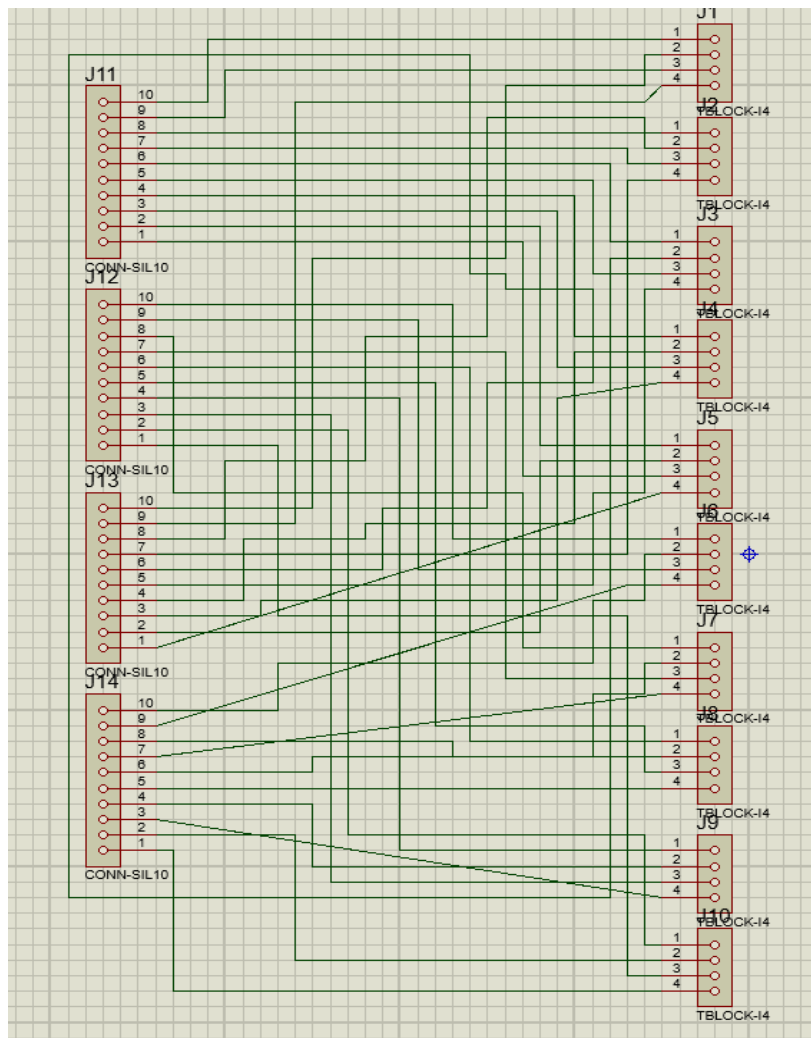


Fuente: Los Autores

6.5 Tarjeta para conexión de bus de datos

Se usó una tarjeta de extensión para la Raspberry Pi B+ con estilo de bus de datos, la cual es de gran utilidad en vista de que la conexión a las borneras colocadas dentro de las láminas está en una posición de 4x10 lo que dificultaba un estándar seguro para su uso.

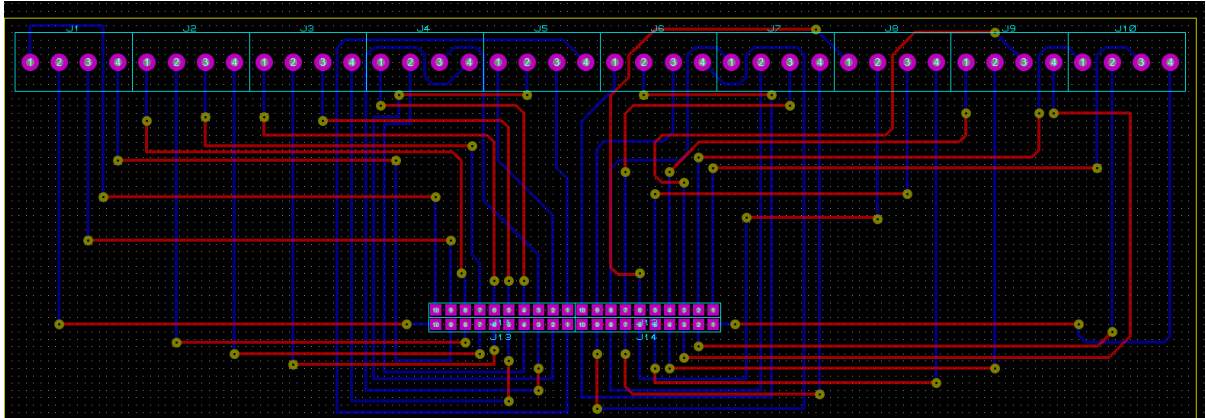
Figura 42. Esquema en Proteus de la conexión Raspberry Pi B+ / Borneras



Fuente: Los Autores

Las conexiones se llevaron a cabo a través de Jumpers los cuales iban conectados desde los espadines de la Raspberry Pi B+ a la tarjeta.

Figura 43. Esquema en Ares de la conexión Raspberry Pi B+ / Borneras



Fuente: Los Autores

6.6 Módulo didáctico de domótica con la tarjeta Raspberry Pi B+

Se usó elementos que trabajan con diferentes valores de voltaje como por ejemplo 3.3V, 120V, 5V y para las cámaras puertos USB.

Este módulo consta de 14 láminas de las cuales las únicas a permanecer fijas son:

- 6.6.1 Módulo de Raspberry Pi B+
- 6.6.2 Módulo de Fuentes

Se llegó a esta determinación en vista de que ambas son láminas de alimentación de voltaje y de señal, así se tiene un mayor confort para su uso.

Figura 44. Módulo didáctico de domótica con la tarjeta Raspberry Pi B+ y el programa Qt Creator




Fuente: Los Autores

Figura 45. Módulo de domótica



Fuente: Los Autores

6.7 Prácticas

		PRÁCTICA #1	
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
		LABORATORIO	AUTOMATIZACIÓN INDUSTRIAL
CARRERA	INGENIERÍA ELECTRÓNICA SEDE GUAYAQUIL		
PRÁCTICA	#1		

1. TEMA

Simulación e implementación de un sistema de control de luces

2. OBJETIVO GENERAL

Realizar la simulación e implementación de funcionamiento con un banco de luces a través de la Raspberry Pi B+ y Qt Creator

3. OBJETIVOS ESPECÍFICOS

- Realizar la respectiva programación en el software Qt Creator para el encendido de luces
- Diseñar una interfaz en el software Qt Creator para la simulación del encendido y apagado de luces
- Utilizar la tarjeta Raspberry Pi B+ para la implementación del proceso de encendido y apagado de luces

4. RECURSOS UTILIZADOS

Lámina de banco de luces, Módulo de Raspberry Pi B+, Software QT Creator, Laptop, Conductor macho y hembra para las conexiones, Fuente de alimentación de 120VAC

5. CONDICIONES

- El proceso para la práctica que simula el encendido y apagado de luces como se muestra en la Figura 36. Debe cumplir las siguientes condiciones:
- El encendido y apagado debe ser presentado en forma ascendente y descendente respectivamente para la primera prueba de funcionamiento

- Posteriormente el encendido y apagado debe darse de forma aleatoria para verificar que todos los focos trabajan correctamente bajo cualquier normativa
- La finalización del programa se dará con el uso exclusivo de un Push Bottom con la leyenda “Salir” para finalizar completamente el proceso

6. MARCO PROCEDIMENTAL

Una vez analizada cada condición del proceso para el desarrollo de la programación se toma en cuenta el uso de las GPIO usadas en el proyecto así como las respectivas alimentaciones de tensión

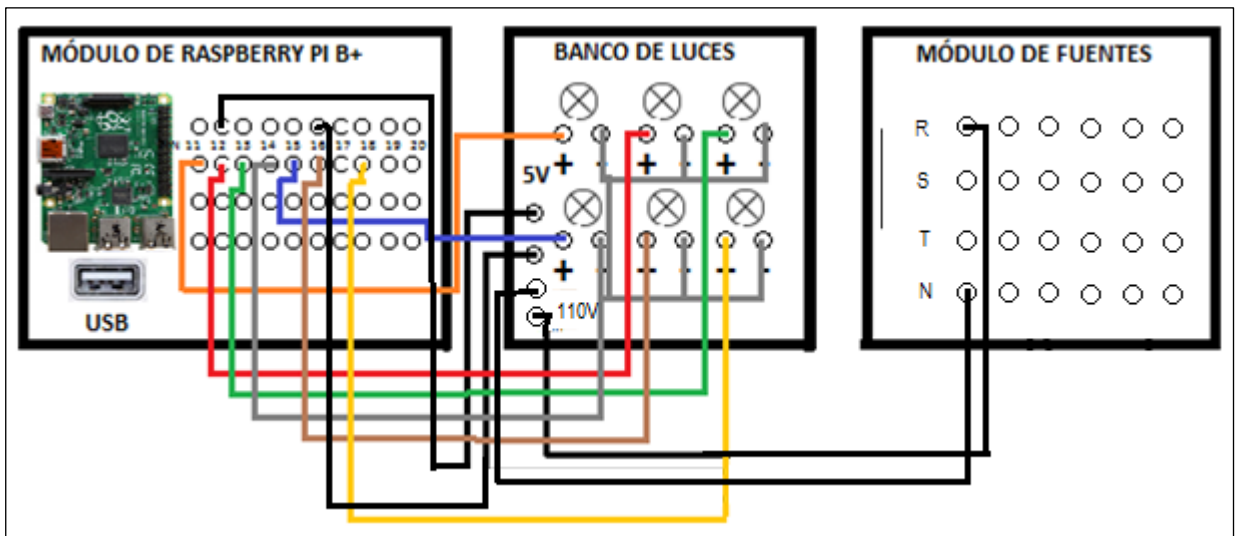
7. INTERFAZ DEL PROGRAMA

Figura 46. Interfaz de un sistema de control de luces Qt Creator



Fuente: Los Autores

Figura 47. Esquema de Control de Luces



Fuente: Los Autores

Figura 48. Funcionamiento de la Practica 1



Fuente: Los Autores

Programación:

Programa 1

Programa1.pro //Se declaran los elementos que irán en el proyecto

```
# Project created by QtCreator 2015-02-17T04:14:02
```

```
#-----
```

```
QT += core gui greaterThan(QT_MAJOR_VERSION, 4): QT += widgets
TARGET = Programa1 TEMPLATE = app
SOURCES += main.cpp\mainwindow.cpp HEADERS += mainwindow.h
FORMS += mainwindow.ui LIBS +=-L/usr/local/lib -lwiringPi RESOURCES
+= \
imagenes.qrc
```

Carpeta Headers

mainwindow.h

```
//Declarar librerias
```

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H
#include <QMainWindow>
#include "wiringPi.h"
#include <QTimer>
namespace Ui { class MainWindow;
}
class MainWindow : public QMainWindow
{
Q_OBJECT
public:
explicit MainWindow(QWidget *parent = 0);
~MainWindow();

private slots: //Declarar los eventos void on_be1_clicked();
void on_ba1_clicked(); void on_be2_clicked(); void on_ba2_clicked(); void
on_be3_clicked(); void on_ba3_clicked(); void on_be4_clicked(); void
on_ba1_4_clicked(); void on_be5_clicked(); void on_ba5_clicked(); void
on_be6_clicked();
void on_pushButton_clicked(); void on_ba6_clicked(); void inicio();
private:
Ui::MainWindow *ui;
};
#endif // MAINWINDOW_H
```


Carpeta Sources

main.cpp

```
#include <QApplication>
#include "mainwindow.h"
int main(int argc, char *argv[])
{
    QApplication a(argc, argv); MainWindow w; w.show();
    return a.exec();
}
```

Carpeta Sources

mainwindow.cpp

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
MainWindow::MainWindow(QWidget *parent) : QMainWindow(parent), ui(new
Ui::MainWindow)
{
    ui->setupUi(this);
    //Declarar variables wiringPiSetup(); pinMode(0,OUTPUT);
    pinMode(1,OUTPUT); pinMode(2,OUTPUT); pinMode(3,OUTPUT);
    pinMode(4,OUTPUT); pinMode(5,OUTPUT); inicio();
}
MainWindow::~MainWindow()
{
    delete ui;
}
//Eventos


void MainWindow::on_be1_clicked()
{
    digitalWrite(0,LOW);
    QPixmap pix("/home/pi/Programa1/encendido.jpg"); ui->lf1->setPixmap(pix);
}
void MainWindow::on_ba1_clicked()
{
    digitalWrite(0,HIGH);
    QPixmap pix("/home/pi/Programa1/apagado.jpg"); ui->lf1->setPixmap(pix);
}
void MainWindow::on_be2_clicked()
{
    digitalWrite(1,LOW);
    QPixmap pix("/home/pi/Programa1/encendido.jpg"); ui->lf2->setPixmap(pix);
}
void MainWindow::on_ba2_clicked()
{
    digitalWrite(1,HIGH);
    QPixmap pix("/home/pi/Programa1/apagado.jpg"); ui->lf2->setPixmap(pix);
}
void MainWindow::on_be3_clicked()
{
    digitalWrite(2,LOW);
```



```

QPixmap pix("/home/pi/Programa1/encendido.jpg"); ui->lf3->setPixmap(pix);
}
void MainWindow::on_ba3_clicked()
{
digitalWrite(2,HIGH);
QPixmap pix("/home/pi/Programa1/apagado.jpg"); ui->lf3->setPixmap(pix);
}
void MainWindow::on_be4_clicked()
{
digitalWrite(3,LOW);
QPixmap pix("/home/pi/Programa1/encendido.jpg"); ui->lf4->setPixmap(pix);
}
void MainWindow::on_ba1_4_clicked()
{
digitalWrite(3,HIGH);
QPixmap pix("/home/pi/Programa1/apagado.jpg"); ui->lf4->setPixmap(pix);
}
void MainWindow::on_be5_clicked()
{
digitalWrite(4,LOW);
QPixmap pix("/home/pi/Programa1/encendido.jpg"); ui->lf5->setPixmap(pix);
}
void MainWindow::on_ba5_clicked()
{
digitalWrite(4,HIGH);
QPixmap pix("/home/pi/Programa1/apagado.jpg"); ui->lf5->setPixmap(pix);
}
void MainWindow::on_be6_clicked()
{
digitalWrite(5,LOW);
QPixmap pix("/home/pi/Programa1/encendido.jpg"); ui->lf6->setPixmap(pix);
}
void MainWindow::on_pushButton_clicked()
{
digitalWrite(0,HIGH); digitalWrite(1,HIGH); digitalWrite(2,HIGH);
digitalWrite(3,HIGH); digitalWrite(4,HIGH); digitalWrite(5,HIGH); exit(0);
}
void MainWindow::on_ba6_clicked()
{
digitalWrite(5,HIGH);
QPixmap pix("/home/pi/Programa1/apagado.jpg"); ui->lf6->setPixmap(pix);
}
void MainWindow::inicio()
{
digitalWrite(0,HIGH); digitalWrite(1,HIGH); digitalWrite(2,HIGH);
digitalWrite(3,HIGH); digitalWrite(4,HIGH); digitalWrite(5,HIGH);
}

```

		PRÁCTICA #2
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	AUTOMATIZACIÓN INDUSTRIAL	
CARRERA	INGENIERÍA ELECTRÓNICA SEDE GUAYAQUIL	
PRÁCTICA	#2	

1. TEMA

Simulación e implementación de un sistema cerrado con cámaras de seguridad

2. OBJETIVO GENERAL

Realizar la simulación e implementación de un circuito cerrado de seguridad con dos webcam a través de la Raspberry Pi B+, Qt Creator y el software Motion

3. OBJETIVOS ESPECÍFICOS

- Realizar la respectiva programación en el software Qt Creator para dar inicio al monitoreo de seguridad desde las webcam (cámara de seguridad).
- Diseñar una interfaz en el software Qt Creator que permita la comunicación con el software MOTION
- Utilizar la tarjeta Raspberry Pi B+ para la implementación del proceso de encendido y apagado de luces
- Aprender a comunicar el software QT Creator con el software MOTION para el monitoreo de webcam (cámaras de seguridad)

4. RECURSOS UTILIZADOS

Cámaras de seguridad (Webcam), Software QT Creator, Software MOTION, Módulo de Rapsberry Pi B+, Laptop (Control remoto), Cable de extensión USB

5. CONDICIONES

- El proceso para la práctica que simula el monitoreo de un circuito cerrado de seguridad con webcam(cámaras de seguridad) como se muestra en la Figura 38 Debe cumplir las siguientes condiciones:
- Las condiciones para capturas de imágenes y tiempo de duración del video de seguridad debe ser dado en el software MOTION, las imágenes y videos captados por las webcam (cámaras de seguridad) debe estar en ruta en una carpeta creada por el usuario como se muestra en la Figura 38 (Interfaz del programa)
- Para accionar el funcionamiento de las cámaras se debe programar y diseñar en el software QT Creator la interfaz con un Push Bottom, el cual será el único medio para que las cámaras empiecen a realizar su trabajo.
- Para visualizar la grabación de las cámaras de lo puede hacer desde un dispositivo móvil o una laptop, para ambas opciones se debe contar con Internet para desde un navegador poder observar lo que se está grabando en OJO (tiempo real)
- La finalización del programa se dara una vez que el tiempo dado en el Software Motion cumpla la función del tiempo de activación y los archivos sean guardados correctamente en la carpeta de ruta.

6. MARCO PROCEDIMENTAL

Una vez analizada cada condición del proceso para el desarrollo de la programación se toma en cuenta el uso de puertos USB los cuales servirán para la activación correcta de las webcam (cámaras de seguridad)

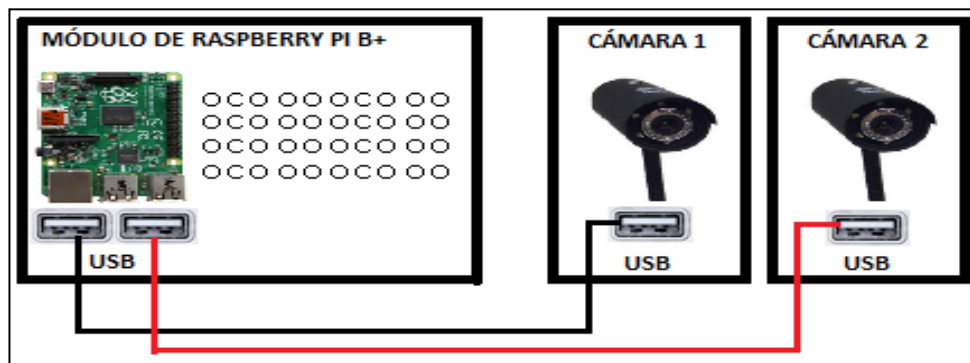
7. INTERFAZ DEL PROGRAMA

Figura 49. Interfaz del programa de la Practica 2



Fuente: Los Autores

Figura 50. Esquema de un sistema cerrado con cámaras de seguridad



Fuente: Los Autores

Figura 51. Prueba del manejo de las cámaras en el laboratorio



Fuente: Los Autores

Programación:

Programa2.pro //Se declaran los elementos que iran en el proyecto

```
#-----  
# Project created by QtCreator 2015-11-24T23:19:26  
#-----  
  
QT += core gui  
QT += webkit  
greaterThan(QT_MAJOR_VERSION, 4): QT += widgets TARGET = Programa2  
TEMPLATE = app  
SOURCES += main.cpp\ mainwindow.cpp  
HEADERS += mainwindow.h FORMS += mainwindow.ui
```

Carpeta Headers

```
Programa2.pro  
#ifndef MAINWINDOW_H  
#define MAINWINDOW_H  
#include <QMainWindow> namespace Ui { class MainWindow;  
}  
class MainWindow : public QMainWindow  
{  
Q_OBJECT  
public:  
explicit MainWindow(QWidget *parent = 0);  
~MainWindow();  
private slots: //Declarar los eventos void on_biniciar_clicked(); void on_bsalir_clicked();  
private:  
Ui::MainWindow *ui;  
};#endif // MAINWINDOW_H
```

Carpeta Sources


```
main.cpp  
#include <QApplication>  
#include "mainwindow.h"  
int main(int argc, char *argv[])  
{  
QApplication a(argc, argv); MainWindow w;  
w.show(); return a.exec();  
}
```

Carpeta Sources

```
mainwindow.cpp  
#include "mainwindow.h"  
#include "ui_mainwindow.h"  
MainWindow::MainWindow(QWidget *parent) : QMainWindow(parent), ui(new  
Ui::MainWindow)  
{  
ui->setupUi(this);  
}
```

```
MainWindow::~MainWindow()
{
delete ui;
}
//Eventos

void MainWindow::on_biniciar_clicked()
{
system ("sudo motion -n");
}
void MainWindow::on_bsalir_clicked()
{
system ("sudo service motion stop"); exit(0);
}
```

		PRÁCTICA
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS
LABORATORIO	AUTOMATIZACIÓN INDUSTRIAL	
CARRERA	INGENIERÍA ELECTRÓNICA SEDE GUAYAQUIL	
PRÁCTICA	#3	

1. TEMA

Simulación e implementación de un circuito con cerradura, sensores magnéticos y sensores de proximidad

2. OBJETIVO GENERAL

Realizar la simulación e implementación de un sistema de seguridad con cerradura, sensor PIR y sensores magnéticos Raspberry Pi B+ y Qt Creator

3. OBJETIVOS ESPECÍFICOS

- Realizar la respectiva programación en el software Qt Creator para la activación de la cerradura del sensor PIR y sensores magnéticos
- Diseñar una interfaz en el software Qt Creator para la simulación del arranque de la cerradura, sensor PIR y sensores magnéticos
- Utilizar la tarjeta Raspberry Pi B+ para la implementación del proceso de seguridad con cerradura, sensor PIR y sensores magnéticos

4. RECURSOS UTILIZADOS

Cerradura eléctrica, Sensores Magnéticos, Sensor PIR (Distancia Aprox. 50cm alcance máx. 3mts), Software QT Creator, Módulo de Rapsberry Pi B+, Laptop (Control remoto), Conductor macho y hembra para las conexiones

5. CONDICIONES

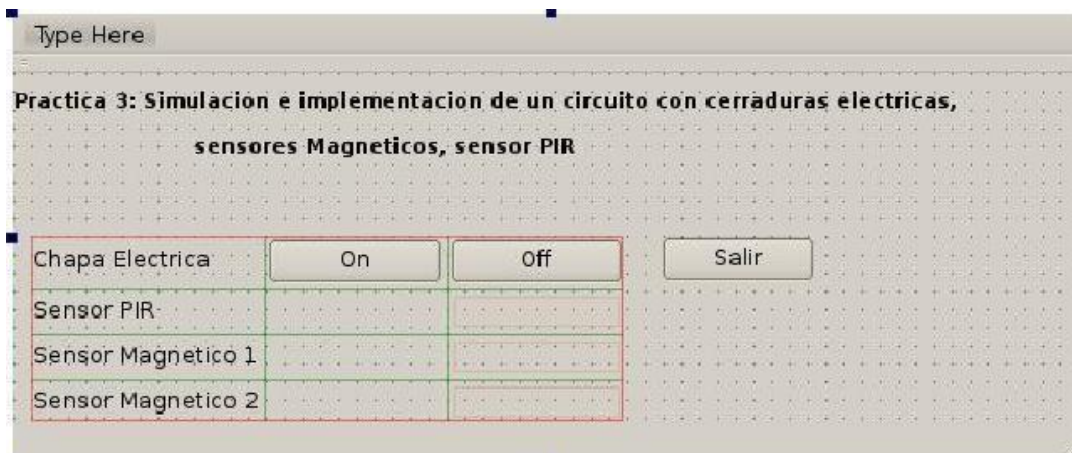
- El proceso para la práctica que simula la activación de una cerradura eléctrica, un sensor PIR y dos sensores magnéticos como se muestran en las Figura 40. Debe cumplir las siguientes condiciones:
- La activación y desactivación de los elementos será a través de Push Bottom
- Posteriormente el encendido y apagado debe darse de forma aleatoria para verificar que todos los elementos incluidos en la práctica trabajen correctamente bajo cualquier normativa.
- La finalización del programa se dará con el uso exclusivo de un Push Bottom con la leyenda “Salir” para finalizar completamente el proceso.

6. MARCO PROCEDIMENTAL

Una vez analizada cada condición del proceso para el desarrollo de la programación se toma en cuenta el uso de las GPIO usadas en el proyecto así como las respectivas alimentaciones de tensión

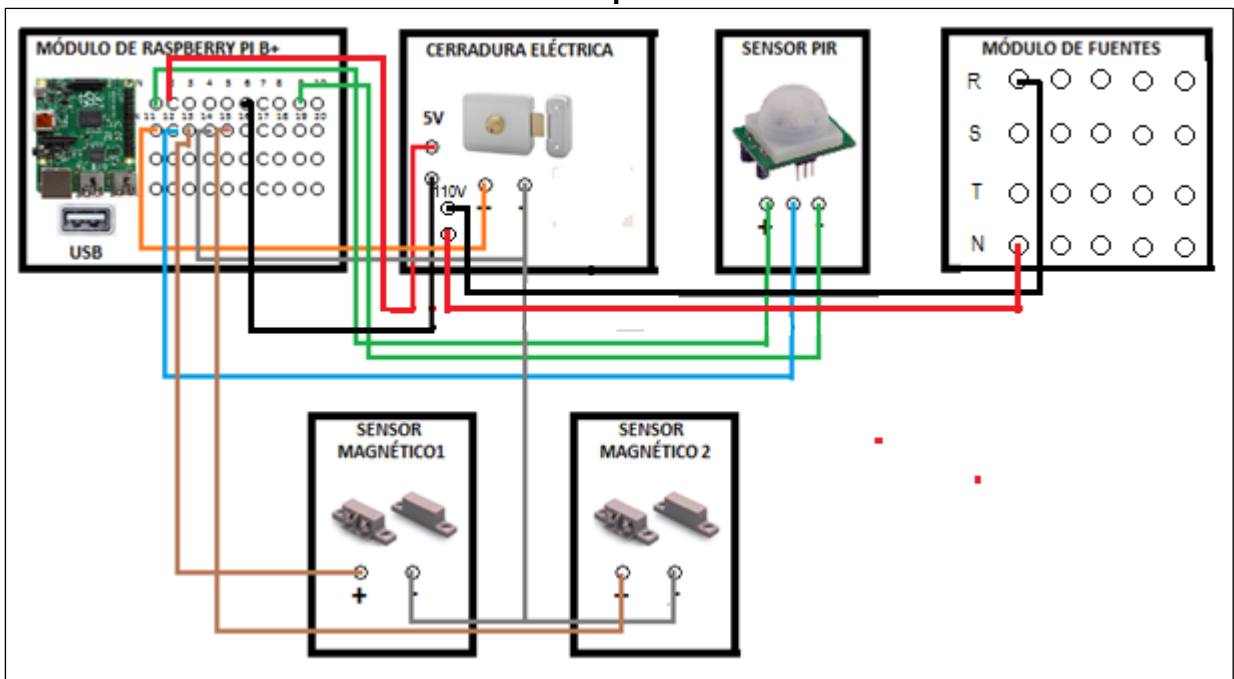
7. INTERFAZ DEL PROGRAMA

Figura 52. Interfaz del programa de la Practica 3



Fuente: Los Autores

Figura 53. Esquema de un circuito con cerradura, sensores magnéticos y sensores de proximidad



Fuente: Los Autores

Figura 54. Interfaz del programa en funcionamiento con los sensores activados



Fuente: Los Autores

Programación:

Programa3.pro //Se declaran los elementos que iran en el proyecto

```
#-----
```

```
# Project created by QtCreator 2015-11-29T19:57:16
```

```
#-----
```

```
QT += core gui greaterThan(QT_MAJOR_VERSION, 4): QT += widgets TARGET =
Programa3 TEMPLATE = app
SOURCES += main.cpp\mainwindow.cpp HEADERS += mainwindow.h
FORMS += mainwindow.ui LIBS +=-L/usr/local/lib -lwiringPi RESOURCES += \
imagenes.qrc
```

Carpeta Headers

mainwindow.h:

```
//Declarar librerias
```

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H
#include <QMainWindow>
#include "wiringPi.h"
#include <QTimer>
namespace Ui { class MainWindow;
}
class MainWindow : public QMainWindow
{
Q_OBJECT
```

```
public:
```

```
explicit MainWindow(QWidget *parent = 0);
```

```
~MainWindow();
```

```
private slots: //Declarar eventos void on_b1_clicked(); void on_b2_clicked();
```

```
void update_estado_boton(); void on_bsalir_clicked(); private:
```

```
Ui::MainWindow *ui;
```

```
};
```

```
#endif // MAINWINDOW_H
```

Carpeta Sources

main.cpp

```
#include <QApplication>
```

```
#include "mainwindow.h"
```

```
int main(int argc, char *argv[])
```

```
{
```

```
QApplication a(argc, argv); MainWindow w; w.show();
```

```
return a.exec();
```

```
}
```

Carpeta Sources

```
mainwindow.cpp
#include "mainwindow.h"
#include "ui_mainwindow.h"
//Definir variables


#define BOTON 0 //gpio 17 - pin 11
#define BUTON 1 //gpio18 - pin 12
#define BATON 2
MainWindow::MainWindow(QWidget *parent) : QMainWindow(parent),
ui(new Ui::MainWindow)
{
ui->setupUi(this); wiringPiSetup();

//Declarar variables
pinMode(5,OUTPUT); pinMode(BUTON,INPUT); pinMode(BOTON,INPUT);
pinMode(BATON,INPUT);
QTimer *timer= new QTimer(this);
connect (timer,SIGNAL (timeout()),this,SLOT(update_estado_boton())); timer-
>start(100);
}
MainWindow::~MainWindow()
{
delete ui;
}

//Eventos

void MainWindow::on_b1_clicked()
{
digitalWrite(5,HIGH);
}
void MainWindow::on_b2_clicked()
{
digitalWrite(5,LOW);
}
void MainWindow::update_estado_boton()
{
QPixmap myPixmap;
if (digitalRead(BOTON)==LOW)
{
myPixmap.load(QString::fromUtf8("./apagado.jpg")); ui->l1-
>setPixmap(myPixmap);
}
else
{
myPixmap.load(QString::fromUtf8("./encendido.jpg")); ui->l1-
>setPixmap(myPixmap);
}
if (digitalRead(BUTON)==LOW)
{
myPixmap.load(QString::fromUtf8("./apagado.jpg")); ui->l2-
>setPixmap(myPixmap);
}
}
```

```
else
{
myPixmap.load(QString::fromUtf8("./encendido.jpg")); ui->I2-
>setPixmap(myPixmap);
}
if (digitalRead(BATON)==LOW)
{
myPixmap.load(QString::fromUtf8("./apagado.jpg")); ui->I3-
>setPixmap(myPixmap);
}
else
{
myPixmap.load(QString::fromUtf8("./encendido.jpg")); ui->I3-
>setPixmap(myPixmap);
}
}
void MainWindow::on_bsalir_clicked()
{
digitalWrite(5,LOW); exit(0);
}
```

		PRÁCTICA #4	
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
		LABORATORIO	AUTOMATIZACIÓN INDUSTRIAL
CARRERA	INGENIERÍA ELECTRÓNICA SEDE GUAYAQUIL		
PRÁCTICA	#4		

1. TEMA

Simulación e implementación de un sistema de seguridad residencial

2. OBJETIVO GENERAL

Realizar la simulación e implementación de un sistema de seguridad con cerradura, sensor PIR, sensores magnéticos, sensores de proximidad, circuito cerrado de cámaras y alarma Raspberry Pi B+ y Qt Creator

3. OBJETIVOS ESPECÍFICOS

- Realizar la respectiva programación en el software Qt Creator para la activación de la cerradura eléctrica, sensor PIR, sensores magnéticos, circuito cerrado de cámaras y alarma
- Diseñar una interfaz en el software Qt Creator para la simulación de funcionamiento de cerradura eléctrica, sensor PIR, sensores magnéticos, circuito cerrado de cámaras y alarma
- Utilizar la tarjeta Raspberry Pi B+ para la implementación del proceso de seguridad con cerradura eléctrica, sensor PIR, sensores magnéticos, circuito cerrado de cámaras y alarma

4. RECURSOS UTILIZADOS

Lámina de banco de luces, Módulo de Raspberry Pi B+, Software QT Creator, Laptop, Conductor macho y hembra para las conexiones, Fuente de alimentación de 120VAC

5. CONDICIONES

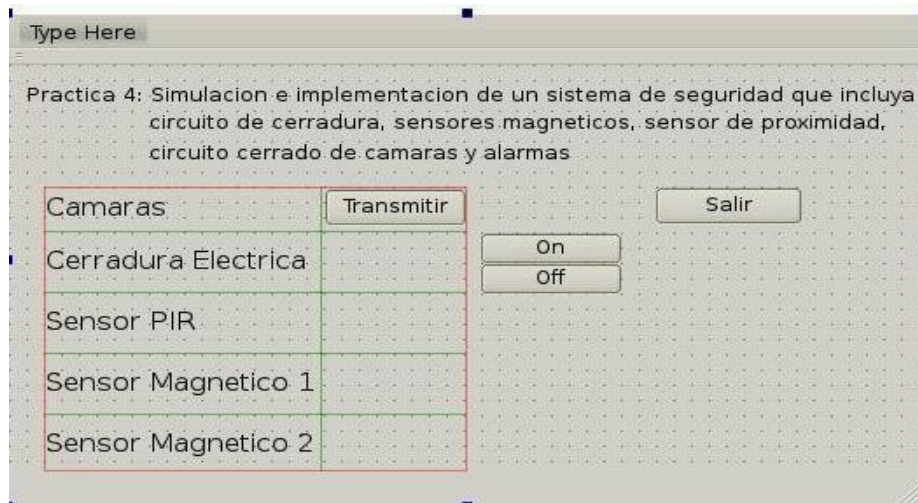
- El proceso para la práctica que simula la activación de una cerradura eléctrica, sensor PIR, sensores magnéticos, circuito cerrado de cámaras y alarma como se muestran en las Figura 42. Debe cumplir las siguientes condiciones:
- La activación y desactivación de los elementos será a través de Push Bottom
- Posteriormente la activación de la cerradura eléctrica se dará mediante un ingreso de clave a través de un teclado matricial y se mostrará en un LCD.
- Una vez que sea ingresada la clave correcta se activara el sistema correctamente, se tendrá un límite de tres opciones para la contraseña adecuada al cuarto ingreso erróneo el sistema se bloqueará automáticamente.
- El circuito cerrado de seguridad trabajará una vez que el Push Bottom de "Transferir" sea activado el tiempo de grabación y la cantidad de imágenes capturadas serán detalladas en el software MOTION.
- Los sensores magnéticos van a interactuar junto a la alarma una vez que cualquiera de los dos sensores se vea violentado en su seguridad, en ese instante se activará la alarma auditiva la cual permanecerá activada hasta que el usuario de la orden de finalizado.
- La finalización del programa se dara con el uso exclusivo de un Push Bottom con la leyenda "Salir" para finalizar completamente el proceso.

6. MARCO PROCEDIMENTAL

Una vez analizada cada condición del proceso para el desarrollo de la programación se toma en cuenta el uso de las GPIO usadas en el proyecto así como las respectivas alimentaciones de tensión.

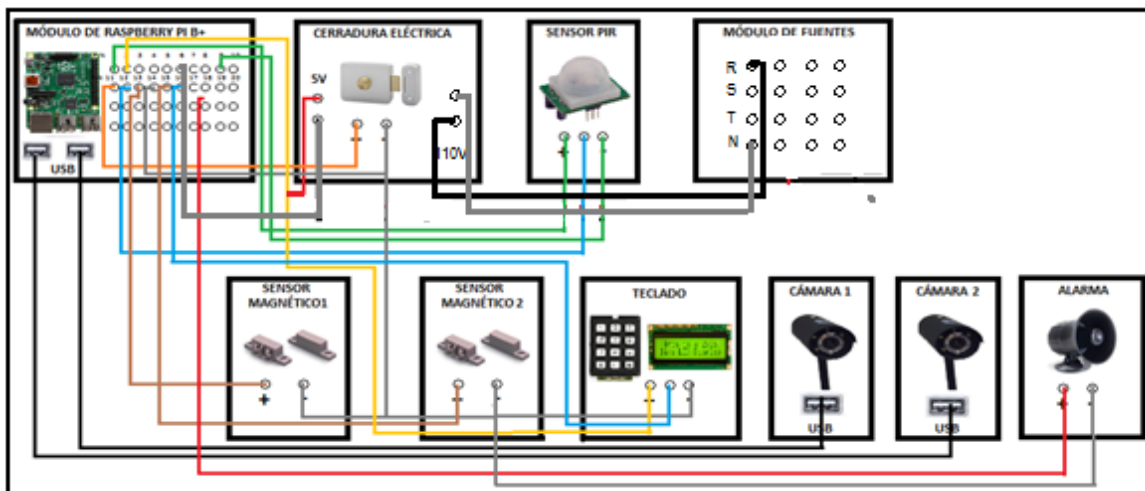
7. INTERFAZ DEL PROGRAMA

Figura 55. Interfaz del programa de la Practica 4



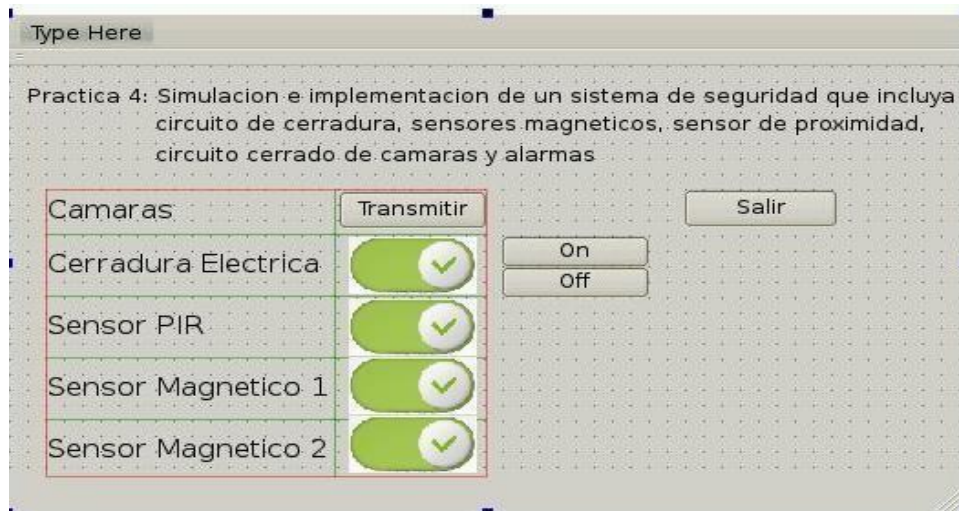
Fuente: Los Autores

Figura 56. Esquema de un Sistema de seguridad residencial



Fuente: Los Autores

Figura 57. Interfaz de la Practica 4 funcionando con los sensores activados



Fuente: Los Autores

Programación:

Programa4.pro //Se declaran los elementos que iran en el proyecto

```
#-----  
# Project created by QtCreator 2015-12-04T00:58:36  
#-----
```

```
QT += core gui greaterThan(QT_MAJOR_VERSION, 4): QT += widgets TARGET =  
Programa4  
TEMPLATE = app  
SOURCES += main.cpp\ mainwindow.cpp HEADERS += mainwindow.h  
FORMS += mainwindow.ui LIBS +=-L/usr/local/lib -lwiringPi RESOURCES += \  
imagenes.qrc
```

Carpeta Headers

mainwindow.h

//Declarar librerias

```
#ifndef MAINWINDOW_H  
#define MAINWINDOW_H  
#include <QMainWindow>  
#include "wiringPi.h"  
#include <QTimer>  
namespace Ui { class MainWindow;  
}  
class MainWindow : public QMainWindow  
{
```



```

Q_OBJECT
public:
explicit MainWindow(QWidget *parent = 0);
~MainWindow();
private slots: //Declarar eventos void lecturadatos(); void on_pushButton_clicked(); void
on_b1_clicked(); void on_bon_clicked(); void on_boff_clicked();
private:
Ui::MainWindow *ui;
};
#endif // MAINWINDOW_H

```

Carpeta Sources

```

main.cpp
#include <QApplication>
#include "mainwindow.h"
int main(int argc, char *argv[])
{
QApplication a(argc, argv); MainWindow w; w.show();
return a.exec();
}

```

Carpeta Sources

```

mainwindow.cpp
#include "mainwindow.h"
#include "ui_mainwindow.h"
//Definir variables
#define CE 0
#define SPIR 1
#define SMU 2
#define SMD 3

MainWindow::MainWindow(QWidget *parent) : QMainWindow(parent), ui(new
Ui::MainWindow)
{

ui->setupUi(this); wiringPiSetup();

//Declarar variables
pinMode(5,OUTPUT); pinMode(6,OUTPUT); pinMode(CE,INPUT);
pinMode(SPIR,INPUT); pinMode(SMU,INPUT); pinMode(SMD,INPUT);

QTimer *timer= new QTimer(this);
connect(timer,SIGNAL (timeout()),this,SLOT(lecturadatos())); timer->start(1);
}
MainWindow::~MainWindow()
{
delete ui;
}
//Eventos
void MainWindow::lecturadatos()
{
QPixmap myPixmap;

```


```

if (digitalRead(SPIR)==LOW)
{
myPixmap.load(QString::fromUtf8("./apagado.jpg")); ui->l2-
>setPixmap(myPixmap);
}
else
{
myPixmap.load(QString::fromUtf8("./encendido.jpg")); ui->l2-
>setPixmap(myPixmap);
}
if (digitalRead(SMU)==LOW)
{
myPixmap.load(QString::fromUtf8("./apagado.jpg")); ui->l3-
>setPixmap(myPixmap); digitalWrite(6,LOW);
}
else
{
myPixmap.load(QString::fromUtf8("./encendido.jpg")); ui->l3-
>setPixmap(myPixmap); digitalWrite(6,HIGH);
}
if (digitalRead(SMD)==LOW)
{
myPixmap.load(QString::fromUtf8("./apagado.jpg")); ui->l4-
>setPixmap(myPixmap); digitalWrite(6,LOW);
}
else
{
myPixmap.load(QString::fromUtf8("./encendido.jpg")); ui->l4-
>setPixmap(myPixmap);
digitalWrite(6,HIGH);
}
}

void MainWindow::on_pushButton_clicked()

{
system ("sudo service motion stop"); digitalWrite(5,LOW); digitalWrite(6,LOW); exit(0);
}
void MainWindow::on_b1_clicked()
{
system ("sudo motion");
}
void MainWindow::on_bon_clicked()
{
myPixmap.load(QString::fromUtf8("./encendido.jpg")); ui->l1->setPixmap(myPixmap);
digitalWrite(5,HIGH);
}
void MainWindow::on_boff_clicked()
{
myPixmap.load(QString::fromUtf8("./apagado.jpg")); ui->l1->setPixmap(myPixmap);
digitalWrite(5,LOW);
}

```

		PRÁCTICA #5	
		MANUAL DE PROCEDIMIENTOS DE PRÁCTICAS	
LABORATORIO	AUTOMATIZACIÓN INDUSTRIAL		
CARRERA	INGENIERÍA ELECTRÓNICA SEDE GUAYAQUIL		
PRÁCTICA	#5		

1. TEMA

Simulación e implementación de un sistema industrial con una válvula selenoide, sensor de temperatura y un ventilador que simula un motor DC

2. OBJETIVO GENERAL

Realizar la simulación e implementación de un sistema industrial con una válvula selenoide, sensor de temperatura y un ventilador que simula un motor DC con Raspberry Pi B+ y Qt Creator

3. OBJETIVOS ESPECÍFICOS

- Realizar la respectiva programación en el software Qt Creator para un sistema industrial
- Diseñar una interfaz en el software Qt Creator para la simulación y funcionamiento de un circuito industrial
- Utilizar la tarjeta Raspberry Pi B+ para la implementación del proceso industrial
- Aprender a enviar datos analógicos a la Raspberry Pi B+, convirtiendo este valor a digital a través de un PIC 18F4550

4. RECURSOS UTILIZADOS

Lámina Sensores de temperatura LM 35 (35°C), Válvula solenoide, Ventilador 120VAC, Software QT Creator, Módulo de Rapsberry Pi B+, Laptop (Control remoto), Conductor macho y hembra para las conexiones

5. CONDICIONES

- El proceso para la práctica que simula un proceso industrial como se muestran en las Figura 44. Debe cumplir las siguientes condiciones:
- Se llevará a cabo un simulación de proceso industrial el cual constará de tres “tanques de pintura” dados por dos niveles (Bajo y Alto)
- Para dar paso al primer llenado se activará el Tanque 1 con Push Bottom con la Leyenda “Tanque 1” Una vez que el Tanque 1 finalice su llenado se dará paso al llenado del Tanque 2 el cual iniciará cuando se pulse un Push Bottom con la leyenda “Tanque 2.”
- Una vez que Tanque 1 y Tanque 2 se encuentren llenos se da paso a la etapa de “vaciado” en la cual ambos tanques depositan pintura en un recipiente llamado Tanque 3
- Cuando Tanque 3 se encuentre lleno se activará el Ventilador que simula un Motor DC
- Una vez que el Ventilador (Motor DC) este activado entra en función el sensor de temperatura LM 35 se dará un valor de set de 35°C para garantizar una mezcla óptima, una vez que se llegue al valor seteado el ventilador dejara de estar activado.
- Cuando se tenga la mezcla lista se procederá al último paso el cual consta de la activación de una válvula solenoide la cual permitirá el paso de la mezcla final
- La finalización del programa se dará con el uso exclusivo de un Push Bottom con la leyenda “Salir” para finalizar completamente el proceso.

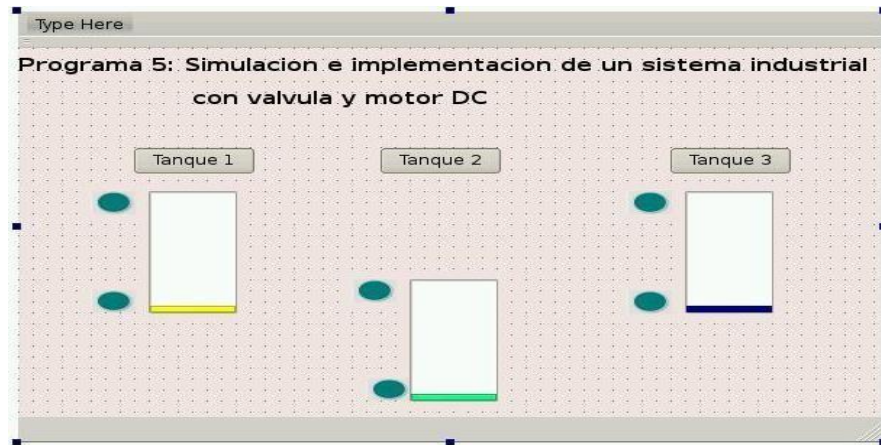
6. MARCO PROCEDIMENTAL

Una vez analizada cada condición del proceso para el desarrollo de la programación se toma en cuenta el uso de las GPIO usadas en el proyecto así como las respectivas alimentaciones de tensión

7. INTERFAZ DEL PROGRAMA

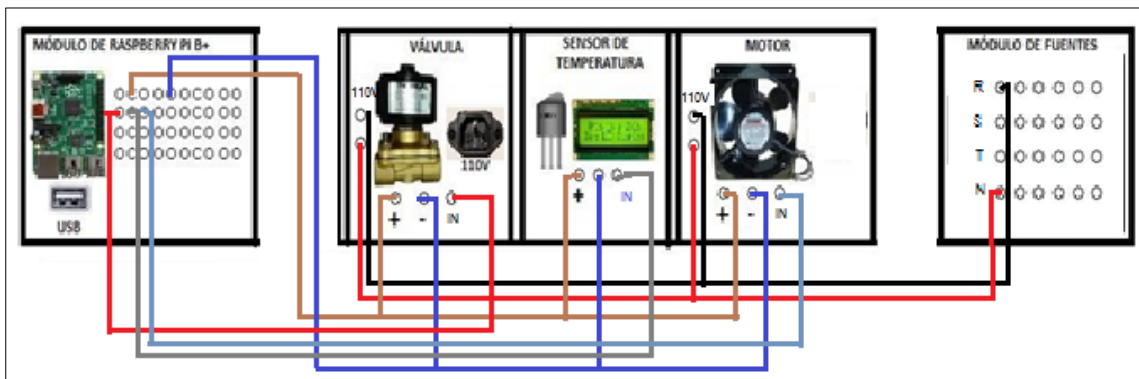
Interfaz del programa:

Figura 58. Interfaz del programa de la Practica 5



Fuente: Los Autores

Figura 59. Esquema de un sistema industrial con válvulas, motor DC



Fuente: Los Autores

Programación:

```
Carpeta Sources mainwindow.cpp
//Declarar librerias
```

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#define ST 0
```

```
MainWindow::MainWindow(QWidget *parent) : QMainWindow(parent),
ui(new Ui::MainWindow)
```

```
{
ui->setupUi(this); wiringPiSetup();
//Declarar variables pinMode(5,OUTPUT); pinMode(6,OUTPUT); pinMode(ST,INPUT);
```

```

QTimer *timer= new QTimer(this);
connect(timer,SIGNAL (timeout()),this,SLOT(lecturadatos())); timer->start(1);
}
MainWindow::~MainWindow()
{
delete ui;
}

```

//Eventos

```

void MainWindow::on_bt1_clicked()
{
int pos=0;
QPixmap (myPixmap); do
{
if (pos<21)
{
myPixmap.load(QString::fromUtf8(":/imagen2.png")); ui->ll1-
>setPixmap(myPixmap);
}
else
{
myPixmap.load(QString::fromUtf8(":/imagen1.png")); ui->ll1-
>setPixmap(myPixmap);
}
if (pos>79)
{
myPixmap.load(QString::fromUtf8(":/imagen2.png")); ui->lh1-
>setPixmap(myPixmap);
}
else
{
myPixmap.load(QString::fromUtf8(":/imagen1.png")); ui->lh1-
>setPixmap(myPixmap);
}
::usleep(50000);
ui->pb1->setValue(pos); pos=pos+1;
} while (pos<101);
}

```

```

void MainWindow::on_bt2_clicked()
{
int pos=0;
QPixmap (myPixmap); do
{
if (pos<21)
{
myPixmap.load(QString::fromUtf8(":/imagen2.png")); ui->ll2-
>setPixmap(myPixmap);
}
else
{
myPixmap.load(QString::fromUtf8(":/imagen1.png")); ui->ll2-
>setPixmap(myPixmap);
}
}
}

```

```

}
if (pos>79)
{
myPixmap.load(QString::fromUtf8(":/imagen2.png")); ui->lh2-
>setPixmap(myPixmap);
}
else
{

myPixmap.load(QString::fromUtf8(":/imagen1.png")); ui->lh2-
>setPixmap(myPixmap);
}
::usleep(50000);
ui->pb2->setValue(pos); pos=pos+1;
} while (pos<101);
}
void MainWindow::on_bt3_clicked()
{
int p1=101; int p2=101; int p3=0; int p4=101; QPixmap (myPixmap); do
{
if (p3<21)
{
myPixmap.load(QString::fromUtf8(":/imagen2.png")); ui->ll3-
>setPixmap(myPixmap);
}
else
{
myPixmap.load(QString::fromUtf8(":/imagen1.png")); ui->ll3-
>setPixmap(myPixmap);
}
if (p3>79)
{
myPixmap.load(QString::fromUtf8(":/imagen2.png")); ui->lh3-
>setPixmap(myPixmap);
}
else{
if (p2<21)
{
myPixmap.load(QString::fromUtf8(":/imagen2.png")); ui->ll2-
>setPixmap(myPixmap);
}
else
{
myPixmap.load(QString::fromUtf8(":/imagen1.png")); ui->ll2-
>setPixmap(myPixmap);
}
}
if (p2>79)
{
myPixmap.load(QString::fromUtf8(":/imagen2.png")); ui->lh2-
>setPixmap(myPixmap);
}
else
{
myPixmap.load(QString::fromUtf8(":/imagen1.png")); ui->lh2-
>setPixmap(myPixmap);
}
}
}

```

```

}
if (p1<21)
{
myPixmap.load(QString::fromUtf8(":/imagen2.png")); ui->ll1-
>setPixmap(myPixmap);
}
else
{

myPixmap.load(QString::fromUtf8(":/imagen1.png")); ui->ll1-
>setPixmap(myPixmap);
}
if (p1>79)
{
myPixmap.load(QString::fromUtf8(":/imagen2.png")); ui->lh1-
>setPixmap(myPixmap);
}
else
{
myPixmap.load(QString::fromUtf8(":/imagen1.png")); ui->lh1-
>setPixmap(myPixmap);
}
::usleep(50000);
ui->pb3->setValue(p3); ui->pb2->setValue(p2); ui->pb1->setValue(p1); p1=p1-1;
p2=p2-1; p3=p3+1;
}
while (p3<101);
if (digitalRead(ST)==HIGH) digitalWrite(5,HIGH); digitalWrite(6,HIGH); do
{
if (p4<21)
{
myPixmap.load(QString::fromUtf8(":/imagen2.png")); ui->ll3-
>setPixmap(myPixmap);
}
else
{
myPixmap.load(QString::fromUtf8(":/imagen1.png"));
ui->ll3->setPixmap(myPixmap);
}
if (p4>79)
{
myPixmap.load(QString::fromUtf8(":/imagen2.png")); ui->lh3-
>setPixmap(myPixmap);
}
else
{
myPixmap.load(QString::fromUtf8(":/imagen1.png")); ui->lh3-
>setPixmap(myPixmap);
}
::usleep(50000);
ui->pb3->setValue(p4); p4=p4-1;
} while (p4<1); digitalWrite(5,LOW); digitalWrite(6,LOW);
}

```


7. Conclusiones

Una vez culminado este proyecto de titulación podemos decir con gran satisfacción que se ha cumplido los puntos propuestos y se ha dado paso a conocer un nuevo sistema que si bien es cierto no es difícil de aprender tiene su rango de complicidad.

Los conocimientos impartidos en el lapso académico son de gran importancia en vista de que la base de programar en Qt Creator es muy similar a programar en Java medio que se usa en las instalaciones de la UNIVERSIDAD POLITÉCNICA SALESIANA.

Lo complicado fue la instalación del software RAPBIAN ya que debe tener cierta compatibilidad con la tarjeta Raspberry Pi B+, pero una vez solucionado este inconveniente solo resto conocer como programar en Qt Creator lo cual fue logrado en su totalidad.

El uso de las interfaces entre la Raspberry Pi B+ que trabaja con poca corriente y un voltaje de máximo 5V a elementos como cerradura eléctricas, luces piloto, válvulas y demás fue exitoso, y abre las posibilidades de uso de la Raspberry Pi B+ para cualquier otro tipo de elementos.

8. Recomendaciones

Se debe tener en cuenta que la Raspberry Pi B+ tiene rangos técnicos específicos los cuales deben ser respetados y seguidos rigurosamente, por ejemplo las salidas máximas de voltaje y corriente.

Tener en consideración con las cargas grandes de 110V las cuales deben ser reguladas a máximo 3.3V para que trabajen correctamente.

Para las tarjetas de Módulo de Interfaz que se muestran en el banco de focos se debe tener en cuenta que deben tener optoacopladores para que al instante de switcheo los relés no se queden enclavados y puedan liberarse con facilidad, ya que de no tenerlos se corre el riesgo de que se activen en momentos en los que no es deseado.

Cuando se use las tarjetas de interfaz hay que tener en cuenta de que si se usan muchos elementos de 110V o elementos que requieran una gran cantidad de corriente puede presentar problemas en la Raspberry Pi B+ ya que esta no cuenta con una gran cantidad de corriente.

9. Referencias Bibliográficas

- Curiosoando. (2011). Curiosoando. Obtenido de <https://curiosoando.com/que-es-una-valvula-solenoid>
- Directlink. (2015). Tienda Web. Obtenido de http://www.directlink.cl/catalog/product_info.php/cPath/243_247/products_id/6187
- Domotica. (s.f.). Recuperado el 4 de Marzo de 2016, de http://www.profesormolina.com.ar/tecnologia/domotica/que_esdomo.htm
- EcuRed. (9 de Marzo de 2016). Qt Creator. Obtenido de http://www.ecured.cu/Qt_Creator
- Electroensaimada. (2014). Escritorio remoto Raspberry Pi. Obtenido de <http://www.electroensaimada.com/escritorio-remoto.html>
- Electronics, T. (s.f.). Obtenido de <http://www.tomsonelectronics.com/products.php>
- Foundation, R. P. (2011 - 2016). Hardware Raspberry Pi. Obtenido de <http://www.raspberrypi.org/hardware-raspberry-pi.php>
- FUNDACIÓN, F. P. (2011). Power Supply - Raspberry Pi Documentation. Obtenido de <https://www.raspberrypi.org/documentation/hardware/raspberrypi/power/README.md>
- Glatelier. (15 de Mayo de 2009). Qt Creator, desarrollando aplicaciones rápidamente. Obtenido de <http://glatelier.org/2009/05/15/qt-creator-desarrollando-aplicaciones- rapidamente/>
- Harok. (2015). Luces Piloto. Obtenido de Induelectro: <http://www.induelectro.cl/PDF/harok.pdf>
- Marcelo, J. I. (2006). Lcd Alfa. Obtenido de http://eii.unex.es/profesores/jisuarez/descargas/ip/lcd_alfa.pdf
- Microcontroladores. (2009). Obtenido de http://galia.fc.uaslp.mx/~cantocar/microcontroladores/SLIDES_8051_PDF/20_M A TRI.PDF
- Mileti. (12 de Febrero de 2012). Obtenido de <http://www.mileti.com.ar/hoy/?p=413>
- Motion. (2016). Motion. Obtenido de

http://www.lavrsen.dk/foswiki/bin/view/Motion/MotionGuide3x1x20#Un_45install

Nergiza. (04 de 05 de 2012). Raspberry Pi 2 model B. Obtenido de <http://nergiza.com/raspberry-pi-2-b-que-es-y-para-que-nos-puede-servir/>

Nextiafenix. (3012). PIC. Obtenido de [http://www.nextiafenix.com/producto/pic18f4550/PE, I. \(12 de agosto de 2014\). Comparativa y análisis: Raspberry Pi vs competencia.](http://www.nextiafenix.com/producto/pic18f4550/PE,%20I.%20(12%20de%20agosto%20de%202014).Comparativa%20y%20an%C3%A1lisis:%20Raspberry%20Pi%20vs%20competencia.)

Obtenido de <http://comohacer.eu/comparativa-y-analisis-raspberry-pi-vs-competencia/>

Raspberrypi. (05 de 2015). RaspberryPi. Obtenido de <https://www.raspberrypi.org/forums/viewtopic.php?f=76&t=128492>

Rpiplus. (7 de Junio de 2013). LIBRERIA WIRING PI. Obtenido de <http://rpiplus.blogspot.com/2013/06/libreria-wiring-pi.html>

Tvc. (2000). Obtenido de http://tvc.mx/shop/catalog/product_info.php?products_id=2305 Yalelatinoamerica. (2015). Electricas. Obtenido de

<http://www.yalelatinoamerica.com/Yale/yalelatinoamericaCOM/Home/Electricas.pdf>

10. Anexos

10.1 Equipos para Programación

Figura 60. Lámina con Motor a 110v



Fuente: Los Autores

Figura 61. Lámina con Cerradura Eléctrica a 110V



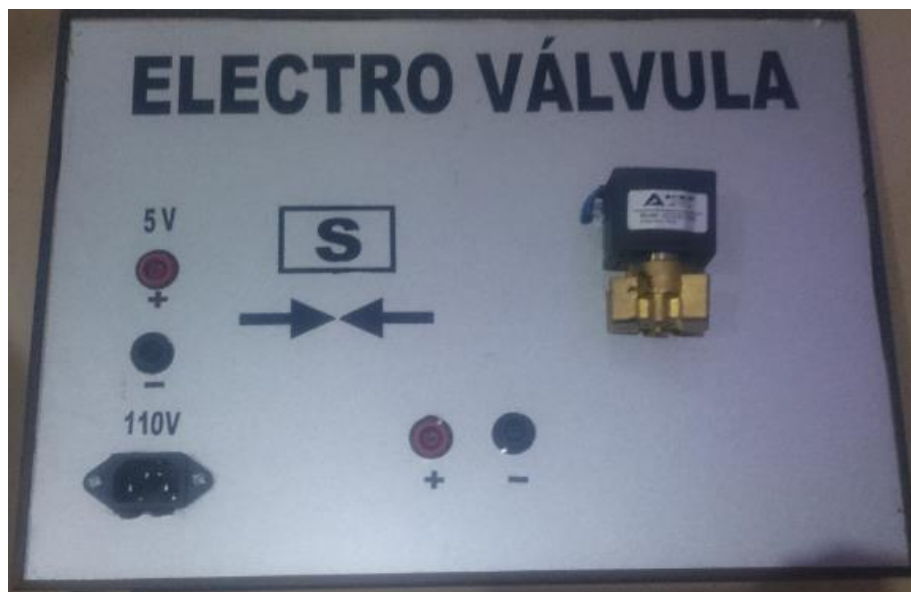
Fuente: Los Autores

Figura 62. Lámina de sensor magnético 1



Fuente: Los Autores

Figura 63. Lámina de la Electroválvula



Fuente: Los Autores

Figura 64. Lámina de Banco de Luces



Fuente: Los Autores

Figura 65. Lámina de Sensor PIR



Fuente: Los Autores

Figura 66. Lámina de Ingreso de Clave



Fuente: Los Autores

Figura 67. Lámina con cámara de seguridad 2



Fuente: Los Autores