

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE TECNOLOGÍA
CARRERA: ELECTRÓNICA Y TELECOMUNICACIONES



NIVEL LICENCIATURA
EXAMEN DE GRADO
(TRABAJO DE APLICACIÓN)

**“SISTEMA DE CONTROL DOMÓTICO
BASADO EN LA PLATAFORMA ARDUINO VÍA
BLUETOOTH”**

POSTULANTE: Univ. Edgar Antonio Mamani

La Paz – Bolivia
2016

AGRADECIMIENTO

En primer lugar, quiero agradecer a mi mamá por el apoyo para que pudiera lograr mis sueños, por motivarme y el esfuerzo que hizo para que estudie, agradezco también a todos los docentes, que me brindaron sus conocimientos y como personas su amistad sincera, que me han formado profesionalmente durante toda mi vida universitaria, mil gracias porque de alguna manera forman parte de lo que ahora soy.

DEDICATORIA

Este trabajo lo dedico a mi querida madre María Lidia Mamani Mamani, ella fue la que me impulso para seguir adelante, y continuar creciendo como persona tanto en lo familiar como en lo profesional.

ÍNDICE

ÍNDICE	i
FIGURAS	ii
TABLAS	iii
FOTOGRAFÍAS	iii
RESUMEN	iv
CAPITULO I	1
ANTECEDENTES DEL PROYECTO	1
1.1.- Planteamiento del problema	2
1.1.1.- Formulación del problema.....	2
1.2.- Objetivos del proyecto	3
1.3.- Justificación	4
1.4.- Delimitaciones del proyecto	5
1.5.- Metodología de investigación	5
CAPITULO II	6
MARCO TEORICO CONCEPTUAL	6
2.1.- Definición de domótica	6
2.2.- Características generales	7
2.2.1.- Programación y ahorro de energía.....	7
2.2.2.- Confort.....	7
2.2.3.- Seguridad.....	8
2.2.4.- Comunicaciones.....	8
2.2.5.- Accesibilidad.....	8
2.3.- ¿Qué es ARDUINO?	9
2.4.- ¿Por qué ARDUINO?	9
2.5.- Dispositivos y sensores	11
2.5.1.- Arduino Mega 2560.....	11
2.5.2.- Sensor HC-SR501 PIR de movimiento.....	15
2.5.3.- Módulo comunicación bluetooth HC-05.....	19
2.5.4.- Módulo de 4 relés para Arduino.....	23
2.5.5.- Sensor de temperatura y humedad DHT 11.....	26
2.5.6.- Sensor magnético reed-switch.....	28
2.5.7.- Aplicación Electrónica Bluetooth.....	29
CAPITULO III	33

INGENIERIA DEL PROYECTO	33
.....	33
3.1.- Actividades a realizar	35
3.1.1.- Modulo bluetooth	36
3.1.1.1.- Código para el módulo bluetooth	37
3.1.2.- Sensor de temperatura DHT11.....	38
3.1.2.1.- Código para el sensor DHT11	39
3.1.3.- Sensor PIR HC-SR501	40
3.1.3.1.- Código para el sensor HC-SR501	41
3.2.- Circuito final.....	41
3.3.- Ventajas y desventajas	43
3.4.- Instalación del proyecto.....	44
3.5.-Criterio de implementación.....	48
3.6.- Código de programa final	48
3.4.- Costo total del proyecto.....	54
CONCLUSIONES	56
SUGERENCIAS.....	56
BIBLIOGRAFIA	57

FIGURAS

Figura 1 - Arduino Mega 2560	11
Figura 2 - Características del arduino ATmegaega 2560	12
Figura 3 - Sensor HC-SR501.....	16
Figura 4 - Sensor infrarrojo.....	16
Figura 5 - Forma de trabajo del sensor PIR	17
Figura 6 - Lente de fresnel.....	18
Figura 7- Condensación de luz	18
Figura 8 - Característica del sensor PIR	19
Figura 9 - Diagrama maestro esclavo.....	20
Figura 10 - Modulo bluetooth HC-06 esclavo y HC05 maestro esclavo	21
Figura 11 - Partes del módulo	24
Figura 12 - Circuito esquemático.....	25
Figura 13 - Sensor DHT11	27
Figura 14 - Reed-switch	28
Figura 15 - Aplicación bluetooth electronics.....	29

Figura 16 - Menú principal.....	31
Figura 17 - Editar pantalla	31
Figura 18 - Ejemplo de gestión de pantalla.....	32
Figura 19 - Diagrama en bloques de control por arduino.....	33
Figura 20 - Diagrama de conexión del Bluetooth HC-05.....	36
Figura 21 - Diagrama de conexión sensor de temperatura DHT11	38
Figura 22 - Diagrama de conexión del sensor PIR.....	40
Figura 23 - Circuito final a utilizar.....	42

TABLAS

Tabla 1 - Costo de componentes	55
---	-----------

FOTOGRAFIAS

Fotografía 1 - Diseño de botones de control	44
Fotografía 2 - Primeras pruebas	45
Fotografía 3 - Cajas de cartón.....	45
Fotografía 4 - Cartón trabajado.....	46
Fotografía 5 - Forado del cartón	46
Fotografía 6 - Implementación final en la maqueta	47
Fotografía 7 - Vista interior de la maqueta.....	47

RESUMEN

Este proyecto trata sobre el diseño e implementación un sistema de seguridad y domótico para el hogar u otros establecimientos.

El sistema consta de un detector de intrusos, que detecta la presencia por diferentes métodos, como pueden ser el movimiento, la vibración de ventanas y puertas producida. El usuario también puede acceder a los datos de todos los sensores instalados.

Para controlar todo el sistema se dispone de una interfaz en bluetooth, que puede ser accedida desde un teléfono con dicho sistema. Desde la interfaz el usuario puede activar o desactivar la alarma, activar la simulación de presencia, entre otras cosas. Además se puede configurar los medios por los que comunicarse en caso de que se produzca una alarma.

En el caso de producirse una alerta, se activa una señal acústica y se genera una alerta en la aplicación Android.

Para que todo el sistema funcione, se ha diseñado e implementado una arquitectura, hardware, software y de comunicaciones, desde la programación de los sensores, y hasta la interfaz. Para la elección de tecnologías se han utilizado de hardware y software libre en la medida de lo posible.

CAPITULO I

ANTECEDENTES DEL PROYECTO

La idea para realizar este proyecto, surge con la manera de controlar de manera manual y automáticamente un hogar Para responder a esta pregunta, se investigan diferentes métodos para obtener datos de sensores, y repetidamente aparece el nombre de Arduino, y el concepto de hardware libre.

De forma resumida, lo que permite hacer esta plataforma, es hacer de interfaz entre elementos electrónicos y un equipo, ya sea Windows, Linux o Mac, facilitando enormemente la interacción con elementos como sensores. Aunque Arduino es hardware libre, esta plataforma también puede interaccionar con sensores o actuadores que no sean de hardware libre y no conozcamos sus diseños internos, ya que de los elementos utilizados nos interesa como comunicarnos con ellos, no como están hechos.

1.1.- Planteamiento del problema

En la actualidad la dinámica productiva enfrenta a las personas e instituciones al desafío de la eficiencia, la velocidad y la ubicuidad; en el mundo moderno cualquier actividad que no pueda ser realizada en forma eficiente mediante herramientas remotas representa una enorme desventaja, tanto competitivamente como de calidad de vida. Dentro de esta perspectiva, la automatización de los espacios y su dinamización con la incorporación de herramientas y tecnologías de la información y las telecomunicaciones, es una necesidad inminente. Tareas tan simples como controlar el acceso en entornos con elevados requerimientos de seguridad en hogares se convierte en la vertiginosa dinámica global en un problema, para aquellas instituciones y personas.

Son múltiples los factores que pueden caracterizar las formas de vida de las personas hoy en día, como ejemplo, adultos mayores que viven solos, población con algún tipo de discapacidad y que viven de forma independiente, incluso los mismos avatares de la modernidad que requieren todo con prontitud y ahorro del tiempo que no les permite siquiera descender del vehículo.

Es así como, labores tan comunes como abrir o cerrar una puerta y la inseguridad se convierten en un problema; pero con la implementación de la domótica en el hogar se disminuye el grado de dificultad para realizar estas tareas además de dar una seguridad. En el mercado actual se cuenta con un gran número de aplicaciones de muy alto nivel sobre el tema, las cuales en su mayoría tienen costos elevados y están pensadas más en generar una experiencia de lujo al cliente que en facilitar el desarrollo de una actividad específica a las personas.

1.1.1.- Formulación del problema

¿Cómo ayuda un sistema domótico en la vida de las personas y en la seguridad de sus hogares o recintos, bajo el esquema de integración tecnológica?

1.2.- Objetivos del proyecto

A continuación se exponen cuáles son los objetivos principales para realizar el proyecto:

General

Diseñar e implementar un sistema de control domótico basado en una plataforma de arduino controlado por un móvil a través de bluetooth mediante una aplicación android, que facilitara el control de un recinto.

Específicos

- Implementar un sistema domótico y la interfaz de hardware, sensores y actuadores requerido para la plataforma Arduino.
- Diseñar e implementar un sistema domótico, para el control remoto de la apertura y cierre de puertas.
- Diseñar controles en una aplicación android, para el control vía bluetooth.
- Controlar la seguridad del recinto.
- Controlar a través de un dispositivo móvil los relés actuadores.

1.3.- Justificación

La plataforma libre de arduino posee un potencial, que permite brindar soluciones tecnológicas eficientes, así que para la aplicación específica es la seguridad en el hogar o recintos, permitiendo al usuario realizar una acción sobre el evento que se produzca.

El entorno de desarrollo de arduino por su parte, permite crear una lógica sencilla sobre los sensores y actuadores, y no solo recibe instrucciones, si no que interactúa con el usuario.

Tecnológica

En Bolivia no hay mucho de estas innovaciones de estos sistemas integrados a sistemas domótico controlados con Arduino, que se presenten formalmente como desarrollo practico.

La implementación del sistema domótico para la seguridad es un punto de partida para el incentivo del desarrollo domótico controlado desde un dispositivo móvil, y otras soluciones que puedan generar eficiencia combinando las telecomunicaciones y los sistemas de control, desde la plataforma android.

Social

Los hogares que sean implementados con el sistema domótico con android, tendrán la oportunidad de monitorear de mejor manera su hogar. Teniendo acceso es esta tecnología por medio de un software donde recibe información del estado de la puerta del lugar.

Académica

El sistema tiene como aplicación en particular dos áreas de la electrónica y telecomunicaciones porque así lo requiere el mismo para la función que debe desempeñar.

Económica

Los sistemas de seguridad eléctrica en general son de altos costos, y el monitoreo de alarmas es un servicio que los usuarios pagan mensualmente. Con la implementación del sistema domótico con arduino estos costos serán deducidos en gran medida, debido a que se desarrollan sobre una plataforma libre y el usuario final quien mantiene el monitoreo sobre su propio sistema.

1.4.- Delimitaciones del proyecto

Temporal

Debido a corto tiempo que se dispone, el proyecto se estima terminar en aproximadamente tres semanas.

Espacial

El lugar donde será implementado el sistema será en la empresa DATAN BOLIVIA y que busca mejorar la condición de vida de las personas será la ciudad de La Paz principalmente el área rural.

Temática

Este proyecto será en cuanto a domótica, en base a un arduino y sensores para el respectivo control.

1.5.- Metodología de investigación

Para el desarrollo de este proyecto se ha utilizado el método de investigación aplicada, ya que se enfoca en la aplicación de los conocimientos para la producción de tecnología al servicio de la población.

El método posteriormente permite aplicar algunos de los conocimientos adquiridos para implementar un sistema domótico que permita controlar remotamente desde un dispositivo móvil con un dispositivo bluetooth.

CAPITULO II

MARCO TEORICO CONCEPTUAL

2.1.- Definición de domótica

La palabra domótica proviene de la unión de dos palabras: *domus* (casa en latín) y *tica* (de automática, palabra en griego, “que funciona por si sola”), entonces el significado de domótica es: casa robotizada. Se entiende por lo tanto como domótica el conjunto de los sistemas capaces de automatizar y controlar una vivienda, aportando servicios como la gestión energética, seguridad, bienestar y comunicación.

Algunos de los elementos que se pueden controlar en una casa domótica son los siguientes: la iluminación, la climatización, puertas, ventanas, persianas y toldos, electrodomésticos o el suministro de agua, gas y electricidad. En general, puede llegar a controlarse todo aquello que se proponga, nadie ha puesto un límite aún a la domótica.

En un sistema domótico se pueden encontrar elementos de los siguientes tipos:

- **Controladores:** dispositivos encargados de gestionar toda la información sobre los otros elementos del sistema. También incluye las interfaces necesarias para interactuar con el usuario u otra aplicación.
- **Actuadores:** dispositivos capaces de recibir una orden del controlador y realizar la acción pertinente. Pueden ser el motor de una puerta o el interruptor de una luz, por ejemplo.
- **Sensores:** Su función es la de obtener información del entorno para enviársela al controlador para que este realice las acciones necesarias. Pueden ser sensores de presencia o temperatura, interruptores o un mando de control remoto.

2.2.- Características generales

2.2.1.- Programación y ahorro de energía

- ✓ El ahorro energético no es algo tangible, sino un concepto al que se puede llegar de muchas maneras. En muchos casos no es necesario sustituir los aparatos o sistemas del hogar por otros que consuman menos energía sino una gestión eficiente de los mismos.
- ✓ Con un mando a distancia o control central se puede accionar un producto o agrupación de productos y activar o desactivar el funcionamiento del sensor.
- ✓ Se puede encender o apagar algunos sistemas mediante un dispositivo móvil.

2.2.2.- Confort

El confort conlleva todas las actuaciones que se puedan llevar a cabo que mejoren el confort en una vivienda. Dichas actuaciones pueden ser de carácter tanto pasivos, como activo o mixtos.

- ✓ Iluminación.
- ✓ Apagado general de todas las luces de vivienda
- ✓ Automatización del apagado/encendido en cada punto de luz.
- ✓ Automatización de los distintos sistemas /instalaciones/dotándolos de control eficiente y fácil manejo.
- ✓ Control vía bluetooth.
- ✓ Generación de marcos y programas de forma sencilla para el usuario y automatización.

2.2.3.- Seguridad

Consiste en una red de seguridad encargada de proteger tanto los bienes patrimoniales, como la seguridad personal y la vida.

- ✓ Alarma de intrusión: se utilizan para detectar o prevenir la presencia de personas extrañas en una vivienda o edificio.
- ✓ Sensores de magnéticos que pueden ser usados en ventanas y otros que permiten tener un control sobre estos.

2.2.4.- Comunicaciones

Son los sistemas o infraestructuras de comunicaciones que posee el hogar.

- ✓ Ubicación en el control tanto externo como interno, controles remotos desde internet, PC, mandos inalámbricos (por ejemplo con Wi FI).
- ✓ Transmisión de alarmas.
- ✓ Intercomunicación.

2.2.5.- Accesibilidad

Bajo este epígrafe se incluye las aplicaciones o instalaciones de control remoto del entorno que favorecen la autonomía personal de personas con limitaciones funcionales, o discapacidad.

El concepto diseño tendrá en cuenta las necesidades de los usuarios, incluyendo las personas con diferentes capacidades o discapacidades, es decir, favorecer un diseño accesible para la diversidad humana. La inclusión social y la igualdad son términos o conceptos más generalistas y filosóficas. La domótica aplicada a favorecer la accesibilidad es un reto ético y creativo pero sobre todo es la aplicación de la tecnología en el campo más necesario, para suplir limitaciones funcionales de las personas, incluyendo las discapacidades o mayores.

2.3.- ¿Qué es ARDUINO?

Arduino es una plataforma de prototipos electrónica de código abierto (open-source) basada en hardware y software flexibles y fáciles de usar. Está pensado para artistas, diseñadores, como hobby y para cualquiera interesado en crear objetos o entornos interactivos.

Arduino puede sentir el entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros artefactos. El microcontrolador de la placa se programa usando el “Arduino Programming Language” (basado en Wiring) y el “Arduino Development Environment” (basado en Processing2). Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador (por ejemplo con Flash, Processing, MaxMSP, etc.).

Las placas se pueden ensamblar a mano o encargarse preensambladas; el software se puede descargar⁵ gratuitamente. Los diseños de referencia del hardware (archivos CAD) están disponibles bajo licencia open-source, por lo que eres libre de adaptarlas a tus necesidades.

Arduino recibió una mención honorífica en la sección Digital Communities del Ars Electronica Prix en 2006.

2.4.- ¿Por qué ARDUINO?

Hay muchos otros microcontroladores y plataformas microcontroladoras disponibles para computación física. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, y muchas otras ofertas de funcionalidad similar.

Arduino también simplifica el proceso de trabajo con microcontroladores, pero ofrece algunas ventajas para estudiantes y aficionados interesados sobre otros sistemas:

- Barato: Las placas Arduino son relativamente baratas comparadas con otras plataformas microcontroladoras. La versión menos cara del módulo Arduino puede ser ensamblada a mano, e incluso los módulos de Arduino preensamblados cuestan menos de 50\$.
- Multiplataforma: El software de Arduino se ejecuta en sistemas operativos Windows, Macintosh OSX y GNU/Linux. La mayoría de los sistemas microcontroladores están limitados a Windows.
- Entorno de programación simple y clara: El entorno de programación de Arduino es fácil de usar para principiantes, pero suficientemente flexible para que usuarios avanzados puedan aprovecharlo también. Para profesores, está convenientemente basado en el entorno de programación Processing, de manera que estudiantes aprendiendo a programar en ese entorno estarán familiarizados con el aspecto y la imagen de Arduino.
- Código abierto y software extensible: El software Arduino está publicado como herramientas de código abierto, disponible para extensión por programadores experimentados. El lenguaje puede ser expandido mediante librerías C++, y la gente que quiera entender los detalles técnicos pueden hacer el salto desde Arduino a la programación en lenguaje AVR C en el cual está basado. De forma similar, puedes añadir código AVR-C directamente en tus programas Arduino si quieres.
- Código abierto y hardware extensible: El Arduino está basado en microcontroladores ATMEGA8 y ATMEGA168 de Atmel. Los planos para los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores experimentados de circuitos pueden hacer su propia versión del módulo, extendiéndolo y mejorándolo. Incluso usuarios relativamente inexpertos pueden construir la versión de la placa del módulo para entender cómo funciona y ahorrar dinero.

2.5.- Dispositivos y sensores

El dispositivo y los sensores nos ayudaran con controlar de manera efectiva el recinto estos son:

2.5.1.- Arduino Mega 2560

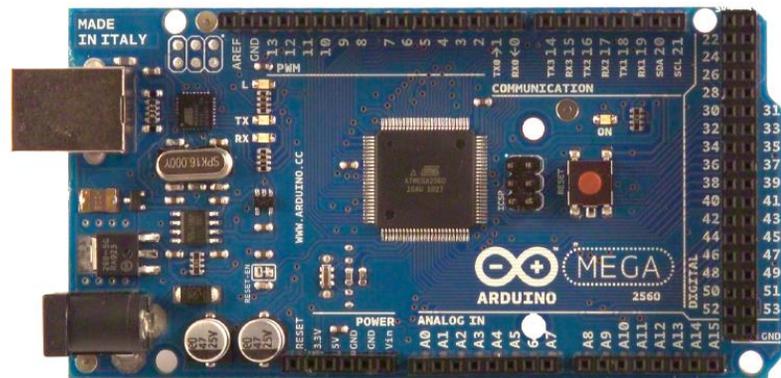


Figura 1 - Arduino Mega 2560

Fuente: <https://www.arduino.cc/en/Main/arduinoBoardMega>

El Arduino Mega 2560 es una placa microcontroladora basada en el ATmega2560. Cuenta con 54 pines digitales de entrada / salida (de los cuales 14 se pueden utilizar como salidas PWM), 16 entradas analógicas, 4 UART (puertos serie de hardware), un oscilador de cristal de 16 MHz, un puerto USB Conexión, una toma de corriente, una cabecera ICSP y un botón de reinicio. Contiene todo lo necesarios para soportar el microcontrolador; Simplemente se debe conectar a un ordenador con un cable USB o Conéctelo con un adaptador AC-DC o una batería para comenzar. El arduino Mega es compatible con la mayoría de los escudos diseñados para el Arduino Duemilanove o Diecimila.

Especificaciones técnicas

- Microcontrolador ATmega2560
- Tensión de funcionamiento 5V

- Tensión de entrada (recomendado) 7-12V
- Tensión de entrada (límites) 6-20V
- Pines de E / S digitales 54 (de los cuales 14 proporcionan salida PWM)
- Pines de entrada analógica 16
- Corriente CC por pin de E / S 40 mA
- Corriente CC para 3.3V Pin 50 mA
- Memoria Flash 256 KB de los cuales 8 KB utilizados por bootloader
- SRAM 8 KB
- EEPROM 4 KB
- Velocidad de reloj 16 MHz

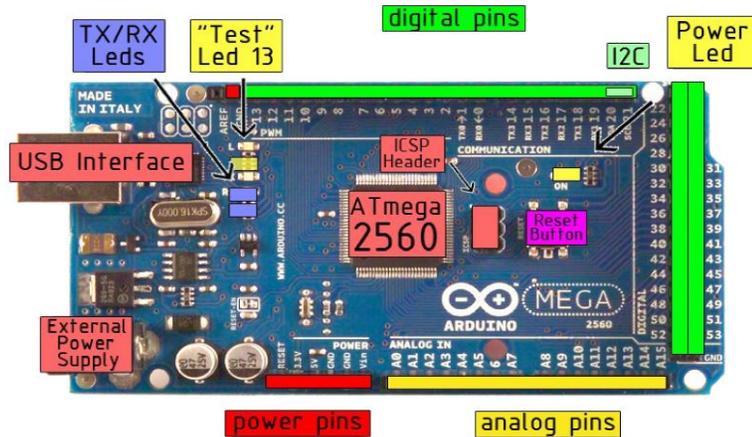


Figura 2 - Características del arduino ATmega2560

Fuente: <https://www.arduino.cc/en/Main/arduinoBoardMega>

Potencia

El Arduino Mega2560 se puede alimentar a través de la conexión USB o con una fuente de alimentación externa. La fuente de alimentación es seleccionada automáticamente. La alimentación externa (no USB) puede venir de un adaptador de CA a CC o de la batería. El adaptador se puede conectar enchufando un enchufe positivo de centro de 2.1mm en el enchufe de la energía de la tarjeta. Conduce desde una batería puede insertarse en los conectores POWER y Gnd. La placa puede funcionar con un suministro externo de 6 a 20 voltios. Si se suministra con menos de 7V, sin embargo, el pin de 5V puede suministrar menos

de cinco voltios y el tablero puede ser inestable. Si se utiliza más de 12V, el regulador de tensión puede sobrecalentarse y dañar el tablero. El rango recomendado es de 7 a 12 voltios.

El Mega2560 difiere de todos los tableros anteriores en que no utiliza el chip driver FTDI USB a serial. En su lugar, cuenta con el Atmega8U2 programado como un convertidor USB a serie.

Los pines de alimentación son los siguientes:

- **VIN.** Es el voltaje de entrada cuando se usa una fuente de alimentación externa (no tiene en cuenta la conexión USB). Se puede proporcionar voltaje a la placa a través de este pin, o en caso de que se esté utilizando una fuente de alimentación externa tomar el valor que está siendo suministrado.
- **5V.** La fuente de alimentación regulada utilizada para alimentar el microcontrolador y otros componentes de la placa. Esta puede venir de VIN a través de un regulador de a bordo, o ser suministrado por USB u otro suministro regulado de 5V.
- **3V3.** Una fuente de 3.3 voltios generada por el regulador de a bordo. El consumo máximo de corriente es de 50 mA.
- **GND.** Pasadores de tierra.

Memoria

El ATmega2560 tiene 256 KB de memoria flash para almacenar código (de los cuales 8 KB se utiliza para el gestor de arranque), 8 KB de SRAM y 4 KB de EEPROM (que se pueden leer y escribir con la librería EEPROM).

Entradas y salidas

Cada uno de los 54 pines digitales del Mega se puede utilizar como entrada o salida, usando pinMode (), digitalWrite (), y funciones digitalRead (). Funcionan a 5 voltios. Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una

resistencia interna pull-up (desconectada por defecto) de 20-50 kOhms. Además, algunos pins tienen funciones especializadas:

- **Serial: 0 (RX) y 1 (TX); Serie 1: 19 (RX) y 18 (TX); Serie 2: 17 (RX) y 16 (TX); Serie 3: 15 (RX) y 14 (TX).** Se utiliza para recibir (RX) y transmitir (TX) datos en serie TTL. Los pins 0 y 1 también están conectados al correspondientes de la ATmega8U2 USB-to-TTL Serial chip.
- **Interrupciones externas: 2 (interrupción 0), 3 (interrupción 1), 18 (interrupción 5), 19 (interrupción 4), 20 (interrupción 3) y 21 (Interrupción 2).** Estos pines pueden configurarse para activar una interrupción en un valor bajo, un flanco ascendente o descendente o un cambio de valor. Vea la función `attachInterrupt ()` para más detalles.
 - **PWM: 0 a 13.** Proporcione salida PWM de 8 bits con la función `analogWrite ()`.
 - **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** Estos pines soportan la comunicación SPI, la cual, aunque proporcionado por el hardware subyacente, no está incluido actualmente en el lenguaje Arduino. Los pines SPI también estallado en la cabecera ICSP, que es físicamente compatible con el Duemilanove y Diecimila.
 - **LED: 13.** Hay un LED incorporado conectado al pin digital 13. Cuando el pin tiene el valor HIGH, el LED está encendido, cuando el pin es LOW, está apagado.
 - **I2C: 20 (SDA) and 21 (SCL).** La comunicación I2C (TWI) utilizando la biblioteca de cables. Tenga en cuenta que estos pines no están en la misma ubicación que los pines I2C en el Duemilanove.

El Mega2560 tiene 16 entradas analógicas, cada una de las cuales proporciona 10 bits de resolución (es decir, 1024 valores diferentes). Por defecto miden de tierra a 5 voltios, aunque es posible cambiar el extremo superior de su gama usando el perno de AREF y la función `analogReference ()`.

Hay otros dos pines en la placa:

- **Referencia para pines analógicos (AREF):** Tensión de referencia para entradas analógicas. Se utiliza con la función *analogReference()*.
- **Reiniciar.** Podemos imitar el funcionamiento del botón reset suministrando un valor LOW (0V) para reiniciar el microcontrolador.

Comunicación

El Arduino Mega2560 tiene una serie de facilidades para comunicarse con una computadora, otro Arduino, u otros microcontroladores. El ATmega2560 proporciona cuatro UART de hardware para la comunicación en serie TTL (5V).

Un ATmega8U2 en la placa canaliza uno de estos a través de USB y proporciona un puerto virtual a software en el equipo (las máquinas con Windows necesitarán un archivo .inf, pero las máquinas OSX y Linux reconocerán la placa.

Como un puerto COM automáticamente. El software Arduino incluye un monitor en serie que permite que se envíen desde y hacia el consejo. Los LED RX y TX de la tarjeta parpadearán cuando se estén transmitida a través del chip ATmega8U2 y conexión USB a la computadora (pero no para la comunicación serial en los pines 0 y 1).

Una biblioteca de SoftwareSerial permite la comunicación en serie en cualquiera de los pines digitales del ATmega.

El ATmega2560 también soporta comunicación I2C (TWI) y SPI. El software Arduino incluye un cable biblioteca para simplificar el uso del bus I2C; Consulte la documentación en el sitio web de cableado para obtener más detalles. Para usar el SPI comunicación, consulte la hoja de datos ATmega2560.

2.5.2.- Sensor HC-SR501 PIR de movimiento

Los sensores PIR, que son elementos que detectan cambios en la radiación infrarroja que reciben y que disparan una alarma al percibirlo.



Figura 3 - Sensor HC-SR501

Fuente: <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work>

El propio sensor IR se aloja en una lata metálica sellada herméticamente para mejorar la inmunidad de ruido / temperatura / humedad. Hay una ventana hecha de material transmisor de IR (normalmente silicio recubierto ya que es muy fácil de encontrar) que protege el elemento sensor. Detrás de la ventana están los dos sensores balanceados.

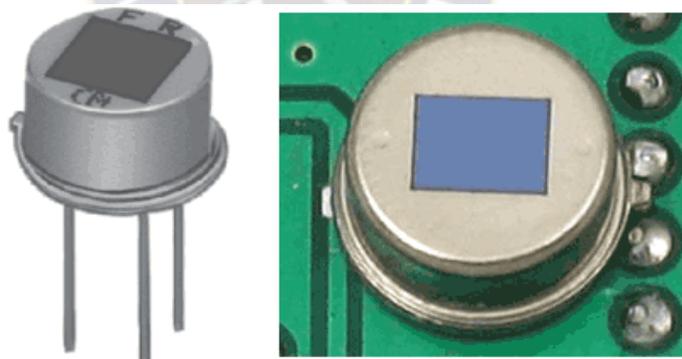


Figura 4 - Sensor infrarrojo

Fuente: <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work>

La lente usada aquí no está realmente haciendo mucho y así vemos que las dos ranuras pueden ver más allá de una cierta distancia (básicamente la sensibilidad del sensor). Cuando el sensor está inactivo, ambas ranuras detectan la misma cantidad de IR, la cantidad ambiental irradiada desde la habitación o las paredes

o al aire libre. Cuando pasa un cuerpo caliente como un ser humano o un animal, primero intercepta una mitad del sensor PIR, lo que provoca un cambio diferencial positivo entre las dos mitades. Cuando el cuerpo caliente sale del área de detección, sucede lo contrario, por lo que el sensor genera un cambio diferencial negativo. Estos impulsos de cambio son lo que se detecta.

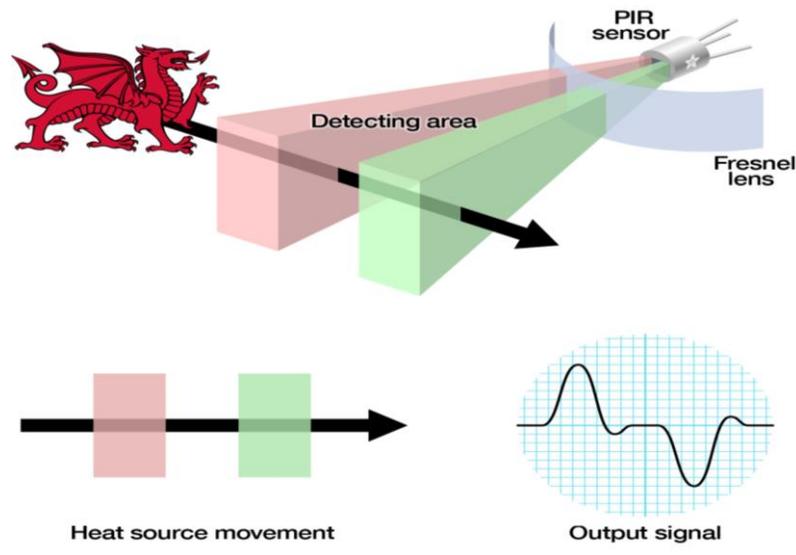


Figura 5 - Forma de trabajo del sensor PIR

Fuente: <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work>

Lentes

Los sensores PIR son bastante genéricos y en su mayor parte varían sólo en cuanto a precio y sensibilidad. La mayor parte de la verdadera magia ocurre con la óptica.

En el diagrama encima de la tapa, la lente es apenas un pedazo de plástico, pero eso significa que el área de la detección es apenas dos rectángulos. Por lo general, nos gustaría tener un área de detección que es mucho más grande. Para ello utilizamos una lente simple como la que se encuentra en una cámara: condensa una gran área (como un paisaje) en una pequeña (en película o un sensor CCD). Por razones que serán evidentes pronto, nos gustaría hacer que las lentes PIR pequeños y delgados y moldeables de plástico barato, a pesar de

que puede añadir distorsión. Por esta razón los sensores son en realidad lentes de Fresnel:

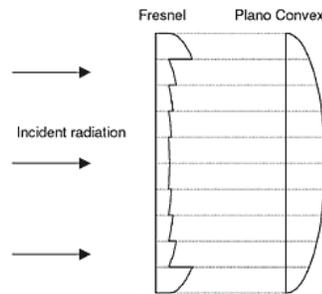


Figura 6 - Lente de fresnel

Fuente: <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work>

La lente de Fresnel condensa la luz, proporcionando un mayor rango de IR al sensor.

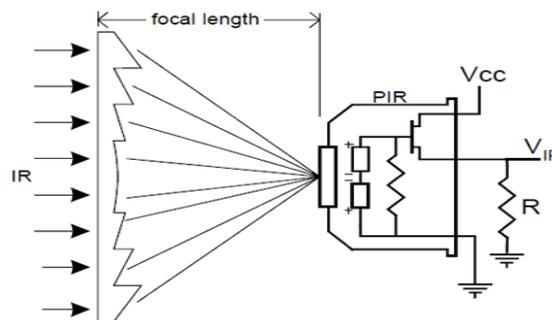


Figura 7- Condensación de luz

Fuente: <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work>

Características:

En la figura se ve que tiene 3 pines en el lateral, que usan para conectar el sensor PIR **HC-SR501**.

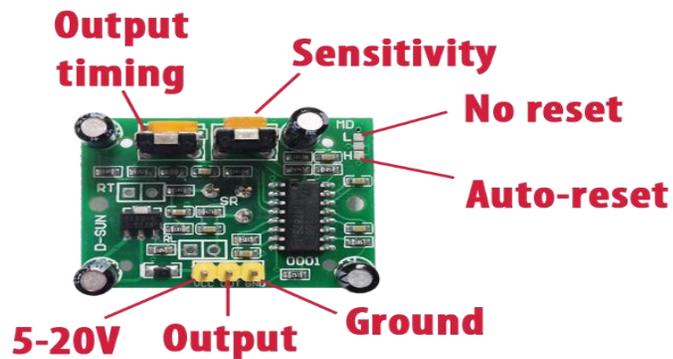


Figura 8 - Característica del sensor PIR

Fuente: <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work>

También se ve en la figura que hay un par de **potenciómetros**, que podemos usar para ajustarlo para la sensibilidad y el tiempo de respuesta del sensor

2.5.3.- Módulo comunicación bluetooth HC-05

Para empezar diremos que los dispositivos Bluetooth pueden actuar como Masters o como Slaves (Amos o esclavos).

La diferencia es que un Bluetooth Slave solo puede conectarse a un master y a nadie más, en cambio un master Bluetooth, puede conectarse a varios Slaves o permitir que ellos se conecten y recibir y solicitar información de todos ellos, arbitrando las transferencias de información (Hasta un máximo de 7 Slaves)

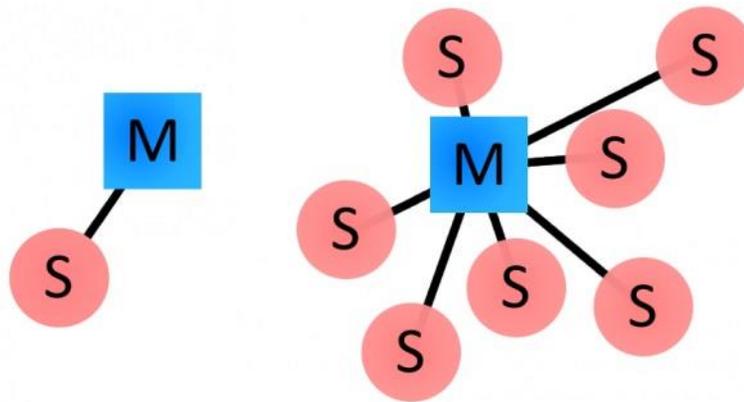


Figura 9 - Diagrama maestro esclavo

Fuente: <http://www.geekfactory.mx/tutoriales/bluetooth-hc-05-y-hc-06-tutorial-de-configuracion/>

Cada uno de los dispositivos que se identifican vía Bluetooth presenta una dirección única de 48 bits y además un nombre de dispositivo que nos sirva para identificarlo cómodamente a los humanos. Por eso cuando configuras tu móvil puedes especificar un nombre propio que será el que mostraras a los demás cuando busquen tu teléfono en la inmediaciones.

La dirección propia también se puede identificar pero lógicamente, es un poco menos cómoda y tiene menos utilidad. Tampoco es raro establecer un protocolo IP sobre transporte Bluetooth, con lo que además de su identificación interna Bluetooth (Equivalente al MAC Ethernet) dispondrá de una dirección IP para conectarse a Internet.

Así pues un nodo Bluetooth puede ser Master o Slave y dispone de una dirección única, así como de un nombre para identificarse y muy habitualmente también incluye un PIN de conexión o número de identificación que debe teclearse para ganar acceso al mismo.

Como el Bluetooth lo desarrolló Nokia para conectar teléfonos móviles, a otros dispositivos como auriculares, micrófonos o conexiones al audio del coche, existe

un procedimiento definido que se llama Pairing (o emparejamiento) que vincula a dos dispositivos Bluetooth.

Cuando vinculas dos dispositivos BT (bluetooth), se inicia un proceso en el que ellos se identifican por nombre y dirección interna y se solicitan la clave PIN para autorizar la conexión.

Si el emparejamiento se realiza con éxito, ambos nodos suelen guardar la identificación del otro y cuando se encuentran cerca se vuelven a vincular sin necesidad de intervención manual.

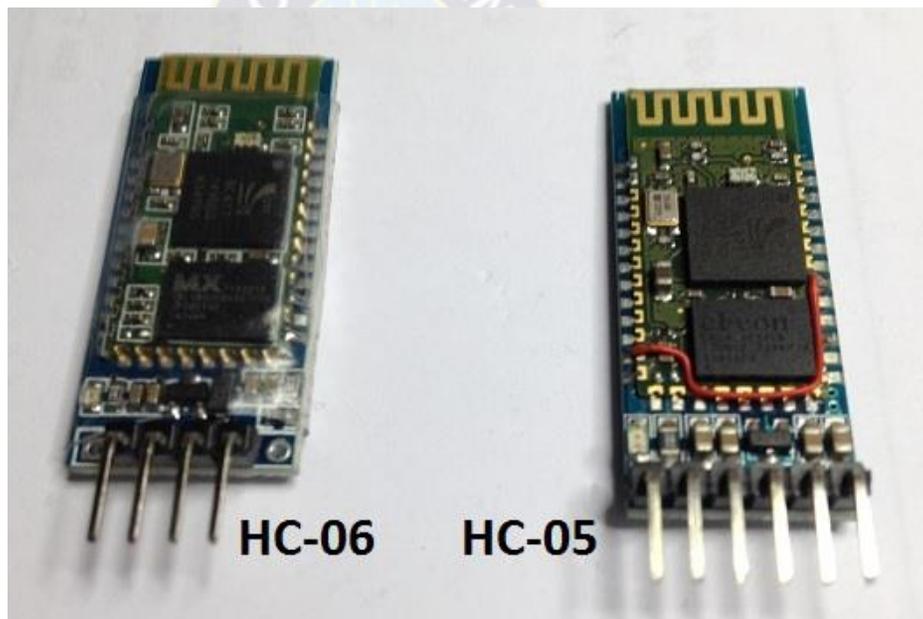


Figura 10 - Modulo bluetooth HC-06 esclavo y HC05 maestro esclavo

Fuente: <http://www.geekfactory.mx/tutoriales/bluetooth-hc-05-y-hc-06-tutorial-de-configuracion/>

El modelo HC-06 dispone de 4 pines, en lugar de los 6 que incluye el modelo HC-05, pero hay además importantes diferencias de funcionalidad y de manejo que hace que merezca la pena dedicar una sesión a cada modelo.

Básicamente el modelo HC-06 solo puede actuar como esclavo y además dispone de un juego reducido de instrucciones a las que atiende, mientras que el modelo HC-05 puede actuar como master o como Slave y acepta un número mayor de órdenes de configuración.

Características:

- Modulo Bluetooth Master y Slave HC-05
- Protocolo bluetooth: Bluetooth especificación V2.0+EDR
- Frecuencia: 2.4Ghz ISM Band
- Modulación: GFSK
- Poder de transmisión: ≤ 4 dBm Class 2
- Security features: Authentication and encryption
- Pines: VCC, GND, TxD, RxD, KEY, State (LED)
- Distancia bluetooth: 10 metros
- Tamaño compacto

Interfaz de configuración de comandos AT en HC-05

El puerto serie en modo de configuración para el HC-05 debe configurarse de la siguiente manera: **34800 bps, 8 bits de datos, Sin paridad, Sin control de flujo.**

Para entrar al modo de comandos AT seguimos los siguientes pasos:

1. Poner a estado alto en el pin 34 (PIO11)
2. Conectar la alimentación del módulo (o resetearlo de preferencia)
3. Enviar un comando **AT\r\n** para comprobar que estemos en modo de comando AT.

La siguiente lista es una compilación de los comandos que consideramos importantes:

- **AT\r\n**: Comando de prueba, debe responder con OK\r\n

- **AT+ROLE=1\r\n**: Comando para colocar el módulo en modo Maestro (Master)
- **AT+ROLE=0\r\n** :Comando para colocar el módulo en modo Esclavo (Slave)
- **AT+VERSION?\r\n**: Obtener la versión del firmware
- **AT+UART=115200,1,2\r\n**: Configurar el modo de funcionamiento del puerto serie en “modo puente”
- **AT+PIO=10,1\r\n**: Colocar el pin de IO de propósito general a nivel alto

2.5.4.- Módulo de 4 relés para Arduino

El relé es un dispositivo electromecánico. Funciona como un interruptor controlado por un circuito eléctrico en el que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes.

Hecha esta salvedad, comencemos con una descripción general de la placa. Se trata de un módulo de 4 relés (o relays) que funcionan a 5 Voltios, capaces de manejar cargas de hasta 10 Amperes en 250 Voltios, convenientemente aislados mediante optoacopladores de las entradas, las que cuentan con leds individuales que sirven como indicadores de estado.

Los distintos componentes del módulo pueden verse en la siguiente imagen:



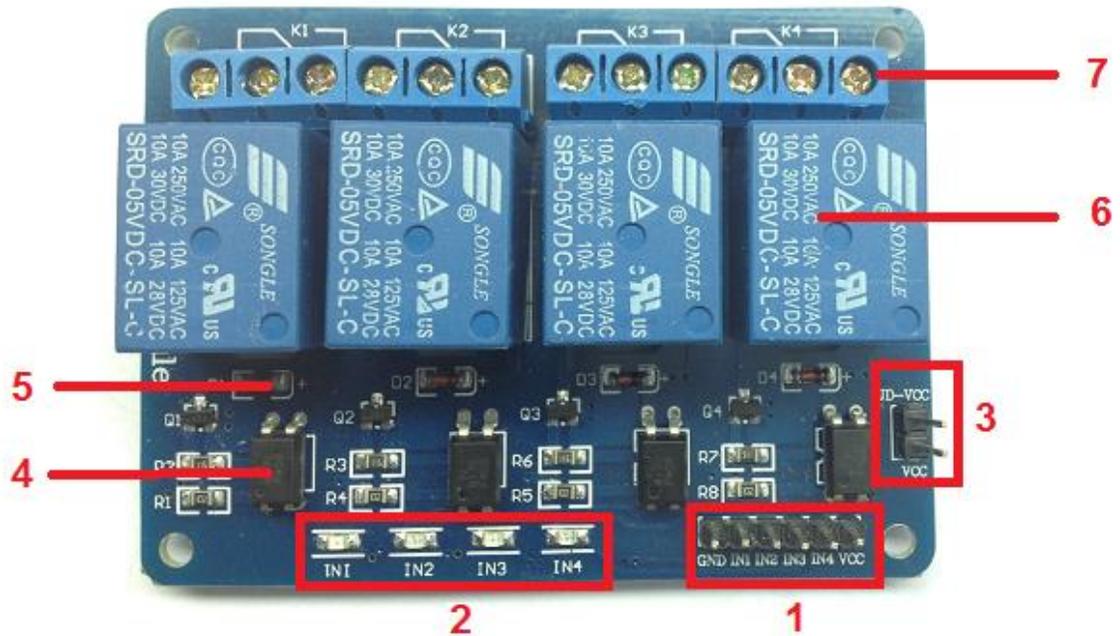


Figura 11 - Partes del módulo

Fuente: <http://dinastiatecnologica.com/producto/modulo-rele-de-4-canales-arduino/>

Como se puede apreciar, la placa tiene un conector de entradas (IN1 a IN4) y alimentación (GND es masa o negativo y Vcc es el positivo) [1], cuatro leds que indican el estado de la entradas [2], un jumper selector para la alimentación de los relés [3], cuatro optoacopladores del tipo FL817C [4], cuatro diodos de protección [5], cuatro relés marca SONGLE con bobinas de 5V y contactos capaces de controlar hasta 10 Amperes en una tensión de 250V [6] y cuatro borneras, con tres contactos cada una (Común, Normal abierto y Normal cerrado), para las salidas de los relés [7].

Esquemático

En la imagen de más abajo se puede apreciar el circuito esquemático de un canal, el resto de los canales repite la misma configuración.

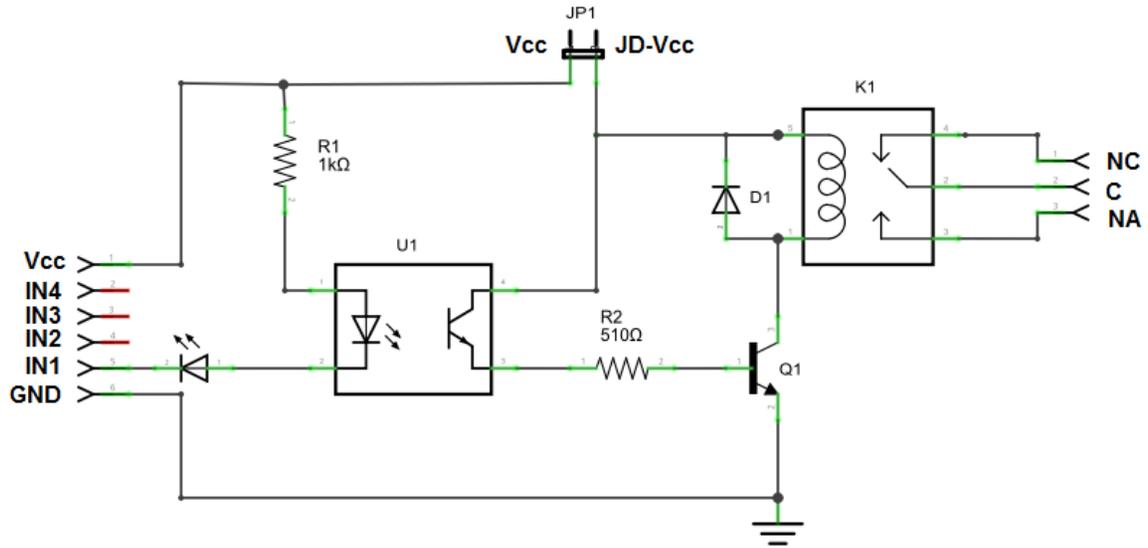


Figura 12 - Circuito esquemático

Fuente: <http://dinastiatecnologica.com/producto/modulo-rele-de-4-canales-arduino/>

Funcionamiento

Analizando la figura 12 la entrada IN1 está conectada al cátodo del diodo del optoacoplador a través del led indicador. El ánodo del diodo del optoacoplador se conecta a Vcc (positivo) por intermedio de R1, una resistencia de 1000 ohms. Estos tres componentes, el diodo indicador, el diodo del opto y la R1 forman un circuito serie por el cual circula la corriente cuando la entrada está a un nivel BAJO (conectada a GND) y no circula si la entrada está a un nivel ALTO (conectada a Vcc).

El transistor del opto tiene su colector a JD-Vcc y su emisor conectado a Q1 a través de una resistencia de 510 ohms. Este es otro circuito serie por el cual circula corriente cuando el transistor del opto conduce al ser “iluminado” por su diodo, con lo que se introduce corriente en la base de Q1 a través de R2.

Finalmente, Q1 está conectado en una típica configuración emisor común, con su emisor a masa (GND) y la bobina del relé como carga en el colector. Cuando circula corriente por la base desde el opto, Q1 se satura permitiendo el paso de la corriente a través de la bobina del relé, lo que produce que se cierren los contactos del mismo (común con normal abierto). El diodo D1 protege al transistor

de la tensión que aparece en la bobina del relé cuando deja de circular corriente por la misma.

En síntesis, al ponerse la entrada a nivel BAJO se pone a la saturación el transistor Q1 a través del optoacoplador con lo que se cierra el contacto normal abierto del relé.

Alimentación y consumo

La forma más sencilla de alimentar este módulo es desde Vcc y GND de la placa Arduino, manteniendo el Jumper en su lugar, con lo que $JD-Vcc = Vcc$. Esta conexión tiene dos limitaciones importantes:

- Se pierde la aislación eléctrica que brindan los optoacopladores, lo que aumenta la posibilidad de daño al Arduino si hay algún problema con las cargas de los relés.
- La corriente consumida por las bobinas de los relés debe ser provista por la placa Arduino. Cada bobina consume unos 90 mA y las cuatro juntas suman 360 mA. Si a esto le sumamos los consumos que pueden tener otras salidas, estamos muy cerca de los 500 mA que puede suministrar un puerto USB. En este caso se debería alimentar al Arduino con una fuente externa, lo que aumenta el límite de corriente a 1 A (en el caso de la Arduino UNO).

2.5.5.- Sensor de temperatura y humedad DHT 11

Los sensores DHT11 y DHT22 son los más básicos y los más utilizados para implementarlos con Arduino, estos sensores están compuestos en dos partes, un sensor de humedad capacitivo y un termistor, también constan de un circuito integrado básico en el interior que hace la conversión de analógico al digital y este envía una señal digital con la temperatura y la humedad.

El módulo de la ilustración corresponde a un DHT11 capaz de representar digitalmente la humedad ambiental medida en % además de la temperatura en

°C. Tiene una precisión decimal y dispone de su propia librería que contiene los métodos para recoger sus mediciones.

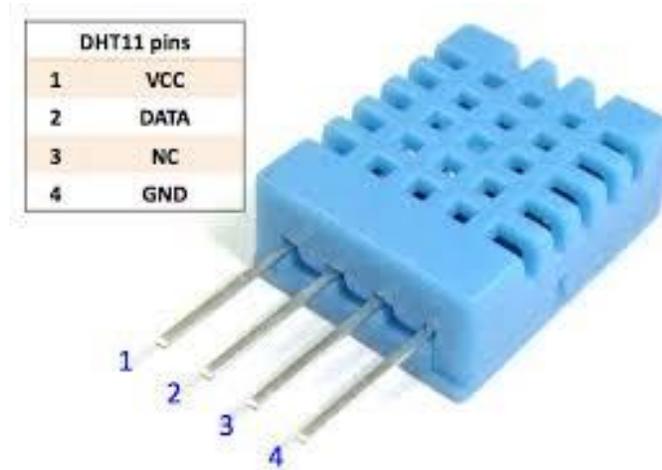


Figura 13 - Sensor DHT11

Fuente: <http://www.prometec.net/sensores-dht11/>

Los sensores DHT11 cuentan con cuatro pines:

- VCC (3.3VDC A 5VDC) (pin 1)
- Salida de datos
- No se conecta (pin 3)
- GND

Características técnicas

- Alimentación de 3.3V a 5VDC
- Corriente máxima 2.5mA durante la conversión
- Lectura de humedad con un +/- 5% de precisión
- Lectura de temperatura con un +/- 2°C de precisión
- Capaz de medir humedad de 20% a 80%
- Capaz de medir temperatura de 0 a 50°C
- No más de 1 Hz en velocidad de muestreo (una vez cada segundo)
- Dimensiones: 15.5mm x 12mm x 5.5mm

De acuerdo a las características es un sensor ligeramente más preciso y cuenta con un rango un poco mayor. Utiliza un único pin digital de salida y son lentos en cierto aspecto, ya que no se puede consultar más de una vez cada uno o dos segundos.

2.5.6.- Sensor magnético reed-switch

Consiste en un par de láminas metálicas de materiales ferromagnéticos metidas en el interior de una cápsula que se atraen en presencia de un campo magnético, los contactos se cierran provocado por un dispositivo imantado alojado en el objeto a detectar; cuando el campo magnético se acerca al sensor, estos transmiten una señal eléctrica o neumática a los controles.

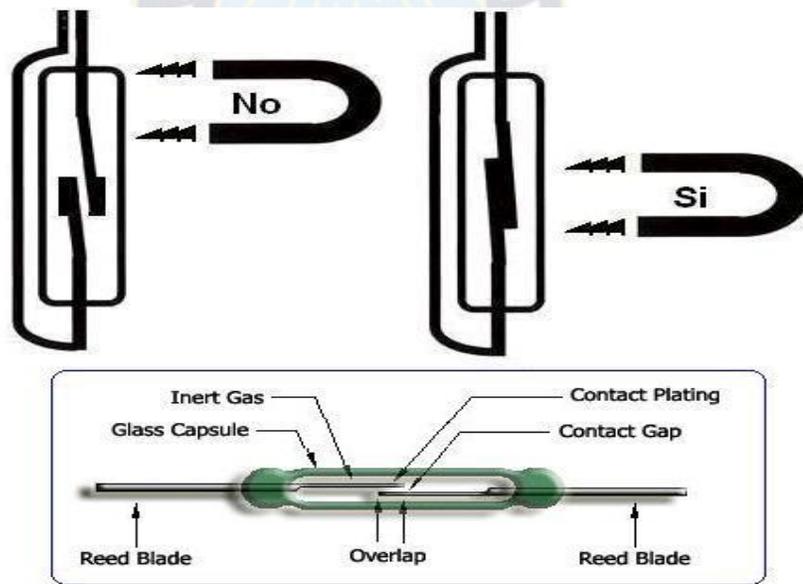


Figura 14 - Reed-switch

Fuente: <http://arduineando.matem.unam.mx/proyectos/reedSwitch>

Características

- Montaje de superficie
- Distancia de detección 15 a 20mm

- Salida NORMALMENTE CERRADA NC
- Bornera de conexión
- Sensor conectable a la central de alarma a través de un cable.
- Voltaje de entrada, de 5 a 16 voltios.
- Grado de protección: IP67
- Temperatura de funcionamiento: -40°C ~ +70°C

2.5.7.- Aplicación Electrónica Bluetooth



Figura 15 - Aplicación bluetooth electronics

Fuente: <http://dinastiatecnologica.com/producto/modulo-rele-de-4-canales-arduino/>

Esta aplicación se comunica mediante Bluetooth a un módulo Bluetooth HC-06 o HC-05 en su proyecto. Esta aplicación viene con una biblioteca que contiene 10 ejemplos de Bluetooth para Arduino. Consulte la página electrónica para ver ejemplos.

Gran selección de controles disponibles, incluyendo botones, interruptores, deslizadores, almohadillas, luces, medidores, terminales, acelerómetros y gráficos, arrastrar y soltar en la cuadrícula del panel. A continuación, edite sus propiedades.

20 paneles personalizables disponibles. Importar / Exportar paneles para compartirlos.

Brillo LED - PWM con control deslizante

Demostración de coches RC - Controles de botones básicos

Persistencia de la visión - Control de texto

Demostración del repetidor - Terminales de envío y recepción

Sensor de Distancia Ultrasónica - Indicador de Luz

MEGA Monitor - Gráficos

UNO Monitor - Más gráficos

Temperatura y humedad - Medidor de temperatura

Configurar HC-06 Demo - En caso de que desee cambiar la velocidad de transmisión

Motor Control Demo - Acelerómetro y controles de la almohadilla

Guía del usuario

La guía del usuario se encuentra en la aplicación haciendo clic en el botón 'i' en la pantalla principal.

Algunas capturas de pantalla:

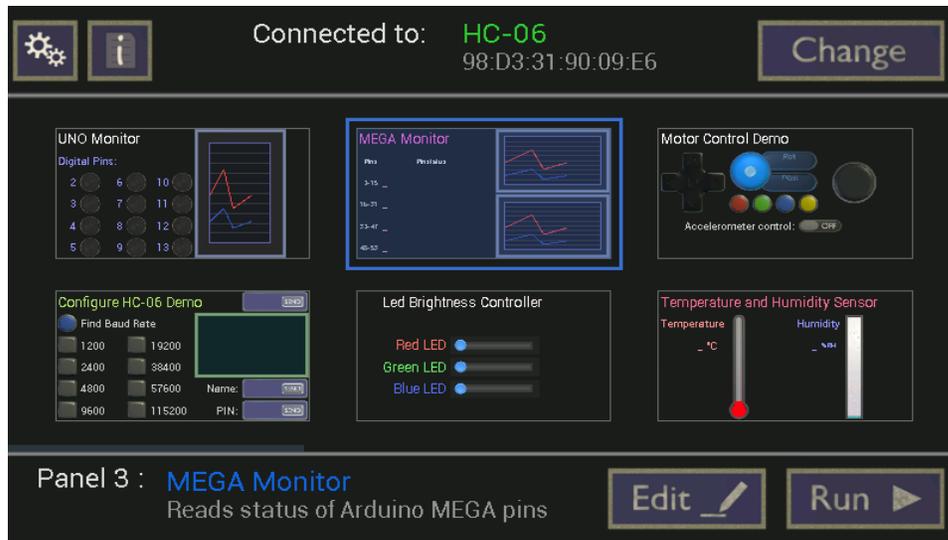


Figura 16 - Menú principal

Fuente: <http://dinastiatecnologica.com/producto/modulo-rele-de-4-canales-arduino/>



Figura 17 - Editar pantalla

Fuente: <http://dinastiatecnologica.com/producto/modulo-rele-de-4-canales-arduino/>

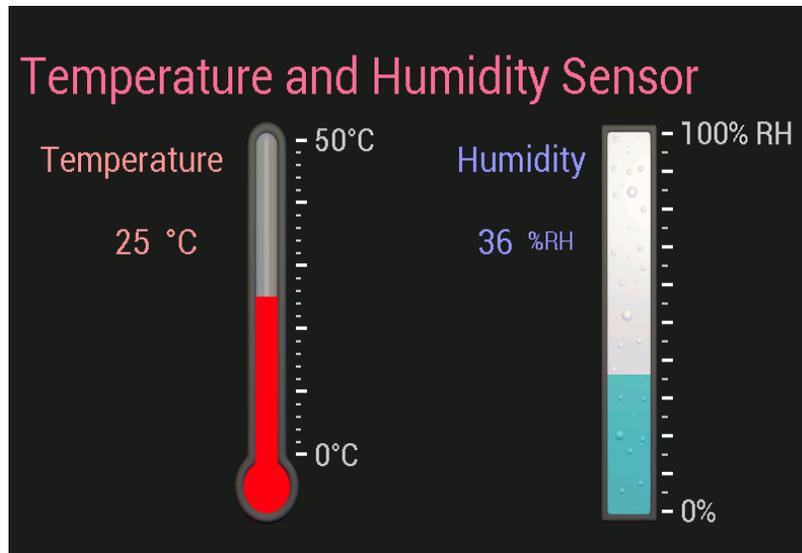


Figura 18 - Ejemplo de gestión de pantalla

Fuente: <http://dinastiatecnologica.com/producto/modulo-rele-de-4-canales-arduino/>



CAPITULO III

INGENIERIA DEL PROYECTO

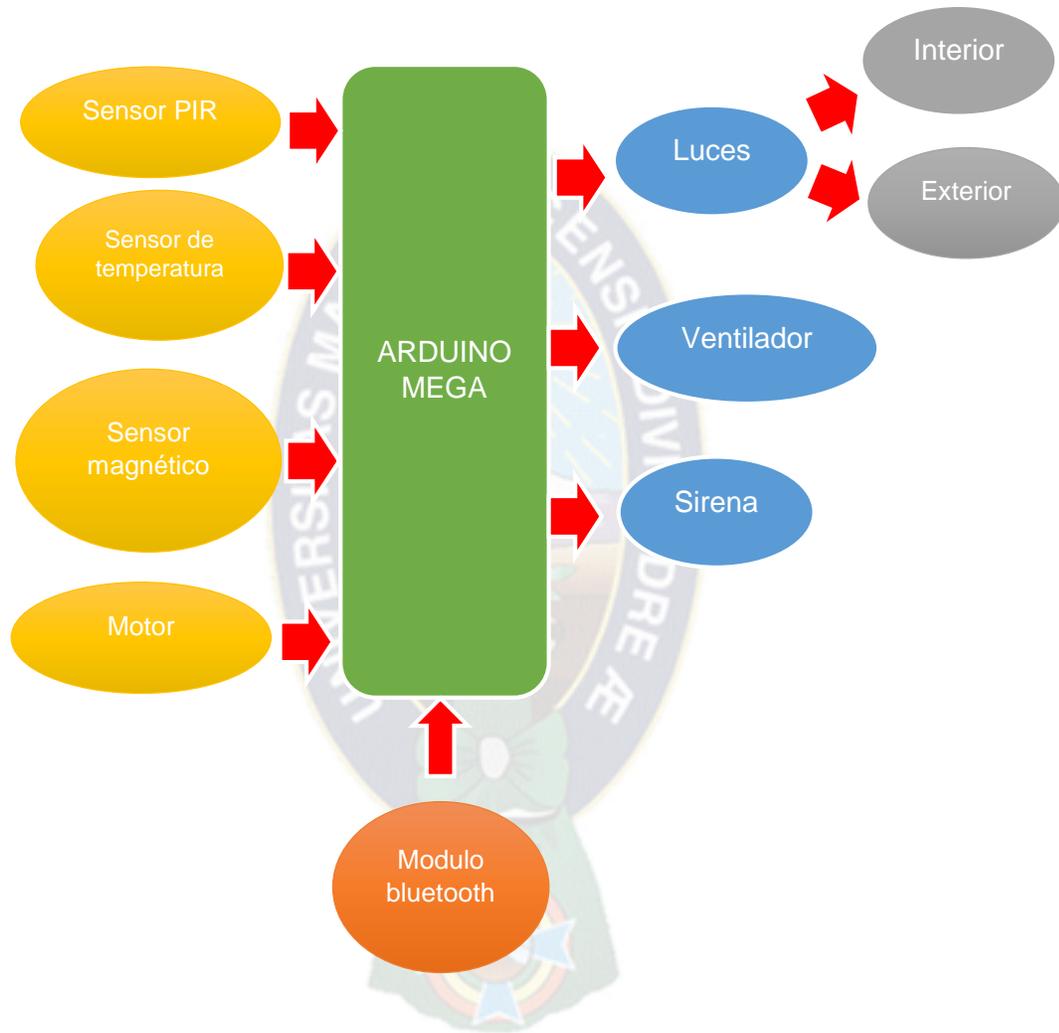


Figura 19 - Diagrama en bloques de control por arduino

Fuente: Propia

Descripción del diagrama de bloque

El proyecto en sí está elaborado, tanto para el control de seguridad, el control remotamente de algunas piezas de un recinto o una casa.

El proyecto funciona de la siguiente forma:

Modulo bluetooth: Este módulo nos permite interactuar con un dispositivo android, como es el celular, misma que permite controlar remotamente, el encendido y apagado de focos, la apertura y cierre de un garaje o puerta, el encendido y apagado del ventilador.

Sensor PIR DH-SH501: El sensor se encarga de detectar cambios en la radiación infrarroja que reciben del área que se dese controlar, al detectar un cambio en la habitación, envía un pulso que deriva en la acción de la sirena de alarma, alertando al usuario.

Sensor magnético reed-switch: Este sensor tiene la finalidad de precautelar las entradas en puertas o ventanas, la función del sensor actúa, casi de la misma manera que un relé, que se encuentra en estado normalmente abierto, cuando está dentro de un campo magnético generado por un imán que al separarse a una cierta distancia, se cierra dejando circular la corriente, de forma que se envía un uno lógico al arduino, misma pone en activo la sirena de alarma.

Sensor de temperatura DHT11: Este sensor nos permite medir la temperatura de 0° a 100°C además de la Humedad. Mediante el arduino se verifica que la temperatura da la habitación, no exceda los 20°, de ser así, se acciona el ventilador hasta que la temperatura este por debajo del nivel establecido.

Sirena: Es usada por el arduino, que mediante el control que realiza por los sensores cuando son activadas de esta manera alertar al usuario,

Ventilador: Funciona en conjunción con el sensor de temperatura, cuya función es mantener el ambiente a un nivel determinado, además que puede ser accionado remotamente mediante el celular a través del módulo bluetooth.

Focos: Los focos proporcionan la iluminación a los recintos, que en este caso será dos uno en plata baja y la otra en planta alta, son controladas remotamente por el celular a través del bluetooth.

Motor: El motor nos permite controlar el garaje, para este caso, también se controla remotamente, permitiendo la accesibilidad al usuario sin la necesidad de usar una llave. El giro del motor se lleva a cabo mediante el circuito integrado L293D que facilita este trabajo.

3.1.- Actividades a realizar

Las actividades que se llevarán a cabo para este proyecto son las siguientes:

Tarea 1: Para empezar se probará el módulo Bluetooth mediante el Arduino Mega y realizar su respectivo código, misma que nos permitirá enviar acciones al Arduino mediante dispositivo móvil.

Tarea 2: Para la parte dos, nos enfocaremos en los sensores, mismas que se probarán, cada una de acuerdo a la necesidad que se requiera, y controlarlos mediante el Arduino.

Tarea 3: Nos enfocaremos en la parte del control de la puerta que, tendrá un acceso mediante un motor DC a través de Bluetooth.

Tarea 4: Se llevarán las primeras pruebas, en base a las tareas ya mencionadas anteriormente.

Tarea 5: Se trabajará en la parte de la aplicación, desarrollando la misma de acuerdo a los requerimientos, que será mediante una aplicación Android, que nos permita interactuar con el Arduino.

Tarea 6: Una vez ya casi finalizando se realizará una segunda prueba con todos los componentes que se mencionó.

Tarea 7: Para esta etapa se procederá al montaje del prototipo con las especificaciones que se ha descrito.

Tarea 8: Finalmente se realizará la última prueba con el prototipo montado para la posterior presentación del proyecto.

3.1.1.- Modulo bluetooth

Dentro de lo que podemos hacer con Arduino como amantes de la electrónica, es imprescindible hoy en día que nuestros proyectos tengan una comunicación con los dispositivos móviles, ya sean tabletas, teléfonos móviles inteligentes y computadoras portátiles.

Se realizara una comunicación a través de un módulo Bluetooth HC-05 entre un Arduino Mega 2560 y un teléfono móvil inteligente y mostraremos los mensajes que sean enviados desde el teléfono al arduino Mega 2560.

Para este ejemplo disponemos de un módulo Bluetooth HC-05 que podemos conectar al puerto Serial de Arduino. Resumidamente hemos de conectar los pines de recepción (RX) y transmisión (TX); que por defecto son los pines 0 y 1 respectivamente de nuestra placa con los pines de transmisión (TX) y recepción (RX) del módulo de forma inversa; tal y como se indica en la figura.

Lo más importante a tener en cuenta es que debemos de activar el modo de configuración del Bluetooth a través del PIN EN o KEY, que debe de estar activo con un voltaje y que en este caso conectemos al pin de 3,3V.

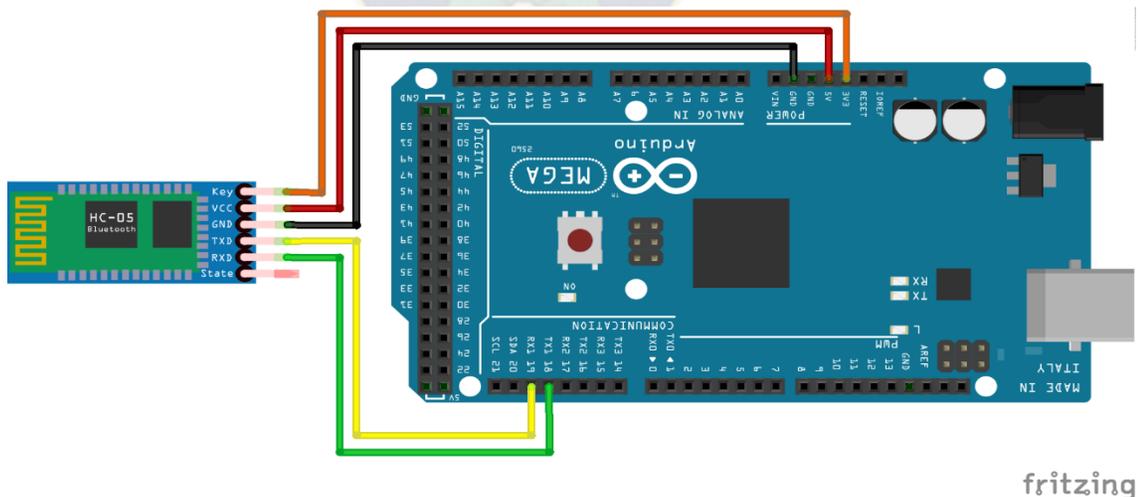


Figura 20 - Diagrama de conexión del Bluetooth HC-05

Fuente: <http://www.geekfactory.mx/tutoriales/bluetooth-hc-05-y-hc-06-tutorial-de-configuracion/>

3.1.1.1.- Código para el módulo bluetooth

Éste código es de los más cómodos para utilizar, ya que podemos enviar, como recibir en el momento la información de configuración a través del monitor u otro. Así que no necesitaremos cargar continuamente distintos sketches para configurar el módulo. Con esta opción también se pueden conectar varios módulos Bluetooth y poder intercambiar información entre ellos, que es un buen modo de desarrollo y aprendizaje para este fin.

```
char rxChar;          // Variable para recibir datos del puerto serie
int ledpin = 13;      // Pin donde se encuentra conectado el led (pin 13)

// Configurar el arduino
void setup()
{
  // Pin 13 como salida
  pinMode(ledpin, OUTPUT);
  // Comunicación serie a 9600 baudios
  Serial.begin(9600);
}

// Ciclo infinito, programa principal
void loop()
{
  // Si hay datos disponibles en el puerto serial
  if( Serial.available() )
  {
    // Leer un byte y colocarlo en variable
    rxChar = Serial.read();

    if( rxChar == 'A' )
    {
```

```

digitalWrite(ledpin, HIGH);
Serial.println("ON");
}
else if ( rxChar == 'a' )
{
digitalWrite(ledpin, LOW);
Serial.println("OFF");
}
}

```

3.1.2.- Sensor de temperatura DHT11

Este sensor nos proporciona la temperatura y la humedad relativa. Usan un encapsulado de 4 pines, aunque solo usan 3, y emplean para su funcionamiento un pin digital. Su uso se complica un poco más respecto el anterior, sin embargo haciendo uso de la librería adjunta se facilita todo mucho.

El DHT11 tiene un rango de medición de 0 a 50°C y 20 a 90%RH con una tolerancia del 5% y una precisión de 1. Este sensor es bastante lento, refrescando cada 2 segundos aproximadamente.

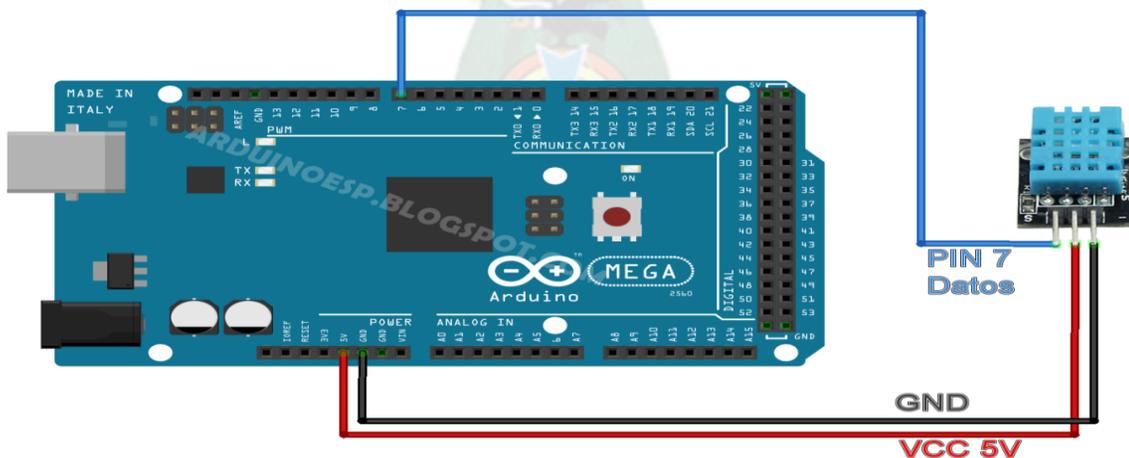


Figura 21 - Diagrama de conexión sensor de temperatura DHT11

Fuente: <http://www.prometec.net/sensores-dht11/>

3.1.2.1.- Código para el sensor DHT11

Para empezar se debe descargar una librería para manejarlos cómodamente, DX11.zip e importarla.

Hacemos el importar la librería DHT11:

```
#include <DHT11.h> //libreria
int pin=4; //pin de entrada de datos
DHT11 dht11(pin); //combinando el pin con la variable de dht11
void setup()
{
  Serial.begin(9600); //velocidad de comunicacion serial
  while (!Serial) {
    ;
  }
}
void loop()
{
  int err;
  float temp, humi;
  if((err=dht11.read(humi, temp))==0)
  {
    Serial.print("temperature: ");
    Serial.print(temp);
    Serial.print("C");
    Serial.print(" humidity: ");
    Serial.print(humi);
    Serial.print("%");
    Serial.println();
  }
}
```

```

else
{
  Serial.println();
  Serial.print("Error No :");
  Serial.print(err);
  Serial.println();
}
delay(DHT11_RETRY_DELAY); //delay for reread
}

```

3.1.3.- Sensor PIR HC-SR501

Sensores PIR pueden disparar directamente una alarma con una señal de 3.3V y son capaces de excitar pequeños relés, de modo que no necesitáis micro controladores, si lo único que queréis es encender una luz o dispara una alarma.

Por eso se va a proceder de la siguiente forma, en primer lugar, un pequeño circuito de prueba, de modo que nos sirva para corroborar el correcto funcionamiento de este sensor. Se encenderá un LED cuando perciba una variación en el área de cobertura.

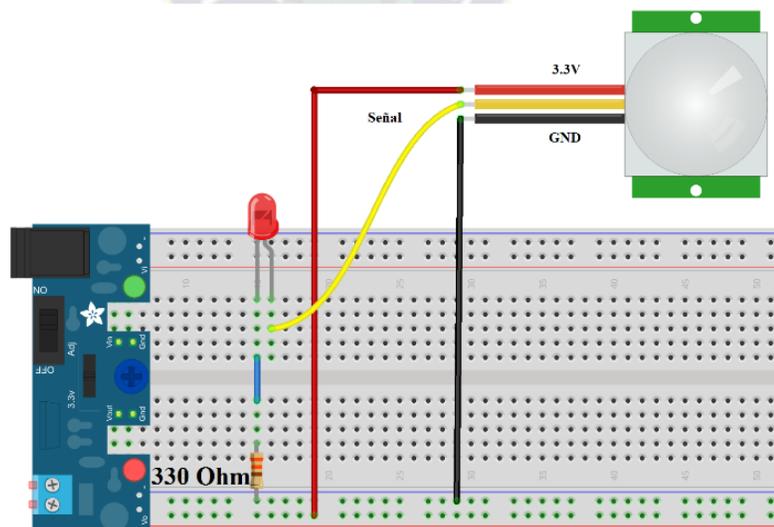


Figura 22 - Diagrama de conexión del sensor PIR

Fuente: <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work>

3.1.3.1.- Código para el sensor HC-SR501

Para el diagrama de conexión que se presentó anteriormente, y se configuro de la siguiente forma:

```
byte sensorpir = 7; // definimos pin 7 como entrada del sensor
```

```
byte led = 13;    // pin 13 led de prueba
```

```
void setup() {  
  pinMode(sensorpir,INPUT); //def. pins como I/O  
  pinMode(led,OUTPUT);  
  Serial.begin(9600);  
}  
  
void loop() {  
  if(digitalRead(sensorpir)== HIGH){  
    Serial.println("detectado movimiento por el sensor PIR");  
    digitalWrite(led,HIGH);  
    delay(1000);  
    digitalWrite(led,LOW);  
  }  
}
```

3.2.- Circuito final

Para el circuito final se efectuó la designación de los puertos, con los diferentes sensores y dispositivos de control, que el proyecto requiere, de la siguiente forma:

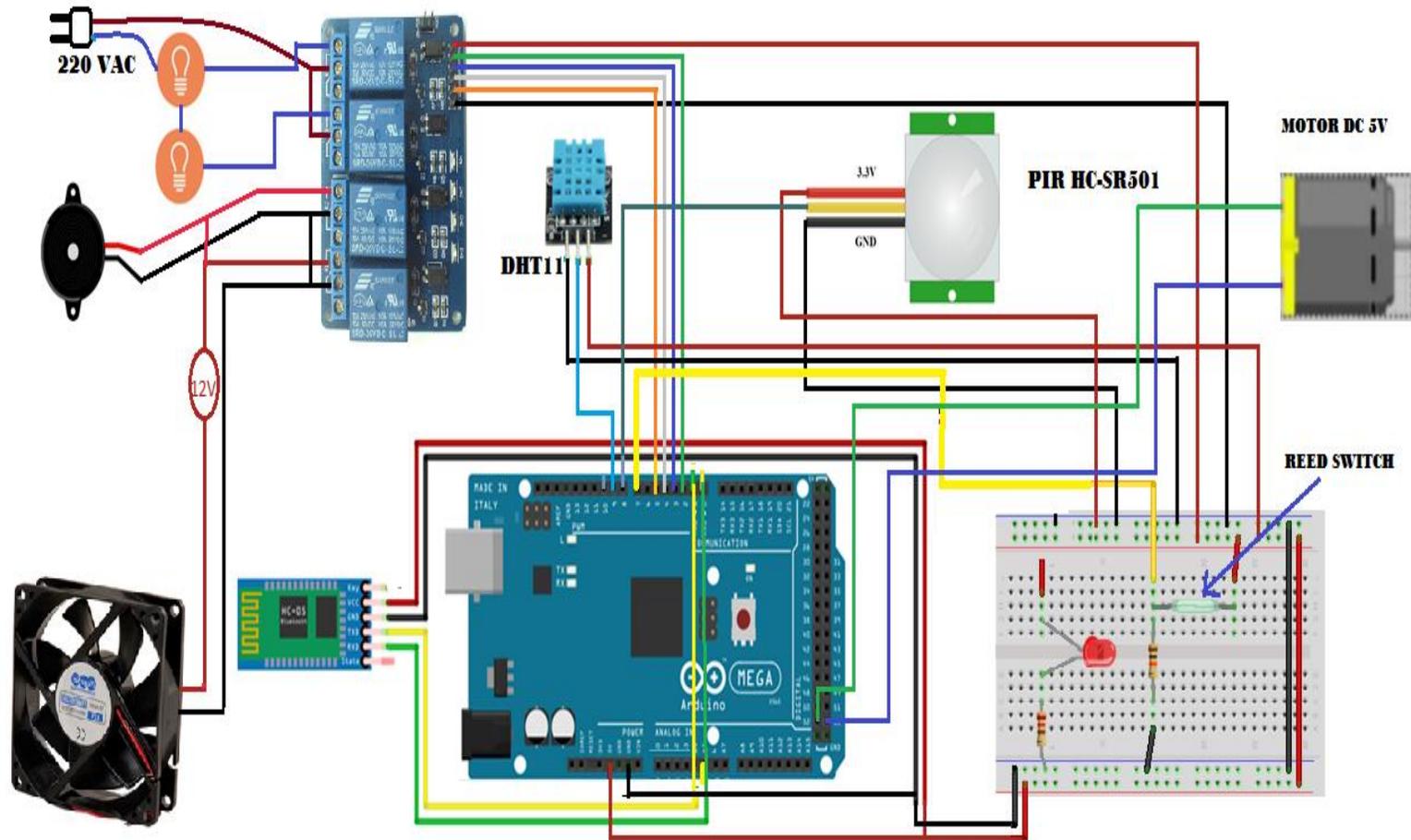


Figura 23 - Circuito final a utilizar

Fuente: propia

El circuito opera de la siguiente manera:

Permite controlar un recinto mediante sensores. El sensor HC-SH501 es un sensor que detecta el cambio de radiación, que al detectar ese cambio activa la sirena de alarma, también está el sensor magnético reed switch, que actúa como un relé normalmente abierto, misma que es mantenida por el campo magnético de un imán, que al alejarse de a una cierta distancia fuera del campo el circuito se cierra y se activa la sirena.

Cuenta con sensor de temperatura el cual acciona un ventilador cuando sobrepase los 24°C, permite controlar remotamente dos focos además de la puerta mediante un servomotor, este control se hace mediante el uso de una aplicación android en un dispositivo que presente conexión bluetooth.

3.3.- Ventajas y desventajas

Ventajas

- Contribuye en la disminución de gastos energéticos a través del control de la temperatura interna de los cuartos, la iluminación, teniendo como resultado mayor ahorro y cuidado del medio ambiente.
- Permite llevar un control sobre un recinto de forma remota.
- El sistema domótico permite integrar cualquier dispositivo que no sea inteligente al sistema.
- Ofrece seguridad personal y patrimonial
- Es alcanzable con cualquier dispositivo que tenga bluetooth.

Desventajas

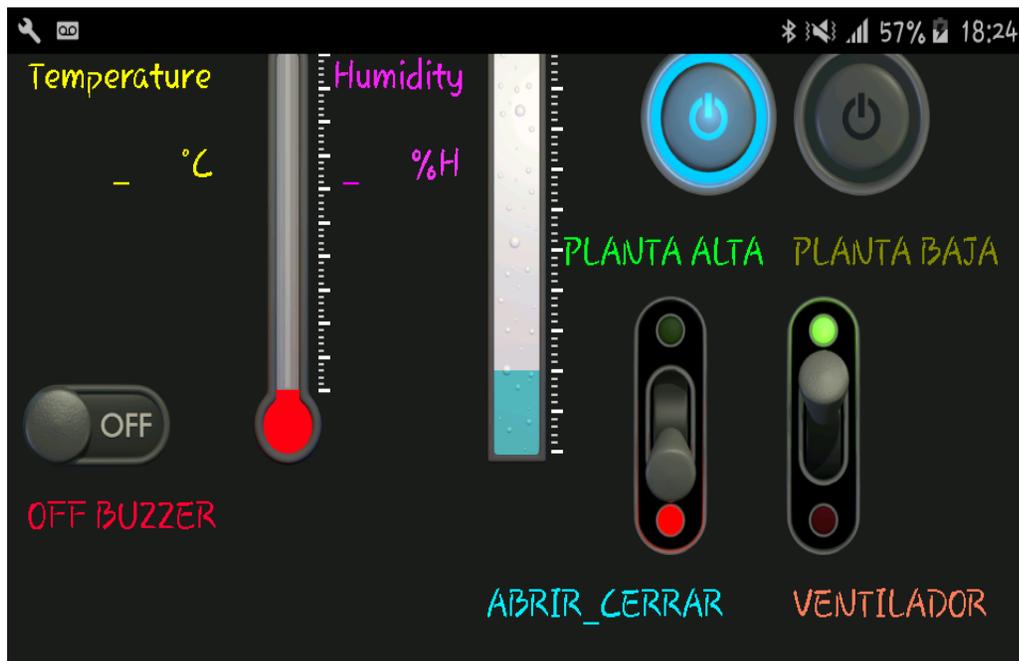
- Si el sistema de suministro de energía cae o llega a fallar, es sistema queda inhabilitada casi en su totalidad.
- Al ser bluetooth su medio de comunicación, tiene un rango limitado para su respectivo control

3.4.- Instalación del proyecto

Primeras pruebas

Para llevar a cabo esta prueba, primero se comprobó el funcionamiento de cada sensor, posteriormente se realizó el código final en arduino.

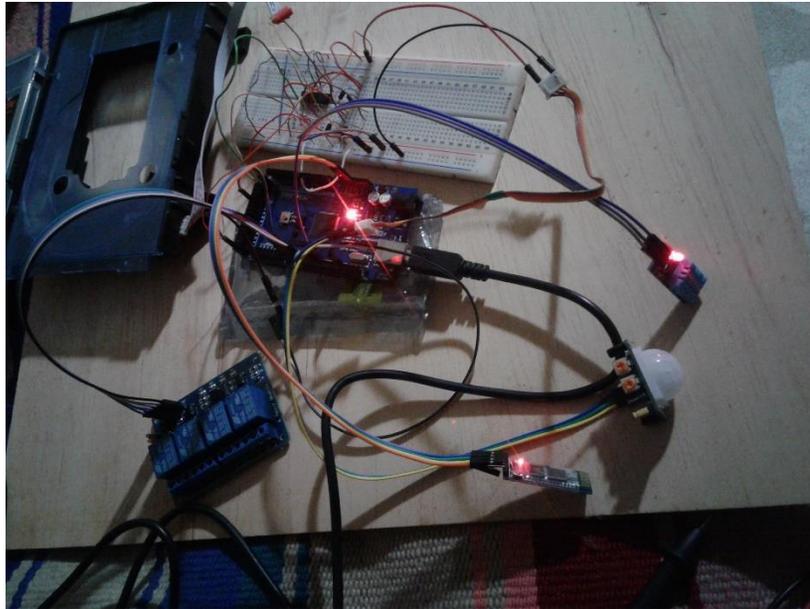
Se diseñó los controles necesarios dentro de la aplicación bluetooth electrónicas, de manera que se pueda controlar el recinto.



Fotografía 1 - Diseño de botones de control

Fuente: Propia

Se conectó todo los sensores y circuitos necesarios al arduino y se cargó el programa en el arduino se probó en un protoboard, el funcionamiento del proyecto respondió de manera satisfactoria, de acuerdo a lo planificado, como se ve en la siguiente fotografía.



Fotografía 2 - Primeras pruebas

Fuente: Propia

El siguiente paso, corresponde al armado de la maqueta, para ello se usó cajas de cartón, a estas cajas se le dan forma de modo que se pueda implementar en ella el circuito ya probado como se divisa en la siguiente fotografía.



Fotografía 3 - Cajas de cartón

Fuente: Propia

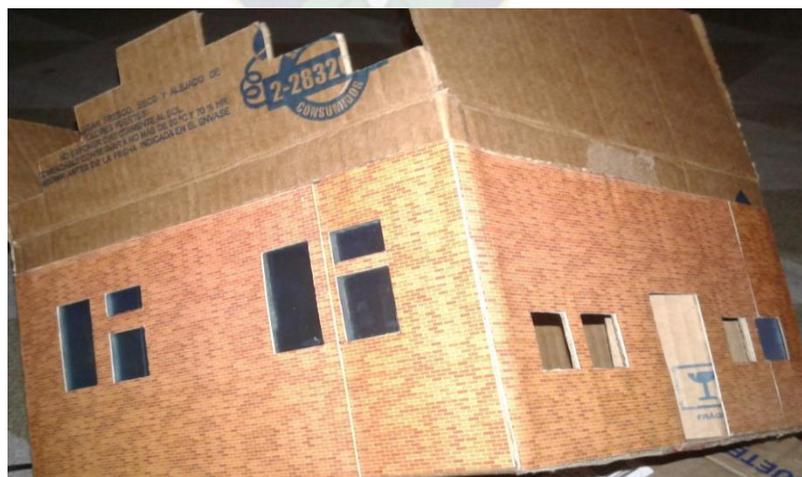
A las cajas se les va dando forma, contando de forma que tenga la forma de una casa, haciendo uso de la tijera, estilete y una regla, que será adecuado para el proyecto como se muestra en la fotografía.



Fotografía 4 - Cartón trabajado

Fuente: Propia

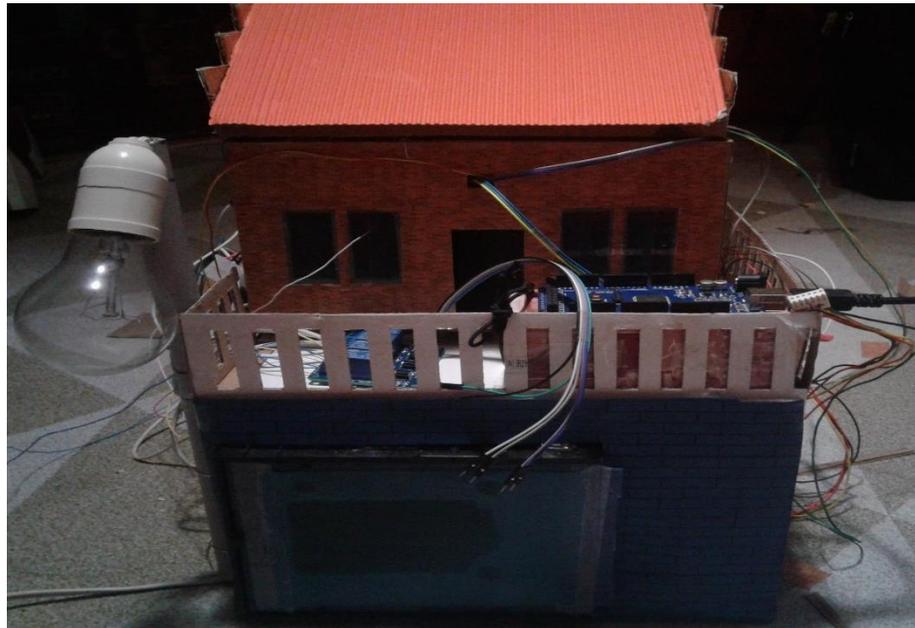
Posteriormente se hace el forado, de tal manera que se pueda parecer a una casa, como se ve la siguiente fotografía.



Fotografía 5 - Forado del cartón

Fuente: Propia

Finalmente se implementó con todos los sensores en la maqueta preparada para el prototipo del proyecto.



Fotografía 6 - Implementación final en la maqueta

Fuente: Propia

Dentro de la maqueta se tiene los sensores instalados, y se puede verificar en la fotografía siguiente:



Fotografía 7 - Vista interior de la maqueta

Fuente: Propia

3.5.-Criterio de implementación

El criterio de instalación se realizó tomando en cuenta los lugares más vulnerables de una casa o un ambiente, como son las ventanas, donde se implementó el sensor magnético reed switch, la presencia de una persona dentro de un ambiente restringido, mediante un sensor de movimiento HC-SH501, y la temperatura del ambiente y mantener a un nivel deseado como también el control de luz de los ambientes remotamente mediante un celular por el bluetooth.

3.6.- Código de programa final

```
#include <DHT11.h>
#define RELAY_ON //el relé se activa con bajo
#define RELAY_OFF
int sensT=9;
DHT11 dht11(sensT);

byte sensorpir = 8;
byte led = 13;

const int pinSwitch = 7; // el numero de pin del reed switch
const int pinLed = 12; // n° pin del led indicador de reed switch
int estadoSwitch = 0; // para leer y guardar el estado del reed switch

int estado=0; //variable para leer el estado de BLUETOOTH
int band1=0;
int band2=0;

const int m1=49; //definiendo pin`s
const int m2=53; //de control
```

```

const int ledmot=50;    // led

void setup()
{
  digitalWrite (2, RELAY_OFF); //Asegurar el relé
  digitalWrite (3, RELAY_OFF);
  digitalWrite (4, RELAY_OFF);
  digitalWrite (5, RELAY_OFF);

  pinMode (2, OUTPUT);    //Definir los pines como salida
  pinMode (3, OUTPUT);
  pinMode (4, OUTPUT);
  pinMode (5, OUTPUT);

  pinMode(sensorpir,INPUT);
  pinMode(led,OUTPUT);
  digitalWrite(led,LOW);

  pinMode(pinLed, OUTPUT);    // inicializa el pin del led como de salida
  pinMode(pinSwitch, INPUT); // inicializa el pin del reed switch como de entrada
  digitalWrite(pinLed, LOW);
  pinMode(m1,OUTPUT); //def pin como
  pinMode(m2,OUTPUT); //salida
  pinMode(led,OUTPUT);
  Serial.begin(9600); //inicia la comunicacion serial
  while (!Serial) {
    ;
  }
}

```

```

}

void loop()
{
  int err;
  float temp, humi;
  if((err=dht11.read(humi, temp))==0)
  {
    Serial.print("temperature: ");
    Serial.print(temp);
    Serial.print("°C");
    Serial.print(" humidity: ");
    Serial.print(humi);
    Serial.print("%");
    Serial.println();
    if(temp>=20){
      digitalWrite (2, RELAY_ON); //Activa relé 1
      delay (8000);
      digitalWrite (2, RELAY_OFF); //Desactiva relé 1
    }
  }
  else
  {
    Serial.println();
    Serial.print("Error No :");
    Serial.print(err);
    Serial.println();
  }
}

```

```

delay(DHT11_RETRY_DELAY); //espera para volver a intentar
if(digitalRead(sensorpir)== HIGH){
  Serial.println(detectedo movimiento por el sensor PIR");
  digitalWrite(led,HIGH);
  digitalWrite (3, RELAY_ON);    // alarma
}
estadoSwitch = digitalRead(pinSwitch);    // lee el valor del estado
if (estadoSwitch == HIGH) {              // verifica el estado
  digitalWrite(pinLed, HIGH);            // si es HIGH
  digitalWrite (3, RELAY_ON);            //ON
}
else {
  digitalWrite(pinLed, LOW);              // entonces es LOW
}

if(Serial.available()){ //Si el puerto serie esta habilitado
  estado = Serial.read(); //Lee lo que llega por el puerto Serie
  switch(estado){
    case'A':
      onfocoA();
      break;
    case'Z':
      offocoA();
      break;
    case'B':
      onfocoB();
      break;
    case'j':

```

```

offfocoB();
break;
case'V':
onvent();
break;
case't':
offvent();
break;
case'0':
abrir();
break;
case'C':
cerrar();
break;
case'K':
offbuzzer();
break;
}
}
}
void onfocoA(){
    if( 'A' && band1 == 0 ){ //ON Foco1
        digitalWrite (5, RELAY_ON);
        estado = 0;
        band1 = 1;
    }
}
void offfocoA(){

```



```

if('z' && band1 == 1){
    digitalWrite (5, RELAY_OFF);
    estado = 0;
    band1 = 0;
}
}
void onfocoB(){
    if('B' && band2 == 0){
        digitalWrite (4,RELAY_ON );
        estado = 0;
        band2 = 1;
    }
}
void offfocoB(){
    if('j' && band2 == 1){
        digitalWrite (4, RELAY_OFF);
        estado = 0;
        band2 = 0;
    }
}
void onvent(){
    digitalWrite (2, RELAY_ON);
    estado = 0;
}
void offvent(){
    digitalWrite (2, RELAY_OFF);
    estado = 0;
}

```



```

void abrir(){
  Serial.println("abrir: ");
  digitalWrite(m1,HIGH);
  digitalWrite(m2,LOW);
  digitalWrite(ledmot,HIGH);
  delay(1500);
  digitalWrite(m1,HIGH);
  digitalWrite(m2,HIGH);
  delay(20);
}

void cerrar(){
  Serial.println("cerrando: ");
  digitalWrite(m1,LOW);
  digitalWrite(m2,HIGH);
  //digitalWrite(ledmot,LOW);
  delay(1500);
  digitalWrite(m1,HIGH);
  digitalWrite(m2,HIGH);
  delay(20);
}

void offbuzzer(){
  digitalWrite (3, RELAY_OFF);
  estado = 0;
  delay(20);
}

```



3.4.- Costo total del proyecto

En la siguiente tabla se detallan todos los costos de cada componente y se obtiene el valor total del costo económico.

MATERIAL	CANTIDAD	COSTO C/U(Bs.)	COSTO(Bs.)
Placa Arduino	1	85	85
Modulo bluetooth	1	70	70
Sensor PIR	1	35	35
Sensor de temperatura humedad	1	30	30
Modulo Relé	1	42	42
Motor DC	1	8	8
Resistencias	4	0.30	1,2
Potenciómetro	1	2	2
Buzzer	1	6	6
Cooler	1	10	10
Focos	3	5	15
Sockets	3	3	9
Sócalo de 16 pines	1	1	1
CI L293D	1	18	18
Cables	Necesarios	1	10
Estaño	1	3,5	3,5
Cartón	2	2	4
Reed-switch	1	8	8
Placa pequeña	2	5	10
Total			367,7 Bs.

Tabla 1 - Costo de componentes

Fuente: Propia

CONCLUSIONES

Durante la realización del proyecto se alcanzó las siguientes conclusiones:

1. Se realizó una amplia búsqueda y depuración de información acerca de proyectos que tuvieran afinidad, que nos facilitaran la elaboración y desarrollo del sistema domótico.
2. El proyecto fue muy enriquecedor en cuanto a la adquisición de conocimiento, ya que no se tuvo mucho conocimiento sobre arduino durante el periodo de formación lo que significa un reto.
3. La implementación realizada comprobó que se puede construir una aplicación hardware para un sistema domótico a menor costo del que se encuentra un dispositivo construido por algunas de las empresas dedicadas a este mercado y solventando el mismo problema.

SUGERENCIAS

Habiendo identificado las conclusiones anteriormente, con el afán de mejorar y lograr mayor eficiencia se identifica las siguientes sugerencias:

- 1) Se sugiere que al cargar el programa al arduino se desconecte la transmisión y recepción del bluetooth, que de no ser generara un error y no se podrá cargar el programa.
- 2) Se sugiere que para tener un control desde cualquier lugar se use un módulo Wi-Fi o un módulo Ethernet.
- 3) Se sugiere no exceder la carga máxima de corriente que proporciona el arduino, a tomar muy en cuenta de lo contrario no funcionara.

BIBLIOGRAFIA

- BARBERAN VILLACAMPA Freancesc “Control Domótico de una Vivienda” Proyecto de la Facultad de Ingeniería. Universidad ROVIRA I VIRGILI De Barcelona.
- RUIZ, José Manuel, (2007), Manual de Programación Arduino, México.
- GRANADO, Emanuel, (2011), Sistemas con Radiofrecuencia, Venezuela.
- <https://www.arduino.cc/en/Main/arduinoBoardMega>
- <http://es.wikipedia.org>
- <http://es.google.com>
- Bluetooth. Sitio web Disponible en: <http://tecbolivia.com/index.php/venta-de-componentes-electronicos-11/comunicaciones/modulo-serial-bluetooth-hc-05-maestro-esclavo-detail>
- Sensor Magnético Disponible en: http://www.chilealarmas.cl/wpcontent/uploads/2014/01/355554737_688.jpg
- <https://github.com/adafruit/DHT-sensor-library>
- <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work>
- <http://dinastiatecnologica.com/producto/modulo-rele-de-4-canales-arduino/>