

UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD DE TECNOLOGIA
CARRERA: ELECTRONICA Y TELECOMUNICACIONES



NIVEL LICENCIATURA
EXAMEN DE GRADO
TRABAJO DE APLICACIÓN

**"SISTEMA DE CONTROL ACTIVADO VIA
BLUETOOTH PARA ALARMA VECINAL Y
DOMICILIARIA"**

POSTULANTE: Luis Ricardo León Flores

La Paz - Bolivia
Noviembre - 2015

Dedicatoria y Agradecimientos

Dedico a Dios Todopoderoso que es el centro de mi vida y a mi familia por apoyarme e inculcarme la idea de que las cosas se consiguen con trabajo y con perseverancia, que sin esta forma de ver las cosas quizá no habría sido capaz de culminar esta Carrera que me brindo una formación con la que podré seguir superándome.

Asimismo, me gustaría agradecer a mis mentores los Docentes y a todas las personas que creyeron en mí y me brindaron su apoyo sin esperar nada a cambio.

¡Gracias!

INDICE

RESUMEN DEL PROYECTO

CAPITULO I INTRODUCCION

1.1 PLANTEAMIENTO DEL PROBLEMA.....	2
1.2 OBJETIVOS.....	3
1.2.1 Objetivo general.....	3
1.2.2 Objetivos específicos.....	3
1.3 JUSTIFICACION.....	3

CAPITULO 2 MARCO TEORICO

2.1 CONCEPTO DE ALARMA - ALARMA VECINAL	5
2.2 RECONOCIMIENTO DEL HABLA O VOZ.....	5
2.3 EL ARDUINO.....	6
2.3.1 Características Básicas de Arduino UNO.....	6
2.3.1.1 Hardware.....	7
2.3.1.2 Arduino UNO.....	7
2.3.1.2.1 Elementos que Componen la Placa Arduino.....	8
2.3.1.3 Software.....	13
2.3.1.4 Dispositivos Acoplables a Arduino.....	14
2.3.1.4.1 Sensores	15
2.3.1.4.2 Actuadores	16
2.3.1.4.3 Interfaces.....	17
2.3.1.4.4 Comunicadores (módulos anexos al arduino)..	17
2.3.1.4.5 Diagrama de Pines del Arduino Uno.....	20
2.4 TECNOLOGÍA BLUETOOTH.....	21
2.4.1 Historia del Bluetooth.....	22
2.4.2 Tipos de Bluetooth.....	23
2.5 ANDROID	24

CAPITULO 3 INGENIERIA DEL PROYECTO

3.1 RELEVAMIENTO DE SISTEMAS DE ALARMAS Y VIGILANCIA

ACTUALES EN LA CIUDAD DE LA PAZ 26

3.2 MARCO GEOGRÁFICO 28

3.3 FUNCIONAMIENTO DEL "SISTEMA DE CONTROL ACTIVADO VIA

BLUETOOTH PARA ALARMA VECINAL Y DOMICILIARIA" 29

3.4 MATERIALES 30

3.5 DESARROLLO

3.5.1 Proceso de Implementación Arduino - Módulo de Reles..... 31

3.5.2 Módulo Bluetooth y su Diagrama de Conexión..... 33

3.5.3 Comandos AT 35

3.5.4 Programa en IDE del Arduino del Proyecto "SISTEMA DE CONTROL
ACTIVADO VIA BLUETOOTH PARA ALARMA VECINAL Y DOMICILIARIA

..... 36

3.5.5 Recepción de un Dispositivo Android 40

3.5.6 Configurar la APP Android 42

3.6 COSTOS..... 46

3.6.1 Costos por Equipos y Materiales del Sistema 46

CAPITULO 4 CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES 47

4.2 RECOMENDACIONES 48

CAPITULO 5 REFERENCIAS BIBLIOGRÁFICAS

5.1 Paginas WEB 49

RESUMEN DEL PROYECTO

El proyecto "**SISTEMA DE CONTROL ACTIVADO VIA BLUETOOTH PARA ALARMA VECINAL Y DOMICILIARIA**" permite a los vecinos en general tener una herramienta de prevención, persuasión y comunicación, para indicar la presencia de antisociales, previo análisis de la situación (si muestra actitud sospechosa, si está realizando actos delincuenciales, etc.), para que por medio de la aplicación instalada en uno o varios celulares, a una distancia larga de los dispositivos, vía bluetooth activar una alarma y/o las luces que indicaras la emergencia y solicitara la presencia de los vecinos.

El Sistema se basa en la Plataforma Arduino UNO r2 que es de costo bajo que se complementa con un módulo bluetooth el cual esta conectado a un dispositivo móvil que cuenta con dicha aplicación y que la comunicación entre los componentes del proyecto (Arduino-Modulo Bluetooth-Celular),es a través del IDE ARDUINO y la Aplicación Inventor2.

CAPITULO I INTRODUCCION

La inseguridad y la delincuencia en las calles de las ciudades de La Paz y El Alto están al orden día, esto nos muestra las estadísticas elaboradas al efecto, donde señalan que cada mes aproximadamente 1.000 casos de violencia y delincuencia se dan en la ciudad de La Paz en un promedio de 30 hechos por día, en la ciudad de El Alto a diario se producen 20, en la ciudad de Santa Cruz 25 y en Cochabamba 15. Del total de estos hechos delictuosos y violentos traducidos en robos, asaltos a mano armada, raptos, peleas, golpizas a menores y agresiones de pandillas juveniles que se dan en nuestras ciudades, el 60 por ciento ocurren en horas de la noche.

Cada día se cometen al menos cuatro casos de robos, hurtos o atracos en el municipio de La Paz. Estos delitos causan temor en 20 de sus 23 distritos. De los 6.306 casos atendidos el año pasado por la Fuerza Especial de Lucha Contra el Crimen (FELCC) en la sede de gobierno, 1.491 fueron “delitos contra la propiedad” (que comprenden robo, hurto y robo agravado, a personas e inmuebles). Es el mayor índice, seguido por el de “delitos contra la corrupción pública”, con 1.361 y el de “delitos económico financieros”, con 1.011.

Al respecto, el actuar de los antisociales es cada vez más osado, agresivo y violento, las casas y la vecindad en general no quedan exentas de este mal, estableciéndose como una necesidad la prevención. Si bien las Autoridades y la ciudadanía en general aplican métodos de seguridad que van desde lo técnico hasta lo jurídico para prevenir la delincuencia en nuestras ciudades y que existen empresas especializadas que brindan sistemas de vigilancia contando con equipos que van desde una simple cámara hasta los más complicados, haciendo uso de dispositivos muy complejos y caros.

Es ahí donde este Proyecto de bajo costo, útil y muy practico, se convierte en una herramienta de persuasión y vigilancia por medio de una alarma y luces a objeto de precautelar la integridad física de las personas y de los que lo rodean, su

manipulación es a una distancia considerable (10 a 15m es la cobertura que tienen los dispositivos Bluetooth) manipulan el encendido y apagado de las luces y la alarma que se encuentran instaladas en lugares estratégicos, mediante el uso de un celular vía bluetooth (7 de 10 personas que cuentan con un celular, tienen la aplicación Bluetooth), la que se conecta al sistema de control que está en base a una placa Arduino, en la que nos apoyaremos para poder construir, se podría decir un sistema domótico simple. Y a propósito la domótica se podría definir como el conjunto de tecnologías aplicadas al control y la automatización inteligente de la vivienda, que permite una gestión eficiente del uso de la energía además de aportar seguridad, confort, y comunicación entre el usuario y el sistema. Arduino es una plataforma de hardware libre basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios, en este caso para nuestras necesidades apoyado en la comunicación con un aparato celular que cuente con una conexión vía bluetooth.

1.1 PLANTEAMIENTO DEL PROBLEMA

La inseguridad y delincuencia siempre ha sido un problema que ha afectado a la sociedad y que al pasar del tiempo sus índices se incrementaron principalmente en las casas y por ende en los Barrios de nuestra ciudad, el actuar criminal de estos malhechores al amparo de la oscuridad que ingresan a las viviendas para cometer actos delictivos, que en el peor de los casos llegan a agredir físicamente e incluso matar a las personas que tienen contacto con los mismos.

Existen Empresas de seguridad de alarmas y vigilancia privadas que brindan sus servicios principalmente a las viviendas que son bastante elevados y no dan avisos inmediatos al usuario. Asimismo, en el mercado hay dispositivos de vigilancia caseros que en muchos de los casos no satisfacen las necesidades de seguridad.

1.2 OBJETIVOS

1.2.1 OBJETIVO GENERAL

Implementar un Sistema de Control activado por voz, para alarma vecinal y domiciliaria, usando Bluetooth y Android.

1.2.2 OBJETIVOS ESPECÍFICOS

- **Analizar los Comandos y Módulos de conexión**
- **Programar el Arduino**
- **implementar la comunicación vía Bluetooth**
- **Desarrollar el software en Android**
- **Implementar un Prototipo (maqueta) del Sistema de Control**

1.3 JUSTIFICACIÓN

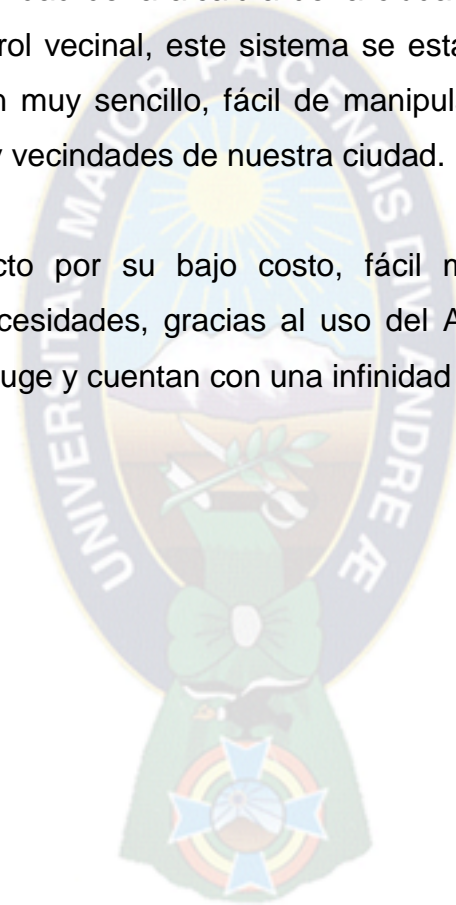
La necesidad de implementar este Proyecto radica en la evidencia de que es necesario tomar conciencia de contar con herramientas que ayuden a la vigilancia de los hogares y barrios para prevenir la inseguridad y actos delincuenciales hechos por personas malhechoras y que es tarea de todos los vecinos y familias participar de la misma (ayudará a que los vecinos se conozcan); es ahí donde este sistema se convierte en una herramienta que ayuda a realizar las tareas descritas con la participación de todos...

Este Proyecto garantiza la seguridad física de la persona que visibiliza un acto sospechoso o delincencial, porque la activación de las luces o la alarma de emergencia, se la realiza a una distancia de hasta 15 m de distancia del equipo instalado (distancia que abarca la plataforma bluetooth), por medio de la voz o el tacto, a través del dispositivo móvil que contara con la aplicación bluetooth(en Bolivia

7 de 10 personas cuentan con un dispositivo móvil), teniendo la ayuda y presencia de los vecinos que se encuentren alrededor. Asimismo, promueve la participación vecinal que sirve para prevenir delitos y otros eventos de emergencia en cada barrio y sirve como mecanismo de disuasión para ahuyentar a algún sospechoso o para recibir ayuda de la Policía o del resto de los vecinos.

Este proyecto se complementa a los sistemas de vigilancia ya existentes tal el caso de las cámaras de seguridad de la alcaldía de la ciudad de La Paz y si no cuentan con ningún tipo de control vecinal, este sistema se establece como herramienta de prevención y persuasión muy sencilla, fácil de manipular pero eficaz y útil, para la seguridad de las casas y vecindades de nuestra ciudad.

Se realiza este Proyecto por su bajo costo, fácil manipulación y modificación constante según las necesidades, gracias al uso del Arduino y de las tecnologías celulares que están en auge y cuentan con una infinidad de facilidades.



CAPITULO 2 MARCO TEORICO

2.1. CONCEPTO DE ALARMA - ALARMA VECINAL

Se entiende por alarma la señal o aviso que advierte sobre la proximidad de un peligro. El aviso de alarma informa a la comunidad en general o a una entidad específica (como ser los bomberos o la policía) que deben seguir ciertas instrucciones de emergencia dado que se ha presentado una amenaza.

Las alarmas Vecinales, son una herramienta para brindar la seguridad y la convivencia, pero su efectividad está en función de la organización de los vecinos que respondan a las sirenas, prendan luces o denuncien los hechos.

2.2. RECONOCIMIENTO DEL HABLA O VOZ

Un Sistema de reconocimiento de voces es una herramienta computacional capaz de procesar la señal de voz emitida por el ser humano y reconocer la información contenida en esta, convirtiéndola en texto emitiendo ordenes que actúan sobre un proceso. En su desarrollo intervienen diversas disciplinas tales como la fisiología, la acústica, lingüística, procesamiento de señales, inteligencia artificial y otros.

Los usos más frecuentes del reconocimiento del habla, son agentes telefónicos que interactúan con el usuario únicamente mediante la voz, juguetes, robots, y teléfonos inteligentes.

Los principales actores comerciales en el campo del reconocimiento del habla ofrecen soluciones hardware en forma de chips dedicados al reconocimiento del habla que pueden ser empotrados en otros elementos (juguetes, robots, etc.). Sensory Inc. es la empresa detrás de los sistemas de reconocimiento de voz de los últimos teléfonos inteligentes de Samsung.

2.3. EL ARDUINO

Arduino es una plataforma de prototipos de código abierto (open-source) basada en hardware y software flexible y fácil de usar que nació en Italia en el año 2006, basado principalmente en un microcontrolador ATMEGA. Tanto los diseños del hardware como el código de su IDE están disponibles bajo licencias de tipo código abierto.

El Arduino como otras plataformas similares oculta y simplifica los detalles de programación de los microcontroladores en los que se basan a la vez que proporcionan interfaces de entrada/salida más comunes como puertos serie o puertos USB. De esta forma, estas plataformas hacen más sencillo y asequible el iniciarse en el uso de estos pequeños ordenadores.

Una de las características más importantes de Arduino es la disposición estándar de los pines. Esta disposición estándar permite conectarlo a otros módulos que aportan nuevas características al controlador. A los módulos que integran esta disposición estándar se les llama "Shields".

2.3.1. CARACTERÍSTICAS BÁSICAS DE ARDUINO UNO

Las características básicas que esta compuesta el Arduino UNO, se detalla a continuación:

2.3.1.1. HARDWARE

Al ser Arduino una plataforma de hardware libre tanto su diseño como su distribución puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin haber adquirido ninguna licencia. Por eso existen varios tipos de placa oficiales, las creadas por la comunidad Arduino o las no oficiales creadas por terceros pero con características similares. En la placa Arduino es donde se conecta los módulos,

sensores, actuadores y otros elementos necesarios para comunicarnos con el sistema.

2.3.1.2. ARDUINO UNO

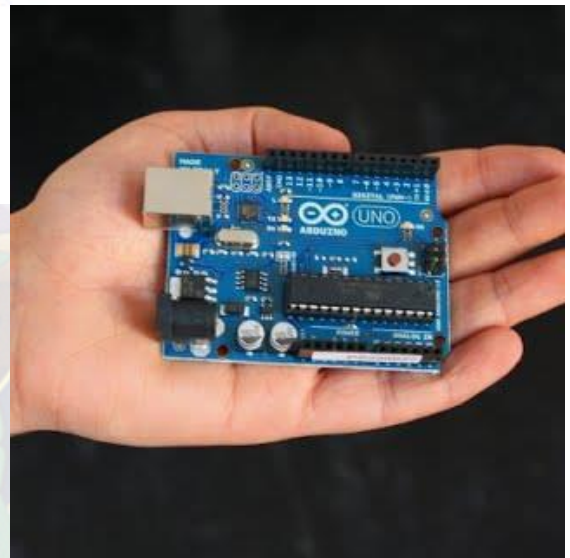


Fig. 1 Arduino UNO

La placa del Arduino UNO tiene un tamaño de 75x53mm, su unidad de procesamiento consiste en un microcontrolador ATmega328. Puede ser alimentada mediante USB o alimentación externa (7 – 12 V) y contiene pines tanto analógicos como digitales:

Microcontrolador	ATmega328
Voltaje operativo	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (limites)	6-20V
Pines digitales E/S	14 (de los cuales 6 proporcionan salida PWM)
Pines de entrada analógica	6
Corriente continúa para pines E/S	40 mA
Memoria Flash	32 KB (ATmega328) de los cuales 0.5 KB son para el bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Velocidad del reloj	16 MHz

Cuadro. Características

2.3.1.2.1 ELEMENTOS QUE COMPONEN LA PLACA ARDUINO UNO.

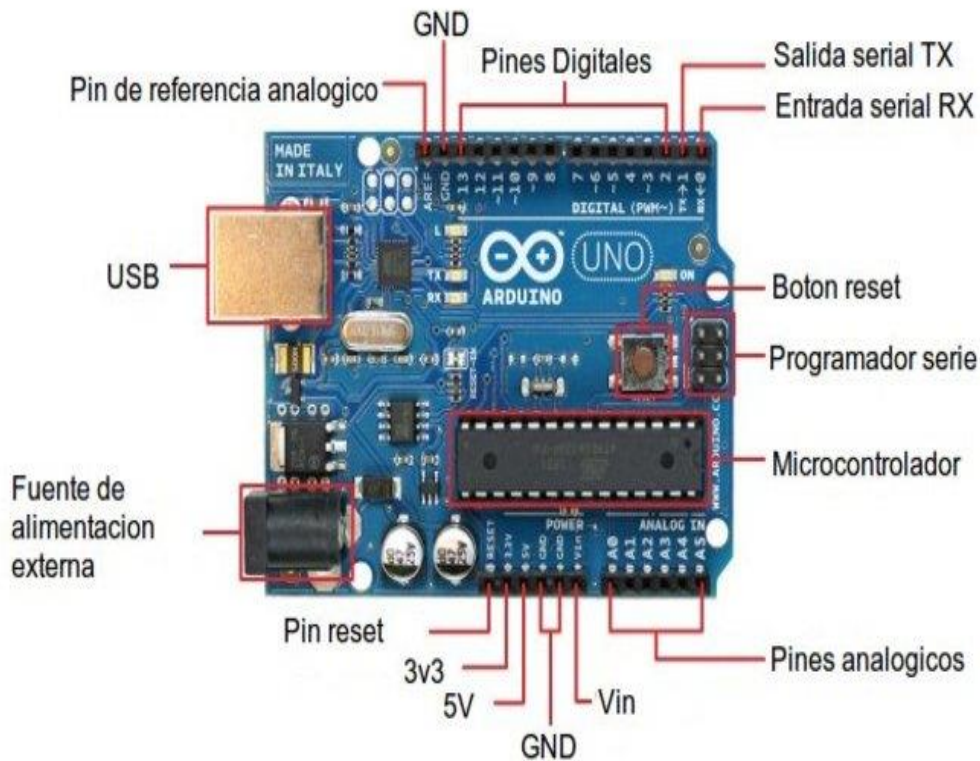


Fig. 2 Elementos del Arduino

A continuación detallaremos los elementos de la placa Arduino UNO:

CONECTOR USB

Existen varios tipos de conectores USB, pero la que utilizaremos será el tipo B hembra. Con lo cual se necesitará un cable tipo B macho – tipo A macho que deberá conectarse a un conector tipo A hembra. La placa se puede alimentar con la tensión de 5V que le proporciona el bus serie USB. Cuando carguemos un programa a la placa desde el software de Arduino se inyectará el código del ordenador por este bus.

FUENTE DE ALIMENTACIÓN EXTERNA

La placa puede ser alimentada también mediante corriente continua suministrada por el conector jack de 3.5mm que podrá recibir entre 7 y 12V.

PIN DE RESET

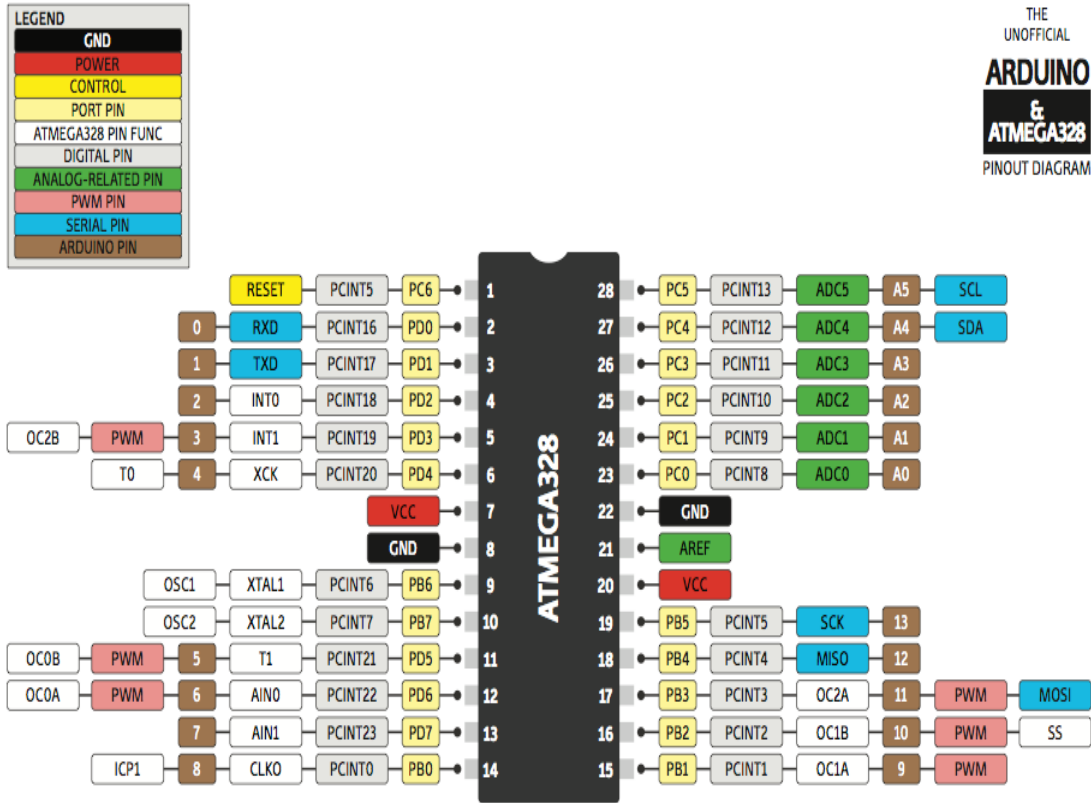
Podemos imitar el funcionamiento del botón reset suministrando un valor LOW (0V) para reiniciar el microcontrolador.

MICROCONTROLADOR ATMEGA328

Un **microcontrolador** es un sistema que integra en un mismo circuito integrado una unidad de procesamiento, memoria y pines para la entrada y salida de datos. Se puede pensar en un microcontrolador como un pequeño ordenador.

El cerebro del Arduino es el Atmega328 AVR 8-bit que es un Circuito integrado de alto rendimiento que está basado un microcontrolador RISC, combinando 32 KB ISP flash una memoria con la capacidad de leer-mientras-escribe, 1 KB de memoria EEPROM, 2 KB de SRAM, 23 líneas de E/S de propósito general, 32 registros de proceso general, tres temporizadores flexibles/contadores con modo de comparación, interrupciones internas y externas, programador de modo USART, una interface serial orientada a byte de 2 cables, SPI puerto serial, 6-canales 10-bit

Convertor A/D (8-canales en TQFP y QFN/MLF packages), "watchdog timer" programable con oscilador interno, y cinco modos de ahorro de energía seleccionables por software. El dispositivo opera entre 1.8 y 5.5 voltios. Por medio de la ejecución de poderosas instrucciones en un solo ciclo de reloj, el dispositivo alcanza una respuesta de 1 MIPS, balanceando consumo de energía y velocidad de proceso.



THE UNOFFICIAL
ARDUINO
&
ATMEGA328
PINOUT DIAGRAM

www.pighox.com
CC BY-NC-SA
29 JAN 2013

Fig. 3 Pines del Microcontrolador ATMEGA-328

PARÁMETROS del ATMEGA328

PARÁMETROS	VALORES
Flash	32 Kbytes
SRAM	2 Kbytes
Cantidad Pines	28
Frecuencia máxima de operación	20 MHz
CPU	8-bit AVR
Número de Canales variables	16
Pines máximos de E/S	26
Interrupciones internas	24

Un remplazo alternativo común al ATmega328 es el ATmega328P.

PINES DIGITALES DE ENTRADA Y SALIDA E/S

En estos pines se conectan a los sensores, módulos, etc. Hay 14 pines digitales que pueden utilizarse como entrada o salida con las funciones `pinMode`, `digitalWrite` y `digitalRead`. Operan a 5 voltios, cada pin proporciona o recibe como máximo 40mA y disponen de una resistencia pull-up (desconectada por defecto) de 20-50 kOhmios.

Entre los pines digitales E/S, se tienen los siguientes:

- Serie: 0(RX) y 1(TX). Utilizados para recibir (RX) y transmitir (TX) datos serie. Están directamente conectados a los pines serie del microcontrolador. Utilizando estos pines podremos conectarnos con otras placas.

- Interrupciones externas: 2 y 3. Estos pines pueden ser configurados para activar interrupciones.
- PWM: 3, 5, 6, 9, 10 y 11. Proporcionan una salida de 8 bits en modo PWM.
- SPI: 10-13. Estos pines soportan la librería de comunicación de dispositivos SPI.
- LED: 13. Este pin está conectado con un led de la placa. Cuando se le asigne un valor HIGH se encenderá, en cambio si lo dejamos en LOW estará apagado.

RESET

Utilizando este botón podremos reiniciar la ejecución del código del microcontrolador.

AREF (REFERENCIA PARA PINES ANALÓGICOS)

Tensión de referencia para entradas analógicas. Se utiliza con la función analogReference.

GND (PINES DE TIERRA)

Tensión de referencia de 0 V.

ICSP (IN CIRCUIT SERIAL PROGRAMMING)

Es un conector utilizado en los dispositivos PIC para programarlos sin necesidad de tener que retirar el chip del circuito del que forma parte.

PIN DE 3.3V

Sirve para suministrar 3.3V a los dispositivos que lo necesiten con una corriente máxima de 50mA. Es generada gracias al chip FTDI integrado en la placa.

PIN DE 5V

Este pin saca una tensión de 5v del regulador de la placa. El regulador es necesario puesto que puede ser alimentada con distintos voltajes.

PIN DE 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.

PINES ANALÓGICOS

Esta placa contiene 6 pines de entrada analógicos. Los elementos que se conecten aquí suelen tener mayor precisión que los digitales pero su uso requiere de una lógica levemente mayor.

2.3.1.3. SOFTWARE

Puesto que Arduino, a diferencia de un ordenador, no tiene pantalla ni teclado, se necesita un programa externo ejecutado en otro ordenador para poder escribir programas para la placa Arduino. Éste software es lo que llamamos **Arduino IDE**. IDE significa “Integrated Development Environment” (Entorno de Desarrollo Integrado), y es un término común para llamar a este tipo de desarrollo de software. Escribes tu programa en el IDE, lo cargas en el Arduino, y el programa se ejecutará en la placa.

El entorno de desarrollo de Arduino es sencillo e intuitivo además puede descargarse gratuitamente desde su página oficial para distintos sistemas operativos. Ha sido implementado con Processing, un lenguaje similar a Java.



```
programa_arduino_143
Archivo Editar Programa Herramientas Ayuda
programa_arduino_143
programa_arduino_143

#include <Arduino.h> //Librería que permite establecer comunicación entre el arduino y el pc
//Definición de constantes (se usa para definir el número de pines de salida)
#define LED_PIN 13 //El pin 13 es el pin de salida

//Definición de constantes (se usa para definir el número de pines de salida)
#define LED_PIN 13 //El pin 13 es el pin de salida

int led = LED_PIN; //Define el pin de salida

void setup()
{
  pinMode(LED_PIN, OUTPUT); //Define el pin de salida como salida
}

void loop()
{
  //Cuando haya datos disponibles
```

PROCESSING es un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java, de fácil utilización, y que sirve como medio para la enseñanza y producción de proyectos multimedia e interactivos de diseño digital. Fue iniciado por Ben Fry y Casey Reas a partir de reflexiones en el Aesthetics and Computation Group del MIT Media Lab dirigido por John Maeda.

Se distribuye bajo la licencia GNU GPL y al estar basado en Java, puede heredar todas sus funcionalidades, convirtiéndose en una herramienta poderosa a la hora de encarar proyectos complejos.

Está formado por una serie de menús, una barra de herramientas con botones para las funciones comunes, un editor de texto donde se escribe el código, un área de mensajes y una consola de texto.

2.3.1.4. DISPOSITIVOS ACOPLABLES A ARDUINO

Los Dispositivos que pueden proporcionarnos algunas tareas para propósitos específicos en el uso del Arduino, pueden ser muchos, los cuales estarán conectados mediante cables o directamente acoplados, muchos disponen de librerías que deberemos adjuntar al programa para poderlos usar

Para utilizar los métodos de sensores y actuadores digitales debemos tener en cuenta que solo tenemos dos posibles valores, HIGH representa el nivel alto y LOW el nivel bajo.

En el caso de los analógicos su uso es levemente más complejo pero también más configurable ya que tiene que leerse/escribir un voltaje de 0 a 5 voltios que se representa en 10 bits (lectura) o en 8 bits (escritura), es decir la tensión puede tener 1024 (lectura) o 256 (escritura) valores distintos.

2.3.1.4.1. SENSORES

Un sensor es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. A continuación se describirán algunos sensores:

MÓDULO DE GAS



MÓDULO PIR



MÓDULO DE LUZ



MÓDULO DE TEMPERATURA



MÓDULO DE HUMEDAD (Y TEMPERATURA)

Fig. 4 Módulos Arduino

2.3.1.4.2. ACTUADORES

Un actuador es un dispositivo capaz de transformar energía en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado:

MÓDULO RELÉ



Este Módulo es la que usaremos para implementar el Proyecto, 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. Este tipo de módulos permite activar actuadores como por ejemplo la puerta del garaje o el de una bombilla.

MÓDULO LED



MÓDULO TIMBRE



MÓDULO PULSABLE



MÓDULO VISUALIZADOR



Fig. 5 Módulos complementarios al Arduino

2.3.1.4.3. INTERFACES

Su principal objetivo es ofrecer comunicación entre el sistema y el ser humano. Consisten en elementos visuales/auditivos que avisan de eventos y podemos mencionar los siguientes;

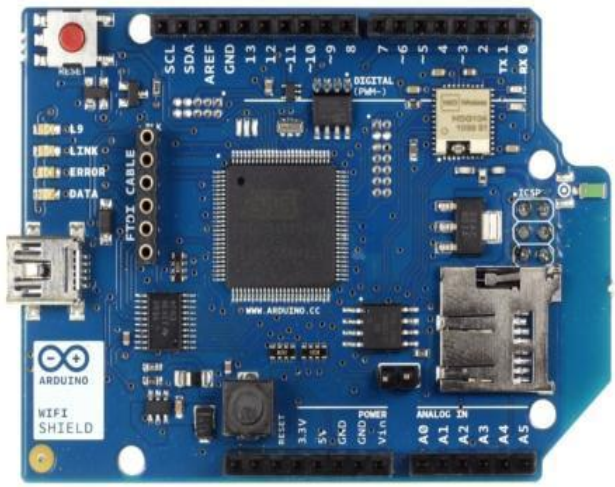
2.3.1.4.4. COMUNICADORES (MODULOS ANEXOS AL ARDUINO)

Son un conjunto de elementos que permiten la comunicación entre distintas placas Arduino y el servidor. El medio por el que circula la información puede ser por aire (modulación de ondas electromagnéticas) o físico (por cable) teniendo sus ventajas e inconvenientes. Normalmente estos dispositivos tendrán a nuestra disposición librerías con funciones ya implementadas que nos facilitará su manejo.

MÓDULO ETHERNET



MÓDULO WI-FI



MÓDULO XBEE

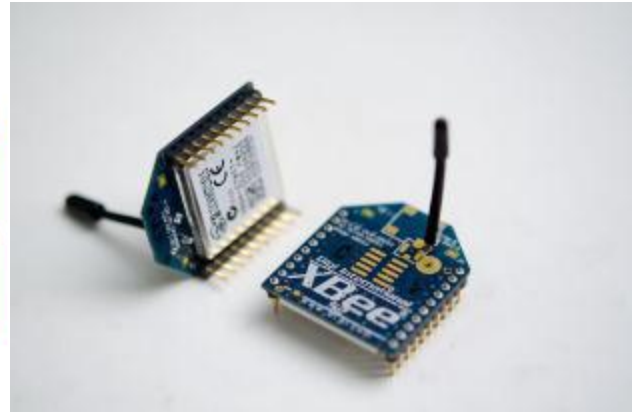


Fig. 6 Módulos Anexos

MÓDULO BLUETOOTH



Se denomina Bluetooth al protocolo de comunicaciones diseñado especialmente para dispositivos de bajo consumo, que requieren corto alcance de emisión y basados en transceptores de bajo costo. Opera mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz. Su uso es adecuado cuando puede haber dos o más dispositivos en un área reducida sin grandes necesidades de ancho de banda.

Bluetooth tiene la ventaja de simplificar el descubrimiento y configuración de los dispositivos, ya que éstos pueden indicar a otros los servicios que ofrecen, lo que redundaría en la accesibilidad de los mismos sin un control explícito de direcciones de red, permisos y otros aspectos típicos de redes tradicionales.

MÓDULO INFRARROJO

Otra forma de comunicación sin medio físico es la utilización de rayos infrarrojos. Este tipo de comunicación consigue tener menos interferencias debido a la mayor frecuencia del espectro electromagnético en la que trabaja. Normalmente no es un método de comunicación entre placas Arduino sino que es utilizado para contactar con los electrodomésticos del hogar.



PUERTO SERIE

Un puerto serie o puerto serial es una interfaz de comunicaciones de datos digitales, frecuentemente utilizado por computadores y periféricos, donde la información es transmitida bit a bit enviando un solo bit a la vez, en contraste con el puerto paralelo que envía varios bits simultáneamente.

La placa Arduino tiene un buffer de recepción de datos de 128 bytes. Con la función `available`, podremos conocer cuántos caracteres (bytes) hay disponibles en el buffer. En caso de que haya datos en el buffer la función `read` devolverá el primer carácter (byte) guardado en el puerto serie y lo retirará del buffer. En caso de que no haya datos devolverá un `-1`.



Fig. 7 Puerto Serie

2.3.1.4.5. DIAGRAMA DE PINES DEL ARDUINO UNO

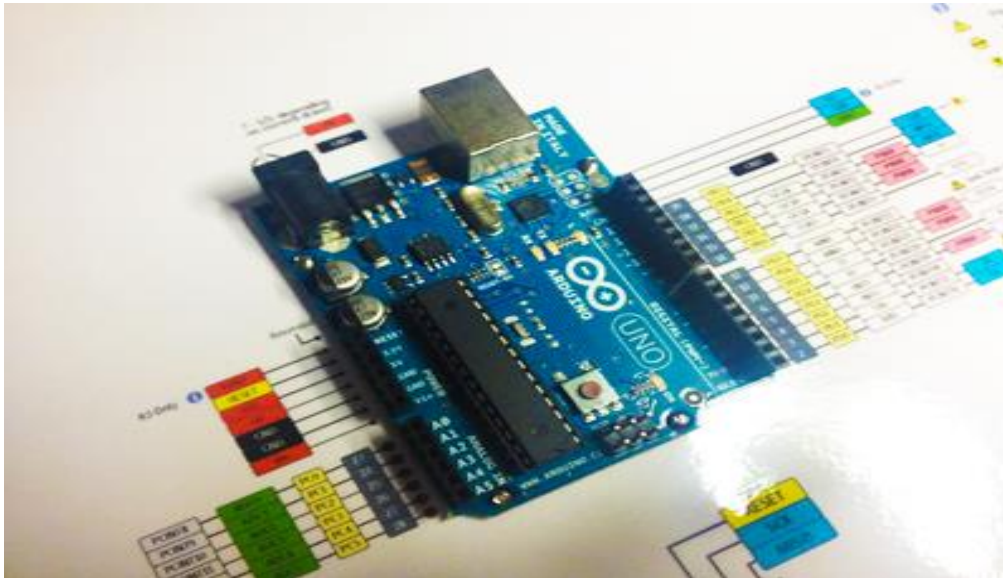


Fig. 8-A Diagrama de pines del Arduino

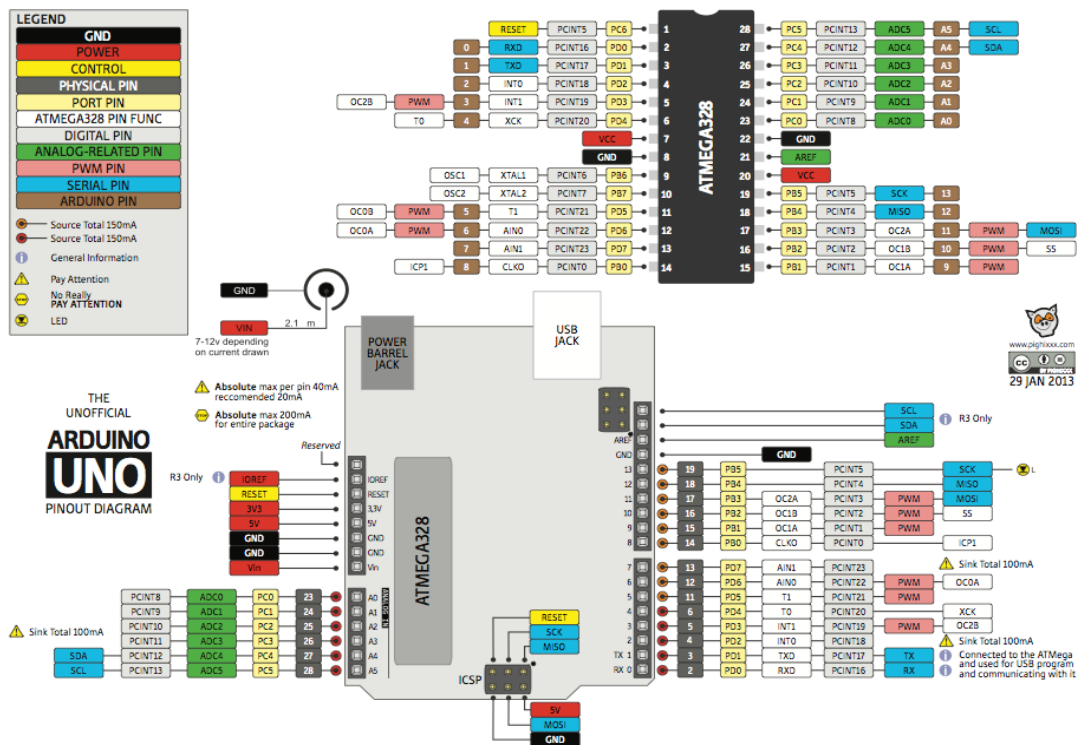


Fig. 8-B Diagrama de pines del Arduino

2.4. TECNOLOGÍA BLUETOOTH

La tecnología Bluetooth es un estándar mundial para comunicaciones inalámbricas entre dispositivos a corta distancia (hasta unos 10 m). Es el estándar utilizado para implementar las redes inalámbricas de área personal (WPAN – Wireless Personal Area Network) definidas en el IEEE 802.15.1.

La principal característica de las redes WPAN es la simplicidad y el bajo coste, debido principalmente a que son redes Ad Hoc que no necesitan de infraestructura alguna para operar y debido a las bajas potencias con las que se emiten las ondas de radiofrecuencia.

Los dispositivos Bluetooth se clasifican en 3 clases en función de la potencia emitida y del alcance de la señal.

Clase	Potencia máxima permitida	Potencia máxima permitida	Alcance (aproximado)
Clase 1	100 mW	20 dBm	~30 metros
Clase 2	2.5 mW	4 dBm	~10-5 metros
Clase 3	1 mW	0 dBm	~1 metro

Cuadro 1. Clases de dispositivos Bluetooth

2.4.1. HISTORIA DEL BLUETOOTH

- Creado por Ericsson en 1994, originariamente fue concebido como alternativa inalámbrica a la comunicación por cables de datos RS-232.
- En 1998, se crea el organismo Bluetooth Special Interest Group (SIG) formado por cinco compañías – Ericsson, Intel, Nokia, Toshiba e IBM, con el objetivo de promover desarrollar esta tecnología.
- En 1999, se produce el lanzamiento de la especificación Bluetooth 1.0.
- En el año 2000, se fabrica el primer chip que integra radiofrecuencia, banda base, y software de tecnología Bluetooth. Debido a este hecho, este mismo año sale al mercado el primer teléfono móvil con tecnología Bluetooth.
- En 2002, IEEE aprueba la especificación 802.15.1 que conforma la tecnología inalámbrica Bluetooth para redes WPAN.
- En el año 2004, el SIG realiza el lanzamiento de la versión 2.0, mientras que la tecnología Bluetooth alcanza una base instalada de 250 millones de dispositivos en todo el mundo.
- En el año 2009, el SIG lanza la especificación Bluetooth 3.0 High Speedy, convirtiendo el estándar Bluetooth en una tecnología de alta velocidad.
- En el año 2011, el SIG adopta el nuevo perfil para Bluetooth 4.0.
- Actualmente es una de las aplicaciones funcionales para la conexión entre celulares y su futuro está en el hecho de que ahora los dispositivos Bluetooth podrán ser periféricos o centros de conexión.

2.4.2. TIPOS DE BLUETOOTH

BLUETOOTH V1.2

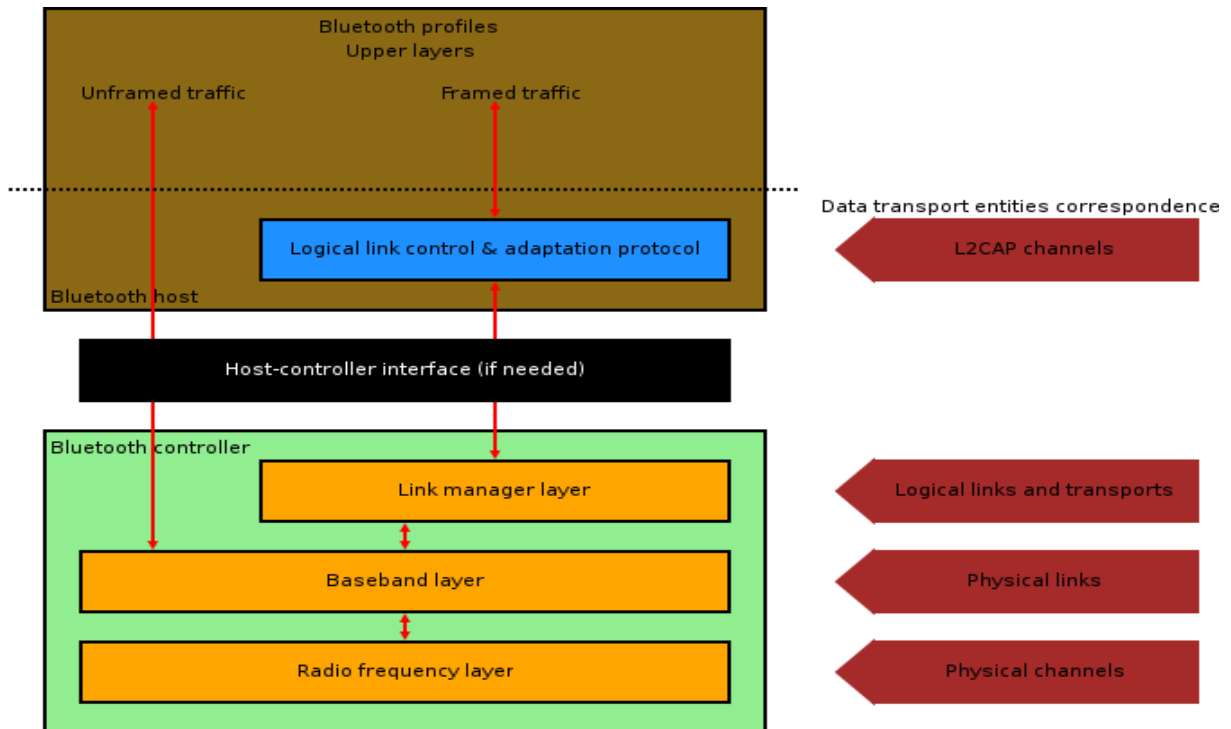


Fig. 9 Tipos de Bluetooth

Se trata de la primera versión ya estandarizada en el IEEE 802.15.1. Diseñada para operar en la banda libre ISM – 2.4 GHz – 2,48 GHz. La especificación Bluetooth cubre 4 niveles de la pila de protocolos (Figura 3.2.2-a) y utiliza saltos de frecuencia y espectro ensanchado– “Frequency Hopping Spread Spectrum” – que se dividen en un total de 79 frecuencias con intervalos de 1 MHz, aportando seguridad y robustez al sistema ante posibles interferencias o Fading. Permite una comunicación Full Duplex y es capaz de transmitir hasta una tasa binaria teórica de 1Mbit/s.

Otras Versiones son: **Bluetooth V2.0 + EDR, Bluetooth V3.0 + HS, Bluetooth V4.1 + LOW ENERGY**

2.5. ANDROID

En los últimos años los teléfonos móviles han experimentado una gran evolución, desde los primeros terminales, grandes y pesados, pensados sólo para hablar por teléfono en cualquier parte, a los últimos modelos, con los que el término “medio de comunicación” se queda bastante pequeño. Es así como nace Android.

Android es un sistema operativo y una plataforma software, basado en Linux para teléfonos móviles. Además, también usan este sistema operativo (aunque no es muy habitual), tablets, netbooks, reproductores de música e incluso PC's. Android permite programar en un entorno de trabajo (framework) de Java, aplicaciones sobre una máquina virtual Dalvik (una variación de la máquina de Java con compilación en tiempo de ejecución). Además, lo que le diferencia de otros sistemas operativos, es que cualquier persona que sepa programar pueda crear nuevas aplicaciones, widgets¹, o incluso, modificar el propio sistema operativo, dado que Android es de código libre, por lo que sabiendo programar en lenguaje Java, va a ser muy fácil comenzar a programar en esta plataforma.

HISTORIA DE ANDROID Y SU FUTURO

Fue desarrollado por Android Inc., empresa que en 2005 fue comprada por Google, aunque no fue hasta 2008 cuando se popularizó, gracias a la unión al proyecto de Open Handset Alliance, un consorcio formado por 48 empresas de desarrollo hardware, software y telecomunicaciones, que decidieron promocionar el software libre. Pero ha sido Google quien ha publicado la mayor parte del código fuente del sistema operativo, gracias al software Apache, que es una fundación que da soporte a proyectos software de código abierto.

Dado que Android está basado en el núcleo de Linux, tiene acceso a sus recursos, pudiendo gestionarlo, gracias a que se encuentra en una capa por encima del Kernel,

accediendo así a recursos como los controladores de pantalla, cámara, memoria flash...

VERSIONES DISPONIBLES

El sistema operativo Android, al igual que los propios teléfonos móviles, ha evolucionado rápidamente, acumulando una gran cantidad de versiones, desde la 1.0 para el QWERTY HTC G1, hasta el mshasmellow que acaba de salir al mercado:

- CUPCAKE: ANDROID VERSION 1.5
- DONUT: ANDROID VERSION 1.6
- ECLAIR: ANDROID VERSION 2.0/2.1
- FROYO: ANDROID VERSION 2.2
- GINGER BREAD: ANDROID VERSION 2.3
- HONEY COMB: ANDROID VERSION 3.0/3.4
- ICE CREAM SANDWICH: ANDROID VERSION 4.0
- ANDROID 4.3 JELLY BEAN (MICHEL)
- ANDROID 4.4 KITKAT (DOGER)
- ANDROID 5.0 LOLLIPOP
- Y LA ULTIMA VERSION ANDROID 6.0 MARSHMALLOW

CAPITULO 3 INGENIERIA DEL PROYECTO

3.1 RELEVAMIENTO DE SISTEMAS DE ALARMAS Y VIGILANCIA ACTUALES EN LA CIUDAD DE LA PAZ

El levantamiento de datos o relevamiento se lo realiza mediante acopio de datos de manera individual, de las Instituciones y Empresas que son parte de la Seguridad Ciudadana Domiciliaria y Vecinal; de acuerdo a esa recopilación de datos tenemos los siguientes resultados:

CAMARAS DE VIGILANCIA DEL GOBIERNO AUTÓNOMO MUNICIPAL DE LA PAZ - GAMLP

Se trata de un sistema integral que procura combinar las capacidades tanto de la Policía Boliviana como del Gobierno Municipal, mediante la instalación de Cámaras de vigilancia que van a permitir administrar el flujo vehicular de La Paz y para paliar el tema de la seguridad Ciudadana. Son 113 que están en funcionamiento actualmente y se tiene programado comprar otras 140 Cámaras.



Foto 1. Monitoreo de cámaras GAMLP

Estos dispositivos se encuentran distribuidos en los siete macrodistritos de la ciudad, el Centro, Max Paredes, Periférica, Cotahuma, Sur, San Antonio y Mallasa, para usar reforzar la seguridad ciudadana, además de las alarmas vecinales que son requeridas en las diferentes zonas de la ciudad, con un costo **1.546.500 bolivianos**,

Las alarmas vecinales son una especie de sirenas que son activadas cuando se está cometiendo un ilícito. En otros barrios de La Paz, los vecinos se organizan y hacen sonar silbatos para actuar y también con fines disuasivos.

COSTO DE EQUIPOS PARA VIGILANCIA OFERTADOS POR EMPRESAS DE VIGILANCIA

Cámara Domo de 22x Motorizada

Costo de Instalación: 210.00 Bs.

Precio Unitario: 7000.00 Bs.

Características: Resolución: 768X494, ZOOM OPTICO: 36 X,

ZOOM DIGITAL: 12X, posicionamiento inteligente 3d.

Tipo de Equipo: Equipo de vigilancia, circuito cerrado.



Cámara Domo PAN/TILT Motorizada

Costo de Instalación: 210.00 Bs.

Precio Unitario: 2500.00 Bs.

Características: Resolución: 510x492 (ntsc), 24 leds, rango de

detección 20 m.

Tipo de Equipo: Equipo de vigilancia, circuito cerrado.



Cámara Ir, Tubo twd-I 102-480

Costo de Instalación: 210.00 Bs.

Precio Unitario: 1150.00 Bs.

Características: Resolución: (510x492) ntsc, 42 led s, rango efectivo: 40-45m.

Tipo de Equipo: Equipo de vigilancia, circuito cerrado.



Cámara Oculta

Costo de Instalación: 210.00 Bs.

Precio Unitario: 1150.00 Bs.

Características: Cámara espía

Tipo de Equipo: Equipo de investigación privada.



3.2 MARCO GEOGRÁFICO

Contempla cualquier Barrio de la ciudad de La Paz.



Foto 2 Vista Aérea de Barrios de La Paz



Foto 3 Barrios de La Paz

3.3 FUNCIONAMIENTO DEL "SISTEMA DE CONTROL ACTIVADO VIA BLUETOOTH PARA ALARMA VECINAL Y DOMICILIARIA"



Fuente "La Razón"

Fig. 10 Funcionamiento Alarma Vecinal

3.4. MATERIALES PARA IMPLEMENTAR EL "SISTEMA DE CONTROL ACTIVADO VIA BLUETOOTH PARA ALARMA VECINAL Y DOMICILIARIA"

MATERIALES

- Arduino Uno Rev2
- Módulo Bluetooth GOMCU
- Modulo de reles
- Dispositivo Android con Bluetooth
- Fuente de 9VDC/500mA con plug p4 o Bateria de 9V co4 plug p4
- Cable USB para la conexión entre la PC y el Arduino, tipoA p/ tipoB
- Focos 220 v
- Alarma o Sirena
- Cables de conexión Dupont macho/hembra
- Cables de 10 mm, para cableado



3.5 DESARROLLO

3.5.1. PROCESO DE IMPLEMENTACION ARDUINO - MÓDULO DE RELES

La implementación del Sistema, se la realiza en base a una placa Arduino por las siguientes características:

- **Asequible** - Las placas Arduino son más asequibles comparadas con otras plataformas de microcontroladores. La versión más cara de un modulo de Arduino puede ser montada a mano, e incluso ya montada cuesta bastante menos de 500 Bs.

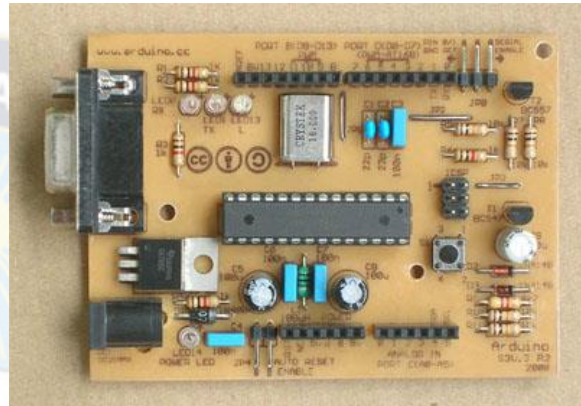


Fig. 11 Placa Arduino

- **Entorno de programación simple y directa** - El entorno de programación de Arduino es fácil de usar para principiantes y lo suficientemente flexible para los usuarios avanzados.
- **Software ampliable y de código abierto**- El software Arduino esta publicado bajo una licencia libre y preparada para ser ampliado por programadores experimentados.
- **Hardware ampliable y de Código abierto** - Arduino está basado en el microcontrolador ATMEGA328. Los planos de los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores de circuitos con experiencia pueden hacer su propia versión del módulo,

Estas características del Arduino UNO r2, hacen que este Proyecto sea practico, barato (su precio oscila entre los Bs. 110 y Bs. 150), fácil de usar, fácil de modificar

según las necesidades (conectar mas lámparas, alarmas u otro dispositivo) y programado por los mismos usuarios (vecinos), previa enseñanza del software y Hardware que son fáciles de realizar y sin necesidad de conseguir permisos legales para su uso comercial.

Una vez explicada el porque de la selección del Arduino para implementar este Proyecto, procedemos a su conexión con el Módulo de relés y de ahí a los focos de 220v y la alarma o sirena, de la siguiente forma:

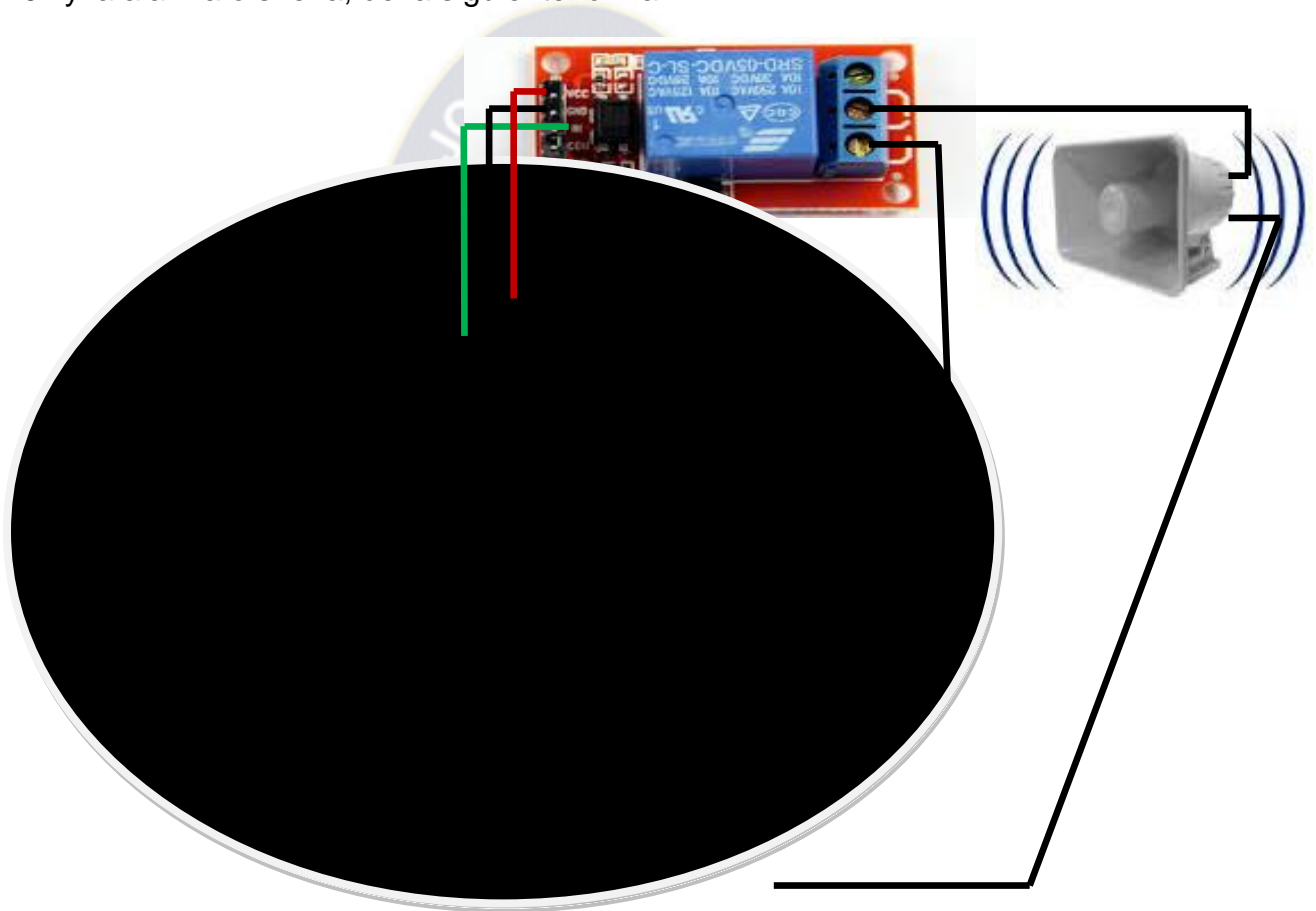


Fig. 12 Esquema de conexión Sistema de Alarma

Conexiones:

Los pines GND y 5 v del Arduino conectan a GND y Vcc del Modulo de relés.

De los pines 2 al 5 de entrada/salida Digital del Arduino conectan a los pines IN1 a IN4 del Módulo de relés.

La alimentación 220v de la línea eléctrica se conectan a las terminales en modo abierto del modulo de relés y de ahí a los focos.

3.5.2 MÓDULO BLUETOOTH Y SU DIAGRAMA DE CONEXIÓN

El Bluetooth es un estándar de comunicación inalámbrica que permite la transmisión de datos a través de radiofrecuencia en la banda de 2,4 GHz. Existen muchos módulos Bluetooth para usarlos en nuestros proyectos de electrónica, pero el que utilizaremos será el CZ-HC-05 gomcu, maestro/esclavo (master/slave), ya que es muy económicos y fácil de encontrar en el mercado. Son módulos pequeños y con un consumo muy bajo que nos permitirán agregar funcionalidades Bluetooth a nuestro Arduino. Estos módulos contienen el chip con una placa de desarrollo con los pins necesarios para la comunicación serie:

- Vcc: Alimentación del módulo entre 3,6V y 6V.
- GND: La masa del módulo.
- TXD: Transmisión de datos.
- RXD: Recepción de datos a un voltaje de 3,3V.
- KEY: Poner a nivel alto para entrar en modo configuración del módulo (solo el modelo HC-05)
- STATE: Para conectar un led de salida para visualizar cuando se comuniquen datos.

El diagrama de conexión es el siguiente:

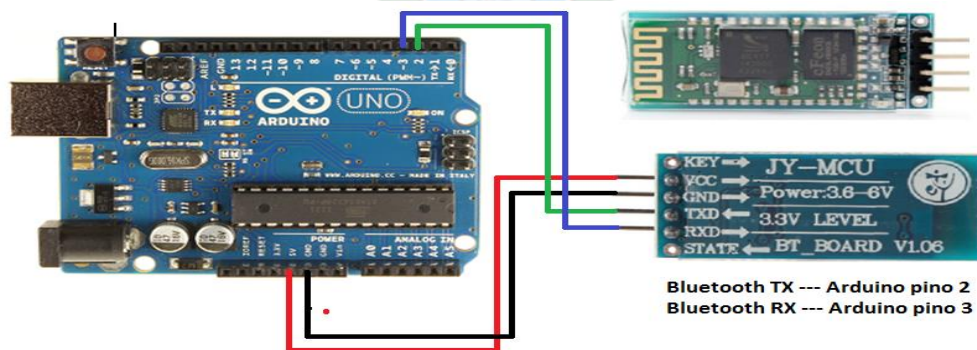
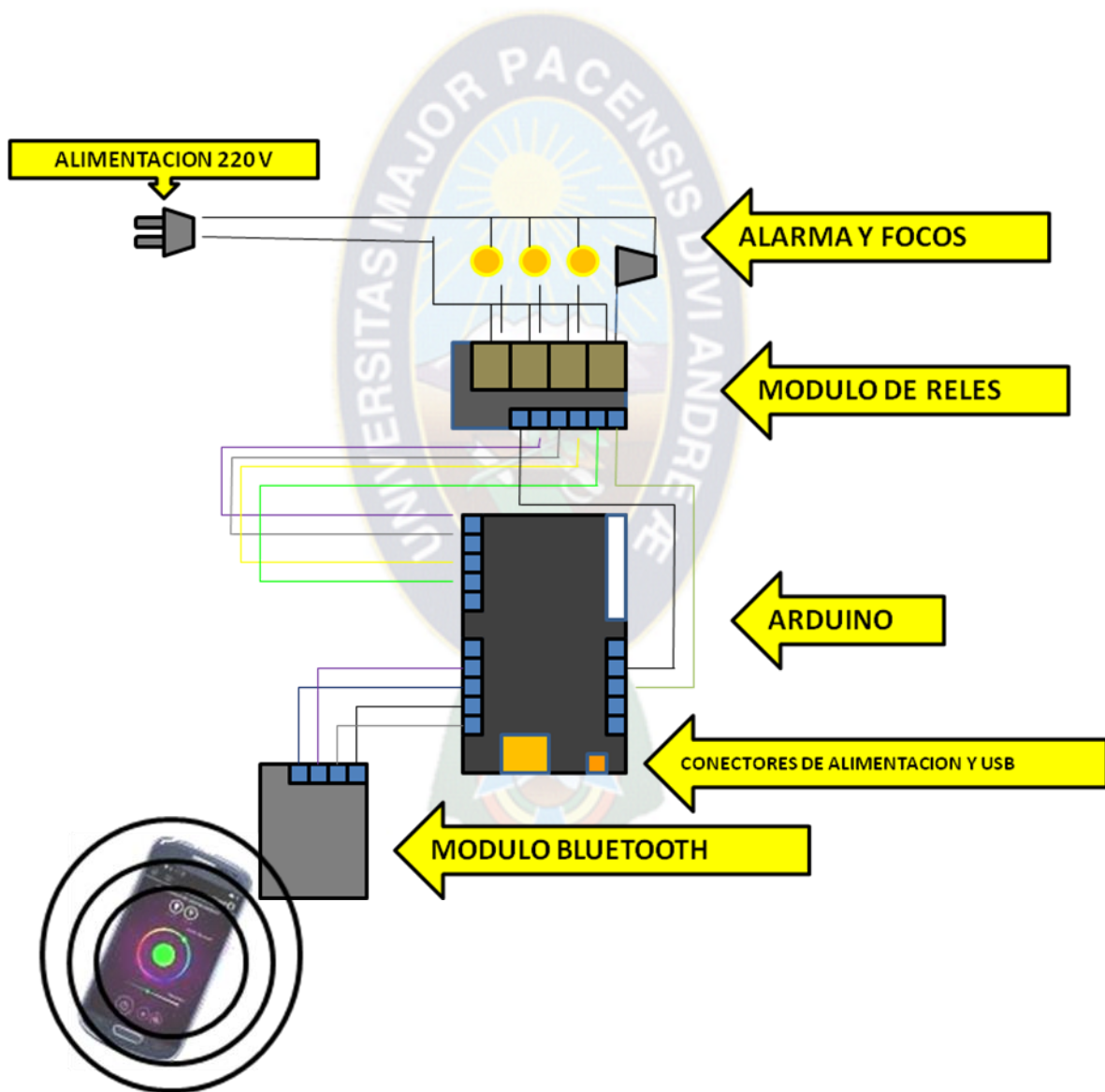


Fig. 13 Conexión de Bluetooth

DIAGRAMA DE BLOQUES DEL "SISTEMA DE CONTROL ACTIVADO VIA BLUETOOTH PARA ALARMA VECINAL Y DOMICILIARIA"



3.5.3. COMANDOS AT

Los comandos AT son un tipo de comandos que sirven para configurar el módulo Bluetooth a través de un microcontrolador, un ordenador o con cualquier dispositivo que posea una comunicación serie (Tx/Rx). Son unas instrucciones que nos permiten cambiar los baudios del módulo, el PIN, el nombre, etc. Para usar los comandos AT el módulo Bluetooth no debe estar vinculado a ningún dispositivo (led rojo del módulo parpadeando). Según las especificaciones del módulo, el tiempo que se tiene que respetar entre el envío de un comando AT y otro tiene que ser de 1 segundo. Si se envía un comando AT y en menos de un segundo se envía otro, el módulo no devuelve respuesta.

Ahora vamos a configurar el módulo Bluetooth a través de comandos AT enviados desde el Monitor Serial del Arduino. En nuestro Proyecto al enviar un comando AT no se debe enviar ningún carácter de LF (nueva línea) ni CR (retorno de carro) al final del comando.

Los comandos AT que se pueden enviar en este módulo son los siguientes:

Comando AT	Descripción	Respuesta
AT	Test de comunicación.	Responde con un OK
AT+VERSION	Retorna la versión del Modulo	OKlinvorV1.8
AT+BAUDx	Configura la velocidad TX del modulo según el valor de "x"	
1 = 1200 bps		
2 = 2400 bps		
3 = 4800 bps		
4 = 9600 bps (por defecto)		
5 = 19200 bps		
6 = 38400 bps		
7 = 57600 bps		
8 = 115200 bps		

9 = 230400 bps

A = 460800 bps

B = 921600 bps

C = 1382400 bps

Una vez compilado el código en el Arduino, abrimos el Monitor Serial y enviamos el comando AT, vemos que el módulo Bluetooth nos responde con un OK.

Con el HC-05 es necesario poner el pin KEY del módulo a nivel alto (5V) para enviar comandos AT. Además, al enviar un comando AT hay que enviar un carácter de NF (nueva línea) y de CR (retorno de carro) al final del comando. Por lo tanto, en el Monitor Serial de Arduino pondremos Ambos NL & CR.

3.5.4 PROGRAMA EN IDE DEL ARDUINO DEL PROYECTO "SISTEMA DE CONTROL ACTIVADO VIA BLUETOOTH PARA ALARMA VECINAL Y DOMICILIARIA"

```
#include<SoftwareSerial.h> //Librería que permite establecer comunicación serie en otros pins
```

```
//Aquí conectamos los pins RXD,TDX del módulo Bluetooth.
```

```
SoftwareSerialBT(10,11); //10 RX, 11 TX.
```

```
boolean a=1;
```

```
boolean b=1;
```

```
boolean c=1;
```

```
boolean d=1;
```

```
int cmd1=2;
```

```
int cmd2=3;
```

```
int cmd3=4;
```

```
int cmd4=5;
```

```
int onbt=12;
```

```
char cadena[255]; //Creamos un array de caracteres de 256 cposiciones
```

```
int i=0; //Tamaño actual del array
```

```
voidsetup()
```

```
{
```

```
BT.begin(9600);
```

```
Serial.begin(9600);
```

```
pinMode(cmd1,OUTPUT);
```

```

pinMode(cmd2,OUTPUT);
pinMode(cmd3,OUTPUT);
pinMode(cmd4,OUTPUT);
pinMode(onbt,OUTPUT);
digitalWrite(onbt,LOW);
delay(50);
digitalWrite(onbt,HIGH);
digitalWrite(cmd1,HIGH);
digitalWrite(cmd2,HIGH);
digitalWrite(cmd3,HIGH);
digitalWrite(cmd4,HIGH);
}

voidloop()
{
  //Cuando haya datos disponibles
  if(BT.available())
  {
    char dato=BT.read(); //Guarda los datos carácter a carácter en la variable "dato"

    cadena[i++]=dato; //Vamos colocando cada carácter recibido en el array "cadena"

    //Cuando reciba una nueva línea (al pulsar enter en la app) entra en la función
    if(dato=='\n')
    {
      Serial.print(cadena); //Visualizamos el comando recibido en el Monitor Serial

      //comando casa
      if(strstr(cadena,"casa")!=0)
      {
        a=!a;
        digitalWrite(cmd2,a);
      }

      //comando calle
      if(strstr(cadena,"calle")!=0)
      {
        b=!b;
        digitalWrite(cmd3,b);
      }

      //comando patio
      if(strstr(cadena,"patio")!=0)
      {
        c=!c;
        digitalWrite(cmd1,c);
      }
    }
  }
}

```

```

//comando alarma
if(strstr(cadena,"alarma")!=0)
{
    d=!d;
digitalWrite(cmd4,d);
}

```

```

//todos encendidos menos alarma
if(strstr(cadena,"encender")!=0)
{
    a=0;
    b=0;
    c=0;
    //d=0;
digitalWrite(cmd1,a);
digitalWrite(cmd2,b);
digitalWrite(cmd3,c);
    //digitalWrite(cmd4,d);
}

```

```

//todos apagados menos alarma
if(strstr(cadena,"Apagar")!=0)
{
    a=1;
    b=1;
    c=1;
    //d=1;
digitalWrite(cmd1,a);
digitalWrite(cmd2,b);
digitalWrite(cmd3,c);
    //digitalWrite(cmd4,d);
}

```

```

BT.write("\r"); //Enviamos un retorno de carro de la app. La app ya crea una línea nueva
clean(); //Ejecutamos la función clean() para limpiar el array
}
}
}

```

```

//Limpia el array
void clean()
{
for (int cl=0; cl<=i; cl++)
{
    cadena[cl]=0;
}
    i=0;
}

```



```

programa_arduino Arduino 1.6.5
Archivo Editar Programa Herramientas Ayuda
programa_arduino

#include <SoftwareSerial.h> //libreria que permite establecer comunicacion serie en otros pines
//Appli conectamos los pines RXD, TXD del modulo Bluetooth.
SoftwareSerial BT(10,11); //10 RX, 11 TX.
boolean a=0;
boolean b=0;
boolean c=0;
boolean d=0;

int cmd1=1;
int cmd2=2;
int cmd3=3;
int cmd4=4;
char cadena[255]; //Creamos un array de caracteres de 256 posiciones
int i=0; //Tamaño actual del array

void setup()
{
  BT.begin(9600);
  serial.begin(9600);
  pinMode(cmd1,OUTPUT);
  pinMode(cmd2,OUTPUT);
  pinMode(cmd3,OUTPUT);
  pinMode(cmd4,OUTPUT);
}

void loop()
{
  //Cuando haya datos disponibles
  if(BT.available())
  {
    char dato=BT.read(); //Guarda los datos carácter a carácter en la variable "dato"
    cadena[i++]=dato; //Vamos colocando cada carácter recibido en el array "cadena"

    //Cuando recibimos una nueva línea (al pulsar enter en la app) entra en la función
    if(dato=='\n')
    {
      Serial.println(cadena); //Visualizamos el comando recibido en el Monitor Serial
      //Comando de encendido
      if(strcmp(cadena,"encendido")!=0)
      {
        a=1;
        digitalWrite(cmd1,a);
      }
      //Comando para
      if(strcmp(cadena,"a=1")!=0)
      {
        b=1;
        digitalWrite(cmd2,b);
      }
      //Comando apaga
      //Comando encendido
      if(strcmp(cadena,"encendido")!=0)
      {
        a=1;
        b=1;
        c=1;
        d=1;
        digitalWrite(cmd1,a);
        digitalWrite(cmd2,b);
        digitalWrite(cmd3,c);
        digitalWrite(cmd4,d);
      }
      //Comando apagado
      if(strcmp(cadena,"apagar")!=0 || strcmp(cadena,"apagar")!=0)
      {
        a=0;
      }
    }
  }
}

```

```

programa_arduino Arduino 1.6.5
Archivo Editar Programa Herramientas Ayuda
programa_arduino

void loop()
{
  //Cuando haya datos disponibles
  if(BT.available())
  {
    char dato=BT.read(); //Guarda los datos carácter a carácter en la variable "dato"
    cadena[i++]=dato; //Vamos colocando cada carácter recibido en el array "cadena"

    //Cuando recibimos una nueva línea (al pulsar enter en la app) entra en la función
    if(dato=='\n')
    {
      Serial.println(cadena); //Visualizamos el comando recibido en el Monitor Serial
      //Comando de encendido
      if(strcmp(cadena,"encendido")!=0)
      {
        a=1;
        digitalWrite(cmd1,a);
      }
      //Comando para
      if(strcmp(cadena,"a=1")!=0)
      {
        b=1;
        digitalWrite(cmd2,b);
      }
      //Comando apaga
      //Comando encendido
      if(strcmp(cadena,"encendido")!=0)
      {
        a=1;
        b=1;
        c=1;
        d=1;
        digitalWrite(cmd1,a);
        digitalWrite(cmd2,b);
        digitalWrite(cmd3,c);
        digitalWrite(cmd4,d);
      }
      //Comando apagado
      if(strcmp(cadena,"apagar")!=0 || strcmp(cadena,"apagar")!=0)
      {
        a=0;
      }
    }
  }
}

```

```

programa_arduino Arduino 1.6.5
Archivo Editar Programa Herramientas Ayuda
programa_arduino

//Comando encendido
if(strcmp(cadena,"encendido")!=0)
{
  a=1;
  digitalWrite(cmd3,a);
}

//Comando b=0
if(strcmp(cadena,"b=0")!=0)
{
  b=0;
  digitalWrite(cmd4,b);
}

//Comando apagado
if(strcmp(cadena,"apagar")!=0 || strcmp(cadena,"apagar")!=0)
{
  a=0;
}
}
}

```

```

programa_arduino Arduino 1.6.5
Archivo Editar Programa Herramientas Ayuda
programa_arduino

digitalWrite(cmd3,c);
digitalWrite(cmd4,d);
}

//Comando apagado
if(strcmp(cadena,"apagar")!=0 || strcmp(cadena,"apagar")!=0)
{
  a=0;
  b=0;
  c=0;
  d=0;
  digitalWrite(cmd1,a);
  digitalWrite(cmd2,b);
  digitalWrite(cmd3,c);
  digitalWrite(cmd4,d);
}

BT.write("\n"); //Enviamos un retorno de carro de la app. La app ya crea una línea nueva
clean(); //Ejecutamos la función clean() para limpiar el array
}

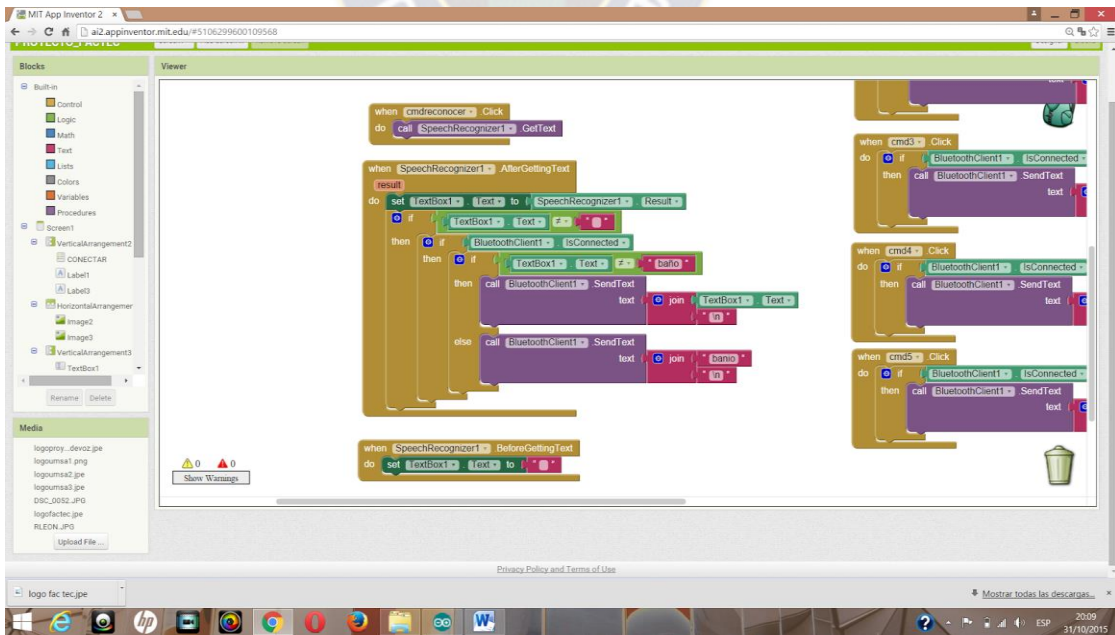
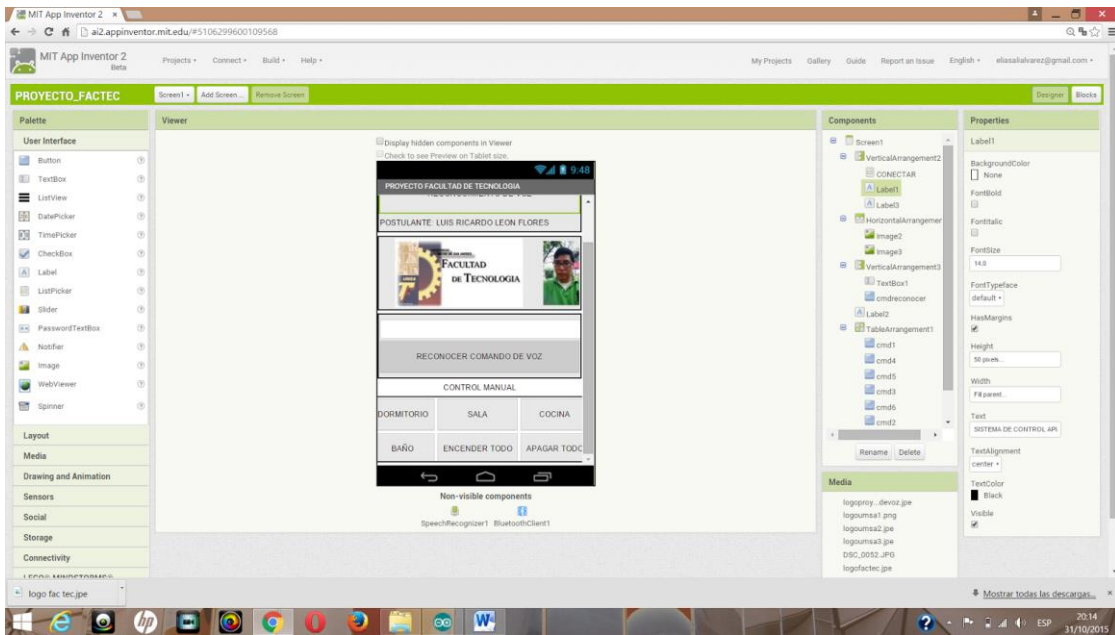
//Limpia el array
void clean()
{
  for (int i=0; i<=i; i++)
  {
    cadena[i]=0;
  }
  i=0;
}
}

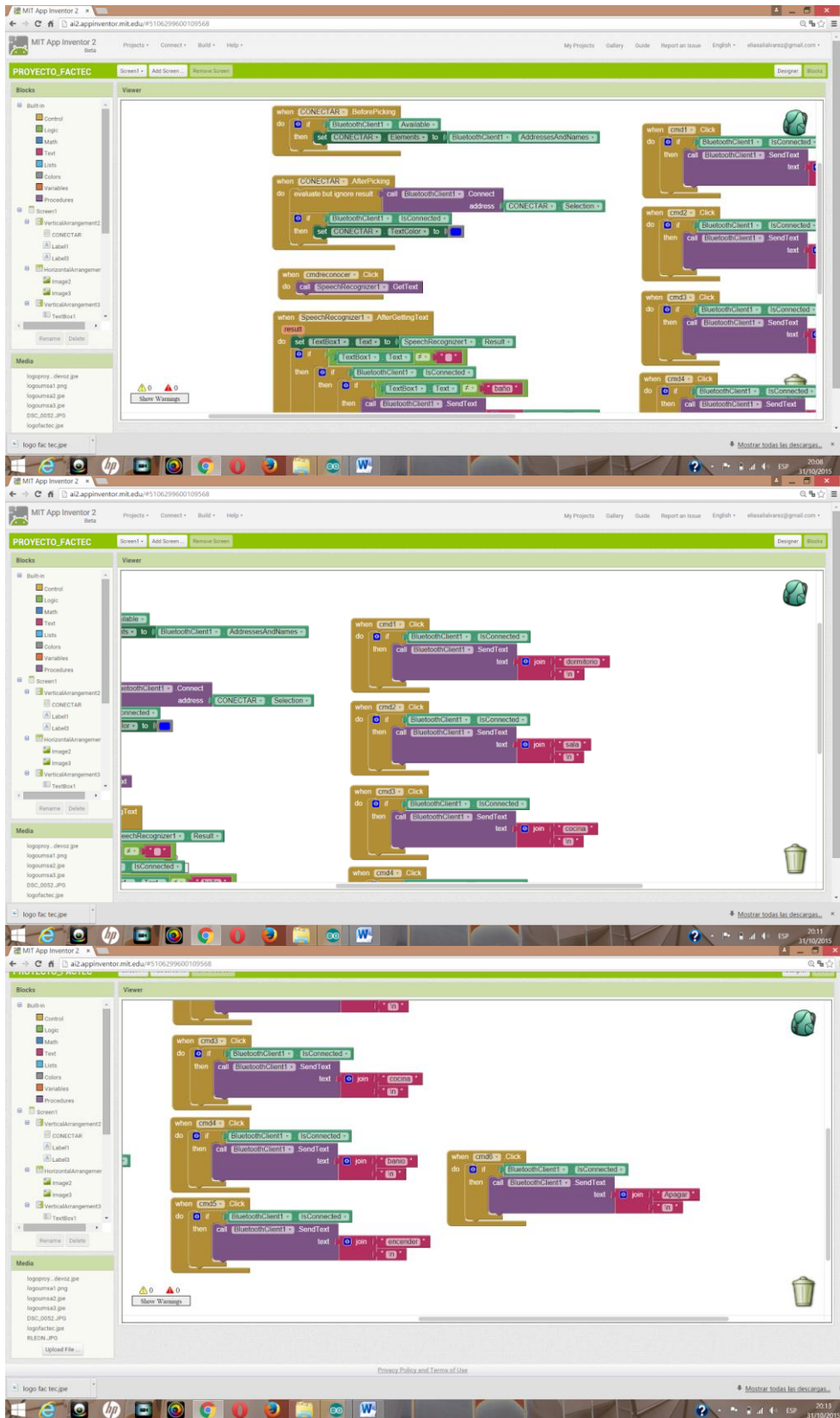
```

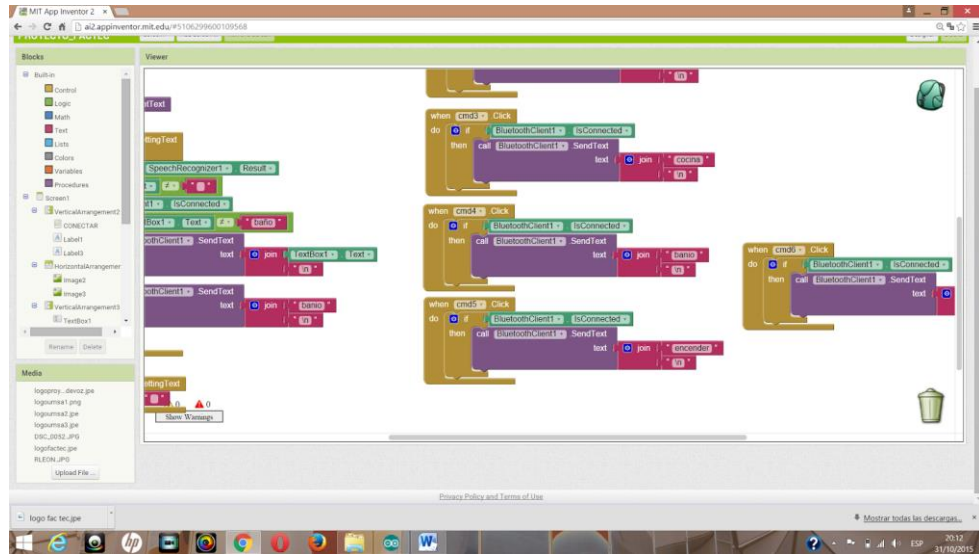
Cuadros 2 Programa IDE Arduino

3.5.5 RECEPCION DE UN DISPOSITIVO ANDROID

Ahora que ya conectamos y configuramos un módulo Bluetooth en nuestro Arduino, vamos a vincularlo a un dispositivo Android con Bluetooth y enviar datos al Arduino mediante la aplicación inventor2.







Cuadros 3. Programa Bluetooth en Inventor2

3.5.6 CONFIGURAR LA APP ANDROID

Una vez descargada la app en el dispositivo móvil, la abrimos y en ajustes ponemos “Conectarse a un dispositivo”.

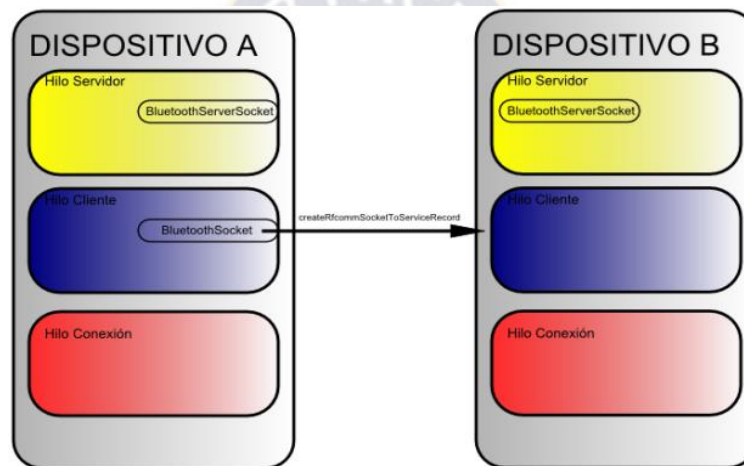
Pulsamos Buscar dispositivos y escogemos nuestro módulo Bluetooth que en nuestro caso le hemos puesto el nombre de FACTEC.

Ponemos el PIN del módulo. Si no se modifico será por defecto 1234.

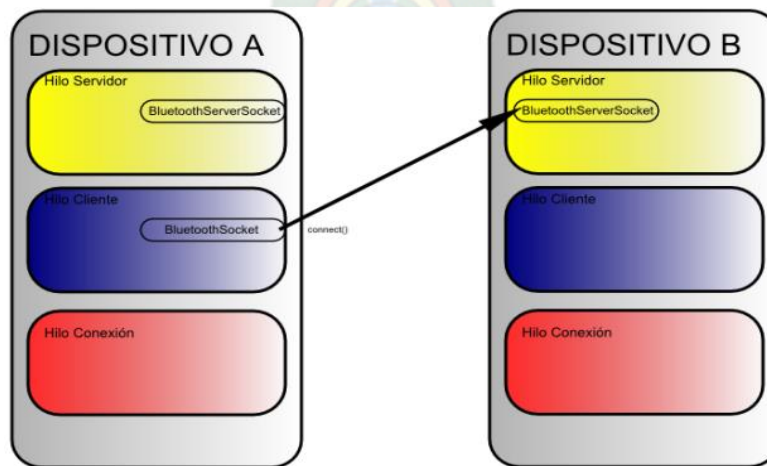
Después de esto, arriba a la derecha pondrá conectado:<nombre> Significa que tenemos asociado nuestro dispositivo Android con nuestro módulo Bluetooth y el led de este deja de parpadear y se queda fijo.

Para entender el funcionamiento del intercambio de información por medio del protocolo Bluetooth que esta en la arquitectura cliente-servidor, recordamos lo siguiente:

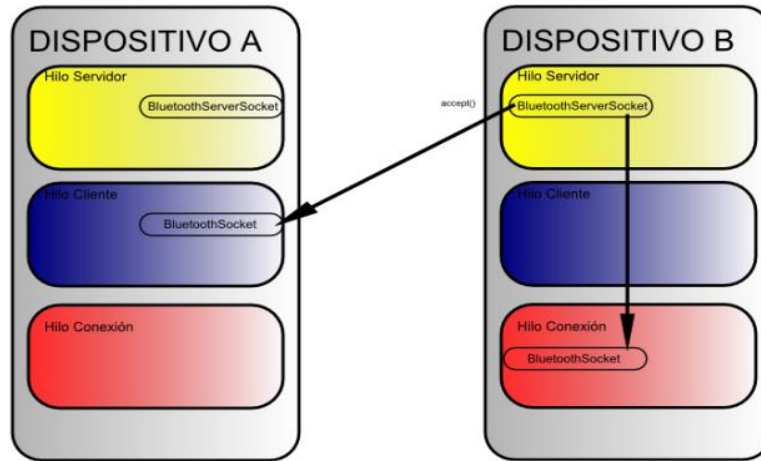
- El servidor crea un socket de tipo `BluetoothServerSocket` y lo pone a la escucha de peticiones de conexión.
- El servidor se mantiene en espera ocupada atendiendo solicitudes de conexión mediante el método `accept()`. Este método bloqueará el hilo hasta que una conexión entrante sea recibida.
- El cliente instancia un dispositivo Bluetooth (`BluetoothDevice`) a partir de la dirección del dispositivo.
- El cliente abre un nuevo socket de tipo `BluetoothSocket` a partir del dispositivo que obtuvo previamente.



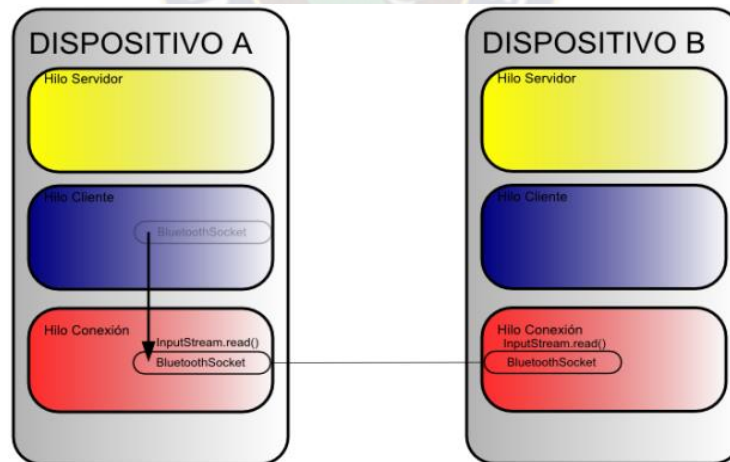
- El cliente realiza, a través del socket, una solicitud de conexión. Esta llamada es bloqueante, por lo que el resultado sólo podrá ser éxito o fracaso.



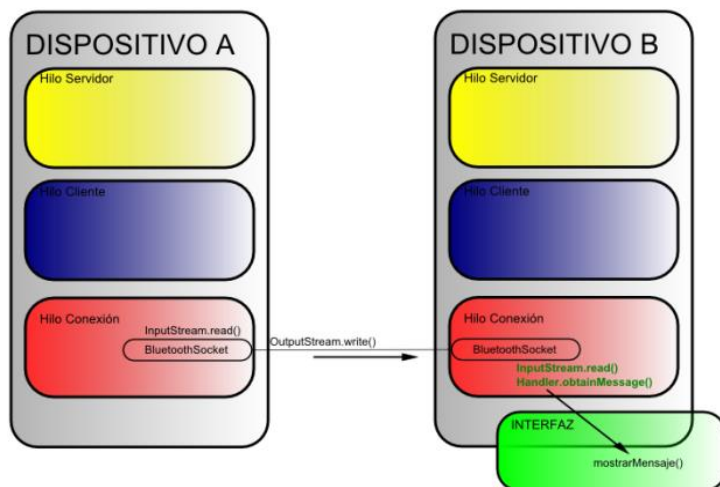
- El servidor acepta la conexión, notificándose al cliente. A continuación, abre un socket de tipo BluetoothSocket como resultado de aceptar la conexión.



- El cliente recibe la notificación del servidor, dando como resultado una conexión correcta y saliendo de espera ocupada
- Tanto cliente como servidor obtienen los flujos de entrada y salida de su respectivo socket.



- El hilo de la conexión comienza a sondear el flujo de entrada esperando obtener datos.
- El envío de datos se realiza mediante la escritura en el flujo de salida.



Cuadros 4. Descripción enlace cliente/servidor de bluetooth

- Cada uno de los elementos involucrados (Servidor, cliente y conexión) estará gestionado por un hilo independiente.

El cliente hace uso de un sólo socket (BluetoothSocket). Se obtiene a partir del BluetoothDevice al cual se pretende conectar, y una vez solicitada y aceptada la conexión (método connect()) se le pasa como parámetro al hilo encargado de mantener la conexión.

- En un principio ambos dispositivos involucrados en la conexión actúan como clientes y como servidores, esperando conexiones entrantes y también pudiendo solicitarlas.
- El servidor será, finalmente, el primero de los dispositivos que acepte una conexión entrante y el proceso de aceptar conexiones se detendrá mientras exista una conexión activa.

Después de esta explicación, finalmente solo hay que conectarnos para comenzar a MANIPULAR LOS COMANDOS QUE SE ENCUENTRAN DETALLADOS EN LA PANTALLA DEL DISPOSITIVO, haciendo que estas se activen o desactiven una por una o que se activen o desactiven en conjunto. AHORA LO MÁS IMPORTANTE ES LA COMUNICACIÓN MEDIANTE LA VOZ, para esto necesitamos conectarnos al Internet ya que gracias a las aplicaciones para este tipo de comunicación que Google nos proporciona, podemos accionar mediante la voz los dispositivos que queramos.

3.6 COSTOS

Una vez realizado el diseño del Sistema se da a conocer los costos en cuanto a equipos y materiales para su implementación.

Cabe mencionar que la información del costo referencial de los materiales, son obtenidos de las diferentes tiendas que expenden materiales eléctricos y electrónicos de la calle Colombia de la ciudad de La Paz.

3.6.1 COSTOS POR EQUIPOS Y MATERIALES DEL SISTEMA

Una parte de la inversión inicial será el costo de los equipos y materiales, necesarios y la otra parte de la instalación en el lugar.

Descripción	Unidad	Precio Bs.	Cantidad	Precio total Bs.
Arduino Uno Rev2	Equipo	130	1	130
Modulo Bluetooth GOMCU	Equipo	50	1	40
Módulo de Relés	Equipo	40	1	40
Batería de 9V co4 plug p4	Equipo	15	1	15
Cable USB para conexión tipo A o B	Metro	1	20	20
Focos y sokcets	Equipo	15	3	45
Alarma	Equipo	100	1	100
Cables de conexión Dupont macho/hembra	Metro	30	1	30
Cables de 10 mm, para cableado	Metro	1,50	100	150
Caja de Protección del Circuito	Equipo	10	1	10
Costo de Instalación		100	1	100
COSTO TOTAL				680
Sensor de Movimiento (opcional)		95	1	95
COSTO TOTAL mas sensor				775

Tabla 1. Costo de Equipos y Materiales para Sistema de Alarma Vecinal

CAPITULO 4 CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

Al ser el Arduino una plataforma de código y hardware abierto (open source initiative), nos da la posibilidad de implementar varios Sistemas de Alarma Vecinal y hasta comercializar sin necesidad de procedimientos legales, haciendo que su programación en el Microcontrolador ATMELE-328, se hace sencilla porque trae incorporado su programador, por lo que por su sencillez y facilidad que ha sido determinante para la implementación y funcionamiento del Proyecto, se establece como una herramienta muy poderosa al momento de prevención y persuadir actos delictivos en los Barrios de la ciudad de La Paz.

Las Plataformas Bluetooth y Android, al ser de código abierto, sencillo, novedoso y fácil de modificar sus programas, permiten que en el Sistema de Alarma Vecinal, se puedan ampliar puntos de salida (hasta 14), para instalar más focos u otros dispositivos como sensores de movimiento, luces, temperatura, etc., que complementarían más la función de vigilancia.

En resumen, los vecinos teniendo un conocimiento básico del funcionamiento, componentes y el software del **"SISTEMA DE CONTROL ACTIVADO VIA BLUETOOTH PARA ALARMA VECINAL Y DOMICILIARIA"**, pueden utilizarlo sin ningún problema y modificarlo a gusto.

4.2 RECOMENDACIONES

Si bien estas tecnologías nos proporcionan una solución a las necesidades actuales no debemos bajar la guardia ya que día a día se generan nuevas tecnologías las cuales pueden mejorar este diseño.

Se recomienda discrecionalidad y responsabilidad al momento del uso del **"SISTEMA DE CONTROL ACTIVADO VIA BLUETOOTH PARA ALARMA VECINAL Y DOMICILIARIA"**, porque permitirá que los vecinos salgan a la calle cada vez que escuchen el sonido de la alarma, tomando en cuenta que se trata de una emergencia.



CAPITULO 5 REFERENCIAS BIBLIOGRÁFICAS

5.1 PAGINAS WEB

- [http://www.eldeber.com.bo/la delincuencia en Bolivia](http://www.eldeber.com.bo/la-delincuencia-en-bolivia)
- <http://www.inesad.edu.bodslm/2014/10/seguridad>
- [https://www.youtube.com/watch?v= MhAIJGQ330](https://www.youtube.com/watch?v=MhAIJGQ330)
- <http://ardx.org/datasheet/ATM328-Guide.pdf>
- <https://riunet.upv.es/bitstream/handle/10251/17631/Memoria.pdf?sequence=1>
- <https://www.arduino.cc/en/Main/ArduinoBoardBT>
- http://www.tutorialspoint.com/android/android_bluetooth.htm
- <https://www.youtube.com/watch?v=j-cUDfmtq5g>
- <https://sites.google.com/site/appinventor>
- http://www.la-razon.com/suplementos/informe/Paz-sufre-robos-venta-drogas-INFORME_0_2271372976.html
- [https://es.wikipedia.org/wiki/Anexo:Historial de versiones de Android](https://es.wikipedia.org/wiki/Anexo:Historial_de_versiones_de_Android)