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

TRABAJO DE APLICACION

PROTOTIPO DE CONTROL DE ILUMINACIÓN PARA LA SALA DE RECUPERACION DE UN HOSPITAL

Examen de Grado presentado para obtener el Grado de Licenciado en Electrónica y Telecomunicaciones

POSTULANTE: José Martin Zuazo Callisaya

La Paz – Bolivia
Noviembre, 2017

RESUMEN

En este trabajo de aplicación se propone un sistema de bajo costo capaz de controlar la iluminación de las lámparas inalámbricamente a través del protocolo de comunicación Bluetooth y en una plataforma Android. Para realizar dicho objetivo se plantea desarrollar un control de PWM realizado a través de una placa Arduino Mega 2560, de manera que este sea el encargado de realizar los procesos necesarios para el correcto funcionamiento del prototipo. La implementación del sistema muestra beneficios en acceso y control al momento de querer controlar la iluminación. Se podrá observar que se puede usar en cualquier dispositivo que se tenga el sistema android y pueda conectarse inalámbricamente a través de Bluetooth.

El prototipo se ha diseñado para cubrir las necesidades de una sala de recuperación de un Hospital, de modo que el personal hospitalario pueda controlar la iluminación. Por otra parte se diseñó la posición de los focos, para su continua mejora, obteniendo un prototipo funcional, que se convierte en una alternativa para el desarrollo de atención y recuperación de un paciente con tecnología apropiada.

INDICE

RESUMEN ii

CAPITULO I INTRODUCCION 1

1.1. PLANTEAMIENTO DEL PROBLEMA 2
1.2. JUSTIFICACIÓN DEL TRABAJO 3
1.3. OBJETIVOS 4
1.3.1. OBJETIVO GENERAL 4
1.3.2. OBJETIVOS ESPECIFICOS 4

CAPITULO II FUNDAMENTACION TEORICA 5

2.1. SALA DE RECUPERACION 5
2.2. EFICIENCIA LUMINICA EN LA SALA DE RECUPERACION 5
2.3. LA PLACA ARDUINO 7
2.3.1. HARDWARE DEL ARDUINO 7
2.3.2. SOFTWARE DE ARDUINO 8
2.4. TECNOLOGIA BLUETOOTH 9
2.4.1. TIPOS DE BLUETOOTH 10
2.4.2. COMANDOS AT PARA BLUETOOTH 12
2.5.MODULACION POR ANCHO DE PULSO (PWM) 13
2.6.ILUMINACION LED DECORRIENTE CONTINUA 15

CAPITULO III DESARROLLO DEL TRABAJO 18

3.1. DISEÑO DEL PROTOTIPO A IMPLEMENTAR 18
3.1.1. LA APLICACIÓN 18
3.1.2. COMUNICACIÓN ENTRE EL CELULAR
Y LA PLACA ARDUINO 20
3.1.3. LA PLACA ARDUINO Y CIRCUITOS 29
3.1.4. CONEXIONES DEL BLUETOOTH Y COMUNICACIÓN SERIAL 32
3.1.5. FOCOS LEDS 33

3.1.6. BOTON DE RESET EXTERNO 34

3.2. DIAGRAMA BÁSICO DE LA SALA DE RECUPERACIÓN DE UN HOSPITAL 34

3.3. MATERIALES Y CIRCUITO PARA IMPLEMENTAR EL POTOTIPO DE CONTROL DE ILUMINACION 36

3.4. COSTOS DEL PROTOTIPO 40

CAPITULO IV CONCLUSIONES Y RECOMENDACIONES 41

4.1. CONCLUSIONES 41

4.2. RECOMENDACIONES 42

4.3. BIBLIOGRAFIA 43

CAPITULO I

INTRODUCCION

En el transcurso del tiempo se ha encontrado con el avance de la tecnología en los hogares mejor conocida como domótica. Entre los diversos sistemas que se pueden controlar de manera inteligente, en un hogar se puede controlar la iluminación. Las bombillas incandescentes de hoy son cada vez más reemplazadas por bombillas LED que proporcionan una mayor eficiencia y ahorro. Típicamente las bombillas LED utilizadas en la iluminación tienen una potencia en el gama de 5 W a 20 W.

La mejor manera de realizar un manejo eficaz y cómodo de la iluminación en el hogar es mediante la combinación de tecnologías móviles y microcontroladores de manera que se logre un control óptimo para el sistema. Además con el desarrollo de este sistema es posible sentar las bases para el control de otros sistemas en el hogar e incluso para controlar procesos industriales.

Las capacidades del software de código abierto y los microcontroladores se utilizan para construir un dispositivo para experimentos de iluminación controladas.

Un sistema de sombreado de iluminación y control de la luz del día-sensible implementado en edificios hace uso de la detección en tiempo real y simulación de iluminación. Este sistema puede controlar la posición de persianas de la ventana y el estado de las luminarias.

El interfaz estándar de la habitación del hospital para el control de los dispositivos de comunicación y entretenimiento asume que el paciente tiene la capacidad de mantener y presionar los interruptores mecánicos. Si el paciente no tiene estas habilidades, entonces el paciente debe esperar a una enfermera para caminar por la habitación para pedir ayuda.

Los dispositivos móviles que utilizan los sistemas operativos IOS y Android se comparan para poder ayudar a las limitaciones de un paciente para la comunicación y el control del entorno de la sala de recuperación del hospital.

1.4. PLANTEAMIENTO DEL PROBLEMA

El correcto diseño de la iluminación de hospitales puede afectar de forma muy positiva en la recuperación de los pacientes, y a la vez, la correcta iluminación de hospitales también puede mejorar la eficiencia del personal. Hoy en día, los hospitales son auténticos centros tecnológicos y de investigación médica y el bienestar del personal investigador y científico también es un factor a tener en cuenta por los administradores hospitalarios.

Por otro lado, las habitaciones también son áreas de trabajo para el personal y por consiguiente requieren de instalaciones integradas así como unos niveles de iluminación adecuados en ciertos momentos puntuales para llevar a cabo su trabajo.

En la actualidad están incrementando los artículos que advierten sobre los efectos nocivos del ambiente no adecuado en las unidades de recuperación, convirtiéndose en fuentes potenciales de enfermedades y recaídas. El control de la luz comienza a ser una difícil tarea, ya que la mayoría de estos estímulos son generados por las actividades durante la asistencia.

Un punto importante para anotar son las variaciones que requieren los distintos tipos de paciente. Y también se aconseja que el diseñador explore a fondo las exigencias específicas del tipo de paciente. Estos incluyen áreas de circulación, camas de pacientes, y cada uno necesita un tratamiento particular para conseguir la recuperación efectiva del paciente.

Se ve así la necesidad de plantearnos la principal pregunta:

¿Cuál será el Prototipo de control de luminosidad para la sala de recuperación de un hospital?

1.5. JUSTIFICACIÓN DEL TRABAJO

Las habitaciones de recuperación del hospital son el lugar donde los pacientes pasan la mayor parte del tiempo, por este motivo, el objetivo principal será el de describir un prototipo de control de iluminación agradable que contribuya a mejorar su bienestar con la intención de crear un ambiente saludable y propiciar una recuperación más rápida.

Con el siguiente prototipo de control de luminosidad se tendrá una solución adecuada a los requerimientos que se necesita tanto por parte de los funcionarios como de los pacientes. Al ser inalámbrica será cómoda en su uso, también será accesible, ya que es utilizable en cualquier celular que contenga el sistema operativo android y tenga una conexión inalámbrica bluetooh, las cuales la contiene cualquier celular en la actualidad.

Como se dijo anteriormente la iluminación puede contribuir al rendimiento del personal, estado de bienestar del paciente y a los demás usuarios de ambiente, sabiendo también que el bienestar de las personas es un valor que no se debe perder en la sociedad.

Una iluminación bien planificada y el esquema de un prototipo de este control de luminosidad pueden ayudar a alcanzarlo.

1.6. OBJETIVOS

1.6.1. OBJETIVO GENERAL

- Desarrollar un prototipo de control de iluminación para la sala de recuperación de un hospital.

1.6.2. OBJETIVOS ESPECIFICOS

- Presentar los conceptos electrónicos relacionados con el prototipo de sistema de control de iluminación.
- Describir de manera breve los componentes y los materiales para controlar el prototipo del sistema de control de iluminación.
- Describir la interfaz gráfica para controlar el hardware de manera inalámbrica.
- Describir la función de los pequeños sistemas empleados dentro del prototipo.
- Consolidar los conceptos referidos a la modulación de ancho de pulso con la plataforma Arduino.
- Repasar el sensor LDR como detector de presencia para el apagado de un foco.
- Describir los conceptos básicos acerca de la comunicación serial que utiliza Arduino para comunicarse con el modulo Bluetooth.

CAPITULO II

FUNDAMENTACION TEORICA

2.7. SALA DE RECUPERACION

La sala de recuperación o unidad de cuidados postanestésicos, es la sala donde se deriva a un paciente luego de habérsele realizado una operación para que pueda recuperarse adecuadamente. La recuperación es supervisada por los distintos profesionales en salud, los cuales básicamente hacen el control de:

- Los signos vitales
- Cualquier complicación que se presente en el paciente
- La temperatura del paciente
- El grado de conciencia del paciente
- La revisión de dispositivos conectados al paciente
- La revisión de la herida del paciente
- Las infusiones por vía intravenosa
- La comodidad del paciente

Figura 1. Sala de recuperación
Fuente: Grupo Lledó

2.8. EFICIENCIA LUMINICA EN LA SALA DE RECUPERACION

La eficiencia de luces dentro de un hospital es importante ya que a través de ella se llega a los siguientes puntos que hacen de un hospital sea eficiente:

- Luz para el bienestar, una adecuada iluminación favorece a la recuperación de los pacientes.
- Luz para el trabajo, contribuye de forma óptima al desarrollo de las diferentes funciones visuales vinculadas al tratamiento y al reconocimiento médico.
- Altamente funcional, 24 horas al día, 365 días a la semana

A su vez todo esto nos lleva a las salas de recuperación de los hospitales en los cuales se debe tomar en cuenta los siguientes puntos:

- Elementos oscuros contra superficies claras: contraste y definición
- Elevados niveles generales > 100 lux
- Luminancias equilibradas en techo y paredes
- Reducción de contraste con la luz solar precedente de ventanas y del deslumbramiento resultante

Figura 2. Distintos niveles de iluminación en una sala de recuperación
Fuente: Grupo Lledó

Actividad o función	**Iluminancia en lux/m2**	**Observación**
Iluminación general	100	Se deben evitar luminancias excesivas en el campo visual del paciente
Iluminación de Lectura	300	
Visitas medicas	300	
Visitas médicas y tratamiento	1000	
Ilutación nocturna	5	

Tabla 1. Distintos tipos de iluminación en la sala de recuperación
Fuente: Grupo Lledó

2.9. LA PLACA ARDUINO

2.9.1. HARDWARE DEL ARDUINO

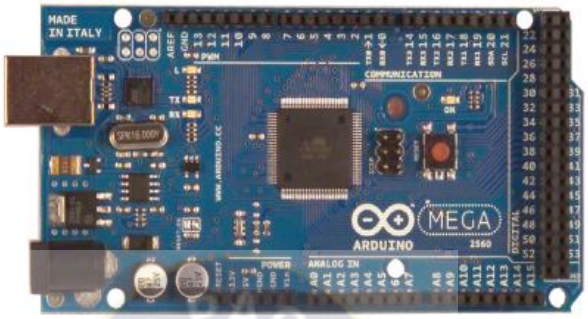


Figura 3. Arduino Mega 2560
Fuente: www.arduino.cc

Arduino Mega es una tarjeta de desarrollo *open-source* construida con un microcontrolador modelo Atmega2560 que posee pines de entradas y salidas (E/S), analógicas y digitales. Esta tarjeta es programada en un entorno de desarrollo que implementa el lenguaje *Processing/Wiring*. Arduino puede utilizarse en el desarrollo de objetos interactivos autónomos o puede comunicarse a un PC a través del puerto serial (conversión con USB) utilizando lenguajes como Flash, Processing, MaxMSP, etc. Las posibilidades de realizar desarrollos basados en Arduino tienen como límite la imaginación.

El Arduino Mega tiene 54 pines de entradas/salidas digitales (14 de las cuales pueden ser utilizadas como salidas PWM), 16 entradas análogas, 4 UARTs (puertos serial por hardware), cristal oscilador de 16MHz, conexión USB, jack de alimentación, conector ICSP y botón de reset. Arduino Mega incorpora todo lo necesario para que el microcontrolador trabaje; simplemente conéctalo a tu PC por medio de un cable USB o con una fuente de alimentación externa (9 hasta 12VDC).

Esta nueva versión de Arduino Mega 2560 adicionalmente a todas las características de su sucesor utiliza un microcontrolador ATMega8U2 en vez del circuito integrado FTDI. Esto permite mayores velocidades de transmisión por su puerto USB y no requiere drivers para Linux o MAC (archivo inf es necesario para Windows) además ahora

cuenta con la capacidad de ser reconocido por el PC como un teclado, mouse, joystick, etc.

Para más información sobre este producto podemos visitar www.arduino.cl, allí encontraremos más tutoriales y ejemplos con Arduino.

Características:

- Microcontrolador ATmega2560.
- Voltaje de entrada de: 7-12V.
- 54 pines digitales de Entrada/Salida (14 de ellos son salidas PWM).
- 16 entradas análogas.
- 256k de memoria flash.
- Velocidad del reloj de 16Mhz.

2.9.2. SOFTWARE DE ARDUINO

Los pasos para poder utilizar el software de Arduino

- **Descarga el entorno Arduino**. A que ingresar a la página de Arduino (http://arduino.cc/en/Main/Software) y elejimos el sistema operativo donde lo vamos a utilizar.
- **Instalación del entorno Arduino.** Es posible que en el momento de instalar el IDE (entorno de desarrollo) haya una versión nueva disponible, pero esencialmente los pasos serán los mismos. Aceptamos los términos de condición, le damos siguiente a los controladores a instalar, escogemos la carpeta donde estarán los archivos de Arduino, finalmente instalamos el programa Arduino.
- **Conexión del Arduino.** Usando el cable USB conectamos el Arduino a la computadora.
- **Ejecución del IDE**. Buscamos en el escritorio el acceso directo al IDE Arduino y damos doble clic. Luego verificamos que en Herramientas -> Puerto se encuentre seleccionado el puerto COM que tiene conectado el Arduino. Aparecerá entre paréntesis “Arduino Uno”.

- **Empecemos a realizar programas.** Se necesita saber las funciones y variables aceptables dentro del IDE de Arduino, por lo que mostramos a continuación un resumen hecho por ELINSI:

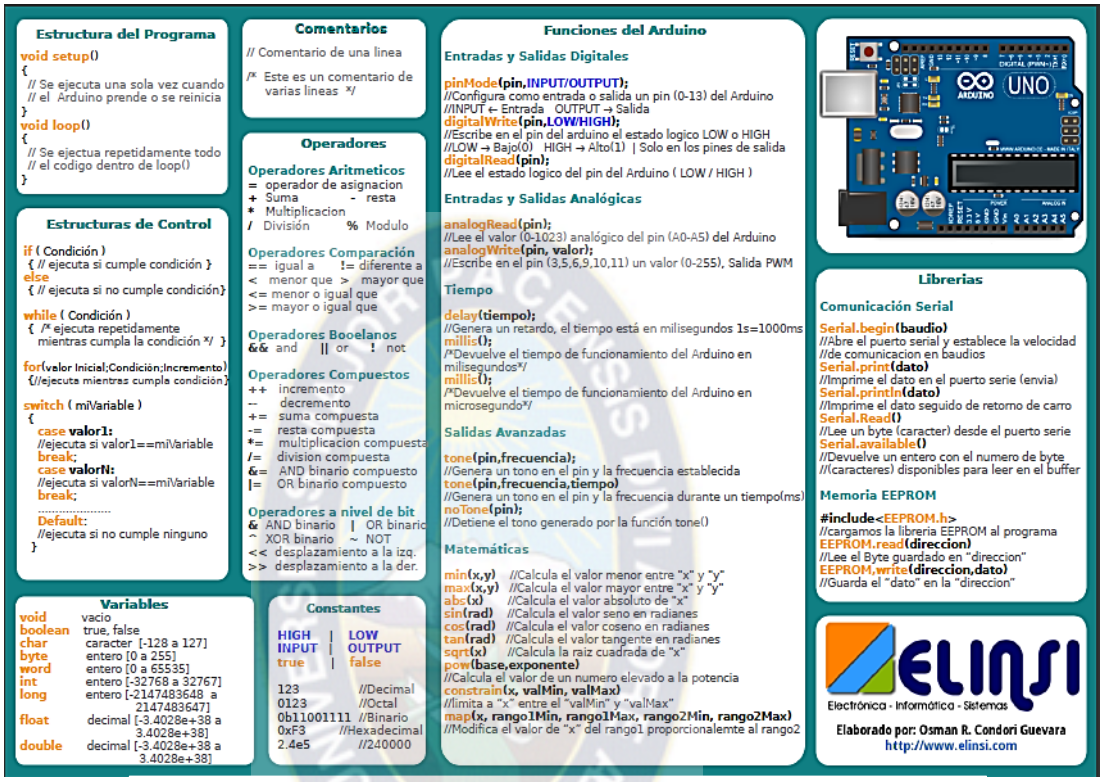


Figura 4. Ayuda memoria para IDE Arduino
Fuente: Grupo ELINSI

2.10. TECNOLOGIA BLUETOOTH

Bluetooth es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2.4 GHz. Los principales objetivos que se pretenden conseguir con esta norma son:

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

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. La especificación de Bluetooth define un canal de comunicación a un máximo 720 kbit/s (1 Mbit/s de capacidad bruta) con rango óptimo de 10 m (opcionalmente 100 m con repetidores). Opera en la frecuencia de radio de 2,4 a 2,48 GHz

2.10.1. TIPOS DE BLUETOOTH

Los dispositivos que incorporan este protocolo pueden comunicarse entre sí cuando se encuentran dentro de su alcance. Las comunicaciones se realizan por radiofrecuencia de forma que los dispositivos no tienen que estar alineados y pueden incluso estar en habitaciones separadas si la potencia de transmisión es suficiente. Estos dispositivos se clasifican como "Clase 1", "Clase 2" o "Clase 3" en referencia a su potencia de transmisión.

Los dispositivos con Bluetooth también pueden clasificarse según su capacidad de canal:

- Versión 1.2 1 Mbit/s
- Versión 2.0 + EDR 3 Mbit/s
- Versión 3.0 + HS 24 Mbit/s
- Versión 4.0 32 Mbit/s

con amplio espectro y saltos de frecuencia con posibilidad de transmitir en Full Duplex con un máximo de 1600 saltos por segundo. Los saltos de frecuencia se dan entre un total de 79 frecuencias con intervalos de 1 MHz; esto permite dar seguridad y robustez.

Tanto Bluetooth clásico como Bluetooth de baja energía aplican la característica de salto de frecuencia adaptativa (AFH – Adaptive Frequency Hopping) que detecta interferencias de. Por ejemplo, un dispositivo WLAN 802.11 b, g, n que transmite cercano, si se detecta tal interferencia, el canal se coloca automáticamente en la lista negra. Con el fin de manejar la interferencia temporal, un esquema implementado vuelve a intentar los canales de la lista negra y si la interferencia ha cesado el canal se puede

utilizar. AFH impide que Bluetooth interfiera con otras tecnologías inalámbricas cercanas.

El hardware que compone el dispositivo Bluetooth está compuesto por dos partes: un dispositivo de radio, encargado de modular y transmitir la señal. un controlador digital, compuesto por una CPU, un procesador de señales digitales (DSP – Digital Signal Processor) llamado Link Controller (o controlador de Enlace) y de las interfaces con el dispositivo anfitrión.

El LC o Link Controller se encarga del procesamiento de la banda base y del manejo de los protocolos ARQ y FEC de la capa física; además, se encarga de las funciones de transferencia tanto asíncrona como síncrona, la codificación de audio y el cifrado de datos.

Los dispositivos BlueTooth pueden actuar como Masters o como Slaves. 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).

Cada uno de los dispositivos que se identifican vía BlueTooth presentan 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 mostrarás 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. Por eso puedes conectarte vía Bluetooth a tu PC, por ejemplo, y a través de él conectarte 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, 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.

2.10.2. COMANDOS AT PARA BLUETOOTH

Para manejar los módulos Bluetooth usamos comandos Hayes o AT: Comandos AT básicos:

- AT+VERSION, versión del Firmware
- AT+NAMEXXX, Programa el nombre que queremos presentar cuando alguien nos busque
- AT+BAUDX, Fija la velocidad de comunicación entre el modulo y la consola según a la siguiente tabla:

1 configura	1200bps
2 configura	2400bps
3 configura	4800bps
4 configura	9600bps (Default)
5 configura	19200bps
6 configura	38400bps
7 configura	57600bps

8 configura	115200bps

Tabla 2. Velocidades de comunicación de bits de un módulo bluetooh
Fuente: www.elprofegarcia.com

- AT+PINXXXX, configura el número de identificación personal, que se requerirá para establecer la vinculación
- AT+ROLE Nos informa de si está configurado como Maestro 1, o como esclavo 0.
- AT+ROLE1 Configura el modulo como Master.
- AT+ROLE0 Configura el modulo como Slave.

Los módulos HC-05 y HC-06 son Bluetooth V2. Pero con el tiempo han ido apareciendo módulos como los que conocemos que soportan el protocolo Bluetooth V4.0 o Low Energy al alcance de todos los bolsillos y los fabricantes chinos han empezado a suministrarlos de forma accesible, tales como los modelos HC-08 y HC-10.

El nuevo Bluetooth 4.0 es un nuevo protocolo diseñado pensando en disminuir todo lo posible las necesidades de energía de los dispositivos que lo usan, y de sobre todo de la propia comunicación que tradicionalmente ha sido de consumo considerable.

2.11. MODULACION POR ANCHO DE PULSO (PWM)

Una señal eléctrica analógica es aquella en la que los valores de la tensión o voltaje varían constantemente y pueden tomar cualquier valor. En el caso de la corriente alterna, la señal analógica incrementa su valor con signo eléctrico positivo (+) durante medio ciclo y disminuye a continuación con signo eléctrico negativo (–) en el medio ciclo siguiente.

Un sistema de control (como un microcontrolador) no tiene capacidad alguna para trabajar con señales analógicas, de modo que necesita convertir las señales analógicas en señales digitales para poder trabajar con ellas.

La señal digital obtenida de una analógica tiene dos propiedades fundamentales:

- Valores. Que valor en voltios define 0 y 1. En nuestro caso es tecnología TTL (0 – 5V)
- Resolución analógica: nº de bits que usamos para representar con una notación digital una señal analógica.

En el caso de un Arduino Uno, el valor de 0 voltios analógico es expresado en digital como B0000000000 (0) y el valor de 5V analógico es expresado en digital como B1111111111 (1023). Por lo tanto todo valor analógico intermedio es expresado con un valor entre 0 y 1023, es decir, sumo 1 en binario cada 4,883 mV.

La modulación por ancho de pulsos también se usa para controlar servomotores, los cuales modifican su posición de acuerdo al ancho del pulso enviado cada un cierto período que depende de cada servo motor. Esta información puede ser enviada utilizando un microprocesador o un microcontrolador (por ejemplo, un PIC 16F877A, 16F1827, 18F4550, etc. de la empresa Microchip), o un microcontrolador de hardware libre como es Arduino.

Otra aplicación es enviar información de manera analógica. Es útil para comunicarse de forma analógica con sistemas digitales.

Para un sistema digital, es relativamente fácil medir cuánto dura una onda cuadrada. Sin embargo, si no se tiene un conversor analógico digital no se puede obtener información de un valor analógico, ya que sólo se puede detectar si hay una determinada tensión, 0 o 5 voltios por ejemplo (valores digitales de 0 y 1), con una cierta tolerancia, pero no puede medirse un valor analógico. Sin embargo, el PWM en conjunción con un oscilador digital, un contador y una puerta AND como puerta de paso, podrían fácilmente implementar un ADC.

2.12. ILUMINACION LED DE CORRIENTE CONTINUA

Figura 5. Focos Leds
Fuente: Grupo ELINSI

Una situación muy común al tratar de sustituir focos halógenos de exterior es seleccionar focos LED de potencia superior a la que le correspondería por rendimiento lumínico, ya sea por "curarse en salud" o por desconfianza en la tecnología LED. Esto puede derivar fácilmente en que nos encontremos con una iluminación excesiva en la zona a tratar que puede llegar a resultar desagradable.

Aunque lo ideal sería guiarnos por el flujo luminoso (lumen) de cada modelo, esta tabla puede servirnos como aproximación:

Focos halógenos	Focos LED
400w	60W
450W	80W
550w	90W
750W	120W

Tabla 3. Focos halógenos y su equivalente en leds dc
Fuente: www.nergiza.com

Los focos halógenos de exterior es difícil encontrarlos de potencias inferiores a 400W, pero los focos LED podemos comprarlos con potencias desde 10W si necesitamos menos iluminación.

A la hora de utilizar un foco LED y en general cualquier tipo de dispositivo en el exterior, debemos fijarnos en su grado de protección IP (en la etiqueta de características), el cual define lo protegido que está el equipo frente al polvo a líquidos.

Si por ejemplo nos encontramos con un IP65, la primera cifra (6) sería la protección frente a cuerpos sólidos (polvo) y se puede consultar en la siguiente lista:

- 0.- Sin protección
- 1.- Protegido contra cuerpos sólidos de más de 50mm de diámetro
- 2.- Protegido contra cuerpos sólidos de más de 12,5mm de diámetro
- 3.- Protegido contra cuerpos sólidos de más de 2,5mm de diámetro
- 4.- Protegido contra cuerpos sólidos de más de 1mm de diámetro
- 5.- Protegido contra polvo sin sedimentos perjudiciales
- 6.- Protegido contra polvo totalmente

La segunda cifra (5) nos definiría el grado de protección frente al agua siguiendo estos índices:

- 0.- Sin protección
- 1.- Protegido contra caídas verticales de gotas de agua
- 2.- Protegido contra caídas de gostas de agua con hasta 15º de inclinación
- 3.- Protegido contra el agua de lluvia con hasta 60º de inclinación
- 4.- Protegido contra salpicaduras de agua en todas las direcciones
- 5.- Protegido contra chorros de agua en todas las direcciones
- 6.- Protegido contra lanzamiento de agua similar a los golpes de mar
- 7.- Protegido contra la inmersión pasajera
- 8.- Protegido contra inmersiones prolongadas

En el caso de los focos LED para exterior buscaremos siempre como mínimo una protección IP65: protegido contra polvo totalmente y contra chorros de agua en todas las direcciones.

Temperatura de color en focos LED

Es muy común encontrase gente descontenta con la tecnología LED en general, ya que afirman que dan una luz “demasiado blanca”, pareciendo que estás en un quirófano en vez de en tu casa tranquilamente.

Esto es debido a la selección incorrecta de nuestro foco LED o bombilla, como ya dijimos anteriormente, es necesario conocer la temperatura de color del foco, pudiendo ir desde azulado hasta blanco cálido (amarillento) pasando por simplemente “blanco”.

En esta imagen podéis consultar la correspondencia entre los “grados de color” y el tipo de luz que nos va a ofrecer nuestro foco LED.

Aunque no es un tema exclusivo de la iluminación LED, existen en el mercado focos LED con detector de movimiento incorporado. Este tipo de focos pueden hacernos ahorrar cantidades importantes de energía y dinero si lo instalamos en zonas de uso ocasional como por ejemplo un parking exterior.

Además, estos focos con detector de movimiento pueden ofrecernos un confort superior al foco tradicional accionado con interruptor, además de una mayor facilidad de instalación (sin cableado hasta interruptor).

CAPITULO III

DESARROLLO DEL TRABAJO

3.1. DISEÑO DEL PROTOTIPO A IMPLEMENTAR

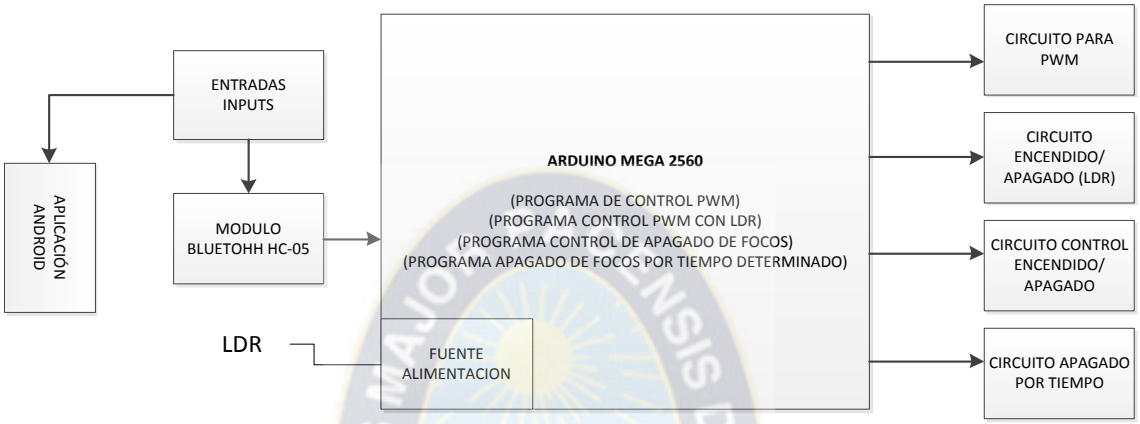


Figura 6. Diagrama de Bloques
Fuente: Elaboración propia

A continuación describimos los distintos bloques para su implementación**:**

3.1.1. LA APLICACIÓN

El celular tendrá un programa en android (.apk) para poder controlar el funcionamiento de los circuitos propuestos, se conectara de manera inalámbrica a través del módulo bluetooth.

Lo que haremos en esta aplicación es configurar una app Android ya hecha, para controlar el PWM que estará conectado en el Arduino comunicados a través de Bluetooth. Primero estudiaremos los datos que nos envía la app explicando su proceso. Después haremos el código de Arduino y explicaremos los aspectos más importantes de este programa.

El diseño esquemático es el siguiente:

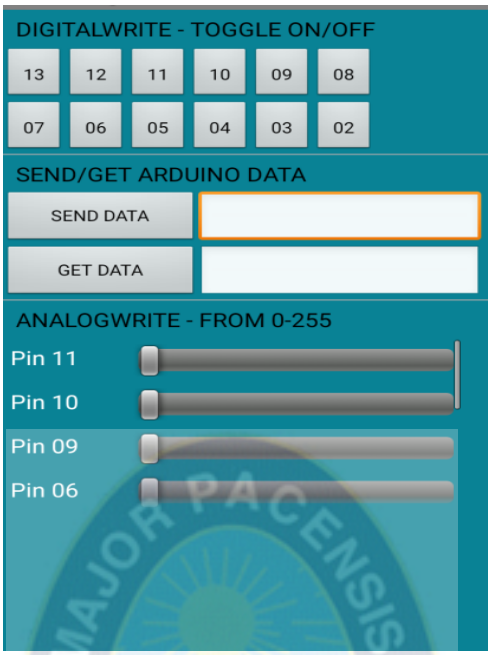


Figura 7. Aplicación Android
Fuente: Elaboración propia

El programa hace las siguientes funciones para cada botón:

- Los botones 12, 11, 10, 09, 07, 06, 05, 04, 03 para el control de iluminación, y sus correspondientes botones de encendido y apagado.
- Los botones 13 y 02 son funciones especiales, el botón 13 enciende y apaga todos los focos o salidas digitales 12 – 03. Y el botón 02 da paso al bucle de control del ambiente ósea a través de un sensor nos da la posibilidad de regular la intensidad lumínica.
- Los botones de SEND y GET: A través de SEND enviaremos el tiempo de apagados de los leds, esto en segundos. Y el GET nos muestra los valores que recibe externamente nuestro pin A0 donde está conectado el LDR.
- Los sliders 11,10, 09, 06, 05 y 03 son el control que se hace con PWM al conjunto de leds que representan un foco normal, con esto podemos regular la potencia de las luminarias de acuerdo a las especificaciones externas que se nos dan.

También cabe recalcar que el control de presencia se desactiva presionando el botón de reset externo que tiene el Arduino, ya que el control de presencia requiere de la constante atención a lo que sucede a su entrada analógica A0, y además entra en un bucle, con lo cual vuelve a su estado actual para poder ser manipulado con sus otras características.

3.1.2. COMUNICACIÓN ENTRE EL CELULAR Y LA PLACA ARDUINO

El módulo Bluetooth es programado para poder comunicar el celular con la placa Arduino, y así reciba instrucciones del celular para que la placa Arduino trabaje.

A continuación se muestra el código para programar el Arduino el cual esta comentado y ya estudiado para su mejor comprensión. Podemos notar los siguientes puntos importantes:

- Se definen los caracteres y números que nos enviara la aplicación para poder trabajar con ellos durante todo el programa.
- Hay tres variables importantes dentro del programa: ard_comand, pin_num y pin_value, los cuales nos dan los valores con los cuales trabajaremos durante todo el programa.
- Hay una función especial: set_digitalwrite(pin_num, pin_value), en el cual podemos introducir una función especial que queramos que realiza al oprimir un botón del 13 – 02.

```
/*
 PROYECTO: TRABAJO DE APLICACION
 PROGRAMADOR: original escrito por  Hazim Bitar y configurado por José Martin
 FECHA: 16 de Noviembre
 LICENCIA: Publica
*/

#define CARACTER_INICIO_CMD '*'          //Carcater que inicia el programa
#define CARACTER_FINAL_CMD '#'           //Caracter que acaba con el envio de
datos desde el celular
#define CARACTER_DIV_CMD '|'             //Caracter que divide el texto enviado
desde el celular
#define ESCRITURA_DIGITAL_CMD 10         //Caracter para iniciar con la escritura
Digital de pines (12-03)
```

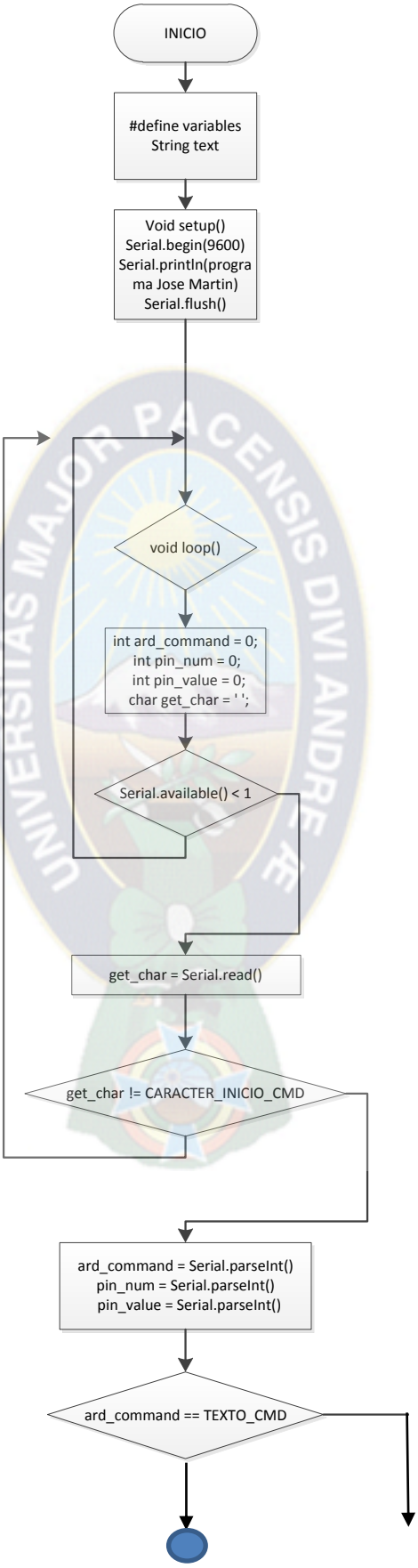
```
#define ESCRITURA_ANALOGA_CMD 11          //Caracter para iniciar con la
escritura analogica (dimmer)(11,10,09,06,05,03)
#define TEXTO_CMD 12                      //Caracter que inicia el envio de datos del
celular al arduino
#define LECTURA_ARDUDROID_CMD 13          //Caracter que inicia el envio de
datos desde el arduino al celular
#define MAX_COMMAND 20
#define MIN_COMMAND 10
#define LONGITUD_ENTRADA_STRING 40
#define ESCRITURA_ANALOGICA_MAX 255
#define PIN_ALTO 3                        //Caracter 3 enciende un led en digital write
#define PIN_BAJO 2                        //Carcater 2 apaga el led en digital write

String inText;                            //nos servira para enviar dato nuevo de apagado



void setup() {
  Serial.begin(9600);                     //Empieza la comunicacion serial
  Serial.println("programa jose martin");     //por consola imprimir y espaciar "Programa
jose martin"
  Serial.flush();                         //esperar a que complete la transmision serial
}

void loop()
{
  Serial.flush();                         //esperar a que complete la transmision serial

  int ard_command = 0;                    //variables que se repetiran cada vez que enviamos
una instruccion
  int pin_num = 0;
  int pin_value = 0;
  char get_char = ' ';                    //lee serial

// esperar a que los datos entren
  if (Serial.available() < 1) return;     // si no hay datos en el serial retornar al Loop().

// analizando entrada de indicador de inicio de comando
  get_char = Serial.read();

  if (get_char != CARACTER_INICIO_CMD) return; // si no hay indicación de inicio
del sistema, volver  loop ().

// analizando tipo de acomando entrante
  ard_command = Serial.parseInt();        //Leer el comando

// analizando el tipo de comando entrante
```

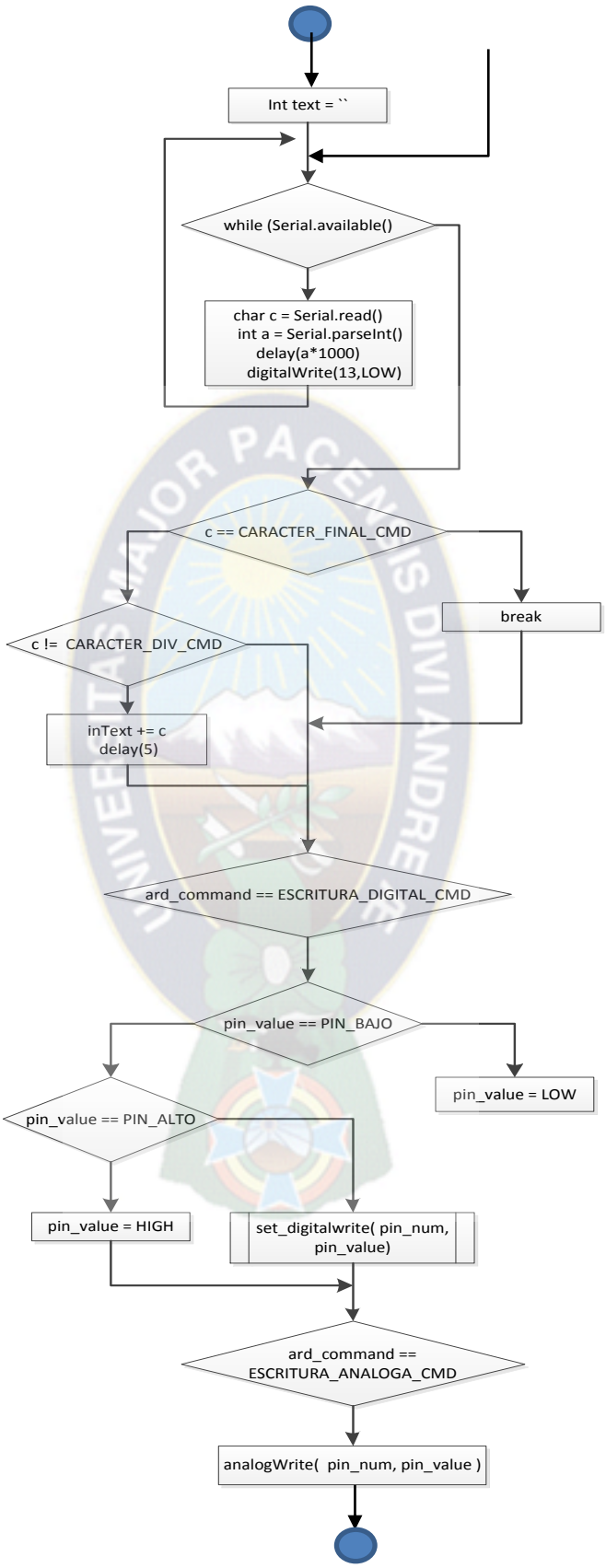
```
pin_num = Serial.parseInt();              // leer el pin
pin_value = Serial.parseInt();            // leer el valor


// 1) OBTENER COMANDO DE TEXTO PARA ARDUDROID

if (ard_command == TEXTO_CMD){
  inText ="";                         // borra variable para nueva entrada
  while (Serial.available())  {
    char c = Serial.read();               // recibe un byte de la memoria intermedia serie
    Serial.println(c);
    int a = Serial.parseInt();            // recibe el dato enviado desde la aplicacion

    Serial.println(a);
    delay(a*1000);                    // espera el tiempo mandado por la aplicacion

      digitalWrite(13,LOW);               // apagamos todos los leds
      digitalWrite(12,LOW);
      digitalWrite(11,LOW);
      digitalWrite(10,LOW);
      digitalWrite(9,LOW);
      digitalWrite(8,LOW);
      digitalWrite(7,LOW);
      digitalWrite(6,LOW);
      digitalWrite(5,LOW);
      digitalWrite(4,LOW);
      digitalWrite(3,LOW);
      digitalWrite(2,LOW);

    if (c == CARACTER_FINAL_CMD) { // si la cadena completa ha sido leida
      break;
    }
    else {
      if (c !=  CARACTER_DIV_CMD) {
        inText += c;
        delay(5);
      }
    }
  }
}

// 2) OBTENER DATOS DE digitalWrite (ENCENDIDO Y APAGADO DE
FOCOS)Y L DESDE LA 12 HASTA 03

if (ard_command == ESCRITURA_DIGITAL_CMD){
  if (pin_value == PIN_BAJO) pin_value = LOW;
  else if (pin_value == PIN_ALTO) pin_value = HIGH;
```

```
    else return; // error en el valor de PIN. regresar.

    set_digitalwrite( pin_num,  pin_value);  // al select case de los pines
    return;  // regrese al inicio de loop()
  }

  // 3) DIMMERS analogWrite (PWM)(11,10,09,06,05,03)
  if (ard_command == ESCRITURA_ANALOGA_CMD) {
    analogWrite(  pin_num, pin_value );
    return;  // regrese al loop();
  }

  // 4) Enviar datos a la aplicacion
  if (ard_command == LECTURA_ARDUDROID_CMD) {

    Serial.print(" Luxes = ");
    Serial.println(analogRead(A0));  // Leer y enviar valor analógico del Pin de Arduino
    return;  // regrese al loop();
  }
}

// 2a) seleccionar el pin # solicitado para la acción digitalWrite
void set_digitalwrite(int pin_num, int pin_value)
{
  switch (pin_num) {

  case 13:

        pinMode(13,OUTPUT);
        pinMode(12,OUTPUT);
        pinMode(11,OUTPUT);
        pinMode(10,OUTPUT);
        pinMode(9,OUTPUT);
        pinMode(8,OUTPUT);
        pinMode(7,OUTPUT);
        pinMode(6,OUTPUT);
        pinMode(5,OUTPUT);
        pinMode(4,OUTPUT);
        pinMode(3,OUTPUT);

        digitalWrite(13, pin_value);
        digitalWrite(12, pin_value);
        digitalWrite(11, pin_value);
        digitalWrite(10, pin_value);
        digitalWrite(9, pin_value);
        digitalWrite(8, pin_value);
        digitalWrite(7, pin_value);
```

```
    digitalWrite(6, pin_value);
    digitalWrite(5, pin_value);
    digitalWrite(4, pin_value);
    digitalWrite(3, pin_value);

  break;

case 12:
  pinMode(12, OUTPUT);
  digitalWrite(12, pin_value);
  break;

case 11:
  pinMode(11, OUTPUT);
  digitalWrite(11, pin_value);
  // adicione su código aquí, para este número de pin del Arduino
  break;

case 10:
  pinMode(10, OUTPUT);
  digitalWrite(10, pin_value);
  // adicione su código aquí, para este número de pin del Arduino
  break;

case 9:
  pinMode(9, OUTPUT);
  digitalWrite(9, pin_value);
  // adicione su código aquí, para este número de pin del Arduino
  break;

case 8:
  pinMode(8, OUTPUT);
  digitalWrite(8, pin_value);
  // adicione su código aquí, para este número de pin del Arduino
  break;

case 7:
  pinMode(7, OUTPUT);
  digitalWrite(7, pin_value);
  // adicione su código aquí, para este número de pin del Arduino
  break;

case 6:
  pinMode(6, OUTPUT);
  digitalWrite(6, pin_value);
  // adicione su código aquí, para este número de pin del Arduino
  break;
```

```
  case 5:
    pinMode(5, OUTPUT);
    digitalWrite(5, pin_value);
    // adicione su código aquí, para este número de pin del Arduino
    break;

  case 4:
    pinMode(4, OUTPUT);
    digitalWrite(4, pin_value);

    break;

  case 3:
    pinMode(3, OUTPUT);
    digitalWrite(3, pin_value);

    break;

  case 2:

LDR3:
   pinMode(11, OUTPUT);
   pinMode(10, OUTPUT);
   pinMode(9, OUTPUT);
   pinMode(6, OUTPUT);
   pinMode(5, OUTPUT);
   pinMode(3, OUTPUT);


    analogWrite(11,analogRead(A0)/3);
    analogWrite(10,analogRead(A0)/3);
    analogWrite(9,analogRead(A0)/3);
    analogWrite(6,analogRead(A0)/3);
    analogWrite(5,analogRead(A0)/3);
    analogWrite(3,analogRead(A0)/3);

    delay(500);

    Serial.print(" Luxes = ");
    Serial.println(analogRead(A0));
    goto LDR3;
    break;
    // por defecto, si nada más fue seleccionado, hacer el defecto (default) default es
opcional
  }
}
```


INICIO
#define variables
String text
Void setup()
Serial.begin(9600)
Serial.println(programa Jose Martin)
Serial.flush()
void loop()
int ard_command = 0;
int pin_num = 0;
int pin_value = 0;
char get_char = ' ';
Serial.available() < 1
get_char = Serial.read()
get_char != CARACTER_INICIO_CMD
ard_command = Serial.parseInt()
pin_num = Serial.parseInt()
pin_value = Serial.parseInt()
ard_command == TEXTO_CMD


Int text = ``
while (Serial.available()
char c = Serial.read()
int a = Serial.parseInt()
delay(a*1000)
digitalWrite(13,LOW)
c == CARACTER_FINAL_CMD
break
c != CARACTER_DIV_CMD
inText += c
delay(5)
ard_command == ESCRITURA_DIGITAL_CMD
pin_value == PIN_BAJO
pin_value = LOW
pin_value == PIN_ALTO
pin_value = HIGH
set_digitalwrite(pin_num, pin_value)
ard_command == ESCRITURA_ANALOGA_CMD
analogWrite(pin_num, pin_value)

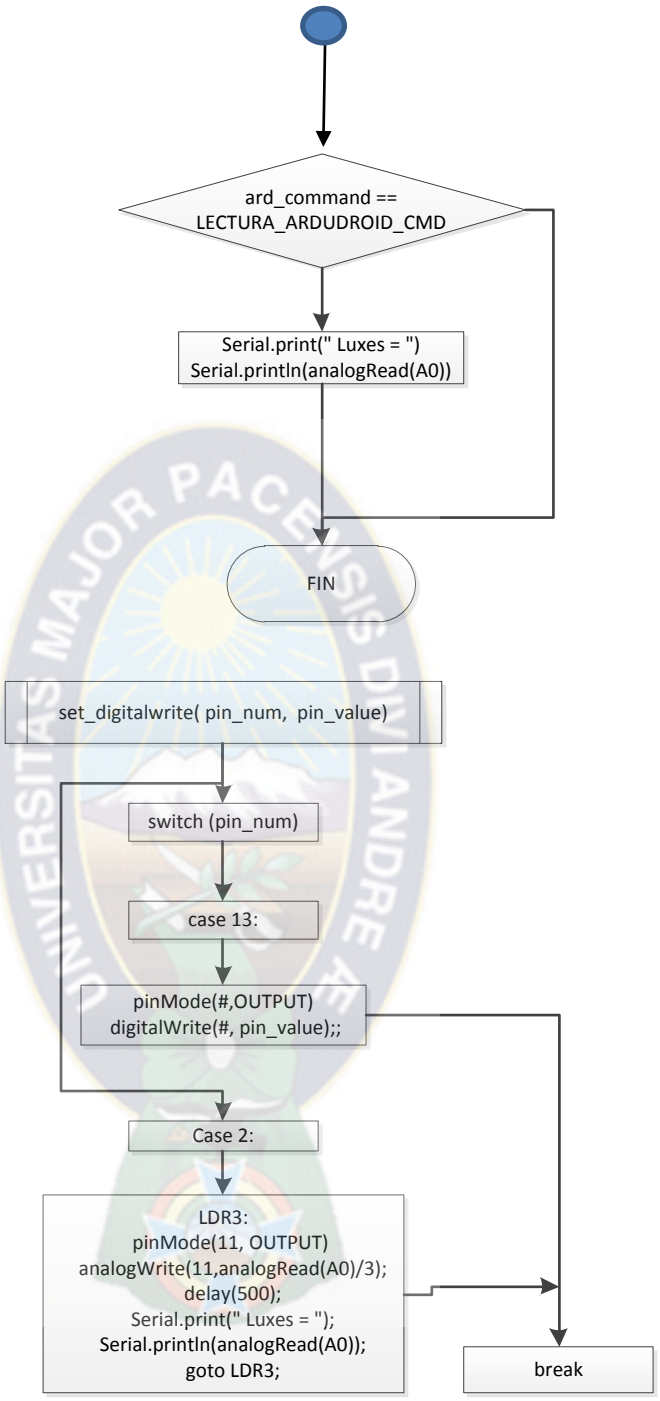


Figura 8. Diagrama de flujo del programa de Arduino
Fuente: Elaboración propia

3.1.3. LA PLACA ARDUINO Y CIRCUITOS

La placa Arduino es el corazón del proyecto ya que este a través de tres salidas en modo PWM, ósea manda pulsos digitales hacia los circuitos que controlan los focos.

Los principales procedimientos que hace el Arduino son:

- **Control en función de la luz natural.** Con el sensor de luz natural, la intensidad de la luz artificial cambia en función de la luz natural incidente. Esto se logra por un LDR que por un divisor de tensión logra darnos un voltaje entre 0 (V) hasta los 5 (V) aproximadamente, esta variaciones en el LDR nos da los voltajes que leerá el puerto Analógico/Digital de la placa Arduino, más abajo se detallan los cálculos que se hicieron para poder obtener los voltajes.

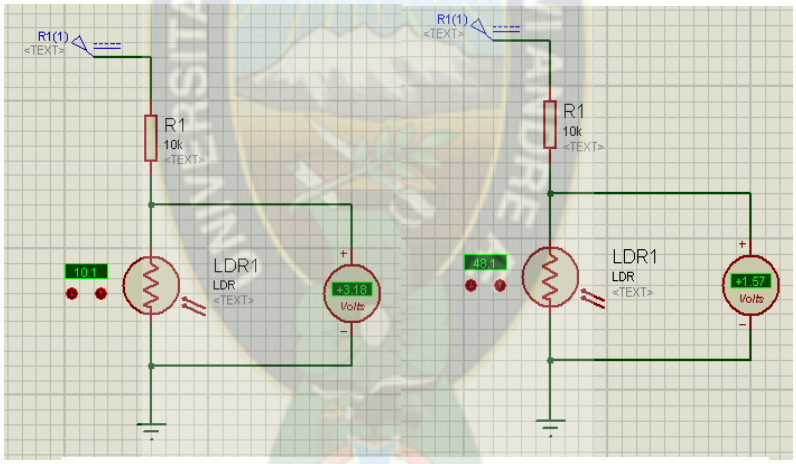


Figura 9. Circuito del detector de presencia
Fuente: Elaboración Propia – ISIS Proteus

Para la implementación de esta función se tiene que tener presente el LDR como sensor de presencia, tiene las siguientes características:

En obscuras en LDR indica una resistencia de 16 kΩ
En luz en LDR indica una resistencia de 300 Ω
Hay un voltaje de 5 (V)
Y la corriente que entrega la fuente es 20 m(A) aproximadamente.

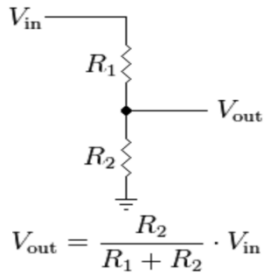


Figura 10. Circuito divisor de tension
Fuente: Elaboración Propia – ISIS Proteus

Como vemos en la figura 10, es el circuito que debemos armar para poder tener nuestro control de luminosidad mediante LDR.

Es un divisor de voltaje, en cual podemos notar que nuestro LDR o resistencia variable es la resistencia R2.

Vout ira conectada a la entrada analógica A0 del Arduino Mega.

Los Arduino entregan un voltaje 0 a 5 voltios, pero los recibe en valores discretos que van del 0 – 1024, el problema aquí es que para escribir en la sentencia *analogWrite(Pin,valor)* necesitamos que el valor este entre los 0 – 255 de valor, y la entrada A0 nos da un valor más grande.

La solución a esto es que cuando se lea el puerto analógico a través de la sentencia *analogRead(pin)* se lo divida entre cuatro, así tendremos un valor entre 0 – 255 aproximados.

También al realizar este tipo de sensores debemos de tomar en cuenta que la luz varía de acuerdo al lugar en donde nos encontremos, y puede ser que tengamos más luz o menos luz y la intensidad de los focos sea más alta o menos alta, por lo que se hizo tres divisiones se dividió entre dos, entre tres y entre cuatro para que adapte a las distintos ambientes.

- **Adaptación a la tarea,** Evitará la fácil adaptación a los niveles de iluminación estándar, en función de tareas o aplicaciones concretas.

En esta tarea que realiza nuestro controlador podemos ver que hay tres entradas que son potenciómetros simulando lo que serán las entradas que estarán conectadas a nuestro modulo a través del módulo bluetooth, estos variaran el voltaje entre 0 (V) – 5 (V) con los cuales habrá 256 voltajes distintos ósea el Arduino contara desde 0 hasta el 255 cada cual equivaldrá a una cierta potencia lumínica en las salidas (PWM) a los cuales está conectado las lámparas Leeds.

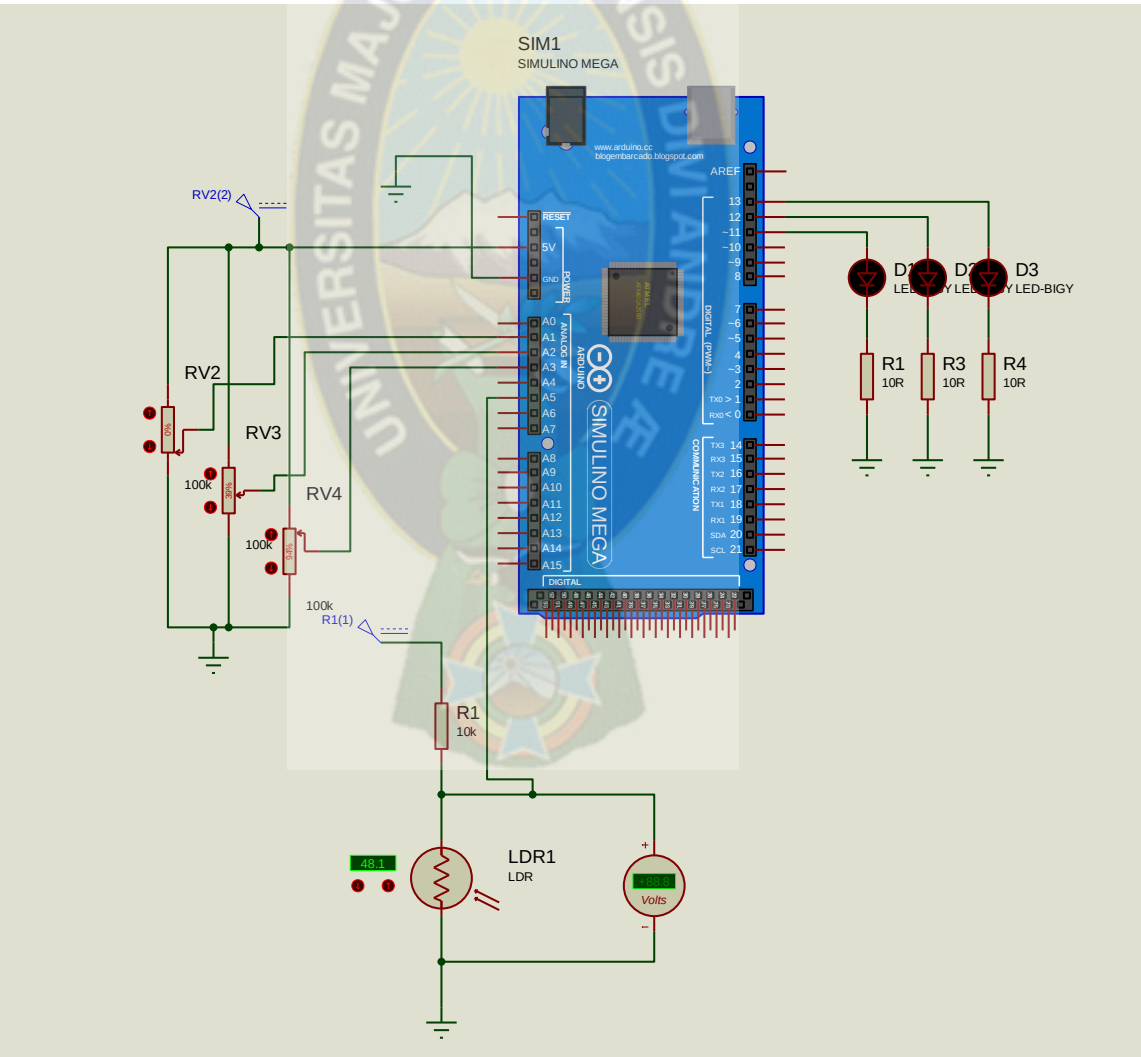


Figura 11. Circuito control de potencia lumínica y detector de presencia
Fuente: Elaboración Propia – ISIS Proteus

3.1.4. CONEXIONES DEL BLUETOOTH Y COMUNICACIÓN SERIAL

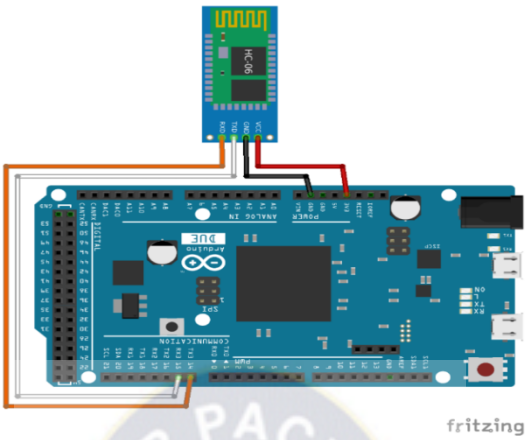


Figura 12. Conexión del módulo Bluetooth
Fuente: Elaboración Propia – Fritzing

Para la conexión por Bluetooth, Arduino lo hace a través de su módulo serial, por lo que necesita 4 entradas las cuales son: 5 V, GND, Tx, Rx, la transmisión y recepción deben ir en Rx-Tx y Tx-Rx, ya que al momento de comunicarse con el Arduino, lo hace con este protocolo UART que es bidireccional y se necesita de este tipo de conexión, además que se utilizan los comandos:

Serial.begin(9600);	Velocidad e inicio de la comunicación
Serial.Availabe();	Muestra la cantidad de bytes en el buffer de comunicaciones
Serial.flush();	Espera a que termine la comunicación serial
Serial.Read();	Lee el caracter del buffer de comunicación
Serial.parseInt();	Lee la siguiente instrucción

3.1.5. FOCO DE LEDS

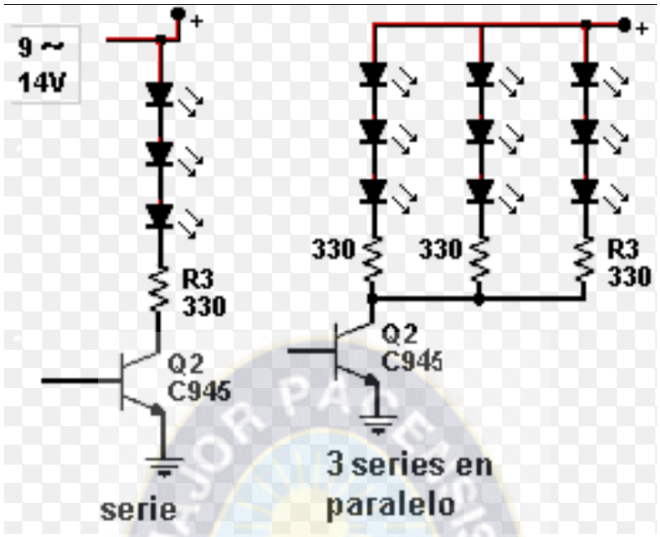


Figura 13. Circuito FOCO DE LEDS
Fuente: www.arduino/forum.cc

Se hace un arreglo de leds de alto brillo para representar lo que es un foco real, aunque últimamente se ve que se usa este tipo de tecnología porque es más eficiente con el ahorro de energía, además que brinda mayor manejabilidad en cuanto al manejo de circuitos externos para poder controlar.

El circuito que es mostrado en la figura 11 nos muestra que debemos conectar todos los leds al colector del transistor, esto debido a que se está controlando a través de un puerto de Arduino y este puerto solo nos da una corriente de alrededor 20 miliamperios, no es suficiente, lo que hace el transistor es amplificar la corriente y a la vez nos ofrece una protección del Arduino y de los leds de alto brillo.

Las formulas usadas para dimensionar este foco de leds son las siguientes:

Voltaje de cada led = 1.5(V)

Corriente del adaptador = 600 m(A)

Estamos conectando a una fuente de 20 (V).

Entonces tenemos

R = 20 - (4*0.75) / 350m(A) = 50 (Ω) aproximadamente

3.1.6. BOTON DE RESET EXTERNO.

Este pin de *reset* ha sido elaborado para poder *resetear* el arduino de forma que no utilicemos el reset que viene ya instalado por defecto dentro de la placa.

Al ser un pin se puede acomodar fuera de la placa de arduino, la conexión es sencilla como se ve, solo se necesita hacer un corto circuito entre pin RESET y el GND del Arduino. Al presionarlo ocurre el reset, solo necesita que este apretado 2,5 u(s) aproximadamente.

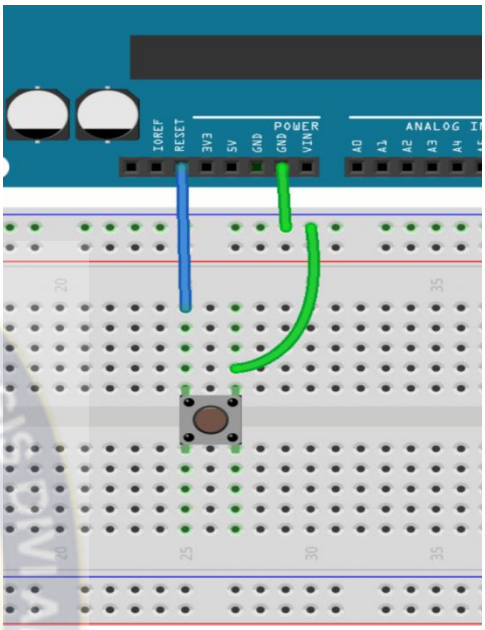


Figura 14. Conexión Reset externo
Fuente: www.electrotec.com Fritzing

3.5. DIAGRAMA BÁSICO DE LA SALA DE RECUPERACIÓN DE UN HOSPITAL

En función de las necesidades del paciente, podemos crear tres escenas lumínicas diferentes: luz ambiente, luz para curas y luz de lectura,

Por su estética y superficies lisas, facilita las tareas de limpieza y desinfección, mientras que por su funcionalidad y flexibilidad se adapta a las necesidades concretas de cada proyecto.

En el presente proyecto se intentara modelar una sala de recuperación en donde se implementara los focos leds para poder representar como es aplicable este proyecto, por lo cual podemos las siguientes figuras a continuación poder comprender mejor lo que se quiere realizar:

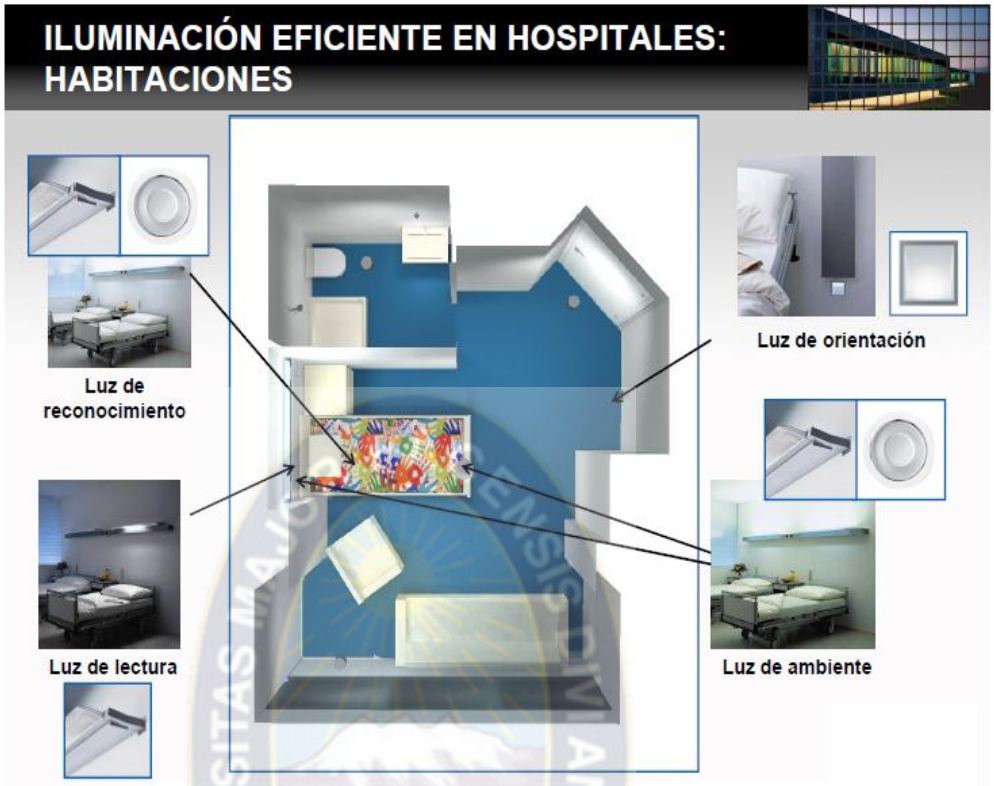


Figura 15. Iluminación eficiente
Fuente: Grupo Lledó

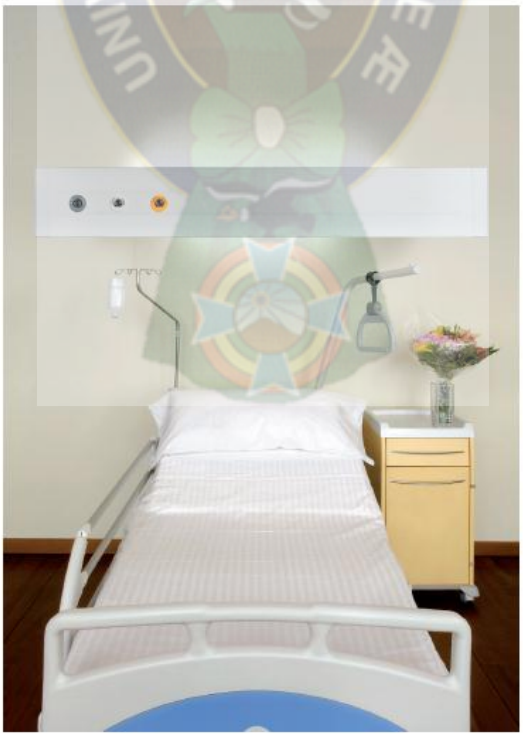


Figura 16. Posición de los focos
Fuente: Empresa LAMP HEADQUARTER

3.6. MATERIALES Y CIRCUITO PARA IMPLEMENTAR EL PROTOTIPO DE CONTROL DE ILUMINACION

Los materiales que se utilizan son:

- Arduino MEGA 2560
- Modulo Bluetooth HC-05 (Slave)
- Leds de alto brillo
- Resistencias entre 100 Ω a 10 KΩ
- Fotoresistencia (LDR)
- Conectores Arduino
- Adaptador 5(V) para Arduino
- Adaptador 12 - 20 (V) para leds
- Transistores 2N2222A (NPN)
- Cables de interconexión
- Cable USB para Arduino
- Pulsador

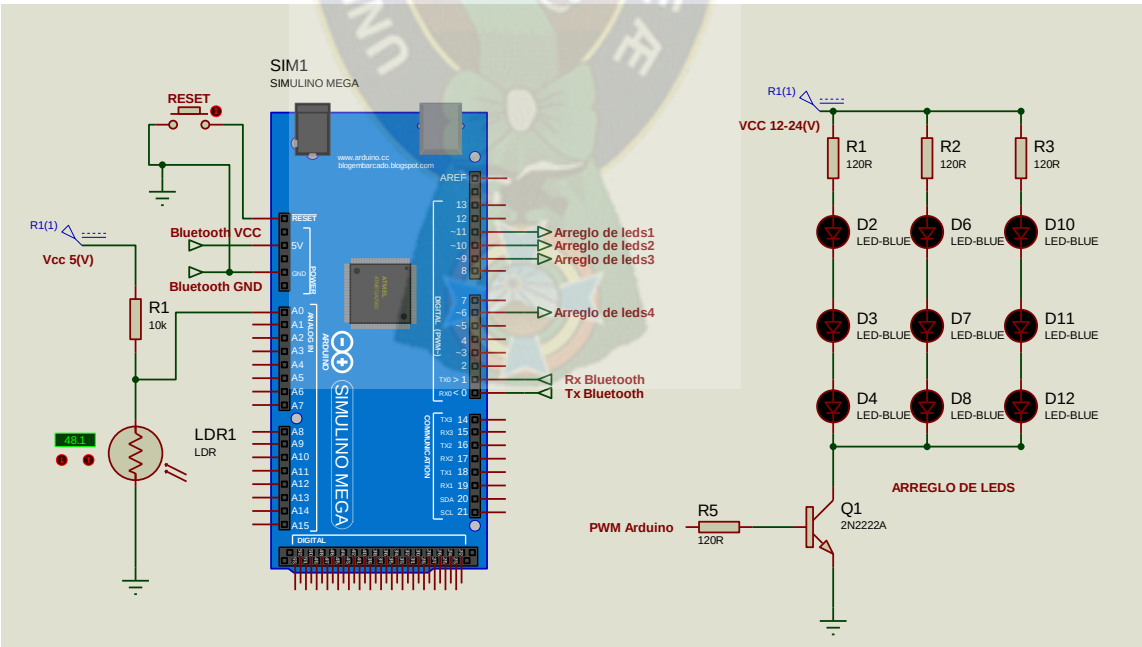


Figura 17. Circuito de implementacion
Fuente: Elaboración Propia ISIS Proteus

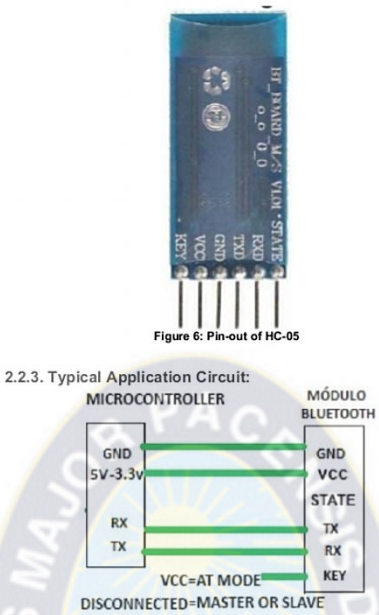


Figura 18. Conexión del módulo Bluetooth
Fuente: Monografias.com

A continuación se detalla lo que se quiere realizar en un esquema más generalizado del proyecto, para poder entender las funciones que queremos que realice el Arduino, los focos representan el arreglo de leds.

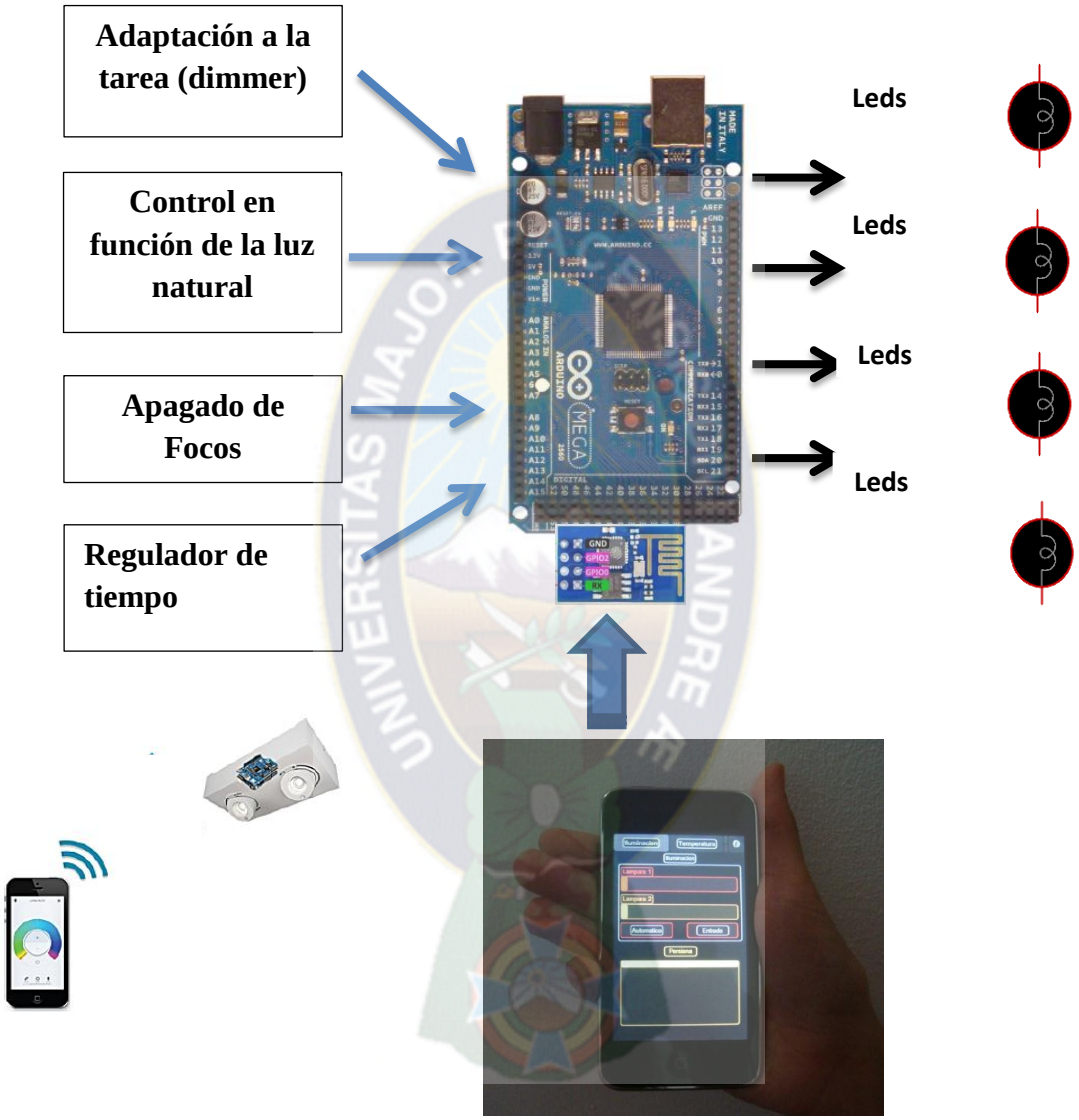


Figura 19. Imagen del Proyecto
Fuente: Elaboración propia

Imágenes del prototipo funcionando, tanto con leds normales y leds de alto brillo

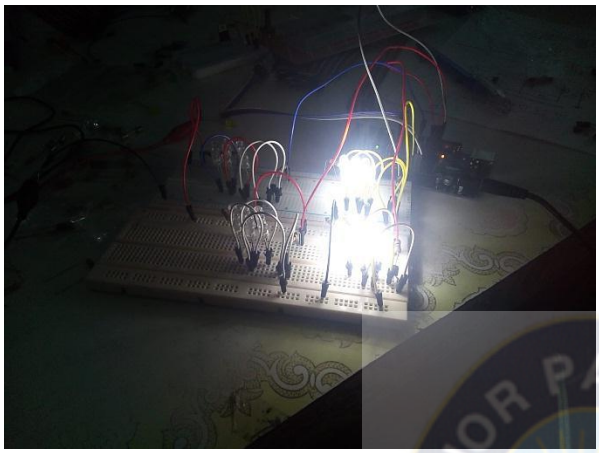


Foto1. Imagen del Proyecto conjunto de leds alto brillo 2 apagados/2 encendidos
Fuente: Elaboración propia

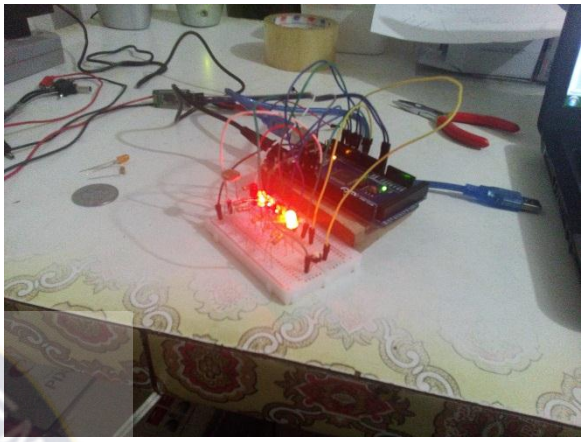


Foto2. Imagen del Proyecto conjunto de leds
Fuente: Elaboración propia

Foto2. Imagen del Proyecto conjunto de leds todos prendidos
Fuente: Elaboración propia

3.7. COSTOS DEL PROTOTIPO

Descripción	Unidad	Precio Unitario (Bs)	Cantidad	Precio Total (Bs)
Arduino MEGA 2560	Equipo	140	1	140
Modulo Bluetooth HC-05 (Slave)	Equipo	80	1	80
Leds de alto brillo	Unidad	1	54	54
Resistencias entre 100 Ω a 10 KΩ	Unidad	0.25	24	6
Fotoresistencia (LDR)	Unidad	5	1	5
Conectores Arduino	Tira	30	3	90
Adaptador 5(V) para Arduino	Equipo	50	1	50
Adaptador 12 - 20 (V) para leds	Equipo	70	1	70
Transistores 2N2222A (NPN)	Unidad	2	6	12
Cables de interconexión	Metro	3	10	30
Cable USB para Arduino	Unidad	10	1	10
Pulsador	Unidad	1	1	1
Placas huecas para soldar	Unidad	3	6	18
Estaño	Metro	3	3	9
TOTAL				**575**

Los focos leds pueden ser reemplazados por lámparas o focos de corriente continua, pero teniendo en cuenta que debemos de poner resistencia y un transistor de potencia media, ose un transistor tipo TIP y una resistencia que disipe una potencia de 1 (W).

CAPITULO IV

CONCLUSIONES Y RECOMENDACIONES

4.4. CONCLUSIONES

El uso de este tipo de control en una sala de espera de hospital mejoraría tanto el aspecto como el ahorro de energía, y más aún mejoraría la atención al paciente que está en recuperación, haciéndolo sentir cómodo, que es crucial para su satisfactoria recuperación.

Dentro de los conceptos electrónicos podemos decir que la modulación de ancho de pulso es sencilla de implementar en sistemas de corriente continua, a la vez que este tipo de tecnología es la que dominara por su eficiencia tanto en el consumo de energía como en las formas que podemos controlarlo, es más factible que la corriente alterna, ya que para esta se necesita circuitos alternos que salen más caros, y además que no se adaptan muy bien al periodo de vida de los focos incandecentes.

Al realizar este prototipo pudimos descubrir una amplia variedad de control de iluminación así como sus múltiples aplicaciones que esto conlleva. Si otros sistemas utilizaran este tipo de tecnología se puede esperar una nueva forma de vida más sencilla donde todos pueden controlar las luces e incluso nos podemos conectar a otras plataformas que nos permitan controlar estos circuitos sin estar presente.

En este diseño de prototipo se mostró la manera en que se puede controlar un sistema de iluminación mediante la combinación de una placa Arduino con cualquier dispositivo con acceso inalámbrico. Los resultados fueron satisfactorios logrando el control del encendido y apagado de la lámpara, pero se desea que en un futuro el sistema cuente con múltiples funciones.

4.5. RECOMENDACIONES

Para el uso de focos incandescentes, se necesita un control de potencia, y dos pines para poder controlar de manera adecuada este foco, un optoacoplador que servirá de acople entre la corriente alterna y la corriente continua, además que en el Arduino se debe programar a través de interrupciones una señal de paso por cero el cual activara el pwm, es un poco más detallados dependerá mucho del criterio del que arme el circuito.

Debemos tener presente que al tener un módulo Bluetooth podemos conectarnos a el con un programa adecuado, a la vez que debe el programa con el cual trabajara el modulo Bluetooth debe ser grabado cuando este módulo no esté conectado al Arduino.

Al probar circuitos en corriente alterna se debe tener cuidado de no tocar la parte de potencia, ya que es voltaje alto y una intensidad de corriente que es capaz de matar.

Al conectar varios leds de brillo se debe tomar en cuenta la corriente de nuestra fuente, así como las resistencias para poder controlarlas, ya que podríamos quemar estos leds, además la polarización del transistor para que este entre en corte o saturación adecuadamente.

4.6. BIBLIOGRAFIA

- Enriquez H., Rafael (2009) Guía de usuario de Arduino San Francisco California – USA Universidad de Córdoba recuperado de: http://creativecommons.org/licenses/by-nc-sa/3.0/
- http://diymakers.es/crear-app-para-arduino-con-app-inventor/
- http://diymakers.es/crear-app-para-arduino-con-app-inventor/
- http://electronicaentidadoscura.blogspot.com/2013/09/control-de-luminosidad-con-slider-en.html
- http://www.cun.es/diccionario-medico/terminos/sala-recuperacion
- http://www.instructables.com/id/Control-LED-Using-MIT-App-Inventor-and-Arduino/
- http://www.proyectoelectronico.com/leds/como-conectar-leds.html
- https://es.wikipedia.org/wiki/Conjunto_de_comandos_Hayes
- https://nergiza.com/iluminacion-portatil-a-12v-con-baterias-todo-lo-que-hay-que-saber/
- https://www.element14.com/community/thread/37707/l/app-inventor-bluetooth-and-arduino-part-3-android-slider-controls-led-brightnes?displayFullThread=true
- https://www.prometec.net/circuito-multiples-leds/
- Kraft, Oscar (1998) *Comunicación Inalámbrica* Recuperado de: http://catarina.udlap.mx/u_dl_a/tales/documentos/lmt/padilla_m_o/capitulo4.pdf.
- LAMP HEADQUARTER (2010) *Sistemas de iluminación para sanitarios* Cordoba – España Lamp Light recuperado de : www.lamp.es
- LLedo Iluminacion (2010) *Intslacaiones de iluminación y funcionalidad y eficiencia* recuperado de: www.lledosa.com
- Torrente A., Oscar (2013) *Arduino Curso Practico de formación* Ciudad de Mexico – Mexico Editores Alfa y Omega recuperado de: www.alfaomega.com.mx