



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

UNIDAD DE GESTIÓN DE  TECNOLOGÍAS

**DEPARTAMENTO DE ELECTRÓNICA Y
COMPUTACIÓN**

**CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN
& AVIÓNICA**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE TECNÓLOGO EN ELECTRÓNICA MENCIÓN
INSTRUMENTACIÓN & AVIÓNICA**

**TEMA: “REHABILITACIÓN DEL MARCADOR ELECTRÓNICO
MEDIANTE LEDS DE ALTO BRILLO Y MANDO DE
RADIOFRECUENCIA PARA LA UNIDAD DE GESTIÓN DE
TECNOLOGÍAS”**

AUTOR: TOAPANTA CAGUANA EDISON GEOVANNY

DIRECTOR: ING. JORGE PARDO

LATACUNGA

2017



**DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN
CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN &
AVIÓNICA**

CERTIFICACIÓN

Certifico que el trabajo de titulación, **“REHABILITACIÓN DEL MARCADOR ELECTRÓNICO MEDIANTE LEDS DE ALTO BRILLO Y MANDO DE RADIOFRECUENCIA PARA LA UNIDAD DE GESTIÓN DE TECNOLOGÍAS”** realizado por el señor **TOAPANTA CAGUANA EDISON GEOVANNY**, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar al señor **TOAPANTA CAGUANA EDISON GEOVANNY** para que lo sustente públicamente.

Latacunga, 13 de febrero del 2017

SR. ING. JORGE PARDO

DIRECTOR



**DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN
CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN &
AVIÓNICA**

AUTORÍA DE RESPONSABILIDAD

Yo, **TOAPANTA CAGUANA EDISON GEOVANNY**, con cédula de identidad N° 0503500605 declaro que este trabajo de titulación **“REHABILITACIÓN DEL MARCADOR ELECTRÓNICO MEDIANTE LEDS DE ALTO BRILLO Y MANDO DE RADIOFRECUENCIA PARA LA UNIDAD DE GESTIÓN DE TECNOLOGÍAS”** ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico de trabajo de grado en mención.

Latacunga, 13 de febrero del 2017

TOAPANTA CAGUANA EDISON GEOVANNY

C.I: 0503500605



**DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN
CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN &
AVIÓNICA**

AUTORIZACIÓN

Yo, **TOAPANTA CAGUANA EDISON GEOVANNY**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación **“REHABILITACIÓN DEL MARCADOR ELECTRÓNICO MEDIANTE LEDS DE ALTO BRILLO Y MANDO DE RADIOFRECUENCIA PARA LA UNIDAD DE GESTIÓN DE TECNOLOGÍAS”** cuyo contenido, ideas y criterios son de mi autoría y responsabilidad

Latacunga, 13 de febrero del 2017

TOAPANTA CAGUANA EDISON GEOVANNY

C.I: 0503500605

DEDICATORIA

A Dios por todas sus bendiciones recibidas a mis padres ya que si ellos no hubiera podido culminar mis carrera universitaria pero de manera especial dedico mi tesis a la persona que me dio fuerza y valor para llegar hasta donde estoy, a ti que desde el cielo me supiste guiar día a día con tus bendiciones gracias de todo corazón esta tesis va dedicada para Elizabeth Toapanta.

AGRADECIMIENTO

Agradezco a Dios por darme la vida y sabiduría, a mis padres porque gracias a ellos he podido estudiar una carrera universitaria, a la Universidad que me abrió las puertas para poder optar por una carrera y así para poder ser un profesional, a mis profesores por sus conocimientos impartidos día a día.

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN.....	ii
AUTORÍA DE RESPONSABILIDAD	iii
AUTORIZACIÓN.....	iv
DEDICATORIA	v
AGRADECIMIENTO	vi
ÍNDICE DE CONTENIDOS.....	vii
ÍNDICE DE TABLAS	xi
ÍNDICE DE FIGURAS.....	xii
RESUMEN.....	xiv
ABSTRACT.....	xv
CAPÍTULO I.....	1
PLANTEAMIENTO DEL PROBLEMA.....	1
1.1 ANTECEDENTES.....	1
1.2 PLANTEAMIENTO DEL PROBLEMA.....	2
1.3 JUSTIFICACIÓN.....	2
1.4 OBJETIVOS.....	3
1.4.1 Objetivo General	3
1.4.2 Objetivos Específicos	3
1.5 ALCANCE.....	3
CAPÍTULO II.....	4
MARCO TEÓRICO	4
2.1 MARCADOR ELECTRÓNICO	4
2.1.1 Marcador electrónico para básquet.....	4

2.2	ARDUINO	5
2.3	ARDUINO MEGA 2560.....	5
2.3.1	Programación.....	7
2.4	ENTORNO DE PROGRAMACIÓN DE ARDUINO.....	7
2.4.1	Estructura de un Sketch.....	7
2.4.2	Estructura setup().....	8
2.4.3	Estructura loop().....	8
2.4.4	Funciones.....	8
2.4.5	Entradas/Salidas digitales	9
2.4.6	Entradas/Salidas analógicas	10
2.4.7	Variables	10
2.4.8	Declaración de variables	10
2.4.9	Utilización de una variable	11
2.4.10	Tipos de datos	12
2.4.11	Entre llaves {}.....	13
2.4.12	Punto y Coma ;.....	13
2.4.13	Línea de comentario //	13
2.4.14	Constantes.....	13
2.4.15	High/Low	13
2.4.16	Input/Output	14
2.5	COMUNICACIÓN SERIE.....	14
2.6	LED.....	15
2.6.1	LEDs de alto brillo	15
2.6.2	Vida útil del led normal y led de alto brillo	16

2.6.3	Color.....	16
2.7	DECODIFICADORES.....	16
2.7.1	Decodificador 74LS48.....	17
2.8	DISPLAYS.....	17
2.8.1	Tipos de Displays.....	18
2.8.2	Display de 7 segmentos.....	19
2.9	MICROCONTROLADOR.....	20
2.9.1	El microcontrolador PIC16F877A.....	20
2.10	XBEE Y ZIGBEE.....	22
2.10.1	XBee-PRO RF Module RPSMA S1 2.4 GHz.....	22
2.10.2	Tipos de antenas.....	23
2.10.3	Característica del XBee pro.....	24
2.11	ANTENA RP-SMA 2.4GHz.....	25
2.12	TIP 127.....	26
2.13	TRANSISTOR NPN 2N3904.....	26
2.14	FUENTE DE ALIMENTACIÓN.....	27
2.14.1	Fuente de alimentación lineal y conmutada.....	27
CAPÍTULO III.....		29
DESARROLLO DEL PROYECTO.....		29
3.1	PRELIMINARES.....	29
3.2	REQUERIMIENTOS MÍNIMOS.....	29
3.3	ARMADO DEL TABLERO ELECTRÓNICO.....	30
3.4	DIAGRAMA DE LOS DECODIFICADORES 74LS48.....	35
3.5	IMPLEMENTACIÓN DE POTENCIA.....	36

3.5.1	Diagrama de potencia Darlington	36
3.6	IMPLEMENTACIÓN DEL CONTROL INALÁMBRICO.....	37
3.7	ARMADO DEL CONTROL INALÁMBRICO	38
3.7.1	Diagrama control de mando con el microcontrolador PIC16F877A.....	38
3.8	INSTALACIÓN DE LA FUENTE DE 24V 5A.....	39
3.9	FINALIZACIÓN DEL MARCADOR ELECTRÓNICO.....	40
3.10	DIAGRAMA DEL MARCADOR ELECTRÓNICO EN PROTEUS 8.1 42	
3.11	CONFIGURACIÓN DEL ARDUINO MEGA 2560	43
3.12	CÓDIGO DE PROGRAMACIÓN PARA EL TABLERO ELECTRÓNICO CON ARDUINO MEGA 2560, XBEE Y PIC 16F877A....	45
3.12.1	Funcionamiento del tablero electrónico	45
3.12.2	Código de programación de tiempo, puntos, faltas y periodo	46
3.13	CÓDIGO DE PROGRAMACIÓN DEL MICROCONTROLADOR PIC16F877A	47
3.13.1	Código de control inalámbrico.....	47
	CAPÍTULO IV.....	50
	CONCLUSIONES Y RECOMENDACIONES	50
4.1	CONCLUSIONES	50
4.2	RECOMENDACIONES.....	51
	GLOSARIO DE TÉRMINOS	52
	REFERENCIAS BIBLIOGRÁFICAS.....	53
	ANEXOS.....	¡Error! Marcador no definido.

ÍNDICE DE TABLAS

Tabla 1 Micro controlador Arduino AVR	6
Tabla 2 Características de los HBLEDs	16
Tabla 3 Las características principales del PICs16f877A.....	21
Tabla 4 Características del XBee pro S1.....	24
Tabla 5 Características de la Antena RP-SMA.....	25

ÍNDICE DE FIGURAS

Figura 1 Marcador electrónico multifunción	4
Figura 2 Marcador electrónico para básquet.....	4
Figura 3 Diferentes modelos de Arduino	5
Figura 4 Diagrama del Arduino Mega 2560	6
Figura 5 Leds de alto brillo	15
Figura 6 Decodificador de BCD 7 segmentos	17
Figura 7 Sistemas de indicadores de Display	18
Figura 8 Display de 7 segmentos.....	19
Figura 9 Tabla de la verdad del display de 7 segmentos	20
Figura 10 Distribución de pines del PICF877A	21
Figura 11 XBee-pro S1	23
Figura 12 Tipos de Antenas de los Xbee	24
Figura 13 Antena RP-SMA 2.4 GHz.....	25
Figura 14 Transistor Darlington tip127	26
Figura 15 Transistor 2N3904.....	27
Figura 16 Fuente Meanwell 120W 24V 5A.....	28
Figura 17 Tablero electrónico antiguo.....	30
Figura 18 Marco de display 7 segmentos	31
Figura 19 Segmentos desmontados	31
Figura 20 Componentes e mal estado	31
Figura 21 Desconexión del marco de display 7 segmentos	32
Figura 22 Juego de resistencias	32
Figura 23 Limpieza del cobre	33
Figura 24 Planchado del circuito	33
Figura 25 Limpiado del papel sobrante	33
Figura 26 Display de 7 segmentos.....	34
Figura 27 Colocación de las resistencias.....	34
Figura 28 Decodificador 74ls48.....	35
Figura 29 Decodificadores 74ls48.....	35
Figura 30 Circuito de potencia	36

Figura 31	Circuito Darlington.....	36
Figura 32	Instalación del circuito Darlington	37
Figura 33	Decodificador rehabilitado.....	37
Figura 34	Prueba de funcionamiento del XBee Pro S1	37
Figura 35	Conexión del módulo XBee Pro S1	38
Figura 36	Diagrama de control inalámbrico	38
Figura 37	Circuito impreso del control de mando.....	39
Figura 38	Suelda de componentes en la placa	39
Figura 39	Fuente de alimentación de 24V	40
Figura 40	Parte trasera del tablero electrónico	40
Figura 41	Parte delantera del tablero electrónico.....	41
Figura 42	Diagrama del tablero electrónico	42
Figura 43	Verificación del programa	43
Figura 44	Selección del puerto serial	44
Figura 45	Cargar el programa en la tarjeta Arduino	44

RESUMEN

El presente Proyecto Técnico tiene como objetivo la rehabilitación del marcador electrónico mediante leds de alto brillo y mando de radiofrecuencia para la Unidad de Gestión de Tecnologías. En el primer capítulo se detalla el planteamiento del problema, la justificación y los antecedentes, objetivos y alcance del proyecto técnico. En el segundo capítulo se desarrolla el marco teórico con las definiciones necesarias para la realización del proyecto técnico. El tercer capítulo detalla el proceso llevado a cabo para la rehabilitación del marcador electrónico con nuevas tecnologías, utilizando una placa Arduino Mega 2560, display 7 segmentos, decodificadores 74LS48 cátodo común, circuitos darlington y leds de alto brillo que por su tecnología ofrecen una alta iluminación a fin de mejorar la visibilidad en los encuentros deportivos que desarrolle la Unidad de Gestión de Tecnologías de la Universidad de las Fuerzas Armadas – ESPE; de igual manera se muestra el diseño e implementación del control inalámbrico para el manejo del marcador electrónico mediante módulos XBee que a través de un microcontrolador PIC16F877A y pulsadores realizan el reseteo, control de tiempo, periodo, y faltas del marcador electrónico. Finalmente el capítulo cuarto presenta las respectivas conclusiones y recomendaciones obtenidas durante el desarrollo del proyecto técnico.

PALABRAS CLAVES

TABLERO ELECTRÓNICO

ARDUINO MEGA 2560

PIC 16F877A

XBEE PRO S1

ABSTRACT

The present Technical Project aims at the rehabilitation of the electronic marker using high-brightness LEDs and radiofrequency control for the Technology Management Unit. The first chapter details the approach to the problem, the justification and the background, objectives and scope of the technical project. In the second chapter the theoretical framework with the necessary definitions for the realization of the technical project is developed. The third chapter details the process carried out for the rehabilitation of the electronic marker using new technologies, using an Arduino Mega 2560 board, 7 segment display, 74LS48 common cathode decoders, darlington circuits and high brightness leds that, because of their technology, offer high illumination In order to improve visibility in the sports meetings developed by the Technology Management Unit of the University of the Armed Forces (ESPE), the same way shows the design and implementation of wireless control for the electronic marker using XBee modules that Through a PIC16F877A microcontroller and pushbuttons perform the reset, time control, period, and electronic marker faults. Finally the fourth chapter presents the respective conclusions and recommendations obtained during the development of the technical project.

Keywords

Electronic scoreboard

Arduino mega 2560

Pic 16f877a

Xbee pro s1

Lic. Wilson E Villavicencio F MSc.

DOCENTE UGT

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

1.1 ANTECEDENTES

La Unidad de Gestión de Tecnologías de la Universidad de las Fuerzas Armadas – ESPE (Ex Instituto Tecnológico Superior Aeronáutico ITSA), brinda servicios educativos superiores a la juventud del país con carreras innovadoras, ofreciendo nuevos campos laborales en la industria aeronáutica y en la industria en general; la cual estimula a los estudiantes a la investigación de nuevas tecnologías.

Con el fin desarrollar actividades deportivas de manera interesante, la carrera de Electrónica mención instrumentación y aviónica del ex ITSA, desarrolló el trabajo de graduación del Sr. Luis Tana, cuyo tema fue “Implementación de un marcador electrónico para el departamento de cultura física del Instituto Tecnológico Superior Aeronáutico”, donde se plantea la necesidad de contar con un marcador electrónico para indicar el tiempo y el marcador de diferentes Deportes que se practican en el ITSA.

La universidad de San Buenaventura Bogotá la cual cuenta con un polideportivo y una cancha de football, debido a la necesidad de la universidad pretende diseñar un tablero electrónico deportivo multifuncional, para brindar al público presente y a los jugadores, información detallada de un certamen deportivo, diseñado y fabricado por alumnos de ingeniería electrónica. (Henao, 2008)

Referente al trabajo de titulación del Sr Ricardo Carrasco, tiene por objetivo diseñar un Marcador Electrónico con control inalámbrico para visualizar la información de resultados en el estadio central de Liga Deportiva Parroquial Atahualpa de la ciudad de Ambato, a través de displays luminosos. En la cual su prioridad es evitar la incertidumbre de los aficionados y demás asistentes durante el desarrollo de los encuentros deportivos. (Carrasco, 2012)

1.2 PLANTEAMIENTO DEL PROBLEMA

En la Unidad de Gestión de Tecnologías (exITSA) existe un marcador electrónico que no se encuentra operativo, puesto que el control de mando no funciona, así como los leds se encontraban quemados y algunos decodificadores en mal estado; por lo que se requiere una rehabilitación del mismo a fin de que la institución disponga de un marcador electrónico moderno para el desarrollo de los juegos deportivos que se realizan en cada periodo académico. Por lo que se analizaron dispositivos eficientes y de nuevas tecnologías como los leds de alto brillo, para mejorar la visualización del marcador electrónico en actividades realizadas en las canchas deportivas de la Institución observando el tiempo de juego, faltas, periodo de cada juego en un partido de básquet; todo esto controlado desde un módulo inalámbrico con los equipos de telemetría XBee-PRO, transmitiendo datos a largas distancias, fortaleciendo así el proceso de enseñanza-aprendizaje gracias a la ayuda de dispositivos de comunicación inalámbrica de última tecnología.

1.3 JUSTIFICACIÓN

El presente proyecto técnico se vio la necesidad de realizará un marcador electrónico, como su nombre lo indica es un visualizador que transmite lo que esta sucediendo durante un partido de básquet, con el fin de colaborar a la institución y a los alumnos para los próximos campeonatos (UGT); además fortalecerá el proceso de enseñanza-aprendizaje gracias a la ayuda de los dispositivos de última tecnología.

Debido a la necesidad que presenta la institución de no contar con un marcador electrónico fácil de utilizar para los encuentros deportivos, se implementa un control inalámbrico, para el cambio de tiempo, periodo, puntuación y faltas, que se encuentra establecidos dentro del marcador electrónico de básquet utilizando; módulos XBee-PRO para realizar la comunicación inalámbrica.

1.4 OBJETIVOS

1.4.1 Objetivo General

Rehabilitar el marcador electrónico mediante leds de alto brillo con control inalámbrico para visualizar información de resultados deportivos en la Unidad de Gestión de Tecnologías.

1.4.2 Objetivos Específicos

- Inspeccionar el estado actual del marcador electrónico para determinar los equipos que van a ser utilizados y los que deben adquirirse.
- Implementar un circuito de potencia, para conectar leds de alto brillo en el marcador electrónico.
- Implementar un sistema de control inalámbrico para el manejo del marcador electrónico mediante dispositivos o módulo XBee-PRO.

1.5 ALCANCE

La Unidad de Gestión de Tecnologías debe disponer de un Marcador Electrónico de básquet personalizado, para visualizar en las canchas deportivas el tiempo de juego, los resultados de cada equipo, el periodo y las faltas de los jugadores dentro de la institución o coliseos, posee con un control fácil de utilizar para el marcador electrónico mediante la comunicación inalámbrica, se enviara datos establecidos a una distancias de 1000mts, de igual manera controla el tiempo, periodo, faltas y un reset para volver al estado original del marcador.

CAPÍTULO II

MARCO TEÓRICO

2.1 MARCADOR ELECTRÓNICO

Los tableros digitales para campos deportivos son diseñados y creado, para observar el tiempo, marcador y faltas de un partido que se esté realizando, donde se está adquiriendo una importancia cada vez mayor, desde pequeños barrios hasta grandes ciudades. (Ferney Salazar Rojas, 2009)



Figura 1 Marcador electrónico multifunción

Fuente: (SAC, 2008)

2.1.1 Marcador electrónico para básquet

El marcador electrónico de básquet está constituido por el tiempo del partido, el periodo en la cual se juega, las anotaciones para cada equipo participante y faltas para cada jugador, los tableros para básquet están constituidos de metal. (Baloncesto, 2014)



Figura 2 Marcador electrónico para básquet

Fuente: (Rojas, 2009)

2.2 ARDUINO

Arduino es una plataforma de prototipos electrónica de código abierto (open-source) basada en hardware y software flexibles y fáciles de usar. Está pensado para artistas, diseñadores, como hobby y para cualquiera interesado en crear objetos o entornos interactivos. (Antony García González, 2013)

Existen diferentes modelos de Arduino, con múltiples características en cuanto a tamaño, formas, funciones y precios.

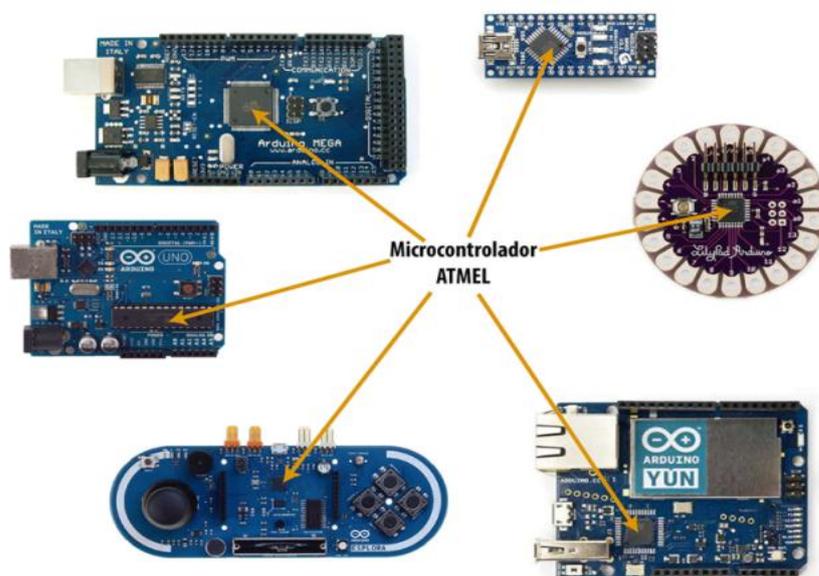


Figura 3 Diferentes modelos de Arduino

Fuente: (Antony García González, 2013)

2.3 ARDUINO MEGA 2560

Según la pagina wed Patagoniatec 2015 define lo siguiente.

Arduino Mega 2560 es una versión ampliada de la tarjeta original de Arduino, viene con más memoria para el programa, más RAM y más pines, se utiliza un potente procesador de AVR ATMEGA2560 con un amplio espacio de memoria para programar y corriendo a 16Mhz Anexo A.

Esta placa debido a su gran poder es utilizada para grandes proyectos, entre los más importantes se encuentran los de DOMOTICA e IMPRESORAS 3D, ya que lo más destacado es su elevada cantidad de pines de entrada y salida y sus 4 puertos UART por hardware.

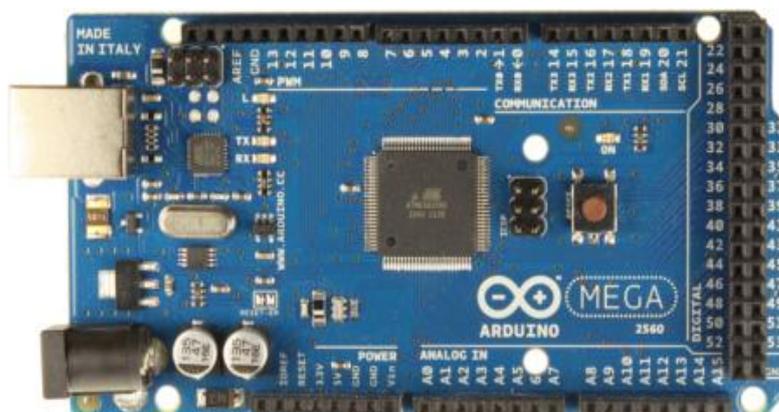


Figura 4 Diagrama del Arduino Mega 2560

Fuente: (González, 2013)

El Arduino Mega 2560 tiene 54 pines de entradas/salidas digital (14 de ellos pueden ser usados como salidas PWM). 16 entradas analógicas. 4UARTs (Puertos seriales por Hardware), un oscilador de 16Mhz, conexión USB, jack de alimentación, un conector ICSP y un botón de reset. Contiene todo lo necesario para funcionar, simplemente conéctelo al ordenador mediante un cable USB o aliméntalo utilizando un adaptador AC/DC o baterías para empezar a utilizar. (Patagoniatec, 2015)

Las características principales del Arduino Mega 2560 son:

Tabla 1

Arduino Mega 2560

Micro controlador	ATmega2560
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (limite)	6-20V
Entrada/Salida digitales	54(de los cuales 14 proporcionan
Entradas analógicas	16
Corriente continua para Pin E / S	40 mA
Corriente CC para Pin 3.3V	50 mA
Memoria flash	256 KB (de los cuales 8 KB utilizado
SRAM	8 KB

CONTINUA

EEPROM	4 KB
Velocidad de reloj	16 MHz

Fuente: (González, 2013)

Para un detalle pormenorizado de las características de la tarjeta Arduino, favor dirigirse al Anexo B.

2.3.1 Programación

El ATmega2560 en el Arduino Mega viene precargado con un gestor de arranque (bootloader) que permite cargar nuevo código sin necesidad de un programador por hardware externo. (Arduinodhtics, 2015)

2.4 ENTORNO DE PROGRAMACIÓN DE ARDUINO

2.4.1 Estructura de un Sketch

Según la página web de rduinostar se especifica lo siguiente.

Se denomina Sketch a una parte de código fuente listo para abrir con el entorno de desarrollo integrado de Arduino y ser cargado sobre nuestro dispositivo. Sobre él se escribirá todo el comportamiento que tendrá nuestro proyecto, incluyendo respuesta ante entradas determinadas, cálculos internos, salidas del sistema, etc.

Un sketch se compone al menos de dos partes. Mejor dicho, un sketch necesita de dos bloques para su compilación: (cidstar, 2016)

De acuerdo con la página web de Arduino 2016 se detalla.

```
void setup( )
```

```
{ sentencias; }
```

```
void loop( )
```

```
{ sentencias; }
```

En donde setup() es la parte encargada de recoger la configuración y loop() es la que contienen el programa que se ejecutara cíclicamente (de ahí el termino loop -bucle-). Ambas funciones son necesarias para que el programa trabaje.

La función de configuración debe contener la declaración de las variables. Es la primera función a ejecutar en el programa, se ejecuta solo una vez, y se utiliza para configurar o inicializar pinMode (modo de trabajo de las E/S), configuración de la comunicación en serie y otras.

La función bucle (loop) contiene el código que se ejecutara continuamente (lectura de entradas, activación de salidas, etc) Esta función es el núcleo de todos los programas de Arduino y la que realiza la mayor parte del trabajo. (Arduino, 2016)

2.4.2 Estructura setup()

Según la pagina web de Arduino 2016 describe lo siguiente:

La estructura setup() se invoca una sola vez cuando el programa empieza. Se utiliza para inicializar los modos de trabajo de los pins, o el puerto serie. Debe ser incluido en un programa aunque no haya declaración que ejecutar. Así mismo se puede utilizar para establecer el estado inicial de las salidas de la placa.

void setup()

```
{
  pinMode(pin, OUTPUT);           // configura el 'pin' como salida
} . (Arduino, 2016)
```

2.4.3 Estructura loop()

Después de llamar a setup(), la estructura loop() hace precisamente lo que sugiere su nombre, se ejecuta de forma cíclica, lo que posibilita que el programa este respondiendo continuamente ante los eventos que se produzcan en la placa. (Arduino, 2016)

void loop()

```
{
  digitalWrite(pin, HIGH);      // pone en uno (on, 5v) el 'pin'
  delay(1000);                  // espera un segundo (1000 ms)
  digitalWrite(pin, LOW);       // pone en cero (off, 0v.) el 'pin'
  delay(1000);                  //
}
```

2.4.4 Funciones

Desde el punto de vista de Omar 2015 se define lo siguiente (pag 05).

Una función es un bloque de código identificado por un nombre y que es ejecutado cuando la función es llamada. La declaración de una función incluye en primer lugar el tipo de datos que devuelve la función. Después del tipo de datos se especifica el nombre de la función.

Las funciones de usuario pueden ser escritas para realizar tareas repetitivas y para reducir el tamaño de un programa. Las funciones se declaran asociadas a un tipo de valor. Este valor será el que devolverá la función, por ejemplo 'int' se utilizará cuando la función devuelva un dato numérico de tipo entero. Si la función no devuelve ningún valor entonces se colocará delante la palabra "void", que significa "función vacía". Después de declarar el tipo de dato que devuelve la función se debe escribir el nombre de la función y entre paréntesis se escribirán, si es necesario, los parámetros que se deben pasar a la función para que se ejecute.

tipo nombreFunción(parámetros)

```
{
instrucciones;
}
```

La función siguiente devuelve un numero entero, delayVal() se utiliza para poner un valor de retraso en un programa que lee una variable analógica de un potenciómetro conectado a una entrada de Arduino.

Al principio se declara como una variable local, 'v' recoge el valor leído del potenciómetro que estará comprendido entre 0 y 1023, luego se divide el valor por 4 para ajustarlo a un margen comprendido entre 0 y 255, finalmente se devuelve el valor 'v' y se retornaría al programa principal. (Omar, 2015)

int delayVal()

```
{
int v;          // crea una variable temporal 'v'
v= analogRead(pot);    // lee el valor del potenciómetro
v /= 4;        // convierte 0-1023 a 0-255
return v;      // devuelve el valor final
}
```

2.4.5 Entradas/Salidas digitales

Como se detalla en el párrafo Omar 2015 lo siguiente (pag 10).

Función pinMode(pin, mode)

Función usada en la function setup() para configurar un pin dado para comportarse como INPUT o OUTPUT. Ej. pinMode(pin, OUTPUT); configura el pin número 'pin' como de salida. Los pines de Arduino funcionan por defecto como entradas, de forma que no necesitan declararse explícitamente como entradas empleando pinMode().

Función digitalRead(pin)

Lee el valor desde un pin digital específico. Devuelve un valor HIGH o LOW. El pin puede ser especificado con una variable o una constante (0-13). Ej. `v = digitalRead(Pin);`

Funcion digitalWrite(pin, value)

Introduce un nivel alto (HIGH) o bajo (LOW) en el pin digital especificado. De nuevo, el pin puede ser especificado con una variable o una constante 0-13. Ej. `digitalWrite(pin, HIGH);` . (Omar, 2015)

2.4.6 Entradas/Salidas analógicas

Empleando las palabras de Omar de 2015 se halló lo siguiente (pag 10).

Función analogRead(pin)

Lee el valor desde el pin analógico especificado con una resolución de 10 bits. Esta función solo funciona en los pines analógicos (0-5). El valor resultante es un entero de 0 a 1023. Los pines analógicos, a diferencia de los digitales no necesitan declararse previamente como INPUT o OUTPUT.

Función analogWrite(pin, value)

Escribe un valor pseudo-analógico usando modulación por ancho de pulso (PWM) en un pin de salida marcado como PWM. Esta función está activa para los pines 3, 5, 6, 9, 10, 11. Ej `analogWrite(pin, v);` // escribe 'v' en el 'pin' analógico. Puede especificarse un valor de 0 - 255. Un valor 0 genera 0 V en el pin especificado y 255 genera 5 V. Para valores de 0 a 255, el pin alterna rápidamente entre 0 V y 5 V, cuanto mayor sea el valor, más a menudo el pin se encuentra en HIGH (5 V). Por ejemplo, un valor de 64 será 0 V tres cuartas partes del tiempo y 5 V una cuarta parte. Un valor de 128 será 0 V la mitad del tiempo y 5 V la otra mitad. Un valor de 192 será 0 V una cuarta parte del tiempo y 5 V tres cuartas partes. (Omar, 2015)

2.4.7 Variables

Puede ser declarada al inicio del programa antes de `setup()`, localmente a una determinada función e incluso dentro de un bloque como pueda ser un bucle. El sitio en que la variable es declarada determina el ámbito de la misma. (Omar, 2015)

2.4.8 Declaración de variables

Según el sitio web de Arduino 2016 se encontró lo siguiente.

Todas las variables tienen que declararse antes de que puedan ser utilizadas. Para declarar una variable se comienza por definir su tipo como `int` (entero), `long` (largo), `float` (coma flotante), etc, asignándoles siempre un nombre, y, opcionalmente, un valor inicial.

Antes de ser utilizadas, todas las variables tienen que ser declaradas. La declaración de una variable define su tipo, y, opcionalmente, el establecimiento de un valor inicial (inicializar la variable). Las variables no tienen que ser inicializadas (asignado un valor) cuando se declaran, pero a menudo es útil.

```
int inputVariable1;

int inputVariable2 = 0; // ambos son correctos
```

Los programadores deben tener en cuenta el tamaño de los números que desee almacenar en la elección de los tipos de variables. Variables dadas la vuelta cuando el valor almacenado supera el espacio asignado para almacenarlo. (Arduino.c.c, 2016)

2.4.9 Utilización de una variable

Como señala la página web de Arduino 2016 lo siguiente.

Una variable puede ser declarada al inicio del programa antes de la parte de configuración `setup()`, a nivel local dentro de las funciones, y, a veces, dentro de un bloque, como para los bucles del tipo `if.. for..`, etc. En función del lugar de declaración de la variable así se determinará el ámbito de aplicación, o la capacidad de ciertas partes de un programa para hacer uso de ella.

Una variable global es aquella que puede ser vista y utilizada por cualquier función y estamento de un programa. Esta variable se declara al comienzo del programa, antes de `setup()`.

Una variable local es aquella que se define dentro de una función o como parte de un bucle. Solo es visible y solo puede utilizarse dentro de la función en la que se declaró.

Por lo tanto, es posible tener dos o más variables del mismo nombre en diferentes partes del mismo programa que pueden contener valores diferentes. La garantía de que solo una función tiene acceso a sus variables dentro del programa simplifica y reduce el potencial de errores de programación.

El siguiente ejemplo muestra como declarar a unos tipos diferentes de variables y la visibilidad de cada variable: (Arduino, 2016)

```
int value;    // 'value' es visible para cualquier función
void setup()
{
  // no es necesario configurar nada en este ejemplo
}
void loop()
{
  for (int i=0; i<20;)    // 'i' solo es visible dentro del bucle for
```

```

{
  i++
}/
float f;      // 'f' es visible solo dentro de loop()
}

```

2.4.10 Tipos de datos

De acuerdo con Valero, 2014 se menciona lo siguiente.

Los datos que guardamos en las variables pueden ser de diferentes tipos, vamos a listar algunos de ellos. Para una referencia completa de los tipos de variables en Arduino se puede consultar esta página web.

char, se utilizan para almacenar caracteres, ocupan un byte.

byte, pueden almacenar un número entre 0 y 255.

int, ocupan 2 bytes (16 bits), y por lo tanto almacenan número entre 2^{15} y $2^{15} - 1$, es decir, entre -32,768 y 32,767.

unsigned int, ocupa también 2 bytes, pero al no tener signo puede tomar valores entre 0 y $2^{16} - 1$, es decir entre 0 y 65,535.

long, ocupa 32 bits (4 bytes), desde -2,147,483,648 a 2,147,483,647.

float, son números decimales que ocupan 32 bits (4 bytes). Pueden tomar valores entre $-3.4028235 \times 10^{38}$ y $+3.4028235 \times 10^{38}$.

Siempre que elegimos un tipo de dato debemos escoger el que menos tamaño necesite y que cubra nuestras necesidades, ya que ocuparán espacio en la memoria de nuestra placa y podría darse el caso de que nuestro programa requiera más memoria de la disponible. (Valero, 2014)

char miCaracter='a'.

//declaramos una variable tipo char, llamada mi Carcater y la inicializamos con valor a

byte unNumero = 189.

// variable tipo byte inicializada a 189

int unEntero

//variable tipo int sin inicializar

unsigned int numeroPositivo = 2343.

//entero positivo inicializada

float numDecimal = 12.212.

//numero decimal inicializado a 12,212

2.4.11 Entre llaves {}

Las llaves sirven para definir el principio y el final de un bloque de instrucciones. Se utilizan para los bloques de programación `setup()`, `loop()`, `if..`, etc.

type funcion()

{ instrucciones; } (Coursehero, 2016)

2.4.12 Punto y Coma ;

El punto y coma “;” se utiliza para separar instrucciones en el lenguaje de programación de Arduino. También se utiliza para separar elementos en una instrucción de tipo “bucle for”. (Coursehero, 2016)

int x = 13; /* declara la variable 'x' como tipo entero de valor 13 */

2.4.13 Línea de comentario //

Una línea de comentario empieza con // y terminan con la siguiente línea de código. Al igual que los comentarios de bloque, los de línea son ignoradas por el programa y no ocupan espacio en la memoria. (Coursehero, 2016)

// esto es un comentario

2.4.14 Constantes

El lenguaje de programación de Arduino tiene unos valores predeterminados, que son llamados constantes. Se utilizan para hacer los programas más fáciles de leer. Las constantes se clasifican en grupos. (Garcia, 2016)

2.4.15 High/Low

Se definen los niveles de salida altos o bajos y se utilizan para la lectura o escritura digital para las patillas. ALTO como es nivel 1, ON, o 5 voltios, mientras que BAJO es nivel 0, OFF, o 0 voltios. (Gutiérrez, 2007)

```
digitalWrite(13, HIGH); // activa la salida 13 con un nivel alto.
```

2.4.16 Input/Output

Estas constantes son utilizadas para definir, al comienzo del programa, el modo de funcionamiento de los pines mediante la instrucción `pinMode` de tal manera que el pin puede ser una entrada INPUT o una salida OUTPUT. (Gutiérrez, 2007)

```
pinMode(13, OUTPUT); // designamos que el PIN 13 es una salida
```

2.5 COMUNICACIÓN SERIE

La comunicación serie es un “interfaz de comunicación de datos digitales que nos permite establecer transferencia de información entre varios dispositivos”. (Martínez, 2015)

De acuerdo con la página web del García, 2013 se indica lo siguiente.

La comunicación permite interactuar con el Arduino, recibiendo información y enviándosela la programación creada en el software lo cual se da órdenes, previamente programadas, para que ejecute sin ningún problema en la transmisión y el rango deseado.

```
void setup()
{
  Serial.begin(9600);
}
void loop()
{ Serial.available();
  { Serial.read();
  } } }
```

El 9600 indica el baud rate, o la cantidad de baudios que manejará el puerto serie. Se define baudio como una unidad de medida, usada en telecomunicaciones, que representa el número de símbolos por segundo en un medio de transmisión ya sea analógico o digital. (Antony García González, 2013)

Según Orozco, 2014 da a conocer lo siguiente.

Serial.begin(9600);

Dicha instrucción solo requiere un parámetro y es la tasa de baudios a la cual se estará llevando a cabo la comunicación, este valor varía

dependiendo del dispositivo aunque comúnmente es 9600 como es nuestro caso con la pc.

Serial.available()

La cual devuelve el número de bytes disponibles para ser leídos por el puerto serie, razón por la que la hemos usado dentro de un if para que cuando haya algún dato disponible guardarlo en la variable "input" mediante la función:

Serial.read()

Esta función simplemente lee los datos provenientes del puerto serie. (Orozco, 2014).

2.6 LED

Led (Light Emitting Diode) es un diodo compuesto por la superposición de varias capas de material semiconductor que emite luz en una o más longitudes de onda, cuando es polarizado correctamente. Al aplicarle una pequeña corriente eléctrica (15 – 20 mA). (Luis, 2012)

2.6.1 LEDs de alto brillo

Los leds de alto brillo también conocidos como los BHLED, proporcionan una alta gama de luminosidad que los led estándar, los leds de alta luminosidad tienen numerosas ventajas sobre los leds estándar que se utilizan previamente en indicadores y pantallas modernas.



Figura 5 Leds de alto brillo

Fuente: (Cree, 2014)

Ventajas

- Más brillantes
- Vida más larga
- Bajo costo

2.6.2 Vida útil del led normal y led de alto brillo

Led normales. El funcionamiento LED es de 50.000 h. La cantidad de horas hacen que “envejezca” y “se debilite”, por lo que se establece que la vida útil de este dispositivo llega está al 70% de su flujo lumínico. (Iluminacionalve, 2015)

Led de alto brillo. La vida útil de los LEDS de alta calidad es más de 80,000 horas la cual el 50% de los LEDS fallan por sobre voltaje, vibración y otros ambientes que afectan la vida de los LEDS. (Leon, 2015)

2.6.3 Color

Los HBLEDs pueden ser rojos, verdes, azules, ámbar, blanco y RGB, dependiendo a la composición del material del color emitido. (Leon, 2015)

Tabla 2

Características de los HBLEDs

Color emitido	rojo	azul	verde
Tamaño (mm)	3/5mm	3/5mm	3/5mm
Color de lente	transparente	transparente	transparente
Voltaje	1.8 ~ 2.2	3.2 ~ 3.8	3.2 ~ 3.8
Duración de vida	100.000HS	100.000HS	100.000HS
Disipación de energía	80mw	80mw	80mw
Corriente continua	30mA	30mA	30mA
Temperatura	-25°C~+85°C	-25°C~+85°C	-25°C~+85°C
Rango de temperatura	-30°C~+100°C	-30°C~+100°C	-30°C~+100°C

Fuente: (Hyperbrillo, 2015)

Para un detalle pormenorizado de las características de los HBLEDs, favor dirigirse al Anexo C.

2.7 DECODIFICADORES

Según la especificación de Meteo 2011, detalla lo siguiente:

Los decodificadores son compuertas lógicas que transforman un código de tipo binario en código decimal. Consiste en activar una sola salida dependiendo el estado lógico en la cual se encuentra las entradas.

Un decodificador posee n líneas de entrada para gestionar n bits y en una de las 2^n líneas de salida indica la presencia de una o más combinaciones de n bits. (Meteo, 2011)

2.7.1 Decodificador 74LS48

El escrito de los decodificadores 2013 detalla.

Este decodificador sirve para mostrar salidas decimales a entradas binarias. Las entradas pueden estar dadas por cualquier dispositivo que tenga 4 salidas digitales como un puerto de un PIC, un micro, un Arduino, o utilizando switches para conmutar los unos y ceros como en el ejemplo de circuito propuesto.

El circuito integrado 7448 o subfamilia (74LS48, 74F48, 74S48, 74HCT48,..) es un circuito integrado que convierte el código binario de entrada en formato BCD a niveles lógicos que permiten activar un display de 7 segmentos de cátodo común en donde la posición de cada barra forma el número decodificado. (Decodificador, 2013)



Figura 6 Decodificador de BCD 7 segmentos

Fuente: (Mecacronica, 2014)

2.8 DISPLAYS

Un diodo emisor de luz es capaz de indicar un estado prendido/apagado por lo cual ha desarrollado unidades de indicación, también denominadas displays, frecuentemente muy variadas. Los más conocidos son los indicadores de display, indicadores LCD entre otro.

En el caso de los indicadores de los displays se juntan diversos diodos emisores de luz y son encendidos individualmente para conformar una letra o un número deseado. Por otro lado los LCD se utilizan el alineamiento de

los cristales líquidos en un campo eléctrico, se puede producir no solo letras y números sino también cualquier otro tipo de símbolo.

2.8.1 Tipos de Displays

Representan la generación de símbolos con los tres sistemas más importantes.

Display de 7 segmentos, los diez números son representados perfectamente por medio del sistema de 7 LEDs con la ayuda de lentes superpuestos que producen áreas iluminadas angostas de forma rectangular.

Display alfanumérico es considerablemente más versátil, se han ajustado 16 LEDs en una unidad de tal forma que se puede indicar todos los números, letras en mayúsculas y minúsculas en forma legible.

Display de nariz de punto los diodos están ordenados en filas y columnas. La mayoría de estos tipos de indicadores tiene siete LEDs por cada columna y cuatro o cinco LEDs por fila, de forma de que la unidad está constituido de 28 o 35 diodos emisores de luz.



Figura 7 Sistemas de indicadores de Display

Fuente: (Krylov, 2003)

2.8.2 Display de 7 segmentos

El display es un conjunto de 7 leds conectados en un punto común en la salida, ya sea ánodo común o cátodo común. Cada segmento puede encender o apagar individualmente, está compuesto de pequeñas líneas y se le denomina a, b, c, d, e, f y g se encuentra ensamblados de forma que se permita activar cada segmento separado así formando cualquier dígito numérico Anexo D.

Display cátodo común

El display cátodo común maneja una configuración de diodos LED donde el los cátodos están en serie y se manda el común del display a tierra prendiendo cada led del display con VCC en el ánodo.

Display Ánodo común

El display ánodo común es un conjunto de leds conectados los ánodos en serie y el común del display se alimenta al positivo para el encendido de cada uno de los leds del display.

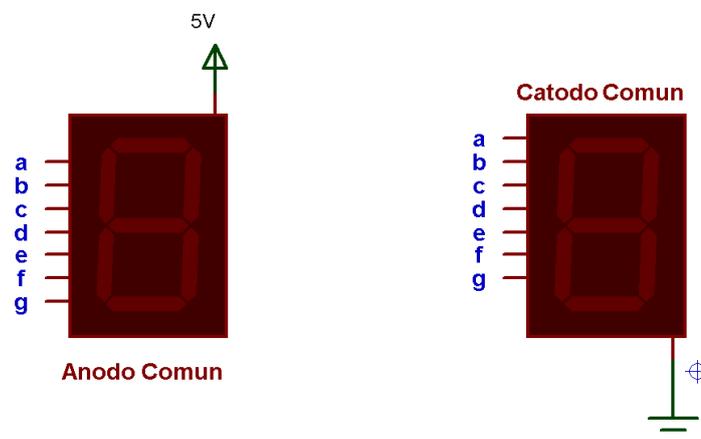


Figura 8 Display de 7 segmentos

Fuente: (Proteus8.1, 2016)

Tabla de verdad para el indicador de 7 segmento

Nº	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

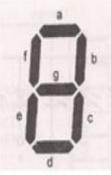



Figura 9 Tabla de la verdad del display de 7 segmentos

Fuente: (Tajani, 2013)

El control de los indicadores de LEDs se realiza hoy en día solamente con elementos integrados que contienen un gran número de interruptores electrónicos los cuales pueden ser conmutados por convertidores de código

2.9 MICROCONTROLADOR

Según la pagina web de cómputo integrado, 2012 describe.

Un microcontrolador es un circuito integrado, en cuyo interior posee una arquitectura de una computadora, esto es CPU, memoria RAM, EEPROM y circuitos de entrada y salida.

Un microcontrolador de fábrica, no realiza tarea alguna, este debe ser programado para que realice desde un simple parpadeo de un led hasta un sofisticado control de un robot. (Integrado, 2012)

2.9.1 El microcontrolador PIC16F877A

El escrito del Sellers, 2015 puntualiza lo siguiente.

El PIC16F877 es un microcontrolador con memoria de programa tipo FLASH, lo que representa gran facilidad en el desarrollo de prototipos y en su aprendizaje ya que no se requiere borrarlo con luz ultravioleta como las versiones EPROM, sino que permite reprogramarlo nuevamente sin ser borrado con anterioridad.

Los pines de entrada/salida de este microcontrolador están organizados en cinco puertos, el puerto A con 6 líneas, el puerto B con 8 líneas, el puerto C con 8 líneas, el puerto D con 8 líneas y el puerto E con 3 líneas. Cada pin de esos puertos se puede configurar como entrada o como salida independiente programando un par de registros diseñados para tal

fin. En ese registro un bit en "0" configura el pin del puerto correspondiente como salida y un bit en "1" lo configura como entrada. (Sellers, 2015)

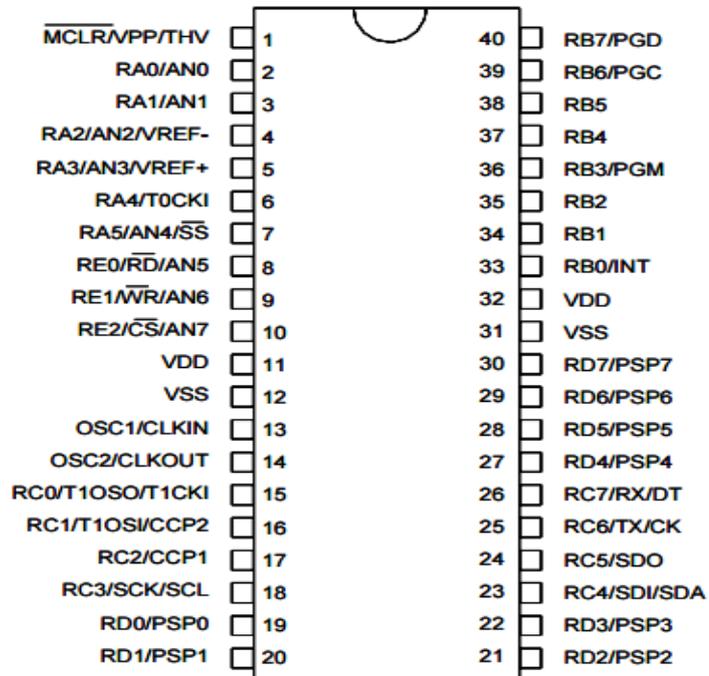


Figura 10 Distribución de pines del PICF877A

Fuente: (datasheet, 2013)

Tabla 3

Las características principales del PICs16f877A

CARACTERÍSTICAS DE PICs 16F877A	
Frecuencia máxima	20MHz
Memoria de programa flash palabra de 14 bits	8KB
Posiciones RAM de datos	368
Posiciones EEPROM de datos	256
Puertos entrada y salida	A, B, C, D, E
Numero de pines	40
Interrupciones	15
Timers	3

Continúa

Módulos CCP	2
Comunicaciones serie	MMSSP, USART
Comunicaciones paralelo	PSP
Modulo analógico a digital de 10 bit	8 canales de entrada
Juego de instrucciones	35 instrucciones
Longitud de la instrucción	14bits
Arquitectura	Harvard
CPU	RISC
Módulos comparador/PWM	2

Fuente: (Julian, 2015)

2.10 XBEE Y ZIGBEE

Desde el punto de vista de Ojeda 2015, expresa lo siguiente.

XBee

Los XBee's son pequeñísimos chip azules capaces de comunicarse de forma inalámbrica unos con otros. Pueden hacer cosas simples, como reemplazar un par de cables en una comunicación serial, lo cual es genial cuando deseas crear, por ejemplo, un vehículo radiocontrolado.

Los módulos XBee son soluciones integradas que brindan un medio inalámbrico para la interconexión y comunicación entre dispositivos. Estos módulos utilizan el protocolo de red llamado IEEE 802.15.4 para crear redes FAST POINT-TO-MULTIPOINT (punto a multipunto); o para redes PEER-TO-PEER (punto a punto). En términos simples, los XBee son módulos inalámbricos fáciles de usar Anexo E.

ZigBee

ZigBee es una tecnología de transferencia de datos en redes inalámbricas. Tiene bajo consumo de energía y está diseñado para sistemas multi-canal de control, sistemas de alarma y control de iluminación. También tiene otras diversas aplicaciones del hogar y la industria. ZigBee es más económico que el Wi-Fi mientras protocolo Bluetooth también consume más energía y tiene una mayor dispersión de los parámetros de las unidades convencionales compatibles. (Ojeda, 2015)

2.10.1 XBee-PRO RF Module RPSMA S1 2.4 GHz

De acuerdo con la pagina wed DigiKey, 2015 señala.

El módulo XBee-PRO S1 RF de 2,4 GHz es una solución compatible con IEEE 802.15.4, que satisfaga las necesidades únicas de bajo costo, las redes de sensores inalámbricas de bajo consumo. El módulo es fácil de usar, requiere un mínimo de energía y proporciona una entrega fiable de datos críticos entre dispositivos.

El módulo opera dentro de la banda ISM (Industrial, Scientific and Medical) utilizando una frecuencia de 2.4 GHz.

Los requerimientos mínimos de conexión consisten en proveer alimentación mediante dos terminales (VCC y GND) y dos de datos (Vin y Vout).

Los módulos XBee fueron diseñados para ser montados en un zócalo, no requiriendo ningún soldado. Para ello disponen dos hileras de 10 pines separadas entre ellas por 22 mm La separación entre pines es de 2mm. (DigiKey, 2015)



Figura 11 XBee-pro S1

Fuente: (DigiKey, 2015)

2.10.2 Tipos de antenas

Desde el punto de vista de Ojeda 2015, señala lo siguiente

Chip Antenna – Básicamente es un pequeño chip que actúa como antena. Rápido, sencillo y barato.

Wire Antenna (Whip Antenna)– Es un pequeño cable que sobresale.

u.FL Antenna – Un conector pequeño para conectar tu propia antena. Esto es perfecto si tienes tu equipo en una caja y deseas la antena afuera de ésta.

RPSMA Antenna – Un conector más grande para conectar tu propia antena. Nuevamente, esto es perfecto si tienes tu equipo en una caja y deseas la antena afuera de ésta. (Ojeda, 2015)

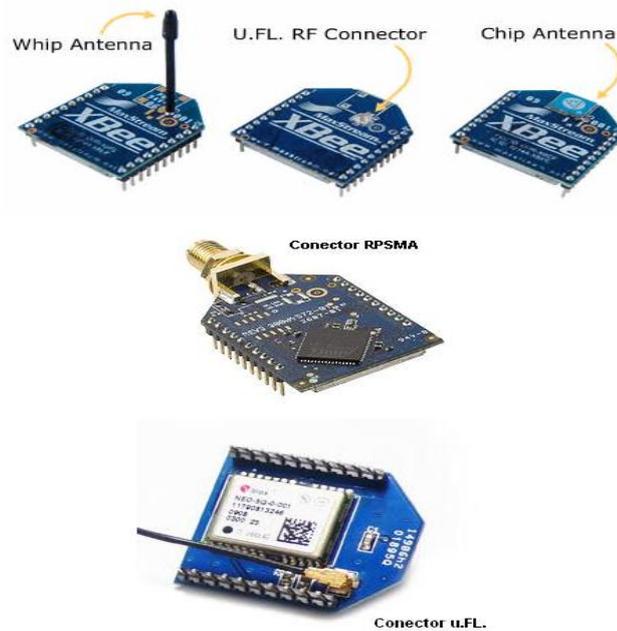


Figura 12 Tipos de Antenas de los Xbee

Fuente: (Ojeda, 2015)

2.10.3 Característica del XBee pro

Las características más importantes en este estándar son su flexibilidad de red, bajos costos, bajo consumo de energía; este estándar se puede utilizar para muchas aplicaciones en el hogar que requieren una tasa baja en la transmisión de datos. (Catarina, 2015)

Tabla 4

Características del XBee pro S1

XBEE PRO S1 RP-SMA	
Categoría	RF/IF Y RFID
Protocolo	Zigbee
Familia/estándar RF	802.15.4
Frecuencia	2.4 GHz
Velocidad de datos	250 kbps
Salida de Potencia	100mW
Temperatura de operación	-40 a 85°C

Continúa

Voltaje de la fuente	2.8 V a 3.4 V
Corriente recepción	55mA
Corriente transmisión	215mA

Fuente: (digikey, 2015)

2.11 ANTENA RP-SMA 2.4GHz

La antena RP-SMA es util para la TX/RX a 2.4 GHz y es ideal para los modulos de alta potencias como son los XBee-pro ZigBee, en su interior esta constituido de un cable de cobre de 10cm de largo. (Farnel, 2014)

La Antena posee una ganancia de 2.2dBi polarizada inversa con un conector SMA RF. Perfecto para la creación de prototipos con nuestros circuitos integrados de RF como los modulos XBee Pro S1. Impedancia de 50 ohmios. 4 "de largo. (Sparkfun, 2015)



Figura 13 Antena RP-SMA 2.4 GHz

Fuente: (Sparkfun, 2015)

Tabla 5

Características de la Antena RP-SMA

Especificación	
Tipo de antena	Colineal
Polarización	Vertical
Rango de Frecuencia	2.4 - 2.5GHz

CONTINUA

Ganancia	2.2dBi
Conector	RP SMA macho
Impedancia	50Ω
Dimensión	10mm x 108mm
Radiación	Omnidireccional

Fuente: (Microchip, 2015)

2.12 TIP 127

Desde el punto de vista de Alejandro, 2014 indica lo siguiente.

El tip 27 es un transistor tipo Darlington de media potencia PNP, soporta 100V /5A/65W con disipador.

Cuenta (precisamente por ser Darlington), con una alta ganancia de corriente (1000 mínimo) (por multiplicación de beta de los transistores internos.) , lo cual lo hace apto para manejar gran corriente de colector con baja corriente de base. Para que se entienda en números redondos, este transistor polarizado convenientemente con solo 1mA de base puedes manejar corrientes de 1A en colector. Lógicamente la IB debe ser mayor que 1mA en base en la práctica, dado que la corriente de corte .Ibeo es de 2mA.

No es un transistor común de uso general, su aplicación es bien específica. (Sería como 2 transistores en cascada dentro un mismo encapsulado). (Alejandro, 2014)

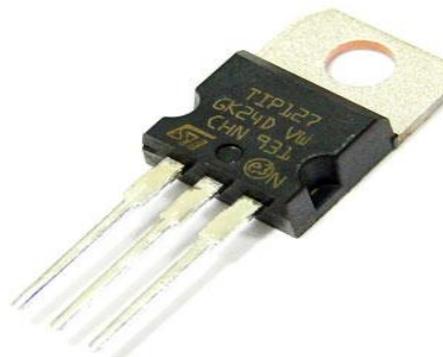


Figura 14 Transistor Darlington tip127

Fuente: (Alejandro, 2014)

2.13 TRANSISTOR NPN 2N3904

La información encontrada de la wed Electrónica, 2013 se escribe o siguiente.

Los transistores NPN consisten en una capa de material semiconductor dopado P (la “base”) entre dos capas de material dopado N. Una pequeña corriente ingresando a la base en configuración emisor-común es amplificada en la salida del colector.

La mayoría de los transistores bipolares usados hoy en día son NPN, debido a que la movilidad del electrón es mayor que la movilidad de los “huecos” en los semiconductores, permitiendo mayores corrientes y velocidades de operación. (Electronica, 2013)

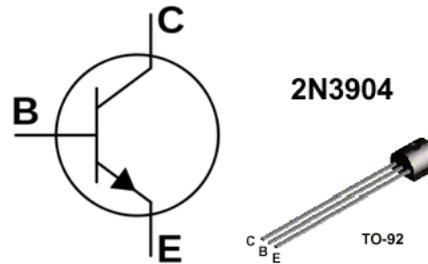


Figura 15 Transistor 2N3904

Fuente: (Electronica, 2013)

2.14 FUENTE DE ALIMENTACIÓN

La fuente de alimentación, es un dispositivo que se encarga de transformar la corriente alterna que se recibe en los domicilios, en corriente continua o directa, que es la que utilizan los dispositivos electrónicos tales como televisores, computadoras, leds ,etc. (Concepto, 2015)

2.14.1 Fuente de alimentación lineal y conmutada

Según la presentación Carbajal, 2014 se describe.

Fuentes lineales: Aquellas que utilizan un transformador para disminuir el nivel de tensión; Circuito compuesto por componentes electrónicos que convierten el voltaje alterno en directo.

Fuentes conmutadas: Aquellas que utilizan circuitos basados en transistores y bobinas trabajando en conmutación para reducir la tensión. Es un dispositivo electrónico que transforma energía eléctrica mediante transistores en conmutación. (Carbajal, 2014)



Figura 16 Fuente Meanwell 120W 24V 5A

Fuente: (Industrial, 2016)

CAPÍTULO III

DESARROLLO DEL PROYECTO

3.1 PRELIMINARES

En este capítulo se describirá, de una forma técnica y detallada sobre el procedimiento de la elaboración del marcador electrónico por lo cual se inspecciona el estado actual del marcador electrónico para determinar los equipos que van a ser utilizados y los que deben adquirirse para el nuevo funcionamiento, de igual manera se realizara un circuito de potencia, para conectar los leds de alto brillo por cada segmento dentro del marcador electrónico. Para controlar el tiempo, periodo, puntos y faltas, se realizara un sistema de control inalámbrico para el manejo con los módulos XBee-PRO.

3.2 REQUERIMIENTOS MÍNIMOS

Para la implementación del proyecto técnico de Titulación cuyo tema es: **“REHABILITACIÓN DEL MARCADOR ELECTRÓNICO MEDIANTE LEDS DE ALTO BRILLO Y MANDO DE RADIOFRECUENCIA PARA LA UNIDAD DE GESTIÓN DE TECNOLOGÍAS”** fue necesario lo siguiente:

- Arduino Mega
- Pic 16f877A
- Dos módulos XBee-pro S1
- Leds de alto brillo 5mm
- Tip127
- Transistores 2N3904
- Pulsadores
- Decodificadores 74LS48
- Resistencias
- Cables UTP flexibles
- Conectores
- Fuente de alimentación de 24 VDC 5 A
- Tablero electrónico

Software utilizado

- Plataforma Arduino
- PIC C Compiler
- PICkit 2 v2.61
- Proteus 8.1
- XCTU

3.3 ARMADO DEL TABLERO ELECTRÓNICO

Se procede a revisar el estado en la que se encuentra el tablero, de igualmente sus componentes que constituye, para verificar sus partes defectuosas para proceder a realizar la nueva rehabilitación del marcador electrónico.



Figura 17 Tablero electrónico antiguo

Al destapar el marcador electrónico se encontró los materiales en mal estado, cada componente se fue separando para verificar el funcionamiento. Se procede a desarmar en conjunto del display de 7 segmentos para cambiar los leds normales por los leds de alto brillo de 5mm.



Figura 18 Marco de display 7 segmentos

Para la nueva colocacion de los leds se procedio a quitar los segmentos del antiguo tablero, de la misma forma a los cables sobrantes.



Figura 19 Segmentos desmontados

Los materiales en mal estado como: el control general, los decodificadores, conectores, etc, del antigua tablero electrónico se desmonto de la placa para poder verificar el funcionamiento.



Figura 20 Componentes e mal estado

Se procede a sacar el marco antiguo del tablero, la cual esta constituido los segmentos a, b, c, d, e, f y g en cada uno de los displays (tiempo, periodo local, visitante, faltas) dentro del marcador.

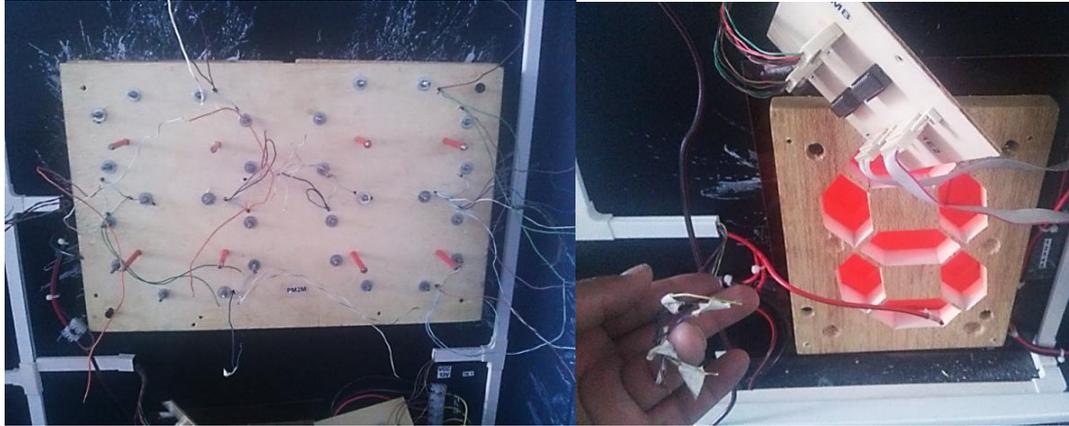


Figura 21 Desconexión del marco de display 7 segmentos

De la misma forma se fue separando las resistencias que se encontraba defectuosas de cada segmento desmontado anterior mente.



Figura 22 Juego de resistencias

Para los nuevos segmentos se cortó tiras de 10cm largos para los segmentos a, d, g y 44 tiras para los segmentos b, c, e, f. Una vez cortado se procede a limpiar las impurezas que contiene el cobre con lustre.

El circuito de los leds realizado en el software proteus es recomendado imprimir en papel fotográfico para el quemado en la placa de baquelita.

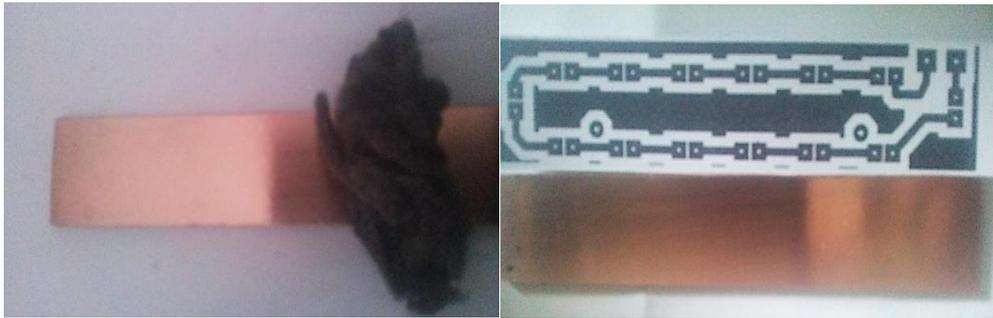


Figura 23 Limpieza del cobre

Con la ayuda de la plancha se procede al planchado de la pista sobre el cobre a una alta temperatura.



Figura 24 Planchado del circuito

Una vez planchado insertamos en un recipiente el circuito con agua fría para que saque el papel del cobre, con las gomas de los dedos se procede a retirar con cuidado el papel sobrante hasta que quede solo la tinta negra sobre el cobre.



Figura 25 Limpieza del papel sobrante

Se procede a colocar los segmentos en el marco con sus respectivas tiras hasta formar un display 7 segmentos cátodo común.



Figura 26 Display de 7 segmentos

Una vez colocado el marco dentro del tablero, las salidas de los segmentos se proceden a colocar las resistencias para limitar la corriente.



Figura 27 Colocación de las resistencias

3.4 DIAGRAMA DE LOS DECODIFICADORES 74LS48

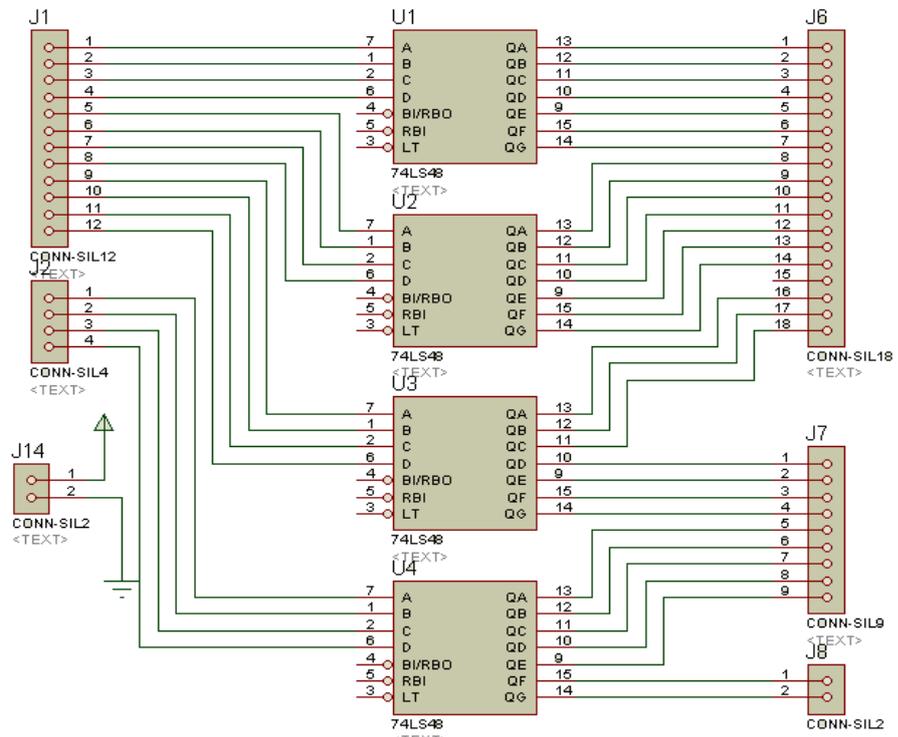


Figura 28 Decodificador 74ls48

Se verificó si los decodificadores 74ls48 se encontraba en perfecto estado para la rehabilitación del tablero y la reubicación de la misma.

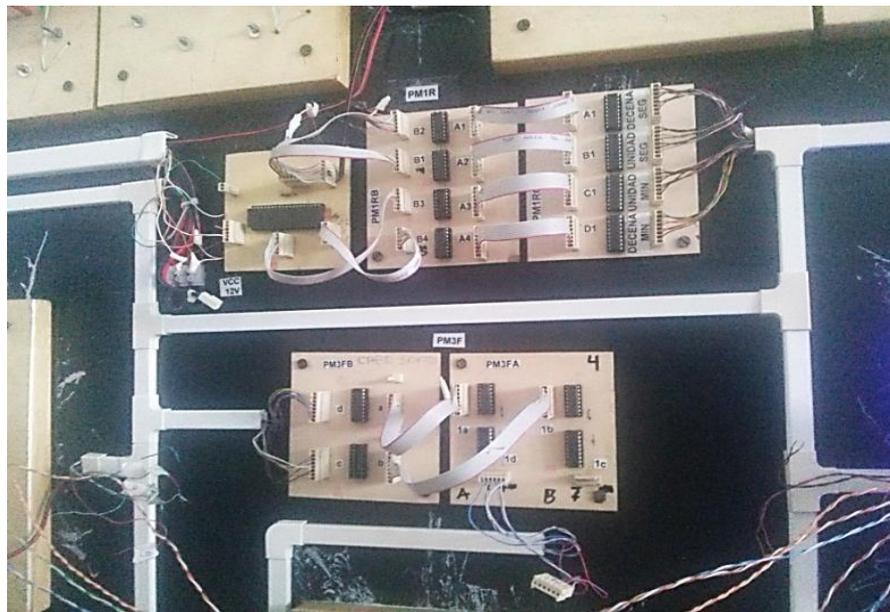


Figura 29 Decodificadores 74ls48

3.5 IMPLEMENTACIÓN DE POTENCIA

3.5.1 Diagrama de potencia Darlington

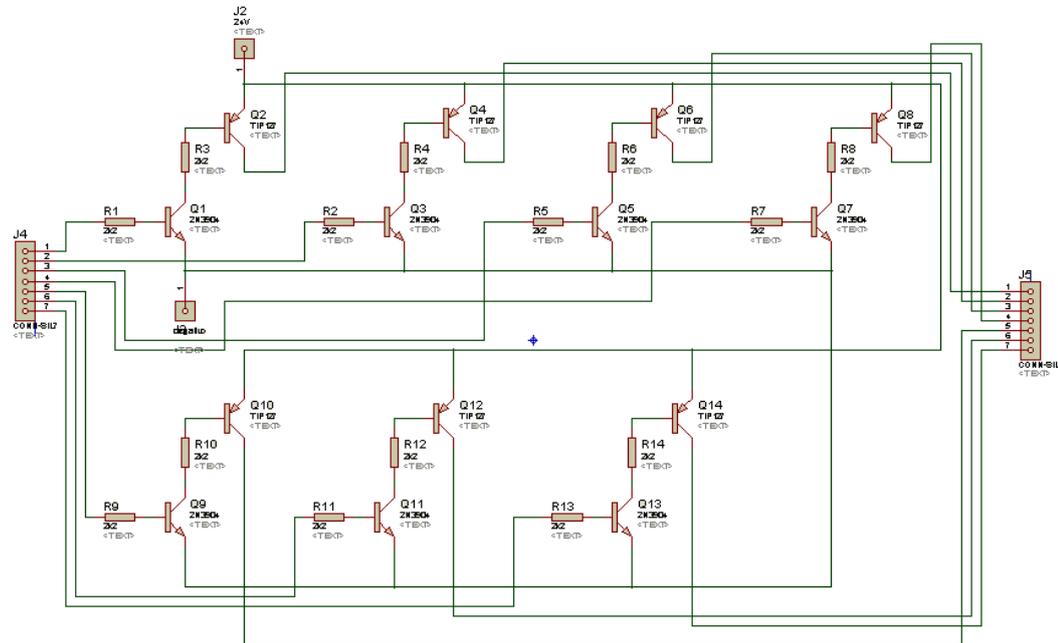


Figura 30 Circuito de potencia

Para la parte de potencia para los leds de alto brillo se utilizó un circuito Darlington que va conectado a la salida de las resistencias de los display de 7 segmentos, de los conectores en su posición original.

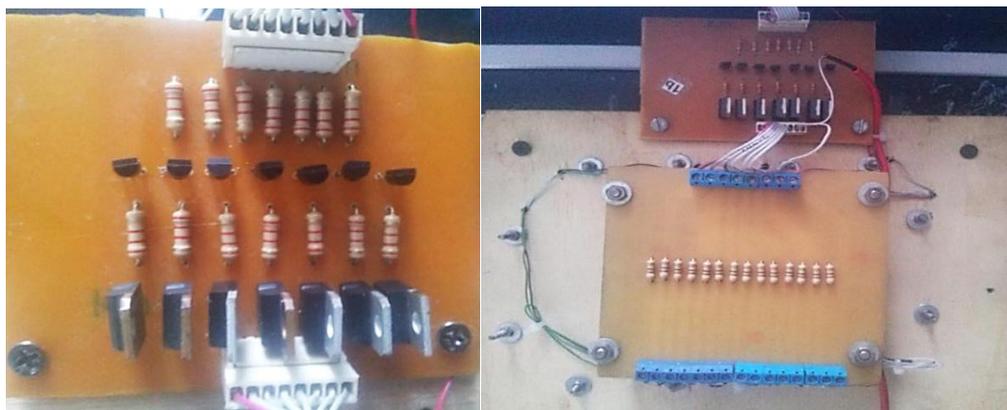


Figura 31 Circuito Darlington

La instalación del circuito Darlington con cada display de 7 segmentos fue instalado correctamente dentro del tablero electrónico asegurando su funcionamiento dentro de la misma.

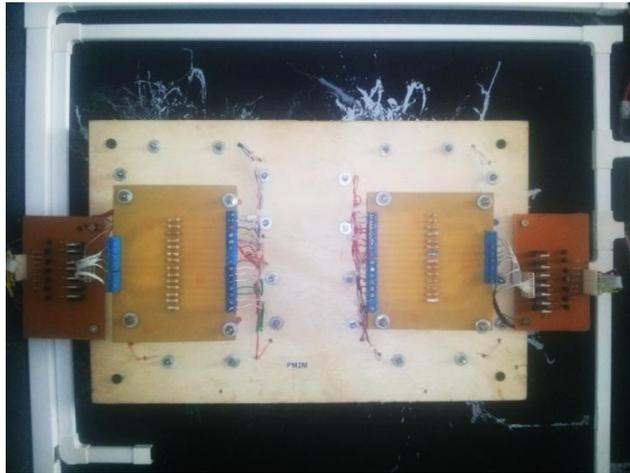


Figura 32 Instalación del circuito Darlington

Los decodificadores 74LS48 que se encontraban en perfecto estado se procedió a conectar a las entradas BCD al Arduino de la misma forma la salida a, b, c, d, e, f y g a las entradas del circuito Darlington.

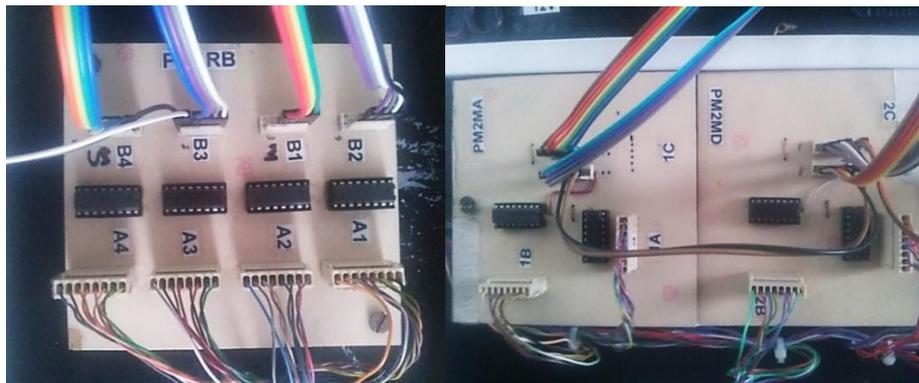


Figura 33 Decodificador rehabilitado

3.6 IMPLEMENTACIÓN DEL CONTROL INALÁMBRICO

Con los módulos XBee Pro S1 se realizó una prueba de funcionamiento y prueba de distancia de transición y receptor.



Figura 34 Prueba de funcionamiento del XBee Pro S1

En el módulo XBee Pro S1 se coloca una barra de madera de 10cm con la tuerca apretada para sujetar dentro del tablero electrónico.



Figura 35 Conexión del módulo XBee Pro S1

3.7 ARMADO DEL CONTROL INALÁMBRICO

3.7.1 Diagrama control de mando con el microcontrolador PIC16F877A

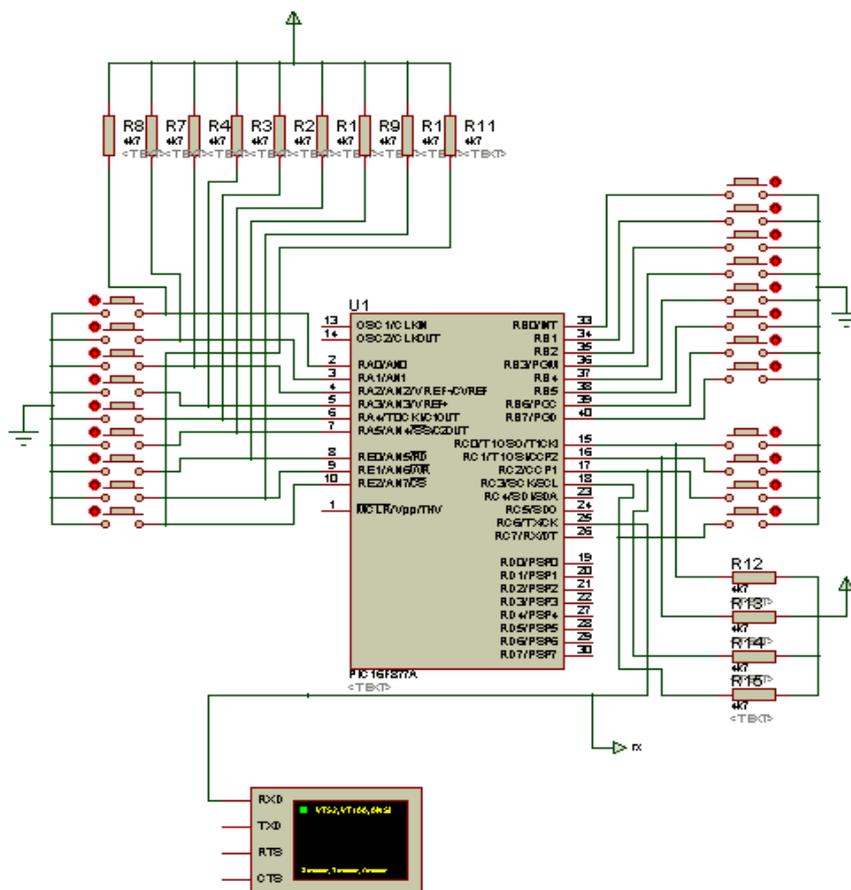


Figura 36 Diagrama de control inalámbrico

Para realizar el control de mando inalámbrico para el tablero electrónico, se procede al planchado y quemado con el ácido férrico en la placa de baquelita

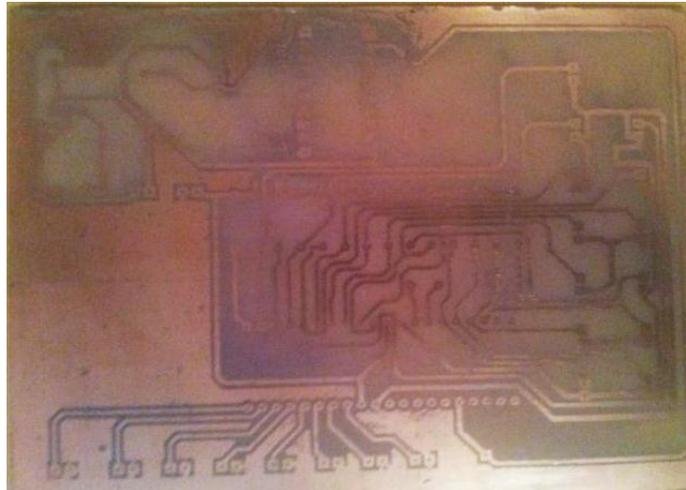


Figura 37 Circuito impreso del control de mando

Se procede a realizar perforaciones en el circuito, para soldar los componentes (PIC16F877A, resistencias, condensadores, reguladores, conectores, pulsadores, etc.) correspondientes dentro de la placa.

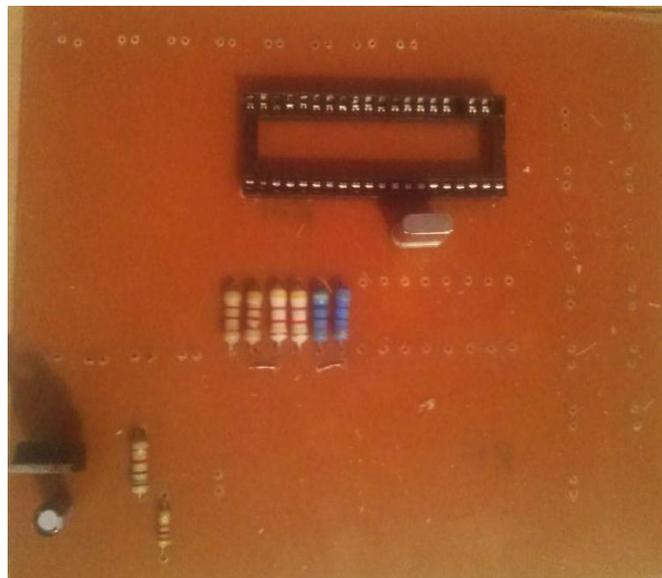


Figura 38 Suelda de componentes en la placa

3.8 INSTALACIÓN DE LA FUENTE DE 24V 5A

Para la instalación de la fuente de 24V se utilizó cable eléctrico para su mejor desempeño y con una protección de fusible de 5A.

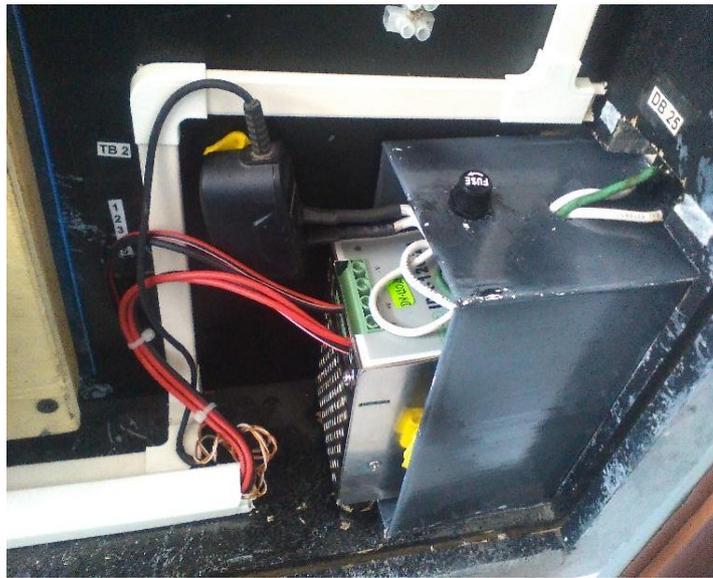


Figura 39 Fuente de alimentación de 24V

3.9 FINALIZACIÓN DEL MARCADOR ELECTRÓNICO

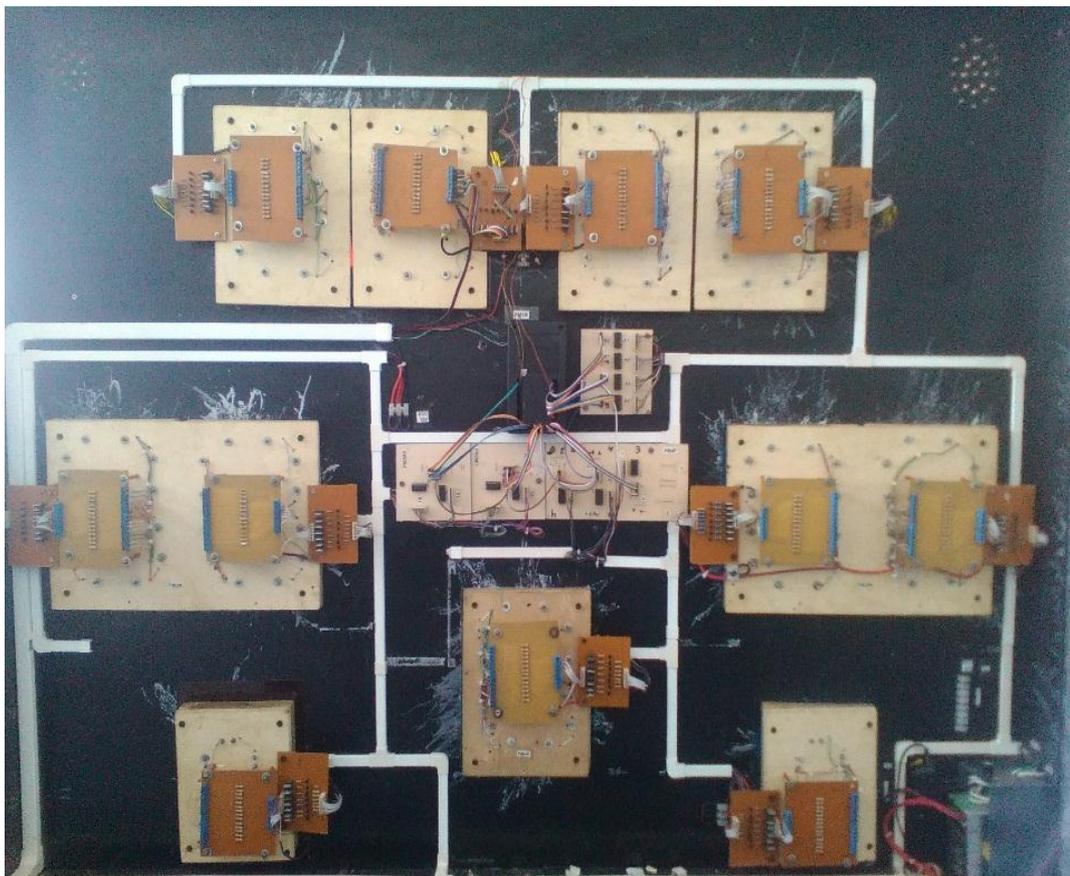


Figura 40 Parte trasera del tablero electrónico

Parte delantera finalizado



Figura 41 Parte delantera del tablero electrónico

3.10 DIAGRAMA DEL MARCADOR ELECTRÓNICO EN PROTEUS 8.1

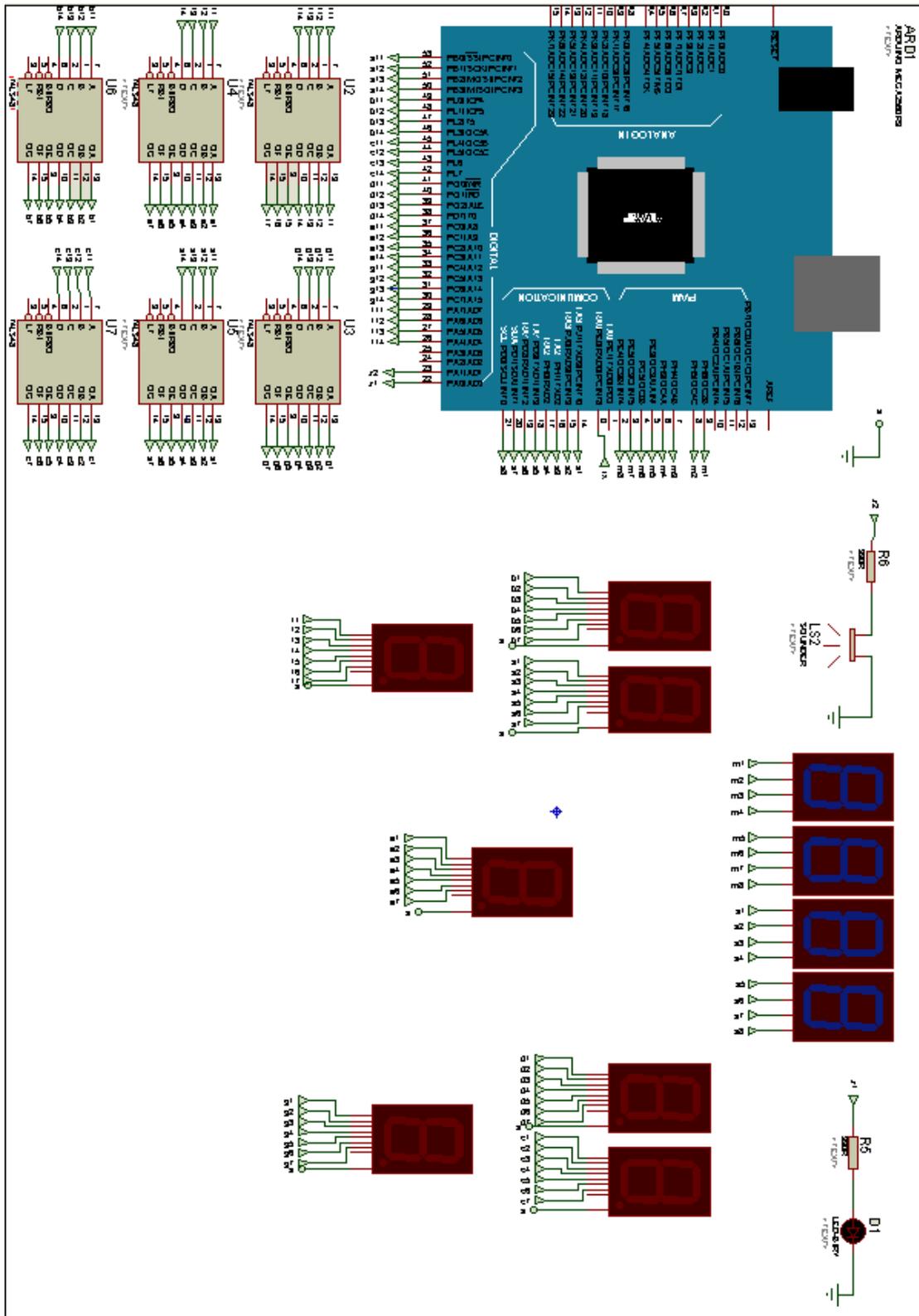


Figura 42 Diagrama del tablero electrónico

3.11 CONFIGURACIÓN DEL ARDUINO MEGA 2560

Para programar en el software y conectar la tarjeta Arduino mega 2560 se realiza los siguientes pasos:

1) Descargar e software Arduino 1.6.9

El software Arduino es un plataforma libre se puede descargar desde la página oficial Arduino.cc la última versión.

2) Conexión del Arduino Mega 2560

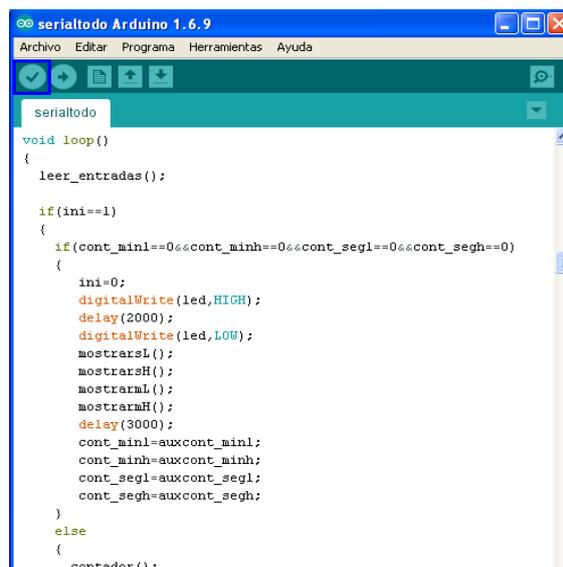
Una vez instalado el software Arduino en la computadora, se conecta el cable USB a la entrada del CPU con la tarjeta del Arduino Mega 2560, conjuntamente se alimenta con la fuente del USB a la tarjeta Arduino mega.

3) Instalación de los controladores

Al conectar el cable de la tarjeta USB, automáticamente comienza a instalar los driver necesario, para asignar un puerto COM en el Software Arduino Mega 2560.

4) Crear y guardar el programa

Al realizar la programación, se crea varios datos de salida como leds de alto brillo, zumbador, módulo XBee, decodificadores. Una vez creado el programa click en verificar para saber si no existe ningún error alguno, al terminar la compilación se guarda automáticamente en la carpeta mis documentos-Arduino



```

serialtodo Arduino 1.6.9
Archivo  Editar  Programa  Herramientas  Ayuda

serialtodo
void loop()
{
  leer_entradas();

  if(ini==1)
  {
    if(cont_minl==0&&cont_minh==0&&cont_segl==0&&cont_segh==0)
    {
      ini=0;
      digitalWrite(led,HIGH);
      delay(2000);
      digitalWrite(led,LOW);
      mostrarsL();
      mostrarsH();
      mostrarmL();
      mostrarmH();
      delay(3000);
      cont_minl=auxcont_minl;
      cont_minh=auxcont_minh;
      cont_segl=auxcont_segl;
      cont_segh=auxcont_segh;
    }
    else
    {
      contador();
    }
  }
}

```

Figura 43 Verificación del programa

5) Selección de la tarjeta Arduino Mega 2560

Para seleccionar el puerto de la tarjeta Arduino Mega 2560, se dirige a la barra de menu principal, herramientas, puerto COM del Arduino click para seleccionar

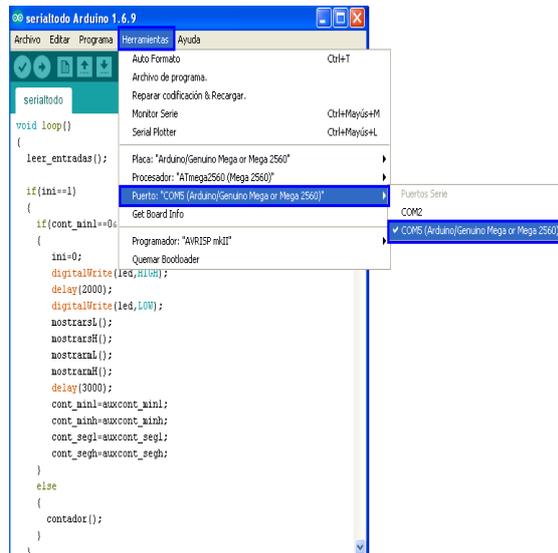


Figura 44 Selección del puerto serial

6) Cargar el programa

Para cargar el software en la tarjeta Arduino Mega 2560 click en la flecha subir, espere unos segundos hasta que compile el programa en la tarjeta se observa la barra de estado de compilación.

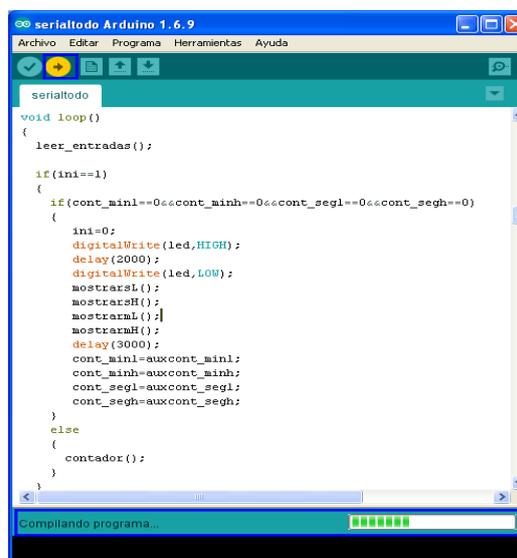


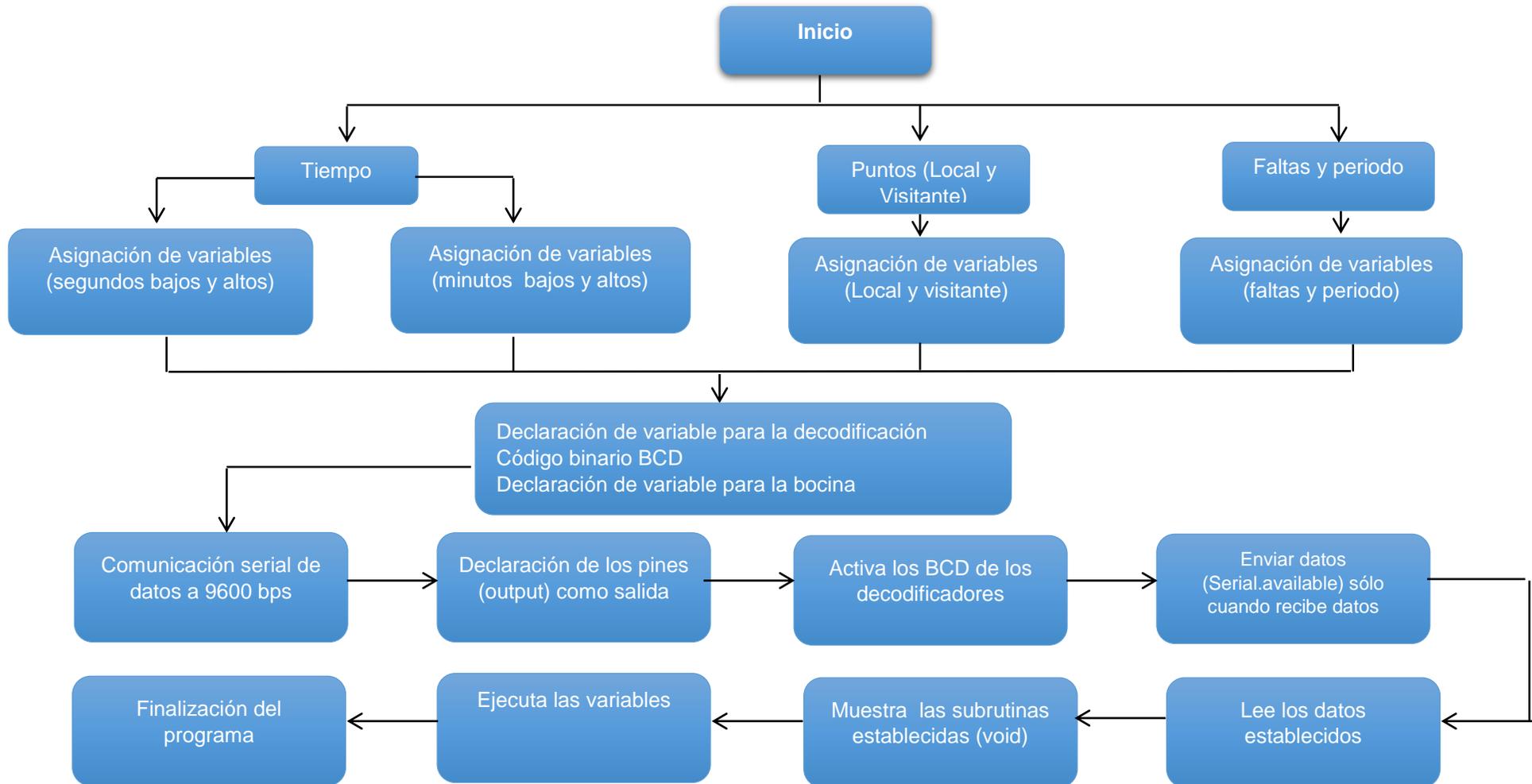
Figura 45 Cargar el programa en la tarjeta Arduino

3.12 CÓDIGO DE PROGRAMACIÓN PARA EL TABLERO ELECTRÓNICO CON ARDUINO MEGA 2560, XBEE Y PIC 16F877A.

3.12.1 Funcionamiento del tablero electrónico

La programación realizada en el software Arduino está basado en un tablero electrónico, lo cual cumple la función de mostrar el tiempo de juego, periodo puntos y faltas de cada partido, está realizado con display cátodo común de 7 segmentos 24x32, decodificadores 74ls48, módulos XBee receptor, para el control inalámbrico se utilizó el microcontrolador Pic 16F877A, módulo XBee transmisión, pulsadores.

3.12.2 Código de programación de tiempo, puntos, faltas y periodo



3.13 CÓDIGO DE PROGRAMACIÓN DEL MICROCONTROLADOR PIC16F877A

Para programar en el software PIC C compiler se debe tener en cuenta algunos paso.

1) Descargar el programa PIC C compiler

Descargar el programa del internet, es una plataforma es una plataforma libre de descarga.

2) Instalación del programa PIC C compiler

Una vez descargado el programa, dirigir a la carpeta de descarga y buscar el programa doble click se procede a la instalación del software click en siguiente asta llegar a la finalización del software.

3) Crear y guardar el programa

Al abrir el programa se comienza a programar, una vez realizado la programación se procede a guardar el archivo con la extensión ".hex" para luego quemar en el microcontrolador con el software PICkit.

4) Cargar el programa en el PIC16F877A

Al guardar el programa creado en el software PIC C compiler, se procede al quemado en el microcontrolador 16F877A, con la ayuda del quemador PICkit el archivo .hex

3.13.1 Código de control inalámbrico

```
// La librería del Pic
#include <16f877A.h>
// activación de los fusibles, perro guardián
#fuses HS, NOWDT,PUT,NOBROWNOUT
// El tiempo del oscilador
#use delay (clock=4M)
// Comunicación serial tx y rx
#use rs232(baud=9600,xmit=PIN_C6,rcv=PIN_C7)
void main()
{
int a;
```

```
port_b_pullups(TRUE);
while(true)
{
// habilitación de los puertos
a=input_b();
if (a==254)
printf("a");
if (a==253)
printf("b");
if (a==251)
printf("c");
if (a==247)
printf("d");
if (a==239)
printf("e");
if (a==223)
printf("f");
if (a==191)
printf("g");
if (a==127)
printf("h");
if(input(pin_a0)==0)
printf("i");
if(input(pin_a1)==0)
printf("j");
if(input(pin_a2)==0)
printf("k");
if(input(pin_a3)==0)
printf("l");
if(input(pin_a4)==0)
printf("m");
```

```
if(input(pin_a5)==0)
printf("n");
if(input(pin_e0)==0)
printf("o");
if(input(pin_e1)==0)
printf("p");
if(input(pin_e2)==0)
printf("q");
if(input(pin_c0)==0)
printf("r");
if(input(pin_c1)==0)
printf("s");
if(input(pin_c2)==0)
printf("t");
if(input(pin_c3)==0)
printf("v");
delay_ms(1000);
}
}
```

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- De la inspección realizada al marcador electrónico, se determinó que el 85% de los elementos se encontraban defectuosos (Atmega, resistencias, ULN2803, conectores, cables, leds, puentes) y el 15 % de elementos podían ser reutilizados (decodificadores, tablero de madera, conectores de 8 pines) por lo que se procedió a adquirir transistores, placa Arduino, PIC 16F877A, leds de alto brillo, resistencias, xbee, antenas, etc. a fin de implementar el nuevo marcador electrónico.
- Para el funcionamiento de los Display de 7 segmentos se construyó un circuito de potencia, para conectar los leds de alto brillo de cada segmento que se encuentra conectado en serie la cual consume 0.04mA por cada segmento en un total de 0.28 mA el conjunto del display.
- Para controlar el marcador electrónico se implementó un sistema de control inalámbrico, para entrelazar la comunicación serial entre el pic16F877A y el arduino mega 2560 la velocidad de transmisor y receptor debe ser de 9600 baud, con dos módulos XBee Pro S1 para el manejo inalámbrico entre el transmisor y el receptor con una antena de pato adicional, para una mejor recepción y transición de datos fijos.

4.2 RECOMENDACIONES

- Al momento de realizar la inspección dentro del marcador electrónico verificar que los equipos se encuentren en perfecto estado para poder ser reutilizados como algunos decodificadores, ULN2803, tener en cuenta los circuitos que no están funcionando retirar cuidadosamente.
- Tener en cuenta el voltaje para el encendido de los Display de 7 segmentos, estos display consume una alta gama de corriente por lo cual se debe construir un circuito de potencia, para conectar los leds de alto brillo.
- Revisar el sistema de control inalámbrico que los voltajes estén independiente para el módulo XBee Pro S1 del transmisor los 3.3V para el manejo inalámbrico al receptor del marcador electrónico con una antena de pato adicional.

GLOSARIO DE TÉRMINOS

BHLED	Diodo emisor de luz de alto brillo
EEPROM	Memoria de sólo lectura programable y borrable eléctricamente
LCD	Pantalla de cristal líquido
LED	Diodo emisor de luz
IEEE	Instituto de ingenieros eléctricos y electrónicos
IP	Protocolo de internet
ISM	Industrial, Científica y Médica
PIC	Código de identificación del participante
PWM	Modulación por ancho de pulso
RAM	Memoria de acceso aleatorio
RF	Radio Frecuencia
RFID	Identificación por radiofrecuencia
RGB	Rojo, Verde y Azul
RX	Recepción
TX	Transmisión
UART	Transmisión-Receptor Asíncrono Universal
USB	Bus Universal en Serie

REFERENCIAS BIBLIOGRÁFICAS

- Alejandro. (2014). <http://electronica.yoreparo.com>. Obtenido de <http://electronica.yoreparo.com>:
<http://electronica.yoreparo.com/datasheets/tip127-t1517383.html>
- Antony García González, K. N. (27 de enero de 2013). <http://panamahitek.com>. Obtenido de <http://panamahitek.com>:
<http://panamahitek.com/comunicacion-serial-con-arduino/>
- Arduino. (2016). <http://playground.arduino.cc>. Obtenido de <http://playground.arduino.cc>:
<http://playground.arduino.cc/ArduinoNotebookTraduccion/Structure>
- Arduino.c.c. (2016). <https://www.arduino.cc>. Obtenido de <https://www.arduino.cc>:
<https://www.arduino.cc/en/Reference/VariableDeclaration>
- Arduinodhtics. (2015). <http://arduinodhtics.weebly.com>. Obtenido de <http://arduinodhtics.weebly.com>:
<http://arduinodhtics.weebly.com/tipos-de-arduino.html>
- Baloncesto. (2014). <http://www.fbrm.es>. Obtenido de <http://www.fbrm.es>:
<http://www.fbrm.es/reglamento/>
- Carbajal, E. R. (28 de julio de 2014). <https://prezi.com>. Obtenido de <https://prezi.com>: <https://prezi.com/bcoczedtsyfn/diferencia-entre-fuentes-linales-y-conmutadas/>
- Carrasco, R. (febrero de 2012). <http://redi.uta.edu.ec>. Obtenido de <http://redi.uta.edu.ec>:
http://redi.uta.edu.ec/bitstream/123456789/2334/1/Tesis_t682ec.pdf
- Catarina. (2015). <http://catarina.udlap.mx>. Obtenido de <http://catarina.udlap.mx>:
http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/archundia_p_fm/capitulo4.pdf
- cidstar. (04 de junio de 2016). <http://rduinostar.com>. Obtenido de <http://rduinostar.com>:
<http://rduinostar.com/documentacion/programacion-arduino/estructura-de-un-sketch/>
- Concepto. (2015). <http://concepto.de>. Obtenido de <http://concepto.de>:
<http://concepto.de/fuente-de-alimentacion/>
- Coursehero. (2016). <https://www.coursehero.com>. Obtenido de <https://www.coursehero.com>:
<https://www.coursehero.com/file/p2n6uns/Se-utilizan-para-los-bloques-de-programaci%C3%B3n-setup-loop-if-etc-type-funcion/>

- Cree. (14 de Febrero de 2014). <http://www.cree.com>. Obtenido de <http://www.cree.com: http://www.cree.com/LED-Components-and-Modules/Products/High-Brightness/5mm-Round-P2/503-Series-Color-round>
- datasheet. (2013). <http://www.utp.edu.co>. Obtenido de <http://www.utp.edu.co: http://www.utp.edu.co/~eduque/arquitec/PIC16F877.pdf>
- Decodificador. (04 de Noviembre de 2013). <http://electronica-teoriaypractica.com>. Obtenido de <http://electronica-teoriaypractica.com: http://electronica-teoriaypractica.com/circuito-7448-ttl/>
- digikey. (2015). <http://www.digikey.com>. Obtenido de <http://www.digikey.com: http://www.digikey.com/product-detail/es/digi-international/XBP24CZ7PISB003/602-1240-ND/3026323>
- DigiKey. (2015). <http://www.digikey.com>. Obtenido de <http://www.digikey.com: http://www.digikey.com/product-detail/es/digi-international/XBP24-ASI-001/XBP24-ASI-001-ND/2175373>
- Electronica. (15 de Junio de 2013). <https://electronicaprincipiantes.wordpress.com>. Obtenido de <https://electronicaprincipiantes.wordpress.com: https://electronicaprincipiantes.wordpress.com/tag/2n3904/>
- Farnel. (2014). <http://www.farnell.com>. Obtenido de <http://www.farnell.com: http://www.farnell.com/datasheets/483910.pdf>
- Ferney Salazar Rojas, M. H. (26 de septiembre de 2009). <http://tablerosdijitalesparacamposdeportivos.blogspot.com>. Obtenido de <http://tablerosdijitalesparacamposdeportivos.blogspot.com: http://tablerosdijitalesparacamposdeportivos.blogspot.com/>
- Garcia, J. (18 de Mayo de 2016). <http://aprendrearduino.blogspot.com>. Obtenido de <http://aprendrearduino.blogspot.com: http://aprendrearduino.blogspot.com/2016/06/constantes-en-arduino.html>
- González, A. G. (23 de enero de 2013). <http://panamahitek.com>. Obtenido de <http://panamahitek.com: http://panamahitek.com/arduino-mega-caracteristicas-capacidades-y-donde-conseguirlo-en-panama/>
- Gutiérrez, J. M. (Agosto de 2007). <http://lacajamakerspace.org>. Obtenido de <http://lacajamakerspace.org: http://lacajamakerspace.org/index.php/electronica/arduino/tutoriales?download=1:manual-de-programacion-de-arduino>.

- Henao, J. (2008). <http://biblioteca.usbbog.edu.co:8080>. Obtenido de <http://biblioteca.usbbog.edu.co:8080>: <http://biblioteca.usbbog.edu.co:8080/Biblioteca/BDigital/42134.pdf>
- Hyperbrillo, D. L. (2015). <http://www.shoptronica.com>. Obtenido de <http://www.shoptronica.com>: <http://www.shoptronica.com/images/Led%20alto%20brillo%203-5mm.pdf>
- Iluminacionalve. (2015). <http://www.iluminacionalve.com>. Obtenido de <http://www.iluminacionalve.com>: <http://www.iluminacionalve.com/la-vida-util-del-led/>
- Industrial, Z. (2016). <http://www.zonaindustrial.cl>. Obtenido de <http://www.zonaindustrial.cl>: <http://www.zonaindustrial.cl/product/dr-120-24>
- Integrado, C. (15 de enero de 2012). <http://computo-integrado.blogspot.com>. Obtenido de <http://computo-integrado.blogspot.com>: <http://computo-integrado.blogspot.com/2012/01/microcontroladores-pic.html>
- Julian, P. (2015). <http://slideplayer.es>. Obtenido de <http://slideplayer.es>: <http://slideplayer.es/slide/1641777/>
- Krylov, V. (2003). <http://www.screens.ru>. Obtenido de <http://www.screens.ru>: <http://www.screens.ru/es/2003/7.html>
- Leon. (2015). <http://www.hbleds.com.mx>. Obtenido de <http://www.hbleds.com.mx>: <http://www.hbleds.com.mx/informacion-tecnica/>
- Luis. (03 de Mayo de 2012). <http://colombiansolarsystems.com>. Obtenido de <http://colombiansolarsystems.com>: <http://colombiansolarsystems.com/contenido-definicion-de-led-11.html>
- Martínez, F. (22 de Mayo de 2015). <https://openwebinars.net>. Obtenido de <https://openwebinars.net>: <https://openwebinars.net/tutorial-arduino-comunicacion-serie/>
- Mecacronica. (12 de Septiembre de 2014). <http://55555213.blogspot.com>. Obtenido de <http://55555213.blogspot.com>: <http://55555213.blogspot.com/?view=snapshot>
- Meteo. (2011). <http://meteo.ieec.uned.es>. Obtenido de <http://meteo.ieec.uned.es>: http://meteo.ieec.uned.es/www_Usumeteog/comp_comb_decodificados.html
- Microchip. (2015). <http://ww1.microchip.com>. Obtenido de <http://ww1.microchip.com>:

<http://ww1.microchip.com/downloads/en/DeviceDoc/Acc-RN-SMA4-RP-DS.pdf>

Ojeda, L. T. (2015). <http://xbee.cl>. Obtenido de <http://xbee.cl>:
<http://xbee.cl/que-es-xbee/>

Omar. (2015). <http://dfists.ua.es>. Obtenido de <http://dfists.ua.es>:
<http://dfists.ua.es/~jpomares/arduino/index.htm>

Orozco, M. (01 de Mayo de 2014). <https://www.academia.edu>. Obtenido de
<https://www.academia.edu>:
https://www.academia.edu/8586520/Ethernet_Arduino_Servidor_Html

Patagoniatec. (2015). <http://saber.patagoniatec.com>. Obtenido de
<http://saber.patagoniatec.com>: <http://saber.patagoniatec.com/arduino-mega-2560-atmega-mega-arduino-clon-compatible-argentina-tutorial-basico-informacion-arduino-argentina-ptec/>

Proteus8.1. (2016).

Rojas, F. S. (09 de septiembre de 2009). <http://2.bp.blogspot.com>. Obtenido de
<http://2.bp.blogspot.com>:
http://2.bp.blogspot.com/_MW362jIropA/Sr5GbdUAmMI/AAAAAAAAAB0/6TbyPTTRe3k/s320/3db972ca94b06b890f27c678f365-1-4%5B1%5D.jpg

SAC, B. L. (01 de octubre de 2008). <https://barcoledssac.wordpress.com>.
Obtenido de <https://barcoledssac.wordpress.com>:
<https://barcoledssac.wordpress.com/tag/publicidad-electronica-para-eventos-comerciales/>

Sellers, P. (2015). <https://www.academia.edu>. Obtenido de
<https://www.academia.edu>:
https://www.academia.edu/9699463/EI_Microcontrolador_PIC16F877

Sparkfun. (2015). <https://www.sparkfun.com>. Obtenido de
<https://www.sparkfun.com>: <https://www.sparkfun.com/products/145>

Tajani, A. G. (2013). <http://www.monografias.com>. Obtenido de
<http://www.monografias.com>:
<http://www.monografias.com/trabajos102/algebra-boole-aplicaciones-elctronica-digital/algebra-boole-aplicaciones-elctronica-digital.shtml>

Valero, A. (14 de Diciembre de 2014). <http://diwo.bq.com>. Obtenido de
<http://diwo.bq.com>: <http://diwo.bq.com/variables-en-arduino/>

Anexos

INDICE DE ANEXOS

Anexo A Diagrama esquemático del Arduino mega 2560

Anexo B Especificaciones técnicas del Arduino mega 2560

Anexo C Especificaciones técnicas del Xbee Pro S1

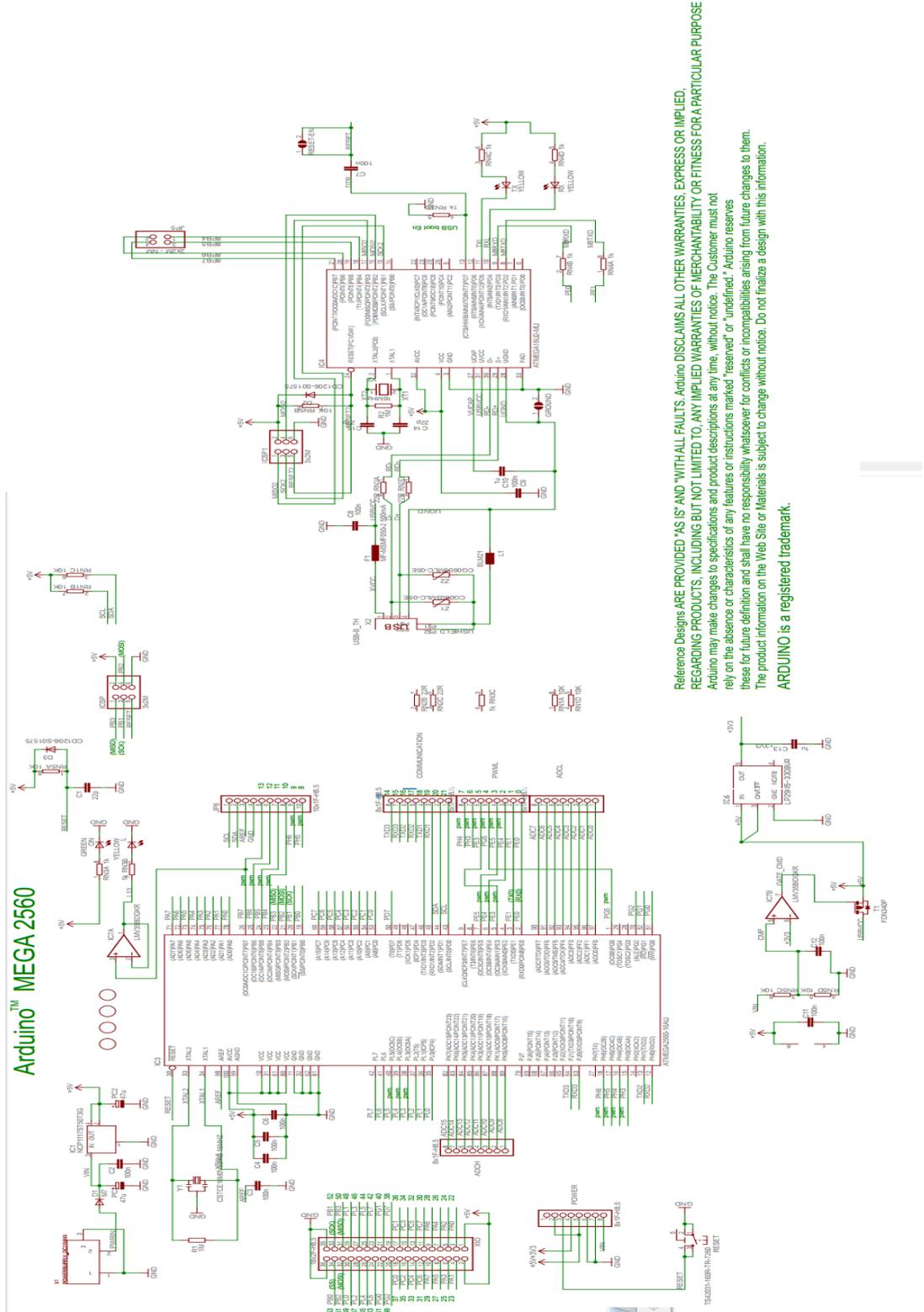
Anexo D Especificaciones técnicas del Xbee Pro S1

Anexo E Diagrama del display de 7 segmento cátodo común

Anexo F Programación realizada en la plataforma Arduino

Anexo A

Diagrama esquemático del Arduino mega 2560



Anexo B

Especificaciones técnicas del Arduino mega 2560

Technical Specification

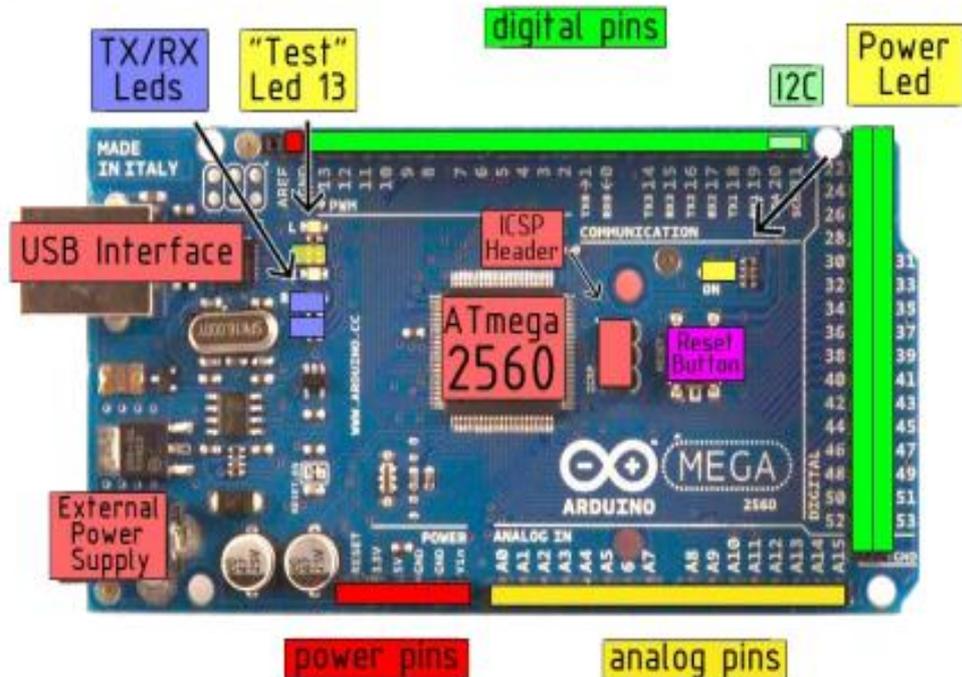


EAGLE files: [arduino-mega2560-reference-design.zip](#) Schematic: [arduino-mega2560-schematic.pdf](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board

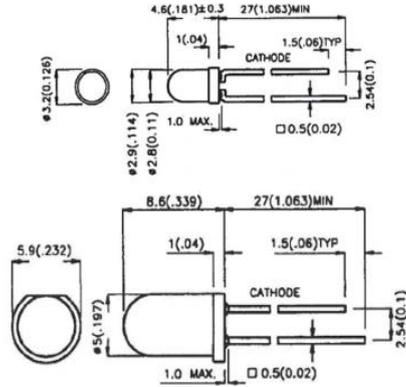


Anexo C

Especificaciones técnicas del Xbee Pro S1



Diodo Led Hyperbrillo 3 y 5mm



Rojo

Emitted Color : RED
 Size (mm) : 3/5mm
 Lens Color : Water Clear
 Peak Wave Length (nm) : 620 ~ 630
 Forward Voltage (V) : 1.8 ~ 2.2
 Reverse Current (uA) : <=30
 Life Rating : 100,000 Hs
 Viewing Angle : 20 ~ 25°
 Absolute Maximum Ratings (Ta=25°C)
 Max Power Dissipation : 80mw
 Max Continuous Forward Current : 30mA
 Max Peak Forw.Current for 10% Pulse Width:75mA
 Reverse Voltage : 5V
 Lead Soldering Temperature : 240°C (<5Sec)
 Operating Temperature Range : -25°C ~ +85°C
 Preservative Temperature Range : -30°C~+100°C

Azul

Emitted Color : OCEAN BLUE
 Size (mm) : 3/5mm
 Lens Color : Water Clear
 Peak Wave Length (nm) : 460 ~ 470
 Forward Voltage (V) : 3.2 ~ 3.8
 Reverse Current (uA) : <=30
 Life Rating : 100,000 Hs
 Viewing Angle : 20 ~ 25°
 Absolute Maximum Ratings (Ta=25°C)
 Max Power Dissipation : 80mw
 Max Continuous Forward Current : 30mA
 Max Peak Forw.Current for 10% Pulse Width:75mA
 Reverse Voltage : 5~6V
 Lead Soldering Temperature : 240°C (<5Sec)
 Operating Temperature Range : -25°C ~ +85°C
 Preservative Temperature Range : -30°C~+100°C

Verde

Emitted Color : PURE GREEN
 Size (mm) : 3/5mm
 Lens Color : Water Clear
 Peak Wave Length (nm) : 515 ~ 530
 Forward Voltage (V) : 3.2 ~ 3.8
 Reverse Current (uA) : <=30
 Life Rating : 100,000 Hs
 Viewing Angle : 20 ~ 25°
 Absolute Maximum Ratings (Ta=25°C)
 Max Power Dissipation : 80mw
 Max Continuous Forward Current : 30mA
 Max Peak Forw.Current for 10% Pulse Width:75mA
 Reverse Voltage : 5~6V
 Lead Soldering Temperature : 240°C (<5Sec)
 Operating Temperature Range : -25°C ~ +85°C
 Preservative Temperature Range : -30°C~+100°C

Blanco/Blanco cálido

Emitted Color : WHITE/WARM WHITE
 Size (mm) : 3/5mm
 Lens Color : Water Clear
 Peak Wave Length (nm) : N/A
 Forward Voltage (V) : 3.2 ~ 3.8
 Reverse Current (uA) : <=30
 Life Rating : 100,000 Hours
 Viewing Angle : 20 ~ 25°
 Absolute Maximum Ratings (Ta=25°C)
 Max Power Dissipation : 80mw
 Max Continuous Forward Current : 30mA
 Max Peak Forw.Current for 10% Pulse Width:75mA
 Reverse Voltage : 5~6V
 Lead Soldering Temperature : 240°C (<5Sec)
 Operating Temperature Range : -25°C ~ +85°C
 Preservative Temperature Range : -30°C~+100°C

Amarillo

Emitted Colour : YELLOW
 Size (mm) : 3/5mm
 Lens Colour : Water Clear
 Peak Wave Length (nm) : 585 ~ 595
 Forward Voltage (V) : 1.8 ~ 2.2
 Reverse Current (uA) : <=30
 Life Rating : 100,000 Hours
 Viewing Angle : 20 ~ 25°
 Absolute Maximum Ratings (Ta=25°C)
 Max Power Dissipation : 80mw
 Max Continuous Forward Current : 30mA
 Max Peak Forward Current : 75mA
 Reverse Voltage : 5~6V
 Lead Soldering Temperature : 240°C (<5Sec)
 Operating Temperature Range : -25°C ~ +85°C
 Preservative Temperature Range : -30°C~+100°C

Naranja

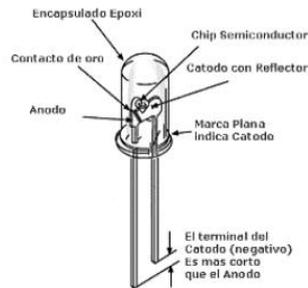
Emitted Colour : ORANGE
 Size (mm) : 3/5mm
 Lens Colour : Water Clear
 Peak Wave Length (nm) : 600 ~ 610
 Forward Voltage (V) : 1.8 ~ 2.2
 Reverse Current (uA) : <=30
 Life Rating : 100,000 Hours
 Viewing Angle : 20 ~ 25°
 Absolute Maximum Ratings (Ta=25°C)
 Max Power Dissipation : 80mw
 Max Continuous Forward Current : 30mA
 Max Peak Forw.Current for 10% Pulse Width:75mA
 Reverse Voltage : 5~6V
 Lead Soldering Temperature : 240°C (<5Sec)
 Operating Temperature Range : -25°C ~ +85°C
 Preservative Temperature Range : -30°C~+100°C

Ultravioleta

Emitted Color : ULTRA VIOLET
 Size (mm) : 3/5mm
 Lens Color : Water Clear
 Peak Wave Length (nm) : 410 - 4205
 Forward Voltage (V) : 3.2 ~ 3.8
 Reverse Current (uA) : <=30
 Life Rating : 100,000 Hours
 Viewing Angle : 20 ~ 25°
 Absolute Maximum Ratings (Ta=25°C)
 Max Power Dissipation : 80mw
 Max Continuous Forward Current : 30mA
 Max Peak Forw.Current for 10% Pulse Width:75mA
 Reverse Voltage : 5~6V
 Lead Soldering Temperature : 240°C (<5Sec)
 Operating Temperature Range : -25°C ~ +85°C
 Preservative Temperature Range : -30°C~+100°C

Rosa

Emitted Color : PINK
 Size (mm) : 3/5mm
 Lens Color : Water Clear
 Peak Wave Length (nm) : 430-440
 Forward Voltage (V) : 3.2 ~ 3.4
 Reverse Current (uA) : <=30
 Life Rating : 100,000 Hours
 Viewing Angle : 20 ~ 25°
 Absolute Maximum Ratings (Ta=25°C)
 Max Power Dissipation : 80mw
 Max Continuous Forward Current : 20mA
 Max Peak Forw.Current for 10% Pulse Width:75mA
 Reverse Voltage : 5~6V
 Lead Soldering Temperature : 240°C (<5Sec)

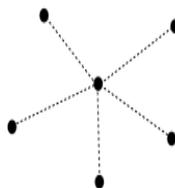


Anexo D

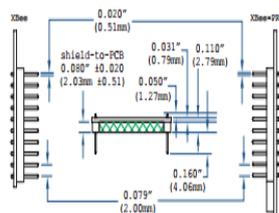
Especificaciones técnicas del Xbee Pro S1

SPECIFICATIONS	Legacy XBee® S1 802.15.4	Legacy XBee-PRO® S1 802.15.4
PERFORMANCE		
RF DATA RATE	250 kbps	250 kbps
INDOOR/URBAN RANGE	100 ft (30 m)	300 ft (100 m)
OUTDOOR/RF LINE-OF-SIGHT RANGE	300 ft (100 m)	1 mi (1.6 km)
TRANSMIT POWER	1 mW (+0 dBm)	60 mW (+18 dBm)*
RECEIVER SENSITIVITY (1% PER)	-92 dBm	-100 dBm
DIGI HARDWARE	S1	
TRANSCIEVER CHIPSET	Freescale MC13212	
FEATURES		
SERIAL DATA INTERFACE	3.3V CMOS UART	
CONFIGURATION METHOD	API or AT Commands, local or over-the-air	
FREQUENCY BAND	2.4 GHz	
INTERFERENCE IMMUNITY	DSSS (Direct Sequence Spread Spectrum)	
SERIAL DATA RATE	1200 bps - 250 kbps	
ADC INPUTS	(6) 10-bit ADC inputs	
DIGITAL I/O	8	
ANTENNA OPTIONS	Chip, Wire Whip, U.F.L., & RPSMA	
NETWORKING & SECURITY		
ENCRYPTION	128-bit AES	
RELIABLE PACKET DELIVERY	Retries/Acknowledgments	
IDS AND CHANNELS	PAN ID, 64-bit IEEE MAC, 16 Channels	
POWER REQUIREMENTS		
SUPPLY VOLTAGE	2.8 - 3.4VDC	2.8 - 3.4VDC
TRANSMIT CURRENT	45 mA @ 3.3VDC	215 mA @ 3.3VDC
RECEIVE CURRENT	50 mA @ 3.3VDC	55 mA @ 3.3VDC
POWER-DOWN CURRENT	<10 uA @ 25° C	
REGULATORY APPROVALS		
FCC (USA)	OUR-XBEE	OUR-XBEEPRO
IC (CANADA)	4214A-XBEE	4214A-XBEEPRO
ETSI (EUROPE)	Yes	Yes - Max TX 10 mW
C-TICK AUSTRALIA	Yes	
TELEC (JAPAN)	Yes	

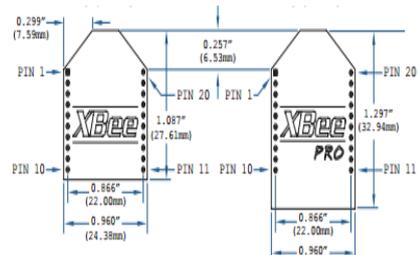
802.15.4 - Star



(Side Views)



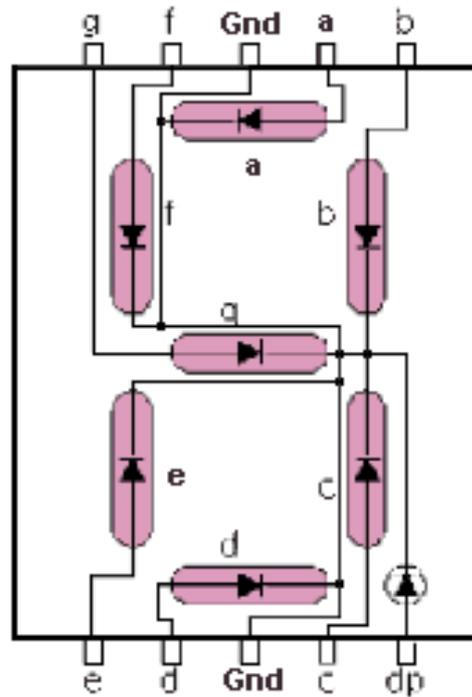
(Top View)



Anexo E

Diagrama del display de 7 segmento cátodo común

Common Cathode



		Catodo Comun							
		Numero	A	B	C	D	E	F	G
Enable	0	1	1	1	1	1	1	1	0
0	1	0	1	1	0	0	0	0	0
0	2	1	1	0	1	1	0	1	1
0	3	1	1	1	1	0	0	1	1
0	4	0	1	1	0	0	1	1	1
0	5	1	0	1	1	0	1	1	1
0	6	1	0	1	1	1	1	1	1
0	7	1	1	1	0	0	0	0	0
0	8	1	1	1	1	1	1	1	1
0	9	1	1	1	1	0	1	1	1

Anexo F

Programación realizada en la plataforma Arduino

```
byte a;  
int a1 = 53;  
int b1 = 52;  
int c1 = 51;  
int d1 = 50;
```

```
int a2 = 49;  
int b2 = 48;  
int c2 = 47;  
int d2 = 46;
```

```
int a3 = 45;  
int b3 = 44;  
int c3 = 43;  
int d3 = 42;
```

```
int a4 = 41;  
int b4 = 40;  
int c4 = 39;  
int d4 = 38;
```

```
int a5 = 37;  
int b5 = 36;  
int c5 = 35;  
int d5 = 34;
```

```
int a6 = 33;  
int b6 = 32;  
int c6 = 31;  
int d6 = 30;
```

```
int a7 = 29;  
int b7 = 28;  
int c7 = 27;  
int d7 = 26;
```

```
//TIEMPO  
int at1 = 21;  
int bt1 = 20;  
int ct1 = 19;  
int dt1 = 18;
```

```
int at2 = 17;  
int bt2 = 16;  
int ct2 = 15;
```

```
int dt2 = 14;
```

```
int at3 = 2;  
int bt3 = 3;  
int ct3 = 4;  
int dt3 = 5;
```

```
int at4 = 6;  
int bt4 = 7;  
int ct4 = 8;  
int dt4 = 9;
```

```
int T1=0, T2=0;  
int T3=0, T4=0;  
int T5=0, T6=0;  
int T7=0;
```

```
int cont_seg1;  
int cont_min1;  
int cont_segH;  
int cont_minH;
```

```
int auxcont_seg1;  
int auxcont_min1;  
int auxcont_segH;  
int auxcont_minH;
```

```
int val=0;  
int ini=0;  
int i=0;
```

```
int setM=10;  
int setS=11;  
int inicio=12;  
int parar=13;
```

```
int led=22;
```

```
int CERO[]={0,0,0,0}; // NUMERO CERO  
int UNO[]={0,0,0,1}; // NUMERO UNO  
int DOS[]={0,0,1,0}; // NUMERO DOS  
int TRES[]={0,0,1,1}; // NUMERO TRES  
int CUATRO[]={0,1,0,0}; // NUMERO CUATRO  
int CINCO[]={0,1,0,1}; // NUMERO CINCO  
int SEIS[]={0,1,1,0}; // NUMERO SEIS  
int SIETE[]={0,1,1,1}; // NUMERO SIETE  
int OCHO[]={1,0,0,0}; // NUMERO OCHO  
int NUEVE[]={1,0,0,1}; // NUMERO NUEVE
```

```
void setup()
{
  Serial.begin(9600);
  pinMode(a1,OUTPUT);
  pinMode(b1,OUTPUT);
  pinMode(c1,OUTPUT);
  pinMode(d1,OUTPUT);

  pinMode(a2,OUTPUT);
  pinMode(b2,OUTPUT);
  pinMode(c2,OUTPUT);
  pinMode(d2,OUTPUT);

  pinMode (a3,OUTPUT);
  pinMode (b3,OUTPUT);
  pinMode (c3,OUTPUT);
  pinMode (d3,OUTPUT);

  pinMode (a4,OUTPUT);
  pinMode (b4,OUTPUT);
  pinMode (c4,OUTPUT);
  pinMode (d4,OUTPUT);

  pinMode(a5,OUTPUT);
  pinMode(b5,OUTPUT);
  pinMode(c5,OUTPUT);
  pinMode(d5,OUTPUT);

  pinMode (a6,OUTPUT);
  pinMode (b6,OUTPUT);
  pinMode (c6,OUTPUT);
  pinMode (d6,OUTPUT);

  pinMode (a7,OUTPUT);
  pinMode (b7,OUTPUT);
  pinMode (c7,OUTPUT);
  pinMode (d7,OUTPUT);

  pinMode(at1,OUTPUT);
  pinMode(bt1,OUTPUT);
  pinMode(ct1,OUTPUT);
  pinMode(dt1,OUTPUT);

  pinMode(at2,OUTPUT);
  pinMode(bt2,OUTPUT);
  pinMode(ct2,OUTPUT);
  pinMode(dt2,OUTPUT);
}
```

```
pinMode (at3,OUTPUT);
pinMode (bt3,OUTPUT);
pinMode (ct3,OUTPUT);
pinMode (dt3,OUTPUT);
```

```
pinMode (at4,OUTPUT);
pinMode (bt4,OUTPUT);
pinMode (ct4,OUTPUT);
pinMode (dt4,OUTPUT);
pinMode(led,OUTPUT);
```

```
cont_minl=0;
cont_minh=1;
cont_segl=0;
cont_segh=0;
```

```
auxcont_segl=0;
auxcont_minl=0;
auxcont_segh=0;
auxcont_minh=0;
```

```
digitalWrite(led,LOW);
```

```
BCD(0,d1,c1,b1,a1);
BCD(0,d2,c2,b2,a2);
BCD(0,d3,c3,b3,a3);
BCD(0,d4,c4,b4,a4);
BCD(0,d5,c5,b5,a5);
BCD(0,d6,c6,b6,a6);
BCD(0,d7,c7,b7,a7);
}
void loop()
{
  leer_entradas();
  if(ini==1)
  {
    if(cont_minl==0&&cont_minh==0&&cont_segl==0&&cont_segh==0)
    {
      ini=0;
      digitalWrite(led,HIGH);
      delay(2000);
      digitalWrite(led,LOW);
      mostrarsL();
      mostrarsH();
      mostrarmL();
      mostrarmH();
      delay(3000);
    }
  }
}
```

```

    cont_minl=auxcont_minl;
    cont_minh=auxcont_minh;
    cont_segl=auxcont_segl;
    cont_segh=auxcont_segh;
}
else
{
    contador();
}
}
mostrarsL();
mostrarsH();
mostrarmL();
mostrarmH();
}
// subrutinas
void leer_entradas()
{
    if(Serial.available(>0)
    {
        a=Serial.read();
        if(a=='m') // lee boton minutos
        {
            ini=0;
            cont_minl++;
            if(cont_minl>9)
            {
                cont_minl=0;
                cont_minh++;
                if(cont_minh>5)
                {
                    cont_minh=0;
                }
            }
        }
    }
    if(a=='n')// lee boton segundos
    {
        ini=0;
        cont_segl++;
        if(cont_segl>9)
        {
            cont_segl=0;
            cont_segh++;
            if(cont_segh>5)
            {
                cont_segh=0;
            }
        }
    }
}

```

```

    }
    if(a=='o') // lee boton inicio
    {
        ini=1;
        digitalWrite(led,LOW);
        auxcont_segl=cont_segl;
        auxcont_minl=cont_minl;
        auxcont_segh=cont_segh;
        auxcont_minh=cont_minh;
    }
    if(a=='p') // lee boton parar
    {
        ini=0;
    }
    if(a=='q') // lee boton parar
    {
        cont_minl=0;
        cont_minh=1;
        cont_segl=0;
        cont_segh=0;
        auxcont_segl=0;
        auxcont_minl=0;
        auxcont_segh=0;
        auxcont_minh=0;
    }

```

```

T1=0;
T2=0;
T3=0;
T4=0;
T5=0;
T6=0;
T7=0;
}

```

```

//LOCAL
if(a=='a')//Codigo para aumentar
{
    T1++;
    if(T1>9){T2++; T1=0;}
    if(T2>9){T2=0;}
}
if(a=='b') //Codigo para disminuir
{
    T1--;
    if(T1<0){T2--; T1=9;}
    if(T2<0){T2=9;}
}
if(a=='c') //Codigo para resetear

```

```

{
  T1=0;
  T2=0;
}
//VISITANTE
if(a=='d') //Codigo para aumentar
{
  T3++;
  if(T3>9){T4++; T3=0;}
  if(T4>9){T4=0;}
}
if(a=='e') //Codigo para disminuir
{
  T3--;
  if(T3<0){T4--; T3=9;}
  if(T4<0){T4=9;}
}
if(a=='f') //Codigo para resetear
{
  T3=0;
  T4=0;
}
//periodo
if(a=='g') //Codigo para aumentar
{
  T5++;
  if(T5>4)
  {
    T5=0;
  }
}
if(a=='h') //Codigo para resetear
{
  T5=0;
}
//falta local
if(a=='i') //Codigo para aumentar
{
  T6++;
  if(T6>5)
  {
    T6=0;
  }
}
if(a=='j') //Codigo para resetear
{
  T6=0;
}

```

```

// faltavi sitante

if(a=='k') //Codigo para aumentar
{
    T7++;
    if(T7>5)
    {
        T7=0;
    }
}
if(a=='l') //Codigo para resetear
{
    T7=0;
}
BCD(T1,d1,c1,b1,a1);
BCD(T2,d2,c2,b2,a2);
BCD(T3,d3,c3,b3,a3);
BCD(T4,d4,c4,b4,a4);
BCD(T5,d5,c5,b5,a5);
BCD(T6,d6,c6,b6,a6);
BCD(T7,d7,c7,b7,a7);
}
}

void contador()
{
    cont_seg1--;
    if(cont_seg1<0)
    {
        cont_seg1=9;
        cont_seg1--;
        if(cont_seg1<0)
        {
            cont_seg1=5;
            cont_min1--;
            if(cont_min1<0)
            {
                cont_min1=9;
                cont_min1--;
                if(cont_min1<0)
                {
                    cont_min1=5;
                }
            }
        }
    }
}
delay(20);
return;

```

```

}

void mostrarsL()
{
    if(cont_seg1==0)
    {
        for(i=0;i<4;i++)
        {
            digitalWrite((dt1+i),CERO[i]);
        }
        return;
    }
    if(cont_seg1==1)
    {
        for(i=0;i<4;i++)
        {
            digitalWrite((dt1+i),UNO[i]);
        }
        return;
    }
    if(cont_seg1==2)
    {
        for(i=0;i<4;i++)
        {
            digitalWrite((dt1+i),DOS[i]);
        }
        return;
    }
    if(cont_seg1==3)
    {
        for(i=0;i<4;i++)
        {
            digitalWrite((dt1+i),TRES[i]);
        }
        return;
    }
    if(cont_seg1==4)
    {
        for(i=0;i<4;i++)
        {
            digitalWrite((dt1+i),CUATRO[i]);
        }
        return;
    }
    if(cont_seg1==5)
    {
        for(i=0;i<=3;i++)
        {

```

```

        digitalWrite((dt1+i),CINCO[i]);
    }
    return;
}
if(cont_seg1==6)
{
    for(i=0;i<=3;i++)
    {
        digitalWrite((dt1+i),SEIS[i]);
    }
    return;
}
if(cont_seg1==7)
{
    for(i=0;i<=3;i++)
    {
        digitalWrite((dt1+i),SIETE[i]);
    }
    return;
}
if(cont_seg1==8)
{
    for(i=0;i<=3;i++)
    {
        digitalWrite((dt1+i),OCHO[i]);
    }
    return;
}
if(cont_seg1==9)
{
    for(i=0;i<=3;i++)
    {
        digitalWrite((dt1+i),NUEVE[i]);
    }
    return;
}
}
void mostrarsH()
{
    if(cont_segH==0)
    {
        for(i=0;i<=3;i++)
        {
            digitalWrite((dt2+i),CERO[i]);
        }
        return;
    }
}
if(cont_segH==1)

```

```

    {
        for(i=0;i<=3;i++)
        {
            digitalWrite((dt2+i),UNO[i]);
        }
        return;
    }
if(cont_segh==2)
{
    for(i=0;i<=3;i++)
    {
        digitalWrite((dt2+i),DOS[i]);
    }
    return;
}
if(cont_segh==3)
{
    for(i=0;i<=3;i++)
    {
        digitalWrite((dt2+i),TRES[i]);
    }
    return;
}
if(cont_segh==4)
{
    for(i=0;i<=3;i++)
    {
        digitalWrite((dt2+i),CUATRO[i]);
    }
    return;
}
if(cont_segh==5)
{
    for(i=0;i<=3;i++)
    {
        digitalWrite((dt2+i),CINCO[i]);
    }
    return;
}
}
void mostrarmL()
{
    if(cont_minl==0)
    {
        for(i=0;i<4;i++)
        {
            digitalWrite((dt3-i),CERO[i]);
        }
    }
}

```

```

    return;
}
if(cont_minl==1)
{
    for(i=0;i<4;i++)
    {
        digitalWrite((dt3-i),UNO[i]);
    }
    return;
}
if(cont_minl==2)
{
    for(i=0;i<4;i++)
    {
        digitalWrite((dt3-i),DOS[i]);
    }
    return;
}

if(cont_minl==3)
{
    for(i=0;i<4;i++)
    {
        digitalWrite((dt3-i),TRES[i]);
    }
    return;
}
if(cont_minl==4)
{
    for(i=0;i<4;i++)
    {
        digitalWrite((dt3-i),CUATRO[i]);
    }
    return;
}
if(cont_minl==5)
{
    for(i=0;i<=3;i++)
    {
        digitalWrite((dt3-i),CINCO[i]);
    }
    return;
}
if(cont_minl==6)
{
    for(i=0;i<=3;i++)
    {
        digitalWrite((dt3-i),SEIS[i]);
    }
}

```

```

    }
    return;
}
if(cont_minl==7)
{
    for(i=0;i<=3;i++)
    {
        digitalWrite((dt3-i),SIETE[i]);
    }
    return;
}
if(cont_minl==8)
{
    for(i=0;i<=3;i++)
    {
        digitalWrite((dt3-i),OCHO[i]);
    }
    return;
}
if(cont_minl==9)
{
    for(i=0;i<=3;i++)
    {
        digitalWrite((dt3-i),NUEVE[i]);
    }
    return;
}
}
void mostrarmH()
{
    if(cont_minh==0)
    {
        for(i=0;i<=3;i++)
        {
            digitalWrite((dt4-i),CERO[i]);
        }
        return;
    }
    if(cont_minh==1)
    {
        for(i=0;i<=3;i++)
        {
            digitalWrite((dt4-i),UNO[i]);
        }
        return;
    }
    if(cont_minh==2)
    {

```

```

        for(i=0;i<=3;i++)
        {
            digitalWrite((dt4-i),DOS[i]);
        }
        return;
    }
    if(cont_minh==3)
    {
        for(i=0;i<=3;i++)
        {
            digitalWrite((dt4-i),TRES[i]);
        }
        return;
    }
    if(cont_minh==4)
    {
        for(i=0;i<=3;i++)
        {
            digitalWrite((dt4-i),CUATRO[i]);
        }
        return;
    }
    if(cont_minh==5)
    {
        for(i=0;i<=3;i++)
        {
            digitalWrite((dt4-i),CINCO[i]);
        }
        return;
    }
}
void BCD(byte Contador, byte D, byte C, byte B, byte A)
{
    switch(Contador)
    {
        case 0:
            digitalWrite(D,LOW);digitalWrite(C,LOW);digitalWrite(B,LOW);digitalWrite(A,LOW);
            break;
        case 1:
            digitalWrite(D,LOW);digitalWrite(C,LOW);digitalWrite(B,LOW);digitalWrite(A,HIGH);
            break;
        case 2:
            digitalWrite(D,LOW);digitalWrite(C,LOW);digitalWrite(B,HIGH);digitalWrite(A,LOW);
            break;
    }
}

```

```
case 3:
digitalWrite(D,LOW);digitalWrite(C,LOW);digitalWrite(B,HIGH);digitalWrite(A,
HIGH);
    break;
case 4:
digitalWrite(D,LOW);digitalWrite(C,HIGH);digitalWrite(B,LOW);digitalWrite(A,
LOW);
    break;
case 5:
digitalWrite(D,LOW);digitalWrite(C,HIGH);digitalWrite(B,LOW);digitalWrite(A,
HIGH);
    break;
case 6:
digitalWrite(D,LOW);digitalWrite(C,HIGH);digitalWrite(B,HIGH);digitalWrite(A,
LOW);
    break;
case 7:
digitalWrite(D,LOW);digitalWrite(C,HIGH);digitalWrite(B,HIGH);digitalWrite(A,
HIGH);
    break;
case 8:
digitalWrite(D,HIGH);digitalWrite(C,LOW);digitalWrite(B,LOW);digitalWrite(A,
LOW);
    break;
case 9:
digitalWrite(D,HIGH);digitalWrite(C,LOW);digitalWrite(B,LOW);digitalWrite(A,
HIGH);
    break;
}
```

DATOS PERSONALES



Nombre: Toapanta Caguana Edison Geovanny
Nacionalidad: Ecuatoriana
Fecha De Nacimiento: 27 DE SEPTIEMBRE DE 1993
Cedula de Ciudadanía: 0503500605
Teléfonos: 0998116588/032728202
Correo Electrónico: edyilove@hotmail.com
Dirección Domicilio: SALCEDO BARRIO LA ARGENTINA
(PASOLATERAL)

ESTUDIOS REALIZADOS:

Primaria:

- Escuela Cristóbal Colon Salcedo

Secundaria:

- Colegio Técnico Trajano Naranjo Latacunga

Superior:

- Unidad de Gestión de Tecnologías "ESPE"

TÍTULOS OBTENIDOS:

- Bachiller técnico "Electrónico"
- Tecnólogo en Electrónica Mención Instrumentación & Aviónica

EXPERIENCIAS LABORALES:

- Instalación de redes y cableado para COTOPAXI TV
- Asistente de QA/QC eléctrico LARGENERGY
- Centro de investigación y desarrollo CID-FAE
- Instalación y mantenimiento del avión escuela UGT

ACEPTACIÓN DEL USUARIO

Latacunga, febrero del 2017

Yo, ING PABLO PILATÁSIG en calidad de encargado del Laboratorio de Instrumentación Virtual de la Unidad de Gestión de Tecnologías, me permito informar lo siguiente:

El proyecto técnico elaborado por el Sr. **TOAPANTA CAGUANA EDISON GEOVANNY**, con el tema: **“REHABILITACIÓN DEL MARCADOR ELECTRÓNICO MEDIANTE LEDS DE ALTO BRILLO Y MANDO DE RADIOFRECUENCIA PARA LA UNIDAD DE GESTIÓN DE TECNOLOGÍAS”**, ha sido efectuado de forma satisfactoria en las dependencias de mi cargo y que la misma cuenta con todas las garantías de funcionamiento, por lo cual extiendo este aval que respalda el trabajo realizado por el mencionado estudiante.

Por tanto me hago cargo de todas las instalaciones realizadas por el Sr. estudiante.

Atentamente,

ING. PABLO PILATÁSIG

ENCARGADO DEL LABORATORIO DE INSTRUMENTACIÓN VIRTUAL

Latacunga, febrero del 2017

HOJA DE LEGALIZACIÓN DE FIRMAS

**DEL CONTENIDO DE LA PRESENTE INVESTIGACIÓN SE
RESPONSABILIZA EL AUTOR**

TOAPANTA CAGUANA EDISON GEOVANNY

ID L00264971

**DIRECTOR DE LA CARRERA DE ELECTRÓNICA MENCIÓN
INSTRUMENTACIÓN & AVIÓNICA**

**Ing. Pablo Pilatásig Director Carrera de Electrónica Mención
Instrumentación & Aviónica**