

TABLERO ELECTRONICO MULTIDEPORTIVO CON MANDO INALAMBRICO

ISAIAS DAVID TORO

INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO

FACULTAD DE INGENIERÍA

TECNOLOGÍA EN ELECTRONICA

MEDELLÍN

2017

TABLERO ELECTRONICO MULTIDEPORTIVO CON MANDO INALAMBRICO

ISAIAS DAVID TORO

Trabajo de grados para optar al título de Tecnólogo Eletrônico

Asesor

Carlos Alberto Monsalve Jaramillo
Especialista en Gerencia de Mantimento

INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO
FACULTAD DE INGENIERÍA
TECNOLOGÍA EN ELECTRONICA
MEDELLÍN

2017

Contenido

	Pág.
Introducción	9
1.Planteamiento del problema.....	10
1.1 Descripción.....	10
1.2 Formulación del problema.....	10
2. Justificación.....	11
3. Objetivos	12
3.1 Objetivos generales	12
3.2 Objetivos específicos.....	12
5. Marco teórico	13
5.1 Microcontrolador PIC 16F887	13
5.2 SAA1064	15
5.3 ULN2803A	16
5.4 Display Matricial o Matriz de leds	17
6.Resultado.....	18
6.1 Programación del PIC.....	21
6.1.1 Programación del pseudocódigo.....	21
6.1.2 Compilación del programa.....	23
6.1.3 Programador de microcontroladores PIC	26
6.2 Diseño del circuito Electrónico.	30
6.3 Montaje del circuito.....	32
6.4 Diseño de la PCB.....	33
6.5 Control de mando.....	34
7. Conclusiones	36
8. Bibliografía	37
9.Referencias bibliográficas.....	38
10. Anexos.....	39

Lista de figuras

<i>Figura 1</i> Diagrama del PIC 16f887	14
<i>Figura 2</i> Diagrama del integrado SAA1064.....	15
<i>Figura 3</i> Diagrama del ULN2803A.....	16
<i>Figura 4</i> Diagrama de una matriz de leds 1588AS 8x8.....	17
<i>Figura 5</i> Recolección de información manual de los eventos deportivos	18
<i>Figura 6</i> Pérdida de la información recolectada.....	18
<i>Figura 7</i> Integrado TM1637	19
<i>Figura 8</i> Integrado Max9611.....	19
<i>Figura 9</i> Esquema del Pseudocódigo	20
<i>Figura 10</i> Software de Proteus 8.0 Profesional.....	21
<i>Figura 11</i> Ventana inicial de C++	22
<i>Figura 12</i> Crear un nuevo proyecto en c++.....	22
<i>Figura 13</i> Selección del lenguaje y nombre del nuevo proyecto.....	22
<i>Figura 14</i> Programa activado para ser utilizado.....	23
<i>Figura 15</i> logo del PIC C COMPILER	24
<i>Figura 16</i> Ventana inicial del PIC CCS PCW.....	24
<i>Figura 17</i> Nuevo proyecto en el PCW	25
<i>Figura 18</i> Importación del programa en C++ a PCW	25
<i>Figura 19</i> Indicadores de compilación del programa.....	26
<i>Figura 20</i> Programador Pickit 2	26
<i>Figura 21</i> Diagrama interno del Pickit 2.....	27
<i>Figura 22</i> Instalación del software del pickit 2.....	28
<i>Figura 23</i> Reconocimiento y verificación del programa del pickit 2.....	28
<i>Figura 24</i> Reconocimiento del PIC 16f887.....	29
<i>Figura 25</i> Importación del programa compilado en PIC CCS PWC al Pickit 2	29
<i>Figura 26</i> Programación del microcontrolador	30
<i>Figura 27</i> Ventana principal de Proteus	30
<i>Figura 28</i> Creación del nuevo proyecto en PROTEUS.....	31
<i>Figura 29</i> Activación de las herramientas para comenzar el diseño	31
<i>Figura 30</i> Columna para la selección de componentes electrónicos.....	31
<i>Figura 31</i> Simulación del circuito completo	32
<i>Figura 32</i> Montaje de los componentes electrónicos en Boards.....	33
<i>Figura 33</i> Diseño de la PCB.....	33
<i>Figura 34</i> Circuito del control remoto en Proteus.....	34
<i>Figura 35</i> Montaje del control en la board.....	34
<i>Figura 36</i> Tablero deportivo completo.....	35

Resumen

La institución Universitaria Pascual Bravo, cuenta con diversos escenarios para la práctica de deportes, estos escenarios hoy en día no cuentan con un tablero deportivo donde la gente pueda saber lo que ocurre dentro del desarrollo de un evento que se esté presentando, de manera que si se presenta algún evento los datos son llevados por medio de un lápiz y un cuaderno u hojas, además de que los datos son tomados con un celular o cronometro no convencional para deportes por una persona que luego le comunica a otra persona de registrar los datos, se ha presentado que una misma persona debe registrar los datos y llevar el tiempo, esto puede ocasionar confusión y distorsión en la información, para solventar esta situación se fabricó un tablero multideportivo que se pueda manejar por medio de un control.

Para la fabricación de este, como primer paso fue la realización de un pseudocódigo lógico y bien estructurado para que luego sirviera como guía en la realización del programa que ejecute las órdenes y acciones para así poder brindarle de la mejor manera al espectador la información del evento, luego vincular dicha programación con la simulación del circuito del tablero electrónico en Proteus, finalizando con la integración del control y verificar su funcionamiento, simultáneamente se consultó la facilidad para adquirir los componentes electrónicos ya que en Colombia no todo los integrados se encuentran con gran facilidad y si se hallan se consiguen a elevados costos; además del poco conocimiento y uso del software Proteus 8.0 Profesional que traen consigo muchas herramientas de más , luego de conseguir todos los materiales y practicar con el software se realizaron las simulaciones y montajes para las respectivas pruebas.

Se obtuvo un tablero electrónico multideportivo en el cual podemos observar la información de deportes tales como Microfútbol, Fútbol, Baloncesto y Voleibol que es controlado por medio de un control RF que tiene un alcance de aproximadamente 25 mts, con cuyo control se puede graduar el tiempo, modificar los marcadores y los tiempos haciéndolo accesible para los espectadores.

Abstract

The University institution Pascual Bravo, tells gift diverse scenes for the practice of sports, these scenes nowadays do not possess a sports board where the people could know what happens inside the development of an event that one is presenting, so that if one presents some event the information is taken by means of a pencil and a notebook or leaves, besides which the information is taken by the cellular one or not conventional chronometer for sports for a person who then reports to another person of registering the information, has appeared that the same person must register the information and go the time, this can cause confusion and distortion of the information, to settle this situation made a multisport board that it could handle by means of a control.

For the manufacture of this one, since the first step was the accomplishment of a logical pseudocódigo and structured good in order that then it was serving as guide in the accomplishment of the program that executes the orders and actions this way to be able to offer to him of the best way to the spectator the information of the event, then link the above mentioned programming with the simulation of the circuit of the electronic board in Proteus, Finishing with the integration of the control and to check his functioning, simultaneously the facility consulted to acquire the electronic components since in Colombia not quite the integrated ones meet great facility and if they are situated they are obtained to high costs; besides little knowledge and use of the software Proteus 8.0 Professional that they bring I obtain many tools of more, after obtaining all the materials and practising with the software the simulations and assemblies were realized for the respective tests.

There was obtained an electronic multisport board in which we can observe the information of such sports as Micro football, Football, Basketball and Volleyball that is controlled by means of a control RF that has a scope of approximately 25 mts, with whose control can graduate the time, modify the scoreboards and the times making it accessible for the spectators.

Glosario

Byte: Unidad de información compuesta de 8 bits ("1" o "0"). Esta unidad de información es suficiente para representar un número o una letra entre otros.

Compilador: Un compilador es un pequeño programa informático, que se encarga de traducir (compilar) el código fuente de cualquier aplicación que se esté desarrollando. En pocas palabras, es un software que se encarga de traducir el programa hecho en lenguaje de programación, a un lenguaje de máquina que pueda ser comprendido por el equipo y pueda ser procesado o ejecutado por este. Básicamente, un compilador cuenta con dos partes fundamentales: El Front End y el Back End. Estas partes se complementan para lograr el objetivo del compilador.

Interface: Conexión mecánica o eléctrica que permite el intercambio de información entre dos dispositivos o sistemas. Normalmente se refiere al 20 software y hardware necesarios para unir dos elementos de proceso en un sistema o bien para describir los estándares recomendados para realizar dichas interconexiones.

Lenguaje C: También conocido como “Lenguaje de programación de sistemas” desarrollado en el año 1972 por Dennis Ritchie para UNIX un sistema operativo multiplataforma. El lenguaje C es del tipo lenguaje estructurado como son Pascal, Fortran, Basic.

Memoria flash: Es una forma desarrollada de la memoria EEPROM que permite que múltiples posiciones de memoria sean escritas o borradas en una misma operación de programación mediante impulsos eléctricos, frente a las anteriores que sólo permite escribir o borrar una única celda cada vez. Es por esto que flash permite funcionar a velocidades muy superiores cuando los sistemas emplean lectura y escritura en diferentes puntos de esta memoria al mismo tiempo.

Pcw: Es un Compilador, por ende, un Compilador es un programa informático que traduce un programa escrito en un lenguaje de programación a otro lenguaje de programación, generando un programa equivalente que la máquina será capaz de interpretar. Usualmente el segundo lenguaje es código máquina, pero también puede ser simplemente texto. Este proceso de traducción se conoce como compilación.

Proteus: Es un entorno integrado diseñado para la realización completa de proyectos de construcción de equipos electrónicos en todas sus etapas: diseño, simulación, depuración y construcción.

Protocolo: Un protocolo es un conjunto de estándares que controlan la secuencia de mensajes que ocurren durante una comunicación entre entidades que forman una red.

Tensión: Es la diferencia de potencial eléctrico que tiene que existir entre los bornes de conexión o entre dos partes activas de una instalación, para que la corriente eléctrica circule por dicha instalación.

Introducción

El deporte es quizás una de las actividades más antiguas de todas las historias y que hasta en nuestros días hoy perdura, cada uno de estos deportes necesitan escenarios diferentes, pero hoy en día los polideportivos permiten la práctica de varios deportes en un solo escenario, por esto los tableros deportivos o marcadores deben contar con diferentes opciones y funcionalidades para permitir la visualización de los acontecimientos durante el desarrollo del evento deportivo.

Los tableros deportivos hoy en día cuentan con diversas características que permiten que se adapten a cualquier escenario deportivo lo cual nos beneficia en gran medida ya que el polideportivo de la institución tiene un punto específico para la visualización de los acontecimientos por parte del público.

Durante varios eventos deportivos se ha visto el disgusto de las personas porque creen que se ha alterado el registro que se le está llevando al evento deportivo, por tal motivo se implementará un tablero deportivo con el cual se pretende que todo el público tenga acceso en cualquier momento al registro que se le está llevando y evitar así disgustos y poder desarrollar el evento de forma transparente y que brinde a toda la comunidad seguridad tranquilidad con el registro y así disfrutar del evento.

1.Planteamiento del problema

1.1 Descripción

En la institución no se cuenta con un tablero en los escenarios deportivos donde las personas puedan visualizar los resultados de forma clara y en tiempo real.

1.2 Formulación del problema

¿Se puede fabricar un tablero con las especificaciones que se requieren para llevar a cabo un evento deportivo de forma adecuada?

2. Justificación

Se va realizar un tablero deportivo matricial donde nos permita visualizar la información pertinente a lo ocurrido durante el desarrollo de los eventos deportivos del Pascual bravo que se desarrollan, tales como microfútbol, basquetbol o voleibol y así brindar una mayor confiabilidad y transparencia de la información llevada a cabo en el desarrollo de los mismos.

3. Objetivos

3.1 Objetivos generales

Fabricar un tablero deportivo multifuncional controlado por medio de un control manual RF.

3.2 Objetivos específicos

Diseñar la programación de un tablero deportivo multifuncional para la visualización de los ítems de tiempo, marcadores y set teniendo en cuenta el evento deportivo, además de un control remoto donde se pueda controlar desde este.

Acoplar todos los elementos necesarios para que se pueda ejecutar bien cada una de las acciones realizadas en la programación, para luego integrarla con el control remoto RF.

Inspeccionar y cerciorarse del funcionamiento de todos los componentes, programación y del control remoto que se implementaron para evitar mal funcionamiento del tablero y poder así corregir estos percances y poder tener el tablero óptimo.

5. Marco teórico

“Cuando estos escenarios deportivos cuentan con buenas opciones de accesibilidad y seguridad de la información que transcurre, se propicia más uso” (Sierra, 2009).

El tablero deportivo matricial tiene como finalidad el permitir la visualización clara de la información más importante que ocurre durante el desarrollo de cualquier evento deportivo que se lleve a cabo, para esto internamente se llevan a cabo miles de operaciones sistemáticas que son programadas por el usuario según los requerimientos que este tenga, tales conjunto de operaciones son llamadas algoritmos, que se encuentran organizadas de manera lógica y ordenada para poder que estas puedan ser interpretadas, sincronizadas y ejecutadas por los diferentes componentes electrónicos a interactuar.

5.1 Microcontrolador PIC 16F887

Los PIC16F887 forman una subfamilia de micro controladores PIC (Peripheral Interface Controller) de gama media. Tenga en cuenta el PIC16F887. Este (ejecución de la instrucción de 200 nanosegundos) potente (sólo 35 operaciones de palabras individuales) fáciles de programa basado en FLASH CMOS de 8 bits .. (Microchip, 1999, pág. 102).

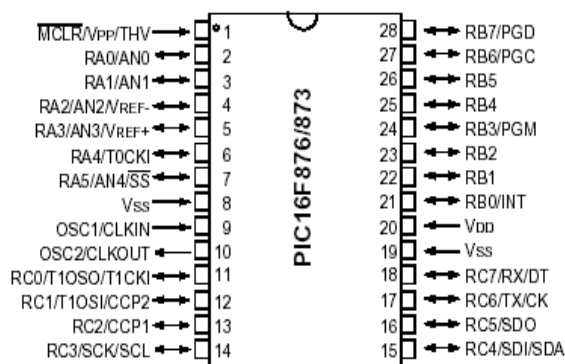


Figura 1 Diagrama del PIC 16f887

<http://ww1.microchip.com/downloads/en/DeviceDoc/41291D.pdf>

En la figura 1 se observa el Microcontrolador físicamente, en el cual están distribuidos todas las funciones que puede realizar cada pin del microcontrolador.

Los microcontroladores son circuitos integrados capaces de llevar uno o varios procesos lógicos, los cuales son acciones programadas de forma segmentada y que siempre se encuentran de manera lógica y ordenada, dicha programación está sujeta a un lenguaje ensamblador que será implementada por el usuario.

Para la ejecución de un programa es necesario contar con una segmentación de algoritmos bien estructurados para que estos sean interpretados de forma correcta por el microcontrolador y sean ejecutados ciclo por ciclo en el lenguaje máquina, para esto el microcontrolador PIC 16f876 cuenta con características como: 28 pines cada uno de 8 bits a 200 nanosegundos que permiten mayor velocidad en ejecución de órdenes, en el momento de programación cuenta con 35 instrucciones, tienen memoria flash; esto quiere decir que podemos programar y borrarla las veces que nosotros deseemos, además trae consigo incorporado un convertor analógico a digital de 10 bits, CPU de arquitectura RISC (Reduced Instruction Set Computer), para llevar un orden en las secuencias de cualquier programa cuenta con una frecuencia de reloj de hasta 20MHz (ciclo de instrucción de 200ns), Rango de voltaje de operación desde 2,0V a 5,5V y una corriente de salida de 25mA contiene un circuito supervisor de instrucciones como lo es el Watchdog Timer o Perro Guardián, 2 módulos de captura/comparación/PWM, Puerto Paralelo Esclavo de 8

bits (PSP), Puerto Serie Síncrono (SSP) con SPI. Patillaje compatible con PIC16C73B/74B/76/77 además de una memoria de datos de 368 bytes. -Memoria EEPROM de datos de 256 bytes.

5.2 SAA1064

El SAA1064 es un controlador LED o de pantalla, el circuito integrado este hecho para ser compatible con I2L. El circuito está diseñado especialmente para conducir cuatro display LED de 7-segmento incluyendo el punto decimal por medio de multiplexor entre dos pares de dígitos. (Philips, Febrero 1991).

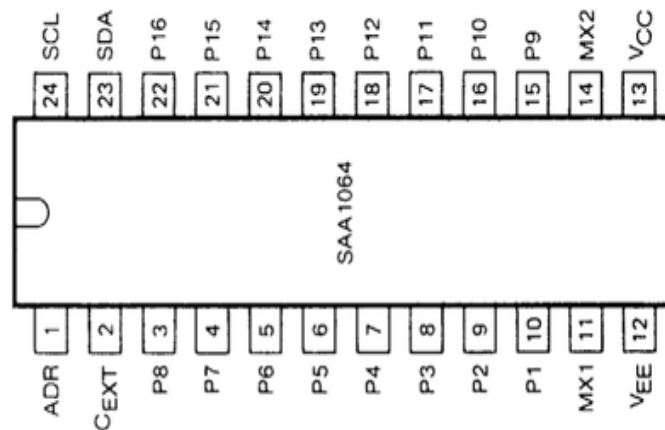


Figura 2 Diagrama del integrado SAA1064
<http://tronixstuff.com/wp-content/uploads/2011/07/nxp-saa1064.pdf>

El SAA1064 es un componente que nos permite controlar 4 display solamente de 7 segmentos el cual consume solo 21 mA, este recibe la información del microcontrolador y este se encarga de distribuir las ordenes a los demás componentes para que sea mostrada en el display correspondiente y se visualice dicha información, como se manejan varios leds encendidos en un mismo tiempo, el microcontrolador solo no es lo suficientemente capaz de dar voltaje corriente a los led ,se debe utilizar el SAA1064 para mantener el mismo brillo y lucidez de estos, para que

suceda esto el integrado cuenta con características especiales donde sobresalen el voltaje de alimentación que es de 4.5 como mínimo hasta 15 Voltios máximo. Corriente que va desde los 7 hasta los 14 mA y además es de tecnología I2c es decir un estándar que facilita la comunicación entre microcontroladores, memorias y otros dispositivos, sólo requieren de dos líneas de señal (SCL (Línea por donde los pulsos del reloj sincronizan el sistema) y SDA (Línea por donde la información pasa a los diferentes dispositivos)) y un común o masa.

5.3 ULN2803A

Los dispositivos ULN2803A son diseñados específicamente para la amplificación de voltaje y corriente, conformado internamente por 7 transistores NPN Darlington, los cuales los hacen muy requeridos para trabajos donde se necesite una alta eficiencia de Corriente y voltaje, cada entrada de este tiene un diodo Zener y una resistencia en serie para controlar la corriente de entrada a un límite seguro, teniendo así protección incluida (Smith, 2005).

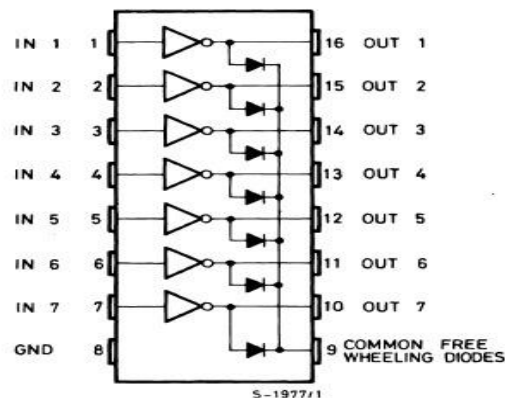


Figura 3 Diagrama del ULN2803A
<http://www.ti.com/lit/ds/symlink/uln2003a.pdf>

Debido a la pequeña potencia que suministra el microcontrolador, sólo podemos hacer funcionar debidamente los leds, los cuales serán controlados por el integrado SAA1064. pero los demás componentes como lo son buzzer, leds sencillos y demás elementos no van a contar con la suficiente potencia para que estos puedan realizar sus procesos normales, por este motivo es

necesario amplificar el voltaje en las salidas del microcontrolador, empleando el integrado ULN2803A, conformado por conjuntos de transistores Darlington de alta tensión y alta corriente, cada uno consta de siete pares NPN Darlington que cuentan con salidas de alto voltaje con diodos de cátodo común para la conmutación de cargas inductivas además de la protección de sobretensiones. Los pares Darlington pueden conectarse en paralelo para una mayor capacidad de corriente, ya según se trabaje con TTL o CMOS.

5.4 Display Matricial o Matriz de leds

Las matrices de leds son dispositivos que se utilizan para la visualización de información más compleja tales como mensajes, horas, símbolos, etc. Cada matriz está constituida por varios puntos LEDS que, al activarse, es decir, cuando circula una corriente a través suyo, se ilumina. (Viejo, 2005, pág. 139).

Los leds totalmente normales solo que se encuentran organizados en forma de matriz, solo que para poder formar caracteres o letras debemos multiplexarlos, esto para que se conecten unos a otros.

Ánodo común (AC): Es cuando los diodos se unen por el lado positivo.

Cátodo común (KC): Es cuando los diodos se unen por el lado negativo.

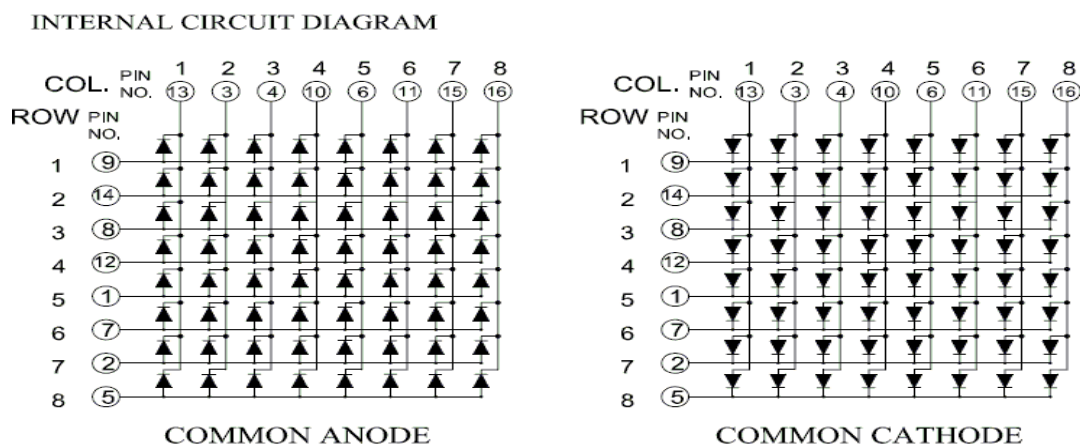


Figura 4 : Diagrama de una matriz de leds 1588AS 8x8

http://www.prototiposelectronicos.com/2012/04/curso-pic16f88-compiler-c-ccs-2_14.html

6.Resultado

Debido a que la Institución no cuenta con un tablero deportivo que muestre la información más relevante de los diferentes eventos deportivos, se estaba obteniendo la información de los marcadores de forma manual, donde se registraban los marcadores en una hoja de datos y a la vez llevar el tiempo con un cronometro o celular, en ocasiones estos dos registros eran llevadas por dos personas diferentes.



Figura 5 Recolección de información manual de los eventos deportivos
<http://mx.depositphotos.com/10563142/stock-photo-business-woman-taking-note-on.html>.

Esta circunstancia permitía que en ocasiones se confundiera y alterara la veracidad de la información suministrada, o en otros casos se dañaban las hojas de datos que contenían dicha información haciéndola poca confiable para los espectadores, además daba mala imagen cuando los equipos de la institución se fogueaban con equipos de otras partes.



Figura 6 Perdida de la información recolectada
http://www.siscorp.com.co/gestion_fisicos.phpque

El proyecto de construcción del tablero presento problemas en el diseño ya que alguno de los componentes iniciales que se implementarían no se adquirirían en Colombia, debido a que en la gran mayoría de países fabricantes de estos componentes hay una constante renovación de elementos que son cada vez más pequeños, pero de mayor complejidad, esto hace que se frene la fabricación de elementos obsoletos, y en caso de poder conseguirlos se tendría que adquirir por un costo muy elevado, por tal motivo se vio en la tarea de buscar componentes similares en cuanto a características y capacidad de trabajo para luego consultar la presencia de estos en Colombia y así saber con qué elementos se pueden trabajar y realizar un nuevo diseño sin que afecte la funcionalidad de los demás componentes; componentes que son difíciles de encontrar como integrado TM1637 que es el controlador de varios display en simultaneo y también el manejo de controles inalámbricos.

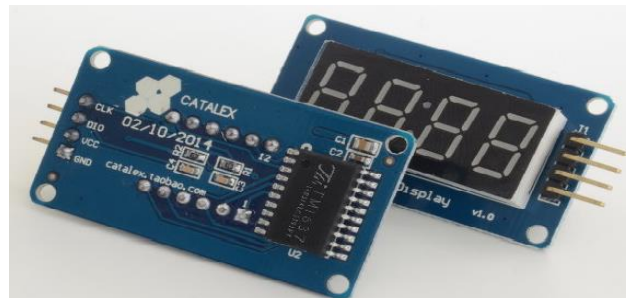


Figura 7 Integrado TM1637

<https://polaridad.es/tm1637-driver-display-led-7-segmentos-teclado/>

y otros cuyo valor son muy elevados como el de la figura 8.

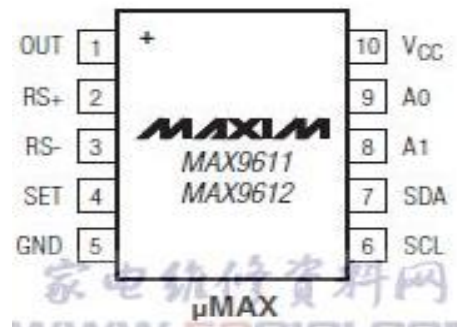


Figura 8 Integrado Max9611

<http://www.520101.com/html/circuitry/113106412.html>

Además, se presentó inconvenientes en la programación del lenguaje C++ ya que se requieren conocimientos para desarrollar dicho programa que supla las necesidades, por lo cual se utilizó Pseint, ya que este nos permite hacer un Pseudocódigo con instrucciones fáciles de reconocer e implementar pero que a la misma vez nos permite pasar este pseudocódigo a líneas de código en lenguaje C++ y así podemos ir identificando el uso de cada una de las instrucciones y comandos del lenguaje C, afianzando el conocimiento.

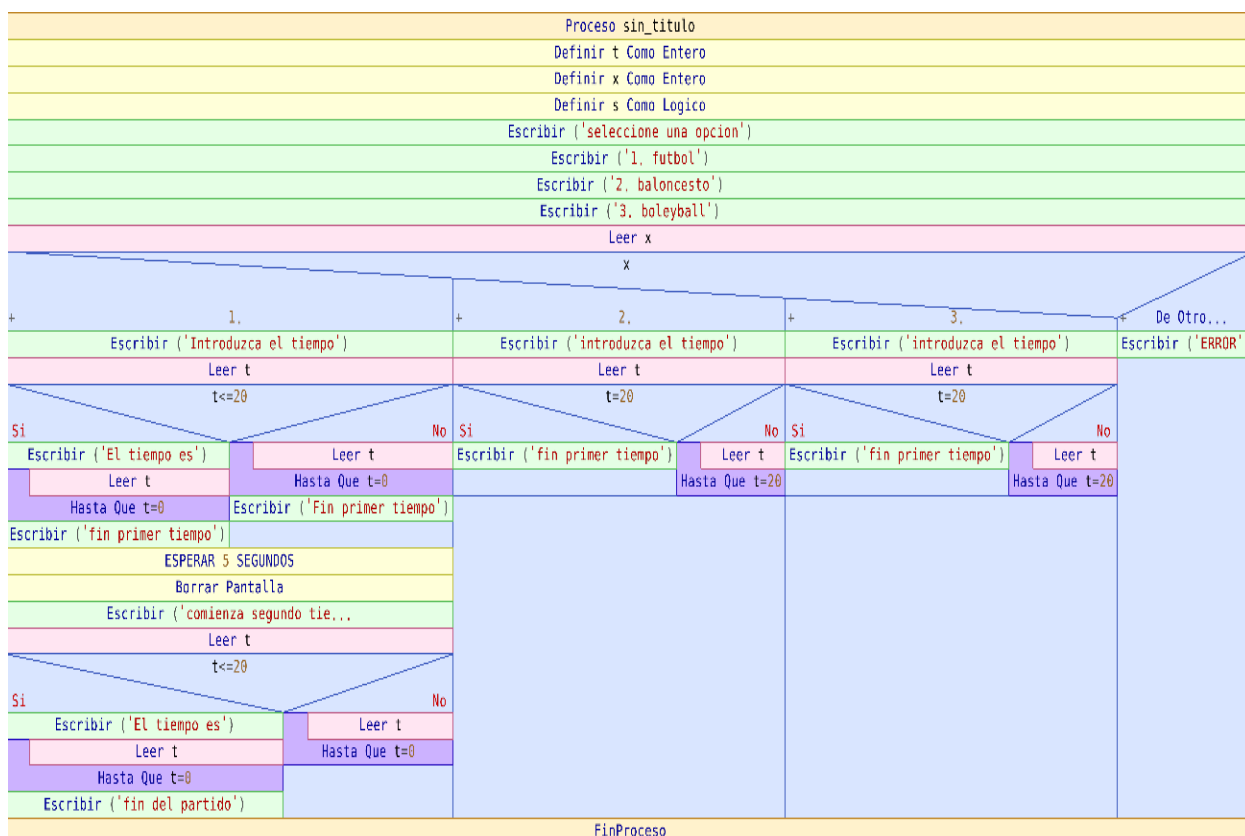


Figura 9 Esquema del Pseudocódigo
[Diseño propio](#)

Otro de los inconvenientes que se presentó fue el manejo de Proteus 8.0 Profesional ya que por primera vez se manejaba esta versión, debido a que para la simulación de todo el circuito se necesitaban muchas más herramientas que solo esta versión las contenía, por lo cual se debió ver primero tutoriales y practicar en un curso on line que Lab Center, fabricantes y programadores de este software brinda a las personas que por primera vez utilizan esta versión, finalizando con

la realización de un sencillo circuito, comenzándolo totalmente desde cero hasta la simulación y fabricación de la PCB.

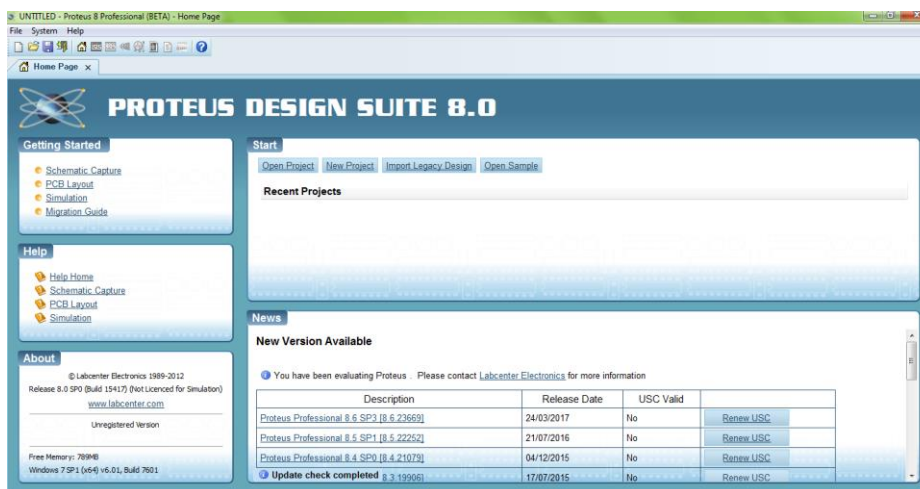


Figura 10 Software de Proteus 8.0 Profesional.

<https://www.labcenter.com/>

Para evitar confusión y dar buena imagen de los escenarios deportivos de la institución hacia los espectadores, se implementó un tablero electrónico con el cual nos suplirá estas necesidades, para esto se requieren varios procedimientos en la fabricación del tablero deportivo comenzando con:

6.1 Programación del PIC

Para la programación de un microcontrolador PIC se requiere elaborar un Pseudocódigo bien ordenado y estructurado, el cual nos permitirá ejecutar las acciones programadas de manera efectiva y lógica, para dicha programación se debe de tener en cuenta dos aspectos muy importantes; saber que Microcontrolador se utilizará y que lenguaje de programación se manejará, en el presente caso se manejara el PIC 16f887 y se utilizó el lenguaje C++ .

6.1.1 Programación del pseudocódigo. Para realizar la programación del PIC primero realizamos nuestro programa en C++, para crear un proyecto nuevo realizaremos los siguientes pasos.

Abrimos el software de C++.

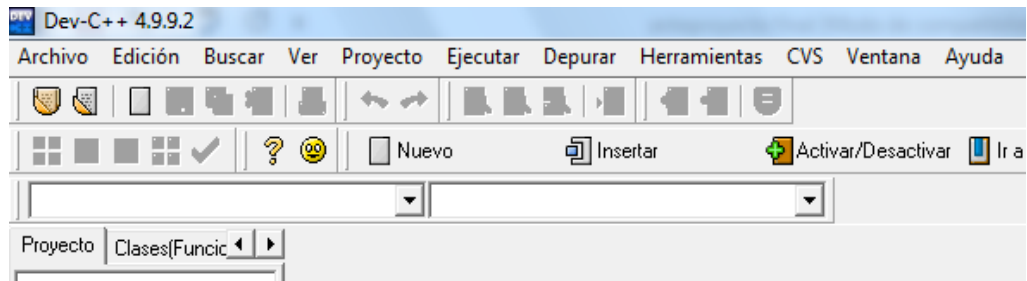


Figura 11 Ventana inicial de C++

<https://www.reddit.com/r/devcpp/>

Como vemos en la anterior figura, se nos muestra todas las opciones herramientas que podemos utilizar, luego procedemos a dar click en la opción archivo y nos mostrara lo siguiente

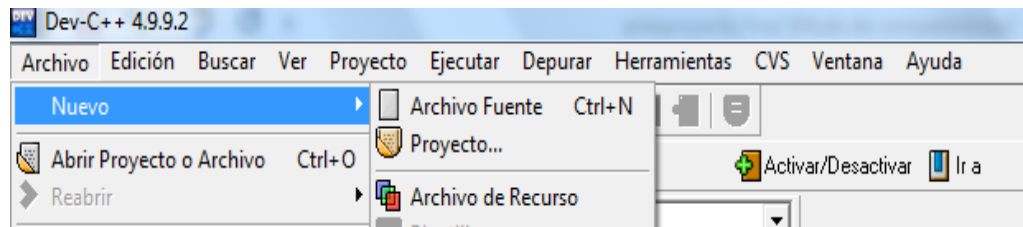



Figura 12 Crear un nuevo proyecto en c++

<https://www.reddit.com/r/devcpp/>

Le damos click en Nuevo, luego escogemos la opción  Proyecto... y nos va aparecer un cuadro de diálogos donde nos pide lo siguiente:

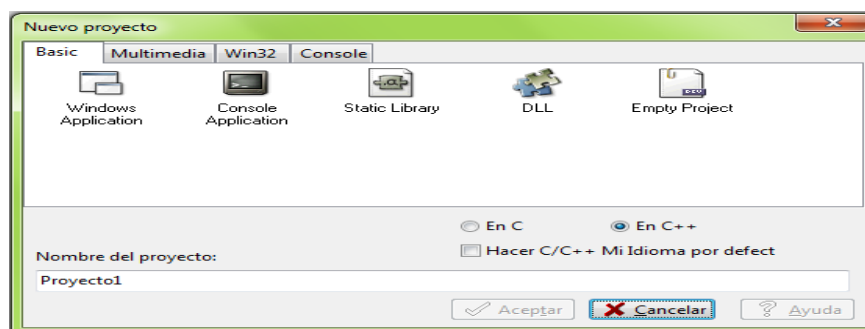

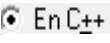
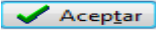


Figura 13 Selección del lenguaje y nombre del nuevo proyecto

<https://www.reddit.com/r/devcpp/>

En los recuadros de encima seleccionaremos la opción  , luego en la opción “Nombre del proyecto”, colocamos el nombre que deseamos, en la parte inferior derecha se visualizan 3 opciones con recuadros seleccionables, en este rellenaremos el círculo que tiene la opción  , esto quiere decir que trabajaremos en esta clase de lenguaje de programación, una vez realizado los pasos anteriores, la opción  se nos va a habilitar, para luego darle click, luego de darle aceptar se nos pide elegir la ubicación del archivo donde lo queremos guardar, y listo ya podemos comenzar a crear nuestra programación.

Como vemos en la figura 11 en la barra de herramientas solo están habilitadas las opciones de crear un nuevo proyecto o abrir uno existente, las demás opciones se nos muestran en color gris, una vez se haya realizado los pasos antes mencionados ya podremos utilizar el resto de herramientas.

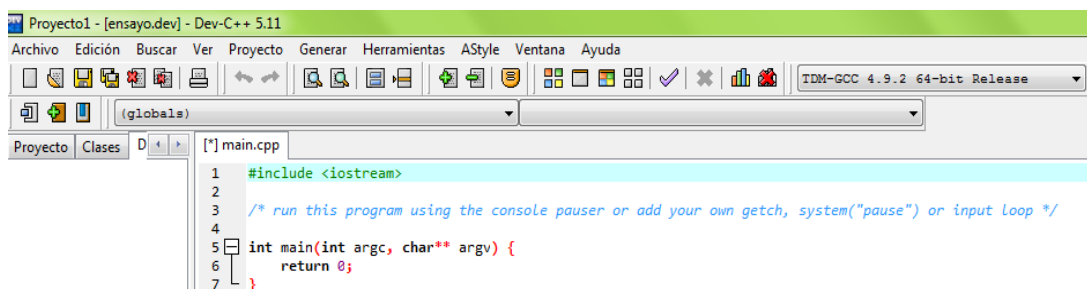



Figura 14 Programa activado para ser utilizado
<https://www.reddit.com/r/devcpp/>

Una vez se finalice todo el programa debemos ejecutar y compilar este,  para que se realice la verificación de todos los procesos lógicos programados y así corregir o modificar dichos inconvenientes.

6.1.2 Compilación del programa. Un compilador convierte el lenguaje de alto nivel a instrucciones en código máquina; un cross-compiler es un compilador que funciona en un procesador (normalmente en un PC) diferente al procesador objeto. El código máquina puede ser cargado del PC al sistema PIC mediante el ICD2 o mediante cualquier programador y puede ser

depurado (puntos de ruptura, paso a paso etc.) desde el entorno de trabajo del PC (Breijo, 2009, pág. 32).



Figura 15 logo del PIC C COMPILER
<http://www.ccsinfo.com/content.php?page=compilers>

Una vez se tenga el programa en C++ finalizado, se requiere utilizar el software PIC C COMPILER de la versión PCW CCS C, el cual nos permite convertir y codificar el programa en lenguaje C++ a un lenguaje que el Microcontrolador PIC interprete, conocido como lenguaje ensamblador o lenguaje máquina, dicho compilador nos genera ficheros en formato Intel-hexadecimal, que es el necesario para programar (utilizando un programador de PIC) un Microcontrolador de 6, 8, 18 o 40 pines. El compilador de C que vamos a utilizar es el PCW de la versión CCS C y que, al mismo tiempo, este lo incorporaremos en un entorno de desarrollo integrado (IDE) que nos va a permite la edición hasta la compilación pasando por la depuración de errores. Para realizar la compilación del lenguaje C++ haremos lo siguiente. Abrimos el compilador PCW, nos va a abrir la siguiente pantalla.

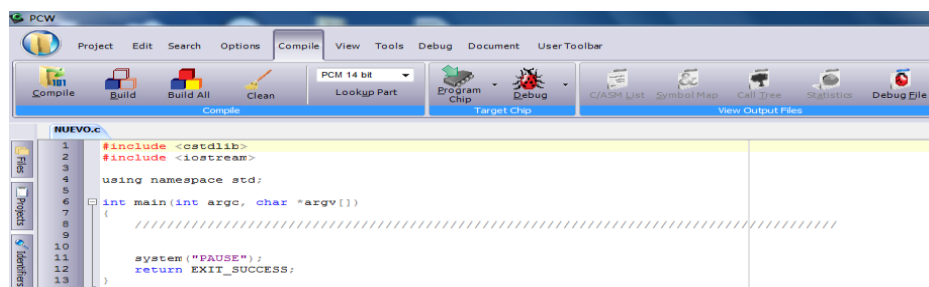


Figura 16 Ventana inicial del PIC CCS PCW
<http://www.ccsinfo.com/content.php?page=compilers>

Luego en el logo de “Carpeta” le damos click y nos va a mostrar varias opciones.

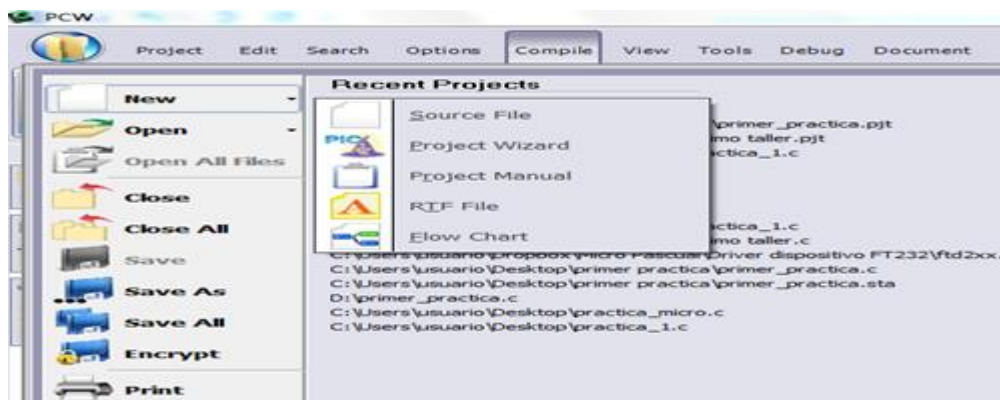


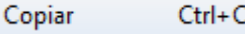
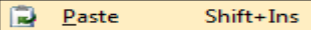


Figura 17 Nuevo proyecto en el PCW <http://www.ccsinfo.com/content.php?page=compilers>

Como vamos a crear un nuevo proyecto seleccionamos la opción  al lado se nos despliega otras opciones y seleccionamos  luego de seleccionarla se nos pregunta en qué lugar se guardará la compilación, ya elegida la ruta para guardarla, seleccionamos todo el programa que se encuentra en C++ enseguida damos click derecho y seleccionamos la opción  **Ctrl+C**, luego vamos a la plataforma de PCW en la hoja nueva del archivo que acabamos de crear y le damos  **Shift+Ins** de inmediato se nos copia todo el programa.

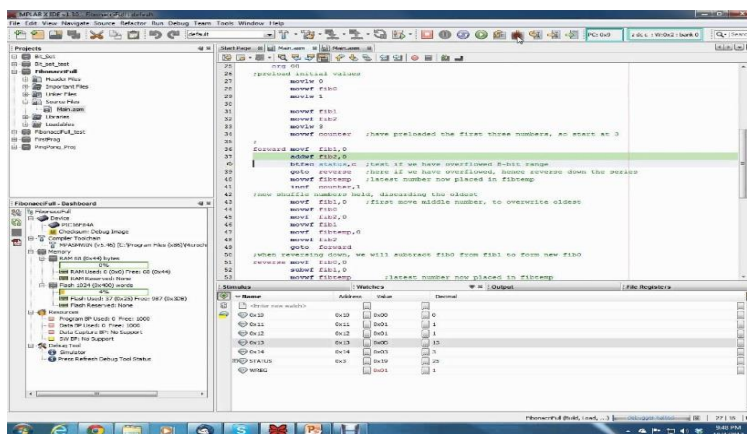




Figura 18 Importación del programa en C++ a PCW <http://www.ccsinfo.com/content.php?page=compilers>

Luego damos click en la opción de  para que verifique que el programa no tiene errores lógicos, cuando ya no se tengan errores daremos click en  y saldrá un cuadro de diálogos diagnosticando el programa donde nos dirá la cantidad de ROM y RAM que el programa consume en el momento de su funcionamiento.

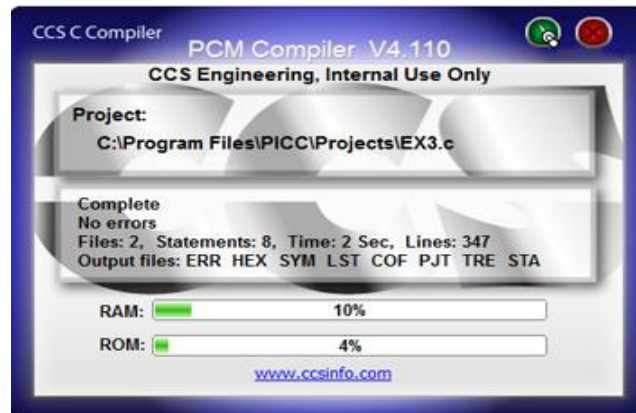


Figura 19 Indicadores de compilación del programa
<http://www.ccsinfo.com/content.php?page=compilers>

Así se culminaría con la fabricación, corrección y compilación del programa para ya luego almacenar la información dentro del PIC.

6.1.3 Programador de microcontroladores PIC. Para el cual se utilizó el programador Pickit 2, ya que este trae consigo el software de instalación, lo cual nos permite más facilidad en el trabajo.



Figura 20 Programador Pickit 2
<http://www.microchip.com/DevelopmentTools/ProductDetails.aspx?PartNO=pg164120>

Pasos antes de grabar el Microcontrolador:

- Selección del microcontrolador.
- Selección / Instalación del compilador / ensamblador.
- Selección / Instalación del software programador.
- Desarrollo del software y programación, los programas para realizar la programación pueden ser: MPLAB o DevC++ o también conocido como C++.

La ventaja de trabajar con el Pickit 2, es que cuenta con características especiales tales como:

- Programador y Debugger Express compatible con MPLAB.
- Soporta las familias PIC10, PIC12, PIC16, PIC18, PIC24, dsPIC30, dsPIC33 y PIC32.
- Incluye Easy Jack y conector RJ11 para Target Application Board.
- Zócalo ZIF para programación en board de hasta 40 pines.
- Firmware actualizable con cada nueva versión del MPLAB.
- No requiere alimentación externa para micros de 3.3 a 5.0 v.
- Incluye analizador lógico de 3 canales.
- Compatible con USB 1.1 y 2.0.

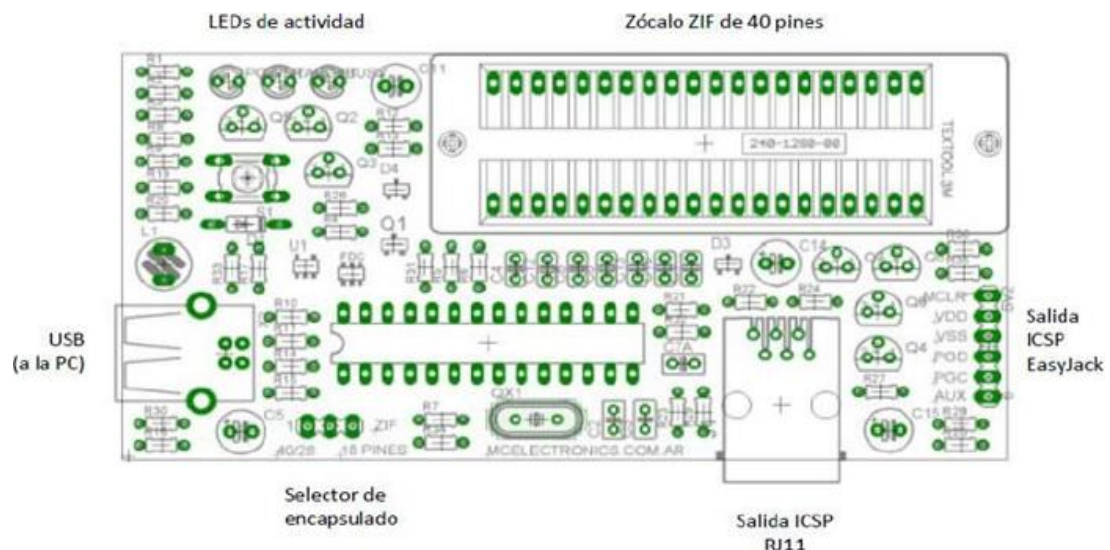


Figura 21 Diagrama interno del Pickit 2

<http://www.microchip.com/DevelopmentTools/ProductDetails.aspx?PartNO=pg164120>

Este programador trae consigo el software para ser instalado y ejecutado, con el cual se procederá a programar el PIC 16f887, para esto abrimos el software y podremos visualizar lo siguiente.

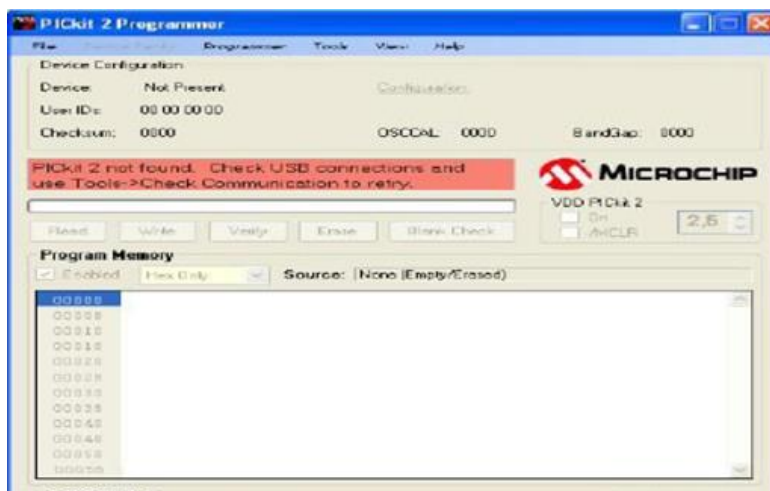


Figura 22 Instalación del software del pickit 2
<http://www.microchip.com/DevelopmentTools/ProductDetails.aspx?PartNO=pg164120>

En este caso se nos indicara que PICKIT 2 not found. Check USB connections and use Tools->Check Communication to retry., para lo cual vamos a conectar el MCE PDX USB(Pickit 2) a la PC a través del puerto USB para comenzar a trabajar. Una vez conectado a la PC, abrimos nuevamente el programa “PICKIT2 v2.61” (Programas->Microchip-> PICKIT2 v2.61).

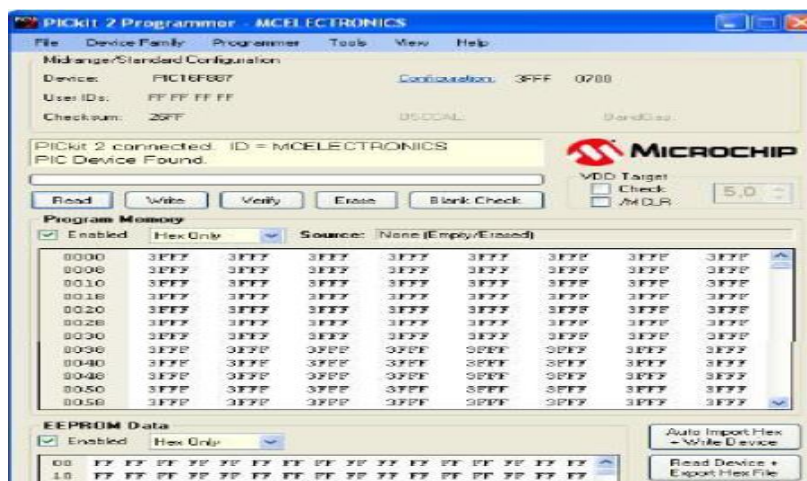


Figura 23 Reconocimiento y verificación del programa del pickit 2
<http://www.microchip.com/DevelopmentTools/ProductDetails.aspx?PartNO=pg164120>

Como se ve en la figura anterior ya ha sido reconocido el Pickit 2 por medio del software, luego introducimos el PIC y de inmediato el software reconoce con cual se trabajará.

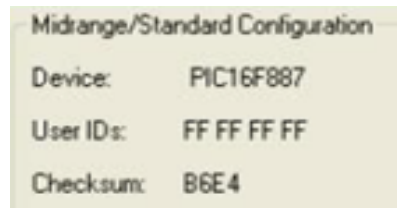


Figura 24 Reconocimiento del PIC 16f887

<http://www.microchip.com/DevelopmentTools/ProductDetails.aspx?PartNO=pg164120>

Ya se verificaron tanto el funcionamiento del software como reconocimiento del PIC, por lo cual ya está todo listo para la programación, se realizarán los siguientes pasos. Se dirigirá a File> Import Hex (recordar que la carpeta Hex o Hexadecimal es el resultado de la compilación con el PIC C de CCS C), y seleccionaremos la carpeta creada por el compilador que se encuentra en lenguaje ensamblador o máquina.

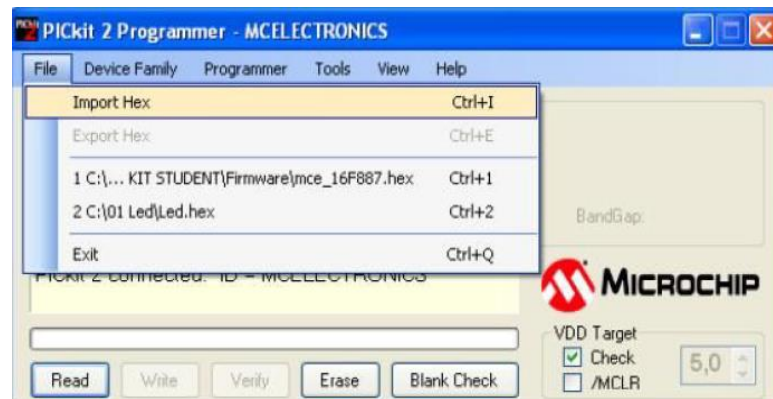


Figura 25 Importación del programa compilado en PIC CCS PWC al Pickit 2

<http://www.microchip.com/DevelopmentTools/ProductDetails.aspx?PartNO=pg164120>

Una vez se halla cargado totalmente la carpeta daremos click en la opción “Read” y en simultaneo se nos habilita la opción de “Write” para luego dar click en esta, luego de cargar una barra de estado debe aparecer la leyenda “Programación exitosa” una vez concluida la transmisión de la información.

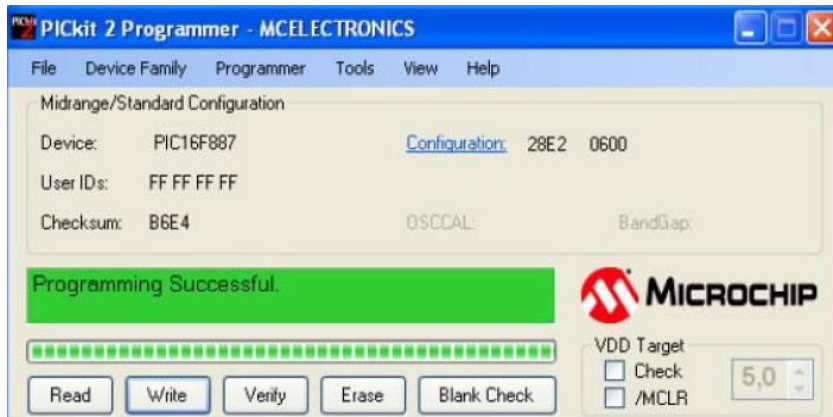



Figura 26 Programación del microcontrolador

<http://www.microchip.com/DevelopmentTools/ProductDetails.aspx?PartNO=pg164120>

6.2 Diseño del circuito Electrónico.

En este paso se diseña el esquema eléctrico del circuito que se desea realizar. Para ello se va a utilizar el programa PROTEUS concretamente la parte de ISIS que está más enfocada a la parte eléctrica y electrónica del circuito. Para comenzar a simular el circuito comenzaremos abriendo el programa de  PROTEUS ISIS cuando lo abramos nos aparecerá lo siguiente.

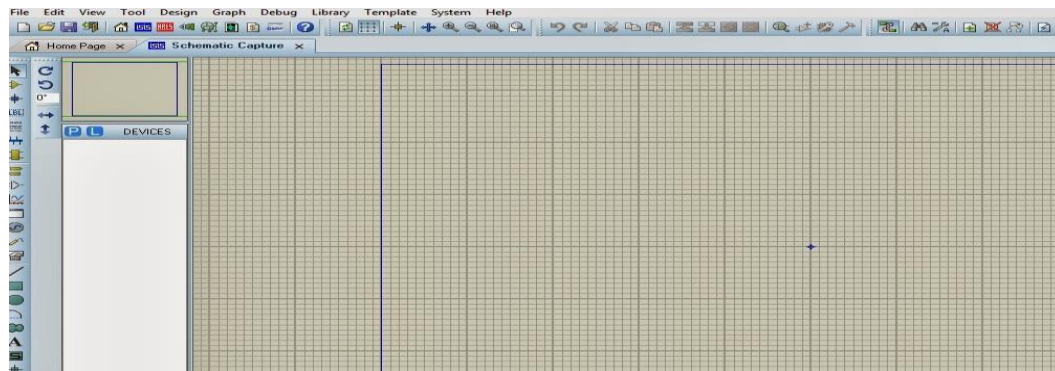


Figura 27 Ventana principal de Proteus

<https://www.labcenter.com/tutorials/>

Para crear nuestro nuevo circuito seguiremos los siguientes pasos, iremos a “File” le daremos click y se nos despliegan varias opciones, de las cuales elegiremos “New Project”.

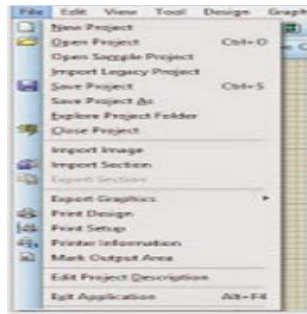


Figura 28 Creación del nuevo proyecto en PROTEUS
<https://www.labcenter.com/tutorials/>

Dándole click pedirá que le coloquemos un nombre para identificarlo seguido de la ubicación donde se desea guardar este proyecto nuevo, una vez completada la información nos aparecerá nuestra área de trabajo.

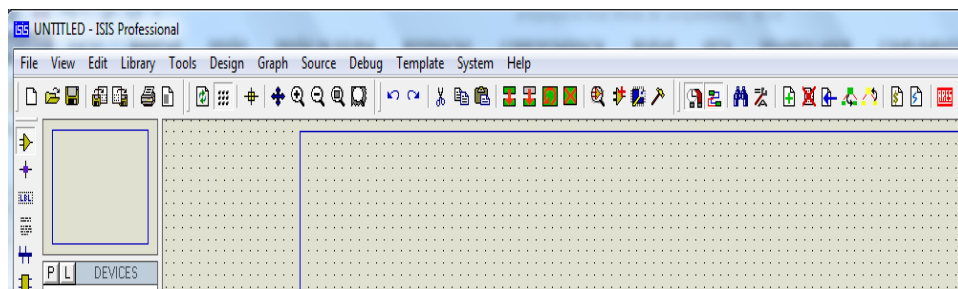


Figura 29 Activación de las herramientas para comenzar el diseño
<https://www.labcenter.com/tutorials/>

La parte que más utilizaremos es esta, debido a que es donde hallamos todos los componentes a utilizar para el desarrollo, tales como Display, PIC16f887, resistencias y demás integrados.

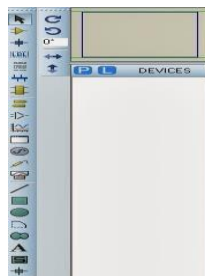


Figura 30 Columna para la selección de componentes electrónicos
<https://www.labcenter.com/tutorials/>

Una vez seleccionado y unido todos los componentes seleccionados, se tendría el siguiente resultado.

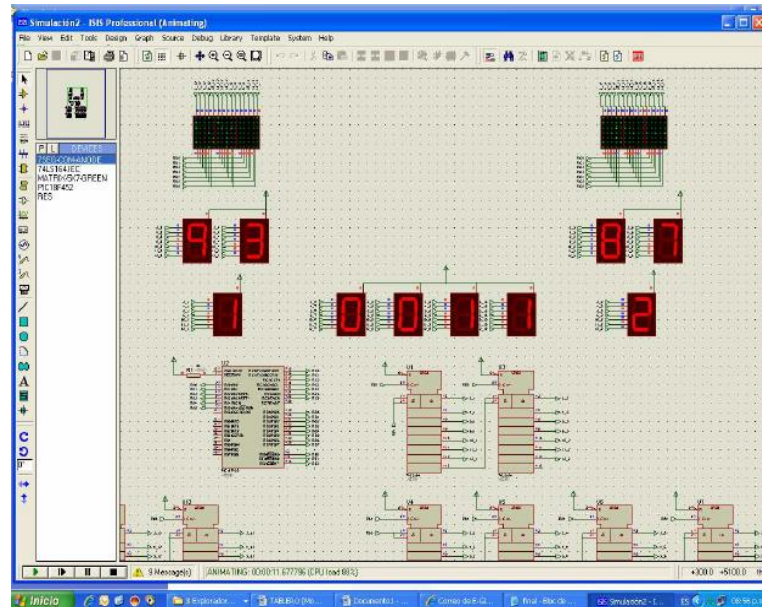


Figura 31 Simulación del circuito completo

<https://www.labcenter.com/tutorials/>

Ya verificada las conexiones de todos los componentes realizaremos la simulación del mismo, este paso nos permite observar el comportamiento de los componentes y asegurar de que todas las conexiones estén bien y que ningún componente tiene invertida la polaridad. En esta ocasión al ser un circuito muy completo, con bastantes componentes y con una larga programación por lo cual es muy lenta en ejecutar al momento de ejecutarlo.

6.3 Montaje del circuito. Una vez se halla verificado su correcto funcionamiento en el simulador se procede a hacer el montaje en la board, se montan todos los componentes de la manera que están en el esquema, se procede a hacer el cableado de todo el circuito y se hace un check list para garantizar que en los puntos más importantes del circuito si se esté entregando el suficiente voltaje para que los demás componentes funcionen de manera óptima.

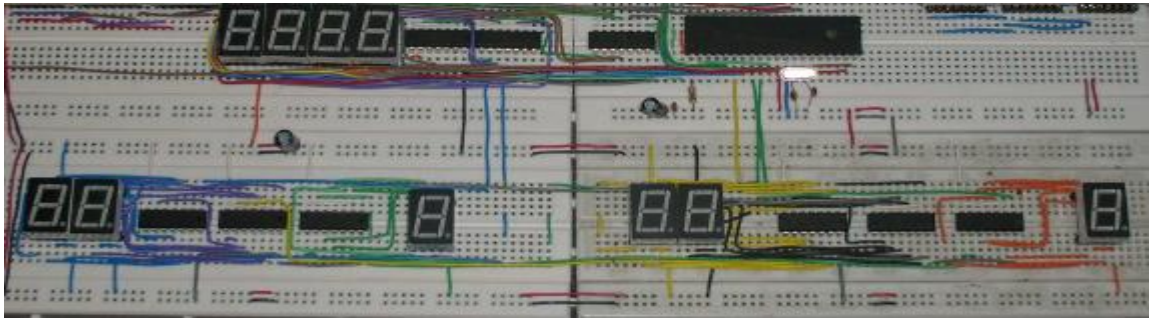


Figura 32 Montaje de los componentes electrónicos en Boards
[Diseño Propio](#)

6.4 Diseño de la PCB. También conocida como tarjetas cableadas impresas, es decir un a tarjeta que internamente tiene las conexiones necesarias para comunicarse unos elementos con otros y así ejecutar todos los procesos requeridos. El diseño de la PCB se realizó con ARES (un programa del software PROTEUS). Para hacer el diseño se tienen que asignar todas las huellas (todos los componentes tienen plantillas o también conocidas como footprints, según la cantidad de pines de cada elemento) a los componentes del esquema en el programa ISIS para que cuando lo pasemos a ARES cada componente tenga su huella, esto sirve para saber la mejor distribución y así evitar cortocircuitos y ahorrar algo de espacio.

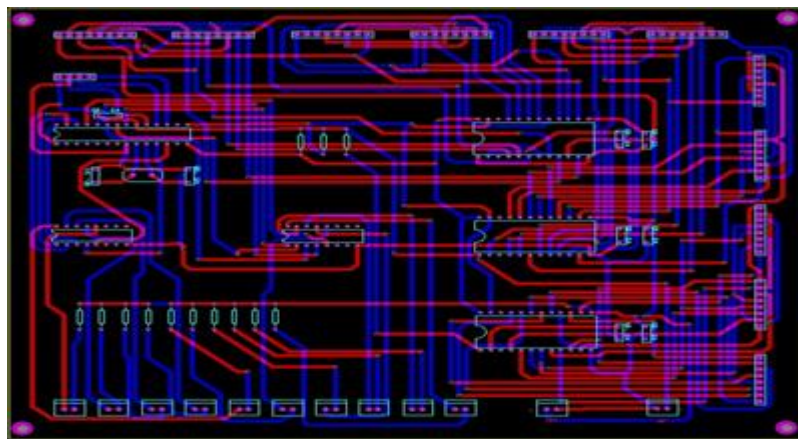


Figura 33 Diseño de la PCB
<https://www.labcenter.com/tutorials/>

Como se ve en la figura se realizó el diseño de la PCB en dos caras para así ahorra algo de espacio.

Una vez ensamblado todos los componentes en sus respectivos lugares, se espera tener como resultado un tablero similar a este.



Figura 36 Tablero deportivo completo
<http://www.zeit.com.mx/tableros-deportivos-futbol.htm/>

7. Conclusiones

Se realizó el diseño e implementación del tablero con información detallada de tal forma que el espectador posee datos oportunos y reales del marcador que tiene cada uno de los equipos.

Se desarrolló un sistema de comunicación por medio de un control de tal forma que el operario pueda ingresar la información que sea requerida en el tablero electrónico.

Se logró el desarrollo del tablero electrónico, con cada una de las especificaciones requeridas.

8. Bibliografía

<http://biblioteca.usbbog.edu.co:8080/Biblioteca/BDigital/42134.pdf>

<http://elektronikadonbosco.blogspot.com.co/2013/03/marcador-de-polideportivo-controlado.html>

<http://www.comunidadelectronicos.com/>

<http://www.zeit.com.mx/tableros-deportivos-zeit.html>

<http://www.areatecnologia.com/electronica/electronica-digital.html>

<http://electronica-electronics.com/Digital/Electronica-digital.html>

http://www.ecured.cu/Electr%C3%B3nica_digital

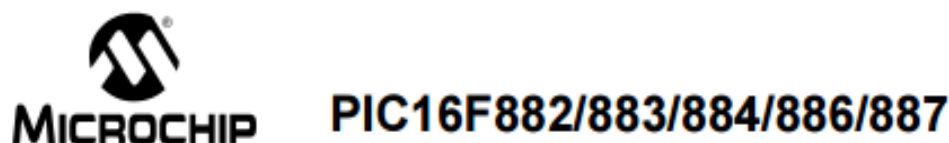
<http://www.forosdeelectronica.com/tutoriales/resistencia.htm>

9.Referencias bibliográficas

- Breijo, E. G. (2009). *Compilador C CCS y simulador proteus para PIC*. Barcelona(Esaña): marcombo ediciones tecnicas.
- Microchip. (1999). *Pic 16f887*. Taiwan: Copyright © 1998-2017 Microchip Technology Inc.
- Philips. (Febrero 1991). *Integrated Circuits*. Philips Semiconductors.
- Sierra, E. R. (2009). *Globalizacion, Deporte y desarrollo*. Medellin: Funambulos Editores.
- Smith, J. R. (2005). *Programación de microcontroladores PIC con MBASIC*. Oxford, Inglaterra: ELSEVIER.
- Viejo, C. B. (2005). *Fundamentos de electronica digital*. Madrid, España: Paraninfo.

10. Anexos

ANEXO A. Características principales del microcontrolador PIC 16f887.



28/40/44-Pin Flash-Based, 8-Bit CMOS Microcontrollers with nanoWatt Technology

High-Performance RISC CPU:

- Only 35 instructions to learn:
 - All single-cycle instructions except branches
- Operating speed:
 - DC – 20 MHz oscillator/clock input
 - DC – 200 ns instruction cycle
- Interrupt capability
- 8-level deep hardware stack
- Direct, Indirect and Relative Addressing modes

Special Microcontroller Features:

- Precision Internal Oscillator:
 - Factory calibrated to $\pm 1\%$
 - Software selectable frequency range of 8 MHz to 31 kHz
 - Software tunable
 - Two-Speed Start-up mode
 - Crystal fail detect for critical applications
 - Clock mode switching during operation for power savings
- Power-Saving Sleep mode
- Wide operating voltage range (2.0V-5.5V)
- Industrial and Extended Temperature range
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Brown-out Reset (BOR) with software control option
- Enhanced low-current Watchdog Timer (WDT) with on-chip oscillator (software selectable nominal 268 seconds with full prescaler) with software enable
- Multiplexed Master Clear with pull-up/input pin
- Programmable code protection
- High Endurance Flash/EEPROM cell:
 - 100,000 write Flash endurance
 - 1,000,000 write EEPROM endurance
 - Flash/Data EEPROM retention: > 40 years
- Program memory Read/Write during run time
- In-Circuit Debugger (on board)

Low-Power Features:

- Standby Current:
 - 50 nA @ 2.0V, typical
- Operating Current:
 - 11 μ A @ 32 kHz, 2.0V, typical
 - 220 μ A @ 4 MHz, 2.0V, typical
- Watchdog Timer Current:
 - 1 μ A @ 2.0V, typical

Peripheral Features:

- 24/35 I/O pins with individual direction control:
 - High current source/sink for direct LED drive
 - Interrupt-on-Change pin
 - Individually programmable weak pull-ups
 - Ultra Low-Power Wake-up (ULPWU)
- Analog Comparator module with:
 - Two analog comparators
 - Programmable on-chip voltage reference (CVREF) module (% of V_{DD})
 - Fixed voltage reference (0.6V)
 - Comparator inputs and outputs externally accessible
 - SR Latch mode
 - External Timer1 Gate (count enable)
- A/D Converter:
 - 10-bit resolution and 11/14 channels
- Timer0: 8-bit timer/counter with 8-bit programmable prescaler
- Enhanced Timer1:
 - 16-bit timer/counter with prescaler
 - External Gate Input mode
 - Dedicated low-power 32 kHz oscillator
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Enhanced Capture, Compare, PWM+ module:
 - 16-bit Capture, max. resolution 12.5 ns
 - Compare, max. resolution 200 ns
 - 10-bit PWM with 1, 2 or 4 output channels, programmable "dead time", max. frequency 20 kHz
 - PWM output steering control
- Capture, Compare, PWM module:
 - 16-bit Capture, max. resolution 12.5 ns
 - 16-bit Compare, max. resolution 200 ns
 - 10-bit PWM, max. frequency 20 kHz
- Enhanced USART module:
 - Supports RS-485, RS-232, and LIN 2.0
 - Auto-Baud Detect
 - Auto-Wake-Up on Start bit
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I²C™ Master and Slave Modes with I²C address mask

