

**ORGANIZADOR DE TURNOS INALÁMBRICO POR RADIOFRECUENCIA**

**FRANCINY ALBERTO CANO QUINCHÍA**

**TECNOLÓGICO PASCUAL BRAVO INSTITUCIÓN UNIVERSITARIA**

**TECNOLOGÍA ELECTRÓNICA**

**MEDELLÍN**

**2011**

**ORGANIZADOR DE TURNOS INALÁMBRICO POR RADIOFRECUENCIA**

**FRANCINY ALBERTO CANO QUINCHÍA**

**Trabajo de grado para optar al título de Tecnólogo en Electrónica**

**Asesor**

**Fabio León Suárez Álvarez.  
Ingeniero Electrónico U. de A.**

**TECNOLÓGICO PASCUAL BRAVO INSTITUCIÓN UNIVERSITARIA**

**TECNOLOGÍA ELECTRÓNICA**

**MEDELLÍN**

**2011**

## CONTENIDO

INTRODUCCIÓN .....	7
1. DESCRIPCIÓN DEL PROBLEMA .....	8
1.1. FORMULACIÓN DEL PROBLEMA.....	8
2. JUSTIFICACIÓN.....	9
3. OBJETIVOS.....	10
3.1. OBJETIVO GENERAL .....	10
3.1 OBJETIVOS ESPECÍFICOS.....	10
4. MARCO TEÓRICO .....	11
4.1. MICROCONTROLADORES .....	11
4.2. DRIVER DE CORRIENTE .....	23
4.3. TRANSMISIÓN Y RECEPCIÓN DE RADIOFRECUENCIA.....	26
4.4. FUENTE DE ALIMENTACIÓN.....	31
5. METODOLOGÍA .....	39
5.1 TIPO DE PROYECTO.....	39
5.2. TÉCNICAS PARA LA RECOLECCIÓN DE INFORMACIÓN .....	39
5.2.1 Fuentes primarias.....	39
5.2.2 Fuentes secundarias.....	39
6. RESULTADOS DEL PROYECTO.....	40
7. CONCLUSIONES .....	53
8. RECOMENDACIONES .....	54
<u>    </u> BIBLIOGRAFÍA.....	55
<u>    </u> CIBERGRAFÍA.....	56
<u>    </u> ANEXOS.....	57

## LISTA DE TABLAS

Tabla 1. Banco de registros PIC 16F877.....	16
Tabla 2. Set de instrucciones y sintaxis de los PIC 16F877 y 16F628A .....	17
Tabla 3. Rangos de frecuencia para la oscilación en base a cristal.....	19
Tabla 4. Descripción de los pines del PIC 16F628A.....	22
Tabla 5. Características eléctricas del ULN 2803 .....	25
Tabla 6. Características del módulo Transmisor y Receptor de Radiofrecuencia..	30
Tabla 7. Referencias de la familia 78xx y corrientes que manejan .....	34
Tabla 8. Características estándar de los modelos 78xx.....	35
Tabla 9. Características eléctricas del regulador de voltaje 7805.....	35
Tabla 10. Especificaciones técnicas para el display de ánodo común.....	38

## LISTA DE FIGURAS

Figura 1. PIC 16F877.....	13
Figura 2. Circuito de reset de un microcontrolador .....	15
Figura 3. Conexión de la memoria de instrucciones y la memoria de datos .....	20
Figura 4. Pines del PIC 16F628A.....	23
Figura 5. Pines del integrado ULN 2803 .....	25
Figura 6. Transmisión y recepción de información en una dirección .....	27
Figura 7. Los efectos del ruido sobre una señal .....	29
Figura 8. Descripción física del Transmisor y Receptor de Radiofrecuencia. ....	31
Figura 9. Transformador de hierro. ....	32
Figura 10. Forma física y simbología eléctrica para un fusible .....	33
Figura 11. Forma física de un condensador .....	34
Figura 12. Display de siete segmentos .....	37
Figura 13. Medidas caja en acrílico. ....	46
Figura 14. Diagrama de bloques de los circuitos del organizador de turnos.....	52

## LISTA DE ANEXOS

Anexo A. Diagrama de flujo etapa de transmisión y visualización de datos.. .....	57
Anexo B. Diagrama de flujo etapa de recepción y visualización de datos.. .....	58
Anexo C. Diagrama de flujo “PRINTCONTADOR” (Función de multiplexación)....	59
Anexo D. Programa en C de transmisión y visualización de datos.....	60
Anexo E. Programa en C de recepción y visualización de datos.....	64
Anexo F. Esquemático fuente de alimentación.....	67
Anexo G. Esquemático alimentación de los ánodos de los display del receptor. ..	68
Anexo H. Esquemático circuito de visualización receptor de datos. ....	69
Anexo I. Esquemático etapa de transmisión y recepción de datos.....	70
Anexo J. Esquemático conexión ULN 2803.....	71
Anexo K. Conexión PIC 16F877 y displays de la etapa transmisora. ....	72
Anexo L. Conexión PIC 16F877 con el teclado 4x4.....	73
Anexo M. Circuito impreso fuente.....	74
Anexo N. Circuito impreso etapa transmisora.....	75
Anexo O. Circuito impreso etapa receptora.....	76
Anexo P. Fotografía proyecto de grado. ....	77
Anexo Q. Fotografía etapa de control en funcionamiento.....	78
Anexo R. Fotografía etapa receptora en funcionamiento.....	79

## INTRODUCCIÓN

Con el nacimiento de la electrónica digital se han creado nuevos mecanismos y dispositivos que plantean a la humanidad cambios organizados en su manera de vivir utilizando elementos de bajo consumo energético y de tamaño reducido, un ejemplo de ello es el organizador de turnos inalámbrico por radiofrecuencia, en el cual se visualiza un número que indica dónde va el correspondiente turno en espera de una atención determinada.

Con base en el microcontrolador PIC, se desarrolló un sistema electrónico de alta precisión de comunicación serial inalámbrico por Radiofrecuencia, que requiere programación en el lenguaje C para permitir la interacción entre el microcontrolador y el usuario. El envío de datos se logrará con la utilización de los recursos del microcontrolador, de los módulos del transmisor y receptor de radiofrecuencia y de los demás elementos que componen este proyecto.

Este proyecto se basa en la investigación y explicación de los microcontroladores como parte esencial para el funcionamiento del organizador de turnos, del fenómeno eléctrico y electrónico de la transmisión y recepción de datos, de las características de los módulos transmisor y receptor de radiofrecuencia. También consta de una breve descripción de los demás elementos que se utilizarán, de sus especificaciones técnicas, además de su importancia y el papel en cada una de las dos etapas del proyecto: la etapa transmisora y la etapa receptora. Finalmente se presentan las conclusiones y recomendaciones, las cuales están basadas en las consideraciones y los ensayos realizados en el organizador de turnos.

El organizador de turnos inalámbrico por radiofrecuencia contribuye sin duda al mejoramiento en la prestación de la atención de usuarios, y se podría comercializar en la industria para beneficio del realizador del proyecto.

## **1. DESCRIPCIÓN DEL PROBLEMA**

Se ha observado en diferentes oportunidades la necesidad de mejorar los organizadores de turnos para pequeñas, medianas y grandes empresas que siempre tienen en sus clientes el punto de partida para desarrollar y ofrecer diferentes servicios y así cumplir con las necesidades básicas para todos sus consumidores.

En la actualidad, muchos de los organizadores de turnos que se instalan requieren gran despliegue de cableado para el envío de datos, lo que genera un mayor costo en su implementación, por el valor que tiene el cableado en sí. El uso de un Organizador de Turnos Inalámbrico por Radiofrecuencia representa una gran ventaja sobre los organizadores de turnos actuales, ya que se pueden instalar más módulos de turneros en un mismo lugar sin aumentar mucho los implementos requeridos.

La fácil instalación de este nuevo dispositivo de turnos lo hace más accesible a cualquier lugar, ya que sólo se requiere colgarlo y conectarlo a la energía eléctrica; además, el envío de datos vía cableado no ofrece tantas garantías en la instalación del mismo turnero debido al acceso a los espacios en los que se debe colocar dicho cableado.

### **1.1 FORMULACIÓN DEL PROBLEMA**

¿Será posible la innovación de un producto como el organizador de turnos inalámbrico por radiofrecuencia?

## 2. JUSTIFICACIÓN

El organizador de turnos inalámbrico por radiofrecuencia ayudará en la implementación de un sistema económico y de fácil instalación, ya que al no necesitarse cableado entre el transmisor y el receptor para el envío de datos, se genera un ahorro en el costo de la fabricación del sistema.

Actualmente en el comercio los organizadores de turno los produce solamente una empresa en Medellín y su costo oscila para una sola taquilla entre \$1.000.000 y \$1.200.000. El diseño propuesto permitirá abrir más el mercado, ya que sería una opción más de dónde escoger al momento de necesitarse el producto en el comercio, y su costo sería menor en relación con los modelos existentes.

La implementación del módulo transmisor y receptor inalámbrico de radiofrecuencia constituye una mejora tecnológica frente a los organizadores de turno actuales que hay en el mercado, a lo cual se suma que es un dispositivo de alta calidad y de excelente presentación.

Este proyecto del Organizador de Turnos Inalámbrico por Radiofrecuencia puede ser también el inicio de una microempresa para fomentar el desarrollo personal y empresarial del realizador del proyecto y contribuir al avance tecnológico y laboral del país.

El proyecto desarrollado en el presente trabajo servirá también de apoyo para estudiantes como material de consulta en la utilización de la programación en C, como alternativa en la elaboración de programas para microcontroladores PIC.

### **3. OBJETIVOS**

#### **3.1. OBJETIVO GENERAL**

Diseñar un sistema electrónico aplicativo consistente en un organizador de turnos inalámbrico como alternativa para mejorar la atención de usuarios, de manera oportuna, ordenada, rápida y eficaz en todo tipo de establecimientos y empresas que realicen atención al público.

#### **3.1 OBJETIVOS ESPECÍFICOS**

Realizar el montaje de un organizador de turnos inalámbrico de manera que su visualización sea fácil y agradable para cualquier tipo de usuarios.

Diseñar e implementar las fuentes de alimentación para los diferentes circuitos del organizador de turnos.

Elaborar e implementar el circuito para la transmisión y control de datos para el organizador de turnos inalámbrico.

Crear e implementar los circuitos de recepción de datos y de potencia para la visualización de los datos en los diferentes display.

Aplicar los conocimientos adquiridos sobre microcontroladores en la elaboración de los programas para el correcto funcionamiento del organizador de turnos inalámbrico.

## 4. MARCO TEÓRICO

Con el nacimiento de la electrónica digital se han creado nuevos mecanismos y dispositivos que permiten en la humanidad cambios organizados en su manera de vivir, como casas controladas electrónicamente, dispositivos electrónicos instalados en los vehículos, como el geoposicionador GPS para ubicarse mejor en determinada ciudad o país. Otro ejemplo son los organizadores de turno como mecanismos electrónicos que ayudan a la organización para la atención controlada, rápida y fácil de usuarios, estas aplicaciones son gobernadas por microcontroladores.

### 4.1. MICROCONTROLADORES<sup>1</sup>

Los microcontroladores están conquistando el mundo: están presentes en nuestro trabajo, en nuestro hogar y en nuestra vida en general, pero esto es solo el comienzo; el siglo XXI será testigo de la conquista masiva de estos diminutos computadores, que gobernarán la mayor parte de los aparatos que fabricamos y usamos los humanos.

Los PIC son los circuitos integrados de Microchip Technology Inc., que pertenecen a la categoría de los microcontroladores, es decir, aquellos que integran en un único dispositivo todos los circuitos necesarios para realizar un completo sistema digital programable; internamente contienen todos los dispositivos típicos de un sistema a microprocesador.

El microcontrolador ha sido diseñado para que en su memoria de programa se almacenen todas las instrucciones del programa de control. Como el programa por ejecutar siempre es el mismo, este debe ser grabado de forma permanente.

---

<sup>1</sup> Toda la información que sirvió de consulta para el tema de los microcontroladores se encontró en las páginas de internet: <http://www.alos.5u.com/indexpic.877.htm> y <http://www.ziddu.com/download/5597452/DATASHEETSESPAOLPARTES.rar.html>

La memoria de datos que maneja los programas es de lectura-escritura, como la memoria RAM estática. La familia PIC 16X8XX dispone de 256 bytes de memoria EEPROM para contener datos; de esta forma, un corte en el suministro de alimentación no producirá la pérdida de los datos.

Las líneas de E/S sirven para establecer la comunicación con los periféricos que se van a controlar, y se adaptan para manejar la información en paralelo. Algunos modelos soportan comunicación serial, como el PIC 16F877 y el PIC 16F628A, además esta gama de microcontroladores 16F87X posee más características, entre las cuales podemos mencionar:

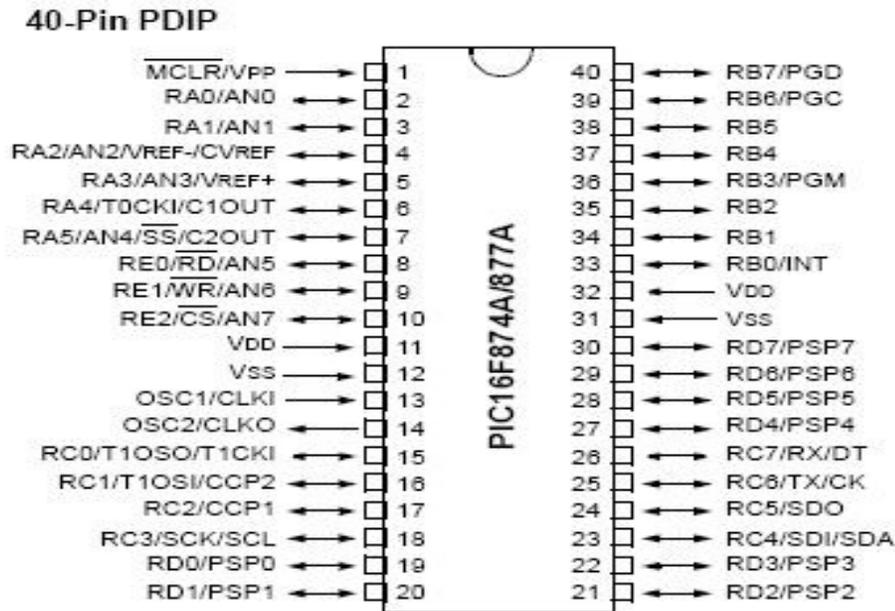
- Procesador de arquitectura RISC avanzada.
- Juego de 35 instrucciones de 14 bits de longitud.
- Frecuencia de operación de 4 a 20 MHz.
- 368 bytes en memoria de datos RAM.
- 256 bytes en memoria EEPROM.
- Pila con 8 niveles.
- Perro Guardián (WDT).
- Modo Sleep de bajo consumo.
- Voltaje de alimentación comprendido entre 2 y 5.5 VDC.
- Bajo consumo en corriente (menos de 2 mA a 5 VDC).

En el programa, cuando el microcontrolador recibe una señal de interrupción, el contador de programa apunta a la dirección 0004h de la memoria de programa. Es por eso que allí debe escribir toda la programación que se desea que el PIC ejecute al momento de producirse dicha interrupción; esto se logra escribiendo "ORG 0x04", que define a partir de esa dirección que todo lo que escriba corresponde al vector de interrupción.

La familia PIC de memoria tipo Flash permite reprogramarlos las veces que sea necesario sin necesidad de usar ningún otro dispositivo más que el propio programador. Este aspecto es muy importante a la hora del diseño de un dispositivo para evitar pérdida de tiempo en borrar los microcontroladores y volver a programarlos.

En la figura 1 se ve el tipo de encapsulado en que viene uno de los modelos pertenecientes a esta familia, cada uno de sus pines o terminales tiene más de un uso, dependiendo de cómo se los configure, excepto los terminales  $V_{DD}$  y  $V_{SS}$  que son los encargados de alimentar este microcontrolador.

**Figura 1.** PIC 16F877.



Fuente: tomado de <http://www.alos.5u.com/indexpic.877.htm>

Los microcontroladores tienen un espacio de memoria que varía según la familia y según el modelo dentro de cada familia. Es allí donde se guarda el programa que se creó, dado que el microcontrolador sólo interpreta el lenguaje de máquina, es decir, estados lógicos de 1 ó 0; es complicado programar con este lenguaje, por eso existe un compilador, —el “CC compiler”—, el cual compila en lenguaje C para ser más corto en instrucciones, y además facilita buscar errores con base en el mismo lenguaje de bajo nivel llamado “ensamblador”. Este compilador ofrece más ventajas, como las denominadas librerías, que con solo llamarlas en el programa ejecuta una serie de instrucciones sin necesidad de programarlas. Una de estas librerías es “kbdd\_4x4.c” para el teclado.

La frecuencia máxima de trabajo depende del modelo específico del microcontrolador. En el PIC 16F877 la máxima frecuencia del reloj externo es de 20 MHz, para el caso del organizador de turnos será de 4 MHz; un ciclo de instrucción equivale a 4 ciclos de reloj, esto es porque con el primer ciclo de reloj el microcontrolador busca en su memoria la instrucción que debe leer, en el segundo ciclo la instrucción se carga en la memoria principal, el tercer ciclo se encarga de ejecutarla, y el cuarto ciclo limpia la memoria para volver a buscar la siguiente instrucción, y así sucesivamente.

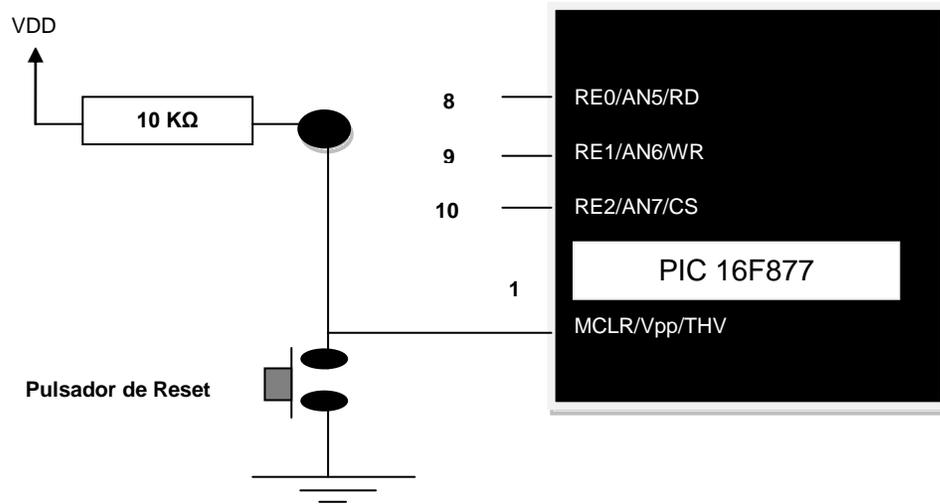
Los microcontroladores de la familia 16F87X se alimentan con 5 V<sub>CC</sub>, pero también funcionan con 3,5 voltios, consumiendo un poco más de corriente, la corriente máxima que puede circular por el terminal V<sub>DD</sub> (donde se conecta el positivo de la fuente de 5 V) es de 250 ma y por cada salida no se debe hacer circular más de 25 ma.<sup>2</sup>

Los microcontroladores tienen un terminal llamado MCLR o master-clear, el cual debe estar en “1” lógico para que se lean y ejecuten las instrucciones, para *reseteo* el microcontrolador es hacer que empiece a leer el programa grabado en ellos desde el principio, una de las técnicas más sencillas es implementar el *reset* con el circuito.

---

<sup>2</sup> Estos datos los proporciona Microchip, en: <http://ww1.microchip.com/downloads/en//35007.pdf>

**Figura 2.** Circuito de *reset* de un microcontrolador.



Fuente: tomado de <http://www.alos.5u.com/indexpic.877.htm>

Todos los microcontroladores de la firma Microchip cuentan con registros internos (la mayoría de estos son de escritura-lectura), los cuales permiten configurar cada bloque para que funcione de acuerdo a las necesidades del programador; para cada microcontrolador existen tres bloques bien diferenciados de memoria, que son:

- Memoria de programa: es el lugar físico donde se guarda el programa.
- Memoria de datos: es el lugar donde se guardan los datos, en la EEPROM.
- Memoria de registros: es el lugar físico de los registros que controlan cada módulo.

La memoria de registros es la parte donde se encuentran los registros que manejan cada módulo del microcontrolador, también existen registros no definidos o bien para ser definidos por el programador para cualquier uso (variables, constantes, acumuladores auxiliares, etc.). Estos registros de propósito general también son de 8 bits y se definen asignándoles un nombre y una dirección.

Como existen registros de control en los 4 bancos disponibles para los 16F87X y 2 bancos para los 16F8X, se debe tener bien claro que para modificarlos o



Como se ha mencionado, la organización de la memoria del PIC 16F877 se divide en memoria de datos y programa. La memoria de datos se divide a su vez en SFR (Special Function Register), unos registros que ayudan a configurar el hardware interno, y que sirven además para escribir o leer valores de los diferentes componentes que constituyen el microcontrolador; y en un juego de 35 instrucciones de 14 bits cada una y diferenciadas en tres tipos que son:

Orientadas a byte: reservan los 7 bits de menor peso para indicar la dirección del registro que será operado. Después de la operación se usa el bit d para indicar dónde se guardará el resultado; si d es cero se guarda en W, y si d es 1 se guarda en el mismo registro que se utilizó en la operación.

Orientadas a bit: estas buscan escribir una posición (bit) dentro de un archivo o registro.

Orientadas a control: son las que ayudan a formar bucles dentro de los programas, y también sirven para llamar a subrutinas o procedimientos (instrucciones CALL y GOTO).

**Tabla 2.** Set de instrucciones y sintaxis del PIC 16F877 y 16F628A.

Sintaxis	Descripción	Ciclos de Instrucción
BCF f,b	Pone en "0" el bit b del registro f	1
BSF f,b	Pone en "1" el bit b del registro f	1
BTFSS f,b	Lee el bit b del registro f, inserta la instrucción siguiente si está en "1"	1(2 si salta)
ADDWF f,d	Suma el contenido del registro W con el de f	1
ANDWF f,d	Realiza la operación lógica AND entre el contenido del registro W con el de f	1
CLRF f	Borra el contenido del registro f	1
CLRW	Borra el contenido del registro W	1

**Tabla 2.** Continuación.

Sintaxis	Descripción	Ciclos de Instrucción
COMF f,d	Niega el valor del registro f	1
DECF f,d	Decremento en una unidad el contenido del registro f	1
DECFSZ f,d	Decremento en una unidad el contenido del registro f y salta la instrucción siguiente si es cero	1(2 si salta)
INCF f,d	Incrementa en una unidad del contenido del registro f	1
RLF f,d	Rota el contenido del registro f hacia la izquierda a través del bit de acarreo	1
RRF f,d	Rota el contenido del registro f hacia la derecha a través del bit de acarreo	1
SUBWF f,d	Resta el contenido del registro W al registro f	1
ANDLW k	Realiza la operación lógica AND entre el valor k y el registro W	1
CALL k	Llamada a la subrutina k	2
GOTO k	Salto incondicional a la etiqueta k	2
IORLOW k	Realiza la operación lógica OR entre el valor k y el registro W	1
MOVLW k	Guarda en el registro W el valor k	1
RETFIE	Retorno de una interrupción	2
RETLW	Retorno de una subrutina con el valor k en el registro W	2
SLEEP	Se pone el PIC en estado de reposo (dormido)	1
SUBLW k	Resta el valor k al registro W	1
XORLW	Realiza la operación lógica XOR entre el valor k y el registro W	1
XORWF f,d	Realiza la operación lógica XOR entre el contenido del registro W con el de f	1
SWAPF f,d	Intercambia los cuatro primeros bits con los cuatro últimos del registro f	1
ADDLW k	Suma el valor k al registro W	1
INCFSZ f,d	Incrementa en una unidad el contenido del registro f y salta la instrucción siguiente si es cero	1(2 si salta)
IORWF f,d	Realiza la operación lógica OR entre el contenido del registro W con el de f	1
MOVF f,d	Mueve el contenido del registro f	1
MOVWF f	Mueve el contenido del registro W al registro f	1
NOP	No realiza ninguna operación	1

Fuente: tomado de <http://www.alos.5u.com/indexpic.877.htm>

Este tipo de microcontroladores PIC permiten cuatro tipos de osciladores externos para aplicar la frecuencia de funcionamiento durante el proceso de grabación, antes de introducir el programa en memoria, debe indicárseles el tipo de oscilador empleado en los bits  $F_{OSC0}$  y  $F_{OSC1}$  de la palabra de configuración. Los tipos de osciladores que puede utilizar son:

- Oscilador de cristal o resonador de alta velocidad “HS”: este oscilador trabaja a una frecuencia comprendida entre 4MHz y 20MHz (Cristal de alta velocidad/Resonador).
- Oscilador o resonador cerámico “XT”: se trata de un oscilador estándar que permite una frecuencia de reloj comprendida entre 100KHz y 4MHz (Cristal/Resonador).
- Oscilador de cuarzo o resonador cerámico de baja potencia “LP”: se trata de un oscilador de bajo consumo, con un cristal o resonador diseñado para trabajar con frecuencias comprendidas entre 32KHz y 200KHz (Cristal de bajo poder).

Los tres modos, LP, XT y HS, usan un cristal o resonador externo, la diferencia, sin embargo, es la ganancia de los *drivers* internos, lo cual se ve reflejado en el rango de frecuencia admitida y la potencia consumida.

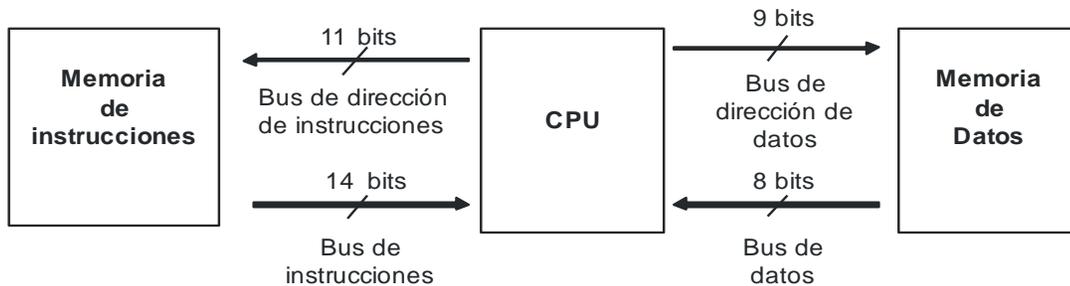
**Tabla 3.** Rangos de frecuencia para la oscilación en base a cristal.

Modo	Frecuencia típica	Capacitores recomendados	
		C1	C2
LP	32 KHz 200 KHz	68 a 100 pf 15 a 30 pf	68 a 100 pf 15 a 30 pf
XT	100 KHz 2 MHz 4 MHz	68 a 150 pf 15 a 30 pf 15 a 30 pf	150 a 200 pf 15 a 30 pf 15 a 30 pf
HS	8 MHz 10 MHz 20 MHz	15 a 30 pf 15 a 30 pf 15 a 30 pf	15 a 30 pf 15 a 30 pf 15 a 30 pf

Fuente: tomado de <http://www.alos.5u.com/indexpic.877.htm>

Otro microcontrolador del proyecto, que hace parte de la etapa del receptor, es el PIC 16F628A, estos microcontroladores soportan hasta 100.000 ciclos en su memoria Flash y un millón de ciclos en su memoria EEPROM, utiliza un procesador con arquitectura Harvard, y logra mayor rendimiento en el procesamiento de las instrucciones.

**Figura 3.** Conexión de la memoria de instrucciones y la memoria de datos en la arquitectura Harvard.



Fuente:<http://www.ziddu.com/download/5597452/DATASHEETSESPAOLPARTES.rar.html>

La CPU del microcontrolador emplea una avanzada arquitectura RISC de 35 instrucciones importantes, pertenecientes a la gama media de los microcontroladores PIC. La mayoría de las instrucciones se ejecutan en un ciclo de instrucción, a excepción de los saltos, que requieren dos ciclos. Su procesador contiene una pila de 8 niveles que permiten el anidamiento de subrutinas, lo que significa que puede recordar 8 lugares diferentes de línea de programa e ir regresando a cada uno en el orden inverso al que fueron anidados.

La memoria de programa también se conoce como memoria de instrucciones, y en ella se describen las órdenes para que la CPU las ejecute. El PIC 16F628A tiene memoria de programa no volátil tipo Flash, otra de sus características es la memoria EEPROM, la cual es más rápida en el proceso de escritura/borrado eléctrico, y dispone de mayor capacidad de almacenamiento. Esto hace que sea ideal para prácticas de laboratorio donde la grabación y el borrado son frecuentes, ya que puede soportar hasta 100.000 ciclos de escritura /borrado.

El otro tipo de memoria es una memoria auxiliar no volátil llamada EEPROM, con capacidad de 128 posiciones de 8 bits cada una; el usuario puede acceder a esta memoria mediante programación, es muy útil para almacenar datos que el usuario necesita conservar aún sin el voltaje de alimentación; tal es el caso de la clave de una alarma, que puede ser modificada pero no debe perderse por un corte de energía.

Como este microcontrolador es fabricado con tecnología CMOS, su consumo de potencia es muy bajo (2 mA a 4 MHz) y además es completamente estático, lo que significa que si el reloj se detiene los datos de la memoria RAM no se pierden, siempre y cuando el microcontrolador siga alimentado. La memoria de datos RAM, tiene 512 líneas de 8 bits cada una y está particionada por 4 bancos: el banco 0, banco 1, banco 2 y banco 3, cada uno con 128 bytes, el acceso a cada banco de memoria lo realizan los bits RP1 y RP0 del registro Status. La mayoría de los bytes son ocupados por los Registros de Funciones Especiales (SFR) o no están implementadas para los diferentes bancos de acceso. Para el caso del PIC 16F628A sólo están disponibles 224 posiciones de memoria RAM para los registros de propósito general (GPR).

Las características generales más relevantes son:

- Oscilador interno RC (resistencia condensador) de 4 MHz calibrado de fábrica al 1%.
- Admite 8 configuraciones de oscilador.
- Ocho niveles de PILA.
- Procesador con arquitectura Harvard.
- Instrucciones de un ciclo, excepto los saltos (200ns por instrucción a 20 MHz).
- Resistencias de Pull-Up, programables en el puerto B.
- Pin RA5, el máster *clear* programable como *reset* externo o como pin de entrada.
- Rango de operación desde 3V hasta 5.5V.
- Quince pines de I/O y 1 solo de entrada (RA5).
- Temporizador Perro guardián (WDT) independiente del oscilador.
- Programable con bajo voltaje LPV (5V).
- Programación serial en Circuito CSP por 2 pines: RB6 (reloj) y RB7 (datos).
- Código de protección programable por sectores.
- Memoria de programa Flash 2048K de 100.000 ciclos de escritura/borrado.
- Dos circuitos comparadores analógicos con entradas multiplexadas.
- Tres *Timers*: *Timer 0* a 8 bits, *Timer 1* a 16 bits y *Timer 2* a 8 bits.
- Diez fuentes de interrupción.
- Módulo de comunicación serial USART/SCI.

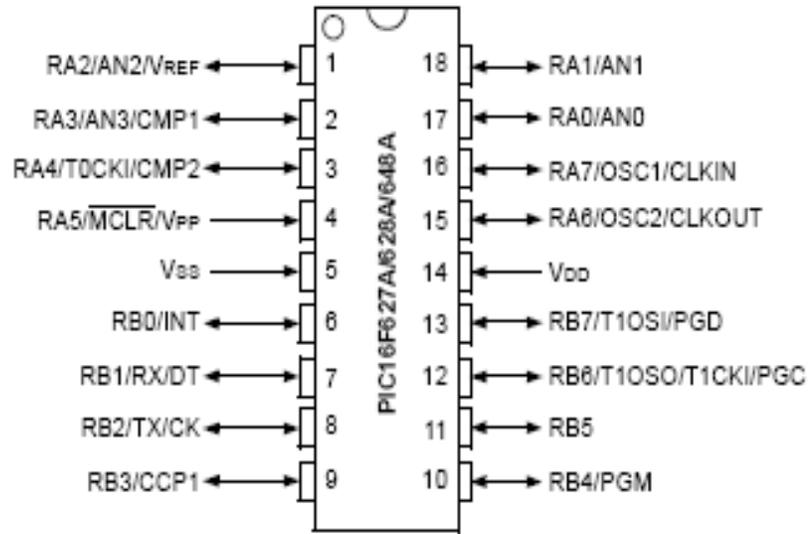
- Capacidad de corriente para encender leds directamente (25 mA I/O) por cada pin.

Tabla 4. Descripción de los pines del PIC 16F628A.

<b>PIN</b>	<b>Nombre</b>	<b>Descripción</b>
17	RA0/AN0	Pin bidireccional I/O, entrada comparador análogo.
18	RA1/AN1	Pin bidireccional I/O, entrada comparador análogo.
1	RA2/AN2/VREF	Pin bidireccional I/O, entrada Comp. Análogo y Voltaje de referencia
2	RA3/AN3/CMP1	Pin I/O entrada Comp. Análogo y salida del comparador análogo1.
3	RA4/T0CK/CMP2	Pin I/O, entrada reloj TIMER0 y salida del comparador análogo 2.
4	RA5/MCLR/VPP	Pin de entrada, en modo MCLR activa RESET externo
15	RA6/OSC2/CLKOUT	Pin I/O, entrada oscilador externo, salida de ¼ de la frecuencia OSC1.
16	RA7/OSC1/CLKIN	Pin I/O, entrada oscilador externo, entrada del reloj externo.
6	RB0/INT	Pin I/O, resistencia Pull-Up programable, entrada de interrupción externa.
7	RB1/RX/DT	Pin I/O, resist. Pull-Up, entrada dato RS232, I/O dato serial asincrónico.
8	RB2/TX/CK	Pin I/O, resist. Pull-Up, entrada dato RS232, I/O señal de reloj asincrónico.
9	RB3/CCP1	Pin I/O, resist. Pull-Up, módulo CCP/PWM entrada o salida.
10	RB4/PGM	Pin I/O, resist. Pull-Up, entrada de voltaje bajo.
11	RB5	Pin I/O, resist. Pull-Up programable.
12	RB6/T1OSO/T1CKI	Pin I/O, resist. Pull-Up, salida oscilador TIMER1, entrada reloj de ICSP.
13	RB7/T1OSI	Pin I/O, resist. Pull-Up, entrada oscilador TIMER1, I/O de datos ICSP.

Fuente:<http://www.ziddu.com/download/5597452/DATASHEETSESPAOLPARTES.rar.html>

**Figura 4.** Pines del PIC 16F628A.



Fuente:<http://www.ziddu.com/download/5597452/DATASHEETSESPAOLPARTES.rar.html>

## 4.2. DRIVER DE CORRIENTE<sup>3</sup>

Para los circuitos TTL, el dominio en el voltaje de entrada que puede representar un BAJO válido (0 lógico) es de 0 a 0,8 Voltios, para representar un ALTO válido (1 lógico) es de 2 V a V<sub>CC</sub> (usualmente 5 V). El rango de valores entre 0,8 V y 2 V es una región de desempeño impredecible, cuando un voltaje de entrada está en este rango, puede interpretarlo el circuito lógico como alto o como bajo, por lo tanto, las compuertas TTL no pueden operarse con confianza cuando los voltajes están en ese dominio.

Un circuito TTL con una salida de pilar totémico está limitado en la cantidad de corriente que puede absorber en el estado BAJO ( $I_{OL}$  (máx.)) hasta 16 mA para TTL estándar y 20 mA para AS TTL. En muchas aplicaciones especiales, una compuerta debe excitar dispositivos externos, como Leds, lámparas o relevadores,

---

<sup>3</sup> La hoja de datos sobre el regulador de voltaje 7805 fue de gran ayuda, y sirvió como referencia para el diseño de los circuitos del proyecto. Se encuentra en la dirección: [http://www.datasheetcatalog.org/datasheets/228/390068\\_DS.pdf](http://www.datasheetcatalog.org/datasheets/228/390068_DS.pdf)

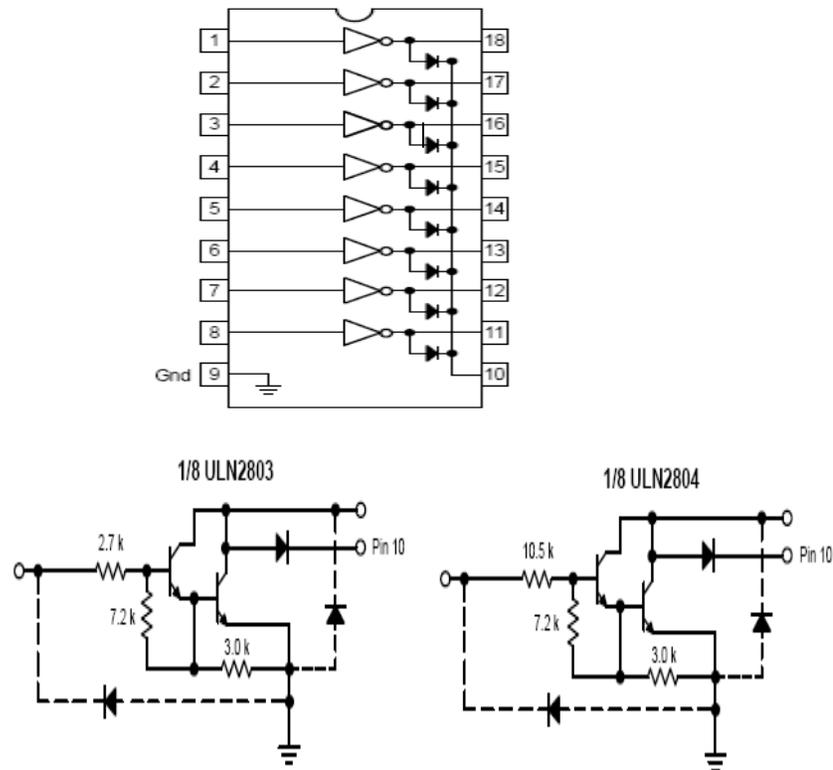
que pueden requerir más corriente que ésta. Por su voltaje y capacidad de manejo de corriente más alto, los circuitos con salidas de colector abierto se usan generalmente para excitar Leds, lámparas o relevadores, sin embargo, las salidas de pilar totémico pueden usarse mientras la corriente de salida, requerida por el dispositivo externo, no exceda la cantidad que el excitador TTL puede absorber.

Con una compuerta TTL de colector abierto, el colector del transistor de salida se conecta a un LED o una lámpara incandescente, también se conecta un resistor en serie con el LED para mantener la corriente por debajo de su valor máximo.

El ULN 2803 es un circuito integrado ideal para ser empleado como interfaz amplificadora entre las salidas de un PIC, o cualquier integrante de las familias TTL o CMOS, y dispositivos que necesiten una corriente más elevada para poder funcionar, como por ejemplo un relé, un display, etc. En este proyecto se utilizará un ULN 2803, ya que es un dispositivo de fácil adquisición y bajo presupuesto, y además es compatible con la tecnología TTL y CMOS para la parte de la visualización de datos.

Todas sus salidas son a emisor abierto, contiene 8 transistores NPN Darlington y dispone de un diodo en cada salida para evitar las corrientes inversas. El modelo ULN 2803 está especialmente diseñado para ser compatible con entradas TTL, mientras que el modelo ULN 2804 está optimizado para voltajes entre 6 y 15 VDC; además tiene una resistencia de entrada de 2,7 K $\Omega$  para 5 VDC TTL y CMOS, cada Transistor Darlington se caracteriza por un rango de corriente máximo de 600 mA (500 mA continuos), y puede oponerse a voltajes menores de 50 voltios fuera de este estado. Las salidas pueden conectarse en paralelo con otro ULN 2803 para producir más cantidad de corriente.

**Figura 5.** Pines del integrado ULN 2803.



Fuente: tomado de <http://www.ucontrol.com.ar/wiki/index.php?title=ULN2803>

**Tabla 5.** Características eléctricas del ULN 2803.

Símbolo	Parámetro	Condiciones a prueba	Mínimo	Típico	Máximo	Unidad	Figura
ICEX	Corriente de pérdida de salida	VCE = 50V Tamb = 70°C, VCE = 50V Tamb = 70°C			50 100	µa µa	1 1
VCE(sat)	Voltaje de saturación Colector - Emisor	IC = 200 ma, IB = 350 µa		1.1 <b>1.1</b>	1.3 1.3	V V	2
li(on)	Corriente de entrada	Vi = 3.85		0.93	1.35	ma	3
li(off)	Corriente de entrada	Tamb = 70°C, IC = 500 µa	<b>50</b>	65		µa	4

**Tabla 5.** Continuación.

Símbolo	Parámetro	Condiciones a prueba	Mínimo	Típico	Máximo	Unidad	Figura
Vi(on)	Voltaje de entrada	IC = 200 ma IC = 250 ma IC = 300 ma			2.7 3.0 3.0	V V V	5
hFE	Ganancia de corriente DC	-----	1000	-----	-----		2
Ci	Capacitancia de entrada			15	25	pF	-----
tPLH	Tiempo retardo de encendido	0.5 Vi a 0.5 Vo		0.25	1	µs	-----
tPHL	Tiempo retardo de apagado	0.5 Vi a 0.5 Vo		0.25	1	µs	-----
IR	Corriente de pérdida diodo de sujeción	VR = 50V Tamb = 70 °C			50 100	µa µa	6
VF	Voltaje diodo de sujeción	IF = 350 ma		1.7	2.0	V	7

Fuente:[http://www.datasheetcatalog.org/datasheet/on\\_semiconductor/Uln2803-D.PDF](http://www.datasheetcatalog.org/datasheet/on_semiconductor/Uln2803-D.PDF)

### 4.3. TRANSMISIÓN Y RECEPCIÓN DE RADIOFRECUENCIA<sup>4</sup>

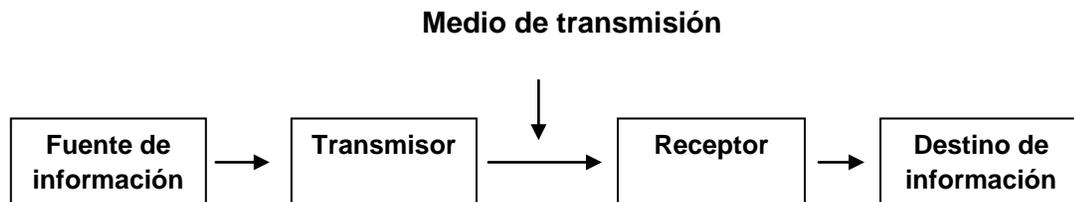
La innovación de este proyecto es la adaptación de un módulo transmisor y receptor de datos inalámbrico por radiofrecuencia, el cual aventaja en buena medida a los antiguos organizadores de turno elaborados con cableado, para el envío, ya sea en forma serial o en forma paralela, de la información de los turnos.

Un transmisor es un conjunto de uno o más dispositivos o circuitos electrónicos que convierte la información de la fuente original en una señal que se presta más a su transmisión a través de determinado medio de transmisión. El medio de transmisión transporta las señales desde el transmisor hasta el receptor, y puede

<sup>4</sup> La información de la transmisión y recepción de Radiofrecuencia fue tomada del libro: TOMASI, WAYNE. Sistemas de comunicaciones electrónicas, PEARSON EDUCACIÓN, México, 2003.

ser tan sencillo como un par de conductores de cobre que propaguen las señales en forma de flujo de corriente eléctrica.

**Figura 6.** Transmisión y recepción de información en una dirección.



Fuente: tomado de TOMASI, Wayne. Sistemas de Comunicaciones Electrónicas. México: Pearson Educación, 2003.

Existen dos tipos de sistemas de comunicaciones electrónicas: analógico y digital; un sistema de comunicaciones analógico es aquel en el cual la energía electromagnética se transmite y recibe en forma analógica (una señal que varía continuamente como una onda senoidal). Un sistema de comunicaciones digital es aquel en el cual la energía electromagnética se transmite y recibe en forma digital (niveles discretos tal como +5 voltios y tierra). Los sistemas binarios utilizan señales digitales que solo tienen dos niveles discretos (*bi* significa dos). Con frecuencia la información de la fuente original está en una forma que no es la adecuada para la transmisión y se debe convertir a una forma más adecuada antes de la transmisión, por ejemplo, con los sistemas de comunicaciones digitales, la información analógica se convierte a una forma digital antes de la transmisión, y con los sistemas de comunicaciones analógicos, la información digital se convierte a la forma analógica antes de la transmisión.

El propósito de un sistema de comunicaciones electrónico es comunicar información entre dos o más ubicaciones (generalmente llamadas estaciones); esto se logra convirtiendo la información de la fuente original a energía electromagnética y después transmitiendo la energía a uno o más destinos, en donde se convierte de nuevo a su forma original. La energía electromagnética

puede propagarse de varios modos: como un voltaje o una corriente a través de un cable metálico; como ondas de radio emitidas por el espacio libre, o como ondas de luz por una fibra óptica.

Cuando se trata de ondas de radio, es común usar las unidades de longitud de onda en vez de las de frecuencia. La longitud de onda es la longitud que ocupa en el espacio un ciclo de una onda electromagnética (es decir, la distancia entre los puntos semejantes en una onda repetitiva).

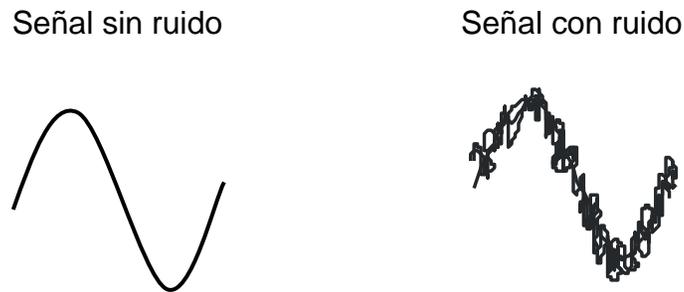
Las dos limitaciones más significativas en el funcionamiento de un sistema de comunicaciones son: el ruido y el ancho de banda. El ancho de banda de un sistema de comunicaciones es la banda de paso mínima (rango de frecuencias) requerida para propagar la información de la fuente a través del sistema.

El ruido eléctrico se define como cualquier energía eléctrica no deseada presente en la pasabanda útil de un circuito de comunicaciones, esencialmente, el ruido puede dividirse en dos categorías generales: correlacionado y no correlacionado, la correlación implica una relación entre la señal y el ruido. El ruido no correlacionado está presente sin importar si hay una señal presente o no, el ruido no correlacionado se puede dividir en dos categorías generales: externo e interno.

El ruido externo se genera externamente a un circuito y se introduce en el, las señales generadas externamente se consideran ruido, solo si sus frecuencias caen dentro de la banda útil del filtro de entrada del circuito. El ruido interno es la interferencia eléctrica generada dentro de un dispositivo.

El ruido correlacionado es una energía eléctrica no deseada que está presente como resultado directo de una señal, tales como las distorsiones armónicas y de intermodulación. Las distorsiones armónicas y de intermodulación son formas de distorsión no lineal y son producidas por la amplificación no lineal.

**Figura 7.** Los efectos del ruido sobre una señal.



Fuente: tomado de TOMASI, Wayne. Sistemas de Comunicaciones Electrónicas. México: Pearson Educación, 2003.

La transmisión por radiofrecuencia está a cargo del transmisor y el receptor. El módulo transmisor y el receptor trabajan con una portadora de 433 MHz, la información se maneja en ASK (ver anexo I). La modulación por desplazamiento de amplitud es una forma de modulación en la cual se representan los datos digitales como variaciones de amplitud en la onda portadora. La amplitud de la señal portadora análoga varía conforme a la corriente de bit (se modula la señal), manteniendo la frecuencia y la fase constantes; el nivel de amplitud puede ser usado para representar los valores binarios ceros y unos. En la señal modulada, el valor lógico cero es representado por la ausencia de la portadora y el valor lógico uno es representado por la presencia de portadora. En el anexo I se muestra la conexión del módulo transmisor y el receptor con el PIC 16F877 y el 16F628A para la correcta transmisión y recepción de la taquilla y los turnos.

El envío de la información se da por un vector que tiene la información de la taquilla, turno y un fin de trama, para saber hasta dónde se han enviado estos tres datos, y así el receptor controlado por el PIC 16F628A puede saber qué le ha llegado. El módulo transmisor tiene la capacidad de transmitir a una velocidad de 9.600 bits por segundo, pero para mayor confiabilidad en la transmisión se trabajará a una velocidad de 2.400 bits por segundo.

El módulo receptor capta la información enviada por el transmisor y trabajará a una velocidad de 2.400 bits por segundo. Este pequeño módulo trabaja a 5 VDC,

su frecuencia de operación también es la misma que la del receptor y su tamaño es físicamente ajustable a cualquier aplicación, más específicamente para el turno.

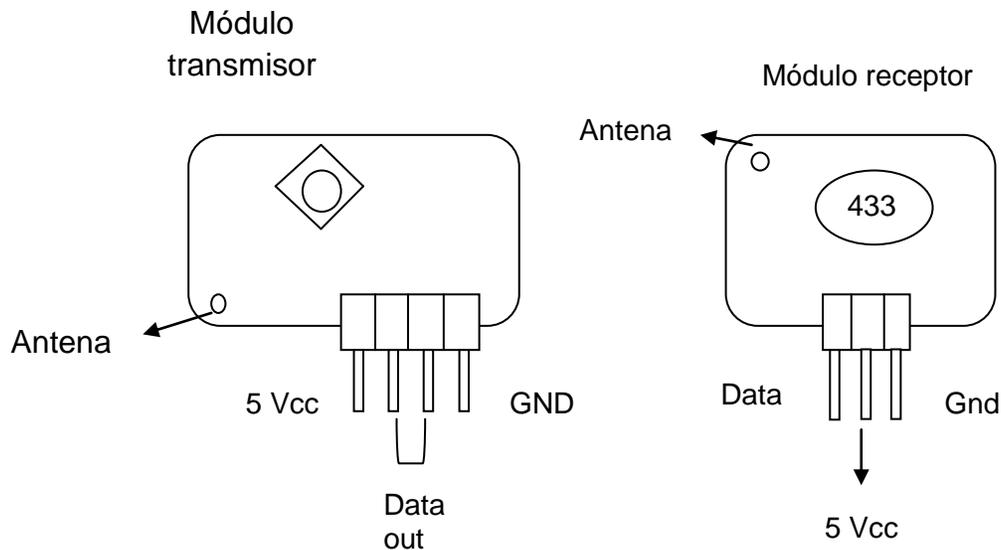
**Tabla 6.** Características del módulo Transmisor y Receptor de radiofrecuencia.

<b>Características del transmisor</b>	<b>Valores para el transmisor</b>
Voltaje de operación	(3 a 12) voltios DC
Corriente de operación	(5 a 45) mA
Tasa de transferencia	2.4 KHz (Max. 9.6 KHz)
Modulación	ASK
Frecuencia de operación	433 MHz
Rango de transmisión	1000 mts
Dimensiones	19 mmx 19 mmm x8 mmm
<b>Características del receptor</b>	<b>Valores para el receptor</b>
Voltaje de operación	5 VDC
Demodulación	ASK
Frecuencia de operación	433 MHz
Dimensiones	14 mm x 38 mm

Fuente: tomado de [http://articulo.mercadolibre.com.mx/MLM-71080464-modulo-rf-433mhz-superheterodino-rf-kit-arduino-pic-mcu-\\_JM](http://articulo.mercadolibre.com.mx/MLM-71080464-modulo-rf-433mhz-superheterodino-rf-kit-arduino-pic-mcu-_JM)

El transmisor consta de cuatro pines: el pin 1 va conectado a  $V_{cc} = 5$  VDC, el pin 2 y 3 van cortocircuitados por especificaciones del fabricante, y son a su vez la salida de los datos hacia el receptor, y el pin 4 va referenciado a GND para su correcto funcionamiento. El receptor recibe los datos por el pin 1 denominado DATA, el pin 2 va conectado a la alimentación de 5 VDC, el pin 3 va referenciado a GND y también posee un pin al cual va conectada su antena.

**Figura 8.** Descripción física del Transmisor y Receptor de Radiofrecuencia.



Fuente: tomado de: [http://articulo.mercadolibre.com.mx/MLM-71080464-modulo-rf-433mhz-superheterodino-rf-kit-arduino-pic-mcu-\\_JM](http://articulo.mercadolibre.com.mx/MLM-71080464-modulo-rf-433mhz-superheterodino-rf-kit-arduino-pic-mcu-_JM)

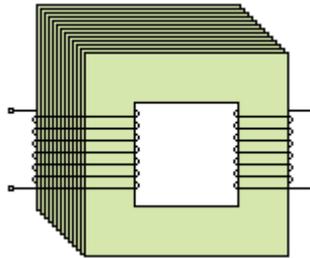
#### 4.4. FUENTE DE ALIMENTACIÓN<sup>5</sup>

La tensión de la red es demasiado elevada para la mayor parte de los dispositivos empleados en circuitos electrónicos, por ello generalmente se usa un transformador en casi todos los diseños electrónicos. Este transformador reduce la tensión a niveles inferiores, más adecuados para su uso en dispositivos como diodos y transistores. Un transformador es un conjunto de chapas de hierro muy juntas que tienen dos arrollamientos, uno a cada lado del conglomerado de las chapas de hierro.

---

<sup>5</sup> La información para la investigación de los reguladores de tensión fue tomada de la página de datos del regulador 7805. En: [http://www.datasheetcatalog.org/datasheets/228/390068\\_DS.pdf](http://www.datasheetcatalog.org/datasheets/228/390068_DS.pdf). Además se adiciona información de: ROJAS, William. Curso de electrónica fácil. Colombia: Cekit S.A., 1998.

**Figura 9.** Transformador de hierro.



Fuente: tomado de ROJAS, William. Curso fácil de electrónica básica. Colombia: CEKIT S.A., 1998.

En estos transformadores la bobina izquierda se llama arrollamiento primario y la derecha arrollamiento secundario. El número de vueltas en el arrollamiento primario es  $N_1$  y en el arrollamiento secundario es  $N_2$ . La relación entre el número de vueltas y la tensión es:  $V_1/N_1 = V_2/N_2$ . El transformador elevador se da cuando el arrollamiento secundario tiene más vueltas que el arrollamiento primario ( $N_2 > N_1$ ), y la tensión del secundario es superior a la del primario ( $V_2 > V_1$ ), es decir,  $N_2:N_1$  es mayor que 1. Cuando el arrollamiento secundario tiene menos vueltas que el arrollamiento primario ( $N_2 < N_1$ ), se induce en el secundario una tensión menor que la que hay en el primario, o sea transformador reductor.

Este transformador-reductor dará la fuerza necesaria para la fuente, es el "TR-12" de 2 amperios de corriente alterna, con una entrada de 110 VAC para su bobinado primario y como salida tiene 12 VAC PICO o 24 VAC AC PICO-PICO en su secundario. Solamente bastará con los 12 VAC PICO y su corriente de 2 amperios para que los circuitos funcionen correctamente.

Además habrá un fusible como elemento de protección, el cual se utiliza para proteger dispositivos eléctricos y electrónicos, y que permite el paso de la corriente mientras no supere el valor establecido en las características del mismo, si el valor de la corriente que pasa es superior a ese valor establecido, el fusible se quema, se abre el circuito y no pasa corriente.

El fusible normalmente se coloca entre la fuente de alimentación y el circuito a alimentar, en equipos eléctricos o electrónicos comerciales el fusible se coloca dentro de éstos, el fusible está constituido por una lámina de hilo metálico que se funde con el calor producido por el paso de la corriente.

Los fusibles tienen la capacidad de conducir una corriente ligeramente superior a la que supuestamente se ha de quemar, esto con el propósito de permitir picos de corriente que son normales en algunos equipos. En las fuentes para el circuito transmisor y receptor se utilizará un fusible de protección de 2 amperios de valor nominal, para que no afecte por alguna sobrecarga el funcionamiento de los circuitos del proyecto.

**Figura 10.** Forma física y simbología eléctrica para un fusible.

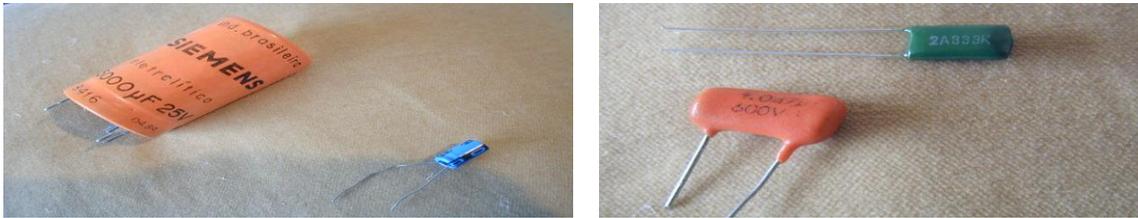


Fuente: tomado de: <http://unicrom.com>

Otro elemento importante para la fuente de alimentación es el condensador, cuya principal característica es la capacidad de almacenar energía eléctrica en forma temporal e irla entregando a diferentes partes del circuito, después de las resistencias, el condensador es el componente más usado en los circuitos electrónicos, y además es un componente pasivo.

En las fuentes de alimentación (de corriente y/o voltaje) se utilizan los condensadores para eliminar (filtrar) el rizado o riple remanente de la conversión de corriente alterna (AC) en continua (DC) realizada por el circuito rectificador. Se tendrán valores variados en algunos circuitos, como en las fuentes de alimentación, que tendrán filtros polarizados de un valor de 2.200 $\mu$ f a un valor para la tensión de 50 VDC, con el fin de darle un mejor filtrado y evitar rizados del componente AC. También habrá condensadores que actuarán como atenuadores de ruido, como lo es el “104”, que es equivalente a 0.1  $\mu$ f.

**Figura 11.** Forma física de un condensador.



Fuente: tomado de: ROJAS, William. Curso fácil de electrónica básica. Colombia: CEKIT S.A., 1998.

Para la fuente de alimentación se necesitará la ayuda de los reguladores de tensión 78xx, los cuales hacen parte de la denominación de una familia de reguladores de tensión positiva 78xx, de tres terminales, con voltaje de entrada  $V_i$ , con voltaje de salida  $V_o$ , y la pata central, que es la masa o común, con especificaciones similares y que solo difieren en la tensión de salida suministrada y en la corriente de trabajo; de ello dependen las letras que intercala detrás de los dos primeros dígitos.

**Tabla 7.** Referencias de la familia 78xx y corrientes que manejan.

Referencia	Corriente de depósito
78 xx (sin letra)	1 amperio
78 Lxx	0.1 amperio
78 Mxx	0.5 amperio
78 Txx	3 amperios
78 Hxx	5 amperios (híbrido)
78 Pxx	10 amperios (híbrido)

Fuente: [http://www.datasheetcatalog.org/datasheets/228/390068\\_DS.pdf](http://www.datasheetcatalog.org/datasheets/228/390068_DS.pdf)

La tensión de salida varía entre 5 y 24 voltios, dependiendo del modelo, y está especificada por los dos últimos dígitos: el 7805 que entrega 5 voltios y el 7812 que suministra 12 voltios de corriente directa, respectivamente, además son dispositivos que se consiguen fácilmente en el mercado, de bajo costo y de trabajo

confiable en la parte de regulación de voltaje. El encapsulado en el que usualmente se los utiliza es el TO220, aunque también se encuentran en encapsulados pequeños de montaje superficial y en encapsulados grandes metálicos (TO3).

La tensión de alimentación debe ser un poco más de 2 voltios superior a la tensión que entrega el regulador y menor a 35 voltios, usualmente soporta corrientes de hasta 1 amperio. Este dispositivo posee como protección un limitador de corriente por cortocircuito, y además otro limitador por temperatura, que puede reducir el nivel de corriente. Las características estándares de los principales modelos son:

**Tabla 8.** Características estándar de los modelos 78xx.

Características y modelos	7803	7805	7806	7808	7809	7810	7812	7815	7818	7824
$V_{out}$	3.3V	5V	6V	8V	9V	10V	12V	15V	18V	24V
$V_{in} - V_{out}$		2.2 V a 30V					2.5V A 23V	2.6V A 20V		
Temperatura de operación (°C)	0 a 125	0 a 125		0 a 125	0 a 125			0 a 125		
$I_{max}$ de salida	1A	1A	1A	1A	1A	1A	1A	1A	1A	1A

Fuente: [http://www.datasheetcatalog.org/datasheets/228/390068\\_DS.pdf](http://www.datasheetcatalog.org/datasheets/228/390068_DS.pdf)

**Tabla 9.** Características eléctricas del regulador de voltaje 7805.

Símbolo	Parámetro	Condiciones	Min	Tip	Max	Unid
$V_o$	Voltaje de salida	$T_j = +25^\circ\text{C}$ $5\text{mA} \leq I_o \leq 1\text{A}$ $V_i = 7\text{V a } 25\text{V}$	4.8 4.75	5.0 5.0	5.2 2.25	V
Reg. Line	Regulación de línea	$T_j = +25^\circ\text{C}$ $V_o = 7\text{V a } 25\text{V}$ $V_i = 8\text{V a } 12\text{V}$	_____	4.0 1.6	100 50	mV mV
$I_q$	Corriente inactiva	$T_j = +25^\circ\text{C}$	_____	5.0	8.0	mA

**Tabla 9.** Continuación.

$\Delta I_Q$	Cambio de la corriente inactiva	$I_o = 5\text{mA}$ a $1^a$ $V_I = 7\text{V}$ a $25\text{V}$	_____	0.03	0.5	mA
			_____	0.3	1.3	mA
$\Delta V_O/\Delta T$	Corrimiento voltaje salida	$I_o = 5\text{Ma}$	_____	-0.8	_____	mV/°C
$V_N$	Voltaje señal ruido	$F = 10\text{Hz}$ a $100\text{kHz}$ , $T_A = +25^\circ\text{C}$	_____	42.0	_____	$\mu\text{V}/V_o$
RR	Rechazo de onda	$F = 120\text{Hz}$ , $V_o = 8\text{V}$ a $18\text{V}$	62.0	73.0	_____	db
$V_{\text{DROP}}$	Voltaje de descenso	$I_o = 1\text{A}$ , $T_J = +25^\circ\text{C}$	_____	2.0	_____	V

Fuente: [http://www.datasheetcatalog.org/datasheets/228/390068\\_DS.pdf](http://www.datasheetcatalog.org/datasheets/228/390068_DS.pdf)

Una parte fundamental del proyecto es la visualización de datos, la cual estará a cargo de un elemento primordial como el LED o el grupo formado por estos llamados display. Los LED son elementos de fácil acceso en el mercado, tienen buen tamaño para lo que se requieren y no son de gran consumo de energía, ya que el empleo de la multiplexación de sus ánodos en la programación de los PIC, hace que esta tecnología sea de gran beneficio para el proyecto.

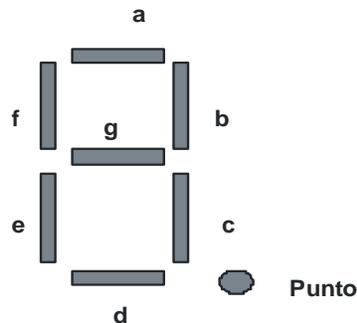
Un LED (Light Emitting Diode, “Diodo Emisor de Luz”), es un dispositivo semiconductor que emite radiación visible, infrarroja o ultravioleta cuando se hace pasar un flujo de corriente eléctrica a través de él en sentido directo; esencialmente es una unión PN cuyas regiones P y N pueden estar hechas del mismo o de diferente semiconductor. El color de la luz emitida está determinado por la energía del fotón, y en general es aproximadamente igual a la energía de salto de banda del material semiconductor en la región activa del LED.

Los elementos que componen los LED son transparentes o coloreados, de un material resina-epoxi, con la forma adecuada, e incluyen el corazón de un LED: el chip semiconductor. Los terminales se extienden por debajo de la cápsula del LED o foco e indican cómo deben ser conectados al circuito. El lado negativo está

indicado de dos formas: 1) Por la cara plana del foco; 2) Por el de menor longitud (el terminal negativo debe ser conectado al terminal negativo de un circuito).

Los LED operan con un voltaje relativamente bajo, entre 1 y 4 voltios, la corriente está en un rango entre 10 y 40 miliamperios. Los voltajes y corrientes superiores a los indicados pueden derretir el chip del LED, existe una gran variedad de formas, intensidades luminosas, dimensiones, colores, etc, los displays de siete segmentos se utilizan en toda clase de productos. Para este proyecto, estos display se utilizan junto con circuitos lógicos, en las salidas de los puertos del PIC que decodifican un número BCD y activan los dígitos adecuados del display.

**Figura 12.** Display de siete segmentos.



Fuente: tomado de TOCCI, Ronald. Sistemas Digitales: Principios y Aplicaciones. 6.ª ed. México: Prentice-Hall Hispanoamericana, 1996.

Al excitar determinadas combinaciones de estos segmentos, se pueden obtener cada uno de los diez dígitos decimales. Este tipo de display digital es para cada uno de los diez dígitos, en los que se utiliza un segmento para indicar cuál está excitado, para generar un 1, se excitan los segmentos b y c; para producir un 2, se excitan los segmentos a, b, g, e y d, y así sucesivamente.

Un tipo muy común de display de 7 segmentos es el de diodos emisores de luz (LED) colocados de manera especial. Cuando se aplica un nivel alto a la entrada de un segmento, el LED se enciende y circula corriente a través de él, en este proyecto se utilizará el display de ánodo común.

**Tabla 10.** Especificaciones técnicas para el display de ánodo común.

<b>Tipo de display</b>	Ánodo común
<b>Voltaje mínimo de operación:</b>	10 voltios
<b>Corriente de operación:</b>	160 miliamperios
<b>Inclinación:</b>	10°
<b>Color:</b>	Rojo
<b>Tamaño:</b>	Ancho 9 cm. – Altura 12 cm
<b>Composición de cada segmento:</b>	5 LED'S en serie.

Fuente: tomado de: <http://www.didacticaselectronicas.com>

## 5. METODOLOGÍA

### 5.1 TIPO DE PROYECTO

Se trata de un proyecto aplicativo y de desarrollo, ya que partiendo de los conocimientos obtenidos durante la carrera se ha procedido a la creación propia de un organizador de turnos inalámbrico por radiofrecuencia.

### 5.2. MÉTODO

Para el desarrollo del organizador de turnos, se realizó un tipo de estudio teórico-práctico, el cual por medio del análisis y la síntesis que son procedimientos que se complementan, ya que una sigue a la otra en su ejecución, básicamente consiste en analizar por partes los elementos constitutivos para determinar su importancia, la relación entre ellos, cómo están organizados y cómo funcionan.

### 5.3. TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN

**5.3.1 Fuentes primarias.** El organizador de turnos inalámbrico por radiofrecuencia se hizo como una mejora a los actuales turneros; porque se reemplazo el cableado que hay entre la taquilla y los turnos por un módulo inalámbrico para la transmisión y la recepción de datos.

**5.3.2 Fuentes secundarias.** Asesores en programación de microcontroladores, Internet, fichas técnicas de los display y los transmisores y receptores de radiofrecuencia.

## 6. RESULTADOS DEL PROYECTO

Como todo circuito electrónico necesita de una fuente de alimentación, se empezó con el diseño de las mismas como primera etapa; ya que son la fuerza para poder poner a trabajar las demás partes del trabajo de investigación.

La fuente corresponde a un transformador con entrada de 110 a 115 VAC aproximadamente y una salida de 0 a 12 VAC con una corriente de 2.0 amperios de corriente alterna. Este transformador es el encargado de rebajar el voltaje de la red de 110 VAC a un voltaje aproximado de 12 voltios, que es el que se necesita para alimentar el circuito. Además, las dos fuentes cuentan con su fusible de protección de 2 amperios, un filtro de línea que absorbe los excesos de voltaje que de manera imprevista se presentan en la red AC. En la elaboración de las fuentes no hubo ningún problema de cortocircuitos ni sobrecalentamientos de las tarjetas y los componentes.

La rectificación está constituida por un puente rectificador de onda completa que maneja una tensión de 110 voltios AC y una corriente máxima de 4 amperios, además, en este puente rectificador se colocaron 4 condensadores de referencia 104 (0,1 $\mu$ f), para eliminar interferencias de frecuencias parásitas que afecten el funcionamiento de la fuente por ruido, los cuales sirvieron mucho; ya que en la transmisión y recepción de datos se presentaron algunas fluctuaciones de ruido y con la ubicación de estos condensadores se mejoro mucho pero no se solucionó del todo el problema, ya que se tuvo que colocar una puesta de malla a tierra en cada una de las cajas del proyecto.

En el filtrado se utilizan dos capacitores de 2.200 $\mu$ F a 50 voltios y un capacitor de 0,1 $\mu$ F a 50 voltios, que son los encargados de filtrar la corriente suministrada por el puente rectificador. A la salida de esta etapa de filtrado se tiene una tensión aproximada de 14,5 VDC que no es regulada, es decir que puede variar con los cambios de voltaje de la carga (circuitos). La regulación de voltaje se hizo en cada una de las tarjetas con los dispositivos reguladores de voltaje 7812 y 7805 para mayor precisión en los voltajes.

El regulador 7812 es el encargado de recibir el voltaje de la fuente de aproximadamente 14,5 VDC y regularlo a 12 VDC, con este voltaje trabaja la etapa receptora de los display y también alimenta los integrados *driver* de corriente ULN 2803 y los transistores Darlington TIP 121.

Con el regulador de voltaje 7805 se trabaja la parte lógica del circuito, como lo es el PIC 16F877, el PIC 16F628A, los decodificadores 7447 y 7448, el optoacoplador 4N33 y los display de la etapa transmisora. También regulan el voltaje para los módulos de transmisión y recepción de radiofrecuencia, ya que estos están trabajando con un voltaje de alimentación de 5 VDC.

El circuito principal es la tarjeta de control o transmisora de datos, que es la encargada de procesar los datos entregados por el teclado y los pulsos lógicos generados por los respectivos mouses, y también de manejar la transmisión de radiofrecuencia con los respectivos datos del turnero, además consta del PIC 16F877, que es el cerebro de esta tarjeta. Para que este PIC pueda funcionar tiene conectados unos componentes externos especificados por la empresa microchip, como son su cristal de 4 MHz y dos filtros de 22 pf, para generar en conjunto su frecuencia de trabajo con la cual manejará el programa de este microcontrolador. También tenemos que los cinco puertos del PIC están configurados de la siguiente manera: como entrada se tiene el puerto A, C y D; como entradas y salidas el puerto B, y como salida el puerto E. Además se cuenta con 3 display pequeños que nos ayudarán a visualizar los datos que se manejan en esta tarjeta, los cuales son taquilla, decenas y unidades.

En la elaboración de esta tarjeta al principio se tuvieron problemas como la respectiva ubicación de cada componente para poder ahorrar espacio, ya que no podía ser demasiado grande porque es la que va a usar el programador de los turnos y estéticamente no se veía bien.

Como entrada principal de periféricos está el teclado matricial 4x4 y los mouses 1 y 2, de donde se programan los turnos, una tecla importante es el "\*", del cual se puede programar el turno en caso de algún error por parte del programador en la

secuencia de los turnos, los mouses 1 y 2 solamente sirven para seleccionar la taquilla e incrementar el turno cada vez que se pulse.

Como periféricos de salida solamente se tiene un display sencillo y uno doble en la tarjeta de control y transmisión: el sencillo para la taquilla y el doble para el turno, vale la pena aclarar que los display son de tamaño pequeño.

Los display de la etapa receptora son 3, son sencillos y de buen tamaño, su color rojo los hace perfectos para la visualización. El voltaje mínimo para su operación es de 10 VDC, pero en el circuito se conectaron a 12 VDC sin ocasionar ningún daño en ellos.

Cada vez que se presiona alguna tecla del teclado matricial 4x4 se genera un voltaje bajo de 0 voltios en el puerto B del PIC 16F877 de 8 bits, de los cuales sus 4 bits más significativos RB7 a RB4 están configurados como salidas, y los cuatro bits menos significativos RB3 a RB0 como entradas.

Con el teclado numérico se le coloco un retardo de 200 ms, en el cual el programa no va a preguntar si se presiono alguna tecla para que no genere algún dato incorrecto en la visualización de los displays tanto de la etapa transmisora como receptora.

El PIC constantemente estará chequeando si se presiona alguno de los mouses que están conectados a los bits RD7, RD6 y RD5, respectivamente, del puerto D; al momento de generar un voltaje de 0 voltios a la salida de los pines que van al microcontrolador 16F877. Cuando el PIC detecta este voltaje, los display mostrarán la taquilla que corresponde al pin RA4 y se incrementará un contador que va de 0 a 99, el cual manejará la parte del conteo lógico de los turnos y a su vez siempre estará mandando los datos por medio de la función *enviar*. Para ahorrar en consumo de energía de la fuente y además en la implementación de más elementos en las tarjetas, se utiliza la técnica de multiplexación de los display, tanto de la tarjeta de control como de la tarjeta receptora, la cual por software, o sea el programa en los respectivos PIC, deja los segmentos de los display prendidos, lo único que varía es la secuencia de prendido y apagado de

los ánodos de cada display con un determinado tiempo de encendido y apagado llamado retardo.

La utilización de los mouse se dio por una estética mejor al proyecto y utilizar los pulsadores que poseen en su estructura como el clic izquierdo; este pulsador se utilizó para incrementar los turnos pero si se colocaba con toda la circuitería interna que este poseía internamente generaba el problema que el PIC 16F877 no reconocía su pulso por la impedancia de todo el mouse ( $4,6\Omega$ ); entonces se procedió a quitarle todos sus componentes y solamente dejarle su respectivo pulsador que da una impedancia de  $0,4\Omega$  aproximadamente.

También se podrán programar los turnos desde el teclado presionando la tecla “\*” por 4 segundos, y el programa entrará a un modo de configuración, donde se podrá poner la secuencia del turno correspondiente en los números del 0 al 99, para poder poner este valor numérico es necesario dejar cada tecla presionada 4 segundos, y así el microcontrolador puede reconocer cuáles fueron los dos números presionados. Para salirse de la programación no hay que presionar ninguna tecla de más, cuando se haya presionado el segundo número automáticamente se sale del modo de programación, el PIC 16F877 procesa el valor y lo transmite para que sea visualizado en la pantalla del turnero por medio de la función *enviar*.

El circuito visualizador de la tarjeta de control son 3 display pequeños cuyos segmentos se conectan todos en paralelo para poder multiplexar sus ánodos y así poder reducir el número de salidas del puerto del microcontrolador. Cuando las salidas del microcontrolador por su puerto A en sus bits RA0 a RA3 son altas, se introduce un código binario en el integrado 7447, el cual es un decodificador que convierte este código de entrada en formato BCD a niveles lógicos, que permiten activar los displays de siete segmentos de ánodo común alimentados con 5 voltios de nuestra tarjeta de control.

La multiplexación de los display de ánodo común se da por los otros puertos del puerto A, que son RA4, RA5 y un bit del puerto E que es el bit 0. Estos bits darán un pulso alto (5 voltios) a la base de los transistores 2N2222, que a su vez

excitarán los ánodos de los display 1, 2 y 3 que son, respectivamente, la taquilla y los turnos. Todas estas etapas cuentan con un voltaje regulado de 5 voltios para mayor seguridad en la comunicación serial de los datos y en su transmisión por las etapas del circuito.

Con la multiplexación de los display de ánodo común se generó un inconveniente como lo era la conmutación de los transistores 2N222 conectados con su resistencia de base del valor de  $1K\Omega$ ; ya que estaba produciendo un switcheo demasiado rápido en la base, entonces se decidió cambiar este elemento por una de valor de  $3,3K\Omega$  y así dando un mejor rendimiento para la polarización de base del transistor.

La etapa de control cuenta con un bit de transmisión que lo proporciona el PIC 16F877, el cual es el pin 26 (RC7), que mandará un vector con cuatro caracteres: el valor de la taquilla, decenas, unidades y un fin de trama, que cierra este vector de transmisión. El transmisor de radiofrecuencia de 433 MHz transmitirá esta secuencia al receptor ubicado en la etapa de potencia y recepción para poder visualizar la taquilla y los turnos en la pantalla para los usuarios, que tendrá 3 display de mayor tamaño que los de la tarjeta de control. Cuando se está en el modo de programación, la secuencia de envío es: una letra "t" que representa taquilla, y dos "c" que representan el contador con las variables decenas y unidades y un fin de trama. Mientras no termine el ciclo de configuración, en el receptor solamente se verán estos valores. Al momento de presionar la segunda tecla del teclado, o sea las unidades (último valor del contador), automáticamente cierra el ciclo y manda el vector de transmisión.

Una ventaja del PIC 16F877 es que si se apagara el circuito no sabríamos la secuencia de los turnos, pero como el programa siempre está guardando el valor del contador en la EEPROM del PIC, al momento de energizar otra vez el circuito, este mostrará el valor en el que iba sin perderse ningún dato.

La etapa receptora cuenta con el PIC 16F628A como cerebro, el cual se encarga de recibir el dato del receptor por el pin 7, que es a su vez RB1 configurado como entrada, el programa siempre estará preguntando si hay alguna interrupción en el

pin RB1. Este pin está habilitado en el programa como interrupción, para poder así atender la transmisión de los datos enviados desde el transmisor, con este dato, el PIC 16F628A será el encargado de enviar por el puerto B, configurado como salida, los bits RB7, RB6 y RB5, los cuales serán unidades, decenas y la taquilla, para mandar el pulso a los ánodos de los display para su multiplexación con un tiempo de 5 mseg. Con esta misma información procesada con el dato recibido, mandará un código en binario por el puerto A configurado como salida por los pines RA0, RA1, RA2 y RA3.

El proyecto se estanco mucho tiempo; ya que no se lograba la correcta recepción de los datos, al principio se averiguo si había algún tipo de integrado que podía manejar la secuencia de pulsos mandada desde el transmisor pero eran escasos de conseguir y se tenía muy poca información sobre ellos, después se hicieron pruebas con la herramienta hyperterminal del computador y se veía que el transmisor si cumplía con su función de transmitir el vector de trama respectivo, entonces se llegó a la conclusión que lo mejor era cambiar el PIC 16F84A por uno que tuviera también el módulo de comunicación USART, además los mismos pines para no alterar el diseño de la tarjeta y este PIC es el 16F628A que cumple con las requisitos necesitados.

El decodificador 7448 recibe este código binario y da como salida la información necesaria para visualizar el respectivo número desplegado en los display. La corriente para mantener los segmentos iluminados es reforzada con el driver de corriente ULN 2803, el cual ayudará en la visualización. Internamente en cada segmento se hallan cinco LED conectados en serie para poder formarlo en total, y este integrado es ideal porque entrega una corriente más elevada de hasta 500 mA en caso de que sea necesario. La corriente que manejan estos display es de 160 mA con un voltaje aproximado de 10 VDC entre cada segmento y tierra.

La parte lógica del receptor está protegida por el optoacoplador 4N33, el cual está compuesto de seis pines, alimentado por 5 voltios y tiene alta sensibilidad a las bajas y altas entradas de corriente. El pin de entrada de este optoacoplador es el 1, y está en serie con una resistencia de polarización para el LED interno del optoacoplador. El optoacoplador polariza la base del transistor Darlington TIP121,

el cual les dará más corriente de entrada a los ánodos de los display 1, 2 y 3 para su óptimo funcionamiento.

Para la parte de la visualización de turnos en la parte del receptor se necesitó una caja en acrílico de color negro matizado de dimensiones 55 cm x 15 cm.

**Figura 13.** Medidas de caja en acrílico.



El software del proyecto consiste en un programa realizado en programación C, el cual consta de dos etapas: etapa transmisora y etapa receptora.

El programa del transmisor empieza en el Main, en el cual se inician algunas variables que son dígito, tm “tiempo de multiplexación” para los display del transmisor. Para evitar parpadeos se coloca una frecuencia de 60 Hz. El periodo para un display sería  $T=1/60$  Hz, y esto nos da 0,016 segundos para cada display, pero como son tres display se multiplica por tres, y esto da un tiempo de 5 ms en total aproximadamente. Otra variable es KBD, pero como el teclado va conectado al puerto D entonces se define como KBDD, que es una librería para iniciar el teclado.

Ya después se lee la dirección de la EEPROM que es la dirección 0x2100 en la posición cero. Lo que contenga se lleva a la variable *contador*, la cual nunca va a estar en cero, a su vez, siempre que se incremente la variable contador también se guarda este valor en la memoria EEPROM. Antes de entrar al ciclo infinito, el programa pregunta si algún mouse ha sido presionado y se activa una variable

llamada *configurar*, que en el lenguaje c está definida como una variable entera de 8 bits, de los cuales el compilador escoge el bit menos significativo, más propiamente definida como "integer (int)", que solamente va a tomar dos valores, ya sean "1" ó "0".

En la rutina del Main; siempre el microcontrolador estará preguntando si se ha presionado alguno de los mouse conectados al puerto D, esto lo hará cada vez que sea prendido todo el sistema; en caso que sea verdadero automáticamente irá a la función configurar.

El valor leído por el teclado se guarda en las variables K1 y K2, estos valores son en ASCII, y para poderlos convertir a un valor numérico se convierten por medio de una expresión matemática para el primer número leído por el teclado, que sería K1 (decenas), es la siguiente:

$$- X1 = K1 - 48.$$

Para el segundo número leído por el teclado, que tomaría el valor de K2 en valor ASCII, el programa en C lo convierte en su valor numérico en decimal (unidades), así:

$$- X2 = K2 - 48.$$

Después de convertir estos valores a decimal, se tienen que juntar para formar el valor, entonces las decenas tomarán su valor real aplicándoles otra ecuación, así:

$$- Decena2 = (int) (X1 * 10).$$

Al contador se le asigna otra ecuación, de manera que quede con otro valor, así:

$$- Contador = (int) (decena2 + X2).$$

Este valor que toma el contador se guarda en la memoria EEPROM en su posición de memoria (0X2100) y la función CONFIG quedará con el valor de "0".

Pero para poder enviar el dato del turno y la taquilla, se tiene que volver a separar el valor en el contador de las unidades y decenas respectivamente. Para las decenas se expresará de la siguiente forma:

- Decenas = (int) (contador/10).

Para las unidades quedará definido así:

- Unidades = (int) (contador-(decenas\*10)).

Por último se transmite el dato serialmente por medio de la función “ENVIAR”. En la función “ENVIAR” tenemos la instrucción “putc”, que es para extraer los datos por el bit de transmisión, estos datos son transmitidos en código ASCII, se tendrá en cuenta tres valores con la instrucción “putc”, así:

- putc (bandera+49) = toma el valor para la taquilla.
- putc (decenas+48) = toma el valor para las decenas (turno).
- putc (unidades+48) = toma el valor para las unidades (turno).
- putc (0x24) = carácter de envío para el cierre del vector de transmisión.

En la función “PrintContador1” se lee, de la memoria EEPROM, el conteo en el que va el contador, después se separan del contador los valores de unidades y decenas por medio de las relaciones siguientes:

Decenas = (int) (contador/10). Al dividirlo por “10”, queda un valor fraccionado por el punto decimal, pero con la función “int” solo toma la parte entera.

Unidades = (int) (contador-(decenas\*10)). Primero se multiplica por “10” el valor que tienen las decenas y después el contador lo resta, y se obtienen las unidades.

En la función Main hay una variable llamada “dígito” que empieza con el valor de cero y se incrementa cada vez que empieza una secuencia diferente para la multiplexación de los display a una frecuencia de encendido y apagado de 60 Hz; pero siempre transmitiendo los datos para los turnos.

Cuando “dígito=0”, se direcciona el puerto e con el número001 para habilitar el ánodo de unidades; luego se prenden los segmentos del mismo display con el “VectorDisplay[unidades] con su retardo de 5 ms definido por la variable “tm”.

Si “dígito=1”; entonces “output\_e”=(0x02), se activa “RE1” para encender el display de decenas.

Con “OUTPUT\_A “VectorDisplay [decenas]”; se encienden los segmentos del display para el valor de las decenas.

Si “dígito=2” y “bandera=0”; se habilita “RE0” con la instrucción “output\_e (0x1) para colocar 5 voltios en el ánodo del display de taquilla y así mostrar el número correspondiente; ya sea un “1” o un “2” con el VectorDisplay [1]; siempre indicando su tiempo de multiplexación de 5 ms.

En la función “ENVIAR” tenemos la instrucción “putc”, que es para sacar los datos por el bit de transmisión, los cuales son transmitidos en código ASCII. Para esta transmisión se tendrán en cuenta tres valores con la instrucción “putc”, así:

- putc (bandera+49) = toma el valor para la taquilla.
- putc (decenas+48) = toma el valor para las decenas (turno).
- putc (unidades+48) = toma el valor para las unidades (turno).
- putc (0x24) = carácter de envío para el cierre del vector de transmisión.

En la función *configurar* ya se había mencionado que el valor que tiene la memoria EEPROM se lleva a contador, y en K1 se guardaba el valor de la tecla presionada. Mientras “K1=0” se habilita los tres bits del puerto e con la instrucción “output\_e (0xF)” y el puerto A para visualizar la letra c. Después de enviar los datos se habilita el tercer display por medio del puerto e con la instrucción “output\_e (0x4)”. Para visualizar el dato se ingresa la instrucción “OUTPUT\_A (VectorDisplay [x2])” más un retardo de 1.500 ms.

En el receptor se tienen variables enteras como: “R”, “tm”, “dígito”, “bandera”, “uniaux”, “decaux”, “unidades”, “decenas”, “decena2”, “banaux”, “bandera2”, “contador”. También se tienen variables tipo carácter: “ch”, “ban”, “dec”, “uni”, dataRx.

Después de inicializadas todas las variables, el programa entra al "Main". En el "MAIN" se coloca el puerto A como salida con la instrucción "SET\_TRIS\_A (0x00)", se coloca el puerto B de tal manera que únicamente "RX", sea una entrada la instrucción "SET\_TRIS\_B (0b00000010)", también se habilitan las interrupciones para la recepción serial "enable\_interrupts (int\_rda)" y las interrupciones generales "enable\_interrupts (GLOBAL)". Las siguientes variables se inicializan para poder correr el programa en:

- R=0, tm=4, dígito=0, contador=0, uniaux=0, decaux=0.

Mientras lo que esté llegando sea diferente de 24 y  $R \leq 2$ , lo que el programa hará será capturar lo que llega en el vector "dataRx [R]=ch", esto es, guarda en la variable "ch en la posición R". Cada vez que llega un dato, la función "R++" se incrementa en uno hasta que ya no sea menor o igual a dos para terminar de cerrar el vector de recepción. Después de cerrar este ciclo se coloca la variable "R" en cero para la próxima recepción de datos.

Para saber qué se mandó en la posición de bandera (ban), se lee el valor dataRx [0] en la posición cero, dataRx [1] posición uno para las decenas (dec) y dataRx [2] en la posición dos para las unidades (uni). Para tener la conversión numérica de estos valores se procede con las siguientes expresiones:

- uniaux=uni-48, conversión para las unidades.
- decaux=dec-48, conversión para las decenas.
- banaux=ban-49, conversión para la bandera (taquilla).

Para empezar a configurar el turno se tiene la variable bandera2, en la cual el turno queda en "100" o en "200", así: bandera2= (int16) (banaux\*100).

Para ver el contador se tiene la función "Printcontador", de la cual se mostrarán los datos, para eso hay que preguntar por las variables "banaux=2" y dígito=0; si se cumplen se habilita RE7 y a su vez se dará salida al puerto A para los segmentos del display de unidades con la variable "uniaux".

Cuando “dígito=0”, se coloca esta variable en uno, después se enciende el bit 7 del puerto e mediante “output\_b (0b10000000)”, se visualiza el dato en el display de unidades con el vector display de unidades “OUTPUT\_A (VectorDisplay [unidades]) más el retardo de 4 msg para mostrar las unidades.

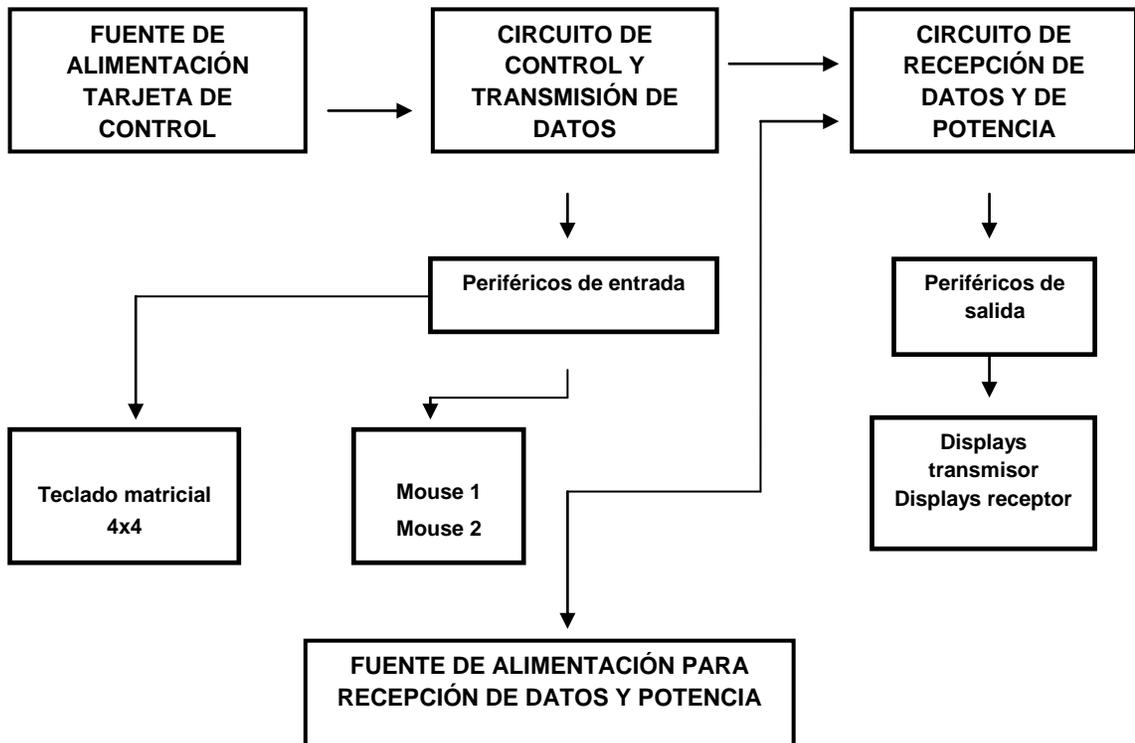
Para “dígito=1”, se coloca esta variable en dos, después se enciende el bit 6 del puerto e mediante la función “output\_b (0b01000000)”, se visualiza el dato en el display de decenas “OUTPUT\_A (VectorDisplay [decenas])”, con el respectivo retardo de 4 msg. Como la variable dígito es igual a 2, se pregunta si la variable bandera es igual a cero, si esta condición se cumple, la variable dígito volverá a comenzar en cero para volver a iniciar el ciclo.

Lo más importante de todo este proyecto son los logros realizados en el diseño electrónico y en la realización del organizador de turnos inalámbrico ya que fue innovador en la aplicación de un módulo transmisor y receptor de radiofrecuencia, además de la integración de todos los elementos explicados en el marco teórico y algunos que no se mencionaron ya que fueron vistos durante la carrera y solamente se hizo ahincó en los que se creían más importantes.

También hubo más logros como la implementación del lenguaje C como una alternativa en la programación de microcontroladores PIC; ya que es un lenguaje sin muchas instrucciones y fácil de entender, además los programas son más cortos.

Otro punto importante es la aplicación de todo lo aprendido y visto en la carrera en un proyecto real y que se pueda aplicar en la industria como alternativa en los organizadores de turnos y hasta de pronto en un futuro no muy lejano formar una microempresa relacionada con este tema.

**Figura 14.** Diagrama de bloques de los circuitos del organizador de turnos.



## 7. CONCLUSIONES

Se creó un producto de buena calidad y a un costo competitivo si se lo compara con los que ofrece el mercado. El organizador de turnos inalámbrico cumplió con las expectativas generadas como: la implementación de la transmisión inalámbrica por radiofrecuencia para el envío de los datos, la programación en lenguaje c para el funcionamiento del proyecto, el fácil uso del teclado para la programación de los turnos y la fácil instalación para la visualización de la taquilla y el turno.

Se logró diseñar las fuentes de alimentación con gran éxito, ya que ninguna presentó problemas de calentamiento ni cortocircuitos, y lo más importante: pudieron soportar la carga necesaria para las diferentes etapas que se diseñaron, que en total fue de 250 mA para el transmisor y de 400 mA para el receptor, ya que cada fuente tiene la capacidad de entregar hasta 2 amperios DC sin ningún problema.

Se logró la elaboración de las tarjetas electrónicas, para la transmisión y la recepción de datos, bajo los requerimientos técnicos del programa computarizado "EAGLE", del cual se obtuvo mucho conocimiento tanto en la parte teórica como en la práctica del diseño electrónico, ejemplos: diseño esquemático de los circuitos, diseño del board-schematic de los componentes y su respectiva ubicación en el impreso, y correcta utilización de las librerías de cada una de los elementos utilizados.

Con la integración del módulo transmisor y receptor de radiofrecuencia se logró un ahorro considerable de cableado entre el transmisor y el receptor, además de la confiabilidad en el envío de la información y en el alcance entre estas dos etapas, ya que se hicieron pruebas como ubicar el transmisor en un primer piso y el receptor en un segundo piso sin que se afectara ningún dato enviado.

## 8. RECOMENDACIONES

El módulo Transmisor - receptor de radio frecuencia que se utilizó no es tan confiable como se creía, debido a que presenta unas fluctuaciones con señales parásitas (ruido), pero se podría pensar en otros módulos híbridos que ofrece el comercio que, aunque más costosos, son de mayor confiabilidad.

Por tratarse de un sistema de transmisión y recepción por radiofrecuencia, se deberán implementar fuentes más especiales contra ruidos e interferencias, ya que las utilizadas tuvieron que someterse a un tratamiento especial, lo que obligó a incluir algunos elementos que no hacían parte del proyecto inicial. Se recomienda, por lo anterior, utilizar tarjetas con malla a tierra y pistas de tierra más gruesas.

En el receptor se tuvieron muchas complicaciones, ya que se estaba trabajando con el PIC 16F84 para tratar de establecer la comunicación inalámbrica entre el módulo receptor y el PIC por medio de un tren de pulsos, pero se llegó a la conclusión que era mejor un PIC con transmisión y recepción de información para poder efectuar el envío de datos.

La programación del PIC se realizó en un lenguaje de alto nivel, como lo es el lenguaje C, el cual a la hora programar ofrece economía sintáctica, control de flujo, estructuras sencillas y un buen conjunto de operadores para realizar cualquier programa. Este tipo de programación debería ser tenida más en cuenta en la carrera como alternativa al básico y clásico lenguaje ensamblador.

Por estética se recomienda un acrílico negro matizado para permitir que en la parte del receptor sobresalgan los segmentos de los display, así mismo, el tamaño de éstos fue el adecuado para permitir una excelente visualización de los números para la taquilla y el turno.

Al momento de utilizar el mouse como accesorio para el incremento de los turnos, hubo que quitarle todos sus elementos internos, ya que el pulsador interno que tenía no dio los resultados esperados, porque presentaba más impedancia de lo normal, alrededor de 2.5 a 3.5  $\Omega$ .

## BIBLIOGRAFÍA

ANGULO USATEGUI, José María; ROMERO YESA, Susana y ANGULO MARTÍNEZ, Ignacio. Microcontroladores PIC. Diseño práctico de aplicaciones. 2.<sup>a</sup> ed. España: McGraw-Hill/Interamericana S.A., 1999.

BOYLESTAD, Robert L. y NASHELSKY, Louis. Electrónica: teoría de circuitos y dispositivos electrónicos. 8.<sup>a</sup> ed. México: Pearson Educación, 2003.

DUQUE, Edison. Curso Avanzado de Microcontroladores PIC. Compañía Editorial Tecnológica, 2002.

FLOYD, Thomas L. Fundamentos de sistemas digitales. 9.<sup>a</sup> ed. Madrid: Pearson Educación S.A., 2006.

MONTOYA DÍAZ, Luis Guillermo. Microcontroladores PIC. Medellín: Edilatinas, 2007.

PERRY, Greg. C con Ejemplos. Prentice Hall, 2000.

ROJAS, William. Curso fácil de electrónica básica. Colombia: Cedit S.A., 1998.

SABINO, Carlos A. Cómo hacer una tesis y elaborar todo tipo de proyectos. Colombia: Panamericana Editorial, 1996.

STREMLER G., Ferrel. Sistemas de comunicación. Massachusetts. USA: Addison Wesley publishing Company. Fondo Educativo Interamericano, 1989.

TOCCI, Ronald. Sistemas Digitales: Principios y Aplicaciones. 6.<sup>a</sup> ed. México: Prentice-Hall Hispanoamericana, 1996.

## CIBERGRAFÍA

[http://es.wikipedia.org/wiki/Diagramas de flujo](http://es.wikipedia.org/wiki/Diagramas_de_flujo)

<http://www.mis-algoritmos.com/diagramas-flujo.html>

[http://www. Datasheetcatalog.com](http://www.Datasheetcatalog.com)

<http://www.fairchildsemi.com>

[http://www.datasheetcatalog.org/datasheets/228/390068\\_DS.pdf](http://www.datasheetcatalog.org/datasheets/228/390068_DS.pdf)

<http://www.sparkfun.com/datasheets/Wireless/General/MO-RX3400.pdf>

<http://sparkfun.com/datasheets/Wireless/General/MO-SAWR.pdf>

<http://ww1.microchip.com/downloads/en/devicedoc/30292c.pdf>

<http://es.wikipedia.org/wiki/PIC16F84>

<http://ww1.microchip.com/downloads/en/devicedoc/35007.pdf>

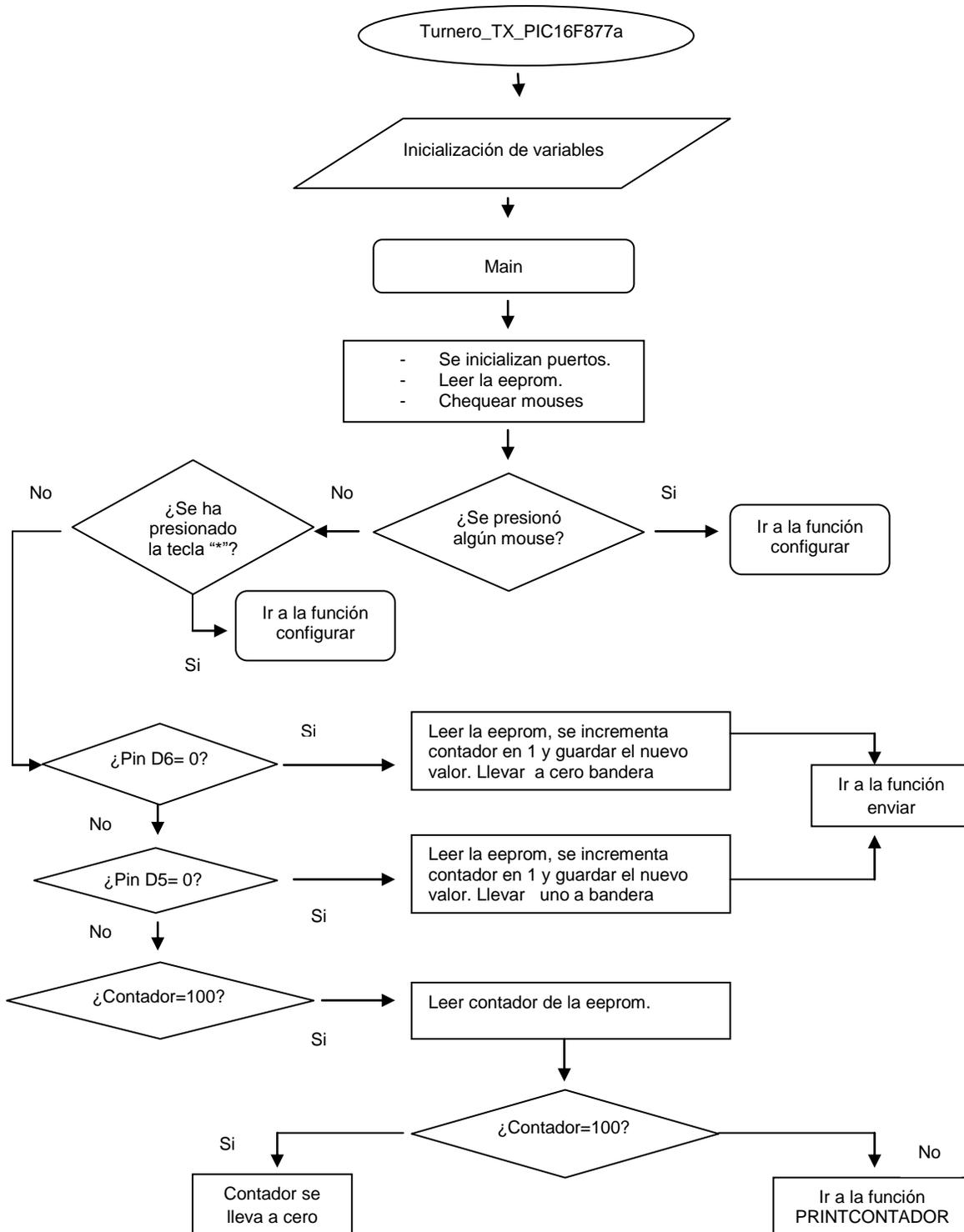
<http://www.monografias.com/trabajos10/formulac/formulac.shtml>

[http://www.itpuebla.edu.mx/Oferta\\_Educativa/sistemas/TESIS/PDF/Como%20se%20hace%20una%20introducci%C3%B3n.pdf](http://www.itpuebla.edu.mx/Oferta_Educativa/sistemas/TESIS/PDF/Como%20se%20hace%20una%20introducci%C3%B3n.pdf)

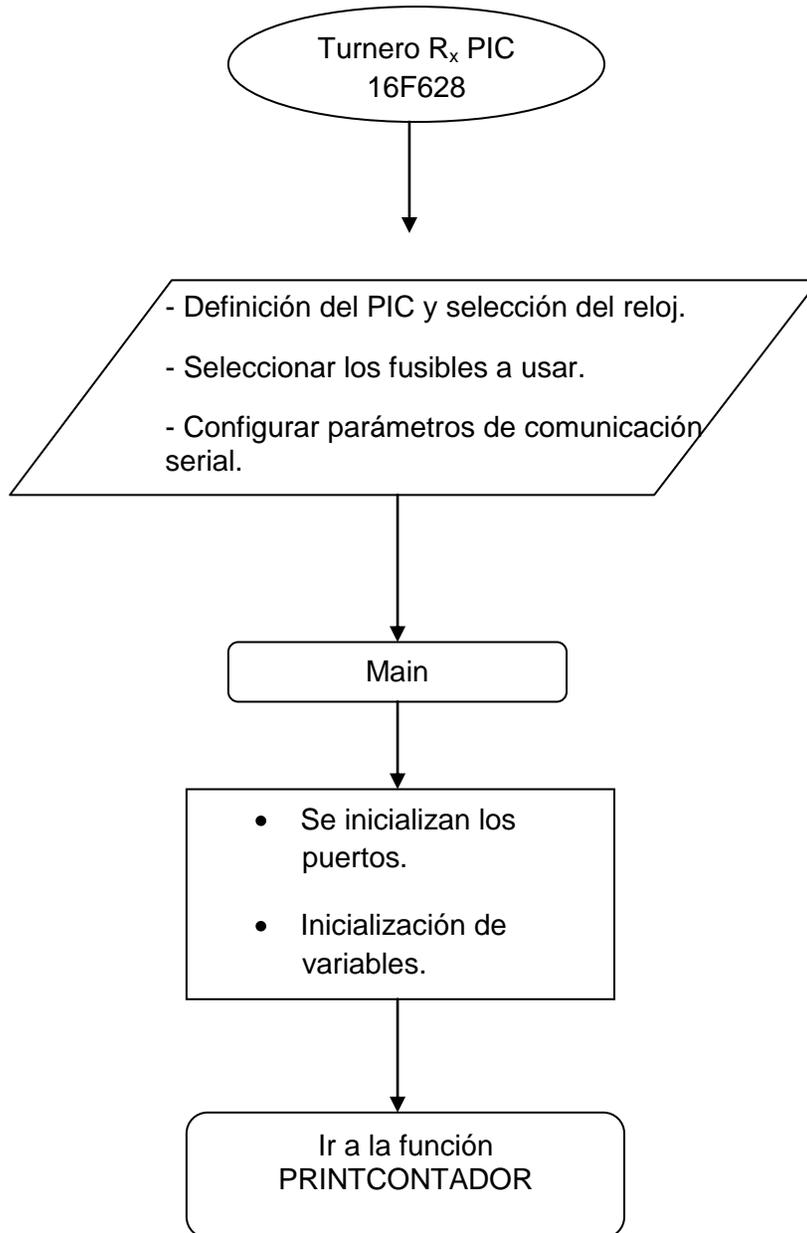
<http://www.circuitoselectronicos.org/2011/01/el-pic-16f628.html>

<http://www.ziddu.com/download/5597452/DATASHEETSESPAOLPARTES.rar.html>

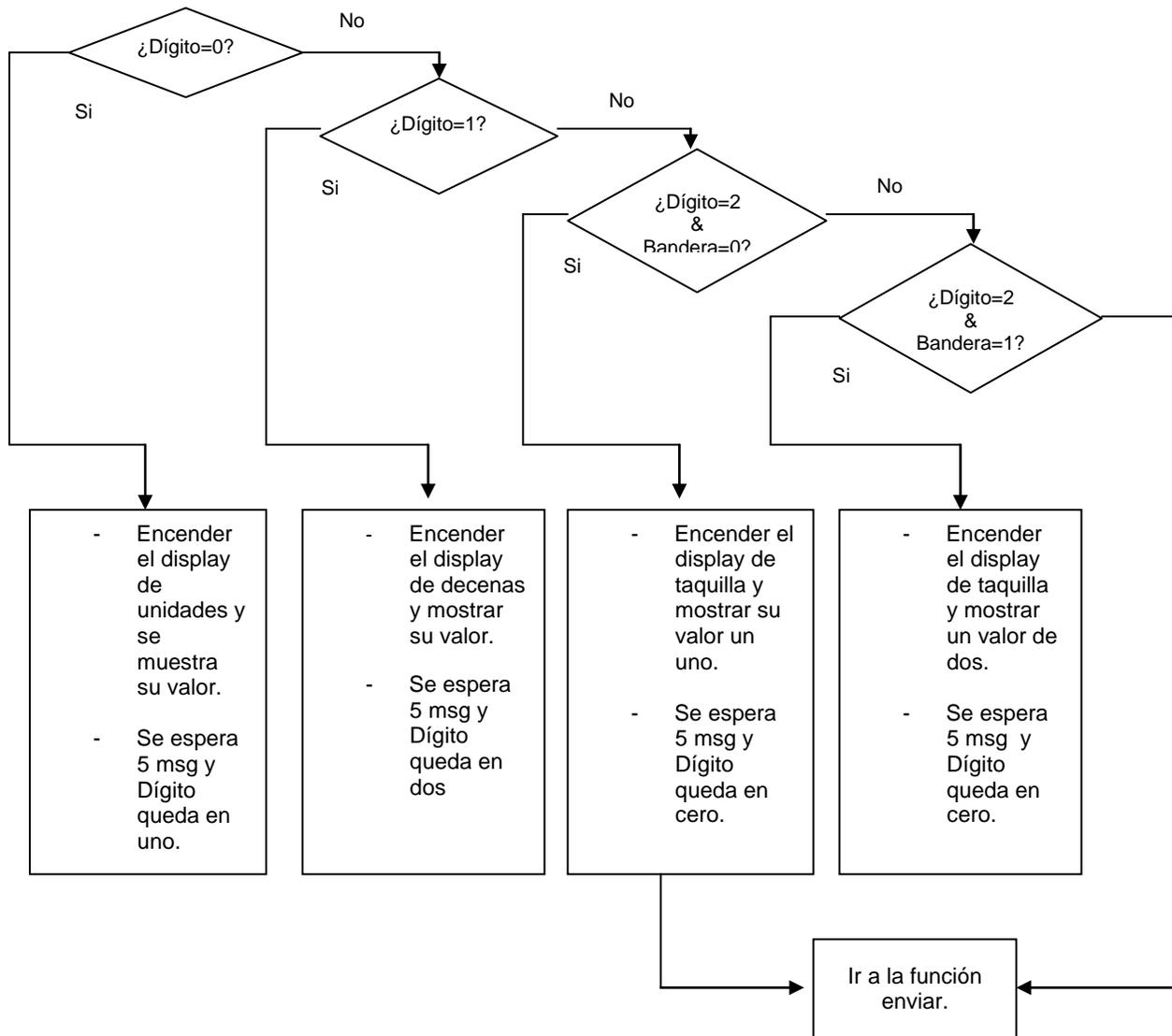
**Anexo A. Diagrama de flujo etapa de transmisión y visualización de datos.**



**Anexo B.** Diagrama de flujo etapa de recepción y visualización de datos.



Anexo C. Diagrama de flujo "PRINTCONTADOR" (Función de multiplexación).



## Anexo D. Programa en C de transmisión y visualización de datos.

```
/******  
* TURNERO  
* MÓDULO TRANSMISOR  
* Definición: Pascual Bravo.  
*  
* by:  
* FRANCINY  
*****/  
  
#include <16F877a.h> //se selecciona el PIC a trabajar  
#USE DELAY( CLOCK=4000000) //valor cristal a usar  
#FUSES XT,NOWDT,NOPROTECT,NOPUT  
#use rs232(baud=2400, xmit=PIN_C6, rcv=PIN_C7)  
#include <kbdd_4x4.c> //invocamos la librería para teclado.  
#rom 0x2100={0x0} //es la dirección de memoria EEPROM interna.  
#USE standard_io(E) //el puerto E será de entrada o salida.  
#USE standard_io(D) //el puerto D será de entrada o salida.  
#USE fast_IO (A) //configuración de puerto.  
#BYTE PORTA=0x05  
byte CONST VectorDisplay[10] = {0x0,0x1,0x2,0x3,0x4,0x5,0x6,0x7,0x8,0x9};  
int tm, digito, contador=0, decenas, x1, x2, unidades, decena2, config=0, bandera=0;  
char k1, k2;  
void enviar(void) // contiene la trama de datos que se envían.  
{  
    putc(bandera+49); //envió dato 1.  
    putc(decenas+48); //envió dato 2.  
    putc(unidades+48); //envió dato 3.  
    putc(0x24); //fin de trama.  
}  
void enviar2(void) // se muestra una t-c-c.  
{  
    putc(51); //envio dato 1 //para visualizar la t.  
    putc(64); //envio dato 3 // posición para visualizar la c.  
    putc(0x24); //fin de trama.  
}  
void enviar3(void)  
{  
    putc(51); //envio dato 1 // para visualizar la t.  
    putc(x1+48); //envió dato 2.  
    putc(64); //envio dato 3 //posición para visualizar la c.  
    putc(0x24); //fin de trama.  
}  
void enviar4(void)  
{
```

```

putc(51); //envio dato 1           // posición para visualizar la T.
putc(x1+48);                          //envió dato 2; primer número.
putc(x2+48);                          //envió dato 3; segundo número.
putc(0x24);                          //fin de trama.
}
void configurar(void)                //configuración turno.
{
    contador=read_eeprom(0); //lee la EEPROM.
    k1=kbd_getc();            //en k1 guarda el valor de la tecla.
    while (k1==0)            //ciclo infinito.
    {
        k1=kbd_getc();
        output_e(0xF);      //se habilita el display.
        OUTPUT_A(0b1010);  //es la combinación para visualizar la "c."
        enviar2();
    }
    if (k1!=0)                //cuando presionan la primera tecla.
    {
        k2=kbd_getc();
        x1=k1-48;            //conversión numérica.
        output_e(0x2);
        OUTPUT_A(VectorDisplay[x1]); //valor display.
        enviar3();
        while(k2==0)        //segunda tecla presionada.
        {
            k2=kbd_getc();
            enviar3();
        }
        x2=k2-48;           //se hace la conversión numérica.
        enviar4();
        output_e(0x4);
        OUTPUT_A(VectorDisplay[x2]);
        delay_ms(1500);    //retardo para ver números.
    }
    x1=k1-48;                //se asegura la conversión numérica.
    x2=k2-48;
    decena2=(int)(x1*10);    //valor de contador en dos cifras.
    contador=(int)(decena2+x2);
    write_eeprom(0x2100,contador);
    config=0;
    decenas = (int)(contador/10);
    unidades = (int)(contador-(decenas*10));

    enviar();                //transmito el valor por serial.
}
void PrintContador1(void)        //se imprime contador.
{
    contador=read_eeprom(0);    //leo el valor de la EEPROM.
    decenas = (int)(contador/10);

```

```

    unidades = (int)(contador-(decenas*10));
if (digito==0)
    {
    output_e(0x4);                //display de unidades en ON.

    OUTPUT_A(VectorDisplay[unidades]);
    delay_ms(tm);                //retardo para evitar parpadeos.

    digito=1;                    //prosigo al siguiente display
    }
if (digito==1)
    {
    output_e(0x2);                //enciendo únicamente el display de decenas.
    OUTPUT_A(VectorDisplay[decenas]); //tiro el valor a mostrar "decenas.
    delay_ms(tm);                //espero un tiempo tm=5ms.
    digito=2;                    //prosigo al siguiente display.
    }
if ( (digito==2) && (bandera==0))
    {
    output_e(0x1);                //enciendo únicamente el display de taquilla.
    OUTPUT_A(VectorDisplay[1]); //tiro el valor a mostrar en taquilla.
    delay_ms(tm);                //espero un tiempo de 5 ms.
    digito=0;                    //prosigo al siguiente display.
    }
else if ( (digito==2) && (bandera==1))
    {
    output_e(0x1);                //enciendo display taquilla.
    OUTPUT_A(VectorDisplay[2]); //tiro el valor a mostrar en
                                //taquilla que es 2(bandera).
    delay_ms(tm);                //espero un tiempo tm=5ms para evitar
                                //parpadeos.
    digito=0;
    }
    enviar();
}

void main()
{
SET_TRIS_A(0x00);                // se configura el puerto a como salida.
port_b_pullups(true);           //activamos los resistencias de pull-ups.
//se inicializan variables.
digito=0;
tm=5;
kbd_init();
//Cargamos en la variable contador el valor albergado en la EEPROM, nunca arranca en
cero (100)
contador=read_eeprom(0);
enviar();                        //enviar por el puerto serial.
}

```

```

if( (input(pin_D6)==0)||(input(pin_D5)==0) ) config=1;
else if( (input(pin_D6)==1)&&(input(pin_D5)==1) ) config=0;
for( ;; )
{
    if (config==1) //si config=1 se va a configurar.
    {
        configurar(); //caracteres de configuración.
    }
    else if(config==0) //si config=0 no se configura el turno
    {
        k1=kbd_getc(); //se almacena la tecla presionada.
        while ((k1==0) || (k1!='*'))
        {
            k1=kbd_getc(); //guarda el valor de la tecla en k1.
            if(input(pin_D6)==0) //se presiono el mouse 1.
            {
                delay_ms(200); //retardo de anti rebote.
                contador=read_eeprom(0); //se pregunta por el valor de la EEPROM.
                bandera=0; //se lleva a cero la variable bandera.
                contador=contador+1; //incremento en 1 el valor del contador.
                write_eeprom(0x2100,contador); //escribo el valor en la EEPROM.
                decenas = (int)(contador/10); //para obtener las decenas.
                unidades = (int)(contador-(decenas*10));
                enviar(); //enviamos por serial mediante RF,
                //los datos decenas y unidades.
            }
            if(input(pin_D5)==0)
            {delay_ms(200);
            contador=read_eeprom(0);
            bandera=1;
            contador=contador+1;
            write_eeprom(0x2100,contador);
            decenas = (int)(contador/10);
            unidades = (int)(contador-(decenas*10));
            enviar();
            }
        }

        if(contador<=100) //pregunta cuando contador es ≤100.
        {
            contador=read_eeprom(0);
            if(contador==100) //si el contador=100 se lleva a cero.
            {
                contador=0; //llevamos el contador a 0.

                write_eeprom(0x2100,contador); //guardamos en la EEPROM.
            }
        }
        else PrintContador1(); //va a imprimir (visualizar) el valor del contador.
    }
}

```

## Anexo E. Programa en C de recepción y visualización de datos.

```
*****
* TURNERO *
* MÓDULO RECEPTOR *
* Definición: Pascual Bravo. *
* *
* by: *
* Tecn: Franciny Cano *
*****/

#include <16F628a.h>
#USE DELAY( CLOCK=4000000)
#FUSES XT,NOWDT,NOPROTECT,NOPUT
#use rs232(baud=2400, xmit=PIN_B2, rcv=PIN_B1)
#USE fast_IO (A)
#BYTE PORTA=0x05
#USE fast_IO (B)
#BYTE PORTB=0x06
#USE fast_IO (A)
#BYTE PORTA=0x05
#USE fast_IO (B)
#BYTE PORTB=0x06
byte CONST VectorDisplay[17] = {0x0,0x1,0x2,0x3,0x4,0x5,0x6, 0x7,0x8,
                                0x9,0x0,0x0,0x0,0x0,0xE,0xA,0xA};

volatile char dataRx[3];
volatile int R=0,tm, digito=0, bandera=0, uniaux, decaux, unidades, decenas,
decena2;
volatile int16 banaux, bandera2, contador=0;
char ch, ban, dec, uni;
putc(banaux+49);
putc(decaux+48);
putc(uniaux+48);
putc(0x24);

if (banaux==2) //es para cuando estamos configurando el turno.
{
    if (digito==0)
    {
        digito=1;
        //output_b(~0x80);
        output_b(0b10000000); // rb7 encendido
        OUTPUT_A(VectorDisplay[uniaux]);
        delay_ms(tm);
    }

    if (digito==1)
    {
```

```

    digito=2;
    //output_b(~0x40);
    output_b(0b01000000); //rb6 encendido
    OUTPUT_A(VectorDisplay[decaux]);
    delay_ms(tm);
}
if (digito==2)
{
    digito=3;
    //output_b(~0x20);
    output_b(0b00100000); //rb5 encendido
    OUTPUT_A(VectorDisplay[14]);
    delay_ms(tm);
}
if (digito==3)
{
    digito=0;
    output_b(0x00);           //se genera el efecto intermitente en los displays
    OUTPUT_A(0xF);
    delay_ms(400)
}
}
else
}
bandera = (int)(contador/100);
decenas = (int)(contador-(bandera*100))/10;
unidades = (int)(contador-(bandera*100)-(decenas*10));
if (digito==0)
{
    digito=1;
    //output_b(~0x80);
    output_b(0b10000000); //rb7 encendido
    OUTPUT_A(VectorDisplay[unidades]);
    delay_ms(tm);
}

if (digito==1)
{
    digito=2;
    //output_b(~0x40);
    output_b(0b01000000); //rb6 encendido
    OUTPUT_A(VectorDisplay[decenas]);
    delay_ms(tm);
}

if ((digito==2) && (bandera==0)) //para encender el display de la taquilla.
{
    digito=0;
    //output_b(~0x20);

```

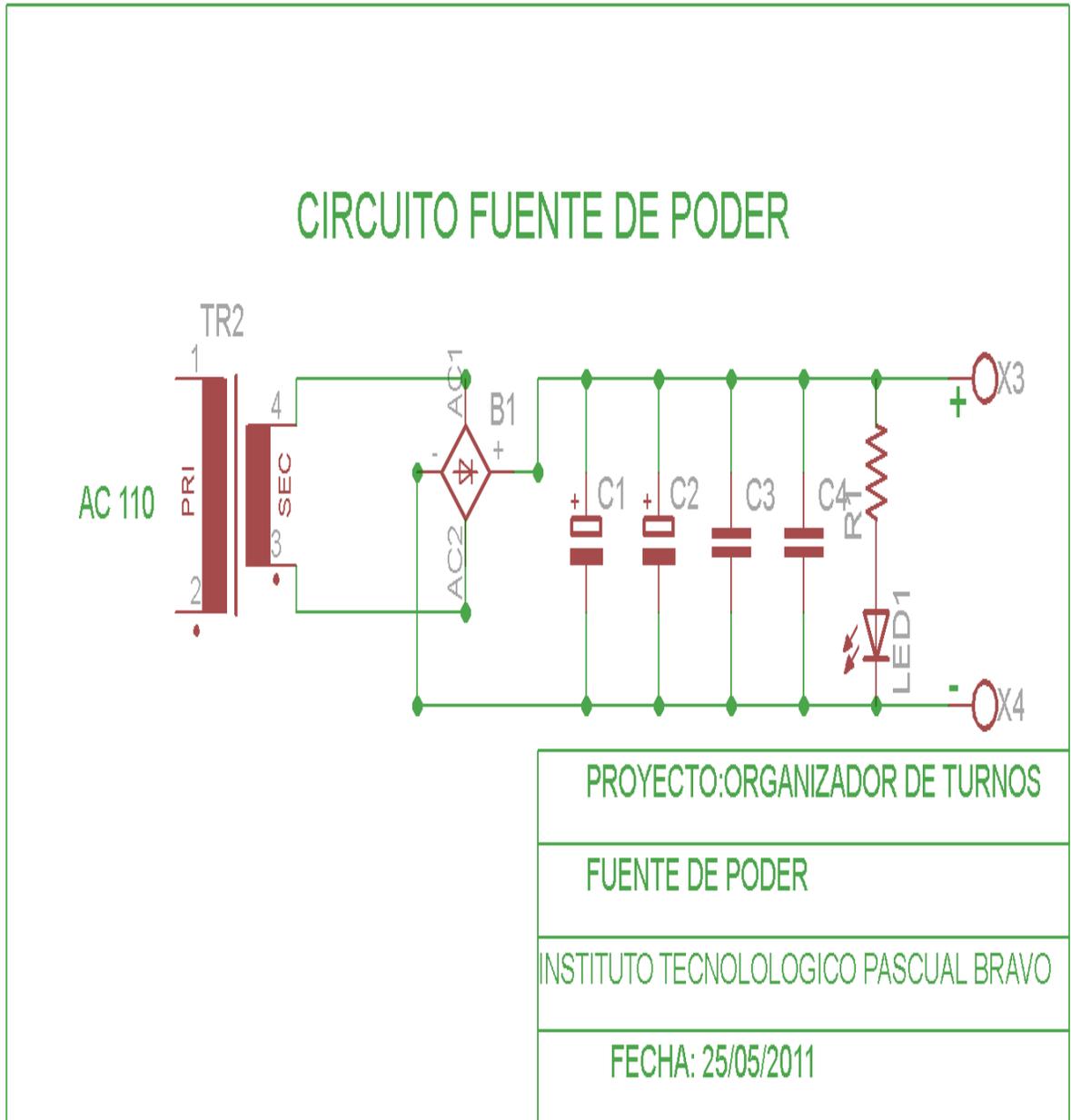
```

    output_b(0b00100000);          //rb5 encendido.
    OUTPUT_A(VectorDisplay[1]);    //con este formamos los números.
    delay_ms(tm);
}
if ((digito==2) && (bandera==1))
{
    digito=0;
    //output_b(~0x20);
    output_b(0b00100000); //rb5 encendido
    OUTPUT_A(VectorDisplay[2]);
    delay_ms(tm);
}
}
}
#INT_rda
void serial_isr()
{
    ch=getchar();                  //guardo lo recibido.

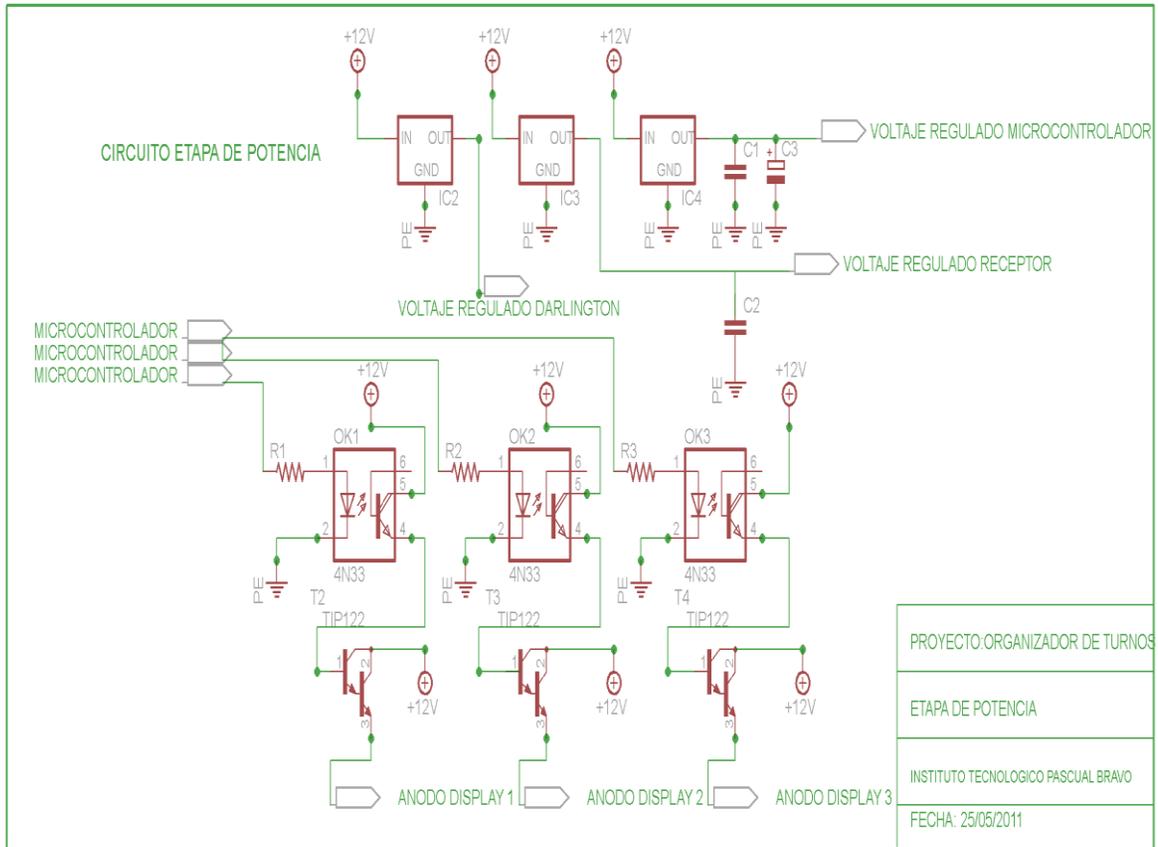
    if((ch!=0x24)&& (R<=2)) //vector de tres datos.
    {
        dataRx[R]=ch;
        R++;                      // "R=R+1"
    }
    else
    {
        R=0;                      //lleva a cero la variable r.
        ban=dataRx[0];            //la primera posición es bandera.
        dec=dataRx[1];            //la segunda posición es decena.
        uni=dataRx[2];            //la tercera posición es unidades.
        uniaux=uni-48;            //se hace la conversión numérica.
        bandera2=(int16)(banaux*100); //se forma el 100 o el 200.
        decena2=(int)(decaux*10); //se multiplica por 10 decenas.
        contador=(int16)(bandera2+decena2+uniaux); //se forma el turno.
    }
}
void main()
{
    SET_TRIS_A(0x00);             //todo el puerto a como salida.
    SET_TRIS_B(0b00000010);      //"RX" RB1 sea una entrada.
    enable_interrupts(int_rda);   //habilito recepción serial.
    enable_interrupts(GLOBAL);    //habilito las interrupciones en general.
    R=0, tm=4, digito=0, contador=0, uniaux=0, decaux=0;
    for(;;)
    {
        Printcontador();
    }
}

```

Anexo F. Esquemático fuente de alimentación.

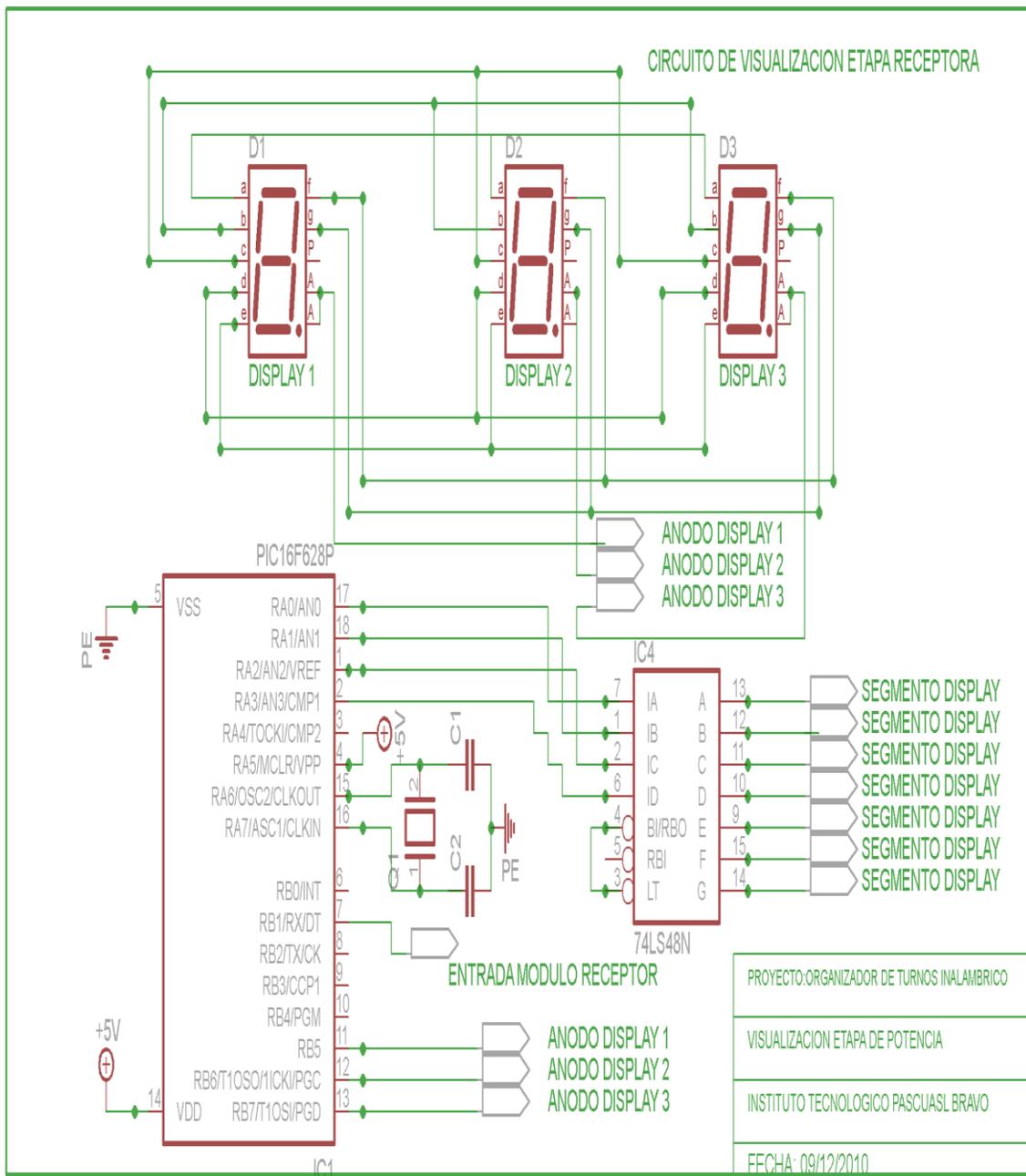


**Anexo G. Esquemático alimentación de los ánodos de los display del receptor.**

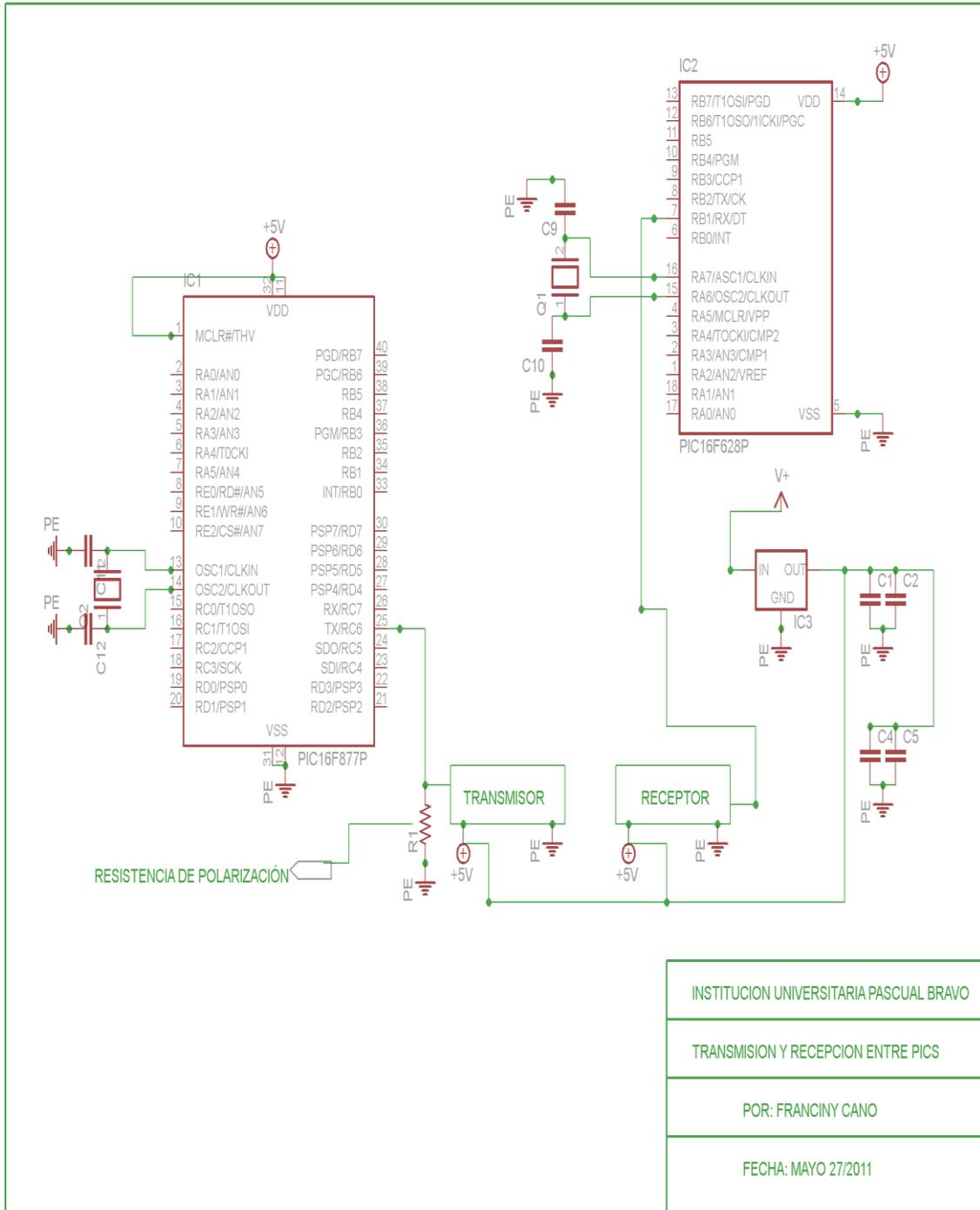


En el Anexo G se puede ver la conexión de tres display de ánodo común para poder visualizar un respectivo número, acompañada del PIC 16F628A que indica la habilitación mediante un pulso de 5 voltios DC; se emplea además el integrado 74LS48, el cual, mediante una entrada binaria suministrada por el microcontrolador, sacará por alguna de sus salidas los segmentos que se verán en el display.

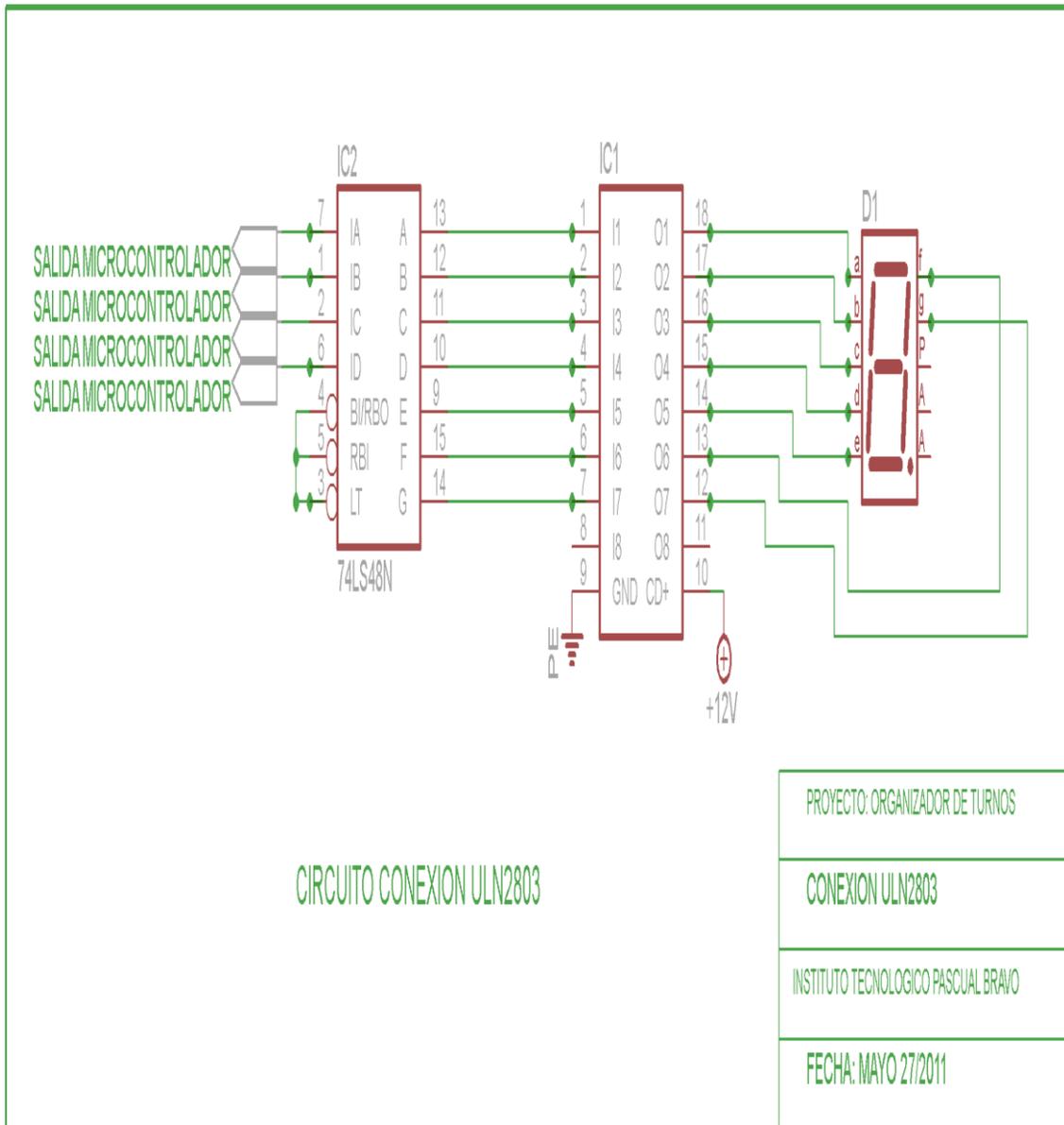
## Anexo H. Esquemático circuito de visualización receptor de datos.



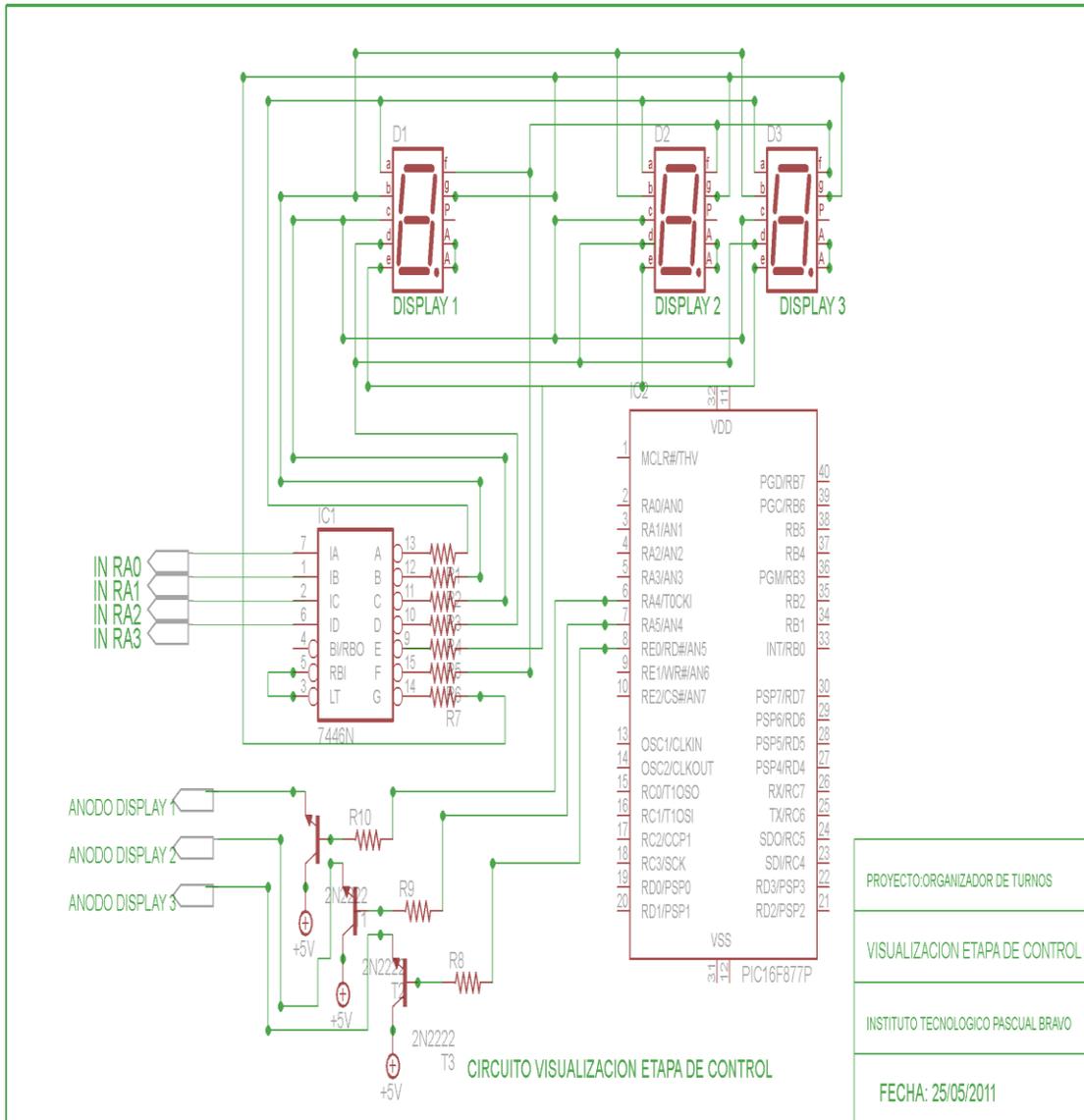
## Anexo I. Esquemático etapa transmisión y recepción de datos.



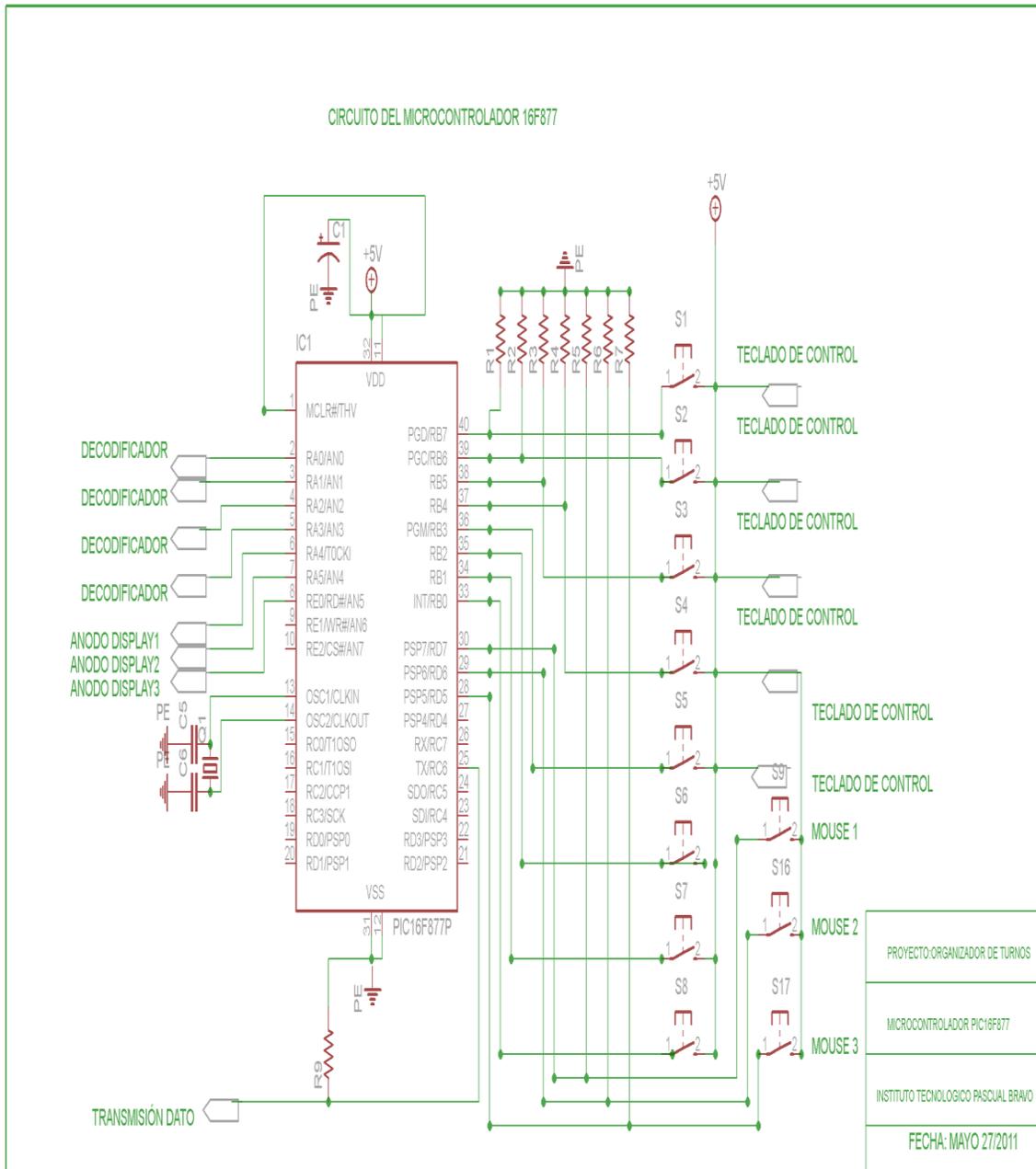
**Anexo J. Esquemático conexión ULN 2803.**



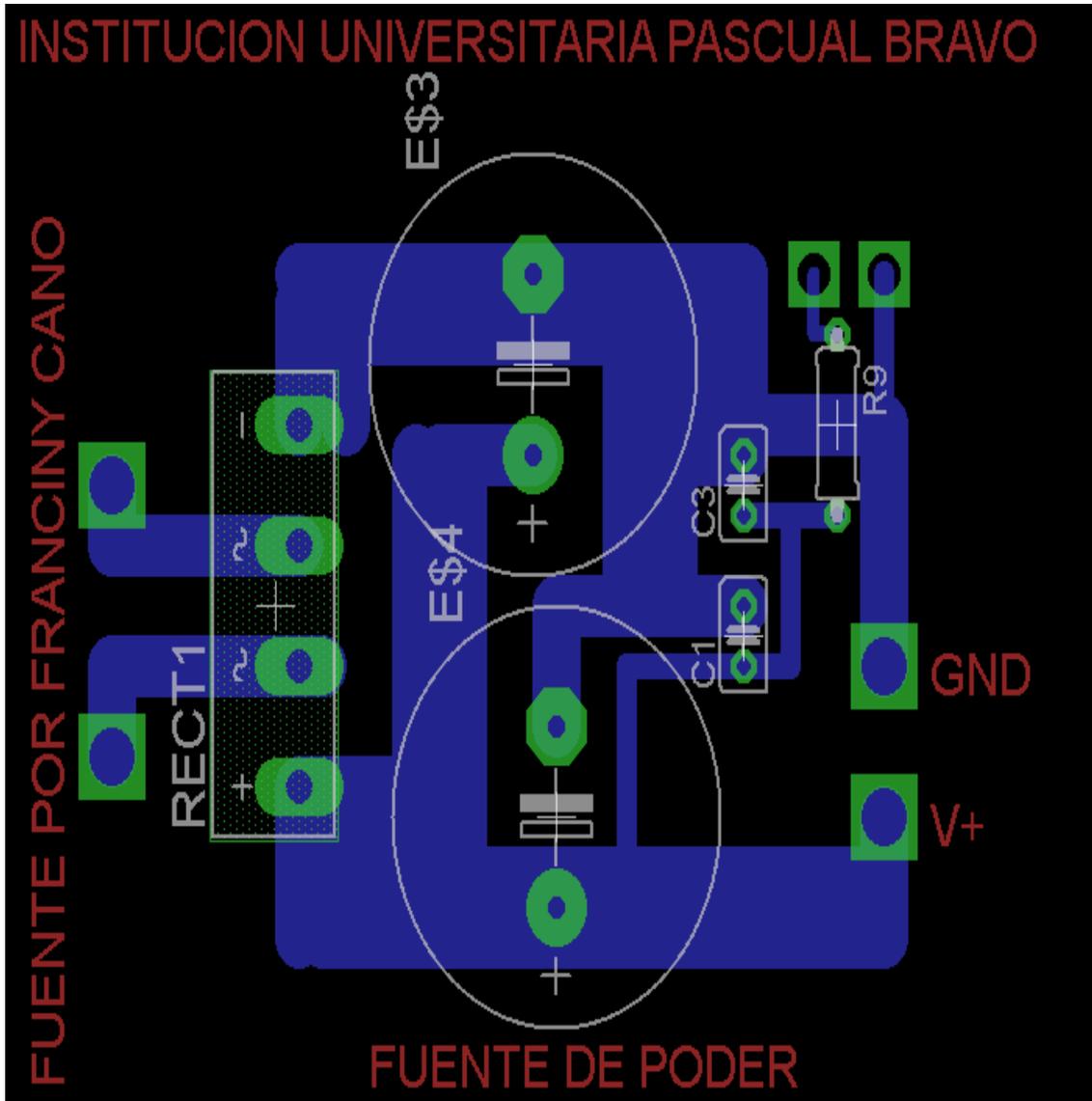
## Anexo K. Conexión PIC 16F877 y displays de la etapa transmisora.



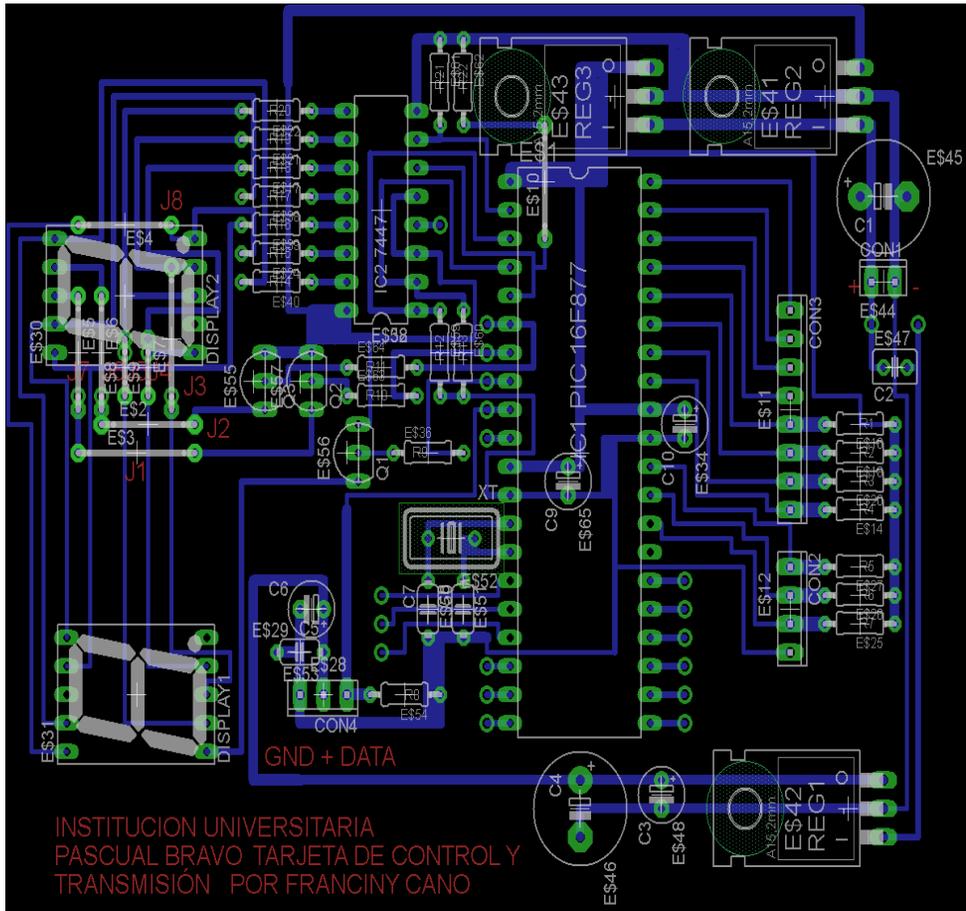
## Anexo L. Conexión PIC 16F877 con el teclado 4x4.



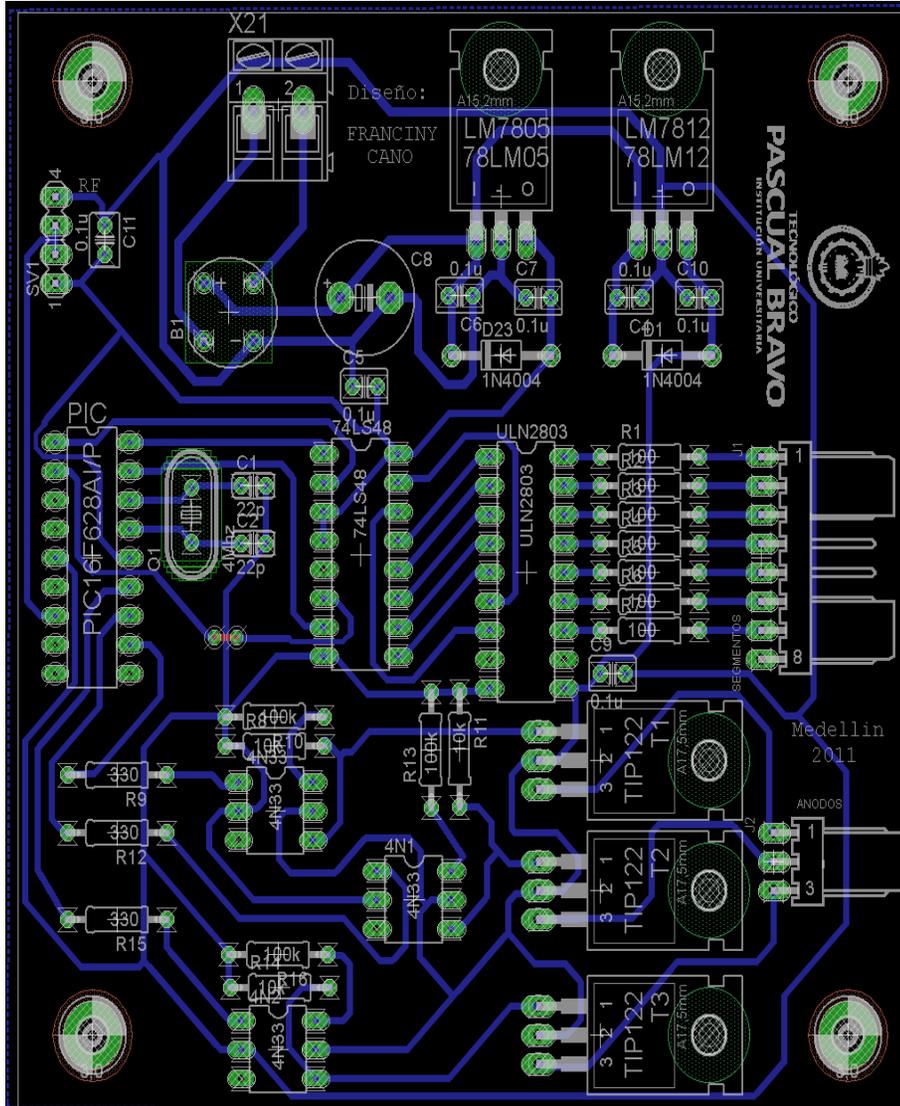
Anexo M. Circuito impreso fuente.



Anexo N. Circuito impreso etapa transmisora.



Anexo O. Circuito impreso etapa receptora.



Anexo P. Fotografía proyecto de grado.



Anexo Q. Fotografía etapa de control en funcionamiento.



Anexo R. Fotografía etapa receptora en funcionamiento.

