**UNIVERSIDAD NACIONAL DE CORDOBA**

Facultad de Ciencias Exactas, Físicas y Naturales



**Proyecto Final**

**Electrónica Digital II**

GRUPO N° 11.

INTEGRANTES:

AGUERREBERRY, MATTHEW (93.739.112) (ICOMP)

TROMBOTTO, AGUSTÍN (39.071.116) (ICOMP)

DOCENTE: Ing. Carmen Rodríguez.

AÑO: 2016.

## Introducción

El proyecto final de la materia consiste en un cubo de Leds de 4x4x4 con su correspondiente hardware y software para manejar su funcionamiento. Se utilizó un microcontrolador PIC 16F887 y el entorno de desarrollo (IDE) para assembler: MPLAB X IDE v3.26.

## Descripción del Proyecto

La funcionalidad del sistema es utilizar la comunicación serie para mandar el número de secuencia a realizar, procesar este dato en el PIC y reproducir la dicha secuencia en el cubo. A continuación se muestra el número correspondiste a cada secuencia:

|  |  |
| --- | --- |
| Numero de Secuencia | Nombre de Secuencia |
| 0 | PRENDE\_TODO\_CUBO |
| 1 | PRENDE\_1x1 |
| 2 | PRENDE\_PERIMETRO |
| 3 | PRENDE\_COLUM |
| 4 | PRENDE\_BOLICHE |

**PRENDE\_TODO\_CUBO:** enciende todo los leds del cubo

**PRENDE\_1x1:** enciendo un led a la vez, rotando del bit menos significativo al más significativo, un led por piso

**PRENDE\_PERIMETRO:** enciende el perímetro del cubo

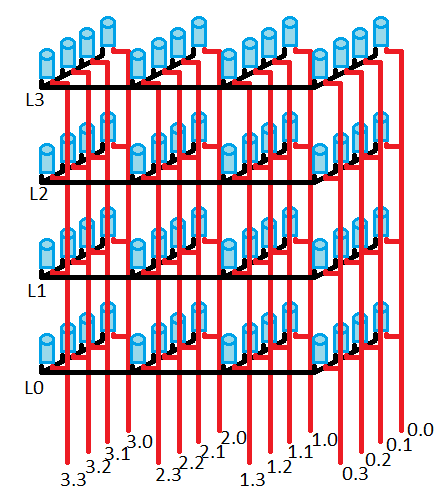
**PRENDE\_COLUM:** enciende una columna a la vez, variando desde el bit0 al bit7

**PRENDE\_BOLICHE:** enciende y apaga todo los leds del cubo cada un segundo

## Especificaciones Técnicas

### 3.1 Armado del Cubo

Como muestra la siguiente imagen, la conexión de los leds es cátodo común entre los del mismo piso, mientras que por columna los leds están conectados de forma ánodo común.



### 3.2 Conexión Cubo - PIC

Los pines RC0 a RC3 se conectan a la base de un transistor NPN cuyo emisor se conecta a masa y su colector a L0-L3 según corresponda. Los pines de los puertos A y D irán conectados a la base de otro transistor NPN, donde su emisor se conecta al ánodo del Led y su colector se conecta a VDD. Este segundo transistor fue utilizado para que el led demande corriente de la fuente de alimentación y no del PIC, evitando así dañar por corriente el dispositivo si se setea simultáneamente todo el puerto A y D.

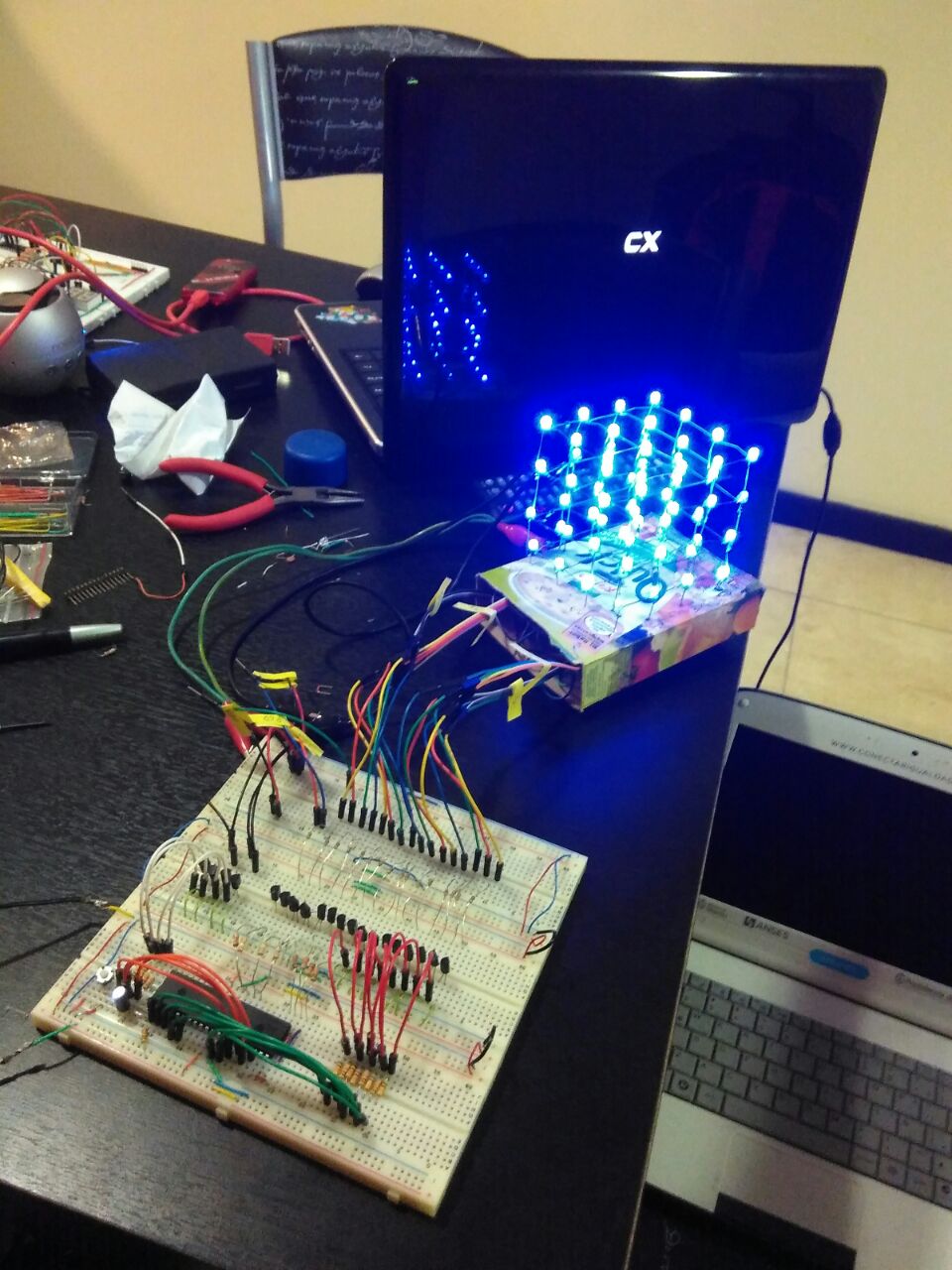


La imagen anterior es un ejemplo de la conexión de un solo Led, los 64 leds del cubo poseen la misma conexión. Esto nos permite controlar el comportamiento de cada led, a través del puerto C elegimos el piso, y con el puerto A y D seleccionamos que led prender de dicho piso. Esto nos da un total de 20 transistores NPN, 4 para los cátodos y 16 para los ánodos.

### 3.2 Conexión PIC- Computadora

Para la conexión de puerto serie utilizamos la programadora PICkit2. Esta trae incorporada la herramienta “UART Tool” que nos permite establecer una comunicación serie entre el PIC y la computadora.

En la siguiente imagen vemos como quedo el circuito construido.



## Código Fuente

; PIC16F887 Configuration Bit Settings

; ASM source line config statements

#include "p16F887.inc"

; CONFIG1

; \_\_config 0xEFF4

\_\_CONFIG \_CONFIG1, \_FOSC\_INTRC\_NOCLKOUT & \_WDTE\_OFF & \_PWRTE\_OFF & \_MCLRE\_ON & \_CP\_OFF & \_CPD\_OFF & \_BOREN\_ON & \_IESO\_ON & \_FCMEN\_ON & \_LVP\_OFF

; CONFIG2

; \_\_config 0xFFFF

\_\_CONFIG \_CONFIG2, \_BOR4V\_BOR40V & \_WRT\_OFF

udata 0x22

numPiso res 1

contaLed res 1

dato res 1

err\_or res 1

secuencia res 1

boliche\_flag res 1

veces\_on res 1

org 0x00

goto INICIO

org 0x04

goto ISR

INICIO

;------- Port Config

banksel TRISB

clrf TRISC ;El puerto C ira conectado al cátodo de cada piso.

clrf TRISD ;Los pines de los puertos A y D irán conectados

clrf TRISA ;a los ánodos de cada columna.

;------- Config Comunicacion Serie

bsf STATUS,RP0

movlw b'00000000' ;TXEN=0 no transmito desde el PIC

;BRGH=0 para utilizarlo en el BaudRate

movwf TXSTA

movlw .25

movwf SPBRG

bsf STATUS,RP1

bcf BAUDCTL,BRG16 ;La velocidad de transmisión será de 2400 baudios

bcf STATUS,RP0

bcf STATUS,RP1

movlw b'10010000' ;CREN=1 habilito la recepción

;SYNC=0 transmisión asincrónica

;SPEN=1 setea los pines RX y TX como input y output respectivamente

movwf RCSTA

;------- Config Interrumpciones

banksel OPTION\_REG

movlw b'10100000' ;Interrumpe por TMR0

movwf INTCON

movlw b'10000011' ;PS=16

movwf OPTION\_REG

banksel ANSEL

clrf ANSEL

banksel ANSELH

clrf ANSELH

;------- Inicialización Variables

bcf STATUS, RP0

bcf STATUS, RP1

clrf PORTA

clrf PORTD

clrf PORTC

clrf numPiso

clrf contaLed

clrf dato

clrf secuencia

clrf err\_or

clrf boliche\_flag

clrf veces\_on

movlw .6

movwf TMR0 ;Inicialmente el TMR0 tendrá un período de 4ms

RECEPCION

btfsc RCSTA,OERR ;Testeo si tuve error por Overrun

goto ERR\_O

btfss PIR1,RCIF

goto RECEPCION

movf RCREG,w

movwf dato

movlw 0x0F

andwf dato,f ;Los datos son enviados en ASCII

call SET\_DATO

goto RECEPCION ;Uso polling para la recepción, el Timer 0

;interrumpirá mientras se ejecuta este bucle.

ERR\_O:

incf err\_or,f ;Si tuve error por Overrrun, reseteo la recepción

bcf RCSTA,CREN

bsf RCSTA,CREN

goto RECEPCION

SET\_DATO:

movf dato,w

sublw .4 ;Los datos validos van entre 0 y 4,

;entonces si la resta da negativa el dato

;recibido es erróneo y no modifico el número

;de secuencia a transmitir.

btfss STATUS,C

return

call SET\_SECUENCIA

movwf secuencia

return

SET\_SECUENCIA:

movf dato,w

addwf PCL,f

retlw .0

retlw .1

retlw .2

retlw .3

retlw .4

ISR:

btfss INTCON,T0IF

goto VOLVER

bcf INTCON,T0IF

call GET\_PISO ;Cada vez interrumpa TMR0 cambio el piso del cubo

;que habilito.

movwf PORTC

goto GET\_SECUENCIA

VOLVER

call SET\_NUMPISO

movlw .6

movwf TMR0

retfie

GET\_PISO:

movf numPiso,w ;Selecciona el piso que habilito

addwf PCL,f

retlw b'00000001'

retlw b'00000010'

retlw b'00000100'

retlw b'00001000'

SET\_NUMPISO:

incf numPiso,f

movlw .4

xorwf numPiso,w ;Chequeo que no me pase de los 4 pisos

btfsc STATUS,Z

clrf numPiso

return

GET\_SECUENCIA

movf secuencia,w

addwf PCL,f ;Selecciono la secuencia a reproducir

goto PRENDE\_TODO\_CUBO

goto PRENDE\_1x1

goto PRENDE\_PERIMETRO

goto PRENDE\_COLUM

goto PRENDE\_BOLICHE

PRENDE\_TODO\_CUBO

banksel OPTION\_REG

movlw b'10000011' ;PS=16

movwf OPTION\_REG

bcf STATUS,RP0

movlw 0xFF

movwf PORTD

movwf PORTA

goto VOLVER

PRENDE\_1x1

banksel OPTION\_REG

movlw b'10000111' ;PS=128

movwf OPTION\_REG ;Esta secuencia requiere un período de TMR0 más

;grande, es decir, cambiar de piso más lentamente.

bcf STATUS,RP0

call SELECT\_LED\_ON

movwf PORTD

movwf PORTA

incf contaLed,f

movlw .8

xorwf contaLed,w

btfsc STATUS,Z

clrf contaLed

goto VOLVER

SELECT\_LED\_ON:

movf contaLed,w

addwf PCL,f

retlw b'00000001'

retlw b'00000010'

retlw b'00000100'

retlw b'00001000'

retlw b'00010000'

retlw b'00100000'

retlw b'01000000'

retlw b'10000000'

PRENDE\_PERIMETRO

banksel OPTION\_REG

movlw b'10000011' ;PS=16

movwf OPTION\_REG

bcf STATUS,RP0

movf numPiso,w

addwf PCL,f

goto PRIMERO\_ULTIMO ;En esta secuencia los leds del piso 0 y 3 prenden igual

goto MEDIO ;mientras que los pisos centrales prenden iguales entre si

goto MEDIO

goto PRIMERO\_ULTIMO

PRIMERO\_ULTIMO

movlw b'10011111'

movwf PORTA

movlw b'11111001'

movwf PORTD

goto VOLVER

MEDIO

movlw b'00001001'

movwf PORTA

movlw b'10010000'

movwf PORTD

goto VOLVER

PRENDE\_COLUM

banksel OPTION\_REG

movlw b'10000111' ;PS=128

movwf OPTION\_REG ;Esta secuencia requiere un período de TMR0 más

;grande, es decir, cambiar de piso más lentamente.

bcf STATUS,RP0

call SELECT\_LED\_ON

movwf PORTD

movwf PORTA

movlw .3

xorwf numPiso,w

btfss STATUS,Z

goto VOLVER ;Espero que se prendan los leds de toda la columna

;antes de pasar a la siguiente columna

incf contaLed,f

movlw .8

xorwf contaLed,w

btfsc STATUS,Z

clrf contaLed

goto VOLVER

PRENDE\_BOLICHE

banksel OPTION\_REG

movlw b'10000011' ;PS=16

movwf OPTION\_REG

bcf STATUS,RP0

movlw .3

xorwf numPiso,w

btfsc STATUS,Z

incf veces\_on,f ;El contador 'veces\_on' lleva la cuenta de

;cuantas veces se habilitaron TODOS los pisos

;del cubo.

call SET\_ON\_OFF

movwf PORTD

movwf PORTA

goto VOLVER

SET\_ON\_OFF:

movlw .8

xorwf veces\_on,w

btfsc STATUS,Z

clrf veces\_on

btfsc STATUS,Z

comf boliche\_flag,f ;Una vez que dicho contador llega al límite

;establecido (en esta caso .8), se complementa

;una bandera indicando si toca apagar o prender

;el cubo y así generar el efecto 'Boliche'.

movlw 0xFF

btfss boliche\_flag,0

movlw 0x00

return

end

## Conclusión

Este proyecto nos sirvió para plasmar los conocimientos teóricos de la materia, consideramos que fue una instancia de aprendizaje muy importante porque logramos entender el funcionamiento de algunos de los módulos que trae integrado el complejo microcontrolador PIC16F887.