Resolución de tarea de Electrónica Digital II

Valenzuela Gabriel

Universidad Nacional de Córdoba - FCEFyN

8 de abril de 2019

Ejercicio 3.1

Gran cantidad de sistemas embedidos necesitan transferir información digital entre su CPU y el mundo exterior. Esta transferencia, se clasifica en un número de categorias, las cuales se lista a continuación:

- Interfaz directa de usuario, incluye switches, teclados, diodos emisores de luz (LEDs) y displays
- Entrada de medida de información, desde sensores exteriores, posiblemente siendo adquirida esta a través de un conversor analógico digital.
- Control de la salida de información, e.g motores u otros actuadores.
- Transferencia de bloques de información hacia o desde otros sistemas/subsistemas, moviendose de forma serial o paralela, e.g. enviando datos seriales a una memoria externa.

Con esta multitud de información llendo y viniendo, es probable que necesitaremos tener una varidad de entradas y/o salidas digitales. Estas son divididas en general en serial y paralelo. En la transferencia serial de información, la información es transferida un bit a la vez; i.e. solamente una sola interconexión es usada para llevar a la información en si, aunque otras lineas son incluidas para sincronización y control.
En la transferencia paralela de información, un conjunto (e.g. 8) de interconexiones es usado. Cada uno puede llevar 1 bit, y trabaja en paralelo con los otros. La información por tanto debe transferirse en grupo de bits, e.g en bytes. Un grupo de interconexiones I/O, manifestado en los pines del microcontrolador, se denomina **puerto paralelo.**
En los microcontroladores de escala media, como el **PIC 16F887** contiene 36 pins de propósito general de I/O disponibles los cuales se agrupan en:

- **PORT A** <0:7>
- **PORT B** <0:7>
- **PORT C** <0:7>
- **PORT D** <0:7>
- **PORT E** <0:4>

Cada pin está identificado por **RX<Y>** donde X simboliza el puerto e Y el número de pin, que está identificado por un bit.
Cada puerto paralelo tiene dos registros de funciones especiales usados para manipular el puerto. Estos se denominan **PORT** y **TRIS**: El registro PORT almacena la salida de información, mientras que TRIS se utliza para programar cada linea en el puerto como entrada o salida. (Un 1 es una entrada, un 0 una salida).
Existen dos registros importantes que controlan la forma de salida o entrada de los pines, i.e, si actúan de manera analógica o digital. Estos son **ANSEL** y **ANSELH**. El registro ANSEL es usado para configurar el modo de entrada de un pin I/O a analógico, seteando de manera apropiada el bit de ANSEL en HIGH causará que todas las lecturas digitales en el pin sean leidas como '0' y permitir las funciones analógicas del pin operen de forma correcta. El estado de los bits de ANSEL no afectan en la funciones de salida digital. Por tanto, un pin con $TRIS < x >= 0$ y $ANSEL < x >= 1$ operará con su salida **digital** pero la entrada será **analógica**. (Nota: Todas las operaciones de puerto son leer-modificar-escribir).
Al setear un pin como entrada analógica se desactiva de manera automática la circuitería de entrada digital, pull-ups, e interrupciones al cambio si está disponible. El correspondiente bit x de TRIS<x>debe ser puesto en modo Input a fin de permitir el control externo de voltage en el pin. El registro ANSELH opera de manera similar, pero para el PORT B únicamente.
Esto se debe a que, solo se tienen 14 posibles entradas analógicas, por eso ANSEL <0:7>y ANSELH <8:13>que están en el PORT A, y parte del PORT B. En cuanto a las características eléctricas de cada pin y puerto se tiene:

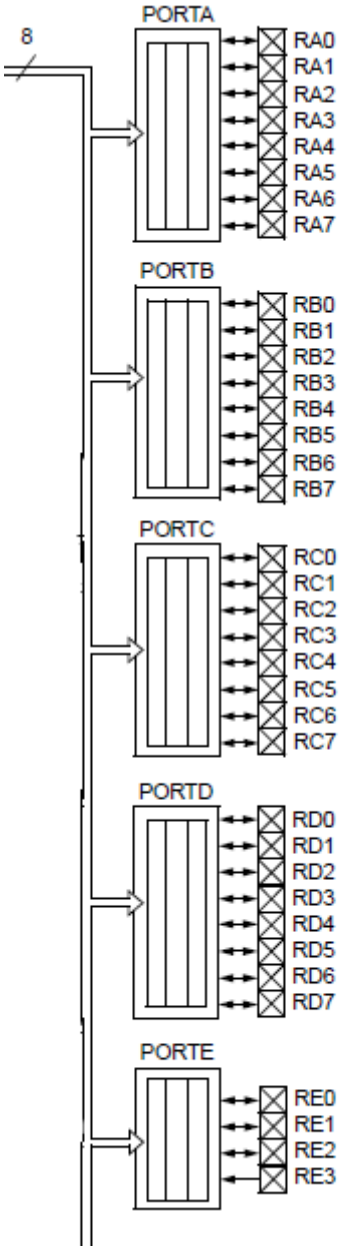


Figura 1: Puertos PIC16F887

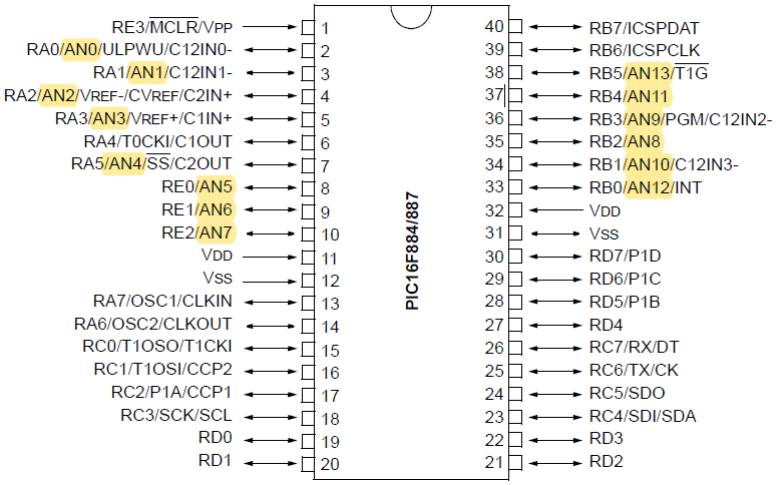


Figura 2: Pines de entrada analógica

Bank 0		Bank 1		Bank 2		Bank 3	
Indirect addr. (1)	00h	Indirect addr. (1)	80h	Indirect addr. (1)	100h	Indirect addr. (1)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h	WDTCON	105h	SRCON	185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h	CM1CON0	107h	BAUDCTL	187h
PORTD(2)	08h	TRISD(2)	88h	CM2CON0	108h	ANSEL	188h
PORTE	09h	TRISE	89h	CM2CON1	109h	ANSELH	189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDAT	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2(1)	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved	18Eh
TMR1H	0Fh	OSCCON	8Fh	EEADRH	10Fh	Reserved	18Fh
T1CON	10h	OSCTUNE	90h	General Purpose Registers 16 Bytes	110h	General Purpose Registers 16 Bytes	190h
TMR2	11h	SSPCON2	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h	WPUB	95h		115h		195h
CCPR1H	16h	IOCB	96h		116h		196h
CCP1CON	17h	VRCON	97h		117h		197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah	SPBRGH	9Ah		11Ah		19Ah
CCPR2L	1Bh	PWM1CON	9Bh		11Bh		19Bh
CCPR2H	1Ch	ECCPAS	9Ch		11Ch		19Ch
CCP2CON	1Dh	PSTRCON	9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
General Purpose Registers 96 Bytes	20h–6Fh	General Purpose Registers 80 Bytes	A0h–EFh	General Purpose Registers 80 Bytes	120h–16Fh	General Purpose Registers 80 Bytes	1A0h–1EFh
accesses 70h-7Fh	70h–7Fh	accesses 70h-7Fh	F0h–FFh	accesses 70h-7Fh	170h–17Fh	accesses 70h-7Fh	1F0h–1FFh

Figura 3: Ubicación en la memoria de datos de los registros PORT, TRIS, ANSEL y ANSELH

Corriente máxima de salida de pin V_{SS}	95[mA]
Corriente máxima de entrada de pin V_{DD}	95[mA]
Corriente máxima de fuente/sumidero en cualquier pin de I/O	25[mA]
Corriente máxima de fuente/sumidero en los puertos de (combinada) I/O	90[mA]

Cuadro 1: Tabla de características eléctricas

Por tanto como máximo, si se tienen 8 puertos funcionando, sólo se podrá emitir/recibir por puerto 11,25$[mA]$ (Puerto de 8 bits)

Ejercicio 3.2

```
;*********************************************************************************
; TITLE : EXERCISE 3.2
; PROFESSOR: MARTIN DEL BARCO
; AUTHOR: VALENZUELA GABRIEL EMANUEL
; SUBJET: DIGITAL ELECTRONICS II
; INSTITUTE : FEFYN - UNC
; DATE: 04-08-2019
; VERSION: 1.0.2
; CHANGE VERSION 1.0.0 TO 1.0.2: SOLVE BUG WITH THE OUPUTS.
;
; RESUME:
; THIS PROGRAM SWICHT TWO LEDS ON RB2/RB3 ACCORDING TO RA4 AND RB0
;*********************************************************************************
#include "p16f887.inc"  ;HEADER THAT CONTAINS ALL ADDRESS MEMORY DEFINED BY MI-
                        ;CROCHIP
    LIST P=16F887       ;DEFINE THE MICROCONTROLLER TO BE USED.
;*********************************************************************************
    ; __config 0x3FF1
 __CONFIG _CONFIG1, _FOSC_XT & _WDTE_OFF & _PWRTE_ON & _MCLRE_ON & _CP_OFF & _CPD_OFF & _BOREN_ON & _IE
; CONFIG2
; __config 0x3FFF
 __CONFIG _CONFIG2, _BOR4V_BOR40V & _WRT_OFF
    ORG 0x00
    GOTO PROGRAM


PROGRAM: ORG 0x05

        CLRW            ;CLEAR THE WORK REGISTER
        CALL CONFIGURE_PORTS
        CLRW
        CLRF PORTA
        CLRF PORTB
LOOP:   MOVFW PORTA         ;MOVE THE CONTENT OF PORTA TO W
        BTFSC PORTA,RA4     ;IF RA4 == 0 ?
        GOTO ILRB2          ;FALSE --> LED RB3 ON
        GOTO OLRB2          ;TRUE -->  LED RB3 OFF
LOOPS:  BTFSC PORTB,RB0     ;IF RB0 == 0 ?
        GOTO ILRB3          ;FALSE --> LED RB4 ON
        GOTO OLRB3          ;TRUE -->  LED RB4 OFF
OLRB2:  BCF PORTB,RB2
        GOTO LOOPS
ILRB2:  BSF PORTB,RB2
        GOTO LOOPS
OLRB3:  BCF PORTB,RB3
        GOTO LOOP
ILRB3:  BSF PORTB,RB3
        GOTO LOOP
        ;INFINITY LOOP
;*********************************************************************************
;                                   SUBRUTINE
CONFIGURE_PORTS: ORG 0X1A
    ;GO TO BANK 1, THAT IS, RP1 = 0 AND RP0=1
    BCF STATUS,RP1
    BSF STATUS,RP0
```

```
    CLRF TRISA
    CLRF TRISB
    MOVLW B'00010000'    ;RA<4> IS A INPUT
    MOVWF TRISA
    MOVLW B'00000001'    ;RB<0> IS A INPUT
    MOVWF TRISB
    ;GO TO ANSELH TO SET RB0 LIKE DIGITAL INPUT
    BSF STATUS,RP1
    CLRF ANSELH
    CLRF ANSEL
    BCF STATUS,RP0
    BCF STATUS,RP1
    RETURN

    END
```

Ejercicio 3.7

```
;*********************************************************************************
; TITLE : EXERCISE 3.7
; PROFESSOR: MARTIN DEL BARCO
; AUTHOR: VALENZUELA GABRIEL EMANUEL
; SUBJET: DIGITAL ELECTRONICS II
; INSTITUTE : FEFYN - UNC
; DATE: 04-08-2019
; VERSION: 1.0.0
;
; RESUME:
; THIS PROGRAM COUNT FROM 0 TO E USING A SEVEN SEGMENT DISPLAY
;*********************************************************************************
#include "p16f887.inc"  ;HEADER THAT CONTAINS ALL ADDRESS MEMORY DEFINED BY MI-
                        ;CROCHIP
#INCLUDE "__tables.inc"
#INCLUDE "__subrutines.inc"

    LIST P=16F887       ;DEFINE THE MICROCONTROLLER TO BE USED.
;*********************************************************************************
    ; __config 0x3FF1
 __CONFIG _CONFIG1, _FOSC_XT & _WDTE_OFF & _PWRTE_ON & _MCLRE_ON & _CP_OFF & _CPD_OFF & _BOREN_ON & _IE
; CONFIG2
; __config 0x3FFF
 __CONFIG _CONFIG2, _BOR4V_BOR40V & _WRT_OFF
    ORG 0x00
    GOTO PROGRAM
    ORG 0x04
    BCF STATUS,RP0
    BCF STATUS,RP1
    BTFSS PORTB,RB0
    GOTO TMR_INT
    GOTO BTN_INT

DELAY_50MS EQU 0XC3
MULTIPLIER_FACTOR EQU 0X14
COUNTER EQU 0X20
TIME_REG EQU 0x21
SIZE_ARRAY SET  0xA
PROGRAM: ORG 0x20

START
    CLRW
    CLRF COUNTER
```

```
        BANKSEL TRISB
        MOVLW B'00000001'
        MOVWF TRISB
        MOVLW 0xFF
        MOVWF TRISD
        BANKSEL INTCON
        BCF INTCON,INTE
        BANKSEL PORTB
        MOVLW 0x00
        MOVWF PORTB
IN      BTFSS PORTB,RB0
        GOTO IN
        CLRF PORTD
        CALL CONFIGURE_TRIS

        CALL INIT_TIMER
        MOVLW 0xA
        MOVWF COUNTER
        GOTO PROGRAM_MAIN
PROGRAM_MAIN
        GOTO $
            ;000-0-1010 --- MAX NUMBER
;*********************************************************************************
;                                       SUBRUTINE


TMR_INT: ORG 0x100
        MOVLW   DELAY_50MS
        MOVWF   TMR0
        DECFSZ  TIME_REG,F
        GOTO    END_INT
        DECF    COUNTER,F               ;DECREMENT COUNT
        BTFSS   STATUS,Z                ;THE Z FLAG IS AFFECTED ?
        GOTO    INCR                    ;FALSE--> INCREMENT THE COUNT
        GOTO    RESET_COUNT             ;TRUE --> RESET THE COUNT
INCR
        MOVFW   COUNTER
        SUBLW   SIZE_ARRAY              ;COUNTER (10) - X --> W
        CALL    NUMBERS                 ;CALL TABLE WITH THE RESULT. E.G 10 - 10 = 0
        MOVWF   PORTD                   ;LATCH PORTD
        MOVLW   MULTIPLIER_FACTOR       ;RESET TIMER
        GOTO    TIME                    ;MOV TO MULTIPLER FACTOR
RESET_COUNT
        MOVF COUNTER,W
        CALL NUMBERS
        MOVWF PORTD
        MOVLW 0xA
        MOVWF COUNTER
        MOVLW MULTIPLIER_FACTOR
TIME
        MOVWF   TIME_REG
END_INT
        BCF     INTCON,T0IF
        RETFIE                  ;GIE
BTN_INT: ORG 0x0E
        MOVFW COUNTER
        SUBLW SIZE_ARRAY
        BTFSS STATUS,Z
        GOTO RESET_COUNT
        MOVLW 0x0
        CALL NUMBERS
```

```
        MOVWF PORTD
        RETFIE
        END
```

Tables.inc

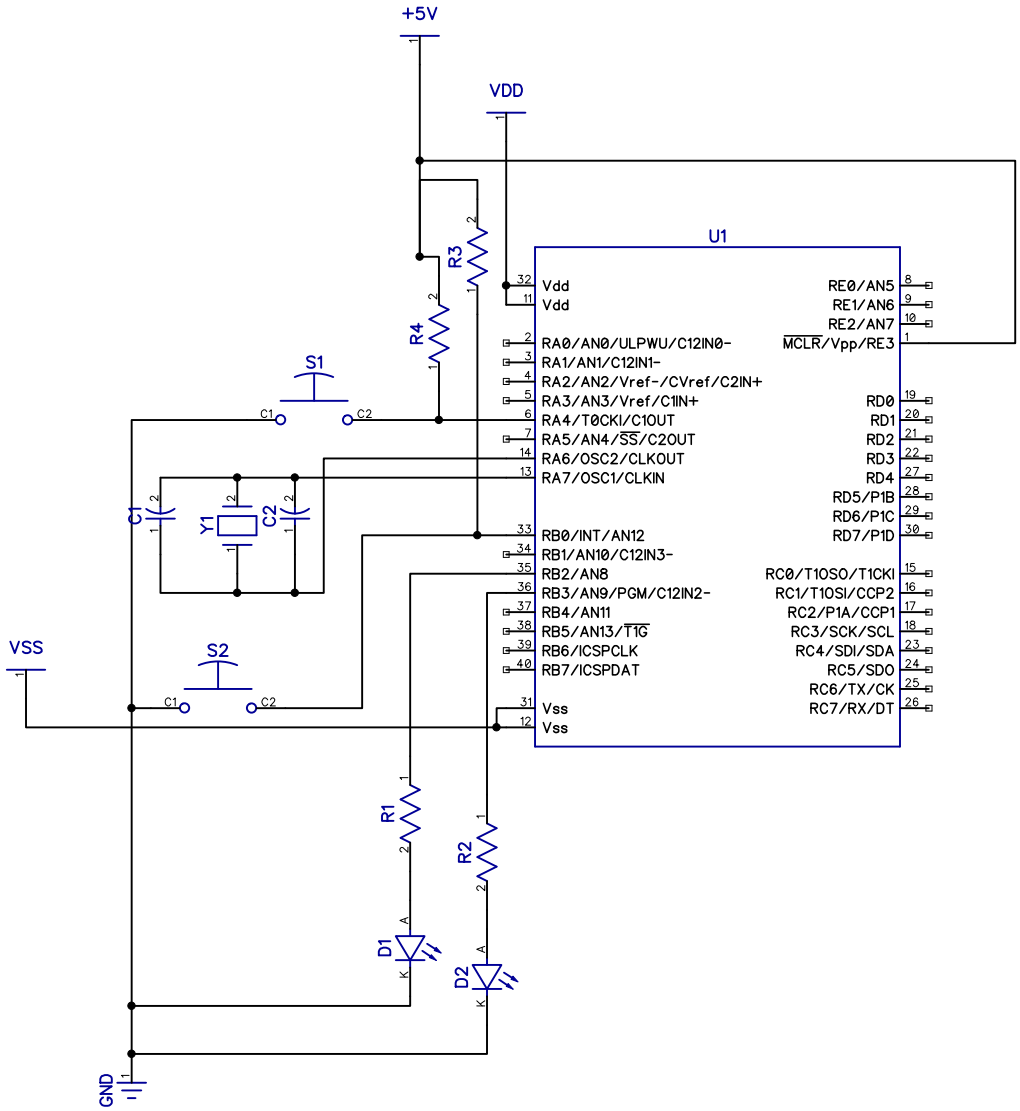
```
NUMBERS ORG 0X80
        ADDWF PCL
        DT 0X3F
        DT 0X06
        DT 0X5B
        DT 0X4F
        DT 0X66
        DT 0X6D
        DT 0X7D
        DT 0X07
        DT 0X7F
        DT 0X67
```

Subrutines.inc

```
DISPLAY_SS MACRO NUMBER ORG 0x400

 ENDM

INIT_TIMER ORG 0x500
        BSF STATUS,RP0
        BCF STATUS,RP1
        ;---01----- SWITCH BANK 1
        MOVLW B'11000111'
        MOVWF OPTION_REG
        BCF STATUS,RP0
        ;---00----- SWITCH BANK 0
        CLRF TMR0
        MOVLW DELAY_50MS
        MOVWF TMR0
        MOVLW MULTIPLIER_FACTOR
        MOVWF TIME_REG
        ;MOVWF TIME_REG
        MOVLW   B'11110000'
        MOVWF   INTCON          ;T0IE - GIE
        RETURN

CONFIGURE_TRIS ORG 0X600
        BSF STATUS,RP0
        BCF STATUS,RP1
        MOVLW 0X00
        MOVWF TRISD
        BSF STATUS,RP1
        MOVLW 0x00
        MOVWF ANSEL
        MOVWF ANSELH
        BCF STATUS,RP0
        BCF STATUS,RP1
        CLRF PORTD
        CLRF PORTB
        RETURN
```

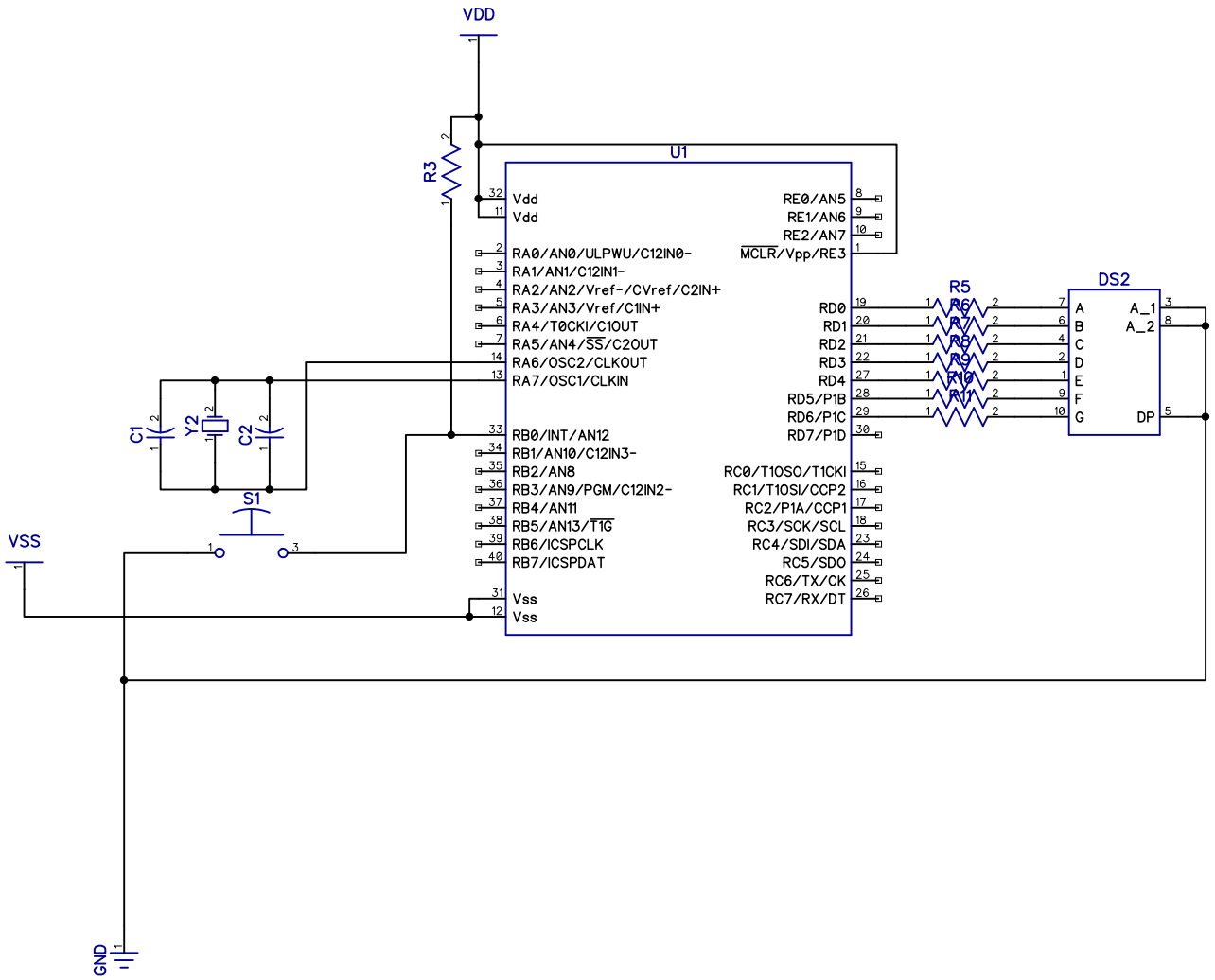
Esquemáticos

RevNo	Revision note	04/08/2019	ED2	Checked



Itemref	Quantity	Excersice 3.2				
Designed by Valenzuela Gabriel					Date 04/08/2019	Scale 1:1
		EXC 3.2				
					Edition 1	Sheet 1/1

RevNo	Revision note	04/08/2019	ED2	Checked



Itemref	Quantity	Excersice 3.7			
Designed by Valenzuela Gabriel				Date 04/08/2019	Scale 1:1
		EXC 3.7			
				Edition 1	Sheet 1/1

Diagramas

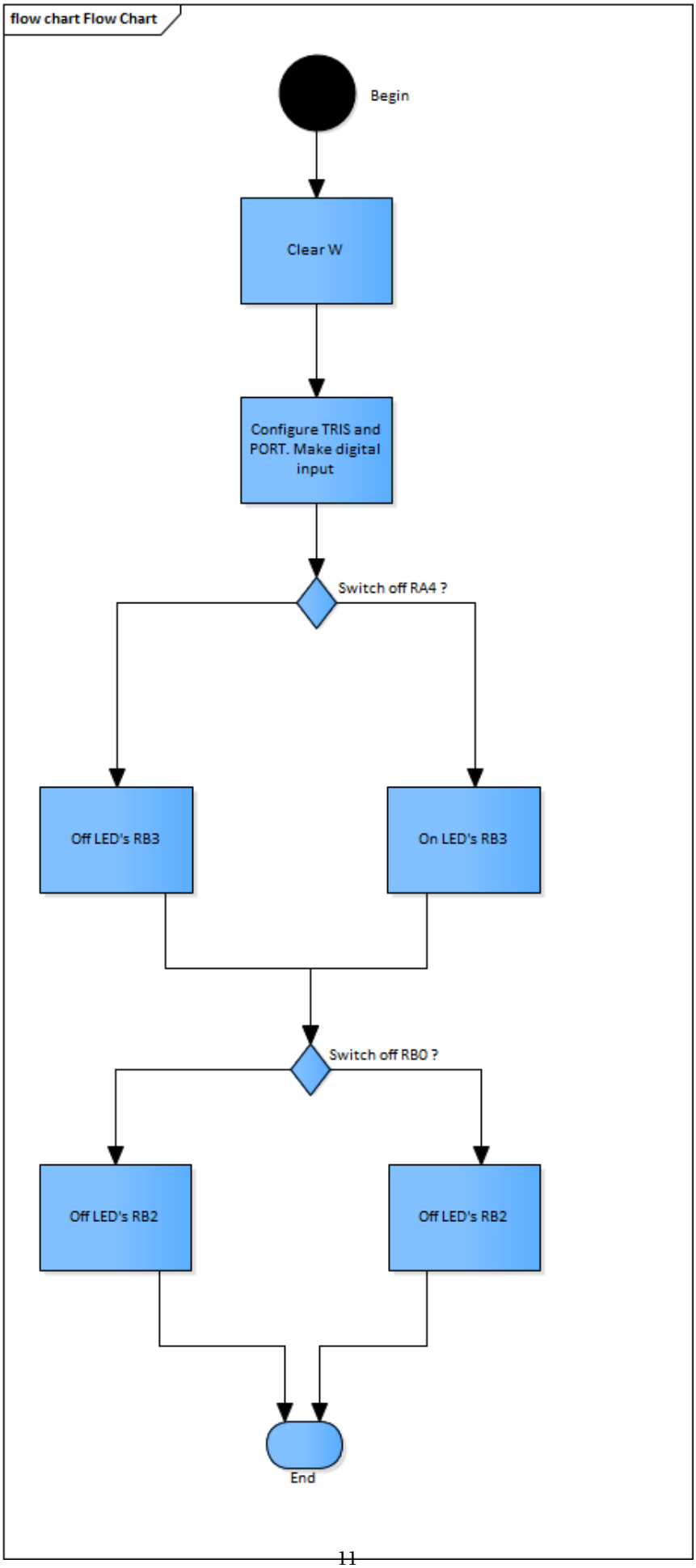


Figura 4: Flow chart

Referencias

[1] Valdes-Perez, Fernando E., and Ramon Pallas-Areny. Microcontrollers: fundamentals and applications with PIC. CRC press, 2009.

[2] PIC16F882/883/884/886/887 Data Sheet

[3] WILMSHURST, Tim. Designing embedded systems with PIC microcontrollers: principles and applications. Elsevier, 2006.

[4] MEIKLEJOHN, D. Introduction to PIC Programming Baseline to Enhanced Mid-Range Architecture. Gooligum Electronics, 2013.