Guía 5

1. TRISBbits.TRISB3 = 0; //Forma de poner el LED como salida

TRISBbits.TRISB2 = 0; //Forma de poner el LED como salida

LATXbits.LATXY = valor; //Escribir bit

valor = PORTXbits.RXY; //Leer bit

1. PLL

FIN = 10 MHz (Cristal)

Fos= FIN\*M/N1\*N2= 10 \*32\*2\*2=80 MHz

Fcy= 40 MPIS (Cada ciclo tarda 2 ciclos del clock)

DAC1STATbits.ROEN = 1; /\* Right Channel DAC Output Enabled \*/

DAC1STATbits.LOEN = 1; /\* Left Channel DAC Output Enabled \*/

DAC1STATbits.RITYPE = 1; /\* Right Channel Interrupt if FIFO is not Full \*/

DAC1STATbits.LITYPE = 1; /\* Left Channel Interrupt if FIFO is not Full \*/

DAC1CONbits.AMPON = 0; /\* Amplifier Disabled During Sleep & Idle Modes \*/

DAC1CONbits.DACFDIV = 15; /\* Divide Clock by 15 \*/

/\* Se busca una F\_DAC = 39062.5

\* F\_IN = 160MHz / 256

\* F\_DAC = F\_IN / (DACFDIV + 1) ---> Debe dividirse por 16, entonces:

\* DACFDIV = 15

\*/

DAC1CONbits.FORM = 1; /\* Data Format is Unsigned \*/

DAC1DFLT = 0x0000; /\* Default value set to Midpoint when FORM = 0 \*/

IFS4bits.DAC1RIF = 0; /\* Clear Right Channel Interrupt Flag \*/

IFS4bits.DAC1LIF = 0; /\* Clear Left Channel Interrupt Flag \*/

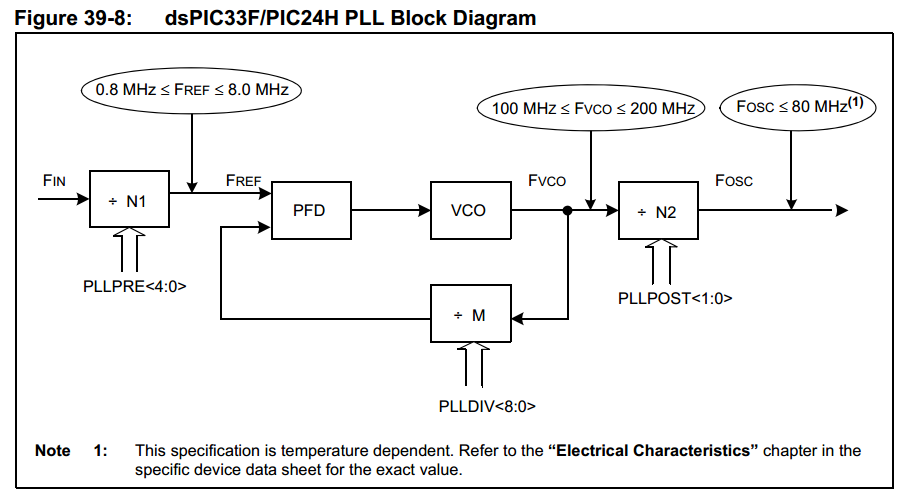
IEC4bits.DAC1RIE = 1; /\* Right Channel Interrupt Enabled \*/

IEC4bits.DAC1LIE = 1; /\* Left Channel Interrupt Enabled \*/

DAC1CONbits.DACEN = 1; /\* DAC1 Module Enabled \*/

ACLKCONbits.SELACLK = 0; //MUY IMPORTANTE CONFIGURAR EL OSCILADOR QUE USE EL PLL

ACLKCONbits.APSTSCLR = 0b111; //NO DIVIDE LA FRECUENCIA



1. DMA

DMA2PAD = 0x0300; //DMA lee desde periferico ADC

DMA2REQ = 13; //Configuro DMA para ADC1 señal que dispara

DMA2STA = \_\_builtin\_dmaoffset(RightBufferB);

DMA2STB = \_\_builtin\_dmaoffset(RightBufferA);

DMA0CONbits.DIR = 1; /\* Ram-to-Peripheral Data Transfer \*/

1. ADC

AD1CON1bits.AD12B = 1; // 1 = 12 bits, 0 = 10 bits

AD1CON2bits.VCFG = 4; // 4 = Vdd y Vss como referencia de voltaje

AD1CON3bits.ADRC = 0; // Clock de conversion derivado clock general

AD1CON3bits.ADCS = 9; // TAD = TCY \* (ADCS + 1) = (1/40M) \* 10 = 250n s (4 MHz)

AD1PCFGL = 0xFFFF; //Configuro todos los bits como digitales. 1 = D 0 = A

AD1PCFGLbits.PCFG0 = 0; // Defino AN0 como analogico. Primer bit

AD1CHS0bits.CH0SA = 0; // Entrada positiva en AN0

AD1CON1bits.ASAM = 1; // 1 = muestreo automatico. 0 = muestreo manual

DAC1CONbits.FORM = 1; /\* Data Format is signed \*/

AD1CON2bits.SMPI = 0; // Incrementa el DMA luego de cada muestra/conversion

AD1CON1bits.SSRC = 0b100; // Timer 5 es el temporizador de muestreo.

AD1CON1bits.ADON = 1; // Turn on the ADC

1. UART

TRISBbits.TRISB2 = 0;

RPINR18bits.U1RXR = 5; //Defino puerto de recepcion

RPOR9bits.RP19R = 6; //Defino puerto de transmision

U1MODEbits.UARTEN = 0; // Enable UART

U1MODEbits.STSEL = 0; // 1-Stop bit

U1MODEbits.PDSEL = 0; // No Parity, 8-Data bits

U1MODEbits.ABAUD = 0; // Auto-Baud disabled

U1MODEbits.BRGH = 0; // Standard-Speed mode

U1BRG = BRGVAL; // Baud Rate setting for 9600

U1STAbits.URXISEL = 0;

IEC0bits.U1RXIE = 1; // Activa la interrupcion por valores recibidos

IFS0bits.U1RXIF = 0;

U1MODEbits.UARTEN = 1; // Enable UART

U1STAbits.UTXEN = 1; // Activa la transmision de la UART