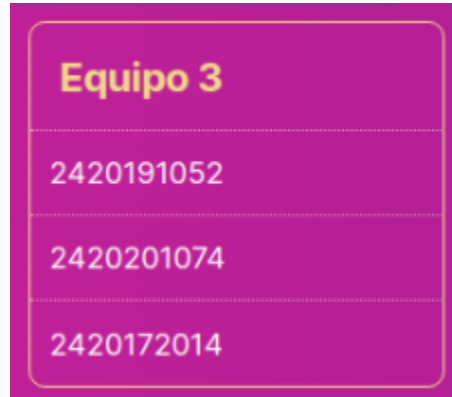


María Camila López López 2420201074

Juan José Hernández 2420191052

Juan Sebastián Corredor 2420172014



## Examen II - Parte B 2021a

1. Cierta aplicación de instrumentación de temperatura para agua tiene la siguiente configuración:
  - Sensor de temperatura con rango  $[-100\ 1000]^{\circ}\text{C}$  y respuesta equivalente en voltaje de  $[0\ 10]$  volts.
  - La temperatura del proceso a instrumentar nunca excede los  $100^{\circ}\text{C}$  ni se decrementa por debajo de los  $0^{\circ}\text{C}$ .
  - El microcontrolador Para Utilizar tiene una cuantificación de 8 bits en un rango de voltaje de entre  $[0\ 3.3]$  volts
  - Mencionado lo anterior se le solicita como ingeniero de instrumentación:
    - Proponer un modelo que describa la relación  $V(\text{Temperatura})$  **[0.3 puntos]**
    - Sugerir un acondicionamiento del sensor para que 0 volts a la entrada del microcontrolador representen  $0^{\circ}\text{C}$  y 3.3 volts a la entrada del microcontrolador representen  $100^{\circ}\text{C}$ . **[0.3 puntos]**
    - Proponer una ecuación a programar que a partir del valor digital entero cuantificado de 8 bits determine el valor de Temperatura en  $^{\circ}\text{C}$  dentro del microcontrolador. Temperatura (valor digital). **[0.2 puntos]**
    - Determinar la resolución para voltaje y temperatura. EL cambio mínimo en volts percibido por el sistema y su equivalente, cambio mínimo en temperatura percibido. **[0.2 puntos]**
2. Utilizando la tarjeta de Microchip Curiosity Nano y el MCU PIC16F15244 configure un proyecto que tenga como entrada: un pulsador y como salida: el led amarillo de la tarjeta.

envíe la señal PWM hacia el LED con las siguientes condiciones:

- Mientras ESTE PRESIONADO el pulsador el PWM debe incrementar lentamente desde el valor que tenga asignado en ese momento hasta 100% en 2 segundos. **[0.75 puntos]**

- Mientras NO ESTE PRESIONADO el pulsador el PWM debe decrementarse lentamente desde el valor que tenga en ese momento hasta 0% en un tiempo aproximado de 2 segundos. [0.75 puntos]

El entregable es el archivo ".C" que se programa sobre la tarjeta para funcionar. NOTA. Tenga en cuenta que el valor PWM no puede ser mayor a 100% ni menor al 0%.

## SOLUCION

1.

Monica Camila Lopez, Juan Sebastian Concedor, Juan Jose H

A) -100 a 1000° Temp  
0 a 10 V  $VS = m \cdot T + b$

$$m = \frac{10 - 0}{1000 - (-100)} = \frac{1}{110}$$

$$VS = \frac{1}{110} \cdot T + b$$

$$0 = \frac{1}{110} \cdot (-100) + b \quad b = \frac{10}{11}$$

$$VS = \frac{1}{110} \cdot T + \frac{10}{11}$$

1.1  $VS = \left(\frac{1}{110}\right) \cdot (-100) + \frac{10}{11}$   
 $V_{min} = 0$

1.2  $VS = \left(\frac{1}{110}\right) \cdot (1000) + \frac{10}{11}$   
 $V_{max} = 10$

$$b) \quad V_S(100) = \frac{1}{110} \cdot 100 + \frac{10}{11} = \frac{20}{11}$$

$$V_S(0) = \frac{1}{110} \cdot 0 + \frac{10}{11} = \frac{10}{11}$$

$$\begin{array}{ccccc} \text{---} & 0 & \text{a} & 100^\circ & T \\ & 0 & \text{a} & 3,3 & V \end{array}$$

$$m = \frac{3,3 - 0}{\frac{21}{11} - \frac{9}{11}} = 3,025 = \left( \frac{121}{40} \right)$$

$$V_2 \cdot 3,3 = \frac{121}{40} \cdot \left( \frac{21}{11} \right) + b$$

$$b = 3,3 - \frac{121}{40} \cdot \left( \frac{21}{11} \right) = -\frac{99}{40}$$

$$V_2 = \frac{121}{40} \cdot V_S - \frac{99}{40} \rightarrow \text{Ec. 2}$$

$$V_2 = \frac{121}{40} \cdot V_s - \frac{99}{40} \rightarrow \text{Ec. 2}$$

$$\therefore T = -10^\circ\text{C} \Rightarrow V_s = \frac{9}{11} \text{ V.}$$

$$V_2 = \frac{121}{40} \cdot \left(\frac{9}{11}\right) - \frac{99}{40} = 0 \text{ V}$$

$$T = 110^\circ\text{C} \Rightarrow V_s = \frac{21}{11} \text{ V.}$$

$$V_2 = \frac{121}{40} \cdot \left(\frac{21}{11}\right) - \frac{99}{40} = 3.3 \text{ V}$$

c.

$A_0 = \text{Lectura Digital}$

$n = 8 \text{ bits.}$

$$A_0 = V_2 \left( \frac{2^n - 1}{V_{\text{pic}}} \right) = V_2 \cdot \left( \frac{2^8 - 1}{3.3} \right) = V_2 \left( \frac{255}{3.3} \right)$$

$$A_0 = V_2 \left( \frac{255}{3.3} \right) = \left( \frac{121}{40} \cdot V_s - \frac{99}{40} \right) \left( \frac{255}{3.3} \right)$$



$$A_0 = \left( \frac{121}{40} \cdot \left( \frac{1}{110} \cdot T + \frac{10}{11} \right) - \frac{99}{40} \right) \cdot \left( \frac{255}{3.3} \right)$$

$$A_0 = \frac{13}{4} \cdot T + \frac{85}{4}$$

$$T = \frac{8}{13} \cdot A_0 - 10$$

$$\therefore A_0 = 255$$

$$T = \frac{8}{13} (255) - 10 = 130^\circ\text{C}$$

$$A_0 = 0$$

$$T = \frac{8}{13} (0) - 10 = -10^\circ\text{C}$$

D. Resolución Voltage y temp.

$$RV = \frac{\text{Rango } V}{2^n} = \frac{V_{\max} - V_{\min}}{2^n}$$

$$RV = \frac{3.3 - 0}{256} = \frac{3.3}{256} \approx 12.89 \text{ mV}$$

256 niveles  $\rightarrow 0.4687^\circ\text{C}$  + que el anterior.

V(T)

$$V_s(T) = \frac{1}{110} \cdot T + \frac{10}{11}$$

$$V_s(T) = \frac{1}{110} (50) + \frac{10}{11} = \frac{15}{11} \text{ V}$$

$$V_{S+1}(T) = \frac{1}{110} (51) + \frac{10}{11} = \frac{151}{110} V$$

$$V_{S+1} - V_S = \frac{151}{110} V - \frac{15}{11} V = \frac{1}{110} V = 9,09 \text{ mV}$$

$T(V)$

$$T_S(V) = 110(V) - 100$$

$$T_S(V) = 110 \cdot V - 100$$

2.

/\*

////////////////// DATOS DEL PROGRAMA ////////////////////

// Nombre: Parcial3Final

// MICRO: PIC16F15244

// ESTUDIANTES: Juan Sebastian Corredor gongora 2420172014, María Camila López 2420201074,

\* Juan José Hernández Bravo 2420191052-

\*

// Profesor: Harold F MURCIA

// FECHA: 22 de mayo del 2021

////////////////// CONFIGURACION del MCU //////////////////// \*/

#include <xc.h>

#include <stdint.h>

#include <stdbool.h>

#include <conio.h>

#pragma config FEXTOSC = OFF // External Oscillator Mode Selection bits->Oscillator not enabled

#pragma config RSTOSC = HFINTOSC\_1MHZ // Power-up Default Value for COSC bits->HFINTOSC (1 MHz)

#pragma config CLKOUTEN = OFF // Clock Out Enable bit->CLKOUT function is disabled; I/O function on RA4

#pragma config VDDAR = HI // VDD Range Analog Calibration Selection bit->Internal analog systems are calibrated for operation between VDD = 2.3V - 5.5V

// CONFIG2

#pragma config MCLRE = EXTMCLR // Master Clear Enable bit->If LVP = 0, MCLR pin is MCLR; If LVP = 1, RA3 pin function is MCLR

#pragma config PWRTS = PWRT\_OFF // Power-up Timer Selection bits->PWRT is disabled

#pragma config WDTE = OFF // WDT Operating Mode bits->WDT disabled; SEN is ignored

#pragma config BOREN = ON // Brown-out Reset Enable bits->Brown-out Reset Enabled, SBOREN bit is ignored

#pragma config BORV = LO // Brown-out Reset Voltage Selection bit->Brown-out Reset Voltage (VBOR) set to 1.9V

#pragma config PPS1WAY = ON // PPSLOCKED One-Way Set Enable bit->The PPSLOCKED bit can be cleared and set only once in software

#pragma config STVREN = ON // Stack Overflow/Underflow Reset Enable bit->Stack Overflow or Underflow will cause a reset

// CONFIG4

#pragma config BBSIZE = BB512 // Boot Block Size Selection bits->512 words boot block size

#pragma config BBEN = OFF // Boot Block Enable bit->Boot Block is disabled

#pragma config SAFEN = OFF // SAF Enable bit->SAF is disabled

#pragma config WRTAPP = OFF // Application Block Write Protection bit->Application Block is not write-protected

#pragma config WRTB = OFF // Boot Block Write Protection bit->Boot Block is not write-protected

#pragma config WRTC = OFF // Configuration Registers Write Protection bit->Configuration Registers are not write-protected

#pragma config WRTSAF = OFF // Storage Area Flash (SAF) Write Protection bit->SAF is not write-protected

```
#pragma config LVP = ON // Low Voltage Programming Enable bit->Low Voltage programming
enabled. MCLR/Vpp pin function is MCLR. MCLR Configuration bit is ignored.
```

```
// CONFIG5
```

```
#pragma config CP = OFF // User Program Flash Memory Code Protection bit->User Program
Flash Memory code protection is disabled
```

```
////////// DEFINICIONES //////////
```

```
#define _XTAL_FREQ 1000000
```

```
#define LED0_SetHigh()      LATAbits.LATA2 = 1;
```

```
#define LED0_SetLow()      LATAbits.LATA2 = 0;
```

```
#define ACQ_US_DELAY 10
```

```
//////////
```

```
////////// VARIABLES GLOBALES //////////
```

```
uint16_t dutyCycle0 = 0x00;
```

```
uint16_t dutyCycle20 = 0x0C8;
```

```
uint16_t dutyCycle40 = 0x0190;
```

```
uint16_t dutyCycle60 = 0x0258;
```

```
uint16_t dutyCycle80 = 0x0320;
```

```
uint16_t dutyCycle100 = 0x03E7;
```

```
int led = 0;
```

```
////////// DECLARACI?N DE FUNCIONES Y PROCEDIMIENTOS //////////
```

```
void PIN_MANAGER_Initialize(void)
```

```
{
```

```
    // LATx registers
```



```
LATA = 0x00;
```

```
LATB = 0x00;
```

```
LATC = 0x00;
```

```
// TRISx registers
```

```
TRISA = 0x3B;
```

```
TRISB = 0xF0;
```

```
TRISC = 0xFF;
```

```
// ANSELx registers
```

```
ANSELC = 0xFB;
```

```
ANSELB = 0xF0;
```

```
ANSELA = 0x33;
```

```
// WPUx registers
```

```
WPUB = 0x00;
```

```
WPUA = 0x04;
```

```
WPUC = 0x04;
```

```
// ODx registers
```

```
ODCONA = 0x00;
```

```
ODCONB = 0x00;
```

```
ODCONC = 0x00;
```

```
// SLRCONx registers
```

```
SLRCONA = 0x37;
```

```
SLRCONB = 0xF0;
```

```
SLRCONC = 0xFF;
```

```

// INLVLx registers

INLVLA = 0x3F;

INLVLB = 0xF0;

INLVLC = 0xFF;


// TRISx registers

TRISA2 = 0;    // Definiendo puerto A2 como salida digital

TRISC2 = 1;    // Definiendo puerto C2 como entrada digital

WPUC2 = 1;    // Activando resistencia d pull-up

ANSELAbits.ANSA1 = 1; // Definiendo entrada como analoga


RA2PPS = 0x03;    // RA2->PWM3:PWM3OUT;
}


void OSCILLATOR_Initialize(void)
{
    OSCEN = 0x00;    // MFOEN disabled; LFOEN disabled; ADOEN
disabled; HFOEN disabled;

    OSCFRQ = 0x00;    // HFFRQ0 1_MHz

    OSCTUNE = 0x00;

}


void TMR2_Initialize(void)
{
    T2CLKCON = 0x01;    // T2CS FOSC/4;

    T2HLT = 0x00;    // T2PSYNC Not Synchronized; T2MODE
Software control; T2CKPOL Rising Edge; T2CKSYNC Not Synchronized;

    T2RST = 0x00;

```

```

T2PR = 249;

T2TMR = 0x00;

PIR1bits.TMR2IF = 0;                // Clearing IF flag.

T2CON = 0b10000000;                // T2CKPS 1:1; T2OUTPS 1:1; TMR2ON on;
}

```

```

void PWM3_Initialize(void)
{
    PWM3CON = 0x90;                // PWM3POL active_low; PWM3EN
    enabled;

    PWM3DCH = 0x3E;

    PWM3DCL = 0x40;
}

```

```

void PWM3_LoadDutyValue(uint16_t dutyValue)
{
    PWM3DCH = (dutyValue & 0x03FC)>>2;        // Writing to 8 MSBs of PWM
    duty cycle in PWMDCH register

    PWM3DCL = (dutyValue & 0x0003)<<6;        // Writing to 2 LSBs of PWM
    duty cycle in PWMDCL register
}

```

```

void LOAD(void)
{
    if (!PORTCbits.RC2 && led <=100)
    {
        PWM3_LoadDutyValue(led);

        led += 1;
    }
}

```

```

        if (led >= 100)
        {
            led = 100;
        }
        __delay_ms(20);
    }
else
{
    PWM3_LoadDutyValue(led);
    led -= 1;

    if (led <= 0)
    {
        led = 0;
    }
    __delay_ms(20);
}
}

////////// INICIO DEL PROGRAMA PRINCIPAL //////////

```

```

void main(void)
{
    PIN_MANAGER_Initialize();
    OSCILLATOR_Initialize();
    TMR2_Initialize();
    PWM3_Initialize();
    while (1)
    {
        LOAD();
    }
}

```

}

}