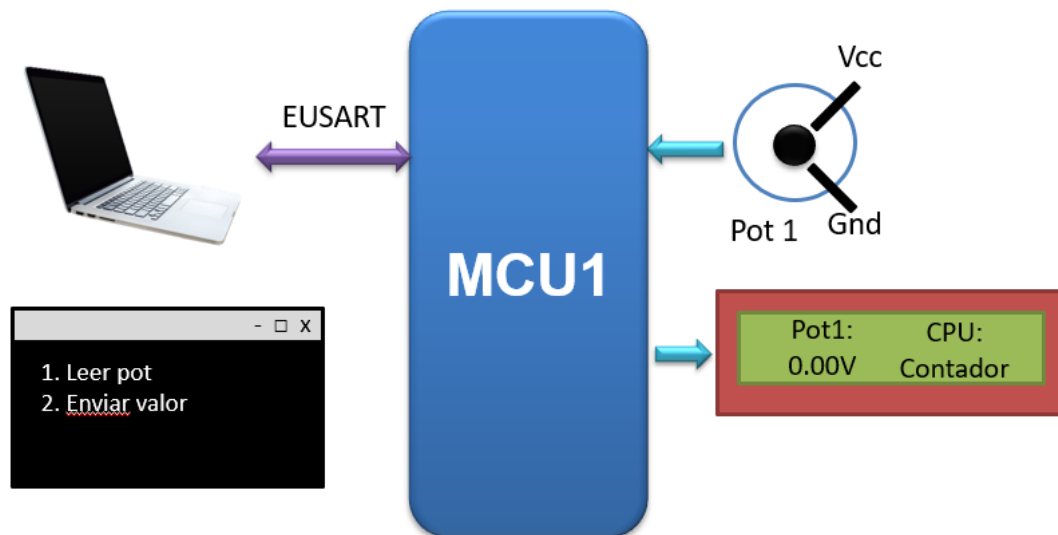


Laboratorio 2 – LCD

Utilizando el microcontrolador PIC16F887 con el compilador XC8



Pre-lab (20%) – Físico/Simulado

*Se sube antes del inicio del laboratorio en canvas en formato *.zip con el nombre prelab.*

Implemente una rutina la cual obtenga a partir de un potenciómetro el valor de voltaje y lo despliegue en la LCD como se muestra en la figura del circuito. **Deberá utilizar su librería de ADC y su librería de 4 BITS para inicializar la LCD.**

Lab (30%) - Físico

Deberá mostrarlo al catedrático o auxiliar durante el tiempo del laboratorio.

Obtenga el valor de voltaje de un potenciómetro haciendo su propia función de map en su librería:

```
int map (int val, int minx, int maxx, int miny, int maxy);
```

Despliegue el voltaje en la LCD como se muestra en la figura del circuito.

Deberá utilizar su librería de 8 BITS para inicializar la LCD.

Post-lab (40%) - Físico

*Se entrega después del tiempo de laboratorio según el portal. Deberá subir los entregables en formato *.zip con el nombre entregables.*

Implemente la comunicación USART con la computadora para poder enviar y recibir datos. Desde el microcontrolador tendrá que enviar el voltaje del potenciómetro y desde consola deberá poder incrementar un contador en el

microcontrolador enviando los caracteres “+” y “-” respectivamente. **Deberá utilizar su librería para la comunicación USART.**

La librería del UART deberá tener al menos las siguientes funciones:

- void **UART_RX_config** (uint16_t **baudrate**)
- void **UART_TX_config** (uint16_t **baudrate**)
- void **UART_write_char** (char **c**)
- char **UART_read_char** ()

Entregables (10%)

Deberá entregar en Canvas un archivo comprimido que contenga:

- El folder completo de su proyecto
- Link a Video explicando su código (Youtube). Deberá estar en Unlisted o Público ó no se puede ver el video la nota será cero.
- Su código debe estar dividido en secciones y bien comentado
- Link a repositorio en Github

Lista de pines

FUNCTION	PIN NUMBER	NAME	LOGIC STATE	DESCRIPTION
Ground	1	Vss	-	0V
Power supply	2	Vdd	-	+5V
Contrast	3	Vee	-	0 - Vdd
Control of operating	4	RS	0 1	D0 – D7 are interpreted as commands D0 – D7 are interpreted as data
	5	R/W	0 1	Write data (from controller to LCD) Read data (from LCD to controller)
	6	E	0 1 From 1 to 0	Access to LCD disabled Normal operating Data/commands are transferred to LCD
Data / commands	7	D0	0/1	Bit 0 LSB
	8	D1	0/1	Bit 1
	9	D2	0/1	Bit 2
	10	D3	0/1	Bit 3
	11	D4	0/1	Bit 4
	12	D5	0/1	Bit 5
	13	D6	0/1	Bit 6
	14	D7	0/1	Bit 7 MSB

Lista de comandos

COMMAND	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	EXECUTION TIME
Clear display	0	0	0	0	0	0	0	0	0	1	1.64mS
Cursor home	0	0	0	0	0	0	0	0	1	x	1.64mS
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	40uS
Display on/off control	0	0	0	0	0	0	1	D	U	B	40uS
Cursor/Display Shift	0	0	0	0	0	1	D/C	R/L	x	x	40uS
Function set	0	0	0	0	1	DL	N	F	x	x	40uS
Set CGRAM address	0	0	0	1	CGRAM address						40uS
Set DDRAM address	0	0	1	DDRAM address							40uS
Read "BUSY" flag (BF)	0	1	BF	DDRAM address							-
Write to CGRAM or DDRAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	40uS
Read from CGRAM or DDRAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	40uS

I/D 1 = Increment (by 1)
 0 = Decrement (by 1)
S 1 = Display shift on
 0 = Display shift off
D 1 = Display on
 0 = Display off
U 1 = Cursor on
 0 = Cursor off
B 1 = Cursor blink on
 0 = Cursor blink off
R/L 1 = Shift right
 0 = Shift left
DL 1 = 8-bit interface
 0 = 4-bit interface
N 1 = Display in two lines
 0 = Display in one line
F 1 = Character format 5x10 dots
 0 = Character format 5x7 dots
D/C 1 = Display shift
 0 = Cursor shift

Figura 3. Interconexión de la computadora con el microcontrolador

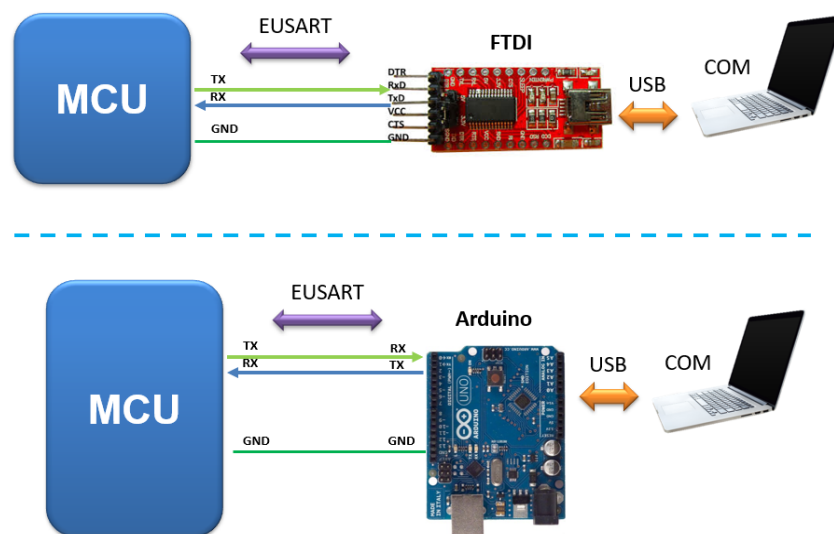


Figura 4. Código de Arduino para implementar FTDI con Arduino



Links de ayuda:

- <https://pythonprogramming.net/python-3-tkinter-basics-tutorial/>
- <https://pythonprogramming.net/tkinter-python-3-tutorial-adding-buttons/?completed=/python-3-tkinter-basics-tutorial/>
- <https://pythonprogramming.net/tkinter-tutorial-python-3-event-handling/?completed=/tkinter-python-3-tutorial-adding-buttons/>
- <https://www.programcreek.com/python/example/1568/serial.Serial>
- <http://www.varesano.net/blog/fabio/serial%20rs232%20connections%20python>