

**Reporte Práctica 1 - Parallel Port Management**

Diseño en sistemas de chip

Diego Armando Limón de León A01638247

Jose Miguel Figarola Prado A01632557

Giancarlo Franco Carrillo A01638108

Tecnológico de Monterrey

08 de abril del 2021

### Parallel Port

LCD es una pantalla que se destaca por ser plana y estar formada por píxeles que contienen moléculas de cristal de líquido, dichos píxeles se encuentran entre dos capas de cristales polarizados que son iluminados con una fuente de luz fija debido al paso de moléculas que se da en el proceso. Por lo tanto, dicha pantalla muestra las imágenes programadas por medio de la retroiluminación ya que permite la combinación de colores; no obstante, una desventaja de esta es que no puede brindar un negro absoluto causando contraste y efectos en la calidad de imagen.

El Keypad es un teclado de las máquinas de escribir que trae consigo una disposición de botones o teclas con el fin de que estas actúen como palancas o interruptores electrónicos que envíen información. Cabe mencionar que en base a la matriz se reutilizan los pines con respecto a la cantidad de botones que son presionados.

Cuando se presiona una tecla o un botón la señal no cambia 1 a 0 o viceversa de manera limpia, es decir, la señal no cambia correctamente y se pueden mandar datos innecesarios y/o irrelevantes. Por lo tanto es necesario generar un debounce, el debounce nos permite limpiar la señal y todo el ruido que se genera en esta señal. Básicamente genera un “delay” que permite que la señal se pase correctamente sin que la señal tenga ese ruido.

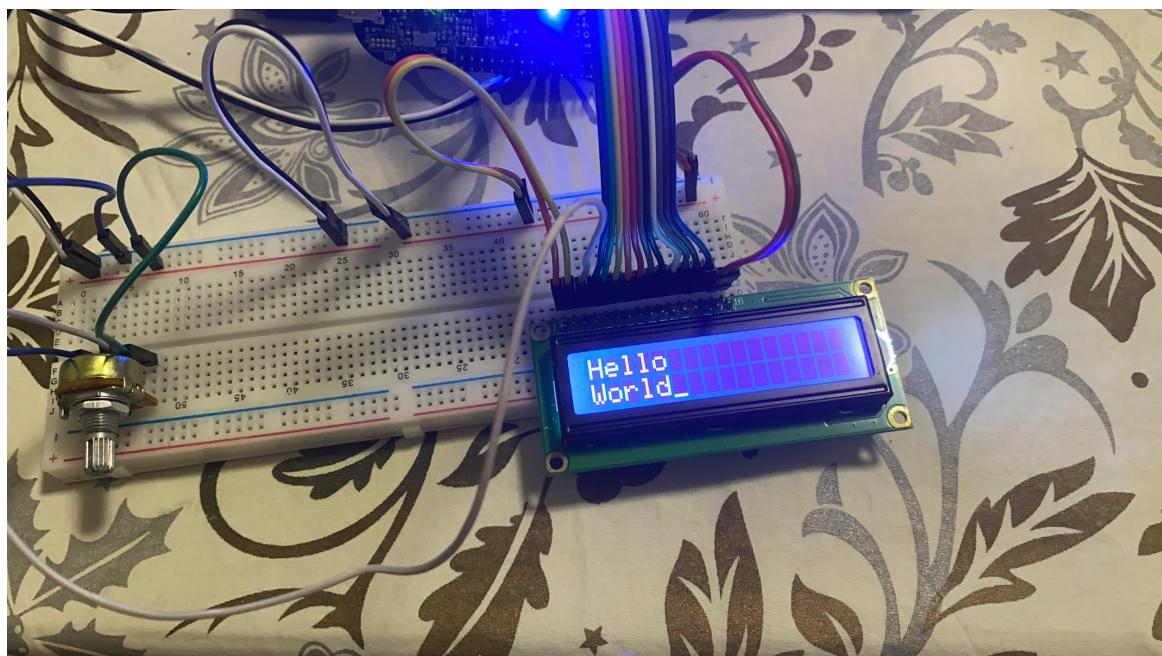
### Parte 1

En este apartado usamos una LCD para desplegar el texto que mandamos desde el código, así mismo utilizamos un potenciómetro para poder generar el contraste y así poder visualizar los datos en la pantalla. Primeramente tenemos una pantalla LCD con circuito de configuración 8-bit de datos, 5x7, 1 línea de display, el cual nos despliega el “Hello World” solamente en la primera línea.



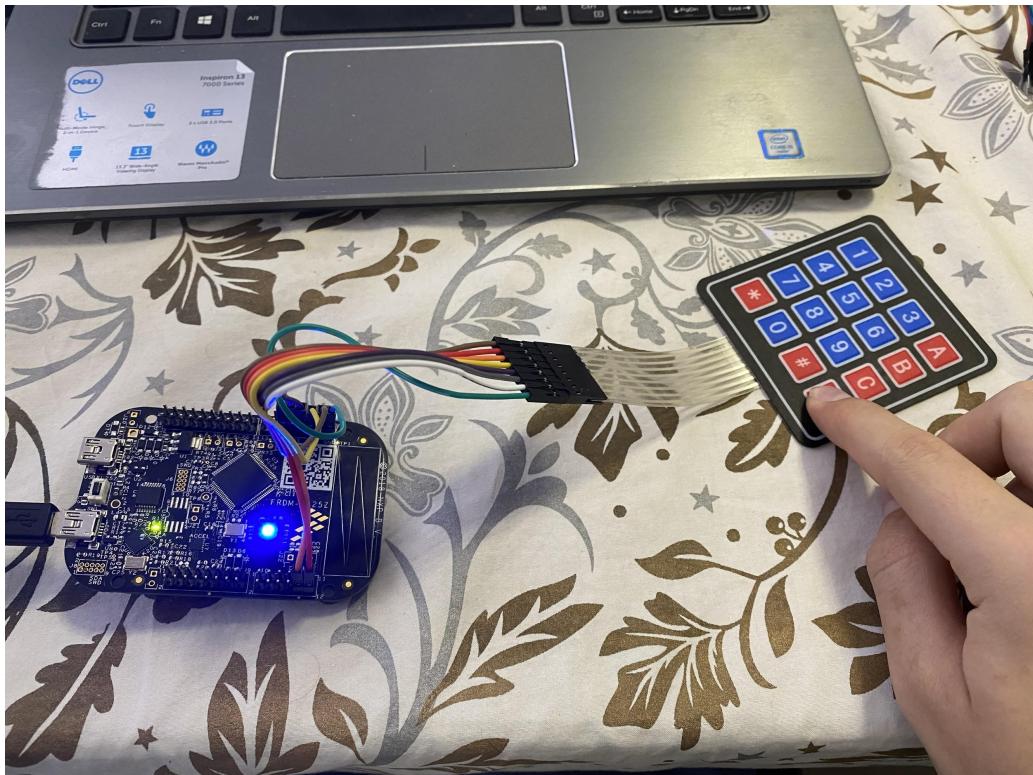
## Parte 2

Pantalla LCD con circuito de configuración de 4 bit de datos, 5x7 y 2 líneas de display, usando y leyendo nibles para decodificar correctamente el texto y poder desplegarlo en la pantalla. Igualmente que en la primera parte, desplegamos un “Hello World” usando las dos líneas de las pantallas.



## Parte 3 y Parte 4

Así mismo podemos implementar un Keypad 4x4, para poder entender su funcionamiento es importante generar una función de reciba la señal de alguno de sus botones, en este caso el LED color azul prende si una tecla manda una señal, cabe resaltar que para el uso de switches o botones es necesario implementar un debounce para quitar el ruido de las señales de estos switches/botones.



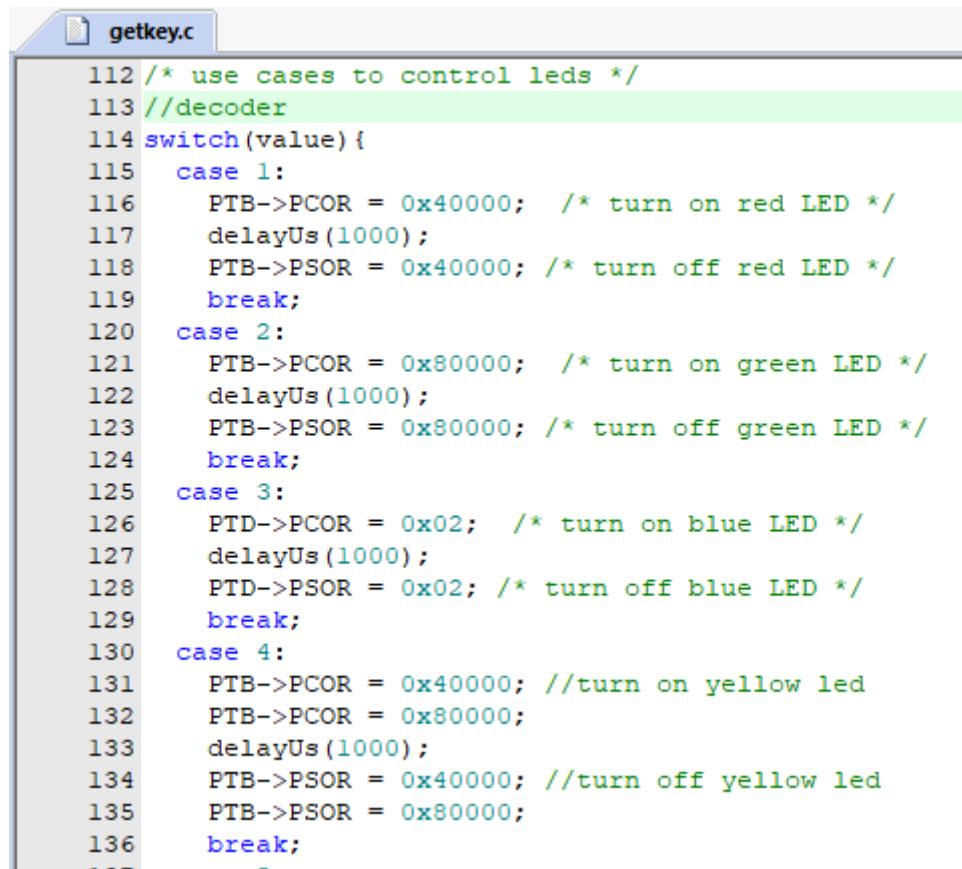
```

14 int main(void)
15 {
16     unsigned char key;
17
18     keypad_init();
19     LED_init();
20
21     while(1)
22     {
23
24         key = keypad_getkey();
25         delayUs(20); //debouncing
26         LED_set(key); /* set LEDs according to the key code */
27

```

Ya que tenemos implementada la función que nos permite recibir una señal del teclado matricial, es importante ahora sí, generar el debounce que lo podemos definir como

un delay para que la señal se propague completamente y podamos recibir bien el dato que queremos. Finalmente generamos un decoder que por cada tecla diferente que se presione, el LED genera un color distinto, es decir, podemos recibir una tecla distinta y generar un dato (como podemos ver en el código siguiente). Así podemos saber cual tecla es la presionada de manera visual y física, sin solo mandar una misma señal, es decir, solo prender el LED azul.



```

112 /* use cases to control leds */
113 //decoder
114 switch(value){
115   case 1:
116     PTB->PCOR = 0x40000; /* turn on red LED */
117     delayUs(1000);
118     PTB->PSOR = 0x40000; /* turn off red LED */
119     break;
120   case 2:
121     PTB->PCOR = 0x80000; /* turn on green LED */
122     delayUs(1000);
123     PTB->PSOR = 0x80000; /* turn off green LED */
124     break;
125   case 3:
126     PTD->PCOR = 0x02; /* turn on blue LED */
127     delayUs(1000);
128     PTD->PSOR = 0x02; /* turn off blue LED */
129     break;
130   case 4:
131     PTB->PCOR = 0x40000; //turn on yellow led
132     PTB->PCOR = 0x80000;
133     delayUs(1000);
134     PTB->PSOR = 0x40000; //turn off yellow led
135     PTB->PSOR = 0x80000;
136     break;
137 }

```

## Video

Ya que tenemos el reporte, se anexará un video donde mostramos el comportamiento de cada uno de los códigos y poder mostrar su funcionamiento. Link del video:

<https://youtu.be/GFCviXtU0J4>

## Referencias

Pérez, J., Gardey, A. (2018). *Definición de LCD*. Definición.de. Recuperado de:

<https://definicion.de/lcd/>

Electrónica y Circuitos. (s.f.). *Simulación de manejo de keypads (teclados) en arduino*.

Recuperado de: <http://electrocirc.blogspot.com/p/simulacion-de-manejo-de-keypads.html>