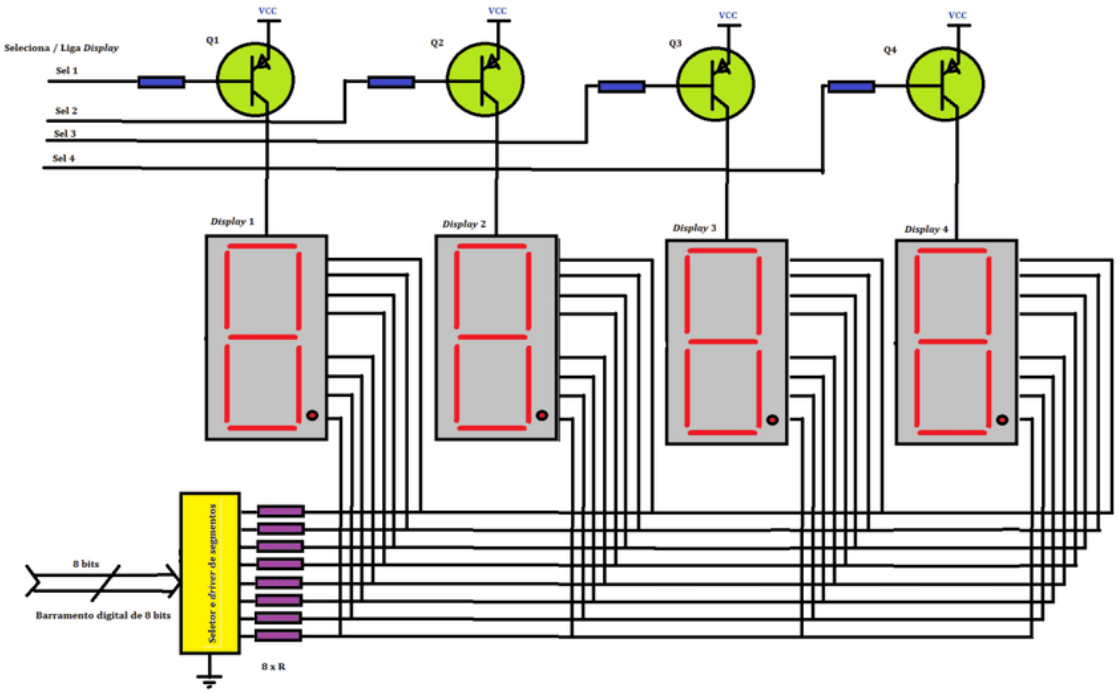
Trabajo práctico de laboratorio Nro 4:
Teclado matricial y display 7 segmentos

Para el presente trabajo práctico se considerará el teclado y los displays conectados según cada grupo considere pertinente, y por lo tanto no se mencionarán direcciones de Puerto y Pin en donde se encuentren conectados los periféricos. Se recomienda el uso de diodos en las columnas del teclado matricial, con el cátodo apuntando a la salida del microcontrolador, para evitar cortocircuitos en la línea, y proteger los pines del microcontrolador que se utilicen para los displays 7 segmentos mediante un circuito similar al que se muestra (los displays del gráfico son ánodo común, en caso de ser cátodo común el transistor debería ser NPN y el emisor iría a masa):



En donde el integrado amarillo es un buffer, y los transistores están colocados en ese sentido considerando displays de ánodo común. En caso de tratarse de displays cátodo común se utilizarán transistores NPN en lugar de PNP, y el emisor irá a masa en lugar de VCC. Estos circuitos son simplemente a modo de ejemplo, pudiendo utilizarse otros según consideren conveniente.

1. Realizar un programa que detecte la opresión de la tecla conectada al pin P0.4. Ante la opresión de dicha tecla se deberá incrementar un contador, cuyo valor se deberá ver reflejado EN TODO MOMENTO en los displays 7 segmentos. Para la, lectura de la tecla realizar el antirebote correspondiente, y para la escritura en los displays realizar una primitiva Display(uint32_t nro), que reciba el número que se quiere mostrar y lo convierta en un array global, en donde cada posición del array contendrá un dígito de dicho número. Luego realizar una función driver BarridoDisplay(), que será invocada dentro del Systick, y mostrará cíclicamente, en cada uno de los displays el dígito correspondiente.

2. Modificar el ejemplo anterior de manera de invocar a la función BarridoDisplay cada N llamados a la función de interrupción Systick_Handler. El objetivo es encontrar la menor frecuencia con la que se puede invocar a la función BarridoDisplay() dentro del Systick, sin notar que los dígitos parpadeen.
3. Modificar el ejemplo anterior de manera de que la lectura de la tecla se haga ahora desde el teclado matricial. En el driver solo se deberá modificar la función de lectura de la tecla (BarridoTecladoHW), mientras que en la aplicación se le pueden asignar diferentes funcionalidades a cada tecla. Por nombrar algunas, podrían ser: incrementar el valor de la variable en 1, en 10, en 100, decrementarla en la misma cantidad, resetear el contador, modificar la frecuencia en el llamado a la función BarridoDisplay, etc. Elegir las funcionalidades que se consideren para cada botón.
4. Modificar la función primitiva (solo la primitiva) para que en lugar de manejar un solo display de N dígitos de longitud, la aplicación maneje 2 displays de N/2 dígitos de longitud (por ejemplo, si tengo 6 displays de 7 segmentos, podría pensarlos como 2 bloques de 3 dígitos cada uno). A partir de esta modificación, utilizar uno de los displays para mostrar la variable que se incrementa y decrementa, y en el otro display mostrar el tiempo (en segundos) desde que se enciende el equipo. El prototipo de la función primitiva quedaría como:

   void Display (uint8_t nro_Display, uint32_t valor);

   En donde nro_display es el número del display donde quiero mostrar la información, y valor es el número que quiero que se vea en el display.