Primer parte

Es una placa Arduino con dos displays de 7 leds c/u. Lo que se hizo fue conectarlas a los pines de salida/ entrada digital. Los pines respectivos de los leds son de salida digital, los de pantalla de salida digital (ya que sus valores oscilan entre 0 y 1 nomas, no lee voltajes) y los botones de entrada digital configurados con un pullup para no tener que colocar manualmente el resistor. Estos botones están conectado a tierra y a su pin correspondiente.

Primero definimos constantes que representaran nuestros pines

*// C++ code*

*//*

*//Define de LEDs*

*#define c 5*

*#define d 6*

*#define e 7*

*#define g 8*

*#define f 9*

*#define a 10*

*#define b 11*

*//Define de pantalla LED seleccionada*

*#define UNIDAD A4*

*#define DECENA A5*

*//Define de Botones*

*#define BOTONSUMA 2*

*#define BOTONRESET 3*

*#define BOTONRESTA 4*

Luego aquellas variables que necesitamos definirla por fuera de loops o de las funciones y lo hacemos al principio para tenerlas de manera ordenada con valores 0 o 1.

Primero configuramos que sucede con los botones:

Primero configuramos el loop, que se encarga de leer lo que sucede en los pins de la placa. Aca usamos una función que lee que botón se presiona (llamada botonPresionado() ) y en función en que se presiona, el contador varía e imprimimos por los leds el resultado del contador, usando la función creada printContador()

Para armar las funciones mencionadas anteriormente necesitamos otras funciones que las complementen.

printDigit(digito): Maneja los LOW y HIGH de los pines. Usamos un switch para configurar los leds que se prendan cuando queremos mostrar de 0 a 9.

prendePantalla(digito):Prende y apaga los pines conectado en “común” en los leds de 7 digitos, alternando uno y el otro con un delay muy chico. Esto se hace para evitar que titilen los leds.

Usando las funciones descriptas anteriormente, armamos printDigito(digito). Esta función con printDigit, imprime la decena, prendePantalla el led de decena, luego se apaga el led (unos milisegundos) usando la misma función, luego con printDigit. Tanto como para imprimir la decena y luego la unidad, añadimos una serie de cálculos donde configuramos la salida en int, para que siempre nos tome los números sin coma y pueda representar de una manera adecuada.

Por último, configuramos botonPresionado() que lee a los botones y lo configuramos de manera que toquemos el botón pero si lo mantenemos presionado, no siga leyéndose. Esto se logra comparando estados previos del botón con lo leído actualmente. Dependiendo de que lea, devuelve un valor que será analizado en el primer loop configurado anteriormente.

Segunda parte:

Agregamos un botón switch que tiene tres conexiones. Una va a ir hacia un pin de entrada digital, otra conectado a la tierra y el ultimo conectado a un cable que relaciona los tres botones. La única diferencia es que tenemos que configurar que sucede cuando nuestro switch está activado. Yo relacioné el switch con un pin de entrada digital y cuando se prende, va a ir iterando un for que imprime aquellos números que sean primos, que pase por un print digit. El problema que identifiqué acá, es que la decena titila, pero la unidad no. Jugando con el Arduino, me di cuenta que “no le gusta” que se usen muchos if anidados para las salidas digitales, entonces habría que encontrar una solución que no incluya tantas estructuras condicionales para evitar este error.

Cuando se apague el switch, la botonera se activa.

Luego se pedía agregar algo más. En mi caso, elegimos un sensor de temperatura y lo anidé de la siguiente forma:

Lo que cambia es que agregamos un segundo switch donde estaba conectada la botonera hacia el primer switch. Este segundo switch se activa cuando se apaga el primer switch. Cuando se prende el segundo switch, se activa la botonera y cuando se apaga este segundo switch, se activa la lectura de temperatura y es CONSTANTE. Si movemos el lector de temperatura, se actualiza constantemente.

Lo que se tuvo que actualizar es el lector, agregamos uno más. Y actualizamos el printContador(digito) para que lea valores mayores a 100 y valores negativos. En el caso de los valores negativos, en la centena configuramos para que devuelva un signo negativo y hacemos los mismos cálculos pasando por un auxiliar=-contador y después haciendo los mismos cálculos.

En el caso de las centenas, decimos que si el contador es mayor a 100, los cálculos se modifican pero el principio es el mismo: primero obtenemos la centena, y así hasta la unidad.