Trabajo Práctico N°2

Alarma Hogareña

22.11 Electrónica III

Grupo N°2

Legajo N°	Nombre
61428	Kevin Amiel Wahle
61430	Francisco Basili
61431	Nicolás Bustelo
61433	Sergio Andrés Peralta
61463	Bautista Schneeberger

18/11/2021

Índice

1. Introducción

El objetivo de este informe consiste en el armado de una alarma aplicando los conceptos aprendidos en la materia. La alarma debe avisar mediante leds si se detecta que un sensor esta activo, o si se encuentra configurada en algún estado en particular. La alarma dispone de un teclado mediante el cual el usuario introduce un código de 4 dígitos para activar y desactivar el modo de funcionamiento en el que se encuentra. En la siguiente figura se encuentra el esquema de todos los módulos.

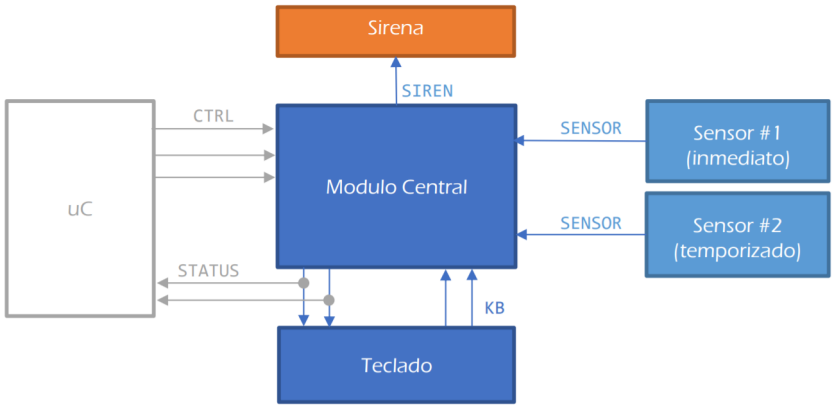

Figura 1: Diagrama de los componentes del sistema.

2. Hardware

2.1. Teclado

Como no se contaba con un shift register paralelo-serie, se optó por realizar un encoder con flip flops de sincronización para la parte del teclado. Las conexiones son las siguientes:

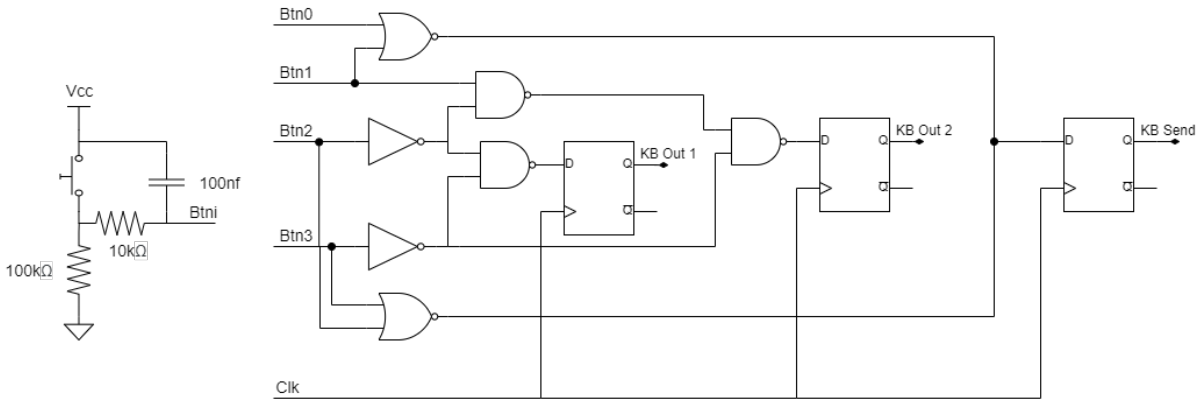

Figura 2: Circuito del teclado.

La tabla de verdad correspondiente al circuito anterior es:

Btn_0	Btn_1	Btn_2	Btn_3	KB Out 2	KB Out 1	KB send
0	0	0	0	0	0	0
1	0	0	0	0	0	1
0	1	0	0	1	0	1
0	0	1	0	0	1	1
0	0	0	1	1	1	1

Tabla 1: Tabla de verdad del teclado.

Se calculó la frecuencia máxima como la frecuencia máxima del flip flop, segun el datsheet. En éste se puede observar que no se encuentra una f_{max} para $3.3V$. Entonces, a modo de precuación se tomo la $f_{max_{minima}}$ para una tension de $2V$ y una temperatura de $25C$. Se obtuvo $f_{max} = 6\,MHz$. Sin embargo carece de sentido utilizar un clock con una frecuencia tan alta, cuya señal pasa por cables y un protoboard. Se optó por utilizar una frecuencia de clock de $f_{clk} = 10\,KHz$.

Se calculó el tiempo de metaestabilidad como

$$\begin{aligned} t_r = t_{clk} - t_{comb_{max}} - t_{setup_{max}} &= \frac{1}{f_{clk}} - [t_{not} + 2 \cdot t_{nand}] - t_{setup_{max}} \\ &= \frac{1}{10\,KHz} - [19ns + 2 \cdot 18ns] - 20ns \\ &\approx 0.1\,mseg \end{aligned}$$

El tipo de modelo que se aplicó, es el que se muestra a continuación:

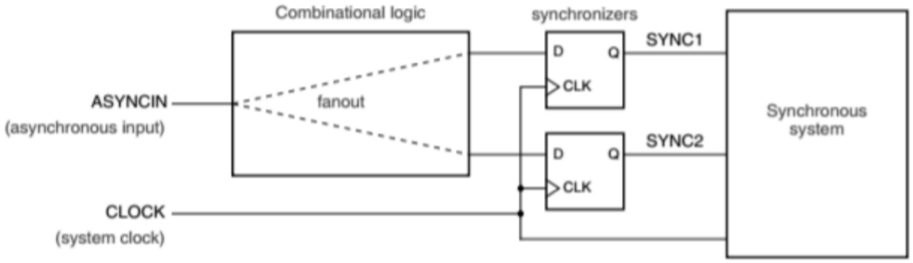

Figura 3: Modelo Aplicado

A pesar de los inconvenientes que este tipo de configuración conlleva, esto no va a ser tan grave, debido a la bajísima frecuencia de la acción de pulsar un botón. Para mayores frecuencias se podría realizar una mejora agregando sincronizadores a la entrada de los botones del teclado. Sin embargo esto no se llevó a cabo debido a las limitaciones de componentes para el trabajo práctico.

2.2. Mediciones de los pulsadores

En las siguientes figuras, se muestra la respuesta ante la activación de los pulsadores. Se puede ver, que a pesar de que el circuito anti rebote con capacitores, evita justamente los rebotes, provoca una respuesta muy lenta que termina siendo contra produciente ya que a la salida del circuito encoder, comienza a oscilar en las zonas de valor de entrada indefinido entre 0V y 3.3V.

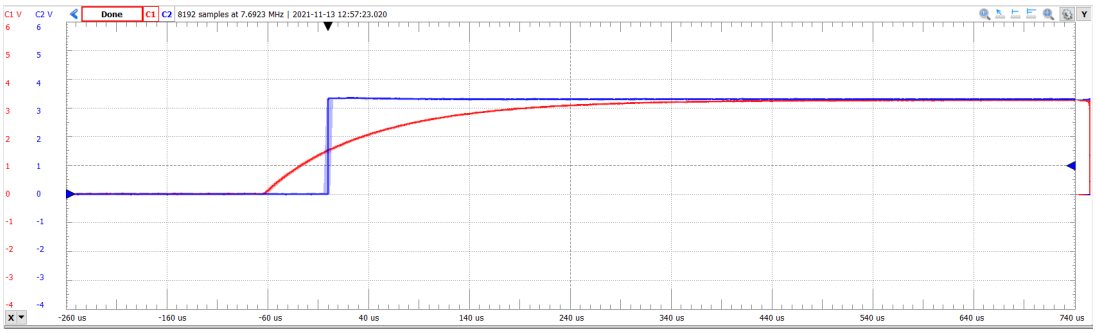

Figura 4: Señal del pulsador y salida del encoder. **Con capacitor** 10nF

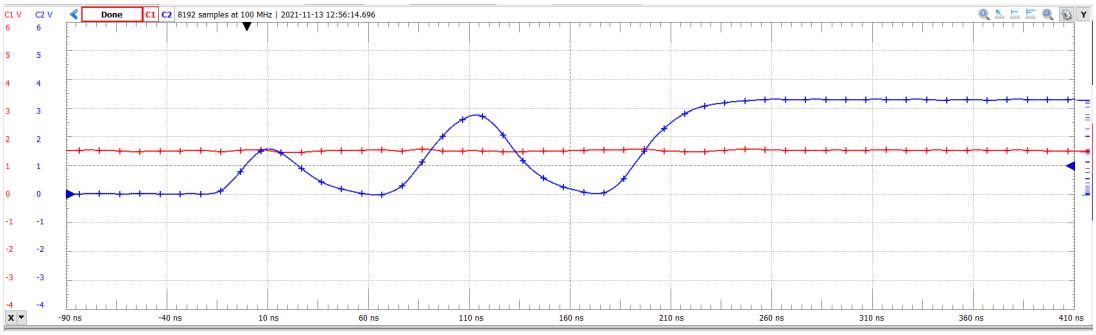

Figura 5: Señal del pulsador y salida del encoder. **Con capacitor** 10nF

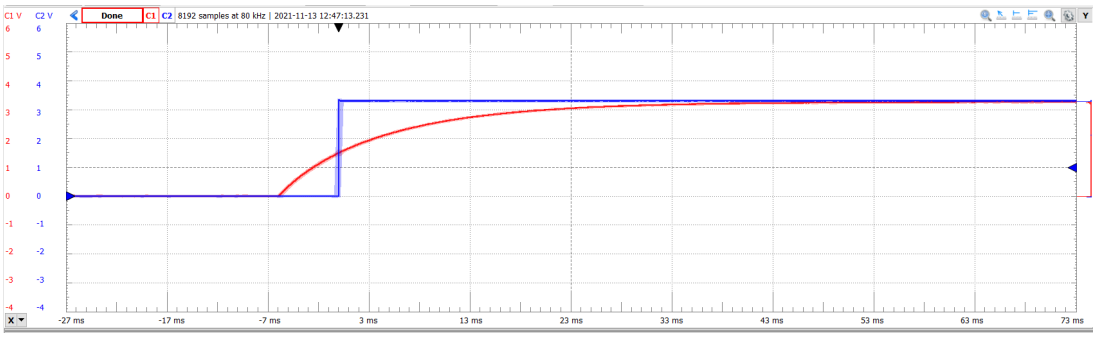

Figura 6: Señal del pulsador y salida del encoder. **Con capacitor** 1uF

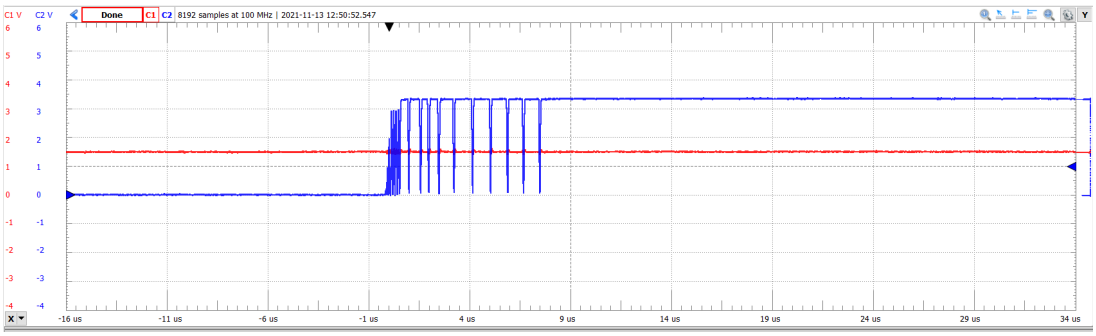

Figura 7: Señal del pulsador y salida del encoder. **Con capacitor** 1uF

2.3. LEDs

La conexión para los leds utilizada fue la siguiente:

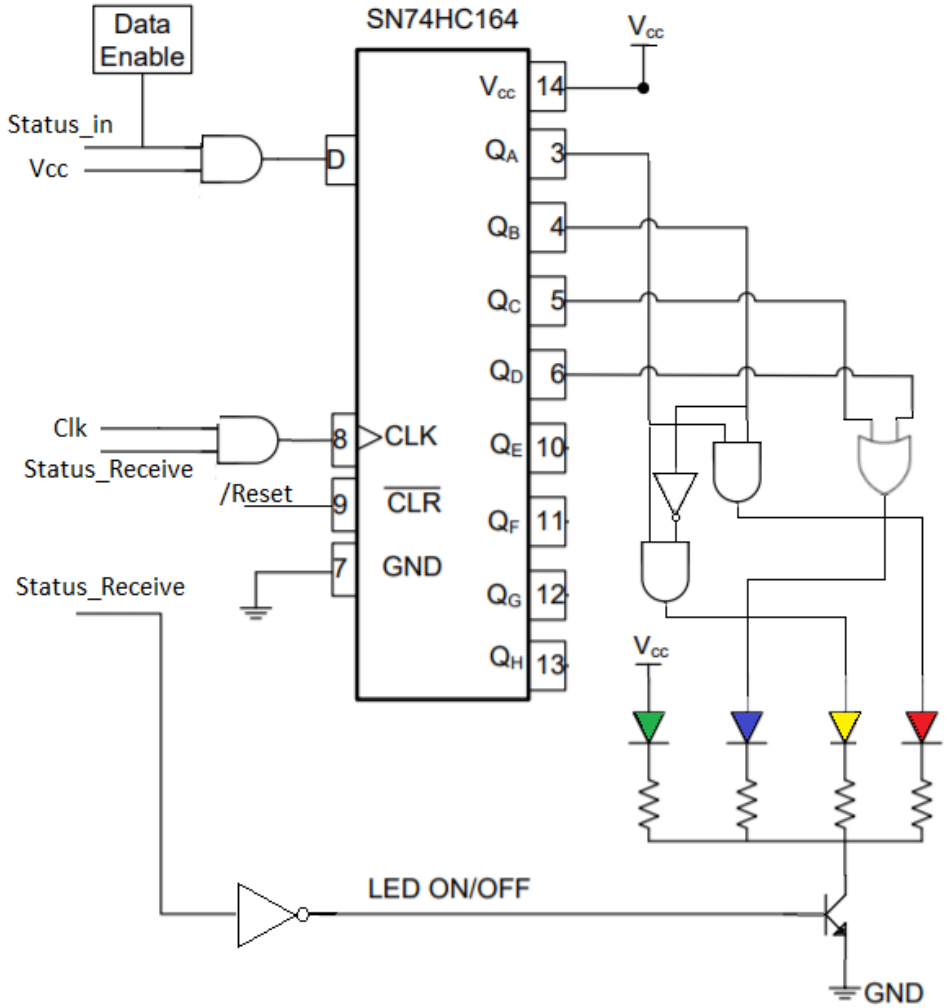

Figura 8: Circuito de LEDs.

El circuito consta de un shift register paralelo-serie al cual le llega por entrada serie los leds que se deben encender. Los pulsos del canal serie viajan a lo largo del shift register hasta que termina la señal (es decir Status_Receive = 0). En ese momento se fijan las salidas del shift register y quedan encendidos los leds correspondientes, lo cuales se mantienen en el mismo estado hasta el nuevo pulso de Stauts_Receive. El transistor evita que se enciendan los LEDs mientras la señal serie de la entrada viaja por los shift register.

3. Descripción del módulo central

Se realizó un diagrama de estados para facilitar la programación del módulo central:

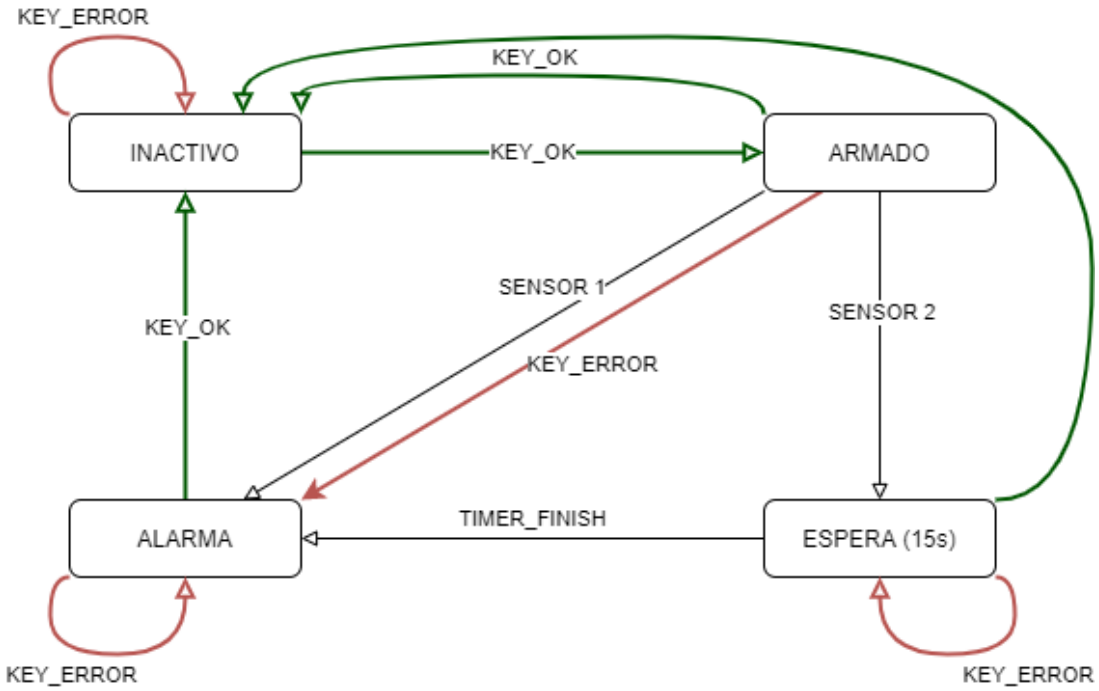

Figura 9: Diagrama de estados.

Debido a que el sintetizador de *Aldec* es distinto al de *Radiant* se programó directamente en *Radiant*.

Se llevó a cabo el código en el lenguaje *Verilog*, el cual se puede encontrar en su totalidad el repositorio de Github.

Se decidió programar distintos módulos individuales que luego instanciaríamos en un módulo central, el cual sería una máquina de estados que los integraría y uniría a todos ellos.

El primero de los módulos es un *timer*. Este consta de un contador de pulsos customizable. Es decir, que se le ingresa una señal de clock y la cantidad de pulsos a contar. Cuando la cuenta termina, enciende un pulso indicando que finalizó el contador.

El segundo de los módulos es un transmisor serial (serialOut) que recibe el mensaje a enviar y el indicador de inicio de transmisión y envía por dos canales distintos el aviso de que se está transmitiendo la información por uno y la información por otro. A su vez, este módulo fue interfaceado por otro (easySerialOut), el cual lo adaptaba para cortar el canal de transmisión durante cierta cantidad de ciclos customizable, para que el encendido de los LEDs sea apreciable.

El tercer y último gran módulo es un Verificador de Passwords (codeCkeck) que recibe cuándo se está enviando un botón, cuál botón se está apretando y cuál es la contraseña correcta. Su única salida es un canal en el que se indica si la contraseña es correcta ("KEY_OK"), incorrecta ("KEY_ERROR") o si no se termino de cargar ("NO_KEY"). Una vez que se recibe una contraseña completa, la salida se pone en "KEY_OK.º "KEY_ERRORz al paso de un ciclo de clock se la fuerza a "NO_KEY"para no afectar el desarrollo de la máquina de estados.

4. Funcionamiento en tiempo real

Se combinó la parte de software y de hardware mediante la FPGA UPDuino, tomando como sensores a los botones, y se obtuvo el resultado que se puede observar en el siguiente video.

La implementación de hardware se puede ver en la siguiente figura.

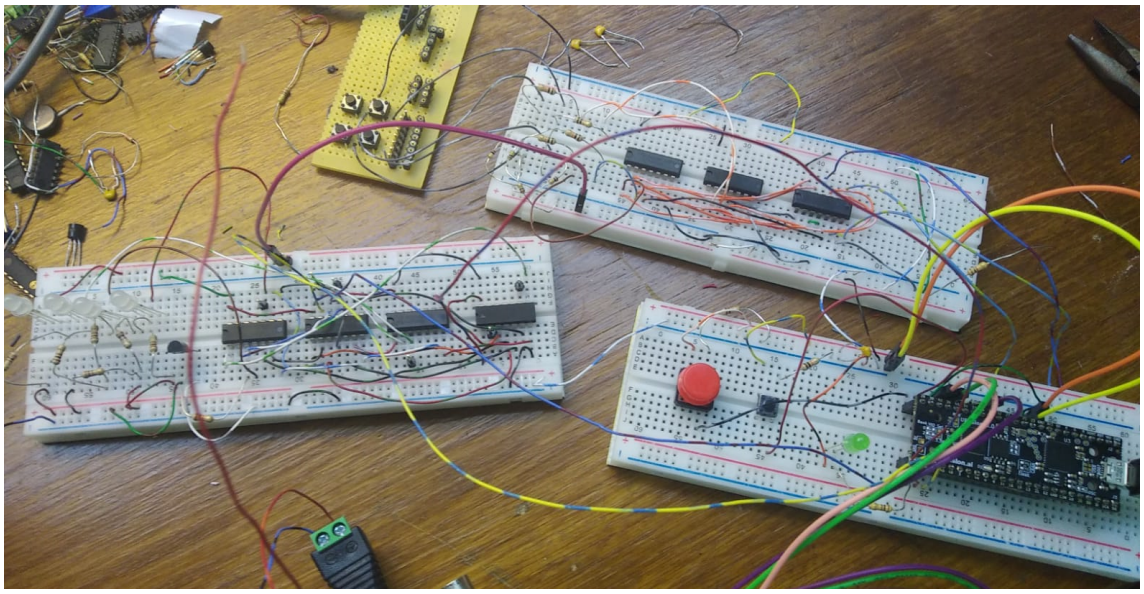

Figura 10: Circuito Físico.