

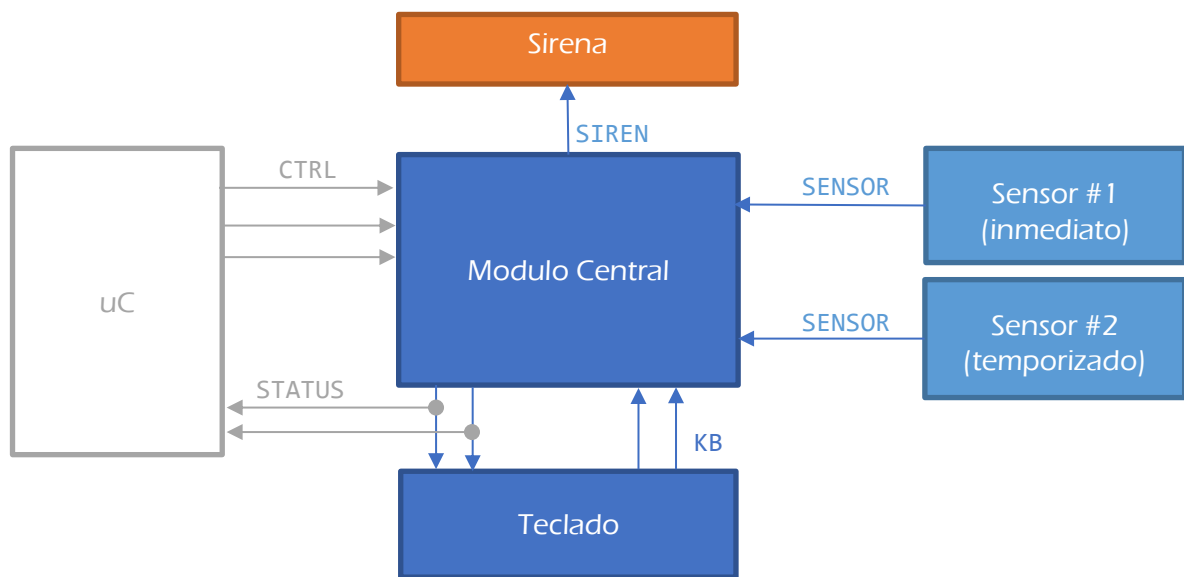
1 Descripción del sistema

Como parte de la implementación de un sistema de alarma hogareña “inteligente” IoT, su equipo tiene que diseñar ciertos módulos para su funcionamiento.

La conectividad a internet (IoT) se realizará con un microcontrolador [que queda fuera de este trabajo], sin embargo, debido a que el sistema básico de alarma tiene que tener una confiabilidad excepcional, se decidió implementarla con una FPGA y componentes electrónicos discretos.

La función básica de la alarma es avisar mediante una sirena si se detecta que un sensor esta activo. La alarma dispone de un teclado mediante el cual el usuario introduce un código de 4 dígitos para activar y desactivar la sirena.

A continuación, se encuentra un diagrama de los componentes del sistema

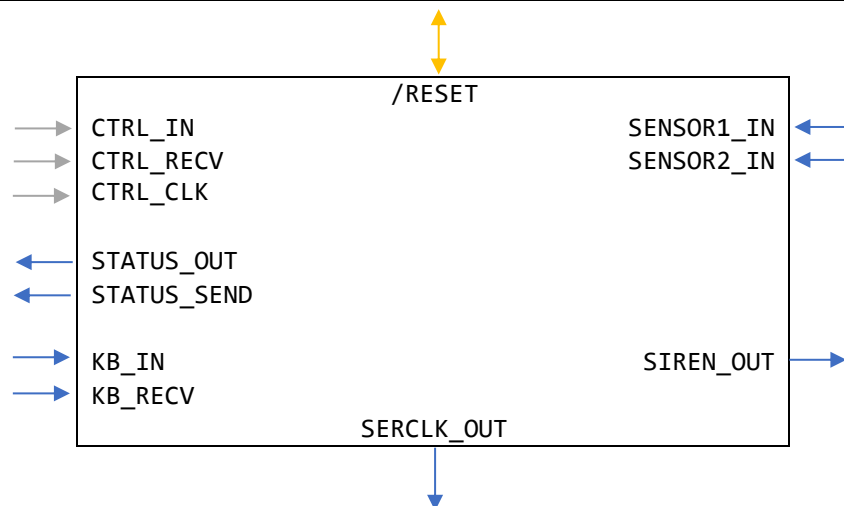


Resumen de conexiones:

MODULO	DESCRIPCIÓN	ENTRADAS	SALIDAS
Central	Modulo central de alarma Implementado en FPGA o FPGA + Lógica	<ul style="list-style-type: none"> • Entrada serie control • Entrada 'serial receive' control • Entrada de clock para ctrl • Entrada serie teclado • Entrada 'serial receive' teclado • Entrada sensor inmediato • Entrada sensor temporizado 	<ul style="list-style-type: none"> • Salida serie estado • Salida 'serial send' • Salida sirena • Salida CLK bus

Teclado	Entrada de código, LEDs de estado Implementado con lógica discreta	<ul style="list-style-type: none"> • Entrada serie estado alarma • Entrada 'serial receive' • Entrada reset • Entrada CLK bus 	<ul style="list-style-type: none"> • Salida serie teclado de teclado • Salida 'serial send'
Sensores	Sensor activo alto, salida paralela	<ul style="list-style-type: none"> • N/A 	<ul style="list-style-type: none"> • Salida sensor 1 bit (activo alto)
Sirena		<ul style="list-style-type: none"> • Encendido (activo alto) 	<ul style="list-style-type: none"> • N/A

1.1 Módulo Central



El módulo central se encarga de controlar el comportamiento básico de la alarma. Tiene tres estados principales: inactivo, armado, y alarma:

- **Inactivo:**
 - Pasa al modo armado al introducir el código en el teclado
 - Mantiene actualizada la palabra de estado según el estado de los sensores
- **Armado:**
 - Mantiene actualizada la palabra de estado según el estado de los sensores
 - En caso de activación de una entrada de sensor inmediata (1), pasa al modo alarma
 - En caso de activación de una entrada de sensor temporizada (2), pasa al modo alarma si no se introduce el código de desactivación en 15 segundos.

- En este caso, con el código correcto, pasa a modo inactivo
 - Con el código incorrecto, continua el contador de tiempo y permite intentar nuevamente
- Con el sistema activo
 - Introducir el código correcto pasa al modo inactivo
 - Introducir el código incorrecto pasa al modo alarma
- **Alarma:**
 - Activa la salida sirena
 - Mantiene actualizada la palabra de estado según el estado de los sensores
 - Si se introduce el código correcto pasa al modo inactivo
 - Se mantiene en este estado aun si los sensores se apagan

Al iniciar por primera vez, la alarma debe tener un código de desactivación por defecto **distinto de 0000**.

Nota: Se deja a su criterio la línea de RESET del módulo principal. Puede ser una entrada (y se utiliza para resetear todo el sistema), o una salida (que alimenta al reset de otros módulos como el teclado)

Opcional: Se puede extender el código o realizar otras modificaciones para expandir el funcionamiento del sistema.

1.1.1 ESTADO

El módulo central mantiene una palabra de estado que envía mediante la salida serie. Los bits de salida deben estar sincronizados con el CLK externo. En el periodo de clock anterior a enviar el bit 0, la salida 'serial send' se activa. Los valores de cada bit de la palabra son según la tabla:

Bit 3	Bit 2	Bit 1	Bit 0
Estado sensor 2	Estado sensor 1	1: Modo Alarma 0: Modo Armado o Inactivo	1: Modo Armado o Alarma 0: Modo Inactivo

El módulo central dispone de una entrada en la que el teclado envía los botones presionados mediante una conexión serie. El teclado envía el valor del botón presionado en 4 bits. El teclado envía un pulso en 'serial send' antes de enviar el primer bit del valor del botón. Ver más en la sección **Teclado**.

El módulo central debe enviar la palabra de estado de manera frecuente, no solo cuando cambia.

Opcional: Se pueden agregar mayor cantidad de entradas (sensores), así como otras modificaciones que expandan el sistema. En el caso de modificarlas para expandir el funcionamiento, se recomienda mantener las palabras serie de 4 u 8 bits.

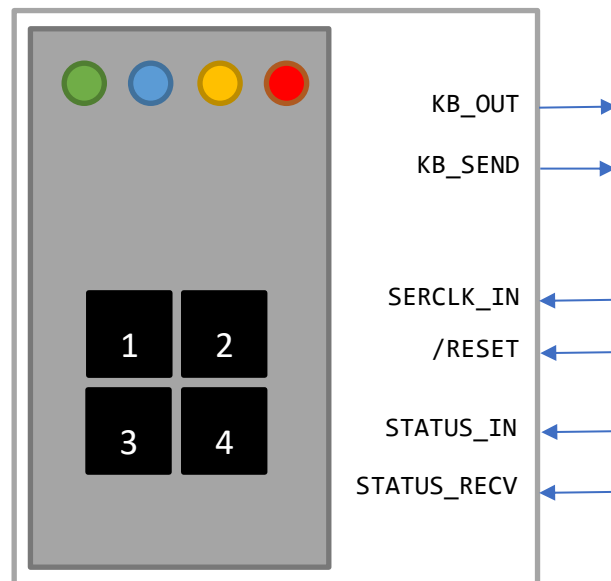
1.1.2 CONTROL EXTERNO

El módulo central puede recibir varias instrucciones mediante la entrada serie. El módulo central comienza a recibir la palabra de control de 4 bits al recibir un 1 en la entrada de 'serial receive'. Ciertos comandos interpretan los bits siguientes a los primeros 4 como parámetros

COMANDO	DESCRIPCIÓN	PARÁMETROS	CONDICIONES
0001	Pasar a modo alarma		Desde modo inactivo o modo alarma
0010	Pasar a modo armado		Solo desde modo inactivo
0011	Pasar a modo inactivo	4x [4 bits]: código	Solo pasa a modo inactivo si el código es correcto
0100	Configurar código	4x [4 bits]: código	Solo en modo inactivo

Nota: Esta funcionalidad es para la parte bonus (opcional)

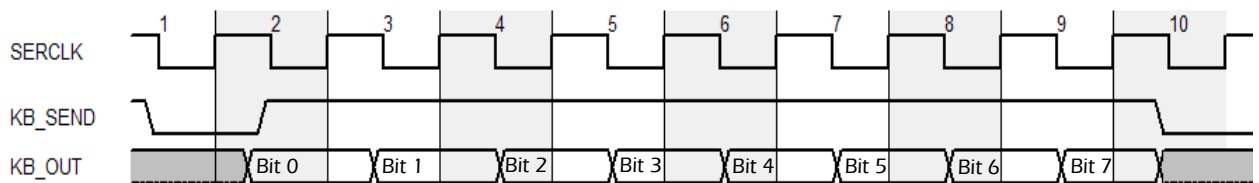
1.2 Teclado



El teclado consiste en una serie de botones que al presionarse, envían el valor del botón presionado mediante la salida serie. Se usan 4 bits para representar 16 posibles botones.

El teclado envía un pulso en 'serial send' antes de enviar el primer bit del valor del botón, de manera que el módulo central pueda recibir el valor correctamente. Los bits se envían sincronizados con el CLK externo.

Una posible implementación de la comunicación serie: (no la única)



Nota: Para este trabajo, el teclado se puede asumir de 4 botones, o más si se desea.

Posibles implementaciones: se puede diseñar una máquina de estados para esperar un botón, y en base a eso generar la salida. Considere que tiene que implementarse en lógica discreta.

O bien puede utilizar FFs para conservar el valor de un botón presionado, y use ese bit de 'memoria' para la entrada de un shift register para la comunicación serie. Una vez enviado el valor serie, resete los FFs para esperar el próximo botón.

El teclado tiene al menos tres leds de estado:

- Alimentación (verde)

- Sensor activo (azul)
- Modo armado (amarillo)
- Modo alarma (rojo)

Los tres últimos leds se encienden basados en los valores de la palabra de estado recibida mediante STATUS_IN; el de sensor debe encenderse si cualquier sensor está detectando.

Opcional: Se pueden agregar leds para los sensores individuales si lo desea.

1.3 Sensores

Dispositivos asincrónicos que indican actividad mediante un '1' lógico en su señal de salida

1.4 Sirena

Dispositivo que enciende un aviso visual y/o sonoro al recibir un '1' lógico en su entrada

2 Implementación

1. Implemente en Verilog el módulo central
 - Utilice los conceptos de máquinas de estado
 - Use un diseño modular – Utilice submódulos en Verilog para implementar la distinta funcionalidad de la central
 - Utilice un clock interno para la progresión de estados, y uno distinto para la comunicación serie externa
 - a. Se recomienda implementar primero **sin la funcionalidad de control externa**. Diseñe la primera iteración considerando que se debe agregar esto eventualmente
 - i. Muestre los diagramas de estados de las máquinas de estado utilizadas
 - ii. Muestre diagramas de tiempo de las entradas y salidas
 - b. Simule usando Verilog para algunos casos representativos (cambios de estado, activación de sensores, disparo sirena, etc.)
2. Implemente el módulo principal en la placa Upduino y pruebe el funcionamiento con las entradas y salidas de la placa Digilent
 - a. ¿Hasta que frecuencia de clock interno puede funcionar?
3. Diseñe e implemente el circuito de hardware para el teclado
 - Use las líneas de datos **serie** especificadas (2x STATUS, 2x KB, CLK, RESET) además de alimentación
 - Si necesita alguna línea adicional, justifique
 - Se recomienda usar una placa/breadboard separado para más claridad

- a. Detalle el esquemático de su circuito. Incluya materiales de diseño (máquinas de estado, ecuaciones lógicas, etc.)
 - i. Determine la frecuencia de clock de bus (SERCLK) que va a usar
 - b. Pruebe el teclado usando la placa Digilent
 - c. Calcule/Determine la frecuencia máxima de clock de bus que soporta su teclado (ignorando los cables de interconexión con el módulo principal)
 - d. Si es posible. calcule el tiempo de metaestabilidad del circuito de su teclado, asumiendo que va a recibir entradas externas (botón) a un máximo de 10Hz y la frecuencia de SERCLK que determino anteriormente
4. Combine el teclado junto con el módulo principal. Utilice pulsadores y LEDs, o la placa Digilent, para las entradas de sensores y salida de alarma
 - a. Muestre el funcionamiento en un video
 - b. Capture el comportamiento con Digilent
 - c. El informe debe incluir un resumen de los módulos y su interconexión, partiendo del que se presenta aquí, con los detalles de su implementación.

Recomendación: Implemente y pruebe los módulos (y submódulos del central) de manera independiente en Verilog para comprobar el funcionamiento correcto. Luego lleve los diseños a la FPGA. Un diseño modular ayuda a verificar el funcionamiento y resolver problemas con el diseño de manera más sencilla.

3 Bonus opcional

Implemente y muestre el funcionamiento del control externo. Use la placa Digilent para enviar los comandos. Muestre el efecto de programar un nuevo código.

Opcional: también puede modificar los canales series para usar alguna codificación de tal manera de no necesitar el clock de bus externo (e.j. 1-wire).