Práctica final

Alfonso José Mateos Hoyos (DNI: 44059172G)
Manuel Salgado de la Iglesia (DNI: 70925985B)
Francisco Pinto Santos (DNI: 70918455)

Puesto de trabajo número 9

I. Innovaciones

El ejercicio final nos pedía realizar una alarma con un sensor de distancia, de forma que cuando el sensor detectase algo a cierta distancia, avisase de esto a un usuario mandando un SMS.
No obstante, nosotros hemos añadido una serie de características adicionales:

- Uso de varios elementos para mostrar si se ha activado la alarma: hemos incorporado para ello un buzzer y un LED que se activaran cuando el sistema se establezca en periodo de alerta.
- Uso de un lector RFID: Con este podremos desactivar la alarma y hacerla pasar de activa a inactiva y viceversa.
- Uso de un LCD: con él mostramos en todo momento el estado de la alarma de forma que el usuario pueda leerlo.

II. Circuito

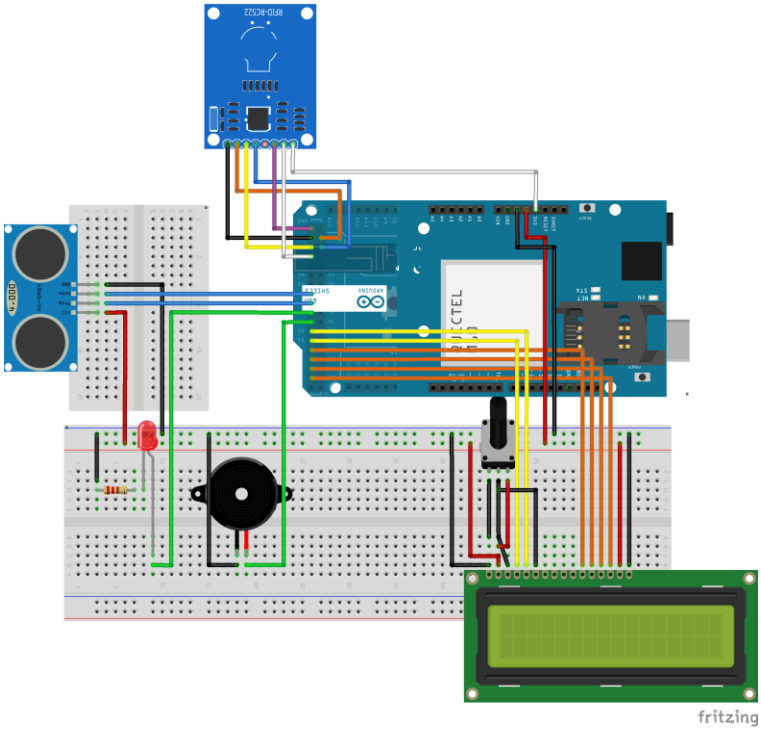

Figura 2.0.0:Diagrama de circuito completo

a) Sensor de distancia

El sensor que hemos usado para medir distancias es el HC-SR04, el cual ya conocíamos de prácticas anteriores realizadas en el laboratorio. El uso de este

es quizás el más relevante, pues nos ayudará a detectar movimiento para activar nuestra alarma.

Su funcionamiento es el siguiente:

- Envía un pulso de alta frecuencia no audible por el ser humano.
- Este pulso rebota en objetos cercanos y es reflejado hacia el sensor, el cual dispone de un micrófono para esta frecuencia.
- Se mide el tiempo entre pulsos, conociendo la velocidad del sonido, y estimamos la distancia del objeto contra el que ha rebotado.

b) Lector RFID

El lector RFID, es un dispositivo que emite una señal de radio, la cual es recibida por los RFID Tag cercanos, los cuales la procesan y devuelven una información u otra en función de dicha señal.

En concreto, hemos usado el módulo MRFC522. Utiliza 3.3V como voltaje de alimentación y las comunicaciones con este se realizan a través del SPI*. Utiliza un sistema de modulación/demodulación para todo tipo de dispositivos pasivos de 13.56Mhz. Lo usaremos para activar/desactivar la alarma con su correspondiente tarjeta.

Por otro lado, la tarjeta, es un RFID TAG cuenta con 64 bloques de memoria (0-63), donde se hace lectura y/o escritura. Cada uno de estos bloques tiene una capacidad máxima de 16 Bytes. El número serie de la tarjeta consiste en cinco valores hexadecimales, los cuales usamos como código de desbloqueo.

*SPI: El Serial Peripherical Interface, es un estándar de bus ideado en los años 80, el cual incorporan gran cantidad de dispositivos hoy en día. Este lo incorpora Arduino, y consiste en una interfaz de comunicación para múltiples periféricos con una arquitectura maestro-esclavo.
Esto nos permite especificar un pin de selección (SS) y uno de reset (RR) del dispositivo, permitiendo que el resto de la comunicación se realice a partir del resto de pines, los cuales pueden compartir varios dispositivos.

c) GSM

GSM es un estándar internacional para teléfonos móviles, es el acrónimo de **Global System for Mobile Communications**.

En nuestro caso, hemos utilizado un GSM Shield que permite a una placa Arduino conectarse a internet, enviar y recibir SMS y hacer llamadas de voz, usando la biblioteca de GSM.
En nuestro caso concretamente, si la alarma no se desactiva en un minuto, se enviará un SMS al teléfono indicado.

Cabe destacar que al igual que el modulo MRFC522, este se comunica mediante el SPI. Debido a esto, nos surgió un problema de compatibilidad del GSM Shield con la placa, por lo que ha sido necesario no conectar el pin numero 12, y puentear el pin 10 y el 12, porque este Shield ha sido diseñado para el Arduino Uno y, al cambiar el modelo de placa, es necesario hacer estos pequeños ajustes, si no la comunicación no se establece de forma apropiada.

d) LCD Display

LCD es el acrónimo de **LiquidCrystal Display**, utiliza las propiedades de la luz polarizada para mostrarnos la información en una pantalla.
Esto lo consigue haciendo que, en cada pixel, haya dos filtros polarizadores, uno rotado 90 grados respecto al otro, con cristal líquido en medio. Al aplicarle corriente a este material, hace que las propiedades de este cambien, afectando a cómo pasa la luz a través de este, haciendo que el píxel se torne oscuro o claro.
Este mecanismo necesita una fuente de luz externa para funcionar, debido a lo cual, hay un luz a la parte derecha de este.

Cabe destacar que, en este LCD, el circuito respectivo al LED y el LCD va por separado, debido a lo cual hay que establecer muchas conexiones. Además, es necesario usar un potenciómetro para regular el contraste del LCD.

Este componente se encarga de convertir las señales eléctricas de la placa en información visual entendible por los seres humanos. Gracias a esto, podremos ir mostrando el estado actualizado de la alarma, además de otros mensajes relevantes.

e) LED y Buzzer

Por último, estos dos componentes serán otra forma de comunicar el estado de la alarma. Cuando esta esté activa y detecte movimiento, el LED se encenderá y el Buzzer comenzará a sonar hasta que logremos desactivarla.

Ambos son componentes que ya conocemos de las anteriores prácticas. Por un lado, el LED, es un diodo emisor de luz. Por otro lado, el buzzer o zumbador, es un dispositivo que genera un sonido de una frecuencia determinada y fija cuando es conectado a tensión.

III. Código

a) Concepto y estructura

Para modelar este sistema, propusimos 3 estados, los cuales se pueden apreciar en la Figura 3.a.1:

- Desactivado: El sistema no realiza la detección de intrusos, solo escanea el lector RFID para que cuando se introduzca la tarjeta, pase a modo activado.
- Activado: El sistema detecta intrusos y escanea el lector RFID para que cuando se introduzca la tarjeta se pase a modo desactivado.
  En el caso de detectar un intruso se pasa a modo alerta.
- Alerta: Se enciende el LED, se emite sonido por el buzzer y en el LCD se muestra el tiempo restante para introducir la tarjeta.
  Si no se introduce la tarjeta se envía un SMS, se paran el buzzer y el LED y se finaliza.

Cada estado está representado mediante una función, la cual se llama desde la función loop.
El estado se guarda en la variable state, que es una variable tipo AlarmState, el cual es un enum con 3 valores cuyos nombres coinciden con los 3 estados posibles de la alarma.
Como se puede apreciar, en cada iteración de la función loop, aparte de llamar a la función correspondiente al estado, se esperan DELAY segundos.

Respecto a la inicialización, esta se realiza en la función setup, donde inicializamos el LCD, SPI (Serial Pheriferical Interface, utilizado para comunicar el Arduino con el MFRC522 y el GSM), MFRC522, GSM, establecemos los pines del buzzer y LED como salida y, por último, establecemos el estado inicial como inactivo e imprimimos el estado.
Cabe destacar que esta función puede durar bastante tiempo, porque la función de inicialización del GSM es bastante lenta.

En cuanto a las 3 funciones correspondientes a cada estado:

- alarmInactiveBehaviour: en el estado inactivo se mira el lector de tarjetas.
  En el caso de que se haya introducido la tarjeta correcta, se pone el estado a activo y se imprime este, tras lo cual en la siguiente iteración de loop se ejecutara la función alarmActiveBehaviour.
- alarmActiveBehaviour: en esta función se mira el sensor de distancia, y en el caso de que haya variado lo suficiente (esto se explicara más a fondo adelante), se establece e imprime por el LCD el estado de alerta, además de guardar el tiempo actual, lo cual se utilizara para calcular el tiempo restante para enviar el SMS. En este caso también se enciende el LED y se activa el buzzer.
  También se escanea el lector de tarjetas, pues si se acerca la tarjeta correcta, se establece el estado inactivo, para ello se notifica por el LCD que la alarma se ha desactivado, tras lo cual se reinicia el Arduino, lo que hará que se vuelva al estado inactivo.

- alarmAlertBehaviour: en esta función lo primero que se hace es calcular el tiempo restante. En el caso de que sea menor que cero se apagan el LED y buzzer y se envía el SMS. Tras esto se reinicia el Arduino, lo que hará que se pase a estado inactivo.
  Después de esto, se mira la tarjeta para ver si se ha conseguido desactivar la alarma, en cuyo caso, se paran el LED y buzzer, se notifica y, al igual que en el caso anterior, se reinicia el Arduino, lo que nos llevará a modo inactivo.

Para reiniciar el Arduino, se ha utilizado la función RESET_ARDUINO(), la cual es un puntero a, la cual lo que hace es que se ejecute desde un principio la función setup y luego la función loop.
También cabe destacar que el reiniciar el Arduino se hace porque como estamos utilizando un contador de 60 segundos para permitir poner la tarjeta y desactivar la alarma, daba problemas con la medición de tiempo.

Por último, en este punto, queríamos destacar que hemos utilizado un fichero de cabecera llamado constants.h, en el cual hemos definido todos los pines, tiempos, parámetros, … que necesitamos, para facilitar el cambio de cualquiera de estos.

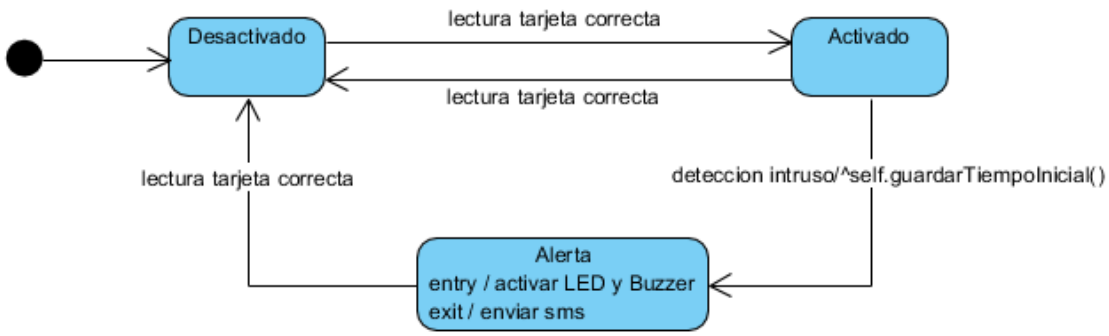

Figura 3.a.1: Diagrama de estados

IV. Referencias

- https://hetpro-store.com/TUTORIALES/modulo-lector-rfid-rc522-rf-con-arduino/
- https://telectronica.com/que-es-un-lector-rfid/
- https://programarfacil.com/tutoriales/fragmentos/arduino/texto-en-movimiento-en-un-lcd-con-arduino/
- https://www.luisllamas.es/arduino-buzzer-activo/
- https://www.arduino.cc/en/Tutorial/LiquidCrystalDisplay
- https://github.com/Martinsos/arduino-lib-hc-sr04
- https://www.luisllamas.es/medir-distancia-con-arduino-y-sensor-de-ultrasonidos-hc-sr04/
- https://www.geekfactory.mx/tutoriales/tutoriales-arduino/pantalla-lcd-16x2-con-arduino/
- https://github.com/PaulStoffregen/Time
- https://playground.arduino.cc/Learning/MFRC522/
- https://randomnerdtutorials.com/security-access-using-mfrc522-rfid-reader-with-arduino/
- https://www.prometec.net/buzzers/
- https://es.wikipedia.org/wiki/RFID
- https://en.wikipedia.org/wiki/Liquid-crystal_display
- https://es.wikipedia.org/wiki/Sistema_global_para_las_comunicaciones_m%C3%B3viles
- https://www.arduino.cc/en/Reference/GSM
- https://forum.arduino.cc/index.php?topic=364386.0
- https://github.com/Seeed-Studio/GPRS_SIM900