



## PracticaSemana8

### Ejercicio 1.F

Realice una implementación donde se cargue registros de tag rfid, los cuales se deben cargar en un archivo csv.

### Grupo 5 Integrantes:

- Silvana Barea
- Mauro Martinez
- Celeste Suarez
- Ernesto Canio
- Juan Diego Gonzalez Antoniazzi
- Ivan Canio
- Matias Flores

### Docentes:

- Gonzalo Vera
- Jorge Morales

# Sistema de asistencia de tiempo Arduino con RFID

En este proyecto, se presenta un sistema de control de asistencia con el lector RFID MFRC522 y Arduino.

- Contiene un lector **RFID** que lee etiquetas **RFID**;
- Nuestra configuración tiene un módulo de reloj en tiempo real para realizar un seguimiento del tiempo;
- Cuando el lector **RFID** lee una etiqueta **RFID**, guarda la hora actual y el **UID** de la etiqueta en una tarjeta **SD**;
- El Arduino se comunica con la tarjeta **SD** usando un módulo de tarjeta **SD**;
- Puede establecer una hora de entrada para comparar si llega a tiempo o tarde;
- Si llegas a tiempo, se enciende un **LED** verde, si llegas tarde, se enciende un **LED** rojo;
- El sistema también tiene un zumbador que emite un pitido cuando se lee una etiqueta.

Componentes a utilizar:



Tablero de circuitos



Arduino uno



2 resistencias de 220 ohmios

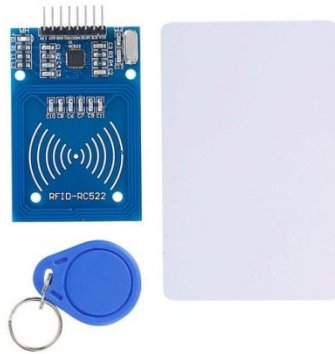


cables



2 Leds (1 rojo y 1 verde)

Un sistema **RFID** necesita etiquetas y un lector:



Las etiquetas se adjuntan al objeto a identificar, en este ejemplo tenemos un llavero y una tarjeta electromagnética. Cada etiqueta tiene su propia identificación única (**UID**).

Reader es un transmisor-receptor de radio bidireccional que envía una señal a la etiqueta y lee su respuesta.

El lector **RFID MFRC522** funciona a 3,3 V y puede utilizar comunicación **SPI** o **I2C**. La biblioteca que vamos a usar para controlar el lector **RFID** solo es compatible con **SPI**, así que ese es el protocolo de comunicación que vamos a usar.

### Módulo de tarjeta **SD**



Cuando se lee una etiqueta, su **UID** y hora se guardan en una tarjeta **SD** para que pueda realizar un seguimiento de los registros. Hay diferentes formas de usar una tarjeta **SD** con el Arduino.

Para comunicarnos con la tarjeta **SD** vamos a utilizar una librería llamada **SD.h**, que ya viene instalada en el **IDE** de Arduino por defecto.

## Módulo RTC (reloj en tiempo real)

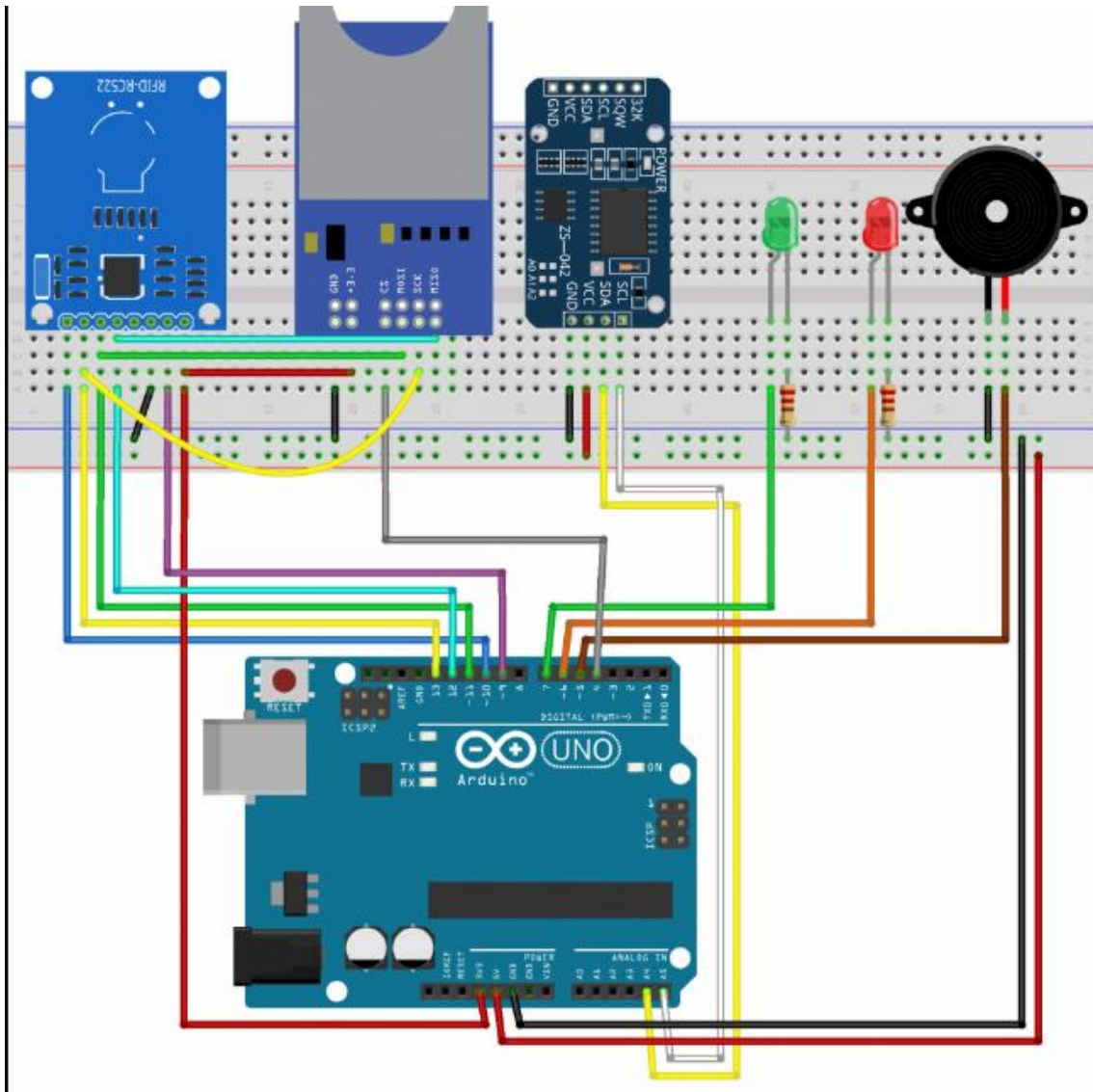


Para realizar un seguimiento del tiempo, estamos utilizando el módulo SD1307 RTC. Sin embargo, este proyecto funciona bien con el DS3231, que es muy similar. Una diferencia principal entre ellos es la precisión. El DS3231 es mucho más preciso que el DS1307.

Este módulo usa comunicación I2C y usaremos la biblioteca RTCLib.h para leer la hora del RTC.

Este proyecto utiliza la biblioteca MFRC522.h para controlar el lector RFID. Esta biblioteca no viene instalada en Arduino IDE de forma predeterminada, por lo que debe instalarla.

El circuito para este proyecto se muestra en los esquemas del circuito a continuación:



Código:

Preparación de lector RFID, tarjeta SD y RTC

```
// definir pines para RFID
#define CS_RFID 10
#define RST_RFID 9
// definir el pin de seleccion para el modulo de la tarjeta SD
#define CS_SD 4
```

Define los pines para el lector RFID y el módulo de la tarjeta SD. Para RFID, el pin SCK ( CS\_RFID ) está conectado al pin 10 y el pin RST ( RST\_RFID ) está conectado al pin 9. Para el módulo de tarjeta SD, el pin Chip Select ( CS\_SD ) está conectado al pin 4.

En este código creas 3 funciones: **readRFID()** , **logCard()** y **checkCheckIn ()**.

La función **readRFID()** lee el UID de la etiqueta, lo guarda en la variable `uidString` y lo muestra en el monitor serie. Además, cuando lee la etiqueta, el zumbador emite un pitido.

La función **logCard()** crea un archivo en su tarjeta SD llamado `DATA.txt` . Puede editar el nombre del archivo, si lo desea, en la siguiente línea.

```
// Abrir archivo
myFile=SD.open("DATA.txt", FILE_WRITE);
```

Luego, guarda el `uidString` (que contiene el UID de la etiqueta) en la tarjeta SD y la hora actual.

```
myFile.print(uidString);
```

```
// Guarde tiempo en la tarjeta SD
DateTime now = rtc.now();
myFile.print(now.year(), DEC);
myFile.print('/');
myFile.print(now.month(), DEC);
myFile.print('/');
myFile.print(now.day(), DEC);
myFile.print(',');
myFile.print(now.hour(), DEC);
myFile.print(':');
myFile.println(now.minute(), DEC);
```

También guarda la hora y los minutos de verificación del usuario en las siguientes variables para una mayor comparación con el tiempo de verificación predefinido.

```
// Guardar hora de entrada
userCheckInHour = now.hour();
userCheckInMinute = now.minute();
```

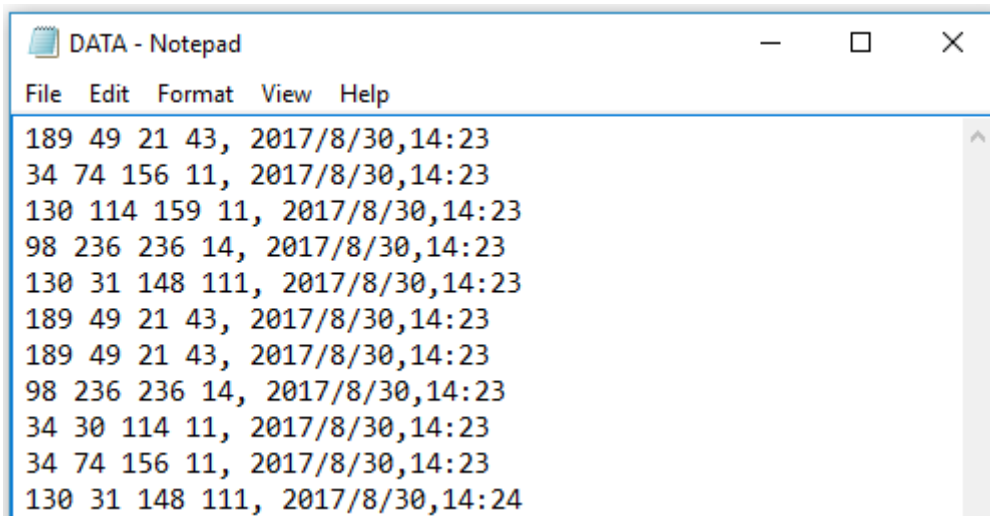
La función **verifyCheckIn()** simplemente compara la hora de entrada del usuario con la hora de entrada predefinida y proporciona retroalimentación en consecuencia. Si el usuario llega tarde, el LED rojo se enciende; si el usuario está a tiempo, el LED verde se enciende.

En resumen el código verifica si se pasó una etiqueta **RFID**. En caso afirmativo, leerá el UID de **RFID**, registrará el UID y la hora en la tarjeta SD y luego le dará retroalimentación al usuario encendiendo uno de los **LED**.

Para verificar los datos guardados en la tarjeta SD, retírela del módulo de la tarjeta SD e insértela en su computadora.

Abra la carpeta de la tarjeta SD y debería tener un archivo llamado `DATA.txt`.

Abra el archivo con un editor de texto. Tendrás algo como lo siguiente:



```
DATA - Notepad
File Edit Format View Help
189 49 21 43, 2017/8/30,14:23
34 74 156 11, 2017/8/30,14:23
130 114 159 11, 2017/8/30,14:23
98 236 236 14, 2017/8/30,14:23
130 31 148 111, 2017/8/30,14:23
189 49 21 43, 2017/8/30,14:23
189 49 21 43, 2017/8/30,14:23
98 236 236 14, 2017/8/30,14:23
34 30 114 11, 2017/8/30,14:23
34 74 156 11, 2017/8/30,14:23
130 31 148 111, 2017/8/30,14:24
```

Cada valor está separado por comas. Esto lo hace más fácil si desea importar estos datos a Excel, Hojas de cálculo de Google u otro software de procesamiento de datos.