
Desarrollo de sistema inmótico basado en plataforma Arduino

Subproyecto: 3. Diseño de sistema de control de fichajes y control de rondas de vigilancia basado en tecnología sin contacto 125 Khz

MODIFICADO EN SUS ESPECIFICACIONES

En distintos sitios del edificio se instalarán tarjetas RFID, las cuales no necesitan alimentación y con un aparato electrónico se leerá el código interno de las tarjetas RFID, el cual será guardado en una tarjeta SD junto a la fecha y hora de la lectura realizada.

Autores:

IES Universidad Laboral de Albacete

Carlos Villora Fernández

Antonio Victoriano Bejarano Rísquez

Antonio Morcillo Ibañez

Índice de contenido

- 1.- Introducción.....2
- 2.- Especificaciones.....2
 - 2.1.- Tarjetas RFID.....2
 - 2.2.- Tarjeta Electrónica.....2
 - 2.3.- Ordenador con lector de tarjetas SD.....3
- 3.- Desarrollo de la tarjeta electrónica.....3
- 4.- Programa.....5

1.- Introducción.

Este proyecto surge de la necesidad del control de vigilancia de una instalación, siendo necesario saber la fecha y hora en la que el vigilante paso por determinada zona.

2.- Especificaciones.

2.1.- Tarjetas RFID.

Lo primero que hay que hacer es buscar en que zonas de la instalación se quiere poner un punto de control para que cuando el vigilante haga la ronda deje constancia de que ha estado allí. Para ello, se colocará en la pared una tarjeta RFID por cada zona a controlar.

Para ello, hay que intentar colocarlas de tal forma que no se puedan quitar, romper, deteriorar, etc. Se pueden meter en pequeñas cajas como las de un interruptor.

2.2.- Tarjeta Electrónica.

La tarjeta electrónica es la encargada de leer el código de la tarjeta RFID que hemos instalado anteriormente en los puntos estratégicos necesarios.

Dicha tarjeta electrónica será la que llevará el vigilante durante sus rondas, por lo que en su fabricación hay que darle una alimentación de baterías, ha ser posible recargable.

Sus partes serán:

- 1.- Alimentación.
- 2.- Tarjeta Arduino.
- 3.- Lector RFID.
- 4.- Reloj.
- 5.- Modulo tarjeta SD.

2.3.- Ordenador con lector de tarjetas SD.

Cuando se quiera ver el estado de las rondas, fecha, hora, tarjetas RFID leídas, etc, lo que hay que hacer es extraer la tarjeta del módulo SD de la tarjeta electrónica e insertar en el lector SD del ordenador.

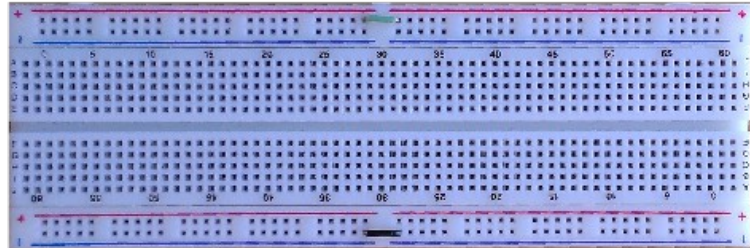
Con un programa de hoja de cálculo se importará el fichero que nos interese leer, teniendo en cuenta que hay que importarlo.

3.- Desarrollo de la tarjeta electrónica.

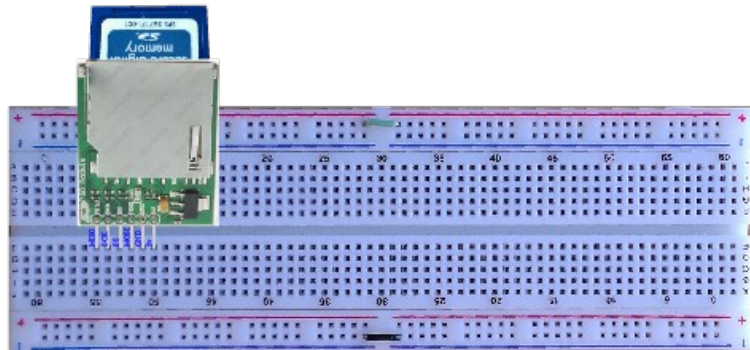
Lector de tarjetas RFID. Almacena en una tarjeta SD la fecha, hora y número de la tarjeta leída.

La tarjeta se leerá con un lector, pudiendo recuperar los datos en una hoja excel.

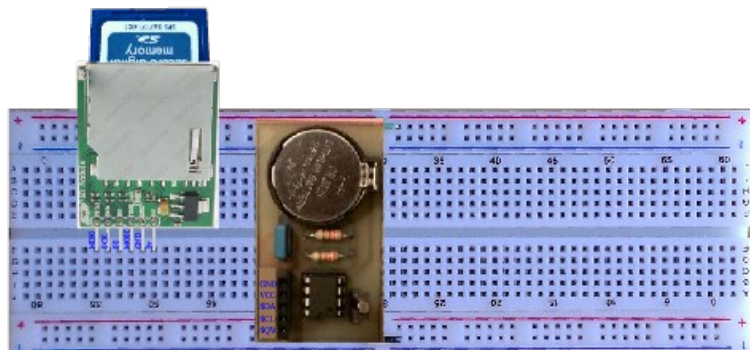
MONTAJE PASO A PASO



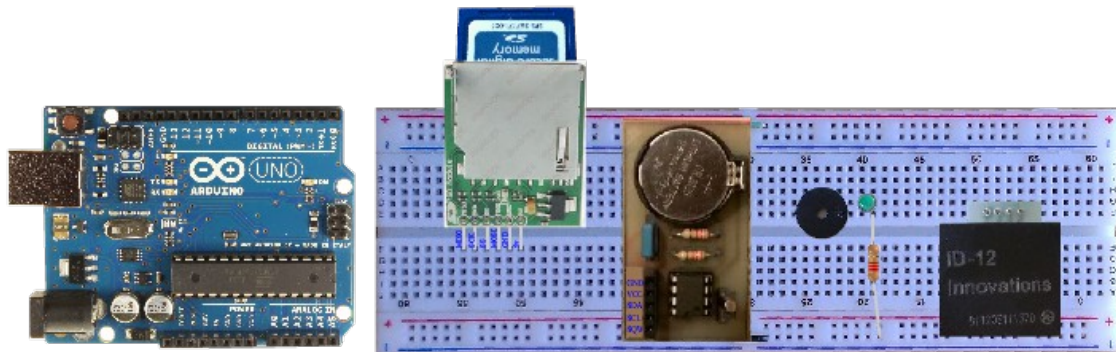
1.- Colocamos el módulo sd sobre la placa protoboard.



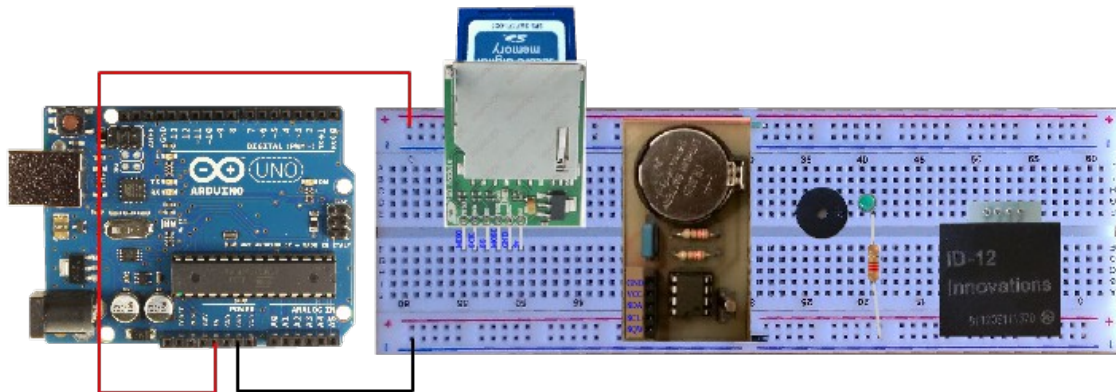
2.- Colocamos el módulo de reloj sobre la placa



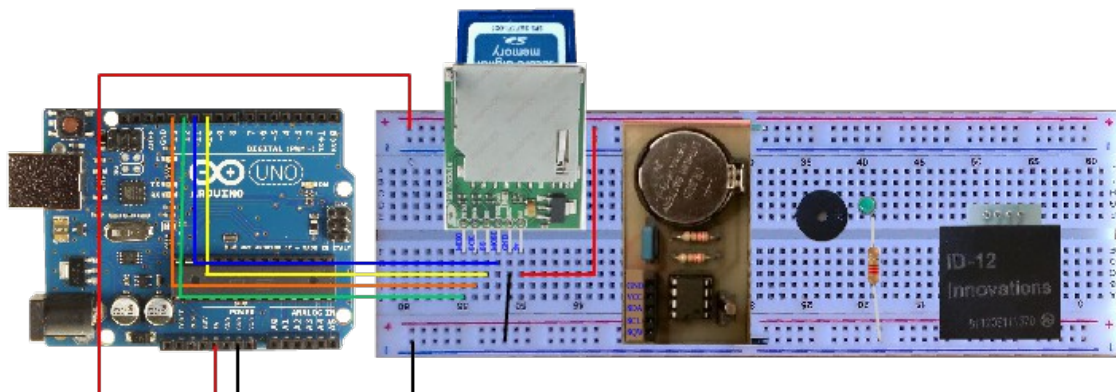
3.- Colocamos el módulo de RFID sobre la placa protoboard.



4.- Colocamos los cables de alimentación del Arduino en la placa protoboard. Arriba +5 voltios y abajo GND.



5.- Colocamos los cables correspondientes al módulo de SD.



Cuando ya este cargado podremos alimentarlo externamente, como por ejemplo, con una pila de 9 voltios.

4.- Programa.

En nuestro programa se crea un archivo cada vez que alimentemos el arduino, por lo que, para un uso eficaz, se recomienda encender el dispositivo, hacer la ronda y, al terminar, apagarlo. De esta forma, se crea un archivo por cada vez que se haga una ronda.

```

// Programa a cargar en el prototipo de control de vigilancia.
// Fecha de realizacion: Septiembre 2012.

#include <SoftwareSerial.h> // RFID
#include <Wire.h> // RTC
#include "RTCLib.h" // RTC
#include <SD.h>

SoftwareSerial RFID(3,5);
RTC_DS1307 RTC;

File archivo;

DateTime fecha;
const int ss = 10; // SD  Arduino UNO
String cadena;
byte Bytes_Recibidos;

void setup ()
{
  pinMode(8, OUTPUT);
  Serial.begin(9600);
  Serial.print("Inicializando tarjeta SD ...");
  pinMode(ss, OUTPUT); // SD
  if (!SD.begin(ss))
  {
    Serial.println("Inicializacion Tarjeta fallida!");
    return;
  }
  Serial.println("Inicializacion Tarjeta CORRECTA.");
  char nombrearchivo[] = "datos00.txt";

  for (byte i=0; i < 100; i++)
  {
    nombrearchivo[5] = i/10 + '0';
    nombrearchivo[6] = i%10 + '0';
    if (!SD.exists(nombrearchivo))
    {
      archivo = SD.open(nombrearchivo, FILE_WRITE);
      break; // va creando archivos nuevos cada vez que reiniciamos
    }
  }

  if (!archivo) {
    Serial.println("No se puede crear el archivo.");
    return;
  }

  //archivo = SD.open(nombrearchivo, FILE_WRITE);
  archivo.println("fecha, hora, tag");

  Wire.begin(); // RTC

```

```

RTC.begin();

if (! RTC.isrunning()) {
  Serial.println("RTC no esta funcionando!");
  RTC.adjust(DateTime(__DATE__, __TIME__));
}
else
{
  RTC.adjust(DateTime(__DATE__, __TIME__));
}
RFID.begin(9600);

Serial.println("*****");
Serial.println("PASA LA TARJETA RFID .....");
Serial.println(nombrearchivo);
Serial.println("fecha, hora, tag");
}

void loop ()
{

  fecha = RTC.now();

  byte tag[20];
  Bytes_Recibidos = LeerRFID(tag);
  if (Bytes_Recibidos > 0)
  {
    archivo.print(fecha.day(), DEC);
    archivo.print("/");
    archivo.print(fecha.month(), DEC);
    archivo.print("/");
    archivo.print(fecha.year(), DEC);
    archivo.print(", ");
    archivo.print(fecha.hour(), DEC);
    archivo.print(":");
    archivo.print(fecha.minute(), DEC);
    archivo.print(":");
    archivo.print(fecha.second(), DEC);

    MostrarFecha();
    //Serial.print("Bytes_Recibidos=");
    //Serial.println(Bytes_Recibidos);
    MostrarTag(tag, Bytes_Recibidos);
    Serial.println();

    archivo.println();
    archivo.flush();

  }

}

```



```
//Ej. Codigo TAG = 2 50 54 48 48 70 70 70 70 57 65 66 67 13 10 3
```

```
byte LeerRFID(byte *tag)
```

```
{
    byte Bytes = 0;
    do {
        while (RFID.available() > 0)
        {
            tag[Bytes++] = RFID.read();
        }
    }
    while (Bytes < 16);
    if ((tag[0] == 2) && (tag[15] == 3))

        delayMicroseconds(1500);

    if ((tag[0] == 2) && (tag[15] == 3))
    {
        return Bytes;
    }
    else {
        return 0;
    }
}
```

```
void MostrarTag(byte *tag, byte Bytes)
```

```
{
    Serial.print(", ");
    archivo.print(", ");
    for (byte k=0; k < Bytes; k++)
    {
        archivo.print(tag[k], DEC);
        archivo.print(" ");
        Serial.print(tag[k]);
        Serial.print(" ");
    }
}
```

```
void MostrarFecha()
```

```
{
    Serial.print(fecha.day(), DEC);
    Serial.print("/");
    Serial.print(fecha.month(), DEC);
    Serial.print("/");
    Serial.print(fecha.year(), DEC);
    Serial.print(", ");
    Serial.print(fecha.hour(), DEC);
    Serial.print(":");
    Serial.print(fecha.minute(), DEC);
    Serial.print(":");
    Serial.print(fecha.second(), DEC);
}
```

Nota.- La función Serial se usa para la depuración del programa. Las líneas correspondientes a ésta función se pueden eliminar o comentar para ahorrar espacio en la flash y hacer más rápido al dispositivo.

5.- Mejoras.

- Construir en una placa PCB todo el prototipo, ocupando el mínimo espacio posible.
- Implementar un sistema de alimentación recargable para nuestro dispositivo.
- Implementar la opción de comunicarse con el ordenador para transmitir los archivos.
- Desarrollar el software para que al recibir los archivos los interprete y guarde en una base de datos, para en cualquier momento poder consultarla.