Interfacing EM-18 RFID reader with Arduino Uno

Arduino Uno is an open source physical computing platform based on ATmega328 microcontroller and provides a development environment for writing software for the board. It can be used for a variety of projects.

EM-18 RFID reader is one of the commonly used RFID reader to read 125KHz tags. It features low cost, low power consumption, small form factor and easy to use. It provides both UART and Wiegand26 output formats. It can be directly interfaced with microcontrollers using UART and with PC using an RS232 converter.

Working of EM-18 RFID module

The module radiates 125KHz through its coils and when a 125KHz passive RFID tag is brought into this field it will get energized from this field.

Reading the code present on RFID tag:

The following code will read the code present on RFID tag and print it in serial monitor.

Connection:

5V-Arduino 5V GND-Arduino GND Tx-pin 9

Code:

```
#include<SoftwareSerial.h>

SoftwareSerial mySerial(9, 10);

int count = 0;                                  // count = 0
char input[12];                                  // character array of size 12
boolean flag = 0;                                // flag =0
void setup()
{
  Serial.begin(9600);                            // begin serial port with baud rate 9600bps
  mySerial.begin(9600);
}
void loop()
{
  if(mySerial.available())
  {
    count = 0;
     while(mySerial.available() && count < 12)          // Read 12 characters and store them in
input array
    {
      input[count] =mySerial.read();
      count++;
      delay(5);
    }
    Serial.print(input);                         // Print RFID tag number

      }
}
```

The output consists of 12 character ASCII data, where first 10 bits will be the tag number and last 2 bits will be the XOR result of the tag number which can be used for error correction. For eg : If the RFID tag number is 500097892E, output of EM-18 Reader will be 500097892E60, where **60 is 50 xor 00 xor 97 xor 89 xor 2E.**

Access control through RFID:

The following code will read the code present on RFID tag tapped. If the code matches with the previously known tag(configured in the code), it will grant access(here LED will glow), otherwise access will be denied.

Connection:

5V-Arduino 5V GND-Arduino GND Tx-pin 9 Led-pin 12

Code:

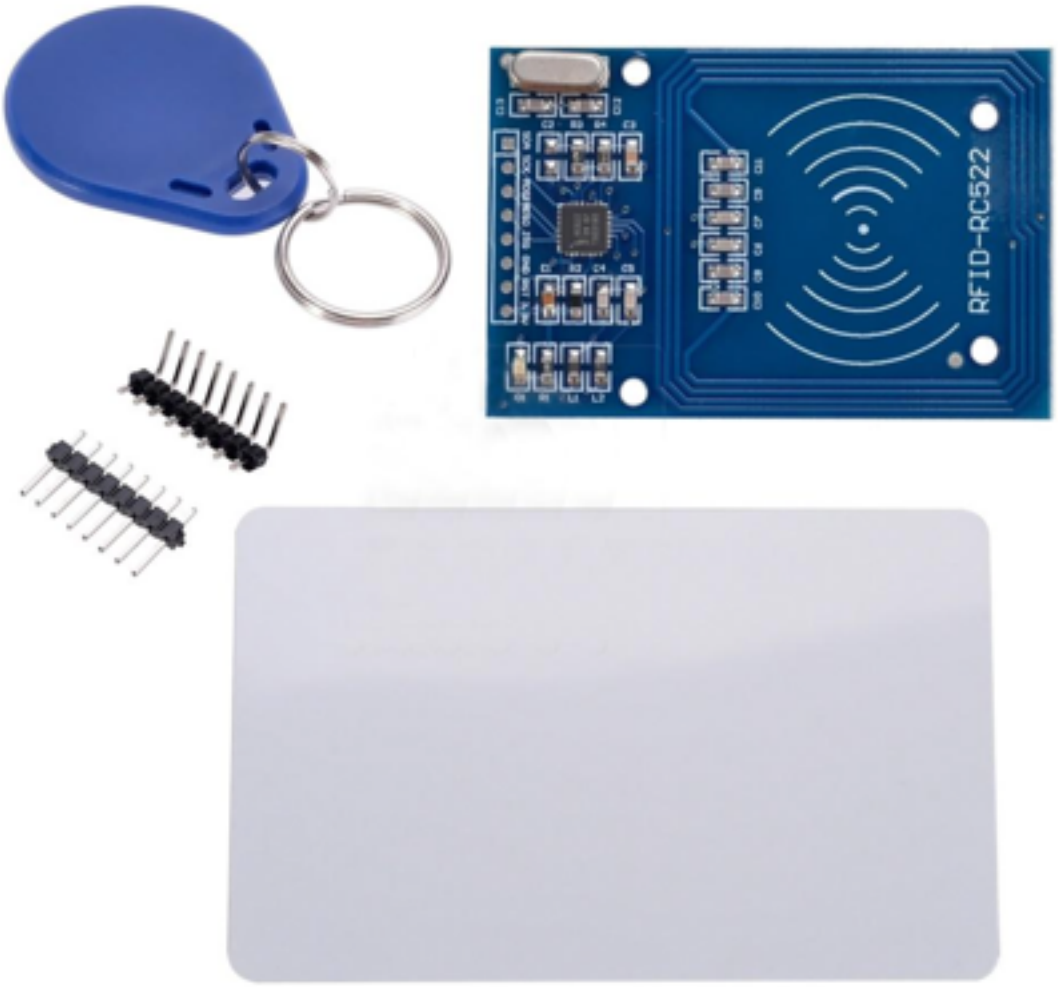
```
#include<SoftwareSerial.h>

SoftwareSerial mySerial(9, 10);

#define LEDPIN 12
char tag[] ="3C0087D597F9"; // Replace with your own Tag ID
char input[12];                                  // A variable to store the Tag ID being presented
int count = 0;                          // A counter variable to navigate through the input[] character array
boolean flag = 0;                       // A variable to store the Tag match status
void setup()
{
        Serial.begin(9600);             // Initialise Serial Communication with the Serial Monitor
           mySerial.begin(9600);
        pinMode(LEDPIN,OUTPUT); //WRONG TAG INDICATOR
}
void loop()
{
        if(mySerial.available())// Check if there is incoming data in the RFID Reader Serial Buffer.
        {
                count = 0; // Reset the counter to zero
                /* Keep reading Byte by Byte from the Buffer till the RFID Reader Buffer is empty
                   or till 12 Bytes (the ID size of our Tag) is read */
                while(mySerial.available() && count < 12)
                {
                        input[count] = mySerial.read();
                                // Read 1 Byte of data and store it in the input[] variable
                                  Serial.write(input[count]);
                        count++; // increment counter
                        delay(5);
```

```
                }
                /* When the counter reaches 12 (the size of the ID) we stop and compare each
value
                   of the input[] to the corresponding stored value */
                if(count == 12) //
                {
                        count =0; // reset counter varibale to 0
                        flag = 1;
                        /* Iterate through each value and compare till either the 12 values are
                          all matching or till the first mistmatch occurs */
                        while(count<12 && flag !=0)
                        {
                                if(input[count]==tag[count])
                                flag = 1; // everytime the values match, we set the flag variable
to 1
                                else
                                flag= 0;
                               /* if the ID values don't match, set flag variable to 0 and
                                 stop comparing by exiting the while loop */
                                count++; // increment i
                        }
                }
                if(flag == 1) // If flag variable is 1, then it means the tags match
                {
                        Serial.println("Access Allowed!");
                        digitalWrite(LEDPIN,HIGH);
                        delay (2000);
                        digitalWrite (LEDPIN,LOW);
                }
                else
                {
                        Serial.println("Access Denied"); // Incorrect Tag Message
                        digitalWrite(LEDPIN,LOW);
                        delay(2000);
                                }
                /* Fill the input variable array with a fixed value 'F' to overwrite
                all values getting it    empty for the next read cycle */
                for(count=0; count<12; count++)
                {
                        input[count]= 'F';
                }
                count = 0; // Reset counter variable
        }
}
```

Mifare MFRC522 RFID Reader/Writer



There are cheap RFID modules that can read and write Mifare's tags and being sold at several web stores, like eBay and included with many "starter kits" nowadays. Simply search RFID-RC522 (MF-RC522). The microcontroller and card reader uses SPI for communication (chip supports I2C and UART protocols but not implemented on library) (Maybe someone implements?). The card reader and the tags communicate using a 13.56MHz electromagnetic field. (ISO 14443A standart tags)