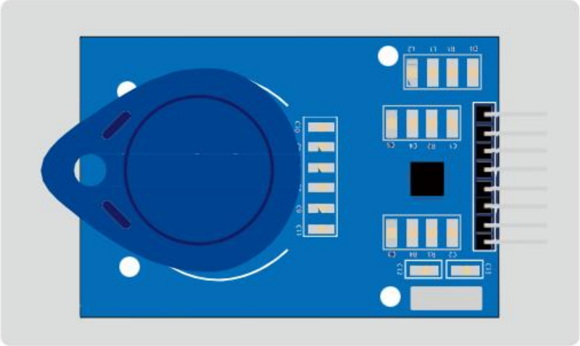
Part 2

lesson 17

RC522 RFID Module

Overview

In this lesson, you will learn how to apply the RC522 RFID Reader Module on ESP32. This module uses the Serial Peripheral Interface (SPI) bus to communicate with controllers such as Arduino, Raspberry Pi, beagle board, etc.



Component Required:

(1) x Elegoo ESP32

(1) x RC522 RFID module

(7) x F-M wires (Female to Male DuPont wires)

Component Introduction

RC522

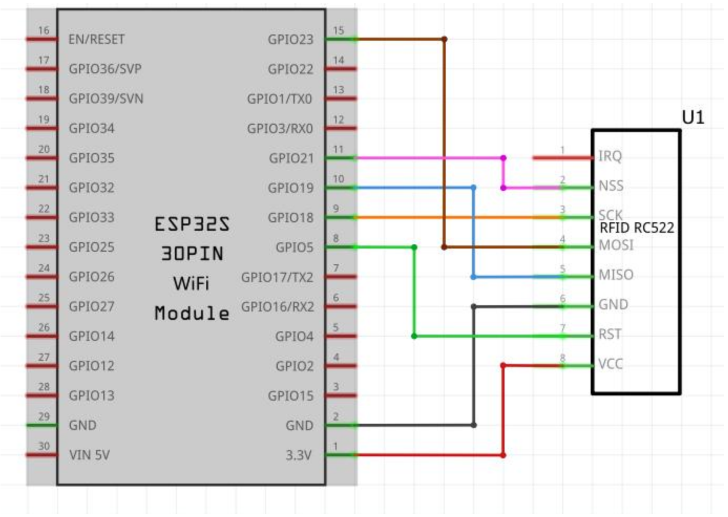
The MFRC522 is a highly integrated reader/writer for contactless communication at 13.56 MHz. The MFRC522 reader supports ISO 14443A / MIFARE® mode.

The MFRC522's internal transmitter part is able to drive a reader/writer antenna designed to communicate with ISO/IEC 14443A/MIFARE® cards and transponders without additional active circuitry. The receiver part provides a robust and efficient implementation of a demodulation and decoding circuitry for signals from ISO/IEC 14443A/MIFARE® compatible cards and transponders. The digital part handles the complete ISO/IEC 14443A framing and error detection (Parity & CRC).The MFRC522 supports MIFARE®Classic (e.g. MIFARE® Standard) products. The MFRC522 supports contactless communication using MIFARE® higher transfer speeds up to 848 kbit/s in both directions.

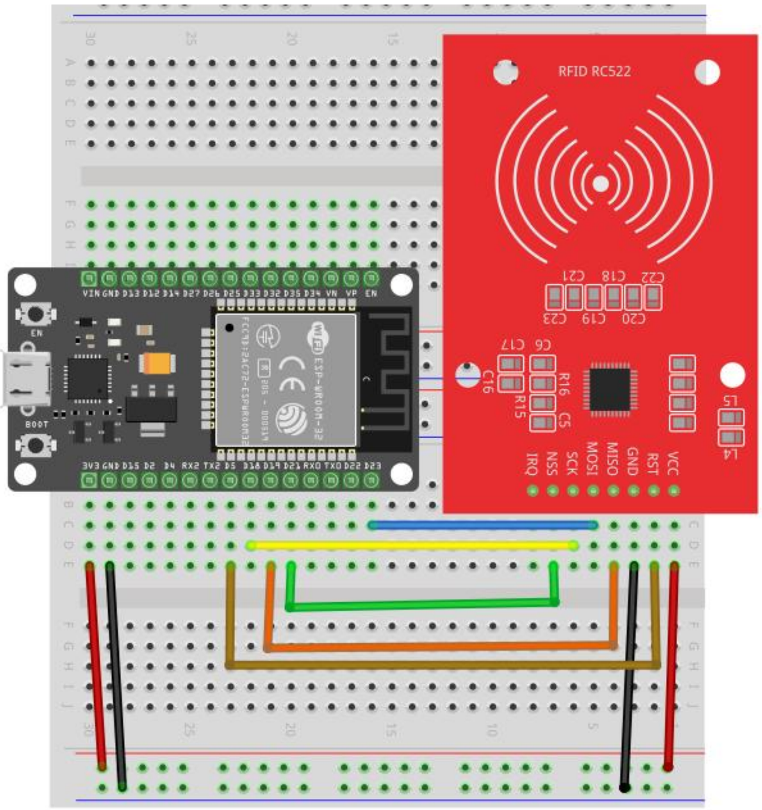
Various host interfaces are implemented:

•SPI interface
•Serial UART (similar to RS232 with voltage levels according pad voltage supply)
•I2C interface.

The figure below shows a typical circuit diagram, using a complementary antenna connection to the MFRC522.



Connection Schematic


RFID RC522
VIN GND D13 D12 D14 D27 D26 D25 D33 D32 D35 D34 VN VP EN
3V3 GND D15 D2 D4 RX2 TX2 D5 D18 D19 D21 RX0 TX0 D22 D23
EN
BOOT
ESP-WROOM-32
IRQ
NSS
SCK
MOSI
MISO
GND
RST
VCC

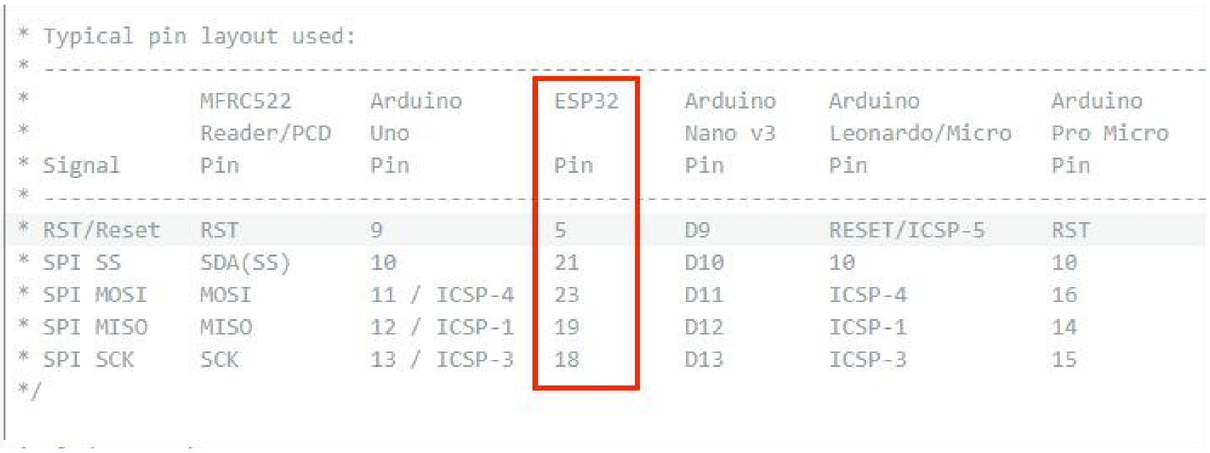
Wiring diagram

Code

After wiring, please open the program in the code folder- **MF-RC522_RFID** and press UPLOAD to upload the program. See Lesson 5 of part 1 for details about program uploading if there are any errors.

Before you can run this, make sure that you have installed the **< rfid >** library or re- install it, if necessary. Otherwise, your code won't work.

For details about loading the library file, see Lesson 5 of part 1 .



```
 * Typical pin layout used:
 * -----------------------------------------------------------------------------------------
 *             MFRC522      Arduino       ESP32     Arduino    Arduino          Arduino
 *             Reader/PCD   Uno                     Nano v3    Leonardo/Micro   Pro Micro
 * Signal      Pin          Pin           Pin       Pin        Pin              Pin
 * -----------------------------------------------------------------------------------------
 * RST/Reset   RST          9             5         D9         RESET/ICSP-5     RST
 * SPI SS      SDA(SS)      10            21        D10        10               10
 * SPI MOSI    MOSI         11 / ICSP-4   23        D11        ICSP-4           16
 * SPI MISO    MISO         12 / ICSP-1   19        D12        ICSP-1           14
 * SPI SCK     SCK          13 / ICSP-3   18        D13        ICSP-3           15
 */
```

The locations of SPI pins vary with different chips, and you have to make a minor modification of the function.

#define RST_PIN	5	// Configurable, see typical pin layout above
#define SS_PIN	21	// Configurable, see typical pin layout above

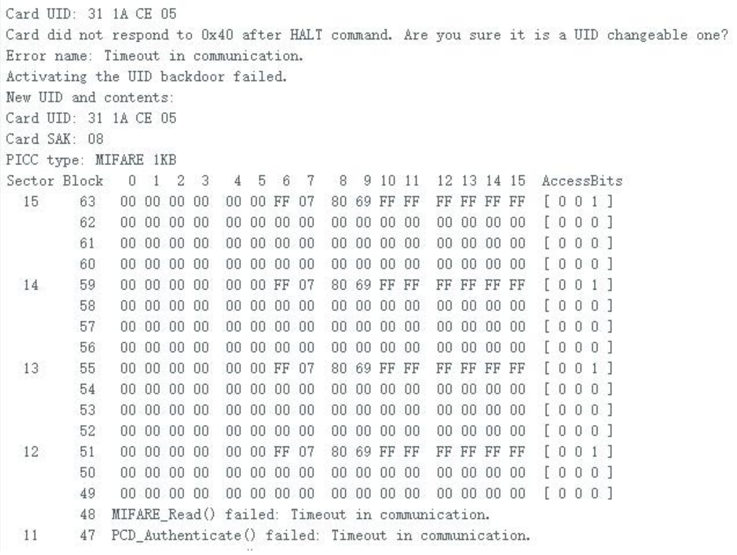
```
if ( ! mfrc522.PICC_IsNewCardPresent() || ! mfrc522.PICC_ReadCardSerial() )
{ return;
}
```

!
[Boolean Operators] Description

Logical NOT results in a true if the operand is false and vice versa.

Open the monitor then you can see the data as blow:

Click the Serial Monitor button to turn on the serial monitor. The basics about the serial monitor are set out in detail in Lesson 4 of part 2.



```
Card UID: 31 1A CE 05
Card did not respond to 0x40 after HALT command. Are you sure it is a UID changeable one?
Error name: Timeout in communication.
Activating the UID backdoor failed.
New UID and contents:
Card UID: 31 1A CE 05
Card SAK: 08
PICC type: MIFARE 1KB
Sector Block   0  1  2  3   4  5  6  7   8  9 10 11  12 13 14 15  AccessBits
  15     63   00 00 00 00  00 00 FF 07  80 69 FF FF  FF FF FF FF  [ 0 0 1 ]
         62   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
         61   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
         60   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
  14     59   00 00 00 00  00 00 FF 07  80 69 FF FF  FF FF FF FF  [ 0 0 1 ]
         58   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
         57   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
         56   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
  13     55   00 00 00 00  00 00 FF 07  80 69 FF FF  FF FF FF FF  [ 0 0 1 ]
         54   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
         53   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
         52   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
  12     51   00 00 00 00  00 00 FF 07  80 69 FF FF  FF FF FF FF  [ 0 0 1 ]
         50   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
         49   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
         48  MIFARE_Read() failed: Timeout in communication.
  11     47  PCD_Authenticate() failed: Timeout in communication.
```