7.Interfacing MFRC522 RFID board with ESP 32.(Practical)

MFRC522 RFID Module

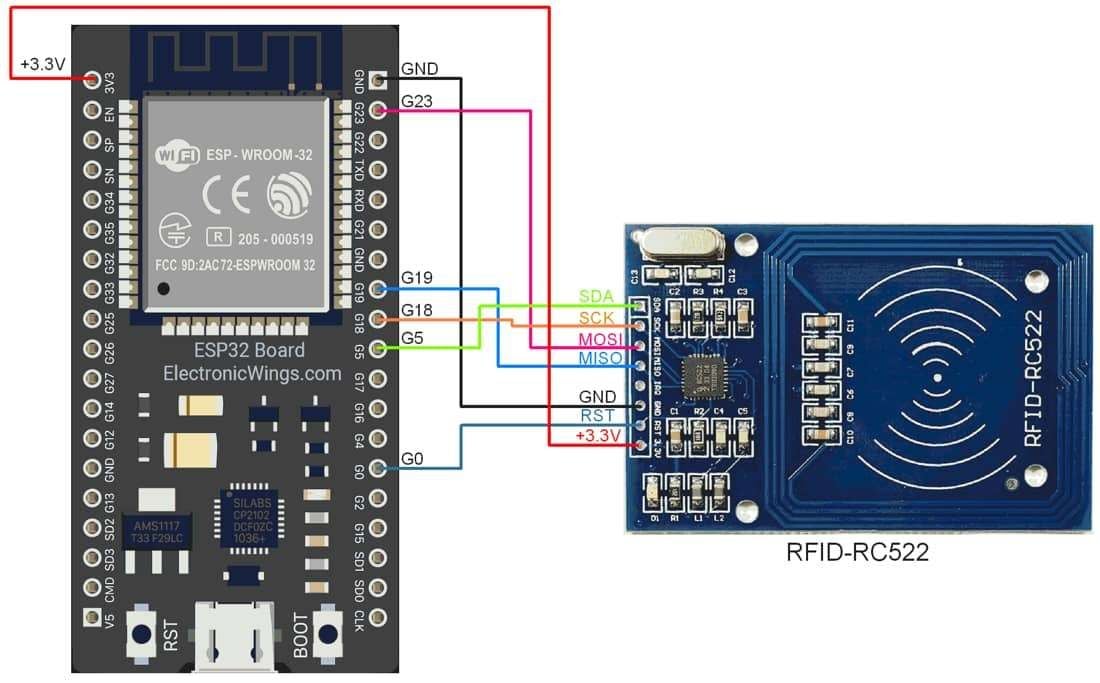
RC522 RFID Modules work on **13.56MHz** which is based on the MFRC522 controller which is designed by NXP Semiconductor.

The RC522 module supports SPI and I2C communication protocols. Here we are using the SPI communication Interface with the ESP32 board.

This module operates on a **+3.3V/13-26mA** Power Supply. The RFID reader automatically goes to power save mode after its operation is complete and it takes a **10-13mA standby current**.

We can use this RFID for inventory management, attendance system, access control system, etc.

**RFID RC522 Hardware Connection with ESP32**



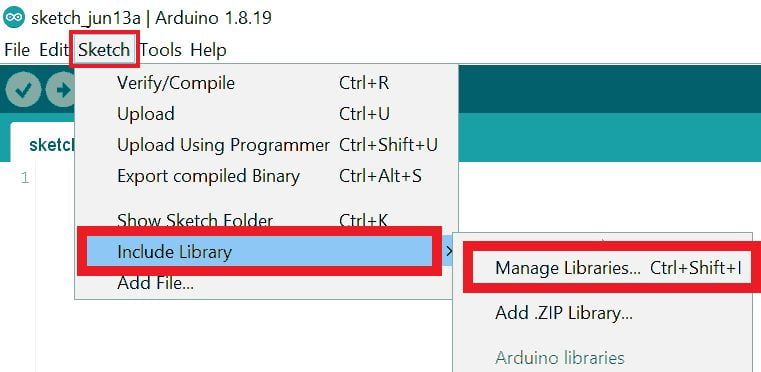
RFID RC522 interfacing with ESP32

|  |  |  |
| --- | --- | --- |
| **Sr No** | **RFID RC522 Pins** | **ESP32 Pins** |
| 1 | VCC | +3.3V |
| 2 | RST | D0 |
| 3 | GND | GND |
| 4 | MISO | 19 |
| 5 | MOSI | 23 |
| 6 | SCK | 18 |
| 7 | SS/SDA | 5 |

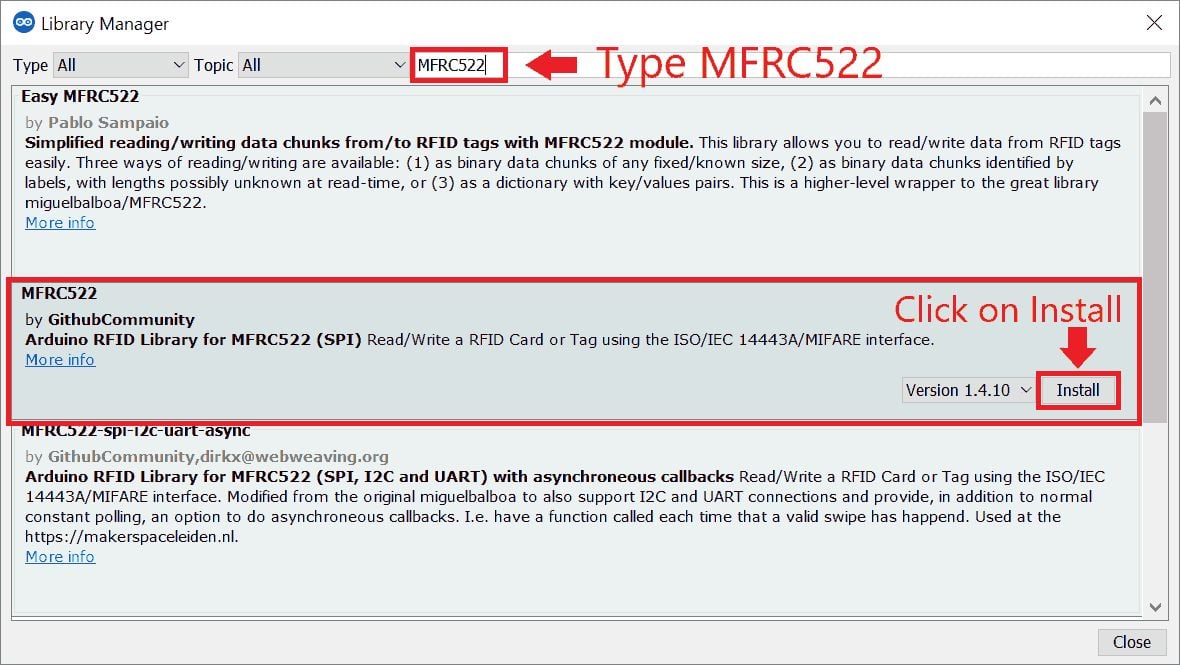
**Read the MFRC522 RFID using ESP32**

Here we are using MFRC522 libraries to read the RFID tag example. We will need to install the **MFRC522** library using the Arduino Library Manager.

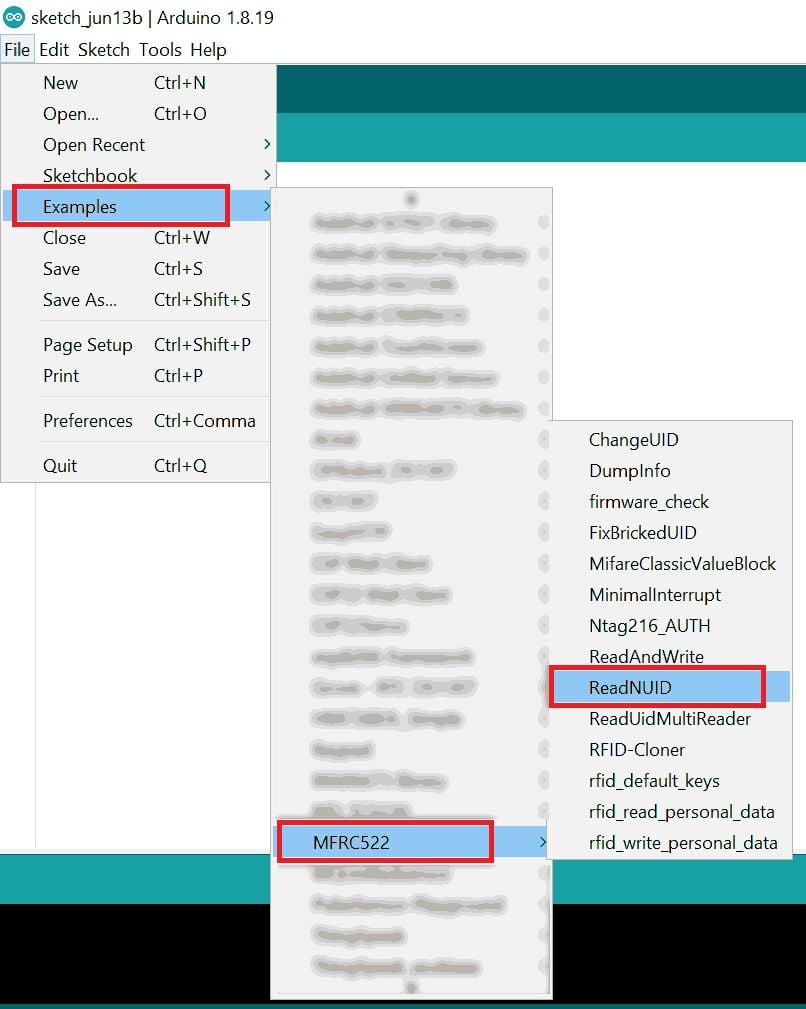
Open the Arduino IDE and navigate to **Sketch ► Include Library ► Manage Libraries…**



The library Manager window will pop up. Now enter **MFRC522** into the search box and click Install on the **MFRC522** option to install version 1.4.10 or higher.



Now open the example of MFRC522. To open it navigate to **File ► Examples ► MFRC522 ► ReadNUID**



**Note:**When you open the code, please change the **SS** and **RST** pin numbers in the code as shown below.

#define SS\_PIN 5

#include <SPI.h>//https://www.arduino.cc/en/reference/SPI

#include <MFRC522.h>//https://github.com/miguelbalboa/rfid

//Constants

#define SS\_PIN 5

#define RST\_PIN 0

//Parameters

const int ipaddress[4] = {103, 97, 67, 25};

//Variables

byte nuidPICC[4] = {0, 0, 0, 0};

MFRC522::MIFARE\_Key key;

MFRC522 rfid = MFRC522(SS\_PIN, RST\_PIN);

void setup() {

  //Init Serial USB

  Serial.begin(115200);

  Serial.println(F("Initialize System"));

  //init rfid D8,D5,D6,D7

  SPI.begin();

  rfid.PCD\_Init();

  Serial.print(F("Reader :"));

  rfid.PCD\_DumpVersionToSerial();

}

void loop() {

  readRFID();

}

void readRFID(void ) { /\* function readRFID \*/

  ////Read RFID card

  for (byte i = 0; i < 6; i++) {

    key.keyByte[i] = 0xFF;

  }

  // Look for new 1 cards

  if ( ! rfid.PICC\_IsNewCardPresent())

    return;

  // Verify if the NUID has been readed

  if (  !rfid.PICC\_ReadCardSerial())

    return;

  // Store NUID into nuidPICC array

  for (byte i = 0; i < 4; i++) {

    nuidPICC[i] = rfid.uid.uidByte[i];

  }

  Serial.print(F("RFID In dec: "));

  printDec(rfid.uid.uidByte, rfid.uid.size);

  Serial.println();

  // Halt PICC

  rfid.PICC\_HaltA();

  // Stop encryption on PCD

  rfid.PCD\_StopCrypto1();

}

/\*\*

   Helper routine to dump a byte array as hex values to Serial.

\*/

void printHex(byte \*buffer, byte bufferSize) {

  for (byte i = 0; i < bufferSize; i++) {

    Serial.print(buffer[i] < 0x10 ? " 0" : " ");

    Serial.print(buffer[i], HEX);

  }

}

/\*\*

   Helper routine to dump a byte array as dec values to Serial.

\*/

void printDec(byte \*buffer, byte bufferSize) {

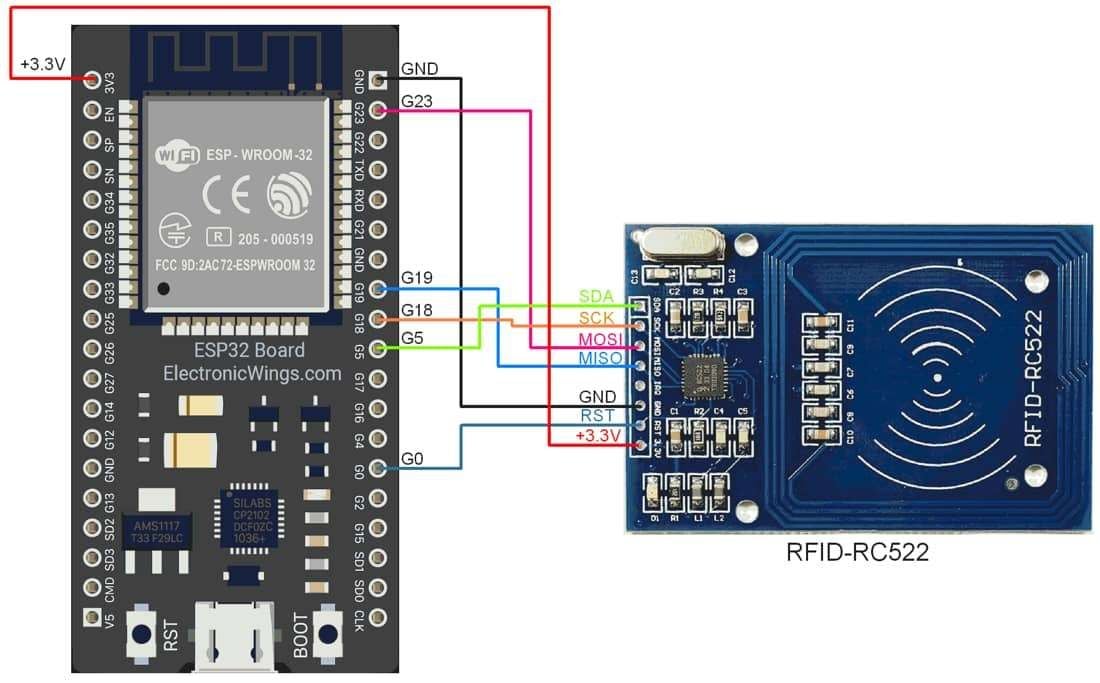
  for (byte i = 0; i < bufferSize; i++) {

    Serial.print(buffer[i] < 0x10 ? " 0" : " ");

    Serial.print(buffer[i], DEC);

  }

}



Exp 7 alternate

#include <SPI.h>

#include <MFRC522.h>

// Define the pins used for the RC522 module

#define RST\_PIN 22  // Reset pin

#define SS\_PIN 5    // Slave Select pin (SDA)

// Create an instance of the MFRC522 class

MFRC522 mfrc522(SS\_PIN, RST\_PIN);  // Create MFRC522 instance

void setup() {

  // Start the Serial Monitor

  Serial.begin(115200);

  // Initialize SPI bus and RC522

  SPI.begin();  // Initialize SPI bus

  mfrc522.PCD\_Init();  // Initialize RC522

  Serial.println("RFID Reader initialized. Scan a card.");

}

void loop() {

  // Look for new RFID card

  if ( !mfrc522.PICC\_IsNewCardPresent()) {

    return;  // If no new card is present, exit the function

  }

  // Select one of the cards

  if ( !mfrc522.PICC\_ReadCardSerial()) {

    return;  // If reading the card's serial number failed, exit the function

  }

  // Print the UID of the scanned card

  Serial.print("UID: ");

  for (byte i = 0; i < mfrc522.uid.size; i++) {

    Serial.print(mfrc522.uid.uidByte[i], HEX);  // Print each byte in HEX format

    Serial.print(" ");

  }

  Serial.println();

  // Halt the card (optional, to stop communication)

  mfrc522.PICC\_HaltA();

  // Stop the communication with the RFID card

  mfrc522.PCD\_StopCrypto1();

  delay(1000);  // Wait 1 second before reading again

}

7.Finding the Default SPI pins of ESP 32(Skilling)

Source Code:

//Find the default SPI pins for your board

//Make sure you have the right board selected in Tools > Boards

void setup() {

  // put your setup code here, to run once:

  Serial.begin(115200);

  Serial.print("MOSI: ");

  Serial.println(MOSI);

  Serial.print("MISO: ");

  Serial.println(MISO);

  Serial.print("SCK: ");

  Serial.println(SCK);

  Serial.print("SS: ");

  Serial.println(SS);

}

void loop() {

  // put your main code here, to run repeatedly:

}

Output:

configsip: 0, SPIWP:0xee

clk\_drv:0x00,q\_drv:0x00,d\_drv:0x00,cs0\_drv:0x00,hd\_drv:0x00,wp\_drv:0x00

mode:DIO, clock div:1

load:0x3fff0030,len:1344

load:0x40078000,len:13964

load:0x40080400,len:3600

entry 0x400805f0

MOSI: 23

MISO: 19

SCK: 18

SS: 5