Lesson 29 RC522 RFID Module

Introduction

In this lesson, you will learn how to apply the RC522 RFID Reader Module on UNO R3. This module uses the Serial Peripheral Interface (SPI) bus to communicate with controllers such as Arduino, Raspberry Pi, beagle board, etc.

Hardware Required

- ✓ 1 *RuiiGuu UNO R3
- ✓ 1 *RC522 RFID module
- ✓ 7 * F-M Jumper Wires

Principle

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 to pad voltage supply)
- ✓ I2C interface.
- ✓ The figure below shows a typical circuit diagram, using a complementary antenna connection to the MFRC522.

Code interpretation

Typical pin layout used:

Signal	MFRC522 Reader/PCD Pin	Arduino Uno Pin	Arduino Mega Pin	Arduino Nano v3 Pin	Arduino Leonardo/Micro Pin	Arduino Pro Micro Pin
RST/Reset	RST	9	5	D9	RESET/ICSP-5	RST
SPI SS	SDA(SS)	10	53	D10	10	10
SPI MOSI	MOSI	11 / ICSP-4	51	D11	ICSP-4	16
SPI MISO	MISO	12 / ICSP-1	50	D12	ICSP-1	14
SPI SCK	SCK	13 / ICSP-3	52	D13	ICSP-3	15

```
#include <SPI.h>

#include <MFRC522.h>


#define RST_PIN     9          // Configurable, see typical pin layout above

#define SS_PIN      10         // Configurable, see typical pin layout above

MFRC522 mfrc522(SS_PIN, RST_PIN);      // Create MFRC522 instance
```

```
/* Set your new UID here! */
#define NEW_UID {0xDE, 0xAD, 0xBE, 0xEF}
MFRC522::MIFARE_Key key;
void setup() {
  Serial.begin(9600);     // Initialize serial communications with the PC
  while (!Serial);        // Do nothing if no serial port is opened (added for Arduinos based on ATMEGA32U4)
  SPI.begin();            // Init SPI bus
  mfrc522.PCD_Init();     // Init MFRC522 card
  Serial.println(F("Warning: this example overwrites the UID of your UID changeable card, use with care!"));
  // Prepare key - all keys are set to FFFFFFFFFFFFh at chip delivery from the factory.
  for (byte i = 0; i < 6; i++) {
    key.keyByte[i] = 0xFF;
  }
}

// Setting the UID can be as simple as this:
//void loop() {
//  byte newUid[] = NEW_UID;
```

```
//   if ( mfrc522.MIFARE_SetUid(newUid, (byte)4, true) ) {
//     Serial.println("Wrote new UID to card.");
//   }
//   delay(2000);
//}
// But of course this is a more proper approach

void loop() {
  // Look for new cards, and select one if present
  if ( ! mfrc522.PICC_IsNewCardPresent() || ! mfrc522.PICC_ReadCardSerial() ) {
    delay(1000);
    return;
  }
  // Now a card is selected. The UID and SAK is in mfrc522.uid.
  // Dump UID
  Serial.print(F("Card UID:"));
  for (byte i = 0; i < mfrc522.uid.size; i++) {
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
    Serial.print(mfrc522.uid.uidByte[i], HEX);
  }
  Serial.println();
```

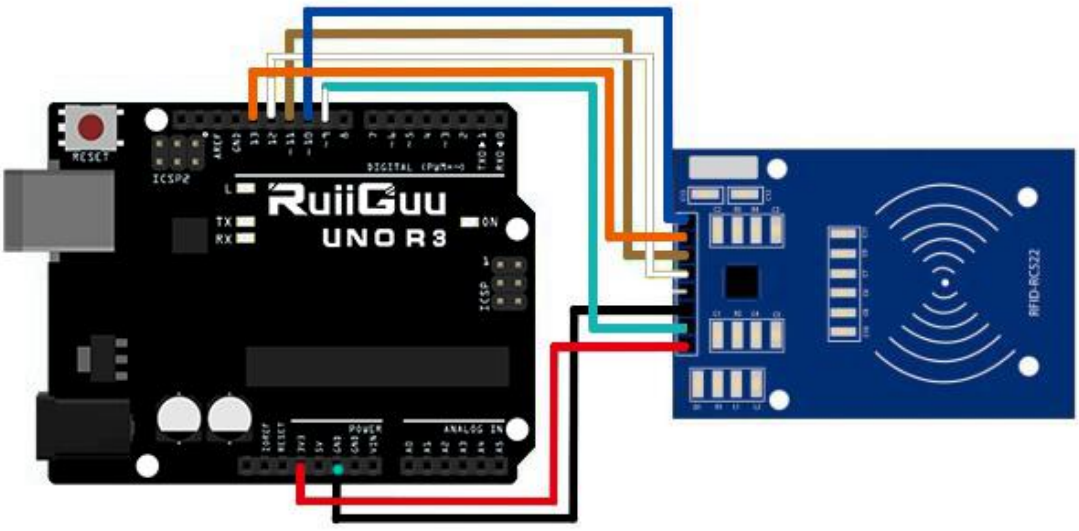
```
  // Dump PICC type
//MFRC522::PICC_Type piccType =
mfrc522.PICC_GetType(mfrc522.uid.sak);
//   Serial.print(F("PICC type: "));
//   Serial.print(mfrc522.PICC_GetTypeName(piccType));
//   Serial.print(F(" (SAK "));
//   Serial.print(mfrc522.uid.sak);
//   Serial.print(")\r\n");
//   if (   piccType != MFRC522::PICC_TYPE_MIFARE_MINI
//     &&   piccType != MFRC522::PICC_TYPE_MIFARE_1K
//     &&   piccType != MFRC522::PICC_TYPE_MIFARE_4K) {
//      Serial.println(F("This sample only works with MIFARE
Classic cards."));
//     return;
//   }

  // Set new UID
  byte newUid[] = NEW_UID;
  if ( mfrc522.MIFARE_SetUid(newUid, (byte)4, true) ) {
    Serial.println(F("Wrote new UID to card."));
  }
  // Halt PICC and re-select it so DumpToSerial doesn't get
```

confused

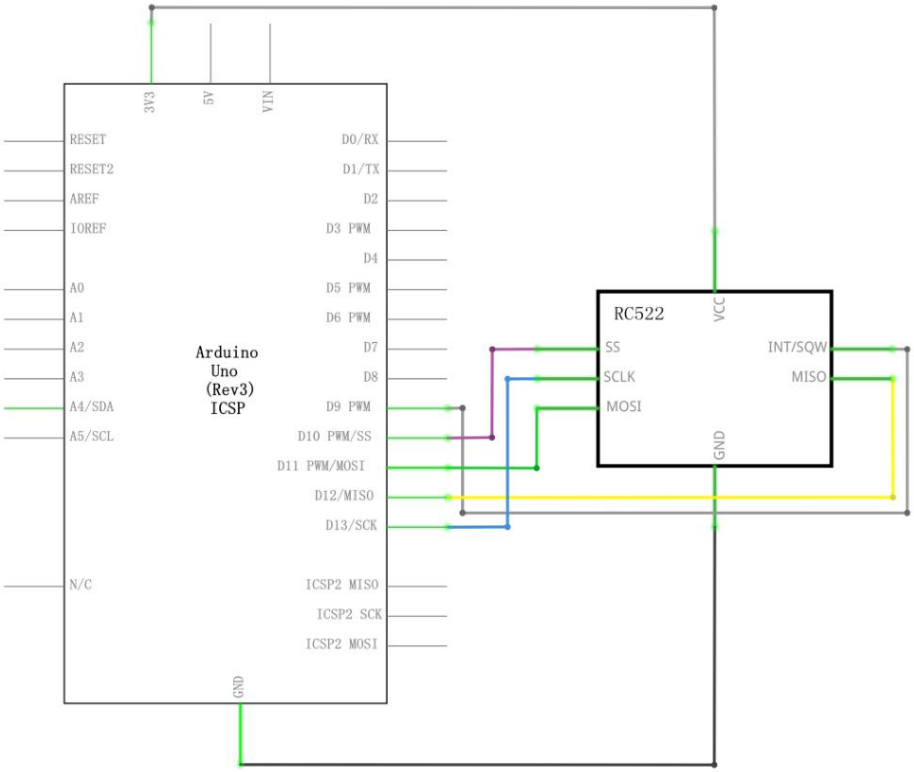
```
  mfrc522.PICC_HaltA();
  if ( ! mfrc522.PICC_IsNewCardPresent() || ! mfrc522.PICC_ReadCardSerial() ) {
    return;
  }
  // Dump the new memory contents
  Serial.println(F("New UID and contents:"));
  mfrc522.PICC_DumpToSerial(&(mfrc522.uid));
  delay(3000);
}
```

Experimental Procedures

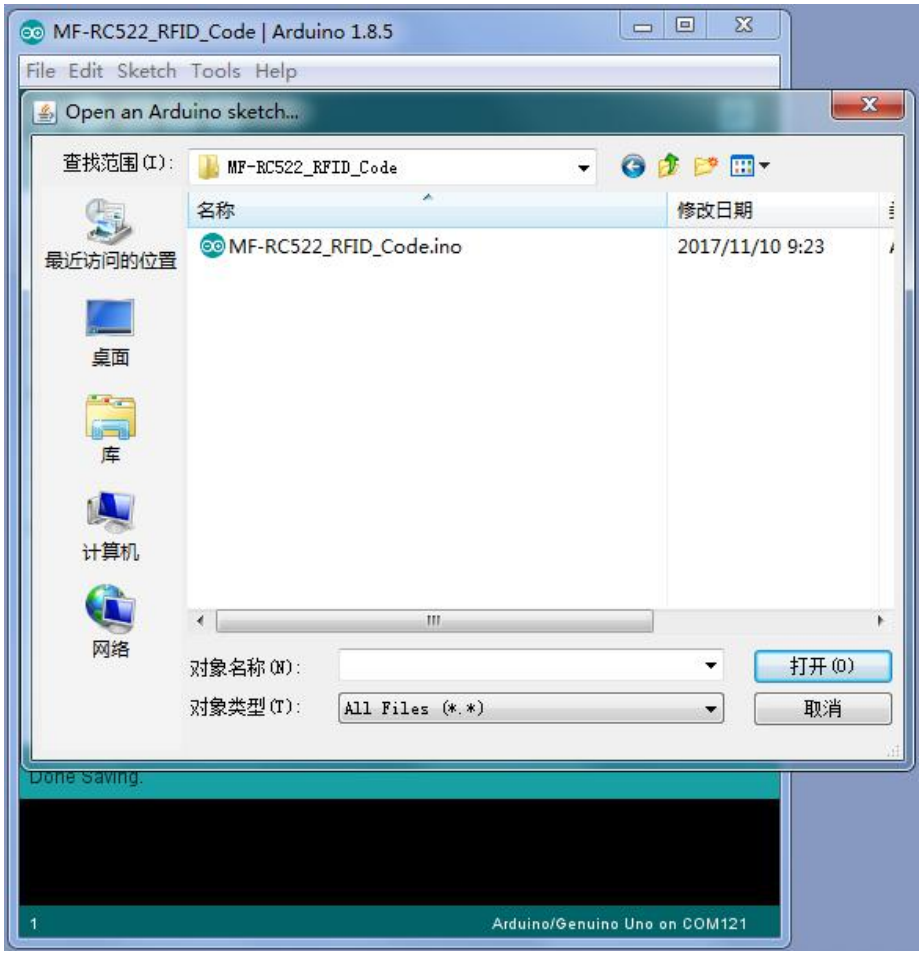
Step 1:Build the circuit



Schematic Diagram

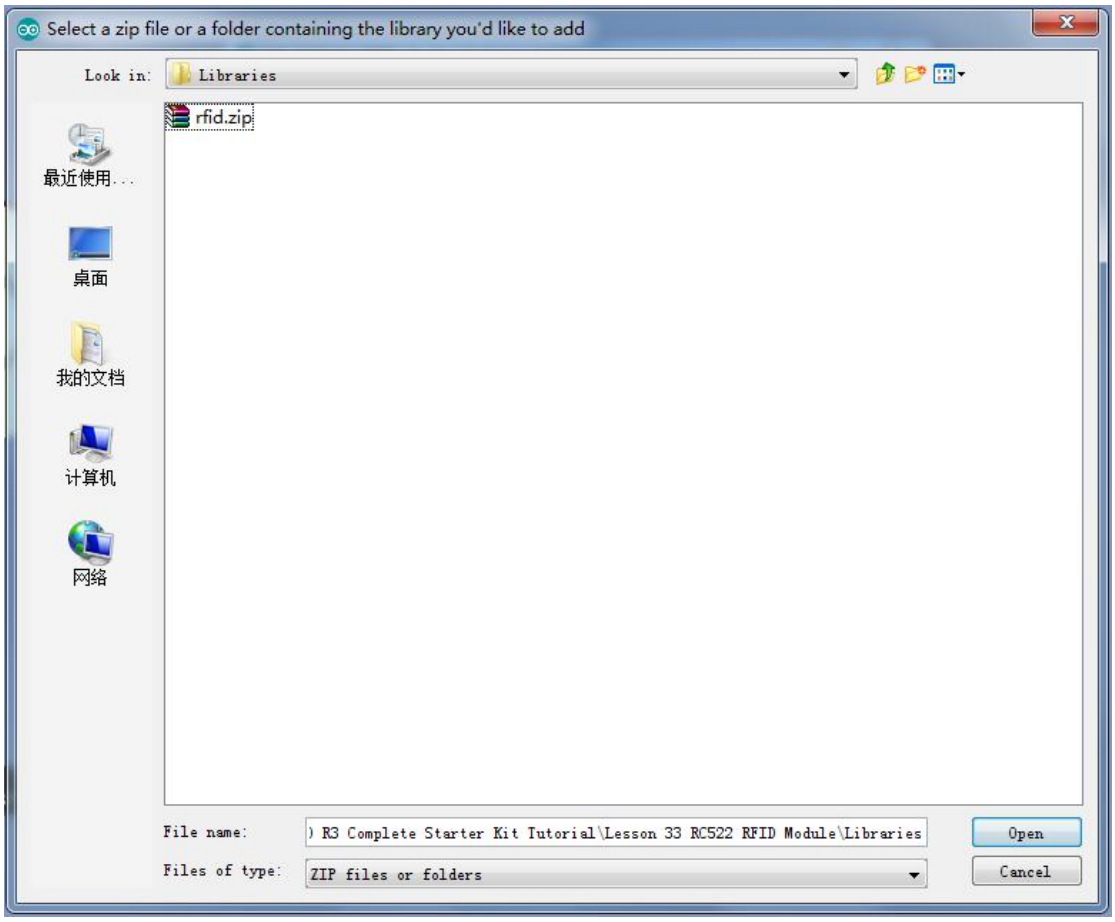


Step 2: Open the code:MF-RC522_RFID_Code



Step 3: Attach Arduino UNO R3 board to your computer via USB cable and check that the 'Board Type' and 'Serial Port' are set correctly.

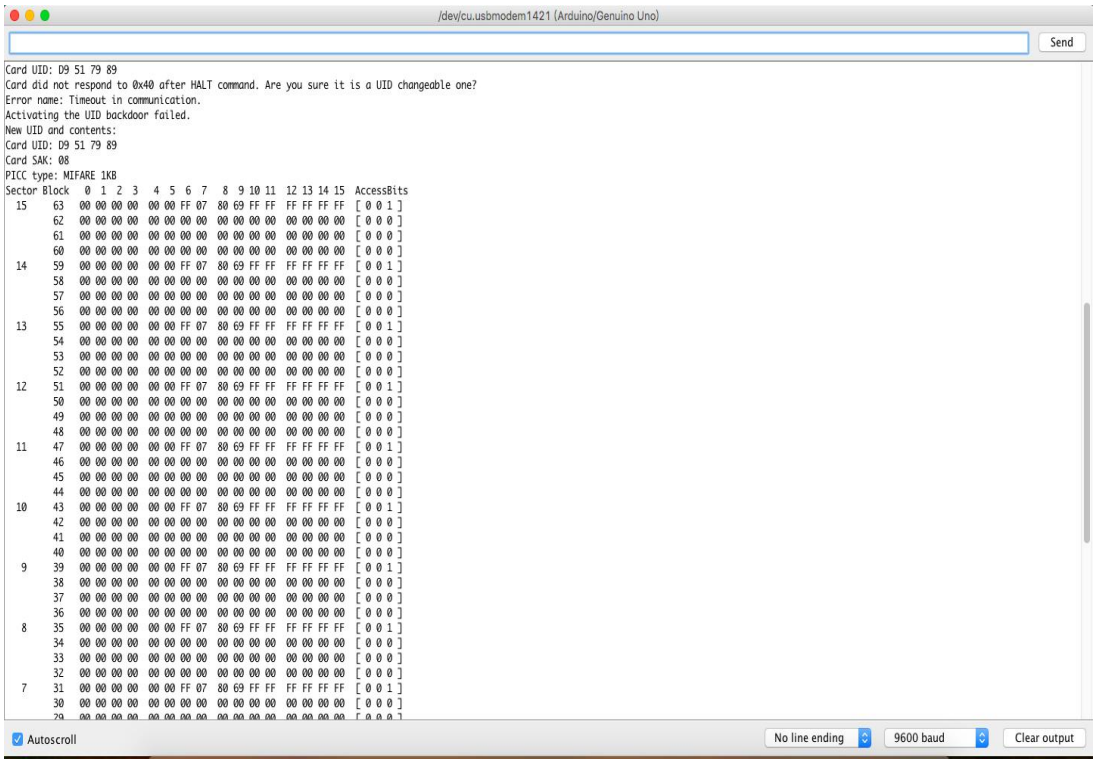
Step 4: Load the Library:rfid

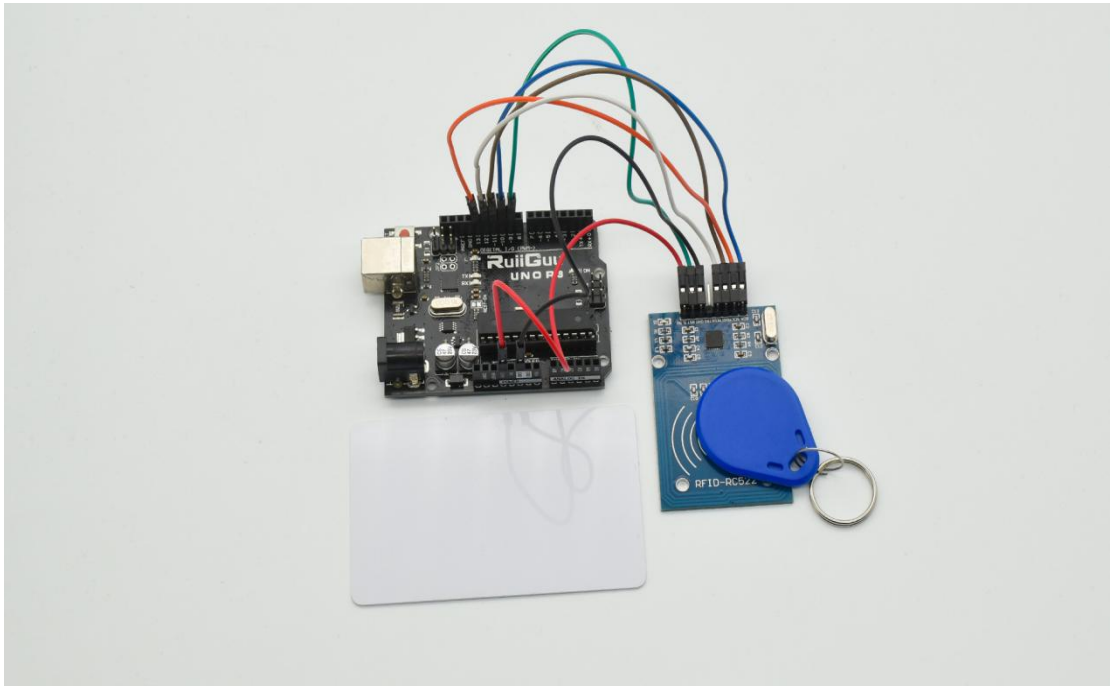


Step 5: Upload the code to the RuiiGuu UNO R3 board.

Step 6: Open the Serial Monitor, then you can see the data as below:

(How to use the Serial Monitor is introduced in details in Lesson 0 Preface)





If it isn’t working, make sure you have assembled the circuit correctly, verified and uploaded the code to your board. For how to upload the code and install the library, check Lesson 0 Preface.