

Internet of Things

+ Practical No 9: Study and Implement RFID, NFC using Arduino.

- In this tutorial, we are going to learn how to use RFID/NFC with Arduino. The RFID/NFC system includes two components: reader and tag. There are two popular RFID/NFC readers: RC522 and PN532 RFID/NFC reader.
- This tutorial focuses on RC522 RFID/NFC reader. PN532 RFID/NFC reader will be presented in an upcoming tutorial.
- RC522 RFID/NFC reader (also called RFID-RC522 Module) can:
 - Read the UID of RFID/NFC tag
 - Change the UID of RFID/NFC tag (only if the tag is UID-writable)
 - Read data from RFID/NFC tagIn above capabilities, for Arduino, reading the UID is the most widely-used. This tutorial focuses on reading the UID of RFID/NFC tag. The other will be present in next tutorials

+ Hardware Required:

- 1) Arduino Uno or Genuino Uno
- 2) USB 2.0 cable type A/C
- 3) RFID/NFC RC522 kit (Reader/Tags)
- 4) Jumper Wires

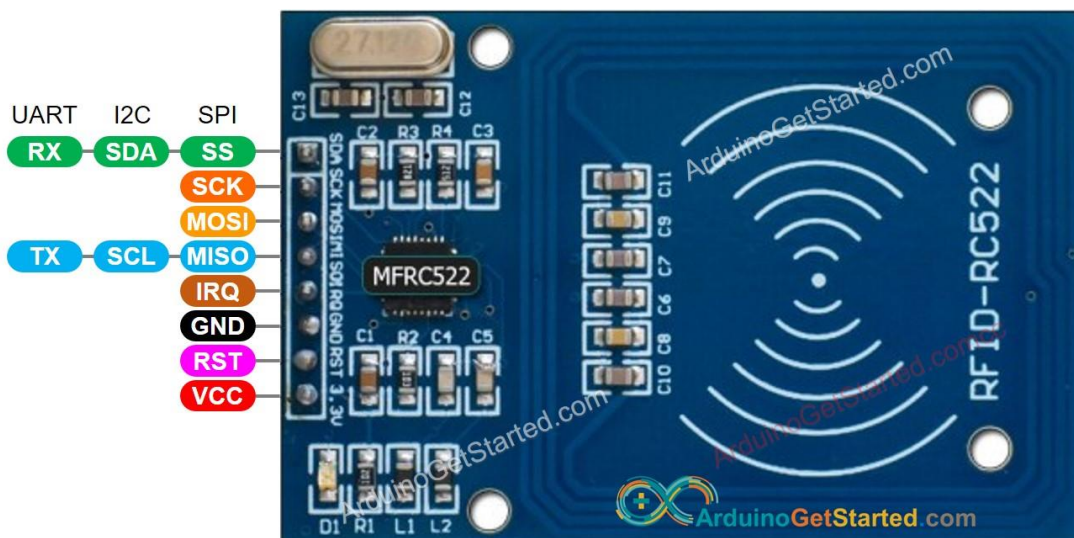
+ About RFID-RC522 Module:

- PinoutRFID-RC522 has 8 pins, some of them are common pins, the others are shared among three communication modes: SPI, I2C, UART. At a time, only one communication mode can be used.

The pin is:

- **GND pin:** needs to connected to Gnd(0v)
- **VCC pin:** needs to connected to vcc(3.3)
- **RST pin:** is a pin for reset and power-down,when this pin goes low,hard power doen is enable,the module is reset.
- **IRQ pin:** is an interrupt pin that can alert the microcontroller when RFID tag comes into its vicinity.

- **MISO/SCL/TX PIN:** acts as MISO when spi interface is enabled, act as SCL when i2c interface is enabled and acts as TX when UART interface is enabled.
- **MOSI pin :** acts as mosi when spi interface is enabled.
- **SCK pin:** acts as Sck when spi interface is enabled.
- **SS/SDA/RX pin:** acts as SS when spi interface is enabled, acts as SDA when I2C interface is enabled, acts as RX when UART interface is enabled.



The reader consists of a radio frequency module and an antenna which generates high frequency electromagnetic field

The tag is usually a passive device, which doesn't need to have power source. The tag contains a microchip that stores and processes information, and an antenna to receive and transmit a signal. The tag is used to store the information: UID (Unique ID) and data.



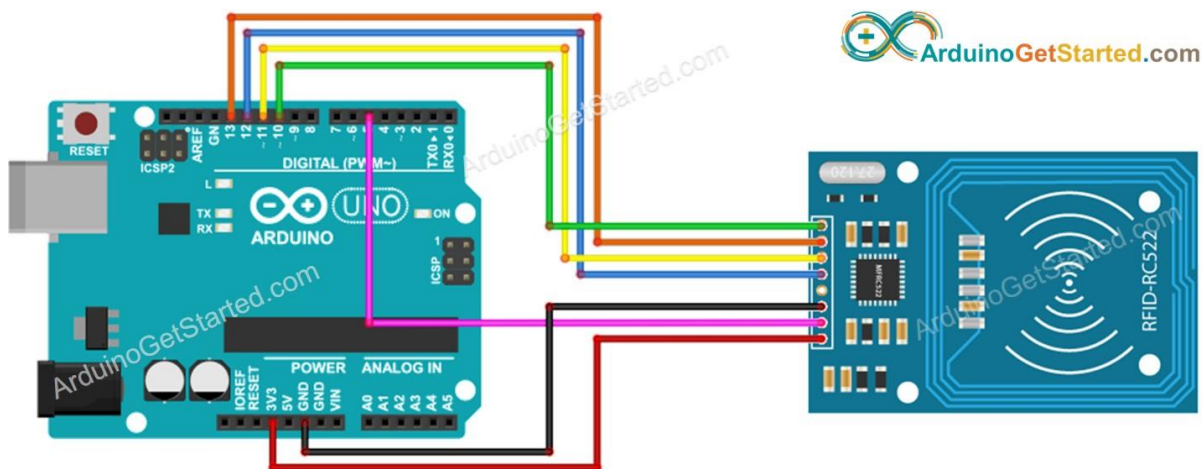
RFID/NFC Reader



RFID/NFC Tag

- To read the information on a tag, the tag must be in close proximity to the reader (does not require the direct line-of-sight). The reading processes:
- The reader generates an electromagnetic field which causes electrons to move through the tag's antenna and subsequently power the chip.
- The chip inside the tag then responds by sending the requested information back to the reader in the form of another radio signal.
- The reader detects the signal and transform the signal into data

Arduino reads the data from reader.



- If you use the male-to-female jumper wires, you can connect Arduino UNO directly to RFID-RC522 module. If you use the male-to-male jumper wires, you need to connect Arduino UNO to RFID-RC522 module via a breadboard.

CODE:

```
#include <SPI.h>
#include <MFRC522.h>

#define SS_PIN 10
#define RST_PIN 5

MFRC522 rfid(SS_PIN, RST_PIN);

void setup() {
  Serial.begin(9600);
  SPI.begin(); // init SPI bus
  rfid.PCD_Init(); // init MFRC522
```

```
Serial.println("Tap RFID/NFC Tag on reader");
}

void loop() {
  if (rfid.PICC_IsNewCardPresent()) { // new tag is available
    if (rfid.PICC_ReadCardSerial()) { // NUID has been readed
      MFRC522::PICC_Type piccType = rfid.PICC_GetType(rfid.uid.sak);
      //Serial.print("RFID/NFC Tag Type: ");
      //Serial.println(rfid.PICC_GetTypeName(piccType));

      // print NUID in Serial Monitor in the hex format
      Serial.print("UID:");
      for (int i = 0; i < rfid.uid.size; i++) {
        Serial.print(rfid.uid.uidByte[i] < 0x10 ? " 0" : " ");
        Serial.print(rfid.uid.uidByte[i], HEX);
      }
      Serial.println();

      rfid.PICC_HaltA(); // halt PICC
      rfid.PCD_StopCrypto1(); // stop encryption on PCD
    }
  }
}
```

OUTPUT:

See UID on Serial Monitor.

