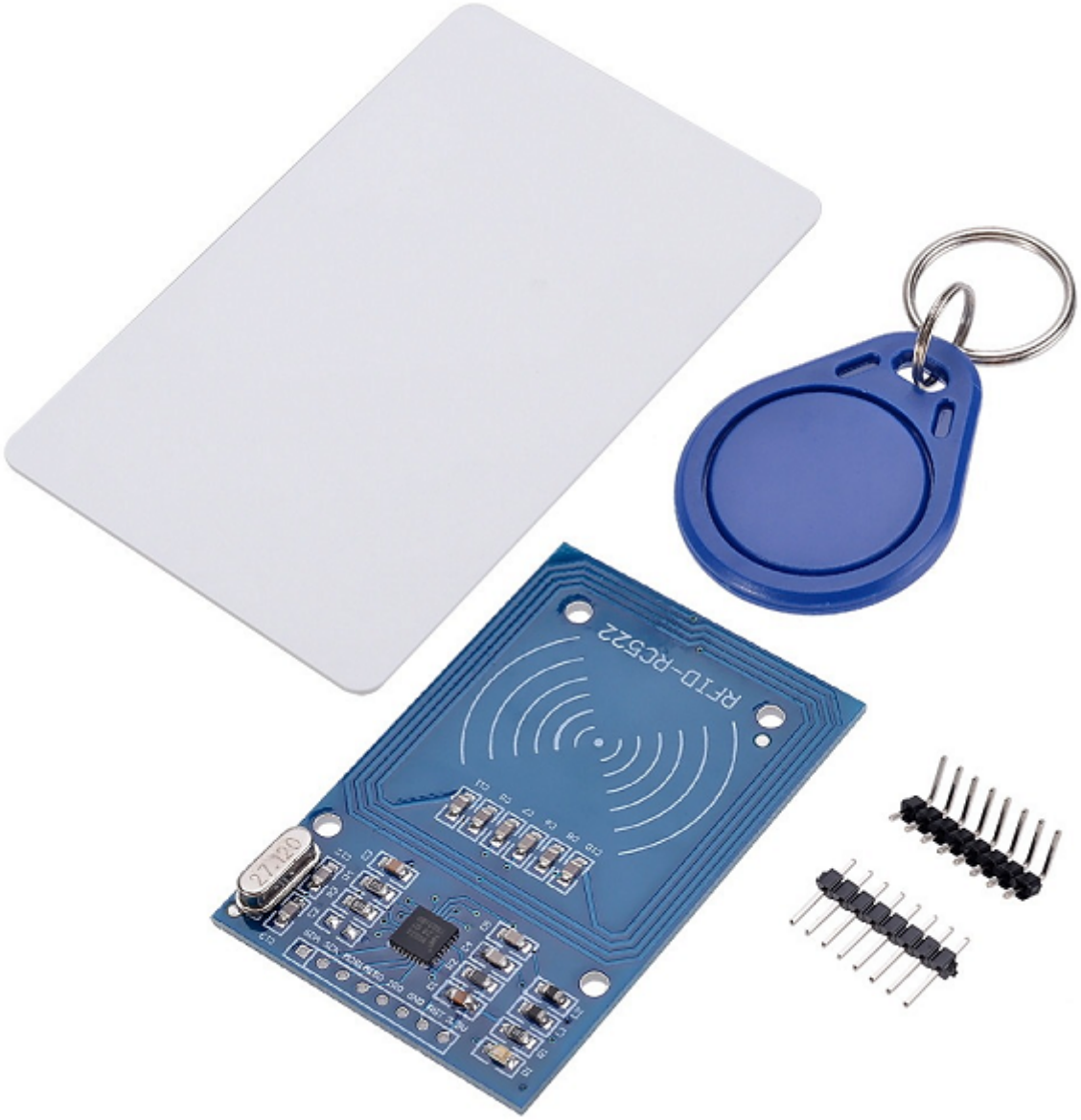
Using an MFRC522 RFID reader

What is an MFRC522 RFID reader

RFID means radio-frequency identification. RFID uses electromagnetic fields to transfer data over short distances. RFID is useful to identify people, to make transactions, etc...

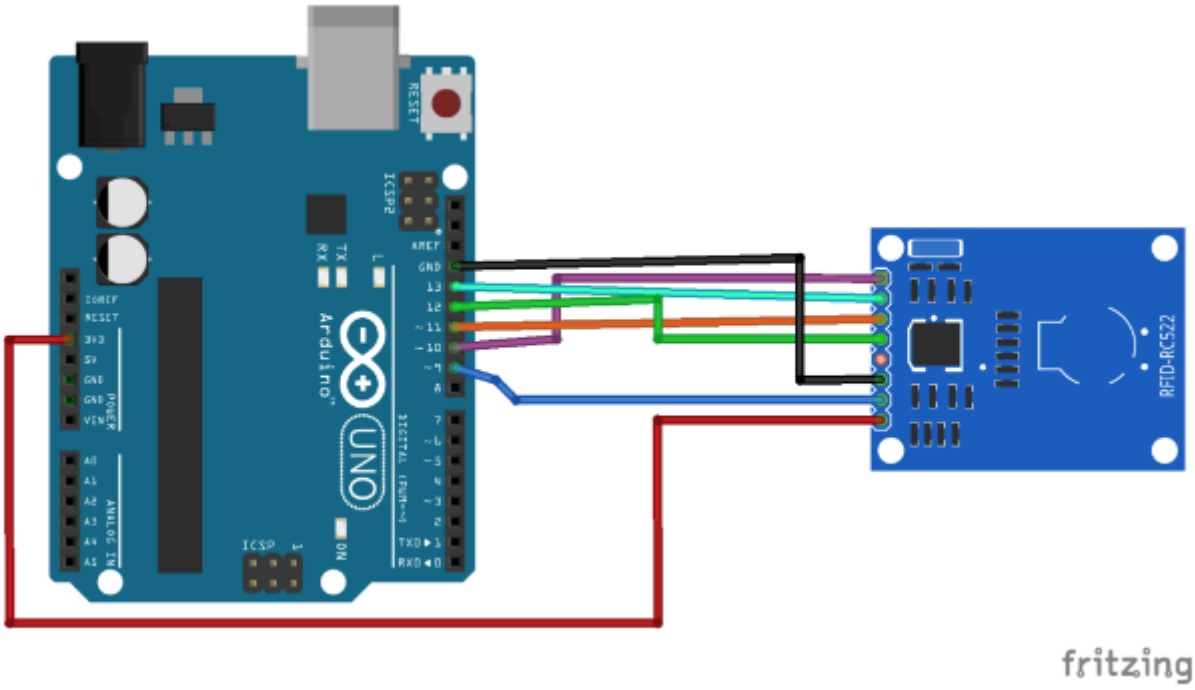
You can use an RFID system to open a door. For example, only the person with the right information on his card is allowed to enter. An RFID system uses tags with each identification and a two-way radio transmitter-receiver as a reader.



Wiring

> **Caution**
> You must power this device to 3.3V!

1. SDA to Digital 10
2. SCK to Digital 13
3. MOSI to Digital 11
4. MISO to Digital 12
5. IRQ (unconnected)
6. Ground (GND to GND)
7. RST to Digital 9
8. Power (3.3V to 3.3V)



Getting started

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

String previous_card_number;
unsigned long readTimeout;

// A struct used for passing the UID of a PICC.
typedef struct {
  byte     size;     // Number of bytes in the UID. 4, 7 or 10.
  byte     uidByte[10];
  byte     sak;      // The SAK (Select acknowledge) byte returned from the PICC after successful selection.
} RFIDUid;
```

```
// A struct used for passing a MIFARE Crypto1 key
typedef struct {
  byte      keyByte[6];
} MIFARE_Key;

#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

void setup() {
  Serial.begin(115200);
  SPI.begin();
  mfrc522.PCD_Init();
}

void loop() {
  rfidRead();
}
// Check for card and get uid
void rfidRead() {
  String card_number;
  MIFARE_Key key;
  MFRC522::Uid *uid = &(mfrc522.uid);

  // Look for card
  if ( ! mfrc522.PICC_IsNewCardPresent() ) return;

  // Select one of the cards
  if ( ! mfrc522.PICC_ReadCardSerial() ) return;

  // UID
  for ( byte i = 0; i < uid->size; i++ ) {
    String byteVal = String( uid->uidByte[i], HEX );
    if (byteVal.length() <= 1) {
      byteVal = "0" + byteVal;
    }
    card_number = card_number + byteVal;
    //    Serial.print( uid->uidByte[i] < 0x10 ? "0" : "" );
    //    Serial.print( uid->uidByte[i], HEX );
```

```
  }

  // Only print card if not the same as last read
  if ( card_number != previous_card_number ) {
    Serial.println(card_number);
  }

  // Remember card for next read
  previous_card_number = card_number;
}
```

Revision #2
Created 25 February 2019 21:27:49 by Tom Lynch
Updated 16 September 2022 13:49:33 by Joanne Leung