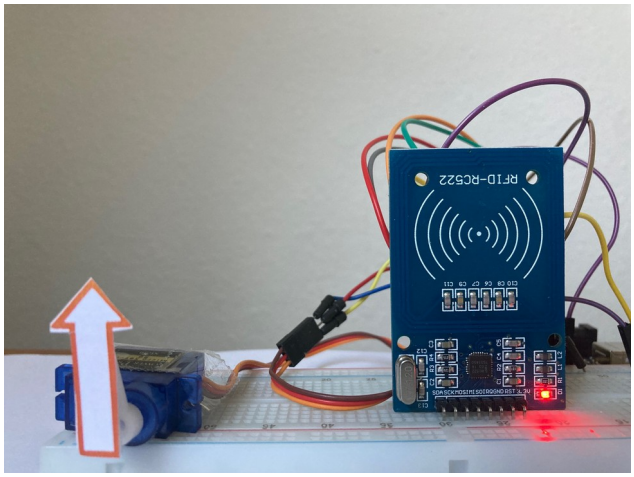
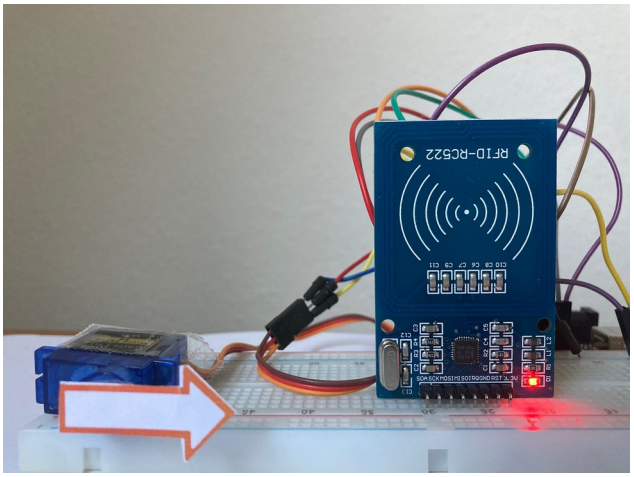
Ein RFID-Modul (radio-frequency identification) besteht aus sogenannten „Tags“, und einem Lesegerät.

Tags

Wird die korrekte Karte an das RFID-Modul gehalten, öffnet der Servomotor die Schranke, anderenfalls bleibt die Schranke geschlossen. Wenn die Karte erneut vor das Lesegerät gehalten wird, schließt die Schranke wieder.



geöffnete Schranke

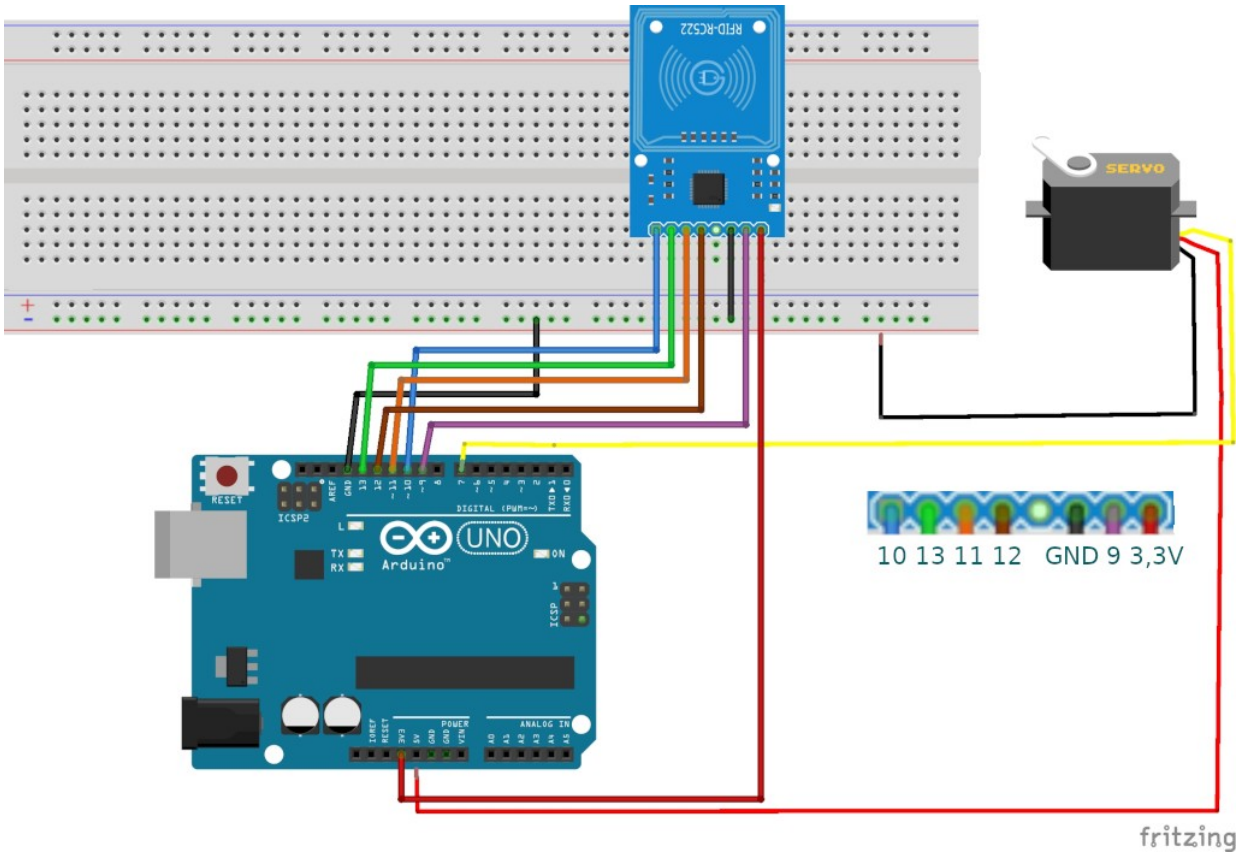


geschlossene Schranke

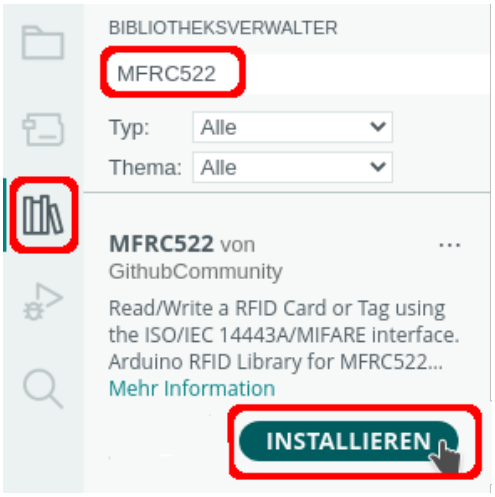
Benötigte Bauteile:

- RFID-Modul
- Servomotor
- kodierte Karten
- Leitungsdrähte

Baue die Schaltung auf.



Benötigte Bibliothek:



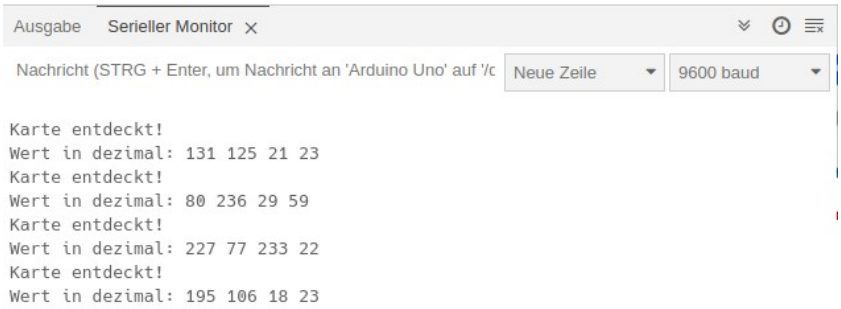
Sketch → Bibliothek einbinden → Bibliotheken verwalten

oder das entsprechende Icon anklicken

Zunächst musst du die individuelle Kennung der Karte herausfinden:

```
// RFID-Bibiothek hinzufügen
# include <MFRC522.h>

// Anschlüsse definieren
# define SDA 10
# define RST 9
```

```
// RFID-Empfänger benennen
MFRC522 mfrc522(SDA, RST);

void setup()
{
  Serial.begin(9600);
  SPI.begin();

  // Initialisierung des RFID-Empfängers
  mfrc522.PCD_Init();
}

void loop()
{
  String WertDEZ;

  // Wenn keine Karte in Reichweite ist ..
  if (!mfrc522.PICC_IsNewCardPresent())
  {
    // .. wird die Abfrage wiederholt.
    return;
  }

  // Wenn kein RFID-Sender ausgewählt wurde ..
  if (!mfrc522.PICC_ReadCardSerial())
  {
    // .. wird die Abfrage wiederholt.
    return;
  }

  Serial.println(F("Karte entdeckt!"));

  // Dezimal-Wert in Strings schreiben
  for (byte i = 0; i < mfrc522.uid.size; i++)
  {
    WertDEZ = WertDEZ + String(mfrc522.uid.uidByte[i], DEC) + " ";
  }

  // Kennung dezimal anzeigen
  Serial.println("Wert in dezimal: " + WertDEZ);

  delay(1000);
}
```



Binde die Bibliotheken ein und definiere die Variablen:

```
// eingebaute Bibliothek einbinden
# include <Servo.h>

// RFID-Bibiothek hinzufügen
# include <MFRC522.h>

// Anschlüsse definieren
# define SDA 10
# define RST 9

// RFID-Empfänger benennen
MFRC522 mfrc522(SDA, RST);

// Bezeichnung des Motors
Servo Motor;

// Schranke nach Betreten wieder schließen
bool Status = false;
```

Der setup-Teil. Beachte die Kommentare:

```
void setup()
{
  // Pin des Motors zuordnen
  Motor.attach(7);
  Serial.begin(9600);

  SPI.begin();

  // Initialisierung des RFID-Empfängers
  mfrc522.PCD_Init();

  // Motor in "geschlossen"-Position fahren
  Motor.write(0);
}
```

Der loop-Teil. Beachte die Kommentare.

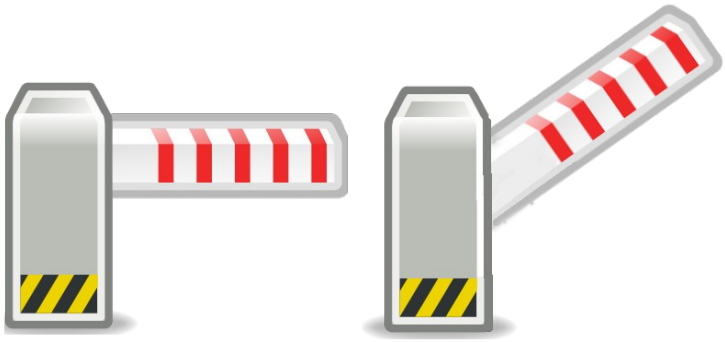
```
void loop()
{
  String WertHEX;
  String WertDEZ;

  // Wenn keine Karte in Reichweite ist ..
  if (!mfrc522.PICC_IsNewCardPresent())
  {
    // .. wird die Abfrage wiederholt.
    return;
  }
  // Wenn kein RFID-Sender ausgewählt wurde ..
  if (!mfrc522.PICC_ReadCardSerial())
  {
    // .. wird die Abfrage wiederholt.
    return;
  }

  Serial.println(F("Karte entdeckt!"));

  // Dezimal-Werte in Strings schreiben
  for (byte i = 0; i < mfrc522.uid.size; i++)
  {
    WertDEZ = WertDEZ + String(mfrc522.uid.uidByte[i], DEC) + " ";
  }

  Serial.println();

  // Kennung dezimal anzeigen
  Serial.println("Wert in dezimal: " + WertDEZ);

  // Leerzeichen am Ende entfernen
  WertDEZ.trim();

  // WertDEZ mit korrektem Wert vergleichen -> Schranke öffnen
  if (WertDEZ.compareTo("80 236 29 59") == 0)
  {
    Serial.println("korrekte Karte -> Schranke \u00f6ffnen");
    Motor.write(90);
    Status = !Status;
  }

  if (WertDEZ.compareTo("80 236 29 59") == 0 && !Status)
  {
    Serial.println("  korrekte Karte/Zutritt erfolgt -> Schranke schlie\u00dfen");
    Motor.write(5);
  }
```

```
  // wenn die Karte falsch ist
  else if (WertDEZ.compareTo("80 236 29 59") != 0)
  {
    Motor.write(5);
    Serial.println("falsche Karte -> Schranke geschlossen lassen");
    Status = false;
  }

  delay(500);
}
```

```
Wert in hexadezimal: 93 DB 28 17
Wert in dezimal: 147 219 40 23
falsche Karte -> Schranke geschlossen lassen
Karte entdeckt!

Wert in hexadezimal: 50 EC 1D 3B
Wert in dezimal: 80 236 29 59
korrekte Karte -> Schranke öffnen
Karte entdeckt!

Wert in hexadezimal: 50 EC 1D 3B
Wert in dezimal: 80 236 29 59
korrekte Karte -> Schranke öffnen
korrekte Karte/Zutritt erfolgt -> Schranke schließen
```

Ausgabe im Seriellen Monitor

Anhang:



Quelle: https://openclipart.org/image/400px/16132 und eigene Bearbeitung

Hartmut Waller (hartmut-waller.info/arduinoblog) Letzte Änderung: 12.11.23