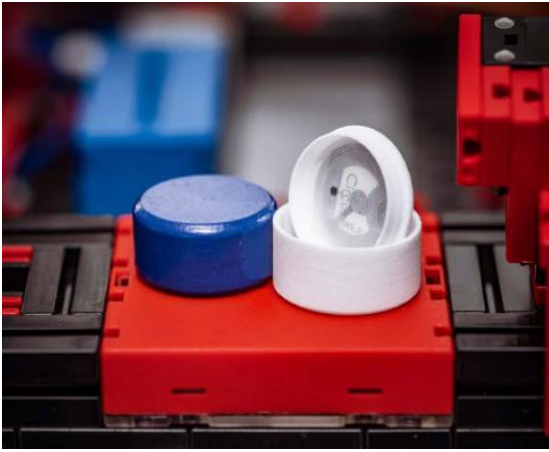


NFC für TXT



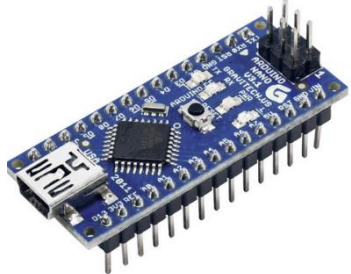



Wenn man im Internet nach NFC für FT sucht, stößt man nur auf den Hinweis, dass es so etwas für die Lernfabrik 4.0 gibt. Dazu stellt FT sogar eigene Werkstücke bereit (174622 bis 174627). Das hat mich dazu inspiriert, ein NFC-System für meine Zwecke zu erstellen. Allerdings musste ich einen Umweg nehmen und einen Arduino-Nano für die entsprechenden NFC-Leseinheit einsetzen. Für die Auswertung am TXT habe ich dann 2 Lösungsansätze gefunden. Zum einen, über 3 Ausgänge, womit es möglich ist bis zu 7 verschiedene NFC-Tags auszuwerten. Die zweite Lösung funktioniert mittels PWM an nur einem Ausgang. Dabei wird jedem NFC-Tag ein unterschiedliches PWM-Signal zugewiesen, welches im TXT dann als entsprechender Spannungswert ausgelesen wird.

Begriffsbestimmung

NFC steht für Near Field Communication (Deutsch: Nahfeldkommunikation), da die Funktechnik, Daten innerhalb einer kurzen Distanz von wenigen Zentimetern überträgt. NFC ist eine RFID-Technologie (radio-frequency identification), geht aber insofern weiter, da hierbei der NFC-Tag auch beschrieben werden kann.

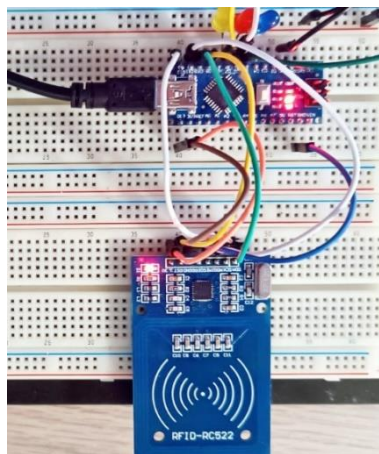
Es gibt die NFC-Tags in vielen unterschiedlichen Formen, wie Aufkleber, Anhänger, Karten oder Armbändern. Er enthält einen Mikrochip, der eine kleine Datenmenge, wie eine URL oder eine digitale Visitenkarte und eine eindeutige Kennung speichern kann. Heutzutage ist es möglich, einen NFC-Tag mit dem Handy auszulesen.

Hardware

<p>Arduino Nano Ca. 7 Euro</p> 	<p>NFC-Leseinheit RFID-RC522 Ca. 7 Euro (wird meist mit 2 Tag's verkauft)</p> 
<p>PWM2 Volt-Wandler ca. 4 Euro Wandelt das PWM-Signal in eine Spannung um</p> 	<p>BERRYBASE 125452 RFID / NFC Tags, selbstklebend, farbig, 6 Stück ca. 4 Euro</p> 

Anschluss Arduino

RFID-RC522	Arduino Nano
SDA	D10
SCK	D13
MOSI	D11
MISO	D12
GND	GND
RST	D9
3,3V	3V3



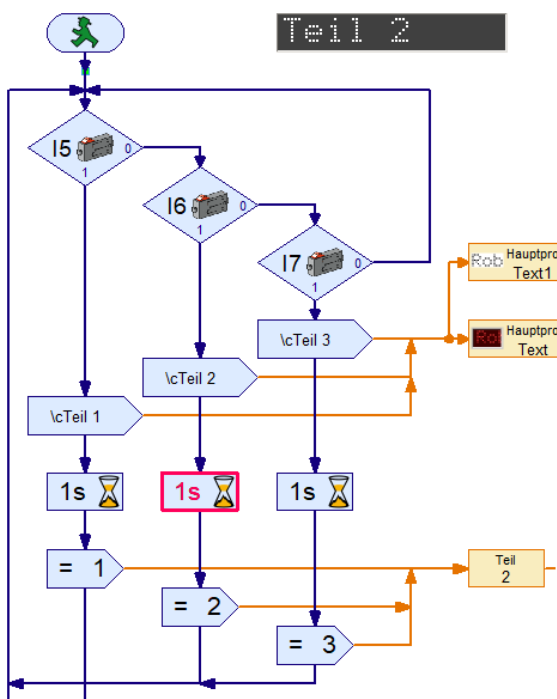
Variante 1 (Auswertung über 3 Eingänge)

Zuerst wurde der Arduino mit dem RFID-RC522 laut Tabelle verbunden. An den Ausgängen D5 bis D7 sind zur Kontrolle noch 3 LED's angeschlossen, die entsprechend dem Tag leuchten.

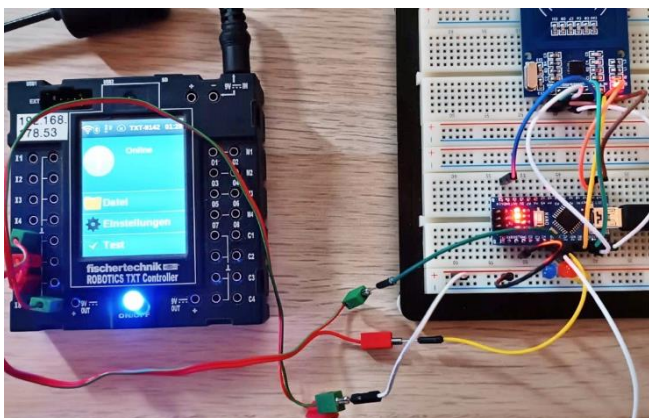
Für die Arduino-Software wird dann nachfolgendes Programm benutzt. Hierbei besteht auch noch die Möglichkeit, sich die ID auf der Konsole anzeigen zu lassen.

```
COM9
Die Kartennummer lautet:141412980
Die Kartennummer lautet:141412980
Die Kartennummer lautet:118791290
```

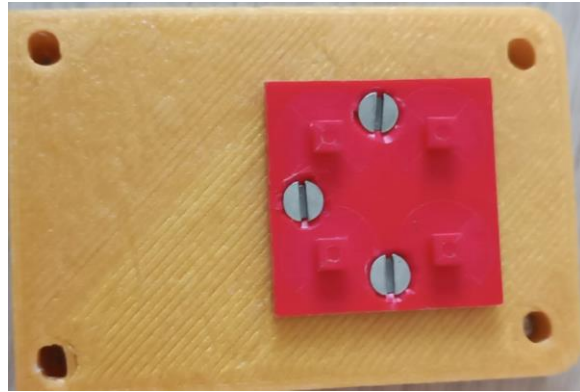
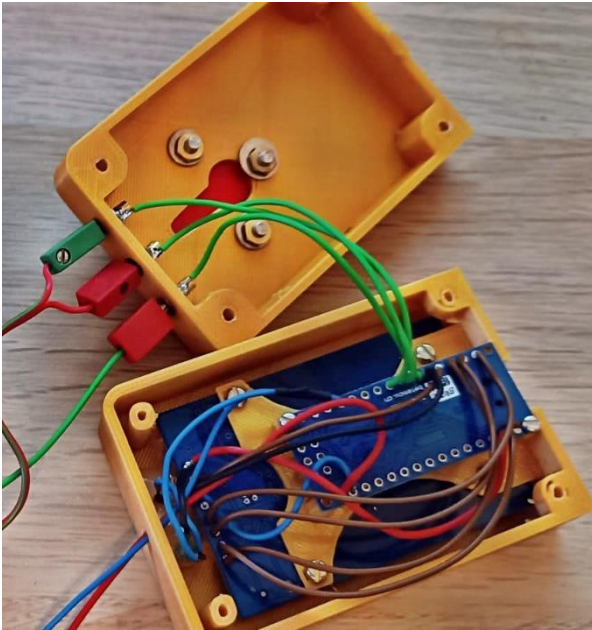
```
#include <MFRC522.h>
#define SS_PIN 10 // SDA an Pin 10
#define RST_PIN 9 // RST an Pin 9
MFRC522 mfrc522(SS_PIN, RST_PIN);
void setup()
{
  Serial.begin(9600);
  SPI.begin();
  mfrc522.PCD_Init();
  pinMode (6, OUTPUT);
  pinMode (5, OUTPUT);
  pinMode (7, OUTPUT);
}
void loop()
{
  if ( ! mfrc522.PICC_IsNewCardPresent() )
  {
    return; // gehe weiter...
  }
  if ( ! mfrc522.PICC_ReadCardSerial() )
  {
    return; //
  }
  long code=0;
  for (byte i = 0; i < mfrc522.uid.size; i++)
  {
    code=((code+mfrc522.uid.uidByte[i])*10);
  }
  Serial.print("Die Kartennummer lautet:");
  Serial.println(code);
  if (code==118791290)
  {
    digitalWrite (6, HIGH);
    delay (2000);
    digitalWrite (6, LOW); }
  if (code==158791290)
  {
    digitalWrite (5, HIGH);
    delay (2000);
    digitalWrite (5, LOW);}
  if (code==141412980)
  {
    digitalWrite (7, HIGH);
    delay (2000);
    digitalWrite (7, LOW);
  }
}
```



Die Arduino-Ausgänge D5 bis D7 und GND wurden direkt mit dem TXT verbunden. Die Eingangsart ist auf 10 V festgelegt. Mit dem abgebildeten Programm, kann nun das Werkstück erkannt werden, um dann weitere Aktionen durchzuführen.



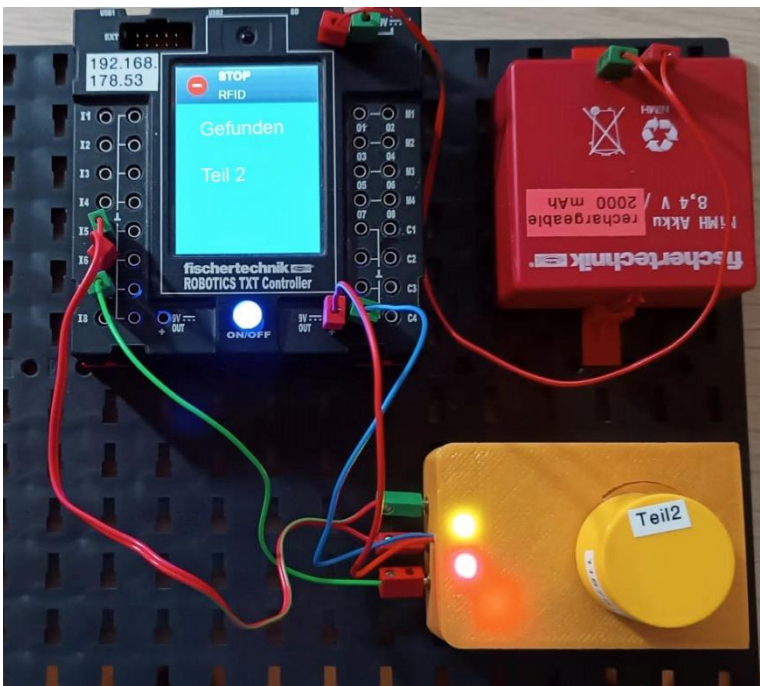
Nach dem letzten Test geht es um das zusammenbauen. Als Gehäusegrundlage habe ich die Vorlage <https://www.thingiverse.com/thing:1413827> benutzt.



Für die Befestigung an den FT-Teilen wurde eine FT-Platte 38259 mit Senkkopfschrauben angebracht.



Während die Kabel für die Stromversorgung direkt herausgeführt werden, sind die 3 Ausgänge als Buchsen angebracht. Dafür habe ich passende Holnieten eingesetzt und von hinten auf eine Unterlegscheibe verlötet.



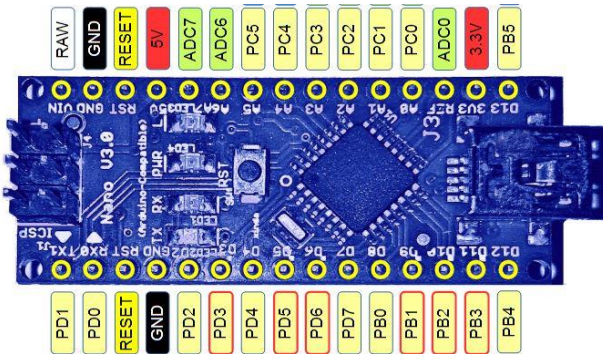
Um die Funktion real zu testen, wurde die Rohrhülse 35409 mit den entsprechenden Deckeln benutzt. Von unten passt genau ein selbstklebender NFC-Tag drauf. Oben wurde die Beschriftung des Werkstückes und an der Seite die Identifizierungsnummer vom Tag aufgeklebt.



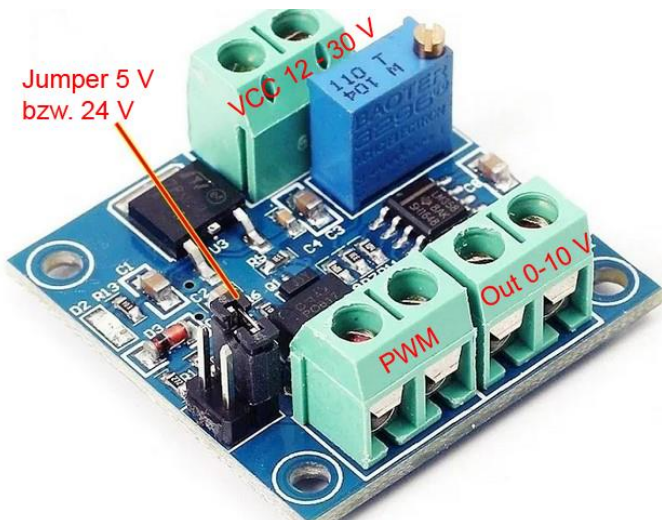
Ein Beispielvideo kann unter chobe.info/Video/RFID.MV4 aufgerufen werden.

Variante 2 (Auswertung über Spannungswert)

In der zweiten Variante wird das PWM-Signal am Ausgang des Arduino mit der RoboPro-Software ausgewertet. Dabei sind 2 Dinge zu beachten. Erstens muss der Arduino-Ausgang für das PWM-Signal geeignet sein (betrifft die rotumrandeten Ausgänge).



Zweitens ist der TXT nicht in der Lage ein PWM-Signal auszuwerten. Dazu kommt ein kleiner Wandler, wie z.B. der LC-LM358-PWM2V zum Einsatz. Er wandelt das digitale PWM-Signal in ein Analogsignal von 0 bis 10 V um.



Die Werte für den PWM-Ausgang sollten experimentell ermittelt werden, da viele Faktoren das Ergebnis beeinflussen. Im Programm wird deshalb neben dem Ergebnis des gesuchten Teils, auch die Eingangsspannung angezeigt.

```
#include <SPI.h>
#include <MFRC522.h>
#define SS_PIN 10 // SDA an Pin 10
#define RST_PIN 9 // RST an Pin 9
MFRC522 mfrc522(SS_PIN, RST_PIN);
void setup()
{
  Serial.begin(9600);
  SPI.begin();
  mfrc522.PCD_Init();
  pinMode (8, OUTPUT);
}
void loop()
{
  if ( ! mfrc522.PICC_IsNewCardPresent() )
  {
    return;
  }
  if ( ! mfrc522.PICC_ReadCardSerial() )
  {
    return;
  }
  long code=0;
  for (byte i = 0; i < mfrc522.uid.size; i++)
  {
    code=((code+mfrc522.uid.uidByte[i])*10);
  }
  Serial.print("Die Kartenummer lautet:");
  Serial.println(code);
  if (code==158791290) //Teil 1
  {
    analogWrite(5,35); // ca. 1 Volt
    digitalWrite(8, HIGH);
    delay (1000);
    digitalWrite(8, LOW);
  }
  if (code==118791290) //Teil 2
  {
    analogWrite(5,75); // ca. 2 Volt
    digitalWrite(8, HIGH);
    delay (1000);
    digitalWrite(8, LOW);
  }
  if (code==141412980) // Teil 3
  {
    analogWrite(5,115); // ca. 3 Volt
    digitalWrite(8, HIGH);
    delay (1000);
    digitalWrite(8, LOW);
  }
}
```

