

Arduino-gesteuerte Schatztruhe mit NFC-Tag als Schlüssel



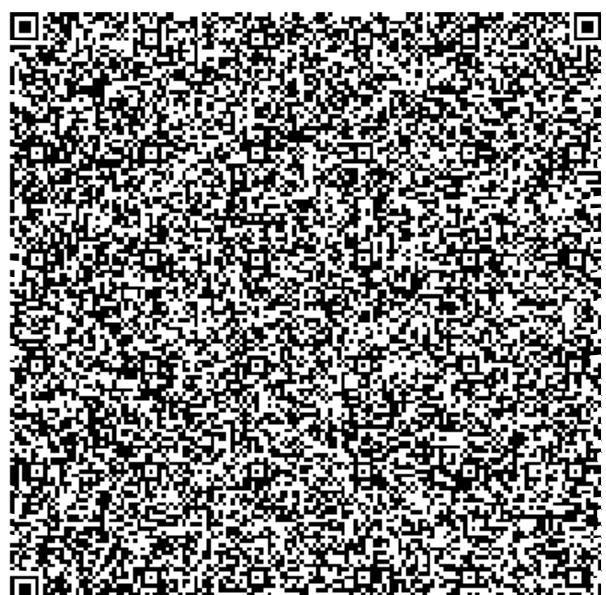
Powerbank für die Stromversorgung: Diese könnte auch über ein Netzgerät oder einen Rechner erfolgen.

Diese Nummer sollte ursprünglich einmalig sein, das konnte leider nicht aufrechterhalten werden. Es lassen sich mittlerweile geklonte Tags programmieren. Darum werden für sicherheitsrelevante Anwendungen, z.B. finanzielle Transaktionen, Ausweisfunktionen oder Transponder-Schließanlagen raffinierte Verschlüsselungsverfahren in die NFC-Technik einbezogen. Für einfache Anwendungen lässt sich aber weiterhin die UID nutzen, vielleicht, um über eine Arduino-Schaltung eine Katzenklappe zu entriegeln oder andere kreative Automatisierungsideen umzusetzen. Wer an der Programmierung interessiert ist, kann auf der Rückseite der Schatztruhe den Arduino-Code (NFC-Riegel-LCD) lesen. Außerdem ist der Code über den QR-Code rechts verfügbar. Aus dem Code ergibt sich, wie die Bauteile an den Arduino angeschlossen werden.



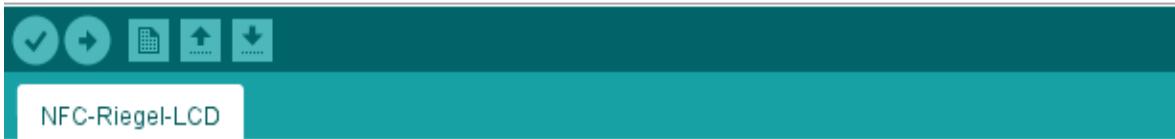
NFC-Transponder als Türöffner sind weitverbreitet. Um ihre Funktionsweise detaillierter verstehbar zu machen, haben wir die Schatztruhe gebaut. Sie zeigt, wie Informationen des NFC-Tags (hier in Form eines Schlüsselanhängers) von einem RFID-Empfänger (hier ein RFID-RC522) erfasst und an den Arduino übertragen werden. Dieser steuert außerdem ein LC-Display (1602A V2.0) und ein Modul mit einem Relais (JQC-3FF-S-Z) an. Das Relais wiederum steuert den elektromagnetischen Riegel, der zurückgezogen wird, wenn das NFC-Tag die richtige Information liefert hat. Der elektromagnetische Riegel, der hier verwendet wird, benötigt 440 mA bei 7 V und wird mit zwei in Reihe geschalteten 4,5-Volt-Flachbatterien betrieben. Er lässt sich ersetzen durch einen Riegel vom Typ „JF-0530B 5V 0,7A“, der an die VIN-Buchs des Arduinos angeschlossen werden kann, wodurch die Schaltung deutlich kompakter wird.

Ein NFC-Tag hat eine Nummer (UID – Unique Identification Number), die es an ein Lesegerät sendet, sobald es Kontakt mit ihm aufnimmt.



NFC-Riegel-LCD | Arduino 1.8.13

Datei Bearbeiten Sketch Werkzeuge Hilfe



```
//Elektromagnetischen Riegel mit NFC und Relais(Tongling JQC-3FF-S-Z) entsperren
#include <Wire.h>
#include <LiquidCrystal_I2C.h> //LC-Display 1602A V2.0
LiquidCrystal_I2C lcd(0x27, 16, 2);
#define I2C_ADDRESS 0x3C
#include <SPI.h>
#include <MFRC522.h> //RFID-RC522 liest NFC
#define S8_PIN 10 //SDA 10, SCK 13, MOSI 11, MISO 12
#define RST_PIN 9 //Wird hier nicht verbunden wegen Komplikationen mit Riegel
#define I2C_ADDRESS 0x3C
MFRC522 mfrc522(S8_PIN, RST_PIN);
void setup(){
pinMode(8, OUTPUT); //Relais
Serial.begin(9600); Serial.println("Bitte NFC-Tag praesentieren!");
Wire.begin(); Wire.setClock(400000L);
lcd.init(); lcd.backlight();
SPI.begin(); mfrc522.PCD_Init();
lcd.clear(); lcd.print("Bitte NFC-Tag"); lcd.setCursor(0,1); lcd.print("praesentieren!");
void loop(){
if ( ! mfrc522.PICC_IsNewCardPresent()) {return;}
if ( ! mfrc522.PICC_ReadCardSerial()) {return;}
byte schluessel_0= mfrc522.uid.uidByte[0];
byte schluessel_1= mfrc522.uid.uidByte[1];
byte schluessel_2= mfrc522.uid.uidByte[2];
byte schluessel_3= mfrc522.uid.uidByte[3];
Serial.print("Die ersten swei Bytes der Kartennummer:");
Serial.print(schluessel_0,HEX); Serial.print(":"); Serial.print(schluessel_1,HEX);
Serial.print(schluessel_2,HEX); Serial.print(":"); Serial.print(schluessel_3,HEX);
lcd.clear(); lcd.print("1.bis2.Byte:"); lcd.setCursor(0,1);
lcd.print(schluessel_0,HEX); lcd.print(" "); lcd.print(schluessel_1,HEX); lcd.print(" ");
lcd.print(schluessel_2,HEX); lcd.print(" "); lcd.print(schluessel_3,HEX);
delay(1000); lcd.clear(); lcd.print("Bitte NFC-Tag"); lcd.setCursor(0,1); lcd.print("praesentieren!");
if(schluessel_0==0xf9){digitalWrite(8, HIGH); lcd.clear();
lcd.print("Riegel fuer"); lcd.setCursor(0,1); lcd.print("5 Sekunden offen!"); delay(5000);
digitalWrite(8, LOW);}
lcd.clear(); lcd.print("Bitte Tag"); lcd.setCursor(0,1); lcd.print("praesentieren!"); delay(500);}
```