

Projekt – Make & Repair

von Gerrit Hardung

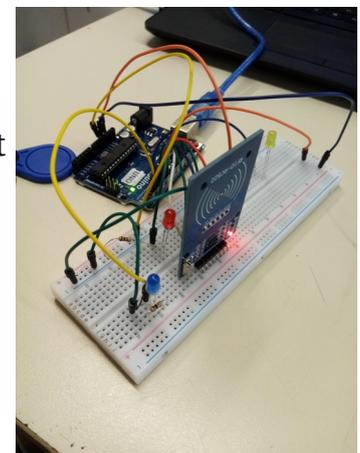
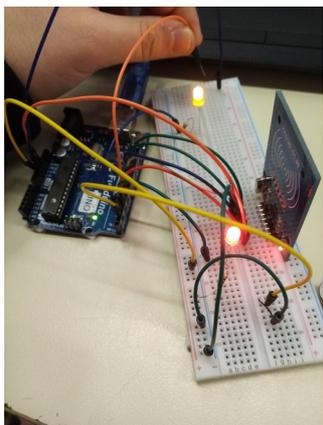
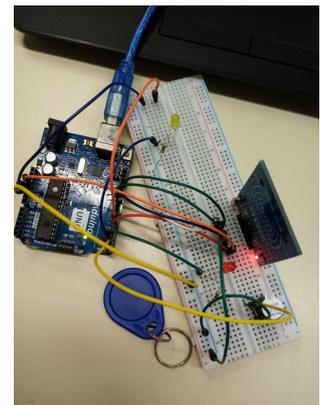
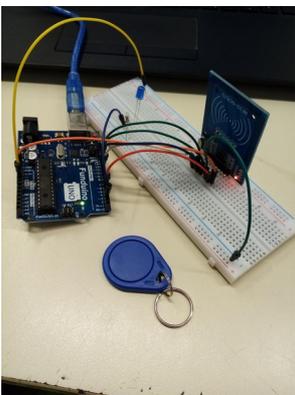
Mikrocontroller: Programmieren einer Alarmanlage mit einem Funduino UNO (Hilfe durch das Buch „Arduino lernen“)

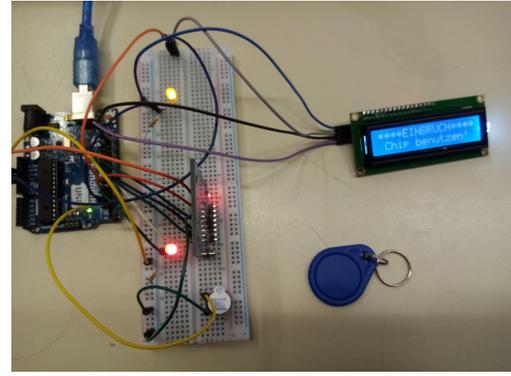
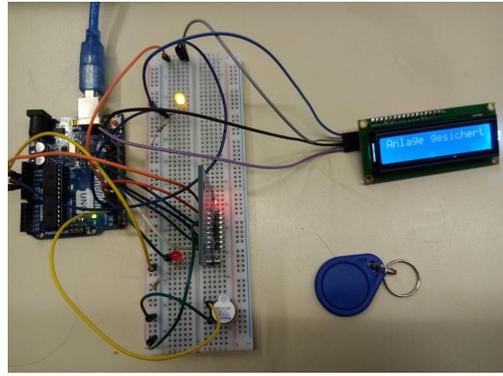
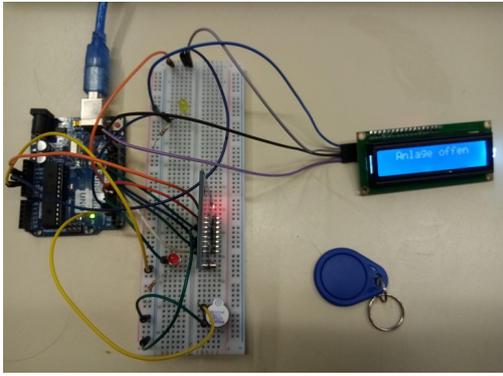
- Signalton (Piezo-Speaker) + blinkende LED
- Alarm sichern durch eine Chipkarte (RFID-Reader + RFID-TAG) + Codeeingabe durch ein Tastenfeld (Keypad)
- Display um Text anzuzeigen (I2C LCD Modul)
- evtl. visualisieren durch eine Tür



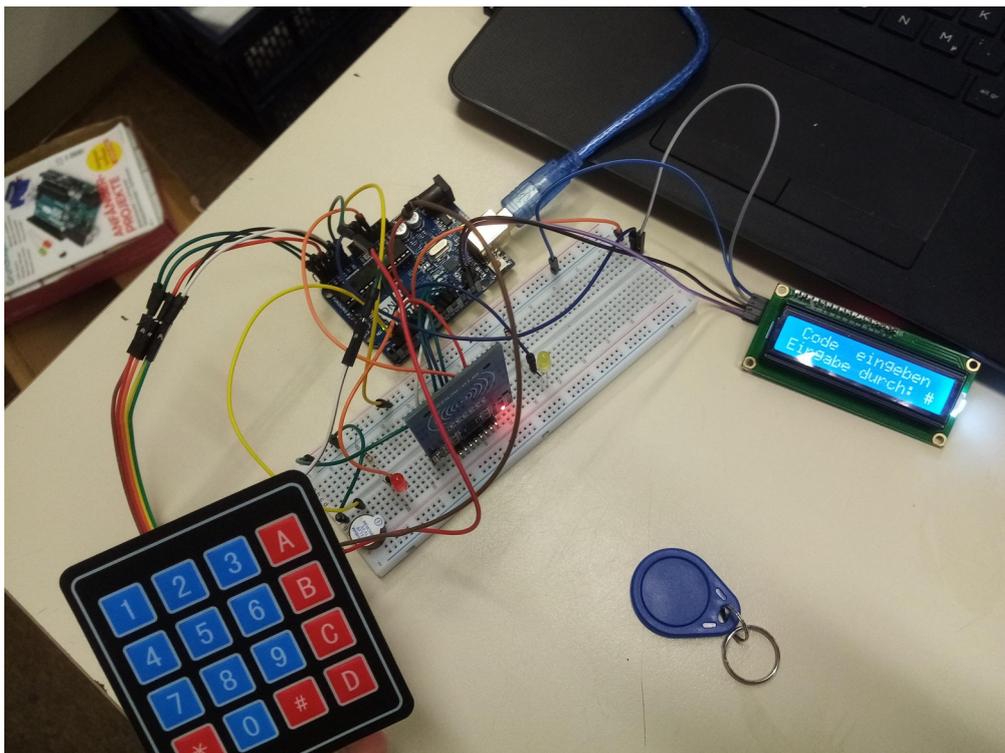
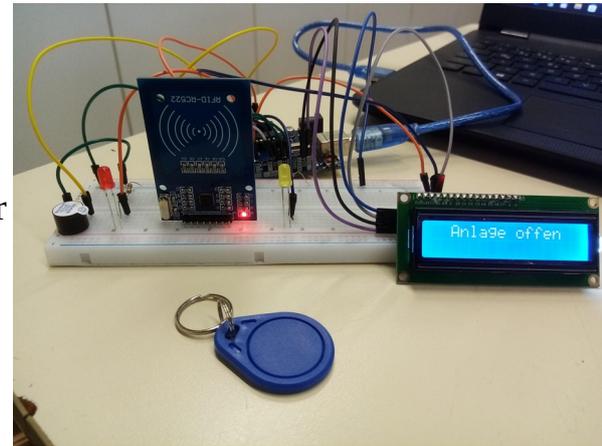
Inhalt:

1. Chipkarte installiert und ans Laufen gebracht
2. gelbe LED angeschlossen, um zu zeigen, wenn die Anlage eingeschaltet ist
3. eine Verbindung angebracht, welche bei Trennung den Alarm auslöst
4. eine rote LED angeschlossen, welche blinkt wenn der Alarm ausgelöst wurde
5. Problem: die Tür wäre nur ohne Alarm zu öffnen, wenn die Chipkarte die ganze Zeit am Signal hängt (wenn das Signal zur Entriegelung kommt muss diese so lange bestehen, bis das Signal erneut kommt)
6. Problem behoben durch Einführung einer Variable „boolean gesichert“
7. Die Alarmanlage ist zu Beginn nicht gesichert
8. Problem: Der Alarm geht auch aus, wenn die „Tür“ zufällt bzw. wieder eine Verbindung besteht (Der Alarm soll erst wieder aus gehen, wenn die Chip erkannt wird)
9. Den Programmcode durch Kommentare ergänzt
10. Problem gelöst durch Einführen zwei boolean Variablen („unterbrochen“ und „alarm“)
11. Platine fürs Chipscannen auf dem Steckbrett umgedreht zur besseren Bedienung
12. Dem Piezo-Speaker durch „tone(pieps,50)“ einen Alarmcharakter gegeben
13. Die Anlage mit einem LCD Display verbunden, um Texte auszugeben





14. Problem: Die Zeitspanne vom Unterbrechen der Verbindung (Tür öffnen) und dem Auslösen des Alarms ist zu lang => Das Display verzögert das Programm
15. Problem behoben durch Reihenfolgenänderung der „print“ Befehle und erneutem ab und an schließen des Displays (1 sek. Abweichung höchstens noch)
16. Steckbrettbelegung geändert zur besseren Übersicht
17. Tastenfeld mit dem Mikrocontroller verbunden und Programm geschrieben
18. Das einzige was fehlt ist, dass der Alarm sofort ausgeht, wenn der Chip gescannt wird und nicht, wie es sein sollte, erst wenn der Code überprüft wurde



Programmcode

```
#include <Keypad.h> //Tastenfeld-Bibliothek einbinden
#include <Wire.h> //benötigt für das Display (Wire-Bibliothek)
#include <LiquidCrystal_I2C.h> //Display-Bibliothek einbinden
LiquidCrystal_I2C lcd(0x27, 16, 2); //um welches Display es sich handelt (mit 16
Zeichen und 2 Zeilen)
#include <SPI.h> //benötigt für die Chipsannung (SPI-Bibliothek)
#include <MFRC522.h> //RFID-Bibliothek
#define SS_PIN 10 //SDA an Pin 10
#define RST_PIN 9 //RST an Pin 9
MFRC522 mfrc522(SS_PIN, RST_PIN); //RFID-Empfänger benennen

int blinker=4; //Alarmlampe
int pieps=2; //Alarmsignal
int aktiv=3; //Sicherheitslampe
boolean gesichert=false; //Anlage aktiv oder nicht
boolean unterbrochen=false; //Tür offen oder nicht
boolean alarm=false; //Alarm an oder aus
char P1='1';char P2='2';char P3='3';char P4='4'; //die einzelnen Passwort Zeichen
char C1, C2, C3, C4; //Variablen für die Eingabe auf dem Tastenfeld (Speicherplatz)
const byte COLS = 4; //wie viele Spalten das Tastenfeld besitzt
const byte ROWS = 4; //wie viele Zeilen das Tastenfeld besitzt
int z1=0, z2, z3, z4; //Variablen um zu zeigen, welches eingegebene Zeichen auf dem
Tastenfeld gespeichert ist(1) und welches noch nicht(0)
char hexaKeys[ROWS][COLS]={ //die Platzierung der Tasten auf dem Tastenfeld
  {'D','#','0','*'}, //letzte Reihe von rechts nach links
  {'C','9','8','7'},
  {'B','6','5','4'},
  {'A','3','2','1'} //erste Reihe von rechts nach links
};
byte colPins[COLS] = {8,6,A0,A1}; //mit welchen Pins auf dem Mikrocontroller...
byte rowPins[ROWS] = {A2,A3,A4,A5}; //...das Tastenfeld verbunden ist
char Taste; //Variable für die eingegebene Taste
Keypad Tastenfeld = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS,
COLS); //Tastenfeld declarieren

void setup()
{
  SPI.begin(); //SPI-Verbindung aufbauen
  pinMode(blinker, OUTPUT); //die rote Alarmleuchte
  pinMode(pieps, OUTPUT); //der Piezo-Speaker
  pinMode(aktiv, OUTPUT); //die gelbe Sicherheitslampe
  pinMode(7, INPUT); //die Verbindung der Tür
  mfrc522.PCD_Init(); //Initialisierung des RFID-Empfängers
```

```

lcd.init();           //Display starten
lcd.backlight();     //Hintergrund des Displays anschalten
lcd.setCursor(0, 0); //Cursor auf die erste Stelle des Displays setzen
lcd.print(" Anlage offen "); //"Anlage offen" auf das Display schreiben
}

void loop()
{
  //Alarm
  Alarm: //hierhin kann zurückgesprungen werden
  if(digitalRead(7)==LOW || alarm==true) //wenn die Tür geöffnet wird oder wenn
der Alarm läuft
  {
    unterbrochen=true; //ist die Anlage sozusagen unterbrochen
  }
  else
  {
    unterbrochen=false;
  }

  //Sicherung
  if(unterbrochen==true && gesichert==true) //wenn die Anlage unterbrochen ist und
die gelbe Sicherheitslampe leuchtet...
  {
    alarm=true; //Alarm aktivieren
    tone(pieps, 50); //...wird der Alarm ausgelöst...
    digitalWrite(blinker, HIGH); //...und die Alarmleuchte blinkt
    delay(200);
    tone(pieps, 100);
    digitalWrite(blinker, LOW);
    delay(200);
    lcd.setCursor(0,0); //erste Zeile
    lcd.print("****EINBRUCH****"); //auf dem Display
    lcd.setCursor(0,1); //zweite Zeile
    lcd.print(" Chip benutzen! "); //auf dem Display
  }
  else
  {
    alarm=false; //Alarm deaktivieren
    noTone(pieps); //sonst ist nichts (die Anlage muss bei einem Alarm zunächst
deaktiviert werden, um den Alarm auszuschalten)
    digitalWrite(blinker, LOW);
    lcd.setCursor(0,1);
    lcd.print(" "); //zweite Zeile löschen
  }
}

```

```

}

//Chipkarte + Tastenfeld
if(mfrc522.PICC_IsNewCardPresent()) //wenn eine Chipkarte erkannt wird...
{
  if(mfrc522.PICC_ReadCardSerial()) //...und ein RFID Sender diese liest
  {
    long code=0; //Variable code (länger als int)
    for(byte i=0; i<mfrc522.uid.size; i++)
    {
      code=((code+mfrc522.uid.uidByte[i])*10); //der Chipkartencode wird
ausgelesen
    }
    if(code==1114960 || 687930) //wenn der Code zu den beiden möglichen Karten
passt (Chip oder Karte)
    {
      Anfang: //hierhin kann zurückgesprungen werden
      Taste = Tastenfeld.getKey(); //die gedrückte Taste wird in "Taste" gespeichert
      lcd.setCursor(0,0); //erste Zeile
      lcd.print(" Code eingeben "); //auf dem Display
      lcd.setCursor(0,1); //zweite Zeile
      lcd.print("Eingabe durch: #"); //auf dem Display
      if(Taste) //wenn eine Taste gedrückt wurde...*(siehe unten bei
"//Tasteneingabe")
      {
        if(Taste=='#') //nach der Tasteneingabe (unten) muss das # gedrückt werden,
um überprüft zu werden
        {
          if(C1==P1&&C2==P2&&C3==P3&&C4==P4) //wenn die Tastenfolge
korrekt ist
          {
            z1=0; z2=1; z3=1; z4=1; //wird gezeigt, dass bei den nächsten Abfragen die
erste Taste wieder gespeichert werden muss
            if(gesichert==true) //falls die Sicherheitsanlage aktiv ist...
            {
              gesichert=false; //...wird diese ausgeschaltet (entsichert)...
              digitalWrite(aktiv, LOW); //...und die Sicherheitslampe geht aus (um u
zeigen, dass es sicher ist die Tür zu öffnen)
              lcd.setCursor(0,0); //erste Zeile
              lcd.print(" Anlage offen "); //auf dem Display
              lcd.setCursor(0,1); //zweite Zeile
              lcd.print(" "); //zweite Zeile löschen
              goto Alarm; //zurück zur Alarm Abfrage
            }
          }
        }
      }
    }
  }
}

```

```

else
{
    gesichert=true; //sonst wird die Sicherheitsanlage aktiviert...
    digitalWrite(aktiv, HIGH); //...und die Sicherheitslampe angeschaltet
(Anlage aktiv)
    lcd.setCursor(0,0); //erste Zeile
    lcd.print("Anlage gesichert"); //auf dem Display
    lcd.setCursor(0,1); //zweite Zeile
    lcd.print(" "); //zweite Zeile löschen
    alarm = false; //der Alarm wird ausgeschaltet, wenn nach auslösen eines
(durch falsche Tasteneingabe) der Chip benutzt wird
    goto Alarm; //zurück zur Alarm Abfrage
}
}
else //wenn die Tastenfolge nicht korrekt ist...
{
    lcd.setCursor(0,0); //erste Zeile
    lcd.print(" Falscher Code! "); //auf dem Display
    lcd.setCursor(0,1); //zweite Zeile
    lcd.print(" "); //zweite Zeile löschen
    alarm = true; //...wird der Alarm aktiviert (um bei der Alarm
abfrage auszulösen)
    delay(1000);
    z1=0; z2=1; z3=1; z4=1; //und es wird gezeigt, dass bei den nächsten
Abfragen die erste Taste wieder gespeichert werden muss
    if(gesichert==false) //wenn die Anlage nicht gesichert war...
    {
        goto Anfang; //...muss die Tasteneingabe erneut erfolgen
    }
    else
    {
        goto Alarm; //...sonst wird der Alarm ausgelöst
    }
}
}

//Tasteneingabe
if(z1==0) /*...wird überprüft, ob die erste Taste gespeichert wurde...
{
    C1=Taste; //Taste wird gespeichert
    z1=1; z2=0; z3=1; z4=1; //und es wird gezeigt, dass jetzt die zweite
gespeichert werden muss
    goto Anfang; //zurück zum Anfang (wo gefragt wird ob eine Taste gedrückt
wurde)
}
}

```

```

if(z2==0) //...oder die zweite...
{
    C2=Taste;    //Taste wird gespeichert
    z1=1; z2=1; z3=0; z4=1; //und es wird gezeigt, dass jetzt die dritte gespeichert
werden muss
    goto Anfang; //zurück zum Anfang (wo gefragt wird ob eine Taste gedrückt
wurde)
}
if(z3==0) //...oder die dritte...
{
    C3=Taste;    //Taste wird gespeichert
    z1=1; z2=1; z3=1; z4=0; //und es wird gezeigt, dass jetzt die vierte
gespeichert werden muss
    goto Anfang; //zurück zum Anfang (wo gefragt wird ob eine Taste gedrückt
wurde)
}
if(z4==0) //...oder die vierte
{
    C4=Taste;    //Taste wird gespeichert
    z1=1; z2=1; z3=1; z4=1; //und es wird gezeigt, dass jetzt alle gespeichert
wurden
}
}
goto Anfang; //zurück zum Anfang (wo gefragt wird ob eine Taste gedrückt
wurde)
}
}
}
}
}
}
}

```