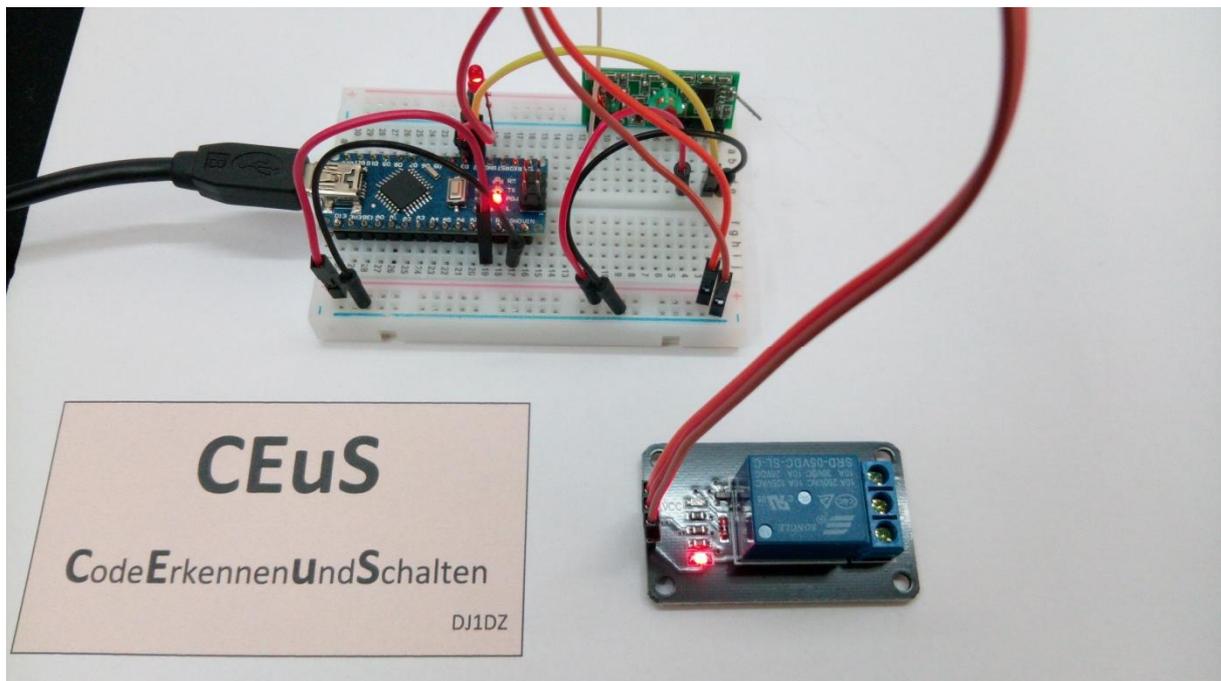


CEuS



Autor : W.Breiling – DJ1DZ – www.dj1dz.darc.de
Entstanden aus: <https://github.com/sui77/rc-switch/>

Autor : W.Breiling – DJ1DZ – www.dj1dz.darc.de
21.10.2017

Inhaltsverzeichnis

	Seite
Vorwort	4
Das Konzept	5
Der Probeaufbau	6
Der Arduino nano	7
Die Leuchtdiode	7
Der Empfänger	8
Das Relais	8
Das Programm CEuS.ino	9
Stichwortsuche	11
Notizen:.....	12

Vorwort

Die Anregung zu diesem Projekt ist aus der Idee entstanden, einen bestimmten Code per Funk zu senden, der einen Schaltvorgang startet. Hier im konkreten Fall soll eine Warnleuchte blinken, sobald ein Bewegungsmelder das Betreten eines Raumes erkennt. Der Auslöser kann aber auch ein beliebiges Gerät sein, das im Erkennungsfall ein Signal aussendet, welches vom Empfänger dekodiert werden kann.

All das soll natürlich wenig kosten und der Aufbau soll in wenigen Schritten realisierbar sein. Das Konzept wird in den nachfolgenden Kapiteln klargelegt.

Das Konzept

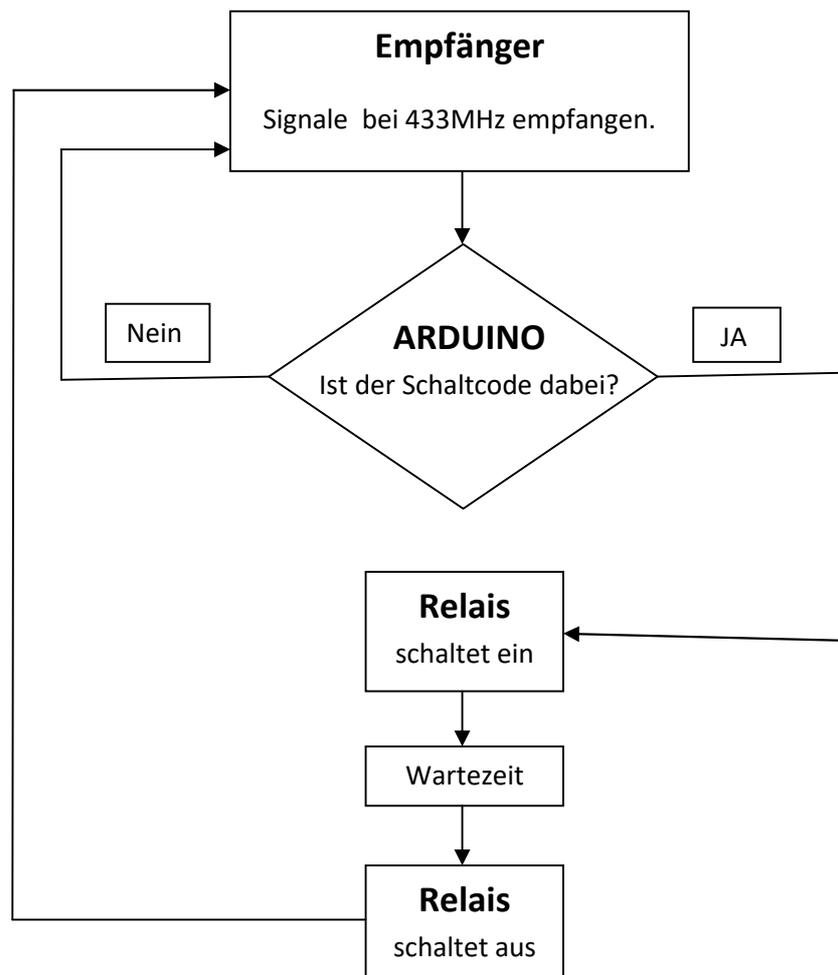
Der Aufbau ist einfach. Zum Probeaufbau benötigen wir ein Steckboard 400. Der Probeaufbau ist wichtig, um das Zusammenspielen der Komponenten einfach zu testen, um spätere Änderungen zu vermeiden.

Wir benötigen nachstehende Teile.

1. Steckboard 400
2. Arduino nano
3. Empfängermodul XY-MK-5V
4. Leuchtdiode mit Vorwiderstand 330 Ohm
5. Relais Modul 5V/230V mit Optokoppler
6. Die 230V Blinklampe oder was auch immer geschaltet wird.

Der Empfänger soll im Bereich 344MHz alles empfangen und an den Arduino weitergeben. Hier wird geprüft ob der eingetragene Code vorhanden ist. Wenn ja, dann schalte das Relais ein, wartet eine definierte Zeit ab und schalte wieder aus.

Das sieht dann so aus:



Der Probeaufbau

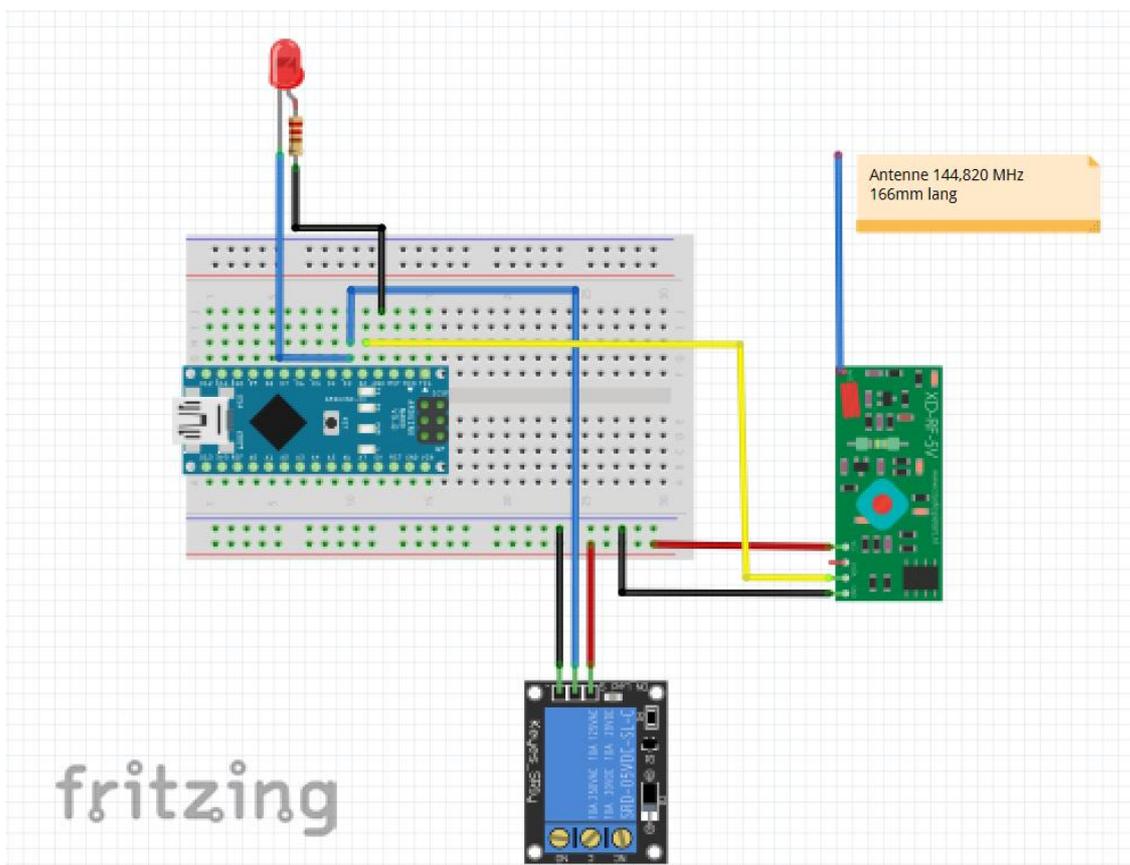
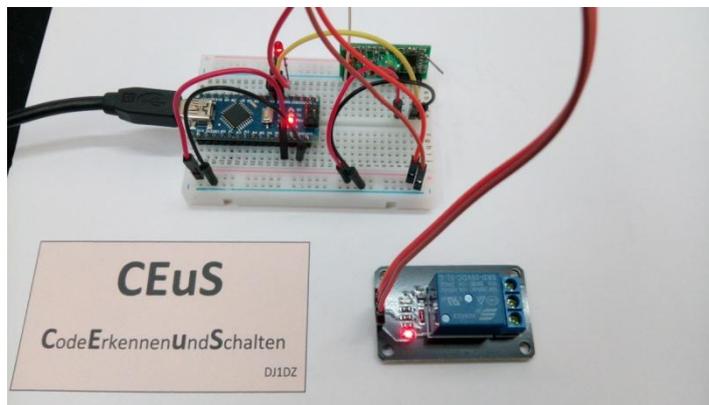
Wir benutzen das Steckboard 400 und platzieren unsere Bauteile wie nachstehend abgebildet.

Das Relais wird an Plus 5v und an GND angeschlossen. Der Schaltkontakt liegt am Arduino nano an D3.

Das Empfangsmodul hat auch die Anschlüsse an Plus 5V und GND. Die Signalleitung wird an D2 angeschlossen. D2 wird als Interrupt Kontakt benutzt. Er erkennt wenn am Empfangsmodul Daten anliegen.

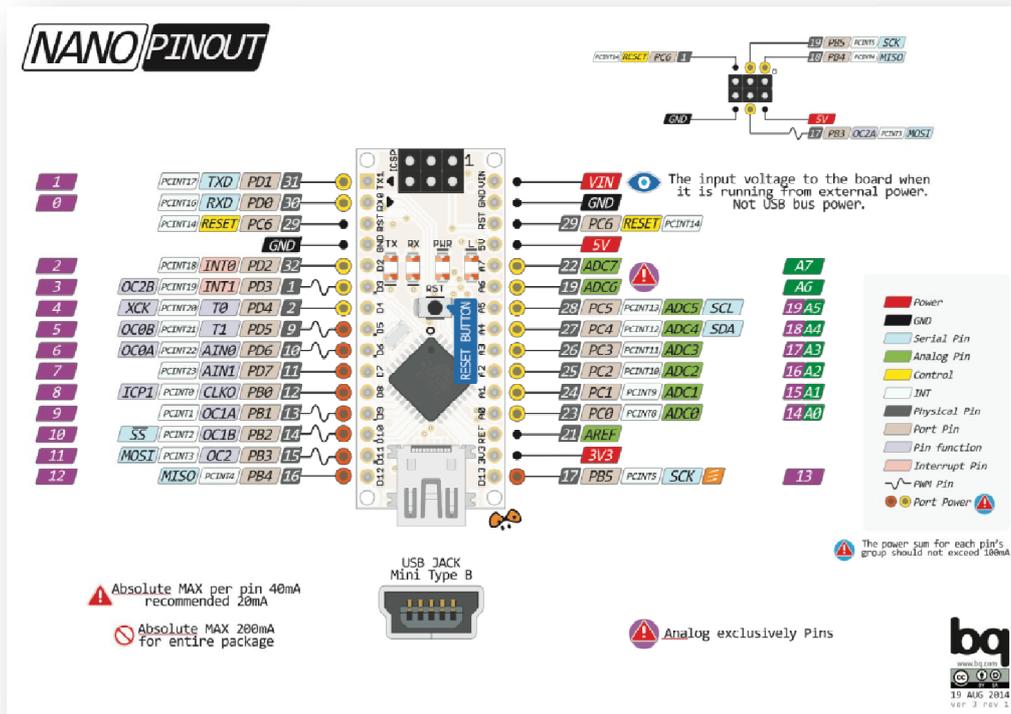
Die Leuchtdiode LED zeigt den Betriebszustand und erlischt sobald das Relais geschaltet wird und leuchtet sobald das Relais wieder abfällt.

Der Arduino nano wird über die USB-Schnittstelle mit Strom versorgt und gibt uns die empfangenen Daten im Probeaufbau am Arduino Monitor aus.



Der Arduino nano

Um mal eine Vorstellung zu bekommen, welche Anschlussmöglichkeiten der Arduino nano bietet, hier mal eine ausführliche Darstellung.



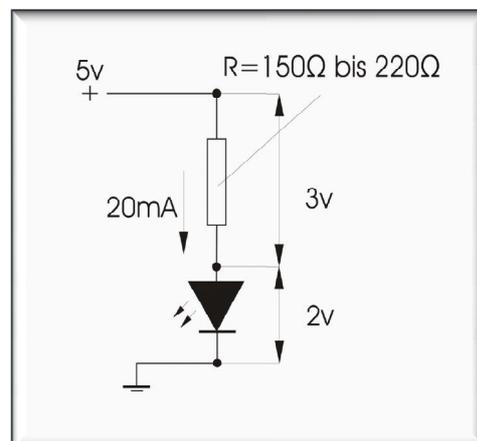
Die Leuchtdiode

Um Schaltzustände nach außen sichtbar zu machen, benutzen wir eine rote Leuchtdiode (LED). Diese wird laut Datenblatt mit 2 Volt und 20 mA betrieben.

Aus dem Arduino Nano wird die LED mit knapp 5Volt angesteuert. Also müssen wir 3Volt über den Vorwiderstand abfallen lassen. Nach dem Ohmschen Gesetz: $U = R \cdot I$ und nach $R = U / I$ umgestellt ist

$$R = 3V / 0,02A = 150 \text{ Ohm}$$

Der kleinste Widerstand sollte also 150 Ohm, besser 220 Ohm sein. Bei einem etwa doppelten Wert (möglich ist 470 Ohm) fließt dann auch nur die Hälfte an Strom. Allerdings, je höher der Widerstand, desto geringer auch die Leuchtkraft. Aber wir wollen ja auch Strom sparen.



Der Empfänger

Als Empfänger wählen wir den XY-MK-5V. Dieser ist leicht und günstig zu bekommen, meist sogar mit Sender. Eine Antenne, die auf die Frequenz 433,820Mhz abgestimmt ist, wird auf die Länge wie folgt berechnet:

$$\gamma = c / f$$

c — Lichtgeschwindigkeit [300.000.000 m/s]
 f — Frequenz [433.820.000Hz]
 Wellenlänge $300.000.000 / 433.820.000 = 0,692\text{m}$ [m/s x s = m]

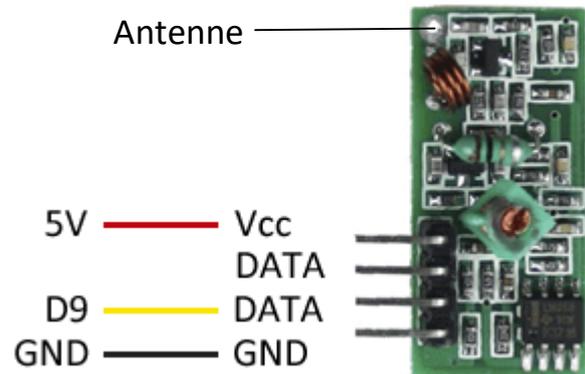
Oder einfach $\gamma = 300 / f$ [MHz] also $300 / 433,82\text{MHz} = 0,692\text{m}$

Für $\gamma / 2$ gilt $0,692 / 2 \times 0,96 = 0,332\text{m}$

Für $\gamma / 4$ gilt $0,692 / 4 \times 0,96 = 0,166\text{m}$

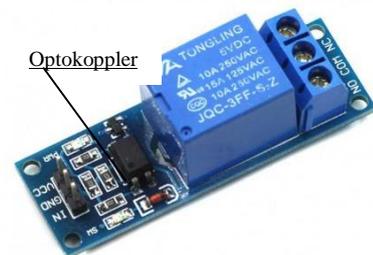
Verkürzungsfaktor = 0,96

Ist alles gut abgestimmt sind Verbindungen bis 50m die Regel. Für eine 100m² Wohnung dürfte das ausreichend sein.



Das Relais

Ein idealer Baustein zum Ein- und Ausschalten, insbesondere für das 230V-Netz. Auf der einen Seite wird mit 5V angesteuert und auf der anderen Seite können Ströme bis 10A und bis 230V geschaltet werden. Und weil Niederspannung (5V) und Hochspannung (230V) so dicht beianander liegen, sollte man darauf achten, dass ein Optokoppler auf der Relaischaltung vorhanden ist. Denn es wird auch hier, wie beim Lichtschalter, **nur** der **L-Leiter** geschaltet.



Das Programm CEuS.ino

```

/*Programm: CEuS.ino - Code Erkennen und Schalten -
https://github.com/sui77/rc-switch/
https://dj1dz.darc.de
*/
#include <RCSwitch.h>
RCSwitch mySwitch = RCSwitch();
int Relais = 3; //Das Relais wird über D3 geschaltet
String CODEein="2196628"; // dieser Code löst den Schaltvorgang aus
String CODEaus="348161"; // dieser Code wird nicht benötigt
String Rec;
void setup() {
  Serial.begin(9600);
  mySwitch.enableReceive(0); // Receiver an interrupt INTO => das ist pin D2

  Serial.println("#####"); // Monitorausgabe
  Serial.println("##### CEuS.ino #####"); // Monitorausgabe
  Serial.println("##### CODE Erkennung und Schalten #####"); // Monitorausgabe
  Serial.println("#####dj1dz#"); // Monitorausgabe
  Serial.println(" "); // Monitorausgabe

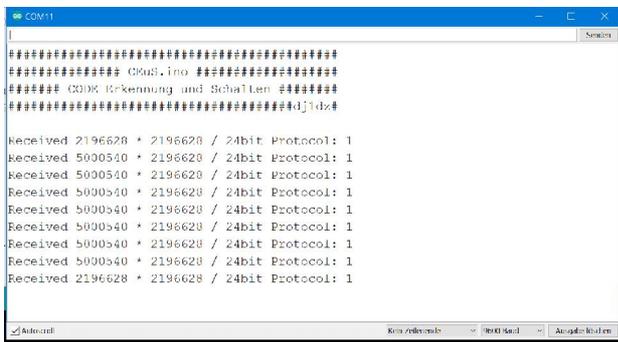
  pinMode(Relais, OUTPUT);
  digitalWrite (Relais, HIGH);
}

void loop() // Hauptschleife
{
  if (mySwitch.available()) {
    int value = mySwitch.getReceivedValue();
    if (value == 0) {
      Serial.print("Unknown encoding"); // Monitorausgabe
    } else {
      Serial.print("Received "); // Monitorausgabe
      Rec=mySwitch.getReceivedValue();
      Serial.print( mySwitch.getReceivedValue() ); // Monitorausgabe
      // Monitorausgabe
      Serial.print(" * ");
      Serial.print(CODEein);
      Serial.print(" / ");
      Serial.print( mySwitch.getReceivedBitlength() );

```

```
Serial.print("bit ");
Serial.print("Protocol: ");
Serial.println( mySwitch.getReceivedProtocol() );
if (Rec== CODEein)
  { digitalWrite (Relais, LOW);
    delay (1200);
    digitalWrite (Relais, HIGH);
  }
}
mySwitch.resetAvailable();
}
} // ProgrammEnde
```

Die Ausgabe am Arduino Programmeditor sieht dann so aus.



```
#####
##### C00d1.ino #####
##### C00K Erkennung und Schalten #####
#####dj1dz#
Received 2196628 * 2196628 / 24bit Protocol: 1
Received 5000540 * 2196628 / 24bit Protocol: 1
Received 2196628 * 2196628 / 24bit Protocol: 1
```

Stichwortsuche

Index

344MHz	5	Konzept	3, 4, 5
Arduino	3, 5, 6, 8	LED	6, 8
Arduino nano	5	Leuchtdiode	5
Arduino Programmeditor	11	Lichtgeschwindigkeit	9
Autor	1, 2	Monitor	6
DJ1DZ	1, 2	Monitorausgabe	10, 11
Empfänger	3, 4, 5, 9	Ohmschen Gesetz	8
Empfängermodul	5	Optokoppler	5, 9
Empfangsmodul	6	Plus 5V	6
Frequenz	9	Relais	3, 5, 6, 9, 10, 11
github	1, 10	Steckboard 400	5, 6
GND	6	Verkürzungsfaktor	9
Hauptschleife	10	Wellenlänge	9
Interrupt	6	Widerstand	8

Notizen:
