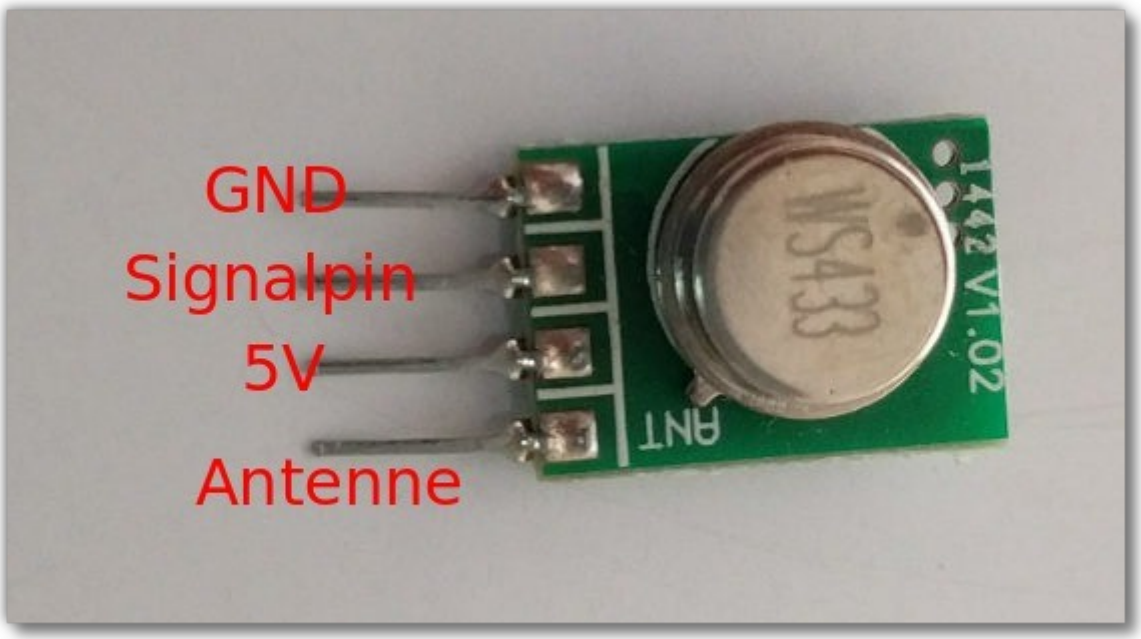
Funkmorsen an die Steckdose

Vorbemerkungen

Die Funksteckdosen haben einen eigenen „Morsecode“ - den man erst mal rauskriegen muss. Das kann man z.B. mit der Soudkarte und einem Aufnahmeprogramm machen.

- https://web.archive.org/web/20190606165653/http://www.dserv01.de/howtos/funksteckdose-fernsteuern-mit-arduino/

Der Sender



Der Code, um eine Steckdose anzusteuern, besteht aus 12 „Zeichen“ – 0 und 1 – die jeweils mit 8 Bit (HIGH/LOW) codiert sind. Das Ende eines Codes wird durch ein „Sync-Signal“ markiert (x). Dieses Sync-Signal besteht aus 32bit, wie in der Tabelle zu sehen ist. Die einzelnen „Zeichen“ sind ca 2.8ms lang, ein Bit dauert etwa 350us:

Code-Zeichen	Bitfolge	Erklärung
0	10001110	8bits zu 350us insgesamt 2.8ms
1	10001000	8bits zu 350us insgesamt 2.8ms
x	10000000000000000000000000000000	32bits zu 350us insgesamt 11.2ms

Jetzt kann man für jede Taste einen Code bilden, die 0 und 1 in der folgenden Tabelle stehen für die Code-Zeichen (nicht für die einzelnen Bits).

	On	Off
A	000111000010x	000111000001x

B	000110100010x	000110100001x
C	000110010010x	000110010001x
D	000110001010x	000110001001x

Man sieht, dass die ersten 10 Zeichen die Steckdose codieren, die letzten beiden Zeichen übermitteln die Information, ob die Steckdose an oder aus sein soll:

```
 0001110000      10      x
Steckdosencode AN/AUS  sync
```

- Rohform erster Sketch: funksteckdose.ino.zip

funksteckdose.ino

```
// Funksteckdose Brennenstuhl RCS 1000N
// mit einem Arduino Uno und einem 433 MHZ Modul
// schalten.
//
// Original Ideen und Sketche:
// CC-BY-NC Dominik Krupke
// http://creativecommons.org/licenses/by-nc-sa/3.0/deed.de
// http://www.dserv01.de/howtos/funksteckdose-fernsteuern-mit-arduino/
//
// Modifiziert, verschandelt und angepasst:
// Frank Schiebel 06/2016
//


// Der Pin auf dem der Datenpin des 433MHz
// Senders angeschlossen ist.
short rc_pin=7;

void setup() {
  // Datenpin fungiert als Ausgabepin
  pinMode(rc_pin, OUTPUT);
}

void loop() {
  sendCode("100011000001"); //sende den Beispiel Code[Aus]
  delay(2000);
  sendCode("100011000010"); //sende den Beispiel Code[An]
  delay(2000);
}

// -----------------
// Eigene Funktionen
// -----------------

//
// Sendet einen Steckdosencode
```

```
// Eingabe: 12-stelliger Code, 10 Stellen Steckdose, zwei Stellen
an/aus
// Rueckgabe: true/false
//
boolean sendCode(const char code[]) {
  // Der Code wird 7 mal gesendet
  for(short z = 0; z<7; z++) {
     // Jeder Code besteht aus den übergebenen 12 Bit, die
     // einzeln gesendet werden.
    for(short i = 0; i<12; i++) {
      // Jedes Bit wird von der Funktion sendByte einzeln
      // über den 433 MHz Sender gesendet
      sendByte(code[i]);
    }

  // Jeder Code schließt mit x/sync ab.
  // Immer anhaengen!
  sendByte('x');

  }
  return true;
}

// Diese Funktion soll 0,1 oder x senden koennen.
// Die gewuenschte Ausgabe wird als Argument an die
// Funktion uebergeben:
//
//  sendByte(0); // sendet 0
//  sendByte(1); // sendet 1
//  sendByte(x); // sendet x als "Trennsignal"
//
// Die Funktion hat einen leeren (void) Rueckgabewert
// In der Funktion wird die "Unterfunktion" wait
// aufgerufen, die jeweils die uebergebene Anzahl
// an 350 Mikrosekunden Intervallen wartet.
void sendByte(char i) {

  switch(i){
    // Wenn i=0 ist, wird der Code fuer '0' gesendet
    case '0':{
      digitalWrite(rc_pin,HIGH);
      wait(1); //da die Pausen x*350us lang sind, machen wir daraus
eine Funktion
      digitalWrite(rc_pin,LOW);
      wait(3);
      digitalWrite(rc_pin,HIGH);
      wait(3);
      digitalWrite(rc_pin,LOW);
      wait(1);
      return;
    }
```

```
    // Wenn i=1 ist, wird der Code fuer '1' gesendet
    case '1':{ //Der Code fuer '1'
      digitalWrite(rc_pin,HIGH);
      wait(1);
      digitalWrite(rc_pin,LOW);
      wait(3);
      digitalWrite(rc_pin,HIGH);
      wait(1);
      digitalWrite(rc_pin,LOW);
      wait(3);
      return;
    }
    // Wenn i=x ist, wird der Code fuer '1' gesendet
    case 'x':{ //Der Code fuer x(sync)
      digitalWrite(rc_pin,HIGH);
      wait(1);
      digitalWrite(rc_pin,LOW);
      wait(31);
    }

  }
}

// Wartet fuer x 350Mikrosekunden-Intervalle
void wait(int x) {
  //warte x*350 Mikrosekunden
  delayMicroseconds(x*350);
}
```

From:
https://wiki.qg-moessingen.de/ - **QG Wiki**

Permanent link:
https://wiki.qg-moessingen.de/faecher:nwt:bluetooth_steckdosen:433_mhz:funksteckdose:start?rev=1676397813

Last update: **14.02.2023 19:03**