

433 Mhz – Funksender und -empfänger

Informationen (Daten) können durch Wellen in Lichtgeschwindigkeit drahtlos übertragen werden. Ein sehr bekanntes Beispiel ist die Übertragung von Daten in und vom Smartphone (*Referat: Funktionsprinzip der Übertragung von einem Handy zu einem anderen*).

Das charakteristische Merkmal von Wellen ist die Frequenz – also die Anzahl der Schwingungen pro Sekunde. Die Einheit ist das Hertz. Ein Hertz ist eine Schwingung pro Sekunde.

Sprache oder Musik, die wir hören, werden mit Hilfe von Wellen (Schall) in der Luft übertragen. Menschen hören typischerweise Frequenzen zwischen 50 Hz bis 15000 Hz – abhängig vom Alter. Die Geschwindigkeit von Schallwellen in Luft beträgt nur etwa 340 m/s.

Elektromagnetische Wellen können Informationen auch im Vakuum und wie bereits gesagt mit Lichtgeschwindigkeit (300000 km/s im Vakuum) übertragen. Abhängig von der Frequenz können bestimmte Stoffe mehr oder weniger gut von ihnen durchdrungen werden. Die Reichweite der Wellen ist ebenfalls stark von der Frequenz (und natürlich von der Sendeleistung) abhängig und kann zwischen wenigen Zentimetern und mehreren tausend Kilometern variieren.

Das Prinzip ist folgendes: Eine Trägerwelle wird durch die eigentliche Information moduliert, d.h. leicht verändert. Eine heutzutage kaum noch verwendete Technik ist die Amplitudenmodulation (AM) – die Amplitude des Trägersignals wird verändert.

(*Referat: Geschichte und Prinzip der Rundfunktechnik*)

Man kann auch die Frequenz der Trägerwelle ändern (FM:Frequenzmodulation).

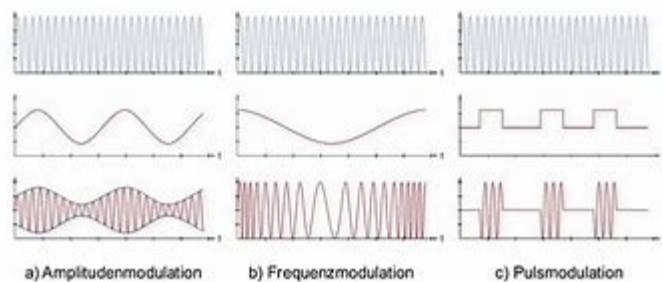
Die Frequenzen der Trägerwellen reichen von 100KHz bis zu mehreren Ghz.

Die Nutzung dieser Frequenzbereiche wird vom Staat geregelt, auch um z.B. gleichzeitige Nutzung der gleichen Frequenzen zu verhindern.

Es gibt aber freie Frequenzbereiche, die jedermann benutzen kann – natürlich auf die Gefahr der gegenseitigen Störung durch gleichzeitige Nutzung.

Als **ISM-Bänder** (Industrial, **S**cientific and **M**edicalBand) werden **Frequenzbereiche** bezeichnet, die durch Geräte in Industrie, Wissenschaft, Medizin, in häuslichen und ähnlichen Bereichen lizenzfrei und meist genehmigungsfrei genutzt werden können.

Beispiele: Babyphone (27 MHz, 433 MHz)
 Bluetooth (2,4 Ghz)
 Modellbau-Fernsteuerungen (27 Mhz)
 WLAN (2,4 Ghz, 5 Ghz, 60 Ghz)



In unserem Aufbau verwenden wir 433 MHz, die in einem freien Band liegen.

Sender:

Es gibt bei diesen Modulen verschiedene Sendemöglichkeiten und Protokolle, wie das Senden von Binärcode, Dezimalzahlen oder Tri-state. Wir werden uns in dieser Anleitung auf das Senden von Dezimalzahlen beschränken.

Empfänger:

Auf der 433MHz Frequenz prüft der Empfänger stetig, ob Signale empfangen werden. Sobald der Empfänger ein Signal aufzeichnet, unterbricht der Arduino Mikrocontroller das Programm (Loop). Die empfangenen Daten können dann im Serial Monitor angezeigt und verarbeitet werden. Der Empfänger wird mit 5V betrieben. Eine höhere Spannung führt hier nicht zu einer höheren Reichweite. Jedoch macht eine Antenne auch beim Empfänger Sinn.

Für den Empfänger und für den Sender gibt es in den folgenden Anleitungen jeweils unterschiedliche Sketche, da beide völlig unterschiedliche Aufgaben zu erfüllen haben. In dieser Anleitung verwenden wir die RCSSwitch Library. Sie enthält neben der notwendigen Library auch einige einfache Beispielcodes, die wir für diese Anleitung verwenden und erweitern werden.

Material: Arduino / Breadboard / Breadboardkabel / 433MHz Sender und Empfänger / Stromversorgung oder ein zweites Arduino-Board, da Sender und Empfänger nicht gleichzeitig an einem Board betrieben werden.

Für diese Anleitung wird die RCSSwitch Library benötigt: <https://github.com/sui77/rc-switch>
Die Library muss installiert werden bevor man den Sketch verwendet.

Sketch 1: Senden und empfangen

Mit dem Sender soll ein 433MHz Signal gesendet werden. Ein zweiter Arduino Mikrocontroller empfängt die Daten und zeigt sie im Serial-Monitor an.

Der Sender:

Der Sender hat drei Anschlüsse.

Beim Blick auf die Vorderseite:

links: GND / in der Mitte: VCC / rechts: DATA

Um den Sender zu verkabeln, wird der Data-Pin mit Pin 10 des Arduinos, VCC mit 5V und GND mit GND verbunden.



Der Sketch (Sender):

```
#include <RCSSwitch.h>
```

```
RCSSwitch mySwitch = RCSSwitch();
```

```
void setup() {
```

```
  mySwitch.enableTransmit(10); // Der Sender wird an Pin 10 angeschlossen
```

```
}
```

```
void loop() {
```

```
  mySwitch.send(1234, 24); // Der 433MHz Sender versendet die Dezimalzahl „1234“
```

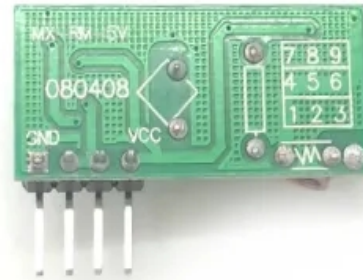
```

delay(1000);           // Eine Sekunde Pause, danach startet der Sketch von vorne
}

```

Der Empfänger

Der Empfänger hat 4 Anschlüsse.
 Beim Blick auf die Rückseite:
 links: GND / in der Mitte(2x): Data / rechts: VCC



Zur Verwendung wird ein (beliebiger) Data-Pin mit dem Pin 2 des Arduinos verbunden.
 Im Sketch wird dieser als Pin 0 beschrieben, da er der Interrupt-Pin 0 ist.

Der Sketch (Empfänger):

```

#include <RCSwitch.h>

RCSwitch mySwitch = RCSwitch();

void setup() {
  Serial.begin(9600);
  mySwitch.enableReceive(0); // Empfänger ist an Interrupt-Pin "0" - also am UNO der Pin2
}

void loop() {
  if (mySwitch.available()) // Wenn ein Code empfangen wird...
  {
    int value = mySwitch.getReceivedValue(); // empfangene Daten werden gespeichert.
    if (value == 0) // Wenn die empfangenen Daten "0" sind, wird "Unbekannter Code" angezeigt.
      { Serial.println("Unbekannter Code"); }
    else // Wenn der empfangene Code brauchbar ist, wird er hier an den Serial Monitor gesendet.
      {Serial.print("Empfangen: ");
        Serial.println( mySwitch.getReceivedValue() ); }
    mySwitch.resetAvailable(); // Hier wird der Empfänger "resettet"
  }
}

```

Aufgabe: Die Reichweite der Sender-Empfänger-Einheit soll getestet werden – einmal mit und einmal ohne Antenne.

Anmerkung: Die optimale Länge einer Antenne hängt von der ausgesendeten bzw. empfangenen Frequenz ab. Die optimale Länge entspricht der halben Wellenlänge der Welle. Es gilt bekanntlich: Wellenlänge mal Frequenz ergibt die

Ausbreitungsgeschwindigkeit.

Für 433 Mhz-Wellen bedeutet es eine optimale Antennenlänge von etwa 35 cm.

Sketch 2: Kommunikation zwischen Arduino Mikrocontrollern.

Wenn beim ersten Arduino ein Taster gedrückt wird, soll beim zweiten Arduino eine LED leuchten. Dazu schließen wir an Pin 12 des Empfänger-Arduinos zusätzlich eine LED an (Die Onboard-LED an Pin13 kann in diesem Fall nicht verwendet werden).

Sketch für den Sender:

```
#include <RCSwitch.h>
RCSwitch mySwitch = RCSwitch();
int taster=7;
int tasterstatus=0; //Das Wort „tasterstatus“ steht jetzt zunächst für den Wert 0. Später wird unter dieser
                    //Variable gespeichert, ob der Taster gedrückt ist oder nicht.

void setup() {
  mySwitch.enableTransmit(10); // Der Sender wird an Pin 10 angeschlossen
  pinMode(taster, INPUT);
}
void loop() {
  tasterstatus=digitalRead(taster); // „HIGH“ für 5 Volt oder „LOW“ für 0 Volt
  if (tasterstatus == HIGH)
  { mySwitch.send(5678, 24); // Der 433mhz Sender versendet die Dezimalzahl „5678“
    delay (50); }
  else
  { mySwitch.send(1234, 24); // Der 433mhz Sender versendet die Dezimalzahl „1234“ }
}
```

Sketch für den Empfänger:

```
#include <RCSwitch.h>
int LED=12;
int value=0;
RCSwitch mySwitch = RCSwitch();
void setup() {
  Serial.begin(9600);
  mySwitch.enableReceive(0); // Empfänger ist an Interrupt-Pin "0" - Das ist am UNO der Pin2
  pinMode(LED, OUTPUT); }
void loop() {
  if (mySwitch.available()) {
    value = mySwitch.getReceivedValue(); // Empfangene Daten unter "value" gespeichert.
    if (value == 0)
      { Serial.println("Unbekannter Code"); }
    else {
      Serial.print("Empfangen: ");
```

```

    Serial.println( value );
    if (value == 5678) { //Verarbeitung: Wenn der Arduino die Zahl "5678" empfängt, dann...
        digitalWrite(LED, HIGH);
        delay (500);
        digitalWrite(LED, LOW);
    }
}
mySwitch.resetAvailable();
}
}

```

Sketch 3: Messwerte per Funk übertragen

Am Arduino mit dem Sendemodul ist zusätzlich ein Temperatursensor angeschlossen, Der Messwert des Sensors soll per 433mhz an einen zweiten Arduino gesendet und dort am Serial Monitor angezeigt werden. Außerdem soll am Empfänger-Arduino eine "Warn-LED" leuchten, wenn die Temperatur 25°C oder höher ist.

Der Temperatursensor wird mit der Datenleitung an Pin A0 angeschlossen.

Sketch für Sender:

```

#include <RCSwitch.h>
RCSwitch mySwitch = RCSwitch();
int TMP36 = A0; //Der Sensor soll am analogen Pin A0 angeschlossen werden. Wir nennen den Pin ab jetzt "TMP36"
int sensorwert; // Erstellen einer Variablen für den Sensorwert.
int temperatur = 0; //Unter der Variablen "temperatur" wird später der Temperaturwert abgespeichert.

void setup() {
    mySwitch.enableTransmit(10); // Der Sender wird an Pin 10 angeschlossen
}

void loop() {
    sensorwert=analogRead(TMP36); //Auslesen des Sensorwertes.
    temperatur= map(sensorwert, 0, 410, -50, 150); //Umwandeln des Sensorwertes mit Hilfe des "map" Befehls.

    mySwitch.send(temperatur, 24); // Der 433Mhz Sender versendet die Temperatur als Dezimalzahl.
    delay (1000);
}

```

Sketch für Empfänger:

```

#include <RCSwitch.h>
int LED=12;
int value=0;
RCSwitch mySwitch = RCSwitch();

void setup() {
    Serial.begin(9600);
}

```

```
mySwitch.enableReceive(0); // Empfänger ist an Interrupt-Pin "0" - Das ist am UNO der Pin2
pinMode(LED, OUTPUT);
}
```

```
void loop() {
  if (mySwitch.available()) {
    value = mySwitch.getReceivedValue(); // empfangene Daten werden gespeichert.
    Serial.print("Temperatur: ");
    Serial.println( value );
    if (value >= 25) //Verarbeitung: Wenn die Temperatur über 25°C ist...
      { digitalWrite(LED, HIGH); }
    else
      { digitalWrite(LED, LOW); }
  }
  mySwitch.resetAvailable();
}
```