

# UKW-Radio mit dem TEA5767-Modul



## Das Programm

### Benötigte Bibliotheken einbinden und Variablen definieren

```
#include "TEA5767Radio.h"
#include "Wire.h"
#include "LCDIC2.h"

// 4-zeiliges Display
LCDIC2 lcd(0x27, 20, 4);

// String für die Eingabe im Seriellen Monitor
String Eingabe;

// Name des Radios
TEA5767Radio radio = TEA5767Radio();
/*
  Senderliste Raum Köln
  1Live 87,6
  bigFM 104.9
  Deutschlandfunk 89,1
  Radio Berg 99.7
  Radio Köln 107.1
  RPR1 103.5
  SWR1 92.4
  SWR3 94.8
  SWR4 97.4
  WDR2 98.6
  WDR3 95.9
  WDR4 90,7
  WDR5 88.0
*/

// Anzahl der Sender
#define Anzahl 5

// Variable für den gewählten Sender
int SenderNummer;

// Namen der Sender, je nach dem Empfangsgebiet anpassen
String Senderliste[Anzahl] =
{
```

```
"WDR2 ", "1Live", "WDR4 ", "Radio Berg", "SWR3"
};

// Frequenzen der Sender, Dezimalpunkt beachten
float Frequenzliste[Anzahl]
{
    98.6, 87.6, 90.7, 99.7, 94.8
};
```

## Der setup-Teil

```
void setup()
{
    Wire.begin();
    Serial.begin(9600);

    // LCD starten
    lcd.begin();

    // Cursor "verstecken"
    lcd.setCursor(false);
}
```

## Der loop-Teil

```
void loop()
{
    // Tasten des Tastenpads abfragen
    SenderNummer = Tasterabfrage();

    // -1 -> keine Taste gedrückt
    if (SenderNummer != -1)
    {
        // Namen und Frequenz des gewählten Senders anzeigen
        lcd.clear();
        radio.setFrequency(Frequenzliste[SenderNummer]);
        lcd.setCursor(0, 0);
        lcd.print("Sender: " + Senderliste[SenderNummer]);
        lcd.setCursor(0, 1);

        // . durch , ersetzen
        String Frequenz = String(Frequenzliste[SenderNummer]);
        Frequenz.replace(".", ",");
        lcd.print("Frequenz: " + Frequenz);
    }

    // auf serielle Eingabe warten
    while (Serial.available() > 0)
    {
        // Eingabe im Seriellen Monitor lesen
        String Eingabe = Serial.readStringUntil('\n');

        // letztes Zeichen entfernen
        Eingabe.substring(0, Eingabe.length() - 1);

        // Eingabe in Großbuchstaben umwandeln
        // Groß- und Kleinschreibung der Eingabe möglich
        Eingabe.toUpperCase();

        // Eingabe abfragen -> Zählung beginnt mit 0
        // Namen der Sender anpassen
    }
}
```

```
    if (Eingabe == "WDR2")
    {
        SenderNummer = 0;
    }

    if (Eingabe == "1LIVE")
    {
        SenderNummer = 1;
    }
    if (Eingabe == "WDR4")
    {
        SenderNummer = 2;
    }

    if (Eingabe == "RADIOBERG")
    {
        SenderNummer = 3;
    }

    if (Eingabe == "SWR3")
    {
        SenderNummer = 4;
    }

    // Namen und Frequenz des gewählten Senders anzeigen
    lcd.clear();
    radio.setFrequency(Frequenzliste[SenderNummer]);
    lcd.setCursor(0, 0);
    lcd.print("Sender: " + Senderliste[SenderNummer]);
    lcd.setCursor(0, 1);

    // . durch , ersetzen
    String Frequenz = String(Frequenzliste[SenderNummer]);
    Frequenz.replace(".", ",");
    lcd.print("Frequenz: " + Frequenz);
    Serial.println("Sender: " + Senderliste[SenderNummer]);
    Serial.println("Frequenz: " + Frequenz);
}

// String Eingabe leeren
Eingabe = "";
}
```

## Die Funktion Tasterabfrage

Das Tastenpad am Anschluss A0 gibt einen Wert zwischen 0 und 1023 (D1 0 bis 1024) zurück. Dieser Wert wird in der Variable Analogwert gespeichert. Je nach dem ermittelten Wert wird der Sender (Sendernummer) festgelegt. Leider verlaufen die abgefragten Werte für den Arduino UNO und den D1 Mini sehr unterschiedlich.

Beim ESP32 gibt es eine Besonderheit: der ADC-Wandler arbeitet mit einer Auflösung von 12 Bit. Die digitalen Werte befinden sich zwischen 0 und 4095.

## Arduino UNO

```
int Tasterabfrage()
{
    int Analogwert = analogRead(A0);
```

```
// kurzes delay() -> doppelten Tastendruck so weit wie möglich verhindern
delay(200);

/*
  A0 gibt je nach Taster einen Wert aus
  über den Seriellen Monitor wird dieser Wert angezeigt
  und kann dann eventuell angepasst werden
*/

// für die Anzeige des Werts Kommentarzeichen entfernen
// Serial.println(Analogwert);

switch (Analogwert)
{
  // links
  case 0 ... 20:
    SenderNummer = 0;
    break;

  // oben
  case 30 ... 60:
    SenderNummer = 1;
    break;

  // unten
  case 70 ... 120:
    SenderNummer = 2;
    break;

  // rechts
  case 150 ... 200:
    SenderNummer = 3;
    break;

  // rechts außen
  case 300 ... 400:
    SenderNummer = 4;
    break;

  // keine Taste gedrückt
  default:
    return -1;
}

// gedrückte Taste zurückgeben
return SenderNummer;
}
```

## D1 Min

```
int Tasterabfrage()
{
  int Analogwert = analogRead(A0);

  // kurzes delay() -> doppelten Tastendruck so weit wie möglich verhindern
  delay(200);

  /*
    A0 gibt je nach Taster einen Wert aus
    über den Seriellen Monitor wird dieser Wert angezeigt
    und kann dann eventuell angepasst werden
  */
}
```

```
*/
Serial.println(Analogwert);

switch (Analogwert)
{
    // links
    case 0 ... 20:
        SenderNummer = 0;
        break;

    // oben
    case 30 ... 100:
        SenderNummer = 1;
        break;

    // unten
    case 120 ... 180:
        SenderNummer = 2;
        break;

    // rechts
    case 200 ... 300:
        SenderNummer = 3;
        break;

    // rechts außen
    case 500 ... 600:
        SenderNummer = 4;
        break;

    // keine Taste gedrückt
    default:
        return -1;
}

// gedrückte Taste zurückgeben
return SenderNummer;
}
```

## ESP32-Wroom

```
int Tasterabfrage()
{
    int Analogwert = analogRead(35);

    // kurzes delay() -> doppelten Tastendruck so weit wie möglich verhindern
    delay(200);

    /*
    A0 gibt je nach Taster einen Wert aus
    über den Seriellen Monitor wird dieser Wert angezeigt
    und kann dann eventuell angepasst werden
    */
    // Serial.println(Analogwert);

    switch (Analogwert)
    {
        // links
        case 0 ... 50:
            SenderNummer = 0;
            break;
```

```
// oben
case 0 ... 200:
    SenderNummer = 1;
    break;

// unten
case 300 ... 700:
    SenderNummer = 2;
    break;

// rechts
case 900 ... 1200:
    SenderNummer = 3;
    break;

// rechts außen
case 1800 ... 2500:
    SenderNummer = 4;
    break;

// keine Taste gedrückt
default:
    return -1;
}

// gedrückte Taste zurückgeben
return SenderNummer;
}

##### E n d e #####
```