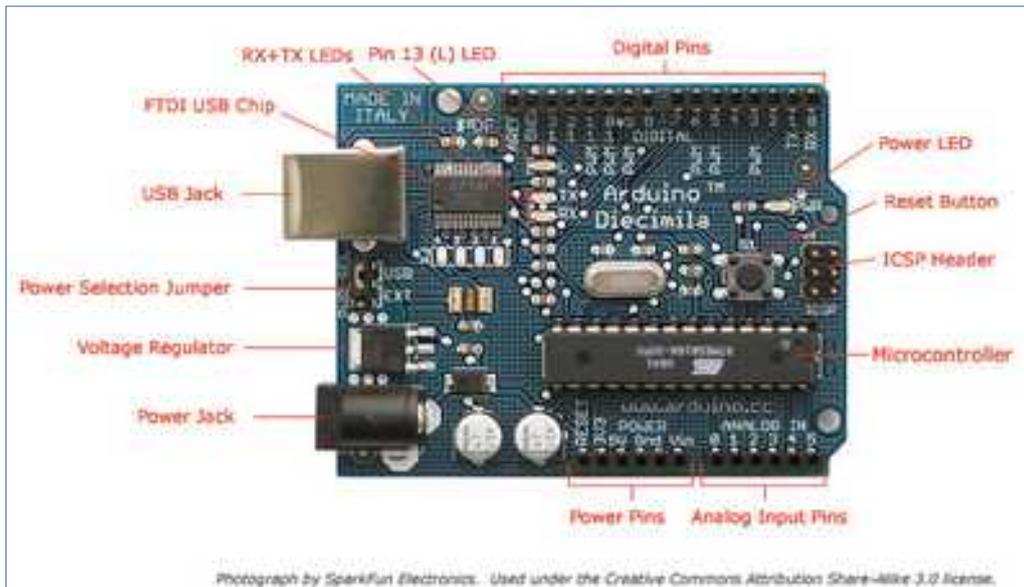


Original Arduino



26.10.22

Arduino & Co

2/21

Was ist überhaupt ein Arduino ? :
eine kleine Platine, die ca 2009 in Italien
erfunden wurde, um Bastlern das einfache
Programmieren nahezubringen.
Hier werden die Hardwarekomponenten
dargestellt.

Verschiedene Boards



26.10.22

Arduino &Co

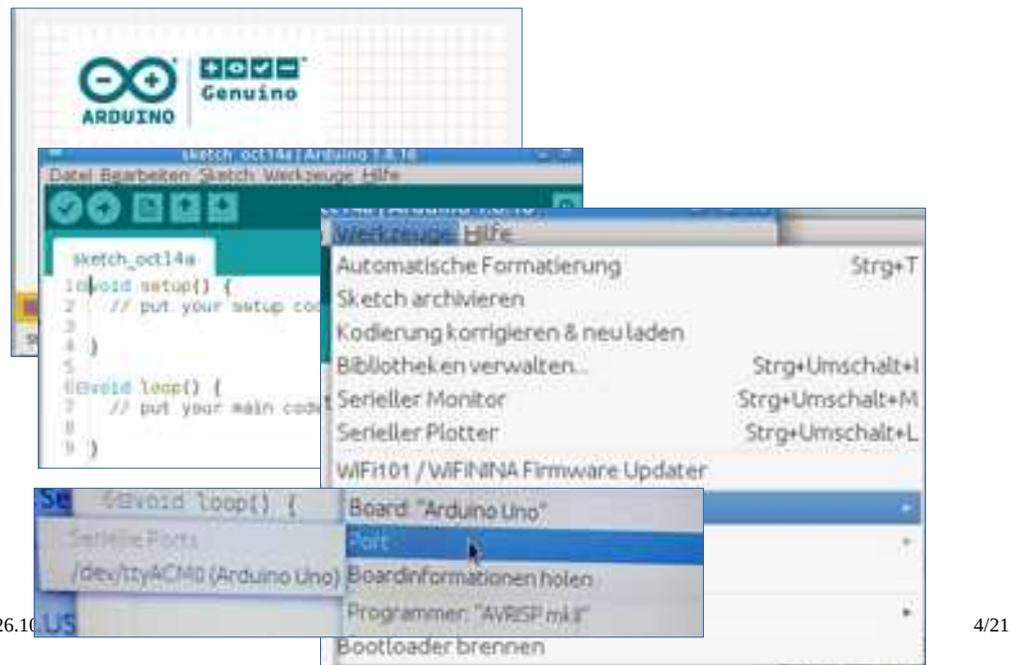
3/21

Neben dem Original-Arduino „Uno“ gibt es noch:

- Arduino Nano
- Arduino Mini ohne USB-Serial-Wandler,
- ATTiny85 8Pin
- ESPduino (ESP8266 CPU)

Diese Boards lassen sich alle mit der Arduino Oberfläche programmieren.

Arduino-Oberfläche



Die Entwicklungsoberfläche fasst alles zusammen, was man zum Programmieren braucht:

Editor
Compiler
Linker
Downloader via USB

Aktuelle Version: 2.0 (Okt. 2022)

Kritiker sagen : „Arduino-Kram“
(wahrscheinlich, weil es nicht wissenschaftlich genug ist)

Nachschlagewerk



.....	Einleitung
1.	Struktur
2.	Datentypen
3.	Arithmetik
4.	Konstanten
5.	Ablaufsteuerung
6.	Digitaler Input - Output
7.	Analoger Input - Output
8.	Zeit
9.	Mathematisch
10.	Zufallsfunktionen
11.	Serielle Kommunikation ..
12.	Anhang

<https://www.ov-meschede.de/workshop/Arduino%20Programmierhandbuch.pdf>

26.10.22

Arduino & Co

5/21

Arduino-Referenz:

<https://www.arduino.cc/reference/de/>

Arduino-Handbuch

<https://www.ov-meschede.de/workshop/Arduino%20Programmierhandbuch.pdf>

Funduino Anleitungen:

<https://fuduino.de/anleitungen>

Literatur



Hier einige Bücher.

„Shields“ , Beispiele



26.10.22

Arduino &Co

7/21

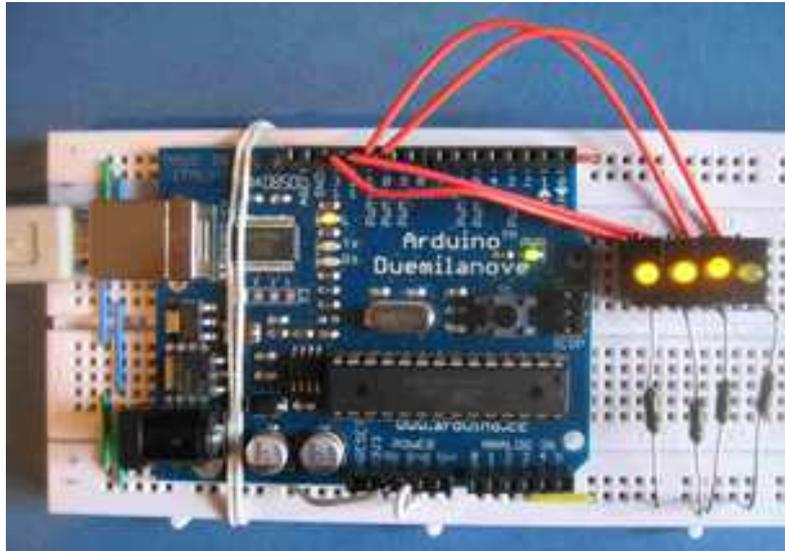
Als Zusatzhardware gibt es sogenannte Shields, die auf den Original Arduino aufgesteckt werden. Andere Boards werden mit Steckbrücken verbunden; ein Steckbrett ist sehr praktisch.

Hier:

- Relais-Modul
- GPS-Modul
- Wetter-Sensor
- LC-Display
- Uhren-Modul

Die Stromversorgung des Arduino ist 5 Volt über die USB-Schnittstelle oder extern 9 bis 12 Volt. Vorsicht bei 3,3V Zubehör, hier muss evtl. angepasst werden.

Steckbrett



26.10.22

Arduino & Co

8/21

Für die ersten Versuche eignet sich ein Steckbrett mit Steckbrücken.

Tools in der Arduino Oberfläche

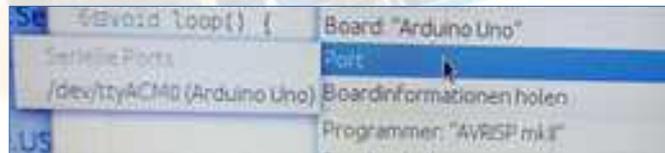
Regie: Arduino IDE aufrufen...

- Editor
- Compiler, Linker
- Downloader via USB
- Serieller Monitor (Serial.print(...);
- Plotter
- Boardverwaltung(Arduino, ESP32, Arm7...)
- Bibliotheksverwaltung (Libraries)



```
sketch_oct15a | Arduino 1.8.16
Datei Bearbeiten Sketch Werkzeuge Hilfe

sketch_oct15a
1 void setup() {
2   // put your setup code here, to run once
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly
8
9 }
```



26.10.22

Arduino & Co

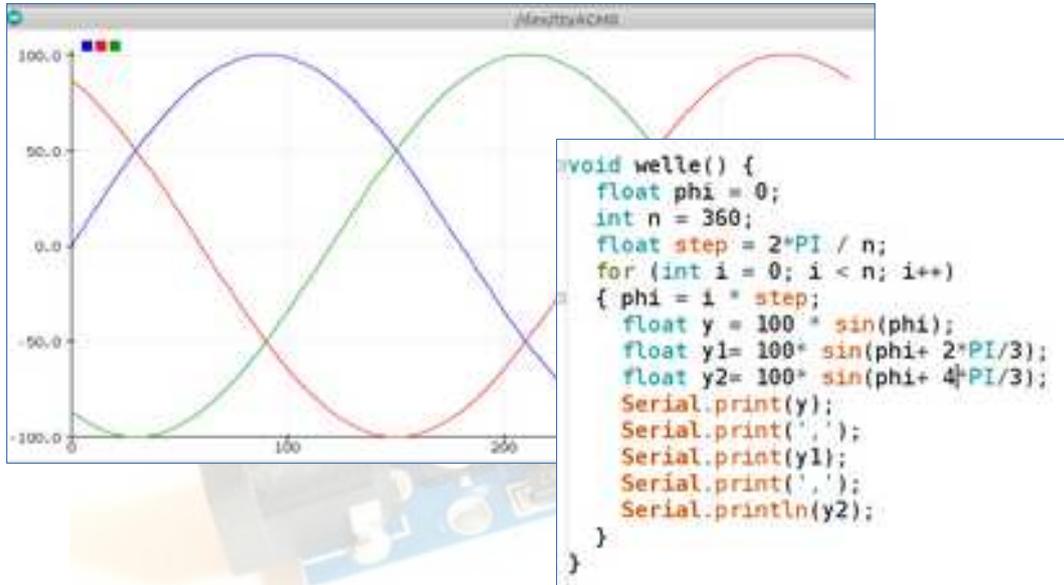
9/21

Regie: Arduino IDE aufrufen...

Die Bedienoberfläche enthält zusätzliche Tools wie z.B. serielle Schnittstelle oder Plotter für Anzeige von analogen Daten, so dass man z.B. einen Temperaturverlauf von einem Sensor als Grafik darstellen kann.

Weiterhin ist natürlich eine Hilfefunktion dabei, die den richtigen Link im Internet aufruft, wo man etwas nachschlagen kann, wenn man nicht mehr weiter weiß.

Arduino Tool:Plotter



26.10.22

Arduino &Co

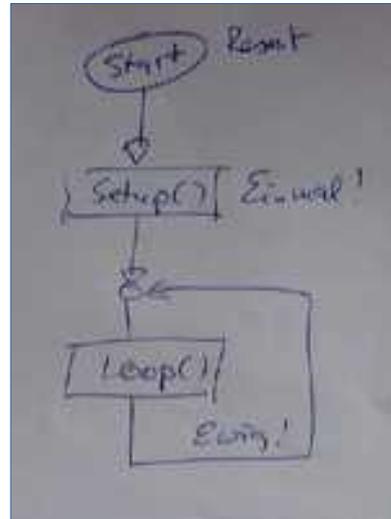
10/21

Hier ein Beispiel zur Verwendung des Plotters:

„ Drehstrom“

Hauptprogramm Vorlage

```
sketch_oct22a
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```



Programm „blink.c“



Video

Blinkende Leuchtdiode.

```
1
2  /* Blinkende LED */
3
4  void setup()           // die setup-Funktion läuft nur einmal nach dem Start
5  {
6    pinMode(13, OUTPUT); // digital pin als Ausgang konfigurieren
7  }
8
9
10 void loop()           // die loop Funktion wird ständig wiederholt
11 {
12   digitalWrite(13, HIGH); // LED einschalten (HIGH entspricht ca. 5 volt)
13   delay(1000);           // eine Sekunde ( = 1000 ms ) warten
14   digitalWrite(13, LOW); // LED ausschalten ( LOW entspricht 0 Volt )
15   delay(1000);          // eine Sekunde ( = 1000 ms ) warten
16 }
```

26.10.22

Arduino &Co

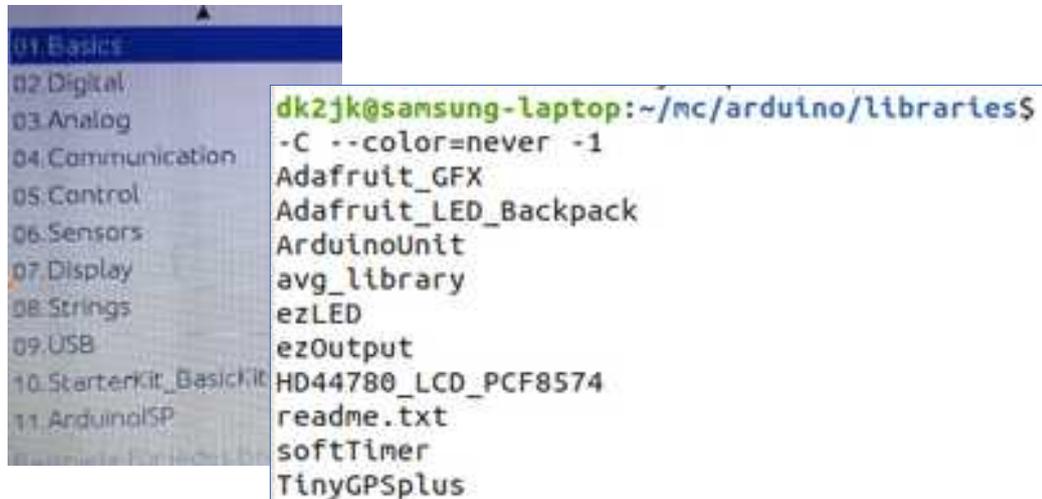
12/21

Beispiel: Blinkende Leuchtdiode.
Das Programm ist durch Kommentare erklärt.

Siehe Video / Steckplatine

Original unter :
Datei -- Beispiele – 01. basics -- blink

Beispiele, Libraries



```
01. Basics
02. Digital
03. Analog
04. Communication
05. Control
06. Sensors
07. Display
08. Strings
09. USB
10. StarterKit_Basickit
11. ArduinoISP

Adafruit_GFX
Adafruit_LED_Backpack
ArduinoUnit
avg_library
ezLED
ezOutput
HD44780_LCD_PCF8574
readme.txt
softTimer
TinyGPSplus
```

26.10.22

Arduino & Co

13/21

Man braucht für zusätzliche Hardware z.B für Relais, Displays und ähnliches sog. Libraries.

Die Libraries dienen dazu, dass nicht jeder „das Rad wieder von vorne erfinden“ muss .

Es gibt eingebaute Libraries und welche, die von der Arduino Seite heruntergeladen werden können; dort sind auch Beschreibungen, wie die Libraries zu bedienen sind.

Uhren



26.10.22

Arduino & Co

14/21

Eines meiner Lieblingsthemen sind Uhren; hier einige Beispiele.

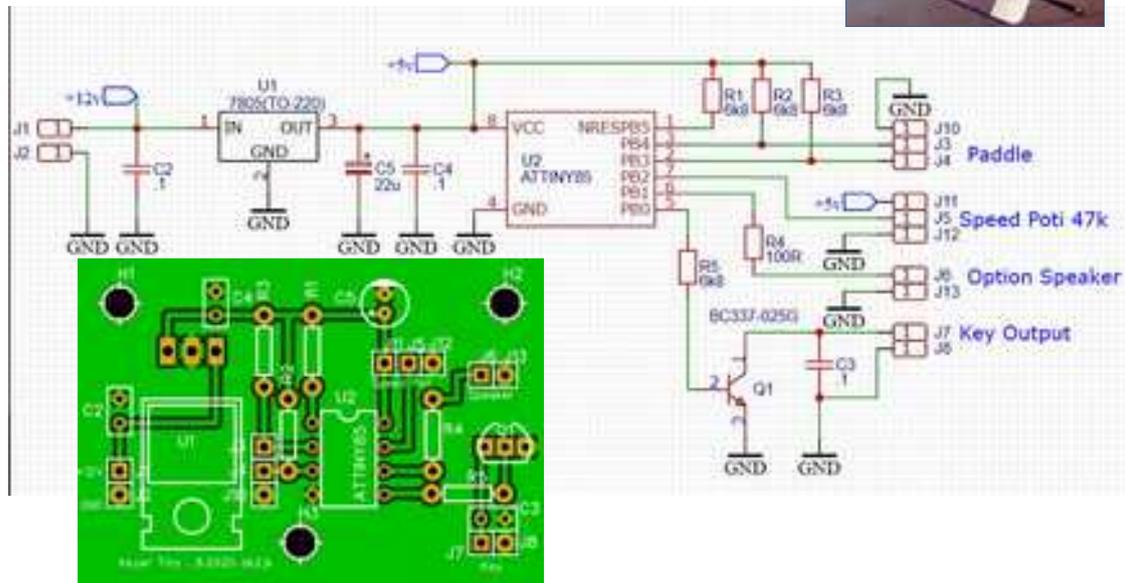
Led-Matrix Uhr

GPS Uhr/ Tacho

Uhr im Rahmen, Netz-Zeit über Wifi

Uhr mit alten Tankstellen-Displays (Raspi)

Keyer= Elektronische Taste



26.10.22

Arduino & Co

15/21

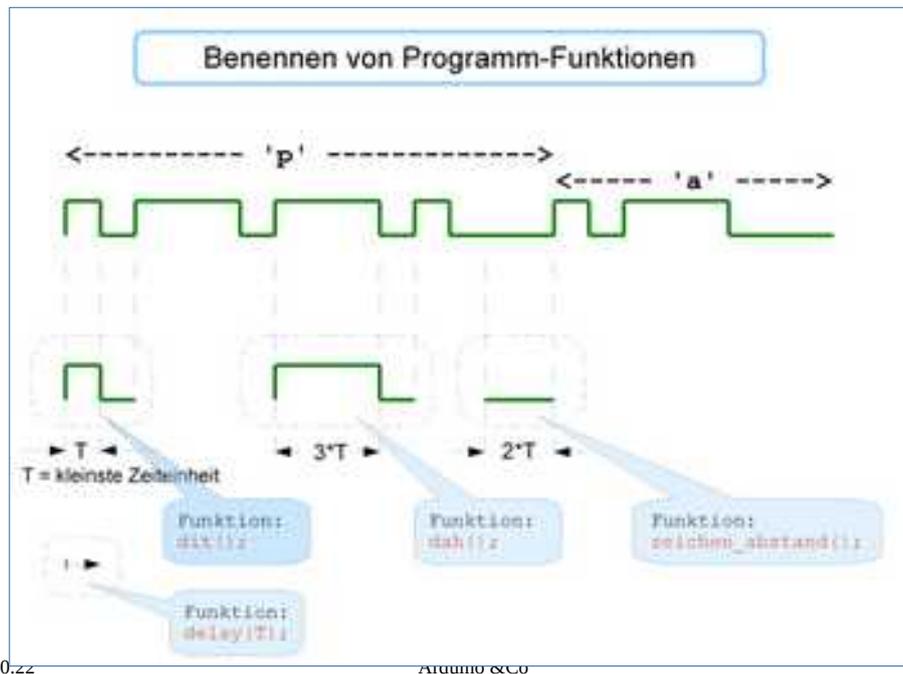
Eine Anwendung im Amateurfunk:

Keyer, eine elektronische Taste für 2-armigen Geber. Die Logik lässt sich in einem 8-beinigen Chip unterbringen.

Test erfolgte mit dem Original-Arduino.

Zur Programmierung braucht man einen Programmieradapter, der sehr einfach ist, denn ein anderer Arduino kann als Programmiergerät dienen. dazu müssen nur sechs Leitungen zu dem zu programmierenden Chip gelegt werden.

Morsezeichen-Geber



Eine CW-Bake sendet in Zeitabständen Text in Morsezeichen. Hier wird gezeigt, wie die Morsezeichen aufgebaut sind am Beispiel der Zeichenfolge ,pa'. Dazu eine kleine Grafik; sie zeigt, wie man so eine Programmieraufgabe angeht. Die Aufgabe wird in Funktionen unterteilt („Unterprogramme“)

Programm Morsezeichen-Geber

```
//Variablen:  
int pin_ton = 4;  
int beitz = 650;  
#define WPM 12  
int t_dit = 1200 / WPM;  
  
// Initialisierung:  
void setup()  
{  
}
```

```
//Hauptprogramm  
void loop()  
{  
  buchstaben_abstand();  
  dit(); //  
  dit();  
  dit();  
  buchstaben_abstand();  
  
  dah(); //  
  dah();  
  dah();  
  dah();  
  buchstaben_abstand();  
  wort_abstand();  
  delay(1000);  
}
```

Arduino-Programm

```
void dit()  
{  
  tone(pin_ton,beitz);  
  delay(t_dit);  
  noTone(pin_ton);  
  delay(t_dit);  
}
```

```
void dah()  
{  
  tone(pin_ton,beitz);  
  delay(t_dit*3);  
  noTone(pin_ton);  
  delay(t_dit);  
}
```

```
void buchstaben_abstand()  
{  
  delay(t_dit*2);  
}
```

```
void wort_abstand()  
{  
  delay(t_dit*4);  
}
```

Komplettes Bakenprogramm
•hier Beispiel nur mit Tonausgabe
•Hauptprogramm verkürzt
•Wort-Abstand hier 5 Zeiteinheiten

26.10.22

17/21

Hier die Unterprogramme als Text in der Programmiersprache ,C‘

Projekte...

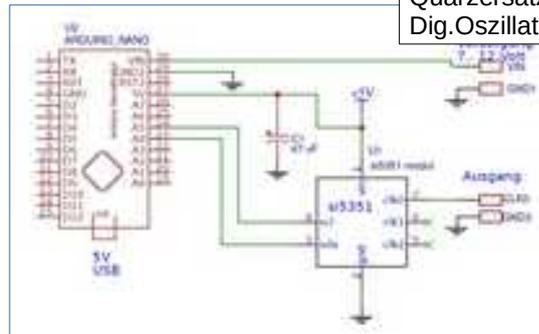


GPS-Uhr /
Tacho



Stratosphärenballon:
2m Sender

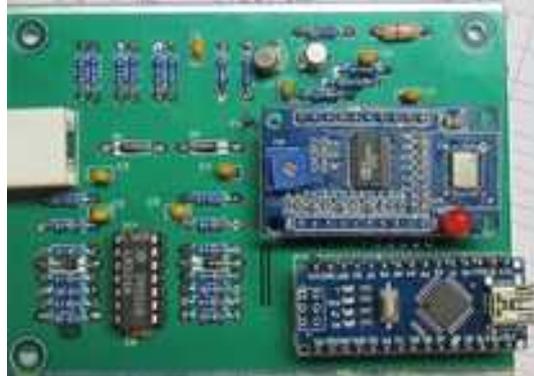
Frequenz-Display
,Bitx'



Quarzersatz=
Dig.Oszillator

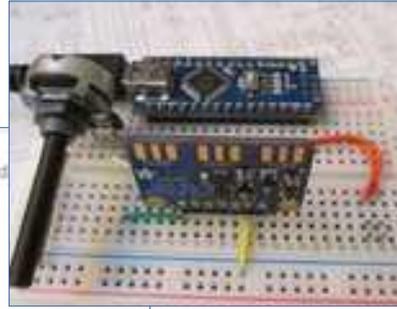
- GPS-Uhr/ Tacho
- Stratosphärenballon 2m Sender
- LCD Frequenz-Display Bitx
- Quarzersatz

Projekt Antennenanalyser



Haben wir mal als Gruppenprojekt im OV
Meschede gebaut.

Projekt Digitaler VFO



```
1 // https://github.com/etherkit/Si5351Arduino
2 // http://mt7s.com/2014/06/si5351-libraries-and-breakout-board
3
4 #include "si5351.h"
5 #define PIN_POTI A0
6 Si5351 si5351;
7
8 void setup()
9 { si5351.init(SI5351_CRYSTAL_LOAD_10PF);
10   si5351.set_pll(SI5351_PLL_FIXED, SI5351_PLLA); // 900 MHz
11   si5351.clock_enable(SI5351_CLK0, 1); // CLK0 freigeben
12 }
13 void loop()
14 { //frequenz vom Poti ablesen:
15   long frequenz = map(analogRead(PIN_POTI), 0, 1023, 7000000, 7040000);
16   //frequenz neu einstellen:
17   si5351.set_freq(frequenz, SI5351_PLL_FIXED, SI5351_CLK0);
18 }
19
20
```

Experiment „dig. VFO“

Es fehlt ein Tiefpass.