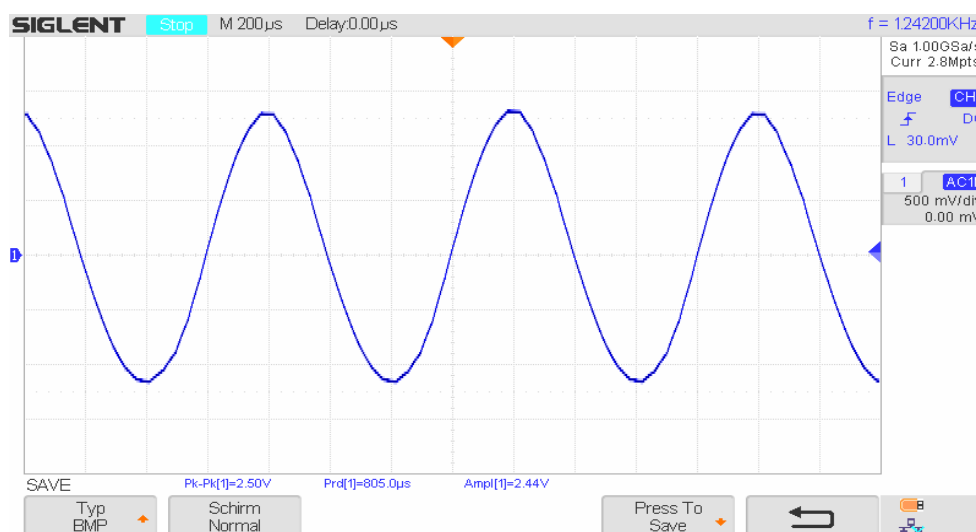


Arduino: Nur ein Spielzeug?

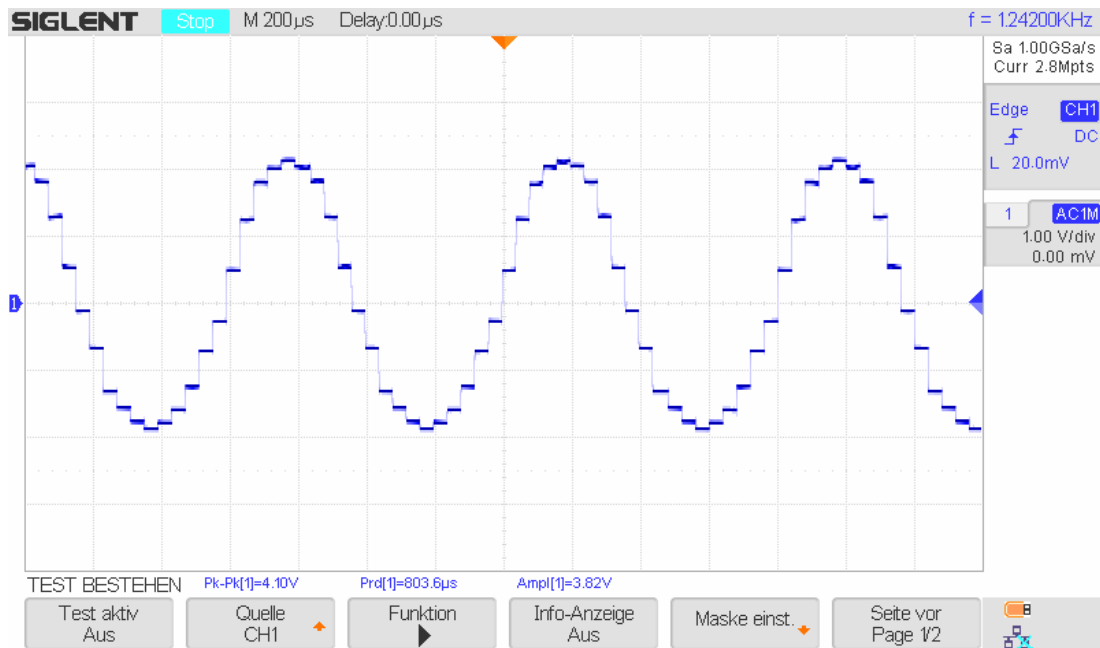
Eigentlich brauche ich schon seit langer Zeit einen Wobbler für den NF-Frequenzbereich. Natürlich gibt es da leistungsfähige Anwendungen für den WIN-PC. Sie kosten auch nicht viel. Aber diese Software braucht den „großen“ Bildschirm. Und dafür ist auf meinem Basteltisch kein Platz vorgesehen. Lötten und messen am selben Ort wäre ja ideal, aber verlangt eine großzügige Umgebung. Die habe ich nicht. Und so träume ich immer noch von einem handlichen Gerät mit Bildschirm, das ich mit einem 9V-Block betreiben kann.

Der Arduino kam ins Blickfeld meiner Vorstellungen. Und ich versuchte es zunächst mal mit dem Nano. Primär ging es um die Erzeugung eines Signals mit Sinusform. Also programmierte ich den Port mit digitalen Ausgabewerten, die dann mit einem Widerstands-Netzwerk R/2R nach Sinus umgeformt werden. Nicht ganz einfach, wie ich erleben konnte. Erst nach einigen Versuchen gelang es mir, einen einigermaßen brauchbaren Sinus zu erzeugen. Die nächsten Bilder zeigen Signale, die ich von 10 Hz bis 47 kHz erzeugen konnte.

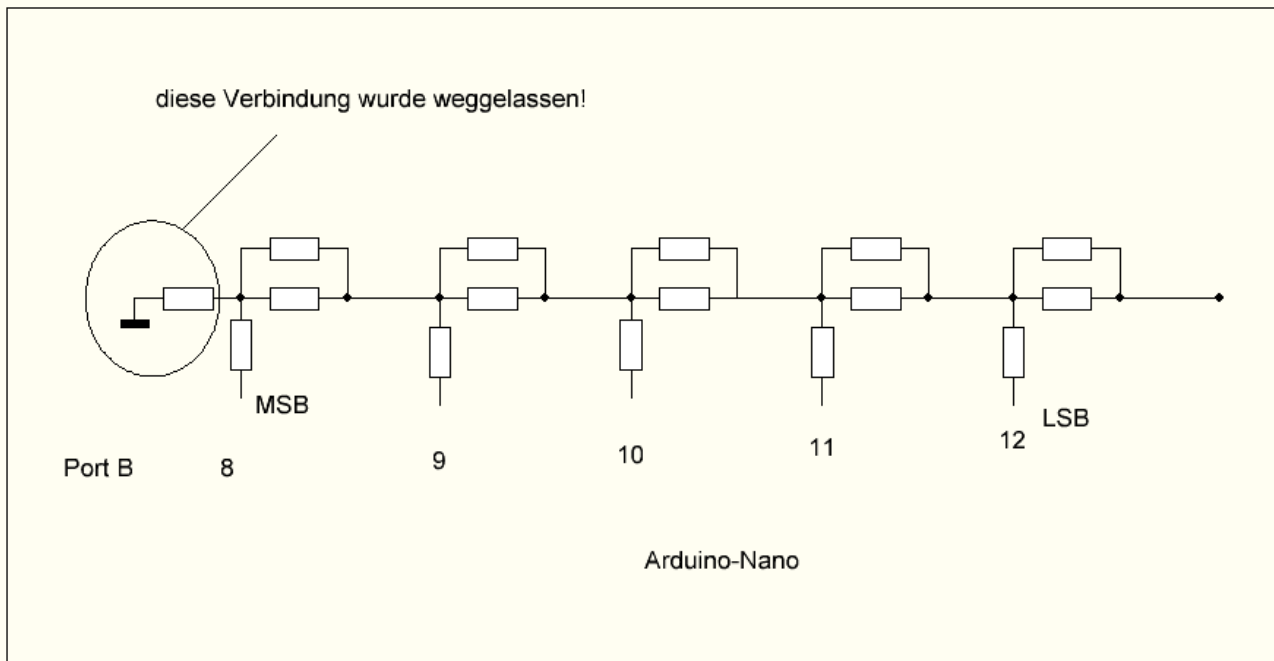


Das Foto zeigt ein Signal bei ca. 1 kHz. Es war nach dem Netzwerk ein Glättungskondensator angeschaltet. Im nächsten Bild wurde der Kondensator weggelassen. Man sieht den

treppenförmigen Aufbau der Ausgangsspannung.

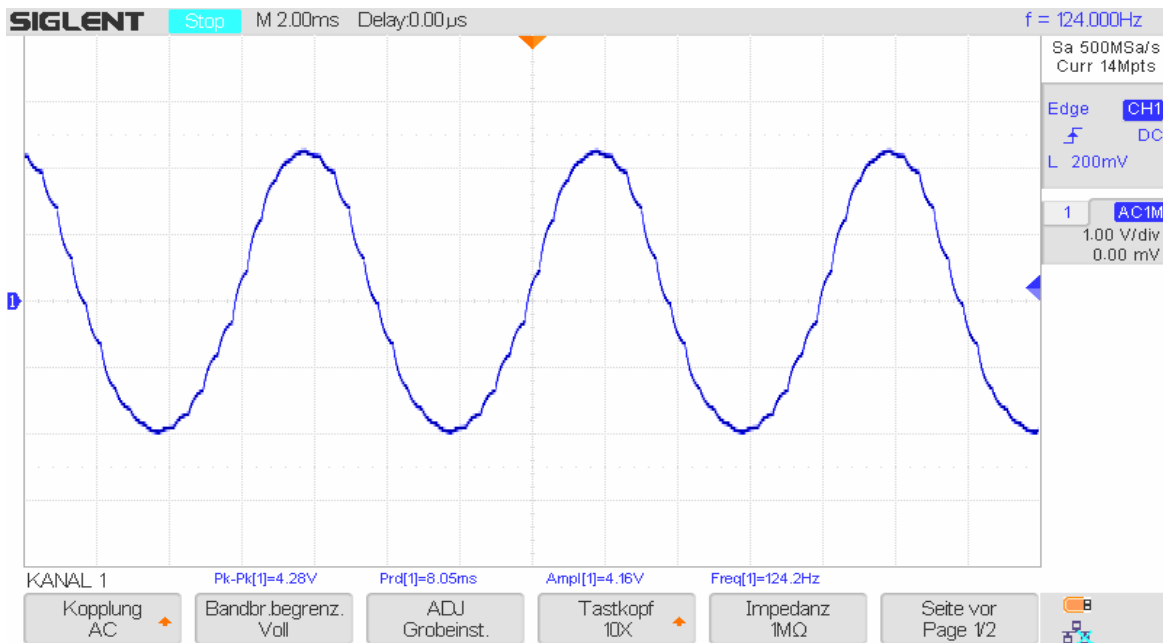


Die Umsetzung ist immer ein Kompromiss. Ich habe 2k2-Widerstände(Kohleschicht, 5%) verwendet. Der Aufbau war nicht ganz korrekt, aber nur so erreichte ich einen einigermaßen brauchbaren Signalverlauf.

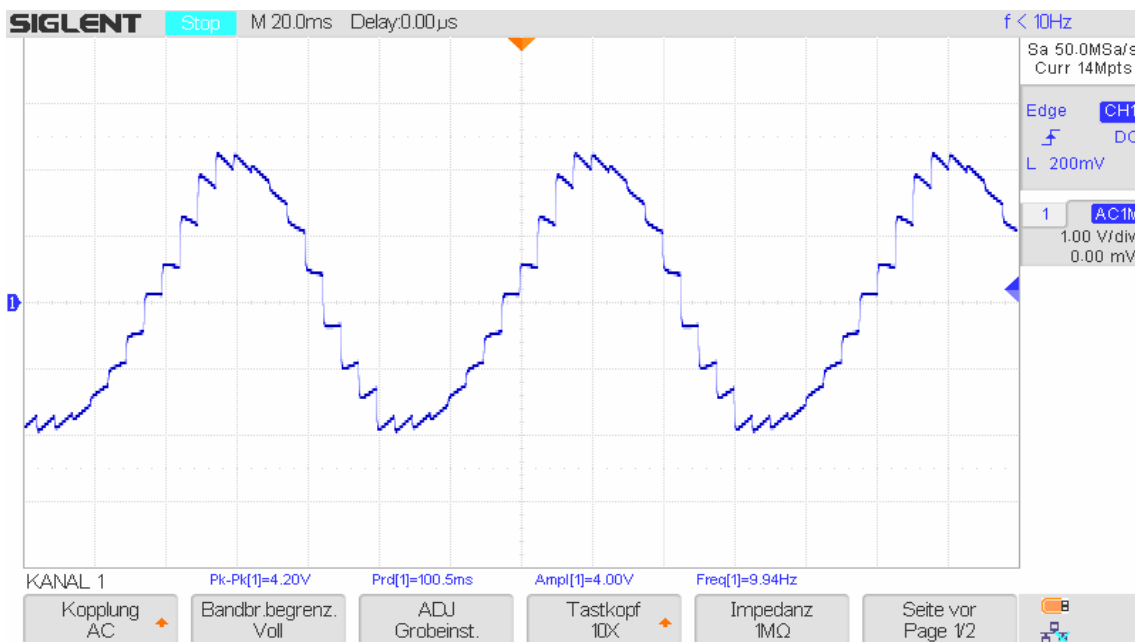


Bei niedriger Frequenz muss man den Kondensator(Tiefpass) vergrößern. Umgekehrt kann man ihn verkleinern, wenn man eine

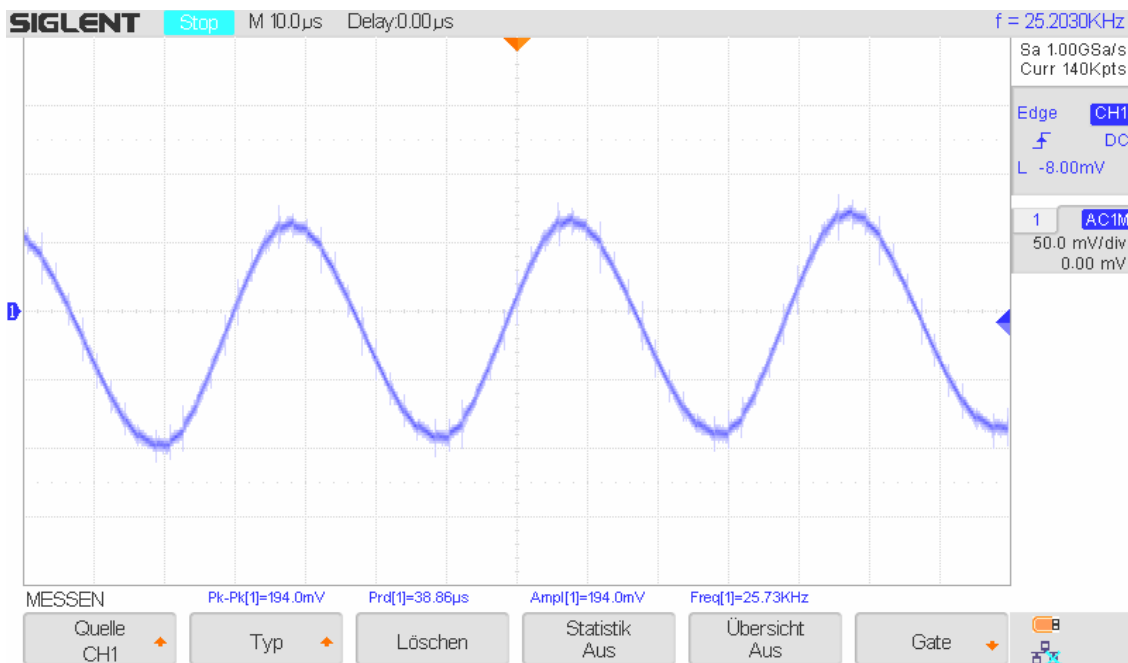
zu starken Pegelabfall vermeiden will.



124 Hz (Steptime 400 us)



< 10 Hz (Steptime 5000us). Der Kondensator ist zu klein. Allerdings wäre das Signal zum Wobbeln durchaus geeignet. Es kommt da ja nur bei höchsten Anforderungen an die Genauigkeit darauf an. Den Klirrfaktor gering zu halten. Bemerkenswert ist der konstante Pegel im unteren Frequenzbereich.



25 kHz (Steptime 2 us). Man beachten den niedrigen Pegel.

Hier könnte man den Glättungskondensator mit einem niedrigen Wert anschalten.

Bewertung

Man erkennt, dass es auf preiswerte Art möglich ist, mit dem Arduino(16MHz) den Frequenzbereich von 10 Hz bis 20 kHz zu befriedigen.

Allerdings muss man den Pegel regeln, damit man einen konstanten Spannungsverlauf hat. Hierzu sind noch einige Probleme zu lösen. In allen Bilder hatte der Glättungskondensator denselben Wert(15n). Er bestimmt die Welligkeit und den Pegel.

Mit den Netzwerkwiderständen 2k2 werden der digitalen Ausgänge noch nicht überlastet. Mit höheren Widerständen holt man sich Störgeräusche rein und der Pegel ist noch welliger. Mein „fliegender“ Aufbau auf den üblichen Steckbrettern war nicht optimal.

Hier einige Ausschnitte aus der Software:

```
//Funktion für den Oszillator
```

```
void dac(int d) { //Bildung der Stepzeit für die  
//Frequenzberechnung
```

```
//Serial.println("DAC läuft");//TEST
```

```
do {
```

```
//Serial.println("DAC läuft");//TEST
```

```
    delayMicroseconds(d);//          0  
    akt= 45;  
    PORTB = akt;
```

```
    delayMicroseconds(d);//          1  
    akt= 50;  
    PORTB = akt;
```

```
    delayMicroseconds(d);//          2  
    akt= 55;  
    PORTB = akt;
```

```
    delayMicroseconds(d);//          3  
    akt= 59;  
    PORTB = akt;
```

```
    delayMicroseconds(d);//          4  
    akt= 61;  
    PORTB = akt;
```

```
    delayMicroseconds(d);//          5
```

```
akt= 62;  
PORTB = akt;
```

```
//----- 5 steps
```

```
delayMicroseconds(d);//      6  
akt= 61;  
PORTB = akt;
```

```
delayMicroseconds(d);//      7  
akt= 59;  
PORTB = akt;
```

```
delayMicroseconds(d);//      8  
akt= 55;  
PORTB = akt;
```

```
delayMicroseconds(d);//      9  
akt= 50;  
PORTB = akt;
```

```
delayMicroseconds(d);//     10  
akt= 46;  
PORTB = akt;
```

```
//-----bis hier 10 steps
```

```
delayMicroseconds(d);//     11  
akt= 10;  
PORTB = akt;
```

```
delayMicroseconds(d);//      12
akt= 5;
PORTB = akt;
```

```
delayMicroseconds(d);//      13
akt= 3;//5;
PORTB = akt;
```

```
delayMicroseconds(d);//      14
akt= 2;//3
PORTB = akt;
```

```
delayMicroseconds(d);//      15
akt= 1;//
PORTB = akt;
```

//----- nach 15 steps

```
delayMicroseconds(d);//      16
akt= 2;//
PORTB = akt;
```

```
delayMicroseconds(d);//      17
akt= 3;//
PORTB = akt;
```

```
delayMicroseconds(d);//      18
akt= 6;//
PORTB = akt;
```

```
// delayMicroseconds(d);//      19
// akt= 8;//
// PORTB = akt;
```

```

delayMicroseconds(d);//          20
akt= 10;//
PORTB = akt;
//-----ENDE der
D/A-//Wandlung

} while (oszilliere == 1); // Wiederhole das ganze, falls
//oszilliere ausgelöst wurde

//sonst Rücksprung in die Loop

}

//;

```

Ich wende hier die schnelle Portausgabe an. Und das Einschalten des „Oszillators“ wird mit einer Taste über den Interrupteingang 1 gemacht.

Der Sinus wird in 20 Steps erzeugt.

```

//interruptroutine:

void osz(){

    if (oszilliere == 0){

        oszilliere = 1;
    }
else {
    oszilliere = 0;
}
}

```



```
//Serial.print("oszilliere = ");//TEST
//Serial.println(oszilliere);//TEST
//delay(1000);

}
```

Fazit

Der Aufwand ist nicht gering. Fast alle Portpins des NANO waren belegt, weil ich für das Setzen der Anfangsfrequenz und der Endfrequenz, des Starts und Stopps und des Löschsens der Eingaben Tasten brauchte. Wenn man dann noch eine Ausgabe der Wobbelkurve auf einem Display machen will, reicht ein NANO nicht aus.

Der NANO ist hinreichend für einen Funktionsgenerator. Dafür gibt es aber bessere Software im Netz.

Meine nächste Idee ist, einen Wobbler mit dem Einsatz des AD9833 zu entwickeln. Hier ist der Sinus bis 1MHz im Pegel konstant. Bis 20 kHz kann man das fast garantieren. Außerdem ist der Sinusverlauf sehr gut. Der Baustein ist bis 12 MHz spezifiziert. Ich brauche ihn aber nur ganz unten. Natürlich werden auch hier Tasten für die vielen Funktionen benötigt. Vielleicht setze ich ein Touchdisplay ein. So ein 3,5 Zoll könnte für eine Anzeige ausreichend sein. Und wenn es sein muss, sind zwei NANOS auch nicht wesentlich kostensteigernd. Der AD9833 ist in China für 2,7 EUR/Platine zu haben, der NANO für 2,5 EUR.

DF8ZR; im April 2021