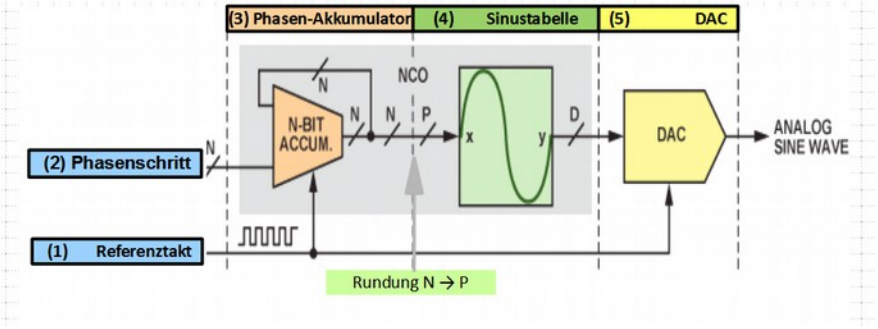
Arduino Sinus-Generator
nach der Direkten Digitalen Synthese (DDS)

Um die DDS- Methode in Software zu implementieren, werden folgende Komponenten benötigt:

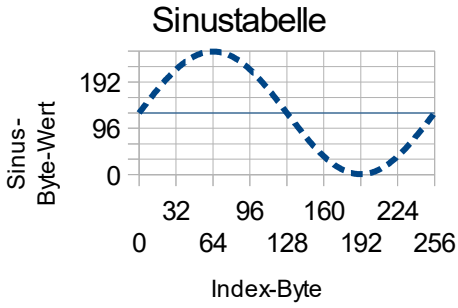
1. Referenztakt
2. Phasenschritt
3. Phasen-Akkumulator
4. Sinustabelle
5. Digital-Analog-Wandler (DAC)



Prinzip de DDS

Sinustabelle

Die Sinustabelle besteht aus 256 Bytes mit den Werten einer Sinusperiode mit sinus(0) =128 , sinus(π/2) = 255 und sinus(3*π/2) = 0. Da der Ausgang der PWM keine negativen Wert erzeugen kann, erhält der Sinus einen Offset von 128 , entsprechend einem PWM- Ausgang von 2,5 Volt.

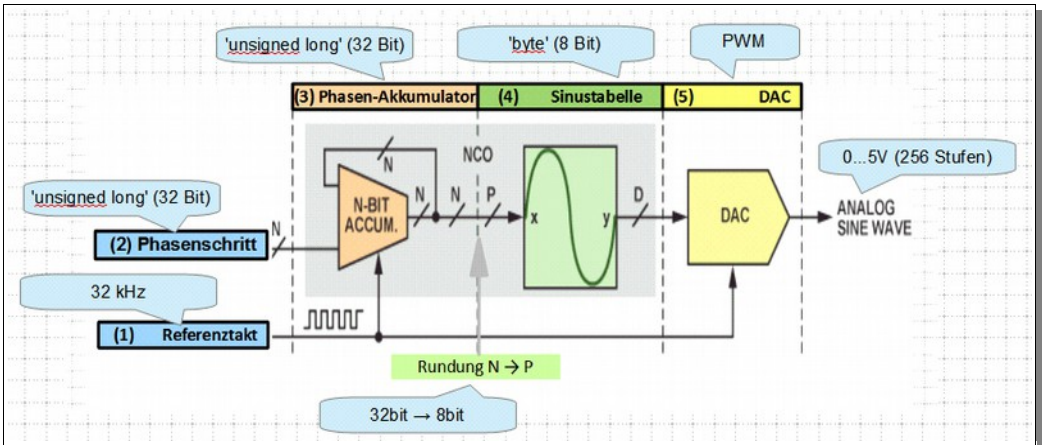


Referenztakt

Als Referenztakt wird 32 kHz gewählt; diese Frequenz wird beim 16 Mhz- CPU-Takt durch Teilung 16000000/510 = 31372.55 Hz erzeugt.

Realisierung der DDS

Im Phasen-Akkumulator wird bei jedem Takt ein Phasenschritt-Wert aufaddiert. Die Phase wird dargestellt durch eine Zahl von 0 … 2**32-1 (32 Bit). Das entspricht einem 'unsigned long' – Datentyp.



Praktische Ausführung der DDS mit Arduino

Wertezuordnung:

Phasen-Akkumulator	Index P	Grad	Winkel	sin	Ausgang D
0x00000000	0	0	0	0	127
0x40000000	64	90	0.5*pi	1	255
0x80000000	128	180	pi	0	127
0xc0000000	192	270	1.5 *pi	-1	0
0xFFFFFFFF	255	<360	<2*pi	<0	<127
0x00000000	0	360	2*pi	0	127

Die obersten 8 Bits (grau markiert) können direkt als Index in der Sinus-Tabelle dienen(0x00 = 1. Wert in der Tabelle , 0xFF= letzter Wert in der Tabelle). Diese Rundung betrifft nur die Sinustabelle, der Phasen-Akkumulator behält seine Auflösung von 32 Bit.
Beim Überlauf des Phasen-Akkumulator von 0xFFFFFFFF auf 0x00000000 ergibt sich automatisch ein Sprung zum Beginn der Sinustabelle.
Mit der Größe des Phasenschritts wird die Frequenz eingestellt: kleine Werte ergeben niedrige Frequenz , große Werte hohe Frequenz.

Es gilt folgende Formel:

f = (Phasenschritt * Taktfrequenz) / 2n**
n=32 (bits)
Taktfrequenz = 31372.55 Hz

f = (Phasenschritt* 31372.55) / 232**

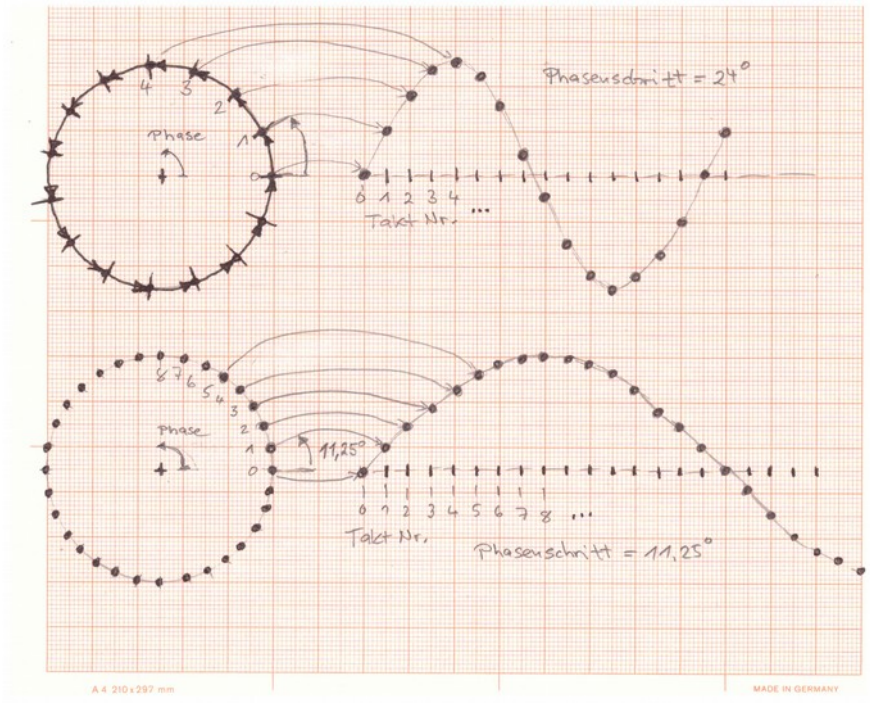
Aufgelöst nach Phasenschritt:

Phasenschritt= 232* f/31372.55)**

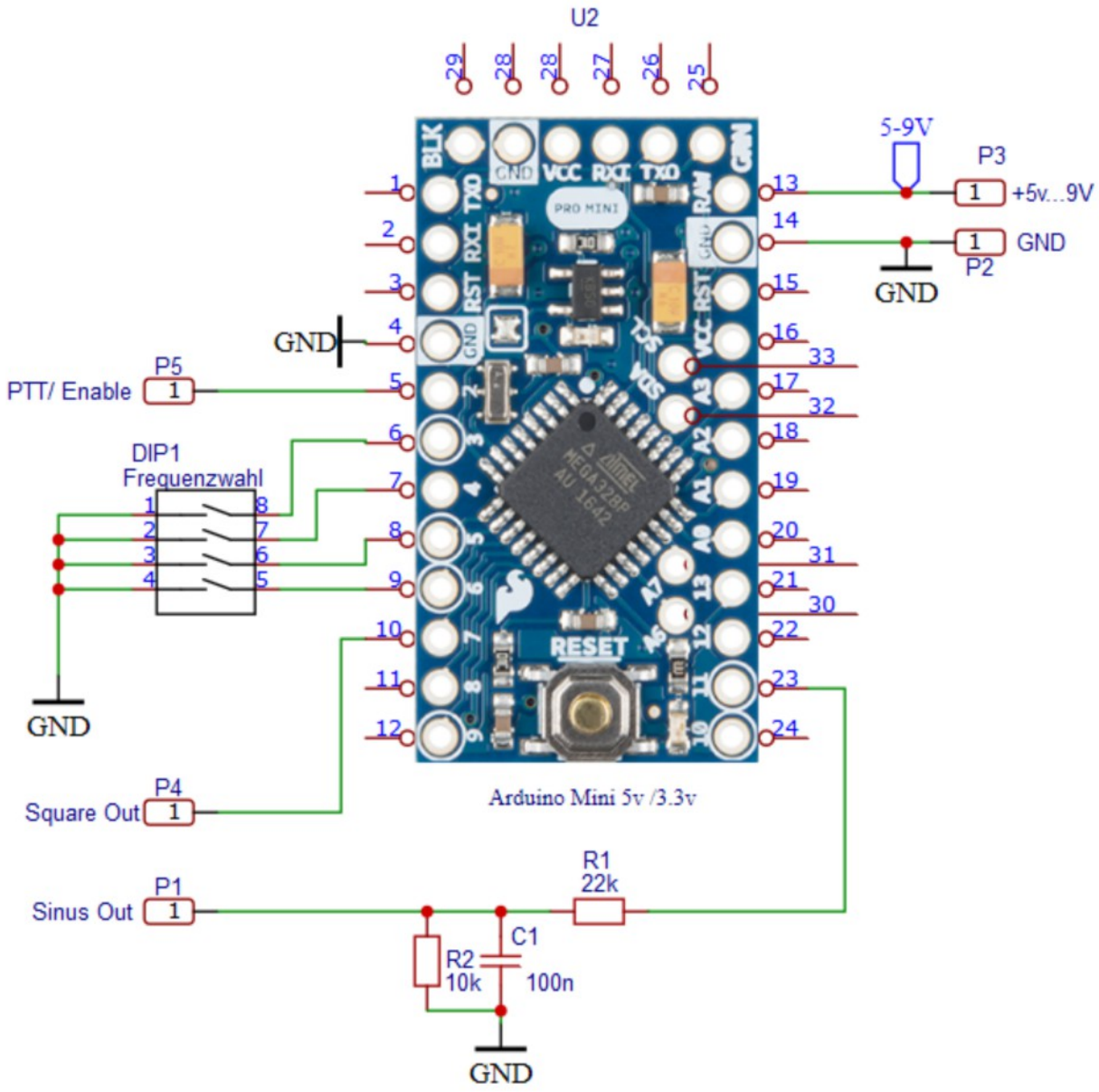
'Digital Phase Wheel'

Das 'Digital Phase Wheel' demonstriert die Generierung eines Sinus per DDS:

Digital Phase Wheel



Schaltung



frequenzTabelle[16] =

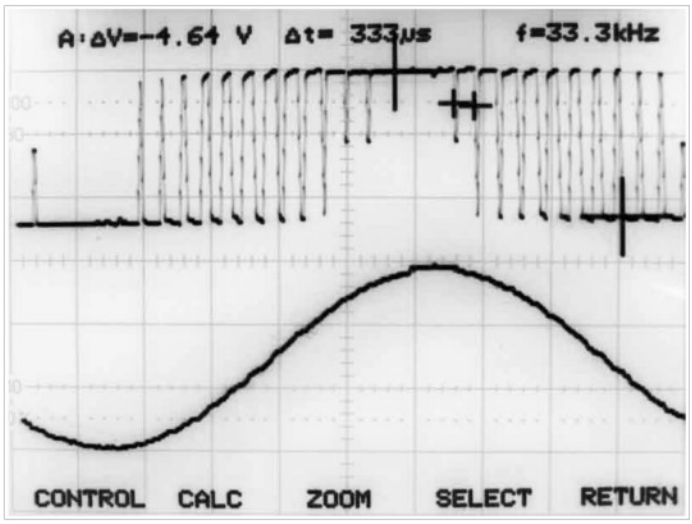
```
///*6543*/  Brücken an Port 6543 gegen GND
{ /*0000*/ 67.0  ,/*0001*/ 71.9  ,/*0010*/ 74.4  ,/*0011*/ 85.4  ,
  /*0100*/ 88.5  ,/*0101*/ 91.5  ,/*0110*/ 94.8  ,/*0111*/ 100.0 ,
  /*1000*/ 103.5 ,/*1001*/ 110.9 ,/*1010*/ 114.8 ,/*1011*/ 123.0 ,
  /*1100*/ 127.3 ,/*1101*/ 131.8 ,/*1110*/ 136.5 ,/*1111*/ 141.3
};

Index wird über Port D[6:3] eingegeben( Brücken )
z.b.: 88.5Hz => Brücke D5 auf GND.
```

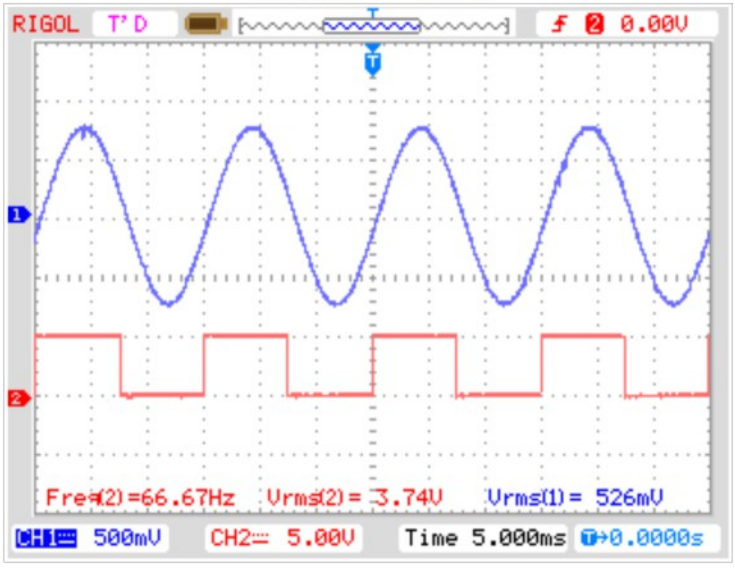
Digital-Analog-Wandler

Die Digital-Analog-Wandlung wird durch eine Pulsweitenmodulation(PWM) gebildet, deren Pulsweite alle 32 us per Interrupt aktualisiert wird. Durch nachgeschalteten Tiefpass (z.B. R-C-Glied) wird daraus ein Analogwert.
Man benötigt ein Tiefpassfilter, um die 32 kHz Abtastfrequenz im Ausgangssignal zu entfernen.

PWM / Analogsignal nach Tiefpass



Analogsignal / Rechteck

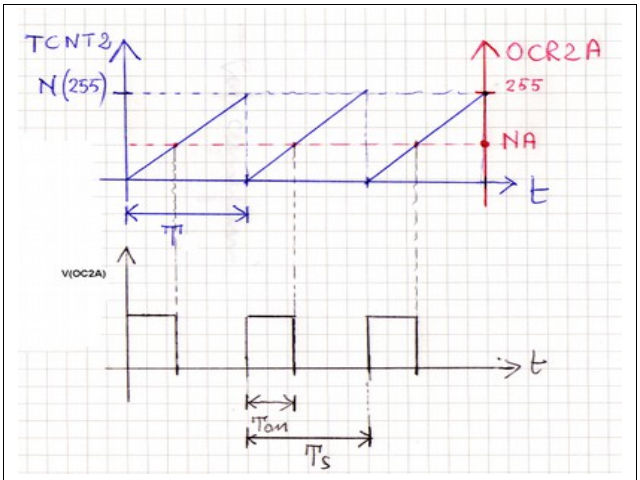


Programm

```
//*********************************************************************
// Timer2 Interrupt Service at 31372,550 KHz = 32uSec
// this is the timebase REFCLOCK for the DDS generator
// FOUT = (M (REFCLK)) / (2 exp 32)
ISR(TIMER2_OVF_vect) {

  // for SIN1
  phaccu_a = phaccu_a + tword_a; // soft DDS, phase accu with 32 bits
  icnt_a = phaccu_a >> 24;   // use upper 8 bits for phase accu as frequency information
  // read value fron ROM sine table and send to PWM DAC
  OCR2A = pgm_read_byte_near(sine256 + icnt_a);
```

Programmausschnitt Timer2- Interrupt



Verhalten des PWM-Timer2 Arduino

[http://telpar.altervista.org/]