



Hochschule Konstanz
Technik, Wirtschaft und Gestaltung

Signale, Systeme und Sensoren

DIGITALISIERUNG

Kiattipoom Pensuwan, Thanh Son Dang

Konstanz, 16. Januar 2019

Zusammenfassung

Thema:	DIGITALISIERUNG	
Autoren:	Kiattipoom Pensuwan	ki851pen@htwg-konstanz.de
	Thanh Son Dang	th851dan@htwg-konstanz.de
Betreuer:	Prof. Dr. Matthias O. Franz	mfranz@htwg-konstanz.de
	Jürgen Keppler	juergen.keppler@htwg-konstanz.de
	Simon Christofzik	si241chr@htwg-konstanz.de

In diesem Versuch wird die Umwandlung von analogen in digitale Daten untersucht. Es wird die Eigenschaften wie Genauigkeit der AD - bzw. DA-Wandlung (der theoretische Quantisierungsfehler) , Zeitverhalten und Verhalten zwischen Abtastfrequenz und die von dem Generator erzeugte Frequenz (Abtasttheorem) untersucht und analysiert, indem 3 verschiedene Messgeräte benötigt werden. Um das Experiment zu realisieren, sollte ein Python-Programm mithilfe der Programmbibliothek RedLab-Python geschrieben und angewendet.

Inhaltsverzeichnis

Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
1 Programmierung der AD/DA-Wandlerkarte	1
2 Genauigkeit der AD-Wandlung	2
2.1 Fragestellung, Messprinzip, Aufbau, Messmittel	2
2.2 Messwerte	3
2.3 Auswertung	3
2.4 Interpretation	4
3 Genauigkeit der DA-Wandlung	5
3.1 Fragestellung, Messprinzip, Aufbau, Messmittel	5
3.2 Messwerte	5
3.3 Auswertung	6
3.4 Interpretation	6
4 Zeitverhalten der DA-Wandlung	7
4.1 Fragestellung, Messprinzip, Aufbau, Messmittel	7
4.2 Messwerte	8
4.3 Auswertung	9
4.4 Interpretation	9
5 Abtasttheorem	10
5.1 Fragestellung, Messprinzip, Aufbau, Messmittel	10
5.2 Messwerte	11
5.3 Auswertung	14

5.4	Interpretation	14
Anhang		15
A.1	Quellcode	15
A.1.1	Quellcode Versuch 1-3	15
A.1.2	Quellcode Sinusschwingung-Generator Versuch4	15
A.1.3	Quellcode Versuch 5	16

Abbildungsverzeichnis

4.1	Erzeugte Sinusschwingung durch DA-Wandlung	8
4.2	Abgeschnittene und vergrößerte Sinusschwingung	8
4.3	Erzeugte Sinusschwingung ohne Pause	9
5.1	Schwingung mit der Frequenz 10Hz	11
5.2	Schwingung mit der Frequenz 2000Hz	11
5.3	Schwingung mit der Frequenz 2750Hz	12
5.4	Schwingung mit der Nyquist-Frequenz 4000Hz	12
5.5	Schwingung mit der Frequenz 5500Hz	13
5.6	Schwingung mit der Frequenz 6000Hz	13
5.7	Schwingung mit der Abtastfrequenz 8000Hz	14

Tabellenverzeichnis

2.1	Messwerte	3
3.1	Messwerte	5

1

Programmierung der AD/DA-Wandlerkarte

Den Code davon bitte den Anhang A.1.1 sehen.

2

Genauigkeit der AD-Wandlung

2.1 Fragestellung, Messprinzip, Aufbau, Messmittel

In diesem Versuch wird die Genauigkeit der AD-Wandlung von 3 Messgeräten (Multifunktionsbox ME-RedLab USB-1208LS, das hochgenaue Feinmessgerät Keithley TRMS 179 und das analoge Multimeter Philips PM 2503) ermittelt. Die Messgeräten werden mit Gleichspannungsquelle verbunden und die Spannung wird von 1 - 10 V in Schritten von 1 V variiert. Mithilfe des Python-Programms aus Versuch 1 werden die Messwerte von AD/DA-Wandlerkarte abgelesen. Bei Keithley TRMS 179 werden aus dem Oszilloskop die Messwerte angezeigt. Und bei Philips PM 2503 werden die Messwerte direkt auf dem Zeiger abgelesen. Mann nimmt die Werte von Feinmessgeräts als Referenz um Messfehler zu berechnen. Daraus werden die Standardabweichungen der beiden Geräte bestimmt. Diese werden als Genauigkeitsmaß benutzen. Zudem wird der theoretische Quantisierungsfehler berechnet.

2.2 Messwerte

Spannung in V	AD-Wandler	Philips	Keithley
1	1,035156250	1,09	1,039
2	1,992187500	2	2,001
3	3,007812500	3,08	3,06
4	4,062500000	4,09	4,07
5	5,078125000	5,09	5,082
6	5,986328125	6,05	5,998
7	7,050781250	7,06	7,01
8	8,066406250	8,08	8,022
9	9,082031250	9,09	8,986
10	9,980468750	9,69	9,999

Tabelle 2.1: Messwerte

2.3 Auswertung

Der theoretische Quantisierungsfehler:

$$\Delta U = \frac{U_{max} - U_{min}}{2^n}$$

indem U_{max} und U_{min} jeweils die maximale und minimale Eingangsspannung in Volt sind, n ist die Gesamtzahl der darstellbaren Bits.

Die Quantisierungsfehler des 11-Bit-AD-Wandlers ($-10V \dots 10V$):

$$\Delta U = 0,00976563$$

Die Standardabweichung der Messgeräten:

$$S = \sqrt{\frac{1}{n-1} \sum_{i=1}^n e_i^2} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (U_{i,ref} - U_i)^2}$$

wobei die Werte des Feinmessgeräts als Referenz $U_{i,ref}$ genommen werden, U_i sind die Messwerte der beiden anderen Messgeräten.

Die Standardabweichungen des AD-Wandler und des analogen Multimeter Philips PM 2503 sind jeweils: 0,04245769 und 0,11466424

Den Code davon bitte den Anhang A.1.1.

2.4 Interpretation

Wegen des technischen Rauschen, Anzeigefehler und Parallaxenfehler sind die Messwerte von jedem Messgerät verschieden. Deswegen sind die Standardabweichungen voneinander unterschiedlich. Umso kleiner die Standardabweichung ist, desto näher liegen die Messwerte zusammen und sich den richtigen Wert nähert. Durch die Ergebnisse könnte bestimmt werden, dass der AD-Wandler bessere Messwerte als der von dem analogen Multimeter Philips PM 2503 liefert.

Theoretisch sollte die Messfehler kleiner als 0,00976563 V sein(berechnet sind 11-Bit-AD-Wandler aber der Versuch wurde mit 16-Bit-AD-Wandler durchgeführt), aber laut der Formel entstammt dieser Wert(0,00976563 V) nur aus Quantisierungsfehler und die anderen Fehlerquelle(Rauschen, Anzeigefehler, Parallaxenfehler usw.) werden hierfür nicht berücksichtigt.

3

Genauigkeit der DA-Wandlung

3.1 Fragestellung, Messprinzip, Aufbau, Messmittel

In diesem Versuch wird die Genauigkeit der DA-Wandlung des Multifunktionsbox ME-Redlab USB-1208LS untersucht, indem die Werte von 0,5 - 5 V in 0,5 V Schritten mithilfe des Python-Programms in Konsole eingegeben werden. Die Messwerte sind in eine Tabelle einzutragen. Daraus wird die Standardabweichung bestimmt und mit dem theoretischen Quantisierungsfehler verglichen.

Den Code davon bitte den Anhang A.1.1.

3.2 Messwerte

Spannung in V	DA-Wandler
0,5	0,519
1	1,08
1,5	1,60
2	2,12
2,5	2,63
3	3,13
3,5	3,67
4	4,18
4,5	4,69
5	5,17

Tabelle 3.1: Messwerte

3.3 Auswertung

Der Quantisierungsfehler und die Standardabweichungen werden nach den Formeln aus Versuch 2 berechnet und beträgt:

$$\Delta U = 0,004882813$$

und

$$S = 0,14577799$$

3.4 Interpretation

Von der Auswertung kann es eingesehen werden, dass die Standardabweichung von der DA-Wandlung(0,14577799) größer als die von der AD-Wandlung(0,04245769). Das ist verständlich, da die Umwandlung von Analog in Digital immer genauer als die umgekehrte Umwandlung ist.

4

Zeitverhalten der DA-Wandlung

4.1 Fragestellung, Messprinzip, Aufbau, Messmittel

In diesem Versuch wird das Verhalten im Laufe der Zeit der DA-Wandlung untersucht, indem die Werte unter eine Sinusspannung sequentiell hintereinander auf die Karte ausgegeben werden. Es sollte beachtet werden, dass zwischen den einzelnen Ausgaben geeignete Pausen einhalten müssen, um die Samplingrate des DA-Wandlers einzuhalten. Hier wurde 10ms als Pause gewählt.

Den Code zum Generieren einer Sinusspannung bitte den Anhang A.1.2.

4.2 Messwerte

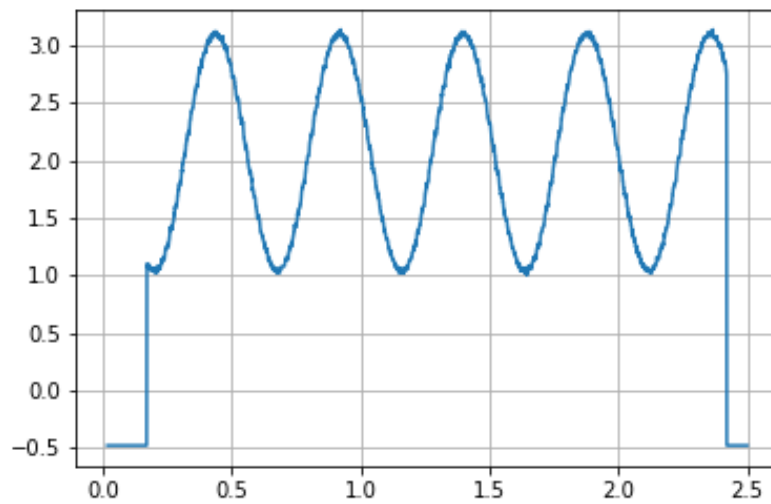


Abbildung 4.1: Erzeugte Sinusschwingung durch DA-Wandlung

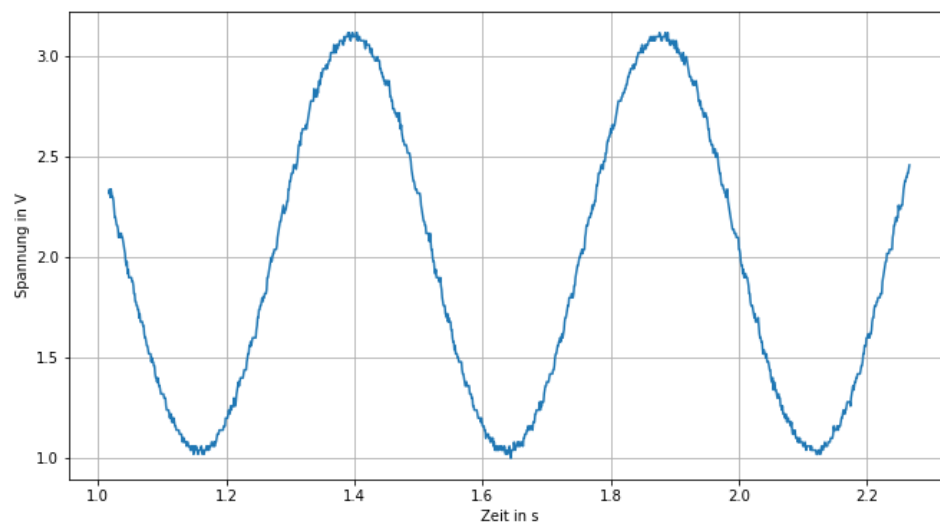


Abbildung 4.2: Abgeschnittene und vergrößerte Sinusschwingung

4.3 Auswertung

Wir haben das Programm modifiziert, dass zwischen den einzelnen Ausgaben keine Pausen entstehen sollten, um die Wirkung von der Pause auf die Samplingrate zu analysieren.

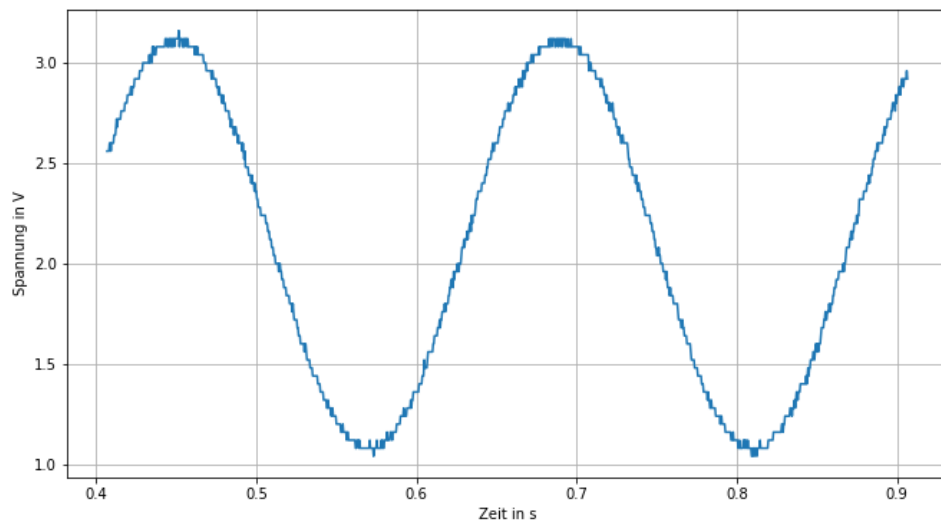


Abbildung 4.3: Erzeugte Sinusschwingung ohne Pause

Deutlich liefert es eine Sinusschwingung mit der größeren Frequenz als die von der mit Pause. Und die ohne Pause Schwingung hat mehr eckige Stelle in seiner Form(geringere Genauigkeit, da die Samplingrate von DA-Wandler nicht groß genug ist).

4.4 Interpretation

Allgemein sehen die beide Sinusspannung ähnlich wie eine tatsächliche Sinusschwingung aus. Der Unterschied liegt daran, dass das Signal bei der Umwandlung von Digital in Analog nie kontinuierlich richtig dargestellt wird.

5

Abtasttheorem

5.1 Fragestellung, Messprinzip, Aufbau, Messmittel

In diesem Versuch wird die tatsächliche Abtastfrequenz des AD-Wandlers ermittelt. Dazu soll eine Abtastfrequenz im Intervall $[6000, 8000]$ ausgewählt und mit dem entsprechenden RedLab-Befehl an den AD-Wandler gegeben werden. Dazu wird auch die Nyquist-Frequenz bestimmt.

Mit dem Sinusgenerator wird die Frequenz in 7 Schritten (angefangen von der halben Nyquist-Frequenz bis zur doppelten Nyquist-Frequenz) variiert und die Kurven davon grafisch darzustellen.

Den Code davon bitte den Anhang A.1.3.

5.2 Messwerte

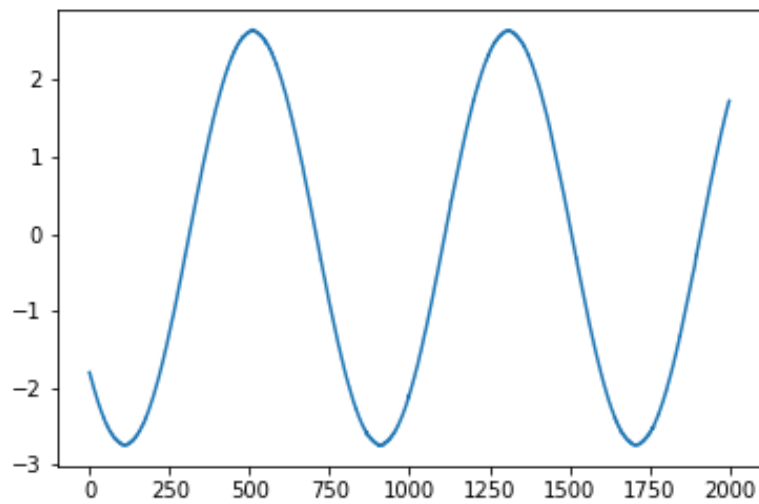


Abbildung 5.1: Schwingung mit der Frequenz 10Hz

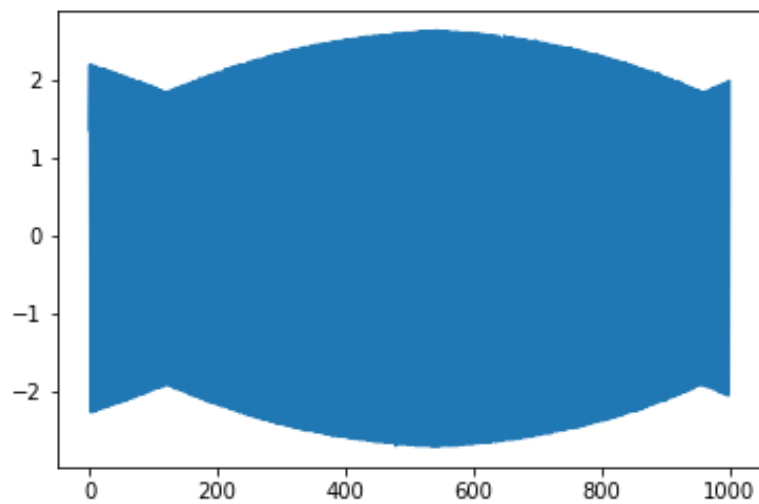


Abbildung 5.2: Schwingung mit der Frequenz 2000Hz

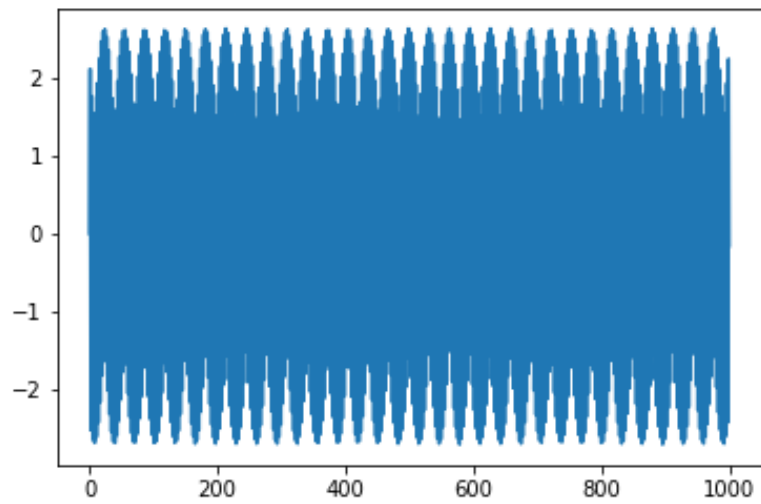


Abbildung 5.3: Schwingung mit der Frequenz 2750Hz

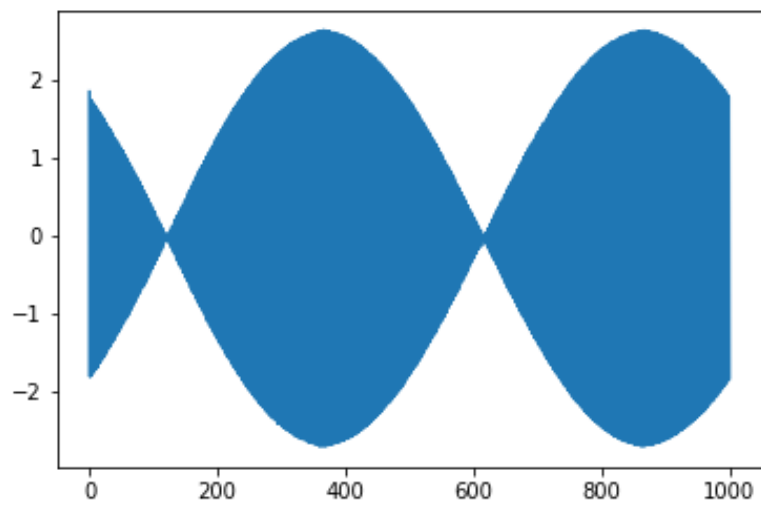


Abbildung 5.4: Schwingung mit der Nyquist-Frequenz 4000Hz

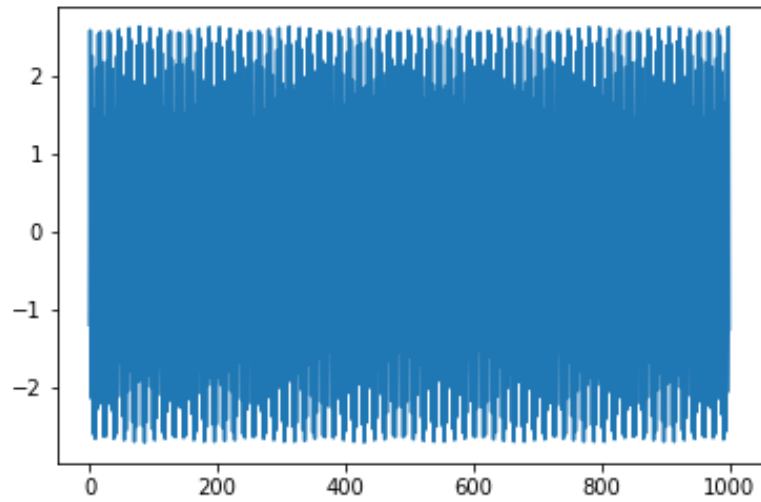


Abbildung 5.5: Schwingung mit der Frequenz 5500Hz

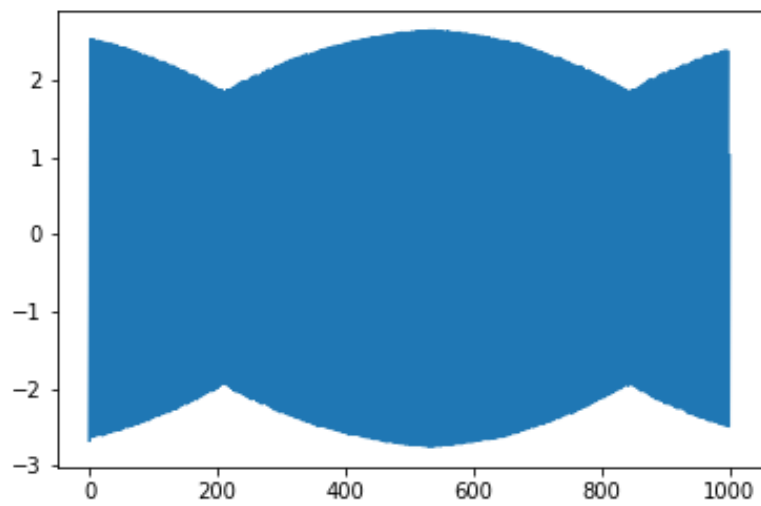


Abbildung 5.6: Schwingung mit der Frequenz 6000Hz

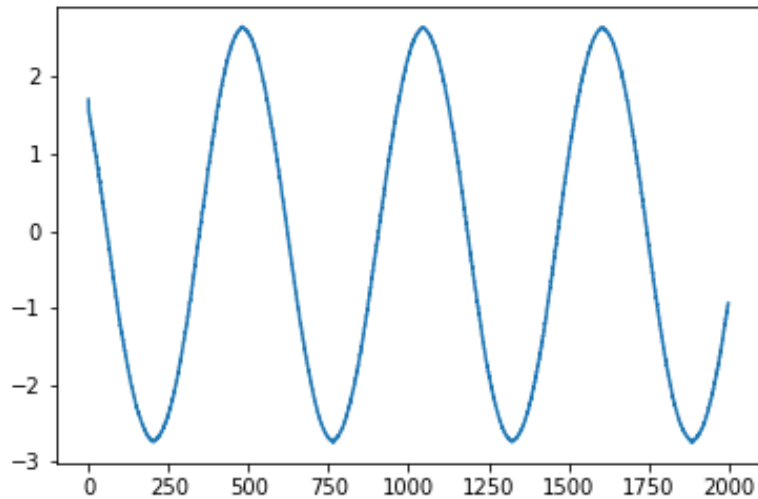


Abbildung 5.7: Schwingung mit der Abtastfrequenz 8000Hz

5.3 Auswertung

Hierfür wurde 8000 Hz als Abtastfrequenz ausgewählt. Die tatsächliche Abtastfrequenz des AD-Wandlers beträgt 8021 Hz. Die Nyquist-Frequenz ist halbe Abtastfrequenz(4000 Hz).

5.4 Interpretation

Nach dem Abtasttheorem sind die Sinusschwingungen mit der Frequenz unter Nyquist-Frequenz darstellbar. D.h. die Wellenform ist keine Sinusform mehr trotzdem bleibt die Frequenz erhalten. Eine Sinusschwingung mit der Frequenz höher als Nyquist-Frequenz aber kleiner als Abtastfrequenz wird nicht mehr fehlerfrei aus dem abgetasteten Signal rekonstruiert. Es wird so rekonstruiert als ob es eine Spiegelung an Nyquist-Frequenz stattfindet z.B. 6000 Hz($4000 \text{ Hz} + 2000 \text{ Hz}$) ist die Spiegelung von 2000 Hz($4000 \text{ Hz} - 2000 \text{ Hz}$). Deswegen sehen Abbildungen 5.2 und 5.6 / 5.1 und 5.7 fast gleichförmig aus.

Anhang

A.1 Quellcode

A.1.1 Quellcode Versuch 1-3

```
1 import redlab as rl
2 import matplotlib.pyplot as pl
3
4 print("----- einzelne Werte -----")
5 print("16 Bit Value: " + str(rl.cbAIn(0,0,1)))
6 print("Voltage Value: " + str(rl.cbVIn(0,0,1))) #Versuch2
7 print("----- Messreihe -----")
8 a = rl.cbVInScan(0,0,0,1000,8000,1)
9 print("Messreihe: " + str(a))
10 pl.plot(a)
11 pl.show()
12 print("Messreihe: " + str(rl.cbVInScan(0,0,0,300,8000,1)))
13 print("SampleRate: " + str(rl.cbInScanRate(0,0,0,8000)))
14 ###Konsoleneingaben### #versuch3
15 x = input("Spannung:")
16 x = float(x)
17 rl.cbVOut(0,0,101,x)
```

A.1.2 Quellcode Sinusschwingung-Generator Versuch4

```
1 import redlab as rl
2 import numpy as np
3 import time
4 import matplotlib.pyplot as pl
5
6 x = np.linspace(0,2 * np.pi,30, endpoint=False)
```

```

7 print(np.sin(x))
8 while(1):
9     for i in x:
10         rl.cbVOut(0,0,101,np.sin(i)+2)
11         time.sleep(0.01)
12
13 #####für Darstellung###
14 #v4 = np.genfromtxt("v4_Messwert.csv",delimiter=";",skip_header=17)
15 #x = v4[1000:2250,3:4]
16 #y = v4[1000:2250,4:5]
17 #pl.figure(figsize=(11,6))
18 #pl.xlabel('Zeit in s')
19 #pl.ylabel('Spannung in V')
20 #pl.plot(x,y)
21 #pl.grid()
22 #pl.savefig('v4neu.png')
23 #pl.show()

```

A.1.3 Quellcode Versuch 5

```

1 import redlab as rl
2 import numpy as np
3 import matplotlib.pyplot as pl
4
5 print("Samplerate: " + str(rl.cbInScanRate(0,0,0,8000)))
6 a = rl.cbVInScan(0,0,0,1000,8000,1)
7 np.save('5500Hz', a)
8
9 index = [10,2000,2750,4000,5500,6000,8000]
10
11
12 for i in index:
13     v5 = np.load(str(i)+'Hz.npy')
14     pl.plot(v5)
15     pl.savefig(str(i) + '.png')
16     pl.show()

```