



**HOCHSCHULE KONSTANZ TECHNIK, WIRTSCHAFT UND GESTALTUNG**  
UNIVERSITY OF APPLIED SCIENCES

**Technische Grundlagen  
der angewandten Informatik**

# **RedLab - Python**

**J. Keppler**

**Konstanz, 12. Juni 2017**

## Zusammenfassung (Abstract)

Thema: RedLab - Python

Autoren: J. Keppler

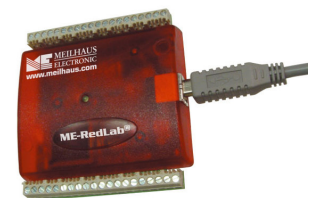
jkeppler@htwg-konstanz.de

Python ist eine universelle, üblicherweise interpretierte Programmiersprache. Sie unterstützt mehrere Programmierparadigmen, z.B. die objektorientierte, die aspektorientierte und die funktionale Programmierung. Wie viele dynamische Sprachen wird Python oft als Skriptsprache genutzt.



RedLab-Python ist eine Programmbibliothek, welche einen Wrapper für die API von ME RedLab® zur Verfügung stellt (nur spezielle Funktionen). Basis der API ist die Universal Library™ von MC MEASUREMENT COMPUTING. Die Python Programmbibliothek *redlab* wurde unter Windows®7 mit Visual Studio® für Python 3.5 (64 Bit) entwickelt. Eingesetzt wird die Bibliothek in den Laborversuchen von SSS mit der DAQ-USB-Box RedLab 1208LS.

Python gilt als einfach zu erlernende Sprache, da sie über eine klare und übersichtliche Syntax verfügt. Ferner besitzt sie umfangreiche Programmbibliotheken.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Installation . . . . .	1
<b>2</b>	<b>Funktionsreferenzen</b>	<b>2</b>
2.1	Analog I/O Funktionen . . . . .	2
2.1.1	cbAIn() Funktion . . . . .	2
2.1.2	cbAInScan() Funktion . . . . .	3
2.1.3	cbAOut() Funktion . . . . .	4
2.1.4	cbVIn() Funktion . . . . .	5
2.1.5	cbVInScan() Funktion . . . . .	6
2.1.6	cbVOut() Funktion . . . . .	7
2.1.7	cbInScanRate() Funktion . . . . .	7
	<b>Anhang</b>	<b>9</b>
A.1	A/D-Eingabe und D/A-Ausgabe . . . . .	9
A.2	Auszug Datenblatt ME-1208LS (Spezifikationen) . . . . .	10

# 1

## Einleitung

### 1.1 Installation

*RedLab-Python* ist eine Wrapper-Klasse für die *Universal Library* von Measurement Computing. Die API stellt entsprechende Funktionen für den Zugriff auf die DAQ-USB-Box RedLab 1208LS zur Verfügung.

Das **RedLab-Python** - Modul wurde unter Windows 7 für Python 3.5 mit 64 Bit kompiliert. Dazu wurde Visual Studio 2013 verwendet.

Die Installation wird in folgenden Schritten durchgeführt:

1. Das Treiberpaket von Meilhaus für die RedLab-Baureihe wird zunächst auf dem Computer installiert.
2. Das Microsoft Visual C++ 2013 Redistributable Package (x64) [vcredist\_x64.exe] wird zunächst auf dem Computer ausgeführt.
3. Die Datei redlab.pyd wird in das Unterverzeichnis <python directory>\DLLs kopiert.

# 2

## Funktionsreferenzen

### 2.1 Analog I/O Funktionen

#### 2.1.1 cbAIn() Funktion

Der angegebene A/D - Eingangskanal auf der spezifizierten Karte wird ausgelesen und der vorzeichenlose Wert im 16-Bit Format zurückgegeben. Wenn die spezifizierte A/D - Karte einen programmierbaren Verstärker besitzt, wird zunächst die Verstärkung eingestellt und der A/D - Wert entsprechend konvertiert.

#### Funktionsprototyp

*Python*

cbAIn(int BoardNum, int Channel, int Rang) -> Long

#### Argumente

BoardNum

Die Nummer der Karte, welche verwendet werden soll. BoardNum liegt im Bereich von 0 und 99. Die Nummerierung der Karten erfolgt durch das Programm InstaCal.

Channel

A/D Kanalnummer. Die größte Kanalnummer hängt vom Type der A/D-Karte welche benutzt wird ab. Bei Karten mit Single-ended und "differentielle" Eingängen zusätzlich von der Konfiguration.

Rang

A/D Spannungsbereichscode. Für den Bereich  $-10V$  bis  $+10V$  ist dieser **1**.

## **Rückgabewert**

Long

Der gelesene Wert des A/D Kanals. Entsprechend der Auflösung des Kanals.

### **2.1.2   cbAInScan() Funktion**

Liest einen Bereich von A/D-Eingangskanälen auf der spezifizierten Karte. Die Funktion liest eine angegebene Anzahl von A/D-Werten von diesen Kanälen . Wenn die spezifizierte A/D-Karte einen programmierbaren Verstärker besitzt, wird zunächst die Verstärkung eingestellt und der A/D-Wert entsprechend konvertiert.

## **Funktionsprototyp**

*Python*

```
cbAInScan(int BoardNum, int LowChan, int HighChan, long Count, long Rate, int  
Rang) -> List[Long]
```

## **Argumente**

BoardNum

Die Nummer der Karte, welche verwendet werden soll. BoardNum liegt im Bereich von 0 und 99. Die Nummerierung der Karten erfolgt durch das programm InstaCal.

LowChan

Die kleiner A/D Kanalnummer, untere Grenze des Bereichs welcher gelesen werden soll.

HighChan

Die größere A/D Kanalnummer, obere Grenze des Bereichs welche gelesen werden soll.

Count

Die Anzahl der zu lesenden A/D-Werte. Dies entspricht der Gesamtzahl der A/D-Werte. Werden mehrere Kanäle gelesen, berechnet sich die Anzahl pro Kanal wie folgt  $Count / (HighChan - LowChan + 1)$ .

Rate

Die Rate für die Datenerfassung, in Samples pro Sekunde pro Kanal.

Rang

A/D Spannungsbereichscode. Für den Bereich  $-10V$  bis  $+10V$  ist dieser **1**.

### **Rückgabewert**

Long

Der gelesene Wert des A/D Kanals. Entsprechend der Auflösung des Kanals.

### **2.1.3 cbAOut() Funktion**

Diese Funktion setzt den Wert für die D/A-Ausgabe.

#### **Funktionsprototyp**

*Python*

cbAOutn(int BoardNum, int Channel, int Rang, unsigned DataValue) -> None]

#### **Argumente**

BoardNum

Die Nummer der Karte, welche verwendet werden soll. BoardNum liegt im Bereich von 0 und 99. Die Nummerierung der Karten erfolgt durch das Programm InstaCal.

Channel

D/A-Kanalnummer. Die Kanalnummer sind von der Karte abhängig.

Rang

D/A Spannungsbereichscode. Wenn die spezifizierte D/A-Karte keinen programmierten Verstärker besitzt, wird dieser Wert ignoriert. Für den Bereich 0 - 5 V ist dieser **101**.

DataValue

Wert welcher am D/A-Kanal gesetzt werden soll. Dieser muss im Bereich  $0 - n$  liegen, wobei  $n = 2^{\text{Auflösung}} - 1$  der entsprechende Auflösung ist.

## **Rückgabewert**

None

### **2.1.4   cbVIn() Funktion**

Der angegebene A/D - Eingangskanal auf der spezifizierten Karte wird ausgelesen und der Spannungswert zurückgegeben.

## **Funktionsprototyp**

*Python*

cbAIn(int BoardNum, int Channel, int Rang) -> Long

## **Argumente**

BoardNum

Die Nummer der Karte, welche verwendet werden soll. BoardNum liegt im Bereich von 0 und 99. Die Nummerierung der Karten erfolgt durch das Programm InstaCal.

Channel

A/D Kanalnummer. Die größte Kanalnummer hängt vom Type der A/D-Karte welche benutzt wird ab. Bei Karten mit Bingle-ended und "differentiell" Eingängen zusätzlich von der Konfiguration.

Rang

A/D Spannungsbereichscode. Für den Bereich  $-10V$  bis  $+10V$  ist dieser **1**.

## **Rückgabewert**

float

Der gelesene Spannungswert des A/D Kanals.



### 2.1.5 cbVInScan() Funktion

Liest einen Bereich von A/D-Eingangskanälen auf der spezifizierten Karte. Die Funktion liest eine angegebene Anzahl von Spannungswerten von diesen Kanälen .

#### Funktionsprototyp

*Python*

```
cbAInScan(int BoardNum, int LowChan, int HighChan, long Count, long Rate, int  
Rang) -> List[Long]
```

#### Argumente

BoardNum

Die Nummer der Karte, welche verwendet werden soll. BoardNum liegt im Bereich von 0 und 99. Die Nummerierung der Karten erfolgt durch das programm InstaCal.

LowChan

Die kleiner A/D Kanalnummer, untere Grenze des Bereichs welcher gelesen werden soll.

HighChan

Die größere A/D Kanalnummer, obere Grenze des Bereichs welche gelesen werden soll.

Count

Die Anzahl der zu lesenden A/D-Werte. Dies entspricht der Gesamtzahl der A/D-Werte. Werden mehrere Kanäle gelesen, berechnet sich die Anzahl pro Kanal wie folgt  $Count / (HighChan - LowChan + 1)$ .

Rate

Die Rate für die Datenerfassung, in Samples pro Sekunde pro Kanal.

Rang

A/D Spannungsbereichscode. Wenn die spezifizierte A/D-Karte keinen programmierten Verstärker besitzt, wird dieser Wert ignoriert. Für den Bereich  $-10V$  bis  $+10V$  ist dieser 1.

#### Rückgabewert

Long

Der gelesene Wert des A/D Kanals. Entsprechend der Auflösung des Kanals.

### **2.1.6   cbVOut() Funktion**

Diese Funktion setzt den Spannungswert für die D/A-Ausgabe.

#### **Funktionsprototyp**

*Python*

cbVOut(int BoardNum, int Channel, int Rang, float DataValue) -> None]

#### **Argumente**

BoardNum

Die Nummer der Karte, welche verwendet werden soll. BoardNum liegt im Bereich von 0 und 99. Die Nummerierung der Karten erfolgt durch das Programm InstaCal.

Channel

D/A-Kanalnummer. Die Kanalnummer sind von der Karte abhängig.

Rang

D/A Spannungsbereichscode. Für den Bereich 0 - 5 V ist dieser **101**.

DataValue

Spannungswert welcher am D/A-Kanal gesetzt werden soll.

#### **Rückgabewert**

None

### **2.1.7   cbInScanRate() Funktion**

Liefert die tatsächliche Sampling Rate.

#### **Funktionsprototyp**

*Python*

cbInScanRate(int BoardNum, int LowChan, int HighChan, long Rate) -> Float

### **Argumente**

BoardNum

Die Nummer der Karte, welche verwendet werden soll. BoardNum liegt im Bereich von 0 und 99. Die Nummerierung der Karten erfolgt durch das Programm InstaCal.

LowChan

Die kleinere A/D Kanalnummer, untere Grenze des Bereichs welcher gelesen werden soll.

HighChan

Die größere A/D Kanalnummer, obere Grenze des Bereichs welcher gelesen werden soll.

Rate

Die Rate für die Datenerfassung, in Samples pro Sekunde pro Kanal.

### **Rückgabewert**

Float

Die tatsächliche Sampling Rate.

# Anhang

## A.1 A/D-Eingabe und D/A-Ausgabe

```
import redlab as rl

print("-----_einzelne_Werte_-----")
print("16_Bit_Value:_" + str(rl.cbAIn(0,0,1)))
print("Voltage_Value:_" + str(rl.cbVIn(0,0,1)))
print("-----_Messreihe_-----")
print("Messreihe:_" + str(rl.cbAInScan(0,0,0,300,8000,1)
    ))
print("Messreihe:_" + str(rl.cbVInScan(0,0,0,300,8000,1)
    ))
print("Samplerate:_" + str(rl.cbInScanRate(0,0,0,8000)))
print("-----_Ausgabe_-----")
print("Voltage_Value:_" + str(rl.cbVOut(0,0,101,2.5)))
```

Listing 3.1: A/D input and D/A output

## **A.2 Auszug Datenblatt ME-1208LS (Spezifikationen)**

# Spezifikationen

Sofern nicht anders vermerkt, gelten alle Angaben für 25 °C.

## Analogeingabe

Parameter	Bedingungen	Spezifikationen
A/D-Wandlertyp		A/D-Wandler vom Typ der sukzessiven Approximation
Eingangsspannungsbereich für lineare Operation, single-ended	CHx nach GND	±10 V max.
Eingangs-Common-Mode-Spannungsbereich für lineare Operation, differentieller Modus	CHx nach GND	-10 V min., +20 V max.
Maximal zulässige Eingangs-Absolutspannung	CHx nach GND	±40 V max.
Stromaufnahme (Hinweis 1)	Vin = +10 V	70 µA typ.
	Vin = 0 V	-12 µA typ.
	Vin = -10 V	-94 µA typ.
Anzahl der Kanäle		8 single-ended / 4 differentielle, per Software wählbar
Eingangsbereiche, single-ended		±10 V, G=2
Eingangsbereiche, differentiell		±20 V, G=1 ±10 V, G=2 ±5 V, G=4 ±4 V, G=5 ±2,5 V, G=8 ±2,0 V, G=10 ±1,25 V, G=16 ±1,0 V, G=20 per Software wählbar
Durchsatz	per Software gesteuert	50 S/s
	kontinuierliche Erfassung	1,2 kS/s
	blockweise Erfassung ins 4k Werte-FIFO	8 kS/s
Kanalliste	bis zu 8 Elemente	Kanal, Bereich und Verstärkung per Software konfigurierbar
Auflösung (Hinweis 2)	differentiell	12 Bit, keine fehlenden Codes
	single-ended	11 Bit
CAL-Genauigkeit	CAL = 2,5 V	±0,05 % typ., ±0,25 % max.
Integraler Linearitätsfehler		±1 LSB typ.
Differentieller Linearitätsfehler		±0,5 LSB typ.
Wiederholgenauigkeit		±1 LSB typ.
CAL Strom	Quelle	max. 5 mA
	Senke	20 µA min., 200 nA typ.
Triggerquelle	per Software wählbar	extern digital: TRIG_IN

Hinweis 1: Der Eingangsstrom ist von der an die Analogkanäle angelegten Spannung abhängig. Bei einer gegebenen Eingangsspannung Vin ist der Eingangssperrstrom näherungsweise  $(8,181 \cdot V_{in} - 12) \mu A$ .

Hinweis 2: Der Wandler AD7870 gibt im single-ended Modus nur 11 Bits (Codes 0-2047) zurück.

Tabelle 4-1 Genauigkeit, differentieller Modus

Bereich	Genauigkeit (LSB)
±20 V	5,1
±10 V	6,1
±5 V	8,1
±4 V	9,1
±2,5 V	12,1
±2 V	14,1
±1,25 V	20,1
±1 V	24,1

Tabelle 4-2 Genauigkeit, single-ended Modus

Bereich	Genauigkeit (LSB)
±10 V	4,0

Tabelle 4-3 Genauigkeitskomponenten – differentieller Modus - alle Werte sind (±)

Bereich	% der Anzeige	Verstärkungs-fehler bei Vollausschlag (mV)	Offset (mV)	Genauigkeit bei Vollausschlag (mV)
±20 V	0,2	40	9,766	49,766
±10 V	0,2	20	9,766	29,766
±5 V	0,2	10	9,766	19,766
±4 V	0,2	8	9,766	17,766
±2,5 V	0,2	5	9,766	14,766
±2 V	0,2	4	9,766	13,766
±1,25 V	0,2	2,5	9,766	12,266
±1 V	0,2	2	9,766	11,766

Tabelle 4-4 Genauigkeitskomponenten – single-ended Modus - alle Werte sind (±)

Bereich	% der Anzeige	Verstärkungs-fehler bei Vollausschlag (mV)	Offset (mV)	Genauigkeit bei Vollausschlag (mV)
±10 V	0,2	20	19,531	39,531

## Analogausgang

Parameter	Bedingungen	Spezifikationen
D/A-Wandlertyp		PWM
Auflösung		10 Bit, 1 in 1024
Maximaler Ausgangsspannungsbereich		0 bis 5 V
Anzahl der Kanäle		2 Spannungsausgänge
Durchsatz	per Software gesteuert	100 S/s Einkanal-Betrieb 50 S/s Zweikanal-Betrieb
Einschalt- und Rücksetzspannung		initialisiert zum Code 000h
Maximale Spannung (Hinweis 3)	ohne Last	Vs
	1 mA Last	0,99*Vs
	5 mA Last	0,98*Vs
Ausgangsstrom	pro D/A OUT	30 mA
Anstiegszeit		0,14 V/mS typ.

Hinweis 3: Vs ist die +5 V Spannung vom USB-Bus. Die maximale Analogausgangsspannung entspricht Vs ohne Last, ist systemabhängig und kann auch weniger als 5 V betragen.

## Digitaleingang/-ausgang

Digitaltyp	82C55
Anzahl der Eingänge/Ausgänge	16 (Port A0 bis A7, Port B0 bis B7)
Konfiguration	2 Ports zu je 8 Bit
Pull-up/Pull-down-Konfiguration	Alle Anschlüsse sind über einen 47-kOhm-Widerstand mit Vs verbunden (Standard). Pull-down Verbindung gegen Masse ist möglich. Über einen 0-Ohm-Widerstand wählbar.
Eingangsspannung HIGH	2,0 V min., 5,5 V absolut max.
Eingangsspannung LOW	0,8 V max., -0,5 V absolut min.
Ausgangsspannung HIGH (IOH = -2,5 mA)	3,0 V min.
Ausgangsspannung LOW (IOL = 2,5 mA)	0,4 V max.

## Externer Trigger

Parameter	Bedingungen	Spezifikationen
Triggerquelle (Hinweis 4)	extern digital	TRIG_IN
Trigger-Modus	per Software wählbar	pegelabhängig: benutzerkonfigurierbar für HIGH- oder LOW-TTL-Eingangspegel
Triggerverzögerung	blockweise	25 µs min., 50 µs max.
Triggerimpulsbreite	Block	40 µs min.
Eingangsspannung für HIGH		3,0 V min., 15,0 V absolut max.
Eingangsspannung für LOW		0,8 V max.
Eingangssperrstrom		± 1,0 µA

Hinweis 4: TRIG\_IN ist durch einen 1,5-kΩ-Serienwiderstand geschützt.

## Zähler

Zählertyp	Ereigniszähler
Anzahl der Kanäle	1
Eingangstyp	TTL, triggert auf steigende Flanke
Eingangsquelle	Schraubklemme CTR
Auflösung	32 Bit
Schmitt-Trigger-Hysteresese	20 mV bis 100 mV
Eingangssperrstrom	± 1 µA
Max. Eingangsfrequenz	1 MHz
Impulsbreite für HIGH	500 ns min.
Impulsbreite für LOW	500 ns min.
Eingangsspannung für LOW	0 V min., 1,0 V max.
Eingangsspannung für HIGH	4,0 V min., 15,0 V max.

## Nichtflüchtiger Speicher

Speichergröße	8192 Byte		
Speicherkonfiguration	<b>Adressbereich</b>	<b>Zugriff</b>	<b>Beschreibung</b>
	0x0000 – 0x17FF	lesen/schreiben	A/D-Daten (4K Abtastdaten)
	0x1800 – 0x1EFF	lesen/schreiben	Benutzerdatenbereich
	0x1F00 – 0x1FEF	lesen/schreiben	Kalibrierdaten
	0x1FF0 – 0x1FFF	lesen/schreiben	Systemdaten



## Stromversorgung

Parameter	Bedingungen	Spezifikationen
Stromaufnahme (Hinweis 5)		20 mA
Verfügbare +5V USB-Spannungsversorgung (Hinweis 6)	mit eigenversorgtem Hub	4,5 V min., 5,25 V max.
	mit über den Bus versorgtem Hub	4,1 V min., 5,25 V max.
Ausgangsstrom (Hinweis 7)	mit eigenversorgtem Hub	450 mA min., 500 mA max.
	mit über den Bus versorgtem Hub	50 mA min., 100 mA max.

Hinweis 5: Hierbei handelt es sich um den gesamten vom RedLab 1208LS benötigten Strom einschließlich bis zu 5 mA für die Zustands-LED.

Hinweis 6: Eigenversorgt bezieht sich auf USB-Verteiler und Hosts mit einer Stromversorgung. Über den Bus versorgt bezieht sich auf USB-Hub und Hosts ohne eigene Stromversorgung.

Hinweis 7: Dies bezieht sich auf den Gesamtstrom, der vom USB +5 V, den Analog- und Digitalausgängen zur Verfügung gestellt werden kann.

## Allgemeines

Parameter	Bedingungen	Spezifikationen
Taktfrequenzfehler des USB-Controller	25 °C	±30 ppm max.
	0 bis 70 °C	±50 ppm max.
	-40 bis 85 °C	±100 ppm max.
Gerätetyp		USB 1.1 (niedrige Datenübertragungsrate)
Geräte-Kompatibilität		USB 1.1, USB 2.0

## Umgebungsbedingungen

Betriebstemperaturbereich	-0 bis 70 °C
Lagertemperaturbereich	-40 bis 85 °C
Feuchtigkeit	0 bis 90 % nicht kondensierend

## Mechanisches

Abmessungen	79 mm (L) x 82 mm (T) x 25 mm (H)
Länge des USB-Kabels	max. 3 Meter
Länge der Benutzerverbindung	max. 3 Meter

## Hauptanschluss und Anschlussverdrahtung

Anschlusstyp	Schraubklemmen
Leitungsquerschnitt	AWG 16 bis 30

#### 4 Kanäle, differentieller Modus

Anschluss	Signal-bezeichnung	Anschluss	Signal-bezeichnung
1	CH0 IN HI	21	Port A0
2	CH0 IN LO	22	Port A1
3	GND	23	Port A2
4	CH1 IN HI	24	Port A3
5	CH1 IN LO	25	Port A4
6	GND	26	Port A5
7	CH2 IN HI	27	Port A6
8	CH2 IN LO	28	Port A7
9	GND	29	GND
10	CH3 IN HI	30	PC+5V
11	CH3 IN LO	31	GND
12	GND	32	Port B0
13	D/A OUT 0	33	Port B1
14	D/A OUT 1	34	Port B2
15	GND	35	Port B3
16	CAL	36	Port B4
17	GND	37	Port B5
18	TRIG_IN	38	Port B6
19	GND	39	Port B7
20	CTR	40	GND

#### 8 Kanäle, single-ended Modus

Anschluss	Signal-bezeichnung	Anschluss	Signal-bezeichnung
1	CH0 IN	21	Port A0
2	CH1 IN	22	Port A1
3	GND	23	Port A2
4	CH2 IN	24	Port A3
5	CH3 IN	25	Port A4
6	GND	26	Port A5
7	CH4 IN	27	Port A6
8	CH5 IN	28	Port A7
9	GND	29	GND
10	CH6 IN	30	PC+5V
11	CH7 IN	31	GND
12	GND	32	Port B0
13	D/A OUT 0	33	Port B1
14	D/A OUT 1	34	Port B2
15	GND	35	Port B3
16	CAL	36	Port B4
17	GND	37	Port B5
18	TRIG_IN	38	Port B6
19	GND	39	Port B7
20	CTR	40	GND