

# Python data tree

*andere Daten einlesen, anschauen, ...*

**engelbert gruber**

16.12.2020

Rev. 0.1

## INTRODUCTION

Im google “ecg signal sample data” und dann bekomme ich:

<https://physionet.org/about/database/>, [PTB-XL, a large publicly available electrocardiography dataset](#), [TELE ECG Database: 250 telehealth ECG records](#), ...

## MATERIALS

1. Verschiedene Messdatensätze
2. PC mit python3
3. Interesse

## DAS ANZEIGE PROGRAMM

Gegenüber der ersten Version leicht verändert.

- Den Dateinamen von der Kommandozeile aus übernehmen, damit wir nicht für jede ECG-Datei das Programm ändern müssen.
- Hilfe-Text.

Präambel für Unix-Shell, damit die shell den richtigen Interpreter aufruft.

```
#!/usr/bin/python3
```

Python doc-String. Zwischen zwei dreifach Anführungszeichen kann ein beliebig langer Text eingegeben werden. Wenn kein Name vergeben wird, speichert python den Text in dem “`__doc__`”-Attribut der Datei, Klasse oder Funktion.

```
"""
plot a csv files content

USAGE: plt.py <csv-file-name>
"""
```

Wir geben dem doc-String den Namen “USAGE”

```
USAGE = __doc__
```

Das Modul “`sys`” um auf die Kommandozeilenargumente () zugreifen zu können.

Das Modul “`pathlib`” um überprüfen zu können, ob etwas eine Datei ist.

```
import sys, pathlib
```

“`sys.argv`”, argument values. Ist eine Liste, der erste Eintrag ist der Programmname, der zweite soll der Name der EKG-Datei sein. Wenn das nicht zutrifft wird der kurze Hilfetext ausgegeben und das Programm beendet.

```
if len(sys.argv) < 2 or not
pathlib.Path(sys.argv[1]).is_file():
    print(USAGE)
    sys.exit(1)
```

Wenn das zweite Argument eine existierende Datei ist, geben wir den Namen “`fname`”.

```
fname = sys.argv[1]
```

Ab hier ist es (beinahe) das Programm vom ersten Teil.

```
import csv
import matplotlib.pyplot as plt
data = []
time_in_ms = []
for ln in csv.reader(open(fname)):
    time_in_ms.append(len(time_in_ms) * 2)
```

Wir hängen die alle Spalten aus der csv-Datei an, nicht nur die erste.

```
data.append( list(map(float, ln)) )
```

Wir konstruieren uns eine Legende, damit die Spalte erkennbar ist.

```
legend = []
```

“data” die Liste (Array) der Werte. Jeder Eintrag enthält eine Liste mit den Werten aus einer Zeile der csv-Datei.

```
[[3.63587, 0.0, 1.0, 1.0], [3.70913, 0.0, 1.0, 1.0],
[4.007597, ...
```

“data[0]” die Liste der ersten Kanalwerte

```
[3.63587, 0.0, 1.0, 1.0]
```

“len(data[0])” die Länge der Liste, in diesem Fall 4.

“range(len(data[0]))” erzeugt eine Liste mit “4” Einträgen: 0, 1, 2, 3

““d{0}”.format(i)” erzeugt einen String, bei dem “{0}” durch den Wert von “i” ersetzt wird. “legend” ist nach der for-Schleife “[ ‘d0’, ‘d1’, ‘d2’, ‘d3’ ]”

```
for i in range(len(data[0])):
    legend.append("d{0}".format(i))
plt.plot(time_in_ms, data)
```

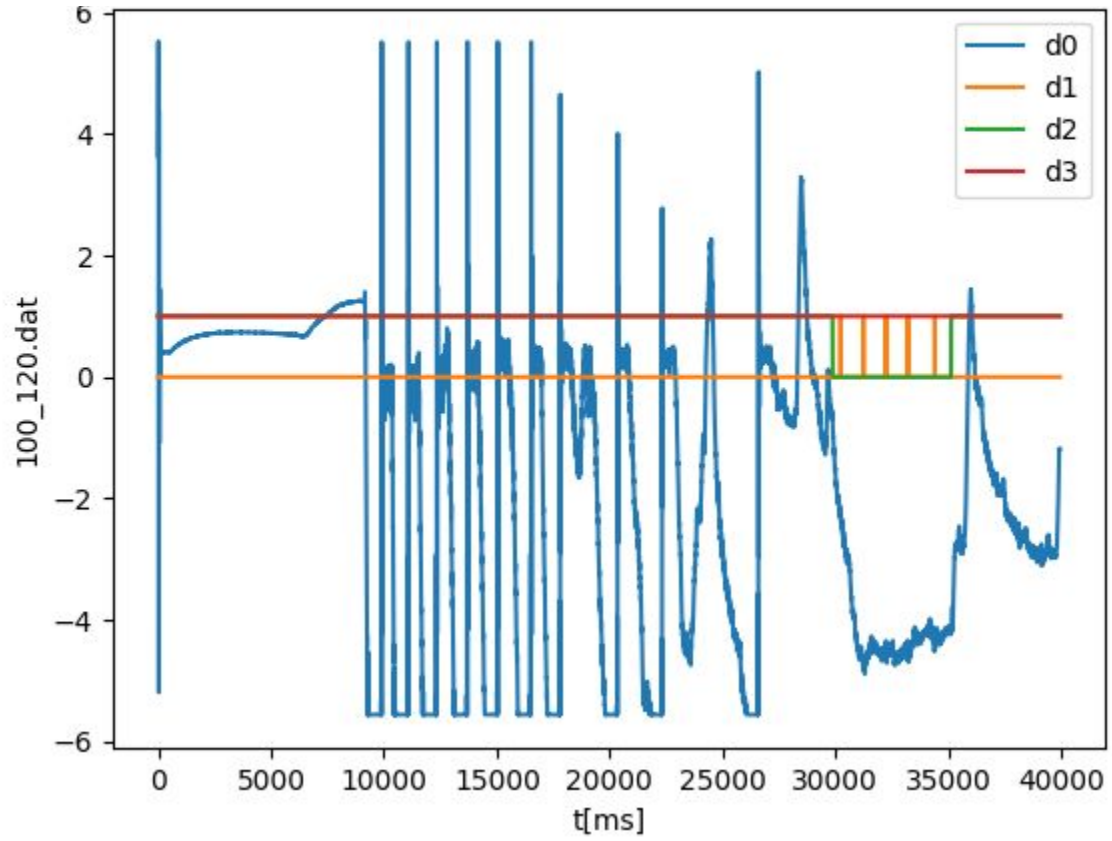
Wir geben die Texte für die Kanallegende und setzen den Dateinamen als ylabel.

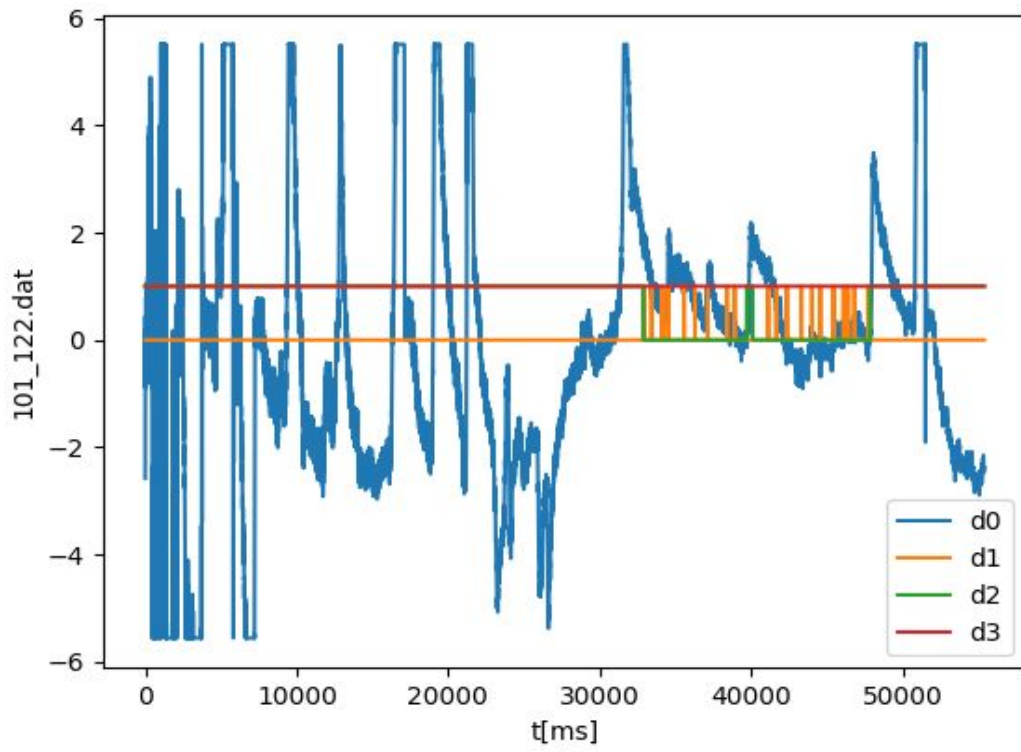
```
plt.legend(legend)
plt.ylabel(pathlib.Path(fname).parts[-1])
plt.xlabel("t[ms]")
```

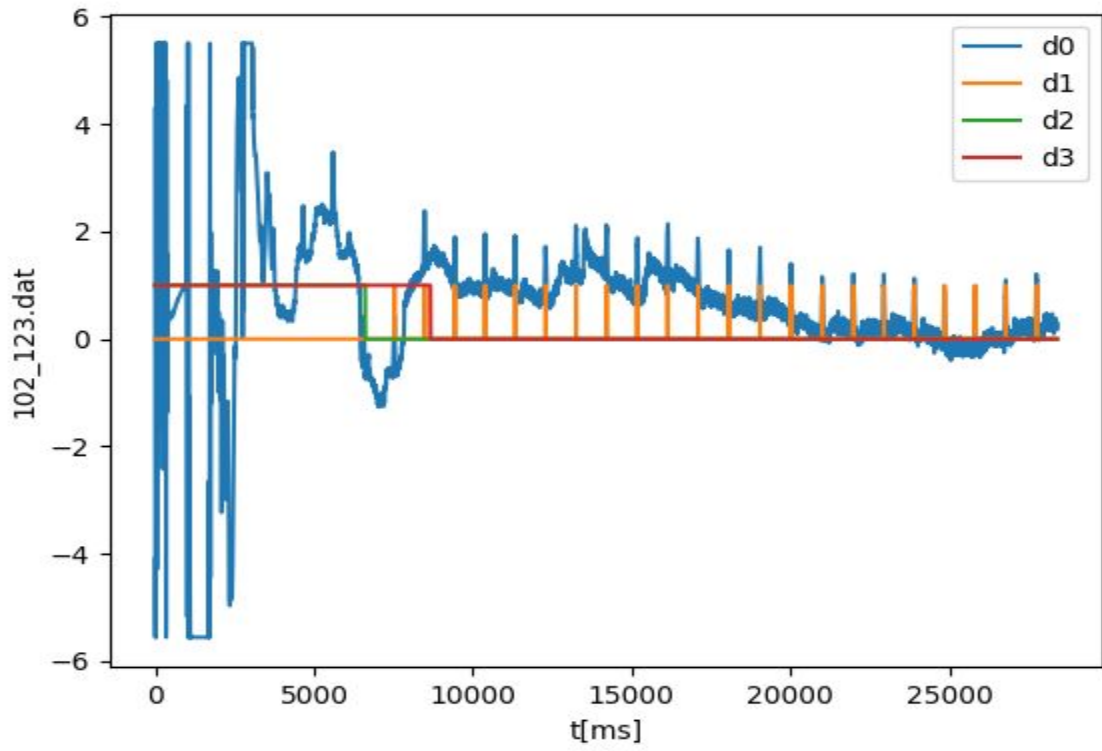
## 2 arduino data tree

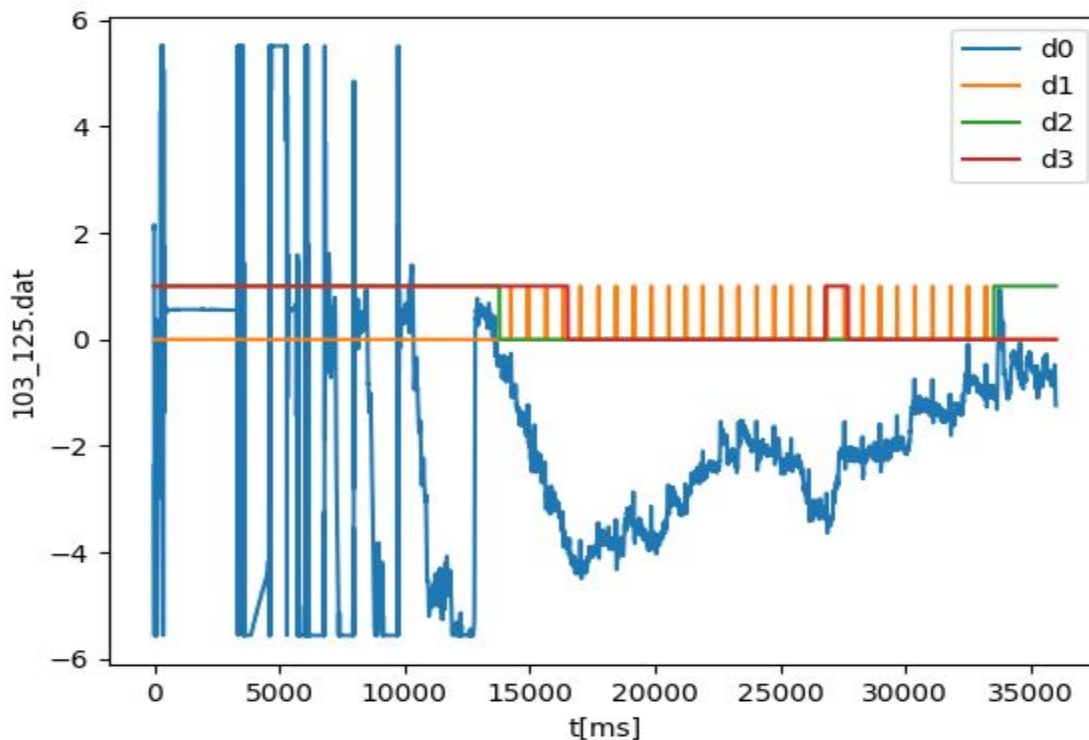
```
plt.show()
```

## DATEN VON TELE ECG









## ANDERE DATEN

Viele sind binär abgelegt.

physionet hat eine Anzeige der Kurvenform dabei  
<https://physionet.org/lightwave/?db=apnea-ecg/1.0.0> .

Das binär-Format scheint hier [https://archive.physionet.org/faq.shtml#file\\_types](https://archive.physionet.org/faq.shtml#file_types) kurz beschrieben.

### MIT Format

- MIT Signal files ([.dat](#)) are binary files containing samples of digitized signals. These store the waveforms, but they cannot be interpreted properly without their corresponding header files. These files are in the form: *RECORDNAME.dat*.
- MIT Header files ([.hea](#)) are short text files that describe the contents of associated signal files. These files are in the form: *RECORDNAME.hea*.
- MIT Annotation files are binary files containing annotations (labels that

generally refer to specific samples in associated signal files). Annotation files should be read with their associated header files. If you see files in a directory called *RECORDNAME.dat*, or *RECORDNAME.heg*, any other file with the same name but different extension, for example *RECORDNAME.atr*, is an annotation file for that record.

#### European Data Format (EDF)

- [EDF](#) files contain digital signals stored in their standard international format. EDF files store their header information at the beginning of the file, as opposed to MIT format which has a separate header file. Since recent versions of the WFDB library can read them directly, EDF is a WFDB and PhysioBank-compatible format. EDF files may also have associated annotation files. For example if a directory contains *RECORDNAME.edf* and *RECORDNAME.edf.qrs*, the *.qrs* file is the annotation file associated with the record.
- [EDF+](#) files are EDF files that also contain annotations encoded as signals.

Um die Daten in python zu verwenden gibt es hier

<https://github.com/MIT-LCP/wfdb-python> Code.

Tutorial dazu <https://github.com/MIT-LCP/wfdb-python/blob/master/demo.ipynb>

## REFERENCES

- Matplotlib: <https://matplotlib.org/tutorials/introductory/pyplot.html>
- python csv: <https://docs.python.org/3/library/csv.html>
- "Time and tide waits for no man" (Quote from Shakespeare, Julius Caesar)
- [TELE ECG Database: 250 telehealth ECG records](#)

## REVISION HISTORY

- 0.1 16.12.2020 - sys.argv