Slider 2

Otahal, Holecek, Jevtic,

Sprint 3

Sprintdokumentation

14.04.-28.04.2016

Inhaltsverzeichnis

[1 Sprintbericht 3](#_Toc449611285)

[1.1 Tasks 3](#_Toc449611286)

[2 GUI mittels Python programmieren 5](#_Toc449611287)

[2.1 Prototyp 6](#_Toc449611288)

[3 SQLite Datenbank 5](#_Toc449611289)

[3.1 Warum SQLite 5](#_Toc449611290)

[3.2 Installation 5](#_Toc449611291)

[3.3 Erstellen einer SQLite3 Datenbank 5](#_Toc449611292)

[4 Überarbeitung eines Webinterfaces 11](#_Toc449611293)

[4.1 Einleitung 11](#_Toc449611294)

[5 Burndown Charts 13](#_Toc449611295)

[5.1 Sprint Burndown Chart 13](#_Toc449611296)

[5.2 Product Burndown Chart 13](#_Toc449611297)

# Sprintbericht

## Tasks

### Von Sprint übernommen

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Name | Zugewiesen | Akzeptanzkriterium | anfängl.  Schätz. | verstri-chen | verblei-  bend | Status |
| 1335 | GUI mittels Python programmieren | MO | GUI muss funktionstüchtig sein | 20 hrs |  |  | Nicht fertiggestellt |
| 1283 | Prototyp von Grafischer Live Darstellung | RH | GUI soll mittels den Testdaten angezeigt werden | 6 hrs |  |  | fertiggestellt |
| 1273 | Datenbank am Beaglebone aufsetzen | BJ | Zugriff auf Server möglich | 6 hrs |  |  | Nicht fertiggestellt |
| 1333 | Überarbeitung des Webinterfaces | BJ | Webinterface wird angezeigt | 4 hrs |  |  | fertiggestellt |

### Neu hinzugefügt

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Name | Zugewiesen | Akzeptanzkriterium | anfängl.  Schätz. | verstri-chen | verblei-  bend | Status |
| 1271 | Serielle Kommunikation herstellen | MO | Testdaten |  |  |  | Nicht fertiggestellt |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

# SQLite Datenbank

## Warum SQLite

Unser Projektteam hat sich für das DBMS SQLite wegen den Folgenden punkten entschieden:

* SQLite benötigt viel weniger Ressourcen als andere herkömmliche DBMS. Dies ist wichtig da der Beagle Bone nicht viel Rechenleistung zur Verfügung stellen kann.
* Es ist File basiert (die ganze Datenbank besteht aus 1. File) und ist damit sehr Portable.
* Es ist zum Testen von Daten geeignet da es nicht zusätzliche Prozesse für diesen Vorgang benötigt.

## Installation

In unserem Fall wird die SQLite Version 3 verwendet. Um den Download der .tar.gz Datei zu initialisieren wird der Befehl

wget http://www.sqlite.org/sqlite-autoconf-3070603.tar.gzverwendet. Darauffolgend wird die jetzt heruntergeladene Datei mit dem tar Befehl

tar xvfz sqlite-autoconf-3070603.tar.gz

die Datei entpackt. Danach wird in eines der extrahierten Verzeichnisse mit dem Befehl cd sqlite-autoconf-3070603gewechselt. Hier befindet sich dann die Datei die mit dem ./configureBefehl in ein make fähiges gemacht wird. Mit den Befehlen makeund darauffolgend make installwird dann SQLite3 installiert.

Nachdem die makeBefehle ausgeführt werden erschient ein Output der besagt dass die SQLite3 Binaries im Verzeichnis /usr/local/bininstalliert werden.

## Erstellen einer SQLite3 Datenbank

Für das Projekt wird eine Sehr simple Datenbank erstellt die die Peak Werte der Differenz der Beiden Laser speichert. Die Datenbank wird mit den Befehl ***sqlite3 beaglebone*** erstellt.

Es wird dann ein Table mit dem namen ***PEAK*** mit dem Datentypen ***INTEGER*** mit dem Befehl ***create table PEAK(value INTEGER)*** erstellt.

Die Tabelle wird dann den folgenden Testdaten gefüllt: ***Insert into PEAK value(5)***

# GUI mittels Python programmieren

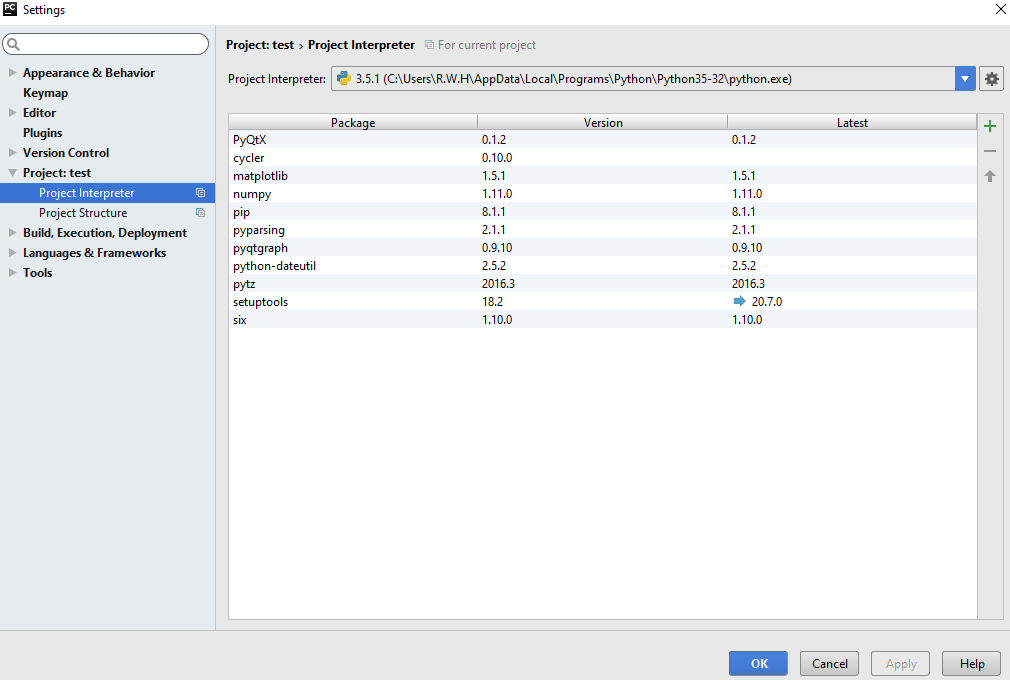
## Prototyp

Der erste Gedanke der zur Konzipierung der GUI einfiel war sie in python(Pycharm) selber zu realisieren. Dies wurde mit den Modulen Tkinter (GUI) und Matplotlib(Graph) ermöglicht. Als IDE wurde eine Windows x64 version von Netbeans Pycharm verwendet.

### Installieren der Module

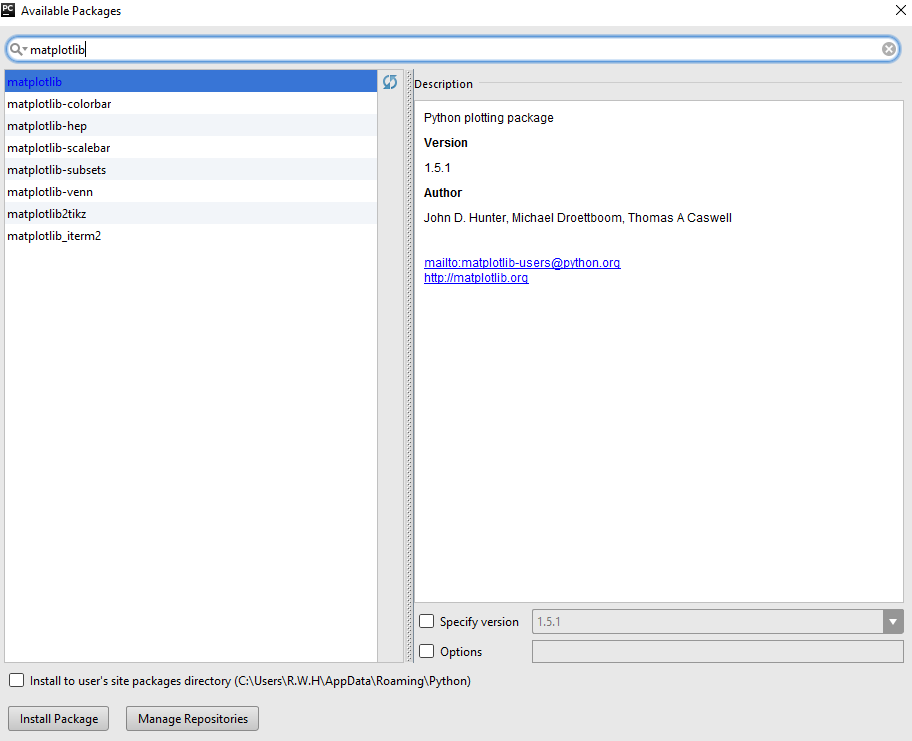
Tkinter ist bei Python 2.7 per default dabei und wird mit den Befehl *from tkinter import \** in eine Klasse importiert. Wenn man nicht den Konstruktor von Tkinter direkt aufrufen will wenn man ihm eine Variable zuweist, kann man den Befehl *import tkinter as varname* verwenden*.*

Das Modul Matplotlib hingegen muss mittels pip noch extra installiert werden. In Pycharm wird die installation über pip mit einer GUI erleichtert. Diese wird in *File -> Settings -> Project -> Project Interpreter* gefunden. Zu der Suche der verschiedenen Module gelangt man über das Betätigen des *+ Buttons.*



Nach der Betätigung erfolgt die Öffnung des folgenden Fensters.

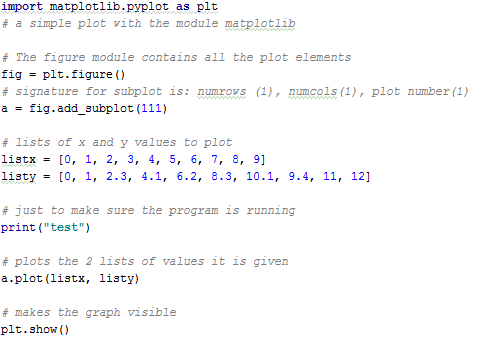
In diesem können mittels des Suchfeldes alle möglichen Module gefunden und mit betätigung des *Install Package* Buttons dann installiert werden. In unserem fall sind während der Installation keine Fehler aufgetreten.



Danach kann matplotlib in einer beliebigen Klasse mit dem Befehl *import matplotlib* verwendet werden.

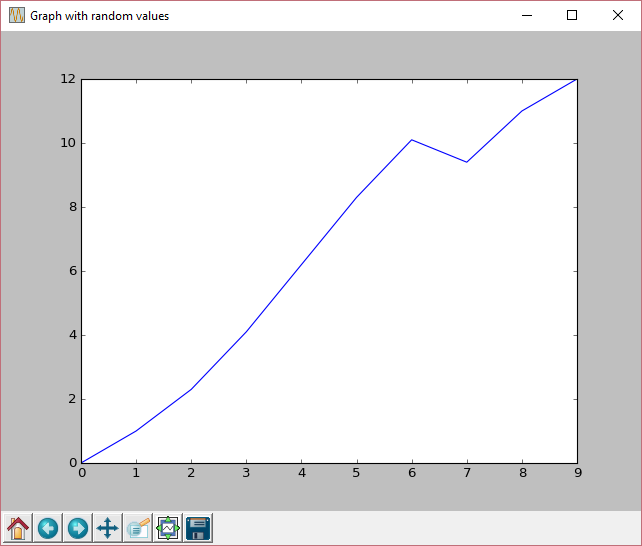
### Matlplotlib Probeklasse

In der folgenden gut dokumentierten Klasse wird gezeigt wie man ein einfaches Programm mittels matplotlib erstellt:



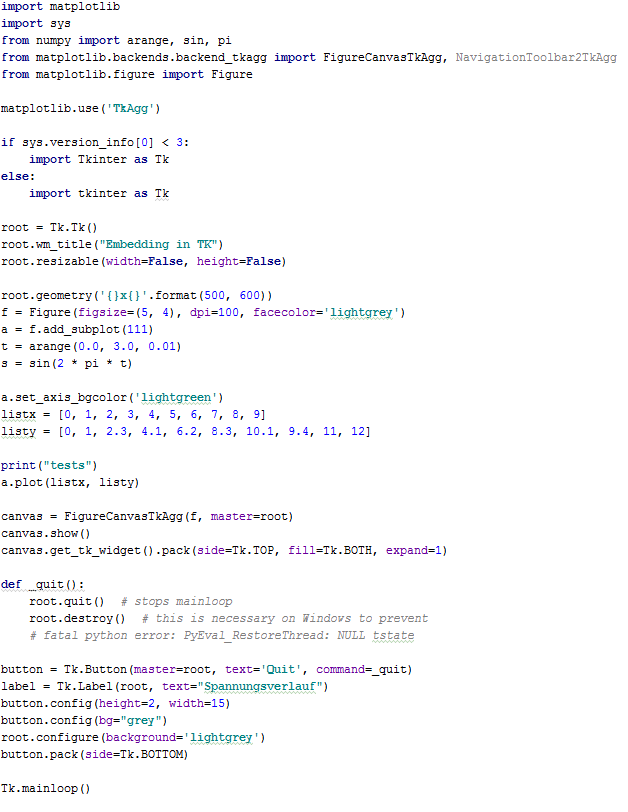
Die ausgegebene GUI sieht wie gefolgt aus:

Die Buttons im unteren Teil des Fensters sind Standardmäßig bei einem Plot dabei.

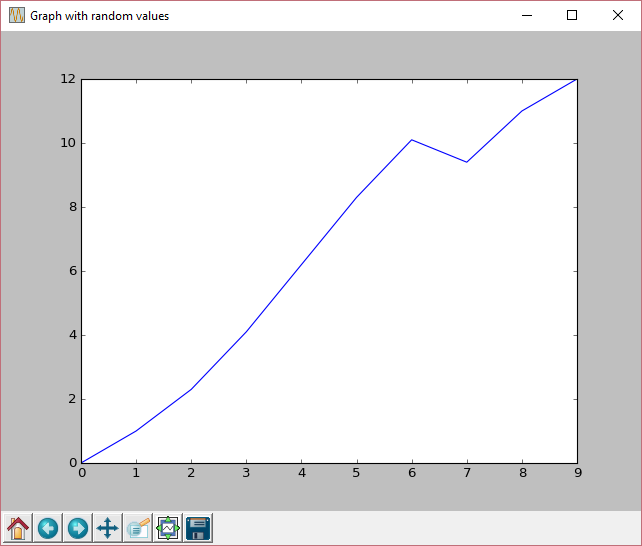


### Embedden in TKinter

Die nächste Aufgabe war die veraltet aussehen Buttons mit selbst erstellten zu ersetzen. Nach einer Recherche hat sich die Folgende Lösung ergeben:



Die Ausgabe dieses Codes sieht wie gefolgt aus:



Die Buttons im unteren Teil wurden durch einen Button des Moduls Tkinter ersetzt. Eigentlich wird gehört der ganze südliche Teil zu Tkinter.

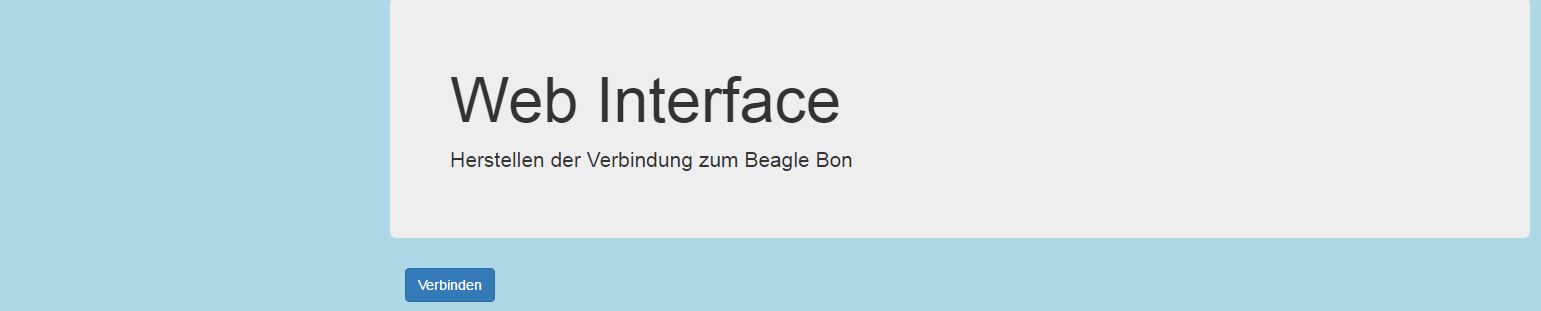
# Überarbeitung eines Webinterfaces

## Einleitung

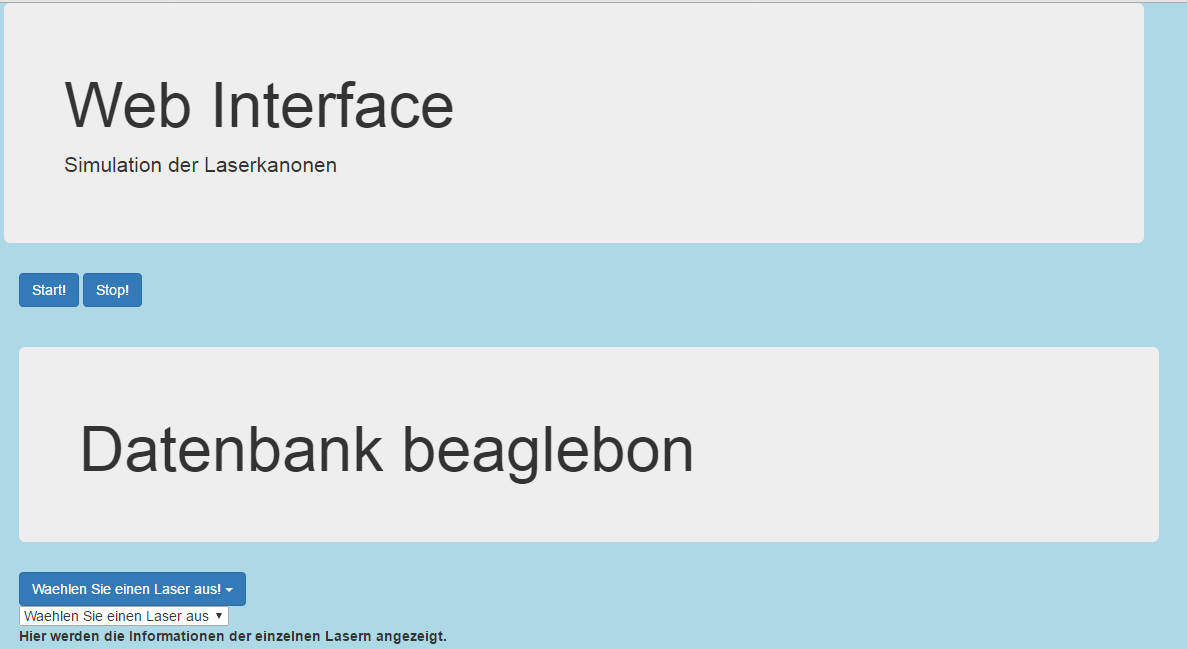
Das Webinterface muss noch entsprechend der Benutzerfreundlichkeit überarbeitet werden. Dabei wird das Webdesign verbessert und verschönert. Jedoch ist die Überarbeitung der Webschnittstelle noch nicht vollständig abgeschlossen, weil die Dropdown Liste (dort, wo die einzelne Laser ausgewählt werden sollen) noch bearbeitet werden muss. Das Webinterface wird noch entsprechend angepasst, um auch wirklich die Verbindung mit dem Beagle Bon herzustellen.

Beispielsweise wurden die Überschriften, Unterüberschriften und Buttons verschönert. Dafür wurde das freie CSS Framework Bootstrap für die Programmierung verwendet, um das Design etwas mehr zu verbessern. Die Hintergrundfarbe ist jedoch gleich geblieben.

connection.html



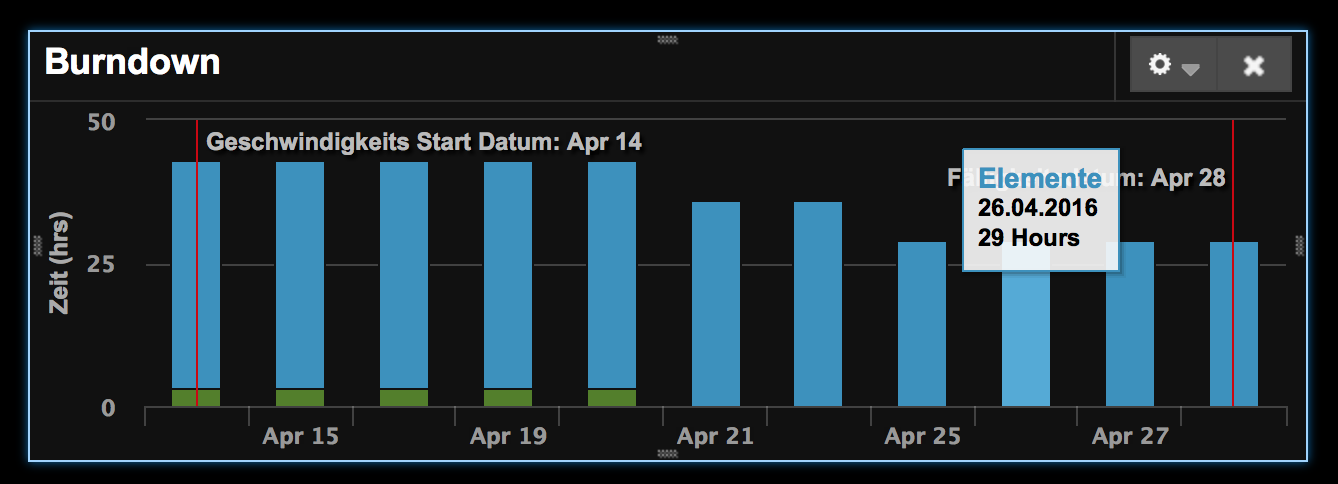
webinterface.html



Als eines der weiteren Verbesserungsvorschläge wäre noch, ein schönes Template zu finden und dementsprechend anzupassen.

# Burndown Charts

## Sprint Burndown Chart



## Product Burndown Chart

