

1 Software auf dem RaspberryPi

Der RaspberryPi liest die vom Arduino gemessenen Daten über die serielle Schnittstelle ein, speichert sie und stellt sie in einer Website dar. Für die Datenverarbeitung wird Python 3.6 verwendet und das Programm piapp wird mit xvpf-run ausgeführt, da dem Programm im headless-Modus kein X-server zur Verfügung steht.

Der RaspberryPi wurde verwendet, weil er ein W-Lan-Modul und USB-Schnittstellen schon eingebaut hat und einen Webserver betreiben kann. Unter dem Strich war es günstiger ein RaspberryPi zu verwenden, als ein einfacheres Board mit den benötigten Modulen auszurüsten. Zudem ist die Softwareentwicklung auf dem RaspberryPi erleichtert, da er ein volles Linux-Betriebssystem hat, mit dem sich ein Teil des Teams schon auskannte.

1.1 Datenverarbeitung

Die Datenverarbeitung wird mit dem Programm 'piapp' durchgeführt. 'piapp' wurde in Python 3.6 programmiert und kann von der Linux-Comandline aufgerufen werden. Damit 'piapp' automatisch nach dem Boot-Prozess ausgeführt wird, wurde es zu den /etc/init.d/-Skripts hinzugefügt.

Da 'piapp' ohne einem X-Display-Server funktionieren muss, nutzt 'piapp' das Programm xvpf-run. Xvpf-run wurde aus dem Raspbian-Repository installiert.

1.1.1 main.py

Im Modul main.py befindet sich der Einstiegspunkt des Skripts. Hier werden Argumente und Optionen des Commandline-Interfaces eingelesen und verarbeitet.

Der Commandline-Befehl 'piapp' hat mehrere Argumente und Optionen, mit denen bestimmt werden kann, wie 'piapp' sich verhalten soll.

- data_base bestimmt welche Datenbank erweitert werden soll.
 Generiert eine neue Datenbank wenn keine Angaben gemacht werden.
- plot_path bestimmt wo die generierten Plots gespeichert werden soll.
- out_path bestimmt wo die generierte HTML-Seite gespeichert werden soll.
- plot_interval bestimmt wie oft geplottet werden soll.
 Werden keine Angaben gemacht wird alle 100 Samples geplottet.
- buffer_size bestimmt wie gross der im RAM gespeicherte Cache sein soll.
 Werden keine Angaben gemacht, umfasst der Cache 1000 Samples.

Die Funktion main enthält den Hauptprogrammfluss, der die im Arduino gemessenen Daten einliest, darstellt und speichert. In per Commandline-Option definierten Intervallen werden Plots der Daten im Cache und Plots der Daten im Flashdrive gemacht.

1.1.2 __init__.py

Das Modul __init__.py stellt das Objekt DataPoint bereit, das die Messwerte einer Messung beinhaltet.

Die Funktion make_text_file schreibt die Werte einer Liste von DataPoints in ein Text-File.

Die Funktion read_text_file liest die Werte einer Liste von DataPoints aus einem Text-File.

1.1.3 input.py

Das Modul input.py stellt alle für das Einlesen benötigten Funktionen bereit.

Die Funktion init_devs initialisiert die serielle Schnittstelle. Es probiert alle USB devices aus, bis es ein geeignetes Gerät gefunden hat.

Die Funktion request_data liest ein String über die serielle Schnittstelle ein. Es parsed den String mit einer Regex, die Spannung, Strom, Schein-, Wirk-, Blindleistung

1.1.4 output.py

Das Modul output.py stellt alle für das Darstellen der Daten benötigten Funktionen bereit.

Die Funktion make_plot generiert ein .png-file, in dem die gegebenen Daten aufgezeichnet sind. Es werden dabei zwei Plots gemacht, einen für Strom und Spannung und einen für Schein-, Wirk- und Blindleistung.

Die Funktion make_html generiert ein .html-file, das zwei Plots und eine Liste der schon vorhandenen Daten enthält.

Die Funktion make_table generiert eine html-Tabelle mit allen Daten, die in früher gemessen wurden.

Die Funktion make_text konvertiert eine Liste von Messwerten in ein Text-file, das zum Download bereitgestellt werden kann.

1.2 W-Lan Hotspot

Damit die Website mit den Daten vom Nutzer gesehen werden kann, hat der RaspberryPi ein eigenes W-Lan-Netzwerk mit der SSID WiPi. Zudem kann der RaspberryPi mit einem Ethernetkabel mit einem Router mit Internetanschluss verbunden werden, sollte der RaspberryPi Internet benötigen.

Für den W-Lan-hotspot und DHCP-Server wurde `hostapd`, `hostap-utils`, `iw` und `bridge-utils` aus dem Raspbian-Repository installiert.

1.3 Webserver

Die Website mit den Daten wird von einem Apache2-Server gehostet. Der Apache2-Webserver wurde aus dem Raspbian-Repository installiert und mit symbolischen Links so konfiguriert, dass er zwei Plots und eine Liste von alten Messdaten auf seiner Homepage darstellt.

Die Homepage kann aufgerufen werden, indem man die IP-Adresse des Routers in einem Webbrowser eingibt. Ist man direkt über den W-Lan-hotspot des RaspberryPis verbunden lautet die IP-Adresse 192.168.42.1, ist der RaspberryPi per Ethernet an einem Router angeschlossen, wird die IP-Adresse vom Router bestimmt.