Web Based Altimeter

Introduction

This document provides installation and configuration instructions for the altimeter web application. The app features a convenient way to view charts of altitude, barometric pressure, and temperature for any calendar period for which data points are stored in the database. The app demonstrates the use of Python, Javascript, PHP, HTML, Rrdtool, and how to hookup an I2C serial i/o device to the Raspberry Pi's GPIO. The app also demonstrates the use of a round-robin database (RRD) that can be configured for any depth, limited only by file system storage capacity. The altimeter data can be accessed by any client with a web browser that is on the network.

Referring to figure 1, the Raspberry Pi boot process starts the altimeter agent process. The agent process performs a number of tasks including: continuously reads the altimeter sensor, writes the data to a round-robin database, writes the data to a JSON file, and periodically generates charts of the data. When the client browser loads the web page, Javascript periodically requests the output data file and chart files created by the altimeter agent process. Additionally a PHP script running on the web server generates charts with custom, user defined date ranges.

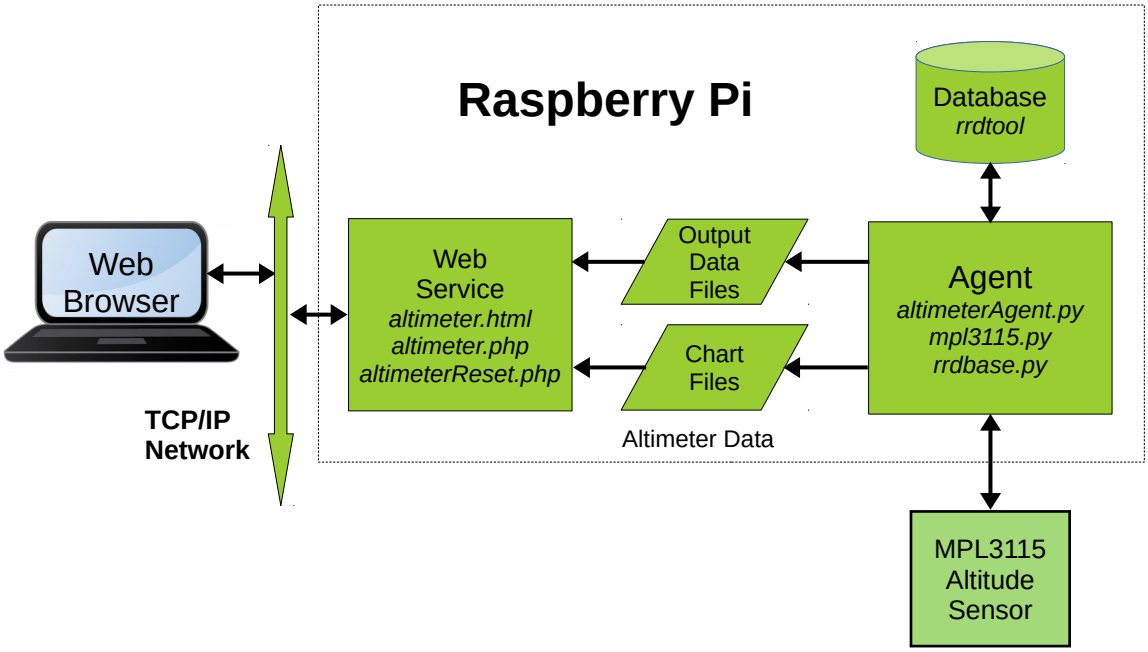

Figure 1. Overall conceptual view of node power sensor on AREDN mesh network.

Hardware Hookup

Referring to figure 2, the MPL3115 breakout pin VCC is connected 3.3 Volts via header pin 1, while breakout pin GND connected to ground via header pin 6. Breakout serial clock (SCL) is connected to GPIO SCL, which is

GPIO pin 3 (pin 5 on the header); and breakout serial data (SDA) is connected to GPIO SDA, which is GPIO pin 2 (pin 3 on the header).

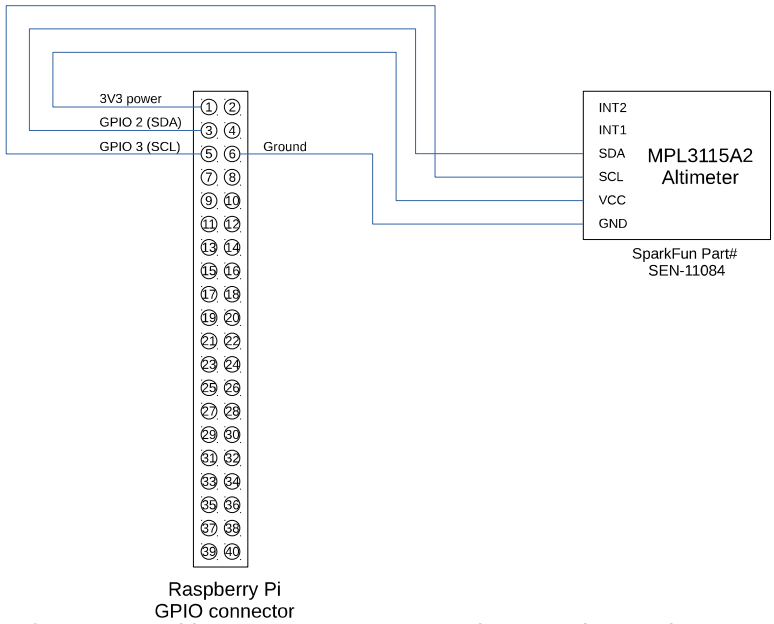

Figure 2. Schematic of MPL3115 altimeter sensor connected to Raspberry Pi GPIO.

Software Description

Referring to figure 1, the software consists of essentially of two components: a web server component and an agent component. The web server component includes HTML documents containing Javascript. The agent, a Python script, requests data from the altimeter sensor, manages data conversion and re-formatting, updates a database, and generates stock charts. Events flow in the following manner.

The agent periodically sends a request for data to the altimeter sensor. The agent then converts specific data items to other formats where required. Selected data items get written to a round-robin database for permanent storage. Besides these functions, the agent manages generation of a set of stock charts for display in HTML documents. After formatting the altimeter data, the agent writes the data to the Output Data File for use by HTML documents.

When a client browser requests the HTML document, Javascript embedded in the document reads the output data file and displays this data in an HTML document. The agent, as mentioned earlier, generates graphic charts for display in HTML documents. The graphic charts are stored as image files which Javascript in the HTML documents can load and display in the web page.

The MPL3115 Module

The *mpl3115.py* module acts as a hardware abstraction layer providing a software interface between higher level calling routines and the MPL3115 device. The module defines a class containing all the necessary class methods to read and write the MPL3115 altimeter. Class methods handle conversion and digital formatting of input and output data.

The Rrdbase Module

The *rrdbase.py* module acts as a software abstraction layer providing a software interface between higher level calling routines and the rrdtool command line utility. The module handles updating of the rrdtool database and

the generation of charts from data in the database. Class methods handle updating the database and generation of charts.

Software Installation

Important notes:

1. The altimeter application should be installed on a Linux host which meets the following requirements:

    - The server software should be installed on a recent distribution such as Debian or Raspberry Pi OS.
    - Apache 2 should be installed and configured to allow serving HTML documents from the user's public_html folder.
    - PHP 7 should be installed and configured to allow running PHP scripts from the user pi public_html folder.
    - Rrdtool should be installed (see instructions).
    - The Python I2C smbus library should be installed.
    - Python 3 usually comes per-installed in virtually all Linux distributions. Type "python3" at a command line prompt to verify Python has been installed.

2. See *Setting Up the Raspberry Pi for Internet of Things* hookup guide for detailed notes on how to set up a Raspberry Pi server, and how to install and configure the components in note 1 above.

Software Inventory

The following software items from the repository need to be installed on the Raspberry Pi

altimeter html folder:

- altimeter.html
- altimeter.php
- altimeterReset.php
- index.html

altimeter bin folder:

- altimeterAgent.py
- altimeterReset
- altstart
- altstop
- createAltimeterRrd.py
- mpl3115.py
- rrdbase.py

Installing on a Raspberry Pi

Note that the following installation procedure assumes that the document root for HTML documents will be the user's public_html folder. Typically the full path name to this folder will be something like **/home/{user}/public_html**, where **{user}** is the name of the user account hosting node power. The following steps will assume the altimeter app is running under the **pi** user account. The **pi** account is the default account when Raspberry Pi OS is first installed on a Raspberry Pi.

1. Follow the instructions in *Setting Up the Raspberry Pi for Internet of Things* hookup guide to install and configure web services on your Raspberry Pi.

2. If it doesn't already exist, use **mkdir** to create a folder **public_html** in the **pi** user's home folder. The full path should look like **/home/pi/public_html**.

3. In the p**ublic_html** folder, use **mkdir** to create a folder called **altimeter** to contain the altimeter HTML and PHP files. The full path should look like **/home/pi/public_html/altimeter/**.

4. Copy all the contents of the repository **html/altimeter** folder to the **altimeter** folder created in step 3.

5. Copy all the contents of the repository **altimeter/bin** folder to the **bin** folder in the user pi home folder. The full path to this folder should look like **/home/pi/bin/**.

6. Create a folder in the temporary file system for dynamic content, and modify the folder's ownership and permissions to allow the Apache www-data user to have access.

   mkdir /tmp/altimeter
   sudo chown :www-data /tmp/altimeter
   chmod g+w /tmp/altimeter

7. The commands in step 6 may be placed in a startup shell script and run at boot up time by launching a startup script with the **su** command from **/etc/rc.local**. For example, place the above commands in a script **/home/pi/bin/startup.sh** and place the following line in the **/etc/rc.local** file.

   (su - pi -c "~/bin/startup.sh")&

   Whenever the host boots up, the **startup.sh** script will run the commands in step 6. Be sure that you grant **startup.sh** execute permissions by running

   chmod u+x ~/bin/startup.sh

8. In the **altimeter** folder created in step 3, create a symbolic link to **/tmp/altimeter** in the temporary file system by running

   ln -s /tmp/altimeter dynamic

9. In the user home folder, create a folder named **database**. The full path to this folder should look like **/home/pi/database**. In the **bin** folder run the python script **createAltimeterRrd.py**. Running this script creates an empty round robin database file where the agent will periodically store the altimeter data. This script should be run once and then kept in a secure place. Running it accidentally at some future date will result in *total loss of all previously stored data*.

10. In the user's home folder create a folder named **log**. This is where the agent will keep its error logs.

11. Acting as superuser open **/etc/sudoers** in a text editor. Add the following line to the end of the file

    ```
    www-data ALL=(ALL) NOPASSWD: /home/pi/bin/altimeterReset
    ```

    Adding this line gives the Apache www-data user permission to execute the *altimeterReset* shell script via the *altimeter.php* script.

12. For convenience two scripts have been provided to make it easy to turn the agent on and off. The **altstart** script starts up the agent and causes all diagnostic output and error messages to be written to a log file in the log folder. The **altstop** script stops the agent from running. Start the altimeter agent by running **altstart**. Alternatively the **altstart** command can be placed in the **startup.sh** script mentioned in

step 7 to automatically start the agent when the host boots up.

13. Open the Raspberry Pi home page in a browser running on a computer connected to the same network as the Pi. Click on the “Altimeter” link to open the altimeter web page. You should see the current altitude, barometric pressure, and temperature data, as well as the charts.

This completes installation of the function generator software.