Appendix A

Data collecting software installation guide

No previous research described the experiment design in detail, to aid in research dissemination this step-by-step guide covering how to install the data collecting system used in this thesis was written. There are often many ways of installing and running software. The steps described in this appendix are the ones that worked for the author, and hopefully, they will work for the reader as well. Good Luck!

1. Set up the Plugwise system on a Windows machine to make sure the plugs and the Plugwise Stick is working correctly.

2. Make a spreadsheet of the MAC-addresses and corresponding given plug numbers, names, locations, type and a short version of the MAC A.1.

MAC	Plug number	Name	Location	Type	Short MAC
000D6F0003BD6969	01	Electric heater Andre	Bedroom	Circle	BD6969
000D6F0003BD8974	02	Water Heater	Shed/Storeroom	Circle	BD8974
000D6F0002C0DDDB	03	Oven	Kitchen	Circle	C0DDDB
000D6F000416DD83	04	TV	Livingroom	Circle	16DD83
000D6F0003BD5FE2	05	Coffee Maker	Shed/Storeroom	Circle	BD5FE2
000D6F0003561FC7	06	Electric kettle	Kitchen	Circle	561FC7
000D6F0004B1EC41	07	Electric heater salong	Livingroom	Circle	B1EC41
000D6F0002C0E746	08	Microwave oven	Kitchen	Circle	C0E746
000D6F0000469B3E	09	Electric heater terrace	Livingroom	Circle+	469B3E
000D6F0004B613A8	10	Dishwasher	Kitchen	Circle	B613A8
000D6F0004A29FA3	11	Refrigerator	Kitchen	Circle	A29FA3
000D6F0002C0EFF3	12	Washing machine	Shed/Storeroom	Circle	C0EFF3

Table A.1: Smart Plug details in experiment.

3. Use a Raspberry Pi with a correct power supply to make sure it runs as it should. If a power supply with lower power is used, the CPU will be throttled.

4. Use a high class SD card that will not throttle the Raspberry Pi's performance.

5. Download an operating system for the Raspberry Pi. In my case i used the Raspian Stretch Desktop (https://www.raspberrypi.org/downloads/raspbian/)
6. Flash the SD-card with the OS just downloaded using Etcher (https://www.balena.io/etcher/)
7. Plug the SD-card into the Raspberry Pi and connect a keyboard, mouse, Ethernet internet or WiFi dongle and a display for initial setup.
8. In the desktop menu, open "Raspberry Pi Configuration". Under "Interfaces", enable SSH and VNC for best remote access.You should also change the default password to a more secure one.
9. Download and install RealVNC (`https://www.realvnc.com/en/connect/download/vnc/`). Open RealVNC and configure a remote connection to the Raspberry Pi. Since I have not configured a static IP-address I made a free account and made the Raspberry Pi accessible through their cloud service.
10. To connect to the Raspberry Pi within your local network, go to terminal and type one of the two commands:

    ```
    $ ssh pi@<raspberry ip>
    $ ssh pi@raspberrypi.local
    ```

    Then log in with your password.
11. To enable file sharing on the Raspberry Pi, follow this guide (`http://raspberrypituts.com/access-raspberry-pi-files-in-your-os-x-finder/`)

    ```
    $ sudo apt-get install netatalk
    $ ifconfig
    $ open afp://192.168.0.10
    ```
12. Plugwise-2-py (https://github.com/SevenW/Plugwise-2-py)
13. Installing the software. To make it easier for yourself, install it in the /home/pi folder.

    ```
    $ sudo python get-pip.py
    $ git clone https://github.com/SevenW/Plugwise-2-py.git
    $ cd Plugwise-2-py
    $ sudo pip install .
    ```
14. Moving the config-files. From Plugwise-2-py folder, copy the config-files to the main folder:

    ```
    $ cp -n config-default/pw-hostconfig.json config/
    $ cp -n config-default/pw-control.json config/
    $ cp -n config-default/pw-conf.json config/
    ```
15. Configuring pw-hostconfig.json:

- Open the config/pw-hostconfig.json and check if it has the correct paths.
- If Plugwise-2-py was installed in the /home/pi the file should be contain the right path:

```
{"permanent_path":"/home/pi/datalog","tmp_path":"/tmp","log_path":"/home/pi/pwlog",
"serial":"/dev/ttyUSB0","log_format":"epoch","mqtt_ip":"127.0.0.1","mqtt_port":"1883
```

Note that:

- "Serial" might not point to the Plugwise stick in all cases. Check therefore if this is correct for you.
- To enable MQTT messaging later the "log_format", "mqtt_ip" and "mqtt_port" is added.
- Editing JSON files is error-prone. Use a JSON Validator such as http://jsonlint.com/ to check the config files.

16. Configuring pw-conf.json:

    - Here you add the plugs values created in step 1.
    - Be sure to use the JSON Validator!
    - Example data:

```
{"static": [
{"mac":"000D6F0003BD5FE2","category":"Coffee-maker","name":"Coffee-maker",
"loginterval":"60","always_on":"False","production":"False","location":"Shed/Storeroom"},
{"mac":"000D6F0004B613A8","category":"Dishwasher","name":"Dishwasher",
"loginterval":"60","always_on":"False","production":"False","location":"Kitchen"},
{"mac":"000D6F0003BD6969","category":"Electric heater","name":"Electric heater Andre",
"loginterval":"60","always_on":"False","production":"False","location":"Bedroom Andre"},
{"mac":"000D6F0004B1EC41","category":"Electric heater","name":"Electric heater salong",
"loginterval":"60","always_on":"False","production":"False","location":"Living room"},
{"mac":"000D6F0000469B3E","category":"Electric heater","name":"Electric heater terrace",
"loginterval":"60","always_on":"False","production":"False","location":"Living room"},
{"mac":"000D6F0003561FC7","category":"Electric kettle","name":"Electric kettle",
"loginterval":"60","always_on":"False","production":"False","location":"Kitchen"},
{"mac":"000D6F0002C0E746","category":"Microwave oven","name":"Microwave oven",
"loginterval":"60","always_on":"False","production":"False","location":"Kitchen"},
{"mac":"000D6F0002C0DDDB","category":"Oven", "name":"Oven",
"loginterval":"60","always_on":"False","production":"False","location":"Kitchen"},
{"mac":"000D6F0004A29FA3","category":"Refrigerator","name":"Refrigerator",
"loginterval":"60","always_on":"False","production":"False","location":"Kitchen"},
{"mac":"000D6F000416DD83","category":"TV","name":"TV",
"loginterval":"60","always_on":"False","production":"False","location":"Living room"},
{"mac":"000D6F0002C0EFF3","category":"Washing machine","name":"Washing machine",
"loginterval":"60","always_on":"False","production":"False","location":"Shed/Storeroom"},
{"mac":"000D6F0003BD8974","category":"Water heater vessel","name":"Water heater vessel",
"loginterval":"60","always_on":"False","production":"False","location":"Shed/Storeroom"}
]}
```

17. Configuring pw-control.json: - Here you also plug in values from step 1. - Be sure to use the JSON Validator! - Example data:

```
{"dynamic": [
{"mac": "000D6F0003BD5FE2", "switch_state": "on", "name":"Coffee-maker",
"schedule_state": "off", "schedule": "", "savelog": "yes","monitor": "yes"},
{"mac": "000D6F0004B613A8", "switch_state": "on", "name":"Dishwasher",
"schedule_state": "off", "schedule": "", "savelog": "yes","monitor": "yes"},
{"mac": "000D6F0003BD6969", "switch_state": "on", "name":"Electric heater Andre",
"schedule_state": "off", "schedule": "", "savelog": "yes","monitor": "yes"},
{"mac": "000D6F0004B1EC41", "switch_state": "on", "name":"Electric heater salong",
"schedule_state": "off", "schedule": "", "savelog": "yes","monitor": "yes"},
{"mac": "000D6F0000469B3E", "switch_state": "on", "name":"Electric heater terrace",
"schedule_state": "off", "schedule": "", "savelog": "yes","monitor": "yes"},
{"mac": "000D6F0003561FC7", "switch_state": "on", "name":"Electric kettle",
"schedule_state": "off", "schedule": "", "savelog": "yes","monitor": "yes"},
{"mac": "000D6F0002C0E746", "switch_state": "on", "name":"Microwave oven",
"schedule_state": "off", "schedule": "", "savelog": "yes","monitor": "yes"},
{"mac": "000D6F0002C0DDDB", "switch_state": "on", "name":"Oven",
"schedule_state": "off", "schedule": "", "savelog": "yes","monitor": "yes"},
{"mac": "000D6F0004A29FA3", "switch_state": "on", "name":"Refrigerator",
"schedule_state": "off", "schedule": "", "savelog": "yes","monitor": "yes"},
{"mac": "000D6F000416DD83", "switch_state": "on", "name":"TV",
"schedule_state": "off", "schedule": "", "savelog": "yes","monitor": "yes"},
{"mac": "000D6F0002C0EFF3", "switch_state": "on", "name":"Washing machine",
"schedule_state": "off", "schedule": "", "savelog": "yes","monitor": "yes"},
{"mac": "000D6F0003BD8974", "switch_state": "on", "name":"Water heater vessel",
"schedule_state": "off", "schedule": "", "savelog": "yes","monitor": "yes"}
], "log_level": "info", "log_comm": "no"}
```

18. Running plugwise-2-py: - To run, type the following in terminal (Plugwise-2-py main directory):

    ```
    $ sudo python Plugwise-2.py
    ```

    - The first time it runs it collects the buffered data from the plugs. This might take minutes, or hours. You can watch the progress by tailing the log:

    ```
    $ tail -f /home/pi/pwlog/pw-logger.log
    ```

    or look at the files updating in the Plugwise-2-py folder.

19. To run the web interface, type in terminal (Plugwise-2-py main directory):

    ```
    $ sudo python Plugwise-2-web.py
    ```

    - You can access the web interface at `http://localhost:8000/pw2py.html`. I got the web interface running, but at this point it didn't update the values. The values showed up after implementing the MQTT service. - Beacuase of the changes in sockets, the smart plug values might not show up in Safari web browser until the protocol version in modified in swutil/HTTPWebSocketsHandler.py: After the line with

    ```
    "_opcode_pong = 0xa"
    ```

    add

    ```
    "protocol_version = 'HTTP/1.1'"
    ```

    and restart Plugwise-2-web.py

20. Check the log-files to see if you receive data - If you receive data and the system is initialized correctly the following log files should be updated:

    - /home/pi/datalog/2018/pwact/pwact-2018-10-25-000D6F0003BD6969.log (One file per plug per day)

- /home/pi/datalog/2018/pwlog/pw-000D6F0002C0E746.log (One file per device)
- /home/pi/datalog/pwlastlog.log (Contains the last values for each device)
- /home/pi/pwlog/pw-logger.log (Logs the logging, should say save log for each device)
- /home/pi/pwlog/pw-web.log (Logs the web-interface, should be filled with MQTT-messages after step XX)

21. Installing mosquitto

    - The terminal line below did not work for me and I installed using this guide: `http://mosquitto.org/blog/2013/01/mosquitto-debian-repository/`

    ```
    $ sudo apt-get install mosquitto
    ```

22. Installing Domoticz is not necessary to collect the smart meter data. The Domotics installation was a source to great frustration because the message handling in Plugwise-2-py. At the time of writing the thesis, none of the Domoticz versions 4.X worked and it is therefore necessary to Install a 3.X version of the software. The easiest way to find a previous version is to compile and make it yourself from their github repo `https://github.com/domoticz/domoticz`. You might also be able to find an installation file online at `https://egregius.be/2018/previous-domoticz-versions/`. When installed and running, Domoticz can be reached at: http://127.0.0.1:8080

23. Node-RED To send the values to Domoticz, you have to use Node-RED to relay the messages from Plugwise-2-py.

    (a) Install node.js
    `https://www.instructables.com/id/Install-Nodejs-and-Npm-on-Raspberry-Pi/`

    (b) Install Node-RED (urlhttps://nodered.org/docs/getting-started/installation)

    (c) Run Node-RED in terminal:

    ```
    $ node-red
    ```

    (d) Access at `http://<Raspberrypiip>:1880`

    (e) Configure the flow by going to the top right corner and choose "Import". The standard flow is found in /home/pi/Plugwise-2-py/domoticz/plugwise2py-domoticz.nodered. The modified version also saves to file is found here: `https://github.com/powermundsen/thesis`.

    (f) Deploy! You should now see the messages being sent in the terminal. To double check you can download a more graphical version like `https://mqttfx.jensd.de`.

    (g) To make Node-RED start at boot, do the following (`https://nodered.org/docs/hardware/raspberrypi`)

NOTE: Node-RED warns about errors when you try to deploy, these disappeared when I clicked on the nodes and changed the server address. However, this is not a problem because when set up correctly it will work either way.

NOTE 2: It became clear at a later point that the best way to log the consumption is not in Domoticz, but directly in Node-RED. I expanded the flow above to do this and it is saved as a new flow in the "Master" folder (scripts for ......) .

24. Connecting Domoticz and Node-RED
    (a) Follow the guide in /home/pi/Plugwise-2-py/domoticz/README.md
    (b) I have not yet gotten this connection to work but I can see that the messages are being sent out on the network.

- To change the sampling time, change the following line in /home/pi/Plugwise-2-py/Plugwise-2.py to the time you want:

```
proceed_at = ref + timedelta(seconds=(2 - ref.second%2), microseconds=-ref.microsecond)
```

NOTE: Even though the polling frequency was changed to 1 second, the values received was usually between 2-3 seconds apart. I think this is because Plugwise-2-py filters out duplicate messages.

- Make a backup of the Raspberry Pi disk:
  1. Check disk utility to see which disk the Raspberry Pi SD card has
  2. In terminal:

```
$ mkdir raspberry_backup
$ cd raspberry_backup/
$ sudo dd if=/dev/disk3 of=~/Desktop/raspberrypi.dmg
```

- Congratulations, the system should now be functional! When starting up, run the following commands (in separate terminal windows):

```
$ sudo python /home/pi/Plugwise-2-py/Plugwise-2.py
$ sudo python /home/pi/Plugwise-2-py/Plugwise-2-web.py
$ node-red
$ ./domoticz_3.8153_linux_armv7l/domoticz
```

Note: In this experiment, the Raspberry Pi rebooted itself periodically. This was circumvented by executing the commands on the Pi itself through VNC.