

1 Openremote API

Along with the Z-Wave Stack, RaZBerry also installs an instance of Openremote that can be used to remotely send commands to the RaZBerry that will forward them to the thermostat.

Openremote hosts a Webservice on the RaZBerry that can be used to send commands or retrieve information. Table 1 shows which commands have been used and the relative URLs to call them. The base URL is:

http://<RaZBerry IP>:8083/ZWaveAPI/Run/devices[<device number>]

Table 1: Used services and their relative URLs	
COMMAND	URL ADDITION
get current temperature	.instances[0].SensorMultiLevel.data[1].val.value
get current setpoint	.ThermostatSetPoint.data[1].setVal.value
set setpoint	.instances[0].ThermostatSetPoint.Set(1,<setpoint>)

For example, to set the current setpoint on device 2 that is reachable under the IP 1.2.3.4 to 22 degree Celsius, one would call the following URL:

http://1.2.3.4:8083/ZWaveAPI/Run/devices[2].instances[0].ThermostatSetPoint.Set(1,22)

2 The control code

The main task of the control module is to monitor the current temperature and setpoint, communicate those to the main optimiser, to receive suggested setpoints and to send these to the RaZBerry.

This is done in one main control loop shown in algorithm 1. The algorithm first retrieves the current setpoint and temperature from the thermostat (lines 2&3). This data then gets send to the server responsible to optimise the heating profile. After giving the server some time to do the optimisation (line 7), the controller asks for the new setpoint (line 9).

If the new setpoint is different from the current setpoint, it'll be sent to the thermostat, otherwise the loop repeats.

Algorithm 1 The main control loop

```
1: while True do
2:   get current setpoint
3:   get current temperature
4:
5:   tell server current setpoint
6:   tell server current temperature
7:   wait 60 seconds
8:
9:   receive new setpoint from server
10:
11:   if new setpoint != current setpoint then
12:     set new setpoint
13:   end if
14: end while
```
