**O-MI reference implementation: project definition**

1. **Introduction**

This is a project of Home Automation. Home Automation is the science of automating things within a house, room or other facility. It uses a wide range of different sensors for data input, and can implicitly react based on that data with automated adjustments of the environment and ecosystem.

**2. Goal of this project**

This project will be developed as a reference implementation for the Aalto University. The goal is to take one specific room in the University and ‘automate’ it as described in the introduction. This room can be used as an example of how the whole University could be controlled in the future. So you could call it a proof of concept, based on one can drive forward the technical infrastructure of the University.

**3. The Room**

The room that will be used as reference is within the Department of Computer Science of the Aalto University in Helsinki. The exact address is:

*Konemiehentie 2, room B126,**02150 Espoo*

**4. The system architecture**

There are multiple components, all together build the system. The core component is a O-MI node. O-MI is an Internet of Things data server. It implements the Open Messaging Interface as well as the Open data format for exchanging data and communicating with other components of the system. The O-MI node holds all the current data in a database. Agents can push data to the server and update the current state of the room, which is represented by the economic data delivered by the sensors. Agents can as well request those data and then decide to control hardware that is able to adjust the state of the room.

The input sensors are connected to multiple Raspberry Pi computers. On those computers there are running agents that keep track of the sensor inputs and push O-MI messages to the O-MI Node if there are data change events. So the O-MI node keeps up to date of the current room state.

The adjustment hardware is connected to various Raspberry Pi computers as well. The agents on those machines pull the required data from the O-MI node periodically to keep track of any changes that matter for the connected hardware. If the data lead to an neccessary action, the machine can react and control the adjustment hardware accordingly.

**4. INPUT: Used sensors**

To keep track of the room, there are a few different sensors listed below.

* Temperature and light sensors
* Occupancy sensor
* Humidity sensor
* CO2 sensor
* Power Consumption sensor
* Contact sensor for door monitoring

**4.1 Temperature, light, humidity sensors**



For temperature, light and humidity monitoring,

there are used multiSensors that combine two sensors

each.

They are connected to the Raspberry Pi via 1-Wire bus,

and are connected among themselves in a chain.

**4.5. Power consumption sensor**

**[to to]**

**4.6. Contact sensor for door monitoring**

**[to to]**

**5. Output: Used controlling hardware**

For being able to control the room, there are various pieces of adjustment hardware. The most used hardware are remote outlets. They can be turned on and off from a Raspberry Pi and so switch the plugged in electronic devices on and off. Electronic devices can be literally anything with a power plug, from a coffee maker over a TV until an air condition machine.

The used electronic devices in combination with the electronic outlets are:

* A fan to control the air flow in the room
* A standard lamp

**6. Mounting the 1Wire-Hardware on a Raspberry Pi**

* To install, follow this guide:

http://wiki.m.nu/index.php/OWFS\_with\_i2c\_support\_on\_Raspberry\_Pi\_(English\_version)#Installation\_of\_OWFS

* To actually mount the hardware, use this command:

sudo /opt/owfs/bin/owfs -u /dev/ttyUSB0 --allow\_other /mnt/1wire/

(you can put the command in /etc/rc.local for auto mounting on boot)

**7. Controlling Output**

As output controlling device are Allnet ALL3073WLAN outlets used. To use it successfully, there are a few configuration steps needed. This can be done in the outlet’s own web-interface. To access it, follow the instruction manual.

* First, open the web interface of the (factory resetted) outlet, choose language, and apply the default network settings without changing anything. Save inputs.
* Then go to Configuration/Device settings --> Remote control, enable both - read only and switching - and enter *aalto* as username and password. Save inputs.
* Next, go to Configuration/Web server and users --> SSH Server setting. Set an SSH password (default would be *aalto*)*.* Note that the SSH server setting is set to enabled. Save inputs.
* Copy the scripts from the folder ip\_management/client\_scripts/ to the outlet. The simplest way is to use SCP for that, but you can use SSH as well. Copy the script 'say\_hello.sh' to /root and the script 'S90\_sayhello' to /etc/init.d. Next, make sure both scripts are executable (chmod 775). These scripts will automatically broadcast the devices' IP address to the ip management server if it changes. To identify each of the outlets, the MAC address of the wired network adapter is used.
* After this step is done, go to Configuration/LAN settings and set the network settings to DHCP to automatically optain an IP address. The device will reboot, and from now on the device will notify the IP management server every time the IP address changes.
* **If used as wireless device:** 
  + Configure WLAN settings (SSID, password) too.
  + **!! CATUTION: wifi only works when lan settings is set to DHCP !!**
* Optional: set things like device name, description etc.  
    
    
  **================================================================**
* **IMPORTANT:** The IP management server needs to be in the same subnet as the outlets, otherwise the communication will be blocked by the Aalto firewall. If you face any problems, consider asking the IT guys.  
  **================================================================**
* On every new plug, you should print its MAC address to identify them. By now, the MAC addresses are just printed and fixed with scotch tape. The font type is Bitstream Vera Sans Mono, size 13, bold.