# Home Automation Code Camp 2016

# Heating and Ventilation Control System

Aigerim Zhalgasbekova, Chandara Chea, Giang Tran, Henrique Sarmento, Nhi Vo, Victor Araujo

(PERCCOM Cohort 3)

## Abstract

Most of the people have bad habits in using electricity or consuming energy in home/office environment such as leaving their house without turning off the lamps and computer/laptop, and opening the windows while the heating is still turned on. These bad habits lead to more energy waste and higher costs for the electricity bills. To tackle this problem, the smart heating and ventilation control system is implemented to ensure comfort in the apartment by adjust the radiators and ventilation system when the room temperature and humidity change. In addition, with the collected data from home automation system, we also implement the machine learning technique and application to encourage people change their bad behavior as well as save more energy.

## 1 Introduction

Holding on to undesirable routine of carelessly and unnecessarily leaving the light and electric devices on, and opening the windows while heating system working could be probably due to several reason such as as (1) People are lazy to turn off/on these electrical devices whenever they go out. (2) These systems are centralized controlled, so they are always kept running. (3) People are not aware of energy waste, sustainability. (4) People are usually forgetful though they know that they should turn devices off.

Home automation technology provides alternatives throughout the home or office based on time of day or occupancy of the room. The capability to manage and monitor the energy consumption could provide an effective solution to understand how people consume energy and where changes can be made to reduce the consumption.

In this report, we will discuss about our solution, implementation, result and future work which is included in the conclusion.

## 2 Solution

We present a smart home system that can automatically control the radiators and ventilation based on users' occupancy status in the room in order to ensure the comfort level in the room and save energy.

### 2.1 Scenario

- If users are inside the room (motion detected), in the daytime (6am to 12pm), and the windows are closed (door sensor in closed status), then keep the heating at normal level (comfort level based on temperature and humidity).
- During the night (12am to 6am), level down the heating in a certain level to keep warm enough and save energy; level up again at 6am.
- During the night (12am to 6am), but if users stay working late, then keep the heating at normal level for 15 minutes whenever there is motion detection.

- If users are outside of the apartment (no motion detection in 5 mins after the door opened and closed), then turn off the radiator.
- If users are inside the room (motion detected), but the windows are opened (door sensor in opened status), level down the heating.

Additionally, we use machine learning to improve the efficiency of our solution. In this case, the home automation system learns the human behavior from the schedule. FHEM is used to feed a data store accessible through an API for machine learning or creating other services on top of it. A neural network was trained to attempt to learn a fictional schedule.

- It can learn to lower the temperature when owners are away during their working hours.
- User experience can be improved by eliminating the need to manually program thermostats.
- The machine learning algorithms can be trained with other sources of information to improve context detection. For example, Google calendar data might be accessed with the consent of the user to predict the needs of the user in advance.

## 2.2 Technology and Accessed Method

**FHEM** is a home automation management software with different front-ends and Web interface integrated. It is used to automate several tasks in household such as controlling heating, switching lights on/off, logging events (temperature, humidity, door opening). FHEM can integrate with a large amount of home automation technologies like FS20, HomeMatic, KNX, ZWave, EnOcean, X10.

We installed FHEM on Raspberry PI with a perl interpreter and a HomeMatic Wireless Adapter LAN to access the sensors such as motion sensor, door/window sensor, temperature and humidity sensor. Our server uses fhem.cfg file to store configuration.

**HomeMatic** is a proprietary system for home automation system from the company eQ.3. It uses wireless protocol BidCos (Bidirectional Communication System) which special feature of this protocol is that the recipient of a message has to acknowledge (ACK). Moreover, HomeMatic supports authentication (AES) and encryption (only XOR operation).

HomeMatic system can be basically divided into the following classes of devices:

- Central and gateways (CCU, USB and LAN adapters for configuration)
- Transmitters and controllers (buttons, remote)
- Sensors (thermostats, motion detectors)
- Actuators (heating controller)
- Wired (roller shutter)

In our project, we also use a CCU and LAN configuration adapter for configuring direct links between transmitters and actuators. Indeed, the adapter allows to interact with sensors as it transmits messages in both directions.

## 2.3 Estimated Cost to Implement the Solution

| Device | Price |
|---|---|
| Raspberry PI | 32 EUR |
| HomeMatic Adapter LAN | 79,95 EUR |
| HomeMatic Motion sensor | 59,95 EUR |
| HomeMatic Door/window sensor | 32,99 EUR |

| | |
|---|---|
| HomeMatic Temperature and Humidity sensor | 53,99 EUR |
| HomeMatic Radiator Thermostat | 39,95 EUR |
| **Total** | **298,83 EUR** |

## 2.4 Setup

In our solution, Raspberry Pi plays role of a hardware for FHEM server. The Raspberry Pi and LAN adapter have been connected to a switch for getting an access to the same network. Due to this, sensors can communicate with the server sending messages through the adapter. Figure 1 illustrates the topology of our system and figure 2 is how the sensors will be implemented in the room.


*Figure 1 - Topology of our home automation system*

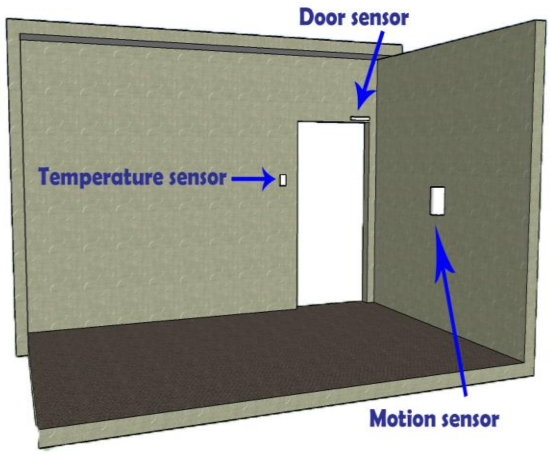


*Figure 2 - Installation of sensors in home environment*

# 3 Implementation

## 3.1 FHEM

Regarding the server implementation, initially a pseudo code was developed in C to represent the whole system. Therefore, it was possible to verify the specification of timers, events and variables that were going to be used in the system. After that the code was implemented in PERL and using FHEM commands.

Basically, the occupation is verified through events triggered by the motion and door sensor. When the door sensor is triggered, user is considered to be outside his apartment, and when motion sensor is triggered, user is assumed to be inside his house. After a certain period (15 to 30 minutes) if there is no movement inside the house, it is supposed that the user is either sleeping or not in home, so it is supposed that the user is not occupying the house.

After the occupation is verified there is a loop event which is triggered each 3 minutes in order to control the temperature of the house. If there is occupation, the heater will be configured to reach the desired temperature by increasing or decreasing the power of the heating system. If there is no one inside the apartment, the heater will continuously decrease its power until it is totally turned off.

Figure 3 describes the dataflow and communication between sensors, actuators, all devices used in the current system.

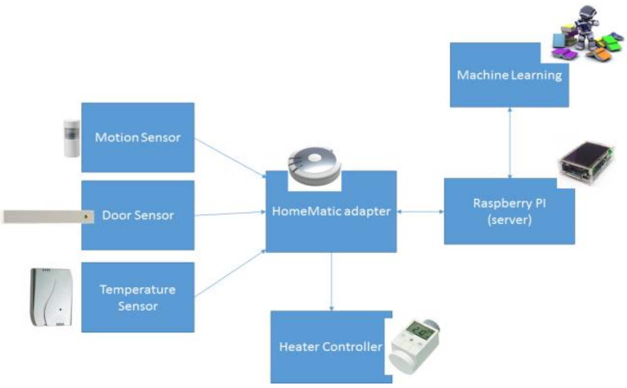


*Figure 3 - Data flow diagram of device and systems used in the current approach*

## 3.2 Machine Learning

We use an artificial neural network with two input neurons, a hidden layer with twelve neurons and 10 neurons in the output layer for classification. The model is trained using the structure of the API built for the project.

### The scenario

We tried to simulate a scenario with the following characteristics:

- Weekdays: 16 degrees during work hours (9-17). Else 21 degrees
- Saturdays: 25 degrees from 11 to 22. Else 19 degrees
- Sunday: 23 all day.
- Tuesdays and Thursdays: Return home from 13 to 15

The algorithm should be able to learn the changes of temperature during the schedules, in order to prevent the user from forgetting to turn down the temperature and to improve the user's experience.

### Training accuracy

For this dataset, accuracies range from ~70% to ~95%. This can of course be tweaked and improved given enough time. The model learns most of the features, including temperatures for specific days (Sundays) and adjusts to the provided schedule. The Tuesdays and Thursdays from 13 to 15 is usually not picked up well, a higher dimensional model or tweaking of the current parameters of the neural network would be needed to learn these features.

## 3.3 REST API

Some REST API functionality was implemented to expose data to other services, accessible through the Internet. The database of the API is fed through FHEM events targeting a device of interest.

A demo for both the machine learning and the API can be accessed in http://lutcodecamp.herokuapp.com/index.php

# 4 Result

We assume that the solution is applied in a student apartment using 2 radiators. The electricity usage of a radiator in normal mode is 1500W, and in low heating level mode is 1000W. In average, the student spends 11 hours at apartment in daytime and sleep from 12am-6am. The electricity cost is 0.15euro per kWh (Finland 2014).

In order to verify the energy consumption of our system, it was calculated the total amount of energy when the radiator operated in both normal scenario and after applying our solution.

| Scenario | Normal scenario | After applying our solution |
|---|---|---|
| Description | Radiators work 24/7 | Radiators work only when user is inside the apartment. Heating level decrease during the night when user is sleeping. |
| Energy consumption | [1500(W) * 2(heaters) * 24(hours)] * 30(days) = **2,160 kWh** | [1500(W) * 2 * 11(hours) + 1000(W) * 6(hours)] * 30 (days) = **1,350 kWh** |
| Electricity cost | 2,160 * 0.15 = **324 EUR** | 1,350 * 0,15 = **202.5 EUR** |
| Carbon Emission (0.225 kg CO2e/kWh) | 2,160 (kWh) * 0.225 kg CO2e/kWh = **486 kg CO2e** | 1,350 (kWh) * 0.225 kg CO2e/kWh = **303.75 kg CO2e** |

According to the above calculation, we clearly see that we can not only save energy usage but also save the electricity cost 121.5 EUR/month. Therefore, the cost of electricity saving for 2 months and a half can compensate to installation cost which cost around 298.83 EUR. In terms of CO2 emission, we reduce 182.25 kg CO2e by applying our solution.

# 5 Conclusions

During code camp, we implemented our scenario for home automation heating system using HomeMatic devices and Raspberry Pi for accessing the FHEM server. The main idea of the project was to reduce a heating energy consumption and CO2 emission which was brought into life by the scenario. According the calculation results, it saves 121.5 EUR/month and decreases the emissions on 182.25 kg CO2e.

Unfortunately, we did not have any possibility to implement our scenario using a thermostat and radiator. Instead, we exploit a light bulb with light controller to emulate a heater behavior. Also, humidity was not considered in the scenario. In addition, our machine learning solution, the neural network, has to be improved and integrated with the implemented system.

In the future, a comfort level has to be verified by applying humidex level. Further, an application to control the system remotely could be designed which should include options for users such as scheduling, notifications and heating in advance.

# References


[1] "Home of FHEM", *Fhem.de*, 2016. [Online]. Available: http://fhem.de/fhem.html. [Accessed: 11-Mar- 2016].

[2] "Electricity usage of a Space Heater - Energy Use Calculator", *Energyusecalculator.com*, 2016. [Online]. Available: http://energyusecalculator.com/electricity_spaceheater.htm. [Accessed: 11- Mar-2016].

[3] T. Peffer, M. Pritoni, A. Meier, C. Aragon and D. Perry, "How people use thermostats in homes: A review", *Building and Environment*, vol. 46, no. 12, pp. 2529-2541, 2011.

[4] U.S. Department of Energy Thermostats, 2016. [Online]. Available: http://energy.gov/energysaver/articles/thermosta. [Accessed: 11- Mar- 2016].