

Interreg



2 Seas Mers Zeeën

INCASE

European Regional Development Fund

Final Report on
Large Scale Smart Home Pilot
by
HZ University of Applied Sciences

| | |
|---------|---------------|
| Version | Final |
| Date | 18 April 2020 |
| by | Edward Mouw |
| ID | D2.3.3 |

The INTERREG project INCASE is partially funded by the European Union, through the INTERREG Va 2 SEAS program (2014-2020).

The fellow Dutch partner of HZ University of Applied Sciences (HZ UAS) was Impuls Zeeland.



The Dutch project partners were partially funded by the Province of Zeeland.



The development of the LSP at the HZ UAS was in close cooperation with CITC. We want to thank Franck Gaultier of CITC for the fine cooperation.



The INCASE project partners are:



Contents

| | |
|--|----|
| Introduction | 4 |
| Pilot Description | 4 |
| Pilot Setup | 5 |
| The hardware | 5 |
| The software | 6 |
| Integrated Demonstrators in the LSP environment | 6 |
| Achievements | 6 |
| Development of the LSP Integrated Smart Home System (several integrated platforms) | 6 |
| Development of energy sensing and energy logging techniques..... | 6 |
| Development of a flexible (HMI) dashboard for monitoring and analysis of energy production and consumption..... | 6 |
| Development of strategies for influencing human behavior in energy consumption | 7 |
| Dissemination of knowledge | 7 |
| References..... | 9 |
| Annex; Report “ Reducing the Energy Consumption of Vending Machines” | 10 |
| Annex; Smart Desk | 14 |
| Annex; Smart Gate Event Zeeland | 28 |

Introduction

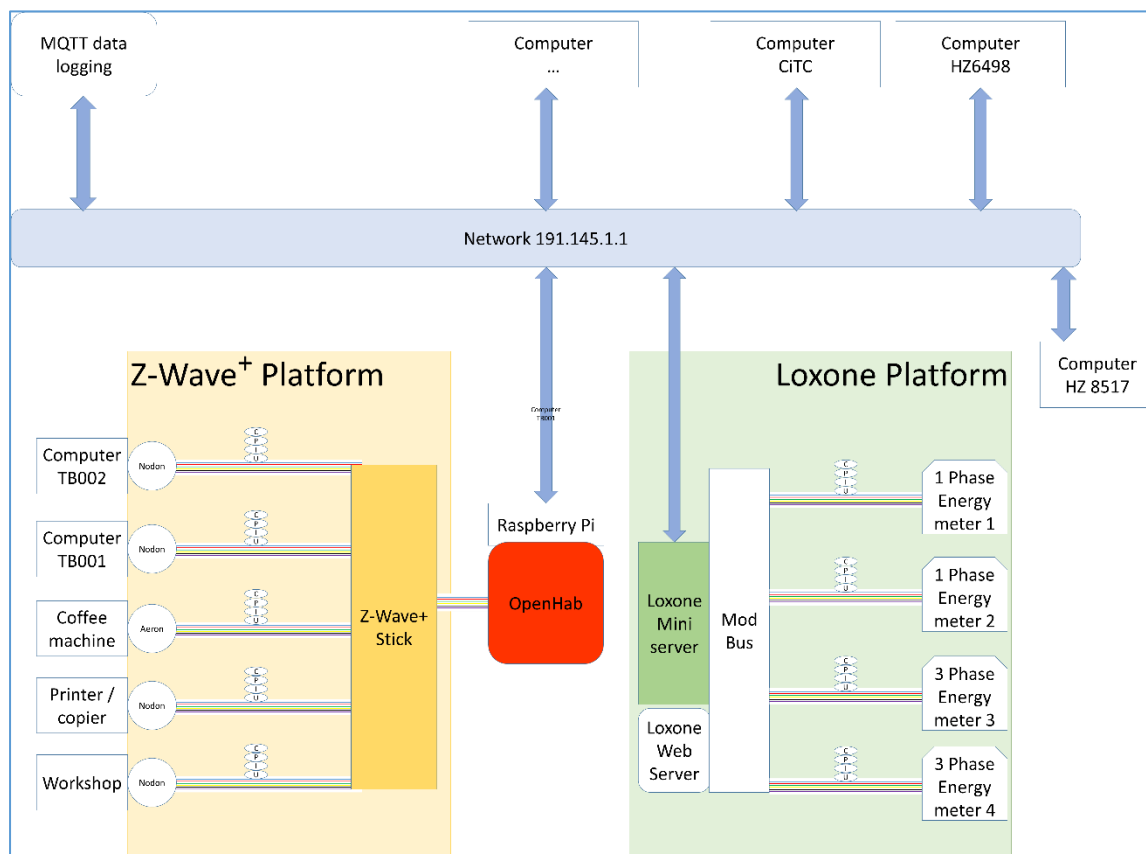
In 2016 the HZ University of Applied Sciences (HZ) started developing and studying Smart Home Platforms for health use and energy saving. It was discovered the development of functionality of the platforms increased rapidly, and new platforms were introduced (and others disappeared). A close watch was required.

In July 2018 The HZ started in close cooperation with our Incase partner CITC to develop the LSP. (Mouw 2018, Mouw, 2019)

Main research task of the LSP is on the one hand combining power saving sensors with (automated) switching tools, and on the other hand developing "influencer tools" in order to persuade people to change their power consumption behavior.

Pilot Description

The first test site was an office and workshop of a design & engineering firm in Biggekerke. The site occurred be to too remote from the HZ. The present site was established at the HZ premises in the building of Technisch Beheer (Technical Services). The building comprises machinery, an office environment, solar panels on the roof, air-conditioned rooms and an electric car loading station. These energy consumers and producer are included in the LSP. See the scheme below.

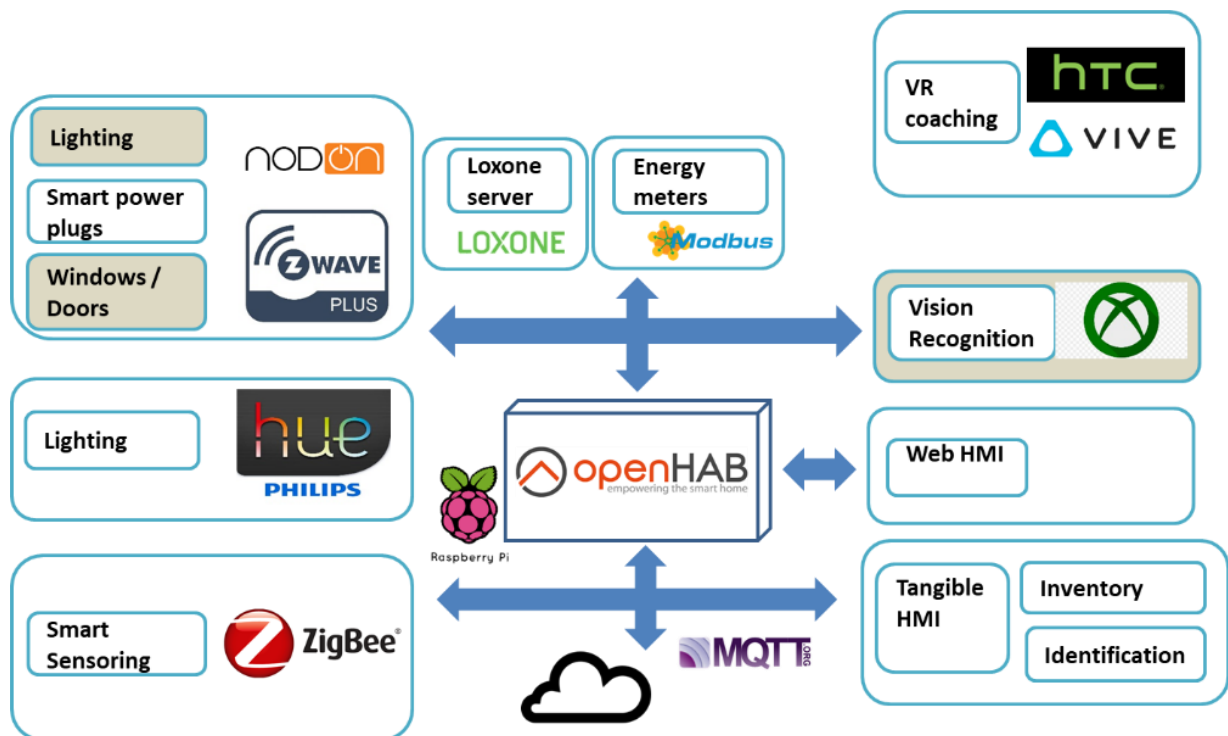


The goals of the LSP demonstrator are in detail;

1. Develop techniques for energy sensing
2. Develop techniques for energy logging
3. Develop integrated Smart Home system (several integrated platforms)
4. Develop a flexible (HMI) dashboard for monitoring and analysis of energy production and consumption
5. Research the human behavior in context of energy usage
6. Develop strategies for influencing human behavior in energy consumption

Pilot Setup

In the scheme below the basic setup with the different platforms are shown.



The hardware

A new main fuse box with Loxone/Modbus energy meters is installed in the TB building, a Loxone mini server with Modbus connection was installed and connected to the fuse box and the local network.

A local LAN/wifi network is built with a WAN-connection to cloud MQTT data logger and external operators. The mentioned Loxone server, a Raspberry Pi controller and several Windows computers are connected (for development, logging and controlling).

A Z-wave+ Smart Home platform is configured with several sensors on different office equipment. A Loxone Smart Home platform is installed with the energy meters on the solar panels, the electric car loading station and the air-conditioning equipment of the building.

The different subsystems are built and tested. Then the system as a whole was integrated and tested.

The software

Three smart home platforms were installed and connected;

7. Loxone
8. Z-wave+
9. Philips Hue (Zigbee)

The last one is disconnected since it was borrowed and not important in this stage of the pilot.

The sub systems were interconnected and hosted by an open source smart home system called OpenHab. And OpenHab is connected to the cloud based data base MQTT by LAN/WAN.

Integrated Demonstrators in the LSP environment

1. The Mobile Energy Loggers for Coffee and Vending machines. They were connected to the LSP MQTT system for energy logging. Now they have their own MQTT logging environment. (See Annex; Report "Reducing the Energy Consumption of Vending Machines") (Nahas, el. 2019)
2. The Smart Desk, a desktop device for influencing the energy consumption behavior of the office worker is build and demonstrated. See Annex "Smart Desk" (Luijmes 2019)
3. The Hygiene Buddy a demonstrator for hand washing hygiene and energy logging (it is presently used at the department of Health Education for educational purposes. This is a Raspberry Pi based system and can be connected to the LSP MQTT system for energy logging. (Markerink 2018)

Achievements

Development of the LSP Integrated Smart Home System (several integrated platforms)

The Integrated Smart home system is described in the Pilot Setup. In this system several platforms are integrated;

- Open Hab
- Zigbee
- Loxone PLC
- Z-wave

Energy logging systems and sensors (motion, temperature, body detection) are integrated.

Development of energy sensing and energy logging techniques

Development and building of a Mobile Energy Logging System based on Z-Wave, connected with the Eduroam network and hosted by the LSP MQTT database. (See Annex; Report "Reducing the Energy Consumption of Vending Machines"),

Presently 5 energy loggers are installed at the HZ University of Applied Sciences to monitor the energy consumption of coffee and vending machines. Next to monitoring they switch the machine of during night time in order to save energy.

Development of a flexible (HMI) dashboard for monitoring and analysis of energy production and consumption

1. A Grafana Dashboard is developed for both the LSP and Mobile Energy Logging System. See Annex; Report "Reducing the Energy Consumption of Vending Machines"

2. An interactive coaching dashboard is developed for the Hygiene Buddy. (Markerink 2018)

Development of strategies for influencing human behavior in energy consumption

The Smart Desk should be used in a pilot for “energy consumption reduction by active influencing” This pilot was planned by the Energy Board of the HZ UoAS and should be developed and built by students of the Department of Engineering (Design & Innovation) in spring 2020. Due to the COVID-19 issues the development of this pilot is postponed.

Dissemination of knowledge

1. Advising the Energy Board of the HZ UoAS concerning the energy consumption of coffee machines. This is based on the data recorded with the Mobile Energy Logging System. (Nahas, el. 2019)
2. A presentation and demonstration of the LSP is held for the deputy of the Province of Zeeland, mr. Harry van der Maas.



presentation for the deputy of the Province of Zeeland, mr. Harry van der Maas

3. A big event is organized at 13 June 2019; Smart Gate Event Zeeland. See Annex “Smart Gate Event Zeeland”. Here the demonstrators are presented and used. (Male, Roos 2019)



Workshop for companies in Province of Zeeland

4. At 9 December 2019 a workshop was organized for provincial companies to inform them about the LSP and logging techniques. (Mouw 2019)
5. Educational material such as cases, presentations are derived from the LSP and demonstrators are used in the educational environment
6. The Hygiene Buddy (demonstrator for hand washing hygiene and energy logging) is used at the department of Health education (Hygiene) and the department of Technology (HMI and Control Systems)

References

Luijmes, J.; *Smart Desk. Manual for an Internet of Things presence detector*. HZ UaS, 2019

Male, R, Roos, A; *Smart Gate Zeeland*; 2019

Markerink, T. et al.; *CU08061-Incase Hygiene Buddy 2018-02-01V1*. HZ UaS, 2018

Mouw, E.: *Milestone Report LSP Smart Homes 2018 - 20190111*. HZ UaS, 2018

Mouw, E.: *Milestone Report LSP Smart Homes April 2019 – 20190422*. HZ UaS, 2019

Mouw, E.: *Workshop Incase Large Scale Pilot HZ Smart Home*. HZ UaS, 2019

Mouw, E.: *Milestone Report LSP Smart Homes October 2019 - 20191008*. HZ UaS, 2019

Nahhas, O. el; *Reducing the Energy Consumption of Vending Machines*, HZ UaS, 2019

Nahhas, O. el; *Portable Energy Logger Manual_V1_18-6-2019*. HZ UaS, 2019

Annex; Report “ Reducing the Energy Consumption of Vending Machines”

1

Minor Fit for the Future

Reducing the Energy Consumption of Vending Machines

Omar El Nahhas

Engineering, HZ University of Applied Sciences, Vlissingen – The Netherlands

Abstract – The HZ University of Applied Sciences has multiple vending machines. By using Z-Wave, OpenHAB and a Raspberry Pi, experiments have been done to give an insight into the energy consumption of 2 different types of coffee machines. The Douwe Egberts Caffeisse had a nominal usage of 961 kWh per year. The Douwe Egberts Espresso Omni had a nominal usage of 851 kWh per year. The Douwe Egberts Espresso Omni its usage can be reduced from 851 kWh to 415,5 kWh on a yearly basis by actively turning off the coffee machine, which can be done by defining rules in OpenHAB which will turn the coffee machine on and off at pre-defined times. This results in a theoretical energy consumption reduction of 51,2% for the Douwe Egberts Espresso Omni on a yearly basis.

Keywords – Internet of Things, Vending Machines, Coffee Machines, Energy Consumption, Energy Saving, Home Automation, Z-Wave, Mesh Network, OpenHAB, Raspberry Pi.

I INTRODUCTION

FOR the minor Fit for the Future (with trajectory Internet of Things) sensors have been placed in the TD-building at the HZ University of Applied Sciences. The sensors create insight into the consumption of energy. When word came out that it is possible to create insight for principally every device which requires electricity, the university showed interest and wanted insight into the energy consumption of the university its coffee machines. This research report is based upon experiments done with the university its coffee machines. This report answers the following question: “How can the energy consumption of vending machines be reduced to 50% of its nominal energy consumption on a yearly basis?”.

II THEORY

The sensors used for the measurements are sensors which communicate through the Z-Wave Plus protocol. Z-Wave is a wireless communication protocol that essentially focusses on connectivity within the smart home. Z-Wave operates using low-energy radio waves to communicate from device to device. Z-Wave Plus, also known as Z-Wave 500 series, took the technology that was launched in 2004 and added improvements such as an increased range, extended battery life, OTA (Over the air) upgrading and additional RF (Radiofrequency) channels. Z-Wave

operates on the 800-900 MHz radio frequency range. Unlike Wi-Fi, where devices have to connect to a central hub, Z-Wave devices all link up together to form a mesh network.



Figure 1: Mesh network

There is one central hub that is connected to the internet. The devices themselves use Z-Wave connectivity to talk to the hub. That connectivity doesn't have to be direct; the mesh network means signals can hop from device to device. This mesh network supports up to 232 nodes. For security, Z-Wave Plus uses AES-128 symmetric encryption [1].

The open Home Automation Bus (OpenHAB) is an open source, technology agnostic home automation platform. OpenHAB communicates electronically with smart devices, performs user-defined actions and provides web-pages with user-

| Concepts | Meaning |
|----------|---|
| Bindings | are the openHAB component that provides the interface to interact electronically with devices |
| Things | are the first openHAB (software) generated representation of your devices |
| Channels | are the openHAB (software) connection between "Things" and "Items" |
| Items | are the openHAB (software) generated representation of information about the devices |
| Rules | that perform automatic actions (in its simplest form: if "this" happens, openHAB will do "that") |
| Sitemap | is the openHAB (software) generated user interface (web site) that presents information and allows for interactions |

Table 1: OpenHAB concepts

defined information as well as user-defined tools to interact with all devices. To achieve this, OpenHAB segments and compartmentalizes certain functions and operations. Table 1 gives a top-level description of the most important concepts [2].

III. METHOD OF RESEARCH

To measure the energy consumption of the coffee machines, a working prototype has been made by assembling, connecting and configuring the following items: Raspberry Pi, SD card with OpenHAB Raspberry Pi image, TP-Link router, Z-Wave+ protocol stick, AeonLabs Switch 6 power plug (3000W), ethernet cable and a Schneider Electric case.



Figure 2: Inside of the prototype

The router is used to create a local network. The Raspberry Pi (RPI) will connect to the local network through an ethernet cable. The SD card goes into the RPI and it will install OpenHAB. The dashboard of OpenHAB can be opened in a browser by typing in #IP address of the RPI#:8080 in the search bar. Now the Z-Wave+ stick can be put in the RPI and configured in the dashboard. The smart plug can be connected to the OpenHAB platform following its own inclusion procedure¹. The prototype is now ready to collect data. By using persistence databases in OpenHAB and the Grafana binding, the data can be viewed in Grafana through #IP address of the RPI#:3000.



Figure 3: Outside of the prototype

¹ AeonLabs Switch 6 manual. Available: <https://www.intin.no/wp-content/uploads/2015/09/aeotecss6manual.pdf>



Figure 4: Results of the experiment with the Douwe Egberts Espresso Omni

IV. RESULTS

Douwe Egberts Cafitesse Excellence:

Two sets of experiments have been executed on the same (older) machine during different periods of activity. The first experiment has been done during a one week holiday break. The second experiment has been done during a regular week of school, including the weekend. Idle is defined as “Available to make coffee, but there is no user available”. Active is defined as “Available to make coffee and makes coffee when an user demands it”.

Experiment 1 had a duration of 282 hours (11 days and 18 hours). After 282 hours of an idle coffee machine, the usage was 27,6 kWh. This results in a yearly usage of:

$$\frac{8765,81277}{282} * 27,6 \approx 857,9 \text{ kWh.}$$

Experiment 2 had a duration of 120 hours (5 days, 3 weekdays and 2 weekend

days). After 120 hours of an active coffee machine, the usage was 13,15 kWh.

This results in a yearly usage of:

$$\frac{8765,81277}{120} * 13,15 \approx 960,6 \text{ kWh.}$$

Douwe Egberts Espresso Omni: One experiment has been executed on the newer coffee machine which had a duration of 339 hours. After 339 hours, the usage of the active newer coffee machine was 32,9 kWh, see figure 4. This results in a yearly usage of:

$$\frac{8765,81277}{339} * 32,9 \approx 851 \text{ kWh.}$$

During the experiment, the usage from 18:00 – 08:00 (outside office times) was on average 1 kWh. This results in a yearly usage of 365 kWh during the nights. The usage in the weekend (Friday 18:00 – Monday 08:00) was on average 4,2 kWh. This results in a yearly usage of 220 kWh during the weekends. While calculating the actual energy that could be saved, adding 365 and 220 would be a faulty calculation, because the nights in the weekends were included in the 365 kWh/year. When

excluding the period from 18:00 to 08:00 (14 hours) 3 times from the total weekend period of 62 hours, a total of 20 hours “weekend” remain. Hence, alongside the 365 kWh per night, the following calculation for the weekend on a yearly basis would be correct:

$$\frac{4.2}{62} * 20 * 52 \approx 70,5 \text{ kWh.}$$

In theory, this would lead to a reduction in energy consumption of $365 + 70,5 = 435,5 \text{ kWh}$ per year, which is a 51,2% decrease in energy consumption. Rules in OpenHAB can be defined which will turn the AeonLabs Switch 6 power plug on and off autonomously at pre-defined times. This will take away human effort in the process of saving energy.

V. CONCLUSION

When comparing the results of the experiments, there is an usage difference of 110 kWh between the older and newer coffee machine. Both coffee machines have a high usage, because they are never turned off. It can be concluded that, in theory, the energy usage of the newer coffee machine, the Douwe Egberts Espresso Omni, can be cut by 51,2% of its nominal energy consumption by actively shutting down the coffee machines when they are not used. This can be done autonomously through OpenHAB.

VI. DISCUSSION

The results of the experiments can be considered as inaccurate because of its small sample size. For a better experiment, it is advised to have measurements of at least 6 months to a year in different states of activity, between multiple machines. Because of the duration of the minor and the obligation to deliver before a certain deadline, the choices were limited regarding the time period for the experiments. The energy that can be saved are theoretical calculations based on the small size of the experiments. For all experiments, it is assumed that the coffee machines will continue the same behaviour throughout the year as they did during the experiments. Also, it is assumed the coffee machines are not turned off. Thus, the coffee machines are running 24 hours per day for 365 days.

REFERENCES

- [1] Lamkin, P. (2019, January 22). Z-Wave explained: What is Z-Wave and why is it important for your smart home? Retrieved from The Ambient: <https://www.the-ambient.com/guides/zwave-z-wave-smart-home-guide-281>
- [2] Introduction. (n.d.). Retrieved from OpenHAB: <https://www.openhab.org/docs/>

Annex; Smart Desk

Smart Desk

Manual for an Internet of Things presence detector



Joost Luijmes
September 2019

Introduction

The 'Smart Desk' is an Internet of Things device developed by Joost Luijmes at the HZ University of Applied Sciences as a 'small scale demonstrator' of the InCase project. This project is part of the Interreg 2 Seas initiative and focusses on smart applications in Industry 4.0. The Smart Desk is a device related to innovation within a smart home or office environment. The latter clarifies the device's name, a device which can be conveniently set up on a desk to get a view of its surroundings.

Contents

| | |
|--|----|
| What is it? | 17 |
| How does it work? | 18 |
| Technical specifications | 19 |
| Source code | 20 |
| Set-up guide | 25 |
| Making the Smart Desk | 25 |
| Configuring the smart home controller | 26 |
| Communication between OpenHAB and the Smart Desk | 26 |
| Further development | 27 |

What is it?

Concisely, the goal of the Smart Desk is to save energy. The device is aimed at a human whose body temperature can be detected. That way, the Smart Desk detects presence or absence. It relays that information to a smart home controller via the MQTT protocol. The smart home controller then takes a certain action as programmed by the user, which may be able to save energy. The application of the Smart Desk is therefore only limited by the smart home controller's capacity. An example of the Smart Desk's use may be lights and air conditioning that are turned off if no one has been within the Smart Desk's field of view for 5 minutes.

Since the Smart Desk would be a device situated on a desk, its appearance should be harmless and not cause discomfort to the person sitting at the desk. Moreover, the user should not feel like he/she is monitored all day which may lead to an uncomfortable feeling. This is one of the reasons the Smart Desk is not equipped with a camera but with a sensor containing an array of infrared detectors. These sensors 'see' body temperature in a low resolution. Besides this, the use of a camera would require more advanced image processing which may require more expensive hardware. A PIR sensor would not suffice as a sensor since such a sensor can only detect motion and would therefore misinterpret a user sitting still for no user at all.

How does it work?

Put simply, the Smart Desk consists of an eye and a computer. The eye 'sees' human presence and the computer processes these signals and communicates when no presence is detected for a certain amount of time. After this time has passed and still no presence has been detected, the computer sends a message on a configured MQTT topic which can be received by the smart home controller. From there, the smart home controller takes over. In a way, the Smart Desk is the sensor and the smart home controller the actuator.

At the heart of the SmartDesk lays a microcontroller, the NodeMCU V3 (manufactured by Lolin). It is a microcontroller comparable to an Arduino but is equipped with WIFI capabilities necessary for the MQTT communication. It must therefore be connected to a network with a smart home controller (like OpenHAB running on a Raspberry Pi).

The Omron D6T-44L-06 is a sensor that consists of an array of InfraRed detectors (4x4) of which each is able to monitor temperature. It is therefore able to form a heat 'picture' of what it is pointed to (figure 1). To visualise this in real-time you may refer to the Processing source code¹.

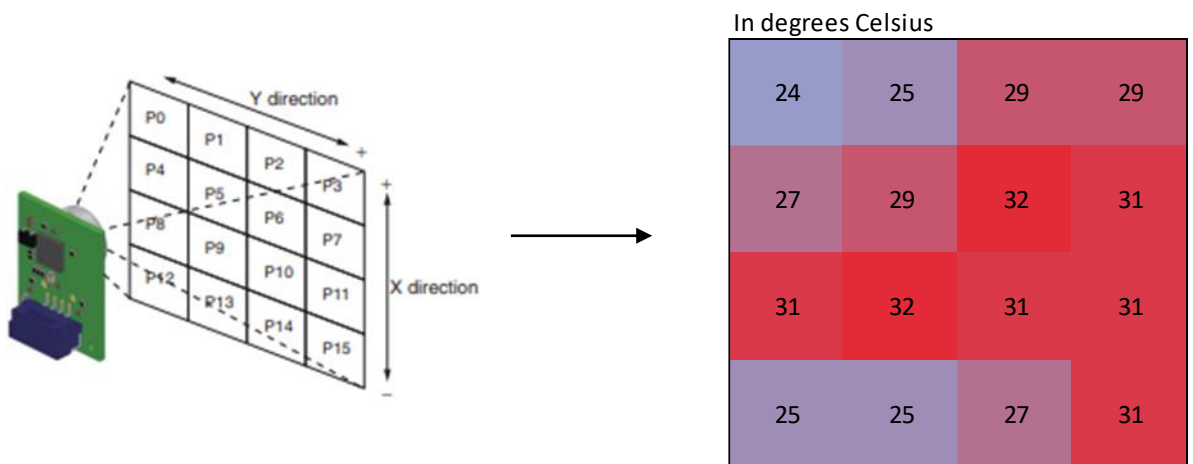


Figure 1; a visualisation of the sensor's captured data.

¹ Processing is a programming environment aimed at visual arts. It can therefore be used to easily process data to a visual form as is demonstrated in the included code.

Technical specifications

The SmartDesk consists of the following parts:

Electrical hardware

Lolin NodeMCU V3 [Link](#)

Omron D6T-44L-06 [Link](#)

Omron Cable [Link](#)

Veroboard (holes that are joined as triplets are recommended to ease the assembly)

5k resistors (2x)

Hardware

M3 screws (3x)

Screw for the side panel

3D printed enclosure

Peripherals

(only when Z-Wave protocol is used for actuating) [Link](#)

NodOn Micro Smart Plug

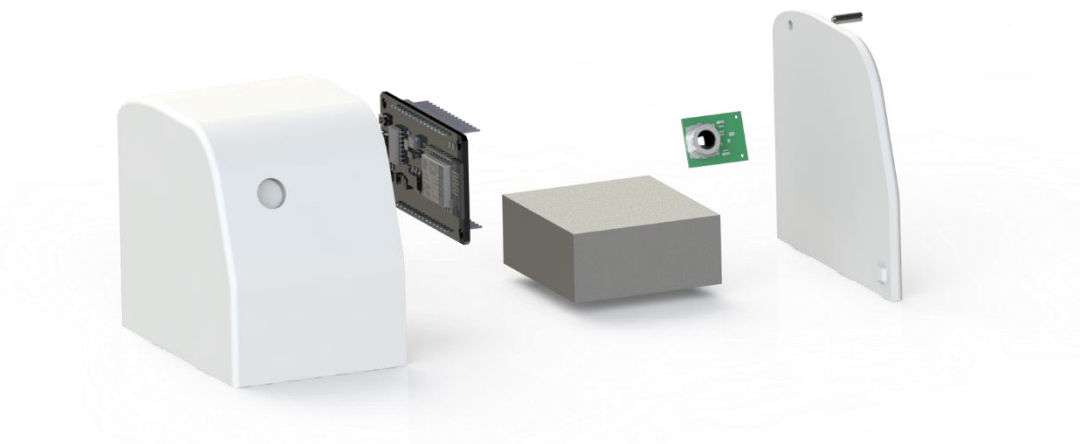
(only when Z-Wave protocol is used for actuating) [Link](#)

Z-Stick Gen5

Raspberry Pi model 3b+ running Openhab

Laptop for live view

Soldering equipment



Source code

NodeMCU source code

```
/*
  Large parts of this code were used from the 'Basic ESP8266 MQTT' example sketch included in the
  ESP8266 Arduino Core and an Arduino forum post (https://forum.arduino.cc/index.php?topic=217394.0).
  Also, parts of the PubSub library example sketches were used
*/

#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <Wire.h>
// Update these with values suitable for your network.

#define D6T_addr 0x0A // Address of OMRON D6T is 0x0A in hex
#define D6T_cmd 0x4C // Standard command is 4C in hex

const char* ssid = "network name";
const char* password = "network password";
const char* mqtt_server = "ip address mqtt broker";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;

int rbuf[35]; // Actual raw data is coming in as 35 bytes. Hence the needed for WireExt so that we
              // can handle more than 32 bytes
int tdata[16]; // The data coming from the sensor is 16 elements, in a 16x1 array
float t_PTAT;

const int tempThreshold = 30;
unsigned long timeSinceLowTemp = 0;
const int noPresenceWaitTime = 5000;

bool personPresent;
bool prevState;
bool firstTime = true;

void setup_wifi() {
  delay(10);
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  randomSeed(micros());

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++) {
```

```
Serial.print((char)payload[i]);
}
Serial.println();

// Switch on the LED if an 1 was received as first character
if ((char)payload[0] == '1') {
digitalWrite(BUILTIN_LED, LOW); // Turn the LED on (Note that LOW is the voltage level
// but actually the LED is on; this is because
// it is active low on the ESP-01)
} else {
digitalWrite(BUILTIN_LED, HIGH); // Turn the LED off by making the voltage HIGH
}
}

void reconnect() {
// Loop until we're reconnected
while (!client.connected()) {
Serial.print("Attempting MQTT connection...");
// Create a random client ID
String clientId = "ESP8266Client-";
clientId += String(random(0xffff), HEX);
// Attempt to connect
if (client.connect(clientId.c_str())) {
Serial.println("connected");
// Once connected, publish an announcement...
client.publish("NodeMCU/Presence", "NodeMCU online");
// ... and resubscribe
client.subscribe("inTopic");
} else {
Serial.print("failed, rc=");
Serial.print(client.state());
Serial.println(" try again in 5 seconds");
// Wait 5 seconds before retrying
delay(5000);
}
}
}

void setup() {
pinMode(BUILTIN_LED, OUTPUT); // Initialize the BUILTIN_LED pin as an output
Wire.begin(4, 2); // 4&2 refer to the GPIO pins [SDA],[SCL] where 4 does not equal D4 but D2. See
a NodeMCU pinout diagram for pin mapping.
Serial.begin(115200);
setup_wifi();
client.setServer(mqtt_server, 1883);
client.setCallback(callback);
}

void loop() {

if (!client.connected()) {
reconnect();
}
client.loop();

delay(500);

Wire.beginTransaction(D6T_addr);
Wire.write(D6T_cmd);
Wire.endTransmission();
Wire.requestFrom(D6T_addr, 35); // Added this line since the standard Wire library is used

// Getting input from the sensor
if (Wire.available() >= 35) {
for (int i = 0; i < 35; i++) {
rbuf[i] = Wire.read();
}
}
}
}
```

```
}

// Do something with temperature compensation that we don't seem to use currently.
t_PTAT = (rbuf[0] + (rbuf[1] << 8)) * 0.1;

// Calculate the temperature values: add the low temp and the bit shifted high value together.
Multiply by 0.1
for (int i = 0; i < 16; i++) {
  tdata[i] = (rbuf[(i * 2 + 2)] + (rbuf[(i * 2 + 3)] << 8)) * 0.1;
}

int maxTemp = tdata[0];
for (int i = 0; i < 16; i++) {
  if (tdata[i] > maxTemp) maxTemp = tdata[i];
  Serial.print(tdata[i]);
  Serial.print(",");
}
Serial.print("\n");

personPresent = maxTemp < tempThreshold ? false : true;
//Serial.println("person Present: " + personPresent);
if(firstTime){
  prevState = !personPresent;
  firstTime = false;
}

if(personPresent == false && timeSinceLowTemp == 0) timeSinceLowTemp = millis();
else if (personPresent == true) timeSinceLowTemp = 0;

if (millis() > timeSinceLowTemp + noPresenceWaitTime && personPresent != prevState) {
  String message = personPresent ? "Presence" : "Absence";
  client.publish("NodeMCU/Presence", message.c_str());
  //Serial.println(message);
  prevState = personPresent;
}
}
```

In this code the user needs to provide the network name, network password, MQTT broker address and 'publish topic'. The user may also change the `noPresenceWaitTime` variable which determines the duration the microcontroller waits before telling the smart home controller that no presence is detected. The code detects presence when one off the sensor's IR detectors measures a certain temperature. This threshold temperature can be adjusted by changing the `tempThreshold` variable. Any temperature larger than `tempThreshold` is considered to be a presence.

Processing source code

```
import processing.serial.*;

Serial myPort; // Create object from Serial class

final int SPACER = 200;

void setup()
{
  size(800, 800);

  String portName = Serial.list()[0];
  myPort = new Serial(this, portName, 115200);
}

void draw()
{
  String inString = null;
  int[] temps;

  if ( myPort.available() > 47) { // If data is available,
    inString = myPort.readString();

    if (inString != null) {
      temps = int(split(inString, ','));
      printArray(temps);
      print();
      drawCells(temps);
    } else {
      println("inString is niets");
    }
  }
}

void drawCells(int[] temps){
  background(255);
  color koudColor = color(66, 203, 245);
  color warmColor = color(245, 50, 50);

  int koudTemp = 20;
  int warmTemp = 40;

  int[] colors = new int[16];

  for (int i = 0; i < 16; i++){
    //if (temps[i] == 0) break;
    float lerpVal = map(temps[i], koudTemp, warmTemp, 0, 1);
    colors[i] = lerpColor(koudColor, warmColor, lerpVal);
  }

  int i = 0;
  for(int x = 0; x < width; x += SPACER){
    for(int y = 0; y < height; y += SPACER){
      fill(colors[i]);
      rect(x,y,SPACER, SPACER);
      i++;
    }
  }
}
```

Rules OpenHAB

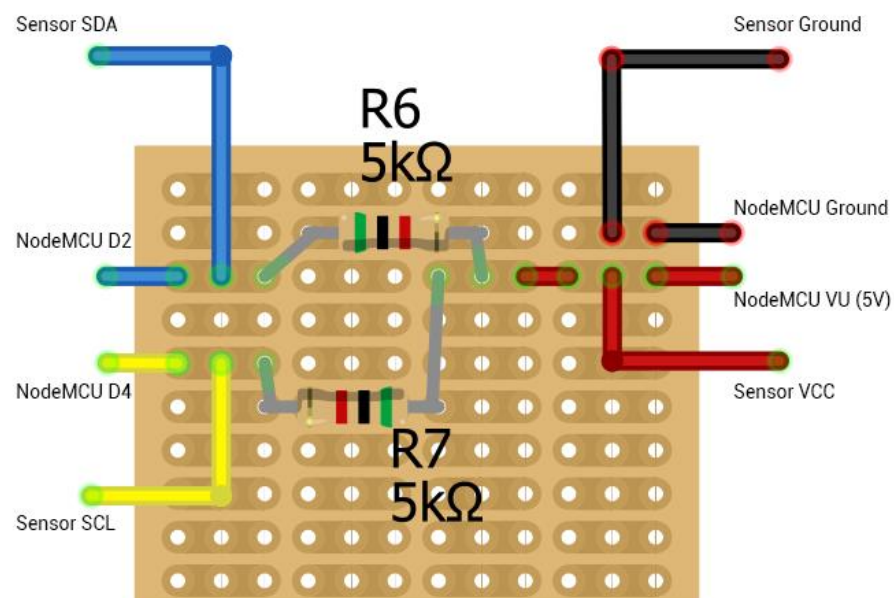
```
rule "Smart_Desk_presence_to_absence"  
when  
  Item SmartDesk_ changed from "Presence" to "Absence"  
then  
  NodOn1_Switch.sendCommand(OFF)  
end  
  
rule "Smart_Desk_absence_to_presence"  
when  
  Item SmartDesk_ changed from "Absence" to "Presence"  
then  
  NodOn1_Switch.sendCommand(ON)  
end
```


Set-up guide

Making the Smart Desk

The enclosure needs to be 3D printed. STEP and Solidworks can be found in the included .rar files. The NodeMCU and Omron sensor can be affixed with screws. A bit of glue is also recommended. Furthermore, we encourage placing a weight in the enclosure. This gives the Smart Desk a more solid and qualitative feeling. In *Further development* the possibility of replacing this weight with a 9V battery is mentioned as an alternative.

Next, the electronics need to be assembled. The NodeMCU's logic voltage is 3.3V yet online sources indicate that the 5V logical level of the sensor's I²C bus can be used without problems. The sensor needs to be powered at 5 volts, for this reason the NodeMCU's VU pin must be used to power it. Take note that the correct output pin may differ per NodeMCU manufacturer, these voltages must therefore always be checked! With the way the diagram is set up, the board can fit directly onto the NodeMCU's pins.



[foto van hoe het veroboard op de Node past]

Configuring the smart home controller

We recommend to use the smart home controller OpenHAB. As such, this paragraph details the steps to be taken to get the Smart Desk working with OpenHAB.

Install OpenHAB on a raspberry pi using the following guide:

<https://www.openhab.org/docs/installation/>. When OpenHab is installed a few things must be configured. First of all, access PaperUI and browse to the Add-on page. There you need to install 'binding-mqtt - 2.4.0' under the Bindings tab and 'Embedded MQTT Broker, misc-mqttbroker - 2.4.0' under the Mics tab. When these are installed, a new 'Thing' must be added, which will be the MQTT broker. When this is done, another 'Thing' must be added, which will be where the MQTT topics will be handled. Configure this Thing to subscribe to the Smart Desk's MQTT topic.

Communication between OpenHAB and the Smart Desk

To utilise MQTT, the Smart Desk (NodeMCU) must be on the same wireless network as the raspberry pi. Therefore, the Wi-Fi connection needs to be configured on both. For the Raspberry Pi, we recommend following this guide:

<https://www.raspberrypi.org/documentation/configuration/wireless/wireless-cli.md>.

For the NodeMCU, the network name and password must be filled into the following fields:

```
const char* ssid = "network name";  
const char* password = "network password";
```

Further development

The most important goal in further development should be to optimise the NodeMCU's code. The energy savings the Smart Desk can achieve should outweigh its own energy usage, something that at the moment cannot be guaranteed in usual user scenarios. Some things to consider is whether a constant WIFI/MQTT is necessary.

Without much trouble the Smart Desk can be made completely wireless by powering with a battery. The NodeMCU used already has the appropriate connections and functions to facilitate a 9V battery. Using such a battery, the weight could also be omitted.

To put the user more at ease, possibilities of hiding the Omron D6T's eye should be researched. It may be possible to hide it without diminishing its functionality.

Annex; Smart Gate Event Zeeland

Smart Gate Zeeland

Hiermee nodigen we u uit voor het evenement *Smart Gate Zeeland*. Een middag met inspirerende en informatieve bijdragen over de digitale transformatie in diverse sectoren. Tijdens deze middag kunt u zelf ook oplossingen zien: voor iedere sector worden twee projecten of producten gepresenteerd.

Aanleiding

In alle sectoren leeft het besef dat de digitale transformatie de komende jaren een grote impact zal hebben. Digitalisering biedt kansen: een krappe arbeidsmarkt is voor sommigen aanleiding om in robotisering te investeren. Anderen willen hun productieprocessen verduurzamen en zetten digitalisering in om energie en grondstoffen te besparen.

Digitalisering roept ook vragen op: wat zijn de trends op dit gebied? Hoe analyseer ik mijn processen? Wat betekent het voor mijn medewerkers? Waar vind ik goede scholing? Zijn er in de buurt kennis- en implementatiepartners die mij kunnen helpen? Hoe gaan anderen, binnen en buiten mijn sector, hier eigenlijk mee om? Er is behoefte aan kennis en inspiratie. Ook geven bedrijven aan rond dit onderwerp betere verbindingen te willen hebben met andere bedrijven/sectoren en kennisinstellingen.

Wanneer en voor wie?

Op 13 juni organiseren Impuls Zeeland en HZ University of Applied Sciences het evenement **Smart Gate Zeeland**. Het doel is om de vertegenwoordigers van verschillende sectoren en ondernemers bij elkaar te brengen, hen te informeren en inspireren om een actieve rol te pakken in deze transitie. U bent ook van harte welkom wanneer u op een andere wijze betrokken bent op dit onderwerp: vanuit beleid, overheid of kennisinstelling.

Datum en tijd

13 juni 13:30 – 17:00, borrel na afloop

Programma

| | |
|---------------|---|
| 13:30 – 14:00 | Inloop en koffie |
| 14:00 – 14:05 | Welkom <i>Rutger van de Male en Arend Roos</i> |
| 14:05 – 14:30 | "Smart Industry, Implementatieagenda 2018-2021" Bijdrage vanuit SI NL |
| 14:30 – 14:55 | "Procesindustrie en Maintenance" <i>Pieter Raes, KI< MPI</i> 2 pitches |
| 14:55 – 15:20 | "Agri en Food" <i>Ronald de Vos, Food Delta</i> 2 pitches |
| Pauze | |
| 15:45 – 16:05 | "INCASE en Smart Industry" <i>Jos Knockaert, Universiteit Gent</i> |
| 16:05 – 16:30 | "Wonen en Zorg" |

| | |
|---------------|---|
| | <i>Hanneke Bastian</i> 2 pitches |
| 16:30 – 16:55 | "Logistiek" <i>Thierry Verduin, HZ University of Applied Sciences</i> 2 pitches |
| 16:55 – 17:00 | Vervolg en afsluiting <i>Rutger van de Male en Arend Roos</i> |

Locatie

PZEM-zaal, Dockwize, Vlissingen

Deze bijeenkomst wordt georganiseerd vanuit het programma: INCASE. Dit project wordt medefinancierd door het INTERREG 2 zeeën programma en de Provincie Zeeland.

Logo's

Logo HZ

Logo KI | <MPI

