

KOSZALIN UNIVERSITY OF TECHNOLOGY
Faculty of Mechanical Engineering

Embedded computer system architecture and programming

Author:

GASNIER Ugo

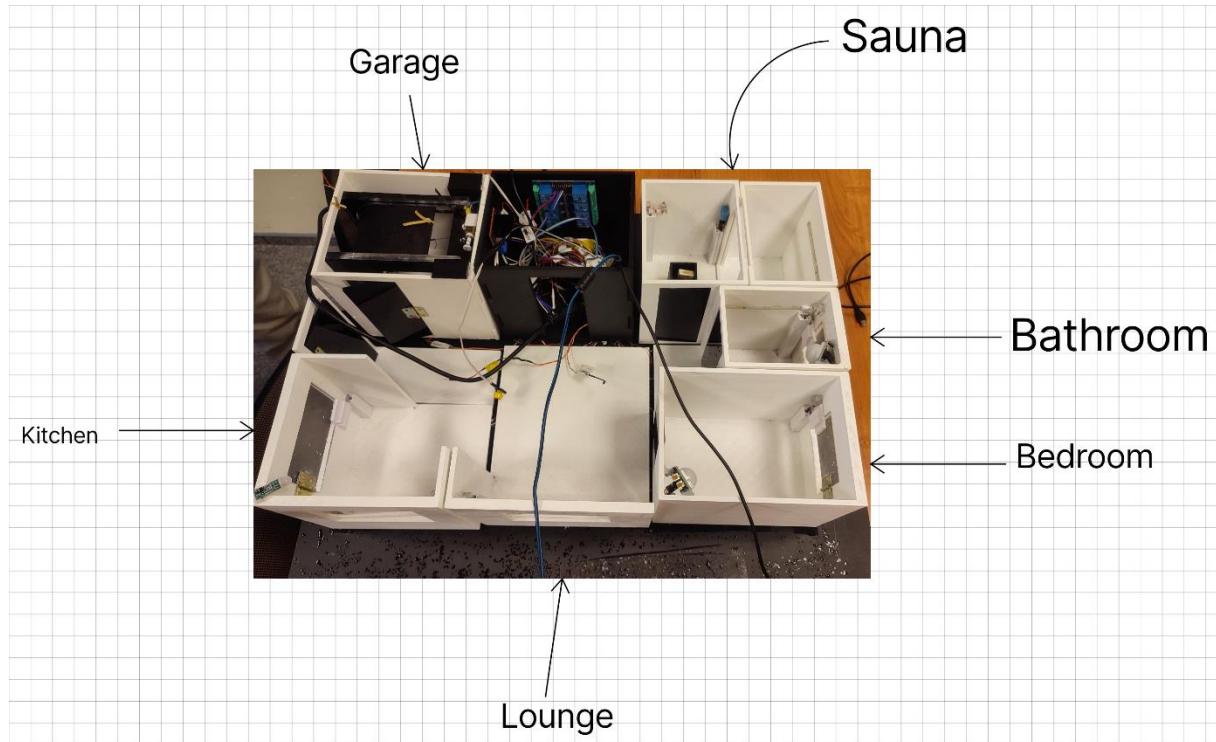
ROCHE Corentin

Course:

ERASMUS, Mechatronics

Introduction

The aim of our project was to develop a smart home using Raspberry pi technology and different types of sensors. This report will detail the current state of the home, explaining what works and what doesn't in terms of hardware and software.



Hardware part

Let's start with the hardware part:

The biggest problem we've had is that we don't have enough pins on the raspberry to connect all the devices. So, we chose not to connect the sensors in the "Sauna" room because we didn't know how to connect the Peltier sensor. In fact, under the room there's an additional card that controls the sensor, but we don't know how to make it work.

Another problem is using the relays to operate the LED strips. We didn't have enough pins to connect them, but they work in software terms.

Another malfunctioning sensor is the PIR sensor in the kitchen, which is connected to the Raspberry pi but detects random movement. The most likely cause is a false contact in the wiring, but we haven't been able to check this because the cables run through the wall of the room.

All the window opening sensors are functional, and when they detect an opening, they light up the corresponding LED.

The garage is the most complex part of the project, as it contains several sensors. The motor's movements are controlled by the button. The ultrasonic sensor has not been connected, so the LEDs that should indicate the distances are not working. The other improvements to be made are:

Change the way the cable is attached to the motor, as it moves when the motor is running.

Insert a spring or similar in place of the elastic that closes the garage door.

In general, the cable management can be improved, particularly the way in which all the sensors and the power board are powered.

Here's a summary of how to connect each pin on the Raspberry pi:

Raspberry pin	Sensor
Garage	
Leds	16/17/18
Echo	13
Trigger	14
Switch Down	2
Switch High	3
Optical barrier	19
Kitchen	
Led	23
Opening sensor	21/22
PIR	20
Sauna	
DHT Out	//
Opening sensor	//
EXO	//
Bedroom	
Opening sensor	25
Led	26

PIR	27
Lounge	
Led	5
Opening sensor	24
Bathroom	
Led	10
Opening sensor	11
PIR	9
Motor	
IN1	12
IN2	8
Relays	
IN1	//
IN2	//
IN3	//

Software part

We used a Raspberry Pi card with the Python language, so it is very easy to use on this card being under Linux.

The programs are managed according to a main program which calls all the subprograms supporting the parts.

A “room” subroutine manages all the home automation in this room. And these subprograms are parallelized using the library called Threading.

In the `__init__` function of each subprograms class, we declare and parameter all the pins for the card. Also, we create the parallelization, until 3 for the garage room.

Programs organisation

- Bathroom:
 - o `_run` -> it's a function which looks if the window is open and alight the alarm Led, also looks the state of the PIR sensor to alight the Led pannel
- Bedroom:
 - o `_run` -> it's a function which looks if the window is open and alight the alarm Led, also looks the state of the PIR sensor to alight the Led pannel
- Garage:
 - o `_mesuring_sensor` -> it's a function witch merure the distance between an object and the wall and alight one of three Led to show the distance
 - o `_run_portal` -> it's a function witch managed the garage's door with optical sensor, switch limit sensors and button
 - o `_sensors` -> it's a function witch look the state changeset of each sensor to modify the interface, change value for the function `_run_portal` and stop the change the garage's door if it's necessary
- Kitchen:
 - o `_run` -> it's a function which looks if the windows are open and alight the alarm Led, also looks the state of the PIR sensor to alight the Led pannel
- Salon:
 - o `_run` -> it's a function which looks if the window is open and alight the alarm Led
- Sauna:
 - o `_run` -> it's a function which looks if the door is open and alight the alarm Led and show in the interface the humidity and the temperature
 - o It's possible to add a function to manage the Peltier in terms of the humidity and the temperature
- Page:
 - o All the interfaces are in this program, it's Tkinter interface which show all the values of output or input of the room
- `__main__`:
 - o Create the tkinter interface and lunch each subroutine thanks the class Home

To lunch this last program, you should write:

```
python -m src          in the parent folder
```

All the programs used on this project where be consultable and downloadable on Github

<https://github.com/BenCestMoiQuoi/home-automation-model>