Embedded Systems & Data Acquisition

**PoC – Proof Of Concept**

**Engineering Design Process Application**



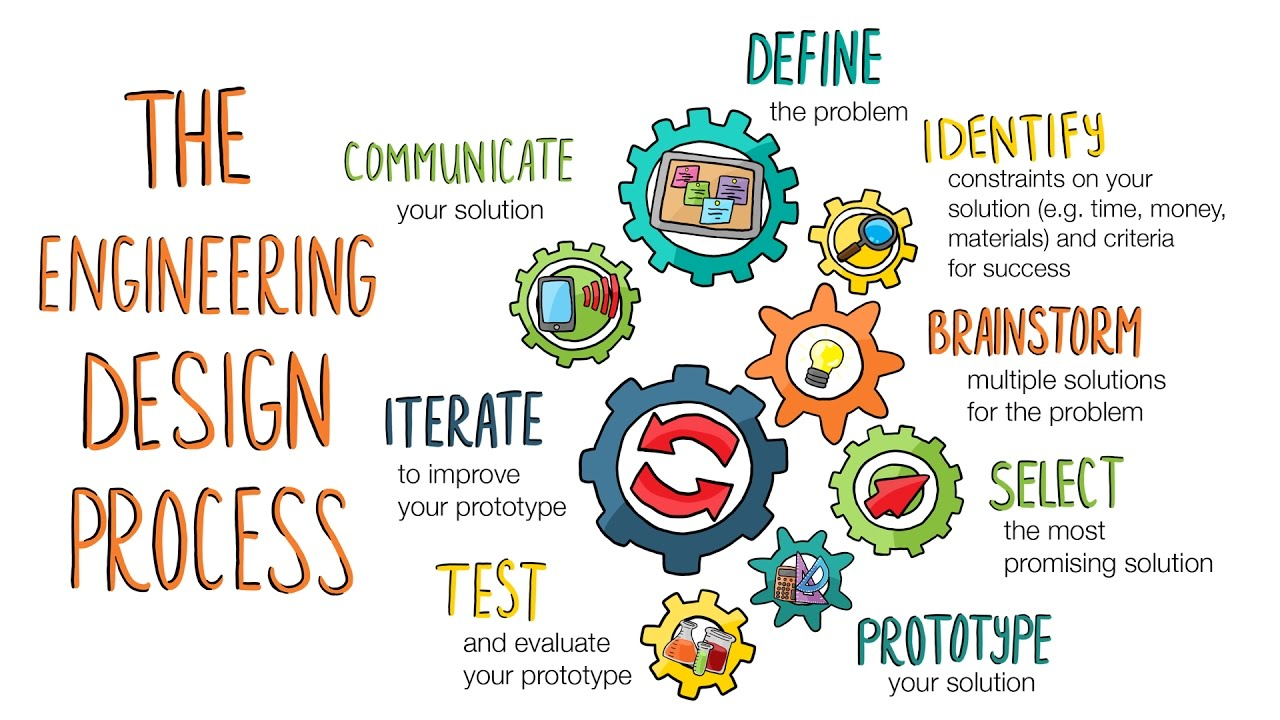
Gauthier FALISE

Grégoire DESFOSSE

ESILV A4 - EVD1

**Introduction**

During this second part of embedded system course, we had to implement a proof of concept, from the search for a problem to be solved, until the installation of a first prototype The goal of this exercise is firstly to allow us to discover more precisely the engineering design process, and secondly to apply this engineering design process, in developing our own proof of concept. We first had to look for a need to be solved, then analyzed the different solutions already in existence. We then researched many possible solutions, to finally arrive at our final solution. And finally, we set up our first prototype, and obtained our first results of our new and innovative concept.



**Report summary**

Table des matières

[1) Definition of the problem 3](#_Toc58787518)

[2) Background Researches 5](#_Toc58787519)

[3) Specification of requirement 7](#_Toc58787520)

[4) Brainstorming of solution and production of the first design 8](#_Toc58787521)

[5) Block diagram of our solution 9](#_Toc58787522)

[6) First prototype : material, assembly, and code 10](#_Toc58787523)

[7) First results 13](#_Toc58787524)

[Conclusion 13](#_Toc58787525)

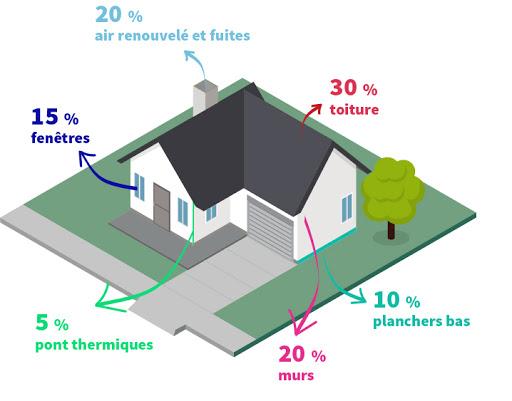
[Bibliography : 14](#_Toc58787526)

## Definition of the problem

First of all, we need to define the problem we want to solve. To do that we will try to answer these three questions: **who** needs **what** because **why?**

In our group we immediately agreed on a problem that was particularly important to us: The energy transition. Subject therefore directly linked to the energy efficiency of buildings.

Here are some quite striking figures to begin with: as you can see on the diagram, 20 percent of the energy losses in a house take place through the walls, and 30 percent through the roofs, i.e. half of the energy losses on these flat surfaces.



*Figure 1:diagram of the distribution of thermal losses in a traditional house*

So, the problem we want to solve with our project is related to energy savings and the insulation of habitat. Indeed we started our reflection from the observation that when a room, or a building is poorly insulated, individuals often have to call on compagnies which will do the complete renovation of the building, or of the wall. In fact we have noticed that it is very complicated for a person to detect what are the real energy leaks points in a building, and to determine which windows have to be renovated for example, or which doors have to be changed, in order to improve the insulation of a building.



*Example of the complete renovation of the thermal isolation of a building*

All these reflexions finally permit us to answer precisely the question: who needs what because why?

So all people or companies wishing to invest themselves in an energy saving process, and in doing precision work for this, need a system that allows them to analyze precisely energy leaks in a building, for example in a roof or in a wall with windows . They need to know the energy leaks of a building to better improve in a targeted and precise way, the energy capacity of the building, and at a lower cost.

A system that would better identify the energy leaks of a building is therefore necessary so that everyone with their budget can participate at their level in the energy transition of building.

## Background Researches

Optimizing the insulation of a building has always been at the core of its design. Thus, finding where the insulation is the least efficient is essential. Over the years, one solution was developed that is widely used nowadays : the thermographic camera.



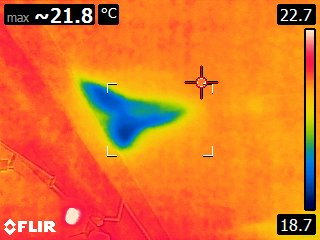
*The quite expensive*

*Fluke Ti25 thermographic camera*

These cameras can detect temperature variations using infrared radiations. The first thermographic camera was invented in 1929 by Hungarian physicist [Kálmán Tihanyi](https://en.wikipedia.org/wiki/K%C3%A1lm%C3%A1n_Tihanyi) for military purposes. It was slow and imprecise, taking one hour to produce an image.

Thermographic cameras come in two variations: cooled infrared detectors and uncooled infrared detectors. The first ones use a complex and costly technology to cryogenically cool the detector in order to avoid its own heat to interfere with the measurements. They are way more efficient but more expensive.

The main issue with this solution is its price as a camera like the one above can easily be over 1000 euros. The cheaper ones can be under 100 euros but will lose accuracy and efficiency.



*Example of energy leak in a wall (seen by thermographic camera)*

We can find lots of small and cheap thermographic cameras used to measure the temperature at a precise point, but they can’t achieve our goal: get a heat flux map of an entire wall.

As we can see in this picture, some of them are provided with a thermometer that can get the temperature in an inaccessible place

Thermographic cameras come with lots of different features and a wide range of price, so it can be difficult for someone to find the one that fits its demands. Moreover, they often lose accuracy when used outside, as they are very sensible to air heat variations. 

We can see that this little piece of engineering has been dominating the market for a very long time, without solving everything. Thus ,we would like to create something that would work on a different scale, something able to give the user temperature information on an outside wall that could be used alongside the thermographic cameras.

## Specification of requirement

Our design must be suitable for personal and professional usage. It must be flexible and should be adaptive: it must fit any kind of wall, regardless of its shape or size. It should endure any kind of weather and be as cheap as possible. Last but not least: it should be precise and efficient.



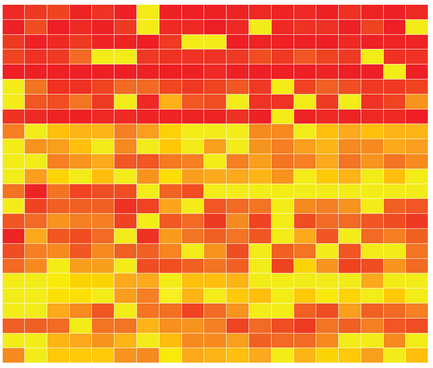
## Brainstorming of solution and production of the first design

At first, we thought about using thermographic cameras combined with drones, so that we could get a view of the walls from above. But as stated before, these can be imprecise and we did not consider them reliable enough. Moreover, the price of a camera and a drone would be way too much to fit within our budget.

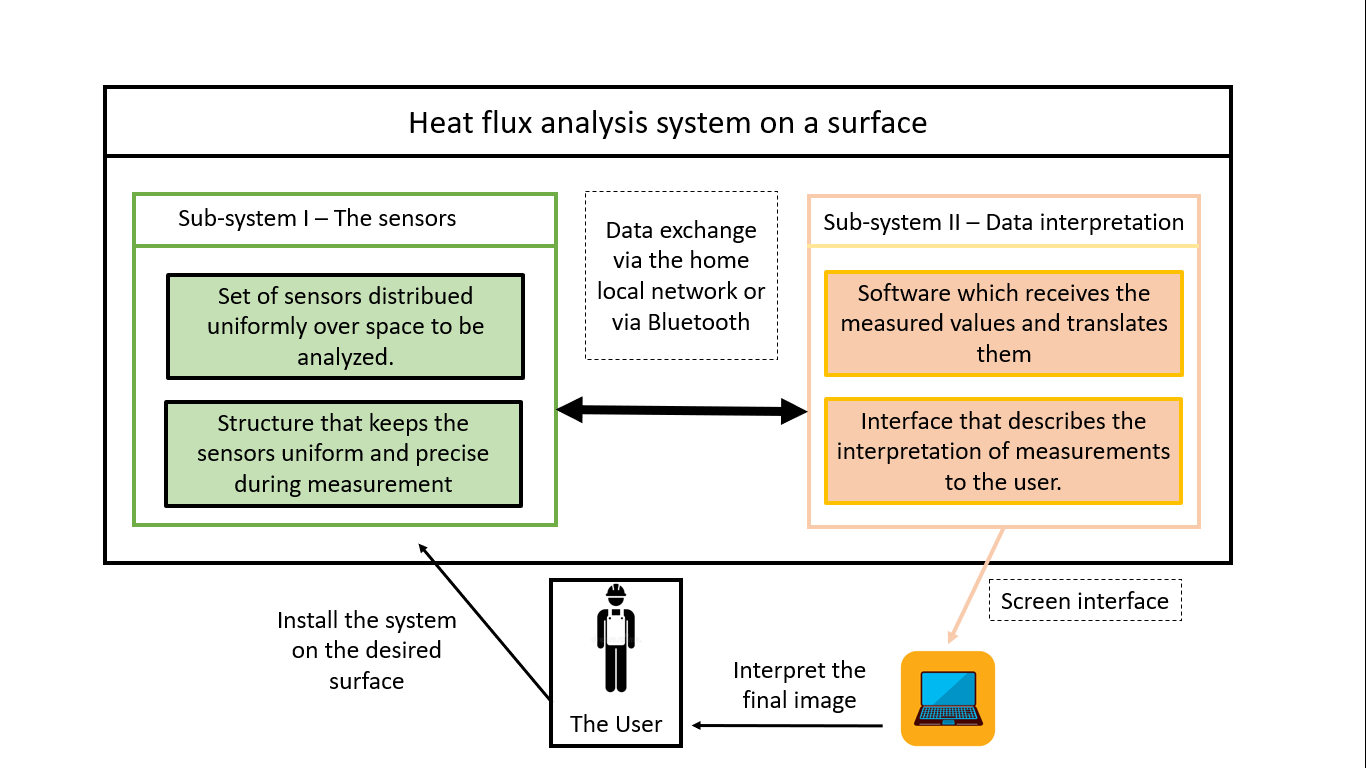
If we would not use thermographic cameras, we had to use sensors. Indeed, they can be quite efficient for a cheap price, but they can’t cover the surface of an entire wall. The idea of sensors in the wall crossed our mind, but it would prevent any kind of maintenance as they would be inaccessible. We decided not to use any kind of intrusive system.

In the end, the idea of a grid of sensors was the best thing we could think of. The ideal thing that we’d like to create is a wallpaper, filled-up with censors, that could be cut or folded to adjust to any sort of wall shape or size.

This wallpaper will be connected *via* Bluetooth and/or Internet to the user’s phone, and as each sensor will send information about the heat flux crossing it, it will display on the phone a thermal image of the wall, giving precise information about the heat losses affecting it.



## Block diagram of our solution

**Block diagram of our solution** 

You can see on this block diagram of our solution that our multifunction tarpaulin solution is broken down into two distinct parts.

The first sub-system is made of all the temperature, humidity and vibration flow sensors distributed over the surface of the tarpaulin. The second sub-system corresponds to the part of reception and interpretation of data. This second sub-part contains in particular the screen which will give the final indications to the user.

These two subsystems communicate with each other via the local web network: The data are captured by the sensors, then transmitted via the network to the second subsystem

## 6) First prototype: material, assembly, and code

To build our first prototype, we had the opportunity to use the BME680 sensor. It is a very complete sensor, providing data about temperature, humidity and pressure. As we couldn’t obtain a vibration sensor, we decided to use the three aforementioned type of data in our proof of concept. The BME680 comes with a python library that makes it easy to use. Finally, we wanted to show our results on a website, so we decided to use what we saw on the course: we used the RaspberryPi and the Apache server.

So the way the prototype would work is very simple : we would open an Internet browser with the php code, the code is calling our python code that gets the data, and the website uses this data to display the heatmap and other info.

Here is the python code :

*"""*

*Created on Thu Dec 3 14:34:33 2020*

*@author: gauth*

*"""*

**import** **time**

**import** **os**

**import** **board**

**from** **busio** **import** I2C

**import** **adafruit\_bme680**

**import** **numpy** **as** **np**

**import** **matplotlib.pyplot** **as** **plt**

**import** **seaborn** **as** **sns**; sns.set()

tempData = np.zeros((5,5))

*#create library object*

i2c = I2C(board.SCL, board.SDA)

bme680 = adafruit\_bme680.Adafruit\_BME680\_I2C(i2c, debug=False)

*#change this to match the location's pressure*

bme680.sea\_level\_pressure = 1013.25

**for** i **in** range(5):

**for** j **in** range(5):

tempData[i,j] = bme680.temperature

time.sleep(5)

**print**("moving sensor")

heatMap = sns.heatmap(tempData)

**print**("The humidity is %s" % bme680.humidity)

**print**("The pressure is %s" % bme680.pressure)

**if** os.path.exists("heatMap.png"):

os.remove("heatMap.png")

plt.savefig("heatMap.png")

As we can see we have to initiate the I2C ports of the RaspberryPi and we’re using the bme690 library. We have a loop to take the different measures, here we have 25 five, for the sake of simplicity. There is a delay of 5 seconds to move the sensor around.

The library used to plot the heatmap is the seaborn library. It is easy to use and easily converts a matrix of values to a heatmap plot. So, we save the values of the sensor in a matrix and then convert it into a heatmap. This plot is then saved as a .png file in the current working directory to be displayed by the website.

Here is the php code:

<?php

**echo**(exec('sudo python3 HeatBache.py',$result));

print\_r($result);

**echo**("<img src = heatMap.png>");

**echo** "<br>";

**echo**("The humidity is at 56.4% **\n**");

**echo**"<br>";

**echo** nl2br("The pressure is of 982hPa");

?>

The php code is really basic. We execute the python code to get both the heatmap and the data. Indeed, the exec function of php allows us to get the output of the executed code as an array of strings. This is the only data that we currently need in our proof of concept. You can see that the heatmap is saved and read as an image.

We tested the code bit by bit, meaning we tried to display only the image, we tried to run the python code by itself to see if the image was correctly created and that the data was properly captured. Every bit of the code works, we managed to have a website with the image loaded and the text corresponding to what we printed with the “echo” function.

However, the main problem we had with this code was the group rights. The apache server user is the www-data user, and he had to have the rights to execute the python code. We had already used this in the practical works, so we didn’t think it would be of any problem.

What happened is that the site would work, but the python code wouldn’t run. The heatmap didn’t update and the output didn’t show on the html page.

As a brute force solution, we added the www-data user to the “root” group. Even if that can be seen as a dangerous solution, we took the risk as it was minimal in our opinion. Unfortunately, that didn’t work out and the cold still wouldn’t run.



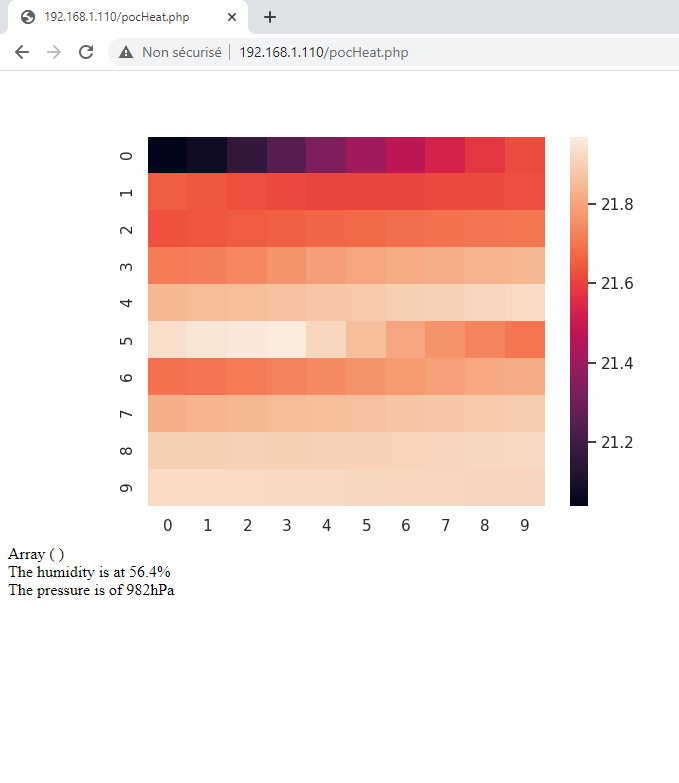
That’s when we started to look up this issue to find out some solutions. We found several forums with similar problems. There were lots of suggestions about adding “sudo” in front of the command in the exec function, which didn’t help. We used the chown command with the Python file to see if it was a problem of which user owned the file, but it didn’t change anything.

This really was confusing, as all the codes work individually, proving that everything should work, but this one thing hindered us.

Five links to similar problems can be found in the bibliography, each of them slightly different, but none of them worked. The python version issue couldn’t be a problem as I call the python script with “python3” to force the right version.

In the end we decided to take this issue into account as we couldn’t find a way around it. We still managed to produce some results, and to achieve our goal of proving the usefulness of our product.

# 7) First results



We can see here the results we managed to get. The two lines at the end were manually put and don’t correspond to what the python script should give us. You can see that the array corresponding to the output of the python code is empty, which isn’t right.

## Conclusion

Here are some improvement ideas we thank about, in order to develop our project to go further in the future:

* Improve the user interface with the development of a mobile application
* permit to add more sensors on the tarpaulin, to be even more precise

Think to use our project to capture others interesting data in a building These are only a few ideas, but the project is expandable and can easily evolve in different directions depending on the use you want to make of it.

To conclude this project, we could say that we learned a lot during this course and with this project. We are satisfied with the result even though we couldn’t get it to work entirely as intended. We hope that we’ll be able to use this new knowledge in our future jobs, as embedded systems can be found in every domain.

## Bibliography :

GITHUB LINK : https://github.com/Youbille/Poc\_EmbeddedSystems/tree/main

* <https://www.detecteur-thermique.fr/>
* <https://blog-travaux.primesenergie.fr/isolation/isolation-thermique/la-camera-thermique-un-appareil-pour-detecter-les-deperditions-thermiques>
* <https://en.wikipedia.org/wiki/Thermographic_camera>
* <https://www.reichelt.com/fr/fr/cam-ra-infrarouge-e95-msx-20-c--120-c-0-c--1500--flir-e95198920.html?PROVID=2810&gclid=Cj0KCQiAy579BRCPARIsAB6QoIaydrhK0VM1rckpfLlSkUzQgU3T3mvnCY1CLPy-J-TGW64CL-GoMAsaAouREALw_wcB>
* <https://www.dronezon.com/learn-about-drones-quadcopters/9-heat-vision-cameras-for-drones-and-how-thermal-imaging-works/>
* <https://en.wikipedia.org/wiki/Block_diagram>
* <https://outdoorworld.reviews/best-thermal-imaging-camera/>
* Php manual : https://www.php.net/manual/fr/index.php
* Seaborn library : <http://seaborn.pydata.org/>
* Savefig python : https://matplotlib.org/3.1.1/api/\_as\_gen/matplotlib.pyplot.savefig.html
* Php Python call problem :
  + <https://stackoverflow.com/questions/43164524/php-show-return-from-python-script>
  + <https://stackoverflow.com/questions/5697304/call-python-script-by-php-and-solve-errors>
  + <https://stackoverflow.com/questions/52714444/calling-a-python-script-from-php-does-not-work>
  + <https://raspberrypi.stackexchange.com/questions/69583/python-script-wont-run-on-php>
  + https://stackoverflow.com/questions/45949926/working-python-script-does-not-work-when-run-via-php/45997863