2021 W ESE 4009 1

EMBEDDED SYSTEM DESIGN PROJECT

SEMESTER: 4

INSTRUCTOR: Takis Zourntos

DATE OF SUBMISSION: 22 February, 2021

STUDENTS: Navyashree Garapati (C0765780)

Vaibhav Jivani (C0765530)

Archana Kalathil Venu (C0765532)

Geethu Anna Jacob (C0760834)

PROJECT PROPOSAL

PROJECT CONCEPT

THE PURPOSE OF THE PROJECT

Our project aims to provide on-site real-time measurements of various parameters to a cloud-based system that investigates the energy performance or energy efficiency of Industrial Refrigeration System (IRS). The importance of such an energy monitoring system stems from the global need to save energy, which has got several implications, like:

1. Reducing energy costs
2. Reducing carbon emissions and the environmental damage that they cause.
3. Reduce risk - the more the energy we consume, the greater the risk of an energy crisis or an energy price hike in the very near future.

This project plans to create a consistent technology that every IRS could rely on in monitoring their overall energy usage and to make informed decisions concerning energy efficiency. The system will create timely reports on the overall energy performance and that way, provides feedback on any improvements made through optimized energy saving/conservation technologies. This tool aims to help implement the following:

- Metering our energy consumption and collecting the data

  Detailed interval energy consumption data on a weekly or monthly basis makes it possible to see patterns of energy wastage.

- Finding new strategies and opportunities to save energy, and estimating how much energy each strategy could save.

- Targeting the opportunities to save energy

  Once we have identified the opportunities, we need to take actions to target them. This might require investing time and money, say, for upgrading insulation or equipment, etc.

- Tracking our progress at saving energy

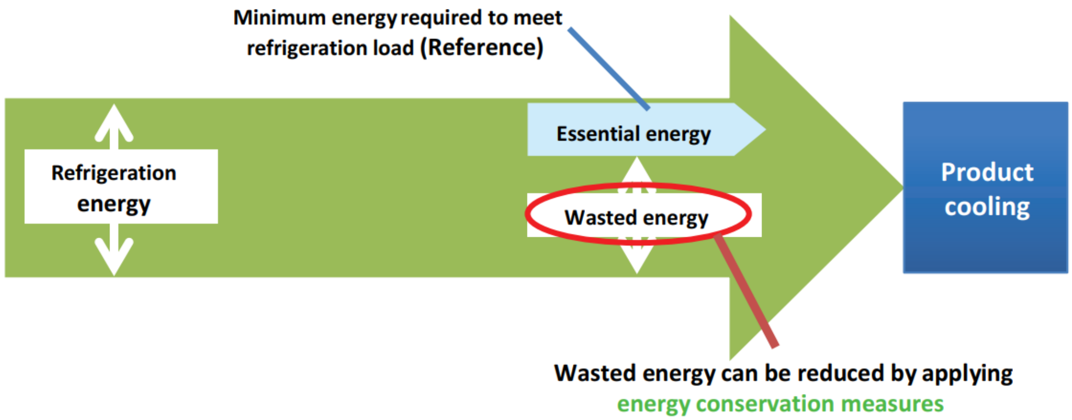
  This can be done by comparing the past and present energy performance factors in the reports generated and stored in the cloud (with all the gathered data) on a time-to-time basis.

CONCEPTUAL OVERVIEW

Our project essentially consists of an IoT network of sensors, aimed to work in conjunction with the energy performance monitoring system. On-site measurement of several dynamic parameters like temperature, humidity, occupancy etc is made with this network and required information is then sent over a suitable protocol like HTTP or MQTT to the AWS (Amazon Web Services) platform for further processing and storage. The project is especially designed for Refrigerated Distribution facilities, Refrigerated Processing Facilities, Refrigerated Processing Systems and Refrigerated Storage Warehouses.

This real-time data is used for the calculation of various refrigeration loads as well as the total essential energy (this is the minimum energy required by a facility when using the best available technology existing in the market). This total essential energy is later compared with the actual(billed) refrigeration energy to find the Benchmark Energy Factor (BEF), which is a measure of the total energy wastage in the system.

BEF = Overall Refrigeration Energy/ Total Essential Energy



SYSTEM LEVEL ARCHITECTURE

The overall HVAC system in any facility might consist of one or more suction systems (the refrigeration system of a zone, or group of zones, with their associated evaporators that feed refrigeration vapors to an individual or group of compressors). Each zone can either be a cooler or a freezer (the zone is considered to be a freezer if their target temperature is less than -9.4 degree celsius, else a cooler). In addition, the facility has a loading dock area where the products are loaded on and off the trucks for transportation.

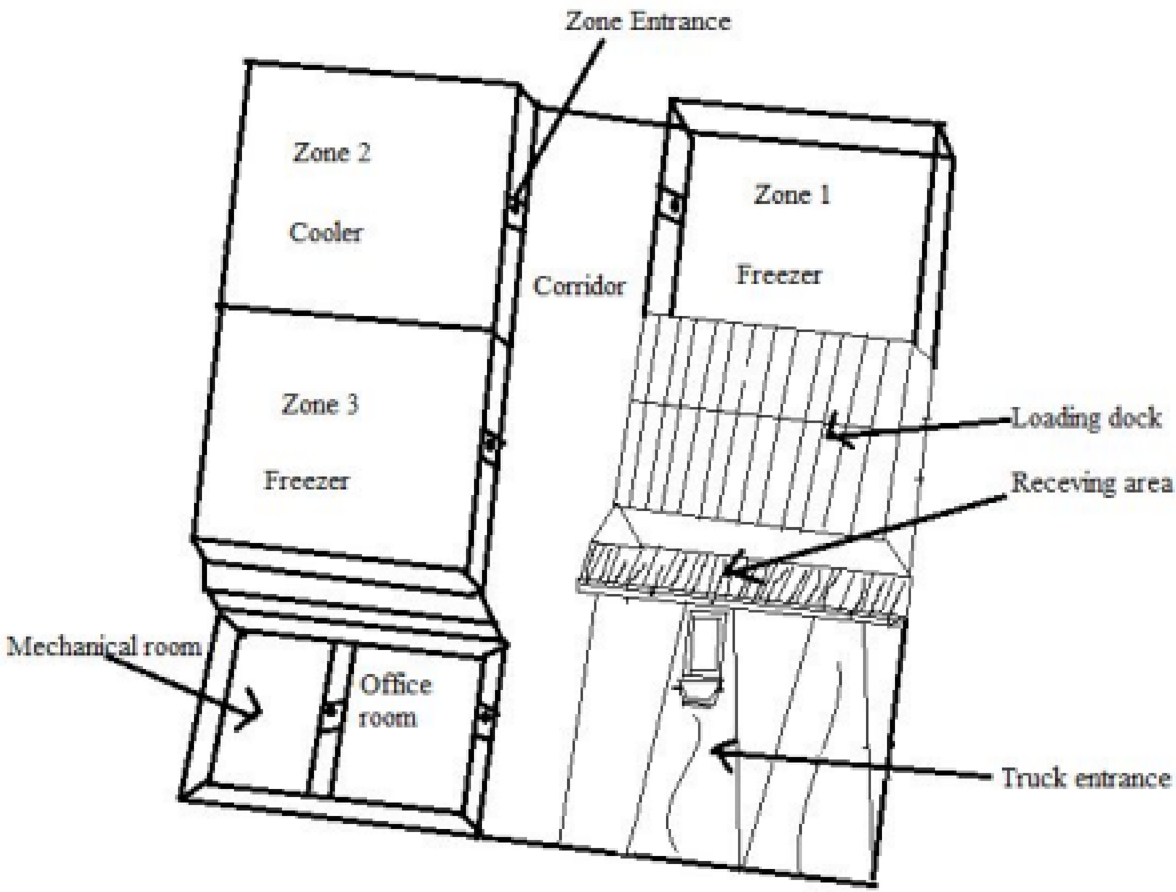


Fig : Typical Industrial Refrigeration facility layout

In this project, our main focus is on measuring the dynamic parameters within a zone. For this, we are going to use an IoT (Internet of Things) system based on a network of various sensors for measuring a particular zone's temperature, occupancy, etc. This forms the base layer or the device layer; which is responsible for collecting all the required data. We use several sensors for this purpose as will be described in the following sections. The second level might consist of several lower level microcontroller boards (teensy boards) connected to these devices which aggregate the data, make the conversions, if any. The processed data can either be manipulated for various control operations or can be sent over to the higher level microcontroller through a Bluetooth network. In addition, the higher level microcontroller is connected to a camera, HDMI display and a relay circuitry for lighting control. The camera will help us find the occupancy of the zone. The microcontroller employs computer vision and image processing for counting the number of people entering and exiting from the zone. The time the zone is occupied is also tracked through a software timer running on the board. This third layer acts as a hub and communicates all the gathered data with a cloud database (like Amazon dynamo DB). At a further stage of the project, we may build a web interface connected to the dynamoDB for displaying live data or measurements within the zone. There are multiple ways one can achieve this. However, one of the most modernized ways of building applications today is using a Serverless Architecture. We may host our website on S3.

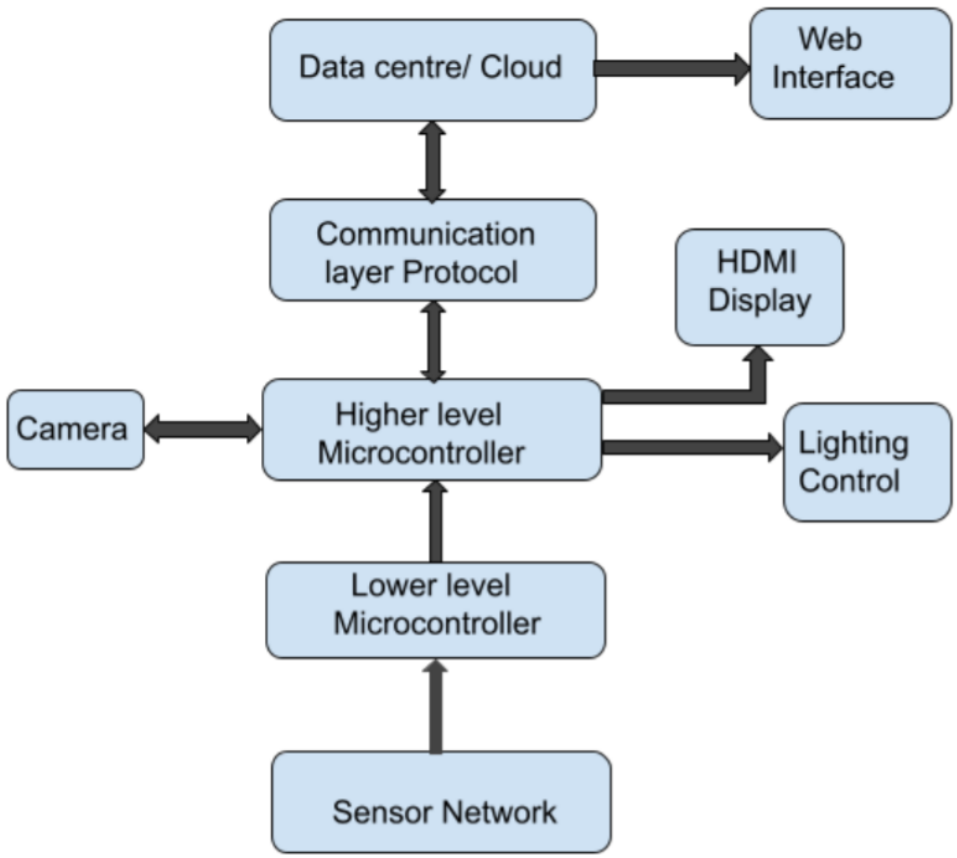


Fig 1: General block diagram of the project

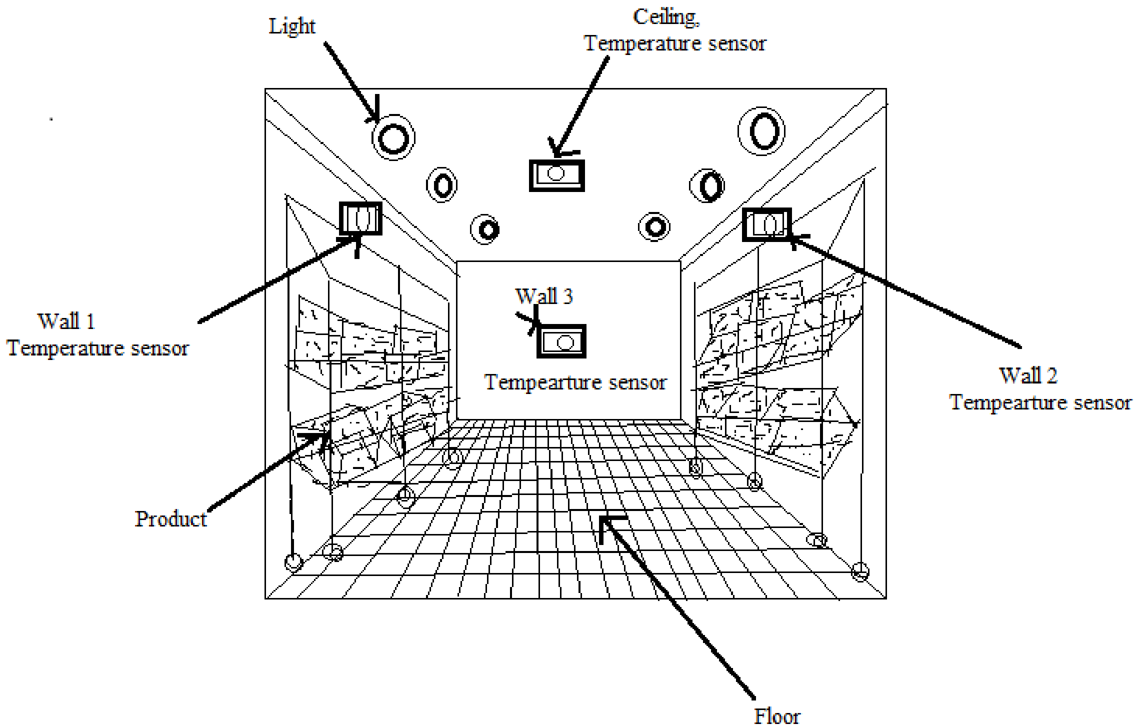


Fig : Interior view of the zone space showing few of the sensors

We employ four temperature sensors on four walls of the zone; each of them connected to separate lower level microcontroller boards. One temperature sensor is mounted on the ceiling together with the higher level microcontroller unit The camera and lighting control as well as the display is connected to the same unit.

We also try to predict the failure or lower efficiency of the evaporator fan motors operating in the zone by recording their sound. We may use a microphone sensor to record live evaporator fan motor running sound. This is also interfaced via one among the wall mounted lower level microcontrollers, which then compares this sound profile with the existing database (it contains different sound profiles for normal motor running conditions as well as for motor running at lower efficiencies or faulty conditions). Once any variation in the existing amplitude or frequency is observed, the microcontroller board detects it. It then alerts the central hub (Higher level microcontroller) which then creates an alert on the display.

Also, a motion sensor will be placed near the zone door. If the motion sensor detects a human in its range, it will automatically turn on the lights in the zone. As the lights turn on, the ceiling mounted camera starts recording and the microcontroller connected to it will process the images and counts the number of people entering and exiting the zone space.

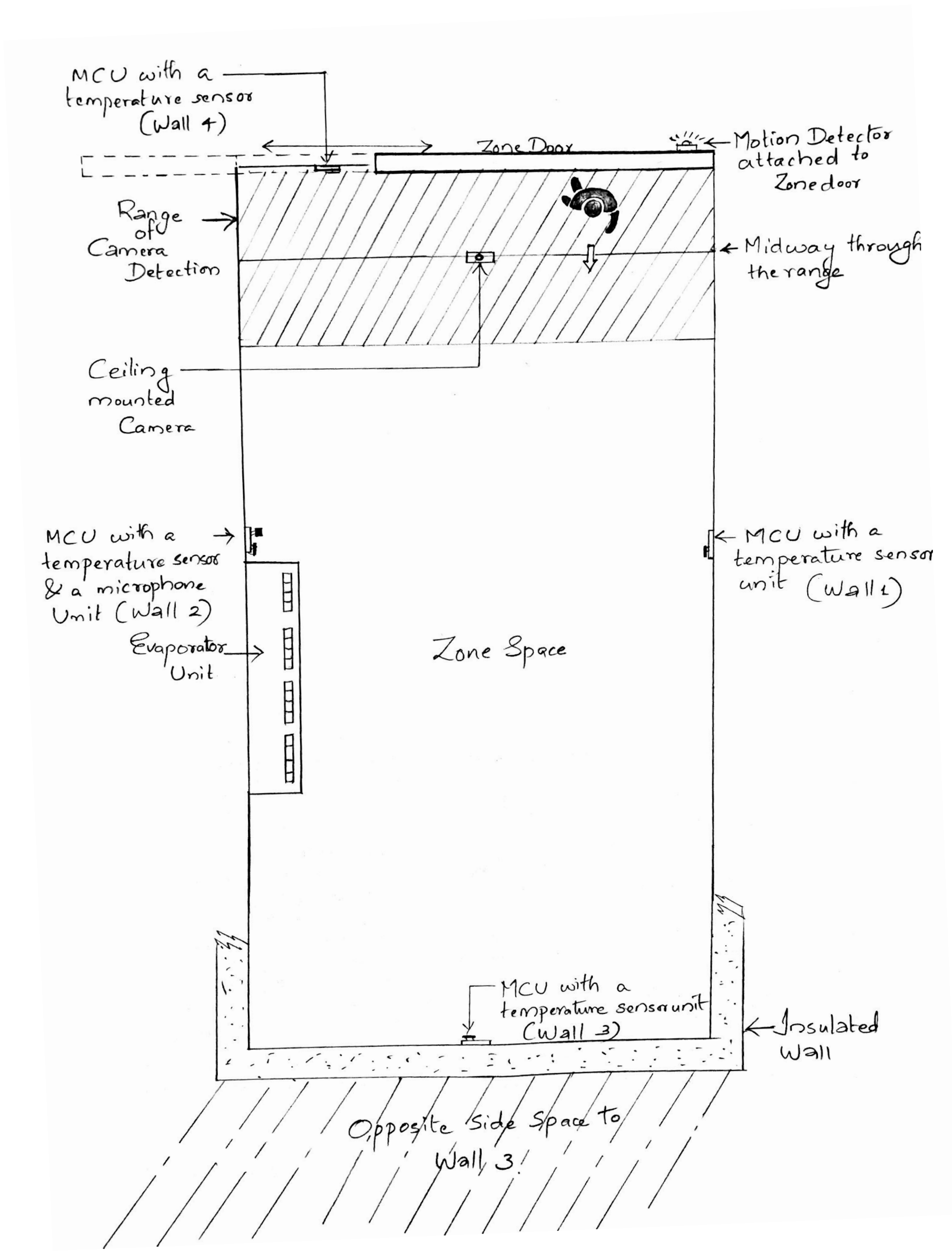


Fig: Top view of the zone

The HDMI display displays the real-time measurements in the zone at any given time. The following data may be displayed :

- The current temperature of the zone
- The number of people in the zone at the time
- The total hours of occupancy till the time
- Whether the evaporators for the zone are running fine or not.

The display also creates alerts for any faulty motor running conditions as well as if the temperature of the zone at any point gets greater than a threshold. This helps to ensure consistency in temperature all over the zone at all times.

HARDWARE DESIGN

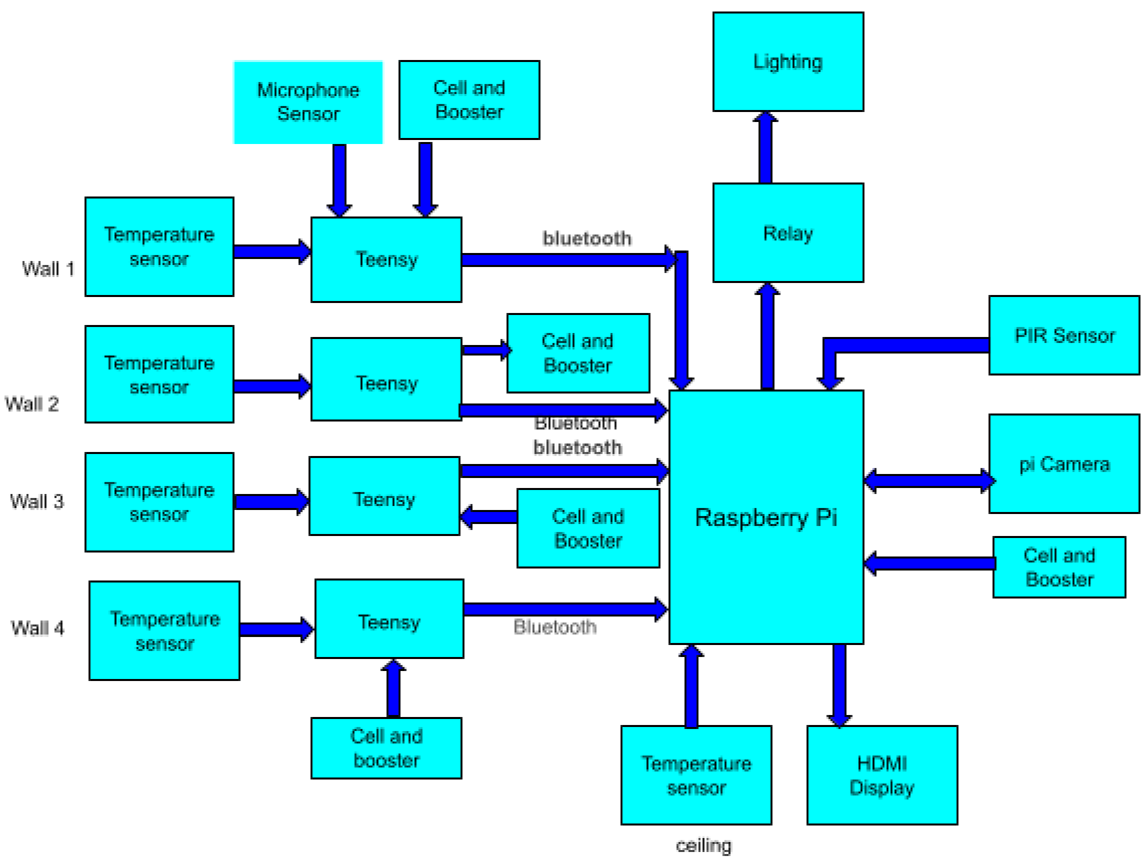


Fig : Hardware design for the project

We are using temperature sensors fixed to each of the four walls and to the ceiling. There is a microphone sensor fixed to the wall near the motors so that it can record the motor sound and we can check for any abnormalities in the motor operating conditions.

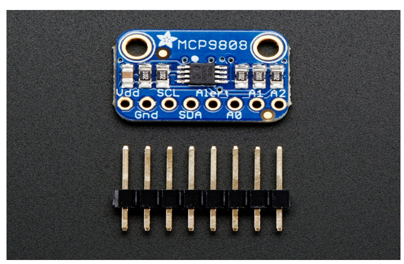
Each of the sensors are interfaced to the teency board via i2c protocol. All the teency boards are then wirelessly connected to the Raspberry pi which act as the central hub through a bluetooth mesh network.

A pie camera mounted on the ceiling near the door is also interfaced to the Raspberry pie to keep track of the number of people entering and exiting the zone. A PIR sensor at the door triggers the Pi in case of any human intervention into the zone and the Pi then controls the lighting inside the zone via a relay circuitry.

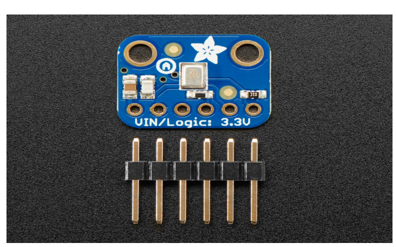
For powering each of the teensy boards as well as the raspberry pi, cells with power boosters are used.

Components Required

- Temperature sensors - 5
  We are using MCP9808 high accuracy temperature sensor from adafruit. It has a typical accuracy of a typical accuracy of ±0.25°C over the sensor's -40°C to +125°C range and precision of +0.0625°C. They work great with any microcontroller using standard i2c. Its voltage range is between 2.7v to 5.5v and its operating current is 200 micro A.



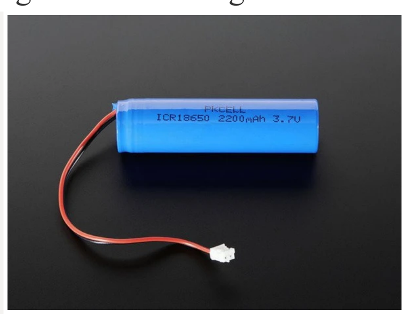
- Microphone Sensor - 1
  Adafruit I2S MEMS Microphone Breakout SPH0645LM4H. The I2S is a small, low cost MEMS mic with a range of about 50Hz - 15KHz, good for just about all general audio recording/detection. It's a 1.6-3.6V max device only.



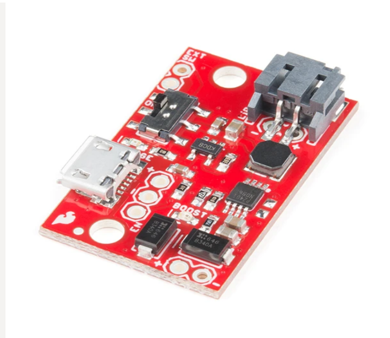
- Teensy Board 3.6/4.1

  It has an ARM Cortex-M4 operating at 120 MHz. It supports near real-time clocking; especially useful in interrupt generations to a higher level microcontroller, compatible with Arduino IDEs

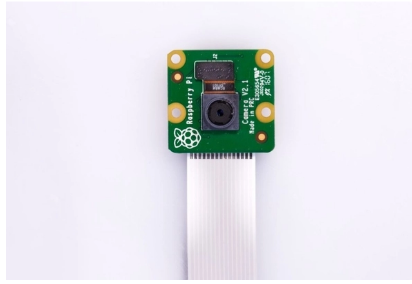
- Battery: Lithium Ion Cylindrical Battery - 3.7V 2200MAH - 5
  This cell can provide 2C of peak current (4400mA). These batteries are not designed to sustain such high loads, it is better to keep the current under 1A. It has the following specifications.
  - Nominal cell voltage of 3.7V-3.9V
  - It has a Standard Charge Current of approximately 0.2C / 0.5A
  - Max Charge Current: 1C / 2.2A
  - Charge Cut-Off Voltage: 4.2V
  - Standard Discharge Current: 0.5C / 1.1A
  - Discharge Cut-Off Voltage: 2.75V



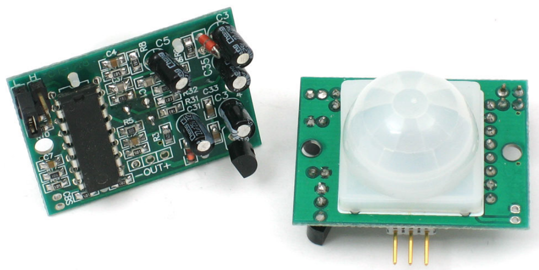
- Cell Booster: SPARKFUN LIPO CHARGER/BOOSTER - 5V/1A - 5
  - Charger, microUSB, 500mA
  - Booster, 5V, 1A output
  - Form factor for our 1,000mAh batteries
  - Disabled current less than 10uA



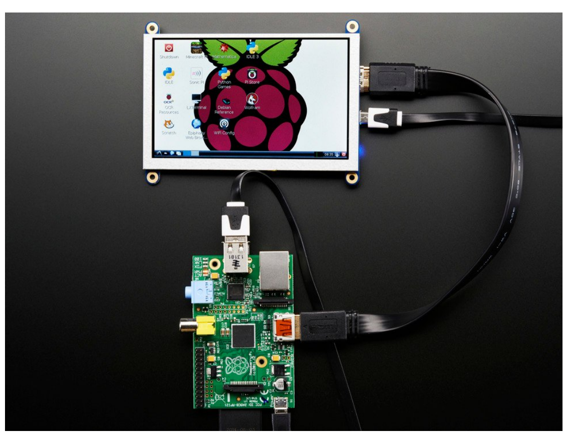
- Raspberry pi 4 - 1
  The Raspberry Pi 4 Model B is the newest Raspberry Pi computer made. It has an updated 64-bit quad core processor running at 1.5GHz with built-in metal heatsink, USB 3 ports, dual-band 2.4GHz and 5GHz wireless LAN, faster Gigabit Ethernet, and PoE capability via a separate PoE HAT. It is available with 2GB or 4GB RAM. It has an operating temperature between 0 degree to 50 degree celsius. It has an input voltage of 5V dc via usb C connector as well as through GPIO pins.
  
- Raspberry Pi Camera V2 - 1
  The v2 Camera Module has a Sony IMX219 8-megapixel sensor. It has a pixel size of 1.12 µm x 1.12 µm. It has a horizontal field of view of 62.2 degrees and vertical field of view of 48.8 degrees.
- Bluetooth module



- PIR Sensor - 1
  This PIR sensor from adafruit runs on 5-12V power. Its digital signal output is 3.3V high/low(when motion detected). Sensing range is about 7 meters(120 degrees corner).



- HDMI Display - 1
  The HDMI display from adafruit. We can power the display from a usb cable. With the default 5" 800x480 display and 50mA backlight current, the current draw is 500mA total. It is possible to reduce that down 370mA by running the backlight at half-brightness (25mA). This display is not a touch screen.



- Any 5V electromechanical relay - 1

SOFTWARE DESIGN

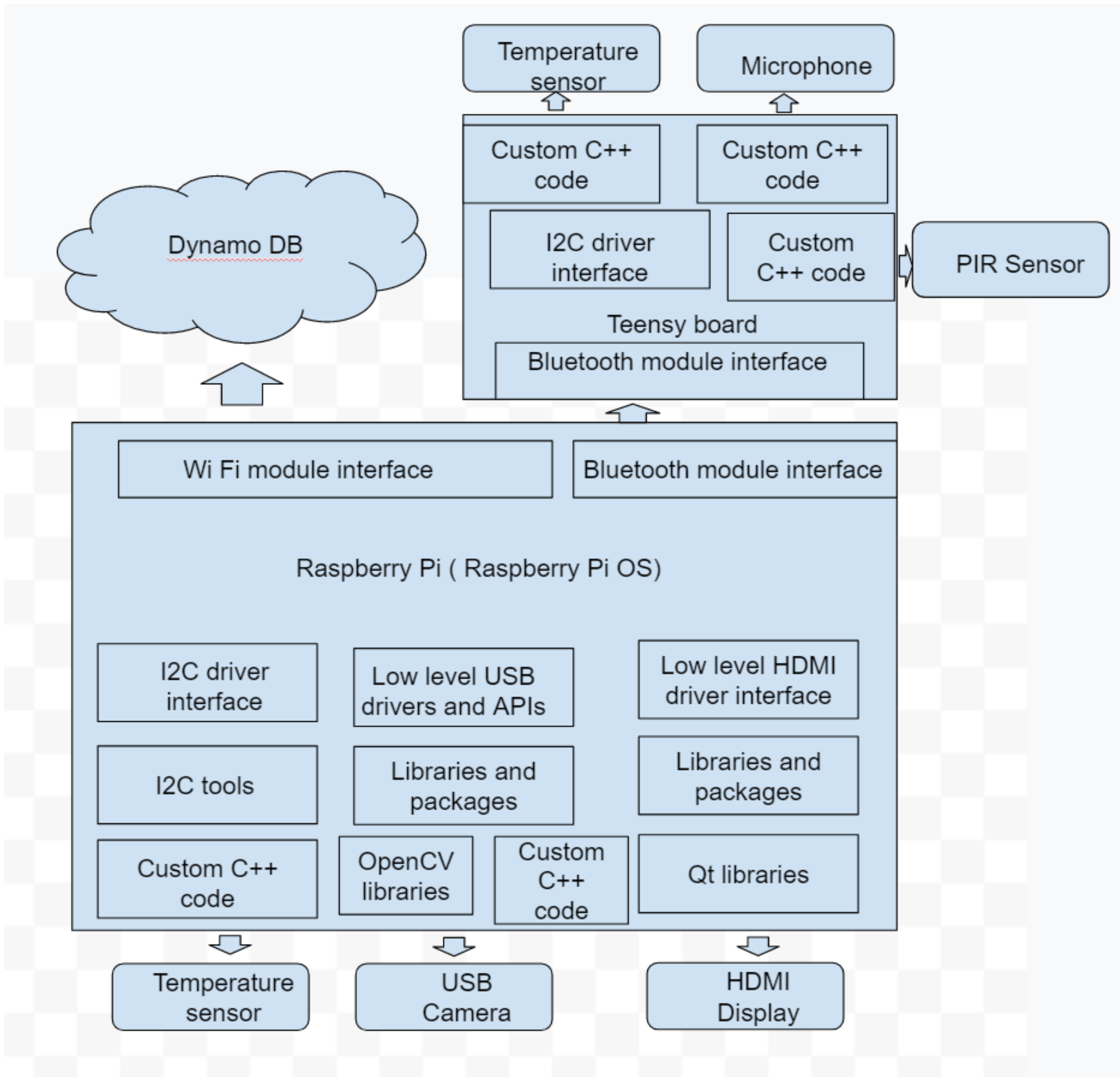


Fig : Software design of the project

In the project, we make use of the I2C serial interface for communicating with the temperature sensor, microphone as well as the PIR sensor. For keeping track of the average number of people in the zone as well as the time for which the zone is occupied, we make use of a camera controlled by the Raspberry Pi and OpenCV object detection and tracking algorithms based on C++. The camera is chosen to provide more accurate information compared to all other people's counter technologies.

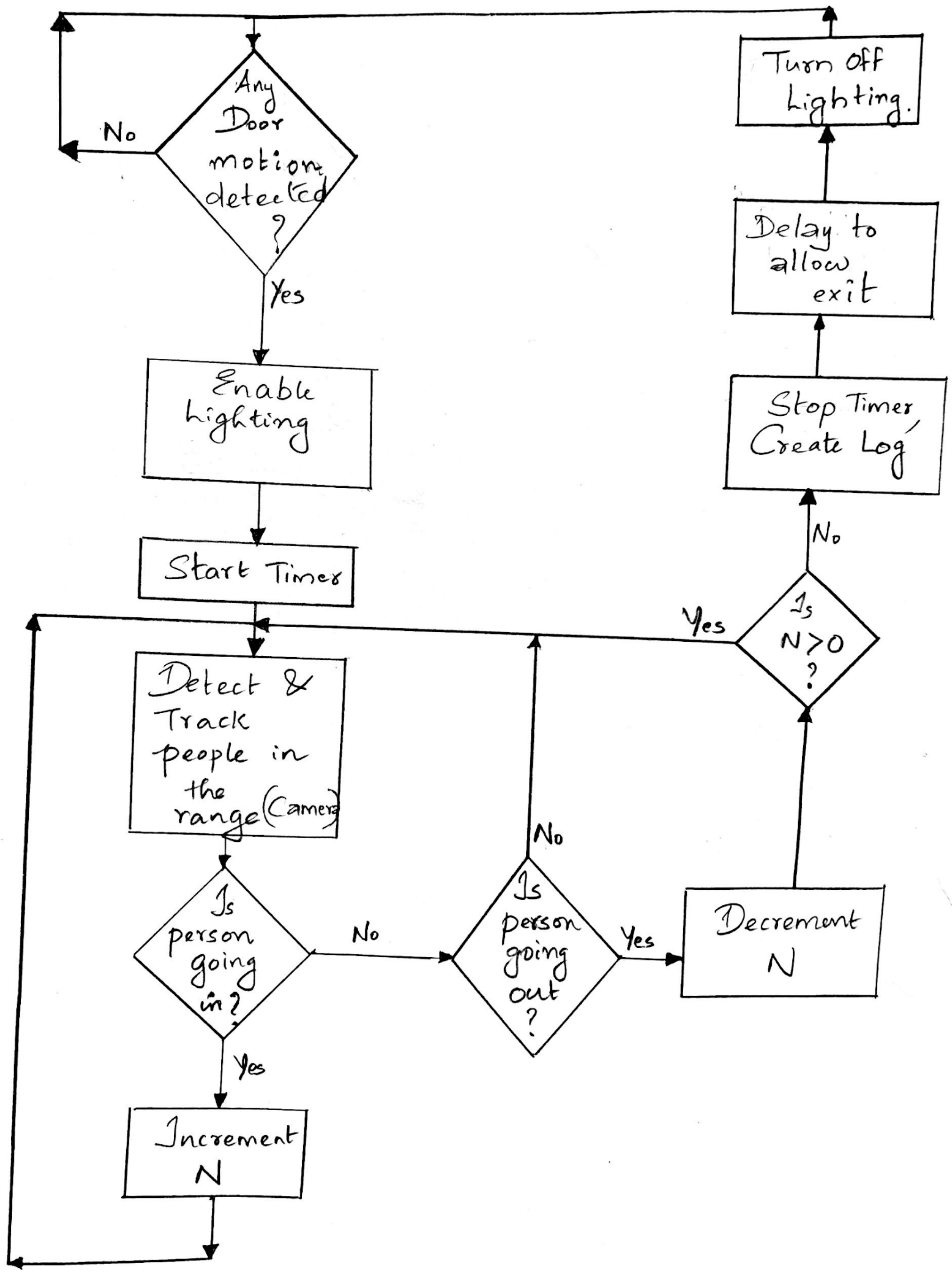


Fig : Flowchart explaining the occupancy tracking algorithm

Initially, the PIR sensor attached to the zone door acts as a trigger to wake the Pi up from its sleep mode, once any motion of the zone door is detected. It will control the relay to turn on the lighting in the zone, so that the camera can now start detecting and tracking persons. A software timer is also started to keep track for the time the zone is occupied.

For counting, we leverage both the object detection as well as object tracking algorithms. We might need to download several libraries including OpenCV for this purpose. Since object detection is a fairly computationally intensive process, we do that only every N frames to check if a new object has entered the range of detection or not. Then for the rest of the time, we track the object based on its previous coordinates and determine the direction of motion.

The above flowchart explains the basic algorithm used for counting. Here, N gives the number of people inside the zone at any given time. We update the value of N after every hour (or any time frame of interest). At the end of each day, will average out the total N over 24 to get the average hourly N for the day.

We will start a timer as soon as someone enters the zone and will keep the timer running until N=0 inside the zone. This log is stored and is later on totalled up to get the total number of hours the zone gets occupied. Once the Pi detects N to be zero, it will send control signals to disable the lighting and in itself will go into a low power mode.

The acoustic signals recorded via the microphone will be sent over to the AWS cloud through the raspberry Pi. The signals could be filtered or transformed (by applying algorithms to enhance the result) to identify any differences in the frequency spectrum of the sound signal. The signal may then be sent over to the AWS cloud every hour (or any time frame of interest). A cloud-based motor condition monitoring system is then implemented where a machine learning algorithm is trained with the previously collected data to filter out the signal and then to analyze its frequency components. There might be sufficient information not only to distinguish between healthy and faulty states, but also to determine the specific fault.

GANTT CHART

Gantt chart represents overall progress and planning of the project. For the project, we have used ClickUp application which has a lot of features and is very organised for project planning.

The following link redirects to the chart.

https://share.clickup.com/g/h/84x5n-28/13e82c343e6a032

APPENDIX

The following github links provide the datasheets for the various components that we intend to use in the project.

Temperature Sensor:
https://github.com/Capstone-redefined/Smart-Sensing-network-through-IoT-for-IRS/blob/main/temprature%20sensor.pdf

Microphone Sensor:
https://github.com/Capstone-redefined/Smart-Sensing-network-through-IoT-for-IRS/blob/main/microphone.pdf

Battery:
https://github.com/Capstone-redefined/Smart-Sensing-network-through-IoT-for-IRS/blob/main/microphone.pdf

Booster:

- https://github.com/Capstone-redefined/Smart-Sensing-network-through-IoT-for-IRS/blob/main/mcp73831.pdf
- https://github.com/Capstone-redefined/Smart-Sensing-network-through-IoT-for-IRS/blob/main/pam2401.pdf

Pi Camera:

- https://github.com/Capstone-redefined/Smart-Sensing-network-through-IoT-for-IRS/blob/main/rpi_MECH_Camera2_2p1.pdf
- https://github.com/Capstone-redefined/Smart-Sensing-network-through-IoT-for-IRS/blob/main/rpi_SCH_Camera2_2p1.pdf

PIR Sensor

- https://github.com/Capstone-redefined/Smart-Sensing-network-through-IoT-for-IRS/blob/main/detectionelement.pdf
- https://github.com/Capstone-redefined/Smart-Sensing-network-through-IoT-for-IRS/blob/main/lenspir.pdf
- https://github.com/Capstone-redefined/Smart-Sensing-network-through-IoT-for-IRS/blob/main/motion%20detector%20ic.pdf
- https://github.com/Capstone-redefined/Smart-Sensing-network-through-IoT-for-IRS/blob/main/PIRSensor-V1.2.pdf

Teensy Board 3.5

- https://github.com/Capstone-redefine/Smart-Sensing-network-through-IoT-for-IRS/blob/main/K64P144M120SF5.pdf
- https://github.com/Capstone-redefined/Smart-Sensing-network-through-IoT-for-IRS/blob/main/K64P144M120SF5RM.pdf