

Cabin Monitor

Petri Hautamäki, Paavo Määttä, Juuso Puolakka, TVT19SPL
Information Technology, Option of Device and Product Design

Introduction

The aim of this project was to create a monitoring system for a cabin that would send data to a website and it could then be monitored there. Second objective was to practice needed skills to be able to develop IoT devices and systems.

Objectives

The objective was to collect data from the sensors at the cabin and store it into the database and then show it on the website. Second objective was to detect motion inside the cabin and take a photo if motion is detected. The system has to recognize if there is a human in the photo and send a message to the website telling how many or if any humans have been detected. The purpose of the website was to show the collected data and had to have the option to select a time period for the data to be shown. User had to be able to disable the motion sensor through the website.

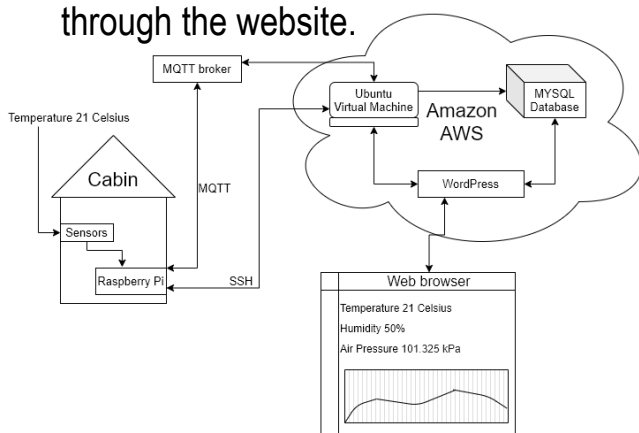


FIGURE 1 The System Diagram

Telecommunication Application Project

ECTS Credits: 6

Date of Publication: 2020, Autumn

Instructors: Kari Jyrkkä, Jukka Jauhiainen

Methods

Raspberry Pi 3B was the center piece for the system in the cabin end. All the sensors, the camera and the GPRS-shield were connected to it. GPRS-shield was used to connect the device to the internet. Sensors included were temperature, humidity, air pressure and motion sensor. Database and the website were located on a virtual machine in the Amazon AWS cloud server. Ubuntu and the LAMP stack were installed on the virtual machine. MQTT protocol and SSH were used in the communication between the Raspberry and the virtual machine. The database was created using MySQLWorkbench and MobaXterm. The website was created using WordPress, HTML and PHP. Human recognition was done with a machine learning model using the OpenCV library. Python was used to code the programs in the Raspberry Pi and in the virtual machine.

Location	Temperature Inside(°C)	Temperature Outside(°C)	Time
Paavon Mökki	29	25	2020-12-08 23:01:32
Paavon Mökki	30	27	2020-12-08 23:02:22
Paavon Mökki	31	29	2020-12-08 23:31:28
Paavon Mökki	31	29	2020-12-09 13:28:07

FIGURE 2. Latest temperatures shown on the website

Results

The system worked as planned. Values taken by the sensors were automatically sent to the database every ten minutes and then shown on the website (FIGURE 2).

Motion sensor was successfully detecting motion and recognized if there were humans in the picture also sending the message to the database. Website was working correctly showing the latest temperatures and messages on the home page. On the other page the user could pick a time period by selecting a starting and ending date for the data to be shown. User can disable the motion sensor through password protected admin page.

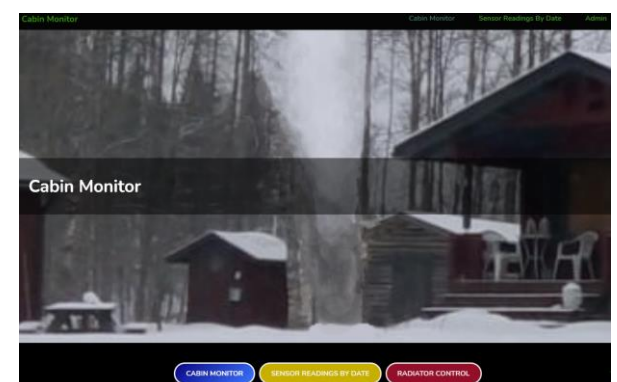


FIGURE 3. The Website

Conclusions

Project was a success, but better sensors could have been chosen for better results. Website functions could have been more diverse. The photo taken by the camera could also be shown on the website.

References

- <https://www.geeksforgeeks.org/pedestrian-detection-using-opencv-python/>
- <https://pypi.org/project/RPi.bme280/>
- <https://www.circuitbasics.com/raspberry-pi-ds18b20-temperature-sensor-tutorial/>
- <https://www.instructables.com/Raspberry-Pi-Human-Detector-Camera-Flask/>
- <https://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-the-raspberry-pi/>
- <https://github.com/eclipse/paho.mqtt.python/>