

2021

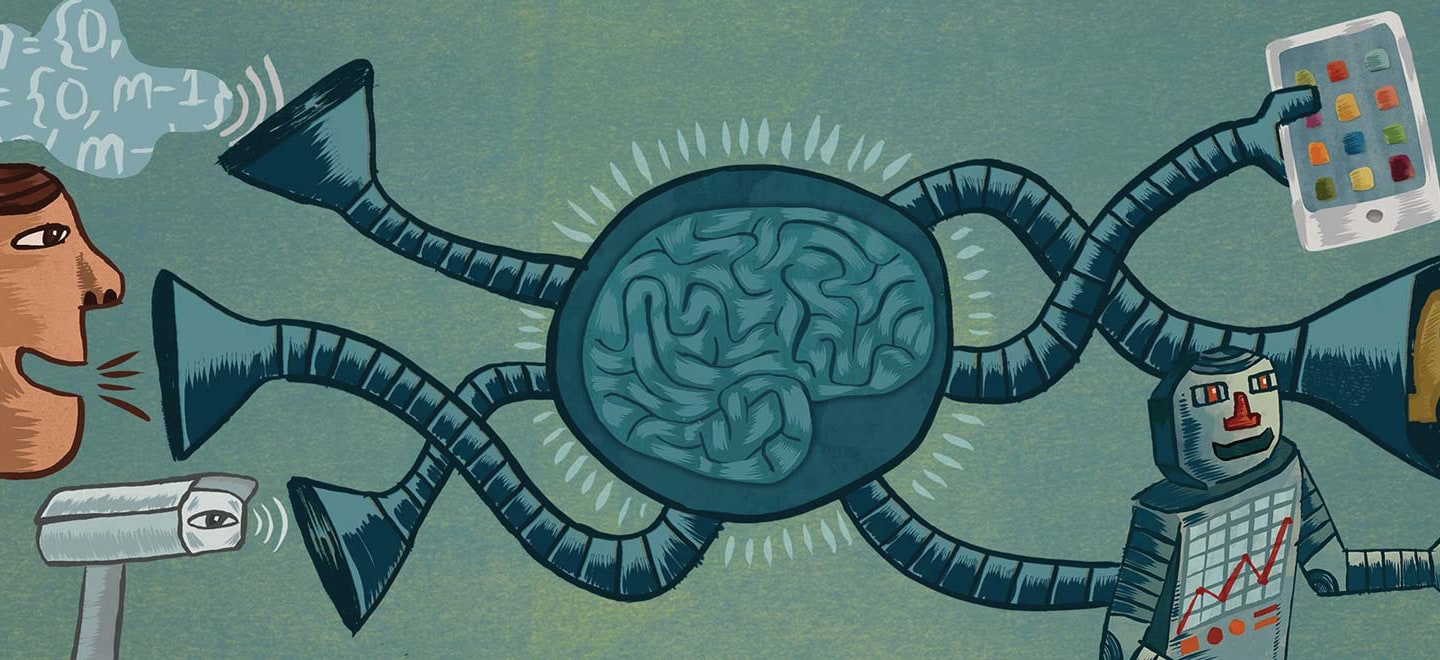
Group members:

* Guga Esitashvili - 822090276
* Tea Gamrekelashvili - 822059336
* Ana Dzamelashvili - 822068449
* Nino Nonikashvili - 822059388
* Uta Buziashvili - 822059947

Group Advisor: David Chxaidze

1/15/2021

Intelligent Automatization System



Contents

[Abstract 2](#_Toc72018656)

[Introduction 3](#_Toc72018657)

[Description 3](#_Toc72018658)

[Server 4](#_Toc72018659)

[Hardware Codes 6](#_Toc72018660)

[Printed Circuit Boards 9](#_Toc72018661)

[Assemble Hardware Components 10](#_Toc72018662)

[Fire Detection 10](#_Toc72018663)

[Gas Leak 13](#_Toc72018664)

[Water Leak 14](#_Toc72018665)

[Door Lock 15](#_Toc72018666)

[Light Control 16](#_Toc72018667)

[Glass Break 18](#_Toc72018668)

[Influxdb 19](#_Toc72018669)

[Grafana 20](#_Toc72018670)

[Application 21](#_Toc72018671)

[Website 22](#_Toc72018672)

[Timeline 23](#_Toc72018673)

[Conclusion and Recommendations 24](#_Toc72018674)

[Technical Specifications 25](#_Toc72018675)

[Hardware 25](#_Toc72018676)

[Software 26](#_Toc72018677)

[Website 26](#_Toc72018678)

[Team-member Roles 26](#_Toc72018679)

[Guga Esitashvili 26](#_Toc72018680)

[Tea Gamrekelashvili 27](#_Toc72018681)

[Nino Nonikashvili 28](#_Toc72018682)

[Ana Dzamelashvili 28](#_Toc72018683)

[Uta Buziashvili 29](#_Toc72018684)

[References 29](#_Toc72018685)

[Appendix 30](#_Toc72018686)

# Abstract

Intelligent Automatization System is an integrative and easy to use, smart house system based on the Internet of Things (IoT). Whole project depends on the IoT server to enable user-to-device communication. It also ensures that data coming from sensors and various other devices are correctly written in database, visualized in Grafana, and therefore presented in Web Application. According to the data that is taken from the database, system is activating lockers and alarms as needed. This smart house system creates a friendly environment for people with various disabilities by making different house tasks more manageable.

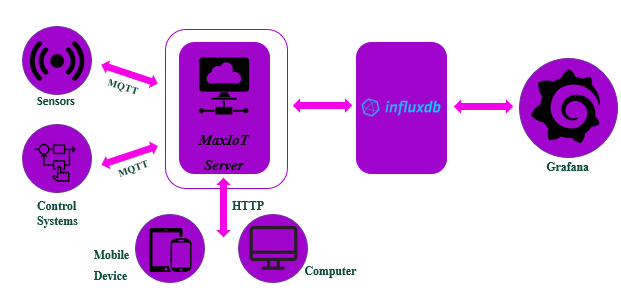
# Introduction

Scientists started house automatization in the early 20s century by introducing labor-saving machines, such as washing machines and refrigerators, to humanity. House automatization is oriented to the comfort and safety of the customer. For almost more than a century people are trying to make their everyday lives easier by making technology and all these machines more user-friendly and accessible. Nowadays, house automatization is a rising trend, as it is not only making human lives effortless, but it is also reducing everyday expenses.

Even though today there are so many companies working on this task, there is still much more left to improve and facilitate. Still, there are some doubts about the safety of these smart houses and the accuracy of the system. Also, one part of society perceives smart houses as a luxury and not a necessity. However, automatizing the house makes the owner’s life not only more comfortable but more carefree about the general state of the place of residence.

# Description

The main goal of our project was to create, integrative and easy to use, smart house system based on the Internet of Things (IoT). Our automated system gives the user ability to manage all the home devices from one place and remotely control home functions using our application. With this application, users can control light and door lock systems from anywhere. IoT gives our application flexibility to add new devices and appliances. Our project is not only focused on the comfort but also the safety of the customer. We are maximizing home security by using alerts for the break of the window and by detecting life and house damaging factors such a fire, CO, etc. Furthermore, our intelligent automatization system can also increase energy efficiency by controlling the heating and cooling of the house.



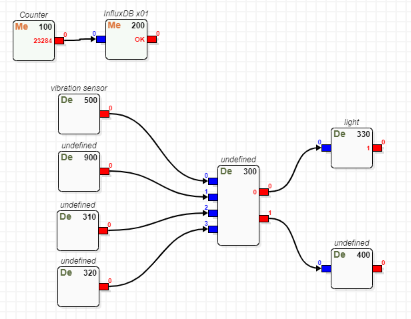
As you can see above on the system block diagram, assembled sensors and control systems should be connected to the Max-IoT server using the MQTT protocol. Information goes to Influxdb from the server and creates a database. To get more visualized data Influxdb sends information to Grafana. Also, the server can be controlled by users from both their mobile devices and computers.

# Server

As you may already know, a big part of our project depends on the server to enable user-to-device communication. It also ensures that data coming from our sensors and various other devices are correctly written in our database, visualized in Grafana, and therefore presented in our Web Application.

Our MaxIOT Server was provided by our mentor David Chxaidze and utilized by us. First of all, our main job was to get hosting service up and running, since we were going to develop a completely remotely accessible environment, we decided not to use local hostin, using RaspberryPI, rather use one of the online hosting services. We decided to go with Hostinger hosting service, since it was the most user friendly and easy to use hosting service that could host on the Ubuntu system.

After we were done with choosing our server and configuring its options and preferences, it was time to upload our MaxIOT server and access it on the remote host. We began with uploading MaxIOT files to the server, followed by installing python on the remote linux machine to be able to use different services. Next, we added root user, and 1 extra user, in order not to lose track of the files and items on the actual server. We spent almost 1 week finalizing everything and making sure they world as they should, and also, we faced a couple of issues during the installation. For one, the server crashed a couple of times during installation, and we later found out that it was a DigitalOcean maintenance issue. Later when we thought we had installed everything, we almost got hacked. We were getting brute force attacks on our ssh port, and the digital ocean forcefully would shut down our port, and we could no longer continue working on the project. It was not until 1 month in, that we fixed this issue by restricting total access on our linux host by geolocation.



# Hardware Codes

As the server was already done the next step was to connect node MCU to the server. The protocol that used for this connection is MQTT. MQTT is a publish/subscribe machine to machine connectivity protocol. This protocol allows multiple devices to communicate at the same time instead of one-to-one message-sending processes. We chose this protocol, as “it is lightweight and energy-saving, which is very suitable for the use scenarios of the Internet of things” (X, E. 2020, March 12). Having done the research about this protocol our group wrote the code for node MCU. The following piece of code connects the MCU to wi-fi:

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.println("Connecting to WiFi..");

}

Serial.println("Connected to the WiFi network")

The code bellow connects the MCU to the server, finds the avatar with the number of 500 and names it the name of the sensor that is attached to the MCU, in this case vibration sensor. If the connection Is not made the appropriate notification is written on serial monitor and the code iterates in the loop again unless the connection is made.

client.setServer(mqttServer, mqttPort);

client.setCallback(callback);

while (!client.connected()) {

Serial.println("Connecting to MQTT...");

if (client.connect("500/vibration sensor")) {

Serial.println("connected");

} else {

Serial.print("failed with state ");

Serial.print(client.state());

delay(2000);

}

After the MCU is successfully connected to the server it is necessary to exchange data between the sensors and the application. The five of our sensors: flame, gas, vibration, water leakage and temperature/humidity sensors only send the data to the application so we used the client.publish("0", "1/0"); method which sends the data of the sensor to the 0th output pin of the avatar on the server. On the other hand, the application avatar is subscribed to this 0th pin and reads the data that are sent from the sensor. The two other functionalities of our projects were the control of light and door from the application. For this we needed the code that can read the information from the application and send it to the MCU. For this purpose we used the other method: client.subscribe("0"); and the callback function (below) which checks the 0th inout pin of the avatar and based on the information either turns on the light/door or turns off.

void callback(char\* topic, byte\* payload, unsigned int length) {

Serial.print("Message arrived in topic: ");

Serial.println(topic);

Serial.print("Message:");

for (int i = 0; i < length; i++) {

Serial.print((char)payload[i]);

}

if (!strncmp((char \*)payload, "1", length)){

digitalWrite(5, HIGH);

}

else{

digitalWrite(5, LOW);

}

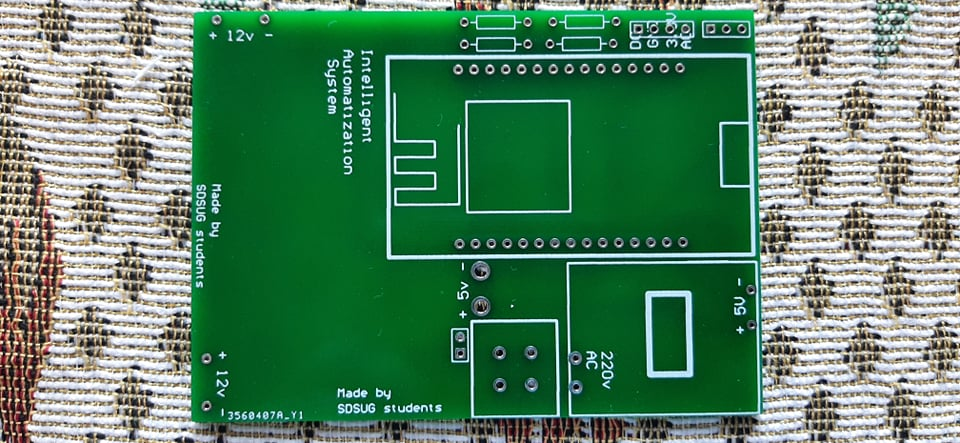
Serial.println();

Serial.println("-----------------------");

}

Flowless functionality of this code was crucial and after completing this task we were able to connect sensors to the ESP module and send the data from it to the server and vice versa.

# Printed Circuit Boards

Printed Circuit Boards (PCBs) are one of the most important parts of our project as it is used to assemble all the necessary components. Its correct working affects every part of the project. We started designing PCBs using Altium and had a little session with our advisor to get to know how to make sketches with this program. At first, we decided to make 3 different boards for the sensors and relays which are working on three different voltages: 3.3V, 5V, and 12V. We created the sketches for all the PCB parts and as everyone who is starting to work in a new program, we had some problems too. The main problem we faced was the fact that the delivery of hardware components was delayed, and we had no exact measurements. Also, when all the components arrived, we faced a new problem we needed different kinds of relays and stabilizers. However, we had no time to buy it on the international market and we started to look for these components here in Georgia. But as always something must go wrong and there was a shortage of the components we needed, such as the 3.3 V relay, on the local market. Apparently, we had to make our 3.3V relay by changing resistors in 5V relays. Also, we needed to get 5V DC power from 220 AC power, as I already mentioned there was a shortage of these components and we only managed to find 3 220 to 5 V converters. Even though the whole universe was trying to stop us from getting our hand on the components we needed, we used a power supply to get 12V from 220V AC power and then used L7805 stabilizer to convert 12V to 5V DC. After that we had to stabilize output voltage as the output from the sensors could be more than 3.3 Volts. Our team stabilized the system by adding two resistors with the same resistances, which created voltage division. After we located all the components on the board there were some spaces that were lost, and we had to reconstruct the whole PCB to use the entire area of the board more efficiently. Furthermore, we understood that creating three different designs for the boards was complicated and impractical. So, we decided to design one universal board which had to be usable for all the components with different voltages. Below you can see the sketch and the actual printed circuit board that we are using to assemble all the hardware components.

# No description available.Assemble Hardware Components

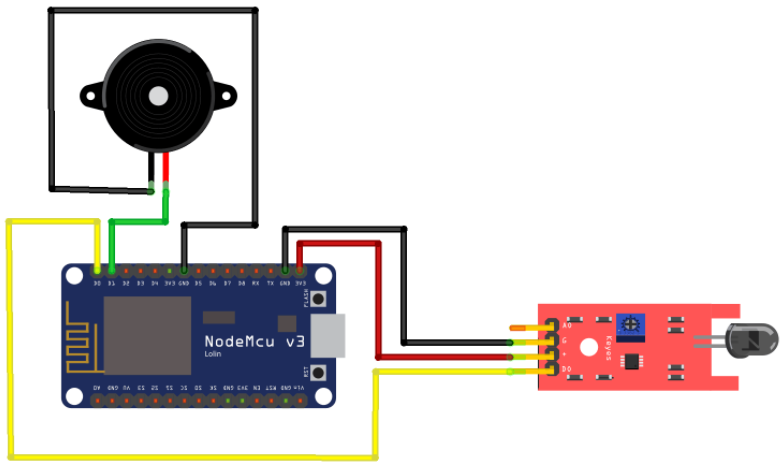
After collecting all the necessary hardware components, we started working on creating intelligent automatization system.

## Fire Detection

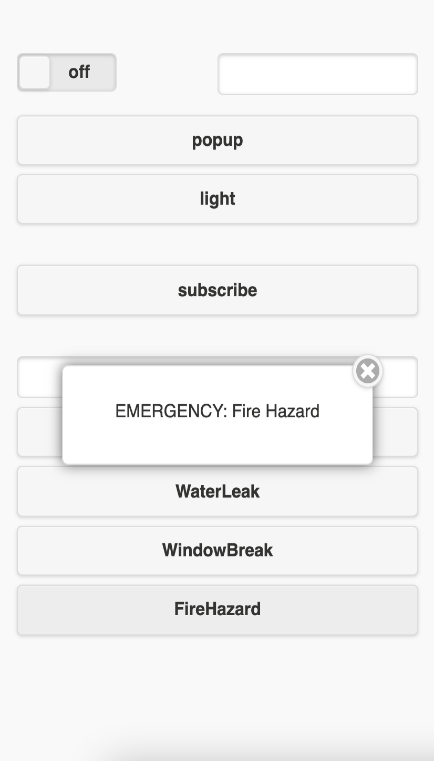
To detect fire in the house we used LM393 4 pin IR flame detection sensor. The features and specifications of this sensor are following:

* the test distance for the lighter flame should be 80cm, the larger the flame, the larger the test distance.
* Small plates analog output mode and the AD conversion process can help get higher accuracy.
* Can detect the flame or the light with the wavelength at the range of 760 nm to 1100 nm.
* The detection angle is about 60 degrees, extremely sensitive to the flame spectrum.
* The comparator output with clean signal, good waveform, strong driving ability of more than 15mA.
* With an adjustable precision potentiometer to adjust the sensitivity.
* Operating voltage: 3.3V to 5V.
* The output form: DO digital switching outputs (0 and 1) and AO analog voltage output.
* Using LM393 comparator with wide voltage.

This sensor had to be connected to the main controlling unit and it had to send notification to the phone that there is a fire in the house. To do so, we created device 400 on the server for flame sensor. After running the code without loop for sensor everything was fine. However, after we tried to save device (after we added subscribe) on server it logged us out and we could not log in again. As it appeared later it was because of the incorrect connection status. I connected sensor and buzzer to ESP as shown on the picture (*IoT based Fire Alarm System Project using NodeMCU ESP8266.* (n.d.).) below.



After that we wrote the loop code to get data from the sensor and as you can see below our fire detection system works properly.



## Gas Leak

To detect gas leak in the house we used MQ-7 carbon monoxide detection sensor. In case of leak system activates siren, sends the notification to the application, and locks the gas pipes. Below you can see some features that were given to us by the seller:

* The switching signal has a DO (TTL) output and analog output AO.
* TTL output valid signal is low.
* Analog output voltage with the higher concentration of higher voltage
* Input voltage: DC5V Power consumption (current): 150mA
* DO output: TTL digital 0 and 1 (0.1 and 5V)
* AO output: 0.1-0 .3 V (relative to pollution), the maximum concentration of a voltage of about 4V
* After the sensor is powered, it needs to warm up around 20S to measure data

As you can see below our gas leak detection system worked properly, so we could move on the next step.

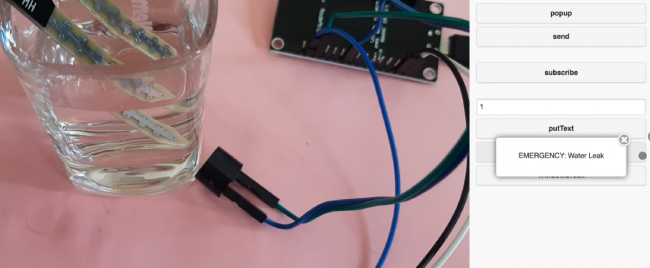
 

## Water Leak

With the help LM393 Soil Moisture Hygrometer Detection Humidity Sensor Module we can detect if there is leak of the water in the house. There are some main features that made us choose this sensor:

* It has adjustable sensitivity (shown in blue digital potentiometer adjustment)
* Operating Voltage is 3.3V-5V
* Module Dual Output mode, a simple digital output, analog output more accurate.
* Small PCB board size: 3cm \* 1.6cm
* Power indicator (red) and the digital switch output indicator (green)
* Using LM393 comparator chip, stable

In case of leak system activates siren, sends the notification to the application, and locks the water pipes.

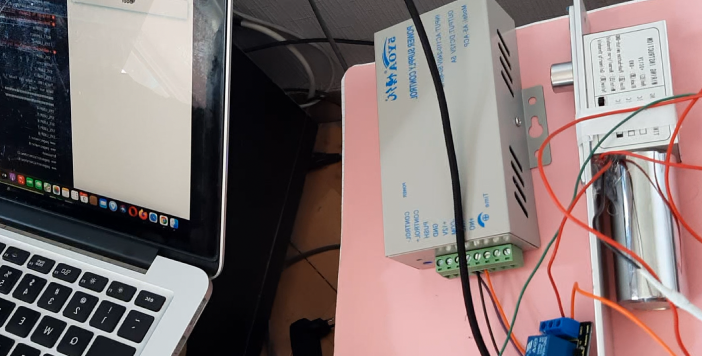


## Door Lock

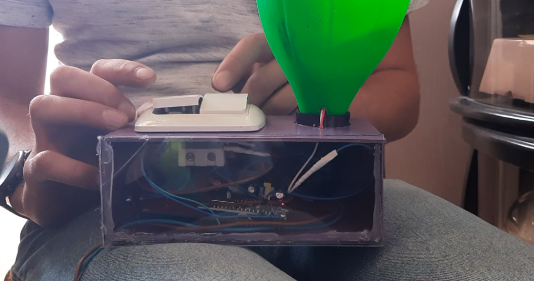
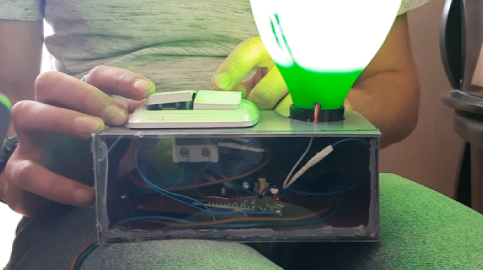
One can lock and open the doors with the key as well as with the application from both mobile and computer. We used electromagnetic door lock Dead Bolt which has following specifications:

* 12V input voltage
* Output Voltage 5V
* 0.75A input current

To create door lock system we used the 12V adapter with microcontroller build in which would be suitable for our board, also we modified the board and added 5V relay which was rebuilt and its actual working voltage changed from 5V to 3,3V, so the electro lock system works that way: 12V adapter powers board and gives the power to the lock ESP takes signal from user to unlock the lock and sends 3.3V to the relay, which is connected to 12V adapter and it opens the door.

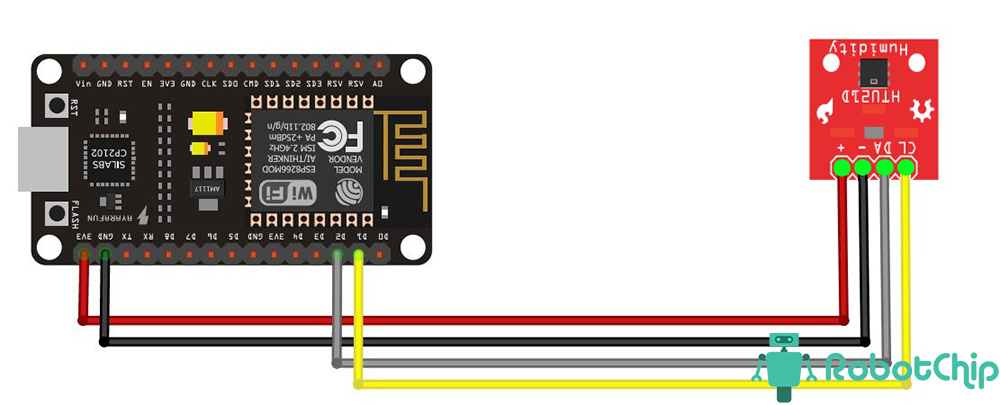


## Light Control

One can control lights from the application as well as from the light switch. We are using 220 to 12 V converter to power the system of the switch. Also, we are using 3V relay to remotely control the lights. As a result, lights can be turned on from the application and turned off from the switch and vice versa.

Temperature Monitoring

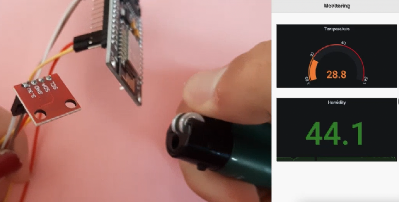
With the application one can monitor temperature and humidity inside and outside of the house. To make it possible we used Temperature and Humidity Sensor GY-213V-HTU21D HTU21D I2C. On the picture (Сергей. (2020, October 2) below you can see little schematic of how we connected this sensor to node MCU.



Below you can see some main features and specifications of this sensor.

* Power supply: 1.5V-3.6V
* Humidity measure range: 0-100%RH
* Temperature measure range: -40degree to 105degree
* Max. power consumption: 2.7uW
* Communication: I2C
* Humidity precision (10%RH to 95%RH): HTU21DÂ±2%RH
* Humidity delay: +/-1%RH
* Measure time: 50ms
* Year drift: -0.5%RH/year
* Response time: 5s
* Size: 1.5x1.5cm (approximately)

As you can see below our temperature monitoring system works properly and gives us data on application from Grafana.

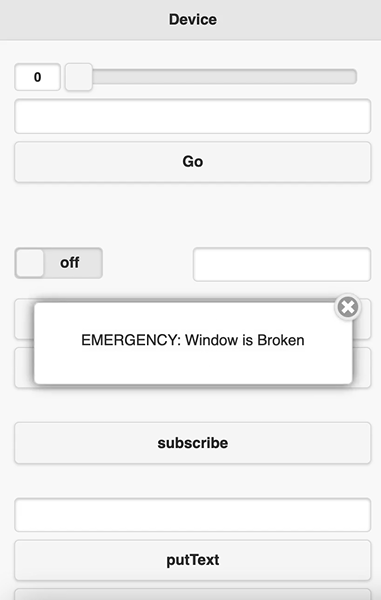


## Glass Break

As theft and earthquake alarm, we used SW-420 vibration sensor, which consists resistors, capacitor, potentiometer, comparator LM393 IC, Power, and status LED in an integrated circuit. The features and specifications of this sensor are following:

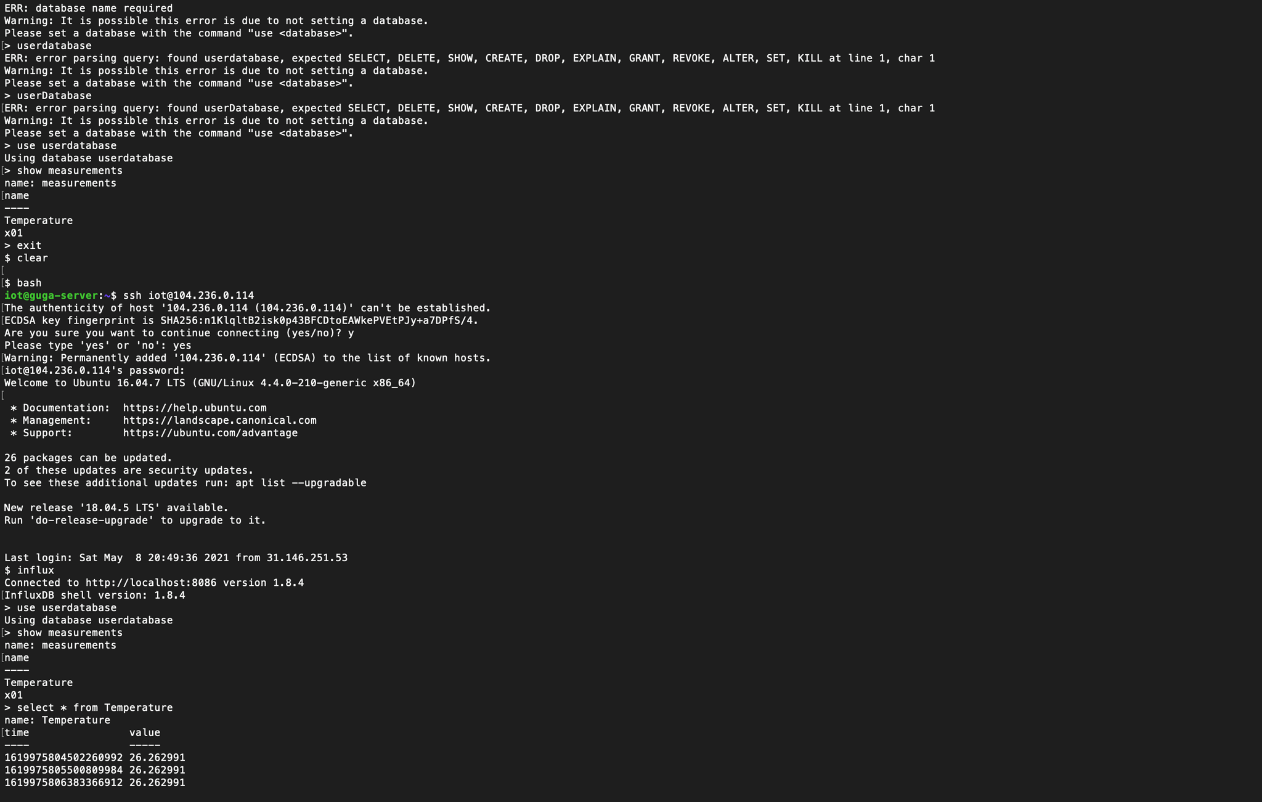
* Operating Voltage: 3.3V to 5V DC
* Operating Current: 15mA
* Using SW-420 normally closed type vibration sensor
* LEDs indicating output and power
* LM393 based design

Glass break sensor should alert the user if the window glass is broken and possibly someone entered the house. So, we created device 500 on the server for the vibration sensor and did everything as we did for flame sensor. As you can see below our system can detect if the glasses in the house were broken and send the notification to the application that intruder is trying to come into the house.

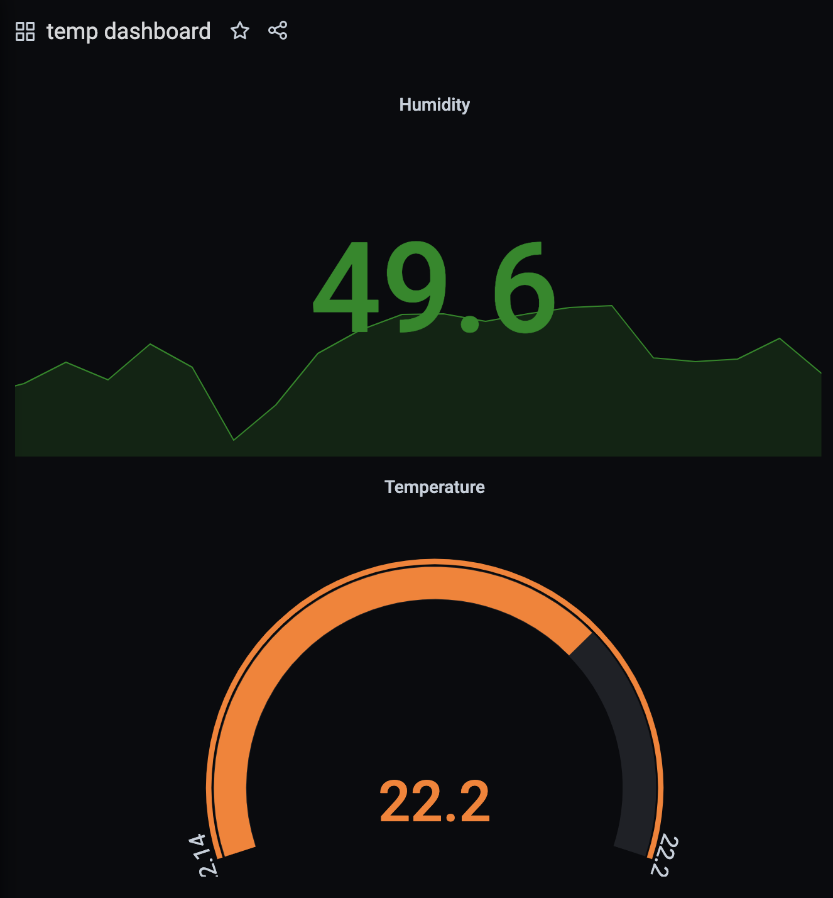
# Influxdb

Our next step in progress was getting a good database up and running. Main requirement that we had was the database to be a time series. The good old SQL or MySQL simply was not an option, since we are working with big data, and those databases would simply not handle all these data inputs constantly going in from our sensors. Even if it could support the data traffic, we would face the problem with visualizing data in real time. It would take more than 5-10 seconds to get the single data and visualize it, and it would be a big problem for us. Fortunately, we found the database we were looking for and would satisfy our needs, and it is InfluxDb. This is a great time series database that was optimized for IOT work. Installation process was super easy, we just followed the instructions given by ourselves, and we then just created a database and assigned it to one user. After that, it was just an issue of creating measurements for each and every device on the server. Code snippets for that will be given.

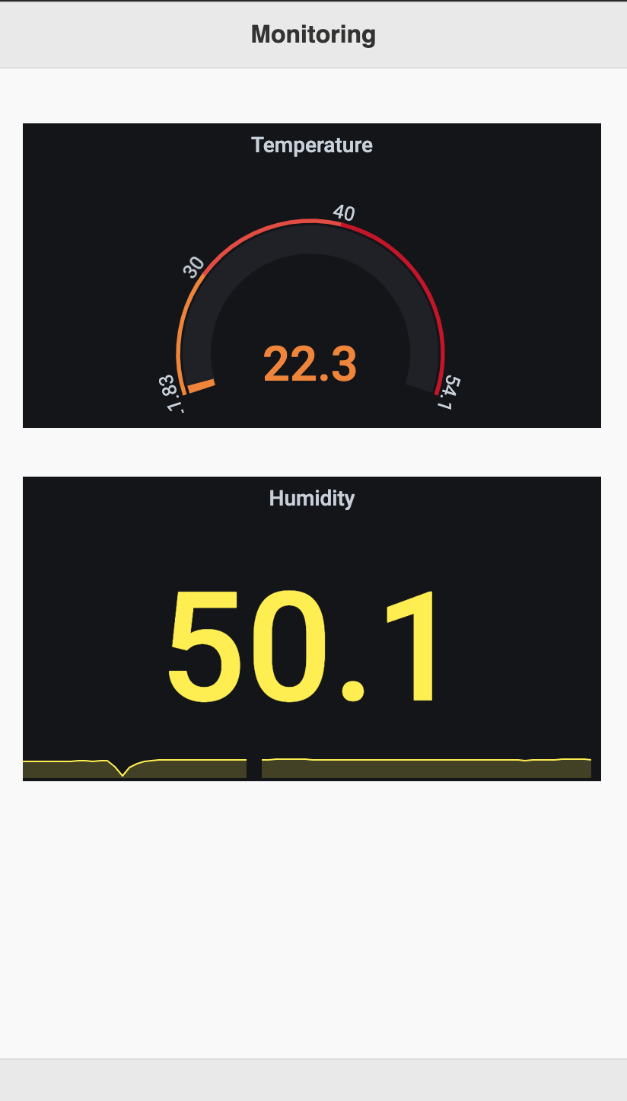
# Grafana

After getting the database working, it was finally time to visualize our data. We did not search visualization tools for too long, since Grafana was our best choice. It was already integrated with Influxdb and they could work together no problem. We simply followed instructions and installed Grafana on our Linux machine. We went with default port 300 for it. Grafana has Graphical User Interface integrated as well, so it was much easier working with it. We would simply go to our server IP and port 3000 and Grafana was already there, with settings, management tools and dashboards ready to go. The most challenging part involving Grafana, was visualizing it into our application. By default, Grafana would not allow access from outside the user, so whenever we tried to integrate it in our web application, it would not show. After 2 days of searching online and trying to fix our problem we finally arrived at the conclusion that the default configuration should be changed, after that it was very simple.



# Application

When we mentioned that we wanted to develop a mobile application, we had in mind a simple connection to the server and simple mobile app, but after working on our project for some time, we decided to challenge ourselves and develop a Web Application that would be accessible not only on mobile, but from the desktop as well. We decided to go with the scripting language that was provided to us by our mentor David Chxaidze. It is basically html and JavaScript fused together, and it is extremely easy to use due to its readability. We developed an application in this scripting language, and then hosted it on the GitHub repository. The application consists of several pages. The first page is the main page, which greets users whenever they enter the application. From there they connect to the server and can start to manage their intelligent house. They can control lights, doors, temperatures and much more, including some hazardous threats. Below you can see how we monitor temperature from the application.



# Website

In this project we decided to do our best during the whole working process in order to satisfy our professional need and succeed. We decided to create our customized website which is one of the important parts for our project’s future development. Interested customers can visit our website and find project aim, working process and what is the most important why our project is special. In future we want to update our website with future goals and new activates.

* Programing languages used in the development: HTML5, CSS, JS.
* Frameworks used: Bootstrap
* Libraries: jQuery

HTML was used for markup alongside with the CSS in order to create design, colors, fonts animations and transitions.

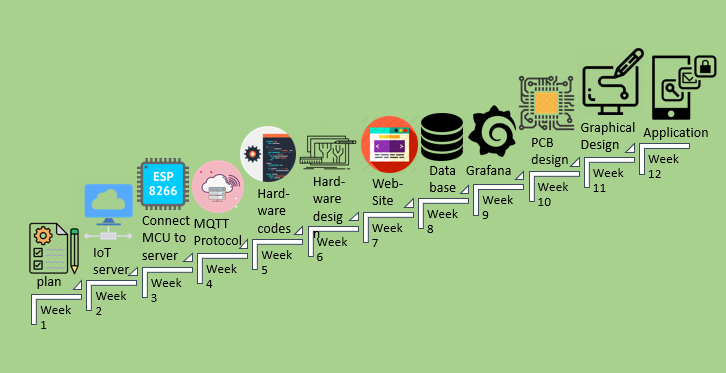
I used different HTML and CSS files for pages in order to easily update and change content or design of the website. There is Index.html file for the main page working with style.css file. I separated JS, font, and animation files and some of the images.

In our website we have five different pages “about project”, “timeline” “meet the team” and “contact us” including the home page. On the home page we have summary of all pages. Starting with the slideshow with the project related images, below is displayed the important aspects of the project description and aim. In the bottom we have timeline, meet the team and motivational poster with the footer itself.

The website is all responsive that means that it fits and looks good in all devices. It will automatically adjust for different screen sizes and viewports.

# Timeline

We started our project by planning every step of it step by step and we worked according to this plan all along, which made our working process easier and more efficient. You can see visual representation of our timeline on the graph below.



# Conclusion and Recommendations

In twenty first century automatization systems are not only making our lives easier, but also, they are making it more enjoyable. Nowadays smart home features are rising trend all over the world, however this technology is very scarce in Georgia. After working together with our group advisor for a year our team managed to create integrative and easy to use, smart house system based on the Internet of Things (IoT). The main features of our project were to control light and door lock systems, detect life threatening factors and monitor the temperature and humidity inside and outside of the house. After making as prototype of our system our team can gladly say that we successfully managed to create a technology which will enable majority of people to get their hand on intelligent automatization systems.

Generally, this capstone project was a great way to reflect the knowledge and skills we gained during last four years. The tasks and problems that we faced throughout the project really tested us not only our knowledge from the past years but ability to be STEM engineer and create real-life product.

As everyone in these days we faced the complications such as delays on deliveries, product shortages etc. However, we never gave up and always tried to make the best out of the bad situations. So, we recommend to never give up on the project because of your current working conditions or global pandemic and always remember that there is no problem that has no solution.

# Technical Specifications

## Hardware

* LM393 4 pin IR flame detection
* MQ-7 carbon monoxide detection
* LM393 Soil Moisture Hygrometer Detection Humidity
* Temperature and Humidity Sensor GY-213V-HTU21D HTU21D I2C
* SW-420 vibration
* dead bolt door lock
* 5V relay
* 12V power supply
* L7805 stabilizer
* 220V AC to 12V DC power adapter
* 220V AC to 5V DC power adapter
* analog to digital converter

## Software

* DigitalOcean - hosting services
* MaxIoT - iot server
* VsCode - development environment for WebApp
* influxDB - DataBase
* Grafana - Visualisation

## Website

* Programing languages used in the development:
  + HTML5
  + CSS
  + JS.
* Frameworks used - Bootstrap
* Libraries - jQuery

# Team-member Roles

First and most important step of our project was to plan everything step by step and divide roles of every team member appropriately. Starting from the beginning with our group leader and mentor we decided to define our roles and responsibilities. More or less, we knew our strengths and weaknesses which helped us to have right duties in order to support the efficiency and effectiveness of the project process.

## Guga Esitashvili

Guga was a leader of our team. His responsibilities were not limited strictly to some one thing, but quite the opposite. As a group leader, he had to be in progress of every single task while developing, therefore he was involved in almost everything, but his contributions to the project are as follows. Firstly, Guga had to develop a mobile application for our users. He did it by himself, because as we mentioned above he knew his abilities, and also because it would get very messy if two people developed the same application together. On the other hand, his other contribution was setting up an IoT server and he shared this job with our teammate Anna Dzamelashvili. They got up and running whole server on a Linux machine on a remote host. Guga’s next responsibility, as a team leader, was managing our schedule, meeting our promised deadlines, and developing our project in such a way that no conflicts would occur between hardware and software. He was also team poster editor for our final project and presentation. With all of the above-mentioned contributions, Guga was also a technical illustrator, and of course, contributed his part to the final report.

## Tea Gamrekelashvili

Tea started working on the project by analyzing our project’s financial needs and planned the efficient use of the budget we were given. And as you can see in our financial analysis above, we used it as effectively as possible. Before we collected all the hardware components, Nino Nonikashvili and Tea Gamrekelashvili had enough time to do a research on MQTT protocol, which is used to connect esp modules to wi-fi and the server and exchange the data between the application and sensors. After this she started hardware coding for her part of the project, which was gas and water leak detection. As we already had all the components and as Uta Buziashvili finished working on the design of printed circuit boards Tea could start working on assembling the parts needed for these detection systems. With all of these technical contributions, she did presentations and the poster for the team. Moreover, her last part as a member of this team for the project was to assemble report together, write her part and make a final version of it.

## Nino Nonikashvili

Nino’s contribution to the project consists of two parts: hardware and software. Regarding hardware, she took part in creating circuits for flame detection, vibration, temperature sensors and light control. We have separate PCBs for every sensor, so Nino created the PCBs for the sensors that she was responsible for. Those PCBs consist of the ESP8266 module, the sensor, the relay, voltage adaptor and resistors. Regarding software she wrote the codes for the ESPs to connect to wi-fi and server and exchange the data between the application and sensors. the code is based on the MQTT protocol. With our team member Tea Gamrekelashvili Nino did the research about the MQTT, learnt it and wrote the basic part of all codes for ESP which connects the node MCU to wi-fi and Server.

Besides the informational and technical contributions, Nonikashvili Nino was a poster editor and helped Tea Gamrekelashvili to eliminate any flaws. With all of this she wrote her parts in this report.

## Ana Dzamelashvili

In this project Ana worked on IoT server, software design, web development, and report writing/editing. She helped Guga, who also worked on IoT server to set up together. The server was developed by our mentor who directed us in this process. Ana worked on setting up the server, especially on devices which are avatars of real sensors in order to deliver information to mediator and generate the result. Dzamelashvili Ana contributed in the software design process by which sheI created a specification of a software artifact intended to accomplish goals, using a set of primitive components.

Since the report is the main illustration of our project and the work done during it, we decided to write some parts of the reports which were related to our tasks and duties. Ana wrote her part of the report and also reviewed the software process sections. Mostly report was written and reviewed and completed by Tea Gamrekelashvili. I was responsible to create the website for our project. The website is all responsive that means that it fits and looks good in all devices.

## Uta Buziashvili

Uta’s first and most important role as a team member was to decide which components should have been used for the hardware part of the project. Before we gathered all components which we wanted to use, he started working on PCB design. When the all necessary components were connected, it appeared that the scratch which he made was not usable, because the real components which we had were not same as expected, that’s why he started to make changes on board and made another board which was perfectly suitable for our project, but as working on PCB was interesting he suggested to make the new universal PCB board and made the new design which would be suitable for any kind of our system. Also, time by time Uta was making report of his work in our group, because it is really important to make all the notes which are necessary for others to understand how system works. Before the delivery of our PCBs, he started to work on hardware coding for door lock system. After that Uta was able to assemble all the necessary hardware components used in door lock system. Also, time by time we needed to make presentation and Uta’s role besides the technical contribution was to edit team presentations.

# References

Santiago Uceda. (n.d.). https://santiagouceda.com/.

User, B. S. (n.d.). *The 7 Greatest Advantages of Smart Home Automation*. Demo. <https://bluespeedav.com/blog/item/7-greatest-advantages-of-smart-home-automation>.

X, E. (2020, March 12). *Upload sensor data to MQTT cloud service via NodeMCU (ESP8266)*. Medium. https://emqx.medium.com/upload-sensor-data-to-mqtt-cloud-service-via-nodemcu-esp8266-b31980b56308.

*IoT based Fire Alarm System using NodeMCU ESP8266*. IoT based Fire Alarm System Project using NodeMCU ESP8266. (n.d.). https://iotdesignpro.com/projects/iot-based-fire-alarm-project-using-esp8266.

Сергей. (2020, October 2). *Метеостанция на HTU21D и ESP8266*. RobotChip. https://robotchip.ru/meteostantsiya-na-htu21d-i-esp8266-1/.

# Appendix

Bills, codes and all the material will be uploaded separately.