



herotech team, Romania  
[herotech.ro](http://herotech.ro)

**Title:** Smart House Lighting

**Student Names:** Daniel Marpozan, Codrin Muntean

**School:** National College “Octavian Goga”

**City and Country:** Sibiu, Romania

**Category:** Science – Innovation

## **The project description paper**

### Abstract

Our application is a WI-FI integrated sensor network which allow users to control the luminaires either automated or manually. This system is meant to help people save energy and make a better life. For the implementation of the project, the team appeal on electronics, mechanics and IT knowledge.

### Introduction

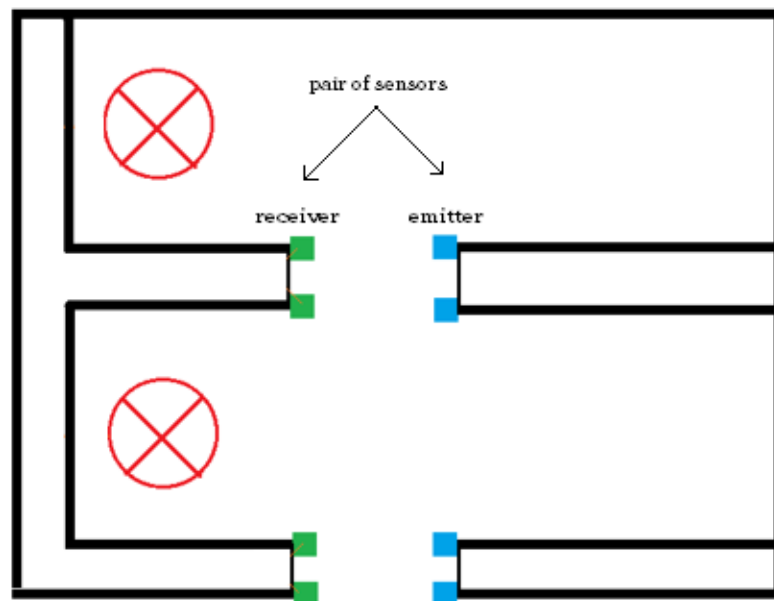
Our world is continuously changing and the natural resources are limited. Unfortunately, in some countries the resources used for creating electricity are non-renewable, so we have to act now to change the future of our planet. By making this project, we hope we could contribute at this change.

## The goal of the project

The main aim of our application is to reduce energy consumption in a house by automatically switching on and off the lights. Another goal is to make our houses more comfortable. By doing this, we will be able to decrease pollution and usage of non-renewable resources. Needless to say that all of these could contribute to a better life on Terra.

## Description of the project

The system is composed of two pairs of IR sensors and a relay controller module. An IR emitter and an IR receiver represent a pair of sensors. These are located at the door frame, as in the picture.



The emitter sends IR signal to the receiver module and this detect when the ray of IR light is interrupted, and therefore someone gets in or out of the room. In this way, our system act like a people counter. Based on the number of people in a room, the lights will be turned on or off in an automated way. Also, we make a cross – platform application that allow us to control every single light in case of the user do not want to control them in the automated mode.

## The team

Our team, herotech, is composed of three high school students (16 years). After the school had begun we decided to participate to GENIUS Olympiad with two projects, one in the Robotics category and this one in the Science-Innovation category.

Each member was in charge of a specific part of the project as follows, but only the first two will represent the team:

- Daniel Marpozan – responsible for electronic part.
- Codrin Muntean – responsible for the programming part.
- Alexandru Radac – responsible for the design of the project and the idea behind it.

## The idea behind our system

The need for a smart energy-saver system led us to the idea to make these sensors. Firstly, we made a scale model of a house and sensors were implemented in it. The result achieved did not satisfy us, because there were a lot of wires which connected all sensors to the controller part, an Arduino board. To resolve this problem, we integrated WI-FI modules (ESP8266) in every sensor board, which are connected to a central server.



*This was our first try: sensors integrated in a LEGO model*

## Methods and experiment

### 1.The hardware part

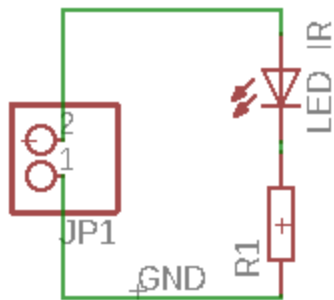
Firstly, we designed the electronic part, using the Autodesk software (EAGLE).

- **The emitter**

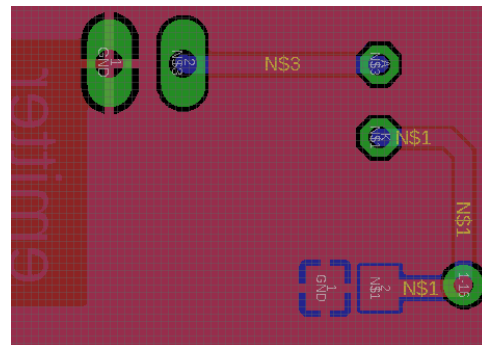
It is a simple electronic circuit with an IR led and a resistor calculated with the formula

$$R = \frac{V_{in} - V_{fw}}{I_{led}}$$

This part is connected at a 9V battery and emit IR light continuously.



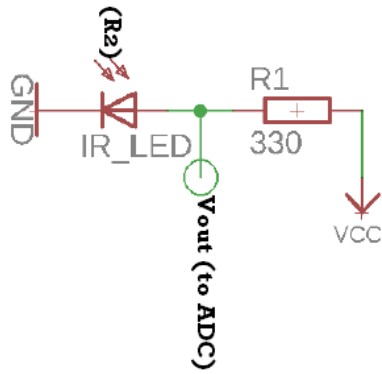
*schematic*



*layout*

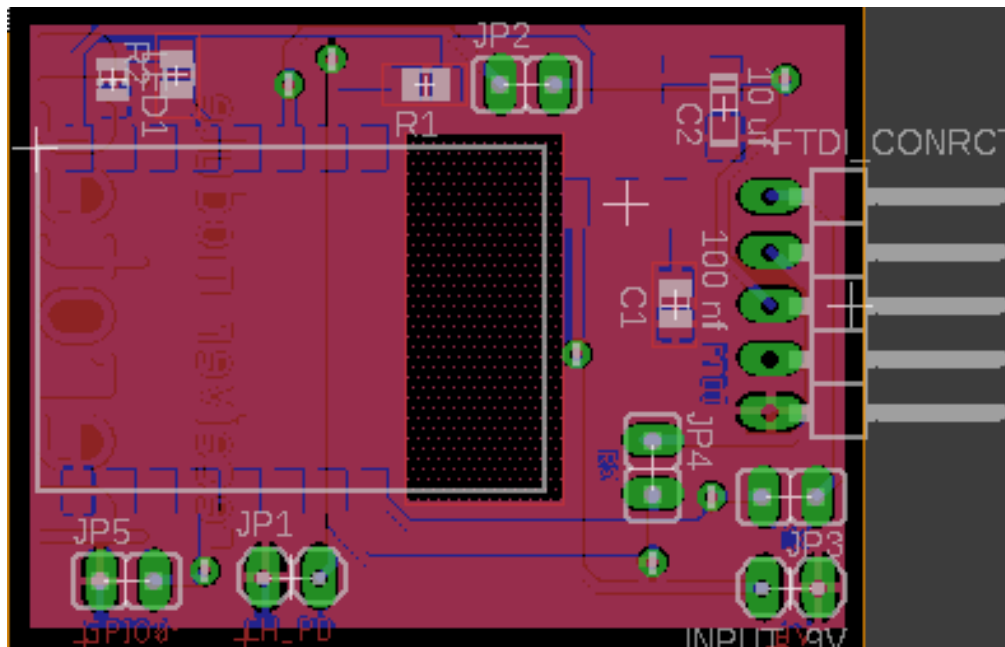
- **The receiver**

It is designed around the ESP8266 chip, that send information based on data read on ADC pin. The IR led receptor acts like a resistor whose value is changing depending on the received light. Based on it, we made a voltage divider with this led and a resistor.



the voltage divider with the formula

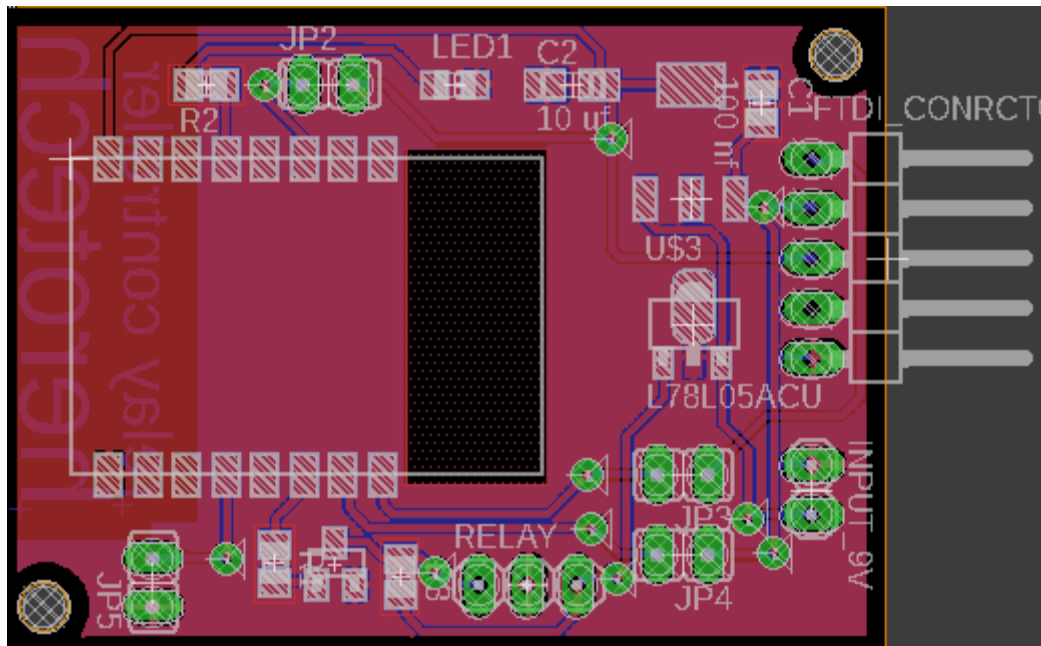
$$V_{out} = \frac{R_2}{R_1 + R_2} \cdot V_{in}$$



Layout in EAGLE Autodesk

- **The relay controller**

This last module switches on or off the lights, based on some information received from server.



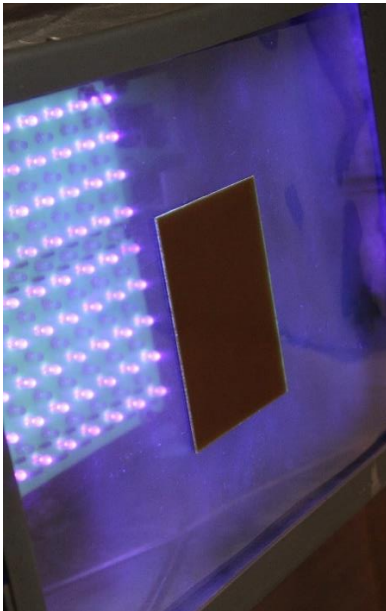
*Layout for the relay module*

The power supply for the last two modules is a 9V battery, so we used a linear 3V3 voltage regulator. Both of these boards are programmed via a FTDI programmer, using some specific pins (3V3, GND, RX, TX). The programming language used for these boards is C++ (we used Arduino IDE).

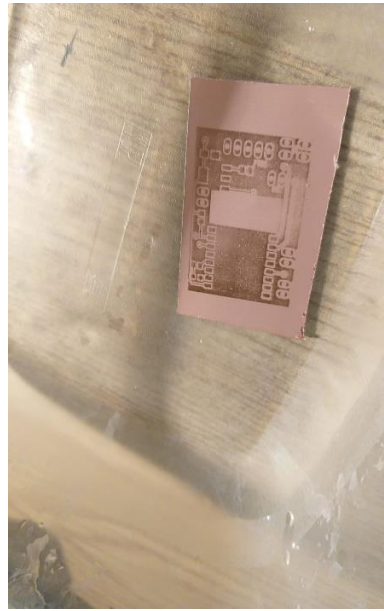
After we had designed these, we made a simulation, using Proteus software, to be sure that all will work properly. By doing this, we realized that we had to change some resistor values and other electronics parts in the schematic.

The next step was the actual PCB realization, using UV exposure. Firstly, we expose the PCB with photoresist layer for 60 seconds, after we developed it with NaOH. Then, we inserted the board in a  $\text{Na}_2\text{S}_2\text{O}_8$  solution.

After making the drills for vias and THT components (through-hole technology), we solder all THT and SMD components. Finally, we uploaded the code on ESP chips and we tested its functionality.



*UV exposure*



*the PCB in NaOH solution*

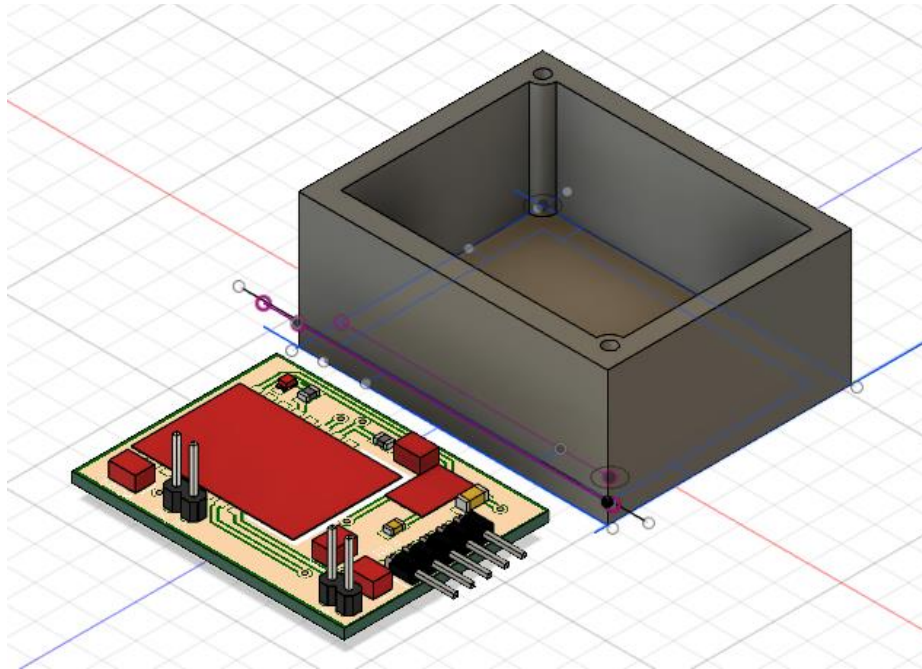


*Daniel soldering the components*

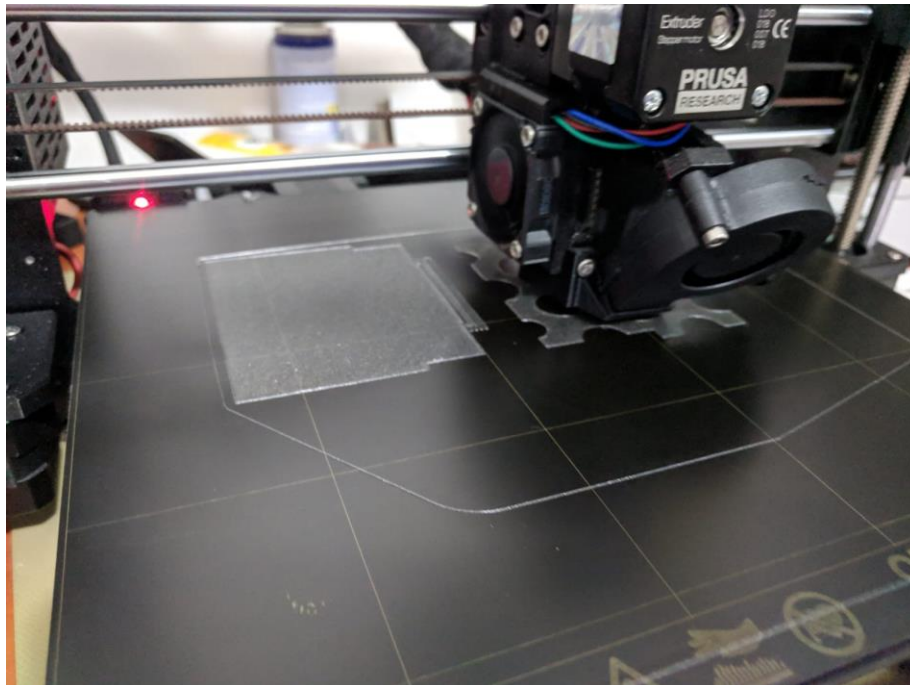
## 2. The design part

Using Autodesk Fusion360 software, we designed some cases for our sensors. We export our PCB-s layout from EAGLE to Fusion360 and we integrated them in simple boxes which we printed, using 3D technology.



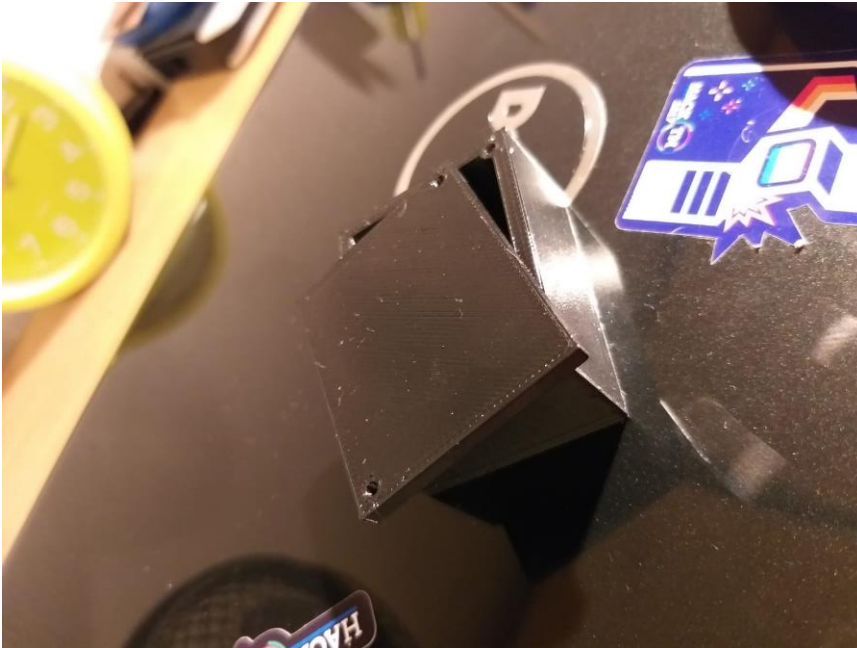


*The pcb and a case for it (Autodesk Fusion360)*



*The 3D printing process using the Prusa i3 mk3 printer*





*A printed case*

### 3.The programming part

- The ESP code:

The process is the following:

The ESP module connects to a local network with the hardcoded SSID and password. When the connection is established it requests a unique ID from the master server that it saves in a global variable. Using the unique id, the program keeps requesting from the server at a constant interval of time any command that needs to be executed (ex. switching on/off the light). When the boards with sensors detect someone entering/exiting a room they also send a POST request to the server letting it know the time that the sensor stops receiving signal.

- Master server code:



herotech team, Romania  
[herotech.ro](http://herotech.ro)

The master server is an ASP.NET CORE application which runs on a machine connected to the local network. The machine is meant to be assigned a local domain or to have a static IP, so the ESP can hardcode that value.

The application is connected to a local DB using the Entity Framework CORE where we store ESP related info: name, IP, command to execute, type (sensor/light), room related info: name, list of lights.

The whole processing happens here: The time values received from the ESP boards are processed, and so the server figures out from what direction the person passes through the door. After that the people counter from the two rooms change (+1/-1).

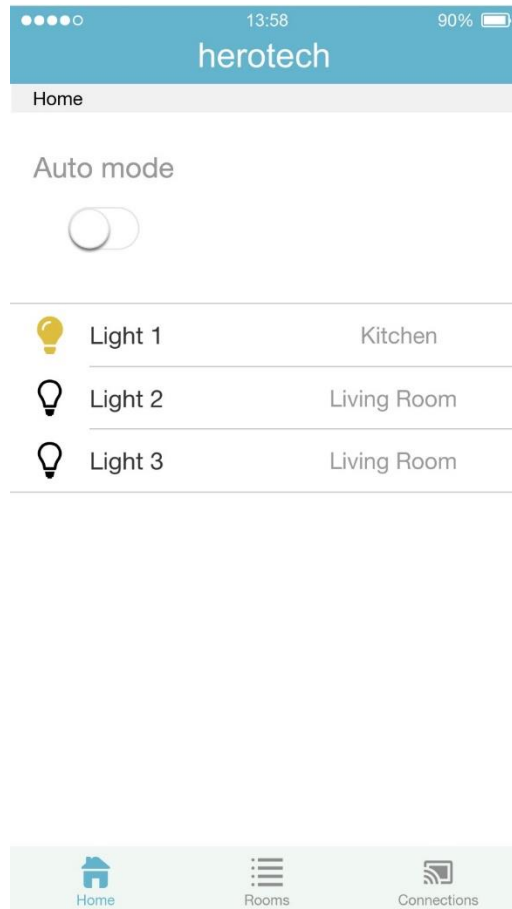
- Application code:

We made a cross-platform application using the Xamarin.Forms Framework.

We have 3 pages:

## 1. The home page

Here we have a switch for setting the mode of our system: automated or manual. For the automated mode we have a view with an interval time picker. We decided to have a specific user selected functioning interval because we don't want the lights to be active during day time when you pass through a door and the automated mode is enabled. When the manual mode is selected, the application displays a list of every ESP that is connected to a light, and it lets the user to manually select which lights should be on.



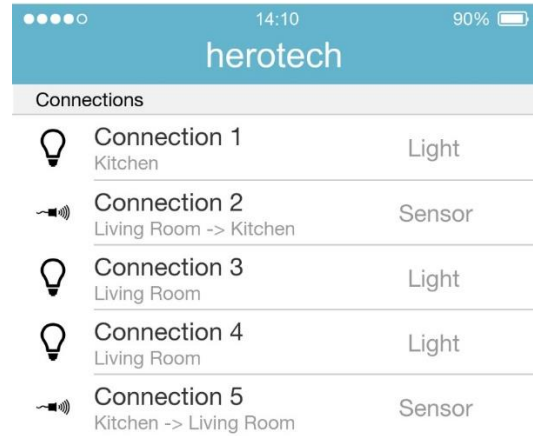
*The home page*






## 2. The rooms page

The rooms page displays a list of the rooms from the database with the assigned lights for each room. We can also create a new room and save it to the database.



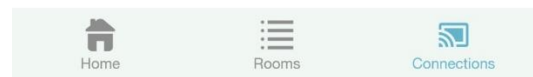
Rooms		
	Living Room	Lights: 2
	Kitchen	Lights: 1
	Hall 1	Lights: 0



Connections		
	Connection 1 Kitchen	Light
	Connection 2 Living Room -> Kitchen	Sensor
	Connection 3 Living Room	Light
	Connection 4 Living Room	Light
	Connection 5 Kitchen -> Living Room	Sensor



*The rooms page*



*The connections page*

### 3. The connections page

This page consists of a list of all the connections saved in the database by the server. By tapping an item from the list we can configure the assigned room if the ESP is connected to a light and the room transition if the ESP is connected to a sensor.

```
[HttpPost]
public async Task<IActionResult> AddEspCommand(int espId, string command)
{
    var esp = _context.Esp.FirstOrDefault(i => i.Id == espId);

    if (esp == null)
    {
        return View("Error");
    }

    esp.Command = command;

    _context.Esp.Update(esp);
    await _context.SaveChangesAsync();

    return View("ActionComplete");
}
```

*Master server: The AddEspCommand action from the ESP controller*

```
void setup()
{
    pinMode(LED_BUILTIN, OUTPUT);

    Serial.begin(115200);
    WiFi.begin(local_SSID, local_password);

    while (WiFi.status() != WL_CONNECTED)
    {
        delay(500);
        Serial.println("Waiting for connection");
    }

    HTTPClient http;

    http.begin("http://herotech.local/EspServerTest");
    http.addHeader("Content-Type", "application/x-www-form-urlencoded");

    int httpCode = http.POST("ip=" + WiFi.localIP().toString());
    uniqueID = http.getString();

    if (httpCode != 200)
    {
        Serial.println("Server-side error");
        pinMode(LED_BUILTIN, HIGH);
        return;
    }
    Serial.println(WiFi.localIP().toString());
    Serial.println(uniqueID);
}
```

*The setup function on ESP board*

## Results:



*Our sensors on the door frame*

## Conclusion

In closing, we hope our project is a useful one, having a positive effect on our lives. By making this application, we want to digitalize our house, make them more comfortable. Besides, a very important aim is the impact on the nature, by diminishing energy consumption. In the future, our team will improve this system, by making it easier to use, with a more attractive interface.

## References

- <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-2.2>
- <https://university.xamarin.com/>
- [https://www.open-electronics.org/guest\\_projects/\\_trashed-3/](https://www.open-electronics.org/guest_projects/_trashed-3/)