# Team Gelzen and Sam

## Design Philosophy

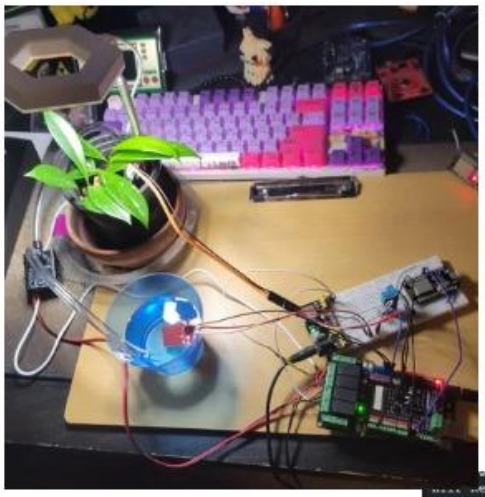
Since our project was going to revolve around working with wireless communication, we first had to decide on a protocol for this requirement. We decided on going with the WiFi protocol as it is easier than the other protocols and can have a lot of functionality if integrated with internet and is able to easily extend with more devices. Additionally, Wifi is also versatile and has a lot of protocol for communication within it and we ultimately decided to use the HTTP protocol as that is a very simple yet effective protocol. Outside of communication, we just tried to follow the project requirements and specification to the best of our ability.

As for our algorithms, we didn't really use algorithm from other sources other than the basic workings of our HTTP protocols, and readings from the DHT sensor. And although we didn't really have a complex algorithm that we implemented, I was proud of doing the communication algorithm. It involved a lot of debugging throughout the whole production since it involves a lot of C pointers and conversion between a collection of bytes into something like floats, and integer. This project really helped me improve my understanding of C language as I used a lot of C features I don't have a lot of experience on like preprocessor macros, use of structs and pointers manipulation.

## Team assignments and report

We organized the team task based on the strengths and weaknesses of our team. Gelzen did the communication part of the code (80%) as he has more grasp and basic background of the HTTP protocol in general. Throughout the whole production time he has both the ESP devices and an Arduino uno so he can test and do some organization of the code structure. While Sam focused on trying to implement all the sensor readings and their calibration (20%) as he brought with him all the sensor devices and the Arduino mega for test. As for the system setup, Sam has more background with circuit development and different the component we needed, so he took on about 75% of the work for wiring the sensor to the system and connecting everything that is on the peripheral side.

# SMARTGROW



THE SYSTEM

DATA COLLECTION

```
Response code: 200
{"sensor":"Photosensor","data":257,"mode":"ON_DEMAND"}

Getting water level reading.......
HTTP Response code: 200
{"sensor":"Water Level","data":327,"mode":"ON_DEMAND"}

Getting soil moisture data.......
HTTP Response code: 200
{"sensor":"Soil Moisture","data":303,"mode":"ON_DEMAND"}

Getting Pump's state.......
HTTP Response code: 200
{"sensor":"Pump State","data":false,"mode":"ON_DEMAND"}

Getting humidity data.......
HTTP Response code: 200
{"sensor":"Humidity","data":37,"mode":"PERIODIC"}

Getting photosensor data.......
HTTP Response code: 200
{"sensor":"Photosensor","data":67,"mode":"PERIODIC"}

Getting water level reading.......
HTTP Response code: 200
{"sensor":"Water Level","data":325,"mode":"PERIODIC"}
```

AN AUTOMATED PLANT MONITORING SYSTEM

| | | Full points | Self-grading points | Brief explanation about why you claim for these points |
|---|---|---|---|---|
| Sensor data collection | periodical/on-demand report | 5 | 4 | We believe that we got all the data collection working except for the ability to configure the periodic reading. |
| | human-readable format, display on terminal | 5 | 4 | although our system only prints out information in the serial monitor, we believe that they are readable and contains all of the important information. However it is still in JSON format |
| | meaningful use of sensor data | 5 | 5 | we believe that our use of sensor data was reasonable and are essential for what we were trying to achieve with this project |
| | | | | |
| Communication | central to peripheral | 5 | 5 | Our system was able to completely communicate with each other in a reasonable timing and were no interferance from other wireless communication protocol |
| | peripheral to central | 5 | 5 | Peripheral was able to communicate with central of all the necessary data with no data lost |
| | display information about communication | 5 | 5 | The system was able to display information about the communication through the value of the HTTP response code |
| | | | | |
| Automation | no intervension | 5 | 5 | Our system was fully automated where if a certain treshold in data was met, our pump and led lights will automatically turn on or turn off |
| | | | | |
| Stability | no crash | 5 | 4 | Though our system never crashed, our pump wasn't strong enough to truly transfer water from our reservoir to our plant |
| | | | | |
| Extra sensors | extra sensor 1 | 1 | | |
| | extra sensor 2 | 1 | | |
| | extra sensor 3 | 1 | | |
| | extra sensor 4 | 1 | | |
| | extra sensor 5 | 1 | | |
| | extra sensor 6 | 1 | | |
| | extra sensor 7 | 1 | | |
| | extra sensor 8 | 1 | | |
| | extra sensor 9 | 1 | | |
| | extra sensor 10 | 1 | | |
| | | | | |
| Extra components | extra component 1 | 1 | 1 | we added a water pump to our system |
| | extra component 2 | 1 | 1 | we added an led lamp to our system |
| | extra component 3 | 1 | | |
| | extra component 4 | 1 | | |
| | extra component 5 | 1 | | |
| | | | | |
| Command and control | command/control 1 | 1 | | |
| | command/control 2 | 1 | | |
| | command/control 3 | 1 | | |
| | command/control 4 | 1 | | |
| | command/control 5 | 1 | | |
| | command/control 6 | 1 | | |
| | command/control 7 | 1 | | |
| | command/control 8 | 1 | | |
| | command/control 9 | 1 | | |
| | command/control 10 | 1 | | |
| | | | | |
| System behavior | added behavior 1 | 2 | 2 | automation for the pump based on certain reading from the soil moisture sensor |
| | added behavior 2 | 2 | 2 | automation for the led lamp based on certain reading from the photosensor |
| | added behavior 3 | 2 | | |
| | added behavior 4 | 2 | | |
| | added behavior 5 | 2 | | |
| | | | | |
| Fancy features | | 5 | | |
| | | 80 | 43 | |