Germinator: Plant Monitoring & Germination Project

Harsh Joshi

Humber College

Author Note

Harsh Joshi is a student at Humber College studying in the Computer Engineering Technology program. This report highlights the Capstone project done by Harsh Joshi for his final year. Special thanks to Dr. Kristian Medri for being a mentor. The project's PCB and acrylic box casing were made possible by the Prototype lab technicians Kelly Gray and Vlad Porcila at Humber College. Consultation for the plant material was made possible with the help of the greenhouse technician Valeria Wuschnakowski at Humber College. Contact Harsh Joshi at harshjoshi114@gmail.com

Project GitHub website: https://github.com/mynameisyoshi/mynameisyoshi.github.io.

Declaration of Authorship

I, Harsh Joshi, confirm that this work submitted for assessment is my work and is expressed in my own words. Any uses made within of other works of any other author, in any form (ideas, equations, figures, previous technologies, tables, programs, texts) are properly acknowledged at the point of use. A list of the references used is included. I, Harsh Joshi, have handled all aspects of this project.

Proposal

Executive Summary

As a student in the Computer Engineering Technology program, I will be integrating the knowledge and skills I have learned from our program into this Internet of Things themed capstone project. This proposal requests the approval to build the hardware portion that will connect to a premade website that allows the user to view the data in a well-organized fashion as well as to a mobile device application that extends the website and allows the user to see the data from the sensors on the custom PCB. The internet connected hardware will include a custom PCB with sensors and actuators for the measuring of humidity, moisture, light, and temperature and a watering pump. The mobile device and the website functionality will include the ability to see the current data. I will be collaborating with Valeria Wuschnakowski, greenhouse technician at Humber. The hardware and the application will be completed by and integrated together in CENG 355 Computer Systems Project.

Background

The problem solved by this project is that small-scale farms, gardens, and greenhouses need accurate ways of measuring the humidity, moisture, light, temperature, and water level of their plants. My project will help the users of the system to determine which circumstances the plant germination process (the process where plants are starting and growing from seeds) will occur the best. By having a smart plant monitoring system in place, plants can have their needs taken care of easily.

Concluding Remarks

This proposal presents a plan for providing an IoT solution for the accurate plant monitoring and germination. This is an opportunity to integrate the knowledge and skills developed in our program to create a collaborative IoT capstone project demonstrating my ability to learn how to support projects. I request approval of this project.

Abstract

Small-scale farms, gardens and greenhouses have a need for consistent and accurate information regarding their crops. This need could be addressed with an automated electronic system that updates users on the changes in the plants' environment. This plant germination and monitoring system displays readings of the plants' ambient temperature, relative humidity, light levels, and the soil moisture levels. This system also allows the plants to be watered automatically if the soil moisture levels are low. The data readings from the system's sensors is uploaded and stored on an online database for easy and convenient retrieval by the user. The online database is accessible using a specialized website and Android application. The Android application and the website allow the user to access recently gathered data using tables and easy-to-read graphs. With the inclusion of the above features, this plant system provides users, especially in small-scale farming, gardening and greenhouse work, with the ability to monitor and water their plants easily.

Table of Contents

Introduction

Monitoring the growth of a plant can be a time-consuming process despite the fact that small-scale growing operations like farms, gardens and greenhouses have more control of their environment. It can be difficult to afford complex systems that can be used to monitor and control these operations' environment. In order to address this issue, I have created an electronic plant monitoring system which can be implemented for a relatively low cost per plant and will also water the plants as needed.

This system will store the data in an online database and is accessible via a website dashboard interface and an Android application. This provides easy and convenient plant monitoring as growers would not be able to if using an in-house only system without this feature.

This system will be valuable to growers that cannot afford other monitoring systems as those systems tend to complex and, as a result, expensive. This system is lightweight and scalable as small-scale farmers and greenhouse users along with home gardeners can benefit from this type of system. This system also has isolated electronics as a preventative measure against water and other environmental damage.

The source code used in this system project is available in accordance with the GPLv2.

Project Overview

Problem

Small-scale farms, gardens and greenhouses are missing a smart monitoring system that can take care of watering and monitoring plants on a daily basis. Users have to stretch out a line of pipe in order to water the plants every day and must wrap it back up at the end of the day. Users would also need to measure the plants’ condition accurately and frequently in order to ensure the plants’ health.

Rationale Behind Project

This project is a prototype that allows us to solve this issue as this is a smaller scale of what we can expect in the future, where the monitoring system will monitor light, soil moisture, air humidity, and the temperature of a small-scale farm, garden or green house.

Project Scope

There is only the Arduino that needs to be programmed with the algorithm to water the soil. The algorithm depends on the soil composition and its ability to retain liquid within itself in order to keep the plant moist. The other part is connecting with thinger.io website in order to send the data from the Arduino. Finally, the Arduino needs to be connected to a network, preferably by Wi-Fi.

Bill of Materials

- Photo resistor
- 2 10kΩ resistors
- 1x MKR 1000 Arduino board
- DHT 11 sensor
- Soil moisture sensor
- Water pump

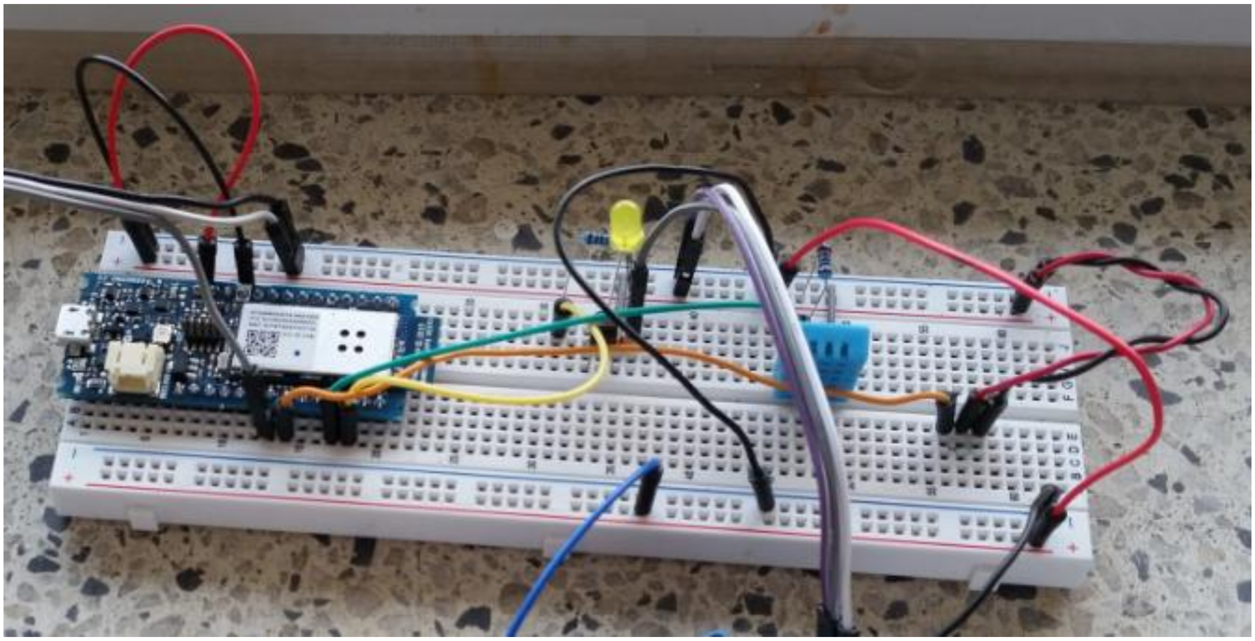
These materials will need to be placed in a case in order to make this portable and waterproof. This case should have 7x3x3 inches as inside dimensions and can be designed using the makercase.com website.

Time Commitment

The time commitment is around 6 to 8 hours of construction. This does not include the time needed to manufacture the PCB, collect the sensors, or to solder the components together. This project can be finished within 2 to 3 days and includes troubleshooting the sensors and connectivity with the web application.

Mechanical Assembly

It is recommended to use a solderless breadboard to wire the components initially and to use the 3.3V power from the MKR1000.



PCB & Soldering

The PCB can be designed using specialized software. For this project, Eagle was used to design the board layout and schematics beforehand. The design must then be printed out into a PCB and the components can then be soldered onto the PCB.

Power Up

For the first power up, using the power from the MKR1000 would be okay. If using an external source of power (for example, a USB), the power source must have its voltage level measured and tested on a solderless PCB board. If the MKR1000 is transmitting data to the website when the Arduino serial monitor is open, the graph will be created from the data log if it has been set up already.

Soil Composition

The soil composition for the plant is really important as it must be in the right ratio of materials. Without this, the soil can be affected from the soil moisture. It can slowly degrade the fiberglass material the material is made out of. For this project, Canadian Sphagnum peat moss (which makes up 75% to 80% of the soil composition), perlite (volcanic rocks), vermiculite, dolomitic and calcitic limestone (which is a PH adjuster for the soil), a wetting agent, and Mycorrhizae fungi to keep the soil moist at all time. The plants do not need to be watered daily as soil will be wet from the pump that will turn on once a day for 10 seconds.

Plant Material

For this project, plants that are okay with being indoors and that do not require a lot of water are ideal. I have found that using *Chlorophytum comosum* (also known as the Spider Plant) is the best species to use if the device is kept indoors.

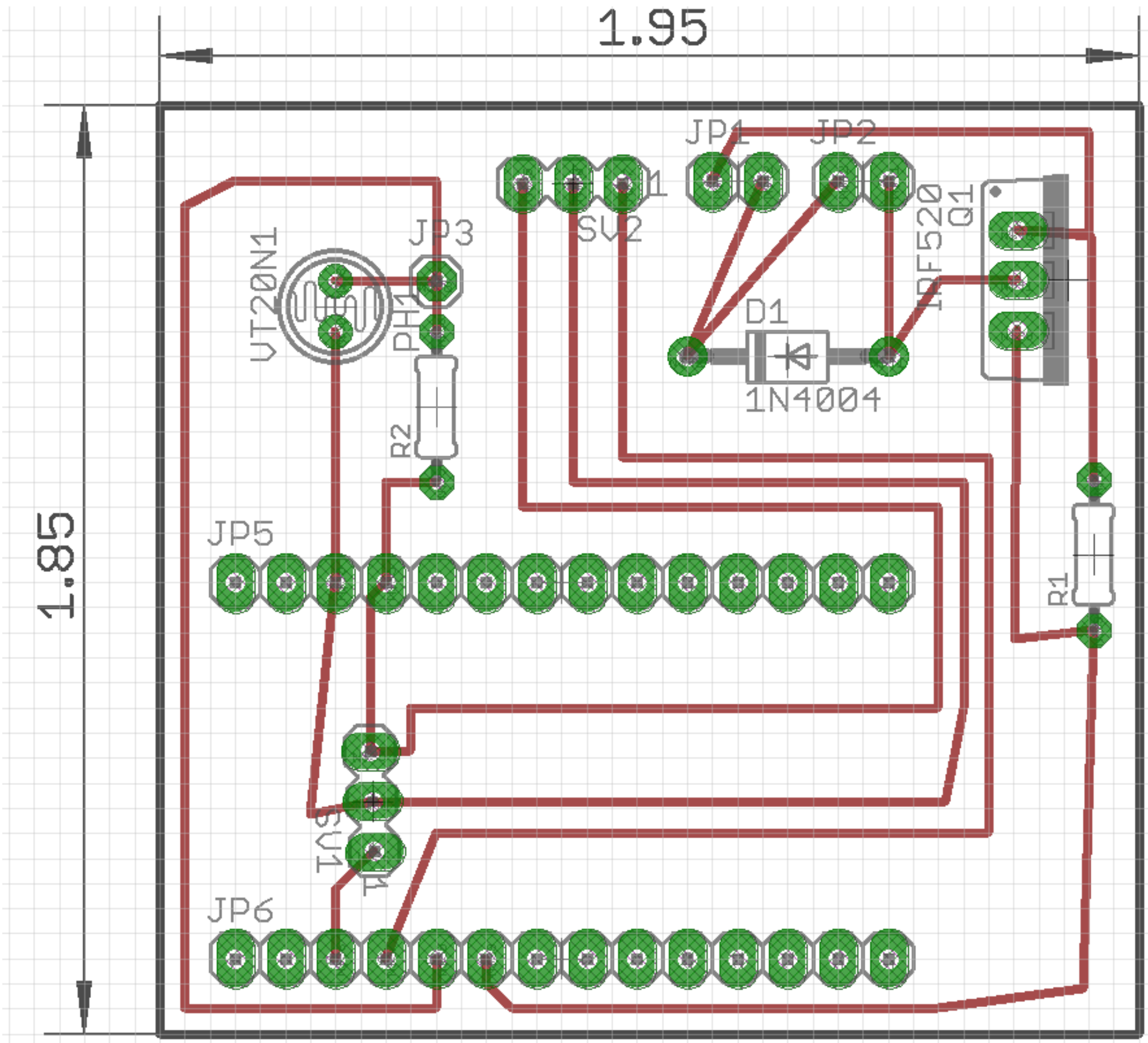
Challenges & Recommendations

There are many challenges that users can face in building this project. The MKR1000 board uses Wi-Fi to connect itself to the website. As a result, the user will have to go through the code and change the SSID and the password manually. A way around this issue is to use a Raspberry PI computer with a Wi-Fi dongle that can be used as a dedicated router for this project. The user will have to have to attach an Ethernet cable to the Raspberry PI. Another issue crops up from the portability of the systems: if the Raspberry PI is stationary, the user could take the project and put it over the plant but supplying the PI with power becomes the issue. There are two ways around this situation: either using two 1.5V AA batteries or using a simple rechargeable power bank can supply the system with power. The MKR1000 has the code build-in so the user does not have to set up the Arduino every time they wish to use the system. The motor mentioned in the project is an external motor with two plastic tubes coming out. This can be switched with a submersible pump like those that are found in aquarium tanks as these are water proof. The MKR1000 can send a signal to a MOSFET or relay that can turn the pump on or off. This change will only one tube coming out of the pump and into the plant: this will prevent water from coming near the circuit board. Instead of having a separate voltage source connecting to the motor, the available 5V from the MKR1000 can be used to drive the motor. This would not be recommended due to the motor current intake. The MKR1000 can also be powered via the input pins with 3.3V rather than using the USB port and the custom PCB can also have a built-in power intake.

Conclusion

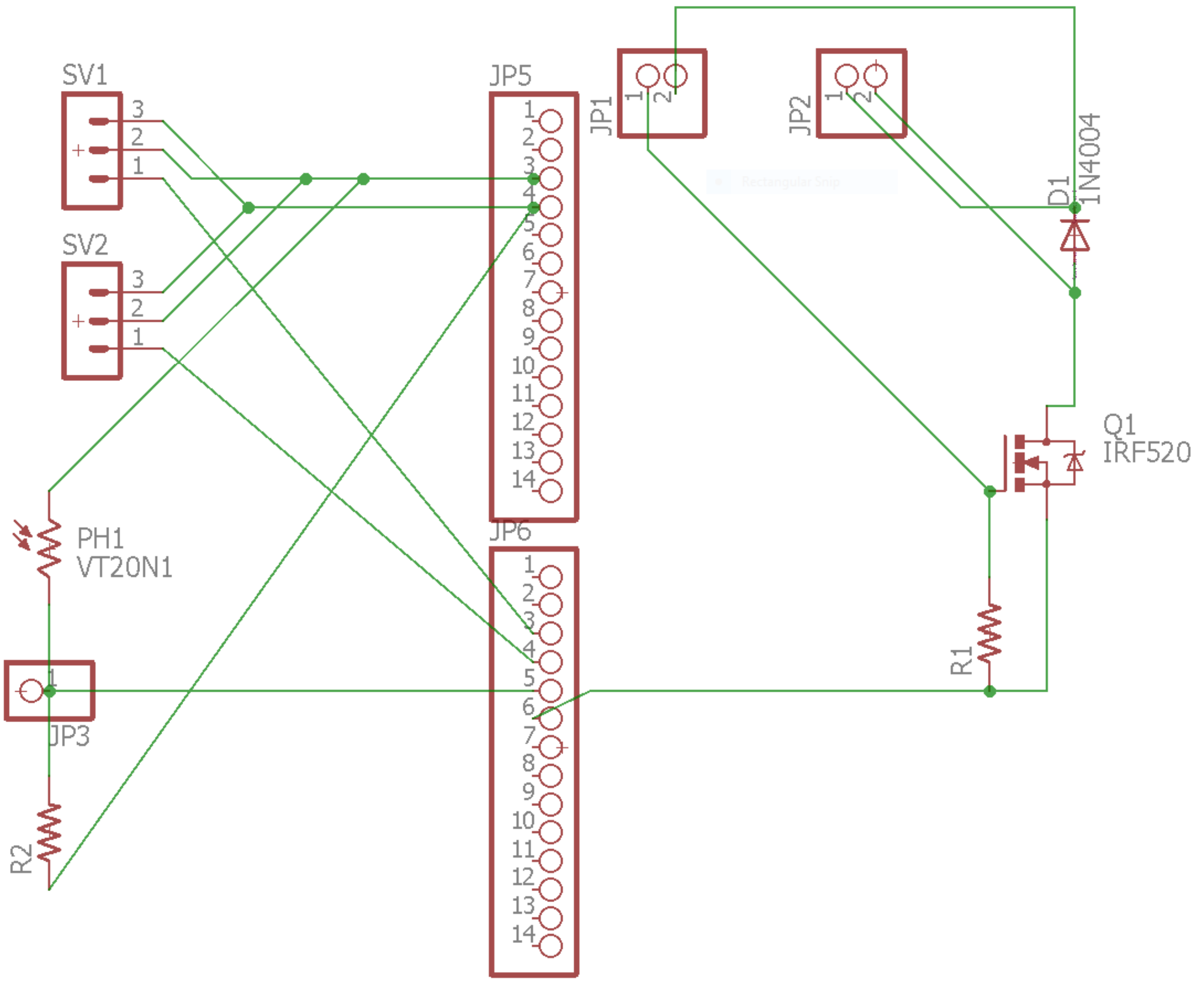
The main idea of this monitoring device is to track the temperature, humidity, light level, and soil moisture, of a plant, and to control a water pump in order to enable remote watering. This project does exactly that and allows for manually control of the pump or automated watering. This project has solved the major issue of needing accurate and convenient plant watering and monitoring. This project can be scaled up to use more types of sensors and more pumps. This system can be used as a prototype for smart and efficient monitoring and watering of plants.

Board Layout & Schematics



The above image is of the board layout, where we have several components. JP5 and JP6 is for the MKR 100 Arduino board, which is the main controller. R1 and R2 are the 10kΩ resistors. SV1 is for the soil moisture sensor. SV2 is for the DHT11 sensor. We can also see that we have a photo resistor for our light sensors. The sensors need 3.3V, which is all powered by the MKR100 board. The board itself is powered by a 5V USB, but further integration will include a battery pack for portability. The entire size of the board is 1.95 inches by 1.85 inches.

The below image is of the PCB schematics:



Dashboard

The dashboard is where the user can see the data coming in from the MKR1000. The dashboard can be accessed using the thinger.io website. The dashboard displays the values in a graph or using a donut chart. The dashboard also has on and off buttons that allow the user to turn the LED on/off and can run the motor for 5 seconds.

Android Phone Application

The Android application connects to the thinger.io website via the built-in libraries and is able to retrieve the sensor data from the plant monitoring and germination system. The Android application does vary its data by 10% of the value shown on the dashboard because the values are closer in representation to the system’s raw data whereas the dashboard website displays a value that is calculated from the raw data. Within the application, the user will have to press the refresh button in order to retrieve new data.

Arduino Source Code

```
#define _DEBUG_

#include <SPI.h>
#include <WiFi101.h>
#include <ThingerWifi.h>
#include <DHT.h>

#define USERNAME "your_user_name"
#define DEVICE_ID "your_device_id"
#define DEVICE_CREDENTIAL "your_device_credential"

#define SSID "your_wifi_ssid"
#define SSID_PASSWORD "your_wifi_ssid_password"

ThingerWifi thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);

const int LedPin = 6;

// PIN configuration
const int DhtSensorPin = 2;
const int WaterRelayPin = 3;
const int SoilMoistureSensorPin = A5;
const int LightSensorPin = A6;

int ledState = LOW;
int waterRelayState = LOW;

const int DhtType = DHT11;

DHT dht(DhtSensorPin, DhtType);

void setup() {
      Serial.begin(9600);
      dht.begin();

      // configure wifi network
      thing.add_wifi(SSID, SSID_PASSWORD);

      pinMode(LedPin, OUTPUT);
      pinMode(WaterRelayPin, OUTPUT);

      // Configure the LED
      // Need to track the state separatelly from the real pin, since MKR1000
does not respond the correct value when reading an output pin
      thing["led"] << [](pson & in) {
            if (in.is_empty()) {
                  in = ledState;
            }
            else {
                  ledState = in ? HIGH : LOW;
```

```
			digitalWrite(LedPin, ledState);
		}
	};

	// Configure the Water Relay
	// Need to track the state separatelly from the real pin, since MKR1000
does not respond the correct value when reading an output pin
	thing["water"] << [](pson & in) {
		if (in.is_empty()) {
			in = waterRelayState;
		}
		else {
			waterRelayState = in ? HIGH : LOW;
			digitalWrite(WaterRelayPin, waterRelayState);
		}
	};

	thing["dht11"] >> [](pson & out) {
		out["humidity"] = dht.readHumidity();
		out["celsius"] = dht.readTemperature();
		out["fahrenheit"] = dht.readTemperature(true);
	};

	thing["light"] >> [](pson & out) {
		out = map(analogRead(LightSensorPin), 0, 1023, 0, 100);
	};

	thing["moisture"] >> [](pson & out) {
		out = map(analogRead(SoilMoistureSensorPin), 0, 1023, 0, 100);
	};
}

void loop() {
	thing.handle();
}
```

Website Creation Process

Now you will need to create an account on thinger.io. The next step will be to add the device to the website and to make the dashboard.

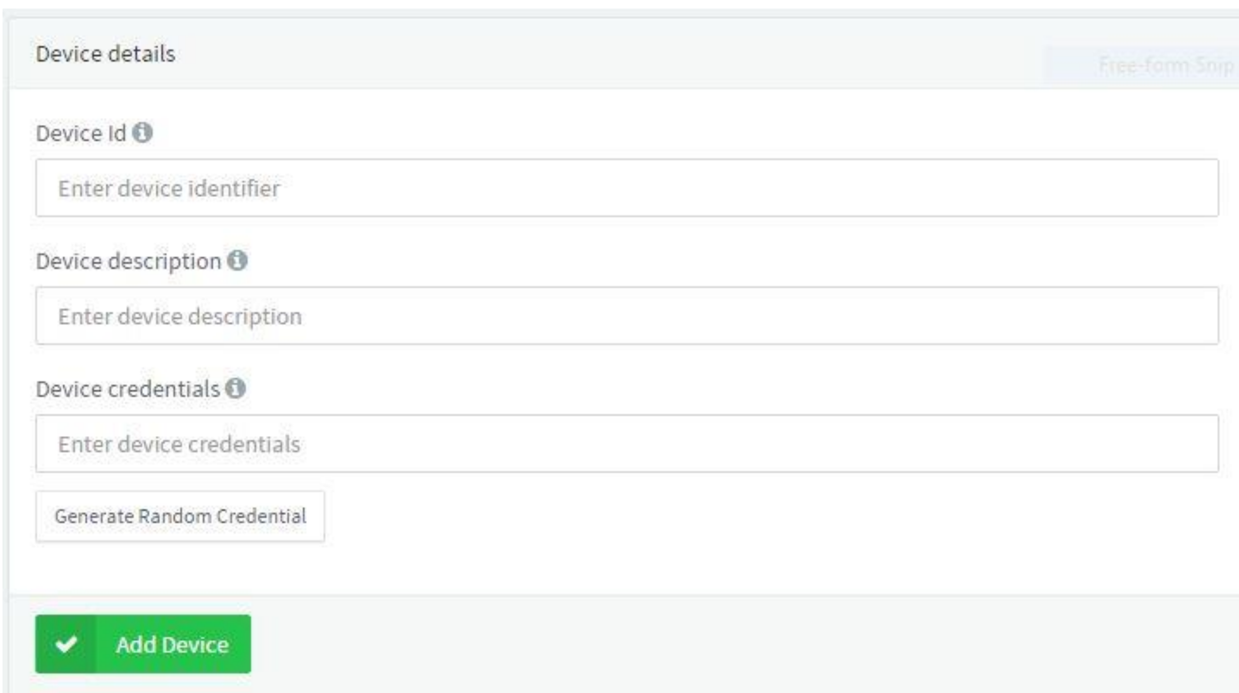
```
#define SSID "your_wifi_ssid"
#define SSID_PASSWORD "your_wifi_ssid_password"
```

This function will take the SSID and the password, which has to be hardcoded into the Arduino.

To get your device connected to the web interface, you will need to use this code:

```
#define USERNAME "your_user_name"
#define DEVICE_ID "your_device_id"
#define DEVICE_CREDENTIAL "your_device_credential"
```

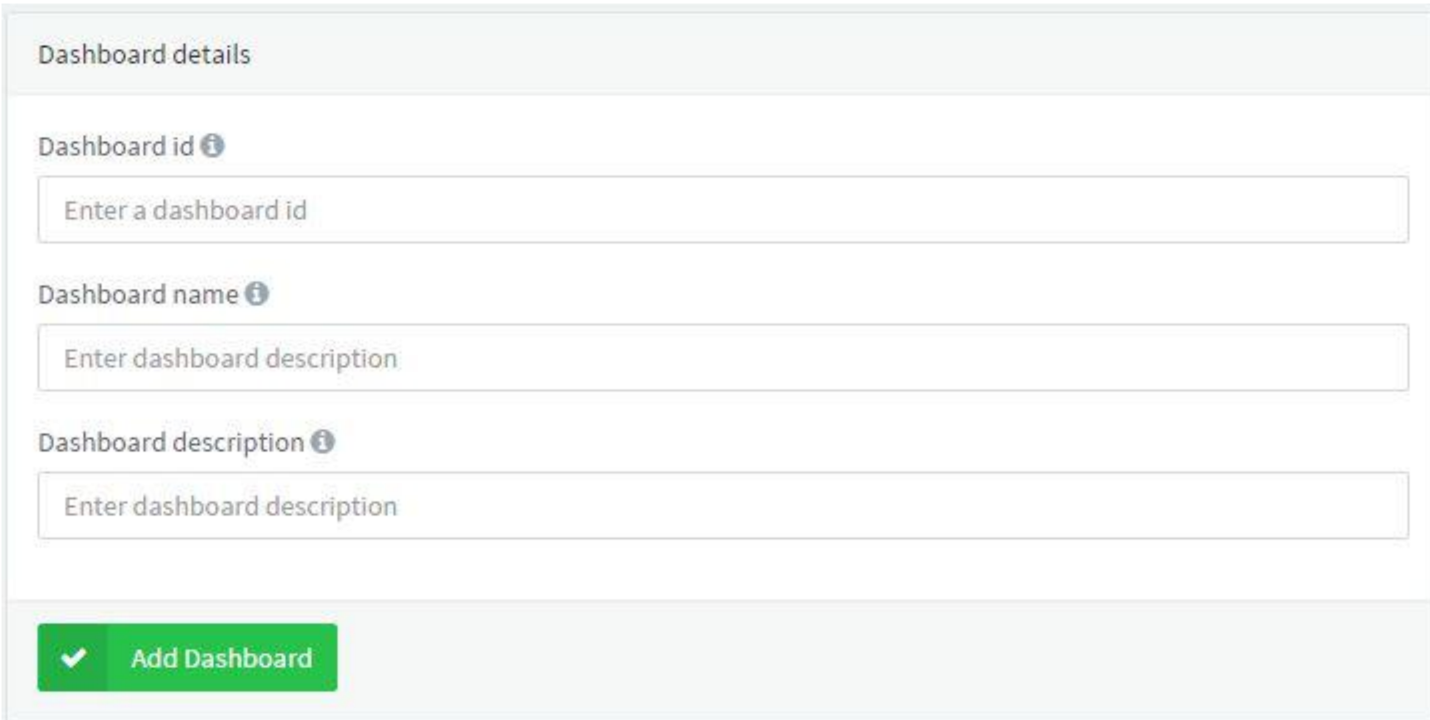
The first line takes in the username that you gave the website.



The device ID, and credentials are made by you but you can randomly generate the credentials as well. When you are done, you can copy and paste in your code.

Dashboard

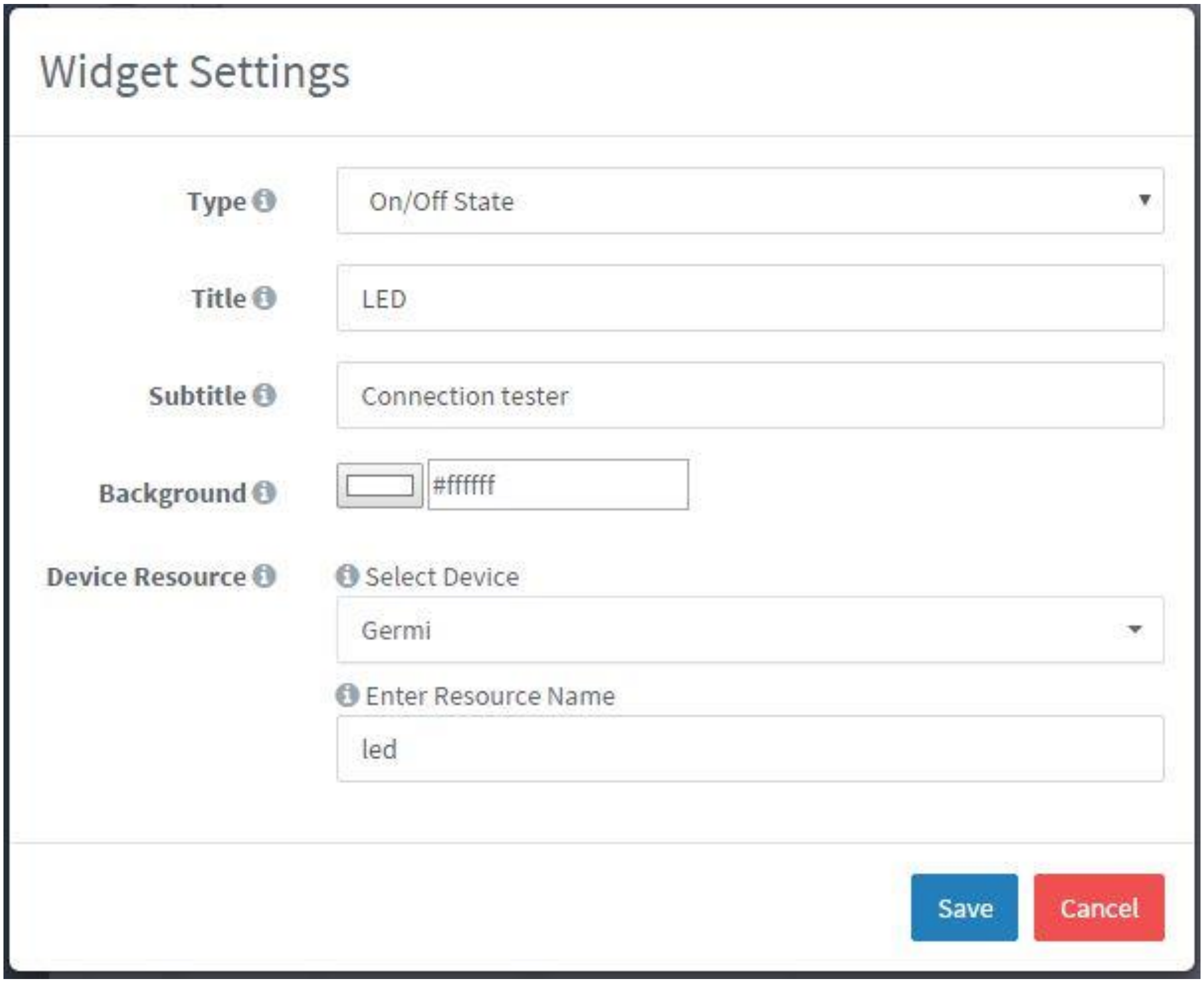
Creating a dashboard allows you to see the data retrieved from the sensors.



In the website you create an ID, name, and a small description of the dashboard.

	Dashboard	Name	Description
	Germi	Data taker	Takes in the data

After creating a dashboard, you will see the below image. You can now enter in the dashboard and make donut charts that will update at a set interval.



As an example, we will set up the LED on/off button that allows the user to turn the LED on/off on the board to see if the connection is live.

```
// Configure the LED
// Need to track the state separately from the real pin, since MKR1000 does
not respond the correct value when reading an output pin
```

```
thing["led"] << [](pson & in) {
if (in.is_empty()) {
in = ledState; }

else { ledState = in ? HIGH : LOW;
digitalWrite(LedPin, ledState);
} };
```

In the code, we have a boolean button that goes high : low and digitalWrite.

Final System Version

The above image is of the Germinator Plant & Germination System in its final version.

References

Gill, R. (2016, March 30). Plant Monitoring System. *Arduino Project Hub*. Retrieved from https://create.arduino.cc/projecthub/ryanjgill2/plant-monitoring-system-88ed2b?ref=tag&ref_id=environmental%20sensing&offset=1

Taibo, N. (2016, June 14). Arduino MKR 1000 Plant Monitor – Project Share - The Internet of Thinger. *Thinger.io Community*. Retrieved from https://community.thinger.io/t/arduino-mkr-1000-plant-monitor/126

Progress Reports

Capstone project

Harsh Joshi <harshjoshi114@gmail.com> Thu, Mar 23, 2017 at 1:26 PM

To: muhammad.virk@humber.ca

Cc: Vincent Shaikh <vincent.shaikh@humber.ca>, Kristian Medri <kristian.medri@humber.ca>, Mihyar Hesson <mihyar.hesson@humber.ca>

Hello,

Thank you to Mr. Muhammad Virk for accepting my capstone project and keeping it in his office. The project is working and send the information to thinger.io and allows you to turn the pump on or off, or the Arduino will do it it self.

The credentials for the project i will send in a separate email to Mr.Virk and how to get he information on his smartphone. There is the issue of humber WiFi not allowing a static IP address to the Arduino MKR1000. I will bring in a raspberry pi 0 with WiFi dongle and that will solve that issue as well.

Thank you so much Mr. Virk for giving my project a new home and I hope you like the new edition in your office plant collection.

--

Thanks,

Harsh Joshi,

Computer Engineering Technology, Humber College

Troubling shooting

Harsh Joshi <harshjoshi114@gmail.com> Tue, Mar 21, 2017 at 10:28 AM

To: Kristian Medri <kristian.medri@humber.ca>

Hello,

There are many challenges that the users can face. The MKR1000 board uses wifi to connect it self with the website. So the user will have to go in the code and change the SSID and the password manually. One way around it is to have a raspberry PI that has a WiFi dongle that can be used as a router just for the project. The owner will have to have to attach a ethernet cable to the Raspberry PI. Second problem comes as portability if the Raspberry PI is stationary the user can take the project and put it over a plant but supplying it power is the issue. There are two way around this situation. Using two 1.5v AA batteries or a simple rechargeable power bank. The MKR1000 has the code inbuild so the user doesn't have to set up the Arduino every time. The motor mentioned in the project is and external motor with two tubes coming out. We can switch that with a submersible pump, the one found in aquarium tanks that are water proof and have the MKR1000 send singe to a mosfet/relay that can turn it on or off so we have only one tube coming out and going to the plant and no water coming near the circuit board. Instead of having a separate voltage coming in for the motor we can use the available 5v coming from the MKR1000 to drive the motor. But it would not be recommended due to the motor current intake. We can also power the MKR1000 via the input pins with 3.3v rather than USB port and have a power intake build into the custom PCB.

--

Thanks,

Harsh Joshi,

Computer Engineering Technology, Humber College

Kristian Medri <Kristian.Medri@humber.ca> Fri, Dec 9, 2016 at 6:07 PM

To: Harsh Joshi <harshjoshi114@gmail.com>

Cc: Vincent Shaikh <Vincent.Shaikh@humber.ca>, Mihyar Hesson <Mihyar.Hesson@humber.ca>

Dear Harsh,

Regarding today's demo, I indeed saw that you were able display readings from a light sensor. You are still struggling with using a microcontroller to turn a DC motor and we found that to be in part due to a floating reference today. I look forward to seeing the new interface PCB that you plan to make help you be able to control the motor with the light sensor.

Sincerely,

Kristian

From: Kristian Medri

Sent: Friday, December 09, 2016 6:08 PM

To: 'Harsh Joshi'

Cc: Vincent Shaikh; Mihyar Hesson

Subject: RE: Today's meeting

Dear Harsh,

Regarding today's demo, I indeed saw that you were able display readings from a light sensor. You are still struggling with using a microcontroller to turn a DC motor and we found that to be in part due to a floating reference today. I look forward to seeing the new interface PCB that you plan to make help you be able to control the motor with the light sensor.

Sincerely,

Kristian

Thursday November 24, 2016 2:29pm-3:40pm, Kristian and Harsh will continue to meet one more time this semester in NX301G North on Tuesday 12/06/2016 from 11:15 AM to 12:15 PM before planning to meet in class next term.

Attendees: Kristian Medri, Harsh Joshi (waiting at reception 2:17pm, in meeting 2:29pm)

Continued discussion of future work in Computer Systems Project (Winter 2017)

Harsh	Kristian
• Harsh says learned about surge… • Acrylic box made held on with tape temporarily till everything is finalized • Everything is recycled acrylic • Made a PCB board for the controller of the motor and to house the sensors. • Using ribbon cable to join the custom PCB board to the Arduino board for more flexibility during construction. • Made customer mount to mount the Raspberry pi to the Arduino and a base plate for the Arduino with recycled ply wood all laser cut • Attached a functioning 12 v motor that runs when 5v is supplied to the mosfet transistor via a1 in the Arduino . • Waiting on the sensor to arrive till then working on the server side and	• Proposal not uploaded to: https://github.com/mynameisyoshi/mynameisyoshi.github.io/blob/master/documentation/ nor documentation site updated since our last meeting. • How about an update on my comment at the last meeting about: "**References to prior art: existing products, ongoing research not provided**. *However, previous student were consulted. It is very important to visit at least the websites of garden supply stores such as Lee Valley, Nurseries, hardware stores, as well as well-known hacking sites such as instructables and adafruit.* " • Which Lee Valley location did you visit? • It would be good to write information you gather to your GitHub site. What about watering systems that they offer?

programming to send data to the SQL server. • Also waiting on new Raspberry pi an figuring out how to power PI and Arduino separately so no future power surge occurs • Sensors are being brought from adafruit • Did visit several nurseries i.e. Lee Valley to get recommendation on fast growing seedling for the project and soil type 25% sand and 75% soil was advised to me. At 590 king st. • Did not ask about the existing system. • So the surge happened when I connected the raspberry pi to the arduino via a port in my laptop that gives 5v at 2amp (settings in the BIOS for power). • Got a messages in my laptop saying “the USB port is consuming more power than it can supply” after immidetailly removing it from the computer and check only the arduino. The arduino was fine, than moving on	• That was one of the main purposes of going there. • Tell me more about the surge that you experienced. • Clarifying: USB A laptop socket was connected to Broadcom development platform via a micro USB cable. The Arduino was already connected to the Broadcom development platform via a type-A plug to type-B plug USB cable. • SD cards in parts crib are for model 2 development boards and they are not compatible with the model 3 development boards unless they are a fresh install. It is worth trying your development platform with a known good image and power supply. Low voltage supplies are known to corrupt SD cards. • Reclaimed acrylic. • Ribbon cable should have been connected to a socket as opposed to directly soldering it to the PCB. • It may make things easier in the future if your MOSFET driven DC motor interface is also compatible with the 3.3V Broadcom development platform GPIO. • Currently missing separate 12VDC power

to the Rasberry pi notice that it would not post. • Tired different memory cards and ever different power supply but no boot • After a quich google serch on the error code that i got on my laptop it said the USB had a power surge. • USB port works fine but the Raspberry Pi I not working. • Replacement on its way with the sensors which were ordered Nov 16/16. • 1 x Light sensors GA1A12S202 • 1 x TMP36 analog temperature sensor • Moisture sensor NEXT0659 made by NEXT on amazon. • Was advice to use a water tight container due to own effects of the acrylic glue having cracks when it dry allowing water to flow within them.	supply. • Do you have something to demonstrate? • Which sensors were ordered and when was the order placed? (ard0659) • A rough exterior case has been made but what about soil, seeds, etc. and their respective compartments? • Silicone should be used to seal acrylic compartments. • Regarding “Engineer” From: https://www.ontario.ca/laws/statute/90p28 Every person who is not a holder of a licence or a temporary licence and who, (a) uses the title “professional engineer” or “ingénieur” or an abbreviation or variation thereof as an occupational or business designation; (a.1) uses the title “engineer” or an abbreviation of that title in a manner that will lead to the belief that the person may engage in the practice of professional engineering; (b) uses a term, title or description that will lead to the belief that the person may engage in the practice of professional engineering; or (c) uses a seal that will lead to the belief that the

	person is a professional engineer, is guilty of an offence and on conviction is liable for the first offence to a fine of not more than $10,000 and for each subsequent offence to a fine of not more than $25,000. R.S.O. 1990, c. P.28, s. 40 (2); 2001, c. 9, Sched. B, s. 11 (59). • From: http://www.peo.on.ca/index.php?ci_id=2058&la_id=1 • The minimum educational level stipulated by the Professional Engineers Act is: • a three-year diploma in technology from a College of Applied Arts and Technology; or • a bachelor's degree in a relevant science area; or • academic qualifications deemed by the Council to be equivalent to a diploma or degree. • http://www.peo.on.ca/index.php?ci_id=2059&la_id=1 • Ontario law requires that you have four years of engineering work experience before you can obtain your professional engineering licence. • There is also a Professional Practice Exam. • There is one more meeting, had hoped market

	survey had been done by the previous meeting. We had discussed that the final meeting would have hardware assembled ready for power up, such that improvements and debugging could take place over the holidays.