Germination Project

Harsh Joshi

Humber College

Author Note

This project is for monitor the external environment around plants that are within someone home.

Contents

Abstract

There may be a huge want for small-scale agricultural operations to constantly show the surroundings round newly growing plant life. To treatment this, a device want to be created to actively display the surroundings at a few levels within the germination manner of a plant in a manner that permits a customer to remain updated on the environment at the same time as not having to be present. This tool reveal ambient temperature, relative humidity, moderate levels in the direction of the plant, and the moisture of the soil, and allow the plant to be watered ought to the soil moisture drop too low. those sensors will hold the facts briefly at the development Platform earlier than it is miles dispatched to an offsite database. This website will show all the sensor information, and make it available to every Android application. The software will show the maximum modern-day information entries, similarly to a limited history of statistics entries from the website, along side graphs for ease of viewing. The internet website will show the maximum present day batch of entries, and offer the entire facts for each character sensor's facts, further to a graph for ease of viewing traits. This machine has the capability to make small-scale growing operations less complex to display, allowing customers to check their plant life' situations from everywhere at any time, supplied they've got internet connectivity.

Germination Project

This project allows the user to look at the environment that their plant is growing in. The user can do so in two ways. One being using the android app or using the web application made by thinger.io.

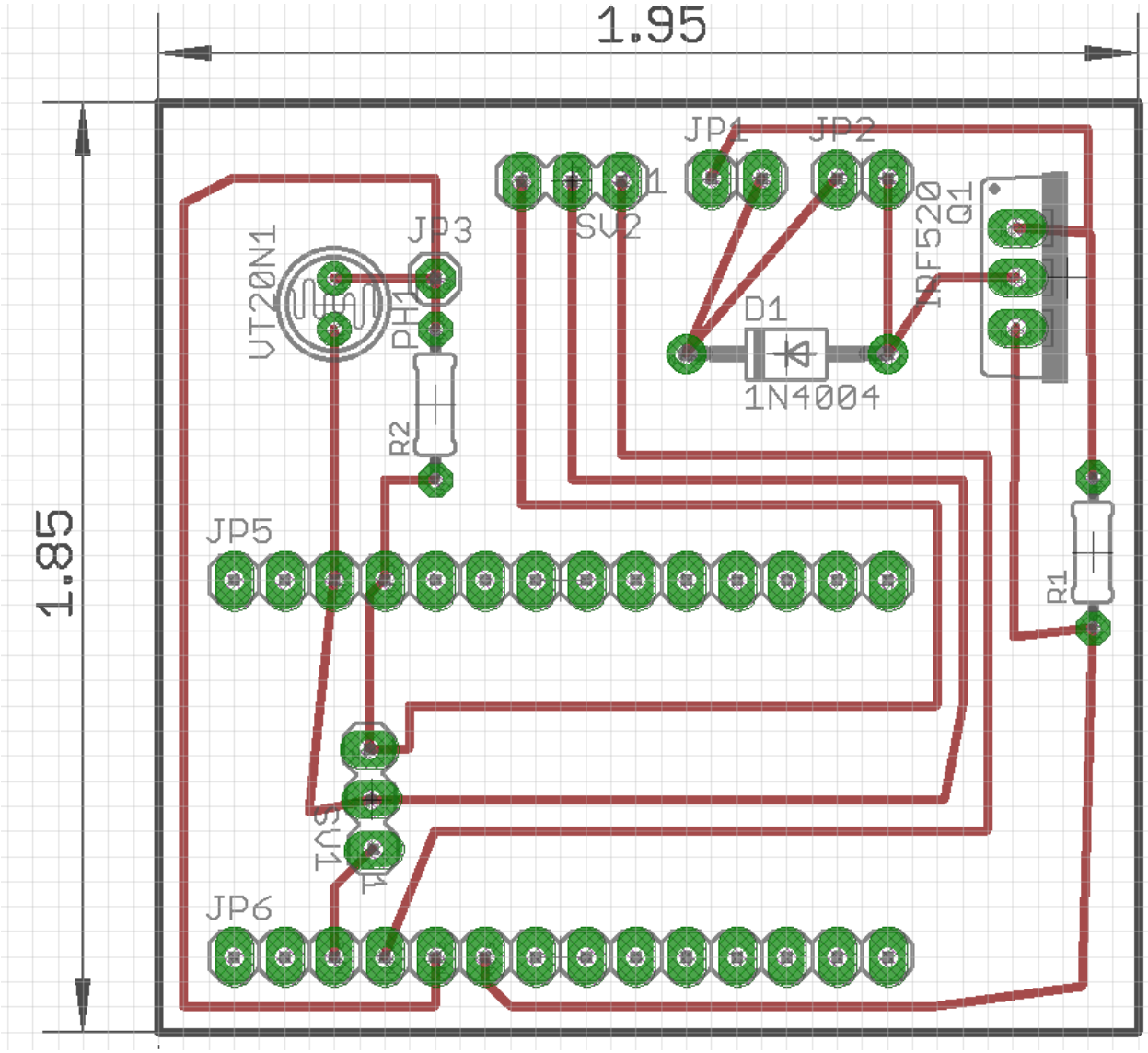
Declaration of Authorship

I, Harsh Joshi, confirm that this work submitted for assessment. 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.

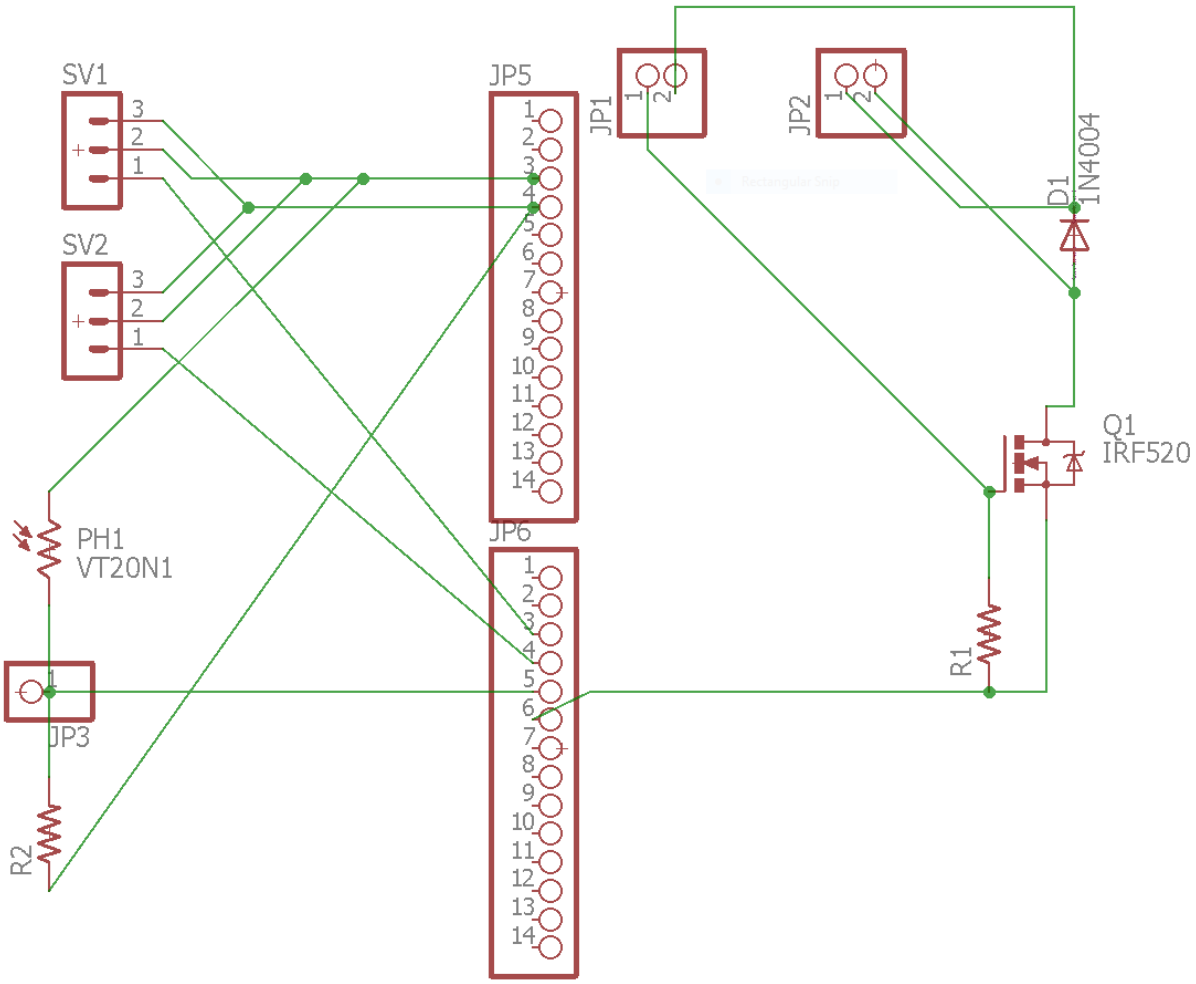
Executive Summary[1]

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 premade website that allows 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. The mobile device functionality will include the ability to see the current data.

Schematics.



As we see on this board layout, we have several components. JP5 and JP6 is for the MKR 100 Arduino board that is the main controller. R1 and R2 are 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 sensor have 3.3v going to them all powered by the MKR100 board. The board it self is powered by 5v USB, but later integration can add a battery pack that will make it portable. The entire size of the board is 1.95 inches by 1.85 inches.



Equipment's.

Quick list of the equipment that you will need to make this project.

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

Now you will need a case to put this all into and make it into a portable/waterproof design. To do so we can go on to makercase.com and build a case that is 7x3x3 inches (inside dimensions).

Challenges/Recommendations

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.

Programming

Arduino 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();
}
```

References

Gill, Ryan, F. M. (2016). Article Title. *Plant Monitoring System*, Pages From - Arduino Projects

nahueltaibo, F. M. (2016). *Plant monitoring.* Name: thinger.io projects.