

My E-Gardener

IT-project with regard to our B.Sc in Software
Engineering at the University of The Faroe
Islands

**Kristmund Ryggstein and Hergeir
Winther Lognberg**

Faculty of Science and Technology
Benadikt Joensen, Gunnar Restorff
University of the Faroe Islands
January 2019

Abstract

We were asked to do an it project using an Arduino Uno. We chose to do an automated gardening system with it, because that is relevant for some work projects that are currently in progress on Sandoy, where we are from. Our Arduino project entails an automated system that waters, measures moisture, heat and light exposure. That is done by customizing an Arduino Uno which has a moisture-, heat and light sensor that does all our measurements. All the data is then collected by the Arduino, and then sent to a Web Service via a esp-05 wifi module. The Web Service receives the data, and presents it to the user. The user is registered with the Web Service, and can track his plant data through his registered user account, where he also can customize water input for the plant/s.

Contents

Abstract

Acknowledgments

1	Introduction	1
2	Setup	1
2.1	FC28	1
2.2	TMP Sensor	2
3	Attachments	4

List of Figures

1	Corroded fc-28 after moderate use.	4
---	--	---

List of Tables

Acknowledgments

1 Introduction

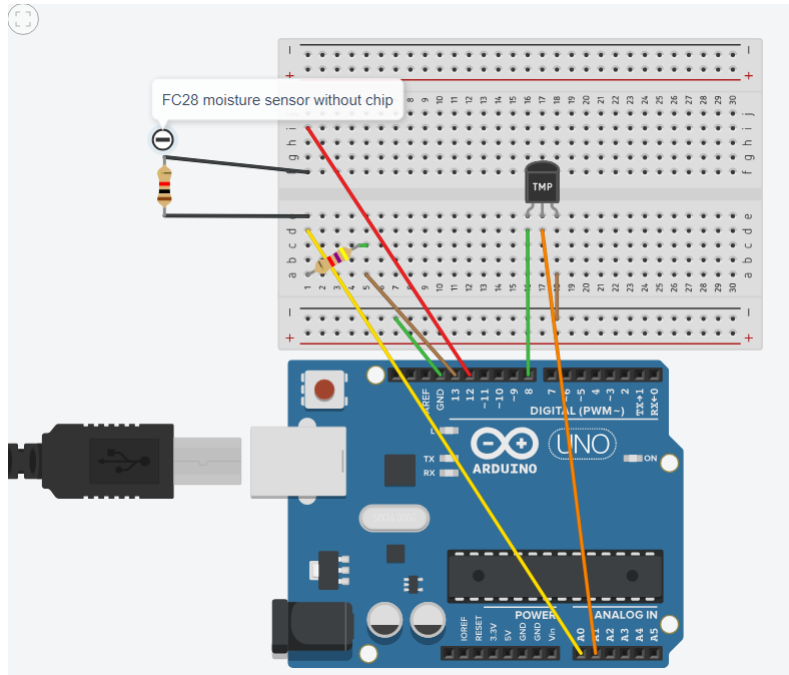
We have chosen to work with automatic a gardening system because it has some tasks, what we believe, are tedious and easy to forget. There has been an increasing interest on Sandoy in trying to grow more vegetables on the islands, and to sell them. That grown interest in thanks to initiatives such as "Veltan" and "Eplafestivalurin". Trying to be self-sufficient with regards to food produce is, according us, a good idea, and that's were we think that our modest electronic gardening project fits well with that philosophy, and will be an interest for those, who want to try to be more self-sufficient.

2 Setup

The Arduino Uno, TMP36GZ heat sensor, resistors and the cables which we use in this project, was provided by the one responsible by the course. We used at the start a FC28 moisture sensor, but it generally uses DC current which means that it corrodes quickly because of electrolysis in the soil and alters the soil composition which could potentially damage the plant, which would eventually lead to tainted readings. We changed it to use an AC pulse that can be seen in figure, but we decided to opt for Capacitive Soil Moisture Sensor V1.2 that is Corrosion Resistant. The FC28 sensor can be seen in figure 1.

2.1 FC28

Our intention was to use the Arduino to induce an alternating current through the sensor. To simplify things, we circumvented the circuit of FC-28 such that we connected directly to the pins on the sensor. To alternate the current, we first emit 5V from pin 12 and set pin 13 to low (0V) for 1 ms then we take a reading through the analog pin A0. Then we flip the pin 12 and 13 (5V from pin 13) for the same amount of time. This should somewhat remedy the problems of electrolysis by shifting the ions back and forth between the moisture sensor pins, instead of just letting them congest at one pin. The chip had to be removed so that it could use AC. The setup can be seen above.



The resistor in the circuit helps increase the amount of current entering the analog sensor. Without it about half the current would go in the pin 13 when the measurement occurs. We are therefore able to steer the range of current entering the analog input. The left leg is connected to 8 digital pin, the middle leg is connected to A1 pin, and the third leg is connected to -18 on the breadboard which is then connected to the ground pin on the Arduino.

2.2 TMP Sensor

The TMP36 temperature is fairly simple. It can measure degrees from -50°C to 250°C , where it has a precision of 0.1°C ¹. It works as a diode so when it changes temperature, it then its voltage changes accordingly. The sensor measures the small change and outputs an analog output based on it, so its possible to calculate the temperature according to that output. We use a `readCelcius()` method from the Temperature Class, and it can be seen below.

¹<https://www.bc-robotics.com/tutorials/using-a-tmp36-temperature-sensor-with-arduino/>

```

const int8_t TemperatureSensor::readCelsius()
{
    digitalWrite(triggerPin, HIGH);
    delayMicroseconds(200);
    uint16_t reading = analogRead(readPin);
    digitalWrite(triggerPin, LOW);
    // converting that reading to voltage, for 3.3v arduino use 3.3
    double voltage = reading * 5.0; // * 0.0009775171;
    voltage /= 1024.0;
    return round((voltage - 0.5) * 100);
}

```

`reading = analogRead(readPin)` reads the output from the sensor, then it's converted to voltage by multiplying with five. That is then divided by 1024 to find the percentage of the analog read, because the arduino outputs between 0 and 1023, where 0 is not voltage and 1023 is 5V. Five is then subtracted from that, and then that result is multiplied with 100 to convert it from mV to degrees. Five is subtracted from the reading, because we want it so that 0V corresponds to -50°C

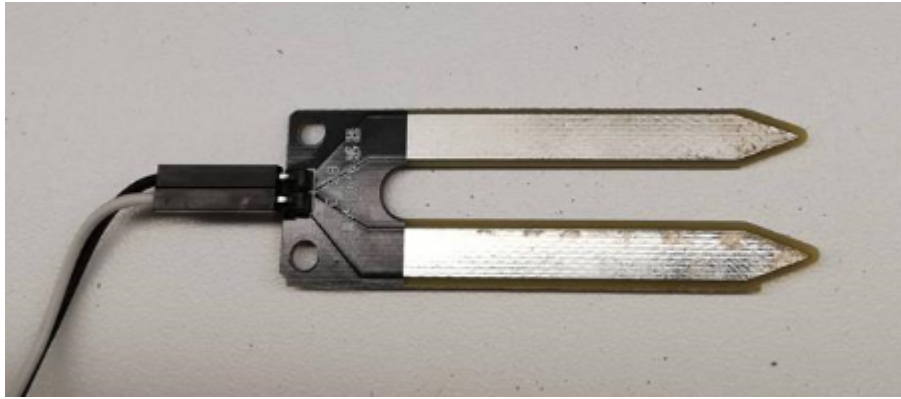


Figure 1: Corroded fc-28 after moderate use.

3 Attachments