Project Journal

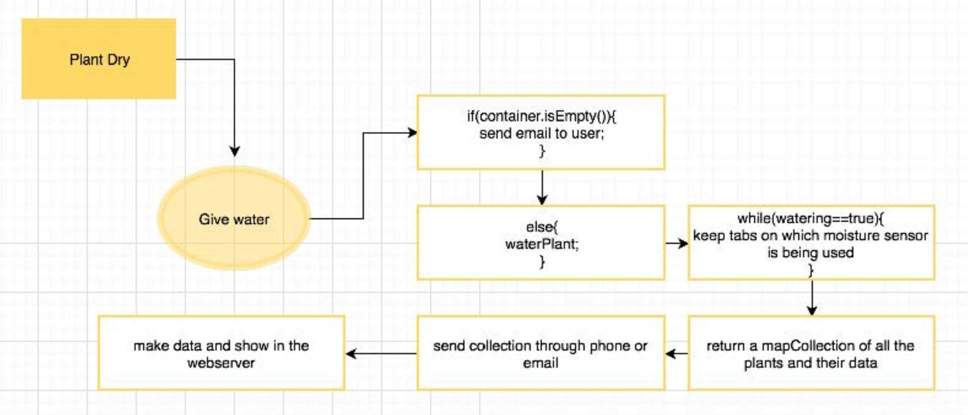
Joanna Rubi ||| 300333893

Chosen Project

SOIL20

Is a device that would water your plant when the soil is almost dry.
I will be using a moister sensor to achieve the purpose of this device, I thought it would be more efficient to use a moisture sensor than a solar panel (to check heat) because sometimes plants won't need water even though the heat is intense because the soil might've stored enough water from the last time it was watered. When the soil reaches a certain number on the moisture scale using a soil moisture sensor to measure the moisture, the device would shower the plants with water. I will then create an app that instructs the "gardener" to add water in the tank when the holder is almost empty – I will achieve this using an email or a text. It will also send data on the plants water patterns so that the gardener can have a regular pattern chart for that plant.

Initial Plan



Plant Dry

Give water

if(container.isEmpty()){
send email to user;
}

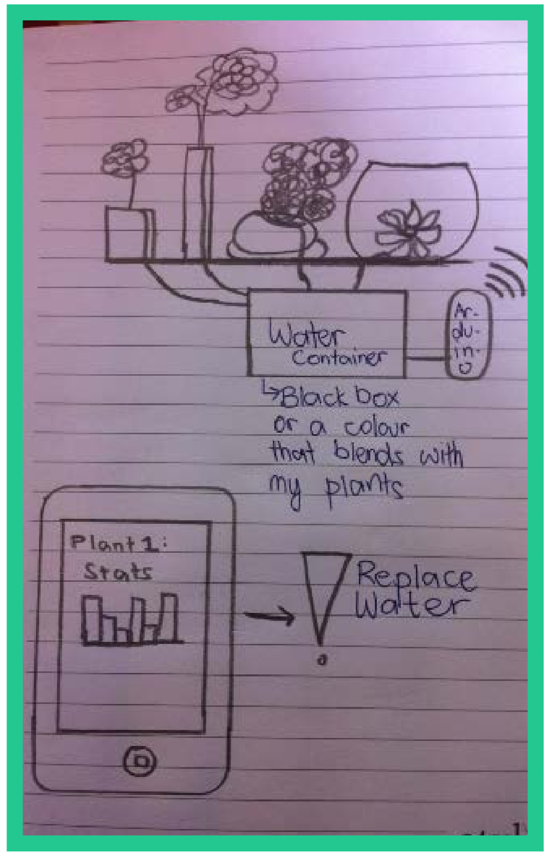
else{
waterPlant;
}

while(watering==true){
keep tabs on which moisture sensor is being used
}

return a mapCollection of all the plants and their data

send collection through phone or email

make data and show in the webserver



Theoretical outline of technical concepts used in your design:

- How to make my Arduino send notifications on my phone or via email

- Developing a webserver

- Making an HTML interface - for data and statistics

- A way for my Arduino to communicate with my webserver and send emails

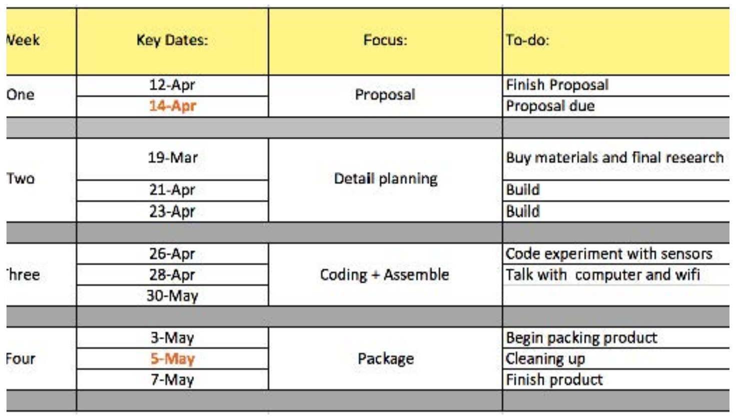
- How my Arduino would check water levels in my container

Developed Plan

When plant is dry give water
When the plant is watered send email
Email = to remind user to add more water to the container

Theoretical outline of technical concepts used in your design:
- An app to help Arduino with email
- I will not be using a webserver

Action Plan + Timeline



Veek	Key Dates:	Focus:	To-do:
One	12-Apr	Proposal	Finish Proposal
	14-Apr		Proposal due
Two	19-Mar	Detail planning	Buy materials and final research
	21-Apr		Build
	23-Apr		Build
hree	26-Apr	Coding + Assemble	Code experiment with sensors
	28-Apr		Talk with computer and wifi
	30-May		
Four	3-May	Package	Begin packing product
	5-May		Cleaning up
	7-May		Finish product

Development Stage

Is a device that would water your plant when the soil is almost dry.
I will be using a moister sensor to achieve the purpose of this device, I thought it would be more efficient to use a moisture sensor than a solar panel (to check heat) because sometimes plants won’t need water even though the heat is intense because the soil might’ve stored enough water from the last time it was watered. When the soil reaches a certain number on the moisture scale using a soil moisture sensor to measure the moisture, the device would shower the plants with water. I will then create an app that instructs the “gardener” to add water in the tank when the holder is almost empty – I will achieve this using an email or a text. It will also send data on the plants water patterns so that the gardener can have a regular pattern chart for that plant.

Planning

In my brainstorm and planning I decided to break down my project into smaller bits to make it easier for me to have a focus on what to do to complete my project.

Methods:

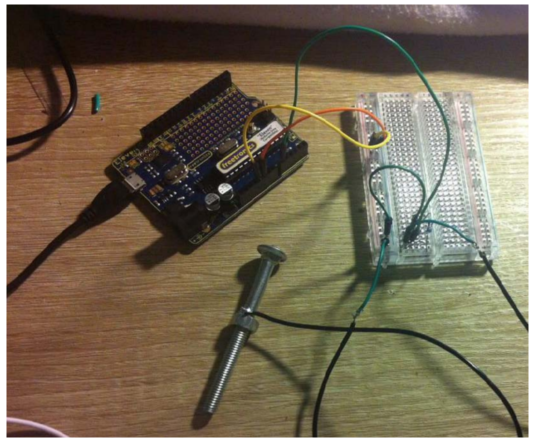
Create a moisture sensor.
The initial plan was to buy a moisture sensor but unfortunately there is no moisture sensor sold in Wellington, I’ve looked in JayCar electronics but they did not sell them so instead I will be creating my moisture sensor.

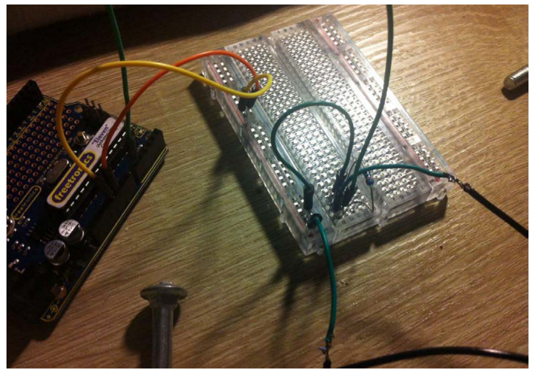
Make device communicate with Gmail.
Instead of buying an Ethernet arduino I decided to explore a cheaper way for my device to communicate with me. I will be doing thorough research to find what to use to make both entities communicate.

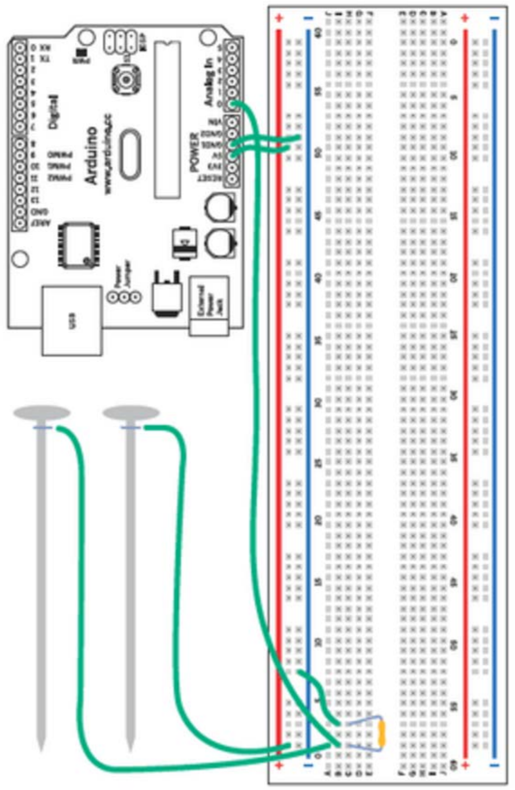
Pump water with Arduino.
When the moisture sensor is around 120 then I will instruct my arduino to give current to my pump so that the plant can get the water it needs.

Battery Powered device.
Research how to make arduino be powered by battery rather than being connected to the computer.

Code and Package.
Do final package by constructing and code together. Document the final stage of the project.







Materials:
2 Galvanised Nails
6 jump wires
Breadboard
Arduino

Moisture Sensor Stage One:

Purpose: By measuring the resistance between the 2 nails stuck in the dirt we can tell the moisture level of the soil. The more water there is in the soil, the more conductive it is. Another way of creating a moisture sensor is using a PCB but because of time constraints I will be using the nails, it is less reliable than using the PCB but it still does the work.

Circuit: As seen in the diagram below it shows how I have assembled my circuit
I wrapped the wires along the nails.
For my first nail (leftGreen wire) it is attached to 5V (positive), the second nail(rightGreen wire) is plugged in 29a connected to GND (negative) and also connected to 10k resistor, it is also connected to the analog input 0 on the arduino.

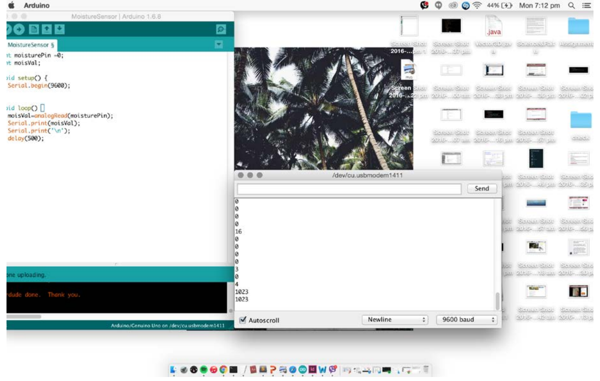
Code:

```
int moisturePin =0; //initialise which pin is being read
int moisVal;
void setup() {
  Serial.begin(9600);//print the values in this version}
void loop() {
  moisVal=analogRead(moisturePin);//read the value in this pin
  Serial.print(moisVal); //print the value that was read
  Serial.print(.\n’); //new line for easy reading
  delay(500); //slow infromation reading}
```

Trial:

Output: 0 if the nail is not touching anything
50-100 if the nail experiences human contact
900+ if both nails touch each other.

I did a test to see what the perfect range is for a wet soil so that if the number falls in that range then that means the motor can stop watering because the plant now has some water. My condition will be between 680 - 750 in respect to the plant size and will have a delay of 15 secs, I will be using this project on a small plant, the bigger the plant the bigger the watering time will be. I also noticed that a very dry plant returns 0 so before serial reads 0 I will give my arduino a condition that if my reading is below 150 or 120 then it is time to water the plant before it starts to wither.
To determine the length of the watering system I did several trials to determine how long my watering system should water the plant. I’ve concluded that a 15 sec delay is the right amount, making sure that the soil is thoroughly watered but not too much that the root will drown.



```
nt moisturePin =0;
nt moisVal;

oid setup() {
 Serial.begin(9600);

oid loop() {
 moisVal=analogRead(moisturePin);
 Serial.print(moisVal);
 Serial.print('\n');
 delay(500);
```

```
0
0
0
0
16
0
0
0
0
3
0
4
1023
1023
```

Reflection:

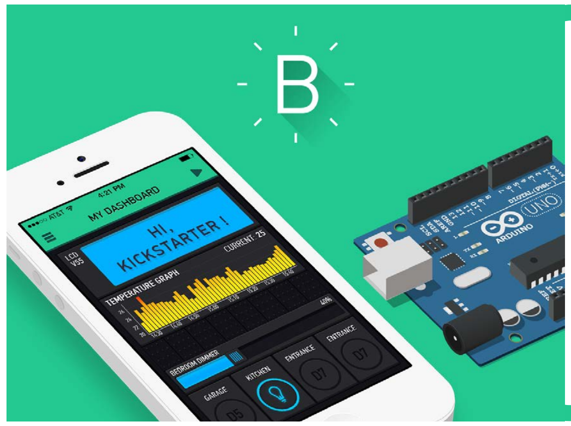
It was relatively easy to create a moisture sensor using the two nails, the coding part of it was really easy and again the biggest difficulty I experienced was understanding how to build the circuit. I need more practice with building circuits. I did a lot of research to help me make the moisture sensor and I found this website to be extremely helpful with helping me with the circuit building. I learnt about conductivity and how we can measure things with how conductive a metal is, the more electricity the more conductive the object is. I found it really interesting that moisture can be measured because of the conductivity of a nail.
Great resource for helping me build the circuit: http://forum.arduino.cc/index.php?topic=37975.0

Internet Connection Stage Two::

I was researching applications that would allow me to have an app and also control my arduino, I found several applications that was compatible with android but because I use IOS it was harder to find an application that was free, had a good interface and was easy to use. I found Blynk, an application that allows you to control your arduino using your phone. I will be using their widgets to get notified if my plant has been watered and a reminder to check the water supply to make sure there is still water to water my plants and to be careful not to damage my water pump. This way my responsibility for my plant is lowered to adding water to my water container.

Code:

```
#define BLYNK_PRINT Serial
#include <SPI.h>
#include <Ethernet.h>
#include <BlynkSimpleEthernet.h>

char auth[] = “62fc05b95b9a422f9eacfd75bd25ca3e”; //unique to each project
//auth = builds the relationship between the phone and the arduno using the wifi as the bridge.
void setup(){
  Serial.begin(9600);
  Blynk.begin(auth);
  while (Blynk.connect() == false) {
    // Wait until connected
  }
  Blynk.email(“rubijoanna@gmail.com”, “Soil20”, “Your plant was watered, she will live! Check water supply!
“);
  pinMode(2, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(2), emailOnButtonPress, CHANGE);
}

void emailOnButtonPress(){
  int isButtonPressed = !digitalRead(2); // Invert state, since button is “Active LOW”
  if (isButtonPressed){ // You can write any condition to trigger e-mail sending
    Serial.println(“Button is pressed.”); // This can be seen in the Serial Monitor
    Blynk.email(“rubijoanna@gmail.com”, “Subject: Button Logger”, “You just pushed the button...”);
  }
}
void loop(){
  Blynk.run();
}
```

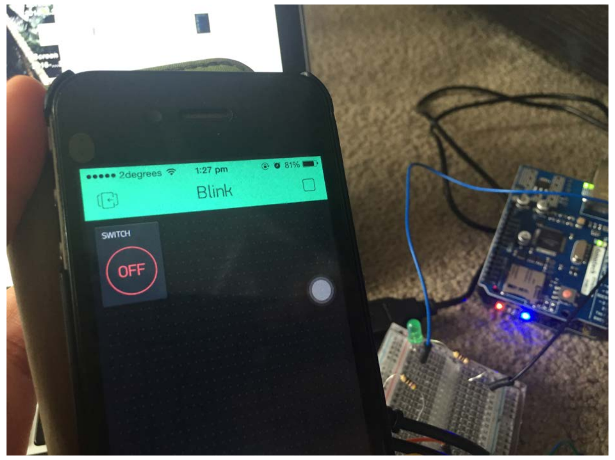


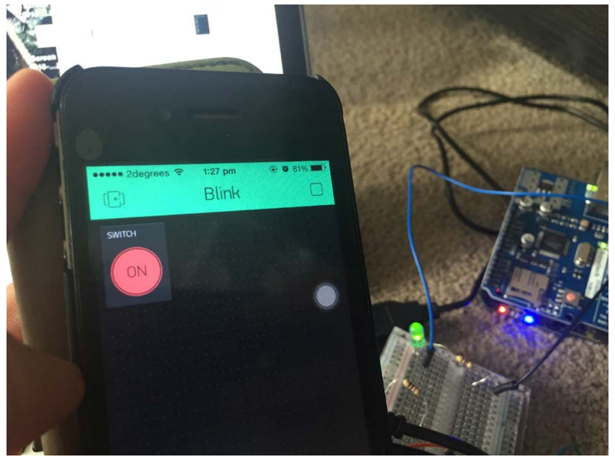
Blynk is a Platform with iOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet.

It’s a digital dashboard where you can build a graphic interface for your project by simply dragging and dropping widgets.

Blynk is not tied to some specific board or shield. Instead, it’s supporting hardware of your choice. Blynk will get you online and ready for the Internet Of Your Things.

Trial:

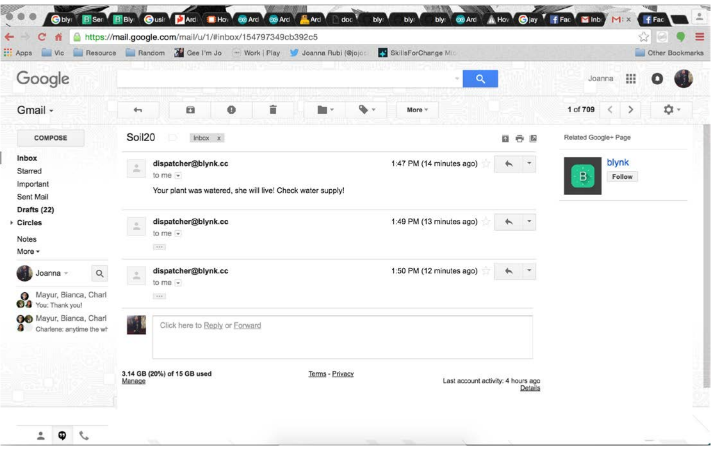




The photos show how I was able to control my LED using a button I added to my board. Blynk is a very powerful tool that can do a lot of really useful methods, I like be using their email and notifications method and incorporate it with my project. The photos show how I was able to control my LED using a button I added to my board. Blynk is a very powerful tool that can do a lot of really useful methods, I like be using their email and notifications method and incorporate it with my project. I do not need to create a circuit for Blynk to work I simply need to incorporate Blynk into my code so that it knows when to send a message and a phone notification. I’ve uploaded a video to support my experiment https://youtu.be/wNz4NYsijco

Implement:

Materials:
Enthernet Shield
Arduino
Cable
Mobile Phone
Blynk App
Wifi



Soil20 Inbox

dispatcher@blynk.cc 1:47 PM (14 minutes ago)
to me
Your plant was watered, she will live! Check water supply!

dispatcher@blynk.cc 1:49 PM (13 minutes ago)
to me

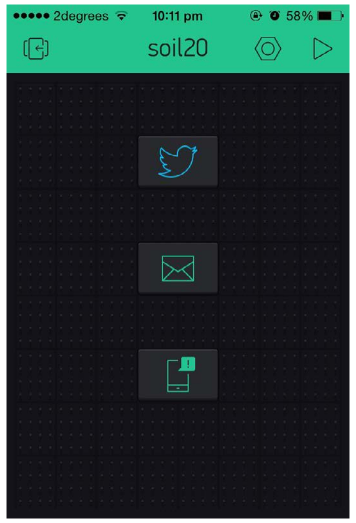
dispatcher@blynk.cc 1:50 PM (12 minutes ago)
to me

Click here to Reply or Forward

3.14 GB (20%) of 15 GB used
Manage

Terms - Privacy

Last account activity: 4 hours ago
Details



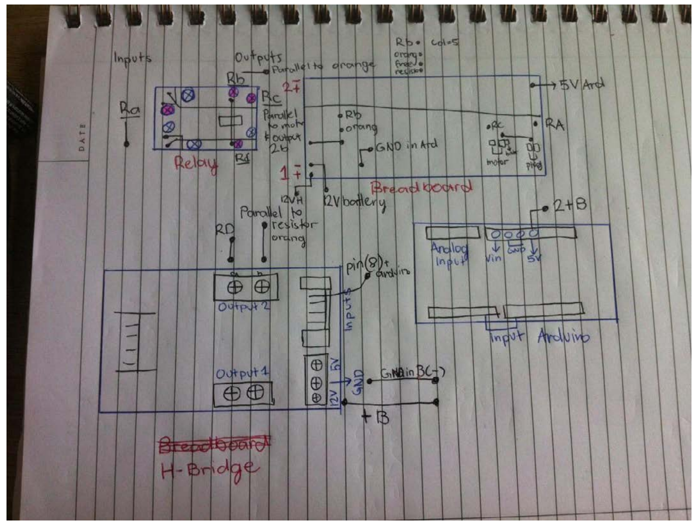
Using the code on the left I’ve successfully connected my arduino with my mobile I have achieved my goal of building making a way for my arduino to communicate with my phone and the Internet. The image on the right is the Blynk interface for my phone.

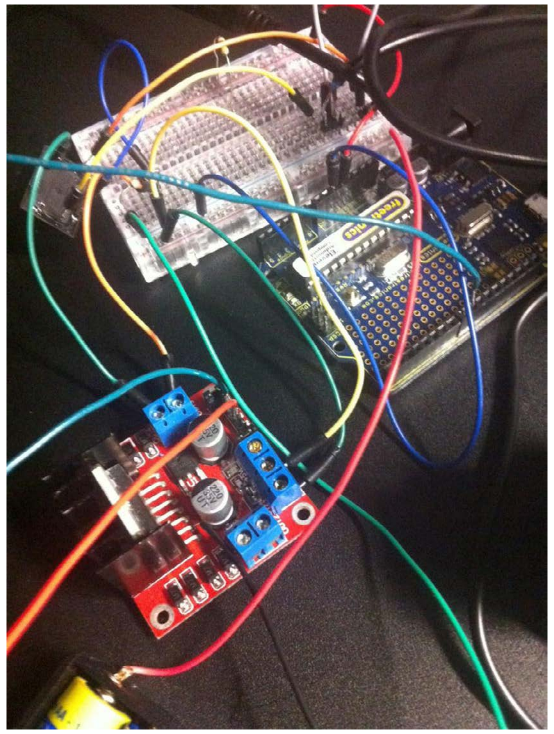
Reflection:

I thought this part of my project would be the hardest part. Initially my plan was to not invest on an Ethernet shield but because of some constraints like time and because I really want to implement a way for my plant to email and notify me I ended up buying an Ethernet shield. Investing on one is really rewarding because I made my project a better one, being emailed by my plants, controlling the LED lights I can already imagine the power I have by buying one. The Ethernet has enhanced my project and has made this project tackle and solve more problems with gardening, such as learning how many times your plants need water weekly. I am also very happy with how Blynk is making my project easier, without having to code how to make my arduino and Internet communicate, Blynk does all the heavy lifting for me.

Pump and Arduino Stage Three::

This part of the project was the hardest part for me, purely because I didn't know how to read circuit boards, reading the PCD lines and knowing what goes where when creating a circuit. I did a lot of research to help me understand how to create this because similar projects have already been made online but because my project is different than others it was hard to find a project that had th same materials and ideas as I had. I bought my own soldering iron, wire stripper so I could work at home and have more time to experiment but even with these at home I struggled to build a proper one. I reached out to my lecturer for help and finally had some assistance. Because of my lecturer who showed me how to build one, I took notes from his circuit and built my own circuit using his circuit as reference. I annotated his work (as seen below) to understand why he did what he did, with that in mind I was able to implement a similar circuit design to my lecturers.



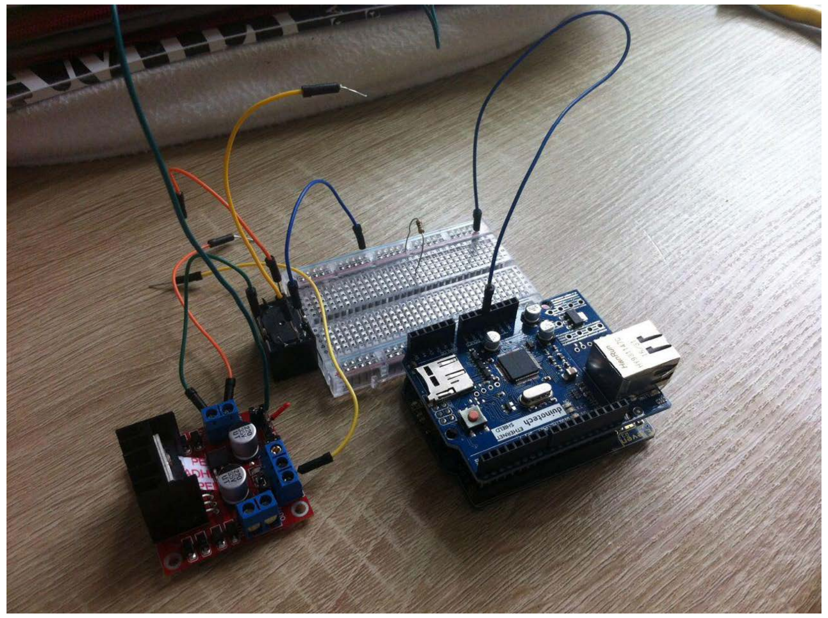


Materials:

The following materials used to make the motor function with my arduino are: H-Bridge, AC water pump, 12V battery, Breadboard, Arduino, 12V relay.

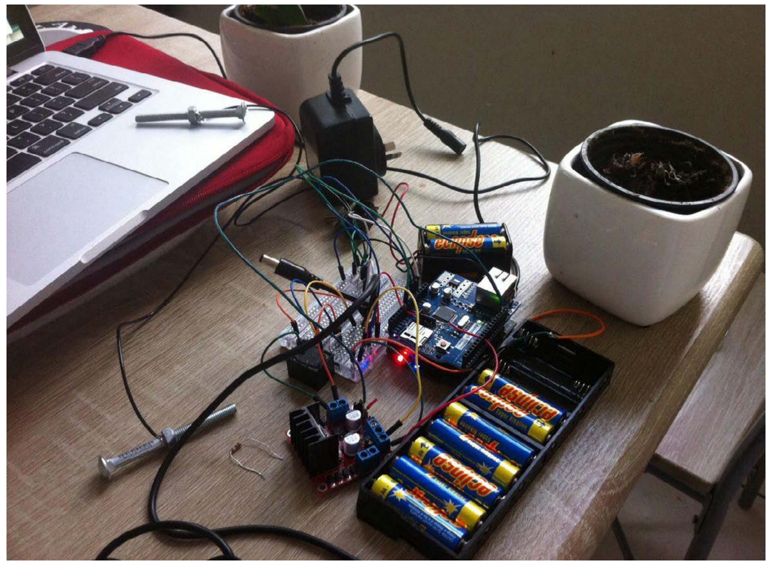
Code:

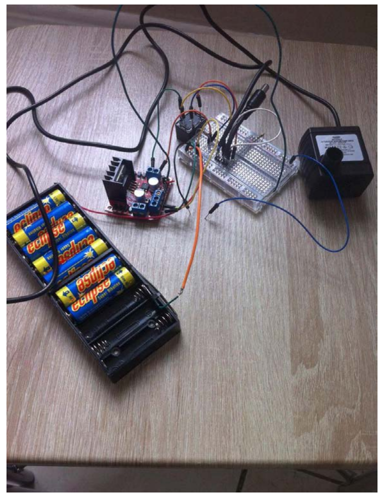
```
int waterPummp = 13;

void setup() {
  pinMode(waterPummp, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);
  delay(15000);
  digitalWrite(13, LOW);
  delay(50000);
}
```

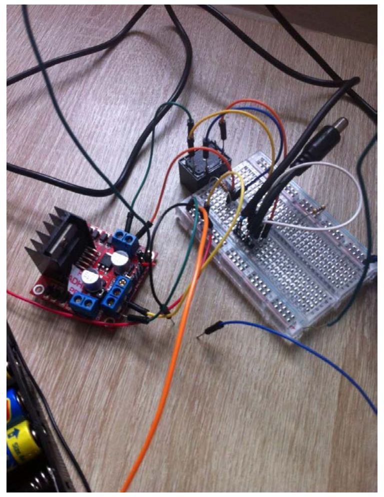


Trial:

While I was trying to test my circuit I noticed that there was smoke coming out from my breadboard and I immediately shut down any current flowing through my arduino, motor and everything. I was very scared that I may have short-circuited my board or did something even worst, but luckily because I noticed the smoke and the smell I immediately turned everything off. It was quiet difficult to create a new design because I needed to really understand why I'm doing what I'm doing, debug problems and figure out why smoke was coming, and why it wasn't working. After a couple of hours of research and trying to understand the diagram I drew from my lecturers design I eventually found a way of making my water pump work. It felt pretty rewarding building it because I was able to gain more insight in circuit design and also understanding more electricity jargon.







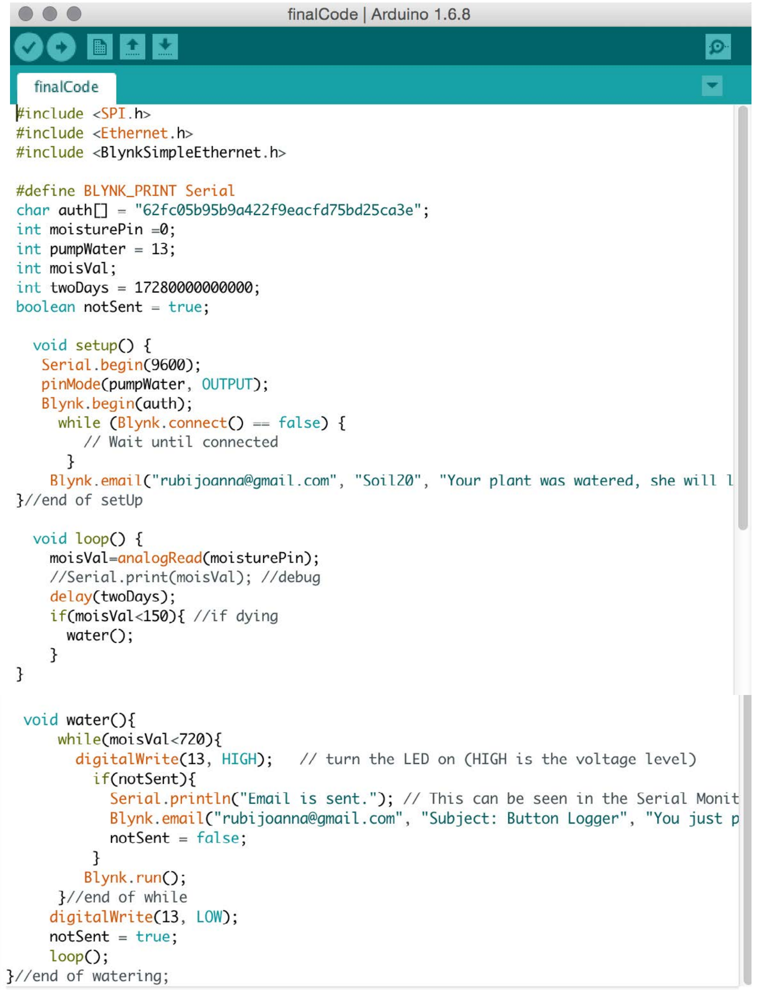
Reflection:

Making the battery circuit was relatively easy I only had to buy the required components to make it portable. I didn't encounter any problems while making this part of the project, I thought that making my device portable would be hard but I'm very happy that it is relatively simple. I've noticed that my budget for this project is quite big compared to my previous projects, which had a total sum of under $20. Even though the scale of my project is relatively the same as the previous one I think more components are needed this time for the device to work. The development of my project is going steadily I have given myself enough time to accomplish what needs to be done I also think the my strategy of breaking down this project to smaller ones has made me productive, goal-oriented and time efficient, I think I will be able to accomplish my goal because of this strategy.

Code Packaging Stage Four::

Code:

```
#include <SPI.h>
#include <Ethernet.h>
#include <BlynkSimpleEthernet.h>

#define BLYNK_PRINT Serial
char auth[] = “62fc05b95b9a422f9eacfd75bd25ca3e”;
int moisturePin =0;
int pumpWater = 13;
int moisVal;
int twoDays = 17280000000000;
boolean notSent = true;

  void setup() {
   Serial.begin(9600);
   pinMode(pumpWater, OUTPUT);
   Blynk.begin(auth);
     while (Blynk.connect() == false) {
        // Wait until connected
       }
    Blynk.email(“rubijoanna@gmail.com”, “Soil20”, “Your plant was watered, she will live! Check water
supply! “);
}//end of setUp

  void loop() {
    moisVal=analogRead(moisturePin);
    //Serial.print(moisVal); //debug
    delay(twoDays);
    if(moisVal<150){ //if dying
      water();
    }
}

  void water(){
      while(moisVal<720){
        digitalWrite(13, HIGH);    // turn the LED on (HIGH is the voltage level)
          if(notSent){
            Serial.println(“Email is sent.”); // This can be seen in the Serial Monitor
            Blynk.email(“rubijoanna@gmail.com”, “Subject: Button Logger”, “You just pushed the button...”);
            notSent = false;
          }
        Blynk.run();
      }//end of while
     digitalWrite(13, LOW);
     notSent = true;
     loop();
}//end of watering;
```



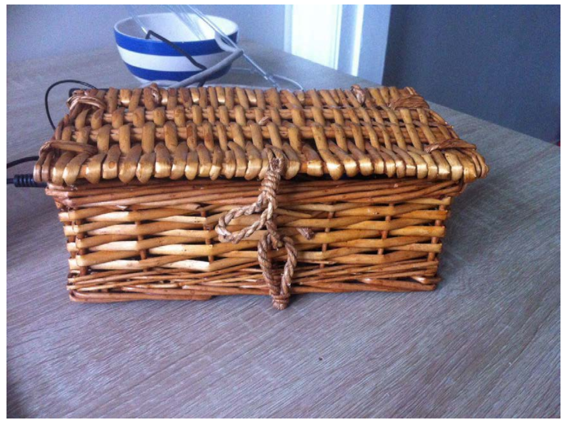
```
#include <SPI.h>
#include <Ethernet.h>
#include <BlynkSimpleEthernet.h>

#define BLYNK_PRINT Serial
char auth[] = "62fc05b95b9a422f9eacfd75bd25ca3e";
int moisturePin =0;
int pumpWater = 13;
int moisVal;
int twoDays = 17280000000000;
boolean notSent = true;

  void setup() {
   Serial.begin(9600);
   pinMode(pumpWater, OUTPUT);
   Blynk.begin(auth);
    while (Blynk.connect() == false) {
       // Wait until connected
     }
    Blynk.email("rubijoanna@gmail.com", "Soil20", "Your plant was watered, she will l
}//end of setUp

  void loop() {
    moisVal=analogRead(moisturePin);
    //Serial.print(moisVal); //debug
    delay(twoDays);
    if(moisVal<150){ //if dying
      water();
    }
}

 void water(){
     while(moisVal<720){
       digitalWrite(13, HIGH);    // turn the LED on (HIGH is the voltage level)
         if(notSent){
           Serial.println("Email is sent."); // This can be seen in the Serial Monit
           Blynk.email("rubijoanna@gmail.com", "Subject: Button Logger", "You just p
           notSent = false;
         }
        Blynk.run();
     }//end of while
    digitalWrite(13, LOW);
    notSent = true;
    loop();
}//end of watering;
```

Reflection:

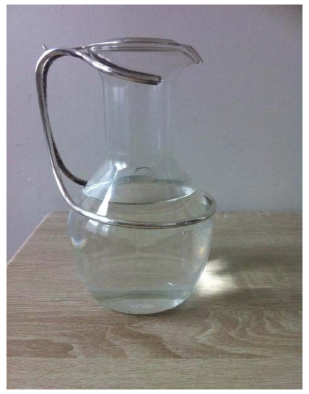
It was quiet easy to create the packaging code but I had a little problem testing its accuracy with my plants because I can't exactly create the desired situation unless it is actually happening. Logically I believe the code works, it compiles and no error is being thrown at me and I hope that when this project is actually put to use that the code works and no bugs or loopholes will come up. I thought I would have a hard time trying to implement the gmail part to my code but it was relatively straightforward because the programmer did very useful commenting that I understood and most of the hard work is done y the library code of Blynk.

I made the right decisions by spitting my project into small bits because it allowed me to focus on one and it was relative easy to compile all the codes into one big code. By making the smaller ones first I ensure that the codes for each component was working so making and integrating them together became easy because I only needed to initialise a couple of codes and conditionals to make the whole thing work together. I also really got to understand what each of the little component does and why it is vital for this project to work “all the little thing counts”.

Material Packaging Stage Five:



Box = I bought a flaxy, bamboo like box purely because I want to achieve a really natural look for my product. It only enhances the purpose because it encourages a very natural design since the theme I am going for this device is nature. Making my device be contained on a box that looks natural and have a forest-like theme it makes everything really blend together. I also think brown really blends with green as green represents the forest while brown represents the mud, the earth and the soil. I wouldn't have achieved this kind of theme if I 3D printed my own case so buying this box was a really good decision.



Water Container = I chose a very alchemy shaped glass container because the whole device is pretty technology reliant despite the design of having a very natural look, it almost contrast. Because of this container it symbolises how both nature and technology can blend together and create something beneficial for nature instead of the stereotype the technology ruins earth. The container also adds a modern taste to the design; it's not just a normal container but it actually looks really good with the whole product and has a special meaning to why I picked this design.

Final Packaging Stage Six::

Video:

For my video I want my first part to show the design of the product so people can see how nice the device can look into their homes. My design doesn't even look like a device because the container used can really blend with other things in the house, it will good on the table or in a cabinet.

The second part of my video I will show how my device works and what it's about so shots of the soil sensor, water, water being pumped so that the audience have an idea of what my device does.

The last part of my video will be a shot of my product name and a little catch phrase for my product, which I decided to be "your plant's drink".

Photos:

For my photos my plan is to basically just take photos of my product in an environment that enhances its design, I will be adding a lot of plants around my device so it looks more like a product you can use inside the house and show ideas to others how you can incorporate my device into your home. The main colours I will be using for my product are green and brown.

Music License:

Song: Drake - Fire & Desire (INSTRUMENTAL) [Prod. Jed Official]

TERMS & CONDITIONS:

BY USING/EXTRACTING/SHARING MY CONTENT, YOU AGREE, ACKNOWLEDGE THIS INFORMATION STATED IS CORRECT, AND ARE BOUND TO THESE TERMS & CONDITIONS IF YOU WISH TO USE ANY OF MY CONTENT ---- UNDERSTAND THIS INSTRUMENTAL IS COPYRIGHTED BY JED OFFICIAL AND NO ONE ELSE. IF YOU ARE USING THIS INSTRUMENTAL OR ANY OF MY CONTENT FOR AND ONLY FOR COVERS, REMIXES, MASHUPS, PARODY VIDEOS, AND BACKGROUND MUSIC, THEN PLEASE PROVIDE CREDIT TO (Jed Official) WITH MY CHANNEL NAME AND A LINK TO THIS VIDEO IN EITHER THE TITLE OR DESCRIPTION OF YOUR VIDEO. YOUR USAGE OF MY CONTENT SHALL BE AND ONLY BE ON YOUTUBE FOR NON-PROFIT PURPOSES (COVERS AND BACKGROUND MUSIC ARE PERMITTED FOR MONETIZATION). HOWEVER KARAOKE VERSIONS ARE NOT ALLOWED AND YOU SHALL NOT RE-UPLOAD ANY OF MY VIDEOS/JED'S ORIGINAL CONTENT. CLAIMING "FAIR USE" OR A "DISCLAIMER" WITHOUT SUFFICIENT AND VALID PROOF IS NOT AN EXCEPTION AND WILL NOT BE CONSIDERED. IF I CATCH YOU VIOLATING ANY OF THESE CONDITIONS WITHIN YOUTUBE, YOUR VIDEO WILL BE TAKEN DOWN AND YOU WILL RECEIVE A STRIKE ON YOUR ACCOUNT! IF YOU DO RECEIVE A STRIKE, I WILL NOT BE HELD RESPONSIBLE FOR ANY LOSS/DAMAGES AND WILL NOT NEGOTIATE TO REMOVE/RETRACT IT. IF I CATCH YOU VIOLATING THESE TERMS OUTSIDE YOUTUBE, THEN I MAY HAVE YOUR CONTENT REMOVED AND/OR YOU MAY FACE SEVERE LEGAL REPERCUSSIONS. UNLESS YOU AGREE TO FOLLOW THESE TERMS, YOU ARE USING MY CONTENT AT YOUR OWN RISK. TERMS & CONDITIONS ARE SUBJECTED TO CHANGE AT ANYTIME WITH/WITHOUT NOTICE. --- OTHER THAN THAT ENJOY :)

Reflection:

I'm really happy with how my video turned out. I edited my photos and increased the contrast by 50-55 because the camera I used didn't capture the full colour of my plant and the props. For my video song I used an instrumental that was slow but had a good base because I didn't want my song to bore people but I didn't want it to have strong beats that it loses some of its elegance. Overall I'm pretty happy with the turn out for my video and photos. Video Link: https://www.youtube.com/watch?v=lkTHYnr9V-M

SOIL2O