Group 8:

LetitGrow! : Autonomous Hydroponic Garden

Leandro Alepuz (**CpE**)

- Web Server Implementation
- System Assembly

Danny Nguyen (**EE**)

- PCB Design
- Hardware troubleshooting

Edwin Rivera (**EE**)

- Parts and Budget
- PCB Manufacturing

Nathan To (**CpE**)

- Software Implementation
- Sensor Troubleshooting

UCF UNIVERSITY OF CENTRAL FLORIDA

Our Solution: **LetitGrow!** A smart hydroponic system

Leandro Alepuz

- ➢ **Deep Water Culture Hydroponic**

- ➢ **Automatic Plant Care**

- ➢ **Web App Integrated**

UNIVERSITY OF CENTRAL FLORIDA

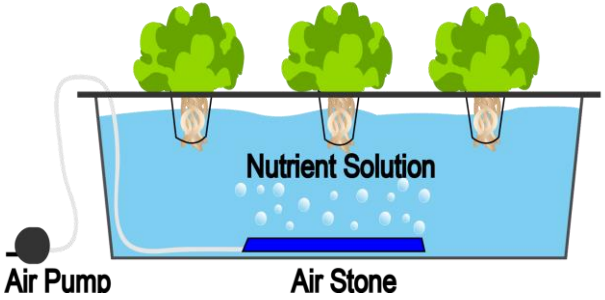
Why Deep Water Culture (DWC) Hydroponics?

Leandro Alepuz

➢ **What is it?**

Hydroponics is an agricultural technique that only uses water to grow plants, completely getting rid of soil.

DWC is a specific type of hydroponics where the roots are directly submerged in a water reservoir with oxygen and nutrients.



➢ **Advantages**

- Fast growth
- Simple setup
- Clean

UNIVERSITY OF CENTRAL FLORIDA

Motivations

Nathan To

- **Food Insecurity**
  - Rising temperatures due to climate change
  - Population outgrowing agricultural output

- **COVID-19**
  - Supply chain issues
  - Scarcity due to panic buyers
  - People acquiring new hobbies

- **Rising popularity of Hydroponics**
  - Uses 70% to 90% less water
  - Faster growing speed
  - Cleaner experience for indoors

UCF UNIVERSITY OF CENTRAL FLORIDA

Current Solutions

Nathan To

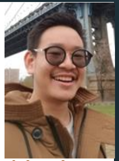
- **Home Hydroponics**
  - For beginners
  - Small capacity
  - Automatic lights
  - Small water pump to oxygenate water
  - Cheap ($50-$80)

- **Hybriponics**
  - Vertical garden
  - Large capacity
  - Costly ($800+)
  - App connectivity

- **Farm.bot**
  - Robotic arm
  - Multiple arm tips for functionalities
  - Soil based garden
  - Scalable
  - Expert level install
  - Expensive ($1700)

UCF UNIVERSITY OF CENTRAL FLORIDA

Goals and Objectives



Nathan To

- **Garden Autonomy**
  - Achieve a minimal amount of time needed for plant care from the user: 2 weeks without any care needed
- **Beginner user friendly experience**
  - Require no plant maintenance experience
- **Total feedback of the plant environment**
  - Use the plant’s pH, nutrients, water level, air and water temperature data to achieve optimal plant growth
- **Monitor remotely from the web**
  - Receive data from your garden

UCF UNIVERSITY OF CENTRAL FLORIDA

Requirements and Specifications

Leandro Alepuz

Requirement	Specification
Sensor Reading Frequency	30 minutes
Power self-sufficiency	Direct connection to US outlet
Water capacity for 12 plants	22 gallons
Sensor monitoring	Minimum 12 GPIO pins
Affordable cost	< $600
Accessible User Interface	Web browser monitoring

Plant Requirements

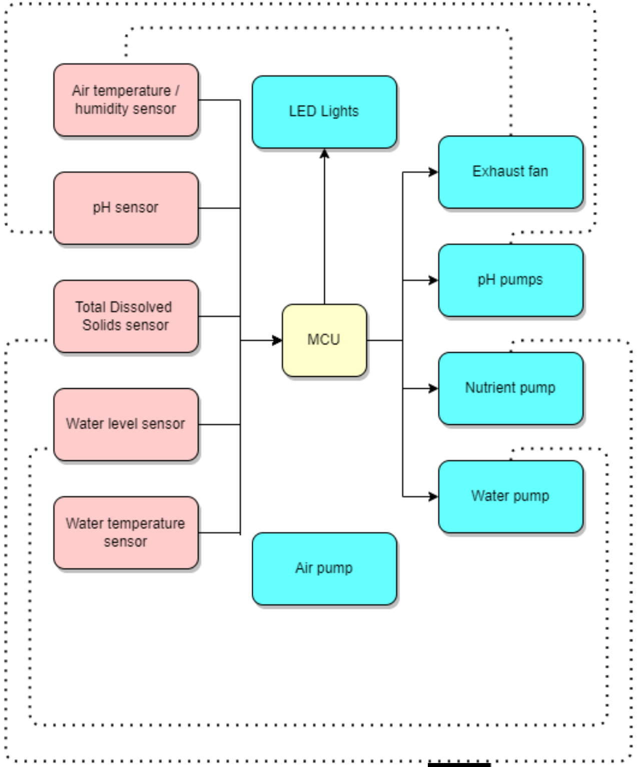
Leandro Alepuz

Plant Parameter	Specification
Water Temperature	[65-80]° F
Nutrients : Total Dissolved Solids (TDS)	[600-1000] ppm
Water pH level	[5.0 , 7.0]
Air Temperature	[60-90]° F
Humidity	[50-70]%
Light	LED Lights | 12-16 hours| Everyday

System Design Diagram

Leandro Alepuz

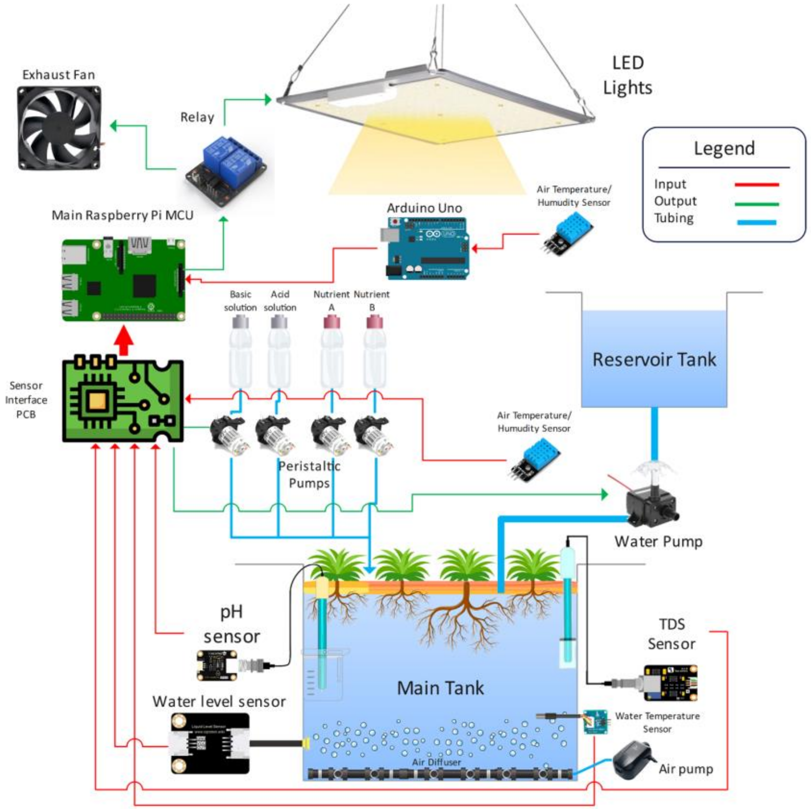
- ➢ **Inputs (red)**
  - Turn output components On/Off
  - Data is collected every 30 minutes under normal levels
  - If system needs calibration, data is actively sampled in a feedback loop
  - Water temperature sensor is used to calculate TDS levels.
- ➢ **Outputs (cyan)**
  - Dependent on the input sensors
  - LED lights directly controlled by MCU by regular scheduling
  - Air pump is completely independent as is always On



UNIVERSITY OF CENTRAL FLORIDA

Final Design

Leandro Alepuz



- **Inputs (red)**
  - Sensor data is received and transmitted to the main MCU
  - Humidity Sensor constantly monitors
- **Output (green)**
  - 4 peristaltic pumps supply pH and Nutrients
  - Lights are automatically turned on/off
  - Exhaust fan helps regulate humidity
  - Water pump refills the main tank
- **Supply Lines (blue)**
  - Tubing connects to the main tank for pH and Nutrient delivery
  - Air pump delivers oxygen to the roots via air diffuser placed in the bottom of the tank
  - Tubes deliver water from the reservoir tank to the main tank

UCF UNIVERSITY OF CENTRAL FLORIDA

Microcontroller

Edwin Rivera

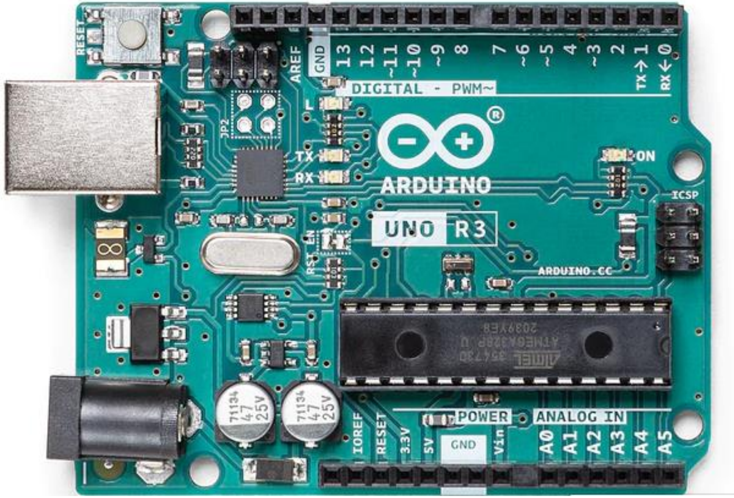
Raspberry Pi 4	
Processor:	Broadcom BCM2711 Quad core Cortex-A72 1.5GHz
Memory:	8GB LPDDR4 SDRAM
Connectivity:	X4 USBs, Ethernet, Bluetooth, x2 micro HDMI, 40 pin GPIO header
Power:	5V DC via USB-C & 5V DC via GPIO header (500mA)

UCF UNIVERSITY OF CENTRAL FLORIDA

Arduino Uno

Edwin Rivera

Arduino Uno Rev3	
Processor:	Atmega328p
Memory:	32KB Flash & 2KB SRAM
Connectivity:	USB, 14 digital I/O pins, and 6 analog inputs
Power:	5V DC operating voltage and 50mA DC current



Relay Module

Edwin Rivera

4-Channel Relay Interface	
Manufacturer:	SunFounder
Driver Current:	15-20mA (Each Channel)
Max Current:	10A
Max Voltage:	250V DC ; 125V AC
Coil Voltage	5V

pH Sensor

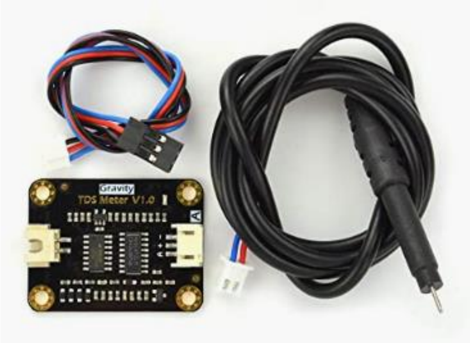
Edwin Rivera

Manufacturer:	GAOHOU
Cost:	$35.88
Measurement Range:	0pH - 14pH
Accuracy:	+/- 0.25pH
Operating Temperature:	0°C - 60°C
Size:	42 x 32 x 20mm

TDS Sensor

Edwin Rivera

Manufacturer:	Gravity
Cost:	$16.80
Measurement Range:	0ppm - 1000ppm
Accuracy:	+/- 10%
Working Voltage:	3.3~5.5V
Size:	42 x 32mm



- ❖ Challenges with calibrations

Water Level Sensor

Edwin Rivera

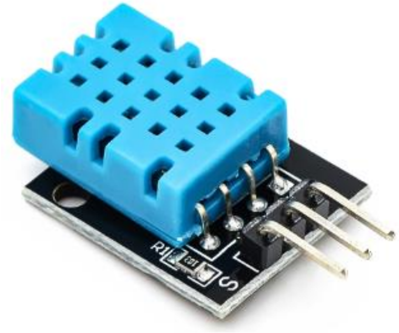
Manufacturer:	CQRobot
Cost:	$8.99
Working Voltage:	5V
Operating Temperature:	-25°C to 105°C
Size:	35 x 36 x 3mm

UNIVERSITY OF CENTRAL FLORIDA

Air Temp/Humidity Sensor

Edwin Rivera

Manufacturer:	Elegoo (DHT11)
Cost:	Owned
Temperature Range:	0°C to 50°C (+/-2°)
Humidity Range:	20% - 90% RH (+/-5%)
Size:	28 x 12 x 7.2mm



- Connection to Arduino was required

UCF UNIVERSITY OF CENTRAL FLORIDA

Water Temperature Sensor

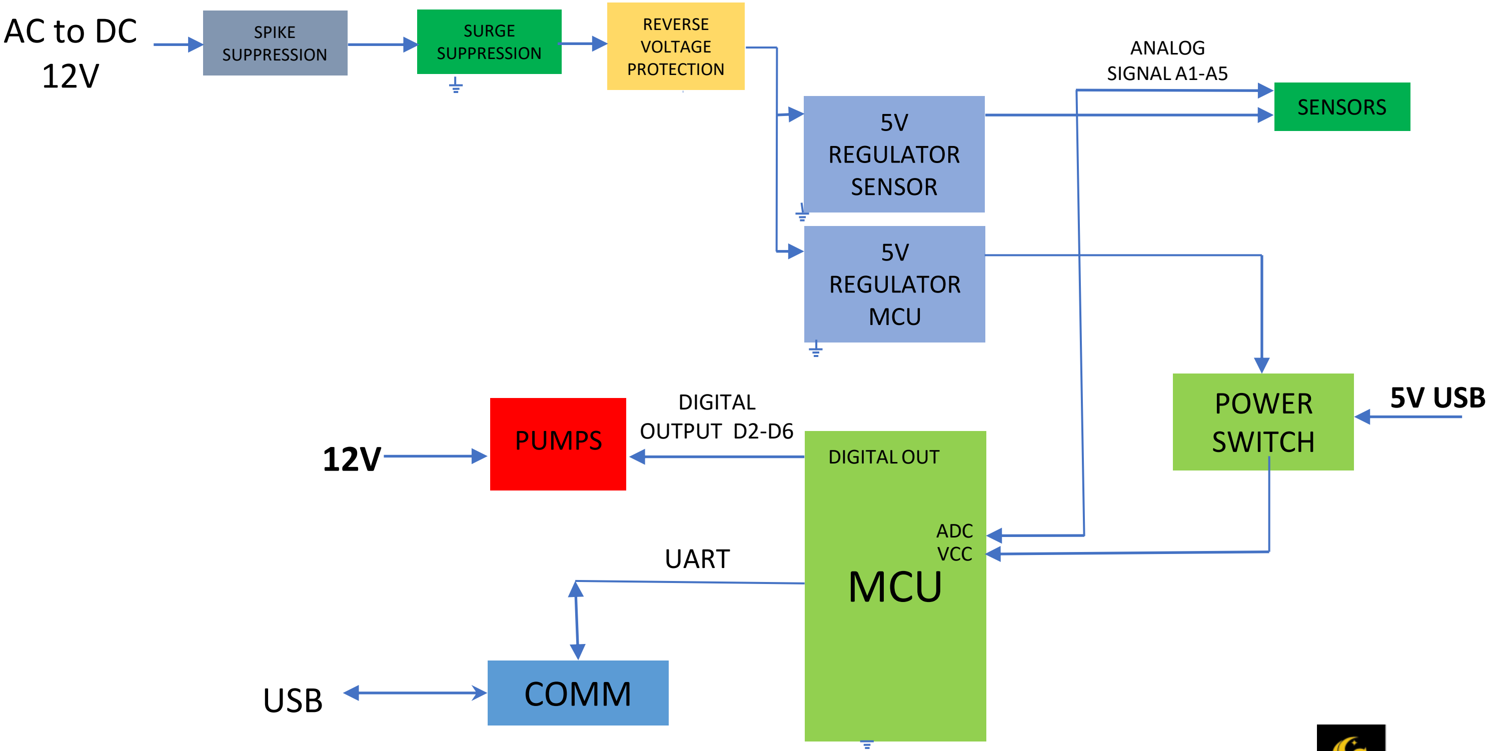
Edwin Rivera

Manufacturer:	GAOHOU
Cost:	$13.99 (x2 Sensors)
Measuring Range:	-55°C to 110°C (+/-2°)
Working Voltage:	3.2 ~ 5.25V
Size:	7 x 26mm

UCF UNIVERSITY OF CENTRAL FLORIDA

Power Block Diagram

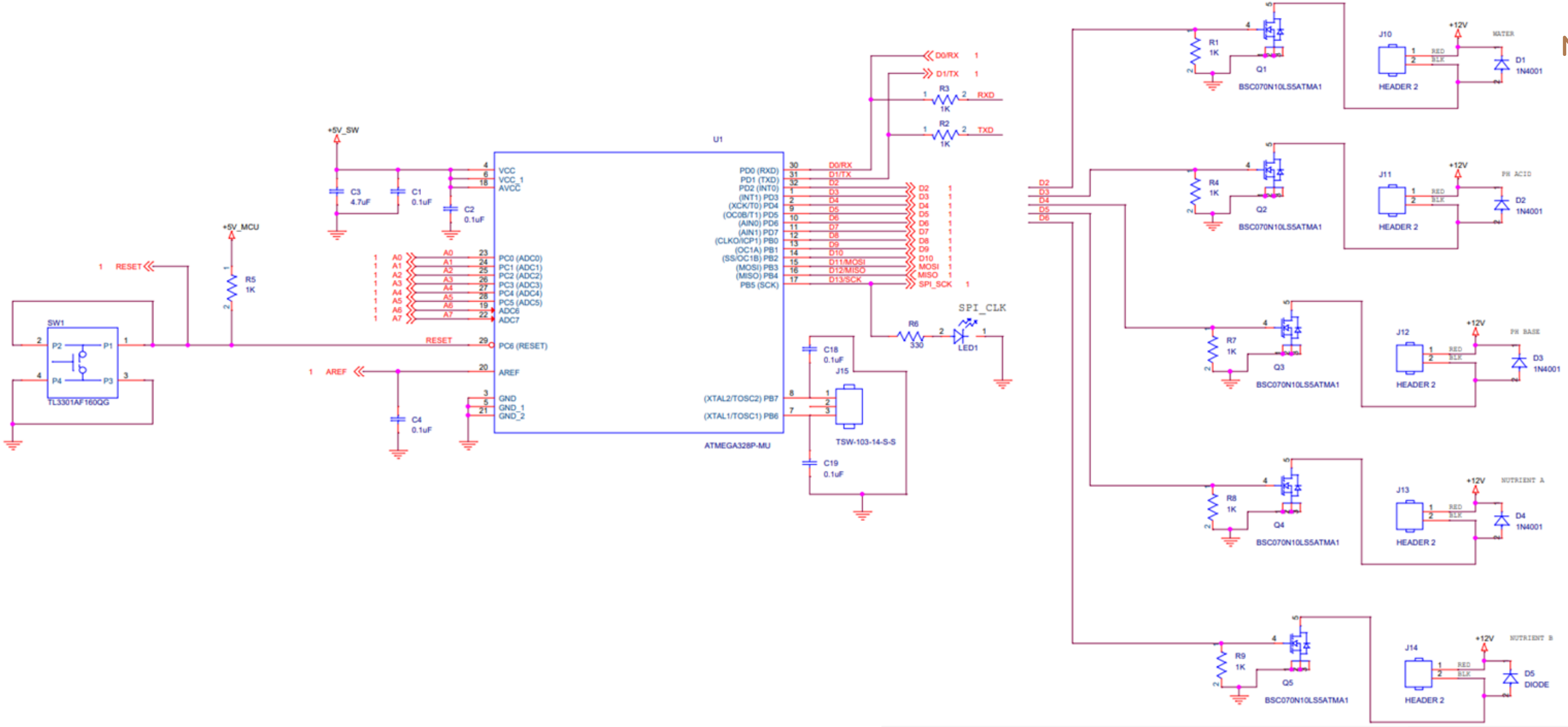
Danny Nguyen



UNIVERSITY OF CENTRAL FLORIDA

Schematics

Danny Nguyen



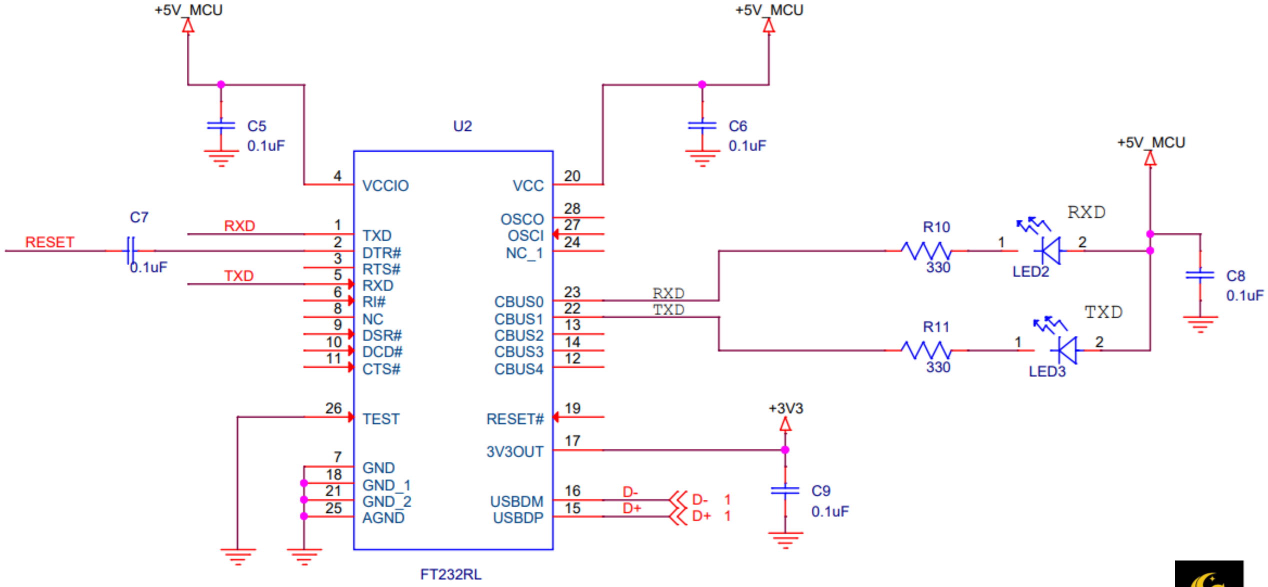
UCF

UNIVERSITY OF CENTRAL FLORIDA

Schematics

Danny Nguyen

FT232RL/USB-UART

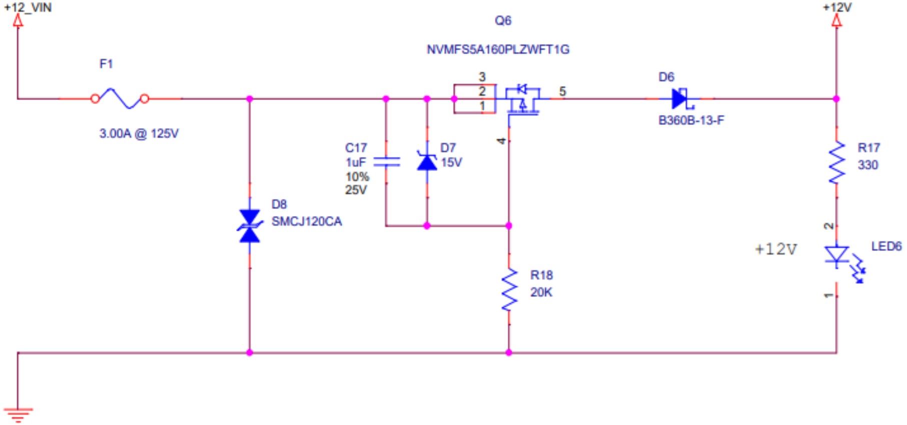


UNIVERSITY OF CENTRAL FLORIDA

Schematics

Danny Nguyen

+12_VIN PROTECTION



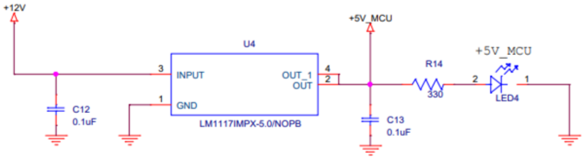
UCF UNIVERSITY OF CENTRAL FLORIDA

Schematics

Danny Nguyen

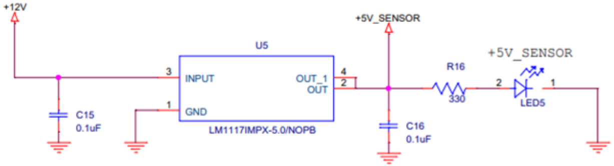
+5V LINEAR REG

SENSOR

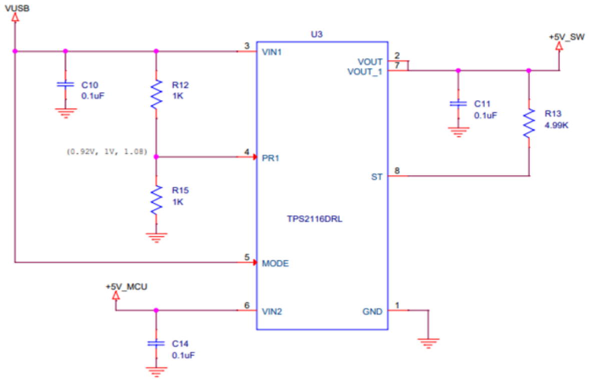


+5V LINEAR REG

MCU



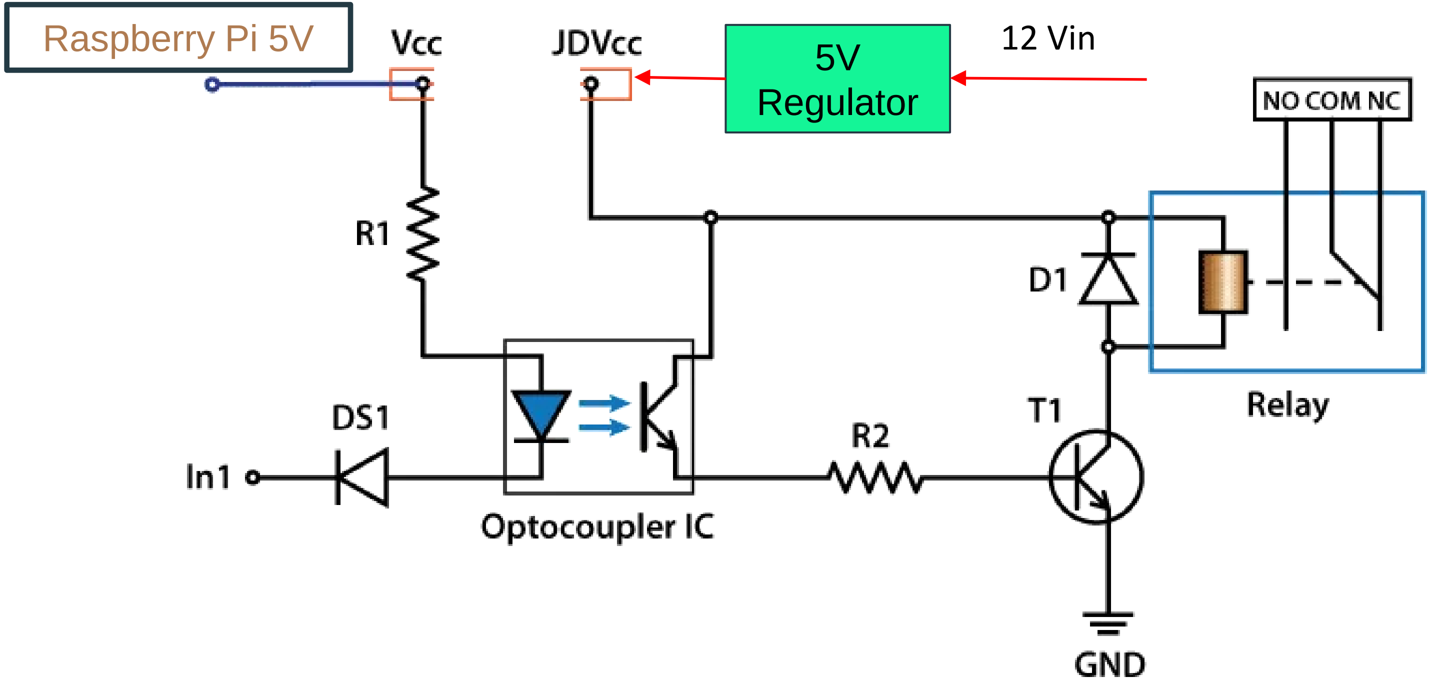
POWER SWITCH



UCF UNIVERSITY OF CENTRAL FLORIDA

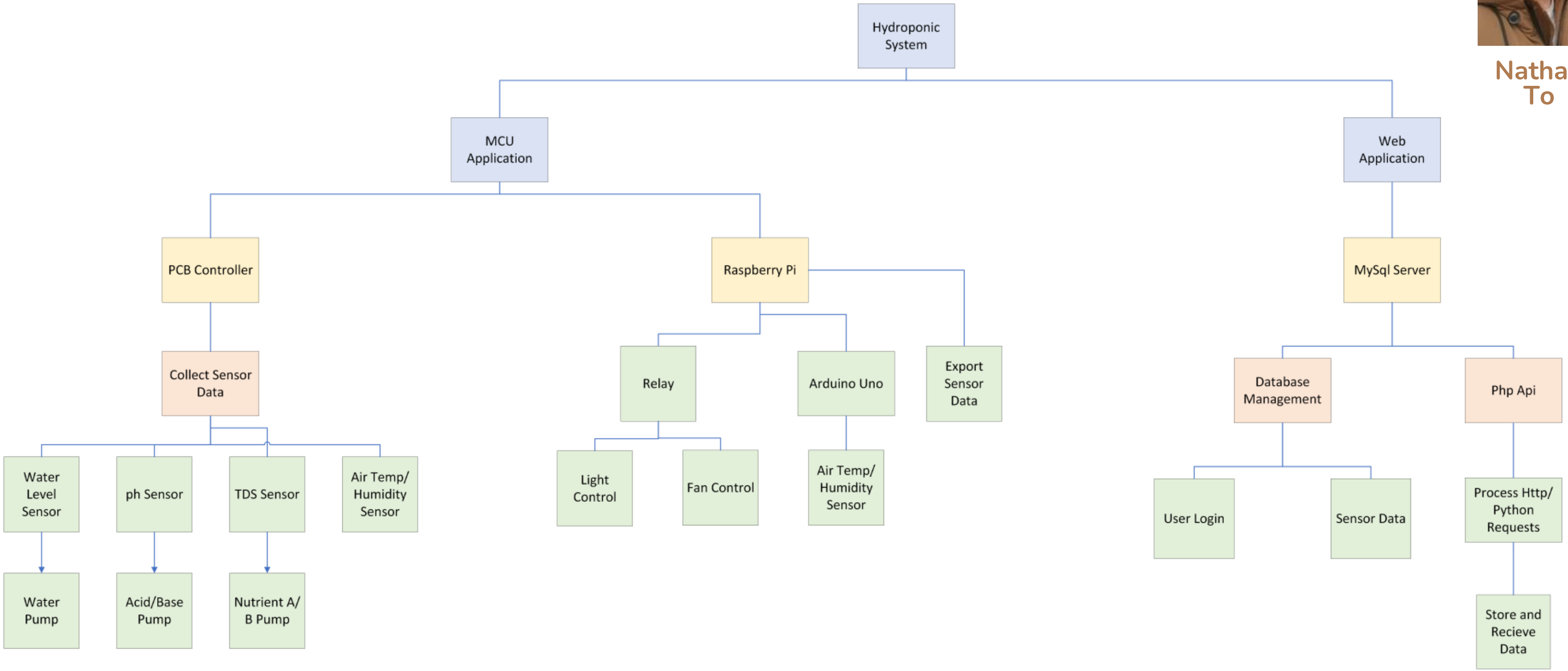
Relay Schematic

Danny Nguyen



UNIVERSITY OF CENTRAL FLORIDA

Software System Overview



Nathan To

UNIVERSITY OF CENTRAL FLORIDA

Defining Libraries, Pins, and Thresholds

```
// include sort library
#include <KickSort.h>

// include air/hum sensor library and initialize variables
#include "DHT.h"
#define DHTPIN A5
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

// include water temperature sensor library and initialize variables
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS A4
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

// initailize water level variables
int waterLevelPin=A3;// define pin A3

// A2 is A0
// must download the zip file for the library and import it from
https://wiki.dfrobot.com/Gravity__Analog_TDS_Sensor___Meter_For_Arduino_SKU__SEN0244
#include <EEPROM.h>
#include "GravityTDS.h"
#define TdsSensorPin A0
GravityTDS gravityTds;
// define tds range
float tdsLow = 600;
```

```
// define pH variables and calibration value
#include <Arduino.h>
const float m = -6.81818182;
int phPin = A1;
// define ph range
float phLow = 5;
float phHigh = 9;

// define pins for pumps
int waterPumpPin = 2;
int phAcidPumpPin = 3;
int phBasePumpPin = 4;
int nutrientAPumpPin = 5;
int nutrientBPumpPin = 6;

// define temporary pump delay
int waterPumpDelay = 5000;
int phPumpDelay = 2000;
int nutrientPumpDelay = 2000;
```

Nathan To

UNIVERSITY OF CENTRAL FLORIDA

Arduino Setup

Nathan To

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  // initialize the air humidity sensor
  dht.begin();

  // initialize water temperature sensor
  sensors.begin();

  // set A3 pin as “input” for water level sensor
  pinMode(waterLevelPin,INPUT);

  // initialize tds sensor
  gravityTds.setPin(TdsSensorPin);
  gravityTds.setAref(5.0);  //reference voltage on ADC,
default 5.0V on Arduino
  gravityTds.setAdcRange(1024);  //1024 for 10bit
ADC;4096 for 12bit ADC
  gravityTds.begin();  //initialization

  // set pump pins for output
  pinMode(waterPumpPin,OUTPUT);
  pinMode(phAcidPumpPin,OUTPUT);
  pinMode(phBasePumpPin,OUTPUT);
  pinMode(nutrientAPumpPin,OUTPUT);
  pinMode(nutrientBPumpPin,OUTPUT);
}
```

UCF UNIVERSITY OF CENTRAL FLORIDA

Parse ph

Nathan To

```
// function to return ph value median of ten
values
// (individual ph values can very greatly)
double getPhValue(){
  double bufferArray[10];
  int size = 10;

  // fill array with po values
  for (int i=0; i<size; i++){
    bufferArray[i] = analogRead(phPin) * 5.0 /
1024;
    delay(30);
  }
```

```
 // sort array
  KickSort<double>::insertionSort(bufferArray,
size);

  // find median
  double Po = (bufferArray[(size-1)/2] +
bufferArray[size/2])/2.0;
  // find ph value
  double phValue = 6.86 - (2.7 - Po) * m;

  return phValue;

}
```

UCF UNIVERSITY OF CENTRAL FLORIDA

Print Sensor Data

Nathan To

```
// function that reads and prints sensor values for testing
void printSensorValRasp(){
  // read air humidity
  float airHumi = dht.readHumidity();
  // read air temperature as Celsius
  float airTempC = dht.readTemperature();
  // read temperature as Fahrenheit
  float airTempF = dht.readTemperature(true);

  // read in water temperature sensor
  sensors.requestTemperatures();
  float waterTempC = sensors.getTempCByIndex(0);
  float waterTempF = sensors.toFahrenheit(waterTempC);

  // read in water level sensor
  int waterLevelVal = digitalRead(waterLevelPin);// read the level
value of pin A3 and assign if to val

  // read in TDS sensor
  gravityTds.setTemperature(waterTempC);  // set the temperature and
execute temperature compensation
  gravityTds.update();  //sample and calculate
  float tdsValue = gravityTds.getTdsValue();  // then get the value
```

```
  // print sensor values for raspberry pi

  // TDS sensor
  Serial.print(tdsValue,0);
  Serial.print(" ");

  // pH Sensor
  Serial.print(getPhValue());
  Serial.print(" ");

  // Water Level Sensor
  // 0 - Indicates no liquid, 1 - Indicates probe is submerged
  Serial.print(waterLevelVal); // print the data from the sensor
  Serial.print(" ");

  // Water Temperature sensor
  Serial.print(waterTempF);
  Serial.print(" ");

  // Air/Hum sensor
  Serial.print(airHumi);
  Serial.print(" ");

  Serial.print(airTempF);
}
```

UCF UNIVERSITY OF CENTRAL FLORIDA

Water Level Sensor Loop

Nathan To

```
// loop to check and raise water level
// 0 means unsubmerged 1 means submerged
void waterSensorLoop(){
  // reads in water sensor value
  int waterLevelVal = digitalRead(waterLevelPin);
  Serial.println(waterLevelVal);

  // loop while water sensor is unsubmerged
  while(waterLevelVal == 0)
  {
    // Run pump for designated amount of time
    digitalWrite(waterPumpPin, HIGH);
    delay(waterPumpDelay);
    digitalWrite(waterPumpPin, LOW);

    // wait 2 min sec
    //delay(120000);

    // wait 30 sec
    //delay(30000);

    // test delay 5 seconds
    delay(5000);

    // read new water sensor value sensor
    waterLevelVal = digitalRead(waterLevelPin);

    // read water level val to serial monitor
    Serial.println(waterLevelVal);
  }
}
```

UNIVERSITY OF CENTRAL FLORIDA

ph Sensor Loop

Nathan To

```
// loop to check water ph and bring water ph within threshold values
void phSensorLoop(){
  // reads in ph sensor value

  double phVal = getPhValue();

  Serial.print("ph value = ");
  Serial.println(phVal);

  // loop while water ph is not within threshold values
  while(phVal < phLow || phVal > phHigh)
  {
    // if the water ph is too acidic run base pump for designated amount of time
    if(phVal < phLow)
    {
      digitalWrite(phBasePumpPin, HIGH);
      delay(phPumpDelay);
      digitalWrite(phBasePumpPin, LOW);
    }
    // if the water ph is too basic run acid pump for designated amount of time
    if(phVal > phHigh)
    {
      digitalWrite(phAcidPumpPin, HIGH);
      delay(phPumpDelay);
      digitalWrite(phAcidPumpPin, LOW);
    }

    // wait 2 min sec
    //delay(120000);

    // wait 30 sec
    //delay(30000);

    // test delay 5 seconds
    delay(5000);

    // read new ph sensor value sensor
    phVal = getPhValue();
    Serial.print("ph value = ");
    Serial.println(phVal);
  }
}
```

UCF UNIVERSITY OF CENTRAL FLORIDA

TDS Sensor Loop

```
// loop to check water tds and bring water tds within threshold values
void tdsSensorLoop(){
  // read water temptերature value and calibrate the tds sensor
  sensors.requestTemperatures();
  float waterTempC = sensors.getTempCByIndex(0);

  // set the temperature and execute temperature compensation
  // gravityTds.setTemperature(24); // manual
  gravityTds.setTemperature(waterTempC); // from water temperature sensor

  gravityTds.update();  //sample and calculate
  float tdsVal = gravityTds.getTdsValue()-100;  // then get the value

  Serial.print("TDS value: ");
  Serial.println(tdsVal);

  // loop while water tds is not within threshold values
  while(tdsVal < tdsLow)
  {
    // if the water tds is too low run nutrientA pump for designated amount of time

    digitalWrite(nutrientAPumpPin, HIGH);
    digitalWrite(nutrientBPumpPin, HIGH);
    delay(nutrientPumpDelay);
    digitalWrite(nutrientAPumpPin, LOW);
    digitalWrite(nutrientBPumpPin, LOW);

    // wait 2 min sec
    //delay(120000);

    // wait 30 sec
    //delay(30000);

    // test delay 5 sec
    delay(5000);

    // read water tempterature value and calibrate the tds sensor
    sensors.requestTemperatures();
    float waterTempC = sensors.getTempCByIndex(0);

    // set the temperature and execute temperature compensation
    // gravityTds.setTemperature(24); // manual
    gravityTds.setTemperature(waterTempC); // from water temperature sensor

    gravityTds.update();  //sample and calculate

    // read new tds sensor value sensor
    tdsVal = gravityTds.getTdsValue()-100;

    Serial.print("TDS value: ");
    Serial.println(tdsVal);
  }
}
```

Nathan To

UCF UNIVERSITY OF CENTRAL FLORIDA

Final Loop

```
void loop() {

  // sensor loops
  printSensorValRasp();
  waterSensorLoop();
  phSensorLoop();
  tdsSensorLoop();


  //testing sensors and pumps
  // runAllPumps();
  // printSensorVal();
  // delay(5000);

  // 30 min test delay
  delay(1800000);

  // check sensors after an hour
  // 1 hour delay
  // delay(3600000);
}
```

Nathan To

Relay and Lights

Nathan To

```
#!/usr/bin/env python3
import serial

import RPi.GPIO as GPIO            # import RPi.GPIO module
import time
from time import sleep             # lets us have a delay
GPIO.setmode(GPIO.BCM)             # choose BCM or BOARD

# 3 controls fan
FAN_RELAY = 3
GPIO.setup(FAN_RELAY, GPIO.OUT)    # set GPIO3 as an output
GPIO.output(FAN_RELAY, 1)          # fan starts off

# 2 controls lights
LIGHT_RELAY = 2
GPIO.setup(LIGHT_RELAY, GPIO.OUT)   # set GPIO2 as an output
GPIO.output(LIGHT_RELAY, 1)         # Light starts off

# stablish communication with Arduino
if __name__ == '__main__':
  ser = serial.Serial('/dev/ttyACM0', 9600, timeout=1)
  ser.reset_input_buffer()

# set On and Off times for the light
lightOn = "8:00"
lightOff = "20:00"

# calls current time and converts to integer
def getCurTime ():
  date=time.asctime().split(' ')

  (h, m, s) = date[3].split(':')
  result = int(h) * 3600 + int(m) * 60

  return result;

# returns time to integer
def timeToInt(time):
  (h, m) = time.split(':')
  result = int(h) * 3600 + int(m) * 60

  return result;
```

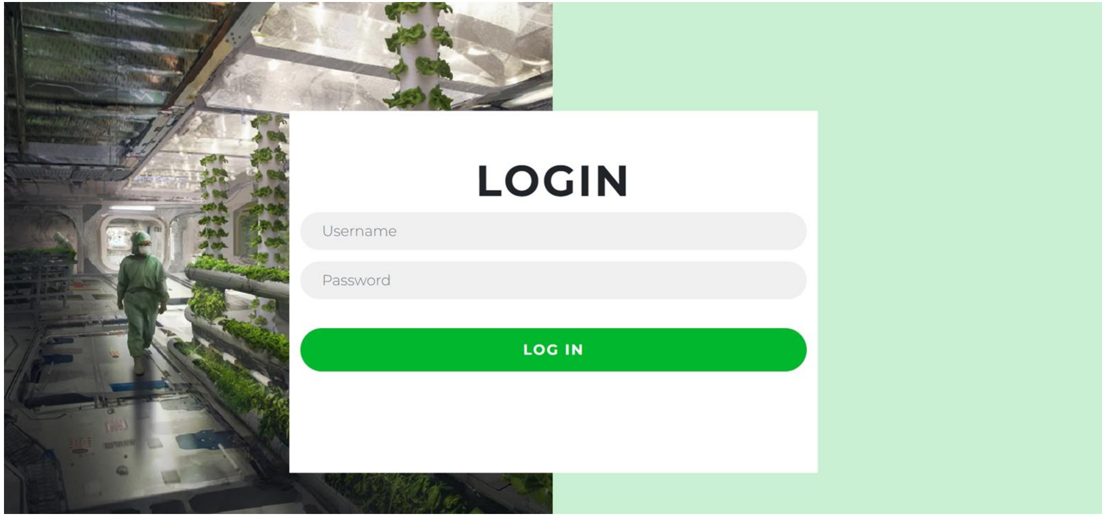
```
try:
  while True:

    if(timeToInt(lightOff) > timeToInt(lightOn)):
      if (timeToInt(lightOn) <= getCurTime() < timeToInt(lightOff)):
        GPIO.output(LIGHT_RELAY, 0)         # Lights on
      else:
        GPIO.output(LIGHT_RELAY, 1)         # Lights off
    else:
      if (timeToInt(lightOff) <= getCurTime() < timeToInt(lightOn)):
        GPIO.output(LIGHT_RELAY, 1)         # Lights off
      else:
        GPIO.output(LIGHT_RELAY, 0)         # Lights on

    if ser.in_waiting > 0:
      humidity = int(ser.readline().decode('utf-8').rstrip())
      print(humidity)


      while humidity > 60:
        print("Adjusting humidity...")
        print(humidity)
        GPIO.output(FAN_RELAY, 0)
        sleep (5)
        humidity = int(ser.readline().decode('utf-8').rstrip())
        print(humidity)
        print("------------------")
      GPIO.output(FAN_RELAY, 1)

except KeyboardInterrupt:          # trap a CTRL+C keyboard interrupt
    GPIO.cleanup()                 # resets all GPIO ports used by this program
```

UCF UNIVERSITY OF CENTRAL FLORIDA

Website Application

Leandro Alepuz

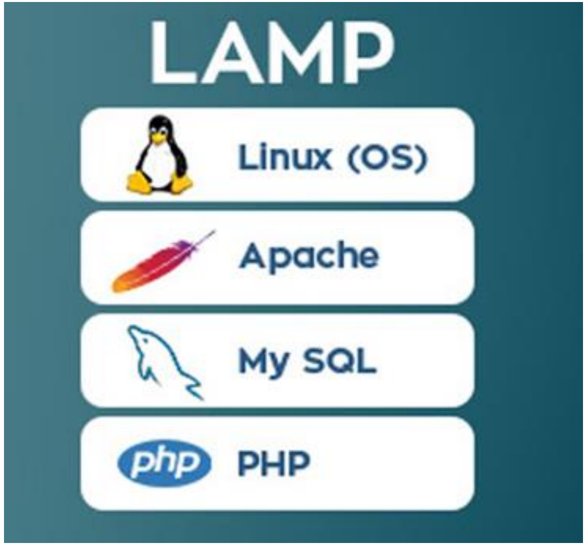
- ❖ Login and user authentication
- ❖ Receive and display sensor data



UNIVERSITY OF CENTRAL FLORIDA

Full-stack technology used : LAMP

Leandro Alepuz

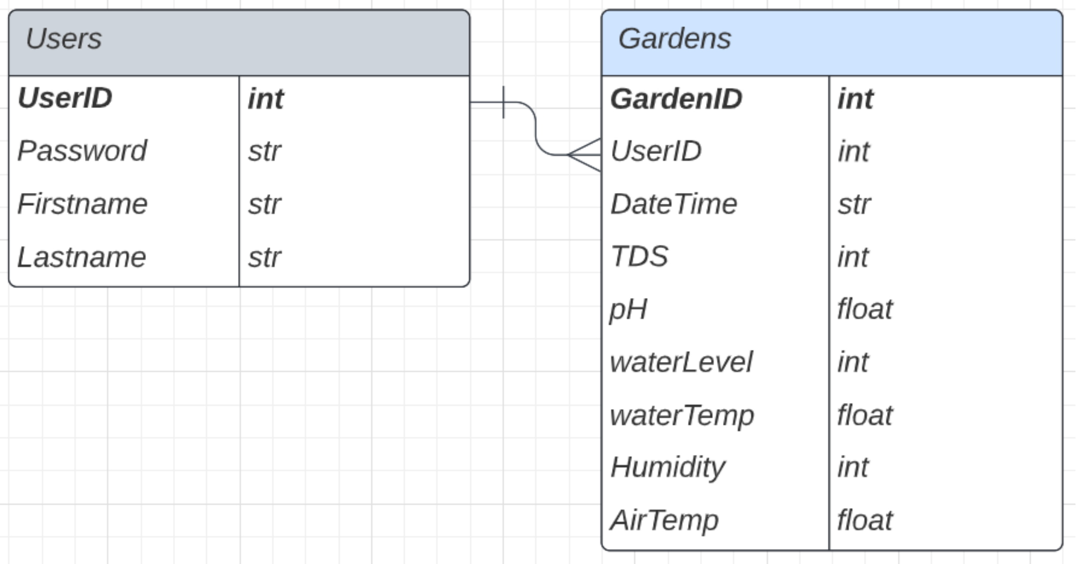


- ➢ Perfect for low-scale projects
- ➢ MySQL fits the needs for our sensor data
- ➢ Previous experience working with Linux through MobaXterm
- ➢ PHP is simple and quick for handling user authentication and display data

UCF UNIVERSITY OF CENTRAL FLORIDA

Database Structure

Leandro Alepuz



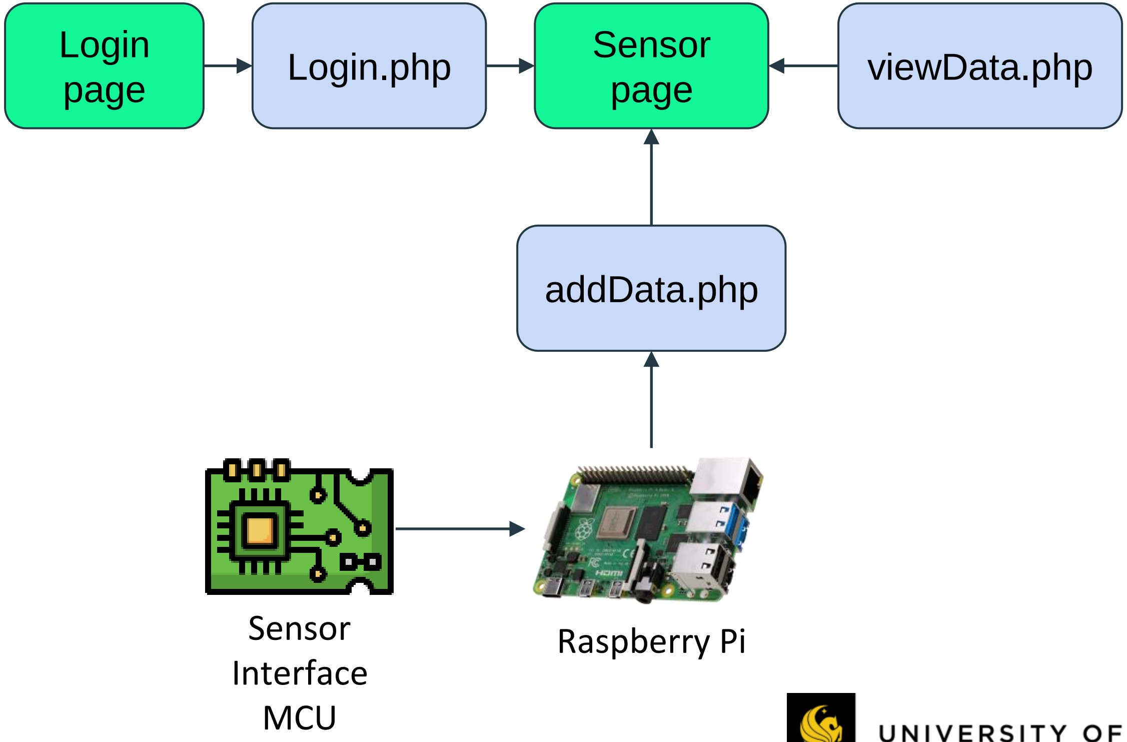
Users	
UserID	***int***
Password	*str*
Firstname	*str*
Lastname	*str*

Gardens	
GardenID	***int***
UserID	*int*
DateTime	*str*
TDS	*int*
pH	*float*
waterLevel	*int*
waterTemp	*float*
Humidity	*int*
AirTemp	*float*

UNIVERSITY OF CENTRAL FLORIDA

Backend Flowchart

Leandro Alepuz



Login page

Login.php

Sensor page

viewData.php

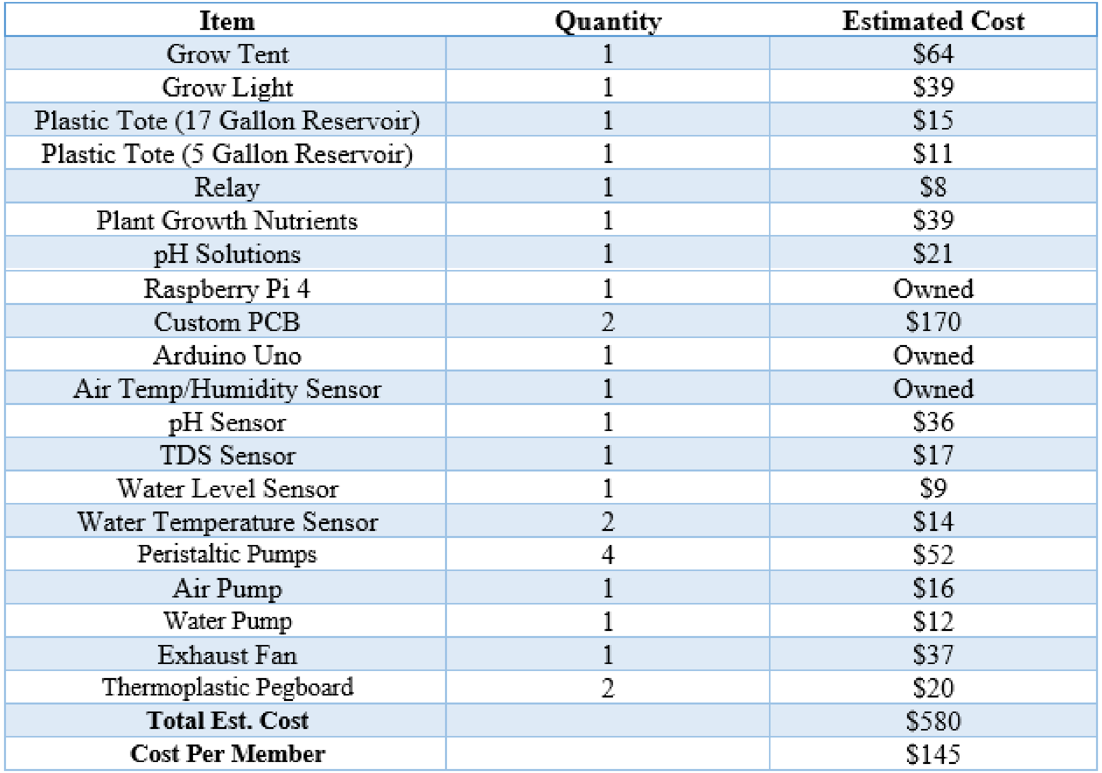
addData.php

Sensor Interface MCU

Raspberry Pi

Budget

Edwin Rivera



Item	Quantity	Estimated Cost
Grow Tent	1	$64
Grow Light	1	$39
Plastic Tote (17 Gallon Reservoir)	1	$15
Plastic Tote (5 Gallon Reservoir)	1	$11
Relay	1	$8
Plant Growth Nutrients	1	$39
pH Solutions	1	$21
Raspberry Pi 4	1	Owned
Custom PCB	2	$170
Arduino Uno	1	Owned
Air Temp/Humidity Sensor	1	Owned
pH Sensor	1	$36
TDS Sensor	1	$17
Water Level Sensor	1	$9
Water Temperature Sensor	2	$14
Peristaltic Pumps	4	$52
Air Pump	1	$16
Water Pump	1	$12
Exhaust Fan	1	$37
Thermoplastic Pegboard	2	$20
Total Est. Cost		$580
Cost Per Member		$145

- Initial estimated budget was between $700 - $800
- Largest expense attributed to the custom PCB
- Other minor components not listed are also already owned
- Total cost is approximately $580