SMART GARDEN CONTROLLER

Group 12:

Alexander Burns	Electrical Engineering
Geovanny Chirino	Electrical Engineering
Temple Corson	Computer Engineering
Renan Coelho Silva	Computer Engineering

Project Description

- The Smart Garden Controller brings gardening to the twenty first century. The Smart Garden controller's technology will allow for water saving features, collection of important metrics, and automation of a garden's watering system.

Motivation

- Automate watering of small business/personal gardens.
- Number of households participating in food gardening continues to grow, 6 million households from 2008 to 2013 alone per the National Gardening Association. Average amount spent on the back yard or balcony nationwide in 2015 was $401 per household.
- Current products with automated features can be expensive and not geared toward gardening.
- A profit could be turned by creating a product which could provide some of the high-end features at a lower price point.
- Save water and ensure plants are water adequately.
- Gardening has many health benefits such as immune regulation, brain health, dexterity, heart health, and mental health.

Goals & Objectives

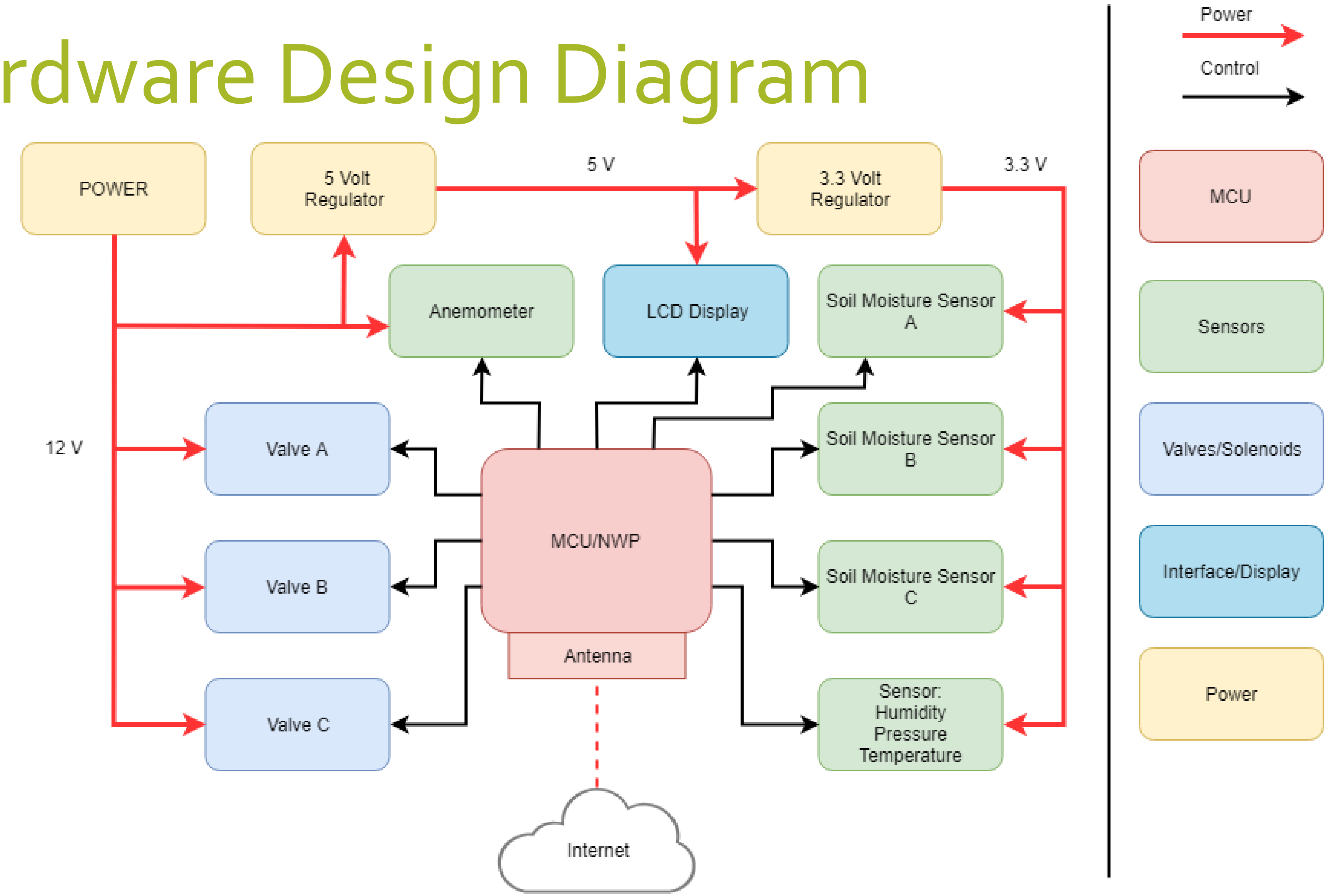
The Goal of this project is to design and build a smart garden controller which will help home gardeners ensure that their plants thrive with the right amount of irrigation provided. In addition, the system should also provide important data to the user which will be helpful in tuning the watering settings of the device. The controller should be affordable when compared to the competition while providing similar features.

- Automate garden watering
- Environmental Data Collection
- Maximizing efficient gardening practices, save water based on weather conditions
- Bring home gardening to the 21st century

Specifications

Requirement	Value
Communication from controller to Web Server	WiFi (802.11)
On demand Watering controlled remotely	Essential
Water saving features	Stretch feature
Web Application for configuration and sensor metrics review	Essential
Data Collected	
Humidity	%
Pressure	HectoPascals (hPa)
Temperature	Degrees Fahrenheit
Light	Relative
Moisture	Relative
Wind	m/s
Hardware	
Number of zones	Three
Smart home device integration	Stretch feature
Dimensions	Less than 10'x10'x10'
Water Resistant	Stretch feature
Weight	Less than 10 lbs.

Hardware Design Diagram

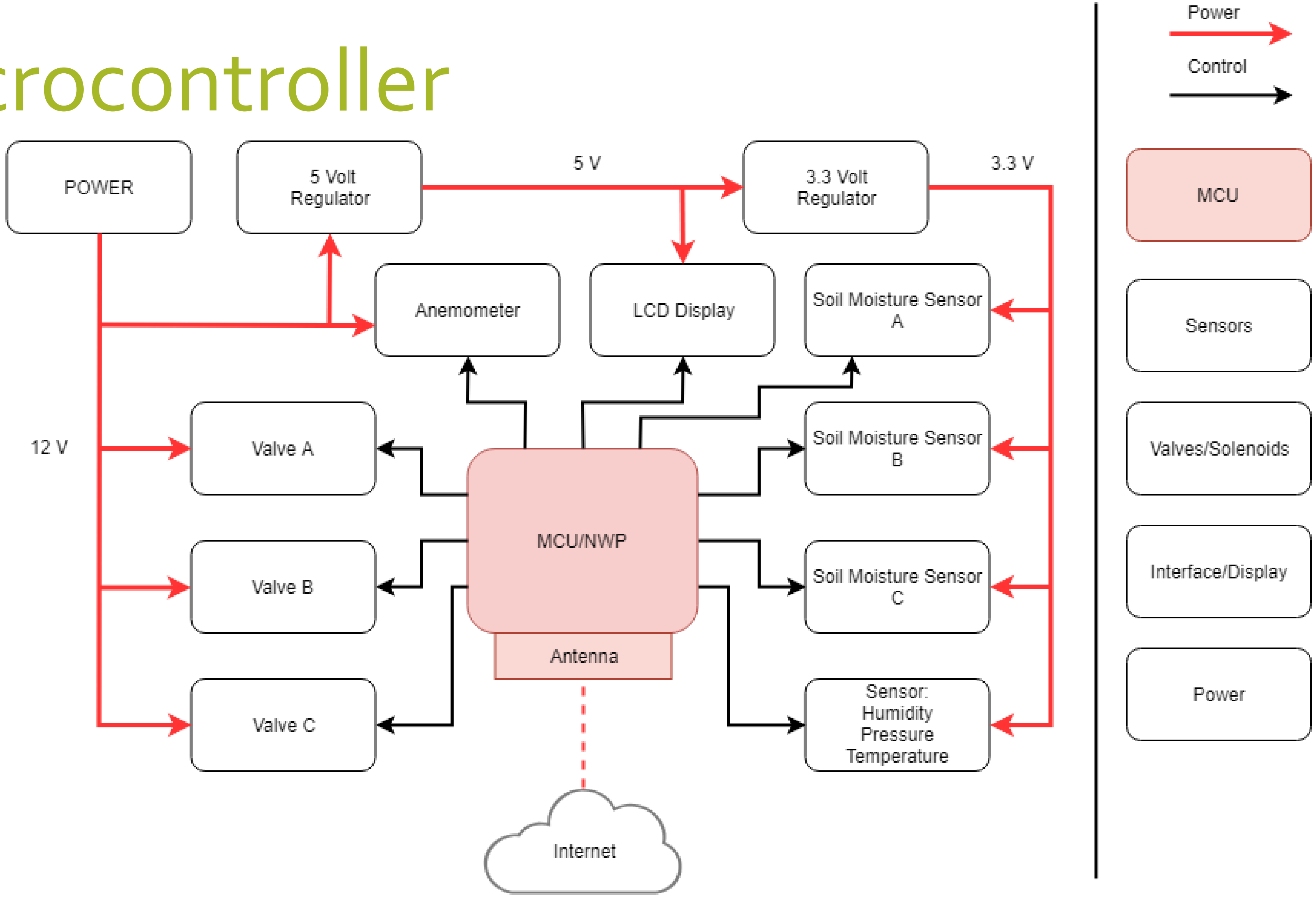


Design & Component Decisions

The following sections will discuss the design decisions our team made to ensure the required features of the system are provided. These design choices include...

- The microcontroller chosen to save & process data, manipulate the sub-systems, and communicate with the software components.
- The sensors that will be used to collect environmental data.
- The valve components that will control the flow of water to each plot in the garden.
- The software to maintain a database of user and garden information, a web application for users to control the system and view garden data, and logic for the system to water autonomously.

Microcontroller



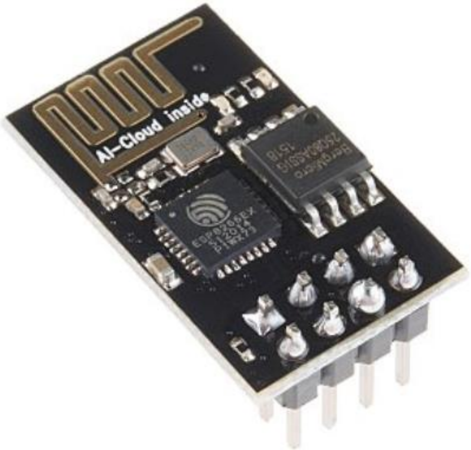
Microcontroller

Purpose:

- Keep schedule which will control the watering of plants.
- Communicate with sensors.
- Control LCD.
- Commutate to web server and upload data.

Selection Criteria:

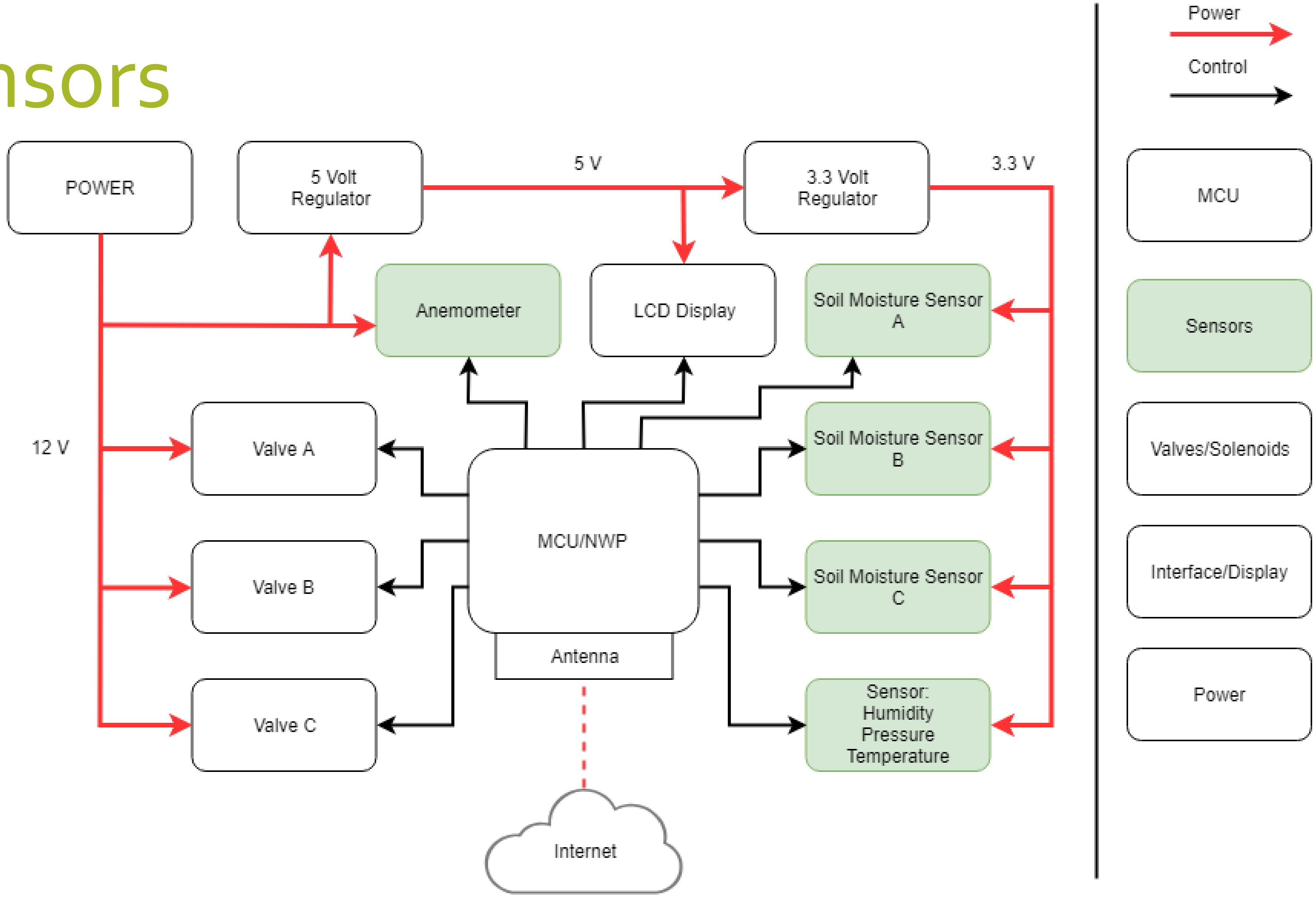
- Memory
- Storage
- Large number of I/O lines
- Integrated NWP



Microcontroller

Microcontroller Trade Analysis			
Selected	No	No	Yes
Module	ATWINC1500	ESP 8266	CC3220MODASF
Price	$8.08	$5.00	$11.64
Voltage Range	2.7V-3.6V	2.7V-3.6V	2.3V-3.6V
Processor	Cortus APS3	Tensilica L106	Cortex–M4
Bits	32 - bit	32 - bit	32 – bit
RAM	128kB	50kB	256kB
Flash	4MB	n/a	1MB
Data Rate	72.2Mbps	72.2Mbps	54Mbps

Sensors

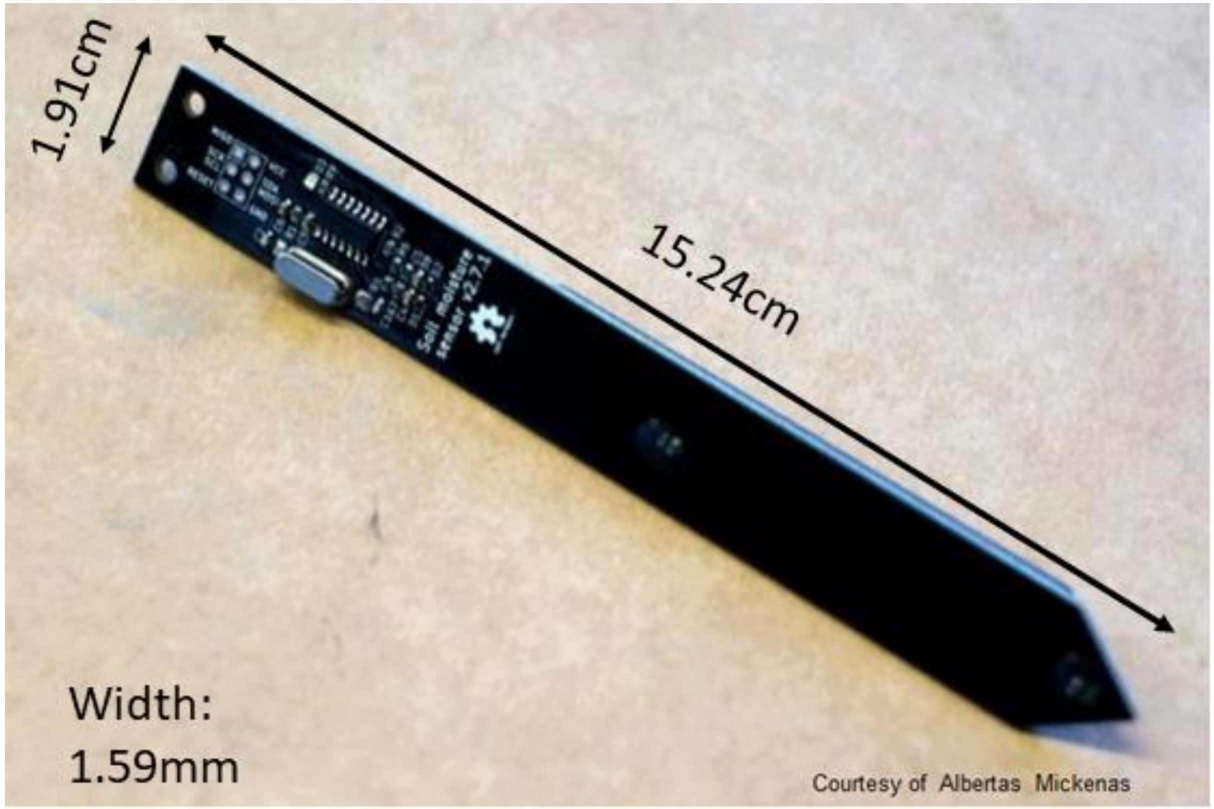

POWER
5 Volt Regulator
5 V
3.3 Volt Regulator
3.3 V
Anemometer
LCD Display
Soil Moisture Sensor A
12 V
Valve A
Soil Moisture Sensor B
MCU/NWP
Valve B
Soil Moisture Sensor C
Antenna
Valve C
Sensor: Humidity Pressure Temperature
Internet
Power
Control
MCU
Sensors
Valves/Solenoids
Interface/Display
Power

Sensors

- The collection and intelligent analysis of environmental data is a key objective of the automated Smart Garden.
- To provide users an advanced and informed connection to their garden, the system employs three sensor modules to collect environmental data.
- The system measures six major environmental metrics: temperature, relative humidity, atmospheric pressure, soil moisture, ambient light, and wind speed.
- The following sections will provide insight and information regarding the sensors used in the design of our system.

Moisture, Temperature, & Ambient Light

This I2C Soil Moisture Sensor uses a capacitive circuit to measure a relative moisture level in soil, while also returning a relative ambient light measurement, and temperature within a range of 0°C to 85°C (32°F to 185°F). Purchased at $12.49 ea. from Catnip Electronics via Tindie.



Moisture, Temperature, & Ambient Light (con't)

Description	Value
Version	2.7.5
Supply Voltage	3.3V - 5V
Current Consumption (idle)	0.7mA - 1.1mA
Current Consumption (measure)	7.8mA - 14mA
Current Consumption (constant)	2.8mA - 4.5mA
Operating Temperature	0°C - 85°C
Crystal Speed	16MHz
Default I2C Address	0x20

Moisture, Temperature, & Ambient Light

Available Registers:

Initially all soil moisture sensors will have the same I2C address. Will need to update each address first in order to use them.

Name	Register	R/W	Data length
GET_CAPACITANCE	0x00	(r)	2
SET_ADDRESS	0x01	(w)	1
GET_ADDRESS	0x02	(r)	1
MEASURE_LIGHT	0x03	(w)	0
GET_LIGHT	0x04	(r)	2
GET_TEMPERATURE	0x05	(r)	2
RESET	0x06	(w)	0
GET_VERSION	0x07	(r)	1
SLEEP	0x08	(w)	0
GET_BUSY	0x09	(r)	1

Humidity & Pressure

The Bosch BME280 Humidity and pressure sensor measures relative humidity in the standard range of 0% - 100% and atmospheric pressure within the range of 300 - 1100 hectopascals (hPa). This sensor uses I2C or SPI protocol and was purchased at $16.09 from Digi-key Electronics.

Description	Value
Supply Voltage	1.71V - 3.6V
Temperature range	-40°C - 85°C
Interface	I2C, SPI
Package size (Metal lid LGA)	2.5x2.5x0.93mm^3
Current consumption @ 1Hz	3.6µA
Operating supply voltage	1.8V
Operating supply current	0.5µA

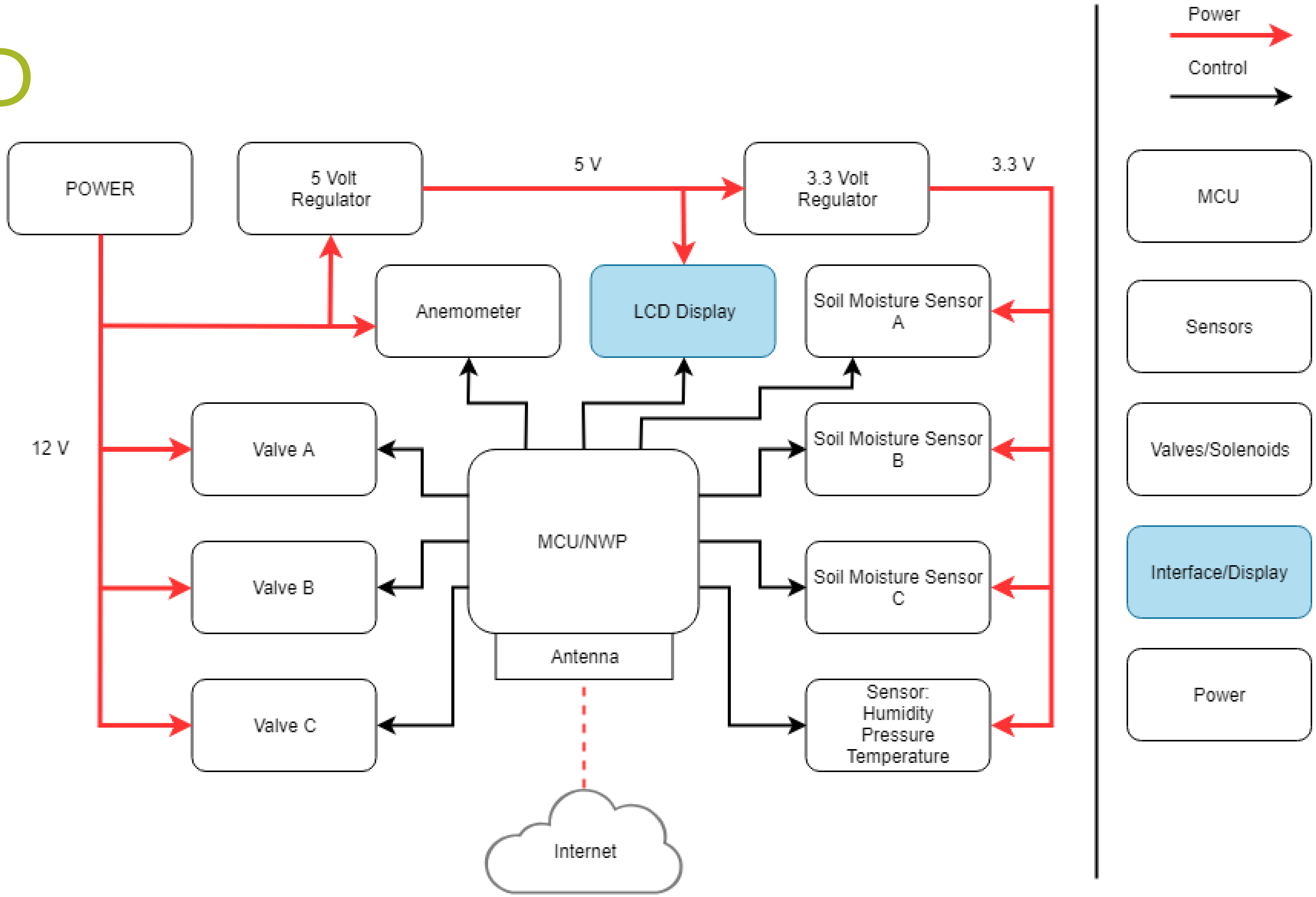
Wind Speed

This Anemometer can measure wind speeds up to 32.4 m/s (72.48 mph), and communicates via analog voltage output. This was the most expensive sensor at $44.95 from Adafruit, however on a single component is used in the final design, unlike the three necessary soil sensors.

Wind Speed (con't)

Description	Value
Output Voltage	0.4V - 2.0V
Testing Range	0.5 m/s - 50 m/s
Resolution	0.1 m/s
Accuracy (Worst Case)	+/- 1 m/s
Max wind speed	70 m/s
Mass	111.8 g
Height	105mm
Arm Length	70mm

LCD


POWER
5 Volt Regulator
5 V
3.3 Volt Regulator
3.3 V
Anemometer
LCD Display
Soil Moisture Sensor A
12 V
Valve A
Soil Moisture Sensor B
MCU/NWP
Valve B
Soil Moisture Sensor C
Antenna
Valve C
Sensor: Humidity Pressure Temperature
Internet
Power
Control
MCU
Sensors
Valves/Solenoids
Interface/Display
Power

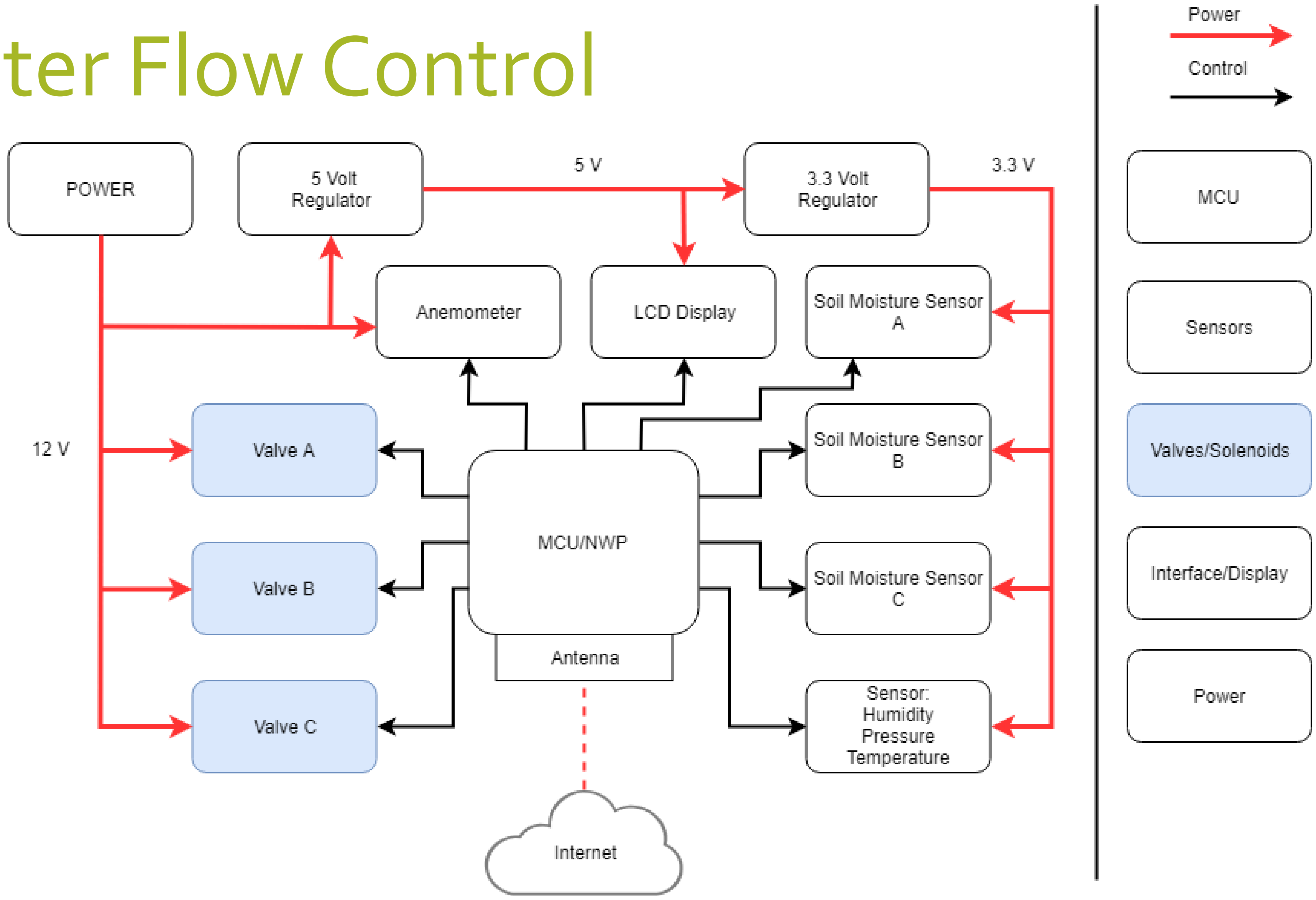
LCD

Purpose:

- View sensor data directly on device.
- Together with buttons, provide a user interface to configure device.
- Configure Username and password.
- Configure wireless network SSID and password.
- View watering schedule configuration.

Description	Value
Interface	I2C
Default Address	0x27
Pins	Gnd, Vcc, SDA, SCL
Text dimensions	4 lines x 20 column
Backlight color	Blue
Text color	White
Supply Voltage	5.0V

Water Flow Control



Water Flow Control components

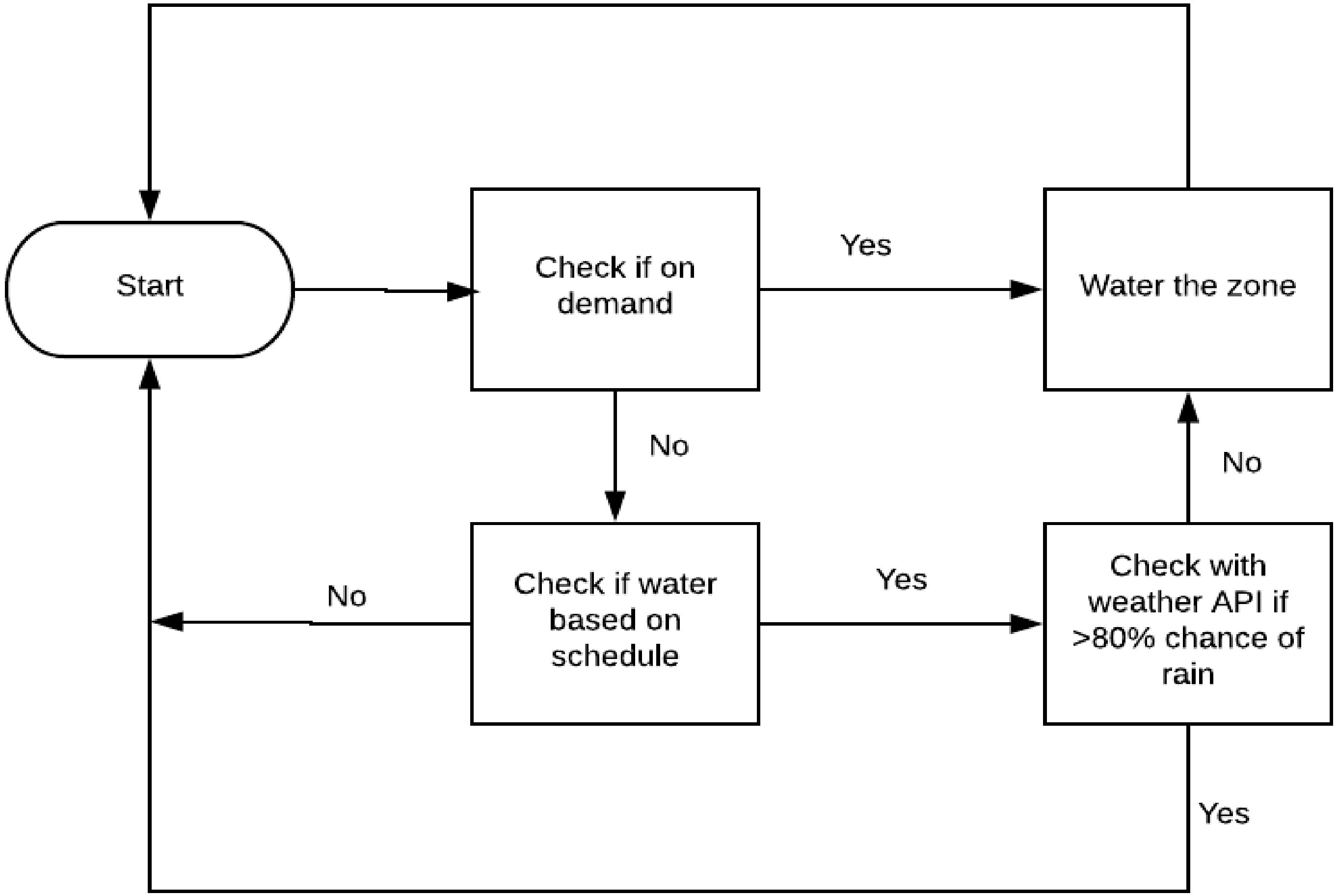
Water Flow Control (con't)

- The solenoid valve that were selected for use in this project were selected for their cost effectiveness as well as their availability and user reviews.

- The solenoid valves that were chosen for this project design are 12-volt devices. 12V applied= OPEN / Lower than 12V=Closed if there is sufficient current from the source. (Solenoid valve has a typical current draw of 2.4A).

- High voltage and current draw meant for the implementation of a high side switch, due to microcontroller operating at 3.3-volts and only consuming an average of 180mA itself, so there is no way it would be able to supply the power to open the solenoid valve itself.

Water flow control logic

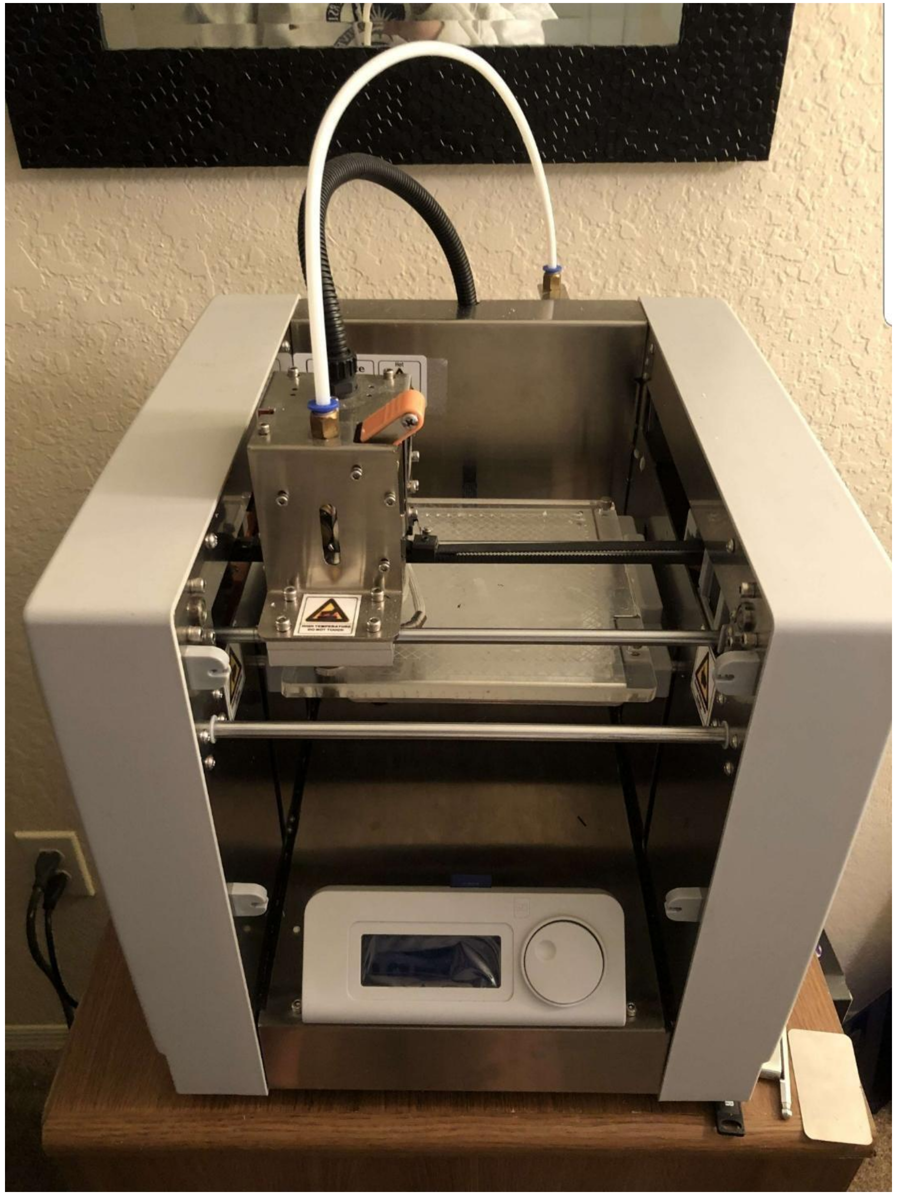


Custom Enclosure

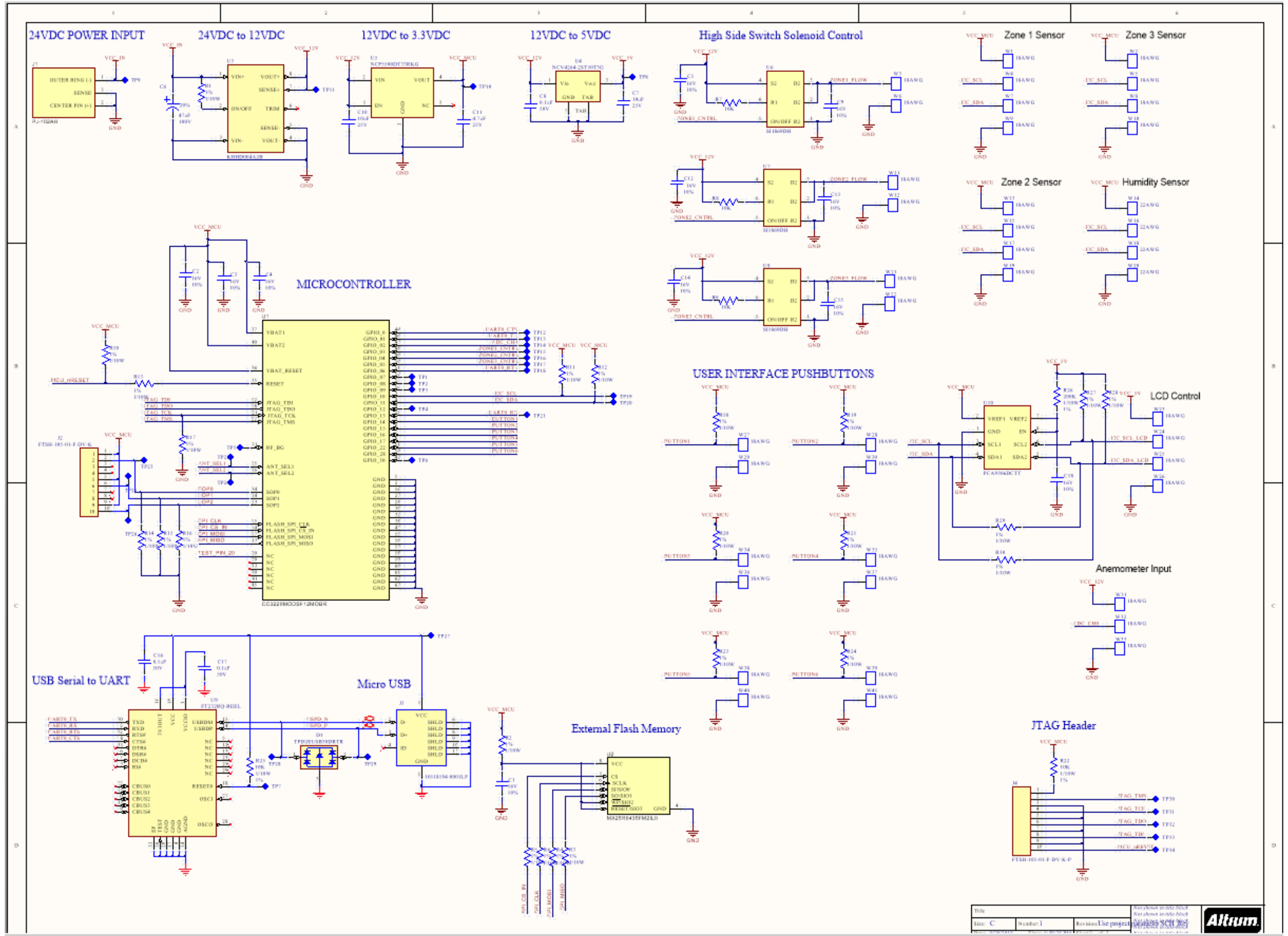
- Enclosure will be 3D printed to allow for secure and optimizable design, as well as to provide more practice for our team in 3D printing.
- To function properly in the operational environment the chassis must be rain resistant from above, to protect the microcontroller and other electrical components.
- Make sure there is enough room for the sensors and ports so that the device is protected from repeated outside use and that the information recorded is accurate and reliable.
- The chassis must also allow the user to access the housed components if necessary.

3D Printer

- 150mm x 150mm x 140mm printable area.
- Extruder can reach temperature of up to 250 degrees Celsius/ 482 degrees Fahrenheit.
- Heated Bed that can reach temperature of up to 90 degrees Celsius/194 degrees Fahrenheit.
- Speed of up to 50mm/s and layer resolution as accurate as 0.1mm.
- Uses both ABS (Water resistant) and PLA (Non-water resistant) plastic to print with.
- Comes with SD memory card and slicing software to convert CAD files into G-code for the printer.

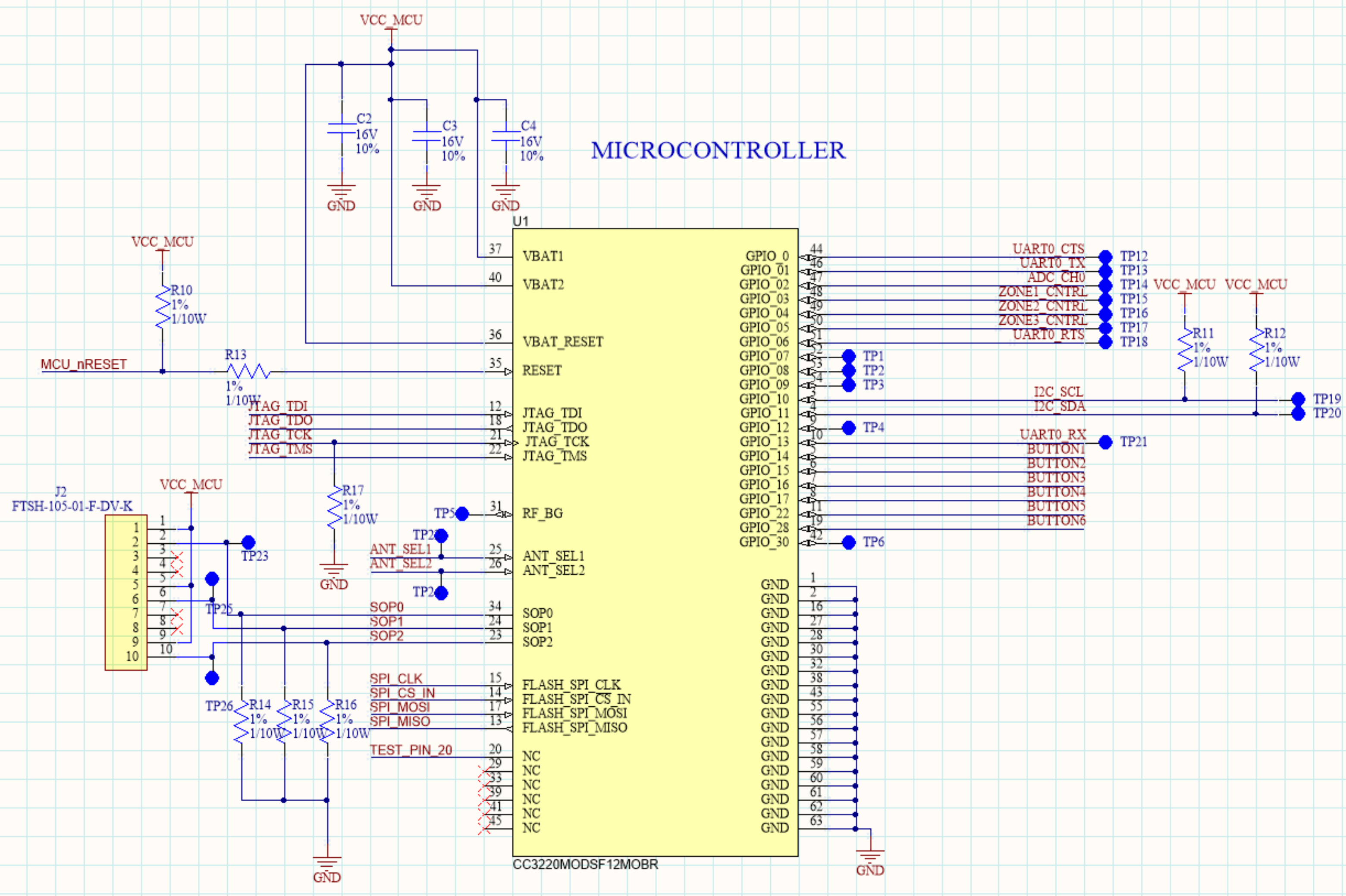


Smart Home Garden Schematics



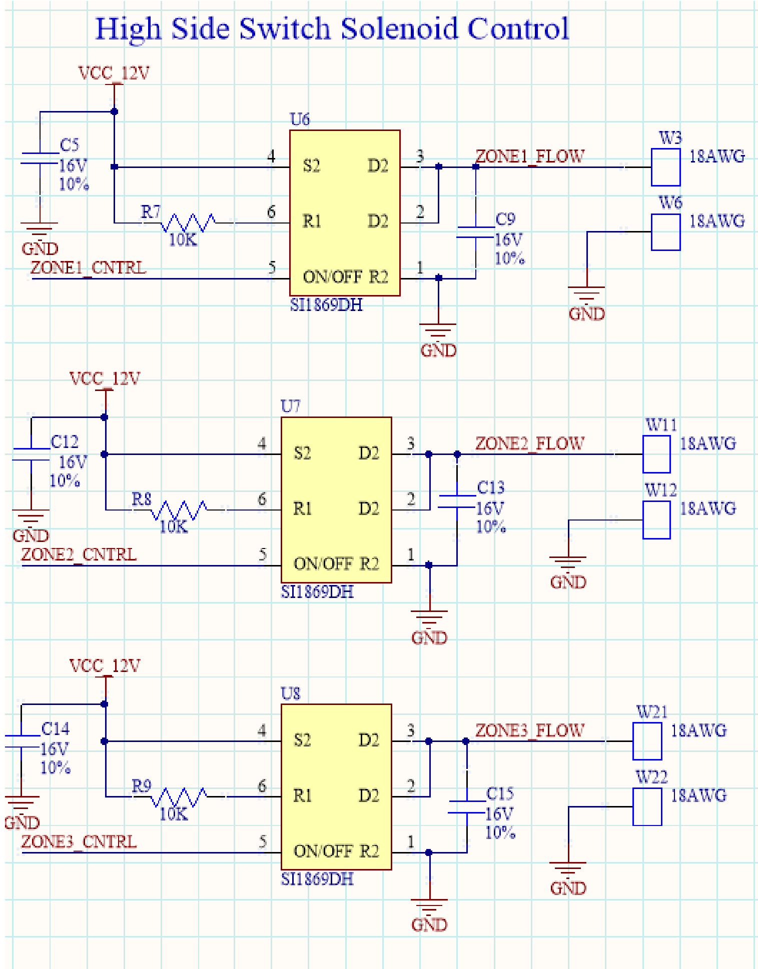
Schematics – Microcontroller/WiFi

- 3.3V Power supply
- Reset signal using pull-up resistor for proper JTAG initialization
- JTAG interface
- SOP signals to connector for bootstrapping options
- SPI signals for FLASH
- GPIO signals for sensors, buttons, LCD
- UART signals for Debug
- I2C signals for peripheral communication



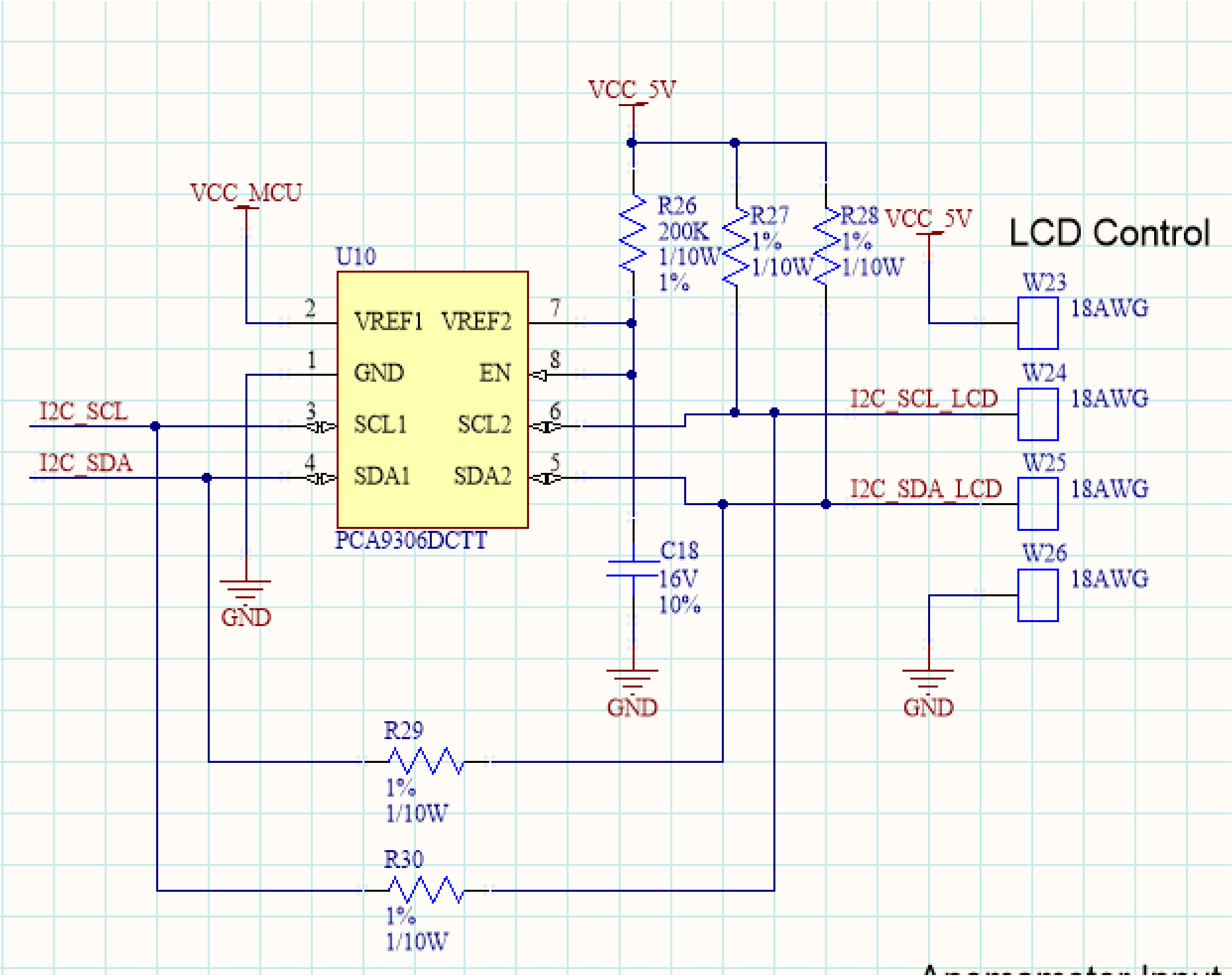
Schematics – High Side Switches

- Offers an electromechanical relay type functionality without any mechanical moving parts.
- 12V power supply.
- Input signal controls voltage/current path from S2 to D2.
- Effectively allows control of 12V signal using 3.3V logic.



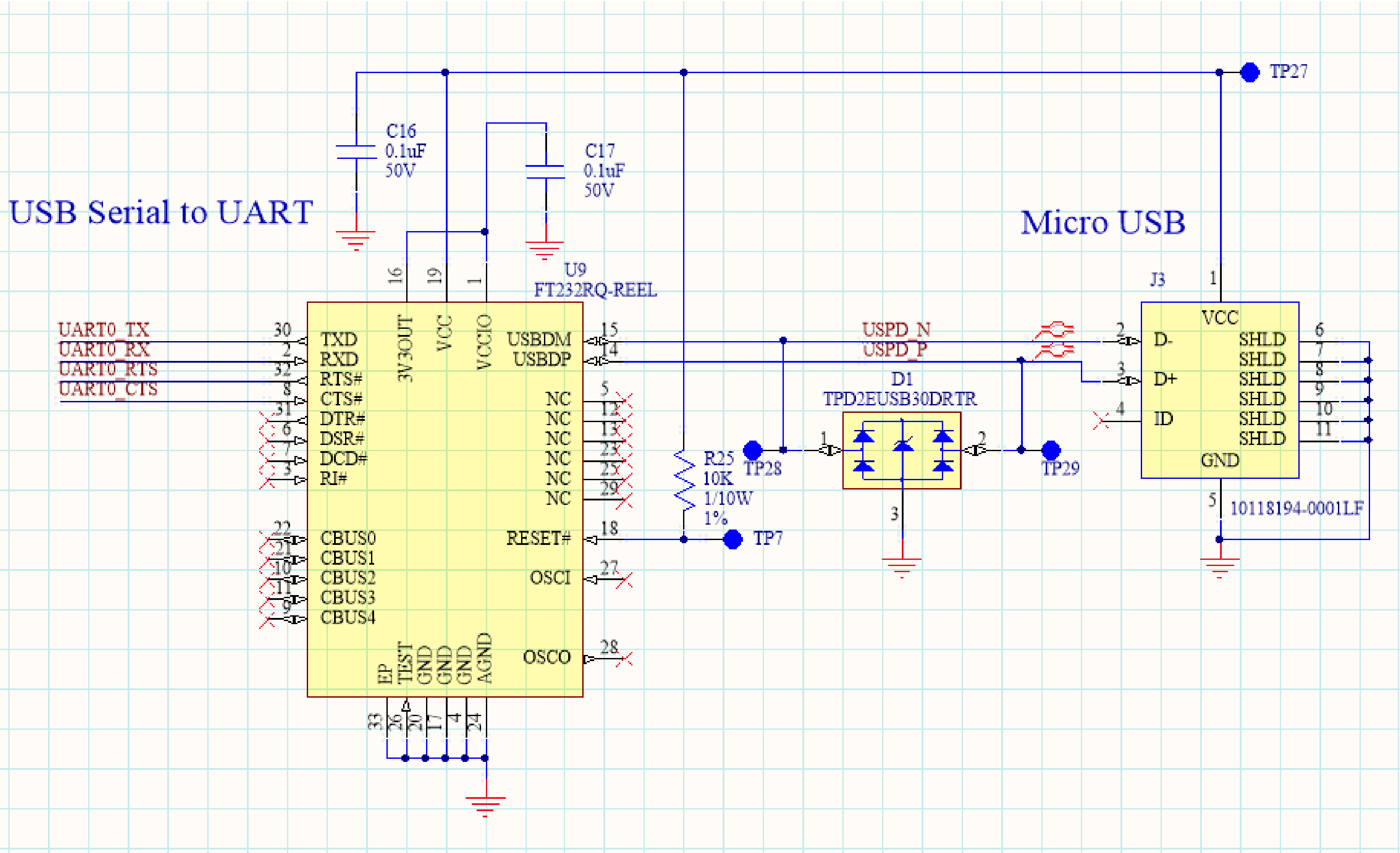
Schematics – I2C Voltage Level Translator

- 3.3V I2C signals in and 5V I2C signals out.
- Allows for control of parts operating at a different logic level than the microcontroller.
- Zero ohm resistors enable a quick change back to 3.3V logic level for LCD if 5V is not required.



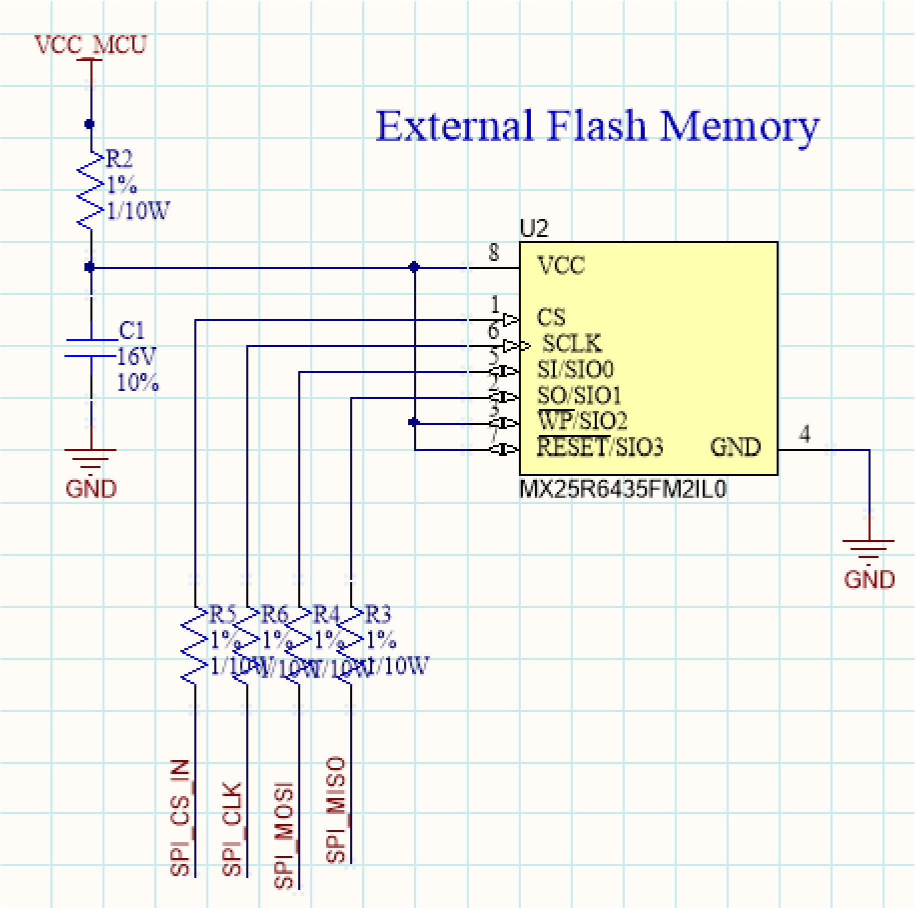
Schematics – USB Serial to UART

- USB to UART interface allows debugging using USB.
- FTDI part does all the work.
- USB port supplies power to the FTDI chip.
- D1 is transient voltage protection to protect against voltage spikes during plug in.

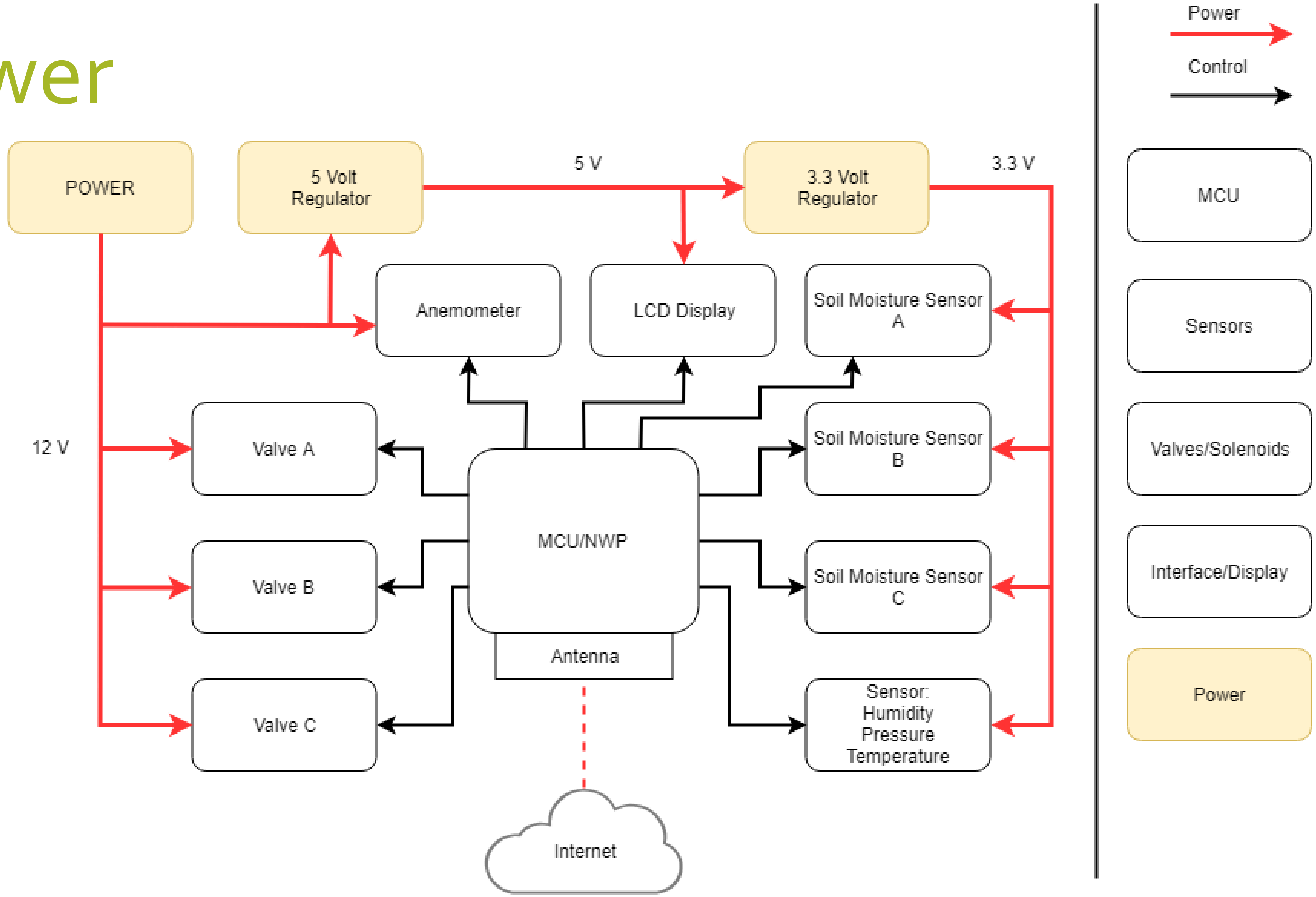


Schematics – External SPI FLASH

- CC3220 has a dedicated SPI FLASH interface.
- 64M-Bit Flash memory.
- For extra data storage.

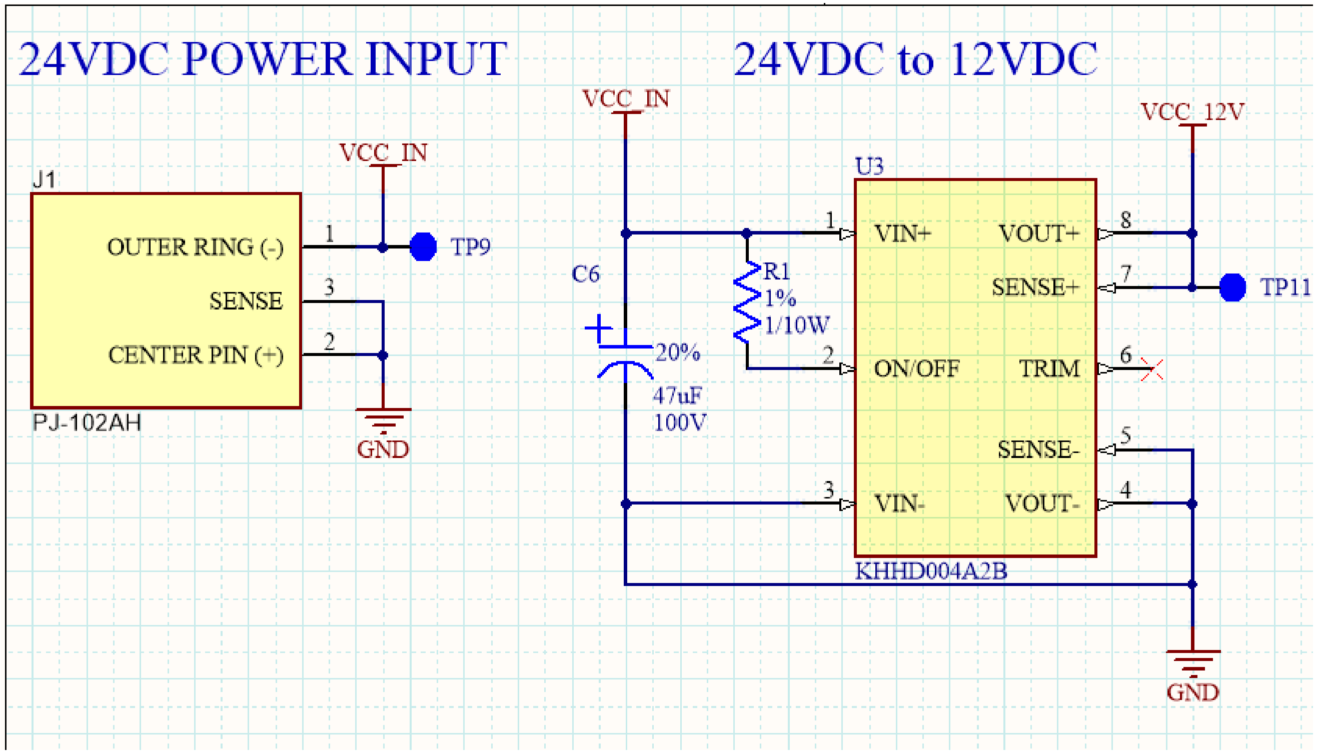


Power


POWER
5 Volt Regulator
5 V
3.3 Volt Regulator
3.3 V
Anemometer
LCD Display
Soil Moisture Sensor A
12 V
Valve A
Soil Moisture Sensor B
MCU/NWP
Valve B
Soil Moisture Sensor C
Antenna
Valve C
Sensor: Humidity Pressure Temperature
Internet
Power
Control
MCU
Sensors
Valves/Solenoids
Interface/Display
Power

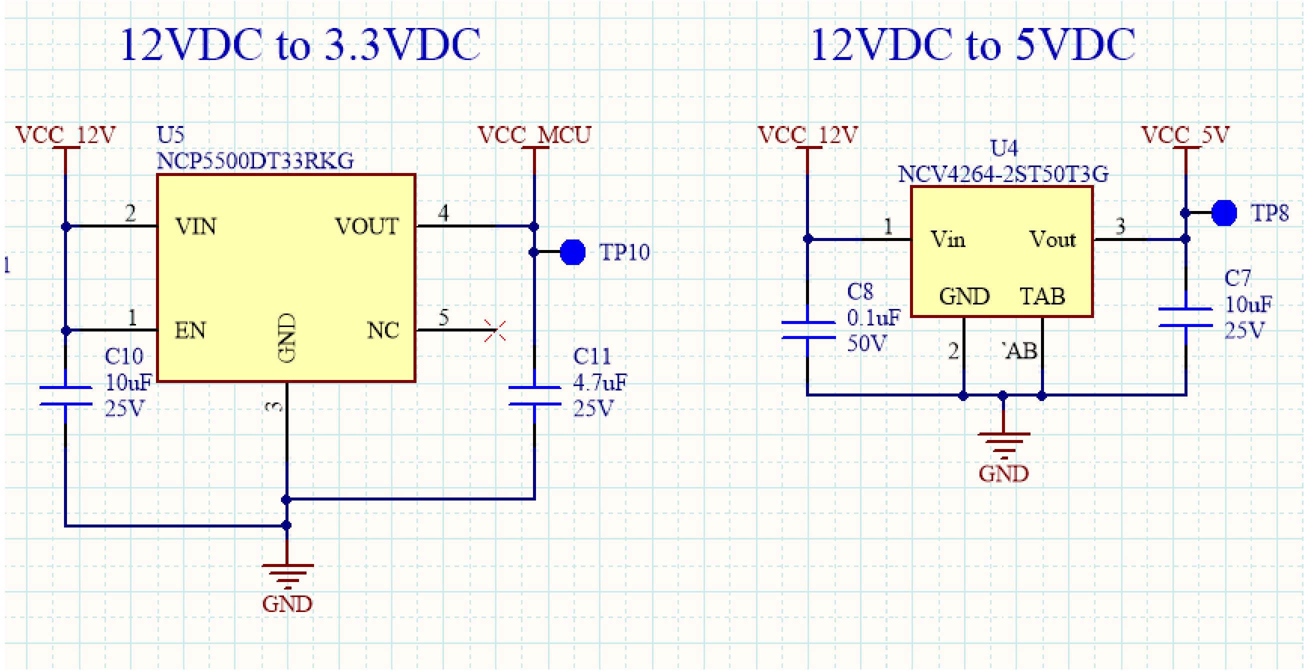
Schematics – Input Power

- 24 VDC power input jack.
- 24VDC supplied by external AC/DC converter.
- Hammerhead™ Series 50W DC to DC Isolated Converter Power Module.
- 90% efficient power switching at constant switching frequency.
- Up to 4.2 amp output.
- 12V rail runs anemometer, and solenoid valves.



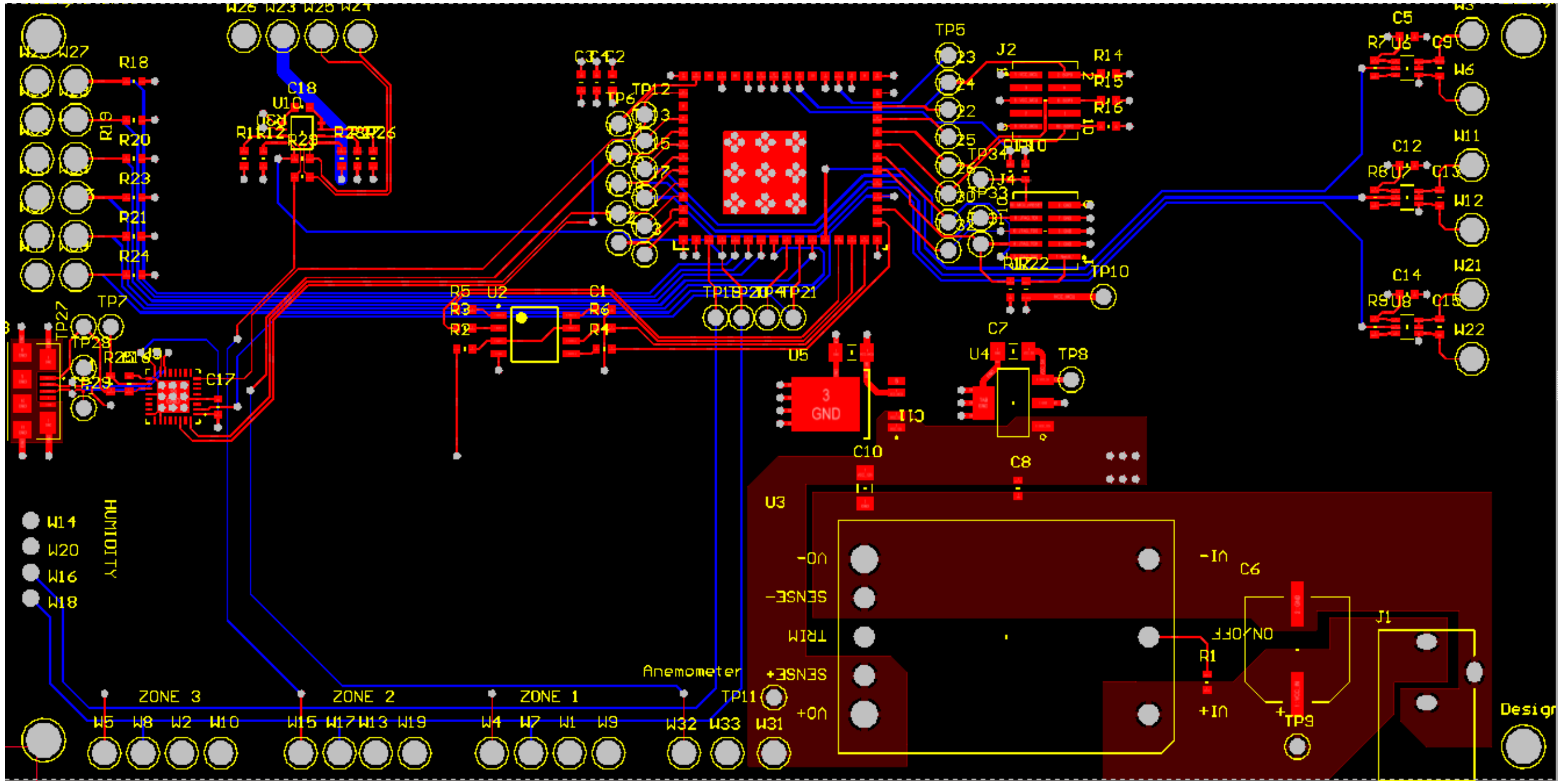
Schematics – Board Power

- Second stage of DC to DC conversion.
- Using simple LDOs for ease of design and to avoid any noise generated by power switching.
- 5V rail is used to supply the LCD screen.
- 3.3V rail is used to supply the microcontroller and all other low power parts.



PCB Board Layout

- Board Design was completed using Altium Designer 18.
- 4 Layer stack up
  - 2 external signal layers (top and bottom)
  - 1 internal ground plane
  - 1 internal power plane
- Dimensions: 6” X 2”.



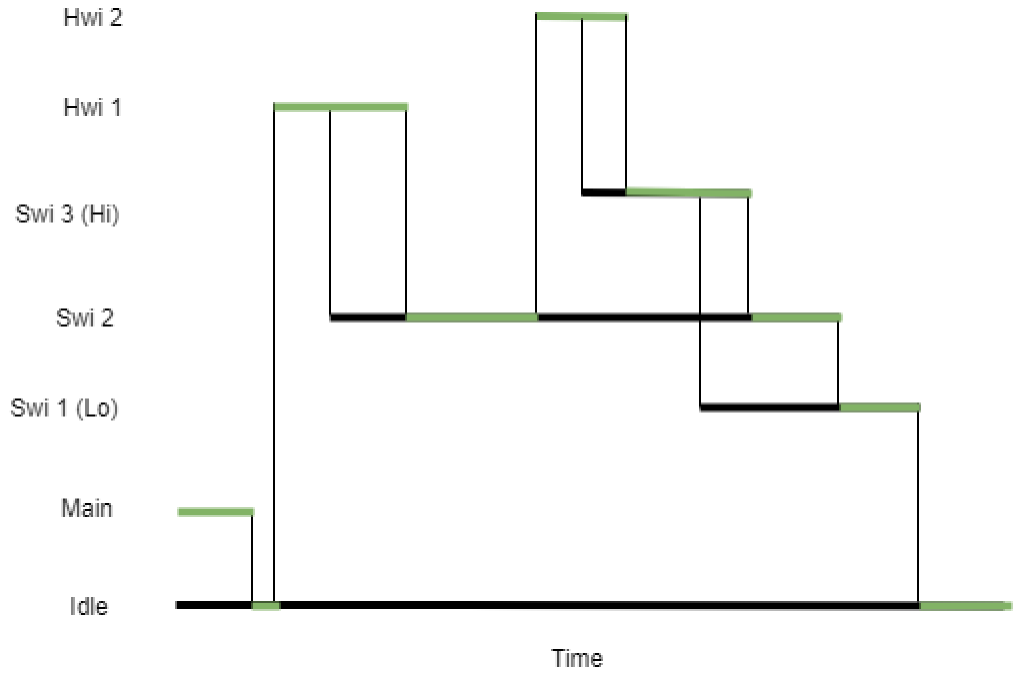
Software Components

Section	Software
MCU	TI-RTOS
Database	MySQL
Web Server	Tomcat
DB and Web Server host OS	Linux
Web Application	Angular/HTML/CSS

MCU: TI-RTOS

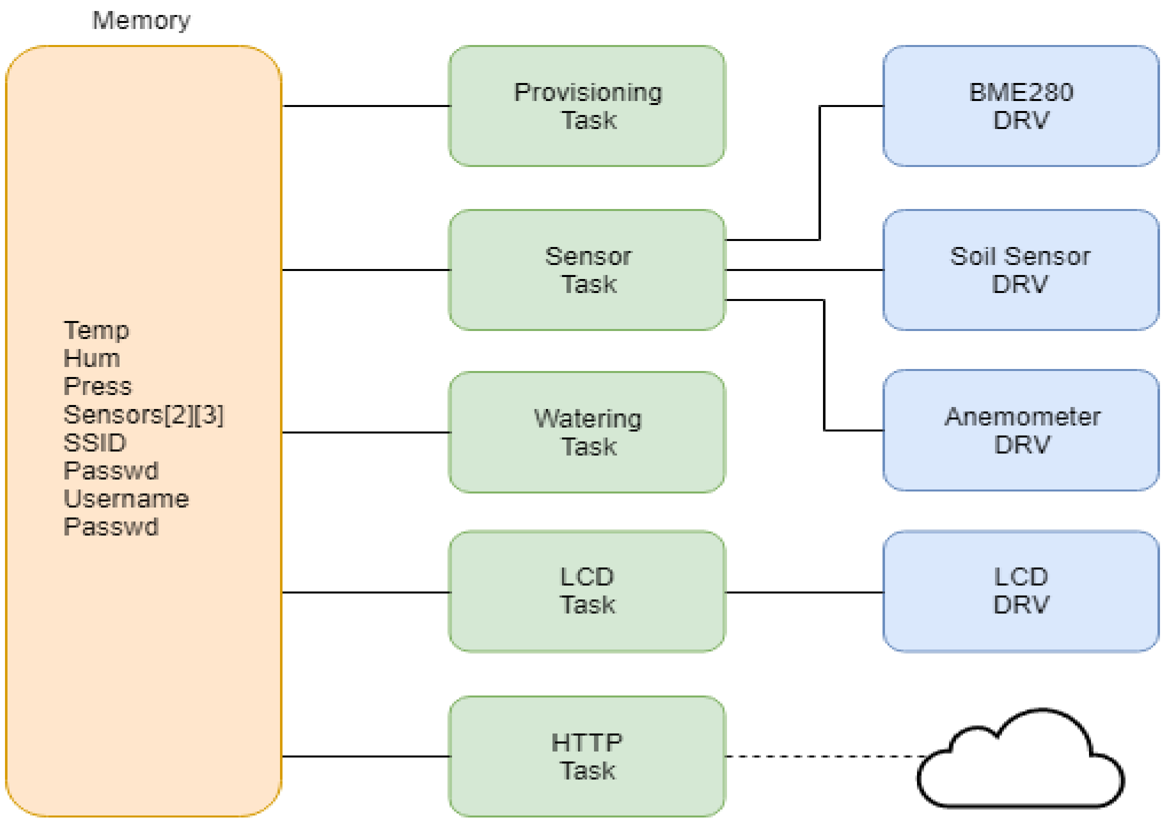
RTOS vs Bare metal

- When using an RTOS there is no need to manage interrupts. RTOS takes care of interrupts.
- RTOS use of drivers and code portability can increase programmer’s performance.
- Bare metal can maximize resources. Likely to be critical only on slower MCUs.



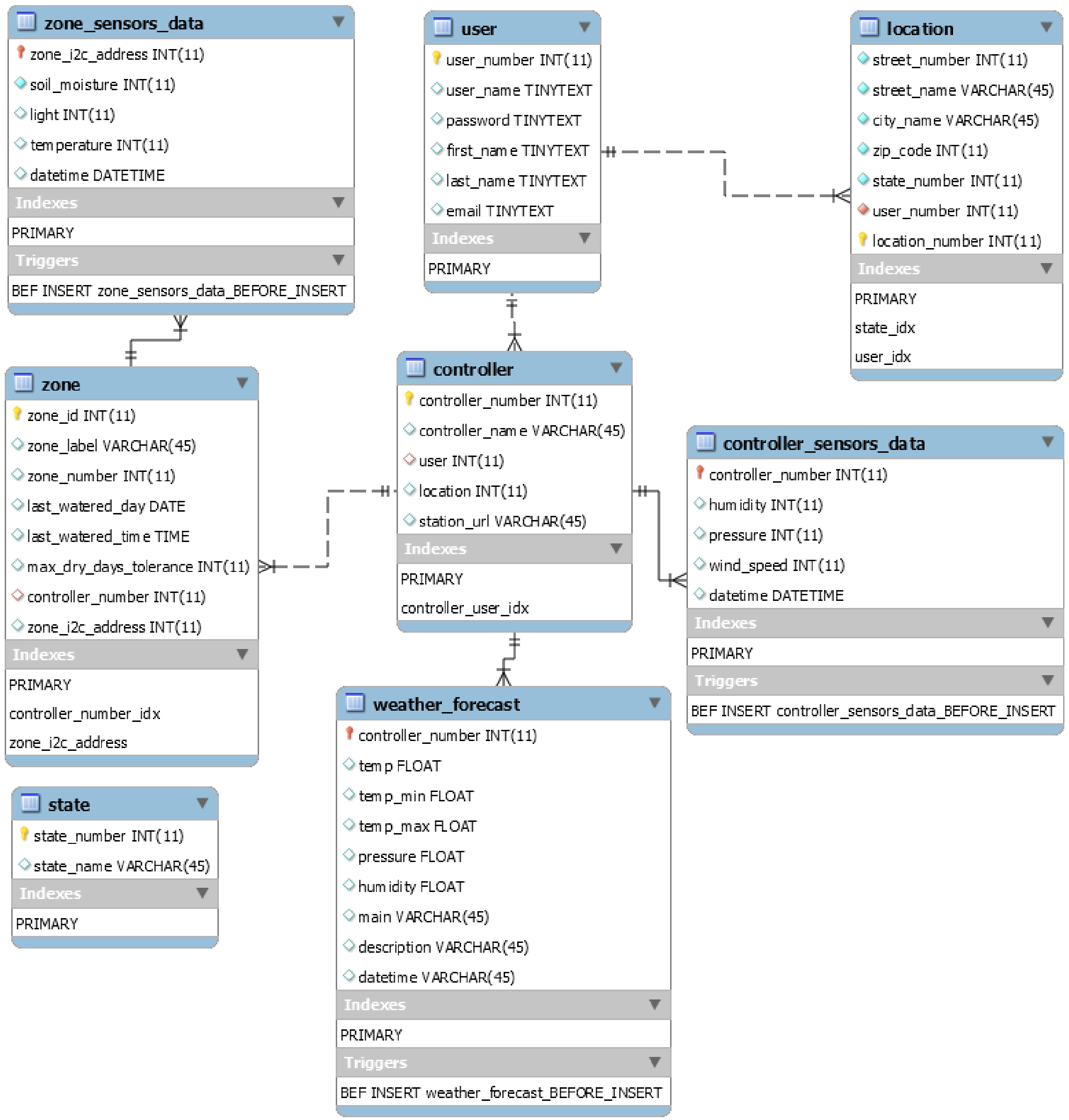
MCU Software: Overview

- Operations will be split amongst task.
- Each task will use the necessary drivers to interact with hardware.
- Provisioning task control connection to access point.
- HTTP task will download and upload data to server as well as check weather forecast.



Database: MySQL

- Database server will keep all the sensor data collected by the smart garden controller.
- Database server will keep all user configuration.
- Database server will keep all scheduling information.
- MySQL software chosen because it is open source.
- MySQL software chosen because has abundant resources available on how to configure and setup.
- MySQL software chosen because it had been used in previous projects as well.



Web Server: Tomcat

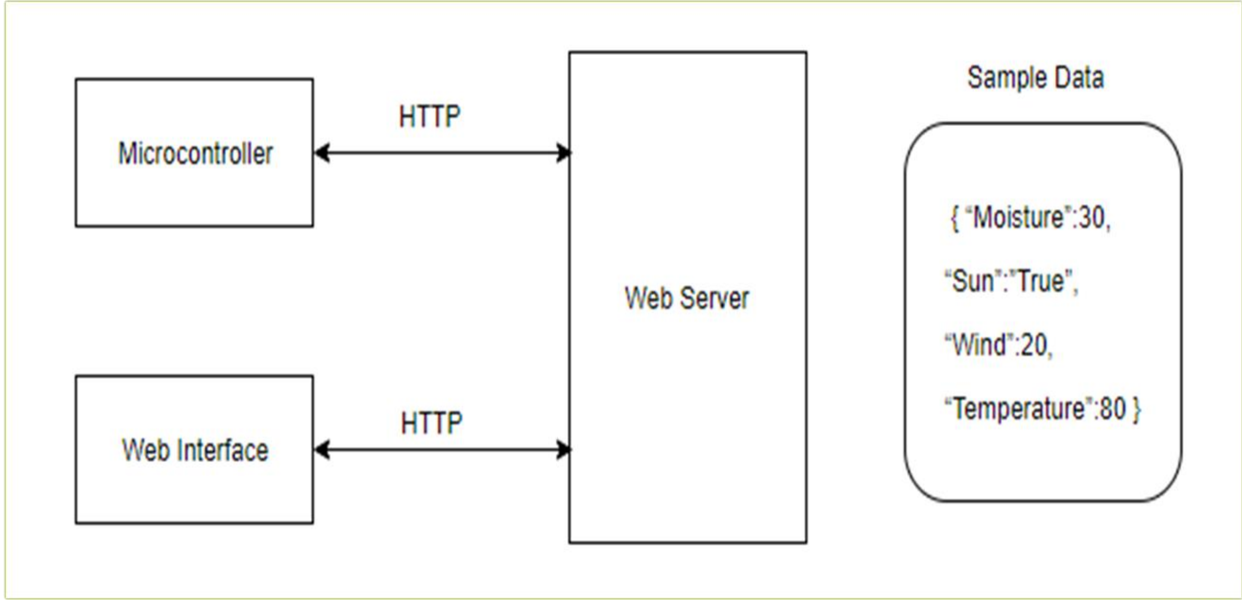
- Programmed in Java with Jersey framework.
- Will work as a gateway between the data and MCD/Web Application.
- Provides RESTful web services to any current or future integration.
- Objects transferred in JSON format.
- Maven used to control libraries in Eclipse.

```java
@Path("user")
public class UserResource {

    public ResourcesDAO userRepo = new ResourcesDAO();

    @GET
    @Produces(MediaType.APPLICATION_JSON)
    public List<User> getUsers() {
       System.out.println("METHOD CALL: UserResource.getUsers()");

       List<User> users = userRepo.getUserList();
       //List<User> users = Arrays.asList(userA, userB);
       return users;
    }
```



Web Application: Angular

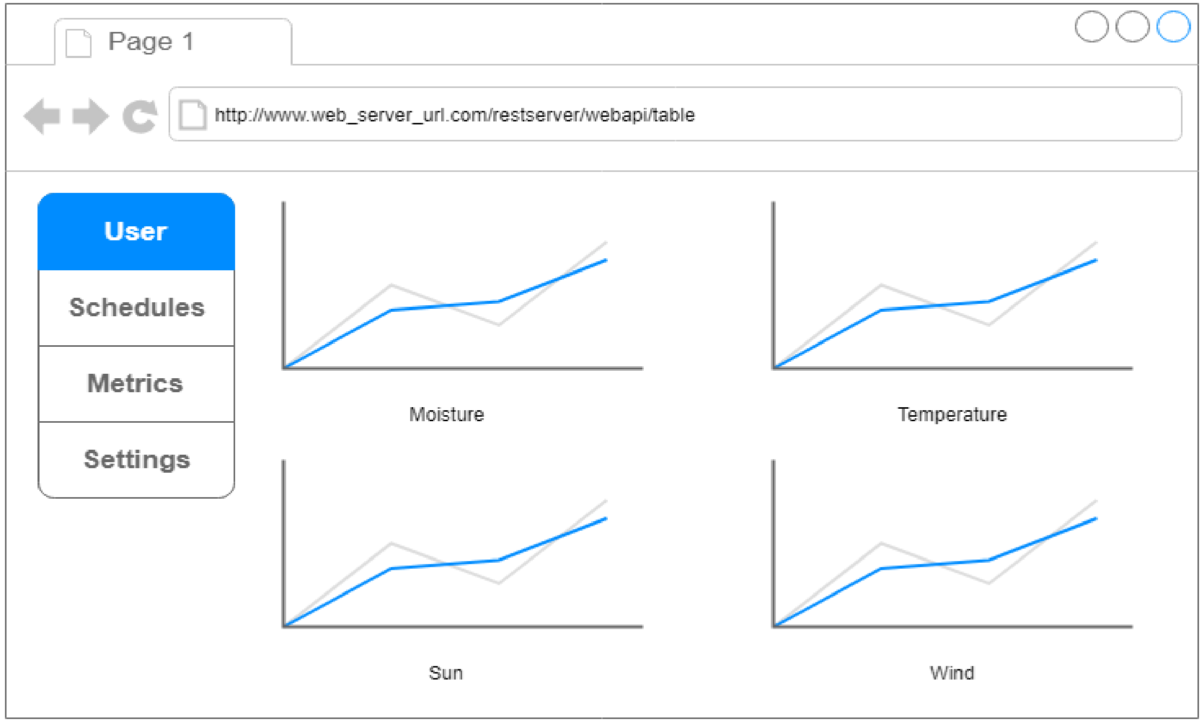
Some available front end frameworks:

- ReactJS
- Ember
- Angular

Why angular?

- Wealth of documentation available.
- Solid reputation.
- Known to be great to develop single page applications (SPA).

In the end, any framework would do it …



Challenges & successes

Issue: None of the hardware components had libraries available for our platform, CC3220SF.
Solution: Read code for Arduino libraries and re-write code for platform being used. This took a lot of time.

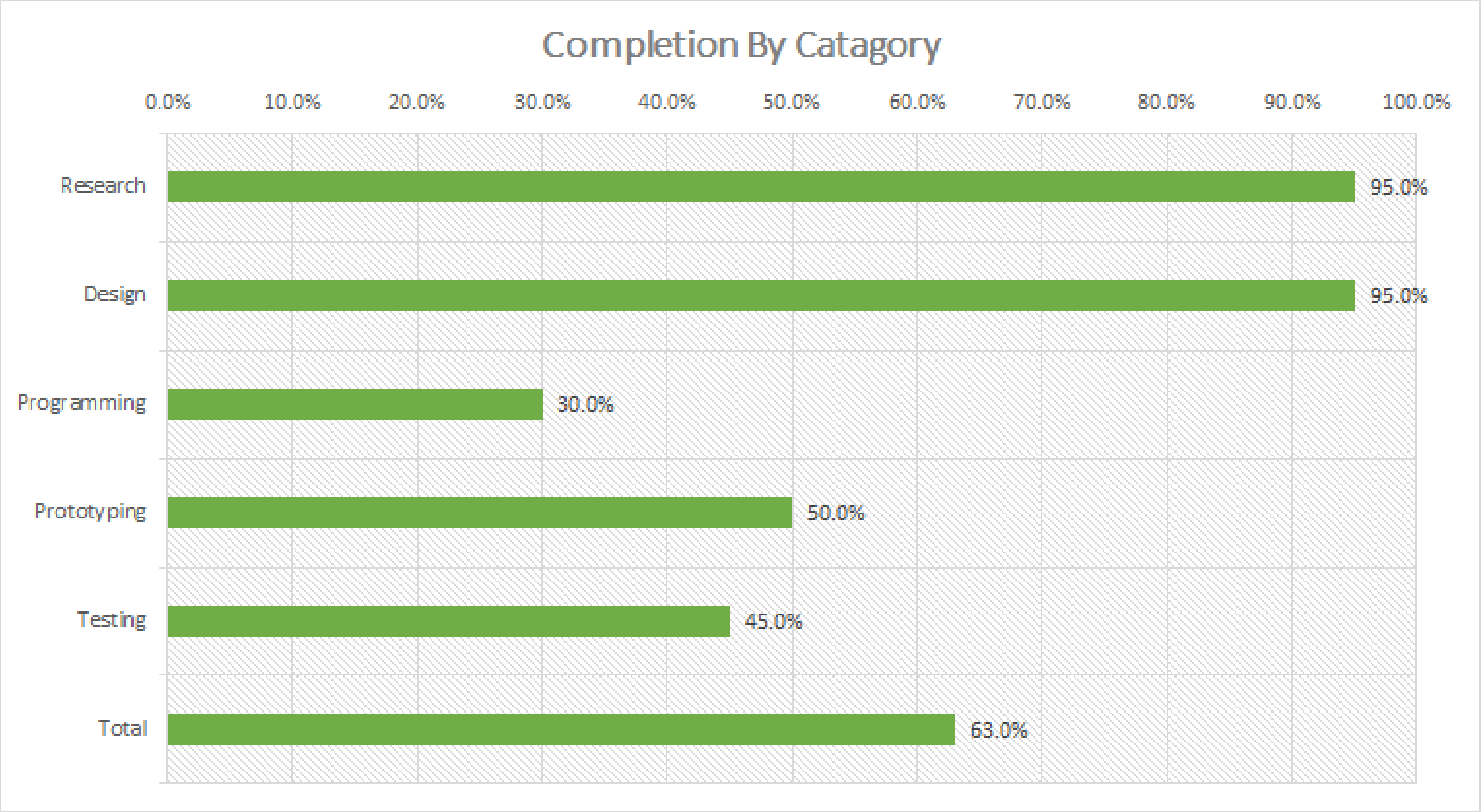
Issue: Power consumption was high due to valves.
Solution: Review of problem showed us that running of the grid would be preferable and battery power would not be necessary.

Issue: Soil sensor shipping took much longer then expected.
Solution: Work in other parts of the project.

Issue: Available documentation for CC3220SF is not beginner friendly.
Solution: A lot of time spent on researching for answers.

Budget & Financing

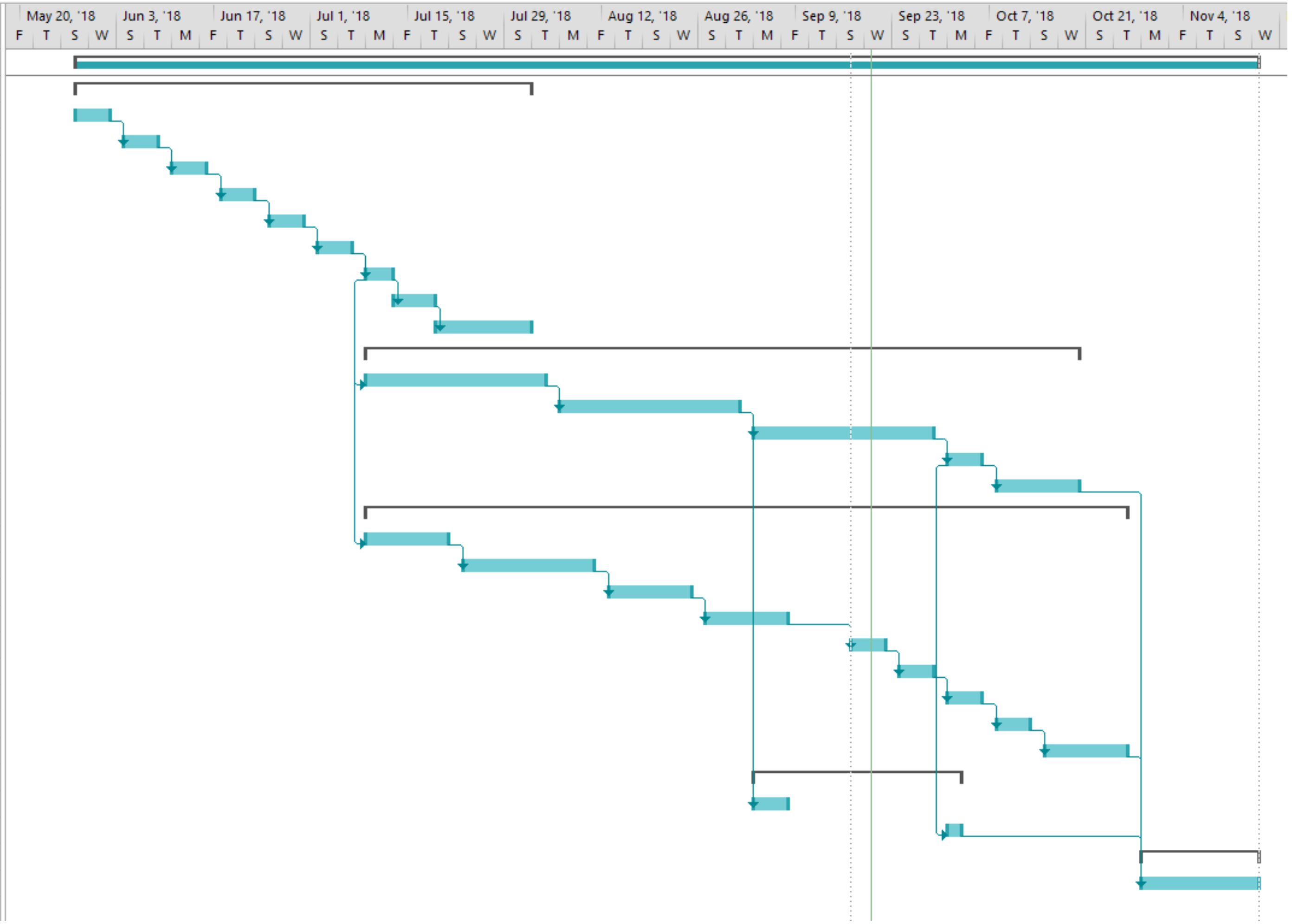
Description	Unit Cost	Project Parts Count	Project Cost Estimate	Final Product Parts Count	Final Product Cost
MCU/Wi-Fi Module	$10.00	3	$30.00	1	$10.00
Anemometer	$44.95	2	$89.90	1	$44.95
Humidity & Pressure sensor	$16.62	3	$49.86	1	$16.62
Moisture, Light, & Temperature sensor	$12.49	5	$62.45	3	$37.47
Water Valves Main	$24.95	3	$74.85	3	$74.85
Water Valves Backup	$6.95	3	$20.85	0	$0.00
PCB	~$30.00	3	~$90.00	1	~$30.00
LCD screen	$24.90	3	$74.70	1	$24.90
Chassis Enclosure	~$1.00	3	~$3.00	1	~$1.00
Power Source	~$10.00	2	~$20.00	1	~$10.00
Miscellaneous (hoses, sprinklers, garden content)	~$100.00	1	$100.00	0	$0.00
Total Cost			$665.61		$249.79



Completion By Catagory
0.0%
10.0%
20.0%
30.0%
40.0%
50.0%
60.0%
70.0%
80.0%
90.0%
100.0%
Research
95.0%
Design
95.0%
Programming
30.0%
Prototyping
50.0%
Testing
45.0%
Total
63.0%

Completion Schedule

Task Name	Duration	Start	Finish
Smart Home Garden Senior Design Project	**123 days**	**Mon 5/28/18**	**Wed 11/14/18**
System Design	**48 days**	**Mon 5/28/18**	**Wed 8/1/18**
Assign Groups	5 days	Mon 5/28/18	Fri 6/1/18
Select Project Idea	5 days	Mon 6/4/18	Fri 6/8/18
Research existing technologies	5 days	Mon 6/11/18	Fri 6/15/18
Requirements definition	5 days	Mon 6/18/18	Fri 6/22/18
Initial parts selection	5 days	Mon 6/25/18	Fri 6/29/18
Block Diagram creation	5 days	Mon 7/2/18	Fri 7/6/18
High Level System Design	4 days	Mon 7/9/18	Thu 7/12/18
Detailed system design	4 days	Fri 7/13/18	Wed 7/18/18
Part sourcing and allocation	10 days	Thu 7/19/18	Wed 8/1/18
Hardware Design	**75 days**	**Mon 7/9/18**	**Fri 10/19/18**
Schematic Capture	20 days	Mon 7/9/18	Fri 8/3/18
PCB Layout	20 days	Mon 8/6/18	Fri 8/31/18
Circuit Card Assembly	20 days	Mon 9/3/18	Fri 9/28/18
Board Bring up	5 days	Mon 10/1/18	Fri 10/5/18
HW SW Integration	10 days	Mon 10/8/18	Fri 10/19/18
Software Design	**80 days**	**Mon 7/9/18**	**Fri 10/26/18**
Software architecture	10 days	Mon 7/9/18	Fri 7/20/18
Initial Software setup	15 days	Mon 7/23/18	Fri 8/10/18
Web Server Implementation	10 days	Mon 8/13/18	Fri 8/24/18
Database Implementation	10 days	Mon 8/27/18	Fri 9/7/18
Microcontroller configuration	5 days	Mon 9/17/18	Fri 9/21/18
WiFi Interface	5 days	Mon 9/24/18	Fri 9/28/18
LCD Interface	5 days	Mon 10/1/18	Fri 10/5/18
Sensor Interface	5 days	Mon 10/8/18	Fri 10/12/18
Web Application	10 days	Mon 10/15/18	Fri 10/26/18
Mechanical Design	**22 days**	**Mon 9/3/18**	**Tue 10/2/18**
Chassis Design	5 days	Mon 9/3/18	Fri 9/7/18
Board Fit Checks	2 days	Mon 10/1/18	Tue 10/2/18
Testing	**13 days?**	**Mon 10/29/18**	**Wed 11/14/18**
Fully Integrated System Testing	13 days	Mon 10/29/18	Wed 11/14/18



Immediate plans for successful completion

Programming:

- Finish server REST code.
- Finish web application & test with data from database.
- Finish water flow control logic & test with operating water valves.

Hardware:

- Build initial prototype with custom 3D chassis.
- Test sensors, water valves, and LCD interface for compatibility with PCB.

Q & A