# Smart Plant Watering System

Kaitlyn Pounds (1), Jillian To (3)

5/1/2023, Introduction to Embedded Systems

**Contribution Chart:**

| | Jillian To | Kaitlyn Pounds |
|---|---|---|
| Initial Research | X | X |
| Block Diagram and UML | X | |
| PCB Design | | X |
| Soldering | | X |
| Coding | X | |
| Analyzing Results | X | X |
| Research Related to Project that Helps Report Writing | X | X |
| Writing Report | X | X |
| Making Presentation Slides | X | X |

**Abstract/Goal:**

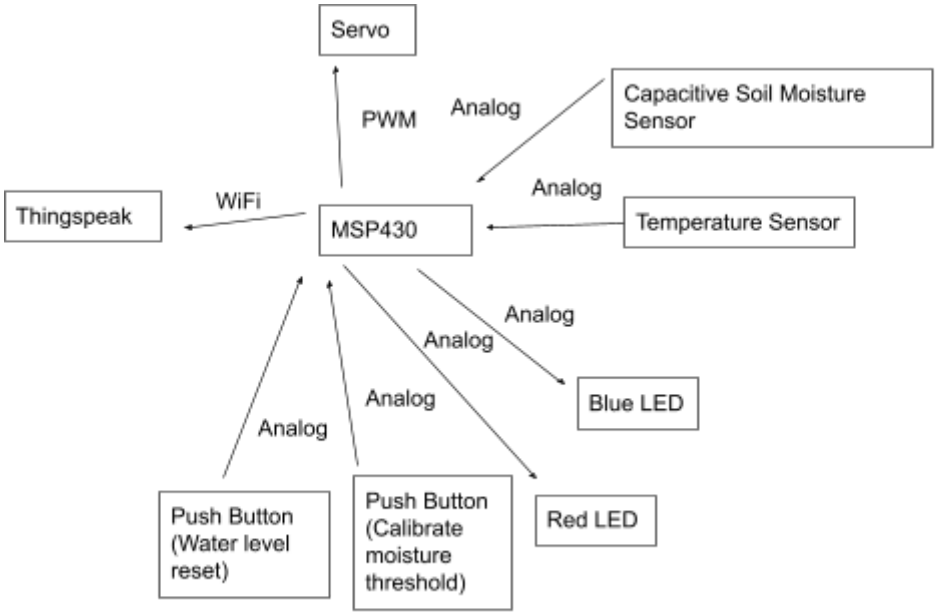
For those living in areas of the world with very little or no rain it can be difficult to care for a plant, especially for the ones that are picky about their living conditions. As a solution to this issue, we propose a smart plant watering system. It will monitor the moisture level of the soil that the plant is in, and if the level is too low it will add water to the soil until the moisture is above a specified acceptabled threshold. Users can also view their plant's moisture level over the internet. The objective of our project is to use technology to make more efficient use of water resources, as well as making life easier for plant owners. The moisture sensor we are using is much more accurate at reading soil moisture than a human can be, so it only makes sense to automate the process. This technology will greatly reduce the overwatering of crops, benefiting the environment, since less water is wasted, as well as saving money on water bills both for the average person and businesses.

**Background/Introduction:**

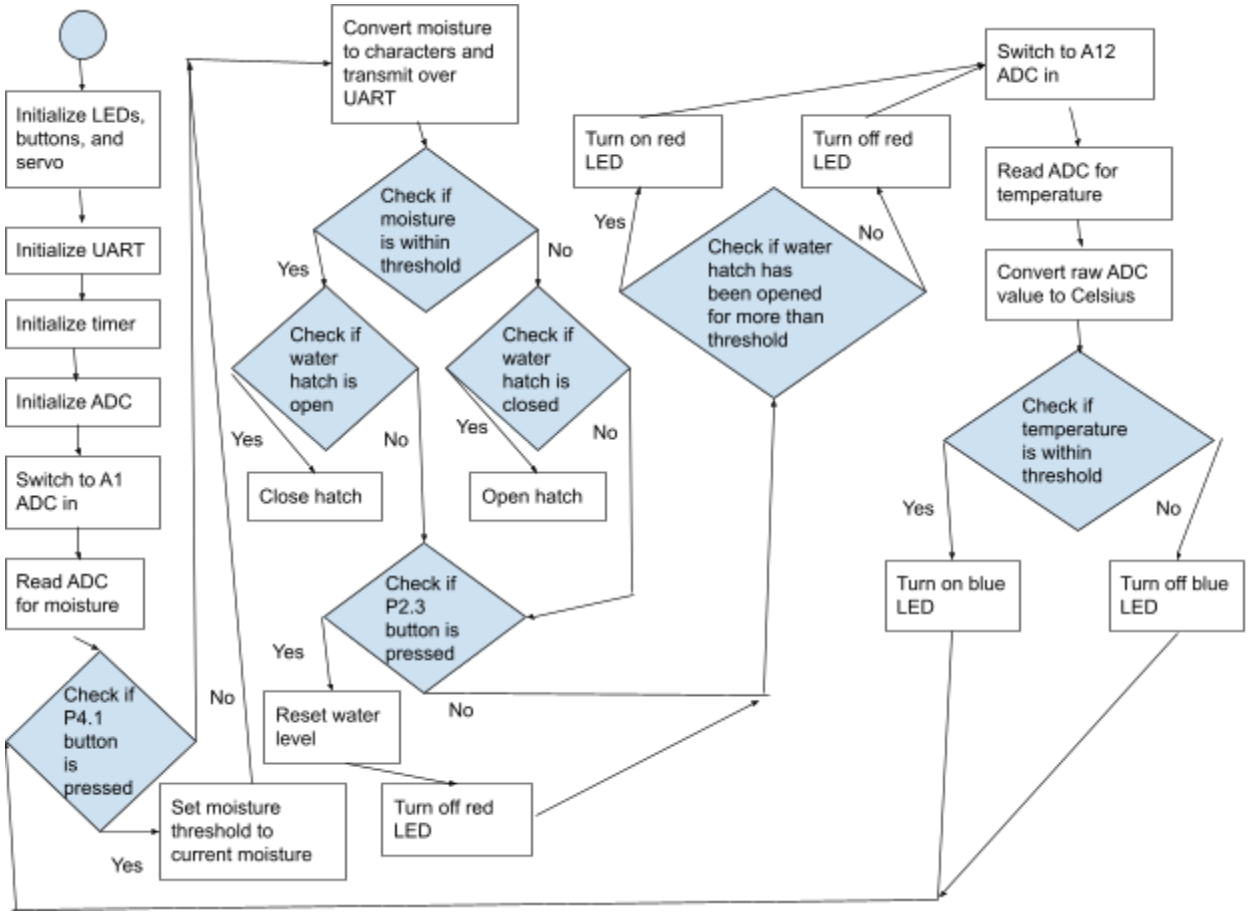
Our initial step for this project was to do research into the things we were unfamiliar with. This included PCB design, C programming, and embedded communication interfaces, such as UART. We first figured out how to make each sensor work individually, then later integrated everything together. We also tried our system with plants we had at home to optimize water use and determine typical moisture levels.

**Methodology and System Design:**

- ✔ Functional block diagram



✔ UML Activity Diagram



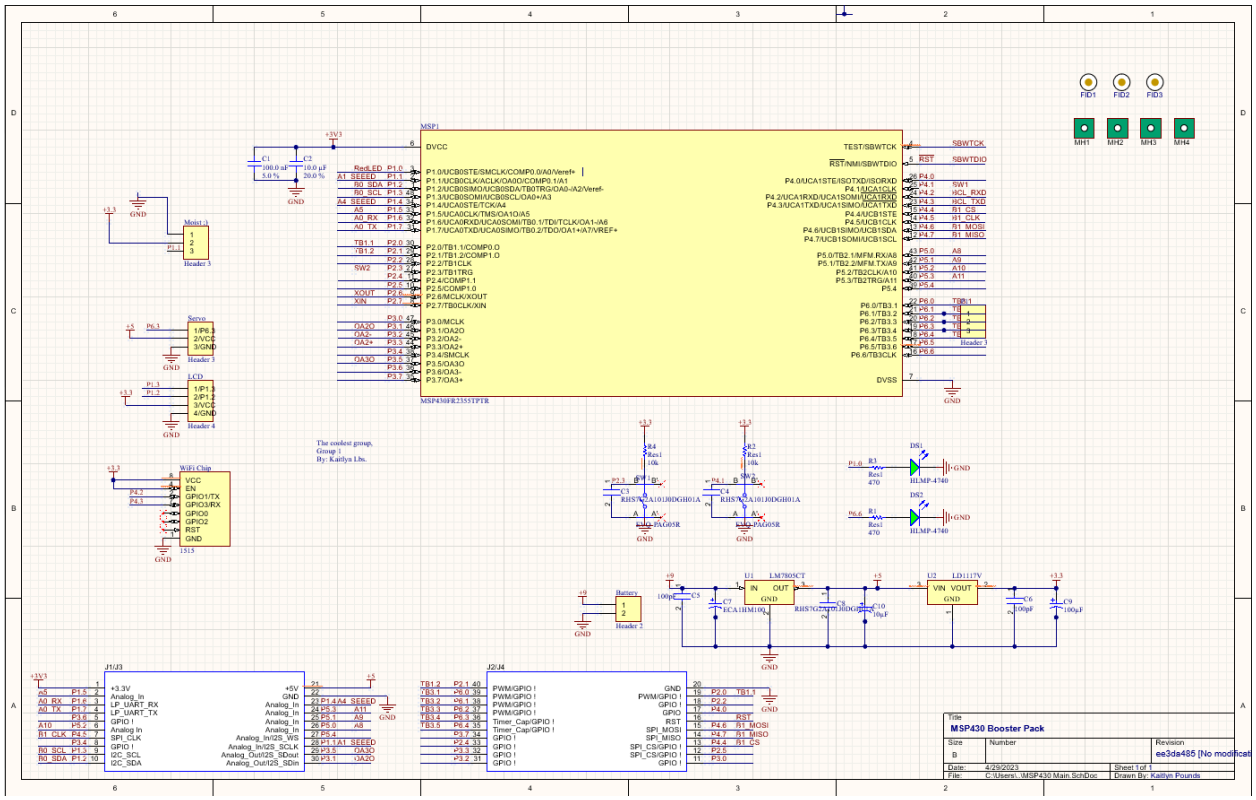

✔ Parts List and Design cost

| **Part** | **Cost ($)** |
|---|---|
| PCB | 5 |
| PCB components (resistors, capacitors, etc.) | 0 |
| Capacitive soil moisture sensor | 0 |
| Servo | 0 |
| Red LED | 0 |
| Blue LED | 0 |

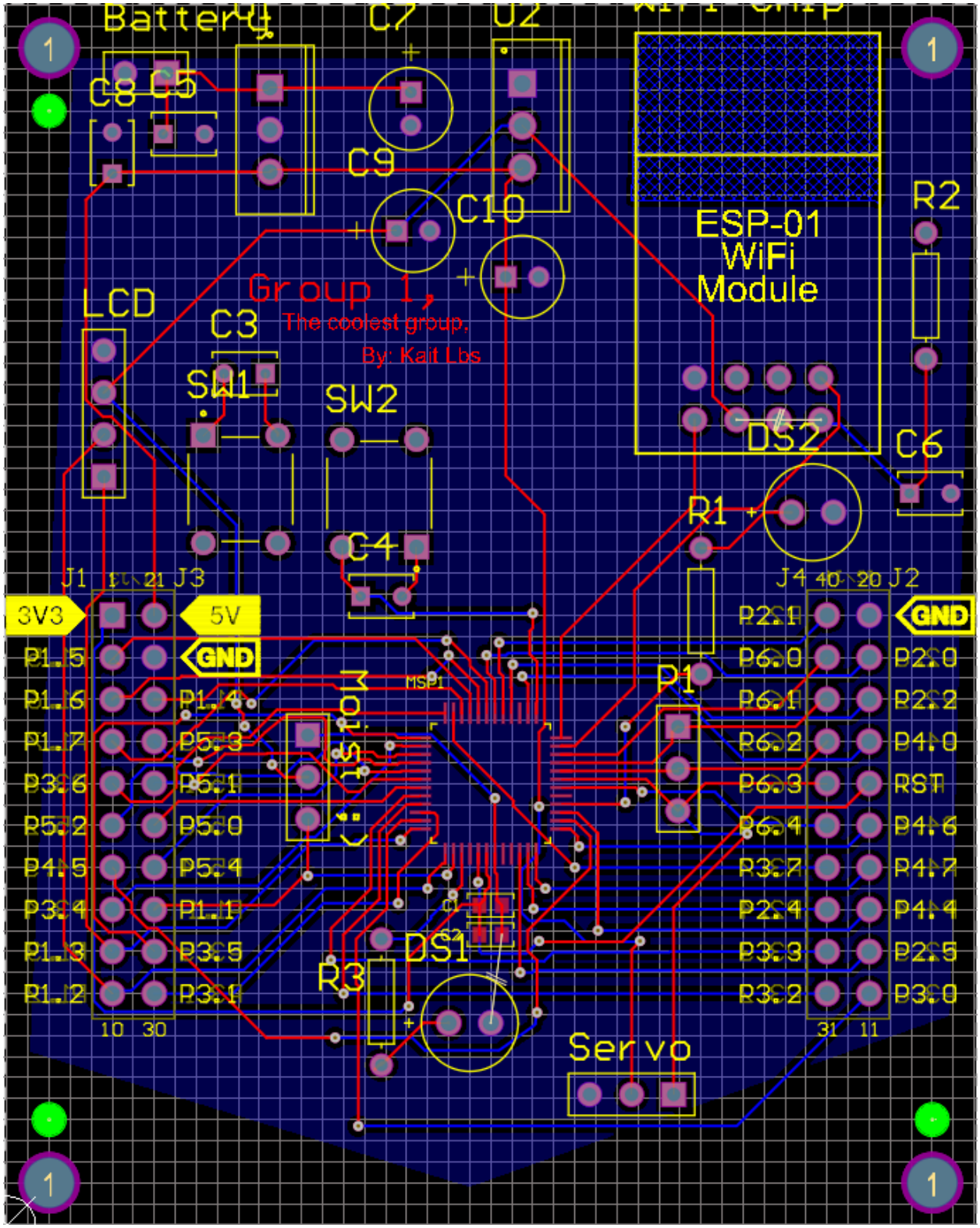
| | |
|---|---|
| Push buttons (2x) | 0 |
| Temperature sensor | 0 |
| ESP-01 WiFi module | 0 |
| Total | 5 |

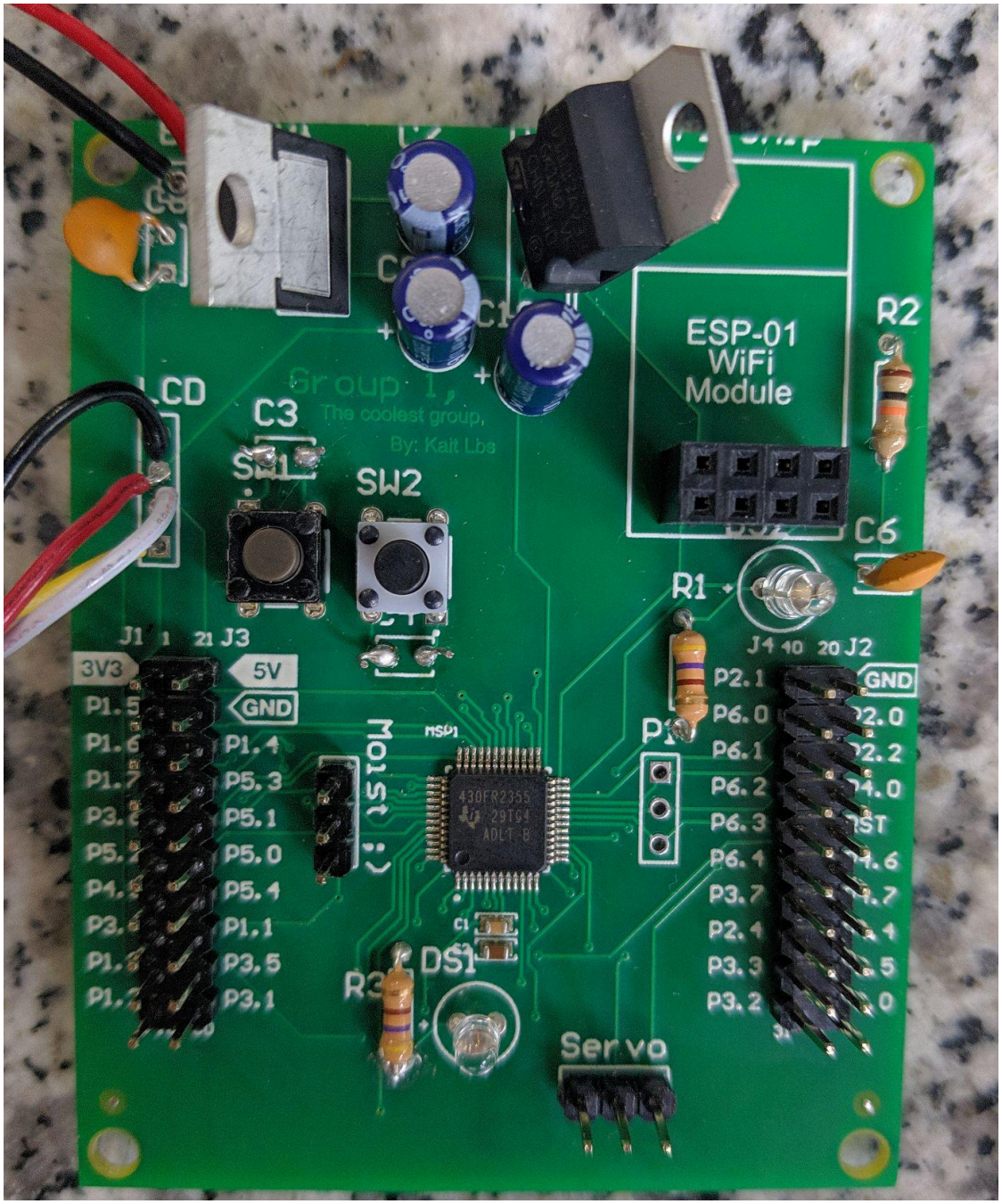
## Experimental Set Up and Results:

- ✔ Schematic:




- ✔ Soldered PCB connection with MSP430:

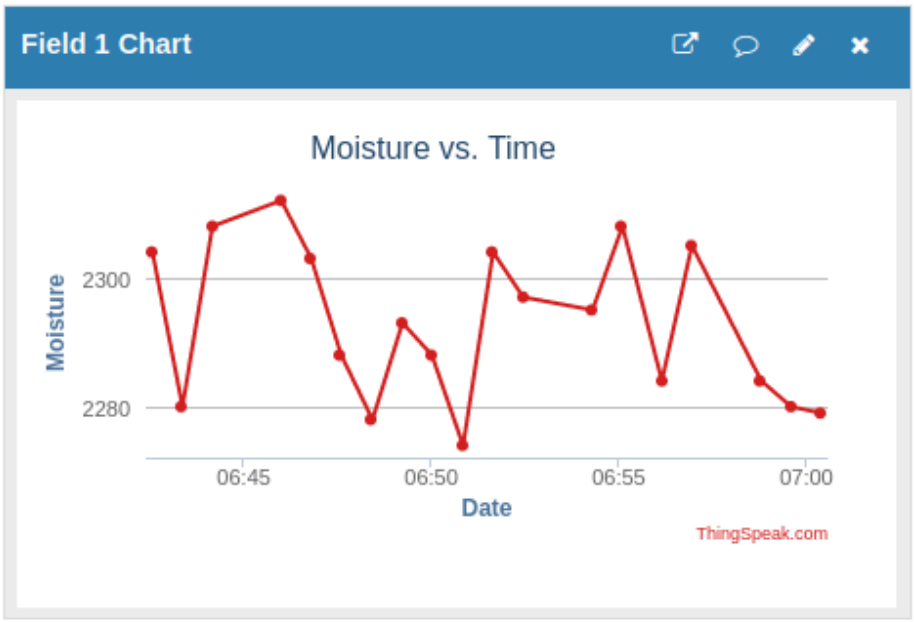

Battery
C7
U2
C8
C5
C9
C10
ESP-01
WiFi
Module
R2
LCD
C3
Group 1,
The coolest group.
By: Kait Lbs
SW1
SW2
DS2
C6
R1
C4
J1
J3
J4
J2
3V3
5V
GND
P1.5
P1.6
P1.7
P3.6
P5.2
P4.5
P3.4
P1.3
P1.2
P1.4
P5.3
P5.1
P5.0
P5.4
P1.1
P3.5
P3.1
P2.1
P6.0
P6.1
P6.2
P6.3
P6.4
P3.7
P2.4
P3.3
P3.2
P2.0
P2.2
P4.0
RST
P4.6
P4.7
P4.4
P2.5
P3.0
P1
MOSI
DS1
R3
Servo
10
30
31
11
40
20
1
21


ESP-01
WiFi
Module
LCD
C3
SW1
SW2
C6
R1
R2
J1
J3
J4
J2
3V3
5V
GND
P1.5
P1.6
P1.7
P1.4
P5.3
P5.1
P5.0
P5.4
P1.1
P3.5
P3.1
Moist :)
MSP1
430FR2355
29TG4
ADLT B
C1
DS1
R3
Servo
P2.1
P6.0
P6.1
P6.2
P6.3
P6.4
P3.7
P2.4
P3.3
P3.2
GND
P2.0
P2.2
P4.0
RST
P4.6
P4.7
P4.1
P4.5
P4.0

✔ Results from ThingSpeak

## Channel Stats

Created: about a month ago
Last entry: less than a minute ago
Entries: 130




**Conclusion/Future Work**

Although there are many more features that could have been added to our project, it is still able to complete the basic functions needed to perform its intended function, to monitor the moisture level of plant soil and to add water when necessary. By incorporating multiple sensors and peripherals together we were able to create an automated plant watering system. In the future, to make a more appealing commercial product, we would add some piezo buzzers to audibly alert the user, a fan to help regulate the plant's temperature, a LCD so the user can view measured information locally instead of only through Thinkspeak, as well as a 3D printed casing to hold everything. Our program could also be optimized to use less power. We did try to utilize timers and interrupts instead of polling, but with the many components, we ran into problems with ISR traps when needing to use many interrupts and delays. We would also hope to have a better-designed PCB, as not everything was done correctly on the board itself. For example, the MSP430 Launchpad does not fit on top of our PCB when the moisture sensor or servo are

plugged in. A refined version of our project would certainly be worthy of selling to customers as an actual product, and it would prove to be very useful.
However, even with the many features we did not include, the features that we did include were a great learning experience and helped us understand the complexities of implementing an embedded system. For one, we can better appreciate higher-level programming languages, like Arduino programming, since we had to learn first-hand how much technical knowledge about both the sensor and the board are needed to implement the libraries that we used to take for granted. Also, we learned how to balance performance and power consumption by managing clock speeds and sampling rates. Finally, we were able to design and build a working PCB, which is essential knowledge for an electrical engineer.
In conclusion, with how many smart appliances have become commonplace in our world, this project helped us learn to appreciate all that goes into incorporating electronics into our everyday lives.

**Appendix**

Link to code in GitHub:
https://github.com/JillianTo/PoundsTo---EmbeddedFinal-Project---WateringSystem