The Fertilizer Frontier

The Creation of a Solar Powered and Fully Automated Dual Tank Fertilizer Mixing System for Reduced Environmental Impact

Design/User's Manual

An Interactive Qualifying Project (IQP) submitted to the faculty of
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the degree of Bachelor of Science

Submitted By:
Max Berman
Elie DeLaVille
Maggie Kirwan
Gabriel Montalvo Reyes

Submission Date:
February 28, 2024

Submitted to:

Project Advisors
Professor Isa Bar-On
Professor Michele Feme-Bagwell

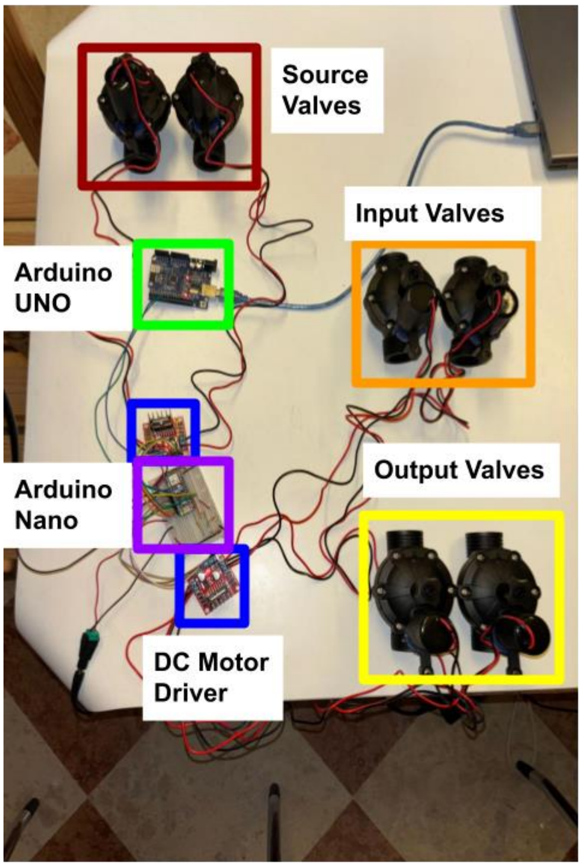
Project Sponsor
Gai Boger (גיא בוגר) and NESS Fertigation

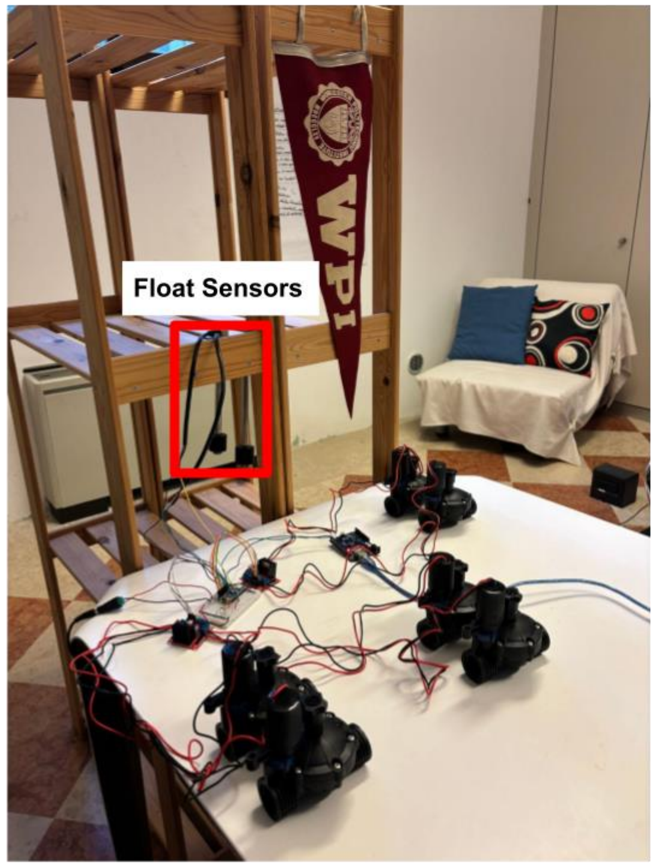
Table of Contents:

Part I: Design Section

1.1 System Design

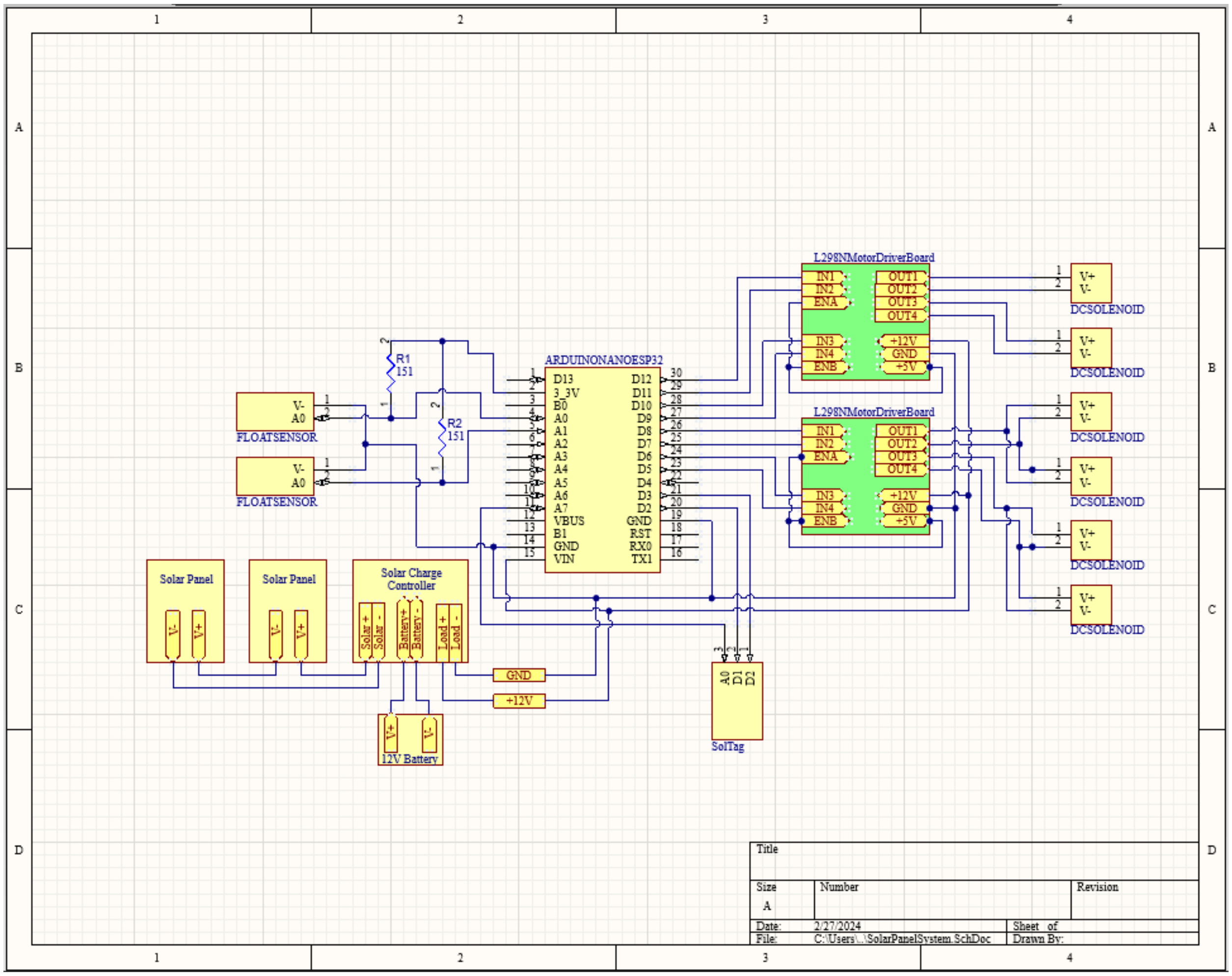
The images below display a top-down view and isometric of the system's final design.



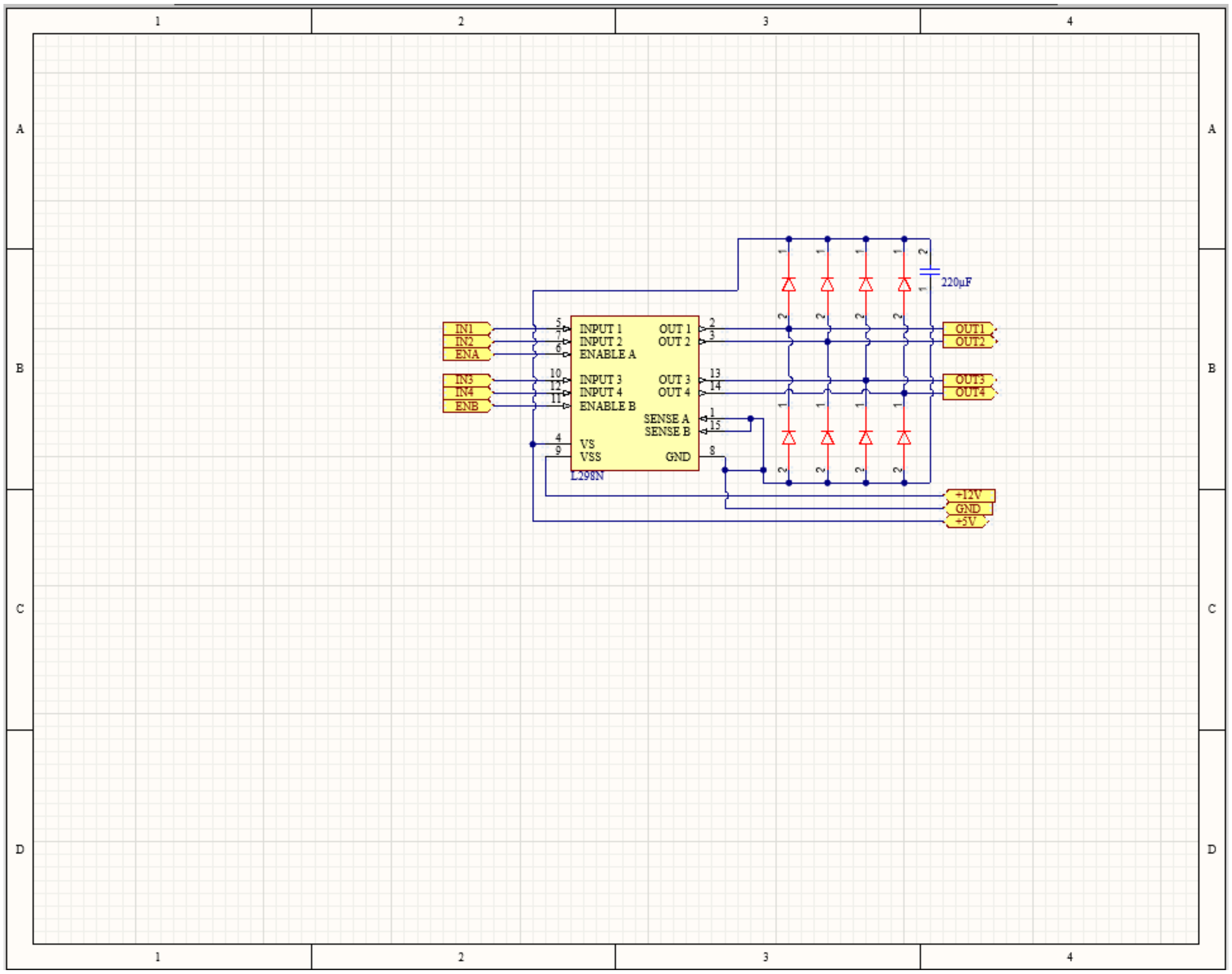


1.2 Schematic

The schematic of the full prototype is shown below including the necessary solar panel system components.



The schematic below shows the internal wiring of the L298N Motor Driver Board



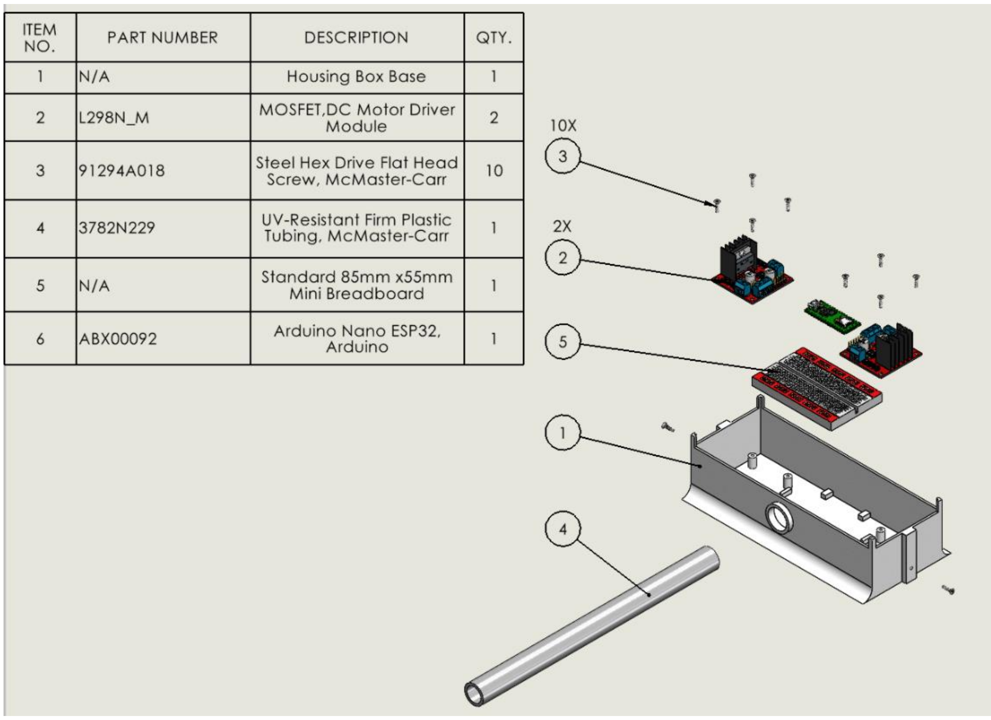
1.3 Arduino Nano ESP 32 Data Sheet

The Arduino Nano ESP 32 is linked in the data sheet below:

https://docs.arduino.cc/resources/datasheets/ABX00083-datasheet.pdf

1.4 Bill of Materials

The image below shows the materials used in the design, referenced in numbers 1-6. This includes component names, part numbers and quantity.



ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	N/A	Housing Box Base	1
2	L298N_M	MOSFET,DC Motor Driver Module	2
3	91294A018	Steel Hex Drive Flat Head Screw, McMaster-Carr	10
4	3782N229	UV-Resistant Firm Plastic Tubing, McMaster-Carr	1
5	N/A	Standard 85mm x55mm Mini Breadboard	1
6	ABX00092	Arduino Nano ESP32, Arduino	1

1.5 Solar Panel System Calculation

While assuming 12 hours of activity per day, the final power consumption of the prototype came out to 44.1262 Wh. We used the following equations to calculate the solar panel system capacity.

$$P_{Daily\ Consumption} \,/\, \eta_{System} = P_{Produced\ Daily}\ [Wh]$$

$$\frac{P_{Produced\ Daily}}{Average\ Daily\ Sunlight\ Hours} = Theoretical\ Capacity_{\ Solar\ Panel}\ [W]$$

$$Theoretical\ Capacity_{Solar\ Panel} \,/\, k = Actual\ Capacity_{Actual\ Solar\ Panel}\ [W]$$

$$Actual\ Capacity_{Actual\ Solar\ Panel} \,/ E_{One\ Panel} = Number\ of\ Panels$$

The completed equations with values are found below.

$$44.1262\ Wh \,/\, 0.20 = 220.631\ Wh$$

$$220.631\ Wh \,/\, 10.1667h = \ 21.701\ W$$

$$21.701\ W \,/\, 0.80 = 27.1267\ W$$

The final solar system capacity assuming a 20% system efficiency would be 27.1267W.

$$27.1267W \;/\; 30W = Number\ of\ Panels$$

Using a 30W solar panel, the total number of panels would be 0.904. Therefore, only one 30W solar panel is required to power the system.

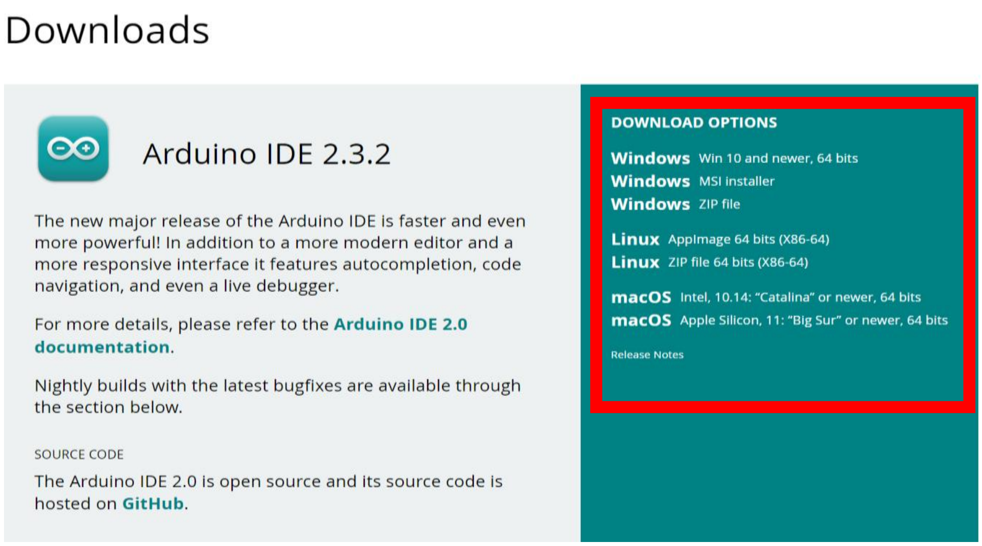
1.6 Link Code

https://github.com/maximus-lazer/Fertigation-Mixing-Controller

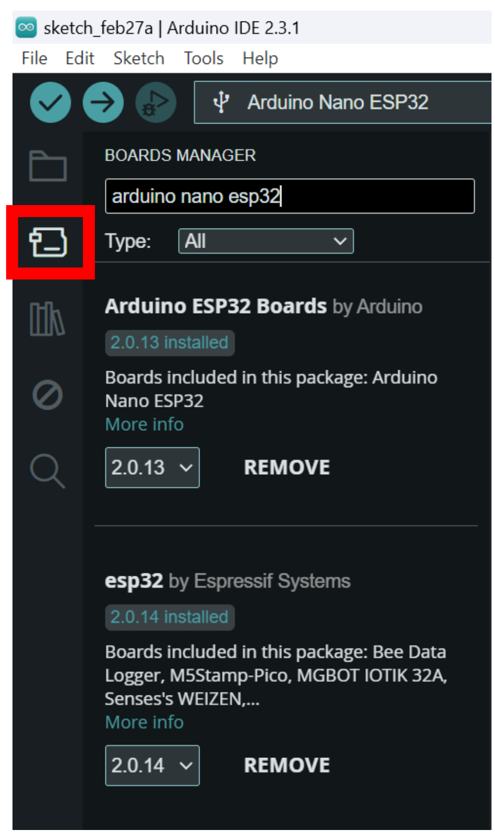
Part II: Users Section

2.1 Controller Setup

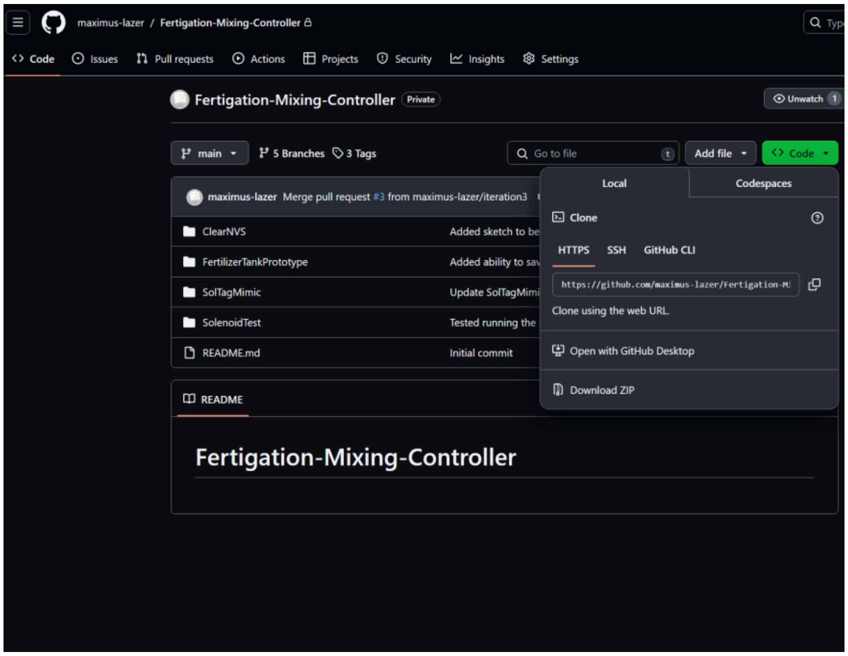
1. Download the Arduino IDE here for your machine: https://www.arduino.cc/en/software



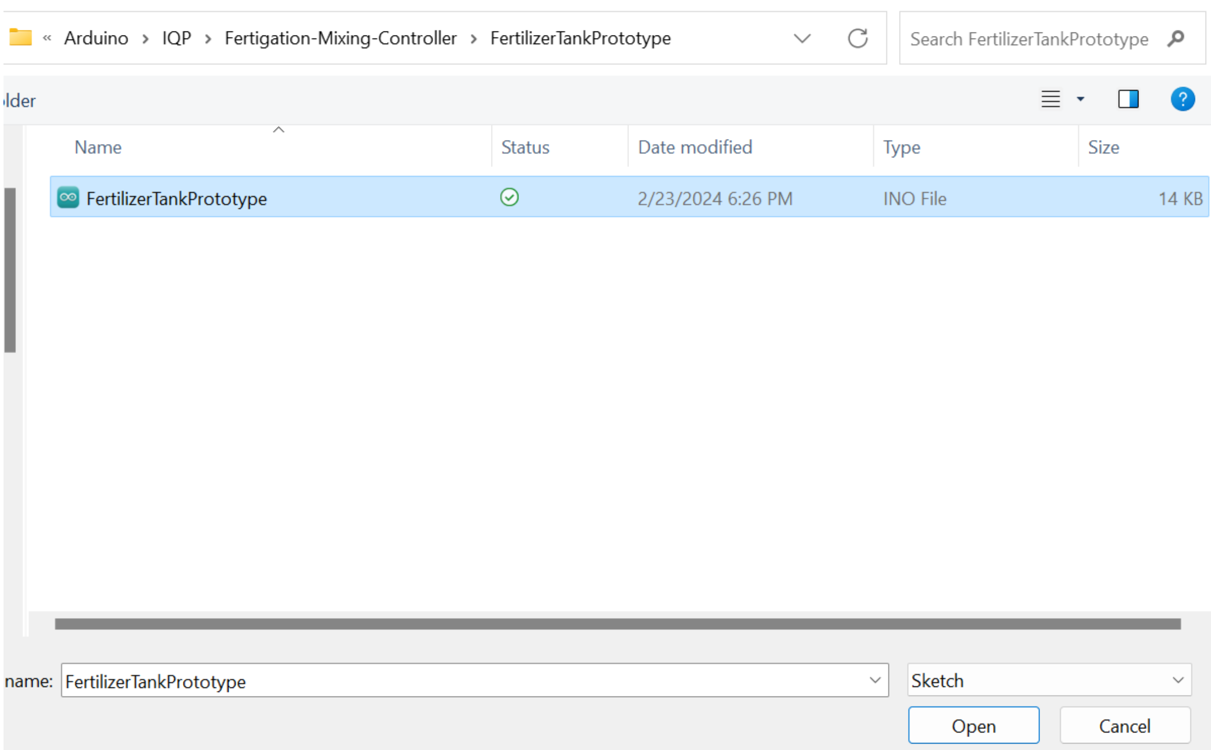
2. Click on "Board Manager" tab and install "Arduino ESP32 Boards"



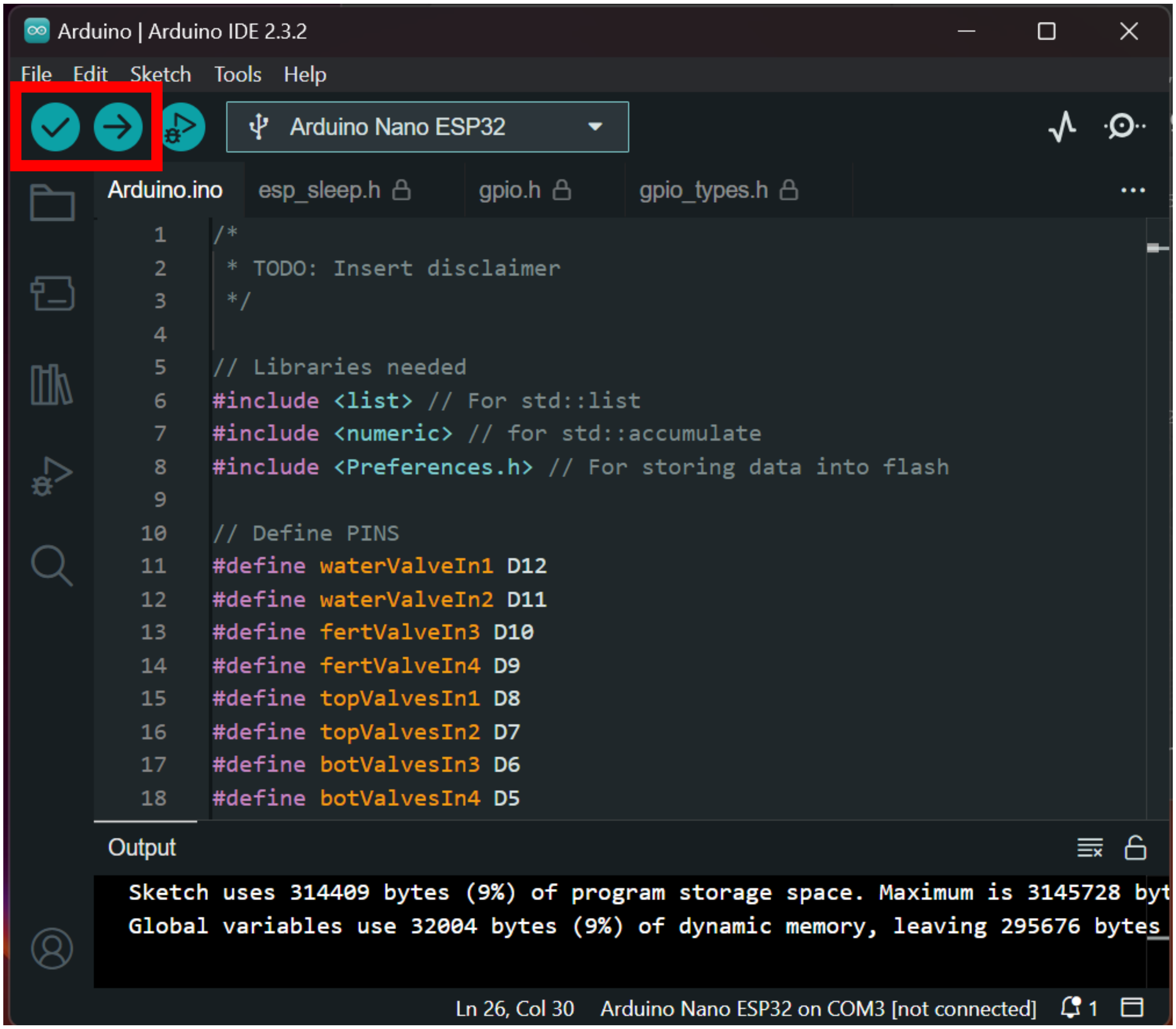
3. Download the repository from the GitHub link in Section 1.5 of this manual.
    a. Click on "Code" to open the dropdown menu
    b. Either clone the repository using the link or download the full code ZIP folder



4. In the "Fertilizer-Mixing-Controller-main" folder, open the "FertilizerTankPrototype" subfolder. The file FertilizerTank.ino is the code that is running on the Arduino Nano ESP 32. Open this file in the Arduino IDE.



5. Compile the code by pressing the button on the left and upload the code by pressing the button on the right.



6. To clear the non-volatile storage where the fertilizer percentage is saved, run the ClearNVS.ino file in the "ClearNVS" subfolder of the "Fertilizer-Mixing-Controller-main" folder.
    a. Be sure to only run this program once, and reupload the FertilzerTankPrototype.ino file after completion. Otherwise, the Arduino will continue to erase the non-volatile memory upon every startup.

2.2 SolTag Connections

The SolTag controller connects to the Arduino Nano ESP32 with two digital I/Os and one analog pin. The digital I/Os dictate the activation and deactivation of the system and the fertilizer percentage in the tanks.

2.2.1 Digital I/O Information sent from SolTag

The Arduino Nano ESP32 wakes up when either of these bits turns high. If the fertilizer bit is high, the Arduino goes into the “FERT_INPUT” state and records this bit every two minutes to determine the fertilizer percentage. Once the bit goes low, the Arduino either goes back to sleep or starts depending on if the start bit is high or low. While the start/stop bit is high, the system continues to run, emptying the fertilizer and water mixture into the field. When the bit is set low, the system keeps running until one tank is full and one is empty, and then goes into deep sleep mode.

2.2.2 How to Read Error Messages

The Arduino Nano ESP32 sends error messages over the analog connection with the SolTag controller when it is in the ACTIVE state. The Arduino sends a value of 1.0V over the analog pin when there are no errors and different voltages when an error occurs. The list of error messages and their corresponding voltages can be found in the table below along with the approximate value read from the analog pin.

Error Type	Voltage (V)	Approximate Analog Value (10-bit, 3.3Vref)
Heartbeat	1	310
General Error Message	1.33	412
Top Valve Set Error	1.67	517
Bottom Valve Set Error	2.0	620
Source Valve Set/Pump Error	2.33	722
Left Float Sensor Broken	2.67	827
Right Float Sensor Broken	3.0	930