Farm Bureau Watershed Project

Christopher Zakrevski, Noah Holloway, and Jacob Horsley

Idaho State University

RCET3375

Prepared for: Tim Rossiter

Date*: 12/18/25*

Authors' Note

We would like to thank our instructors for their advice, guidance, and constant help during the never-ending hunt for parts used in our project.

Table of contents

Table of figures

Abstract

This capstone project is a scaled, interactive physical model designed to simulate the dynamics of a mountain watershed impacted by seasonal snow melt, clearly demonstrating the chain of events leading from glacial melting to potential downstream flooding. During the simulation cycle water is pumped from a hidden reservoir to the top of a styrofoam and plastic mountain structure, where it simulates accelerated snow melt during warmer conditions. Increased runoff flows into a downstream reservoir; upon reaching capacity—detected by water level sensors—a PIC16F1788 microcontroller triggers a servo motor to open spillway gates, releasing water to dramatically flood a model town on the plain below. At the end of the cycle a solenoid-driven valve at the low point of the flood plane is opened to evacuate water back to the main tank, thus completing the process cycle. The process is pre-programmed and runs automatically, and allows for user interaction and control via a combination of physical buttons and an Arduino Mega with touchscreen interface. The model visually illustrates key hydrology concepts, including rising runoff volumes overwhelming reservoir storage during peak melt seasons, dam safety challenges, and flash flood risks to communities. Designed using a combination of off-the-shelf and custom designed components, the demonstration serves as an engaging outreach tool for schools, museums, and science fairs, making seasonal environmental processes tangible and fostering discussion on water resource management.

Background

The project began when the Farm Bureau approached Idaho State University with a proposal to develop a portable, functional watershed model for demonstrations at local schools.

The purpose of this model is to:

1. Visually connect rising temperatures → accelerated ice/snow melt → increased runoff volumes.
2. Illustrate how reservoirs normally store water but can reach capacity during extreme melt events.
3. Dramatically show the consequences of reservoir overflow (or controlled release via spillway) on a downstream model community, emphasizing flash flood risks.

Our team was tasked with designing and building a prototype of a watershed system that includes a mountainside, reservoir, spill gate, and floodplain. The model combines physical landscape construction with integrated electronics to simulate and control water flow and system behavior.

Proposed Problem

Can a student team, working within constraints of regular curriculum workload, create a working model that meets the requirements set by the Farm Bureau?

Constraints

1. No prior work available – The project began with no existing designs or reference material, requiring development entirely from scratch.
2. Short timeline – Only four weeks were available to complete the project while balancing other course assignments.
3. Limited part availability – Delivery constraints required maximizing the use of materials and components already available on hand.

Goals for our team

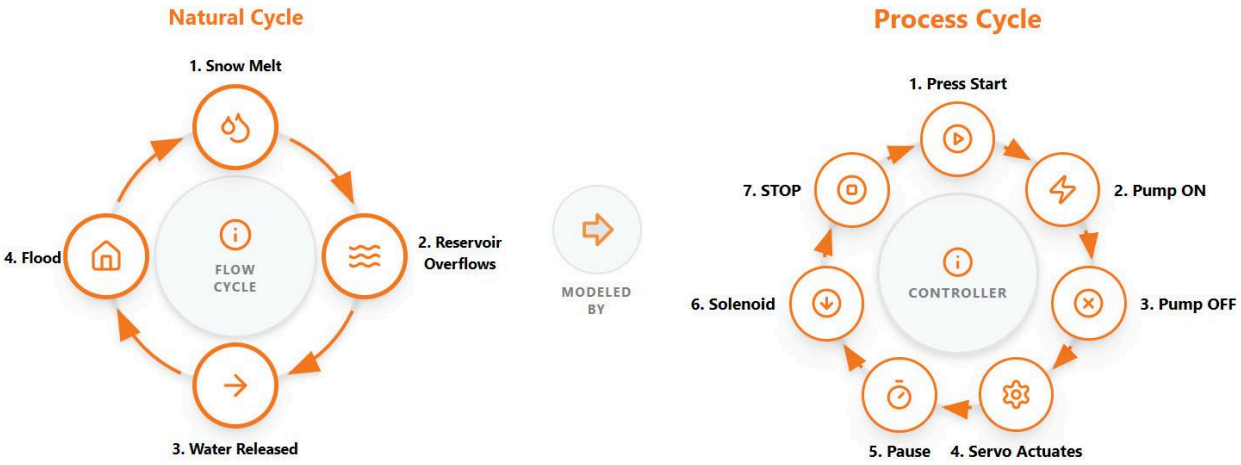
1. Develop a functional prototype that meets the project requirements by accurately representing the watershed system and demonstrating controlled water flow using integrated electronics.
2. Explore, evaluate and document design approaches for the electronics, programming, and physical model to determine the most efficient, reliable, and cost-effective solutions.
3. Gain hands-on experience through prototype development to inform the design and construction of a refined final version of the project.

Strategy

To meet the strict time requirements of this project our team understood that we need to work with a clearly defined strategy and project timeline. We achieved this by analyzing how the process can be simulated, then deciding on prototype methodology to adopt, and selecting a main goal for each of the four weeks.

Simulation

To simulate the natural cycle on our scaled model, we decided to use a system consisting of a microcontroller driving a water pump, servo motor and solenoid-driven valves.
The main steps of the natural phenomena were translated to sequential actions for our control system.



[Figure 1: Process representation of a natural cycle.]

Methodology

Our approach to prototyping was to aim for quick wins by utilizing materials available on hand to test our ideas, comparing the outcomes, and then selecting a solution that represents the best compromise between cost, reliability and the time required to implement.

Each decision was driven by considering both our immediate needs and also long term project goals.

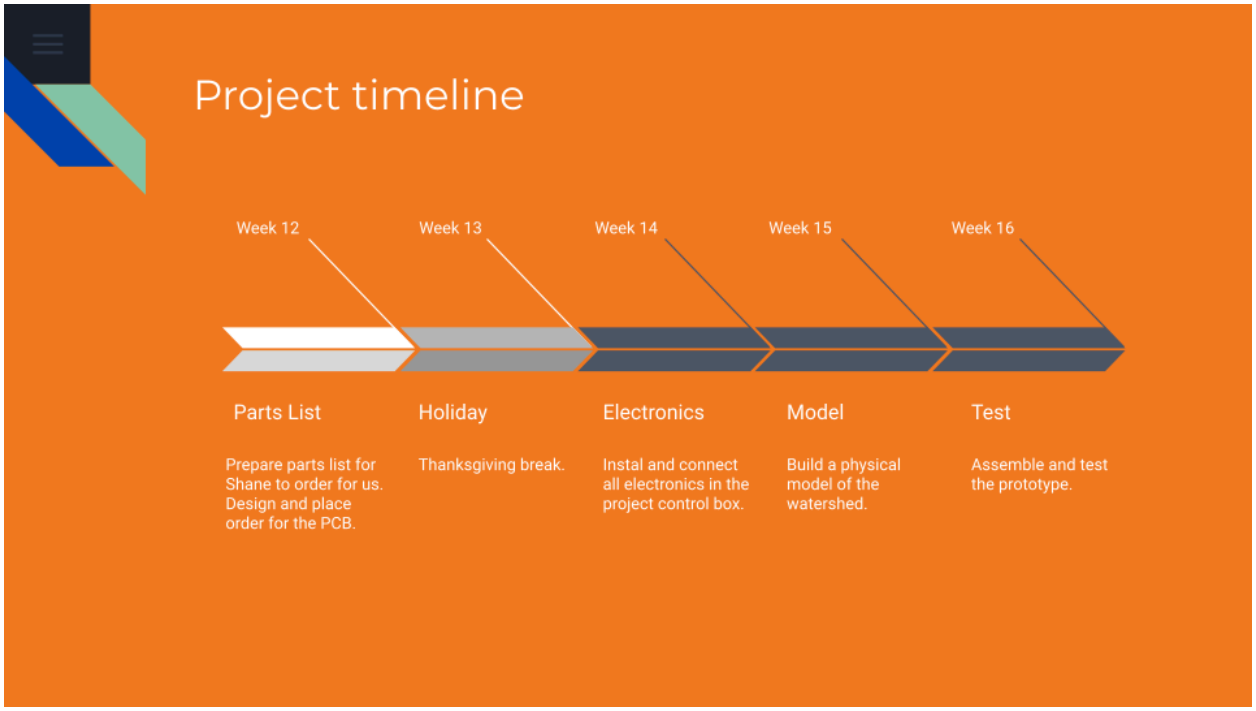
[Figure 2: Prototyping methodology]

Timeline

To ensure that we can finish the project on time we selected one main goal for each week and limited ourselves to only working on those goals.

Main goals for each week:

1. Week one (week 12 of the semester): Understand the process and order parts.
2. Week two (week 14 of the semester): Assemble electronics in the control box and test all actuators.
3. Week three (week 15 of the semester): Build the scaled model of the system.
4. Week four (week 16 of the semester): Assemble everything together and test the system operation.



[Figure 3: Project timeline]

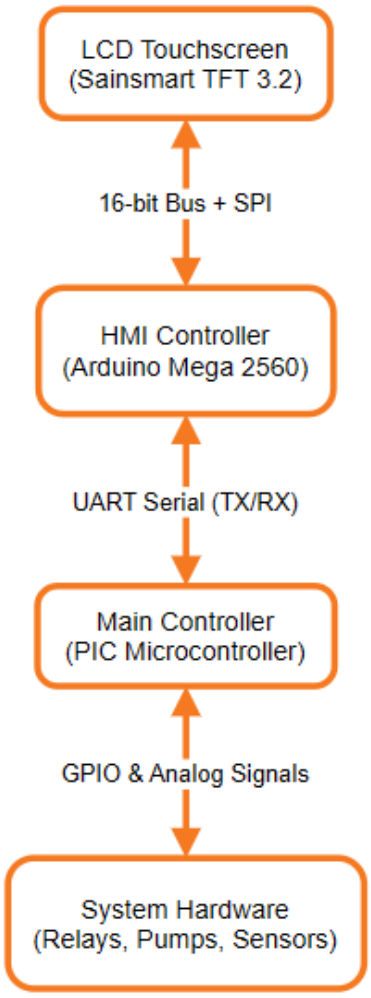
Implementation

In this section we have included technical details and block diagrams.

Arduino HMI

Hardware

- Sainsmart TFT_320QVT screen - mounted directly on Arduino Mega.
- Sainsmart Mega 2560 R3 ATmega2560-16AU.



[Figure 4: Arduino HMI hardware block diagram]

Pin connections:

- Arduino Pin 18 (TX1) → PIC RX Pin
- Arduino Pin 19 (RX1) → PIC TX PinGND Firmware

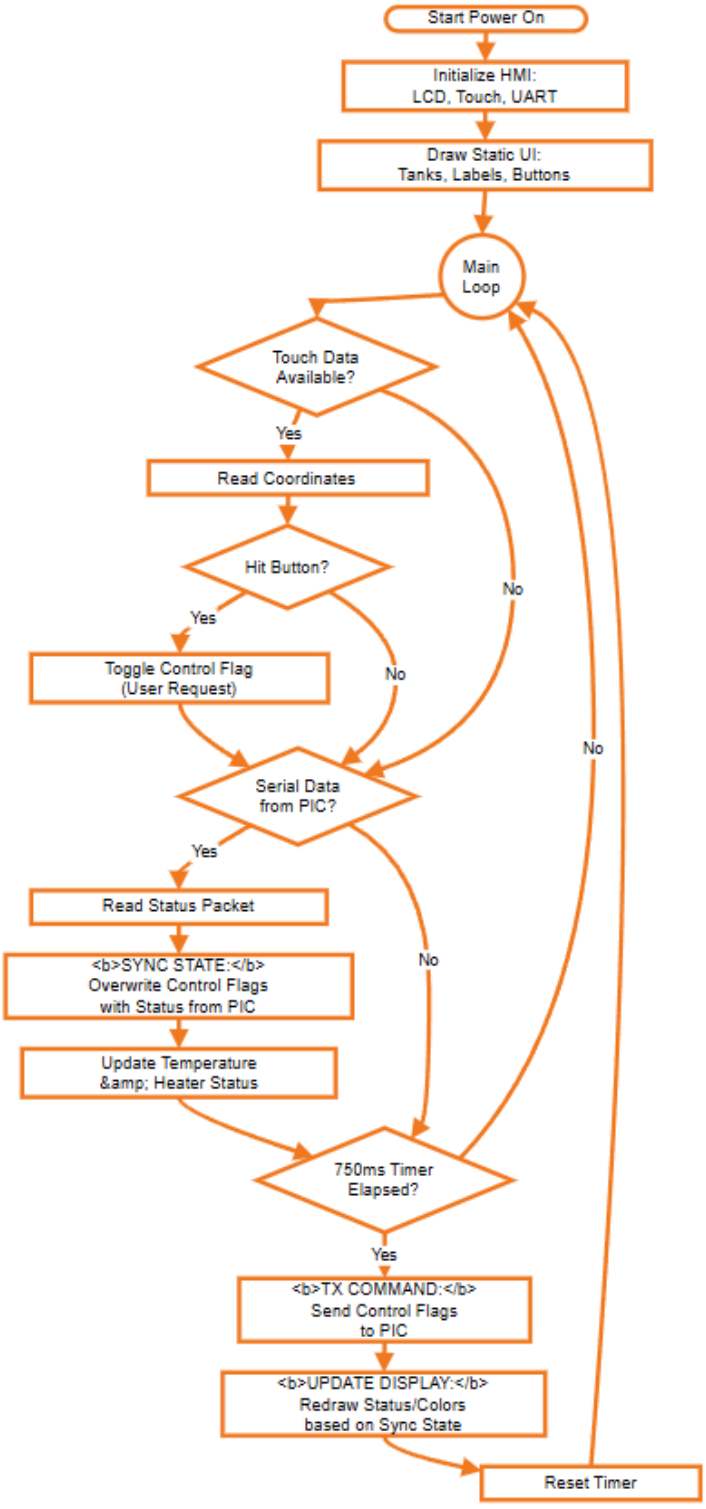
Firmware

The arduino code was written in Arduino IDE 2.3.6 with additional libraries to handle the TFT LCD operation.

Additional Libraries:

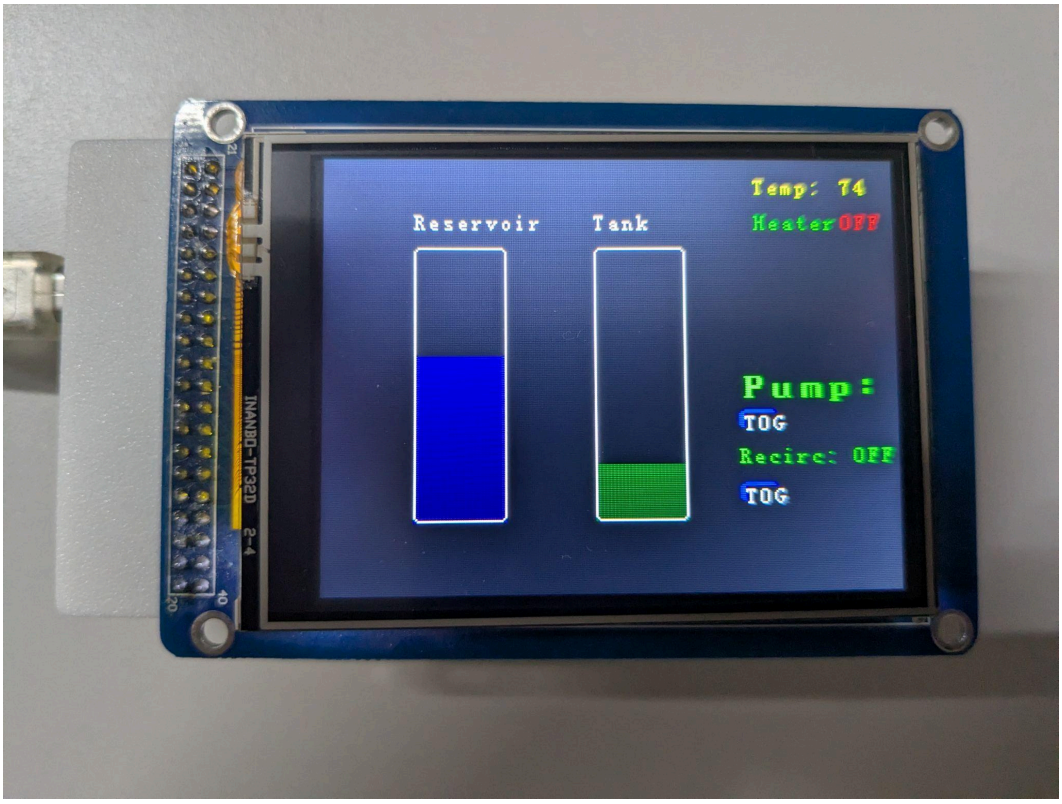
- #include <UTFT.h> : Handles low-level graphics drawing (lines, shapes, text) and communication with the display controller.
- #include <URTouch.h> : Handles the resistive touchscreen overlay. It translates raw analog pressure readings into X/Y coordinates.

The firmware is implemented in C++ for the Arduino Mega 2560, utilizing the UTFT and URTouch libraries to manage a 16-bit parallel graphical interface and an SPI-based resistive touch controller. To ensure high responsiveness without a Real-Time Operating System (RTOS), the application relies on a non-blocking, timer-based execution loop that concurrently polls for touch events and manages an asynchronous UART serial link on the hardware Serial1 port. A custom 16-byte binary protocol is employed for data exchange: incoming packets from the PIC are parsed to overwrite local variables—synchronizing the UI with the authoritative hardware state (temperature and active relays)—while user input is interpreted as command requests and transmitted back to the master controller in 750ms intervals, effectively decoupling the HMI presentation layer from the physical control logic.



[Figure 5: Arduino HMI firmware flowchart]

HMI Interface



[Figure 6: Arduino HMI User Interface]

Model construction.

Introduction to Hardware Contributions

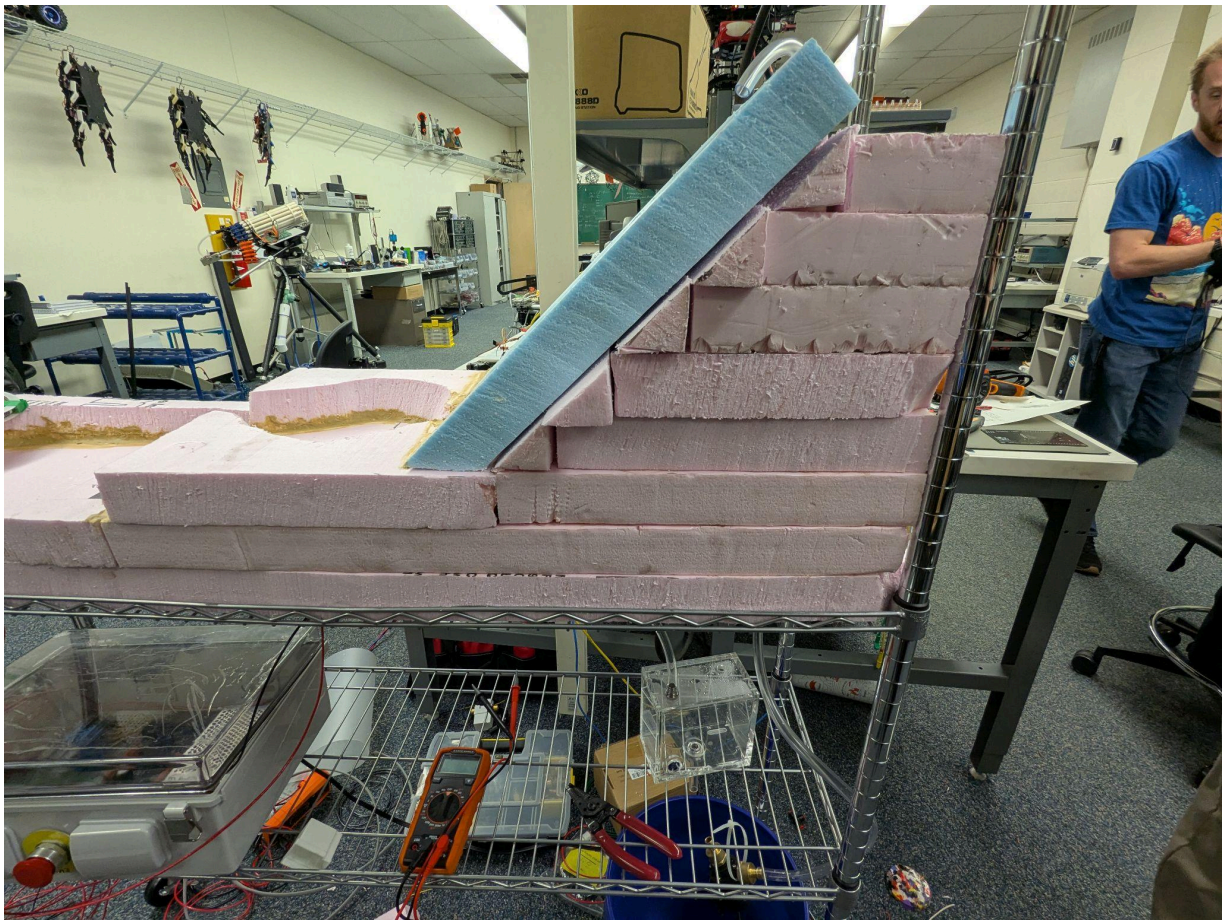
While the software and firmware governed logic, sequencing, and timing, the hardware served as the tangible interface between digital control and real-world physical behavior. Without a robust and well-considered hardware implementation, the project would not have been capable of reliably simulating watershed dynamics or supporting repeated demonstrations in an academic setting.

The watershed model itself is a scaled physical representation of a natural hydrological system. It incorporates three primary regions: an elevated mountainous area intended to represent snow accumulation and melt, a central reservoir that collects runoff, and a downstream town that experiences controlled flooding when reservoir spillway gates are opened. After each flood cycle, water is drained back into a recirculation tank beneath the model, allowing the system to reset and operate continuously without wasting water. This closed-loop design placed additional demands on the hardware, particularly with respect to water handling, electrical isolation, and durability.

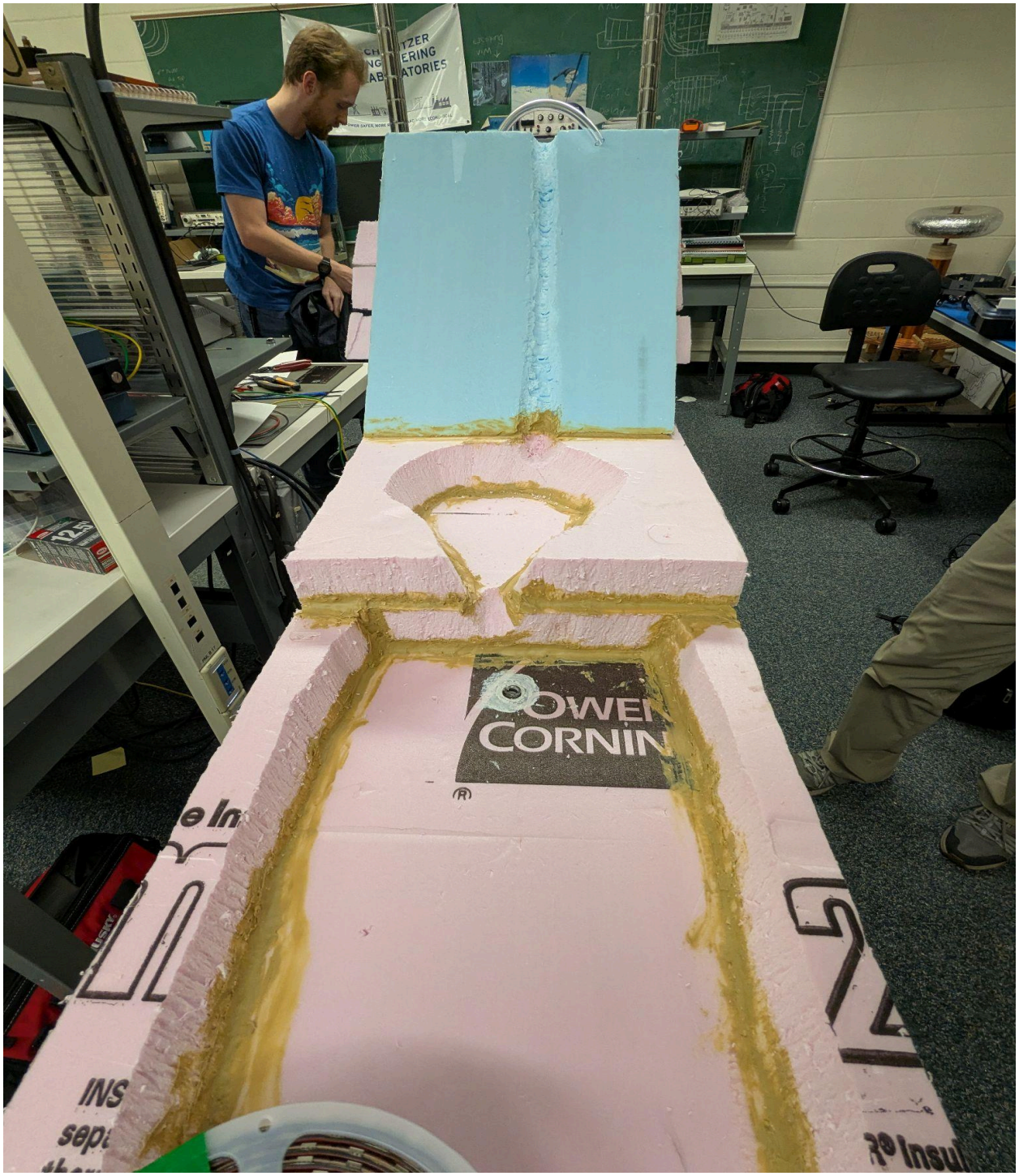
The hardware can be broadly categorized into three areas: physical model construction, electrical enclosure design and wiring, and integration of electromechanical components such as pumps, valves, servos, and lighting. Each of these areas required careful consideration of safety, reliability, and ease of maintenance. The following sections document the design rationale, construction methods, and engineering considerations that guided these decisions, with an emphasis on real-world constraints and lessons learned during implementation.

Materials Selection and Design Rationale

The physical watershed model was constructed using lightweight styrofoam materials selected for structural integrity, ease of use, cost efficiency, and compatibility with the water-based nature of the project. Expanded polystyrene foam was chosen as the primary construction material due to its low density, ease of shaping, and ability to support layered construction without excessive weight. These characteristics allowed the model to be large enough for visual clarity while remaining portable for transport and classroom demonstrations.



[Figure 7: Physical Model Construction, Side View]



[Figure 8: Physical Model Construction, Front View]

An important factor in material selection was the ability to modify and iterate on the design during construction. Foam materials can be cut, carved, and reshaped using common hand tools, making them well-suited for a project that requires iterative refinement of slopes, channels, and water paths. This flexibility proved critical as water flow behavior often differed from initial expectations and required adjustments after testing.

Waterproofing was achieved through the use of flexible sealants applied to seams, joints, and exposed surfaces. These sealants were selected for their ability to bond to foam, cure without becoming brittle, and tolerate repeated wet-dry cycles. Rigid coatings were avoided because they are more prone to cracking under minor deformation or temperature changes. Transparent plastic panels were considered to be incorporated selectively in areas such as the reservoir walls to allow observers to visually monitor water levels during operation, enhancing the educational value of the model. Due to time constraints, however, this was excluded from the prototype model.

Environmental and ethical considerations were also taken into account. While foam materials are not biodegradable, their use minimizes the need for heavier or more resource-intensive alternatives. Construction waste was reduced through careful planning and reuse of offcuts for smaller features such as terrain details and town structures. Additionally, sealants and paints were chosen to be non-toxic once cured, ensuring the recirculating water posed no hazard during demonstrations.

Construction Process

The construction of the physical model followed a staged process that allowed individual subsystems to be tested before full integration. Initial planning involved defining the overall

footprint of the model and allocating space for the mountain, reservoir, town, and drainage system. These regions were arranged to emphasize gravity-driven water flow, reinforcing the educational objective of demonstrating how elevation influences watershed behavior.

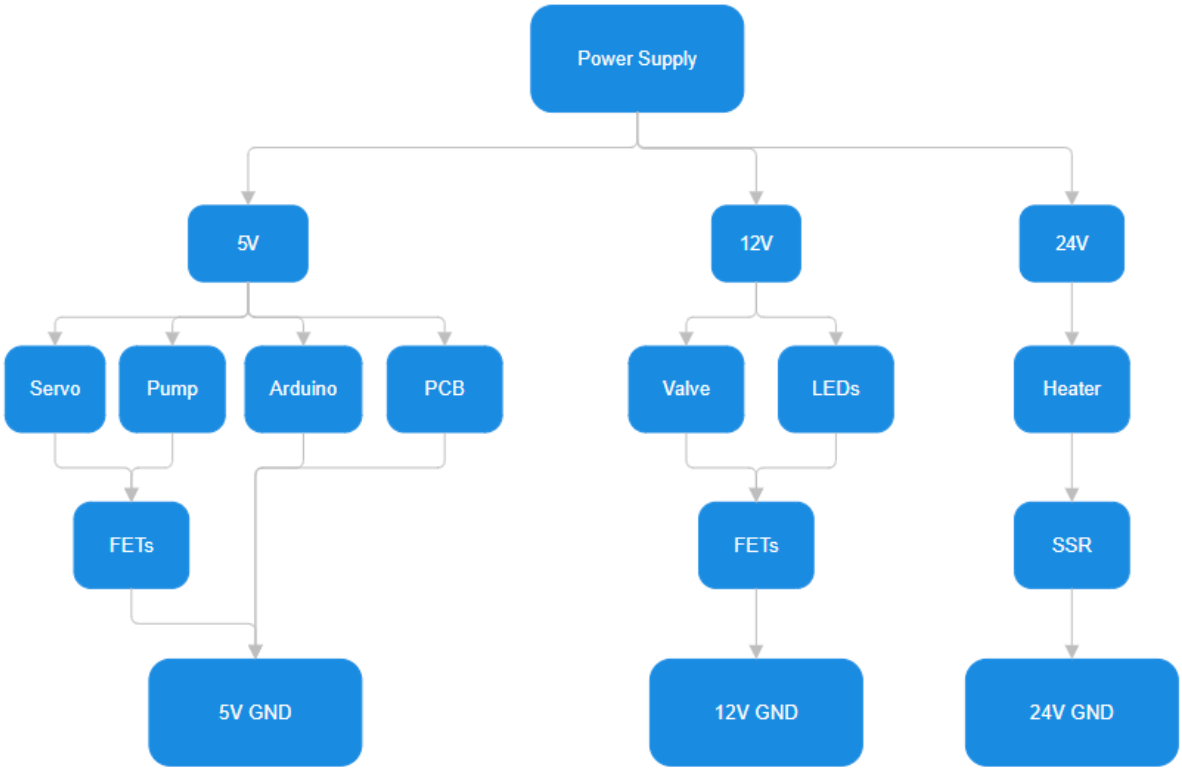
Base layers were assembled first to establish a stable foundation. Foam sheets were bonded together to form a rigid platform. Once the base was complete, additional layers were stacked and carved to form the mountain region. Slopes were shaped gradually to improve visual realism and to promote smooth water flow toward the reservoir.

Channels were carved into the mountain surface to guide simulated snowmelt toward the reservoir. These channels were refined through repeated testing, as small changes in geometry had noticeable effects on flow direction and pooling. After shaping was complete, sealant was applied in multiple passes, with curing time between applications to ensure full adhesion and coverage. Leak testing was performed incrementally, allowing problem areas to be identified and corrected before additional components were added.

The reservoir basin was formed by removing material from the foam base. Particular attention was paid to the interface between the reservoir and the spillway gate mechanism, as this area experiences both mechanical motion and water pressure. The downstream town area was constructed last. The idea was to use smaller foam structures or Legos to represent buildings and infrastructure. Drainage paths were integrated beneath this area to return water to the water recirculation tank after each demonstration cycle.

This iterative approach to construction significantly reduced the risk of major redesign late in the project. By validating each section individually, issues could be addressed while access was still available and before additional complexity was introduced.

Electrical System Design and Architecture



[Figure 9: Hardware Wiring Block Diagram]

This diagram serves as a conceptual representation of power flow from the central power supply to individual subsystems, as well as how control elements and grounding are organized.

At the top of the hierarchy is the centralized power supply, which serves as the sole source of electrical energy for the system. From this supply, power is distributed into three distinct voltage rails: 5 V, 12 V, and 24 V. Each rail is dedicated to components with similar electrical requirements, allowing for clear separation of low voltage logic, moderate power

peripherals, and higher power actuators. This separation reduces electrical noise, simplifies troubleshooting, and improves overall system safety.

5 V Rail: Logic and Low Power Control Subsystems

The 5 V rail supplies power to the core control electronics of the system, including the microcontroller platform and associated printed circuit board. These components are responsible for executing the simulation logic, generating control signals, and coordinating the timing of system events. By isolating logic circuitry on its own voltage rail, the system minimizes the risk of voltage instability caused by higher current loads.

In addition to control electronics, the 5 V rail also supports devices that require precise, low-power operation, such as the servo motor used for spillway gate actuation. Although servos are mechanical devices, their control electronics operate at logic-level voltages and benefit from being powered separately from higher current systems. Field-effect transistors (FETs) are shown downstream of the 5 V rail to indicate their role as electronic switches, allowing low-power control signals to manage connected devices without directly handling load current.

All components powered by the 5 V rail return to a common 5 V ground reference. Maintaining a dedicated ground path for this rail helps preserve signal integrity and reduces susceptibility to electrical noise introduced by inductive loads elsewhere in the system.

12 V Rail: Intermediate Power and Visual Outputs

The 12 V rail is used to supply components that require more power than logic devices but do not necessitate the higher voltage of the pump or heating elements. In this system, the 12 V rail supports the solenoid valve and the LED lighting used to simulate activity within the town portion of the model.

The solenoid valve represents an inductive load that draws transient current during actuation. Routing this device through controlled switching elements, such as FETs, allows the microcontroller to command valve operation without being exposed to electrical transients. Similarly, LED lighting loads are grouped on this rail to provide consistent brightness and simplify power management.

As with the 5 V rail, all 12 V powered components share a dedicated ground reference. This intentional grounding structure helps prevent interference between logic-level signals and higher-current switching events, particularly during rapid actuation of the valve.

24 V Rail: High-Power Actuation and Isolation

The 24 V rail is reserved for the highest power components in the system, the heater element used to enhance the realism of the snowmelt simulation. Higher voltage operation allows required power levels to be delivered with lower current, reducing conductor losses and improving overall efficiency.

Switching of 24 V loads is handled through a solid-state relay (SSR), which provides electrical isolation between the control electronics and the high-power device.

[Figure 10: Solid State Relay (SSR)]

This isolation improves safety and protects sensitive control circuitry from voltage spikes or faults originating on the high-power side of the system.

A dedicated 24 V ground reference is maintained for this rail, further reinforcing electrical separation and ensuring predictable behavior under load.

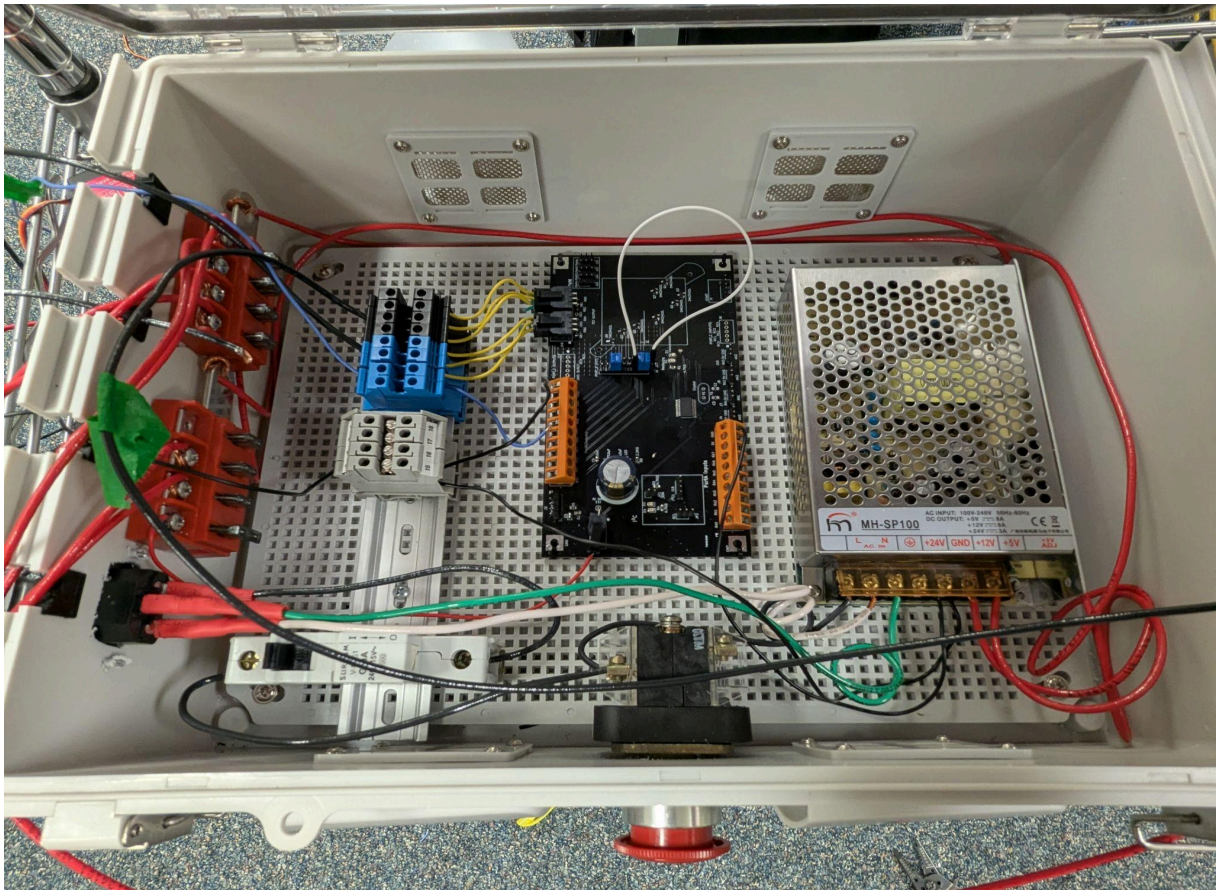
One of the most important aspects highlighted by the block diagram is the intentional separation of grounding paths by voltage rail. While all grounds ultimately reference a common

system ground, organizing return paths by voltage level reduces ground bounce and minimizes coupling between unrelated subsystems.

This approach is particularly important in systems that combine digital logic, inductive actuators, and power electronics.

Overall Electrical Architecture

The electrical system was designed around a centralized enclosure that houses all major power distribution and control components. This enclosure serves as the interface between external power, internal control logic, and the physical actuators distributed throughout the model. Centralizing these components improved safety, simplified troubleshooting, and allowed the system to be powered down or isolated quickly when necessary.



[Figure 11: Electrical System Box, Internal Electrical Wiring]

A layered architecture was adopted to separate high voltage input, low voltage power distribution, and signal level control. Incoming power is conditioned and protected before being converted into multiple regulated voltage levels. These voltage rails are then distributed to individual subsystems through protective devices, ensuring that a fault in one area does not propagate throughout the entire system.

This architecture reflects standard practices used in industrial control panels and laboratory equipment. Although the project itself is educational in nature, applying these

conventions helped ensure safe operation and provided valuable experience in professional hardware design methodologies.

Power Distribution and Protection

Electrical safety was a central consideration throughout the design process, particularly due to the coexistence of water and electrical components. The power input path was designed to include multiple layers of protection, beginning with a primary disconnect mechanism that allows all power to be removed from the system quickly. This feature is especially important during live demonstrations, where unexpected behavior must be addressed immediately.

Downstream of the primary disconnect, the power supply converts incoming AC power into regulated low-voltage outputs suitable for the various components in the system. Each output rail is protected individually using overcurrent protection devices. This approach localizes faults, making it easier to identify and resolve issues without disabling the entire system.

Fuse protection was implemented to safeguard wiring and components against excessive current. Rather than relying solely on nominal component ratings, protection devices were selected based on observed operating behavior and realistic load conditions.

Component	Typical Current	Maximum Current	Notes
Solenoid Valve	350mA	500mA	Inductive load; higher current occurs during

			activation
Servo Motor	250mA	700mA (stall current)	Stall current represents worst case mechanical loading
Water Pump	60mA	100mA	Continuous operation under low mechanical resistance
LED roll	300mA	300mA	Approximated for ¼ of the LED roll
Heating Element	—	15A	High power resistive load, controlled via SSR

[Table X: Measured Component Current Values]

The current requirements of each major electrical load were identified to inform fuse selection, conductor sizing, and switching device selection.

This conservative approach reduced the likelihood of thermal stress, insulation damage, or component failure during extended operation.

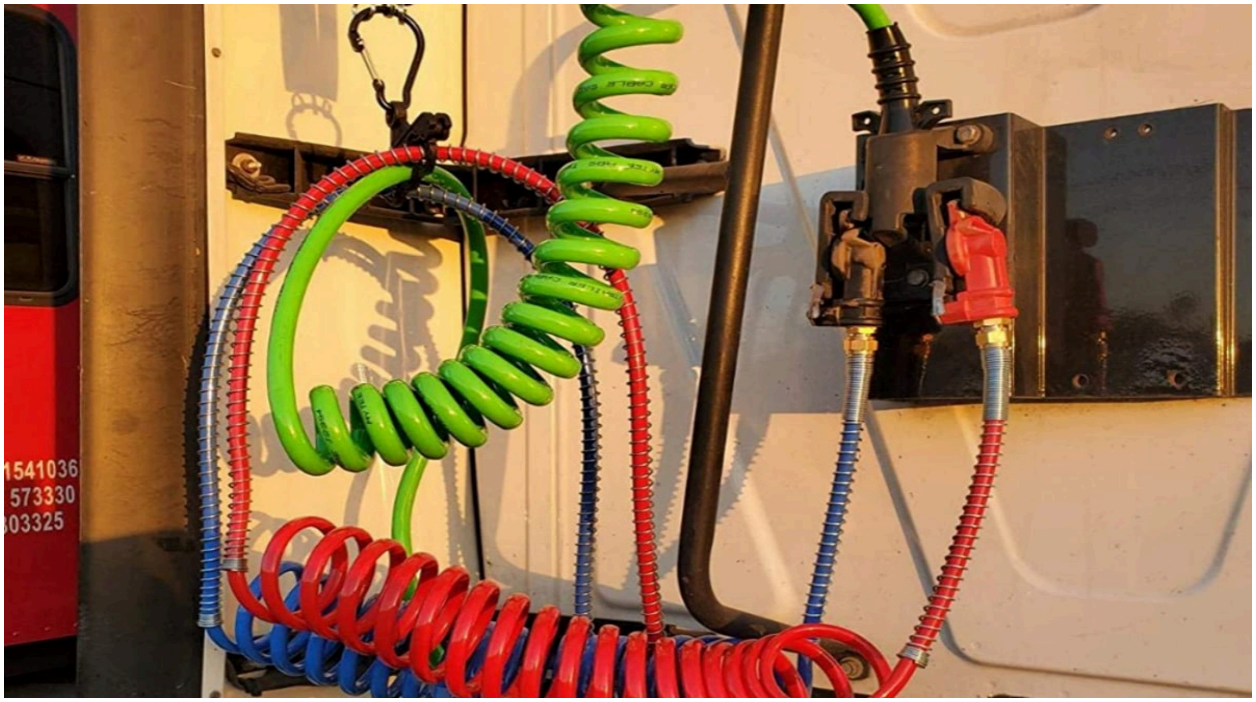
Wire Selection and Routing Considerations

Wire selection played a critical role in ensuring both electrical safety and long-term reliability. Conductors were chosen based on the expected current levels and environmental conditions within the enclosure. Higher current paths were routed using heavier gauge wire to minimize voltage drop and heat generation, while lower current signal and control lines used smaller conductors to improve flexibility and organization.

Color coding was used consistently to distinguish between different voltage levels, ground connections, and control signals. This practice significantly improved clarity during installation and troubleshooting, reducing the risk of wiring errors. Conductors were routed along defined paths and secured to prevent movement, chafing, or accidental disconnection.

Termination methods were selected to ensure secure and repeatable connections. Stranded wires were properly prepared for terminal blocks, and mechanical strain relief was incorporated where wires entered or exited the enclosure. One method for mechanical strain relief, and the method implemented here was spiral strain relief, commonly found in:

- Pneumatic hoses
- Electrical cables
- Hydraulic lines
- Semi truck glad-hand air hoses.



[Figure 12: Spiral Strain Relief Hoses]

[Figure 13: Spiral Strain Relief Wires]

This refers specifically to a helical or coiled geometry used to absorb movement, tension, and vibration. These details, while often overlooked, contributed substantially to the overall robustness of the system.

Current Considerations and Real World Validation

An important lesson learned during this project was the necessity of validating electrical assumptions through direct measurement and observation. Initial testing revealed that some components exhibited higher current demands during startup or transient conditions than during steady state operation. These behaviors can place additional stress on wiring and protection devices if not accounted for.

To address this, current behavior was observed under realistic operating conditions, and the system was designed to tolerate short-duration high currents without compromising safety. This validation process informed decisions related to protective device selection and reinforced the importance of empirical testing in hardware design.

By accounting for real-world behavior rather than idealized specifications, the hardware system achieved a higher level of reliability and resilience. This approach also provided valuable insight into the practical challenges engineers face when bridging the gap between theoretical design and physical implementation.

Integration of Electromechanical Components

Pumps, Valves, and Actuators

The dynamic behavior of the watershed model relies on several electromechanical components that convert electrical signals into physical motion. A pump is used to move water

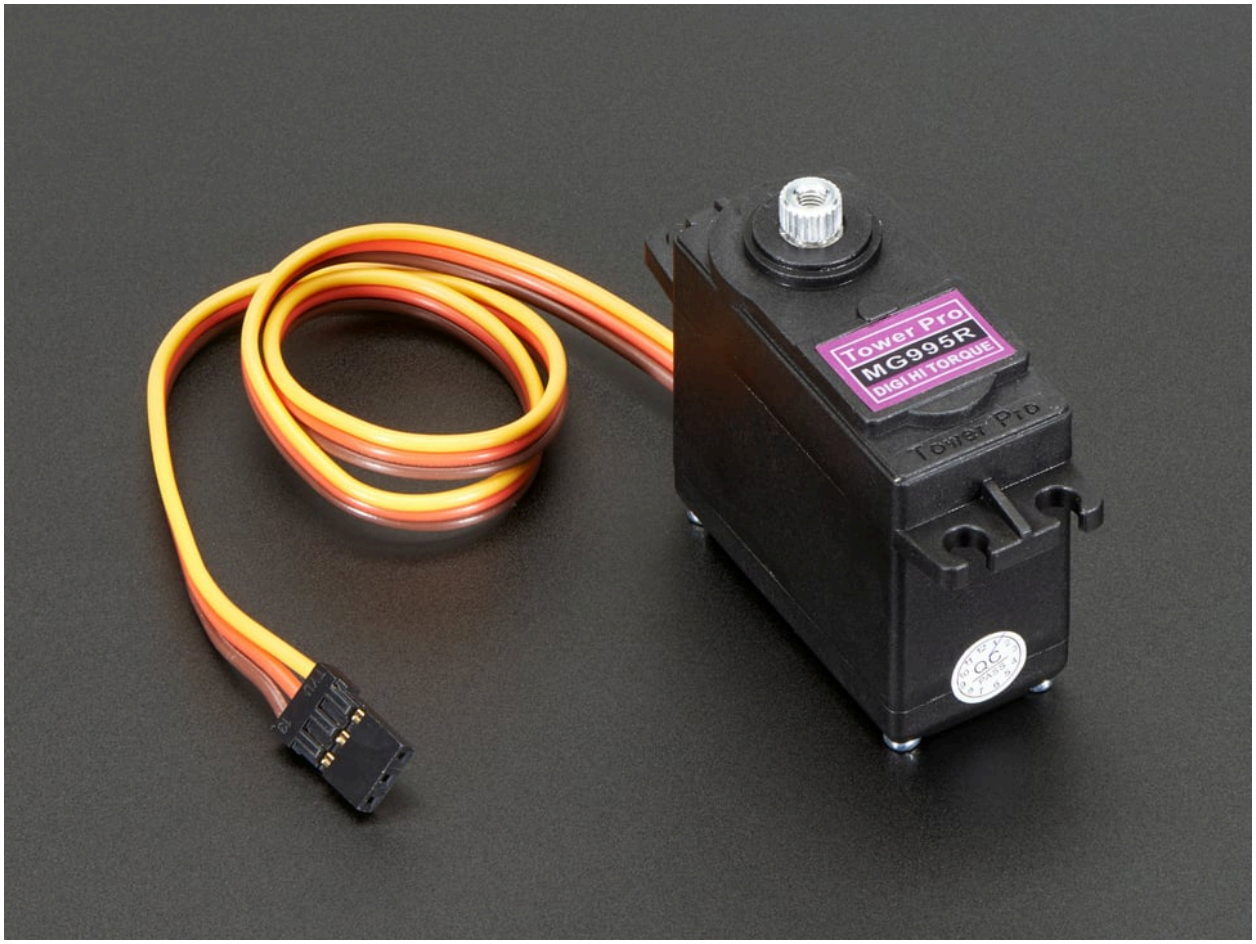
from the recirculation tank to the mountain region, simulating snowmelt runoff. The pump's operation is controlled electronically, allowing its activation to be synchronized precisely with the simulation cycle.

[Figure 14: Water Pump]

Flow control within the system is achieved using a valve that regulates when and how water is released or redirected. This component plays a key role in ensuring repeatable behavior between demonstrations.

[Figure 15: Relays]

A servo motor is used to actuate the spillway gates of the reservoir, providing controlled and visually clear flooding of the downstream town.



[Figure 16: Servo]

Each of these components introduces unique electrical and mechanical considerations, including inductive effects, mechanical wear, and sensitivity to voltage fluctuations. Protective measures such as isolation and appropriate control strategies were incorporated to mitigate these risks.

Lighting and Visual Indicators

Lighting elements were integrated into the town area to enhance the visual impact of the simulation and provide intuitive feedback on system state. These lights serve both an aesthetic

and functional purpose, helping observers understand when specific phases of the simulation are active.

[Figure 17: LED Roll]

From a hardware perspective, these components required careful consideration of power distribution and control to ensure consistent brightness and avoid interference with other

subsystems. Integrating visual indicators into the hardware design reinforced the importance of human-centered considerations in engineering projects.

Calibration and Verification of electrical

Once construction and wiring were complete, a systematic calibration process was undertaken to verify the correct operation of each subsystem. This process began with basic power verification, ensuring that all voltage rails were present and stable under load. Individual components were then tested independently to confirm correct behavior before full system integration.

Mechanical components such as the spillway gate were adjusted to ensure proper alignment and range of motion. Flow behavior was observed and refined through repeated test cycles, allowing adjustments to be made to timing and actuation as needed. This step by step approach minimized the risk of compounding errors and made it easier to isolate issues during troubleshooting.

Troubleshooting Strategy and Maintenance Considerations

A structured troubleshooting approach was developed to address potential faults quickly and safely. By isolating subsystems and verifying power, control signals, and mechanical motion independently, issues could be identified with minimal disruption to the rest of the system.

Maintenance considerations were also incorporated into the design. Components were arranged to remain accessible, and wiring was labeled clearly to facilitate future modifications or

repairs. These design choices are particularly important in educational projects, where systems may be reused or adapted for future classes.

Lessons Learned and Engineering Reflection

This project highlighted the importance of disciplined hardware design practices, particularly when working with systems that combine electrical power, mechanical motion, and water. Seemingly minor decisions related to wire routing, component placement, or protection strategies had a significant impact on system reliability and safety.

Perhaps the most valuable lesson was the necessity of iterative testing and validation. Assumptions made early in the design process were refined through observation and measurement, leading to a more robust final implementation. This experience reinforced the understanding that effective hardware engineering is as much about managing uncertainty and risk as it is about technical knowledge.

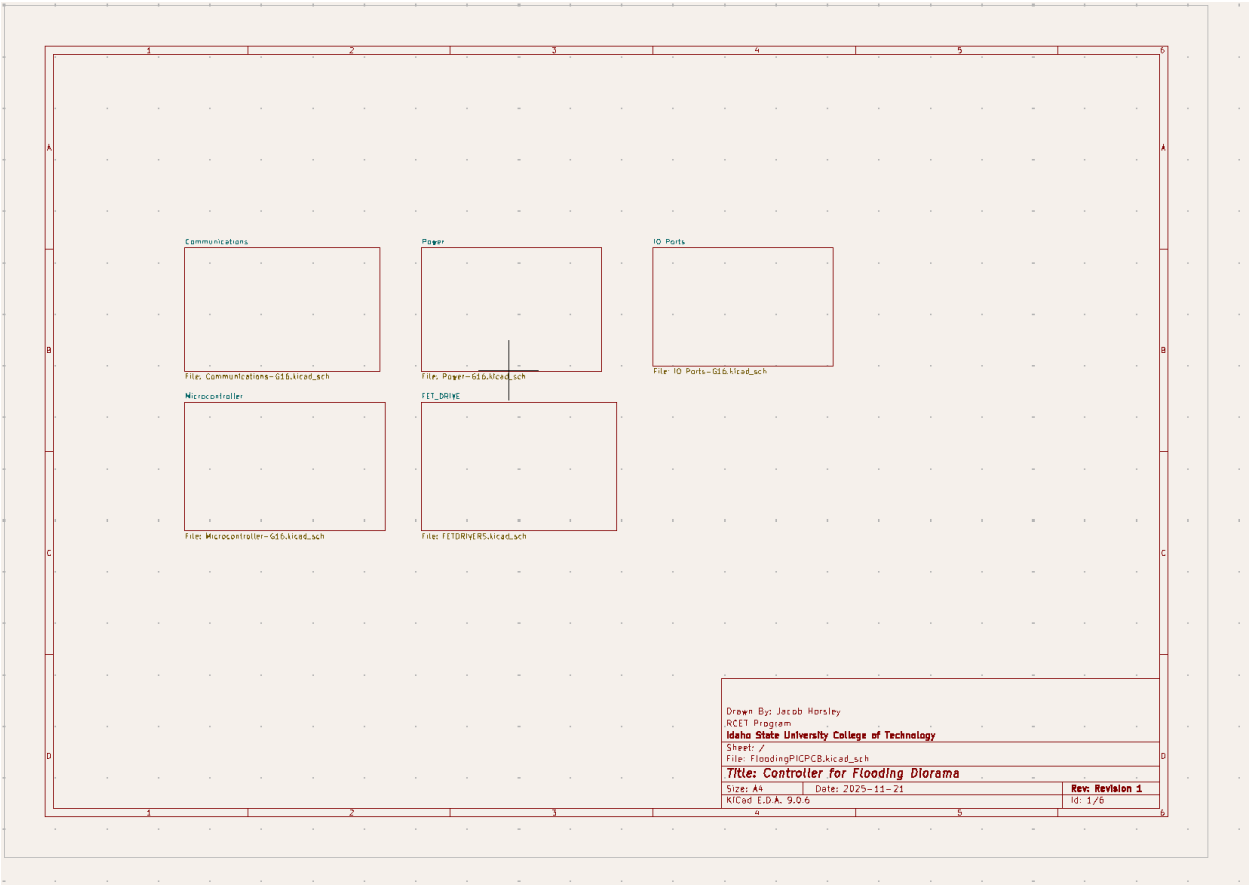
Conclusion and Future Recommendations

The hardware implementation of the Farm Bureau Watershed project successfully supported the goals of the overall system, enabling a repeatable, safe, and visually engaging demonstration of watershed dynamics. The combination of thoughtful material selection, careful electrical design, and iterative testing resulted in a system that performs reliably under demonstration conditions.

Future improvements could include additional sensing capabilities to provide real-time feedback on water levels or flow rates, further enhancing the educational value of the model.

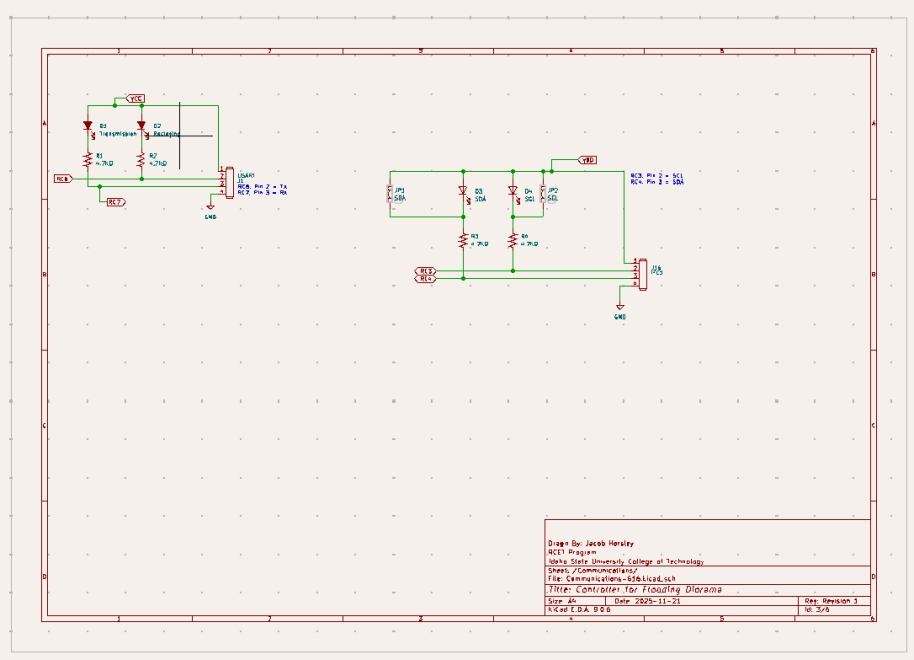
Improvements to enclosure sealing and connector modularity could also increase durability and ease of maintenance. These enhancements would build upon the strong hardware foundation established during this project, and support continued use and expansion in future iterations.

Electronic Design



[Figure 18: Block Diagram Kicad]

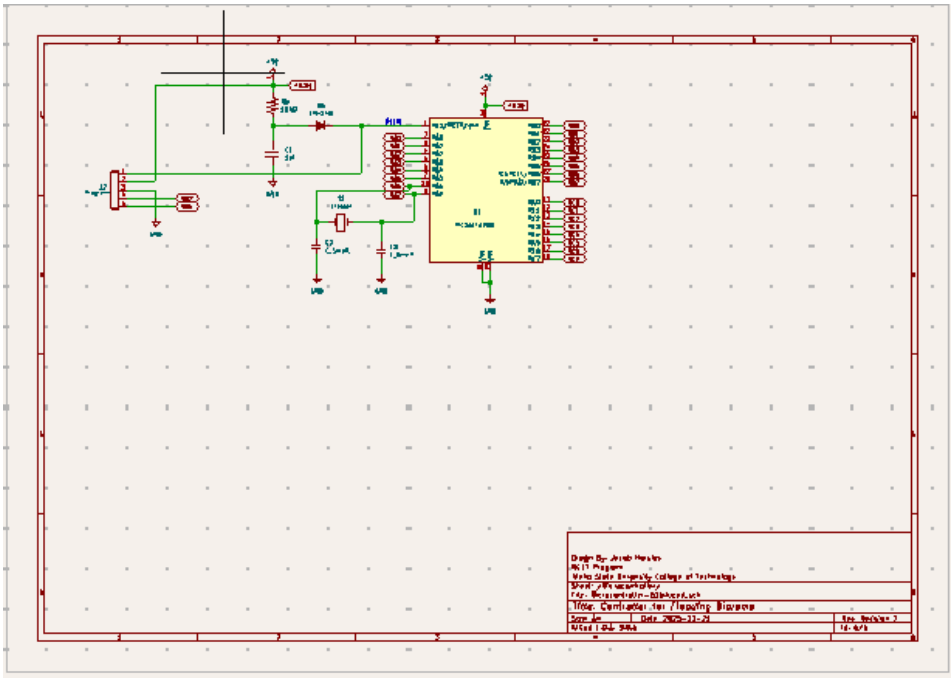
This block diagram covers the entirety of the Flooding watershed PIC schematic. It does not demonstrate how the final hardware outputs are connected. This is due to the urgency of ordering the PCB board and project planning for the hardware still occurring while development of the schematic was taking place. This block diagram can be compared to the final hardware diagram for a more detailed look at how the final outputs turned out.



[Figure 19: Communications schematic]

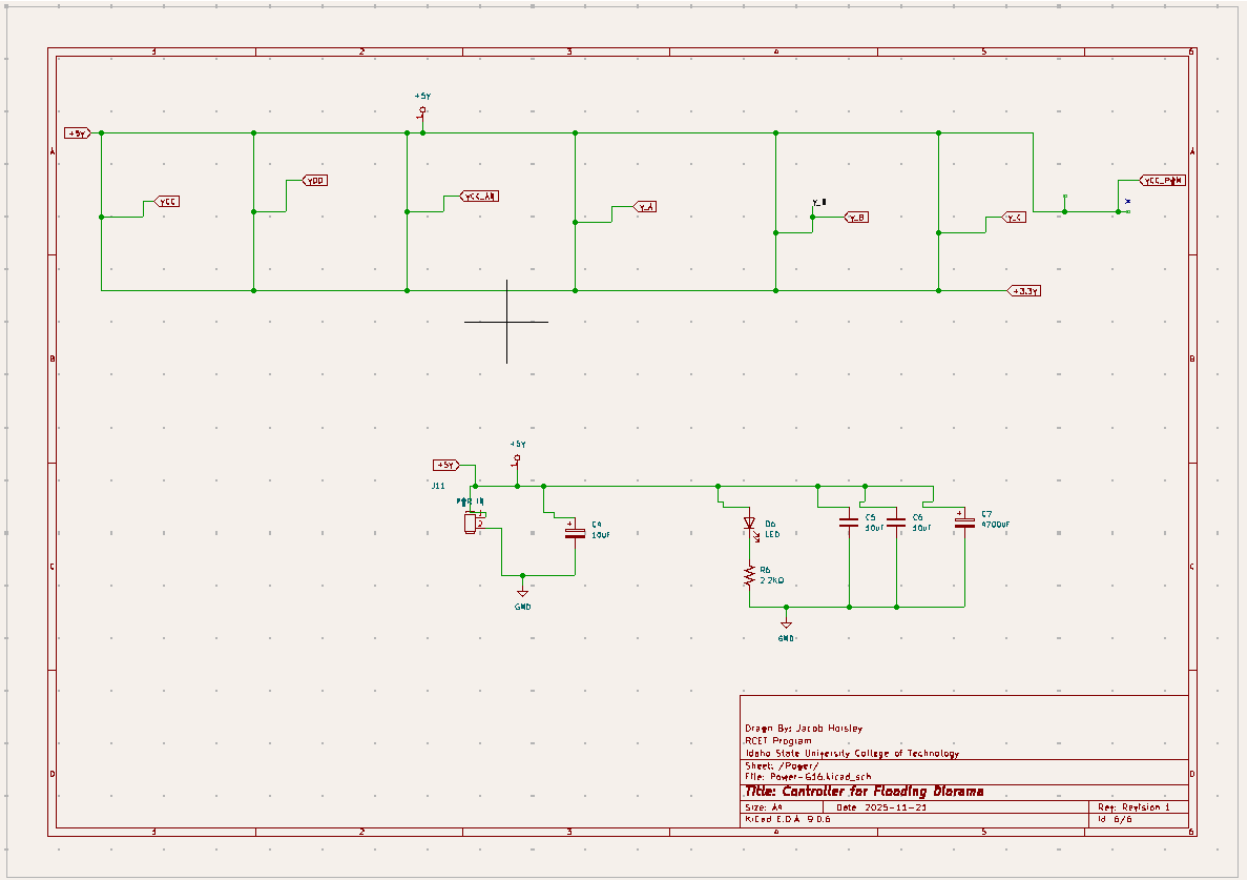
This schematic shows the output headers for the serial communication and the I2C communications. For this schematic, we will only utilize serial communication. For this semester, UART will constantly send its acquired data for simplified transmission to any device. No required code needs to be sent to initialize the communication through UART. The data is

sent. For anyone who decides to continue this project, this simplifies the process of creating external hardware that can just be built to read what data is being sent.



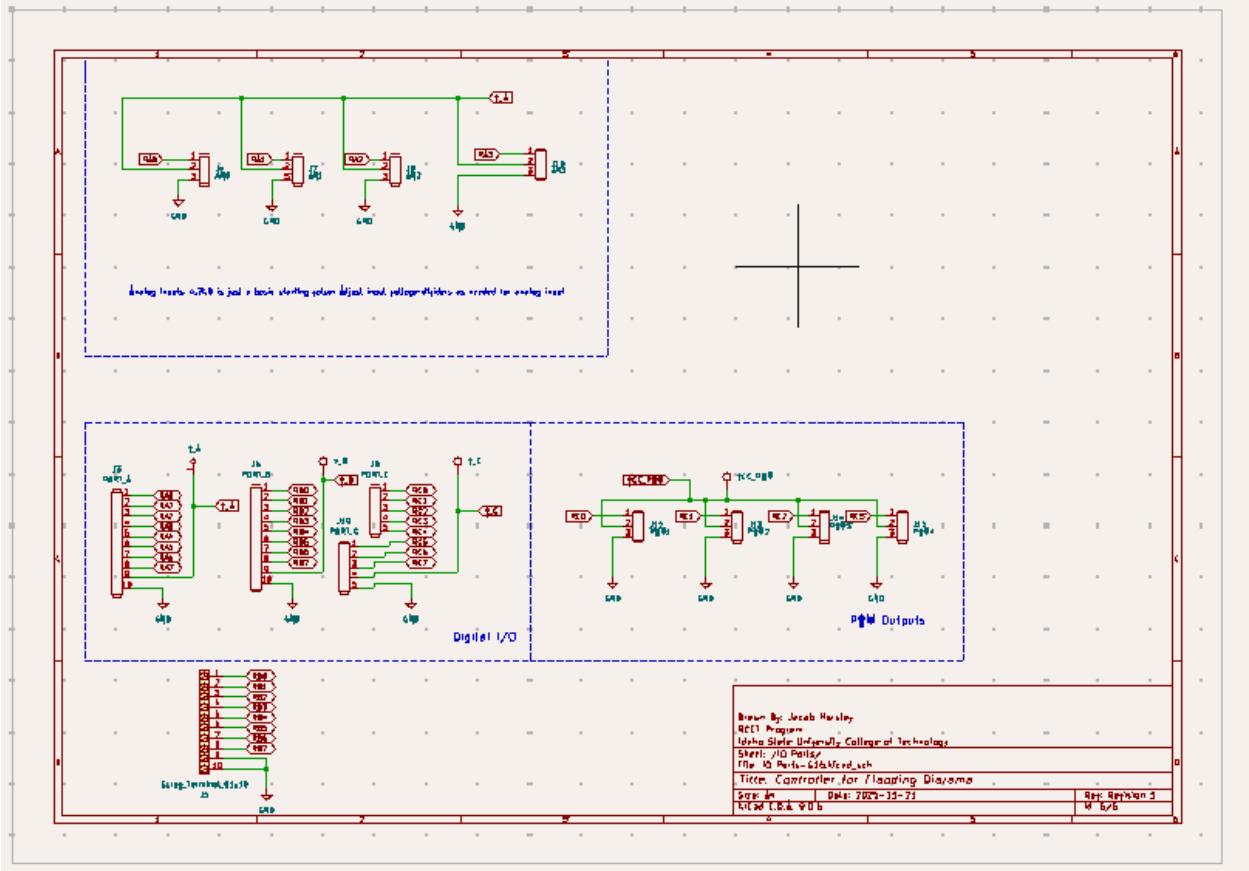
[Figure 20: Microcontroller Schematic]

The above microcontroller schematic shows the basic configuration for the PIC16F1788. This basic controller just shows the basic setup for the programming pins primarily. Nearly all the Port pins go directly to the IO port page of the schematic. Only the TX and RX pins are being used for UART currently. Any future modifications should take into account I2C. The implementation can be carried out if the SCL and SDA pins are left open.



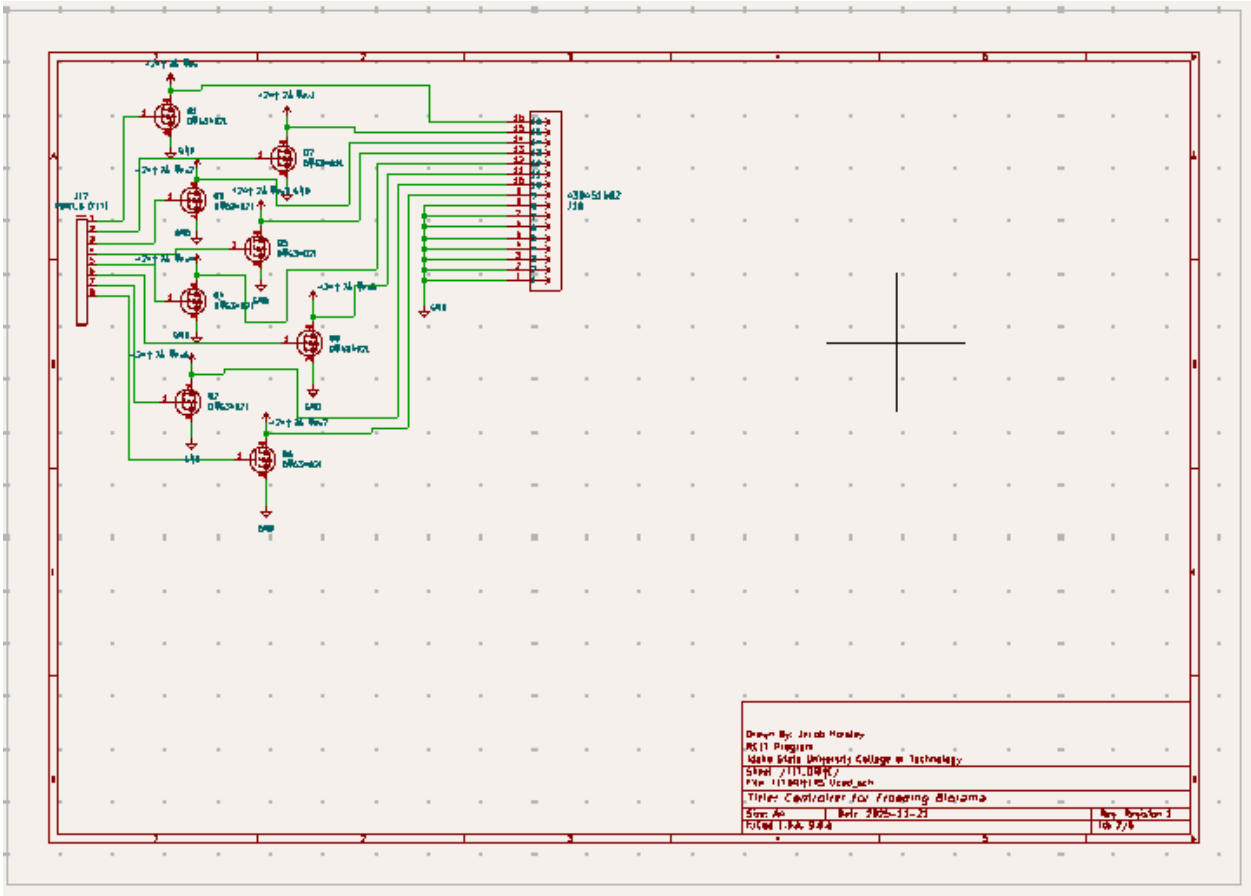
[Figure 21: Power schematic]

This power schematic shows how the PIC16F1788 is connected to power. It has an optional Crystal, but we'll be running the PLL at 16 MHz. That is the internal oscillator. This shows all the inputs and outputs. Referring to the PCB Layout in Figure 19 shows the proper inputs and outputs of all the ports. For the tentative plan, PortA includes all the inputs. PortB is primarily in charge of the Outputs. Serial communications and any other Communications will take place on PortC.



[Figure 22: IO Ports]

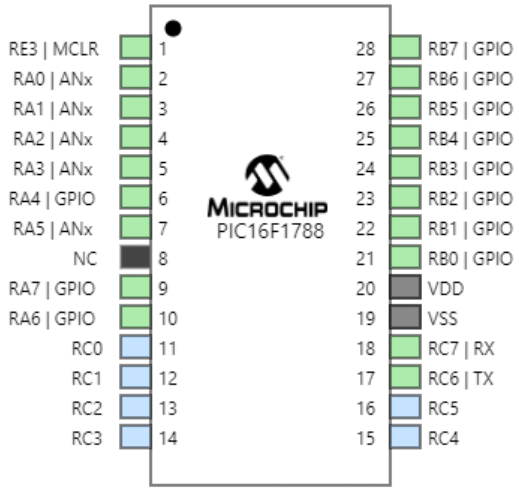
This part of the schematic was borrowed from the PIC development board in the KIcad class. It was prioritized for the PIC16LF1788, which is a low-power version of what is being used right now. This section was intended to drop the voltages down to CMOS levels for the low-power FET. Essentially, this board shorts those options to drop the voltage down. A large filtering cap is present to reduce any rippling or inconsistencies that may be present from the power supply.



[Figure 23: FET Drivers]

These FET drivers connect to a MOLEX connector that drives devices of varying DC inputs. The configurations allow for flexible configuration of devices. The N-channel MOSFET that was available for our project was the DMG3402, which comes in an SOT23L package. This MOSFET is rated for 30V and 4A. The datasheet from Diodes Incorporated,Document number: DS36077 Revision 5-2, was used as a reference. These diodes well exceed any of our hardware requirements. As the valves, pump, and servo don't take more than a measured 500 mA.

Note: the heating element drew more current than was anticipated and will have to be coupled through an SSR.



[Figure 24: Finalized pinout for the PIC16F1788]

Legend

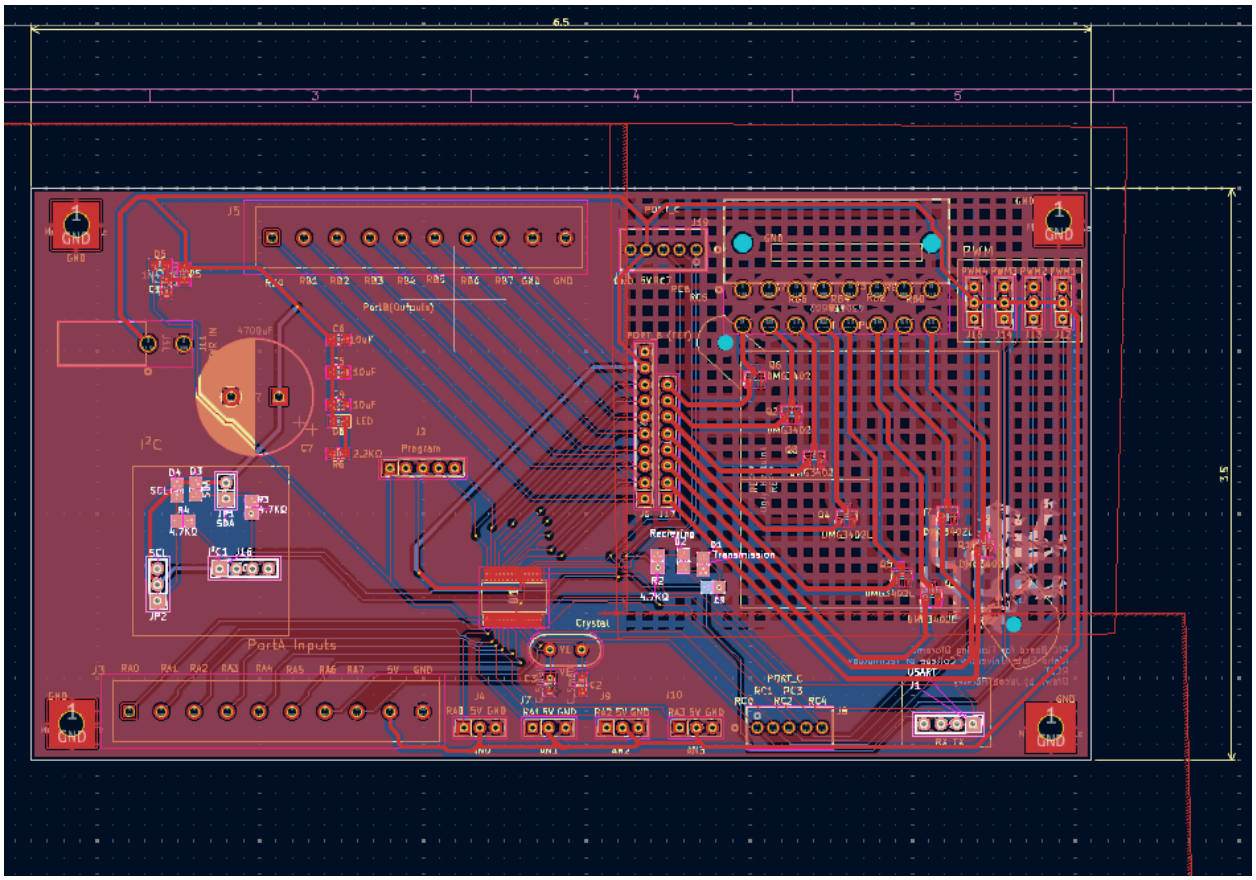
RA0: Tank sensor RA1: Reserve Sensor RA2: Town water sensor

RA4: SCRAM

RA3: Heatsink temp sensor RA5: FET Temp sensor RA6:Pause/Go

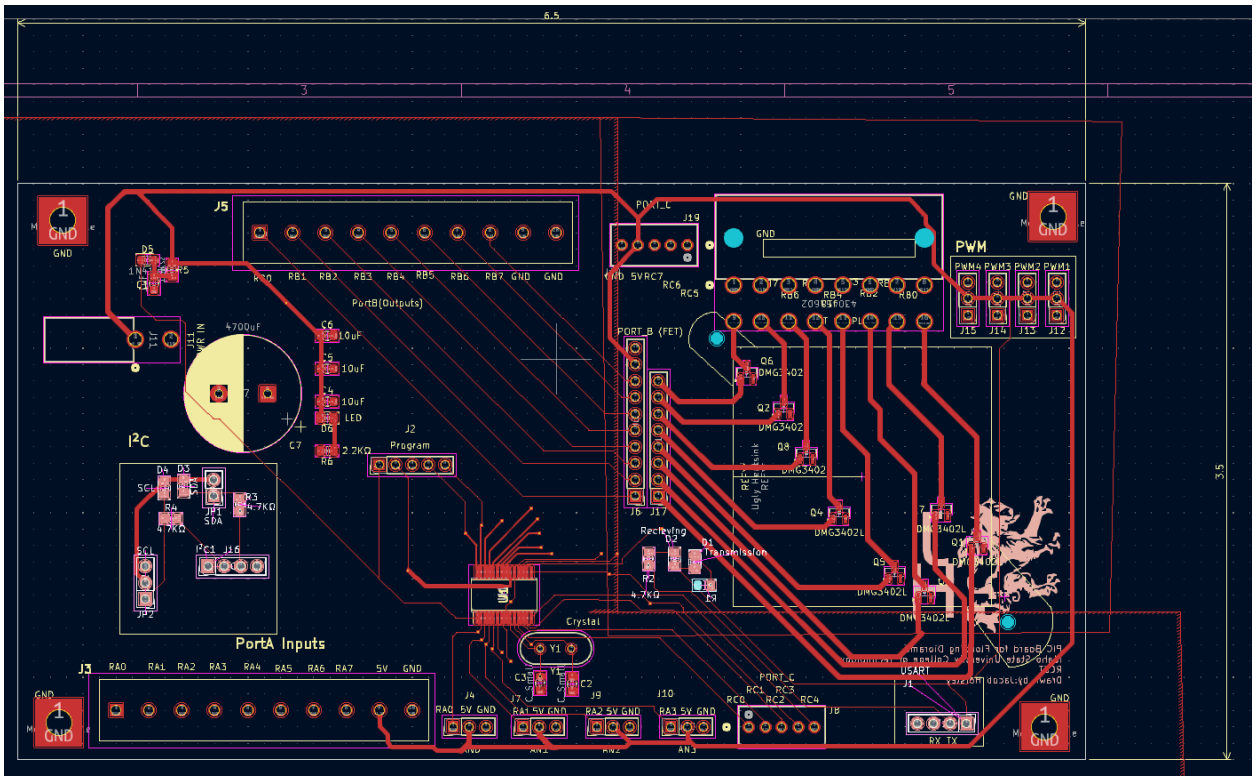
RA7: Return from SCRAM

Component and PCB Layout



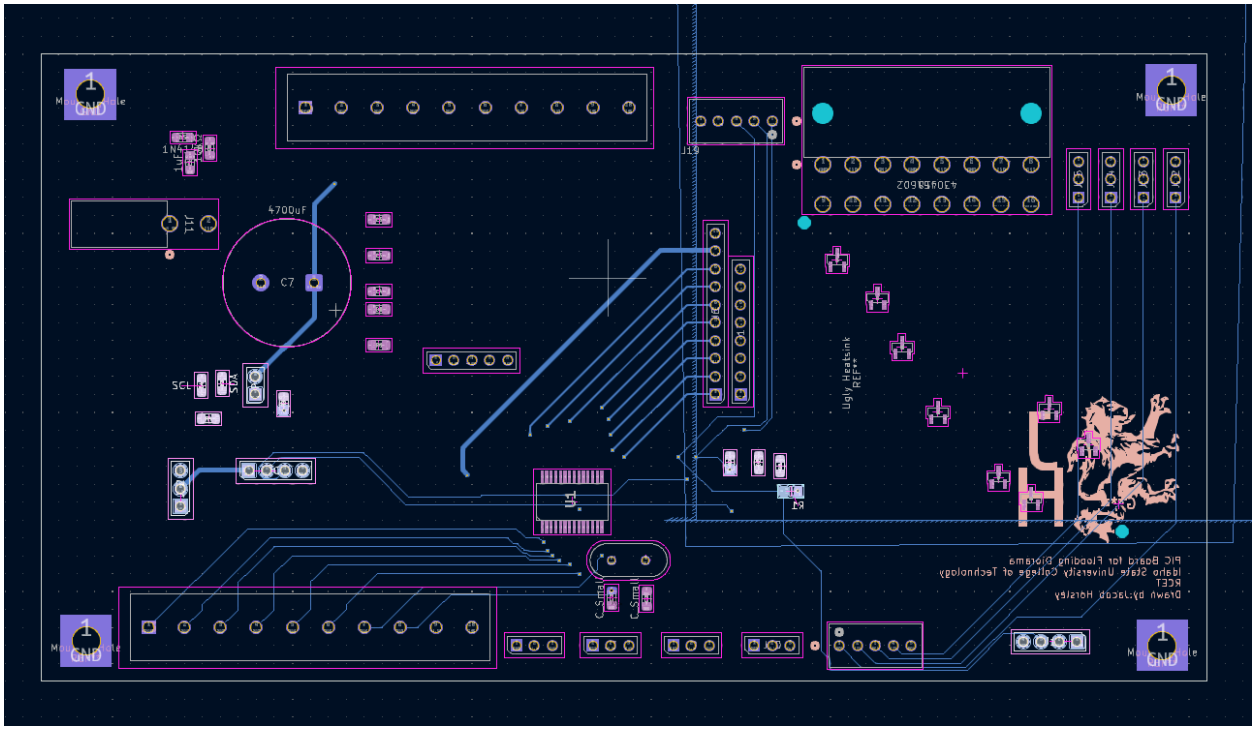
[Figure 25: Full PCB Layout]

This is the finalized PCB design with copper flooded to the ground plane on both sides of the board. The image shows everything, but it may be difficult to interpret. Refer to the following figures.



[Figure 26: Front Copper and SilkScreen only]

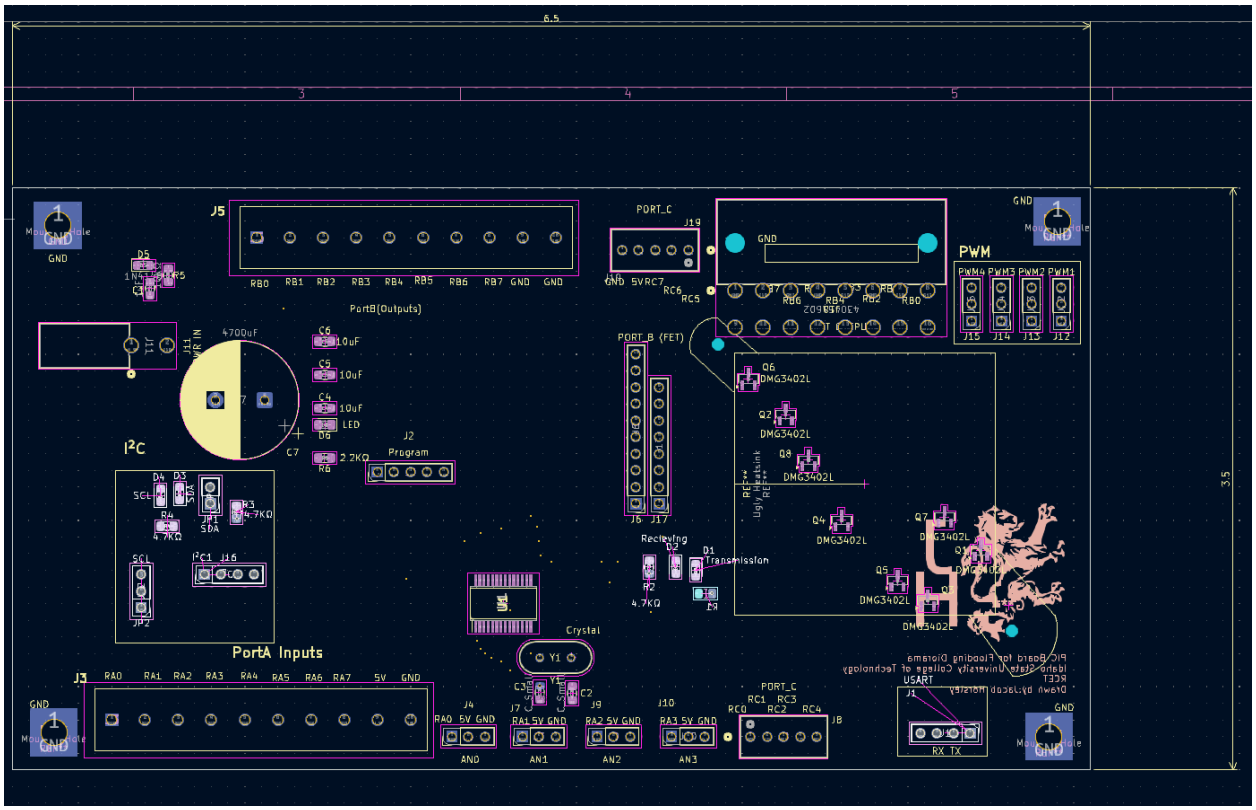
Figure 23, with only the top silkscreen and copper, shows clearly how the components are configured on the PCB. During initial planning, the board was designed for outputs to come from the top of the board and inputs to be connected from the bottom of the board. Realistically this board could be minimized by having ports situated on each side of the board, as there is currently an abundance of empty space in the left, right, and middle sections of the board.



[Figure 27: Back Copper and Silkscreen]

This back copper shows some of the traces that are on the back of the board. Primarily, only connections that require vias are located in this section of the board.

One recommendation is to eliminate vias in further revisions in the case that interference becomes present on the transmission lines that utilize I2C or EUART.



[Figure 28: PCB Component Layout]

The layout of this board was set up in a way to accommodate the wires that would be coming into the board from the bottom and the top. Further revisions should reposition the I2C pins that are present under the filter cap on the left side of the board. Currently,these headers are troublesome to access. The Heatsink for the MOSFETs should be moved. It is apparent now that the MOLEX connector is obstructing the heatsink on the right side of the board.

Calibration Procedure

This device, when it comes to calibration on the code side,e doesn't require much. The device should be inspected and revised to follow the case structure outlined in the future manual. If hardware is being triggered sporadically and unexpectedly, the first thing to check should be the connections. If connections are loose or not connected, these should be attached tightly with the proper tool. If all connections are fine, voltages should be checked next. Ensure power is being supplied as expected to all components. Fuses have been built into the system, and if any of the fuses have blown,visually inspect the cables and all pieces of hardware before installing a new fuse. Fuses must match the rating specified on the box or in the datasheet. If all those values are good, check the PCB with a DMM and ensure all the power is being supplied. If power is supplied, it could be a viable option to start and restart the microcontroller.

When problems persist with the microcontroller, it will then be a viable option to reprogram the PIC with a PICKIT. Code for this hardware can be found on the GitHub linked to this project. The GitHub can be accessed through this URL: https://github.com/Christopher-isu/Farm-Bureau-Project

Calibration with an external device can be carried out. Refer to the coding section in this document to send the appropriate commands for the UART to change the desired values. This is only temporary, though,h and once the power is shut off, the changes are lost. The delays would need to be set upon startup.

Software Description

The software mainly used for this code is C code that is specialized for coding the hardware for the PIC16F1788. Some Assembly is used to ensure compatibility with the hardware. An optional piece of hardware that can send serial commands is possible. The software has the option to interface with any serial device.

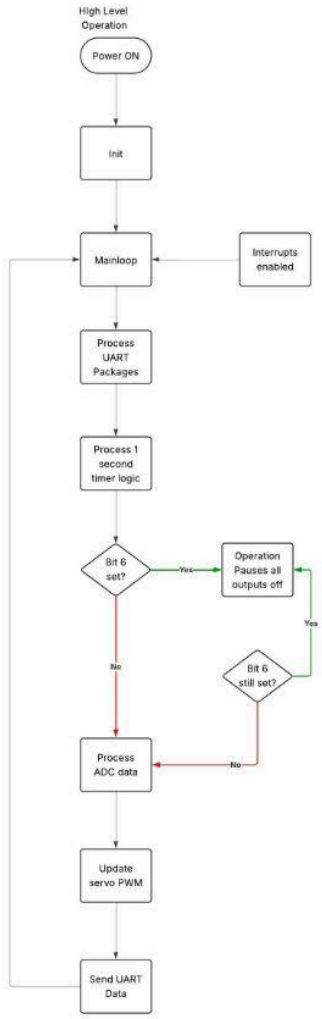
Final code GitHub repository:

https://github.com/Christopher-isu/Farm-Bureau-Project/tree/main/Code

Development code to see the history:

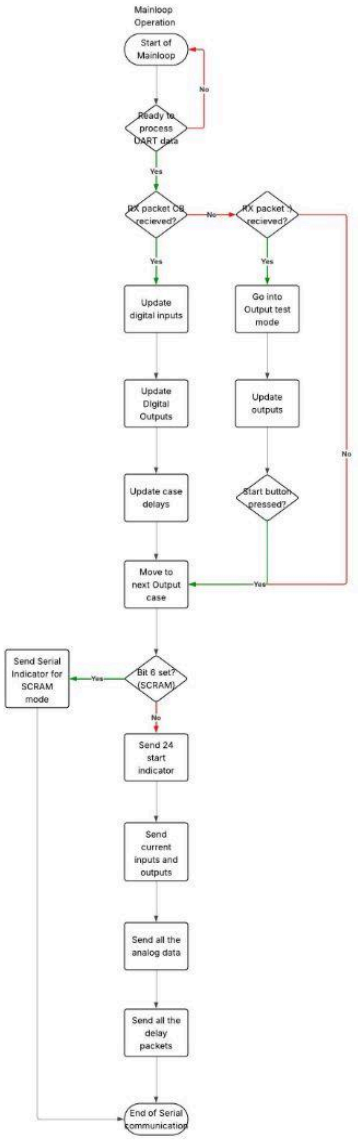
https://github.com/horsjaco117/C-Code-for-Final-watershed-project

Figures of the flowcharts will be included below



[Figure 29: High-level Flow chart]

Above Figure 27 shows the high-level flow of the Code in the PIC16F1788. The program starts in the mainloop. Mainloop sends UART data. After the data is sent, the Processes for interpreting data are carried out. The scram bit is the first bit checked before continuing the operation. The SCRAM bit is only set when a disastrous condition occurs. Then ADC data begins to be processed on all the ADC inputs. The servo is updated. Then the UART data is sent again. Interrupts are enabled throughout the operation of the mainloop. See the interrupt section to see how the interrupts work.

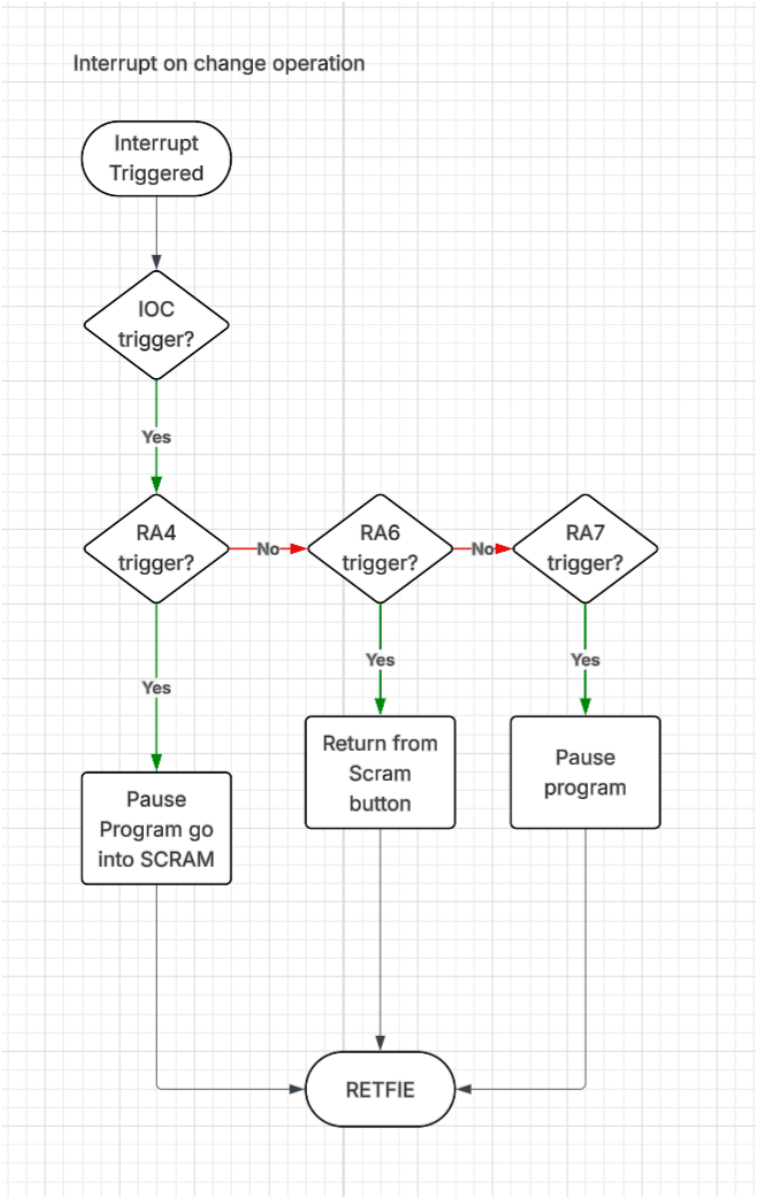


[Figure 30: Mainloop Flow Chart]

The mainloop primarily focuses on preparing data to send through the UART. First, it checks if any manual commands are received on the RX pin. If it receives info,those commands take priority over any of the previous configurations. If no data was received on the RX pin, the

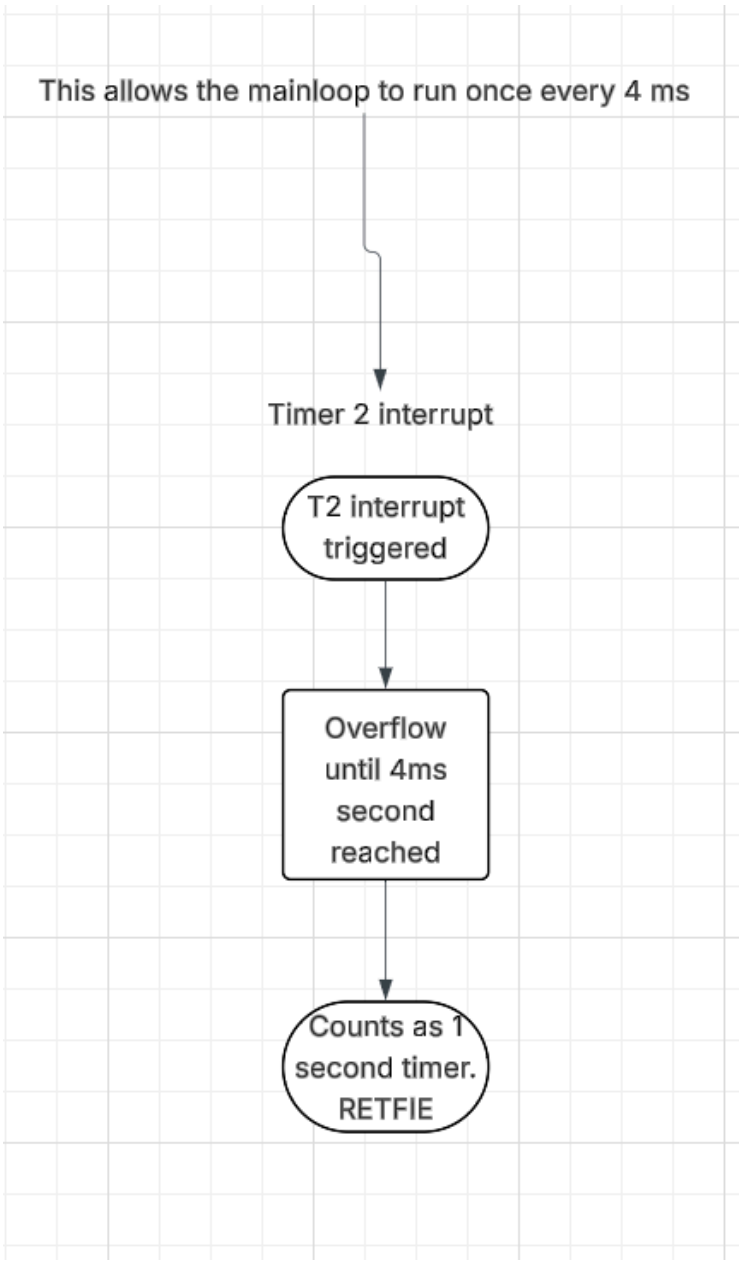
microcontroller moves through the case structure that determines which output is turned on at that specific time. See the flow chart for the cases that control the outputs.

See the end of this section for Serial protocol from flowchart (Standard TX)



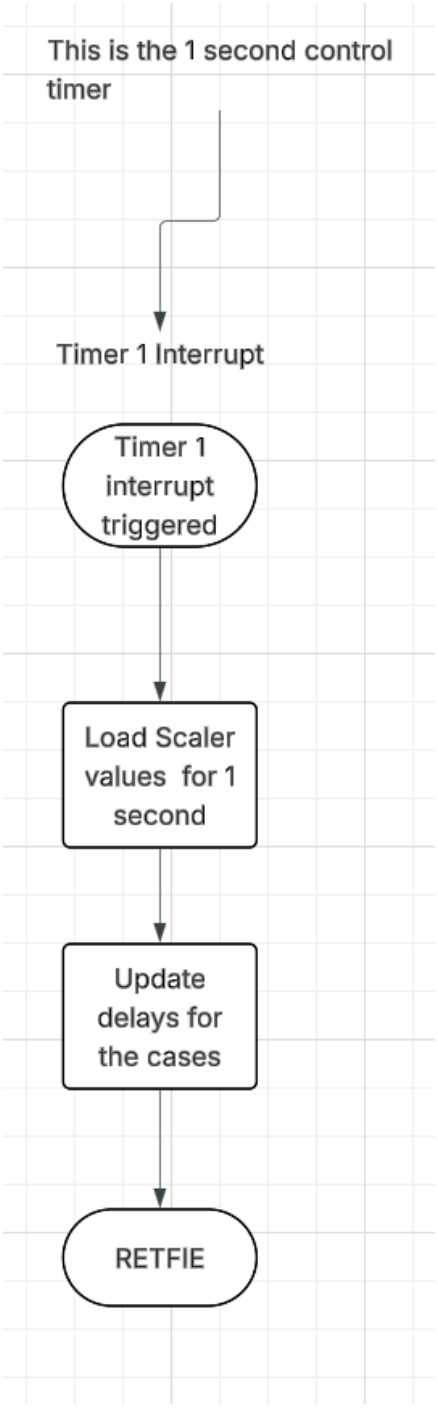
[Figure 31 Interrupt on Change Flowchart]

The figure above shows the Prioritized interrupt on the change order. The Abort function takes priority over all the other interrupts. This is for the Interrupt on change buttons.



[Figure 32 Timer 2 interrupt flow chart]

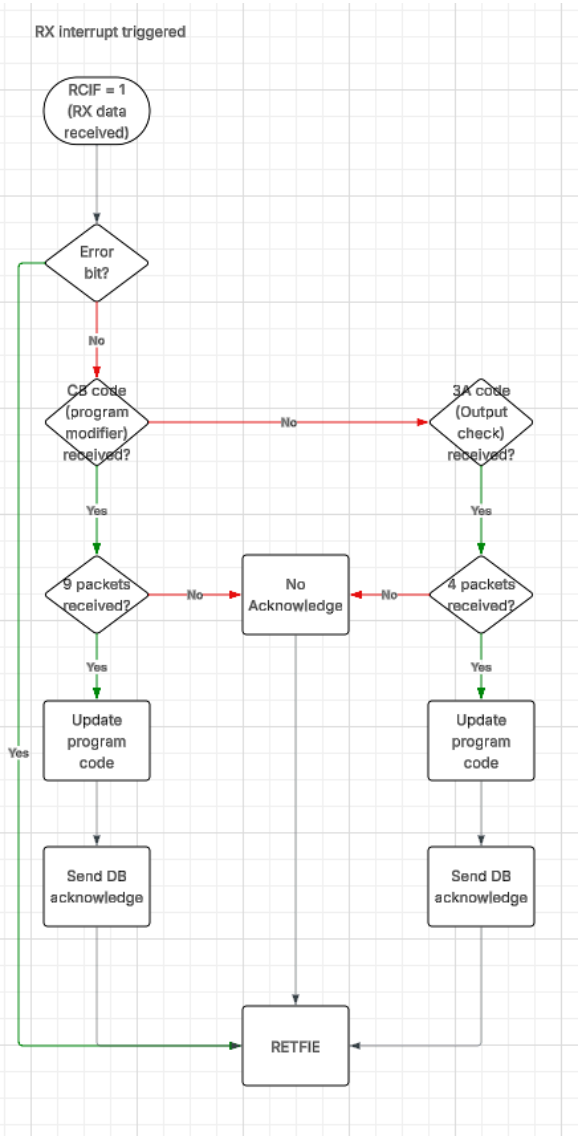
Timing is crucial in this program. Timer 2 allows the ADC to collect the data it needs within a sufficient threshold, then send that data once the mainloop has been entered.



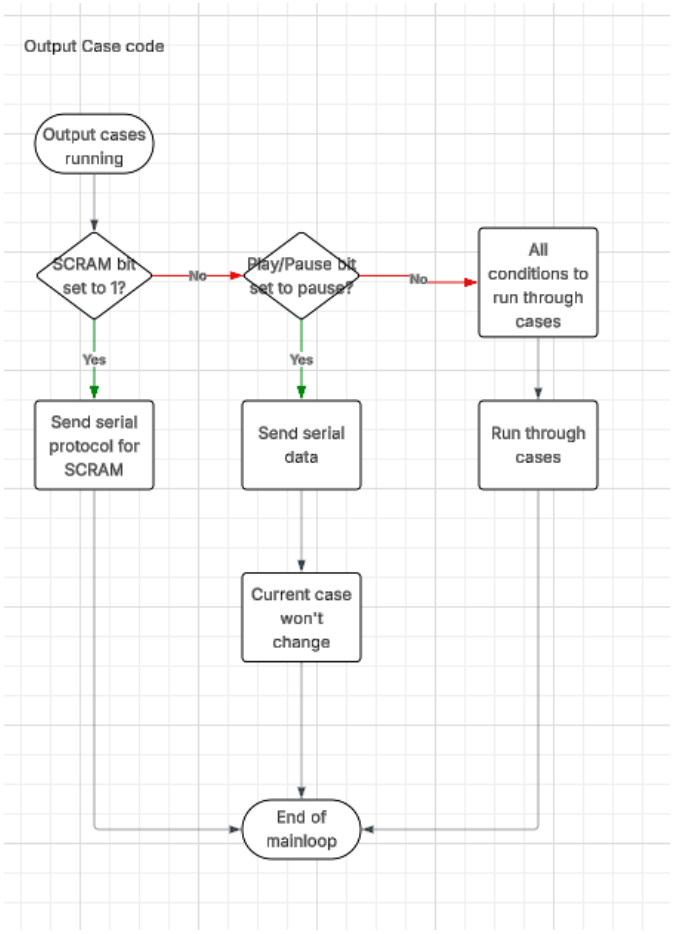
[Figure 33 Timer 1 Interrupt flowchart]

For simplified manipulation of time, the delays can be modified in one-second increments. Changes to the delays can be made simply with this adjustment.

.



[Figure 34: RX interrupt Flow Chart]



[Figure 35: Flow chart for the operation cases]

The above figure helps represent the serial protocol visually.

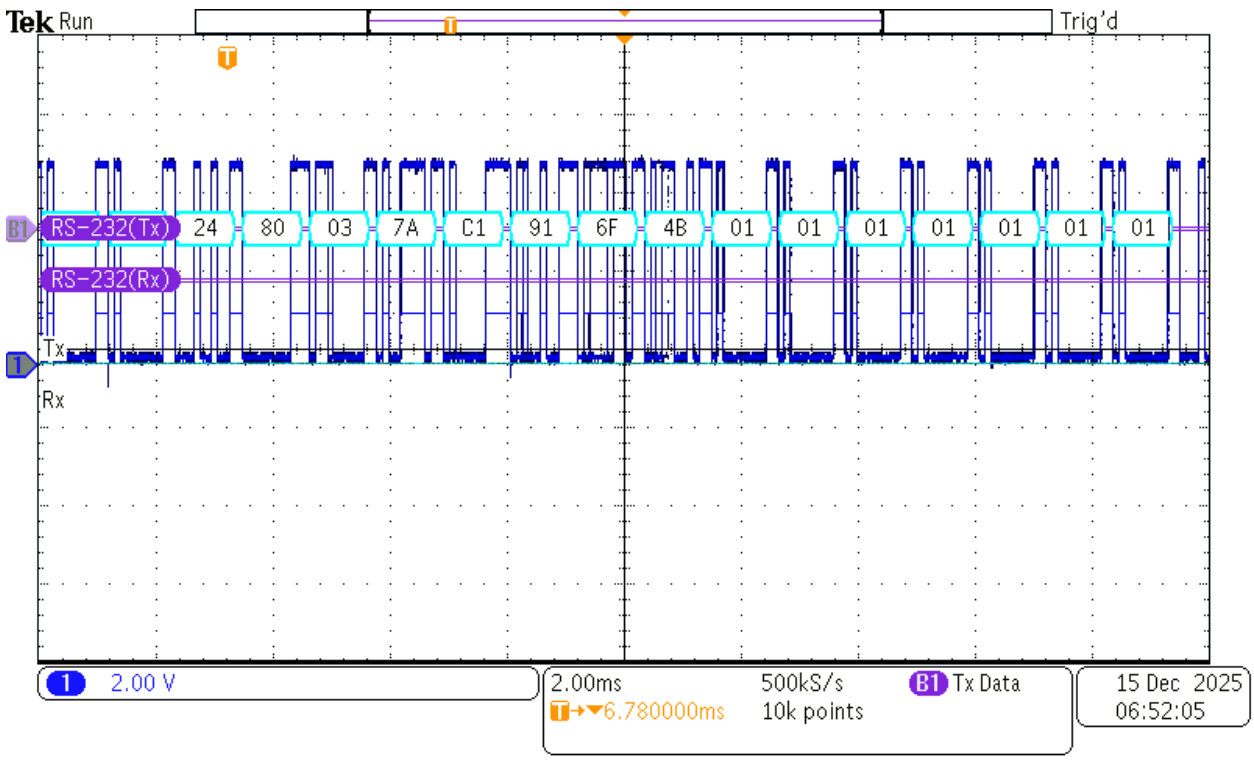
Serial protocol from flowchart (Standard TX)

The serial protocol being sent from the PIC16F1788 follows this pattern:

<Handshake><Digital Inputs><Digital Outputs><ADC 1 Data><ADC2 Data><ADC3 Data><ADC4 Data><ADC5 Data><Delay for case 1><Delay for case 2><Delay for case 3><Delay for case 4><Delay for case 5><Delay for case 6><Delay for case 7>

Refer to Figures 33 and 34 for what this protocol looks like.

For future implementation of UART into this system, the above protocol should be followed exactly. If there are packets with undefined values, the microcontroller shouldn't change any of the values. But, testing has proved that the values sporadically change when fewer than the specified amount are sent.



[Figure 36 Example serial protocol from screenshot]

Tektronix MDO3014, version v1.18, serial number C021017		
Bus Definition: RS232		
Time	Tx	Rx
-3.53E-02	24	
-3.41E-02	A0	
-3.30E-02	F0	
-3.18E-02	85	
-3.07E-02	EA	
-2.95E-02	AD	
-2.84E-02	8B	
-2.72E-02	5A	
-2.61E-02	1	
-2.49E-02	1	
-2.38E-02	1	
-2.27E-02	1	
-2.15E-02	1	
-2.04E-02	1	
-1.92E-02	1	
-1.81E-02	24	
-1.69E-02	A0	
-1.58E-02	F0	
-1.46E-02	85	
-1.35E-02	EA	
-1.23E-02	AD	
-1.12E-02	8C	
-1.01E-02	40	
-8.90E-03	1	
-7.76E-03	1	
-6.61E-03	1	
-5.46E-03	1	
-4.32E-03	1	
-3.17E-03	1	
-2.03E-03	1	
-8.80E-04	24	
2.60E-04	A0	

[Figure 37 Basic Serial Protocol on a spreadsheet with timing.]

Bit mapping for the digital inputs

Data bits: xxxx xxxx

7654 3210

Bit 7: Return from SCRAM and go to normal operation

Bit 6: SCRAM button. Works as an emergency shutoff

Bit 5: Acts as a Pause/Go button

Bit 4: unassigned

Bit 3: Unassigned

Bit2: Unassigned

Bit1: Unassigned

Bit0: Unassigned

This byte is stored in the General Purpose Registers. This allows for universal access across any bank in the PIC microcontroller. A huge advantage of this is that multiple functions can change the bits in the byte. For example, a button press can change the bit for the SCRAM condition. Or the UART can change that same bit in the byte. It decouples the operation in a way. It allows for more efficient usage and was implemented with future development in mind.

Bit mapping for digital Outputs

Data bits: xxxx xxxx

7654 3210

Bit 7: ON/OFF for the pump

Bit 6: ON/OFF for the heating element

Bit 5: ON/OFF Open/Close position for Servo

Bit 4: ON/OFF for the Gate lights

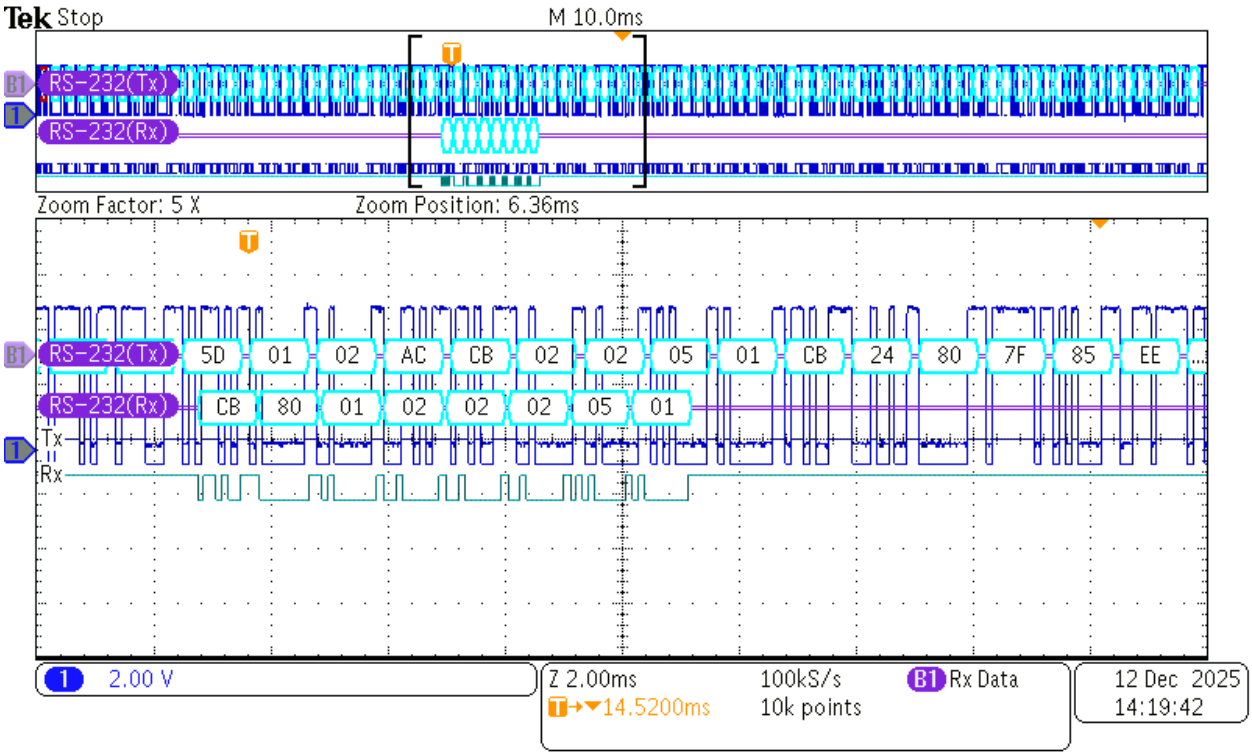
Bit 3: ON/OFF for the Reservoir valve

Bit 2: ON/OFF for the City Lights

Bit 1: ON/OFF for the Plains drain

Bit 0: Unassigned output bit

Digital outputs are located at a separate address of the general-purpose registers. They can be changed either through the UART or the physical buttons.



[Figure 38: Serial protocol with timing in spreadsheet (Changes outputs, input bits, and delays)]

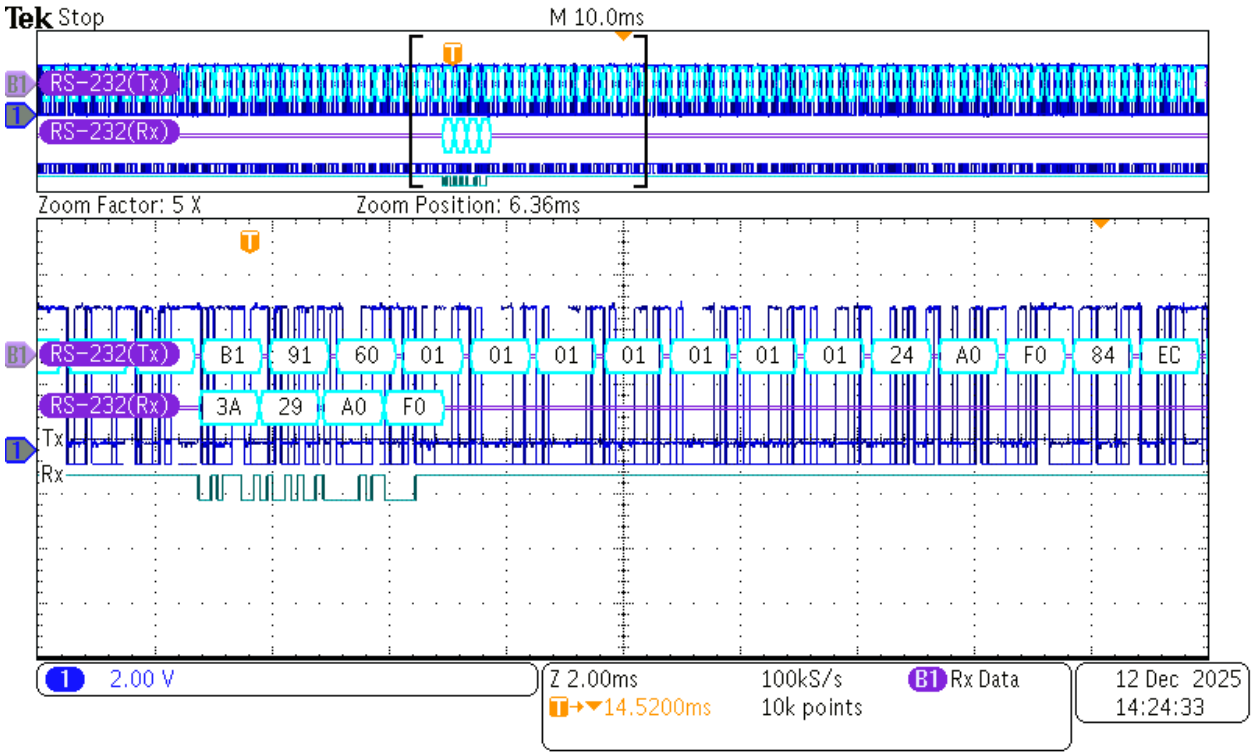
30	-6.31E-03		CB	
31	-5.47E-03	1		
32	-5.25E-03		80	
33	-4.32E-03	CB		
34	-4.19E-03		1	
35	-3.18E-03	AC		
36	-3.13E-03		2	
37	-2.07E-03		2	
38	-2.03E-03	CB		
39	-1.01E-03		2	
40	-8.90E-04	24		
41	5.00E-05		5	
42	2.60E-04	80		
43	1.12E-03		1	
44	1.40E-03	3		
45	2.55E-03	86		
46	3.69E-03	EB		
47	4.84E-03	AE		
48	5.98E-03	8D		
49	7.13E-03	5B		
50	8.27E-03	1		
51	9.42E-03	2		
52	1.06E-02	2		
53	1.17E-02	2		
54	1.29E-02	5		
55	1.40E-02	1		
56	1.52E-02	CB		

[Figure 39: Manual Debug for Setting Outputs]

Serial Protocol for Changing the inputs and the delays for each case is proven to change the packets, and when watched, changes the delays on the physical ports associated. The changes aren't saved in EEPROM. So any changes

<Change indicator><Digital inputs><Delay1><Delay2><Delay3><Delay4><Delay5><Delay6>

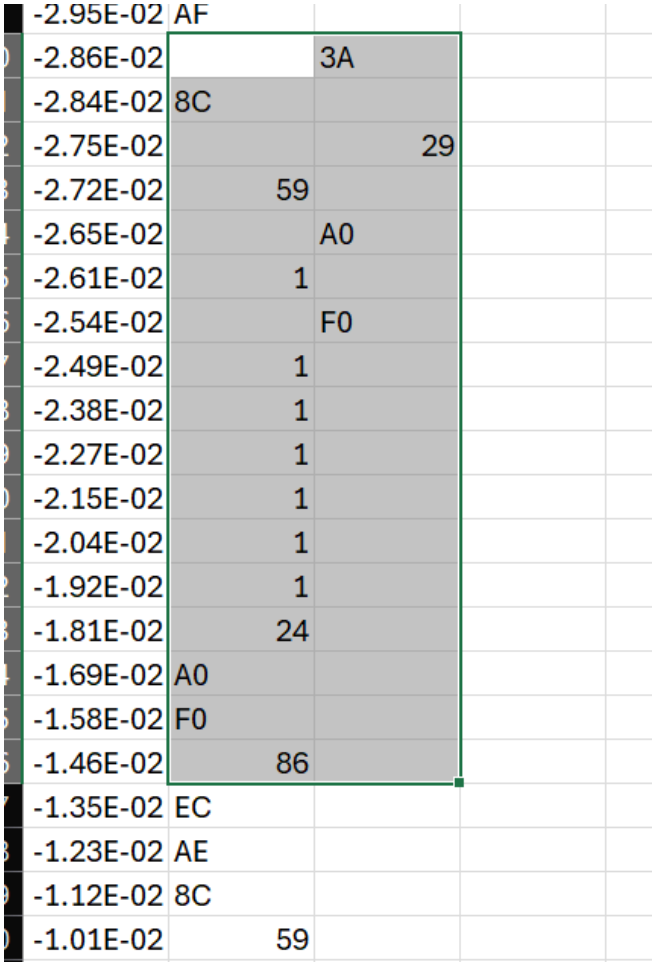
This serial protocol allows for full customization of delays and registers. The written data from the UART is not written to the EEPROM. So for



[Figure 40: Spreadsheet for the manual output change]

Serial data bits to send for Debug mode:

<Handshake part1><Handshake part2><Digital Inputs><Digital Outputs>



	-2.95E-02	AF	
	-2.86E-02		3A
	-2.84E-02	8C	
	-2.75E-02		29
	-2.72E-02	59	
	-2.65E-02		A0
	-2.61E-02	1	
	-2.54E-02		F0
	-2.49E-02	1	
	-2.38E-02	1	
	-2.27E-02	1	
	-2.15E-02	1	
	-2.04E-02	1	
	-1.92E-02	1	
	-1.81E-02	24	
	-1.69E-02	A0	
	-1.58E-02	F0	
	-1.46E-02	86	
	-1.35E-02	EC	
	-1.23E-02	AE	
	-1.12E-02	8C	
	-1.01E-02	59	

[Figure 41: Timing Diagram For Serial Example]

Engineering Log

This is the total account of worked hours for all members of our team.

			Christopher		Jacob		Noah	
Week	Day	Date	Hours	Week Tot	Hours	Week Tot	Hours	Week Tot
First	Mon	17-Nov	0		2			
	Tue	18-Nov	1.5		5		2	
	Wed	19-Nov	3		6.5		3	
	Thu	20-Nov	2		5		2	
	Fri	21-Nov	2.5		6.5		2	
	Sat	22-Nov	0		0			
	Sun	23-Nov	0	9	0			9
Second	Mon	24-Nov	0		0			
	Tue	25-Nov	0		0	3		
	Wed	26-Nov	0		0			
	Thu	27-Nov	0		0			
	Fri	28-Nov	0		0			
	Sat	29-Nov	0		0			
	Sun	30-Nov	0	0	0	0		0
Third	Mon	1-Dec	2.5		2.5	6	2	
	Tue	2-Dec	1		1	5	5	
	Wed	3-Dec	3		3	5	4	
	Thu	4-Dec	1		1	6	4	
	Fri	5-Dec	4		4	5		
	Sat	6-Dec	0		0			
	Sun	7-Dec	2	13.5	2			15
Fourth	Mon	8-Dec	6		6	5	5	
	Tue	9-Dec	5		7	6	5	
	Wed	10-Dec	8		8	6	5	
	Thu	11-Dec	5		5	4	5	
	Fri	12-Dec	8		8	5	5	
	Sat	13-Dec				1		
	Sun	14-Dec		32				25
Fifth	Mon	15-Dec	11		11	11	11	
	Tue	16-Dec	10		10	10	10	
	Wed	17-Dec	4		4		4	
	Thu	18-Dec						
	Fri	19-Dec						
	Sat	20-Dec						
	Sun	21-Dec		25		25		25
			Total:	79.5	Total:	103	Total:	74

[Figure 42: Team Members Engineering Log]

Cost Analysis

The total cost of the project was calculated based on the cost of parts and the cost of labor hours of all the team members.

Total Parts Cost: $570.55

Total Labor Cost: $12,875

Total Project Cost: $13,455.55

#	Part Name	Part Description	Quantity	it Cost (US	ended Cost (US	Notes / Function
1	14 AWG Stranded Wire (Red	Stranded copper hookup wire, 14 AWG	50 ft	$0.75 / ft	$37.50	High-current positive distribution
2	14 AWG Stranded Wire (Blac	Stranded copper hookup wire, 14 AWG	24 ft	$0.75 / ft	$18.00	High-current ground distribution
3	20 AWG Solid-Core Wire	Solid-core copper wire for signal routing	~5 ft	$0.30 / ft	$1.50	Low-current control and logic wiring
4	Arcity Multi-Output Power Su	AC–DC power supply with 5 V, 12 V, and	1	$45.00	$45.00	Primary system power source
5	Abort Switch	Panel-mounted emergency stop / kill sw	1	$12.00	$12.00	Manual system shutdown
6	NEMA 5-15 Power Connecto	Standard grounded AC power inlet	1	$6.00	$6.00	Interface to wall power
7	Outdoor Junction Box	Weather-resistant electrical enclosure	1	$30.00	$30.00	Houses power distribution hardware
8	Fuse Box (4-slot)	DIN-mounted fuse holder	2	$12.00	$24.00	Overcurrent protection per rail
9	2 A Blade Fuses	Replaceable automotive-style fuses	8	$1.00	$8.00	Circuit protection
10	DIN Rail Terminal Blocks	Individual terminal junctions	12	$2.00	$24.00	Organized wire termination
11	DIN Rail	Standard 35 mm DIN rail, ~ 8 in	1	$8.00	$8.00	Mounting structure for terminals
12	Mounting Screws	Assorted machine screws	5	$0.50	$2.50	Mechanical mounting
13	Servo Motor	Standard hobby servo	1	$15.00	$15.00	Spillway gate actuation
14	Solenoid Valve	Electrically actuated water valve	1	$18.00	$18.00	Reservoir flow control
15	Water Pump	Low-voltage DC water pump	1	$20.00	$20.00	Water circulation
16	Heating Element	Resistive heating element	1	$25.00	$25.00	Snowmelt simulation
17	Polyethylene Tubing	Flexible water tubing	~10 ft	$1.50 / ft	$15.00	Fluid transport
18	Styrofoam Boards	Rigid foam insulation boards	~6 ft × 4 ft	$25.00	$25.00	Physical terrain modeling
19	PVC Pipe	Rigid PVC pipe	~3 ft	$5.00	$5.00	Drainage and flow routing
20	Metal Wire Shelving Rack	Steel wire shelving unit	1	$60.00	$60.00	Structural support for model
21	Mega 2560 R3 ATmega2560	Microcontroller board	1	$27.99	$27.99	HMI
22	sainsmart TFT_320QVT scre	TFT LCD	1	$24.10	$24.10	HMI
23	H2W	PCB	5	$4.80	$24.00	PCB components
24	Capacitor_SMD:C_0805_201	10uF	2	$0.70	$1.40	PCB components
25	Capacitor_SMD:C_0805_201	1uF	2	$0.70	$1.40	PCB components
26	Capacitor_SMD:C_0805_201	1uF	2	$1.39	$2.78	PCB components
27	LED_SMD:LED_0805_2012M	LED	3	$0.10	$0.30	PCB components
28	Diode_SMD:D_SOD-123	1N4148	1	$1.72	$1.72	PCB components
29	Pinheader	SDA	2	$1.72	$3.44	PCB components
30	Pinheader	SCL	2	$1.72	$3.44	PCB components
31	Pinheader	Debug 0	2	$1.72	$3.44	PCB components
32	Pinheader	Debug 1	2	$1.72	$3.44	PCB components
33	Pinheader	Debug 2	2	$1.72	$3.44	PCB components
34	Pinheader	Debug 3	2	$1.72	$3.44	PCB components
35	Pinheader	Debug 4	2	$1.72	$3.44	PCB components
36	Pinheader	Debug 5	2	$1.72	$3.44	PCB components
37	Pinheader	Debug 6	2	$1.72	$3.44	PCB components
38	Pinheader	Debug 7	2	$0.86	$1.72	PCB components
39	Connector_PinHeader_2.54r	PWM PWR	1	$2.61	$2.61	PCB components
40	TerminalBlock:TerminalBlock	Power	1	$0.86	$0.86	PCB components
41	Connector_PinHeader_2.54r	AN_0	1	$0.86	$0.86	PCB components
42	Connector_PinHeader_2.54r	AN_1	1	$0.86	$0.86	PCB components
43	Connector_PinHeader_2.54r	AN_2	1	$0.86	$0.86	PCB components
44	Connector_PinHeader_2.54r	AN_3	1	$0.86	$0.86	PCB components
45	Connector_PinHeader_2.54r	Port A	1	$0.86	$0.86	PCB components
46	Connector_PinHeader_2.54r	Port B	1	$0.86	$0.86	PCB components
47	Connector_PinHeader_2.54r	Port C	1	$0.86	$0.86	PCB components
48	Connector_PinHeader_2.54r	PWM1	1	$0.86	$0.86	PCB components
49	Connector_PinHeader_2.54r	PWM2	1	$0.86	$0.86	PCB components
50	Connector_PinHeader_2.54r	PWM3	1	$0.86	$0.86	PCB components
51	Connector_PinHeader_2.54r	PWM4	1	$0.86	$0.86	PCB components
52	Connector_PinHeader_2.54r	Program	1	$0.86	$0.86	PCB components
53	Connector_PinHeader_2.54r	USART	1	$0.86	$0.86	PCB components
54	Connector_PinHeader_2.54r	I²C	2	$0.86	$1.72	PCB components
55	Connector_PinHeader_2.54r	5V	1	$0.86	$0.86	PCB components
56	Connector_PinHeader_2.54	mm:PinHeader_1x02_P2.54mm_Vertical	1	$0.86	$0.86	PCB components
57	Connector_PinHeader_2.54r	VCC_PWM Disc	1	$0.86	$0.86	PCB components
58	Connector_PinHeader_2.54r	Regulator Bypass	1	$0.18	$0.18	PCB components
59	Resistor_SMD:R_0805_2012	2.2k	1	$0.22	$0.22	PCB components
60	Resistor_SMD:R_0805_2012	4.7k	20	$0.12	$2.40	PCB components
61	Resistor_SMD:R_0805_2012	10k	1	$0.95	$0.95	PCB components
62	Package_TO_SOT_SMD:SC	TSB1117-33	1	$5.00	$5.00	PCB components
63	Package_SO:SSOP-28_5.3x	PIC16F1788-xSS	1	$2.50	$2.50	PCB components
64	Crystal:Crystal_HC49-U_Ver	Crystal	1	$2.72	$2.72	PCB components
65	DMG3402L-7	FET	8	$2.72	$21.76	PCB components
					$570.55	TOTAL

Conclusion/Recommendations

The Objectives of this project were to create a scaled interactive physical model simulating climate-driven ice and snow melt in a mountain watershed. The heating element was not implemented due to the heating element drawing too much current. The water pump, moving the water, was implemented successfully with adjustable timers. Successful implementation of the hardware, software, and electromechanical components was achieved. Proper safety considerations were planned. Implementation for future safety protocols and designs is encouraged, as the present hardware placement places the electronic components in a vulnerable position. Recommendations for further reconstruction of the styrofoam to appear more natural are encouraged. The positioning of the pump should be reconsidered as it is placed above the electronics box and creates a potential hazard during maintenance.

Many improvements can be made to the code. For example, there could be a delay implemented for the serial packets so that the external hardware can read the data packets better. There is one glitch with ADC packet 5, which is currently affected by ADC4. There are a couple more ports available for additional functions to be connected. The PCB can be modified. The program pins need to be verified, as there are currently connections that aren't consistent across the board. The RX and TX pins are correct, but the programming pins are not correct. Programming can still occur, but a ribbon cable must be connected to the programmer for consistent programming. The silkscreen could be updated for easier discerning of ports and their associated connections.