

Wetris: CPE 301 Final Project

Team Wetris (Team 42)

December 12, 2025

Contents

1	Project Description	3
2	Cost Analysis for this First Time Build	3
3	Component Details	4
3.1	Arduino Mega 2560	4
3.2	LCD Display (ST7796S controller)	5
3.3	Door Lock Actuator	5
3.4	Fuse	5
3.5	Potentiometer	5
3.6	Buzzers, Resistors, and Capacitors	5
3.7	Relay Module	5
3.8	Optocouplers	6
4	System Overview	6
4.1	Hardware Architecture	6
4.1.1	Microcontroller Resources Table	6
4.1.2	LCD Tetris Module	7
4.1.3	Music Module	7
4.1.4	Water Gun Module	7
4.1.5	Potentiometer Module	8
4.2	Software Architecture	8
4.3	Power	8
5	Images	9
6	Schematic Diagrams	16
6.1	High Level System Schematic	17
6.2	Water Gun Module Schematic	18
6.3	Music Module Schematic	19
6.4	LCD and Button Module Schematic	20
6.5	Potentiometer Module Schematic	21
6.6	3D Casing Technical Drawings	22

7	Links	24
7.1	Repository	24
7.2	Documentation	24
7.3	Demo Videos	24
7.4	3D Printing Files	24
8	Environmental Impact as Detailed in CPE 301 Grading Rubric	25
8.1	Energy Efficiency	25
8.2	Design Safety	25
8.3	Affordability	25
8.4	Sustainability	25
8.5	Accessibility	25
9	Our Team	26
9.1	Gabriel Jordaan	26
9.2	Dieufainson Jean	26
9.3	Jorge Robles	27
9.4	Roman Rosburg	27
10	Acknowledgments	28
11	Wetris Logo	29

1 Project Description

Wetris is a suspenseful and slightly nerve-racking spin-off of Tetris, the classic block-stacking game. The player races to clear enough lines to reach the target score before their stack reaches the top, and if they fail, they get splashed with water! To make it even better, the spring-powered water gun is very loud and is a guaranteed jump scare even if you know it's coming. It makes for a great party game, and would go well as an attraction in an arcade. The game also features a water gun override option that can be accessed by sending an 'x' to the microcontroller via UART so you can prank your friends who look like they are going to exceed the target score and are expecting to stay dry.

2 Cost Analysis for this First Time Build

This cost analysis includes tools that will not need to be repurchased for future builds, such as the multimeter and crimping tool. These tools were necessary for the successful completion of this first build, and their costs are included to provide a comprehensive overview of the initial investment required for the Wetris project. A production version of Wetris could be built for significantly less, since the tools are one-time purchases, and many of the components were purchased in higher quantities than necessary and the cost for those components could be spread across multiple builds. The largest cost would still be the 3D printed casing, but even that could be reduced with bulk printing or alternative manufacturing methods such as injection molding. A cost analysis for a production version is not included here, but may later be developed if there is interest in pursuing that avenue.

Material	Total Price	Notes
3D printed casing	\$263.52	UNR LIBRARIES TRANSACTION 201937 DLM
Arduino ATmega 2560	\$54.02	Amazon
Multimeter	\$34.63	Amazon
Crimping tool	\$34.63	Amazon
LCD Display	\$22.72	Amazon (Jorge also bought us a second one for testing)
Wire stripper	\$20.56	ACE hardware
Door lock actuator	\$17.05	Amazon
Relay Module	\$14.42	Amazon
Syringes	\$14.06	Amazon
12V power adapter	\$14.06	Amazon
Extension cord	\$12.98	Amazon
Caulk gun	\$10.82	Newborn DC012 from ACE hardware
Pin connector for actuator	\$10.27	Amazon
Electrical tape	\$9.30	ACE hardware
Wire	\$8.66	ACE hardware
Inline fuse holder	\$7.13	ACE hardware
Buzzers	\$6.49	Amazon
Sprinkler head	\$5.40	ACE hardware
Fuses	\$5.40	Amazon
Barrel jack splitter	\$4.30	Amazon
Misc. we had on hand	\$0.00	Jumper wires, buttons, tools, resistors, breadboard, 9V battery, USB cable, optocouplers etc.
Epoxy	\$0.00	Took a little from another project of Gabe's
Grand Total:	\$570.43	

Table 1: Component cost breakdown for Wetris system

3 Component Details

This section provides detailed information about each major component used in the Wetris system, including their specifications and roles within the project.

3.1 Arduino Mega 2560

Central microcontroller unit (MCU) responsible for managing game logic, user inputs, and controlling peripherals. Full technical specifications can be found at:

<https://cdn.robotshop.com/media/a/ard/rb-ard-33/pdf/arduinoomega2560datasheet.pdf> for the development board and

<https://www.alldatasheet.com/datasheet-pdf/download/107092/ATMEL/ATMEGA2560.html> for the microcontroller itself.

3.2 LCD Display (ST7796S controller)

Hosyond 4.0" TN Capacitive Touch Display

Display Controller: ST7796S driver IC

Touch Controller: FT6336U capacitive touch IC (not used in project)

Size: 4.0" diagonal, 320x480 resolution

Physical Dimensions: 60.88x108.0x14.80mm (WxHxD)

Interface: 4-wire SPI (display) + I2C (touch)

Power: 5V/3.3V compatible, 0.5W consumption (We used 5V)

Connectivity: 14P 2.54mm header + FPC connector

Additional: Micro SD slot, LED backlight (300 cd/m²) (Both unused in project)

3.3 Door Lock Actuator

The door lock actuator used is intended for a Dodge Grand Caravan, and operates on 12V DC and draws 2A. In order to reverse the direction of the actuator, the polarity of the voltage applied to it must be reversed. It is energy efficient, only drawing current when moving, and has a built-in limit switch to stop current flow when fully extended or retracted. Full specifications could not be found past what the amazon page provided. Pins had to be experimentally probed to determine their proper function. Two pins still have unknown function and are left unconnected. They are possibly for feedback. More information can be found at:

https://www.amazon.com/dp/B0C43BNSQM?ref_=ppx_hzsearch_conn_dt_b_fed_asin_title_3&th=1

3.4 Fuse

Standard 3A automotive blade fuse used to protect the water gun actuator circuit from overcurrent conditions.

3.5 Potentiometer

Standard 10k ohm rotary potentiometer used to adjust game speed.

3.6 Buzzers, Resistors, and Capacitors

Standard piezoelectric buzzers used for audio output, along with 100 Ω resistors and 10 µF capacitors for signal conditioning and circuit protection.

3.7 Relay Module

Forward and Reverse Relay Module, 12V 10A Pre-Wired with LED Light, for Motor/Linear Actuator, Reversing Relay Module. More information can be found at:

https://www.amazon.com/Weasch-Forward-Pre-Wired-Actuator-Reversing/dp/B0FJQV4316?pd_rd_w=4c3s0&content_id=amzn1.sym.97c08ff8-4f39-4f9e-b360-8dbdbec34ee6&pf_rd_p=97c08ff8-4f39-4f9e-b360-8dbdbec34ee6&pf_rd_r=YHHMAAYABW9QQTN9YD5D&pd_rd_wg=pirQI&pd_rd_r=114b95df-9ba3-4874-ba37-76236030403c&pd_rd_i=B0FJQV4316&psc=1&ref_=pd_basp_d_rpt_ba_s_1_t

3.8 Optocouplers

4N35 optocouplers with a forward voltage of 1.18 V and a maximum collector-emitter voltage of 30 V used to isolate the high current 12V actuator control circuit from the low voltage 5V logic of the Arduino. More information can be found at:

<https://www.alldatasheet.com/datasheet-pdf/view/98359/FAIRCHILD/4N35.html>

4 System Overview

For this project we designed a series of peripherals to interface with the microcontroller in the Arduino Mega 2560 development board, and connected them in a 3D printed casing. The design is highly modular, with each peripheral being controlled by a library, and being handled from a high-level main loop. See code and schematics for most in-depth explanation. The peripherals are:

- The Tetris module, consisting only of an LCD, three buttons, and the software to run tetris game logic
- The music module, with three buzzers playing the tetris theme song (Korobeiniki)
- The water gun module, which consists of a syringe, which is pushed by a spring-powered plunger, let back by a door lock actuator controlled by a relay module, which is in turn controlled by opto-couplers.
- The potentiometer module, which is used to adjust the speed of the Tetris module. (Fuse protected.)

4.1 Hardware Architecture

4.1.1 Microcontroller Resources Table

The following table outlines the microcontroller resources allocated to each component of the Wetris system, and details some of the connections used. Full schematic diagrams give all connection details and are included later in this document.

Reserved Component	Name of User	Purpose
OC1A(pin11)+ Timer1	Gabe	Square wave output for music
OC3A(pin5) + Timer3	Gabe	Square wave output for music
OC4A(pin6) + Timer4	Gabe	Square wave output for music
PCINT7 (pin13)	Roman+Jorge+Gabe	Left Movement
PCINT6 (pin12)	Roman+Jorge+Gabe	Right Movement
PCINT4 (pin10)	Roman+Jorge+Gabe	Rotation Movement
Pins for LCD (pins 7-9)	Roman	LCD connections
Pins for LCD(Dig. Pins50-52)	Roman	LCD connections
Pin for LCD LED (pin2)	Roman	Backlight control
FWD Pin (22)	Son+Gabe+Jorge	Drive actuator forward
REV Pin (23)	Son+Gabe+Jorge	Drive actuator reverse
A0 (pin A0)	Son	Potentiometer for game speed adjustment

Table 2: Hardware resource allocation for Wetris system

4.1.2 LCD Tetris Module

In order to reach the back of the casing where the MCU is mounted from the LCD and buttons in the front of the casing, we used 16 AWG wire to connect the buttons, and we used Dupont wires linked end to end to connect the LCD. The LCD and button connections in detail are as follows:

- Pin 2 → LED pin on the LCD
- Pin 7 → CS pin on LCD
- Pin 8 → RESET pin on LCD
- Pin 9 → CD pin on LCD
- Pin 10 → Rotate Piece Button
- Pin 12 → Right Button
- Pin 13 → Left Button
- Pin 50 → SDO
- Pin 51 → SDI
- Pin 52 → SCK

4.1.3 Music Module

The music module uses output compare pins OC1A (pin 11), OC3A (pin 5), and OC4A (pin 6) in CTC (Clear Timer on Compare) mode to generate square wave signals at specific frequencies for three separate piezo buzzers. Each buzzer is connected in series with a 100 ohm resistor to limit current and a $10\mu F$ capacitor to smooth the signal. The code directly loads frequency and duration values into memory and iterates through them to play the song. This method is highly memory intensive, but allows for accurate timing and frequency control. The music was converted from midi format to c style arrays using a custom python script found in the repository.

4.1.4 Water Gun Module

In order to isolate the high current 12V power supply used to drive the water gun actuator from the 5V logic of the Arduino, we used a relay module controlled by opto-couplers. The relay module is powered by the 12V supply, and the opto-couplers are powered by the 5V supply from the Arduino. The opto-couplers are connected to digital pins 22 and 23 on the Arduino, which control the forward and reverse movement of the actuator respectively. When the game is over and the safety threshold has not been reached, the Arduino activates the relay in reverse to trigger the water gun mechanism. The forward function is not used in normal operation, but is included for completeness. Very importantly, an inline fuse is included in the 12V power line to protect against short circuits.

4.1.5 Potentiometer Module

The potentiometer is our most simple peripheral, being connected to the 5V and GND pins on the Arduino, with the wiper connected to analog pin A0. The potentiometer is used to adjust the speed of the game, allowing the player to customize their experience. Also allowing their friends to set it to an impossible speed in the middle of a game for extra fun.

4.2 Software Architecture

The software is structured in a modular fashion, with each peripheral being handled by its own library. The main loop coordinates the interactions between the different modules, ensuring smooth gameplay and timely responses to user inputs. All modules had to be engineered to be non-blocking and work together. This presented a difficult engineering challenge that had to be overcome with many optimisations and plan-B methodologies. For example, the LCD tetris module had to render only when the game state changed, otherwise the music module would stutter as the updates took too much time. Clearing the screen with the LCD library took too long, so we had to implement our own function to only update the parts of the screen that changed, and now we only use the clearing function when absolutely necessary, such as at the start of the game or when a row is cleared.

4.3 Power

The Wetris system needs two power sources to operate correctly. The first is power to the Arduino, which is supplied through the barrel jack port via a 9V battery (the 9V is internally regulated to 5V by the Arduino). The second power source is a 12V power supply which powers the water gun actuator via a relay module. The relay module is controlled by the Arduino through opto-couplers to isolate the two power sources.

5 Images

These images showcase the physical implementation of the Wetris system, including the circuit and various stages of gameplay.

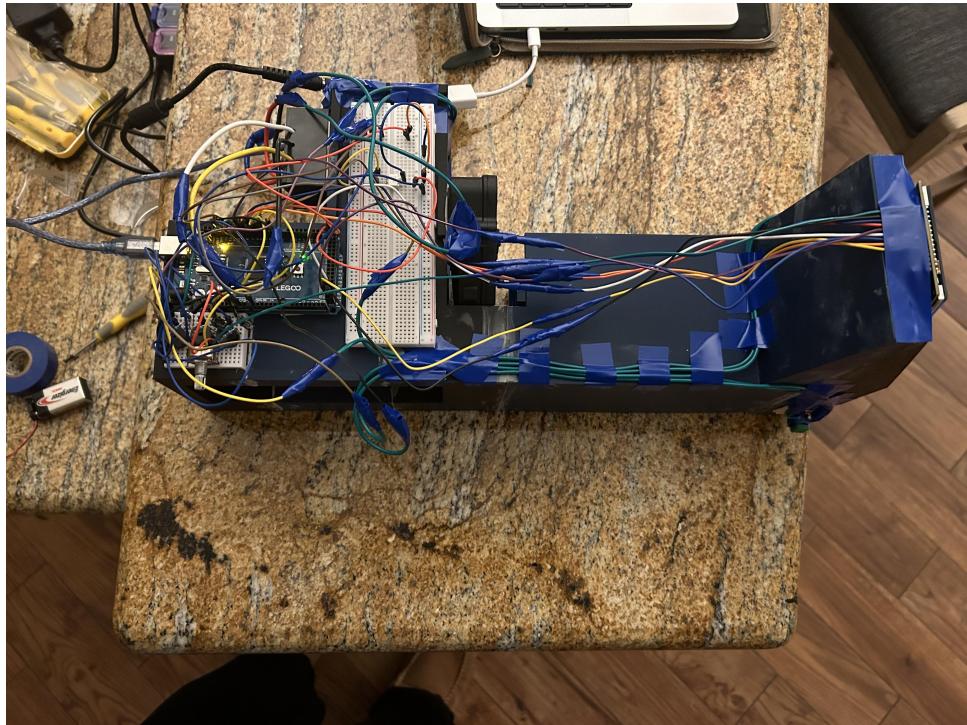


Figure 1: Physical circuit implementation of Wetris system shown from above

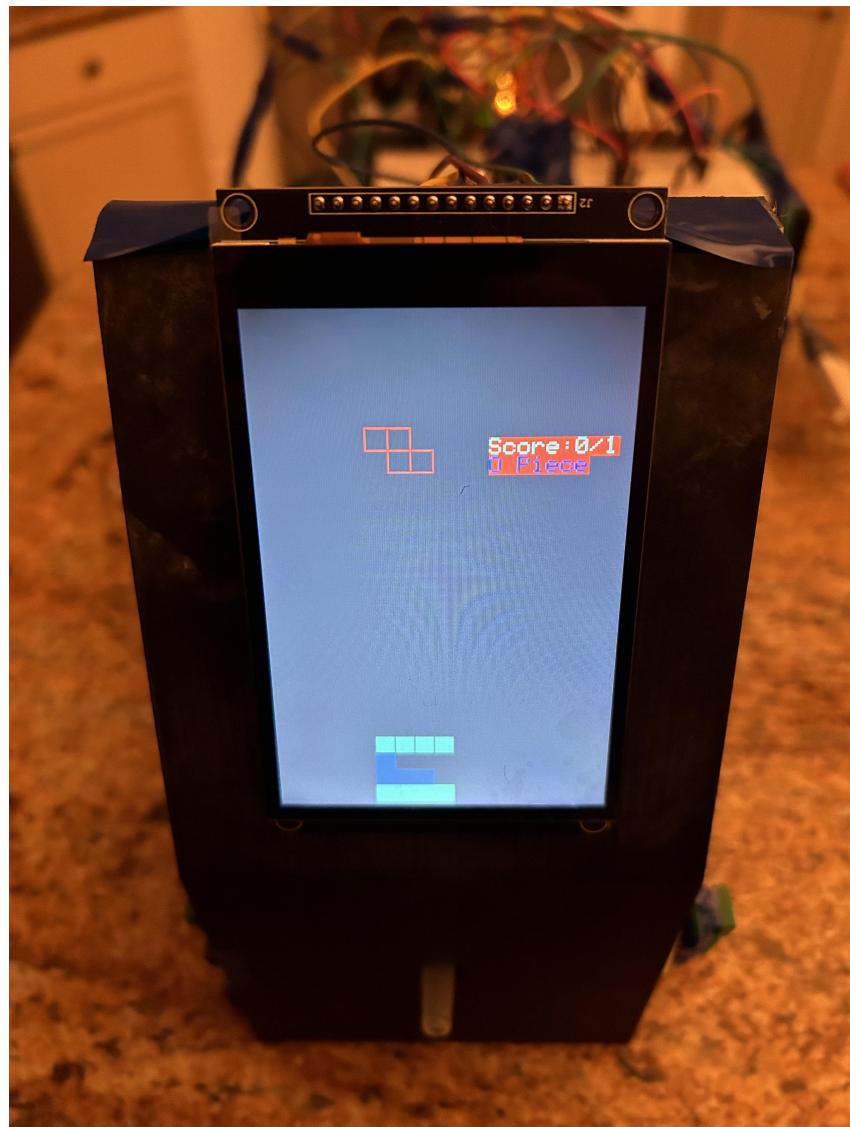


Figure 2: Normal gameplay showing Tetris blocks falling and stacking

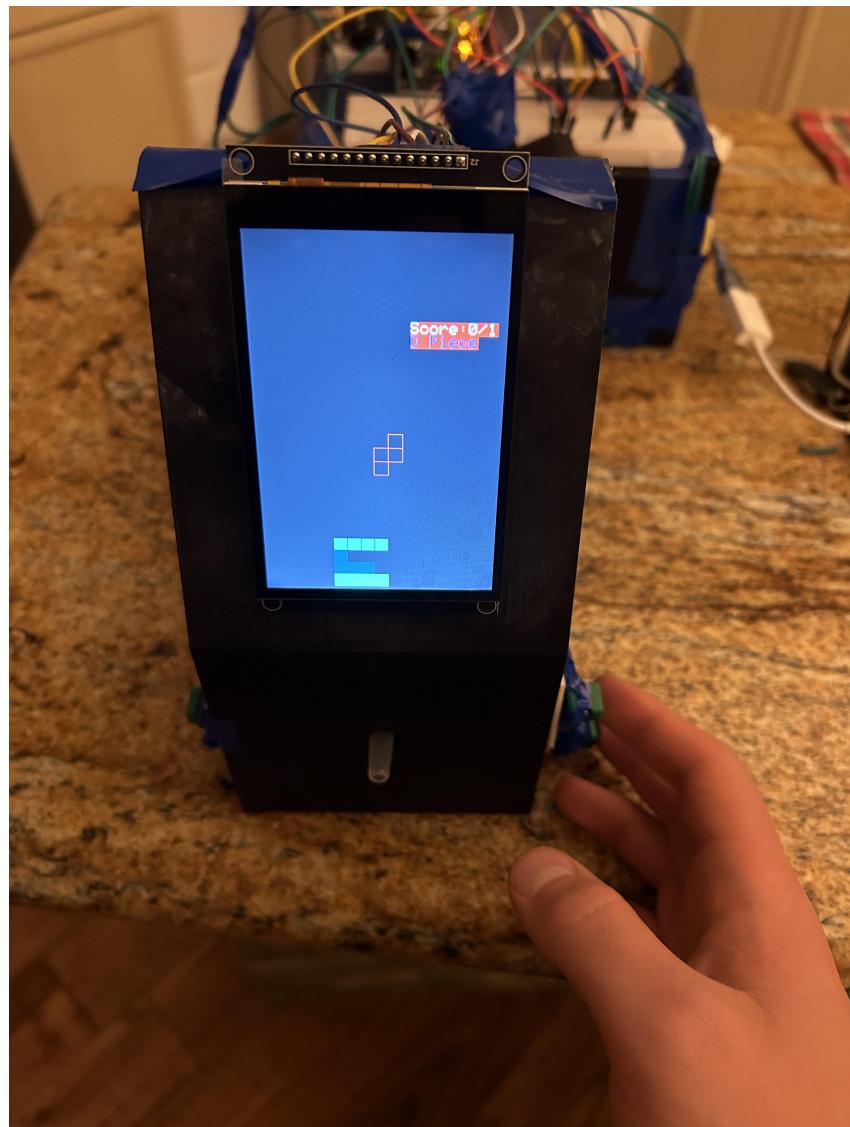


Figure 3: Player moving and rotating blocks during gameplay



Figure 4: Clearing the screen before block clearing animation when a row is completed

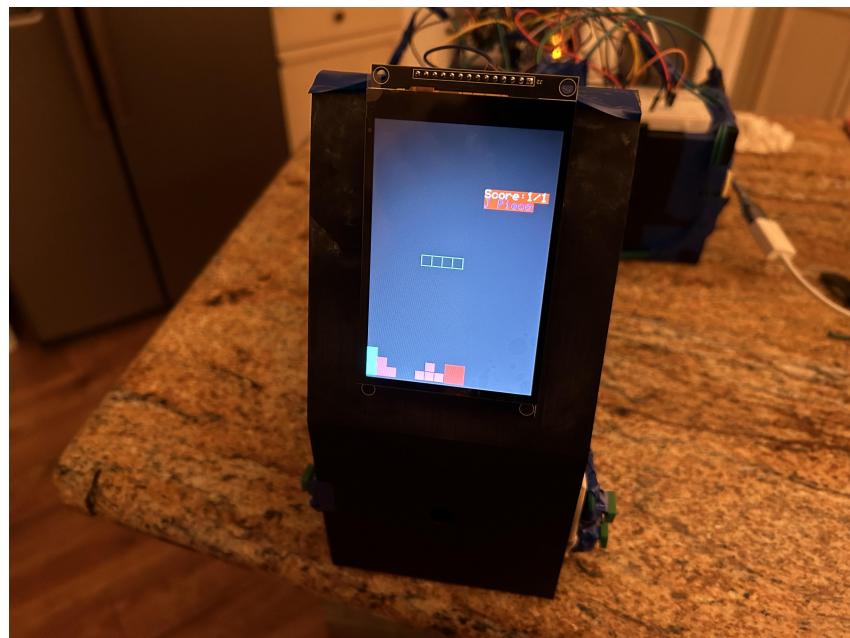


Figure 5: Player scoring a point by completing a row

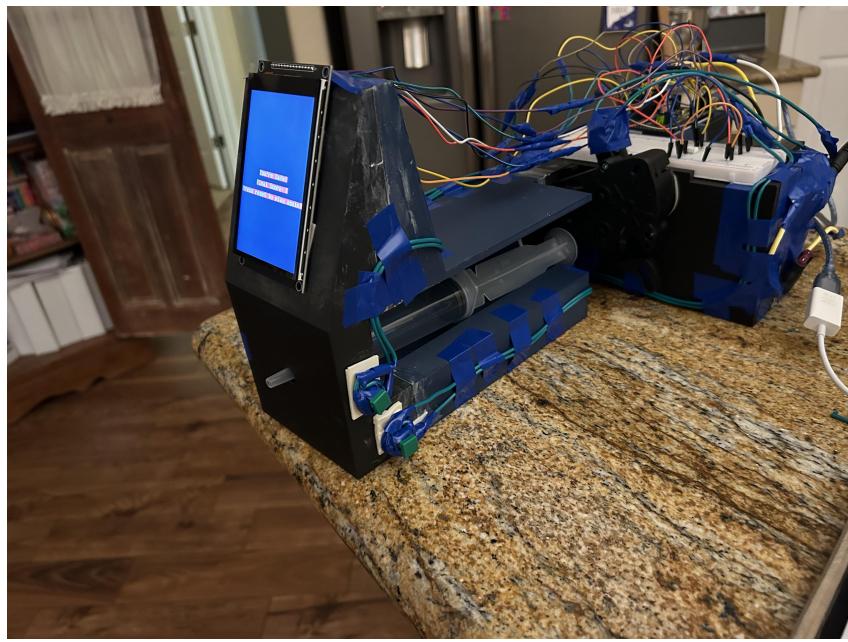


Figure 6: Victory screen displayed when safety threshold is reached after game over



Figure 7: Game over screen displayed when player loses

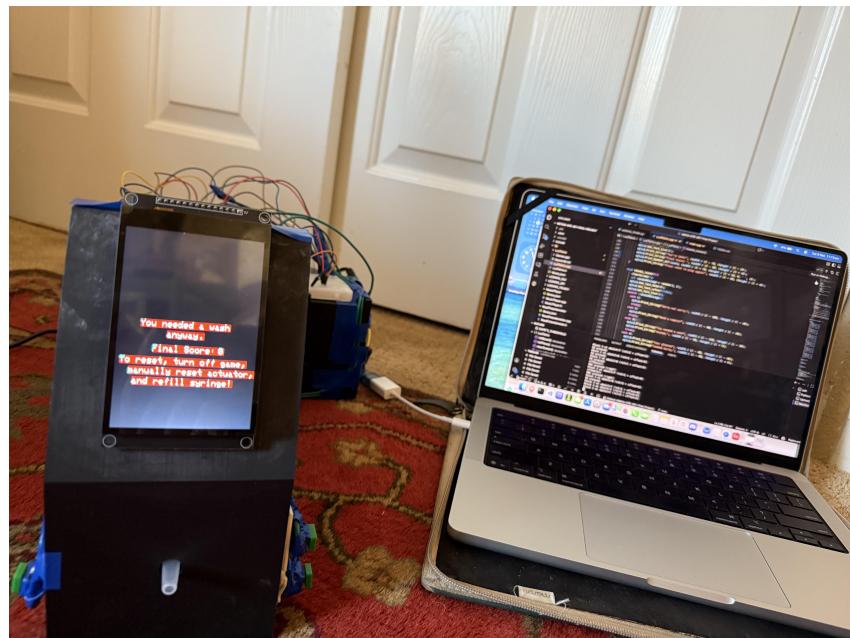


Figure 8: Game over screen with state information shown on serial monitor

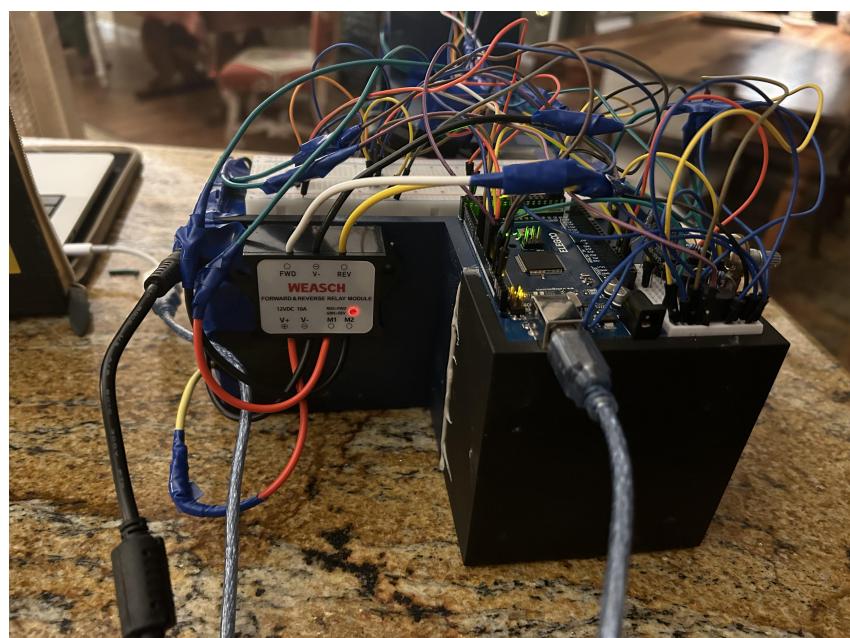


Figure 9: Relay module in activated state controlling water gun actuator

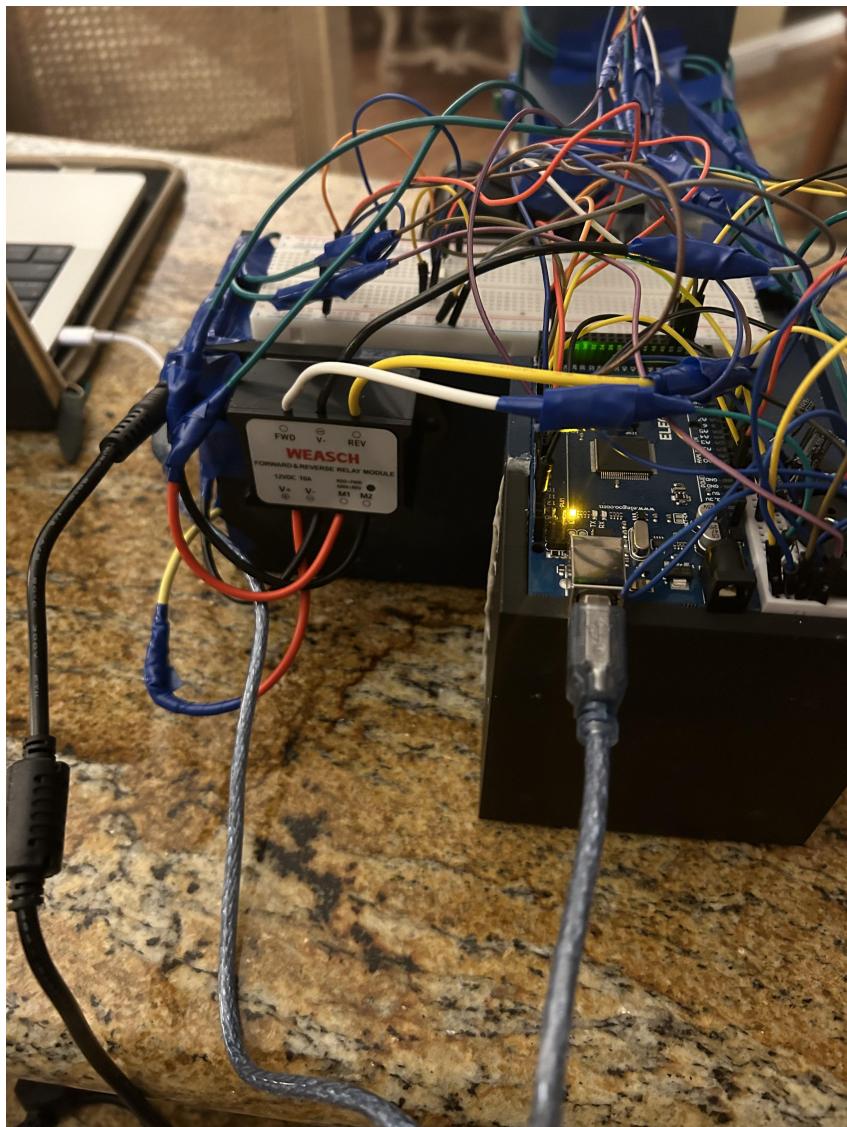


Figure 10: Relay module in deactivated state

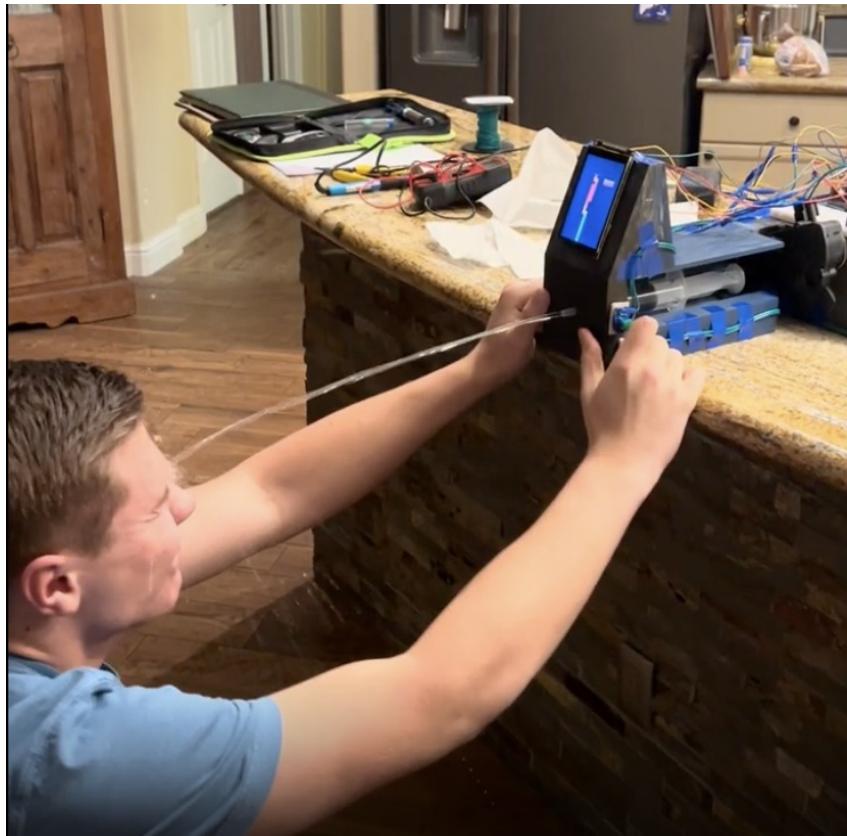


Figure 11: Player getting wet from the water gun after losing the game

6 Schematic Diagrams

Below is a schematic diagram illustrating the electrical connections and components used in the Wetris system. There is a high level diagram that shows how the peripherals are connected, and a more detailed schematic for each peripheral.

6.1 High Level System Schematic

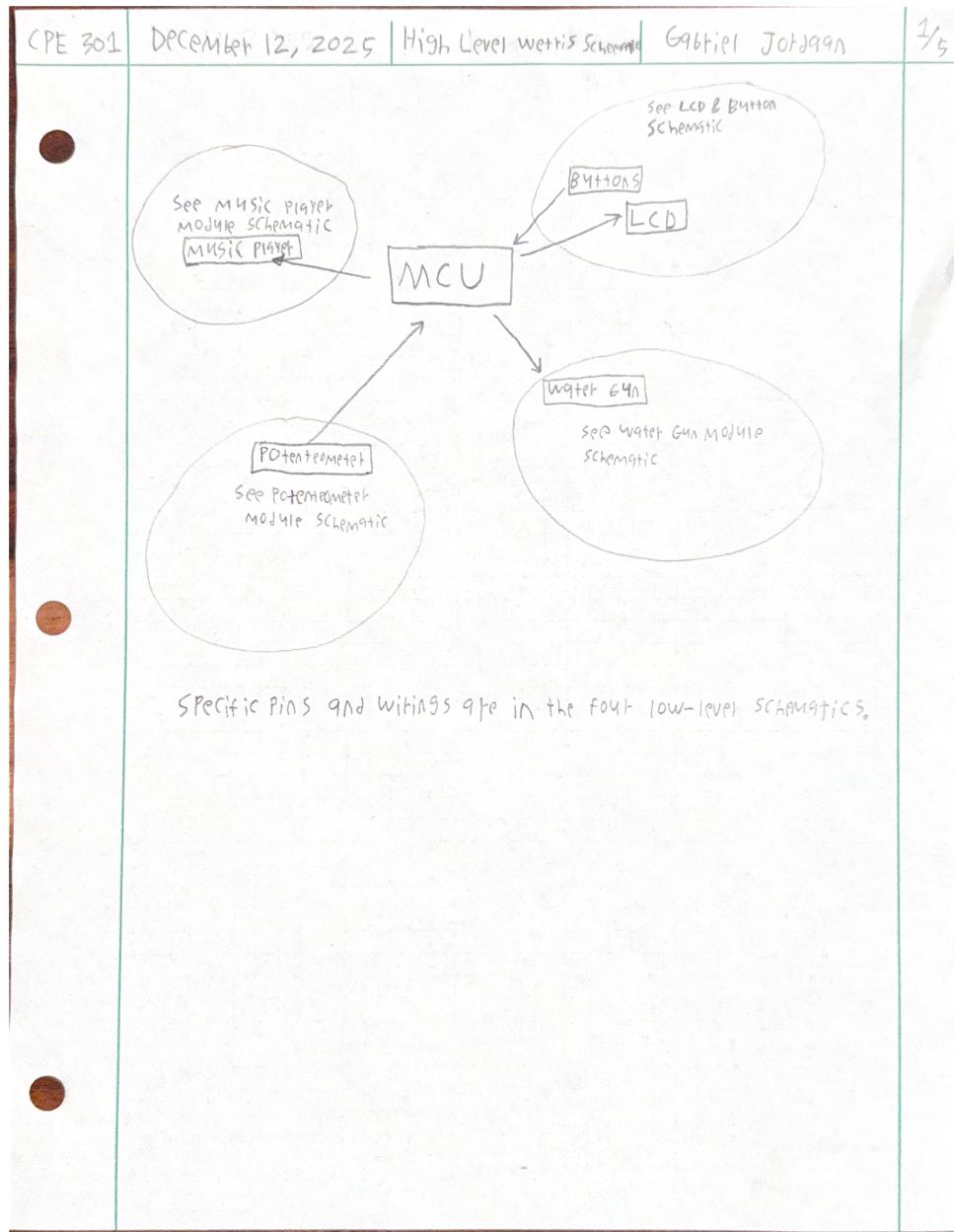


Figure 12: High level electrical schematic of Wetris system

6.2 Water Gun Module Schematic

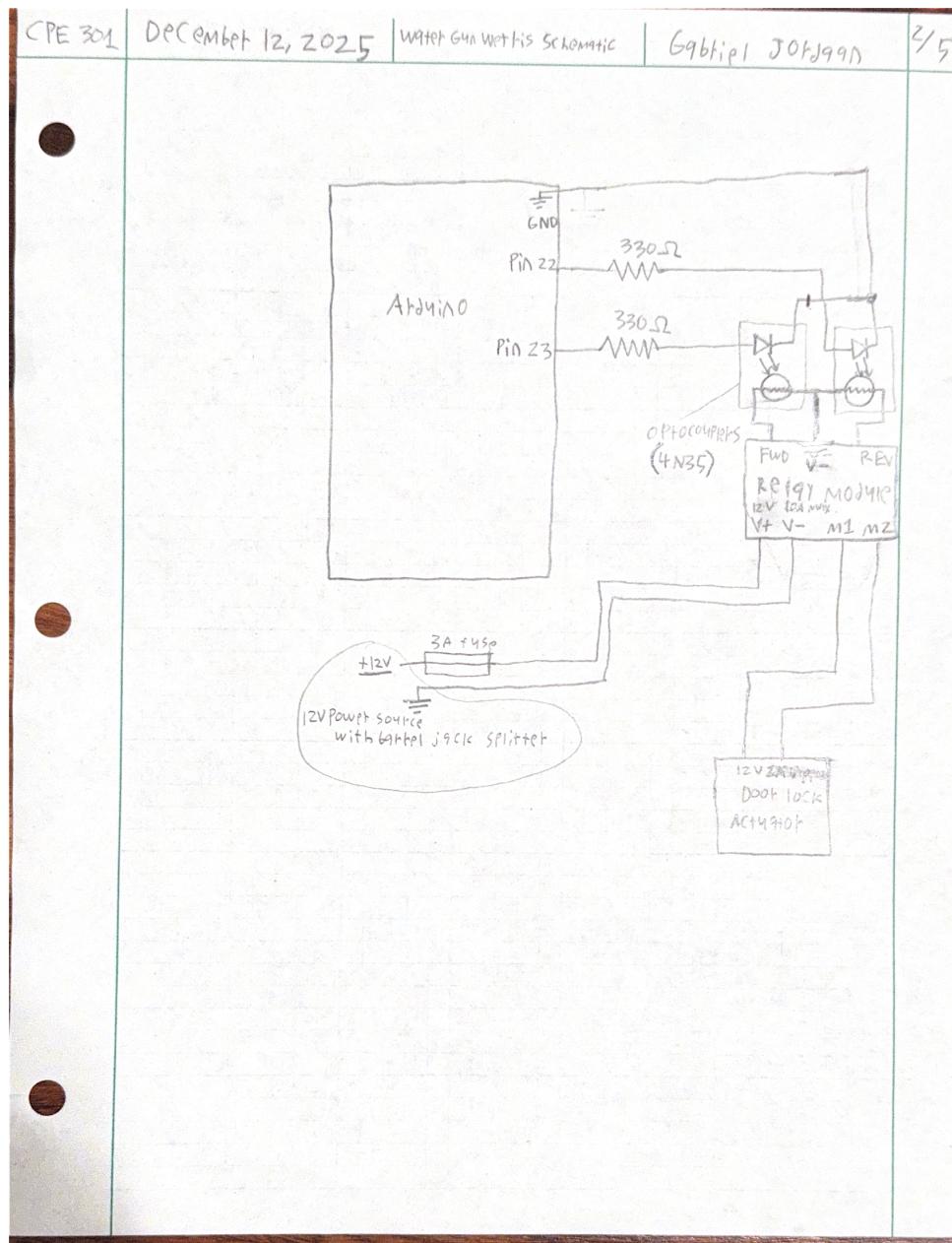


Figure 13: Electrical schematic of water gun module

6.3 Music Module Schematic

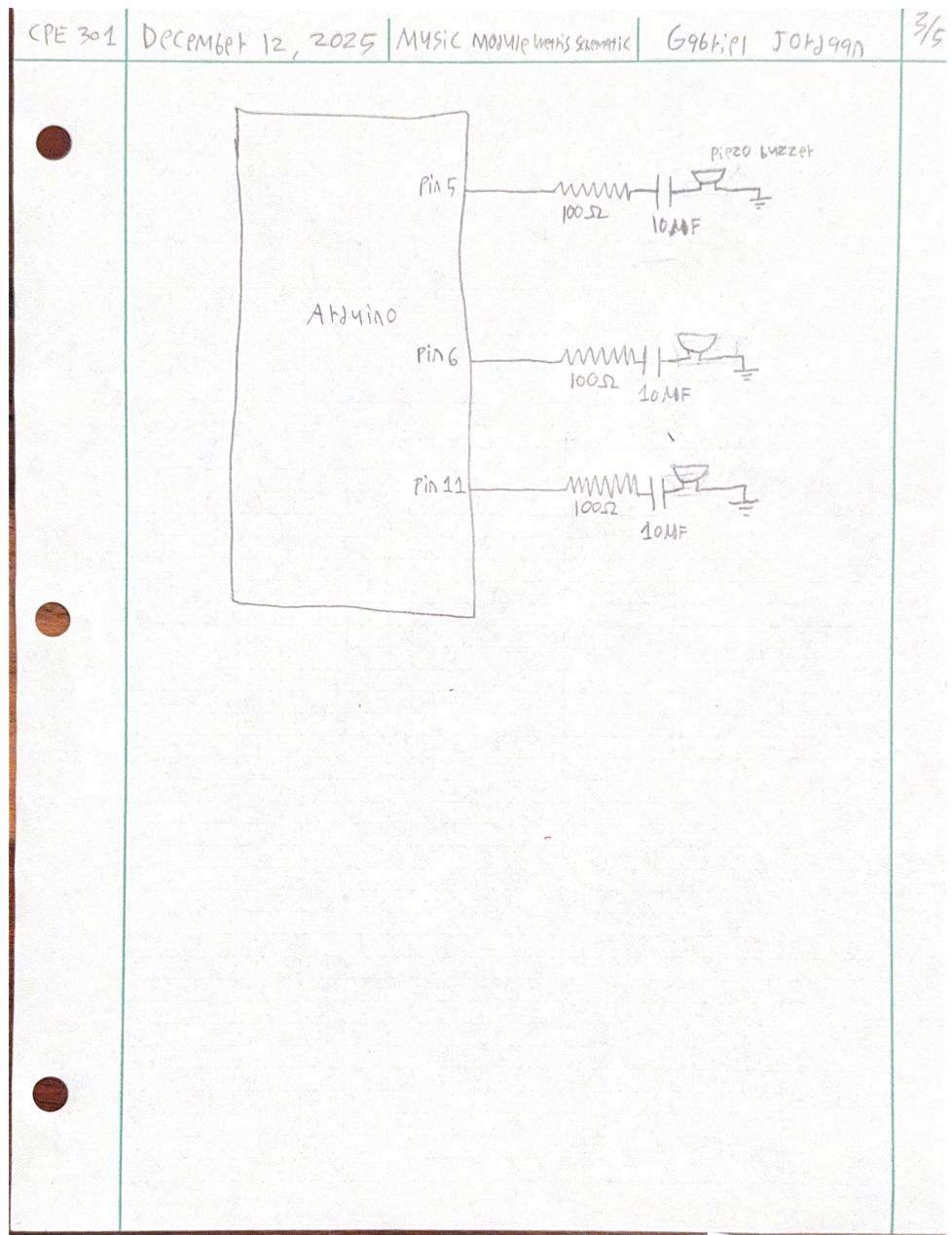


Figure 14: Electrical schematic of music module

6.4 LCD and Button Module Schematic

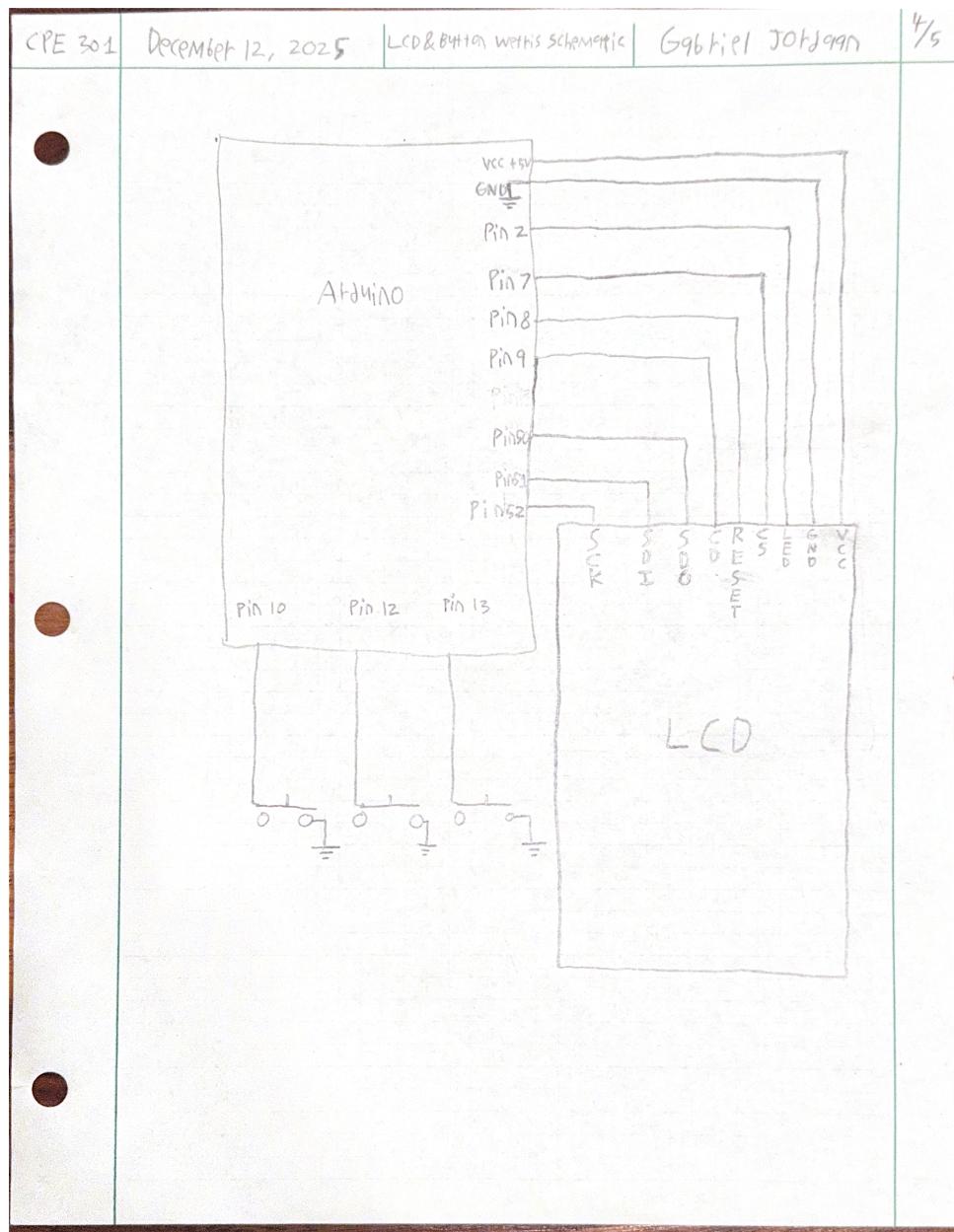


Figure 15: Electrical schematic of LCD and button module

6.5 Potentiometer Module Schematic

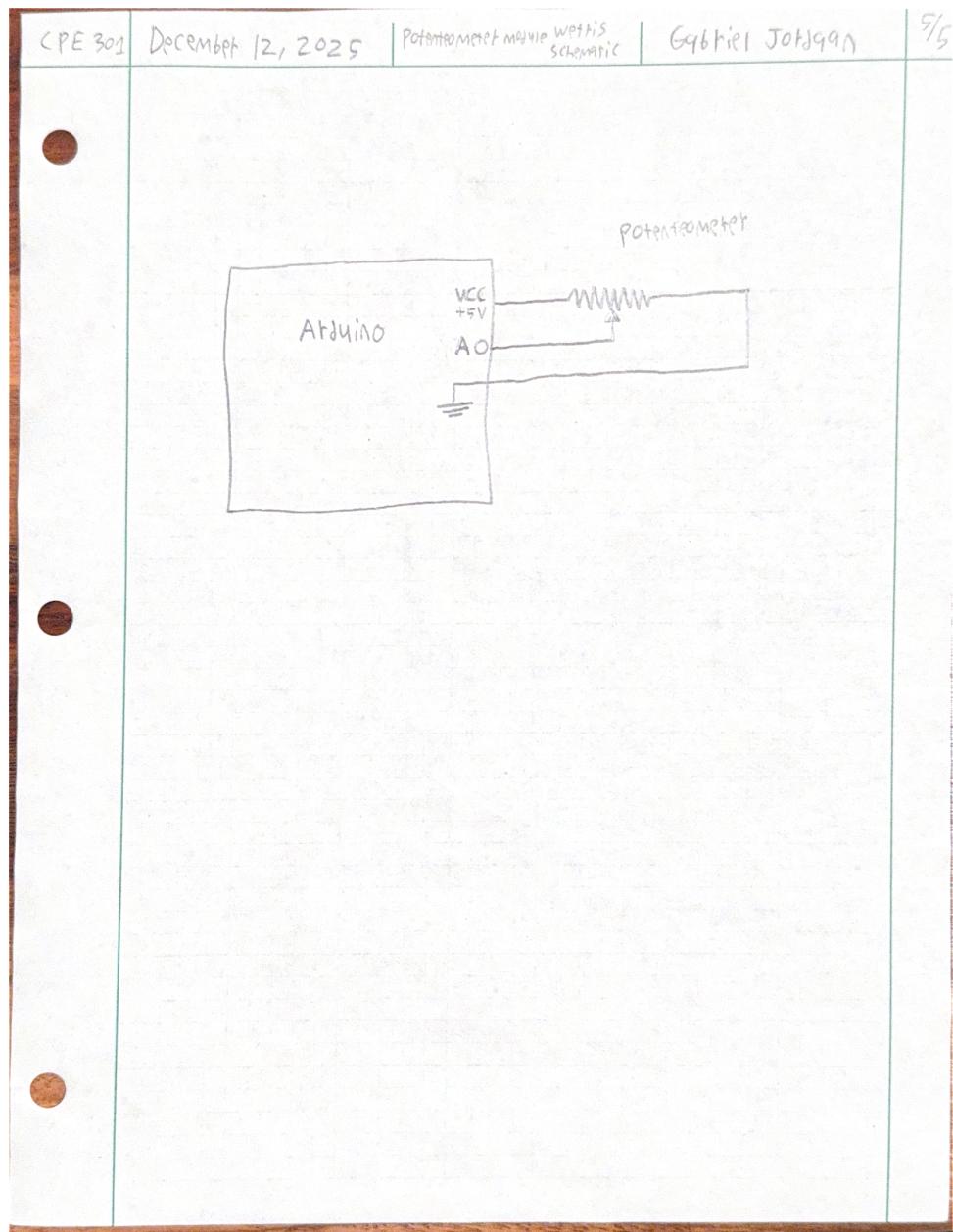


Figure 16: Electrical schematic of potentiometer module

6.6 3D Casing Technical Drawings

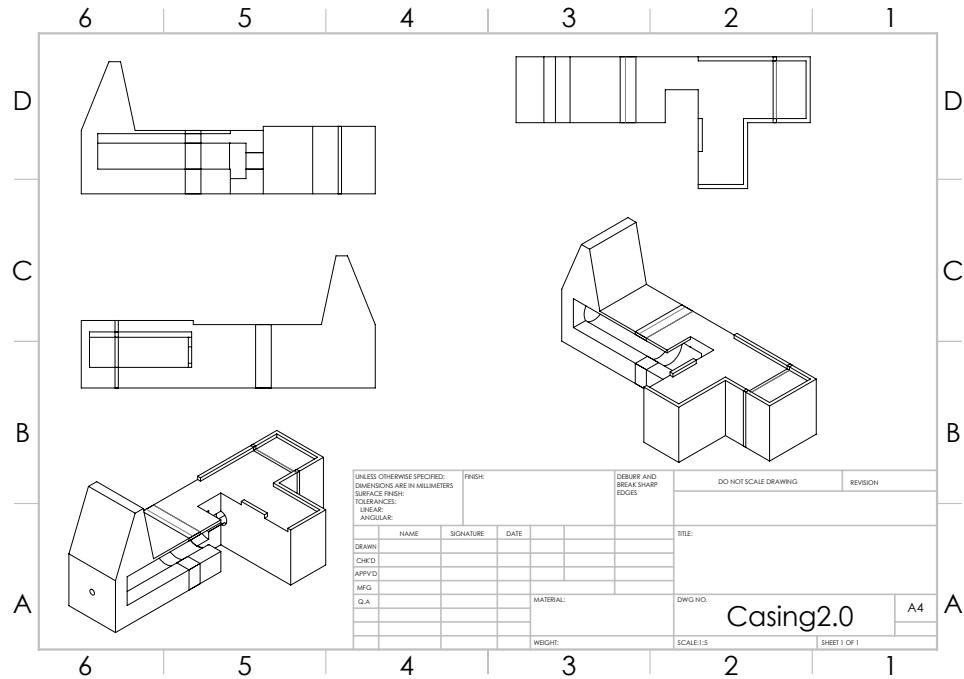


Figure 17: Solidworks technical drawing of 3D printed casing

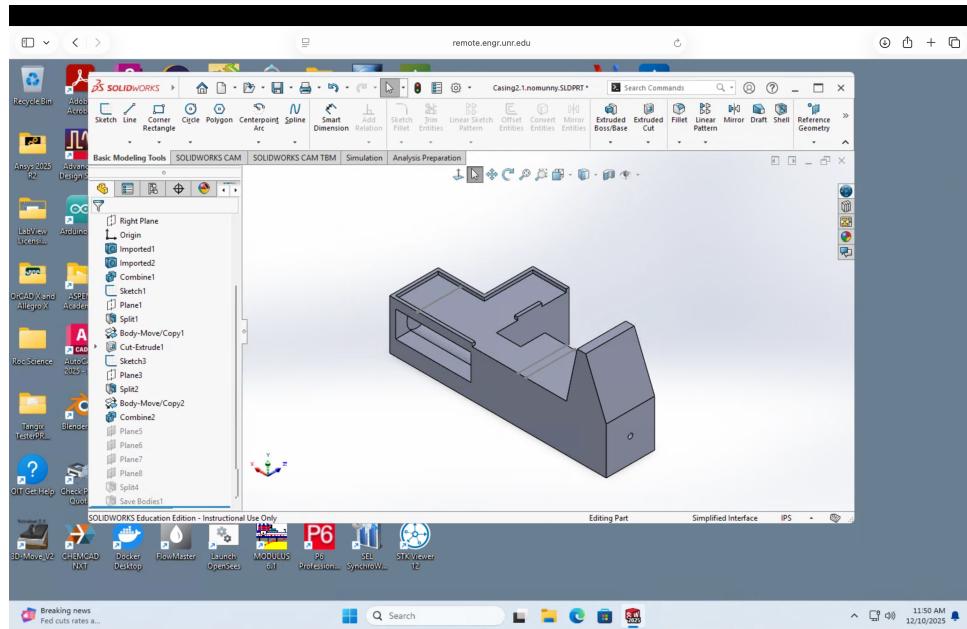


Figure 18: Solidworks model of 3D printed casing

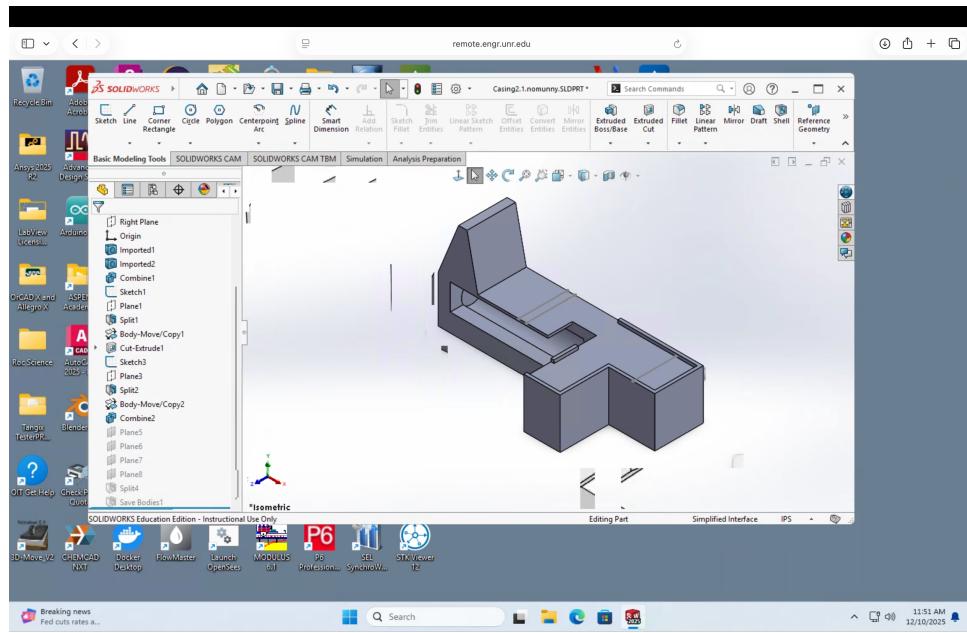


Figure 19: Screenshot of Solidworks model of 3D printed casing from a roughly isometric angle

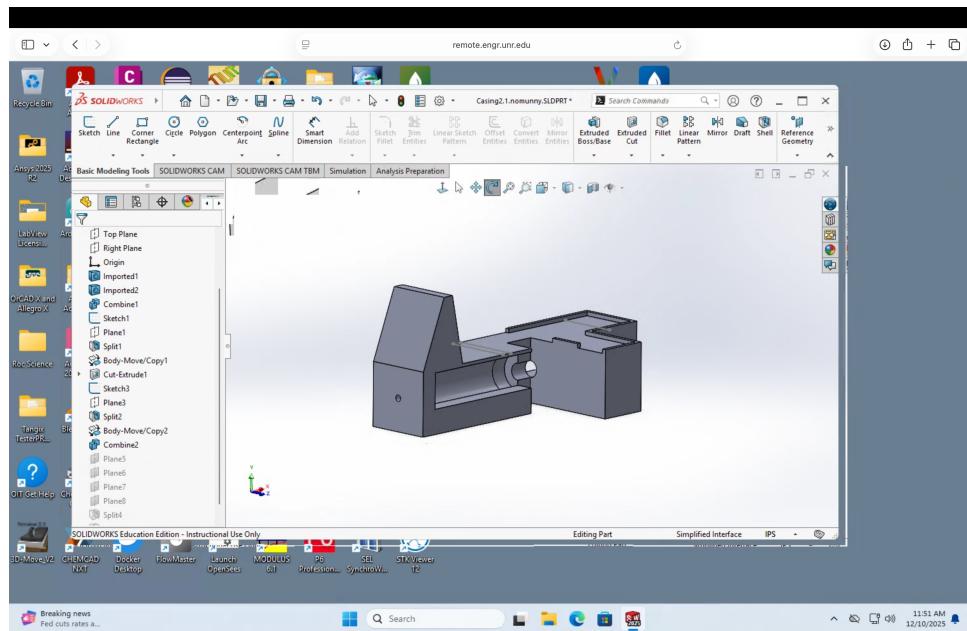


Figure 20: Second screenshot from inside Solidworks from a different angle

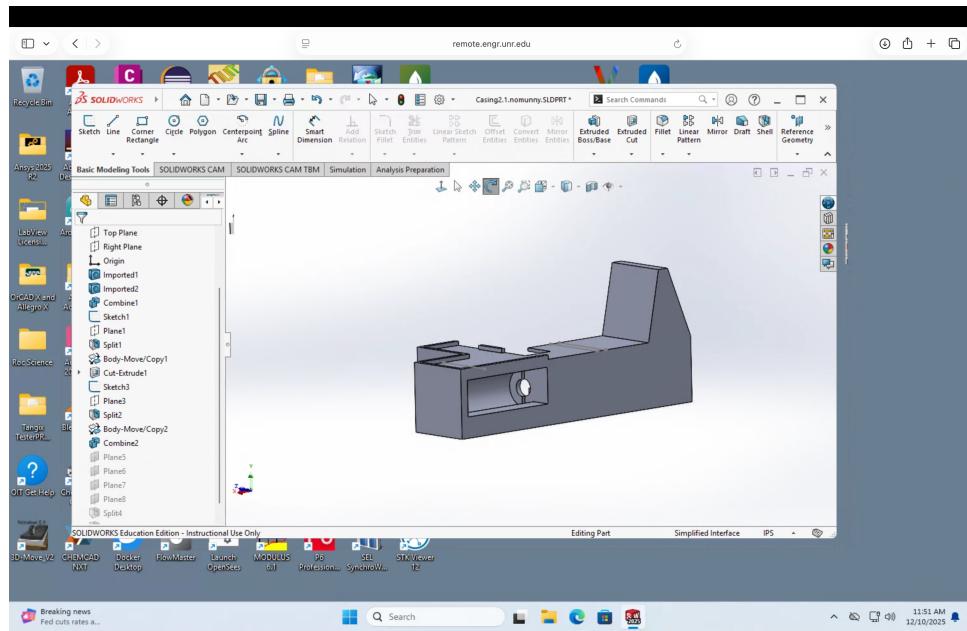


Figure 21: Fourth screenshot from inside Solidworks from a different angle

7 Links

7.1 Repository

GitHub Repository:

<https://github.com/ACertainArchangel/Wetris-CPE-301-Final-Project>

7.2 Documentation

Additional documentation and resources can be found in the project repository.

7.3 Demo Videos

Link to demo videos:

<https://drive.google.com/drive/folders/1Q4Bg4BSVfFWwpvJh6P310yMYkD1fBsM4?usp=sharing>

7.4 3D Printing Files

Link to stereolithography (STL) 3D printing files:

<https://drive.google.com/drive/folders/11z260TU0taoRf2S6610Rd5JktC3WCf8y?usp=sharing>

8 Environmental Impact as Detailed in CPE 301 Grading Rubric

8.1 Energy Efficiency

Wetris is designed to be energy efficient by utilizing low-power components to minimize power consumption during operation. The relay module and door lock actuator are only powered when necessary, reducing overall energy usage.

8.2 Design Safety

Wetris incorporates several safety features to ensure user safety during operation. The use of optocouplers isolates high voltage components from the low voltage control circuitry, preventing electrical hazards. Additionally, an inline fuse protects against overcurrent conditions.

8.3 Affordability

Considerations were given to potential cost reductions for mass production, such as bulk purchasing of components and exploring alternative manufacturing methods for the casing such as injection molding to make Wetris more affordable for consumers (see **cost analysis section**).

8.4 Sustainability

Wetris is designed with sustainability in mind by selecting components that are durable and have a long lifespan, reducing the need for frequent replacements. The 3D printed casing is made from PLA, which is derived from plant matter, to minimize environmental impact.

8.5 Accessibility

Wetris is designed to be accessible to a wide range of users by providing simple and intuitive controls. The use of large buttons and a clear LCD display ensures that users of varying abilities can easily interact with the system.

9 Our Team

9.1 Gabriel Jordaan



Gabriel Jordaan

Major: Electrical Engineering

Email: gvanrijnjordaan@unr.edu

GitHub: <https://github.com/ACertainArchangel>

Role & Contributions:

Project manager, hardware developer, enclosure design with Solidworks and 3D printing, music module, main.cpp, python utilities, documentation, and final report. Added UART water gun override prank. Paid for parts and tools and is going to keep the Wetris unit after the project is over.

About Me:

I am an electrical engineering student in the dual enrollment program at UNR, and as of fall 2025 I am 17 years old. I plan to graduate with my bachelor's degree in 2027 and move on to a master's and PhD in robotics. Embedded systems and machine learning are my main interests in terms of engineering. I enjoy hiking, cooking, studying the Bible, playing the cello, and working with my hands. I speak Afrikaans, English, and really awful Spanish. I have also started trying to learn Russian and am minoring in entrepreneurship.

9.2 Dieufainson Jean



Dieufainson Jean

Major: Electrical Engineering

Email: djean@unr.edu

GitHub: <https://github.com/Sonylove5>

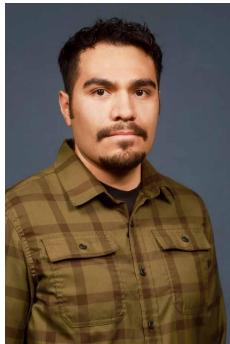
Role & Contributions:

Coding of the Potentiometer Module subsystem, a foundational UART communications layer, and the original Water Gun actuator control software.

About Me:

Currently a Bachelor's student at the University of Nevada, Reno, my focus on power distribution is driven by my background as a licensed electrician and a Haitian immigrant, aiming to reduce global electricity costs and expand access to this essential resource.

9.3 Jorge Robles



Jorge Robles

Major: Electrical Engineering

Email: jorgerobles@unr.edu

GitHub: <https://github.com/jrobles686>

Role & Contributions:

Used GPIO methods to write code that did not implement any functions such as digitalWrite, digitalRead etc. Also added the interrupts, ISR, for the hardware buttons instead of polling. New water gun actuator control software.

About Me:

I am a first generation student working towards my bachelor's degree in Electrical Engineering and am also minoring in Economic Policy. I enjoy learning how electronics communicate and operate in the real world and am looking forward to working in the automotive industry post-graduation. Economics is my guilty pleasure and I enjoy using a different part of my brain to decipher how economic policies and theories are used today.

I've worked in various fields throughout my career including the USPS, USFS, prestigious clubs and in the world of ophthalmology. I have plenty of exposure in customer satisfaction and am also fluent in Spanish. My real passion is electronics, and I am pursuing my dream to become an Electrical Engineer.

9.4 Roman Rosburg



Roman Rosburg

Major: Computer Science

Email: rosburgroman@gmail.com

GitHub: <https://github.com/RomanRosburg>

Role & Contributions:

Lead programmer on the Wetris game logic, handling most of the design and debugging. Also handled majority of graphics using LCD libraries. Victim of previously mentioned UART water gun prank.

About Me: Currently a computer science student at UNR with a specific focus on game development including design and engine development along with some study into cybersecurity. Long-term goal of working in a large studio when I graduate as well as developing a strong indie dev portfolio.

10 Acknowledgments

Thanks to our CPE 301 professor, Dr. Bashira Anima, for a great series of classes in fall 2025 and for being a good sport and letting us soak her with water!

We would also like to acknowledge the open-source libraries that made this project possible:

- LCDWIKI_SPI and LCDWIKI_GUI libraries - Created by the lcdwiki team (https://github.com/lcdwiki/LCDWIKI_gui), based on Adafruit GFX lib and Adafruit SPITFT lib, with init code from Rossum. Released under MIT license.
- Adafruit GFX and SPITFT libraries - For providing the foundational graphics and display functionality.

Thanks to Pastor David Minott for the idea of using a door lock actuator instead of a solenoid.

Thanks to Nikolai Nekrasov for writing the original Korobeiniki song in 1861. (He's dead and can't sue us for copyright infringement right?)

11 Wetris Logo



Figure 22: Wetris Logo