Team Pegasus Technical Report

Team Pegasus

DWW (Don't Waste Water)

Frank Rogina, Cody Newton, Merek Edwards, Cole Wallace, and Alex Duong

GEEN 1400

ABSTRACT

According to the EPA, the average person uses approximately 100 gallons of water per day. Team Pegasus' solution for this problem is a faucet mountable device that gamifies saving water. The game detects water flow from the faucet and assigns points depending on water usage. The less water one uses, the more points they receive. The main component of the game is a Hall Effect water sensor which detects the flow of water. The design will feature a point system displayed on the OLED screen. To include as many people as possible, individuals in a household will be able to select personal profiles for each individual using the product. Points received from playing the game will be assigned to that specific profile. After each use, a leaderboard will be shown on an OLED display to promote competitiveness and obtain who has been the most water conscious in the household. The brain of the game is an Arduino Nano which holds the game code and the memory for the leaderboard. All of these components will be housed in a small, sleek, clean case and powered by a rechargeable battery which can be replaced by opening the bottom of the product. The design's battery life exceeded the expected 13.5 hours by .65 hours for a total of 14.15 hours, which is within the error margin. This solution will not only encourage less water usage in the household but will also promote curiosity and education about the water overuse problem. Through this, DWW (Don't Waste Water) will teach families and future generations about water conservation and build sustainable habits.

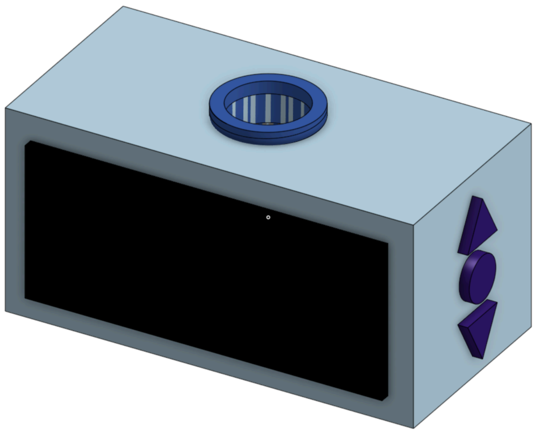


Figure 1: Initial Prototype CAD design

INTRODUCTION AND BACKGROUND

Team Pegasus' goal is to encourage being water conscious using the engineering design process by gamifying water usage in the household. The client is a common family household in the United States. According to the EPA, the average American family uses over 300 gallons of water daily. Roughly 40 gallons of the family's usage comes from a faucet. Americans use more water per person than citizens of any other country. As stated by the CDC, the average person in the United States uses more than twice as much water as the average person in France and four times as much water as the average person in India. To help create a more sustainable water future, team Pegasus is creating DWW (Don't Waste Water) to reduce this water usage.

DESIGN REQUIREMENTS

The design requirements for team Pegasus' "Don't Waste Water" project were imposed by the class and the team. The requirements set by the class are a budget of three hundred dollars, eight week completion time for expo, and the project needed to be environmentally focused. The self-imposed requirements by the team are very much more restrictive. The device should have a compact size, to not get in the way while using a sink and it should have a sleek modern design. The product should also be easy to use, with a simple intuitive design that can be easily understood and operated by someone of any age. The device must have a universal fitting, allowing it to be compatible with a wide range of faucets and water sources. Finally, it should be fun to use, making conservation efforts enjoyable and engaging for users. With these requirements team Pegasus hopes to encourage people of all ages to conserve water.

DESIGN ALTERNATIVES AND PROCESS

Team Pegasus started the design process with sketches of the desired design. The main design goals the team had in mind were safety, ease of use, reliability, and aesthetics. As a team, each of these design goals were discussed and prioritized in order of importance for the finished product. From these discussions, team Pegasus decided that safety should be the top priority because the project deals with water and electronics. As the two do not mix, it is important to keep in mind the safety of the electronics and the safety of the user. The next design goal the team decided to prioritize was reliability. It was clear that the desired final product would be purposeless if it was not reliable and did not work more than once. The team understood that for there to be desire and trust in the product, it had to be reliable. The next priority for the team was to make sure that the final product was easy to use. The team wanted to make sure that there was minimal interruption to typical faucet usage. This way, it would be easy to integrate the product into the user's daily life. The lowest ranking in terms of priority was aesthetics. While the team agreed that mounting a huge monochrome box to a household faucet would be a terrible idea, it was acknowledged that having a working monochrome product would be much better than having a nonfunction aesthetically pleasing product in the end. Team Pegasus had three initial designs, they were: the unit, the water drop, and Bluetooth. The unit was like its sounds, a monochrome box on the faucet with everything self-contained. The water drop was exactly like the unit but instead of a monochrome box, everything was encased in a water drop-type shape. The Bluetooth idea was two independent devices, one detecting the flow rate and the other doing all the calculations while displaying the needed information. The team decided that the unit was the best idea because it was the safest. In addition to this there were not any limitations on the shape like there were with the water drop and there were no worries about communication between the two devices like there were in the Bluetooth idea. The water drop idea was the most aesthetically pleasing but as previously stated the team wanted to focus its efforts on safety and reliability before going for aesthetics. The Bluetooth idea was the proposal that saved the most space on the faucet but none of the team was familiar with Bluetooth communications and were unsure if it could be accomplished within the timeframe so it was avoided. To get more information and get a different set of eyes on the unit design, the team consulted with ITLL engineer Natasha Ouellette. The consultation was brief but informative. The main concern that Natasha addressed to team Pegasus was to make sure that they had an unquestionable method of waterproofing. Natasha gave the team a couple of ideas to work on in terms of waterproofing such as; silicone caulking, spray waterproofing, and saran wrap. Once the team gathered enough information, the team finalized the design for the unit. Team Pegasus' prototype was similar to the initial design but it was not self encompassing, the flow rate sensor was in a separate box for waterproofing, That design did not have enough room for wiring so there was a cluster of wires in the encasement, however, the prototype was completely functional. From that prototype the team added more space in the next design to house all components and wiring. The team also made sure that the holes that were going to be made in the box were going to be easily waterproofed because some of the designs would have been hard to waterproof if proceeded with.

FINAL DESIGN

For the final design, team Pegasus decided to use a pre-built waterproof box for the shell of the project (4.53"x3.54"x1.9"). In this box a hole was drilled through the entire enclosure using a 1 ⅛" hole saw center on the top and approximately 1.5" from the back wall. Team Pegasus then cut out a hole on the front face of the box, using a jigsaw, centered .161" up from the inside of the box; the hole was 2.445"x1.161" cut only through the front face.

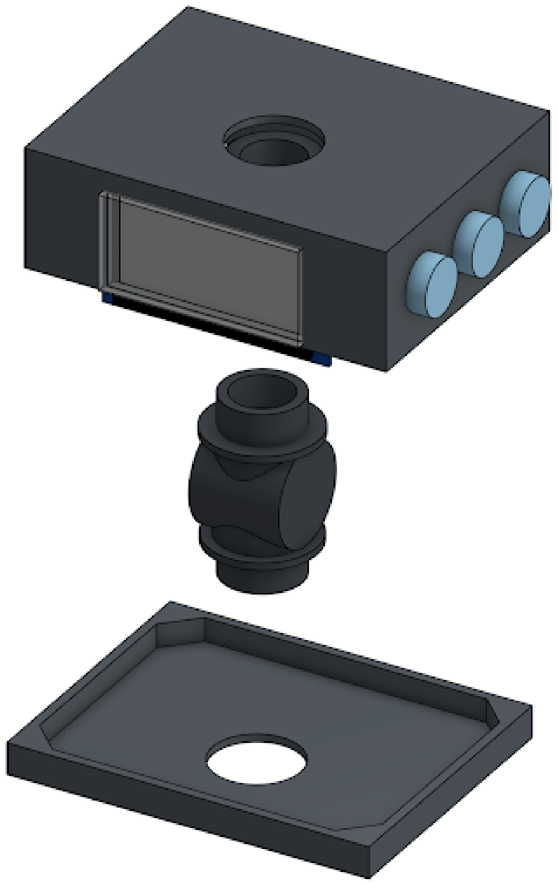


Figure 2: Final Design CAD Model

Three holes were then drilled on the right side of the box for buttons. Two holes were drilled above each other approximately 1.5" from the front, each hole was .5" in diameter; the first was .3" from the bottom with a .3" space between both holes. The third hole was drilled, .5" away from the previous holes referencing the sides again this hole was .5” in diameter. Using contact cement we mounted a Digiten 1” Hall effect flow rate sensor center on the top hole, using silicone caulking to seal the gap between the Digiten sensor and the hole. Team Pegasus then cut out a piece of clear acrylic, using the Epilog Legend 36 EXT laser cutter, measuring 1.2”x2.5”x.125”. This will be used to cover up the hole drilled for the screen. The acrylic piece was mounted onto the front face of the box centered on the 2.445”x1.161” hole using JBWeld. Three Twidec 12mm ½” mounting hole waterproof buttons were installed in the three ½” holes cut out on the right side of the box. The team then mounted an Arduino Nano on the inside of the case centered at approximately 0.5” to the left of the Digiten sensor. A Jex Electronics Quad AAA 6v battery holder was mounted to the lid of the box centered on the back wall. Team Pegasus had a custom PCB built to optimize space usage (Figure 3). The PCB was 0.787” x0.850”. This PCB was not mounted to a wall instead it was held in place below the Arduino Nano by the numerous wires connecting to it (Figure 4).

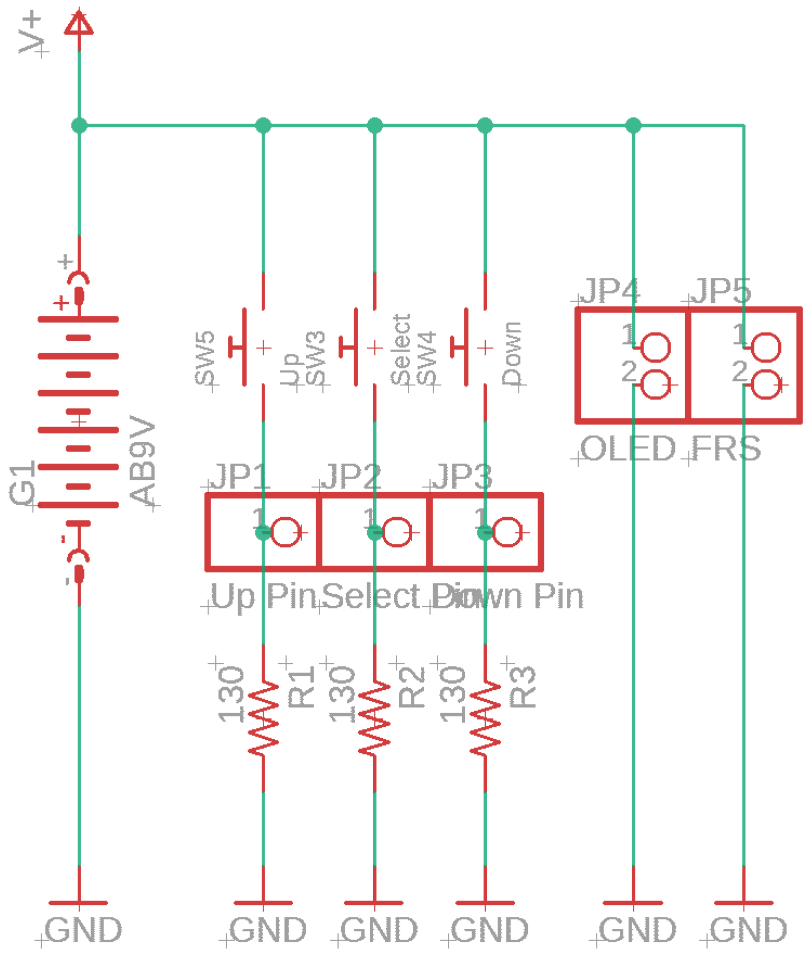


Figure 3: Electrical Diagram

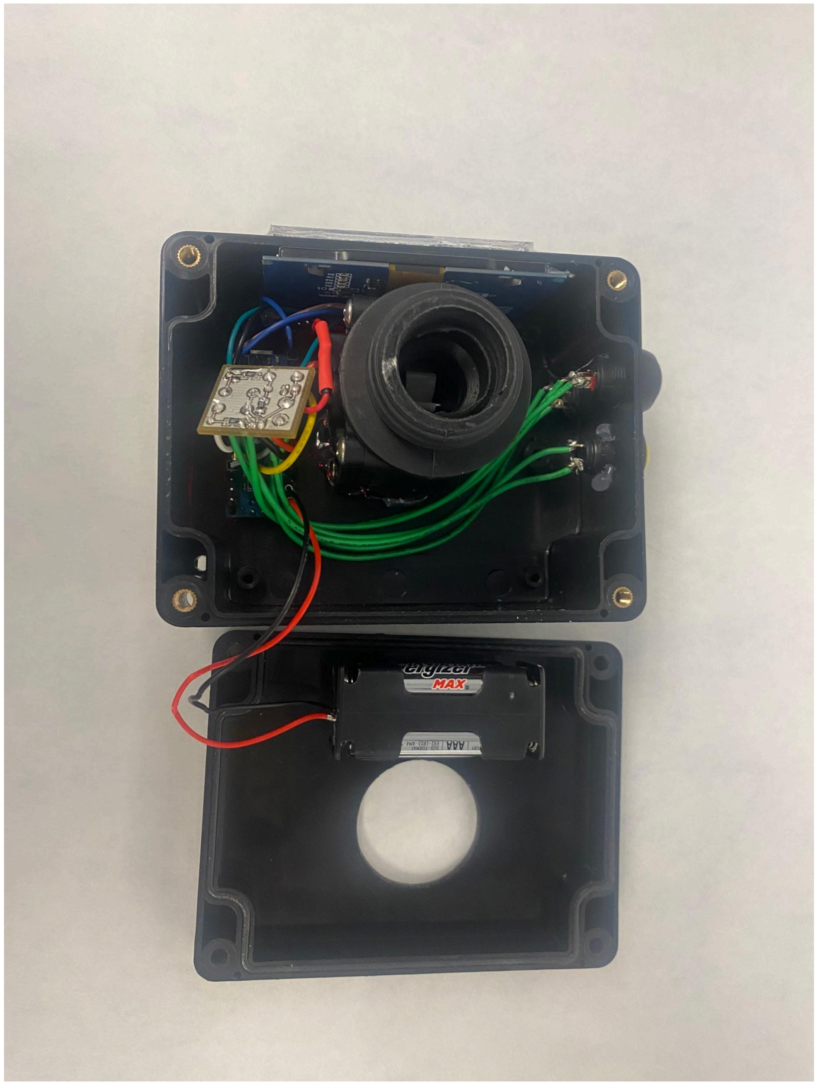


Figure 4: DWW internals

A 2.62” Oled screen was mounted in front of the Digiten sensor center on the hole cut on the front face of the box. After everything was mounted, team Pegasus soldered the buttons, power from the Arduino Nano, power to the OLED, and power to the Digiten Sensor to the PCB. Team Pegasus then connected the signal wires from the PCB, OLED, and Digiten sensor to the Arduino Nano and the device was operational. The code of the project was an ordeal as none of the team members have previous coding experience. The code detects the flow rate, converts it to a score, stores the score under a specific user, displays the users and their scores on a scoreboard, and allows users to customize their usernames on the device, full code is included in Appendix A.

TESTING & ANALYSIS

The first critical component of the project was the power supply, without the power supply the product would not be able to display or detect what is intended. Initially, the design consisted of a 9V battery as the power supply. However, after doing the nominal calculations the 9V was changed to four AAA batteries. To test this setup, team Pegasus utilized a multimeter set at 200mA in series between the power supply power wire and the Arduino Vin pin. The OLED screen draws a different amount of current depending on how many pixels are in use. To find an average of the current draw, the current was measured on each screen setting and then multiplied by the estimated amount of time in a percentage the device would spend displaying that screen (Table 1). After the measurements and calculations were done the average was determined to be 83.5 mA.

	Score Board	UserScreen	Name Selection	Total
Amperage (mA)	90	60	50	
Time Spent on Screen(%)	80	15	5	
Current Draw (mA)	72	9	2.5	83.5

Table 1: Current draw of the circuit depending on the screen display

The product initially started with a 9V battery. After the initial calculations were done, it was determined that the project would only be able to run for approximately 10 hours. It was determined that the team could fit a bigger battery pack in the housing, so the team decided to switch from a 9V battery to four triple-A batteries, which should nominally run the project for 13.5 hours. The nominal calculated power draw of the system was determined to be 0.5 W. When accounting for a variable draw from the Arduino it averaged the actual figure to .4 W. Actual testing determined that the device lasted 14.15 hours which was over 4 hours more than a single 9v (Table 2).

	Nominal	Actual
Average Power Draw (W)	0.5	0.4
(4x) AAA Battery Life (Wh)	7.48	5.98
9v Battery Life (Wh)	5.49	4.4
(4x) AAA Run Time (h)	13.5	14.15

Table 2: Battery run time calculations

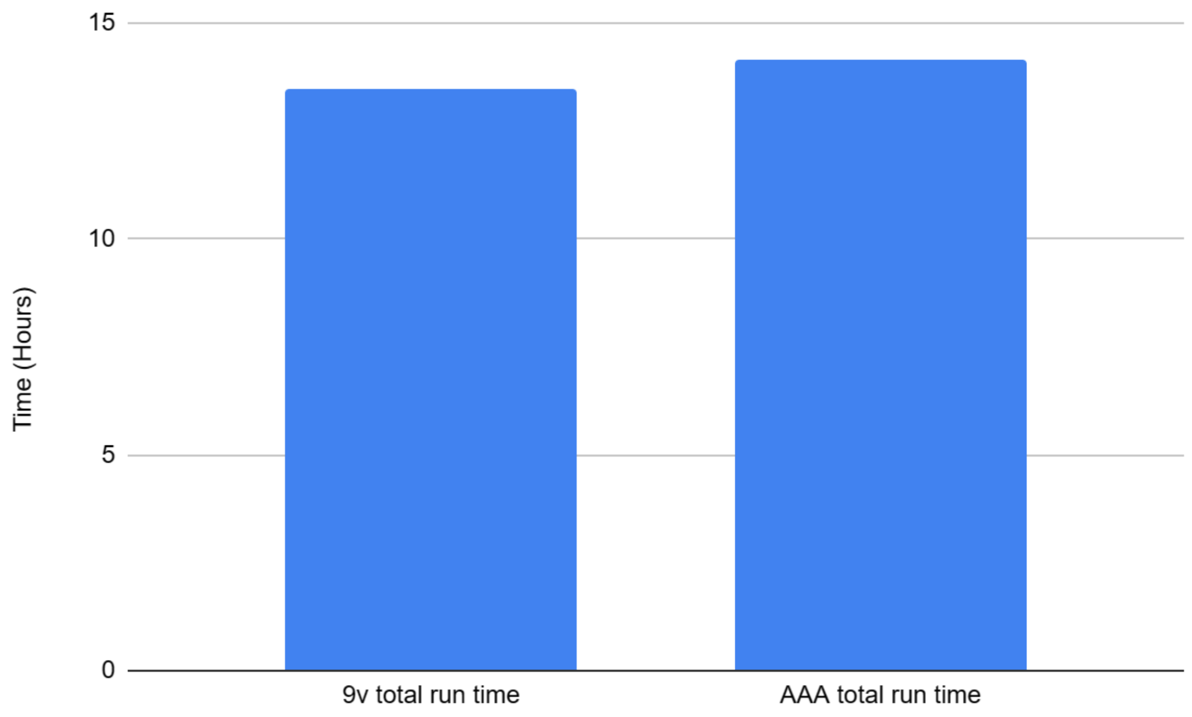


Figure 5: Battery Run time graph

For user testing, team Pegasus decided to use college students because they were readily available and were a mean value for the age range of users of the device. Team Pegasus tested four criteria in user testing: ease of use, reliability, likelihood to use again, and provides enough water (Figure 5). Team Pegasus tested 7 users and had them rank each of the criteria on a scale from 1- 10. Based on the first initial user testing team Pegasus determined that there were changes needed to enhance the use of the device, some instructions were provided on the screen to guide the users through the experience. After those changes were made the data started trending upward.

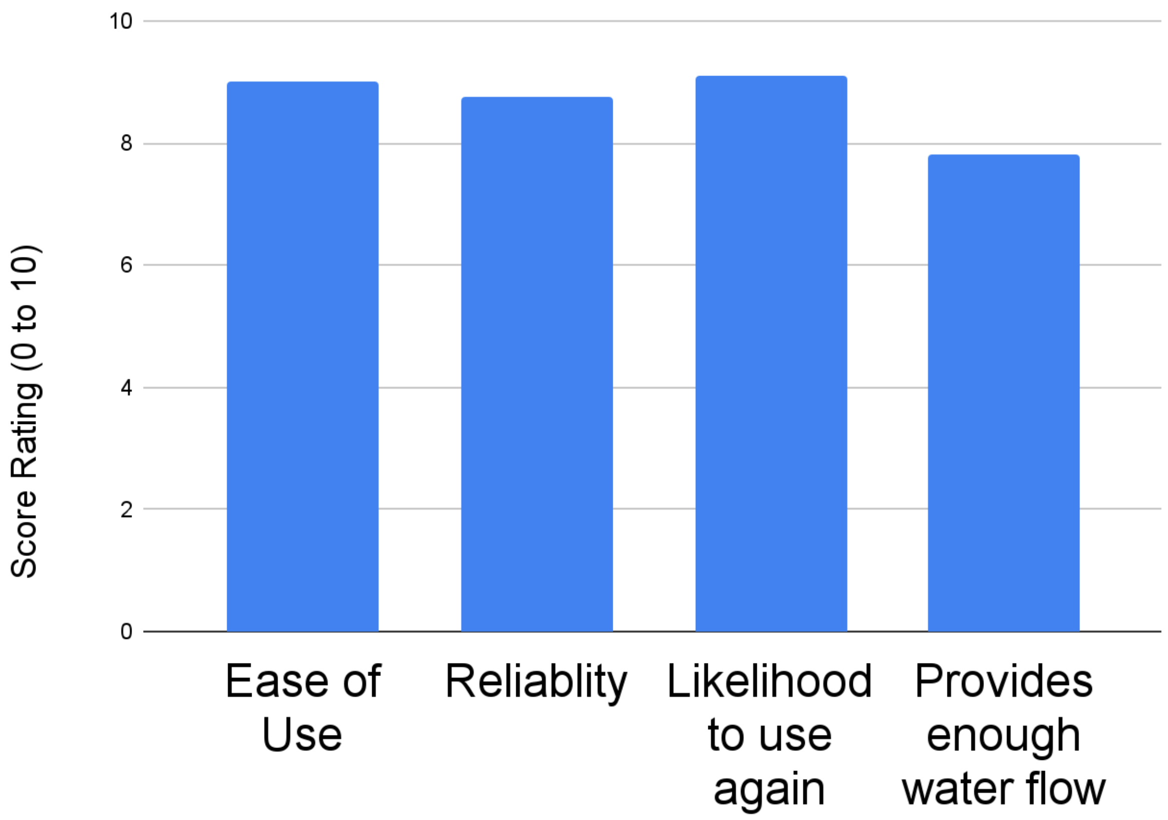


Figure 6: User Testing

The user testing pointed out ways to improve the device, especially in terms of providing enough water flow. Due to the restrictions of the Digiten flow rate sensor and the waterproofing of the device, the water could not flow at the full power on the sink which most people tend to use, this part of the design was not intentional but it proved to be beneficial in a way because it forces users to use less water pressure thus less water for basic activities such as brushing one's teeth and washing one's hands. Team Pegasus was satisfied with the upward trend in the results, especially the ease of use and likelihood to use again sections which shows that the target user will be able to, and most likely, happily use the device in the future to conserve water.

CONCLUSION

The main goal of this project was to encourage water consciousness in American households. By using our product Americans can become more aware of their daily water usage. With this knowledge and the competitive aspect of the game, it encourages Americans to use less water in their daily tasks. There are many areas of sustainability that America is falling behind on. As of 2019 according to the SDSN (Sustainable Development Solutions Network), the USA ranks 35th out of 162 countries. Team Pegasus's project aims to push America in a direction that will benefit not only themselves but the world. Sustainability doesn't have to be a huge change. If every American made small changes that would be better than a small population doing major changes. By using this product, water will be at the forefront of household residents' minds and begin the process of making little changes that can make a big impact. The sustainability aspect that team Pegasus utilized was in the power source. Using rechargeable batteries lessens the impact of electrical waste. The lifetime of rechargeable AAA batteries is over five years. Americans waste the most amount of water per person than any other country. By utilizing this product Americans will become more water conscious leading to less water waste in the household which is something that this country needs to reduce. For future iterations of the project team Pegasus would consider; decreasing size, increasing the efficiency of code, including a power-saving function, and using a more powerful processor.

TIMELINE

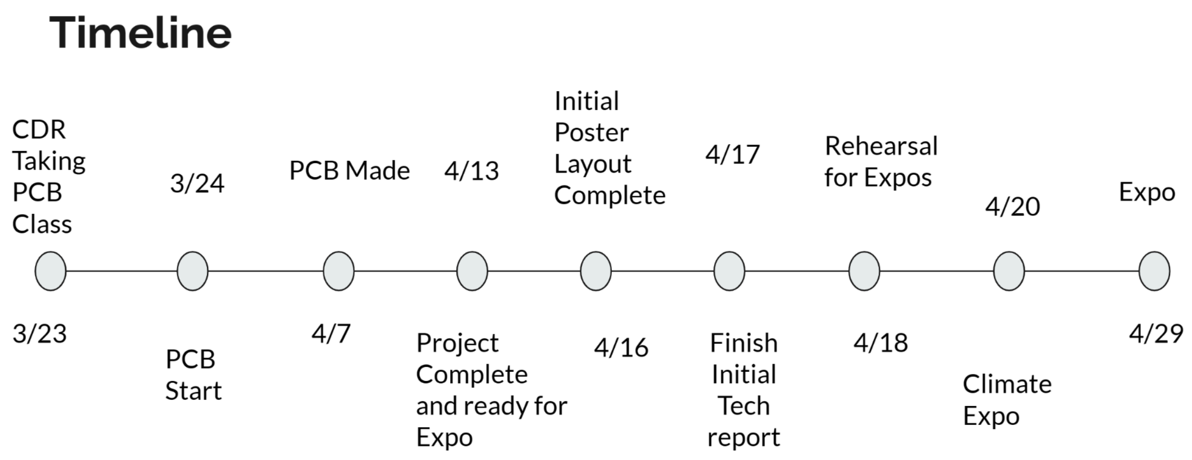


Figure 7: Proposed Timeline

Above is team Pegasus' final proposed timeline. While there were some unforeseen changes made to this timeline, team Pegasus generally followed it. The slight changes that were made were, team Pegasus was unable to have the initial tech report done as early as initially planned instead it was completed after the Expo. Team Pegasus also did not get to rehearse as much as planned outside of class before Expo. Overall, team Pegasus followed the proposed timeline as much as possible and because of that the amount of work that was rushed to be completed the week before Expo was less than five percent of the project. All that was needed to be done was to ensure the wiring connections were good and to finalize the poster.

BILL OF MATERIALS

Power Source	
9V(Donated)	$8.87
AAA(8)	$9.95
Battery Case	$1.79
Rechargeable AAA	$13.87
Charging Case	$17.87

Adapter/Casing	
Waterproof Box	$8.99
Threaded PVC (4)	$8.00
Faucet Adapters	$17.03

Electronics	
Arduino Nano	$7.47
OLED(2)	$33.98
Flow Rate Sensor(2)	$21.99
Buttons(8)	$8.99
PCB	$5.00

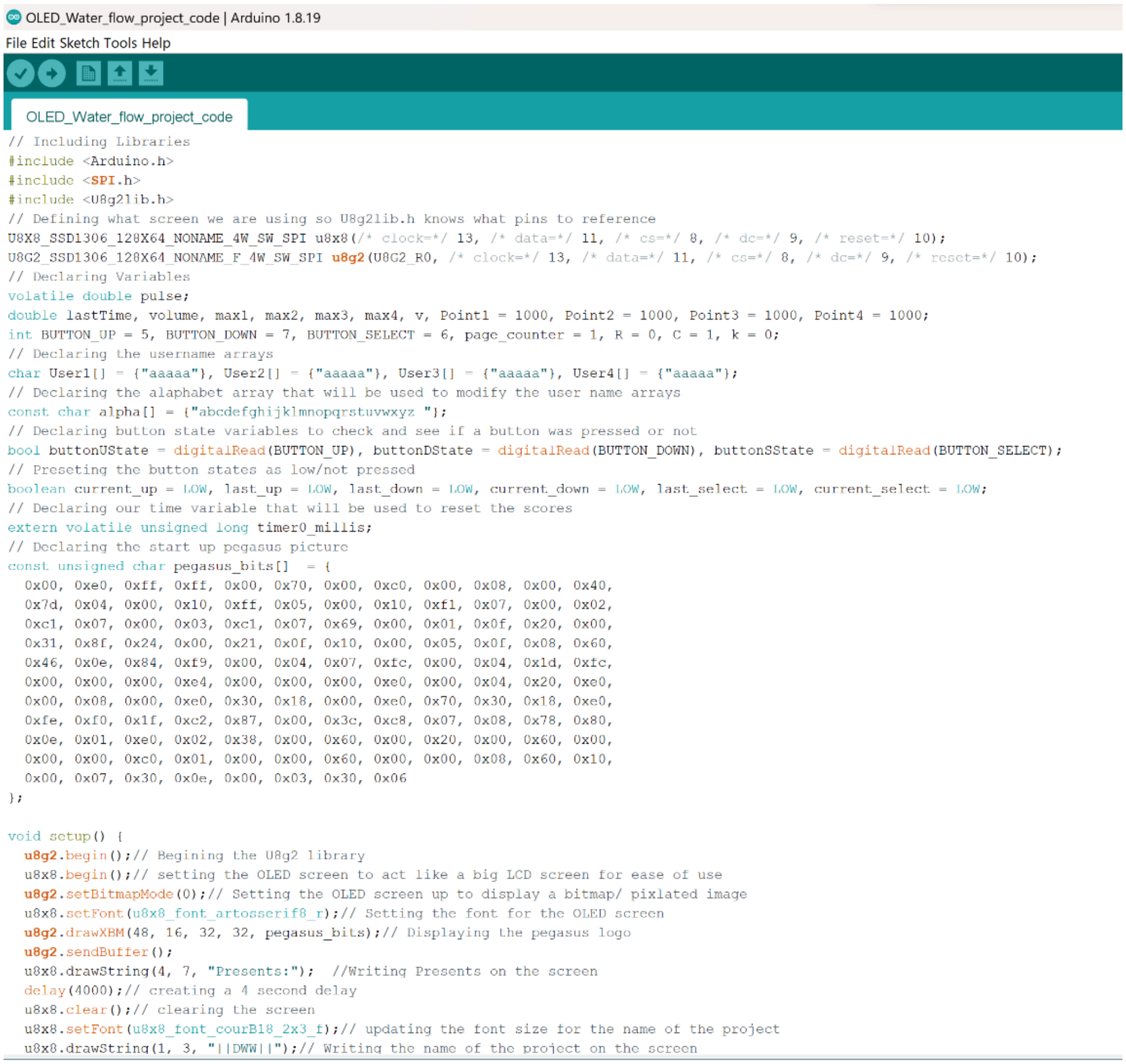
Adhesives	
Ultra Clear Waterproof Sealant	$8.98
JB Weld	$6.54

Total	Per Unit
$179.32	$83.05

Figure 8: Bill of Materials

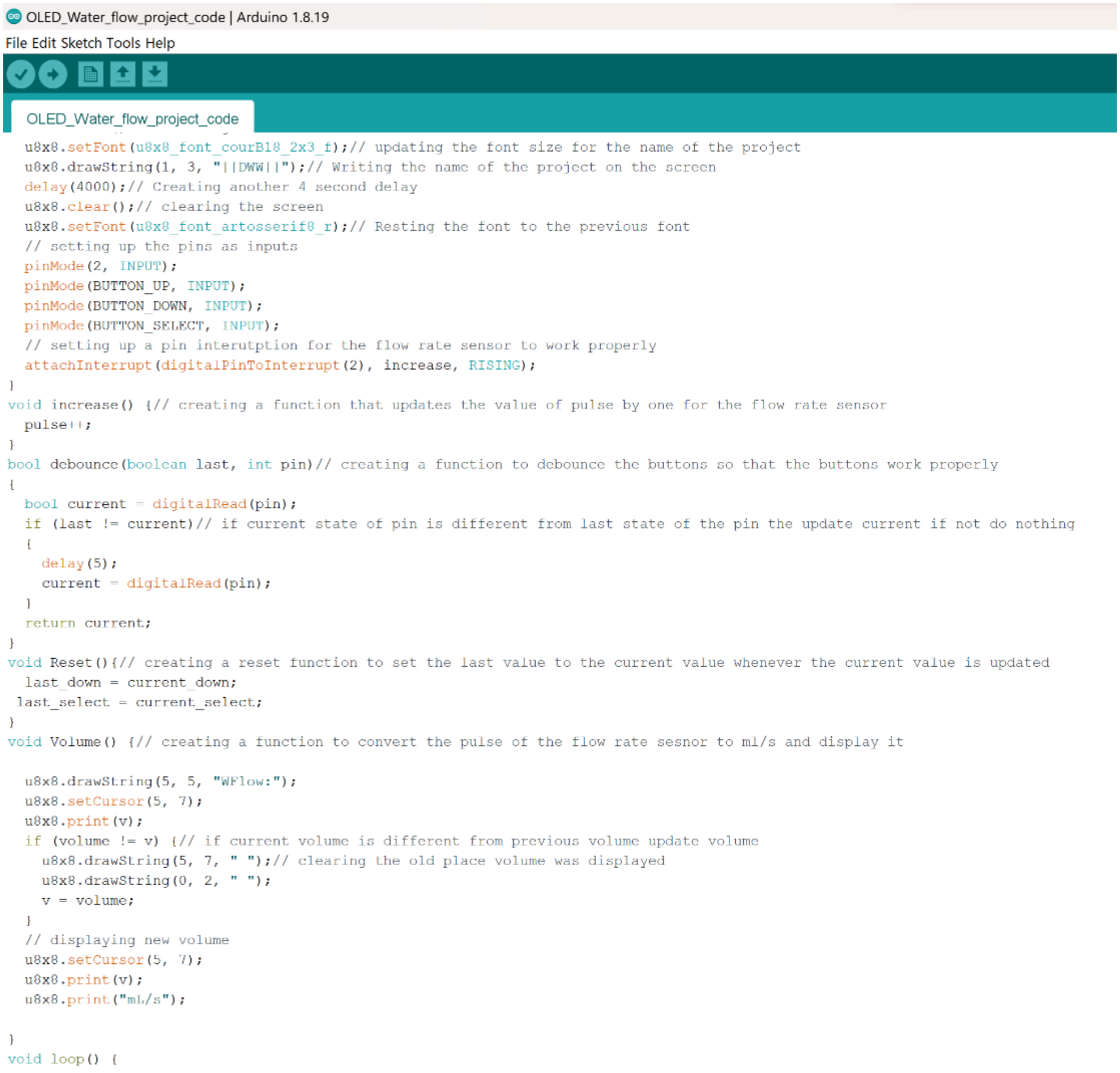
APPENDIX A: ARDUINO CODE USED TO PERFORM FLOW RATE DETECTION, UI, AND SCORE CONVERSION

The goal of the Arduino code was to detect the flow rate of the water, store it under a specific user, and display the username and the user's score against the other users. The team also wanted to have the ability to change the users names.


OLED_Water_flow_project_code | Arduino 1.8.19
File Edit Sketch Tools Help
OLED_Water_flow_project_code

```
// Including Libraries
#include <Arduino.h>
#include <SPI.h>
#include <U8g2lib.h>
// Defining what screen we are using so U8g2lib.h knows what pins to reference
U8X8_SSD1306_128X64_NONAME_4W_SW_SPI u8x8(/* clock=*/ 13, /* data=*/ 11, /* cs=*/ 8, /* dc=*/ 9, /* reset=*/ 10);
U8G2_SSD1306_128X64_NONAME_F_4W_SW_SPI u8g2(U8G2_R0, /* clock=*/ 13, /* data=*/ 11, /* cs=*/ 8, /* dc=*/ 9, /* reset=*/ 10);
// Declaring Variables
volatile double pulse;
double lastTime, volume, max1, max2, max3, max4, v, Point1 = 1000, Point2 = 1000, Point3 = 1000, Point4 = 1000;
int BUTTON_UP = 5, BUTTON_DOWN = 7, BUTTON_SELECT = 6, page_counter = 1, R = 0, C = 1, k = 0;
// Declaring the username arrays
char User1[] = {"aaaaa"}, User2[] = {"aaaaa"}, User3[] = {"aaaaa"}, User4[] = {"aaaaa"};
// Declaring the alaphabet array that will be used to modify the user name arrays
const char alpha[] = {"abcdefghijklmnopqrstuvwxyz "};
// Declaring button state variables to check and see if a button was pressed or not
bool buttonUState = digitalRead(BUTTON_UP), buttonDState = digitalRead(BUTTON_DOWN), buttonSState = digitalRead(BUTTON_SELECT);
// Preseting the button states as low/not pressed
boolean current_up = LOW, last_up = LOW, last_down = LOW, current_down = LOW, last_select = LOW, current_select = LOW;
// Declaring our time variable that will be used to reset the scores
extern volatile unsigned long timer0_millis;
// Declaring the start up pegasus picture
const unsigned char pegasus_bits[]  = {
  0x00, 0xe0, 0xff, 0xff, 0x00, 0x70, 0x00, 0xc0, 0x00, 0x08, 0x00, 0x40,
  0x7d, 0x04, 0x00, 0x10, 0xff, 0x05, 0x00, 0x10, 0xf1, 0x07, 0x00, 0x02,
  0xc1, 0x07, 0x00, 0x03, 0xc1, 0x07, 0x69, 0x00, 0x01, 0x0f, 0x20, 0x00,
  0x31, 0x8f, 0x24, 0x00, 0x21, 0x0f, 0x10, 0x00, 0x05, 0x0f, 0x08, 0x60,
  0x46, 0x0e, 0x84, 0xf9, 0x00, 0x04, 0x07, 0xfc, 0x00, 0x04, 0x1d, 0xfc,
  0x00, 0x00, 0x00, 0xe4, 0x00, 0x00, 0x00, 0xe0, 0x00, 0x04, 0x20, 0xe0,
  0x00, 0x08, 0x00, 0xe0, 0x30, 0x18, 0x00, 0xe0, 0x70, 0x30, 0x18, 0xe0,
  0xfe, 0xf0, 0x1f, 0xc2, 0x87, 0x00, 0x3c, 0xc8, 0x07, 0x08, 0x78, 0x80,
  0x0e, 0x01, 0xe0, 0x02, 0x38, 0x00, 0x60, 0x00, 0x20, 0x00, 0x60, 0x00,
  0x00, 0x00, 0xc0, 0x01, 0x00, 0x00, 0x60, 0x00, 0x00, 0x08, 0x60, 0x10,
  0x00, 0x07, 0x30, 0x0e, 0x00, 0x03, 0x30, 0x06
};

void setup() {
  u8g2.begin();// Begining the U8g2 library
  u8x8.begin();// setting the OLED screen to act like a big LCD screen for ease of use
  u8g2.setBitmapMode(0);// Setting the OLED screen up to display a bitmap/ pixlated image
  u8x8.setFont(u8x8_font_artosserif8_r);// Setting the font for the OLED screen
  u8g2.drawXBM(48, 16, 32, 32, pegasus_bits);// Displaying the pegasus logo
  u8g2.sendBuffer();
  u8x8.drawString(4, 7, "Presents:");  //Writing Presents on the screen
  delay(4000);// creating a 4 second delay
  u8x8.clear();// clearing the screen
  u8x8.setFont(u8x8_font_courB18_2x3_f);// updating the font size for the name of the project
  u8x8.drawString(1, 3, "||DWW||");// Writing the name of the project on the screen
```

Figure 9: Arduino Code Part 1

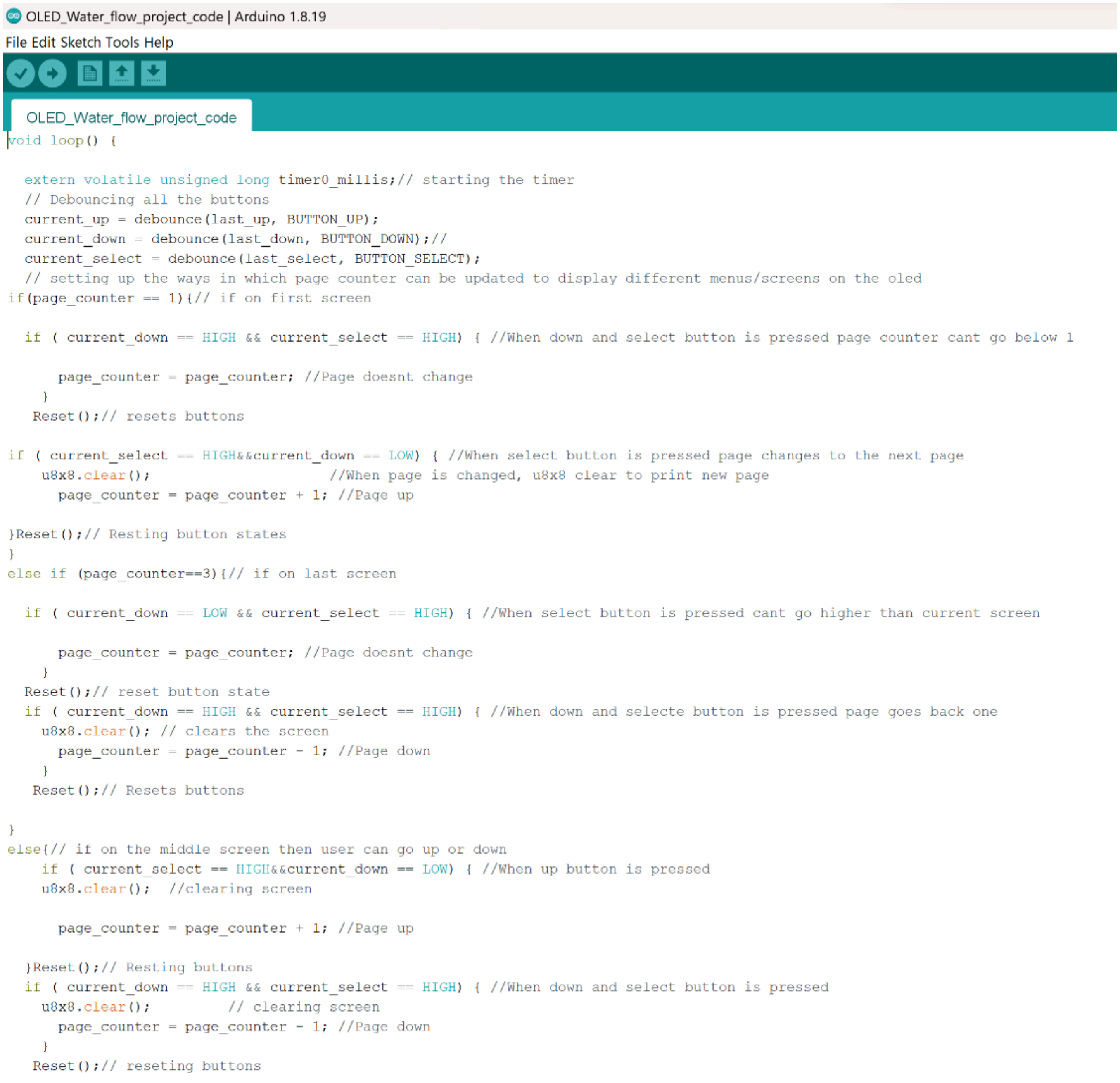

OLED_Water_flow_project_code | Arduino 1.8.19

File Edit Sketch Tools Help

OLED_Water_flow_project_code

```
  u8x8.setFont(u8x8_font_courB18_2x3_f);// updating the font size for the name of the project
  u8x8.drawString(1, 3, "||DWW||");// Writing the name of the project on the screen
  delay(4000);// Creating another 4 second delay
  u8x8.clear();// clearing the screen
  u8x8.setFont(u8x8_font_artosserif8_r);// Resting the font to the previous font
  // setting up the pins as inputs
  pinMode(2, INPUT);
  pinMode(BUTTON_UP, INPUT);
  pinMode(BUTTON_DOWN, INPUT);
  pinMode(BUTTON_SELECT, INPUT);
  // setting up a pin interutption for the flow rate sensor to work properly
  attachInterrupt(digitalPinToInterrupt(2), increase, RISING);
}
void increase() {// creating a function that updates the value of pulse by one for the flow rate sensor
  pulse++;
}
bool debounce(boolean last, int pin)// creating a function to debounce the buttons so that the buttons work properly
{
  bool current = digitalRead(pin);
  if (last != current)// if current state of pin is different from last state of the pin the update current if not do nothing
  {
    delay(5);
    current = digitalRead(pin);
  }
  return current;
}
void Reset(){// creating a reset function to set the last value to the current value whenever the current value is updated
  last_down = current_down;
 last_select = current_select;
}
void Volume() {// creating a function to convert the pulse of the flow rate sesnor to ml/s and display it

  u8x8.drawString(5, 5, "WFlow:");
  u8x8.setCursor(5, 7);
  u8x8.print(v);
  if (volume != v) {// if current volume is different from previous volume update volume
    u8x8.drawString(5, 7, " ");// clearing the old place volume was displayed
    u8x8.drawString(0, 2, " ");
    v = volume;
  }
  // displaying new volume
  u8x8.setCursor(5, 7);
  u8x8.print(v);
  u8x8.print("mL/s");

}
void loop() {
```

Figure 10: Arduino Code Part 2

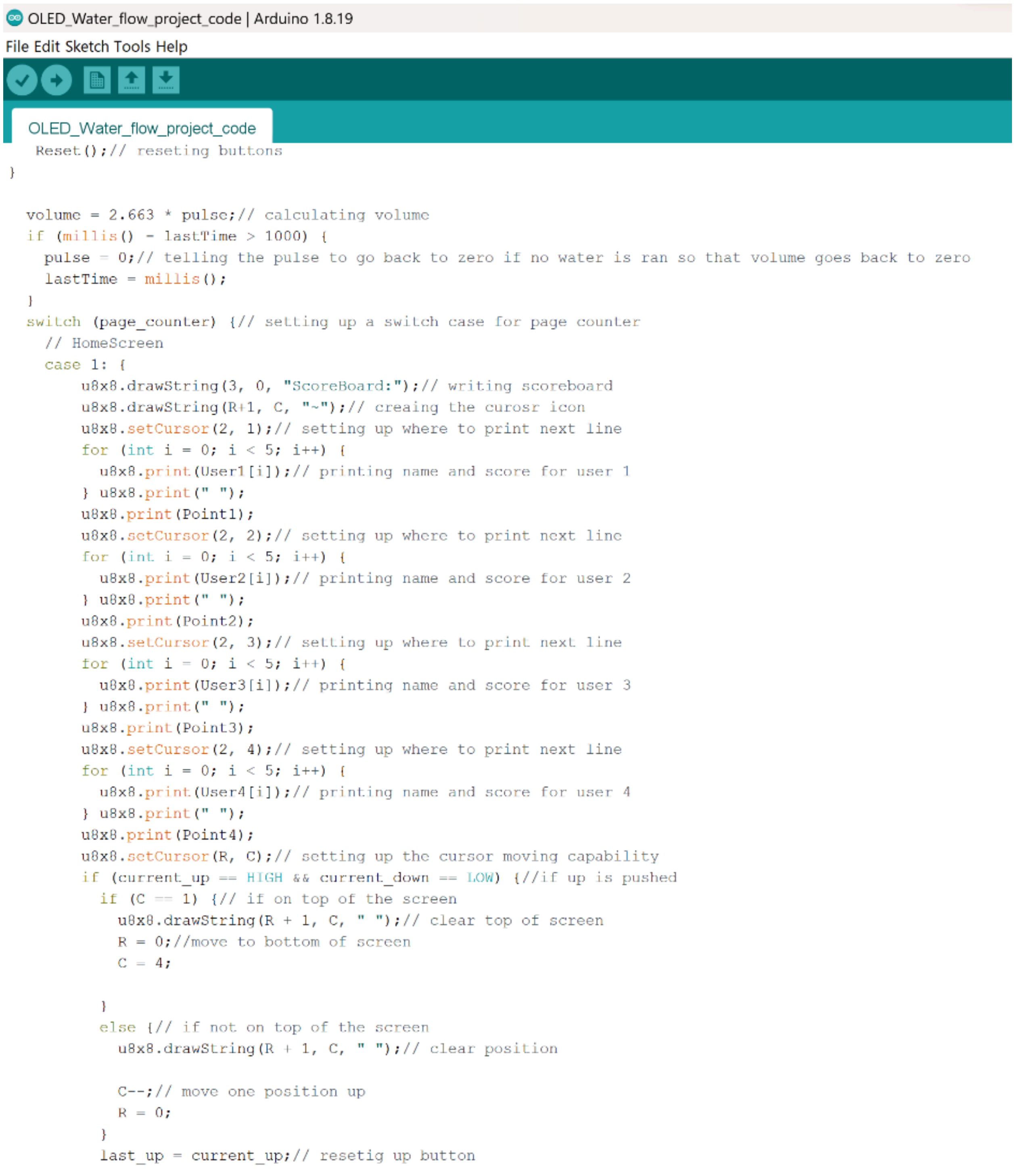

OLED_Water_flow_project_code | Arduino 1.8.19

File Edit Sketch Tools Help

OLED_Water_flow_project_code

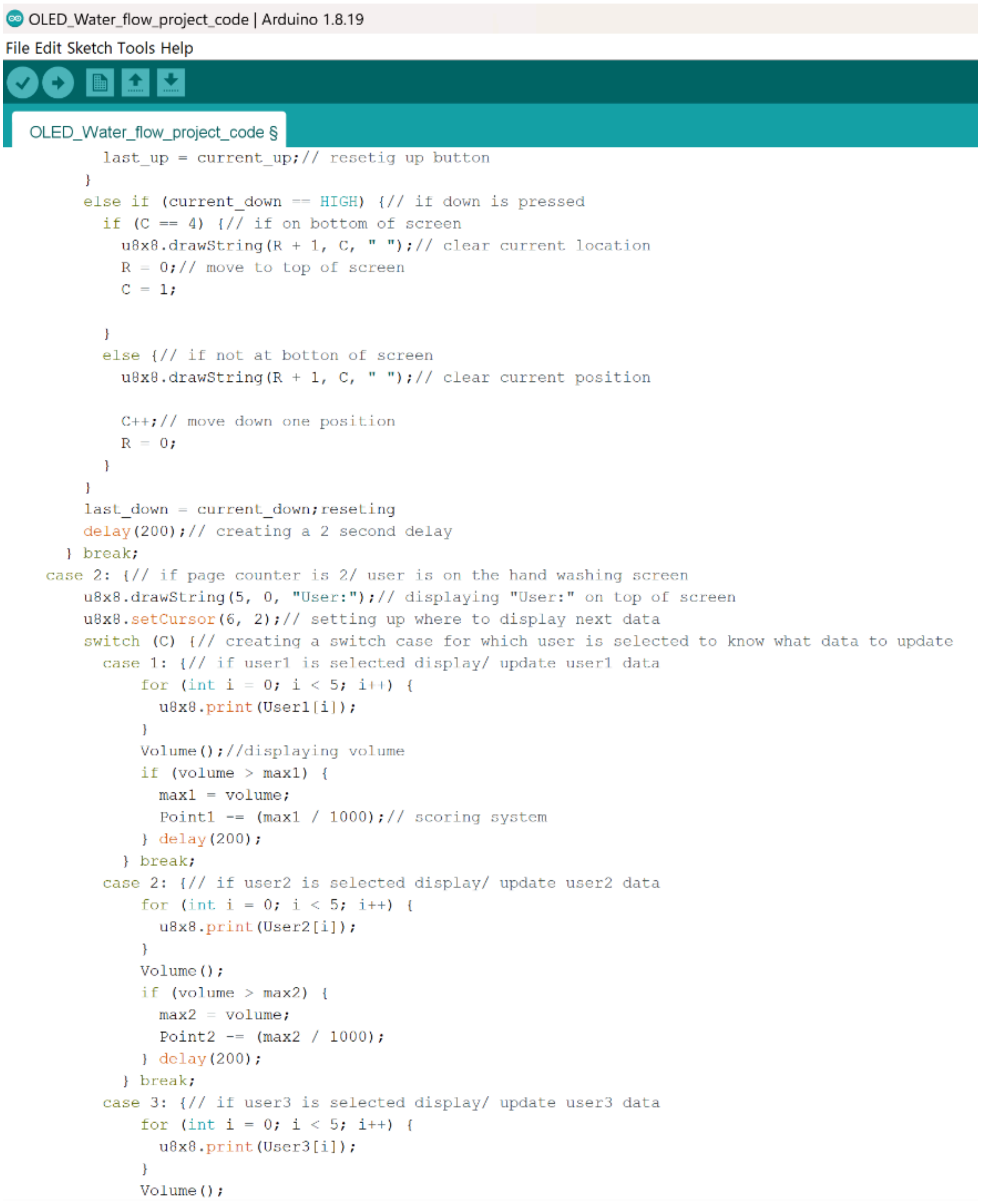
```
void loop() {

  extern volatile unsigned long timer0_millis;// starting the timer
  // Debouncing all the buttons
  current_up = debounce(last_up, BUTTON_UP);
  current_down = debounce(last_down, BUTTON_DOWN);//
  current_select = debounce(last_select, BUTTON_SELECT);
  // setting up the ways in which page counter can be updated to display different menus/screens on the oled
if(page_counter == 1){// if on first screen

  if ( current_down == HIGH && current_select == HIGH) { //When down and select button is pressed page counter cant go below 1

      page_counter = page_counter; //Page doesnt change
    }
   Reset();// resets buttons

if ( current_select == HIGH&&current_down == LOW) { //When select button is pressed page changes to the next page
    u8x8.clear();                     //When page is changed, u8x8 clear to print new page
      page_counter = page_counter + 1; //Page up

}Reset();// Resting button states
}
else if (page_counter==3){// if on last screen

  if ( current_down == LOW && current_select == HIGH) { //When select button is pressed cant go higher than current screen

      page_counter = page_counter; //Page doesnt change
    }
  Reset();// reset button state
  if ( current_down == HIGH && current_select == HIGH) { //When down and selecte button is pressed page goes back one
    u8x8.clear(); // clears the screen
      page_counter = page_counter - 1; //Page down
    }
   Reset();// Resets buttons

}
else{// if on the middle screen then user can go up or down
    if ( current_select == HIGH&&current_down == LOW) { //When up button is pressed
    u8x8.clear();  //clearing screen

      page_counter = page_counter + 1; //Page up

  }Reset();// Resting buttons
  if ( current_down == HIGH && current_select == HIGH) { //When down and select button is pressed
    u8x8.clear();           // clearing screen
      page_counter = page_counter - 1; //Page down
    }
   Reset();// reseting buttons
```

Figure 11: Arduino Code Part 3


OLED_Water_flow_project_code | Arduino 1.8.19
File Edit Sketch Tools Help
OLED_Water_flow_project_code

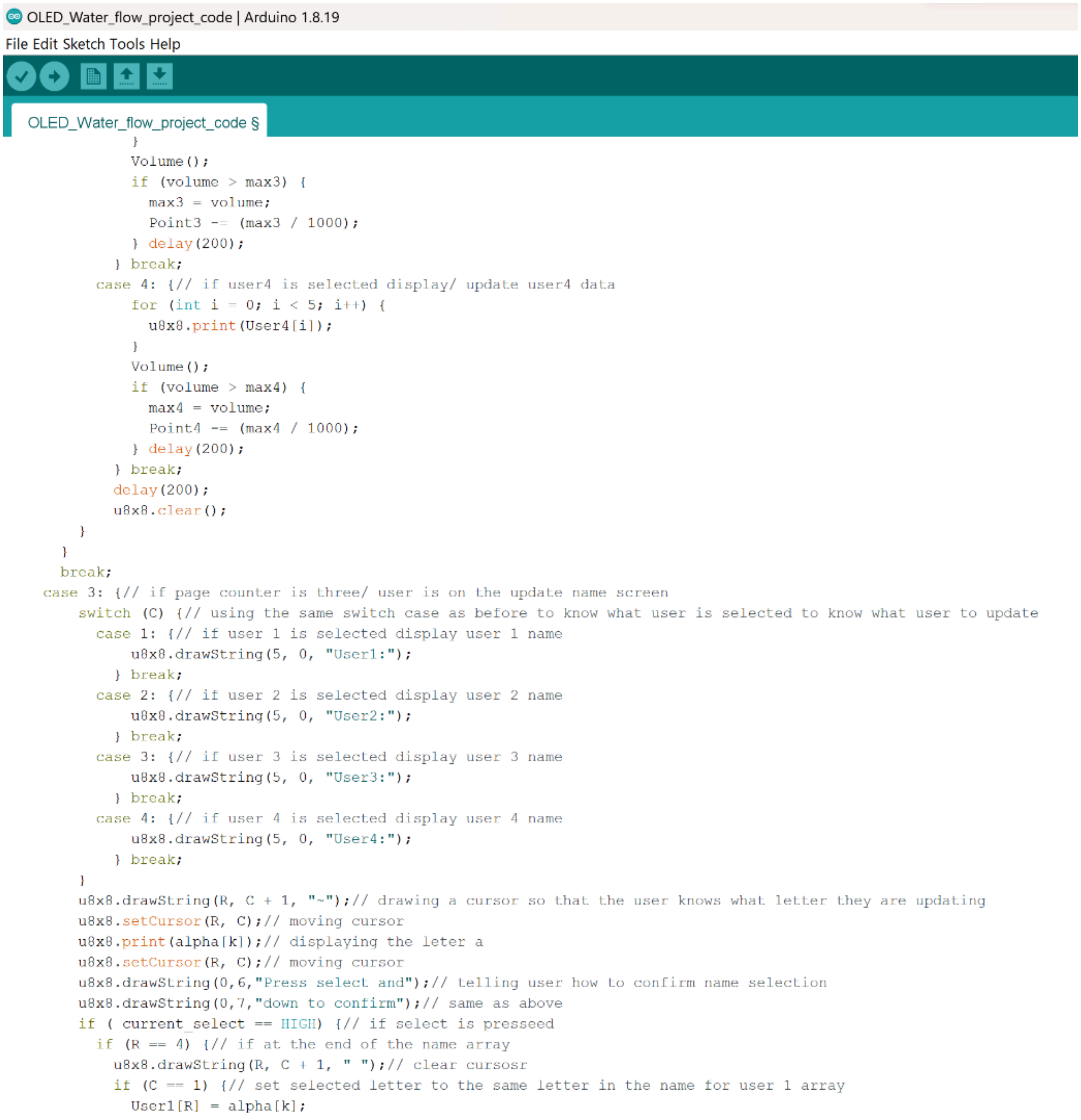
```
    Reset();// reseting buttons
}

  volume = 2.663 * pulse;// calculating volume
  if (millis() - lastTime > 1000) {
    pulse = 0;// telling the pulse to go back to zero if no water is ran so that volume goes back to zero
    lastTime = millis();
  }
  switch (page_counter) {// setting up a switch case for page counter
    // HomeScreen
    case 1: {
        u8x8.drawString(3, 0, "ScoreBoard:");// writing scoreboard
        u8x8.drawString(R+1, C, "~");// creaing the curosr icon
        u8x8.setCursor(2, 1);// setting up where to print next line
        for (int i = 0; i < 5; i++) {
          u8x8.print(User1[i]);// printing name and score for user 1
        } u8x8.print(" ");
        u8x8.print(Point1);
        u8x8.setCursor(2, 2);// setting up where to print next line
        for (int i = 0; i < 5; i++) {
          u8x8.print(User2[i]);// printing name and score for user 2
        } u8x8.print(" ");
        u8x8.print(Point2);
        u8x8.setCursor(2, 3);// setting up where to print next line
        for (int i = 0; i < 5; i++) {
          u8x8.print(User3[i]);// printing name and score for user 3
        } u8x8.print(" ");
        u8x8.print(Point3);
        u8x8.setCursor(2, 4);// setting up where to print next line
        for (int i = 0; i < 5; i++) {
          u8x8.print(User4[i]);// printing name and score for user 4
        } u8x8.print(" ");
        u8x8.print(Point4);
        u8x8.setCursor(R, C);// setting up the cursor moving capability
        if (current_up == HIGH && current_down == LOW) {//if up is pushed
          if (C == 1) {// if on top of the screen
            u8x8.drawString(R + 1, C, " ");// clear top of screen
            R = 0;//move to bottom of screen
            C = 4;

          }
          else {// if not on top of the screen
            u8x8.drawString(R + 1, C, " ");// clear position

            C--;// move one position up
            R = 0;
          }
          last_up = current_up;// resetig up button
```

Figure 12: Arduino Code Part 4


OLED_Water_flow_project_code | Arduino 1.8.19
File Edit Sketch Tools Help
OLED_Water_flow_project_code §

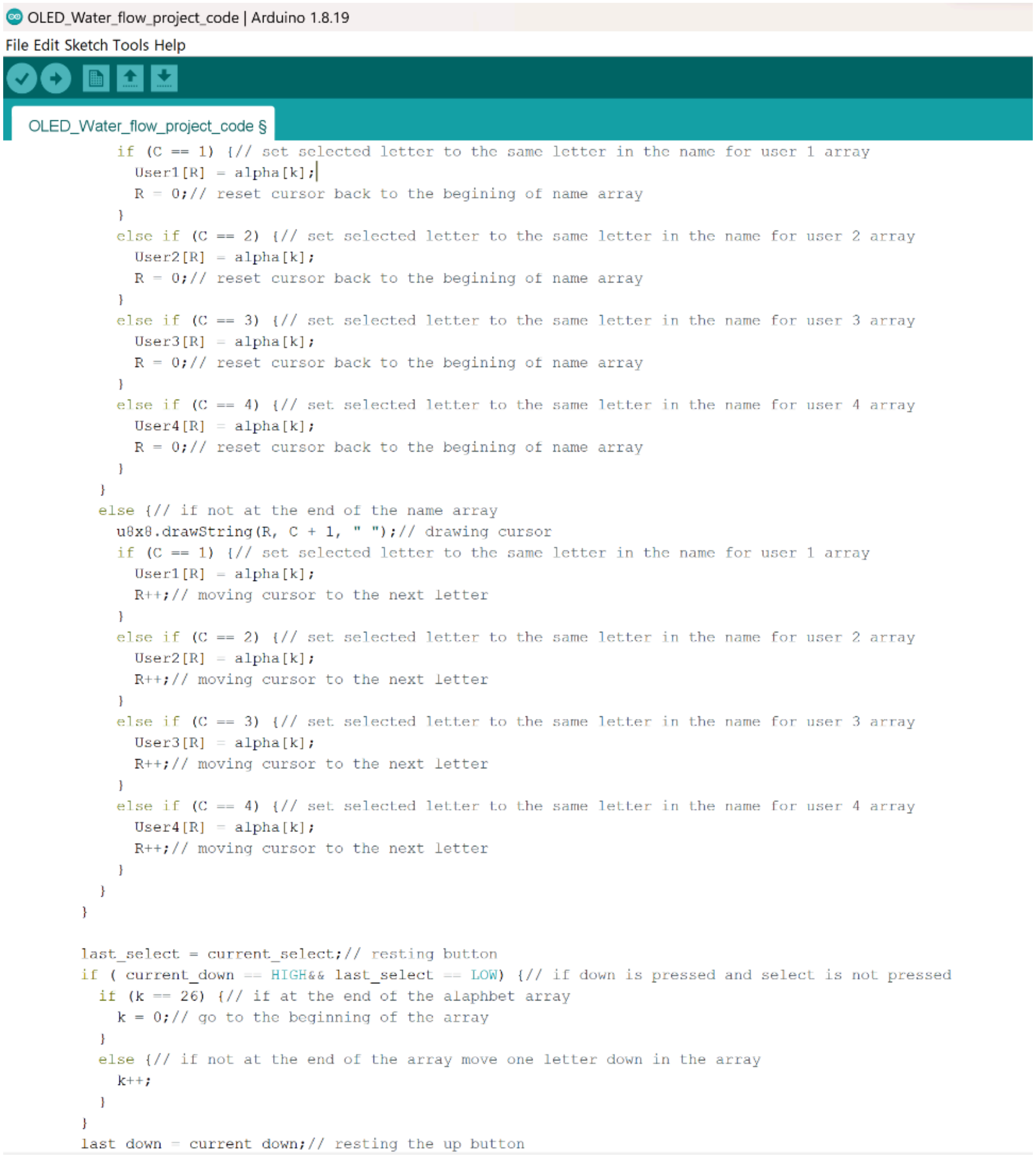
```
          last_up = current_up;// resetig up button
        }
        else if (current_down == HIGH) {// if down is pressed
          if (C == 4) {// if on bottom of screen
            u8x8.drawString(R + 1, C, " ");// clear current location
            R = 0;// move to top of screen
            C = 1;

          }
          else {// if not at botton of screen
            u8x8.drawString(R + 1, C, " ");// clear current position

            C++;// move down one position
            R = 0;
          }
        }
        last_down = current_down;reseting
        delay(200);// creating a 2 second delay
      } break;
    case 2: {// if page counter is 2/ user is on the hand washing screen
        u8x8.drawString(5, 0, "User:");// displaying "User:" on top of screen
        u8x8.setCursor(6, 2);// setting up where to display next data
        switch (C) {// creating a switch case for which user is selected to know what data to update
          case 1: {// if user1 is selected display/ update user1 data
              for (int i = 0; i < 5; i++) {
                u8x8.print(User1[i]);
              }
              Volume();//displaying volume
              if (volume > max1) {
                max1 = volume;
                Point1 -= (max1 / 1000);// scoring system
              } delay(200);
            } break;
          case 2: {// if user2 is selected display/ update user2 data
              for (int i = 0; i < 5; i++) {
                u8x8.print(User2[i]);
              }
              Volume();
              if (volume > max2) {
                max2 = volume;
                Point2 -= (max2 / 1000);
              } delay(200);
            } break;
          case 3: {// if user3 is selected display/ update user3 data
              for (int i = 0; i < 5; i++) {
                u8x8.print(User3[i]);
              }
              Volume();
```

Figure 13: Arduino Code Part 5


OLED_Water_flow_project_code | Arduino 1.8.19
File Edit Sketch Tools Help
OLED_Water_flow_project_code §

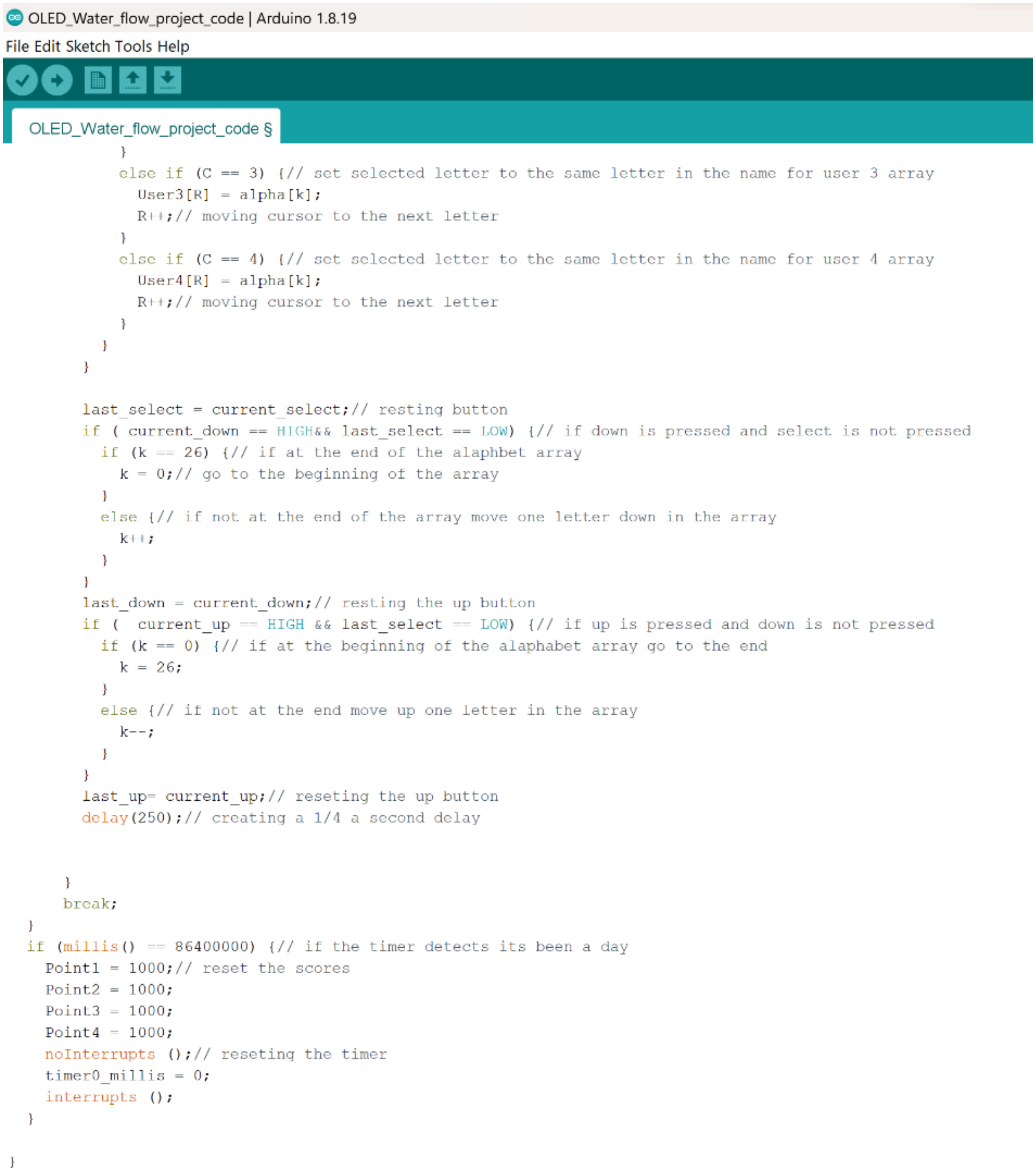
```
            }
            Volume();
            if (volume > max3) {
              max3 = volume;
              Point3 -= (max3 / 1000);
            } delay(200);
          } break;
        case 4: {// if user4 is selected display/ update user4 data
            for (int i = 0; i < 5; i++) {
              u8x8.print(User4[i]);
            }
            Volume();
            if (volume > max4) {
              max4 = volume;
              Point4 -= (max4 / 1000);
            } delay(200);
          } break;
          delay(200);
          u8x8.clear();
      }
    }
    break;
  case 3: {// if page counter is three/ user is on the update name screen
      switch (C) {// using the same switch case as before to know what user is selected to know what user to update
        case 1: {// if user 1 is selected display user 1 name
            u8x8.drawString(5, 0, "User1:");
          } break;
        case 2: {// if user 2 is selected display user 2 name
            u8x8.drawString(5, 0, "User2:");
          } break;
        case 3: {// if user 3 is selected display user 3 name
            u8x8.drawString(5, 0, "User3:");
          } break;
        case 4: {// if user 4 is selected display user 4 name
            u8x8.drawString(5, 0, "User4:");
          } break;
      }
      u8x8.drawString(R, C + 1, "~");// drawing a cursor so that the user knows what letter they are updating
      u8x8.setCursor(R, C);// moving cursor
      u8x8.print(alpha[k]);// displaying the leter a
      u8x8.setCursor(R, C);// moving cursor
      u8x8.drawString(0,6,"Press select and");// telling user how to confirm name selection
      u8x8.drawString(0,7,"down to confirm");// same as above
      if ( current_select == HIGH) {// if select is presseed
        if (R == 4) {// if at the end of the name array
          u8x8.drawString(R, C + 1, " ");// clear cursosr
          if (C == 1) {// set selected letter to the same letter in the name for user 1 array
            User1[R] = alpha[k];
```

Figure 14: Arduino Code Part 6


OLED_Water_flow_project_code | Arduino 1.8.19
File Edit Sketch Tools Help
OLED_Water_flow_project_code §

```
        if (C == 1) {// set selected letter to the same letter in the name for user 1 array
          User1[R] = alpha[k];
          R = 0;// reset cursor back to the begining of name array
        }
        else if (C == 2) {// set selected letter to the same letter in the name for user 2 array
          User2[R] = alpha[k];
          R = 0;// reset cursor back to the begining of name array
        }
        else if (C == 3) {// set selected letter to the same letter in the name for user 3 array
          User3[R] = alpha[k];
          R = 0;// reset cursor back to the begining of name array
        }
        else if (C == 4) {// set selected letter to the same letter in the name for user 4 array
          User4[R] = alpha[k];
          R = 0;// reset cursor back to the begining of name array
        }
      }
      else {// if not at the end of the name array
        u8x8.drawString(R, C + 1, " ");// drawing cursor
        if (C == 1) {// set selected letter to the same letter in the name for user 1 array
          User1[R] = alpha[k];
          R++;// moving cursor to the next letter
        }
        else if (C == 2) {// set selected letter to the same letter in the name for user 2 array
          User2[R] = alpha[k];
          R++;// moving cursor to the next letter
        }
        else if (C == 3) {// set selected letter to the same letter in the name for user 3 array
          User3[R] = alpha[k];
          R++;// moving cursor to the next letter
        }
        else if (C == 4) {// set selected letter to the same letter in the name for user 4 array
          User4[R] = alpha[k];
          R++;// moving cursor to the next letter
        }
      }
    }

    last_select = current_select;// resting button
    if ( current_down == HIGH&& last_select == LOW) {// if down is pressed and select is not pressed
      if (k == 26) {// if at the end of the alaphbet array
        k = 0;// go to the beginning of the array
      }
      else {// if not at the end of the array move one letter down in the array
        k++;
      }
    }
    last_down = current_down;// resting the up button
```

Figure 15: Arduino Code Part 7


OLED_Water_flow_project_code | Arduino 1.8.19
File Edit Sketch Tools Help
OLED_Water_flow_project_code §

```
          }
          else if (C == 3) {// set selected letter to the same letter in the name for user 3 array
            User3[R] = alpha[k];
            R++;// moving cursor to the next letter
          }
          else if (C == 4) {// set selected letter to the same letter in the name for user 4 array
            User4[R] = alpha[k];
            R++;// moving cursor to the next letter
          }
        }
      }

      last_select = current_select;// resting button
      if ( current_down == HIGH&& last_select == LOW) {// if down is pressed and select is not pressed
        if (k == 26) {// if at the end of the alaphbet array
          k = 0;// go to the beginning of the array
        }
        else {// if not at the end of the array move one letter down in the array
          k++;
        }
      }
      last_down = current_down;// resting the up button
      if (  current_up == HIGH && last_select == LOW) {// if up is pressed and down is not pressed
        if (k == 0) {// if at the beginning of the alaphabet array go to the end
          k = 26;
        }
        else {// if not at the end move up one letter in the array
          k--;
        }
      }
      last_up= current_up;// reseting the up button
      delay(250);// creating a 1/4 a second delay


    }
    break;
  }
  if (millis() == 86400000) {// if the timer detects its been a day
    Point1 = 1000;// reset the scores
    Point2 = 1000;
    Point3 = 1000;
    Point4 = 1000;
    noInterrupts ();// reseting the timer
    timer0_millis = 0;
    interrupts ();
  }

}
```

Figure 16: Arduino Code Part 8

References

[1] Lafortune, Guillaume. "THE UNITED STATES RANKED 35TH GLOBALLY ON SUSTAINABLE DEVELOPMENT." *Sdgindex*, edited by Grayson Fuller, Sustainability Development Solutions Network , 28 June 2019. Path: sgindex;news&insights;THE UNITED STATES RANKED 35TH GLOBALLY ON SUSTAINABLE DEVELOPMENT

[2] Ramos, Katherine. "GEEN 1400." University of Colorado Boulder, 17 Jan. 2023, Boulder, CO

[3] Statistics and Facts,Environmental Protection Agency, 24 Apr. 2023, www.epa.gov/watersense/statistics-and-facts

[4] *Water Use Around the World*,CDC,Jan.2020, www.cdc.gov/globalhealth/infographics/food-water/water_use.htm