FORMAN CHRISTIAN COLLEGE
(A CHARTERED UNIVERSITY)

BY LOVE SERVE ONE ANOTHER

Estd. 1864

CSCS 306 – Embedded Systems
Fall 22

Assignment 1

Stopwatch

Adeel Akmal – 231465420
Muhammad Sameed Gilani - 231488347

INTRODUCTION:

In this assignment, we made a Stopwatch that can go up to 99, consisting of two Seven-segment displays and 5 switches.

The two Seven-segment displays are multiplexed using two NPN transistors.

Using the first two switches, we can use one to adjust the unit digit on the display and the other button to adjust the tens digit of the display. The digits count up to 9 and wrap back to 0, respectively.

The third button is used to start the Stopwatch. The display begins to decrement when the start button is clicked after setting a number on them.

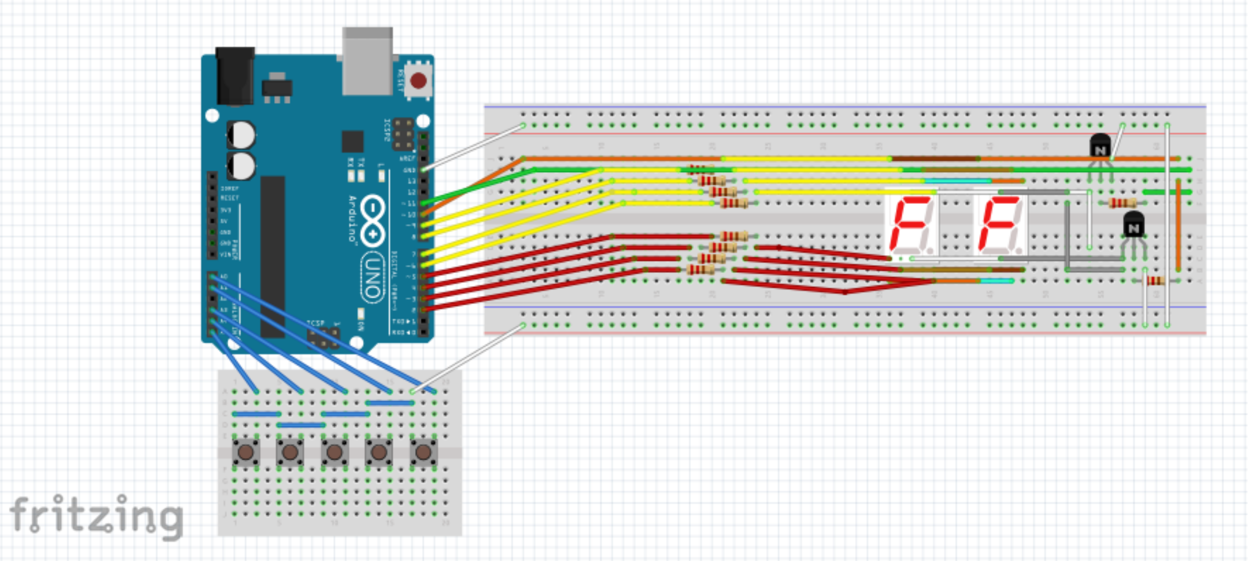
There are two additional buttons to pause and reset the Stopwatch. The reset button can be used at any point to reset both displays back to 0.

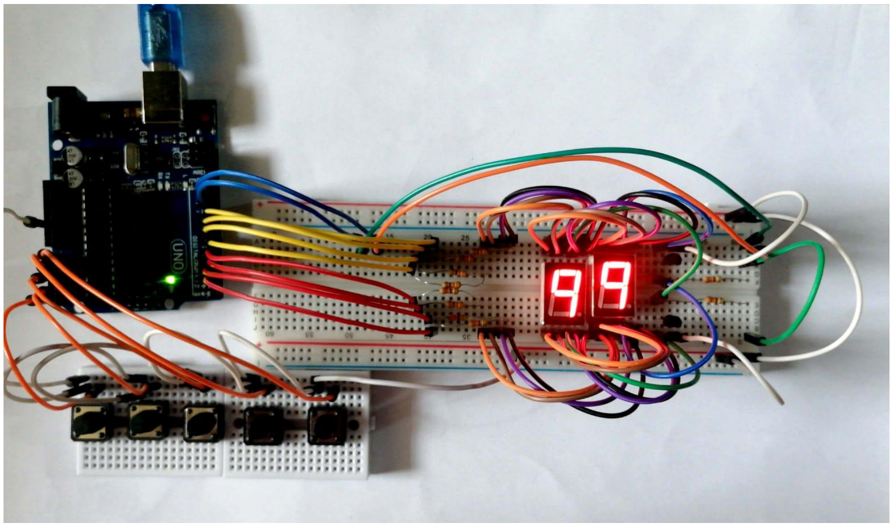
The pause button can pause the Stopwatch, once it has been started. And the Stopwatch can be resumed by clicking the start button.

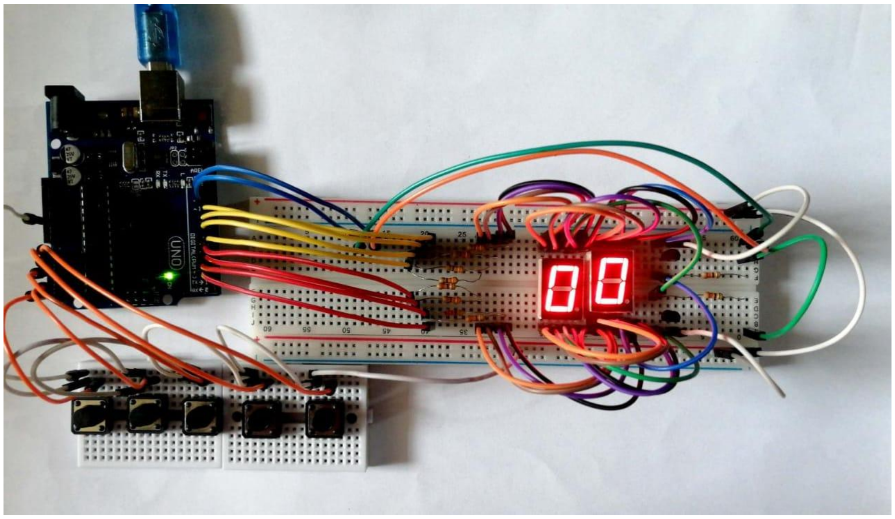
COMPONENTS:

- Arduino Uno
- 7-segment displays - x2
- 330 Ω resistor - x10
- 2N2222A NPN Transistor - x2
- Tactile push button switch - x5
- Jumper wires

CIRCUIT DIAGRAM:







CODE:

```
// Pins for each segment
int a = 8;
int b = 9;
int c = 3;
int d = 4;
int e = 5;
int f = 7;
int g = 6;
int dp = 2;

int digit1 = 10;  // On pin for the units digit
int digit2 = 11;  // On pin for the tens digit

int buttonD1 = A0;  // button to increment units digit
int buttonD2 = A1;  // button to increment tens digit
int buttonStart = A2; // button to start the timer
int buttonPause = A4; // button to pause the timer
int buttonReset = A5; // button to reset the timer to 0

int buttonCounterD1 = 0;  // Counter for the number of times the units digit is
pressed.
int buttonCounterD2 = 0;  // Counter for the number of times the tens digit is
pressed.

int currentButtonD1 = 0;  // Current state of the units digit button. (either high or
low)
int lastButtonD1 = 0;     // last state of the units digit button. (either high or
low)

int currentButtonD2 = 0;  // Current state of the tens digit button. (either high or
low)
int lastButtonD2 = 0;     // last state of the tens digit button. (either high or low)

int currentButtonPause = 0; // Current state of the pause button. (either high or low)
int lastButtonPause = 0;  // last state of the button. (either high or low)

// Counters to track time
int startTime = 0;
int endTime;


void numOnDisplay(int num);               // Displays the appropriate number on the 7
seg display
void refreshDisplay(int dig1, int dig0);  // Used to switch both displays on and off
(With the transistor)
```

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);

  // Initializing all pins

  pinMode(a, OUTPUT);
  pinMode(b, OUTPUT);
  pinMode(c, OUTPUT);
  pinMode(d, OUTPUT);
  pinMode(e, OUTPUT);
  pinMode(f, OUTPUT);
  pinMode(g, OUTPUT);

  pinMode(digit1, OUTPUT);
  pinMode(digit2, OUTPUT);

  pinMode(buttonD1, INPUT_PULLUP);
  pinMode(buttonD2, INPUT_PULLUP);
  pinMode(buttonStart, INPUT_PULLUP);
  pinMode(buttonPause, INPUT_PULLUP);
  pinMode(buttonReset, INPUT_PULLUP);

}

void loop() {

  // This loop allows to set the numbers of the displays
  // loop will break once the start button is pressed
  while(1){

    refreshDisplay(buttonCounterD1,buttonCounterD2); // Constantly refreshes the
display with the current numbers

    // Store current states of both digits
    currentButtonD1 = digitalRead(buttonD1);
    currentButtonD2 = digitalRead(buttonD2);

    // Breaks this while(1) loop once the Start button is clicked
    // Dont need previous state, as multiple highs or lows read wont matter in this
case
    if(digitalRead(buttonStart) == LOW){
      break;
    }

    // Resets the displays to 00, once the reset button is clicked
    // Dont need previous state, as multiple highs or lows read wont matter in this
case
    if(digitalRead(buttonReset) == LOW){
      buttonCounterD1 = 0;
      buttonCounterD2 = 0;
      refreshDisplay(buttonCounterD1,buttonCounterD2);
    }

    // We need to compare the last and current state of the button, so that it only
increments once
    // As multiple highs and lows may may be read when you click the button
    if(currentButtonD1 != lastButtonD1){
      // if button is clicked, the number is incremented
      if(currentButtonD1 == LOW){
        buttonCounterD1++;

      if( buttonCounterD1 > 9) buttonCounterD1 = 0 ; // number wraps back to 0 afer 9
    }

    lastButtonD1 = currentButtonD1; // last state is now the current state
    }

    if(currentButtonD2 != lastButtonD2){
        if(currentButtonD2 == LOW){
          buttonCounterD2++;

        if( buttonCounterD2 > 9) buttonCounterD2 = 0 ;
      }

      lastButtonD2 = currentButtonD2;
      }
  }


  buttonCounterD1 += 1; // +1, so that the first units digit isnt instantly
incremented when the start button is clicked

  // Loop will run until the timer runs out or the reset button is clicked
  // each iteration, after 1 sec, the timer decrements by 1
  while(2){
    if(buttonCounterD1 == 0 && buttonCounterD2 == 0){
      break;
    }
    if(digitalRead(buttonReset) == LOW){
      buttonCounterD1 = 0;
      buttonCounterD2 = 0;
      break;
    }

    refreshDisplay(buttonCounterD1,buttonCounterD2);  // Constantly refreshes the
display with the current numbers

    currentButtonPause = digitalRead(buttonPause);


    // When the pause button is clicked, we enter this loop
    if(currentButtonPause != lastButtonPause){
      lastButtonPause = currentButtonPause;
      while(3){
        if(digitalRead(buttonStart) == LOW) break; // If start button is clicked, the
timer resumes

        if(digitalRead(buttonReset) == LOW){  // If the reset button is clicked,
digits are reset to 00 a
        buttonCounterD1 = 0;
        buttonCounterD2 = 0;
        break;
        }

        refreshDisplay(buttonCounterD1,buttonCounterD2);

      }
    }

    // Decrements the timer by 1 after 1 second has passed
    endTime = millis();
    if ((endTime - startTime) >= 1000){

      if (--buttonCounterD1 < 0){
          buttonCounterD1 = 9;
          if (--buttonCounterD2 < 0) buttonCounterD2 = 9;
      }

      startTime = endTime;
    }


  }

  // Loop flashes 00 on the displays 3 times
  for(int i = 0; i != 4; i++){
  numOnDisplay(0);  // 0 on the display

  // Turns both displays off
  digitalWrite(digit1,LOW);
  digitalWrite(digit2,LOW);
  delay(400);

  // Turns both displays on. (0 is displayed, from the numOnDisplay call)
  digitalWrite(digit1,HIGH);
  digitalWrite(digit2,HIGH);
  delay(400);
  }

  // Displays are initialized to 00.
  // Main loop restarts with 00 displayed.
  refreshDisplay(0,0);

}
```

```
void refreshDisplay(int dig1, int dig0){

  // using transistors, we can switch displays.
  // The transistor that is HIGH, will turn on.
  // with small delays we can simulate both of them being on simultaneously

  digitalWrite(digit1, HIGH);  // displays digit 0
  digitalWrite(digit2, LOW );
  numOnDisplay(dig0);

  delay(5);

  digitalWrite(digit1, LOW);    // displays digit 1
  digitalWrite(digit2, HIGH);
  numOnDisplay(dig1);

  delay(5);


}


void numOnDisplay(int num){
  // Turns the appropriate segments HIGH on the display
  // According to the parameter, num

  if(num == 0){
    digitalWrite(a,HIGH);
    digitalWrite(b,HIGH);
    digitalWrite(c,HIGH);
    digitalWrite(d,HIGH);
    digitalWrite(e,HIGH);
    digitalWrite(f,HIGH);
    digitalWrite(g,LOW);
  }

  if(num == 1){
    digitalWrite(a,LOW);
    digitalWrite(b,HIGH);
    digitalWrite(c,HIGH);
    digitalWrite(d,LOW);
    digitalWrite(e,LOW);
    digitalWrite(f,LOW);
    digitalWrite(g,LOW);
  }

  if(num == 2){
    digitalWrite(a,HIGH);
    digitalWrite(b,HIGH);
    digitalWrite(c,LOW);
    digitalWrite(d,HIGH);
    digitalWrite(e,HIGH);
    digitalWrite(f,LOW);
    digitalWrite(g,HIGH);
  }

  if(num ==3){
    digitalWrite(a,HIGH);
    digitalWrite(b,HIGH);
    digitalWrite(c,HIGH);
    digitalWrite(d,HIGH);
    digitalWrite(e,LOW);
    digitalWrite(f,LOW);
    digitalWrite(g,HIGH);
  }

  if(num == 4){
    digitalWrite(a,LOW);
    digitalWrite(b,HIGH);
    digitalWrite(c,HIGH);
    digitalWrite(d,LOW);
    digitalWrite(e,LOW);
    digitalWrite(f,HIGH);
    digitalWrite(g,HIGH);
  }


  if(num == 5){
    digitalWrite(a,HIGH);
    digitalWrite(b,LOW);
    digitalWrite(c,HIGH);
    digitalWrite(d,HIGH);
    digitalWrite(e,LOW);
    digitalWrite(f,HIGH);
    digitalWrite(g,HIGH);
  }

  if(num == 6){
    digitalWrite(a,HIGH);
    digitalWrite(b,LOW);
    digitalWrite(c,HIGH);
    digitalWrite(d,HIGH);
    digitalWrite(e,HIGH);
    digitalWrite(f,HIGH);
    digitalWrite(g,HIGH);
  }

  if(num == 7){
    digitalWrite(a,HIGH);
    digitalWrite(b,HIGH);
    digitalWrite(c,HIGH);
```

```
    digitalWrite(d,LOW);
    digitalWrite(e,LOW);
    digitalWrite(f,LOW);
    digitalWrite(g,LOW);

  }
  if(num == 8){
    digitalWrite(a,HIGH);
    digitalWrite(b,HIGH);
    digitalWrite(c,HIGH);
    digitalWrite(d,HIGH);
    digitalWrite(e,HIGH);
    digitalWrite(f,HIGH);
    digitalWrite(g,HIGH);
  }

  if(num == 9){
    digitalWrite(a,HIGH);
    digitalWrite(b,HIGH);
    digitalWrite(c,HIGH);
    digitalWrite(d,LOW);
    digitalWrite(e,LOW);
    digitalWrite(f,HIGH);
    digitalWrite(g,HIGH);

  }

}
```

LOGIC:

- The two 7-segment displays are multiplexed using two transistors. The transistors act as a switch, by giving the base pin of the transistor 5v, the collector acts as ground (0v).
  So we can selectively turn one display on by giving its resistor 5v and by introducing small delays between the switching of both the displays, we simulate both the displays being on simultaneously.
- The switches are used to take input with digitalRead(). When it returns LOW, the switch has been clicked. With that, we can use code to do an appropriate action in response to the click.
- The first while(1) loop, runs until the user clicks the start button.
  Inside the loop, refreshDisplay() constantly updates the displays with the newest numbers, as the user adjusts the units and tens digits by their appropriate buttons.
  At any point the user can click the reset buttons to set the digits to 0.
- Once the start button is clicked we enter the while(2) loop. Which runs until either the timer reaches 00 or the user clicks the reset button.
  During the timers running, the user can click the pause button at any point to pause the timer and the start button to resume the timer.
- We use millis() to track 1 second passing. We do not use delay, as it halts the program execution instead of waiting for time to pass, which results in unwanted output.
- Once the timer reaches 0, the display flashes 00 three times. Before resetting to 00. After which the user can adjust the number again and restart the timer.

REFERENCES:

https://www.jameco.com/Jameco/workshop/TechTip/working-with-seven-segment-displays.html
https://www.viralsciencecreativity.com/post/arduino-seven-segment-display-counter-push-button-code-explanation