EMBEDDED SYSTEM AND DESIGN

EXPERIMENT NO. 3

Clock

Name: NAVARRO, ROD GERYK C.

Course/Section: CPE160P-4/A1

Group No.:

Date of Performance: 08/29/2024
Date of Submission: 08/31/2024

CYREL O. MANLISES, PH.D.
Instructor

DISCUSSION

Firstly, in the experiment I created and designed a 12-hour clock circuit with the use of a 4-digit seven- segment LED display. I have use 2 buttons/switches in the circuit which will allow the user to manually adjust the time of the clock. The first button on the right will allow me or the user to choose what I wanted to increase or decrease in the digits of the clock, whether it s the hour or the minutes of the clock. The second button s purpose is to be able to increment or decrement the time of the chosen time the user wants to change. So, the clock displays on the LEDs are the hour and the minutes. The first 2 digits from the left are allocated for the hour of the clock and the last 2 digits are for the minutes.

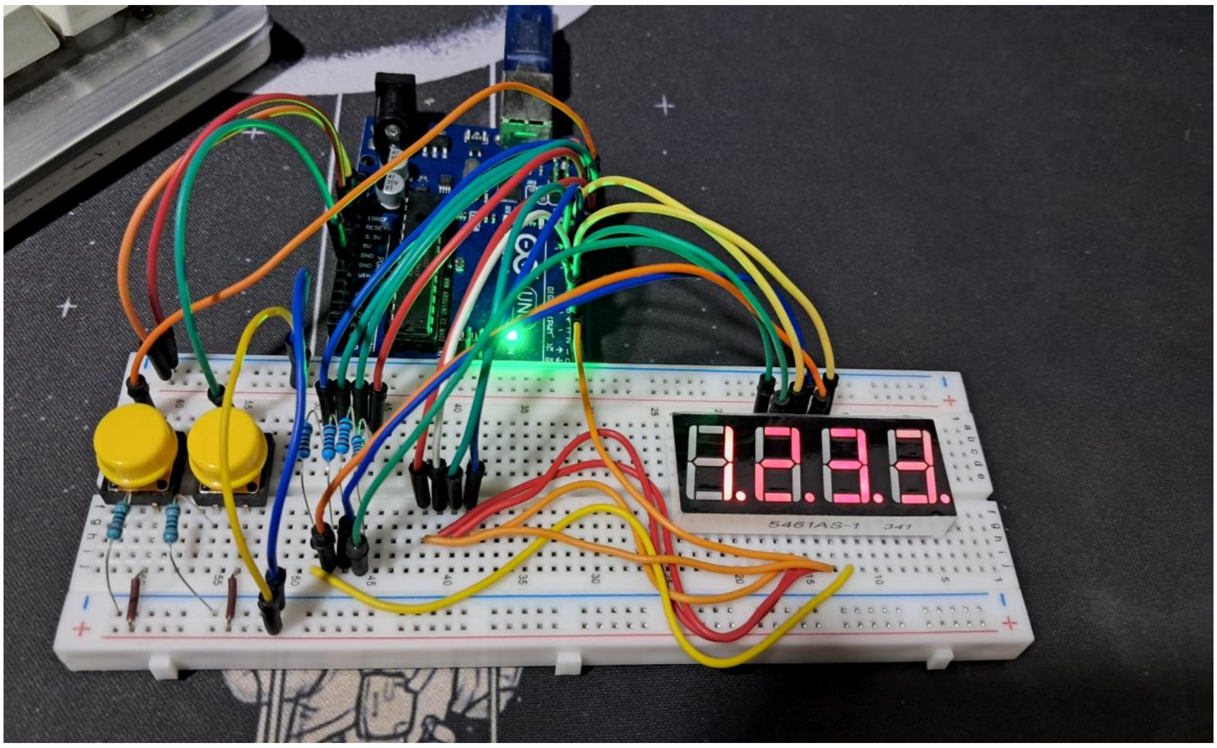
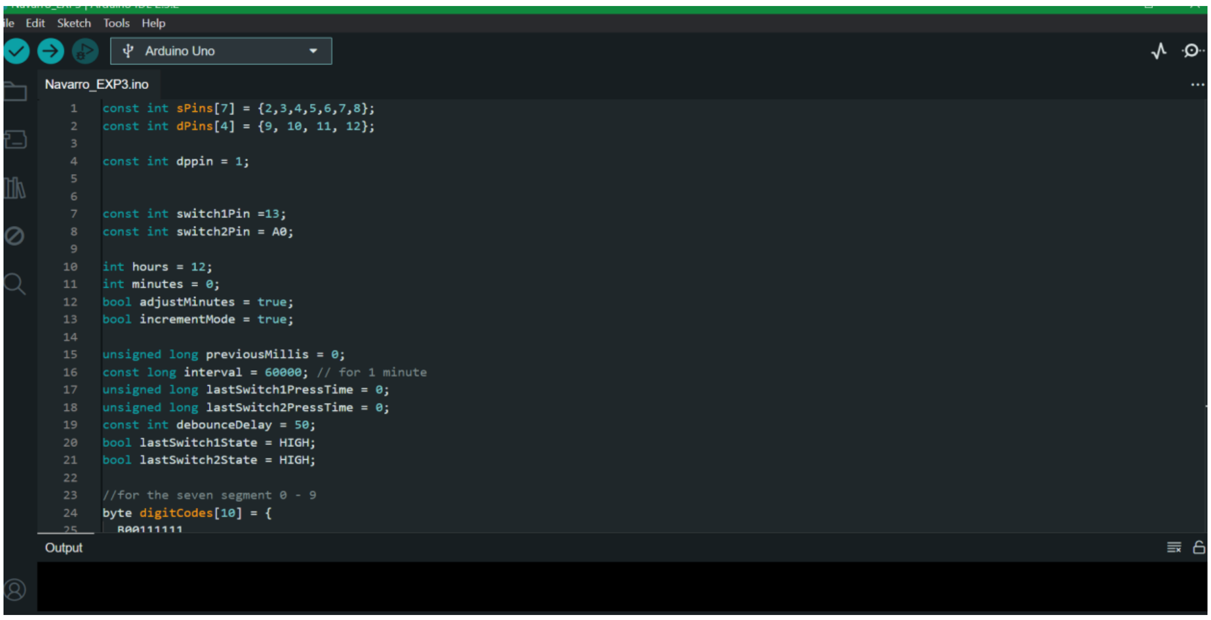


Figure 1: *The circuit connection of the 4-digit seven- segment LED display, and the push button switches connected to the Arduino board.*

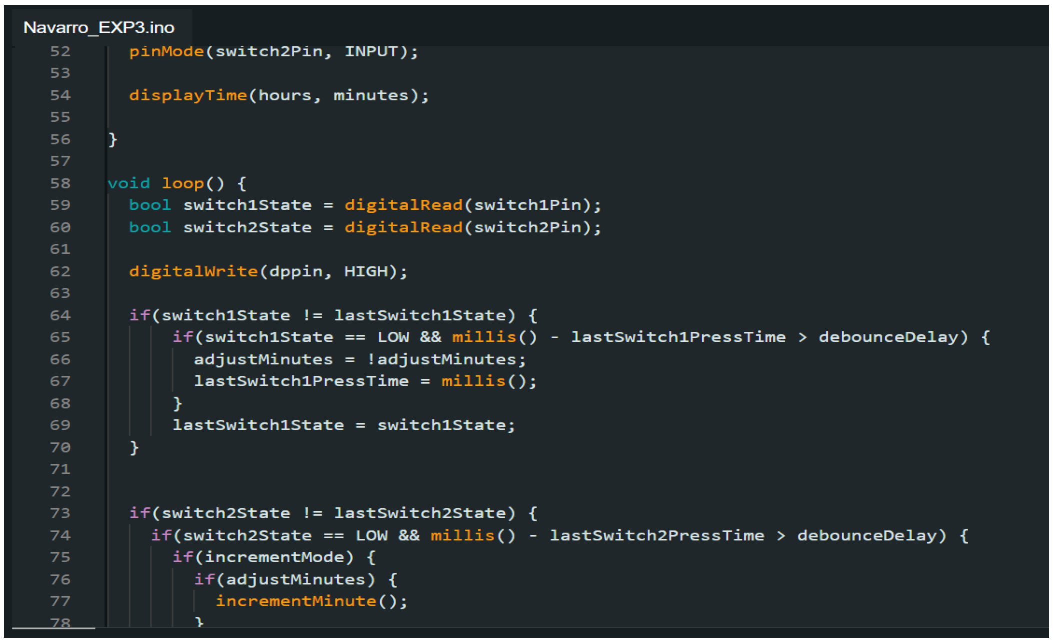
The next thing I did in the experiment is to program the clock in order to have a running real-time clock with the use of a seven-segment display, and to control the hour and minutes of the clock digital pins. So, it starts with assigning the pins for the seven-segment display and the pins definitions for the switches as well as the time variables. In the experiment Ive use a seven-segment display encoding for all the digits of the clock from 0-9.



```
const int sPins[7] = {2,3,4,5,6,7,8};
const int dPins[4] = {9, 10, 11, 12};

const int dppin = 1;


const int switch1Pin =13;
const int switch2Pin = A0;

int hours = 12;
int minutes = 0;
bool adjustMinutes = true;
bool incrementMode = true;

unsigned long previousMillis = 0;
const long interval = 60000; // for 1 minute
unsigned long lastSwitch1PressTime = 0;
unsigned long lastSwitch2PressTime = 0;
const int debounceDelay = 50;
bool lastSwitch1State = HIGH;
bool lastSwitch2State = HIGH;

//for the seven segment 0 - 9
byte digitCodes[10] = {
  B00111111
```

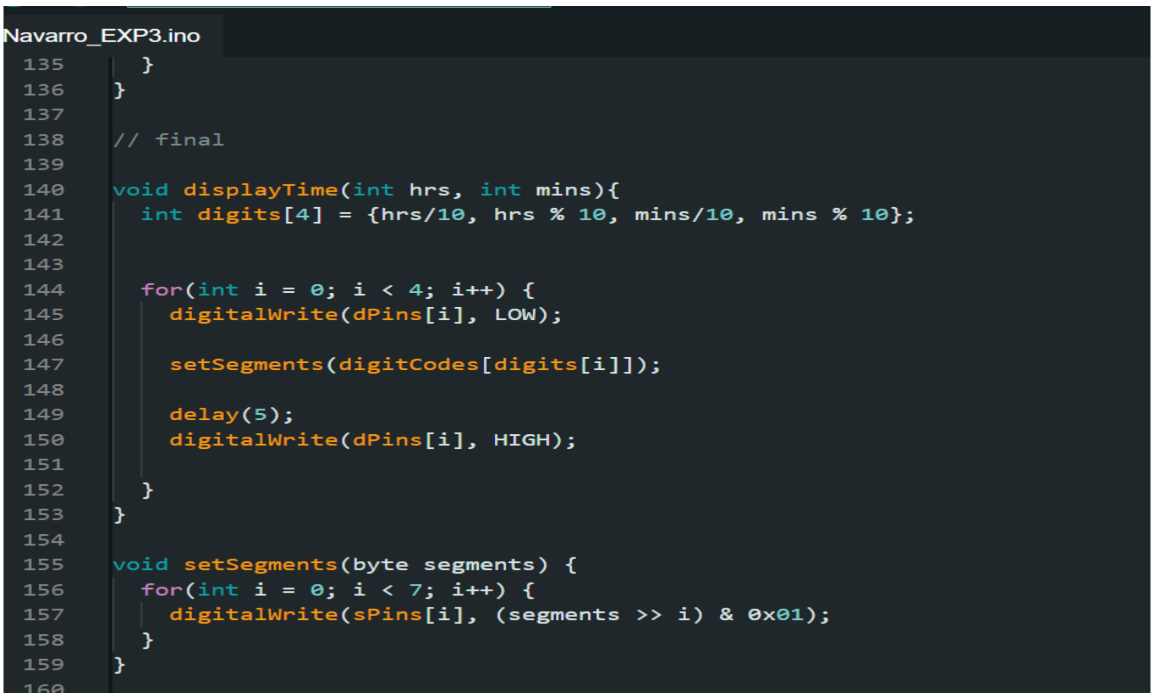
Figure 2: The initialization of the components.

So, after the initialization of the code if where I set the void setup (), and void loop (), where most of the conditions that I ve use Is the if else statement for setting the condition of the controls and running of the clock. It includes the display time, incrementation and decrementation, and delay time of both the hours and minutes of the clock, which I created various function for setting tasks to be able to make the code look clean and easy to debug. In the end of the program is where I configure the display of the digits for the hours and minutes, which I ve use a for loop for the convenience of printing the outputs of the digits pins for the hours and minutes.



```
  pinMode(switch2Pin, INPUT);

  displayTime(hours, minutes);

}

void loop() {
  bool switch1State = digitalRead(switch1Pin);
  bool switch2State = digitalRead(switch2Pin);

  digitalWrite(dppin, HIGH);

  if(switch1State != lastSwitch1State) {
      if(switch1State == LOW && millis() - lastSwitch1PressTime > debounceDelay) {
        adjustMinutes = !adjustMinutes;
        lastSwitch1PressTime = millis();
      }
      lastSwitch1State = switch1State;
  }


  if(switch2State != lastSwitch2State) {
    if(switch2State == LOW && millis() - lastSwitch2PressTime > debounceDelay) {
      if(incrementMode) {
        if(adjustMinutes) {
          incrementMinute();
        }
```

Figure 3: The read of the state of the switches and controls of the clock.


Navarro_EXP3.ino

```
  }
}

// final

void displayTime(int hrs, int mins){
  int digits[4] = {hrs/10, hrs % 10, mins/10, mins % 10};


  for(int i = 0; i < 4; i++) {
    digitalWrite(dPins[i], LOW);

    setSegments(digitCodes[digits[i]]);

    delay(5);
    digitalWrite(dPins[i], HIGH);

  }
}

void setSegments(byte segments) {
  for(int i = 0; i < 7; i++) {
    digitalWrite(sPins[i], (segments >> i) & 0x01);
  }
}
```

Figure 4: *Display outputs for the digits and segments of the hours and minutes on the seven-segment display.*

CONCLUSION

In conclusion, this experiment involved designing and programming a 12-hour clock using a 4-digit seven-segment LED display and two buttons. The first button allows the user to select whether to adjust the hours or minutes, while the second button is used to increase or decrease the chosen time. The left two digits of the display show the hour, and the right two digits show the minutes. The clock was programmed to run in real-time, with functions for adjusting the time and managing the display of digits. Various conditions and functions were used to keep the code clean and easy to understand, allowing the clock to function smoothly and accurately display the set time. The experiment successfully demonstrated how to build a basic clock using embedded systems and microcontroller programming.