REPORTS ON MECHATRONICS SYSTEM INTEGRATION

WEEK 2

DIGITAL LOGIC DESIGN

SECTION 1, SEMESTER 2, 23/24

Date of Experiment:

Date of Submission: ***20/3/2024***

Team Members:

1. IKMAL FIKRI BIN KHAIRUL KHUBAIDILLAH (2218723)
1. MUHAMMAD IRFAN BIN ROSDIN (2214417)
2. ASYRAAF AIMAN MD HASSAN (2010705)
3. MUHAMMAD IKHWAN BIN MUHAMMAD FAUZI (22188845)

Table of Content

Objective

The goal of this experiment is to obtain hands-on experience connecting a common cathode 7-segment display to an Arduino Uno board. By doing this, we will get an understanding of the fundamentals of connecting electronic components, such as segment displays and pushbuttons, to the Arduino platform by constructing the circuit and uploading the supplied Arduino code. In this experiment we learned how to use push buttons to increase and decrease a count shown on the 7-segment display.

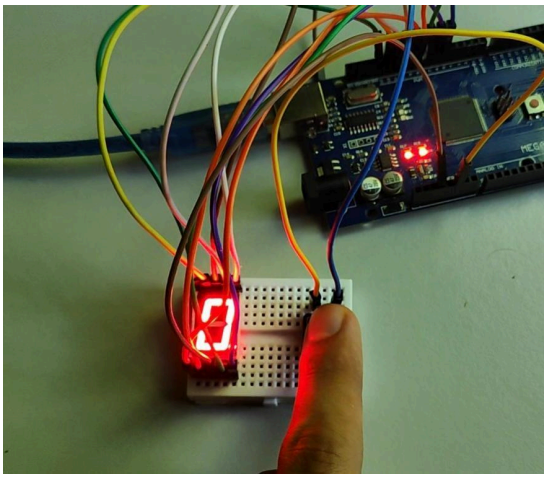
Abstract

In this experiment, we interfaced a common cathode 7-segment display with an Arduino Uno board. The circuit setup involved connecting each of the 7 segments of the display to separate digital pins on the Arduino, along with appropriate current-limiting resistors. Additionally, push buttons were connected to the Arduino to manually increment the count displayed on the 7-segment display and reset it to zero.

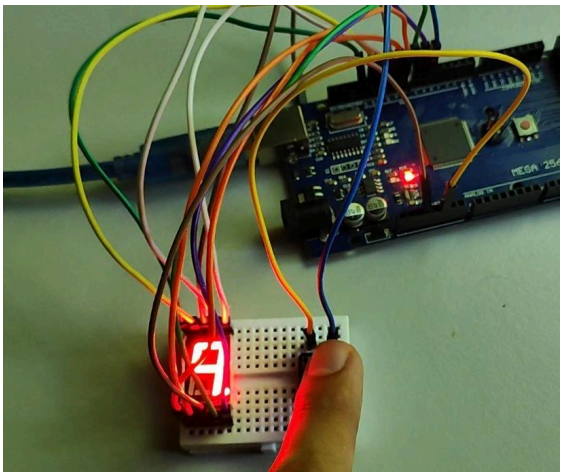
Upon uploading the provided Arduino code, we were able to observe the sequential display of numbers from 0 to 9 on the 7-segment display by pressing the increment button. Furthermore, pressing the reset button reset the count to zero, demonstrating basic control of the display using push buttons.

Through this experiment, we gained practical insights into interfacing electronic components with Arduino, understanding the principles of controlling a 7-segment display, and utilizing push buttons for manual input. This foundational knowledge serves as a basis for more complex projects involving Arduino-based digital logic systems and interfacing applications.

Result



Starting with sequence 0



End with sequence 9

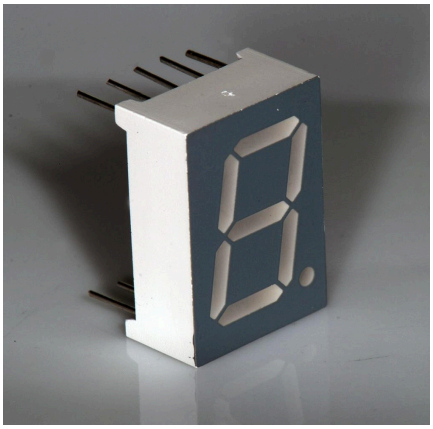
The experiment successfully demonstrated the interfacing of a common cathode 7-segment display with an Arduino Uno board. Upon completing the circuit setup and uploading the provided Arduino code, we observed the following results:

1. **Sequential Display**: Pressing the increment button caused the count displayed on the 7-segment display to increment sequentially from 0 to 9. Each number was displayed for a brief period before transitioning to the next, providing visual feedback of the count incrementing.
2. **Manual Control**: The pushbuttons allowed for manual control of the displayed count. By pressing the increment button, we could increment the count, observing the corresponding numbers on the 7-segment display. Pressing the reset button reset the count to zero, effectively restarting the sequence.
3. **Stability**: The circuit exhibited stability throughout the experiment, with reliable operation of the 7-segment display and pushbuttons. There were no observed instances of erratic behavior or unexpected resets
4. **Educational Value**: We gained practical knowledge of interfacing electronic components with an Arduino board, understanding the roles of segment displays, resistors, and pushbuttons in the circuit. Furthermore, we developed programming skills by uploading and executing Arduino code to control the display behavior.

Overall, the experiment yielded successful results, providing us with valuable hands-on experience and educational insights into Arduino-based digital logic systems and interfacing applications.

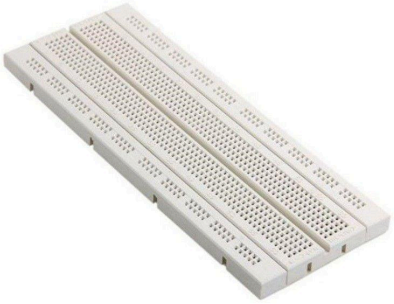
Hardware Discussion

1) 7 Segment Display



7 segment display is used to display numerical numbers, specifically from 0 to 9. The numbers are being displayed according to the software and electrical setup.

2) Breadboard



Breadboard is used to connect components with arduino through wire connection.

3) Arduino Mega 2560

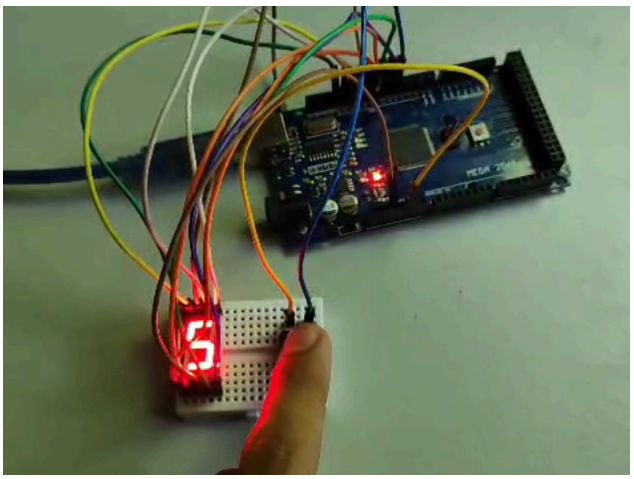
Microcontroller used for our experiment is the Arduino Mega 2560.

4) Pushbutton

Momentary pushbutton is used in our setup to change the displayed number when pushbutton is pressed.

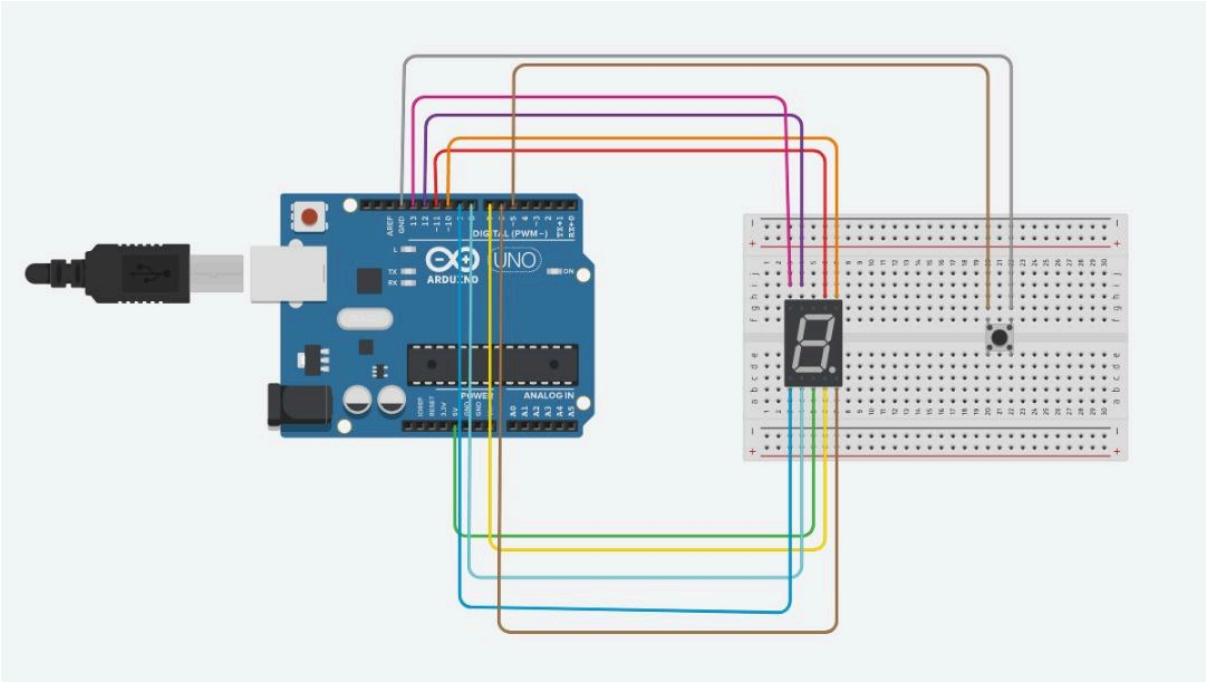
5) Male to male jumper wires

Male to male jumper wires are used to connect all of the components on the breadboard with Arduino Mega.



The above image is when all of the hardware is being connected together. This is the finished hardware, software and electrical setup that can be used to display and change the displayed numerical numbers when pushbutton is pressed.

Electrical Discussion



SEGMENT DISPLAY CONNECTION

SEGMENT A = PWM 11
SEGMENT B = PWM 10
SEGMENT C = PWM 7
SEGMENT D = PWM 8
SEGMENT E = PWM 9
SEGMENT F = PWM 12
SEGMENT G = PWM 13
SEGMENT DP = PWM 6
CATHODE = 5V

BUTTON CONNECTION

CONNECT =
- PWM 5
- GROUND

The diagram provides a detailed overview of the connections required for a Digital Logic Design project utilizing an Arduino Mega microcontroller board. In this project, a segment display is interfaced with the Arduino Mega to demonstrate various sequences programmed into the microcontroller.

Each segment of the display (labeled A, B, C, D, E, F, G, and DP) is connected to a specific PWM (Pulse Width Modulation) pin on the Arduino Mega. This pin allocation enables individual control of each segment's brightness, allowing for the display of different characters or patterns. The connections are as follows: Segment A is connected to PWM pin 11, Segment B to PWM pin 10, Segment C to PWM pin 7, Segment D to PWM pin 8, Segment E to PWM pin 9, Segment F to PWM pin 12, Segment G to PWM pin 13, and Segment DP (decimal point) to PWM pin 6.

Moreover, a button is connected to PWM pin 5 and GND (ground) on the Arduino Mega. This button serves as an input mechanism, allowing users to interact with the project. By pressing the button, users can trigger specific actions or sequences programmed into the microcontroller, enhancing the project's interactivity.

Overall, this setup showcases the versatility of the Arduino Mega in controlling external components, such as segment displays, and demonstrates fundamental concepts of digital logic design and microcontroller programming

Software Discussion

Attached below is the coding of the project. The software part of this mechatronic system project involves programming an Arduino Uno to interface with a 7-segment display and control it manually using buttons. The code is separated into a few parts, namely: Initialization and Pin configurations, displaying numbers, main loop part and handling overflow part.

1. **Initialization and Pin Configuration:**
   The setup() function initializes the pins used for controlling the segments of the 7-segment display (A through H) and the button pin. Each pin is set as an output for the display segments and as an input with an internal pull-up resistor for the button.

2. **Displaying Numbers:**
   There are separate functions (displayZero() to displayNine()) defined to display each digit from 0 to 9 on the 7-segment display. Each function sets the appropriate segments to display the corresponding digit.

3. **Main Loop:**
   In the loop() function, the program continuously checks the state of the button. If the button is pressed (buttonState is LOW), the variable m is incremented. Then, a switch statement is used to determine which digit to display based on the value of m. After displaying a digit, there's a delay of 250 milliseconds.

4. **Handling Overflow:**
   To prevent m from exceeding 9, which represents the maximum digit, an if statement resets m to 0 when it reaches 10.

```
unsigned const int A = 11;
unsigned const int B = 10;
unsigned const int C = 7;
unsigned const int D = 8;
unsigned const int E = 9;
unsigned const int F = 12;
unsigned const int G = 13;
unsigned const int H = 6;
unsigned const int buttonPin = 5;


void setup(void)
{
  pinMode(A, OUTPUT);
  pinMode(B, OUTPUT);
  pinMode(C, OUTPUT);
  pinMode(D, OUTPUT);
  pinMode(E, OUTPUT);
  pinMode(F, OUTPUT);
  pinMode(G, OUTPUT);
  pinMode(H, OUTPUT);
  pinMode(buttonPin , INPUT_PULLUP);
}

//functions to display
//the numbers
void displayZero(void) {
  digitalWrite(A, LOW);
  digitalWrite(B, LOW);
  digitalWrite(C, LOW);
  digitalWrite(D, LOW);
  digitalWrite(E, LOW);
  digitalWrite(F, LOW);
  digitalWrite(G, HIGH);
  digitalWrite(H, LOW);
}

void displayOne(void) {
  digitalWrite(A, HIGH);
  digitalWrite(B, LOW);
  digitalWrite(C, LOW);
  digitalWrite(D, HIGH);
  digitalWrite(E, HIGH);
  digitalWrite(F, HIGH);
  digitalWrite(G, HIGH);
  digitalWrite(H, LOW);
}

void displayTwo(void) {
  digitalWrite(A, LOW);
  digitalWrite(B, LOW);
  digitalWrite(C, HIGH);
  digitalWrite(D, LOW);
  digitalWrite(E, LOW);
  digitalWrite(F, HIGH);
  digitalWrite(G, LOW);
  digitalWrite(H, LOW);
}

void displayThree(void) {
  digitalWrite(A, LOW);
  digitalWrite(B, LOW);
  digitalWrite(C, LOW);
  digitalWrite(D, LOW);
  digitalWrite(E, HIGH);
  digitalWrite(F, HIGH);
  digitalWrite(G, LOW);
  digitalWrite(H, LOW);
```

```
}

void displayFour(void) {
  digitalWrite(A, HIGH);
  digitalWrite(B, LOW);
  digitalWrite(C, LOW);
  digitalWrite(D, HIGH);
  digitalWrite(E, HIGH);
  digitalWrite(F, LOW);
  digitalWrite(G, LOW);
  digitalWrite(H, LOW);
}

void displayFive(void) {
  digitalWrite(A, LOW);
  digitalWrite(B, HIGH);
  digitalWrite(C, LOW);
  digitalWrite(D, LOW);
  digitalWrite(E, HIGH);
  digitalWrite(F, LOW);
  digitalWrite(G, LOW);
  digitalWrite(H, LOW);
}

void displaySix(void) {
  digitalWrite(A, LOW);
  digitalWrite(B, HIGH);
  digitalWrite(C, LOW);
  digitalWrite(D, LOW);
  digitalWrite(E, LOW);
  digitalWrite(F, LOW);
  digitalWrite(G, LOW);
  digitalWrite(H, LOW);
}

void displaySeven(void) {
  digitalWrite(A, LOW);
  digitalWrite(B, LOW);
  digitalWrite(C, LOW);
  digitalWrite(D, HIGH);
  digitalWrite(E, HIGH);
  digitalWrite(F, HIGH);
  digitalWrite(G, HIGH);
  digitalWrite(H, LOW);
}

void displayEight(void) {
  digitalWrite(A, LOW);
  digitalWrite(B, LOW);
  digitalWrite(C, LOW);
  digitalWrite(D, LOW);
  digitalWrite(E, LOW);
  digitalWrite(F, LOW);
  digitalWrite(G, LOW);
  digitalWrite(H, LOW);
}

void displayNine(void) {
  digitalWrite(A, LOW);
  digitalWrite(B, LOW);
  digitalWrite(C, LOW);
  digitalWrite(D, HIGH);
  digitalWrite(E, HIGH);
  digitalWrite(F, LOW);
  digitalWrite(G, LOW);
  digitalWrite(H, LOW);
}

//execute the codes
//by invoking the functions
int m = 0;
void loop(void)
{
 int buttonState = digitalRead(buttonPin);
```

```
if(!buttonState)
{
   m++;
}
//else
 //m = m;

 switch(m)
 {
   case 0 :  displayZero();
             break ;

   case 1 :  displayOne();
             break ;

   case 2 :  displayTwo();
             break ;

   case 3 : displayThree();
             break ;

   case 4 :   displayFour();
             break ;

   case 5 :  displayFive();
             break ;

   case 6 :  displaySix();
             break ;

   case 7 :  displaySeven();
             break ;

   case 8 :  displayEight();
             break ;

   case 9 :  displayNine();
             break ;

  }
  delay (250);
 if(m == 10)
 {
  m = 0;
 }
 //else
  //m = m;

}
```

Conclusion

Interfacing a common cathode 7-segment display with an Arduino Uno board proved to be an educational and insightful experiment in the realm of digital logic systems and electronic circuit interfacing. By following the circuit setup instructions and uploading the provided Arduino code, participants successfully controlled the display to sequentially show numbers from 0 to 9 and reset the count to zero using pushbuttons.

This experiment facilitated a hands-on understanding of Arduino-based interfacing techniques, including connecting segment displays, utilizing pushbuttons for input, and writing Arduino code to control display behavior. Participants gained valuable experience in basic logic gates, electronic circuit interfacing, and utilizing Arduino for simple arithmetic operations.

Moving forward, this foundational knowledge can be extended to more advanced projects involving complex logic systems, additional displays, or integration with other electronic components. Overall, this experiment serves as a fundamental building block for exploring the vast possibilities of Arduino-based digital systems and interfacing applications.