University of Colombo

Faculty of Technology

Department of Instrumentation and Automation Technology

IA 2209 – Microcontroller Laboratory

Design Project

Final Report

Group Number:21

Names & Registration Numbers of Group Members: MLA.Arshad-2020T00584

S.Ashshak-2020T00585

MH.Haneef Ahamed-2020T00579

AL.Nizam-2020T00655

AA.Nafrees - 2020T00647

Submitted date:2023/08/17

Table of Contents.

1. Introduction.

This is our report for the Micro Controller Final Project for IA 2209 – Microcontroller Laboratory. We are 5 Team members in A team. By doing this project we have got a great understating of the ATmega 328p Micro Controller and How To Program It.

2. Objectives (Tasks) of the project.

Task 1: Count Up and Count Down

- Build a circuit that allows counting up and counting down using two push buttons. Each button press should increment or decrement the count value, respectively. The count should range from 0 to 9.

Task 2: Seven-Segment Display

- Connect the seven-segment display to the microcontroller and design a program to display the count value on it. Ensure that the display accurately represents the count from 0 to 9.

Task 3: Binary Value Indication

- Extend the previous circuit to display the binary representation of the count value using 4 LEDs.
- Each LED should represent one bit of the count value, with the least significant bit (LSB) on the rightmost LED. Update the LEDs whenever the count changes.

Task 4: Brightness Control

- Incorporate a separate LED that changes its brightness according to the count value. As the count increases, the LED's brightness should increase proportionally. Implement a suitable method (e.g., pulse width modulation - PWM) to control the LED's brightness based on the count value.

Task 5: Mode Changing

- Add a push button to toggle between different output modes. When this button is pressed, the circuit should cycle through the following modes: displaying the count value on the seven-segment display, indicating the binary value on the 4 LEDs, and adjusting the brightness of the fifth LED. Each press of the button should switch to the next mode.

3. Methodology.

The methodology we have used for this project is

- Initially, We have identified the strong skills in each student from the group Members.
- From Those Identified Skills we have divided the tasks between teammates.
- After finishing individual tasks combine those tasks into a single project.

4. Images of the Simulation.

Binary LED Count Increase And Decrease.

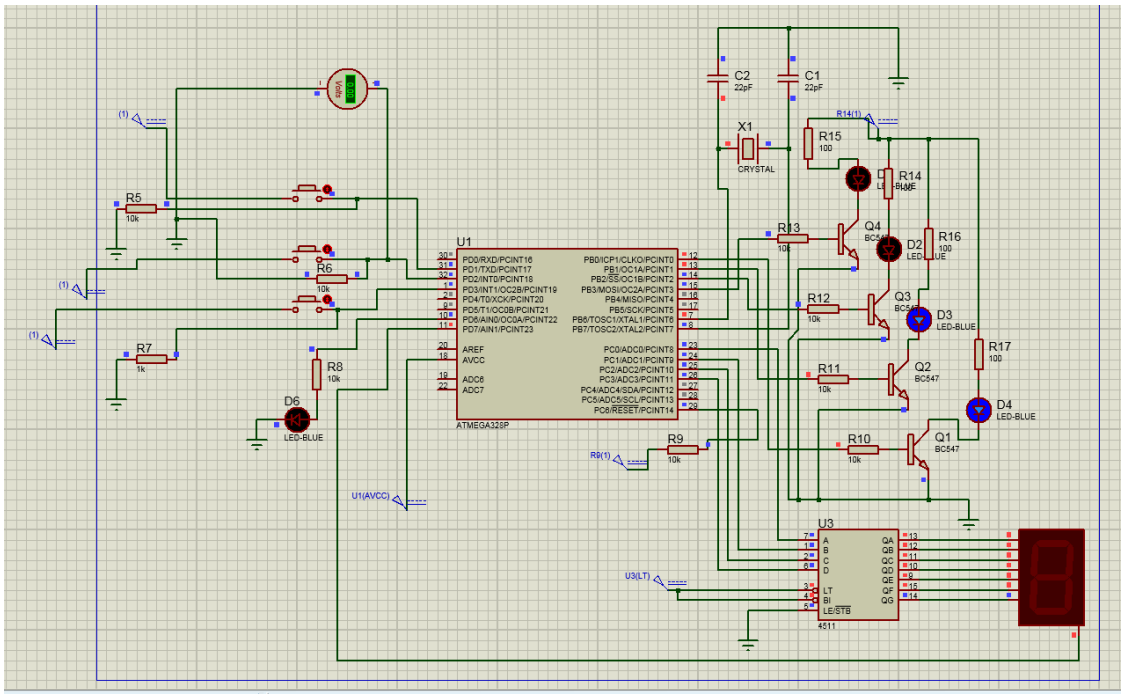


Figure 1 Binary LED Count Simulation

SSD Count Increase and Decrease

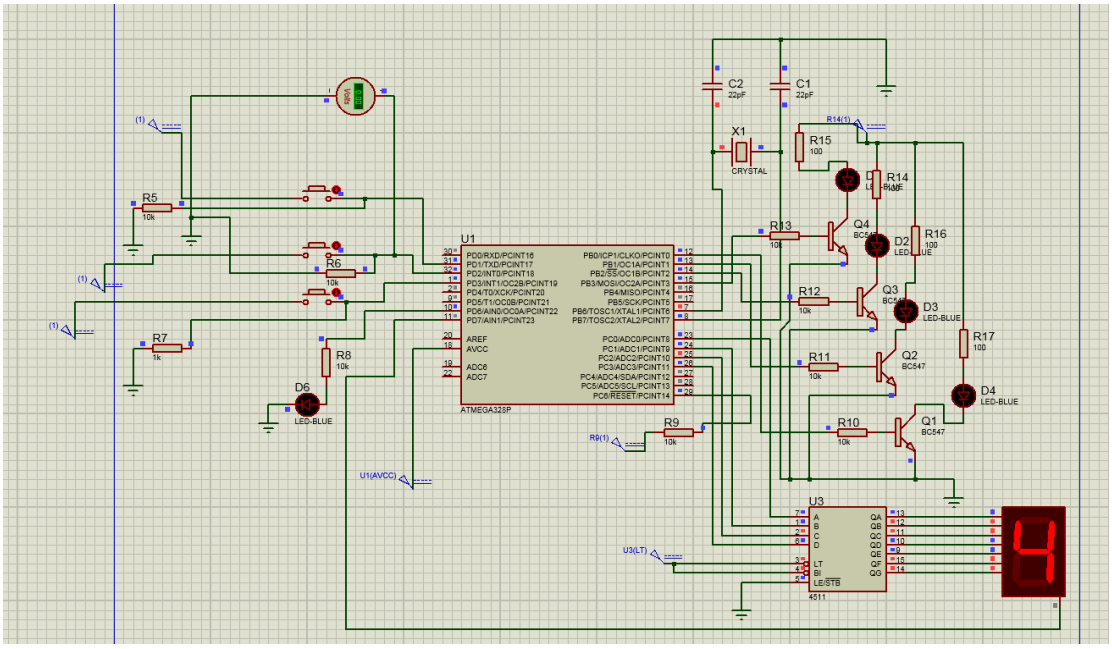


Figure 2 7-Segment Count Simulation

LED Light brightness Increase and Decrease.

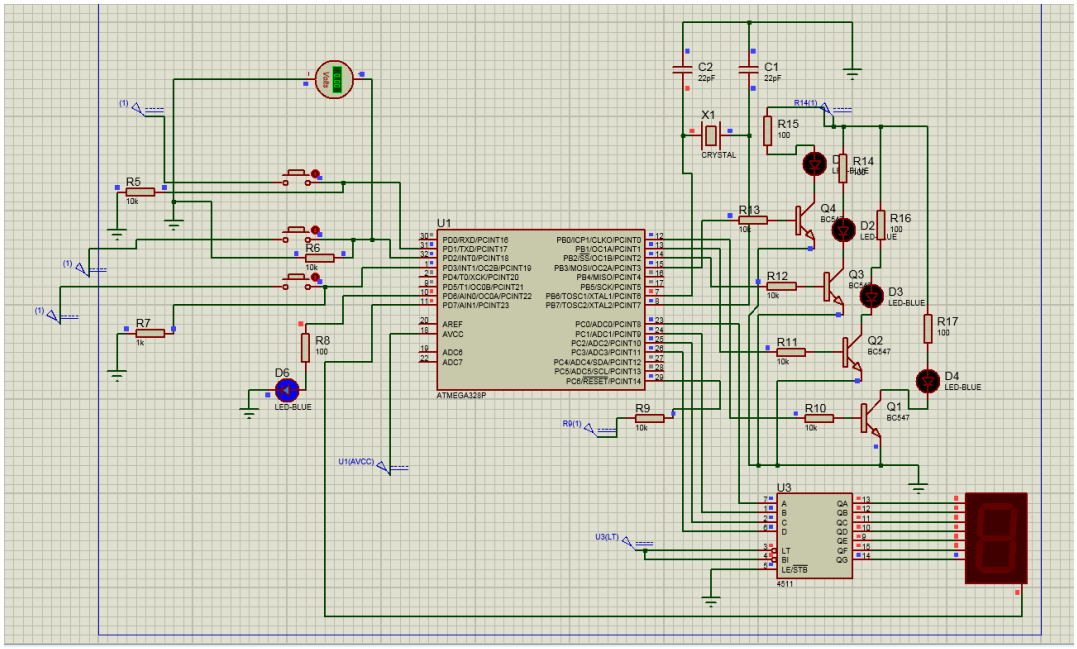


Figure 3 PWM LED Simulation

5. Images of the built circuit.

Binary LED Count Increase And Decrease.

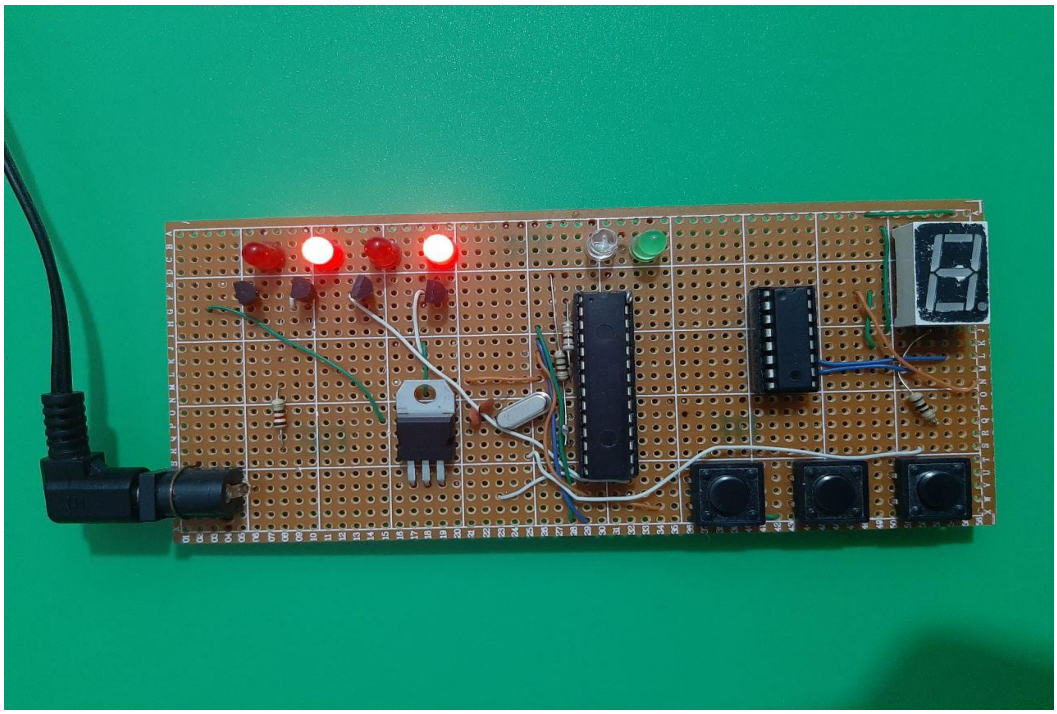


Figure 4 Binary LED Count Circuit

SSD Count Increase and Decrease

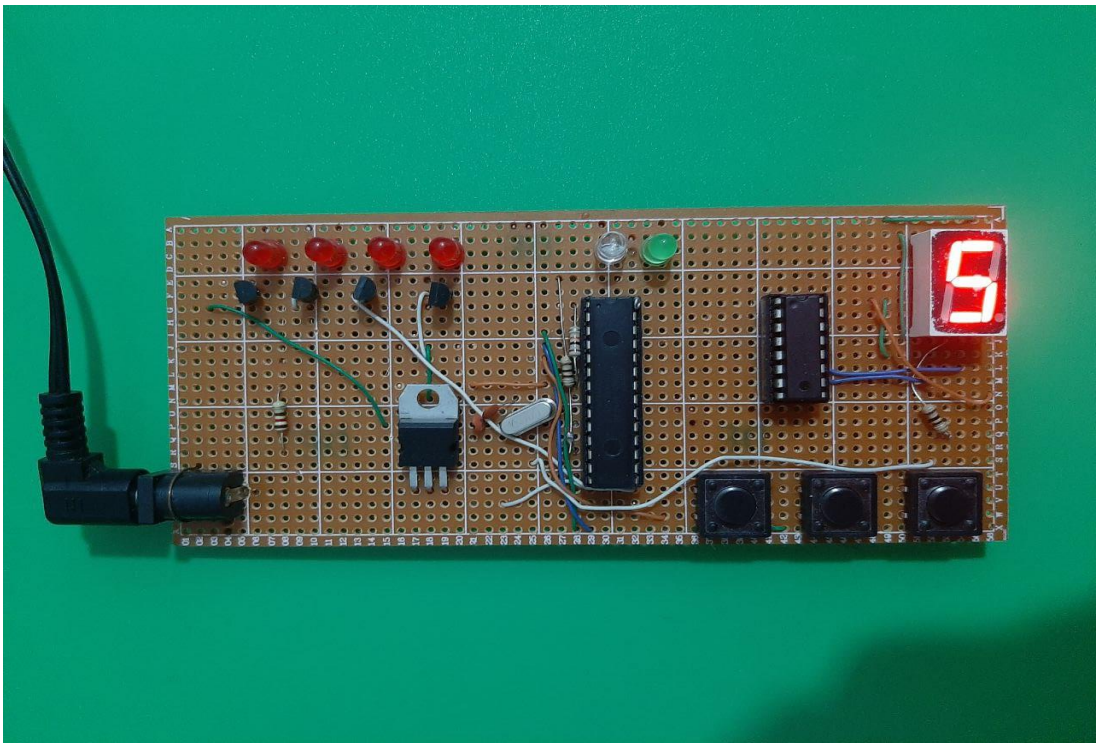


Figure 5 7-Segment Count Circuit

LED Light brightness Increase and Decrease.

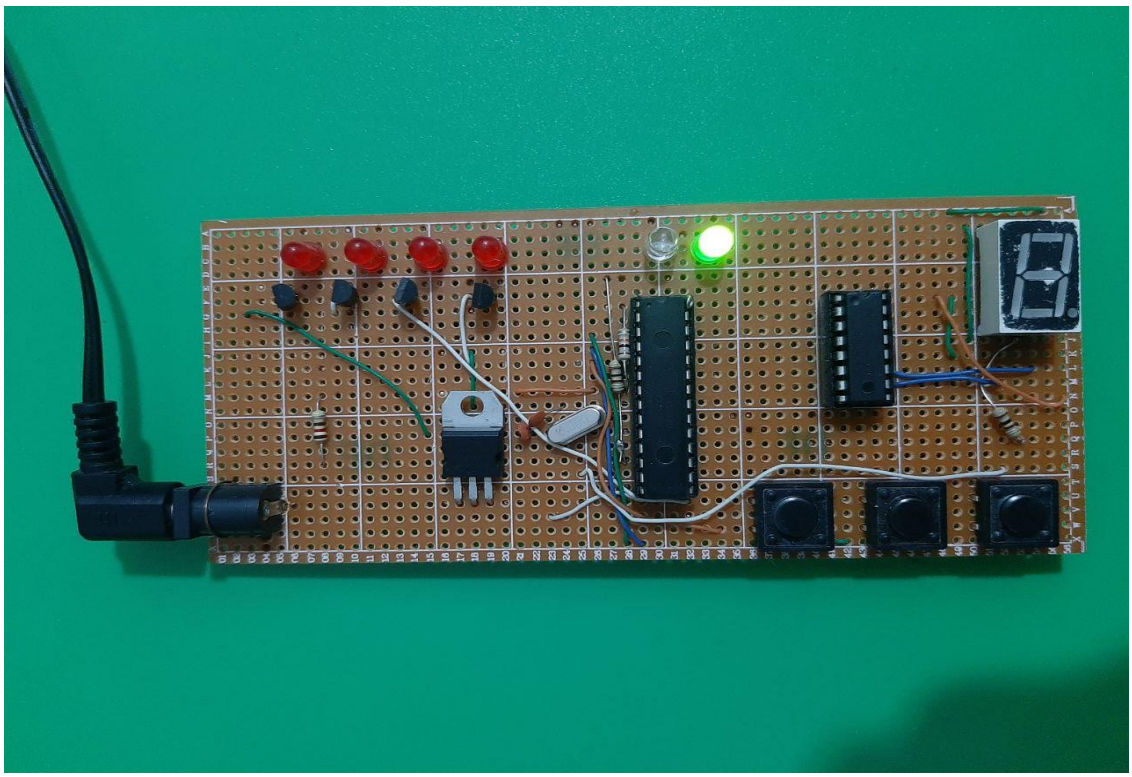
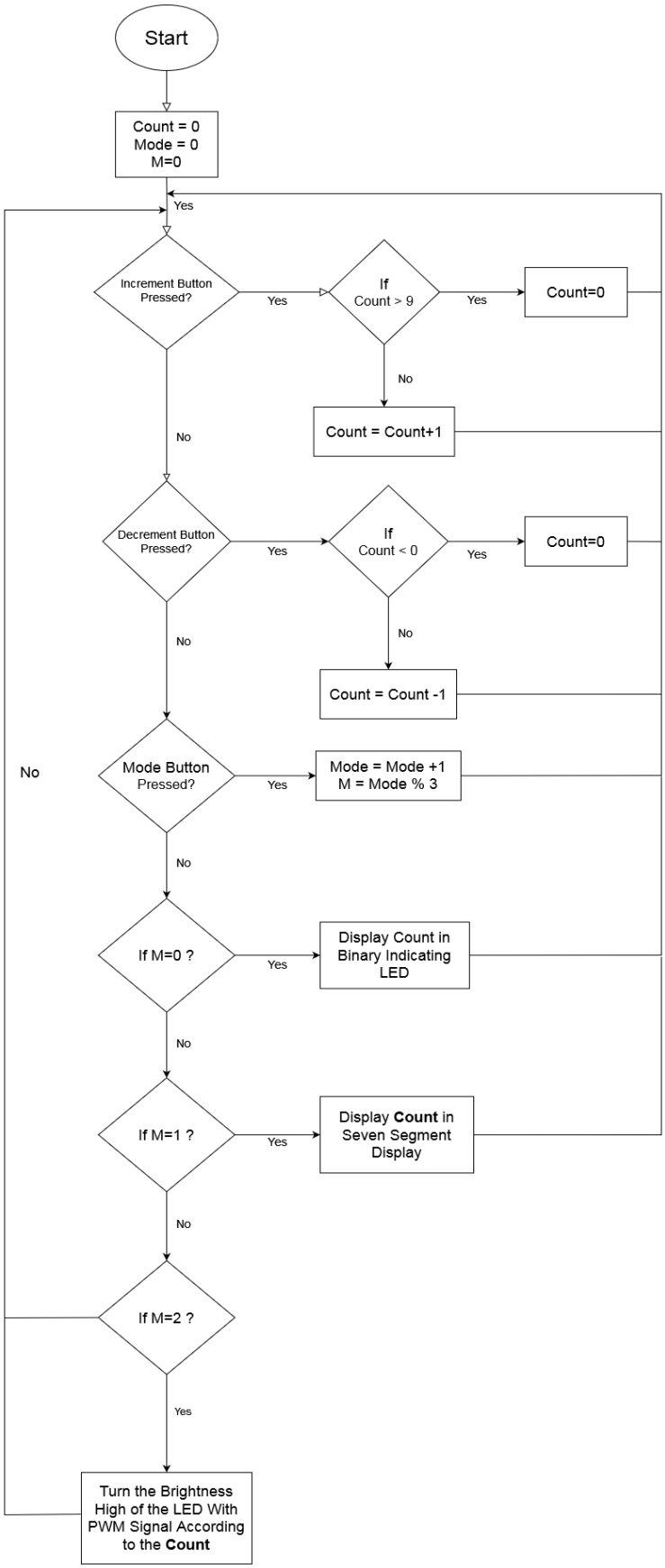


Figure 6 PWM LED Circuit

6. Flow chart of the logic.



7. Code with comments.

```
//Defining the Microcontroller Working Clock Speed As 16MHz
#define F_CPU 16000000UL

//Including the Required Libraries for AVR Programming, Delay, and Interrupts
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>

//Defining required PINS for Outputs(Binary Count LED, 7SD CC PIN, BCD Encode, PWM
LED)
#define LED1_PIN    PB0
#define LED2_PIN    PB1
#define LED3_PIN    PB2
#define LED4_PIN    PB3
#define BCD_PIN_1 PC0
#define BCD_PIN_2 PC1
#define BCD_PIN_3 PC2
#define BCD_PIN_4 PC3
#define SSD_Pin PD7
#define PWM_PIN PD6

//Defining required PINS for Inputs for Increment, Decrement, Mode Changing
#define BUTTON_PIN_1  PD2
#define BUTTON_PIN_2  PD3
#define Mode_Pin PD1

//Set pre-scaler 64 as the value for a variable
#define PWM_PRESCALER 64

//assigning top value as 0 for Timer
float PWM_TOP = 0;

//set brightness level as 10 for PWM LED(0-9)
#define BRIGHTNESS_LEVELS 10

//Set variable for changing modes and counting the Increment and Decrement
volatile uint8_t mode = 0;
volatile uint8_t count = 0;


//function for initializing the PWM for PWM LED
void init_PWM() {
       // Fast PWM, non-inverting mode
       TCCR0A |= (1 << COM0A1) | (1 << WGM01) | (1 << WGM00);

       // Pre scaler: 64 // Set initial PWM duty cycle to 0
       TCCR0B |= (1 << CS01) | (1 << CS00);
}

//Function for disabling the PWM Function
void disable_PWM() {
       // Clear COM0A1 and WGM00 bits in TCCR0A to disable PWM output
       TCCR0A &= ~((1 << COM0A1) | (1 << WGM00));

       // Clear CS01 and CS00 bits in TCCR0B to stop the timer
       TCCR0B &= ~((1 << CS01) | (1 << CS00));
}
```

```
//Function for Setup the Output and Input pins in Data Direction Register and enable
Internal Pull-Up for Push Buttons
void setup()
{
	//enabling the output mode in DDRB for Binary Counting LEDS
	DDRB |= (1 << LED1_PIN) | (1 << LED2_PIN) | (1 << LED3_PIN) | (1 << LED4_PIN);

	//enabling the output mode in DDRC for BCD Encoder for 7SDs
	DDRC |= (1 << BCD_PIN_1) | (1 << BCD_PIN_2) | (1 << BCD_PIN_3) | (1 <<
BCD_PIN_4);

	//enabling the output mode in DDRD for Common Cathode Pin of 7SDs
	DDRD |= (1 << SSD_Pin);

	//enabling the output mode in DDRD for PWM brightness adjustment LED
	DDRD |= (1 << PWM_PIN);

	//enabling the Input mode in DDRD for Increment, Decrement, and Mode changing
Push Buttons
	DDRD &= ~(1 << BUTTON_PIN_1);
	DDRD &= ~(1 << BUTTON_PIN_2);
	DDRD &= ~(1 << Mode_Pin);

	//enabling the internal pull up resistor
	PORTD |= (1 << BUTTON_PIN_1);
	PORTD |= (1 << BUTTON_PIN_2);
	PORTD |= (1 << Mode_Pin);
}

uint8_t isSwitchPressed() {
	// Check the status of the SWITCH_PIN (bit 1 in PIND register)
	// 0 means the switch is pressed because of the pull-up resistor
	// 1 means the switch is not pressed (external pull-up resistor required)
	return (PIND & (1 << Mode_Pin));
}
uint8_t isSwitchincrease() {
	// Check the status of the SWITCH_PIN (bit 1 in PIND register)
	// 0 means the switch is pressed because of the pull-up resistor
	// 1 means the switch is not pressed (external pull-up resistor required)
	return (PIND & (1 << BUTTON_PIN_1));
}
uint8_t isSwitchdecrease() {
	// Check the status of the SWITCH_PIN (bit 1 in PIND register)
	// 0 means the switch is pressed because of the pull-up resistor
	// 1 means the switch is not pressed (external pull-up resistor required)
	return (PIND & (1 << BUTTON_PIN_2));
}
void clearBitInPortD(uint8_t bitPosition) {
	// Check if bitPosition is valid (0-7)
	if (bitPosition >= 0 && bitPosition <= 7) {
		// Create a mask with all bits set to 1 except for the bit at the given
position
		//if it 7SD common anode set to high not inverse it
		uint8_t mask = ~(1 << bitPosition);

		// Perform bitwise AND with the mask to clear the bit at the given
position
		PORTD &= mask;
	}
}
int main(void)
{
```

```
	//Recalling the Setup Function for intializing the Ouput and Input
Configurations in the Setup
	setup();

	//Main While Loop
	while (1)
	{
		if ((isSwitchPressed())) {
			// If Mode Switch is pressed, Mode will increment by 1
			mode ++;

			// Small delay to avoid rapid toggling due to switch bouncing
			_delay_ms(150);
		}
		if((isSwitchincrease()))
		{
		// If Count Increment Switch is pressed, Mode will Increased by 1
			count++;

			// Small delay to avoid rapid toggling due to switch bouncing
			_delay_ms(50);

			//If Count higher than count > 9 Count resets to 0
			if (count > 9){
				count=0;
			}
		}
		if((isSwitchdecrease()))
		{
		// If Count Decrement Switch is pressed, Mode will Decreased by 1
			count--;

			// Small delay to avoid rapid toggling due to switch bouncing
			_delay_ms(50);

			//If Count higher than count < -1 Count resets to 9
			if (count == -1){
				count=9;
			}

			//If Count higher than count > 9 Count resets to 0
			if (count > 9){
				count=0;
			}
		}

		//check the Mode Selection Input
		// If Mode == 0 Binary LED Counting will be initialized other modes will
be turned off
		if ((mode%3)==0){

			//Disabling PWM Function
			disable_PWM();

			//Display Output Count in Binary Counting LED
			PORTB = (count & 0x0F);

			//Disable 7-segment Display BCD Encoder
			PORTC = (0x00);

			//Disable 7-segment Display Common Cathode Pin
			PORTD|=(1<<SSD_Pin);
```

```
                        //Disabling PWM LED from PORTD Register
                        PORTD &= ~(1 << PWM_PIN);
                }

                if ((mode%3)==1){
                        // If Mode == 1 7-Segment Display Counting will be initialized
other modes will be turned off

                        //Disabling PWM Function
                        disable_PWM();

                        //Enable Common Cathode PIN of 7-Segment PIN
                        clearBitInPortD(7);

                        //Display Output Count in 7-Segment Display
                        PORTC = (count & 0xFF);

                        //Disable Binary Counting LED
                        PORTB = (0x00);

                        //Disabling PWM LED from PORTD Register
                        PORTD &= ~(1 << PWM_PIN);
                }

                // If Mode == 2 PWM LED Brightness Adjusting mode will be initialized
other modes will be turned off
                if ((mode%3)==2){

                        //Condition for Turned OFF the PWM LED Completely If Count=0
                        if (count==0){

                                //Disabling PWM Function
                                disable_PWM();

                                //Disable 7-segment Display Common Cathode Pin
                                PORTD|=(1<<SSD_Pin);

                                //Disable Binary Counting LED
                                PORTB = (0x00);

                                //Disable 7-segment Display BCD Encoder
                                PORTC = (0x00);

                                //Disabling PWM LED from PORTD Register
                                PORTD &= ~(1 << PWM_PIN);
                        }

                        //Else Condition for Turned on PWM LED According to The Count If
Count>0
                        else{
                                //Enabling PWM Function
                                init_PWM();

                                //Disable 7-segment Display Common Cathode Pin
                                PORTD|=(1<<SSD_Pin);

                                //Display the PWM Brightness according to The Count In PWM
LED
                                OCR0A = (255/BRIGHTNESS_LEVELS)*count;

                                //Disable Binary Counting LED
```

```
                        PORTB = (0x00);

                        //Disable 7-segment Display BCD Encoder
                        PORTC = (0x00);
                    }
                }
            }
}
```

8. Discussion

Problem 1:
Initially, Our Code worked on Simule-IDE but while we are try to develop the Physical Circuit the Circuit Doesn't Works

Solution:
We have tried the same circuit in Proteus. The Circuit Didn't Work. So We have altered the Circuit in the Pull-Up Resister for Push Buttons and Enabled the Internal Pull-Up for each push Button. Then the circuit also worked in the physical one as well.

Problem 2
Delays not worked properly.

Solution:
Connect a 16MHz external Crystel Oscillator to define Accurate Delays.

Problem 3:
The Circuit Debounces while pressing the Increment and Decrement Buttons.

Solution:
While Troubleshooting this problem, We have found that we have written code for these two buttons using external interrupts. In the external interrupts, delays will be not worked So the debouncing delays didn't work in the Interrupt. So, As a solution, we have altered the code without External Interrupts for the buttons.

9. Conclusion

As beginners in embedded system development, we have understood the basics to medium level of AVR Micro Controller Programming. It is a wonderful practical learning experience. From this Project, We have gained so much of skills more than technical skills. They are

1. Team Working.
2. PCB Designing.
3. Soldering.
4. AVR Architecture.
5. Mico Controler Programming.
6. Report Writing.
7. Demonstrating the Project.
8. Sharing Works