

MUP[EE2016]_LAB_EXPERIMENT_03_AVR_Peripherals
MIRUDHULA
EE23B046

OBJECTIVES:

This experiment introduces assembly programming and interaction with peripherals in Atmel Atmega8 microcontroller.

1. Write an AVR assembly program to blink an LED with pulse width (during ON) of 0.5 sec and pulse repetition rate of 1 cycle /sec
2. Write an AVR assembly program to control an LED using a push button switch (as long as button is pushed, LED should be 'ON').
3. 4-bit addition of two unsigned nibbles from an 8-bit DIP input switch (set by TAs) and display the result obtained in LEDs.

CODE USED:

PROBLEM 1:

```
.equ F_CPU = 1000000
```

```
.INCLUDE "M8DEF.INC"
```

```
LDI R16, HIGH(RAMEND)
OUT SPH, R16
LDI R16, LOW(RAMEND)
OUT SPL, R16
```

RESET:

```
LDI R16, 0x01
OUT DDRB, R16
```

LOOP:

```
; Turn on LED
LDI R16, 0x01
OUT PORTB, R16
```

```
RCALL DELAY_500MS
```

```
LDI R16, 0x00
```

```
OUT    PORTB, R16
```

```
RCALL  DELAY_500MS
```

```
RJMP   LOOP
```

```
DELAY_500MS:
```

```
    LDI    R18, 3
```

```
DELAY_LOOP_1:
```

```
    LDI    R19, 195
```

```
DELAY_LOOP_2:
```

```
    LDI    R20, 250
```

```
DELAY_LOOP_3:
```

```
    DEC    R20
```

```
    BRNE   DELAY_LOOP_3
```

```
    DEC    R19
```

```
    BRNE   DELAY_LOOP_2
```

```
    DEC    R18
```

```
    BRNE   DELAY_LOOP_1
```

```
RET
```

PROBLEM_2:

```
.equ LED_PIN = 0
```

```
.equ BUTTON_PIN = 1
```

```
.equ BUTTON_PRESSED = 0
```

```
.include "m8def.inc"
```

```
.cseg
```

```
.org 0x00
```

```
rjmp RESET
```

```
RESET:
```

```
    ldi r16, (1<< LED_PIN)
```

```
    out DDRB, r16
```

```
    ldi r16, (0<<BUTTON_PIN)
```

out DDRD, r16

**ldi r16, (1<<BUTTON_PIN)
out PORTD, r16**

MAIN_LOOP:

in r16, PIND

**sbrs r16, BUTTON_PIN
rjmp TURN_ON_LED**

**cbi PORTB, LED_PIN
rjmp MAIN_LOOP**

TURN_ON_LED:

**sbi PORTB, LED_PIN
rjmp MAIN_LOOP**

PROBLEM 3:

#include "m8def.inc"

START:

**LDI R16, 0x00;
OUT DDRD, R16;**

**LDI R16, 0xFF;
OUT DDRC, R16;**

ADDITION:

**IN R21, PIND;
MOV R20, R21;
ANDI R20, 0xF0;
SWAP R20;
ANDI R21, 0x0F;
ADD R20, R21;**

END:

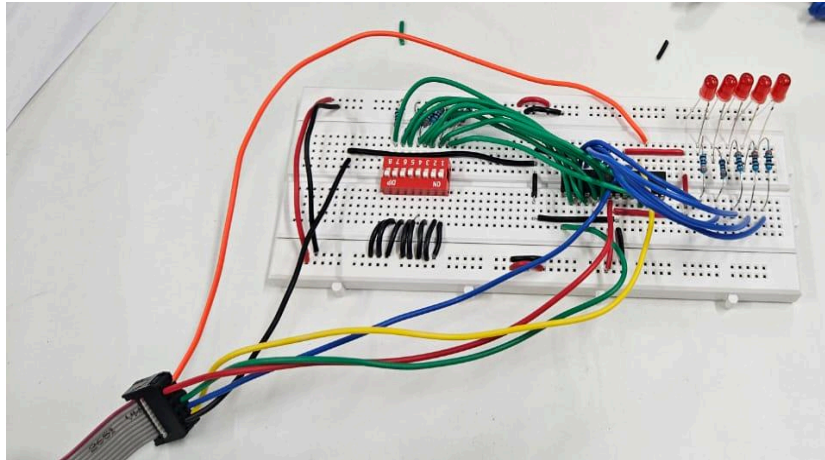
**OUT PORTC, R20;
NOP;**

PROCEDURE FOLLOWED:

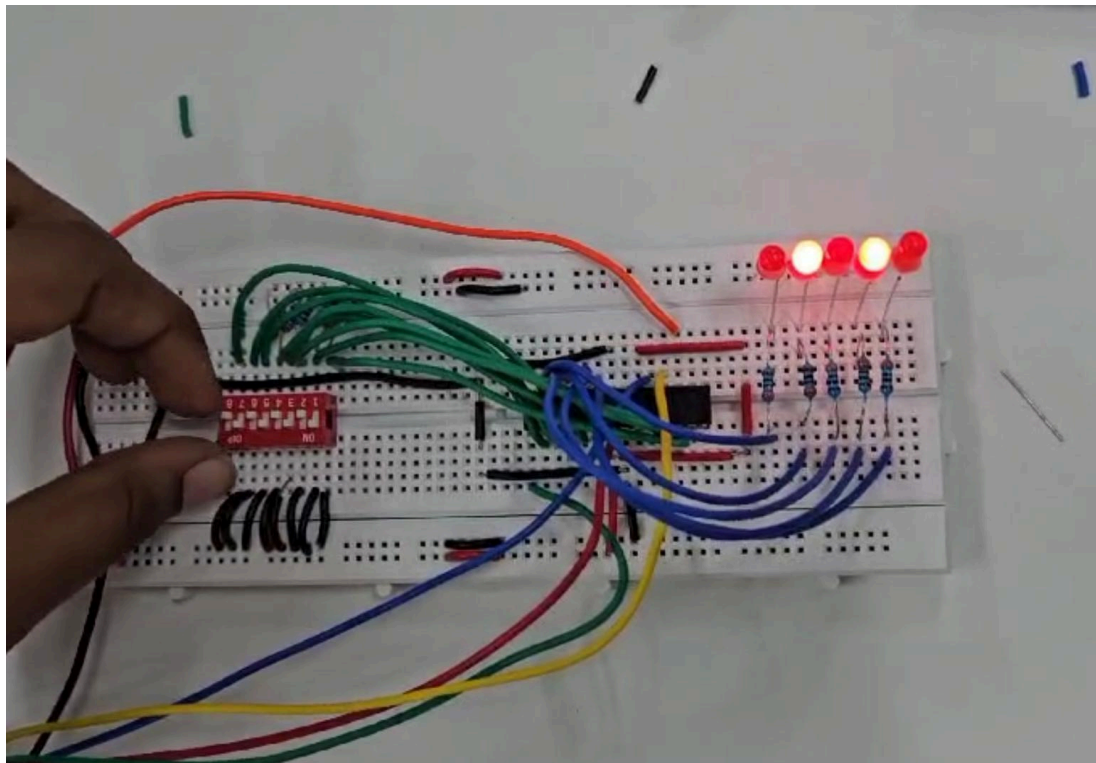
- We first wrote the code for the problem then built the solution and debugged it. Later after correcting the errors we uploaded the hex file in the burn-O-mat software.
- Then we connected the circuit on a breadboard, later we connected the microcontroller to in-system programmer.
- Then we flash into it, and the program is implemented in it.

OBSERVATION:

Problem_3

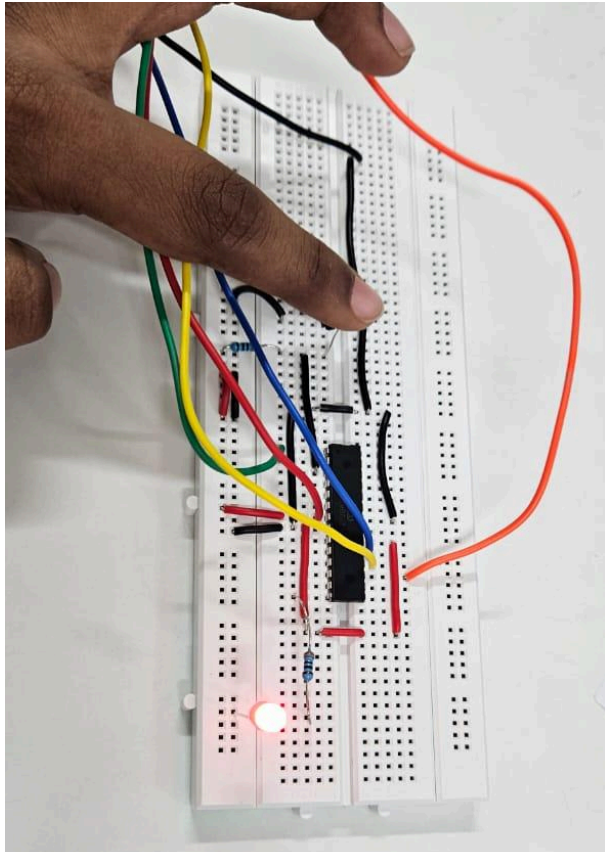


(circuital design)

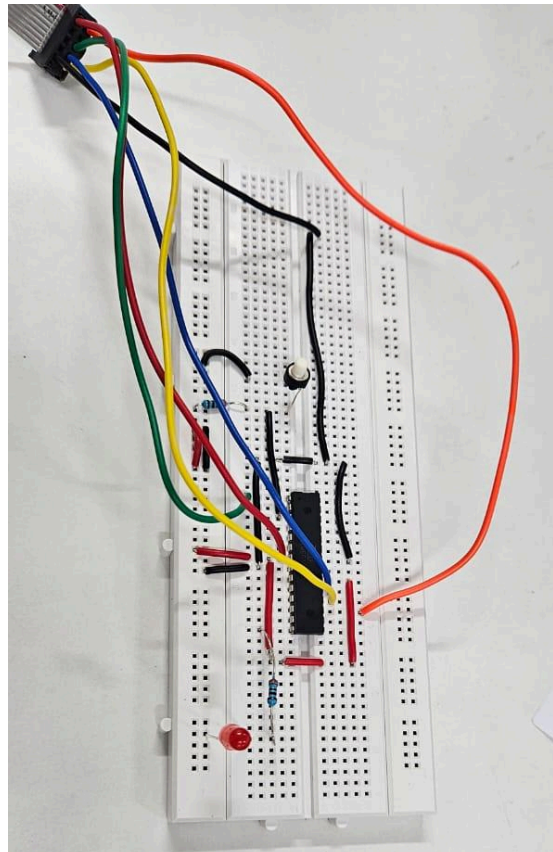


For the inputs 0101 (5) and 0010 (2) the output is 0111(7)

Problem_2

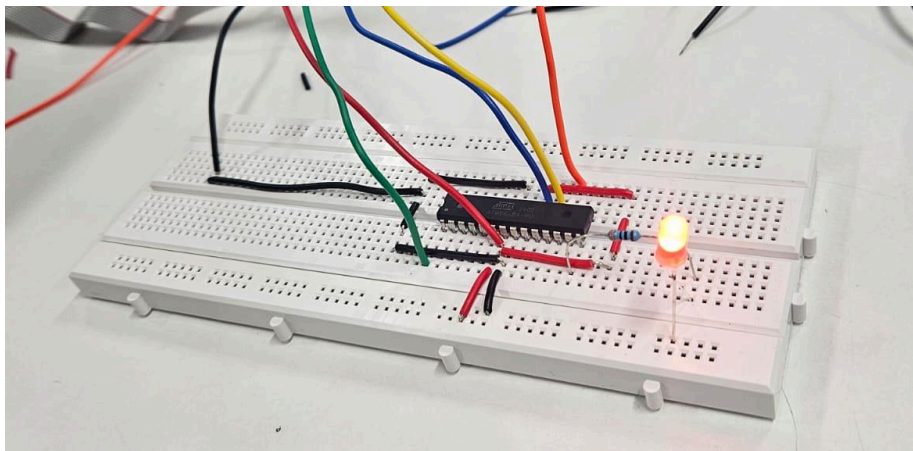


(led is glows when the push button is pressed)

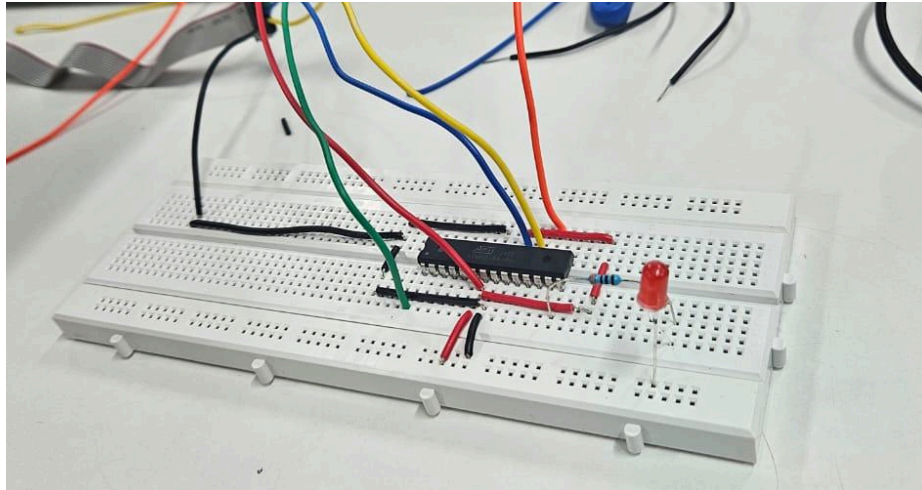


(led is not glowing when push button is not pressed)

Problem_1



(led in ON for 0.5 sec)



(led in OFF for 0.5 sec)

MY PART:

I was responsible for the LED on and off using push button (problem-2) and wiring. And debugging problems caused during writing the program in Atmega8.