MUP[EE2016]_LAB_EXPERIMENT_04_AVR_INTERRUPTS
MIRUDHULA
EE23B046

OBJECTIVES:

This experiment introduces assembly programming and To implement interrupts and DIP switches control in the Atmel ATmega microprocessor.

1.Generate an external (logical) hardware interrupt using an emulation of a pushbutton switch.

2. Write An ISR(Interrupt Service Routine) to switch an LED for a few seconds(10 secs) and then switch OFF. (The lighting of the LED could be verified by monitoring the signal to switch it ON).

- First problem is to use int1 and make the LED blink for 1 sec
- Second problem is to use int0 to redo the same in the demo program (duly filled in). Once the switch is pressed the LED should blink 10 times (ON (or OFF)- 1 sec, duty cycle could be 50 %).

CODE USED:

PROBLEM_1:

```
.org 0
rjmp reset

.org 0x0004
rjmp int1_ISR

.org 0x0100

reset:
    LDI R16,0x70
        OUT SPL,R16
        LDI R16,0x00
        OUT SPH,R16

        LDI R16,0x01
        OUT DDRB,R16

        LDI R16,0x00
        OUT DDRD,R16

        IN R16,MCUCR
```

```
        ORI R16,(1<<ISC11) | (1<<ISC10)
        OUT MCUCR,R16

        IN R16,GICR
        ORI R16,(1<<INT1)
        OUT GICR,R16

        LDI R16,0x00
        OUT PORTB,R16

        SEI

ind_loop:
    rjmp ind_loop

int1_ISR:
    IN R16,SREG
        PUSH R16

        LDI R16,0x0A
        MOV R0,R16

c1:  LDI R16,0xFF
        OUT PORTB,R16

        LDI R16,0xFF
a1:  LDI R17,0xFF
a2:  LDI R18,0x04
a3:  DEC R18
        BRNE a3
        DEC R17
        BRNE a2
        DEC R16
        BRNE a1

        LDI R16,0x00
        OUT PORTB,R16

        LDI R16,0xFF
b1:  LDI R17,0xFF
b2:  LDI R18,0x04
b3:  DEC R18
        BRNE b3
        DEC R17
```

```
        BRNE b2
        DEC R16
        BRNE b1

        DEC R0
        BRNE c1

        POP R16
        RETI
```

PROBLEM_2:

```
.org 0
rjmp reset

.org 0x0002
rjmp int0_ISR

.org 0x0100

reset:
    LDI R16,0x70
        OUT SPL,R16
        LDI R16,0x00
        OUT SPH,R16

        LDI R16,0x01
        OUT DDRB,R16

        LDI R16,0x00
        OUT DDRD,R16

        IN R16,MCUCR
        ORI R16,(1<<ISC01) | (1<<ISC00)
        OUT MCUCR,R16

        IN R16,GICR
        ORI R16,(1<<INT0)
        OUT GICR,R16

        LDI R16,0x00
        OUT PORTB,R16
```

```
        SEI

ind_loop:
    rjmp ind_loop

int0_ISR:
    IN R16,SREG
        PUSH R16

        LDI R16,0x0A
        MOV R0,R16

c1:  LDI R16,0xFF
        OUT PORTB,R16

        LDI R16,0xFF
a1:  LDI R17,0xFF
a2:  LDI R18,0x04
a3:  DEC R18
        BRNE a3
        DEC R17
        BRNE a2
        DEC R16
        BRNE a1

        LDI R16,0x00
        OUT PORTB,R16

        LDI R16,0xFF
b1:  LDI R17,0xFF
b2:  LDI R18,0x04
b3:  DEC R18
        BRNE b3
        DEC R17
        BRNE b2
        DEC R16
        BRNE b1

        DEC R0
        BRNE c1

        POP R16
        RETI
```

PROCEDURE FOLLOWED:

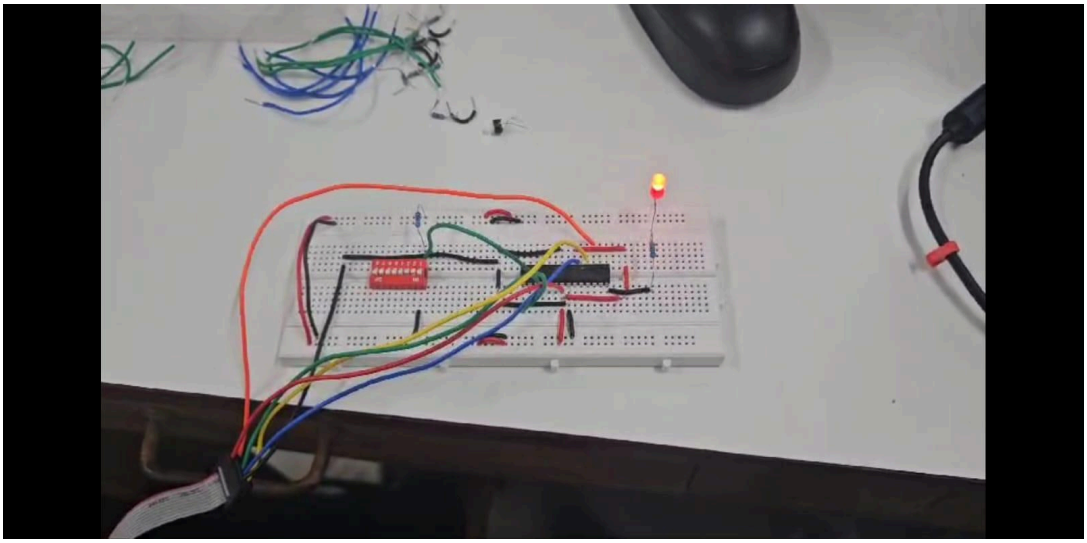
- We first wrote the code for the problem for programming the interrupt and writing the ISR
- Use the Microchip/Atmel Studio to simulate the program. Test the interrupt service routine and the LED control in the simulation environment.
- Use the USBASP programmer and AVR Burn-o-mat software to upload your assembly program to the Atmega8 microcontroller.
- Setup the Hardware: then we made all necessary connections on the breadboard.
- Then we flash into it, and the program is implemented in it.

OBSERVATION:

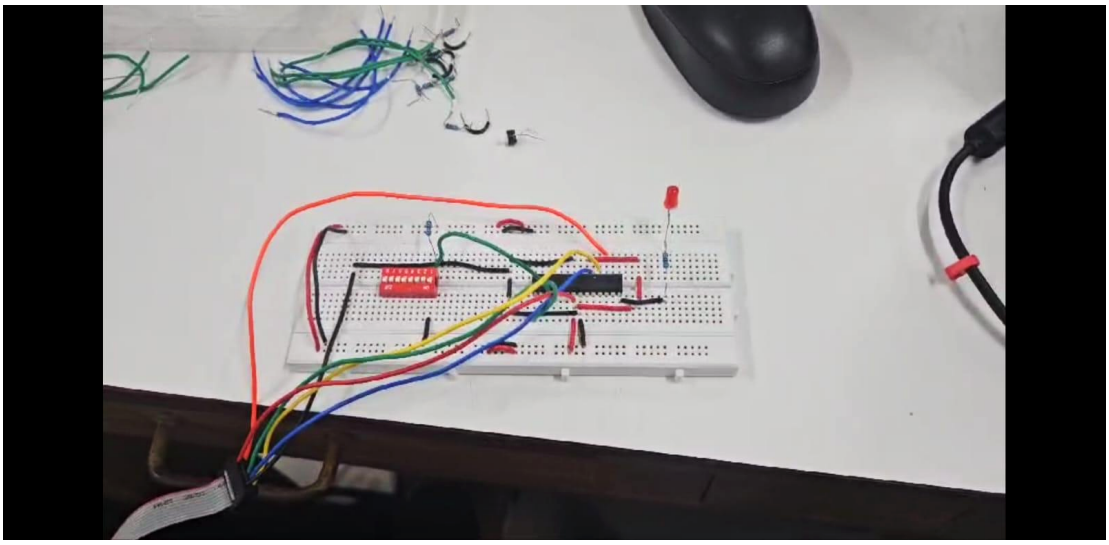
Problem_1

(circuit design using int1)

(LED is ON for 1 sec,then OFF for 1 sec when the push button is pressed)



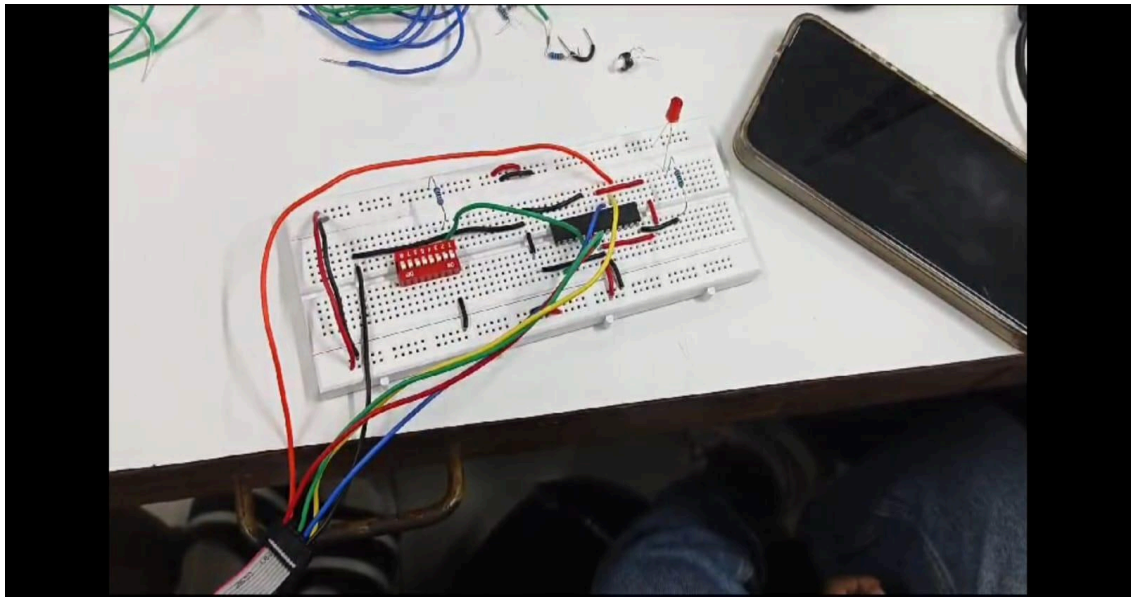
LED-OFF



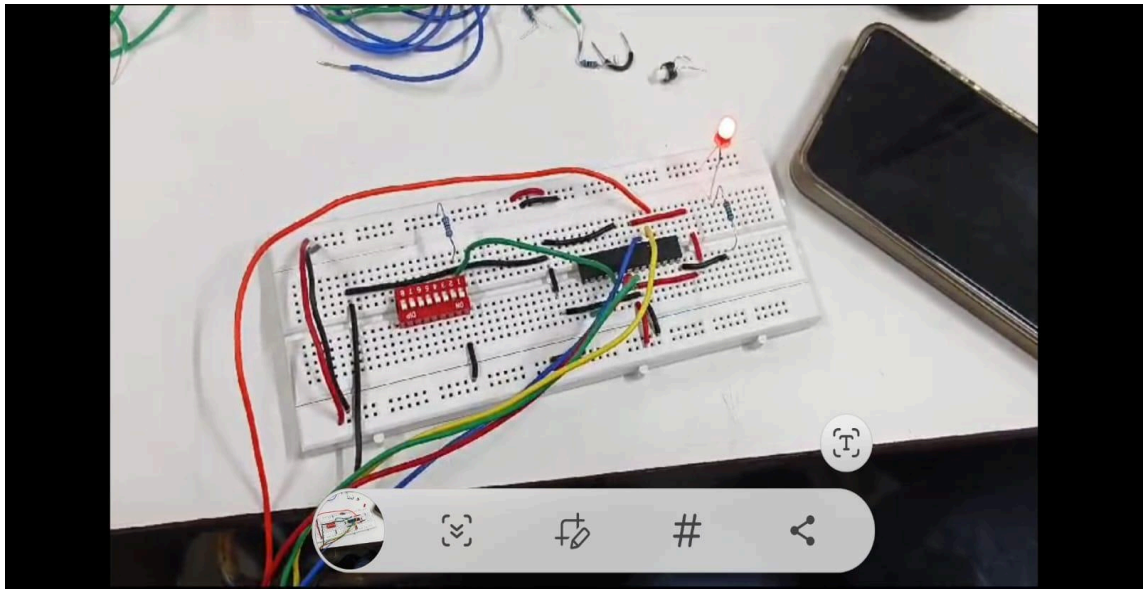
Problem_2

(circuit design using int0)

(LED is ON for 1 sec,then OFF for 1 sec when the push button is pressed)



LED-ON



MY PART:

I was responsible for the LED on and off using interrupt 0 (problem-2) and wiring.
And debugging problems caused during writing the program in Atmega8.