Experiment 2

Interrupts and Timers in Atmel AVR Atmega

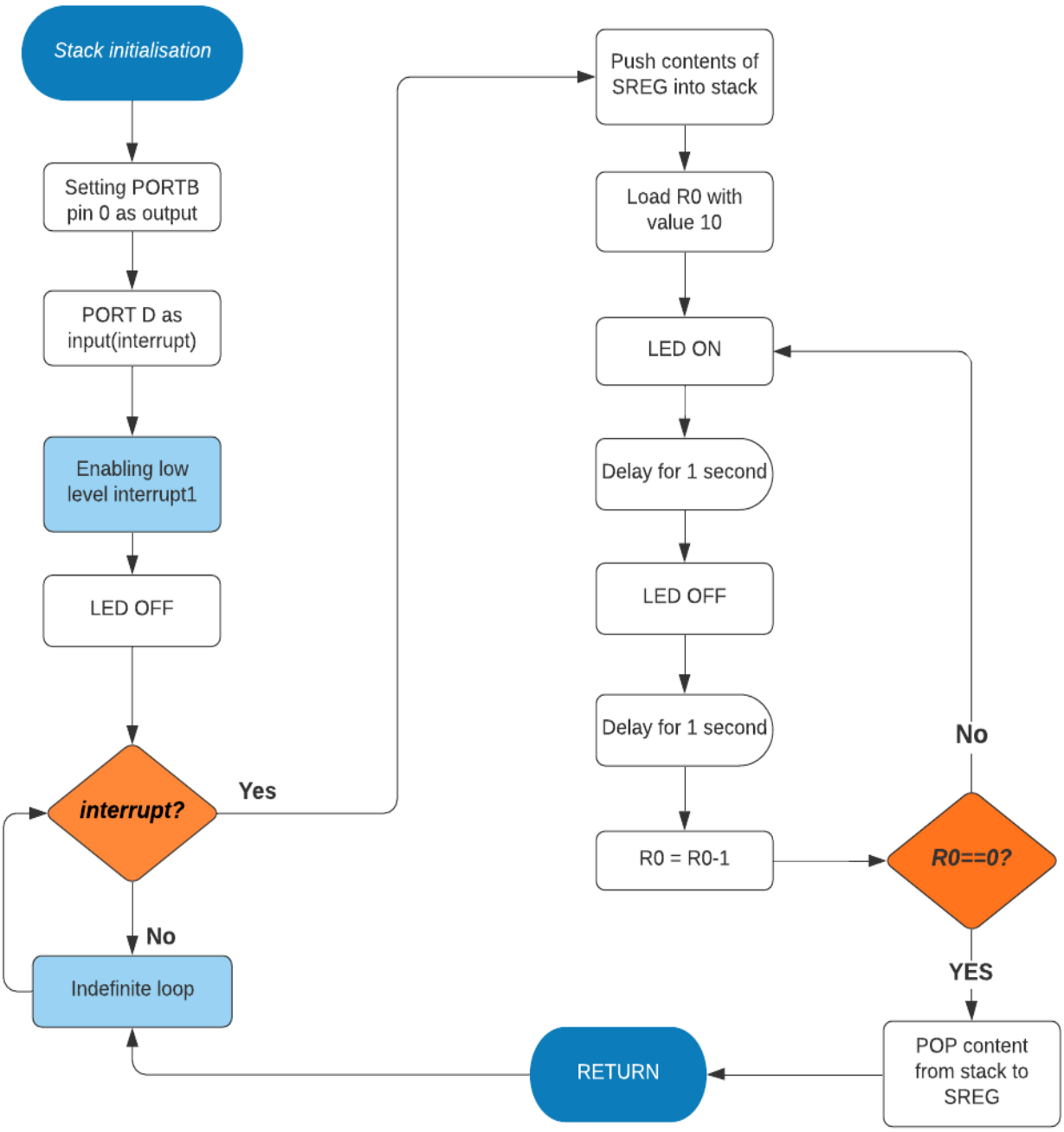
Target :

- Implementation of interrupts and timers in Atmel Atmega microprocessor using assembly and C-interface
- Generating an external hardwire interrupt in emulation software using push button
- Writing an Interrupt Service Routine to switch on and switch off LED for few seconds
- Implementation of the same in C-interfacing

Questions:

1. Fill in the blanks in the assembly code.
2. Use int0 to redo the same in the demo program (duely filled in). Once the switch is pressed the LED should blink 10 times (ON (or OFF) - 1 sec, duty cycle could be 50 %). Demonstrate both the cases.
3. Rewrite the program in 'C' (int1). Rewrite the C program for int0.
4. Demonstrate both the cases (of assembly and C).

Solutions:(a)


Stack initialisation
Setting PORTB pin 0 as output
PORT D as input(interrupt)
Enabling low level interrupt1
LED OFF
interrupt?
Yes
No
Indefinite loop
Push contents of SREG into stack
Load R0 with value 10
LED ON
Delay for 1 second
LED OFF
Delay for 1 second
R0 = R0-1
R0==0?
No
YES
POP content from stack to SREG
RETURN

(b) Assembly code :

Int1

```
.org 0x0000
rjmp reset

.org 0x0002
rjmp int1_ISR

.org 0x0100

Reset:
      ;Loading stack pointer address
      LDI R16,0x70
      OUT SPL,R16
      LDI R16,0x00
      OUT SPH,R16

       LDI R16, 0x01     ; Interface port B pin0 to be output
       OUT DDRB, R16     ; so to view LED blinking

       LDI R16,0x00 ; Interface port D to be input ie,interrupt
       OUT DDRD,R16

       LDI R16, 0x00;Set MCUCR register to enable low level interrupt
       OUT MCUCR,R16

       LDI R16, 1<<INT1;Set GICR register to enable interrupt 1
       OUT GICR,R16

       LDI R16,0x00  ;setting PORTB as 0, LED off initially
       OUT PORTB,R16
```

```
       SEI
ind_loop:rjmp ind_loop

int1_ISR:IN R16,SREG
           PUSH R16 ;Pushing the contents of SREG into Stack

           LDI R16,0x0A;setting R0 with 10 to blink LED 10 times
           MOV R0,R16

     c1:   LDI R16,0x01 ;LED On
           OUT PORTB,R16
     ;Delay program to make a delay of 1 sec
           LDI R16,0x21
     a1:   LDI R17,0x64
     a2:   LDI R18,0x64
     a3:   DEC R18
           BRNE a3
           DEC R17
           BRNE a2
           DEC R16
           BRNE a1
           ; 1 sec over
           LDI R16,0x00 ;LED off
           OUT PORTB,R16
     ;Delay program for 1 sec
           LDI R16,0x21
     b1:   LDI R17,0x64
     b2:   LDI R18,0x64
     b3:   DEC R18
           BRNE b3
           DEC R17
           BRNE b2
           DEC R16
           BRNE b1
```

```
            DEC R0
            BRNE c1 ; LED blinks 10 times
            POP R16
            OUT SREG,R16

            RETI
```

IntO

```
.org 0x0000
rjmp reset

.org 0x0001
rjmp int0_ISR

.org 0x0100

Reset:
      ;Loading stack pointer address
      LDI R16,0x70
      OUT SPL,R16
      LDI R16,0x00
      OUT SPH,R16

       LDI R16, 0x01     ;Interface port B pin0 to be output
       OUT DDRB, R16     ;so to view LED blinking

       LDI R16,0x00  ;Interface port D to be input ie,interrupt
       OUT DDRD,R16

       LDI R16, 0x00;Set MCUCR register to enable low level interrupt
       OUT MCUCR,R16

       LDI R16, 1<<INT0;Set GICR register to enable interrupt 1
```

```
       OUT GICR,R16

       LDI R16,0x00   ;setting PORTB as 0, LED off initially
       OUT PORTB,R16

       SEI
ind_loop:rjmp ind_loop

int0_ISR:IN R16,SREG
          PUSH R16

          LDI R16,0x0A;setting R0 with 10 to blink LED 10 times
          MOV R0,R16

     c1:  LDI R16,0x01 ;LED on
          OUT PORTB,R16
;Delay program to make a delay of 1 sec
          LDI R16,0x21
     a1:  LDI R17,0x64
     a2:  LDI R18,0x64
     a3:  DEC R18
          BRNE a3
          DEC R17
          BRNE a2
          DEC R16
          BRNE a1

          LDI R16,0x00 ;LED off
          OUT PORTB,R16
;Delay program to make a delay of 1 sec
          LDI R16,0x21
     b1:  LDI R17,0x64
     b2:  LDI R18,0x64
     b3:  DEC R18
          BRNE b3
```

```
        DEC R17
        BRNE b2
        DEC R16
        BRNE b1


        DEC R0
        BRNE c1 ;LED blinks 10 times
        POP R16
        OUT SREG,R16

        RETI
```

C program:

Int1

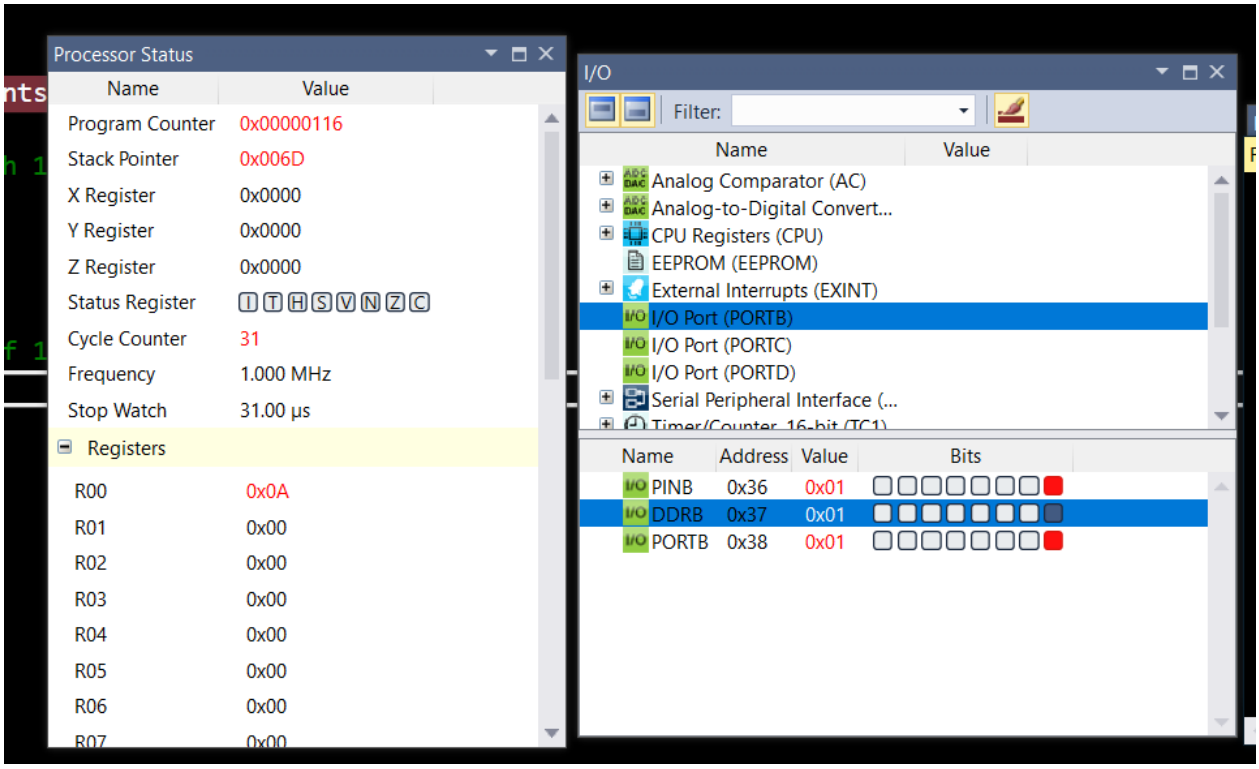
```
#define F_CPU 1000000  // clock frequency

#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>

ISR (INT1_vect)
{
    int i;
    for (i=1;i<=10;i++) // for 10 times LED blink

    {
        PORTB=0x01;
        _delay_ms(1000);   // delay of 1 sec
        PORTB=0x00;
        _delay_ms(1000);

    }
```

```
}
int main(void)
{
     //Set the input/output pins appropriately
     //To enable interrupt and port interfacing
     //For LED to blink
     DDRD=0x00;   //Set appropriate data direction for D
     DDRB=0x01;   //Make PB0 as output
     MCUCR=0x00;  //Set MCUCR to level triggered
     GICR=0x80;   //Enable interrupt 1
     PORTB=0x00;
     sei();       // global interrupt flag

     while (1) //wait
     {

     }
}
```

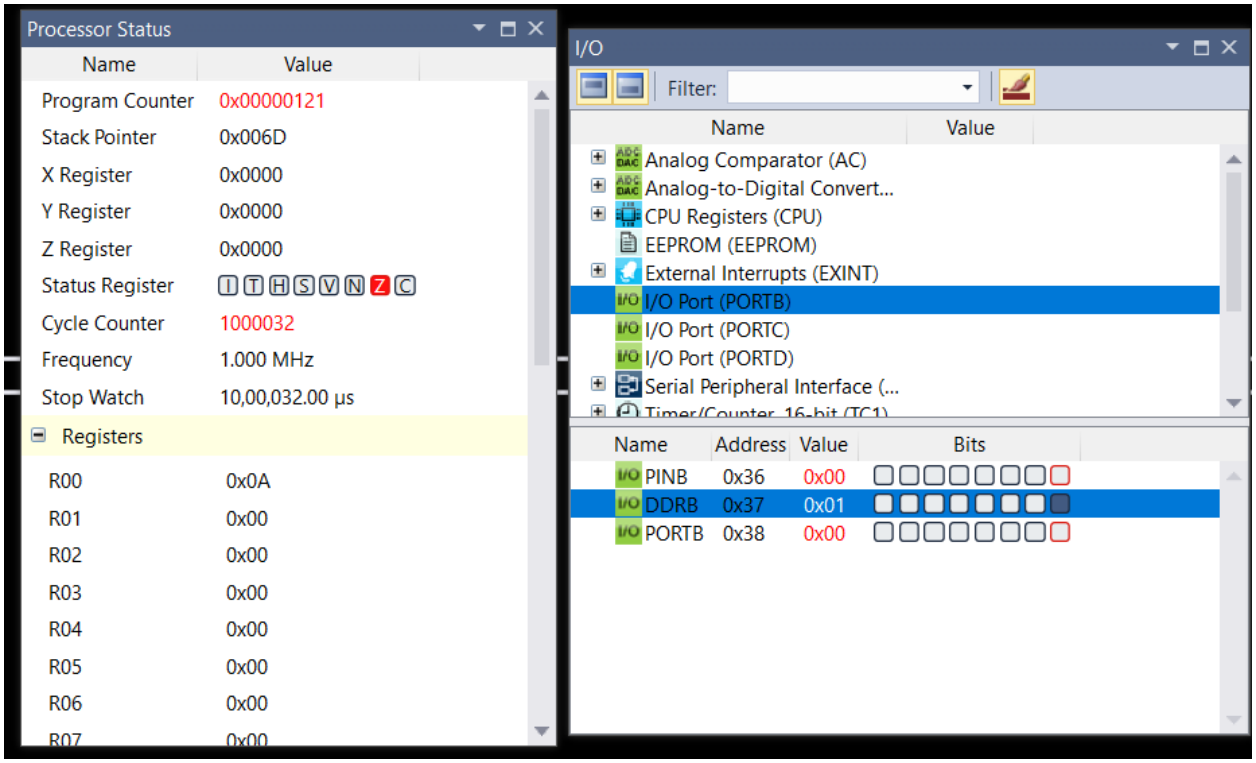
Int0

```
#define F_CPU 1000000  // clock frequency

#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>

ISR (INT0_vect)
{
     int i;
     for (i=1;i<=10;i++) // for 10 times LED blink

     {
```

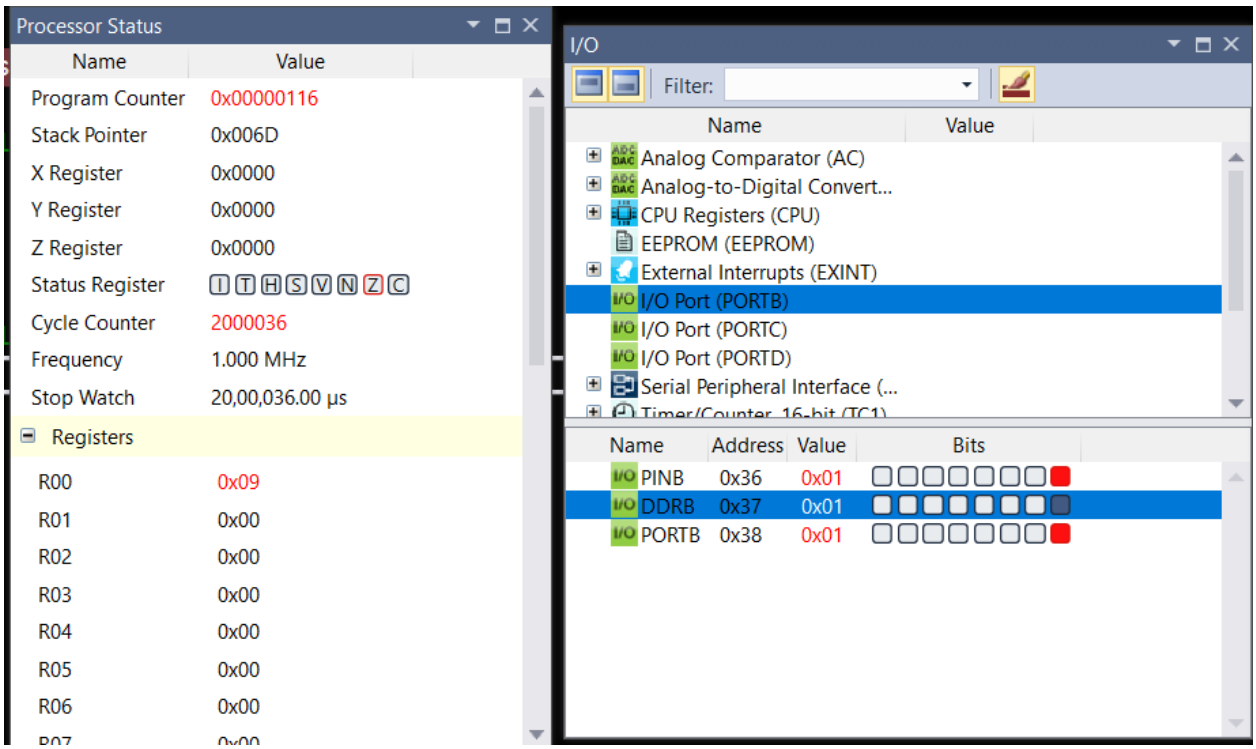
```
          PORTB=0x01;
          _delay_ms(1000);   // delay of 1 sec
          PORTB=0x00;
          _delay_ms(1000);

     }


}
int main(void)
{
     //Set the input/output pins appropriately
     //To enable interrupt and port interfacing
     //For LED to blink
     DDRD=0x00;   //Set appropriate data direction for D
     DDRB=0x01;   //Make PB0 as output
     MCUCR=0x00;  //Set MCUCR to level triggered
     GICR=0x40;   //Enable interrupt 0
     PORTB=0x00;
     sei();       // global interrupt flag

     while (1) //wait
     {

     }
}
```



Just after entering ISR



After 1 second



After another 1 sec

Inferences:

- Learnt the working of interrupts in AVR microcontroller using assembly and c programming language.
- How setting ports as inputs and outputs
- How to enable interrupts and the working of low-level interrupts
- Learnt about the different registers associated with interrupts and their importance
- How to make a required amount of time delays.

- Learnt how to use a loop in loop delay to have a large time delay.
- Learnt how the stack is working and the use of stack pointer
- Could learn how to blink an LED few seconds in Atmega8