GOAL

The goal of this lab is to understand the concept of taking input to AVR, interrupt and analog comparator and finally implement it in AVR.

BACKGROUND

PUSHBUTTON

It is an input device providing 0/1 value signal and is interfaced as shown below:

![Fig. 1. Pushbutton interfacing](image)
EXTERNAL INTERRUPT

Code that is executed is sequential in nature. When we interface our microcontroller to interact with real world, things are not sequential and external world may act in random fashion, interrupt is there to take care of those unpredictable situations. Those external response needs immediate service and interrupt will hold microcontroller's current operation and respond to interrupt service and resume previous ongoing task.

Interrupts are off by default and can be enabled using sei() and disabled by cli(). To use interrupt, we first enable interrupt by using sei() function, that sets I-bit in SREG.

When interrupt occurs, we service the interrupt and disable I-bit so that no other interrupt occurs during the service. External interrupt is in pin PD2 and PD3.

![SREG register](image1)

![External Interrupt Mask Register](image2)
When addressing I/O Registers as data space using LD and ST instructions, the provided offset must be used. When using the I/O specific commands IN and OUT, the offset is reduced by 0x20, resulting in an I/O address offset within 0x00 - 0x3F.

Name: EIFR  
Offset: 0x3C  
Reset: 0x00  
Property: When addressing as I/O Register; address offset is 0x1C

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Access: RAN  
Reset: 0  

Fig. 4. External Interrupt Flag Register

The External Interrupt Control Register A contains control bits for interrupt sense control.

Name: EICRA  
Offset: 0x69  
Reset: 0x00  
Property: -

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ISC11</td>
<td>ISC10</td>
<td>ISC01</td>
<td>ISC00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Access: RAN  
Reset: 0  

Fig. 5. External Interrupt Control Register A

**ANALOG COMPARATOR**

Analog comparator is in pin 12 and 13. If the value at AIN0 > AIN1 then ACO in ACSR is set 1.

**PUSHBUTTON INTERFACE**

```c
#include<avr/io.h>

void main()
{
  DDRC=0b00000010;
  while(1)
  {
    if((PINC&0b00000001)==0b00000001)
      PORTC=0b00000010;
    else
      PORTC=0b00000000;
  }
}
```

UNLV — DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
**INTERRUPT INTERFACE**

```c
#define F_CPU 1000000UL

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

void interrupt_init()
{
    cli();
    EIMSK |= (1 << INT0);
    EIFR = (1 << INTF0);
    EICRA = (1 << ISC01) | (1 << ISC00);
    sei();
}

ISR (INT0_vect)
{
    PORTB = 0xFF;
    _delay_ms(1000);
}

int main()
{
    DDRB = 0xFF;
    DDRD = 0xFF;
    interrupt_init();
    while(1)
    {
        PORTB = 0x00;
    }
}
```

**Fig. 6. Schematic to connect the pushbutton**
**ANALOG COMPARATOR INTERFACE**

```c
#include<avr/io.h>
#include<util/delay.h>

void comp_init(void)
{
    ACSR=0x00;
}

void main()
{
    DDRB=0xFF;
    while(1)
    {
        if((ACSR&0b00100000)==0b00100000)
            PORTB=0xFF;
        else
            PORTB=0x00;
    }
}
```

Fig. 7. Interfacing the pushbutton
Fig. 8. Interfacing Analog Comparator

**PRELAB**

1. **Answer the questions**
   a) Explain in detail each bit of following register: EIMSK, EIFR, EICRA
   b) What is Interrupt Service Routine?
   c) Mention other 4 different interrupt that atmega328p has apart from external interrupt.
   d) What is Switch debouncing? How do you avoid them using hardware and software?
EXPERIMENTS

1. Using the pushbutton
   a) Use given switch code and show the working experiment.
   b) Modify the switch code with different pin for both switch and led. Choose pins of your choice.

2. Programming the interrupt
   a) Use given interrupt code and show the working experiment.
   b) Modify the interrupt code and blink led faster with interrupt and slowly when no interrupt.

3. Programming the comparator
   a) Use given comparator code and show the working experiment.

POSTLAB

Include the following elements in the report document:

<table>
<thead>
<tr>
<th>Section</th>
<th>Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Theory of operation&lt;br&gt;&lt;i&gt;Include a brief description of every element and phenomenon that appears during the experiments.&lt;/i&gt;</td>
</tr>
<tr>
<td>2</td>
<td>Prelab report</td>
</tr>
<tr>
<td>3</td>
<td>Results of the experiments</td>
</tr>
<tr>
<td>Experiment</td>
<td>Experiment Results</td>
</tr>
<tr>
<td>1</td>
<td>a. Modified code with comments&lt;br&gt;b. Picture of the circuit wired on the breadboard</td>
</tr>
<tr>
<td>2</td>
<td>a. Modified code with comments&lt;br&gt;b. Picture of the circuit wired on the breadboard</td>
</tr>
<tr>
<td>3</td>
<td>a. Modified code with comments&lt;br&gt;b. Picture of the circuit wired on the breadboard</td>
</tr>
</tbody>
</table>

4. Answer the questions
<table>
<thead>
<tr>
<th>Question no.</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>What changes needed to be done in experiment 1, to move the operation to the other pins?</td>
</tr>
<tr>
<td>2</td>
<td>What determines the speed of blinking in experiment 2?</td>
</tr>
<tr>
<td>3</td>
<td>What is the principle of operation of the analog comparator in experiment 3?</td>
</tr>
</tbody>
</table>

5. Conclusions<br><i>Write down your conclusions, things learned, problems encountered during the lab and how they were solved, etc.</i>