ECE375
External Interrupt

TA:
School of Electrical Engineering and Computer Science
Oregon State University
External Interrupts

- Understand Interrupts
- Demonstrate BumpBot using external Interrupts
- Explore the ATmega32U4 datasheet
- BumpBot counts each whisker and displays on LCD
- BumpBot clears both whiskers’ counters on LCD
Interrupts Handling

Right Whisker (Interrupt Occurs)
INT0

Go Forward
(Main Program)
Interrupts Handling

Right Whisker
(Interrupt Occurs)
INT0

Check Interrupt Vector
2 $0002 INT0

.org $0002
rcall HitRight
reti

Go Forward
(Main Program)
Interrupts Handling

Right Whisker (Interrupt Occurs) INT0

Check Interrupt Vector
2 $0002 INT0

.org $0002
rcall HitRight
reti

Go Forward (Main Program)

Back up and turn away from the object (Interrupt Subroutine)
ret
Interrupts Handling

Right Whisker (Interrupt Occurs) INTO

Check Interrupt Vector
2 $0002 INTO

.org $0002
rcall HitRight
reti

Go Forward (Main Program)

Back up and turn away from the object (Interrupt Subroutine)

ret
Interrupts Handling

Right Whisker (Interrupt Occurs) INT0

Check Interrupt Vector 2 $0002 INTO

.org $0002
rcall HitRight
reti

Go Forward (Main Program)

Back up and turn away from the object (Interrupt Subroutine)
External Interrupts

- The External Interrupts are triggered by the INT3:0 and INT6 pins.
  - PIN\textsubscript{D}3:0 = INT3:0
  - PIN\textsubscript{E}6 = INT6
Recall that we use PD4-7 for push buttons
Tekbot Bumper Switch Connection

To trigger interrupts with the push buttons, plug jumpers into the board to connect:

- PD0 <=> PD4
- PD1 <=> PD5
- PD3 <=> PD6

Do NOT jump PD2. The pushbuttons cannot pull the PD2 logic low.
External Interrupts

- The External Interrupts are triggered by the INT3:0 and INT6 pins.

- The External Interrupts can be triggered by a falling, a rising edge, or a low level.
  - EICRA (INT3:0) and EICRB (INT6)
External Interrupts

- The External Interrupts are triggered by the INT3:0 and INT6 pins.

- The External Interrupts can be triggered by a falling, a **rising edge**, or a low level.
  - EICRA (INT3:0) and EICRB (INT6)
**External Interrupts**

- The External Interrupts are triggered by the INT3:0 and INT6 pins.

- The External Interrupts can be triggered by a falling, a rising edge, or a low level.
  - EICRA (INT3:0) and EICRB (INT6)
# External Interrupt Control Register

<table>
<thead>
<tr>
<th></th>
<th>INT3</th>
<th></th>
<th>INT2</th>
<th></th>
<th>INT1</th>
<th></th>
<th>INT0</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ISC3</td>
<td>ISC0</td>
<td>ISC2</td>
<td>ISC2</td>
<td>ISC1</td>
<td>ISC1</td>
<td>ISC0</td>
<td>ISC0</td>
<td></td>
</tr>
</tbody>
</table>

## EICRA

<table>
<thead>
<tr>
<th></th>
<th>INT6</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>ISC6</td>
<td>ISC6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

## EICRB

**ISCn1:0 External Interrupt n Sense Control Bits**

- **00** – Low level generates an interrupt
- **01** – Any edge generates an interrupt request (for ISC3-0)
  - Any logical change generates an interrupt request (for ISC6)
- **10** - Falling Edge generates an interrupt request
- **11** – Rising Edge generates an interrupt request
External Interrupt Control Register

<table>
<thead>
<tr>
<th>ISCn1</th>
<th>ISCn0</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>The low level of INTn generates an interrupt request.</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Any edge of INTn generates asynchronously an interrupt request.</td>
</tr>
<tr>
<td><strong>1</strong></td>
<td><strong>0</strong></td>
<td>The falling edge of INTn generates asynchronously an interrupt request.</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>The rising edge of INTn generates asynchronously an interrupt request.</td>
</tr>
</tbody>
</table>

Note: 1. \( n = 3, 2, 1, \text{ or } 0 \).

When changing the ISCn1/ISCn0 bits, the interrupt must be disabled by clearing its Interrupt Enable bit in the EIMSK Register. Otherwise an interrupt can occur when the bits are changed.

ldi mpr, 0b0000_0010
sts EICRA, mpr
## External Interrupt Mask Register

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>INT6</td>
<td>-</td>
<td>-</td>
<td>INT3</td>
<td>INT2</td>
<td>INT1</td>
<td>INT0</td>
<td></td>
</tr>
</tbody>
</table>

**EIMSK**

- `ldi mpr, 0b0000_0001`
- `out EIMSK, mpr`
### Determining Source of Interrupt

- When an interrupt occurs each source of interrupt is mapped to a vector.

<table>
<thead>
<tr>
<th>Vector #</th>
<th>Program Address</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$0000</td>
<td>RESET</td>
</tr>
<tr>
<td>2</td>
<td>$0002</td>
<td>External Interrupt Request 0 (INT0)</td>
</tr>
<tr>
<td>3</td>
<td>$0004</td>
<td>External Interrupt Request 1 (INT1)</td>
</tr>
<tr>
<td>4</td>
<td>$0006</td>
<td>External Interrupt Request 2 (INT2)</td>
</tr>
<tr>
<td>5</td>
<td>$0008</td>
<td>External Interrupt Request 3 (INT3)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>8</td>
<td>$000E</td>
<td>External Interrupt Request 6 (INT6)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>22</td>
<td>$002A</td>
<td>Timer/Counter0 Compare Match A</td>
</tr>
<tr>
<td>23</td>
<td>$002C</td>
<td>Timer/Counter0 Compare Match B</td>
</tr>
<tr>
<td>21</td>
<td>$002E</td>
<td>Timer/Counter0 Overflow</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

There are 43 vectors!
Avoiding Queued Interrupts
Avoiding Queued Interrupts

To clear the queue, set EIFR here.
External Interrupt Flag Register

- Write “1” in order to clear the queue to EIFR
  - \texttt{ldi mpr, 0b0000_0001}
  - \texttt{out EIFR, mpr}

\text{INT0}
ATmega32U4 I/O registers

- ATmega32U4 I/O registers for External Interrupts
  - External Interrupt Control Register A – EICRA
  - External Interrupt Control Register B – EICRB
  - External Interrupt Mask Register – EIMSK
  - External Interrupt Flag Register – EIFR
  - sei ; set interrupt
I/O Port Registers

These I/O registers can be accessed using IN/OUT instructions.
Demo Check

- **BumpBot Behavior using Interrupts**
  - Need to avoid queued interrupts

- **LCD displays two counters**
  - Count Right/Left Whiskers
  - Implement clearing each counter
    - **Hint**: Use Bin2ASCII function in LCDDriver.asm to display decimal numbers.
    - It must be able to display both counters greater than 10.
    - Do not show any garbage data when increment/clear the counters.

- **Implement 3 interrupts properly**
  - INT0 and INT1 for counting Right/Left Whiskers
  - INT3 for clearing Right/Left whisker counters
  - Do not use INT2.
Checklists

- Demo Checklist
  - Standard BumpBot behavior observed
  - Must use interrupts, not polling
  - Queued interrupts explicitly avoided
  - Nested interrupts not enabled
  - Correctly configured INT0 – INT3 to use falling-edge sense control
  - Correctly wired PD0-7 to use interrupt
Questions?