# ECE375
# External Interrupt

**TA: Dongjun Lee**
School of Electrical Engineering and Computer Science
Oregon State University

## External Interrupts

- Understand Interrupts
- Demonstrate BumpBot using external Interrupts
- Explore the ATmega128 datasheet
- BumpBot counts each whisker and displays on LCD

# External Interrupts

Phone rings
(Interrupt Occurs)

Writing
Report
(Main Program)

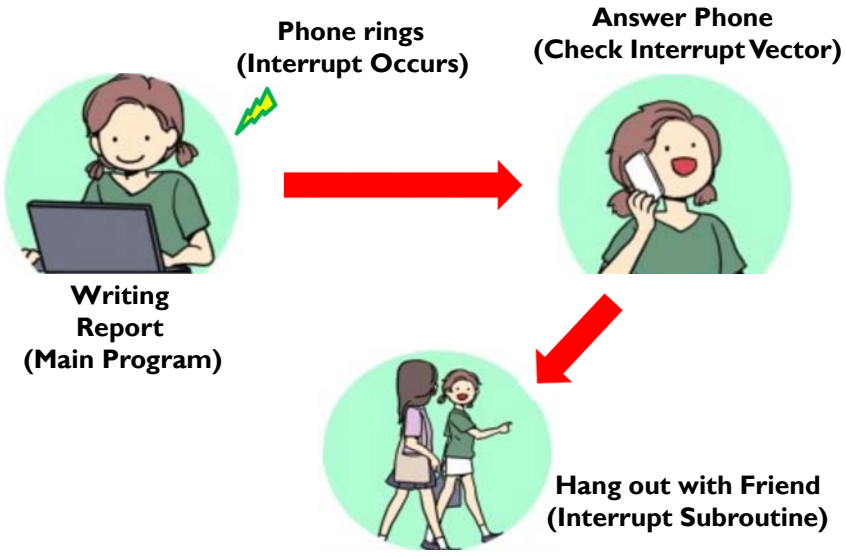
# External Interrupts

Phone rings
(Interrupt Occurs)

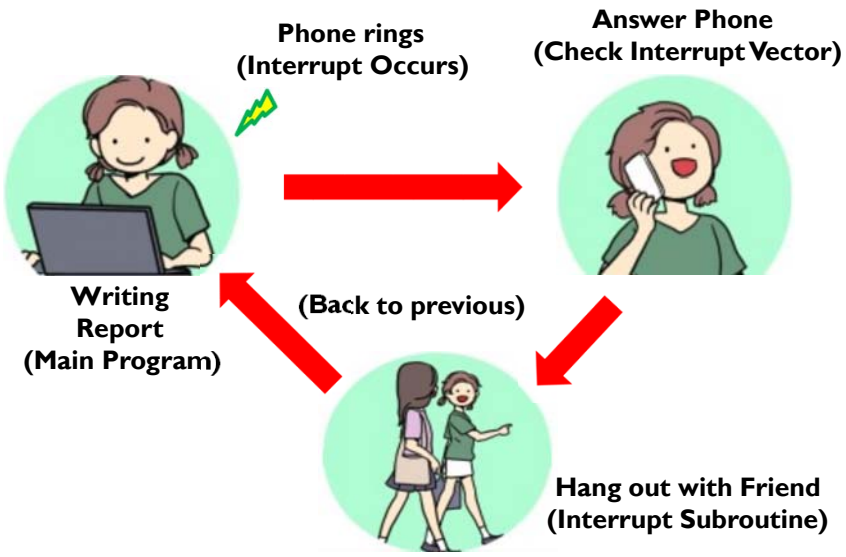
Answer Phone
(Check Interrupt Vector)

Writing
Report
(Main Program)

# External Interrupts




# External Interrupts



# 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)

# 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)

## External Interrupts

- The External Interrupts are triggered by the INT7:0 pins.
- The External Interrupts can be triggered by a falling, a rising edge, or a low level.
  - EICRA (INT3:0) and EICRB (INT7:4)
    - PIND3:0 = INT3:0
    - PINE7:4 = INT7:4


## External Interrupts

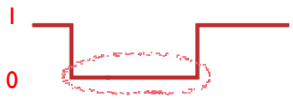
- The External Interrupts are triggered by the INT7:0 pins.
- The External Interrupts can be triggered by a falling, a rising edge, or a low level.
  - EICRA (INT3:0) and EICRB (INT7:4)
    - PIND3:0 = INT3:0
    - PINE7:4 = INT7:4

## External Interrupts

- The External Interrupts are triggered by the INT7:0 pins.
- The External Interrupts can be triggered by a falling, a rising edge, or a low level.
  - EICRA (INT3:0) and EICRB (INT7:4)
    - PIND3:0 = INT3:0
    - PINE7:4 = INT7:4

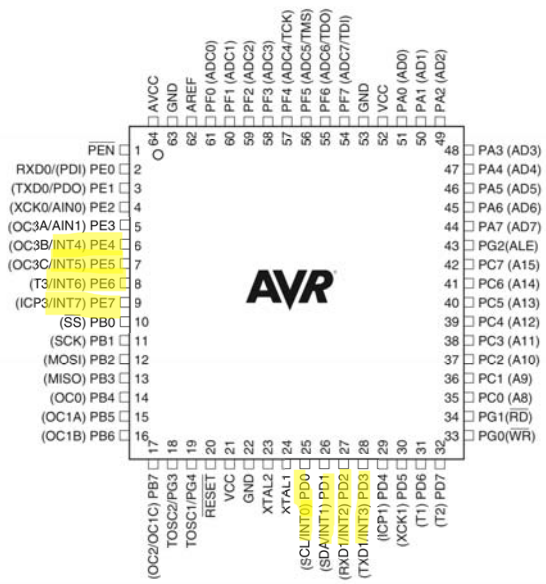

## External Interrupts

- The External Interrupts are triggered by the INT7:0 pins.
- The External Interrupts can be triggered by a falling, a rising edge, or a low level.
  - EICRA (INT3:0) and EICRB (INT7:4)
    - PIND3:0 = INT3:0
    - PINE7:4 = INT7:4



## Pin Configurations

**Figure 1.** Pinout ATmega128




(Recall that we use these for push button as well!)

## External Interrupt Control Register

| | INT2 | | | INT1 | | INT0 | |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ISC31 | ISC30 | ISC21 | ISC20 | ISC11 | ISC10 | ISC01 | ISC00 |

INT3

**EICRA**

| INT7 | | INT6 | | INT5 | | INT4 | |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ISC71 | ISC70 | ISC61 | ISC60 | ISC51 | ISC50 | ISC41 | ISC40 |

**EICRB**

```
ldi mpr, 0b0000_1010
sts EICRA, mpr
```

# External Interrupt Control Register

**Table 48.** Interrupt Sense Control[1]

| ISCn1 | ISCn0 | Description |
|---|---|---|
| 0 | 0 | The low level of INTn generates an interrupt request. |
| 0 | 1 | Reserved |
| 1 | 0 | The falling edge of INTn generates asynchronously an interrupt request. |
| 1 | 1 | The rising edge of INTn generates asynchronously an interrupt request. |

Note: 1. n = 3, 2, 1or 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.

INT1 INT0

**ldi mpr, 0b0000_1010**

**sts EICRA, mpr**

# External Interrupt Mask Register

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| INT7 | INT6 | INT5 | INT4 | INT3 | INT2 | INT1 | INT0 |

**EIMSK**

**ldi mpr, 0b0000_0011**

**out EIMSK, mpr**

# Determining Source of Interrupt

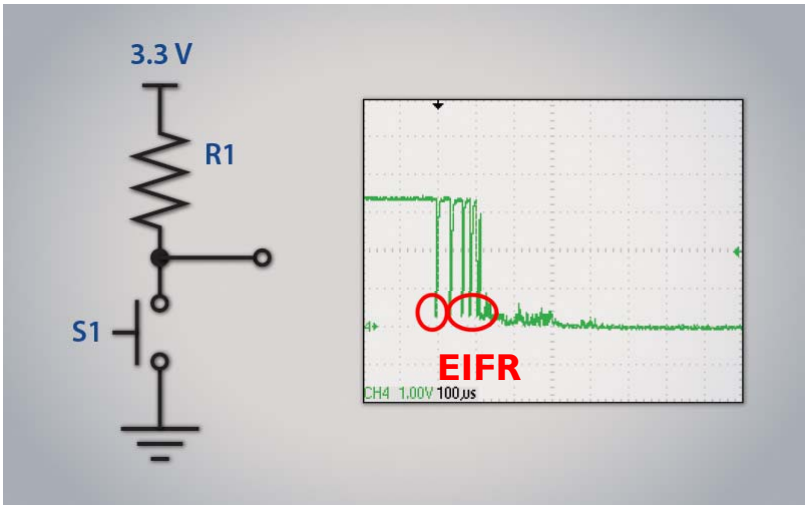
- When an interrupt occurs each source of interrupt is mapped to a vector.

| Vector # | Program Address | Source |
|---|---|---|
| 1 | $0000 | RESET |
| 2 | $0002 | INT0 |
| 3 | $0004 | INT1 |
| … | … | … |
| 9 | $0010 | INT7 |
| 10 | $0012 | TIMER2 COMP |
| 11 | $0014 | TIMER2 OVF |
| … | … | … |

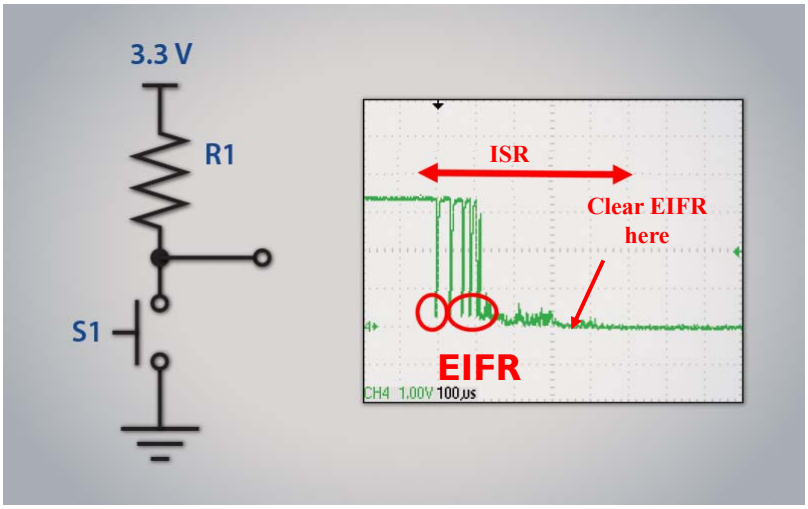
Priority
High
Low

There are 35 vectors!

# Avoiding Queued Interrupts



# Avoiding Queued Interrupts




# External Interrupt Flag Register

- Write "1" in order to clear EIFR
  - ldi mpr, 0b0000_0011
  - out EIFR, mpr

# ATmega128 I/O registers

- ATmega128 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

# 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 4 interrupts properly
  - INT0 and INT 1 for counting Right/Left Whiskers
  - INT2 and INT 3 for clearing Right/Left whisker counters

# Checklists for Lab 6

- Demo Checklist
  - Standard BumpBot behavior observed
  - Actually used interrupts, not polling
  - Queued interrupts explicitly avoided
  - Nested interrupts not enabled
  - Correctly configured INT0 – INT3 to use falling-edge sense control
- Challenge Checklist
  - Correct alternating-whisker behavior
  - Correct repeated-whisker behavior

# Questions?