

Interrupts On AlphaBot

Embedded Real-Time Systems (ERTS) Lab
Indian Institute of Technology, Bombay



Agenda for Discussion

1 Overview

- What is an Interrupt
- Sources of Interrupt
- External Interrupt
- Interrupt Pins
- Position Encoder
- Interrupt Calculation

2 Registers

- SREG
- EIMSK
- EICRA
- ISR
- C-Code



What is an Interrupt



What is an Interrupt

- Any signal that causes break in continuity of some ongoing process



What is an Interrupt

- ✓ Any signal that causes break in continuity of some ongoing process
- ✓ In microcontrollers interrupt signal halts the execution of main program and dedicates processor to another task

Main program execution

```
while ( ) {  
    Instruction 1  
    Instruction 2  
    Instruction 3  
    Instruction 4  
    Instruction 5  
    Instruction 6  
}
```



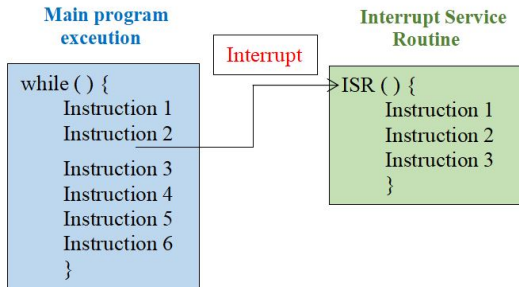
What is an Interrupt

- While main program is running, if an interrupt occurs, execution of main program is stopped, and program counter goes to address of ISR



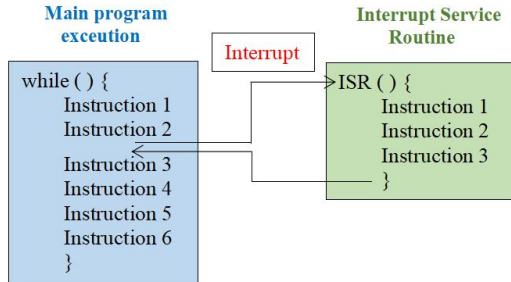
What is an Interrupt

- While main program is running, if an interrupt occurs, execution of main program is stopped, and program counter goes to address of ISR
- Interrupt Service Routine: Program that needs to be executed when interrupt occurs



What is an Interrupt

- After program inside ISR is executed completely, program counter returns back to point where main program was interrupted



Sources of Interrupt in ATmega328p



Sources of Interrupt in ATmega328p

ATmega 328p has **Twenty-six** different sources for Interrupt generation



Sources of Interrupt in ATmega328p

ATmega 328p has **Twenty-six** different sources for Interrupt generation

① RESET Interrupt - [1]



Sources of Interrupt in ATmega328p

ATmega 328p has **Twenty-six** different sources for Interrupt generation

- 1 RESET Interrupt - [1]
- 2 External hardware Interrupt - [2]



Sources of Interrupt in ATmega328p

ATmega 328p has **Twenty-six** different sources for Interrupt generation

- ❶ RESET Interrupt - [1]
- ❷ External hardware Interrupt - [2]
- ❸ Pin Change Interrupt Request - [3]



Sources of Interrupt in ATmega328p

ATmega 328p has **Twenty-six** different sources for Interrupt generation

- ➊ RESET Interrupt - [1]
- ➋ External hardware Interrupt - [2]
- ➌ Pin Change Interrupt Request - [3]
- ➍ Timer/Counter Interrupts



Sources of Interrupt in ATmega328p

ATmega 328p has **Twenty-six** different sources for Interrupt generation

- ➊ RESET Interrupt - [1]
- ➋ External hardware Interrupt - [2]
- ➌ Pin Change Interrupt Request - [3]
- ➍ Timer/Counter Interrupts
 - Timer/Counter0 - [3]
 - Timer/Counter1 - [4]
 - Timer/Counter2 - [3]



Sources of Interrupt in ATmega328p

ATmega 328p has **Twenty-six** different sources for Interrupt generation

- ❶ RESET Interrupt - [1]
- ❷ External hardware Interrupt - [2]
- ❸ Pin Change Interrupt Request - [3]
- ❹ Timer/Counter Interrupts
 - Timer/Counter0 - [3]
 - Timer/Counter1 - [4]
 - Timer/Counter2 - [3]
- ❺ Serial Interrupts



Sources of Interrupt in ATmega328p

ATmega 328p has **Twenty-six** different sources for Interrupt generation

- ➊ RESET Interrupt - [1]
- ➋ External hardware Interrupt - [2]
- ➌ Pin Change Interrupt Request - [3]
- ➍ Timer/Counter Interrupts
 - Timer/Counter0 - [3]
 - Timer/Counter1 - [4]
 - Timer/Counter2 - [3]
- ➎ Serial Interrupts
 - USART - [3]
 - SPI - [1]
 - TWI - [1]



Sources of Interrupt in ATmega328p

ATmega 328p has **Twenty-six** different sources for Interrupt generation

- ➊ RESET Interrupt - [1]
- ➋ External hardware Interrupt - [2]
- ➌ Pin Change Interrupt Request - [3]
- ➍ Timer/Counter Interrupts
 - Timer/Counter0 - [3]
 - Timer/Counter1 - [4]
 - Timer/Counter2 - [3]
- ➎ Serial Interrupts
 - USART - [3]
 - SPI - [1]
 - TWI - [1]
- ➏ Others [5] such as Analog Comparator, ADC Conversion Complete, EEPROM, SPM and Watchdog timer.



What is an External Hardware Interrupt



What is an External Hardware Interrupt

- ① It is an interrupt signal sent to the controller from an external device, like a disk controller or an external peripheral.



What is an External Hardware Interrupt

- ① It is an interrupt signal sent to the controller from an external device, like a disk controller or an external peripheral.
- ② ATmega328p has 2 hardware interrupt pins (namely INT_n where n can be 0 or 1).



What is an External Hardware Interrupt

- 1 It is an interrupt signal sent to the controller from an external device, like a disk controller or an external peripheral.
- 2 ATmega328p has 2 hardware interrupt pins (namely INT_n where n can be 0 or 1).
- 3 To use an external interrupt, the pin has to be configured as a standard IO input.



What is an External Hardware Interrupt

- ① It is an interrupt signal sent to the controller from an external device, like a disk controller or an external peripheral.
- ② ATmega328p has 2 hardware interrupt pins (namely INT_n where n can be 0 or 1).
- ③ To use an external interrupt, the pin has to be configured as a standard IO input.
- ④ If pin is used as an input, external hardware device can be used to interrupt the controller.



Interrupt pins



Interrupt pins

Sr. no	Interrupt	Arduino Pin	Port Pin	AlphaBot Connection
1	INT0	2	PD2	Left encoder
2	INT1	3	PD3	Right encoder



Interrupt pins

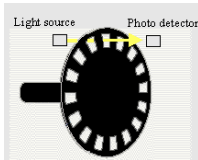
Sr. no	Interrupt	Arduino Pin	Port Pin	AlphaBot Connection
1	INT0	2	PD2	Left encoder
2	INT1	3	PD3	Right encoder



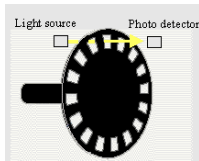
Position encoder



Position encoder

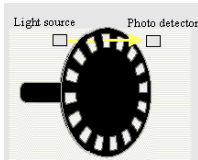


Position encoder



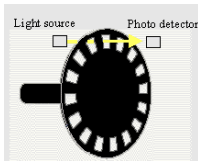
Position encoder

- 1 Encoder consists of IR LED and photo transistor placed opposite of each other

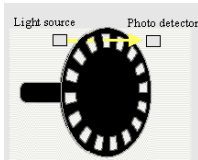


Position encoder

- 1 Encoder consists of IR LED and photo transistor placed opposite of each other
- 2 When IR light is interrupted by encoder disc, its output state changes (high to low or low to high)

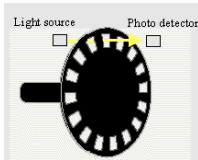


Position encoder



- 1 Encoder consists of IR LED and photo transistor placed opposite of each other
- 2 When IR light is interrupted by encoder disc, its output state changes (high to low or low to high)
- 3 Output of the encoder is connected to the interrupt pin of the microcontroller

Position encoder



- 1 Encoder consists of IR LED and photo transistor placed opposite of each other
- 2 When IR light is interrupted by encoder disc, its output state changes (high to low or low to high)
- 3 Output of the encoder is connected to the interrupt pin of the microcontroller
- 4 Left encoder is connected to INT0 and Right encoder is connected to INT1



Some Mathematics...



Some Mathematics...

① Number of slots in disc = 20



Some Mathematics...

- ① Number of slots in disc = 20
- ② Number of Pulse/rotation = 20



Some Mathematics...

- ① Number of slots in disc = 20
- ② Number of Pulse/rotation = 20
- ③ Diameter of wheel = 60mm



Some Mathematics...

- ① Number of slots in disc = 20
- ② Number of Pulse/rotation = 20
- ③ Diameter of wheel = 60mm
- ④ Resolution of position encoder



Some Mathematics...

- 1 Number of slots in disc = 20
- 2 Number of Pulse/rotation = 20
- 3 Diameter of wheel = 60mm
- 4 Resolution of position encoder



Some Mathematics...

- ① Number of slots in disc = 20
- ② Number of Pulse/rotation = 20
- ③ Diameter of wheel = 60mm
- ④ Resolution of position encoder
$$= (\pi * d) / 20 = 9.42$$



Some Mathematics...

- ① Number of slots in disc = 20
- ② Number of Pulse/rotation = 20
- ③ Diameter of wheel = 60mm
- ④ Resolution of position encoder
$$= (\pi * d) / 20 = 9.42$$
- ⑤ Pulse count
$$= \text{distance} / 9.42$$



SREG- AVR Status Register

This register is used to Globally Enable all Interrupt



SREG- AVR Status Register

This register is used to Globally Enable all Interrupt

Bit	Symbol	Description	Bit Value
7	I	Global Interrupt Enable bit	1
6	T	Bit Copy Storage bit	0
5	H	Half Carry Flag	0
4	S	Sign Bit	0
3	V	Two's Complement Overflow Flag	0
2	N	Negative Flag	0
1	Z	Zero Flag	0
0	C	Carry Flag	0



SREG- AVR Status Register

This register is used to Globally Enable all Interrupt

Bit	Symbol	Description	Bit Value
7	I	Global Interrupt Enable bit	1
6	T	Bit Copy Storage bit	0
5	H	Half Carry Flag	0
4	S	Sign Bit	0
3	V	Two's Complement Overflow Flag	0
2	N	Negative Flag	0
1	Z	Zero Flag	0
0	C	Carry Flag	0

Note: `cli()` and `sei()` are used to clear and set global interrupt respectively



SREG- AVR Status Register

This register is used to Globally Enable all Interrupt

Bit	Symbol	Description	Bit Value
7	I	Global Interrupt Enable bit	1
6	T	Bit Copy Storage bit	0
5	H	Half Carry Flag	0
4	S	Sign Bit	0
3	V	Two's Complement Overflow Flag	0
2	N	Negative Flag	0
1	Z	Zero Flag	0
0	C	Carry Flag	0

Note: `cli()` and `sei()` are used to clear and set global interrupt respectively

(defined in `<avr/interrupt.h>` header file)



EIMSK- External Interrupt Mask Register

This register is Used to enable Individual External Interrupt



EIMSK- External Interrupt Mask Register

This register is Used to enable Individual External Interrupt

Bit	Symbol	Description	Bit Value
7	-	Reserved Bit	0
6	-	Reserved Bit	0
5	-	Reserved Bit	0
4	-	Reserved Bit	0
3	-	Reserved Bit	0
2	-	Reserved Bit	0
1	INT1	External Interrupt Request 1	1
0	INT0	External Interrupt Request 0	1



EIMSK- External Interrupt Mask Register

This register is Used to enable Individual External Interrupt

Bit	Symbol	Description	Bit Value
7	-	Reserved Bit	0
6	-	Reserved Bit	0
5	-	Reserved Bit	0
4	-	Reserved Bit	0
3	-	Reserved Bit	0
2	-	Reserved Bit	0
1	INT1	External Interrupt Request 1	1
0	INT0	External Interrupt Request 0	1

EIMSK = 0x03



Interrupt Sense Control Bits



Interrupt Sense Control Bits

ISC _n 1	ISC _n 0	Description
0	0	The low level of INT _n generates an Interrupt request
0	1	Any edge of INT _n generates asynchronously an interrupt request
1	0	The falling edge of INT _n generates asynchronously an interrupt request
1	1	The rising edge of INT _n generates asynchronously an interrupt request

where n = External Interrupt Number (For Atmega328p: n = 0 or 1)

For External Interrupt = 0

Interrupt Sense Control Bit = ISC01 and ISC00



EICRA- External Interrupt Control Register A

This register is Used to generate Interrupt Signal



EICRA- External Interrupt Control Register A

This register is Used to generate Interrupt Signal

Bit	Symbol	Description	Bit Value
7	-	Reserved Bit	0
6	-	Reserved Bit	0
5	-	Reserved Bit	0
4	-	Reserved Bit	0
3	ISC11	Interrupt Sense control bit for Ext. Interrupt 1	1
2	ISC10	Interrupt Sense control bit for Ext. Interrupt 1	0
1	ISC01	Interrupt Sense control bit for Ext. Interrupt 0	1
0	ISC00	Interrupt Sense control bit for Ext. Interrupt 0	0



EICRA- External Interrupt Control Register A

This register is Used to generate Interrupt Signal

Bit	Symbol	Description	Bit Value
7	-	Reserved Bit	0
6	-	Reserved Bit	0
5	-	Reserved Bit	0
4	-	Reserved Bit	0
3	ISC11	Interrupt Sense control bit for Ext. Interrupt 1	1
2	ISC10	Interrupt Sense control bit for Ext. Interrupt 1	0
1	ISC01	Interrupt Sense control bit for Ext. Interrupt 0	1
0	ISC00	Interrupt Sense control bit for Ext. Interrupt 0	0

EICRA = 0x0A



ISR-Interrupt Service Routine



ISR-Interrupt Service Routine

The format of ISR for external interrupt is



ISR-Interrupt Service Routine

The format of ISR for external interrupt is

ISR Format



ISR-Interrupt Service Routine

The format of ISR for external interrupt is

ISR Format

```
ISR(INTn_vect)
{
    code
}
```



ISR-Interrupt Service Routine

The format of ISR for external interrupt is

ISR Format

```
ISR(INTn_vect)
{
    code
}
```

Where n = External Interrupt Number (For Atmega328p: n=0 or 1)



Syntax for C-Program

Port Initialization



Syntax for C-Program

Port Initialization

Left Encoder Port Initialization



Syntax for C-Program

Port Initialization

Left Encoder Port Initialization

```
void left_encoder_pin_config (void)
{
    DDRD &= ~(1 << left_encoder); //Set the direction of the PORTD 2 pin as input
    PORTD |= (1 << left_encoder); //Enable internal pull-up for PORTD 2 pin
}
```



Syntax for C-Program

Port Initialization

Left Encoder Port Initialization

```
void left_encoder_pin_config (void)
{
    DDRD &= ~(1 << left_encoder); //Set the direction of the PORTD 2 pin as input
    PORTD |= (1 << left_encoder); //Enable internal pull-up for PORTD 2 pin
}
```

Right Encoder Port Initialization



Syntax for C-Program

Port Initialization

Left Encoder Port Initialization

```
void left_encoder_pin_config (void)
{
    DDRD &= ~(1 << left_encoder); //Set the direction of the PORTD 2 pin as input
    PORTD |= (1 << left_encoder); //Enable internal pull-up for PORTD 2 pin
}
```

Right Encoder Port Initialization

```
void right_encoder_pin_config (void)
{
    DDRD &= ~(1 << right_encoder); //Set the direction of the PORTD 3 pin as input
    PORTD |= (1 << right_encoder); //Enable internal pull-up for PORTD 3 pin
}
```



Syntax for C-Program

Interrupt Initialization



Syntax for C-Program

Interrupt Initialization

Left-Encoder Interrupt Initialization



Syntax for C-Program

Interrupt Initialization

Left-Encoder Interrupt Initialization

```
void left_position_encoder_interrupt_init (void) //Interrupt 0 enable
{
    cli();                //Clears the global interrupt
    EICRA = EICRA | 0x02; // INTO is set to trigger with falling edge
    EIMSK = EIMSK | 0x01; // Enable Interrupt INTO for left position encoder
    sei();                // Enables the global interrupt
}
```



Syntax for C-Program

Interrupt Initialization

Left-Encoder Interrupt Initialization

```
void left_position_encoder_interrupt_init (void) //Interrupt 0 enable
{
    cli();                //Clears the global interrupt
    EICRA = EICRA | 0x02; // INTO is set to trigger with falling edge
    EIMSK = EIMSK | 0x01; // Enable Interrupt INTO for left position encoder
    sei();                // Enables the global interrupt
}
```

Right-Encoder Interrupt Initialization



Syntax for C-Program

Interrupt Initialization

Left-Encoder Interrupt Initialization

```
void left_position_encoder_interrupt_init (void) //Interrupt 0 enable
{
    cli();                //Clears the global interrupt
    EICRA = EICRA | 0x02; // INTO is set to trigger with falling edge
    EIMSK = EIMSK | 0x01; // Enable Interrupt INTO for left position encoder
    sei();                // Enables the global interrupt
}
```

Right-Encoder Interrupt Initialization

```
void right_position_encoder_interrupt_init (void) //Interrupt 1 enable
{
    cli();                //Clears the global interrupt
    EICRA = EICRA | 0x08; // INT1 is set to trigger with falling edge
    EIMSK = EIMSK | 0x02; // Enable Interrupt INT1 for right position encoder
    sei();                // Enables the global interrupt
}
```



Thank You!

Post your queries on: support@e-yantra.org

