

JOBSHEET VII

MENGUNAKAN *INTERRUPT* DALAM MIKROKONTROLLER ATMEGA8535

TUJUAN

- Menjelaskan fitur interrupt dalam mikrokontroler.
- Mengetahui dan memahami bagaimana memrogram mikrokontroler untuk menjalankan interupsi pada saat program utama dijalankan

INTERRUPT

Interrupt adalah kondisi di mana pada saat program utama dieksekusi/dikerjakan oleh CPU kemudian tiba-tiba berhenti untuk sementara waktu karena ada rutin lain yang harus ditangani terlebih dahulu oleh CPU. Setelah selesai mengerjakan rutin tersebut, CPU kembali mengerjakan instruksi pada program utama. Port yang berfungsi sebagai interrupt eksternal adalah:

PORTD.2 (PD2) → External Interrupt 0 Input (INT0)

PORTD.3 (PD3) → External Interrupt 1 Input (INT1)

PORTB.2 (PB2) → External Interrupt 2 Input (INT2)

Di dalam mikrokontroler AVR terdapat 21 interrupt baik eksternal maupun internal. Interrupt tersebut dapat dilihat dalam tabel berikut.

Table 19. Reset and Interrupt Vectors

Vector No.	Program Address ⁽²⁾	Source	Interrupt Definition
1	0x000 ⁽¹⁾	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog Reset
2	0x001	INT0	External Interrupt Request 0
3	0x002	INT1	External Interrupt Request 1
4	0x003	TIMER2 COMP	Timer/Counter2 Compare Match
5	0x004	TIMER2 OVF	Timer/Counter2 Overflow
6	0x005	TIMER1 CAPT	Timer/Counter1 Capture Event
7	0x006	TIMER1 COMPA	Timer/Counter1 Compare Match A
8	0x007	TIMER1 COMPB	Timer/Counter1 Compare Match B
9	0x008	TIMER1 OVF	Timer/Counter1 Overflow
10	0x009	TIMER0 OVF	Timer/Counter0 Overflow
11	0x00A	SPI, STC	Serial Transfer Complete
12	0x00B	USART, RXC	USART, Rx Complete
13	0x00C	USART, UDRE	USART Data Register Empty
14	0x00D	USART, TXC	USART, Tx Complete
15	0x00E	ADC	ADC Conversion Complete
16	0x00F	EE_RDY	EEPROM Ready
17	0x010	ANA_COMP	Analog Comparator
18	0x011	TWI	Two-wire Serial Interface
19	0x012	INT2	External Interrupt Request 2
20	0x013	TIMER0 COMP	Timer/Counter0 Compare Match
21	0x014	SPM_RDY	Store Program Memory Ready

Register-register yang digunakan untuk interrupt meliputi:

a. MCU Control Register-MCUCR

Bit	7	6	5	4	3	2	1	0	
	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	MCUCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- Bit 3,2 – ISC11, ISC10: Interrupt Sense Control 1 Bit 1 dan Bit 0

Table 35. Interrupt 1 Sense Control

ISC11	ISC10	Description
0	0	The low level of INT1 generates an interrupt request.
0	1	Any logical change on INT1 generates an interrupt request.
1	0	The falling edge of INT1 generates an interrupt request.
1	1	The rising edge of INT1 generates an interrupt request.

- Bit 1,0 – ISC01, ISC00: Interrupt Sense Control 0 Bit 1 dan Bit 0

Table 36. Interrupt 0 Sense Control

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request.
0	1	Any logical change on INT0 generates an interrupt request.
1	0	The falling edge of INT0 generates an interrupt request.
1	1	The rising edge of INT0 generates an interrupt request.

b. MCU Control and Status Register-MCUCSR

Bit	7	6	5	4	3	2	1	0	
	–	ISC2	–	–	WDRF	BORF	EXTRF	PORF	MCUCSR
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	See Bit Description					

- ISC2 = 0, a falling edge on INT2 activates the interrupt
- ISC2 = 1, a rising edge on INT2 activates the interrupt
- Lebar pulsa minimum: 50 ns

c. Status Register-SREG

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – I: Global Interrupt Enable**

-

The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

d. General Interrupt Control Register-GICR

Bit	7	6	5	4	3	2	1	0	
	INT1	INT0	INT2	–	–	–	IVSEL	IVCE	GICR
Read/Write	R/W	R/W	R/W	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- Bit 7 - INT1:eksternal interrupt request 1 enable
- Bit 6 – INT0 :eksternal interrupt request 0 enable
- Bit 5 – INT2 : eksternal interrupt request 2 enable

e. General Interrupt Flag Register-GIFR

Bit	7	6	5	4	3	2	1	0	
	INTF1	INTF0	INTF2	-	-	-	-	-	GIFR
Read/Write	R/W	R/W	R/W	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- Bit 7 - INT1:eksternal interrupt flag 1 enable
- Bit 6 – INT0 :eksternal interrupt flag 0 enable
- Bit 5 – INT2 : eksternal interrupt flag 2 enable

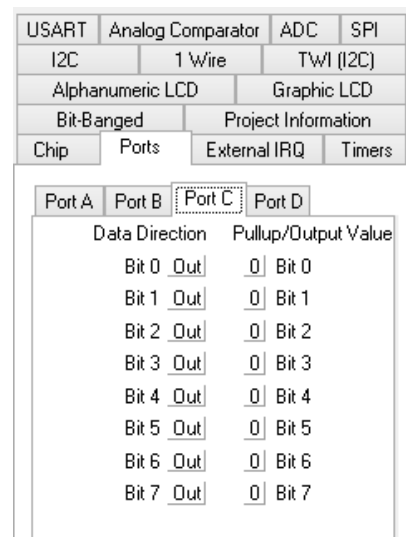
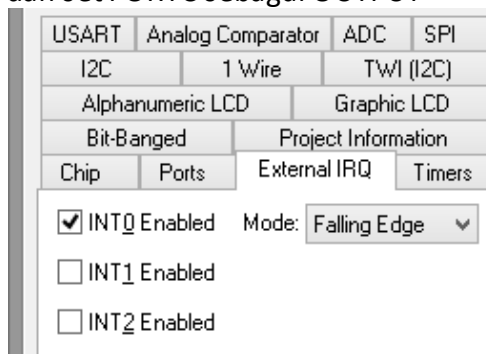
ALAT DAN BAHAN YANG DIGUNAKAN

- 1 set PC/Laptop yang sudah berisi program Code Vision dan Khazama
- 1 buah catu daya DC +5V
- 1 buah multimeter
- 1 buah ISP Downloader AVR
- 1 buah sistem minimum AVR
- 1 buah I/O
- 1 buah kabel printer USB
- 2 buah kabel pita hitam

PROSEDUR

A. MENGGUNAKAN 1 INTERRUPT

1. Hubungkan PORTC pada minimum system dengan soket jumper OUTPUT pada trainer dan PORTD.2 (INT0) dari minimum sistem ke soket IS1 pada trainer.
2. Buka program Code Vision AVR
3. Buatlah project baru. Pada tap External IRQ centang INT0 lalu pilih Mode Falling edge dan set PORTC sebagai OUTPUT



4. Ketik sub routine interrupt seperti sintaks dibawah ini.

```
#include <mega8535.h>
#include <delay.h>
// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
// disini hanya menggunakan 1 interrupt saja
{
// Place your code here
    PORTC = 0b10101010;
    delay_ms(1000);
    PORTC = 0b11111111;
    delay_ms(1000);
}
```

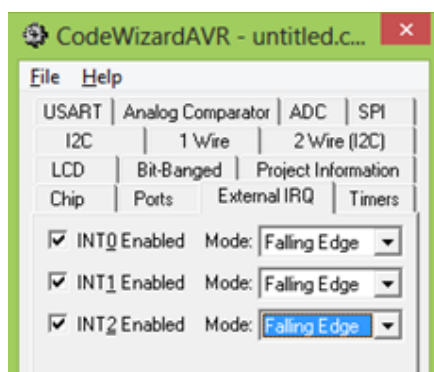
5. Ketik dalam program utama (main program)

```
while (1) // program utama
{
    PORTC = 0b11110000;
}
```

6. Compile dan Build program, jika ada yang error perbaiki program. Masukkan file hex menggunakan Khanzama AVR Programmer. Klik auto program.
7. Amati nyala LED sebelum dan sesudah anda menekan tombol interupsi (lihat tabel interupsi pada data hasil percobaan)
8. Ulangi langkah 3-9 untuk mode interupsi (*Interrupt Sense Control*) Rising Edge, Any Change dan Low Level.

B. MENGGUNAKAN 3 INTERRUPT SECARA BERSAMAAN

1. Hubungkan PORTC pada minimum system dengan soket jumper OUTPUT pada trainer, PORTD.2 (INT0), PORTD.3 (INT1) dan PORTB.2 (INT2) dari minimum sistem ke soket IS1 pada trainer.
2. Buka program Code Vision AVR.
3. Buatlah project baru. Pada tap External IRQ centang INT0 Enabled, INT1 Enabled dan INT2 Enabled. Kemudian pilih Mode Falling Edge untuk semua interrupt dan set PORTC sebagai OUTPUT.



USART		Analog Comparator		ADC		SPI	
I2C		1 Wire		TWI (I2C)			
Alphanumeric LCD				Graphic LCD			
Bit-Banged				Project Information			
Chip		Ports		External IRQ		Timers	
Port A		Port B		Port C		Port D	
Data Direction		Pullup/Output Value					
Bit 0	Out	0	Bit 0				
Bit 1	Out	0	Bit 1				
Bit 2	Out	0	Bit 2				
Bit 3	Out	0	Bit 3				
Bit 4	Out	0	Bit 4				
Bit 5	Out	0	Bit 5				
Bit 6	Out	0	Bit 6				
Bit 7	Out	0	Bit 7				

4. Ketik sub routine interrupt seperti sintaks dibawah ini.

```
#include <mega8535.h>
#include <delay.h>
// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
  // Place your code here
  PORTC = 0b1010101;
  delay_ms(1000);
  PORTC = 0b11111111;
  delay_ms(1000);
}

// External Interrupt 1 service routine
interrupt [EXT_INT1] void ext_int1_isr(void)
{
  // Place your code here
  PORTC = 0b00000001;
  delay_ms(1000);
  PORTC = 0b11111111;
  delay_ms(1000);
}

// External Interrupt 2 service routine
interrupt [EXT_INT2] void ext_int2_isr(void)
{
  // Place your code here
  PORTC = 0b10000000;
  delay_ms(1000);
  PORTC = 0b11111111;
  delay_ms(1000);
}
```

5. Ketik dalam program utama (main program)

```
while (1) // program utama
{
  PORTC = 0b11110000;
}
```

6. Compile dan Build program jika ada yang error perbaiki program. Masukkan file hex menggunakan Khazama AVR Programmer. Klik auto program.
7. Hubungkan INTO, INT1, dan INT2 menjadi satu dengan IS1 pada trainer
8. Amati nyala LED sebelum dan sesudah anda menekan tombol interupsi tombol IS1

DATA HASIL PERCOBAAN

Tabel Interupsi

No	Mode Interupsi	Program yang dijalankan (Interrupt / Main Program)		
		Saat tombol IS1 belum ditekan (Low)	Saat tombol IS1 ditekan dan ditahan (Raising)	Saat tombol IS1 dilepaskan (Falling)
1	Falling Edge			
2	Rising Edge			
3	Any Change			
4	Low Level			

ANALISIS DATA

1. Analisis data hasil pada Tabel Interupsi
2. Ketika ketiga interrupt diaktifkan secara bersamaan tuliskan urutan terjadinya interrupt

