EE-222: Microprocessor Systems

AVR Interrupts:
Programming Timer Interrupts

Instructor: Dr. Arbab Latif

NUST
SCHOOL OF ELECTRICAL ENGINEERING &
COMPUTER SCIENCE (SEECS)

Interrupt Programming in C

What you need for the Interrupt Programming in C?

1. Interrupt include file: `#include <avr\interrupt.h>`
2. Macros to clear and set the I-bit in SREG:

   `cli() and sei()`

3. Defining the ISR: to write an ISR, use the following structure

```
ISR (interrupt vector name)
{
        your program
}
```

Interrupt	Vector Name in WinAVR
External Interrupt request 0	INT0_vect
External Interrupt request 1	INT1_vect
External Interrupt request 2	INT2_vect
Time/Counter2 Compare Match	TIMER2_COMP_vect
Time/Counter2 Overflow	TIMER2_OVF_vect
Time/Counter1 Capture Event	TIMER1_CAPT_vect
Time/Counter1 Compare Match A	TIMER1_COMPA_vect
Time/Counter1 Compare Match B	TIMER1_COMPB_vect
Time/Counter1 Overflow	TIMER1_OVF_vect
Time/Counter0 Compare Match	TIMER0_COMP_vect
Time/Counter0 Overflow	TIMER0_OVF_vect
SPI Transfer complete	SPI_STC_vect
USART, Receive complete	USART0_RX_vect
USART, Data Register Empty	USART0_UDRE_vect
USART, Transmit Complete	USART0_TX_vect
ADC Conversion complete	ADC_vect
EEPROM ready	EE_RDY_vect
Analog Comparator	ANALOG_COMP_vect
Two-wire Serial Interface	TWI_vect
Store Program Memory Ready	SPM_RDY_vect

Example-1: C programming

- Using Timer0 generate a square wave on pin PORTB.5, while at the same time transferring data from PORTC to PORTD.

```
#include "avr/io.h"
#include "avr/interrupt.h"
int main ()
{
    DDRB |= (1<<5);     //DDRB.5 = output
    TCNT0 = -32;        //timer value for 2 µs
    TCCR0 = 0x01;               //Normal mode, int clk, no prescaler
    TIMSK0 = (1<<TOIE0);        //enable Timer0 overflow interrupt
    sei ();                     //enable interrupts
    DDRC = 0x00;                //make PORTC input
    DDRD = 0xFF;                //make PORTD output
    while (1)           //wait here
        PORTD = PINC;
}
ISR (TIMER0_OVF_vect)  //ISR for Timer0 overflow
{
    TCNT0 = -32;
    PORTB ^= 0x20;      //toggle PORTB.5
}
```

Example-2:C programming

- Using Timer1 and CTC mode write a program that toggles pin PORTB.5 every second, while at the same time transferring data from PORTC to PORTD. Assume XTAL = 16 MHz.

```
#include <avr/io.h>
#include <avr/interrupt.h>
int main () {
        DDRB |= (1<<5);          //make DDRB.5 output

        OCR1A = 15624;
        TCCR1A = 0x00; //CTC mode, internal clk, prescaler=1024
        TCCR1B = 0x0D;
        TIMSK1 = (1<<OCIE1A);  //enable Timer1 compare match A int.
        sei ();                  //enable interrupts

        DDRC = 0x00;             //make PORTC input
        DDRD = 0xFF;             //make PORTD output
        while (1)                //wait here
                PORTD = PINC;
}
ISR (TIMER1_COMPA_vect) {      //ISR for Timer1 compare match A
        PORTB ^= (1<<5);         //toggle PORTB.5
}
```

Recommended Reading

- The AVR Microcontroller and Embedded Systems: Using Assembly and C by Mazidi et al., Prentice Hall
  - Chapter 10

THANK YOU