

การอินเตอร์เฟสกับอุปกรณ์ I/O

รศ.ณรงค์ บวบทอง

1

วัตถุประสงค์

- เพื่อให้มีความรู้เกี่ยวกับการอินเตอร์เฟสกับอุปกรณ์ I/O
- สามารถเขียน โปรแกรมติดต่อกับอุปกรณ์ I/O ได้

2

Peripherals อุปกรณ์ต่อพ่วง

Peripherals หรือที่แปลว่าอุปกรณ์ต่อพ่วง เป็นอุปกรณ์เสริมในระบบ เช่น วงจรนับ วงจรนาฬิกา ระบบสื่อสาร

ในระบบ Microprocessor อุปกรณ์ต่อพ่วงจะอยู่นอกชิพ

ในระบบ Microcontroller อุปกรณ์ต่อพ่วง จะถูกสร้างรวมอยู่ในชิพ (system on chip).

3

การใช้งาน Peripherals

- ในระบบที่ไม่มี OS อุปกรณ์ต่อพ่วงเหล่านี้สามารถเข้าถึงได้โดยตรง (เรียกใช้งานได้โดยตรง)
- แต่ในระบบที่มี OS การเข้าถึงอุปกรณ์เหล่านี้ มักกระทำโดยอ้อม โดยผ่านทาง Device driver
- Configure peripheral bus.
 - Prescaler value for I/O clock (system clock is usually faster than peripheral clock).
- Configure peripheral behaviors (I/O registers).
 - Prescaler value for internal clock.
 - Mode of operations.
 - Interrupt source and conditions (option).
- Select pins required by peripheral.
 - Activate peripheral to start.
 - Passive mode: Polling read or write.
 - Active mode: Interrupt with ISR (Interrupt Service Routine) invoked by hardware IRQ (Interrupt Request).

4

รีจิสเตอร์ I/O

I/O register stores information bits that can be written to or read out simultaneously.

- Access via memory-mapped I/O address in memory map.
- Use load/store or in/out to access in word. Or specific instructions to access flags in bits.

ตัวอย่าง AVR Two-Wire registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
(0xBF)	Reserved	-	-	-	-	-	-	-	-
(0xBE)	Reserved	-	-	-	-	-	-	-	-
(0xBD)	TWAMR	TWAM6	TWAM5	TWAM4	TWAM3	TWAM2	TWAM1	TWAM0	-
(0xBC)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
(0xBB)	TWDR	2-wire Serial Interface Data Register							
(0xBA)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE
(0xB9)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	-	TWPS1	TWPS0
(0xB8)	TWBR	2-wire Serial Interface Bit Rate Register							



5

รีจิสเตอร์ I/O (ต่อ)

ตัวอย่างคำสั่งที่เกี่ยวกับรีจิสเตอร์ I/O

IN, OUT = move data with I/O registers.

LD, STD = move data with extended I/O registers.

SBI, CBI = set/clear I/O register bit.

SEI, CLI = set/clear global interrupt flag.

สามารถใช้ได้ทั้งภาษา C และแอสเซมบลี

C code needs hardware-dependent header files.

Use bitwise operations to access I/O registers' flags.

Use specific instruction or macro to access status register.



6

ตัวอย่างการใช้คำสั่ง Bitwise ด้วยภาษา C

Create bit pattern using shift operators or hexadecimal value.

```
#define ADSC (1 << 6)
#define REFS ( (1 << 7) | (1 << 6) )
#define BIT0 (1 << 0)
```

Test bits: single bit, one of bits, or all bits

```
if ( ADCSRA & ADSC )
if ( PINB | (BIT2 | BIT4 | BIT6) )
if ( PINB & (BIT2 | BIT4 | BIT6) )
```

Extract bits: draw some bits in usable form

```
uint8_t lread = ( ADCL & (BIT7 | BIT6) ) >> 6;
uint16_t readout = ((ADCH & (BIT1 | BIT0)) << 8) | ADCL;
```



7

ตัวอย่างการใช้คำสั่ง Bitwise ด้วยภาษา C (ต่อ)

Set bits: set one or more bits to 1

```
#define ADSC (1 << 6)
ADMUX |= 0x03;
ADCSRA |= ADSC;
```

Clear bits: clear one or more bits to 0

```
#define ADEN (1 << 7)
ADCSRA &= ~ADEN;
```

Toggle bits: convert bit status between 0 and 1

```
PINB ^= (1 << 7) | (1 << 5) | (1 << 3);
```



8

ตัวอย่างโปรแกรม

```
#include <avr/io.h>
#define FOSC 1843200
#define BAUD 9600
#define MYUBRR (FOSC/16/BAUD-1)

/* Set baud rate */
void USART_Init( unsigned int ubrr)
{
    UBRR0H = (unsigned char)(ubrr>>8);
    UBRR0L = (unsigned char)ubrr;
    /* Enable receiver and transmitter */
    UCSR0B = (1<<RXEN0)|(1<<TXEN0);
    /* Frame format: 8 data, 2 stop bits */
    UCSR0C = (1<<USBS0)|(3<<UCSZ00);
```



9

ตัวอย่างโปรแกรม (ต่อ)

Reading or writing I/O registers with larger data width may have to disable interrupt before.

ภาษา C

```
unsigned int TIM16_ReadTCNT1( void )
{
    unsigned char sreg; /* 8-bit variable */
    unsigned int i; /* 16-bit variable */
    sreg = SREG; /* Save global interrupt flag */
    _CLI(); /* Disable interrupts */
    i = TCNT1; /* Read TCNT1 into i */
    SREG = sreg; /* Restore global interrupt flag */
    return i;
}
```



10

ตัวอย่างโปรแกรม (ต่อ)

ภาษา Assembly

```
TIM16_ReadTCNT1:
    in     r18,SREG      ; Save global interrupt flag
    cli                    ; Disable interrupts
    in     r16,TCNT1L    ; Read TCNT1 into r17:r16
    in     r17,TCNT1H
    out    SREG,r18      ; Restore global interrupt flag
    ret
```



11

Talking to Peripherals

มี 2 วิธี ที่ใช้ทำงานกับอุปกรณ์ I/O

1. Polling (Passive mode) - processor periodically checks peripherals for any data to be exchanged.
2. Interrupt (Active mode) - processor is notified only when peripherals need to exchange data with processor.



12

Polling

ข้อดี

ง่ายต่อการเขียนโปรแกรม

ตรวจสอบ (debug) ได้ง่าย

ข้อเสีย

เปลืองเวลาการทำงานของซีพียู

ถ้ามีอุปกรณ์เป็นจำนวนมากจะทำให้เกิดการหน่วงเวลามาก

ตอบสนองช้า



13



Interrupt

ข้อดี

ใช้ซีพียูได้อย่างมีประสิทธิภาพ

ตอบสนองได้อย่างรวดเร็ว

ข้อเสีย

เขียนโปรแกรมยาก

ตรวจสอบ (debug) ได้ยาก



14

