

วิชา Data Communication Laboratory

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

การทดลองที่ 3 Programming with asynchronous terminal

วัตถุประสงค์

1. เพื่อให้เข้าใจหลักการสื่อสารผ่านพอร์ตอนุกรม
2. เพื่อให้เข้าใจลักษณะสัญญาณ และการเชื่อมต่อพอร์ตอนุกรม
3. สามารถเขียน โปรแกรมเพื่อติดต่อสื่อสารระหว่างคอมพิวเตอร์ผ่านพอร์ตอนุกรมได้

ทฤษฎี

ในการเขียนโปรแกรม เพื่อติดต่อสื่อสารระหว่างคอมพิวเตอร์ผ่านพอร์ตอนุกรม จำเป็นต้องมีการกำหนดหมายเลขพอร์ทที่จะใช้งาน โดยกำหนดจากตำแหน่งที่ระบุไว้ รวมถึงกำหนดค่าคุณสมบัติการเชื่อมต่อพอร์ตอนุกรมให้ถูกต้องในการติดต่อสื่อสาร และการสื่อสารโดยพื้นฐานคือการส่ง-รับข้อมูลจากการทดลองที่ผ่าน มาจะพบว่าการส่งข้อมูลอนุกรมแบบอะซิงโครนัสจะทำการส่งข้อมูลที่ละไบต์ ซึ่งในการเขียนโปรแกรมต้อง จัดการกับรีจิสเตอร์ในการสื่อสารผ่านพอร์ตอนุกรมหลัก 2 ตัว ได้แก่ Line Control Register (LCR) และ Line Status Register (LSR) โดยที่ Line Control Register จะเป็นที่ใช้กำหนดค่าคุณสมบัติการเชื่อมต่อพอร์ตอนุกรม และ Line Status Register จะเป็นที่ใช้จัดการและตรวจสอบเกี่ยวกับบัพเฟอร์ตัวส่งและรับ

ตำแหน่งของพอร์ตอนุกรมต่างๆ

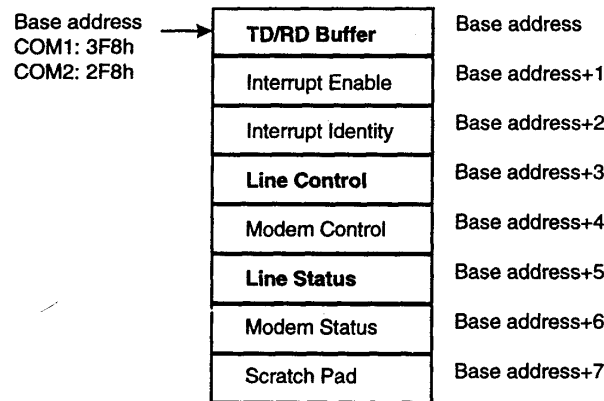
ตำแหน่งของ Primary Port (Com1) โดยทั่วไปกำหนดไว้ที่ 3F8h และ Secondary Port (Com2) กำหนดไว้ที่ 2F8h ตำแหน่งดังกล่าวจะถูกกำหนดค่าในหน่วยความจำ BIOS และตำแหน่งของพอร์ทอื่นๆ แสดงดังตารางที่ 3.1 โดยที่แต่ละพอร์ทจะมีค่าตำแหน่ง และอินเตอร์รัพเป็นค่าเฉพาะตัวเพื่อไม่ให้สับสนในการติดต่อกับอุปกรณ์อื่น

ตารางที่ 3.1 แสดงค่าตำแหน่งและ IRQ ของแต่ละ COM Port ต่างๆ

NAME	ADDRESS	IRQ
COM1	3F8	4
COM2	2F8	3
COM3	3E8	4
COM4	2E8	3

รีจิสเตอร์ในการสื่อสารผ่านพอร์ตอนุกรม

รีจิสเตอร์หลักที่ใช้สำหรับการสื่อสารทางพอร์ตอนุกรม RS-232 คือ Line Control Register (LCR) กับ Line Status Register (LSR) และ บัพเฟอร์ตัวส่งและรับแสดง โดยจะใช้ตำแหน่งของ COM Port เป็น Base Address ในการอ้างอิง Register ดังรูปที่ 3.1 โดยที่รายละเอียดหน้าที่รีจิสเตอร์ที่ตำแหน่งต่างๆ เป็นดังตารางที่ 3.2



รูปที่ 3.1 รีจิสเตอร์หลักในการสื่อสารทางพอร์ตอนุกรม

ตารางที่ 3.2 แสดงตำแหน่งของรีจิสเตอร์ต่างๆที่เกี่ยวข้องกับ COM Port

Base Address	DLAB	Read/Write	Abr.	Register Name
+0	=0	Write	-	Transmitter Holding Buffer
	=0	Read	-	Receiver Buffer
	=1	Read/Write	-	Divisor Latch Low Byte
+1	=0	Read/Write	IER	Interrupt Enable Register
	=1	Read/Write		Divisor Latch High Byte
+2	-	Read	IIR	Interrupt Identification Register
	-	Write	FCR	FIFO Control Register
+3	-	Read/Write	LCR	Line Control Register
+4	-	Read/Write	MCR	Modem Control Register
+5	-	Read	LSR	Line Status Register
+6	-	Read	MSR	Modem Status Register
+7	-	Read/Write	-	Scratch Register

Line Control Register (LCR)

LCR เป็นรีจิสเตอร์ในการกำหนดค่าคุณสมบัตการเชื่อมต่อพอร์ตอนุกรมพื้นฐานในการสื่อสาร หากต้องการกำหนดอัตราบอดต้องกำหนดค่า MSB (C_7) ที่ตำแหน่ง LCR ให้เป็น '1' แล้วทำการคำนวณตัวหาร (Divisor) ขนาด 16 บิต เพื่อกำหนดค่าในตำแหน่งรีจิสเตอร์ Divisor Latch Low Byte และ Divisor Latch High Byte (ตารางที่ 3.2) โดยค่าที่จะทำการโหลดนั้นขึ้นอยู่กับความถี่ของคริสตัลที่ต่ออยู่กับ IC อัตราบอดสามารถคำนวณได้ดังนี้

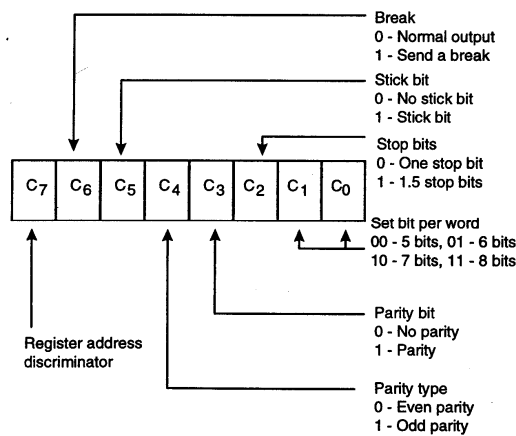
$$\text{Baud rate} = \frac{\text{clock frequency}}{16 \times N}$$

ตัวอย่าง เช่น ต้องการส่งข้อมูล 9600 บอด สามารถคำนวณหาค่า N (Divisor) ได้เท่ากับ $1.8432 \times 10^6 / (9600 \times 16) = 12$ (ค่า 000Ch) ดังตารางที่ 3.3 ที่แสดงถึงการกำหนดอัตราบอดค่าต่างๆ

ตารางที่ 3.3 แสดงค่าที่ใช้ในการกำหนดอัตราบอด

Speed (BPS)	Divisor (Dec)	Divisor Latch High Byte	Divisor Latch Low Byte
50	2304	09h	00h
300	384	01h	80h
600	192	00h	C0h
2400	48	00h	30h
4800	24	00h	18h
9600	12	00h	0Ch
19200	6	00h	06h
38400	3	00h	03h
57600	2	00h	02h
115200	1	00h	01h

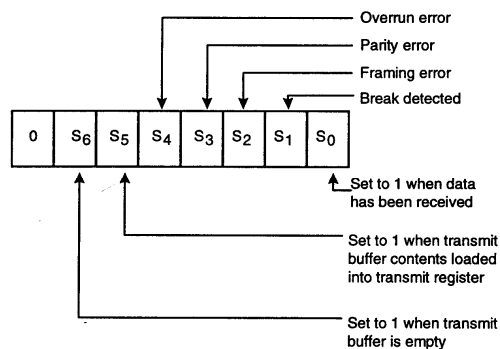
และหากต้องการกำหนดจำนวนของบิตต่อตัวอักษร จำนวนพาริตีบิต จำนวนของ stop bit บิต ต้องกำหนด MSB (C_7) ให้เป็น '0' และบิตอื่นๆ ดังรูปที่ 3.2



รูปที่ 3.2 รีจิสเตอร์ LCR

Line Status Register (LSR)

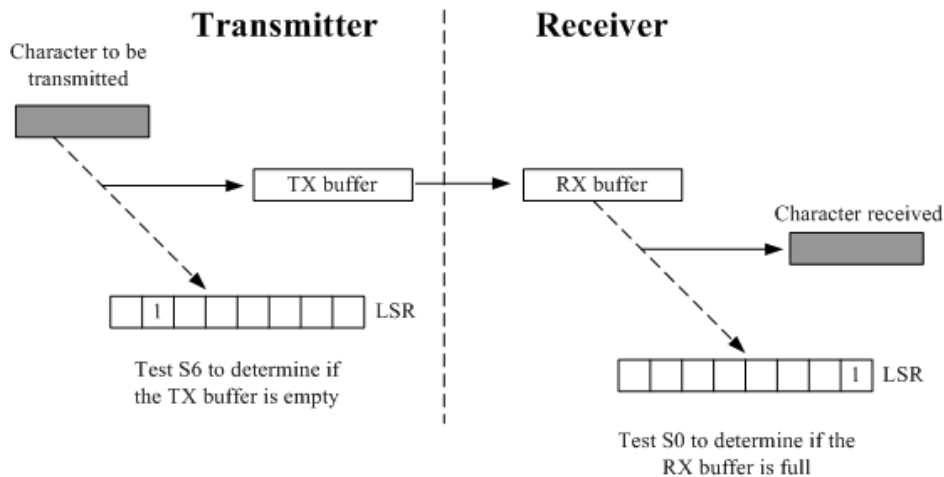
LSR เป็นรีจิสเตอร์ในการตรวจสอบสถานะของการบัพเฟอร์ส่งและรับ ซึ่งสามารถตรวจสอบสถานะต่างๆ ได้จากบิตข้อมูลแต่ละบิตใน LSR ดังรูปที่ 3.3



รูปที่ 3.3 รีจิสเตอร์ LSR

ในการส่งข้อมูลแต่ละครั้งต้องมีการตรวจสอบความพร้อมในการรับส่งข้อมูล เพื่อป้องกันการทับกันของข้อมูลในบัฟเฟอร์ โดยทำการตรวจสอบที่บิต S_6 ของ LSR เมื่อบิตนี้เป็น '1' หมายความว่าบัฟเฟอร์ว่างสามารถส่งข้อมูลออกที่พอร์ตได้

และในการรอรับข้อมูลจะตรวจสอบว่ามีข้อมูลส่งเข้ามาที่พอร์ตหรือไม่ โดยทำการตรวจสอบที่บิต S_0 ของ LSR เมื่อบิตนี้เป็น '1' แสดงว่ามีข้อมูลส่งมาที่พอร์ต ดังรูปที่ 3.4



รูปที่ 3.4 แสดงค่าในรีจิสเตอร์ LSR ในกรณีที่มีการรับและส่งข้อมูล

การเขียนโปรแกรมผ่านพอร์ตอนุกรม

ในการเขียนโปรแกรมด้วยภาษา C/C++ เพื่อติดต่อผ่านพอร์ตอนุกรมจะ ใช้ฟังก์ชัน `inportb()` ในการอ่านค่า หรือค่าสถานะจากพอร์ต และใช้ฟังก์ชัน `outportb()` ในการส่งค่าไปที่พอร์ตหรือกำหนดค่าการทำงานที่พอร์ต โดยฟังก์ชันนี้เป็นโปรโตไทป์ในเฮดเดอร์ไฟล์ `dos.h` สามารถใช้งานดังนี้

```
value = inportb(PORTADDRESS);

outportb (PORTADDRESS, value);
```

เมื่อ PORTADDRESS เป็นตำแหน่งของพอร์ตอนุกรม / เอาท์พุท

value ขนาด 8 บิต ถูกโหลดจากตำแหน่งที่เลือก / ส่งไปตำแหน่งที่เลือก

ขั้นตอนในการกำหนดค่าพอร์ตอนุกรม

1. กำหนดค่าตำแหน่งของ COM Port เป็น Base Address ในการอ้างอิง Register
2. ตั้งค่าอัตรารอบอดโดยกำหนดค่า MSB (C_7) ของ Line Control Register (LCR) ที่ Base Address + 3 (จากรูปที่ 3.1) เป็น '1' เพื่อเตรียมกำหนดค่า DLAB (Divisor Latch Access bit)
 - 2.1. กำหนดค่าตัวหาร (Divisor) ของ DLAB ตามที่ต้องการ
 - 2.2. กำหนดค่า Divisor Latch Low Byte ที่ Base Address (ตารางที่ 3.3)
 - 2.3. กำหนดค่า Divisor Latch High Byte ที่ Base Address + 1 (ตารางที่ 3.3)
3. กำหนดค่าอื่นๆ ของ Line Control Register (LCR) ที่ Base Address + 3 ตามรูปที่ 3.2

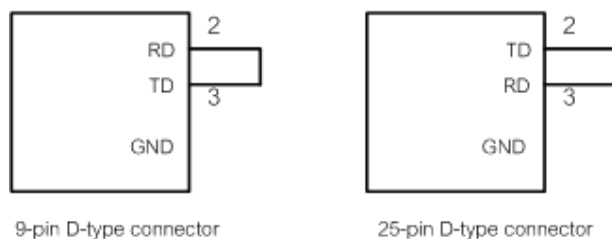
การทดลองที่ 3.1 ทดสอบเขียนโปรแกรมแสดงผลค่าตำแหน่งของพอร์ท

1. เขียนโปรแกรมแสดงผลค่าตำแหน่งของพอร์ทต่างๆ

```
#include <stdio.h>
#include <dos.h>
void main(void)
{
    unsigned int far *ptraddr;
    /* Pointer to location of Port Addresses */
    unsigned int address; /* Address of Port */
    int a;
    ptraddr=(unsigned int far *)0x00000400;
    for (a = 0; a < 4; a++)
    {
        address = *ptraddr;
        if (address == 0)
            printf("No port found for COM%d \n",a+1);
        else
            printf("Address assigned to COM%d is %Xh\n",a+1,address);
        *ptraddr++;
    }
}
```

การทดลองที่ 3.2 ทดสอบเขียนโปรแกรมส่งข้อมูลในคอมพิวเตอร์เครื่องเดียวกัน

1. ให้ทำการเชื่อมต่อขาของคอนเนคเตอร์ดังรูปที่ 3.6 จากนั้นต่อเข้ากับเครื่องคอมพิวเตอร์



รูปที่ 3.6 การเชื่อมต่อขาแบบ 9-ขา และ 25-ขา

2. เขียนโปรแกรมเพื่อส่งข้อมูลภายในเครื่องเดียวกัน ดังนี้

```
/*This program transmits and receives a character from COM3*/
/*it via this port. The TD is connected to RD*/
#define COM3BASE 0x3E8
#define COM4BASE 0x2E8
#define TXDATA COM3BASE
#define LCR (COM3BASE+3) /*0x3F8 line control*/
#define LSR (COM3BASE+5) /*0x3FD line status */
#include <conio.h>
#include <dos.h>
#include <stdio.h>
/*Some ANSI C prototype definitions*/
void setup_serial(void);
void send_character(int ch);
int get_character(void);
```

```

int main(void)
{
    int inchar, outchar;
    setup_serial();
    do
    {
        puts("Enter char to be transmitted (Ctrl-D to end)");
        outchar = getch();
        printf("Character transmitted is %c\n", outchar);
        send_character(outchar);
        inchar = get_character();
        printf("Character received was %c\n", inchar);
    } while(outchar != 4);
    return(0);
}

void setup_serial(void)
{
    outportb(LCR, 0x80);
    /*set up bit 7 to a 1 to set Register address bit*/
    outportb(TXDATA, 0x0C);
    outportb(TXDATA+1, 0x00);
    /*load TxRegister with 12, crystal frequency is 1.8432 MHz*/
    outportb(LCR, 0x0A);
    /*Bit pattern loads is 00001010b, from MSB to LSB these are: */
    /* 0 - access TD/RD buffer, 0 - normal output */
    /* 0 - no stick bit, 0 - even parity */
    /* 1 - parity on, 0 - 1 stop bit */
    /* 10 - 7 data bits */
}

void send_character (int ch)
{
    char status;
    do{
        status = inportb(LSR) & 0x40;
    } while ( status!=0x40);
    /*repeat until Tx buffer empty ie bit 6 set */
    outportb(TXDATA, (char) ch);
}

int get_character(void)
{
    int status;
    do{
        status = inportb(LSR) & 0x01;
    }while (status!=0x01);
    /*Repeat until bit 1 in LSR is set */
    return( (int) inportb(TXDATA));
}

```

3. ทำการส่งตัวอักษรใดๆ โดยการกดแป้นพิมพ์และสังเกตว่าอักษรที่รับได้ถูกต้องหรือไม่

การทดลองที่ 3.3 เขียนโปรแกรมส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์

1. ให้เขียนโปรแกรมเพื่อส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ ที่มีลักษณะดังนี้

โปรแกรมการส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์

```
// send.c
void setup_serial(void);
void send_character(int ch);

int main(void)
{
    Enter char to be transmitted (Ctrl-D to end)
    Transmitted character is ?
}

void setup_serial(void)
{
    Set up bit 7 to a 1 to set Register address bit
    Bit pattern loaded is 00001010b from MSB to LSB these are?
    Access TD/RD buffer, normal output, no stick bit
    Even parity, parity on, 1 stop bit, 7 data bits
}

void send_character(int ch)
{
    Send character to remote computer
    Repeat until Tx buffer empty ie bit 6 set
}
```

โปรแกรมรับข้อมูลระหว่างเครื่องคอมพิวเตอร์

```
// receive.c
void setup_serial(void);
void get_character(void);
int main(void)
{
    Received character is ?
}

void setup_serial(void)
{
    Set up bit 7 to a 1 to set Register address bit
    Load TxRegister with 12, crystal frequency is 1.8432 MHz
    Bit pattern loaded is 00001010b from MSB to LSB these are?
    Access TD/RD buffer, normal output, no stick bit
    Even parity, parity on, 1 stop bit, 7 data bits
}

int get_character(void)
{
    Get character
    Repeat until bit 1 in LSR is set
}
```

2. ทำการเชื่อมต่อคอมพิวเตอร์ 2 เครื่องเข้าด้วยกัน
3. ทำการส่งตัวอักษรใดๆ โดยการกดแป้นพิมพ์และสังเกตว่าอักษรที่รับได้ถูกต้องหรือไม่
4. เปลี่ยนอัตราบอดและพารามิเตอร์ต่างๆ ดังต่อไปนี้

อัตราบอด	จำนวนบิตข้อมูล	พาริตีบิต	ชนิดของพาริตี	บิตหยุด(บิต)
2400	7	ไม่มี	-	2
4800	7	ไม่มี	-	2
9600	8	มี	พาริตีคู่	2
19200	8	มี	พาริตีคู่	2

การทดลองที่ 3.4 เขียนโปรแกรมรับ-ส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์

1. เขียนโปรแกรมรับ-ส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์โดยที่
 - 1.1. ต้องเลือกว่าจะเป็นเครื่องที่ส่ง (S) หรือรับ (R) ก่อน
 - 1.2. สลับกันรับส่งข้อความ
 - 1.3. เมื่อฝั่งส่งต้องการจบการติดต่อให้กด Ctrl+Q เพื่อจบการทำงานทั้ง 2 ฝั่ง

```
Send or Receive : S
Pc1 : Test
Pc2 : Computer Engineering
Pc1 : _
```

```
Send or Receive : R
Pc1 : Test
Pc2 : Computer Engineering.↓
```

2. เชิญอาจารย์ตรวจการทดลอง

.....
ลายเซ็นอาจารย์ผู้ตรวจการทดลอง

คำถามท้ายการทดลอง

1. จงอธิบายขั้นตอนการกำหนดอัตราการส่งและการกำหนดอะซิงโครนัสเฟรมในการโปรแกรมการรับส่งระหว่างเครื่องคอมพิวเตอร์อย่างละเอียด
2. ถ้าต้องการให้คอมพิวเตอร์ส่งข้อมูลที่มีพาริตีคู่ มี 1 stop bit มีข้อมูลที่จะส่งขนาด 8 บิตตัวหนึ่งตัวอักษร จะต้องกำหนดค่าในรีจิสเตอร์ LCR เป็นค่าเท่าใด