การส่งข้อมูลแบบ asynchronous

↳ ไม่ใช้ clock , แต่จะมีสัญญาณเริ่มส่ง low ตามด้วยข้อมูล

↳ ใช้ RS232 { ขา 2 Tx, ขา 3 Rx, ขา 5 G } ใช้แค่นี้พอ

TD/RD Buffer	Base add
Interupt Enable	,, +1
Interupt Identity	,, +2
Line Control	,, +3
Modem Control	,, +4
Line status	,, +5
Modem status	,, +6
Scratch Pad	,, +7

↳ เราต้องกำหนดการเลือกพอร์ตที่ใช้งาน
- ↳ เช่น com1 : 3F8h
  com2 : 2F8h

↳ ต้องมี 2 ค่าที่สำคัญที่ใช้เชื่อมต่อ com port
- ↳ LCR (line control register)
  - ↳ ควบคุมการทำงาน → ความเร็ว parity ไหม ...
- ↳ LSR (line status register)
  - ↳ เช็คว่า port → สามารถส่งได้ไหม , มี input เข้าไหม

เช่น com1 มี address base อยู่ที่ 3F8h

จากตัวอย่างการกำหนดค่า LCR →

TD/RD Buffer	Base add
Interupt Enable	,, +1
Interupt Identity	,, +2
Line Control	,, +3
Modem Control	,, +4
Line status	,, +5
Modem status	,, +6
Scratch Pad	,, +7

syntax การอ่านค่า หรือ เขียนค่า

```
value = inportb(portAddress);  # อ่านค่าที่ portAddress นี้
outportb(portAddress, value);
```

↳ LCR (line control register)
- ↳ คุมความเร็วการส่ง

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

\# ในวงจร
\# ค่าที่เก็บใน LCR
N ↳ what we want

$$N = \frac{\text{clock}}{16 \times \text{Baud}} = $$ → ค่าที่ได้ต้องเป็นเลขฐาน 16

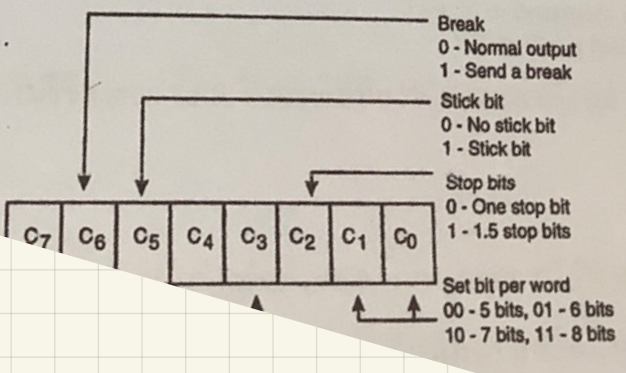
ex ได้ 12 → C

| 0 | 0 | 0 | C | x
high Byte, Low Byte

งค่าที่ใช้ในการกำหนดอัตราบอด

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

และ<u>หากต้องการกำหนดจำนวนของบิตต่อตัวอักษร จำนวนพาริตี้บิต จำนวนของ stop bit บิต</u> ต้อง
าหนด MSB (C_7) ให้เป็น '0' และบิตอื่นๆ ดังรูปที่ 3.2

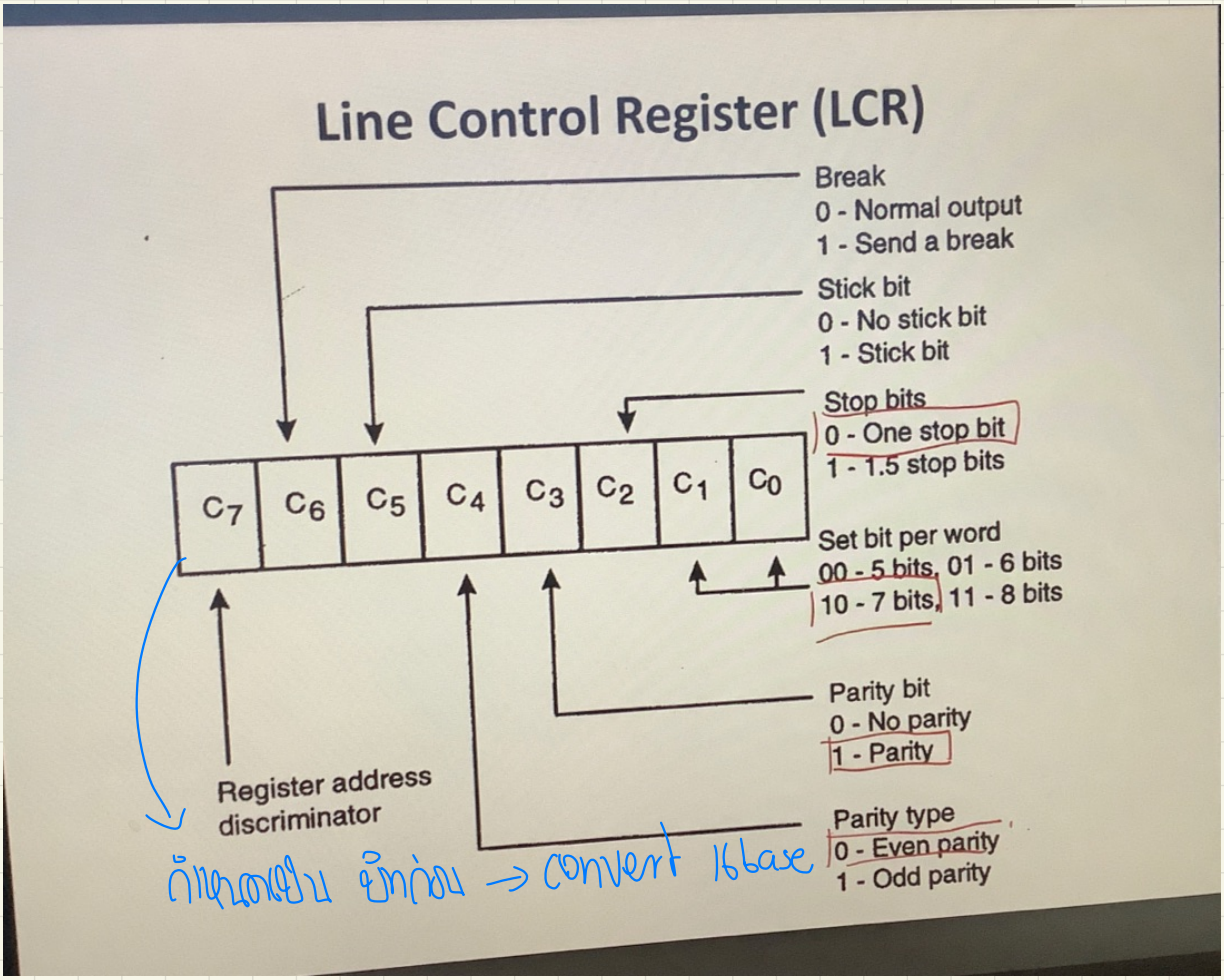


DLAB : Divisor Latch Access bit

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

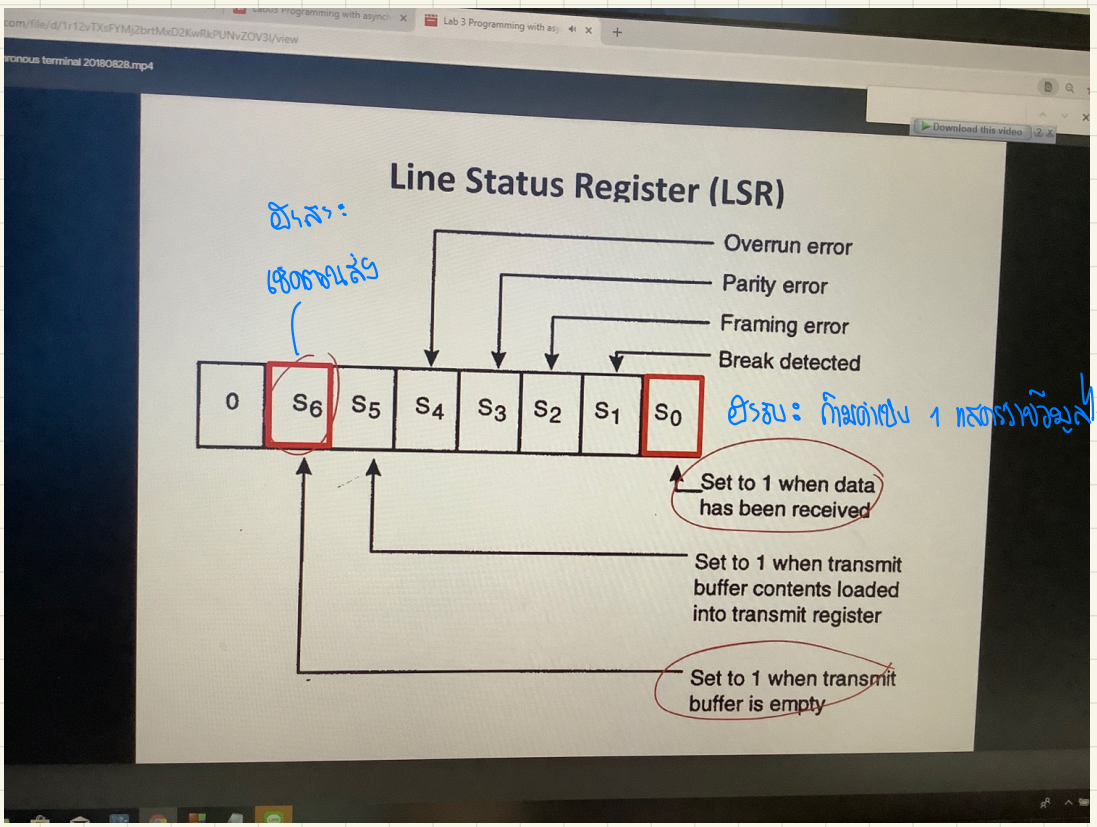
LCR ก็ออกแบบการเอง

Line Control Register (LCR)

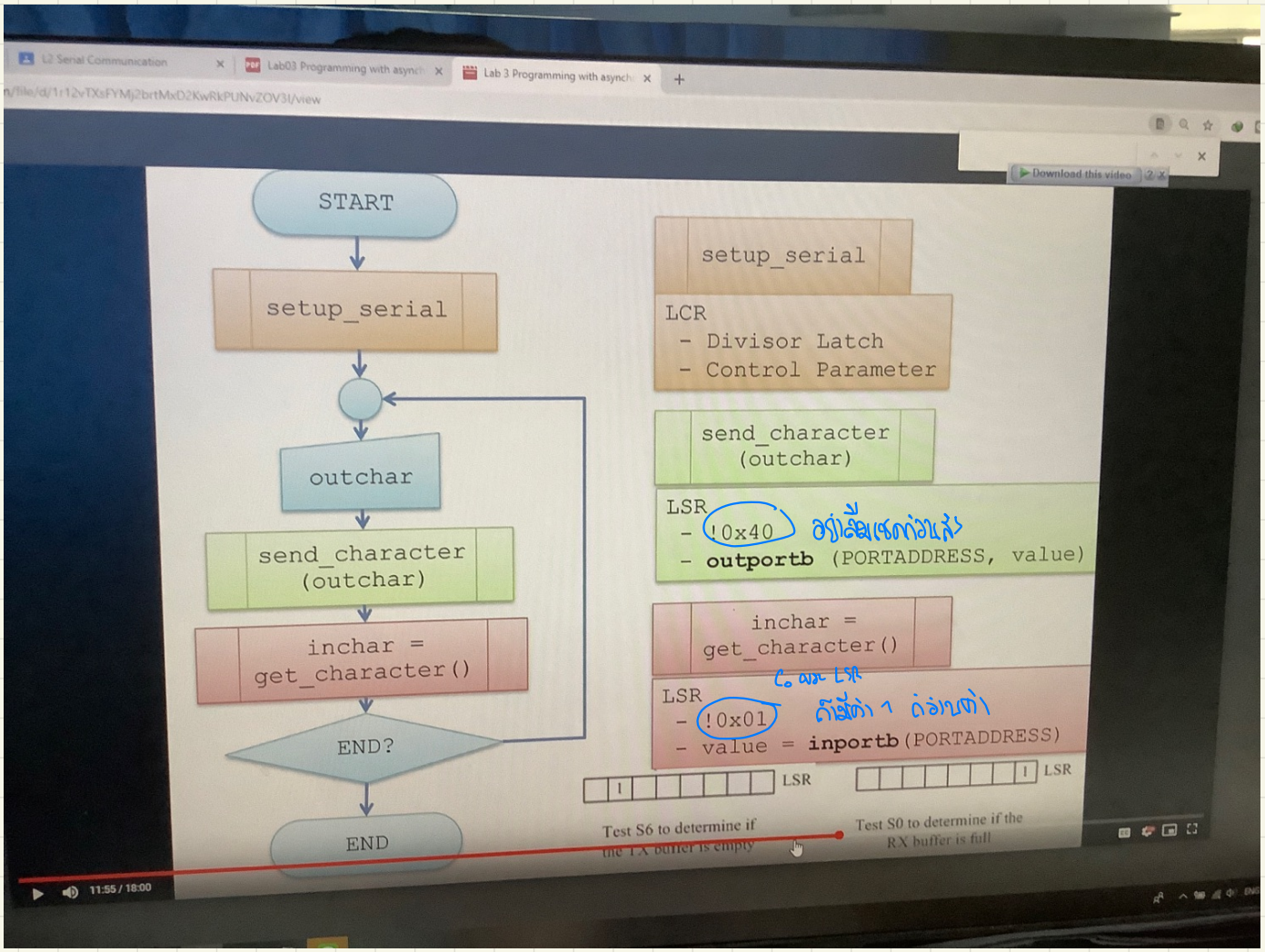


Line status Register

↳ Base address +5



ใช้ Turbo c 3.0 รันใน winxp. ong.



```c
#define combase 0x3F8
#define txData combase
#define LCR (combase +3)
#define LSR (combase +5)
void setupSerial(void){
        outportb(LCR , 0x80);
        outportb( txData , 0X0C);
        outportb (txData+1,0x00);
        outportb (LCR , 0x0A);
}
void sendCharacter (int ch){
      char status;
      do{
           status = inport (LSR) & 0X40;
      }while ( status != 0X40)
      outportb(txData ,(char)ch);
}
int getCharacter(void){
      int status;
      do{
           status = inportb (LSR)& 0X01;
      }while(status != 0X01);
      return (int)inportb(txData);
}

int main(void){
      setupSerial();
      // your code heer
```

1000 0000

ก่อนจะเซตค่า LCR ต้องทำให้บิตที่ 7 ของ LCR = 1 ก่อน

} เซ็ตความเร็ว

กำหนดค่าตามที่ต้องการ

0100000 → ถ้า bit ที่ 6 เป็น 1 ถือว่าพร้อมส่ง

0000 0001 → เป็น = 1 แสดงว่ามีคนส่งข้อมูลมา

```c
#define com3 0x3E8
# define txData com3
# define LCR (com3+3)
# define LSR (com3+5)
# include <conio.h>
# include <dos.h>
# include <stdio.h>
# define enter 13
# define ctrlQ 7
#define backspace 8
```

```c
void setupSerial(void){
    outportb(LCR, 0x80);
    outportb(txData, 0x00);
    outportb(txData+1, 0x0C);
    outport(LCR, 0x1F);
}
```

```c
void sendCharacter(int ch){
    char status;
    do{
        status = inportb(LSR)&0x40;
    }while(status!=0x40);
    outport(txData, (char)ch);
}
```

```c
int getCharacter(void){
    int status;
    do{
        status = inportb(LSR)&0x01;
    }while(status!=0x01);
    return((int) inportb(txData));
}
```

```c
int main(void){
    int turn;
    char name[2][4] = {{"tae"},{"big"}};
    char word[100];
    setChoose(&turn);
    while(10){
        printf("%s :", name[turn%2]);
        if(turn%2==0){ #received
            ...
        }else
            setFunction(word, &turn);
    }
}
```

```c
void sendString(char *string, int size){
    sendCharacter(size);
    for(int i=0; i<size; i++, string++)
        sendCharacter(*string);
}
int getTurn(char outChar, char inChar){
    if(outChar=='s'&& inChar=='R')
        return 1;
    else
        return 0;
}
void setChoose(int *turn){ # pass &turn
    int outChar, inChar;
    puts("Send or Receive: ");
    outChar = getchar();
    sendCharacter(outChar);
    inChar = getCharacter();
    if(outChar==inChar)
        setChoose(&turn);
    else
        *turn = getTurn(inChar, outChar);
}
void setFunction(char *word, int *turn){
    int outChar = getche();
    switch(outChar){
        case backspace:
            deleteCharScreen(word);
            breale;
        case enter: sendCharacter(outChar);
            sendString(word, strlen(word));
            strcpy(word, ""); *turn++;
            breale;
        case ctrlQ:
            sendCharacter(outChar);
            exit();
            breale;
        default:
            word+=strlen(word);
            *word = outChar;
            word++;
            *word = '\0'; }
}
```

```c
void deleteCharScreen(char *word){
    if(strlen(word)){
        strncpy(word, word, strlen(word)-1);
        printf("\b \b");
    }
}
```

```
#define com3  0x3E8
# define txData  com3
# define LCR  (com3+3)
# define LSR  (com3+5)
# include <conio.h>
# include <dos.h>
# include <stdio.h>
# define enter 13
# define ctrlQ 4
#define backspace 8
```

```
void setupSerial(void){
    outportb(LCR, 0x80);
    outportb(txData, 0x00);
    outportb(txData+1, 0x0C);
    outport(LCR, 0x1F);
}
```

```
void sendCharacter(int ch){
    char status;
    do{
        status = inportb(LSR)&0x40;
    }while(status!=0x40);
    outport(txData, (char)ch);
}
```

```
int getCharacter(void){
    int status;
    do{
        status = inportb(LSR)&0x01;
    }while(status!=0x01);
    return((int) inportb(txData));
}
```

```
int main(void){
    int turn, inChar, outChar;
    char name[2][4] = {{"big"},{"tae"}};
    char word[100];
    setChoose(&turn);
    while(10){
        printf("%s :", name[turn%2]);
        if(turn%2==0){ #received
            ...
        }else{
            outChar = getche();
            if(outChar == backspace){
                if(strlen(word)){
                    strncpy(word, word, strlen(word)-1);
                    printf("\b \b");
                }
            }else if(outChar == enter){
                turn++;
                sendString(word, strlen(word));
            }else if(oneChar = ctrlQ){
                sendCharacter(oneChar);
                return 0;
            }else{
                word[strlen] = outChar;
                word[strlen] = '\0';
            }
        }
    }
```

```
void sendString(char *c, int size){
    for(int i=0; i<size ; i++,c++)
        sendCharacter(*c);
}
```

```
int getTurn(char outChar, char inChar){
    if(outChar=='s'&& inChar=='R')
        return 1;
    else
        return 0;
}
```

```
void setChoose(int *turn){ # pass &turn
    int outChar, inChar;
    puts("Send or Receive : ");
    outChar = getchar();
    sendCharacter(outChar);
    inChar = getCharacter();
    if(outChar == inChar)
        setChoose(&turn);
    else
        *turn = getTurn(inChar, outChar);
}
```

```
void setSwitch(char *c, int *
    int outChar = getche();
    switch(outChar){
        case backspace:
            deleteCharScreen();
            breale;
        case enter:
            sendString(c, strlen
            *turn++;
            breale;
        case ctrlQ:
            sendCharacter(outCh
            exit();
            breale;
        default:
            movePointer(outChar,
```

e pointer here