

BÀI 6: TRUYỀN THÔNG NỐI TIẾP QUA USART

Mục tiêu của bài học:

- (1) Trình bày tổng quan về truyền thông nối tiếp;
- (2) Trình bày được đặc điểm, các chế độ hoạt động của module EUSART trên PIC18F4520;
- (3) Thiết lập và điều khiển một số hàm thông dụng trong thư viện usart.h;
- (4) Lập trình và mô phỏng hoạt động của USART.

6.1 Tổng quan về truyền thông nối tiếp

Truyền tin nối tiếp là phương thức truyền tin trong đó các bit mang thông tin được truyền kế tiếp nhau trên một đường dẫn vật lý. Tại một thời điểm phía bên truyền cũng như bên nhận chỉ có thể truyền/nhận được 1 bit.

So với truyền tin song song thì truyền tin nối tiếp có một số ưu, nhược điểm sau:

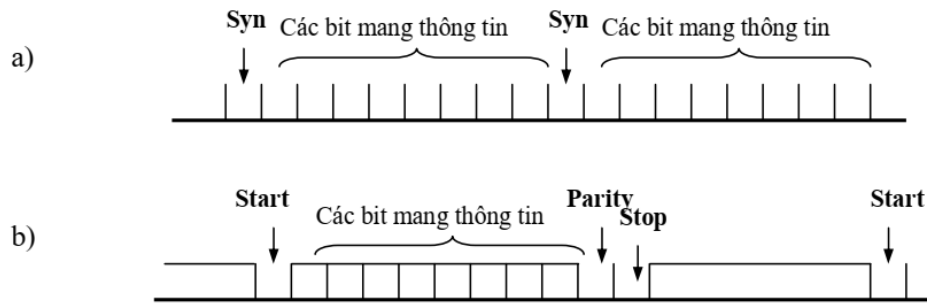
Ưu điểm: Truyền được ở khoảng cách xa, tiết kiệm đường truyền dẫn, truyền tin nối tiếp có ưu thế hơn so với truyền tin song song.

Nhược điểm: Tốc độ truyền chậm hơn, phức tạp hơn do các thiết bị thường phải có các khối chuyển đổi nối tiếp sang song song, song song sang nối tiếp khi sử dụng phương thức này để trao đổi tin.

Có 3 phương thức truyền tin nối tiếp:

- Phương thức đồng bộ: Các byte chứa các bit thông tin được truyền liên tiếp trên đường truyền và chỉ được ngăn cách nhau bằng bit đồng bộ khung (Syn). (Hình 6.1.a.)
- Phương thức không đồng bộ: Các byte chứa các bit thông tin được chứa trong một khung. Một khung được bắt đầu bằng 1 bit Start, tiếp theo là các bit mang thông tin, kế tiếp là bit kiểm tra chẵn lẻ và kết thúc là bit Stop (Hình 6.1.b).

- Phương thức lai: Đây là phương thức kết hợp của 2 phương thức trên



Hình 6.1 Các phương thức truyền nối tiếp

6.1 Modul EUSART trong PIC 18F4520

Các chân của Module EUSART là các chân đa chức năng trên PORTC. Để lựa chọn cấu hình các chân RC6/TX/CK và RC7/RX/DT là chân của Module EUSART cần phải thiết lập các bit sau:

- **Bit SPEN (RCSTA<7>)** phải được thiết lập (=1).
- **Bit TRISC<7>** phải được thiết lập (=1).
- **Bit TRISC<6>** phải được xóa bằng 0.

Hoạt động của Module EUSART được điều khiển thông qua ba thanh ghi:

- Thanh ghi trạng thái truyền và điều khiển (TXSTA).
- Thanh ghi trạng thái nhận và điều khiển (RCSTA).
- Thanh ghi điều khiển tốc độ baud (BAUDCON).

Chức năng từng bit của các thanh ghi xem thêm trong giáo trình

Tốc độ baud hay còn được gọi là tốc độ truyền thông, là số bit được truyền đi trên một giây (bps – bit per second).

Bộ tạo tốc độ baud BRG (Baud Rate Generator) có thể hoạt động ở chế độ 8 bit hoặc 16 bit, hỗ trợ cả chế độ đồng bộ và không đồng bộ của EUSART. Ở chế độ mặc định, BGR hoạt động ở chế độ 8 bit. Chế độ BGR 16 bit được lựa chọn khi bit BAUDCON<3> được thiết lập.

Hai thanh ghi SPBRGH:SPBRG được sử dụng để điều khiển chu kỳ xung tốc độ baud. - Lựa chọn chế độ và công thức tính tốc độ baud:

Cấu hình các bit			Chế độ BRG/EUSART	Công thức tốc độ baud
SYNC	BRG16	BRGH		
0	0	0	8-Bit/Không đồng bộ	$F_{OSC}/[64 (n + 1)]$
0	0	1	8-Bit/Không đồng bộ	$F_{OSC}/[16 (n + 1)]$
0	1	0	16-Bit/Không đồng bộ	
0	1	1	16-Bit/Không đồng bộ	$F_{OSC}/[4 (n + 1)]$
1	0	x	8-Bit/Đồng bộ	
1	1	x	16-Bit/Đồng bộ	

Ghi chú:

x = Giá trị bất kỳ n = Giá trị của cặp thanh ghi SPBRGH:SPBRG

Tính sai số tốc độ baud

$$sai\ so = \frac{Tốc\ do\ tính\ toán\ thực\ tế - Tốc\ do\ mong\ muốn}{Tốc\ do\ mong\ muốn} \%$$

Ví dụ: Với tần số thạch anh $F_{OSC} = 16\text{MHz}$, tốc độ truyền thông mong muốn là 9600, chế độ không đồng bộ, 8 bit BRG:

Tốc độ mong muốn = $F_{OSC}/(64(n+1))$

Giá trị cần nạp vào cặp thanh ghi SPBRGH:SPBRG:

$$x = ((F_{OSC}/\text{tốc độ mong muốn})/64) - 1$$

$$= ((16000000/9600)/64) - 1$$

$$= 25.042$$

Giá trị nạp vào cặp thanh ghi SPBRGH:SPBRG phải là số nguyên nên làm tròn là 25.

$$\text{Tốc độ tính toán thực tế} = 16000000/(64*(25+1)) = 9615$$

$$\text{Sai so} = (9615-9600)/9600 = 0.16\%$$

Tra cứu một số giá trị tính toán trang 207-208 trong tài liệu DataSheet 18F4520

Các bước lập trình truyền nhận chế độ không đồng bộ:

Các bước lập trình truyền dữ liệu chế độ không đồng bộ.

- Thiết lập tốc độ baud
- Cho phép truyền nối tiếp ở chế độ không đồng bộ
- Thiết lập các bit GIE, PEIE, TXIE khi có nhu cầu sử dụng ngắt ở cổng nối tiếp
- Thiết lập chế độ khung truyền là 8 bit hay 9 bit. Nếu khung truyền 9 bit nạp giá trị bit số 9 vào bit TX9D
- Cho phép truyền dữ liệu bit TXEN
- Nạp dữ liệu cần truyền vào thanh ghi TXREG (Quá trình truyền sẽ bắt đầu)

Các bước lập trình nhận dữ liệu chế độ không đồng bộ.

- Thiết lập tốc độ baud
- Cho phép nhận nối tiếp ở chế độ không đồng bộ
- Thiết lập các bit GIE, PEIE, RCIE, CREN khi có nhu cầu sử dụng ngắt nhận ở cổng nối tiếp
- Cờ ngắt RCIF sẽ được thiết lập khi hoạt động ngắt hoàn thành
- Thiết lập chế độ khung truyền là 8 bit hay 9 bit. Nếu khung truyền 9 bit đọc giá trị bit RX9D.
- Đọc 8 bit dữ liệu nhận về trong thanh ghi RCREG.
- Nếu phát hiện dữ liệu nhận bị lỗi, xóa lỗi bằng cách xóa bit cho phép nhận CREN

6.2 Một số hàm thông dụng trong thư viện USART

Để thuận tiện trong quá trình làm việc với cổng nối tiếp USART, MicroChip đã xây dựng một số thư viện trong đó có thư viện USART.h

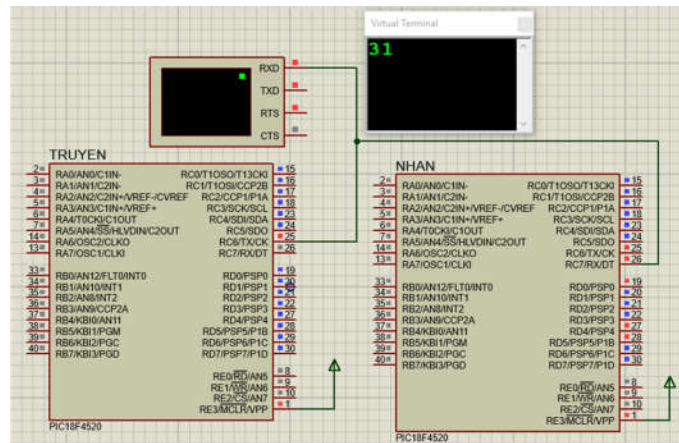
Hàm	Mô tả
BusyUSART	Hàm báo bận của hoạt động truyền nối tiếp.
CloseUSART	Cấm USART.
OpenUSART	Cấu hình PORT nối tiếp.
putsUSART	Truyền một mảng ra PORT USART.
ReadUSART	Đọc 1 byte từ PORT USART.
WriteUSART	Truyền 1 byte ra PORT USART.

Tham khảo thêm cách dùng các hàm trang 70 trong tài liệu “MPLAB-C18-Libraries_51297f.pdf”

6.3 Lập trình mô tả hoạt động của USART

Ví dụ: Viết chương trình giao tiếp giữa 2 vi điều khiển: biết $F_{osc} = 8\text{Mhz}$, tốc độ baud=9600

- VĐK thứ 1 truyền ký tự “1” sang VĐK 2
- VĐK 2 nhận và hiển thị lên PORTD kết quả nhận được dưới dạng nhị phân.



Hình 6.2 Mô phỏng hoạt động của chương trình.

Cách 1: Viết chương trình dưới dạng hàm:

- Vi điều khiển truyền

```
#include <P18f4520.h>
#include <delays.h>
#include <usart.h>
//C?u h?nh cho vi ?i?u khi?n, vi d?:
#pragma config OSC = HS
#pragma config MCLRE = ON
#pragma config WDT = OFF
void main()
{
    char x=0x31;    // Ma ASCII của ký tự cần truyền
    TRISC=0x80;    // Thiết lập RC6 là đầu ra, RC7 là đầu vào
    ADCON1=0x0F; // AN0 - AN12 là đầu vào/ra so
    OpenUSART(USART_TX_INT_OFF & //không sử dụng ngắt truyền
              USART_RX_INT_OFF & //không sử dụng ngắt nhận
              USART_ASYNCH_MODE & //chế độ không đồng bộ
              USART_EIGHT_BIT & //truyền/nhận 8bit
              USART_CONT_RX & //cho phép nhận liên tiếp
              USART_BRGH_HIGH, //tốc độ baud cao
              51); // 9600=8000000(16*(51+1))
    WriteUSART(x); // Truyền số 0x31
    while(1);
}
```

- vi điều khiển nhận

```
#include <P18f4520.h>
```

```

#include <delays.h>
#include<uart.h>
//C?u h?nh cho vi ?i?u khi?n, ví d?:
#pragma config OSC = HS
#pragma config MCLRE = ON
#pragma config WDT = OFF
void main()
{char x;
TRISC=0x80;
TRISD=0;
PORTD=0;
ADCON1=0x0F;
OpenUSART(USART_TX_INT_OFF & //khong su dung ngat truyen
    USART_RX_INT_OFF & //khong su dung ngat nhan
    USART_ASYNC_MODE & //che do khong dong bo
    USART_EIGHT_BIT & //truyen/nhan 8bit
    USART_CONT_RX & //cho phep nhan lien tiep
    USART_BRGH_HIGH, //toc do baud:=9600
    51);
while(1)
{
x=ReadUSART();
PORTD=x;
}
}

```

Cách 2: Viết dưới dạng thanh ghi

- *Vi điều khiển truyền*

```

#include <P18f4520.h>
#include <delays.h>
//C?u h?nh cho vi ?i?u khi?n, ví d?:
#pragma config OSC = HS
#pragma config MCLRE = ON
#pragma config WDT = OFF

void main()
{
char x=0x31 ; //truyen so 1
TRISD=0;
TRISC=0b10000000; //RC7 chieu vao RC6 chieu ra
ADCON1=0x0F; // AN0- AN12 la dau vao/ra so;
PORTD=0;
// 1. Thiet lap toc do Baud = 9615
SPBRG=51;
SPBRGH=0;
TXSTAbits.BRGH=1; // USART toc do cao
BAUDCONbits.BRG16=0; // thanh ghi toc do baud: 8bit
// 2. Thiet lap truyen noi tiep che do khong dong bo
TXSTAbits.SYNC=0; // hoat dong che do khong dong bo
RCSTAbits.SPEN=1; // Serial port Enable
// 3. Khong su dung ngat
// 4. Lua chon che do truyen 8 bit
TXSTAbits.TX9=0; // truyen 8 bit
// 5. Cho phep truyen nhan
RCSTAbits.CREN=1; // nhan lien tiep

```

```
TXSTAbits.TXEN=1; // cho phép truyền
```

```
// 6. Tiến hành truyền dữ liệu
```

```
RCSTAbits.SPEN=1;
```

```
TXREG=x; // Ký tự cần truyền
```

```
while(1);
```

```
}
```

- **Vi điều khiển nhận**

```
#include <P18f4520.h>
```

```
#include <delays.h>
```

```
#pragma config OSC = HS
```

```
#pragma config MCLRE = ON
```

```
#pragma config WDT = OFF
```

```
void main()
```

```
{
```

```
char x ;
```

```
TRISD=0;
```

```
TRISC=0b10000000; //RC7 chiều vào RC6 chiều ra
```

```
ADCON1=0x0F; // AN0- AN12 là đầu vào/ra số;
```

```
PORTD=0;
```

```
// 1. Thiết lập tốc độ Baud = 9615
```

```
SPBRG=51;
```

```
SPBRGH=0;
```

```
TXSTAbits.BRGH=1; // USART tốc độ cao
```

```
BAUDCONbits.BRG16=0; // thanh ghi tốc độ baud: 8bit
```

```
// 2. Thiết lập truyền nối tiếp chế độ không đồng bộ
```

```
TXSTAbits.SYNC=0; // hoạt động chế độ không đồng bộ
```

```
RCSTAbits.SPEN=1; // Serial port Enable
```

```
// 3. Không sử dụng ngắt
```

```
// 4. Lựa chọn chế độ nhận 8 bit
```

```
RCSTAbits.RX9=0; // nhận 8 bit
```

```
// 5. Cho phép truyền nhận
```

```
RCSTAbits.CREN=1; // nhận liên tiếp
```

```
TXSTAbits.TXEN=1; // cho phép truyền
```

```
// 6. Tiến hành nhận dữ liệu
```

```
while(1)
```

```
{
```



```
x=0;  
x= RCREG; // Ky tu can nhan  
PORTD=x; // 00110001  
}  
}
```