

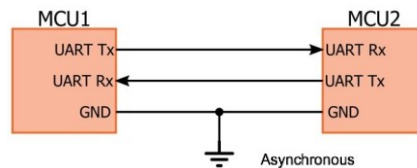
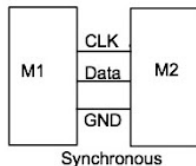


USART

(Universal Synchronous Asynchronous Receiver Transmitter)

Truyền thông nối tiếp (Serial Communication):

- ❑ Truyền thông nối tiếp đồng bộ: có 1 đường dữ liệu và 1 đường xung clock, thiết bị nào cấp xung clock sẽ là chủ, thiết bị nhận xung clock là tớ.
- ❑ Truyền thông nối tiếp bất đồng bộ: 1 đường phát dữ liệu và một đường nhận dữ liệu, không có tín hiệu đồng bộ xung clock. Mỗi bên phát và nhận phải tự tạo xung clock riêng có cùng tần số với nhau (boud = bit/s).



120/85



USART

(Universal Synchronous Asynchronous Receiver Transmitter)

Các thanh ghi liên quan UART:

1. Thanh ghi điều khiển và trạng thái truyền: TXSTA (Transmit Status And Control Register)
2. Thanh ghi điều khiển và trạng thái nhận: RCSTA (Receive Status And Control Register)
3. Thanh ghi điều khiển tốc độ Boud: SPBRG (USART Baud Rate Generator)
4. TXREG (USART Transmit Register)
5. RCREG (USART Receiver Register)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
99h	SPBRG	Baud Rate Generator Register							

121/85

TXSTA: TRANSMIT STATUS AND CONTROL REGISTER (ADDRESS 98h)

	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D

bit 7 bit 0

bit 7 **CSRC:** Clock Source Select bit
Asynchronous mode:
 Don't care.
Synchronous mode:
 1 = Master mode (clock generated internally from BRG)
 0 = Slave mode (clock from external source)

bit 6 **TX9:** 9-bit Transmit Enable bit
 1 = Selects 9-bit transmission
 0 = Selects 8-bit transmission

bit 5 **TXEN:** Transmit Enable bit
 1 = Transmit enabled
 0 = Transmit disabled
Note: SREN/CREN overrides TXEN in Sync mode.

bit 4 **SYNC:** USART Mode Select bit
 1 = Synchronous mode
 0 = Asynchronous mode

bit 3 **Unimplemented:** Read as '0'

bit 2 **BRGH:** High Baud Rate Select bit
Asynchronous mode:
 1 = High speed
 0 = Low speed
Synchronous mode:
 Unused in this mode.

bit 1 **TRMT:** Transmit Shift Register Status bit
 1 = TSR empty
 0 = TSR full

bit 0 **TX9D:** 9th bit of Transmit Data, can be Parity bit

RCSTA: RECEIVE STATUS AND CONTROL REGISTER (ADDRESS 18h)

	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D

bit 7 bit 0

bit 7 **SPEN:** Serial Port Enable bit
 1 = Serial port enabled (configures RC7/RX/DT and RC6/TX/CK pins as serial port pins)
 0 = Serial port disabled

bit 6 **RX9:** 9-bit Receive Enable bit
 1 = Selects 9-bit reception
 0 = Selects 8-bit reception

bit 5 **SREN:** Single Receive Enable bit
Asynchronous mode:
 Don't care.
Synchronous mode – Master:
 1 = Enables single receive
 0 = Disables single receive
 This bit is cleared after reception is complete.
Synchronous mode – Slave:
 Don't care.

bit 4 **CREN:** Continuous Receive Enable bit
Asynchronous mode:
 1 = Enables continuous receive
 0 = Disables continuous receive
Synchronous mode:
 1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
 0 = Disables continuous receive

bit 3 **ADDEN:** Address Detect Enable bit
Asynchronous mode 9-bit (RX9 = 1):
 1 = Enables address detection, enables interrupt and load of the receive buffer when RSR<8> is set
 0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit

bit 2 **FERR:** Framing Error bit
 1 = Framing error (can be updated by reading RCREG register and receive next valid byte)
 0 = No framing error

bit 1 **OERR:** Overrun Error bit
 1 = Overrun error (can be cleared by clearing bit CREN)
 0 = No overrun error

bit 0 **RX9D:** 9th bit of Received Data (can be parity bit but must be calculated by user firmware)

TÍNH TỐC ĐỘ BOUD TƯƠNG ỨNG VỚI FOSC

TABLE 10-1: BAUD RATE FORMULA

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = Fosc/(64 (X + 1))	Baud Rate = Fosc/(16 (X + 1))
1	(Synchronous) Baud Rate = Fosc/(4 (X + 1))	N/A

Legend: X = value in SPBRG (0 to 255)

TABLE 10-4: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 1)

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	-	-	-	-	-	-	-	-	-
2.4	-	-	-	-	-	-	2.441	1.71	255
9.6	9.615	0.16	129	9.615	0.16	103	9.615	0.16	64
19.2	19.231	0.16	64	19.231	0.16	51	19.531	1.72	31
28.8	29.070	0.94	42	29.412	2.13	33	28.409	1.36	21
33.6	33.784	0.55	36	33.333	0.79	29	32.895	2.10	18
57.6	59.524	3.34	20	58.824	2.13	16	56.818	1.36	10
HIGH	4.883	-	255	3.906	-	255	2.441	-	255
LOW	1250.000	-	0	1000.000	-	0	625.000	-	0

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-
1.2	1.202	0.17	207	1.2	0	191
2.4	2.404	0.17	103	2.4	0	95
9.6	9.615	0.16	25	9.6	0	23
19.2	19.231	0.16	12	19.2	0	11
28.8	27.798	3.55	8	28.8	0	7
33.6	35.714	6.29	6	32.9	2.04	6
57.6	62.500	8.51	3	57.6	0	3
HIGH	0.977	-	255	0.9	-	255
LOW	250.000	-	0	230.4	-	0

bit 3

bit 2

Unimplemented: Read as '0'

BRGH: High Baud Rate Select bit

Asynchronous mode:

1 = High speed

0 = Low speed

Synchronous mode:

Unused in this mode.

TRMT: Transmit Shift Register Status bit

1 = TSR empty

0 = TSR full

bit 1

bit 0

TX9D: 9th bit of Transmit Data, can be Parity bit

TABLE 10-1: BAUD RATE FORMULA

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = Fosc/(64 (X + 1))	Baud Rate = Fosc/(16 (X + 1))
1	(Synchronous) Baud Rate = Fosc/(4 (X + 1))	N/A

Legend: X = value in SPBRG (0 to 255)

TABLE 10-3: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 0)

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	1.221	1.75	255	1.202	0.17	207	1.202	0.17	129
2.4	2.404	0.17	129	2.404	0.17	103	2.404	0.17	64
9.6	9.766	1.73	31	9.615	0.16	25	9.766	1.73	15
19.2	19.531	1.72	15	19.231	0.16	12	19.531	1.72	7
28.8	31.250	8.51	9	27.778	3.55	8	31.250	8.51	4
33.6	34.722	3.34	8	35.714	6.29	6	31.250	6.99	4
57.6	62.500	8.51	4	62.500	8.51	3	52.083	9.58	2
HIGH	1.221	-	255	0.977	-	255	0.610	-	255
LOW	312.500	-	0	250.000	-	0	156.250	-	0

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	0.300	0	207	0.3	0	191
1.2	1.202	0.17	51	1.2	0	47
2.4	2.404	0.17	25	2.4	0	23
9.6	8.929	6.99	6	9.6	0	5
19.2	20.833	8.51	2	19.2	0	2
28.8	31.250	8.51	1	28.8	0	1
33.6	-	-	-	-	-	-
57.6	62.500	8.51	0	57.6	0	0
HIGH	0.244	-	255	0.225	-	255
LOW	62.500	-	0	57.6	-	0

bit 3

bit 2

Unimplemented: Read as '0'

BRGH: High Baud Rate Select bit

Asynchronous mode:

1 = High speed

0 = Low speed

Synchronous mode:

Unused in this mode.

TRMT: Transmit Shift Register Status bit

1 = TSR empty

0 = TSR full

bit 1

bit 0

TX9D: 9th bit of Transmit Data, can be Parity bit

125/85



USART

(Universal Synchronous Asynchronous Receiver Transmitter)

Nguyên tắc hoạt động truyền data:

1. Dữ liệu cần truyền được đặt vào thanh ghi TXREG, baud rate được tạo ra, khi TXEN gán bằng 1 → dữ liệu từ thanh ghi TXREG đưa vào thanh ghi TSR đồng thời baud rate tác động đến TSR, đẩy dữ liệu cần truyền ra bộ đếm sau đó xuất ra chân TX.
2. Bit TXIF dùng để báo trạng thái trong thanh ghi TXREG, nếu có dữ liệu trong TXREG thì TXIF = 1. Nếu dữ liệu được truyền xuống thanh TSR thì TXIF = 0. Tương tự bit TRMT dùng để báo trạng thái thanh ghi TSR.

126/85



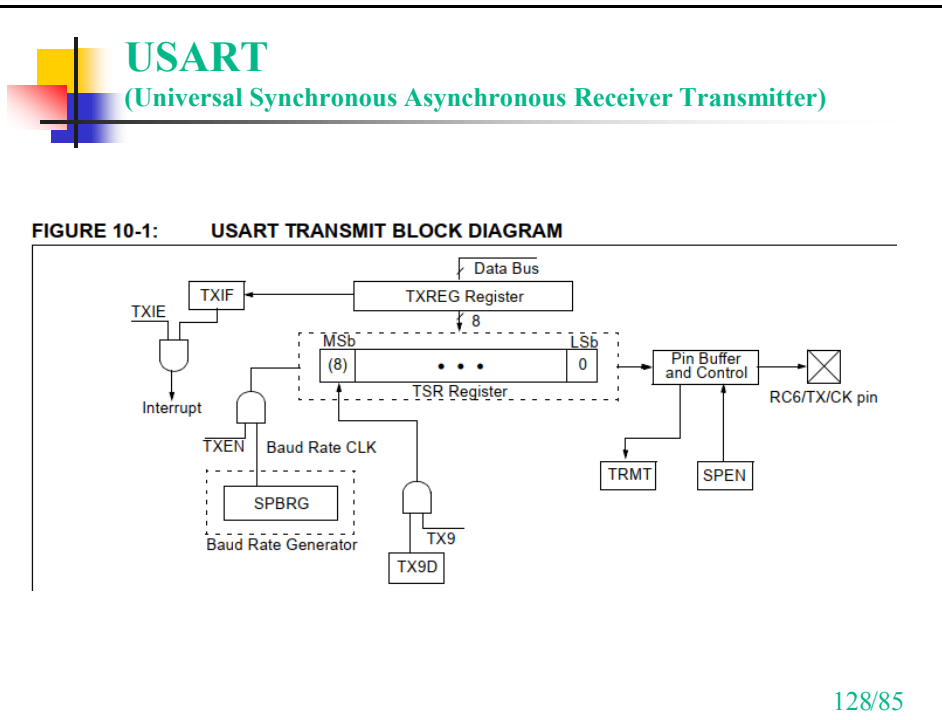
USART

(Universal Synchronous Asynchronous Receiver Transmitter)

Các bước cho quá trình gửi dữ liệu bao gồm:

1. Khởi tạo baud rate: ở thanh ghi SPBRG
Cho phép quá trình truyền thông không đồng bộ bằng cách thiết lập 2 bit thanh ghi TXSTA: SPEN = 1; SYNC = 0;
2. Cho phép truyền dữ liệu bằng cách thiết lập bit TXEN = 1;
3. Khi cần truyền dữ liệu thì cần gán dữ liệu đó vào TXREG.
Thanh ghi quy định chế độ truyền.

127/85



USART
(Universal Synchronous Asynchronous Receiver Transmitter)

Nguyên tắc hoạt động nhận data:

Khi có dữ liệu được truyền đến chân **RX** và bit **SPEN** được cho phép thì dữ liệu sẽ được đồng bộ với khối tạo xung, vì baud rate giữa 2 khối bằng nhau nên xung baud rate mang từng bit vào thanh ghi RSR, khi một frame được truyền hoàn tất (xuất hiện bit stop) thì dữ liệu được truyền xuống thanh ghi RCREG, bit thứ 9 được truyền xuống RX9D, nếu có lỗi thì các bit OERR, FERR dùng để thông báo.

Quá trình nhận cũng tạo ra ngắt $RCIF = 1$, bằng cách thiết lập bit $RCIE = 1$ mỗi khi có dữ liệu truyền đến thì sẽ sinh ra ngắt và PIC sẽ tạm dừng chương trình hiện thời để xử lý dữ liệu vừa nhận được.

0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
-----	------	----------------------	------	-------------	------	-------	--------	--------	--------

129/85



USART

(Universal Synchronous Asynchronous Receiver Transmitter)

Các bước cho quá trình nhận dữ liệu bao gồm:

1. Khởi tạo baud rate: ở thanh ghi SPBRG Cho phép quá trình truyền thông không đồng bộ bằng cách thiết lập SPEN = 1; SYNC = 0;
2. Cho phép ngắt quá trình nhận dữ liệu CREN = 1;
3. Cho phép ngắt toàn cục: CIE = 1; PEIE = 1;
4. Xử lý các phần khác của chương trình khi có ngắt xảy ra thì xử lý dữ liệu.

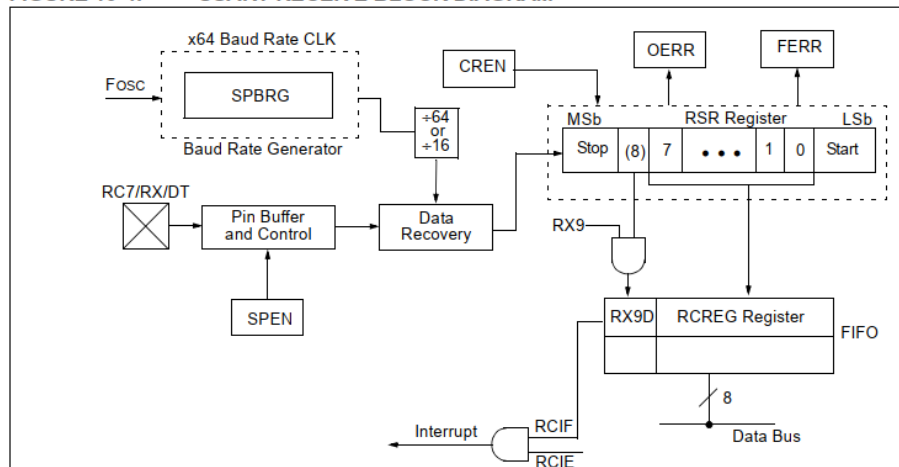
130/85




USART

(Universal Synchronous Asynchronous Receiver Transmitter)

FIGURE 10-4: USART RECEIVE BLOCK DIAGRAM



131/85




USART

(Universal Synchronous Asynchronous Receiver Transmitter)

Code truyền:

<pre style="color: red;">#include <16F877a.h> #FUSES NOWDT, HS, PUT, NOPROTECT, #use DELAY(clock=16000000) #use rs232(baud=9600,xmit=pin_c6,rcv=pin_c7) #define ON PIN_B0 #define OFF PIN_B1 int8 TDATA; void main() { SET_TRIS_B(0xFF); PORT_B_PULLUPS(0xFF); SET_TRIS_D(0x00); OUTPUT_D(0x00); //PUTC(DATA) gửi dữ liệu ra port nối tiếp </pre>	<pre> WHILE(true) { IF (!INPUT(ON)) { TDATA=0xFF; PUTC(TDATA); OUTPUT_D(TDATA); } IF (!INPUT(OFF)) { TDATA=0x00; PUTC(TDATA); OUTPUT_D(TDATA); } } </pre>
---	---

132/85



USART

(Universal Synchronous Asynchronous Receiver Transmitter)

Code nhận:

```
#include <16F877a.h>
#FUSES NOWDT, HS, PUT, NOPROTECT,
#use DELAY(clock=16000000)
#use rs232(baud=9600,xmit=pin_c6,rcv=pin_c7)
int8 RDATA;
void main()
{
  SET_TRIS_D(0x00);
  output_d(0);
  while(1)
  {
    if(kbhit()) // Hàm kbhit() trả về 1 nếu có ký tự đã nhận và 0 nếu không có ký tự
    {
      RDATA=getch(); //nhận ký tự gửi đến
      OUTPUT_D(RDATA);
    }
  }
}

```

133/85

