

### 11.3. Asigurarea calitatii sistemelor EMBEDDED

### 11.3.1. Unitatea pentru comunicații seriale sincrone și asincrone (EUSART)

Modulul EUSART este periferic destinat comunicațiilor seriale. Acesta conține ceasul, regiștrii de deplasare și memoria tampon necesară transferului serial de date. Acest periferic poate fi configurat să lucreze în modul full-duplex asincron sau ca un sistem semi-duplex sincron. Modul asincron este util pentru realizarea comunicației cu calculatoarele personale, în timp ce modul sincron este folosit pentru comunicarea cu convertoarele analog digitale externe, memorii seriale sau alte microcontrolere. Aceste dispozitive nu au ceas intern pentru generarea ratei de transmisie și se folosesc de ceas extern furnizat de unitatea master sincronă.

### Performantele perifericului EUSART:

- transmisie și recepție asincronă în modul full-duplex.
- memorie tampon de intrare pentru două caractere.
- memorie tampon de ieșire pentru două caractere.
- Cuvinte de lungime programabilă 8 sau 9 biți
- detecția adresei în modul de 9 biți.
- detecție de erori
- semiduplex master sincron,
- semiduplex slave sincron.
- detecția și calibrarea ceasului intern
- trezirea din modul de consum redus la recepția caracterului de sincronizare.
- transmiterea caracterului de sincronizare pe 13biți

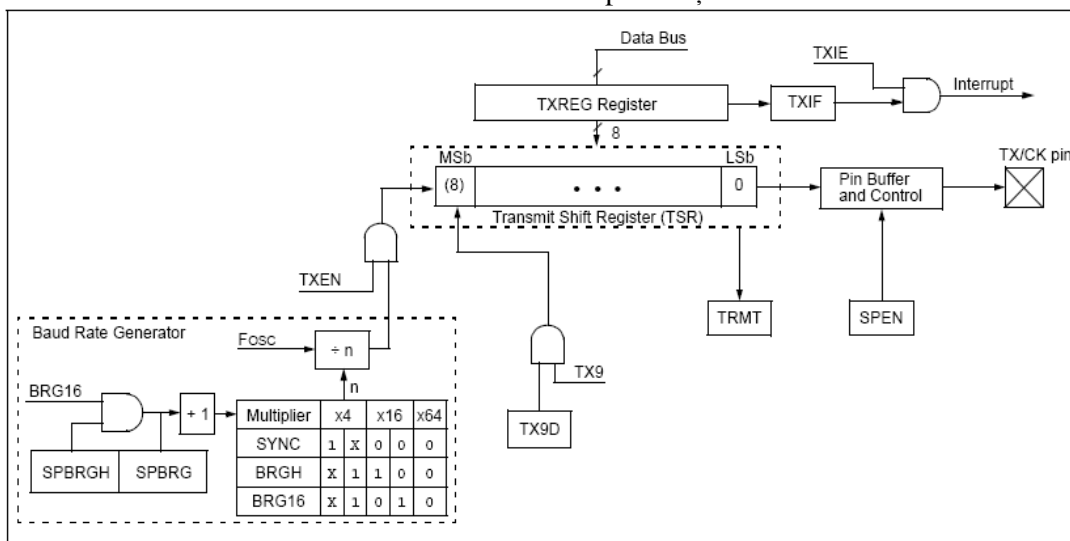


Figura 11. 1. Schema bloc a modului de transmisie serială

### 11.3.2 Modul de comunicații Asincron.

În acest mod modulul de comunicații seriale transmite și recepționează date folosind formatul standard NRZ (non-return-to-zero). Formatul NRZ folosește două stări: Voh ce corespunde valorii logice „1” și Vol ce corespunde valorii logice „0”. Fiecare caracter este format dintr-un bit de start, care este 0 logic, urmat de 8 sau 9 biți de date, și se termină cu 1 sau 1.5 sau 2 biți de stop. În cazul comunicației seriale primul bit trimis este LSB.

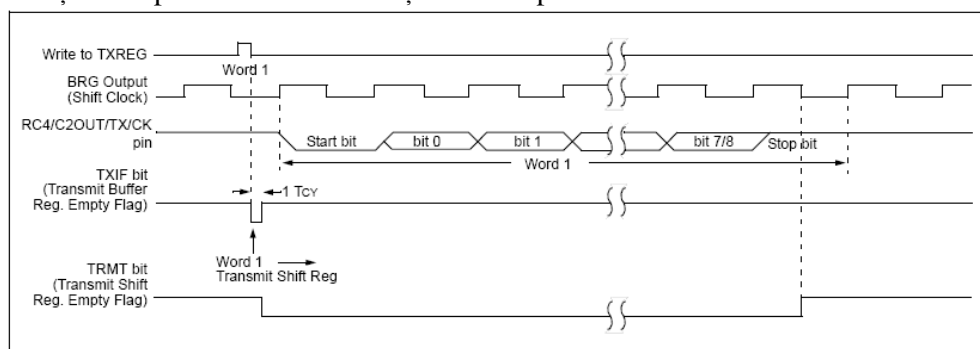


Figura 11. 2. Transmisia unui caracter

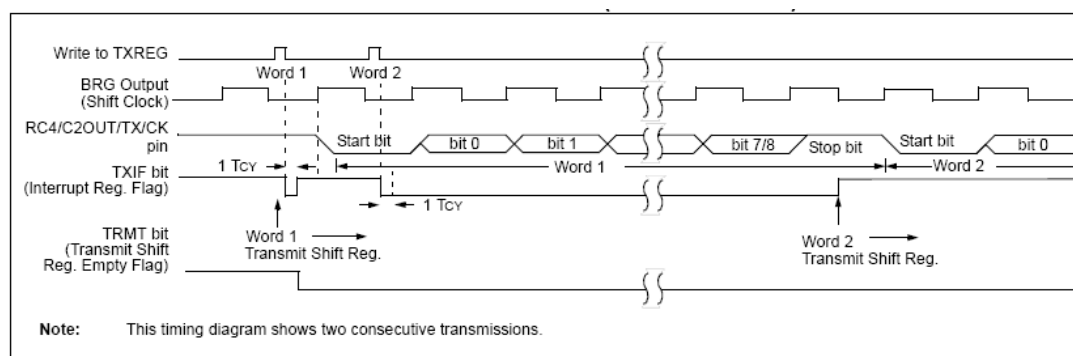


Figura 11. 3. Transmisia a două caractere

### 11.3.4 Receptorul serial asincron

Modul asincron este folosit cu precădere în sistemele ce folosesc magistrala RS-232. diagrama bloc a receptorului este data în Figura 11. 4. Data este recepționată pe pinul RX și controlează blocul de recuperare a datelor.

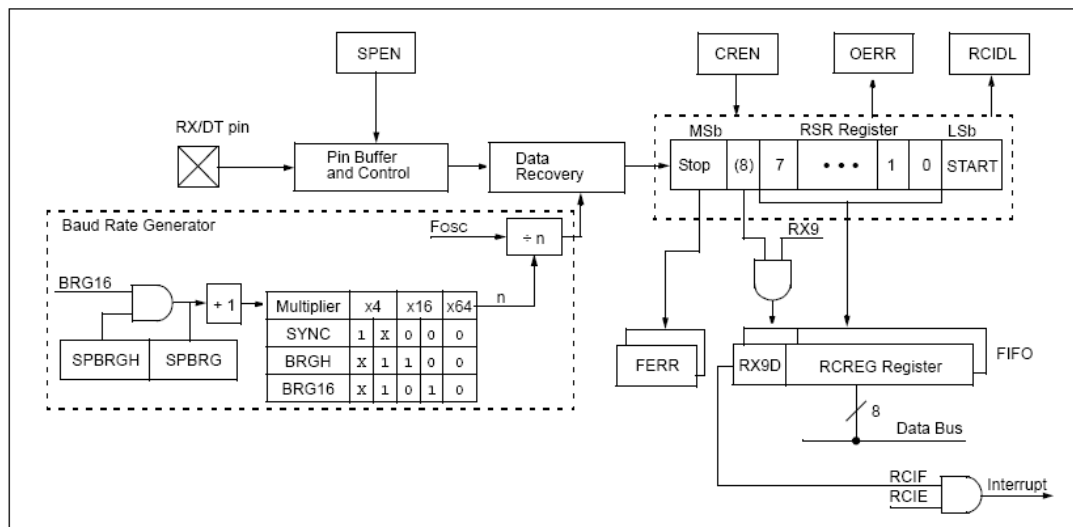


Figura 11.4. Schema bloc a modului de recepție serială

Blocul de recuperare al datelor este de fapt un registru de deplasare rapid ce operează cu o viteză de 16 ori mai mare ca viteza de comunicație, în schimb de registrul de deplasare de recepție lucrează la o viteză egală cu viteza de comunicație. După ce toți biții au fost recepționați, caracterul este transferat într-o memorie tampon cu o dimensiune de două caractere. Astfel că receptorul poate primi două caractere și să înceapă să îl proceseze pe al treilea până când microcontrolerul începe să proceseze datele. Accesul la datele recepționate se face doar prin registrul RCREG.

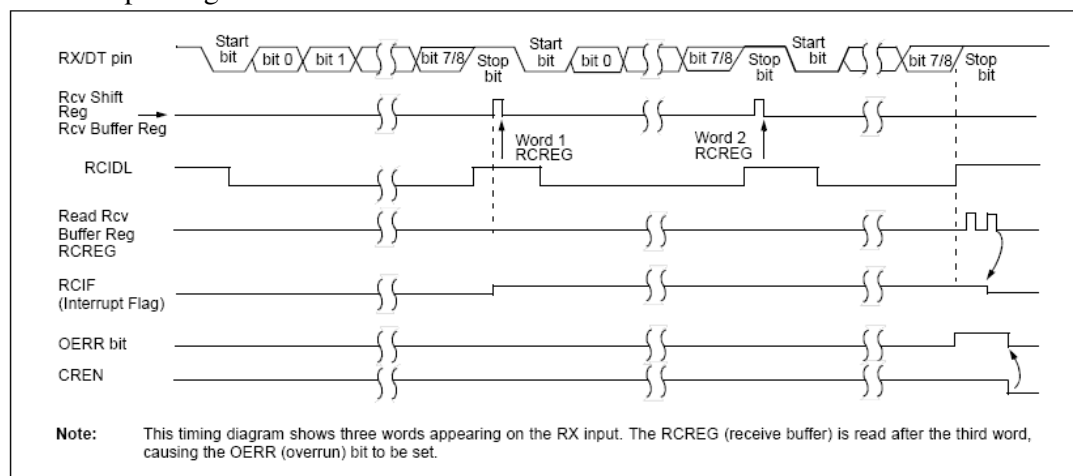


Figura 11.5. Recepția asincronă

### 11.3.7 Descrierea regiștrilor de control

Registrul de stare și control al transmisiei seriale – TXSTA

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN <sup>(1)</sup>	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7							bit 0

## Structuri hardware si algoritmi specifici microsystemelor EMBEDDED

bit 7 **CSRC**: Bitul de selectie a sursei ceasului

în modul asincron:

-nu contează

în modul sincron:

1 = modul master (ceasul este generat interg de catre registrul BRG)

0 = Slave mode (ceasul provine dintr-o sursă externă)

bit 6 **TX9**: bitul de selecție a lungimii cuvântului 8 sau 9 biți

1 = selectează cuvinte pe 9 biți

0 = selectează cuvinte pe 8 biți

bit 5 **TXEN**: bitul de pornire al transmițătorului serial

1 = transmițător pornit

0 = transmițător oprit

bit 4 **SYNC**: modul de lucru al perifericului

1 = modul sincron

0 = modul asincron

bit 3 **SENDB**: transmisia bitului de sincronizare

în modul asincron:

1 = se va transmite bitul de sincronizare la următoarea transmisie (bitul va fi șters printr-un mecanism hardware)

0 = transmisia caracterului de sincronizare s-a efectuat

în modul sincron:

-nu contează

bit 2 **BRGH**: selectarea vitezelor mari de transmisie

în modul asincron:

1 = viteză mare

0 = viteză mică

în modul sincron:

- nu contează

bit 1 **TRMT**: bitul de stare al registrului de deplasare de transmisie

1 = registrul de deplasare este gol

0 = registrul de deplasare este plin

bit 0 **TX9D**: funcția celui de al 9-lea bit

poate fi bit de date sau de paritate.

### 11.3.8 Exemple de program

[illegible]

UNILINE A EUROPEANĂ



MINISTERUL MUNCII, FAMILIEI ȘI  
PROTECȚIEI SOCIALE  
AMPOSOKU



FONDUL SOCIAL EUROPEAN  
POSDRU  
2007-2013

INSTRUMENTE STRUTTURALE  
2007-2013



**Structuri hardware si algoritmi specifici microsystemelor EMBEDDED**

```
        porta.2=1;
        porta.1=0;
    }
    void no4(void)
    {
        trisa.4=1;
        trisa.5=0;
        trisa.2=0;
        trisa.1=1;

        porta.4=0;
        porta.5=1;
        porta.2=0;
        porta.1=0;
    }
    void no5(void)
    {
        trisa.4=1;
        trisa.5=0;
        trisa.2=0;
        trisa.1=1;

        porta.4=0;
        porta.5=0;
        porta.2=1;
        porta.1=0;
    }
    void no6(void)
    {
        trisa.4=1;
        trisa.5=1;
        trisa.2=0;
        trisa.1=0;

        porta.4=0;
        porta.5=0;
        porta.2=1;
        porta.1=0;
    }
    void no7(void)
    {
        trisa.4=1;
        trisa.5=1;
        trisa.2=0;
        trisa.1=0;

        porta.4=0;
        porta.5=0;
        porta.2=0;
        porta.1=1;
    }

    void led_control(void)//rutina de generare a PWM-ului
    {
        if(i==0)
        {
            no0();
        }
    }
```



UNIUNEA EUROPEANĂ

MINISTERUL MUNCII, FAMILIEI ȘI  
PROTECȚIEI SOCIALE  
AMFOSDRUFONDUL SOCIAL EUROPEAN  
POSDRU  
2007-2013INSTRUMENTE STRUCTURALE  
2007-2013

## Structuri hardware si algoritmi specifici microsystemelor EMBEDDED

[illegible]

## Structuri hardware si algoritmi specifici microsystemelor EMBEDDED

[illegible]



## Structuri hardware si algoritmi specifici microsystemelor EMBEDDED

```

tont--;
    }
    if(rcreg == 'c')
    {
        //    no3();
    }
    if(rcreg == 'd')
    {
        //        no4();
    }

}

if(pir1 & (1<<TXIF))    // daca intreruperea este cauzata
                        //de transmiterea unui caracter pe
                        //seriala: se va transmite
                        //urmatorul
{
    clear_bit(pir1, TXIF);
    if(ctx==0)
    {
        txreg=tx_buf[1];
        ctx=1;
        //    no0();
    }
    else
    {
        if(ctx==1)
        {
            txreg=tx_buf[2];
            ctx=2;
            //    no1();
        }
        else
        {
            if(ctx==2)
            {
                txreg=tx_buf[3];
                ctx=3;
                //    no2();
            }
            else
            {
                if(ctx==3)
                {
                    txreg=tx_buf[4];
                    ctx=4;
                    //    no3();
                }
                else
                {
                    if(ctx==4)
                    {
                        while((TXSTA&0x02) == 0){}
                        txreg=13;//tx_buf[4];
                        ctx=5;
                    }
                }
            }
        }
    }
}

```



## Structuri hardware si algoritmi specifici microsystemelor EMBEDDED

```
//          no4();  
            }  
            else  
            {  
  
                if(ctx==5)  
                {  
                    while((TXSTA&0x02) == 0){}  
                    txreg=13;  
                    ctx=6;  
//                    no5();  
  
                }  
                else  
                {  
                    if(ctx==6)  
                    {  
                        while((TXSTA&0x02) == 0){}  
                        txreg=13;  
                        ctx=7;  
//                        no5();  
  
                    }  
                    else  
  
                    {  
                        if(ctx==7)  
                        {  
                            // rcsta = 0b00000000;  
                            txsta = 0b00000000;  
                            ctx=7;  
                            // no6();  
  
                        }  
                        else  
                        {  
                            // rcsta = 0b00000000;  
                            txsta = 0b00000000;  
                            ctx=8;  
                            // no7();  
                        }  
                    }  
                }  
            }  
        }  
    }  
}  
}  
  
void main(void)  
{  
    osccon=0b01110000;           // ceas de 8MHz  
    osctune=0b00000000;  
    ansel=0b00000001;           //RA1 este intrare analogica  
    //Configure port A  
    trisa = 0xFF;  
    //Configure port C
```



UNILINE A EUROPEANĂ



MINISTERUL MUNCII, FAMILIEI ȘI  
PROTECȚIEI SOCIALE  
AMPOSOKU



FONDUL SOCIAL EUROPEAN  
POSDRU  
2007-2013

INSTRUMENTE STRUTTURALE  
2007-2013

**Structuri hardware si algoritmi specifici microsistemelor EMBEDDED**

```

trisc = 0b11101111;          //RC4 este iesire digitala pentru
                               //datele seriale, RC5 este intrare
                               //pentru datele seriale

//Initialize port A
porta = 0x00;
//Initialize port C
portc = 0b00010000;          //cand nu se transmite nimic sunt
                               //5V

i=0;
tont=0;
option_reg = 0b00000110;     //prescaler is giving clock to
                               //timer 0 and the division ration
                               //is 128

//Enable interrupts (Timer0)
intcon = 0b11100000;         //enable global interrupts,
                               //nonmaskable interrupts and Timer 0
                               //interrupt

piel = 0b01100010;           //enable ADC, rx, and tx interrupt
adcon0 = 0b10000001;         //enable ADC
adcon1 = 0b00100000;         //ADC clock =1/32 system clock
adcon0.l=1;                  // (start adc sampling)

txsta = 0b00000000;          // TX este disable
rcsta = 0b10010000;          //UART si RX sunt enable
baudctl = 0b00000000;
spbrg=12;                    // viteza comunicatiei pe seriala
                               //este 9600 baudzi

ctx=0;
// no0();
//Endless loop
while( 1 )
{

}
}

```

**Bibliografie:**

1. Istvan Sztojanov, Sever Pașca, Elisabeta Buzoianu, Aplicații hardware și software cu microcontrolerul PIC12F675, Editura Cavallioti, ISBN 978-973-7622-54-9, Bucuresti 2008
2. Istvan Sztojanov, Alexandru Vasile, Elisabeta Buzoianu, Sever Pașca, *Programarea microcontrolerelor din familia Intel, Aplicații practice hardware cu 80C552*, Editura Man-Dely, ISBN 973-85681-5-3, București 2004.
3. <http://vega.unitbv.ro/~romanca/EmbSys/>
4. <http://facultate.regielive.ro/cursuri/electronica/>
5. [www.microcip.com](http://www.microcip.com)

