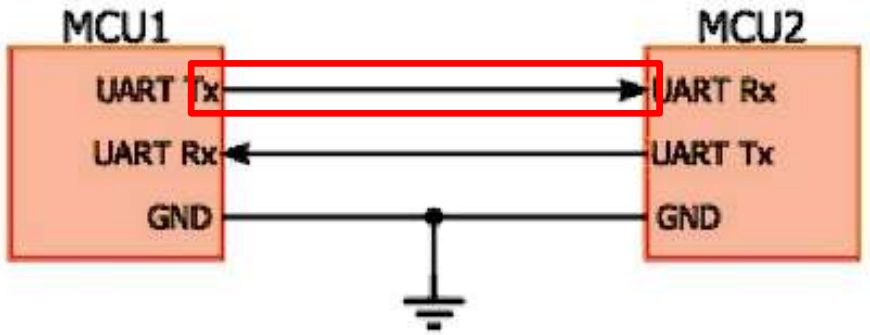
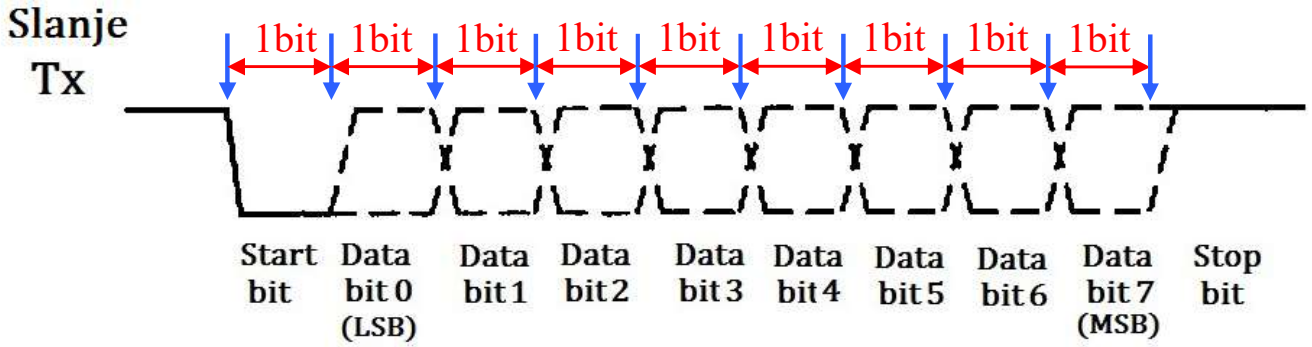
MIKROKONTROLER AT89C51RC2

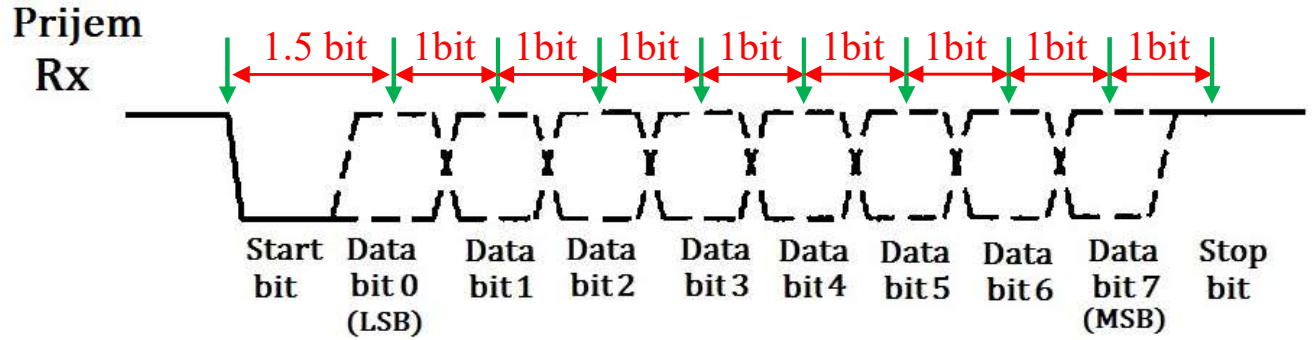
asinhrona serijska komunikacija

Asinhrona serijska komunikacija



Npr. baud rate 9600bps:
1bit=1/9600s





Serijska komunikacija – konfigurisanje 1/3

AT89C51RC2 posjeduje integrisan *UART (universal asynchronous receiver / transmitter)*, odnosno serijski port koji posjeduje svu funkcionalnost UART porta 8051 uz određena proširenja funkcionalnosti. UART omogućuje veoma lako slanje i primanje informacija preko asinhrone serijske komunikacije. Potrebno je definisati mod rada serijskog porta i brzinu (baud rate). Nakon toga, sve što je potrebno da bi se podaci razmjenili je da se vrijednost upiše u odgovarajući SFR, ako se šalje, odnosno da se očita iz istog SFR, kada se prima. Mikrokontroler automatski javlja kada je završio sa slanjem bajta koji je upisan i, isto tako, kada je primio bajt podataka koji može da se obradi. Korisnički program ne mora da vodi računa o prenosu podataka na najnižem, bitskom, nivou.

Definisanje moda rada serijskog porta

Serijski port se konfiguriše pomoću SFR registra SCON (98h) koji je bit adresibilan.

Bit	Ime	Bit adresa	Objašnjenje funkcije
7	SM0/FE	9Fh	FE – kada se desi framing error pri prijemu ima vrijednost 1 (pristup kada je SMOD0=1). Ispravan stop bit ne resetuje njegovu vrijednost, to se radi softverski
			SM0 - Mod rada serijskog porta – bit 0 (pristup kada je SMOD0=0)
6	SM1	9Eh	Mod rada serijskog porta – bit 1
5	SM2	9Dh	Dozvola multiprocesorske komunikacije
4	REN	9Ch	Dozvola primanja. Ovaj bit mora biti na logičkoj 1 da bi se primali podaci.
3	TB8	9Bh	9-i bit za slanje u modu 2 i 3.
2	RB8	9Ah	9-i bit primljen u modu 2 i 3.
1	TI	99h	Transmit Flag. Na logičkoj 1 kada je bajt uspješno poslat.
0	RI	98h	Receive Flag. Na logičkoj 1 kada je bajt uspješno primljen.

Serijska komunikacija – konfigurisanje 3/3

Najviša 2 bita u SCON regustru služe za konfigurisanje serijske komunikacije

SM0	SM1	Mod rada	Objašnjenje	Baud Rate
0	0	0	8-bit Shift Register	Fosc/12 (X1 mod) ili Fosc/6 (X2 mod)
0	1	1	8-bitni UART	Promjenjiv
1	0	2	9-bitni UART	Fosc/64 (X1 mod) ili Fosc/32 (X2 mod) (*)
1	1	3	9-bitni UART	Promjenjiv

(*) Napomena: Definisani baud rate se duplira ukoliko se najviši bit u registru PCON - PCON.7 (SMOD1) nalazi na logičkoj 1.

Izbor moda serijskog porta definiše mod rada (8-bit/9-bit, UART ili Shift Register) i takođe određuje kako se izračunava baud rate. U modovima 0 i 2 baud rate je fiksiran i određen je frekvencijom oscilatora. U modovima 1 i 3, baud rate je promjenjiv. Može biti definisan pomoću internog baud rate generatora ili baziran na tome kako često dolazi do overflowa Timera 1 ili Timera 2.

Bit **SM2** je flag za multiprocesorsku komunikaciju. Koristi se kod nekih naprednih serijskih aplikacija. Za uobičajeno korišćenje serijske komunikacije neophodno je da ovaj flag bude na logičkoj 0.

Bit **REN**, je "Receiver Enable“, odnosno dozvola primanja. Ako se žele primati podaci preko serijskog porta, ovaj bit mora biti na logičkoj 1.

Serijska komunikacija – konfigurisanje 3/3

Najviša 2 bita u specijalnom funkcijskom registru PCON (87h) koriste se pri konfigurisanju serijske komunikacije, odnosno služe za definisanje kontrole stop bita (framing error) i duplog baud rate-a

PCON (87h)

Bit	Ime	Objašnjenje funkcije
7	SMOD1	Kada je ovaj bit na logičkoj jedinici u modovima 1, 2 i 3 je izabran dupli baud rate
6	SMOD0	Kada je ovaj bit na 0, onda se preko 7 bita u SCON registru pristupa SM0 bitu - može da se definiše mod rada Kada je ovaj bit na 1, aktivirana je kontrola stop bita (framing error) i preko 7 bita u SCON registru pristupa se FE bitu

Serijska komunikacija – slanje i primanje podataka 1/2

Niža 4 bita u SCON registru ne služe za konfigurisanje komunikacije, nego se koriste pri slanju i primanju podataka.

TB8 bit se koristi u modovima 2 i 3. U ovim modovima prenosi se ukupno 9 bita. Prvih 8 bita predstavljaju podatak, a deveti bit se uzima iz TB8. Prvo će se preko serijske linije poslati 8 bita koji čine podatak, a nakon njih će da se poslati i deveti bit, ono što smo upisali u TB8.

RB8 bit se takođe koristi u modovima 2 i 3 i funkcioniše u osnovi na isti način kao i TB8, samo sa strane primanja podataka. Kada se primi bajt podataka u modovima 2 i 3, prima se ukupno 9 bita. U tom slučaju, prvih 8 bita koji se primaju su podaci koji predstavljaju primljeni bajt podataka, a vrijednost devetog bita se upisuje u RB8.

TB8 i RB8 se u modovima 2 i 3 koriste za detekciju greške pri prenosu pomoću parity bita.

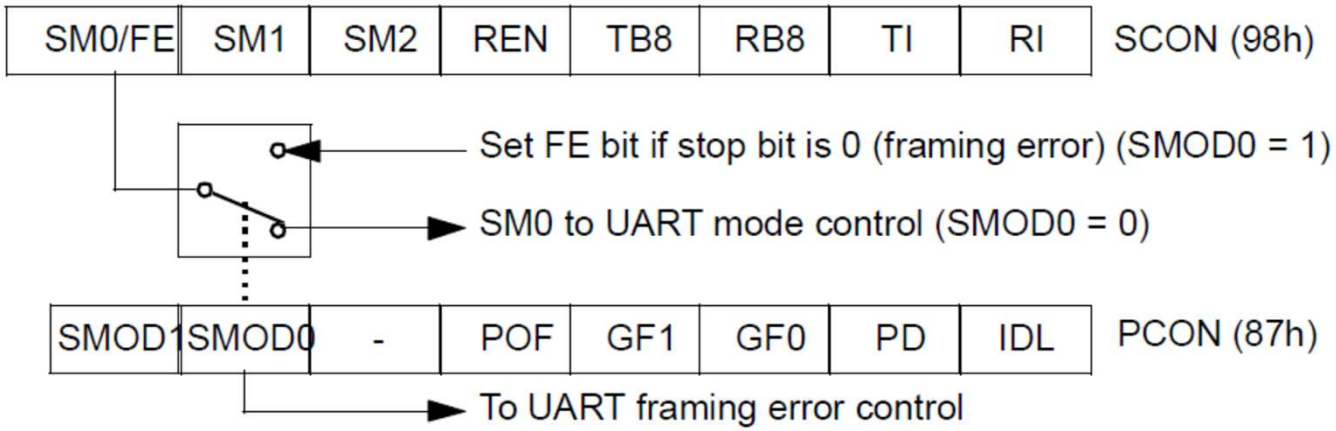
8 bita podataka	vrijednost parity bita	
	even parity	odd parity
00000000	**0**00000000	**1**00000000
10100010	**1**10100010	**0**10100010
11010010	**0**11010010	**1**11010010
11111110	**1**11111110	**0**11111110

Serijska komunikacija - slanje i primanje podataka 2/2

TI znači "Transmit Interrupt“. Kada program upiše bajt na serijski port, potrebno je da protekne neko vrijeme dok se svi biti pošalju preko serijske linije. Ako bi program upisao novi bajt na serijski port prije nego što je prethodni bajt čitav poslat, podaci koji se šalju bi bili neispravni jer bi bio poslat dio podataka iz prethodnog, a dio iz novog bajta. Zbog toga mikrokontroler daje programu do znanja da je poslao čitav posljednji bajt tako što TI postavlja na logičku 1. Kada je TI na logičkoj 1, program može da smatra da je serijski port “slobodan” i spreman da pošalje sljedeći bajt. To što je mikrokontroler postavio TI na logičku 1 će dovesti i do generisanja serijskog prekida, ako je dozvoljen.

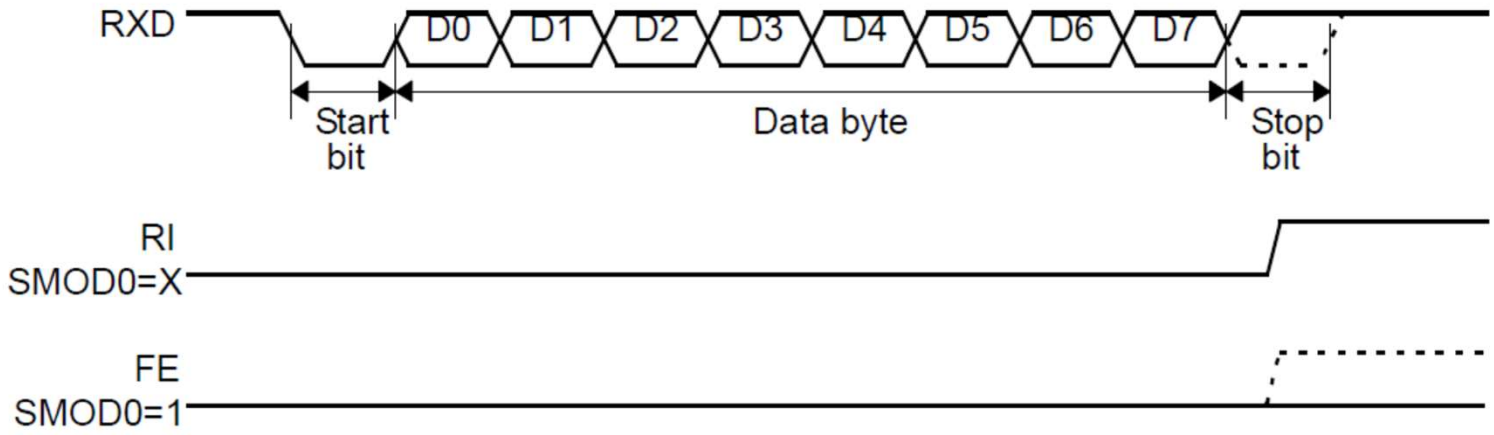
RI znači "Receive Interrupt“. Funkcioniše na sličan način kao i TI bit, ali s tim da pruža informaciju da je primljen bajt podataka. Svaki put kada mikrokontroler primi čitav bajt podataka preko serijskog porta, on će da postavi RI na logičku 1. Na taj način program zna da što prije mora da pročita bajt koji je dobio preko serijskog porta, da bi mogao da ga obradi prije nego što ga prepiše sljedeći bajt koji se primi. Kada mikrokontroler postavi RI na logičku 1 generisaće se serijski prekid, ako je dozvoljen.

Kontrola stop bita pri serijskoj komunikaciji 1/2

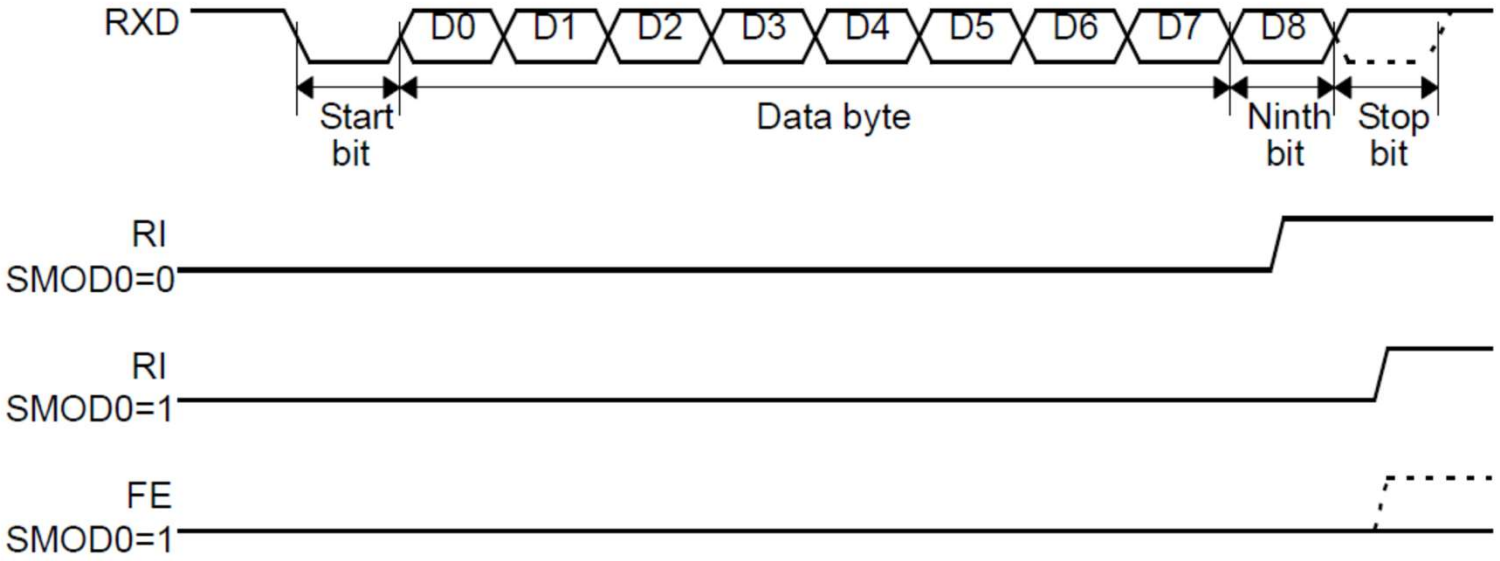


Kontrola stop bita pri serijskoj komunikaciji 2/2

Mod 1



Modovi 2 i 3



Izbor baud rate generatora

BDRCON - Baud Rate Control Register (9Bh)

Bit	Ime	Objašnjenje funkcije
7	-	Rezervisano
6	-	Rezervisano
5	-	Rezervisano
4	BRR	Baud Rate Run Control bit – pokrece i zaustavlja interni Baud Rate Generator
3	TBCK	Izbor Baud Rate Generatora za slanje podataka preko UART-a
2	RBCK	Izbor Baud Rate Generatora za primanje podataka preko UART-a
1	SPD	0 – koristi se spori Baud Rate Generator, 1 – koristi se Brzi Baud Rate Generator
0	SRC	Izbor Baud Rate Generatora za UART u modu 0

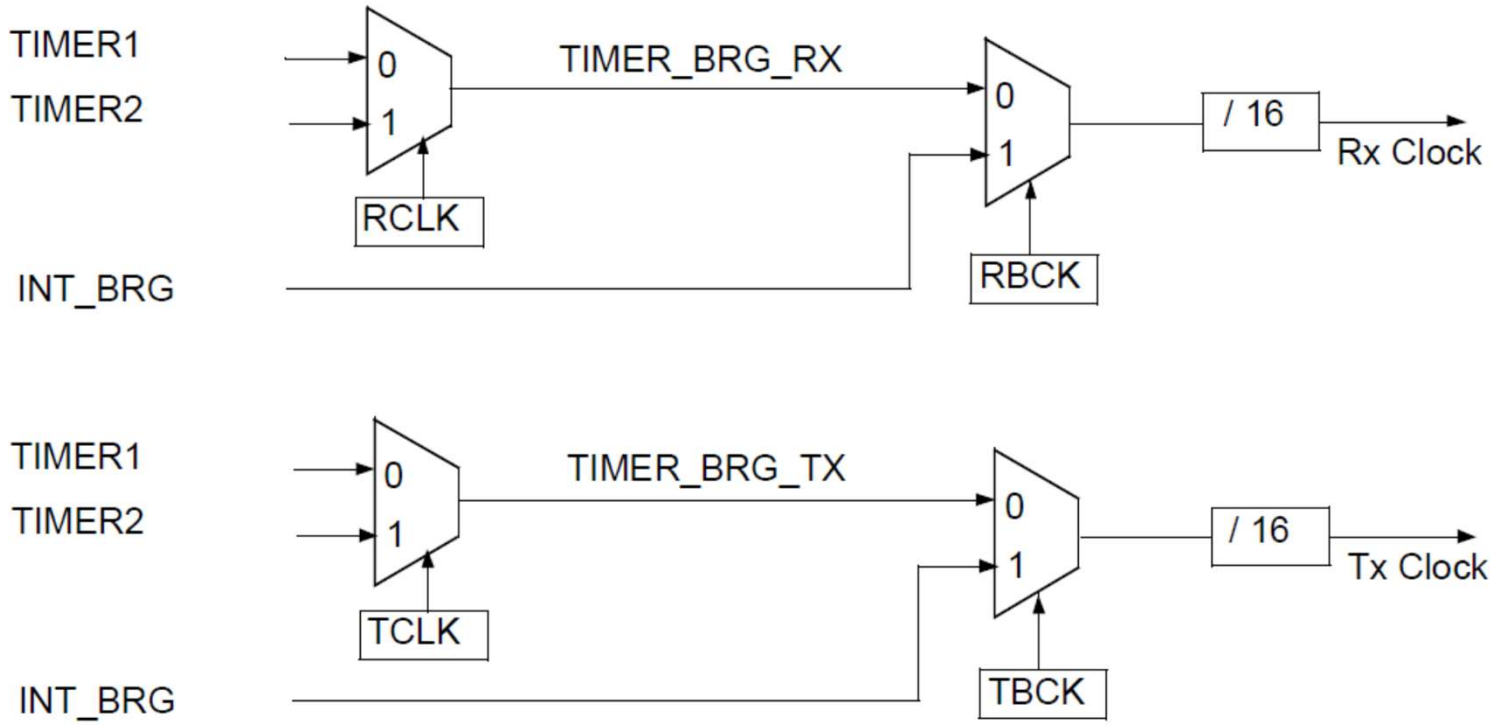
TBCK – Koristi se za definisanje Baud Rate Generatora za slanje podataka preko UART-a. Kada je 0, tajmer 1 ili tajmer 2 se koristi kao Baud Rate Generator, a kada je 1, koristi se interni Baud Rate Generator.

RBCK – Koristi se za definisanje Baud Rate Generatora za primanje podataka preko UART-a. Kada je 0, tajmer 1 ili tajmer 2 se koristi kao Baud Rate Generator, a kada je 1, koristi se interni Baud Rate Generator

SRC - Koristi se za izbor Baud Rate Generatora za UART u modu 0. Kada je 0, FOSC/12 je Baud Rate Generator (FCLK PERIPH/6 u X2 modu). Kada je 1, koristi se interni Baud Rate Generator za UART u modu 0.

Kod internog Baud Rate Generatora koristi se **BRL** (Baud Rate Reload) registar.

Izbor baud rate za serijsku komunikaciju u modovim 1 i 3



Definisanje brzine serijske komunikacije (baud rate) 1/6

Nakon što je definisan mod rada serijskog porta, program mora da konfiguriše i baude rate serijskog porta. Ovo se odnosi samo na modove 1 i 3. U modovima 0 i 2 baud rate je određen na osnovu frekvencije oscilatora (eventualno se, postavljanjem SRC bita iz BDRCON na 1, za baud rate u modu 0 može koristiti Baud Rate Generator pa se u tom slučaju i baud rate u modu 0 može konfigurisati).

U modovima 1 i 3, baud rate je određen na osnovu toga kako često dolazi do "overflova" timera 1, timera 2 ili internog baud rate generatora. Što češće dolazi do "overflowa" veći je baud rate. Ukoliko se koristi tajmer, prekid tajmera mora da bude isključen.

Definisanje brzine serijske komunikacije (baud rate) 2/6

Slučaj kada se baud rate definiše pomoću tajmera 1 ili 2

Opšti obrazac koji pokazuje kako se pomoću npr. tajmera 1 definiše baud rate je sljedeći:

$$\text{Mode 1, 3 Baud Rate} = \frac{2^{\text{SMOD1}}}{32} \cdot (\text{Timer 1 Overflow Rate})$$

Kada se pomoću timera definise baud rate, to se radi tako što se timer koristi kao tajmer u auto-reload modu. U tom slučaju baud rate se za tajmer 1 izračunava na sljedeći način

$$\text{Mode 1, 3 Baud Rate} = \frac{2^{\text{SMOD1}}}{32} \cdot \left(\frac{F_{\text{PER}}}{6 \cdot [256 - \text{TH1}]} \right)$$

Ako je timer 1 konfigurisan kao timer u auto-reload modu, vrijednost koju treba upisati u TH1 da bi se dobio odgovarajući baud rate, dobija se na sljedeći način:

$$\text{TH1} = 256 - \frac{2^{\text{SMOD1}} \cdot F_{\text{PER}}}{32 \cdot 6 \cdot \text{Baud Rate}}$$

Napomena: Ovdje se F_{PER} odnosi na tajmer koji se koristi za definisanje baud rate-a

Definisanje brzine serijske komunikacije (baud rate) 3/6

Primjer

Imamo kristal od 11.059MHz i želimo da definišemo baud rate od 19200 za serijski port u modu 1. Izračunati vrijednost koja se upisuje u TH1 ako je timer 1 konfigurisan kao tajmer u auto-relad modu i nije u X2 modu.

Ukoliko odlučimo da postavimo SMOD1 bit na 0 dobija se vrijednost za TH1:

$$TH1 = 256 - \frac{2^{SMOD1} \cdot F_{PER}}{32 \cdot 6 \cdot \text{Baud Rate}} = 256 - \frac{2^{0} \cdot (11059000/2)}{32 \cdot 6 \cdot 19200} = 256 - 1.5 = 254.5$$

Vrijednost koja treba da se upiše u TH1 je 254.5.
Ukoliko upišemo 254, dobijamo baud rate 14400, odnosno grešku od 25% po bitu
Ukoliko upišemo 255 dobijamo 28800, odnosno grešku od 50% po bitu

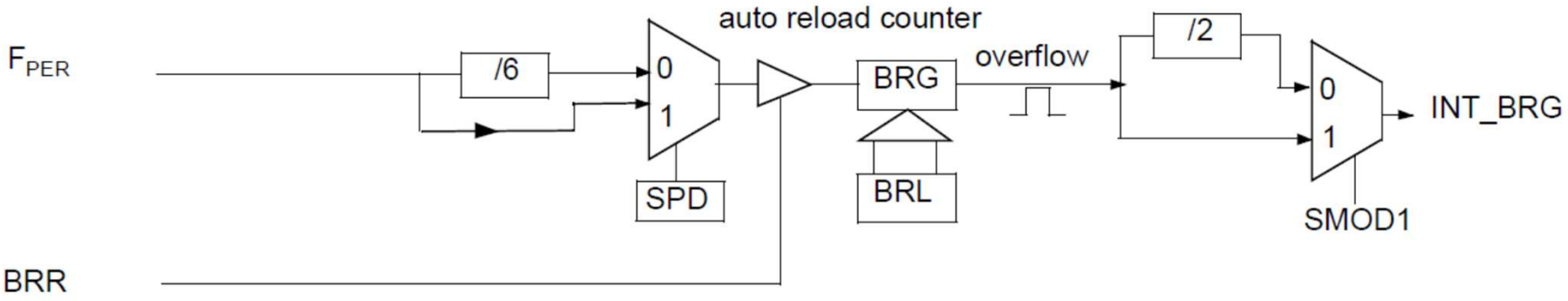
Zbog toga ćemo pokušati da izračunamo reload vrijednost za TH1, ako postavimo SMOD bit na 1.

$$TH1 = 256 - \frac{2^{SMOD1} \cdot F_{PER}}{32 \cdot 6 \cdot \text{Baud Rate}} = 256 - \frac{2^{1} \cdot (11059000/2)}{32 \cdot 6 \cdot 19200} = 256 - 3 = 253$$

Definisanje brzine serijske komunikacije (baud rate) 4/6

Slučaj kada se baud rate definiše pomoću internog baud rate generatora

Baud rate se definiše upisom vrednosti u BRL registar i postavljanjem SPD bita iz BDRCON registra, kao i SMOD1 bita iz PCON registra na odgovarajuću vrednost



Ako se koristi interni Baud Rate Generator, baud rate se izračunava na sledeći način

$$\text{Baud Rate} = \frac{2^{\text{SMOD1}}}{6^{(1\text{-SPD})} \cdot 32} \cdot \left(\frac{F_{PER}}{[256 - \text{BRL}]} \right)$$

Odnosno vrijednost koja treba da se upiše u BRL registar je:

$$\text{BRL} = 256 - \frac{2^{\text{SMOD1}} \cdot F_{PER}}{6^{(1-\text{SPD})} \cdot 32 \cdot \text{Baud Rate}}$$

Napomena: Ovdje se F_{PER} odnosi na clock periferije za Enhenced UART

Definisanje brzine serijske komunikacije (baud rate) 5/6

Primjer

Imamo kristal od 11.059MHz i želimo da definišemo baud rate od 19200 za serijski port u modu 1. Baud rate se definiše pomoću internog baud rate generatora i clock periferije nije u X2 modu.

Ukoliko odlučimo da postavimo SMOD1 bit na 1 i SPD bit na 0 dobija se vrijednost

$$\mathrm{BRL} = 256 - \frac{2^{\mathrm{SMOD1}} \cdot F_{\mathrm{PER}}}{6^{(1-\mathrm{SPD})} \cdot 32 \cdot \mathrm{Baud\ Rate}} = 256 - \frac{2^{1} \cdot (11059000/2)}{6^{(1-0)} \cdot 32 \cdot 19200} = 256 - 3 = 253$$

Definisanje brzine serijske komunikacije (baud rate) 6/6

Primjer

Imamo kristal od 24MHz i želimo da definišemo baud rate od 19200 za serijski port u modu 1. Baud rate se definiše pomoću internog baud rate generatora i clock periferije nije u X2 modu.

Ukoliko odlučimo da postavimo SMOD1 bit na 1 i SPD bit na 0 dobija se vrijednost

$$BRL = 256 - \frac{2^{SMOD1} \cdot F_{PER}}{6^{(1-SPD)} \cdot 32 \cdot \text{Baud Rate}} = 256 - \frac{2^1 \cdot (24000000/2)}{6^{(1-0)} \cdot 32 \cdot 19200} = 256 - 6.51 = 249.49$$

Vrijednost koja treba da se upiše u TH1 je 249.49.
Ukoliko upišemo 250, dobijamo baud rate 20833, odnosno grešku od 8,5% po bitu
Ukoliko upišemo 249, dobijamo baud rate 17857, odnosno grešku od 7% po bitu

Ukoliko postavimo SPD bit na 1:

$$BRL = 256 - \frac{2^1 \cdot (24000000/2)}{6^{(1-1)} \cdot 32 \cdot 19200} = 256 - 39.062 = 216.938 \Rightarrow BRL = 217$$

Ukoliko upišemo 217, dobijamo baud rate 19230, odnosno grešku od 0,2% po bitu

Upis i čitanje sa serijskog porta

Kada je serijski port konfigurisan na odgovarajući način, može da se koristi za slanje i primanje podataka. Slanje i primanje podataka se vrše upisom, odnosno očitavanjem, odgovarajućeg SFR.

Upis na serijski port

Da bi se poslao bajt podataka preko asinhrone serijske komunikacije, program samo treba da upiše taj bajt u SFR registar koji se zove **SBUF** (99h). Npr., ako želite da pošaljete broj 9 preko serijskog porta, dovoljno je da u programskom kodu napišete instrukciju:

MOV SBUF,#9

Kada se izvrši ova instrukcija, mikrokontroler će da počne da šalje zadati karakter preko serijskog porta. Naravno, slanje će da traje određeno vrijeme, koje je određeno preko definisanog baud rate-a. Pošto mikrokontroler nema izlazni bafer za serijsku komunikaciju, moramo da budemo sigurni da je prethodni bajt poslat prije nego što pošaljemo sljedeći.

Mikrokontroler daje informaciju o tome kada je završio prenos tako što postavlja **TI** bit iz registra SCON na logičku 1.

Čitanje sa serijskog porta

Da bi se pročitao bajt koji je primljen preko serijskog porta, samo je potrebno da se pročita vrijednost koja je upisana u SFR registar **SBUF** (99h), nakon što je mikrokontroler automatski postavio **RI** bit iz registra SCON na logičku 1.

Npr. ako se želi očitati primljeni podatak i smjestiti u akumulator, dovoljno je da se definiše sljedeći kod:

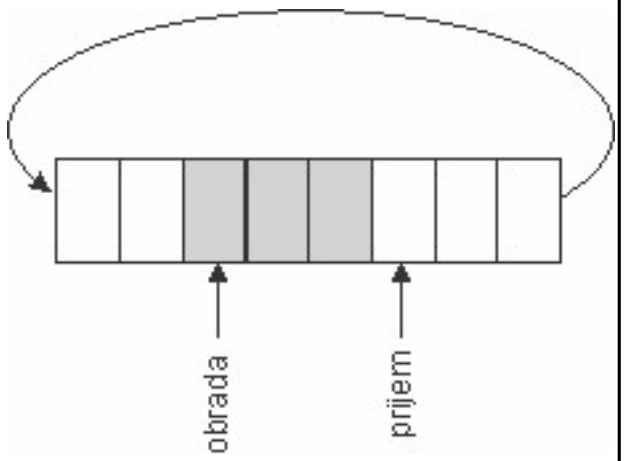
MOV A,SBUF

Korišćenje kružnog bafera za serijsku komunikaciju 1/2

AT89C51RC2 nema ni ulazni ni izlazni bafer za serijsku komunikaciju. To znači da kada se primi jedan bajt podataka, mi moramo odmah da ga očitamo iz SBUF i negdje smjestimo kako ga sljedeći primljeni bajt ne bi prepisao. Slično je i kod slanja podataka preko serijske komunikacije, gdje se bajtovi šalju jedan po jedan. Pošto se obično u programu šalju poruke koje se sastoje od više bajtova potrebno je negdje smjestiti sve podatke koji su određeni za slanje, odakle ćemo ih slati jedan po jedan, tako što ćemo ih upisivati u SBUF svaki put kada slanje bude dozvoljeno.

Zbog toga mora da se definiše poseban niz koji će predstavljati ulazni bafer i drugi niz koji će predstavljati izlazni bafer. Kada primimo bajt preko serijske komunikacije, mi ćemo ga odmah smještati u ulazni bafer, da ga slučajno ne bi prepisao sljedeći podatak koji primamo. Podatke koje smjestimo u bafer ćemo negdje u programu obradjivati, kada na to dodje red, i prema komunikacinom protokolu iz njih izvlačiti dobijene informacije. Slično je i sa izlaznim baferom u koji ćemo smještati čitave poruke koje na osnovu našeg komunikacionog protokola sadrže informacije koje želimo da pošaljemo. Te poruke, koje se sastoje od više bajtova, ćemo onda da očitavamo iz bafera i šaljemo jedan po jedan bajt.

Za serijsku komunikaciju se po pravilu definišu kružni baferi.

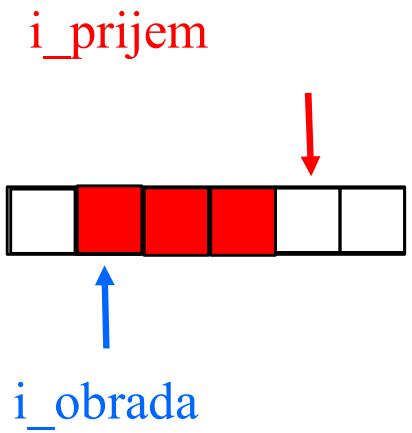


Kružni ulazni bafer

Korišćenje kružnog bafera za serijsku komunikaciju 2/2

Ulazni bafer:

- indeks za prijem – pokazuje na poziciju u koju treba da se upiše sljedeći primljeni podatak
- indeks za obradu – pokazuje na poziciju sa koje treba da se pročita sljedeći podatak koji će se obrađivati
- indeks za obradu "juri" indeks za prijem u krug do kraja bafera i onda opet od početka
- kada su oba indeksa na istoj poziciji, nema podataka za obradu
- ukoliko indeks za prijem dođe do indeksa za obradu:
  - to se dozvoli pa se u tom slučaju izgubi čitav sadržaj bafera
  - ili se indeks za prijem ne menja pa se ne upisuje novi podatak
  - ili se pomjeri indeks za obradu, kako bi se napravilo mesto za upis novog podatka, a čime se gubi najstariji primljeni-neobrađeni podatak



Izlazni bafer - isto kao ulazni samo imamo:

- indeks za upis – pokazuje na poziciju u koju treba da se upiše sljedeći podatak za slanje
- indeks za slanje – pokazuje na poziciju sa koje treba da se pročita sljedeći podatak koji će da se šalje
- indeks za slanje "juri" indeks za upis

Zadatak

Napisati program koji prima podatke preko serijskog porta i stavlja ih u kružni ulazni bafer u koji može da se smjesti 8 bajtova podataka, a takodje šalje podatke iz izlaznog kružnog bafera u koji se može smjestiti 8 bajtova podataka. Koristi se kristal od 11.059MHz, baud rate serijske komunikacije je 19,200 i prenosi se 8 bita podataka bez parity bita.

Rješenje:

Za konfigurisanje serijske komunikacije potrebno je obaviti sljedeće korake:

1. Konfigurisati mod rada 1 za serijski port
2. Konfigurisati Timer 1 kao timer u modu 2 (8-bit auto-reload).
3. Upisati u TH1 vrijednost 253 da bi se na osnovu vrijednosti SMOD1 bita dobio odgovarajući baud rate od 19,200.
4. Postaviti PCON.7 (SMOD1) na logičku 1 da bi se dobio dupli baud rate

Posto ćemo koristiti i serijski prekid potrebno ga je omogućiti, odnosno postaviti globalnu dozvolu prekida EA na 1 i dozvolu serijskog prekida ES na 1.

Programski kod rješenja 1/3

```
#include<REG51RC2.h>

#define TRUE                    1
#define FALSE                   0

volatile unsigned char i_UARTprijem=0;
volatile unsigned char i_UARTobrada=0;
volatile unsigned char i_UARTslanje=0;
volatile unsigned char i_UARTupis=0;
volatile unsigned char ul_bafer[8];
volatile unsigned char iz_bafer[8];

unsigned char bdata flagovi=0;
sbit zauzetaSerijska=flagovi^0;

void InicijalizacijaKontrolera(void)
{
EA=0;//GLOBALNA DOZVOLA PREKIDA
SCON=0x50;//SM0=0,SM1=1 => MOD 1 (8-bitni UART);  SM2=0, REN=1(dozvola primanja)
TMOD=0x20;//GATE1=0(bit 7), C/T1=0 (bit 6), mod 2(biti 4-5)
TR1=1;//DOZVOLA RADA TAJMERA 1
ET1=0;//UKIDANJE DOZVOLE PREKIDA TAJMERA 1
TL1=253;//VRIJEDNOST ZA PRVO BROJANJE TAJMERA
TH1=253;//RELOAD VRIJEDNOST
PCON|=0x80;//POSTAVLJANJE SMOD1 BITA NA LOGICKU 1
BDRCON&=0xE0; /NE KORISTI SE INTERNI BAUD RATE GENERATOR
ES=1; //DOZVOLA SERIJSKOG PREKIDA
EA=1;//GLOBALNA DOZVOLA PREKIDA
}
```

7	SM0/FE	9Fh	FE – kada se desi framing error pri prijemu ima vrijednost 1 (pristup kada je SMOD0=1). Ispravan stop bit ne resetuje njegovu vrijednost, to se radi softverski
			SM0 - Mod rada serijskog porta – bit 0 (pristup kada je SMOD0=0)
6	SM1	9Eh	Mod rada serijskog porta – bit 1
5	SM2	9Dh	Dozvola multiprocesorske komunikacije
4	REN	9Ch	Dozvola primanja. Ovaj bit mora biti na logičkoj 1 da bi se primali podaci.
3	TB8	9Bh	9-i bit za slanje u modu 2 i 3.
2	RB8	9Ah	9-i bit primljen u modu 2 i 3.
1	TI	99h	Transmit Flag. Na logičkoj 1 kada je bajt uspješno poslat.
0	RI	98h	Receive Flag. Na logičkoj 1 kada je bajt uspješno primljen.

$$TH1 = 256 - \frac{2^{SMOD1} \cdot F_{PER}}{32 \cdot 6 \cdot \text{Baud Rate}} = 256 - \frac{2^1 \cdot (11059000/2)}{32 \cdot 6 \cdot 19200} = 256 - 3 = 253$$

Programski kod rješenja 2/3

```
//*****************OBRADA SERIJSKOG PREKIDA*********************************
void SerijskiPrekid(void) interrupt 4
{
if (RI) {//PREKID SE DESIO JER JE PRIMLJEN KARAKTER
                RI=0;//BRISANJE FLAGA
                ul_bafer[i_UARTprijem]=SBUF;//PREUZIMANJE PRIMLJENOG KARAKTERA
                i_UARTprijem++;//INKREMENTIRANJE INDEKSA
                i_UARTprijem&=0x07;
   }
if (TI) {//PREKID SE DESIO JER JE POSLAT KARAKTER
                TI=0;//BRISANJE FLAGA
                if(i_UARTslanje!=i_UARTupis)//U KRUZNOM BAFERU IMA JOS PODATAKA ZA SLANJE
                {
                                SBUF=iz_bafer[i_UARTslanje];//INICIRANJE SLANJA BAJTA
                                i_UARTslanje++;//INKREMENTIRANJE INDEKSA
                                i_UARTslanje&=0x07;
                }
                else zauzetaSerijska=FALSE; //NIJE U TOKU SLANJE BAJTA PREKO SERIJSKOG PORTA
   }
}
```

Programski kod rješenja 3/3

```
//*********************FUNKCIJA ZA INICIRANJE SLANJA PREKO SERIJSKOG PORTA*********
void SlanjeSerijskom(void)
{
if(i_UARTslanje!=i_ UART upis && !zauzetaSerijska)//U BAFERU IMA PODATAKA ZA SLANJE I MOGU DA SE SALJU
                {
                                zauzetaSerijska=TRUE;//U TOKU JE SLANJE BAJTA PREKO SERIJSKOG PORTA
                                SBUF=iz_bafer[i_UARTslanje];//INICIRANJE SLANJA BAJTA
                                i_UARTslanje++;//INKREMENTIRANJE INDEKSA
                                i_UARTslanje&=0x07;
                }
}


//************************************MAIN*****************************************
void main(void)
{
InicijalizacijaKontrolera();
while(1)
{
SlanjeSerijskom();
}
}
```

Protokoli za asinhronu serijsku komunikaciju 1/2

Prema tome kako je definisan protokol

- Standardni protokol – definisan određenim standardom
- Custom protokol – definisan za konkretan uređaj/primenu kako bi se informacije od značaja razmenile na način koji je u skladu sa funkcionalnošću uređaja

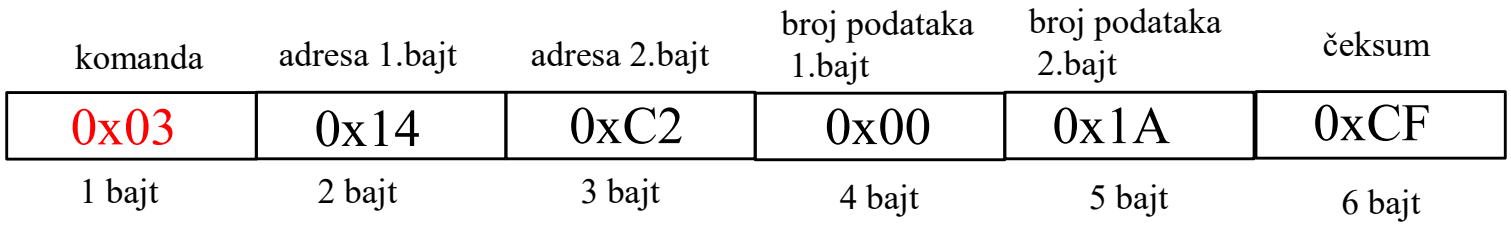
Prema načinu na koji su organizovani podaci u bajtove pri razmene informacija, najčešće se koriste:

- Binarni – svaki bajt koji se razmeni predstavlja dio podatka u njegovom binarnom zapisu, tako da se npr. za jedan 16-bitni celi broj pošalju 2 bajta podataka
- ASCII – podaci koji se razmenjuju organizuju se u tekst koji se sastoji od brojeva koji predstavljaju informacije, kao i slova i specijalnih karaktera koji identifikuju karakteristična mesta u poruci, kao sto su komanda ili početak/kraj poruke

Protokoli za asinhronu serijsku komunikaciju 2/2

Primer: Definisati protokol za poruku koja predstavlja komandu koja traži od uređaja da kao odgovor vrati određeni broj celobrojnih podataka iz memorije. Poruka sadrži kod komande, početnu adresu memorijskog prostora sa koga se očitava niz celobrojnih podataka i broj podataka koji se očitava.

Binarni protokol:



Pošalji podatke od adrese: 0x14C2 -> 5314
Pošalji ukupno podataka: 0x001A -> 26

ASCII protokol: