



Mikroračunarski sistemi

20ER6004



Serijska komunikacija

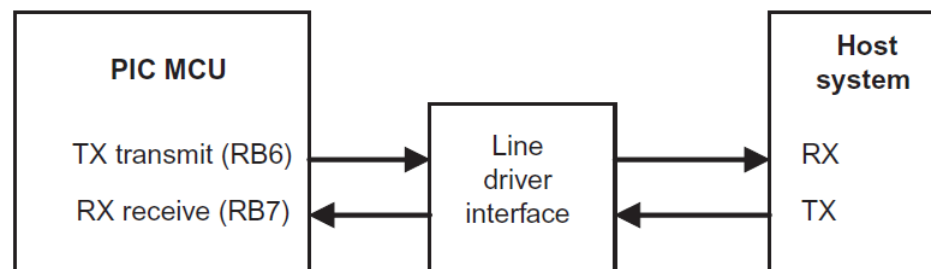


Serijska komunikacija MCU

- Universal Synchronous Asynchronous Receiver Transmitter (USART)
- Serial Peripheral Interface (SPI)
- Inter-Integrated circuit (I²C)
- Controller Area Network (CAN)
- Local Interconnect Network (LIN)
- Universal Serial Bus (USB)
- Ethernet
- Infrared
- Radio
- Wi-Fi
- Zigbee

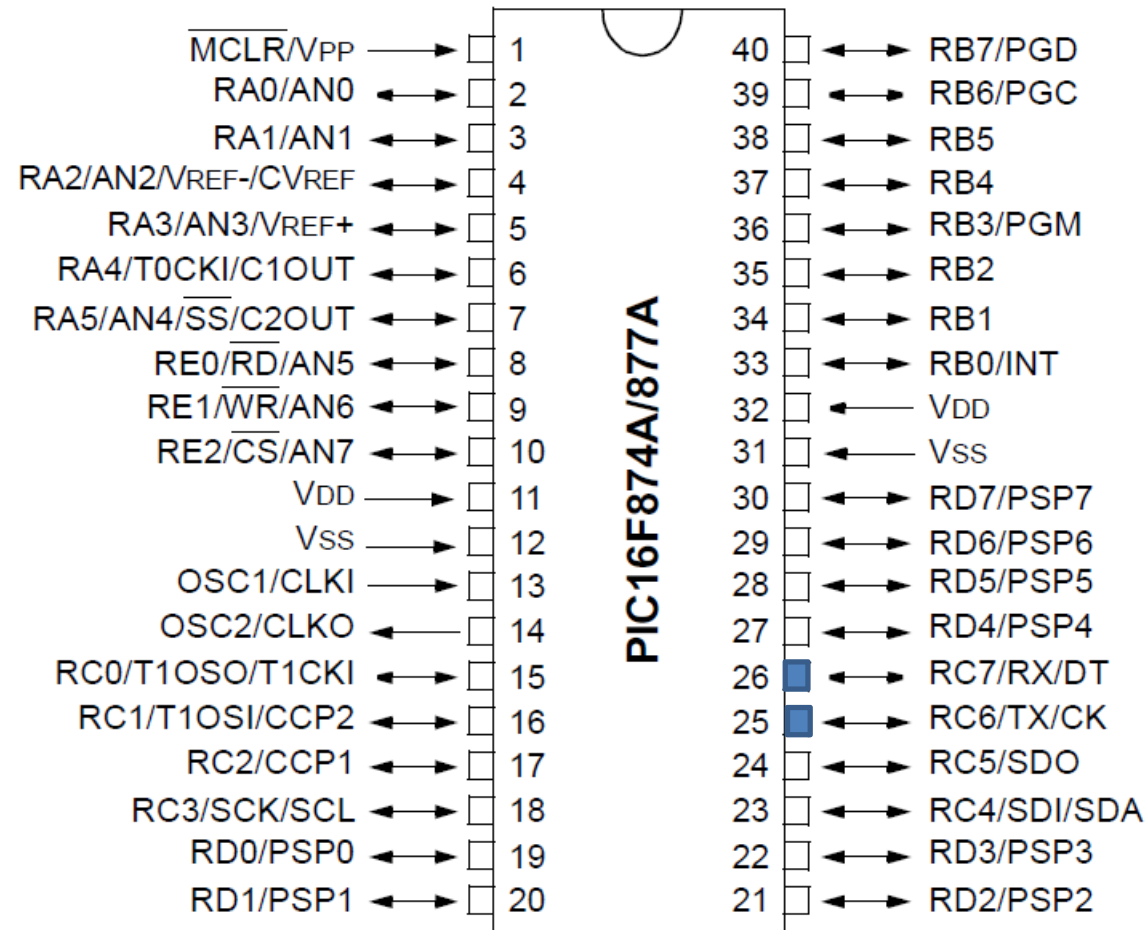
USART (RS-232)

- Koriste se 2 pina (RC6 i RC7 kod PIC 16F877A)
- Kod sinhronog prenosa, RC6 se koristi za prenos takta, a RC7 za prenos podataka
- U asinhronom režimu podaci se simultano mogu slati (RC6/Tx) i primiti (RC7/Rx)
- Zbog potrebe za višim (i simetričnim) naponskim nivoima, potrebna je eksterna komponenta za napajanje linija (TTL nivou se mogu koristiti za rastojanja do 1m, do 10m koristi se $\pm 12V$, a do 100m $\pm 25V$)



PIC 16F87X

pinovi za serijsku komunikaciju



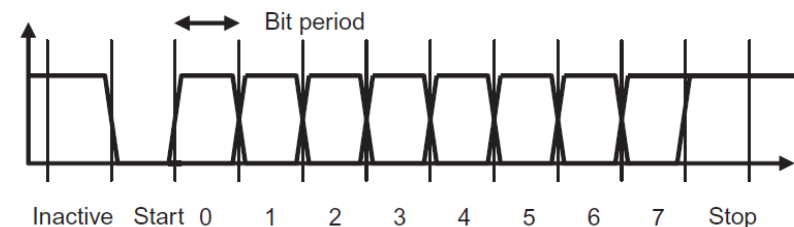
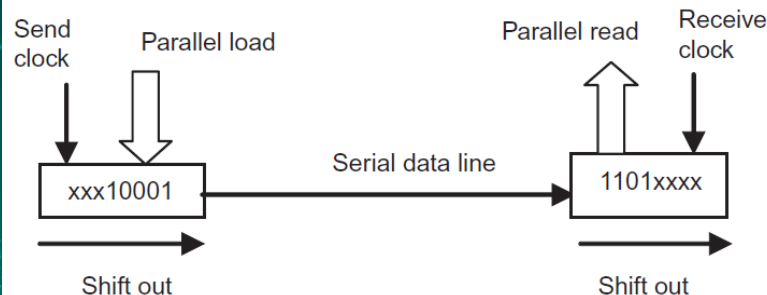
PIC 16F87X registri

RCSTA – Receive Status & Control
TXSTA – Transmit Status & Control
SPBRG – Baud Rate Generator
TXREG – predajni bafer
RCREG – prijemni bafer

Indirect addr. ^(*)	00h	Indirect addr. ^(*)	80h	Indirect addr. ^(*)	100h	Indirect addr. ^(*)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD ⁽¹⁾	08h	TRISD ⁽¹⁾	88h		108h		188h
PORTE ⁽¹⁾	09h	TRISE ⁽¹⁾	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved ⁽²⁾	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved ⁽²⁾	18Fh
T1CON	10h		90h	General Purpose Register 16 Bytes	110h	General Purpose Register 16 Bytes	190h
TMR2	11h	SSPCON2	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPAD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h		95h		115h		195h
CCPR1H	16h		96h		116h		196h
CCP1CON	17h		97h		117h		197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah		11Ah		19Ah
CCPR2L	1Bh		9Bh		11Bh		19Bh
CCPR2H	1Ch	CMCON	9Ch		11Ch		19Ch
CCP2CON	1Dh	CVRCON	9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
General Purpose Register 96 Bytes	20h	General Purpose Register 80 Bytes	A0h	General Purpose Register 80 Bytes	120h	General Purpose Register 80 Bytes	1A0h
		accesses 70h-7Fh	EFh F0h	accesses 70h-7Fh	16Fh 170h	accesses 70h - 7Fh	1EFh 1F0h
Bank 0	7Fh	Bank 1	FFh	Bank 2	17Fh	Bank 3	1FFh

USART

- Tx je na visokom nivou kada nema podataka za slanje
- Kada se napuni bafer za slanje preko serijske komunikacije (**TXREG**, 19h), podaci se automatski šalju (uz dodavanje start i stop bitova)
- Prvo se šalje bit najmanje težine
- Prijem počinje nakon detekcije opadajuće ivice start bita (semplovanje počinje nakon 1.5 bitskih perioda)
- Po završetku prijema jednog bajta, javlja se prekid koji signalizira da treba pročitati **RCREG** (1Ah) pre nego što se upiše sledeći bajt
- Brzina slanja/prijema, tj brzina rada šift-registra definisana je sadržajem **SPBRG** (*Baud Rate Generator Register*, 99h), koji ima postskalera za deljenje izlazne frekvencije sa 16 ili 64, zavisno od stanja **BRGH** kontrolnog bita u **TXSTA** registru (*Transmit Status and Control Register*, 98h)





TXSTA

(Transmit Status and Control Register)

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)

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)



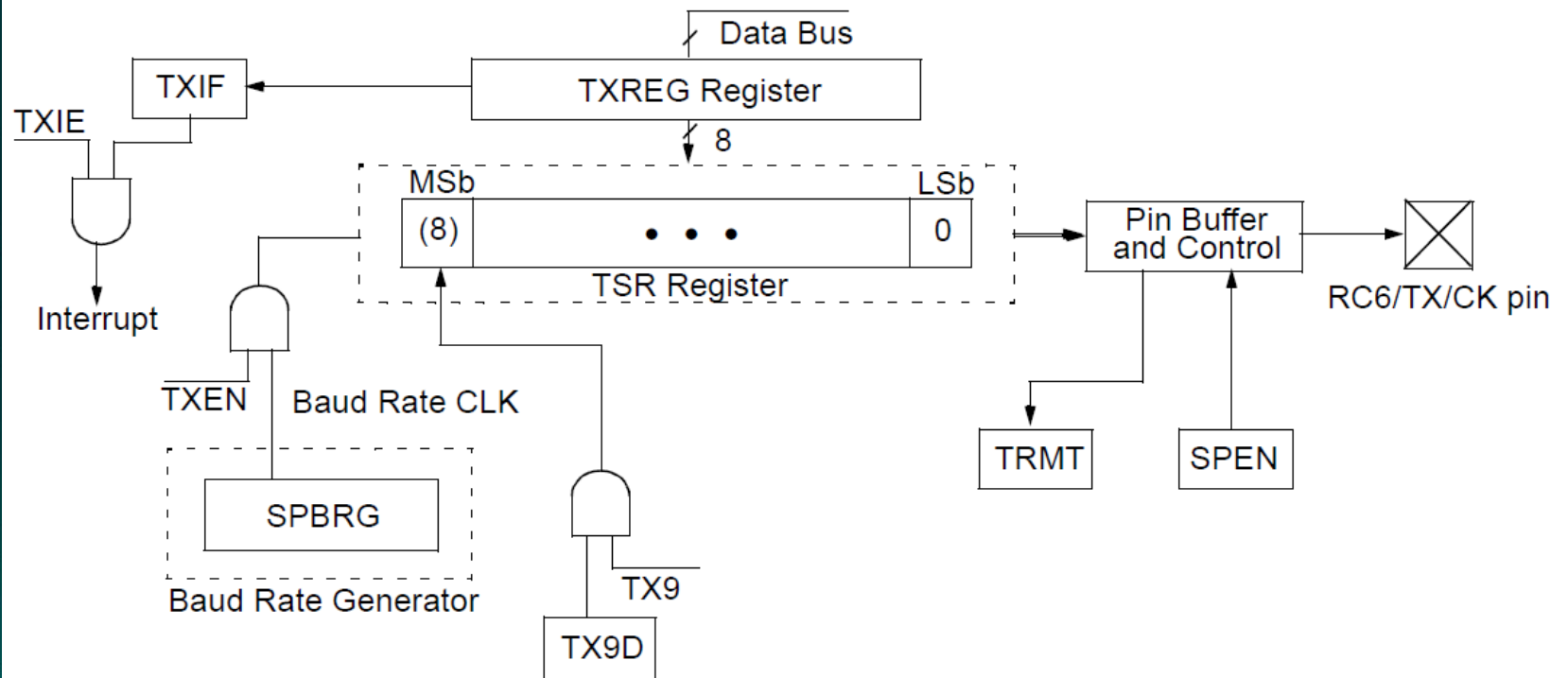
Brzina komunikacije

- Bitovi **SPEN** (**RCSTA**[7]) i **TRISC** [7:6] moraju biti postavljeni da bi se omogućio USART
- Mogući su sledeći modovi:
 - Asinhroni (full-duplex)
 - Sinhroni – master (half-duplex)
 - Sinhroni – slave (half-duplex)
- Brzina slanja/prijema zavisi od frekvencije osnovnog oscilatora (F_{OSC}), vrednosti upisane u SPBRG registru i stanja BRGH bita po sledećoj formuli:

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = $F_{OSC}/(64 (X + 1))$	Baud Rate = $F_{OSC}/(16 (X + 1))$
1	(Synchronous) Baud Rate = $F_{OSC}/(4 (X + 1))$	N/A

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

Blok diagram slanja

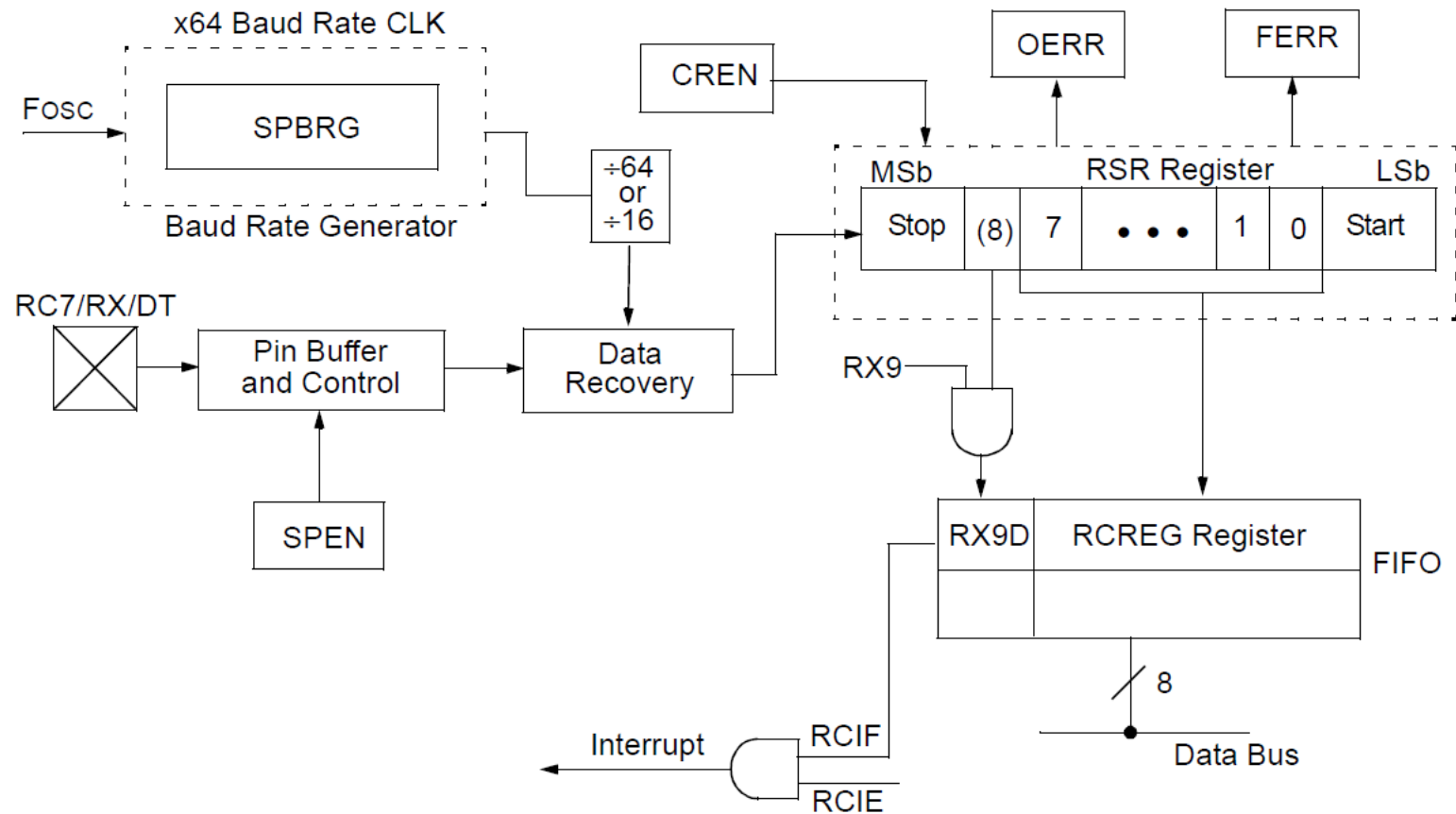




Koraci za USART slanje

- Inicijalizovati **SPBRG** na željenu brzinu rada (*baud rate*). Ako je potrebna veća brzina, postaviti **BRGH**.
- Omogućiti asinhroni serijski port resetovanjem **SYNC** i postavljanjem **SPEN** bitova.
- Ako su prekidi prihvatljivi u tekućem režimu rada, postaviti bit **TXIE**.
- Ako je potrebno prenositi 9 bitova, postaviti **TX9** bit
- Omogućiti slanje postavljanjem bita **TXEN**, što takođe postavlja i TXIF
- Ako se koriste 9 bitova, 9-ti bit upisati u **TX9D**
- Učitati bajt za slanje u **TXREG** (čime se pokreće slanje)
- Ako se koriste prekidi, proveriti da li su bitovi **GIE** (*Global Interrupt Enable*) i **PEIE** (*Peripheral Interrupt Enable*) u registru **INTCON** postavljeni

Blok diagram penerima



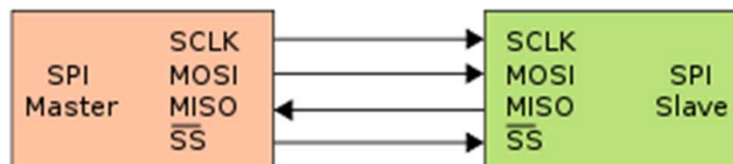


Koraci za USART prijem

- Inicijalizovati SPBRG **SPBRG** na željenu brzinu rada (*baud rate*). Ako je potrebna veća brzina, postaviti **BRGH**.
- Omogućiti asinhroni serijski port resetovanjem **SYNC** i postavljanjem **SPEN** bitova.
- Ako su prekidi prihvatljivi u tekućem režimu rada, postaviti bit **RCIE**.
- Ako je potrebno prenositi 9 bitova, postaviti **RX9** bit
- Omogućiti prijem postavljanjem bita **CREN**
- Fleg bit **RCIF** se postavlja kada je završen prijem, a prekid se generiše, ako postavljen **RCIE**
- Pročitati **RCSTA**, da bi se očitao 9. bit (ako je omogućen) i utvrdilo da li postoji greška u prijemu
- Očitati 8 primljenih bitova iz **RCREG**
- Ako se desila greška, obrisati je resetovanjem bita **CREN**
- Ako se koriste prekidi, proveriti da li su bitovi **GIE** (*Global Interrupt Enable*) i **PEIE** (*Peripheral Interrupt Enable*) u registru **INTCON** postavljeni

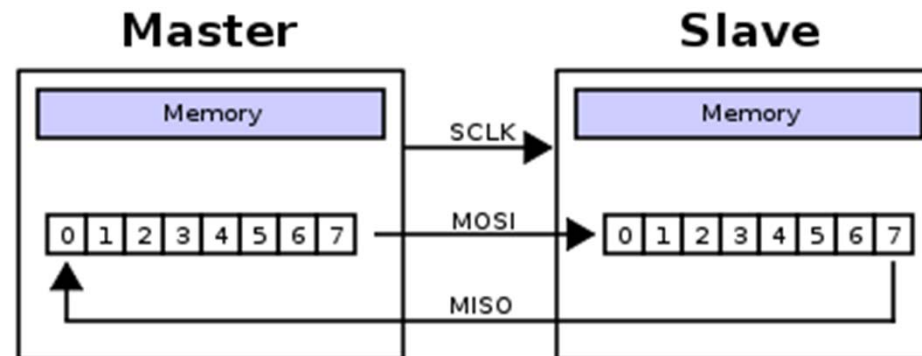
SPI magistrala

- **Serial Peripheral Interface (SPI)** je sinhroni serijski komunikacioni interfejs za komunikaciju na kratkim rastojanjima (najčešće u okviru *embedded* sistema)
- Razvila Motorola sredinom 80-tih
- Full-duplex komunikacija sa master-slave arhitekturom (samo 1 master)
- Master inicira komunikaciju; u konfiguraciji više *slave* uređaja, selekcija se vrši SS (*Slave Select*) signalom
- Magistrala sa 4 linije:
 - SCLK – (*Serial Clock*) izlaz iz mastera
 - MOSI – (*Master Output Slave Input*) izlazni podaci mastera
 - MISO – (*Master Input Slave Output*) izlazni podaci *slave*-a
 - SS – (*Slave Select*) izlaz iz mastera; selekcija *slave*-a niskim nivoom

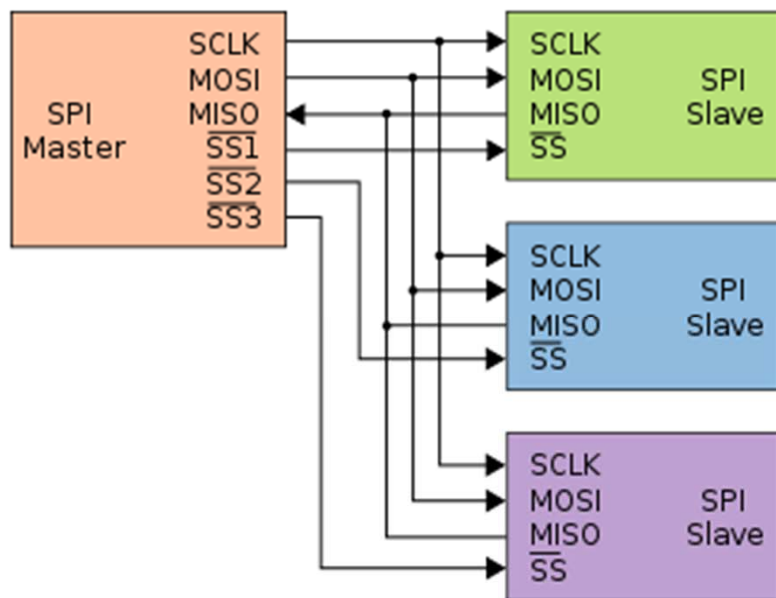


SPI prenos podataka

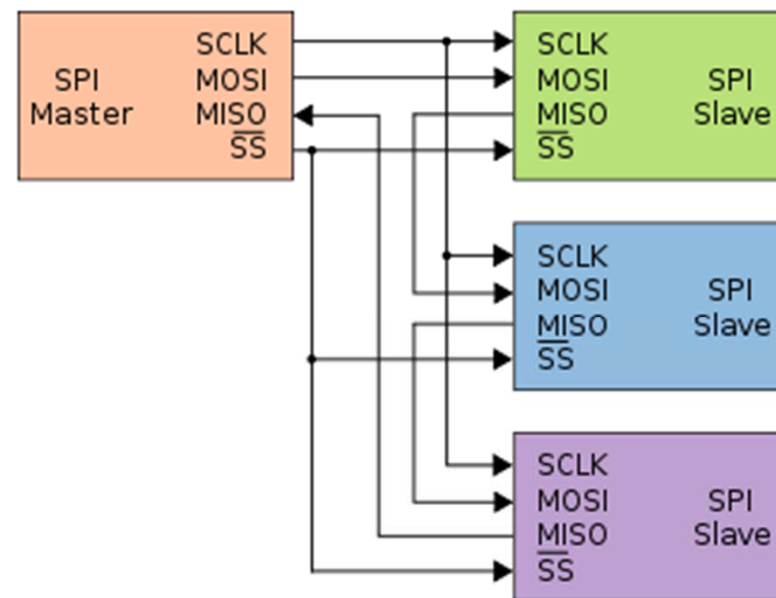
- Master prvo konfigurira takt, a zatim selektuje *slave* uređaj
- Nekada je potrebno da master sačeka izvesno vreme pre aktiviranja takta
- Tokom svakog otkucaja takta ostvaruje se full-duplex komunikacija (master šalje jedan bit preko MOSI linije, koji *slave* prihvata, dok istovremeno *slave* šalje jedan bit preko MISO linije ka masteru; komunikacija je dvosmerna čak i kada samo jedan ima podatke za slanje)
- Komunikacija uključuje dva pomeračka registra, povezana u virtuelni prsten
- Podaci se obično šalju od msb ka lsb ; nakon onoliko taktova kolika je dužina pomeračkih registara, master i *slave* su razmenili njihovu sadržinu (a nakon toga se pune nove vrednosti i proces nastavlja)



SPI konfiguracije



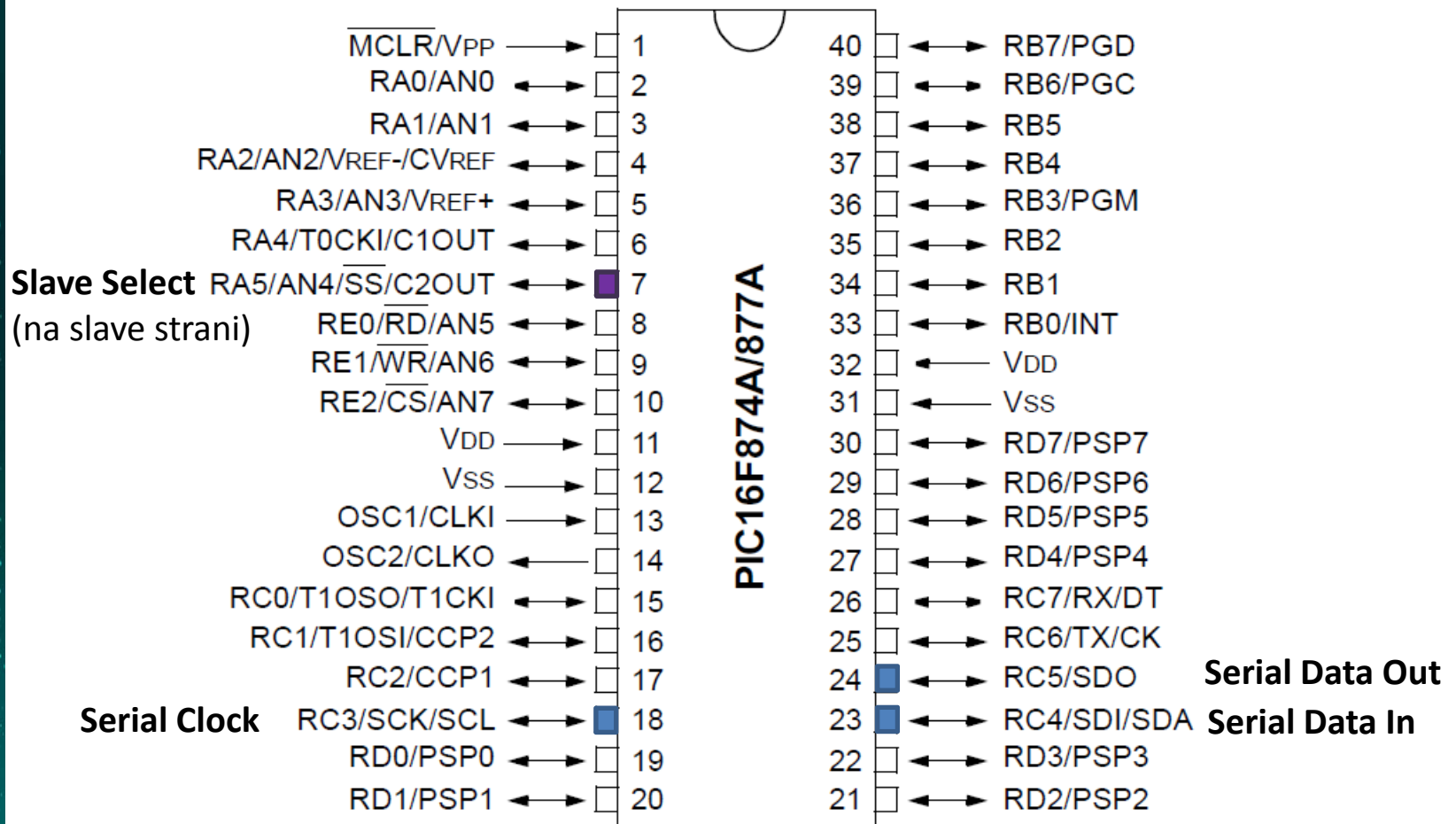
Nezavisni *slave* uređaji



Daisy-chain

PIC 16F87X

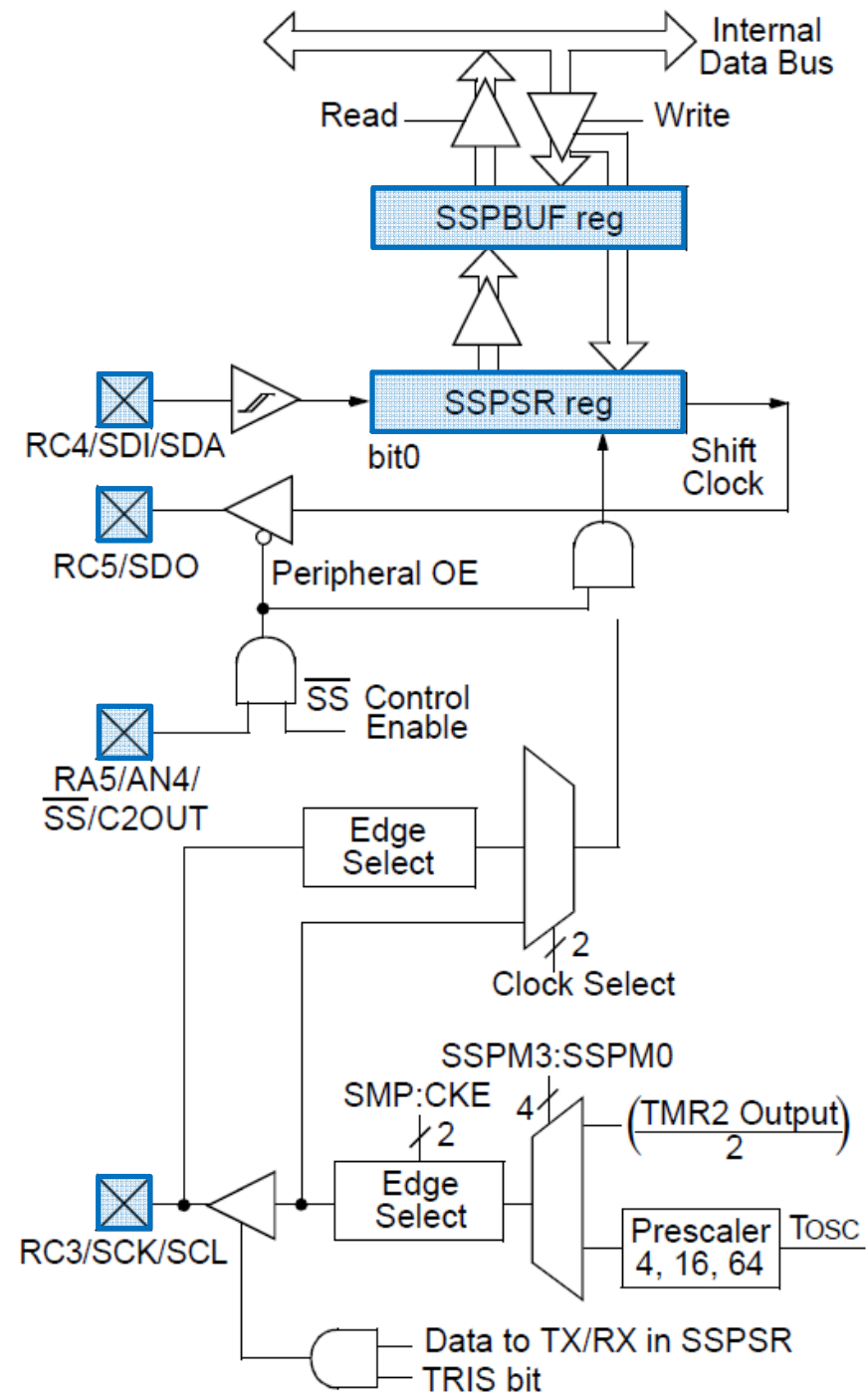
pinovi za SPI komunikaciju



Blok diagram SPI moda

Registri:

- SSP Control Register (SSPCON)
- SSP Status Register (SSPSTAT)
- Serial Rec/Trans Buffer (SSPBUF)
- SSP Shift Register (SSPSR)



SSPSTAT (94h)

statusni registar

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/ \bar{A}	P	S	R/ \bar{W}	UA	BF
bit 7							bit 0

- bit 7 **SMP:** Sample bit
SPI Master mode:
 1 = Input data sampled at end of data output time
 0 = Input data sampled at middle of data output time
SPI Slave mode:
 SMP must be cleared when SPI is used in Slave mode.
- bit 6 **CKE:** SPI Clock Select bit
 1 = Transmit occurs on transition from active to Idle clock state
 0 = Transmit occurs on transition from Idle to active clock state
Note: Polarity of clock state is set by the CKP bit (SSPCON1<4>).
- bit 5 **D/ \bar{A} :** Data/Address bit
 Used in I²C mode only.
- bit 4 **P:** Stop bit
 Used in I²C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.
- bit 3 **S:** Start bit
 Used in I²C mode only.
- bit 2 **R/ \bar{W} :** Read/Write bit information
 Used in I²C mode only.
- bit 1 **UA:** Update Address bit
 Used in I²C mode only.
- bit 0 **BF:** Buffer Full Status bit (Receive mode only)
 1 = Receive complete, SSPBUF is full
 0 = Receive not complete, SSPBUF is empty

SSPCON1 (14h)

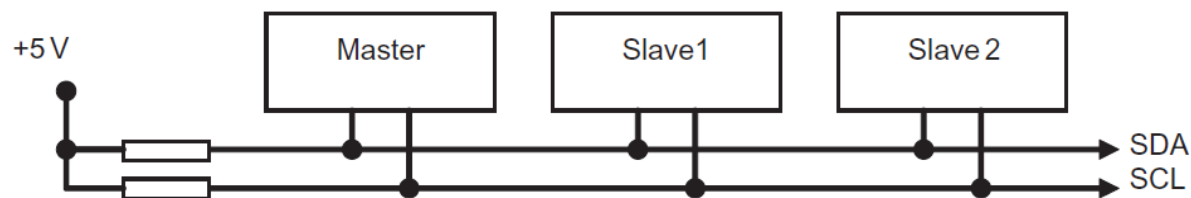
kontrolni registar 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

- bit 7 **WCOL:** Write Collision Detect bit (Transmit mode only)
 1 = The SSPBUF register is written while it is still transmitting the previous word. (Must be cleared in software.)
 0 = No collision
- bit 6 **SSPOV:** Receive Overflow Indicator bit
SPI Slave mode:
 1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. (Must be cleared in software.)
 0 = No overflow
Note: In Master mode, the overflow bit is not set, since each new reception (and transmission) is initiated by writing to the SSPBUF register.
- bit 5 **SSPEN:** Synchronous Serial Port Enable bit
 1 = Enables serial port and configures SCK, SDO, SDI, and \overline{SS} as serial port pins
 0 = Disables serial port and configures these pins as I/O port pins
Note: When enabled, these pins must be properly configured as input or output.
- bit 4 **CKP:** Clock Polarity Select bit
 1 = Idle state for clock is a high level
 0 = Idle state for clock is a low level
- bit 3-0 **SSPM3:SSPM0:** Synchronous Serial Port Mode Select bits
 0101 = SPI Slave mode, clock = SCK pin. \overline{SS} pin control disabled. \overline{SS} can be used as I/O pin.
 0100 = SPI Slave mode, clock = SCK pin. \overline{SS} pin control enabled.
 0011 = SPI Master mode, clock = TMR2 output/2
 0010 = SPI Master mode, clock = Fosc/64
 0001 = SPI Master mode, clock = Fosc/16
 0000 = SPI Master mode, clock = Fosc/4
Note: Bit combinations not specifically listed here are either reserved or implemented in I²C mode only.

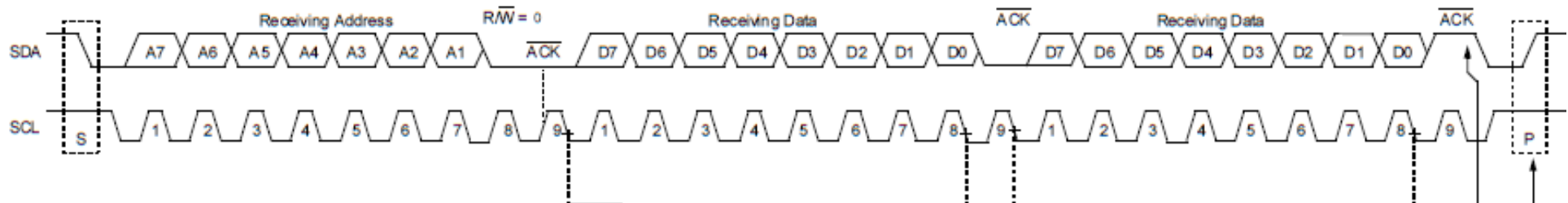
I²C magistrala

- **I²C (*Inter-Integrated Circuit*)** je sinhrona, multi-master serijska magistrala sa skretnim paketima
- Uveo *Philips Semiconductor* 1982.
- Koristi se za povezivanje sporih periferija na kratkim rastojanjima (u okviru ploče) na procesore i mikrokontrolere
- Koristi dve bidirekzione linije: SCL (clock do 1MHz) i SDA (data)
- Referentni model ima 7-bitne adrese, ali postoji i 10-bitna ekstenzija, što omogućuje masteru da poveže do 1023 *slave* uređaja
- Brzina prenosa podataka od 10kb/s, preko 100kb/s i 400kb/s, do 1Mb/s i 3.4Mb/s



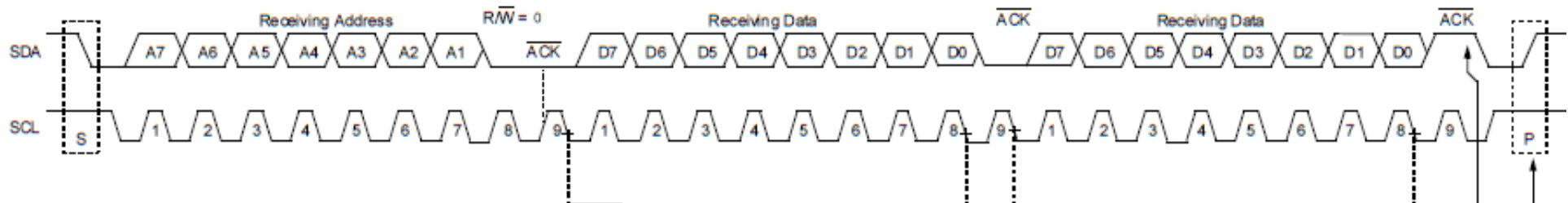
I²C komunikacija

- Master je inicijalno u modu za slanje i šalje START signal, za kojim sledi 7 bitova adrese *slave*-a sa kojim želi da komunicira
- Zatim sledi 1 bit koji definiše da li želi da upiše (0) ili da čita (1) podatak od *slave*-a
- Ako je *slave* na magistrali, odgovara ACK bitom (niska aktivna vrednost)
- Master nastavlja sa slanjem ili prijemom, a *slave* ga prati odgovarajućim režimom
- I adresa i podaci šalju se počinjući od msb
- START bit se signalizira prelaskom SDA sa visokog na niski nivo, dok je SCL (clock) na visokom, STOP prelaskom SDA sa niskog na visoko, dok se sve ostale promene SDA javljaju dok je SCL na niskom nivou



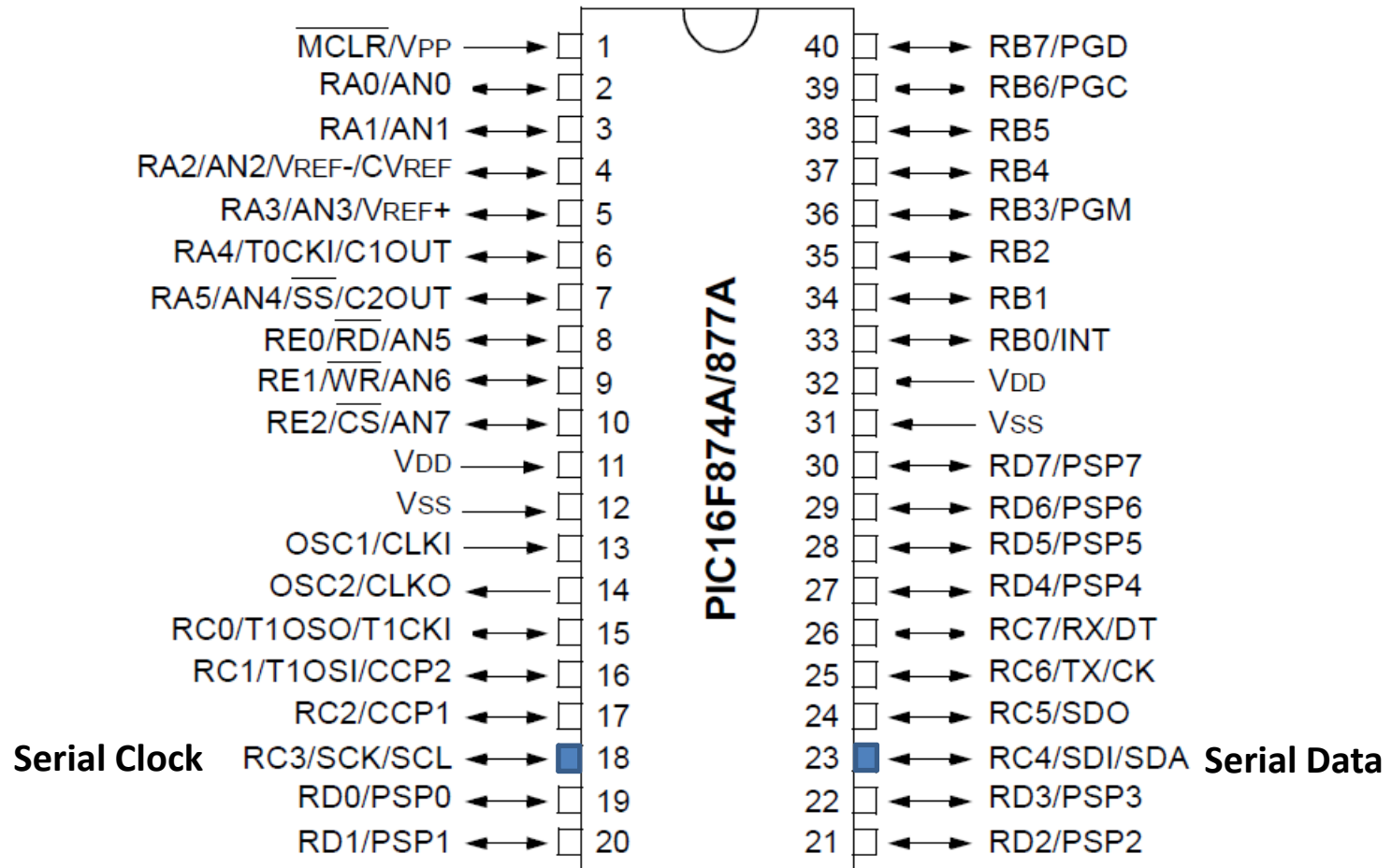
I²C komunikacija

- Ako master želi da šalje podatke *slave*-u (*write*), on uzastopno šalje bajtove, dok *slave* šalje ACK bitove nakon svakog bajta
- Ako master želi da prima podatke da *slave*-a (*read*), *slave* uzastopno šalje bajtove, dok master šalje ACK bitove nakon svakog bajta, osim poslednjeg
- Master završava poruku slanjem STOP signala ukoliko želi da završi transakciju ili šalje novi START signal da bi zadržao kontrolu nad magistralom za narednu poruku (ako se transakcija sastoji od više poruka)

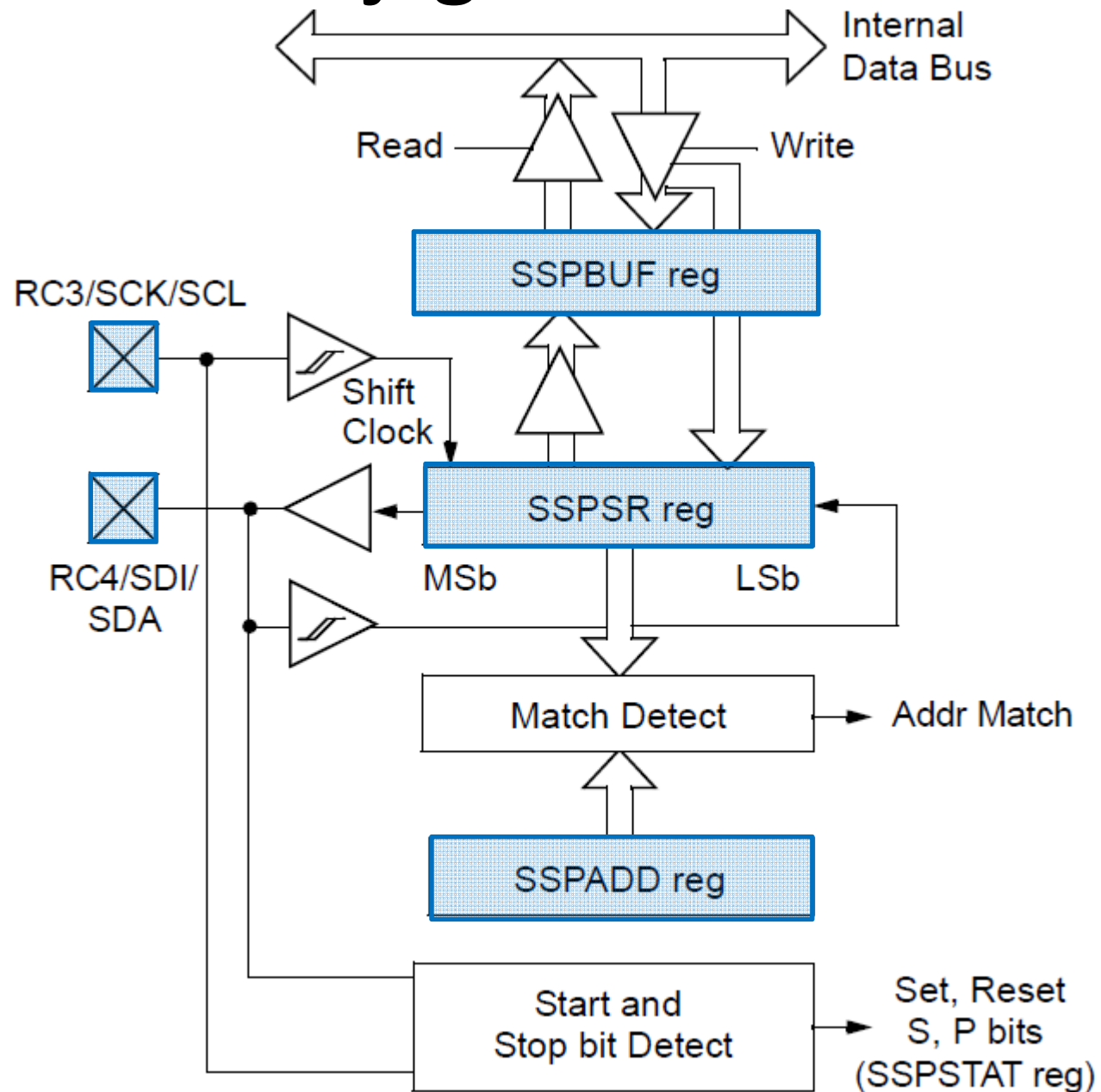


PIC 16F87X

pinovi za I²C komunikaciju



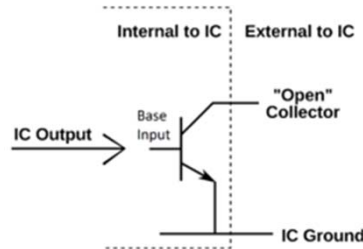
Blok diagram I²C moda





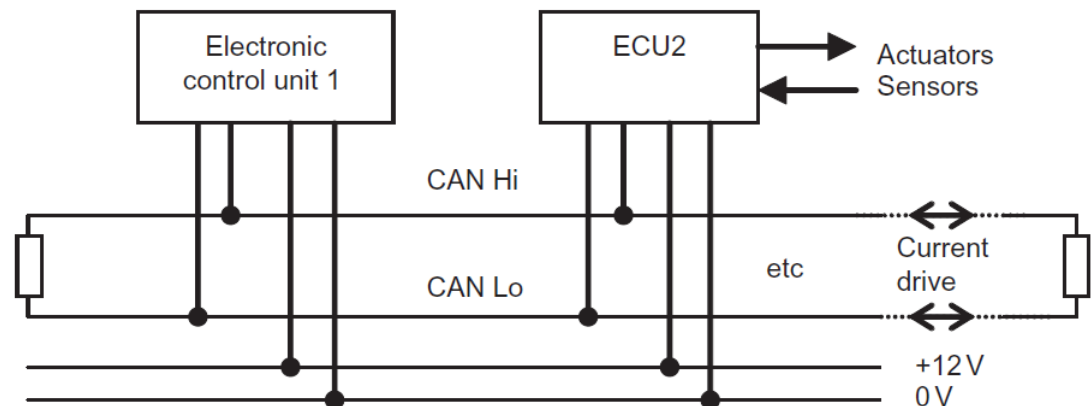
Mrežni (žičani) linkovi

- CAN magistrala
- LIN magistrala
- USB
- Ethernet



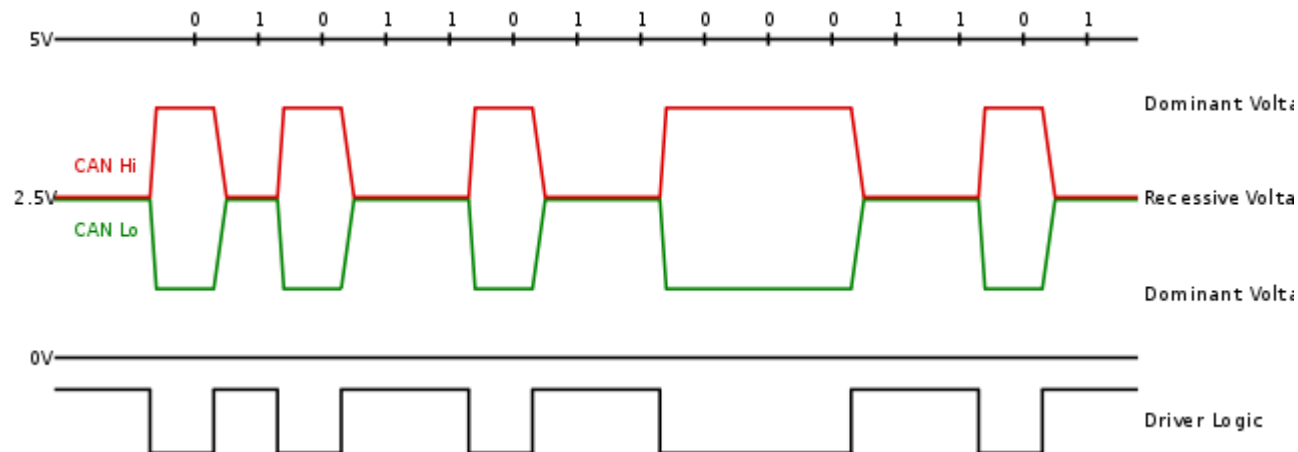
CAN magistrala

- **Controller Area Network (CAN)** razvijena je 80-tih godina (protokol je zvanično objavljen 1986.) za komunikaciju podсистema u motornim vozilima (kontrola motora, vazdušni jastuci, kočioni sistem,...)
- Dvožična magistrala (upredena parica, karakteristične impedanse 120Ω) koja povezuje sve ECU (*electronic control units*) korišćenjem open-collector (žičanih „ili“) veza
- ISO 11898-2 (high speed CAN) omogućuje brzinu prenosa podataka do 1MB/s (5MB/s za CAN-FD), ima linearnu topologiju i na oba kraja je zatvorena otpornikom od 120Ω (max. dužina 40m)
- ISO 11898-3 (low speed ili fault tolerant CAN) omogućuje brzinu prenosa podataka do 125kB/s, ima linearnu topologiju, topologiju zvezde ili višestruke zvezde i na krajevima je zatvorena otpornikom od 100Ω (do 500m)



CAN magistrala

- Kada se prenosi 0, **CAN High** raste ka 5V, a **CAN Low** pada ka 0V (tj. povećava se razlika na $\sim 2V$), a kada se prenosi 1 ne napajaju se (obe linije na istom potencijalu)
- Poruke se sastoje od ID-a (koji određuje i prioritet), podataka (do 8B), CRC-a, ACK bitova
- Svi čvorovi mogu da šalju podatke, ali ne istovremeno
- CAN specifikacija bitove naziva:
 - „dominantnim“ (logička „0“, aktivno napajanje od strane pošiljaoca)
 - „recesivnim“ (logička „1“, otpornik pasivno vraća napajanje)



CAN magistrala

- Neaktivno stanje predstavljeno je recesivnim nivoom (logičko „1“)
- Ako jedan čvor šalje dominantni bit, a drugi recesivni, nastaje kolizija i dominantni pobeđuje
- Obzirom da se prvo prenosi 11-bitni ID, čvor sa manjim brojem pobeđuje i bez zastoja nastavlja da šalje poruku, a čvor sa većim brojem se povlači (rešavanje prioriteta u realnom vremenu)
- CAN mreža može da bude konfigurisana da radi sa dva formata okvira:
 - Standardni (osnovni) – 11-bit ID
 - Prošireni – 29-bit ID (11-bit osnovni + 18-bit proširenje)
- Format se specificira IDE bitom, koje je dominantan (0) u slučaju osnovnog formata, tj. recesivan (1) u slučaju proširenog formata

	Start Bit	ID Bits											The Rest of the Frame
		10	9	8	7	6	5	4	3	2	1	0	
Node 15	0	0	0	0	0	0	0	0	1	1	1	1	
Node 16	0	0	0	0	0	0	0	1	Stopped Transmitting				
CAN Data	0	0	0	0	0	0	0	0	1	1	1	1	

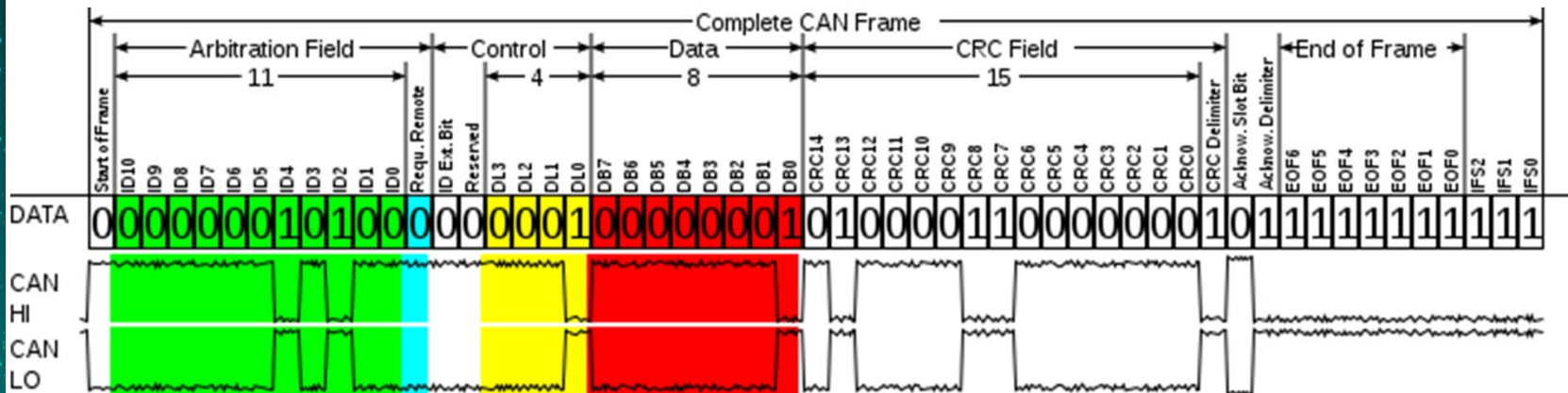


CAN magistrala

- CAN ima 4 tipa okvira:
 - **Data** – sadrži podatke
 - **Remote** – zahtev za slanjem od odgovarajućeg čvora
 - **Error** – šalje čvor koji je otkrio grešku
 - **Overload** – unosi kašnjenje između Data ili Remote okvira

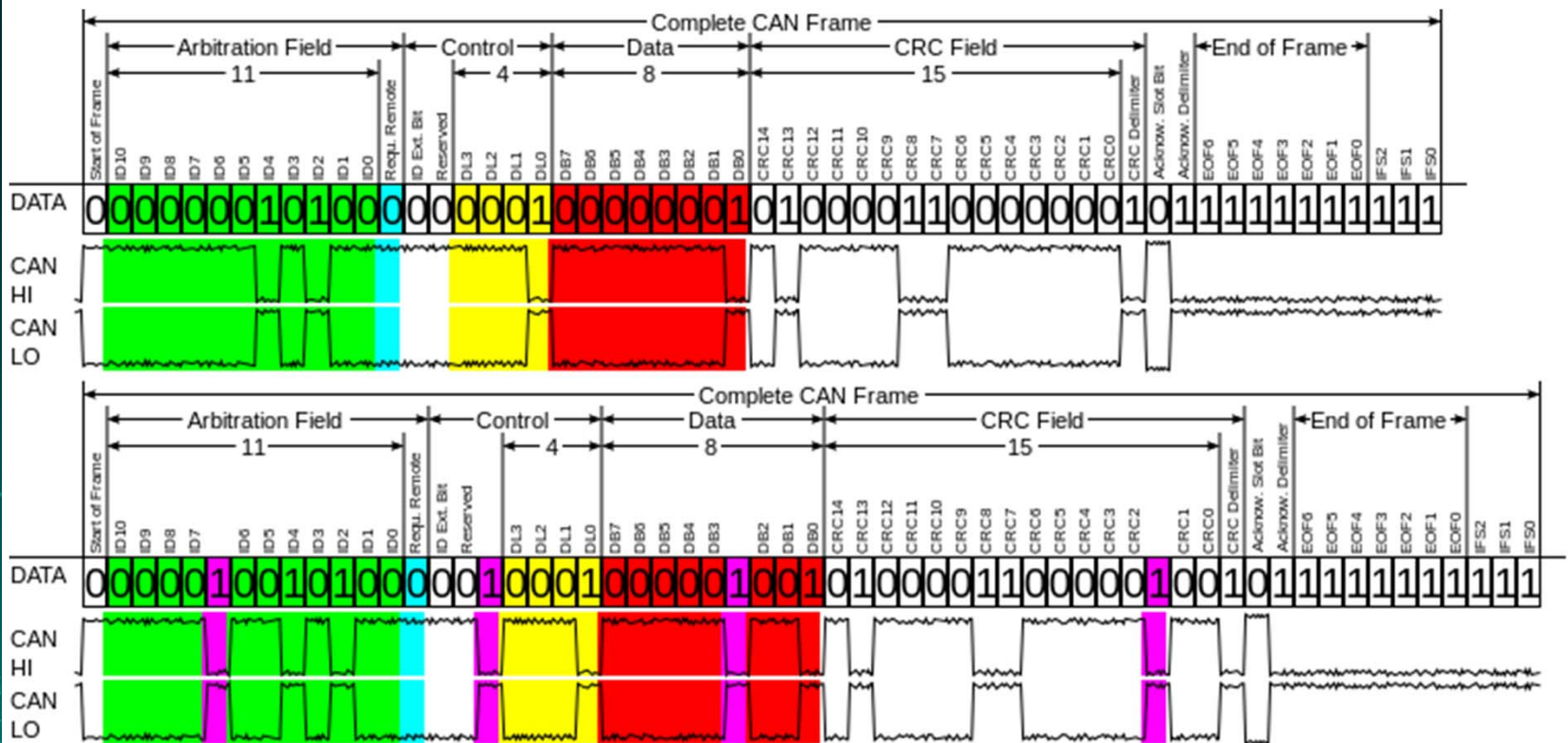
Data okvir (osnovni)

- SOF – (Start of Frame) početak okvira (1 bit)
- ID – identifikator/prioritet (11 bitova, zelena)
- RTR – (*Remote Transmission Request*) mora biti 0 za Data okvir (1 bit, plava)
- IDE – (*Identifier Extension*) mora biti 0 za osnovni format (1 bit)
- DLC – (*Data Length Code*) br. bajtova u Data delu (4 bita, žuta)
- Data – korisni podaci koji se šalju (0-64 bita, crvena)
- CRC – kontrolna suma (15 bitova)
- CRCD – (*CRC Delimiter*) mora biti 1 (1 bit)
- ACKS – (ACK Slot) pošiljalac postavlja 1, a primalac potvrđuje sa 0 (1 bit)
- ACKD – (ACK Delimiter) mora biti 1 (1 bit)
- EOF – (End of Frame) mora biti 1 (7 bitova)



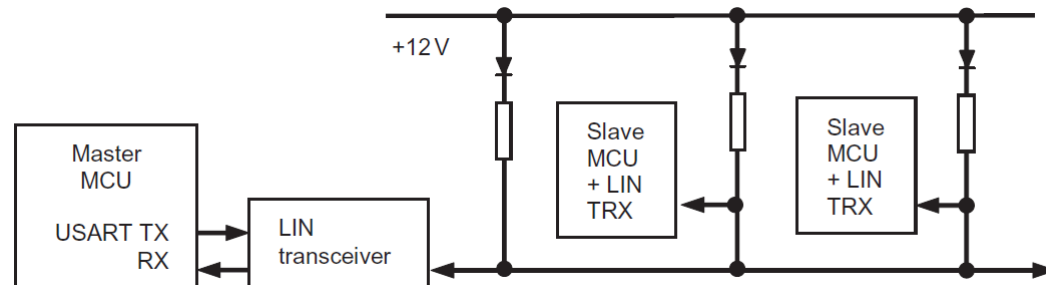
Umetanje bitova (*Bit Stuffing*)

- Da bi se održala sinhronizacija (jer se koristi NRZ kodiranje), bit suprotnog polariteta se umeće nakon 5 uzastopnih bitova istog polariteta



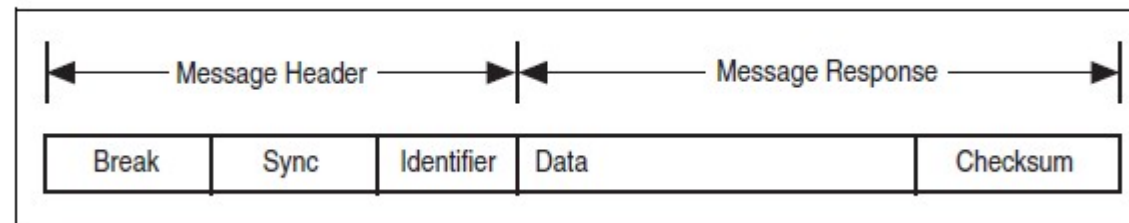
LIN magistrala

- **LIN (*Local Interconnection Network*)** je serijski mrežni protokol za komunikaciju komponenata unutar vozila
- Nastaje kao „jeftina“ varijanta mreže, jer je CAN skup za implementaciju na svakoj komponenti u vozilu
- Kasnih 90-tih formiran je LIN konzorcijum (BMW, Volkswagen Group, Audi Group, Volvo, Mercedes-Benz) koji 2002 objavljuje prvu verziju (tačnije v1.3) LIN specifikacije
- LIN je „broadcast serijska“ mreža koja obuhvata 16 čvorova (1 master i do 15/16 *slave* čvorova)
- Sve poruke inicira master, a najviše 1 *slave* odgovara (master može sam odgovoriti na svoju poruku, tada se ponaša kao *slave*)
- Pošto sve poruke inicira master, **nema kolizije**
- LIN se može javiti kao podsistem, povezan na *backbone* koga čini CAN



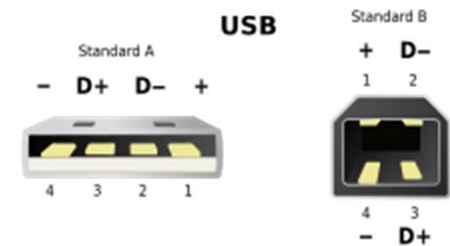
LIN format okvira

- Master postavlja na mrežu zaglavlje (Break, Sync i ID), a *slave* koji prepozna ID dopunjuje ostatak poruke (Data i Checksum)
- **Break** – svaki LIN okvir počinje sa 13 dominantnih i 1 recesivnim bitom
- **Sync** – 0x55, omogućuje *slave* uređajima da utvrde bitsku brzinu prenos i sinhronizuju se sa masterom
- **ID** – identifikator poruke (6+2 bita; 6 za ID, 2 msb za proveru parnosti ⇒ 64 različita ID)
 - 0-59 – Data okviri
 - 60-61 – dijagnostički okviri (uvek 8B u Data delu)
 - 62 – rezervisano za korisnička proširenja
 - 63 – rezervisano za buduća unapređenja protokola
- **Data** – 1-8B korisnih podataka koje prosleđuje *slave*
- **Checksum** – 8-bitna kontrolna suma (suma svih bajtova i oduzimanje 255, svaki put kada se dostigne ili premaši 256; sabiraju se samo Data bajtovi (klasični algoritam) ili i ID („unapređeni“ algoritam))
- Za svaki ID, u planskoj tabeli, čuva se razmak između okvira



Universal Serial Bus (USB)

- Industrijski standard za komunikaciju između računara, periferija i drugih računara
- 4 provodnika: D+ i D- (upredena parica za prenos signala), napajanje (+5V, za napajanje uređaja koji zahtevaju manje od 500mA) i uzemljenje
- 2 tipa standardnih konektora: A (ravni) i B (četvrtasti) i 3 tipa mini i mikro konektora: A, B i AB
- Brzine od 1.5Mb/s (v1.0 low speed), preko 12Mb/s (v1.0 full speed), 480Mb/s (v2.0), 5Gb/s (v3.0), 10Gb/s (v3.1), 20Gb/s (v3.2) do 40Gb/s (v4.0)
- Za detalje pogledati **vežbe**

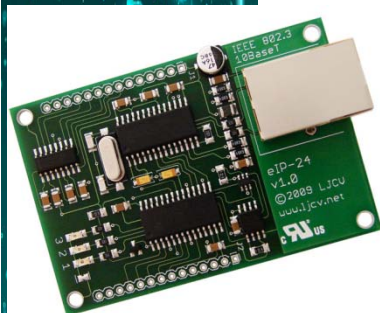




Ethernet



- Najčešća implementacija lokalne mreže
- Brzine od 10Mb/s, preko 100Mb/s i 1GB/s, do 100Gb/s
- Za format poruka i protokol za komunikaciju pogledati **vežbe iz Računarskih mreža**
- Microchip nudi podršku za 10Mb/s Ethernet kroz:
 - nezavisne kontrolere sa SPI interfejsom (npr. ENC28J60)
 - module integrisane u PIC MCU (npr. PIC18F97J60)



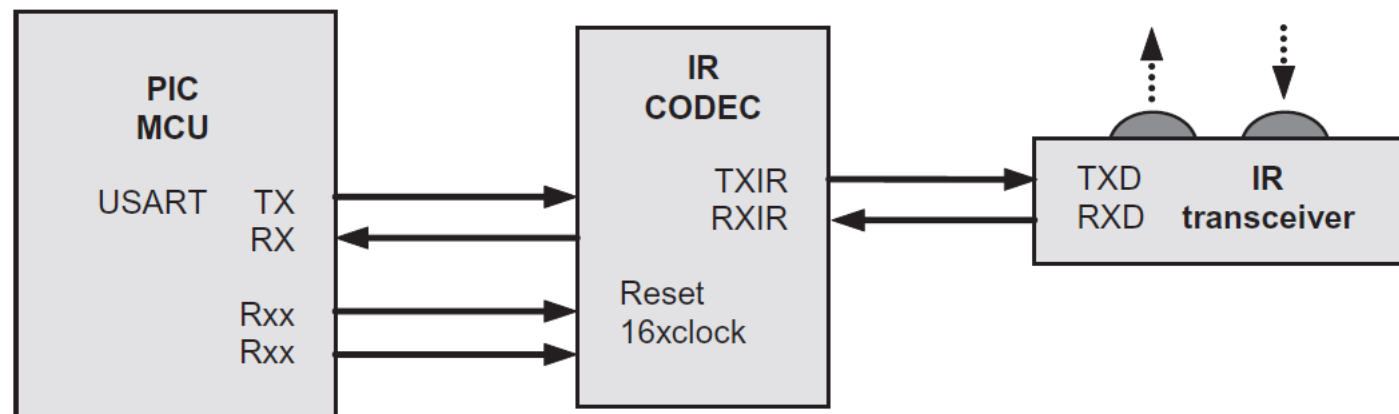


Bežiční linkovi

- Infrared
- Radio
- Wi-Fi
- Zigbee

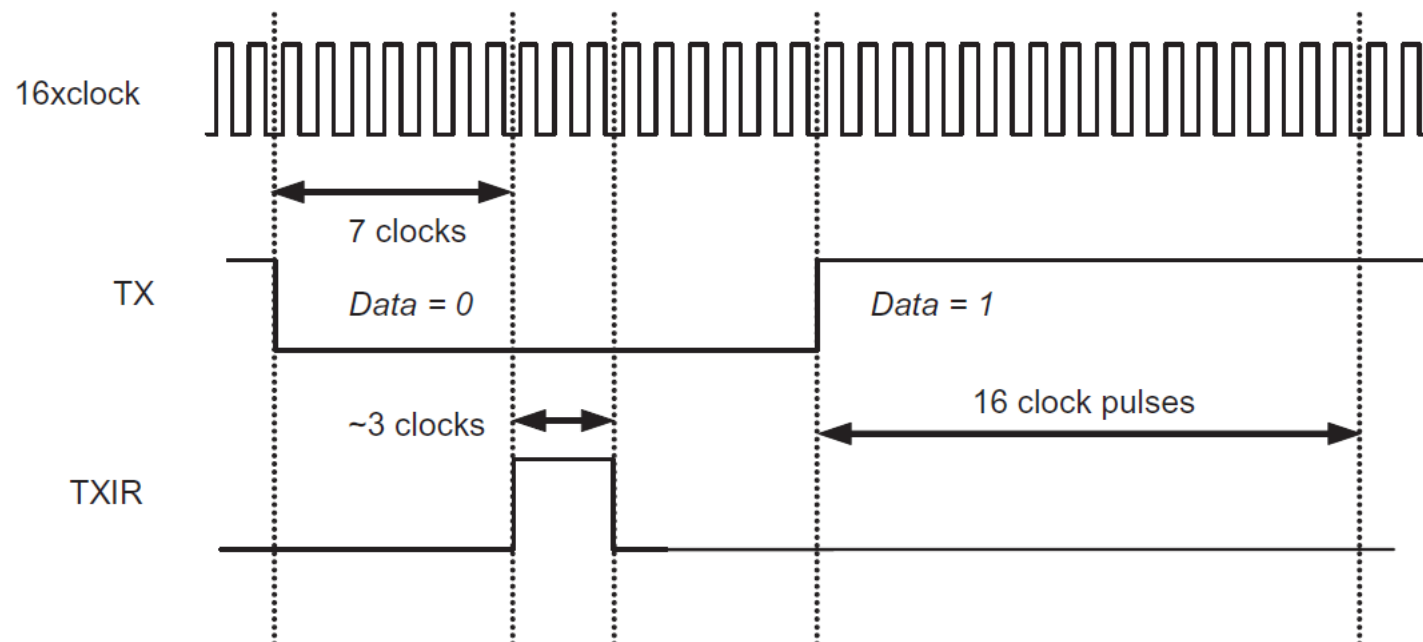
Infrared (IR)

- Već dugo se koristi za upravljanje TV uređajima i drugim kućnim aparatima
- Frekvencija svetlosnih impulsa je neznatno ispod vidljivog dela spektra, tako da ambijentalna svetlost ne utiče na prijemnik
- Potrebna je optička vidljivost (moguća je refleksija od okolnih površina)
- Potreban je posrednik (MCP2122) za komunikaciju PIC MCU (preko USART-a) i IR transivera TFDU 4100
- TFDU 4100 ima IR LED i PIN fotodiodu



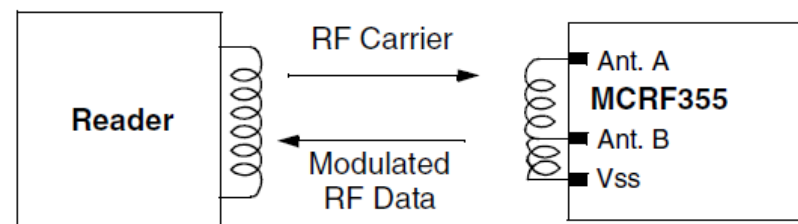
Infrared (IR)

- Potreban je 16x viši takt od takta kojim USART šalje bitove
- Kada se šalje 0, javlja se impuls koji uključuje IR LED (1 ne stvara impuls)
- Impuls je širini ~3 taktne intervala i javlja se na 7 taktova od ivice podatka
- Podrazumeva se 1 start i 1 stop bit (resinhronizacija na 10 bitova)



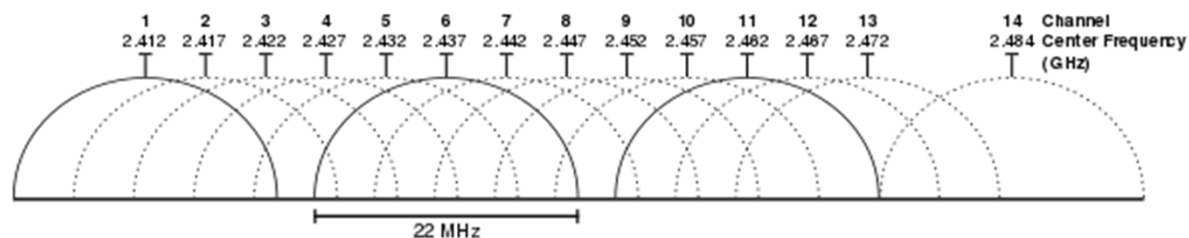
Radio-veza

- Upravljanje preko radio-veze (modelima aviona, brodova i automobila), bila je jedna od prvih primena ovog vida komunikacije
- Identifikacija radio frekvencijom (RFID) se koristi za promet robe, identifikaciju životinja, sigurnosne sisteme i sl. (pasivni identifikator sadrži MCU i predajnik, koga preko magnetne zavojnice napaja čitač ID-a; primer pasivnog RFID uređaja je Microchip MCRF355)
- Daljinski radio-upravljeni prekidači koriste se za automatske kapije, motorna vozila i sl.



Wi-Fi

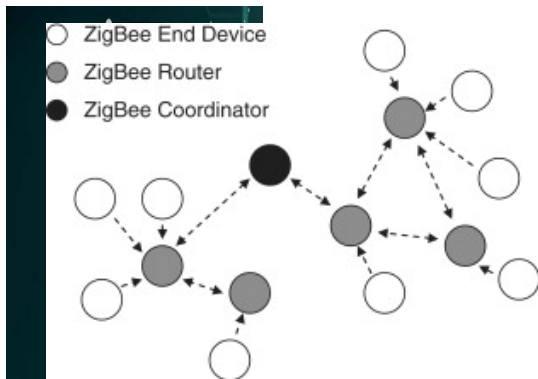
- Najpoznatiji tip bežične komunikacije je „bežična lokalna mreža“ (WLAN)
- Radi na 2.4GHz, sa kanalima razdvojenim po 5MHz
- Domet ~50m
- Microchip Wi-Fi transiver malog dometa - MRF24WB0Mx (vezuje se preki SPI porta) i omogućuje prenos brzinom 2Mb/s
- Zahteva sa PIC18 ili jači MCU, jer je TCP/IP stek potreban za komunikaciju prilično zahtevan



Zigbee



- Mrežni sistem namenjen nadgledanju i upravljanju
- Podržava topologiju zvezde, stabla i meša
- Čine ga mali uređaji, male potrošnje (obično napajani baterijom) koji komuniciraju brzinom 20-900 kb/s, na frekvenciji 868MHz (Evropa), 915MHz (SAD) ili 2.4GHz
- Tri uloge uređaja: koordinator, ruter i krajnji uređaj
- Svaka mreža mora da ima jednog koordinatora (u topologiji zvezde, koordinator je centralni čvor)
- Zigbee ruteri proširuju komunikaciju na mrežnom nivou i omogućuju topologije stabla i meša
- Microchip MRF24J40XX je 2.4GHz Zigbee RF transiver koji se povezuje preko SPI porta (SDI, SDO, CS i SCK) na MCU i omogućuje prenos podataka brzinom 250 ili 625 kb/s



Poređenje serijskih linkova

Link	MCU/ Port	Configuration	Protocol/Type	Signal	Data	Typical Speed (bits/s)	Interface	Error Check	Range	Typical Applications
RS-232	All PICs USART	Synchronous Asynchronous	RS-232	TTL or diff. voltage drive	1 byte	9600–115,200	Voltage driver $\pm 12V$	Parity	100m	Simple slow serial port, wireless transceiver
RS-422	USART Asynch	Up to 10 receivers	RS-232 based	Differential transceivers	1 byte	10M max	Voltage driver $\pm 6V$	Parity	1km	PLC programmer, small controller network
RS-485	USART Asynch	Up to 256 Transceivers	RS-232 based	Differential transceivers	1 byte	10M max	Voltage driver $+12V/-6V$	Parity	1km	Industrial control systems
SPI	MSSP	Hardware slave select	Master/slave	TTL on board	1 byte	10M max	2-wire 5V supply	None	1m	Link MCUs and peripherals Interface with RF transceivers
I ² C	MSSP	Software slave address	Master/slave	TTL on board	1 byte frame	1M max	2-wire 5V Supply	None.	1m	Memory expansion Sensor interface
CAN	18FXX upwards	ECU network	Peer to peer addressing	Differential transceivers	8 byte frame	1M max	2-wire 12V supply	CRC	10m	High-performance motor vehicle control system
LIN	All PICs RS-232	Master + 16 slaves	Broadcast system	Wired OR transceivers	8 byte frame	10k max	1-wire with 12V pull-up	Checksum	10m	Low-cost motor vehicle control network
Infrared	USART	Master/slave	Point to point	IR transceiver	1 byte	9600–115,200	IR diode & photo-detector	Parity, CRC	10m	TV remote control
Radio Control	PWM	Hand-held 2/4 channel RF	Pulse width modulation	1–2ms	N/A	50 pulses per sec	2.4GHz band carrier	N/A	1km	Model craft remote control

Poređenje serijskih linkova

Link	MCU/ Port	Configuration	Protocol/Type	Signal	Data	Typical Speed (bits/s)	Interface	Error Check	Range	Typical Applications
RFID	RS-232	RF transceiver + RFID tag	Simple data frame	Simple pulse modulation	104-bit ID code	70k max	13.56MHz carrier	Checksum	0.1m	Product, customer, etc. Identification tag
Keeloq	Host I ² C	Remote tag transmitter	Secure Encryption	Simple PCM	32 bits	N/A	433MHz Carrier	Block cypher	100m	Motor vehicle remote locking
USB	Selected PICs	Master + 127 slaves	1 to 1 or star network	Differential transceivers	1023	V1 12M V2 480M	2-wire with 5V supply	CRC	10m	Temporary connection to peripherals, memory, etc.
Zigbee	SPI, 24F upwards	WPAN	Lower level networking	Wireless transceivers	Not specified	250k or 625k	2.4GHz transceiver	CRC	100m	Small wireless network for sensor control and data logging
Ethernet	Selected PICs	Server + LAN	Distributed network	Differential transceivers	1500	10M–1 G	2-pair	CRC	1km	Internet monitoring and control
Wi-Fi	SPI, 18F upwards	RF transceiver network + server	Full TCP/IP	Multichannel FDM	1500	2 M max	2.4GHz transceiver	CRC	100m	Internet monitoring and control