

STŠ: “Nikola Tesla”
Sremska Mitrovica

Maturski rad:
Školsko zvono na serijskom portu

Dipl. Inž:
Đoko Krsmanović

Ucenik:
Predrag Miletić

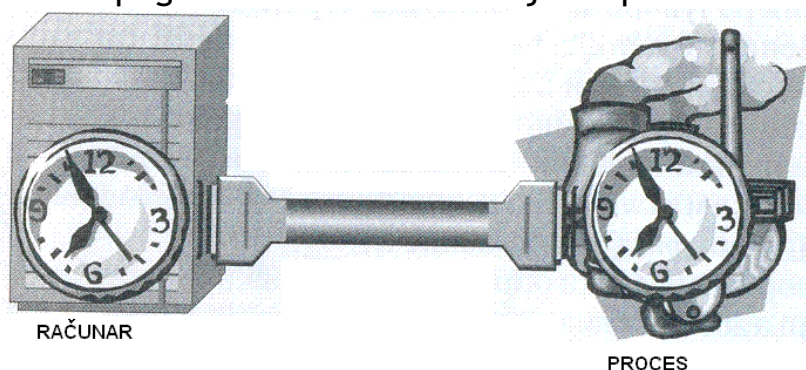
1.UOPSTENO O UPRAVLJANJU PROCESIMA POMOCU RACUNARA

Gledano potpuno uopsteno u svakom procesu postoji neki ulaz(informacija, materijal, signal...) koji se menja menja unutar njega, i napusta ga u izmenjenoj formu(izlaz procesa). Posto nijedan proces nije savrsen uvek postoji neka korekcija ili izmenu u cilju usavrsavanje tog procesa. To se moze postici smanjenjem vremena, energije i sl... da bi se ostvario zeljeni rezultat projektuje se upravljacki sistem koji za cilj ima da menja procesne promenljive u cilju poboljsavanje performansi procesa. Tako formiran sistem se zove sistem automatskog upravljanja(SAU).

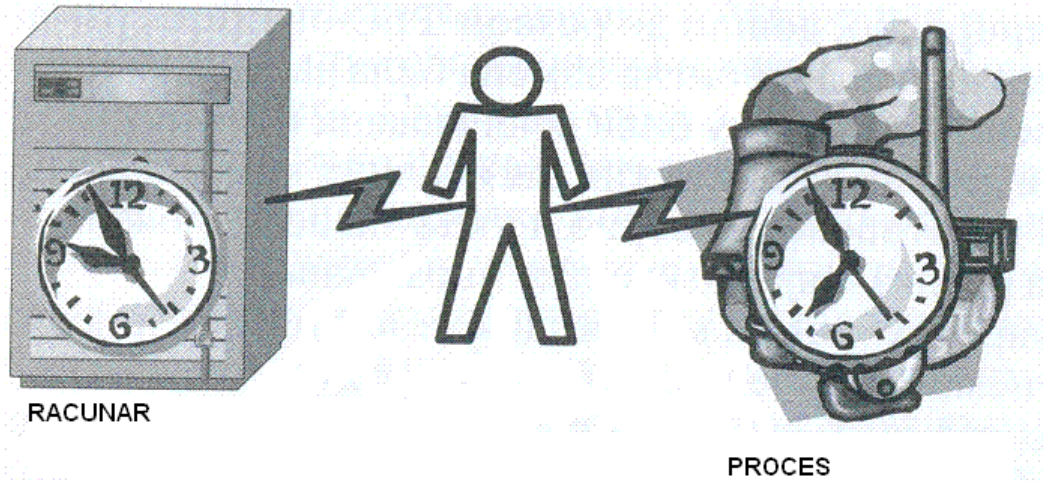
Zbog potreba upravljanja takvim sistemom javlja se ideja da se za upravljanje koristi racunar. Upravljanje sistemom obuhvata merenje, odredjivanje upravljanja i izvrsavanja komandi.

Postoje dva nacina vezivanja procesa i recunara.

Prvi je direktno gde je racunar direktno vezan za proces tj racunar upravlja sistemom u realnom vremenu. U tom slucaju racunar mora da prihvata merene velicine sa procesa, koje se prethodno moraju pretvoriti u digitalni signal, koji je prihvatljiv racunaru. Zatim racur mora da obradi merene velicine i na osnovu njih da da izlaz, koji se iz digitalnog oblika mora pretvoriti u pogodan oblik za delovanje na proces



Druga mogućnost korišćenja računara u upravljanju procesima je da se računar koristi samo za izračunavanje. A rezultati merenja dobijenih na procesu se ručno unose u računar, zatim ih računar obradjuje na matematičkom modelu, pa se prema dobijenim rezultatima ručno koriguje proces.



Real time upravljanje u odnosu na offline je mnogo brže, ali ono zahteva da računar stalno bude vezan na proces, i da je u realnom vremenu koriguje svaku nepravilnost u procesu.

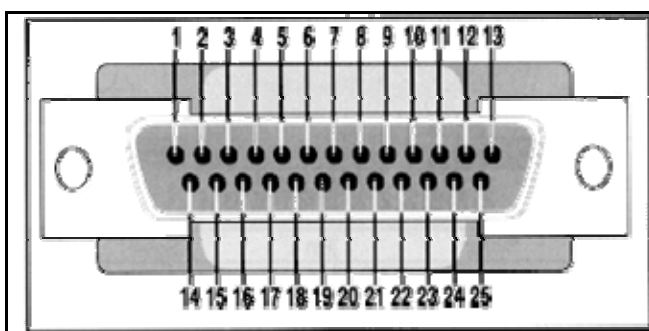
Ovakav sistem može da reaguje u određenim vremenskim intervalima. Tj računar posle određenog vremenskog intervala proverava stanje na procesu, i koriguje ga ako ima potrebe. Drugi način je da računar reaguje na događaj iz procesa. Npr ako je postignuta odgovarajuća temperatura da se isključi grejač, ili ako je postignut odgovarajući nivo da se zatvori ventil i sl.

2.UOPSTENO O SERIJSKOM PORTU

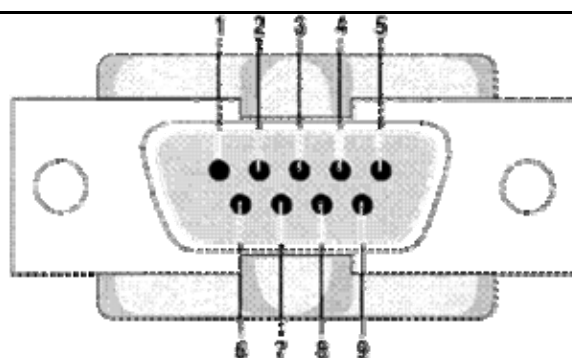
Serijski port PC računara koristi se za serijski prenos podataka (bit po bit) i samim tim je sporiji od paralelnog porta. Uređaji koji koriste serijsku komunikaciju koriste dve vrste kablova i to: DCE(Data Communications Equipment) i DTE(Data Terminal Equipment). DCE koriste modemi, ploteri i sl. dok se DTE koristi za vezu između PC računara. Električne osobine serijskog porta su sadržane u EIA(Electronics Industry Association) RS232C standardu čiji su parametri sledeći:

1. Logička nula "SPACE" nalazi se u opsegu napona +3 do +25 Volti
2. Logička jedinica "MARK" je u opsegu -3 do - 25 Volti
3. Oblast između -3 do + 3 Volta nije definisana
4. Napon na kolu ne sme da pređe 25 Volti u odnosu na masu(GND)
5. Struja kola ne sme preći 500 mA

Pored RS232C standarda postoje i novije verzije EIA-232D i EIA-232E koje su objavljene 1987 i 1991 godine. Serijski port koristi dve vrste konektora i to D-SUB 25 pinski i D-SUB 9 pinski konektor. Oblik konektora i raspored pinova na njima kao i glavni signali na pojedinim pinovima dati su na donjoj slici i tabeli.



DSUB-25



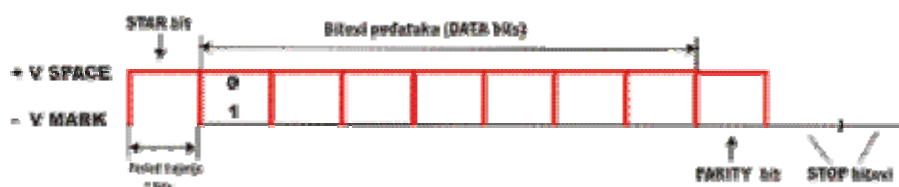
DSUB-9

DSUB-25	DSUB-9	Signal	Opis signala
pin 2	pin 3	TD	Transmit Data
pin 3	pin 2	RD	Receive Data
pin 4	pin 7	RTS	Request To Send
pin 5	pin 8	CTS	Clear To Send
pin 6	pin 6	DSR	Data Set Ready
pin 7	pin 5	SG	Signal Ground
pin 8	pin 1	CD	Carrier Detect
pin 20	pin 4	DTR	Data Terminal Ready
pin 22	pin 9	RI	Ring Indikator

Značenje pojedinih signala je sledeće:

Signal	Opis signala	Značenje signala
TD	Transmit Data	Serijski izlaz podataka(TXD)
RD	Receive Data	Serijski ulaz podataka(RXD)
RTS	Request To Send	Indicira da je modem spreman za razmenu podataka
CTS	Clear To Send	Kada modem detektuje signal "Carrier" od modema sa druge strane linije, onda ona postaje aktivna
DSR	Data Set Ready	DCE signalizira da je spreman za rad
SG	Signal Ground	Masa
CD	Carrier Detect	DCE javlja da je veza uspostavljena
DTR	Data Terminal Ready	Indicira DCE uređaju da je DTE spreman
RI	Ring Indikator	Signalizira da je detektovan signal "zvona" na telefonskoj liniji

Oblik signala RS232 standarda dat je na donjoj slici:



Kada nema slanja podatka tj. linija je slobodna označeno je MARK naponom. Prenos počinje kada linija pređe u SPACE stanje što predstavlja START bit. Iza start bita dolaze bitovi podataka i to tako da je jedinica predstavljena negativnim naponom a nula pozitivnim. Ovo je najčešće u suprotnosti sa ustaljenom logikom da je +5 V logička jedinica, a 0 V logička nula. Problem rešava prijemno kolo konvertujući napone tako da su prilagođeni naponima interfejsa koji se koristi. Posle bita podatka dolazi bit pariteta koji se koristi za detekciju greške u prenosu. Iza bita pariteta signal se postavlja u MARK stanje što označava početak STOP bita. Prema RS232 standardu broj stop bitova može biti: 1, 1.5 ili 2. Nakon završetka STOP bitova linija je spremna za slanje novog karaktera. PC računari koriste kao DATA bite najčešće 7 ili 8 bita. Između slanja dva karaktera linija se nalazi u MARK stanju, a u toku prenosa više puta prelazi iz MARK u SPACE stanje u zavisnosti od broja jedinica i nula. To znači da se linija može nalaziti u SPACE stanju najviše u slučaju kada karakter sadrži sve nule. Ova osobina prenosa upotrebljena je za uvođenje specijalnog znaka nazvanog BREAK(prekid). Ovaj se signal koristi kako bi prijemniku data signalizacija da je došlo do problema u slanju podataka.

Serijski prenos počinje tako da predajna strana šalje prijemnoj strani signal RTS - zahtev za slanjem, dok prijemnik ako je spreman za prijem odgovara signalom CTS - spreman za prijem. Predajnik zatim šalje podatke prijemniku. Nakon prijema podataka prijemnik proverava da li su podaci primljeni bez greške i za to vreme javlja predajniku da je zauzet. Ako su podaci primljeni bez greške, prijemnik šalje predajniku signal potvrde ACK(Acknowledgment - ASCII 6), a u slučaju ako se pojavila neka greška, šalje signal negativne potvrde NAK(ASCII 21). Zavisno od upotrebljenog protokola, odgovor predajnika na dobijeni NAK signal može biti ponovno slanje podataka.

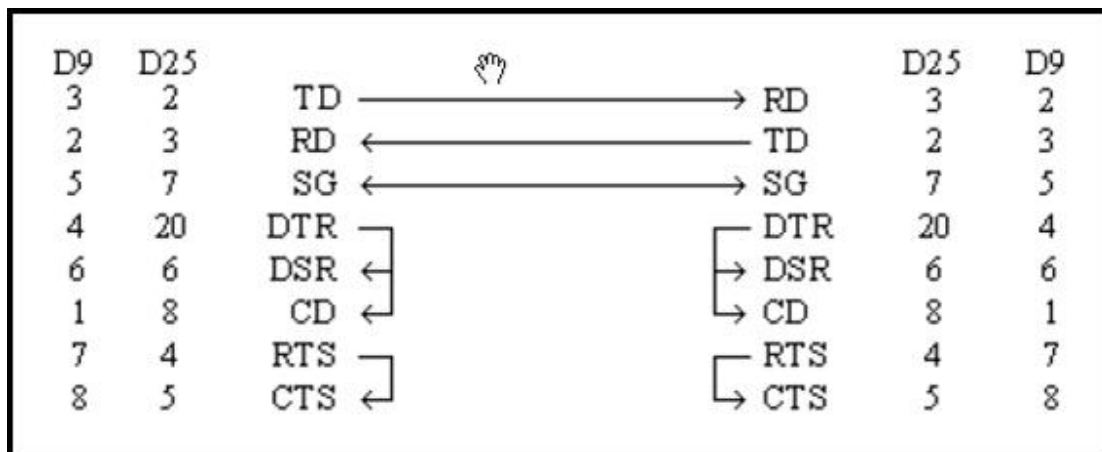
Protokol za prenos može se izvesti na dva načina i to: hardverski i softverski. Za realizaciju hardverskog protokola neophodno je da postoje linije: RTS, CTS i linija za prenos bitova poruke. Softverska realizacija koristi umesto linija RTS i CTS, ASCII znakove XON i XOFF (nazivaju su i DC1 i DC3), zbog čega je umesto tri dovoljna samo jedna linija. Kada se primenjuje ovaj protokol prijemnici i na predaji i prijemu proveravaju svaki prispeli znak radi

utvrđivanja da li je to XON, XOFF ili podatak. Strana koja ne može da primi karakter kao indikator tog stanja drugoj strani šalje XOFF znak.

Prilikom serijskog prenosa podataka moguća je pojava sledećih vrsta grešaka:

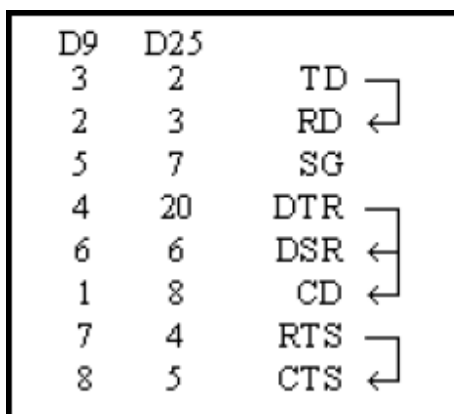
- Greška uokvirenja(framing error). Do ove greške dolazi u slučaju kada prijemnik očekuje STOP bit ali se on ne pojavljuje.
- Greška prekoračenja(overrun error). Nastaje kada se pojavljuje novi znak u prijemniku a prethodni nije upotpunosti primljen.
- Greška pariteta(parity error). Nastaje kada bit pariteta ne odgovara stanju bitova u bitovima podatka.

Standardno za vezu dva PC računara koristi se takozvani **Null Modem** kabal čiji je raspored pinova dat na donjoj slici:



Iz slike se vidi da su i na prijemnoj i predajnoj strani spojeni signali DTR, DSR i CD kao i RTS i CTS, dok su signali TD i RD ukršteni.

Za testiranje serijskog porta i izradu programa koristi se kabal sa zatvoreno petljom (LoopBack) prikazan na donjoj slici:



Serijska komunikacija se kod PC računara izvodi upotrebom Intelovog UART kola 8250 kod XT ili 16450 kod AT računara. Oba su kola slična s tim što kolo 16450 ima neka poboljšanja. Kolo 8250 ima 10 dok kolo 16450 ima 12 programibilnih jednobajtnih registara. Svaki port (COM1 i COM2) ima po jedno UART kolo koja se vezuju na linije prekida IRQ3 i IRQ4. Standardno se koriste dva porta a moguće je i više sa adresama i IRQ prem donjoj tabeli:

Naziv	Adresa	IRQ
COM 1	3F8	4
COM 2	2F8	3
COM 3	3E8	4
COM 4	2E8	3

Bazne adrese COM portova nalaze u BIOS-u na adresama datim u donjoj tabeli:

Startna adresa	Opis
0000:0400	Početna adresa COM1
0000:0402	Početna adresa COM2
0000:0404	Početna adresa COM3
0000:0406	Početna adresa COM4

Serijski port upotrebljava u svom radu sledeću grupu registara:

Bazna Adresa	DLAB	Čitanje/Pisanje	Skraćeni naziv	Naziv registra
+ 0	=0	Pisanje	THR	Transmitter Holding Buffer
	=0	Čitanje	RDR	Receiver Buffer
	=1	Čitanje/Pisanje	DLL	Divisor Latch Niži Bajt
+1	=0	Čitanje/Pisanje	IER	Interrupt Enable Register
	=1	Čitanje/Pisanje	DLH	Divisor Latch Viši Bajt
+2	-	Čitanje	IIR	Interrupt Identification Register
	-	Pisanje	FCR	FIFO Control Register
+3	-	Čitanje/Pisanje	LCR	Line Control Register
+4	-	Čitanje/Pisanje	MCR	Modem Control Register
+5	-	Čitanje	LSR	Line Status Register
+6	-	Čitanje	MSR	Modem Status Register
+7	-	Čitanje/Pisanje	-	Scratch Register

Transmitter Holding(THR) i Receiver Buffer(RDR) registar su registri podatka. Prvom se pristupa prilikom slanja karaktera na port, a drugom u trenutku čitanja karaktera sa porta. Kako se ove dve operacije međusobno isključuju to je nemoguće da dođe do kolizije. Registar THR sadrži bajt koji treba poslati na liniju. Podatak se može upisati kada peti bit Line Status Registra naznači da je on prazan. RDR registar sadrži poslednji primljeni bajt sa

linije. Sadržaj ovog registra se može pročitati kada je nulti bit Line Status Registra označen da je primljen. Na offsetu 0 nalazi se i registar Divisor Latch Niži Bajt u slučaju kada je **DLAB=1**. Ovaj se registar kao i sledeći Divisor Latch Viši Bajt koriste se samo u procesu inicijalizacije UART-a. Proces inicijalizacije se označava setovanjem sedmog bita u Line Control Registru(**DLAB**), tako da ni u ovom slučaju nemože doći do dvostrukog tumačenja.

Za različite brzine prenosa izražene brojem bita u sekundi(bps) moramo znati trajanje signala koji reprezentuje jedan bit. Ta se konstanta ne definiše direktno već preko pomoćne vrednosti koja se dobije iz formula:

$$\text{Divisor Latch} = (\text{clock}) / (16 \times \text{bps}) \text{ ili}$$

$$\text{Divisor Latch} = (1843200 / (16 \times \text{bps})) \text{ ili}$$

$$\text{Divisor Latch} = (115200) / (\text{bps})$$

gde clock predstavlja frekvenciju internog takta UARTa koja je 1.8432 Mhz, a bps je željena brzina prenosa. Divisor Latch je broj kojim se deli takt UART kola da se dobije dužina trajanja jednog bita pri željenoj brzini prenosa.

Taj se broj raspoređuje na niži(DLL) i viši(DLH) bajt što je dato u donjoj tabeli.

Brzina (bps)	Delilac	DLH	DLL
50	2304	09h	00h
300	384	01h	80h
600	192	00h	C0h
2400	48	00h	30h
4800	24	00h	18h
9600	12	00h	0Ch
19200	6	00h	06h
38400	3	00h	03h
57600	2	00h	02h
115200	1	00h	01h

Struktura Interrupt Enable Registra(IER) data je u donjoj tabeli:

Bit	Opis
Bit 7	Ne koristi se
Bit 6	Ne koristi se
Bit 5	Omogućava Low Power Mod(16750)
Bit 4	Omogućava Sleep Mod(16750)
Bit 3	Omogućava Modem Status Interapt
Bit 2	Omogućava Receiver Line Status Interapt
Bit 1	Omogućava Transmitter Holding Registrar Interapt kada je prazan
Bit 0	Omogućava Received Data Interapt

Ovim registrom se određuje na koji će tip prekida UART kolo reagovati. Može se dozvoliti pojava i više tipova prekida. Prekid se generiše kada se ispuni neki od uslova iz date tabele. Tada se postavalja odgovarajući bit iz prve kolone kao indikator pojave prekida. Nakon obrade prekida postavljeni bit se resetuje i UART je spreman da prihvati sledeći prekid.

Interrupt Identification Register(IIR) ima strukturu kao u donjoj tabeli:

Bit	Opis		
Bitovi 6 i 7	Bit 6	Bit 7	
	0	0	Ne koristi se FIFO
	0	1	FIFO je omogućen ali se ne može koristiti
	1	1	FIFO je omogućen
Bit 5	64 Bajtni FIFO je omogućen(samo za 16750)		
Bit 4	Rezervisan		
Bit 3	0	Rezervisan za 8250 i 16450	
	1	16550 Time-out Interapt je u toku	
Bitovi 1 i 2	Bit 2	Bit 1	
	0	0	Modem Status Interapt
	0	1	THR interapt registar kada je prazan
	1	0	Podatak je primljen
	1	1	Primljeni Line Status Interapt
Bit 0	0	Prekid je u toku	
	1	Ne postoje prekidi	

Na osnovu ovog registra komunikacioni programi određuju tip prekida koji se pojavio.

FIFO Control Registar(FCR) ima strukturu kao u donjoj tabeli:

Bit	Opis		
Bitovi 6 i 7	Bit 6	Bit 7	Nivo Interapta
	0	0	1 Bajt
	0	1	2 Bajta
	1	0	8 Bajta
	1	1	16 Bajta
Bit 5	64 Bajtni FIFO je omogućen(samo za 16750)		
Bit 4	Rezervisan		
Bit 3	Izbor DMA moda		
Bit 2	Briše FIFO slanja		
Bit 1	Briše FIFO prijema		
Bit 0	Omogućav FIFO metod		

FIFO je registar u koji se može samo upisivati. Koristi se za kontrolu FIFO bafera koji postoji na UART-ima 16550 i više. Postavljanjem bita 0 omogućuje predajni i prijemni FIFO a njegovim resetovanjem se zabranjuje. Bitovi 1 i 2 kontrolišu brisanje prijemnog i predajnog FIFO bafera. Setovanjem ovih bitova na 1 će samo obrisati sadržaj FIFO bafera i neće uticati na šifrt registar. Ova dva bita se sami resetuju i zato nemoramo bitove setovati na nulu po završetku. Bit 3 omogućava DMA selektovanje DMA moda koji se može naći na UART-ima 16550 i više. Bitovi 6 i 7 se koriste da se postavi nivo okidanja na primajući FIFO. Naprimer ako je bit 7

postavljen na 1 i bit 6 na 0 onda je nivo okidanja setovan na osam bajtova. Kada je primljeno 8 bajtova podataka u primajućem FIFO onda je Received Data Interapt postavljen u IIR registru.

Line Control Registrar(LCR) ima strukturu kao u donoj tabeli:

Bit	Opis			
Bit 7	1	DLAB bit		
	0	Omogućen pristup THR, RDR i IER registru		
Bit 6	0 - Zabranjen Break signal, 1 - Omogućen			
Bitovi 3 : 5	Bit 5	Bit 4	Bit 3	Izbor pariteta
	X	X	0	Bez pariteta
	0	0	1	Neparan(Odd) paritet
	0	1	1	Paran(Even) paritet
	1	0	1	MARK
	1	1	1	SPACE
Bit 2	Dužina stop bita			
	0	Jedan stop bit		
	1	2 stop bita ako je dužina podatka 6, 7 ili 8 bita, odnosno 1.5 stop bit za podatak dužine 5 bita		
Bitovi 0:1	Bit 1	Bit 0	Dužina podatka u bitima	
	0	0	5 bita	
	0	1	6 bita	
	1	0	7 bita	
	1	1	8 bita	

Ovaj registar omogućava postavljanje nekih od parametara veze kao i režim rada registara THR, RDR, DLL i DLH. Režim rada navedenih registara određuje njegov sedmi bit DLAB(Divisor Latch Access Byte).

Modem Control Registrar(MCR) koristi sledeću strukturu:

Bit	Opis
Bit 7	Ne koristi se
Bit 6	Ne koristi se
Bit 5	Autoflow kontrola omogućena(samo za 16750)
Bit 4	Bit povratne sprege
Bit 3	Korisnički izlaz #1
Bit 2	Korisnički izlaz #2
Bit 1	RTS linija aktivna
Bit 0	DTR linija aktivna

DTR linija signalizira modemu da je računar spreman da od njega primi karakter, dok linija RTS signalizira modemu da je računar spreman da mu pošalje podatak. Korisnički izlaz#1 normalno nije spojen ali se na nekim karticama koristi za prebacivanje između 1.8432 Mhz kristala i 4 Mhz kristala koje se koristi za MIDI. Korisnički izlaz#2 se koristi za spajanje vanjskih kola koja kontrolišu UART-CPU interept procese.

Line Status Registar(LSR) koristi sledeću strukturu:

Bit	Opis
Bit 7	Greška u prijemnom FIFO
Bit 6	Prazan Data Holding Registar
Bit 5	Prazan THR registar i može se slati sledeći znak
Bit 4	Detektovan signal prekida (Break)
Bit 3	Greška uokvirenja(Framing error)
Bit 2	Greška pariteta(Parity error)
Bit 1	Greška prekoračenja(Overrun error)
Bit 0	Podatak je primljen i bajt se nalazi u RDR

Ovaj registar ima prevashodno namenu da daje izveštaj o stanju na komunikacionoj liniji.

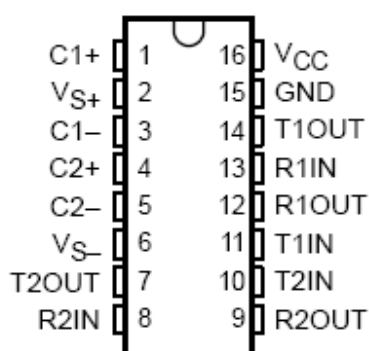
Modem Status Registar(MSR) ima sledeću strukturu:

Bit	Opis
Bit 7	Aktivira "Data Carrier Detect" (DCD)
Bit 6	Aktivira "Ring Indicator"(RI)
Bit 5	Aktivira "Data Set Ready"(DSR)
Bit 4	Aktivira "Clear To Send"(CTS)
Bit 3	Promena u "Data Carrier Detect"(DCD)
Bit 2	Promena u "Ring Indicator"(RI)
Bit 1	Promena u "Data Set Ready"(DSR)
Bit 0	Promena u "Clear To Send"(CTS)

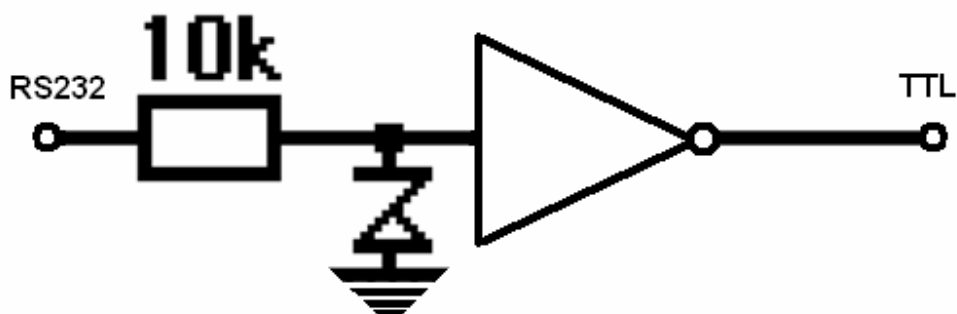
Status modema zavisi pre svega od toga da li je neka linija aktivna ili nije i da li se status pojedine linije promenio od poslednjeg čitanja ovog registra.

Prethodno smo videli da serijski port koristi signale u negativnoj logici tj. logička nula je pozitivna a logička jedinica negativna. Za prilagođenje tih signala TTL/CMOS kolima najčešće se koristi nekoliko načina:

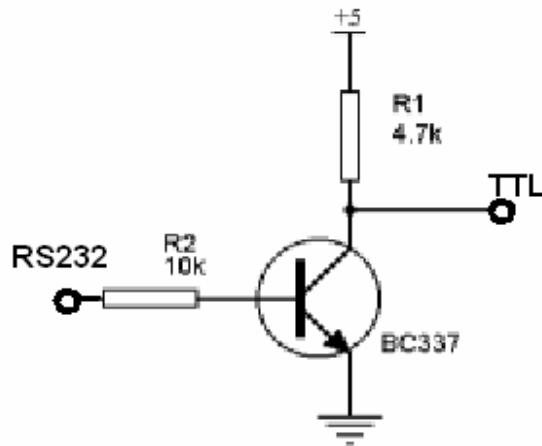
1. Upotreba MAX232 Upotrebom kola max 232 postoji mogućnost da se signali prevedu iz RS232 naponkih nivoa u TTL nivoe i obrnuto.



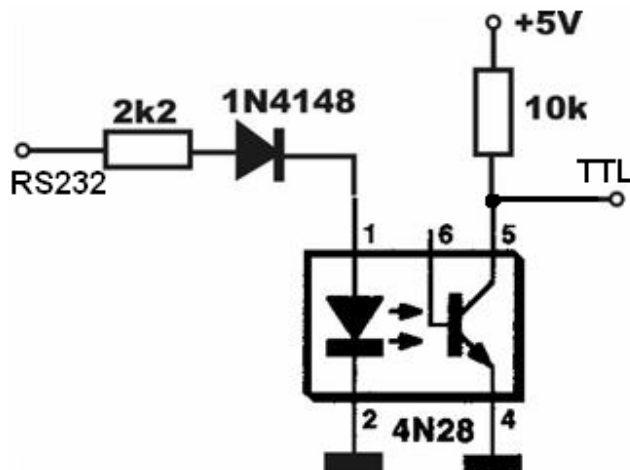
2. Upotreba invertora Upotrebom invertora sa predtoprom moguće je neki od izlaznih signala ovog porta prilagoditi TTL/CMOS kolima što je prikazano na slici



3. Upotreba tranzistora Takođe je moguća upotreba tranzistora za ovo prilagođenje što je prikazano na slici

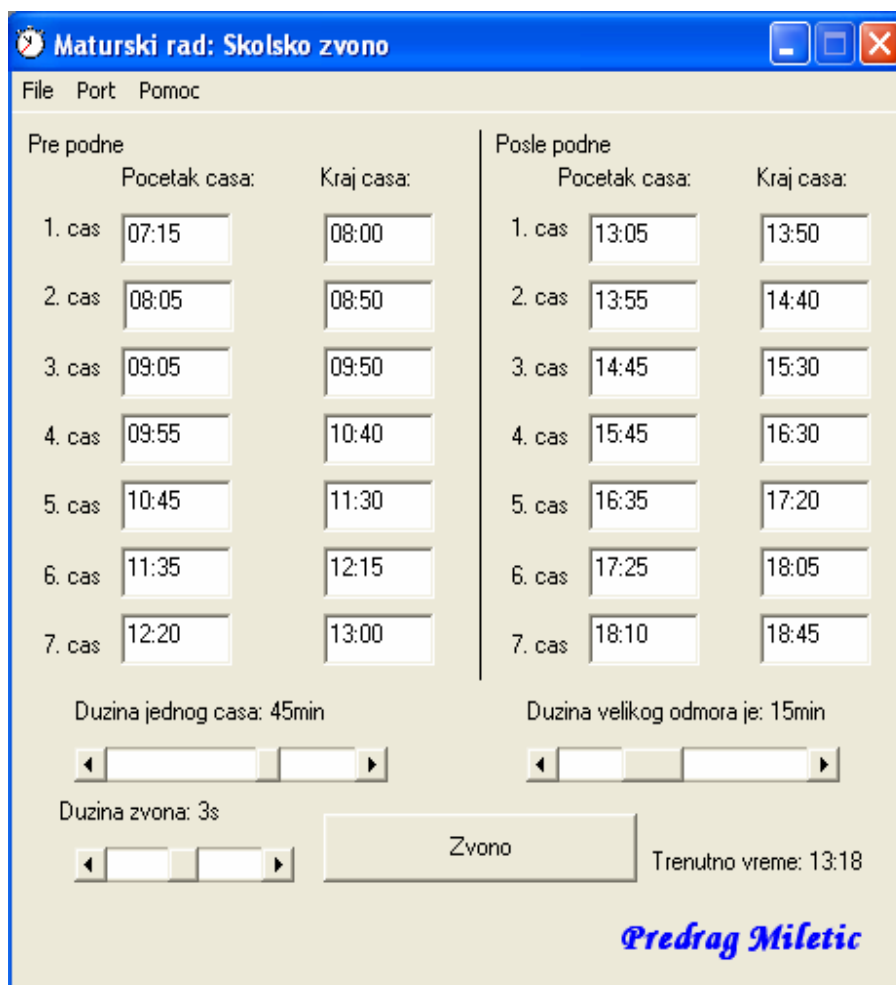


4. Upotreba optokaplera U galvanskom odvajanju izlaznih signala serijskog porta koriste se i optokapleri za prilagođenje TTL/CMOS kolima kao što je prikazano na slici



3.0 PROGRAMU ZA UPRAVLJANJE SKOLSKIM ZVONOM

Izgled programa za upravljanje skolskim zvonom je prikazan na slici

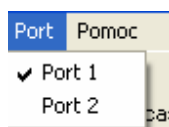


Na meni linije programa se nalaze tri kontrole, kao sto je prikazano na slici

File Port Pomoc

Pod kontrolom File se nalazi komanda Exit koja spada u jednu od standardnih opcija svakog programa.

Klikom na dugme Port dobijamo meni kao na slici



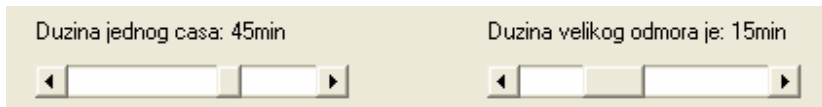
U ovom meniju se vrši izbor serijskog porta na koji je zakacen interfejs.

U drugom delu programa nalaze se text boxovi za unos vremena pocetka i kraja casa, kao sto se vidi na sledecoj slici

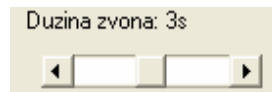
Pre podne			Posle podne		
	Pocetak casa:	Kraj casa:		Pocetak casa:	Kraj casa:
1. cas	07:15	08:00	1. cas	13:05	13:50
2. cas	08:05	08:50	2. cas	13:55	14:40
3. cas	09:05	09:50	3. cas	14:45	15:30
4. cas	09:55	10:40	4. cas	15:45	16:30
5. cas	10:45	11:30	5. cas	16:35	17:20
6. cas	11:35	12:15	6. cas	17:25	18:05
7. cas	12:20	13:00	7. cas	18:10	18:45

Unos vremena se moze vršiti rucno tako sto se u svako polje unosi vreme kada pocinje i kada se završava cas.

Druga mogucnost unosa vremena je da se samo upise pocetak prvog casa pre i posle podne, zatim se na slajderima sa sledece slike podesi trajanje cesa i velikog odmora, a zatim program sam popuni kompletnu tabelu sa sve casove



U programu postoji mogucnost da se menja duzina trajanja zvona ne slajderu prikazanom na slici

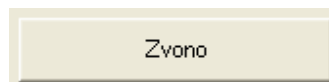


Najkraca duzina zvona je 1 sekund,dok je najdu 5 sekundi.

Program ima prikaz tacnog vremena kao sto se vidi na slici

Trenutno vreme: 13:18

U programu je ostavljena i mogucnost da se zvono moze ukljuciti u bilo kojem trenutku, sto se radi pritiskom na sledece dugme



4.0 KODU PROGRAMA

Pri učitavanju forme prvo se izvršava deo događaj Form_Load.

Prvo se izvršavaju pod programi za popunjavanje tabela pre i posle podne. Za svaku tabelu postoji po jedna pod program o kojima ce kasnije biti reci.

Sledeće je postavljanje vrednosti slajdera, kao što je slajder za dužinu časa on se postavlja na 45min, zatim slajder dužine velikog odmora koji se postavlja na 15min, i na kraju se postavlja dužina zvona na 3 sekunde.

Sledeći korak je odabir porta. Prvo se otvara COM1, a ako je on zauzet onda se prelazi na otvaranje COM2, a ako je i on zauzet onda se prijavljuje greška, da nema slobodnog porta. Za to je zadužen sledeći deo koda:

```
i = OPENCOM("COM1,1200,N,8,1")
Port1.Value = True
If i = 0 Then
i = OPENCOM("COM2,1200,N,8,1")
Port2.Value = True
End If
If i = 0 Then MsgBox ("Ne postoji slobodan COM port")
```

Zatim se očitava sistemsko vreme i vrsi se njegovo formatiranje da bi na svakom sistemu uvek bio isti prikaz. To izvršava pod program koji se naziva Vreme.

```

Private Sub vreme()
h = Hour(Time)
m = Minute(Time)
If h < 10 Then h = "0" & h
If m < 10 Then m = "0" & m
t = h & ":" & m
Label6.Caption = "Trenutno vreme: " & t
End Sub

```

Ovaj pod program se poziva svakog minuta kako bi program imao uvek tacnu informaciju o vremenu.

Sledeci deo programa je tajmer  koji služi da svakog minuta uporedi trenutno vreme i vremena u tabeli. Za to je zaduzen sledeci pod program:

```

Private Sub Timer1_Timer()
Call Vreme
If t = Text1.Text Or t = Text2.Text Or.....Or t =
Text27.Text Or t = Text28.Text Then Call Zvono
End Sub

```

Prvo se poziva pod program vreme kako bi se dobila informacija o vremenu koja je smestena u promenljivoj t. Zatim se svaka vrednost iz textbox-a iz tabele uporedjuje sa vrednoscu u promenljivoj t, i ako se ove dve vrednosti podudaraju, poziva se pod program Zvono.

Kod pod programa Zvono:

```

Private Sub Zvono()
TXD 1
Sleep (z)
TXD 0
End Sub

```

Prvo se postavlja TXD linija na 1, sto znaci da se zvono ukljucuje, zatim se pravi pauza naredbom sleep koja traje od 1-5s sto zavisi od polozenia slajdera. U promenljivoj z se nalazi numericka vrednost polozenia slajdera.

Podprogrami za ispunjavanje tabela za raspored zvonjenja

Kao što je već rečeno za svaku tabelu postoji po jedan podprogram, podprogrami su u sustini isti samo što se u prvom podprogramu vrliki odmor dodaje posle drugog časa, a u drugom podprogramu posle trećeg časa.

Deo koda samo za računanje kraja prvog časa i početka drugog:

```
m = m + c
If m > 59 Then
h = h + 1
m = m - 60
End If
If h < 10 And h > 0 Then h = "0" & h
If m < 10 Then m = "0" & m
Text22.Text = h & ":" & m
```

```
'2 cas pocetak
h = Val(Left(Text22.Text, 2))
m = Val(Right(Text22.Text, 2))
m = m + 5
If m > 59 Then
h = h + 1
m = m - 60
End If
If h < 10 And h > 0 Then h = "0" & h
If m < 10 Then m = "0" & m
Text16.Text = h & ":" & m
```

U promenljivoj m nalaze se minuti. U promenljivoj c nalazi se dužina časa. Prvo se minuti sabiraju sa vrednošću časa, i ako ta vrednost predje 59, onda se promenljiva h u kojoj su smesteni sati uvećava za 1, zatim se od minuta oduzima 60. i na taj način dobijamo kraj prvog časa. Da bi dobili vrednost za početak narednog časa na prethodnu vrednost minuta dodajemo 5 min. A vrednost minuta i sata isčitavamo iz textbox-a pomoću sledećeg koda

```
h = Val(Left(Text22.Text, 2))  
m = Val(Right(Text22.Text, 2))
```

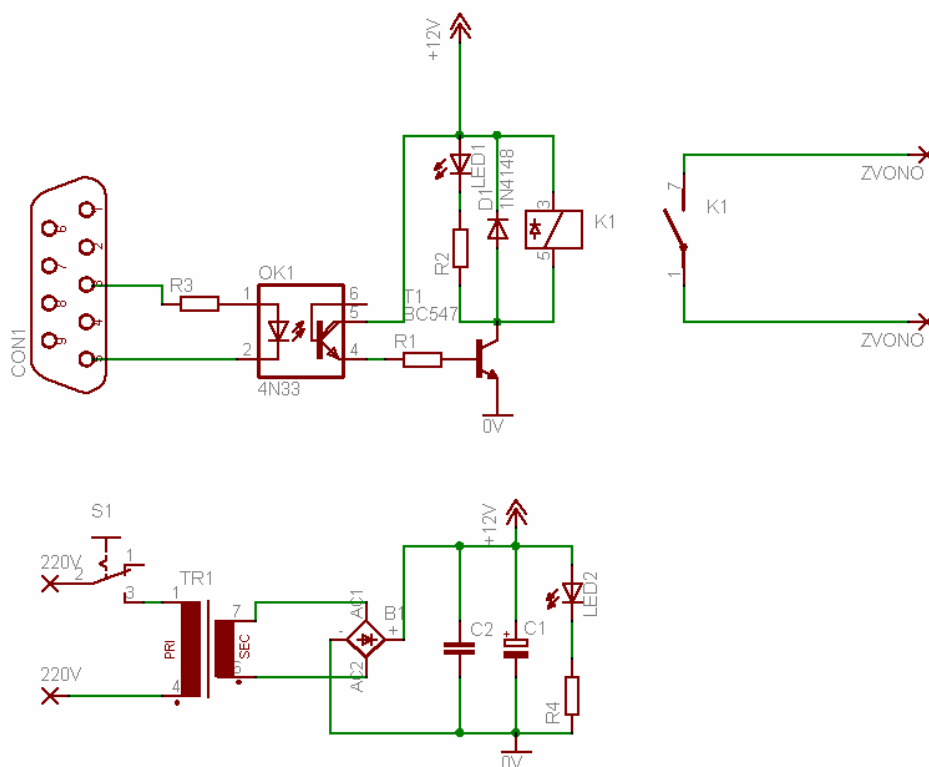
u prvom redu dobijamo sate, tako sto iscitavamo prva dva znaka, a u drugom redu dobijamo vrednost minuta tako sto iscitavamo poslednja dva znaka.

Sada na postojeću vrednost minuta dodajemo još 5 min, da bi dobili početak sledećeg časa. Zatim se formatira dobijen rezultat i zapisuje se u textbox za početak časa. Ista je situacija kada se između dva časa nalazi veliki odmor, samo što se tada ne dodaje 5 min, nego vrednost slajdera za određivanje dužine velikog odmora.

5. INTERFEJS SKOLSKOG ZVONA

Za prilagodjenje signala odlucio sam se za varijantu sa opto kaplerom, jer se sa njom dobija i galvansko odvajanje racunara i interfejsa. Sto je korisno u situacijama kada dodje do kvara na interfejsu. Jer se optokaplerom onemogucuje ostecenje COM porta.

Sema interfejsa:



Na donjem delu seme jetransformator i ispravljac koji sluzi da obezbedi napon za napajanje uredjaja. Sklopkom S1 se ukljucuje i iskljucuje uredjaj. Led 2 sluzi za indicaciju napajanja uredjaja.

CON1 je konektor za seriski port. Na pinu 5 je masa a na pinu 3 je izlaz TXD koji se koristi za kontrolu uredjaja. Otpornik R3 služi za ograničenje struje kroz led koja se nalazi u optokapleru.

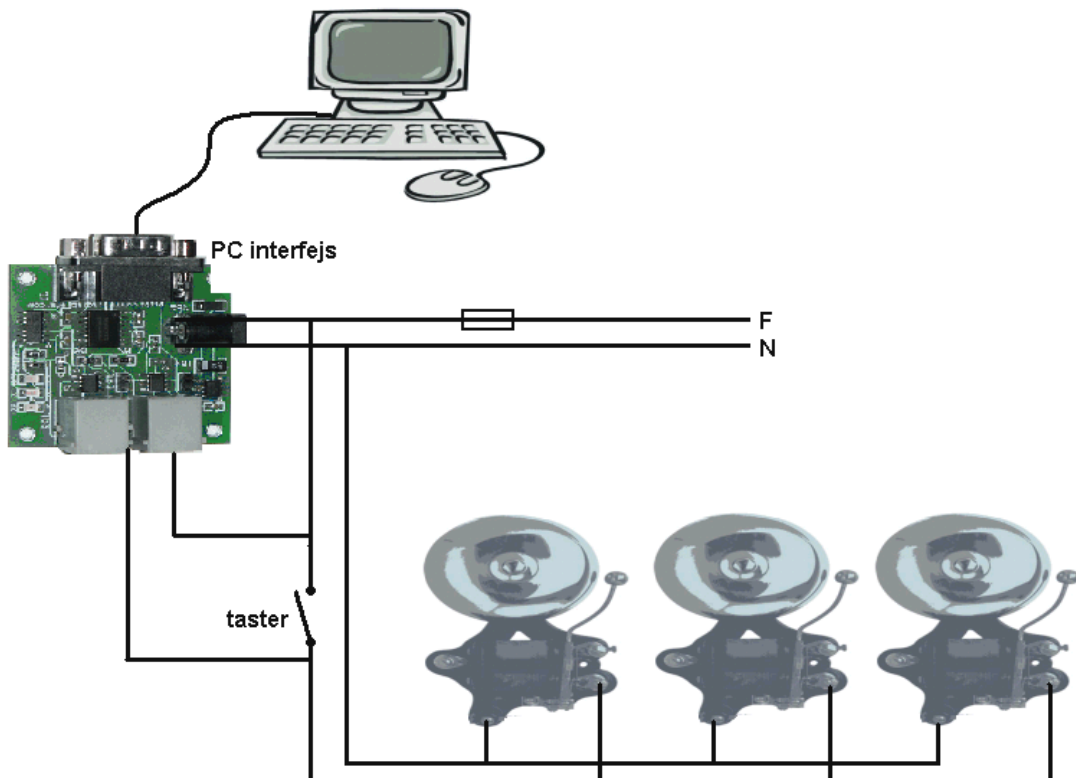
Preko opto kaplera je racunar galvanski odvojen od uredjaja. R2 služi za ograničenje bazne struje. T1 radi u prekidackom rezimu i služi za ukljucivanje i iskljucivanje relea.

LED1 služi za indicaciju kada je rele ukljucen. R2 služi za ograničenje struje kroz Led.

Dioda D1 služi da zastiti tranzistor od indukovane EMS koja nastaje pri iskljucenju relea.

6. NACIN POVEZIVANJA UREDJAJA SA SKOLSKIM ZVONOM

Sema povezivanja:



Kao što je prikazano na semi uređaj se spaja na mrežu 220V, preko koje dobija napajanje. Kontakti relea se spajaju paralelno postojećem tasteru za zvono. I kabal za povezivanje sa racunarom se ukljuci u racunar.

SADRZAJ:

1. UOPSTENO O UPRAVLJANJU PROCESIMA POMOCU RACUNARA.....2
2. UOPSTENO O SERISKOM PORTU.....
3. O PROGRAMU ZA UPRAVLJANJE SKOLSKIM ZVONOM.....
4. O KODU PROGRAMA.....
5. INTERFEJS SKOLSKOG ZVONA.....
6. NACIN POVEZIVANJA UREDJAJA SA SKOLSKIM ZVONOM.....