

Микропроцесорски системи

9. блок: Итерфејси за комуникацију *Serial Communication Interface – SCI*

др Срђан Митровић, дипл. инж.

зимски семестар, 2016./2017. год.

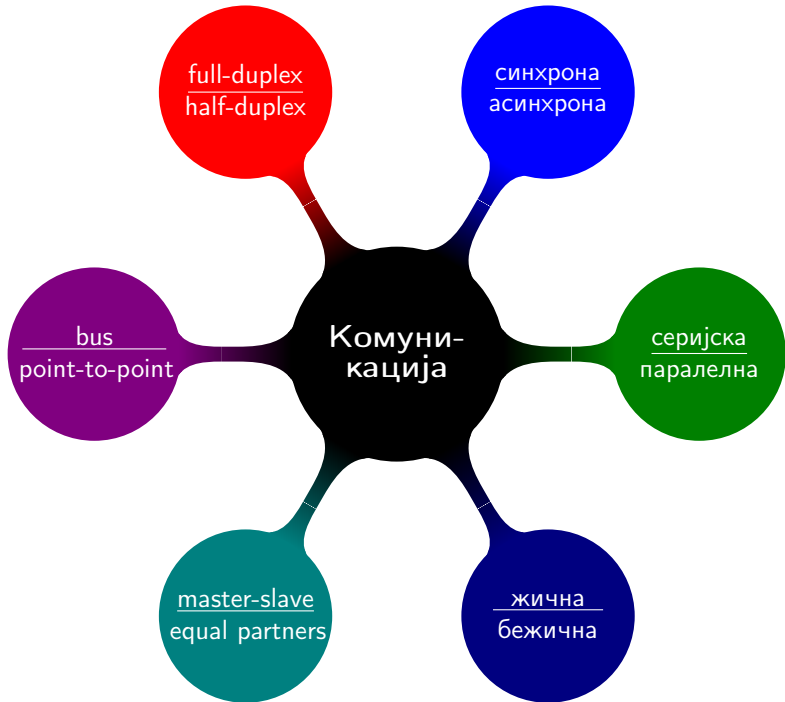
Наставна питања

1 Увод

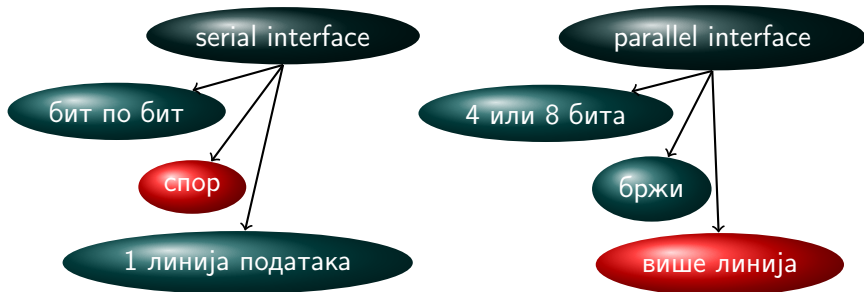
2 SCI

- Data Transmission
- Synchronization and Error Recognition
- Baud Rate Generation
- RS-232
- RS-422
- USART

3



Серијска vs паралелна комуникација



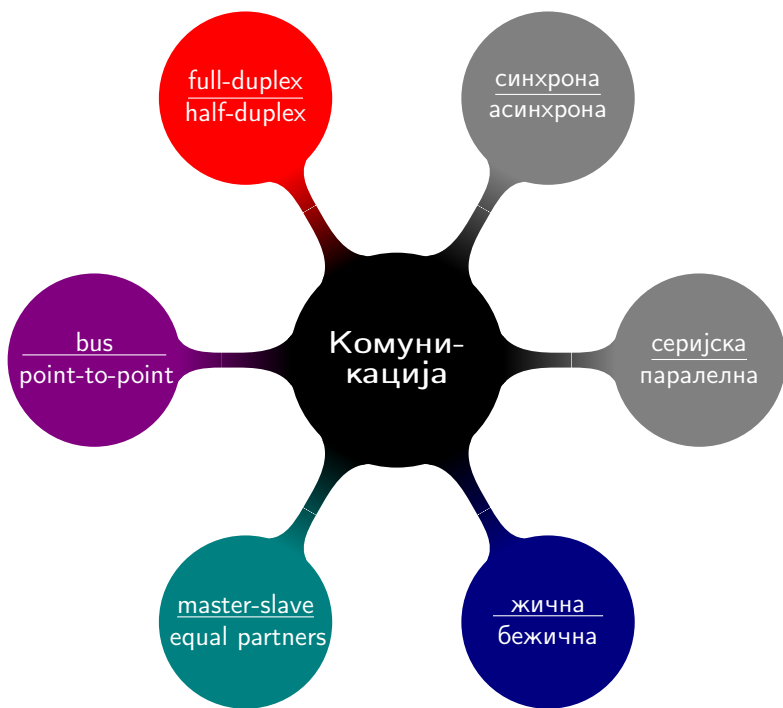
Синхрона vs асинхрона комуникација

синхрона комуникација

- тактови пријема и предаје су усклађени
 - додатна линија такта
 - реконструкција сигнала такта

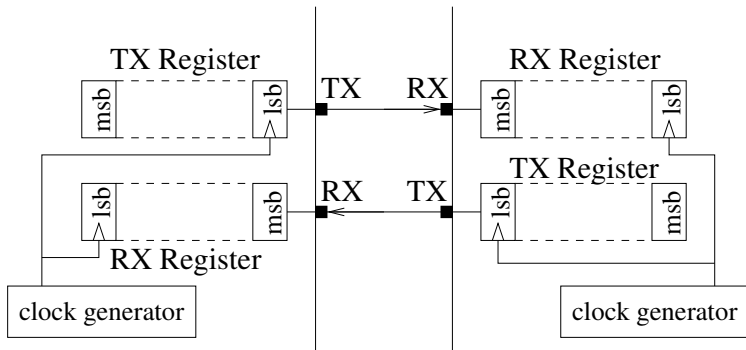
асинхрона комуникација

- тактови нису усклађени
- неопходан *oversampling*
- *START* и *STOP* бит
- **спорије**



SCI – Serial Communication Interface

UART – Universal Asynchronous Receiver Transmitter



- користи две линије:
 - TXD – предаја
 - RXD – пријем

- UART није протокол
- UART је модул

UART – Конфигурациони параметри

Број битова за податке (*Number of Data Bits*) Број битова за подате се обично може одабрати из широког дујапазона. АТмега дозвољава број из опсега [5,9]

Бит парности (*Parity Bit*) Корисник одлучује да ли ће, или неће користити бит парности. Када се користи може бити паран или непаран. Ако је бит парности сетован као паран, он је једнак 0 када је збир јединица битова података паран. Непаран *Parity Bit* се користи супротно.

Стоп бит(ови) (*Stop Bits*) У општем случају корисник може одабрати да ли ће користити једна или два стоп бита.

UART – Конфигурациони параметри

Брзина преноса (*Baud Rate*) UART модул садржи регистар који омогућава кориснику да одабере одговарајућу брзину преноса у битовима у секунду (*bit per second – bps*). Брзина преноса се најчешће налази у интервалу од 9 600 до 115 200 bps. **Фреквенција такта** утиче на bps.

Номенклатура

$$D \{E|O|N\} S$$

где је D број битова података,

S је број стоп битова,

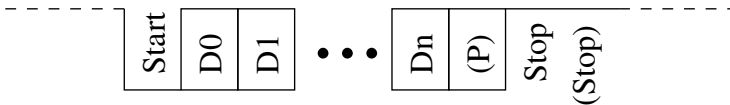
$E|O|N$ значе *Even* (паран), *Odd* (непаран), *No parity*.

Пример 1: 8E1

Број старт битова је увек 1.

Пренос података

- принцип *Non Return to Zero (NRZ)*
- “1” одговара вишем напону
- “0” одговара нижем напону
- подаци су имеђу бар једног старт и стоп бита



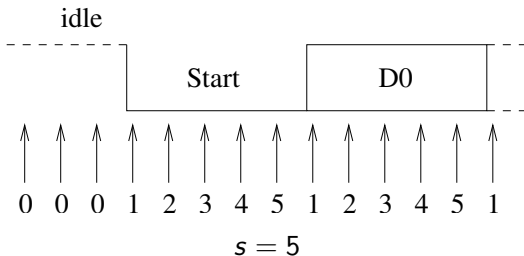
- када нема података линија је “*high*”
- пријемник и предајник се морају подесити на исти начин

Синхронизација и детекција грешке

- Тактови пријема и предаје су међусобно независни.
- Одабиром *baud rate* RX “зна” фреквенцију, али не и тренутак доласка поруке.
- RX мора да се синхронизује са силазном ивицом старт бита.
- *Drift* осцилатора је различит од нуле.
- Након синхронизације на почетку, могуће је да се RX раздеси до краја пријема поруке.

Синхронизација и детекција грешке

- Примењује се *Oversampling*
- RX страна узима узорке s пута по биту.
- Типичан броје s је 16



- Код **ATmega16** s је 16, а користи узорке 8, 9 и 10

Синхронизација и детекција грешке

Грешка због шума:

- Ако се користи бит парности може се детектовати грешка у 1 биту,
- *parity error* се сетује у UART регистру статуса.

Грешка због раздешене синхронизације:

- не препознаје се стоп бит
- генерише се *frame error*

Data overrun:

Нови пакет пре читања старог из бафера.

Брзина преноса

TX:

- Брзина преноса се генерише на основу системског такта.
- ↑ *baud rate register* (попут *output compare* регистра) генерише периодични такт сигнал.
- Овај сигнал се затим скалира (s пута) да би се добила одговарајућа брзина преноса, помоћу *prescaler*-а.

RX:

- -||-||-||-||-||-||-||-||-||-||-||-||-||- осим ...
- сигнал такта се узима пре прескалера
- пријемник узоркује s пута брже него предајник
-

$$bit\ rate \leq \frac{1}{s} \cdot RX\ clock$$

Брзина преноса

- скуп могућих брзина преноса зависи од фреквенције такта
- није могуће прецизно постићи сваку жељену брзину преноса

Пример 2

$$f = 8\text{MHz}$$

$$s = 8$$

Boud rate = 0.5 Mbps можемо остварити тачно

Boud rate = 115.2 kbps не можемо остварити тачно, јер

$$\frac{8\text{MHz}}{8 \cdot 115.2\text{KHz}} = 8.68$$

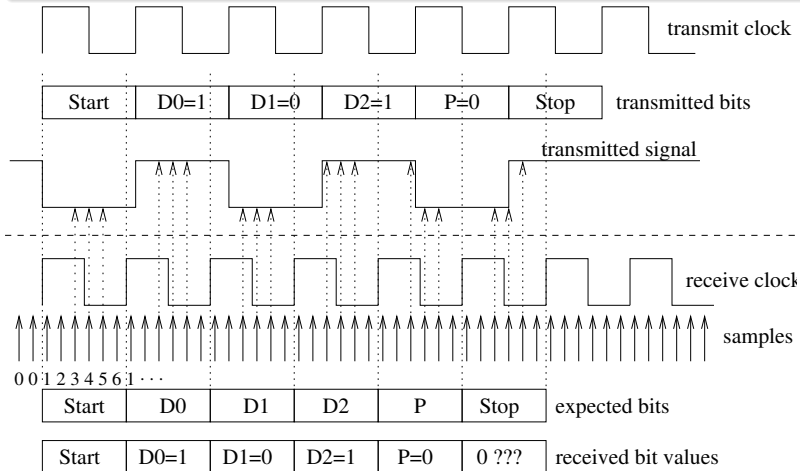
Брзина преноса

Baud Rate (bps)	$f_{osc} = 8.0000 \text{ MHz}$			
	$U2X = 0$		$U2X = 1$	
	UBRR	Error	UBRR	Error
2400	207	0.2%	416	-0.1%
4800	103	0.2%	207	0.2%
9600	51	0.2%	103	0.2%
14.4k	34	-0.8%	68	0.6%
19.2k	25	0.2%	51	0.2%
28.8k	16	2.1%	34	-0.8%
38.4k	12	0.2%	25	0.2%
57.6k	8	-3.5%	16	2.1%
76.8k	6	-7.0%	12	0.2%
115.2k	3	8.5%	8	-3.5%
230.4k	1	8.5%	3	8.5%
250k	1	0.0%	3	0.0%
0.5M	0	0.0%	1	0.0%
1M	–	–	0	0.0%
Max ⁽¹⁾	0.5 Mbps		1 Mbps	

- UBRR – USART Baud Rate Register
- Asynchronous Normal Mode $\rightarrow U2X = 0$
- Asynchronous Double Speed Mode $U2X = 1$

Пример 3

- $s = 6$
- формат: 3E1
- Voting Samples: 3, 4, and 5.



Грешка синхронизације

Релативна грешка синхронизације

$$B' = \frac{f_{osc}}{s \cdot C}$$

C – вредност бројача

$$\Delta[\%] = \frac{B' - B}{B} \cdot 100\%$$

Пример 2

$$f = 8\text{MHz} \quad s = 8$$

$$B = 115.2 \text{ kbps}$$

$$B' = \frac{8\text{MHz}}{8 \cdot 9} = 111111$$

$$\frac{8\text{MHz}}{8 \cdot 115.2\text{KHz}} = 8.68 \Rightarrow C = 9$$

$$\Delta[\%] = \frac{111111 - 115200}{115200} \cdot 100\% = -3.5\%$$

Максимална дужина поруке

f_R реална фреквенција предаје

f_T реална фреквенција пријема

s број *oversampling*

v_n редни број бита за гласање

k број битова поруке

$$k < \frac{v_n \cdot f_T}{s \cdot (f_R - f_T)}, \quad f_R > f_T$$

$$k < \frac{s \cdot f_R - (v_n + 1) \cdot f_T}{s \cdot (f_T - f_R)}, \quad f_R < f_T$$

RS-232

The UART itself is only the implementation of the asynchronous serial communication protocol but does not define any physical characteristics of the interface, like the voltage levels used. In the microcontroller, the bits are just mapped to the controller's voltages for 0 and 1. With a suitable voltage translation device, however, a UART can be used with a wide range of physical interfaces. The most common one is the RS-232 standard which can be found in PCs.

The RS-232 is a single-ended serial connection intended for point to point communication. It defines communication equipment types, electrical specifications, signal lines and signal timing. The RS-232 standard provides for 25 lines, although PCs generally only use 9 of these lines. Of these nine bits, only two (RXD and TXD) are used by the UART. A third (GND) is required because the connection is single-ended. The rest are control lines that can be used with more advanced communication protocols.

RS-232

The RS-232 specification defines the signal voltage levels to be within ± 3 - 15 V, and devices must be able to withstand a maximum of ± 25 V. Control lines use positive logic, data lines use negative logic.

Since the microcontroller is not capable of handling the voltages required by the RS-232 interface, a conversion IC has to be used that translates the GND and VCC levels of the controller to the voltage levels used by the RS-232 and vice versa (MAX232).

The RS-422 standard is also designed for point-to-point communication, just like the RS-232. However, it uses differential lines (both RXD and TXD consist of twisted-pair wires) for transmission. Hence, it does not need a common GND wire. The RS-422 can be used instead of the RS-232 to extend the transmission range of the UART. It is also the better choice in noisy environments.

USART – *Universal Synchronous Asynchronous Receiver Transmitter*

The *Universal Synchronous Asynchronous Receiver Transmitter* (USART) extends the functionality of the UART by a synchronous transmission module. The USART therefore has an additional third line which carries a clock signal. In synchronous mode, the clock signal is generated by one of the communication partners and is used by both for data transmission and reception. Naturally, this synchronous communication makes the oversampling mechanism of the asynchronous module unnecessary, so the synchronous mode is by a factor s faster than the asynchronous mode. The USART module combines the logic for both synchronous and asynchronous communication. If asynchronous communication is used, the clock line is free and can generally be used as a normal digital I/O pin.

Микропроцесорски системи

9. блок: Итерфејси за комуникацију *Serial Communication Interface – SCI*

др Срђан Митровић, дипл. инж.

зимски семестар, 2016./2017. год.