



Vhodno-izhodne naprave (VIN)

Predavanja

4. Serijski prenos podatkov

Robert Rozman

rozman@fri.uni-lj.si

1. Domača naloga – V/I naprave

VIN 2021/22 Domača naloga

Podan je seznam naprav (vhodna, izhodna, vhodno/izhodna):

1. Tipkovnica (ang. keyboard)
2. Optična miška (angl. optic mouse)
3. Optični skener (ang. scanner)
4. Biometrični skener prstnih odtisov (ang. biometric fingerprint scanner)
5. Zaslon LCD (ang. liquid crystal display)
6. Zaslon na dotik (ang. touchscreen)
7. Projektor (ang. projector)
8. Brizgalni tiskalnik (ang. inkjet printer)
9. Laserski tiskalnik (ang. laser printer)
10. 3D tiskalnik (ang. 3D printer)
11. Bliskovni pomnilnik (ang. flash pomnilnik)
12. Zunanji pomnilnik HDD (ang. Hard Disk Drive)
13. Zunanji pomnilnik SSD (ang. Solid State Drive),
14. Čitalnik črtne kode (ang. barcode reader)
15. Mobilni čitalnik RFID (ang. RFID reader)
16. Fiksni čitalnik RFID (ang. RFID reader),
17. Spletna kamera (ang. webcam)
18. Mikrofon (angl. microphone)
19. Zvočnik (angl. speaker)
20. SensorTile.box (<https://www.st.com/en/evaluation-tools/steval-mksbox1v1.html>)
21. Napredna tipala z UI (npr. LSM6DSOX https://www.st.com/content/st_com/en/products/mems-and-sensors/inemo-inertial-modules/lsm6dsiox.html)
22. GPS sprejemniki (navigacija, pozicioniranje)
23. Tipala: pospeškometer, žiroskop, kompas in kombinacije
24. Logični (protokolski) analizatorji (npr. CAN, I2C, ...)
25. STM32: zajem, procesiranje zvoka
26. ADC in DAC pretvorniki
27. Temo predlagam po emailu

Predstavitev

Predstavitev (5 min): od 22. 3. 2022 dalje bo potekala vsak torek v okviru predavanj po predhodnem razporedu na učilnici.

Predstavitev vaše domače naloge - V/I naprave naj vsebuje zelo kratek povzetek v treh alinejah in seznam literature:

1. Kratka predstavitev naprave: slika, arhitektura, tehnologija, »grafični poster«
2. Povezava z računalnikom: prenosni medij, prenos podatkov
3. Funkcije in opis delovanja
4. Literatura

Oddaja poročila: 20. 3. 2022.

Skupno poročilo (ali individualna poročila) oddajte **v PDF obliki**, lahko dodate še grafično skico (»poster«) posebej v višji ločljivosti.

Vsebina

1. Asinhronski serijski prenos

- **UART** (Universal Asynchronous Receiver/Transmitter)
 - (RS232, RS422, RS485)
- **(CANBus)**

2. Sinhronski serijski prenos

- **I2C** (Inter-Integrated Circuit)
- **SPI** (Serial Peripheral Interface)
- **(USB)**

Gradivo:

- ❑ PROTOCOLS: UART - I2C - SPI - Serial communications

https://www.youtube.com/watch?v=lyGwvGzrqp8&t=25s&ab_channel=Electronoobs

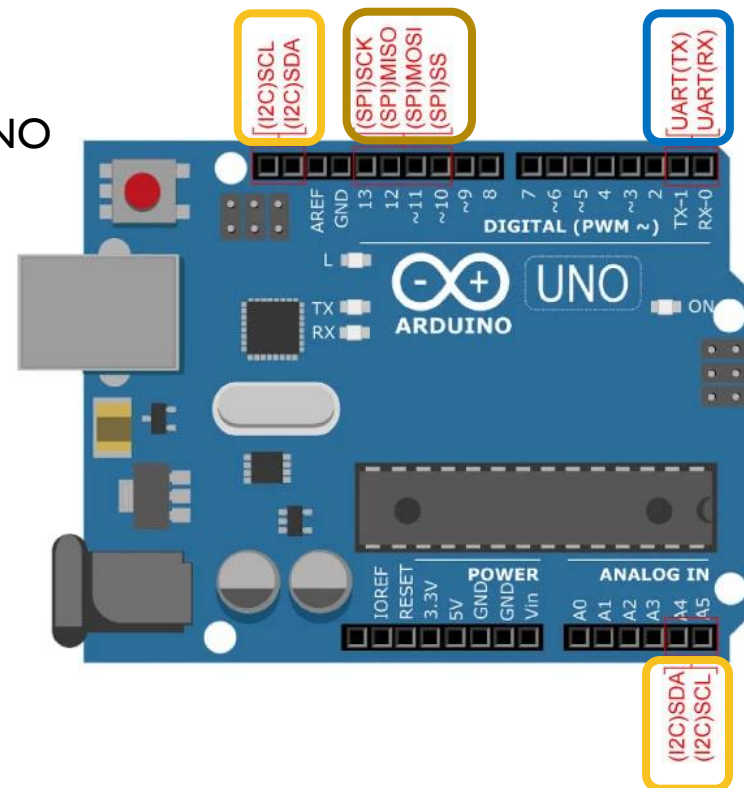
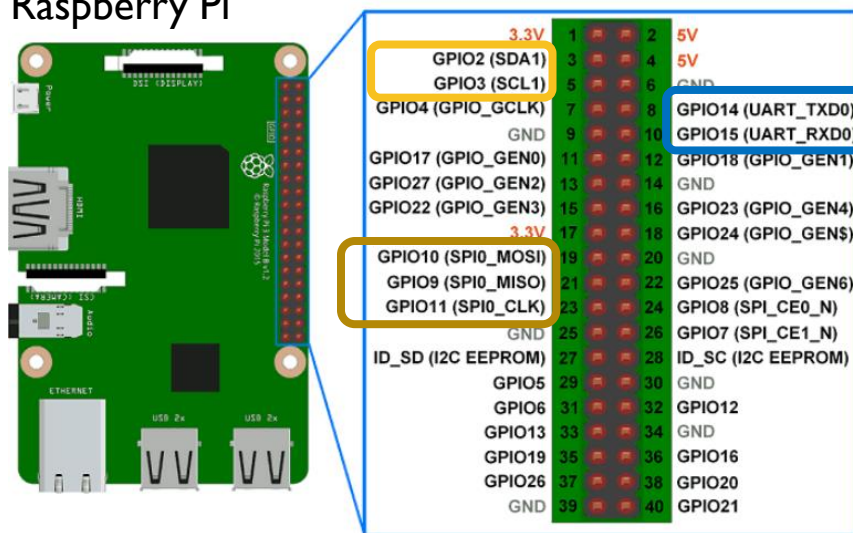
- ❑ Drugi viri so pri posameznih opisih

Uvod

1. Asinhronski serijski prenos
 - **UART** (Universal Asynchronous Receiver/Transmitter)
2. Sinhronski serijski prenos
 - **I2C** (Inter-Integrated Circuit)
 - **SPI** (Serial Peripheral Interface)

Arduino UNO

Raspberry Pi



<https://www.electronicwings.com/raspberry-pi/raspberry-pi-gpio-access>

<https://maker.pro/arduino/tutorial/common-communication-peripherals-on-the-arduino-uart-i2c-and-spi>

Uvod

1. Asinhronski serijski prenos

- **UART** (Universal Asynchronous Receiver/Transmitter)

- **USART1,2,3,4,5,6**

2. Sinhronski serijski prenos

- **I2C** (Inter-Integrated Circuit)

- **I2C1, I2C2, I2C3**

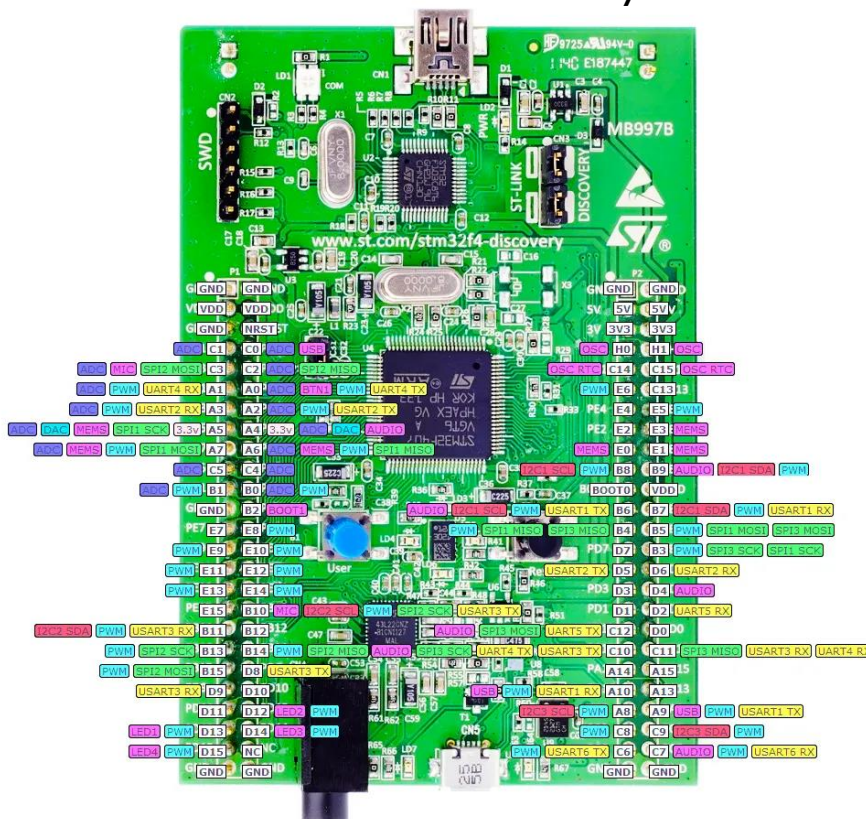
- **SPI** (Serial Peripheral Interface)

- **SPI1, SPI2, SPI3**

STM32F4 Discovery

P1

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	38
39	40
41	42
43	44
45	46
47	48
49	50



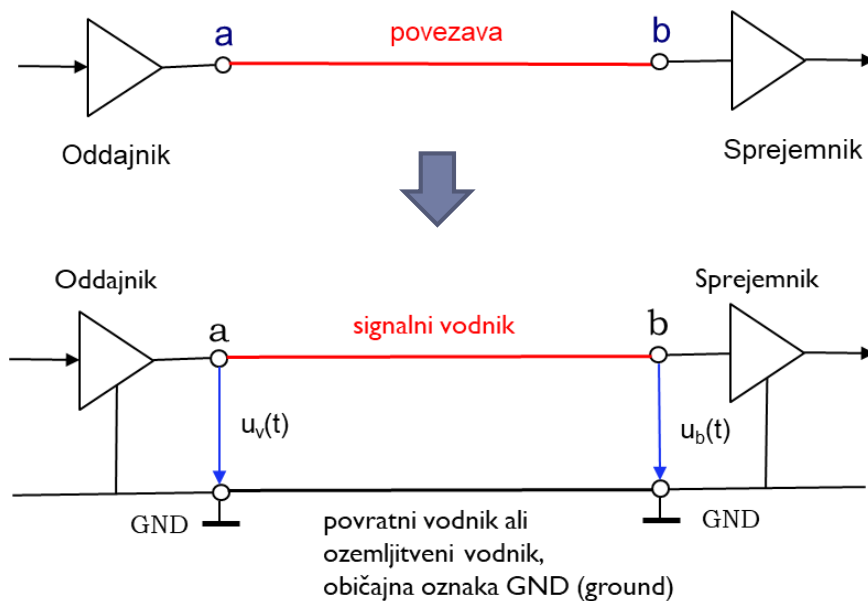
P2

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	38
39	40
41	42
43	44
45	46
47	48
49	50

- ❑ **Naprave:** računalnik, mikroprocesor, mikrokrmilnik, V/I naprave



- ❑ Prenosni medij je fiksni ali žičen – **električna povezava, ali linija**
simbolna predstavitev



❑ **Serijski način prenosa**

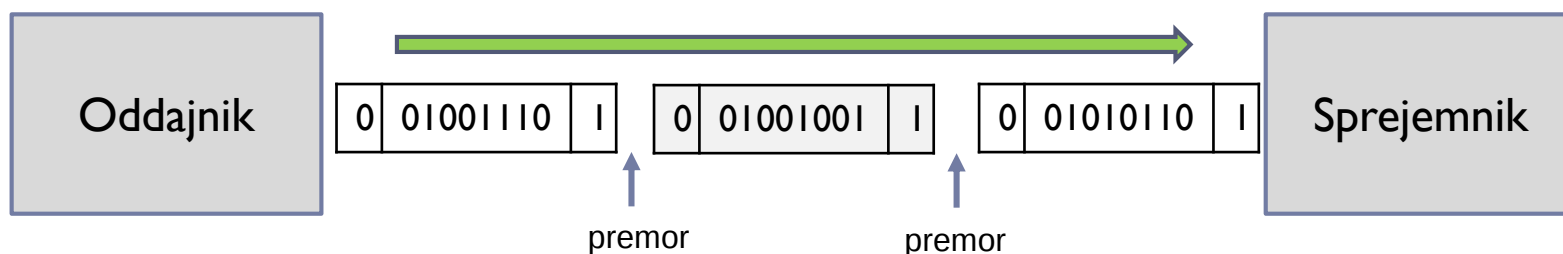
po povezavi se prenaša bit za bitom

❑ **KOMUNIKACIJSKI KANAL**

- Prenos podatkov poteka v **realnem času** (čas med oddajo in sprejemom podatkov je zelo kratek).
- Pri oddaji podatka je potrebna **paralelno – serijska** pretvorba.
- Pri sprejemu podatka je potrebna **serijsko – paralelna** pretvorba.
- Glede na **način sinhronizacije** razlikujemo:
 - **Asinhronski** serijski prenos podatkov
 - **Sinhronski** serijski prenos podatkov
- Potrebno je izvesti tudi **kodiranje podatkov** (pretvorba podatka v signal):
 - NRZI (Non Return to Zero Inverted)
 - PE (Phase Encoded)
 - RLL (Run Length Limited): 8b/10b, ...

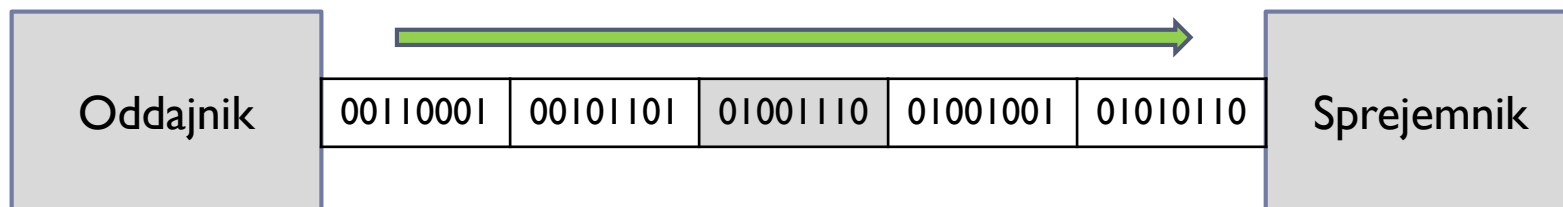
❑ Asinhronski serijski prenos

- UART (Universal Asynchronous Receiver/Transmitter)



❑ Sinhronski serijski prenos

- I2C (Inter-Integrated Circuit)
- SPI (Serial Peripheral Interface)

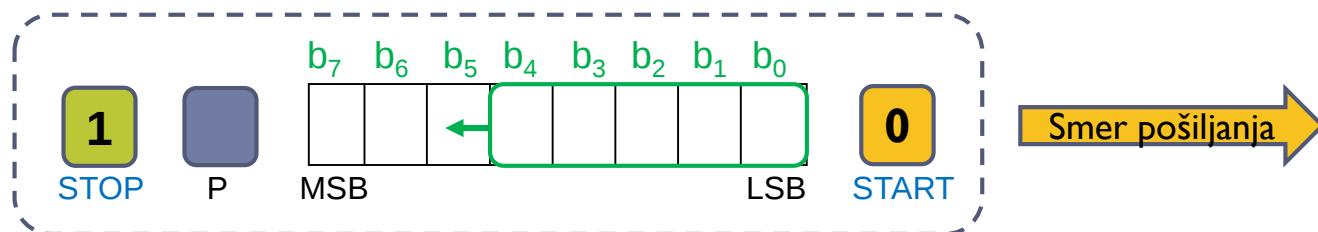


1. Asinhronski serijski prenos

- ❑ **Enota pošiljanja je znak**, ki je dolg od 7 do 12 bitov.
- ❑ **Biti se pošiljajo** po komunikacijskem kanalu.
- ❑ Oddajnik in sprejemnik imata vsak svojo uro. **Urin signal se NE pošilja** po komunikacijskem kanalu.
- ❑ Za pravilen prenos je nujno, da sta **frekvenci oddajne in sprejemne ure enaki in sinhronizirani**
- ❑ Sinhronizacija se vzpostavi za vsak znak posebej.
- ❑ Sinhronizacija sprejemne ure se z oddajno vzpostavi
 - s prvim bitom znaka (START) in
 - mora zagotavljati pravilen sprejem do zadnjega bita znaka (STOP).

- ❑ Prenos podatkov je relativno **počasen**.
- ❑ Primeren je za **neenakomeren prenos znakov** s presledki.
- ❑ Med znaki je pri oddaji lahko **poljubno dolg presledek**.
- ❑ Pošiljamo lahko kakršenkoli **tekst brez protokola** (protokol je samo format znaka).
- ❑ Dodatni biti :
 - Start in
 - stop bit in/ali
 - parnostni (paritetni) bitso v vsakem znaku, kar predstavlja **20% - 30% redundanco** pri prenosu.

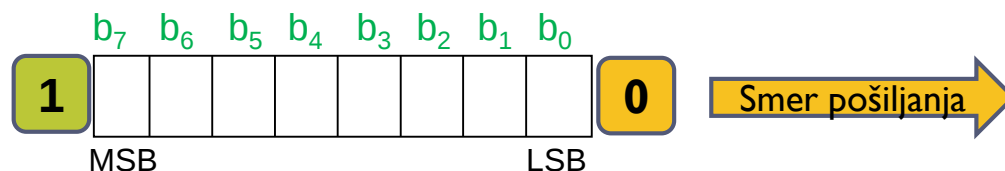
□ Znak sestavljajo:



- **Start bit** je vedno **logična 0** in določa začetek znaka. Pošilja se prvi in služi za vzpostavitev sinhronizacije med sprejemno in oddajno uro.
- **Podatkovni biti** - število bitov se v krmilniku lahko izbere (**5 do 8 bitov**).
 - Bit z najnižjo težo se vedno pošilja prvi: LSB (b_0).
 - Pri pošiljanju ASCII znakov se pošilja 7 ali 8 bitov.
- **Parnostni bit - P (paritetni bit)** ni obvezen. Pri oddaji se izračuna iz podatkovnih bitov in služi za kontrolo pravilnosti sprejetega znaka. V krmilniku se lahko izbere:
 - **Soda parnost** (število vseh enic podatkovnih bitov in parnostnega bita je sodo).
 - **Liha parnost** (število vseh enic podatkovnih bitov in parnostnega bita je liho).
- **Stop bit** je vedno **logična 1** in označuje konec znaka.

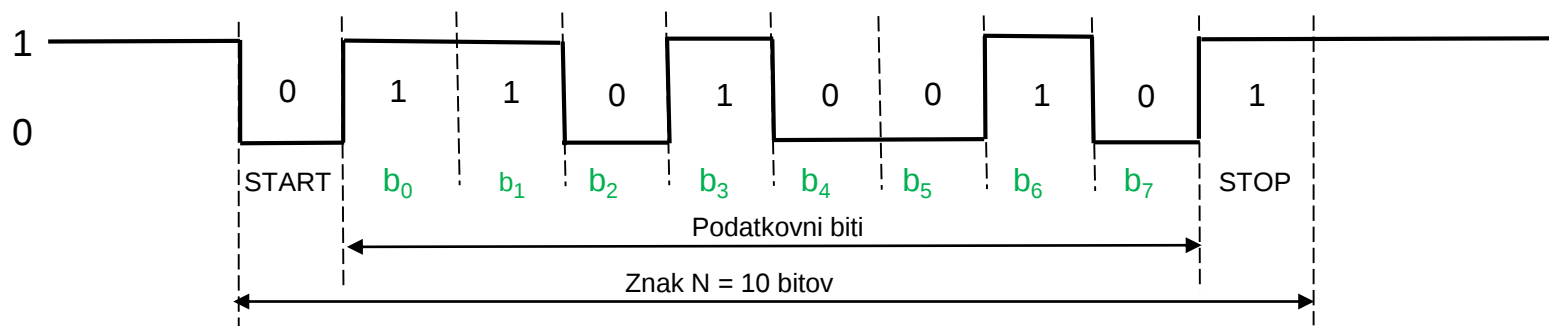
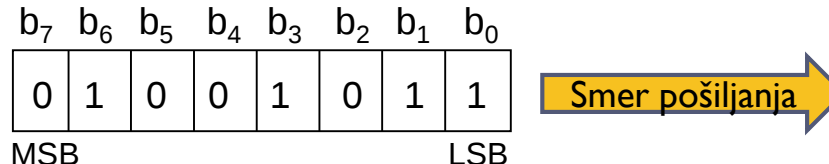
- Najbolj razširjena oblika formata serijskega asinhronskega prenosa pri osebnih računalnikih (PC) je **8-N-1** in pomeni:

- 8 podatkovnih bitov,
- brez parnostnega bita,
- en stop bit,
- dolžina znaka je skupaj s start bitom enaka 10 bitov.

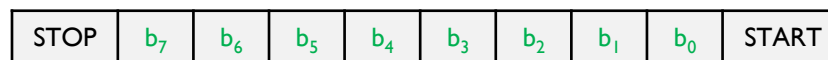


- Primer formata znaka ASCII "K" z enim stop bitom:

- ASCII "K" = 4B (Hex)

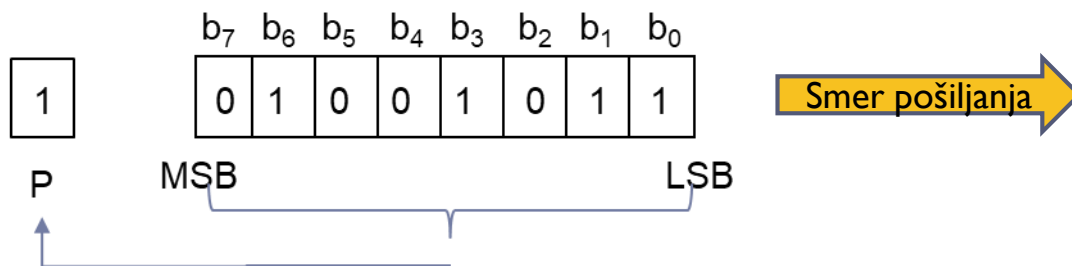


- Format znaka lahko podamo tudi takole:

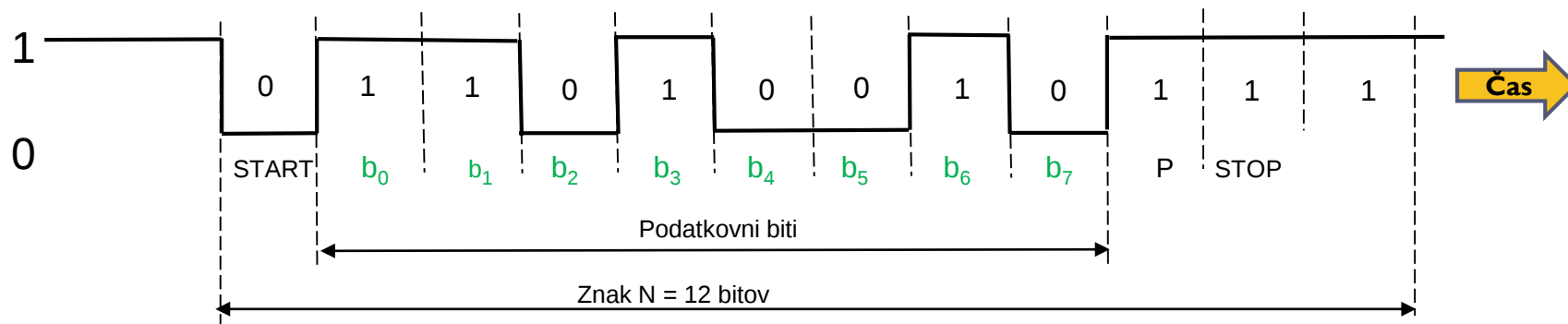


- Primer formata znaka ASCII “K” z **liho parnostjo** in dvema stop bitoma:

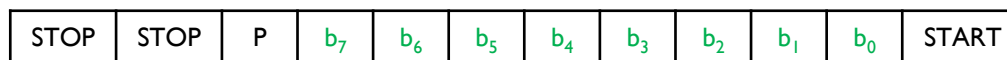
- ASCII “K” = 4B (Hex)
- $P = 1$



- Oblika prenosa: 8-N-1, kjer je $N=12$



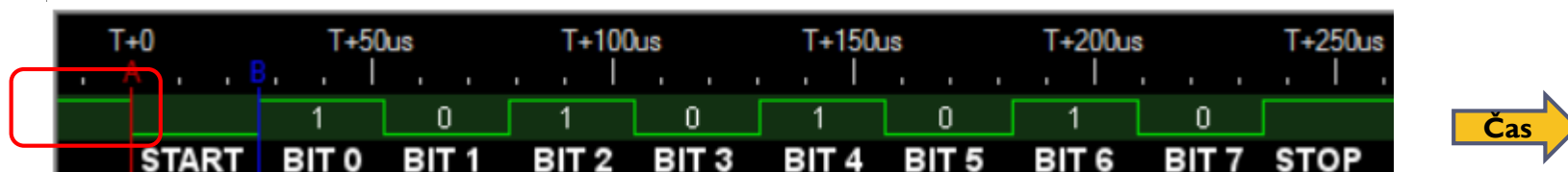
- Format znaka lahko podamo tudi takole (12 bitov):



□ Primer serijskega prenosa črke 'U'

(<http://www.robotroom.com/Asynchronous-Serial-Communication-I.html>)

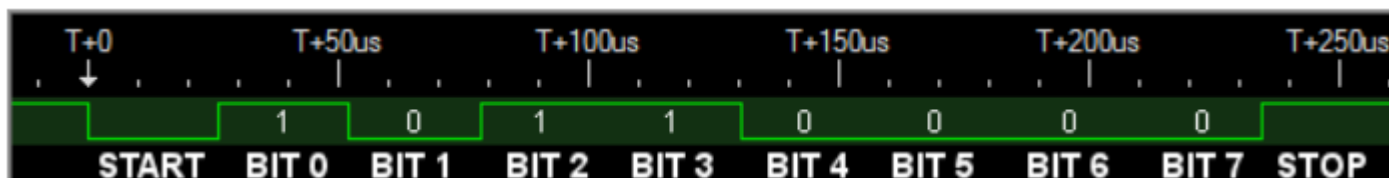
- Koda ASCII ima numerično vrednost 85, ali 55_H).
- Dvojiški 8-bitni zapis podatka ie **01010101**



- Pred prvim bitom START je signal na izhodnem pinu high (5V).
- Signal na izhodnem pinu se spremeni v low (0V), kar označuje začetek prenosa (START).
- Nato sledi prenos podatkovnih bitov (BIT 0 do BIT 7) in bit STOP.

□ Drugi primeri:

- CR (ASCII 13)
(carriage
return)



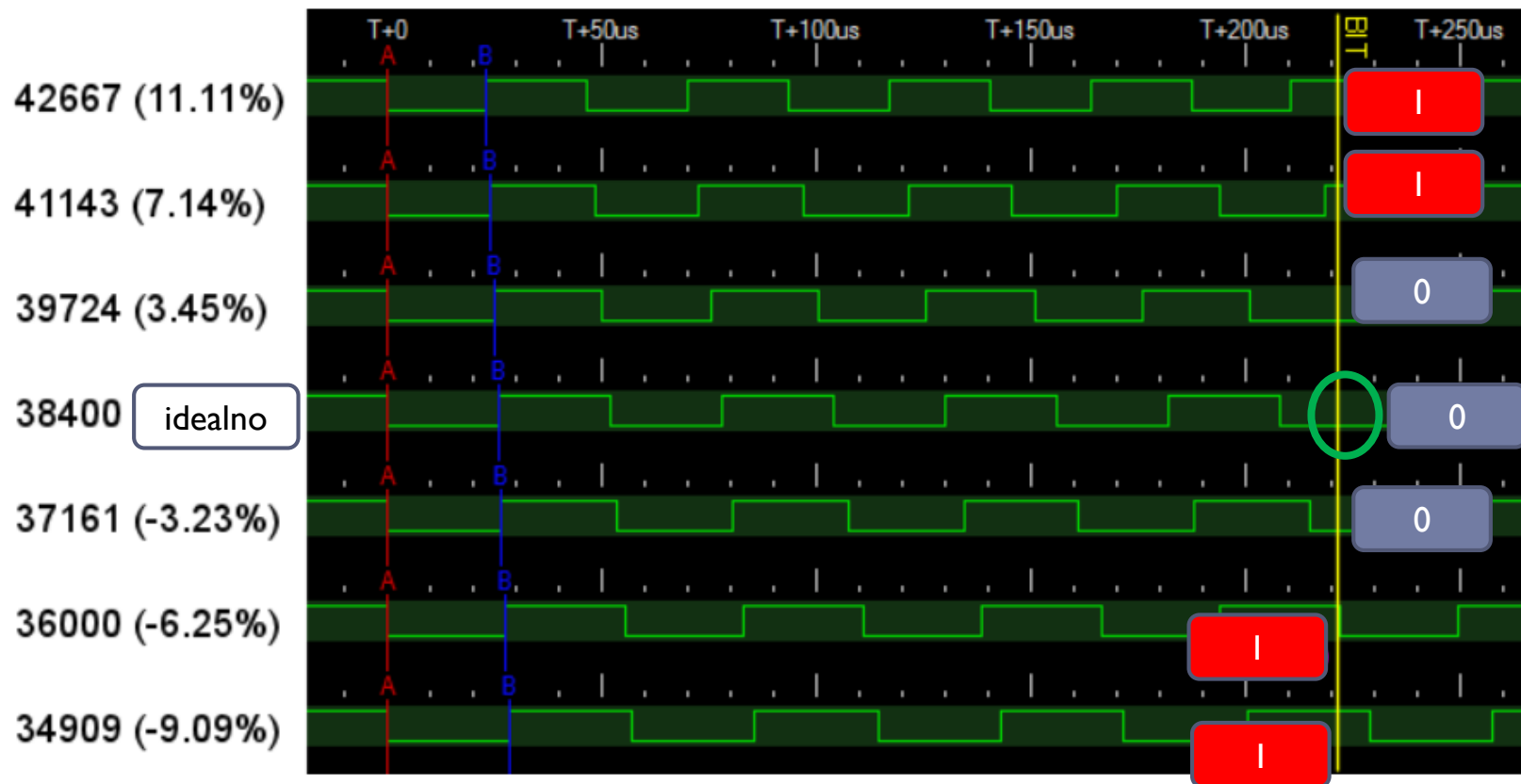
- Podatek = 0



❑ Primer napačnega prenosa (bitna hitrost je 38400 b/s).

- Napaka se pri prenosu zgodi v primeru, ko je oddajnik prehiter (> 38400 b/s) ali prepočasn (< 38400 b/s).
- razlika v času med oddajno in sprejemno hitrostjo $> 5\%$.

<http://www.robotroom.com/Asynchronous-Serial-Communication-2.html>



Univerzalni asinhronski sprejemnik/oddajnik (UART)

□ UART - Universal Asynchronous Receiver/Transmitter

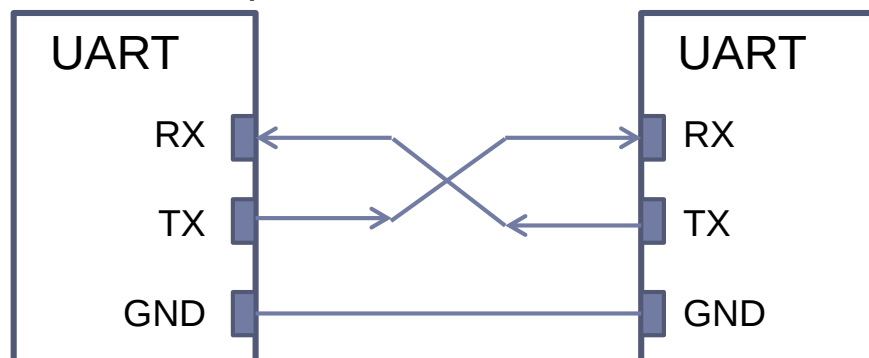
- UART kot samostojen čip v integraciji LSI (Large Scale Integration).
- UART kot logično vezje, ki je del mikrokrmilnika.
- UART ima 2 liniji za prenos podatkov med dvema napravama:
 - TX (ang. transmitter) – oddaja
 - RX (ang. receiver) – sprejem

„Universal“

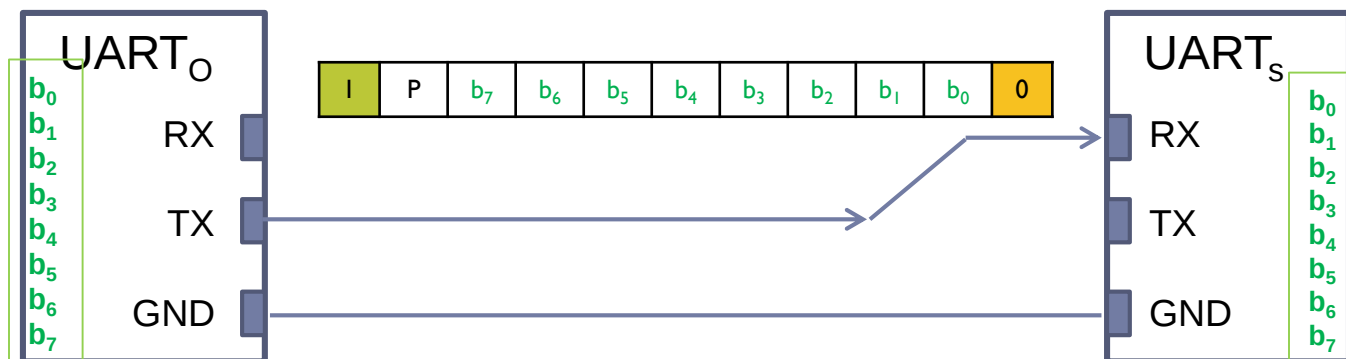
- UART is programmable.

„Asynchronous“

- Sender provides no clock signal to receivers



- Oddajni UART_O pretvori paralelni podatek v serijski podatek in ga z dodatnimi biti prenaša sprejemnemu UART_S, ki pretvori sprejeti serijski podatek v paralelni podatek.



- Uporabimo ga lahko v naslednjih načinih prenosa:
 - Napravi sta povezani samo v eno smer, kot oddajnik-sprejemnik (ang. simplex)
 - Napravi izmenoma pošiljata in sprejemata podatke (ang. half-duplex)
 - Napravi istočasno pošiljata in sprejemata podatke (ang. full-duplex)
- Hitrost prenosa podatkov podajamo kot baudna hitrost
 - Najbolj običajna, standardizirana **baudna hitrost je 9600**.
 - Druge standardne baudne hitrosti so:
1200, 2400, 4800, 19200, 38400, 57600 in 115200.
- UART je cenovno ugodna komunikacijska naprava.
- USART (Universal Synchronous Asynchronous Receiver Transmitter) - univerzalni sinhronski in asinhronski sprejemnik / oddajnik, ki omogoča tako asinhronski, kot sinhronski serijski prenos podatkov.

□ Zahteve za pravilen prenos

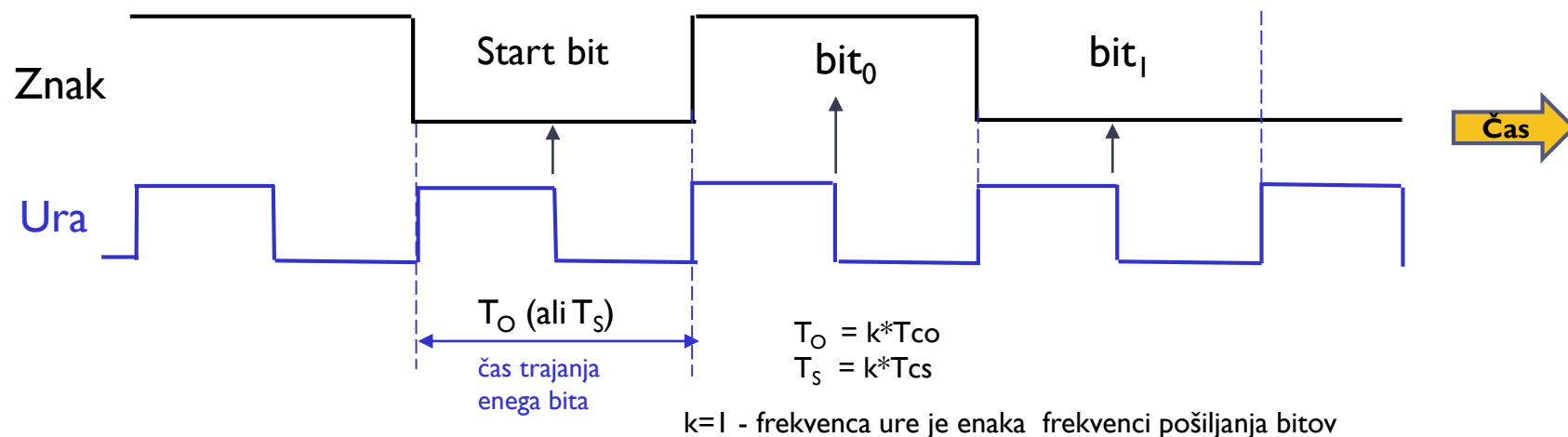
- Enak format, nastavitve na sprejemni in oddajni strani

- Število podatkovnih bitov
- Parnostni bit (da/ne), enaka parnost (soda ali liha)
- Enako število stop bitov (eden ali dva)

- Enaka frekvenca oddajne ure (f_{co}) in sprejemne ure (f_{cs}): $f_{co} = f_{cs}$

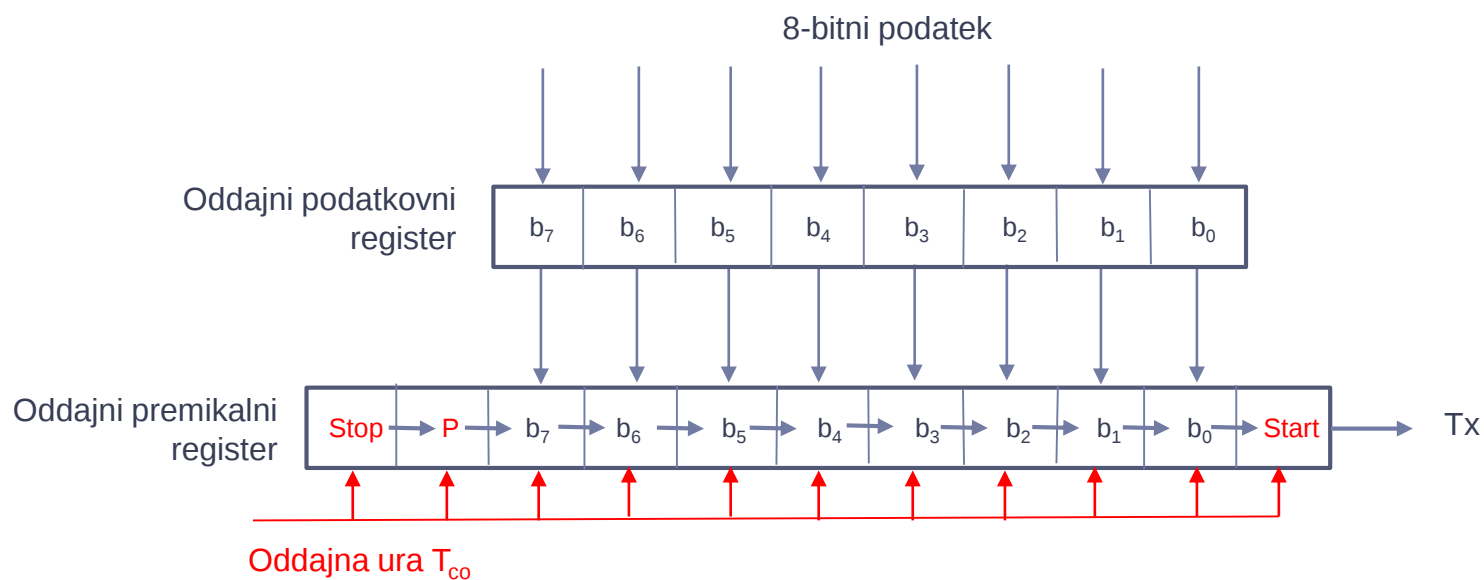
co – clk oddaja, cs – clk sprejem

- Periodi sprejemne ure (T_{cs}) in oddajne ure (T_{co}) se lahko razlikujeta za največ 4 - 5%.



□ UART – Oddaja

- Paralelno serijska pretvorba
- Uokvirjanje znaka (ang. framing): start bit, kontrolni parnostni bit, stop bit)
- Oddaja z izbrano baudno hitrostjo



UART – Oddaja

Oddajna ura - Tco

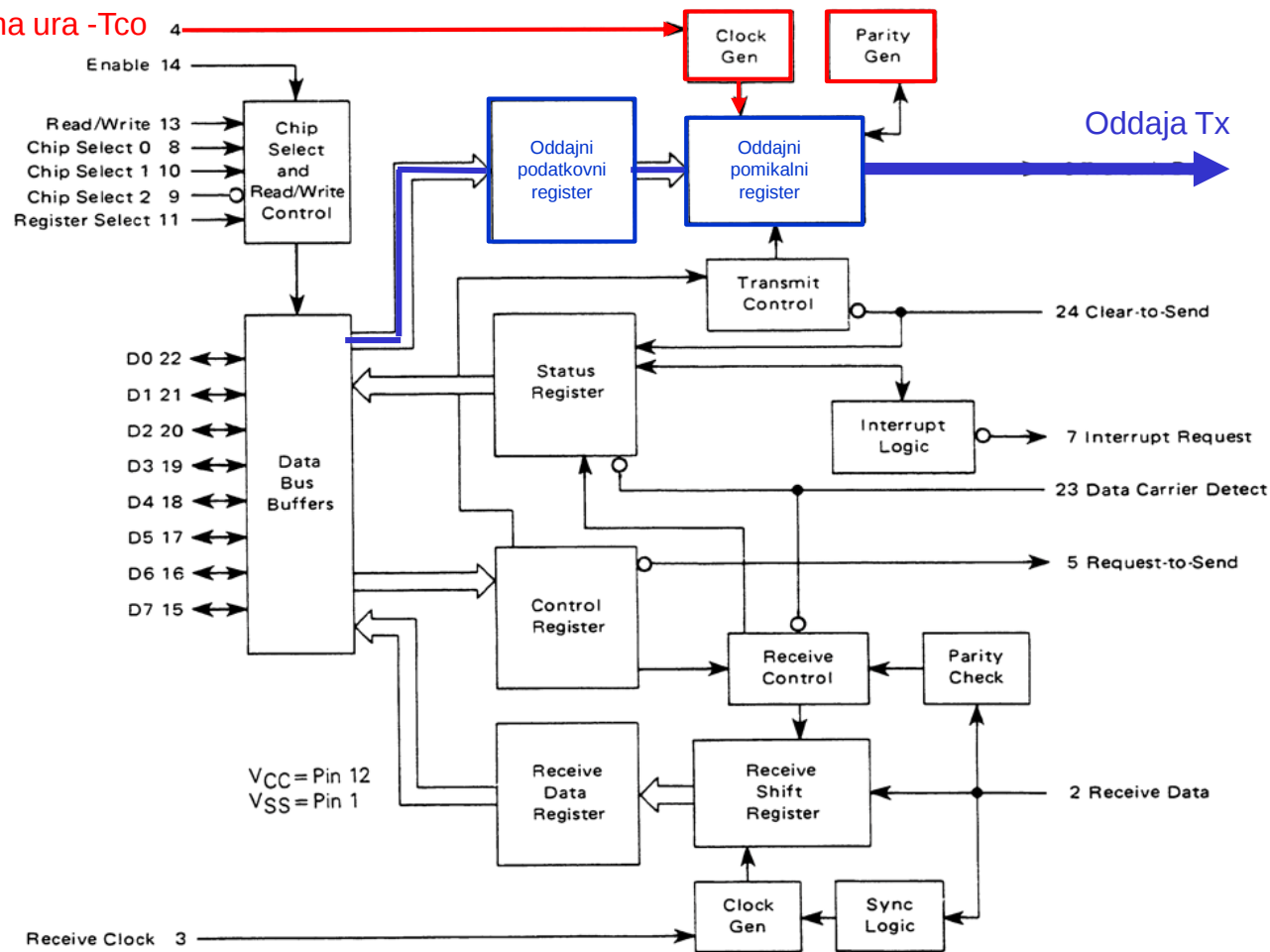
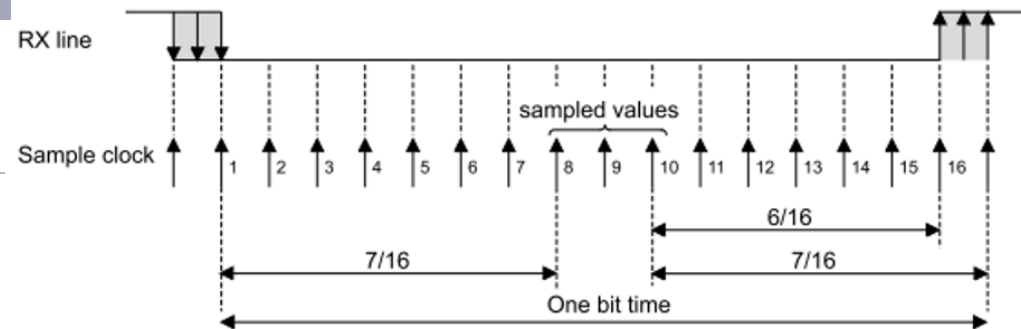


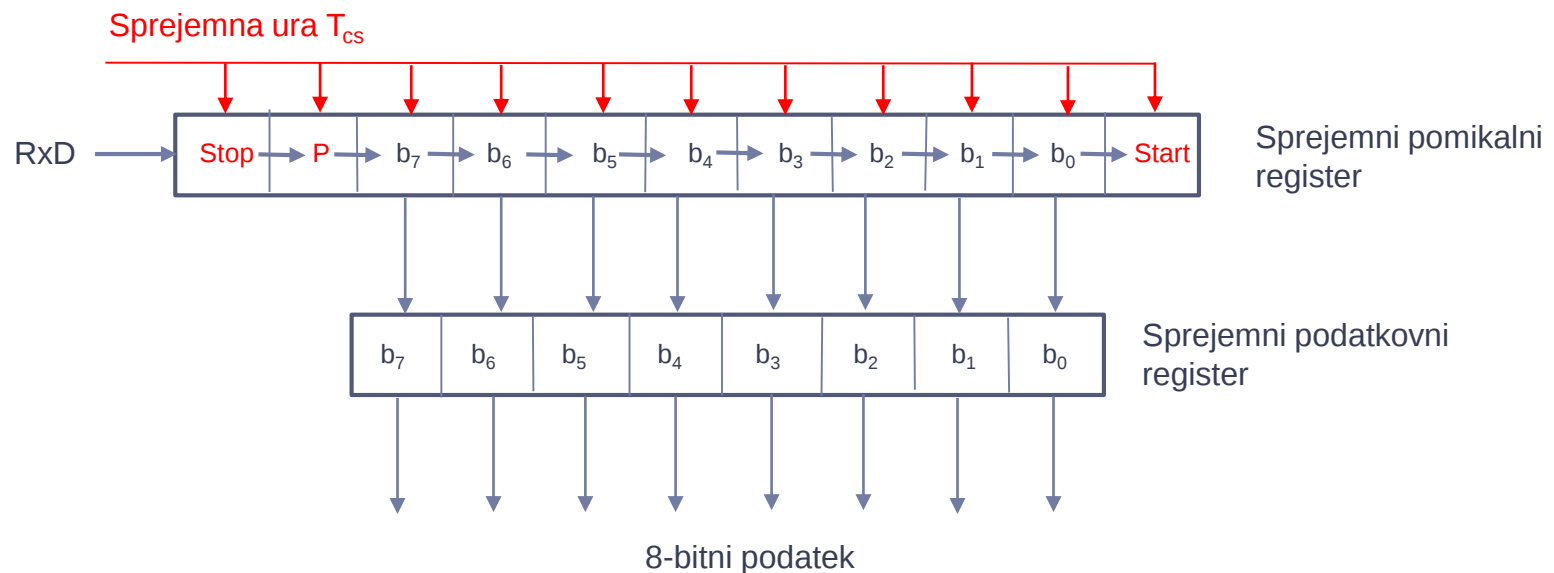
Figure 301. Data sampling when oversampling by 16



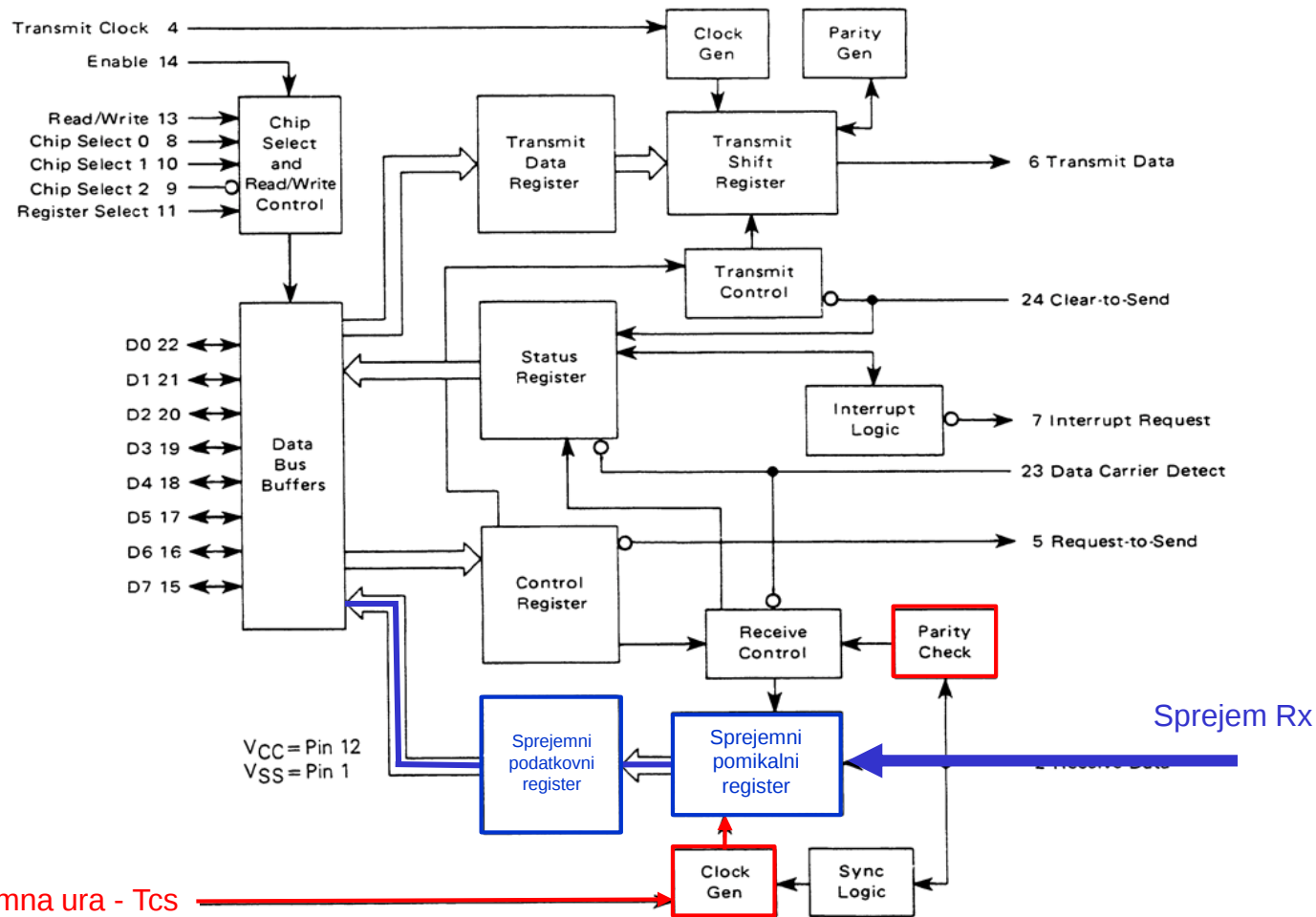
UART – Sprejem

- Serijsko paralelna pretvorba
- Kontrola pravilnosti (parnostni bit)
- Odstranitev okvirja (start, stop bit in parnostni bit)

Sprejemnik običajno bere stanje 8 ali 16x pogostejše (angl. Oversampling)

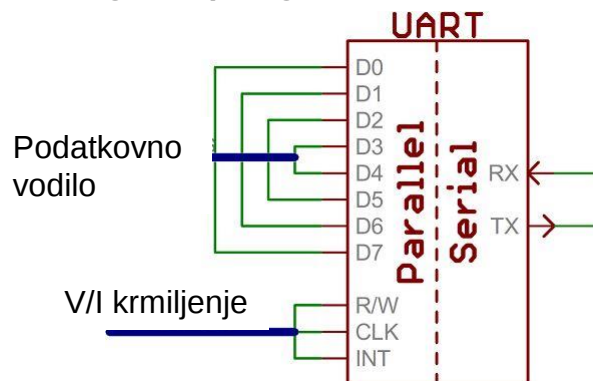


UART – Sprejem



Sprejemna ura - Tcs

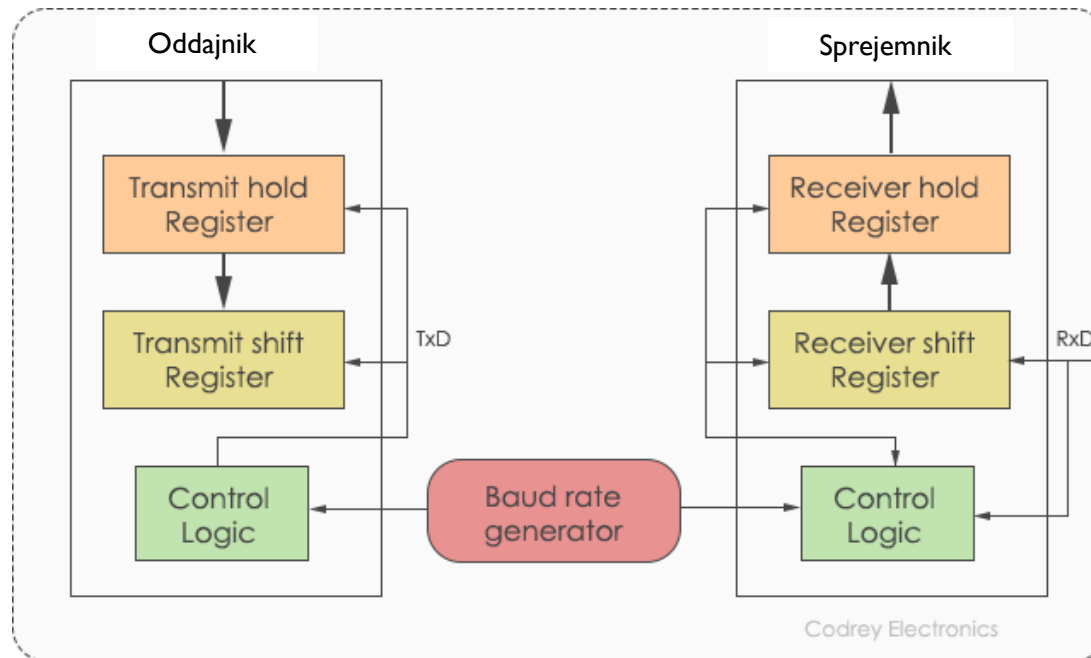
- ❑ UART je mogoče programirati, tako da določimo parametre prenosa:



- Baudno hitrost (baud rate) – baudna hitrost = bitna hitrost
- Število podatkovnih bitov
- Parnost (liha, soda, brez)
- Število stop bitov
- Stanje na kontrolnih izhodih
- Prekinitev ob sprejemu znaka
- Prekinitev po oddaji znaka
- Prenos z dvojnim izravnovanjem ali s FIFO vmesnikom

□ UART - Blok diagram

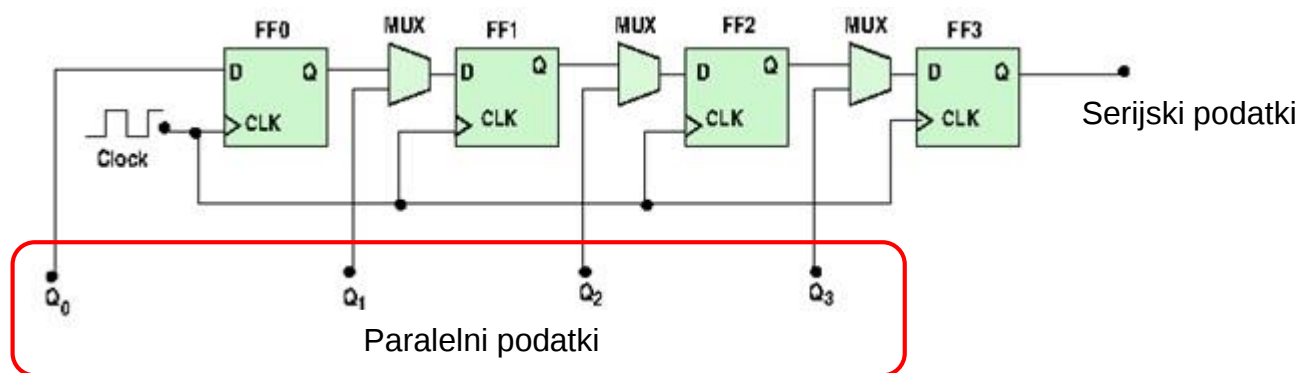
- Sprejemnik
- Oddajnik



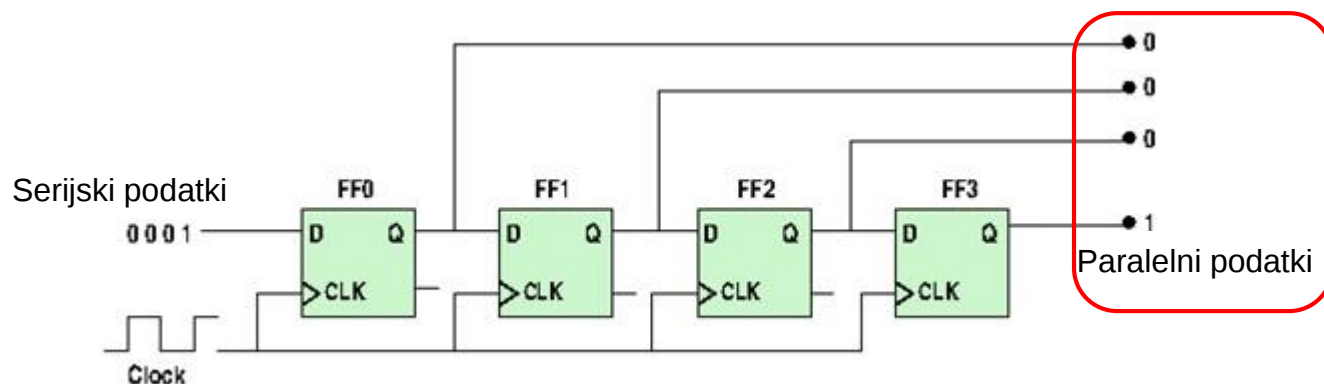
<https://www.codrey.com/embedded-systems/uart-serial-communication-rs232/>

- 'Transmit hold Register' in 'Transmit shift Register' - podatki za pošiljanje
- 'Receiver hold Register' in 'Receiver shift Register' - sprejeti podatki
- 'Control logic' – krmiljenje branja in pisanja
- 'Baud rate generator' – generator baudne hitrosti definira hitrost pri kateri morata oddajnik in sprejemnik pošiljati oz. sprejemati podatke.

❑ Paralelno-serijska pretvorba



❑ Serijsko-paralelna pretvorba

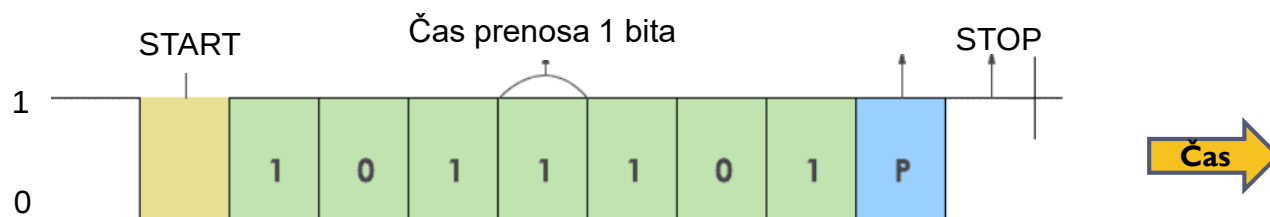


□ Protokol prenosa podatkov z UART

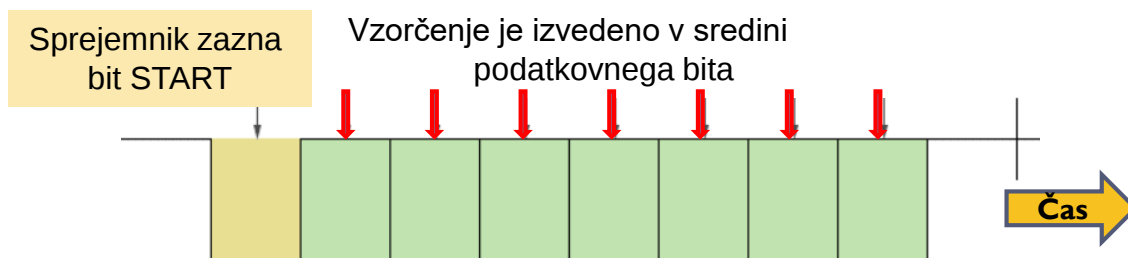
- Format protokola



- Oddajnik pošlje podatek po prenosni liniji TxD (LSB bit je prvi poslan).

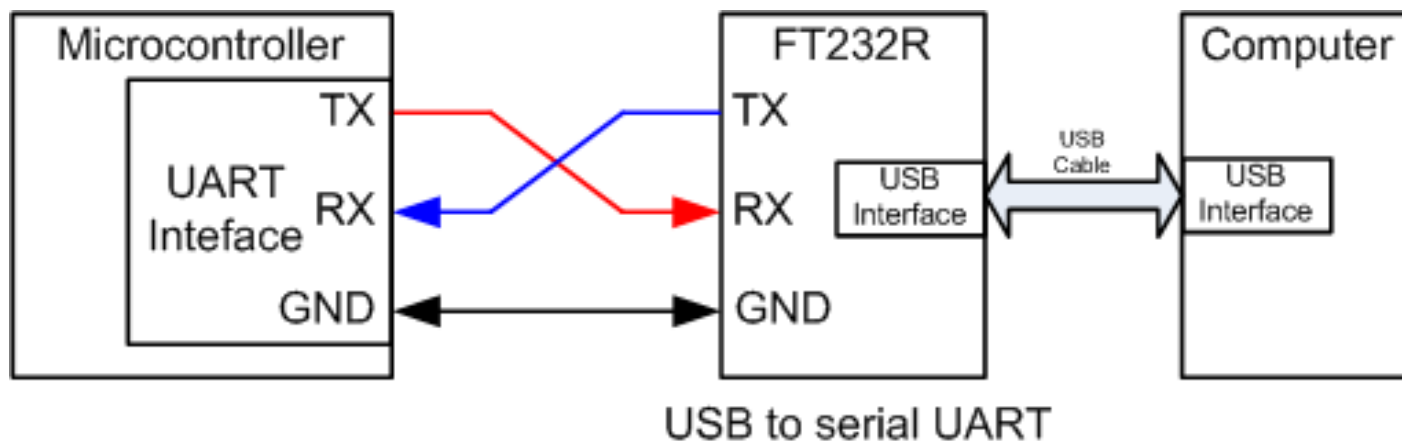


- Sprejemnik sprejme podatek po prenosni liniji RxD (LSB bit je prvi prejet) in
 - zazna bit START
 - izvede vzorčenje



Običajna povezava UART <-> PC

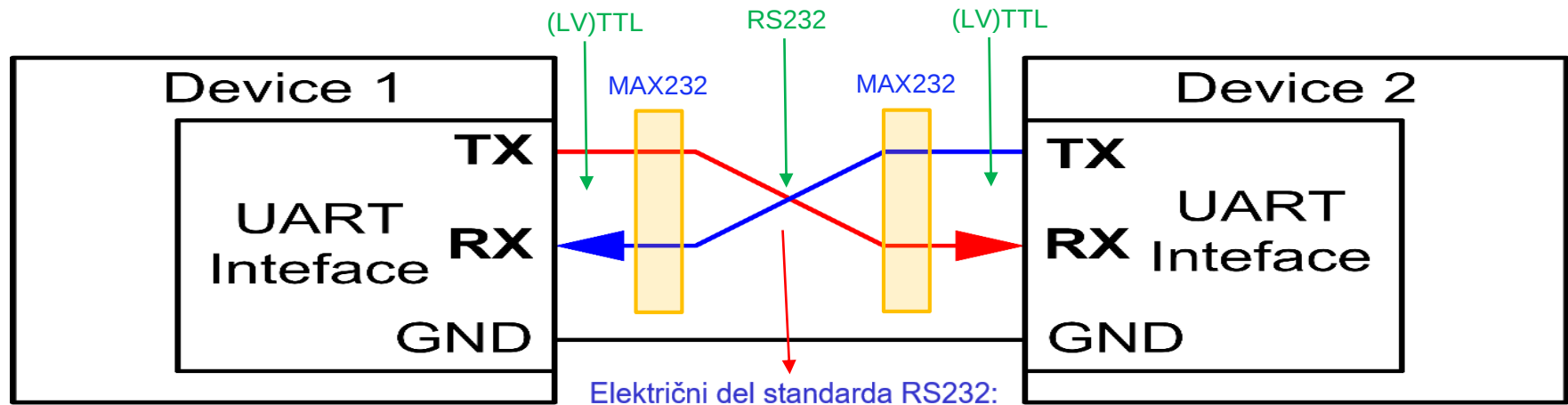
- Primer povezave na PC (običajno nima UART priključka):
 - FT232R pretvarja med UART RS232 in USB vmesnikom (PC)



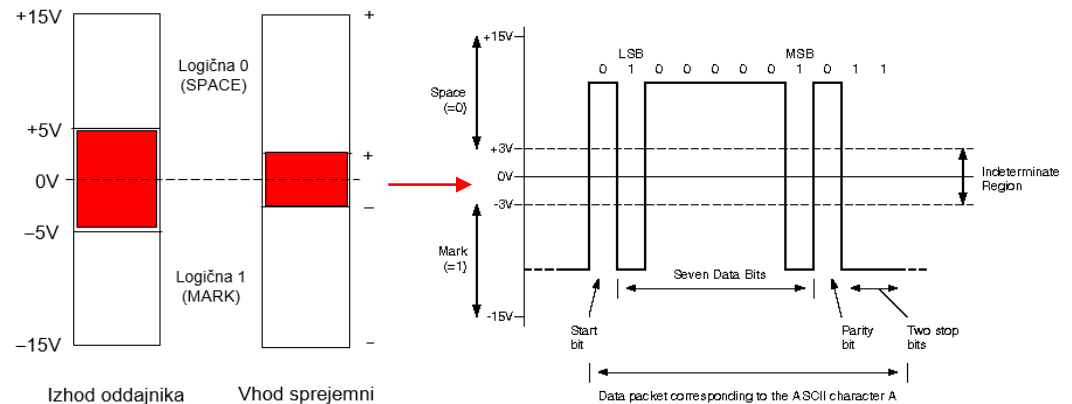
UART - električno

□ Napetostni nivoji so lahko (odvisno od pretvornika):

- (LV)TTL
- RS232 (pretvornik „MAX232“), 422, 485 (pretvornik „MAX485“)



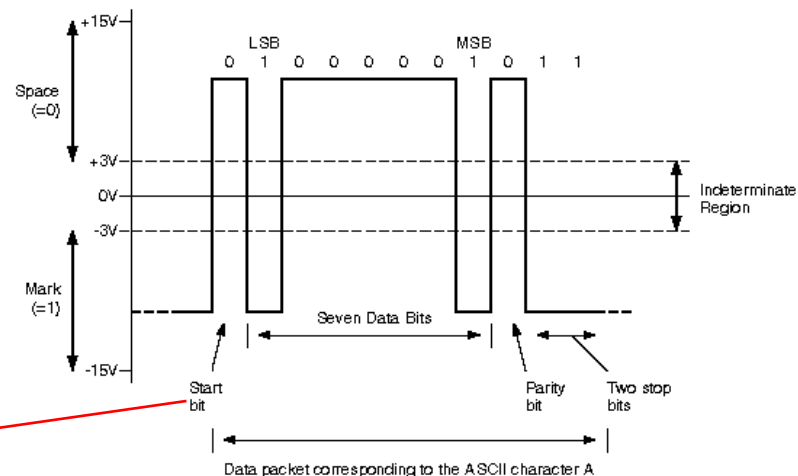
□ Napetostna in logična nivoja



UART - Standardi

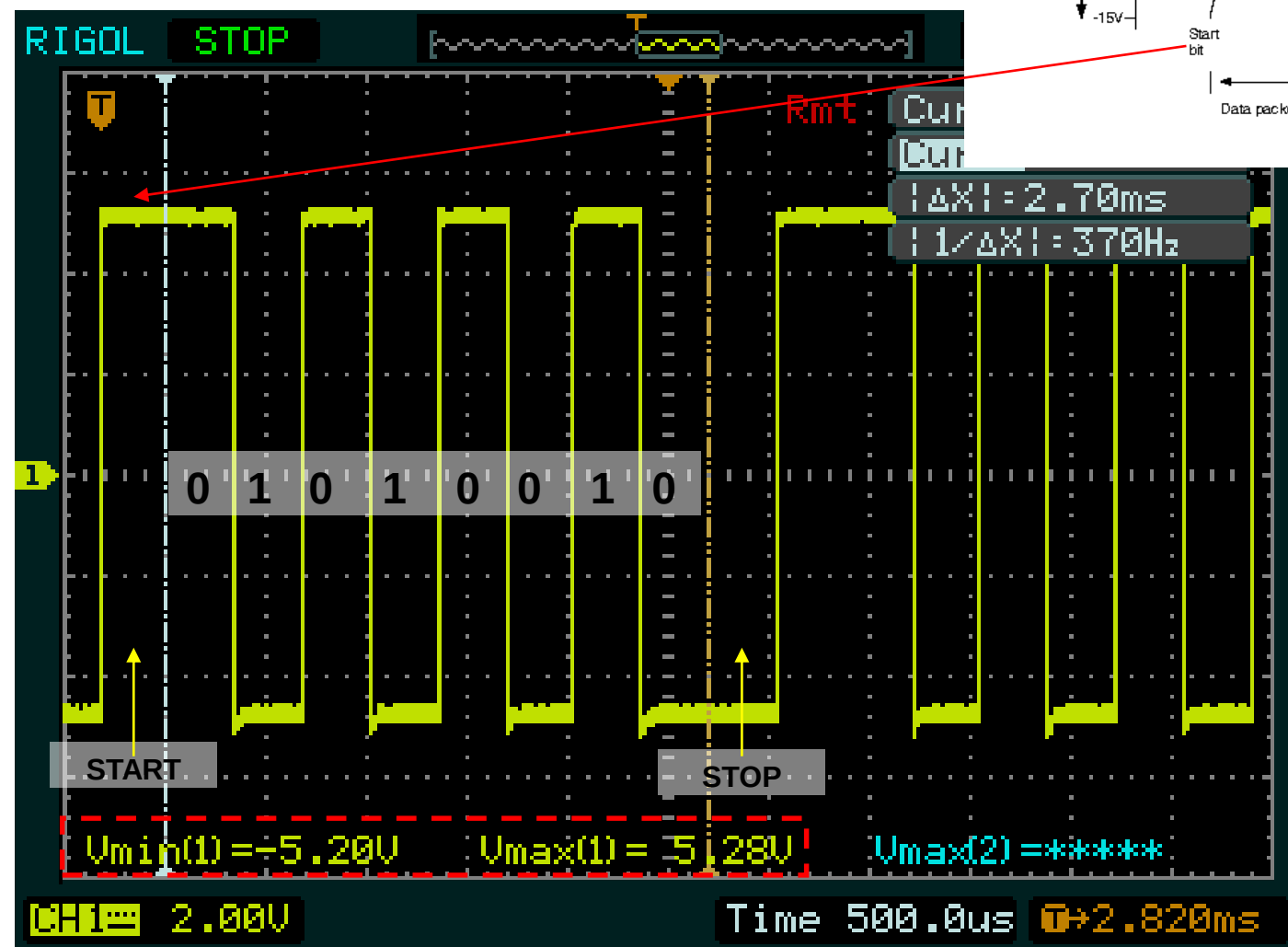
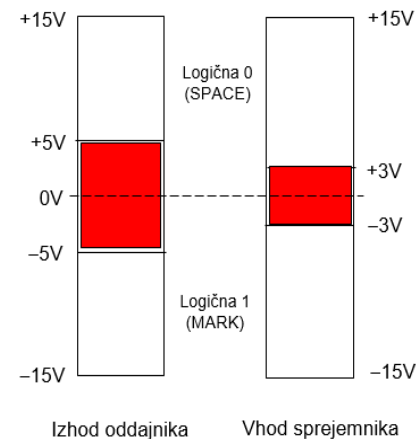
Standard	Voltage signal	Max distance	Max speed	Number of devices supported per port
RS-232	Single end (logic 1: +5 to +15V, logic 0: -5 to -15V)	100 feet	115Kbit/s	1 master, 1 receiver
RS-422	Differential (-6V to +6V)	4000 feet	10Mbit/s	1 master, 10 receivers
RS-485	Differential (-7V to +12V)	4000 feet	10Mbit/s	32 masters, 32 receivers

UART – osciloskop



Električni del standarda RS232:

□ Napetostna in logična nivoja

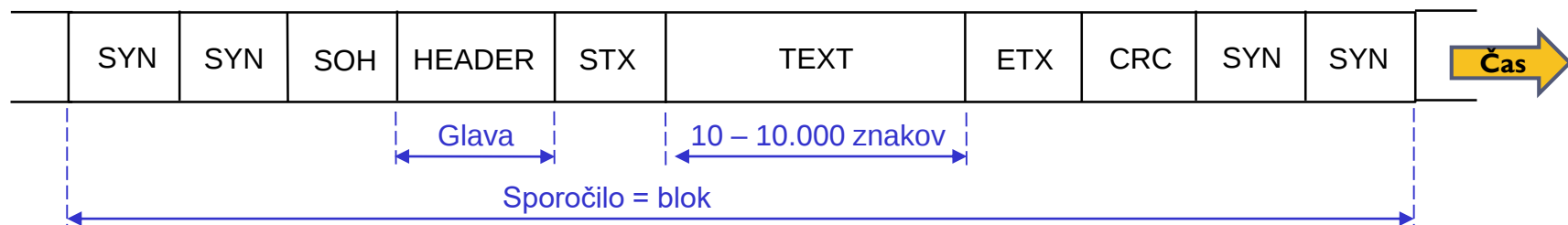


2. Sinhronski serijski prenos

- ❑ Razvit je bil za namensko komunikacijo med računalniki.
- ❑ Enota pošiljanja ni več posamezen znak, temveč **sporočilo ali blok**.
- ❑ Sinhronizacija poteka na nivoju celotnega sporočila ali bloka.
- ❑ Sinhronizacija se doseže na enega od dveh načinov (glede na hitrost):
 - ❑ počasnejše : dodana je povezava za **prenos urinega signala** (ang. Clock) – nižje hitrosti
 - ❑ hitre: brez „Clock“ povezave, oddajna in sprejemna ura se sinhronizirata iz prehodov v podatkovni povezavi
- ❑ Znotraj sporočila ali bloka ni presledkov, niz bitov je neprekinjen. Redundanca se v primerjavi z asinhronskim prenosom zmanjša in je samo nekaj procentov (%).
- ❑ Potrebno je pravilo za organizacijo informacij v bloku $\Rightarrow$ PROTOKOL
- ❑ V uporabi sta dve vrsti protokolov:
 - Znakovno ali bajtno orientirani protokoli (starejši)
 - Bitno orientirani protokoli (novejši)

□ Znakovno ali bajtno orientirani protokoli

- BSC (Binary Synchronous Communication) je IBM-ov znakovno orientirani protokol, 1967. Dolgo časa je bil najbolj razširjen protokol.
- Uporaba abecede z definiranimi **posebnimi kontrolnimi znaki**.
 - STX – Start of text
 - SYN znak - vzpostavitev sinhronizacije med oddajnikom in sprejemnikom
 - ...



- Paritetni bit za kontrolo pravilnosti ne zadošča več.
- Oddajnik pri oddaji iz bitov besedila izračuna CRC (Cyclic Redundancy Check) bite (eden ali dva bajta) in jih doda v sporočilo.
 - Če sprejemnik s pomočjo CRC bitov ugotovi, da je bilo sporočilo pravilno sprejeto, pošlje znak za potrditev ACK (ang. Acknowledgemnt).
 - Če pa ugotovi, da je bilo sporočilo napačno sprejeto, pošlje nazaj znak NACK (ang. Negative Acknowledgemnt) in oddajnik prenos ponovi.

❑ Bitno orientirani protokoli

- Sporočilo ali blok ne sestavljajo več znaki, temveč **poljubno zaporedje bitov**.
- Najbolj razširjena protokola sta
 - SDLC - Synchronous Data Link Control IBM 1970
 - HDLC – High-Level Data Link control ISO 1979, ki ima za osnovo SDLC
- HDLC je pravzaprav standardizirana skupina protokolov, vsem pa je skupno pravilo, da v zaporedju bitov ne sme biti več kot 5 zaporednih enic (1).
- Na ta način ločimo vsebino sporočila od bajtov, ki določajo začetek in konec sporočila – zaporedju šestih enic sledi ničla (Flag bajt $\equiv 0111110$).
- Oddajnik v sporočilu po vsakih petih zaporednih enkah vstavi 0 (ang. bit stuffing), sprejemnik pa te ničle izloča.
 - Pri NRZI kodiranju (0 – sprememba; 1 – brez spremembe) je običajno zagotovljena najmanj ena sprememba nivoja na vsakih 6 bitov $\Rightarrow$ sinhronizacija.
 - Novejši protokoli pa za uspešno sinhronizacijo večinoma uporabljajo kodiranje 8b/10b.

Primer STM32F4 Discovery

□ Sinhronski serijski prenos

Figure 307. USART example of synchronous transmission

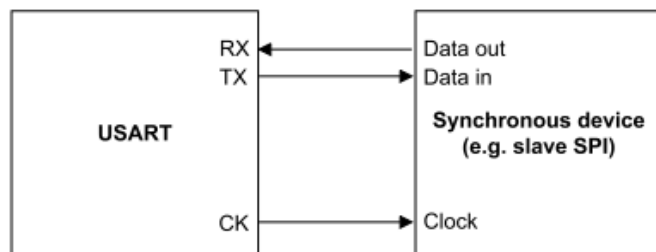


Figure 308. USART data clock timing diagram (M=0)

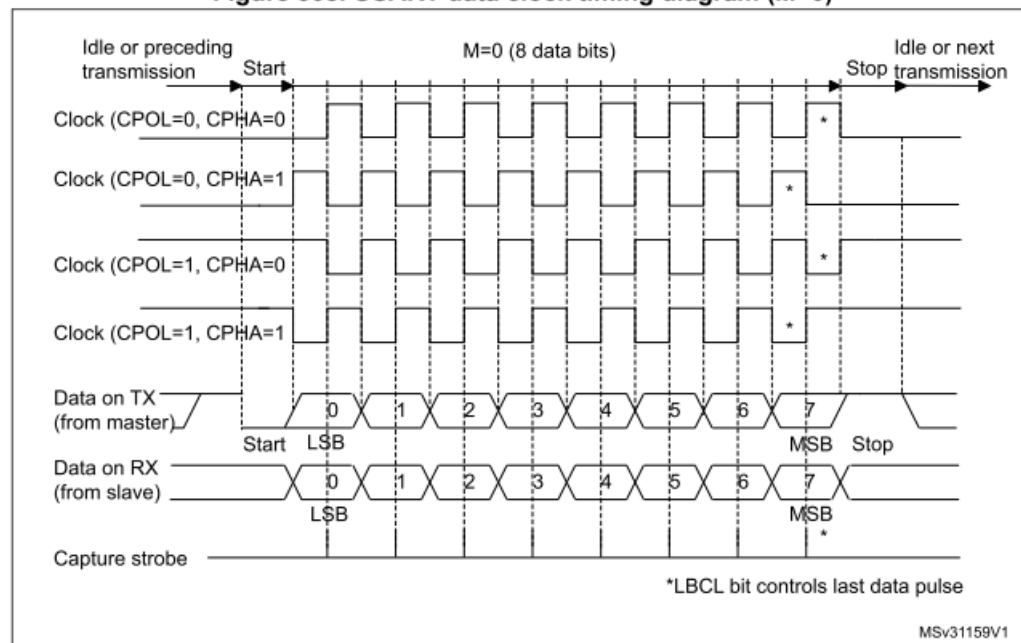
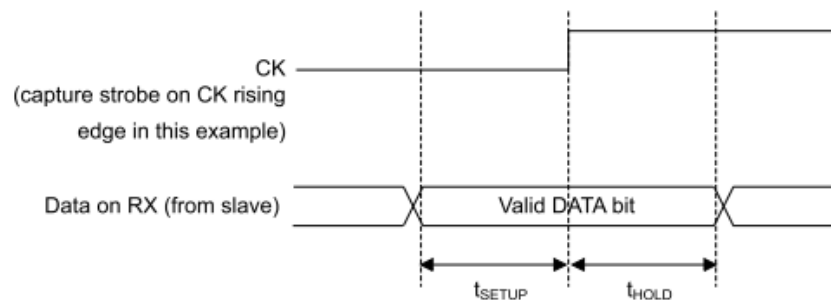
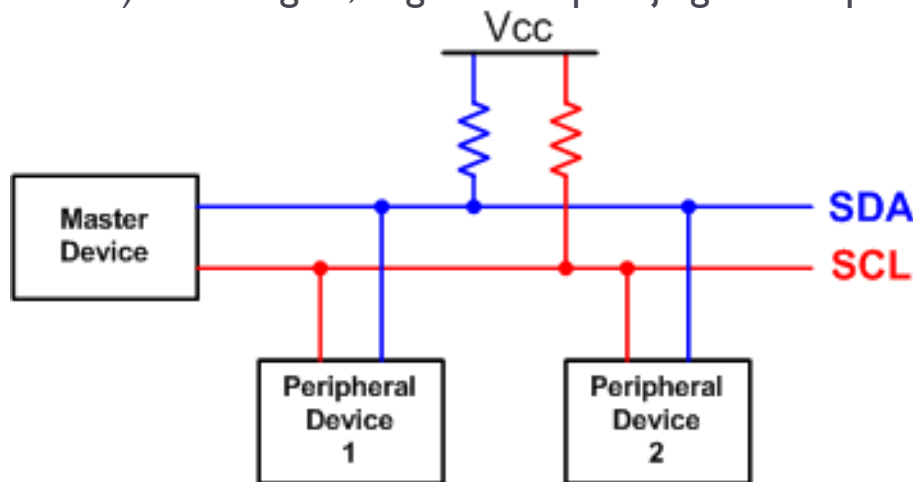


Figure 310. RX data setup/hold time

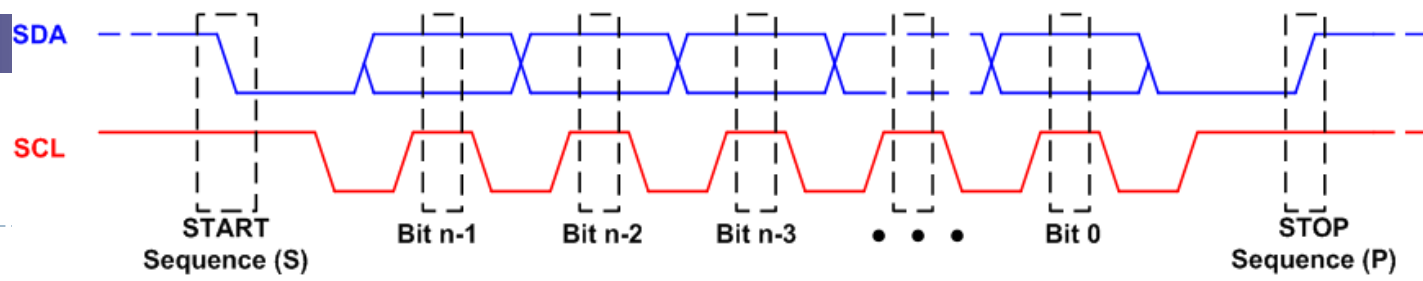


2.1 Vodilo I2C, IIC, I²C (Inter-Integrated Circuit)

- ❑ 1982 razvito kot interno vodilo za povezavo med čipi proizvajalca Philips (NXP).
- ❑ 2 liniji za prenos podatkov med dvema napravama (Master in Slave):
 - SDA (serial data) – linija za pošiljanje in sprejemanje podatkov
 - SCL (serial clock) - urin signal, ki ga vedno pošilja glavna naprava



- ❑ Povežemo glavno napravo (ang. master) in delovno napravo (ang. slave), ali več delovnih naprav na eno glavno napravo, ali več glavnih naprav lahko povežemo na eno ali več delovnih naprav.
- ❑ Podatki se vedno pošiljajo v **sporočilih** (ang. messages), ki so razdeljena v **okvirje** (ang. frames) podatkov in dodatne bite:
 - Naslovni okvir (ang. address frame) – binarni naslov delovne celice
 - Podatkovni okvir (ang. data frame) – podatki, ki se prenašajo.



▪ Dodatni biti:

- pogoj **start** – linija SDA preklopi iz 1 v 0 predno linija SCL preklopi iz 1 v 0
- pogoj **stop** - linija SDA preklopi iz 0 v 1 potem ko linija SCL preklopi iz 0 v 1
- **read/write** – en bit določa prenos iz 'master' v 'slave' napravo (0) ali 'master' zahteva podatek iz 'slave' naprave (1).
- **ACK/NACK** – vsak okvir sporočila ima bit 'acknowledge/noacknowledge'. Če je bil naslovni ali podatkovni okvir uspešno prejet je pošiljatelju vrnjen bit ACK, sicer NACK.

❑ Sporočilo

start	Naslovni okvir	read / write	ACK / NACK	Podatkovni okvir	ACK / NACK	Podatkovni okvir	ACK / NACK	stop
-------	----------------	--------------	------------	------------------	------------	------------------	------------	------

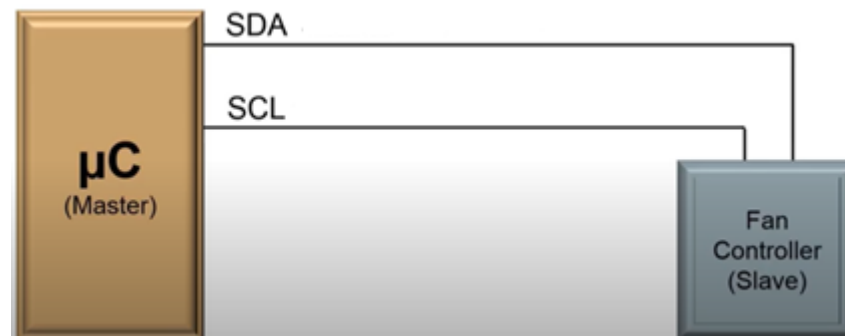


- ❑ Sinhronski serijski prenos, dvosmerni izmenični (ang. half duplex)
- ❑ Hitrost prenosa je podana s frekvenco ure v kHz, ali tudi že v MHz.
- ❑ Uporaba: krmilnik s senzorji, AD pretvorniki, internet stvari (IoT), ...

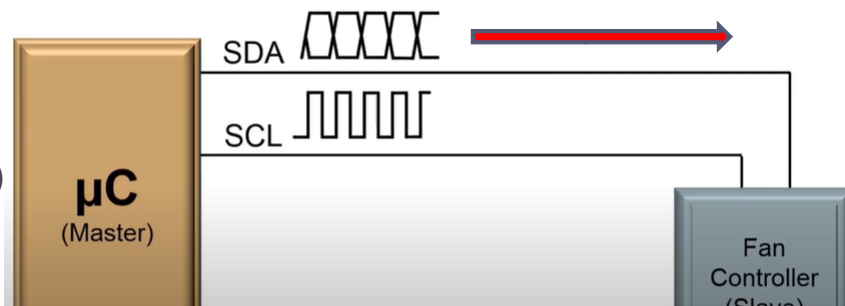
- ❑ Video: Microchip I2C: <https://www.youtube.com/watch?v=qTLRRg6Mee0>

❑ Povezava dveh naprav:

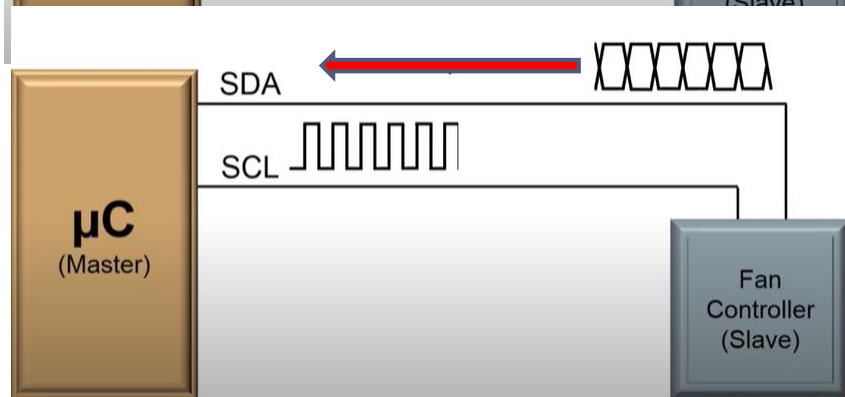
- Mikrokrmilnik (Master)
- Krmiljenje ventilatorja (Slave)



- Master pošilja urin signal (SCL)
- Master zapisuje na napravo Slave (SDA)

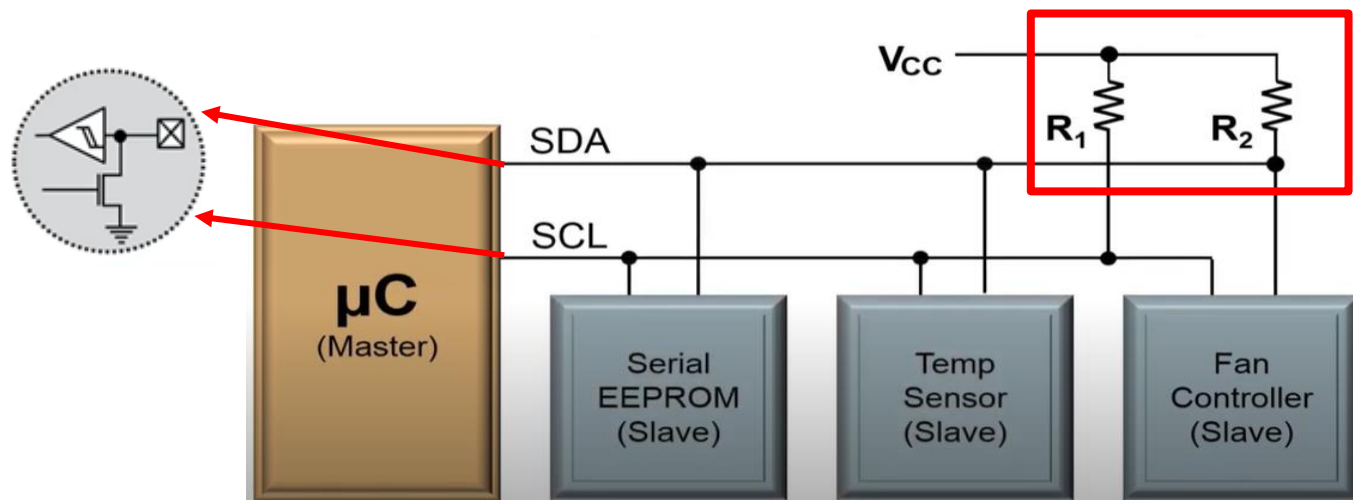


- Master pošilja urin signal (SCL)
- Master **bere iz** naprave Slave



□ Primer povezave mikrokrmilnika (Master) s tremi V/I napravami (Slave)

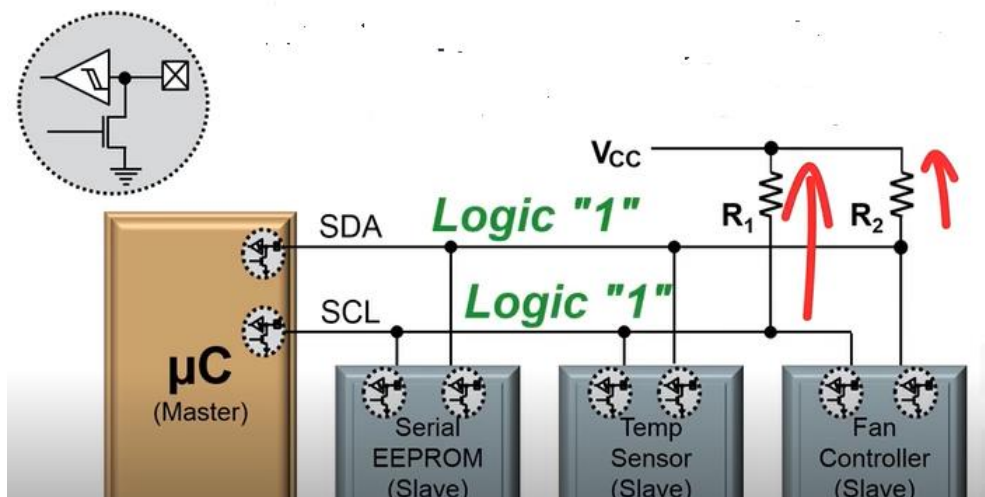
- Vsaka naprava Slave ima svoj naslov
- Za pravilno delovanje sta pri povezavi naprav na vodilo I2C upora R_1 in R_2 ('pull-up').



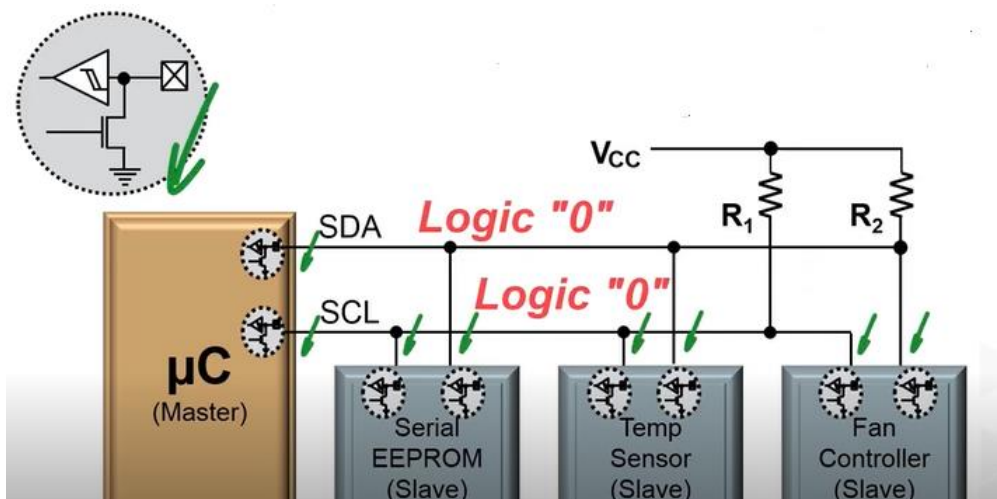
- Protokol I2C deluje na predpostavki, da sta liniji SDA in SCL 'open drain' ali 'open collector'
 - Katerakoli naprava zagotovi nizek nivo (low) na linijah, ne more pa visokega (high).
 - Vsaka linija ima zato dodan upor, da privzeto ohranja visok nivo.
 - To omogoča, da ne pride do kratkega stika, če ena naprava poskuša dvigniti linijo na visok nivo, druga pa na nizek nivo.

Logični nivoji:

Logična 1



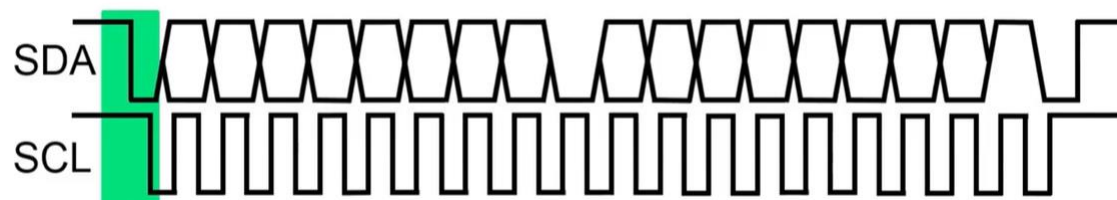
Logična 0



□ I2C signal

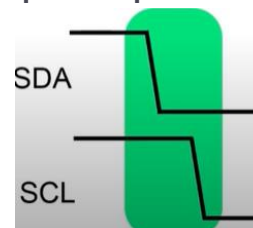


□ START

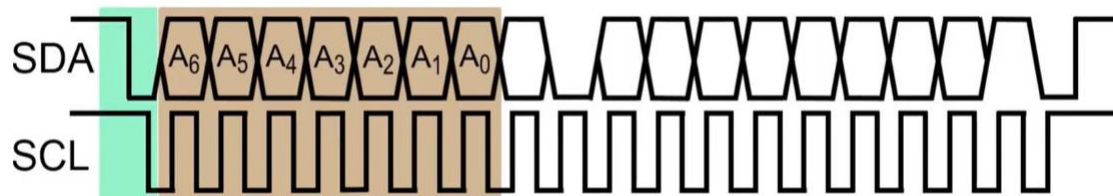


- Linija SDA preklopi iz 1 v 0 predno linija SCL preklopi iz 1 v 0

Idle – SDA = SCL - High
Master SDA – High v Low
Master SCL – drži High

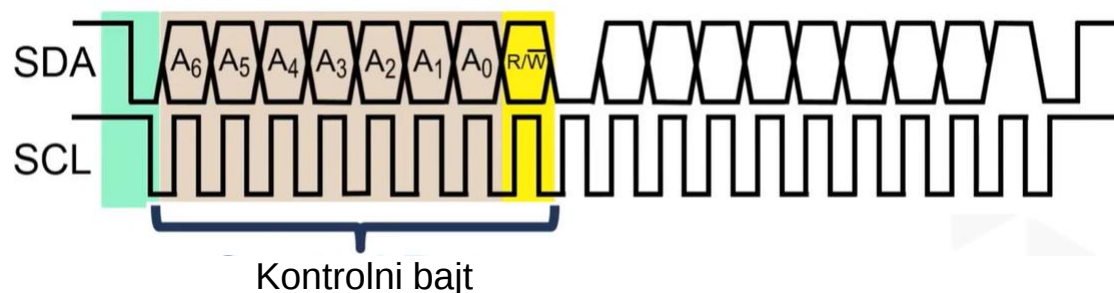


□ Naslov naprave Slave je 7-biten (najpogostejša dolžina)



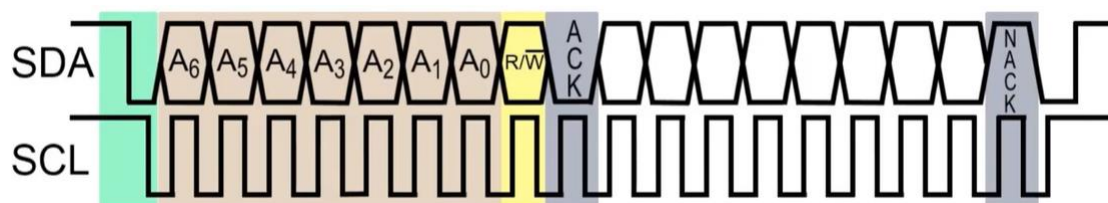
□ Branje/pisanje – $R/\bar{W}$ (kontrolni bajt: $A_6 \dots A_0 R/\bar{W}$)

- $R/\bar{W} = \text{High}$ (Master zahteva branje podatkov iz Slave)
- $R/\bar{W} = \text{Low}$ (Master bo pisal na Slave)

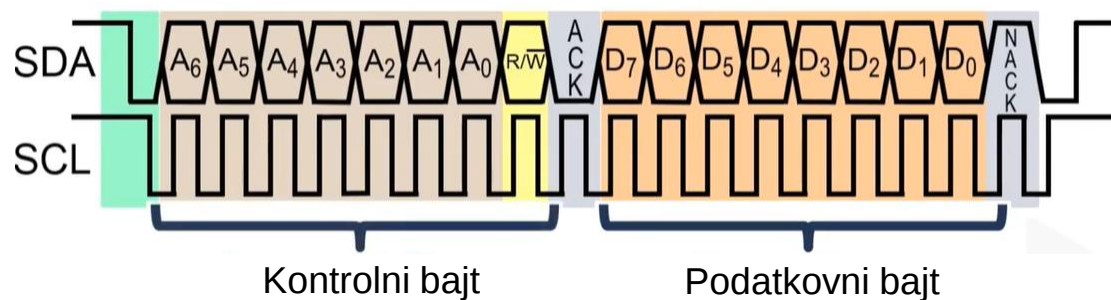


- kontrolni bajt: $A_6 \dots A_0 R/\bar{W}$

□ Potrditev (ang. Acknowledge) – pojavi se na vsakem 9 urinem ciklu



Podatkovni bajt ($D_7 \dots D_0$)



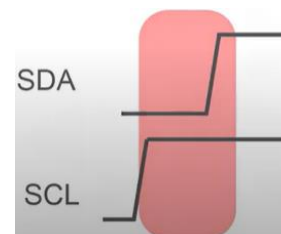
STOP



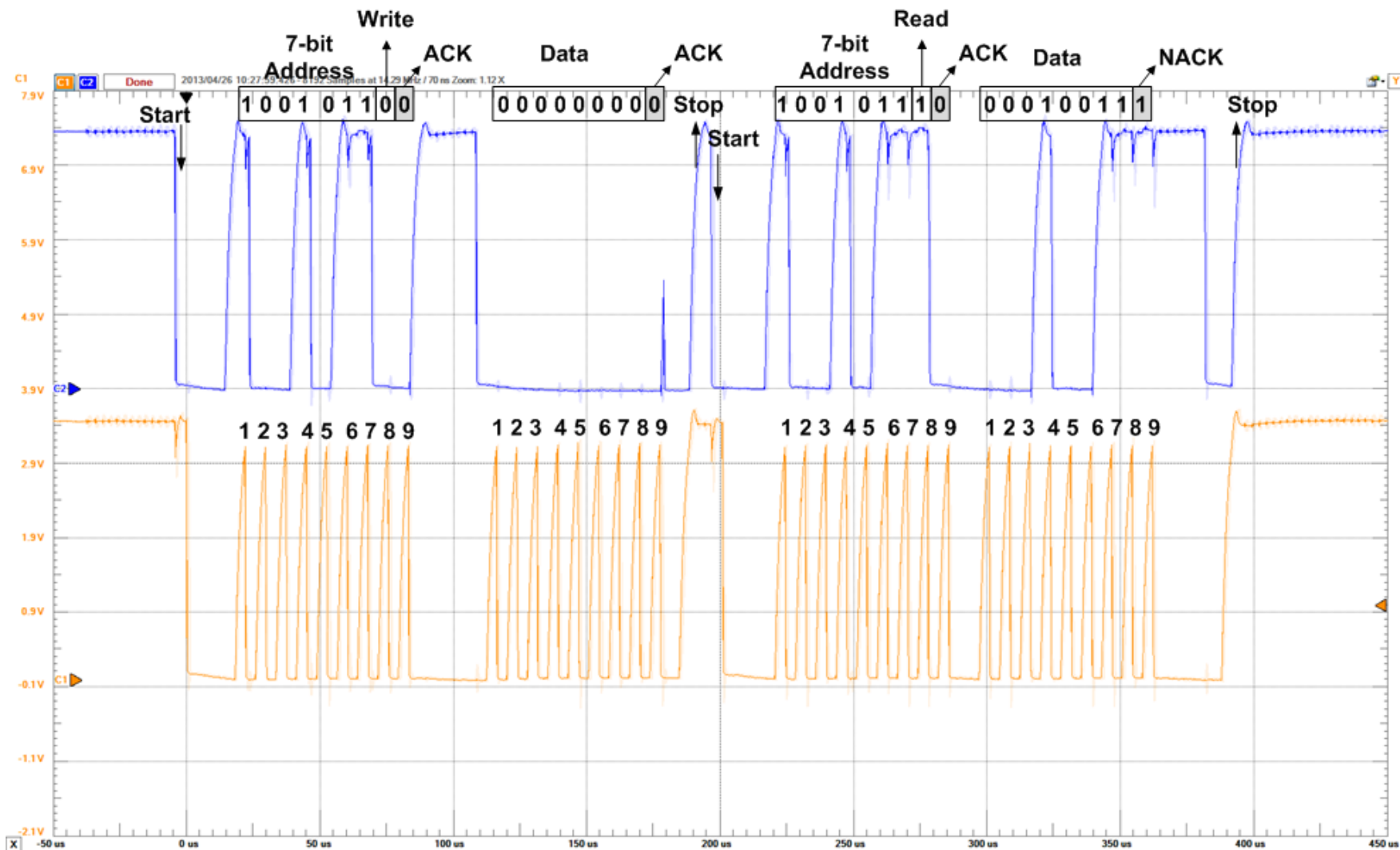
- Linija SDA preklopi iz 0 v 1 potem ko linija SCL preklopi iz 0 v 1

Master SDA – Low v High

Master SCL – drži High

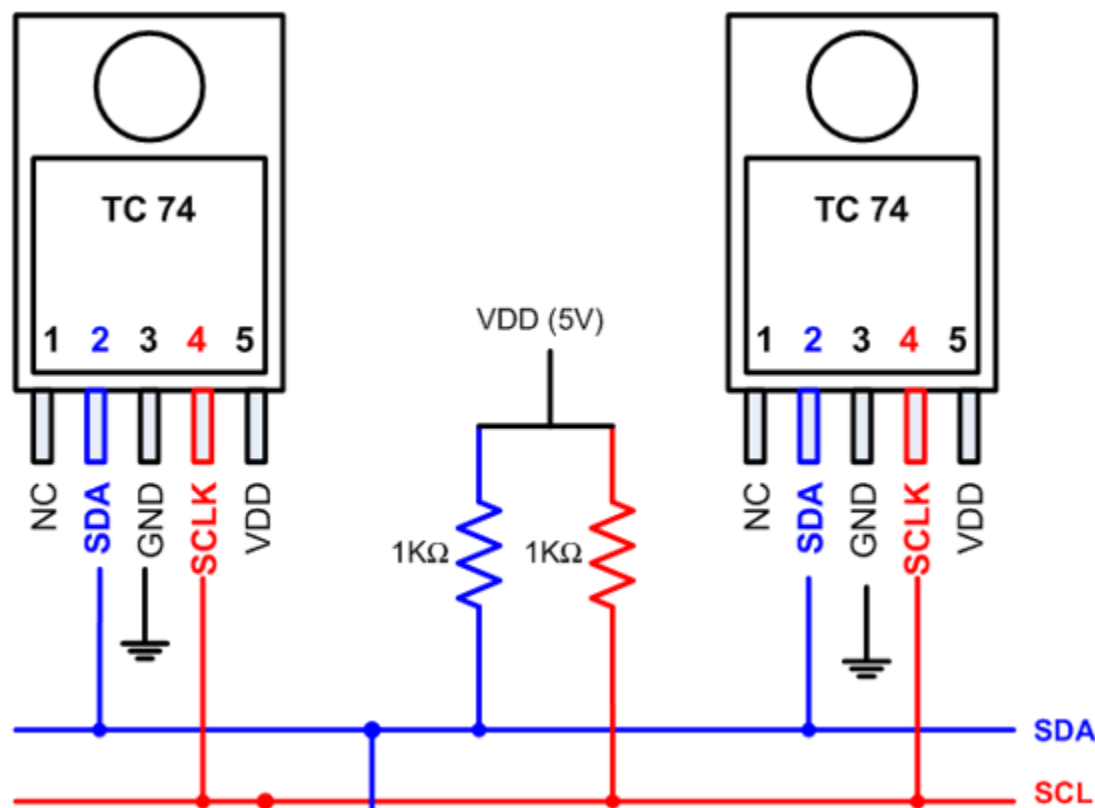


I2C – električno (osciloskop)



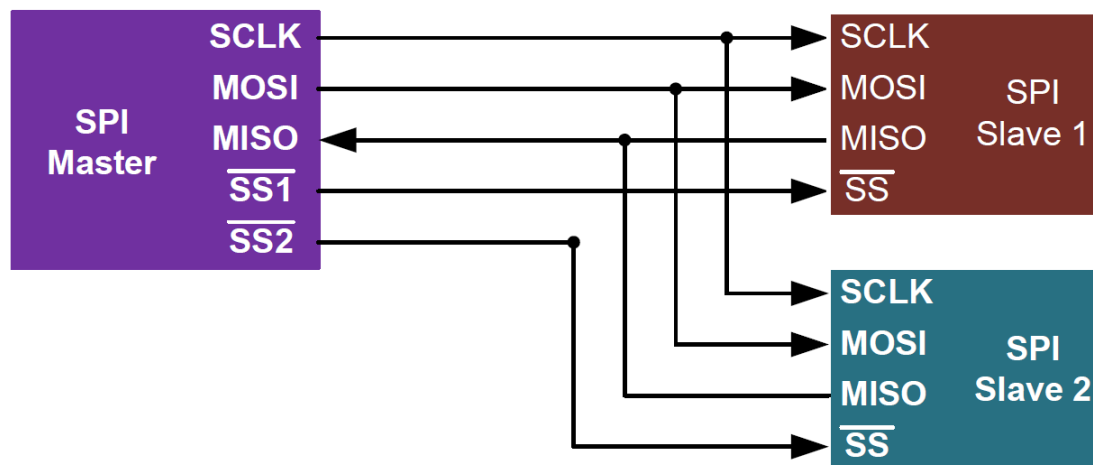
Primer I2C povezave

□ Digitalno temperaturno tipalo



2.2 Vodilo SPI (Serial Peripheral Interface)

- ❑ 1985: pojavi se potreba po višjih hitrostih, kot jih ponuja I2C
- ❑ SPI vmesnik ima 3 ali 4 linije, kjer sta 2 za prenos podatkov:
 - MISO (Master In Slave Out) – naprava Master sprejme podatke od naprave slave
 - MOSI (Master Out Slave In) - naprava master pošlje podatke napravi slave
 - SCK (Serial Clock) – urin signal, ki ga pošlje naprava master
 - $\overline{SS}$ - Slave (Chip) select (omogoča izbiro naprave slave, kadar jih je več povezanih na master)

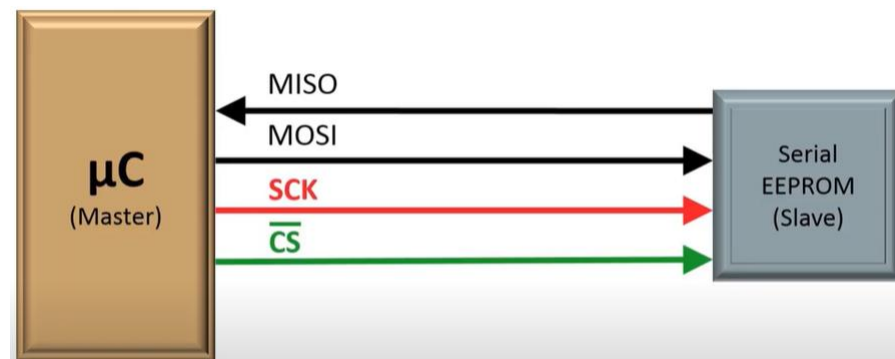


- ❑ Prenos podatkov je omogočen z načinom povezave točka-v-točko (ang. point-to-point)
 - Naprava master pošilja napravi slave bit za bitom po liniji MOSI, običajno najprej MSB bit.
 - Naprava slave pošilja napravi master bit za bitom po liniji MISO, običajno najprej LSB bit.

- ❑ Prenos podatkov poteka istočasno v obe smeri (ang. full-duplex)
- ❑ Hitrost prenosa podatkov je odvisna of frekvence urinega signala
 - SPI najpogosteje omogoča delovanje pri frekvenci 20 MHz.
 - Dual SPI in Quad SPI omogoča delovanje do 144 MHz.
 - Dual SPI - pošiljanje podatkov je izvedeno kot izmenični prenos (ang. half duplex) dveh bitov tako, da je MOSI definiran kot IO0, in MISO pa IO1.
 - QUAD SPI – dodani sta še dve liniji in pošiljanje podatkov je izvedeno kot izmenični prenos (ang. half duplex) štirih bitov preko linij IO0, IO1, IO2, IO3.
- ❑ Uporaba: zabavna elektronika, pomnilniki, RFID, avtomobilska in letalska industrija, ...
- ❑ Video: Microchip: <https://www.youtube.com/watch?v=NyxQkGXbG6I>

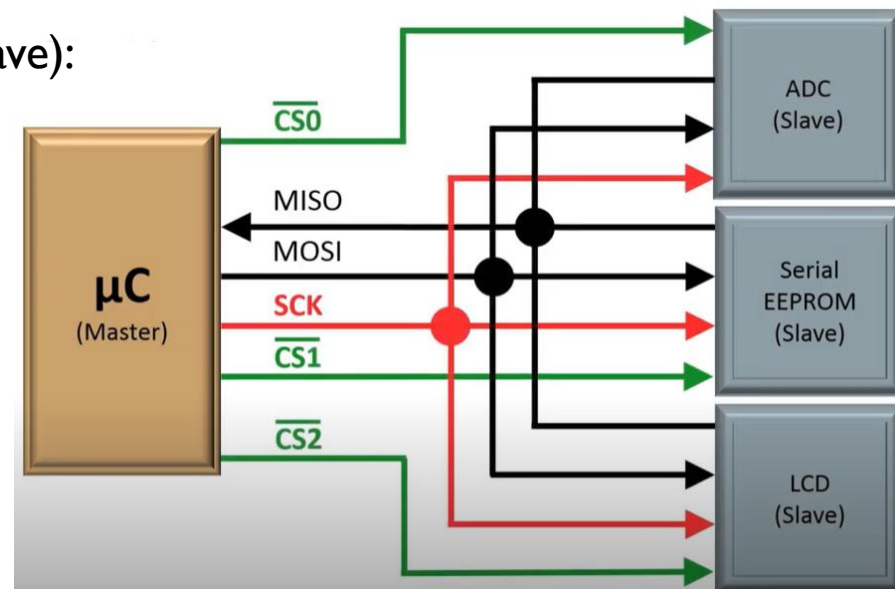
□ Vodilo SPI je vmesnik med dvema napravama, ki ima 4 povezave

- Mikrokrmilnik je glavna naprava (Master)
- EEPROM je delovna naprava (Slave)



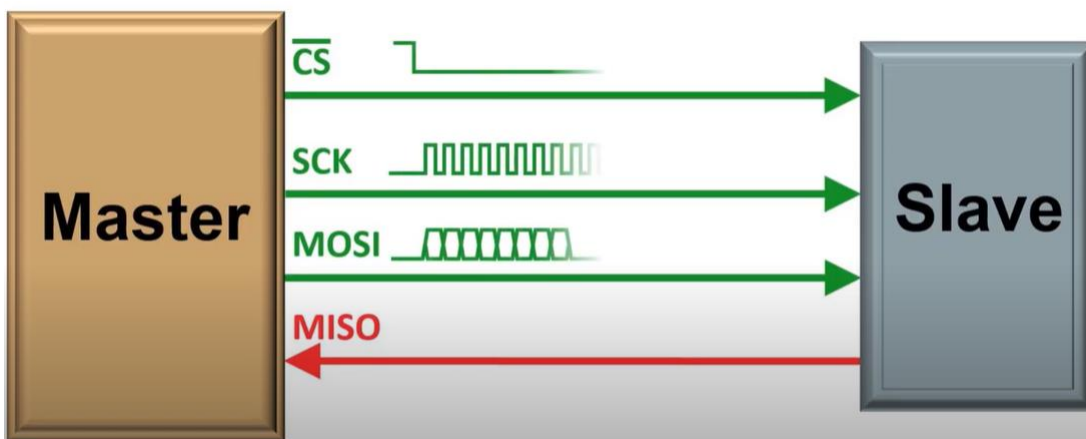
□ 4 naprave na vodilu SPI (Master + 3x Slave):

- Mikrokrmilnik (Master)
- ADC (Slave)
- EEPROM (Slave)
- LCD (Slave)
- Vsaka naprava Slave ima svoj signal $\overline{CS}$

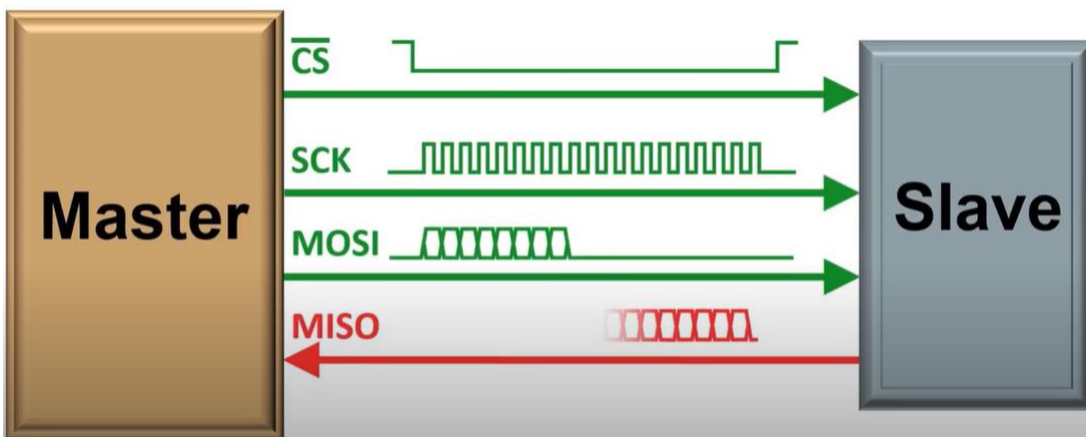


□ Signali SPI pri prenosu podatkov

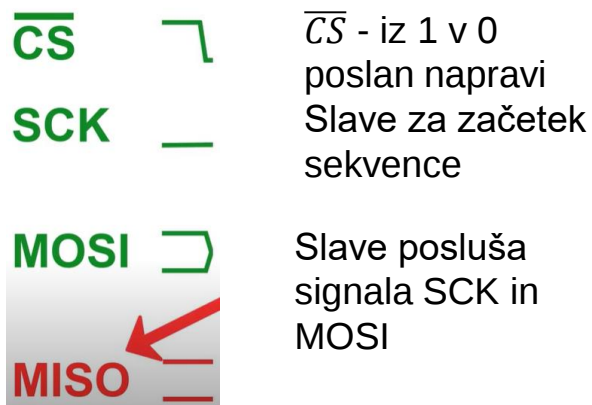
- Master generira signale $\overline{CS}$, SCK, MOSI



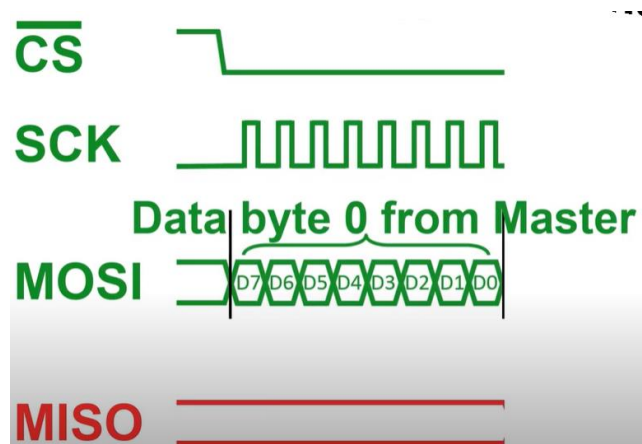
- Slave generira signal MISO



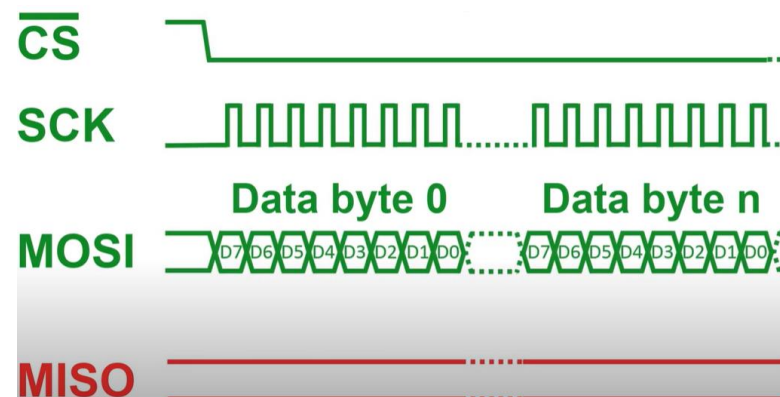
□ Serijski prenos podatkov iz naprave Master v Slave



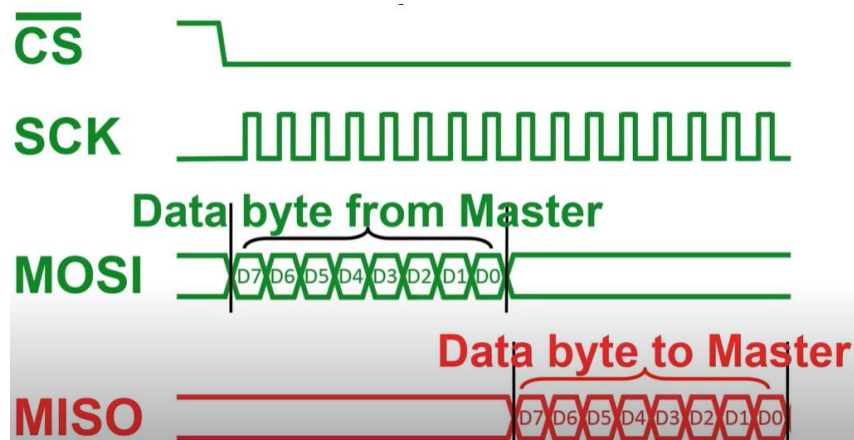
- Master prenese **bajt 0** po liniji MOSI



oz n bajtov se prenese

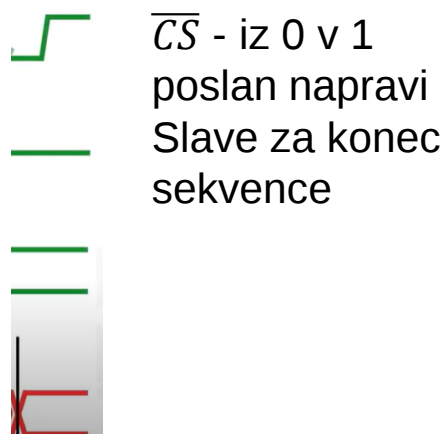


- Prenos podatkov Master to Slave in Slave to Master



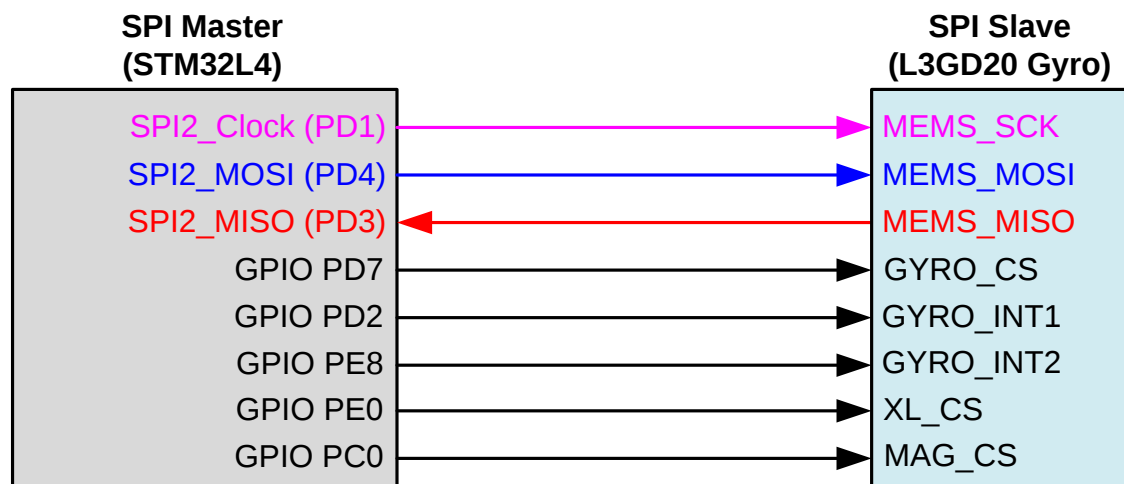
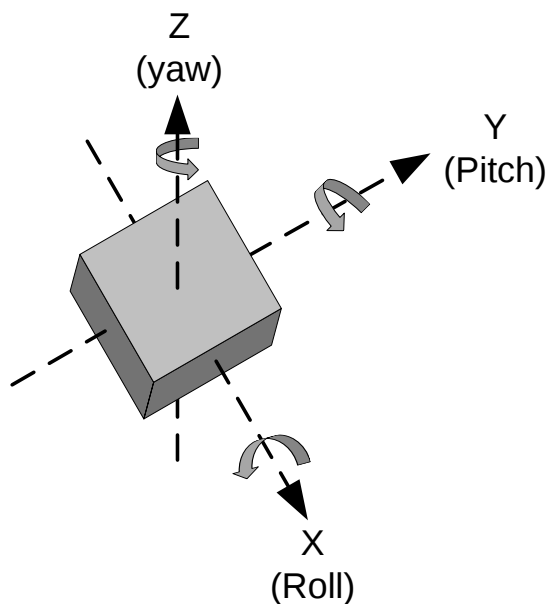
- Konec prenosa:

Chip Select gre iz Low v High



Primer SPI povezave

□ Primer žiroskopa (L3GD20)



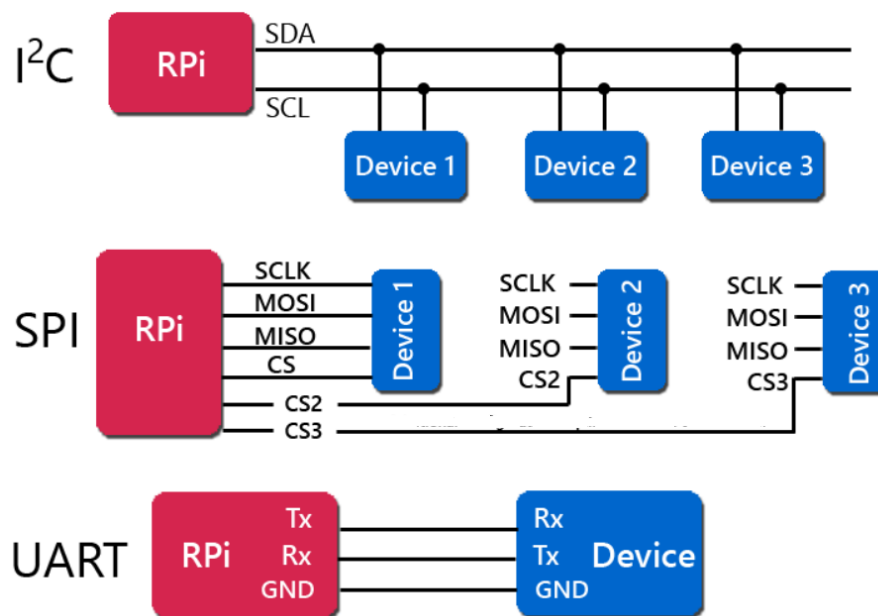
Primerjava SPI vs I²C

- ❑ Oba sinhronska protokola za lokalne komunikacije
- ❑ SPI (Motorola), I2C (Philips – danes NXP)

	SPI	I2C
Prednosti	• Hitrejši (20Mbps+) • Full duplex • „Low power“ • DualSPI, QuadSPI	• Enostavnost (manj povezav, standard za 2 povezavi) • Dodajanje • Multi-Master

Primer: Raspberry Pi $\leftrightarrow$ I²C, SPI, UART

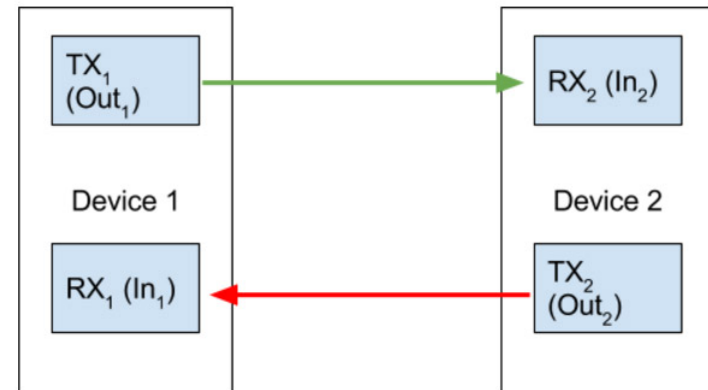
- Diagram povezav
R_{Pi} = Raspberry Pi



- UART - preprost; majhna hitrost; ura ni potrebna; omejena na eno napravo, priključeno na R_{Pi}.
- I²C - hitrejši od UART, vendar ne tako hitro kot SPI; lažje povezovanje številnih naprav; R_{Pi} poganja uro, tako da ni težav s sinhronizacijo.
- SPI - najhitrejši od treh; R_{Pi} poganja uro, tako da ni težav s sinhronizacijo; praktična omejitev števila naprav na R_{Pi}.
- <https://www.mbtechworks.com/hardware/raspberry-pi-UART-SPI-I2C.html>

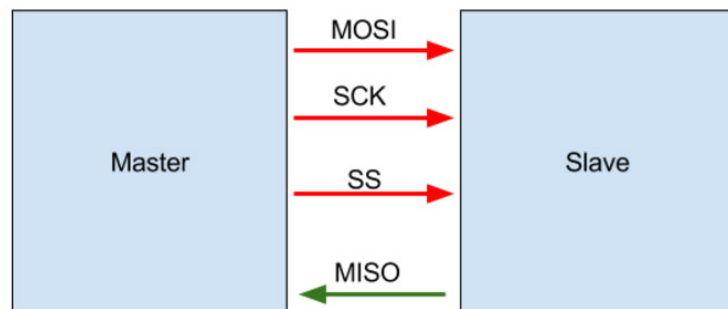
Primer: Arduino $\leftrightarrow$ UART, SPI, I2C

❑ Asinhronski serijski prenos: **UART**

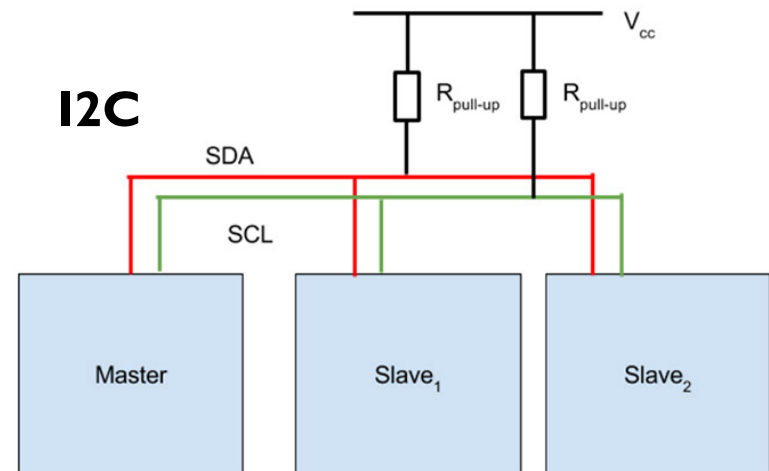


❑ Sinhronski serijski prenos:

SPI



I2C



❑ <https://www.deviceplus.com/arduino/arduino-communication-protocols-tutorial/>