



Vhodno-izhodne naprave (VIN)

Predavanja

4. Serijski prenos podatkov

Robert Rozman

rozman@fri.uni-lj.si

1. Domača naloga – V/I naprave

Poročilo

Zahteve za pripravo poročila

Pregled objav (spletne objave, članki, drugi dokumenti), vsaj 3 objave uporabite za poročilo. Poročilo (okvirno do 5 strani):

- Naslov s podatki: VIN Predavanja – 1. DN (2022/23 Tema:..., Avtorji: ..., Datum: ...)
- **Grafična skica (»poster«)**, povzetek vsebine (v smislu posterskih predstavitev, kot jih vidite tudi na FRI po stenah)
- Predstavitev naprave: opis, slika in razvoj
- Arhitektura in tehnologije
- Opis delovanja
- Uporaba s kratkim opisom delovanja
- **Povezava (kako se programsko povežemo z napravo)**
- Zaključek
- Literatura

2.4.2023

Oddaja poročila: ~~25.3.2023~~

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

Navajanje virov (povzeto po pripravi diplomskega dela):

[1] R. D. Semmel, D. P. Silberg, "Extended entity-relationship model for automatic query formulation", Telematics and Informatics, _st. 10, zv. 3, str. 301-317, 1993.

[2] (2005) IEEE Information for authors. Dostopno na:

<http://www.ieee.org/portal/cms/docs/pubs/transactions/auinfo03.pdf>

Sklicevanje na vir se v predstavitvi označi z zaporedno številko vira v oglatem oklepaju [1].

Predstavitev

4.4.2023

Predstavitev (5 min): ~~od 28.3.2023 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

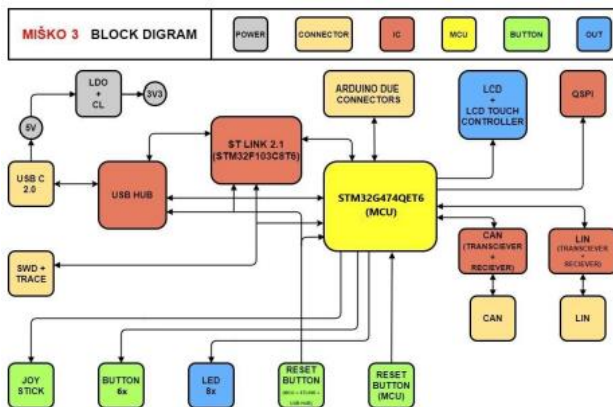
Za izdelavo predstavitve smiselno uporabite dele poročila. Predstavitev je lahko pripravljena po možnosti v Microsoft PowerPoint ali vsaj v OneNote zvezku (Prostor za sodelovanje ali vaš osebni zvezek). Vsebino lahko objavite tudi v trajnejši obliki na spletni strani ali portalih (npr. blog, Github). Vsebina naj bo v primerni obliki za enostavno vključitev v okvir predavanj, morda jo bomo uporabili tudi v enem od terminov proti koncu semestra.

Pomembna izhodišča za vaše delo

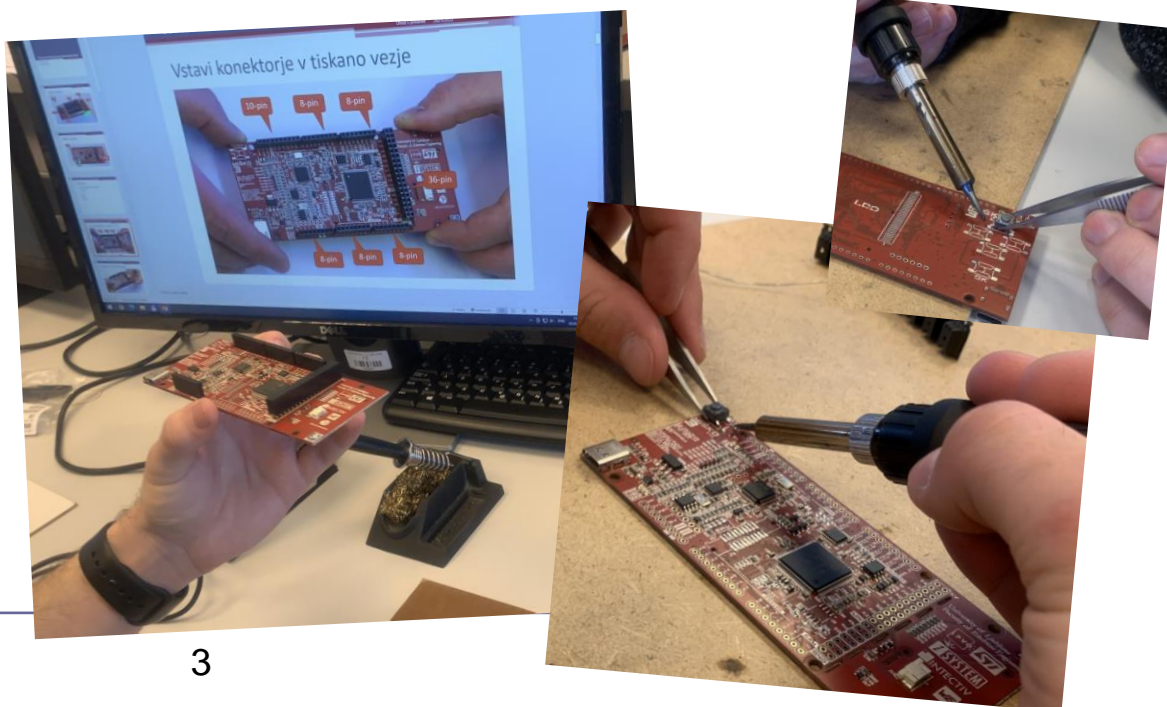
Potrudite se oblikovati vse vsebine **pravopisno in oblikovno korektno**. Pri pripravi sodelujte vsi in pridobivajte veččine pisanja, sodelovanja in predstavitev. Pripravite tudi predstavitev, kjer lahko uporabite grafično skico, ki ste jo naredili. Potrudite se, da bo **predstavitev zanimiva** in karseda **informativna za vaše kolege**.

Tehnične specifikacije

- Mikrokontrolnik STM32G474QET
 - 96kB RAM
 - 512 kB Flash pomnilnika
 - napajanje 3,3 V
 - maksimalna frekvenca ure 170 MHz
- ST-LINK V2.1 za programiranje, razhroščevanje in UART komunikacijo s mikrokontrolnikom
- USB-C s podporo USB 2.0 za USB komunikacijo z mikrokontrolnikom in napajanje sistema
- USB HUB za povezavo na ST-LINK in USB vmesnik mikrokontrolnika
- Barvni LCD zaslon 320x240 s ILI9341 krmilnikom in 16-bitnim paralelnim vodilom prek FMC vmesnika
- Rezistivna folija za detekcijo dotika s XPT2046 krmilnikom na SPI vodilu
- 8 LED in 6 tipk, 2D joystick s tipko
- CAN in LIN vmesnik
- QSPI za razširitev pomnilnika
- SWD trace vmesnik za napredno razhroščevanje
- Arduino DUE kompatibilna dimenzija tiskanega vezja, razporeditev letvic in signalov



Slika 1: Bločni diagram razvojnega Sistema MIŠKO 3



DN1: Domača naloga – Primeri tem OneNote zvezek – dosedanja projekti, ideje za naprej, spisec opreme:

The screenshot shows a OneNote application interface. At the top, there is a navigation bar with several tabs: "VIN-VSP 2022-23 zvezek", "Uporaba knjižnice vsebine", "Tabla 2022-23", "Predavanja 2021-22", "VIN Projekt - Ideje", "VIN Projekti - Done", "Moduli, Tipala", "LAB Oprema", "Docs", "STM32", and "Cube IDE". The "VIN Projekti - Done" tab is currently selected. Below the navigation bar, the main content area is divided into two panes. The left pane displays a page titled "DHT22 Hum&Temp sensor" with a date and time stamp: "sreda, 31. marec 2021 18:51". Below this, there is a large black box with the text "How to use DHT22 with STM32 UPDATE" in yellow and white. Underneath, there is a paragraph of text explaining a method for using DHT sensors with STM32, followed by a URL: https://controllerstech.com/wp-content/uploads/2020/06/DHT_11_22_DWT.zip. Below this, there is another URL: [Z naslova <https://controllerstech.com/temperature-measurement-using-dht22-in-stm32/>](https://controllerstech.com/temperature-measurement-using-dht22-in-stm32/). The right pane shows a list of project ideas and components under the heading "2021/22". The list includes: "Robotska roka", "WIFI-controlled car", "Simulacija dvigala in I2C komunikacije", "Air mouse", "Logični analizator" (with sub-item "Logic analyzer STM32 DISCOVERY"), "Brezžična komunikacija z ultrazvočnimi senzorjema", "Protiploplavna vrata", "Anemometer", "Mushy - Arduino za gojenje gob", "Dovajanje odvečne solarne energije vodnemu", "Neimenovana stran", "Knez Ambilight Rpi+STM32,Neopixel" (with sub-items "Raspberry Pi 3B", "SPI", "STM32", "Neopixel", "Zaključek"), "WAV player", "Miško3 Maze Game+Case", "Miško 3 Game Console", "Miško3 PONG Game", "Miško3 2D Igra", "HOME MONITORING SYSTEM", "Mobilni robot", "Arduino sonar z upravljanjem preko telefona", "Sistem za zaznavanje vsiljivcev - Arduino", "RFID "odklepanje" vrat", "Arduino Krmiljena RGB dioda+lcd", "Stroj za zvijanje žic", "Okretnica za maketo", "VODNA ZALIVALK", "Arduino vremenska postaja", and "ARDUINO WORDLE". To the right of this list, there is a vertical green bar with the letter "R". Further to the right, there is a green sidebar titled "LAB Oprema" containing a list of equipment items: "Preberi me", "Tipala za delo" (with sub-items "HC SR04 UZ Senzor", "IR tipalo", "Time-of-flight", "37 in 1 sensor kit for Arduino", "KY-005 in KY-022 IR TX/RX", "KY-015 DHT11 Hum&Temp sensor", "KY-024 LINEAR MAGNETIC HALL MODULE", "KY-028 Digital temperature module", "KY-032 INFRARED OBSTACLE AVOIDANCE SENSO", "KY-038 Microphone Sound Detection Sensor Mod", "Rain drop Sensor Module", "Soil Moisture sensor", "DHT22 Hum&Temp sensor"), "Tranzistor BC 337", "Sound Sensor Detection Module LM393 Chip Electret", "PIR Napion Senzor", "MPXV10GC7U Senzor pritiska", "LPS35HW tipalo pritiska", "LCD zasloni" (with sub-items "SSD1306 OLED Controller", "Nokia5110 graf. LCD", "lcd 2x16 pvc160203P"), and "Izhodne naprave".

Vsebina

1. Asinhronski serijski prenos
 - **UART** (Universal Asynchronous Receiver/Transmitter)
 - (RS232, RS422, RS485)
 - **(CANBus)**
2. Sinhronski serijski prenos
 - **USART** (Universal Synchronous/Asynchronous Receiver/Transmitter)
 - **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

Primer STM32F4:

1. Asinhronski serijski prenos

- **U(S)ART** (Universal (A)synchronous Receiver/Transmitter)

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

- **CAN 1,2**

2. Sinhronski serijski prenos

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

- **I2C1, I2C2, I2C3**

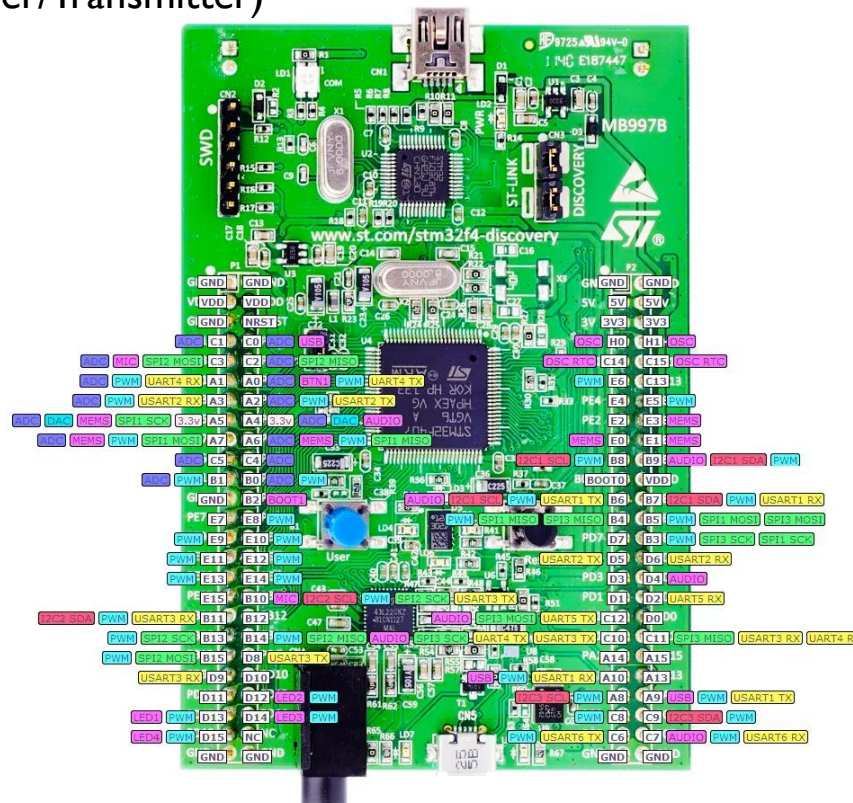
- **SPI** (Serial Peripheral Interface)

- **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

<https://microcontrollerslab.com/stm32f4-discovery-board-pinout-features-examples/>

Uvod

Primer STM32H7:

1. Asinhronski serijski prenos

- **U(S)ART** (Universal Asynchronous Receiver/Transmitter)
 - U(S)ART 1,2,3,4,5,6,7,8
- **FDCAN 1,2**

2. Sinhronski serijski prenos

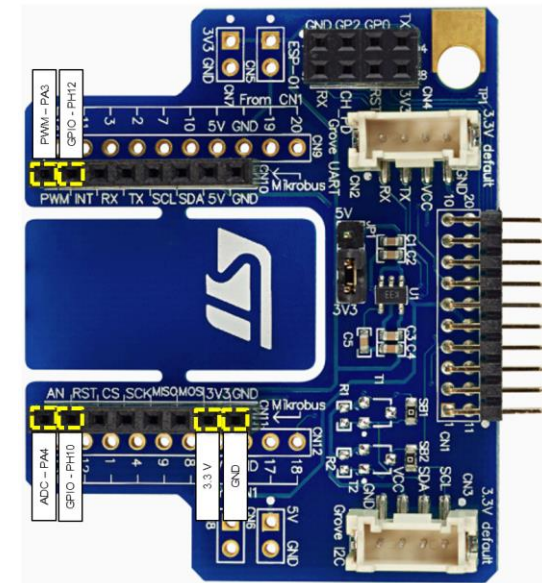
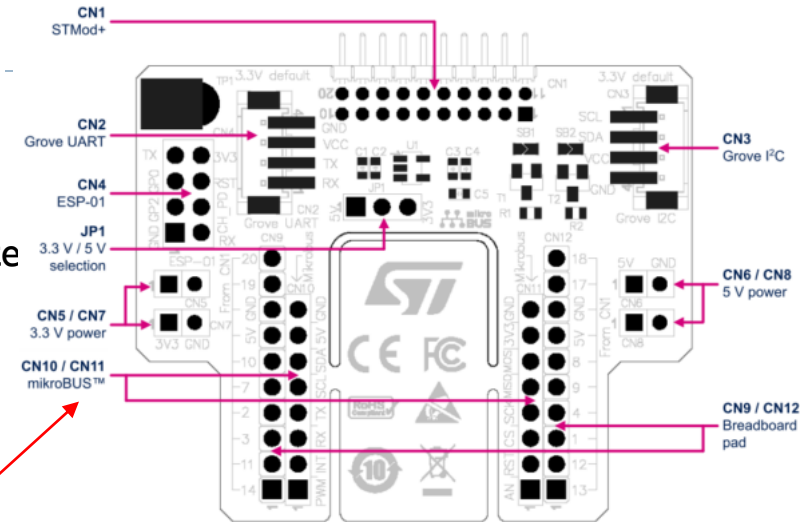
- **I2C** (Inter-Integrated Circuit)
 - I2C 1, 2, 3, 4
- **SPI** (Serial Peripheral Interface)
 - SPI 1,2,3,4,5,6

2.5 mikroBUS™ compatible connectors CN10 and CN11

mikroBUS™ compatible connectors CN10 and CN11 are a pair of 1×8-pin female connectors with a 2.54 mm pitch. Table 7 shows the definition of the pins.

Table 7. Description of the mikroBUS™ connector pins

STMod+ connector CN11	mikroBUS™ function	Pin number	Pin number	mikroBUS™ function	STMod+ connector CN10
STMod+#13-ADC	AN	1	1	PWM	STMod+#14-PWM
STMod+#12-RST	RST	2	2	INT	STMod+#11-INT
STMod+#1-CS	CS	3	3	RX	STMod+#3-RX
STMod+#4-SCK	SCK	4	4	TX	STMod+#2-TX
STMod+#9-MISOs	MISO	5	5	SCL	STMod+#7-SCL
STMod+#8-MOSIs	MOSI	6	6	SDA	STMod+#10-SDA
-	+3.3 V	7	7	+5 V	-
-	GND	8	8	GND	-

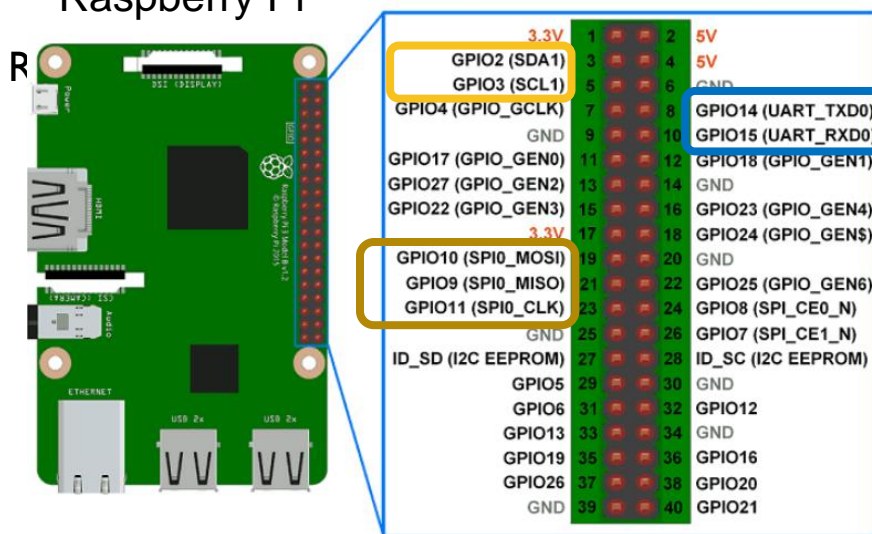


Uvod

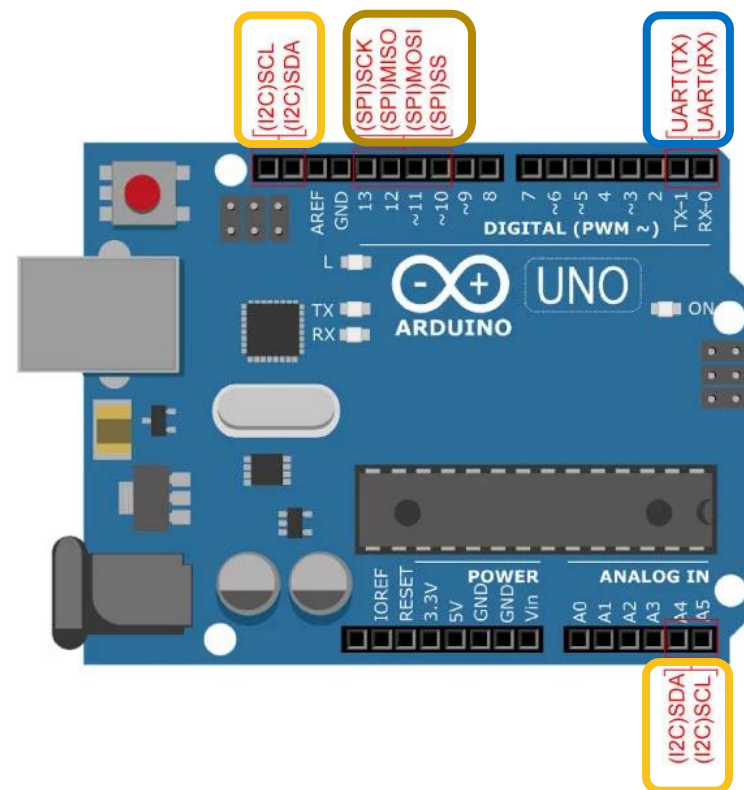
Primer RPi, Arduino:

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

Raspberry Pi



Arduino UNO



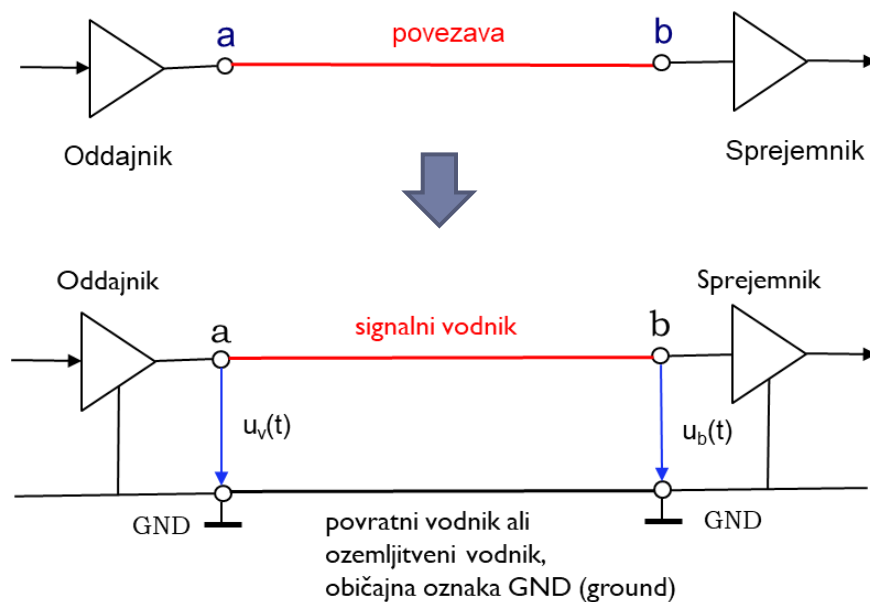
<https://www.electronicwings.com/raspberry-pi/raspberry-pi-gpio-access>

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

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



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



❑ **Serijski način prenosa**

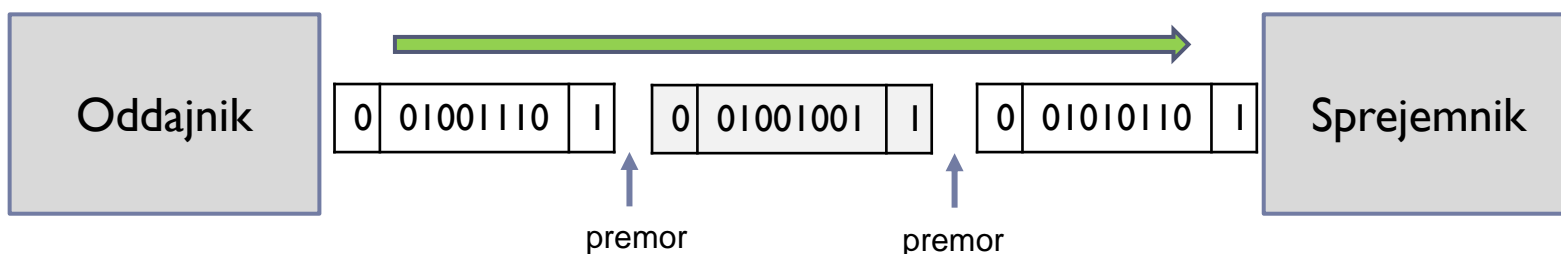
po povezavi se prenaša bit za bitom

❑ **KOMUNIKACIJSKI KANAL**

- Prenos podatkov poteka v **realnem času** (čas med oddajo in sprejemom podatkov je zelo kratek).
- Potrebne pretvorbe:
 - 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

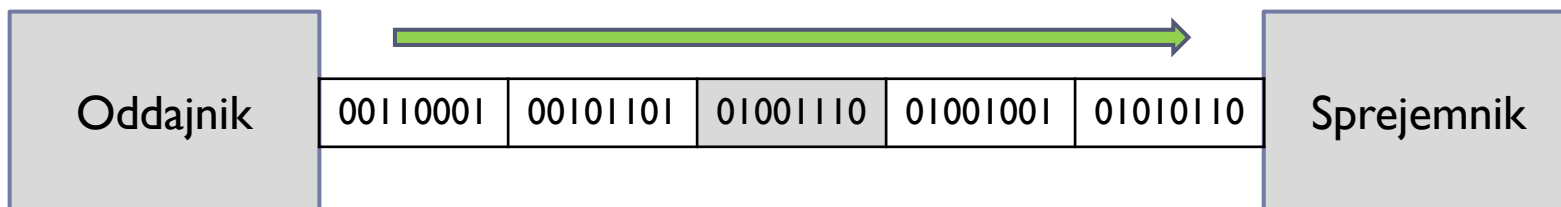
□ Asinhronski serijski prenos

- UART (Universal Asynchronous Receiver/Transmitter)



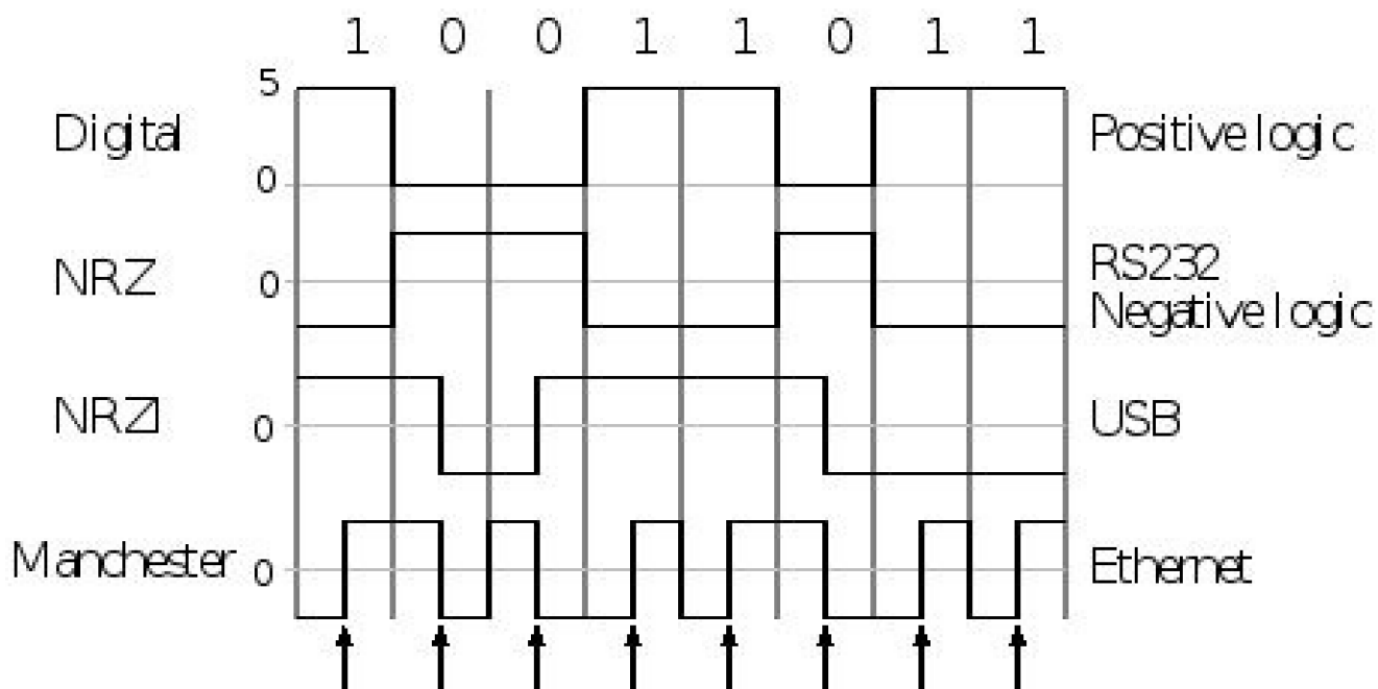
□ Sinhronski serijski prenos

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



▪ **kodiranje podatkov** (pretvorba podatka v dejanski signal):

- NRZ(I) (Non Return to Zero (Inverted))
- PE (Phase Encoded)
- RLL (Run Length Limited): 8b/10b, ...

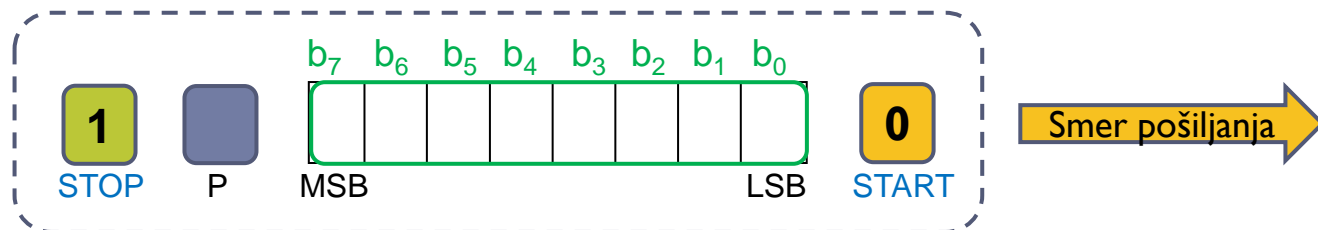


1. Asinhronski serijski prenos - UART

- ❑ **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 („režijski“) biti :
 - START in
 - STOP bit in/ali
 - parnostni (paritetni) bitso v vsakem znaku, kar predstavlja **20% - 30% redundanco** pri prenosu.

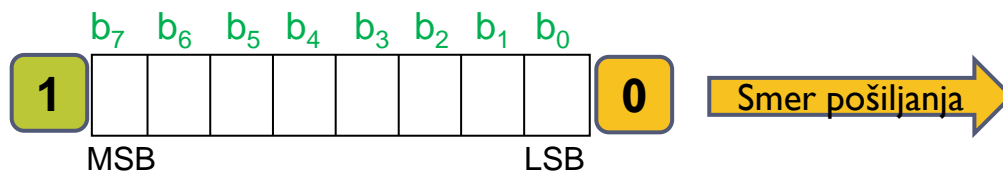
□ Znak sestavljajo:



- **Start bit** je vedno **logična 0** (nasprotno od mirovnega stanja) in določa začetek znaka:
 - 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 običajno pošilja prvi: **LSB** (b_0).
- **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** (sodo število vseh enic podatkovnih in parnostnega bita).
 - **Liha parnost** (liho število vseh enic podatkovnih in parnostnega bita).
- **Stop bit** je vedno **logična 1** (enako mirovnemu stanju) in označuje konec znaka.
 - Mirovno stanje pred začetkom naslednjega znaka

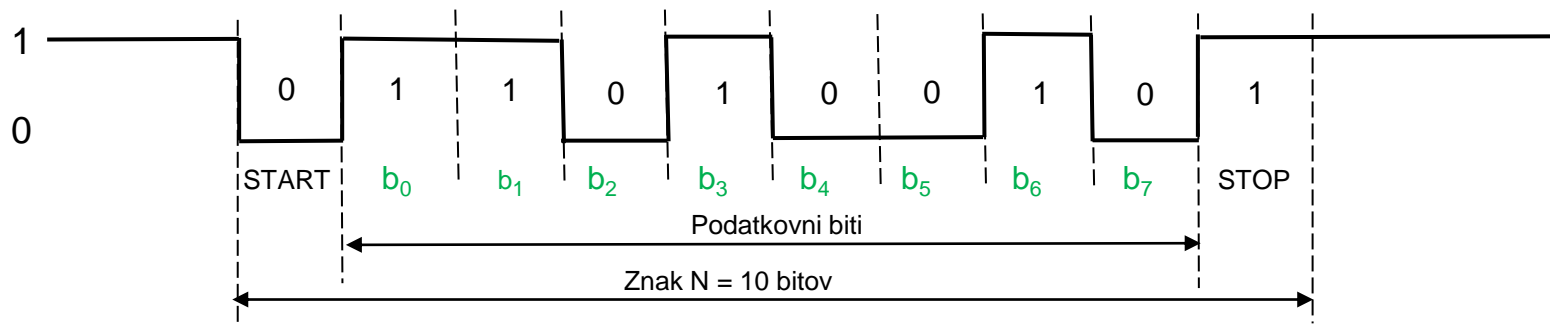
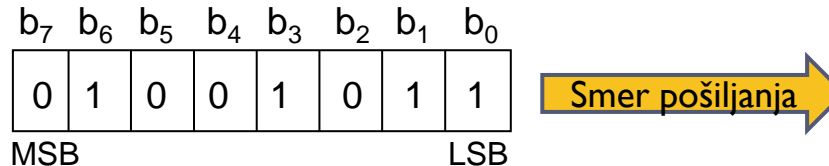
- Najbolj razširjena oblika formata serijskega asinhronskega prenosa pri osebni računalnikih (PC) je **8-N-1** in pomeni:

- 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)

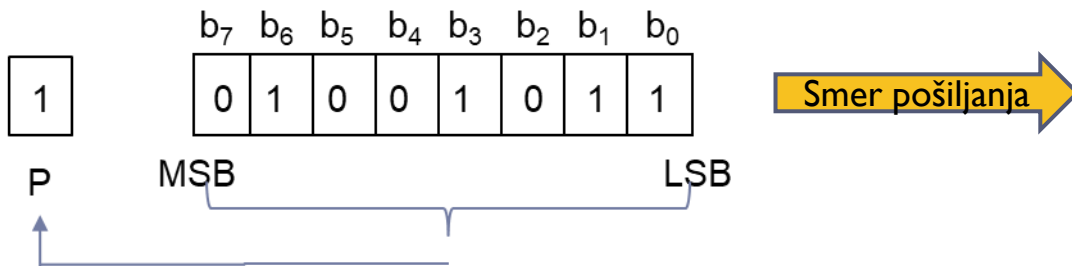


- Format znaka lahko podamo tudi takole:

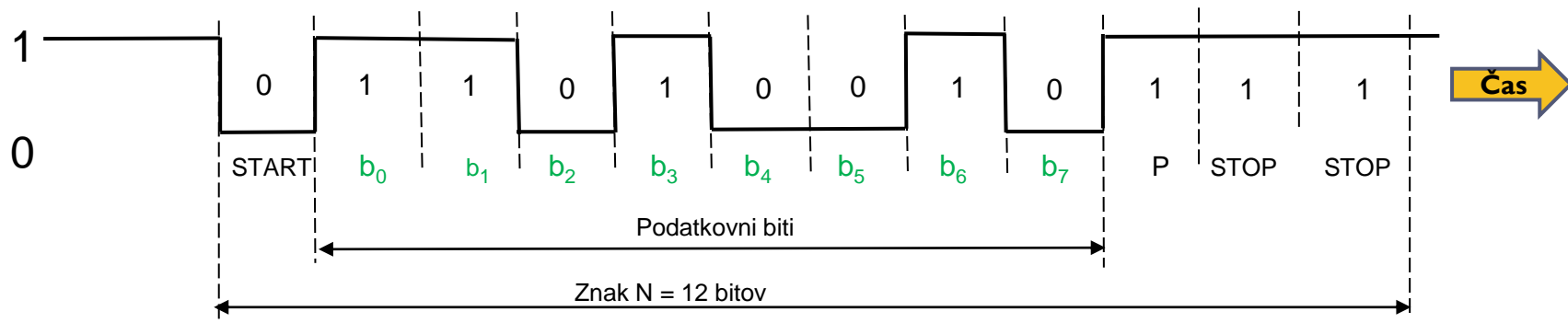


□ Primer formata znaka ASCII "K" z **liho parnostjo** in dvema stop bitoma:

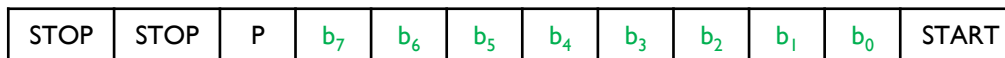
- ASCII "K" = 4B (Hex)
- P = 1



- Oblika prenosa: 8-O-2, kjer je N=12



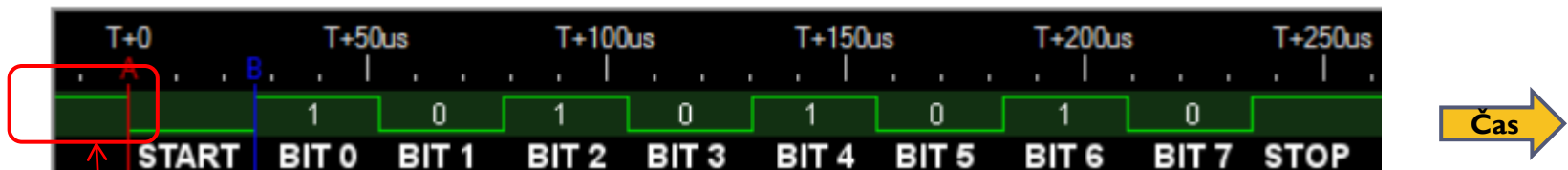
- Format znaka lahko podamo tudi takole (12 bitov):



□ Primer serijskega prenosa črke 'U'

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

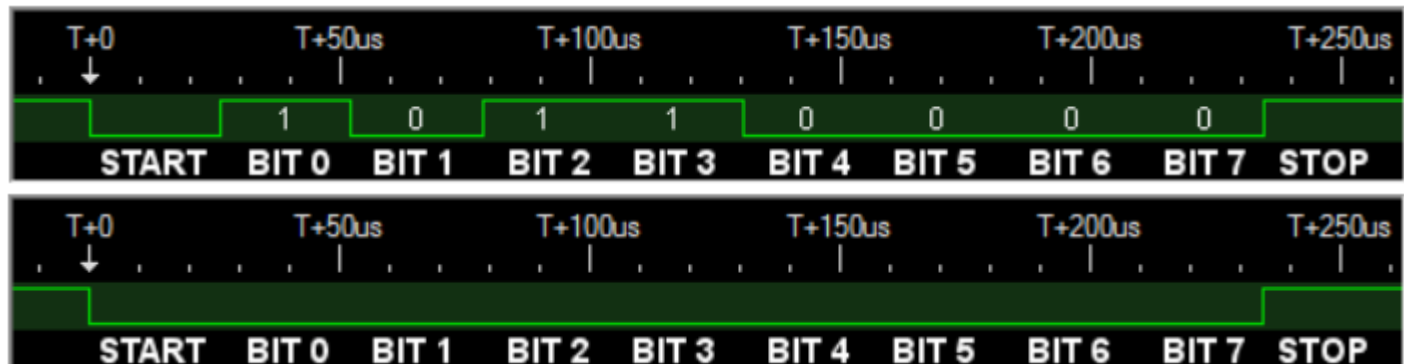
- Koda ASCII ima numerično vrednost 85, ali 55_H).
- Dvojiški 8-bitni zapis podatka je **01010101**



- 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:

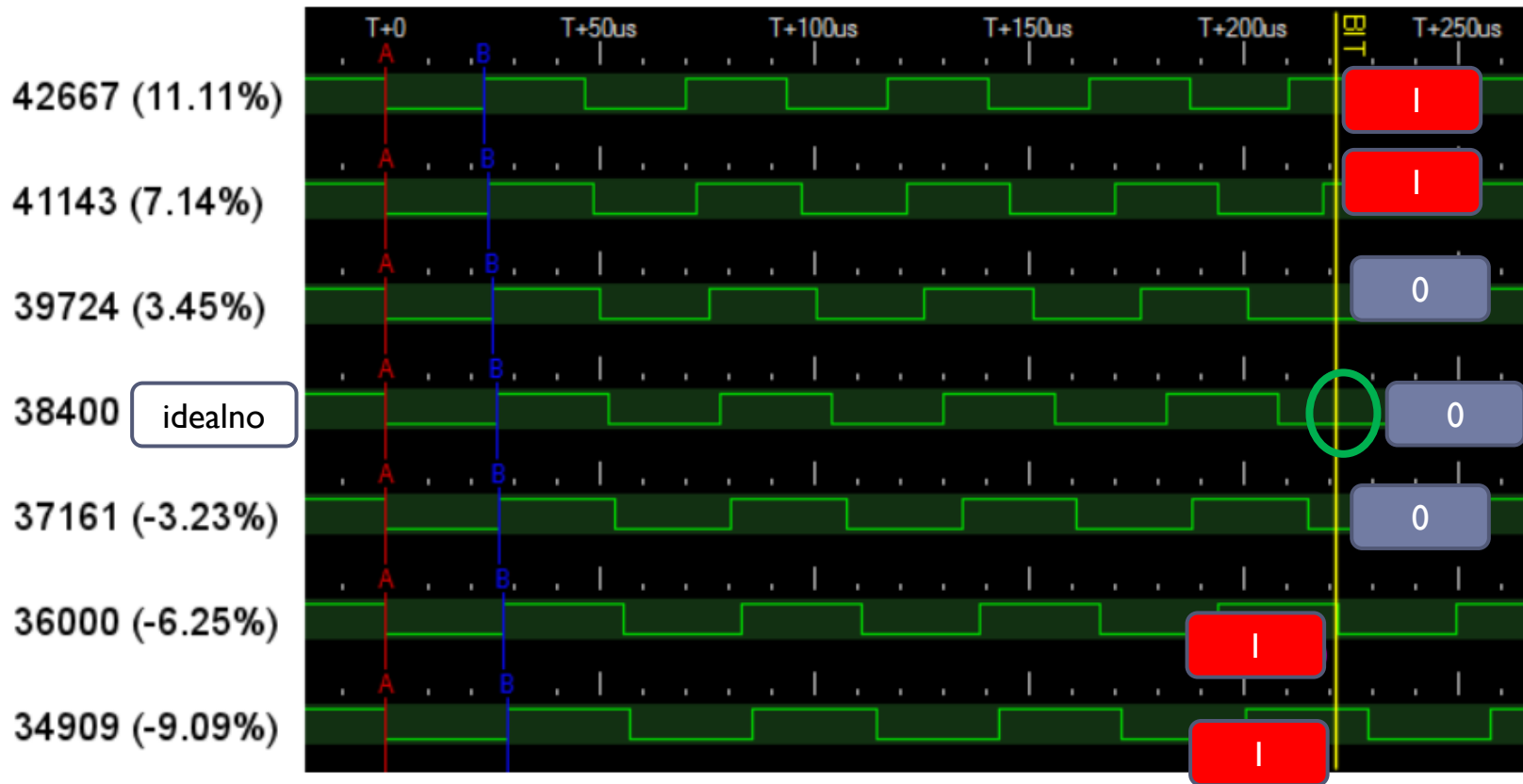
- CR (ASCII 13)
(carriage
return)
- Podatek = 0



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

- Napaka se pri prenosu zgodi v primeru, ko je oddajnik prehitel (> 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

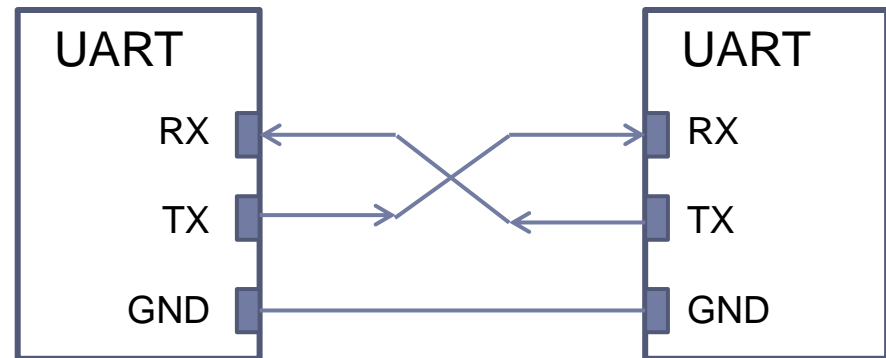
- UART kot samostojen čip v integraciji LSI (Large Scale Integration).
- UART kot logično vezje, ki je del mikrokrmilnika.
- UART ima 2 liniji za prenos podatkov med dvema napravama:
 - TX (ang. transmitter) – oddaja
 - RX (ang. receiver) – sprejem

„Universal“

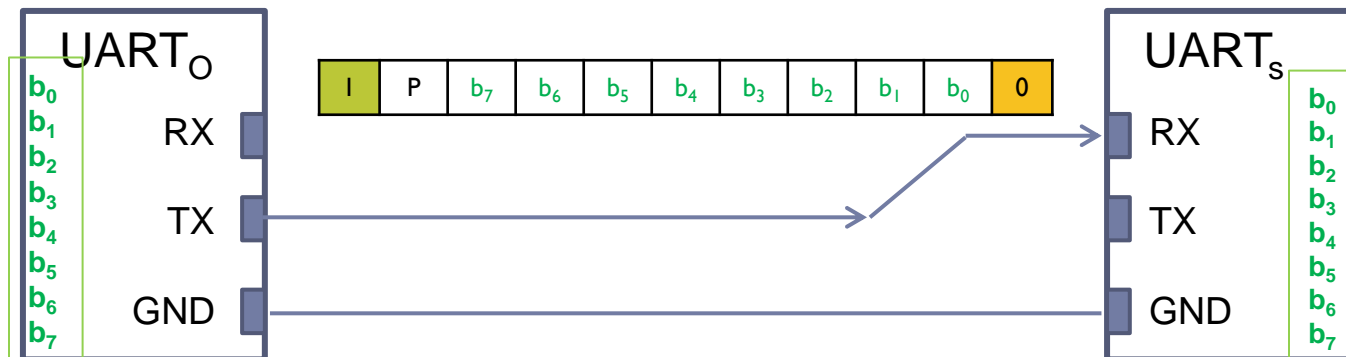
- UART is programmable.

„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)

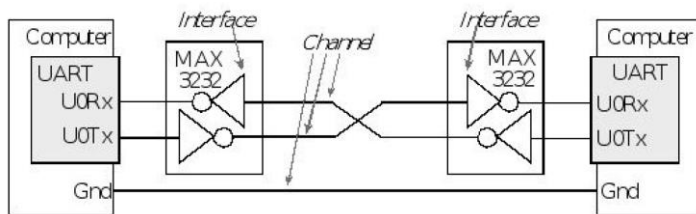
- Hitrost prenosa podatkov podajamo kot baudna hitrost
 - Najbolj običajna, standardizirana **baudna hitrost je 9600**.
 - Druge standardne baudne hitrosti so:
1200, 2400, 4800, 19200, 38400, 57600 in 115200.

- UART je cenovno ugodna komunikacijska naprava.

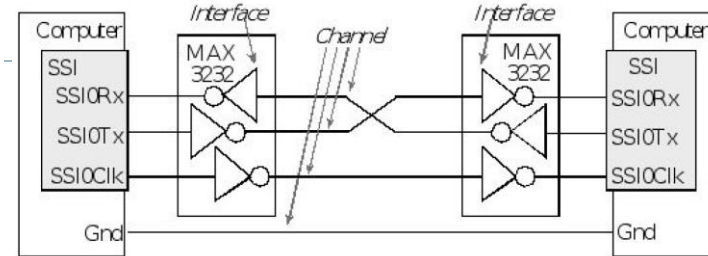
- USART (Universal Synchronous Asynchronous Receiver Transmitter) - univerzalni sinhronski in asinhronski sprejemnik / oddajnik, ki omogoča tako asinhronski, kot sinhronski serijski prenos podatkov.

UART in različne komunikacije

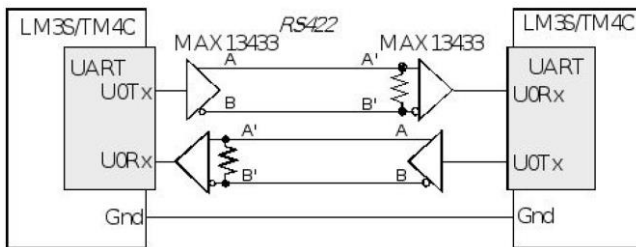
UART (RS232)



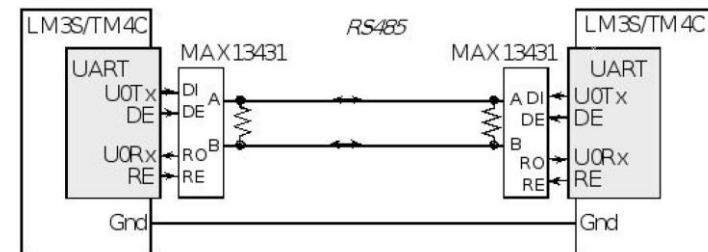
USART (RS232)



UART (RS422)



UART (RS485)



		+3.3V logic	+5V logic	RS232 level	RS422 level
True	Mark	+3V	+5V	TxD = -5.5V	(TxD ⁺ - TxD ⁻) = -3V
False	Space	+0.1V	+0.1V	TxD = +5.5V	(TxD ⁺ - TxD ⁻) = +3V

Table 7.1. Typical voltage levels for the digital logic, RS232 and RS422 protocols.

Specification	RS232D	RS423A	RS422	RS485
Mode of operation	single-ended	single-ended	differential	differential
Drivers on one line	1	1	1	32
Receivers on one line	1	10	10	32
Maximum distance (feet)	50	4,000	4,000	4,000
Maximum data rate	20 kb/s	100 kb/s	10 Mb/s	10 Mb/s

□ Zahteve za pravilen prenos

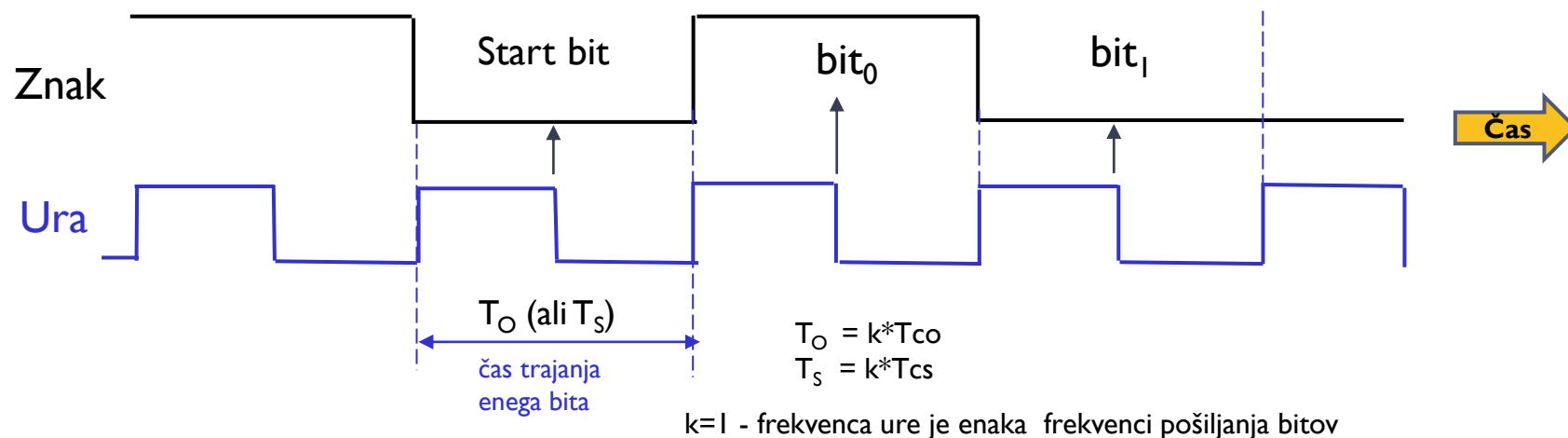
- Enak format, nastavitve na sprejemni in oddajni strani

- 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}): $f_{co} = f_{cs}$

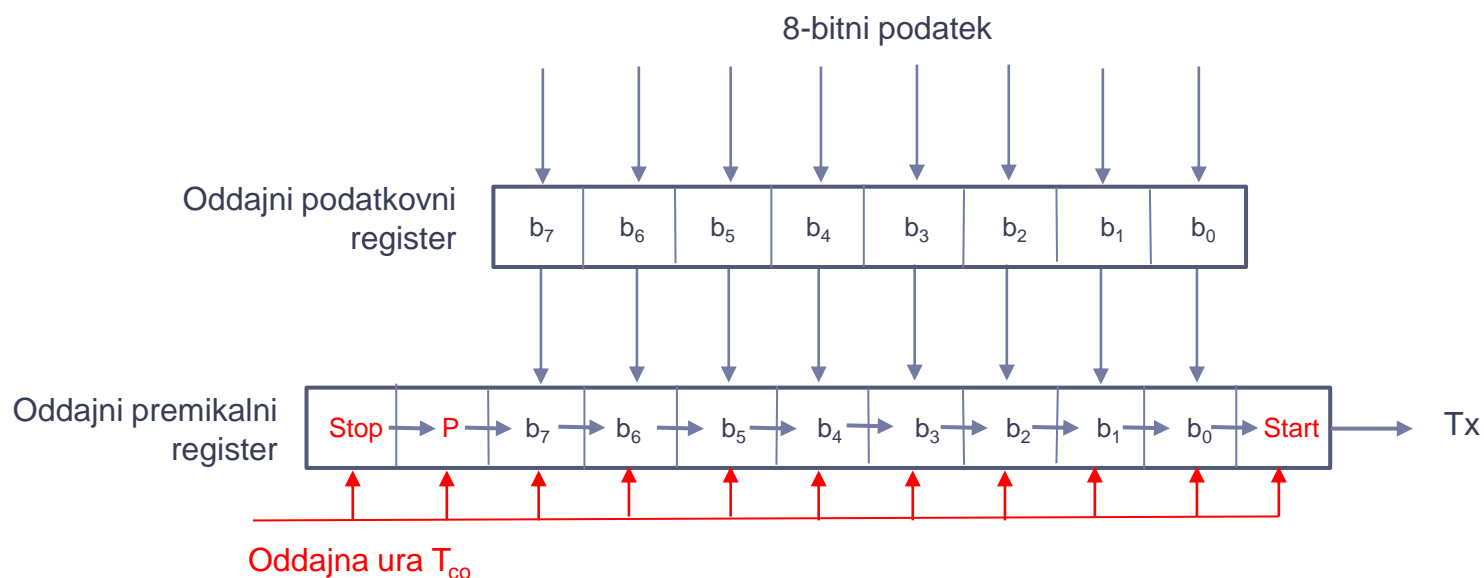
c_o – clk oddaja, c_s – clk sprejem

- 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 4

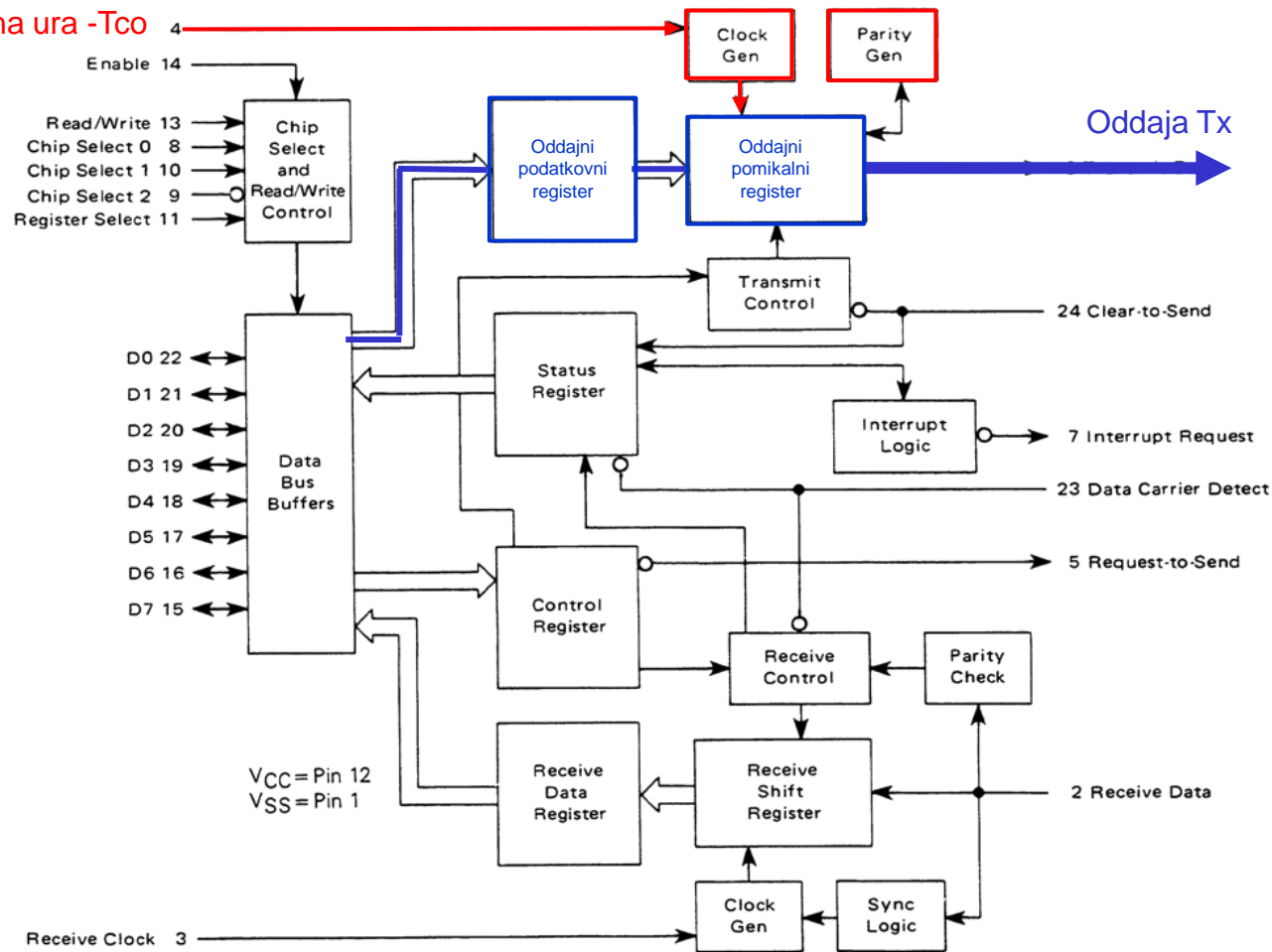
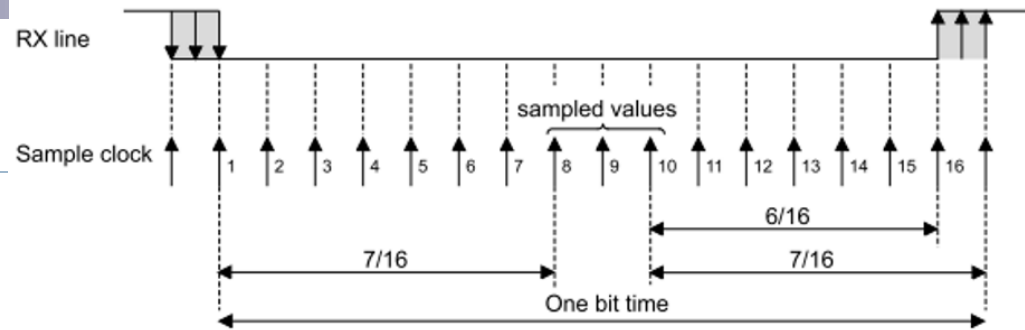


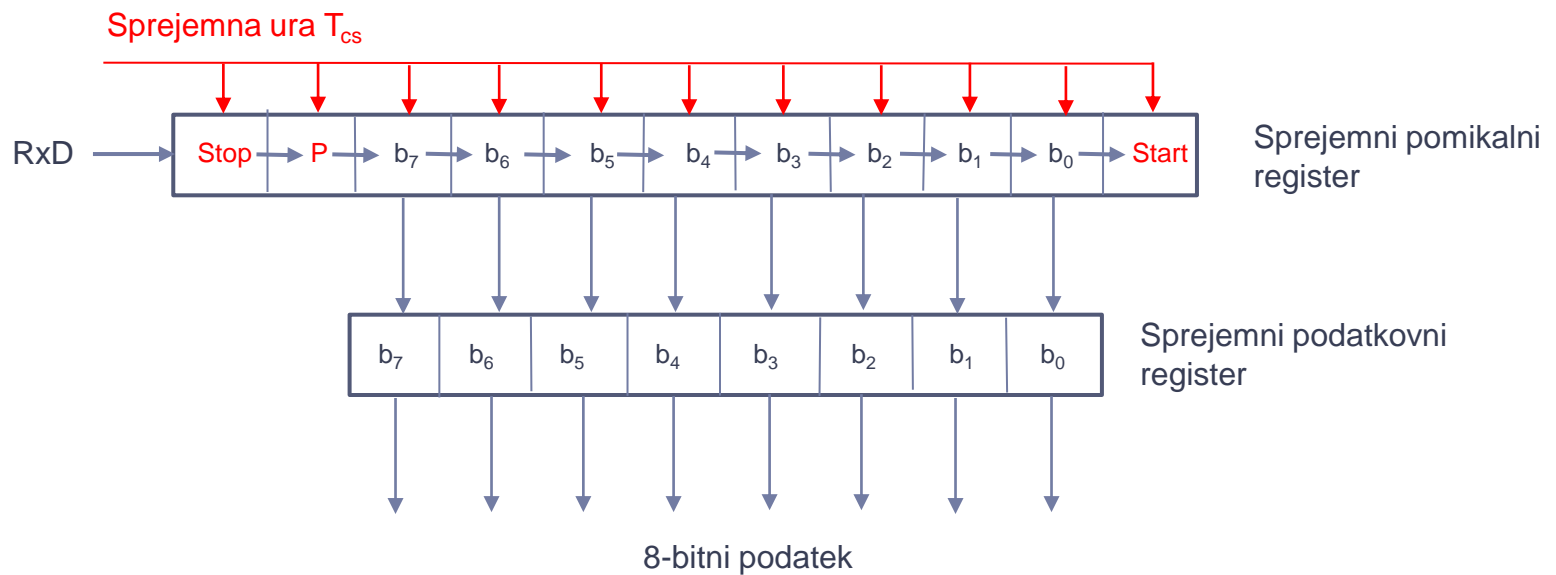
Figure 301. Data sampling when oversampling by 16



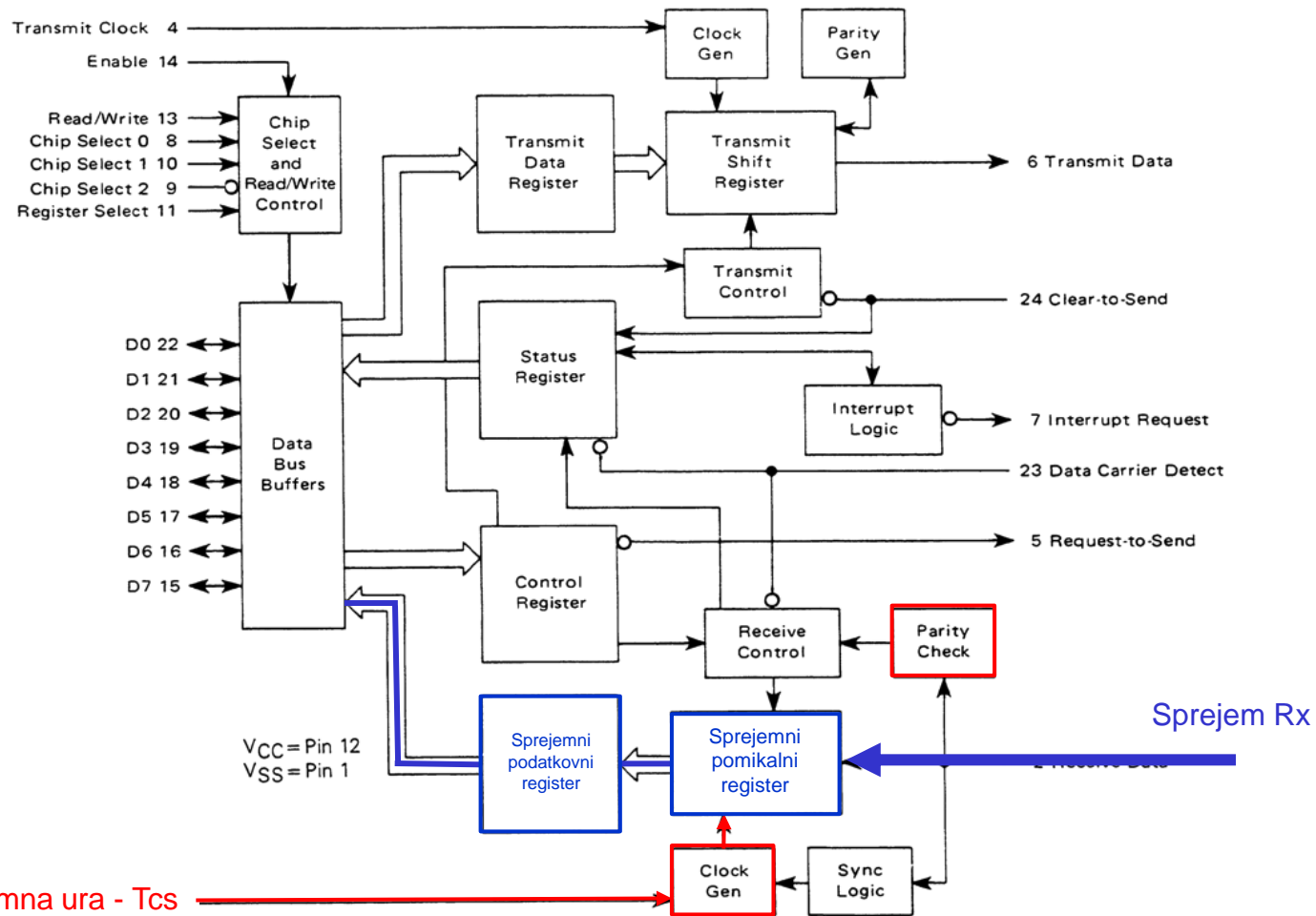
Sprejemnik običajno bere stanje 8 ali 16x pogostejše (angl. Oversampling)

UART – sprejem

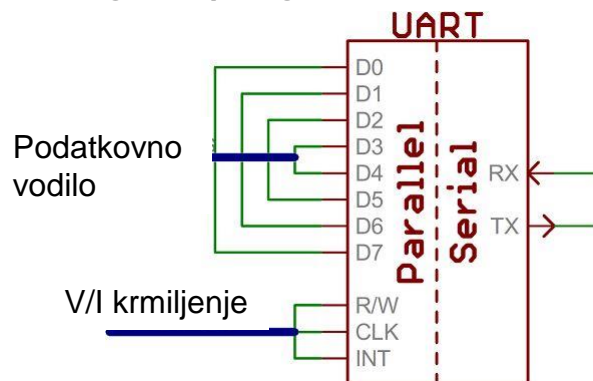
- Serijsko paralelna pretvorba
- Kontrola pravilnosti (parnostni bit)
- Odstranitev okvirja (start, stop bit in parnostni bit)



UART – sprejem



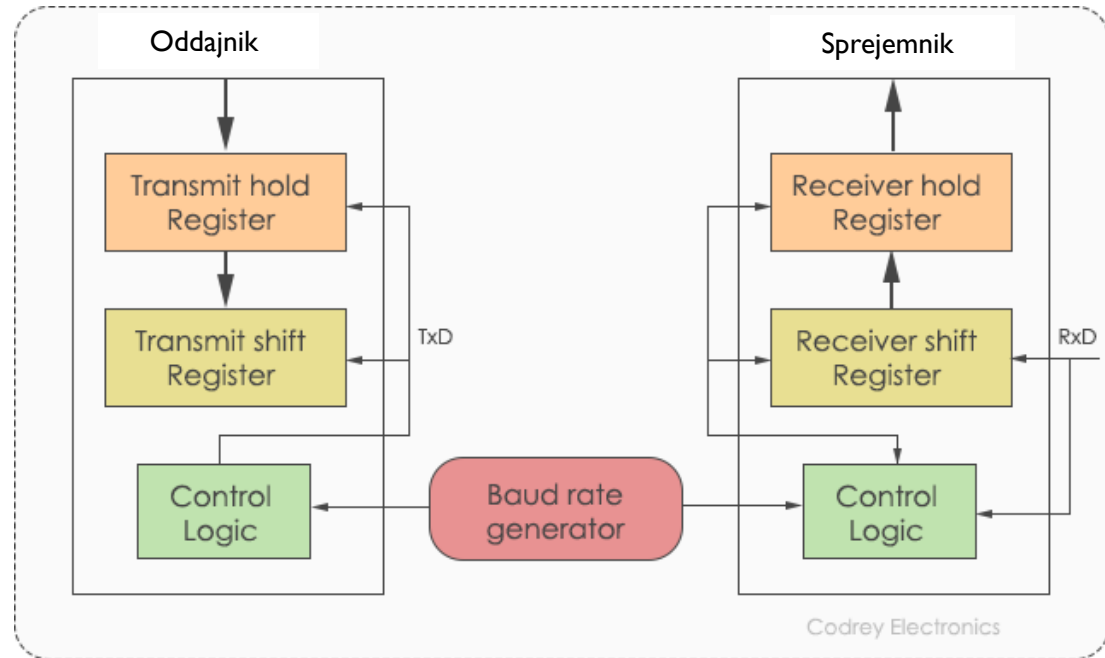
- UART je mogoče programirati, tako da določimo parametre prenosa:



- 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 dvojnimi izravnanimi ali s FIFO vmesnikom

□ UART - Blok diagram

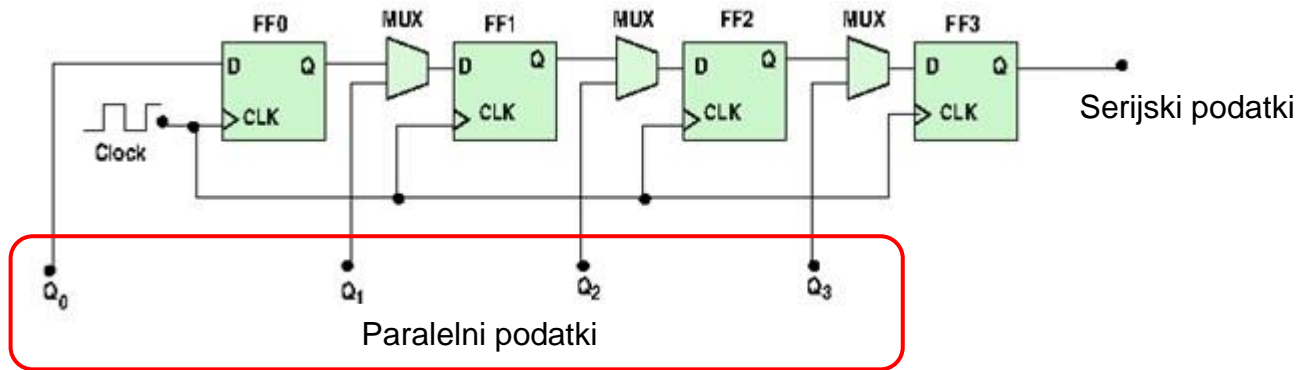
- Sprejemnik
- Oddajnik



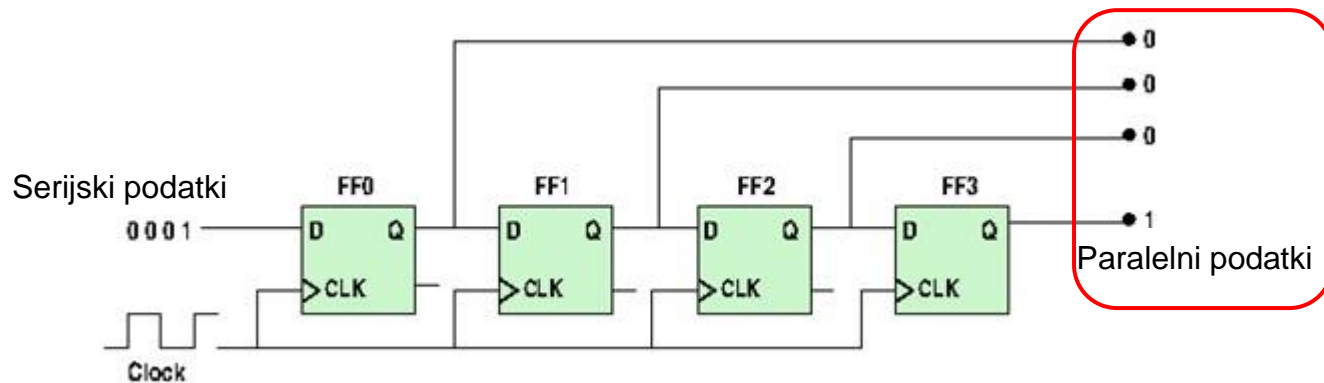
<https://www.codrey.com/embedded-systems/uart-serial-communication-rs232/>

- '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

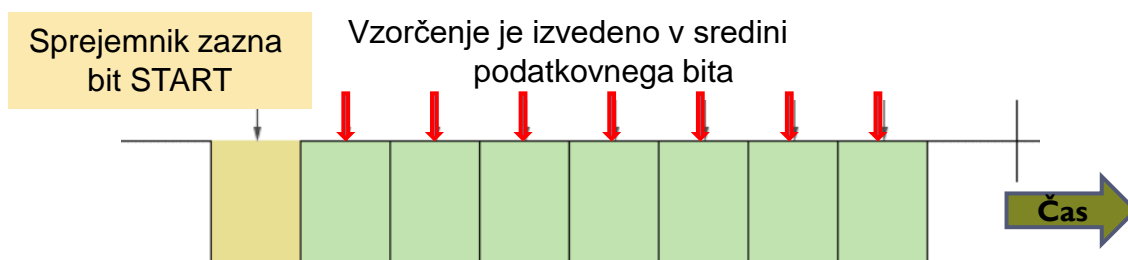
- Format protokola



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

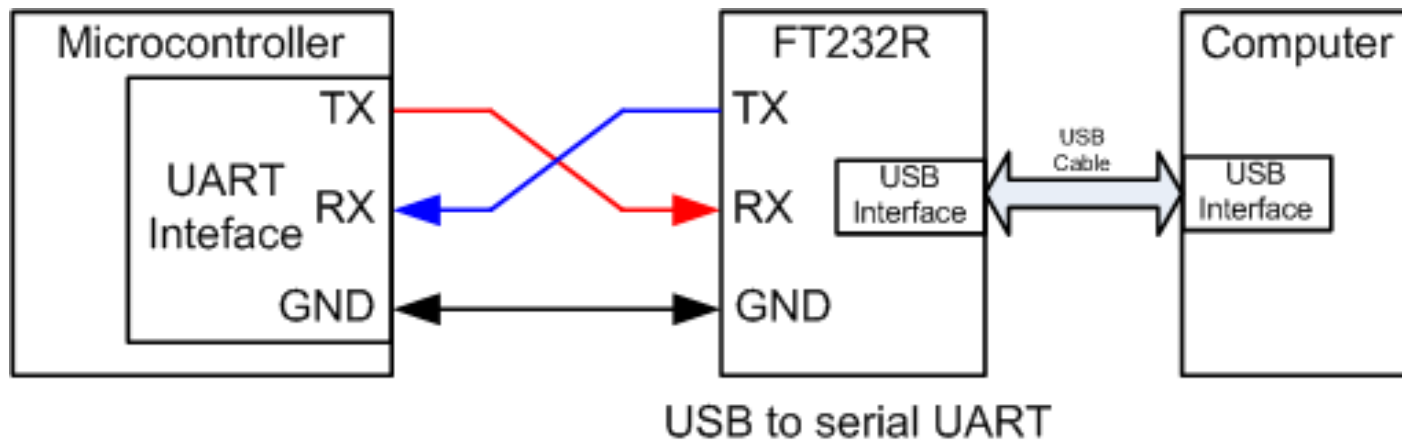


- Sprejemnik sprejme podatek po prenosni liniji RxD (LSB bit je prvi prejet) in
 - zazna bit START
 - izvede vzorčenje



Običajna povezava UART <-> PC

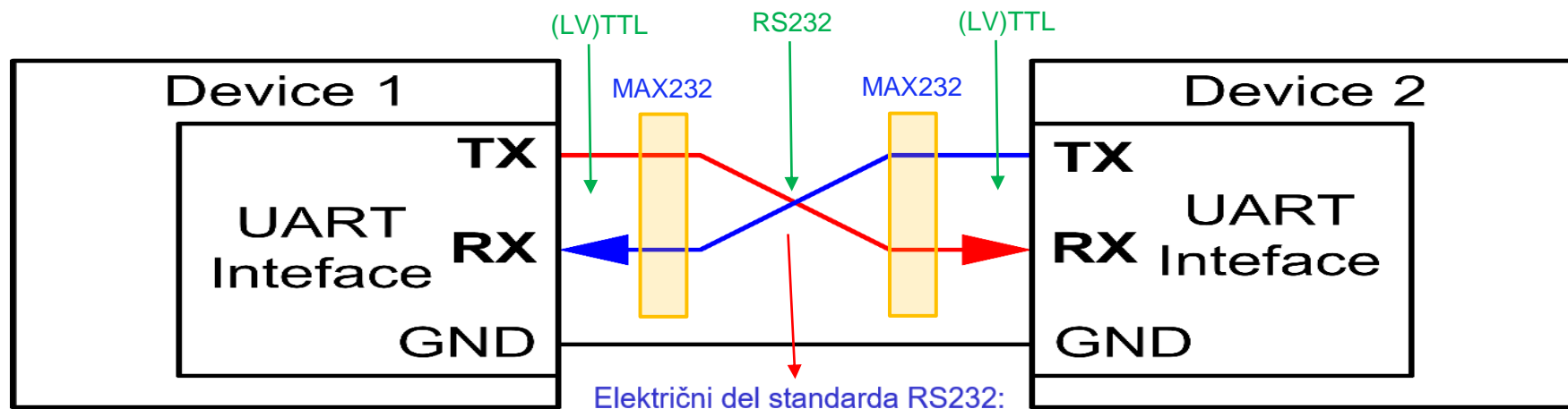
- Primer povezave na PC (običajno nima UART priključka):
 - FT232R pretvarja med UART RS232 in USB vmesnikom (PC)



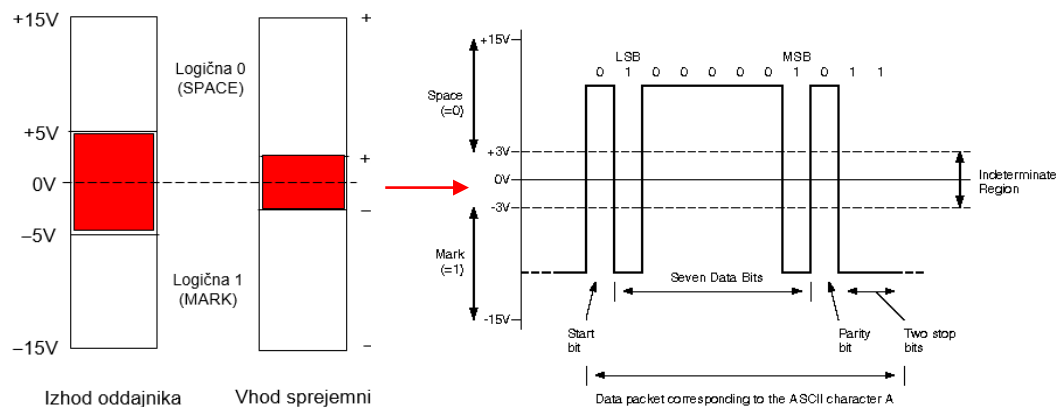
UART - električno

□ Napetostni nivoji so lahko (odvisno od pretvornika):

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



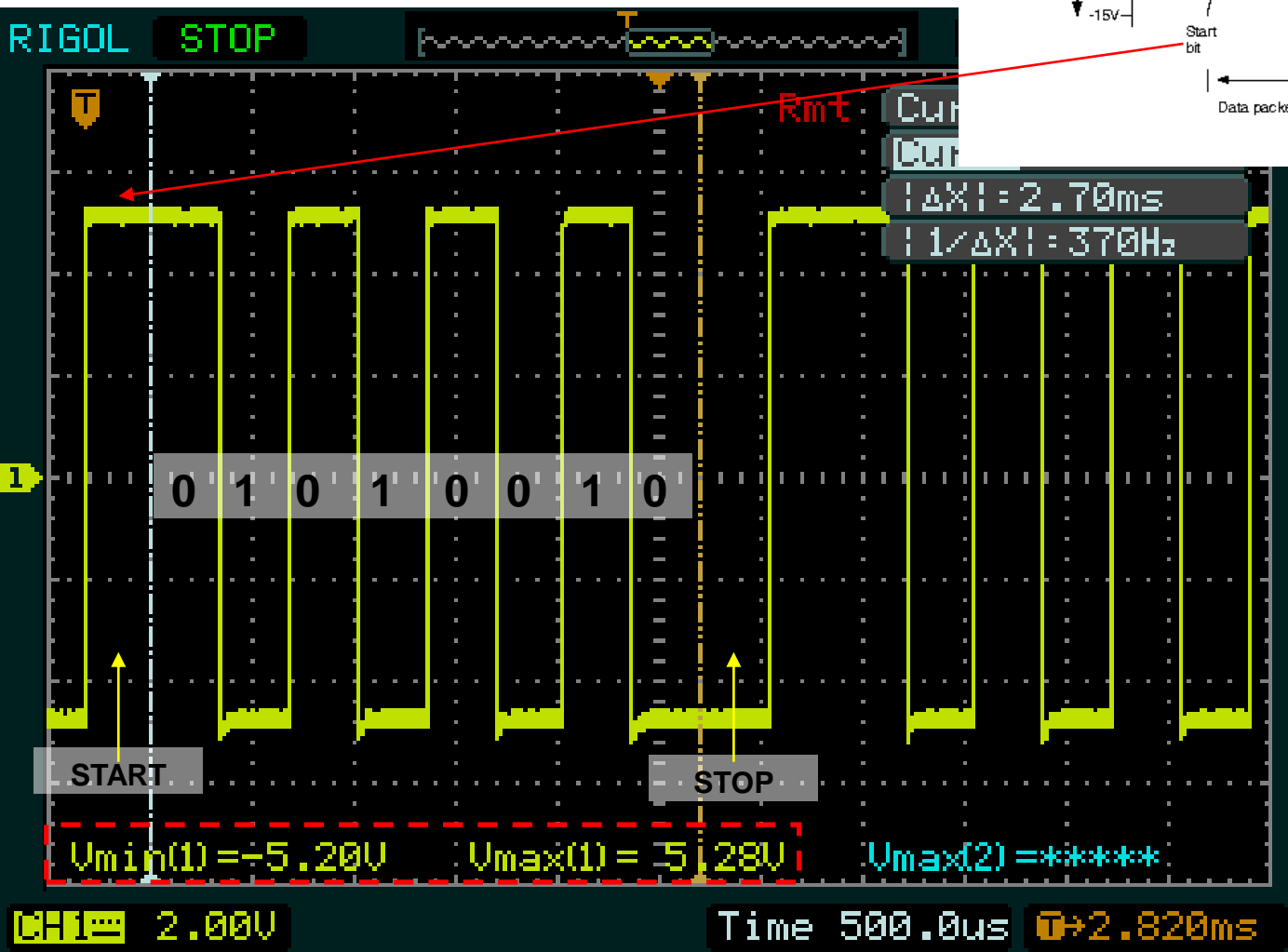
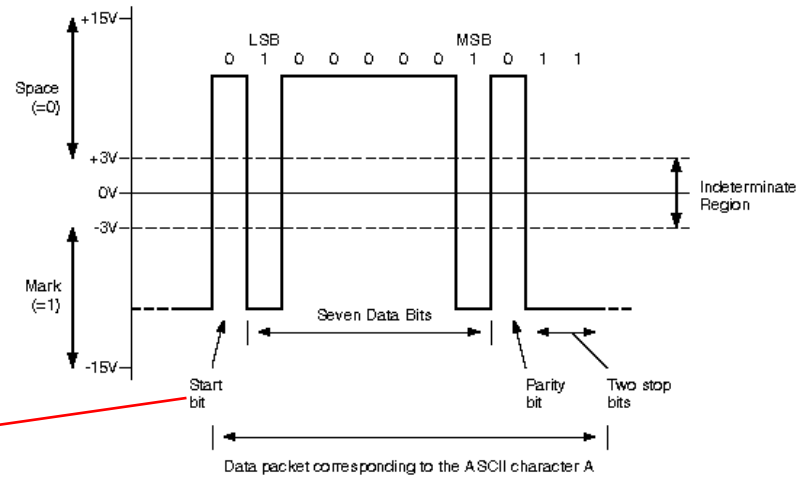
□ Napetostna in logična nivoja



UART - Standardi

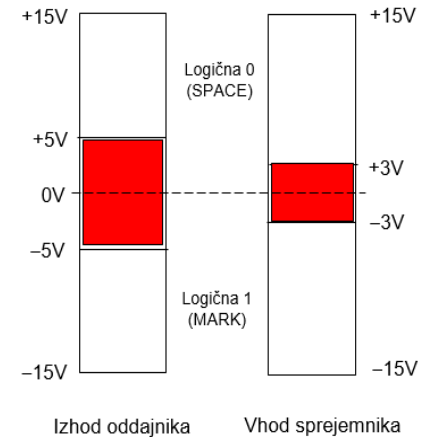
Standard	Voltage signal	Max distance	Max speed	Number of devices supported per port
RS-232	Single end (logic 1: +5 to +15V, logic 0: -5 to -15V)	100 feet	115Kbit/s	1 master, 1 receiver
RS-422	Differential (-6V to +6V)	4000 feet	10Mbit/s	1 master, 10 receivers
RS-485	Differential (-7V to +12V)	4000 feet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

UART – osciloskop



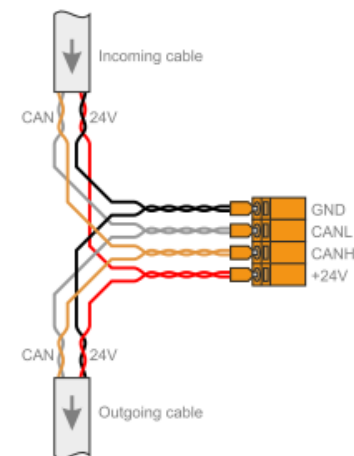
Električni del standarda RS232:

□ Napetostna in logična nivoja



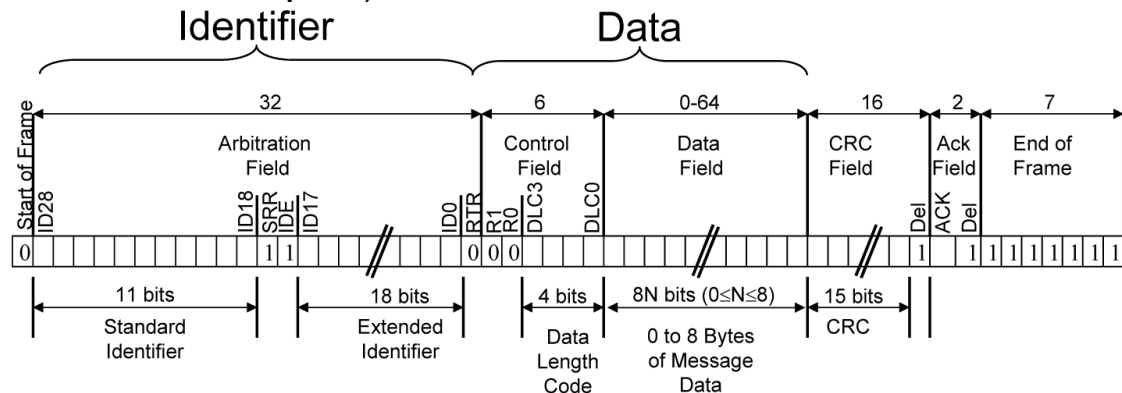
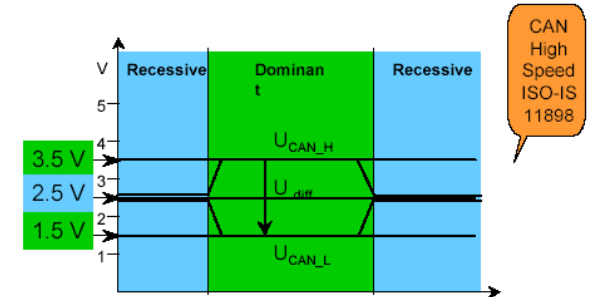
CANbus na kratko

- ▶ **CAN**bus – **C**ontroller **A**rea **N**etwork bus
 - ▶ serijsko vodilo za komunikacijo med vgrajenimi mikrokrmilniki
- ▶ CAN bus na kratko :
 - ▶ dve žici (CAN_H,CAN_L) + napajanje,
 - ▶ diferencialni prenos signala
 - ▶ odpornost na šum.
 - ▶ max 1 Mbit/s, 40m,
 - ▶ sporočila do 8 bajtov (latenca)
 - ▶ CAN-FD tudi daljša sporočila
- ▶ CAN-FD standard, ISO 11898-2:2016
 - ▶ 2Mbps, 5Mbps



CANbus na kratko

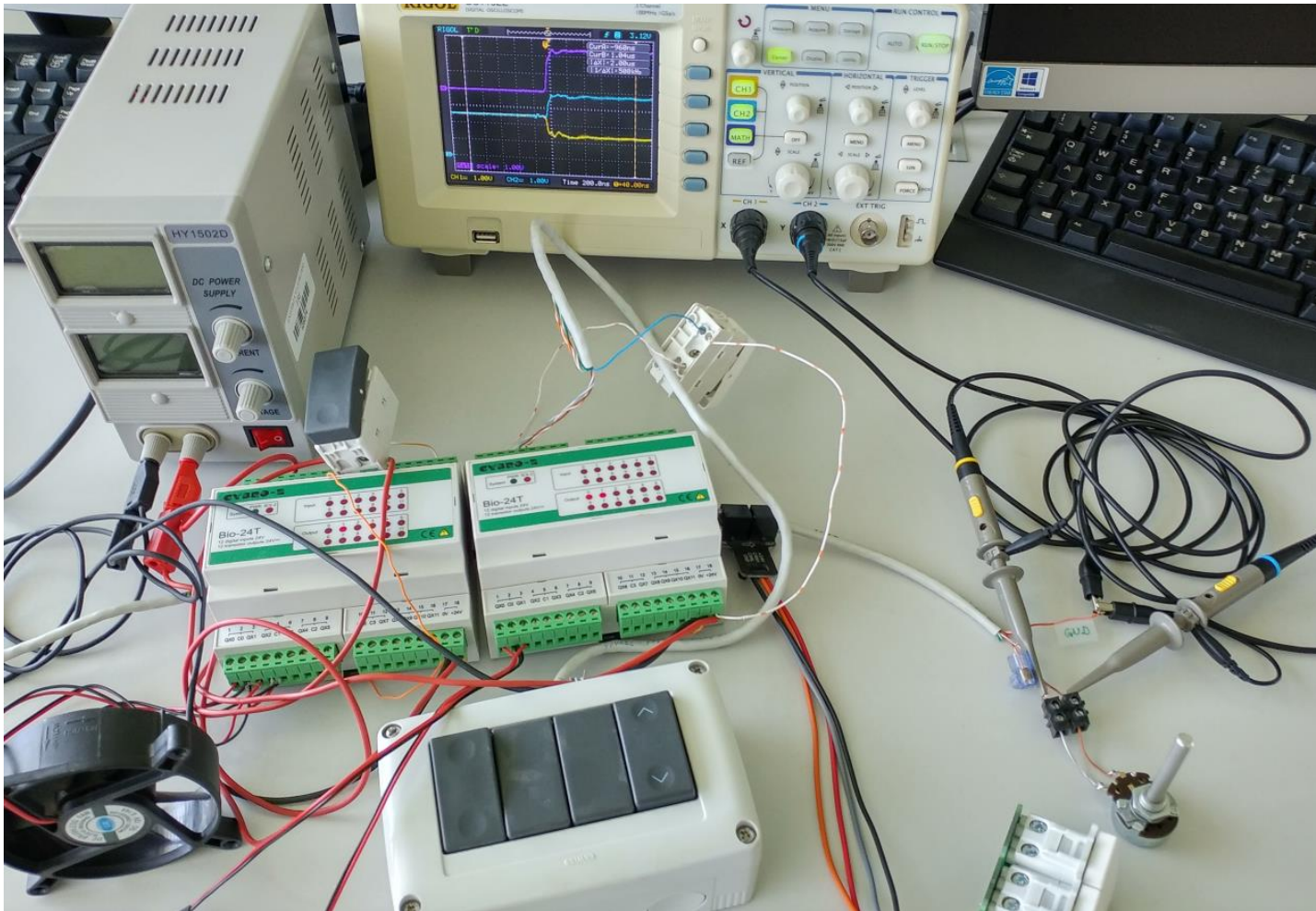
- ▶ **Diferencialni prenos** običajno na parici –
 - ▶ Non-Return To Zero (NRZ) in
 - ▶ bit-stuffing.
- ▶ Wired –AND povezava: vozlišče z logično 0 prevlada
 - ▶ 0 .. „dominant“, 1 .. „recessive“)
- ▶ Prenos podatkov
 - ▶ Protokol – sporočilno naravnan
 - ▶ Detekcija napake
 - ▶ Nivo Bitov (branje, „bit stuffing“)
 - ▶ Nivo sporočila (CRC, okvir, ACK napake)



CANbus na kratko

- ▶ Komunikacijski protokol:
 - ▶ **CSMA/CD**
 - ▶ **Carrier Sense Multiple Access/Collision Detection (Avoidance)**
 - ▶ z **NDA**
 - ▶ **Non-Destructive Arbitration**
- ▶ Sporočilno orientiran :
 - ▶ ni naslova, vsako sporočilo ima svojo **ID številko** (prioriteta, vsebina) – mora biti unikatna
 - ▶ potrjevanje (ACK)
 - ▶ se zazna potrditev vsaj enega sprejemalca

Lab. vaja - meritve



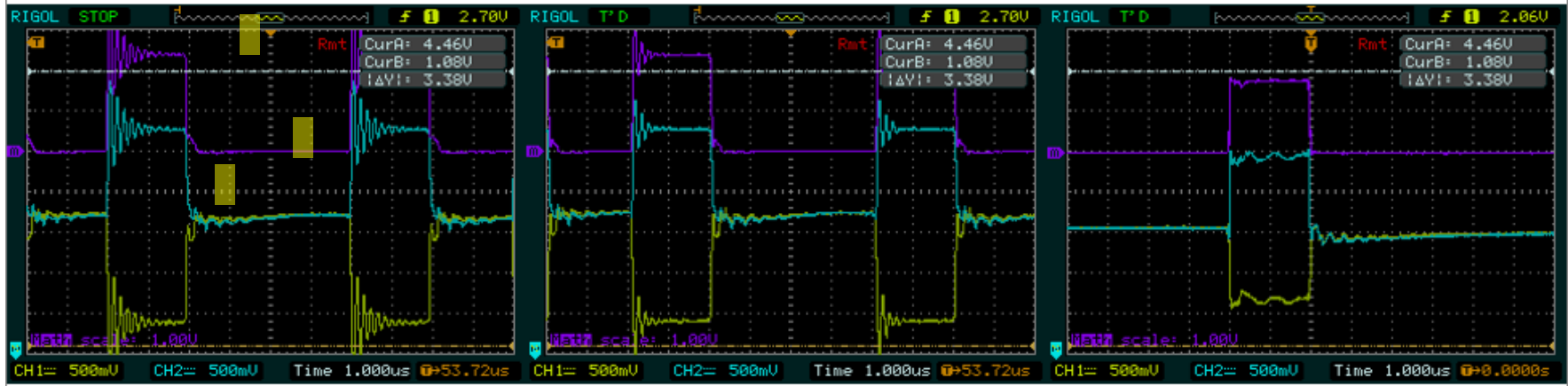
Lab. vaja – primerjava hitrosti

500kb/s:

Odprte sponke

500ohm

107ohm

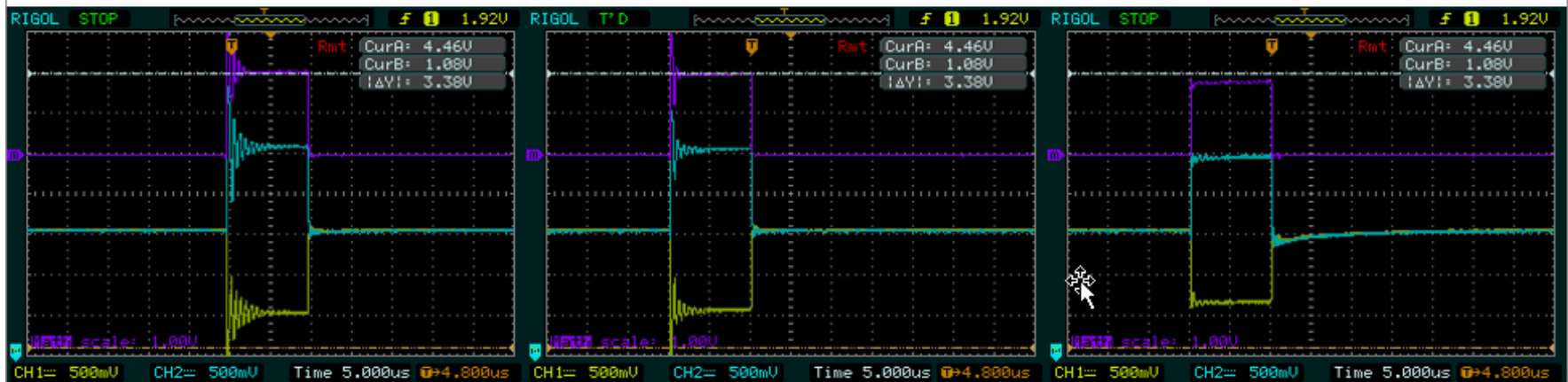


100kb/s:

Odprte sponke

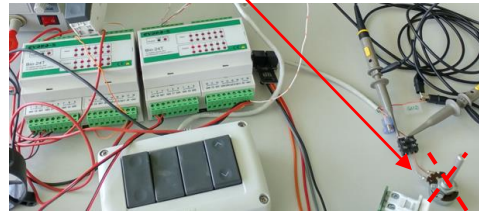
500ohm

107ohm

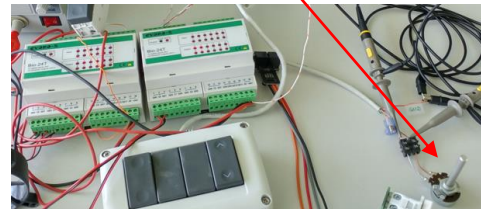


Lab. vaja - meritve

Nezaključena linija

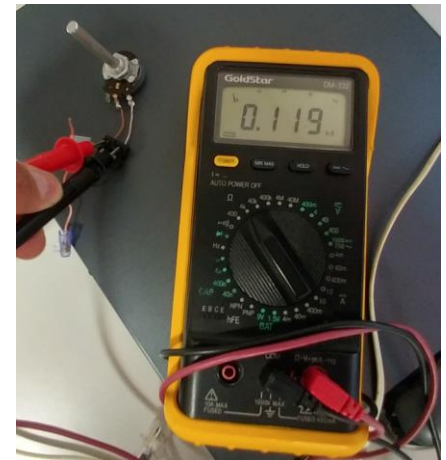


Zaključena linija



Ročna meritev

R_0

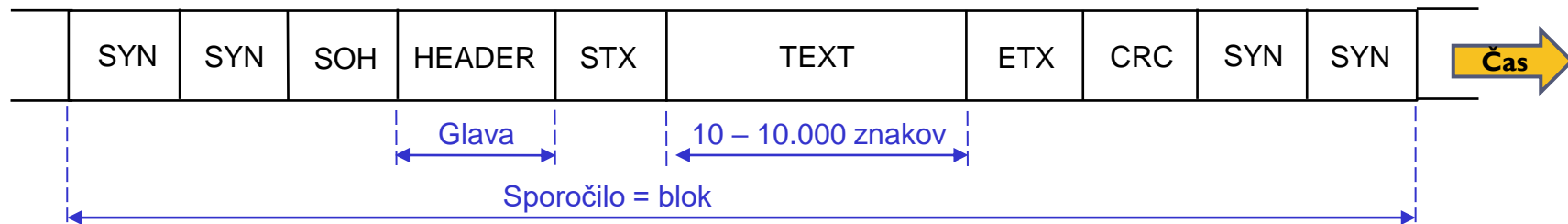


2. Sinhronski serijski prenos

- ❑ 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.
 - ❑ 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 ⇒ **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

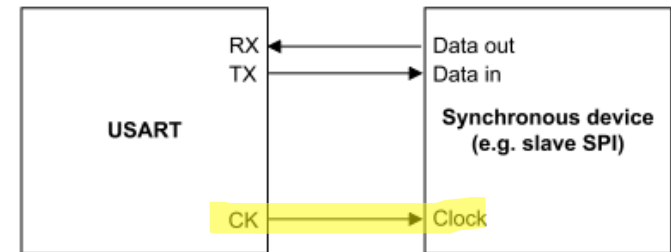


Figure 308. USART data clock timing diagram (M=0)

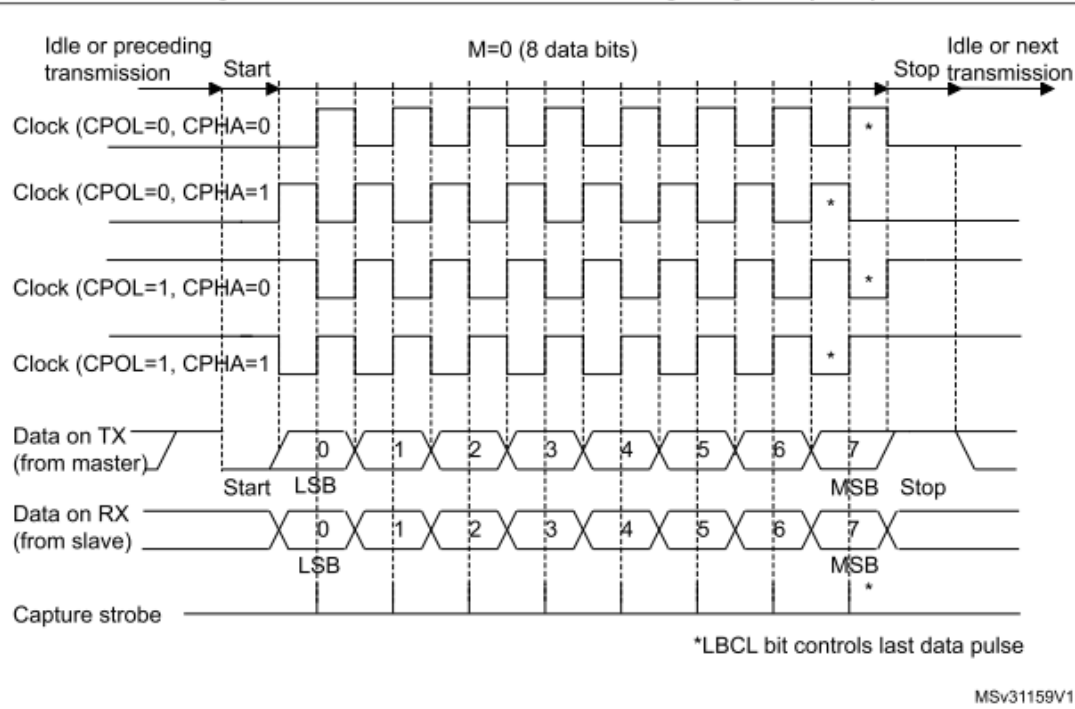
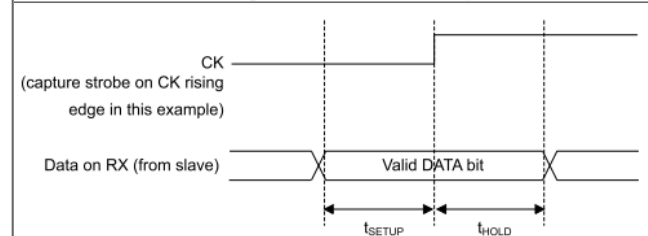
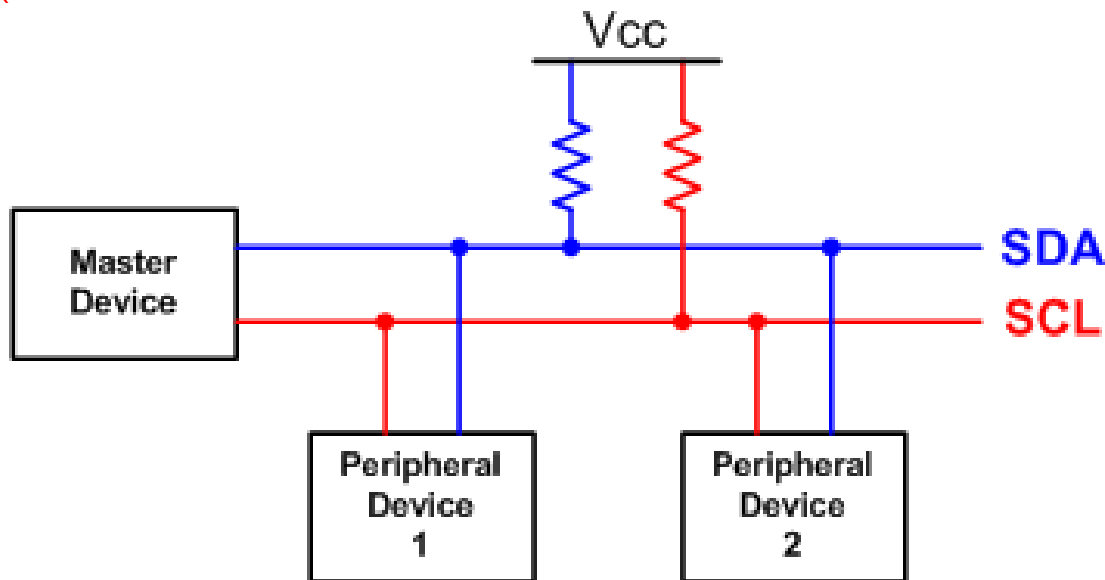


Figure 310. RX data setup/hold time



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

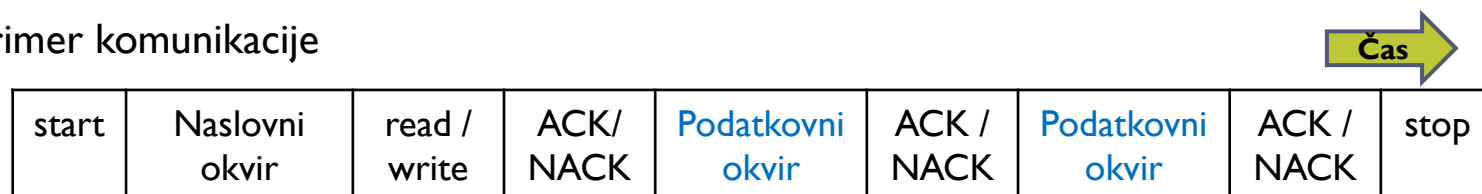
- ❑ **1982** razvito kot interno vodilo za povezavo med čipi proizvajalca **Philips (NXP)**
- ❑ 2 liniji za prenos podatkov med dvema napravama (Master in Slave):
 - **SDA (serial data)** – linija za pošiljanje in sprejemanje podatkov
 - **SCL (serial clock)** - urin signal, ki ga vedno pošilja glavna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



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

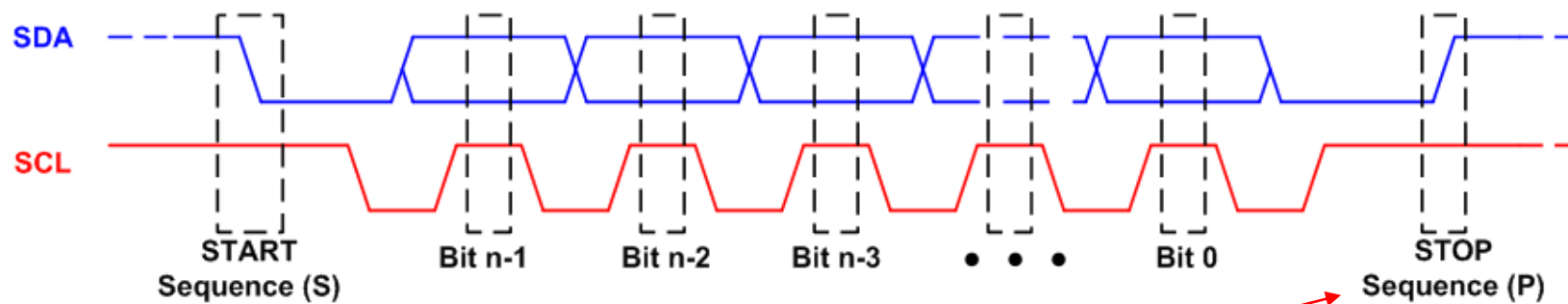
2.1 Vodilo I2C, IIC, I²C (Inter-Integrated Circuit)

- ❑ Sinhronski serijski prenos, dvosmerni izmenični (ang. half duplex)
- ❑ Open drain izhodi, pull-up upori -> „wired-and“ vezava
- ❑ Multi master z arbitražo in sinhronizacijo ure
- ❑ Podatki se vedno pošiljajo v **sporočilih** (ang. messages), ki so razdeljena v **okvirje** (ang. frames) podatkov in dodatne bite:
 - **Naslovni okvir** (ang. address frame) – naslov delovne celice oz. naprave
 - **Podatkovni okvir** (ang. data frame) – podatki, ki se prenašajo.
- ❑ Primer komunikacije



- Dodatni biti:
 - **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.

□ Stanje na linijah



▪ Dodatni bita:

- pogoj **start** – linija SDA preklopi iz 1 v 0 preden 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

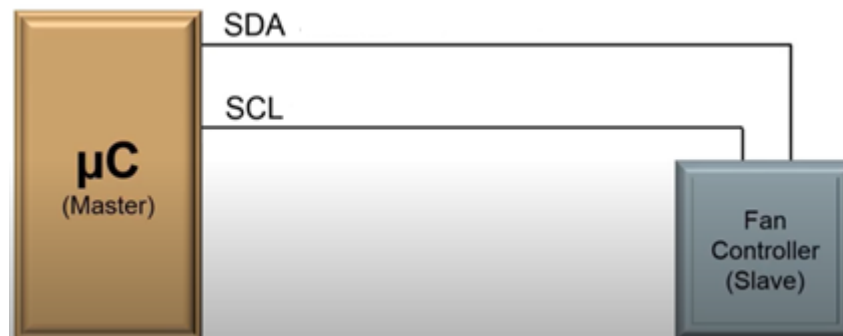


Dodatni biti:

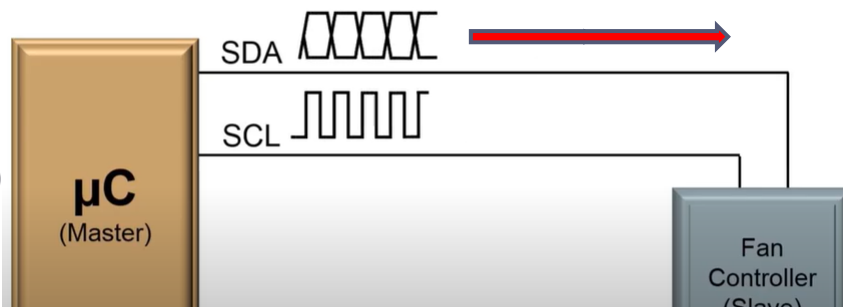
- ▶ **read/write** – en bit določa prenos iz ‘master’ v ‘slave’ napravo (0) ali ‘master’ zahteva podatek iz ‘slave’ naprave (1).
- ▶ **ACK/NACK** – vsak okvir sporočila ima bit ‘acknowledge/noacknowledge’. Če je bil naslovni ali podatkovni okvir uspešno prejet je pošiljatelju vrnjen bit ACK, sicer NACK.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

□ Povezava dveh naprav:

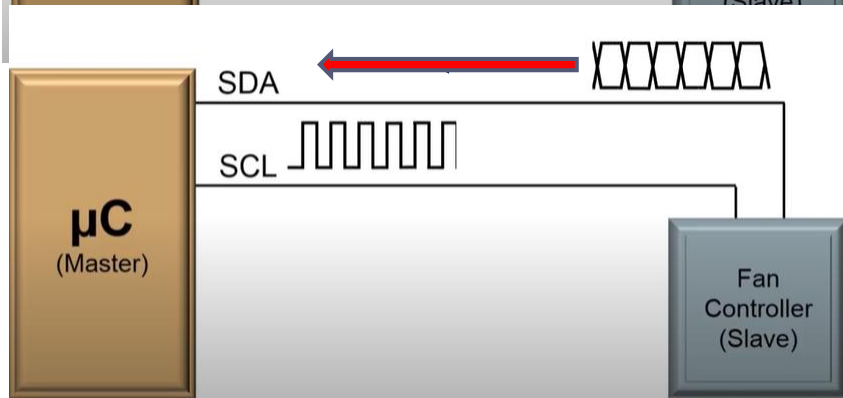
- Mikrokrmilnik (Master)
- Krmiljenje ventilatorja (Slave)



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

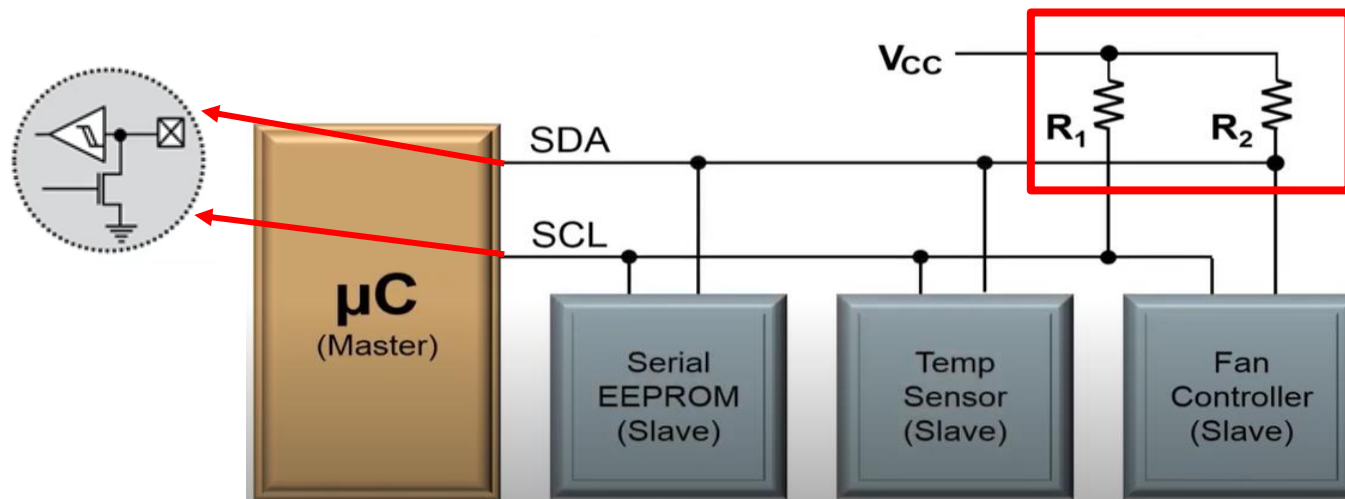


- Master pošilja urin signal (SCL)
- Master **bere iz** naprave Slave



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

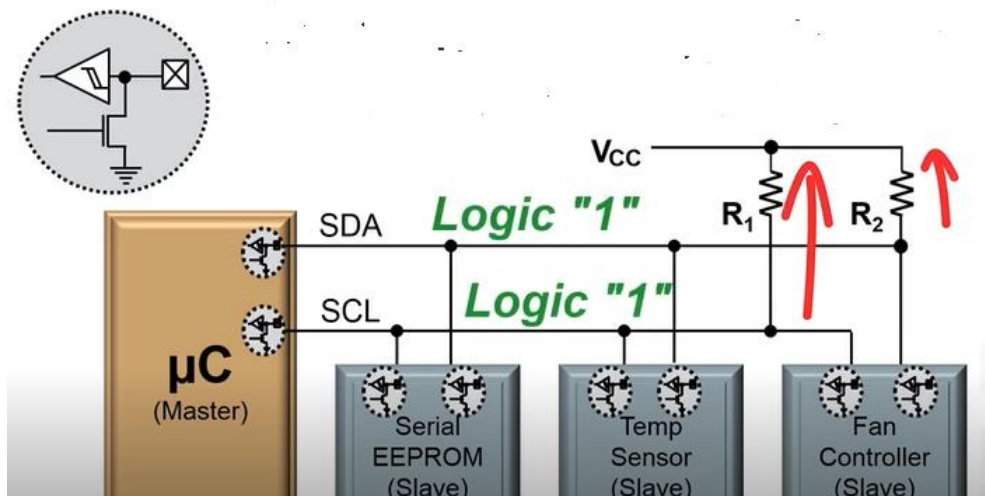
- Vsaka naprava Slave ima svoj naslov
- Za pravilno delovanje sta pri povezavi naprav na vodilo I2C upora R_1 in R_2 ('pull-up').



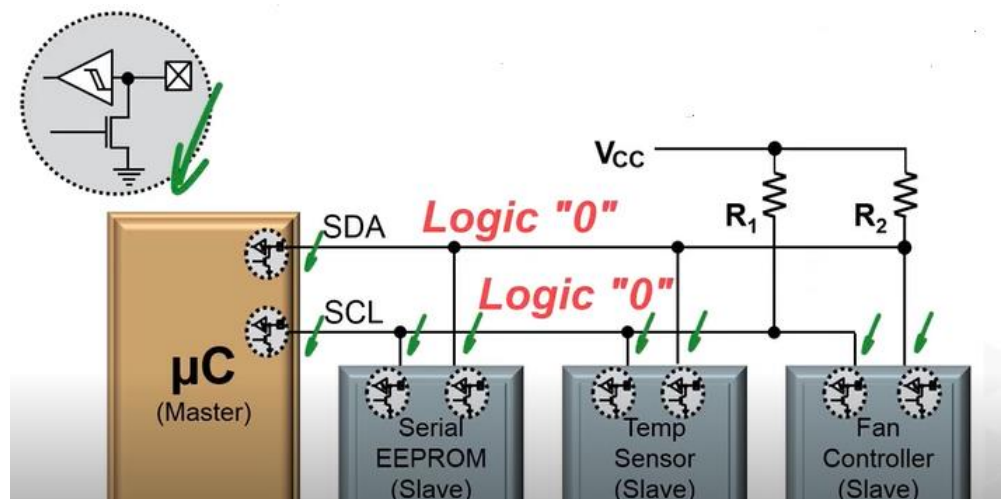
- Protokol I2C deluje na predpostavki, da sta liniji SDA in SCL 'open drain' ali 'open collector'
 - Katerakoli naprava zagotovi nizek nivo (low) na linijah, ne more pa visokega (high).
 - Vsaka linija ima zato dodan upor, da privzeto ohranja visok nivo.
 - To omogoča, da ne pride do kratkega stika,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

□ Logični nivoji:

- Logična 1



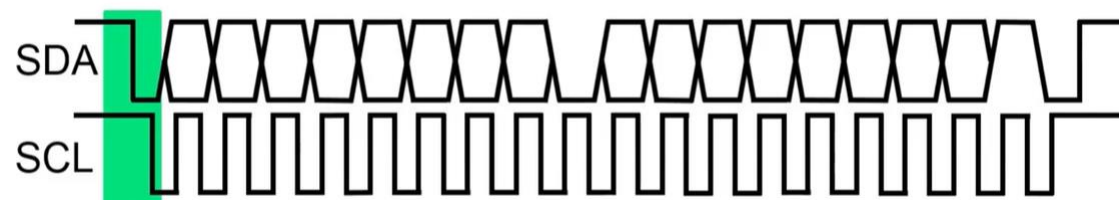
- Logična 0



□ I2C signal

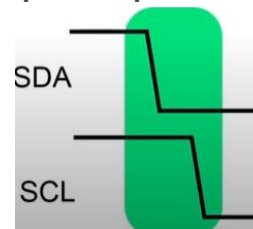


□ START

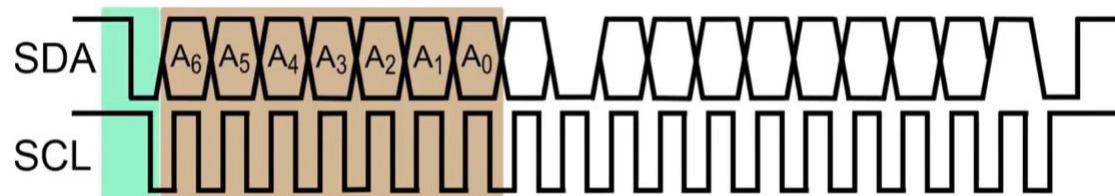


- Linija SDA preklopi iz 1 v 0 predno linija SCL preklopi iz 1 v 0

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

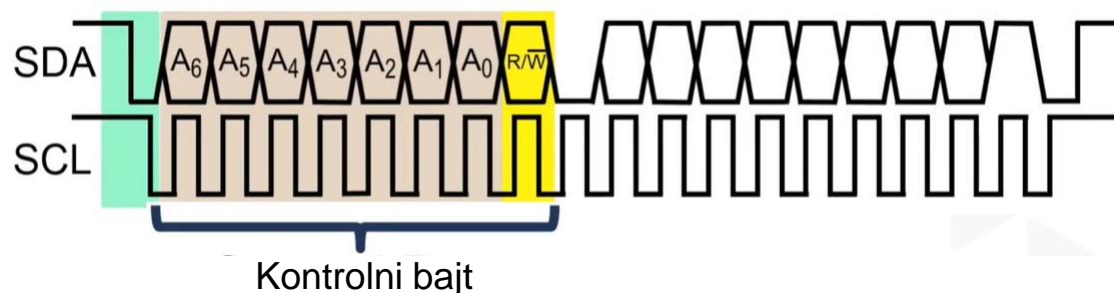


□ Naslov naprave Slave je 7-biten (najpogostejša dolžina)



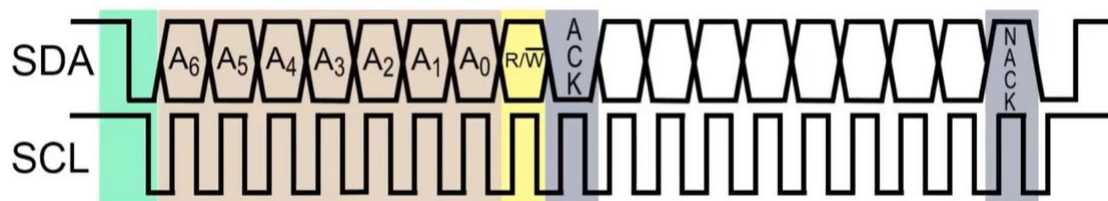
□ Branje/pisanje – R/\bar{W} (kontrolni bajt: $A_6 \dots A_0 R/\bar{W}$)

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

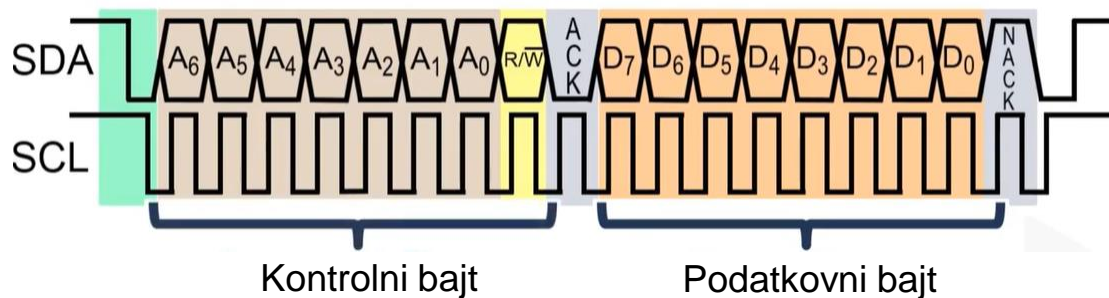


- kontrolni bajt: $A_6 \dots A_0 R/\bar{W}$

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



Podatkovni bajt ($D_7 \dots D_0$)



STOP



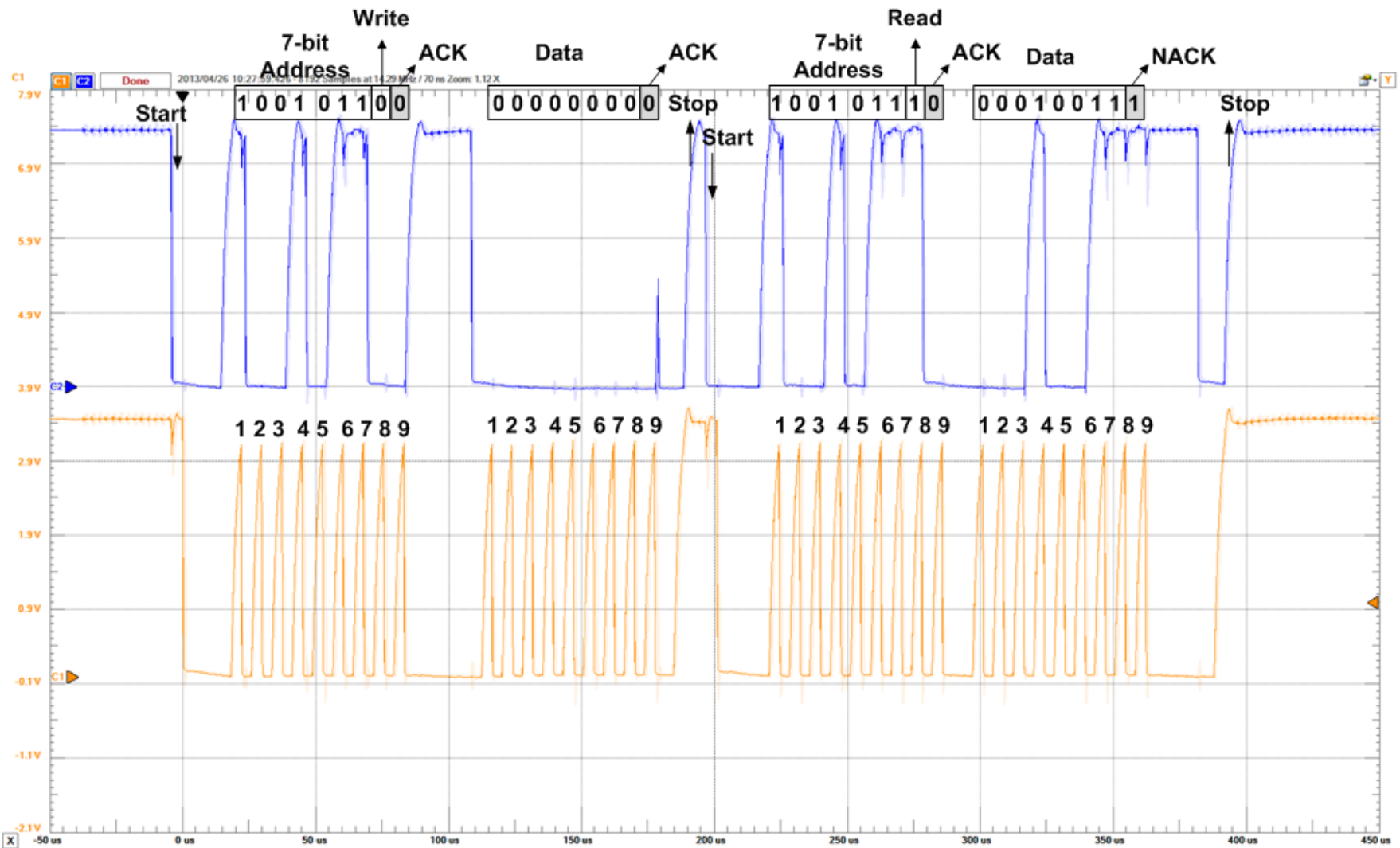
- Linija SDA preklopi iz 0 v 1 potem ko linija SCL preklopi iz 0 v 1

Master SDA – Low v High

Master SCL – drži High

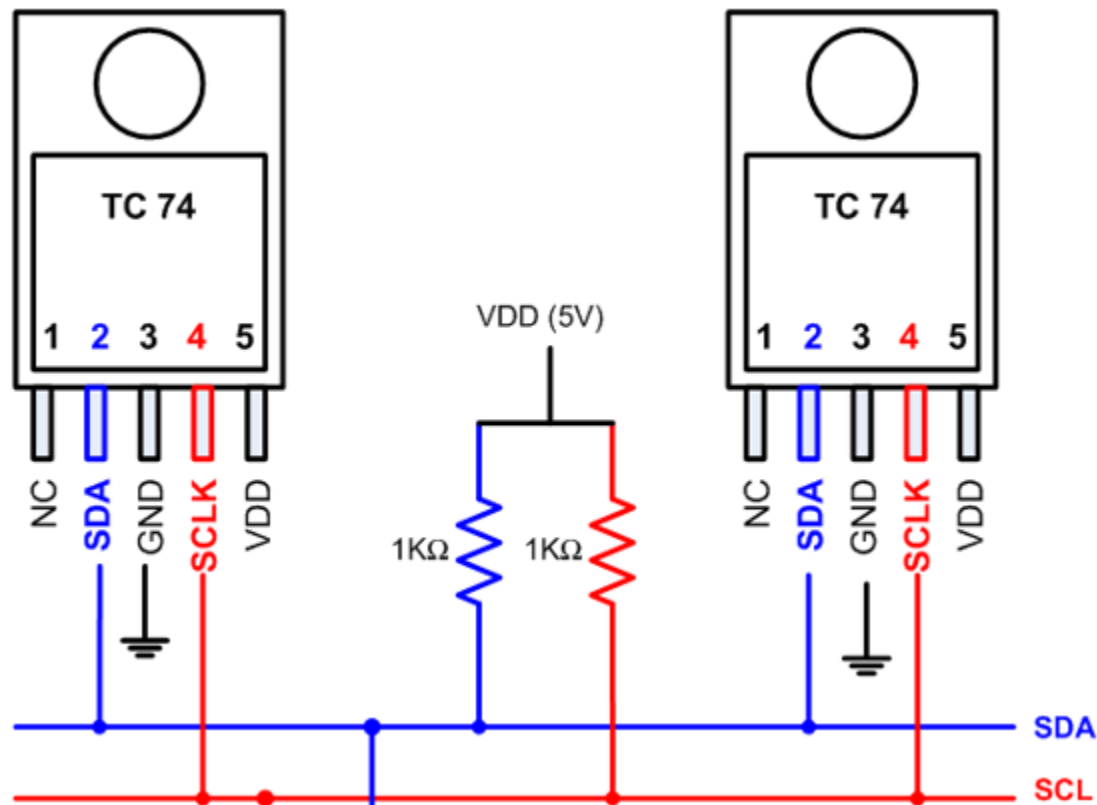


I2C – električno (osciloskop)



Primer I2C povezave

□ Digitalno temperaturno tipalo



Primer I2C komunikacije – STM32F4

```

/* USER CODE BEGIN PV */
HAL_StatusTypeDef retval;

uint8_t ChipID;

/* USER CODE END PV */
MX_I2C1_Init();

/* USER CODE BEGIN 2 */

```

```

// Set Reset Line to 1 (switch device on)
HAL_GPIO_WritePin(GPIOD, GPIO_PIN_4, GPIO_PIN_SET);

```

```

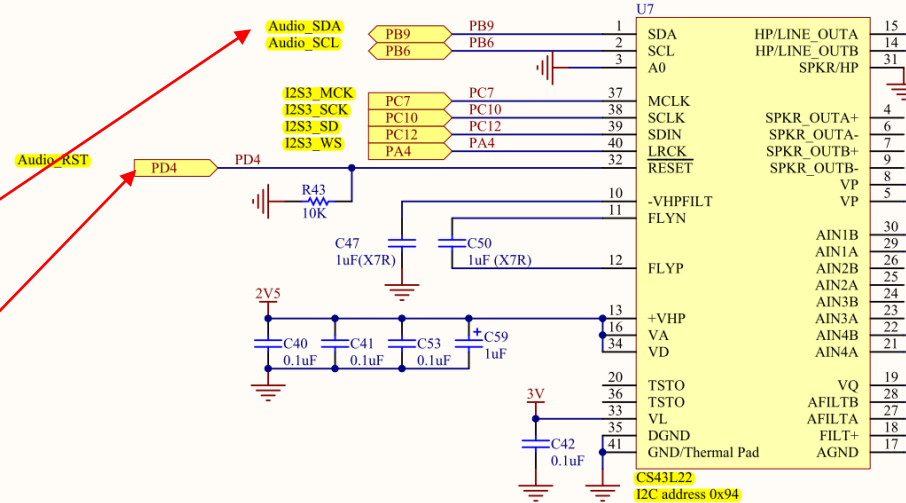
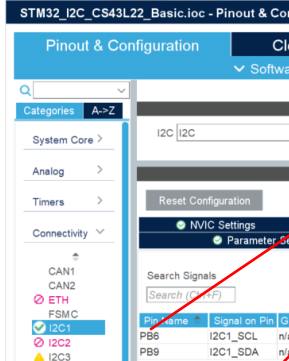
HAL_Delay(1000); // recomended by datasheet

```

```

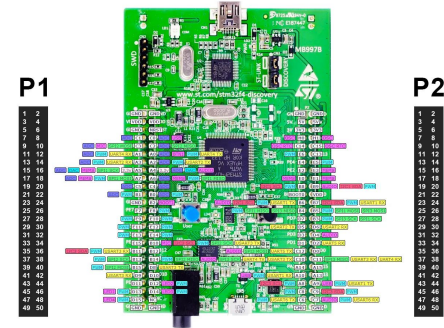
// From Device with address=0x94, Read register with address 0x01 and put value in ChipID
// DevAddress_0x94, tMemAddress=0x01, MemAddSize=8b, *pData,Size, Timeout);
retval = HAL_I2C_Mem_Read(&hi2c1, 0x94, 0x01, I2C_MEMADD_SIZE_8BIT, &ChipID, 1, 1000);

```



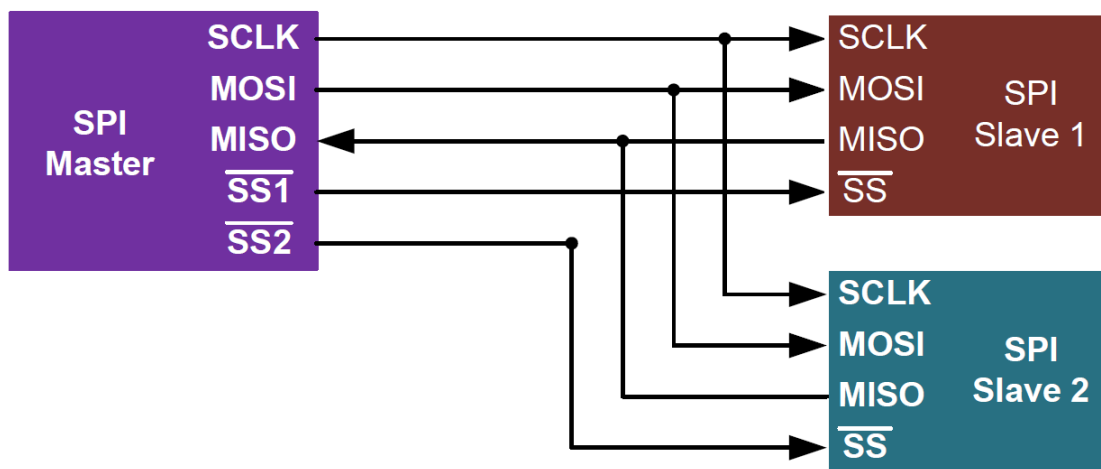
CS43L22

Low Power, Stereo DAC w/Headphone & Speaker Amps



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)



Video: Microchip: <https://www.youtube.com/watch?v=NyxQkGXbG6I>

- ❑ 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.

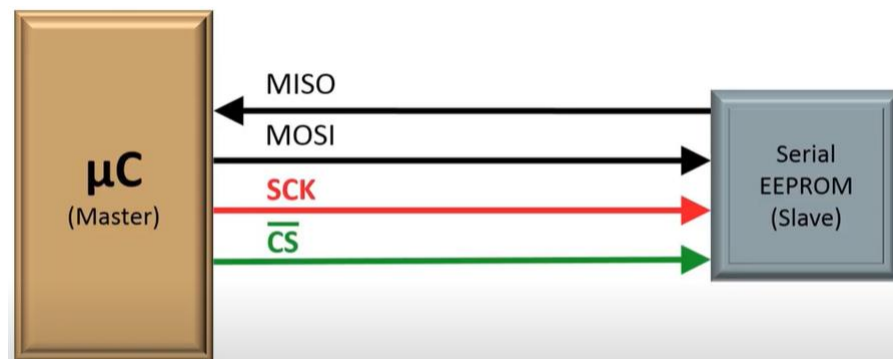
- ❑ Prenos podatkov poteka istočasno **v obe smeri (ang. full-duplex)**

- ❑ 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, ...

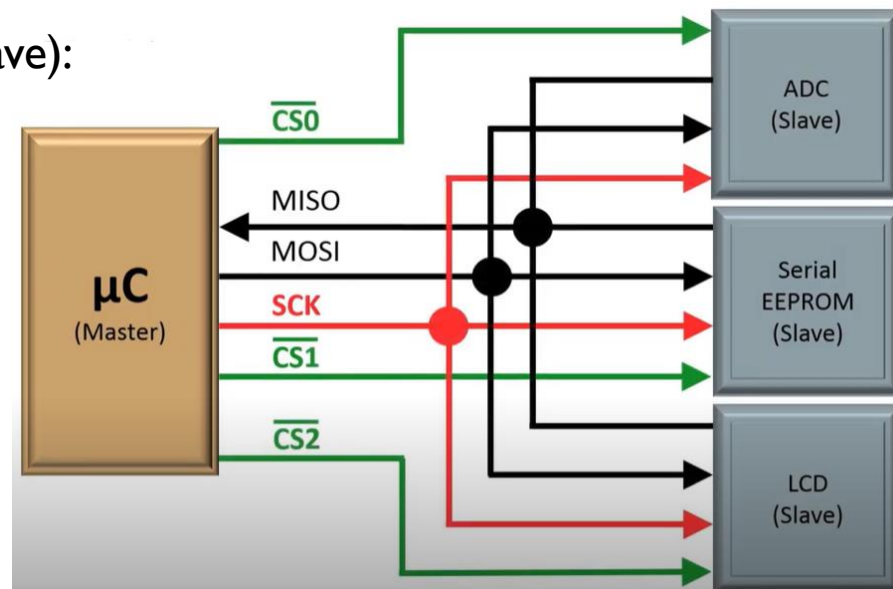
□ Vodilo SPI je vmesnik med dvema napravama, ki ima 4 povezave

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



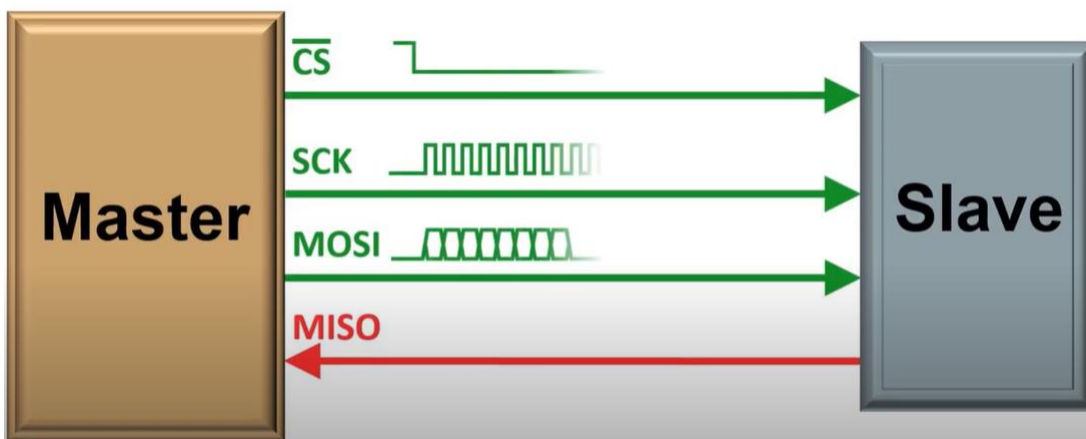
□ 4 naprave na vodilu SPI (Master + 3x Slave):

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

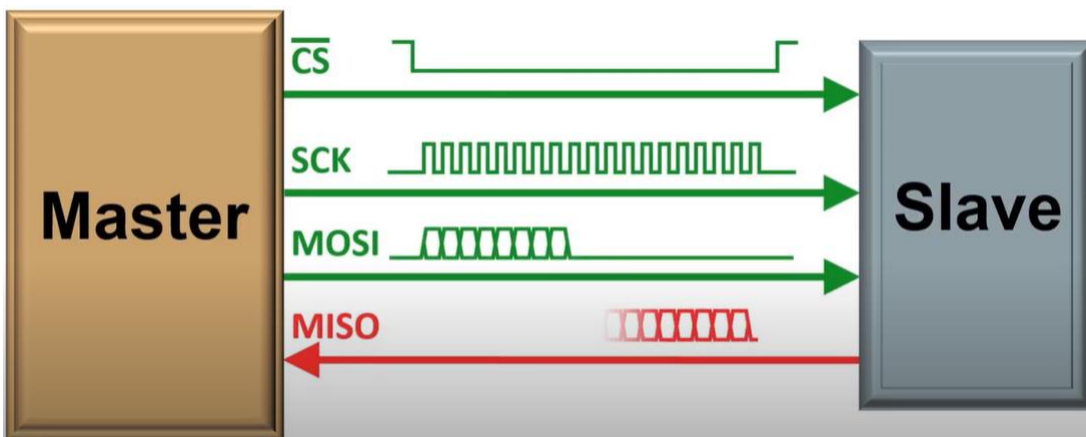


□ Signali SPI pri prenosu podatkov

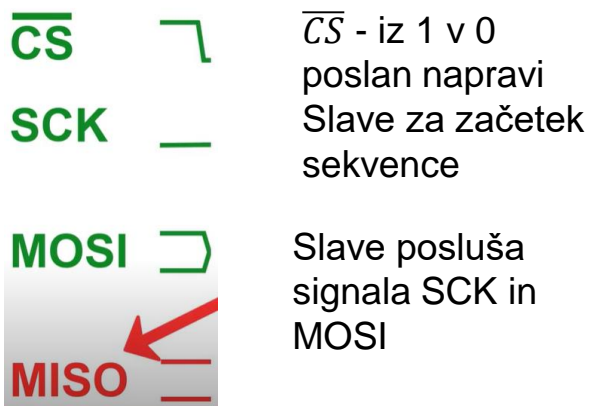
- Master generira signale \overline{CS} , SCK, MOSI



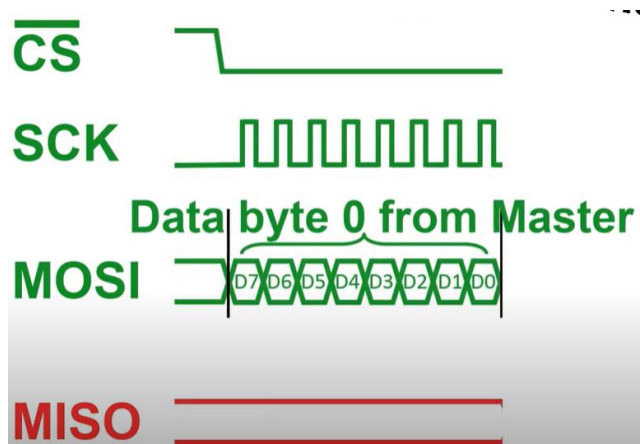
- Slave generira signal MISO



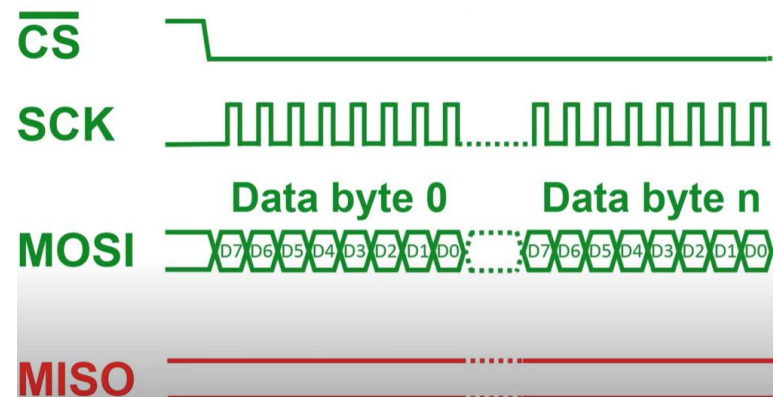
□ Serijski prenos podatkov iz naprave Master v Slave



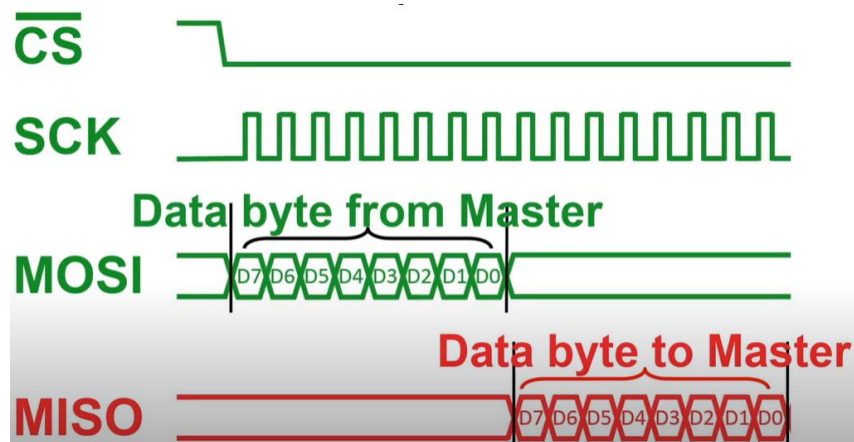
- Master prenese **bajt 0** po liniji MOSI



oz n bajtov se prenese

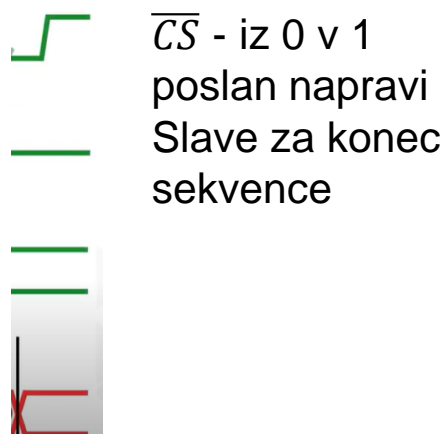


- Prenos podatkov Master to Slave in Slave to Master



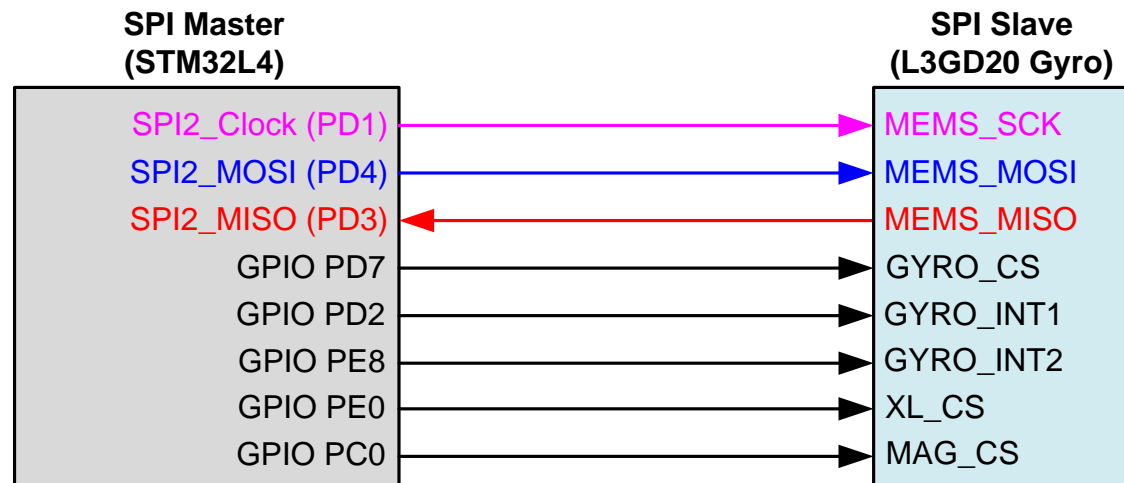
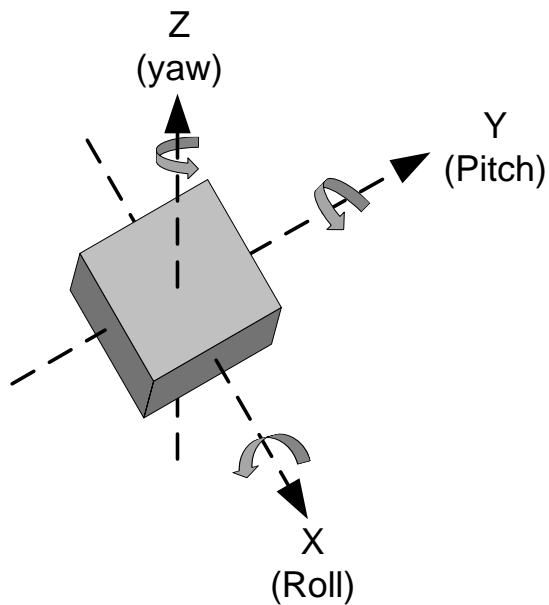
- Konec prenosa:

Chip Select gre iz Low v High



Primer SPI povezave

□ Primer žiroskopa (L3GD20)



Primerjava SPI vs I²C

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

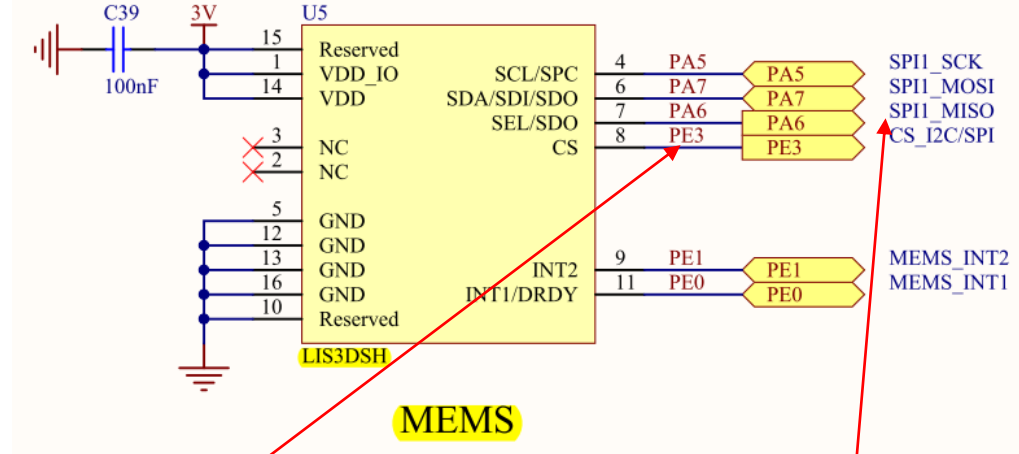
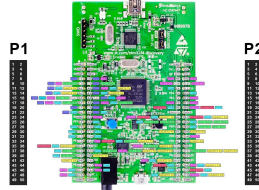
	SPI	I2C
Prednosti	<ul style="list-style-type: none">• Hitrejši (20Mbps+)• Full duplex• „Low power“• DualSPI, QuadSPI	<ul style="list-style-type: none">• Enostavnost (manj povezav, standard za 2 povezavi)• Dodajanje• Multi-Master

Primer SPI komunikacije – STM32F4

```

/* USER CODE BEGIN PV */
// Global variables
uint8_t indata[2];
uint8_t outdata[2] = {0,0};
uint8_t lis_id;
HAL_StatusTypeDef SPIStatus;
/* USER CODE END PV */

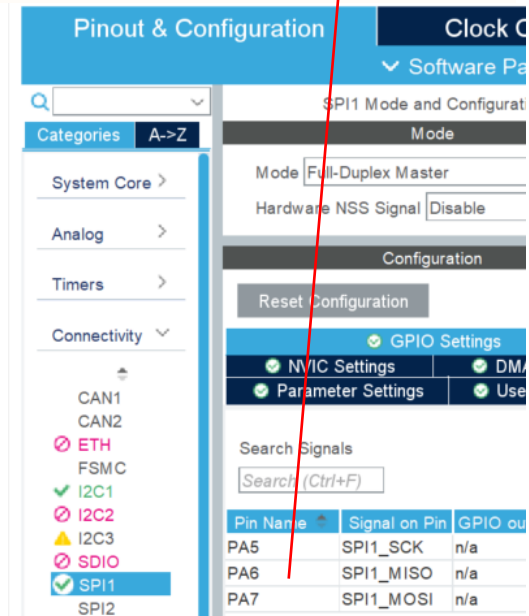
```



```

MX_SPI1_Init();
/* USER CODE BEGIN 2 */
// Config accelerometer
// Read WHOAMI register
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3, GPIO_PIN_RESET);
outdata[0] = 0x0f | 0x80 ; // read whoami
HAL_SPI_TransmitReceive(&hspi1, &outdata, &indata, 2, HAL_MAX_DELAY);
lis_id = indata[1];
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3, GPIO_PIN_SET);

```



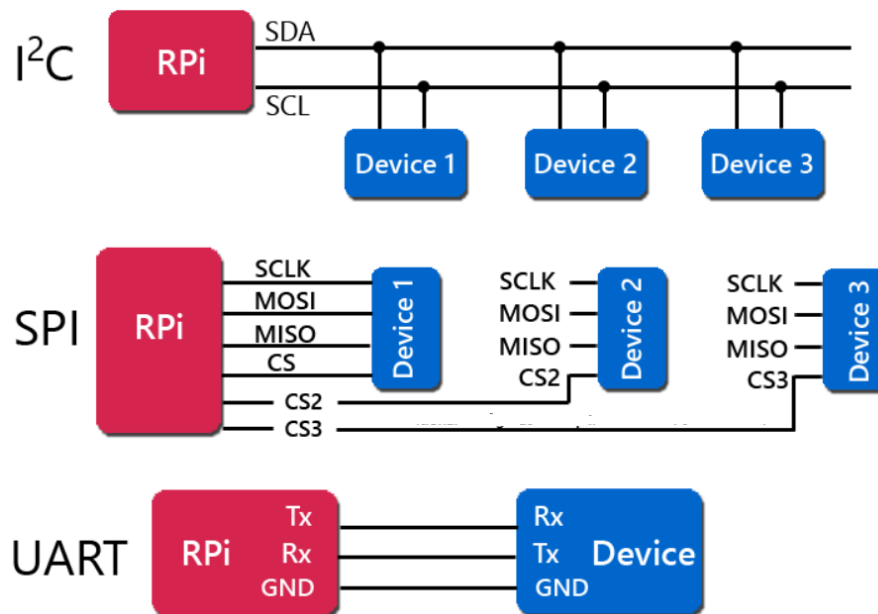
LIS3DSH

MEMS digital output motion sensor

ultra low-power high performance three-axis “nano” accelerometer

Primer: Raspberry Pi ↔ I²C, SPI, UART

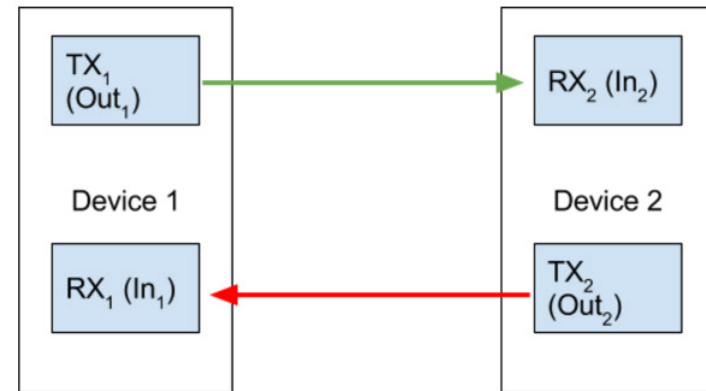
- Diagram povezav
R_{Pi} = Raspberry Pi



- **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>

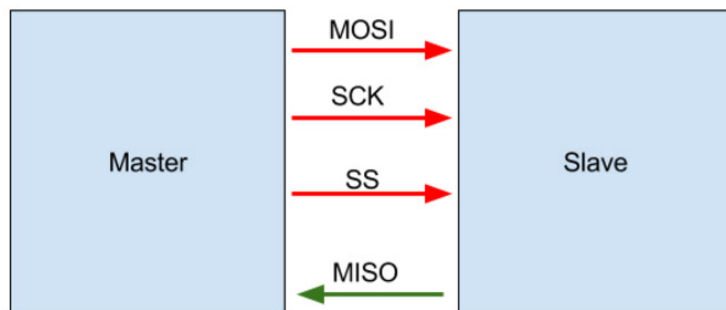
Primer: Arduino ↔ UART, SPI, I2C

❑ Asinhronski serijski prenos: **UART**

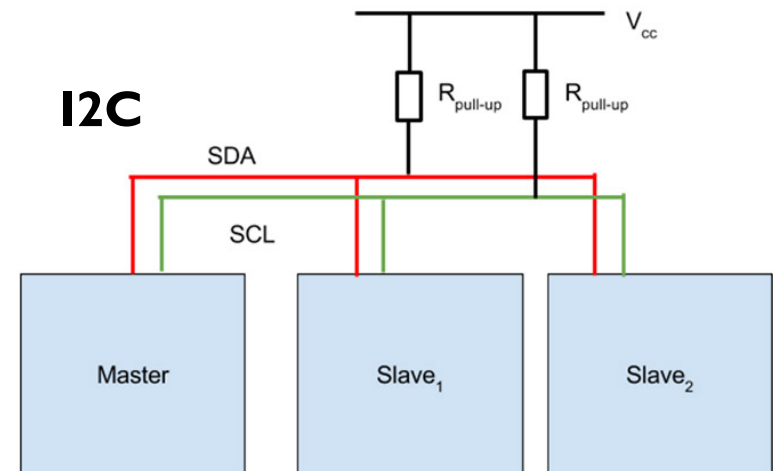


❑ Sinhronski serijski prenos:

SPI



I2C



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