

Abstract geometric lines in the top-left corner of the slide, consisting of several overlapping, tilted rectangles and polygons.

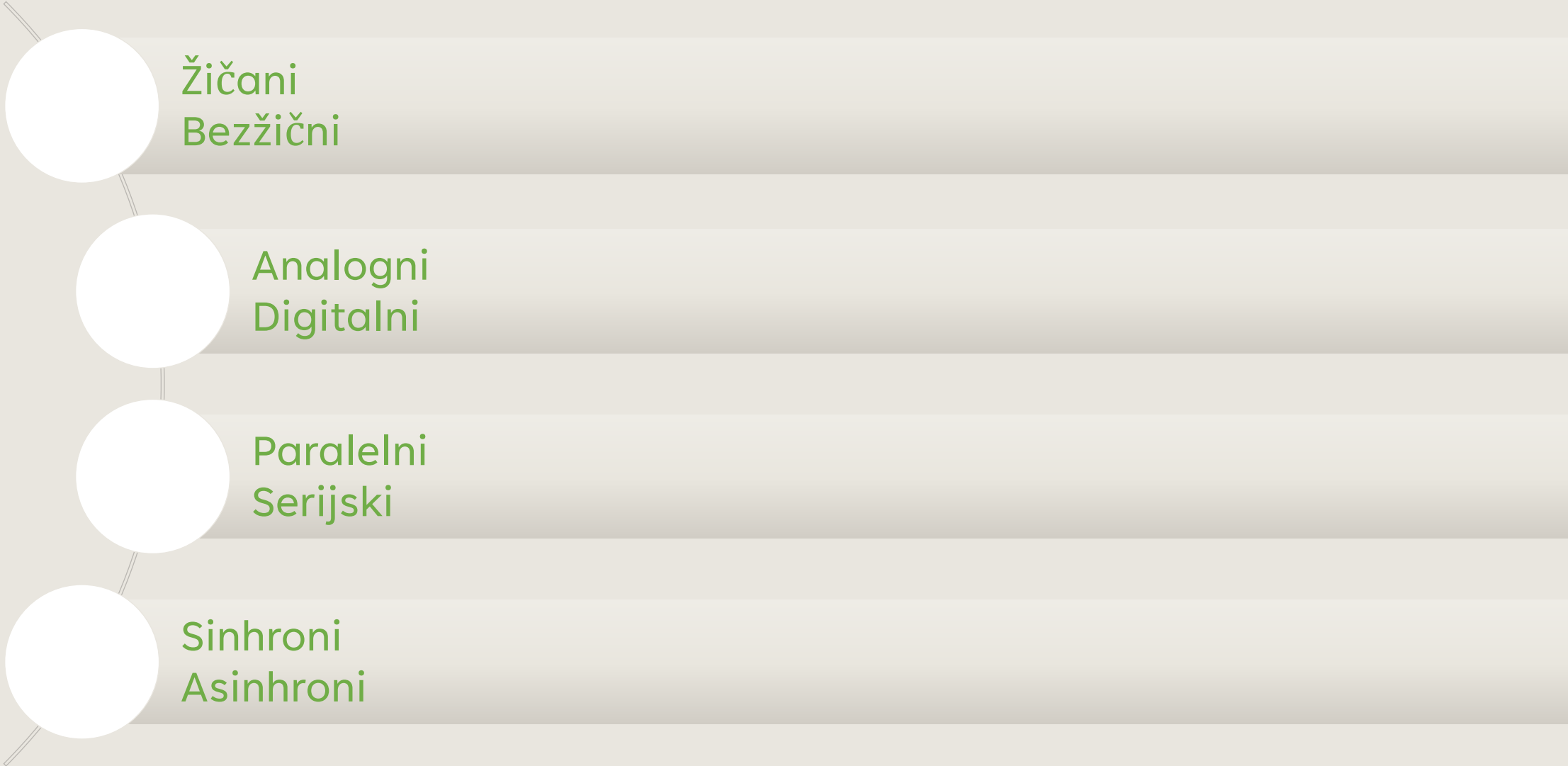
INTERNET PAMETINI UREĐAJA

prof. dr Dejan S. Aleksić

Prirodno-matematički fakultet, Niš

03. PRENOS PODATAKA

PRENOS SIGNALA/PODATAKA



Žičani
Bezžični

Analogni
Digitalni

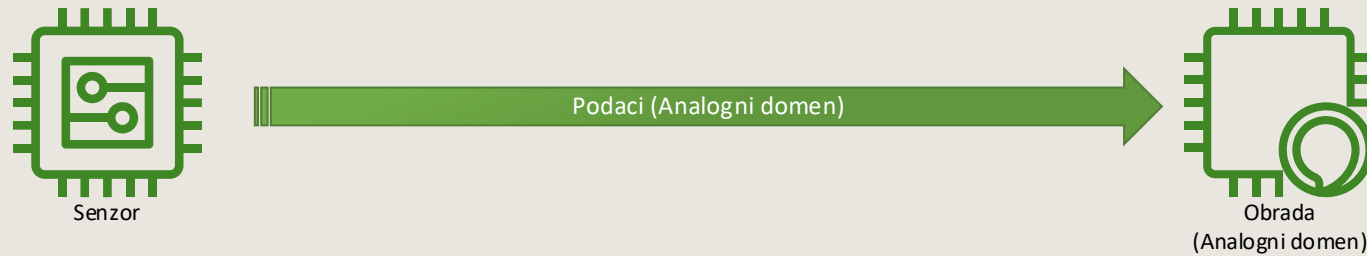
Paralelni
Serijski

Sinhroni
Asinhroni

PRENOS SIGNALA/PODATAKA

Prenos signala/podataka od senzora do mesta obrade može biti:

- Analogni

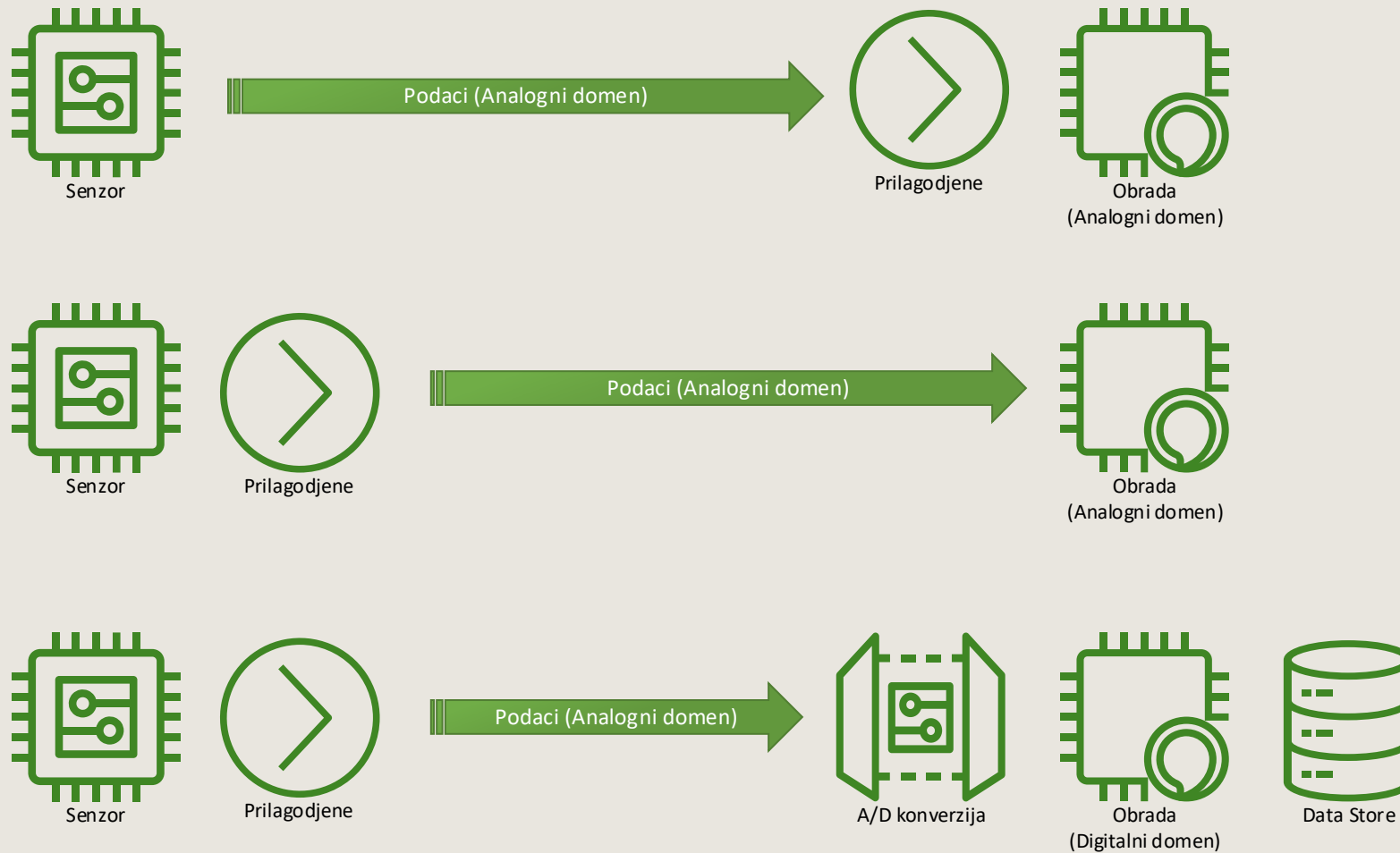


- Digitalni



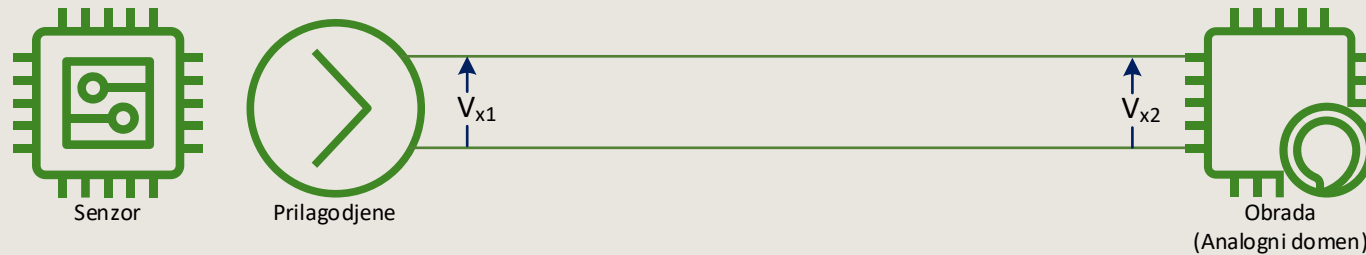
PRENOS SIGNALA/PODATAKA

Analogni prenos signala od senzora do mesta obrade:

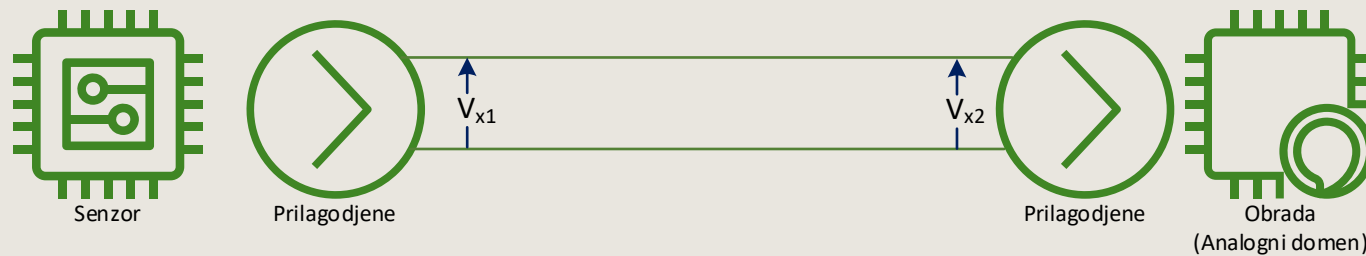


PRENOS SIGNALA/PODATAKA

- Problemi pri prenosu analoginih signala na daljinu:



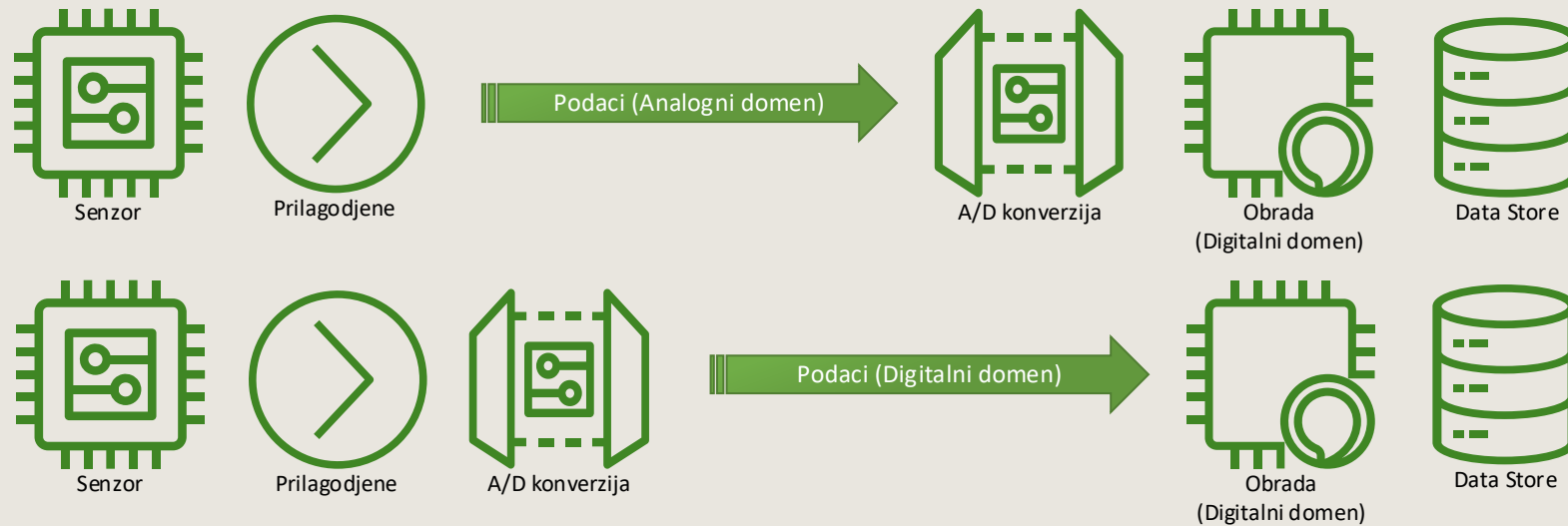
- Slabljenje signala zbog poduzne otpornosti linije
- Slaba otpornost na smetnje
- Pojačavanje amplitude šuma na strani obrade
- Obrada signala u analognom domenu



- Dobre strane analognog prenosa signala na daljinu:
 - Brzina prenosa -

PRENOS SIGNALA/PODATAKA

Digitalni prenos signala/podataka od senzora do mesta obrade:

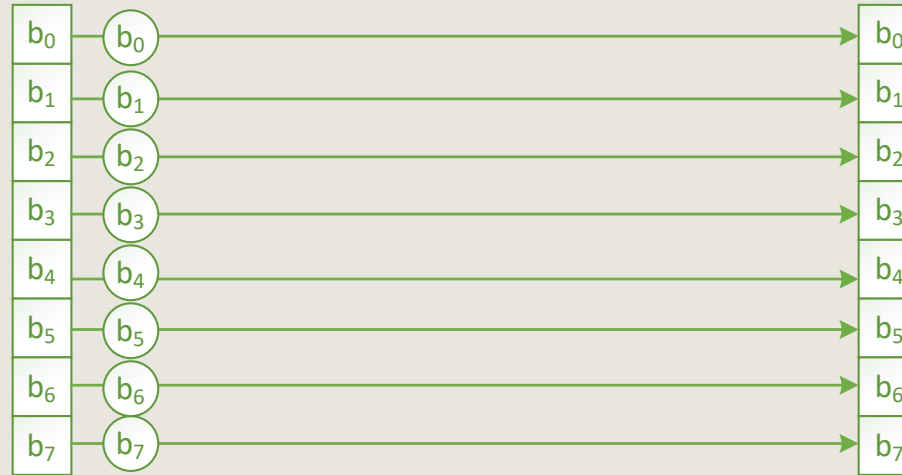


- Slabljenje signala zbog poduzne otpornosti linije
- Slaba otpornost na smetnje
- Pojačavanje amplitude šuma na strani obrade
- Kašnjenje faze signala
- Obrada signala u analognom domenu
- Brzina prenosa podataka

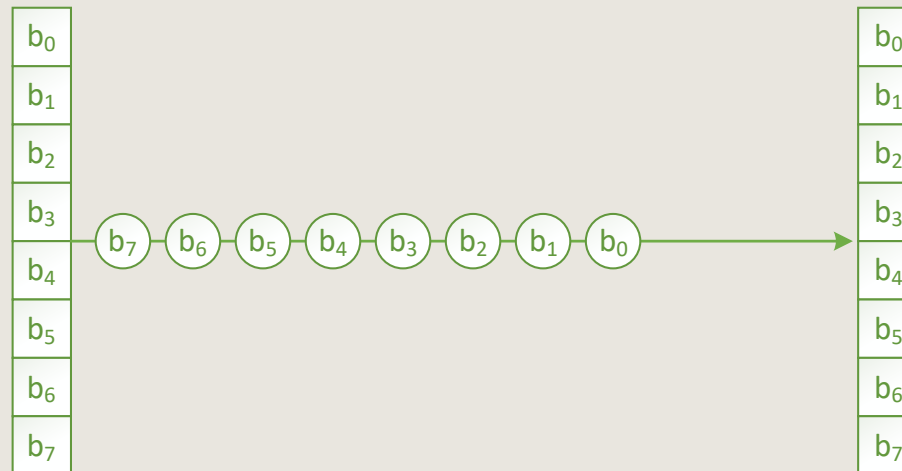
PRENOS SIGNALA/PODATAKA

Digitalni prenos signala/podataka od senzora do mesta obrade može biti:

- Paralelni

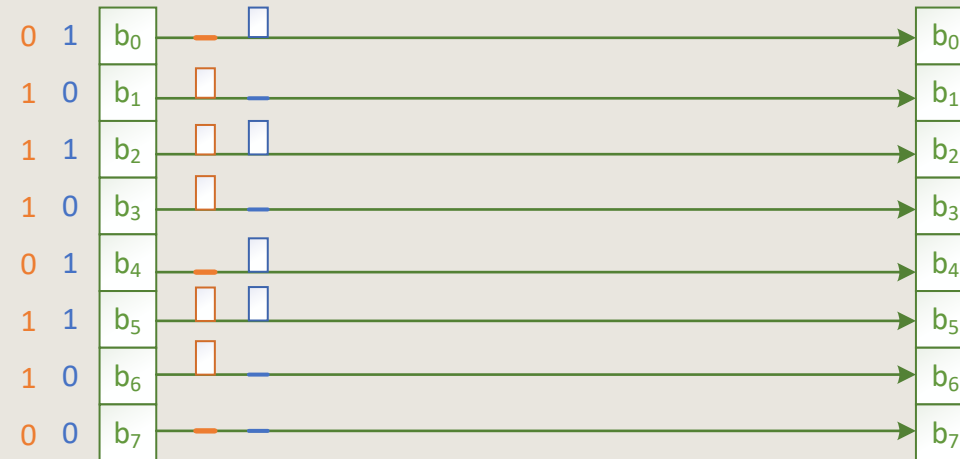


- Serijski



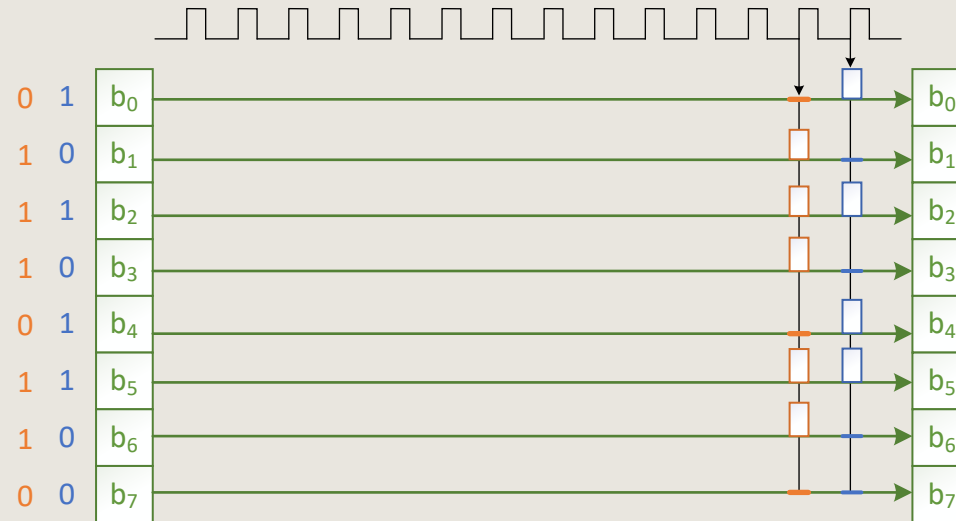
PRENOS SIGNALA/PODATAKA

Paralelni prenos podataka:



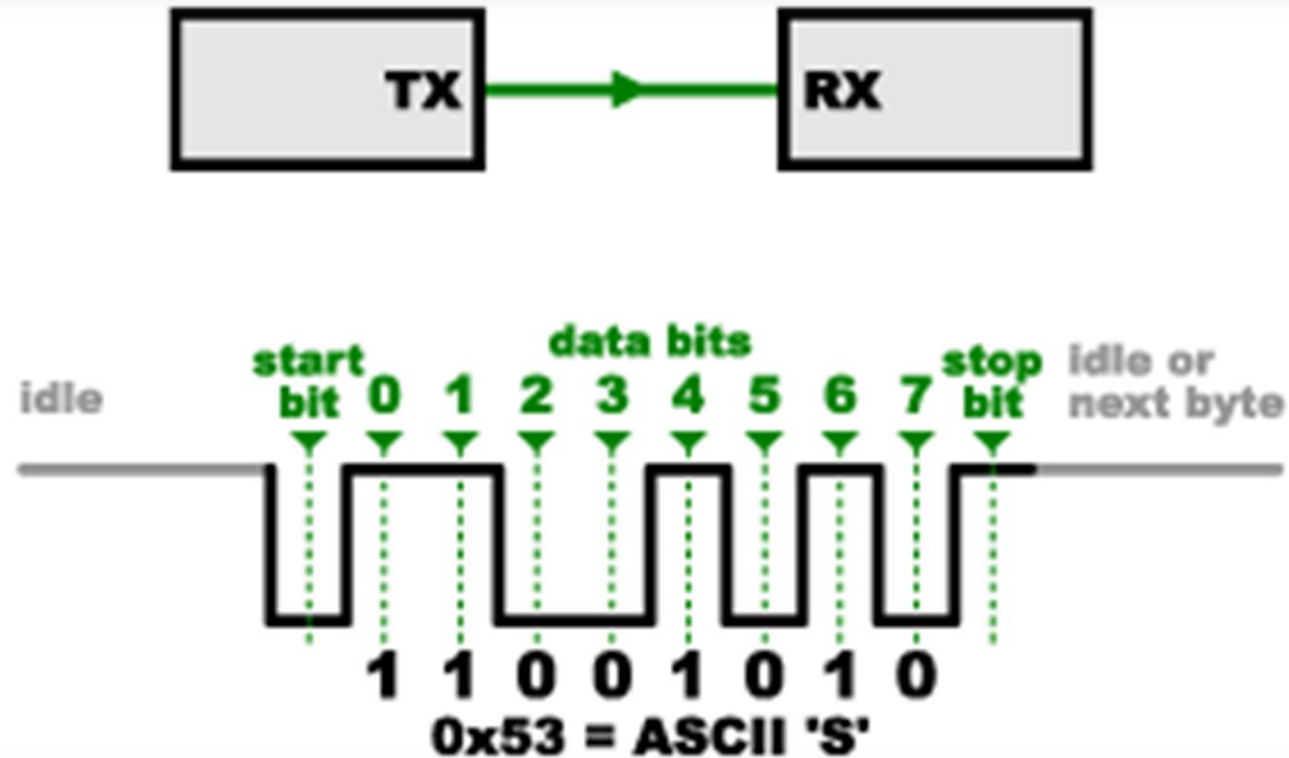
PRENOS SIGNALA/PODATAKA

Paralelni prenos podataka:



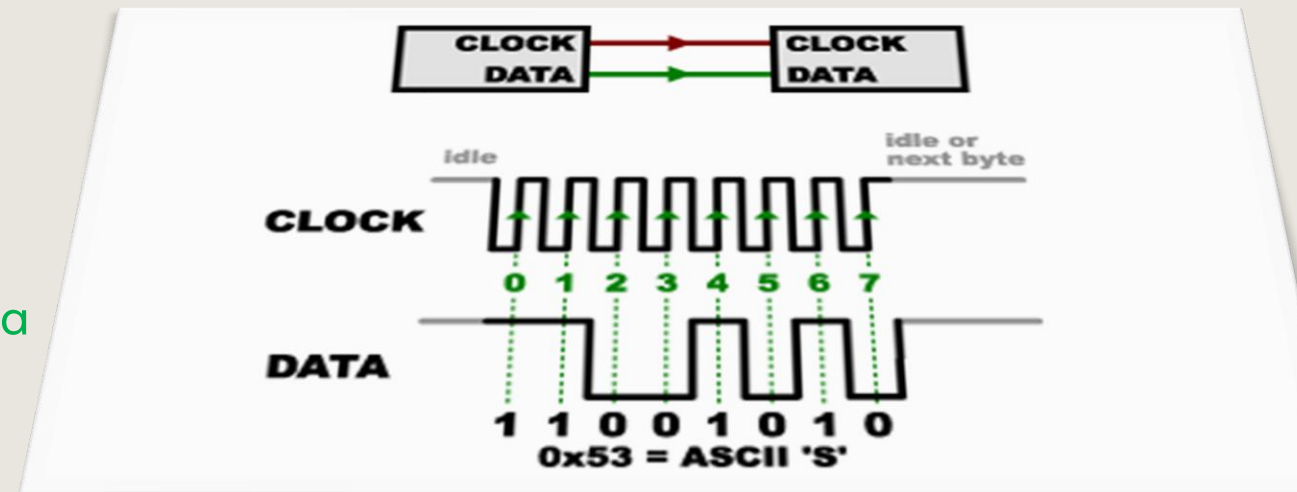
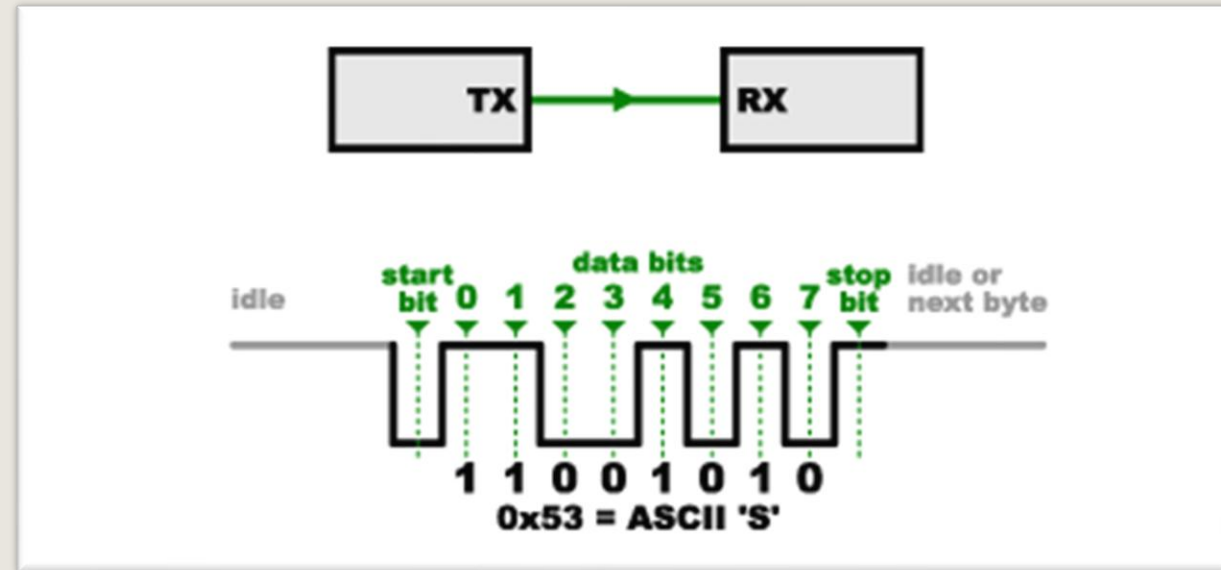
SINHRONI I ASINHRONI PRENOS SIGNALA/PODATAKA

- Pre nego što nastavimo, moramo razumeti dva fundamentalna načina za slanje serijskih podataka.
- Problem je: Kako prijemnik zna kada da očita bit?



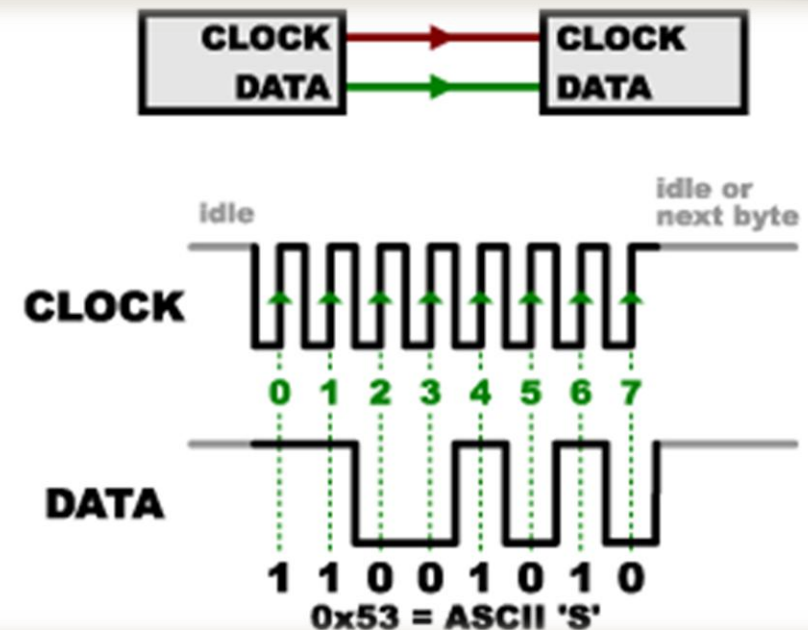
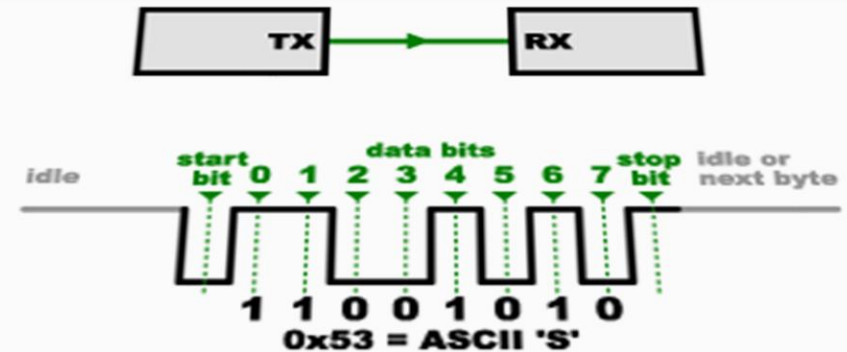
ASINHRONI PRENOS SIGNALA/PODATAKA

- Ključna ideja: Nema zajedničkog takta (clock-a).
- Kako radi?
 - Obe strane se unapred dogovore o brzini (npr. 9600 bita u sekundi - "Baud Rate").
 - Linija je u mirovanju na '1' (HIGH).
 - Pošiljalac šalje specijalni "Start Bit" (prebacuje liniju na '0').
 - Prijemnik detektuje Start Bit i pokreće svoju "štopericu". On onda očitava liniju u tačno definisanim intervalima (na osnovu dogovorene brzine).
 - Na kraju se šalje "Stop Bit" (vraća liniju na '1') da signalizira kraj paketa.
- Primeri: UART (koji koristi RS-232, RS-485), modemi.
- Pros: Jednostavnije, treba manje žica (nema žice za clock).
- Cons: Osetljivost na greške u tajmingu. Ako "štoperice" nisu savršeno usklađene, može doći do greške. Manje brzine.



SINHRONI PRENOS SIGNALA/PODATAKA

- Ključna ideja: Postoji posebna linija za takt (Clock) koju generiše Master.
- Kako radi?
 - Postoji linija za podatke (Data) i linija za takt (Clock ili SCK).
 - Pošiljalac (Master) postavi bit na Data liniju.
 - Master zatim generiše "puls" na Clock liniji.
 - Prijemnik (Slave) ne radi ništa dok ne vidi puls na Clock-u. Kada vidi puls, tačno u tom trenutku očitava vrednost sa Data linije.
- Primeri: SPI, I2C.
- Pros: Veoma velike brzine, pouzdano, nema grešaka u sinhronizaciji.
- Cons: Zahteva najmanje jednu dodatnu žicu za takt.



STANDARD RS-232 I PROTOKOL UART

- RS-232 (Standard): Definiše fizički sloj – pinove i naponske nivoe na linijama.

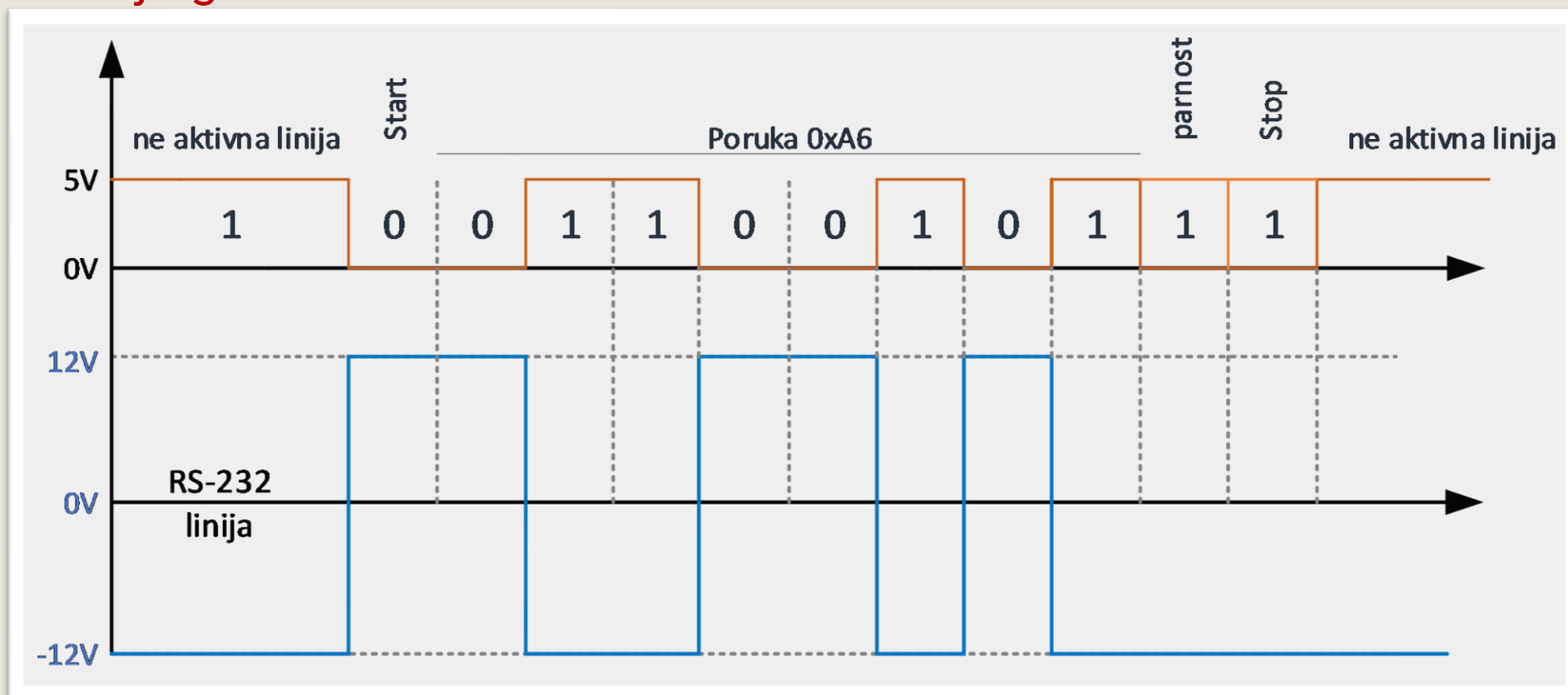


STANDARD RS-232 I PROTOKOL UART

❑ Dva fizička stanja na liniji za prenos:

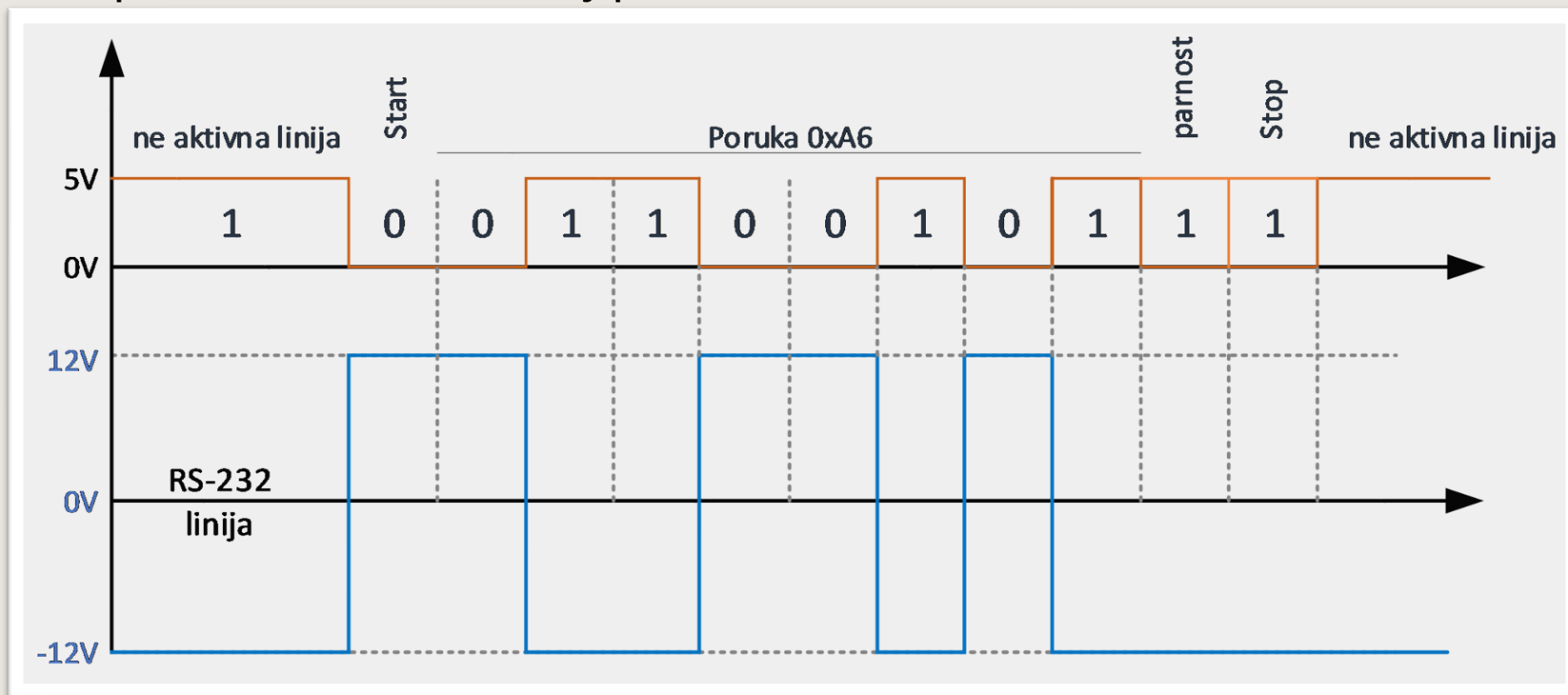
- ✓ ON stanje, logička 1, -12V
- ✓ OFF stanje, logička 0, +12V
- ✓ Kad nema transfera poruke linija je na -12V (stanje logičke 1)
- ✓ Single-Ended: Svi naponi (TX i RX) se mere u odnosu na zajedničku masu (GND).

Ovo je glavna mana.



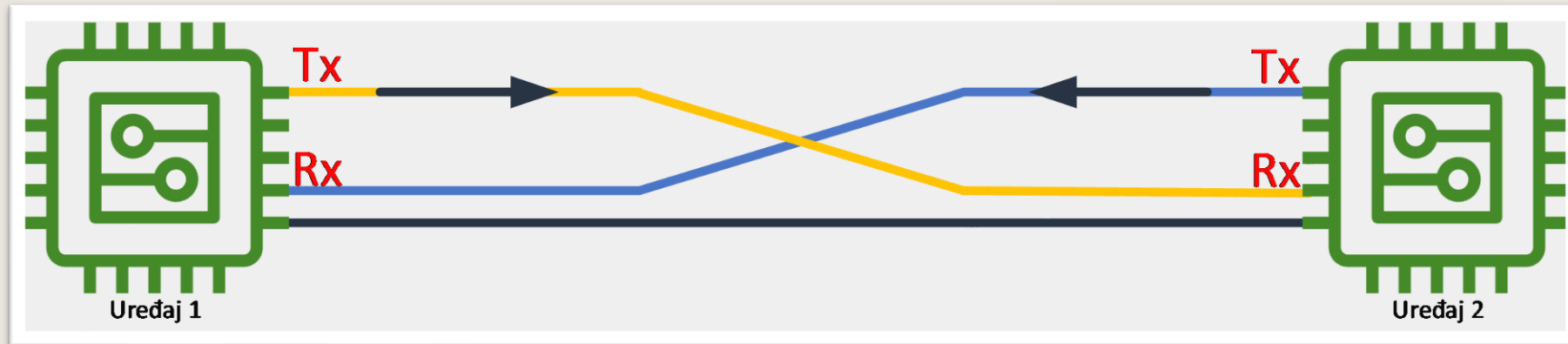
STANDARD RS-232 I PROTOKOL UART

- ❑ UART (Protokol): Asinhrona komunikacija. Definiše format podataka.
- Start bit – omogućava da prijemnik otkrije početak predaje poruke.
- Bitovi podataka – prvo se šalje bit najmanje težine
- Bit parnosti – bez pariti bita, parna parnost, neparna parnost
- Stop bit – označava kraj poruke

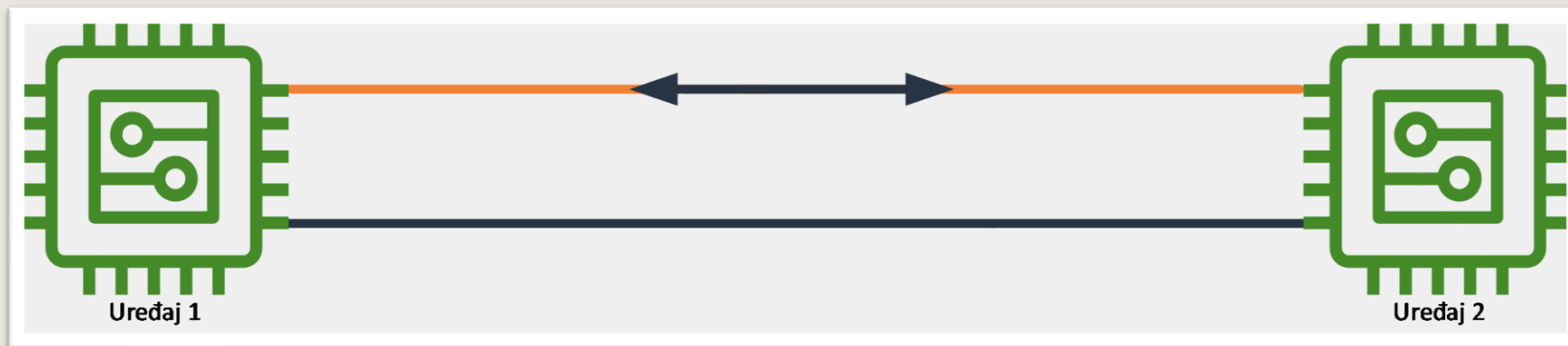


STANDARD RS-232 I PROTOKOL UART

Full duplex veza

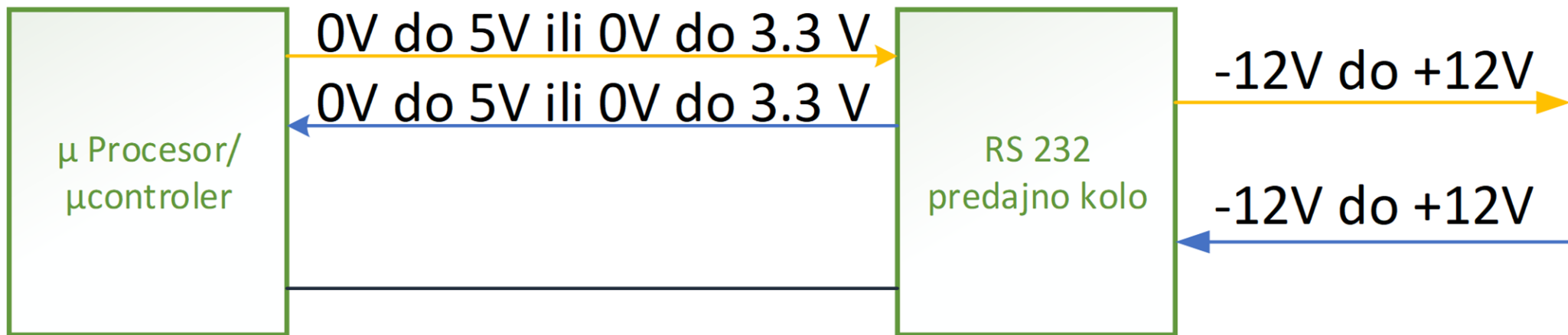


Half duplex veza – master/slave



STANDARD RS-232 I PROTOKOL UART

- Mikroprocesori i mikrokontroleri šalju i primaju signale sa naponskom nivoima 0V /+5V ili 0V/+3.3V
- RS-232 prijemna i predajna kola (driver MAX232) menjaju nivo i invertuju logiku ovih signala na -12V / +12V.



STANDARD RS-232 I PROTOKOL UART

❑ Prednosti:

- Jednostavan (UART), Full-Duplex.

❑ Nedostaci:

- Komunikacija samo između jednog predajnika i jednog prijemnika
- Brzina (max 20 kb/s)
- Dužina kabla (<15 m)
- Single-Ended: Otpornost na smetnje/Osetljiv na šum.

❑ Zaključak

- RS-232 je postavio temelje asinhronne komunikacije, ali njegove mane (domet, šum) su dovele do potrebe za nečim boljim...
- a to je RS-485, koji također koristi UART (dakle, i dalje je asinhron), ali na mnogo robusnijem fizičkom sloju.

STANDARD RS-232 I PROTOKOL UART

Primer Koda za ESP32 (UART/RS-232):

- Koristi ugrađenu Serial biblioteku.
- Prima podatke sa računara (preko RS-232 konvertera poput MAX232) i šalje ih nazad.
- Potreban hardver: ESP32 razvojna ploca + MAX232 modul + RS-232 kabel do PC-a (ili USB-to-RS232 adapter).

```
void setup() {  
    Serial.begin(9600);    // Baud rate 9600, standard za RS-232  
    delay(100);           // Čekaj da se serial inicijalizuje  
}  
  
void loop() {  
    if (Serial.available()) {    // Ako su podaci dostupni  
        char data = Serial.read();    // Čitaj karakter  
        Serial.print(data);    // Pošalji ga nazad (echo)  
    }  
}
```

RS-485 STANDARD

- ❑ RS-485 standard - Kičma industrijske komunikacije
 - Domet, otpornost na smetnje i više uređaja
 - Često korišćen serijski protokol u sistemima za akviziciju i prenos podataka
 - Za kontrolne aplikacije koje rade u realnom vremenu
 - Razmena podataka između više udaljenih čvorova

- ❑ Ključna poruka: RS-485 je standard za fizički sloj (ne protokol) koji rešava probleme RS-232.

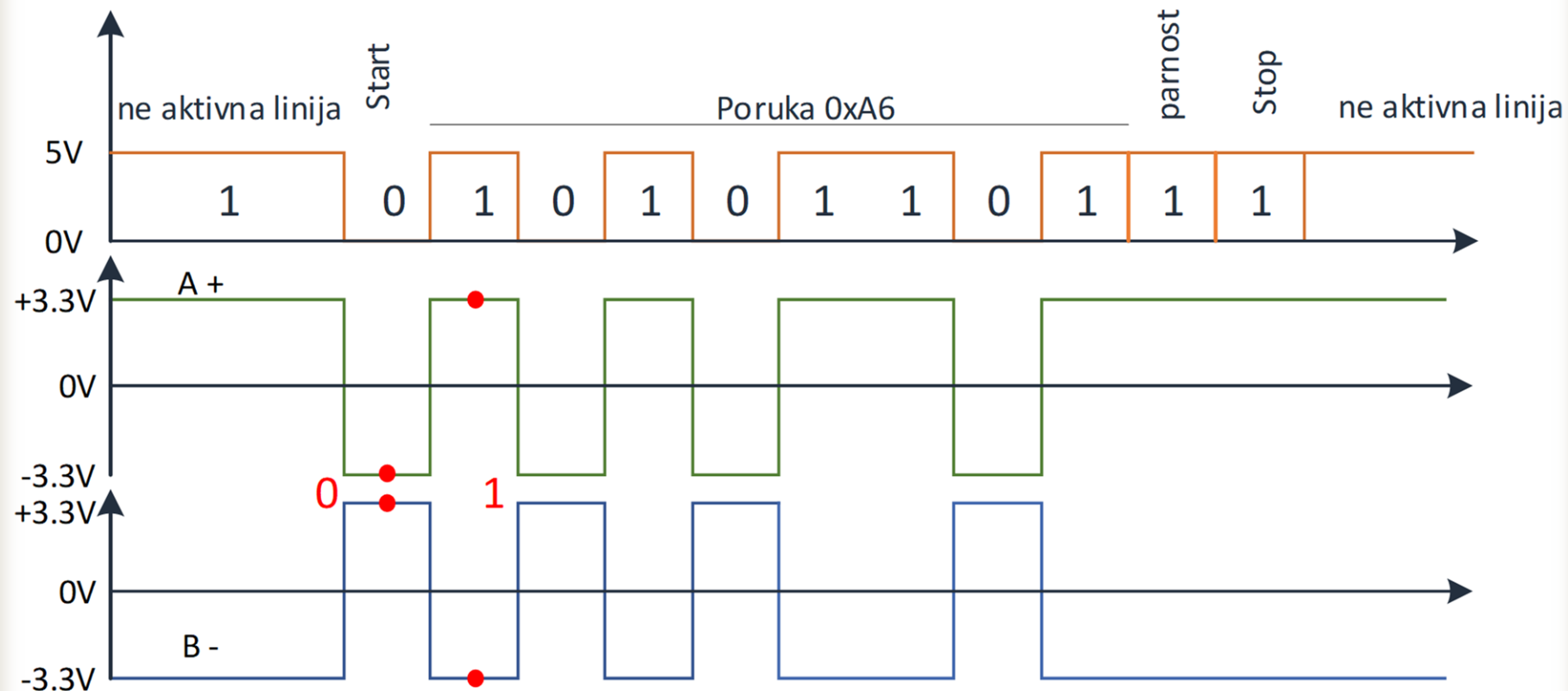
RS-485 STANDARD

- ❑ Half-duplex asinhrona veza - (potrebna kontrola smera DE/RE)
- ❑ Podaci mogu da se prenose u oba smera ali ne u isto vreme
- ❑ Za prenos se koriste oklopljeni kablovi tj. dve oklopljene upletene parice: A + i B -
- ❑ Glavni koncept: **Diferencijalni prenos**.
 - Umesto merenja signala u odnosu na GND (kao RS-232), RS-485 meri razliku napona između dve žice (A i B).
 - Ovo ga čini izuzetno otpornim na šum (Common-Mode Rejection).



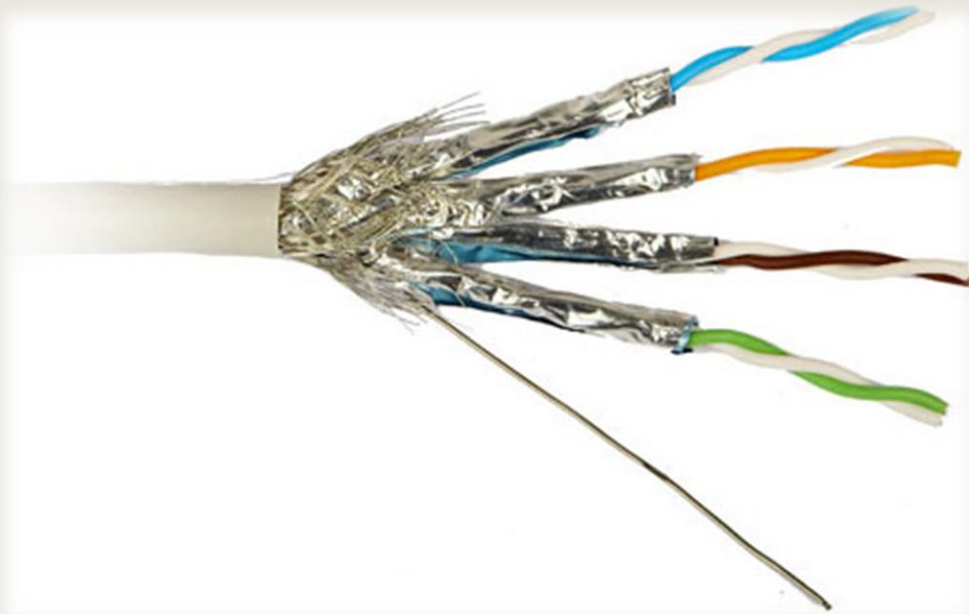
RS-485 STANDARD

- Na prijemu se uvek računa razlika između napona na A+ i B- tj. **A-B**
- Prenos logičke „0“: A+ = -3.3V a B- = +3.3V na prijemu: $A-B = -3.3V - 3.3V = -6.6V$
- Prenos logičke „1“: A+ = +3.3V a B- = -3.3V na prijemu: $A-B = 3.3V + 3.3V = +6.6V$
- **Diferencijalni prenos**

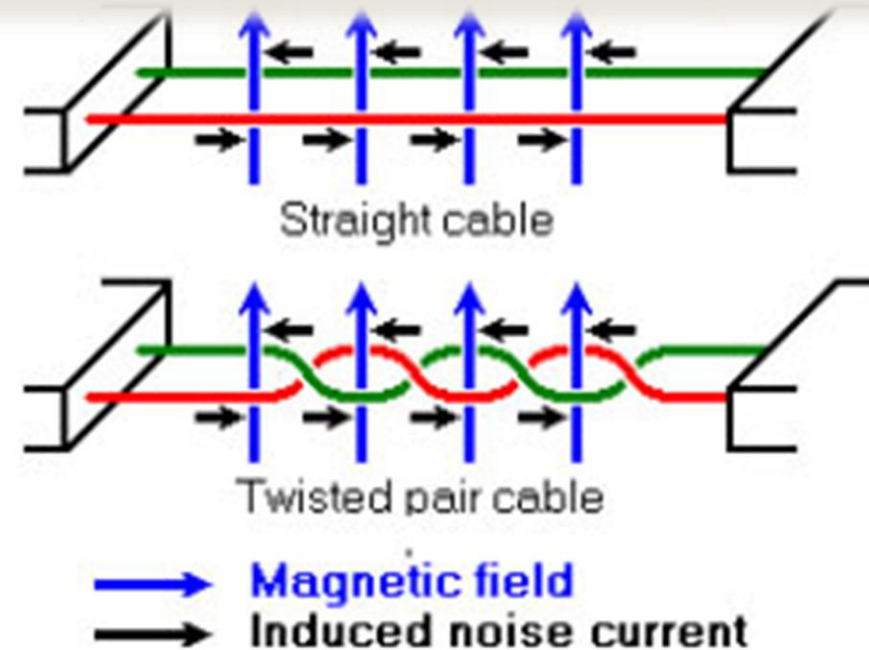


RS-485 STANDARD

- Diferencijalni prenos
- Otpornost na smetnje pri prenosu
- Faradejev kavez



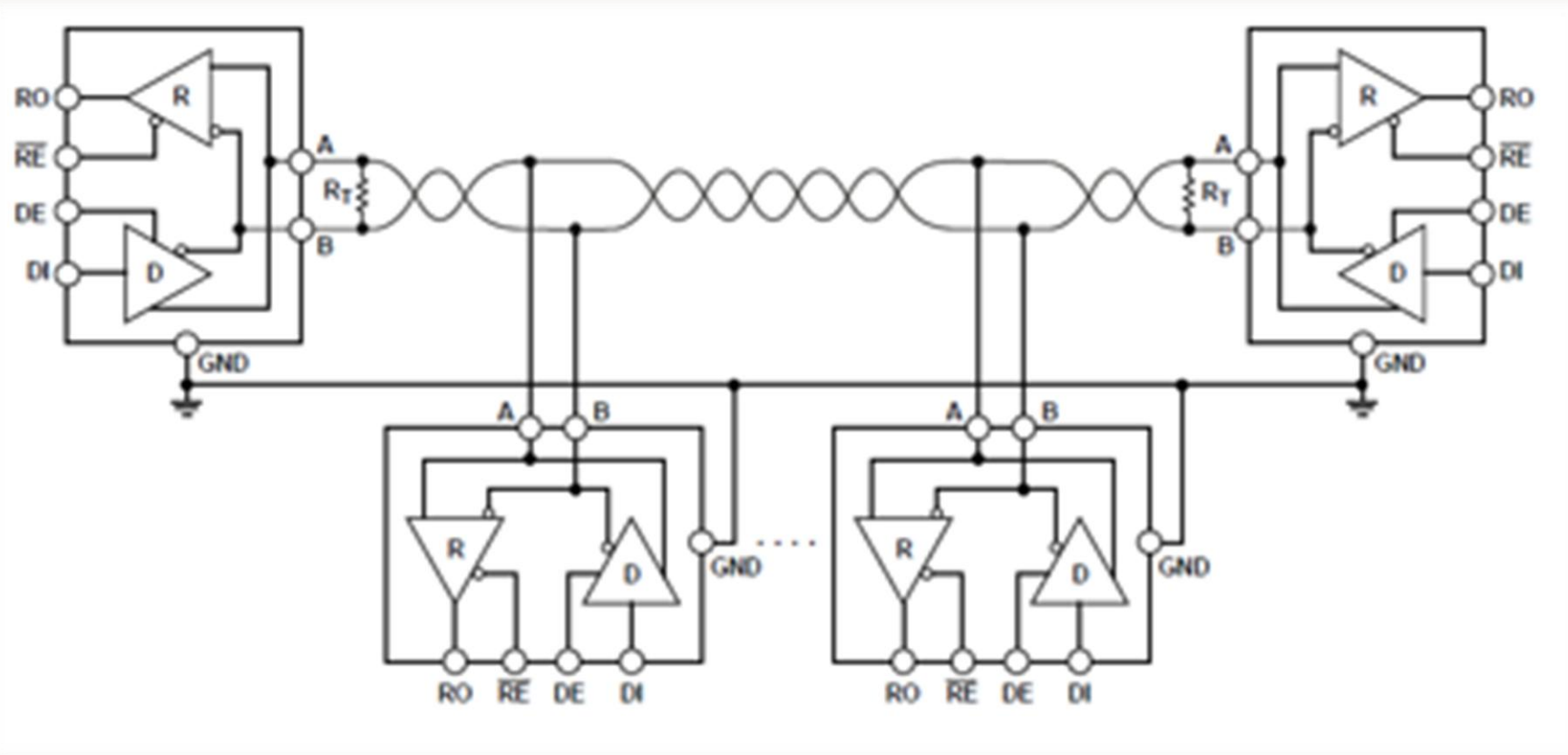
preuzeto sa: <https://www.pearsonitcertification.com/articles/article.aspx?p=1816014>



preuzeto sa <https://instrumentationtools.com/rs485-serial-communication/>

RS-485 STANDARD

- Brzine prenosa od 9.6 kbit/s do 2 Mbit/s (RS-232 max 20 kb/s)
- Max dužina veze do **1200 m** (RS-232 max 15 m) zavisi od brzine !!!
- Max 32 uređaja po segmentu, korišćenje repetitora
- Postavljanje otpornika od $120\ \Omega$ na krajevima segmenta (refleksija)
- Master/slave komunikacija



RS-485 STANDARD

Primer koda za ESP32 (RS-485):

- Koristi ugrađenu Serial biblioteku sa dodatnim pinom za kontrolu smera (DE/RE).
- Ovaj primer je za Master uređaj koji šalje ADC vrednost (sa potencijometra) na bus.
- Potreban hardver: Dva ESP32 + MAX485 moduli (poveži A-A, B-B, GND-GND; DE/RE na digitalni pin)

```
int enablePin = 8;  //Pin za kontrolu smera (HIGH=šalji, LOW=prima)
int analogPin = A0; // Potencijometar na A0
int adcValue = 0;
```

```
void setup() {
  Serial.begin(9600); // Baud rate za RS-485
  pinMode(enablePin, OUTPUT);
  pinMode(analogPin, INPUT);
  digitalWrite(enablePin, HIGH); // Postavi na slanje
}
```

```
void loop() {
  adcValue = analogRead(analogPin); // Čitaj ADC
  Serial.println(adcValue); // Pošalji na RS-485 bus
  delay(100); // Pauza između slanja
}
```

SPI (SERIAL PERIPHERAL INTERFACE)

- ❑ Serijski interfejs za povezivanje periferija
- ❑ SPI je interfejs koji se obično koristi za slanje podataka između mikrokontrolera i perifernih uređaja kao što su registri pomerenja, senzori, SD kartice i sl.
- ❑ Sinhroni prenos podataka
- ❑ Zašto?
 - ❑ Nedostaci serijskog asinhronog prenosa!!

SPI (SERIAL PERIPHERAL INTERFACE)

- ❑ Glavni Koncept: Sinhroni Prenos
- ❑ Sada imamo žicu za takt: SCK (Serial Clock).
- ❑ Master diktira brzinu i Slave-ovi ga prate.
- ❑ Četiri Žice:
 - SCK (Clock)
 - MOSI (Master Out, Slave In)
 - MISO (Master In, Slave Out) -> Ovo omogućava Full-Duplex.
 - SS / CS (Chip Select) -> Hardversko "adresiranje".
- ❑ Adresiranje: Svaki Slave treba svoj CS pin. **Ovo je glavna mana.**
- ❑ CPOL/CPHA: Najčešći problem (4 SPI moda) – obavezno obratiti pažnju kako bi se izbegle greške u implementaciji.

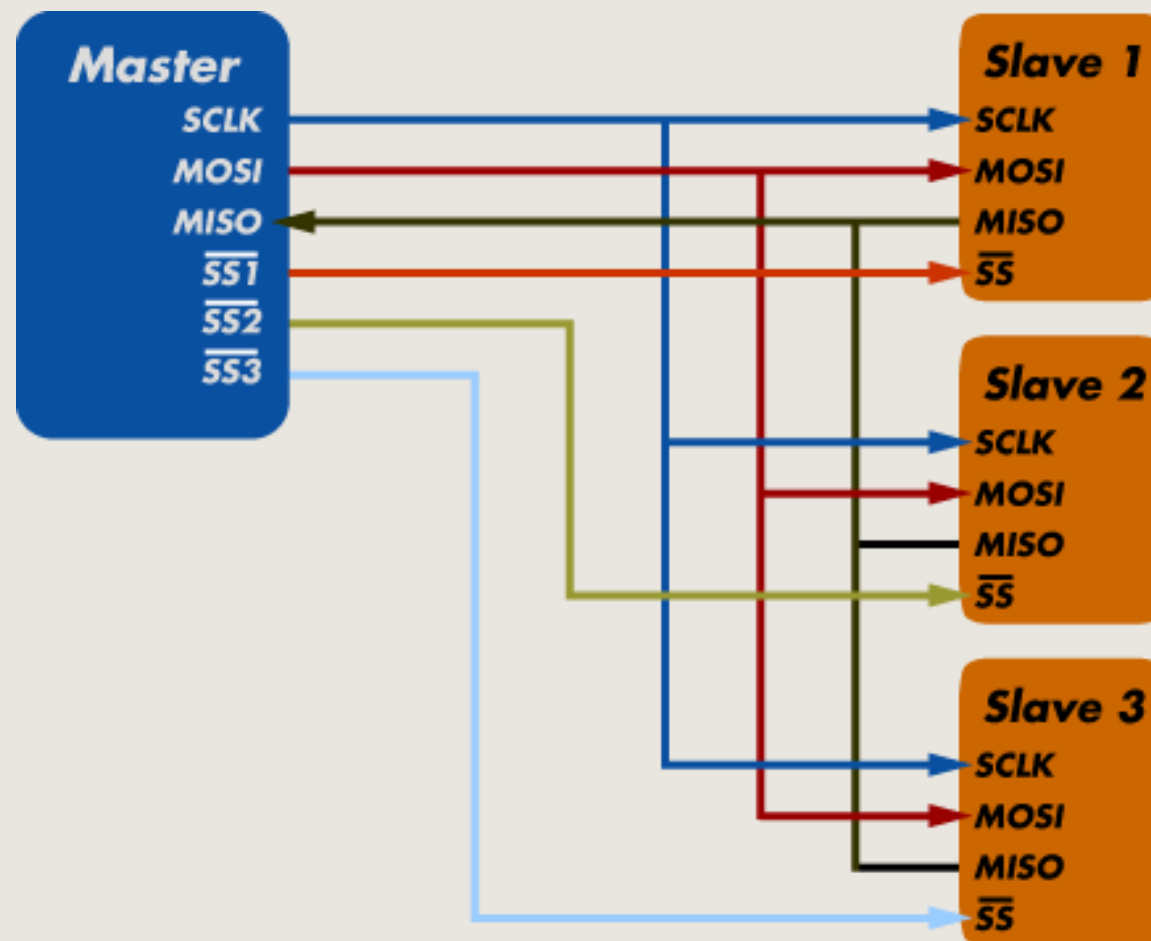
SPI (SERIAL PERIPHERAL INTERFACE)

❑ Četiri Žice:

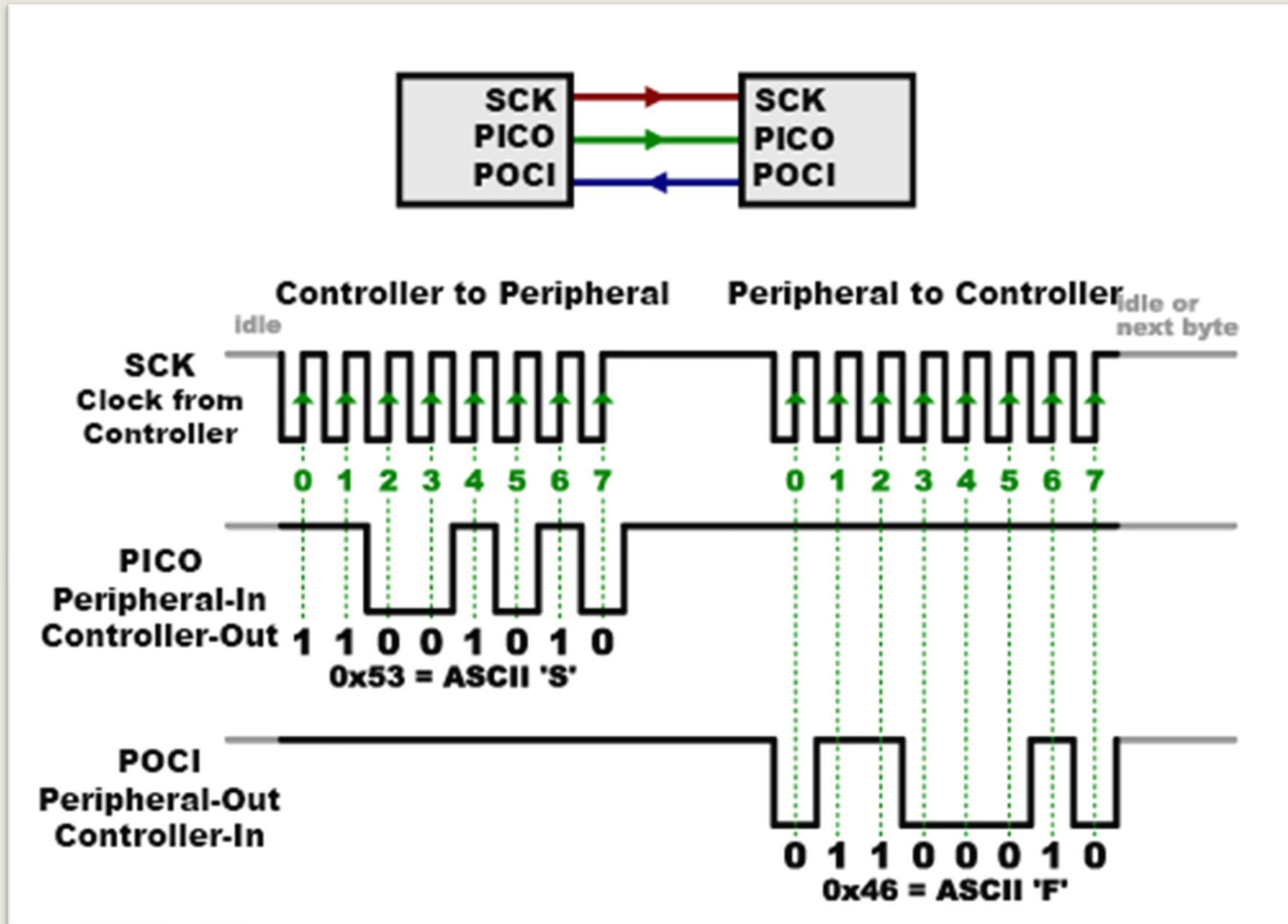
- SCK (Clock)
- MOSI (Master Out, Slave In)
- MISO (Master In, Slave Out)
(Ovo omogućava Full-Duplex).
- SS / CS (Chip Select)
(Hardversko "adresiranje,,).

❑ Adresiranje: Svaki Slave treba svoj CS pin. **Ovo je glavna mana.**

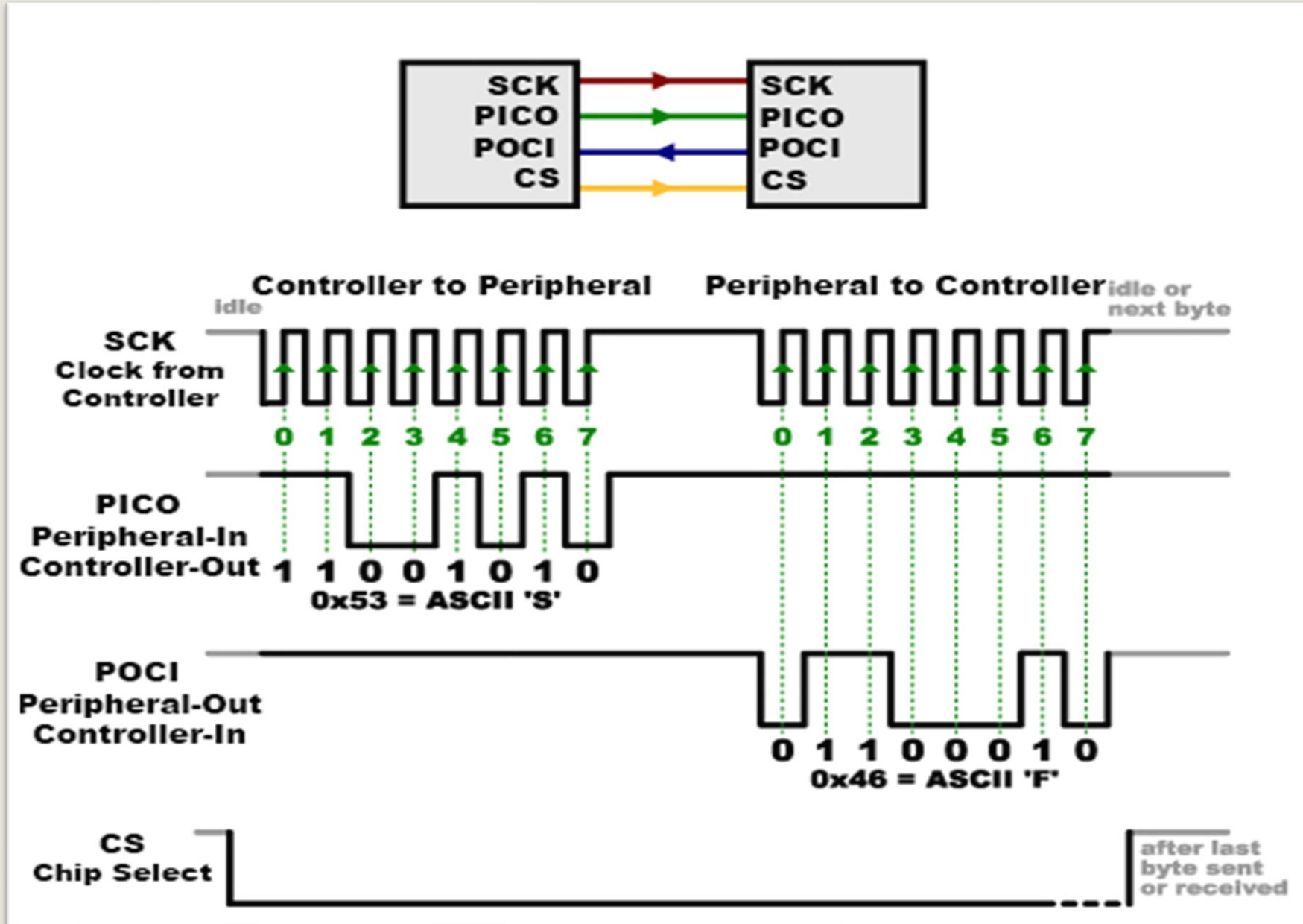
❑ CPOL/CPHA: Najčešći problem (4 SPI moda) – obavezno obratiti pažnju kako bi se izbegle greške u implementaciji.



SPI (SERIAL PERIPHERAL INTERFACE)



SPI (SERIAL PERIPHERAL INTERFACE)



SPI (SERIAL PERIPHERAL INTERFACE)

```
#include <SPI.h>
#define CS_PIN 10 // Chip Select pin
#define BUTTON_PIN 2 // Dugme na pin 2

void setup() {
  Serial.begin(115200);
  pinMode(CS_PIN, OUTPUT);
  pinMode(BUTTON_PIN, INPUT);
  SPI.begin(); // Inicijalizuj SPI kao Master
}

void loop() {
  int buttonValue = digitalRead(BUTTON_PIN); // Čitaj dugme
  digitalWrite(CS_PIN, LOW); // Odaberi Slave
  int received = SPI.transfer(buttonValue); // Pošalji bajt i primi odgovor
  digitalWrite(CS_PIN, HIGH); // Otpusti Slave
  Serial.println(received); // Ispiši primljeno
  delay(500);
}
```

Primer Koda za ESP32 (SPI Master):

Ovaj primer šalje bajt podataka (npr. od dugmeta) na Slave i prima odgovor.

Potreban hardver: Dva ESP32 (jedan Master, jedan Slave), poveži MOSI-MOSI (pin 11), MISO-MISO (pin 12), SCK-SCK (pin 13), CS-CS (npr. pin 10).

PRENOS SIGNALA/PODATAKA

- Prednosti:

- Brži prenos nego I2C i asinhroni serijski
- Jednostavan potreban hardver za prijem (shift registar)
- Full-duplex komunikacija
- Podržava komunikaciju sa više prijemnika

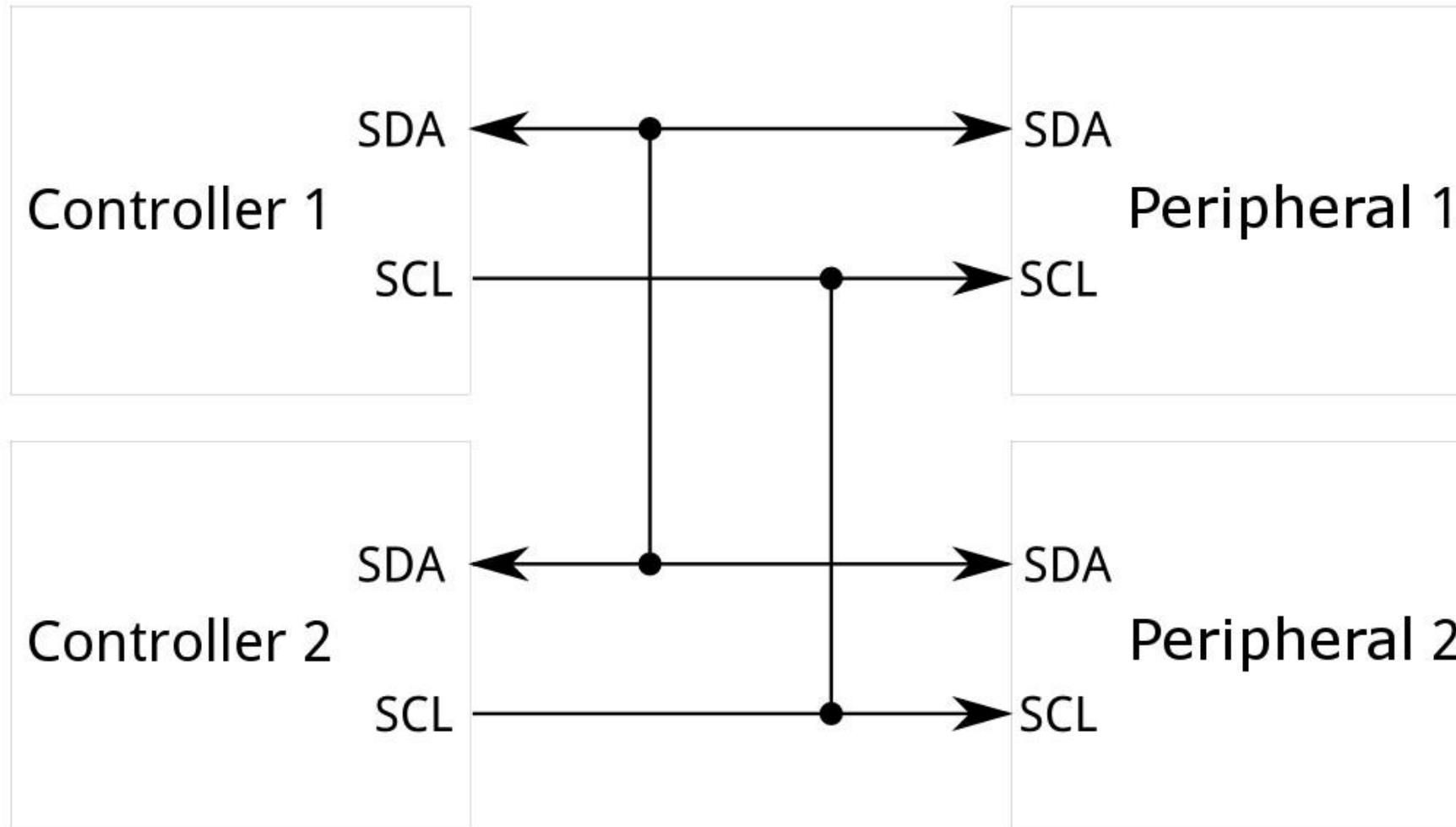
- Nedostaci:

- Zahteva 2, 3 tj. 4 i više linija za komunikaciju
- Protokol komunikacije mora biti unapred definisan
- Kontroler kontroliše komunikaciju (periferni uređaj ne može poslati podatak bez „pozivke“)
- Za rad sa većim brojem perifernih uređaja raste i broj potrebnih CS linija
- Nema potvrde uspešnog prijema (I2C ima)
- Nema proveru tačnosti prenosa (parity)
- Može biti samo jedan master uređaj

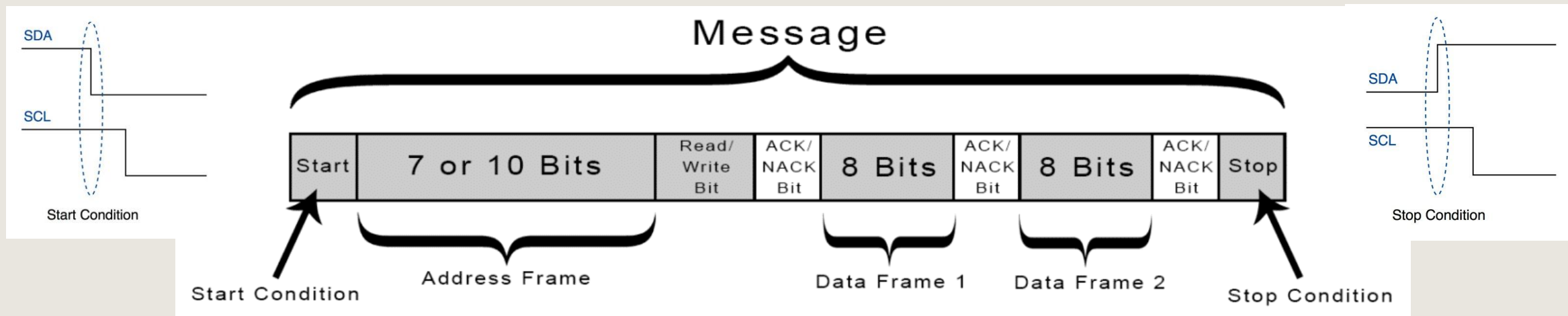
I2C (INTER-INTEGRATED CIRCUIT)

- ❑ Namenjen je za komunikaciju između mikorokotrolera i perifernih integriranih kola
- ❑ Kao i SPI namenjen je za komunikaciju na kratkim rastojanjima (unutar jednog uređaja)
- ❑ **Ključna poruka:** I2C je serijski, sinhroni, multi-master, multi-slave protokol koji koristi samo dve žice zahvaljujući softverskom adresiranju
- ❑ Za razmenu podataka potrebne su mu samo dve veze:
 - SCL (Serial Clock)
 - SDA (Serial Data) -> Podaci idu u oba smera, zato je **Half-Duplex**.
- ❑ **Glavni Koncept:** Softversko Adresiranje
 - Nema CS pinova! Svaki Slave ima jedinstvenu 7-bitnu adresu (do 127 uređaja).
 - Master započinje komunikaciju tako što pošalje adresu Slave-a sa kojim želi da priča. Samo taj Slave odgovara.

I2C (INTER-INTEGRATED CIRCUIT)



I2C (INTER-INTEGRATED CIRCUIT)

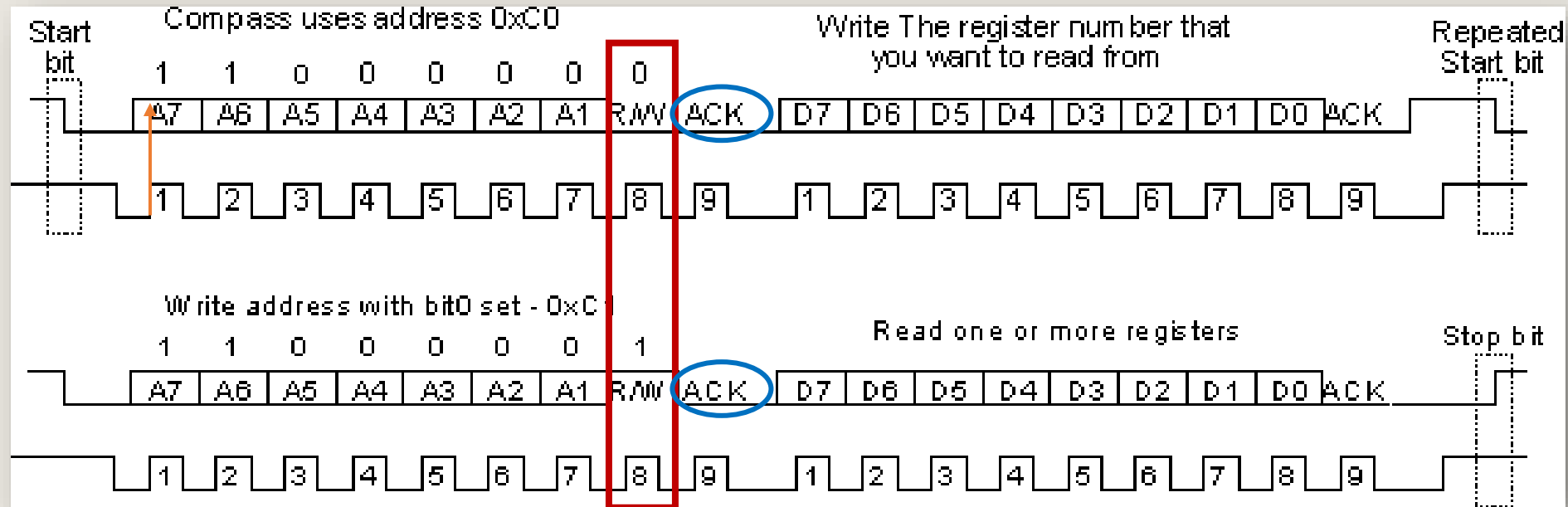


preuzeto sa <https://www.circuitbasics.com/wp-content/uploads/2016/01/Introduction-to-I2C-Message-Frame-and-Bit-2.png>

Protokol I2C komunikacije

- ❑ **START** Uslov: Master povuče SDA na LOW dok je SCL HIGH. (Svi se "probude").
- ❑ **Adresni Frejm:** Master šalje 7 bita adrese + 1 bit (Read/Write).
- ❑ **ACK/NACK** (Potvrda prijema):
 - Ovo je prednost nad SPI!
 - Nakon svakog bajta (adrese ili podatka), primalac mora da povuče SDA liniju na LOW kao potvrdu (ACK) da je primio bajt.
 - Ako Master ne dobije ACK, zna da nešto nije u redu.
- ❑ **Prenos Podataka:** Bajt po bajt, svaki praćen sa ACK.
- ❑ **STOP** Uslov: Master digne SDA na HIGH dok je SCL HIGH. (Kraj, magistrala je slobodna).

I2C (INTER-INTEGRATED CIRCUIT)



I2C (INTER-INTEGRATED CIRCUIT)

❑ Prednosti:

- Samo dve žice! Izuzetna ušteda pinova.
- Multi-Slave/Multi-Master: Lako dodavanje uređaja (do 127).
- Potvrda (ACK/NACK): Robustan, Master zna da li ga Slave čuje.

❑ Nedostaci:

- Brzina: Sporiji od SPI-ja (100 kbit/s, 400 kbit/s).
- Hardverska kompleksnost: Pull-up otpornici su obavezni i znaju da budu zbunjujući (odabir vrednosti).
- Dužina: Ograničen na kratke distance (unutar ploče ili kutije).

❑ Zaključak:

- Koristimo I2C kada imamo puno uređaja (senzori, memorije) a malo pinova.

I2C (INTER-INTEGRATED CIRCUIT)

```
#include <Wire.h>
```

```
byte i2c_rcv = 255;
unsigned long time_start;
int stat_LED = 0;
byte value_pot;
```

```
void setup() {
  Wire.begin(); // Inicijalizuj I2C kao Master
  time_start = millis();
  stat_LED = 0;
  pinMode(13, OUTPUT);
  Serial.begin(9600);
}
```

```
void loop() {
  value_pot = analogRead(A0)/4; // Čitaj potencijometar (0-255)
  Wire.beginTransmission(4);    // Adresa Slave-a
  Wire.write(value_pot);        // Pošalji bajt
  Wire.endTransmission();      // Završi transmisiju

  if ((millis() - time_start) > 500) { // Blink LED za test
    digitalWrite(13, stat_LED);
    stat_LED = !stat_LED;
    time_start = millis();
  }
  delay(10);
}
```

Primer Koda za ESP32 (I2C Master):

Koristi Wire biblioteku.

Ovaj primer šalje vrednost potencijometra na Slave (adresa 4) i kontrolira LED.

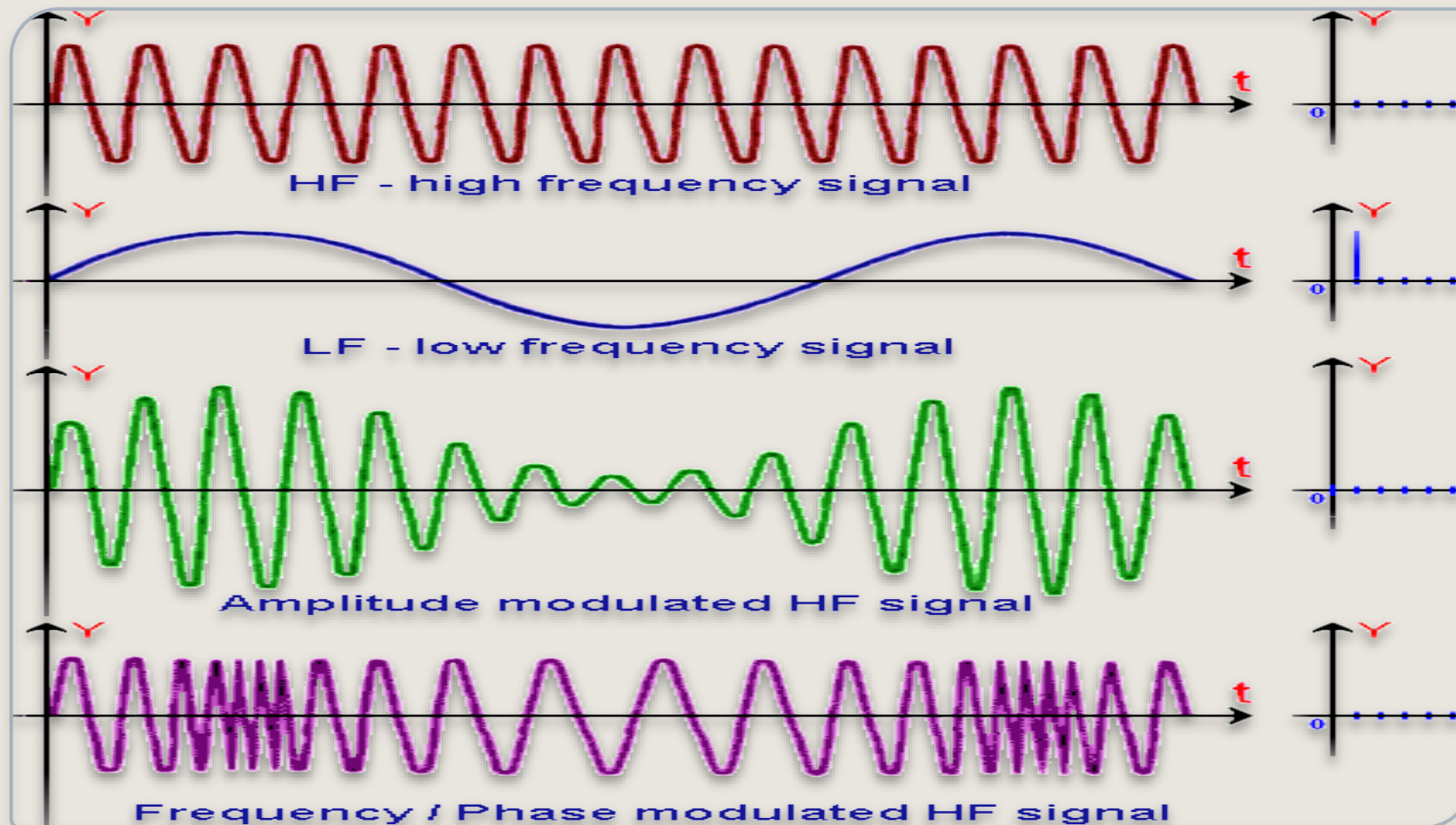
Potreban hardver: Dva ESP32 (Master na A4/SDA, A5/SCL; Slave isto, plus pull-up otpornici).

UPOREDNA TABELA

Karakteristika	RS-232	RS-485	SPI	I2C
Glavna Namena	Stari Point-to-Point	Industrija, Dug Domet	Brzina, Kratka rastojanja	Efikasnost, Kratka rastojanja
Signalizacija	Single-Ended	Diferencijalna	Single-Ended (Push-Pull)	Single-Ended (Open-Drain)
Topologija	Point-to-Point	Multi-Drop (Bus)	Master-Slave (Star)	Multi-Master/Slave (Bus)
Domet	< 15 m	> 1000 m	< 1 m	< 1 m
Brzina	Spora (kbps)	Srednja (kbps/Mbps)	Veoma Brza (Mbps)	Srednja (kbps/Mbps)
Broj žica (min)	3 (TX, RX, GND)	2 (A, B)	4 (SCK, MISO, MOSI, CS)	2 (SDA, SCL)
Duplex	Full-Duplex	Half-Duplex	Full-Duplex	Half-Duplex
Adresiranje	Nema	Softversko (preko protokola)	Hardversko (CS pin)	Softversko (7-bit adresa)
Potvrda (ACK)	Ne	Ne	Ne	Da

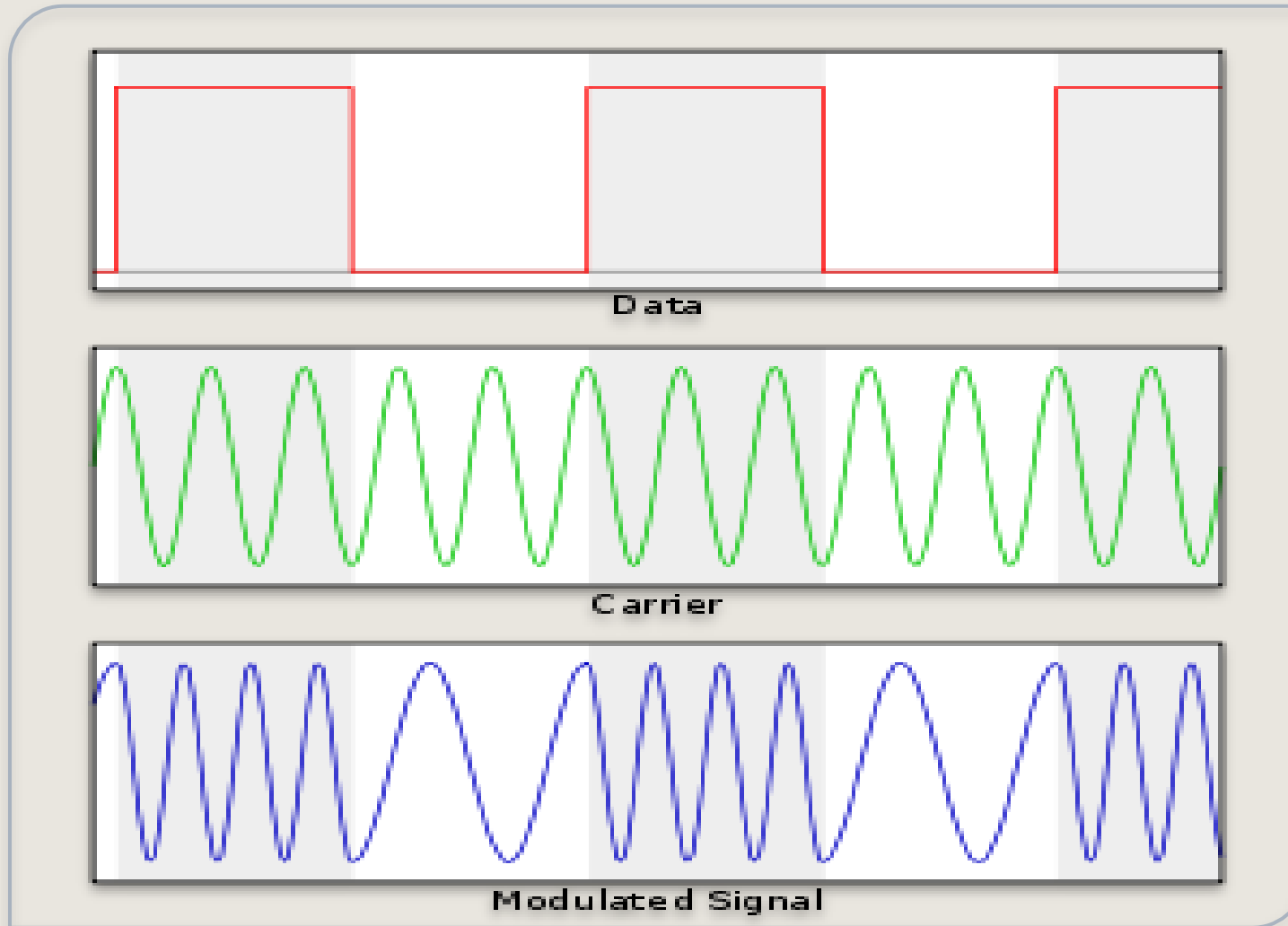
I2C (INTER-INTEGRATED CIRCUIT)

- AM modulacija
- FM modulacija



PRENOS SIGNALA/PODATAKA

- AM modulacija
- FM modulacija



PRENOS SIGNALA/PODATAKA

Optički kablovi za prenos informacije koristi se fenomen totalne refleksije svetlosti

Koriste se tamo gde je potrebno:

- preneti veliku količinu podataka
- preneti podatke na velike razdaljine

Prednosti nad klasičnim bakarnim kablovima:

- manje slabljenje signala
- manja osetljivost na smetnje
- veliki kapacitet prenosa podataka

Za prenos podataka koristi se svetlost umesto strujnih impulsa

Napravljeni su od stakla ili posebne plastike

Imaju mali prečnik i dosta su savitljivi

