



# Vhodno-izhodne naprave (VIN)

Predavanja

## 12. Serijske komunikacije v praksi – RS-485, Canbus, Modbus

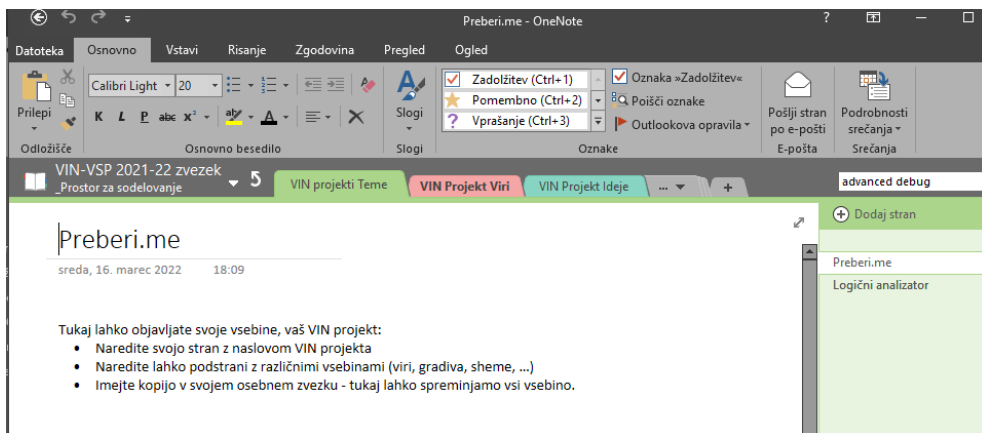
Robert Rozman

[rozman@fri.uni-lj.si](mailto:rozman@fri.uni-lj.si)

# Tekoča obvestila

## Predstavitve Red Pitaya

### VIN projekt – zadnje predavanje



**redpitaya**

**Edino orodje, ki ga potrebuješ za svoj projekt**

**17 MAJ 2022  
13H - 15H V P22 IN  
MAKERSPACE**

**VEČ INFORMACIJ IN  
PRIJAVA:**

Univerza v Ljubljani  
Fakulteta za računalništvo  
in informatiko

# Vsebina

---

## I. RS 485

- ▶ Opis protokola in fizičnega nivoja
- ▶ Primeri uporabe

## 2. CANBUS

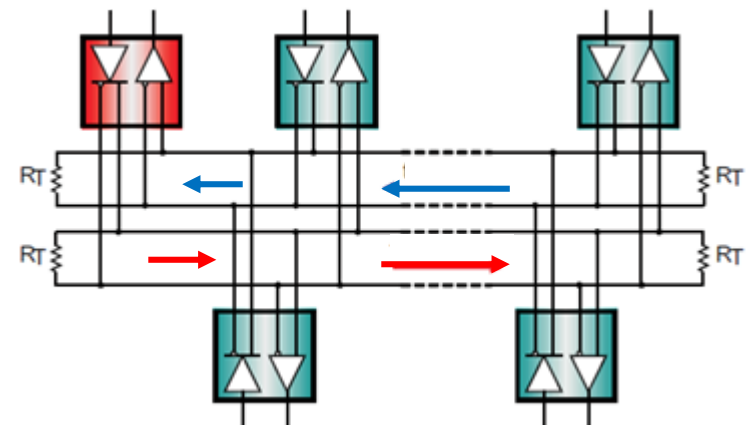
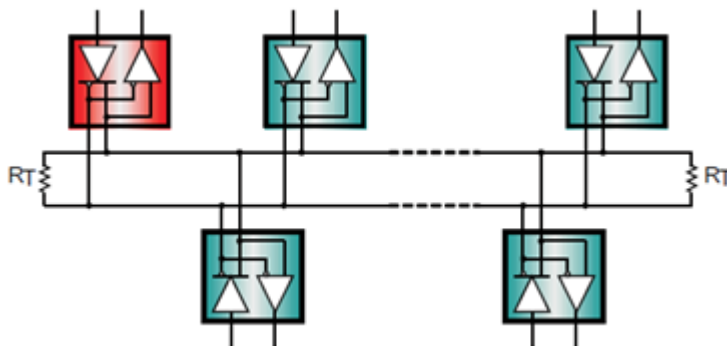
- ▶ Opis protokola in fizičnega nivoja
- ▶ Primer uporabe:
  - Integra BM: HW moduli in IEX-2 odprta protokolska nadgradnja, ...
  - Vsebina LAB vaje : CANBUS, STM32 in IEX-2 protokol

## 3. Modbus:

- ▶ Opis protokola in fizičnih nivojev
- ▶ Primeri Modbus komunikacij:
  - STM32 (MB master) <-> PC (MB slave)
  - Cybro kot MB Slave (višje nivojsko omrežje)

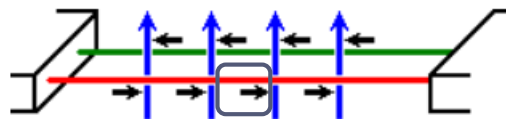
# 1. RS-485 (EIA/TIA-485 )

- Standard RS-485 definira **električne lastnosti** oddajnikov, linije in sprejemnikov za simetrično povezavo v obliki vodila.
  - simetrični, **diferencialni prenos v obe smeri** (half-duplex ali full-duplex).
  - Priporoča **uporabo parice in zaključitev linije** za omejevanje odbojev.
  - Omogoča **hitrosti do 35 Mb/s** in več ter dolžino **povezave do 1200 m**
    - vendar ne hkrati (višja kot je hitrost  $\Leftrightarrow$  krajša je povezava in obratno)
  - **število oddajnikov, kot število sprejemnikov do 32.**
    - Če obremenitev ni presežena, je lahko sprejemnikov **tudi več, do 256.**
  - Primer: half-duplex in full-duplex struktura vodila v RS-485

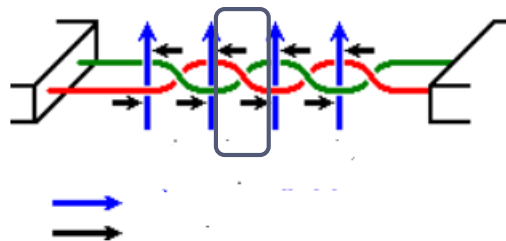


- ❑ Standard ne določa posebnega protokola za prenos podatkov, obratno pa veliko protokolov uporablja RS-485 kot električni standard povezave (**npr. Modbus**)
- ❑ Povezave po standardu RS-485 so zaradi simetrične povezave in uporabe paric zelo uporabne v industrijskih okoljih:
  - **Neobčutljive za šum** in premik potenciala ozemljitve
  - **Ne povzročajo motenj navzven** (presluh na sosednje povezave)
  - Možne so **višje hitrosti prenosa**
  - Kjer ni zahtevana velika hitrost, so **povezave lahko zelo dolge** (npr. v procesni industriji)
  - Primer za raven kabel in sukano parico

Raven kabel



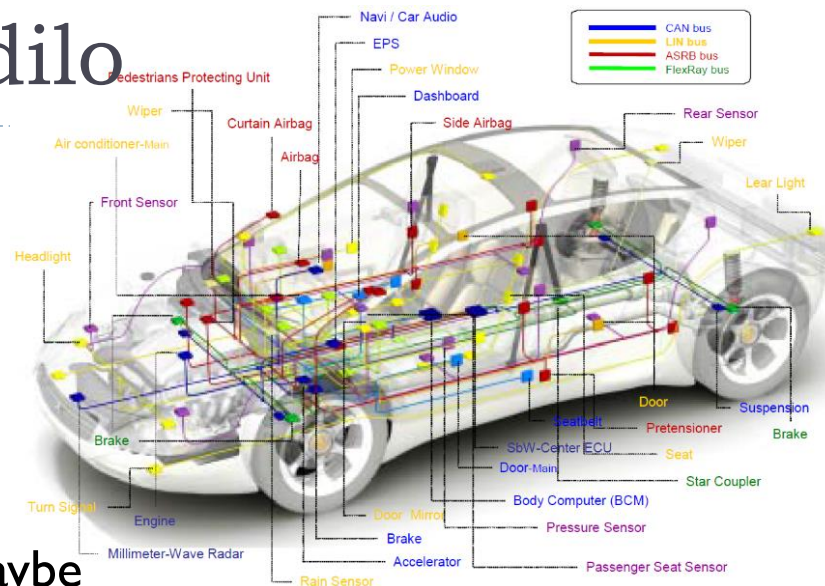
Sukana parica



## 2. CANBUS vodilo

### CANBUS (ISO-11898-2):

- ▶ Zgodovina
- ▶ Področja uporabe
  - ▶ **Avtomobilska** industrija
  - ▶ Industrijska **avtomatika**, pametne stavbe
- ▶ Pregled protokola, arbitraže, fizičnega nivoja
- ▶ Praktični primer: Pametna hiša, IEX-2 protokol



### Lab.Vaja :

- ▶ **Gradniki in shema** testnega sistema
- ▶ **Programiranje** sistema
- ▶ **Meritve signalov** na povezavah

# CANbus zgodovina

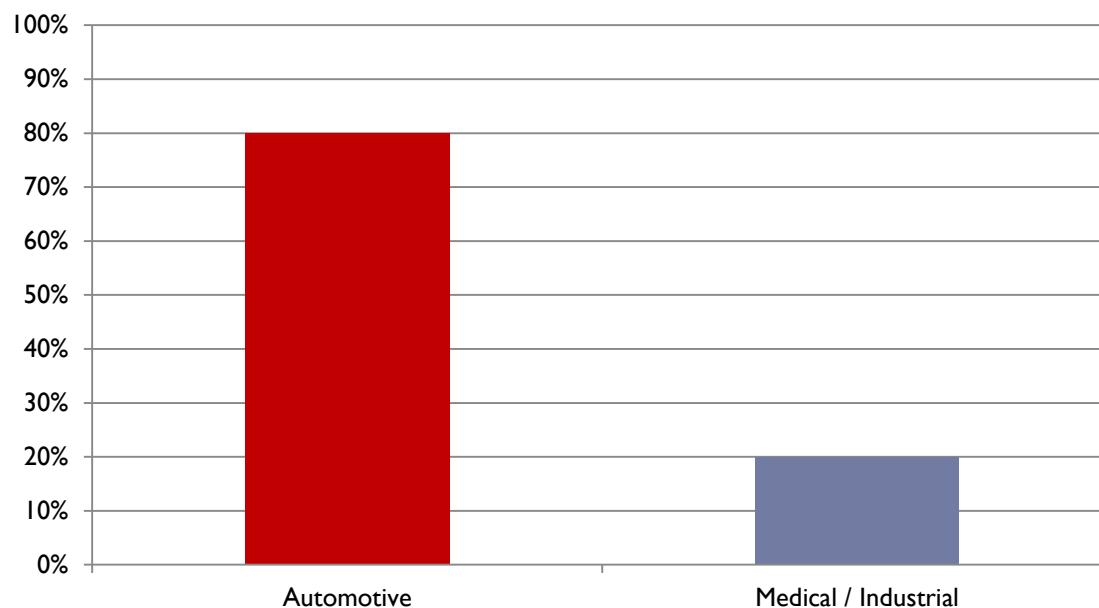
---

- ▶ 1983 : Prvi CANBUS projekt - Bosch
- ▶ 1986 : CAN protokol predstavljen (BMW 850)
- ▶ 1987 : Prvi CAN mikrokontrolerji prodani
- ▶ 1991 : CAN 2.0A specifikacija objavljena
- ▶ 1992 : Mercedes-Benz uporabi CAN omrežje
- ▶ 1993 : ISO 11898 standard
- ▶ 1995 : ISO 11898 amandma
- ▶ Dandanes : večina vozil uporablja CANbus, industrijska avtomatika

# Uporaba CANBUS-a

- ▶ Zasnovan za avtomobilsko industrijo
- ▶ Danes: Industrija/medicina

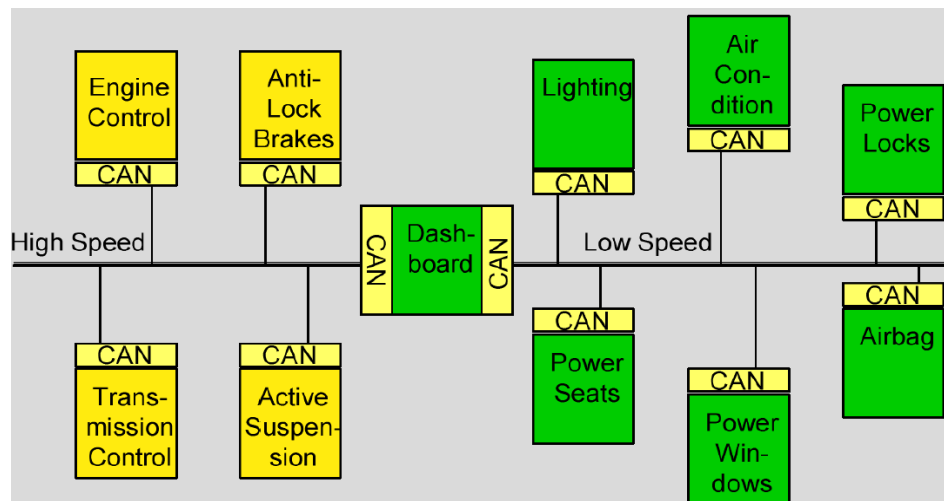
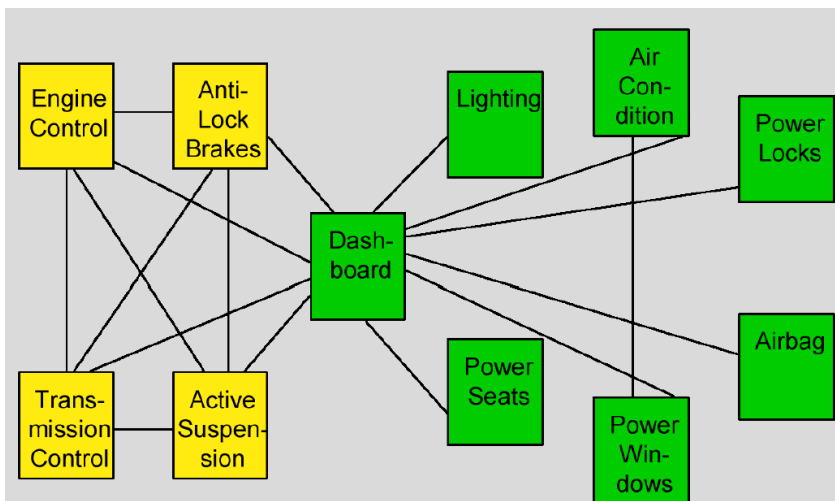
## CANBUS Tržni deleži





# Zakaj vodilo ?

Primeri povezav brez (levo) in z (desno) CANbus v avtomobilu

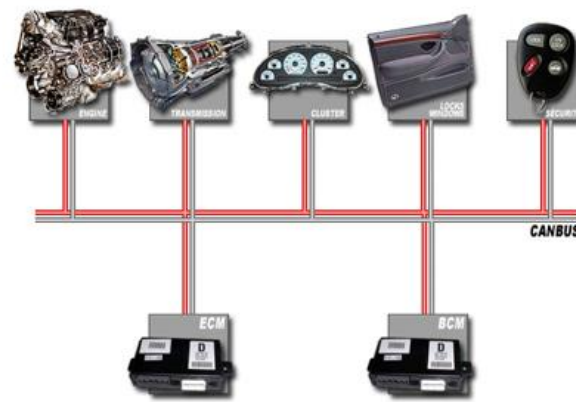


Conventional multi-wire looms

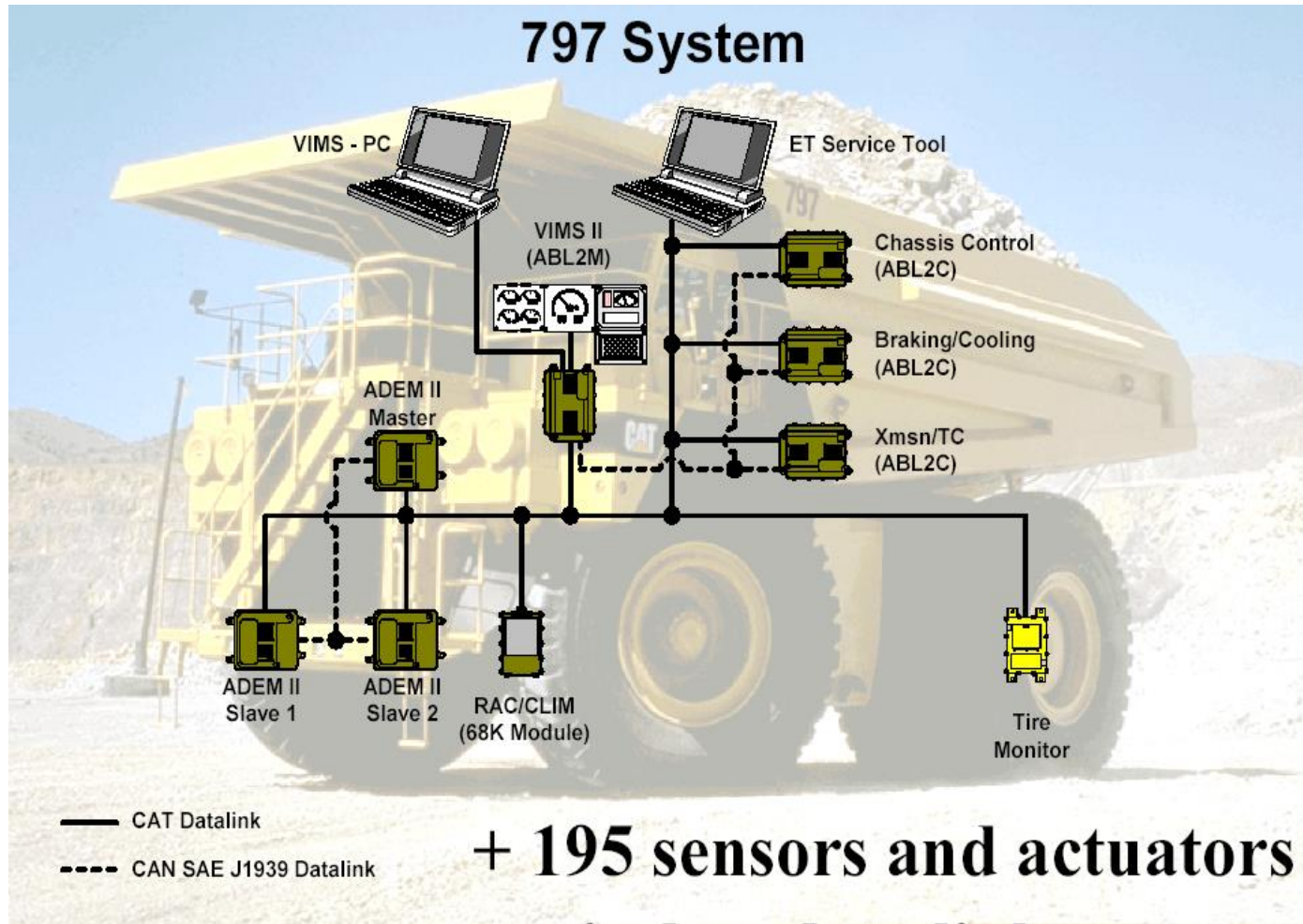


CAN bus network

VS.



# CAN (SAE J1939) Primer: Caterpillar 797



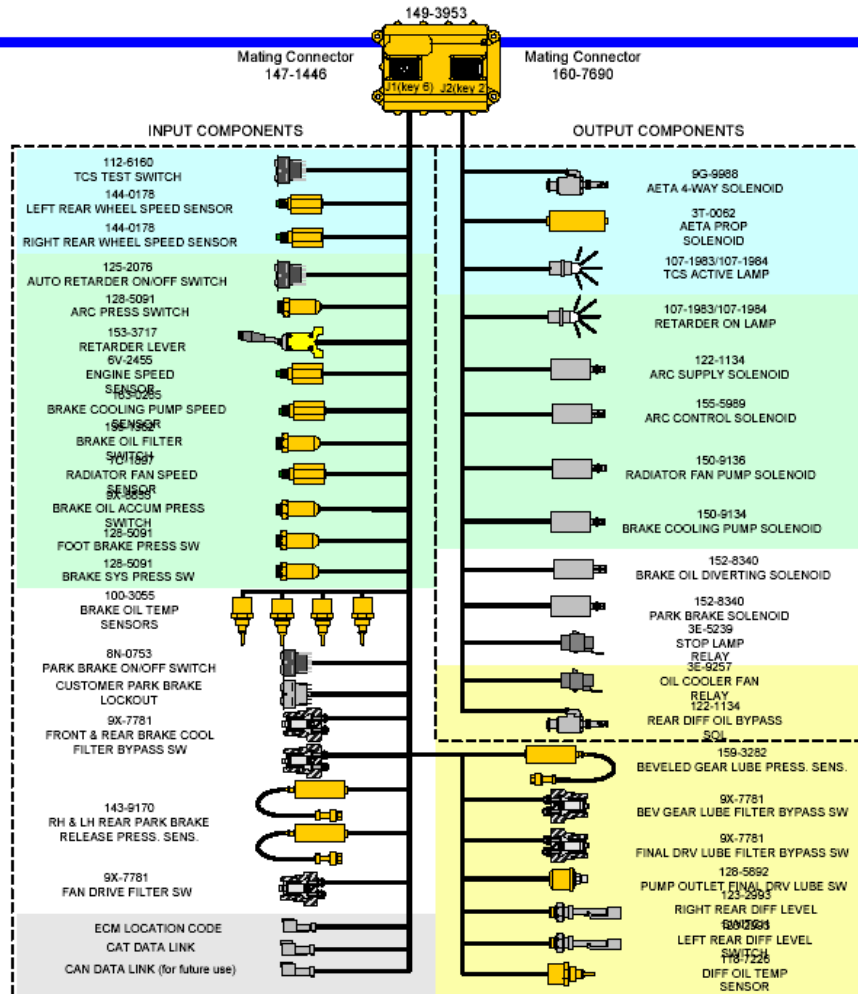
**+ 195 sensors and actuators**  
**+ wireless data link**

797sys.vsd  
6-18-98  
dab/yrf  
Warning: All paper copies of this document are uncontrolled

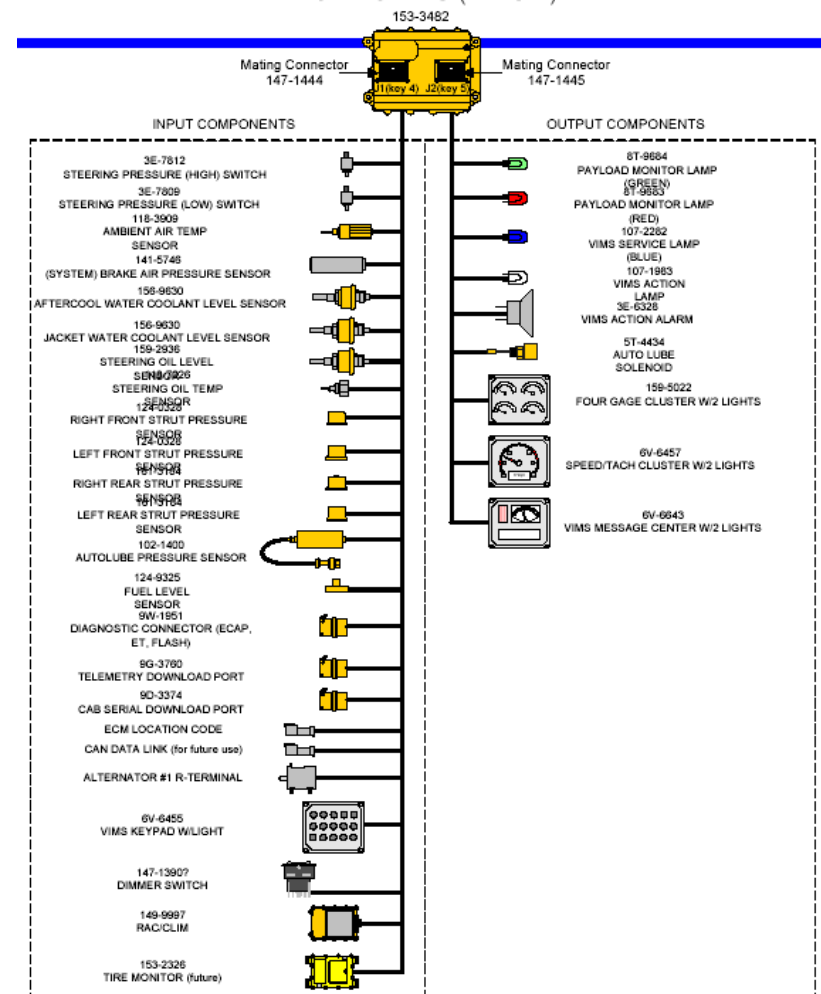
[Caterpillar]

# Caterpillar example

## BRAKING/COOLING CONTROL

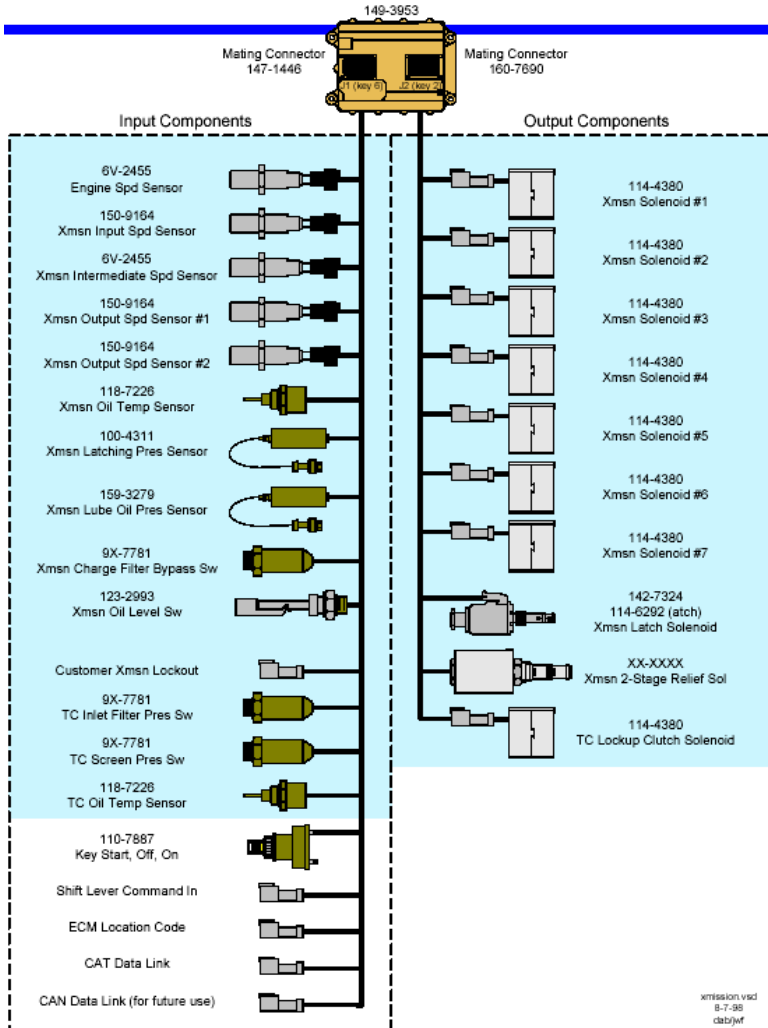


## MONITORING (VIMS-II)

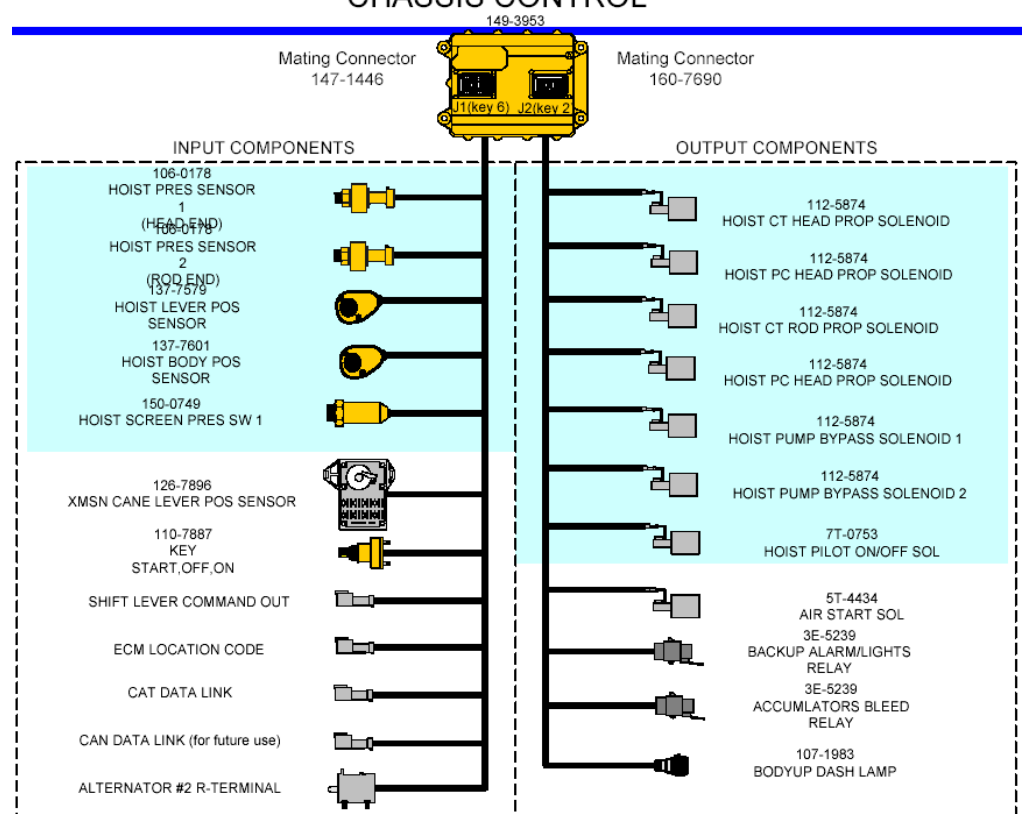


# Caterpillar example

## 797 - TRANSMISSION CONTROL



## CHASSIS CONTROL

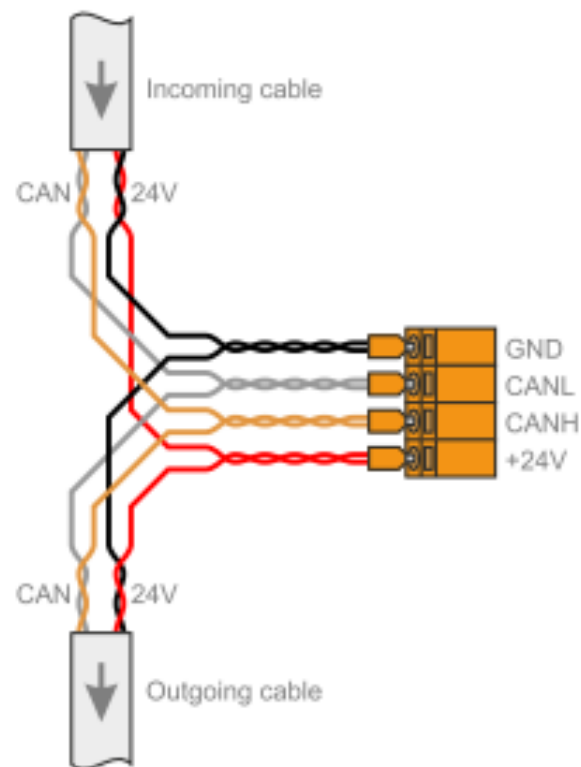


## “SAE Standard Workload” (53 messages) V/C = Vehicle Controller

Signal Number	Signal Description	Size /bits	J /ms	T /ms	Periodic /Sporadic	D /ms	From	To
1	Traction Battery Voltage	8	0.6	100.0	P	100.0	Battery	V/C
2	Traction Battery Current	8	0.7	100.0	P	100.0	Battery	V/C
3	Traction Battery Temp, Average	8	1.0	1000.0	P	1000.0	Battery	V/C
4	Auxiliary Battery Voltage	8	0.8	100.0	P	100.0	Battery	V/C
5	Traction Battery Temp, Max.	8	1.1	1000.0	P	1000.0	Battery	V/C
6	Auxiliary Battery Current	8	0.9	100.0	P	100.0	Battery	V/C
7	Accelerator Position	8	0.1	5.0	P	5.0	Driver	V/C
8	Brake Pressure, Master Cylinder	8	0.1	5.0	P	5.0	Brakes	V/C
9	Brake Pressure, Line	8	0.2	5.0	P	5.0	Brakes	V/C
10	Transaxle Lubrication Pressure	8	0.2	100.0	P	100.0	Trans	V/C
11	Transaction Clutch Line Pressure	8	0.1	5.0	P	5.0	Trans	V/C
12	Vehicle Speed	8	0.4	100.0	P	100.0	Brakes	V/C
13	Traction Battery Ground Fault	1	1.2	1000.0	P	1000.0	Battery	V/C
14	Hi&Lo Contactor Open/Close	4	0.1	50.0	S	5.0	Battery	V/C
15	Key Switch Run	1	0.2	50.0	S	20.0	Driver	V/C
16	Key Switch Start	1	0.3	50.0	S	20.0	Driver	V/C
17	Accelerator Switch	2	0.4	50.0	S	20.0	Driver	V/C
18	Brake Switch	1	0.3	20.0	S	20.0	Brakes	V/C
19	Emergency Brake	1	0.5	50.0	S	20.0	Driver	V/C
20	Shift Lever (PRNDL)	3	0.6	50.0	S	20.0	Driver	V/C
21	Motor/Trans Over Temperature	2	0.3	1000.0	P	1000.0	Trans	V/C
22	Speed Control	3	0.7	50.0	S	20.0	Driver	V/C
23	12V Power Ack Vehicle Control	1	0.2	50.0	S	20.0	Battery	V/C
24	12V Power Ack Inverter	1	0.3	50.0	S	20.0	Battery	V/C
25	12V Power Ack I/M Contr.	1	0.4	50.0	S	20.0	Battery	V/C
26	Brake Mode (Parallel/Split)	1	0.8	50.0	S	20.0	Driver	V/C

# CANbus na kratko

- ▶ **CAN**bus – **C**ontroller **A**rea **N**etwork bus
- ▶ CAN (Controller Area Network) je serijsko vodilo za komunikacijo med vgrajenimi mikrokrmilniki
- ▶ CAN bus na kratko :
  - ▶ serijsko vodilo
  - ▶ 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 standard, ISO 11898-2:2016
  - ▶ 2Mbps, 5Mbps





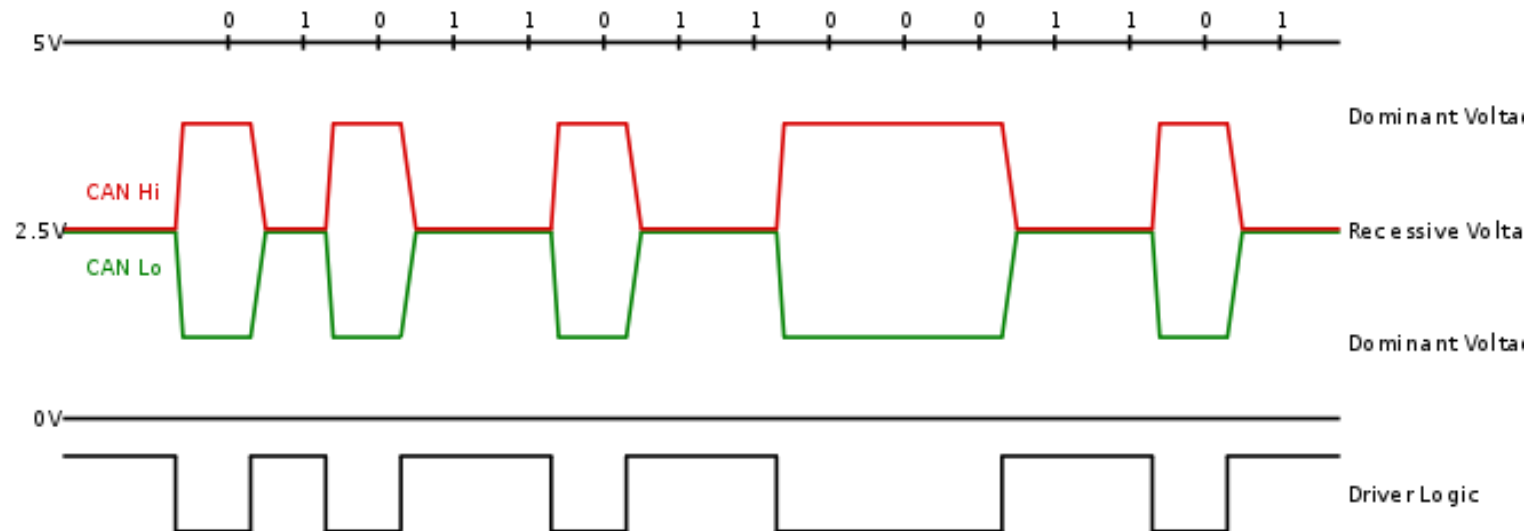
# CANbus na kratko

---

- ▶ 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)
  - ▶ potrjevanje (ACK) vsaj enega sprejemalca

# CANbus na kratko

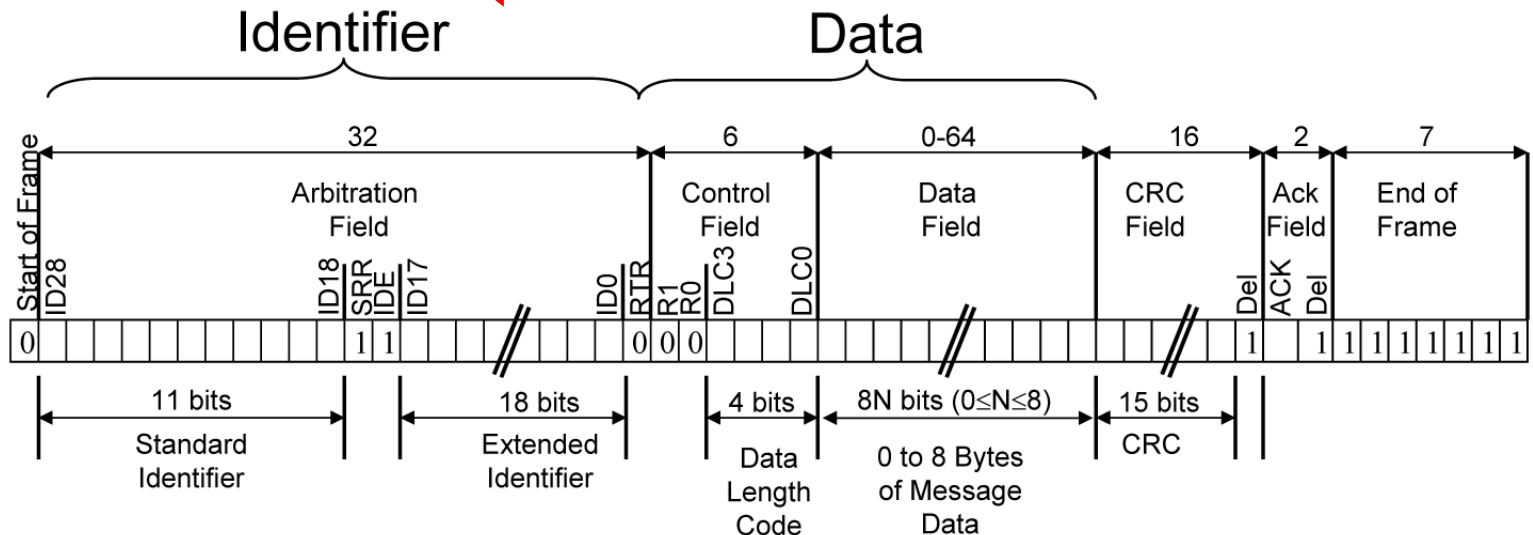
- ▶ **Diferencialni prenos** običajno na parici - Non-Return To Zero (NRZ) in bit-stuffing.
- ▶ **Wired – AND povezava:** vozlišče z logično 0 prevlada
  - ▶ 0 .. „dominant“, 1 .. „recessive“)





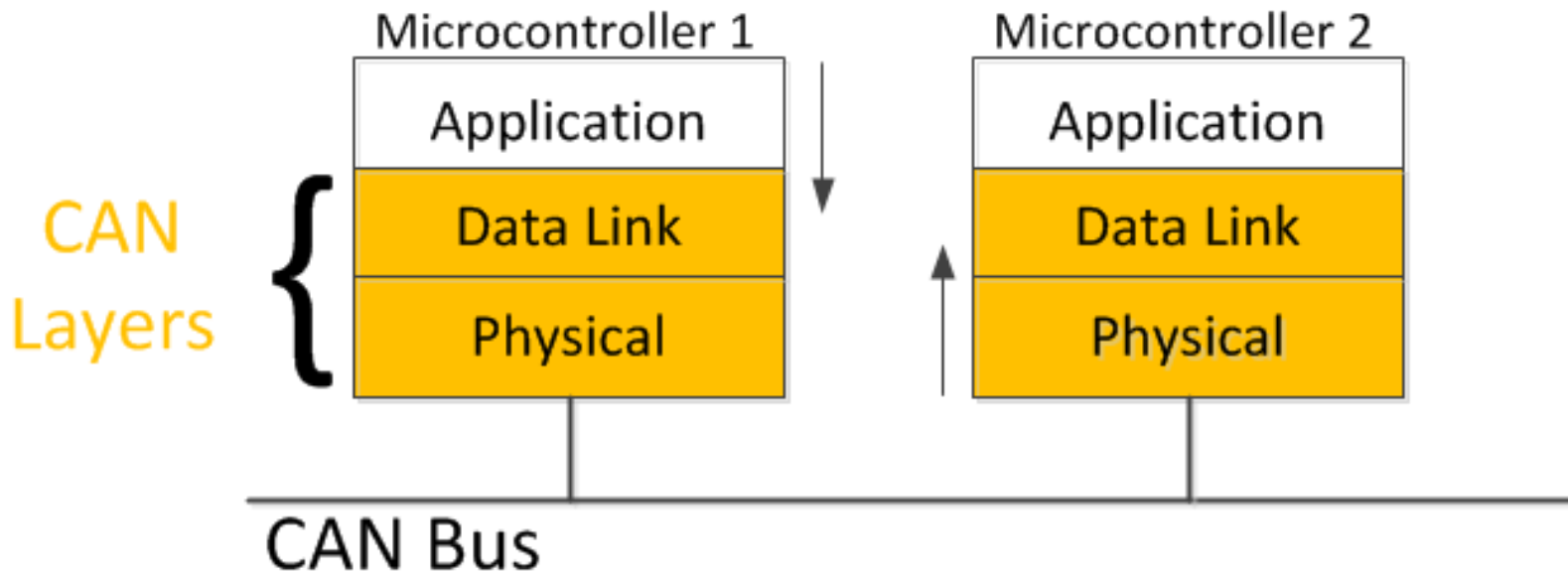
# CANbus na kratko

- ▶ Prenos podatkov
  - ▶ Format okvirja
  - ▶ Protokol – sporočilno naravnan
  - ▶ Detekcija napake
    - ▶ Nivo Bitov (branje, „bit stuffing“)
    - ▶ Nivo sporočila (CRC, okvir, ACK napake)



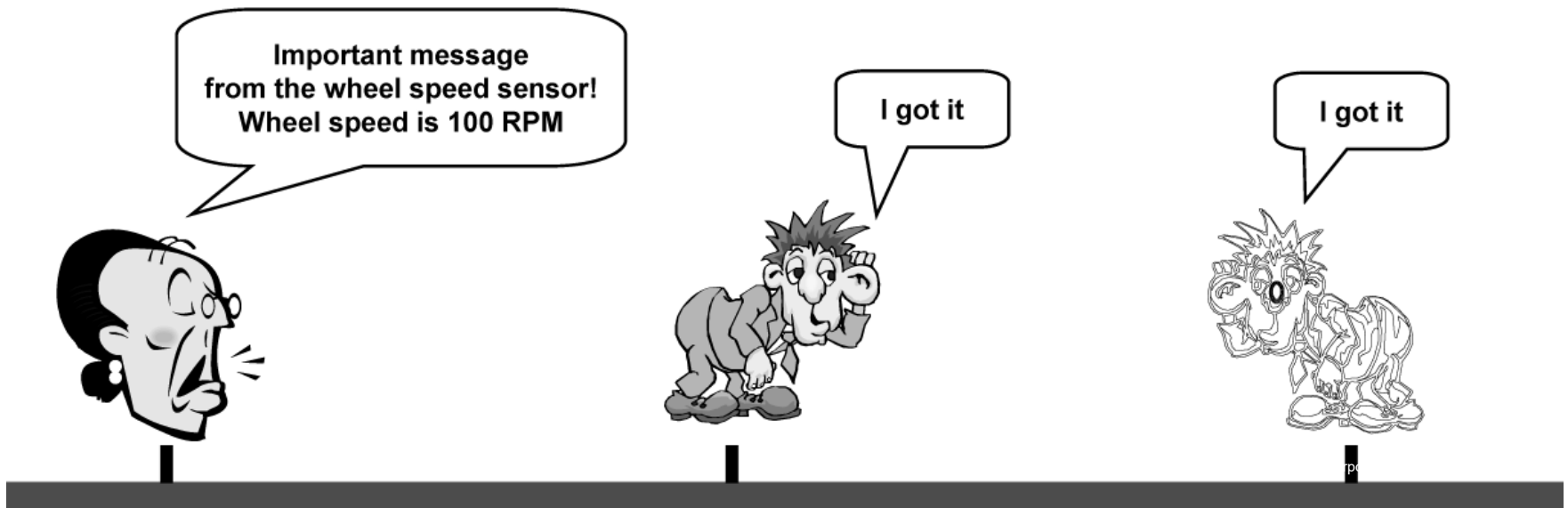
# CANBUS in OSI Model

- ▶ CAN je zaprto omrežje
  - ▶ ni potrebe po varnosti, sejah in prijavah
  - ▶ ni zahtev po uporabniški izkušnji.
- ▶ Fizični in podatkovni nivoji:

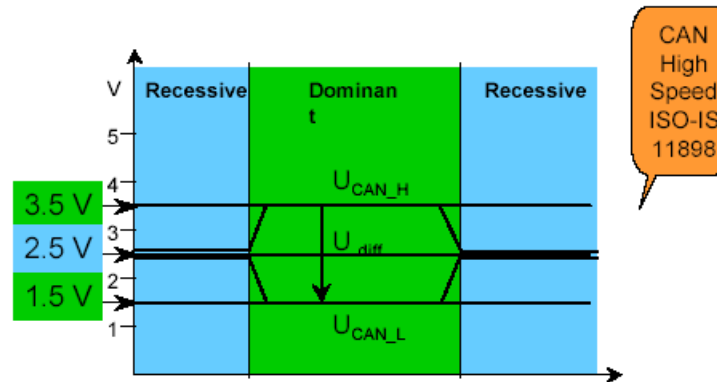


# Sporočilno orientiran protokol

- ▶ Vsako vozlišče – sprejemnik in oddajnik
- ▶ Pošiljatelj pošilja sporočilo **vsem ostalim**
- ▶ **Vsa vozlišča berejo sporočilo** in se odločajo, ali je za njih pomembno.
- ▶ **Vsa vozlišča potrdijo** sprejem brez napak (ne vemo, katero je potrdilo)



# CANbus napetostni nivoji ISO-IS 11898



## •Recesivni bit:

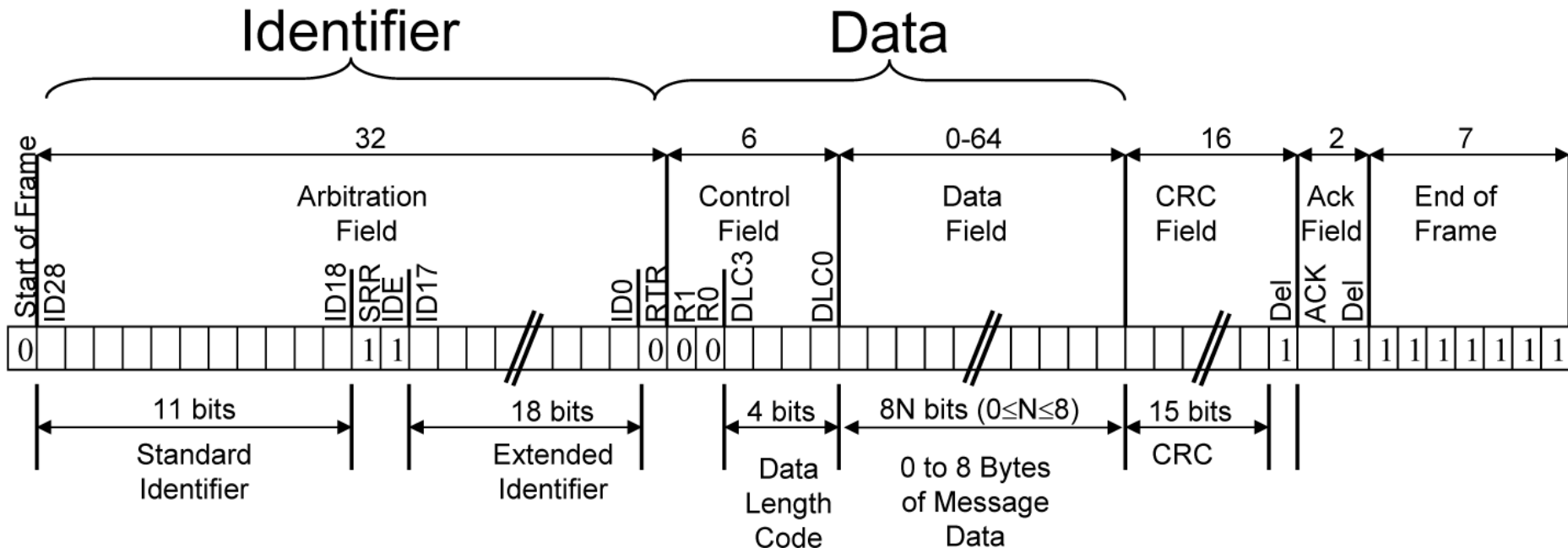
- obe liniji na približno 2.5V
- diferencialna napetost  $CAN\_H$  in  $CAN\_L \approx 0V$

## •Dominantni bit:

- $CAN\_H$  na pribl. 3.5 V in  $CAN\_L$  pribl. 1.5 V
- diferencialna napetost  $CAN\_H$  in  $CAN\_L \approx 2V$

# Format sporočila

- ▶ Vsako sporočilo ima ID, podatke in dodatke
- ▶ ID - 11 ali 29 bitov
- ▶ Data - do 8 bajtov
- ▶ Dodatki - start (SOF), CRC, ACK, end (EOF)



# Format sporočila

## CAN vs. RS-485: Why CAN Is on the Move

By Robert Gee, Executive Business Manager, Core Products Group, Maxim Integrated

Field Name	Bit Length	Description
SOF	1	Start of frame
Identifier (green)	11/29; 12/32	Represents the message priority (11 or 29 bits for standard CAN and extended CAN; 12 or 32 bits for CAN-FD)
RTR (blue)	1	Remote transmission request
IDE	1	Identifier extension bit
r0	1	Reserved bit for future protocol expansion
DLC (yellow)	4/8/9	Code for number of data bytes (4-bit for standard CAN; 8 or 9 bits for CAN-FD)
Data Field (red)	0-64 (0-8 bytes); 0-512 (0-64 bytes)	Data to be transmitted (0-8 bytes for standard CAN; 0-64 bytes for CAN-FD)
CRC	15	Cyclic redundancy check
CRC Delimiter	1	Assigned recessive (1)
ACK slot	1	Dominant bit if error-free message; recessive to discard errant message
ACK Delimiter	1	Acknowledgement delimiter
EOF	7	End of frame

Table 1. CAN Message Data-Frame Format

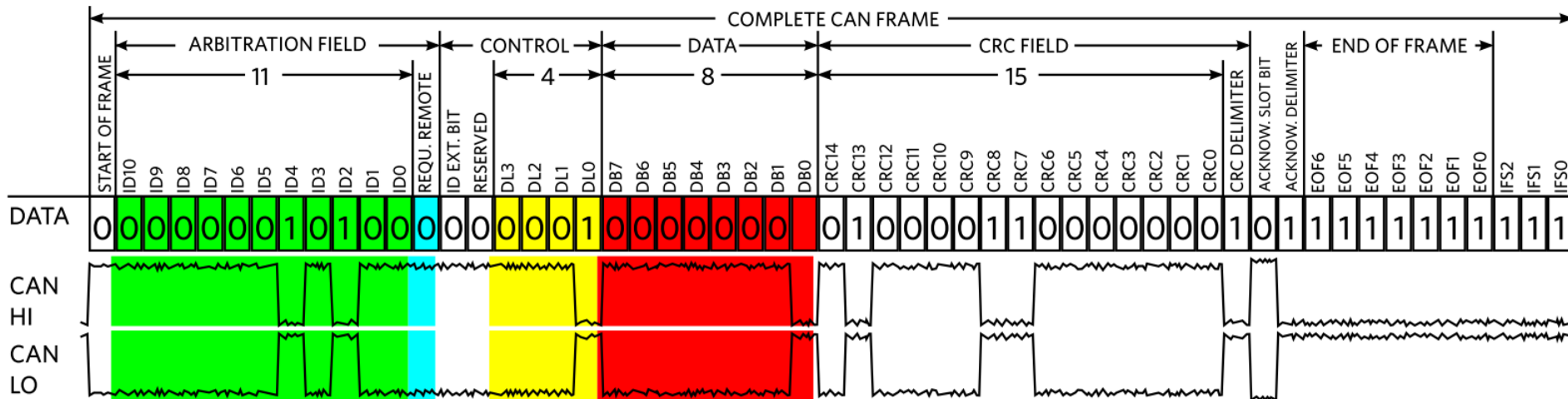
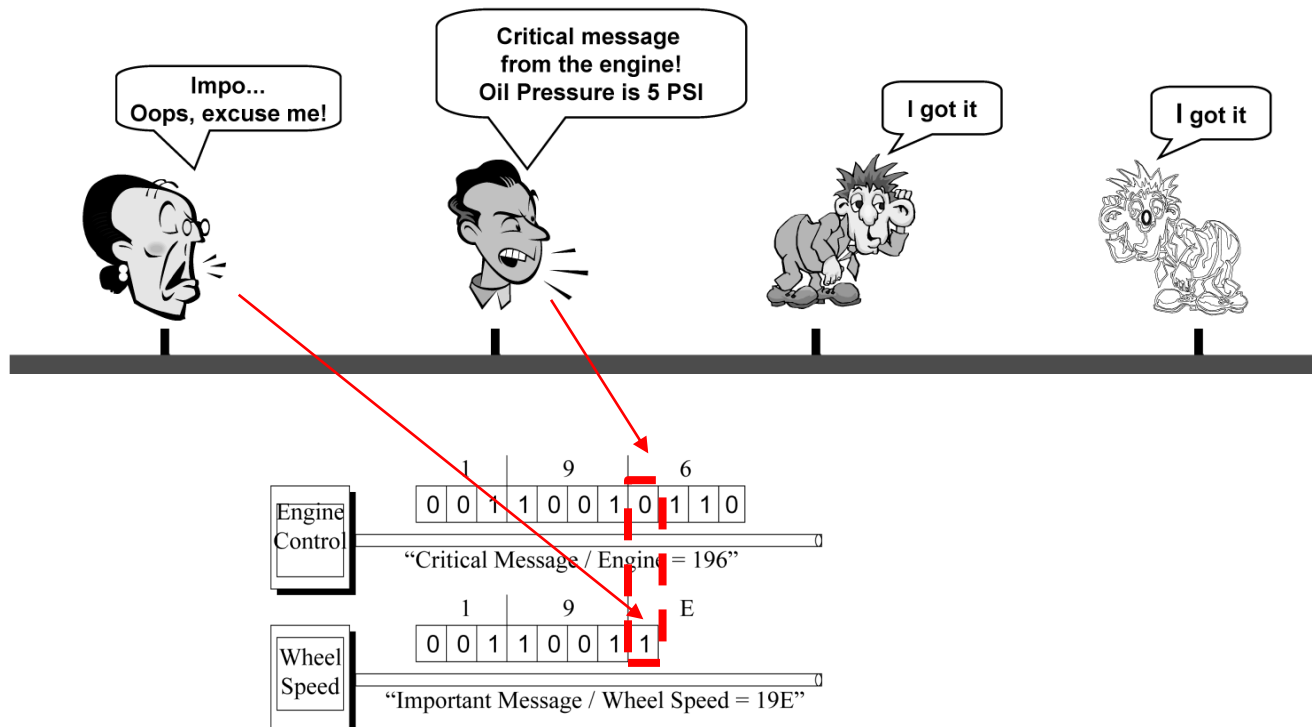


Figure 4. CAN Message Data-Frame Format

# Arbitraža (NDA- Non-Destructive Arbitration)

- ▶ Arbitraža – potrebna, ko več vozlišč poskusi **oddajati hkrati**
- ▶ Do konca sporočila lahko oddaja **samo eden naenkrat**
- ▶ Vozlišča **berejo vodilo** in **oddajajo samo v trenutkih neaktivnosti**
- ▶ Vozlišča z „višjo“ **ID prioriteto** nato nadaljujejo z oddajo.



# Arbitraža (Non-Destructive Arbitration)

- ▶ Pomembnost sporočila je določena z IDjem

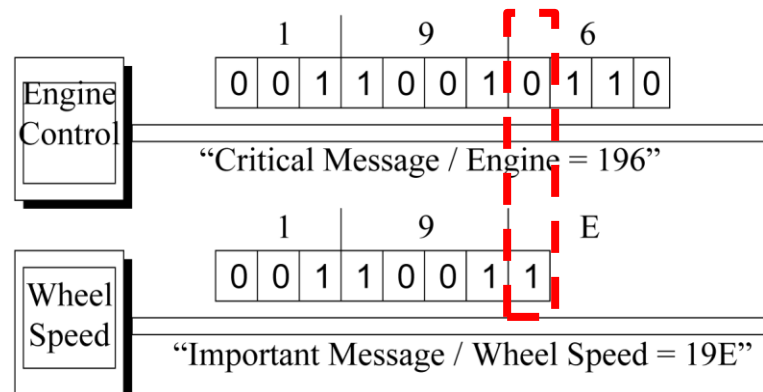
Nižja vrednost = Višja pomembnost

- ▶ Vodilo:

- ▶ “0” (dominantni bit) na vodilu prevlada nad “1” (recesivni bit) na vodilu

- ▶ Naprava:

- ▶ odda “1” in bere “0” → izguba arbitraže
- ▶ odda in bere enako → nadaljuje z oddajo



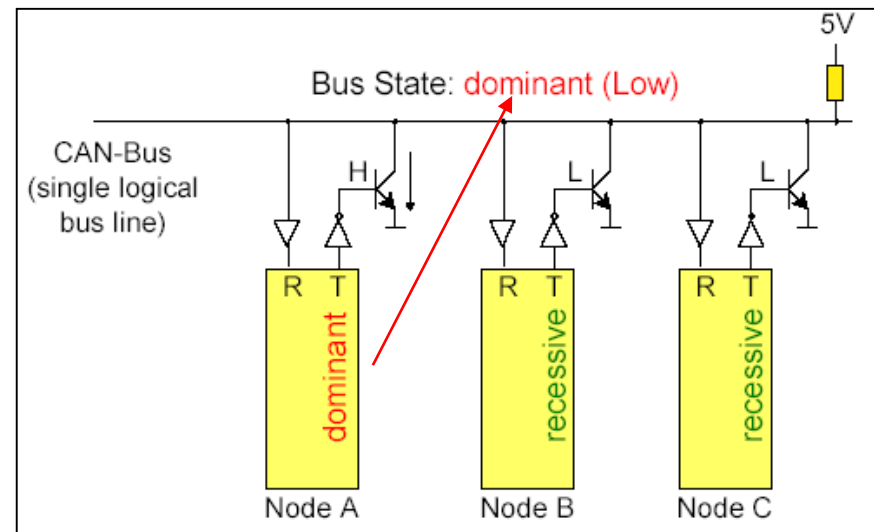
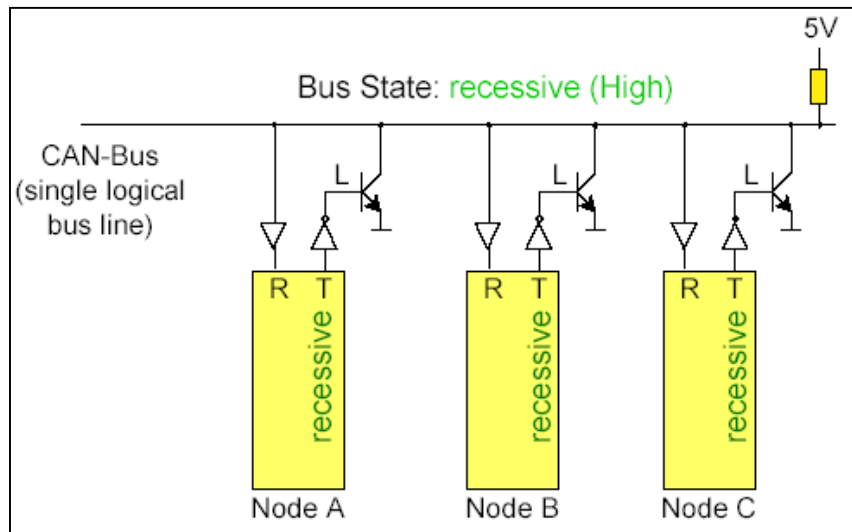


# Wired AND vezava

Stanje "0" (dominantno stanje) na vodilu

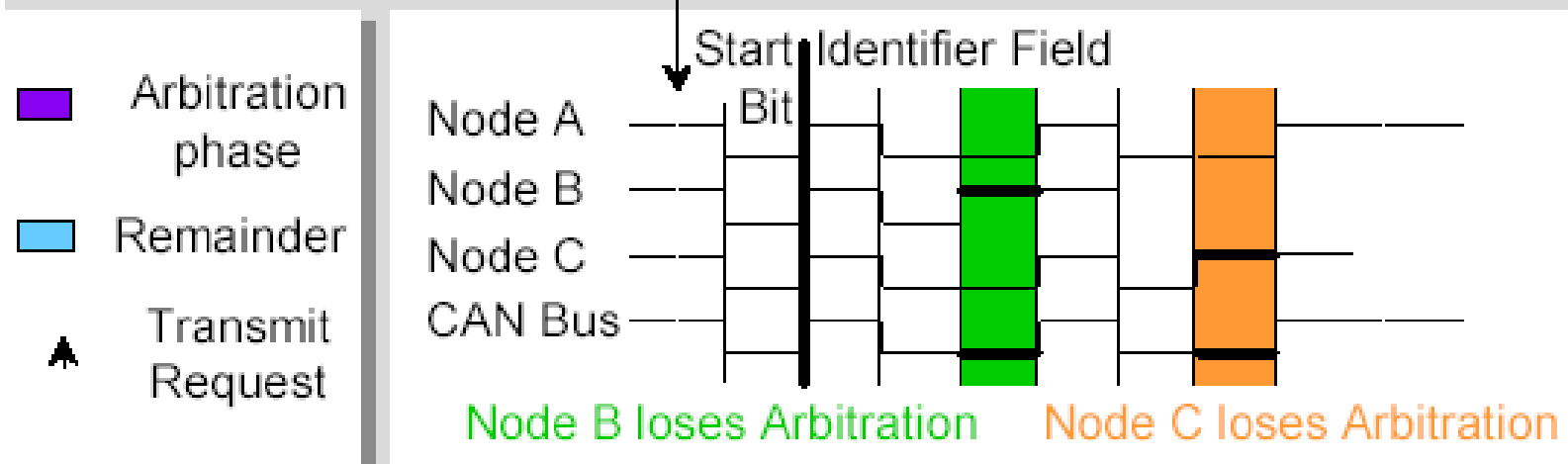
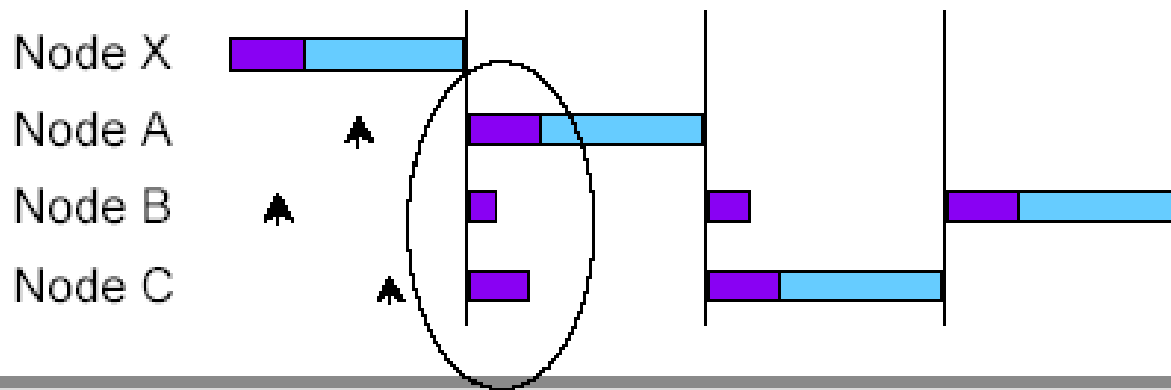
prevlada

ostala stanja "1" (recesivno stanje) na vodilu.



# Primer arbitraže – CSMA/CA NDA

## CSMA/CD NDA – **C**arrier **S**ense **M**ultiple **A**ccess/**C**ollision Avoidance by **N**on **D**estructive **A**rbitration



# Oscilloskop: primer CANbus komunikacije

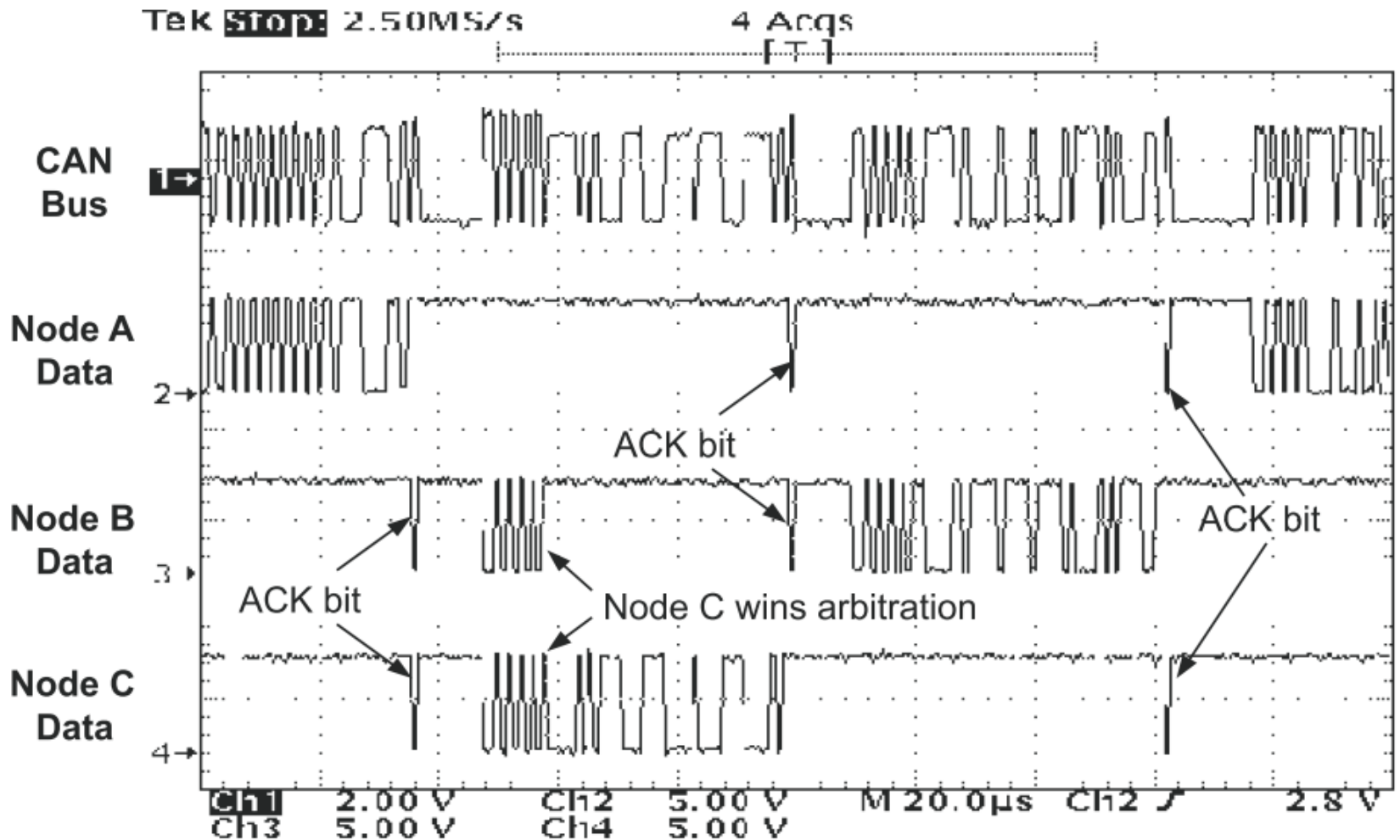
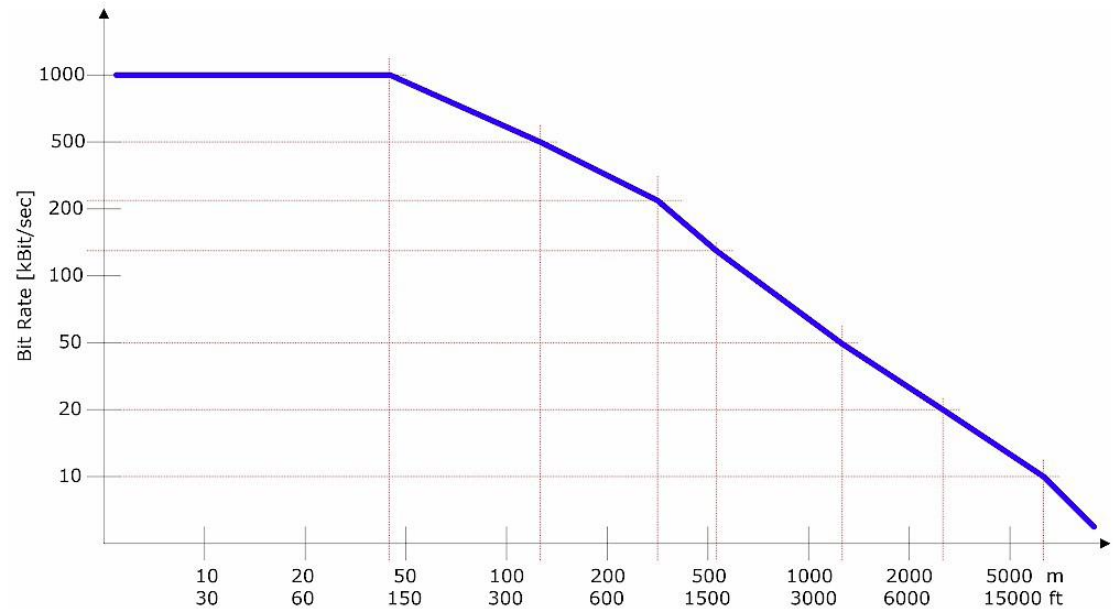


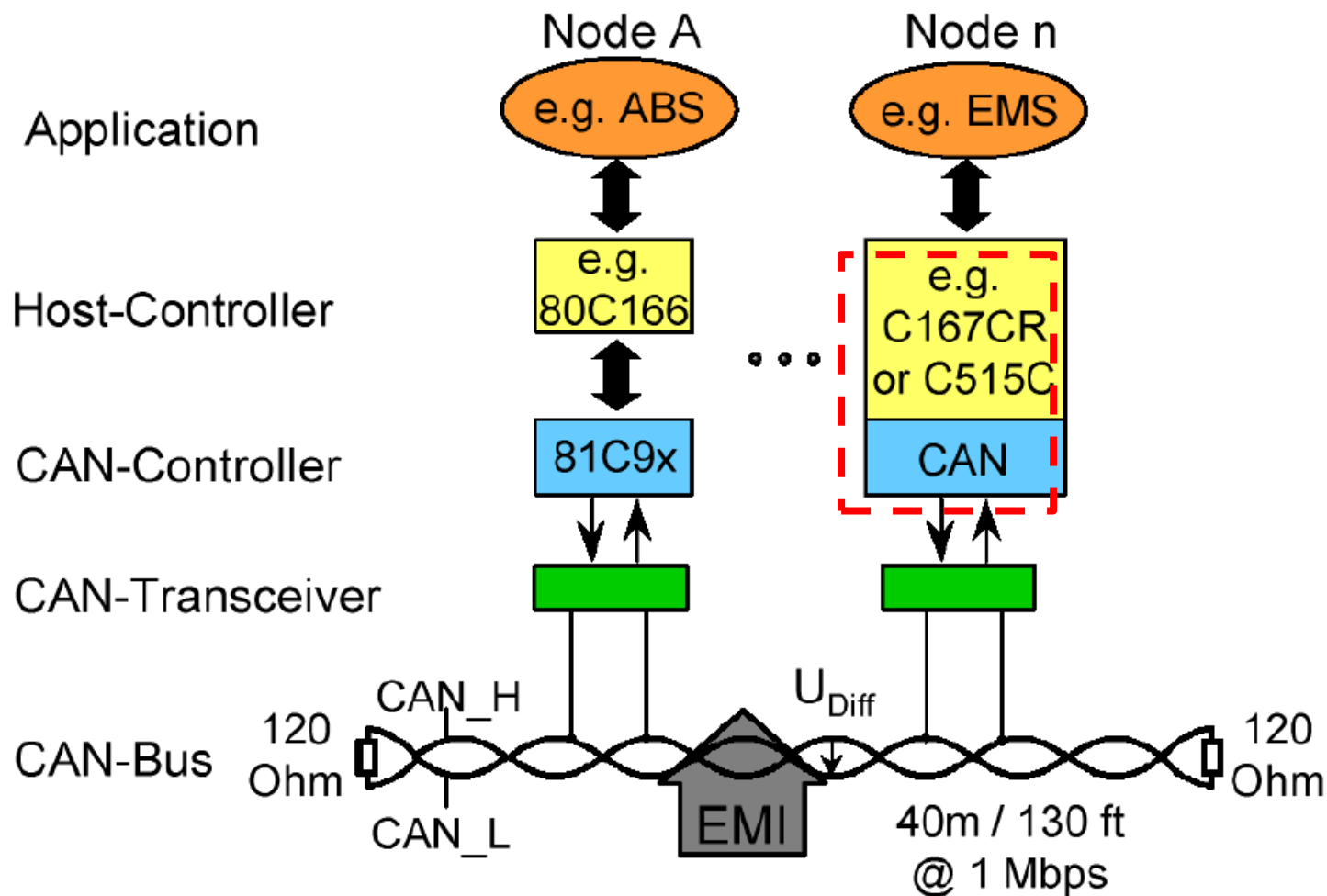
Figure 8. CAN Bus Traffic

# CANBUS - Hitrost komunikacije in razdalja

- ▶ Do 1 Mbit/sec.
- ▶ Standardne hitrosti: 1 MHz, 500 KHz and 125 KHz
- ▶ Max length: do 5000m, odvisno od:
  - ▶ hitrosti
  - ▶ lastnosti povezav
    - ▶ zaključitve, vrsta kabla, topologije, motenj, ...



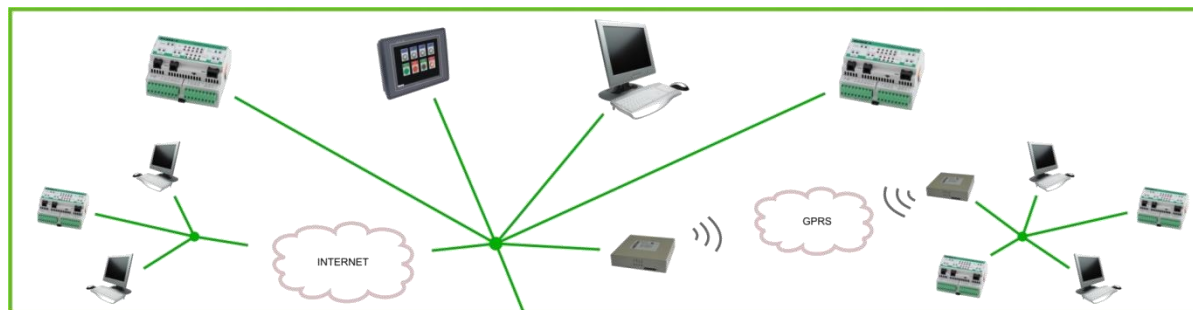
# Možnosti realizacije



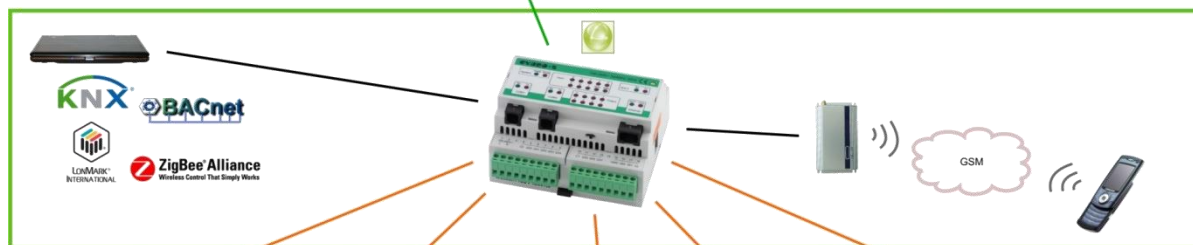
# CANbus v praksi

## INTEGRA BM SYSTEM Industrial & Building Automation

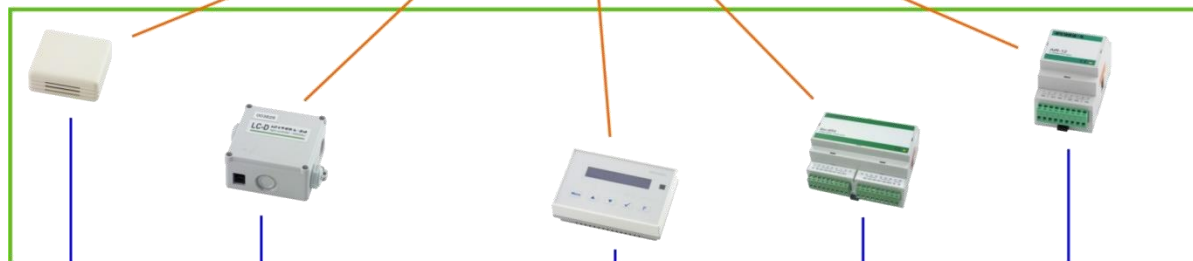
High level network  
(Ethernet, A-Bus,  
Modbus)



CyBro controller



Low level network  
(Canbus)



Dodatki (tipala, daljinci,  
...)



## INTEGRA BM SYSTEM Bus length

## Dolžina, hitrost in topologije

Regarding bus length, two points must be considered:

### 1. Voltage drop

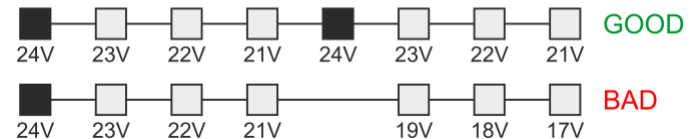
Wire resistance cause voltage drop, which depends of cable length, wire diameter and power consumption. **Cable must be selected** to ensure each module have at least the minimum specified voltage.

### 2. Signal delay

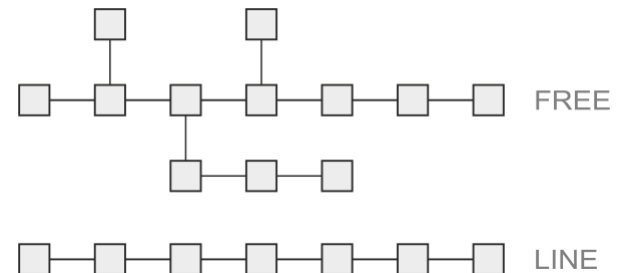
Communication speed is limited with propagation time and bus topology. With **default 100kbps baudrate, 100m is safe without restrictions**. For a longer distance, cable must be connected in **a line (without trunks) and properly terminated**.

Speed\Topology	FREE	LINE
100kbps	100m	300m
50kbps	200m	500m
20kbps	500m	1000m

Secondary power supply



Network topology



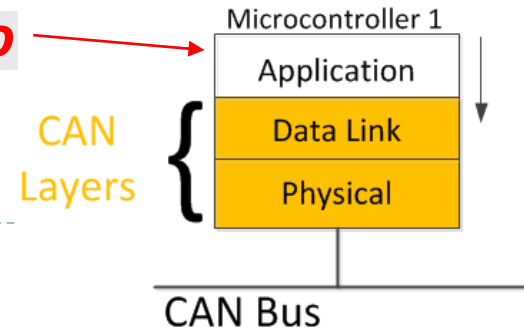
## INTEGRA BM SYSTEM

### IEX protocol

(nadgradnja CANBUS)

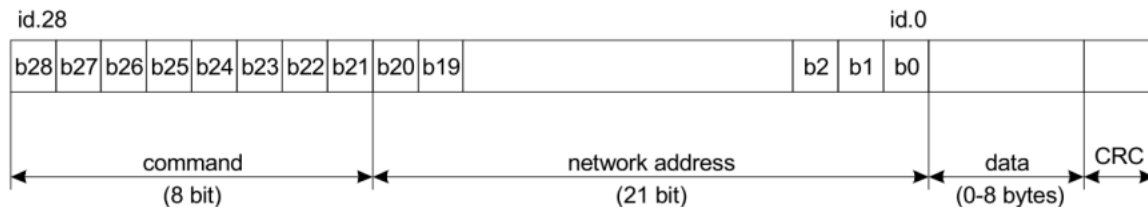
IEX PROTOCOL v2.8

POVZETEK

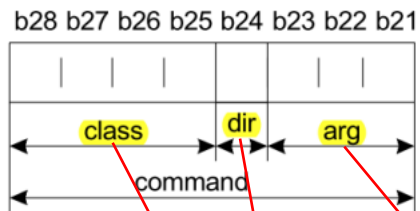


### General

IEX-2 is based on CAN 2.0B. Message format is defined as follows:



### Command summary



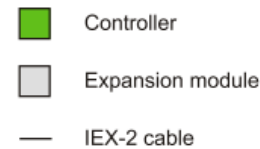
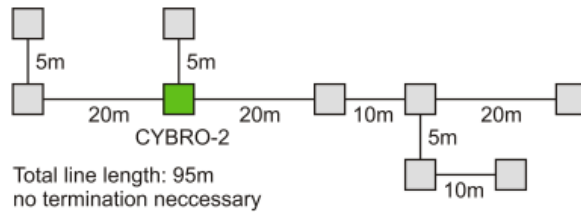
NAD – unikatni naslov IEX modula

command	class	dir	command	arg	data bytes	description	PCAN view
	0000						
	0001						
	0010						
IX_DATA	0011	1		xxx	data(1..4)	binary inputs	070-07Exxxxxh
QX_DATA		0		xxx	data(1..4)	binary outputs	060-06Exxxxxh
	0100						
	0101						
	0110						
IW_DATA	0111	1		xxx	data(2..8)	analog inputs	0F0-0FExxxxxh
QW_DATA		0		xxx	data(2..8)	analog outputs	0E0-0EExxxxxh
BAUDSYNC	1111	1		111	-	autobaud sync msg	1FFFFFFFh

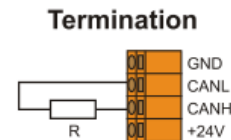
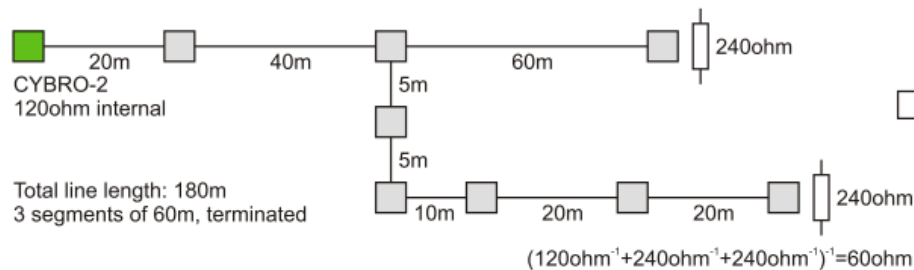


### Cabling topology & termination

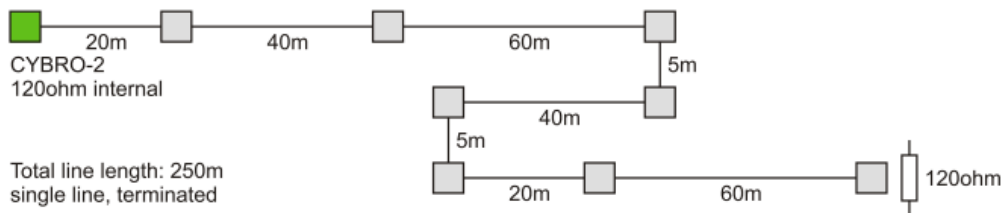
#### 1) Total IEX-2 bus length <100m



#### 2) 100m < Total IEX-2 bus length <200m



#### 3) 200m < Total IEX-2 bus length <300m



# CENTRALNI KRMILNIK CYBRO-2 Controller



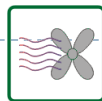
## Primer centralnega krmilnika



# Primeri modulov

## IEX MODULE

FC



### FC

fan coil module

#### SPECIFICATIONS:

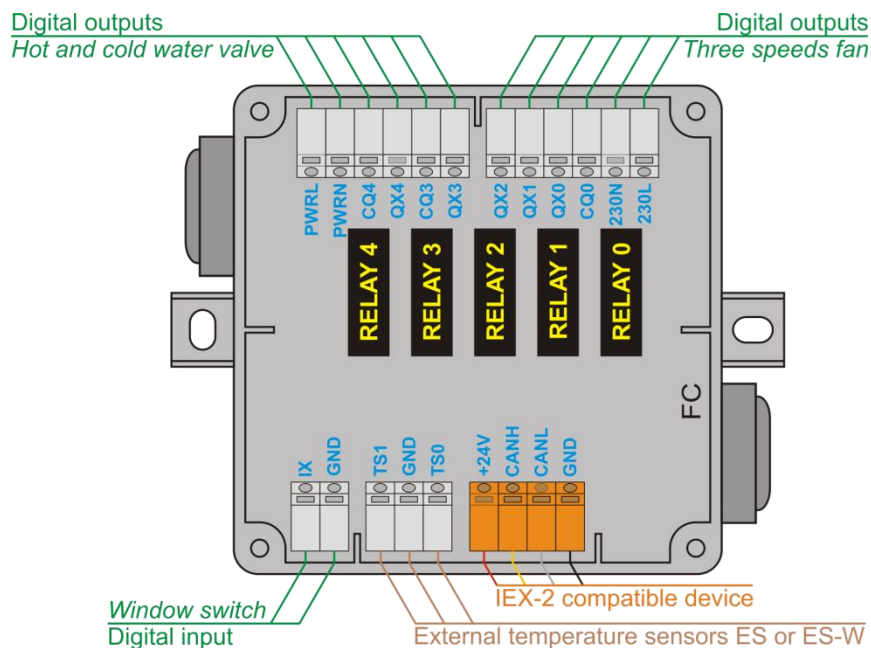
- 1 x digital input
- 5 x relay output
- 2 x input temperature measurement
- 24V DC power supply consumption: 110mA

#### MECHANIC:

field mountable

#### TYPE:

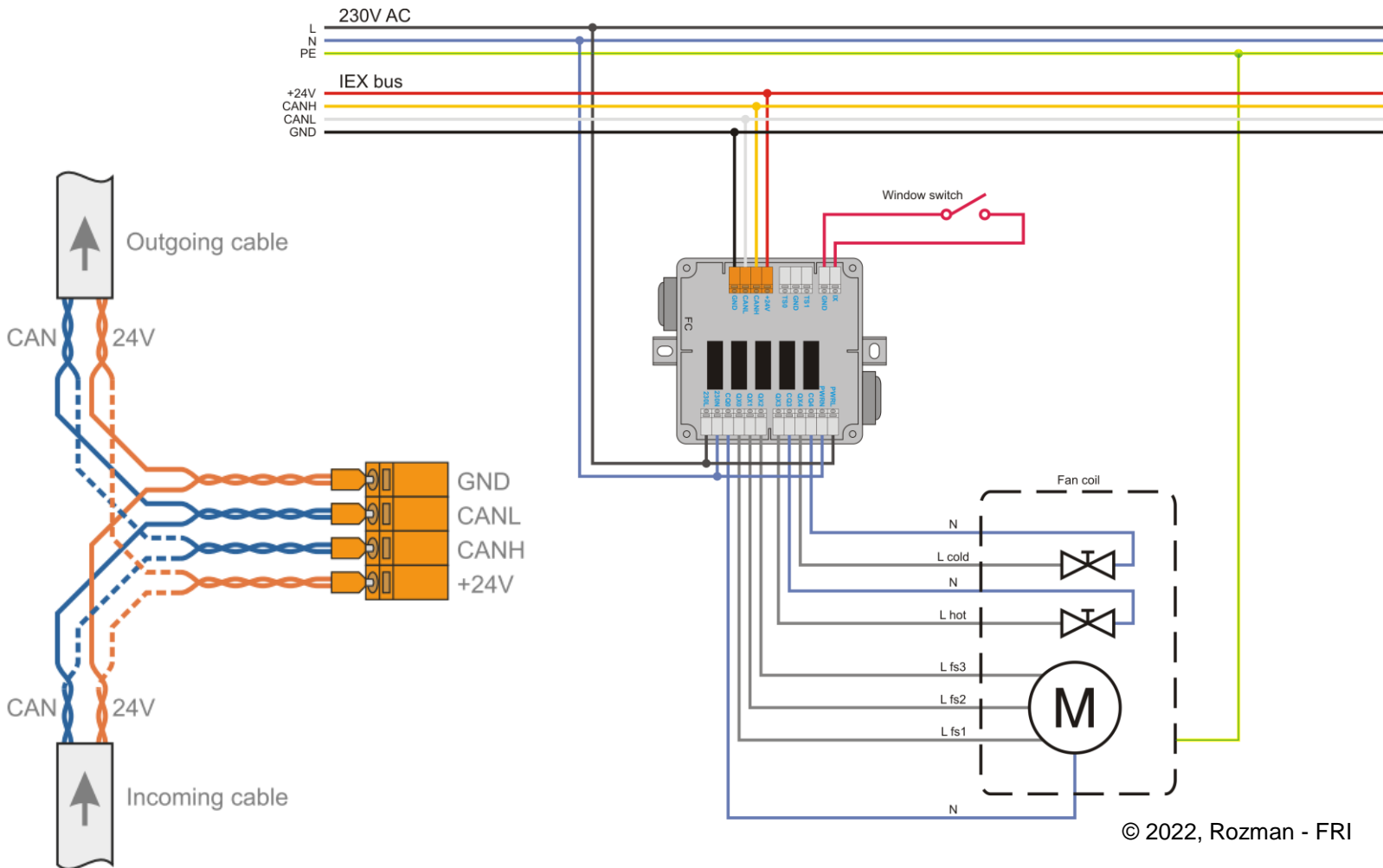
FC-FB



# Primeri modulov

## IEX MODULE FC

FC

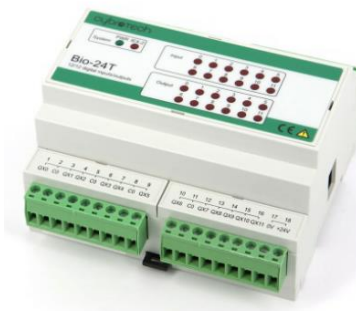
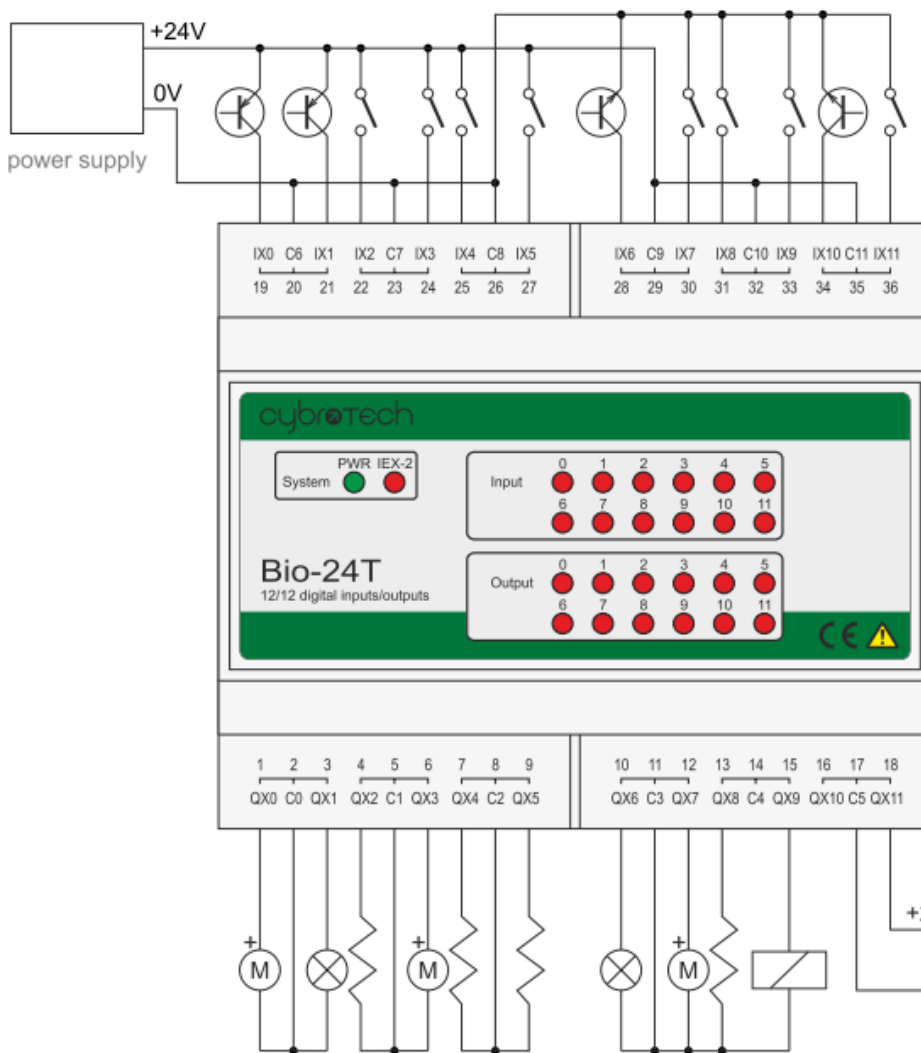


# Primeri modulov

## IEX MODULE Bio-

## Bio-24T

Wiring diagram



IEX-2 module  
12 opto-isolated PNP transistor  
outputs 1A  
12 opto-coupler inputs 24V

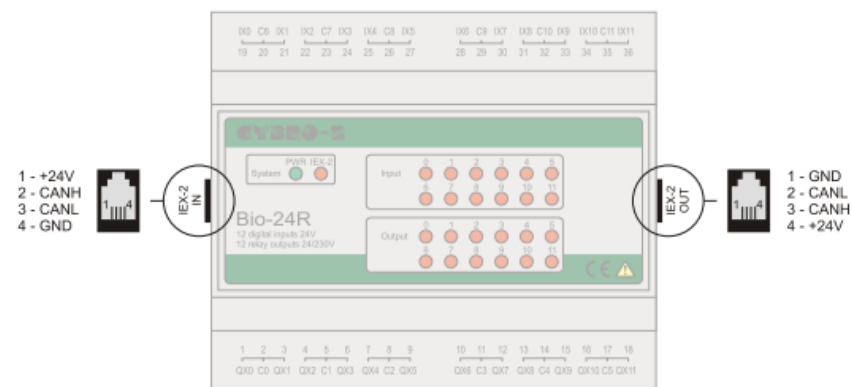
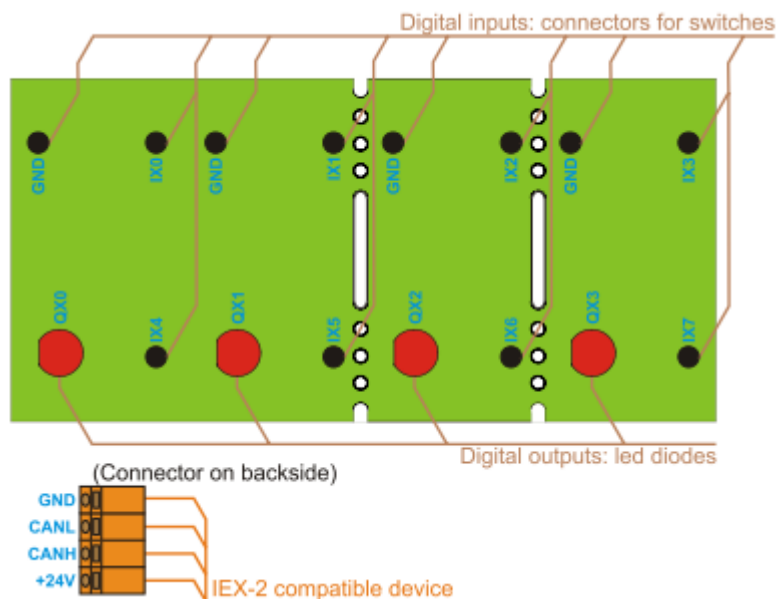


Figure 3: IEX-2 input and output ports.

# Primeri modulov

## IEX MODULE SW-

## SW-L



IEX-2 module  
4 switches  
4 LED illuminations  
Designed for Legrand, Bticino and TEM switches



### Technical specifications

IX (8 digital inputs)	for connecting 4 switches
Current	2.5mA/12V
QX (4 digital outputs)	
Led illumination	3mm red led-diodes
Power supply	24V DC (18..26V DC), over IEX-2 bus
Power consumption	40mA
Mounting	2 x switch: flush box (diameter 60mm, depth 55mm), in wall 3 x switch: flush box (size 95x58mm, depth 49mm), in wall 4 x switch: flush box (size 120x58mm, depth 49mm), in wall
Dimensions	89x44x38mm



CyPro v2.7.6 - C:\Users\R\Documents\Sluzba\Vaje\VIN\_Vh\_Izh\_naprave\VIN\_2016\_17\Vaje\13 Labvaja LV5\_Canbus\VIN\_vaje.cyp

File Edit View Project Program Tools Window Help

New Open Save Print Cut Copy Paste Environment Configuration Hardware Allocation Masks Sockets Send Monitor Start Stop

Project Tree

- Project: VIN\_vaje.cyp
  - Program: New Program
    - Hardware
    - Masks
    - Sockets
    - ST: function main:void;
    - Description

Local Allocation

Name	Type	Attributes	Description
main			

New Program - ST: function main:void;

```
if fp(clock_10s) then
  bio00_qx00 := !bio00_qx00 ;
end_if ;

bio00_qx01 := !bio00_qx00 ;

if fp(bio00_ix00) then
  bio00_qx02 := !bio00_qx02 ;
end_if ;

if fp(sw00_ix01) then
  bio00_qx00 := !bio00_qx00 ;
end_if ;

if fp(clock_10ms) then
  bio00_qx02 := !bio00_qx02 ;
end_if ;
```

Online Monitor

Monitor01

History	Variable name	Type	Value	Base
	clock_10s	bit		0 Dec
	bio00_ix00	bit		0 Dec
	bio00_ix01	bit		0 Dec
	bio00_qx00	bit		0 Dec
	bio00_qx01	bit		0 Dec
	bio00_qx02	bit		0 Dec
	sw00_ix01	bit		0 Dec

Speed: 50ms (16s total)

Close

# Tinia – prijazen dom

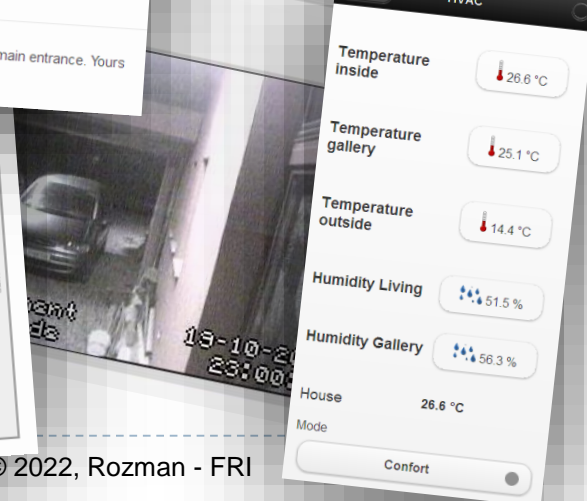
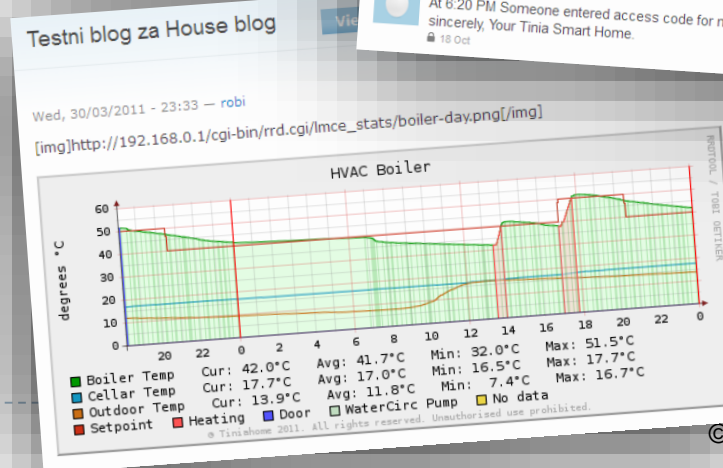
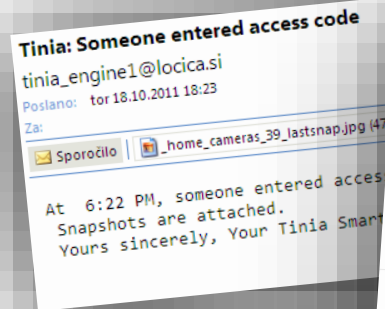
## Kratek opis



### TBS – „Tinija Building Server“:

*Nadzor, upravljanje in vizualizacija delovanja prijaznega doma.*

- majhen, varčen, tih (5W)
- povezuje zgradbo in pametno mesto
- informiranje, povratna inf.
  - pametni telefoni, tablice
  - splet, soc.omrežja
- programiranje s pravili, vtičniki
- povezava s soc.omrežji
  - Twitter, FaceBook



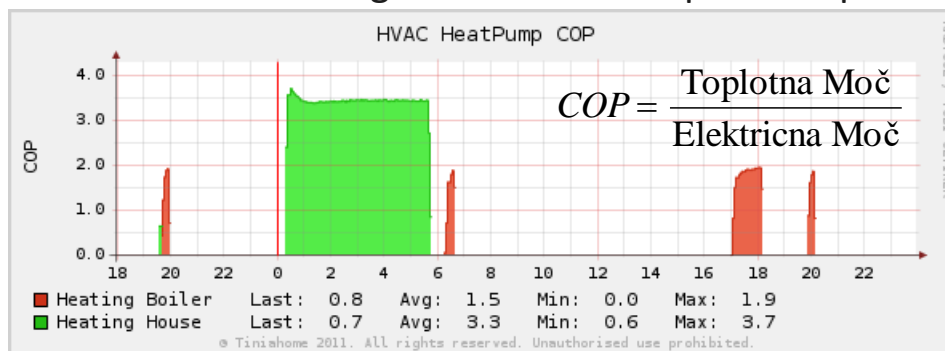
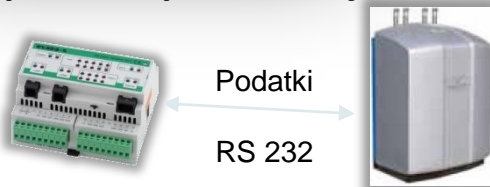


# Ogrevanje, prezračevanje



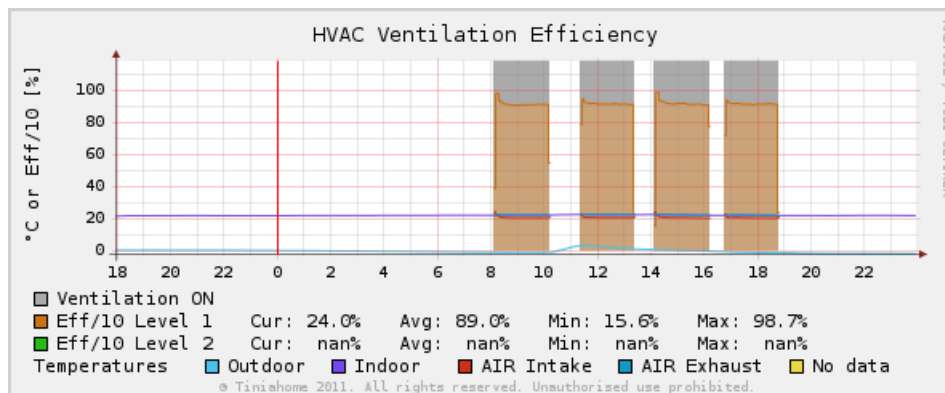
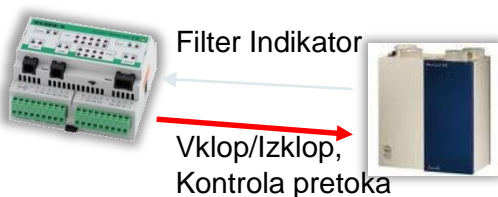
Primer zimskega dneva – COP toplotne črpalke :

Toplotna črpalka zemlja-voda



Primer zimskega dneva – učinkovitost rekuperacije

Prezračevanje s povratkom toplote (rekuperacija)



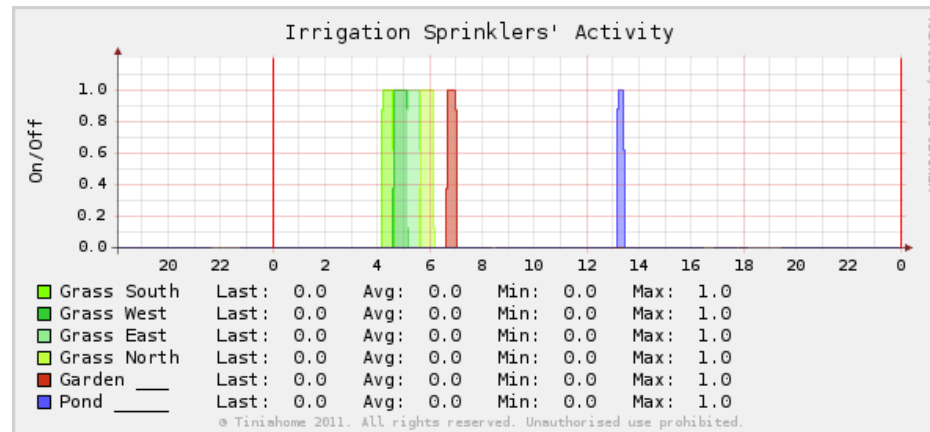
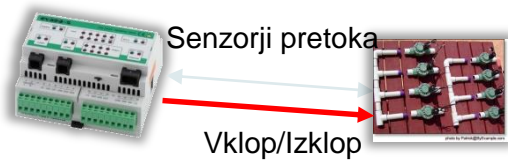
$$EFF. \approx \frac{\text{Svež zrak temp.} - \text{Zunanji zrak temp.}}{\text{Izčrpažrak temp.} - \text{Zunanji zrak temp.}} [\%]$$

# Zalivanje (vrt, zelenica, ribnik)



Primer poletnega dne - Zalivanje

Zalivanje  
(vrt, zelenica, ribnik)

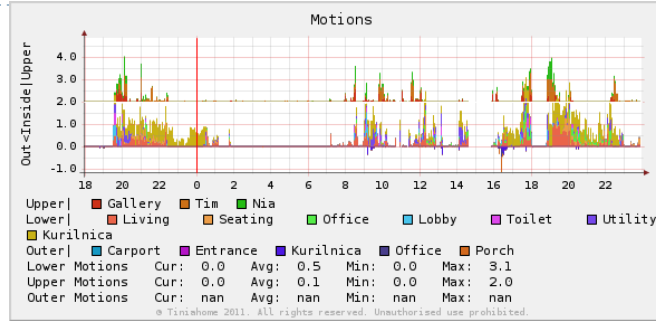
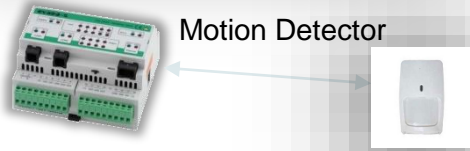


# Tipala

## Zaznave gibanja v hiši - primer

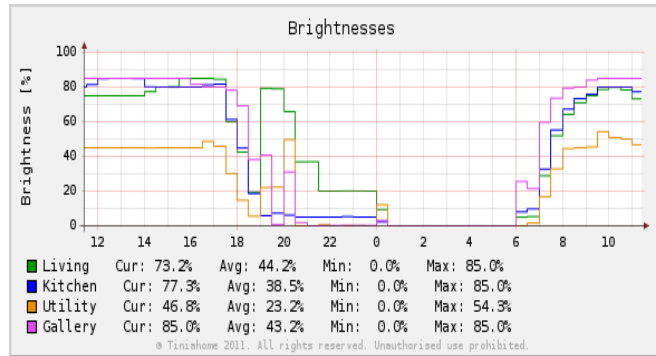


### Detekcija gibanja/prisotnosti

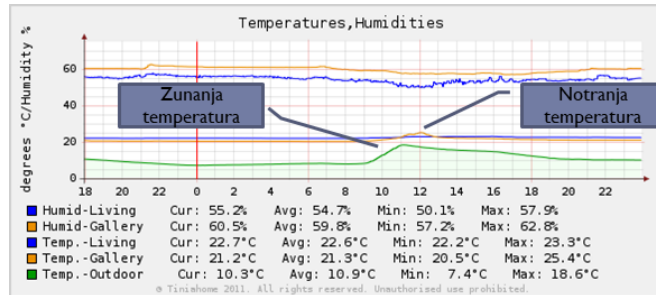


### Osvetljenost prostorov

### Meritve zveznih vrednosti



### Temperatura, vlaga

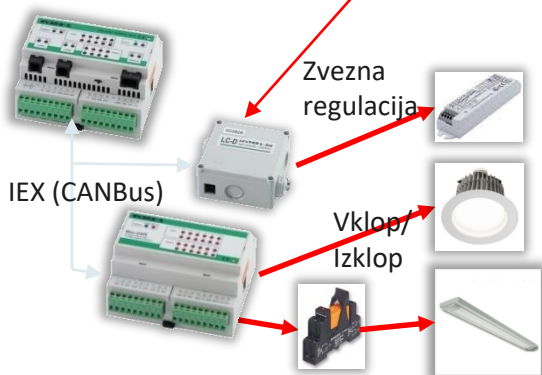




# Praktični primer nadzora in vizualizacije: Osvetlitev (aktuatorji)

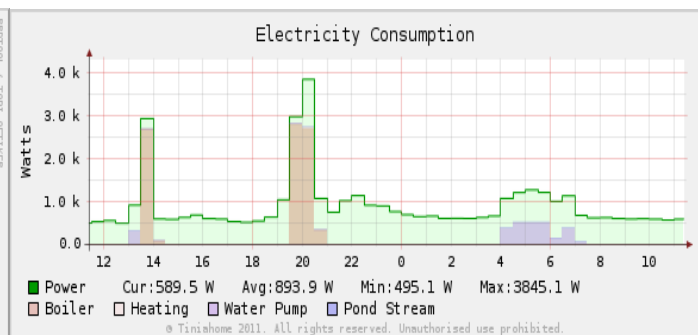
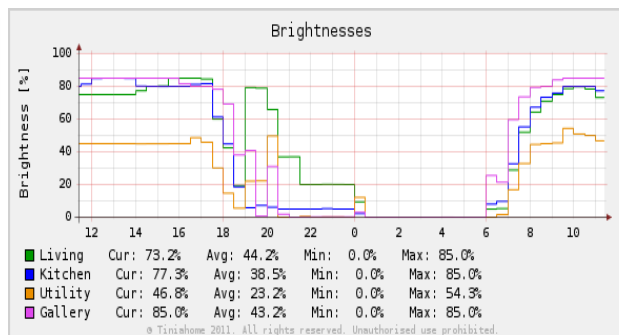
## Vklop/Izklop in zvezna regulacija razsvetljave

- Vklop/Izklop BIO-24R in BIO-24T
- Zvezna regulacija s pomočjo **DALI balastov** - Cybrotech modul LC-D



- Luči se upravljajo v skupinah
- Običajno krmiljena s pomočjo scen in zaznavanje osvetljenosti:
  - Statične scene** – npr. : Prehrana, Obisk, Romantika, TV, Branje, Relaksacija, ...
  - Dogodkovne scene**: Ko se vklopi TV, nastavi bližnjo luč na 20%.
- Zmanjševanje porabe :
  - Časovne luči** (izklopi po določenem času odsotnosti)
  - Vklopi luč samo, ko je to **res potrebno** (trenutna osvetljenost)
  - Nastavi zvezne luči samo na **potrebno stopnjo** (glede na osvetljenost)

## Primer meritev osvetljenosti in nadzora porabe el. energije (glavni porabnik el. energije so posebej izpostavljeni)



# Pasivno ogrevanje/hlajenje...



Rolete, Žaluzije, Okna

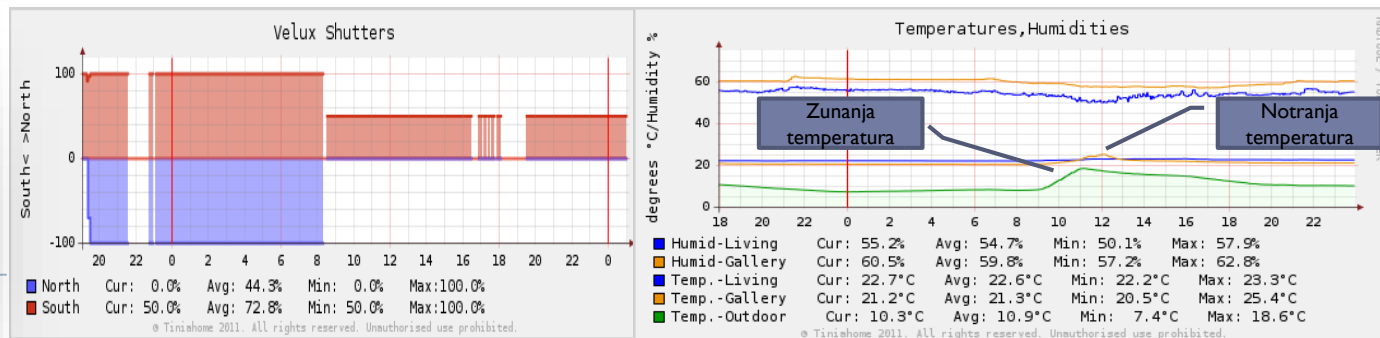
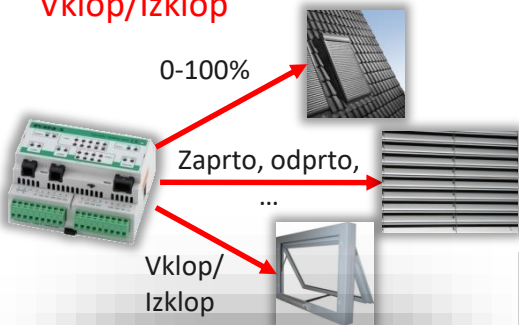
•Rolete: med 0% - 100%  
(0% odprte, 100% zaprte)

•Žaluzije imajo stanja :  
Zaprto(100%), Senčeno(75%),  
Odprto(50%), Solarno pasivno  
(25%), Dvignjeno(0%).

•Motorizirana okna:  
Vklop/Izklop

- Strešna okna z roletami :
  - severna, običajno:
    - Odprta v toplem vremenu za boljšo osvetlitev (poletje)
    - Zaprta v hladnem vremenu za ohranjanje toplote (zima)
  - južna, običajno:
    - Odprta v hladnem, sončnem vremenu za pasivno ogrevanje (zima, pomlad)
    - Zaprta v vročem vremenu proti pregrevanju (poletje)
- Žaluzije:
  - Senčene ali zaprte ob izrazitem sončnem vremenu poleti
  - Odprte v “solarni” poziciji ob sončnih dnevih pozimi
- Motorizirana okna (s komarniki) :
  - Odprta v poletnih nočeh za pasivno ohlajanje

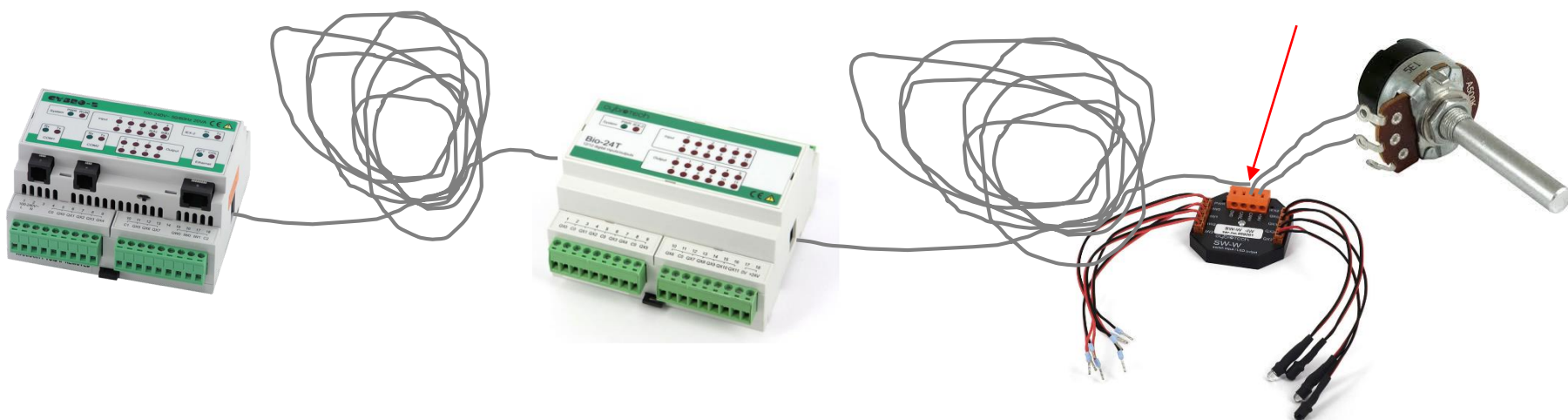
Primer stanj rolet in temperatur v sončnem zimskem dnevu:



# Lab. vaja – meritve CANBUS

Izmerite stanje na vodilu pri :

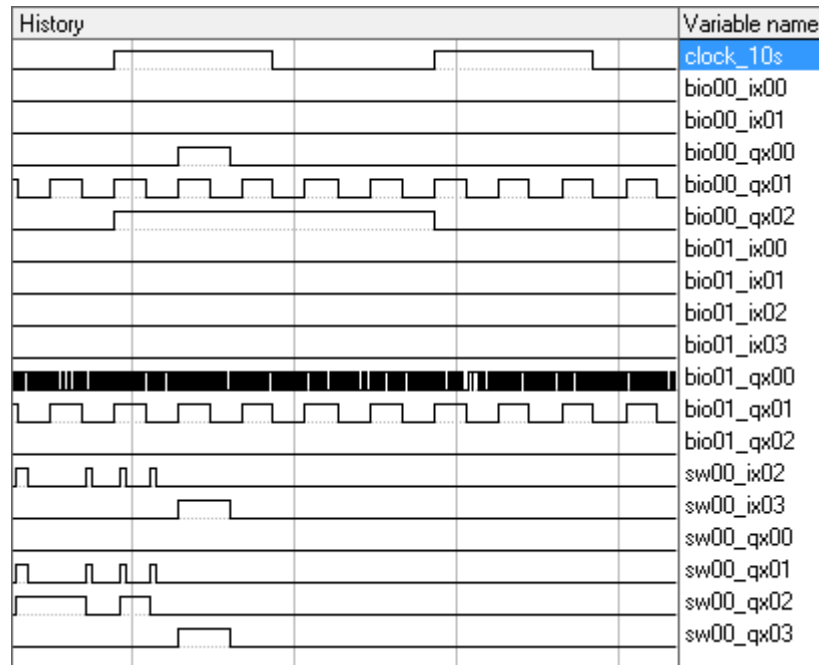
- Različnih zaključitvah na koncu vodila
  - Odprte sponke, 500ohm, zaključitev (107ohm)
- Dveh različnih bitnih hitrostih (500kb/s, 100kb/s)





# Lab. vaja – kontrolni program

## Monitor



## Program

```
// Periodic tasks
if fp(clock_10s) then
    bio00_qx02 := !bio00_qx02 ; // Red LED every 10 secs
end_if ;

if fp(clock_1s) then
    bio00_qx01 := !bio00_qx01 ; // Red LED every 1 sec
    bio01_qx01 := !bio01_qx01 ; // Red LED every 1 sec
end_if ;

if fp(clock_10ms) then
    bio01_qx00 := !bio01_qx00 ; // Red LED every 10 msec
end_if ;

if fp(bio00_ix00) then
    bio00_qx02 := !bio00_qx02 ; // Red LED on keypress
end_if ;

// SW Switch -> LED indicator & ventilator
sw00_qx03 := sw00_ix03;
bio00_qx00 := sw00_ix03;

sw00_qx01 := sw00_ix02; // SW Key -> LED indicator

if fp(sw00_ix02) then
    sw00_qx02 := !sw00_qx02 ; // SW Key -> change LED indicator
end_if ;
```

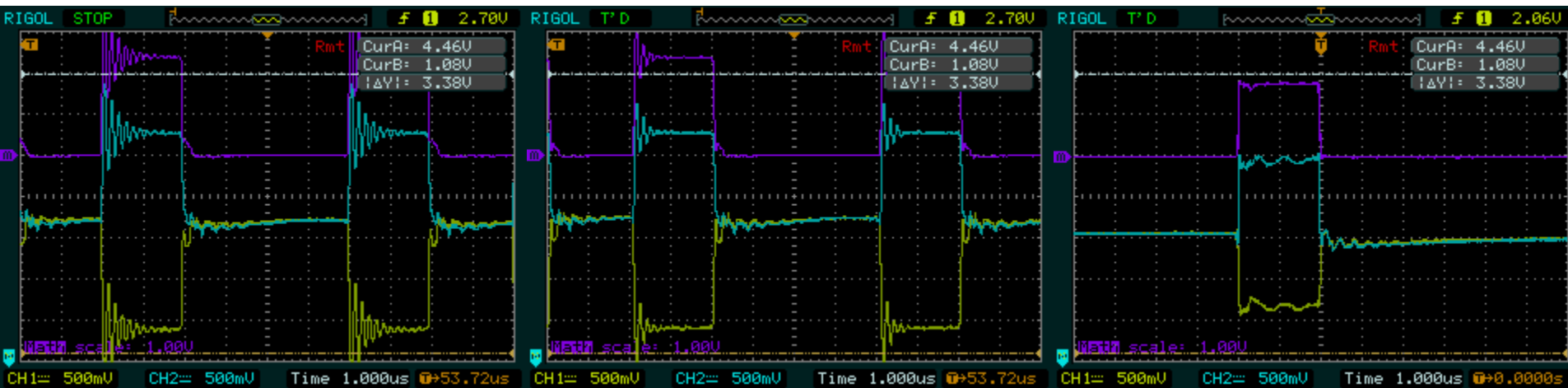
# Lab. vaja – meritve

500kb/s:

Odprte sponke

500ohm

107ohm



3 zavitki UTP kabla s spojniki – cca 40m...





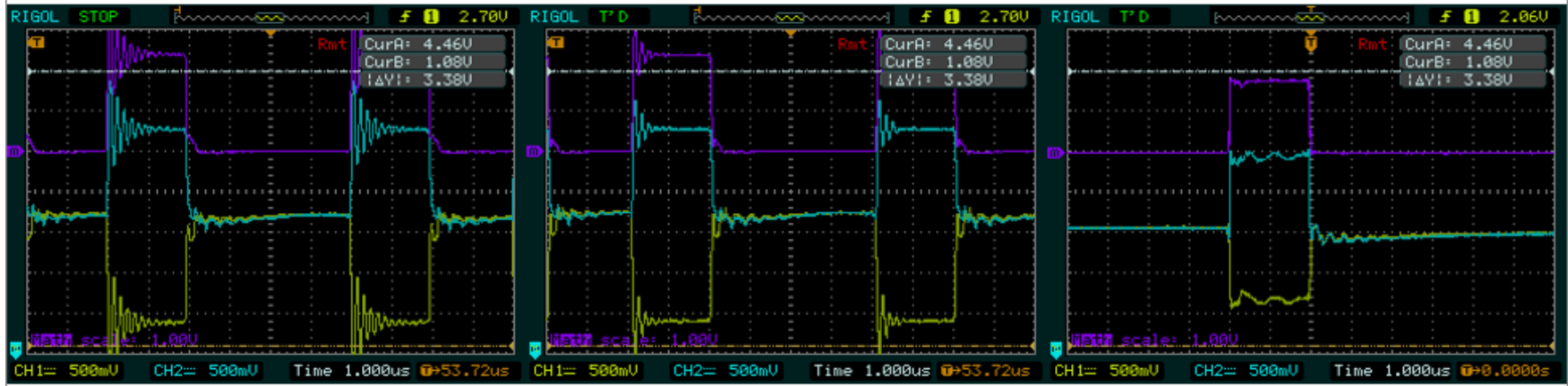
# Lab. vaja – primerjava hitrosti

500kb/s:

Odprte sponke

500ohm

107ohm

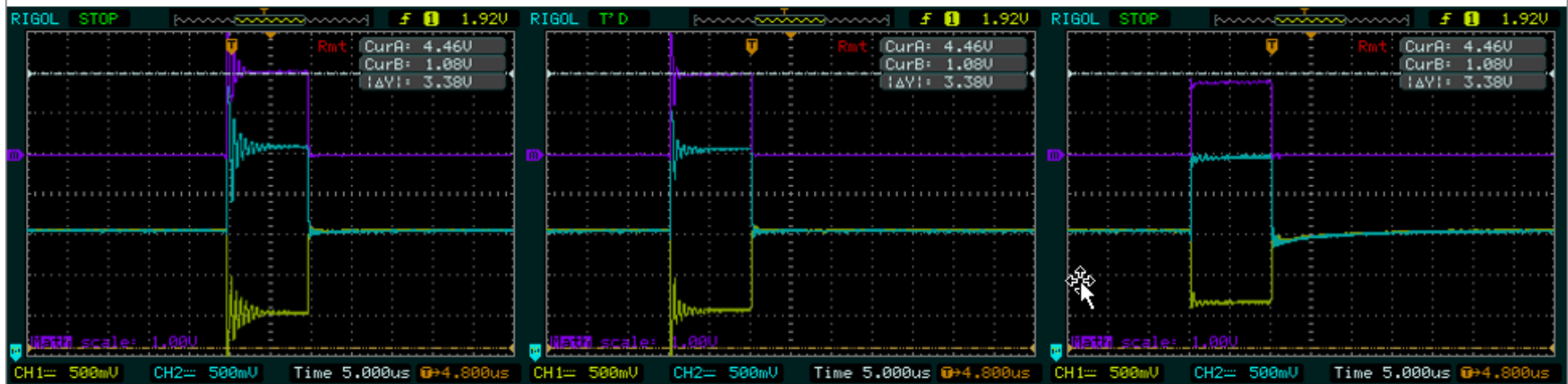


100kb/s:

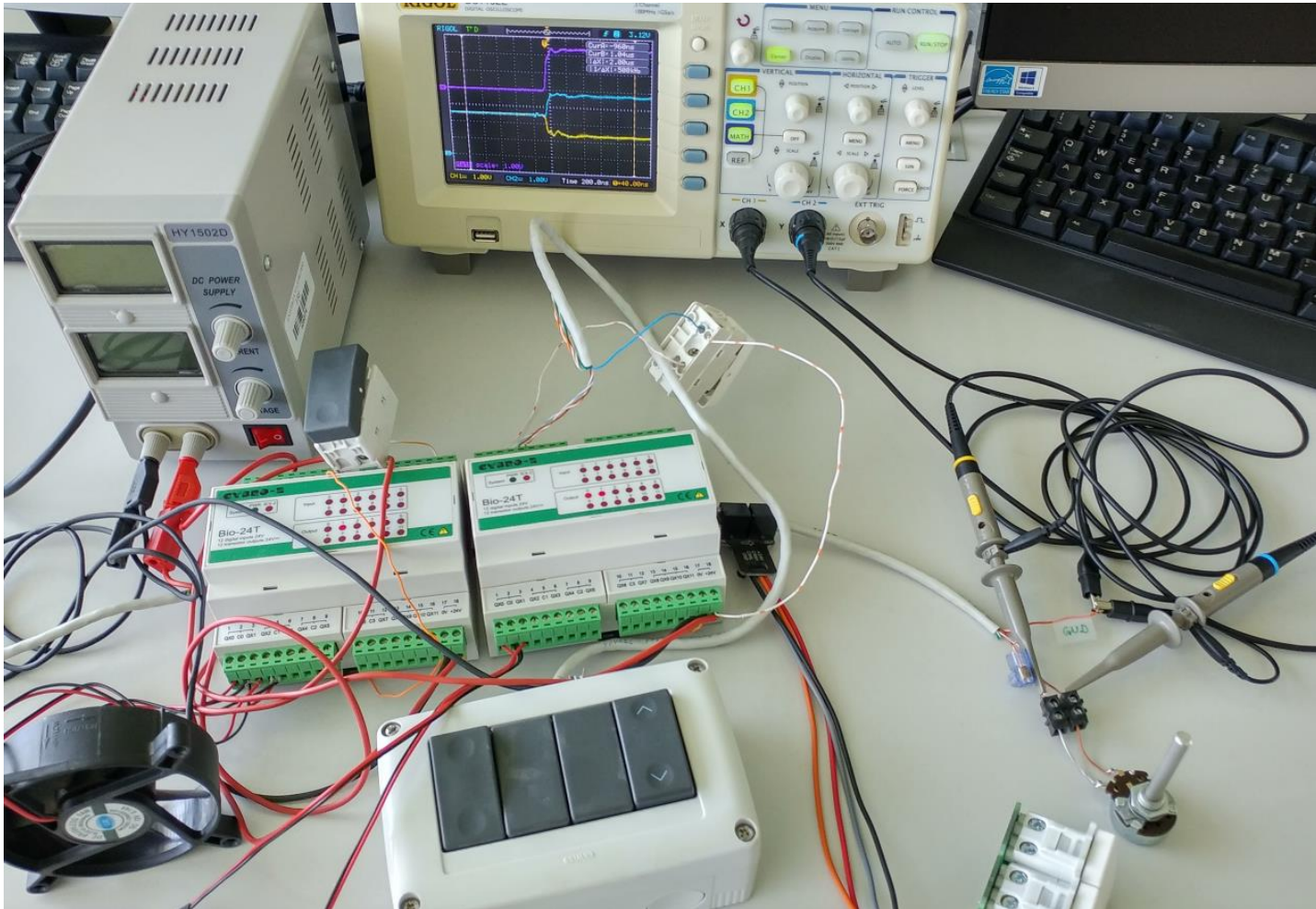
Odprte sponke

500ohm

107ohm

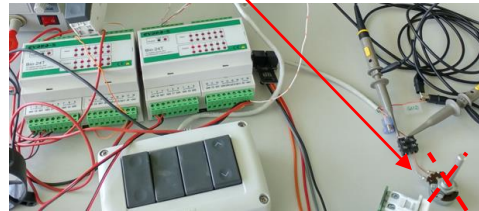


# Lab. vaja - meritve

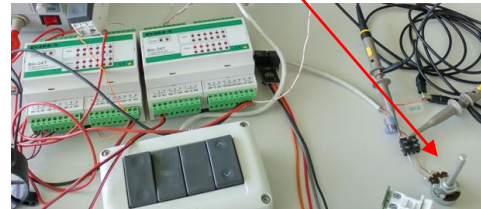


# Lab. vaja - meritve

Nezaključena linija

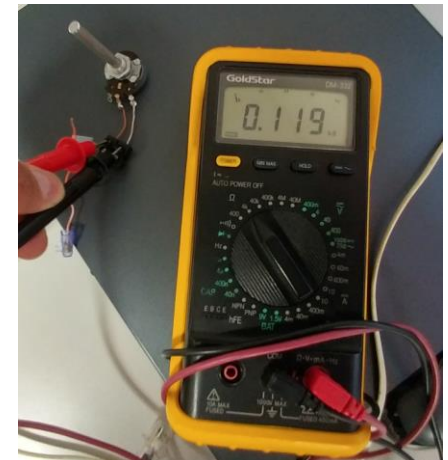


Zaključena linija



Ročna meritev

$R_0$





# Lab. vaja – STM32 kot IEX modul

## STM32F4 Discovery + shield MikroE



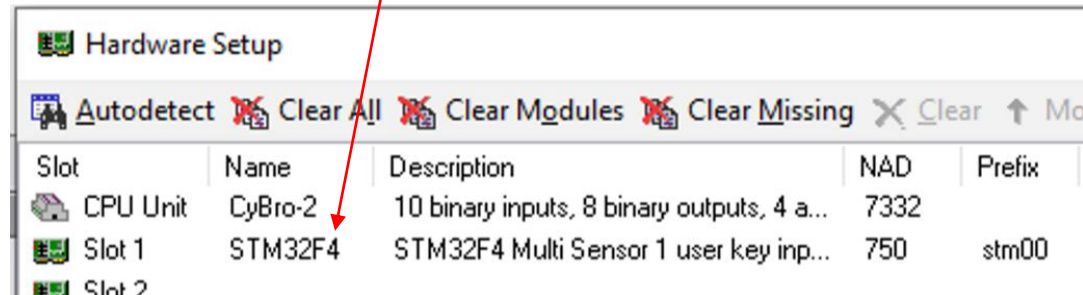
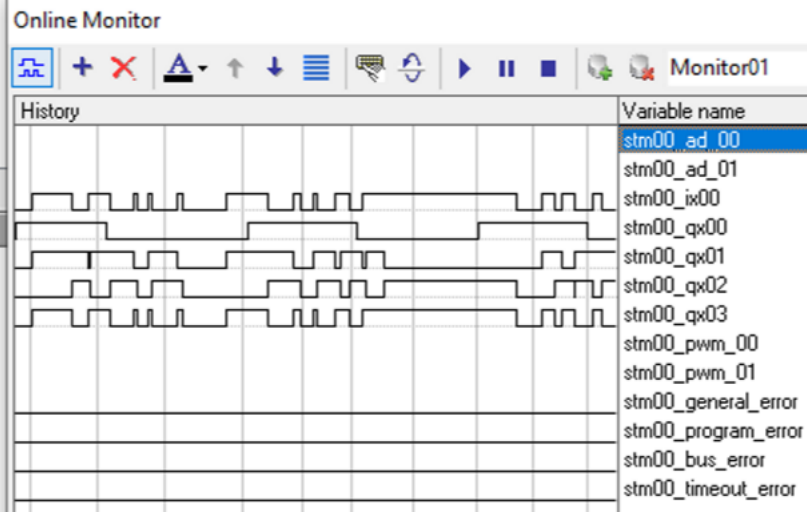
New Program - ST: function main: void;

```
if fp(clock_10s) then  
  stm00_qx00:=!stm00_qx00;  
end_if;
```

```
if fp(stm00_ix00) then  
  stm00_qx01:=!stm00_qx01;  
end_if;
```

```
if fn(stm00_ix00) then  
  stm00_qx02:=!stm00_qx02;  
end_if;
```

```
stm00_qx03:=stm00_ix00;
```



Hardware Setup

Autodetect  Clear All  Clear Modules  Clear Missing  Clear  Mo

Slot	Name	Description	NAD	Prefix
CPU Unit	CyBro-2	10 binary inputs, 8 binary outputs, 4 a...	7332	
Slot 1	STM32F4	STM32F4 Multi Sensor 1 user key inp...	750	stm00
Slot 2				

The figure shows a hardware setup window with a table of slots. A red arrow points from the 'STM32F4' entry in the table to the 'stm00' prefix in the code block above.

# RS-485 vs CANBUS

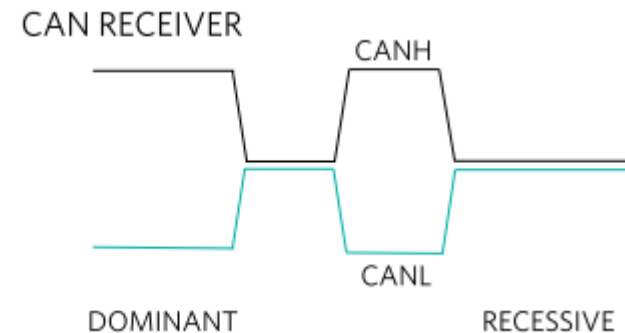
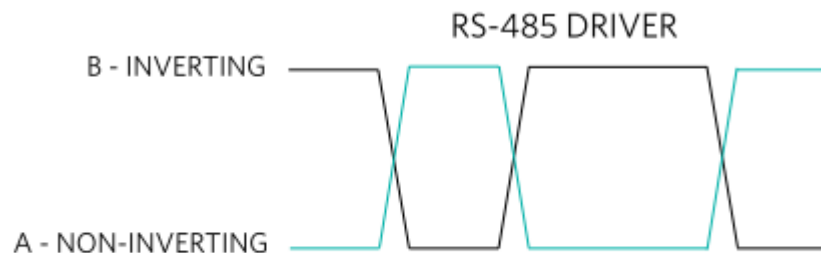
## Podobno/enako:

- ▶ Diferencialni prenos
- ▶ Multi-master
- ▶ Zaključitev 120Ω
- ▶ Različno

## Različno (prednosti CANBUS) :

- ▶ Multi-master oddajanje
  - ▶ CANBUS arbitraža
  - ▶ RS485 –konflikt, poraba toka, segrevanje
- ▶ Dodatna preverjanja (nivo sporočila)
  - ▶ CRC, format sporočila
- ▶ Dodatna preverjanja(bitni nivo)
  - ▶ Spremljanje stanja linije (poslano/sprejeto)
  - ▶ Potrditev (Acknowledge)
  - ▶ Bit-stuff (6. bit)

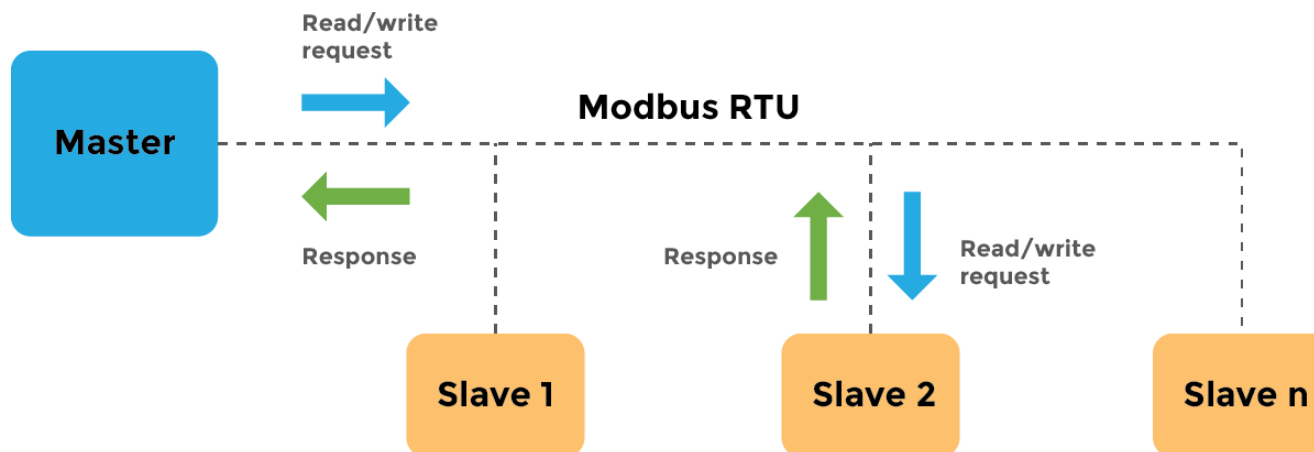
*Kako razrešiti ?*



# 3. MODBUS protokol

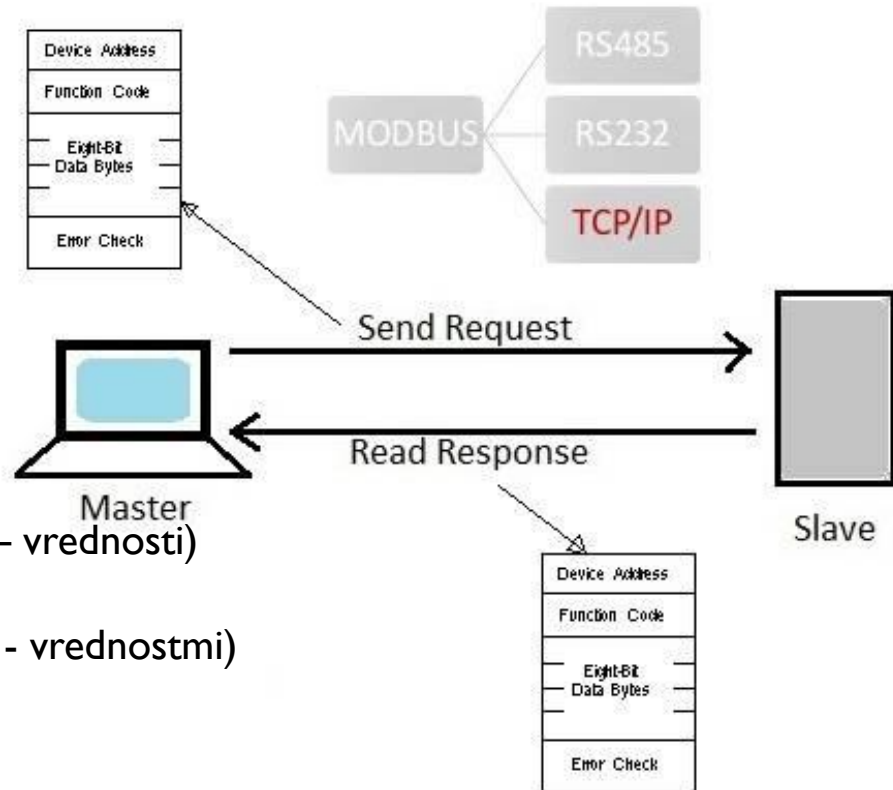
## MODBUS:

- ▶ Opis protokola in fizičnih nivojev
- ▶ Primeri Modbus komunikacij:
  - STM32 (MB master) <-> PC (MB slave)
  - Cybro kot MB Slave (višje nivojsko omrežje)



# MODBUS na kratko

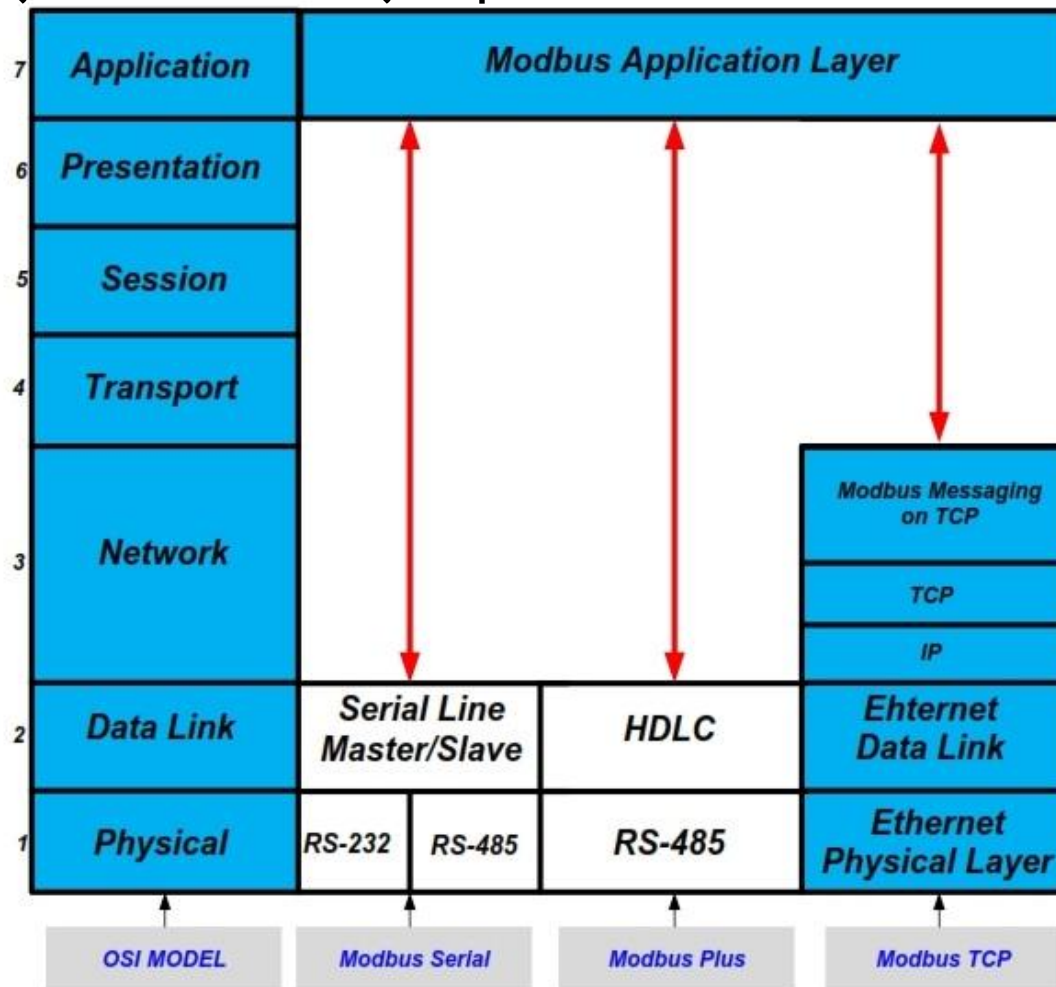
- ▶ Višje nivojski komunikacijski protokol:
  - ▶ Različni „fizični“ nivoji:
    - ▶ RS-232
    - ▶ RS-485
    - ▶ TCP
- ▶ Sporočilno orientiran :
  - ▶ Serijska komunikacija:
    - ▶ ASCII (v obliki ASCII znakov)
    - ▶ RTU (v obliki posameznih bajtov – vrednosti)
  - ▶ Ethernet komunikacija:
    - ▶ TCP (v obliki TCP paketov z bajti - vrednostmi)
  - ▶ dvosmerna komunikacija
    - ▶ zahteva (MB Master) →
    - ▶ ← odziv (MB Slave)
  - ▶ V/I prostor: digitalni V/I, analogni V/I
  - ▶ MB ukazi (Function Codes – FC)





# MODBUS na kratko

- ▶ Višje nivojski komunikacijski protokol:



# MODBUS na kratko

## ▶ Sporočilno orientiran :

### ▶ Serijska komunikacija:

#### ▶ ASCII (v obliki ASCII znakov)

START	ADDRESS	FUNCTION	DATA	LRC CHECK	END
1 CHAR :	2 CHARS	2 CHARS	<i>n</i> CHARS	2 CHARS	2 CHARS CRLF

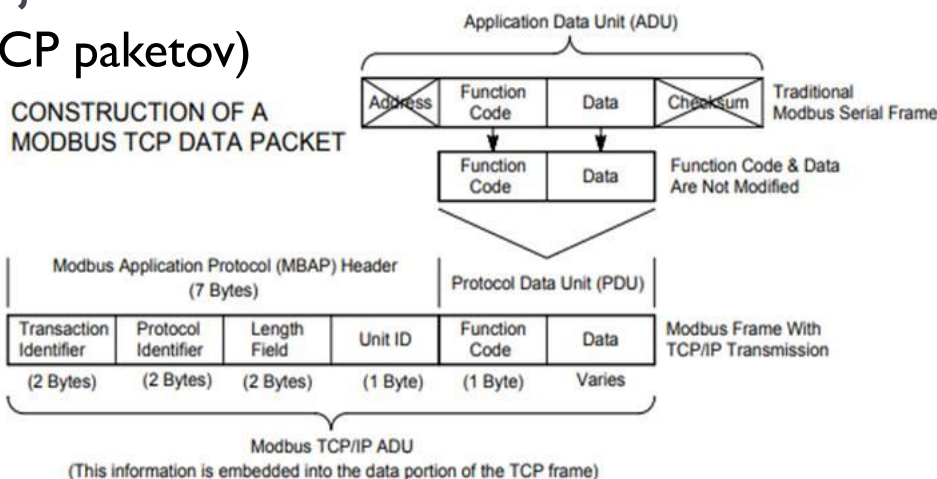
#### ▶ RTU (v obliki posameznih bajtov – vrednosti)

START	ADDRESS	FUNCTION	DATA	CRC CHECK	END
T1–T2–T3–T4*	8 BITS	8 BITS	<i>n</i> x 8 BITS	16 BITS	T1–T2–T3–T4*

\*For T1–T2–T3–T4, 3.5 character times at no communication.

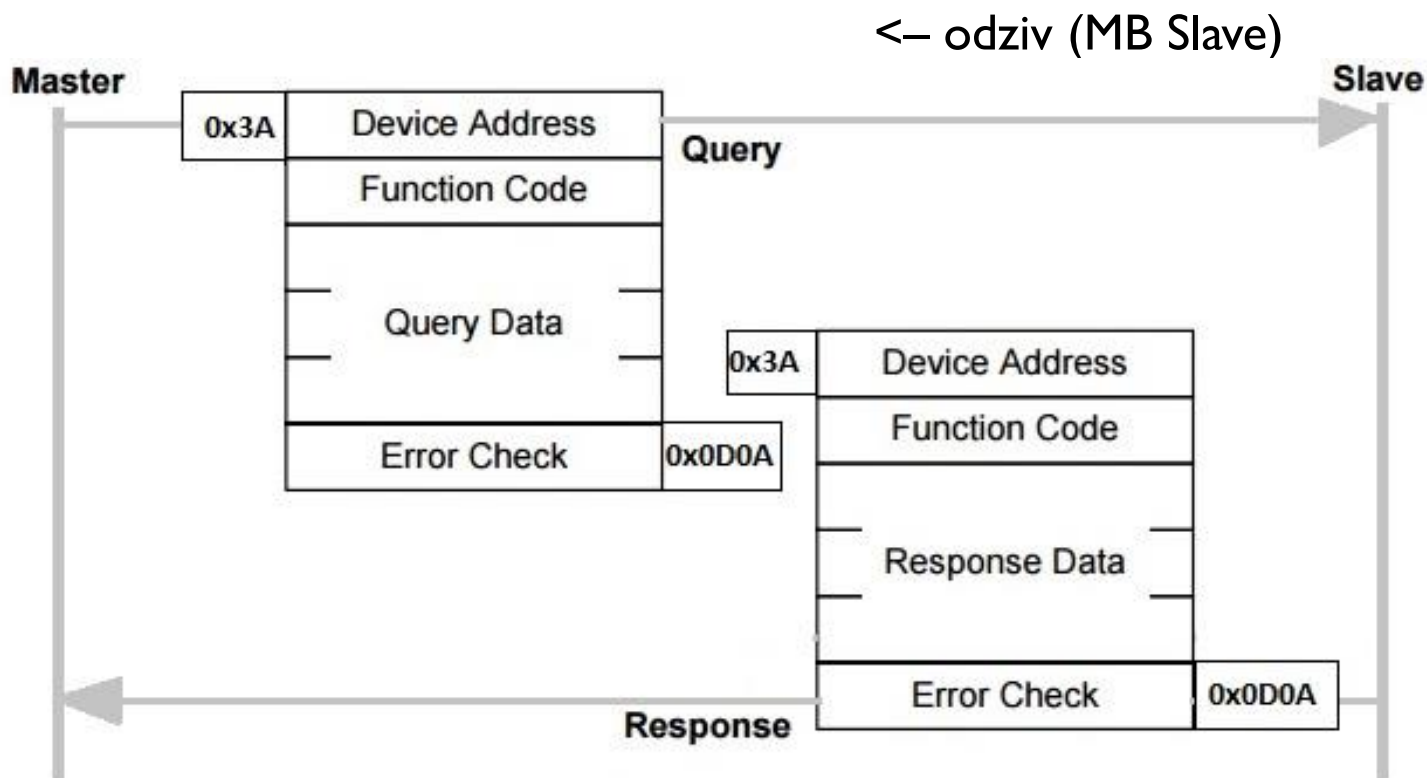
## ▶ Ethernet komunikacija:

### ▶ TCP (v obliki TCP paketov)



# MODBUS na kratko

- ▶ dvosmerna komunikacija
  - ▶ zahteva (MB Master) →



Modbus Message Frames

# MODBUS na kratko

- ▶ V/I prostor: digitalni V/I, analogni V/I

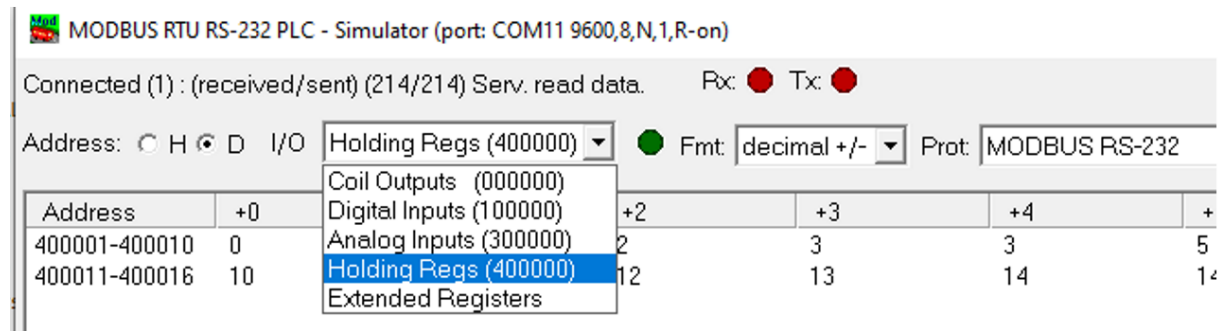
Types of Modbus Data	Data format	Address
Modbus Coils	Bits, Binary values	00001
Digital Inputs	Binary values	10001
Analog Inputs	Analog values	30001
Holding registers	Analog values	40001

- ▶ MB ukazi (Function Codes – FC)

Function Code	Action	Table Name	Function Code	Action	Table Name
01 (01 hex)	Read	Discrete Output Coils	04 (04 hex)	Read	Analog Input Registers
05 (05 hex)	Write single	Discrete Output Coil	03 (03 hex)	Read	Analog Output Holding Registers
15 (0F hex)	Write multiple	Discrete Output Coils	06 (06 hex)	Write single	Analog Output Holding Register
02 (02 hex)	Read	Discrete Input Contacts	16 (10 hex)	Write multiple	Analog Output Holding Registers

# MODBUS – Primer: STM32 <-> PC (USB VCOMPort)

## ▶ PC – MB slave simulator:



## ▶ STM32 – MB master SW :

```
// --- Slave ID=1, Write state 511 of 1 analog output register on addresses 0-15
usModbusUserData[0]= usModbusUserData[1]= usModbusUserData[2]= usModbusUserData[3]=MBCnt;
errorCode = eMBMasterReqWriteMultipleHoldingRegister(1,MBCnt,2,usModbusUserData,1);
MBCnt= (MBCnt + 1) % 16;
```

[https://github.com/LAPSyLAB/STM32F4\\_Discovery\\_VIN\\_Projects](https://github.com/LAPSyLAB/STM32F4_Discovery_VIN_Projects)

# MODBUS – Primer: Cybro kot MB slave

Cybro supports:

- Modbus RTU slave (RS232/RS485)
- Modbus TCP slave (Ethernet)

Function codes:

code	hex	command
1	01h	READ_COILS
3	03h	READ_HOLDING_REGISTERS
5	05h	WRITE_SINGLE_COIL
6	06h	WRITE_SINGLE_REGISTER
15	0Fh	WRITE_MULTIPLE_COILS
16	10h	WRITE_MULTIPLE_REGISTERS

Modbus data model describes how modbus coils and registers are translated to Cybro memory.

