Vhodno izhodne naprave

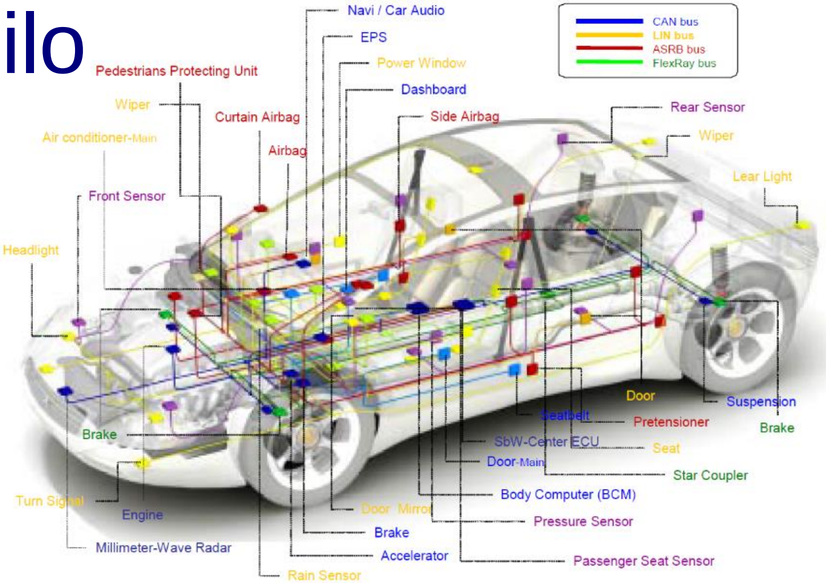
Laboratorijska vaja 13 -   LV 5

CANBUS

Laboratorijska vaja 13 -   LV5

- 13.0: CANBUS osvežitev
- 13.1 Opis primera : Cybrotech CANBUS sistem
- 13.2: Krmiljenje Cybrotech IEX-2 modulov
- 13.3: STM32F4 – osnovni IEX-2 modul
- 13.4: CANBUS meritve

2. CANBUS vodilo



CANBUS (ISO-11898-2):

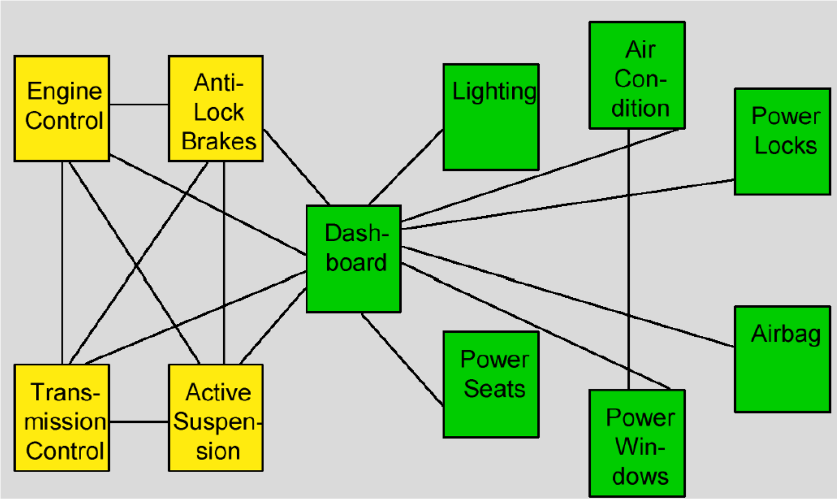
- □ Zgodovina
- □ Področja uporabe
  - ■ Avtomobilska industrija
  - ■ Industrijska avtomatika, pametne stavbe
- □ Pregled protokola, arbitraže, fizičnega nivoja
- □ Praktični primer: Pametna hiša, IEX-2 protokol

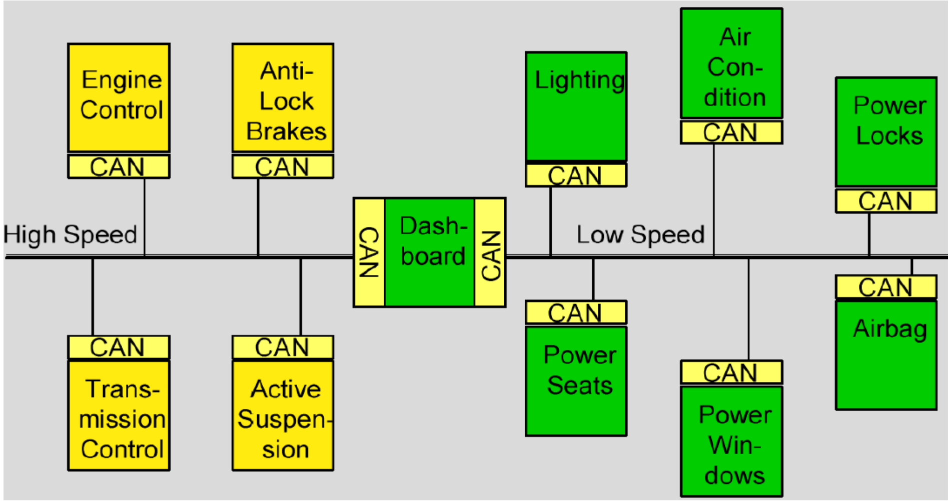
Lab. Vaja :

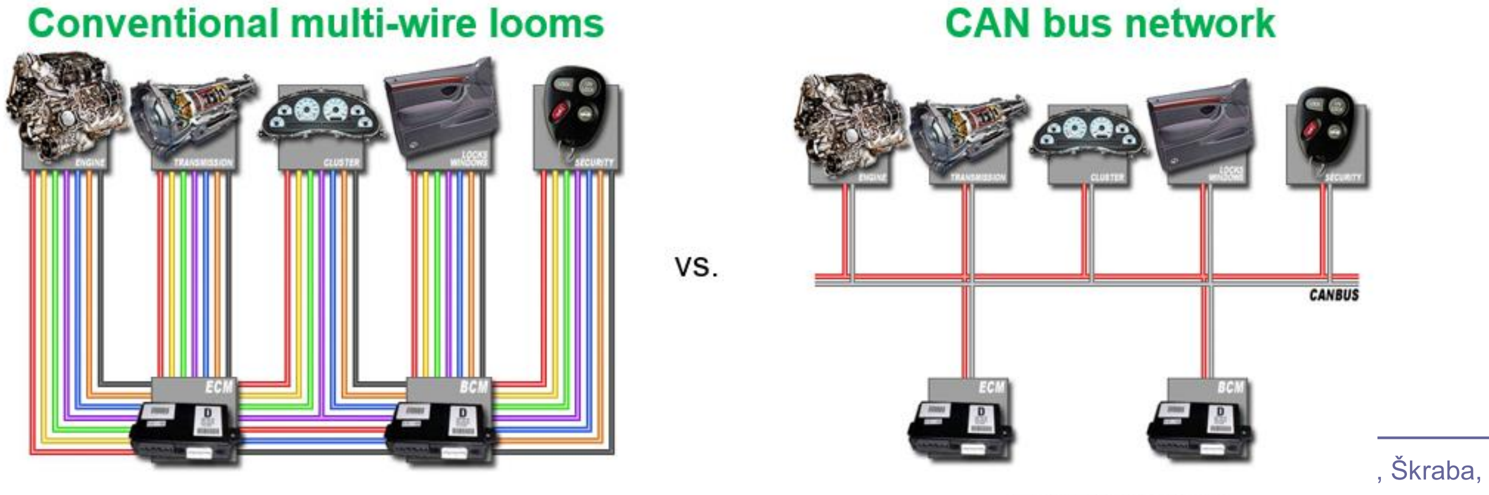
- □ Gradniki in shema testnega sistema
- □ Programiranje sistema
- □ Meritve signalov na povezavah

Zakaj vodilo ?

Primera povezav brez (levo) in z (desno) CANbus vodilom



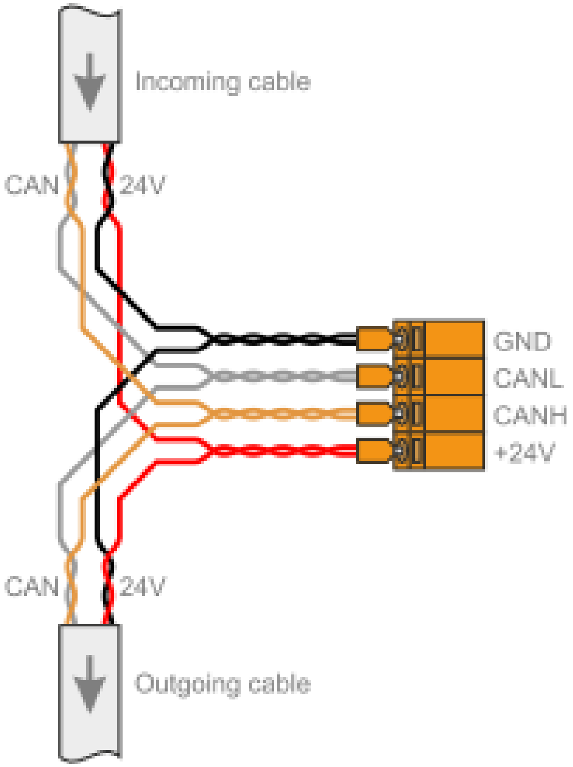




http://canbuskit.com/what.php

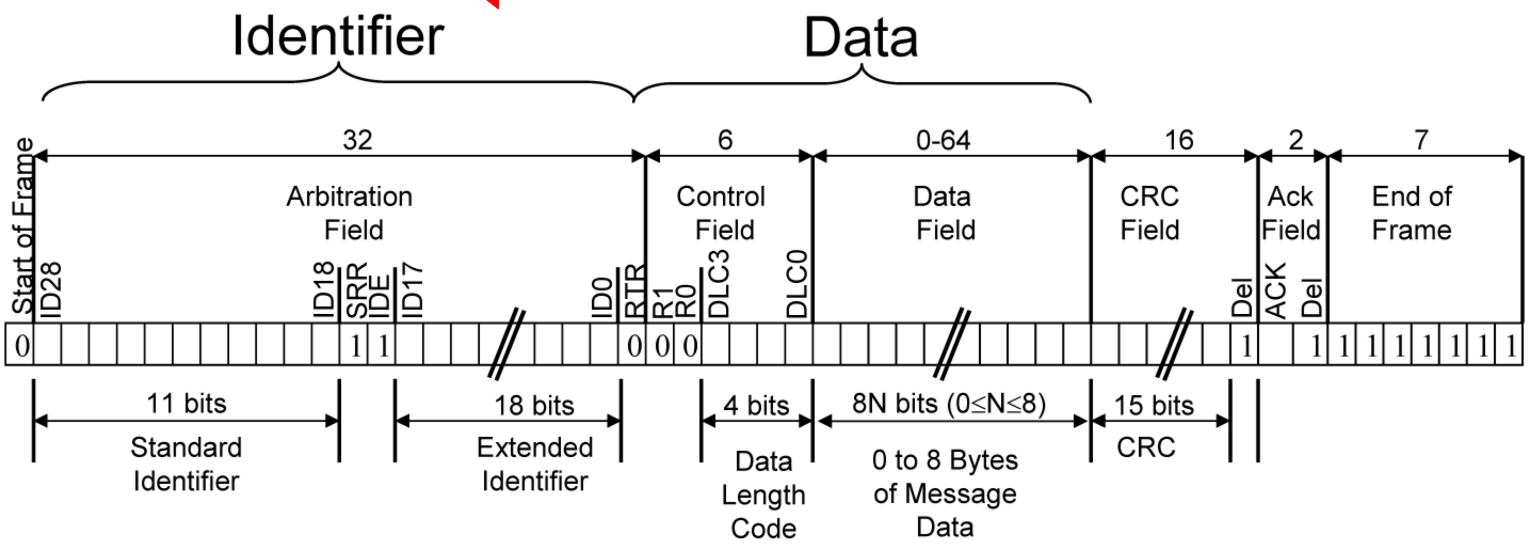
CANbus na kratko

- CANbus – **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 1Mbit/s, 40m,
  - □ sporočila do 8 bajtov (latenca)
- CAN-FD standard, ISO 11898-2:2016
  - □ 2Mbps, 5Mbps



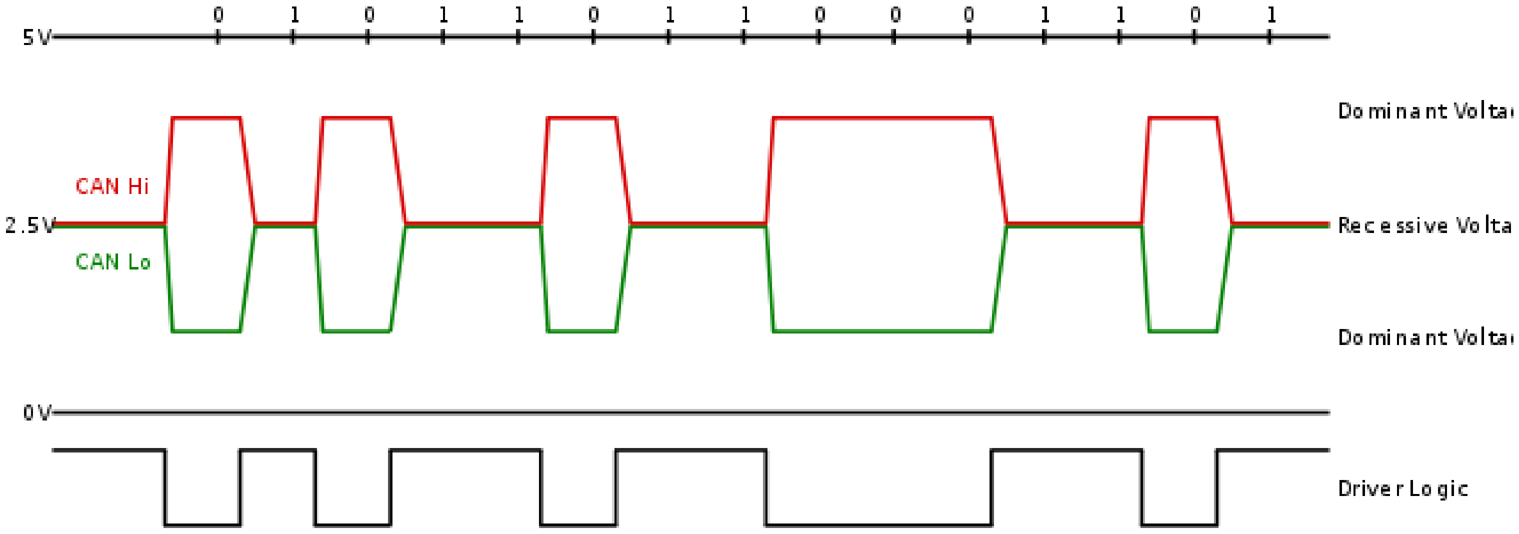
CANbus na kratko

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

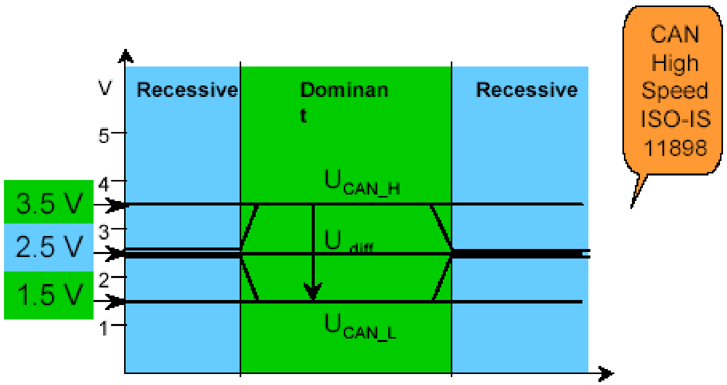


CANbus napetostni nivoji ISO-IS 11898

- 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 napetostni nivoji ISO-IS 11898

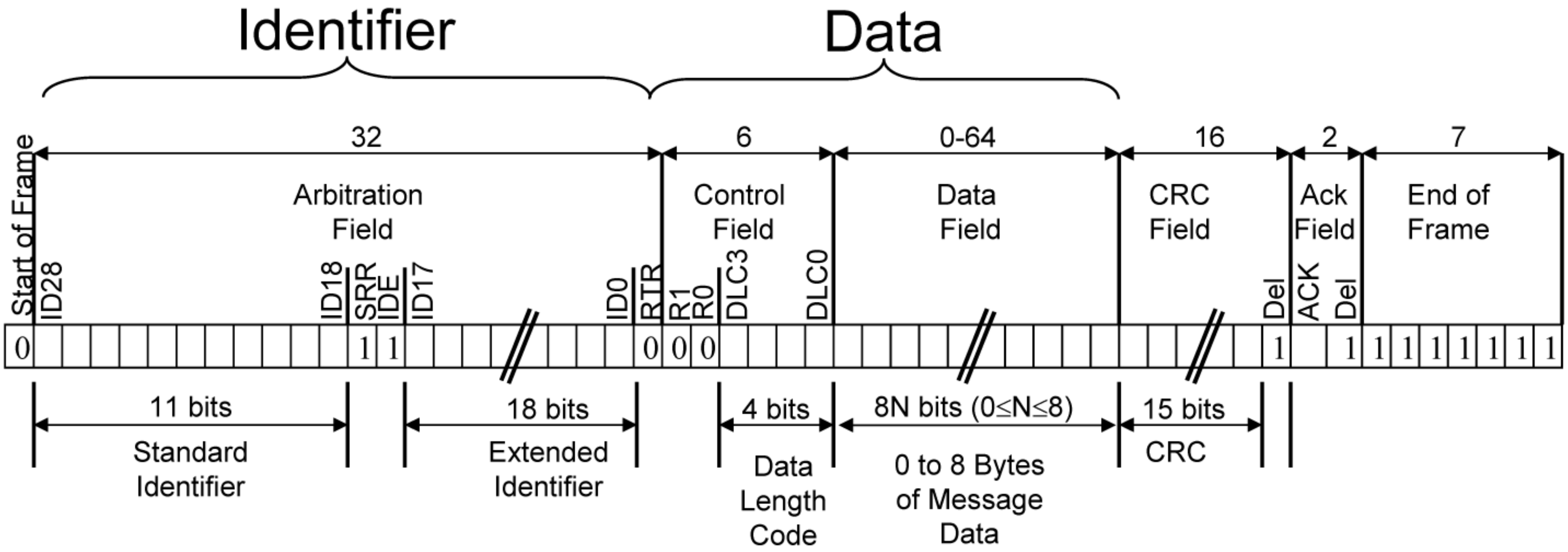


- Recesivni bit „1“:
  - obe liniji na približno 2.5V
  - diferencialna napetost CAN_H in CAN_L ≈ 0 V

- Dominantni bit „0“:
  - CAN_H na pribl. 3.5 V in CAN_L pribl. 1.5 V
  - diferencialna napetost CAN_H in CAN_L ≈ 2 V

Format sporočila

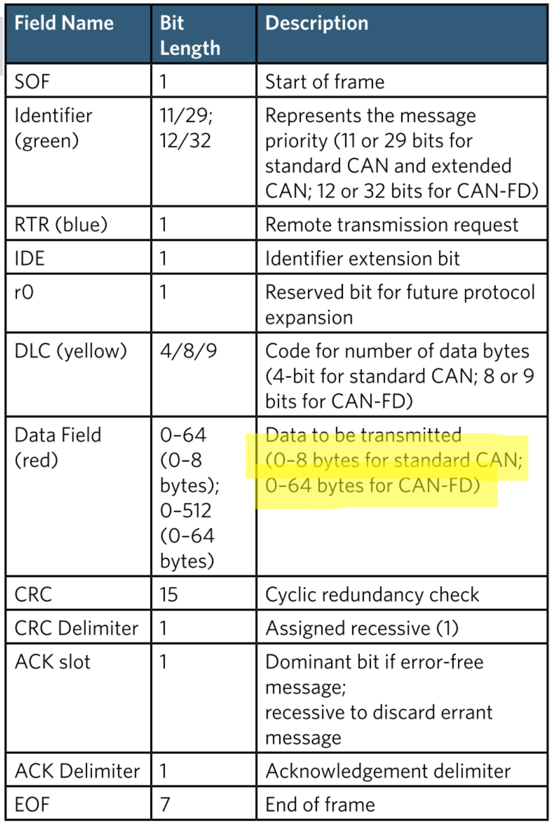
- Vsako sporočilo ima ID, podatke in dodatke
- ID - 11 ali 29 bitov
- Data - do 8 bajtov
- Dodatki - start (SOF), CRC, ACK, end (EOF)



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

- Recesivni bit „1“:
  - obe liniji na približno 2.5V
- Dominantni bit „0“:
  - CAN_H na pribl. 3.5 V in CAN_L pribl. 1.5 V

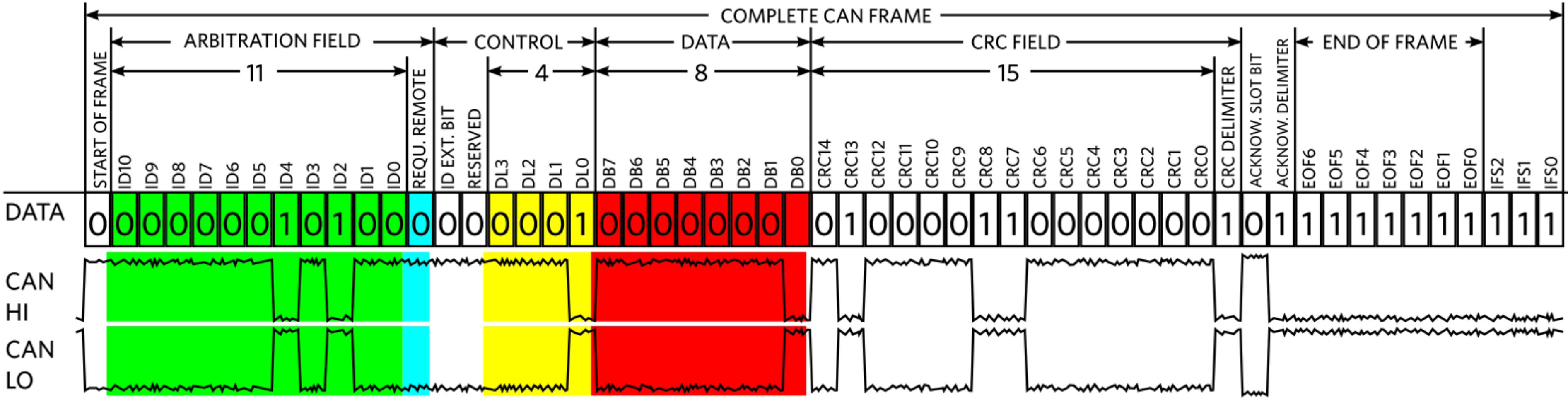
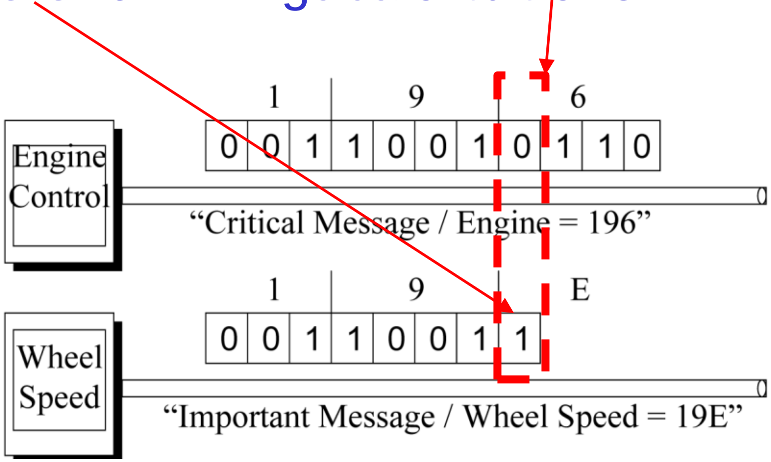


Figure 4. CAN Message Data-Frame Format

Arbitraža (Non-Destructive Arbitration)

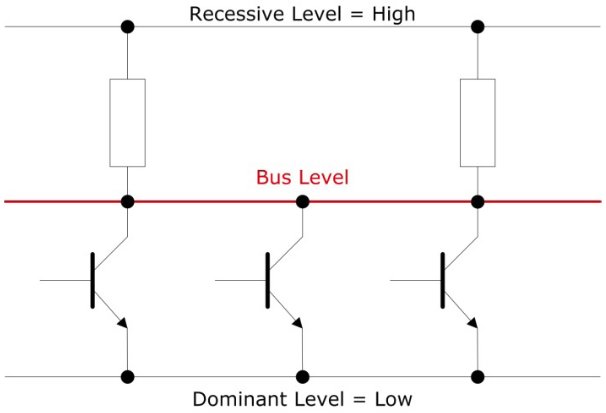
- Pomembnos sporočila je določena z IDjem

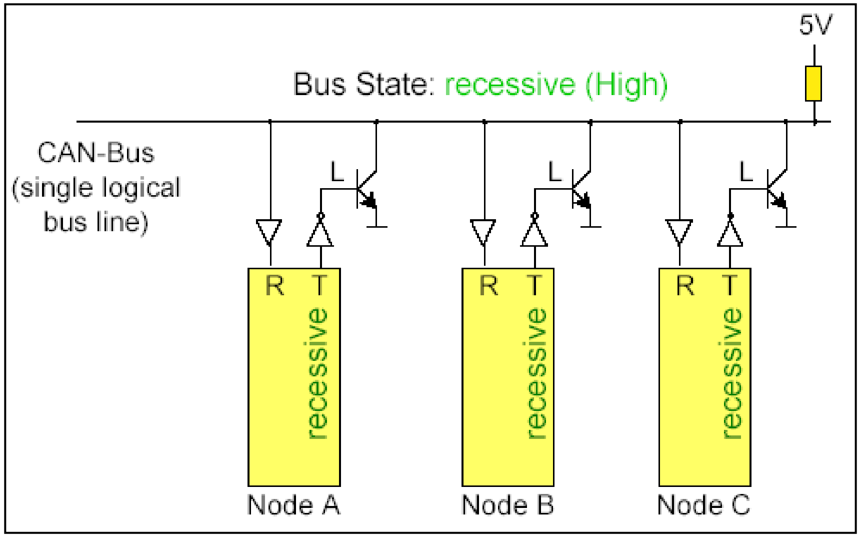
  Nižja vrednost = Višja pomembnost
- Naprava odda in hkrati bere
  - □ "0" na vodilu prevlada "1" na vodilu
- Naprava:
  - □ odda in bere enako –> nadaljuje z oddajo
  - □ odda "1" in bere "0" –> izguba artbitraže

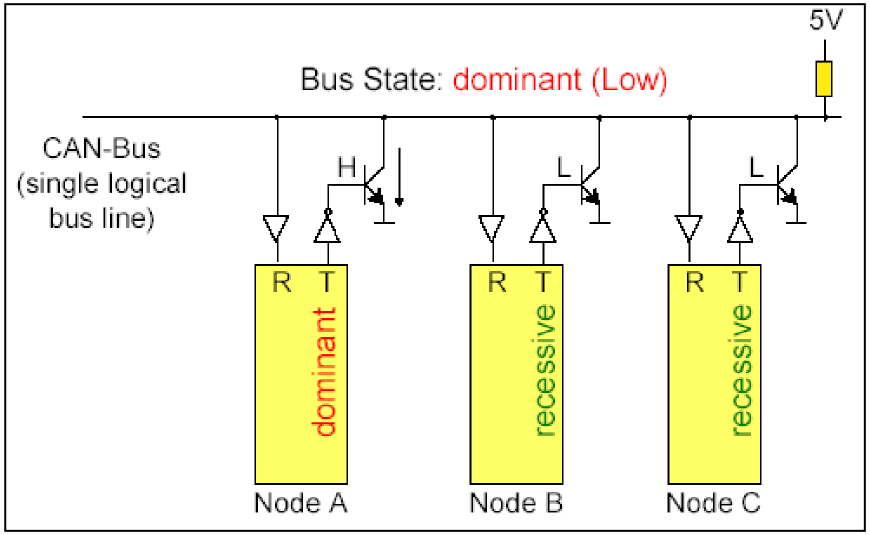

1
9
6
0 0 1 1 0 0 1 0 1 1 0
Engine Control
"Critical Message / Engine = 196"
1
9
E
0 0 1 1 0 0 1 1
Wheel Speed
"Important Message / Wheel Speed = 19E"

Wired AND (Arbitraža)

Stanje “0” (nizka napetost oz. dominantno stanje) na vodilu prevlada ostala stanja “1” (višja napetost oz. recesivno stanje) na vodilu.

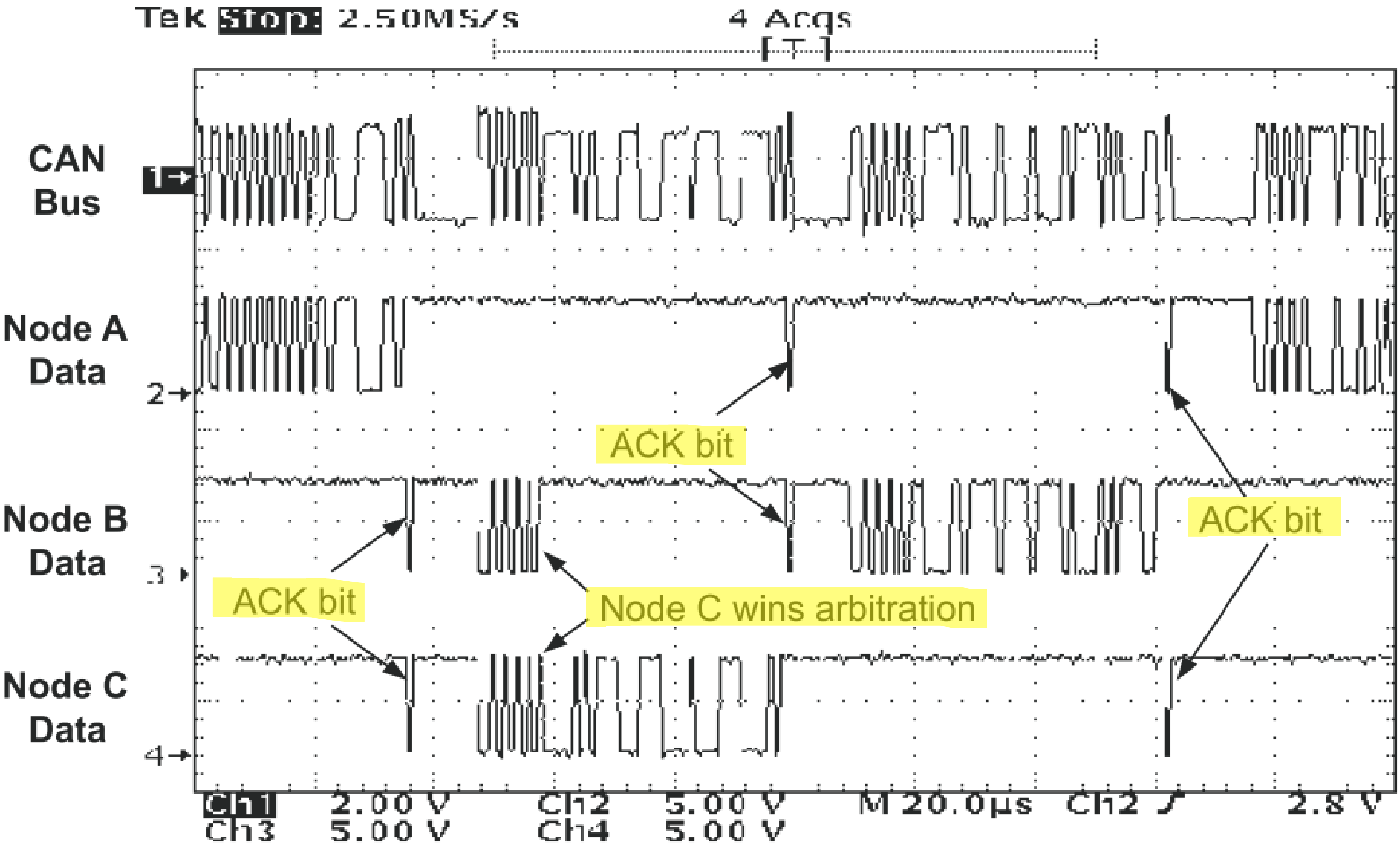






Osciloskop: primer CANbus komunikacije

www.ti.com *The CAN Bus*



ACK bit

ACK bit

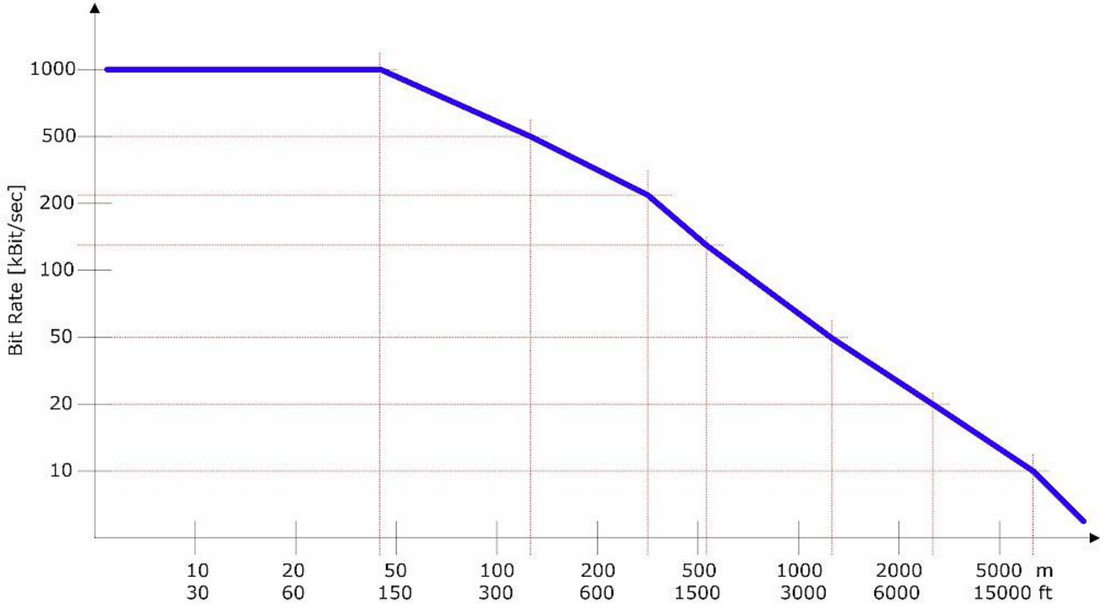
ACK bit

Node C wins arbitration

Figure 8. CAN Bus Traffic

Hitrost komunikacije

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



RS-485 vs CANBUS

Podobno/enako:

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

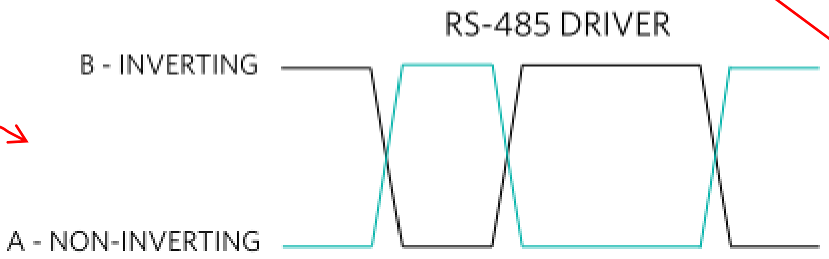
Kako razrešiti ?

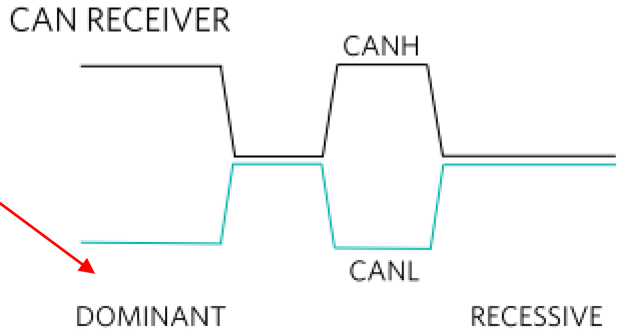
Prednosti CANBUS :

- 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)

Prednosti RS485 :

- Višja hitrost – do 35Mbit/s
  - Obe stanji sta aktivno vodeni
  - CANBUS (Wired AND) ima recesivno in dominantno stanje





Laboratorijska vaja 13 - LV5

- 13.0: CANBUS osvežitev
- 13.1 Opis primera : Cybrotech CANBUS sistem
- 13.2: Krmiljenje Cybrotech IEX-2 modulov
- 13.3: STM32F4 – osnovni IEX-2 modul
- 13.4: CANBUS meritve

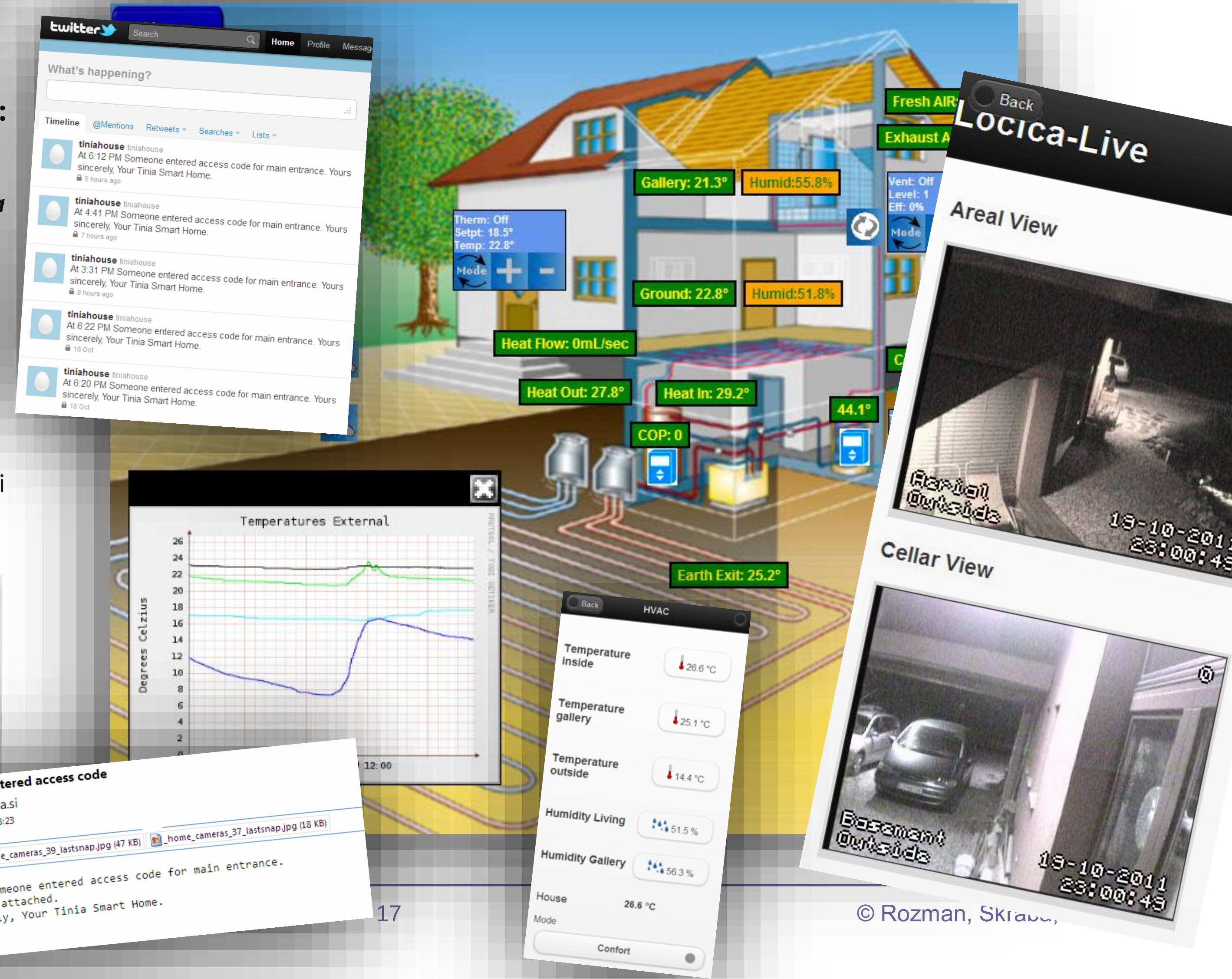
Tinia – prijazen dom
TBS – „Tinia Building Server”

Kratek opis

TBS – „Tinia 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



tiniahouse tiniahouse
At 6:12 PM Someone entered access code for main entrance. Yours sincerely, Your Tinia Smart Home.
5 hours ago

tiniahouse tiniahouse
At 4:41 PM Someone entered access code for main entrance. Yours sincerely, Your Tinia Smart Home.
7 hours ago

tiniahouse tiniahouse
At 3:31 PM Someone entered access code for main entrance. Yours sincerely, Your Tinia Smart Home.
8 hours ago

tiniahouse tiniahouse
At 6:22 PM Someone entered access code for main entrance. Yours sincerely, Your Tinia Smart Home.
18 Oct

tiniahouse tiniahouse
At 6:20 PM Someone entered access code for main entrance. Yours sincerely, Your Tinia Smart Home.
18 Oct

HVAC

Temperature inside	26.6 °C
Temperature gallery	25.1 °C
Temperature outside	14.4 °C
Humidity Living	51.5 %
Humidity Gallery	56.3 %
House	26.6 °C

Mode
Confort

Tinia: Someone entered access code
tinia_engine1@locica.si
Poslano: tor 18.10.2011 18:23
Za:
Sporočilo | _home_cameras_39_lastsnap.jpg (47 KB) | _home_cameras_37_lastsnap.jpg (18 KB)

```
At 6:22 PM, someone entered access code for main entrance.
Snapshots are attached.
Yours sincerely, Your Tinia Smart Home.
```

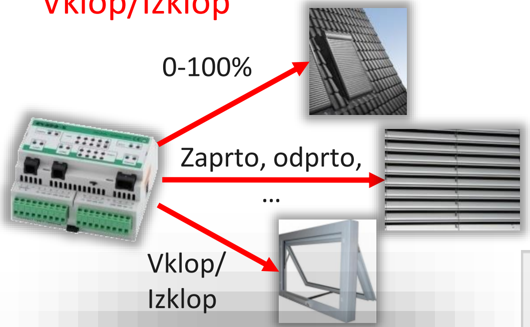
Pasivno ogrevanje/hlajenje…

Rolete, Žaluzije, Okna

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

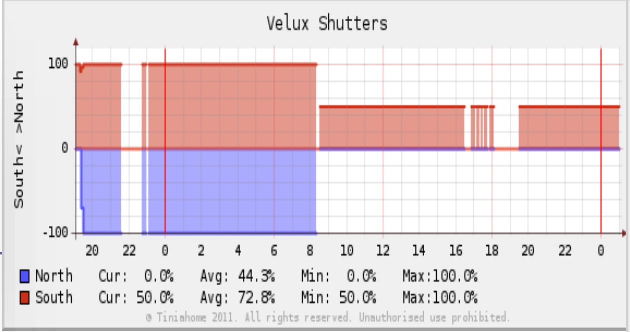
•**Žaluzije** imajo stanja :
Zaprto(100%), Senčeno(75%), Odprto(50%), Solarno pasivno (25%), Dvignjeno(0%).

•**Motorizirana** okna:
Vklop/Izklop



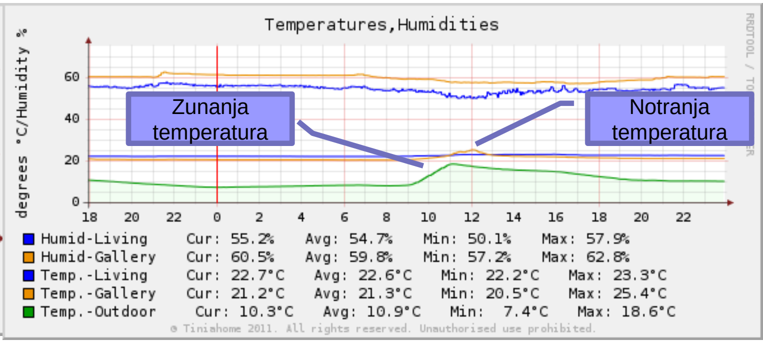
- Strešna okna z roletami :
  - severna, običajno:
    - **Odprta v toplem vremenu** za boljšo osvetlitev (poletje)
    - **Zaprta v hladnem vremenu** za ohranjanje toplote (zima)
  - južna, običajno:
    - **Odprta v hladnem, sončnem vremenu** za pasivno ogrevanje (zima, pomlad)
    - **Zaprta v vročem 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:



```
■ North   Cur:  0.0%   Avg: 44.3%   Min:  0.0%   Max:100.0%
■ South   Cur: 50.0%   Avg: 72.8%   Min: 50.0%   Max:100.0%
```

© Tiniahome 2011. All rights reserved. Unauthorised use prohibited.



```
■ Humid-Living    Cur: 55.2%     Avg: 54.7%     Min: 50.1%     Max: 57.9%
■ Humid-Gallery   Cur: 60.5%     Avg: 59.8%     Min: 57.2%     Max: 62.8%
■ Temp.-Living    Cur: 22.7°C    Avg: 22.6°C    Min: 22.2°C    Max: 23.3°C
■ Temp.-Gallery   Cur: 21.2°C    Avg: 21.3°C    Min: 20.5°C    Max: 25.4°C
■ Temp.-Outdoor    Cur: 10.3°C    Avg: 10.9°C    Min:  7.4°C    Max: 18.6°C
```

© Tiniahome 2011. All rights reserved. Unauthorised use prohibited.

CANbus v praksi

Sistem vgrajen v zgradbo FRI !

INTEGRA BM SYSTEM

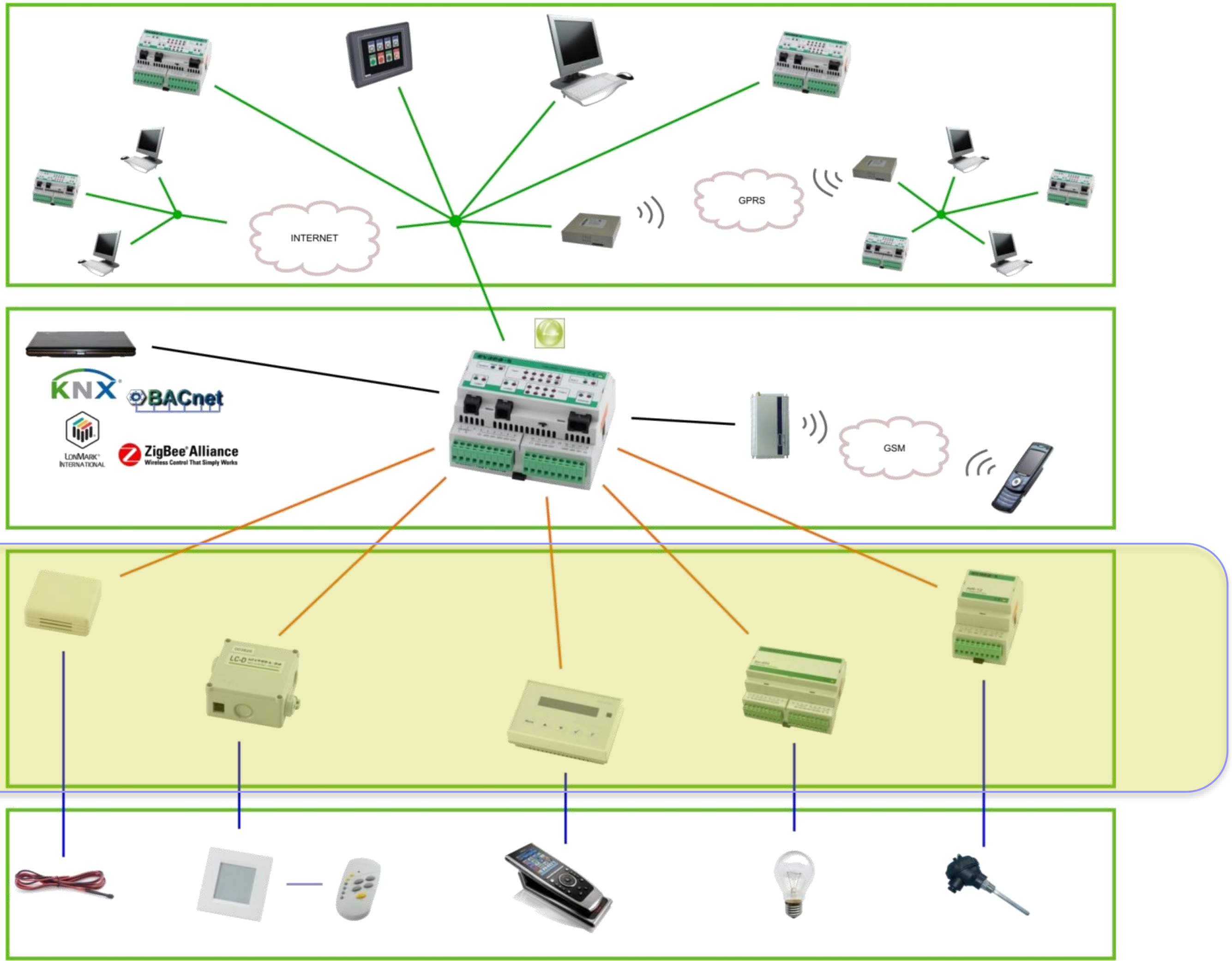
Industrial & Building Automation

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

CyBro controller

Low level network (Canbus)

Dodatki (tipala, daljinci, …)



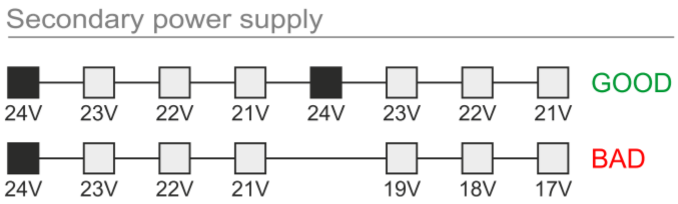
Bus length

Dolžina, hitrost in topologije

Regarding bus length, two points must be considered:

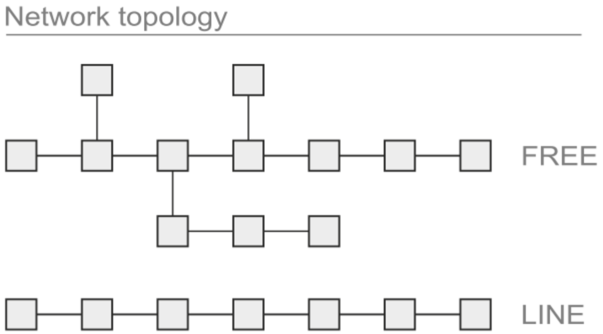
1. Voltage drop

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.

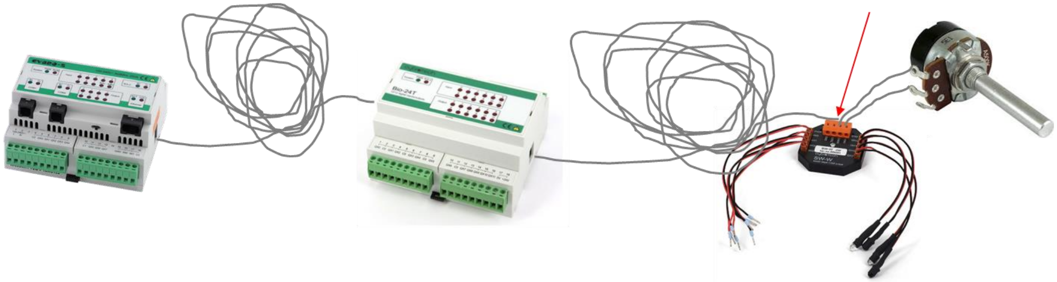


2. Signal delay

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.



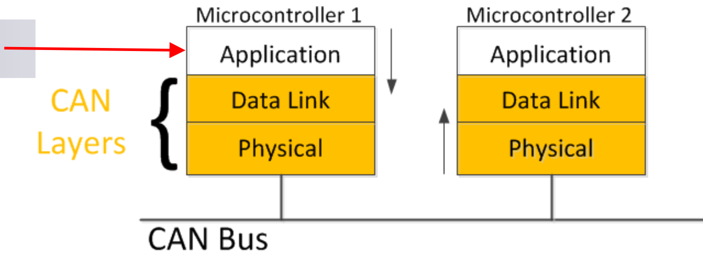
Speed\Topology	FREE	LINE
100kbps	100m	300m
50kbps	200m	500m
20kbps	500m	1000m



Primer protokola 3. nivo

INTEGRA BM SYSTEM

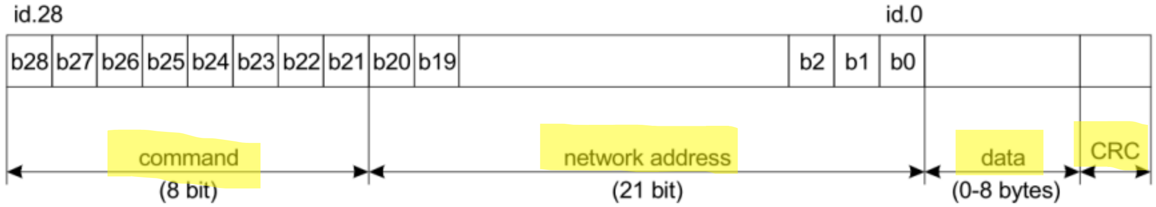
IEX protocol
(nadgradnja CANBUS)



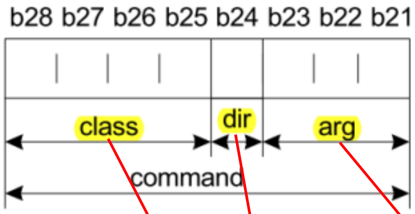
IEX PROTOCOL v2.8 POVZETEK

General

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



Command summary



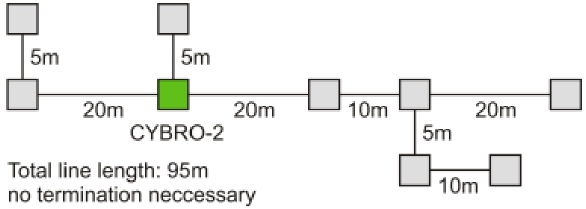
NAD – unikatni naslov IEX modula

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

INTEGRA BM SYSTEM

Cabling topology & Termination

1) Total IEX-2 bus length <100m



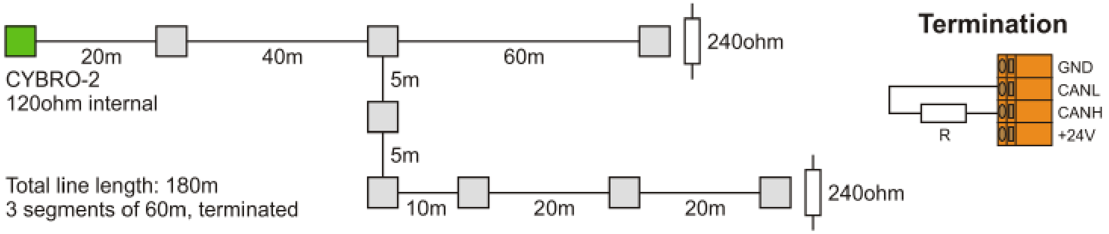
Total line length: 95m
no termination neccessary

🟩 Controller

□ Expansion module

— IEX-2 cable

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

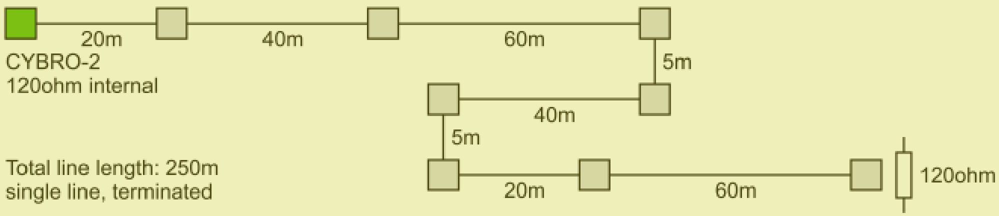


CYBRO-2
120ohm internal

Total line length: 180m
3 segments of 60m, terminated

$(120\text{ohm}^{-1}+240\text{ohm}^{-1}+240\text{ohm}^{-1})^{-1}=60\text{ohm}$

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

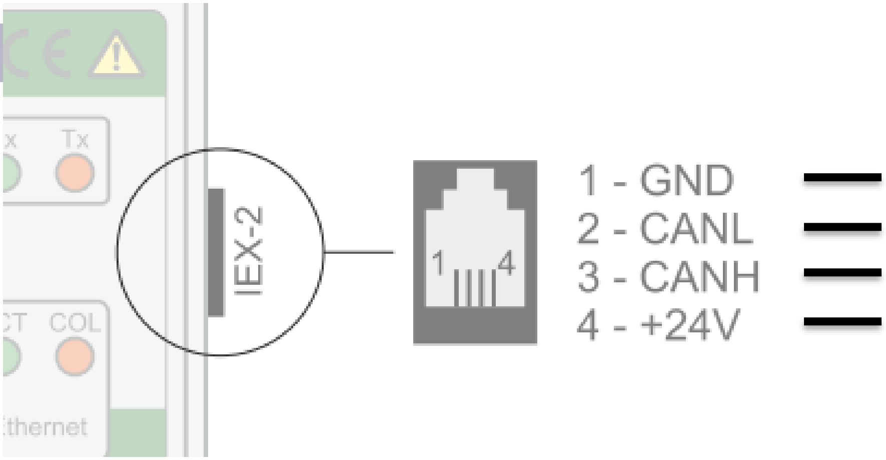


CYBRO-2
120ohm internal

Total line length: 250m
single line, terminated

CENTRALNI KRMILNIK CYBRO-2

Controller



Ethernet port

2 x RS-232 port

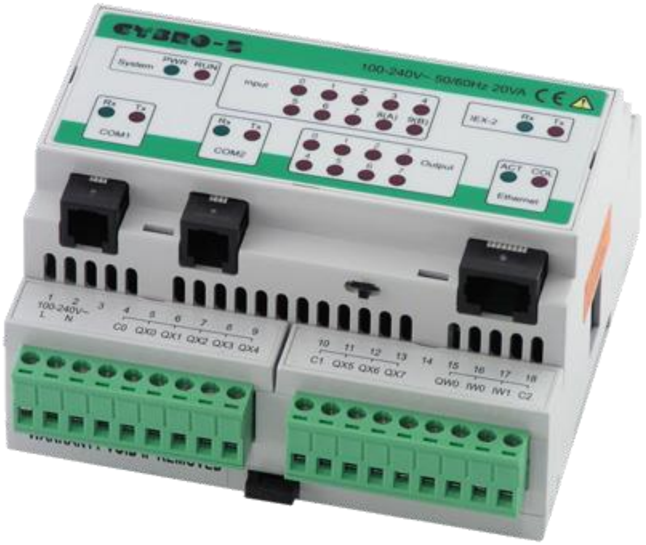
CAN interface

Digital and analog I/O

Core

Modular

Block



Communication and
status LED signalization

Retentive and
permanent EEPROM memory

Removable connectors

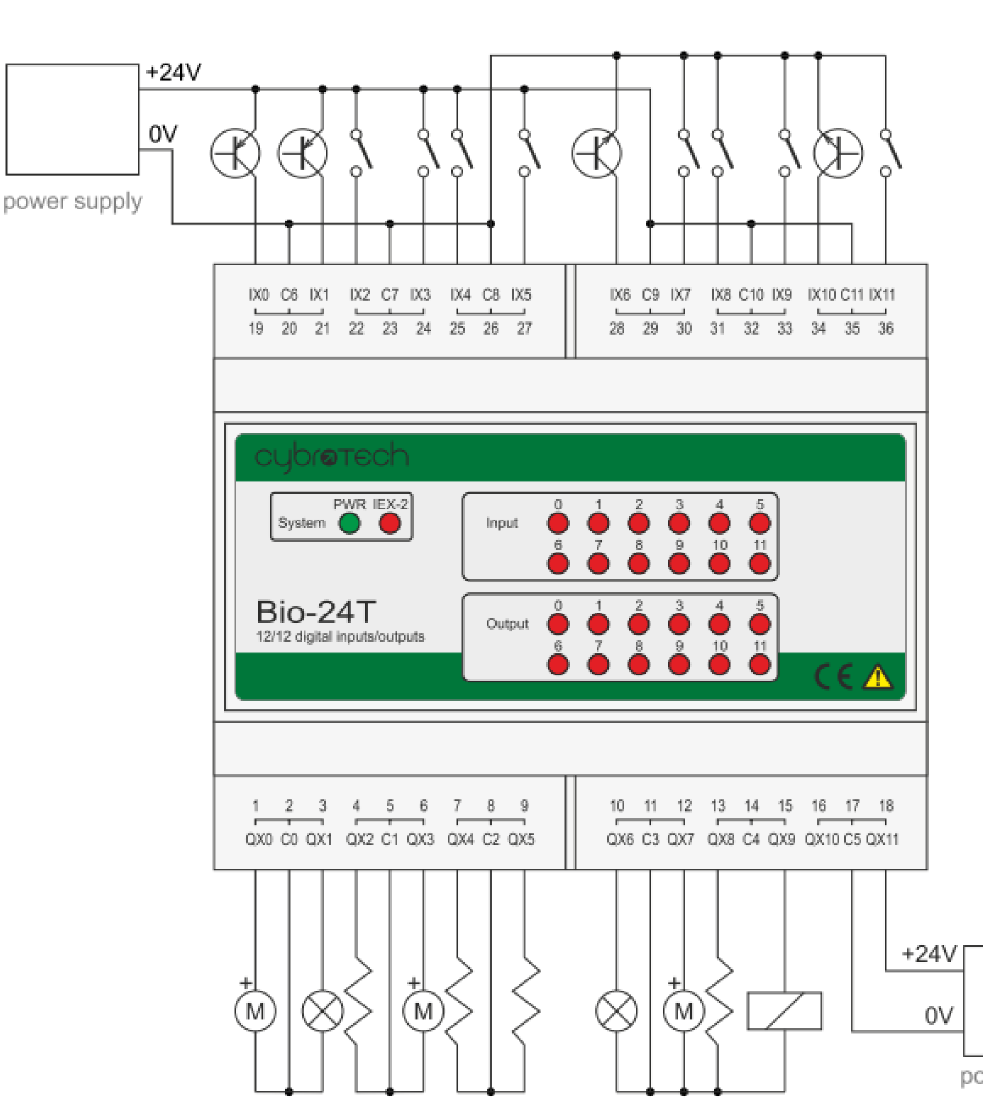
230V AC or 24V DC

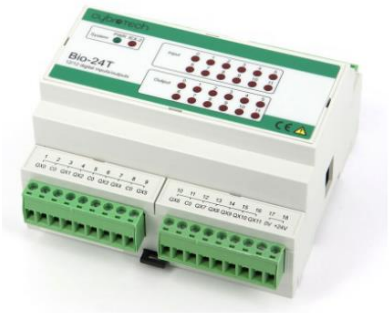
Primeri modulov

IEX MODULE Bio-24T

Bio-24T

Wiring diagram





IEX-2 module

12 opto-isolated PNP transistor outputs 1A

12 opto-coupler inputs 24V

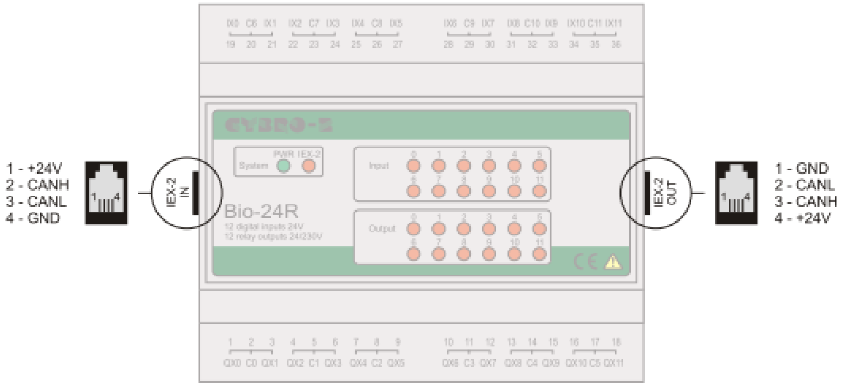


Figure 3: IEX-2 input and output ports.

Primeri modulov

IEX MODULE FC

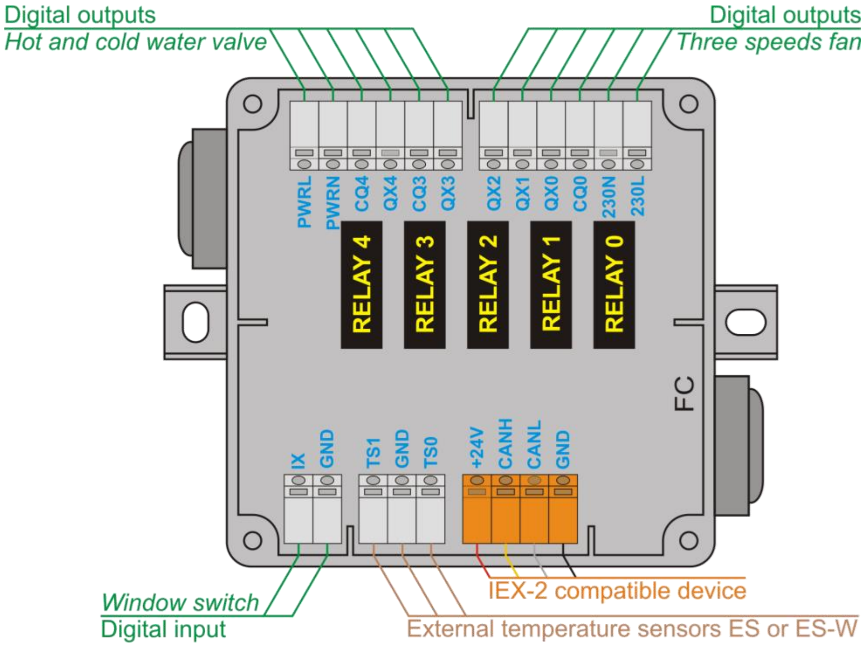
FC

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


Digital outputs
Hot and cold water valve

Digital outputs
Three speeds fan

Window switch
Digital input

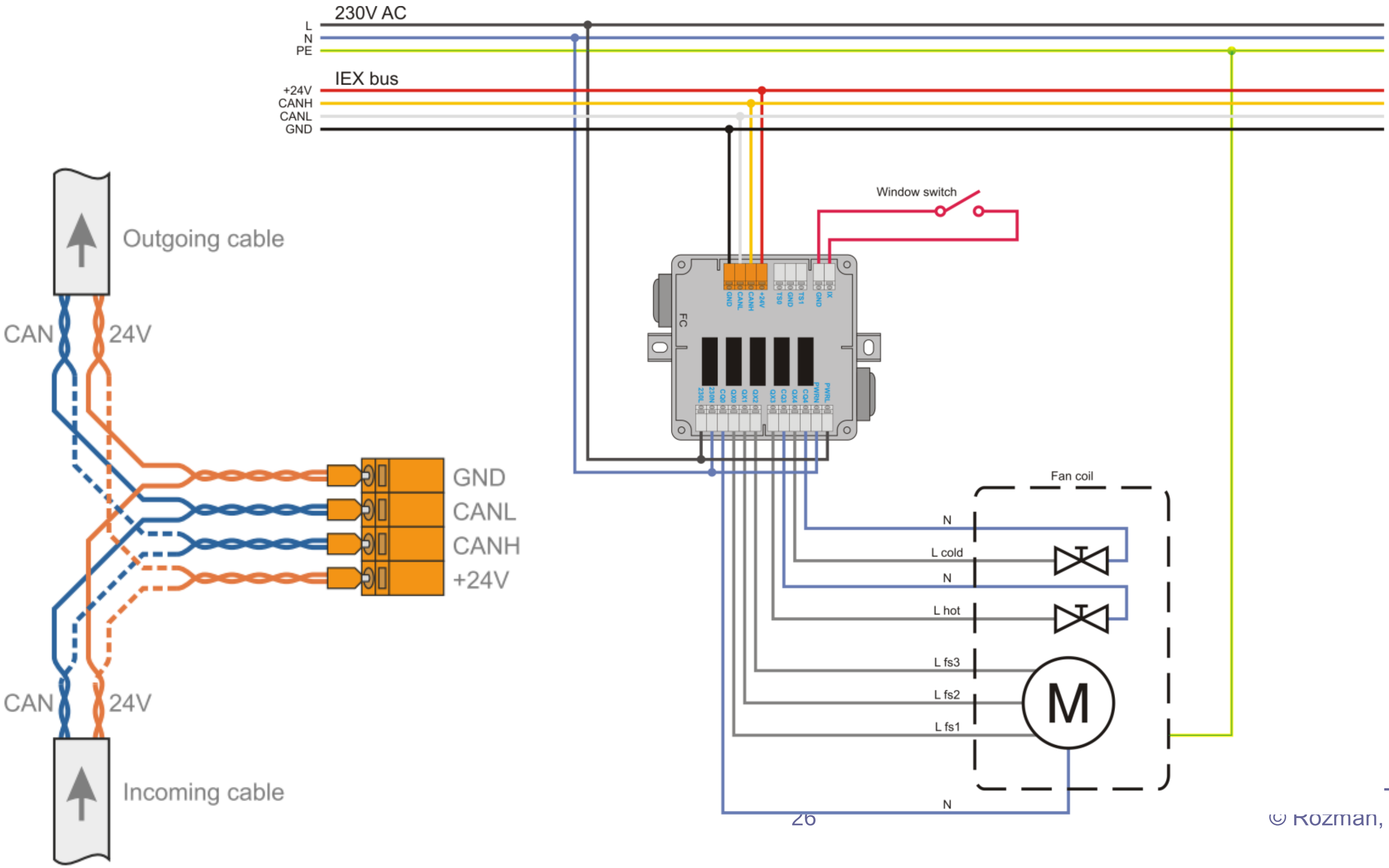
IEX-2 compatible device

External temperature sensors ES or ES-W

IEX MODULE FC

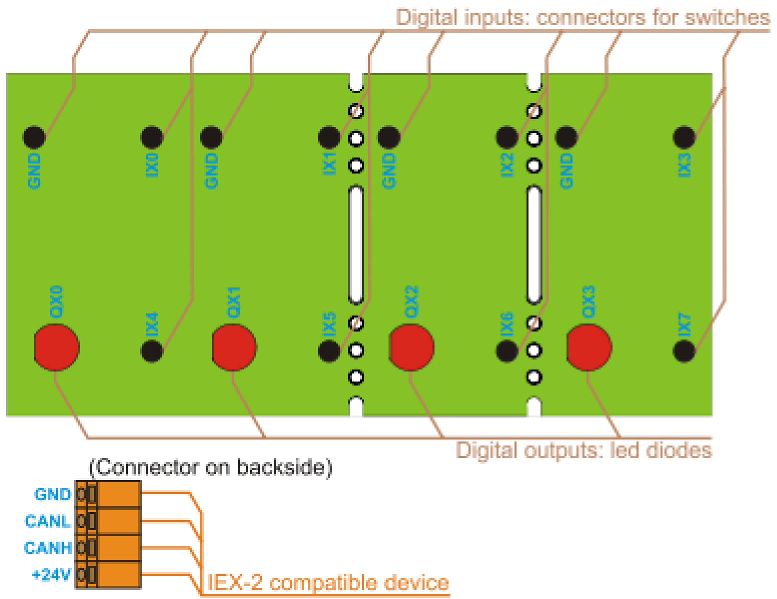
CONNECTING FAN COIL AND WINDOW SWITCH TO FC MODULE

FC



Primeri modulov

IEX MODULE SW-L



SW-L

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

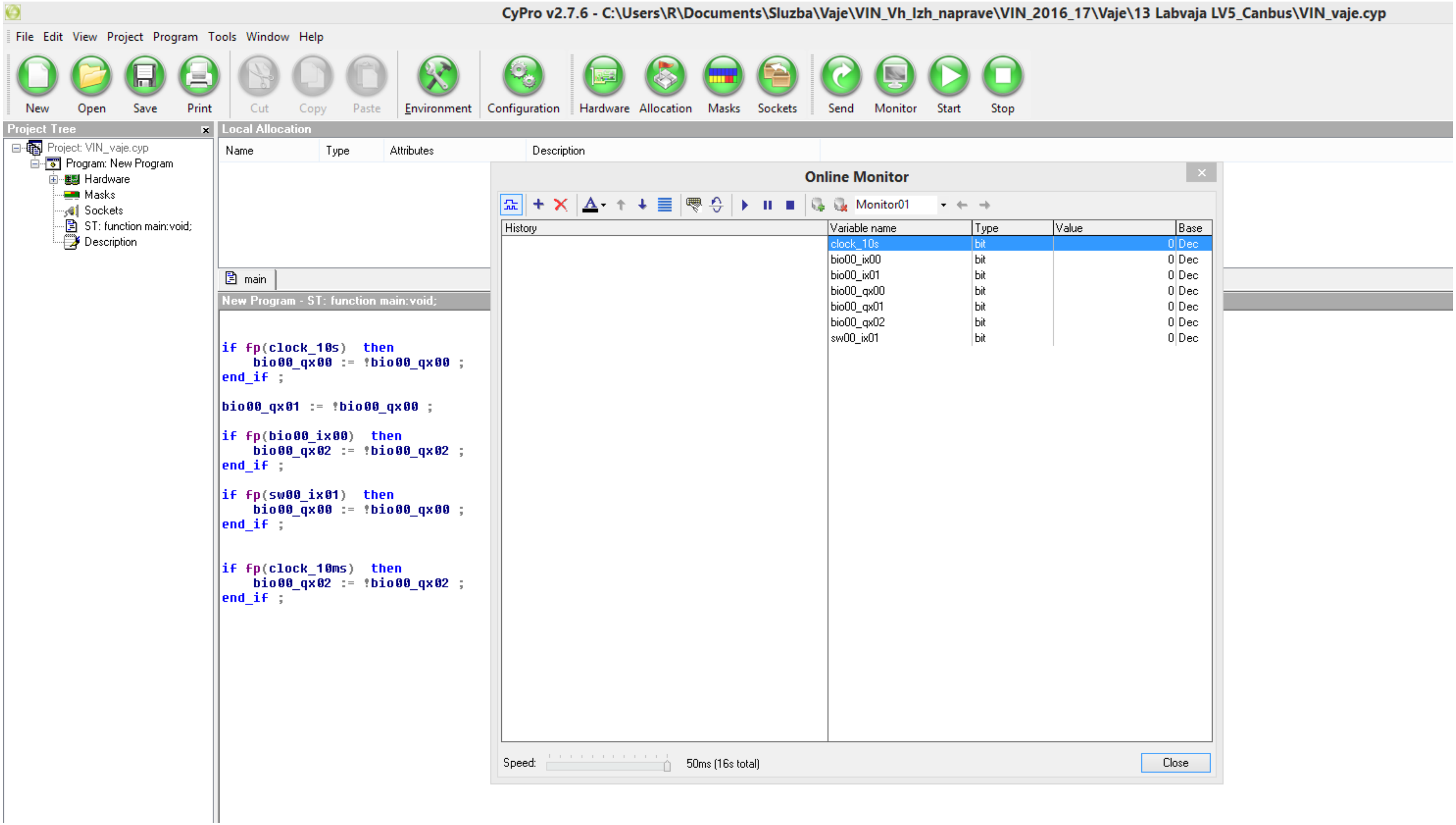
Technical specifications

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 – SOFTWARE IDE

Primer IDE

CyPro



```
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 ;
```

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

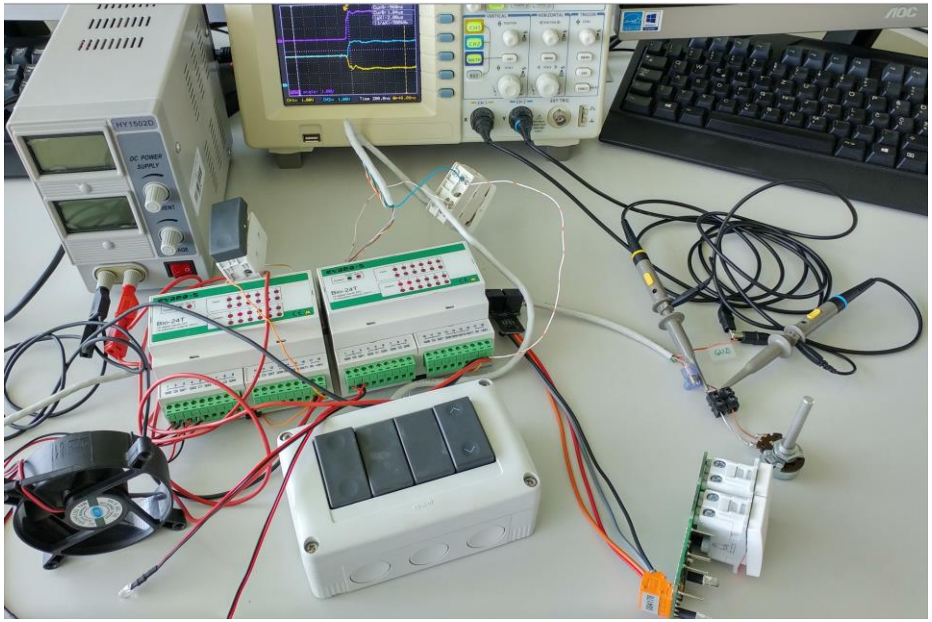
Laboratorijska vaja 13 - LV5

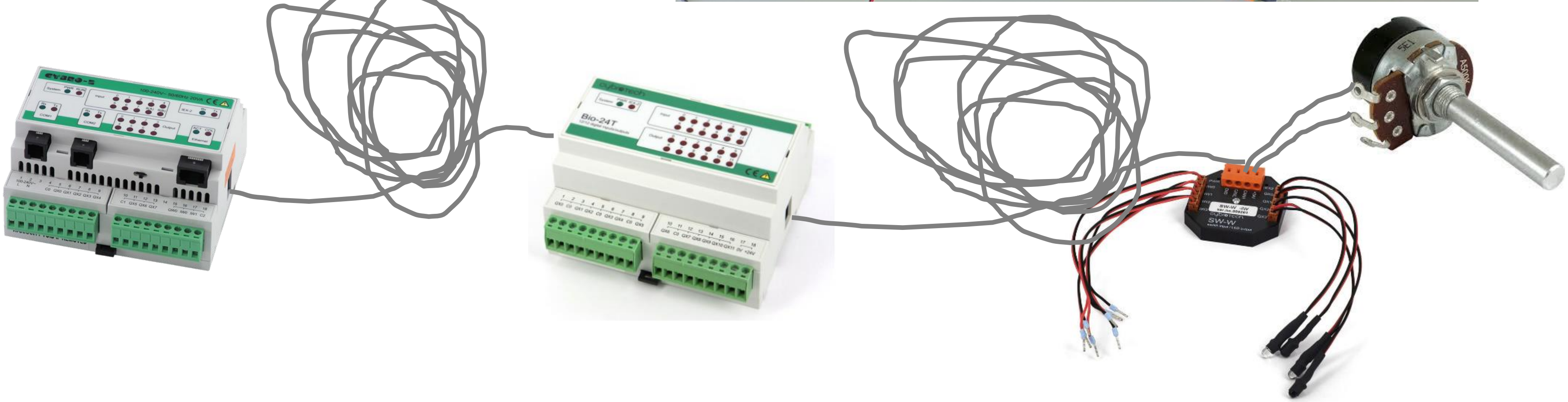
- 13.0: CANBUS osvežitev
- 13.1 Opis primera : Cybrotech CANBUS sistem
- 13.2: Krmiljenje Cybrotech IEX-2 modulov
- 13.3: STM32F4 – osnovni IEX-2 modul
- 13.4: CANBUS meritve

13.2: Krmiljenje Cybrotech IEX-2 modulov

Povežemo enostaven sistem :

- glavni krmilnik Cybro 2
- različni IEX moduli (V/I)

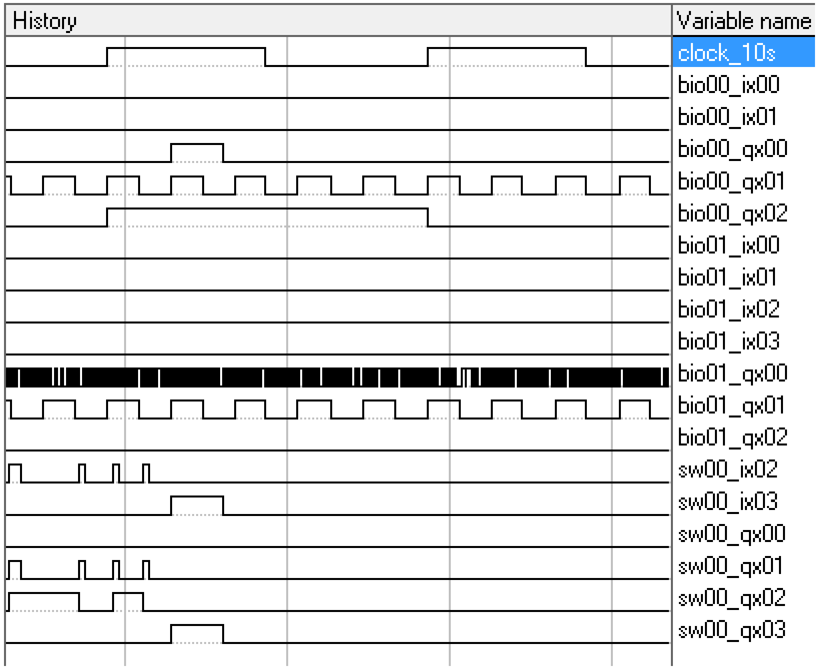




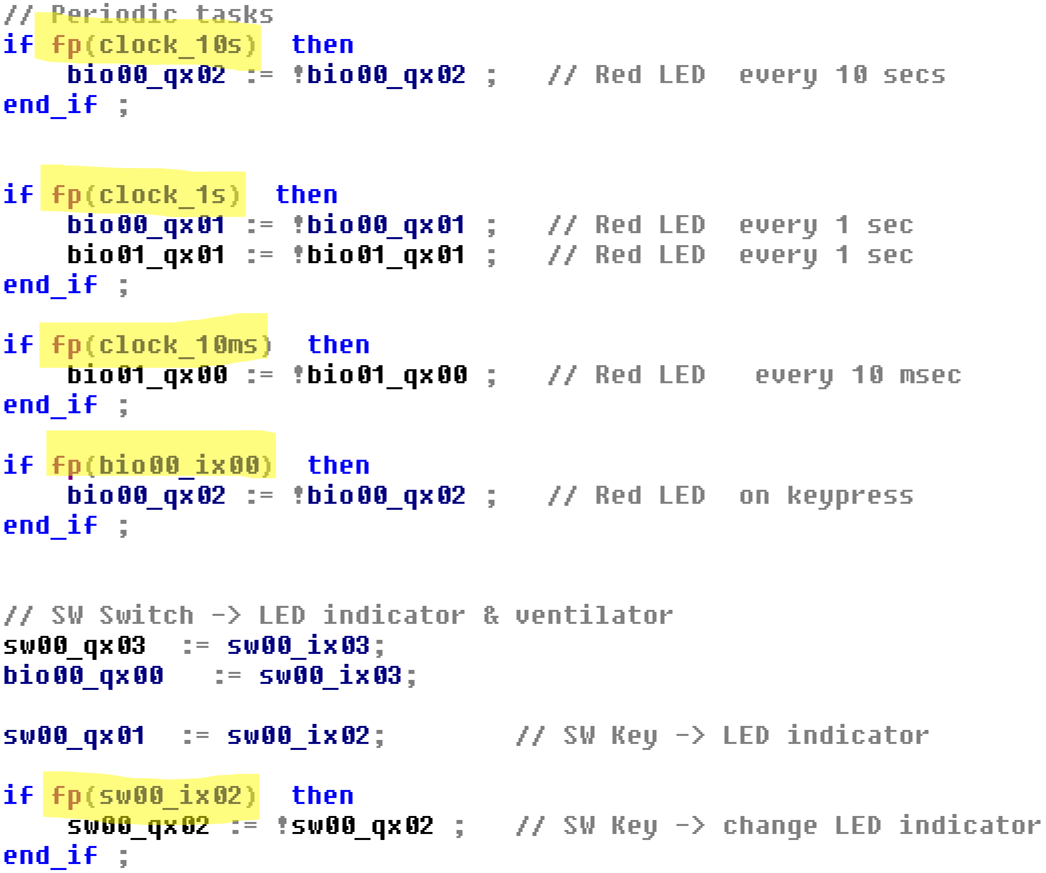
13.2: Krmiljenje Cybrotech IEX-2 modulov

Cypro IDE

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 ;
```

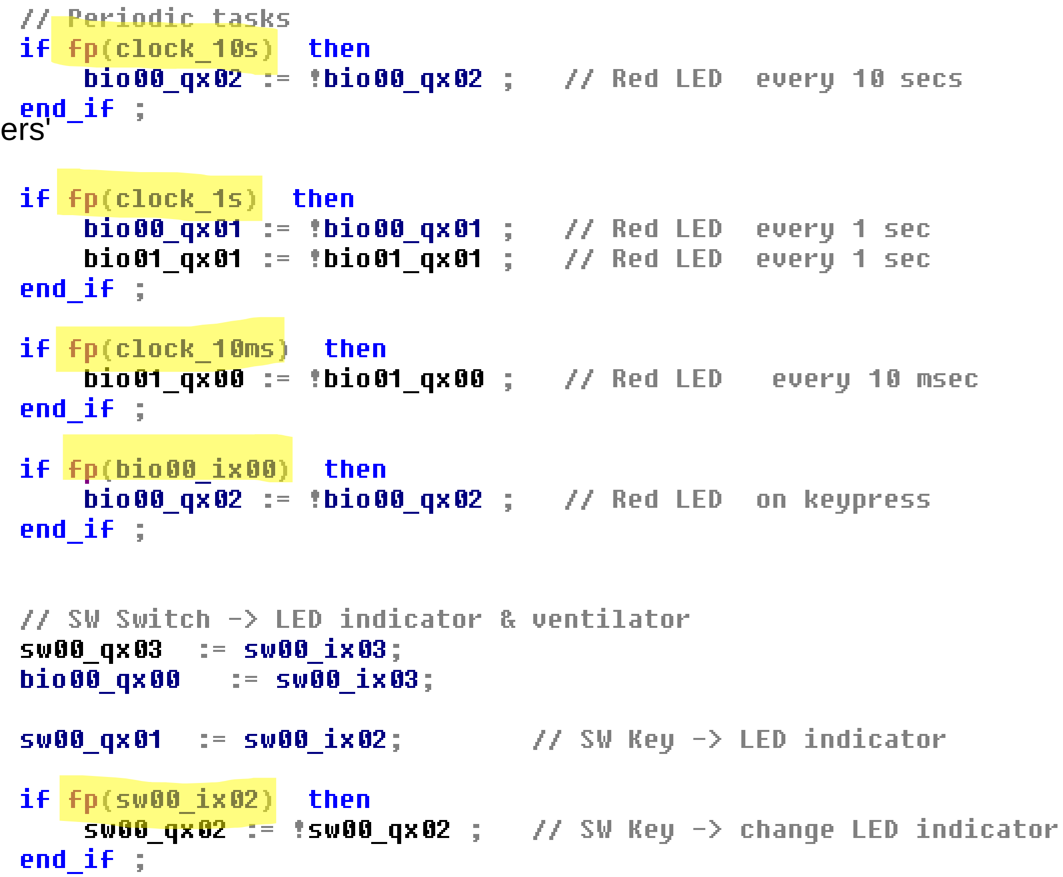
13.2: Krmiljenje Cybrotech IEX-2 modulov

Cypro IDE – opisi modulov v .cym datotekah

BIO-24.cym

```
object THWModule
  Name = 'Bio-24'
  CardID = 11
  Description = 'Binary 12 inputs/12 outputs, 4 fast counters'
  Capabilities = []
  DisplayWidth = 0
  DisplayHeight = 0
  MaskMemorySize = 0
  VarPrefix = 'bio?_'
  IOAllocData =
…
 item
     Typ = vaOutBit
     EventPriority = epOnChange
     Vars =
     <
       item
         Name = 'qx*'
         Description = 'Relay output (0-open, 1-closed).'
         Offset = 0
       end
…
```

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 ;
```

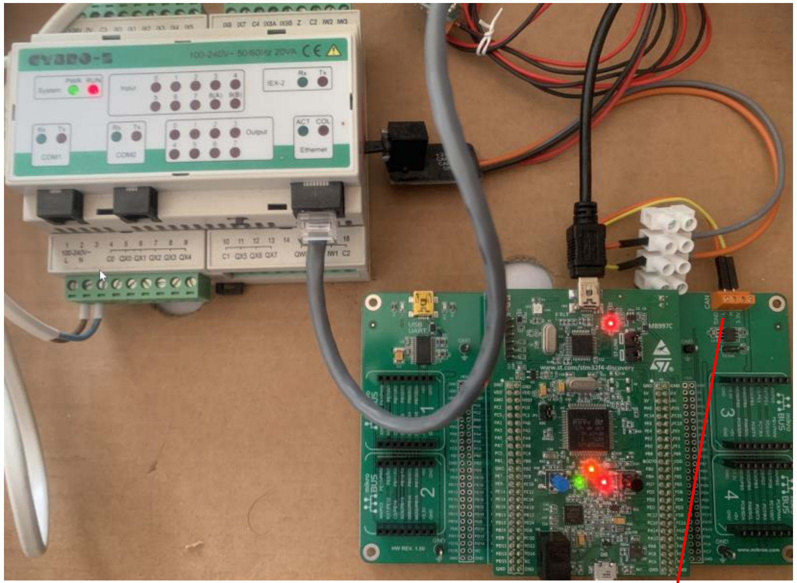
Laboratorijska vaja 13 - LV5

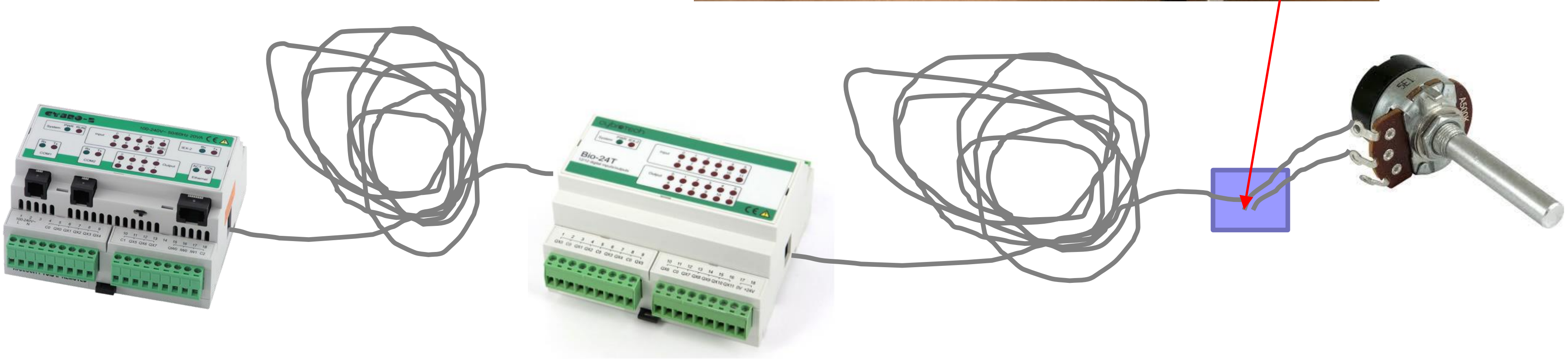
- 13.0: CANBUS osvežitev
- 13.1 Opis primera : Cybrotech CANBUS sistem
- 13.2: Krmiljenje Cybrotech IEX-2 modulov
- 13.3: STM32F4 – osnovni IEX-2 modul
- 13.4: CANBUS meritve

13.3: STM32F4 – osnovni IEX-2 modul

Strojna oprema:

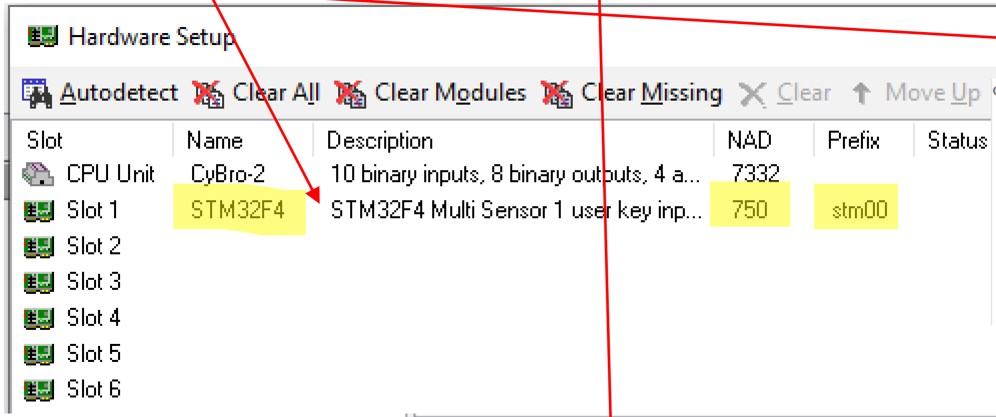
- STM32F4 Discovery in
- shield (Mikroelektronika)
  - vsebuje CANBUS PHY vezje
- ali zunanje CAN PHY vezje





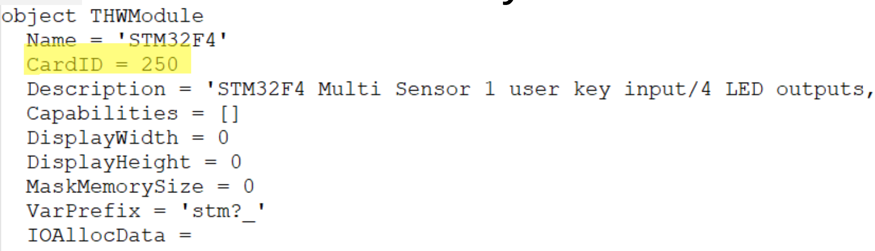
13.3: STM32 – osnovni IEX-2 modul

Vključitev in krmiljenje modula – Cypro IDE

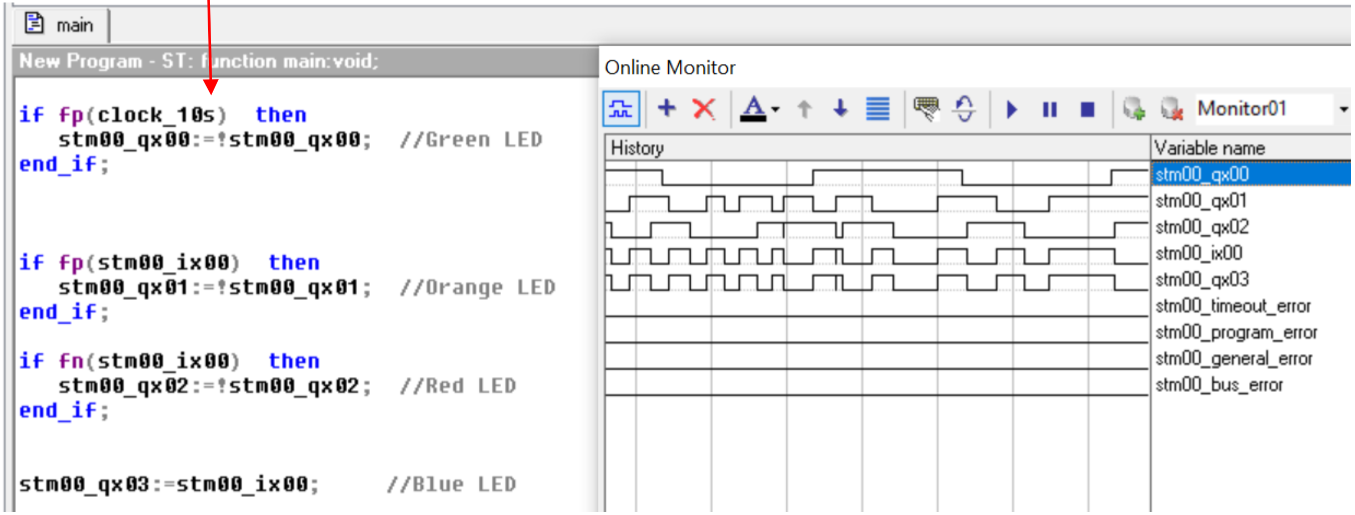


Slot	Name	Description	NAD	Prefix	Status
CPU Unit	CyBro-2	10 binary inputs, 8 binary outputs, 4 a...	7332		
Slot 1	STM32F4	STM32F4 Multi Sensor 1 user key inp...	750	stm00	
Slot 2					
Slot 3					
Slot 4					
Slot 5					
Slot 6					

STM32F4.cym



```
object THWModule
  Name = 'STM32F4'
  CardID = 250
  Description = 'STM32F4 Multi Sensor 1 user key input/4 LED outputs,
  Capabilities = []
  DisplayWidth = 0
  DisplayHeight = 0
  MaskMemorySize = 0
  VarPrefix = 'stm?_'
  IOAllocData =
```



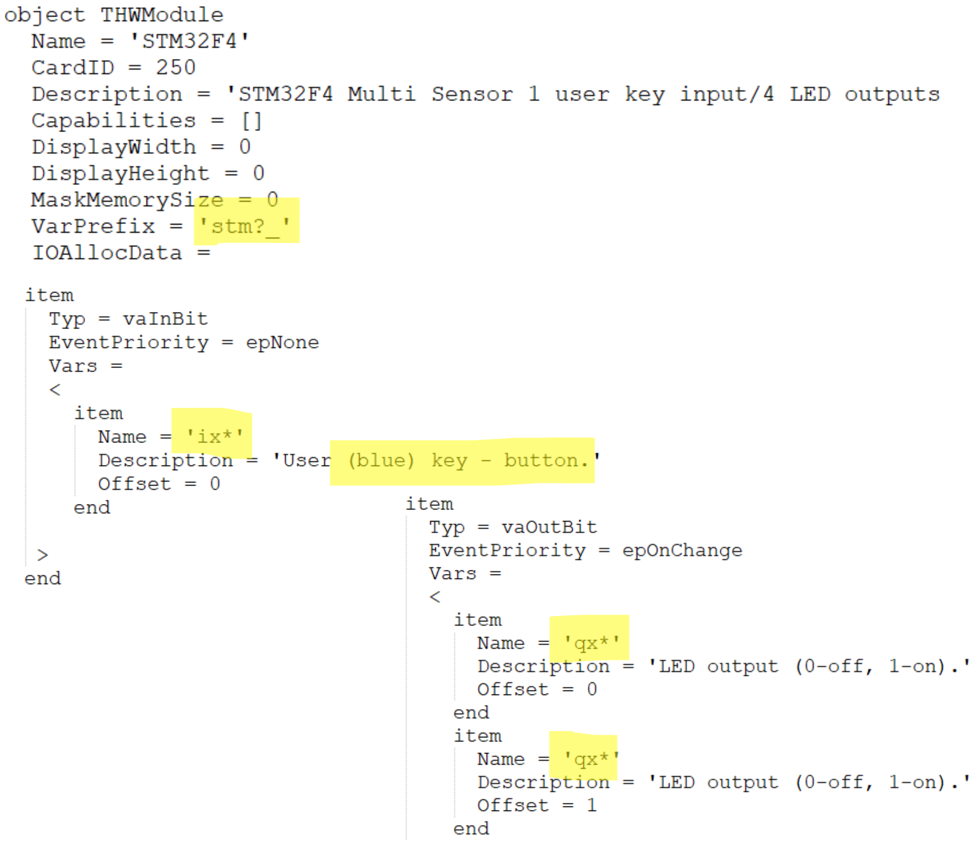
```
if fp(clock_10s)  then
   stm00_qx00:=!stm00_qx00;  //Green LED
end_if;


if fp(stm00_ix00)  then
   stm00_qx01:=!stm00_qx01;  //Orange LED
end_if;

if fn(stm00_ix00)  then
   stm00_qx02:=!stm00_qx02;  //Red LED
end_if;


stm00_qx03:=stm00_ix00;     //Blue LED
```

stm00_qx00
stm00_qx01
stm00_qx02
stm00_ix00
stm00_qx03
stm00_timeout_error
stm00_program_error
stm00_general_error
stm00_bus_error

13.3: STM32 – osnovni IEX-2 modul

Cypro IDE – opisi modulov so v .cym datotekah

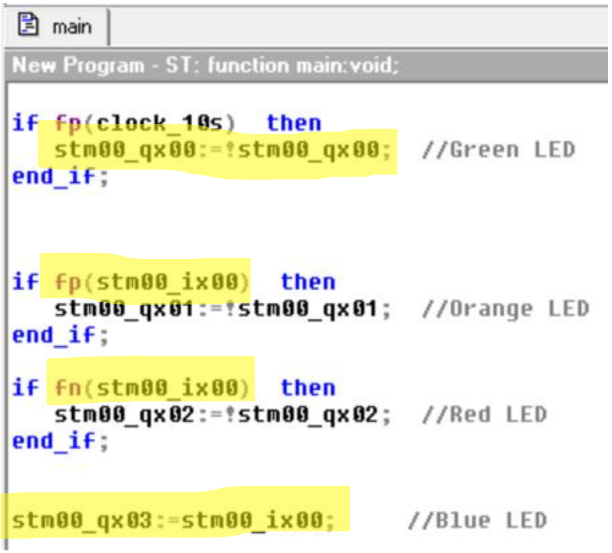
STM32F4.cym (definicija modula)



```
object THWModule
  Name = 'STM32F4'
  CardID = 250
  Description = 'STM32F4 Multi Sensor 1 user key input/4 LED outputs
  Capabilities = []
  DisplayWidth = 0
  DisplayHeight = 0
  MaskMemorySize = 0
  VarPrefix = 'stm?_'
  IOAllocData =

  item
    Typ = vaInBit
    EventPriority = epNone
    Vars =
    <
      item
        Name = 'ix*'
        Description = 'User (blue) key - button.'
        Offset = 0
      end

  >
  end
```

```
item
  Typ = vaOutBit
  EventPriority = epOnChange
  Vars =
  <
    item
      Name = 'qx*'
      Description = 'LED output (0-off, 1-on).'
      Offset = 0
    end
    item
      Name = 'qx*'
      Description = 'LED output (0-off, 1-on).'
      Offset = 1
    end
```

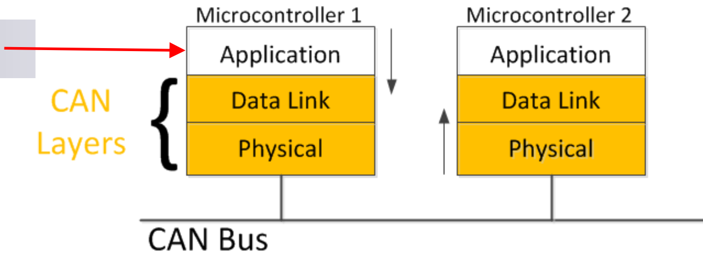
PLC program - uporaba



New Program - ST: function main:void;

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


if fp(stm00_ix00)  then
   stm00_qx01:=!stm00_qx01;  //Orange LED
end_if;

if fn(stm00_ix00)  then
   stm00_qx02:=!stm00_qx02;  //Red LED
end_if;

stm00_qx03:=stm00_ix00;     //Blue LED
```

Primer protokola 3. nivo



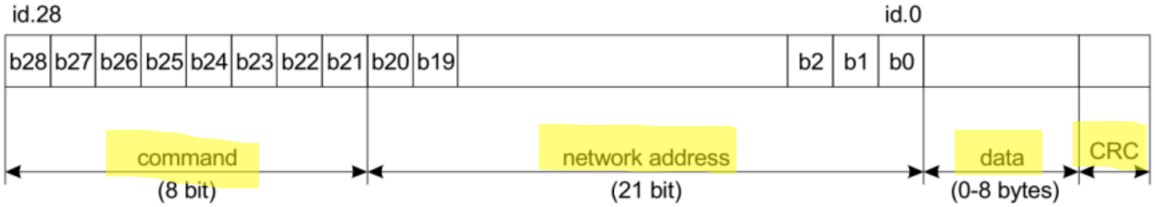
INTEGRA BM SYSTEM

IEX protocol
(nadgradnja CANBUS)

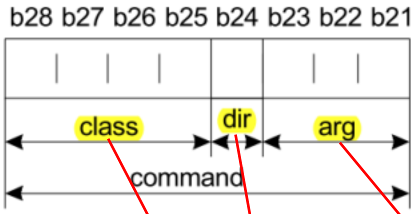
IEX PROTOCOL v2.8 POVZETEK

General

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



Command summary



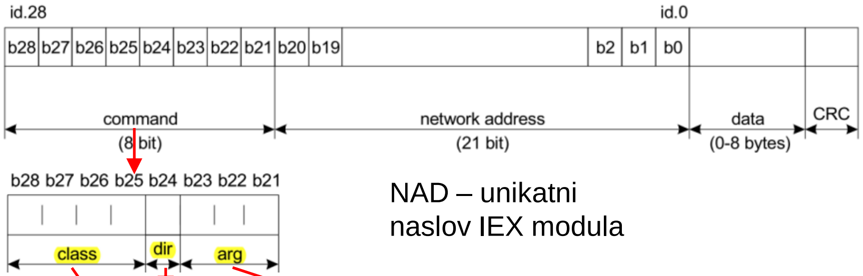
NAD – unikatni naslov IEX modula

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

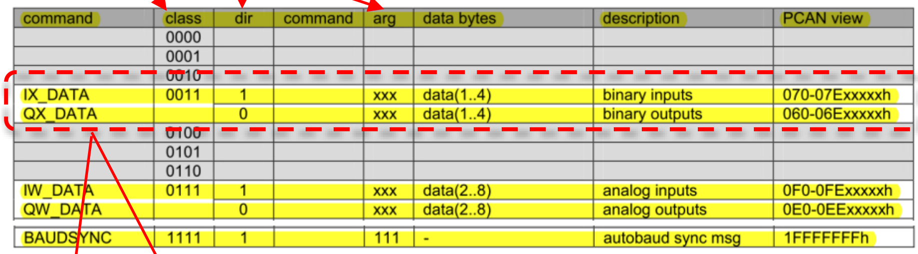
IEX PROTOCOL v2.8 POVZETEK

General

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



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

IX_DATA : modul sporoči stanje dig. vhodov

QX_DATA: modul sprejme stanje dig. izhodov

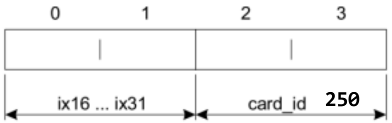
definicije:

```
#define IEX2_CYM_ID_V1      250   // 255 is max, select unique ID, also
specified in .cym file

#define IEX2_DIRECTION_NODE2RC  0x1000000
#define IEX2_COMMAND_BIT_DATA   0x6000000
#define IEX2_ARGUMENT_IO_DATA0   0x000000

//const unsigned long status_id=NAD + 0x7800000 ;
#define IEX2_ID_SEND_ONBUS_STATUS  (IEX2_COMMAND_BIT_DATA |
IEX2_DIRECTION_NODE2RC | IEX2_ARGUMENT_SYS_DATA16)

// IX_id=NAD_v1+0x7000000 ; IX_data command id for sending input bits IX
#define IEX2_ID_SEND_IX0_STATUS    (IEX2_COMMAND_BIT_DATA |
IEX2_DIRECTION_NODE2RC | IEX2_ARGUMENT_IO_DATA0)

//IX_system_data command id for sending onbus status
volatile unsigned long status_id = NAD_V4_default +IEX2_ID_SEND_ONBUS_STATUS;
volatile unsigned char status_data[4] = {0,0,0,IEX2_CYM_ID_V4};
```

STATUS_ID

STATUS_ID is a special case of IX_DATA message. It contains data bits ix16-ix31 (2 bytes) and card_id (2 bytes):

IEX → PLC   IX_DATA + 100 +



```
// Send Status/Info message every 0.5 second
nowTime = HAL_GetTick();
if ((nowTime - lastTime) >= 500) {
        CANBus_Send(status_id, status_data, 4, 0, 0);
        lastTime = nowTime;
}
```

Module must send STATUS_ID every 500ms (+/-10ms). Module may send a range of input bits at any time (IX_DATA with no card_id bytes), but that is not considered as status message. STATUS_ID is used for module autodetection.

13.3: STM32 – osnovni IEX-2 modul

Programska oprema – CubeIDE Projekt - izseki

main.c:

```
//IX_system_data command id for sending onbus status
volatile unsigned long status_id = NAD_default +IEX2_ID_SEND_ONBUS_STATUS;
volatile unsigned char status_data[4] = {0,0,0,IEX2_CYM_ID_V1};
// IX_data command id for sending input bits IX
volatile unsigned long IX_id = NAD_default + IEX2_ID_SEND_IX0_STATUS ;
volatile unsigned char IX_data[2] = {0, 0};

while (1)
  {
//  Check for received CANBUS messages
  if(HAL_CAN_GetRxFifoFillLevel(&hcan1, CAN_RX_FIFO0) != 0)
       {
         HAL_CAN_GetRxMessage(&hcan1, CAN_RX_FIFO0, &RxHeader, CAN_Rx_Msg);
         CanMsgCnt++;

         if (RxHeader.IDE) {
           CANBus_Parse_RX_Message (RxHeader.ExtId,RxHeader.DLC, CAN_Rx_Msg);
         }
         …
  }
  // Send Status/Info message every 0.5 second
  nowTime = HAL_GetTick();
  if ((nowTime - lastTime) >= 500) {
          CANBus_Send(status_id, status_data, 4, 0, 0);
          lastTime = nowTime;
  }
  // Check USER Key state
  temp = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0);
  if (temp != KeyState) {   // Key state changed !!! - send as IX message
          ???
  }
  …
  }
```

main.h:

```
#define NAD_default (long)750// Defines Node V1 NAD for IEX
// These are IDs that are reported to IEX master for module
identification (read appropriate .cym files)
#define IEX2_CYM_ID_V1      250    // 255 is max, select
unique ID, also specified in .cym file

#define IEX2_DIRECTION_NODE2RC  0x1000000
#define IEX2_DIRECTION_RC2NODE  0x0000000

#define IEX2_COMMAND_BIT_DATA   0x6000000
#define IEX2_COMMAND_WORD_DATA  0xe000000

#define IEX2_ARGUMENT_IO_DATA0    0x000000
#define IEX2_ARGUMENT_IO_DATA4    0x200000
#define IEX2_ARGUMENT_SYS_DATA16 0x800000

//const unsigned long status_id=NAD + 0x7800000 ;
#define IEX2_ID_SEND_ONBUS_STATUS   (IEX2_COMMAND_BIT_DATA |
IEX2_DIRECTION_NODE2RC | IEX2_ARGUMENT_SYS_DATA16)

// IX_id=NAD_v1+0x7000000 ; IX_data command id for sending
input bits IX
#define IEX2_ID_SEND_IX0_STATUS    (IEX2_COMMAND_BIT_DATA |
IEX2_DIRECTION_NODE2RC | IEX2_ARGUMENT_IO_DATA0)
```

iex.c:

```
uint_32 CANBus_Parse_RX_Message (uint_32 ID,uint_32 msg_size,
unsigned char dptr [])
{ … }

unsigned char CANBus_Send(volatile unsigned long Id, volatile
unsigned char MessageData[],volatile unsigned char
MessageLen,volatile unsigned char MessageType, volatile
unsigned char Debug) { … }
```

13.3: STM32 – osnovni IEX-2 modul - rešitev

Programska oprema – CubeIDE Projekt - izseki

main.c:

```c
//IX_system_data command id for sending onbus status
volatile unsigned long status_id = NAD_default +IEX2_ID_SEND_ONBUS_STATUS;
volatile unsigned char status_data[4] = {0,0,0,IEX2_CYM_ID_V1};
// IX_data command id for sending input bits IX
volatile unsigned long IX_id = NAD_default + IEX2_ID_SEND_IX0_STATUS ;
volatile unsigned char IX_data[2] = {0, 0};

while (1)
  {
       //  Check for received CANBUS messages
         if(HAL_CAN_GetRxFifoFillLevel(&hcan1, CAN_RX_FIFO0) != 0)
              {
                  HAL_CAN_GetRxMessage(&hcan1, CAN_RX_FIFO0, &RxHeader, CAN_Rx_Msg);
                  CanMsgCnt++;

                  if (RxHeader.IDE) {
                      CANBus_Parse_RX_Message (RxHeader.ExtId,RxHeader.DLC, CAN_Rx_Msg);
                  }
                  …
        }

        // Send Status/Info message every 0.5 second
        nowTime = HAL_GetTick();
        if ((nowTime - lastTime) >= 500) {
                CANBus_Send(status_id, status_data, 4, 0, 0);
                lastTime = nowTime;
        }

        // Check USER Key state
        temp = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0);
                // Simple debounce
                HAL_Delay(50);
                if (temp == HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0)) {
                IX_data[0]=temp;
                 CANBus_Send(IX_id, IX_data, 1, 0, 0);
                 KeyState = temp;

                }
        …
}
```

main.h:

```c
#define NAD_default (long)750// Defines Node V1 NAD for IEX
// These are IDs that are reported to IEX master for module
identification (read appropriate .cym files)
#define IEX2_CYM_ID_V1      250    // 255 is max, select
unique ID, also specified in .cym file

#define IEX2_DIRECTION_NODE2RC  0x1000000
#define IEX2_DIRECTION_RC2NODE  0x0000000

#define IEX2_COMMAND_BIT_DATA   0x6000000
#define IEX2_COMMAND_WORD_DATA  0xe000000

#define IEX2_ARGUMENT_IO_DATA0    0x000000
#define IEX2_ARGUMENT_IO_DATA4    0x200000
#define IEX2_ARGUMENT_SYS_DATA16 0x800000

//const unsigned long status_id=NAD + 0x7800000 ;
#define IEX2_ID_SEND_ONBUS_STATUS  (IEX2_COMMAND_BIT_DATA |
IEX2_DIRECTION_NODE2RC | IEX2_ARGUMENT_SYS_DATA16)

// IX_id=NAD_v1+0x7000000 ; IX_data command id for sending
input bits IX
#define IEX2_ID_SEND_IX0_STATUS    (IEX2_COMMAND_BIT_DATA |
IEX2_DIRECTION_NODE2RC | IEX2_ARGUMENT_IO_DATA0)
```

iex.c:

```c
uint_32 CANBus_Parse_RX_Message (uint_32 ID,uint_32 msg_size,
unsigned char dptr [])
{ … }

unsigned char CANBus_Send(volatile unsigned long Id, volatile
unsigned char MessageData[],volatile unsigned char
MessageLen,volatile unsigned char MessageType, volatile
unsigned char Debug) { … }
```

13.3: STM32 – osnovni IEX-2 modul - rešitev

iex.c:

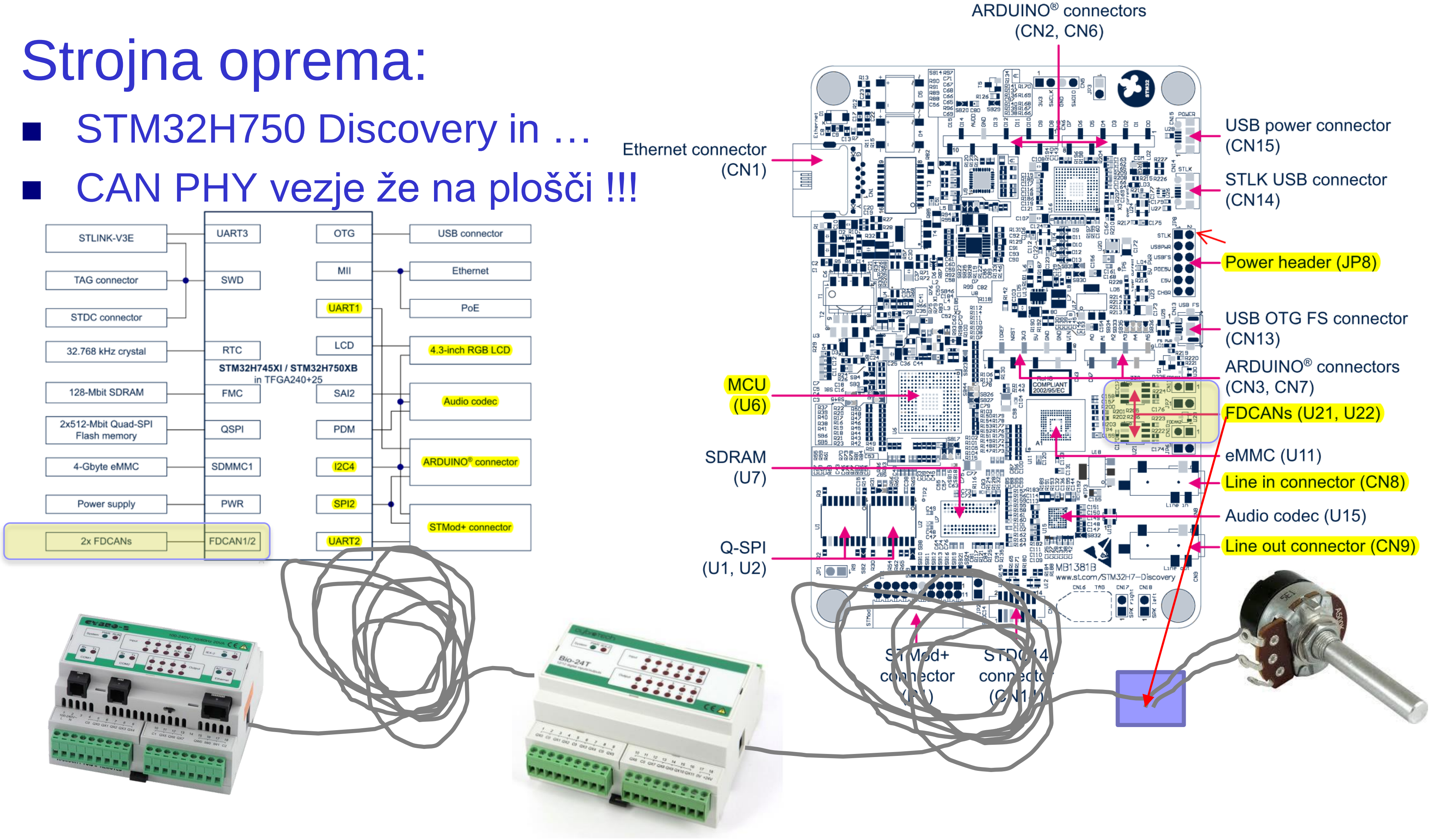
Programska oprema – CubeIDE Projekt - izseki

```
uint_32 CANBus_Parse_RX_Message (uint_32 ID,uint_32 msg_size, unsigned char dptr [])
{
  int iex_cmd;
  long iex_NAD;
  int iex_arg;
  int iex_slot;
  uint_8 bitmask, iex_dir, iex_class;
  uint_16 ix_temp;



  iex_cmd = ID >> 21;
  iex_NAD = ID & 0x1fffff;
  iex_arg = iex_cmd & ARG_MASK;
  iex_dir = (iex_cmd & DIR_MASK) >> 3;
  iex_class = (iex_cmd & CLASS_MASK) >> 4 ;

  if (msg_size >= 0) {
  iex_slot = 0;  //not used


      if (1) {
          if ( 1 ) {
               if ((iex_cmd & (CLASS_MASK | DIR_MASK))== IX_DATA)  {       /* group of IX variables */
                       if (iex_arg==IX_STATUS)   {  /* Status ID message */
                        …
                        }

               } else if ((iex_cmd & (CLASS_MASK | DIR_MASK))== QX_DATA)  {       /* group of IX variables */
                       if  ( (msg_size == 1)  && (iex_arg == 0) ) {
                               ix_temp = dptr[0];
                               bitmask = 0x01;
                                if (iex_NAD == NAD_default)  { // Message for this node - transfer QX data to actual outputs - LEDs !!!
                                       HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12, ix_temp & 0x01);
                                       HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, (ix_temp & 0x02) >> 1);
                                       HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, (ix_temp & 0x04) >> 2);
                                       HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, (ix_temp & 0x08) >> 3);
                               }
```

13.3: STM32H7 – osnovni IEX-2 modul - Ideja

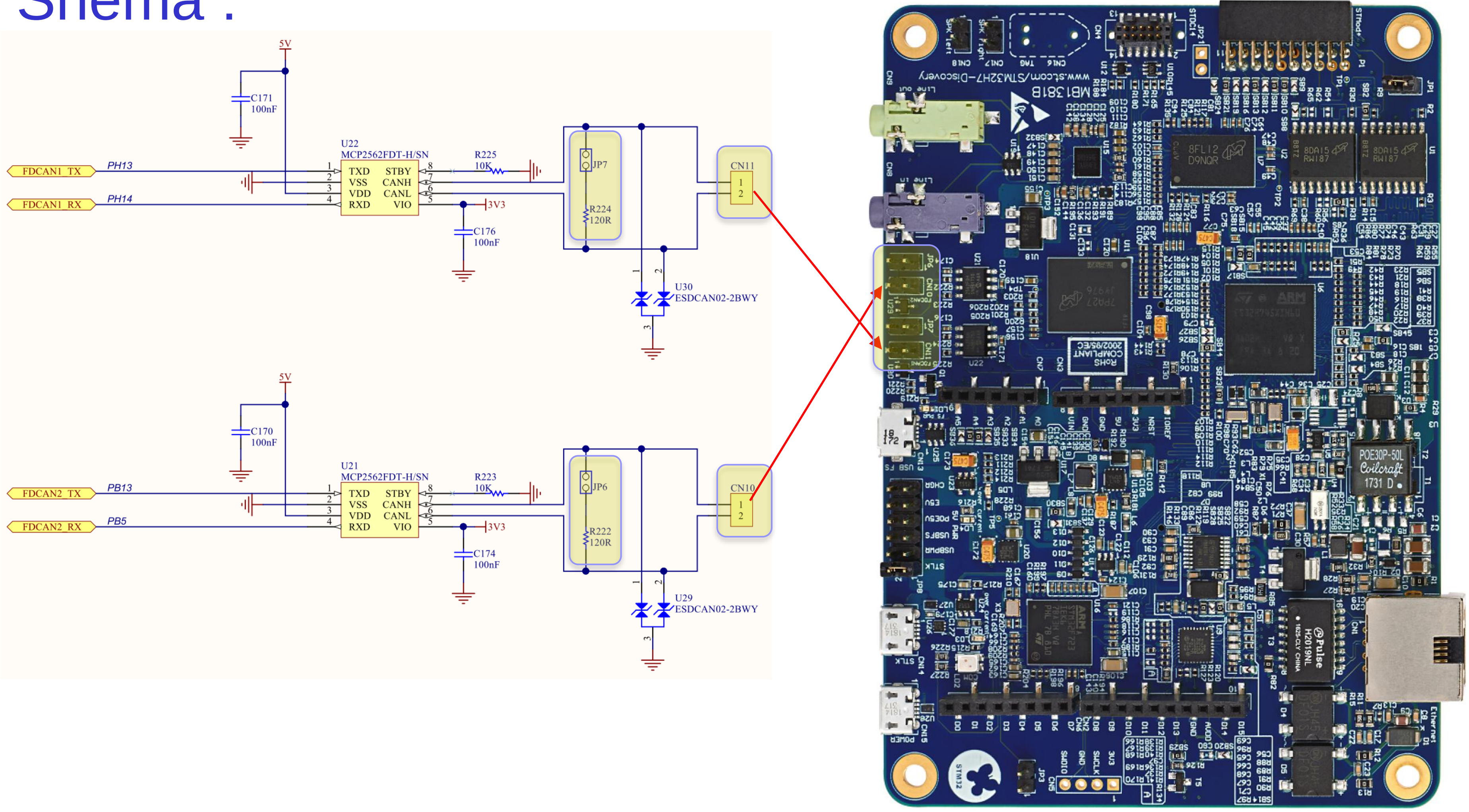
Strojna oprema:

- STM32H750 Discovery in …
- CAN PHY vezje že na plošči !!!



13.3: STM32H7 – osnovni IEX-2 modul - Ideja

Shema :



Laboratorijska vaja 13 - LV5

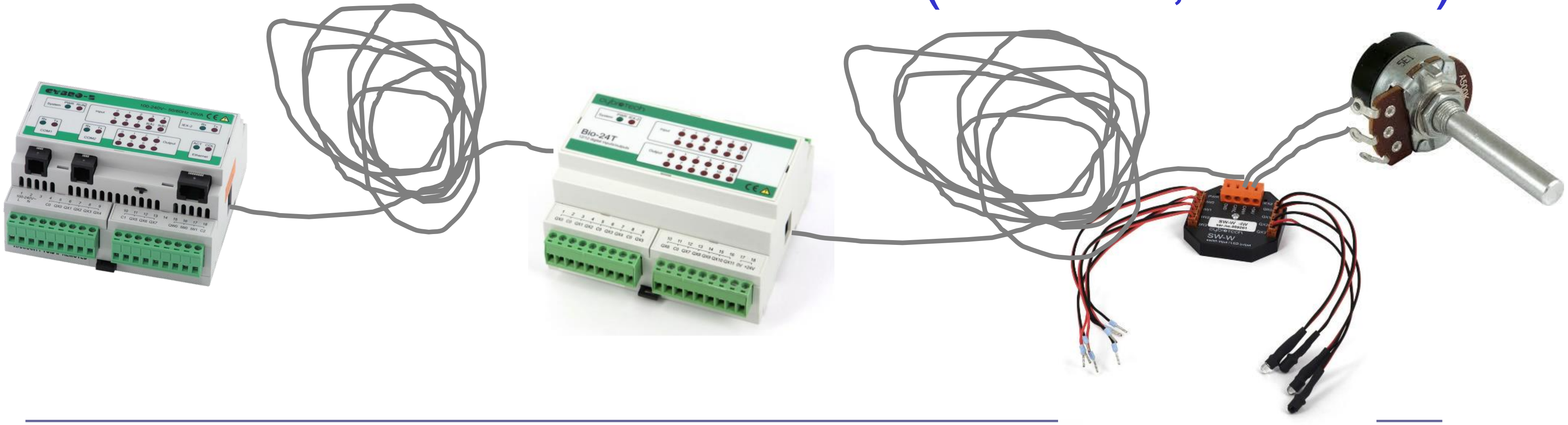
- 13.0: CANBUS osvežitev
- 13.1 Opis primera : Cybrotech CANBUS sistem
- 13.2: Krmiljenje Cybrotech IEX-2 modulov
- 13.3: STM32F4 – osnovni IEX-2 modul
- 13.4: CANBUS meritve

13.4: CANBUS meritve

Izmerite stanje na vodilu pri :

- Različnih zaključitvah na koncu vodila
  - Odprte sponke, 500ohm, zaključitev (107ohm)

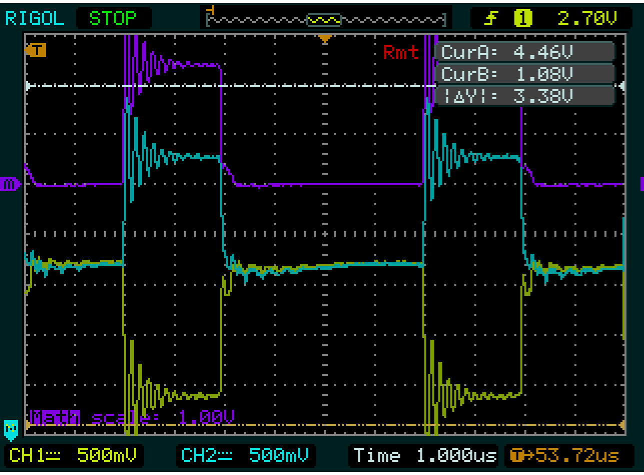
- Dveh različnih bitnih hitrostih (500kb/s, 100kb/s)



13.4: CANBUS meritve

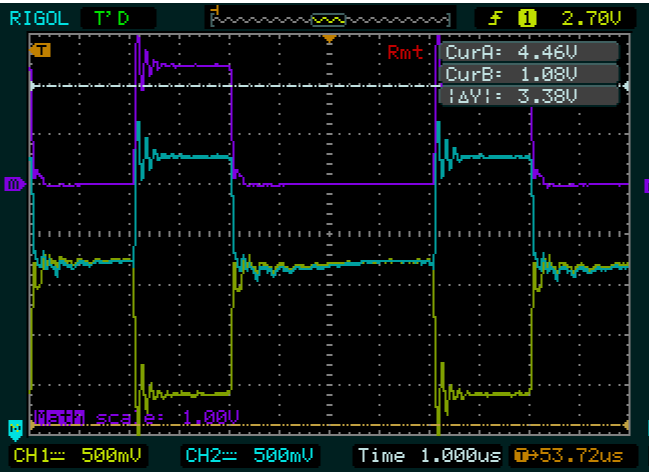
500kb/s:

Odprte sponke



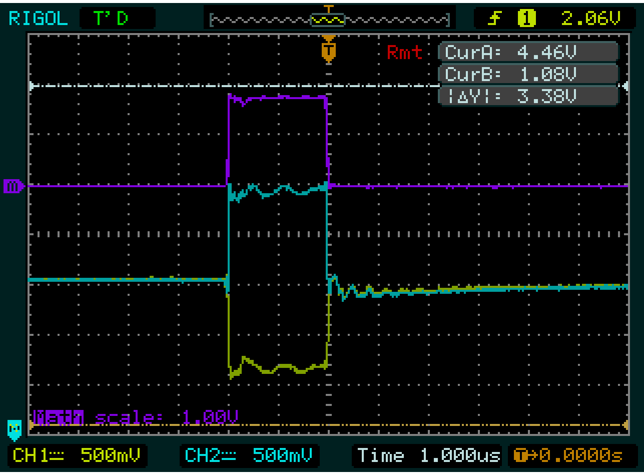
```
RIGOL  STOP                    1  2.70V
Rmt   CurA: 4.46V
      CurB: 1.08V
      |ΔY|: 3.38V
Math  scale: 1.00V
CH1  500mV   CH2  500mV   Time 1.000us   T→53.72us
```

500ohm



```
RIGOL  T'D                     1  2.70V
Rmt   CurA: 4.46V
      CurB: 1.08V
      |ΔY|: 3.38V
Math  scale: 1.00V
CH1  500mV   CH2  500mV   Time 1.000us   T→53.72us
```

107ohm



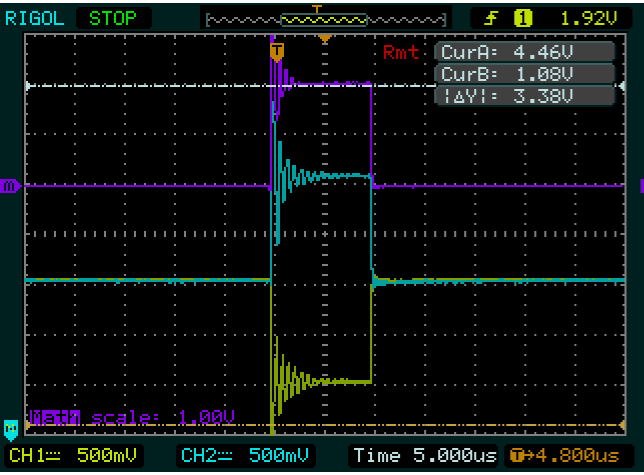
```
RIGOL  T'D                     1  2.06V
Rmt   CurA: 4.46V
      CurB: 1.08V
      |ΔY|: 3.38V
Math  scale: 1.00V
CH1  500mV   CH2  500mV   Time 1.000us   T→0.0000s
```

3 zavitki UTP kabla s spojniki – cca 40m…

13.4: CANBUS meritve

100kb/s:

Odprte sponke



RIGOL STOP

1.92V

Rmt CurA: 4.46V

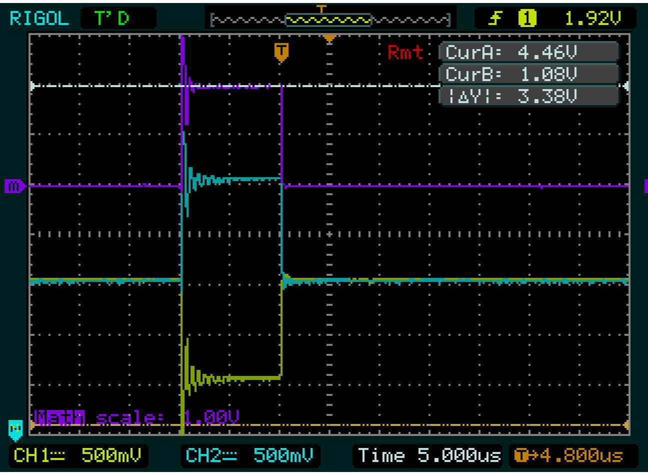
CurB: 1.08V

|ΔY|: 3.38V

Math scale: 1.00V

CH1 500mV CH2 500mV Time 5.000us T→4.800us

500ohm



RIGOL T'D

1.92V

Rmt CurA: 4.46V

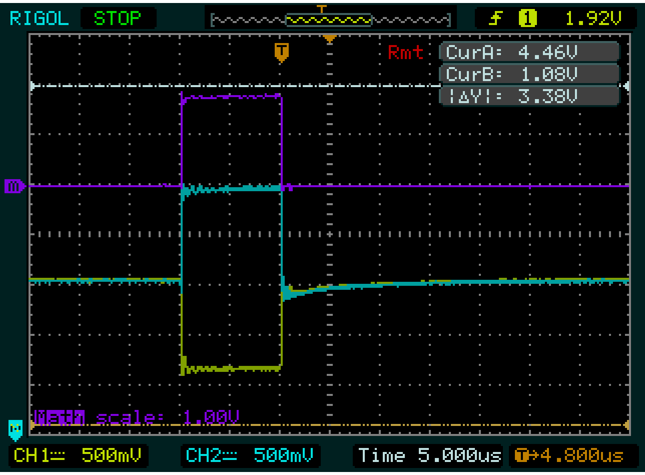
CurB: 1.08V

|ΔY|: 3.38V

Math scale: 1.00V

CH1 500mV CH2 500mV Time 5.000us T→4.800us

107ohm



RIGOL STOP

1.92V

Rmt CurA: 4.46V

CurB: 1.08V

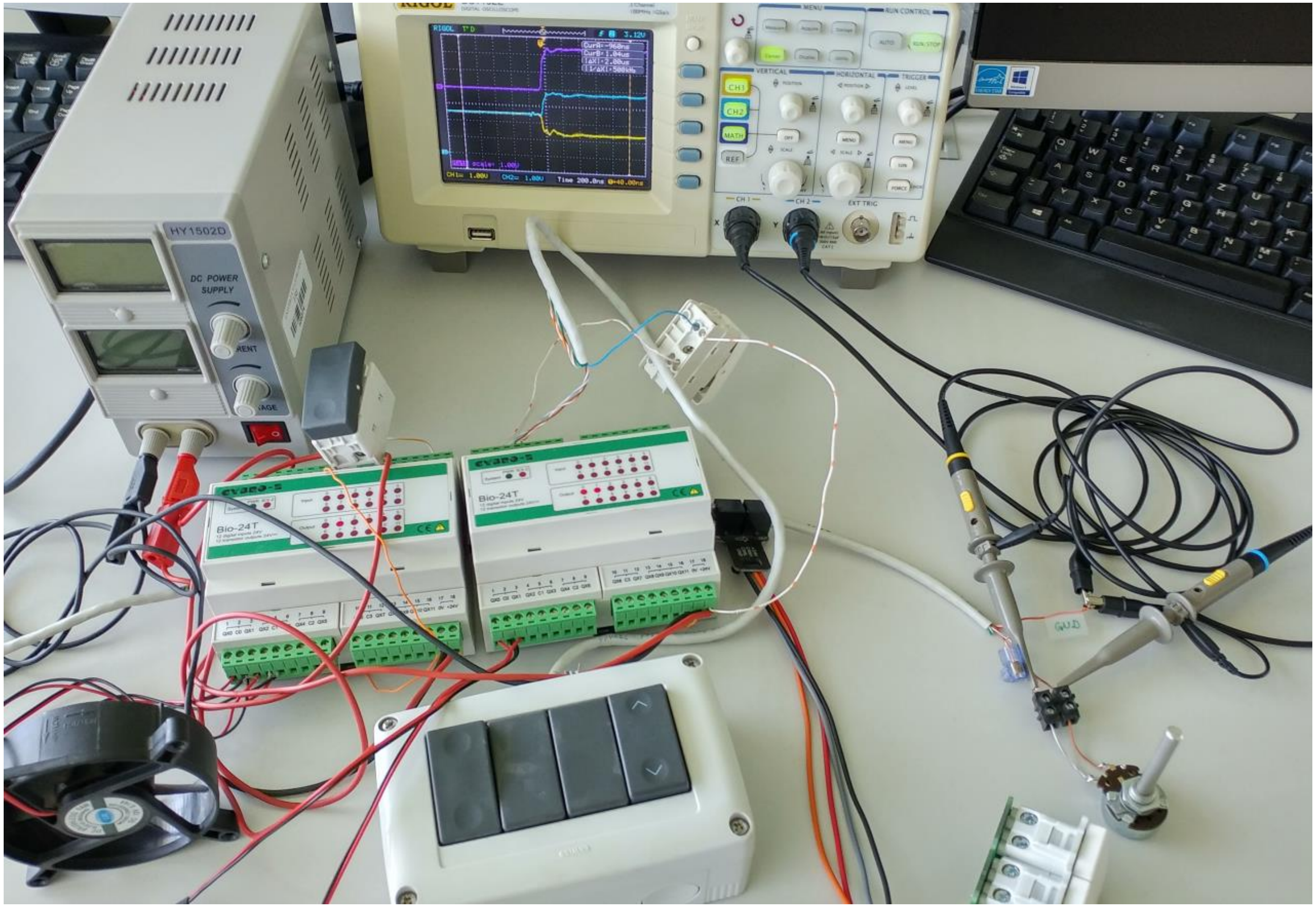
|ΔY|: 3.38V

Math scale: 1.00V

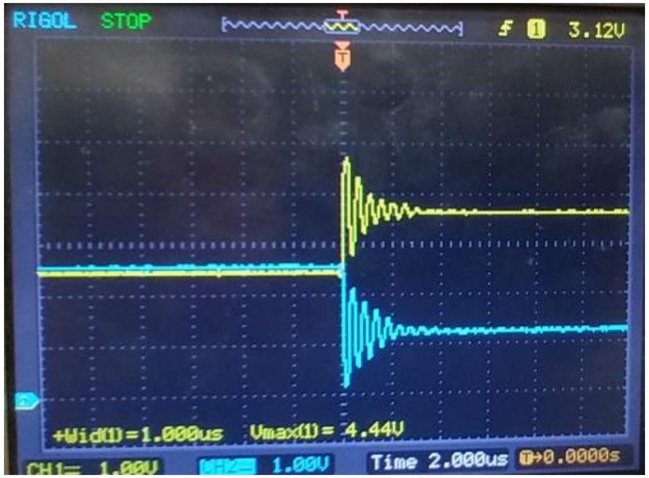
CH1 500mV CH2 500mV Time 5.000us T→4.800us

3 zavitki UTP kabla s spojniki – cca 40m…

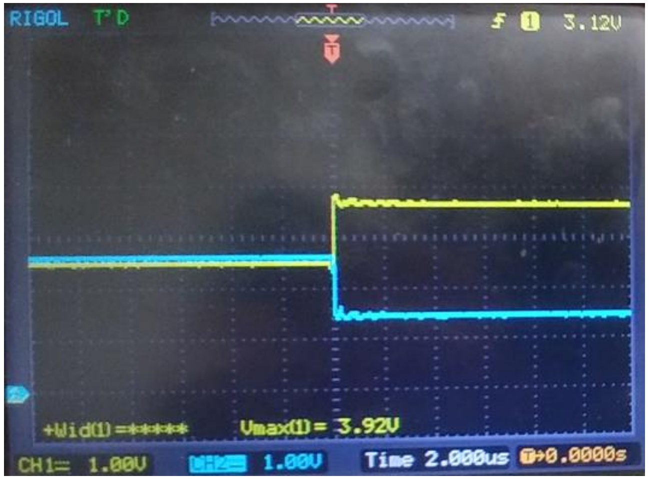
13.4: CANBUS meritve



13.4: CANBUS meritve



Nezaključena linija



Zaključena linija