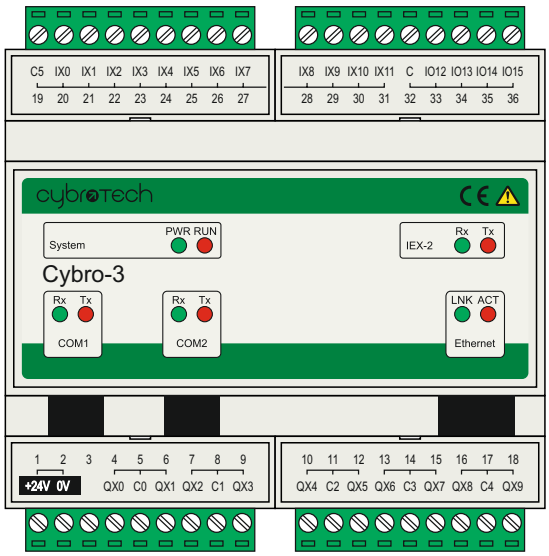
Cybro-3

IoT PLC compatible with IEX-2 modules and tools



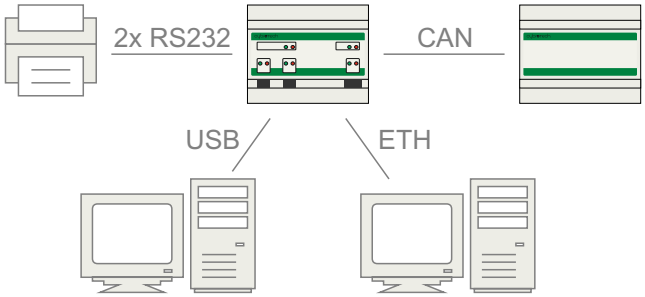
12 opto-isolated inputs 24V
10 relay outputs 230V 8A
4 universal input/outputs
1Mb flash, 64Kb RAM, 32Kb EE
48 expansion slots for I/O modules
USB programming port
CAN diagnostic monitor
extra long life (no electrolytic capacitors)

CPU

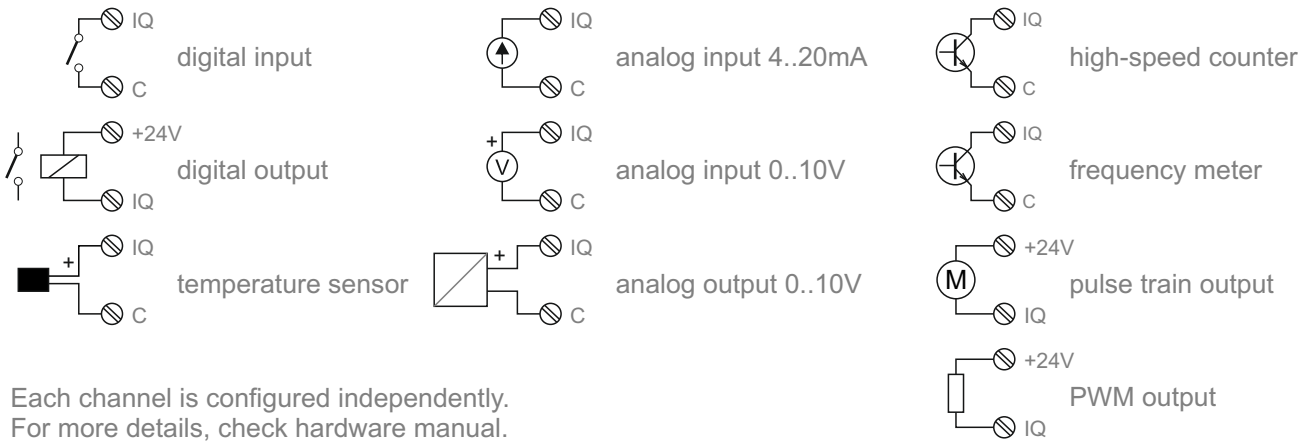
Model	STM32F412
Architecture	ARMv7E-M 32-bit
Core	Cortex-M4 at 100MHz
Coprocessor	FPU single precision
Power supply	3.3V external, 1.2V core
Performance	120 DMIPS, 340 CoreMark
Flash	internal 1Mb, 128-bit
Endurance	10000 program/erase cycles
Data retention	20 years minimum
SDRAM	internal 256Kb, 0 wait state
Retentive RAM	external 64Kb, SPI 20MHz
EEPROM	external 32Kb, I2C 400kHz

Comparison	Cybro-3	Cybro-2
digital inputs	12x 24V	10x 24V
relay outputs	10x 8A	8x 5A
analog i/o	4 universal	4 in, 1 out
CPU rating	120 MIPS	6 MIPS
expansion slots	48	32
plc program	1Mb	64Kb
plc variables	64Kb	32Kb

Communication ports



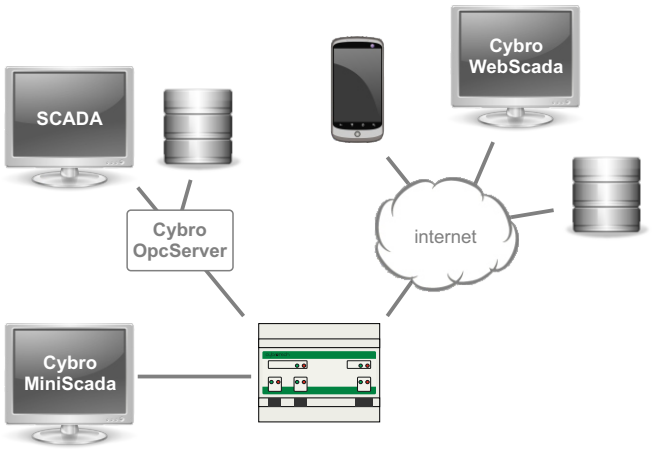
Universal input/output



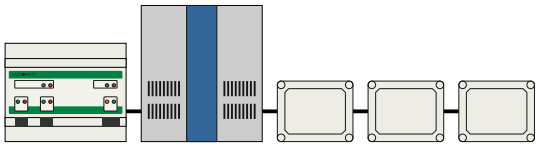
Each channel is configured independently.
For more details, check hardware manual.

Industrial automation

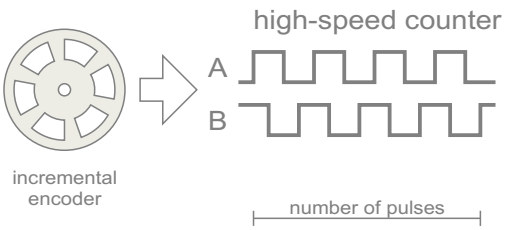
M2M and IoT applications



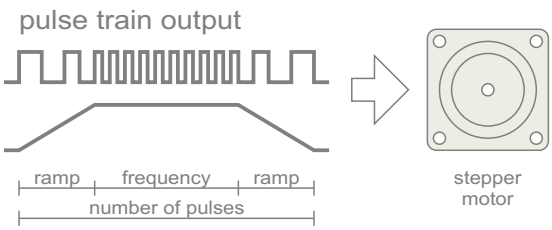
PV monitoring



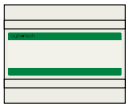
Positioning



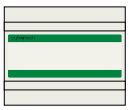
Motion control



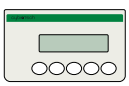
IEX-2 modules for industrial applications



Bio-24R
12 opto inputs
12 relay outputs



Bio-24T
12 opto inputs
12 transistor outputs



OP-2
operator panel

AiR-12
12 Pt100 inputs

AiV-12
12 inputs 0..10V

AoV-12
12 outputs 0..10V

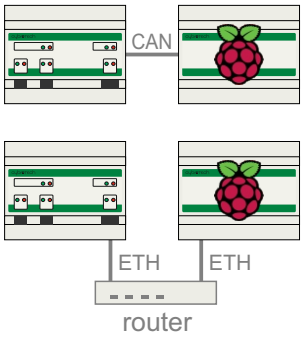
COM-485
protocol converter

COM-232
protocol converter

COM-CAN
protocol converter

For a complete list of modules, check hardware manual.

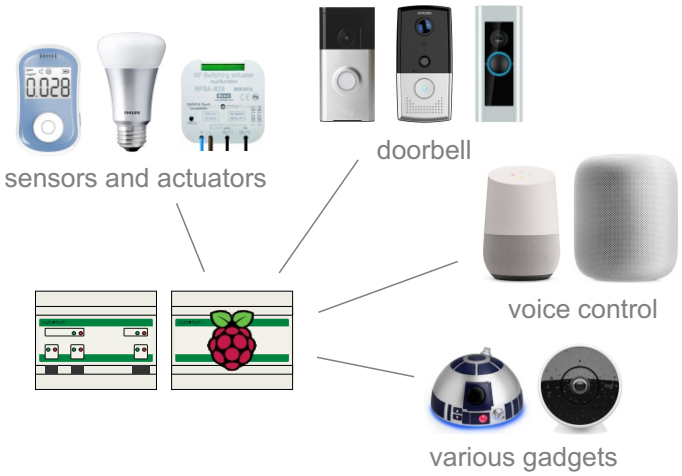
Home and building automation



CAN connection

LAN connection

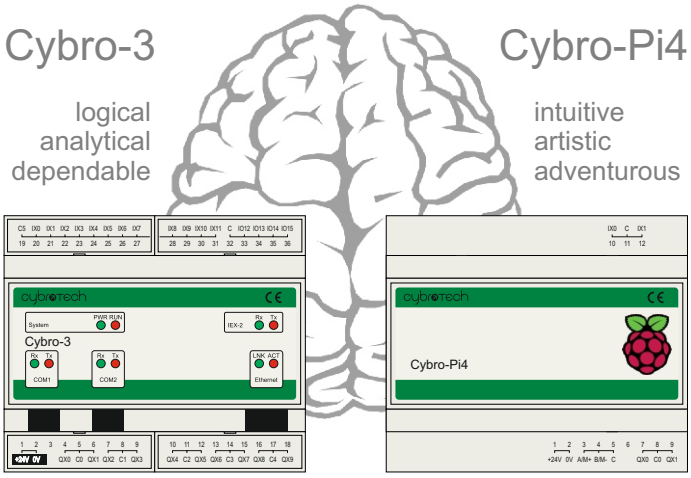
Cybro and Raspberry may connect directly on CAN bus, or using the local network. The first method is recommended for a small amount of data.



Cybro-3

logical
analytical
dependable

- low-level control
- simple automation
- time-related actions



Cybro-Pi4

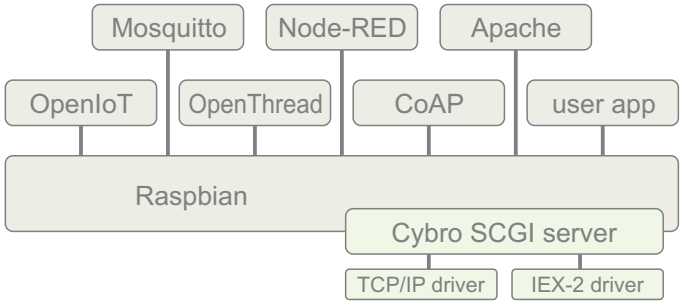
intuitive
artistic
adventurous

- data log
- secure connection
- various applications

Cybro is simple, tough and reliable. No drivers, no updates, no viruses. When button is pressed, light goes on, no matter what.

Raspberry is a huge ecosystem of various projects that can be combined with Cybro. Main system is fully independent, so reliability is not critical.

Application examples



SCGI server make Cybro variables visible to Raspberry. It is compatible with all RPi models and tiny PCs, both Linux and Windows.

IEX-2 modules for buildings

For a complete list of modules, check HIQ manual.

Migration to Cybro-3

This steps ought to be performed to replace Cybro-2 with Cybro-3 controller within the existing system. New controller supersedes the old one in most features, though it's not a drop-in replacement. A few changes should be made, both in hardware and software, to get the system up and running.

Power supply

Cybro-2 was 24V or 230V, Cybro-3 is 24V only, so an external power adapter may be needed.

Terminals

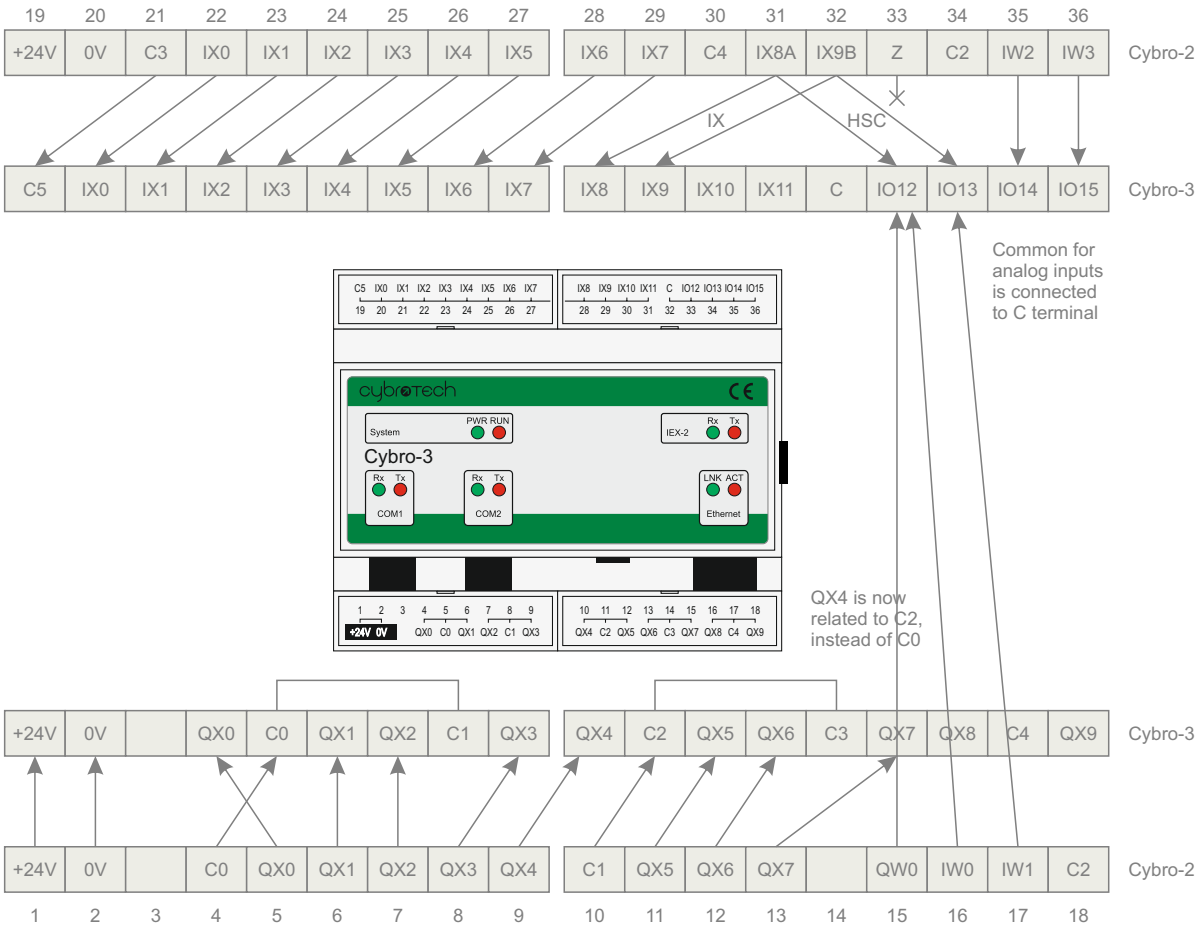
Cybro-2 and Cybro-3 connections are not equal, however, transition to new layout is pretty much straight-forward.

Not supported

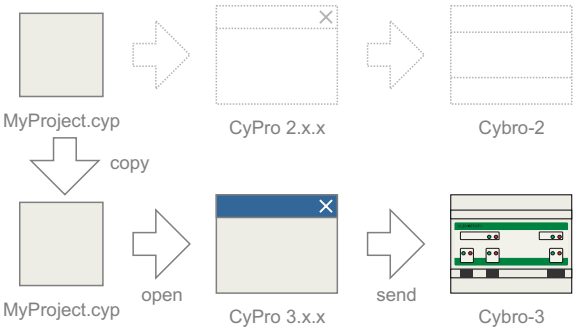
- mixing NPN and PNP input polarity
- zero input on high-speed counter (AB+Z)
- input/output reaction time faster then 5ms
- 4 analog inputs + 1 analog output at the same time

Run/stop switch

Run/stop switch does not exist on Cybro-3, use start (F11) and stop (F12) instead. To set power-on mode, use CyPro "start automatically" option.



Project changes



Variable type

Counter
Not supported any more, must be changed to integer. Increment and decrement can be performed with arithmetic +/- operations.

Word
Not supported any more, must be changed to integer. Bit operations can be performed with logical AND/OR operations.

Timer
Still exists, but the internal implementation is slightly different. When instant on-delay response is needed, write to IN, ET and Q fields directly.

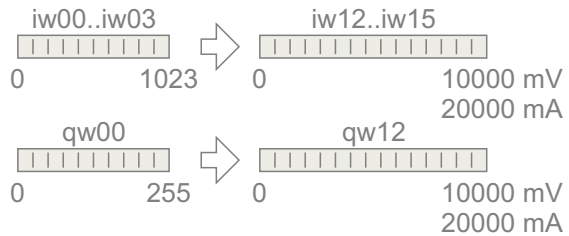
I/O numbering

To get digital and analog terminals consistent, analog numbering is changed:

cybro_iw00 ► cybro_iw12
cybro_iw01 ► cybro_iw13
cybro_iw02 ► cybro_iw14
cybro_iw03 ► cybro_iw15
cybro_qw00 ► cybro_qw12, qw13, qw14 or qw15

I/O range

Measurement is now in physical units:



A simple way to get around this is to manually allocate and recalculate old analog variables:

cybro_iw00:=int((long(cybro_iw12)*1023)/10000);

Similar equation can be used for other inputs and outputs.

Hardware setup

Change CPU unit to Cybro-3, configure universal inputs and outputs. Open IEX modules one by one, set configuration settings. Also check serial port settings in configuration dialog.

System variables

Some system variables does not exist any more, they can be replaced this way:

general_error
Allocate manually and compute as logical OR of all other general errors.

cybro_outputs_off
When active, simply set all outputs to off, near end of the program.

no_input_filter
Digital filtering can not be turned off on Cybro-3.

rtc_read_req
Not needed any more, RTC variables are now updated automatically,

socket_0_id, socket_1_id
This addressing method is obsolete, it can be replaced by adding a new variable to the socket. By checking this variable, program may decide who is the recipient.

ee_write_magic
Negative values are now used to indicate an error, check variable description for details.

System functions

word(expression) ► ulong(expression)
When converting integer to long, word() type was used to imply unsigned. It is now replaced with ulong() function, which does the same thing.

For other conversions and frequently used constructions, check function library.

Timing

If program behaves unexpectedly, a possible explanation is the timing. As new controller is faster, code that rely on execution time may behave unpredictably. To fix that, always write code where the timing strictly relies on system (clock variables, scan time, timer type or RTC).

Connectivity

Native support

Modbus RTU/TCP slave
HW: Cybro-3

Modbus RTU master
HW: Cybro-3 or COM-MB
SW: ModbusRtuMaster.cyp
LIB: CyPro/Examples

HW: Cybro-3
SW: CybroOpcServer
URL: www.cybrotech.com

HW: CyBro-3-ENO or GW-ENO2
SW: EnOceanGateway.cyp
LIB: CyPro/Examples

HW: LC-DC or LD-D8-IQ
SW: DaliDemo.cyp
LIB: CyPro/Examples

HW: COM-DMX
SW: DmxDemo.cyp
LIB: CyPro/Examples

HW: GW-MP
SW: MpBusDemo.cyp
LIB: CyPro/Examples

Using 3rd party adapter

HW: COM-NOK+ZC-GW-485
SW: ZigbeeGateway.cyp
LIB: CyPro/Examples

HW: EIB ASCII Terminal
SW: KnxDemo.cyp
LIB: CyPro/Examples

HW: Techbase Mbus 10
SW: M-BusDemo.cyp
LIB: CyPro/Examples

Using Raspberry Pi

HW: Cybro-Pi4
SW: BACnet protocol stack
URL: bacnet.sourceforge.net

HW: Cybro-Pi4
SW: CybroScgiServer+Mosquitto
URL: www.cybrotech.com
URL: www.mosquitto.org

HW: Cybro-Pi4