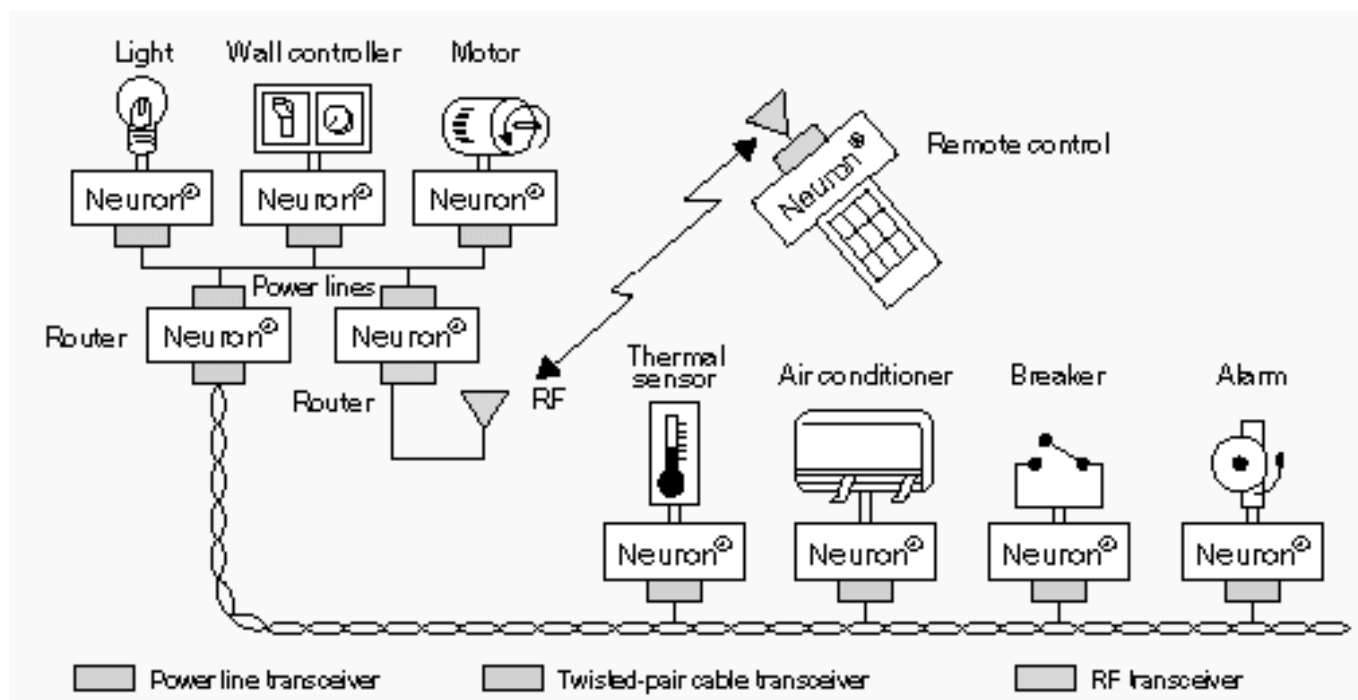
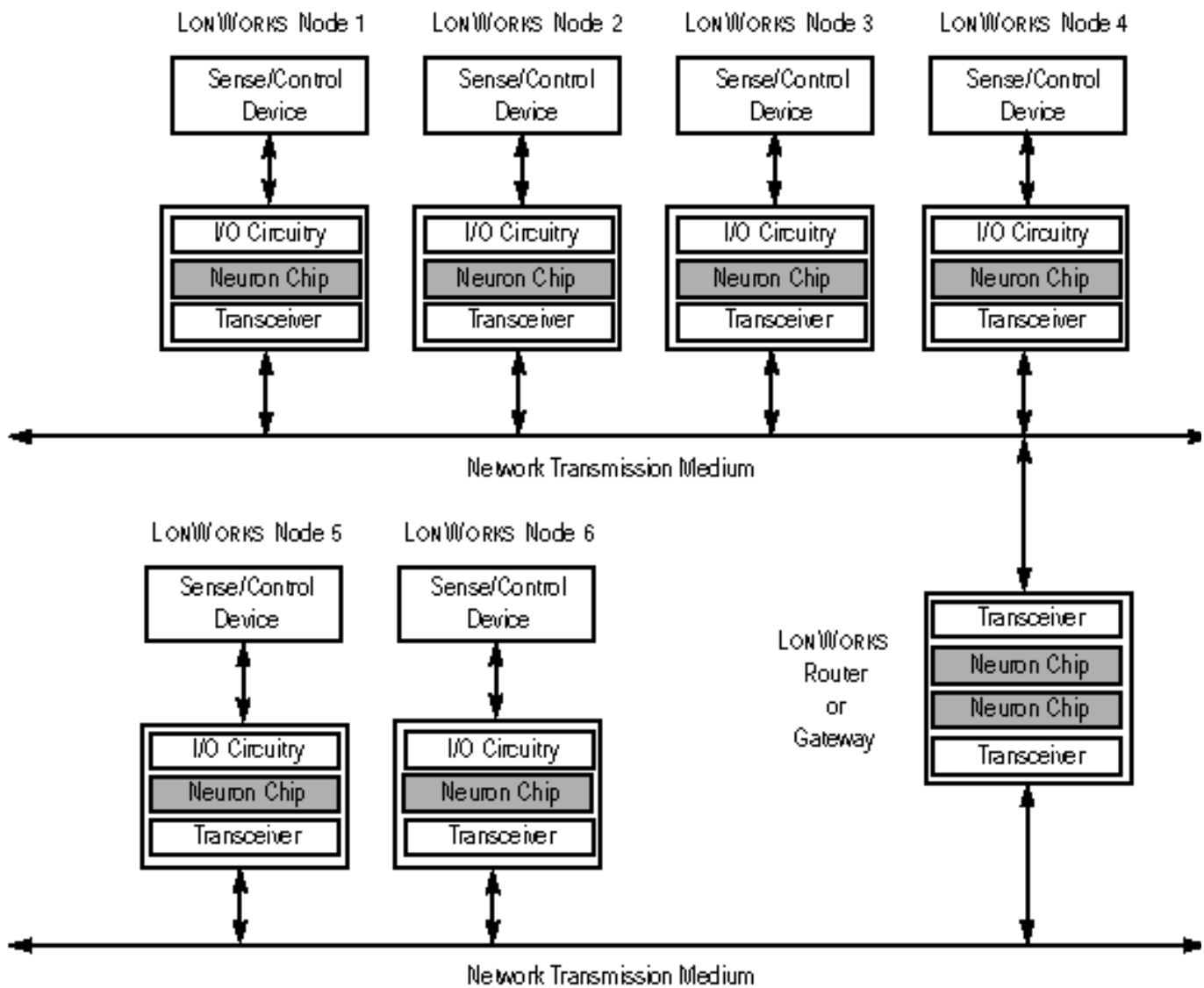


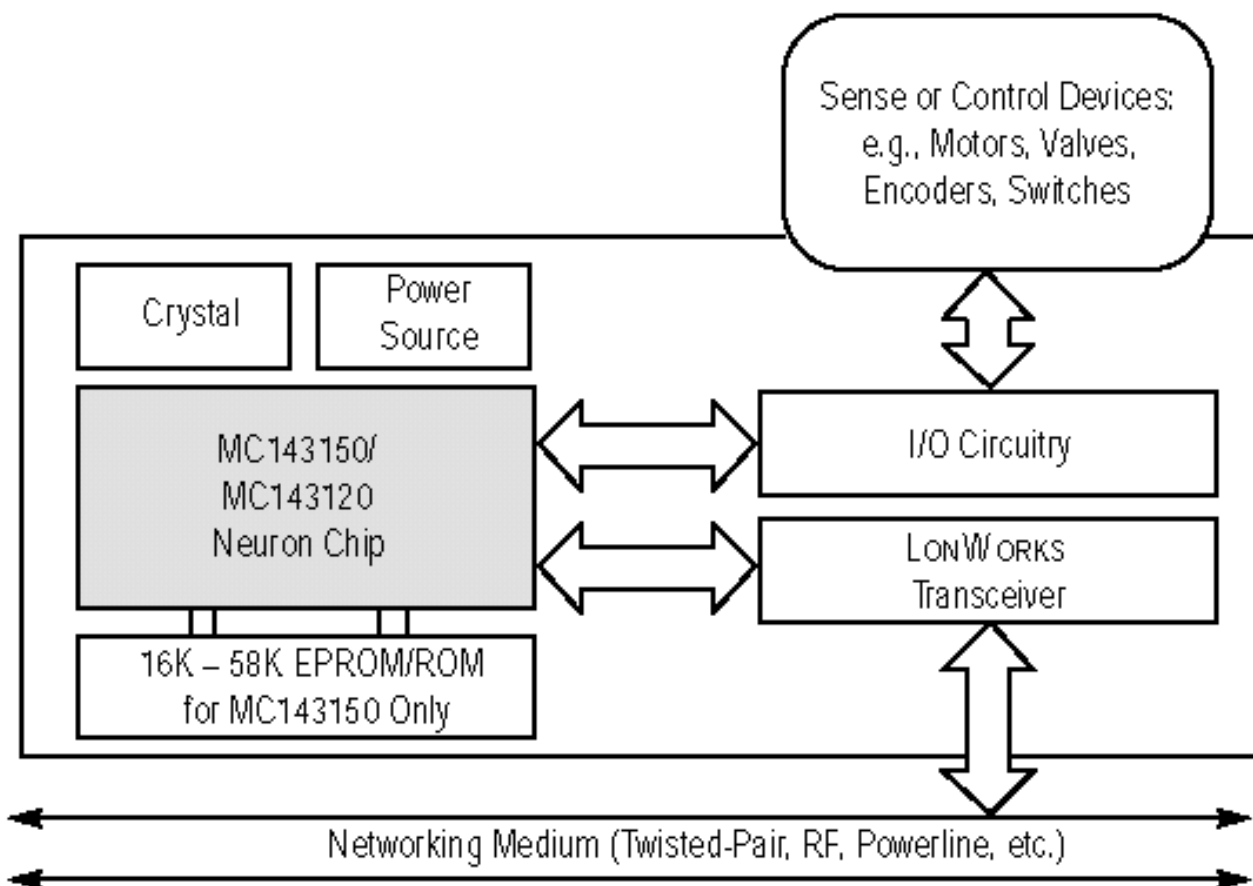
Sieć LON (Local Operating Network) firmy Echelon

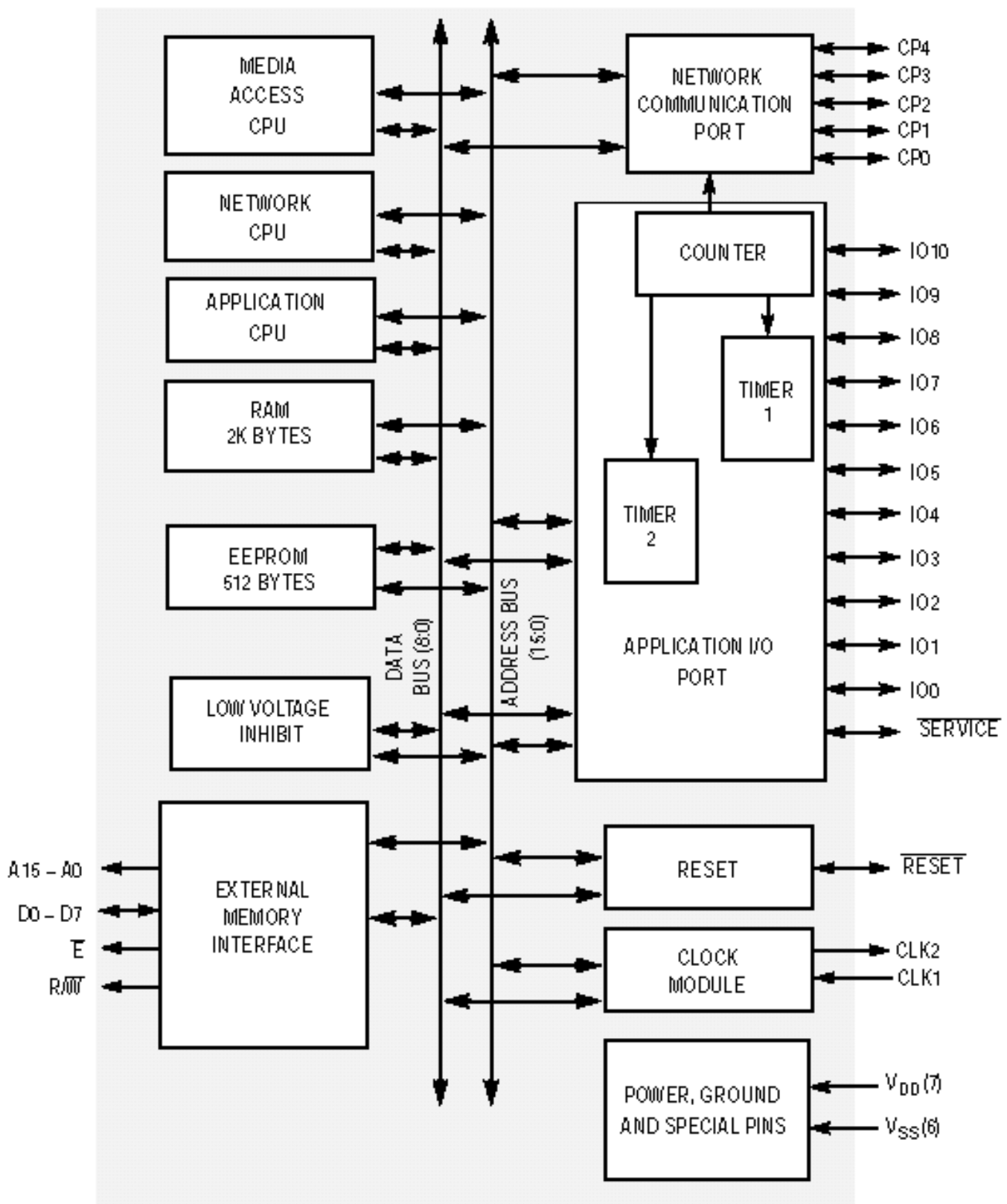


Przykładowa konfiguracja sieci LON

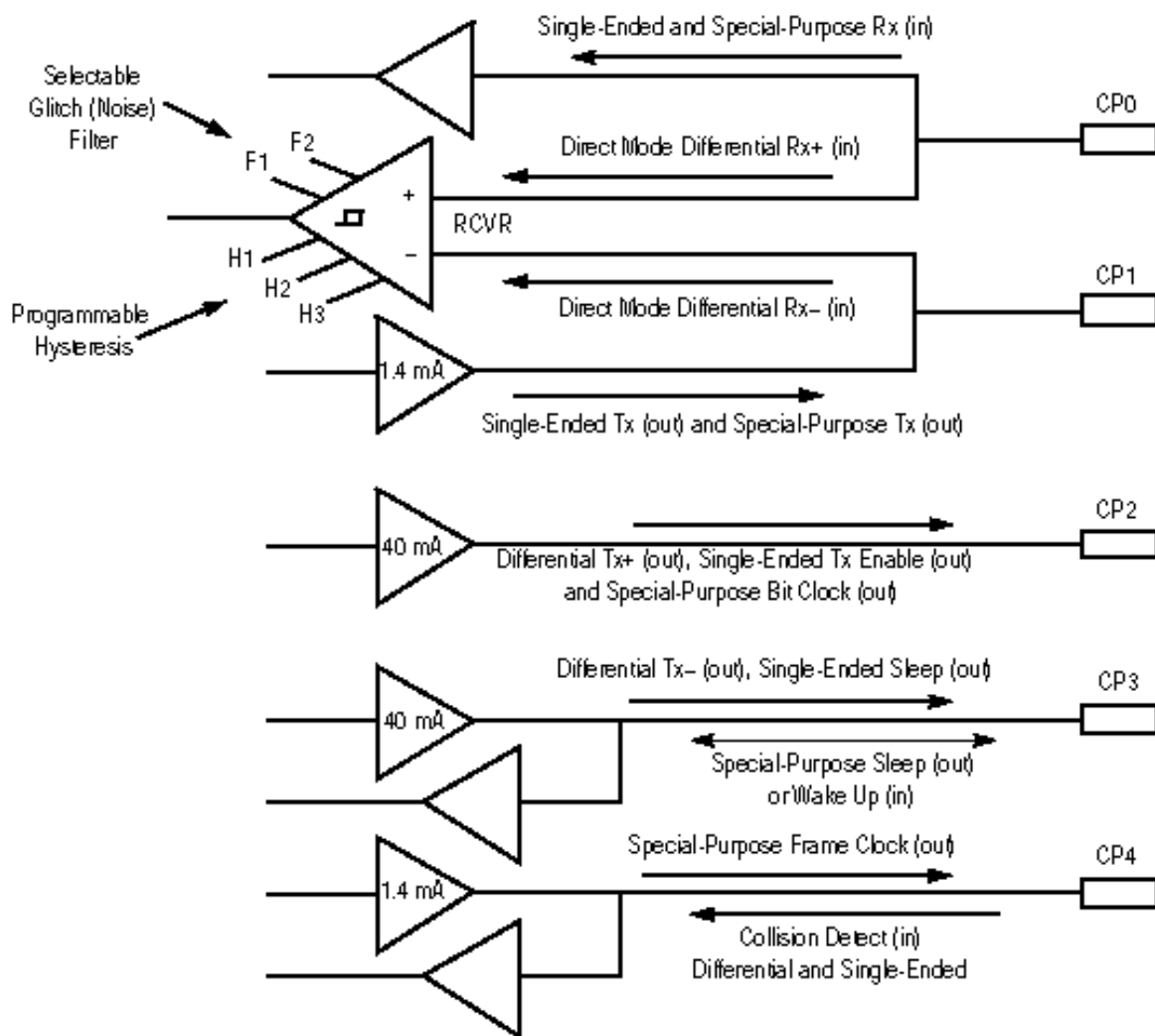


Typowy węzeł sieci LON

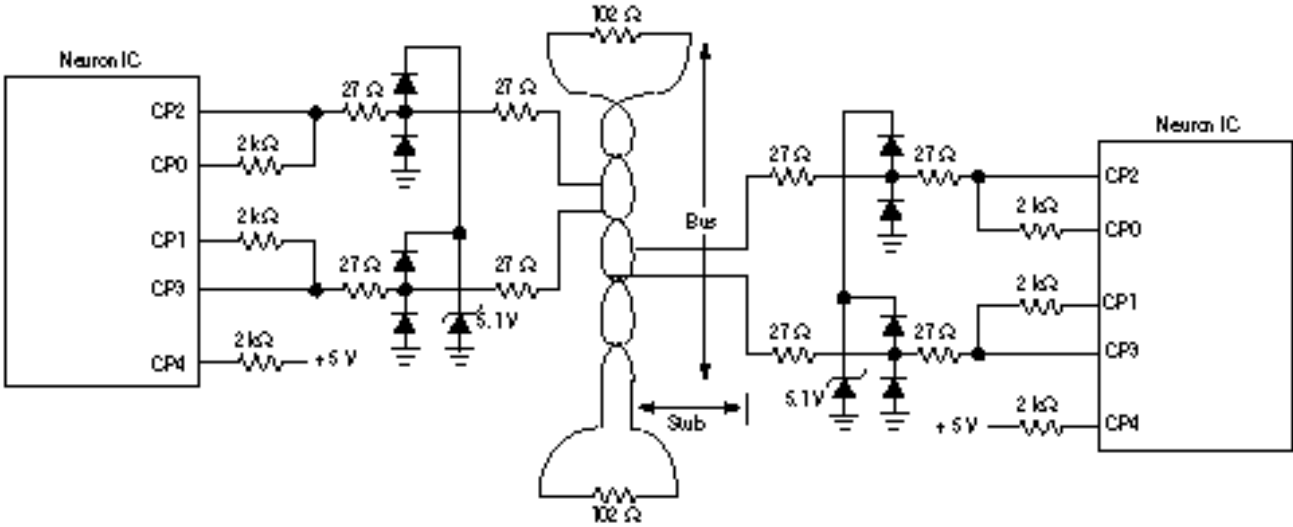




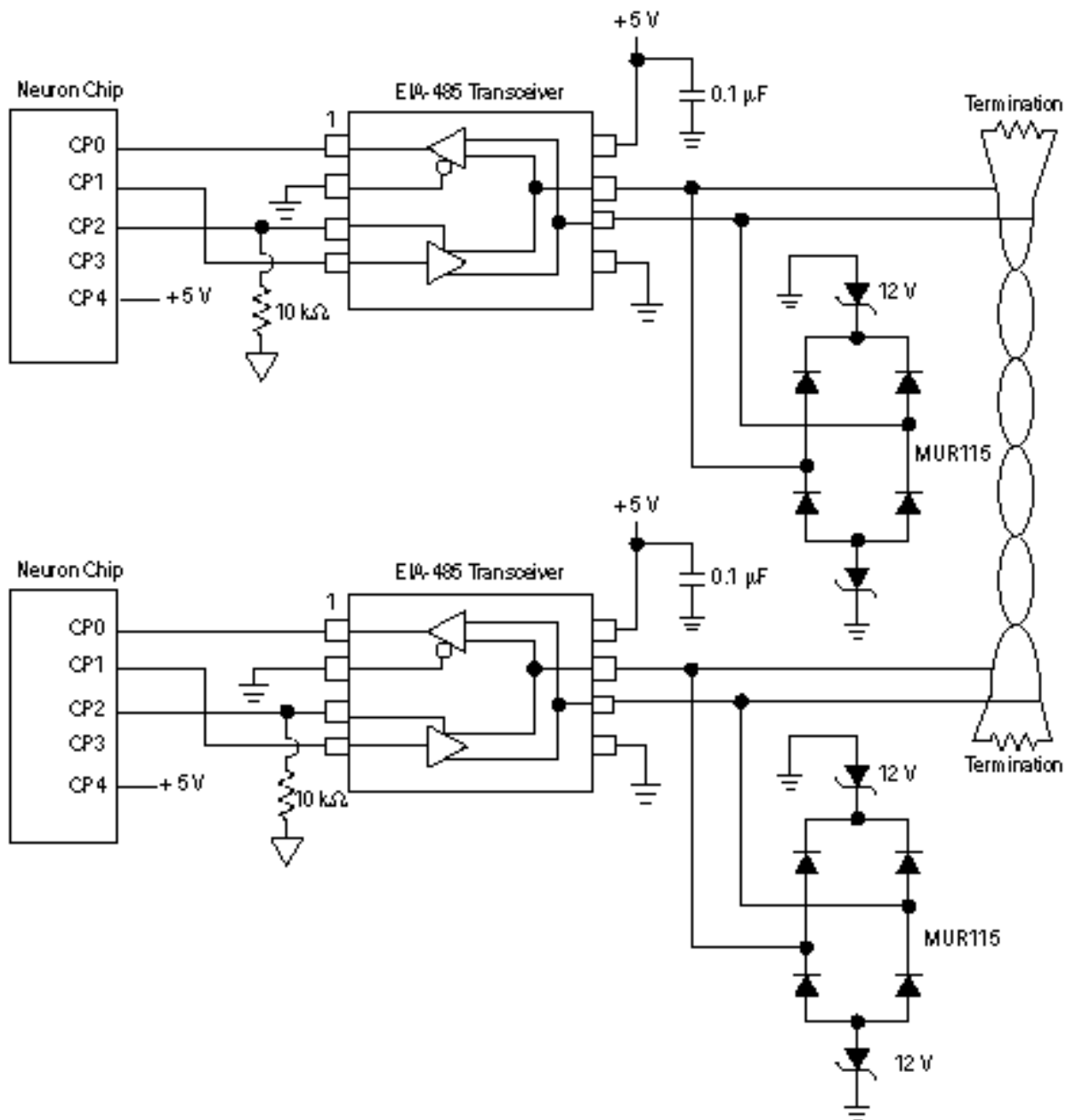
Sygnaly interfejsu sieciowego



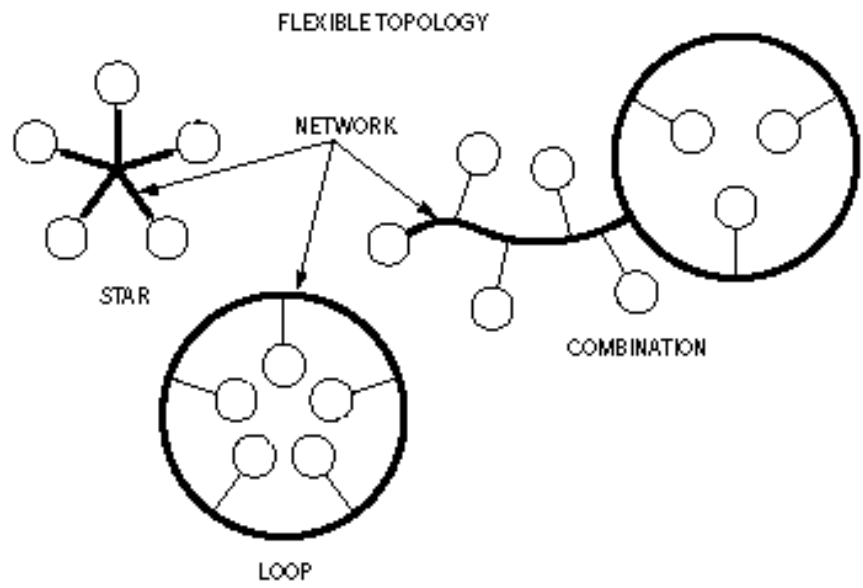
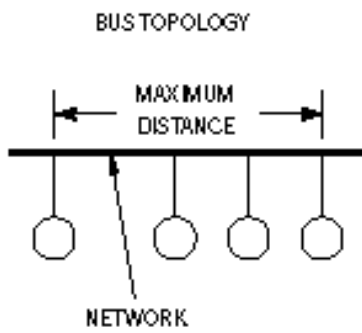
Połączenie skrętka (TP) – bezpośrednio



Połączenie skrętka (RS485) – buforowane

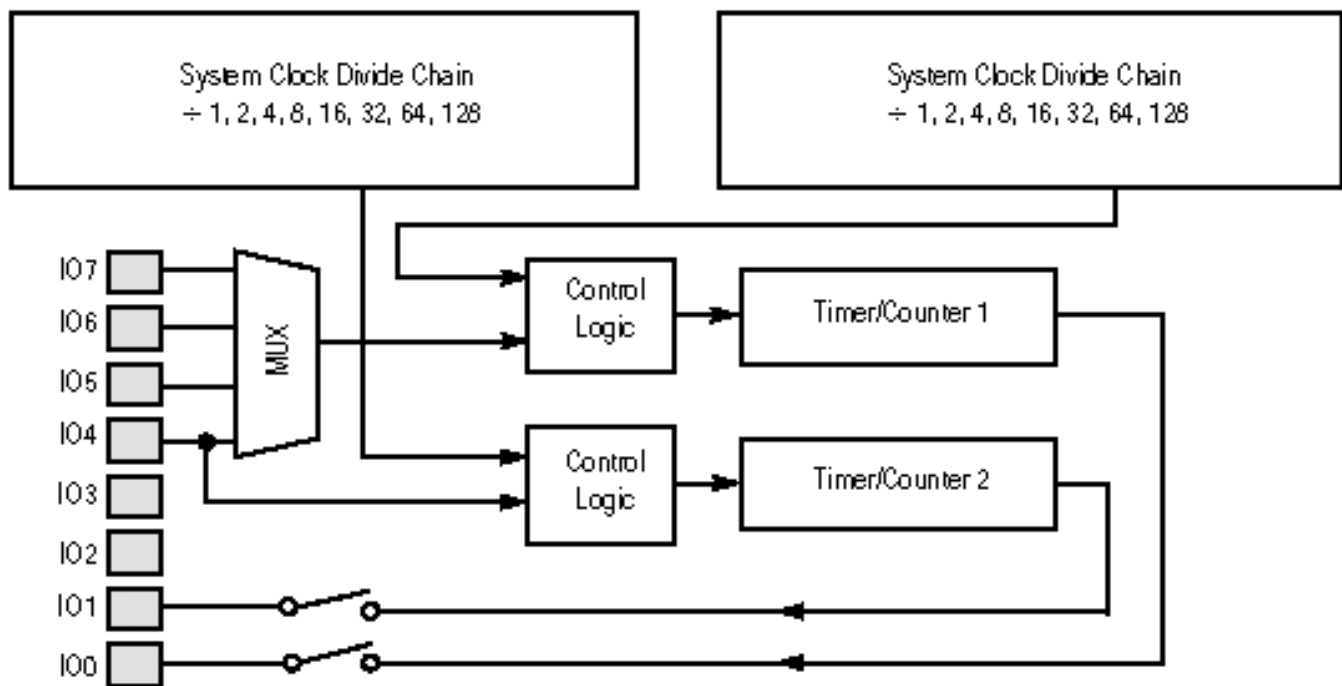


Topologie sieci *LON*

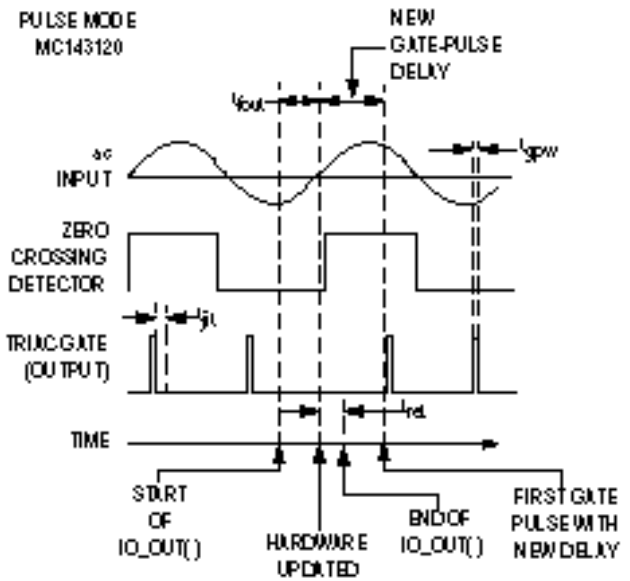
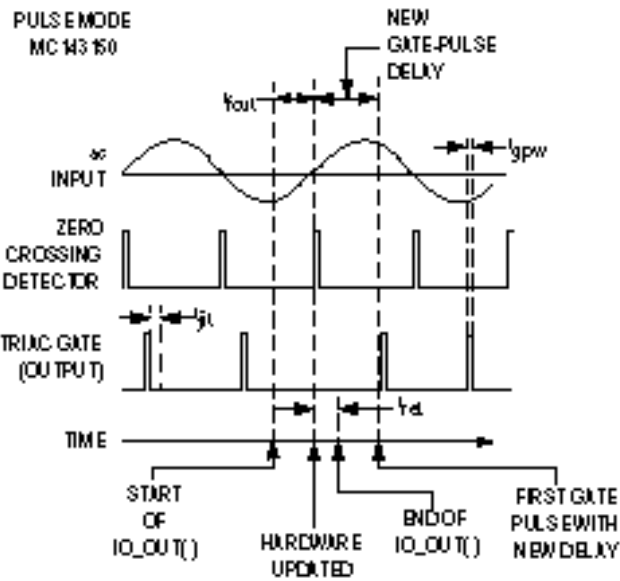
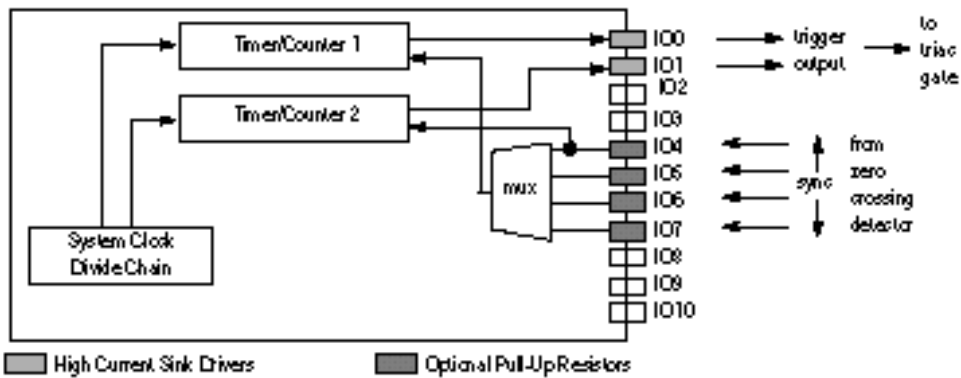


		I/O Pin	0	1	2	3	4	5	6	7	8	9	10
DIRECT I/O MODES	Bit Input, Bit Output												
	Byte Input, Byte Output		All Pins 0 - 7										
	Level Detect Input												
	Nibble Input, Nibble Output		Any Four Adjacent Pins										
PARALLEL I/O MODES	Muxbus I/O		Data Pins 0 - 7							ALS	WS	RS	
	Parallel I/O {	Master/Slave A	Data Pins 0 - 7							CS	R/W	HS	
		Slave B	Data Pins 0 - 7							CS	R/W	AO	
	Bitshift Input, Bitshift Output	C	D	C	D	C	D	C	D	C	D	C	
SERIAL I/O MODES	I ² C I/O										C	D	
	Magcard Input		Optional Timeout							C	D		
	Magtrack1 Input		Optional Timeout							C	D		
	Neurowire I/O {	Master	Optional Chip Select							C	D	D	
		Slave	Optional Timeout							C	D	D	
	Serial Input												
	Serial Output												
Wiegand Input		Any Two Pins (Optional Timeout)											
TIMER/COUNTER INPUT MODES	Dualslope Input	Control											
	Edge/og Input												
	Infrared Input												
	On-time Input												
	Period Input												
	Pulsecount Input												
	Quadrature Input					4 + 5	6 + 7						
	Totalcount Input												
	TIMER/COUNTER OUTPUT MODES	Edgedivide Output					Sync Input						
Frequency Output													
Oneshot Output													
Pulsecount Output													
Pulsewidth Output													
Triac Output		Control				Sync Input							
Triggeredcount Output		Control				Sync Input							
		0	1	2	3	4	5	6	7	8	9	10	

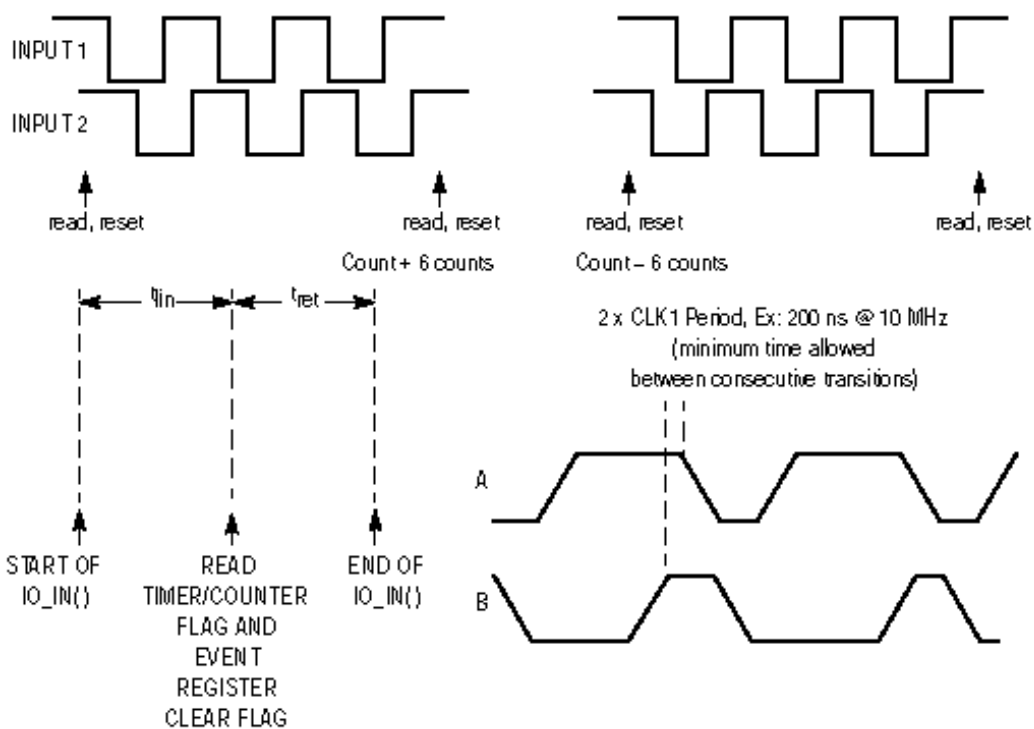
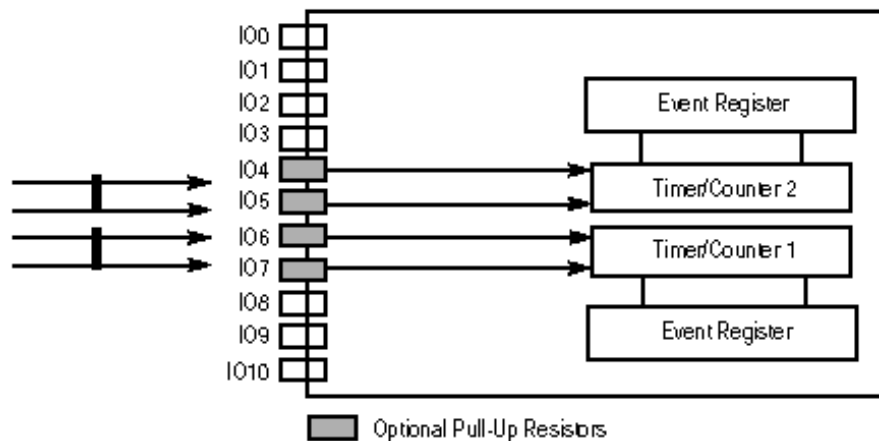
Struktura układu czasowego (*timer*)

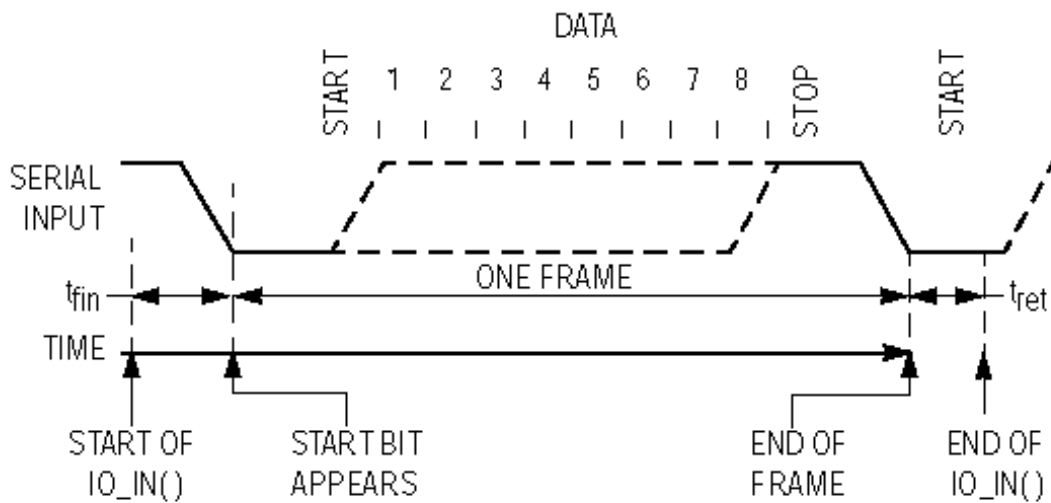
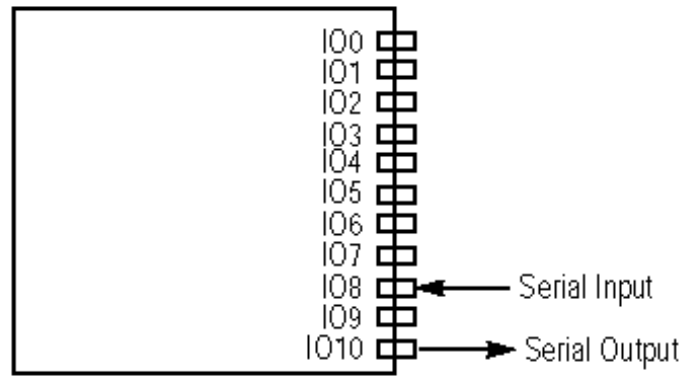
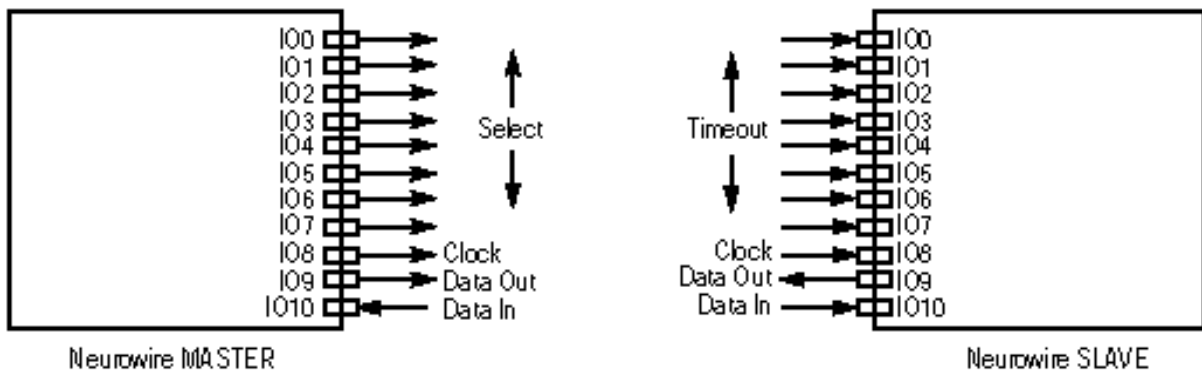


Przykład: sterowanie triakiem

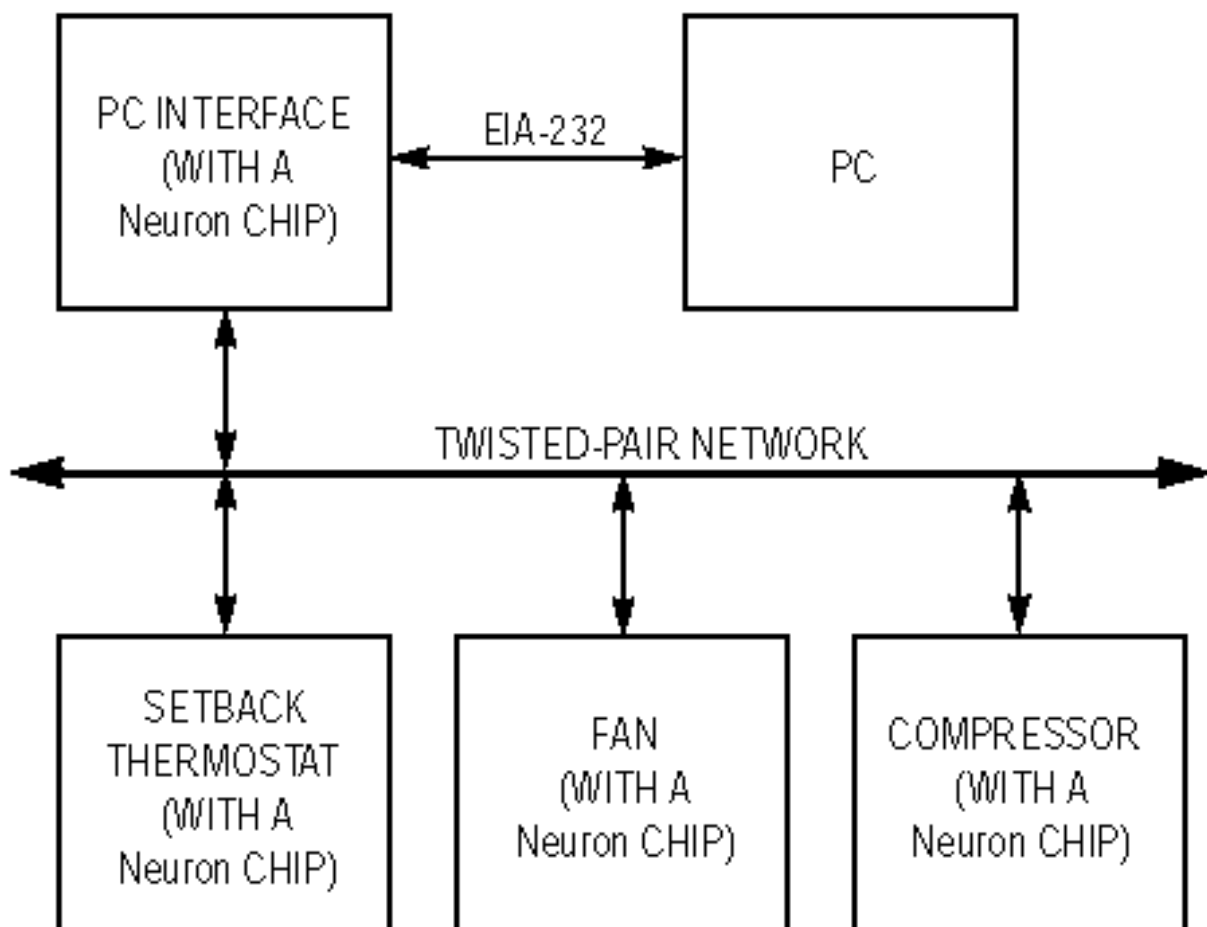


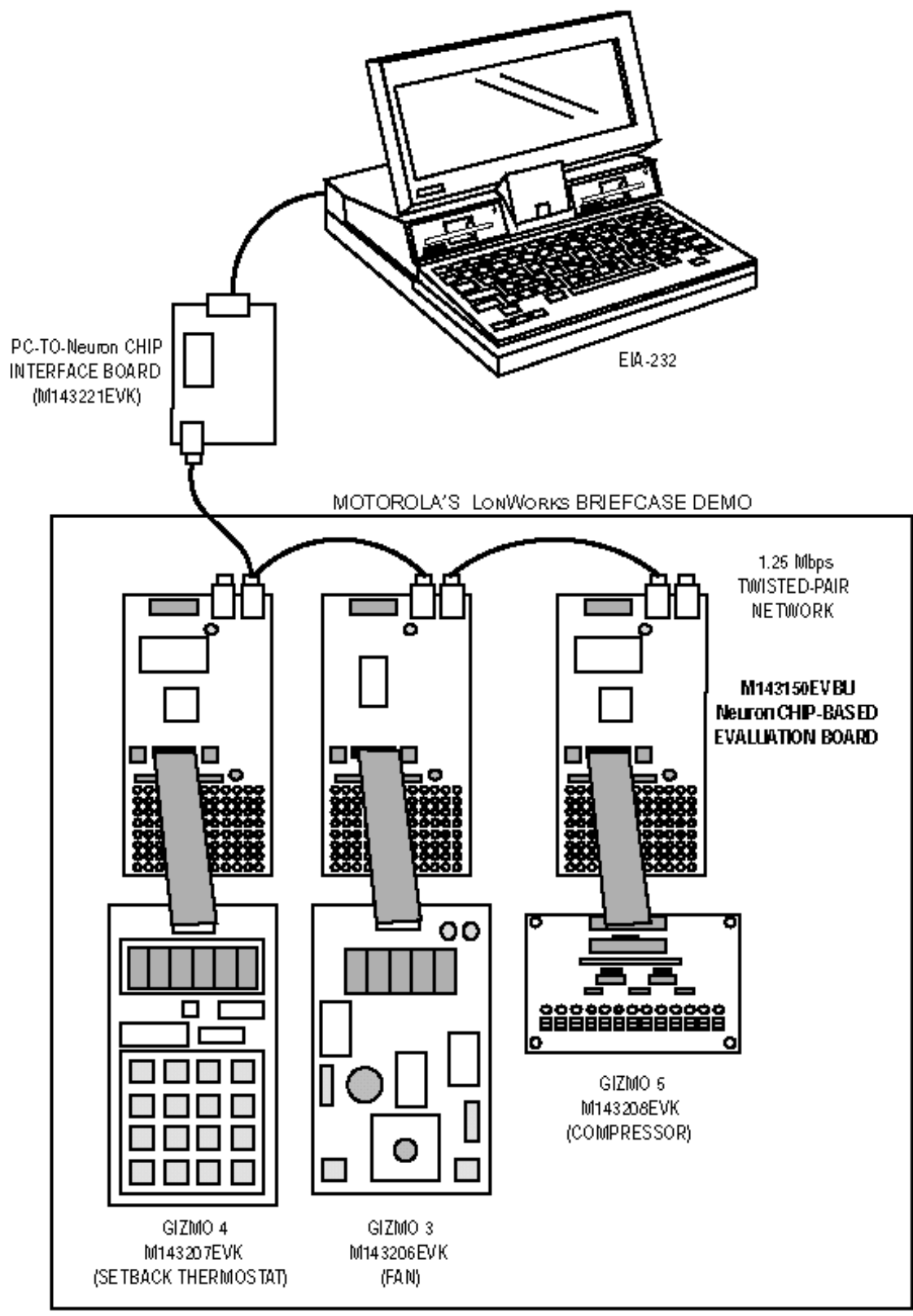
Obsługa kodera kwadraturowego (QDEC)



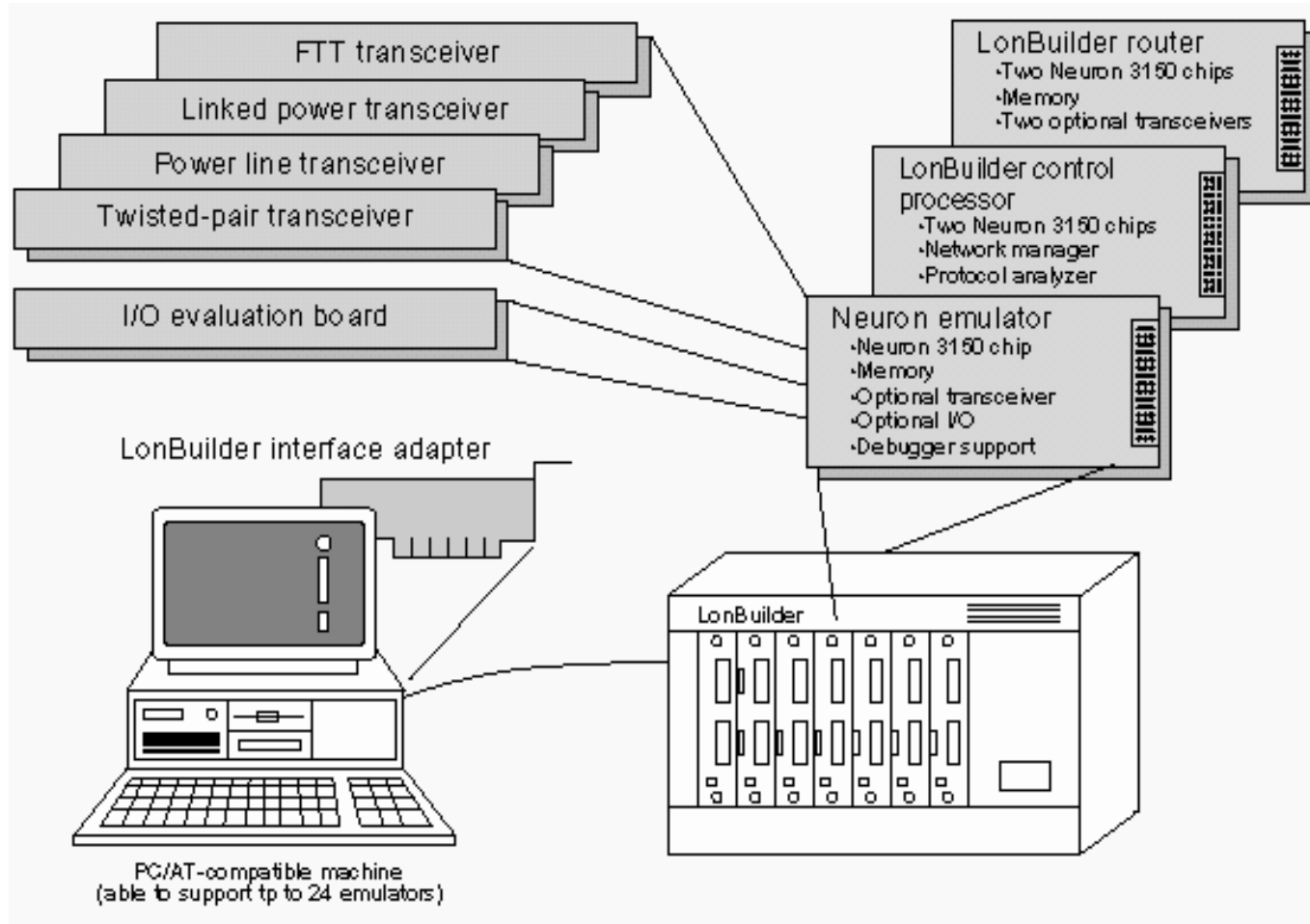


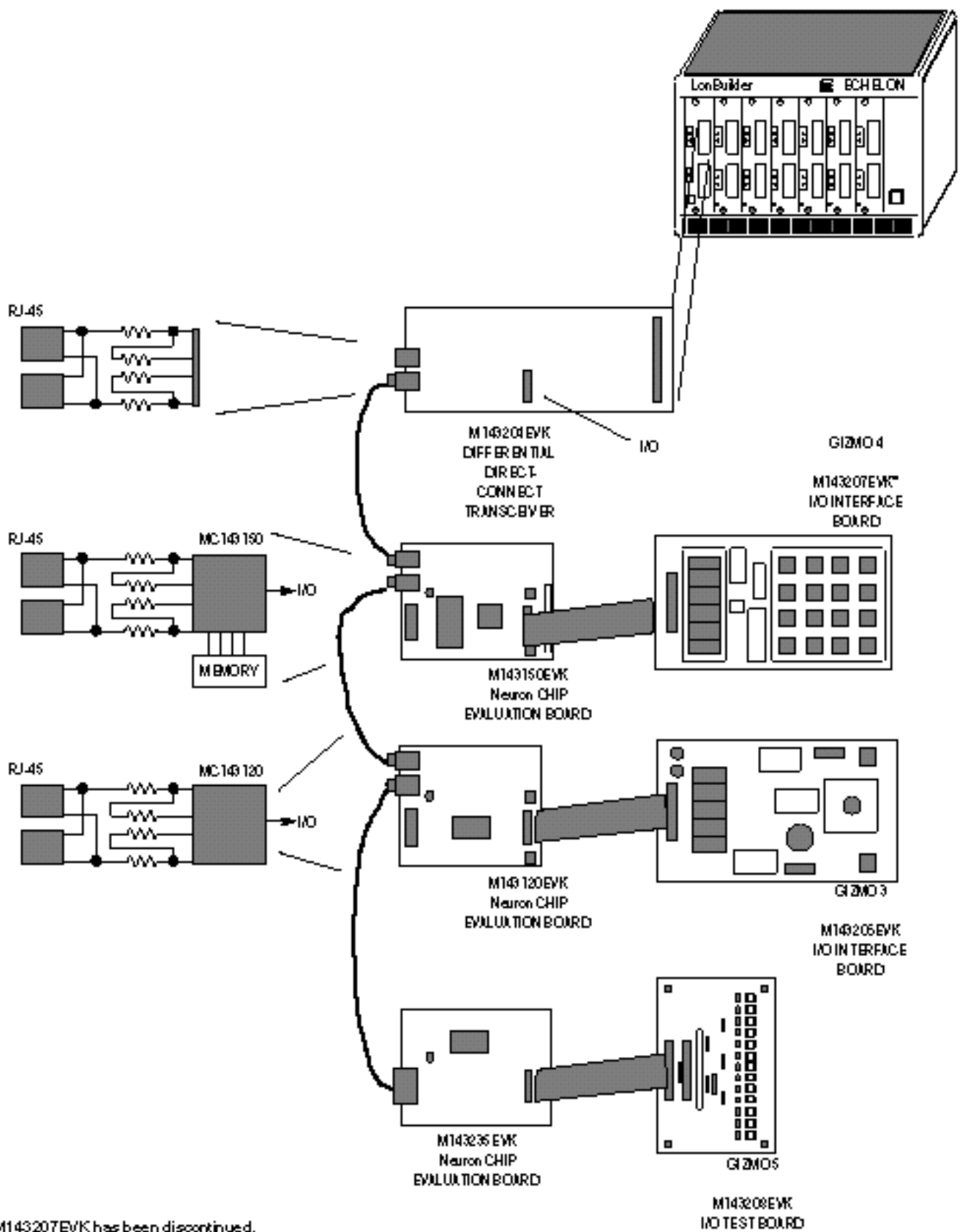
Przykład: klimatyzacja (HVAC)





System uruchomieniowy *LonBuilder* firmy Echelon





*M143207EVK has been discontinued.

MW-ZPCiR-ICT-PW/r

Warstwy sieciowe protokołu *LonTalk*

OSI Layer	Purpose	Services Provided
7 Application	Application Compatibility	Network Variables; Type standardization and Identification; Generic Message Functions
6 Presentation	Data Interpretation	Foreign Frame Transmission
5 Session	Remote Actions	Request–Response Protocol
4 Transport	End-To-End Reliability	Acknowledged/Unacknowledged; Unicast/Multicast; Authentication; Common Ordering; Duplicate Detection
3 Network	Destination Addressing	Connection-Less, Domain-Wide Broadcast; Configured/Learning Routers
2 Link	Media Access and Framing	Framing; Data Encoding; CRC Error Checking; Predictive CSMA; Collision Avoidance; Optional Priority/Collision Detection
1 Physical	Electrical Interconnect	Multi-Media, Media-Specific Protocols (TP, PL, RF, IR, CX, ...)

Standardowe zmienne sieciowe *NeuronC*

Name	Measurement	Unit	Range	Resolution	SNVT#
SNVT_length	Length	m	0–6553.5	0.1m	17
SNVT_length_kilo	Length	km	0–6553.5	0.1km	18
SNVT_temp	Temperature	° C	-274–6279.5	0.1° C	39
SNVT_volt	Voltage	V	-3276.8–+3276.7	0.1V	44
SNVT_count	Count, event	counts	0–65535	1	8
SNVT_flow	Flow	l/s	0–65535	1 l/s	15
SNVT_elec_kwh	Energy, elec.	kWh	0–65535	1 kWh	13
SNVT_lev_cont	Level, continuous	%	0–100	0.5 %	21
SNVT_lev_disc	Level, discrete	OFF LOW MED HIGH ON	0 1 2 3 4		22
SNVT_telcom	Phone state	Null (U0) Call (U1) Overlap (U2) Out Call (U3) Callback (U4)	0 1 0 3 4		38

Przykład **SWITCH–LAMP** (Echelon)

SWITCH.NC - Używa przycisku do przełączania zmiennej sieciowej (NV)

- Wejście HW - Przycisk na wejściu IO_4 Neuron Chip
- Wyjście NV - `nv_switch_state`, przełączana przez przycisk

LAMP.NC - Używa LED do sygnalizacji stanu NV

- Wejście NV - `nv_lamp_state`, wejście wyświetlane na LED
- Wyjście HW - LED na wyjściu IO_0 Neuron Chip

Opis połączeń zmiennych (binding)

Bind: `nv_switch_state` w węźle pracującym jako SWITCH

to: `nv_lamp_state` w węźle pracującym jako LAMP.

Program węzła SWITCH

```
// SWITCH.NC -- Sample switch sensor program
// Copyright (c) 1995 Echelon Corporation.
// All Rights Reserved.
//
// Date last modified: 1-Dec-94

////////////////////// Compiler Pragmas ////////////////////////

#pragma enable_io_pullups

////////////////////// Include Files ////////////////////////

#include <snvt_lev.h>

////////////////////// Network Variables ////////////////////////

network output SNVT_lev_disc nv_switch_state = ST_OFF;

////////////////////// Constants ////////////////////////

#define BUTTON_DOWN 1
#define BUTTON_UP 0

////////////////////// I/O Objects ////////////////////////

IO_4 input bit ioButton = BUTTON_UP;

////////////////////// Tasks ////////////////////////

// I/O task -- handle pushbutton down event

when (io_changes(ioButton) to BUTTON_DOWN) { //button pressed
    nv_switch_state = (nv_switch_state!=ST_OFF)?ST_OFF:ST_ON;
}                                           // toggle state
```

Program węzła *LAMP*

```
// LAMP.NC -- Sample lamp actuator program
// Copyright (c) 1995 Echelon Corporation.
// All Rights Reserved.
//
// Date last modified: 1-Dec-94

////////////////////////////////// Include Files //////////////////////////////////

#include <snvt_lev.h>

////////////////////////////////// Network Variables //////////////////////////////////

network input SNVT_lev_disc nv_lamp_state = ST_OFF;

////////////////////////////////// Constants //////////////////////////////////

#define LED_ON      1
#define LED_OFF    0

////////////////////////////////// I/O Objects //////////////////////////////////

IO_0 output bit ioLED = LED_OFF;

////////////////////////////////// Tasks //////////////////////////////////

// NV update task -- handle update to lamp state
when (nv_update_occurs(nv_lamp_state)) {
    io_out(ioLED, (nv_lamp_state != ST_OFF)?LED_ON:LED_OFF);
}
```