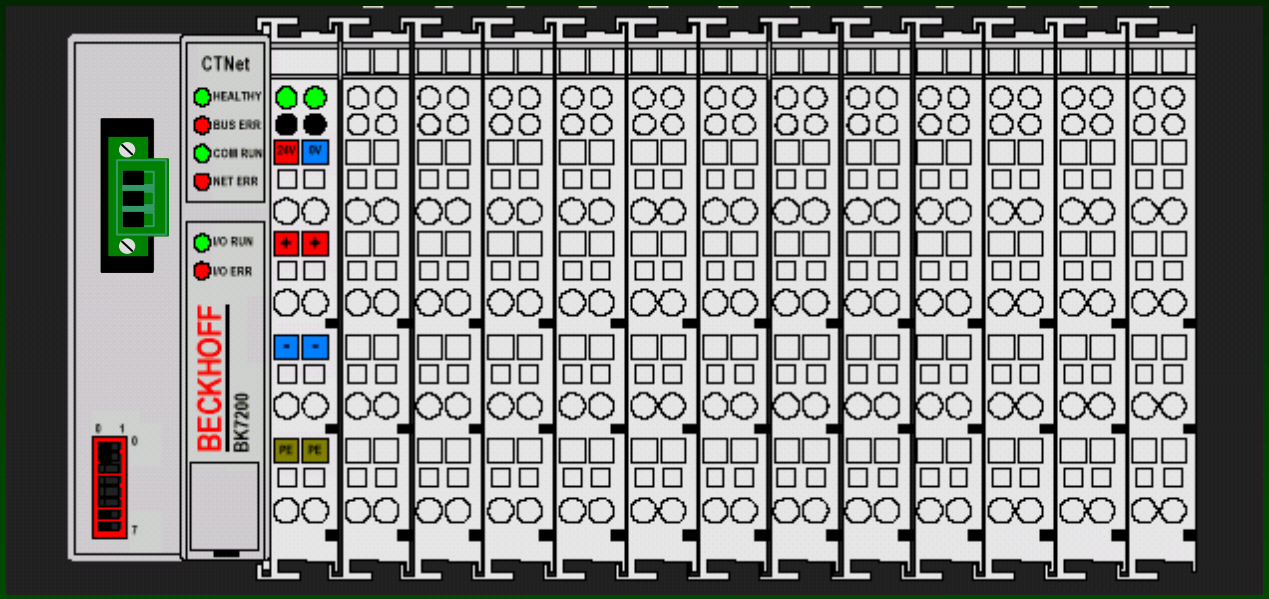
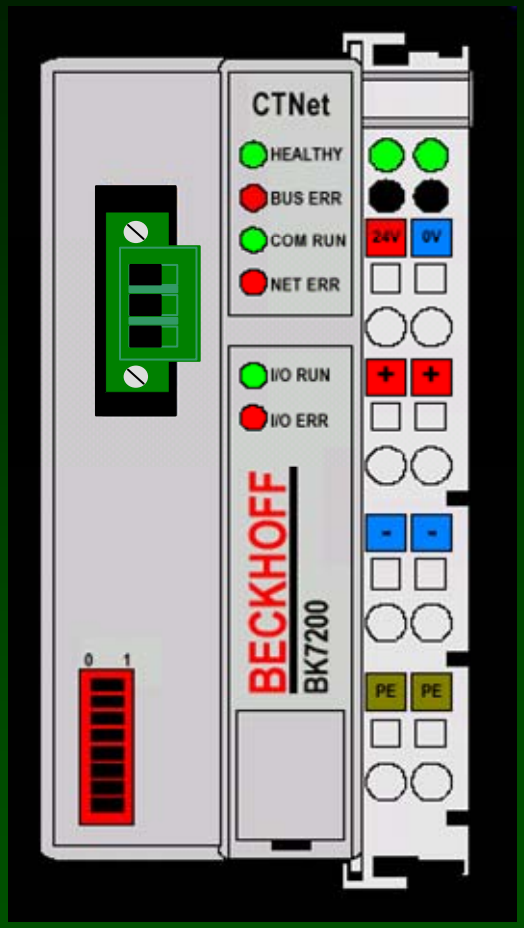
CTNet Remote I/O



Custom, Plug-in I/O Solution from Control Techniques and Beckhoff

Industrial Quality
Easy to Use
CTNet Rev D Compliant

CTNet Remote I/O

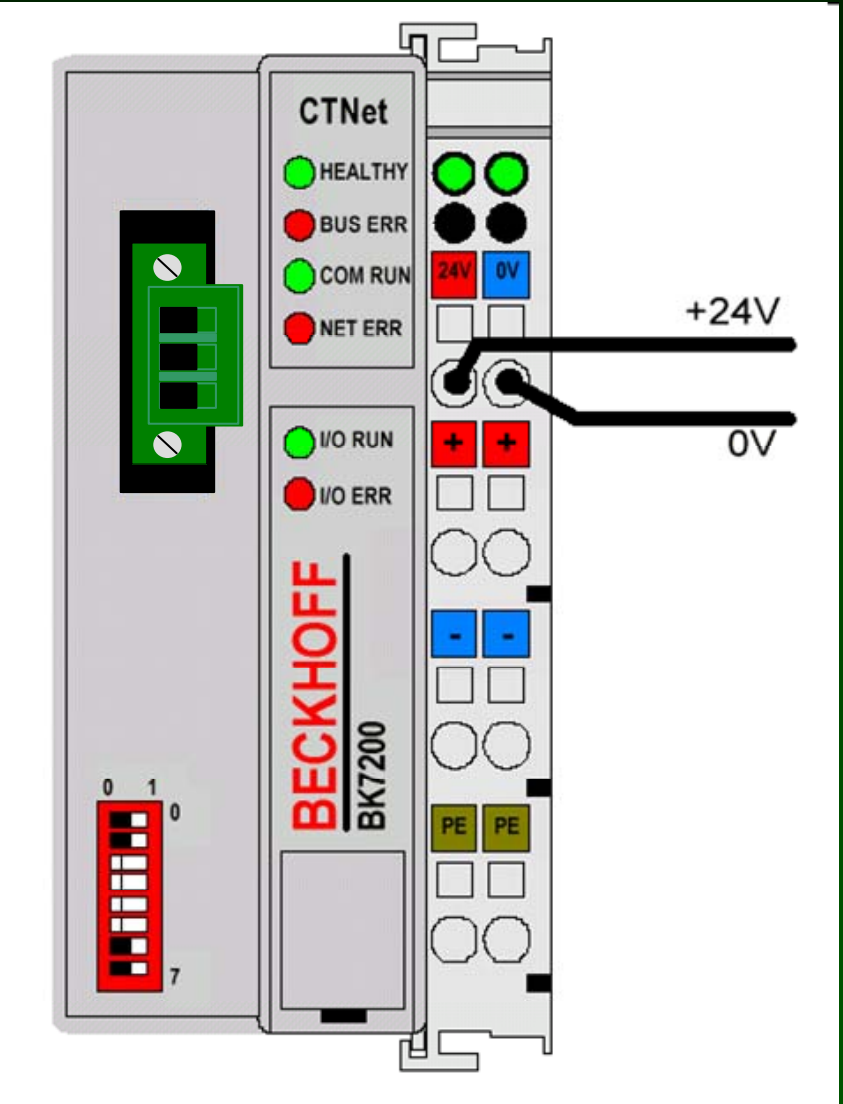


CTNet Bus Coupler

- Designed jointly by CT and Beckhoff
- Communications via CTNet Fieldbus
- Accepts most Beckhoff I/O Modules
- Requires SyPT Pro and one or more SM-Apps or SM-Apps + modules to use
- Only available from CT

CTNet Remote I/O

Electronics powered by regulated +24 volt DC supply

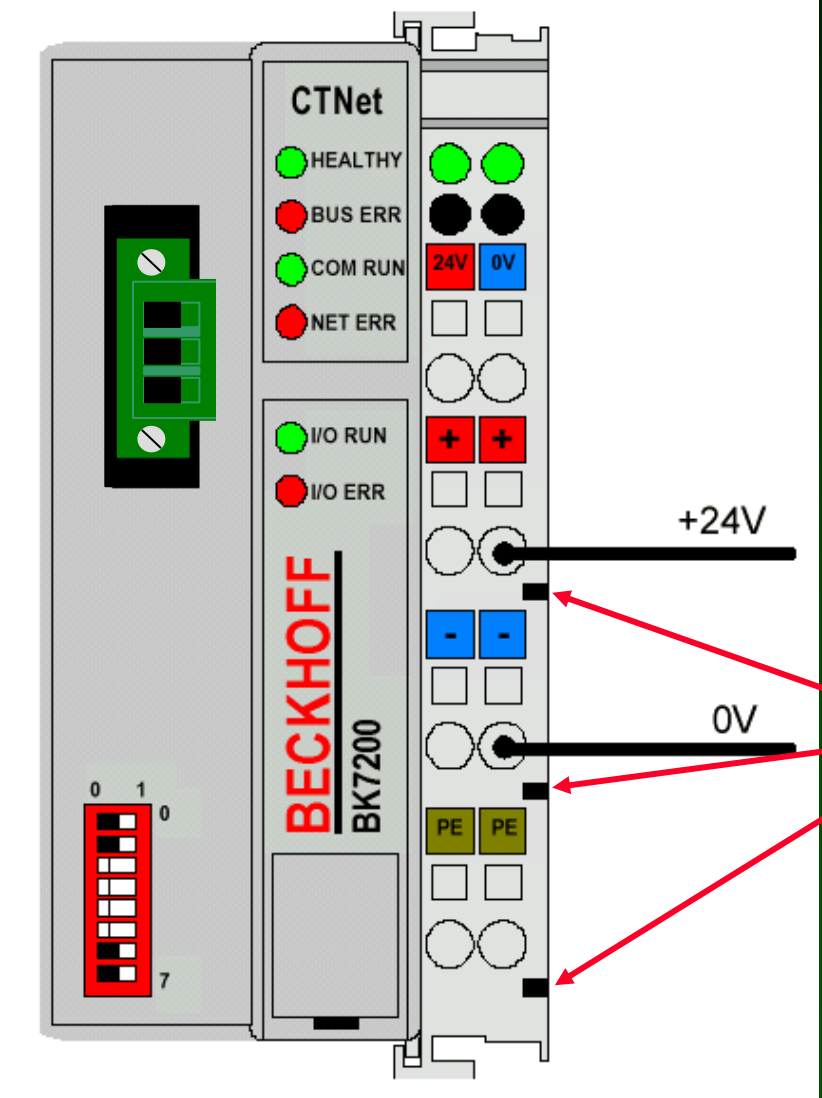


This powers the electronics of the entire system.

The six spring-loaded fingers on the side buss the power and the local K-buss communications to all other modules.

CTNet Remote I/O

Separate 24 volt supply powers the contacts



These hidden knife-edge contacts buss the 24 volts to all other I/O modules.

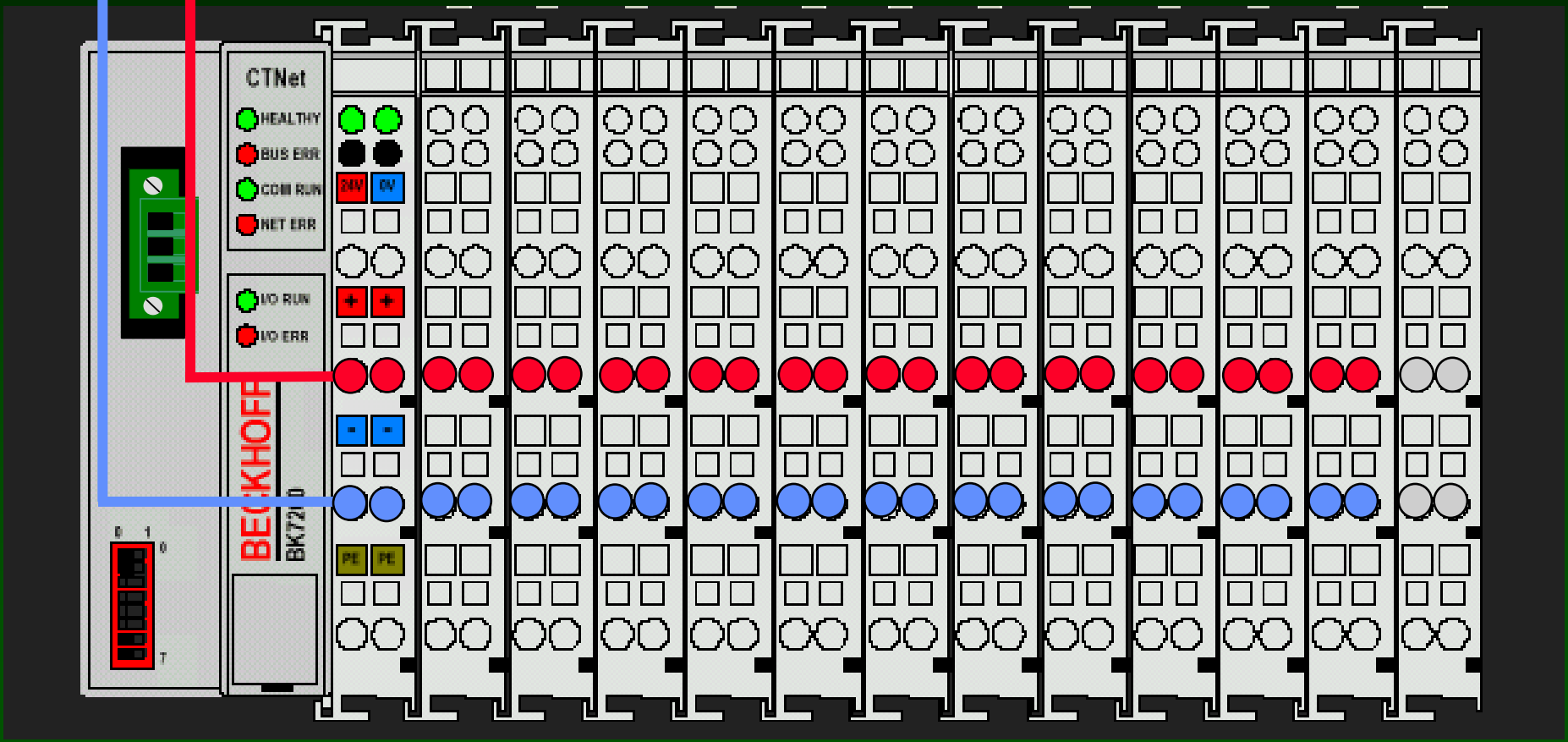
These are used with the digital inputs/outputs.

CTNet Remote I/O

Powering the contacts on the CTNet Bus Terminal automatically busses contact power down the line.

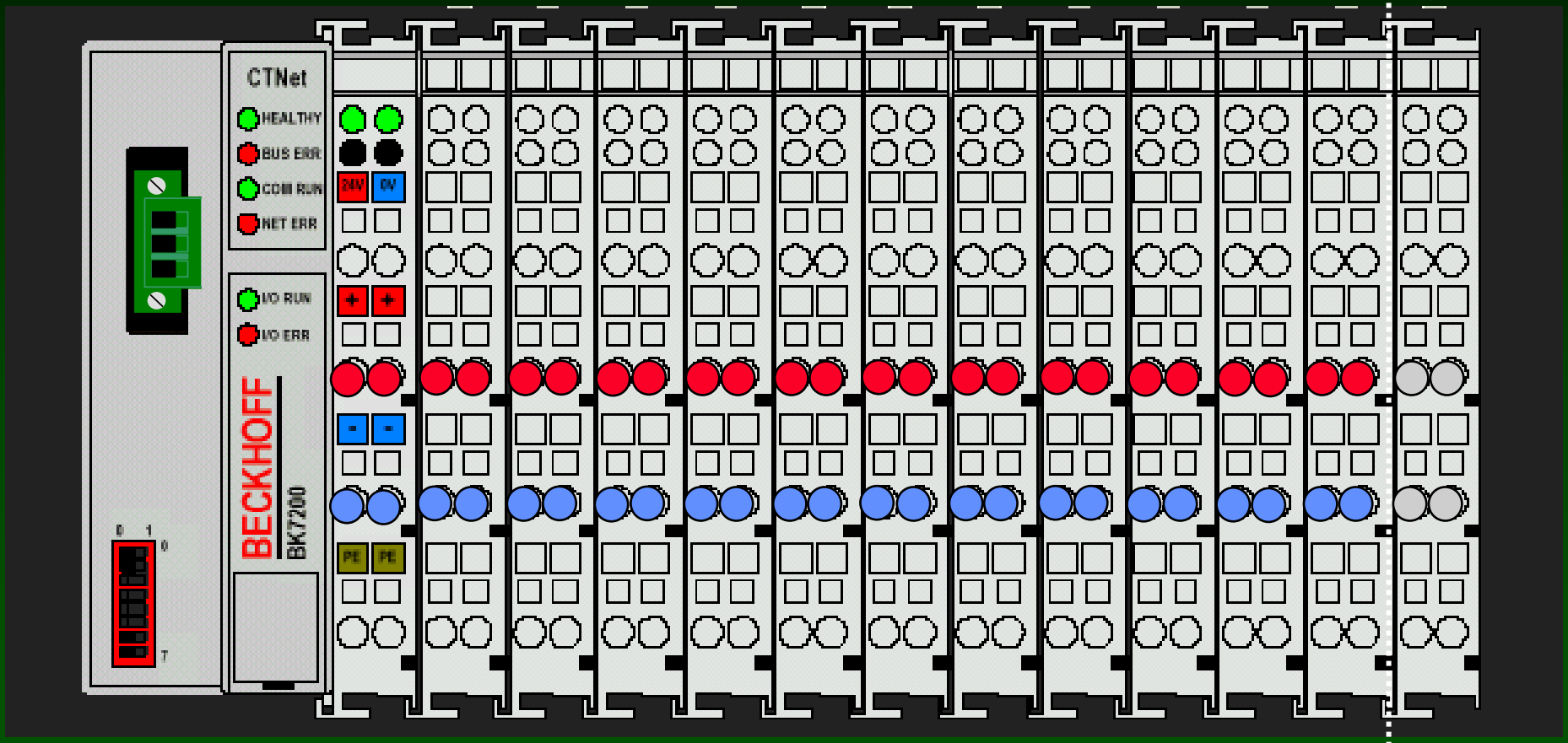
Gnd **+24**

RED = +24 volts **BLUE = ground**



CTNet Remote I/O

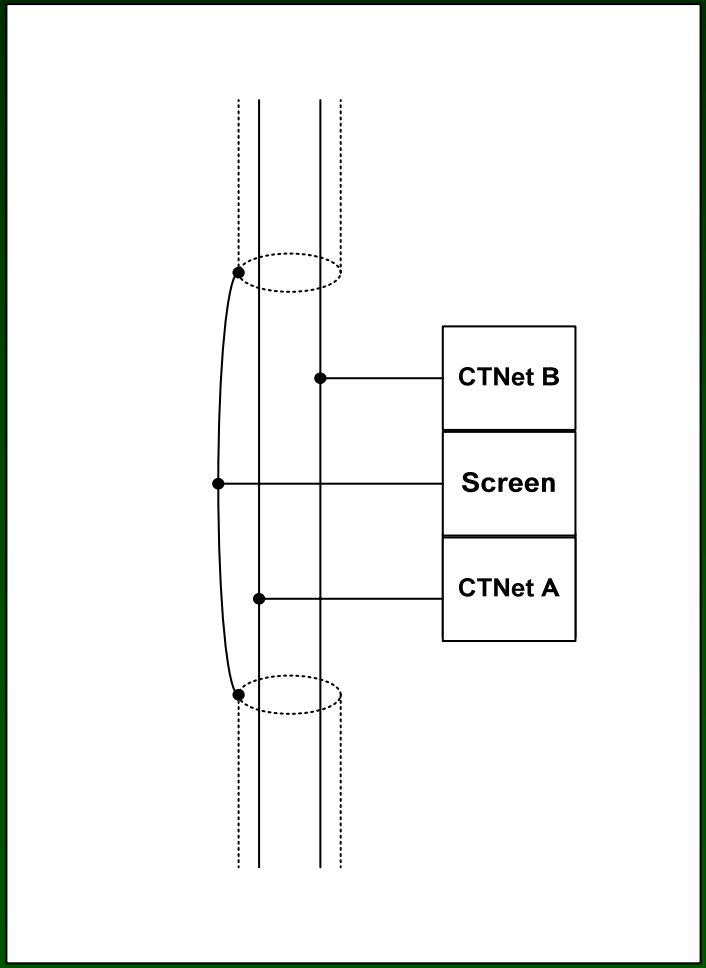
The last module must be a KL9010 Beckhoff End Terminal



CTNet Remote I/O

Bus Coupler includes a 3-pin CTNet Rev D Connector

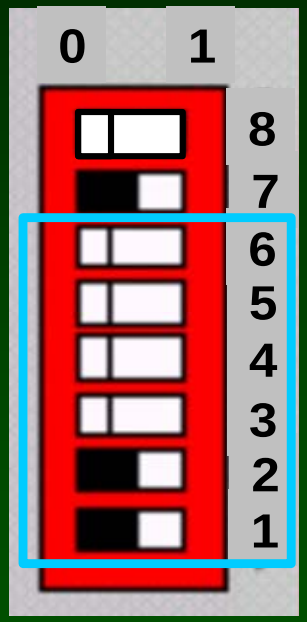
The CTNet connector is wired as shown:



CTNet Remote I/O

DIP-switch selects any CTNet address from 1 to 64

The address you select must be unique.



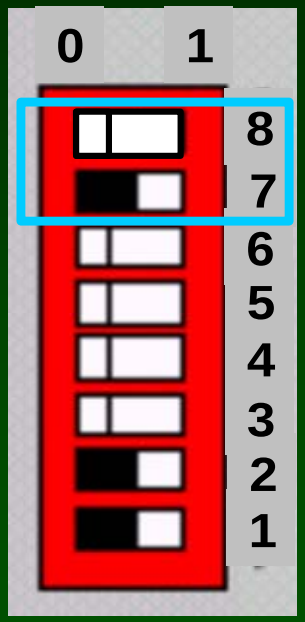
This setting selects CTNet address 4

Node	DIP Switch						Node	DIP Switch						Node	DIP Switch					
Address	6	5	4	3	2	1	Address	6	5	4	3	2	1	Address	6	5	4	3	2	1
1	0	0	0	0	0	0	23	0	1	0	1	1	0	45	1	0	1	1	0	0
2	0	0	0	0	0	1	24	0	1	0	1	1	1	46	1	0	1	1	0	1
3	0	0	0	0	1	0	25	0	1	1	0	0	0	47	1	0	1	1	1	0
4	0	0	0	0	1	1	26	0	1	1	0	0	1	48	1	0	1	1	1	1
5	0	0	0	1	0	0	27	0	1	1	0	1	0	49	1	1	0	0	0	0
6	0	0	0	1	0	1	28	0	1	1	0	1	1	50	1	1	0	0	0	1
7	0	0	0	1	1	0	29	0	1	1	1	0	0	51	1	1	0	0	1	0
8	0	0	0	1	1	1	30	0	1	1	1	0	1	52	1	1	0	0	1	1
9	0	0	1	0	0	0	31	0	1	1	1	1	0	53	1	1	0	1	0	0
10	0	0	1	0	0	1	32	0	1	1	1	1	1	54	1	1	0	1	0	1
11	0	0	1	0	1	0	33	1	0	0	0	0	0	55	1	1	0	1	1	0
12	0	0	1	0	1	1	34	1	0	0	0	0	1	56	1	1	0	1	1	1
13	0	0	1	1	0	0	35	1	0	0	0	1	0	57	1	1	1	0	0	0
14	0	0	1	1	0	1	36	1	0	0	0	1	1	58	1	1	1	0	0	1
15	0	0	1	1	1	0	37	1	0	0	1	0	0	59	1	1	1	0	1	0
16	0	0	1	1	1	1	38	1	0	0	1	0	1	60	1	1	1	0	1	1
17	0	1	0	0	0	0	39	1	0	0	1	1	0	61	1	1	1	1	0	0
18	0	1	0	0	0	1	40	1	0	0	1	1	1	62	1	1	1	1	0	1
19	0	1	0	0	1	0	41	1	0	1	0	0	0	63	1	1	1	1	1	0
20	0	1	0	0	1	1	42	1	0	1	0	0	1	64	1	1	1	1	1	1
21	0	1	0	1	0	0	43	1	0	1	0	1	0							
22	0	1	0	1	0	1	44	1	0	1	0	1	1							

CTNet Remote I/O

DIP-switch also selects the Baud Rate

Baud rate must be the same at all nodes.



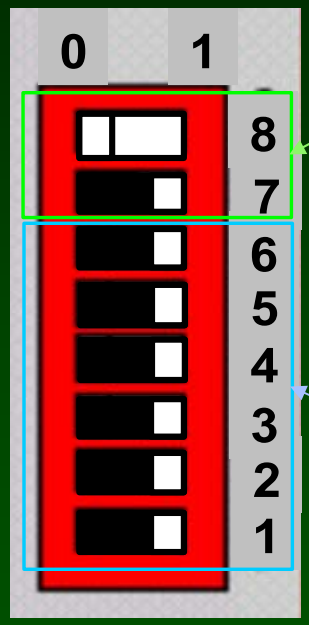
Data Rate (bits/sec)	DIP 7	DIP 8
5.0M	0	0
2.5M	1	0
1.25M	0	1
625K	1	1

Note that 2.50 M baud is the default and most usable setting

This setting selects 2.50 M baud

CTNet Remote I/O

Node 64 and 2.5 Mbaud DIP-switch setting

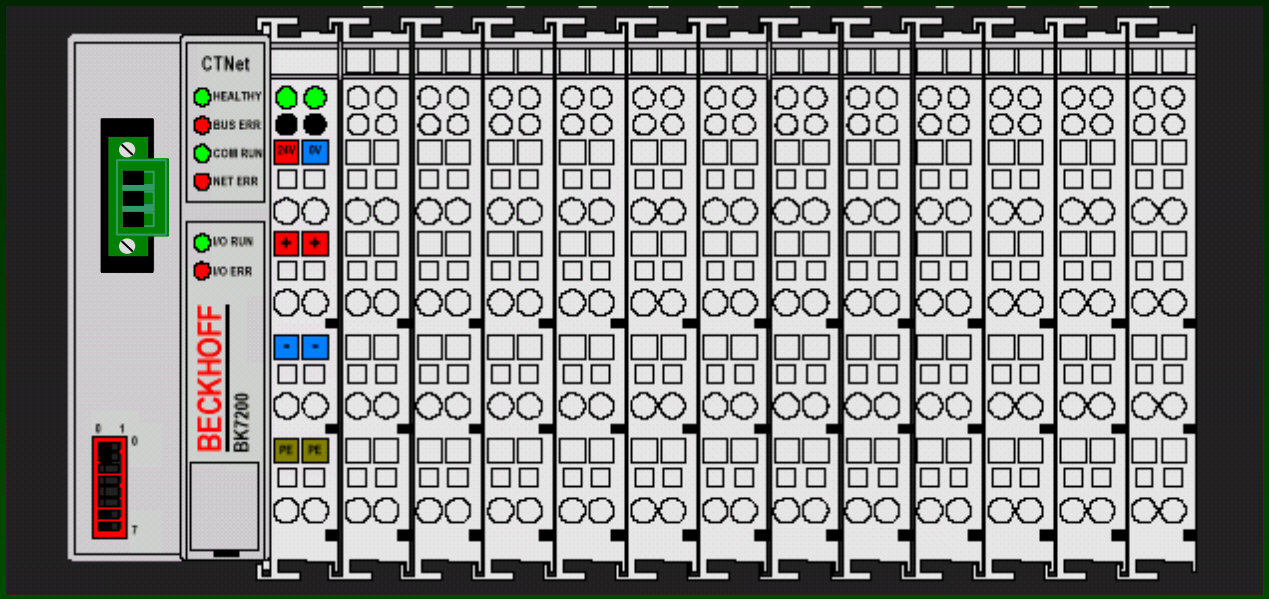


2.5 M baud, node 64

Data Rate (bits/sec)	DIP 7	DIP 8
5.0M	0	0
2.5M	1	0
1.25M	0	1
625K	1	1

Node	DIP Switch						Node	DIP Switch						Node	DIP Switch					
Address	6	5	4	3	2	1	Address	6	5	4	3	2	1	Address	6	5	4	3	2	1
1	0	0	0	0	0	0	23	0	1	0	1	1	0	45	1	0	1	1	0	0
2	0	0	0	0	0	1	24	0	1	0	1	1	1	46	1	0	1	1	0	1
3	0	0	0	0	1	0	25	0	1	1	0	0	0	47	1	0	1	1	1	0
4	0	0	0	0	1	1	26	0	1	1	0	0	1	48	1	0	1	1	1	1
5	0	0	0	1	0	0	27	0	1	1	0	1	0	49	1	1	0	0	0	0
6	0	0	0	1	0	1	28	0	1	1	0	1	1	50	1	1	0	0	0	1
7	0	0	0	1	1	0	29	0	1	1	1	0	0	51	1	1	0	0	1	0
8	0	0	0	1	1	1	30	0	1	1	1	0	1	52	1	1	0	0	1	1
9	0	0	1	0	0	0	31	0	1	1	1	1	0	53	1	1	0	1	0	0
10	0	0	1	0	0	1	32	0	1	1	1	1	1	54	1	1	0	1	0	1
11	0	0	1	0	1	0	33	1	0	0	0	0	0	55	1	1	0	1	1	0
12	0	0	1	0	1	1	34	1	0	0	0	0	1	56	1	1	0	1	1	1
13	0	0	1	1	0	0	35	1	0	0	0	1	0	57	1	1	1	0	0	0
14	0	0	1	1	0	1	36	1	0	0	0	1	1	58	1	1	1	0	0	1
15	0	0	1	1	1	0	37	1	0	0	1	0	0	59	1	1	1	0	1	0
16	0	0	1	1	1	1	38	1	0	0	1	0	1	60	1	1	1	0	1	1
17	0	1	0	0	0	0	39	1	0	0	1	1	0	61	1	1	1	1	0	0
18	0	1	0	0	0	1	40	1	0	0	1	1	1	62	1	1	1	1	0	1
19	0	1	0	0	1	0	41	1	0	1	0	0	0	63	1	1	1	1	1	0
20	0	1	0	0	1	1	42	1	0	1	0	0	1	64	1	1	1	1	1	1
21	0	1	0	1	0	0	43	1	0	1	0	1	0							
22	0	1	0	1	0	1	44	1	0	1	0	1	1							

Using the CTNet Remote I/O



SyPT Pro Tutorial
Basic Functional Test
SP to Beckhoff I/O Coupler

CTNet Remote I/O

SyPT Pro Example: Set up a basic test of a SP and a Beckhoff CTNet I/O

Strategy: CTNet Remote I/O at CTNet0 node address 64 @ 2.5 Mb

SP / SM-Apps at CTNet0 node address 1 @ 2.5 Mb

KL 1114 / KL 2114 Digital In / Out fitted to Beckhoff

KL 3062 / KL 4032 Analog In / Out fitted to Beckhoff

CT-Comm Cable link from PC / SyPT Pro to SP on CT-RTU segment

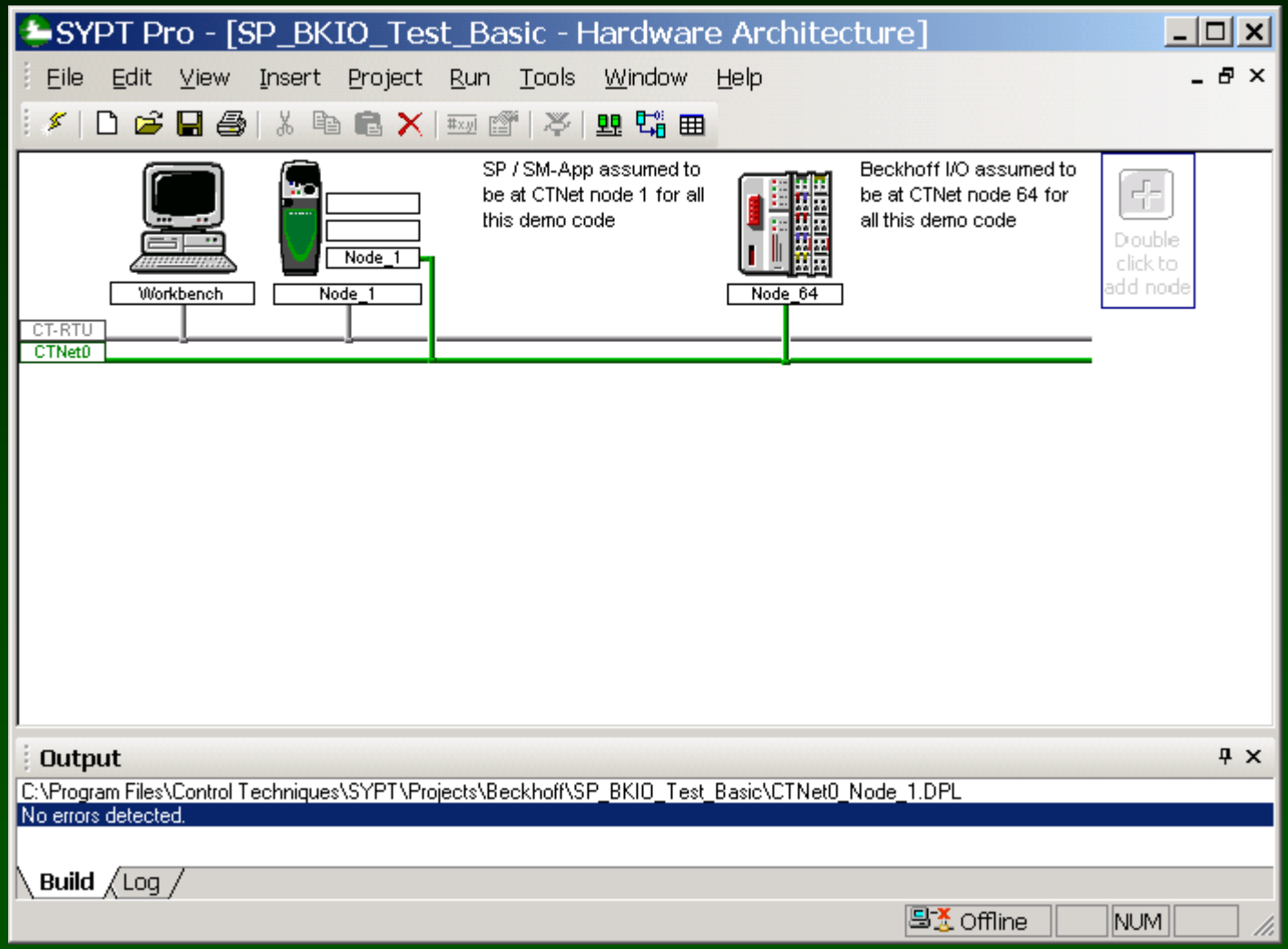
Watch Window will provide the “view” to the Beckhoff diagnostic registers

A simple SyPT program will generate activity

The SyPT project editor will verify everything connected and functioning when on-line.

CTNet Remote I/O

Example: Basic test (CTNet I/O is node 64)



CTNet Remote I/O

Example: Basic test (CTNet I/O is node 64)

DPL Editor - [CTNet0_Node_1.DPL (SP Beckhoff Basic Test)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

rkspace
CTNet0_Node_1.DPL (SP Beckh
Notes
Initial
Background

```
$AUTHOR  JPJ
$COMPANY CT
$TITLE   SP Beckhoff Basic Test
$VERSION V1.0.0
$DRIVE   SM-Applications

Notes{...}

Initial{...}

Background{...}
// line 74 - document explicitly the end of the source
```

Ready | Ln 74, Col 5 | NUM | INS

CTNet Remote I/O

Example: Basic test (CTNet I/O is node 64)

DPL Editor - [CTNet0_Node_1.DPL (SP Beckhoff Basic Test)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL (SP
    - Notes
    - Initial
    - Background

```
$AUTHOR  JPJ
$COMPANY CT
$TITLE   SP Beckhoff Basic Test
$VERSION V1.0.0
$DRIVE   SM-Applications

Notes{...}

Initial{
    // The initial task runs and completes before
    // any other tasks run

    // declare and initialize integer variable i%
    i% = 0

} //Initial

Background{...}
// line 74 - document explicitly the end of the source
```

Ready | Ln 49, Col 1 | NUM | INS

CTNet Remote I/O

Example: Basic test (CTNet I/O is node 64)

DPL Editor - [CTNet0_Node_1.DPL (SP Beckhoff Basic Test)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

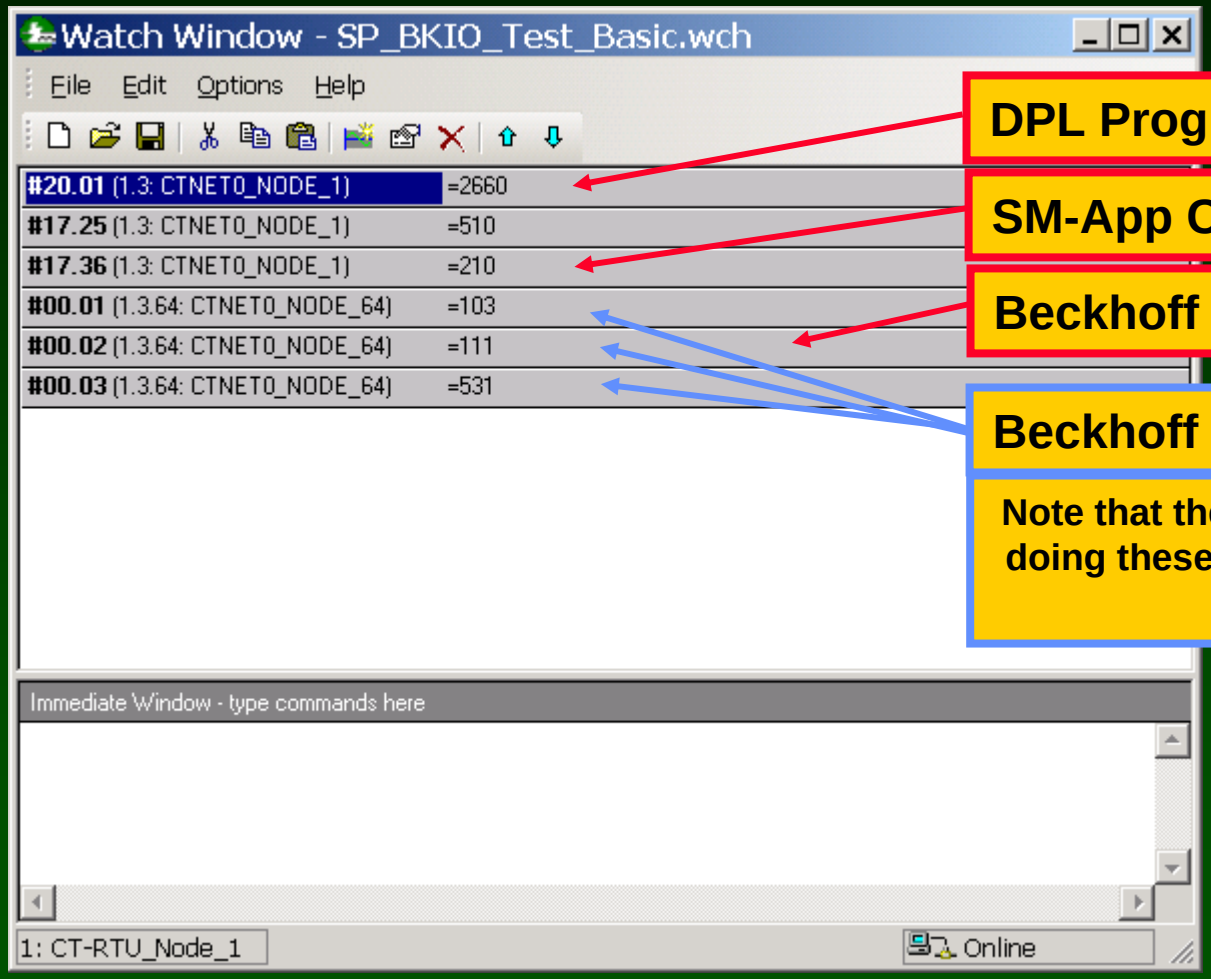
- Workspace
  - CTNet0_Node_1.DPL (SP
    - Notes
    - Initial
    - **Background**

```
$AUTHOR  JPJ
$COMPANY CT
$TITLE   SP Beckhoff Basic Test
$VERSION V1.0.0
$DRIVE   SM-Applications

Notes{...}

Initial{...}

Background{
top:

    // create activity to monitor and display
    if i% < 10000 then
        i% = i% + 1
    else
        i% = 1
    endif

    // write the changing value to a place
    // where it is observable
    #20.01 = i%

goto top: // main background loop
} //Background
// line 74 - document explicitly the end of the source
```

Ready | Ln 58, Col 1 | NUM | INS

CTNet Remote I/O

Watch window: Basic test (CTNet I/O is node 64)



DPL Program activity

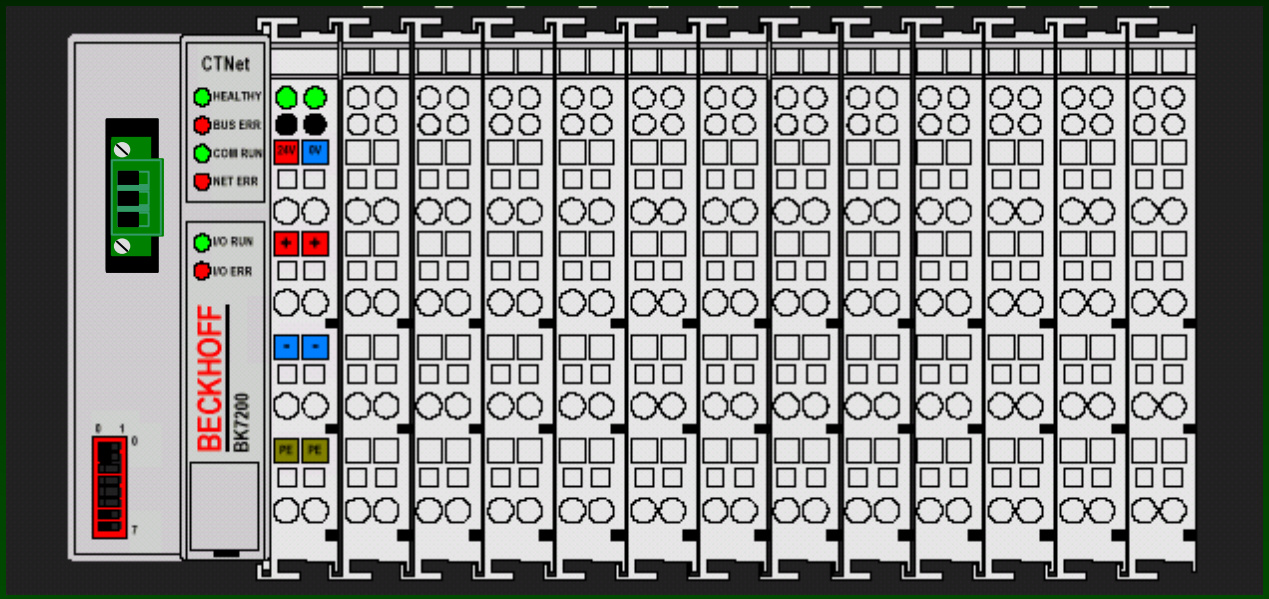
SM-App CTNet activity

Beckhoff CTNet activity

Beckhoff diag registers

Note that the watch window is doing these queries as CTNet reads

CTNet Remote I/O



SyPT Pro Tutorial
Basic Discrete Digital Input handling
noncyclic data transfer

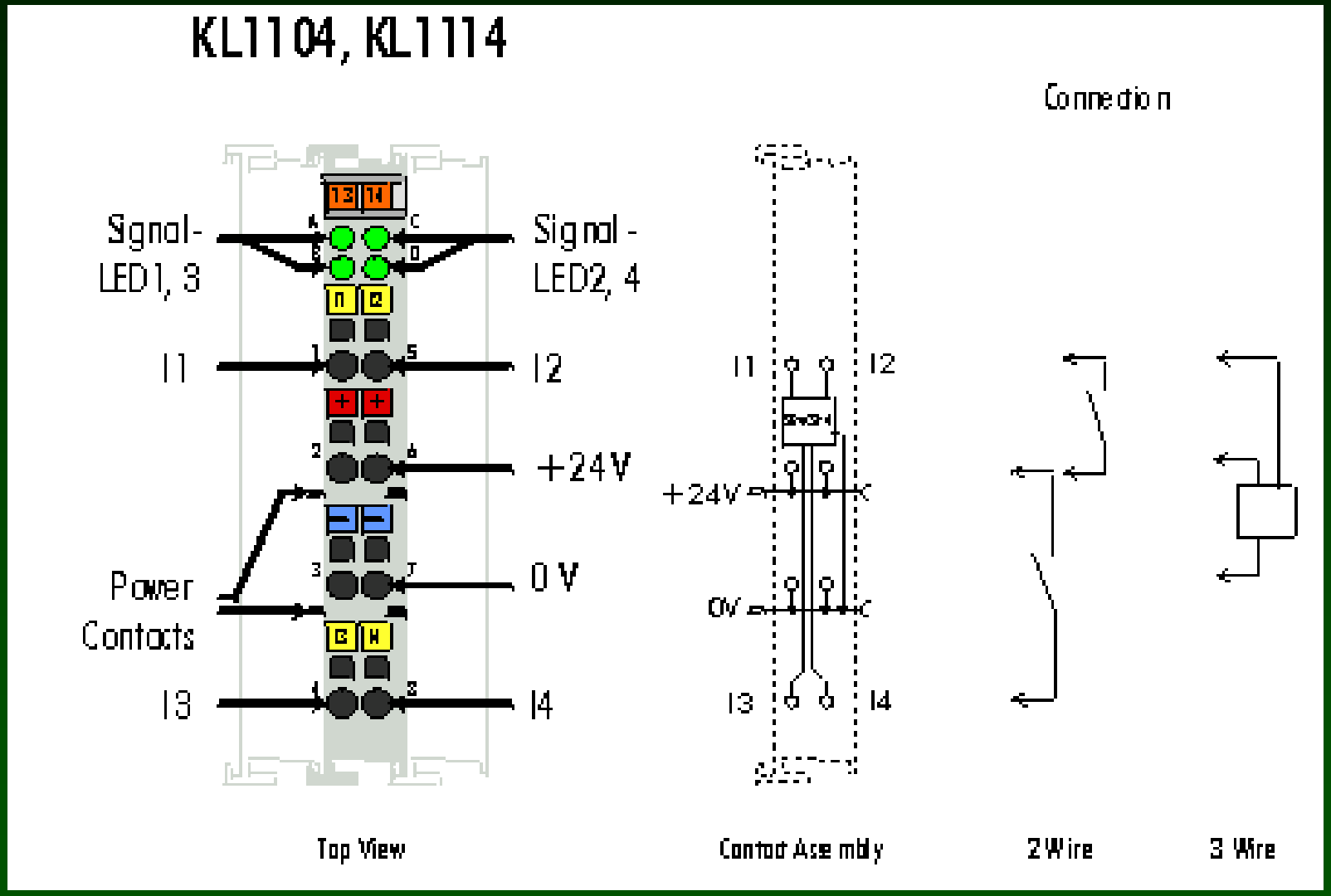
CTNet Remote I/O

A number of discrete input modules are available.

Type	Inputs	Description
KL1002	2	24V DC, filter 3.0 ms
KL1012	2	24V DC, filter 0.2 ms
KL1032	2	48V DC, filter 3.0 ms
KL1052	2	24V DC, filter 3.0 ms, with P/N switching
KL1104	4	24V DC, filter 3.0 ms
KL1114	4	24V DC, filter 0.2 ms
KL1124	4	5V DC, filter 0.2ms
KL1154	4	24V DC, filter 3.0 ms, with P/N switching
KL1164	4	24V DC, filter 0.2 ms, with P/N switching
KL1184	4	24V DC, filter 3.0 ms, with N switching
KL1702	2	230V AC
KL1712	2	120V AC/DC
KL1722	2	230V AC, no power contacts

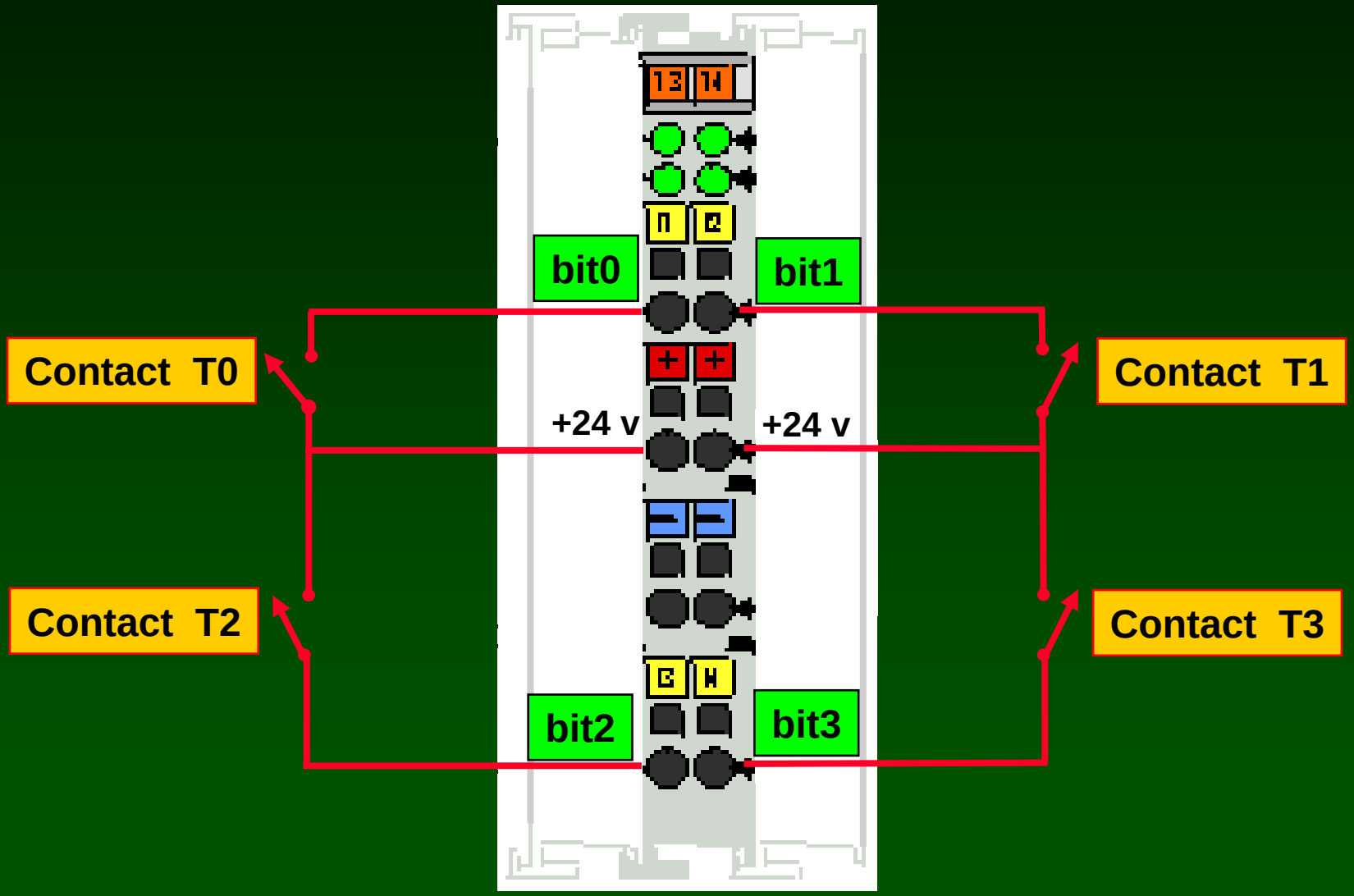
CTNet Remote I/O

Here's the Details on a 4-Input Discrete Input Module KL1104



CTNet Remote I/O

How to hook up a KL1104 Digital Input Module



CTNet Remote I/O

All digital inputs are packed together in the lower 16 bits of Menu 1 parameters.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16

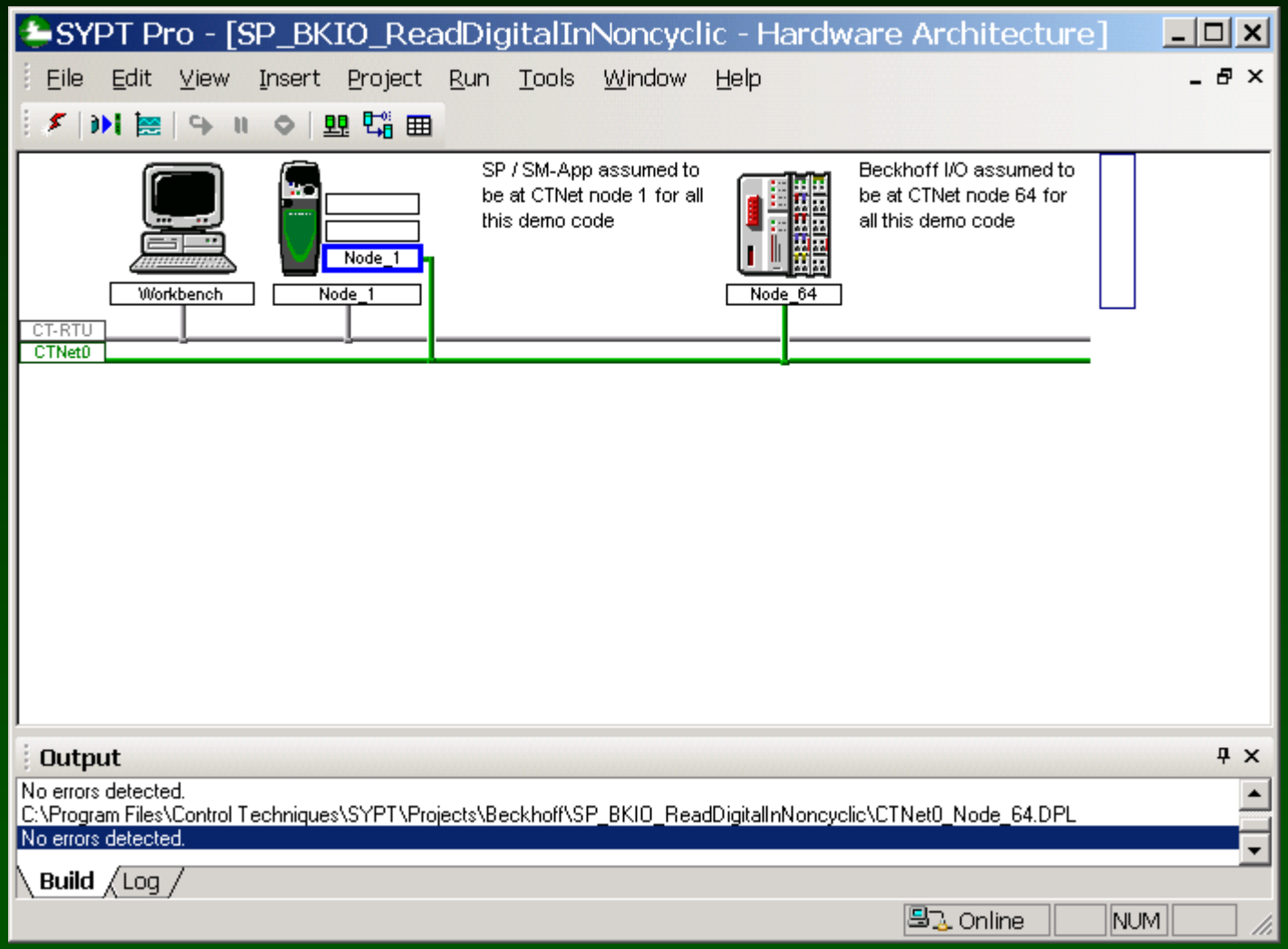
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
T15	T14	T13	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0

255 bits of digital input can be configured with one Bus Coupler.

Reference	I/O Points	Reference	I/O Points
#1.00	T0 - T15	#1.08	T128 - T143
#1.01	T16 - T31	#1.09	T144 - T159
#1.02	T32 - T47	#1.10	T160 - T175
#1.03	T48 - T63	#1.11	T176 - T191
#1.04	T64 - T79	#1.12	T192 - T207
#1.05	T80 - T95	#1.13	T208 - T223
#1.06	T96 - T111	#1.14	T224 - T239
#1.07	T112 - T127	#1.15	T240 - T255

CTNet Remote I/O

Example: Read 4 bits of Digital Inputs (CTNet I/O is node 64)



CTNet Remote I/O

Example: Read 4 bits of Digital Inputs (CTNet I/O is node 64)

DPL Editor - [CTNet0_Node_1.DPL (Read Digital Input Noncyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL
    - Notes
    - Alias
    - Initial
    - **Background**

```
$AUTHOR   Jim Lynch, Jim Jeffers
$COMPANY  Control Techniques
$TITLE    Read Digital Input Noncyclic
$VERSION  V1.0.0
$DRIVE    SM-Applications

Notes{...}

Alias{...}

Initial{...}

Background{...}
// line 110
```

Task Manager

CTNet0_Node_1.DPL

CPU Free: 95%

Initial St

Background St

Ready | Ln 84, Col 1 | Stopped | NUM | INS

CTNet Remote I/O

Example: Read 4 bits of Digital Inputs (CTNet I/O is node 64)

DPL Editor - [CTNet0_Node_1.DPL (Read Digital Input Noncyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL
    - Notes
    - Alias
    - Initial
    - Background

```
$AUTHOR  Jim Lynch, Jim Jeffers
$COMPANY Control Techniques
$TITLE   Read Digital Input Noncyclic
$VERSION V1.0.0
$DRIVE   SM-Applications

Notes{...}

Alias{
// No boolean data type is provided
// by Drive Programming Language
$define ON       1
$define OFF      0
// CTNet RDNET and WRNET status codes
$define NODE_IDLE           0
$define COMMAND_COMPLETE    1
$define NODE_BUSY           3
$define NODE_TIMEOUT        4
$define INTERNAL_ERROR      5
$define PRAM_DOES_NOT_EXIST -2
$define PRAM_IS_READ_ONLY   -2
$define PRAM_IS_WRITE_ONLY  -3
$define PRAM_OUT_OF_RANGE   -5
$define IO_BOX_COMMS_ERROR  -6

} //Alias

Initial{...}

Background{...}
// line 110
```

Task Manager

CTNet0_Node_1.DPL

CPU Free: 95%

Initial

Background

Ready | Ln 49, Col 1 | Stopped | NUM | INS

CTNet Remote I/O

Example: Read 4 bits of Digital Inputs (CTNet I/O is node 64)

DPL Editor - [CTNet0_Node_1.DPL (Read Digital Input Noncyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL
    - Notes
    - Alias
    - **Initial**
    - Background

```
$AUTHOR  Jim Lynch, Jim Jeffers
$COMPANY Control Techniques
$TITLE   Read Digital Input Noncyclic
$VERSION V1.0.0
$DRIVE   SM-Applications

Notes{...}

Alias{...}

Initial{

bit0% = OFF
bit1% = OFF
bit2% = OFF
bit3% = OFF
input%  =   0
status% = NODE_IDLE
// following is to read parameter #1.00 on node 64
BKIOnode% = 64       // node 64 for this example
BKIOmenu% = 1        // parameter #1.00 menu value
BKIOparameter% = 00 // parameter #1.00 parameter value
timeout% = 100       // 100 ms

} //Initial

Background{...}
// line 110
```

Task Manager

CTNet0_Node_1.DPL

CPU Free: 95%

Initial	St
Background	St

Ready | Ln 68, Col 1 | Stopped | NUM | INS

CTNet Remote I/O

Example: Read 4 bits of Digital Inputs (CTNet I/O is node 64)

DPL Editor - [CTNet0_Node_1.DPL (Read Digital Input Noncyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

rkspace
CTNet0_Node_1.DPL (Read
Notes
Alias
Initial
Background

```
$TITLE    Read Digital Input Noncyclic
$VERSION  V1.0.0
$DRIVE    SM-Applications

Notes{...}

Alias{...}

Initial{...}

Background{
top:

// request the contents of the first word of digital input
status% = RDNET(BKIOnode%, BKIOmenu%, BKIOparameter%, timeout%)
// becomes status% = RDNET(64, 1, 0, 100)

// note that 100 ms delay exists between the RDNET command and the
// check for the status.  This approach ok in an initial or in a
// background task.

if status% = COMMAND_COMPLETE then
    // after a 100 ms wait, the result should be available and valid
    input% = NETREPLY(0)
    // input% is a 32 bit integer, with the lower 16-bits reflecting a
    // collection of boolean values

    // decode the individual bits from the collection
    bit0% = input%.0     // DPL bit manipulation operation illustrated
    bit1% = input%.1     // showing one way to un-pack the word into
    bit2% = input%.2     // the individual bits
    bit3% = input%.3
endif

goto top: // main background loop
} //Background
// line 110
```

Task Manager

CTNet0_Node_1.DPL

CPU Free: 95%

Initial

Background

Ready | Ln 48, Col 1 | Stopped | NUM | INS

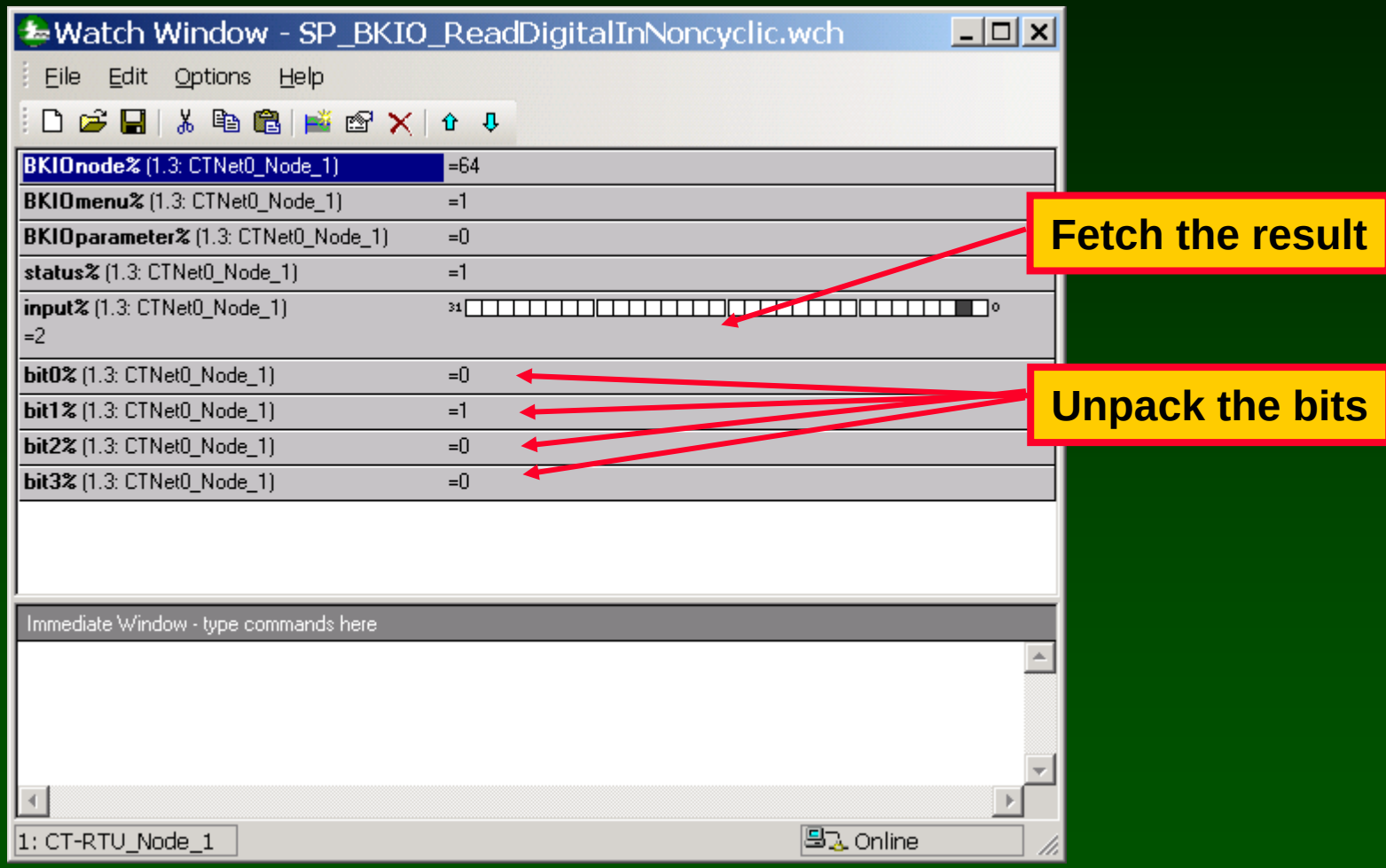
Query for node 64, #01.00

Fetch the result

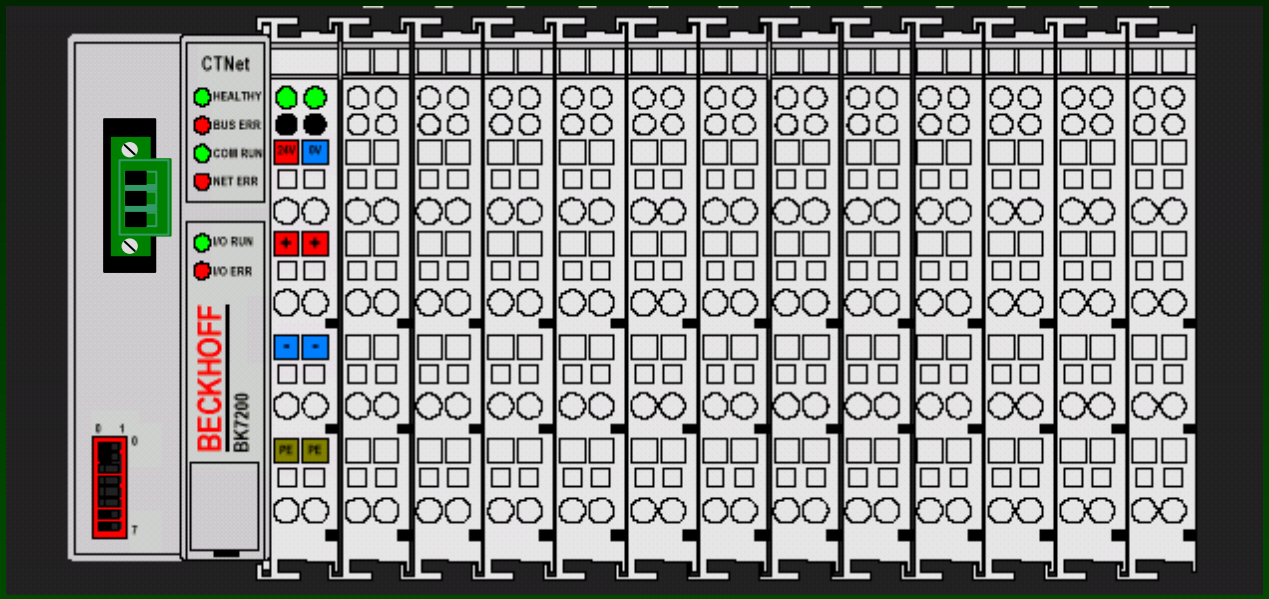
Unpack the bits

CTNet Remote I/O

Watch window: Read 4 bits of Digital Inputs (CTNet I/O is node 64)



CTNet Remote I/O



SyPT Pro Tutorial
Basic Discrete Digital Output handling
noncyclic data transfer

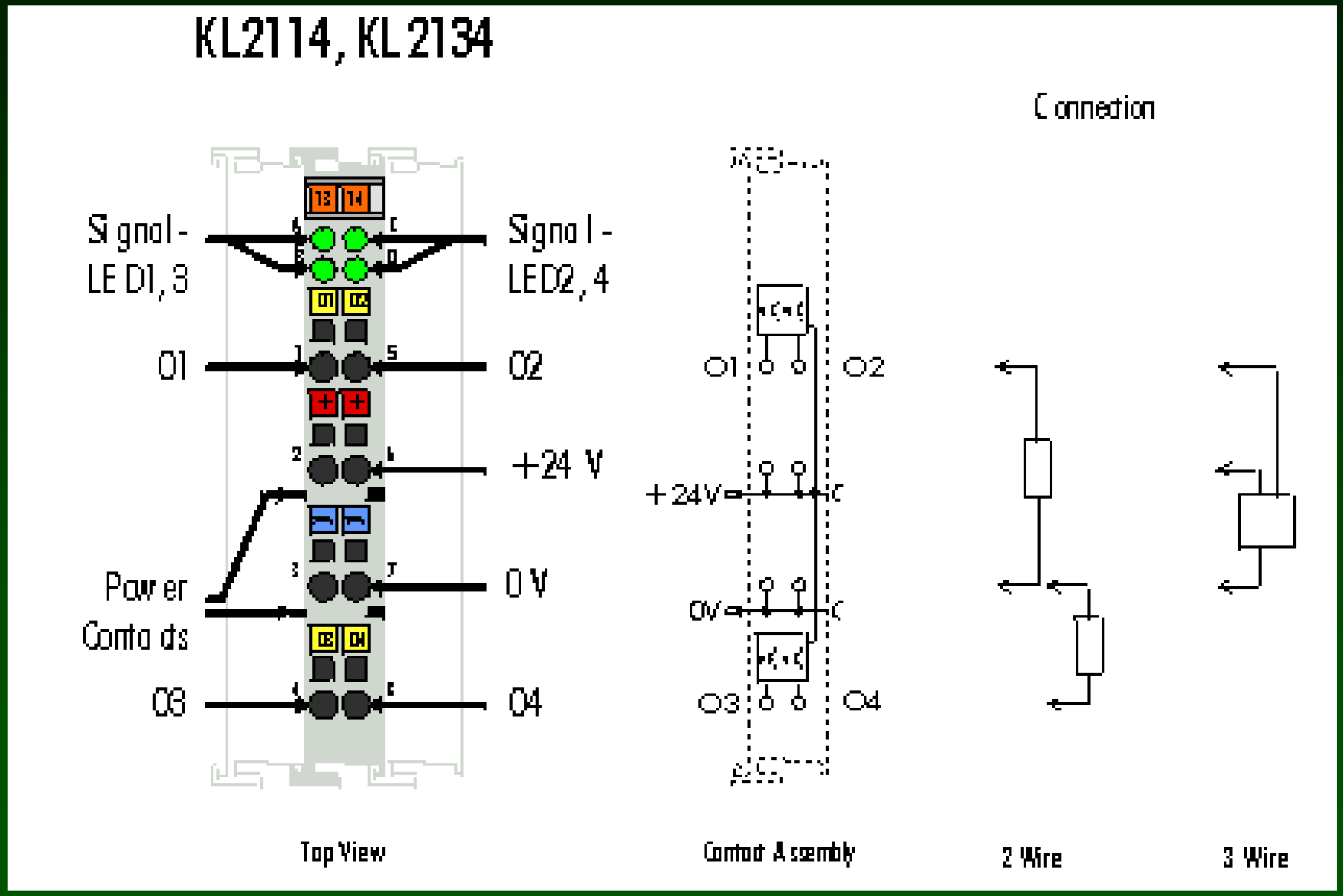
CTNet Remote I/O

A number of discrete output modules are available.

Terminal	Outputs	Description
KL2012	2	24V DC, 0.5A
KL2022	2	24V DC, 2.0A
KL2032	2	48V DC, 0.5A, with polarity protection
KL2114	4	24V DC, 0.5A
KL2124	4	5V DC
KL2134	4	24V DC, 0.5A, with polarity protection
KL2184	4	24V DC, 0.5A, with N switching
KL2602	2	Relay outputs, 230V AC, 2A, normally open
KL2612	2	Relay outputs, 125V AC, 0.5A, changeover
KL2622	2	Relay outputs, 230V AC, 0.5A, isolated normally open
KL2631	1	Relay output, 400V AC, 3A normally open
KL2702	2	Solid state outputs, 230V AC, 0.3A

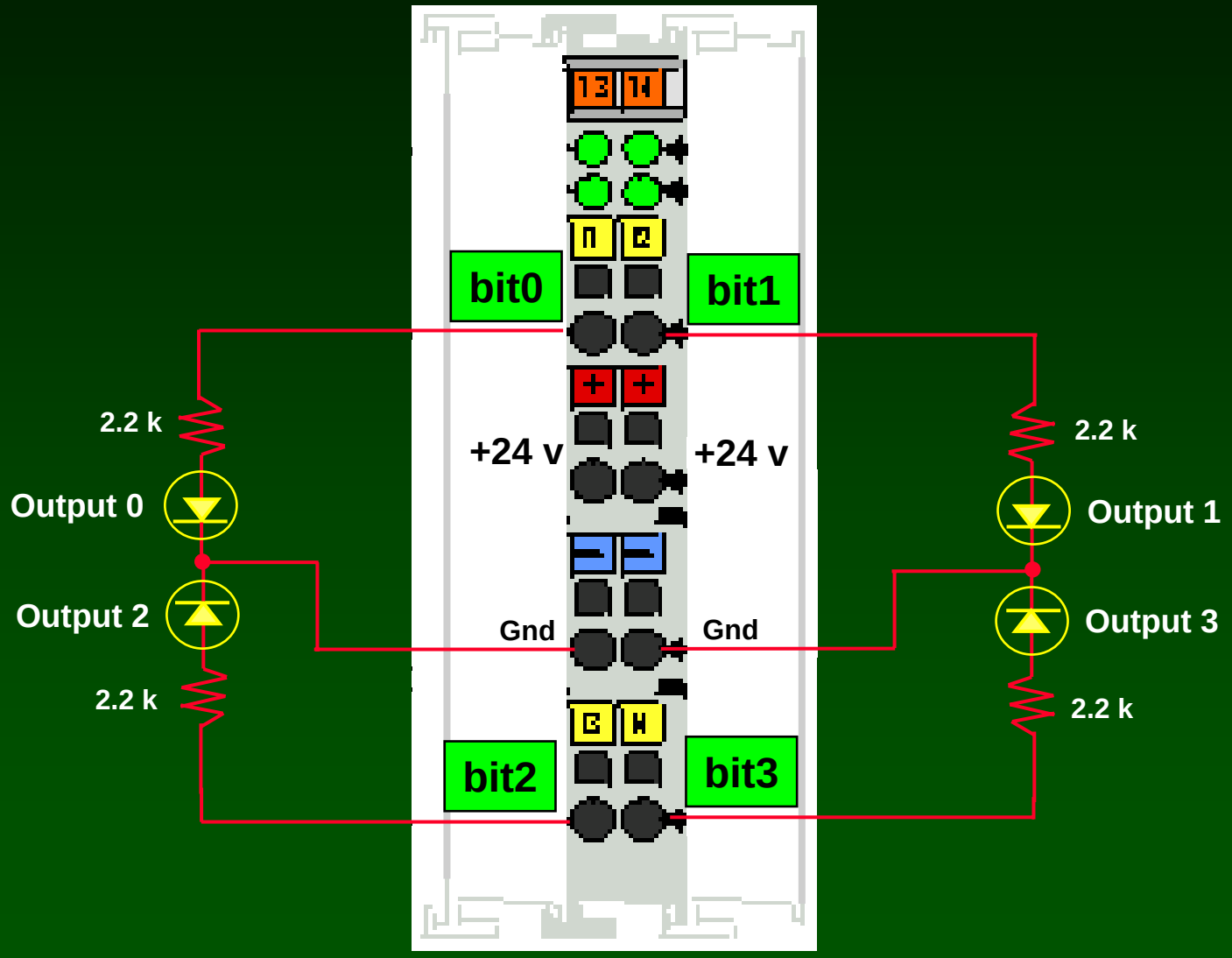
CTNet Remote I/O

Here's the Details on a 4-output Discrete Output Module KL2114



CTNet Remote I/O

How to hook up a KL2114 Digital Output Module



CTNet Remote I/O

All digital outputs are packed together in the lower 16 bits of Menu 1 parameters.

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
M15	M14	M13	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	M0

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
T15	T14	T13	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0

255 bits of digital output can be configured with one Bus Coupler.

Reference	I/O Points	Reference	I/O Points
#2.00	T0 - T15	#2.08	T128 - T143
#4.01	T16 - T31	#2.08	T144 - T159
#2.02	T32 - T47	#2.09	T160 - T175
#2.03	T48 - T63	#2.10	T176 - T191
#2.04	T64 - T79	#2.11	T192 - T207
#2.05	T80 - T95	#2.12	T208 - T223
#2.06	T96 - T111	#2.13	T224 - T239
#2.07	T112 - T127	#2.14	T240 - T255

CTNet Remote I/O

Digital Outputs have "MASK" bits

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
M15	M14	M13	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	M0

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
T15	T14	T13	T12	T11	T10	T9	T8	T7	T6	T5	T4	T3	T2	T1	T0

You have to set the MASK bit to alter a digital bit.

Example: <u>set bit 1</u> (note that 0x<number> signifies hexadecimal format)

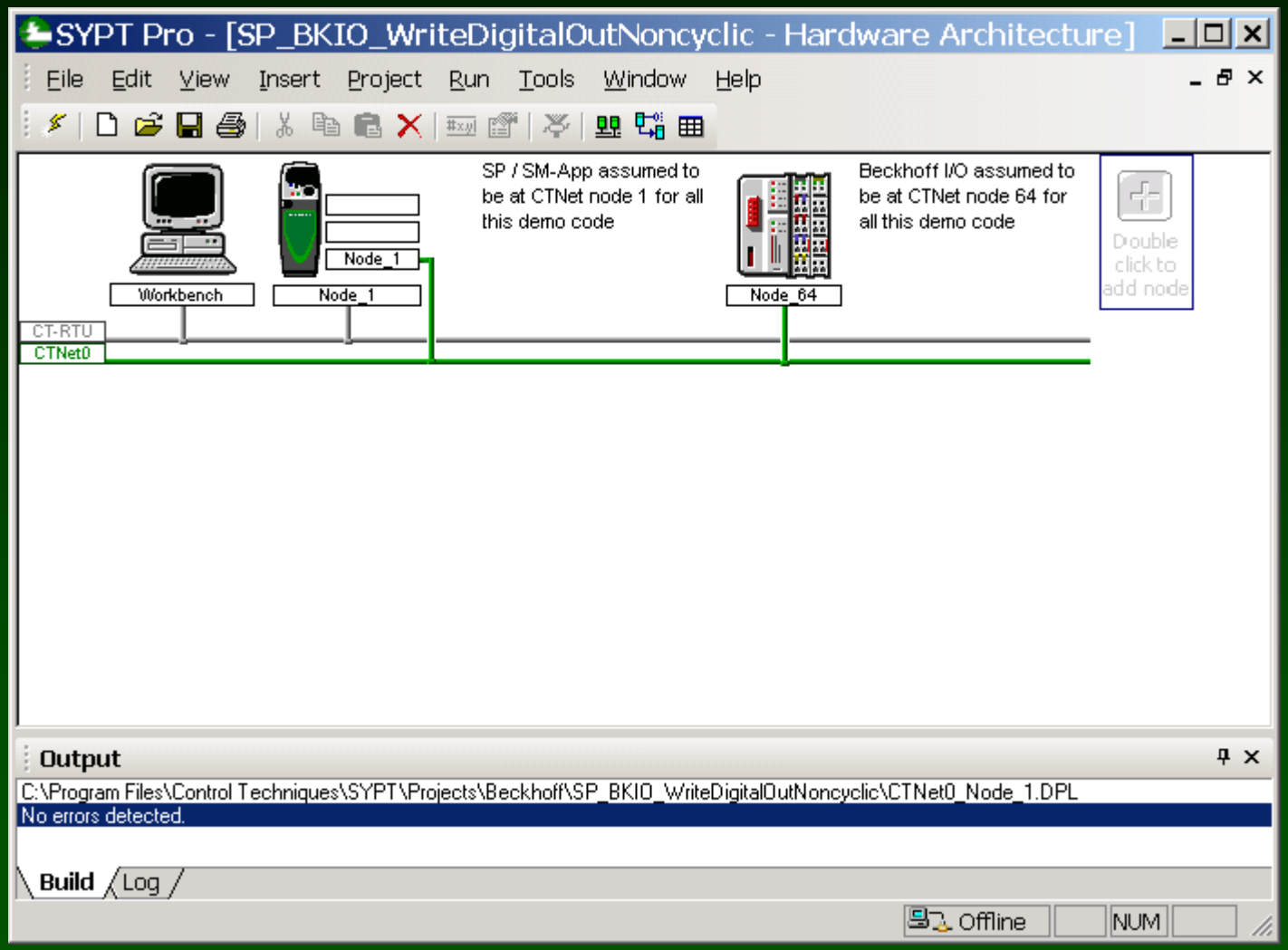
#64.02.00 = 0x20000 // clear digital output bit 1

#64.02.00 = 0x20002 // set digital output bit 1

#64.02.00 = 0x00002 // has no effect (no mask bit set)

CTNet Remote I/O

Example: Blinking bit “b1” of Digital Output at #02.00 (CTNet I/O is node 64)



CTNet Remote I/O

Example: Blinking bit “b1” of Digital Output at #02.00 (CTNet I/O is node 64)

DPL Editor - [CTNet0_Node_1.DPL (Write Digital Output Noncyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

rkspace
CTNet0_Node_1.DPL (Write Dig
Notes
Alias
Initial
Background

Task Manager

CTNet0_Node_1.DPL
CPU Free: 95%
Initial Stoppe
Background Stoppe

```
$AUTHOR  Jim Lynch, Jim Jeffers
$COMPANY Control Techniques
$TITLE   Write Digital Output Noncyclic
$VERSION V1.0.0
$DRIVE   SM-Applications

Notes{...}

Alias{...}

Initial{...}

Background{...}
// line 103
```

Ready | Ln 103, Col 10 | Stopped | NUM | INS

CTNet Remote I/O

Example: Blinking bit “b1” of Digital Output at #02.00 (CTNet I/O is node 64)

DPL Editor - [CTNet0_Node_1.DPL (Write Digital Output Noncyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL (Wr
    - Notes
    - Alias
    - Initial
    - Background

Task Manager

CTNet0_Node_1.DPL

CPU Free: 95%

Initial	Stoppe
Background	Stoppe

```
$AUTHOR  Jim Lynch, Jim Jeffers
$COMPANY Control Techniques
$TITLE   Write Digital Output Noncyclic
$VERSION V1.0.0
$DRIVE   SM-Applications

Notes{...}

Alias{
// No boolean data type is provided
// by Drive Programming Language
$define ON       1
$define OFF      0
// CTNet RDNET and WRNET status codes
$define NODE_IDLE            0
$define COMMAND_COMPLETE     1
$define NODE_BUSY            3
$define NODE_TIMEOUT         4
$define INTERNAL_ERROR       5
$define PRAM_DOES_NOT_EXIST -2
$define PRAM_IS_READ_ONLY   -2
$define PRAM_IS_WRITE_ONLY  -3
$define PRAM_OUT_OF_RANGE   -5
$define IO_BOX_COMMS_ERROR  -6
// only 4 output bits available, b0 TO b3
$define OUT_MASK             0x000F0000

} //Alias

Initial{...}

Background{...}
// line 103
```

MASK bits M0 thru M3 set up here

Ready | Ln 45, Col 15 | Stopped | NUM | INS

CTNet Remote I/O

Example: Blinking bit “b1” of Digital Output at #02.00 (CTNet I/O is node 64)

DPL Editor - [CTNet0_Node_1.DPL (Write Digital Output Noncyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL (Wr
    - Notes
    - Alias
    - Initial
    - Background

Task Manager

CTNet0_Node_1.DPL

CPU Free: 95%

Initial	Comple
Background	Runnir

```
$AUTHOR   Jim Lynch, Jim Jeffers
$COMPANY  Control Techniques
$TITLE    Write Digital Output Noncyclic
$VERSION  V1.0.0
$DRIVE    SM-Applications

Notes{...}

Alias{...}

Initial{

bToggle% = OFF
OutValue% = 0
OutputMask% = OUT_MASK

status% = NODE_IDLE
// following is to write to parameter #2.00 on node 64
BKIOnode% = 64       // node 64 for this example
BKIOmenu% = 2        // parameter #2.00 menu value
BKIOparameter% = 00 // parameter #2.00 parameter value
dpos% = 0            // decimal point in 0 position, x1 format
timeout% = 100       // 100 ms

} //Initial

Background{...}
// line 103
```

Ready | Ln 62, Col 1 | Running | NUM | INS

CTNet Remote I/O

Example: Blinking bit “b1” of Digital Output at #02.00 (CTNet I/O is node 64)

DPL Editor - [CTNet0_Node_1.DPL (Write Digital Output Noncyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL (Wr
    - Notes
    - Alias
    - Initial
    - **Background**

Task Manager

CTNet0_Node_1.DPL

CPU Free: 95%

Initial	Comple
Background	Runnir

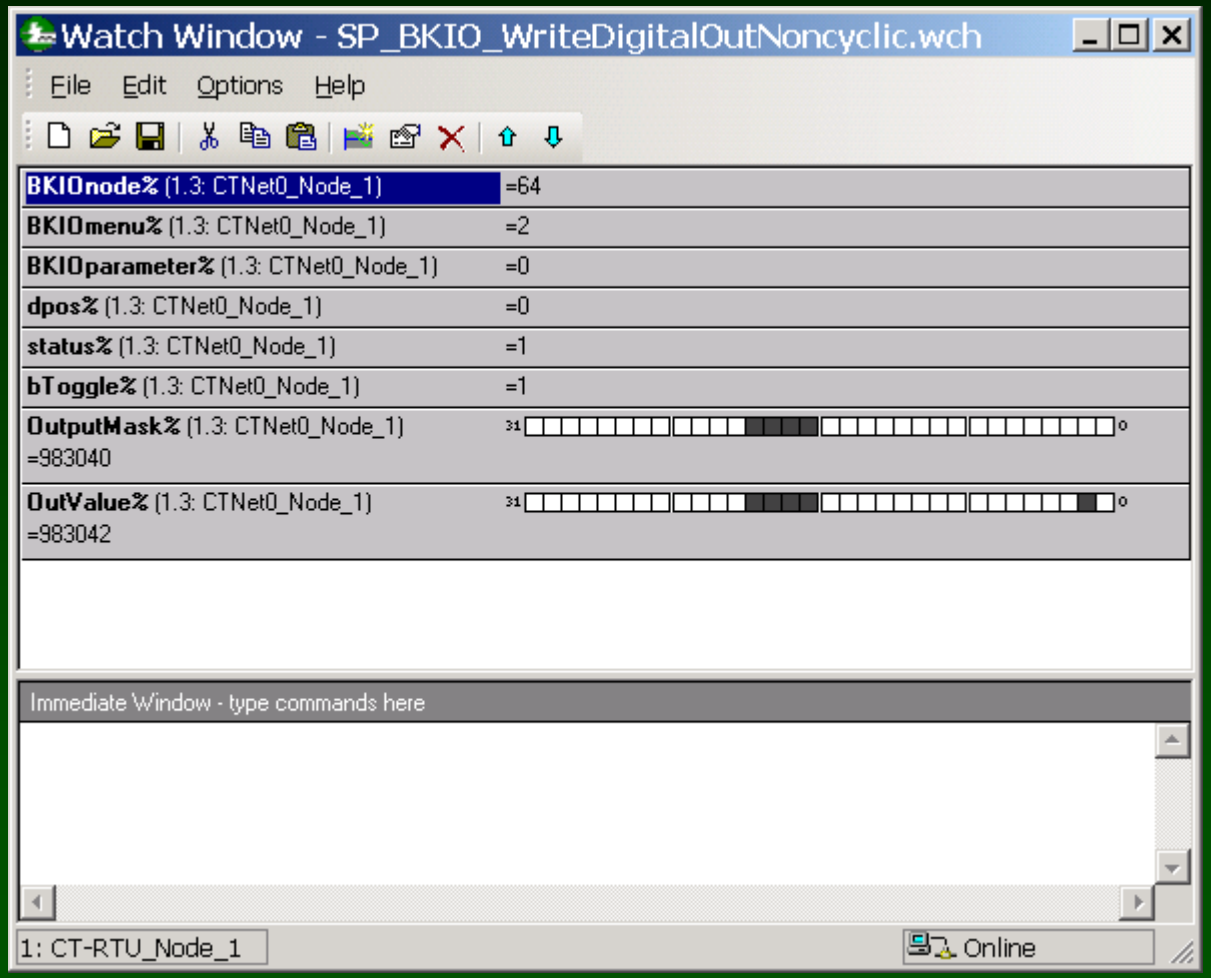
```
$COMPANY Control Techniques
$TITLE   Write Digital Output Noncyclic
$VERSION V1.0.0
$DRIVE   SM-Applications

Notes{...}

Alias{...}

Initial{...}

Background{
top:
// create the toggle bit
bToggle% = bToggle% ^ ON    // bitwise exclusive OR here

// assemble the new OutValue%
OutValue%.1 = bToggle%              // inserted to bit position one (1).
OutValue% = OutValue% | OutputMask% // bitwise OR here to set upper bits

// write to the Beckhoff I/O
status% = WRNET(BKIOnode%, BKIOmenu%, BKIOparameter%, OutValue%, dpos%, timeout%)
// becomes status% = WRNET(64, 2, 0, value%, 0, 100)

// note that 100 ms automatic timeout is being used with the WRNET
// command here.  This means that the program will stop here and wait up to
// 100 ms for a response from node 64 before continuing.  This approach ok
// in an initial or in a background task.

DELAY (5)
// the program stops for an additional 0.5 second wait here.  This has
// the effect of controlling how often the bit changes state and the
// update rate to the Beckhoff I/O.

goto top: // main background loop
} //Background
// line 103
```

Don’t forget to set the MASK bits !!!

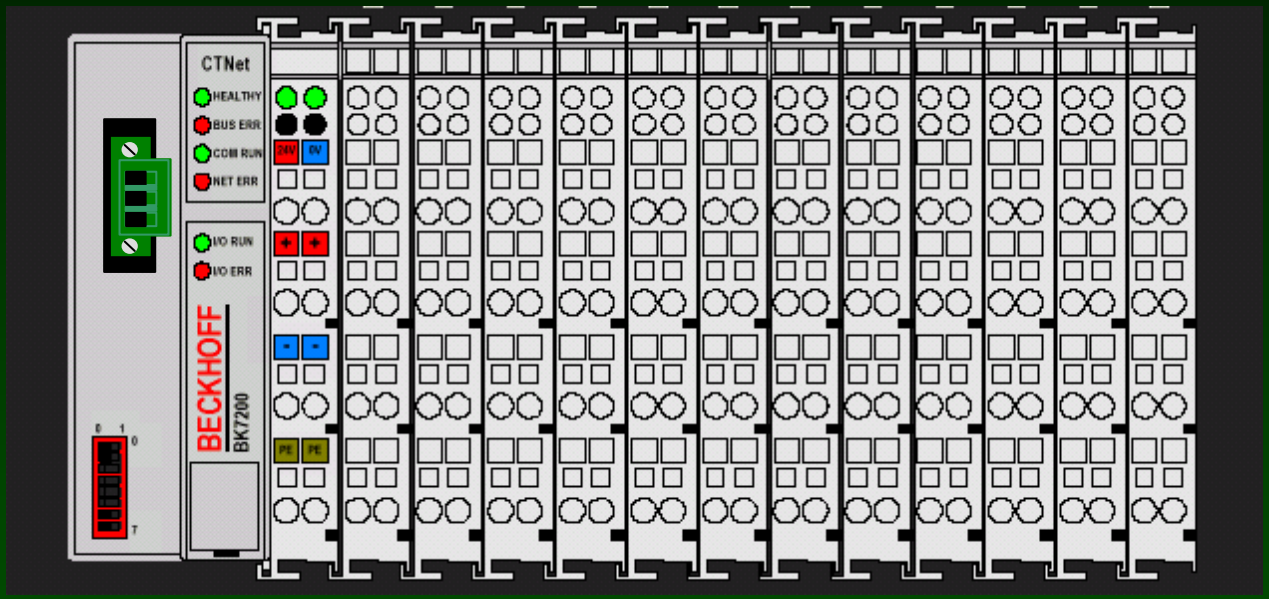
Ready | Ln 78, Col 1 | Running | NUM | INS

CTNet Remote I/O

Watch window:
Blinking bit “b1” of Digital Output at #02.00 (CTNet I/O is node 64)



CTNet Remote I/O



SyPT Pro Tutorial
Basic Analog Input handling
noncyclic data transfer

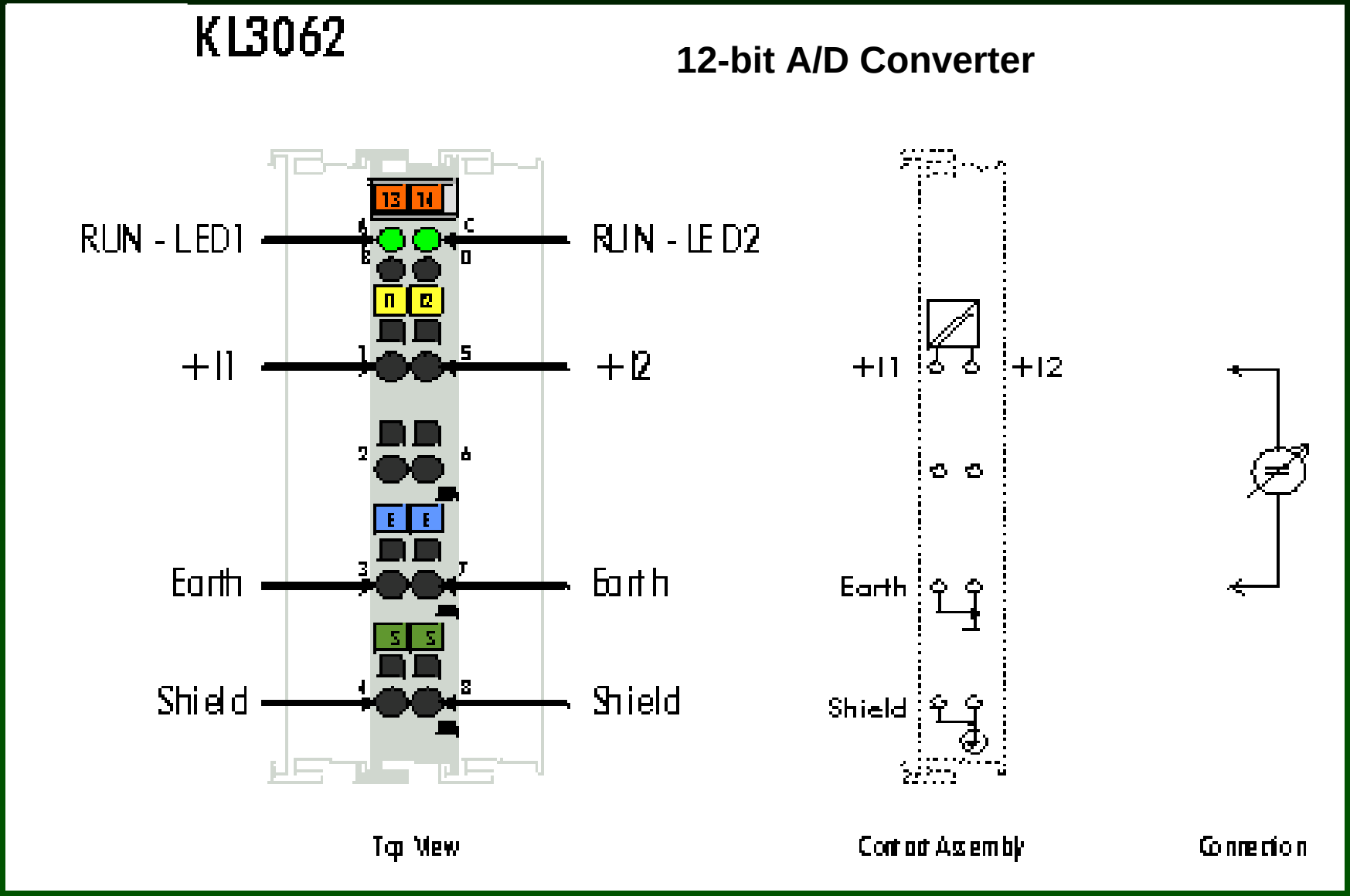
CTNet Remote I/O

A number of analog input modules are available.

Terminal	Inputs	Description
KL3002	2	±10V, differential inputs
KL3012	2	0-20mA, differential inputs
KL3022	2	4-20mA, differential inputs
KL3042	2	0-20mA, power supply for transducers via power contacts
KL3052	2	4-20mA, power supply for transducers via power contacts
KL3062	2	±10V, single-ended inputs
KL3064	4	±10V, single-ended inputs

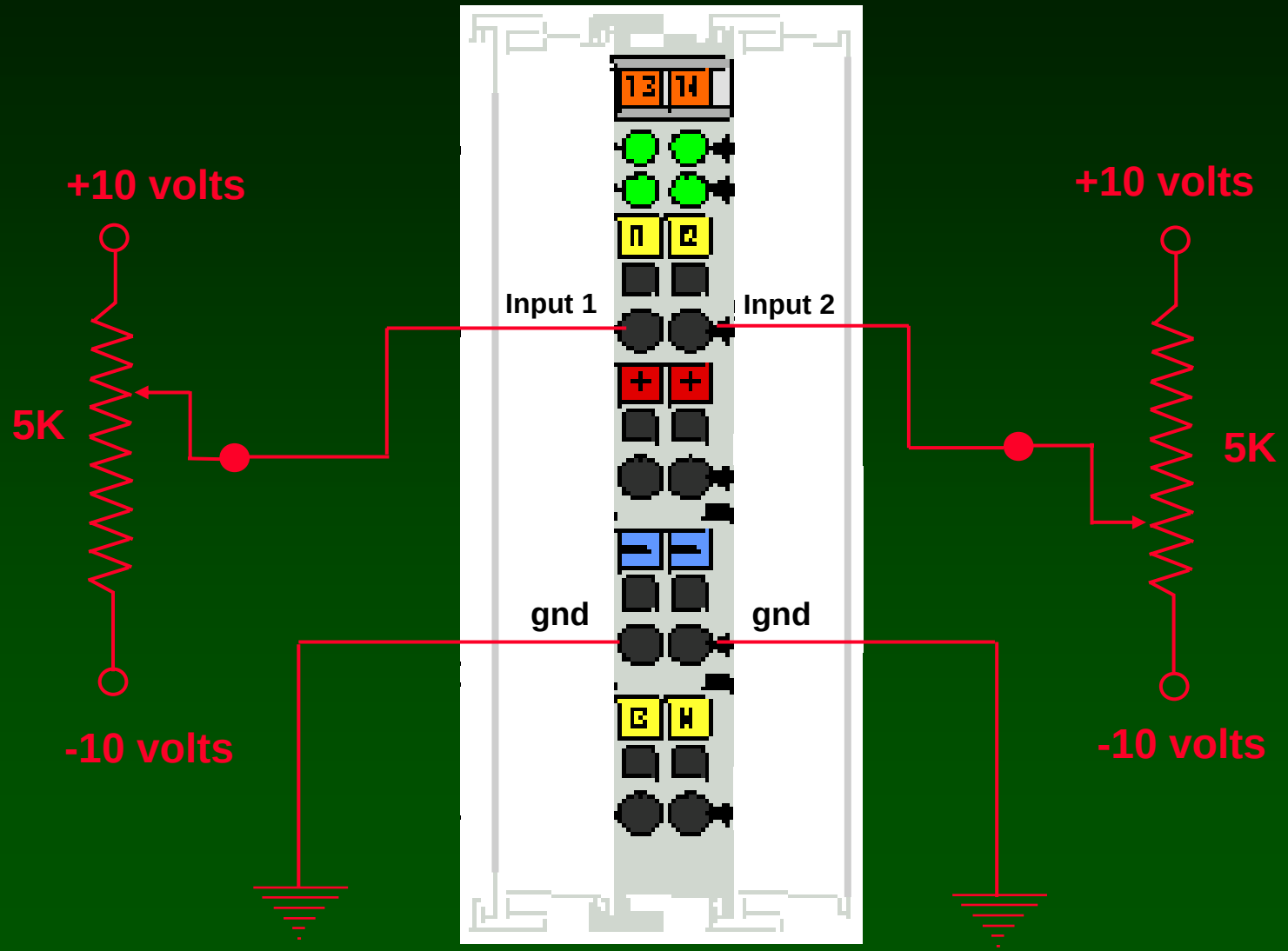
CTNet Remote I/O

Here's the Details on a 2-input Analog Input Module KL3062



CTNet Remote I/O

How to hook up a KL3062 Analog Input Module



CTNet Remote I/O

Analog Inputs are collected into Menu 3

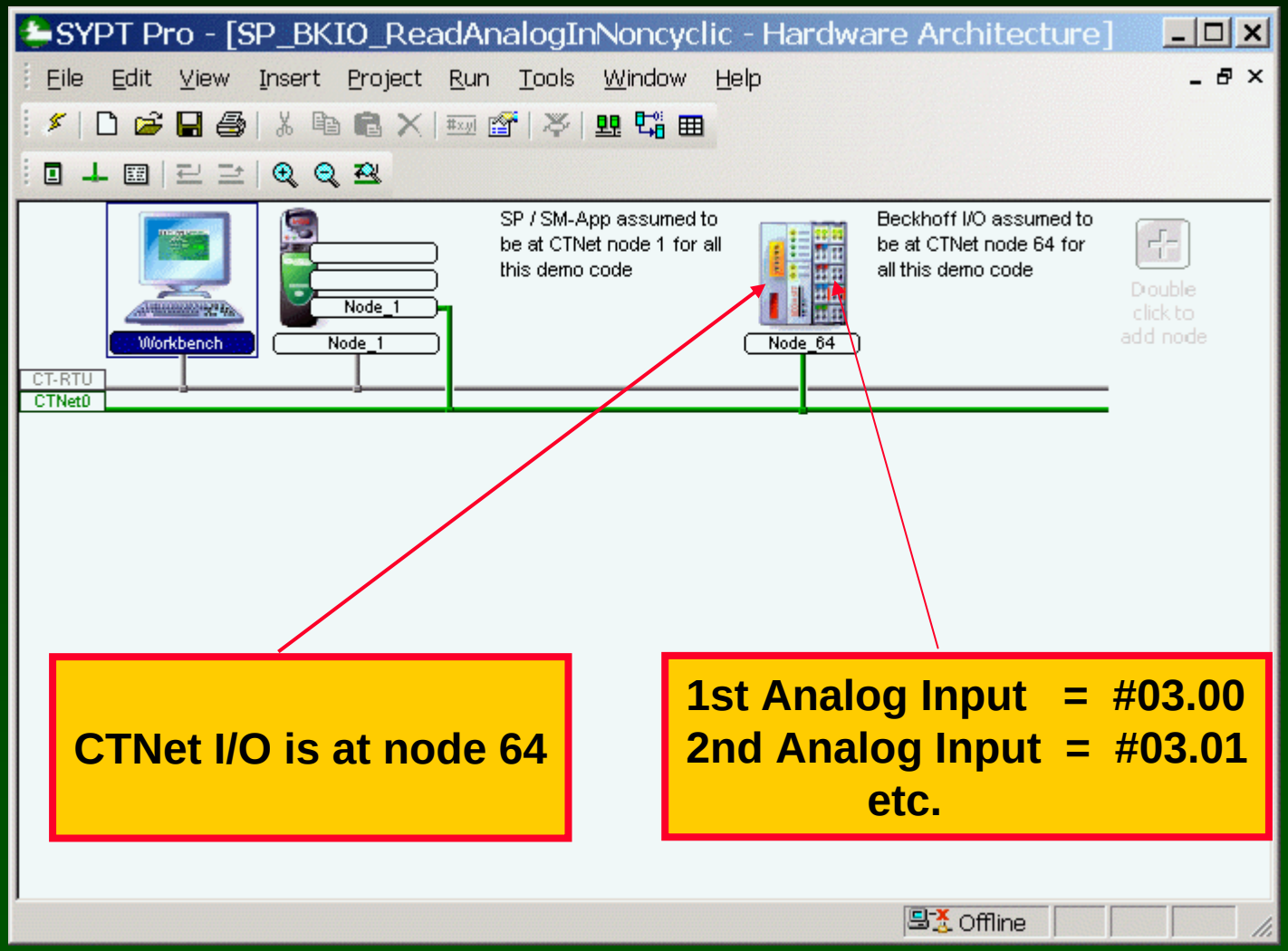
Reference	I/O Point
#3.00	0
#3.01	1
#3.02	2
#3.xx	xx
#3.98	98
#3.99	99

Analog Inputs are “scaled”, 32767 is always full output.

Input Range	Minimum Input	Maximum Input	Resolution
0-10V	0V = 0	+10V = 32767	5mV
±10V	-10V = -32768	+10V = 32767	5mV
0-20mA	0mA = 0	20mA = 32767	5µA
4-20mA	4mA = 0	20mA = 32767	4µA

CTNet Remote I/O

Example: Read the very first analog input



CTNet I/O is at node 64

1st Analog Input = #03.00
2nd Analog Input = #03.01
etc.

CTNet Remote I/O

Example: Read the very first analog input

DPL Editor - [CTNet0_Node_1.DPL (Read Analog Input Noncyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL (Read Analog Input Noncyclic)
    - Notes
    - Alias
    - Initial
    - Background

```
$AUTHOR  Jim Lynch, Jim Jeffers
$COMPANY Control Techniques
$TITLE   Read Analog Input Noncyclic
$VERSION V1.0.0
$DRIVE   SM-Applications

Notes{...}

Alias{...}

Initial{...}

Background{...}
// line 105
```

Ready | Ln 105, Col 12 | INS

CTNet Remote I/O

Example: Read the very first analog input

DPL Editor - [CTNet0_Node_1.DPL (Read Analog Input Noncyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL (Read Analog Input Noncyclic)
    - Notes
    - **Alias**
    - Initial
    - Background

```
$AUTHOR   Jim Lynch, Jim Jeffers
$COMPANY  Control Techniques
$TITLE    Read Analog Input Noncyclic
$VERSION  V1.0.0
$DRIVE    SM-Applications

Notes{...}

Alias{
// No boolean data type is provided
// by Drive Programming Language
$define ON      1
$define OFF     0
// CTNet RDNET and WRNET status codes
$define NODE_IDLE        0
$define COMMAND_COMPLETE   1
$define NODE_BUSY        3
$define NODE_TIMEOUT     4
$define INTERNAL_ERROR      5
$define PRAM_DOES_NOT_EXIST   -2
$define PRAM_IS_READ_ONLY  -2
$define PRAM_IS_WRITE_ONLY -3
$define PRAM_OUT_OF_RANGE  -5
$define IO_BOX_COMMS_ERROR -6

$define MAX_ANALOG          32767

} //Alias

Initial{...}

Background{...}
// line 105
```

Ready | Ln 49, Col 1 | INS

CTNet Remote I/O

Example: Read the very first analog input

DPL Editor - [CTNet0_Node_1.DPL (Read Analog Input Noncyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL (Read Analog Input Noncyclic)
    - Notes
    - Alias
    - **Initial**
    - Background

```
$AUTHOR   Jim Lynch, Jim Jeffers
$COMPANY  Control Techniques
$TITLE    Read Analog Input Noncyclic
$VERSION  V1.0.0
$DRIVE    SM-Applications

Notes{...}

Alias{...}

Initial{

input%    =  0
AnalogInputPercent% = 0
status% = NODE_IDLE
// following is to read parameter #3.00 on node 64
BKIOnode% = 64     // node 64 for this example
BKIOmenu% = 3      // parameter #3.00 menu value
BKIOparameter% = 00  // parameter #3.00 parameter value
timeout% = 100     // 100 ms

} //Initial

Background{...}
// line 105
```

Ready | Ln 70, Col 1 | INS

CTNet Remote I/O

Example: Read the very first analog input

DPL Editor - [CTNet0_Node_1.DPL (Read Analog Input Noncyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL (Read Analog Input Noncyclic)
    - Notes
    - Alias
    - Initial
    - **Background**

```
$AUTHOR   Jim Lynch, Jim Jeffers
$COMPANY  Control Techniques
$TITLE    Read Analog Input Noncyclic
$VERSION  V1.0.0
$DRIVE    SM-Applications

Notes{...}

Alias{...}

Initial{...}

Background{
top:

// request the contents of the first word of analog input
status% = RDNET(BKIOnode%, BKIOmenu%, BKIOparameter%, timeout%)
// becomes status% = RDNET(64, 3, 0, 100)

// note that 100 ms delay exists between the RDNET command and the
// check for the status.  This approach ok in an initial or in a
// background task.

if status% = COMMAND_COMPLETE then
   // after a 100 ms wait, the result should be available and valid
   input% = NETREPLY(0)
   // input% is a 32 bit integer, the lower 16 bits reflecting the value of
   // the Analog input.
   AnalogInputPercent% = input% *100 / MAX_ANALOG

endif

goto top: // main background loop
} //Background
// line 105
```

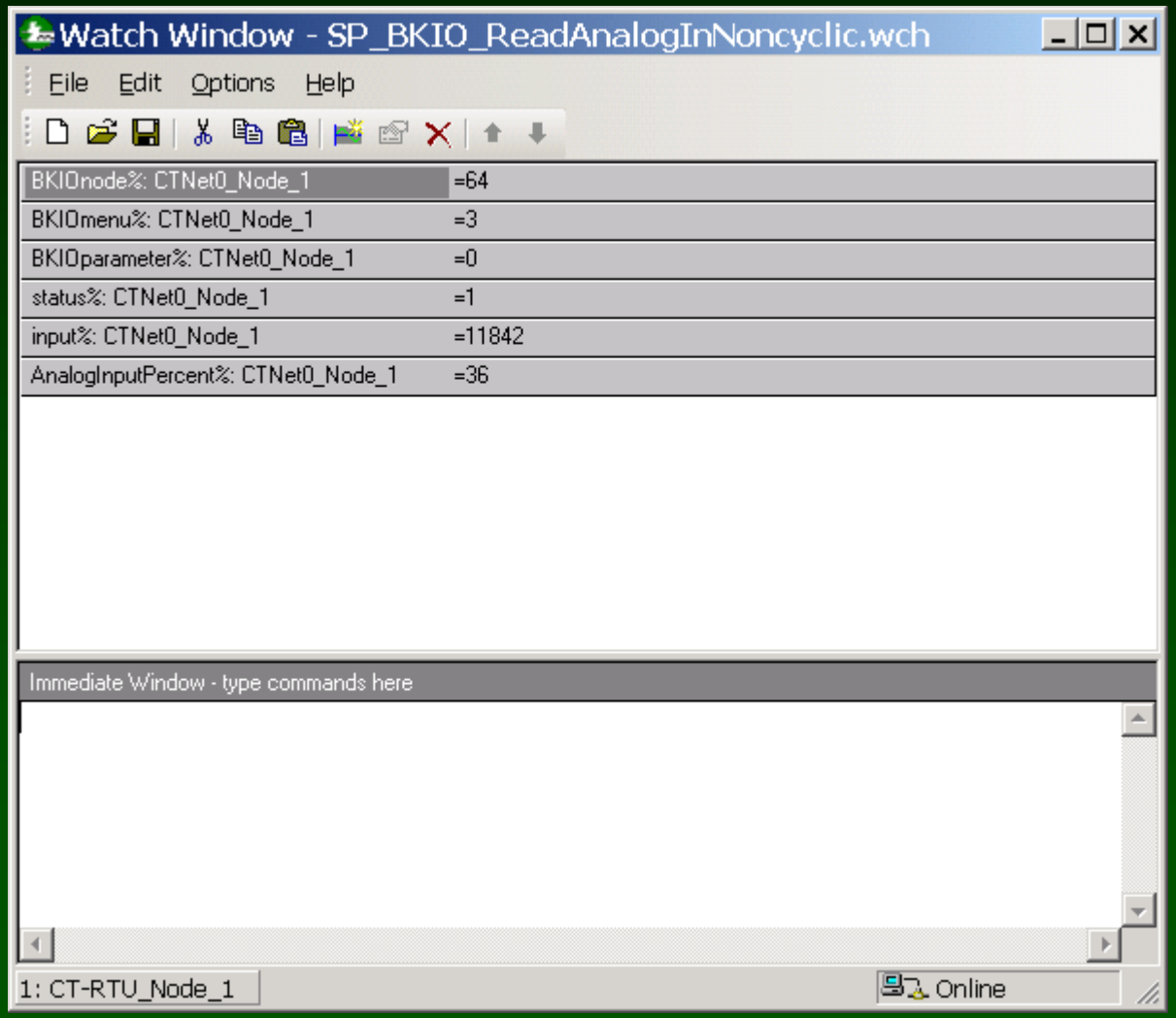
Ready | Ln 83, Col 1 | INS

CTNet I/O is at node 64

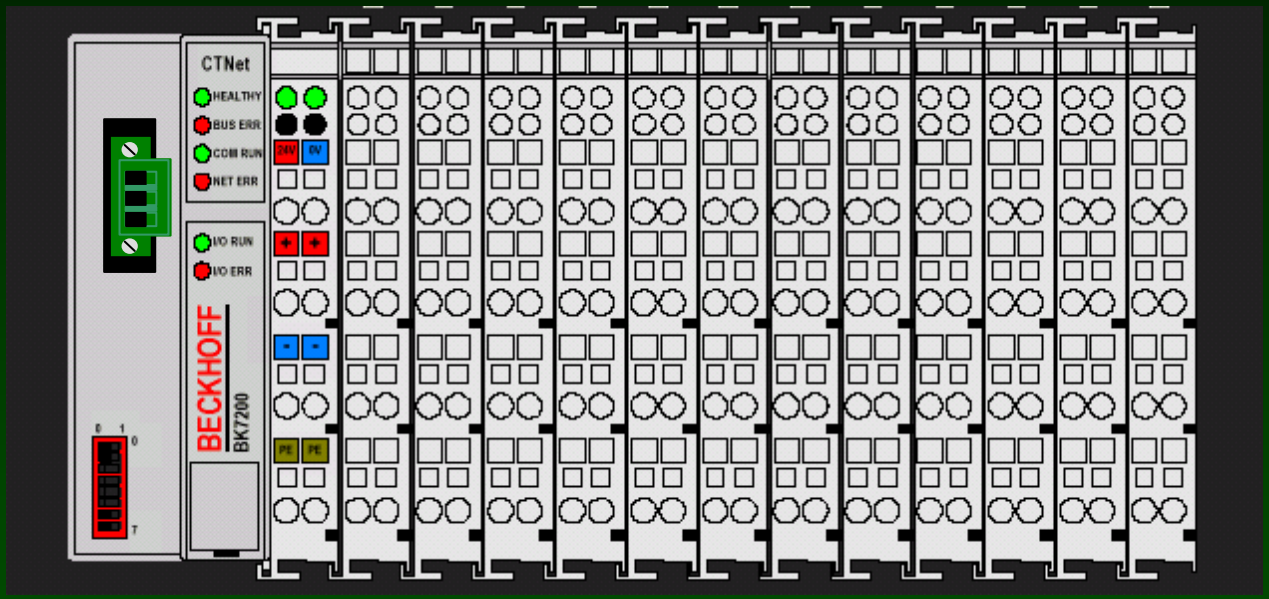
1st Analog Input = #03.00
2nd Analog Input = #03.01
etc.

CTNet Remote I/O

Watch window:
Read the first analog input (CTNet I/O is node 64)



CTNet Remote I/O



SyPT Pro Tutorial
Basic Analog Output handling
noncyclic data transfer

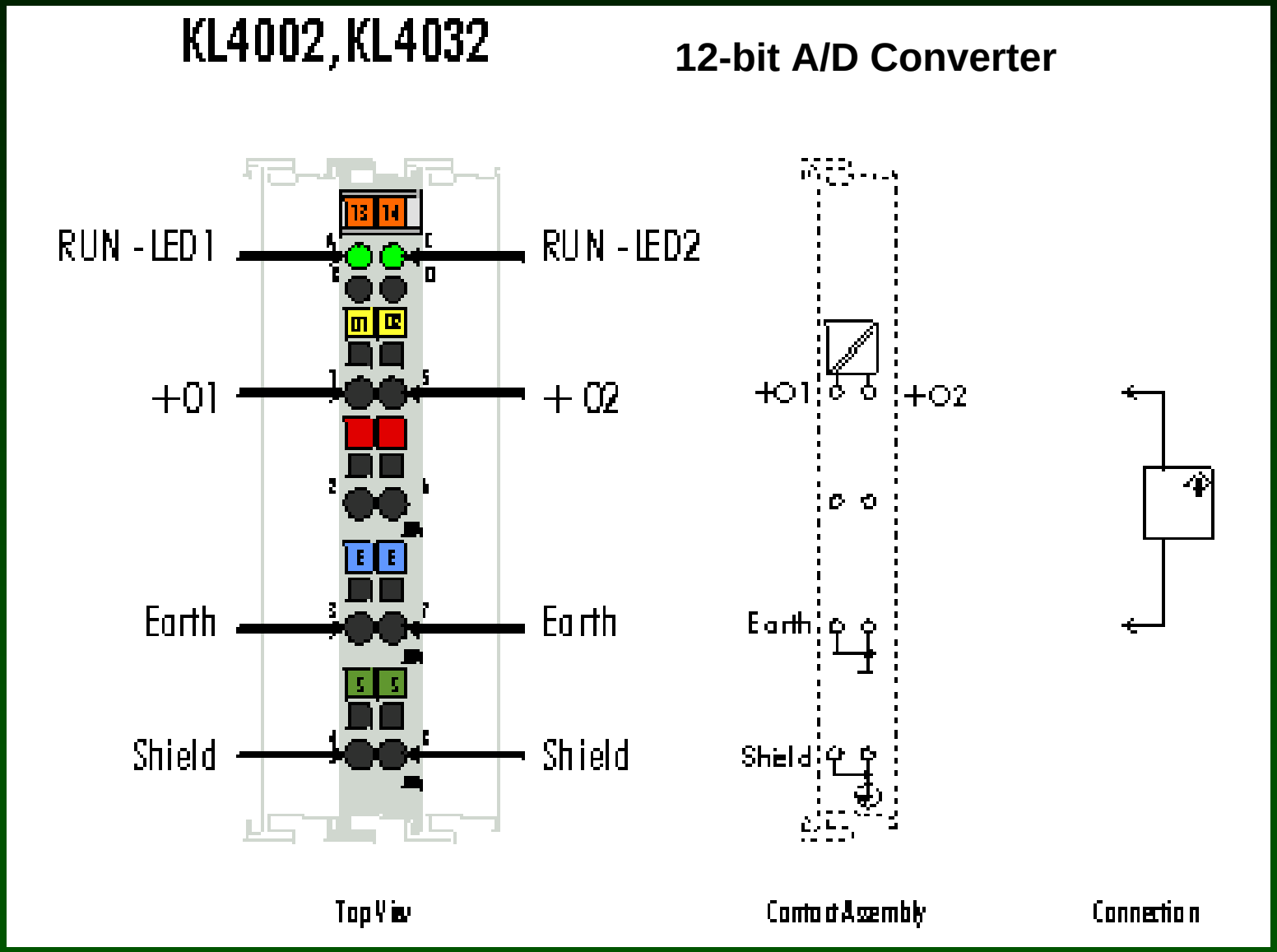
CTNet Remote I/O

A number of analog output modules are available.

Terminal	Inputs	Description
KL4002	2	0 -10V DC
KL4012	2	0-20mA
KL4022	2	4-20mA
KL4032	2	±10V DC

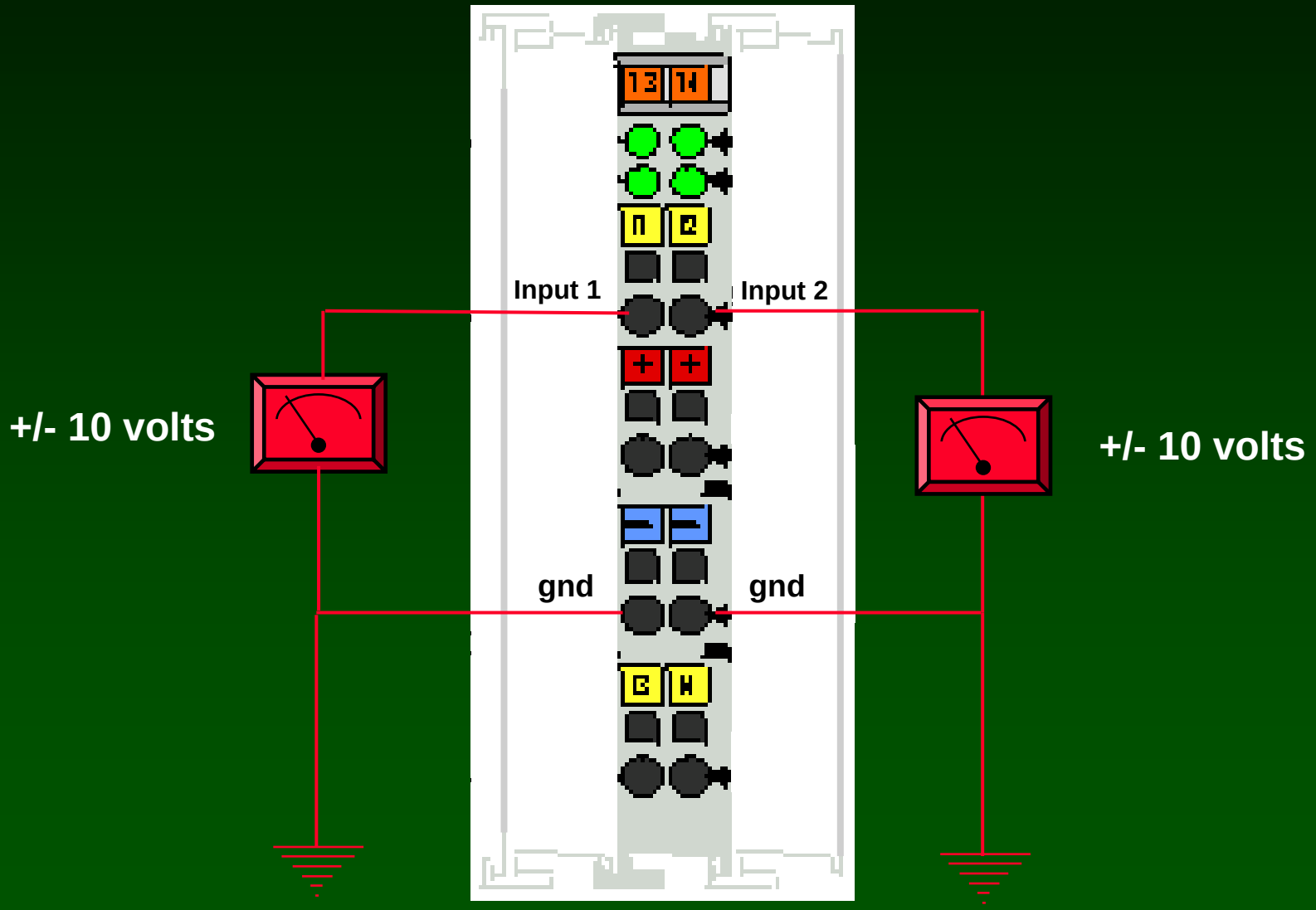
CTNet Remote I/O

Here's the Details on a 2-output Analog Output Module KL4032



CTNet Remote I/O

How to hook up a KL4032 Analog Output Module



CTNet Remote I/O

Analog Outputs are collected into Menu 4

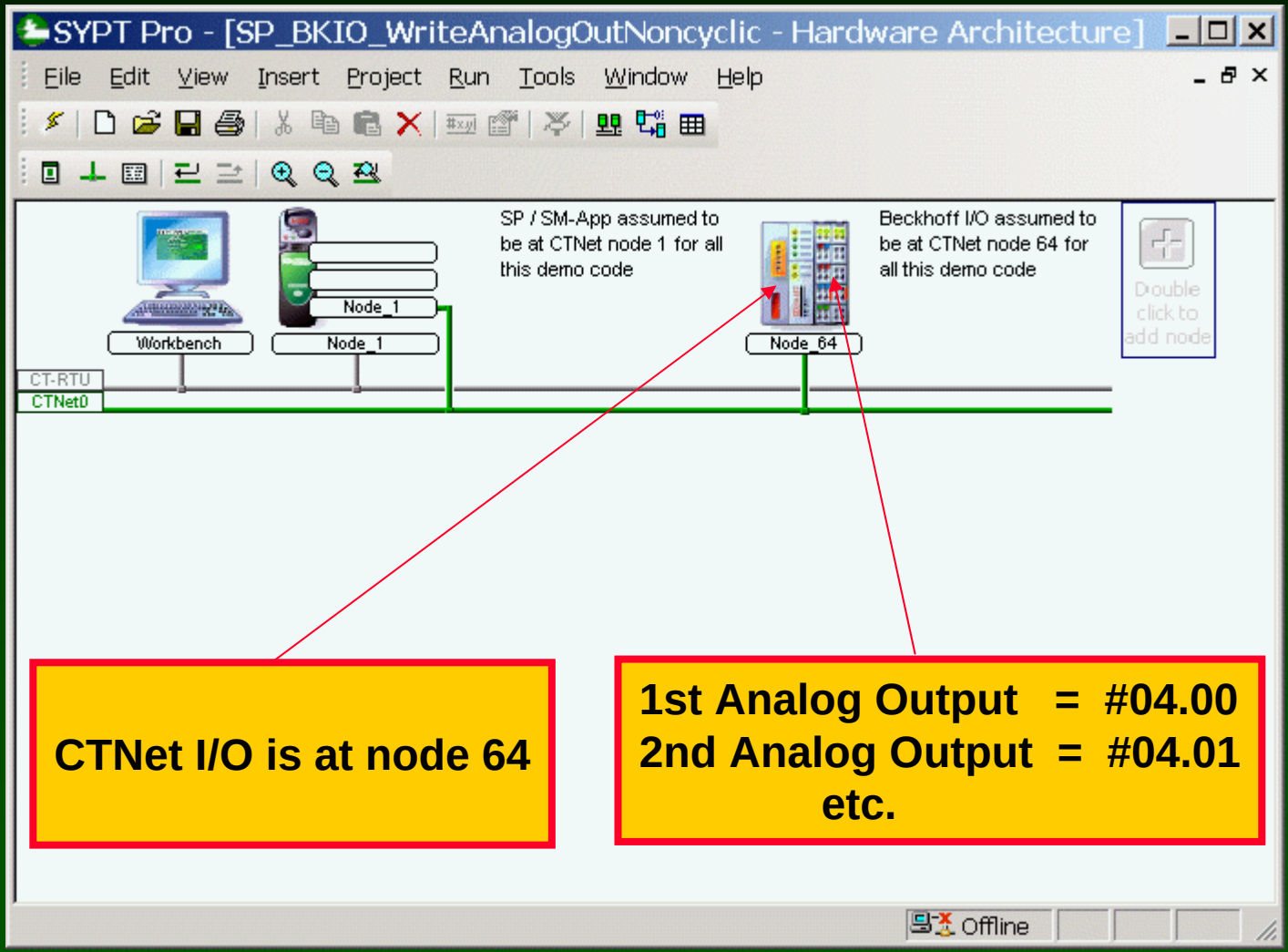
Reference	I/O Point
#4.00	0
#4.01	1
#4.02	2
#4.xx	xx
#4.98	98
#4.99	99

Analog Outputs are "scaled", 32767 is always full output.

Input Range	Minimum Input	Maximum Input	Resolution
0-10V	0V = 0	+10V = 32767	5mV
±10V	-10V = -32768	+10V = 32767	5mV
0-20mA	0mA = 0	20mA = 32767	5µA
4-20mA	4mA = 0	20mA = 32767	4µA

CTNet Remote I/O

Example: Write to the very first Analog Output



CTNet I/O is at node 64

1st Analog Output = #04.00
2nd Analog Output = #04.01
etc.

CTNet Remote I/O

Example: Write to the very first Analog Output

DPL Editor - [CTNet0_Node_1.DPL (Write Analog Output Noncyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL (Write Analog Output Noncyclic)
    - Notes
    - Alias
    - Initial
    - **Background**

```
$AUTHOR  Jim Lynch, Jim Jeffers
$COMPANY Control Techniques
$TITLE   Write Analog Output Noncyclic
$VERSION V1.0.0
$DRIVE   SM-Applications

Notes{...}

Alias{...}

Initial{...}

Background{...}
// line 122
```

Ready | Ln 86, Col 1 | INS

CTNet Remote I/O

Example: Write to the very first Analog Output

DPL Editor - [CTNet0_Node_1.DPL (Write Analog Output Noncyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL (Write Analog Output Noncyclic)
    - Notes
    - **Alias**
    - Initial
    - Background

```
$AUTHOR  Jim Lynch, Jim Jeffers
$COMPANY Control Techniques
$TITLE   Write Analog Output Noncyclic
$VERSION V1.0.0
$DRIVE   SM-Applications

Notes{...}

Alias{
// No boolean data type is provided
// by Drive Programming Language
$define ON      1
$define OFF     0
// CTNet RDNET and WRNET status codes
$define NODE_IDLE        0
$define COMMAND_COMPLETE    1
$define NODE_BUSY        3
$define NODE_TIMEOUT     4
$define INTERNAL_ERROR      5
$define PRAM_DOES_NOT_EXIST   -2
$define PRAM_IS_READ_ONLY  -2
$define PRAM_IS_WRITE_ONLY -3
$define PRAM_OUT_OF_RANGE  -5
$define IO_BOX_COMMS_ERROR -6

$define POS_MAX_ANALOG      32737
$define NEG_MAX_ANALOG      -32737
$define DELTA_LARGE         500
$define DELTA_SMALL         10

} //Alias

Initial{...}

Background{...}
// line 122
```

Ready | Ln 45, Col 1 | INS

CTNet Remote I/O

Example: Write to the very first Analog Output

DPL Editor - [CTNet0_Node_1.DPL (Write Analog Output Noncyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL (Write Analog Output Noncyclic)
    - Notes
    - Alias
    - **Initial**
    - Background

```
$AUTHOR   Jim Lynch, Jim Jeffers
$COMPANY  Control Techniques
$TITLE    Write Analog Output Noncyclic
$VERSION  V1.0.0
$DRIVE    SM-Applications

Notes{...}

Alias{...}

Initial{

bRampUp% = ON
bRampDown% = OFF

OutValue% = 0

status% = NODE_IDLE
// following is to write to parameter #4.00 on node 64
BKIOnode% = 64     // node 64 for this example
BKIOmenu% = 4      // parameter #4.00 menu value
BKIOparameter% = 00  // parameter #4.00 parameter value
dpos% = 0          // descimal point position zero, x1 format
timeout% = 100     // 100 ms

} //Initial

Background{...}
// line 122
```

Ready | Ln 69, Col 1 | INS

CTNet Remote I/O

Example: Write to the very first Analog Output

DPL Editor - [CTNet0_Node_1.DPL (Write Analog Output Noncyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL (Write Analog Output Noncyclic)
    - Notes
    - Alias
    - Initial
    - **Background**

```
Background{
top:

// create the ramping value
IF bRampUp% = ON THEN
   if OutValue% > POS_MAX_ANALOG then
      bRampUp% = OFF
      bRampDown% = ON
      OutValue% = POS_MAX_ANALOG
   elseif OutValue% > (POS_MAX_ANALOG * 98 / 100) then
      OutValue% = OutValue% + DELTA_SMALL
   else
      OutValue% = OutValue% + DELTA_LARGE
   endif
ELSE
   if OutValue% < NEG_MAX_ANALOG then
      bRampUp% = ON
      bRampDown% = OFF
      OutValue% = NEG_MAX_ANALOG
   elseif OutValue% < (NEG_MAX_ANALOG * 98 / 100) then
      OutValue% = OutValue% - DELTA_SMALL
   else
      OutValue% = OutValue% - DELTA_LARGE
   endif
ENDIF

// write to the Beckhoff I/O
status% = WRNET(BKIOnode%, BKIOmenu%, BKIOparameter%, OutValue%, dpos%, timeout%)
// becomes status% = WRNET(64, 4, 0, value%, 0, 100)

// note that 100 ms automatic timeout is being used with the WRNET
// command here.  This means that the program will stop here and wait up to
// 100 ms for a response from node 64 before continuing.  This approach ok
// in an initial or in a background task.

DELAY (1)
// the program stops for an additional 0.1 second wait here.  This has
// the effect of controlling how often the bit changes state and the
// update rate to the Beckhoff I/O.

goto top: // main background loop
} //Background
// line 122
```

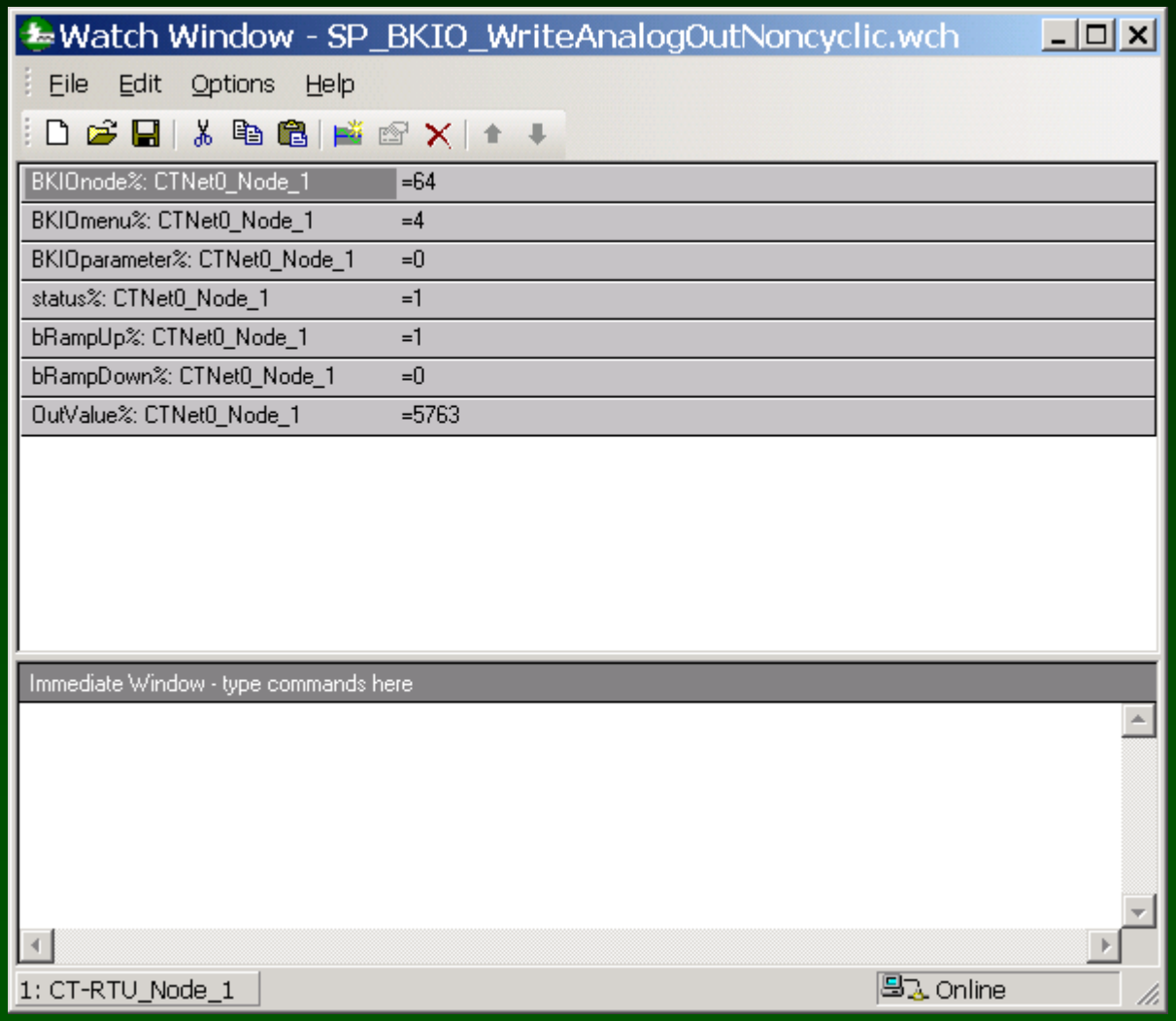
Ready | Ln 86, Col 1 | INS

CTNet I/O is at node 64

1st Analog Output = #04.00
2nd Analog Output = #04.01
etc.

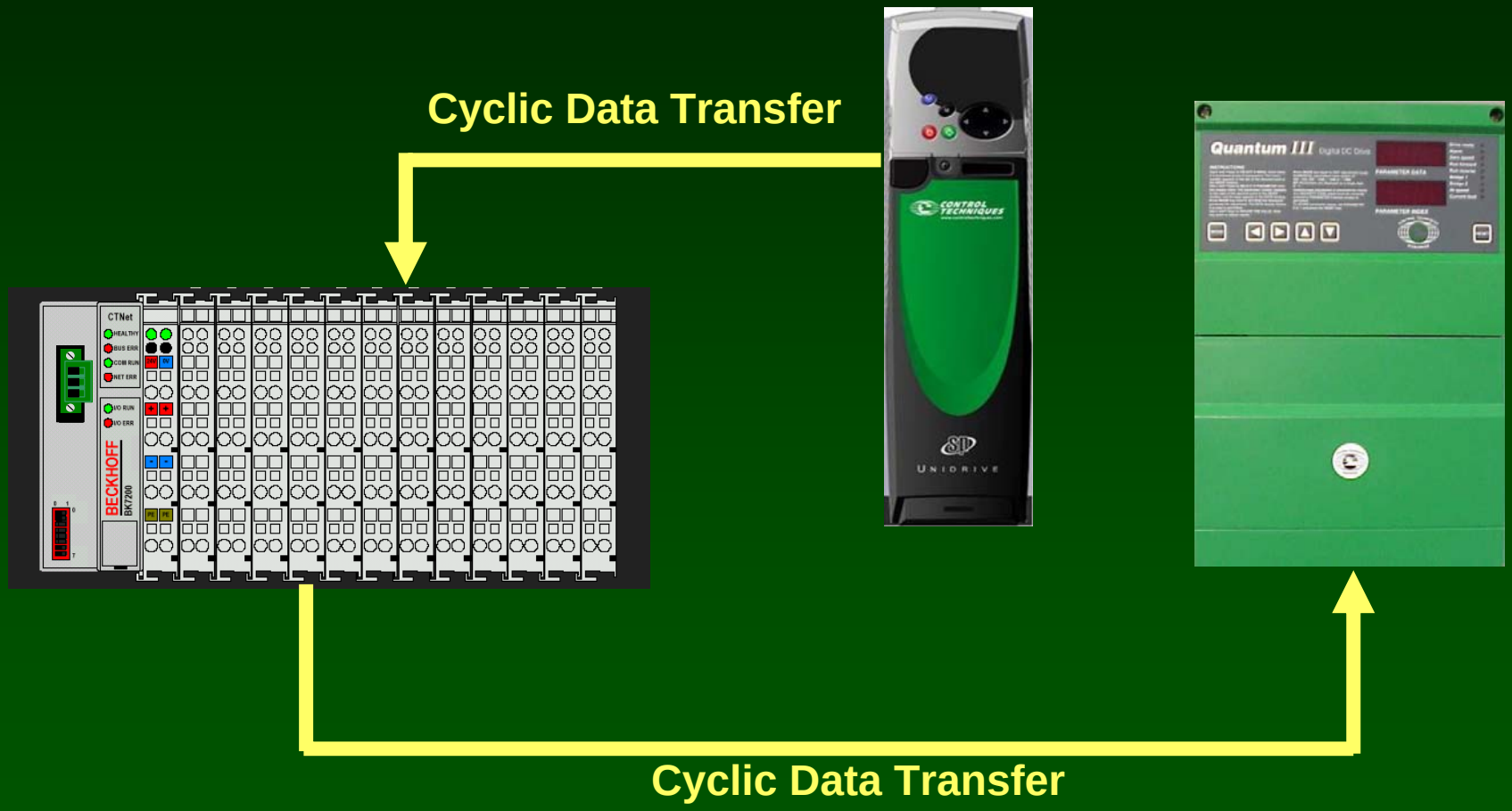
CTNet Remote I/O

Watch Window:
Write to the very first Analog Output



CTNet Remote I/O

CTNet / Beckhoff Remote I/O System supports Cyclic Data Transfers !



Why use Cyclic Data Transfers ?

- FAST
- EFFICIENT
- NO PROGRAMMER INTERVENTION

Note: this can only be set up with the SyPT Pro !

CTNet Remote I/O

Set up the Uni-SP to generate the Cyclic Data synchronizing message. SM-Apps is in slot 3

#17.11	= 10	Clock task set at 10 ms
#17.22	= 0	CTNet token ring ID
#17.23	= 1	node address
#17.24	= 2	baud rate = 2.5 Mbits
#17.25	= 510	FAST cyclic rate = 10 msec SLOW cyclic rate = 50 msec
#17.00	= 1070 - reset	

CTNet Remote I/O

SyPT Pro Example: Let's use Cyclic Data to read a digital Input

Strategy: CTNet Remote I/O is at node address 64

Four Digital inputs are configured

Digital Inputs are at parameter #01.00 on node 64

Digital Inputs are "packed" into a "collection" by the BKIO

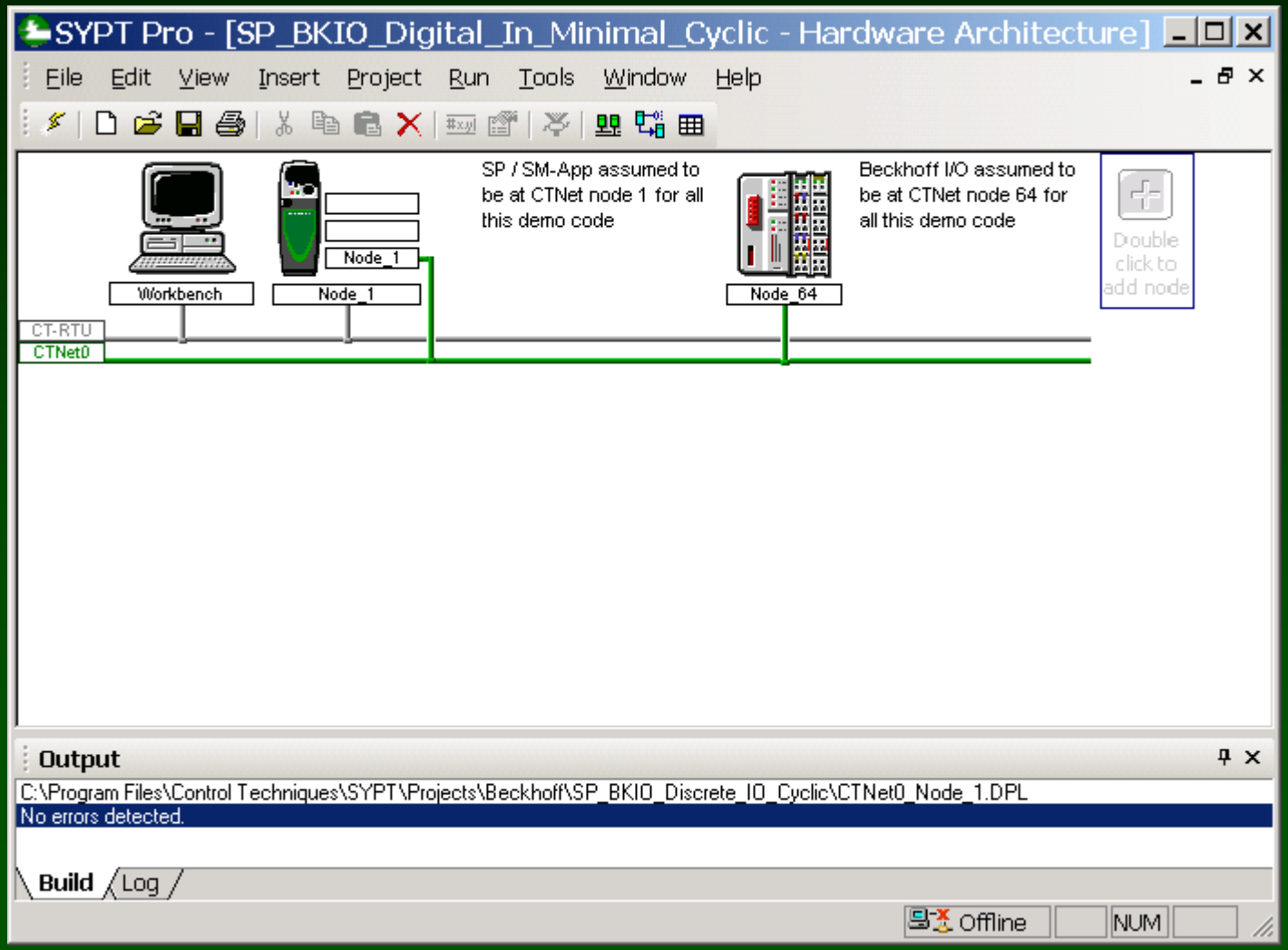
We will send digital inputs , via cyclic data, from node 64 parameter #01.00 to node 1, parameter #73.10 (_S10%), as a "collection"

The SP / SM-Apps at node 1 will generate the Cyclic Data synchronizing message at 10 msec

CTNet Remote I/O

Example: project SP_BKIO_Digital_IN_Minimal_Cyclic

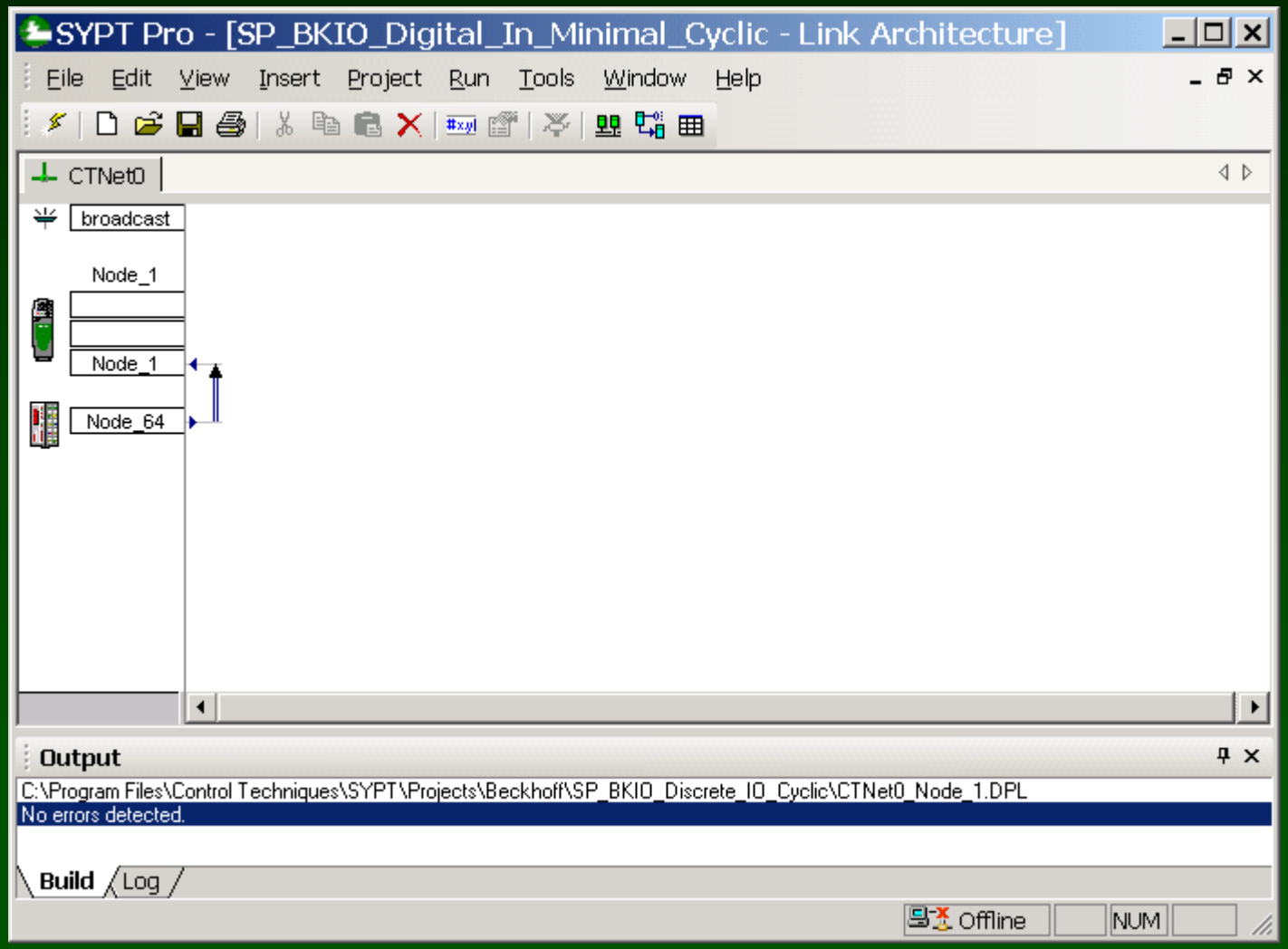
Example: Minimal Digital Input cyclic version (CTNet I/O is node 64)



CTNet Remote I/O

Example: project SP_BKIO_Digital_IN_Minimal_Cyclic

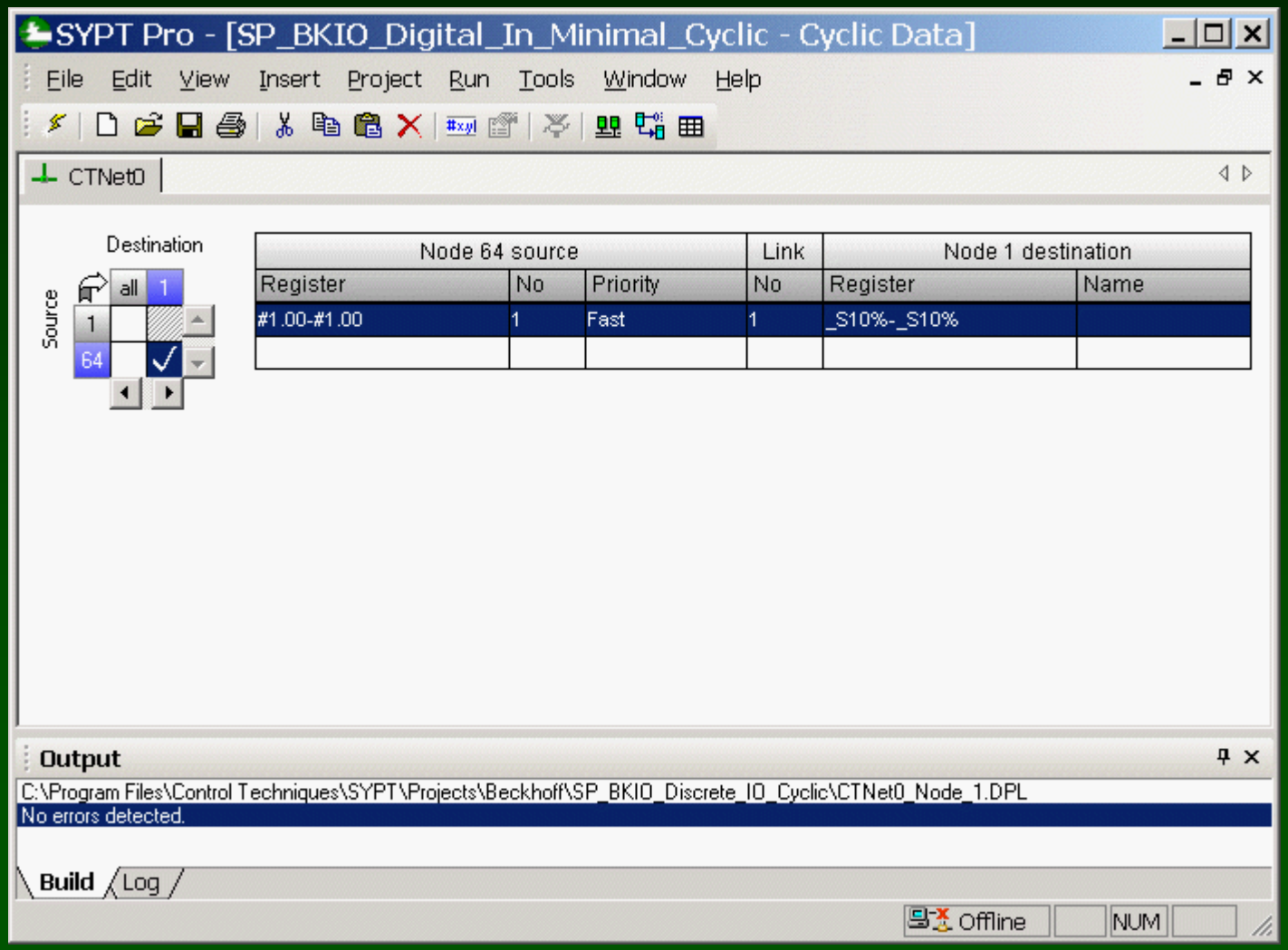
Set up a cyclic link from the Beckhoff to the Unidrive.



CTNet Remote I/O

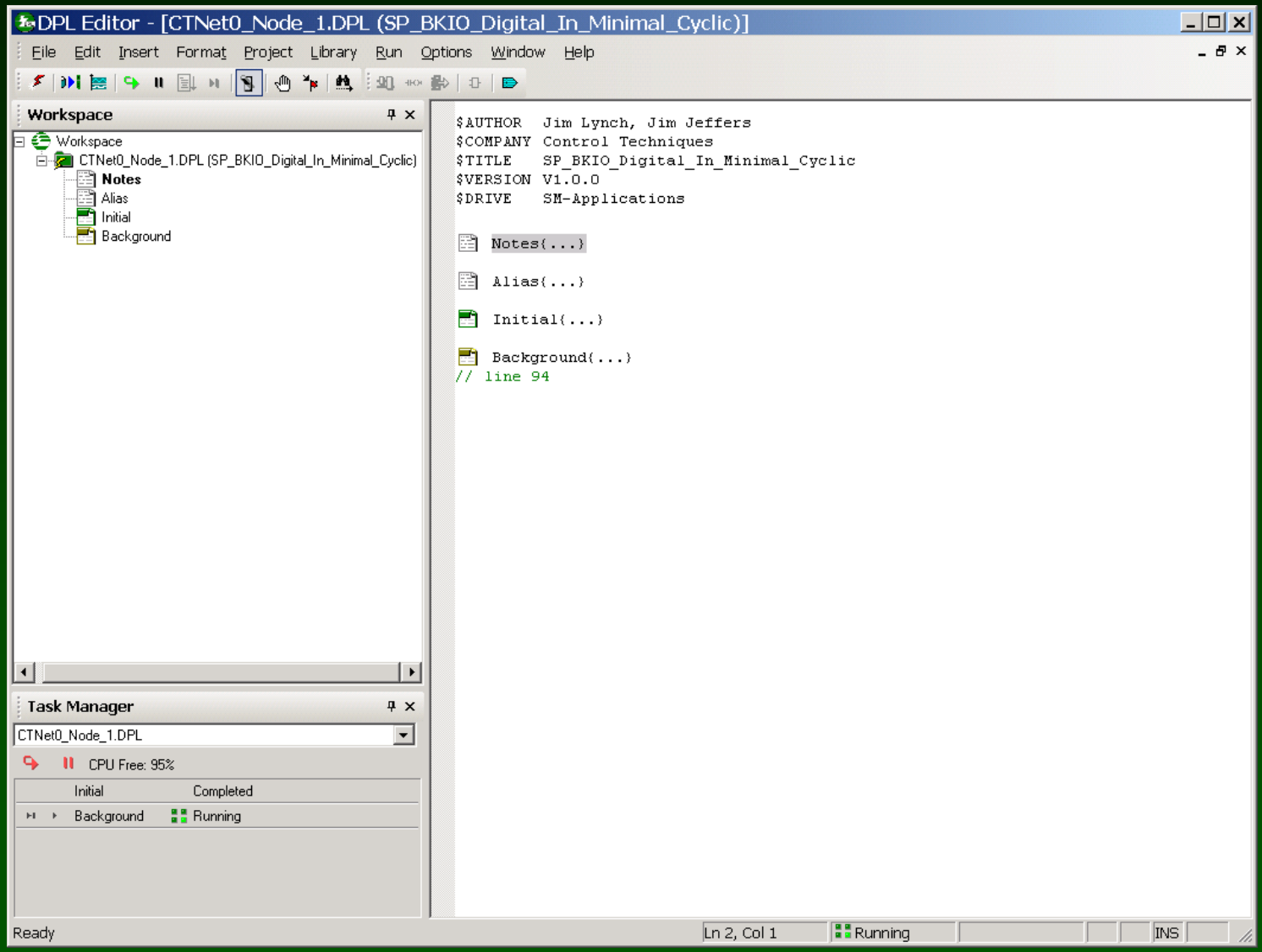
Example: project SP_BKIO_Digital_IN_Minimal_Cyclic

Set up a cyclic link from the Beckhoff to the Unidrive.



CTNet Remote I/O

Open / create the Following DPL program on Unidrive, Node 1



CTNet Remote I/O

Open / create the Following DPL program on Unidrive, Node 1

DPL Editor - [CTNet0_Node_1.DPL (SP_BKIO_Digital_In_Minimal_Cyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL (SP_BKIO_Digital_In_Minimal_Cyclic)
    - Notes
    - **Alias**
    - Initial
    - Background

```
$AUTHOR   Jim Lynch, Jim Jeffers
$COMPANY  Control Techniques
$TITLE    SP_BKIO_Digital_In_Minimal_Cyclic
$VERSION  V1.0.0
$DRIVE    SM-Applications

Notes{...}

Alias{
// No boolean data type is provided
// by Drive Programming Language
$define ON      1
$define OFF     0
$define TRUE    1
$define FALSE   0

// this mask is designed for the four
// least significant bits being "active"
$define INPUT_MASK          0x0000000F


} //Alias

Initial{...}

Background{...}
// line 94
```

Task Manager

CTNet0_Node_1.DPL

CPU Free: 95%

Initial	Completed
Background	Running

Ready | Ln 23, Col 1 | Running | INS

CTNet Remote I/O

Open / create the Following DPL program on Unidrive, Node 1

DPL Editor - [CTNet0_Node_1.DPL (SP_BKIO_Digital_In_Minimal_Cyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL (SP_BKIO_Digital_In_Minimal_Cyclic)
    - Notes
    - Alias
    - **Initial**
    - Background

```
$AUTHOR   Jim Lynch, Jim Jeffers
$COMPANY  Control Techniques
$TITLE    SP_BKIO_Digital_In_Minimal_Cyclic
$VERSION  V1.0.0
$DRIVE    SM-Applications

Notes{...}

Alias{...}

Initial{

bit0% = OFF
bit1% = OFF
bit2% = OFF
bit3% = OFF

InImage% = 0
LastInImage% = InImage%


bAllInOff% = TRUE
bInChanged% = FALSE


} //Initial

Background{...}
// line 94
```

Task Manager

CTNet0_Node_1.DPL

CPU Free: 95%

Initial	Completed
Background	Running

Ready | Ln 38, Col 1 | Running | INS

CTNet Remote I/O

Open / create the Following DPL program on Unidrive, Node 1

DPL Editor - [CTNet0_Node_1.DPL (SP_BKIO_Digital_In_Minimal_Cyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL (SP_BKIO_Digital_In_Minimal_Cyclic)
    - Notes
    - Alias
    - Initial
    - **Background**

Task Manager

CTNet0_Node_1.DPL

CPU Free: 95%

Initial	Completed
Background	Running

```
Notes{...}

Alias{...}

Initial{...}

Background{
top:

// INPUT - Get the inputs
// Note that the cyclic link from the BK IO (node 64) to the
// SM-Apps (node 1) does the actual work here, instead of a
// ReadNet instruction

   InImage% = _S10% & INPUT_MASK

// PROCESS - Solve the inputs
   // detect Input changes
   IF InImage% = LastInImage% THEN
      bInChanged% = FALSE
   ELSE
      bInChanged% = TRUE
   // placeholder
   ENDIF
   LastInImage% = InImage% // Update

   // detect the "all inputs not active" condition
   IF InImage% = 0 THEN
      bAllInOff% = TRUE
   ELSE
      bAllInOff% = FALSE
   ENDIF

   // decode the individual bits from the collection
   bit0% = InImage%.0   // DPL bit manipulation operation illustrated
   bit1% = InImage%.1   // showing one way to un-pack the word into
   bit2% = InImage%.2   // the individual bits
   bit3% = InImage%.3

// OUTPUT - Write the outputs
// Trivial in this demo program.  One uses the watch window for viewing
// this program's activity

goto top: // main background loop
} //Background
// line 94
```

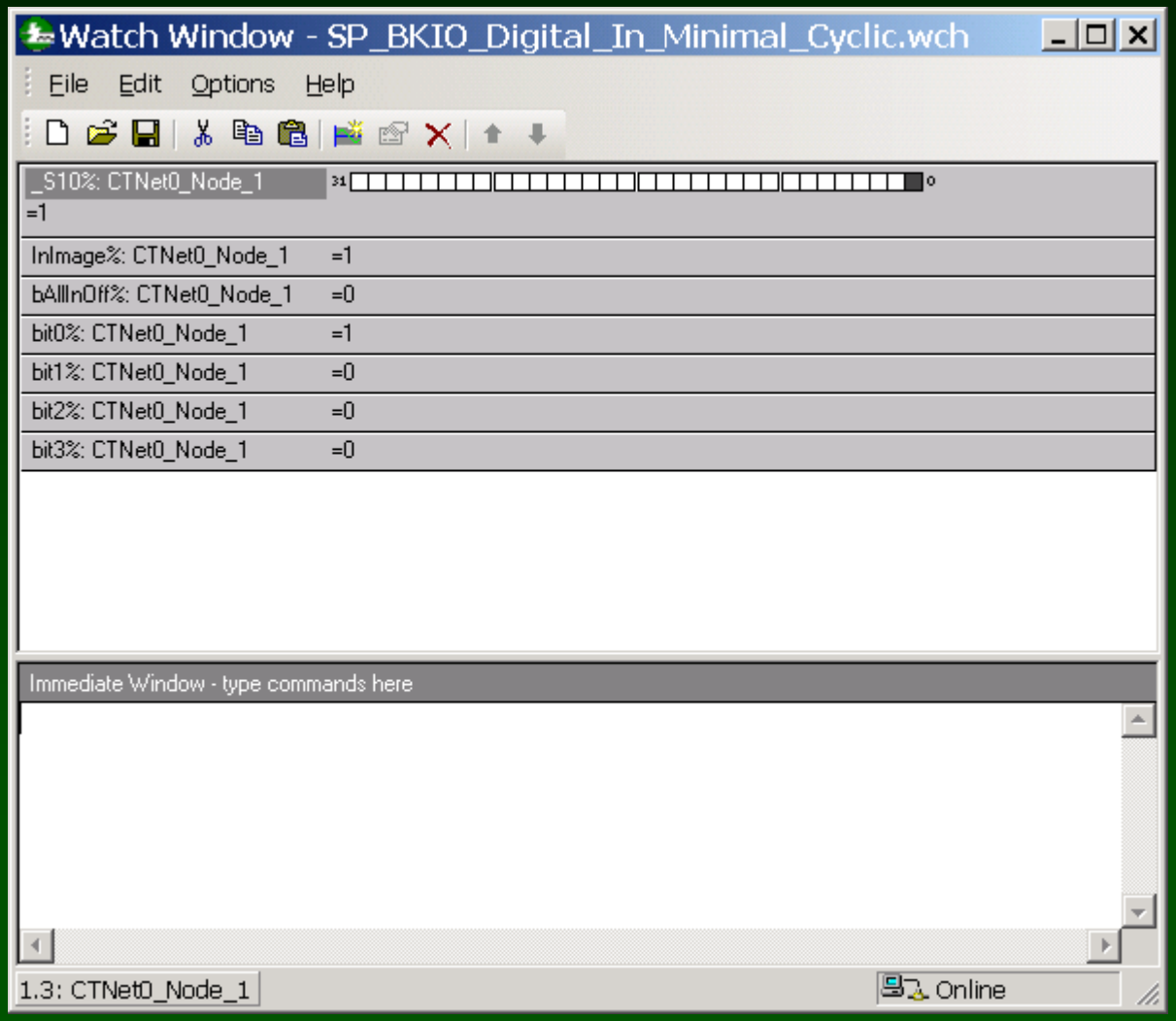
Ready | Ln 55, Col 1 | Running | INS

The I/O bits come in automatically to PLC register _S10%.

Here we just extract the bits.

CTNet Remote I/O

Watch Window:
Open / create the Following DPL program on Unidrive, Node 1



CTNet Remote I/O

SyPT Pro Example: Let's use Cyclic Data to write a digital output

Strategy: CTNet Remote I/O is at node address 64

Four Digital outputs are configured

Digital Outputs are at parameter #02.00 on node 64

Digital Outputs are "unpacked" from a "collection" by the BKIO

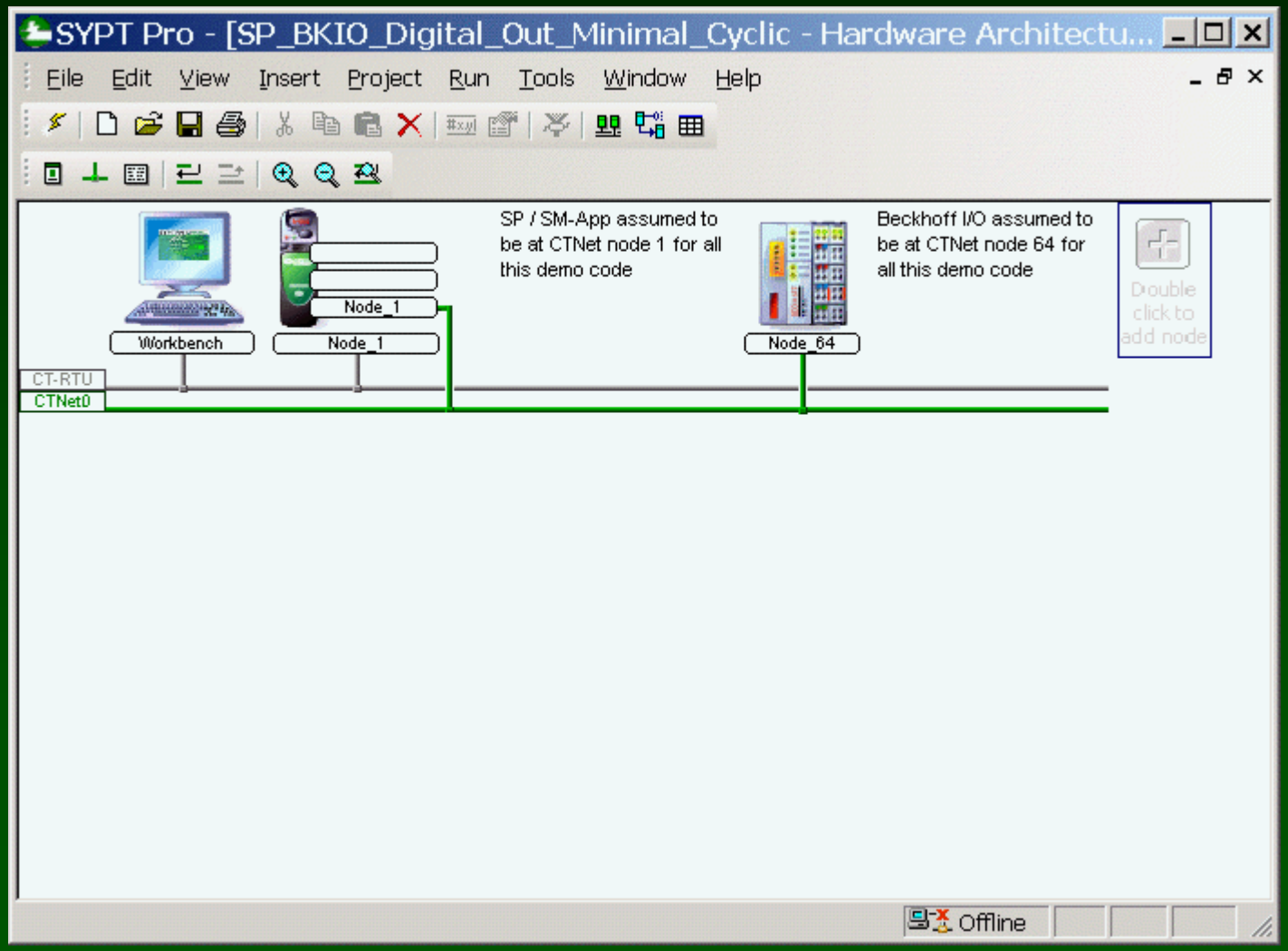
We will send digital outputs , via cyclic data, from node 1 parameter #72.10 (_R10%) to node 64 parameter #02.00, as a "collection"

The SP / SM-Apps at node 1 will generate the Cyclic Data synchronizing message at 10 msec

CTNet Remote I/O

Example: project SP_BKIO_Digital_OUT_Minimal_Cyclic

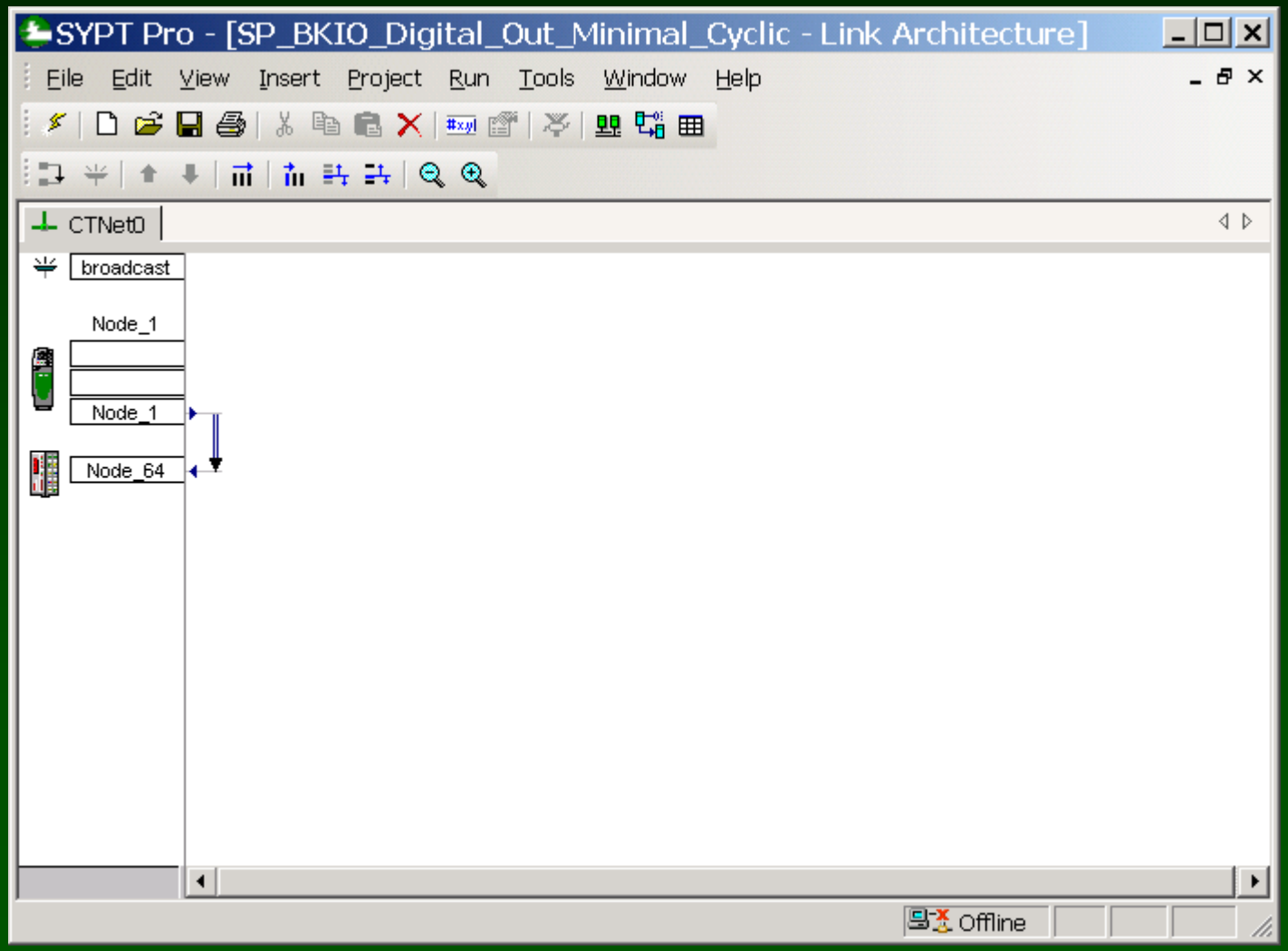
Example: Minimal Digital Output cyclic version (CTNet I/O is node 64)



CTNet Remote I/O

Example: project SP_BKIO_Digital_OUT_Minimal_Cyclic

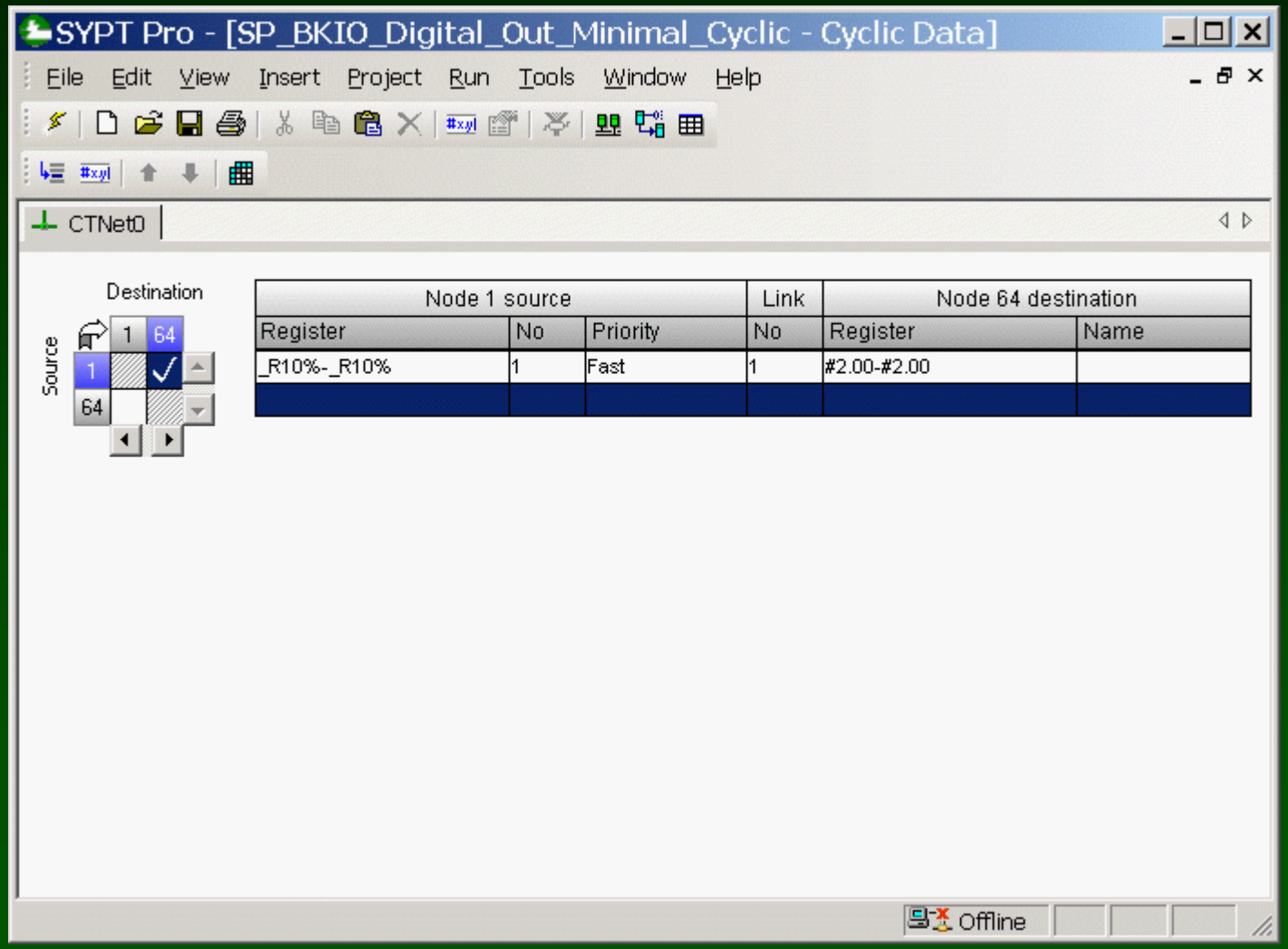
Set up a cyclic link from the Unidrive to the Beckhoff.



CTNet Remote I/O

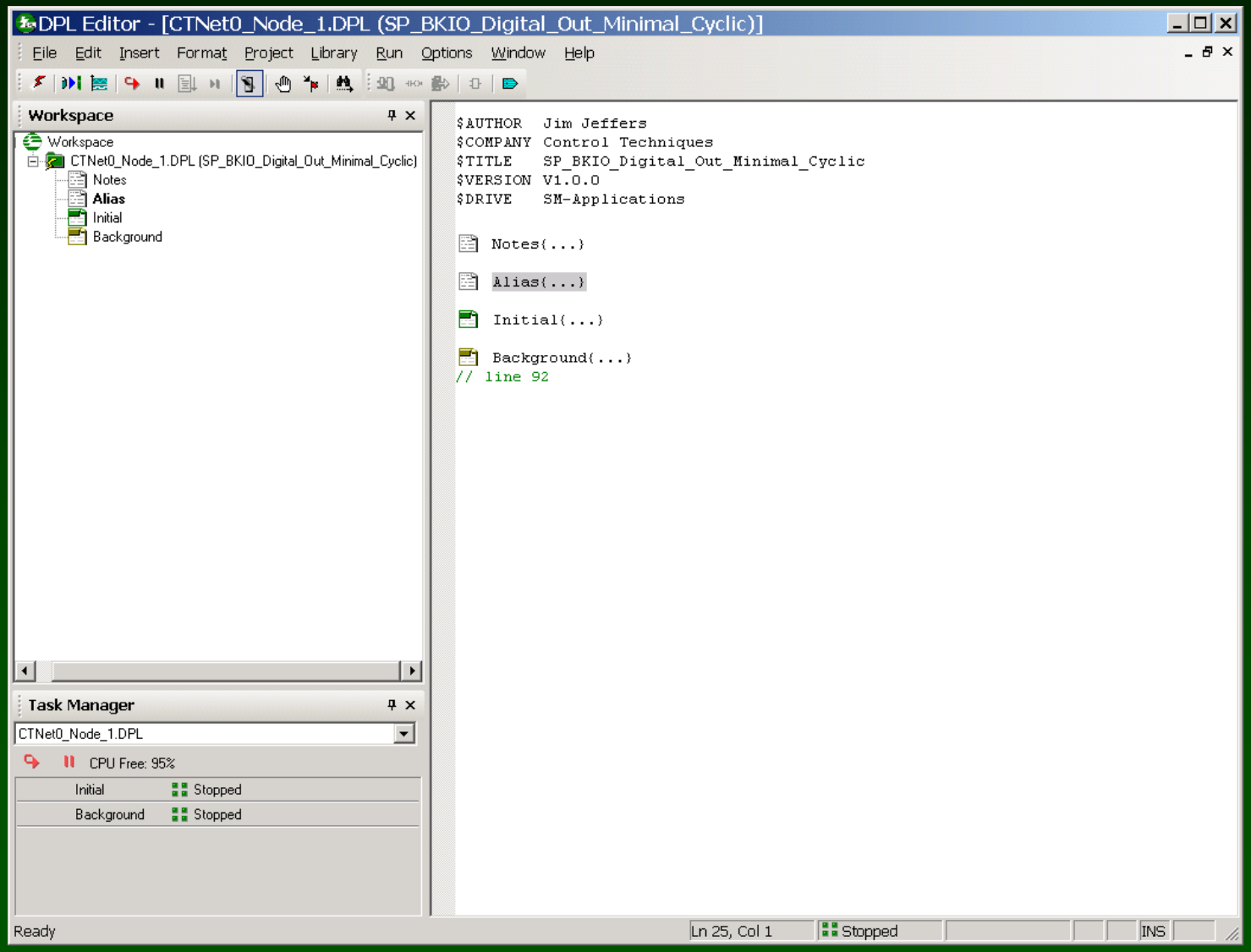
Example: project SP_BKIO_Digital_OUT_Minimal_Cyclic

Set up a cyclic link from the Unidrive to the Beckhoff.



CTNet Remote I/O

Open / create the Following DPL program on Unidrive, Node 1



CTNet Remote I/O

Open / create the Following DPL program on Unidrive, Node 1

DPL Editor - [CTNet0_Node_1.DPL (SP_BKIO_Digital_Out_Minimal_Cyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL (SP_BKIO_Digital_Out_Minimal_Cyclic)
    - Notes
    - Alias
    - Initial
    - **Background**

Task Manager

CTNet0_Node_1.DPL

CPU Free: 95%

Initial	Stopped
Background	Stopped

```
$AUTHOR   Jim Jeffers
$COMPANY  Control Techniques
$TITLE    SP_BKIO_Digital_Out_Minimal_Cyclic
$VERSION  V1.0.0
$DRIVE    SM-Applications

Notes{...}

Alias{
// No boolean data type is provided
// by Drive Programming Language
$define ON      1
$define OFF     0
$define TRUE    1
$define FALSE   0

// these masks are designed for the four
// least significant bits being "active"
$define INPUT_MASK          0x0000000F
$define OUTPUT_MASK         0x000F0000
$define BIT_1            1
$define BIT_2            2
$define BIT_3            3
$define BIT_4            4

$define IDLE_STATE       2

} //Alias

Initial{...}

Background{...}
// line 92
```

Ready | Ln 54, Col 1 | Stopped | INS

CTNet Remote I/O

Open / create the Following DPL program on Unidrive, Node 1

DPL Editor - [CTNet0_Node_1.DPL (SP_BKIO_Digital_Out_Minimal_Cyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL (SP_BKIO_Digital_Out_Minimal_Cyclic)
    - Notes
    - Alias
    - Initial
    - **Background**

```
$AUTHOR   Jim Jeffers
$COMPANY  Control Techniques
$TITLE    SP_BKIO_Digital_Out_Minimal_Cyclic
$VERSION  V1.0.0
$DRIVE    SM-Applications

Notes{...}

Alias{...}

Initial{

OutImage% = 0
stateNow% = IDLE_STATE
RotateBitNow%  = BIT_1

} //Initial

Background{...}
// line 92
```

Task Manager

CTNet0_Node_1.DPL

CPU Free: 95%

Initial	Stopped
Background	Stopped

Ready | Ln 54, Col 1 | Stopped | INS

CTNet Remote I/O

Open / create the Following DPL program on Unidrive, Node 1

DPL Editor - [CTNet0_Node_1.DPL (SP_BKIO_Digital_Out_Minimal_Cyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL (SP_BKIO_Digital_Out_Minimal_Cyclic)
    - Notes
    - Alias
    - Initial
    - **Background**

```
Notes{...}

Alias{...}

Initial{...}

Background{
top:

// INPUT - Get the inputs
   // Placeholder inputs not used for this trivial example
   // InImage% = _S10% & INPUT_MASK

// PROCESS - Solve the inputs
   stateNow% = IDLE_STATE   // this logic has a trival single state

   // Generate a new output word "image"
   IF stateNow% = IDLE_STATE THEN
      SELECT RotateBitNow%
         CASE BIT_1
            OutImage% = 0x00000001 | OUTPUT_MASK
            RotateBitNow% = BIT_2
         CASE BIT_2
            OutImage% = 0x00000002 | OUTPUT_MASK
            RotateBitNow% = BIT_3
         CASE BIT_3
            OutImage% = 0x00000004 | OUTPUT_MASK
            RotateBitNow% = BIT_4
         CASE BIT_4
            OutImage% = 0x00000008 | OUTPUT_MASK
            RotateBitNow% = BIT_1
      ENDSELECT
      // the following works but is not practical in a production app.
      // a later, more practical program issustrates a better approach/
      DELAY (5)   // pause before next pass
// ELSE  // in a different state not defined or used this rev
//   situation considered, no code required this rev
   ENDIF

// OUTPUT - Write the outputs
   _R10% = OutImage%

goto top: // main background loop
} //Background
// line 92
```

Task Manager

CTNet0_Node_1.DPL

CPU Free: 95%

Initial	Completed
Background	Running

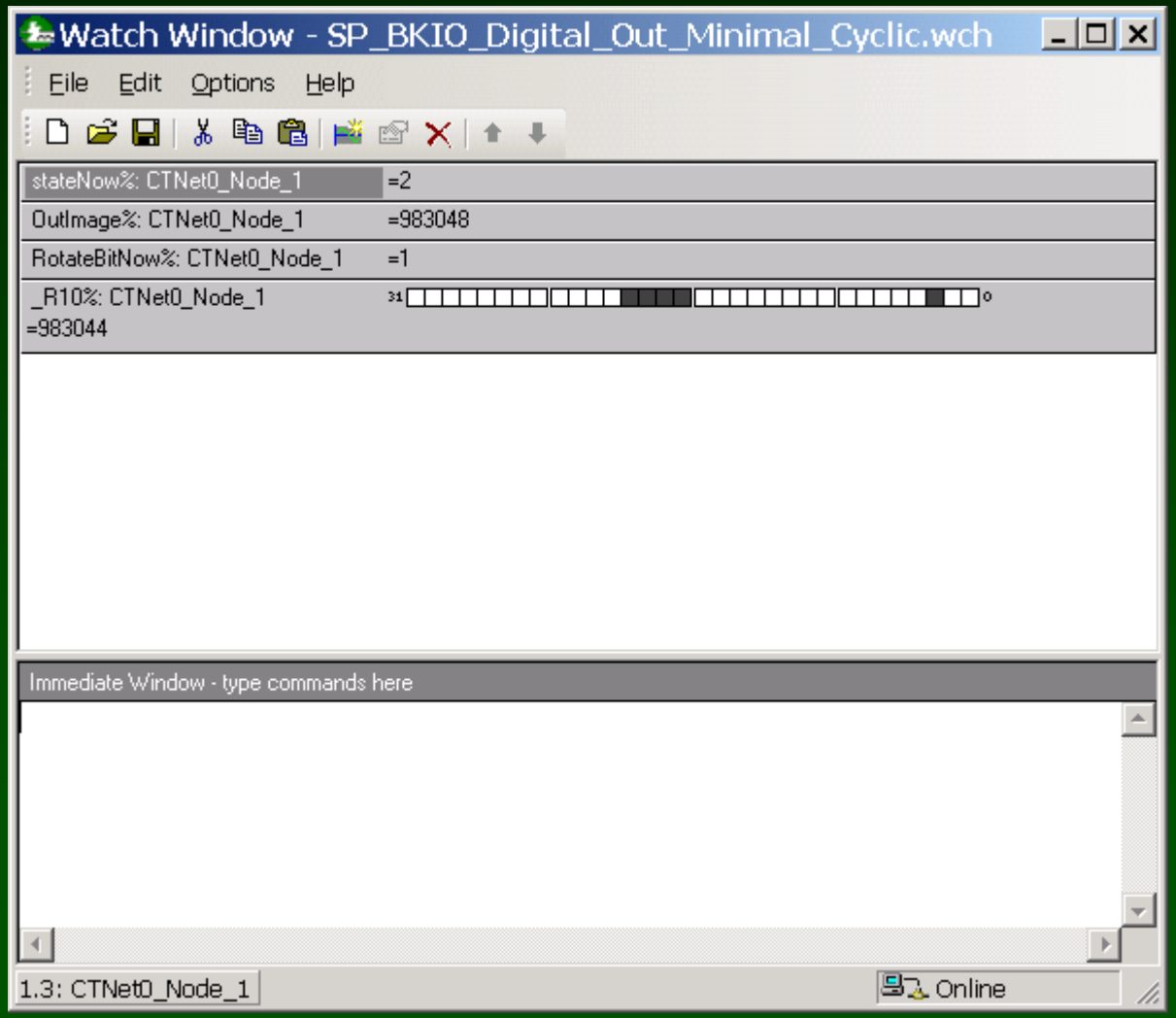
Ready | Ln 54, Col 1 | Running | INS

Here we select the output bit pattern and combine this with the mask bits.

The I/O bits automatically go out from PLC register _R10% as a "collection".

CTNet Remote I/O

Watch Window:
Open / create the Following DPL program on Unidrive, Node 1



CTNet Remote I/O

SyPT Pro Example: Let's use Cyclic Data to read a analog input

Strategy: CTNet Remote I/O is at node address 64

Two Analog inputs are available, we will read both

Analog Inputs start at parameter #03.00 on node 64

Analog Inputs consume one 32-bit word each on the BKIO

Analog Inputs are signed quantities (two's complement)

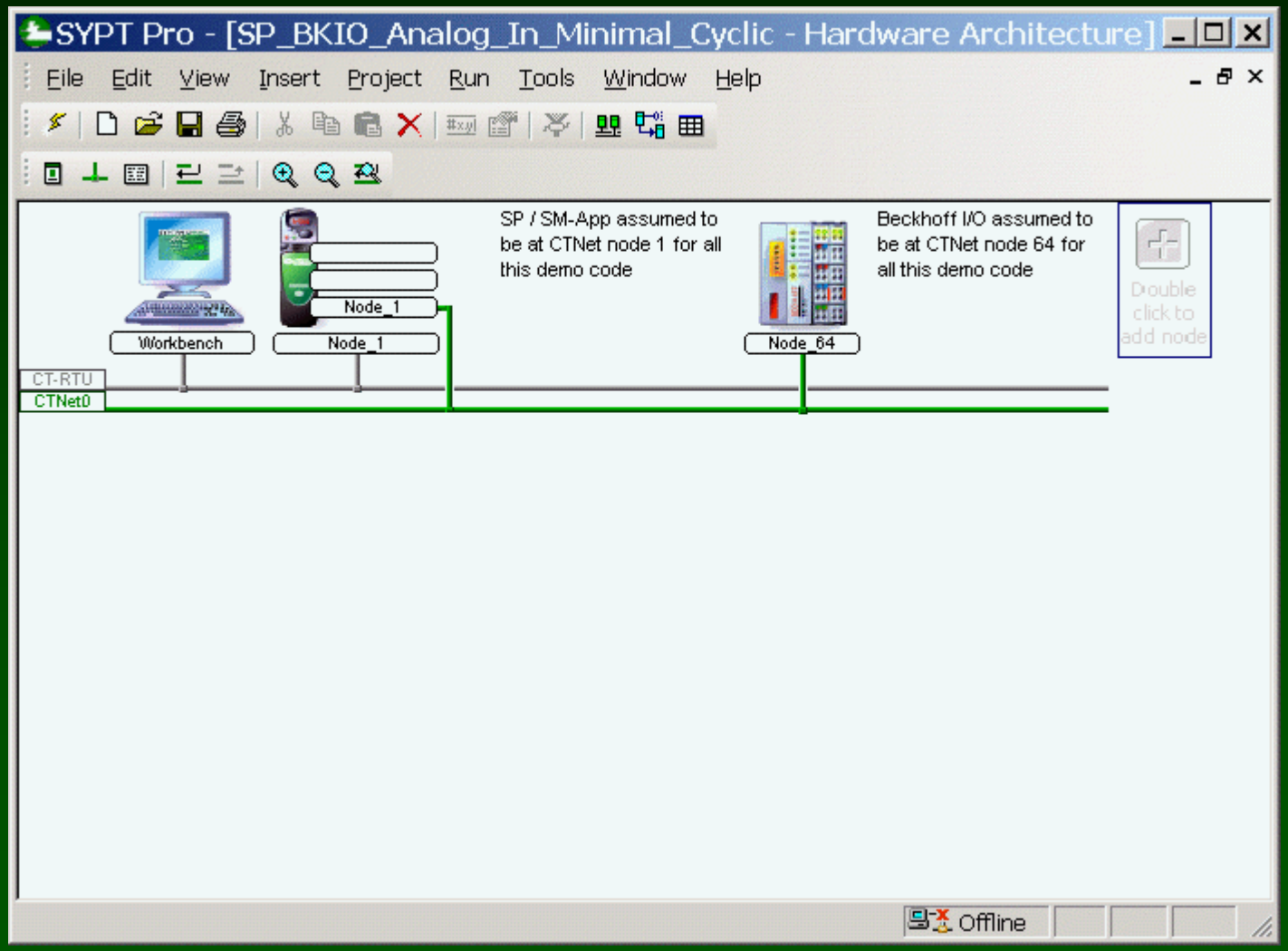
We will send two analog inputs , via cyclic data, from node 64 parameter #03.00 to #03.01, to node 1 parameters #73.10 to #73.11 (_S10% to _S11%), as two 32-bit integers

The SP / SM-Apps at node 1 will generate the Cyclic Data synchronizing message at 10 msec

CTNet Remote I/O

Example: project SP_BKIO_Analog_IN_Minimal_Cyclic

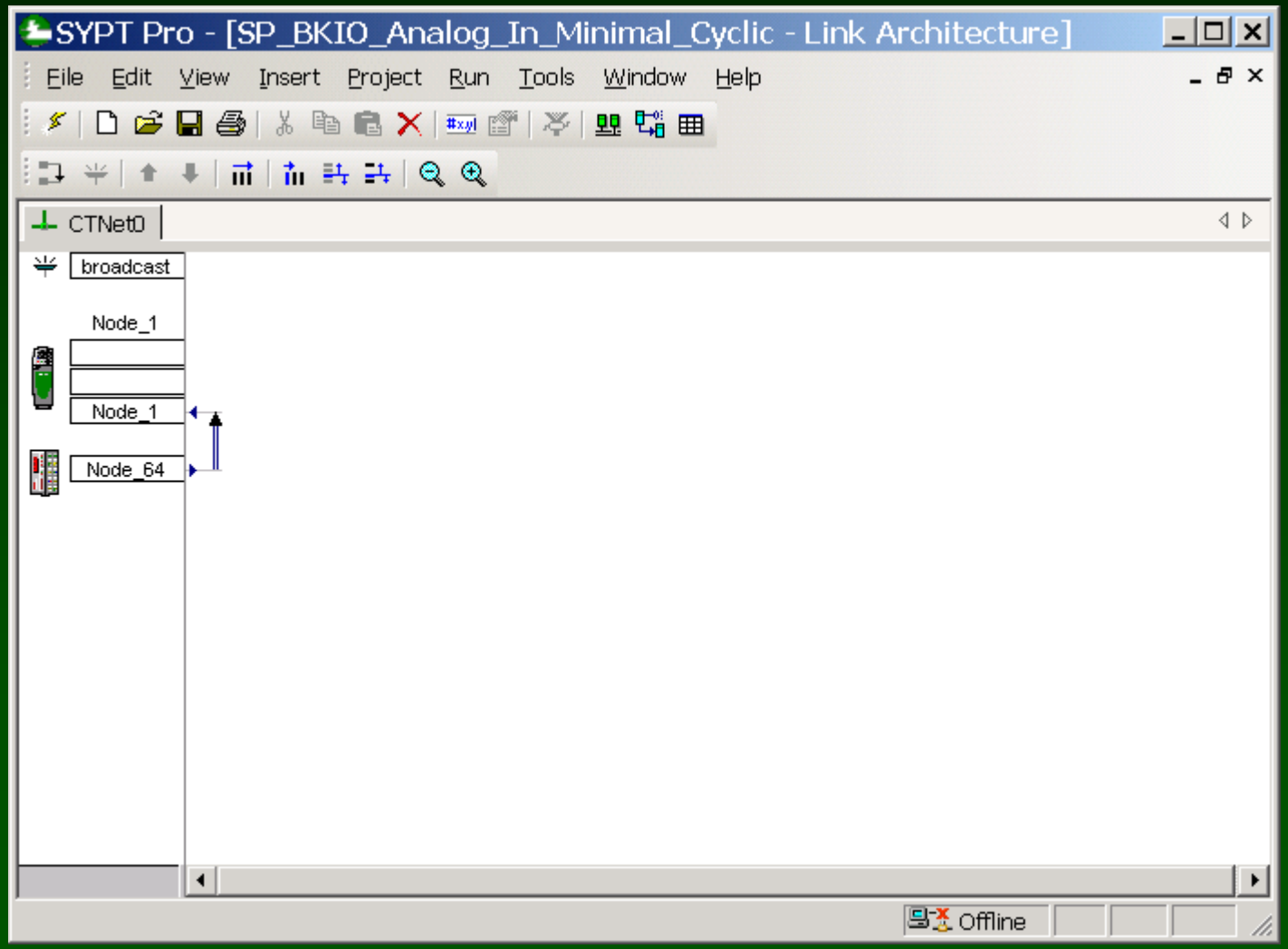
Example: Minimal Analog Input cyclic version (CTNet I/O is node 64)



CTNet Remote I/O

Example: project SP_BKIO_Analog_IN_Minimal_Cyclic

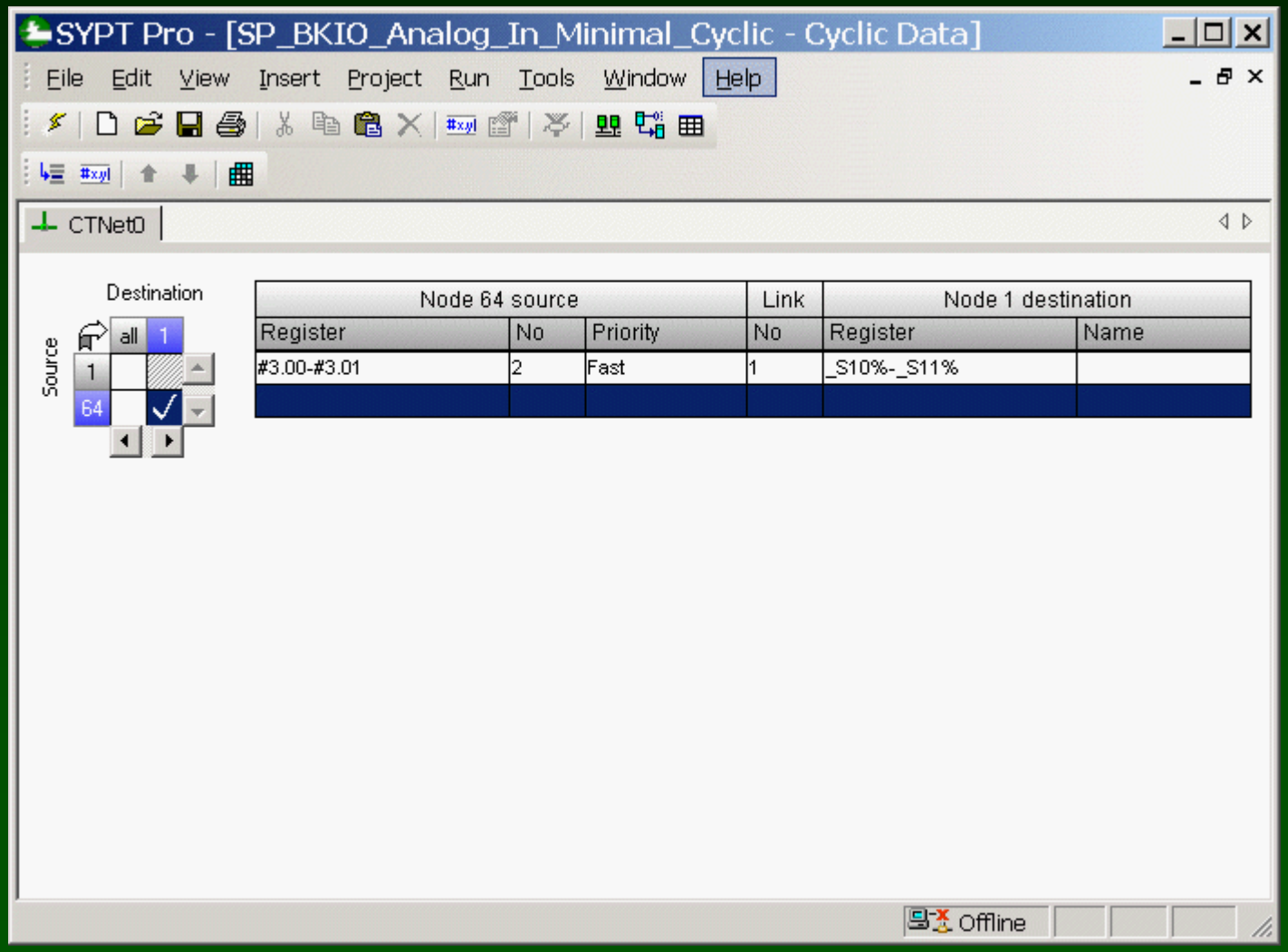
Set up a cyclic link from the Beckhoff to the Unidrive.



CTNet Remote I/O

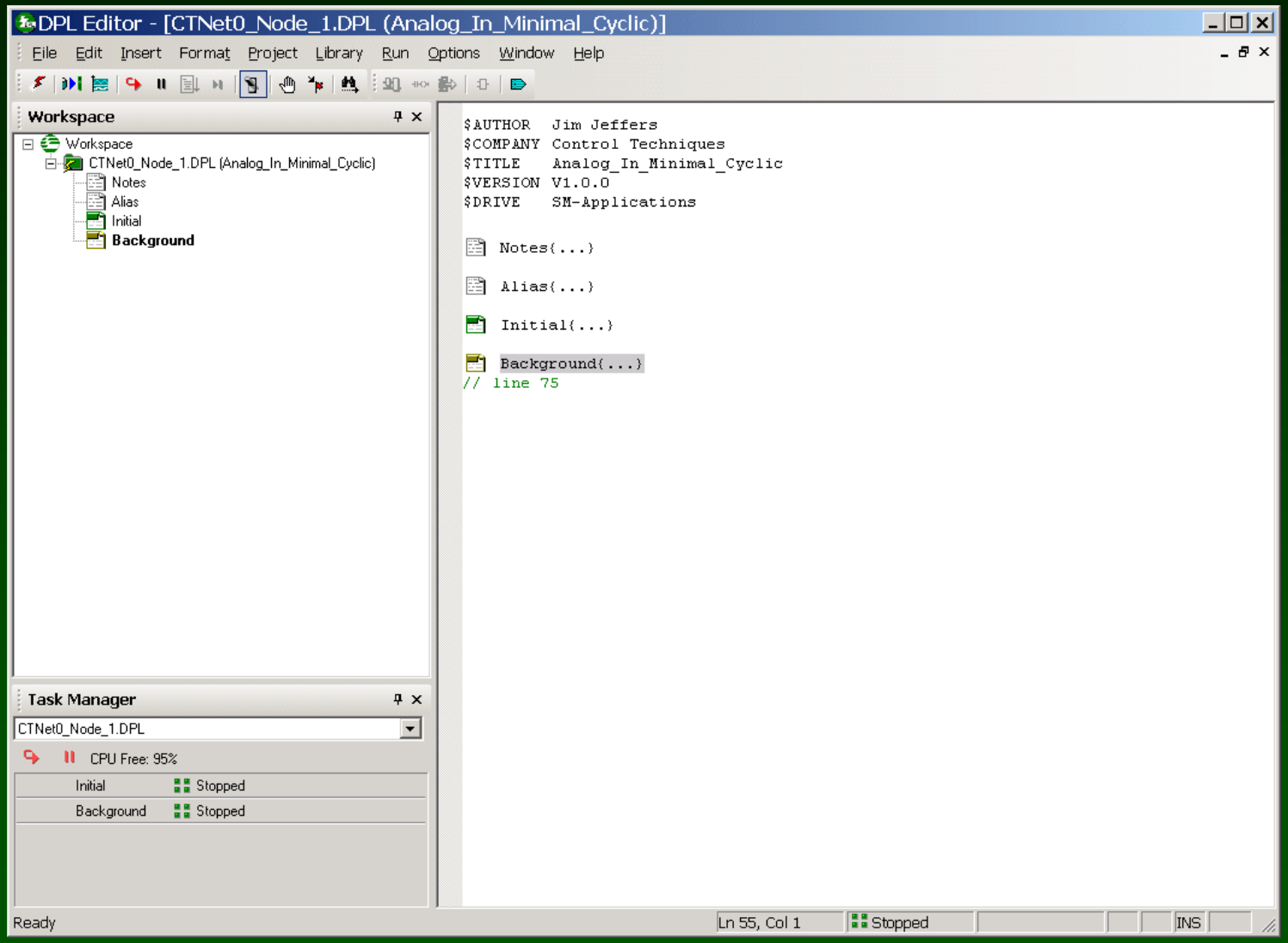
Example: project SP_BKIO_Analog_IN_Minimal_Cyclic

Set up a cyclic link from the Beckhoff to the Unidrive.



CTNet Remote I/O

Open / create the Following DPL program on Unidrive, Node 1



CTNet Remote I/O

Open / create the Following DPL program on Unidrive, Node 1

DPL Editor - [CTNet0_Node_1.DPL (Analog_In_Minimal_Cyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL (Analog_In_Minimal_Cyclic)
    - Notes
    - Alias
    - Initial
    - Background

```
$AUTHOR  Jim Jeffers
$COMPANY Control Techniques
$TITLE   Analog_In_Minimal_Cyclic
$VERSION V1.0.0
$DRIVE   SM-Applications

Notes{...}

Alias{
// No boolean data type is provided
// by Drive Programming Language
$define ON      1
$define OFF     0

$define INPUT_MASK%   0x0000FFFF

// only got 98% max with folowing constant
// define MAX_ANALOG%   32373
// so...
$define MAX_ANALOG_IN%  32100

$define OUTPUT1_MONITOR%   #18.01
$define OUTPUT2_MONITOR%   #18.02

} //Alias

Initial{...}

Background{...}
// line 75
```

Task Manager

CTNet0_Node_1.DPL

CPU Free: 95%

Initial	Stopped
Background	Stopped

Ready | Ln 25, Col 1 | Stopped | INS

CTNet Remote I/O

Open / create the Following DPL program on Unidrive, Node 1

DPL Editor - [CTNet0_Node_1.DPL (Analog_In_Minimal_Cyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL (Analog_In_Minimal_Cyclic)
    - Notes
    - Alias
    - **Initial**
    - Background

```
$AUTHOR  Jim Jeffers
$COMPANY Control Techniques
$TITLE   Analog_In_Minimal_Cyclic
$VERSION V1.0.0
$DRIVE   SM-Applications

Notes{...}

Alias{...}

Initial{

InImage1% = 0
InImage2% = 0

Analog1RawIn%  =  0
Analog2RawIn%   = 0
Analog1InPercent% = 0
Analog2InPercent% = 0

} //Initial

Background{...}
// line 75
```

Task Manager

CTNet0_Node_1.DPL

CPU Free: 95%

Initial	Stopped
Background	Stopped

Ready | Ln 43, Col 1 | Stopped | INS

CTNet Remote I/O

Open / create the Following DPL program on Unidrive, Node 1

DPL Editor - [CTNet0_Node_1.DPL (Analog_In_Minimal_Cyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL (Analog_In_Minimal_Cyclic)
    - Notes
    - Alias
    - Initial
    - **Background**

```
$AUTHOR  Jim Jeffers
$COMPANY Control Techniques
$TITLE   Analog_In_Minimal_Cyclic
$VERSION V1.0.0
$DRIVE   SM-Applications

Notes{...}

Alias{...}

Initial{...}

Background{
top:
   // INPUT - Get the inputs
   // input% is a 32 bit integer, the lower 16 bits reflecting the value of
   InImage1% = _S10% & INPUT_MASK%
   InImage2% = _S11% & INPUT_MASK%

   // PROCESS - Solve the inputs
   // A bit extreem, but coded for maximum clarity
   Analog1RawIn%   =  InImage1%
   Analog2RawIn%   = InImage2%
   Analog1InputPercent% = Analog1RawIn% * 100 / MAX_ANALOG_IN%
   Analog2InputPercent% = Analog2RawIn% * 100 / MAX_ANALOG_IN%

   // OUTPUT - set the outputs
   OUTPUT1_MONITOR% = Analog1InputPercent%   //   #18.01
   OUTPUT2_MONITOR% = Analog2InputPercent%   //   #18.02

goto top: // main background loop
} //Background
// line 75
```

Task Manager

CTNet0_Node_1.DPL

CPU Free: 95%

Initial	Stopped
Background	Stopped

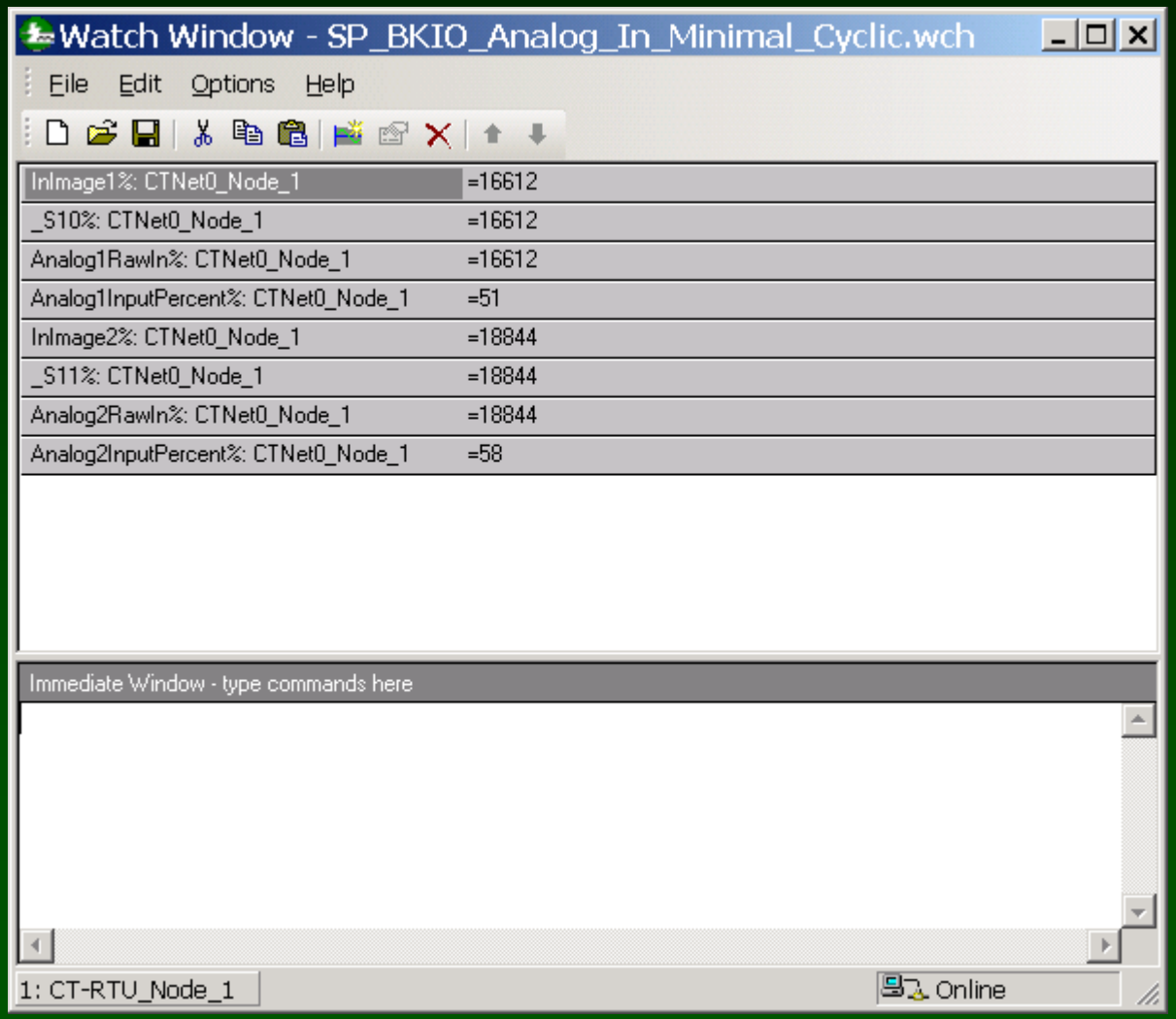
Ready | Ln 55, Col 1 | Stopped | INS

The Analog input values come in automatically to PLC registers _S10% & _S11%.

Here we scale and display the result at #18.01 & #18.02.

CTNet Remote I/O

Watch Window:
Open / create the Following DPL program on Unidrive, Node 1



CTNet Remote I/O

SyPT Pro Example: Let's use Cyclic Data to write an analog output

Strategy: CTNet Remote I/O is at node address 64

Two Analog outputs are available, we will write to both

Analog Outputs start at parameter #04.00 on node 64

Analog Outputs consume one 32-bit word each on the BKIO

Analog Outputs are signed quantities (two's complement)

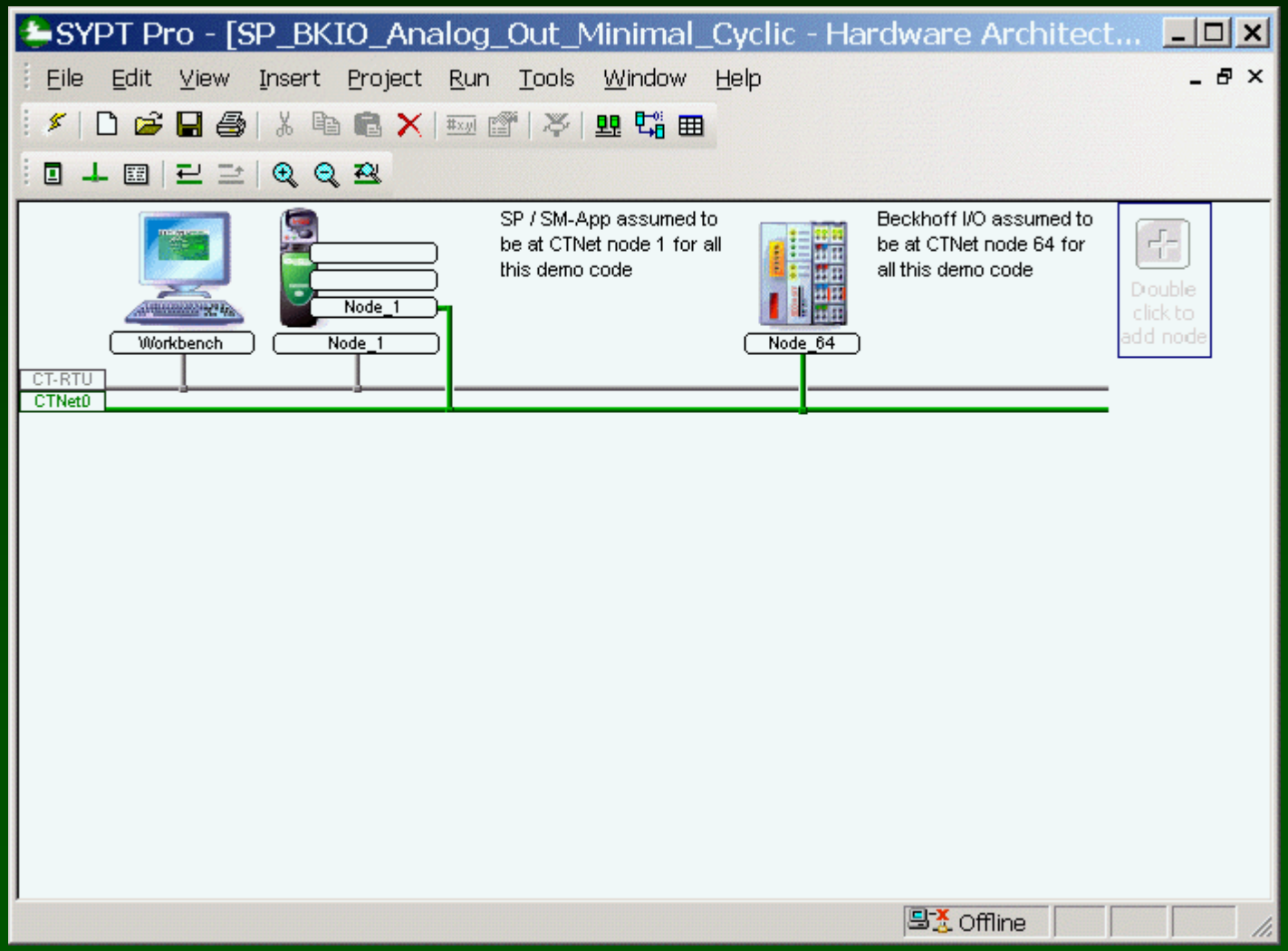
We will send two analog outputs , via cyclic data, from node 1 parameters #74.10 to #74.11 (_R10% to _R11%), to node 64 parameters #04.00 to #04.01, as two 32-bit integers

The SP / SM-Apps at node 1 will generate the Cyclic Data synchronizing message at 10 msec

CTNet Remote I/O

Example: project SP_BKIO_Analog_Out_Minimal_Cyclic

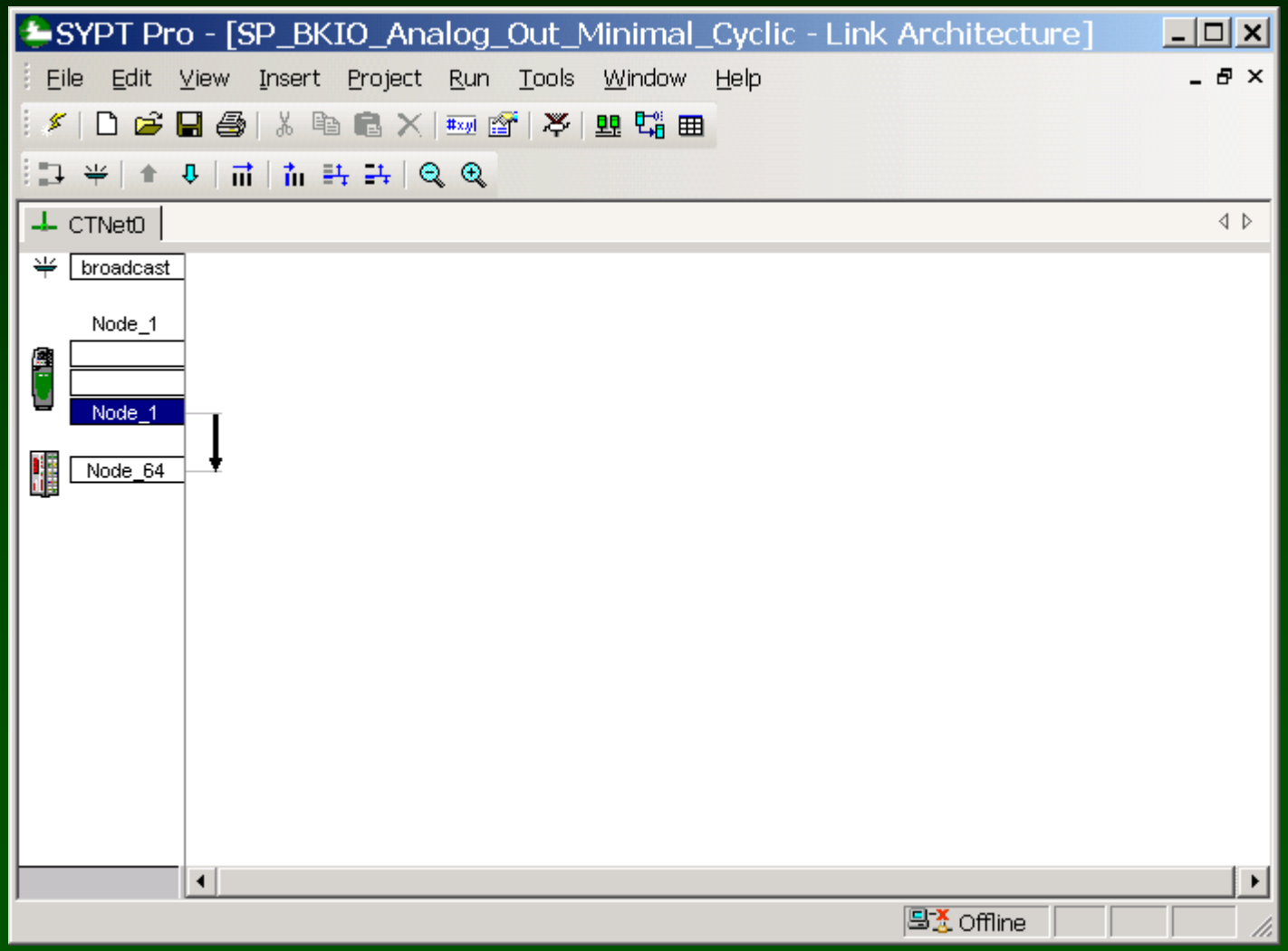
Example: Minimal Analog Output cyclic version (CTNet I/O is node 64)



CTNet Remote I/O

Example: project SP_BKIO_Analog_Out_Minimal_Cyclic

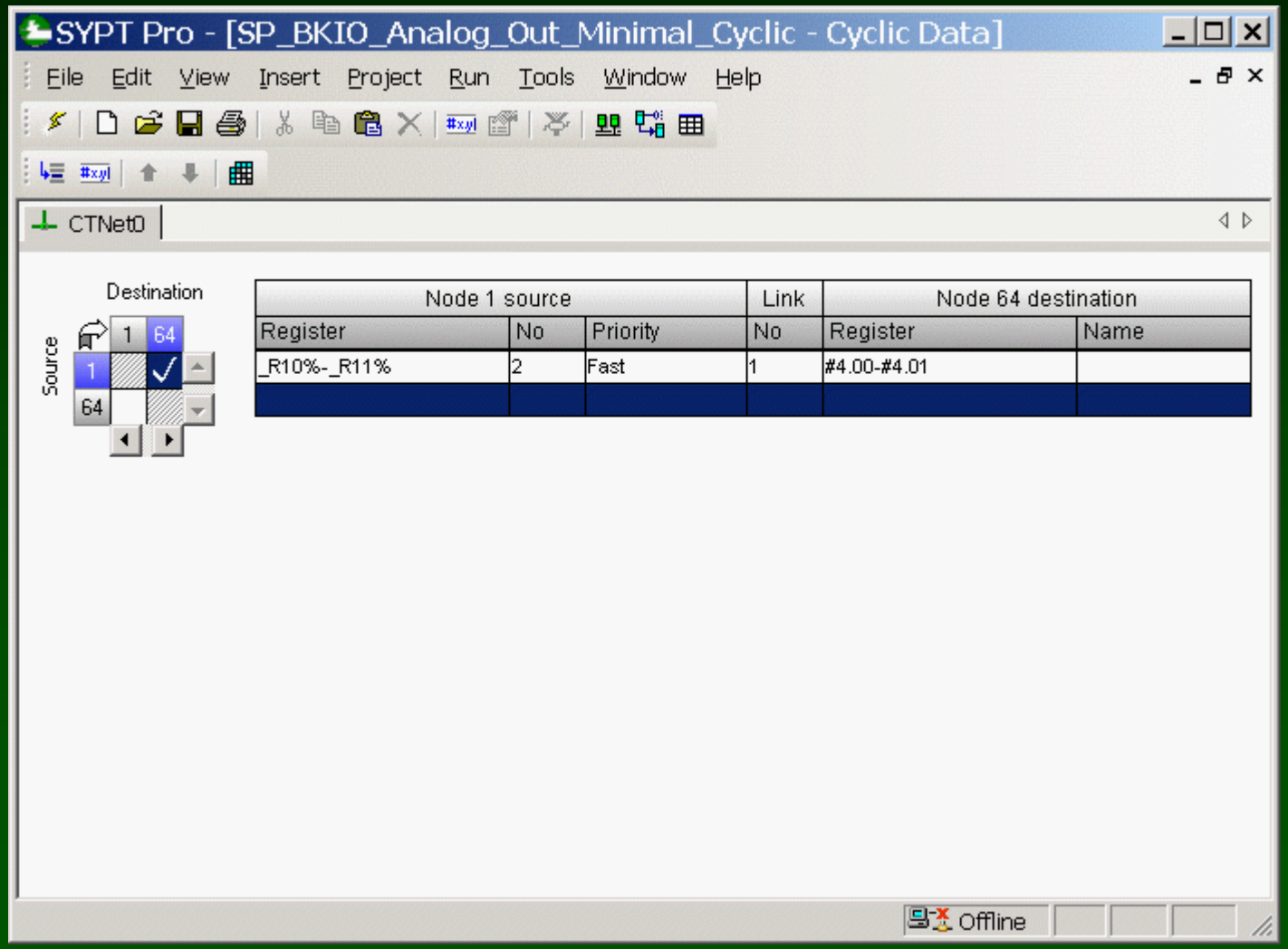
Set up a cyclic link from the Unidrive to the Beckhoff.



CTNet Remote I/O

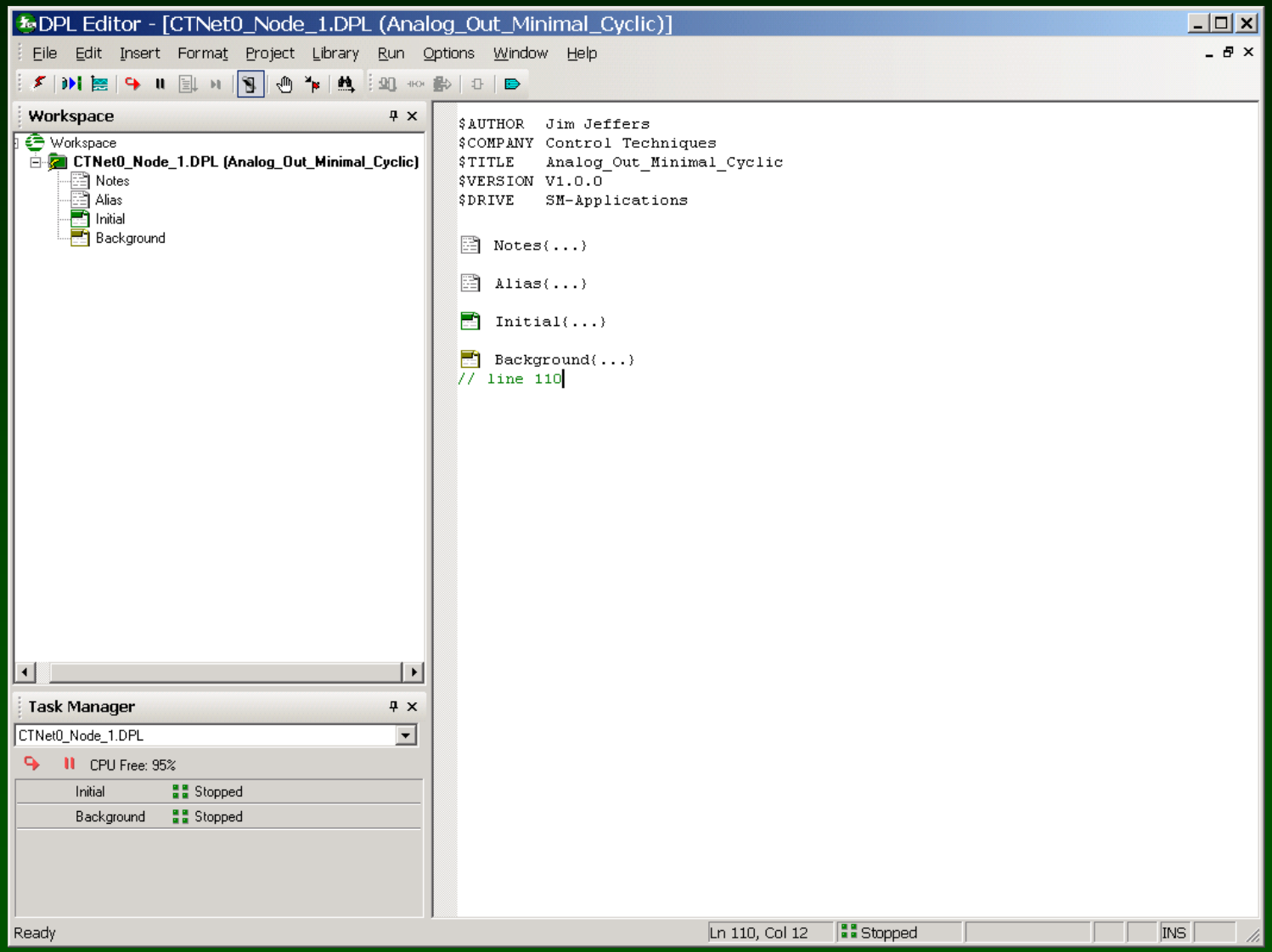
Example: project SP_BKIO_Analog_Out_Minimal_Cyclic

Set up a cyclic link from the Unidrive to the Beckhoff.



CTNet Remote I/O

Open / create the Following DPL program on Unidrive, Node 1



CTNet Remote I/O

Open / create the Following DPL program on Unidrive, Node 1

DPL Editor - [CTNet0_Node_1.DPL (Analog_Out_Minimal_Cyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL (Analog_Out_Minimal_Cyclic)
    - Notes
    - Alias
    - Initial
    - Background

Task Manager

CTNet0_Node_1.DPL

CPU Free: 95%

Initial	Stopped
Background	Stopped

```
$AUTHOR  Jim Jeffers
$COMPANY Control Techniques
$TITLE   Analog_Out_Minimal_Cyclic
$VERSION V1.0.0
$DRIVE   SM-Applications

Notes{...}

Alias{
// No boolean data type is provided
// by Drive Programming Language
$define ON      1
$define OFF     0
$define TRUE    1
$define FALSE   0

$define POS_MAX_LIMIT      32737
$define NEG_MAX_LIMIT      -32737
$define DELTA_LARGE        5
$define DELTA_SMALL        1
// bigger numbers inrease the ramp time
$define RATE_SCALER      5

$define MAX_ANALOG_VALUE%   32737
$define OUTPUT_MASK%   0xFFFFFFFF
$define OUTPUT1_MONITOR%   #18.01
$define OUTPUT2_MONITOR%   #18.02

} //Alias

Initial{...}

Background{...}
// line 110
```

Ready | Ln 27, Col 1 | Stopped | INS

CTNet Remote I/O

Open / create the Following DPL program on Unidrive, Node 1

DPL Editor - [CTNet0_Node_1.DPL (Analog_Out_Minimal_Cyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL (Analog_Out_Minimal_Cyclic)
    - Notes
    - Alias
    - **Initial**
    - Background

```
$AUTHOR  Jim Jeffers
$COMPANY Control Techniques
$TITLE   Analog_Out_Minimal_Cyclic
$VERSION V1.0.0
$DRIVE   SM-Applications

 Notes{...}

 Alias{...}

 Initial{

   bRampUp% = OFF
   PosMaxCount% = POS_MAX_LIMIT * RATE_SCALER
   NegMaxCount% = NEG_MAX_LIMIT * RATE_SCALER
   countValue% = 0

   activityValue% = 0
   Analog1RawOut% =  0
   Analog2RawOut%   = 0
   Analog1OutPercent% = 0
   Analog2OutPercent% = 0

} //Initial

 Background{...}
// line 110
```

Task Manager

CTNet0_Node_1.DPL

CPU Free: 95%

Initial	Stopped
Background	Stopped

Ready | Ln 49, Col 1 | Stopped | INS

CTNet Remote I/O

Open / create the Following DPL program on Unidrive, Node 1

DPL Editor - [CTNet0_Node_1.DPL (Analog_Out_Minimal_Cyclic)]

File Edit Insert Format Project Library Run Options Window Help

Workspace

- Workspace
  - CTNet0_Node_1.DPL (Analog_Out_Minimal_Cyclic)
    - Notes
    - Alias
    - Initial
    - **Background**

```
Notes{...}

Alias{...}

Initial{...}

Background{
top:
   // INPUT - Get the inputs
   // trivial this demo program illustrates outputs (or writes)

   // PROCESS - Solve the inputs
   // create activity to monitor and write to the Beckhoff
   IF bRampUp% = ON THEN
      if countValue% > PosMaxCount% then
         bRampUp% = OFF
         countValue% = PosMaxCount%
      elseif countValue% > (PosMaxCount% * 98 / 100) then
         countValue% = countValue% + DELTA_SMALL
      else
         countValue% = countValue% + DELTA_LARGE
      endif
   ELSE
      if countValue% < NegMaxCount% then
         bRampUp% = ON
         countValue% = NegMaxCount%
      elseif countValue% < (NegMaxCount% * 98 / 100) then
         countValue% = countValue% - DELTA_SMALL
      else
         countValue% = countValue% - DELTA_LARGE
      endif
   ENDIF

   // bit extreem, done this way for maximum clarity
   activityValue% = countValue% / RATE_SCALER

   Analog1RawOut% = activityValue%
   Analog2RawOut% = activityValue%

   Analog1OutPercent% = (Analog1RawOut% * 100 / MAX_ANALOG_VALUE%)
   Analog2OutPercent% = (Analog2RawOut% * 100 / MAX_ANALOG_VALUE%)

   // OUTPUT - set the outputs
   _R10% = Analog1RawOut% & OUTPUT_MASK%   // 0xFFFFFFFF need ALL 32 bits for nega
   _R11% = Analog2RawOut% & OUTPUT_MASK%

   // Write to observable MONITOR location
   OUTPUT1_MONITOR% = Analog1OutPercent%   //   #18.01
   OUTPUT2_MONITOR% = Analog2OutPercent%   //   #18.02

goto top: // main background loop
} //Background
// line 110
```

CTNet0_Node_1.DPL

CPU Free: 95%

Initial	Stopped
Background	Stopped

Ready | Ln 64, Col 1 | Stopped | INS

Here we create activity that becomes the output values.

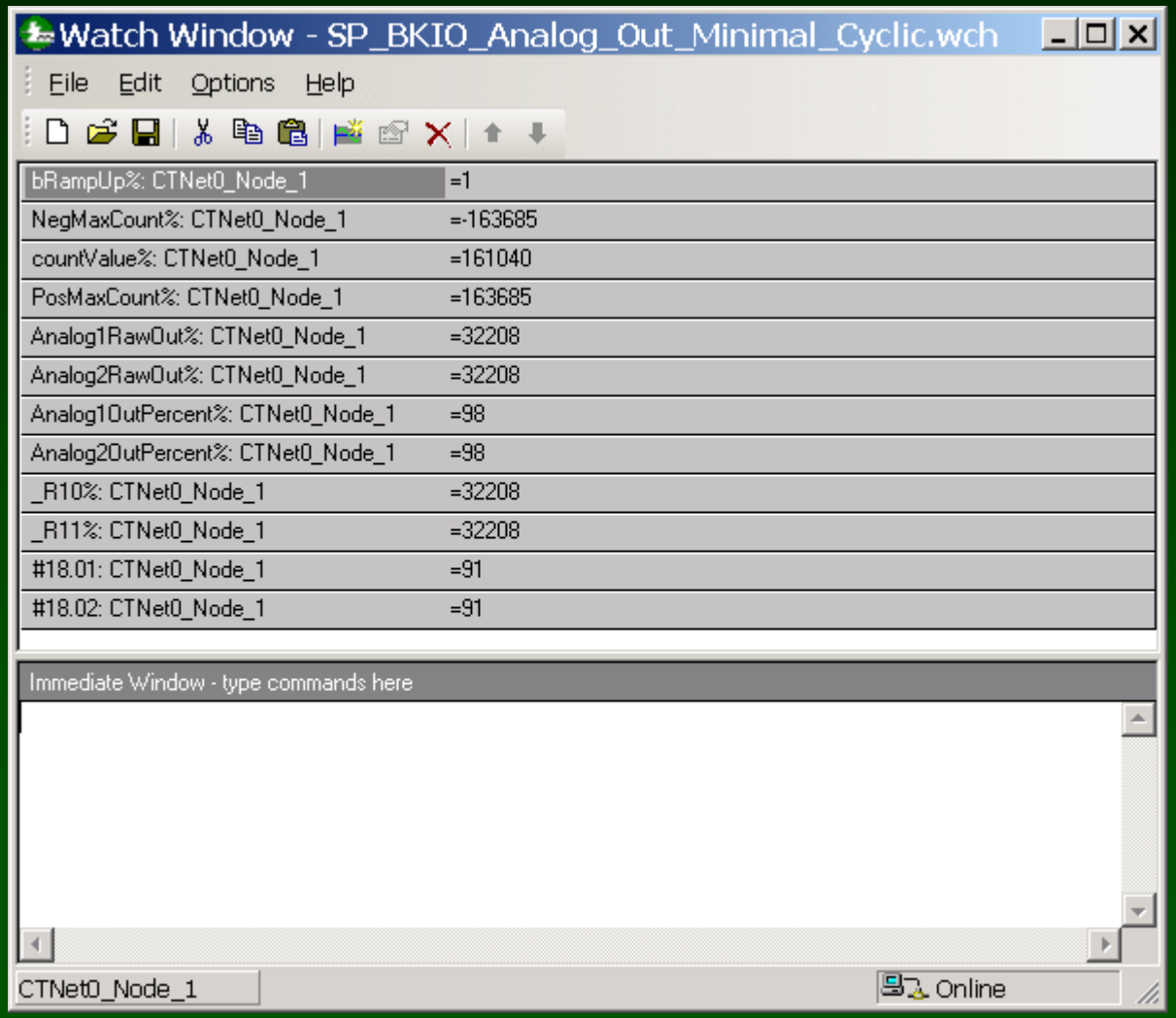
Here we scale the values

Here we combine the output with the mask bits and place at _R10% &_R11% .

The I/O automatically go out from PLC registers _R10% & _R11%.

CTNet Remote I/O

Watch Window:
Open / create the Following DPL program on Unidrive, Node 1



CTNet Remote I/O

Hints to help get Cyclic Transfers to Work.

Build-All while offline. This also “builds” the Beckhoff I/O unit.

Go online and do a “download all”. This downloads cyclic definitions into the Beckhoff I/O unit.

Remember, the Beckhoff I/O Coupler “packs” the digital I/O as sixteen points per word. Forty (40) input = 2 ½ words in and forty (40) output = 2 ½ words out.

Use the Watch window to read the Beckhoff I/O parameter #0.02. This corresponds to #17.36 on the SP/SM-Apps, and a positive number indicates that the configuration has been successfully loaded into the Beckhoff and cyclic communication is possible

Note: The Beckhoff unit can handle about 4 CTNet messages in a millisecond.

Don’t set up so many Cyclic transfers that it exceeds this limit !

End of SyPT Tutorial

SyPT Pro allows the system designer to create and debug a complex multi-drive application from a single workstation.

HAVE FUN !!!!

SCIENCE GATE
Your Automation Partner

SCIGATE AUTOMATION (S) PTE LTD
No.1 Bukit Batok Street 22 #01-01 Singapore 659592
Tel: (65) 6561 0488 Fax: (65) 6562 0588
Email: sales@scigate.com.sg Web: www.scigate.com.sg
Business Hours: Monday - Friday 8.30am - 6.15pm

Jim Lynch
Additions by Jim Jeffers