



ETHERNET COMMUNICATION PROTOCOL

# DynaPCN 10-20-xx

Ethernet/POE Passenger & People Counter

Rev 1.0 – 28 February 2014 – DYPCN-10-20-xx\_EthComProt\_EN\_1.0

## Trademarks

All trademarks both marked and unmarked appearing in this document are the property of their respective owners.

## Revision history

REVISION	DESCRIPTION	DATE
1.0	First release	28 February 2014

# Table of contents

Trademarks.....	2
Revision history.....	2
<b>Table of contents.....</b>	<b>3</b>
<b>1 Introduction.....</b>	<b>5</b>
<b>2 Commands.....</b>	<b>6</b>
2.1 GCOUNTERS.....	6
2.2 FW_VERSION.....	6
2.3 VERSION.....	6
2.4 GDATETIME.....	7
2.5 SDATETIME.....	7
2.6 DISCONNECT.....	7
2.7 RESET.....	8
2.8 RESTORE.....	8
2.9 REBOOT.....	8
2.10 INPUT0.....	9
2.11 INPUT1.....	9
2.12 TESTIN0.....	10
2.13 TESTIN1.....	10
2.14 ENABLE_PC.....	10
2.15 GDOORSTATUS.....	11
2.16 DIAGNOSTIC_EN.....	11
2.17 PCN1001_STATUS.....	12
2.18 RDSAVE.....	13
2.19 UPDATEI.....	13
2.20 UPDATEF.....	14
2.21 GPARMS.....	14
<b>3 Typical application flow.....</b>	<b>16</b>
<b>Notes.....</b>	<b>17</b>

(This page has been intentionally left blank)

# 1 Introduction

This document describes the Ethernet Communication Protocol between a user application and the DynaPCN.

In this protocol, the DynaPCN is the server and the user application is the client.

The following specifications are described for each command:

<b>Command</b>	Name of command
<b>Command Description</b>	Brief description of command purpose
<b>Parameters</b>	Format and type of possible parameter(s)
<b>Parameters Description</b>	Description of parameter(s) : permitted value, default ..
<b>Returns</b>	Format, type and meaning of possible returned parameter(s)
<b>Sample Code</b>	Example of code (only client side)

First of all it is necessary to create a communication (by socket) between the user application and DynaPCN.

In the example below you can find an example of client code to initialize and start the connection with DynaPCN.

```
#include <winsock2.h>
...
struct sockaddr_in my_addr_data,serv_addr_data;
SOCKET sock = socket(AF_INET,SOCK_STREAM,0);
memset((char *)&serv_addr,0,sizeof(serv_addr));
serv_addr.sin_family = AF_INET;
serv_addr.sin_port = htons(PORT);
serv_addr.sin_addr.s_addr = inet_addr(ADDRESS);
if(connect(sock,(struct sockaddr *)&serv_addr, sizeof(sockaddr)) == -1)
{
    printf("Unable to connect to %s", ADDRESS);
    return;
}
...
```

## 2 Commands

### 2.1 GCOUNTERS

Command	Gcounters
Command Description	Returns the current counter values (Incoming & Outgoing)
Parameters	
Parameters Description	
Returns	Two unsigned long values (4 bytes long). The first is the incoming counter value. The second is outgoing counter value.
Sample Code	<pre>unsigned long in,out; SendString(sock,"gcounters"); Recv(sock, &amp;in,sizeof(in)); // incoming counting Recv(sock, &amp;out,sizeof(out)); // outcoming counting</pre>

### 2.2 FW\_VERSION

Command	fw_version
Command Description	Gets the PCN FPGA firmware version
Parameters	
Parameters Description	
Returns	String containing the actual FPGA firmware version number (e.g:"2.9")
Sample Code	<pre>char version[32]; SendString(sock,"fw_version"); RecvString(sock,version);</pre>

### 2.3 VERSION

Command	version
Command Description	Gets the PCN imgserver version
Parameters	
Parameters Description	
Returns	String containing the actual imgserver version number (e.g. "2.3.11.3")
Sample Code	<pre>char version[32]; SendString(sock,"version"); RecvString(sock,version);</pre>

## 2.4 GDATETIME

Command	gdatetime
Command Description	Gets date and time in the PCN system
Parameters	
Parameters Description	
Returns	Two strings. The first contains the actual system date in format <i>DD/MM/YYYY</i> . The second contains the actual system time in format <i>HH.MM</i> .
Sample Code	<pre>char sysdate[16]; char systime[16]; SendString(sock, "gdatetime"); RecvString(sock, sysdate); RecvString(sock, systime);</pre>

## 2.5 SDATETIME

Command	sdatetime
Command Description	Sets date and time in the PCN system
Parameters	Strings
Parameters Description	String containing the date and time. E.g.: 21 May 2014 at 15:39 = "052115392014"
Returns	
Sample Code	<pre>char datetime[16]; // MMDDhhmmYYYY SendString(sock, "sdatetime"); SendString(sock, datetime);</pre>

## 2.6 DISCONNECT

Command	disconnect
Command Description	Closes the communication between users application and the PCN
Parameters	
Parameters Description	
Returns	
Sample Code	<pre>SendString(sock, "disconnect");</pre>

## 2.7 RESET

Command	reset
Command Description	Resets the IN and OUT counters
Parameters	
Parameters Description	
Returns	
Sample Code	<pre>SendString(sock, "reset");</pre>

## 2.8 RESTORE

Command	restore
Command Description	Restores the system factory settings
Parameters	
Parameters Description	
Returns	
Sample Code	<pre>SendString(sock, "restore");</pre>

## 2.9 REBOOT

Command	reboot
Command Description	Reboots the systems
Parameters	
Parameters Description	
Returns	
Sample Code	<pre>SendString(sock, "reboot");</pre>





## 2.10 INPUT0

Command	input0
Command Description	Sets the associated function at rising edge of digital input 0
Parameters	unsigned short (2 byte)
Parameters Description	Refer to Table 2
Returns	
Sample Code	<pre>// To set do nothing functionality unsigned short val = 0; // To set reset counters functionality unsigned short val = 1; // To set Enable/Disable person counting functionality unsigned short val = 2; // To set Test functionality unsigned short val = 3; // Send command SendString(sock, "input0"); Send(sock, &amp;val, sizeof(val));</pre>

Table 1. Mapping Function-Digital Input

Value	Function
0	"Do nothing". The system ignores the signal
1	"Reset Counters". Sets the incoming and outgoing counters to zero when a rising edge is detected
2	"Enable/Disable person counting. When the inputs receive a rising edge (they are enabled) the PCN will start the counting process. When the inputs receive a falling edge (they are disabled) the PCN will stop the counting process.
3	"Test" used to test the signal

## 2.11 INPUT1

Command	input1
Command Description	Sets the associated function at rise edge of digital input 1
Parameters	unsigned short (2 byte)
Parameters Description	Refer to Table 2
Returns	
Sample Code	<pre>// To set do nothing functionality unsigned short val = 0; // To set reset counters functionality unsigned short val = 1; // To set Enable/Disable person counting functionality unsigned short val = 2; // To set Test functionality unsigned short val = 3; // Send command SendString(sock, "input0"); Send(sock, &amp;val, sizeof(val));</pre>

## 2.12 TESTIN0

Command	testin0
Command Description	Tests digital input 0 functionality
Parameters	
Parameters Description	
Returns	unsigned char (1 byte). 1 : Signal detected 0: No signal detected
Sample Code	<pre>unsigned char val; SendString(sock, "testin0"); Recv(sock, &amp;val, sizeof(val));</pre>

## 2.13 TESTIN1

Command	testin1
Command Description	Tests digital input 1 functionality
Parameters	
Parameters Description	
Returns	unsigned char (1 byte). 1 : Signal detected 0: No signal detected
Sample Code	<pre>unsigned char val; SendString(sock, "testin1"); Recv(sock, &amp;val, sizeof(val));</pre>

## 2.14 ENABLE\_PC

Command	enable_pc
Command Description	Enables or disables people counting
Parameters	Unsigned char (1 byte).
Parameters Description	1 : enables people counting 0: disables people counting
Returns	
Sample Code	<pre>unsigned char val = 1; // Enables people counting SendString(sock, "enable pc"); Send(sock, &amp;val, sizeof(val));</pre>

## 2.15 GDOORSTATUS

Command	gdoorstatus
Command Description	Gets the door status
Parameters	
Parameters Description	
Returns	unsigned char (1 byte). 1 : The door is open 0 : The door is close
Sample Code	<pre>unsigned char val; SendString(sock, "gdoorstatus"); Recv(sock, &amp;val, sizeof(val));</pre>

## 2.16 DIAGNOSTIC\_EN

Command	diagnostic_en
Command Description	Enables or disables the diagnostic
Parameters	unsigned char ( 1 byte)
Parameters Description	Diagnostic status : 0 : disabled 1 : enabled
Returns	
Sample Code	<pre>unsigned char val = 1; // Enables diagnostic SendString(sock, "diagnostic_en"); Send(sock, &amp;val, sizeof(val));</pre>

## 2.17 PCN1001\_STATUS

Command	pcn1001_status
Command Description	Gets the status of the PCN and the diagnostic error code
Parameters	
Parameters Description	
Returns	<p>2 unsigned char ( 1 byte) value.</p> <p>The first indicates the status of the PCN, if it is OK or if a problem arises:</p> <ul style="list-style-type: none"> <li>• 1 : OK</li> <li>• other : ERROR, a problem arises</li> </ul> <p>The second value gives an information about the problem detected For the errors code, see Table 4.</p>
Sample Code	<pre> unsigned char status,code_error; SendString(sock,"pcn1001_status"); // gets the PCN status Recv(sock,&amp;status, sizeof(status)); // gets the error code Recv(sock,&amp;code_error, sizeof(code_error)); </pre>

Table 2. Diagnostic Error Code

Error Code	Description
0	No problem arised
1	Right sensoris is occluded
2	Left sensor is occluded
3	Both sensor are occluded
4	Right sensor is obscured
8	Left sensor is obscured
12	Both sensor are obscured
16	High difference between right and left sensor
20,24,28	High difference between right and left sensor and both sensors are obscured

## 2.18 RDSAVE

Command	rdsave
Command Description	Return a buffer string of variable dimension, that containing the log counters file of the PCN. An user application can save this buffer in a file located in own workstation as shown in sample code section.
Parameters	
Parameters Description	
Returns	Values returned: - Size of the log file : int (4byte) - Buffer file : vector of the unsigned char of length <size>
Sample Code	<pre> int size = 0; unsigned char *buffer; FILE *out; // Send the command SendString(sock,"rdsave"); // Receive the size of the buffer Recv(sock,(char *)&amp;size,sizeof(size)); buffer = new unsigned char [size]; // Receive the buffer Recv(sock,(char *)buffer,size); fopen(out,"logfile.txt","w"); fwrite(buffer,size,1,out); fclose(out); delete [] buffer; </pre>

## 2.19 UPDATEI

Command	updateI
Command Description	Updates the imgserver daemon
Parameters	- integer (4 byte) - vector unsigned char ( 1byte * size)
Parameters Description	- Size of the log file - Buffer file : vector of the unsigned char of length <size> containing the file to upload.
Returns	An integer value (4 byte) . If > 0 updating complete successfully, otherwise updating failed.
Sample Code	<pre> int handle,status; handle = fopen(fileName, "rb"); int size = fseek(handle,0,2); fseek(handle,0,0); unsigned char *buffer = new unsigned char [size]; fread(handle, buffer, size); fclose(handle); SendString(sock,"updateI"); // Sends Command Send(sock,&amp;size,sizeof(size)); // Sends file's size Send(sock,(char *)buffer,size); // Sends the buffer Recv(sock,(char *)&amp;status,sizeof(status)); </pre>

## 2.20 UPDATEF

Command	updateF
Command Description	Updates the FPGA firmware
Parameters	- integer (4 byte) - vector unsigned char ( 1byte * size)
Parameters Description	- size of the log file - buffer vector of the unsigned char of length <size> containing the file to upload
Returns	An integer value (4 byte) . If > 0 updating complete successfully, otherwise updating failed.
Sample Code	<pre>int handle,status; handle = fopen(fileName, "rb"); int size = fseek(handle,0,2); fseek(handle,0,0); unsigned char *buffer = new unsigned char [size]; fread(handle, buffer, size); fclose(handle); SendString(sock,"updateF"); // Sends Command Send(sock,&amp;size,sizeof(size)); // Sends file's size Send(sock,(char *)buffer,size); // Sends the buffer Recv(sock,(char *)&amp;status,sizeof(status));</pre>

## 2.21 GPARMS

Command	Gparms
Command Description	Gets the PCN configuration and enviroment parameters
Parameters	Vector of 256 unsigned char ( 256*1 byte)
Parameters Description	Refer to Table 5
Returns	
Sample Code	<pre>unsigned short parameters[256]; SendString(sock,"gparms"); Recv(sock,parameters,sizeof(parameters));</pre>

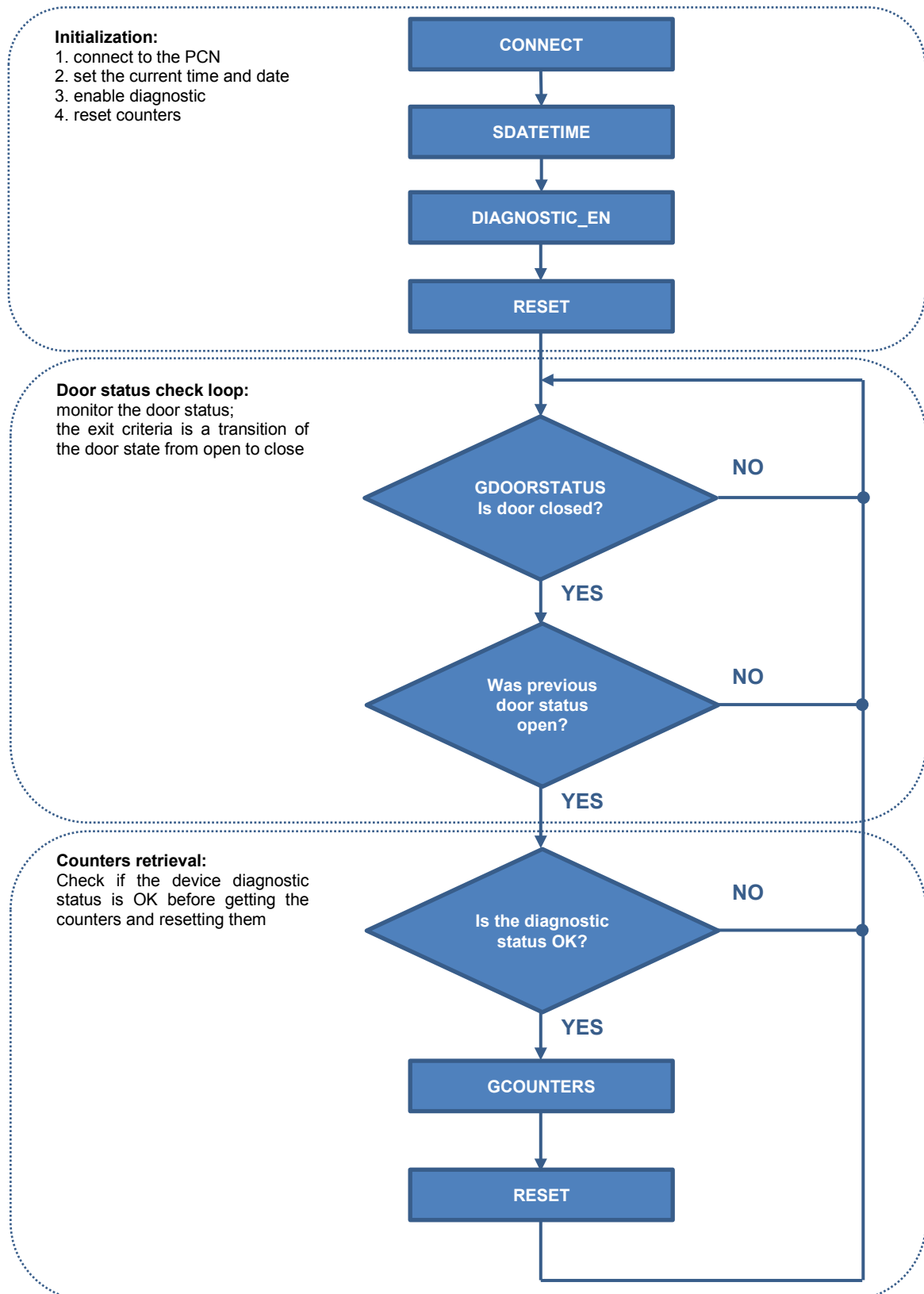


Table 3. List of PCN parameters

NUM	Parameter	Description	Available/Used
0	outime0	Optocoupled open time for digital out 0	YES
1	outime1	Optocoupled open time for digital out 1	YES
2	input0	Associated function at digital IN 0 (cfr 2.16)	YES
3	input1	Associated function at digital IN 1 (cfr 2.17)	YES
4	sled	Led intensity	YES
5	dir	Direction	YES
6	threshold	Threshold of the door (related on the image coordinate)	YES
7	serial_id	RS485 ID	YES
8	serial_br	RS485 Baud rate	YES
9	serial_db	RS485 Data bits	YES
10	serial_pr	RS485 Parity bit	YES
11	serial_sb	RS485 Stop bit	YES
12	detect_area	Distance from sensors	NO
13	autoled	Automatic led intensity	NO
14	inst_height	Installation height	YES
15	serial_ms	Flag for master/slave parameters	YES
16	serial_sid	ID for second serial port	YES
17	serial_sbr	Baud rate for second serial port	YES
18	serial_sdb	Data bits for second serial port	YES
19	serial_dpr	Parity bit for second serial port	YES
20	serial_ssb	Stop bit for second serial port	YES
21	timekbg	Time to wait for background automatic update	NO
22	staticth	Max number of the changed pixels to consider a new background situation	NO
23	slave_id	ID of PCN Slave	YES
24	sx_dx	Power connector is external the two sensor (see the manual)	NO
25	inst_dist	Installation distance between two consecutive PCN in widegate modality	YES
26	wg_check	Check the slave presence in widegate	YES
27	sys_num	Number the systems connected in widegate modality	YES
28	sys_num_index	Index of current PCN in widegate modality	YES
29	sxlimit	Left column limit of the NTZ	YES
30	dxlimit	Right column limit of the NTZ	YES
31	sxlimit_riga_start	Start left row of the NTZ	YES
32	dxlimit_riga_start	Start right row of the NTZ	YES
33	sxlimit_riga_end	Start left row of the NTZ	YES
34	dxlimit_riga_end	End right row of the NTZ	YES
35	cond_diff_1p	Flag to describe a particular situation in the scene	NO
36	diagnostic_en	Flag to enable/disable diagnostic	YES
37	move_det_col0	Start column of the ROI for move detection	NO
38	move_det_col1	End column of the ROI for move detection	NO
39	move_det_row0	Start row of the ROI for move detection	NO
40	move_det_row1	End row of the ROI for move detection	NO
41	move_det_alfareg	Learning rate for background model in move detection algorithm	NO
42	move_det_thr	Threshold for move detection decision algorithm	NO
43	move_det_en	Flag for the activation of the move detection option	YES
44	door_stairs_en	Flag for the stair modality	NO
45	up_line_limit	End row of the NTZ from high	YES
46	down_line_limit	Start row of the NTZ from down	YES
47	dis_autobkg	Flag to enable/disable auto-background modality	NO
48	door_size	Size of the door (in cm)	YES
49	handle_oor	Flag to enable/disable the Out-of-Range handler	YES
50	auto_gain	Flag to enable/disable automatic Vref adjustment	NO

### 3 Typical application flow

The following flow chart describes the logic of a typical transportation application where one digital input (0/1) of the PCN is connected to a circuit which retrieves the open/close state of the guarded port, enabling/disabling the counting accordingly.





## This image shows a single sheet of white paper with horizontal blue ruling lines. The lines are evenly spaced and run across the width of the page. There are approximately 20 lines visible. The paper has a slight shadow on the right side, suggesting it's resting on a surface.

## WORLD SUPPORT

### HEADQUARTERS

EUROTECH  
Via Fratelli Solari, 3/a  
33020 Amaro (UD) – Italy  
Tel: +39 0433.485.411  
Fax: +39 0433.485.499  
E-mail: support.it@eurotech.com  
Web: www.eurotech.com

### EUROPE

#### ITALY

EUROTECH  
Tel: +39 0433.485.411  
Fax: +39 0433.485.499  
E-mail: sales.it@eurotech.com  
E-mail: support.it@eurotech.com  
Web: www.eurotech.com

#### UNITED KINGDOM

EUROTECH  
Tel: +44 (0) 1223.403410  
Fax: +44 (0) 1223.410457  
E-mail: sales.uk@eurotech.com  
E-mail: support.uk@eurotech.com  
Web: www.eurotech.com

#### FRANCE

EUROTECH  
Tel: +33 04.72.89.00.90  
Fax: +33 04.78.70.08.24  
E-mail: sales.fr@eurotech.com  
E-mail: support.fr@eurotech.com  
Web: www.eurotech.com

### AMERICAS

#### USA

EUROTECH  
Toll free: +1 888.941.2224  
Tel: +1 301.490.4007  
Fax: +1 301.490.4582  
E-mail: sales.us@eurotech.com  
E-mail: support.us@eurotech.com  
Web: www.eurotech-inc.com

### ASIA

#### JAPAN

ADVANET  
Tel: +81 86.245.2861  
Fax: +81 86.245.2860  
E-mail: sales@advanet.jp  
E-mail: tsupport@advanet.jp  
Web: www.advanet.jp

#### INDIA

EUROTECH  
Tel: +91 80.43.35.71.17  
E-mail: sales.in@eurotech.com  
E-mail: support.in@eurotech.com  
Web: www.eurotech.com

For your local contact please refer to: [www.eurotech.com/contacts](http://www.eurotech.com/contacts)