ernitec

SYSTEM X

PC –interface
I151SX-PC
I151SX-PCIF

Protocol manual

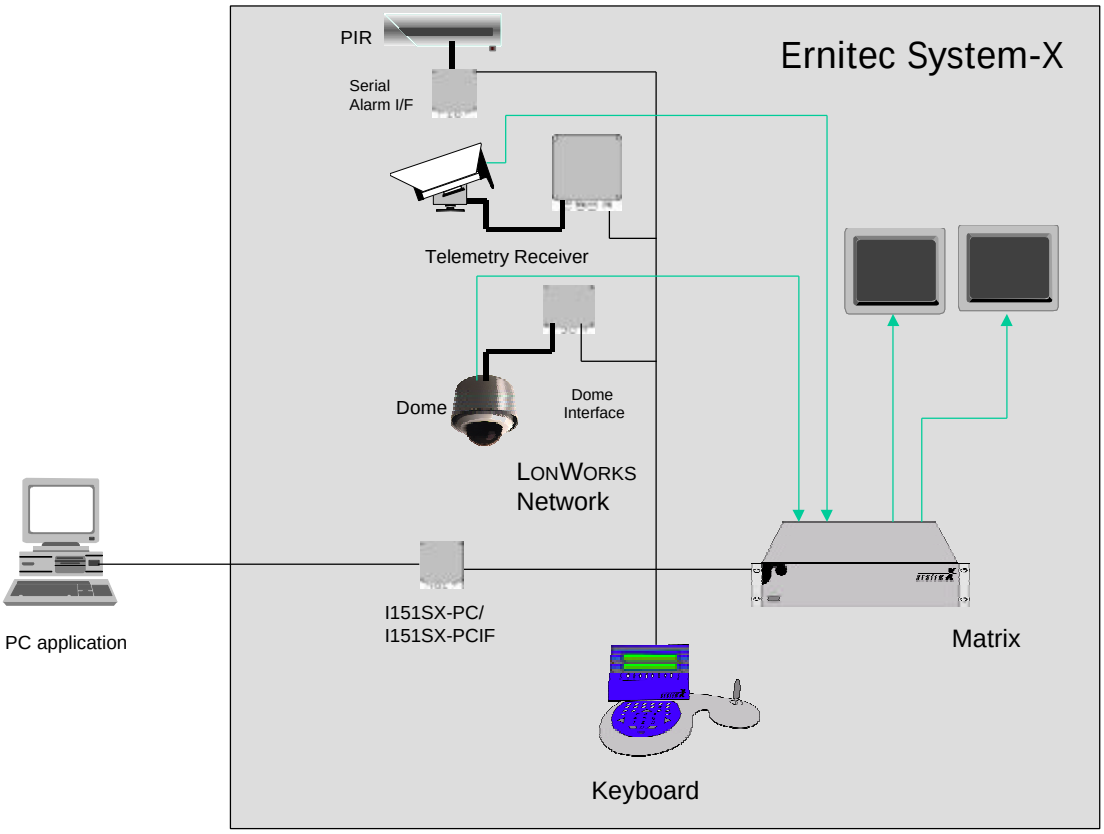
Version 1.3
2853-00019

Table of Contents

Page

Introduction

This document describes how to control the Ernitec System-X from a 3rd party PC-based application, such as a video management system, an access control system, or a similar application. The drawing below shows a typical configuration:



An Ernitec Matrix System, System-X is generally controlled from a dedicated keyboard. By connecting an Ernitec I151SX-PC/ I151SX-PCIF unit in stead of or in addition to a keyboard you can control the System-X from a PC. In fact you are simulating (emulating) a keyboard.

To operate the system from a PC you simply issue commands from the PC in the format described in this document. To most commands you will get an associated response back in the same format and in case of a reset or an alarm the I151SX-PC/ I151SX-PCIF will initiate the communication.

Interfacing to the I151SX-PC

Only to be used in existing designs, for future development use the I151SX-PCIF instead.

Getting started:

The I151SX-PC.

To use your I151SX-PC unit with your System-X, the files described below must be downloaded to the unit:

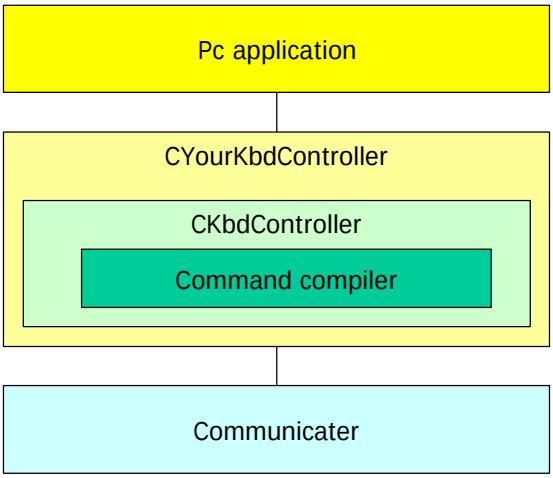
- The Neuron files, I151SX-PC.xif and I151SX-PC.nxe.
- The host file I151SX-PC.mot

See the S111SX NodeManager Installation Manual on how to download the files.

The PC application.

The PC application may use the API supplied on the installation CD to communicate with the I151SX-PC in the I151SX-PC samples directory.

Sample applications show how the API is used through inheritance of the CKBDController class, i.e. CYourKBDController.



API Platform files:

File	Description
Communicator.h	RS232 driver definition
Communicator.cpp	RS232 driver implementation
CommandCompiler.h	Protocol disassembler definition
KBDController.h	The controller class definition
KBDController.cpp	The controller class implementation[1]

Application sample files:

File	Description
I151SXPC_PCIF Test.h	Main application definition
I151SXPC_PCIF Test.cpp	Main application implementation
I151SXPC_PCIF TestDlg.h	Main dialog definition
I151SXPC_PCIF TestDlg.cpp	Main dialog implementation

[1] When using the I151SX-PC **HEX_IF** must be undefined in KBDController.cpp.

The PCIF-B protocol frame.

The I151SX-PC command protocol is a raw binary protocol.

The format of the protocol frame is as follows:

STX	Length	Data0	• • • • • • • •	DataN	Checksum

The value of the STX is 2; the length is the number of bytes equal to STX + length + Data, i.e. the amount of bytes shown in grey above. The checksum is a simple sum of these bytes. The size of the checksum is one byte.

Commands

Command format:

Byte	Data
0	Device command
1	Main Command
2	Sub Command
3/ 3-4 MSB,LSB	Arg1 (Byte or Word value)
4/ 5-6 MSB,LSB	Arg2 (Byte or Word value)
5/ 7-8 MSB,LSB	Arg3 (Byte or Word value)
6/ 9-10 MSB,LSB	Arg4 (Byte or Word value)
7/ 11-12 MSB,LSB	Arg5 (Byte or Word value)

Description of device commands.

The Device Command values are:

Value	Device command
0	System command
1	Camera command
2	Monitor command
8	Alarm event
9	Relay box command
10	DVR command

0 - System commands.

To transmit:

Main Cmd Byte	SubCmd Byte	Arg1 Word
0-Service	0-Host Ready	0- Request
	1-Control ID query	
	2-LogOnConf	Priority 50-99 High-low
2-System msg (Described later in this paragraph)		

Re. Command 0/0 the I151SX-PC request the host to be ready.

To receive:

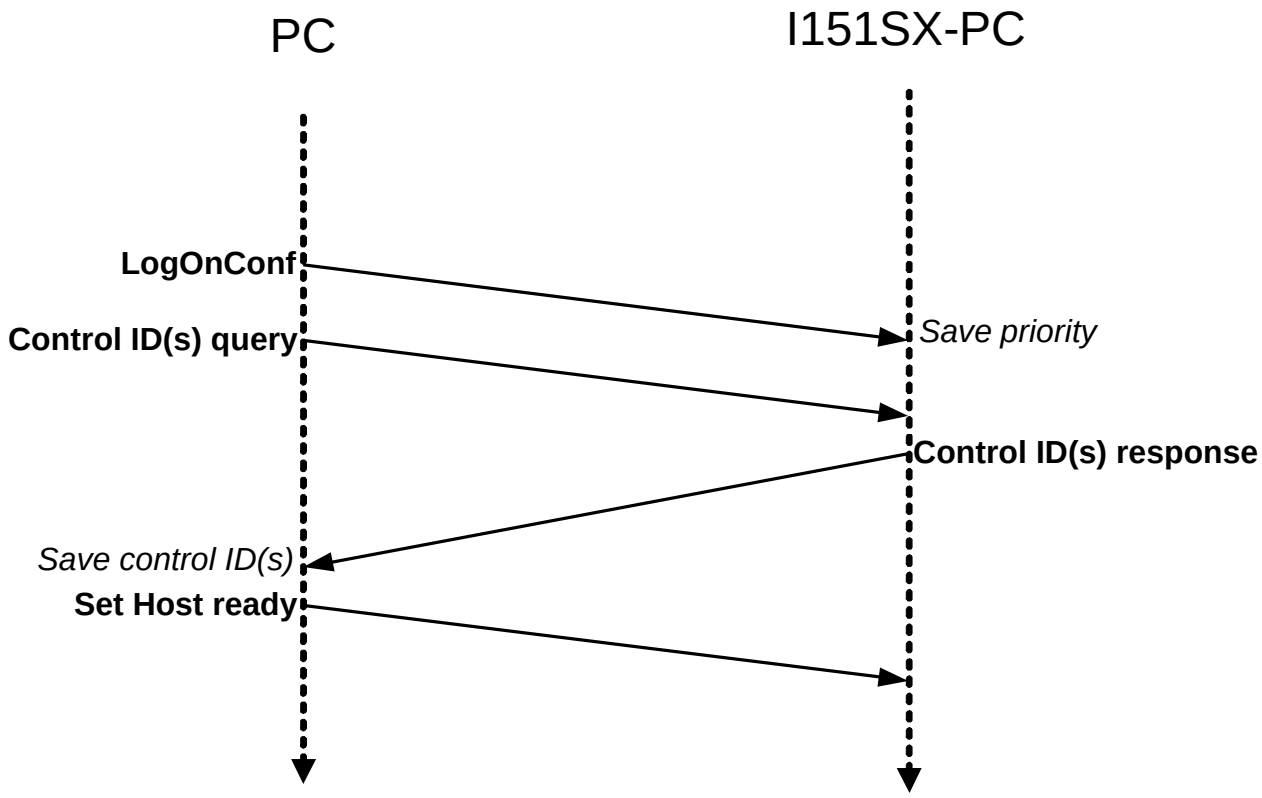
Main Cmd Byte	SubCmd Byte	Arg1 Word
0-Service	0-Host Ready	0-Not ready 1-Ready
2-System msg		
128-Control ID config	1-response	Control ID

To receive:

Main Cmd Byte	SubCmd Byte	Arg1 Word	Arg2 Word	Arg3 Word	Arg4 Word	Arg5 Word	Arg6 Word
2-System message	0	0	1	3	Hours*100)minutes, (date*100)+month, year, (dateformat*256) +weekday		
			2-Request time				
			27-Illegal sequence	MonID	Seq	20	ID
			40-Clear alarm	MonID	KbdID	AlarmID	
			41-View previous Alarm on Mon Queue	MonID	KbdID	AlarmID	
			42-View next Alarm on Mon Queue	MonID	KbdID	AlarmID	
			43-Select alarm	MonID	KbdID	AlarmID	
			50-Toggle text state on Mon	MonID			
			72-Start macro	Macro	Priority	20	ID
			77-Stop macro	Macro	Priority	20	20
			80-Hold macro	Macro	Priority	20	20
			90-Start sequence	MonID	Seq	20	20
			91-Hold Seq	MonID	Don' t Care	20	20

System messages are received from/ transmitted to the control network, without the I151SX-PC verifying the data.

Synchronisation on reset:



1- Camera commands.

Camera control command must be requested and granted prior to camera commands.

To transmit.

Main Cmd Byte	SubCmd Byte	Arg1 Word	Arg2 Word	Arg3 Word	Arg4 Word	Arg5 Word
0 – Control	0 - Request	CTRL Priority 50-99 High-low	Camera ID 1-65535			
	2- Release		Camera ID 1-65535			
Main Cmd Byte	**Arg1 Byte**	**Arg2 Byte**	**Arg3 Byte**	**Arg4 Byte**	**Arg5 Byte**	**Arg6 Byte**
1- PTZ	PAN cmd 0-Stop 1-Right 2-Left	Pan Speed 1-100	TILT cmd 0-Stop 1-Up 2-Down	Tilt Speed 1-100	ZOOM cmd 0-Stop 1-Tele 2-Wide	Zoom speed 1-100
2 – Lens	IRIS cmd 0-Hold 1-Open 2-Close 3-Auto	FOCUS cmd 0-Hold 1-Near 2-Far 3-Auto	BACKLIGHT 0-Off 1-On			
Main Cmd Byte	**SubCmd Byte**	**Arg1 Word**	**Arg2 Word**	**Arg3 Word**	**Arg4 Word**	**Arg5 Word**
3 – Position	POS cmd 0-Save 1-Call 2-Tour	POS NO. 1-255				
4 – Autopan	1-Limit	1-Left Lim. 2-Right Lim.				
	2-Start	1-Start	Percentage speed			
7 – Auxrelay	AUX cmd 0-Off 1-On	Aux no. 1-255, 0-Query				

To receive.

Main Cmd Byte	SubCmd Byte	Arg1 Word	Arg2 Word	Arg3 Word	Arg4 Word	Arg5 Word
0- Control	1- Response	Control Response 0 = No 1 = Pending 2 = Release 3 = Query 4 = Reset 127 = Error 255 = Timeout	Camera ID 1-65535	Ctrl ID 0-65535 *	Controller class	
3 – Position	POS response 0-Saved 1-Called	POS No. 1-255				
7-Auxrelay	0-Status	16 Bit-status				

* The value 65535 corresponds to “my” controller ID

2-Monitor commands.

Monitor control command must be requested and granted prior to monitor commands.

To transmit.

Main Cmd Byte	SubCmd Byte	Arg1 Word	Arg2 Word	Arg3 Word	Arg4 Word	Arg5 Word
0-Control	0-Request	CTRL Priority 50-99 High-low	Monitor ID 1-65535			
	2-Release	Don't care	Monitor ID 1-65535			
1-X Point	0-Request	CTRL priority 1-100	Monitor ID 1-65535	Ctrl ID 1-65535	Camera ID 1-65535	
	2-Release	Don't care	Monitor ID 1-65535	Ctrl ID 0-65535	Camera ID 1-65535	

To receive.

Main Cmd Byte	SubCmd Byte	Arg1 Word	Arg2 Word	Arg3 Word	Arg4 Word	Arg5 Word
0-Control	1-Response	Control Response 0-No 1-Pending 2-Release 3-Query 4-Reset 127 -Error 255-Timeout	Monitor ID 1-65535	Ctrl ID 0-65535 *	Controller class	Camera ID 1-65535
1-X Point	1- Response	Camera Choice Response 0-No 1-Pending 2-Release 3-Query 4-Reset 127-Error	Monitor ID 1-65535	Ctrl ID 1-65535	Camera ID 1-65535	

* The value 65535 corresponds to "my" controller ID

8-Alarm events

To receive:

Main Cmd Byte	SubCmd Byte	Arg1 Word	Arg2 Word	Arg3 Word	Arg4 Word	Arg5 Word
0-Control	3-Alarm Off	Priority	Type	Box	Pin	Device Class
	4-Alarm On	Priority 1-49 High-low	Type	Box	Pin	Device Class

9-Relay box commands:

To transmit.

Main Cmd Byte	SubCmd Byte	Arg1 Word	Arg2 Word	Arg3 Word	Arg4 Word	Arg5 Word
0-Control	0-Request	CTRL Priority 50-99 High-low	Camera ID 1-65535			
	2-Release		Camera ID 1-65535			
7-Auxrelay	AUX cmd 0-Off 1-On	Aux no. 1-255, 0-Query				

To receive.

Main Cmd Byte	SubCmd Byte	Arg1 Word	Arg2 Word	Arg3 Word	Arg4 Word	Arg5 Word
0-Control	1-Response	Control Response 0-No 1-Pending 2-Release 3-Query 4-Reset 127-Error 255-Timeout	Camera ID 1-65535	Ctrl ID 0-65535 *	Controller class	
7-Auxrelay	0-Status	16 Bit-status				

* The value 65535 corresponds to “my” controller ID

Interfacing to the I151SX-PCIF

Getting started:
The I151SX-PCIF.

To use your I151SX-PCIF unit with your System-X, the files described below must be downloaded to the unit:

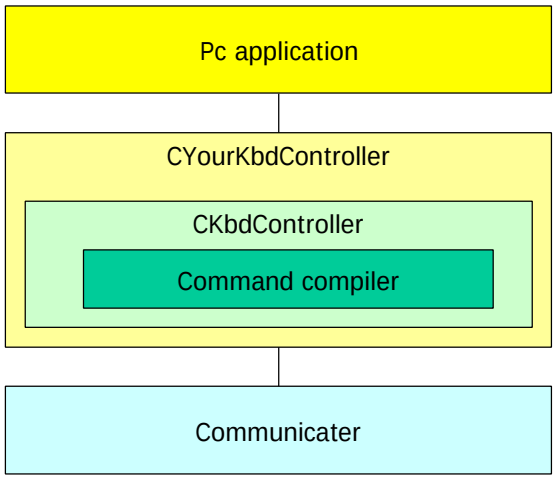
- The Neuron files, I151SX-PCIF.xif and I151SX-PCIF.nxe.
- The host file I151SX-PCIF.mot

See the S111SX NodeManager Installation Manual on how to download the files.

The PC application.

The PC application may use the API supplied on the installation CD to communicate with the I151SX-PCIF in the I151SX-PC samples directory.

Sample applications show how the API is used through inheritance of the CKBDController class, i.e. CYourKBDController.



API Platform files:

File	Description
Communicator.h	RS232 driver definition
Communicator.cpp	RS232 driver implementation
CommandCompiler.h	Protocol disassembler definition
KBDController.h	The controller class definition
KBDController.cpp	The controller class implementation[2]

Application sample files:

File	Description
I151SXPC_PCIF Test.h	Main application definition
I151SXPC_PCIF Test.cpp	Main application implementation
I151SXPC_PCIF TestDlg.h	Main dialog definition
I151SXPC_PCIF TestDlg.cpp	Main dialog implementation

[2] When using the I151SX-PCIF **HEX_IF must** be defined in KBDController.cpp.

The PCIF-A protocol frame.

The I151SX-PCIF command protocol is an ASCII protocol, thus providing for flow-control.

The format of the protocol frame is as follows:

STX	Length	Data0	• • • • • • • •	DataN	Checksum

The value of the STX is 2; the length is the number of bytes equal to STX + 2*(Length + Data), i.e. the amount of bytes shown in dark grey above times two plus one. The checksum is a simple sum of these bytes. The checksum is calculated in one byte, and the value is transmitted as ASCI in two bytes.

Furthermore the I151SX-PCIF command protocol has an additional keyboard array index for internal reference. The value is in Data0.

Keyboard instance ID	Command

The index value 0xff corresponds to general setup of the I151SX-PCIF box, rather than command requests/ responses for a given keyboard instance within the device.

To the external control network the I151SX-PCIF has a range of four keyboards control ID's assigned.

Commands

Command format:

Byte	Data
0	Device command
1	Main Command
2	Sub Command
3/ 3-4 MSB,LSB	Arg1 (Byte or Word value)
4/ 5-6 MSB,LSB	Arg2 (Byte or Word value)
5/ 7-8 MSB,LSB	Arg3 (Byte or Word value)
6/ 9-10 MSB,LSB	Arg4 (Byte or Word value)
7/ 11-12 MSB,LSB	Arg5 (Byte or Word value)

Description of device commands.

The Device Command values are:

Value	Device command
0	System command
1	Camera command
2	Monitor command
8	Alarm event
9	Relay box command
10	DVR command

0 - System commands.

To transmit:

Main Cmd Byte	SubCmd Byte	Arg1 Word	Arg2 Word	Arg3 Word	Arg4 Word	Arg5 Word
0-Service	0-Host Ready	0-Query				
	1-Control ID query					
	2-LogOnConf	Priority 50-99 High-low				
Main Cmd Byte	**SubCmd Byte**	**Arg1 Byte**	**Arg2 Byte**	**Arg3 Byte**	**Arg4 Byte**	Arg5 Byte
1-Setup	0-Users 1-User profiles 2-Monitors 3-Cameras	ACK# 1-254 0 upload req				
2-System msg (Described later in this paragraph)						

Re. Command 0/0, the I151SX-PCIF queries the host ready status only.

The setup command, transfer larger quantities of data as packets in a stop-and-go managed protocol. This implies that the receiver, i.e., the pc must send acknowledge for each packet, by re-transmitting the received packet, with correct ACK# corresponding to the received Seq#.

To receive:

Main Cmd Byte	SubCmd Byte	Arg1 Word	Arg2 Word	Arg3 Word	Arg4 Word	Arg5 Word
0-Service	0-Host Ready	0-Not ready 1-Ready				
Main Cmd Byte	**SubCmd Byte**	**Arg1 Byte**	**Arg2 Byte**	**Arg3 Byte**	**Arg4 Byte**	**Arg5 Byte**
1-Setup	0-Users 1-User profiles 2-Monitors 3-Camaras	Seq # 1-254 0 Notify 255 EOF	Data			
Main Cmd Byte	**SubCmd Byte**	**Arg1 Word**	**Arg2 Word**	**Arg3 Word**	**Arg4 Word**	**Arg5 Word**
2-System msg						
128-Control ID config	1-response	First Control ID	Last Control ID			

To send/receive:

Main Cmd Byte	SubCmd Byte	Arg1 Word	Arg2 Word	Arg3 Word	Arg4 Word	Arg5 Word	Arg6 Word	Arg7 Word
2-System message	**0**	**0**	**1**	**Sender**	**Hours=Arg/100 Min=Arg%100**	**Date=Arg/100 Month=Arg%100**	**Year =Arg/100 Year=Arg%100**	**Dateformat/256 WeekDay%256**

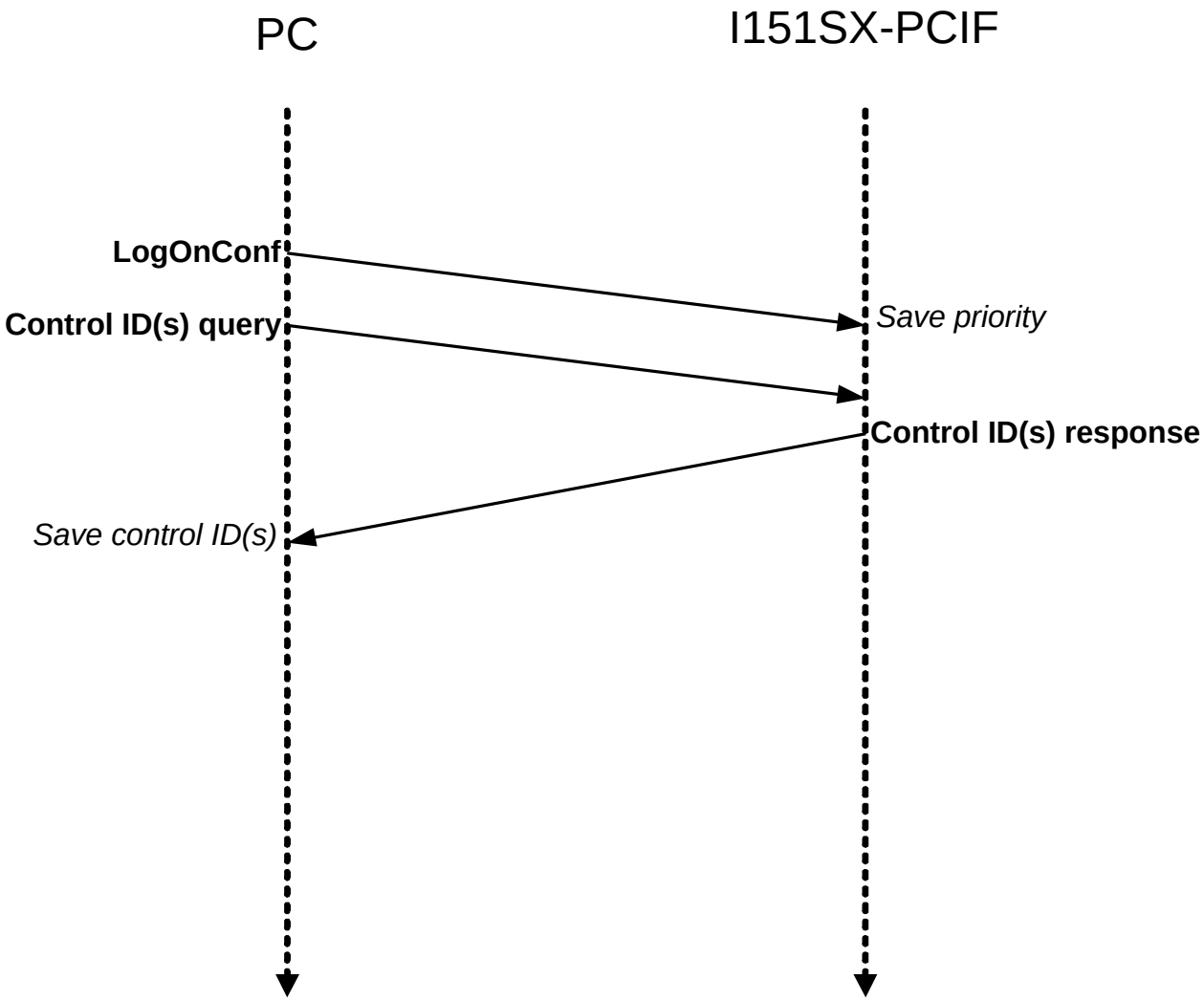
DateFormat	
0	**DD/MM/YY**
1	**YY/MM/DD**
2	**YY/DD/MM**
3	**MM/DD/YY**
4	**DD-MM-YY**
5	**YY-MM-DD**
6	**YY-DD-MM**
7	**MM-DD-YY**

Sender	
1	Master Master matrix (on receive)
2	Master matrix (not used)
3	Keyboard (on send)

Main Cmd Byte	SubCmd Byte	Arg1 Word	Arg2 Word	Arg3 Word	Arg4 Word	Arg5 Word	Arg6 Word
2-System message	0	0	2-Request time				
			27-Illegal sequence	MonID	Seq	20	ID
			40-Clear alarm	MonID	KbdID	AlarmID	
			41-View previous Alarm on Mon Queue	MonID	KbdID	AlarmID	
			42-View next Alarm on Mon Queue	MonID	KbdID	AlarmID	
			43-Select alarm	MonID	KbdID	AlarmID	
			50-Toggle text state on Mon	MonID			
			72-Start macro	Macro	Priority	20	ID
			77-Stop macro	Macro	Priority	20	20
			80-Hold macro	Macro	Priority	20	20
			90-Start sequence	MonID	Seq	20	20
			91-Hold Seq	MonID	Don' t Care	20	20

System messages are received from/ transmitted to the control network, without the I151SX-PCIF verifying the data.

Synchronisation on reset:



Example:

Log on with priority 0x32:

2 0F FF 00 00 02 00 32 EB

Query control ID's:

2 0B FF 00 00 01 21

Query ID response:

2 13 FF 00 80 01 00 01 00 04 A0

Set time to 13:15 date 12 January 2005 in date format 1 (YY/MM/DD).

Hour in hex = 0x0D
Minute in hex = 0x0F

Date in hex = 0x0C
Month in hex = 0x01

Year first two digits (20) in hex = 0x14
Year last two digits (05) in hex = 0x05

Date format in hex = 0x01
Weekday Wednesday (day 3) in hex = 0x03

2 27 FF 00 02 00 00 00 00 01 00 03 0D 0F 0C 01 14 05 01 03 A9

1- Camera commands.

Camera control command must be requested and granted prior to camera commands.

To transmit.

Main Cmd Byte	SubCmd Byte	Arg1 Word	Arg2 Word	Arg3 Word	Arg4 Word	Arg5 Word
0 – Control	0 – Request	CTRL Priority 50-99 High-low	Camera ID 1-65535			
	2- Release		Camera ID 1-65535			
Main Cmd Byte	**Arg1 Byte**	**Arg2 Byte**	**Arg3 Byte**	**Arg4 Byte**	**Arg5 Byte**	**Arg6 Byte**
1- PTZ	PAN cmd 0-Stop 1-Right 2-Left	Pan Speed 1-100	TILT cmd 0-Stop 1-Up 2-Down	Tilt Speed 1-100	ZOOM cmd 0-Stop 1-Tele 2-Wide	Zoom speed 1-100
2 - Lens	IRIS cmd 0-Hold 1-Open 2-Close 3-Auto	FOCUS cmd 0-Hold 1-Near 2-Far 3-Auto	BACKLIGHT 0-Off 1-On			
Main Cmd Byte	**SubCmd Byte**	**Arg1 Word**	**Arg2 Word**	**Arg3 Word**	**Arg4 Word**	**Arg5 Word**
3 - Position	POS cmd 0-Save 1-Call 2-Sequence	POS NO. 1-255				
4 - Autopan	1-Limit	1-Left Lim. 2-Right Lim.				
	2-Start	1-Start	Percentage speed			
7 – Auxrelay	AUX cmd 0-Off 1-On	Aux no. 1-255, 0-Query				

To receive.

Main Cmd Byte	SubCmd Byte	Arg1 Word	Arg2 Word	Arg3 Word	Arg4 Word	Arg5 Word
0- Control	1- Response	Control Response 0 = No 1 = Pending 2 = Release 3 = Query 4 = Reset 127 = Error 255 = Timeout	Camera ID 1-65535	Ctrl ID 0-65535 *	Controller class	
3 - Position	POS response 0-Saved 1-Called	POS No. 1-255				
7-Auxrelay	0-Status	16 Bit-status				

* The value 65535 corresponds to “my” controller ID

Example:

Keyboard 2 index 1 priority 0x32 request to control camera 3:

2 11 01 01 00 00 00 32 00 03 0D

And the response from the camera:

2 17 01 01 00 01 00 00 00 03 FF FF 88

Keyboard 2 index 1 sends pan left 50% tilt up 25% zoom tele 100%:

2 13 01 01 02 32 01 19 01 64 85

2-Monitor commands.

Monitor control command must be requested and granted prior to monitor commands.

To transmit.

Main Cmd Byte	SubCmd Byte	Arg1 Word	Arg2 Word	Arg3 Word	Arg4 Word	Arg5 Word
0-Control	0-Request	CTRL Priority 50-99 High-low	Monitor ID 1-65535			
	2-Release	Don't care	Monitor ID 1-65535			
1-X Point	0-Request	CTRL priority 1-100	Monitor ID 1-65535	Ctrl ID 1-65535	Camera ID 1-65535	
	2-Release	Don't care	Monitor ID 1-65535	Ctrl ID 0-65535	Camera ID 1-65535	

To receive.

Main Cmd Byte	SubCmd Byte	Arg1 Word	Arg2 Word	Arg3 Word	Arg4 Word	Arg5 Word
0-Control	1-Response	Control Response 0-No 1-Pending 2-Release 3-Query 4-Reset 127 -Error 255-Timeout	Monitor ID 1-65535	Ctrl ID 0-65535 *	Controller class	Camera ID 1-65535
1-X Point	1- Response	Camera Choice Response 0-No 1-Pending 2-Release 3-Query 4-Reset 127-Error	Monitor ID 1-65535	Ctrl ID 1-65535	Camera ID 1-65535	

* The value 65535 corresponds to "my" controller ID

Example:

Keyboard 2 index 1 priority 0x32 request to control monitor 4:

2 13 01 02 00 00 00 32 00 04 72

And the response from the monitor (camera 1 shown on monitor 4):

2 1F 01 02 00 01 00 00 00 04 FF FF 98 0C 00 01 3E

Keybord 2 index 1 request camera 7 to be shown:

2 1B 01 02 01 00 00 32 00 04 00 02 00 07 0B

And the response from the monitor:

2 1B 01 02 01 01 00 00 00 04 00 02 00 07 07

8-Alarm events

To receive:

Main Cmd Byte	SubCmd Byte	Arg1 Word	Arg2 Word		Arg3 Word	Arg4 Word	Arg5 Word
0-Control	3-Alarm Off	Priority	Type	Addr	Box	Pin	Device Class
	4-Alarm On	Priority 1-49 High-low	Type	Addr	Box	Pin	Device Class

Alarm types	Type	Addr	Addr. format
0	Alarm	0	Global Addr.
1	Intruder	1	Box,Pin
2	Door	2	Matrix, Box.Pin
3	Key card		
4	Occupan		
5	Tamper		
6	Video loss		
7	Sync loss		

Device class	Product name
0x9600	Matrix X
0x9601	Matrix XN
0x960B	Remote System 1000
0x9700	DC1
0x9701	AC1
0x9702	AC2
0x9800	Keyboard X
0x980A	Keyboard emul
0x980B	I151SX-PC interface
0x980C	I151SX-PC4 interface
0x9900	Alarm 128
0x9A00	Lon box
0x9A0B	Serial alarms
0x9B00	Erna converter
0x9B01	Dome protocol converter
0x9C00	DVR interface

Example:

Keyboard 2 receives an alarm (type = 0) ON event priority 0x10 referenced by Box = 2, Pin = 3 from a I141DX (class ID = 9A00):

2 1F 01 08 00 04 00 10 00 01 00 02 00 03 9A 00 E7

9-Relay box commands:

Relay box or camera control command must be requested and granted prior to relay commands.

To transmit.

Main Cmd Byte	SubCmd Byte	Arg1 Word	Arg2 Word	Arg3 Word	Arg4 Word	Arg5 Word
0-Control	0-Request	CTRL Priority 50-99 High-low	Box ID 1-65535			
	2-Release		Box ID 1-65535			
7-Auxrelay	AUX cmd 0-Off 1-On	Aux no. 1-255, 0-Query				

To receive.

Main Cmd Byte	SubCmd Byte	Arg1 Word	Arg2 Word	Arg3 Word	Arg4 Word	Arg5 Word
0-Control	1-Response	Control Response 0-No 1-Pending 2-Release 3-Query 4-Reset 127-Error 255-Timeout	Box ID 1-65535	Ctrl ID 1-65535 *	Controllers Device class	
7-Auxrelay	0-Status	16 Bit-status				

* The value 65535 corresponds to “my” controller ID

Example:

Keyboard 2 index 1 priority 0x32 request to control relay box 5:

2 13 01 09 00 00 00 32 00 05 7A

And the response from the relay box:

2 1B 01 09 00 01 00 00 00 05 00 02 98 0C 2B

Keyboard 2 index 1 request relay output 1 ON:

2 0F 01 09 07 01 00 01 CB

Appendix – Checksum calculation

B protocol:

```
byte sendCommand(byte *CmdData, byte Len)
{
  byte data[40], i, chksum;

  memcpy(&data[2], CmdData, Len);
  data[0] = STX;
  data[1] = Len + 2;
  chksum = 0;
  for (i = 0; i < Len+2; i++)
    chksum += data[i];
  data[i] = chksum;
  return sendUARTData0(data, Len+3);
}
```

A protocol:

```
byte sendCommand(byte *CmdData, unsigned long int Len)
{
  word status;
  unsigned long int i, j;
  byte data[MAX_DATA_SIO], chksum, dummy;

  status = 0;

  if (Len >= (MAX_DATA_SIO - 5)/2)
    return status;

  data[0] = STX;
  chksum = data[0];
  dummy = (Len*2)+3;
  data[1] = toASCIIHex(dummy, Hi);
  chksum += data[1];
  data[2] = toASCIIHex(dummy, Lo);
  chksum += data[2];

  for (i = 0, j = 3; i < Len; i++, j+=2)
  {
    dummy = CmdData[i];
    data[j] = toASCIIHex(dummy, Hi);
    chksum += data[j];
    data[j+1] = toASCIIHex(dummy, Lo);
    chksum += data[j+1];
  }

  data[j++] = toASCIIHex(chksum, Hi);
  data[j++] = toASCIIHex(chksum, Lo);

  if (j < MAX_DATA_SIO)
  {
    status = sendUARTData0(data, j);
  }
  else
    return status;
}
```

Appendix- dataconversion

From byte to hex in ASCII format:

```
byte toASCIIHex(byte N, byte Order)
{
  byte tmp;
  tmp = (N >> Order) & 0x0f;

  if (tmp <= 9)
    return '0'+tmp;
  else
    if ((tmp >= 10) && (tmp <= 15))
       return 'A'-10+tmp;
   else
       return 0;
}
```

From hex in ASCII format to byte:

```
byte fromASCIIHex(byte *Value)
{
   byte tmp;

   if ((Value[0] <= '9') && (Value[0] >= '0'))
      tmp = ((Value[0]-'0') & 0xf) << 4;
   else
      if ((Value[0] <= 'F') && (Value[0] >= 'A'))
         tmp = ((Value[0]-'A'+10) & 0xf) << 4;
      else
         tmp = 0;

   if ((Value[1] <= '9') && (Value[1] >= '0'))
      tmp += ((Value[1]-'0') & 0xf);
   else
      if ((Value[1] <= 'F') && (Value[1] >= 'A'))
         tmp += ((Value[1]-'A'+10) & 0xf);

   return tmp;
}
```