PARADOX®
SECURITY SYSTEMS

E series
SERIAL INTERFACE PROTOCOL
Version 5.00 and UP

MAGELLAN™

October 27th, 2009

CONFIDENTIAL

Table of content

FUNCTIONAL DESCRIPTION

This document explains how the serial protocol works on the E control panels. It contains all the information that the panel sends/receives through its onboard serial connector and shows how to interpret this information.

The panel‘s onboard serial connector is a 5V serial communicator. Connection is made via a 307 / IP100 / PCS adapter cable that fits directly on the control panel (see figure 1).

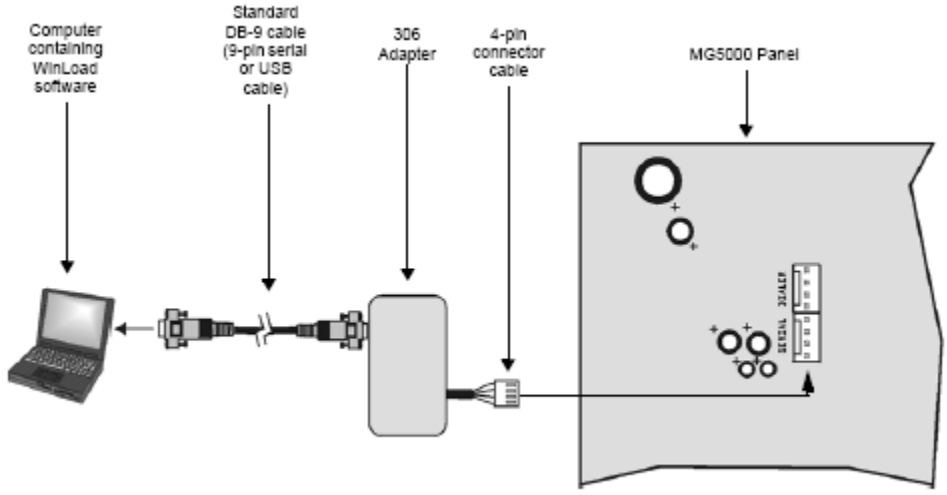

Figure 1

FEATURES

5V communication

The serial interface is an asynchronous 5V serial communicator.

Communication packet

In this document, Bit 7 is the most significant bit in the byte and Bit 0 is the least significant bit in the byte.

Communication speed

Control panel’s onboard serial connector is set to communicate at 9,600 Baud, 1 start bit, 8 data bits, 1 stop bit and no parity.

PARADOX NETWORK PROTOCOL LAYER (CONTAINER)

1.0 BASIC FRAME LAYER

This is the container of all following protocols. It is similar to what the Ethernet layer does in internet communication layers. It provides a mean to validate a packet after it was transmitted on a wire. Of course, in case of a TCP/IP transmission, it becomes redundant but since we support many communication channels (IP, SERIAL, MODEM) then it becomes obvious that the information must be consistent for both peers to understand each other no matter what channel was used.

1.1 BYTE ORDER

Every time a **WORD** or a **DWORD** value is passed from one point to another, the value should be packed as **LITTLE ENDIAN** (reverse order).

Example:

16 bits value 0x1122 would be stored as [0x22, 0x11]
32 bits value 0x11223344 would be stored as [0x44, 0x33, 0x22, 0x11]

1.2 HEADER

SOF	PID	LEN	SRL	DSL	SRC	DST	DAT . . .	CRC

HEADER	
[SOF] Start Of Frame = 0xDE	1 byte
[PID] Protocol ID	2 bytes
[LEN] Packet Length	2 bytes
[SRL] Source Length	1 byte
[DSL] Destination Length	1 byte
[SRC] Source Address	var
[DST] Destination Address	var
[DAT] DATA (Based on PID)	var
[CRC] Circular Redundancy Check	2 bytes

1.3 [SOF] Start of Frame Byte

This byte serves as a marker to identify a possible start of frame.

1.4 [PID] Protocol Identifier

The PID identifies a protocol and its structure. To avoid having incompatibilities with old units or problems with data interpretation we decided that the PID(s) won't have a version and it WILL NOT be possible to change a PID after it's been released.

1.5 [LEN] Packet Length

The packet length is the total number of bytes for the whole frame including the header of the basic frame.

1.6 [SRL/DSL] Source and Destination Addresses Length

These are used to specify if the source and destination addresses are used. If the value is set to **0** than it means the corresponding address field is not present.
E series = always 0.

1.7 [CRC] Cyclic Redundancy Check

The protocol will be using the CCITT Polynomial for CRC16 calculation (x16+x15+x2+x1). The following routine is a typical implementation of that algorithm:

```
#define GENPOLY 0xA001 // Reverse Polynominal

/* The calling routine must pass a byte value to perform the CRC calculation,
and a pointer to an unsigned int location for the storage of the generated CRC
Word (2 bytes). It is the calling function responsibility to initialize the
crc_accum location to 0xFFFF prior to calling this routine for the first time.
Only initialize the location after the sequence of character used in the CRC
generated are completed. */

void Crc_Crc_Calc(BYTE byNew_Data,UWORD *pwCrc_Accum)
{
  BYTE i;

  pwCrc_Accum->by.lo = pwCrc_Accum->by.lo ^ byNew_Data;

  // Loop 8 times to test each bit 0f the new character

  for (i=0; i<8; i++)
  {
    if(pwCrc_Accum->by.lo & (BYTE)0x01)
    {
      pwCrc_Accum->w >>= 1;
      pwCrc_Accum->w ^= (WORD)GENPOLY;
   }
   else
   {
     pwCrc_Accum->w >>= 1;
   }
  }
}
```

1.8 [SRC/DST] Source and Destination Addresses

The source and destination addresses if the source and destination length are not equal to 0.
E series = always 0.

2.0 Live events Protocol (PID = 0004)

The Paradox Network Protocol (PNP) is used as a container to the original E/MG/SP live event protocol.

2.1 Structure of one (1) original live event Commands

SOF	PID	LEN	SRL	DSL	DAT	CRC

[SOF] Start Of Frame = 0xDE	1 byte	0xDE
[PID] Protocol ID (low part)	1 byte	0x04
[PID] Protocol ID (high part)	1 byte	0x00
[LEN] Packet Length (low part)	1 byte	0x2E
[LEN] Packet Length (high part)	1 byte	0x00
[SRL] Source Length	1 byte	0x00
[DSL] Destination Length	1 byte	0x00
[DAT] Original Live Event Protocol	37 bytes	--
[CRC] Circular Redundancy Check (low part)	1 byte	--
[CRC] Circular Redundancy Check (high part)	1 byte	--

2.2 Structure of multiple original live event Commands in the same packet

The flexibility of the Paradox Network Protocol allows the possibility to pack multiple commands in the same transmission. For instance, in E panels, up to four (4) live events can be sent in the same transmission. The length of the whole packet will vary according the number of commands included in the packet. The following shows the structure of a multiple commands packet. In this example, four (4) commands are sent in the same packet.

SOF	PID	LEN	SRL	DSL	DAT1	DAT2	DAT3	DAT4	CRC

[SOF] Start Of Frame = 0xDE	1 byte	0xDE
[PID] Protocol ID (low part)	1 byte	0x04
[PID] Protocol ID (high part)	1 byte	0x00
[LEN] Packet Length (low part)	1 byte	0x9D
[LEN] Packet Length (high part)	1 byte	0x00
[SRL] Source Length	1 byte	0x00
[DSL] Destination Length	1 byte	0x00
[DAT1] Original Live Event – First (1st) event	37 bytes	--
[DAT2] Original Live Event – Second (2nd) event	37 bytes	--
[DAT3] Original Live Event – Third (3rd) event	37 bytes	--
[DAT4 Original Live Event – Fourth (4th) event	37 bytes	--
[CRC] Circular Redundancy Check (low part)	1 byte	--
[CRC] Circular Redundancy Check (high part)	1 byte	--

2.3 Command 0xE0 : Live Event command.

When an event is generated, the panel will send the following command on the serial port. The Event Group Number and the Event Sub-Group Number are listed in the Event Description table of the E/MG/SP Programming Guide.

Panel sends live event command to PC

Byte	Value	Description
00	0xEX	High nibble Command Low nibble Bit 3: Event report enable(0) / disable(1) Bit 2: System in alarm. Bit 1: Winload connected. Bit 0: NEware connected.
01	0xXX	Century
02	0xXX	Year
03	0xXX	Month
04	0x00	Day
05	0xXX	Hour
06	0xXX	Minute
07	0xXX	Event Group Number
08	0xXX	Event Sup-group Number
09	0xXX	Partition Number
10	0xXX	Module Serial Number digit 1 and 2
11	0xXX	Module Serial Number digit 3 and 4
12	0xXX	Module Serial Number digit 5 and 6
13	0xXX	Module Serial Number digit 7 and 8
14	0xXX	Label type 0 = Zone label 1 = User label 2 = Partition label 3 = PGM label 4 = Bus module label 5 = Wireless repeater label 6 = Wireless keypad label
15	0xXX	Module/User/Partition Label byte 00
16	0xXX	Module/User/Partition Label byte 01
17	0xXX	Module/User/Partition Label byte 02
18	0xXX	Module/User/Partition Label byte 03
19	0xXX	Module/User/Partition Label byte 04
20	0xXX	Module/User/Partition Label byte 05
21	0xXX	Module/User/Partition Label byte 06
22	0xXX	Module/User/Partition Label byte 07
23	0xXX	Module/User/Partition Label byte 08
24	0xXX	Module/User/Partition Label byte 09
25	0xXX	Module/User/Partition Label byte 10
26	0xXX	Module/User/Partition Label byte 11
27	0xXX	Module/User/Partition Label byte 12
28	0xXX	Module/User/Partition Label byte 13
29	0xXX	Module/User/Partition Label byte 14
30	0xXX	Module/User/Partition Label byte 15
31	0x00	Event Unique ID High
32	0x00	Event Unique ID Low
33	0x00	Event Pointer In
34-35	0x00	N/U
36	0xXX	Checksum