1. Implemented CM11 protocol

1.1. *Housecodes and Device Codes.*

The housecodes and device codes range from A to P and 1 to 16 respectively although they do not follow a binary sequence. The encoding format for these codes is as follows

Housecode	Device Code	Binary Value
A	1	0110
B	2	1110
C	3	0010
D	4	1010
E	5	0001
F	6	1001
G	7	0101
H	8	1101
I	9	0111
J	10	1111
K	11	0011
L	12	1011
M	13	0000
N	14	1000
O	15	0100
P	16	1100

1.2. *Function Codes.*

Function	Binary Value
All Units Off	0000
All Lights On	0001
On	0010
Off	0011
Dim	0100
Bright	0101
All Lights Off	0110
Extended Code	0111 (not supported)
Hail Request	1000 (not supported)
Hail Acknowledge	1001 (not supported)
Pre-set Dim (1)	1010 (not supported)
Pre-set Dim (2)	1011 (not supported)
Extended Data Transfer	1100 (not supported)
Status On	1101 (not supported)
Status Off	1110 (not supported)
Status Request	1111 (not supported)

1.3. *Serial Parameters.*

The serial parameters for communications between the interface and PC are as follows:

Baud Rate: 4,800bps

Parity:	None
Data Bits:	8
Stop Bits:	1

Cable connections:

Signal	**DB9 Connector**
SIN	Pin 2
SOUT	Pin 3
GND	Pin 5
RI	Pin 9

where:	SIN	Serial input to PC (output from the interface)
	SOUT	Serial output from PC (input to the interface)
	GND	Signal ground
	RI	Ring signal (input to PC)

1.4. X10 transmission.

An X-10 transmission from the PC to the interface typically refers to the communication of a Housecode and Device Code combination or the transmission of a function code. The format of these transmissions is:

	PC	Interface
2 bytes	Header:Code	
1 byte		checksum
1 byte	Acknowledge	
1 byte		interface ready to receive

This format is typical of all transmissions between the PC and the interface with the difference being in the first transmission from the PC.

1.4.1. Header-Code.

The Header:Code combination is configured thus:

```
Header:  7   6   5   4   3   2   1   0
         < Number of Dims>   1  F/A E/S
```

Where:
- Number of Dims is a value between 0 and 22 identifying the number of dims to be transmitted (22 is equivalent to 100%)
- Bit 2 is always set to '1' to ensure that the interface is able to maintain synchronization.
- F/A defines whether the following byte is a function (1) or address (0).
- E/S defines whether the following byte is an extended transmission (1) or a standard transmission (0).

```
Code:     7   6   5   4   3   2   1   0
Address: < Housecode >   <Device Code>
Function:< Housecode >   < Function  >
```

Note: the function only operates for devices addressed with the same Housecode.

1.4.2. Interface Checksum and PC Acknowledge

When the interface receives a transmission from the PC, it will sum all of the bytes, and then return a byte checksum. If the checksum is correct, the PC should return a value of 0x00 to indicate that the transmission should take place. If however, the checksum is incorrect, then the PC should again attempt to transmit the Header:Code combination and await a new checksum.

1.4.3. Interface Ready to Receive.

Once the X-10 transmission has taken place (and this may be quite time consuming in the case of Dim or Bright commands) the interface will send 0x55 to the PC to indicate that it is in a 'ready' state.

1.4.4. Example.

```
PC           Interface   Description
0x04,0x66                Address A1
             0x6a        Checksum ((0x04 + 0x66)&0xff)
0x00                     OK for transmission.
             0x55        Interface ready.

0x04,0x6e                Address A2
             0x72        Checksum ((0x04 + 0x6e)&0xff)
0x00                     OK for transmission.
             0x55        Interface ready.

0x86,0x64                Function: A Dim 16/22*100%
             0xe0        Incorrect checksum.
0x86,0x64                Function re-transmission
             0xea        Checksum ((0x86 + 0x64)&0xff)
0x00                     OK for transmission.
             0x55        Interface ready.
```

This transmission will address lamp modules A1 and A2, and then dim them by 72%. Note multiple addresses cannot be made across housecodes, i.e. A1, B2 Dim 72% is not valid, and would result in B2 being dimmed by 72%.

1.5. Extended X-10 Transmission.

Extended X-10 transmission is simply an extension of the protocol to allow two additional bytes of extended data to be transmitted. In this case, the protocol may be shown as:

```
        PC                       Interface
4 bytes Header:Code:Data:Command
1 byte                           checksum
1 byte  Acknowledge
1 byte                           interface ready to receive
```

The header for an extended transmission is always:

```
Header:  7   6   5   4   3   2   1   0
         0   0   0   0   0   1   1   1
```

Bits 7 to 3 are always zero because the dim level is not applicable to extended transmissions.
Bit 2 must be set to '1' as in all PC header transmissions.
Bit 1 is set to '1' as the extended transmission is always a function.
Bit 0 is set to '1' to define an extended transmission rather than a standard transmission.

The code byte is:

```
Code:       7   6   5   4   3   2   1   0
Function:   < Housecode >   0   1   1   1
```

Again, the housecode must be the same as any previously addressed modules, and for extended data, the function code must be 0111.

Finally, the data and command bytes may take any value between 0x00 and 0xff.
Note that the checksum is one byte and is defined as:

checksum = (header + code + data + command)&0xff

1.6. X-10 Reception.

Whenever the interface begins to receive data from the power-line, it will immediately assert the serial ring (RI) signal to initiate the wake-up procedure for the PC. Once the data reception is complete, the interface will begin to poll the PC to upload its data buffer (maximum 10 bytes). If the PC does not respond, then the interface's data buffer will overrun, and additional data will not be stored within the buffer.

1.6.1. Interface Poll Signal.

In order to poll the PC, the interface will continually send:

```
Poll:       7   6   5   4   3   2   1   0
Value:      0   1   0   1   1   0   1   0       (0x5a)
```

This signal will be repeated once every second until the PC responds.

1.6.2. PC Response to the Poll Signal.

To terminate the interface's polling and initiate the data transfer, the PC must send an acknowledgment to the interface's poll signal. This acknowledgment is:

```
Poll:       7   6   5   4   3   2   1   0
Value:      1   1   0   0   0   0   1   1       (0xc3)
```

Notice that bit #2 of the PC transmission is not set, indicating that this cannot be the beginning of a transmission from the PC.

1.6.3. Interface Serial Data Buffer.

The buffer consists of 10 bytes defined as follows:

Byte	Function
0	Upload Buffer Size
1	Function / Address Mask
2	Data Byte #0
3	Data Byte #1
4	Data Byte #2
5	Data Byte #3
6	Data Byte #4
7	Data Byte #5
8	Data Byte #6
9	Data Byte #7

The interface will only upload the specified number of bytes within the buffer, and will not default to uploading 10 bytes in every transmission. The number of bytes to receive is thus specified in byte 0 of the transmission.

The function address mask indicates whether the following 8 bytes should be interpreted as an address or as a function. The position of the bit in the mask corresponds to the byte index within the data buffer. If the bit is set (1), the data byte is defined as a function, and if reset (0), the byte is an address.

The data bytes are in the same format as for the Code byte in the X-10 transmissions (i.e. Housecode:Device Code or Housecode:Function).

Note that once the data buffer has been uploaded, there is no acknowledgment from the PC to the interface as the contents of the serial data buffer will have been changed. This will not cause a problem as this is simply informing the PC of the external status, rather than controlling a device (as in the case of the PC transmission) which may have safety implications.

1.6.4. Dim or Bright.

After a dim or bright code, the PC will expect the following byte to be the change in brightness level. An X-10 module has 210 discrete brightness levels, and therefore this byte will be equivalent to a brightness change of n/210*100%.

1.6.5. Extended Code.

Extended code is processed in a similar way to Dim and Bright, except that the PC will expect two bytes, which are the Data and Command bytes.

1.6.6. Example.

```
PC    Interface   Description
      0x5a        Poll from interface.
0xc3              'PC Ready' Response from PC
      0x05        5 bytes to follow
```

```
0x04        xxxx x100-> byte 0,1 addresses, 2 function
0xe9        B6
0xe5        B7
0xe5        B Bright
0x58        0x58/210 * 100%
```

This transmission will wake the computer, and then indicate that a transmission of length 5 bytes will occur, data bytes 0 and 1 are addresses and byte 2 is a bright function, which means that the following byte is the change in brightness level.

1.7. Macro.

Not supported

1.8. EEPROM Code.

Not supported but a response is given.

```
PC          Interface   Description

0xfb                    EEPROM download start byte (1st block of data)
0x00
0x00                    EEPROM address 0x0000
0x00
0x0c                    EEPROM offset to macro initiators 0x000c
0x3e                    Day mask x 0111110 (.FTWTM.)
0x00                    Start day [0..7]
0x6d                    Stop day [0..7]
0x49                    (Event start time, Event stop time) x 120 min
0x00                    Start day range [8], Event start time [0..6]
0x80                    Stop day range [8], Event stop time [0..6]
0x00                    Start macro pointer [8..11],
                             Stop macro pointer [8..11]
0x1d                    Start macro pointer [0..7]
0x22                    Stop macro pointer [0..7]
                        Summary:    Start day: 0x000  (Jan 1)
                                    Stop day: 0x16d   (Dec 31)
                                    Start time: 4 x 120mins = 08:00
                                    Stop time: 9 x 120mins = 18:00
                                    Start macro pointer: 0x01d
                                    Stop macro pointer: 0x022
0xff                    Timer table delimiter
0x6a                    Macro initiator house and device code (A4)
0x80                    Macro function (On)
0x11                    Macro pointer (0x011)
0xff
            0xb8        Checksum from the interface
0x00                    Checksum correct
            0x55        Programming complete
```

1.9. Serial Ring Disable

Not supported but a response is given.

If may be required, for the sake of 'trouble-shooting' to disable the serial ring (RI) signal, although undesirable as macros held within the computer will not operate, nor will the computer be able to track the system status.

The following protocol will allow the serial ring (RI) signal to be enabled and disabled:

Enable Ring:

PC	Interface	Description
0xeb		Enable the ring signal
	0xeb	Checksum
0x00		Checksum correct
	0x55	Interface ready

Disable Ring:

PC	Interface	Description
0xdb		Disable the ring signal
	0xdb	Checksum
0x00		Checksum correct
	0x55	Interface ready

The default state of the serial ring (RI) signal after a power on reset is enabled.

1.10. EEPROM Address.

Not supported

1.11. Set Interface Clock.

Not supported, data is ignored.

PC	Interface	Description
0x9b		set timer download header Followed by 6 bytes: Bit:
0x00		47 to 40 Current time (seconds)
0x00		39 to 32 Cur time (min ranging from 0 to 119)
0x00		31 to 23 Cur time (hrs/2, range from 0 to 11)
0x00		23 to 16 Current year day (bits 0 to 7)
0x00		15 Current year day (bit 8)
0x00		14 to 8 Day mask (SMTWTFS)
0x00		7 to 4 Monitored house code
0x00		3 Reserved
0x00		2 Battery timer clear flag
0x00		1 Monitored status clear flag
0x00		0 Timer purge flag

1.12. Status Request.

Not supported but a response is given.

PC	Interface	Description
0x8b		Status Request
	0x00	111 to 96 Bat timer (set to 0xffff on reset)
	0x00	95 to 88 Current time (seconds)
	0x00	87 to 80 Cur time (min ranging from 0 to 119)
	0x00	79 to 72 Cur time (hrs/2, range from 0 to 11)
	0x00	71 to 63 Current year day (MSB bit 63)

```
0x00        62 to 56 Day mask (SMTWTFS)
0x00        55 to 52 Monitored house code
0x00        51 to 48 Firmware revision level 0 to 15
0x00        47 to 32 Currently addressed monitored devices
0x00        31 to 16 On/Off status of the monitored devs
0x00        15 to 0  Dim status of the monitored devices
```