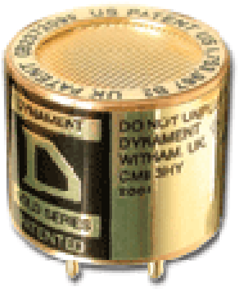
*TAKING INVENTIVE STEPS IN INFRARED....*

# Premier Sensor Communications protocol

**Issue Number:** 1.3R
**Issue Date:** 26th Nov 2007
**Document:** TDS0045
**Firmware Version:** 1.x.x




Dynament
Premier House
The Village
South Normanton
Derbyshire DE55 2DS
United Kingdom

# 1 Communications Protocol

The communications protocol used by the Premier sensor is used for communications between devices connected via an RS232 connection. This point-to-point, P2P, protocol is a frame-based protocol.

Note: the protocal allows two-way communications, read – write, between the sensor and external device, However this document only describes the read aspect of the protocol.

## 1.1 Control Byte Constants

The following control byte constants are used in the P2P protocol[1].

| | | | | |
|---|---|---|---|---|
| Read, | RD | = | 0x13 | (00010011) |
| Data Link Escape, | DLE | = | 0x10 | (00010000) |
| Write, | WR | = | 0x15 | (00010101) |
| Acknowledge, | ACK | = | 0x16 | (00010110) |
| Negative Acknowledge, | NAK | = | 0x19 | (00011001) |
| Single Data Frame, | DAT | = | 0x1A | (00011010) |
| End of Frame, | EOF | = | 0x1F | (00011111) |

## 1.2 Frame Structure

The start of a frame is indicated by a DLE byte followed by the type of frame to follow (RD, WR, ACK, NAK, DAT). The end of frame is indicated by a DLE byte followed by an EOF byte.

*Note: Each of the constants has bit 4 set and so is slip-resistant (i.e. if shifted this bit will be out of position). The values have a Hamming Distance of 2 (each code is at least 2 bits different from every other code).*

Any DLE bytes that occur between a frame's start and end are prefixed with another DLE (*byte-stuffing*).

Following the EOF is a 16-bit checksum of the entire frame, each byte is added to produce the checksum.

## 1.3 Variables

Each piece of accessible data on a device is referred to as a *Variable*. Each variable is referenced by a *Variable ID*. A *variable ID* may be any number of bytes long.

The available Variables and their corresponding Variable IDs depend on the type of device, but here are a few examples for Premier sensor:

| Purpose | Variable id | Comments |
|---|---|---|
| Live Data | 1 | read only |
| Live Data Simple | 6 | read only |

The structure of the data returned in each variable usually depends both on the type of device and the version of firmware running on the device.
Refer to section 2 for more information.

Note: Live data simple only transmits the first 8 bytes of the structure.

## 1.4 Reading a Variable

Send a read frame with the Variable ID to be read:

| DLE | RD | var-id | DLE | EOF | Csum hi | Csum lo |
|---|---|---|---|---|---|---|
| | | Byte stuffing | | | | |

Device response on success, where requested variable data < 255 bytes:

| DLE | DAT | data-len | data | DLE | EOF | Csum hi | Csum lo |
|---|---|---|---|---|---|---|---|
| | | Byte stuffing | | | | | |

Device response on failure:

| DLE | NAK | reason |
|---|---|---|

Where 'reason' is a single byte failure code, the meaning of which depends on the device type, i.e.

1 p2pNAKvarNotReadable,
2 p2pNAKvarNotWritable,
3 p2pNAKoutOfRange,
4 p2pNAKincorrectLength,
5 p2pNAKunexpectedBytes,
6 p2pNAKchecksumFailed,
7 p2pNAKincorrectVersion,
8 p2pNAKbusy,

### 1.4.1 Read example - read live data

Send the following bytes:

**DLE, RD, Variable ID, DLE, EOF, Checksum High byte, Checksum low byte i.e.**

0x10, 0x13, 0x01, 0x10, 0x1F, 0x00, 0x53

Device response on success:

DLE, DAT, Data length, Data, DLE, EOF, Checksum High byte, Checksum low byte, i.e.

| | |
|---|---|
| 0x10 | DLE |
| 0x1A | DAT |
| 0x14 | Data length |
| 0x01, 0x00 | Version |
| 0x00, 0x00 | Status flags |
| 0x00, 0x00, 0x28, 0x41 | Gas reading, 32 bit floating point - IEEE-754 format |
| 0x00, 0x00, 0x1E, 0x42 | Temperature, 32 bit floating point - IEEE-754 format |
| 0x2C, 0x04 | Detector signal |
| 0x86, 0x02 | Reference signal |
| 0x80, 0x1A, 0x09, 0xBC | Absorbance, 32 bit floating point - IEEE-754 format |
| 0x10 | DLE |
| 0x1F | EOF |
| 0x03 | Checksum high byte |
| 0xA5 | Checksum low byte |

Note 1: 0x41280000 = 10.50

Note 2: the data length may increase depending upon the sensor firmware.

The data can be read and any extra bytes can be ignored if not needed.

### 1.4.2 Read example - read live data simple

Send the following bytes:

**DLE, RD, Variable ID, DLE, EOF, Checksum High byte, Checksum low byte i.e.**

0x10, 0x13, 0x06, 0x10, 0x1F, 0x00, 0x58

Device response on success:

DLE, DAT, Data length, Data, DLE, EOF, Checksum High byte, Checksum low byte, i.e.

| | |
|---|---|
| 0x10 | DLE |
| 0x1A | DAT |
| 0x08 | Data length |
| 0x01, 0x00 | Version |
| 0x00, 0x00 | Status flags |
| 0x00, 0x00, 0x28, 0x41 | Gas reading, 32 bit floating point - IEEE-754 format |
| 0x10 | DLE |
| 0x1F | EOF |
| 0x00 | Checksum high byte |
| 0xCB | Checksum low byte |

Note 1: 0x41280000 = 10.50

# 2 VARIABLE STRUCTURES

The data transferred is byte orientated with the following sizes:

| | |
|---|---|
| Byte | 8 bits |
| Integer | 2 bytes |
| Double | 4 bytes |
| Long | 4 bytes |

## 2.1 Live data structure

The live data takes two distinct forms:

```
Struct 1{ unsigned int Version;
          unsigned int StatusFlags;
          double Reading;
          double Temperature;
          unsigned int Det;
          unsigned int Ref;
          double Fa;
} LiveData;

struct  2{ unsigned int Version;
          unsigned int StatusFlags;
          double Reading;
          double Temperature;
          unsigned int Det;
          unsigned int Ref;
          double Fa;
          long Uptime;
} LiveData;
```

Note that structure 2 has an extra variable and is 32 bytes in length, 4 more than structure 1.

The Version number remains the same even when extra variables are added to the structure. Thus if software is written to accept structure 1 but receives more data, i.e structure 2, then it simply ignores the extra bytes.

The StatusFlags field give the user additional information about the sensor as follows:

| Name | Bit |
|---|---|
| FLAG_SIGNAL_TIMEOUT | 0x0001 |
| FLAG_SIGNAL_NOISE | 0x0004 |
| FLAG_DET_LOW | 0x0040 |
| FLAG_REF_LOW | 0x0080 |
| FLAG_VMON_ERROR | 0x0800 |
| FLAG_CONFIG_CSUM | 0x1000 |
| FLAG_PRIVATE_CSUM | 0x2000 |
| FLAG_USER_EEP_CSUM | 0x4000 |
| FLAG_PROG_CSUM_ERROR | 0x8000 |

The flags are added together to form one integer result, 2 bytes. If the sensor signals are too low then the returned flag word would be 0x00C0.