Specification for sensor unit serial protocol

Revised 13 Dec 2015

1 General

Two types of messages are defined for the sensor unit protocol. Control messages are sent by the controller to the sensor unit, while data messages are sent by the sensor unit to the controller. Data messages should be sent on a regular basis, except when the sensor unit receives a control message telling it to stop (see Section 2.4). The sensor unit must be prepared to accept a control message at any time. All multi-byte fields described herein should be interpreted as little-endian. Each individual byte is big-endian, and all fields should be expected in the order they are listed. The UART should be configured at 9600/8-N-1 with no flow control.

2 Control messages

All control messages are three bytes; each consists of a 4-bit **CODE**, an 8-bit payload, and an 8-bit checksum:

```
[ CODE (4b) | resv (4b) ] [ Payload (8b)          ] [ Checksum ]
```

Segments labeled **resv** are to be considered reserved space. Currently there are four **CODE**s defined, each having a different purpose.

Each checksum is calculated such that the sum of all three bytes (truncated to one byte) equals **0xFF**; i.e., `checksum = 0xFF - ((byte1 + byte2) & 0xFF)`.

2.1 Control set (0x2)

This code sets the state of the control options available on the sensor unit. Respectively, these options are occupancy sensor trigger (**O**), temperature range trigger (**T**), manual relay control (**X**), and default relay state (**E**).

```
[ 0x2       | resv (4b) ] [ O T X E   | resv (4b) ] [ Checksum ]
```

If **O** is set, then the occupancy sensor should be used to trigger the relay. If **T** is set, then the temperature range should be used to trigger the relay. If **X** is set, then the relay should be manually controlled according to the state of the **E** bit. The **E** bit should be used as the default relay state on sensor unit reset regardless of whether **X** is set. These options are controlled by one bit each.

2.2 Temperature range set (0x3)

This code is used to set the high and low thresholds for the temperature-based relay control option.

```
[ 0x3       | resv (4b) ] [ HILO (1b) | TEMP (7b) ] [ Checksum ]
```

HILO controls whether the given **TEMP** is to be used as the high or low end of the range. A value of **0** indicates high, while **1** indicates low.

2.3 Time set or reset request (0x4)

This code can be used either to indicate that the sensor module should reset, or to set its internal clock.

```
[ 0x4       | resv (4b) ] [ FLAG (1b) | TIME (7b) ] [ Checksum ]
```

If **FLAG** is 0, then the sensor unit should reset. Upon reset, the unit should send a data frame with the **0x80** bit set (as specified in 3.2). This message will be replied to with another control message; the **FLAG** will be set and the **TIME** value will indicate the value from which the module should start counting.

2.4 Stream rate and on/off (0x5)

This code is used to set streaming on or off (off by default) and the stream rate.

```
[ 0x5       | resv (4b) ] [ FLAG (1b) | RATE (7b) ] [ Checksum ]
```

If **FLAG** is 0, then streaming is will be turned off. If 1, streaming will be turned on. The decimal value of **RATE** is the number of seconds to wait between receiving streamed packets (value of zero leaves the rate unchanged).

3 Data messages

Each data message logically consists of **16** bytes (may be longer on the wire – see 3.4 and 3.3) and corresponds to one second of measurements. The first **12** bytes are payload; the next **1** byte is reserved; the next **1** byte is the **ERROR** flag mask; the final **2** bytes are a 16-bit CRC. They are preceded by a start byte and may include escape bytes.

3.1 FLAGS field

The **FLAGS** field (byte 0) is a status mask with the following indicator bits defined:

0x01 Occupancy status

0x02 AC (relay) status

The remaining bits are reserved.

```
 0 [_FLAGS_______]
 1 [ VOLTAGE     ]
 2 [_____________]
 3 [ CURRENT     ]
 4 [_____________]
 5 [ PHASE       ]
 6 [_____________]
 7 [_TEMPERATURE_]
 8 [ TIMESTAMP   ]
 9 [             ]
10 [             ]
11 [_____________]
12 [_RESERVED____]
13 [_ERROR_______]
14 [ CRC-16      ]
15 [_____________]
```

3.2 ERROR field

The **ERROR** (byte 13) field is a status mask which provides indications of hardware failures. The bits are indicators defined as follows:

0x80 Reset complete / time not set

0x40 Relay failure

0x20 Bus failure

The remaining bits are reserved.

3.3 Start byte

Data messages are to be preceded by a start byte, **0xFF**, when sent on the wire. If the controller receives a start byte (unless it is escaped, per 3.4), it should discard the message it has buffered (if any) and start receiving a new message. The presence of a start byte protects against a total failure in the case that part of a message is lost in transmission.

3.4 Escape bytes

In order to use a start byte, we must also introduce an escape byte, **0xFE**. If **0xFE** precedes a byte, the escape byte should be ignored and the following byte should be interpreted as part of the message. Currently used to escape the start byte (**0xFF**) and the escape byte itself (**0xFE**), should those bytes be used in the raw message body, though it can be expanded to other special bytes.