Temperature Guard

FULL SDK

M307 INTEGRATION GUIDE

Contents

Overview

Communicating with a M307 is accomplished by reading and writing 60 byte records over a TCP/IP connection to port 10001. The first four bytes of each record is the command and the remaining 56 bytes is the data. There are a total of four user records and one current status record. The user records can be modified by reading, modifying, writing the record. The status record is read only.

Network parameters are programmed using the IP Setup Program, using the Web based administration Applet found at http://{IP_ADDRESS_OF_M307}/secure/ltx.conf.html, or telnetting in to the device at port 9999.
Additional security settings can be configured by referring to the document ANSECURE.

Section 1.0 - Reading the Current Status Record

1. Open a TCP/IP connection to the unit at port 10001.
2. Send a 60 byte record with the first four bytes set to 0x3f 0xcd 0xdc 0x00 and the remaining bytes to 0x00.
3. The Temperature Guard will immediately respond with the 60 byte Current Status Record. The first four bytes of the record will be the command and the remaining bytes will be the data which can be seen in table 1.

Status Record (Table 1)

Byte #	**Definition**
1	0x3F (command byte 1)
2	0xCD (command byte 2)
3	0xDC (command byte 3)
4	0x00 (command byte 4)
5	Temperature Sensor 1 Reading MSB
6	Temperature Sensor 1 Reading LSB
7	Temperature Sensor 1 Time out of limits MSB (minutes)
8	Temperature Sensor 1 Time out of limits LSB
9	Temperature Sensor 1 Out of limits state (1 = out of limit, 0 = in limits)
10	Temperature Sensor 2 Reading MSB
11	Temperature Sensor 2 Reading LSB
12	Temperature Sensor 2 Time out of limits MSB (minutes)
13	Temperature Sensor 2 Time out of limits LSB (minutes)
14	Temperature Sensor 2 Out of limits state (1 = out of limit, 0 = in limits)
15	Internal Temperature Sensor Reading MSB
16	Internal Temperature Sensor Reading LSB
17	Internal Temperature Sensor Time out of limits MSB (minutes)
18	Internal Temperature Sensor Time out of limits LSB
19	Internal Temperature Sensor Out of limits state (1 = out of limit, 0 = in limits)
20	Internal Humidity Sensor Reading MSB
21	Internal Humidity Sensor Reading LSB
22	Internal Humidity Sensor Time out of limits MSB (minutes)
23	Internal Humidity Sensor Time out of limits LSB
24	Internal Humidity Sensor Out of limits state (1 = out of limit, 0 = in limits)
25	Door 1 State MSB (not used)
26	Door 1 State LSB (1 = closed, 0 = open)
27	Door 1 Time out of limits MSB (minutes)
28	Door 1 Time out of limits LSB (minutes)
29	Door 1 Out of limits state (0 = not out of limit, 1 = alarm)
30	Door 2 State MSB (not used)
31	Door 2 State LSB (1 = closed, 0 = open) (not used)
32	Door 2 Time out of limits MSB (minutes)
33	Door 2 Time out of limits LSB (minutes)
34	Door 2 Out of limits state (0 = not out of limit, 1 = alarm)
35	Main Power (4 = on, 0 = off)
36	Battery Voltage Highbyte (returned times 100 i.e. 241 = 2.41v)
37	Battery Voltage LowByte
38-39	No Data
59	Specifies temperature measurement resolution (10 = .1°, any other number 1° resolution)
60	Degrees °C or °F (0x43 = C, 0x46 = F)

Section 1.1 - Interpreting Temperature Sensor Data
The temperature is returned as a two byte signed integer. (See conversion examples below).
Bytes 5,6 are sensor 1 readings and bytes 10,11 are sensor 2's readings, and bytes 15, 16 are the internal temperature sensor readings.

1000 is returned if no sensor is connected.
999 is returned if the sensor has open circuited.
-999 is returned if the sensor has shorted.

Byte 59 specifies the temperature measurement resolution. M307 with firmware revision 5 and above hardware has .1° resolution. All other models have 1° resolution.

10 is returned if the temperature measurement resolution is .1°, any other value is 1° resolution.

Byte 60 is the temperature unit of measurement (either "F" or "C")

Bytes 7,8 and 12,13 are the length of time the temperature has been out of limits. The time is returned as a two byte unsigned integer. Time is in minutes.

Bytes 9 and 14 indicate whether the temperature is in alarm condition. The byte is set to 1 when the temperature has been out of limits for a period of time greater than the programmed limit, else the byte is cleared.

Section 1.2 - Interpreting Internal Humidity Sensor Data
The humidity is returned as a two byte signed integer with .1 degree. (See conversion examples below).

Bytes 21,21 are the humidity sensors readings multiplied by 10 with a measurement of .1%RH.

999 is returned if the sensor failed to return data.

Bytes 22, 23 are the length of time the humidity has been out of limits. The time is returned as a two byte unsigned integer. Time is in minutes.

Byte 24 indicates whether the humidity is in alarm condition. The byte is set to 1 when the humidity has been out of limits for a period of time greater than the programmed limit, else the byte is cleared.

Section 1.3 - Interpreting Door Sensor Data
Bytes 25 and 30 indicate the state of the door sensor. The byte is set to 1 when the sensor is closed, and set to 0 when the door is open.

Bytes 27, 28 and 32, 33 are the length of time the door has been open and is returned as a two byte integer. Time is in minutes.

Bytes 29 and 34 indicate whether the door is in alarm condition. The byte is set to 1 when the door has been open for a period of time greater than the programmed limit, else the byte is cleared.

Section 1.3 - *Converting Data Examples*
All data is read and written as a two byte signed integer. The two bytes must be converted to decimal by shifting the MSB left 8 places and adding it to the LSB. The most significant bit of the result must then be tested for a negative value. If the bit is

set, the result is negative. If byte 59 in the Status Record is 10, temperature data must then be divided by 10 to yield the correct value.

Converting two byte unsigned data to a 16 bit signed integer

Visual Basic Example

```
Result = (MSB * 0x100) + LSB          'Shift MSB and add to LSB.
                                      'Result is hex.  Convert result to decimal.
  If (MSB >= 0x80){                   'Check for negative number by checking most significant bit
    Result = Result – 0x10000         'Subtract out offset
    Result = Result * -1              'Multiply by -1 to make the result negative
  }
```

Java Example

```
public static final short twoBytesToInt(byte highbyte, byte lowbyte) {
      short i = 0;

      i |= highbyte & 0xFF;
      i <<= 8;
      i |= lowbyte & 0xFF;
      return i;
}
```

Section 2.0 - Read, Modify, Write User Records

There are four records that the user can read, modify and write. The records are used to configure the M307. The read and write command for each record is shown in the table below.

User Record Number	Read command				Write Command			
	Byte 1	Byte 2	Byte 3	Byte 4	Byte 1	Byte 2	Byte 3	Byte 4
0	0xaa	0xbb	0xcc	0x00	0xdd	0xcc	0xbb	0x00
1	0xaa	0xbb	0xcc	0x01	0xdd	0xcc	0xbb	0x01
2	0xaa	0xbb	0xcc	0x02	0xdd	0xcc	0xbb	0x02
3	0xaa	0xbb	0xcc	0x03	0xdd	0xcc	0xbb	0x03
4	0xaa	0xbb	0xcc	0X04	0xdd	0xcc	0xbb	0x04
5	0xaa	0xbb	0xcc	0X05	0xdd	0xcc	0xbb	0x05

Section 2.1 - Reading a user record

1. Open a TCP/IP connection to the unit at port 10001.
2. Send a 60 byte record with the first four bytes set to the Read command for the desired record and the remaining bytes to 0x00.
3. The Temperature Guard will respond with a 60 byte record. The first four bytes of the record will be the command you sent and the remaining bytes will be the data contained in the User Record. See User Record Charts 0-3 for a byte by byte listing.

Section 2.2 - Modifying and writing a user record

- Read the record as described above.
- Modify the desired parameter(s) and insert into the record.
- Set the command to the Write command for the particular record.
- The M307 will store the record in NVM, read the parameters back from NVM and return the record. The returned record will contain the Send Command (0xaa, 0xbb, 0xcc, {record number}.)
- Compare the sent and returned records to verify that the operation was successful.

Section 2.3 - Converting data into two byte hex (to insert into records)

Visual Basic Example

```
LowByte = DataValue && 0xff
HighByte = (DataValue && 0xff00) / 0x100
HighByte = HighByte && 0xff
```

Section 2.4 – Record Tables

User Record 0: Temperature and door sensor limits

User Record 0		Size (bytes)	Byte #
Command	Command byte 1	1	1
	Command byte 2	1	2
	Command byte 3	1	3
	Command byte 4	1	4
Password	Not implemented	4	5
Temperature Sensor 1 Data	Temperature Sensor 1 Temperature Lower Limit MSB	1	9
	Temperature Sensor 1 Temperature Lower Limit LSB	1	10
	Temperature Sensor 1 Temperature Upper Limit MSB	1	11
	Temperature Sensor 1 Temperature Upper Limit LSB	1	12
	Temperature Sensor 1 Time Delay MSB	1	13
	Temperature Sensor 1 Time Delay LSB	1	14
Temperature Sensor 2 Data	Temperature Sensor 2 Temperature Lower Limit MSB	1	15
	Temperature Sensor 2 Temperature Lower Limit LSB	1	16
	Temperature Sensor 2 Temperature Upper Limit MSB	1	17
	Temperature Sensor 2 Temperature Upper Limit LSB	1	18
	Temperature Sensor 2 Time Delay MSB	1	19
	Temperature Sensor 2 Time Delay LSB	1	20
Internal Temperature Sensor Data	Internal Temperature Sensor Temperature Lower Limit MSB	1	21
	Internal Temperature Sensor Temperature Lower Limit LSB	1	22
	Internal Temperature Sensor Temperature Upper Limit MSB	1	23
	Internal Temperature Sensor Temperature Upper Limit LSB	1	24
	Internal Temperature Sensor Time Delay MSB	1	25
	Internal Temperature Sensor Time Delay LSB	1	26
Internal Humidity Sensor Data	Humidity Lower Limit MSB	1	27
	Humidity Lower Limit LSB	1	28
	Humidity Upper Limit MSB	1	29
	Humidity Upper Limit LSB	1	30
	Humidity Time Delay MSB	1	31
	Humidity 2 Time Delay LSB	1	32
Door 1 Data	Door 1 Time Delay MSB	1	33
	Door 1 Time Delay LSB	1	34
Door 2 Data	Door 2 Time Delay MSB	1	35
	Door 2 Time Delay LSB	1	36
Temperature Sensor 1 CF	Correction Factor MSB (M307 rev5 and above)	1	37
	Correction Factor LSB	1	38
Temperature Sensor 2 CF	Correction Factor MSB (M307 rev5 and above)	1	39
	Correction Factor LSB	1	40
Internal Temp Sensor CF	Correction Factor MSB (M307 rev5 and above)	1	41
	Correction Factor LSB	1	42
Internal Humidity Sensor CF	Correction Factor MSB (M307 rev5 and above)	1	43
	Correction Factor LSB	1	44
Not used		1	45
Input 1 Logic Level	0 = normally open, 1 = normally closed, 2 = water	1	46
Input 2 Logic Level	0 = normally open, 1 = normally closed, 2 = water	1	47

User Record 1: Unit Identification Parameters

User Record 1		Size (bytes)	Byte #
Command Byte 1	command byte 1	1	1
Command Byte 2	command byte 2	1	2
Command Byte 3	command byte 3	1	3
Command Byte 4	command byte 4	1	4
Password	Not Implemented	4	5-8
Device name	20 ascii characters	20	9-28
Unit of measure °C or °F	1 ascii character (0x43 = C, 0x46 = F)	1	29
Device MAC	20 ascii characters (informational only)	20	30-49
Serial Number	20 ascii characters (informational)	10	51-60

User Record 2: Temperature Sensor Names

User Record 2		Size (bytes)	Byte #
Command Byte 1	Command byte 1	1	1
Command Byte 2	Command byte 2	1	2
Command Byte 3	Command byte 3	1	3
Command Byte 4	command byte 4	1	4
Password	Not Implemented	4	5-8
Sensor 1 Name	20 ascii characters (must name to enable alarm)	20	9-28
Sensor 2 Name	20 ascii characters (must name to enable alarm)	20	29-48

User Record 3: Door Input Names

User Record 3		Size (bytes)	Byte #
Command Byte 1	command byte 1	1	1
Command Byte 2	command byte 2	1	2
Command Byte 3	command byte 3	1	3
Command Byte 4	command byte 4	1	4
Password	Not Implemented	4	5-8
Door Sensor 1 Name	20 ascii characters (must name to enable alarm)	20	9-28
Door Sensor 2 Name	20 ascii characters (must name to enable alarm)	20	29-48

User Record 4		Size (bytes)	Byte #
Command Byte 1	command byte 1	1	1
Command Byte 2	command byte 2	1	2
Command Byte 3	command byte 3	1	3
Command Byte 4	command byte 4	1	4
Password	Not Implemented	4	5-8
Relay Logic	Hex (0: relay normally off, 1: relay normally on)	1	9
Alarm Reminder Time Delay	Hex (0 to disable, valid range 1 – 255)	1	10
Enable Buzzer	Hex (0 to disable, 1 to enable)	1	11
Two stage door alarm time delay	Hex (0 to disable, valid range 1 – 255)	1	12

User Record 5: Internal Temperature and Humidity Sensor Names

User Record 3		Size (bytes)	Byte #
Command Byte 1	command byte 1	1	1
Command Byte 2	command byte 2	1	2
Command Byte 3	command byte 3	1	3
Command Byte 4	command byte 4	1	4
Password	Not Implemented	4	5-8
Internal Temperature Sensor Name	20 ascii characters (must name to enable alarm)	20	9-28
Internal Humidity sensor Name	20 ascii characters (must name to enable alarm)	20	29-48

Section 2.5 - Temperature Sensor Limits
Temperature sensor limits consist of a lower temperature limit, an upper temperature limit, a time delay and the name of the sensor. When the temperature has been outside a limit for a length of time greater than the programmed time delay, the M307 will go into alarm.

The name of the sensor, the two temperature limits and the time delay must be written to enable the sensor to cause an alarm condition. The temperature can always be read in the status page even if the sensor is not named or the limits not set, but an “unnamed” sensor will not cause an alarm.

Write a 0x43 (‘C’) into user record 1, byte number 29 to have the M307 transmit the temperature in °C, or write a 0x4f (‘F’) into user record 1, byte number 29 to have the M307 transmit the temperature in °F.

Section 2.6 - Door Sensor Limits
The door sensor limits consist of the name of the sensor and a time delay. When a door has been open for a period of time greater to the time delay, the M307 will go into alarm.

The name of the sensor and the time delay must be written to enable the sensor cause an alarm condition.

The door sensor can always be read in the status page even if the door sensor is not named or the limit not set.

Note: There is no error checking done by the M307. It is up to the programmer to verify that the parameters written to the M307 are valid.

Section 2.7 – External Temperature Field Calibration See User Record 0 for details

The M307 can be field calibrated at a single point via user record 0 (see Section 2.4 User Record 0 above). The correction factor is calculated as:

CF = Temperature measured by a calibrated thermometer – Temperature measured by M307
CF = CF * 10 'multiply result by 10

Convert data into two hex bytes
Insert in User Record 0
Send to M307

For example:

The calibrated thermometer indicated the temperature to be 32.5° and the M307 displays 32.7°
CF = 32.5 – 32.7
CF = -.2 * 10 = -2

Convert -2 to hex. MSB = 0xff, LSB = 0xfe

When calculating the temperature, the M307 will add the correction factor to the final result.
Temperature = Calculated value + correction factor (note this value is transmitted multiplied by 10).

Note: The internal temperature and humidity is factory calibrated and requires no further calibration.

Section 3.0 - On Board Data Logging Summary

The M307 maintains a log of temperature, humidity, status of inputs (open/closed), device power on/off and a date/time stamp. The log is 4000 records, fifteen (15) bytes long and is stored as a circular buffer in NVM memory.

Data logging will continue until the internal battery backup is discharged.

Section 3.1 – On Board Data Logging Command Set

Command Summary				
	Byte 1	**Byte 2**	**Byte 3**	**Byte 4**
Store Date/Time and Update Rate	0xde	0xca	0xde	0x00
Read Date/Time and Update Rate	0xde	0xca	0xde	0x02
Read Log File	0xde	0xca	0xde	0x04

When the M307 is done transmitting the log file it transmits "THE-END"

Store Date/Time and Update Rate		**Data Type**	**Size (bytes)**	**Byte #**
Command Byte 1	0xde	HEX	1	1
Command Byte 2	0xca	HEX	1	2
Command Byte 3	0xde	HEX	1	3
Command Byte 4	0x00	HEX	1	4
Seconds Not used set to 0	0	HEX	1	5
Minutes	00-59	BCD	1	6
Hour (00-24)	00-24	BCD	1	7
Leave Blank	0	HEX	1	8
Date	1-31	BCD	1	9
Month	1-12	BCD	1	10
Year	2000-2099	BCD	1	11
Log Update Rate (minutes)	1-60	HEX	1	12

Read Date/Time, Update Rate, Number of Log Records		**Data Type**	**Size (bytes)**	**Byte #**
Command Byte 1	0xde	HEX	1	1
Command Byte 2	0xca	HEX	1	2
Command Byte 3	0xde	HEX	1	3
Command Byte 4	0x02	HEX	1	4
Seconds	00-59	HEX	1	5
Minutes	00-59	BCD	1	6
Hour (00-24) (See description below)	00-24	BCD	1	7
Day	1-7	BCD	1	8
Date	1-31	BCD	1	9
Month	1-12	BCD	1	10
Year	2000-2099	BCD	1	11
Log Update Rate (minutes)	1-255	HEX	1	12
Total Size of Log Table Highbyte	MSB	HEX	1	13
Total Size of Log Table Lowbyte	LSB	HEX	1	14

Read Log File Command		Data Type	Size (bytes)	Byte #
Command Byte 1	0xde	HEX	1	1
Command Byte 2	0xca	HEX	1	2
Command Byte 3	0xde	HEX	1	3
Command Byte 4	0x04	HEX	1	4
Reset Log Pointer	(0x01 = reset)	HEX	1	5

Read Log File Response		Data Type	Size (bytes)	Byte #
Minute	00-59	BCD	1	1
Hour	00-24	BCD	1	2
Day	1-7	BCD	1	3
Date	1-31	BCD	1	4
Month	1-12	BCD	1	5
Year	00-99	BCD	1	6
Temp 1 Highbyte	MSB	HEX	1	7
Temp 1 Lowbyte	LSB	HEX	1	8
Temp 2 Highbyte	MSB	HEX	1	9
Temp 2 Lowbyte	LSB	HEX	1	10
Internal Temp 1 Highbyte	MSB	HEX	1	11
Internal Temp 1 Lowbyte	LSB	HEX	1	12
Internal Humidity Highbyte	MSB	HEX	1	13
Internal Humidity Lowbyte	LSB	HEX	1	14
Input and Power Status	see sec 3.2	HEX	1	15
~~sequence repeated until last record	~~	~~	~~	~~
Minute	00-59	BCD	1	32745
Hour	00-24	BCD	1	32746
Day	1-7	BCD	1	32747
Date	1-31	BCD	1	32748
Month	1-12	BCD	1	32749
Year	00-99	BCD	1	32750
Temp 1 Highbyte	MSB	HEX	1	32751
Temp 1 Lowbyte	LSB	HEX	1	32752
Temp 2 Highbyte	MSB	HEX	1	32753
Temp 2 Lowbyte	LSB	HEX	1	32754
Internal Temp 1 Highbyte	MSB	HEX	1	32755
Internal Temp 1 Lowbyte	LSB	HEX	1	32756
Internal Humidity Highbyte	MSB	HEX	1	32757
Internal Humidity Lowbyte	LSB	HEX	1	32758
Input and Power StatusDoor Status	see sec 3.2	HEX	1	32759

Section 3.2 – On Board Data Logging Record Detail
Each record contains a date and time stamp including minute, hour, date, month and year of the data values.

The temperature and humidity readings are represented as a two byte signed integer as in Current Status Record (see section 1.0). Conversion to 16 bit signed integer examples are presented in section 1.3.

The Door Status byte contains door input values and the power status in bit form. 1 = active and 0 = inactive.

Door Status Word	Bit
Input 1 Status	0
Input 2 Status	1
Power Status	2
	3
	4
	5
	6
	7

Hour
The hour byte in the log record is packed with additional data as shown below.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	12 /24	/AM PM	10 Hour	Hour			

Bit 6 runs in 12 hour mode so it is always 1
To obtain the hour, mask off bits 5, 6, 7 and determine AM/PM from bit 5.

Section 4.0 – Sample Code
All code presented in this SDK is presented without warranty.

Section 4.1 – Parsing Current Status Record
Java Example **-** Pass the 60 byte record returned by the M307 into ParseData(). Values are stored in the global array "Reading".

```
public byte[] Unit = new byte[1];
public double Divisor = 1.0;
public int InputsToRead = 4;    // M307, M306304 have 4 readings (2 temp, 2 door)
public double[] Reading = new double[InputsToRead];
public int[] OutOfLimitsTime = new int[InputsToRead];
public int[] OutOfLimitsStatus = new int[InputsToRead];

// function twoBytesToInt() listed in section 1.3

public void ParseData(byte[] Data) {
int Input;

 Input = 0;
 if (Data[58] == 10)
        Divisor = 10.0;
 else
        Divisor = 1.0;
 do {
        Reading[Input] = twoBytesToInt( Data[( (5*Input) + 4 )], Data[( (5*Input) + 5 )] );
        Reading[Input] = Reading[Input] / Divisor;
        OutOfLimitsTime[Input] = twoBytesToInt( Data[( (5*Input) + 6 )], Data[( (5*Input) + 7 )] );
        OutOfLimitsStatus[Input] = Data[( (5*Input) + 8 )];
        Input++;
 } while(Input < InputsToRead);
 Unit[0] = Data[59];
```



```
}
```

