Level Plus®
Magnetostrictive Liquid Level Transmitters
with Temposonics® Technology

DDA Interface Manual
LP Series
Table of contents

1. Contact information ................................................. 3
2. Terms and definitions ............................................. 4
3. Introduction ........................................................... 6
4. Safety instructions .................................................. 6
5. Quick start-up guide ............................................... 6
   5.1 Before you begin ............................................... 6
   5.2 Quick start-up procedure ................................. 6
6. DDA Interface ........................................................ 6
7. Hardware and software environments ....................... 6
8. DDA command decoder examples .............................. 6
9. DDA/Host computer communication protocol ............. 6
   9.1 Network protocol/timing considerations .......... 6
   9.2 Special control commands ............................... 6
   9.3 Level commands ............................................. 8
   9.4 Temperature commands ................................... 11
   9.5 Multiple output commands ............................. 11
   9.6 High-level memory read commands .................. 11
   9.7 Diagnostic/Special command set ....................... 11
   9.8 DDA Error codes ............................................. 11
11. Digital setup software installation, setup and calibration 15
   11.1 Data from Device tab ..................................... 15
   11.2 Calibration .................................................. 17
   11.2 Change address .......................................... 17
   11.2 Backup and restore device settings .............. 17
   11.2 Com port .................................................. 17
   11.2 Continuous update ...................................... 17
   11.2 Data logging .............................................. 17
1. Contact information

United States

General
Tel: +1-919-677-0100
Fax: +1-919-677-2343
E-mail: info.us@mtssensors.com
http://www.mtssensors.com

Mailing and shipping address
MTS Systems Corporation
Sensors Division
3001 Sheldon Drive
Cary, North Carolina, 27513, USA

Customer service
Tel: +1-800-633-7609
Fax: +1-800-498-4442
E-mail: info.us@mtssensors.com

Technical support and applications
24 Hour Emergency Technical Support
Tel: +1-800-633-7609
e-mail: levelplus@mts.com

Japan

General
Tel.: +81-42-775-3838
Fax: +81-42-775-5516
E-mail: info.jp@mtssensors.com
http://www.mtssensors.com

Mailing and shipping address
MTS Sensors Technology Corporation
737 Aihara-cho, Machida-shi
Tokyo 194-0211, Japan

Technical support and applications
Tel.: +81-42-775-3838
Fax: +81-42-775-5512

Germany

General
Tel.: +49-2351-9587-0
Fax: +49-2351-56491
e-mail: info.de@mtssensors.com
http://www.mtssensors.com

Mailing and shipping address
MTS Sensor Technologie, GmbH & Co. KG
Auf dem Schüffel 9
D - 58513 Lüdenscheid, Germany

Technical support and applications
Tel.: +49-2351-9587-0
e-mail: info.de@mtssensors.com
http://www.mtssensors.com
2. Terms and definitions

6A Heavy Oils
‘Generalized Crude Oils’, Correction of Volume to 60 °F against API Gravity.

6B Light Oils
‘Generalized Products’, Correction of Volume to 60 °F against API Gravity.

6C Chemical
‘Volume Correction Factors (VCF)’ for individual and special applications, volume correction to 60 °F against thermal expansion coefficients.

6C Mod
An adjustable temperature reference for defining VCF.

A

API Gravity
The measure of how heavy or light a petroleum liquid is compared to water. Allowable values are 0 to 100 degrees API for (6A) and 0 to 85 degrees API for (6B).

D

DDA
‘Direct Digital Access’ – The proprietary digital protocol developed by MTS for use in intrinsically safe areas.

Density
Mass divided by the volume of an object at a specific temperature. The density value should be entered as lb / cu. ft..

E

Explosionproof
Type of protection based on enclosure in which the parts which can ignite an explosive gas atmosphere are placed within, and which can withstand the pressure developed during an internal explosion of an explosive mixture, and which prevents the transmission of the explosion to the explosive gas atmosphere surrounding the enclosure.

F

Flameproof
Type of protection based on enclosure in which the parts which can ignite an explosive gas atmosphere are placed within and which can withstand the pressure developed during an internal explosion of an explosive mixture, and which prevents the transmission of the explosion to the explosive gas atmosphere surrounding the enclosure.

FOUNDATION™ fieldbus
An all digital, serial, two-way communications system that serves as the base-level network in a plant or factory automation environment. Developed and administered by the fieldbus FOUNDATION™.

G

GOVI
‘Gross Observed Volume of the Interface’ – The total volume of the tank occupied by the interface liquid. The GOVI is only given when measuring two liquids and is calculated by subtracting the volume of the product from the total volume of liquid in the tank (GOVT – GOVP).

GOVP
‘Gross Observed Volume of the Product’ – The total volume of the tank occupied by the product liquid. When measuring only one liquid, it is also the total volume of liquid in the tank (GOVT). When measuring two liquids it is the total volume of liquid in the tank minus the volume of the interface liquid (GOVT – GOVI).

GOVT
‘Total Gross Observed Volume’ – The total volume of liquid in the tank. When measuring only one liquid it is equal to the volume of the product (GOVP). When measuring two liquids it is equal to the volume of the product and interface liquids (GOVP + GOVI).

GOVU
‘Gross Observed Volume Ullage’ – the difference in volume between the working capacity of a tank and the total volume in the tank (Working Capacity – GOVT).

H

HART®
A Bidirectional communication protocol that provides data access between intelligent field instruments and host systems.

I

Interface
Noun: The measurement of the level of one liquid when that liquid is below another liquid.

Interface
Adj.: The Software Graphical User Interface (GUI) that allows the user to access software protocols (HART, DDA, MODBUS).

Intrinsic safety
‘Intrinsically safe’ - Type of protection based on the restriction of electrical energy within apparatus of interconnecting wiring exposed to potentially explosive atmosphere to a level below that which can cause ignition by either sparking or heating effects.
**Mass**
The property of a body that causes it to have weight in a gravitational field, calculated by density at the reference temperature multiplied by the volume correction factor \((\text{Density} \times \text{VCF})\).

**MODBUS**
A serial communications protocol published by Modicon in 1979 for use with its programmable logic controllers (PLCs). It has become a de facto standard communications protocol in industry, and is now the most commonly available means of connecting industrial electronic devices.

**NEMA Type 4X**
A product Enclosure intended for indoor or outdoor use primarily to provide a degree of protection against corrosion, windblown dust and rain, splashing water, and hose-directed water; and to be undamaged by the formation of ice on the enclosure. They are not intended to provide protection against conditions such as internal condensation or internal icing.

**NPT**
U.S. standard defining tapered pipe threads used to join pipes and fittings.

**NSVP**
‘Net Standard Volume of the Product’ – The temperature corrected volume for the product liquid in the tank, requires the transmitter to be ordered with temperature measurement capabilities. The NSVP is calculated by multiplying the volume of the product liquid by a volume correction factor based on temperature \((\text{GOVP} \times \text{VCF})\).

**Reference Temperature**
The temperature at which the density measurement is given, the allowable values are 32 °F to 150 °F (10 °C to 66 °C).

**Specific Gravity**
The density ratio of a liquid to the density of water at the same conditions.

**Sphere Radius**
The internal radius of the sphere that contains the liquid, the value is used to calculate the volume along with the Sphere Offset.

**Sphere Offset**
An offset value that accounts for additional volume in a sphere from non-uniform sphere geometry, the value is used to calculate the volume along with the Sphere Radius.

**Strap Table**
A table of measurement correlating the height of a vessel to the volume that is contained at that height. The transmitter can contain up to 100 points.

**TEC**
‘Thermal Expansion Coefficient’ - a value correlating the change in temperature for an object with the change in its volume. Allowable values are 270.0 to 930.0. TEC units are in 10 E-6/Deg F.

**Temperature Correction Method**
One of five product correction methods used to correct the product volume in the tank due to changes in temperature from 60 °F including (6A, 6B, 6C, 6C Mod, and Custom Table).

**Volume Calculation Mode**
One of two methods use to calculate volume measurements from level measurements, including Sphere and Strap Table.

**VCF**
‘Volume Correction Factor’ – A table of measurements correlating temperature points with correction factors for the liquids expansion/contraction. The transmitter can contain up to 50 points.

**Working Capacity**
The maximum volume of liquid that the user desires for their vessel to hold, typically 80% of the vessel's maximum volume before overfill.
3. Introduction

3.1 Purpose and use of this manual

Before starting the operation of the equipment read this documentation thoroughly and follow the safety information.

The content of this technical documentation and of its various annexes is intended to provide information on mounting, installation and commissioning by qualified service personnel according to IEC 60079-14 and/or MTS trained service technicians and local regulations.

3.2 Used symbols and warnings

Warnings are intended for your personal safety and for avoidance of damage to the described product or connected devices. In this documentation, safety information and warnings to avoid dangers that might affect the life and health of personnel or cause material damage are highlighted by the preceding pictogram, which is defined below.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOTICE</td>
<td>This symbol is used to point to situations that may lead to material damage and/or personal injury.</td>
</tr>
</tbody>
</table>

4. Safety instructions

4.1 Intended use

The purpose of this document is to provide detailed information on the protocol interface. All safety related information is in the product specific operation manual. Consult the operation manual before connecting to the level transmitter.

5. Quick start-up guide

5.1 Before you begin

**Note:**

You must use a RS-485 converter with “Send Data Control” and the Set-up Software to ensure proper operation.

**Example:**


FTDI USB-RS485-WE-1800-BT (www.ftdichip.com)

5.2 Quick start-up procedure

1. Connect +24 Vdc to terminals.
2. Connect data lines to terminals.
3. Connect the PC (or other device) to data lines.
   (If you are using a PC, use a RS-232 to RS-485 converter. See Note above for more information.)
4. Turn on power to the transmitter.
5. Start the DDA Setup Software. Click the ‘Data From Device’ tab. Click the ‘Device’ pull down menu (located in the upper right corner of the window) to verify communications using factory default address “192 ” for DDA.

6. DDA Interface

6.1 Data line termination and biasing

Termination and biasing of RS-485 data lines are as follows:

**Biasing**

Each DDA transmitter has internal high impedance biasing resistors (30K Ω) on both RS-485 data lines. No additional biasing resistors should be present on the connecting devices (PLC, DCS, PC, Converter).

**Termination**

Each DDA transmitter has an internal termination resistor (100K Ω) installed across the RS-485 signal lines. No additional termination resistors are necessary in the connecting devices (PLC, DCS, PC, Converter).

6.2 Communication parameters

The 2-wire differential communication interface and all data transmissions must be at half duplex. Only one device (either the master or a single transmitter) can transmit data at any given time. BAUD rate limitations are listed below.

**Modbus:** 4800 BAUD or 9600 8, N, 1
**(Reference) Monitor:** Modbus RTU Variable BAUD Rate 8, E, 1
7. Hardware and software environments

The Level Plus Model MG digital transmitter operates in a networked, intrinsically safe RS-485 DDA software environment. This environment supports up to 8 multi-dropped transmitters on one communication line. The network requires a 4-wire bus to provide both power and communications to each of the transmitters located in the hazardous area. The transmitters are connected in multi-point configuration (see Figure 25).

The RS-485 network operates in a master/slave mode where the master (host computer or similar type network controller) interrogates each slave (DDA transmitter) for a specific type of data. Each slave has a unique switch programmable hardware address that is issued by the host computer to activate a particular transmitter. In addition, the DDA hardware supports a command decoder that supports up to 128 different commands. The host computer interrogates a transmitter for data by sending an address byte, followed immediately by a command byte. The addressed transmitter will ‘wake up’, identify itself by transmitting an echo of its own local address followed by the received command, and then perform the requested action. After the requested action has been completed, the data (if any) will be transmitted back to the host computer on the RS-485 network. Refer to Section ‘DDA Command decoder examples’ on page 43 for more information.

---

8. DDA command decoder examples

8.1 Serial data transmission format

Example 1:

```
0 X X X X X X X P 1
```

<table>
<thead>
<tr>
<th>D1 bit</th>
<th>Stop bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start bit</td>
<td>Parity bit</td>
</tr>
<tr>
<td>D8 bit</td>
<td></td>
</tr>
</tbody>
</table>

is found, the word is ignored and the decoder circuitry resets for the next transmission. If the parity check is good, the decoder circuitry checks for a valid address byte. The address decoder circuitry uses the ‘D8’ bit to distinguish the difference between address bytes and command bytes. Address bytes are defined as having the most significant bit ‘D8’ set equal to one. Valid address byte values include ‘C0’ hex to ‘FD’ hex (192 to 253 decimal). Address byte values from 80 hex to ‘BF’ hex are reserved for future use; address byte values ‘FE’ and ‘FF’ hex are reserved for test functions. (see Example 2).

After the DDA address decoder circuitry receives the 11-bit word, an even parity check is performed across the 8-bit data field. If a parity error
8.2 Address byte

Example 2:

X X X X X X X 1

D8 bit = 1

(8-bit word - shown as D1 bit)

If the received address byte matches the local DDA address, the DDA power supply circuitry is activated. If a valid address byte has been found, the decoder circuitry checks to see if the next received word is a command byte. Valid command byte values include ‘00’ hex to ‘7F’ hex (0 to 127 decimal). In addition, all data byte values are restricted to be within ‘00’ hex and ‘7F’ hex (see Example 3).

8.3 Command byte (and data bytes)

Example 2:

X X X X X X X 0

D8 bit = 1

(8-bit word - shown as D1 bit)

Again, an even parity check is performed on the command byte. If the parity check is good, the eight bit data word is latched into a command buffer. This buffer is read by the DDA software to determine which command to execute. If the parity check fails, the command byte is rejected and the old command (from the previous interrogation sequence) will be left in the command buffer. The DDA hardware cannot determine if the current command was possibly rejected. The host computer must then verify if the correct command was received by reading the echo of the address byte and command byte sent by the DDA transmitter. This is the only guaranteed way to determine that both the address and command bytes were received properly.

This method also insures proper verification, even if the parity check fails to detect a multiple bit data error in either the address byte or command byte. If the host computer determines that either the address byte or command byte has been corrupted, it must wait the proper time-out period and ignore the received message from the DDA transmitter that was improperly interrogated. The time-out period is variable and is based on the duration of the selected DDA.

9. DDA/Host computer communication protocol

The DDA/Host computer communication protocol consists of two parts: the interrogation sequence generated by the host computer and the data response generated by the interrogated DDA transmitter. The host interrogation sequence always consists of an address byte followed immediately by a command byte (see Example 4).

9.1 DDA/Host communication

Example 4:

<address byte><command byte>

00 Hex to 7F Hex (0 to 127 decimal)

C0 Hex to FD Hex (192 to 253 decimal)

The maximum delay between the address byte and the command byte is 5 milliseconds. The DDA transmitter will not receive the new command byte if this delay period is exceeded (and the old command byte will be left in the command buffer). See previous section for additional information on verification of the Address/Command bytes. An example of an interrogation sequence to access a transmitter programmed for address ‘F0’ hex (see Example 5).

The transmitter response consists of several components. After a transmitter has been interrogated, the transmitter first responds by transmitting its own local address and the command that was received from the host computer. This re-transmission of the transmitter address and received command serves two purposes. The first being a simple identification that the correct transmitter received the correct command and that it is currently active. The second purpose is to reset the DDA Address/Command decoder circuitry for the next interrogation sequence.

9.2 Interrogation data sequence

Example 5:

<F0><0A>

Command 0A Hex (10 decimal)

Address F0 Hex (240 decimal)

Note:

If the DDA transmitter does not respond to the first interrogation by the host, the Address/Command decoder will be left in an intermediate state. If this occurs, the host will have to reinterrogate the respective transmitter to reset the Address/Command decoder circuitry and then interrogate the respective transmitter again to perform a new transmitter measurement. This hardware feature must be considered when writing software communication drivers to access DDA transmitter data.
After the DDA transmitter has retransmitted its local address and received command, it will perform the requested measurement as defined by the received command. After the requested measurement has been completed, the data for that measurement will be transmitted to the host in a predefined format including certain control characters. The DDA transmitted data format begins with a ‘start of text’ ‘STX’ character (STX = 02 hex). The ‘STX’ character set is immediately followed by the requested data and then terminated with an ‘end of text’ ‘ETX’ character set (ETX = 03 hex). Certain commands allow multiple data fields to be transmitted within one transmitted data sequence. For these data transmissions, each data field is separated by an ASCII colon ‘:’ character (: = 3A hex), (see Examples 6 and 7).

**Single field data transmission**

Example 6:

```
<STX><dddd.ddd><ETX>
```

**Multiple field data transmission**

Example 7:

```
<STX><dddd.ddd:dddd.ddd:dddd.ddd><ETX>
```

All transmitted data will consist of 7-bit ASCII characters limited to hex values between ‘00’ hex and ‘7F’ hex (i.e. data bit D8 = 0).

After a DDA transmitter has completed a data transmission, the host must wait 50 milliseconds before another interrogation can be performed. This delay is required to enable the previously interrogated transmitter to go into sleep mode and release the network communication lines.

All DDA control commands support a checksum calculation function, Data Error Detection (DED) that allows the host computer (master) to check the integrity of the transmitted data. The actual checksum value that is transmitted is the compliment (2’s compliment) of the calculated value. The checksum scheme is based on a 16-bit summation of the hex data within the transmitted block (including ‘STX’ and ‘ETX’ character sets) without regard to overflow. The two byte result of the adding process is then complimented and appended to the transmitted data block.

This compliment process makes the final checksum comparison more efficient in that the checksum result added to its compliment will always result in a zero sum for uncorrupted data transmissions. Checksum data (two hex bytes) can range from ‘0000’ hex to ‘FFFF’ hex. Since the communication network only allows transmitted data values between 00 and 7F hex, special processing is required on the hex checksum value before it can be transmitted.

This two byte hex value must first be converted to numeric (decimal) ASCII characters before transmission. For example, a checksum value of ‘FFFF’ hex would be transmitted as ASCII 65535. The host computer would then have to convert ASCII 65535 back to FFFF hex and perform its own checksum calculation and comparison for the received data from the DDA transmitter. An example is shown (see Example 8) of a single field data transmission including checksum data and an sample checksum calculation.

**Example 8:**

```
<STX><dddd.ddd><ETX><cccccc>
```

**Append checksum value**

**Note:**

The appended checksum value will always consist of five decimal (ASCII) characters ranging from 00000 to 65535. The checksum function can be enabled or disabled.

**Message transmitted from DDA transmitter (command 12 Hex):**

```
<STX><265.322.109.456><ETX><64760>
```

**Hex character equivalent of transmitted data record including <STX> and <ETX> characters:**

- 02, 32, 36, 35, 2E, 33, 32, 3A, 31, 30, 39, 2E, 34, 35, 36, 03

**Two byte Hex summation of data:**

0308 Hex

**Two’s compliment:**

FCF8 Hex

**Convert to decimal ASCII:**

64760

To verify transmitted data from the DDA transmitter, perform the two byte Hex summation over the data record (including ‘<STX>’ and ‘<ETX>’ ) shown in (Example 8). The result in this example is 0308 Hex. Then convert the decimal ASCII checksum value back to Hex (for example, 64760 to FCF8 Hex). Add the Hex summation value to the Hex checksum value and the result will be zero (disregarding overflow) for uncorrupted data. 0308 Hex + FCF8 Hex = 0000 Hex.

**Note:**

Cyclic Redundancy Check (CRC) error checking will be offered at a later date. A command switch will be defined that will let the DDA data be transmitted with CRC error checking instead of checksum error checking. The checksum calculations will use the CRC-CCITT defined polynomial with a 16-bit CRC result. This 16-bit CRC value will be appended to each transmitted message. Since the communication network only allows transmitted data values between 00 and 7F hex, special processing is required on the 16-bit hex CRC value before it can be transmitted. This 16-bit (two byte) hex value must first be converted to numeric (decimal) ASCII characters before transmission. For example, a checksum value of ‘FFFF’ hex would be transmitted as ASCII 65535.
9.3 Network protocol/timing considerations

The DDA network has several timing constraints that must be considered when designing and coding communication drivers. The DDA network follows the RS-485 standard which defines a multi-drop communication interface that uses differential drivers and receivers operating in half-duplex mode. When using the RS-485 standard configuration, each device’s driver and receiver are wired together (see Figure 26).

Each device drive on the network must be disabled (high impedance) except when the device is ready to transmit data. In order to keep devices from transmitting data at the same time, one device is selected as the host (or master). In a DDA network, the host computer (or other communication interface) is the master, and controls the communication timing and protocol. The DDA transmitters act as slave devices, only transmitting data when requested by the host computer device. In this case, the host computer enables its driver and transmits the ‘Address/Command’ interrogation sequence.

After the Address/Command has been completely transmitted, the host disables its driver to allow reception of the data from the DDA transmitter. The transmitter with the matching address then becomes active, enables its driver and transmits the Address/Command echo followed by the requested data. The transmitter then disables its driver and goes back into sleep mode. Since all devices operate independently, certain timing constraints are imposed on the protocol to eliminate multiple devices from transmitting data simultaneously.

Network protocol timing sequences (interrogation sequences) are shown in (Figure 27. This time line representation of data transmission sequences also provides information about host computer control of the RS-485 communication card and also illustrates driver enabled control through the RTS control line.

**Note:**

Many available communication cards (line drivers) for use with the host computer device use a special control line input to control the enabling and disabling of the RS-485 driver. Typically this input is connected to the computers RTS or DTR communication port control line. The computer can then control the state of the driver by toggling the RTS or DTR signal lines via software control. An example of this control method is shown in (Figure 27). Other control methods are also used depending on the manufacturer of the equipment.

**The following steps provide an interrogation sequence example:**

1. The start of the sequence begins when the host enables its RS-485 driver to transmit the Address/Command bytes (see time line ‘T0’ in Figure 27).

2. After the driver is enabled, the host performs a small time delay ‘T1’. In this example, the host enables the driver by raising the RTS control line of the computer to the active (enabled) state. This typically requires no more than 1 millisecond. If the communication lines are extremely long, additional time may be required due to the additional capacitance of the wires.

3. The host then transmits the address byte followed immediately by the command byte. For 4800 Baud transmission rates, the time to transmit one byte (11-bit word size) is fixed at 2.3 milliseconds. Then time delays ‘T2’ and ‘T4’ are fixed at 2.3 milliseconds. Time delay ‘T3’ is the interbyte transmission time. Normally this is at least one bit time (0.21 milliseconds @ 4800 Baud) which is controlled by the computer communication hardware. Sometimes software overhead can extend this delay. The maximum permissible delay for period ‘T3’ is 5 milliseconds. Then the total maximum delay for periods ‘T2, T3, T4’ is 9.6 milliseconds.

4. After the host transmits the address and command bytes, the host disables its driver to allow the transmitter to transmit the
Address/Command echo and the requested data.
Before the driver is disabled, the software must insure that
the command byte has been completely transmitted. This can
be done by observing control flags from ‘UART’ of the
communication port, such as Transmit Register Empty (TRE)
and Transmit Holding Register Empty (if the UART is double
buffered). Software delay methods based on maximum
character transmission times for 4800 Baud rates can also be
used but are less reliable. Once it has been verified that the
command byte ‘0’ has been completely transmitted, an
additional delay should be added before the driver is disabled.

This delay ‘T5’ will insure that data has propagated the
network wiring before the driver goes to the high impedance
(disabled) state. A delay period of ‘T5’ = 1 millisecond is
adequate for most long cable runs. The maximum delay
allowed for ‘T5’ is based on the fact that time period ‘T6’ is
fixed in the DDA hardware to be 22 (+/- 2) milliseconds. The
host driver should be disabled well before (at least 5
milliseconds) the DDA transmitter enables its driver and
begins transmission of the Address/Command echo.
Assuming the maximum delay of 5 milliseconds for period
‘T3’, and 2.3 milliseconds for ‘T4’, and that the host driver
should be disabled for 5 milliseconds before the transmitter
begins transmitting data, the maximum delay for ‘T5’ then is
7.7 milliseconds.

Note:
If ‘T3’ is less than 5 milliseconds, then the maximum delay for ‘T5’ can
be extended by the difference (5 milliseconds - T3 actual).

5. The transmitter will begin to transmit the Address/Command
echo in 22 (+/- 2) milliseconds after the address byte is
received from the host computer. This is defined as period
‘T6’ and is fixed by the DDA hardware. Based on a Baud rate
of 4800, the address echo is transmitted in 2.3 milliseconds
(period ‘T7’). The interbyte delay period ‘T8’ for the DDA
transmitter is fixed at 0.1 milliseconds and the command
echo is transmitted 2.3 milliseconds (period ‘T9’).

6. Period ‘T10’ is the time required for the DDA electronics
to perform the requested command. This is a variable
delay based on the command requested. The typical
transmitter response time for each command is listed in
section ‘11.4 DDA command definitions’.

7. Period ‘T11’ is the time required for the DDA electronics to
transmit the data for the requested command. This is a
variable delay based on the command requested. The
typical data transmission time for each command is listed in
section ‘11.4 DDA command definitions’.

8. After the transmitter has completed the data transmission
for the requested command, it will disable its driver and
go back to inactive mode. The transmitter electronics
require 50 milliseconds to transition from active mode to
inactive mode. Another transmitter (or the same transmitter)
cannot be interrogated until time period ‘T12’ = 50 milli-
seconds has elapsed.

9. Repeat the sequence for the next transmitter sequences also
provides information about host computer control of the RS-485
communication card and also illustrates driver enabled control
through the RTS control line.

Other protocol considerations

1. The transmitted ASCII data from the DDA transmitter may
contain data fields with ‘Eexx’ error codes. All DDA error codes
are preceded by ASCII ‘E’ (45 hex, 69 decimal). Communica-
tion interface drivers must parse and handle DDA error codes
properly or data processing errors could result. For additional
information about DDA error codes, (see page 54 ).

2. Use the DDA ‘Data Error Detection’ function to verify the
integrity of the data transmitted from the transmitter.

3. Certain RS-485 communication cards and (RS-232 to RS-485
converter cards) allow user control of the receiver function.
This feature must be considered when developing com-
munication drivers. Due to the half-duplex RS-485 loopback
wire connections, all data that is transmitted by the host
computer device will be ‘echoed’ into the receiver inputs.
If the receiver function is enabled, then the host transmitted
data along with the DDA transmitter transmitted data will be
received into the computer receive buffer.

10. DDA Command definitions

9.1 Special control commands

Command 00 Hex (0 Dec) - Transmitter disable command

This command can be used to disable an active transmitter (force
transmitter back to sleep mode). This command does not need to
be preceded by an address byte and can only be issued when DDA
transmitters are not transmitting data. This ‘disabled’ command is
typically used with other commands that could leave the transmitter
in active mode, i.e. certain memory transfer commands, test mode
commands, etc.

Note:
During normal mode operation, a DDA transmitter will force itself back
into sleep mode if any data is transmitted on the network by any other
device. This is a safety feature added to the firmware to avoid data
collisions on the network.

Command 01 Hex (1 Dec): Module identification command
Data format:
<STX><DDA><ETX><cccccc>
- Fixed length record containing 3 ASCII characters ‘<DDA>’
- Five (5) character checksum appended after ‘<ETX>’ character set
10. DDA Command definitions

10.1 Special control commands

Command 00 Hex (0 Dec) - Transmitter disable command

This command can be used to disable an active transmitter (force transmitter back to sleep mode). This command does not need to be preceded by an address byte and can only be issued when DDA transmitters are not transmitting data. This 'disabled' command is typically used with other commands that could leave the transmitter in active mode, i.e. certain memory transfer commands, test mode commands, etc.

**Note:**

During normal mode operation, a DDA transmitter will force itself back into sleep mode if any data is transmitted on the network by any other device. This is a safety feature added to the firmware to avoid data collisions on the network.

Command 01 Hex (1 Dec): Module identification command

Data format:

<STX><DDA><ETX><ccccc>

- Fixed length record containing 3 ASCII characters '<DDA>'
- Five (5) character checksum appended after '<ETX>' character set

Command 02 Hex (2 Dec): Change address

Data format:

<SOH><ddd><EOT>

- Fixed length record with three (3) characters
- The data field is the new address
- The data range is from 192 - 253
- '<SOH>' is ASCII 01 Hex
- '<EOT>' is ASCII 04 Hex
- Default Address is '192'

Command 03 Hex - Command 09 - Not Defined

10.2 Level commands

Command 0A Hex (10 Dec): Output level 1 (product) at 0.1 inch resolution (with checksum)

Data format:

<STX><dddd.d><ETX><ccccc>

- Variable length record with one (1) to four (4) characters to the left of decimal character
- Fixed at two (2) characters to the right of decimal character
- Five (5) character checksum appended after the '<ETX>' character set

Command 0B Hex (11 Dec): Output level 1 (product) at 0.01 inch resolution (with checksum)

Data format:

<STX><dddd.dd><ETX><ccccc>

- Variable length record with one (1) to four (4) characters to the left of decimal character
- Fixed at three (3) characters to the right of decimal character
- Five (5) character checksum appended after '<ETX>' character set

Command 0C Hex (12 Dec): Output level 1 (product) at 0.001 inch resolution (with checksum)

Data format:

<STX><dddd.ddd><ETX><ccccc>

- Variable length record with one (1) to four (4) characters to the left of decimal character
- Fixed at four (4) characters to the right of decimal character
- Five (5) character checksum appended after '<ETX>' character set

Command 0D Hex (13 Dec): Output level 2 (interface) at 0.1 inch resolution (with checksum)

Data format:

Same as Command 0A

Command 0E Hex (14 Dec): Output level 2 (interface) at 0.01 inch resolution (with checksum)

Data format:

Same as Command 0B

Command 0F Hex (15 Dec): Output level 2 (interface) at 0.001 inch resolution (with checksum)

Data format:

Same as Command 0C

Command 10 Hex (16 Dec): Output level 1 (product) and level 2 interface at 0.1 inch resolution (with checksum)

Data format:

<STX><dddd.d:dddd.d><ETX><ccccc>

- Variable length record with one (1) to four (4) characters to the left of each decimal character in each data field
- Fixed at one (1) character to the right of each decimal character in each data field
- Level 1, level 2 data fields separated by ASCII colon (:) character
- Five (5) character checksum appended after the '<ETX>' character set

Command 11 Hex (17 Dec): Output level 1 (product) and level 2 (interface) at 0.01 inch resolution (with checksum)

Data format:

<STX><dddd.d:dddd.d><ETX><ccccc>

- Variable length record with one (1) to four (4) characters to the left of each decimal character in each data field
- Fixed at two (2) characters to the right of each decimal character in each data field
- Level 1, level 2 data fields separated by ASCII colon (:) character
- Five (5) character checksum appended after the '<ETX>' character set

Command 12 Hex (18 Dec): Output level 1 (product) and level 2 (interface) at 0.001 inch resolution (with checksum)
Data format: `<STX><ddd.dd:ddd.dd:ddd.dd:ddd.dd><ETX><cccc>`

- Variable length record with one (1) to four (4) characters to the left of each decimal character in each data field
- Fixed at three (3) characters to the right of each decimal character in each data field
- Level 1, level 2 data fields separated by ASCII colon (:) character
- Five (5) character checksum appended after the `<ETX>` character set

**Note:**
Average temperature is the average temperature reading from all DTs submerged by approximately 1.5 inches of product.

**Command 13 Hex - Command 18 Hex - Not Defined**

### 10.1 Temperature commands

**Command 19 Hex (25 Dec):** Average Temperature at 1.0 °F resolution (with checksum)

Data format: `<STX><dddd><ETX><cccc>`

- Variable length record with one (1) to four (4) characters
- Five (5) character checksum appended after the `<ETX>` character set

**Command 1A Hex (26 Dec):** Average temperature at 0.2 °F resolution (with checksum)

Data format: `<STX><dddd.d><ETX><cccc>`

- Fixed at one (1) character to the right of decimal character
- Five (5) character checksum appended after the `<ETX>` character set

**Command 1B Hex (27 Dec):** Average temperature at 0.02 °F resolution (with checksum)

Data format: `<STX><dddd.dd><ETX><cccc>`

- Variable length record with one (1) to four (4) characters
- Fixed at two (2) characters to the right of decimal character
- Five (5) character checksum appended after the `<ETX>` character set

**Command 1C Hex (28 Dec):** Individual DT temperature at 0.02 °F resolution (with checksum)

Data format: `<STX><dddd.dd:dddd.dd:dddd.dd:dddd.dd:dddd.dd><ETX><cccc>`

- Variable length record with one (1) to four (4) characters
- Fixed at three (3) characters to the right of decimal character
- Five (5) character checksum appended after the `<ETX>` character set

**Command 1D Hex (29 Dec):** Individual DT temperature at 1.0 °F resolution (with checksum)

Data format: `<STX><dddd.ddd:dddd.ddd><ETX><cccc>`

- Variable length record with one (1) to four (4) characters
- Fixed at one (1) character to the right of decimal character
- Five (5) character checksum appended after the `<ETX>` character set

**Command 1E Hex (30 Dec):** Individual DT temperature at 0.02 °F resolution (with checksum)


- Variable length record with one (1) to four (4) characters
- Fixed at two (2) characters to the right of decimal character
- Five (5) character checksum appended after the `<ETX>` character set

**Command 1F Hex (31 Dec):** Average and individual DT temperature at 1.0 °F resolution (with checksum)

10.1 Multiple Output Commands (continued)

Command 29 Hex (41 Dec):  
Level 1 (product) at 0.01 inch resolution, 
and average temperature at 0.2 °F 
resolution (with checksum)

Data format: 
<STX><dddd.dd:dddd.dd:dddd><ETX><ccccc>

- Variable length record with one (1) to four (4) characters to the left 
of decimal character in first data field 
- Fixed at two (2) characters to the right of decimal character in first data field 
- Variable length record with one (1) to four (4) characters to the left 
of decimal character in second data field 
- Fixed at one (1) character to the right of decimal character in second data field. 
- Level 1, temperature data fields separated by ASCII colon (:) character. 
- Five (5) character checksum appended after the ‘<ETX>’ character set

Command 2A Hex (42 Dec):  
Level 1 (product) at 0.001 inch resolution, 
and average temperature at 0.02 °F resolution (with checksum)

Data format: 
<STX><dddd.dd:dddd.dd:dddd><ETX><ccccc>

- Variable length record with one (1) to four (4) characters to the left 
of decimal character in first data field 
- Fixed at three (3) characters to the right of decimal character in first data field 
- Variable length record with one (1) to four (4) characters to the left 
of decimal character in second data field 
- Fixed at two (2) characters to the right of decimal character in second data field 
- Level 1, temperature data fields separated by ASCII colon (:) character. 
- Five (5) character checksum appended after the ‘<ETX>’ character set

Command 2B Hex (43 Dec):  
Level 1 (product), level 2 (interface) at 0.1 inch resolution, 
and average temperature at 1.0 °F resolution (with checksum)

Data format: 
<STX><dddd.d:dddd.d:dddd.d:dddd><ETX><ccccc>

- Variable length record with one (1) to four (4) characters to the left 
of decimal character in first data field 
- Fixed at one (1) character to the right of decimal character in first data field 
- Variable length record with one (1) to four (4) characters to the left 
of decimal character in second data field 
- Fixed at one (1) character to the right of decimal character in second data field. 
- Level 1, level 2, temperature data fields separated by ASCII colon (:) character. 
- Five (5) character checksum appended after the ‘<ETX>’ character set

Command 2D Hex (45 Dec):  
Level 1 (product), level 2 (interface) at 0.001 inch resolution, 
and average temperature at 0.02 °F resolution (with checksum)

Data format: 
<STX><dddd.d:dddd.d:dddd.d:dddd><ETX><ccccc>

- Variable length record with one (1) to four (4) characters to the left 
of decimal character in first data field. 
- Fixed at three (3) characters to the right of decimal character in first data field. 
- Variable length record with one (1) to four (4) characters to the left 
of decimal character in second data field. 
- Fixed at three (3) characters to the right of decimal character in second data field. 
- Variable length record with one (1) to four (4) characters to the left 
of decimal character in third data field. 
- Fixed at two (2) characters to the right of the decimal character in third data field. 
- Level 1, level 2, temperature data fields separated by ASCII colon (:) character. 
- Five (5) character checksum appended after the ‘<ETX>’ character set

Command 2E Hex - Command 30 Hex - Not Defined

Command 31 Hex - Command 40 Hex - Reserved for factory use

10.2 High-level memory read commands

Command 4B Hex (75 Dec):  
Read ‘number of floats and number of DTs’ control variables

Data format: 
<STX><dddd.d><ETX><ccccc>

- Fixed length record with one (1) character in each field. 
- The first data field is the number of floats, second data field is the number of DTs 
- Five (5) character checksum appended after the ‘<ETX>’ character set

Command 4C Hex (76 Dec):  
Read ‘gradient’ control variable

Data format: 
<STX><d:ddd:ddd><ETX><ccccc>

- Fixed length record with seven (7) characters (including decimal point). 
- Five (5) character checksum appended after the ‘<ETX>’ character set

Command 4D Hex (77 Dec):  
Read float zero position data
Data format:

Data format:

Command 4D Hex (77 Dec) (continued):
- Variable length record with one (1) to four (4) characters to the left of decimal character in first data field. The data may include an ASCII (-) negative sign character (2D Hex) in the first character position
- Fixed at three (3) characters to the right of decimal character in first data field
- Variable length record with one (1) to four (4) characters to the left of decimal character in second data field. The data may include an ASCII(-) negative sign character (2D Hex) in the first character position
- Fixed at three (3) characters to the right of decimal character in second data field
- Float #1, float #2 data fields separated by ASCII colon (:) character
- Five (5) character checksum appended after the `<ETX>` character set

Command 4E Hex (78 Dec):
Read DT position data (DTs 1 - 5)

Data format:

- Variable length record with one (1) to four (4) characters to the left of decimal character in each data field
- Fixed at one (1) character to the right of decimal character in each data field
- Variable number of data fields (up to 5) separated by ASCII colon (:) characters. The number of data fields is based on the 'number of DTs' control variable. (see command 4B Hex)
- The first data field is always DT #1, second field is always DT #2, etc.
- Five (5) character checksum appended after the `<ETX>` character set

Note:

DT position data is referenced from the mounting flange of the transmitter housing. DT #1 is the DT closest to the tip of the transmitter.

Command 4F Hex (79 Dec):
Read factory serial number data and software version number

Data format:

- Fixed length record of 50 characters to the left of the colon character and 6 characters to the right of the colon character (57 total)
- Five character checksum appended after the `<ETX>` character set

Command 50 Hex (80 Dec):
Read firmware control code #1

Data format:

- Fixed length record with one (1) character in each data field
- First data field is the control variable for the data error detection (DED) mode
- The second data field is the control variable for the communication time-out timer (CTT)
- The third data field is the control variable for temperature data units
- The fourth data field is the control variable for linearization enable/disble
- The fifth data field is the control variable for innage/ullage level output
- The sixth data field is reserved for future use; the output value for this field is ASCII '0'
- See write command (5A Hex) for field value assignments
- Five (5) character checksum appended after the `<ETX>` character set

Command 51 Hex (81 Dec): Read hardware control code #1

Data format:

- Fixed length record with six (6) characters
- The hardware control code controls various functions in the DDA electronic hardware
- The hardware control code must match the hardware control code stamped on the transmitter label; the control code on the label is preceded by 'CC' (for example, CC001122)
- Five (5) character checksum appended after the `<ETX>` character set
- For additional information about the hardware control code, (see section 8, Quick Start-up Guide Modbus and DDA)

Command 52 Hex (82 Dec): Not defined

Command 53 Hex (83 Dec): Reserved for factory use

Command 54 Hex (84 Dec): Not defined

10.3 High-level memory write commands

Command 55 Hex (85 Dec): Write 'number of floats and number of DTs' control variables

Host Issued Command (Part 1)

Data format:

- `<addr>` is the DDA transmitter address
- `<commands>` is DDA command 55
- After the address and command byte have been transmitted by the host, the respective DDA transmitter will 'wake up' and retransmit (echo) the local DDA address and received command. The DDA transmitter will remain active, waiting for the second part of the memory write command to be issued by the host. If the second part of the memory write command is not received within 1.0 seconds (see note below), or the command is not received in the proper format, the DDA transmitter will cancel the current command sequence and go back to sleep mode.

Note:
The time-out timer function can be enabled or disabled.

Host Issued Command (Part 2)

Data format:

- Fixed length record with two (2) data fields
- `<SOH>` is ASCII 01 Hex
- The first data field contains the 'number of floats' value to be written to the 'number of floats' control variable. This variable is limited to a value of 1 or 2 (ASCII)
- The second data field contains the 'number of DTs' value to be written to the 'number of DTs' control variable. This variable is limited to a value between 0 and 5 (ASCII)
• ASCII colon (:) is the ‘number of floats/number of DTs’ field separator.
• ‘<EOH>’ is ASCII 04 Hex

**DDA Transmitter Response (verification sequence)**

**Data format:**

```
<STX><:d:d><ETX><:cccccc>
```

- Fixed length record with two (2) data fields
- ‘<STX>’ is ASCII 02 Hex
- The first data field contains the ‘number of floats’ value to be written to the ‘number of floats’ control variable. This variable is limited to a value of 1 or 2 (ASCII)
- The second data field contains the ‘number of DTs’ value to be written to the ‘number of DTs’ control variable. This variable is limited to a value between 0 and 5 (ASCII)
- ASCII colon (:) is the ‘number of floats/number of DTs’ field separator
- ‘<ETX>’ is ASCII 03 Hex
- ‘<:cccccc>’ is a five (5) character checksum appended after the ‘<ETX>’ character set

**Host Issued Command (Part 3)**

**Data format:**

```
<EOH>
```

- ‘<EOH>’ is ASCII 05 Hex. This character set is sent by the host to initiate the EEPROM memory locations have been successfully written to, the DDA transmitter will respond back to the host with a ‘<ACK>’ character set signifying the memory write cycle was successful, or with a ‘<NAK>’ character signifying the memory write cycle was unsuccessful. See DDA transmitter response below.
- EEPROM write time is 10 milliseconds per byte. The ‘<ACK>/<NAK>’ response will not be transmitted by the DDA transmitter until the memory bytes have been written and verified or a memory write error has caused the DDA transmitter to time-out.

**Note:**

EEPROM write time is 10 milliseconds per byte. The ‘<ACK>/<NAK>’ response will not be transmitted by the DDA transmitter until the memory bytes have been written and verified or a memory write error has caused the DDA transmitter to time-out.

**DDA Transmitter Response:**

**Data format:**

```
<ACK>
```

- ‘<ACK>’ is ASCII 06 Hex. This character set is sent by the DDA transmitter to confirm to the host that the EEPROM memory write cycle was completed successfully.

**Data format:**

```
<NAK><:Exxx><ETX><:cccccc>
```

- ‘<NAK>’ is ASCII 15 Hex. This character set is sent by the DDA transmitter to confirm to the host that the EEPROM memory write cycle was not completed successfully.
- ‘<:Exxx>’ is an error code defining the memory write error that occurred during the EEPROM write cycle. ‘E’ is ASCII 45 Hex and ‘xxx’ is the numeric ASCII error code ranging from 000 to 999. For additional information about DDA error codes, see section X.X.
- ‘<ETX>’ is ASCII 03 Hex
- ‘<:cccccc>’ is a five character checksum appended after the ‘<ETX>’ character set
- Value can range from 00000 to 65535.

All high level memory write commands adhere to the communication sequence as described above,

and consist of the following six components:

1. **Host issued command (Part 1):** `<address><command>`
2. **DDA transmitter response:** `<address><command> echo`
3. **Host issued command (Part 2):**
   - `data to be written (including necessary control characters)`
4. **DDA transmitter response:** `verification sequence`
5. **Host issued command (Part 3):** `<EOH>`
6. **DDA transmitter response:** `<ACK>` or `<NAK>`

**Descriptions for other high level memory write commands will include only the Data format for Part 2 of each host issued command.**

**Command 56 Hex (86 Dec):** *Write ‘gradient’ control variable*

**Data format:**

```
<SOH><:ddd:<ETX><:cccccc>
```

- Fixed length record with one data field
- ‘<SOH>’ is ASCII 01 Hex
- The fixed length data field contains the ‘gradient’ value to be written to the ‘gradient’ control variable. This variable is limited to a value between 0.00000 and 9.99999 (ASCII)
- ‘<:cccccc>’ is ASCII 04 Hex

**Command 57 Hex (87 Dec):** *Write float zero position data (float #1 or #2)*

**Data format:**

```
<SOH><:ddd:<EOT>
```

- Variable length record with two (2) data fields
- The first data field contains one character that controls which zero position memory location is written to (i.e., float #1 or float #2). This control character is limited to a value of 1 or 2 (ASCII)
- The second data field contains the ‘zero position’ data value to be written to the ‘zero position’ memory location. This is a variable length data field with one (1) to four (4) characters to the left of the decimal character and fixed at three (3) characters to the right of the decimal character. The data may include the ASCII (-) negative sign character (2D Hex) in the first position. The zero position data is limited to a value between -999.999 and 9999.999 (ASCII)
- ‘<:cccccc>’ is ASCII 04 Hex

**Note:**

Zero position is referenced from the mounting flange of the transmitter housing.

**Command 58 Hex (88 Dec):** *Write float zero position data (float #1 or #2) using DDA calibrate mode.*

**Data format:**

```
<SOH><:ddd:<EOT>
```

- Variable length record with two (2) data fields
- The first data field contains one character that controls which zero position memory location is written to (i.e., float #1 or float #2). This control character is limited to a value of 1 or 2 (ASCII)
- The second data field contains the ‘current float position’ data value to be used to calculate the ‘zero position’ value that is to be written to the ‘zero position’ memory location. This is a variable length data field with one (1) to four (4) characters to the left of the decimal character and fixed at three (3) characters to the right of the decimal character. The data may include the ASCII (-) negative sign character (2D Hex) in the first position. The ‘current float position’ data is limited to a value between -999.999 and 9999.999 (ASCII)
- ‘<:cccccc>’ is ASCII 04 Hex

**Command 59 Hex (89 Dec):** *Write DT position data (DT1-5).*

**Data format:**

```
<SOH><:ddd.d><EOT>
```

- Variable length record with two (2) data fields
The first data field contains one (1) character that controls which ‘DT position’ memory location is written to (i.e. DT position #1, 2, 3, 4 or 5).

This control character is limited to a value between 1 and 5 (ASCII).

The second data field contains the ‘DT position’ data value to be written to the respective ‘DT position’ memory location. This is a variable length data field with one (1) to four (4) characters to the left of the decimal character and fixed at one (1) character to the right of the decimal character. The DT position data is limited to a value between 0.0 and 9999.9 (ASCII).

'<EOT>' is ASCII 04 Hex.

Command 5A Hex (90 Dec): Write firmware control code #1

Data format:
• Fixed length record with one character in each data field
• '<SOH>' is ASCII 01 Hex
• The first data field is the control variable for the data error detection (DED) function. This variable can have a value of 0, 1, or 2. A value of 0 enables the DED function, using a 16-bit checksum calculation. A value of 1 enables the DED function, using a 16-bit CRC calculation. A value of 2 disables the DED function.
• The second field is the control variable for the communication time-out timer (CTT) function. This variable can have a value of 0 or 1. A value of 0 enables the CTT function, and a value of 1 disables the CTT function.
• The third field is the control variable for temperature data units. This variable can have a value of 0 or 1. A value of 0 enables Fahrenheit temperature units. A value of 1 enables Celsius temperature units.
• The fourth field is the control variable for linearization control. This variable can have a value of 0 or 1. A value of 0 disables linearization of the level data. A value of 1 enables linearization.
• The fifth field is the control variable for image/ullage level output. This variable can have a value of 0.1 or 2. A value of 0 enables normal image level output. A value of 1 enables ullage level output and a value of 2 enables ullage level output with reversed DT submersion processing. Mode 2 is used for inverted transmitter applications where the transmitter is installed from the bottom of the tank.
• The sixth field is reserved for future use. The data value for this field must be '0' (ASCII 30 Hex).
• '<EOT>' is ASCII 04 Hex.

Command 5B Hex (91 Dec): Write hardware control code #1

Data format:
• Fixed length record with six (6) characters
• '<SOH>' is ASCII 01 Hex
• The hardware control code controls various functions in the DDA electronic hardware.
• The hardware control code must match the hardware control code stamped on the transmitter label. The control code on the label is preceded by 'CC' (i.e. CC001122).
• '<EOT>' is ASCII 04 Hex.

Command 5C Hex (92 Dec): Not Defined

Command 5D Hex (93 Dec): Reserved for factory use

Command 5F Hex - 7F Hex: Reserved for future use

10.4 Diagnostic/Special command set

<table>
<thead>
<tr>
<th>enum alarmStatusBits</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTERFACE_ALARM_HIGH = 0x0001</td>
</tr>
<tr>
<td>INTERFACE_ALARM_LOW  = 0x0002</td>
</tr>
<tr>
<td>PRODUCT_ALARM_HIGH   = 0x0004</td>
</tr>
<tr>
<td>PRODUCT_ALARM_LOW    = 0x0008</td>
</tr>
<tr>
<td>ROOF_ALARM_HIGH      = 0x0010</td>
</tr>
<tr>
<td>ROOF_ALARM_LOW       = 0x0020</td>
</tr>
<tr>
<td>AVG_TEMP_ALARM_HIGH  = 0x0040</td>
</tr>
<tr>
<td>AVG_TEMP_ALARM_LOW   = 0x0080</td>
</tr>
<tr>
<td>MAGNET_IS_MISSING    = 0x0100</td>
</tr>
<tr>
<td>DIG_TEMP0_ERROR      = 0x0200</td>
</tr>
<tr>
<td>DIG_TEMP1_ERROR      = 0x0400</td>
</tr>
<tr>
<td>DIG_TEMP2_ERROR      = 0x0800</td>
</tr>
<tr>
<td>DIG_TEMP3_ERROR      = 0x1000</td>
</tr>
<tr>
<td>DIG_TEMP7_ERROR      = 0x2000</td>
</tr>
<tr>
<td>DIG_AVG_TEMP_ERROR   = 0x4000</td>
</tr>
<tr>
<td>DELIVERY_IN_PROGRESS = 0x8000</td>
</tr>
<tr>
<td>TRIGGER_LEVEL_ERROR  = 0x10000</td>
</tr>
<tr>
<td>EEPROM_ERROR         = 0x20000</td>
</tr>
</tbody>
</table>

10.5 DDA Error codes

All error codes are preceded by a capital letter ‘E’ ASCII (45 hex) and are in the form of ‘Exxx’ where ‘xxx’ can be any number from ‘000’ to ‘999’. Error codes can be embedded in any data field within a transmitted record. Certain DDA commands can generate multiple error codes. Refer to the following examples:

Command 0A Hex:
<STX><Exxx><ETX><cccccc>

Command 2D Hex:
<STX><Exxx:Exxx:ddd.dd><ETX><cccccc>

Command 1E Hex:
<STX><E203:dddd.dd:dddd.dd:E207:dddd.dd><ETX><cccccc>

E102: Missing Float(s) (Level 1 or Level 2)
The number of floats measured by the hardware is less than the ‘number of floats’ control variable.

E201: No DTs Programmed
A request for temperature data has been made with the ‘number of DTs’ control variable set to equal zero (0) or all programmed DTs are set inactive (for example, DT position data is set equal to zero (0.000)).

E212: DT Communication Error
The indicated DT is not active (for example, DT position data is set equal to zero (0) or is not responding).
11. Digital setup software installation, setup and calibration

Adjustments to the calibration and set up parameters of the transmitter can be performed using the M-Series Digital Setup Software package. The software can be run from any PC using a RS-485 to RS-232 converter (see Table 10 MTS part number references). In the 'MTS Digital Gauge Configuration - DDA -COM' window, you will see one tab labeled 'Data From Device' (see Figure 28). You will use this tab and its button selections to calibrate the transmitter and change setup parameters.

**Note:**

You must use a RS-485 converter with 'Send Data Control' when using the M-Series Digital Setup software to ensure proper operation. Example: B & B Electronics 485BAT3 (815-433-5100 www.bb-elec.com).

<table>
<thead>
<tr>
<th>Level Plus PC Digital Setup Software (DDA) CD</th>
<th>RS-485 to RS-232 converter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order number: 625053</td>
<td>Order number: 380075</td>
</tr>
</tbody>
</table>

Table 1. MTS part number references

### 11.1 Data from device tab

Perform the following steps to install the transmitter setup software to establish communications with the transmitter:

1. Install Setup Software from the CD that came with your transmitter or go to www.mtssensors.com to download the latest version.
2. Connect transmitter to the RS-485 to RS-232 converter and attach the converter to your PC. Some PC’s will require an additional Serial to USB converter.
3. Open the Software program.
4. Select COM Port. If you do not know which COM port to select, right click My Computer and select Properties -> Hardware Tab -> Device Manager -> Ports (COM & LPT) to view the list.
5. Click the 'Data From Device' tab, click the Device: pull-down and select the 'transmitter address', the factory default for DDA is 192 (see Figure 28).

Parameter settings and calibration is performed from within the Data From Device tab window (see Figure 28).

### DATA FROM DEVICE tab options:

- Calibrate
- Change Address
- Adjust settings
- COM port
- Backup and restore device settings

**Calibration**

When you click the 'Calibrate' button in the 'Data From Device' tab window, the 'Calibrate DDA Device' window opens. There are two calibration 'Float Methods' to choose from, 'Enter Float Positions (Calibrate)' and 'Enter Float Zero Positions'. Click the 'Offset Method:' drop down menu to select a calibration method. Type a value in the active field, then click the 'Send' button. A confirmation window displays when the send is successful.

**Enter Float Zero Positions**

When you choose 'Enter Float Zero Positions' from the 'Offset Method:' drop down menu, you can adjust the offset where the transmitters zero point is located. This adjustment will significantly shorten the span of the transmitter or counter inactive zones. Adjust the value accordingly and click 'Send'. A confirmation window displays when the send is successful.
Change Address

To change the transmitter address, click the ‘Change Address’ button in the ‘Data From Device’ tab window. In the ‘Change Address’ window, type the ‘New Address’ in the active field and click ‘Change’. A popup window confirms the change is successful.

![Change Address window - New Address entry](image)

Backup and Restore Device Settings

If your electronics requires a replacement or if your current settings need to be refreshed, it is recommended that you create a backup or restoration file. To create a backup, click the ‘Backup/Restore’ button in the ‘Data From Device’ tab window. In the ‘Backup and Restore Device Settings’ window, click the ‘Get Data From Sensor’ button and ‘Save Settings to File’ button. When prompted, save the file to a designated place where you can find it.

To upload a file, click the ‘Read Settings from File’ button and select your backup file. Click ‘Write Data to Sensor’. A popup window confirms the upload is successful.

![Backup and Restore Device Settings window](image)

Adjust settings

To adjust settings, click the ‘Adjust’ button located in the ‘Data From Device’ tab window (see Figure 28). The ‘Adjust DDA Gain’ window displays different parameter settings. All transmitters will have the ability to adjust the ‘Gain, SARA Blanking and Magnet blanking’ from this menu. These parameters are password protected, changes will require assistance from MTS Technical Support.

![Modbus Adjust Gain window](image)

COM Port

To select the Setup Software communication port, click the ‘COM Port’ button in the ‘Data From Device’ tab window. Select the appropriate communication port and click ‘OK’.

![Select a COM Port window](image)

Continuous Update

To view realtime data using the Setup Software interface, select the ‘Continuous Update’ box. The Interval may be changed to slow down updates but is not necessary.

Data Logging

To download a transmitter data log, Click ‘Select File’ in the ‘Data From Device’ tab window. Select an Excel file and check the ‘Log Data to File’ box to save your data.