Rugged Servo Drives & Control Systems for Extreme Environments

RS422 Protocol Document

Document Number: 100121-00K
Date of Doc Rev: 9/14/2018

Revision History

Version	Date	Items Changed
A		Initial Release
B		Clarified bit-transmission order
C		Expanded protocol for dual-axis operation for the Mite family. Added scaling to the HiDS interface for more flexibility. Added position loop to protocol.
E		Corrected the position scaling and added a command example.
F	1/15/16	Updated to current template format.
G	4/11/16	Update to reflect recent loop input control changes
H	4/12/16	Add Command Mode A and B to command packet
J	10/7/16	Corrected command-enable.
K	9/14/18	Updated to new style guide, logo, copyright, warranty disclaimer and Non-ITAR/EAR legends

Important Information:

TABLE OF CONTENTS:

1 Purpose

This document describes the software requirements of the RS422 protocol for ESI Motion's Servo Drive Motor Controllers. This command interface can be used for general control of the several variants of controllers including the Dragon and Mite line. It also defines the status reporting during operation.

2 Background

ESI Motion servo drives, modules and control systems are a complete ruggedized, off-the-shelf motor control solution which include ESI's rugged controller and power driver boards, an integrated EMI filter, military-grade submersible case, controller software, and user-friendly GUI. These systems are ideal for military, aviation, automotive or other heavy industrial applications operating in outdoor, high temperature, high vibration, or other extreme environmental conditions.

3 Support Information

- ESI Motion Controller User Manual, document number 100266

4 RS422 Overview:

4.1 Message Format

The RS422 message is composed of the following three parts:

1.	Header	(2 bytes)
2.	Body	(8 bytes)
3.	CRC	(2 bytes)
	Total	(12 bytes)

All data content is assumed to be little endian byte order.

4.1.1 Header

All messages start with a Header. Each message begins with the following 2 bytes hexadecimal sequence: 0xF0F0.

4.1.2 Body

The Body follows the Header. The body is either a command packet (transmitted to the servo drive) or a status packet (transmitted from the servo drive). Both are 8 bytes in length. Note that the field definitions are different in the status and command packet.

4.1.3 CRC

Following the message body is a 16-bit CRC of the message body (the header value is pre-checked and not part of the CRC result). The CRC is defined as the CRC-16/CCITT-FALSE (implementation). The C-implementation is shown below and the CRC table referred to is in *Appendix A (Page 11 of this document)*. If the CRC calculated from the message doesn't match the CRC received, the message is discarded and ignored. For a single message CRC calculation, the start crc (starting seed) used shall be 0xFFFF.

```
/*****************************************************************************
 * @fn      crc16()
 *
 * @brief   Implements the "CRC-16/CCITT-FALSE" algorithm. The identification
 *          is shown by ASCII "123456789" = 0x29B1.
 *
 * @param   starting CRC, pointer to the buffer, and the len
 *
 * @return  The CRC-16 result
*****************************************************************************/
U16 crc16(const U16 start_crc, Uchar16 *buf_ptr, U32 len)
{
        U16 crc16 = start_crc;
        while (len--)
        {
                crc16 = crc16_table[(crc16 >> 8) ^ *buf_ptr++] ^ (crc16 << 8);
        }
        return(crc16);
}
```

4.2 Command Packet

The Command Packet is received by the ESI Controller. All commands start with the 2 byte header and end with the 2 byte CRC defined above. The Controller must be configured via HiDS as a current, velocity, or position command; see section 4.4. The command packet information is shown below.

Note all bytes shown below are in the order of transmission (Byte 0 is transmitted first)

Signal name	Byte	Bit	Size (bits)	Data Type	Coding
Header	0-1	B[15..0]	16	Unsigned	Always 0xF0F0
Motor Enable A	2	Bit0	1	unsigned	0 - Motor Disable 1 - Motor Enable
Motor Enable B	2	Bit1	1	unsigned	0 - Motor Disabled 1 - Motor Enabled
Clear all errors	2	Bit2	1	unsigned	0 - Do Not Clear Error 1 - Clear All Errors.
Command Mode A	2	Bit3-4	2	unsigned	0 – Torque Mode 1 – Velocity Mode 2 – Position Mode
Command Mode B	2	Bit5-6	2	unsigned	0 – Torque Mode 1 – Velocity Mode 2 – Position Mode
Unused	2	Bit 7	1	N/A	Reserved
Command Motor A	3-4	B[15..0]	16	Signed	Command Mode and Scaling defined via HiDS Mode 0 – Current Command Mode 1 – Velocity Command Mode 2 – Position Command
Command Motor B	5-6	B[15..0]	16	Signed	Command Mode and Scaling defined via HiDS Mode 0 – Current Command Mode 1 – Velocity Command Mode 2 – Position Command
Reserved	7-9	B[23..0]	24	NA	Reserved
CRC	10-11	B[15..0]	16	NA	CRC

4.2.1 Command Packet Example

For example, to enable both motors, with a +1 Amp MotorA command and a -1Amp MotorB command, the 12-byte packet contents would be: **F0 F0 03 E8 03 18 FC 00 00 00 ED 6A**

4.3 Status Packet

The Status Packet is transmitted by the Controller. All Status Packets start with the 2 byte header and end with the 2 byte CRC defined above. As with the Command Packet, within each byte, bit7 is transmitted first and bit0 is transmitted last. The Status Packet provides the run-time and error information about the drive. The body of the status is shown below.

Signal name	Byte	Bit	Size (bits)	Data Type	Coding
Header	0-1	B[15..0]	16	Unsigned	Always 0xF0F0
Motor Enabled A	2	Bit0	1	Unsigned	0 - Motor Disabled 1 - Motor Enabled
Motor Enabled B	2	Bit1	1	Unsigned	0 - Motor Disabled 1 - Motor Enabled
Error Condition Motor A	2	Bit2	1	Unsigned	0 – No Errors 1 – An Error Occurred
Error Condition Motor B	2	Bit3	1	Unsigned	0 – No Errors 1 – An Error Occurred
Error-Overvoltage	2	Bit4	1	Unsigned	0 – No Error 1 – Over Voltage Error
Error-Undervoltage	2	Bit5	1	Unsigned	0 – No Error 1 – Under Voltage Error
Error-PrechargeV	2	Bit6	1	Unsigned	0 – No Error 1 – Precharge Error
Reserved	2	Bit7	1	NA	Reserved
Feedback Motor A	3-4	B[15..0]	16	Signed	Command Mode and Scaling defined via HiDS Mode 0 – Current Command Mode 1 – Velocity Command Mode 2 – Position Command
Feedback Motor B	5-6	B[15..0]	16	Signed	Command Mode and Scaling defined via HiDS Mode 0 – Current Command Mode 1 – Velocity Command Mode 2 – Position Command
Motor A Error-Overcurrent	7	Bit0	1	Unsigned	0 – No Error 1 – Over Current Error

Motor A Error-LossOfFeedback	7	Bit1	1	Unsigned	0 – No Error 1 – Loss Of Feedback Error
Motor A Error-Overspeed	7	Bit2	1	Unsigned	0 – No Error 1 – Over Speed Error
Motor A Error-MotorTemp	7	Bit3	1	Unsigned	0 – No Error 1 – Motor Temperature Error
Motor A Error-IGBTTemp	7	Bit4	1	Unsigned	0 – No Error 1 – IGBT Temperature Error
Motor A Error-I2T	7	Bit5	1	Unsigned	0 – No Error 1 – I2T Error
Motor A Error-BridgeFault	7	Bit6	1	NA	0 – No Error 1 – Bridge Fault Error
Reserved	7	Bit7	1	NA	Reserved
Motor B Error-Overcurrent	8	Bit0	1	Unsigned	0 – No Error 1 – Over Current Error
Motor B Error-LossOfFeedback	8	Bit1	1	Unsigned	0 – No Error 1 – Loss Of Feedback Error
Motor B Error-Overspeed	8	Bit2	1	Unsigned	0 – No Error 1 – Over Speed Error
Motor B Error-MotorTemp	8	Bit3	1	Unsigned	0 – No Error 1 – Motor Temperature Error
Motor B Error-IGBTTemp	8	Bit4	1	Unsigned	0 – No Error 1 – IGBT Temperature Error
Motor B Error-I2T	8	Bit5	1	Unsigned	0 – No Error 1 – I2T Error
Motor B Error-BridgeFault	8	Bit6	1	NA	0 – No Error 1 – Bridge Fault Error
Reserved	8	Bit7	1	NA	Reserved
Reserved	9	B[7..0]	8	NA	Reserved
CRC	10-11	B[15..0]	16	NA	CRC

4.4 HiDS Configuration Variables

The HIDS tool is used to configure the serial interface. The following sections define the modifiable parameters.

4.4.1 Enabling the Serial Interface

The RS422-control interface can be enabled by selecting the "RS422 Command" as one of the two input-source selections for the desired control-loop (there are independent source-inputs for the torque, velocity, and position loops). These options can be found under the "Loop Gains" tab in HiDS under the active control loop.
The RS422-status interface is enabled if the RS422 Command is selected as an input source.

4.4.2 Command Mode

The command can be set for each axis independently. The valid modes are as follows:

- 0 – Torque
- 1 – Velocity
- 2 – Position

CommandModeA and CommandModeB are used to select the command mode for Motor A and Motor B respectively.

4.4.3 Command Scaling

Each motor includes three variables for command scaling, one for torque, velocity, and position mode. There three variables are as follows:

- CommandCurrentScaleX
- CommandVelocityScaleX
- CommandPositionScaleX

The "X" is ether A or B for Motor A or Motor B. Note these variables are common between the RS422 and CAN interfaces.

If CommandCurrentScaleX = 0.001, the range is -32.768 to +32.767 amps.
If CommandVelocityScaleX = 1, the range is -32,768 to +32,767 RPM.
If CommandPositionScaleX = 0.000191753, the range is -2π to +2π radians.

The scale above is the same as the resolution. Software uses the following equation to determine the drive command:

Command = (Signed 16-Bit Serial Input) * Command Scale

4.4.4 Response Rate

The response rate is set by the variable SerialFeedbackRate_ms. The units are milliseconds between packets.

4.4.5 Baud Rate

The Serial baud rate can be set via the HIDS variable SerialBaud. Note the serial parameters are 8 data bits, no parity, and 1 stop bit (8,N,1) and are fixed.

4.4.6 Error Counters

Every valid command packet received triggers an up-count of the HIDS variable SerialChecksumPassCount. Every invalid packet triggers an up-count of the HIDS variable SerialChecksumErrorCount.

Appendix A (CRC Table)

```
static const U16 crc16_table[256] =
{
        0x0000,0x1021,0x2042,0x3063,0x4084,0x50a5,0x60c6,0x70e7,
        0x8108,0x9129,0xa14a,0xb16b,0xc18c,0xd1ad,0xe1ce,0xf1ef,
        0x1231,0x0210,0x3273,0x2252,0x52b5,0x4294,0x72f7,0x62d6,
        0x9339,0x8318,0xb37b,0xa35a,0xd3bd,0xc39c,0xf3ff,0xe3de,
        0x2462,0x3443,0x0420,0x1401,0x64e6,0x74c7,0x44a4,0x5485,
        0xa56a,0xb54b,0x8528,0x9509,0xe5ee,0xf5cf,0xc5ac,0xd58d,
        0x3653,0x2672,0x1611,0x0630,0x76d7,0x66f6,0x5695,0x46b4,
        0xb75b,0xa77a,0x9719,0x8738,0xf7df,0xe7fe,0xd79d,0xc7bc,
        0x48c4,0x58e5,0x6886,0x78a7,0x0840,0x1861,0x2802,0x3823,
        0xc9cc,0xd9ed,0xe98e,0xf9af,0x8948,0x9969,0xa90a,0xb92b,
        0x5af5,0x4ad4,0x7ab7,0x6a96,0x1a71,0x0a50,0x3a33,0x2a12,
        0xdbfd,0xcbdc,0xfbbf,0xeb9e,0x9b79,0x8b58,0xbb3b,0xab1a,
        0x6ca6,0x7c87,0x4ce4,0x5cc5,0x2c22,0x3c03,0x0c60,0x1c41,
        0xedae,0xfd8f,0xcdec,0xddcd,0xad2a,0xbd0b,0x8d68,0x9d49,
        0x7e97,0x6eb6,0x5ed5,0x4ef4,0x3e13,0x2e32,0x1e51,0x0e70,
        0xff9f,0xefbe,0xdfdd,0xcffc,0xbf1b,0xaf3a,0x9f59,0x8f78,
        0x9188,0x81a9,0xb1ca,0xa1eb,0xd10c,0xc12d,0xf14e,0xe16f,
        0x1080,0x00a1,0x30c2,0x20e3,0x5004,0x4025,0x7046,0x6067,
        0x83b9,0x9398,0xa3fb,0xb3da,0xc33d,0xd31c,0xe37f,0xf35e,
        0x02b1,0x1290,0x22f3,0x32d2,0x4235,0x5214,0x6277,0x7256,
        0xb5ea,0xa5cb,0x95a8,0x8589,0xf56e,0xe54f,0xd52c,0xc50d,
        0x34e2,0x24c3,0x14a0,0x0481,0x7466,0x6447,0x5424,0x4405,
        0xa7db,0xb7fa,0x8799,0x97b8,0xe75f,0xf77e,0xc71d,0xd73c,
        0x26d3,0x36f2,0x0691,0x16b0,0x6657,0x7676,0x4615,0x5634,
        0xd94c,0xc96d,0xf90e,0xe92f,0x99c8,0x89e9,0xb98a,0xa9ab,
        0x5844,0x4865,0x7806,0x6827,0x18c0,0x08e1,0x3882,0x28a3,
        0xcb7d,0xdb5c,0xeb3f,0xfb1e,0x8bf9,0x9bd8,0xabbb,0xbb9a,
        0x4a75,0x5a54,0x6a37,0x7a16,0x0af1,0x1ad0,0x2ab3,0x3a92,
        0xfd2e,0xed0f,0xdd6c,0xcd4d,0xbdaa,0xad8b,0x9de8,0x8dc9,
        0x7c26,0x6c07,0x5c64,0x4c45,0x3ca2,0x2c83,0x1ce0,0x0cc1,
        0xef1f,0xff3e,0xcf5d,0xdf7c,0xaf9b,0xbfba,0x8fd9,0x9ff8,
        0x6e17,0x7e36,0x4e55,0x5e74,0x2e93,0x3eb2,0x0ed1,0x1ef0
};
```


2250A Union Place
Simi Valley, CA 93065
2250-A Union Place, Simi Valley, CA 93065
P: 800.823.3235
F: 800.823.9521
www.esimotion.com

