

Proprietary

This drawing and all information herein is the proprietary property of Rheodyne
Any copying, reproduction or unauthorized use without written consent is forbidden.



UART/USB Communication Protocol for TitanEX™/TitanEZ™/TitanHP™, TitanHT™ Driver Boards and MX Series II™ Modules

Design Specifications

Titan and Titan HT – UART

Titan and TitanHT boards have UART communication option available to the user. UART (Universal Asynchronous Receiver Transmitter) communication is asynchronous serial communication based on TTL (0-5V) voltage levels. RS-232 type communication can readily be achieved by adding an adapter board based on MAX232 or similar chips.

MX II Series Modules – USB

MX II Series uses an FTDI FT232R chip to translate the UART communication into USB (Universal Serial Bus). The command structure is identical to that of UART, but in order to communicate with the unit through USB the user must download the drivers of <http://www.ftdichip.com/FTDrivers.htm>

The default FTDI drivers can be used as Rheodyne LLC does not have their own USB vendor ID and instead the FTDI's vendor ID and product ID are used in MX II Series Modules.

Protocol Specifications

UART (Universal Asynchronous Receiver Transmitter) communication is asynchronous serial communication based on TTL (0-5V) voltage levels. RS-232 type communication can readily be achieved by adding an adapter board based on MAX232 or similar chips.

Data format is 8N1 – one start bit, 8 data bits, no parity, and one stop bit. No hardware or software handshaking is implemented. Default baud rate is 19.2K, but other speeds are also possible (see below). Each command packet must be terminated by <CR> [0x0D].

Communication is fully ASCII-based and commands may be sent and responses received using a terminal program, such as HyperTerminal or RealTerm.

There are two groups of command packets. The first group is typically used for issuing commands, while the second group mainly enables the user to read out various system parameters.

The first group of command packets always consists of a single character command, followed by a two-byte value. The value must be in ASCII-encoded hexadecimal format. If the driver board recognizes a valid command and executes it, it will respond with <CR>. If the command is not recognized, or there was an error executing a valid command, there will be no response.

Example

To move the valve to position 10, the following command packet must be sent:

P0A<CR> [0x50 0x30 0x41 0x0D]

The unit will respond to the command with: <CR>

The second group of command packets consists of a single character command, followed by <CR>.

Example

To read the valve status, the following command packet must be sent:

S<CR> [0x53 0x0D]

If we assume that the valve is in position 5, the unit will respond with: 05<CR>

Communication with TitanEX™/TitanHP™, TitanHT™ driver boards and MX Series II™ Modules

Busy Status

During the valve motion profile, driver board will not accept any commands and will respond to any incoming data with '*' [0x2A].

Commands

Command: P

Function: commands the valve to a new position

Value: 0x01 – 0x0X, where X depends on the maximum number of positions allowed for the selected mode of operation [2, 3, 4, 6, 8, A(10), C(12)]

Note: Invalid position commands are ignored by the driver board.

Command: + (only implemented for TitanHP and TitanEX, not implemented for TitanHT or MX Series II)

Function: commands the valve to a new position using counter-clock wise motion (CCW)

Value: 0x01 – 0x0X, where X depends on the maximum number of positions allowed for the selected mode of operation [2, 3, 4, 6, 8, A(10), C(12)]

Note: Invalid position commands are ignored by the driver board.

Command: - (only implemented for TitanHP and TitanEX, not implemented for TitanHT or MX Series II)

Function: commands the valve to a new position using clock wise motion (CW)

Value: 0x01 – 0x0X, where X depends on the maximum number of positions allowed for the selected mode of operation [2, 3, 4, 6, 8, A(10), C(12)]

Note: Invalid position commands are ignored by the driver board.

Command: O

Function: sets valve profile

Value: 0x00 - 0xFF

Note: The new operational mode becomes active after driver board reset. Invalid operational mode will cause error 77 (valve configuration error).

Command: N

Function: sets new slave I2C address

Value: 0x0E - 0xFE (even numbers only)

Note: The new I2C address becomes valid after driver board reset.

Command: F

Function: sets valve command mode

Value: 0x01 - 0x05

Level logic = 0x01

Single pulse logic = 0x02

BCD logic = 0x03

Inverted BCD logic = 0x04

Dual pulse logic = 0x05

Note: The new command mode becomes active after driver board reset. Invalid command mode will cause error 77 (command mode error).

Command: X

Function: sets baud rate for UART communication

Value: 0x01 - 0x04

9600 = 0x01

19200 = 0x02

38400 = 0x03

57600 = 0x04

Note: The new baud rate becomes active after driver board reset.

The following commands are sent without a value:

Command: M

Function: commands the valve to the home position

Command: S

Function: requests valve status

Return Value: two-digit, ASCII-encoded hexadecimal number, followed by <CR>

If no error condition exists, current valve position (01-0A) will be returned.

In case of an error, one of the following codes will be returned:

63 (99 decimal) – valve failure (valve can not be homed)

58 (88 decimal) – non-volatile memory error

4D (77 decimal) – valve configuration error or command mode error

42 (66 decimal) – valve positioning error

37 (55 decimal) – data integrity error

2C (44 decimal) – data CRC error

Command: Q

Function: reads valve profile

Return Value: two-digit, ASCII-encoded hexadecimal number (00-FF), followed by <CR>

Command: R

Function: reads firmware revision

Return Value: two-digit, ASCII-encoded hexadecimal number (00-FF), followed by <CR>

Note: TitanHT board returns firmware revision in uppercase (e.g. 41 for rev.A), while TitanEX board uses lowercase (e.g. 61 for rev.A).

Command: E

Function: reads the latest valve error code

Return Value: two-digit, ASCII-encoded hexadecimal number (00-FF), followed by <CR>

Command: D

Function: reads the valve command mode

Return Value: two-digit, ASCII-encoded hexadecimal number (01-05), followed by <CR>