

# Titan Intelligent BLDC Motor I<sup>2</sup>C Protocol

Overview Ltd  
[www.overview.co.uk](http://www.overview.co.uk)

December 2, 2010

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Message Protocol</b>	<b>1</b>
2.1	Physical Layer . . . . .	1
2.2	Communications Layer . . . . .	1
2.3	Message Structure . . . . .	1
2.3.1	Data Write Message . . . . .	1
2.3.2	Data Request Message . . . . .	2
<b>3</b>	<b>Commands</b>	<b>3</b>
3.1	Summary . . . . .	3
3.2	Detailed Description . . . . .	3

## 1 Introduction

This document describes the use of the I<sup>2</sup>C interface to the Overview Titan Intelligent BLDC Motor. The message protocol for devices on the bus is given, followed by a detailed description of the available motor commands.

## 2 Message Protocol

### 2.1 Physical Layer

A standard 3.3V, two wire, I<sup>2</sup>C interface is provided by the motor consisting of a clock, SCL, and data, SDA, line. Both SDA and SCL need to be pulled up to 3.3V. When the bus is not busy both lines will be high. The SCL signal is always generated by the master and the Titan BLDC motor(s) will always be slaves on the bus. SDA can be controlled by either the master or the slave being communicated to. Up to eight Titan BLDC motors can be controlled on one i2c bus by setting the address lines of the motor appropriately.

### 2.2 Communications Layer

With two exceptions all transitions of SDA will occur when SCL is low. These exceptions being start condition and the stop condition. Start condition (STX): All messages must start with this transition. SDA is set from high to low while SCL is high. Stop condition (ETX): All messages must end with this transition. SDA is set from low to high while SCL is high. Messages are of variable length but are split into 1 byte blocks, most significant bit first, after which an acknowledge bit (ACK /NACK) is required from the receiving party. An ACK bit is low and a NACK bit high.

### 2.3 Message Structure

There are two types of messages, a data write message and a data request message. Shading is used below to indicate whether the bus is under control of the master or slave device during message transmission:

	SDA under master control
	SDA under slave control

#### 2.3.1 Data Write Message

This takes the following form

STX	Control Byte	ACK	Command Byte	ACK	Data 0	ACK 0	Data <i>n</i>	ACK <i>n</i>	ETX
-----	--------------	-----	--------------	-----	--------	-------	---------------	--------------	-----

#### Control Byte

The control byte consists of the control code (0x5) for the Titan BLDC motor, the address to specify which motor the message is intended for and whether the message is a data write or data request message.

Bit	Function
7	Control code bit 3: 0
6	Control code bit 2: 1
5	Control code bit 1: 0
4	Control code bit 0: 1
3	Address bit 2: 1/0
2	Address bit 1: 1/0
1	Address bit 0: 1/0
0	Write/Request bit: 0 (write)

### Command Byte

The command byte specifies the desired action of the message and is described in detail in section 3.

### Data Bytes

Depending on command byte 0, 1, 2, or 3 additional data bytes will be transmitted to complete the message.

#### 2.3.2 Data Request Message

To request data from the Titan BLDC motor two messages are required one to setup up the request and one to deliver the requested data. This is outlined below

#### Setup Request Message

STX	Control Byte	ACK	Command Byte	ACK	ETX
-----	--------------	-----	--------------	-----	-----

##### Control Byte.

Bit	Function
7	Control code bit 3: 0
6	Control code bit 2: 1
5	Control code bit 1: 0
4	Control code bit 0: 1
3	Address bit 2: 1/0
2	Address bit 1: 1/0
1	Address bit 0: 1/0
0	Write/Request bit: 0 (write)

**Command Byte.** See section 3 for a detailed description of request commands.

#### Deliver Requested Data Message

STX	Control Byte	ACK	Data Byte 0	ACK 0	Data Byte <i>n</i>	NACK <i>n</i>	ETX
-----	--------------	-----	-------------	-------	--------------------	---------------	-----

##### Control Byte.

Bit	Function
7	Control code bit 3: 0
6	Control code bit 2: 1
5	Control code bit 1: 0
4	Control code bit 0: 1
3	Address bit 2: 1/0
2	Address bit 1: 1/0
1	Address bit 0: 1/0
0	Write/Request bit: 1 (request)

**Data Bytes.** This is the relevant data as requested from the command in the setup request message. The data payload will be either one or two bytes and the last data byte will be followed by a NACK from the master.

## 3 Commands

### 3.1 Summary

Code	Description	Message type	# data bytes
1	Reset	Data write	0
2	Calibration complete	Data request	1 (in response)
3	Is motor moving	Data request	2 (in response)
4	Return current location	Data request	2 (in response)
5	Goto absolute location	Data write	2
6	Goto relative location	Data write	2
7	Travel at velocity	Data write	2
8	Set acceleration	Data write	2
9	Goto absolute location in set time	Data write	3
10	Goto relative location in set time	Data write	3
11	Get acceleration	Data request	2 (in response)
12	Set tuning algorithm high speed proportional gain	Data write	2
13	Get tuning algorithm high speed proportional gain	Data request	2 (in response)
14	Set tuning algorithm high speed integral gain	Data write	2
15	Get tuning algorithm high speed integral gain	Data request	2 (in response)
16	Set tuning algorithm high speed differential gain	Data write	2
17	Get tuning algorithm high speed differential gain	Data request	2 (in response)
18	Set first endstop distance from mechanical endstop	Data write	2
19	Set mechanical motor range	Data write	2
20	Start calibration	Data write	0
21	Set tuning algorithm low speed proportional gain	Data write	2
22	Get tuning algorithm low speed proportional gain	Data request	2 (in response)
23	Set tuning algorithm low speed integral gain	Data write	2
24	Get tuning algorithm low speed integral gain	Data request	2 (in response)
25	Set tuning algorithm low speed differential gain	Data write	2
26	Get tuning algorithm low speed differential gain	Data request	2 (in response)
27	Set tuning algorithm low/high speed transistion	Data write	2
28	Get tuning algorithm low/high speed transistion	Data request	2 (in response)
30	Get encoder position	Data request	2 (in response)
35	Save pid settings	Data write	0
36	Reload pid defaults	Data write	0

### 3.2 Detailed Description

#### 1 RESET

Purpose	Resets motor
Message Type	Data write message
Notes	Equivalent to power cycling the motor
Command	0x01
Number of data bytes	0

**2 CALIBRATION COMPLETE**

Purpose	Checks if motor is calibrated and ready for normal operation
Message Type	Data request message
Notes	No velocity or position moves can occur until calibration is complete
Command	0x02
Number of data bytes	1 (in response)
Data 0	0x00 - calibration in progress 0x01 - calibration completed successfully 0x02 - calibration failed Else - Error

**3 IS MOTOR MOVING**

Purpose	Returns movement status of motor
Message Type	Data request message
Command	0x03
Number of data bytes	2 (in response)
Data 0	8bit signed value

**4 RETURN CURRENT LOCATION**

Purpose	Returns the current motor location
Message Type	Data request message
Notes	16bit positional accuracy (0 0xFFFF) in motor counts 0xFFFF - Error The position returned is the clockwise distance from the first mechanical endstop in motor counts. 1 degree ~ 182 motor counts Resolution ~ 0.005degrees
Command	0x04
Number of data bytes	2 (in response)
Data 0	MSB of position
Data 1	LSB of position

**5 GOTO ABSOLUTE LOCATION**

Purpose	Moves the motor to an absolute position
Message Type	Data write message
Notes	16bit positional accuracy (0 0xFFFF) in motor counts The position specified is the clockwise distance from the first mechanical endstop in motor counts. 1 degree ~ 182 motor counts Resolution ~ 0.005degrees
Command	0x05
Number of data bytes	2
Data 0	MSB of position
Data 1	LSB of position

## 6 GOTO RELATIVE LOCATION

Purpose	Moves the motor a specified number of counts in a specified direction
Message Type	Data write message
Notes	16bit signed positional accuracy in motor counts 0x8001(180 degrees anticlockwise) 0x7FFF(180 degrees clockwise) 1 degree ~ 182 motor counts Resolution ~ 0.005degrees NB. The move will stop at the software endstop if the supplied range is too large to execute. The maximum relative move is 180
Command	0x06
Number of data bytes	2
Data 0	MSB of position
Data 1	LSB of position

## 7 TRAVEL AT VELOCITY

Purpose	Moves the motor at a specified velocity in a given direction
Message Type	Data write message
Notes	16bit signed velocity range in motor counts per second 0x8001 - max anticlockwise (360 degrees/second) 0 - stopped 7FFF - max clockwise (360 degrees/second) 1 degree/second ~ 91 motor counts/second Resolution ~ 0.01degrees/second
Command	0x07
Number of data bytes	2
Data 0	MSB of speed
Data 1	LSB of speed

## 8 SET ACCELERATION

Purpose	Sets the de/acceleration of the motor
Message Type	Data write message
Notes	10bit acceleration range 0 (minimum) 1 023(max) 0 - 65534 motor counts/second <sup>2</sup> (720degrees/second <sup>2</sup> ) Default value 256 - 163840 motor counts/second <sup>2</sup> (900degrees/second <sup>2</sup> ) 1023 - 327670 motor counts/second <sup>2</sup> (3600degrees/second <sup>2</sup> )
Command	0x08
Number of data bytes	2
Data 0	MSB of acceleration
Data 1	LSB of acceleration

**9 GOTO ABSOLUTE LOCATION IN SET TIME**

Purpose	Moves the motor to an absolute position in a given time
Message Type	Data write message
Notes	16bit positional accuracy (0 0xFFFFE) in motor counts The position specified is the clockwise distance from the first mechanical endstop in motor counts. 1 degree ~ 182 motor counts Resolution ~ 0.005degrees Time: 0 (as fast as possible) to 255 seconds
Command	0x09
Number of data bytes	3
Data 0	MSB of position
Data 1	LSB of position
Data 2	Time

**10 GOTO RELATIVE LOCATION IN SET TIME**

Purpose	Moves the motor a specified number of counts in a specified direction
Message Type	Data write message
Notes	16bit signed positional accuracy in motor counts 0x8001(180 degrees anticlockwise) 0x7FFF(180 degrees clockwise) 1 degree ~ 182 motor counts Resolution ~ 0.005degrees Time: 0 (as fast as possible) to 255 seconds NB. The move will stop at the software endstop if the supplied range is too large to execute. The maximum relative move is 180
Command	0x0A
Number of data bytes	3
Data 0	MSB of position
Data 1	LSB of position
Data 2	Time

**11 GET ACCELERATION**

Purpose	Get the current de/acceleration setting of the motor
Message Type	Data request message
Notes	10bit acceleration range 1 (minimum) 1 023(max) 0xFFFF Error 0 - 65534 motor counts/second <sup>2</sup> (720degrees/second <sup>2</sup> ) Default value 256 - 163840 motor counts/second <sup>2</sup> (900degrees/second <sup>2</sup> ) 1023 - 327670 motor counts/second <sup>2</sup> (3600degrees/second <sup>2</sup> )
Command	0x0B
Number of data bytes	2 (in response)
Data 0	MSB of acceleration
Data 1	LSB of acceleration

**12 SET TUNING ALGORITHM HIGH SPEED PROPORTIONAL GAIN**

Purpose	Set the gain level of the high speed proportional control stage of the tuning PID loop
Message Type	Data write message
Notes	16bit unsigned fixed point (0-0xFFFFE) Range $0 < x < 32$ Resolution - 0.005 Value to send - $x \cdot 2^{11}$ Default - 5.5 (0x2C00)
Command	0x0C
Number of data bytes	2
Data 0	MSB of proportional gain
Data 1	LSB of proportional gain

**13 GET TUNING ALGORITHM HIGH SPEED PROPORTIONAL GAIN**

Purpose	Get the gain level of the high speed proportional control stage of the tuning PID loop
Message Type	Data request message
Notes	16bit unsigned fixed point (0-0xFFFFE) 0xFFFF Error Range $0 < x < 32$ Resolution - 0.005 Value to send - $x \cdot 2^{11}$ Default - 5.5 (0x2C00)
Command	0x0D
Number of data bytes	2 (in response)
Data 0	MSB of proportional gain
Data 1	LSB of proportional gain

**14 SET TUNING ALGORITHM HIGH SPEED INTEGRAL GAIN**

Purpose	Set the gain level of the high speed integral control stage of the tuning PID loop
Message Type	Data write message
Notes	16bit unsigned fixed point (0-0xFFFFE) Range $0 < x < 32$ Resolution - 0.005 Value to send - $x \cdot 2^{11}$ Default - 0 (0x0000)
Command	0x0E
Number of data bytes	2
Data 0	MSB of integral gain
Data 1	LSB of integral gain

**15 GET TUNING ALGORITHM HIGH SPEED INTEGRAL GAIN**

Purpose	Get the gain level of the high speed integral control stage of the tuning PID loop
Message Type	Data request message
Notes	16bit unsigned fixed point (0-0xFFFFE) 0xFFFF Error Range $0 < x < 32$ Resolution - 0.005 Value to send - $x.2^{11}$ Default - 0 (0x0000)
Command	0x0F
Number of data bytes	2 (in response)
Data 0	MSB of integral gain
Data 1	LSB of integral gain

**16 SET TUNING ALGORITHM HIGH SPEED DIFFERENTIAL GAIN**

Purpose	Set the gain level of the high speed differential control stage of the tuning PID loop
Message Type	Data write message
Notes	16bit unsigned fixed point (0-0xFFFFE) Range $0 < x < 32$ Resolution - 0.005 Value to send - $x.2^{11}$ Default - 15 (0x7800)
Command	0x10
Number of data bytes	2
Data 0	MSB of differential gain
Data 1	LSB of differential gain

**17 GET TUNING ALGORITHM HIGH SPEED DIFFERENTIAL GAIN**

Purpose	Get the gain level of the high speed differential control stage of the tuning PID loop
Message Type	Data request message
Notes	16bit unsigned fixed point (0-0xFFFFE) 0xFFFF Error Range $0 < x < 32$ Resolution - 0.005 Value to send - $x.2^{11}$ Default - 15 (0x7800)
Command	0x11
Number of data bytes	2 (in response)
Data 0	MSB of differential gain
Data 1	LSB of differential gain

**18 SET FIRST ENDSTOP DISTANCE FROM MECHANICAL ENDSTOP**

Purpose	Inform the motor of the theoretical distance between the first mechanical endstop to the first software endstop.
Message Type	Data write message
Notes	16bit resolution (0-0xFFFFE) 1 degree ~ 182 motor steps Resolution ~ 0.005degrees
Command	0x12
Number of data bytes	2
Data 0	MSB of Hall effect position
Data 1	LSB of Hall effect position

**19 SET MECHANICAL MOTOR RANGE**

Purpose	Inform the motor of the theoretical distance between the first and second mechanical endstop
Message Type	Data write message
Notes	This message is used to set the operational range of the motor. The motor will use this information to set its software endstops to ensure that the mechanical endstops are not reached during normal operation. 16bit resolution (0-0xFFFFE) 1 degree ~ 182 motor steps Resolution ~ 0.005degrees
Command	0x13
Number of data bytes	2
Data 0	MSB of motor range
Data 1	LSB of motor range

**20 START CALIBRATION**

Purpose	Start the calibration process
Message Type	Data write message
Notes	No velocity or position move command will be processed by the motor until this message is sent and calibration has completed successfully. Messages 18 and 19 MUST be sent to the motor prior to this message being sent.
Command	0x14
Number of data bytes	0

**21 SET TUNING ALGORITHM LOW SPEED PROPORTIONAL GAIN**

Purpose	Set the gain level of the low speed proportional control stage of the tuning PID loop
Message Type	Data write message
Notes	16bit unsigned fixed point (0-0xFFFFE) Range $0 < x < 32$ Resolution - 0.005 Value to send - $x \cdot 2^{11}$ Default - 5.5 (0x2C00)
Command	0x15
Number of data bytes	2
Data 0	MSB of proportional gain
Data 1	LSB of proportional gain

**22 GET TUNING ALGORITHM LOW SPEED PROPORTIONAL GAIN**

Purpose	Get the gain level of the low speed proportional control stage of the tuning PID loop
Message Type	Data request message
Notes	16bit unsigned fixed point (0-0xFFFFE) 0xFFFF Error Range $0 < x < 32$ Resolution - 0.005 Value to send - $x \cdot 2^{11}$ Default - 5.5 (0x2C00)
Command	0x16
Number of data bytes	2 (in response)
Data 0	MSB of proportional gain
Data 1	LSB of proportional gain

**23 SET TUNING ALGORITHM LOW SPEED INTEGRAL GAIN**

Purpose	Set the gain level of the low speed integral control stage of the tuning PID loop
Message Type	Data write message
Notes	16bit unsigned fixed point (0-0xFFFFE) Range $0 < x < 32$ Resolution - 0.005 Value to send - $x \cdot 2^{11}$ Default 0.01 (0x0002)
Command	0x17
Number of data bytes	2
Data 0	MSB of integral gain
Data 1	LSB of integral gain

**24 GET TUNING ALGORITHM LOW SPEED INTEGRAL GAIN**

Purpose	Get the gain level of the low speed integral control stage of the tuning PID loop
Message Type	Data request message
Notes	16bit unsigned fixed point (0-0xFFFE) 0xFFFF Error Range $0 < x < 32$ Resolution - 0.005 Value to send - $x \cdot 2^{11}$ Default 0.01 (0x0002)
Command	0x18
Number of data bytes	2 (in response)
Data 0	MSB of integral gain
Data 1	LSB of integral gain

**25 SET TUNING ALGORITHM LOW SPEED DIFFERENTIAL GAIN**

Purpose	Set the gain level of the low speed differential control stage of the tuning PID loop
Message Type	Data write message
Notes	16bit unsigned fixed point (0-0xFFFE) Range $0 < x < 32$ Resolution - 0.005 Value to send - $x \cdot 2^{11}$ Default 20 (0xA000)
Command	0x19
Number of data bytes	2
Data 0	MSB of differential gain
Data 1	LSB of differential gain

**26 GET TUNING ALGORITHM LOW SPEED DIFFERENTIAL GAIN**

Purpose	Get the gain level of the low speed differential control stage of the tuning PID loop
Message Type	Data request message
Notes	16bit unsigned fixed point (0-0xFFFE) 0xFFFF Error Range $0 < x < 32$ Resolution - 0.005 Value to send - $x \cdot 2^{11}$ Default 20 (0xA000)
Command	0x1A
Number of data bytes	2 (in response)
Data 0	MSB of differential gain
Data 1	LSB of differential gain

**27 SET TUNING ALGORITHM LOW/HIGH SPEED TRANSITION**

Purpose	Set the speed at which the driving algorithm switches from the low speed gain values to the high speed gain values.
Message Type	Data write message
Notes	16bit resolution (0-0xE38) Min 0 0 motor counts/second (0 degrees/second) Default 0x5B 91motor counts/second (1 degree/second) Max 0xE38 3640 motor counts/second (40 degrees/second) 1 degree/second ~ 91 motor counts/second Resolution ~ 0.01degrees/second
Command	0x1B
Number of data bytes	2
Data 0	MSB of switch velocity
Data 1	LSB of switch velocity

**28 GET TUNING ALGORITHM LOW/HIGH SPEED TRANSITION**

Purpose	Get the gain level of the low speed differential control stage of the tuning PID loop
Message Type	Data request message
Notes	16bit resolution (0-0xE38) Min 0 0 motor counts/second (0 degrees/second) Default 0x5B 91motor counts/second (1 degree/second) Max 0xE38 3640 motor counts/second (40 degrees/second) 1 degree/second ~ 91 motor counts/second Resolution 0.01degrees/second 0xFFFF - Error
Command	0x1C
Number of data bytes	2 (in response)
Data 0	MSB of switch velocity
Data 1	LSB of switch velocity

**30 GET ENCODER POSITION**

Purpose	Return the current encoder reading relative to the endstop position
Message Type	Data request message
Command	0x1E
Number of data bytes	2 (in response)

**35 SAVE PID SETTINGS**

Purpose	Save the exiting PID control values to non-volatile FLASH sector
Message Type	Data write message
Notes	Motor will reset after the command, and new PID parameters will take effect after resetting.
Command	0x23
Number of data bytes	0

**36 RELOAD PID DEFAULTS**

Purpose	Restore default settings, including fast initialization parameter and PID parameters in non-volatile flash sector
Message Type	Data write message
Notes	1) Motor will reset after the command. 2) Slow initialization routines will be used after reset; 3) Once slow initialization finishes, the fast initialization parameter will be written to the flash sector, together with default PID values. 4) Motor will reset again with fast initialization parameter and default PID parameters
Command	0x24
Number of data bytes	0