TITAN INTELLIGENT BRUSHLESS MOTOR I2C PROTOCOL

A standard 3.3V, two wire, i²c interface is provided by the motor consisting of a clock, SCL, and data, SDA, line.

PHYSICAL LAYER

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.

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.

MESSAGE STRUCTURE

There are two types of messages, a data write message and a data request message.

1. DATA WRITE MESSAGE

This takes the following form

STX	Control Byte	ACK	Command Byte	ACK	Data Byte *0*	ACK *0*	Data Byte *n*	ACK *n*	ETX

SDA under master control
SDA under slave control

CONTROL BYTE

The control byte consists of the control code (0x05) 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 7	Control code bit 3: 0
Bit 6	Control code bit 2: 1
Bit 5	Control code bit 1: 0
Bit 4	Control code bit 0: 1
Bit 3	Address bit 2: 1/0
Bit 2	Address bit 1: 1/0
Bit 1	Address bit 0: 1/0
Bit 0	Write/Request bit: 0 (write)

COMMAND BYTE

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

DATA BYTES

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

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

SDA under master control
SDA under slave control

Control byte

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

Command Byte
An appropriate command byte must be sent to request data (see below)

DELIVER REQUESTED DATA MESSAGE

STX	Control Byte	ACK	Data Byte *0*	ACK *0*	Data Byte *n*	NACK *n*	ETX

SDA under master control
SDA under slave control

Control byte

Bit 7	Control code bit 3: 0
Bit 6	Control code bit 2: 1
Bit 5	Control code bit 1: 0
Bit 4	Control code bit 0: 1
Bit 3	Address bit 2: 1/0
Bit 2	Address bit 1: 1/0
Bit 1	Address bit 0: 1/0
Bit 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.

COMMAND SET

OVERVIEW

Command Number	Command Description	Command Type	Number of data bytes to transmit/receive
1	Reset Motor	Data Write	0
2	Calibration Complete	Data Request	1
3	Is Motor Moving	Data Request	1
4	Return Current Location	Data Request	2
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
12	Set Tuning Algorithm High Speed Proportional gain	Data Write	2
13	Get Tuning Algorithm High Speed Proportional gain	Data Request	2
14	Set Tuning Algorithm High Speed Integral gain	Data Write	2
15	Get Tuning Algorithm High Speed Integral gain	Data Request	2
16	Set Tuning Algorithm High Speed Differential gain	Data Write	2
17	Get Tuning Algorithm High Speed Differential gain	Data Request	2
18	Set First Mechanical Endstop to Hall Effect Distance	Data Write	2
19	Set Mechanical Motor Range	Data Write	2
20	Start Calibration	Data Write	0
21	Set Tuning Algorithm High Speed Proportional gain	Data Write	2
22	Get Tuning Algorithm High Speed Proportional gain	Data Request	2
23	Set Tuning Algorithm High Speed Integral gain	Data Write	2
24	Get Tuning Algorithm High Speed Integral gain	Data Request	2
25	Set Tuning Algorithm High Speed Differential gain	Data Write	2
26	Get Tuning Algorithm High Speed Differential gain	Data Request	2
27	Set Tuning Algorithm Low/High Speed Transition	Data Write	2
28	Get Tuning Algorithm Low/High Speed Transition	Data Request	2

COMMAND 1 – RESET

Purpose	Resets motor
Message Type	Data write message
Notes	Equivalent to power cycling the motor
Command	0x01
Number of Data Bytes	0
Data 0	N/A
Data 1	N/A
Data 2	N/A

COMMAND 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 in response	1
Data 0	0x00 – calibration in progress 0x01 – calibration completed successfully 0x02 – calibration failed Else – Error
Data 1	N/A

COMMAND 3 – IS MOTOR MOVING

Purpose	Returns movement status of motor
Message Type	Data request message
Notes	
Command	0x03
Number of Data Bytes in response	2
Data 0	8bit signed value -1: moving anticlockwise 0 : stopped 1: moving clockwise Else - Error
Data 1	N/A

COMMAND 4 – RETURN CURRENT LOCATION

Purpose	Returns the current motor location
Message Type	Data request message
Notes	16bit positional accuracy (0 – 0xFFFE) 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 in response	2
Data 0	MSB of position
Data 1	LSB of position

COMMAND 5 – GOTO ABSOLUTE LOCATION

Purpose	Moves the motor to an absolute position
Message Type	Data write message
Notes	16bit positional accuracy (0 – 0xFFFE) 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
Data 2	N/A

COMMAND 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
Data 2	N/A

COMMAND 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
Data 2	N/A

COMMAND 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^2 (720degrees/second^2) Default value 256 - 163840 motor counts/second^2 (900degrees/second^2) 1023 - 327670 motor counts/second^2 (3600degrees/second^2)
Command	0x08
Number of Data Bytes	2
Data 0	MSB of acceleration
Data 1	LSB of acceleration
Data 2	N/A

COMMAND 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 – 0xFFFE) 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

COMMAND 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

COMMAND 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^2 (720degrees/second^2) Default value 256 - 163840 motor counts/second^2 (900degrees/second^2) 1023 - 327670 motor counts/second^2 (3600degrees/second^2)
Command	0x0B
Number of Data Bytes in response	2
Data 0	MSB of acceleration
Data 1	LSB of acceleration

COMMAND 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 (0-0xFFFE) Range $0 < x < 32$ Resolution - 0.005 Value to send - $x.2^{11}$ Default - 5.5 (0x2C00)
Command	0x0C
Number of Data Bytes	2
Data 0	MSB of proportional gain
Data 1	LSB of proportional gain
Data 2	N/A

COMMAND 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 (0-0xFFFE) 0xFFFF – Error Range 0 < x < 32 Resolution - 0.005 Value to send - $x.2^{11}$ Default - 5.5 (0x2C00)
Command	0x0D
Number of Data Bytes in response	2
Data 0	MSB of proportional gain
Data 1	LSB of proportional gain

COMMAND 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 (0-0xFFFE) Range 0 < x < 32 Resolution - 0.005 Value to send - $x.2^{11}$ Default - 0 (0x0000)
Command	0x0E
Number of Data Bytes	2
Data 0	MSB of integral gain
Data 1	LSB of integral gain
Data 2	N/A

COMMAND 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 (0-0xFFFE) 0xFFFF – Error Range 0 < x < 32 Resolution - 0.005 Value to send - $x.2^{11}$ Default - 0 (0x0000)
Command	0x0F
Number of Data Bytes in response	2
Data 0	MSB of integral gain
Data 1	LSB of integral gain

COMMAND 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 (0-0xFFFE) 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
Data 2	N/A

COMMAND 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 (0-0xFFFE) 0xFFFF – Error Range 0 < x < 32 Resolution - 0.005 Value to send - $x.2^{11}$ Default - 15 (0x7800)
Command	0x11
Number of Data Bytes in response	2
Data 0	MSB of differential gain
Data 1	LSB of differential gain

COMMAND 18 – SET FIRST MECHANICAL ENDSTOP TO HALL EFFECT DISTANCE

Purpose	Inform the motor of the theoretical distance between the first mechanical endstop to the position of the Hall effect.
Message Type	Data write message
Notes	Once the start calibration message is sent (msg 20) the motor will travel anticlockwise until it hits the first mechanical endstop. It will then travel clockwise until it passes the Hall effect. The value sent in this message will be used to check that the Hall effect feedback occurs at the appropriate number of motor steps away from the first endstop. 16bit resolution (0-0xFFFE) 1 degree ~ 182 motor steps Resolution ~ 0.005degrees Acceptable position tolerance ±1degree
Command	0x12
Number of Data Bytes	2
Data 0	MSB of Hall effect position
Data 1	LSB of Hall effect position
Data 2	N/A

COMMAND 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 used this information to set its software endstops to ensure that the mechanical endstops are not reached during normal operation. 16bit resolution (0-0xFFFE) 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
Data 2	N/A

COMMAND 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
Data 0	N/A
Data 1	N/A
Data 2	N/A

COMMAND 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 (0-0xFFFE) Range $0 < x < 32$ Resolution - 0.005 Value to send - $x.2^{11}$ Default - 5.5 (0x2C00)
Command	0x15
Number of Data Bytes	2
Data 0	MSB of proportional gain
Data 1	LSB of proportional gain
Data 2	N/A

COMMAND 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 (0-0xFFFE) 0xFFFF – Error Range 0 < x < 32 Resolution - 0.005 Value to send - $x.2^{11}$ Default - 5.5 (0x2C00)
Command	0x16
Number of Data Bytes in response	2
Data 0	MSB of proportional gain
Data 1	LSB of proportional gain

COMMAND 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 (0-0xFFFE) Range 0 < x < 32 Resolution - 0.005 Value to send - $x.2^{11}$ Default – 0.01 (0x0002)
Command	0x17
Number of Data Bytes	2
Data 0	MSB of integral gain
Data 1	LSB of integral gain
Data 2	N/A

COMMAND 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 - $x.2^{11}$ Default – 0.01 (0x0002)
Command	0x18
Number of Data Bytes in response	2
Data 0	MSB of integral gain
Data 1	LSB of integral gain

COMMAND 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 - $x.2^{11}$ Default – 20 (0xA000)
Command	0x19
Number of Data Bytes	2
Data 0	MSB of differential gain
Data 1	LSB of differential gain
Data 2	N/A

COMMAND 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 - $x.2^{11}$ Default – 20 (0xA000)
Command	0x1A
Number of Data Bytes in response	2
Data 0	MSB of differential gain
Data 1	LSB of differential gain

COMMAND 27 –SET TUNING ALGORITHM LOW/HIGH SPEED TRANSISTION

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
Data 2	N/A

COMMAND 28 – GET TUNING ALGORITHM LOW/HIGH SPEED TRANSISTION

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 in response	2
Data 0	MSB of switch velocity
Data 1	LSB of switch velocity