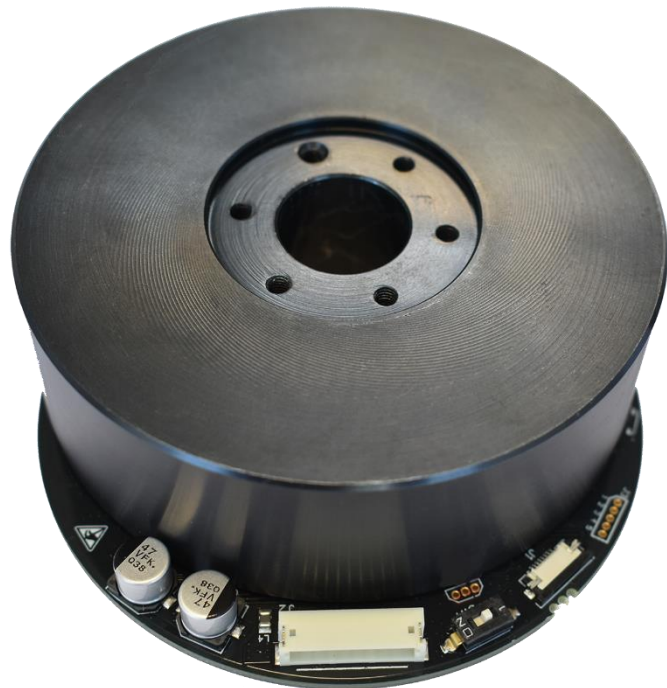




Technical Manual

Miranda BLDC Integrated Servo Motor

Product Code: OVU20012 Continuous Rotation



The information in this document is subject to change without notice and should not be construed as a commitment by Overview Ltd. Overview Ltd assumes no responsibility for any errors that may appear in this document.

In no event shall Overview be liable for incidental or consequential damages arising from use of this document or the hardware and software described in this document.

This document and parts thereof must not be reproduced or copied without Overview's written permission, and contents thereof must not be imparted to a third party nor be used for any unauthorised purpose.

Miranda™ and Servotorq™ are trademarks of Overview Ltd.

Overview Ltd.

Overview House, Kingswey Business Park
Forsyth Road, Woking, GU21 5SA, United Kingdom
Tel: +44 (0) 20 8875 0984
Email: info@overview.co.uk
www.overview.co.uk

All rights reserved.

Table of Contents

1. Introduction	4
2. Product Features.....	4
3. Precautions and Warnings	4
4. Product Specifications.....	5
4.1. Motor Electrical Data	5
5. Product Breakdown	7
5.1. Product Drawing – Miranda Continuous Rotation Motor	7
6. Installation / Connection	8
6.1. Printed Circuit Board Assembly (PCBA)	8
6.2. Mechanical	8
7. Software Set-up	9
7.1. Physical Layer	9
7.2. Communications Layer.....	9
7.3. Message Structure	9
7.3.1. Data Write Message.....	9
7.3.2. Data Request Message.....	10
7.4. Startup.....	11
7.5. Commands	12
7.5.1. Summary	12
7.5.2. Detailed Description.....	13
7.6. Recommended Configurations	34
7.7. Firmware Update Mode.....	36
7.7.1. Firmware Image File.....	36
7.7.2. Programming Procedure.....	36
7.7.3. IFP Commands Summary	38
7.7.4. IFP Commands in Detail	38

1. Introduction

The Miranda BLDC Integrated Servo Motors have been designed to be integrated into a number of environments that require a combination of high positional accuracy as well as smooth and silent motion. These include video conferencing, broadcast and CCTV PTZ applications. The Miranda platform creates a standalone intelligent motor with integral low-level motor control and integrated optical feedback. It is robust to external dynamic disturbances, especially impulses and vibrations.

2. Product Features

- Wide dynamic speed-range from $0.05^{\circ}s^{-1}$ to $720^{\circ}s^{-1}$
- Rapid and precise positional control to $\pm 0.007^{\circ}$ repeatability
- Near silent operation
- Integral Digital Signal Processor (DSP) based motor controller
- High level control commands via a simple I²C bus
- PCBA connections conveniently arranged for use in a pan/tilt camera product
- All control and drive components are contained within the 88mm by 78mm PCB
- Hollow shaft for slip ring integration
- User configurable PID gains and acceleration to allow tuning for the attached load
- Ability to in-field upgrade firmware

3. Precautions and Warnings



Please study this technical manual thoroughly.

- To prevent the risk of hazards such as fire or shock, do not expose this product to water.
- To prevent the risk of damage, fire or burns ensure that Miranda is not used at high input powers for extended durations that could cause excessive temperatures in the servo motor whether or not Miranda is installed in the end user product. Ensure all items near the Miranda servo drive are rated for the maximum possible temperature.
- Do not drop or shake the carton severely. The transit box cannot give complete protection.
- Only connect to the specified voltage.
- Only insert the mating connector when power and signals are off.
- Ensure appropriate ESD preventative actions are taken when handling this product.
- Do not install the product in wet or high humidity areas without adequate protection.
- Do not disassemble the product. This will void the given warranty.
- This product is not certified for the following environments:
 - Explosive atmospheric conditions
 - Life critical systems

4. Product Specifications

4.1. Motor Electrical Data

Electrical Properties	
Specification	Value
Operating voltage	22V to 26V DC
Current at max torque	1A
Initialisation current	1A
Idle, no load, current	50mA
Max. input voltage ripple	250mV@100kHz
Resolution	0.0055°
Max Speed	720°/s
Min Speed	0.05°/s

Table 1: Motor electrical data at 24V unless otherwise specified.

Mechanical Properties	
Specification	Value
Weight	570g
Rotor moment of inertia	2350gcm ²
Locked Rotor/Stall/Holding Torque (with OTP switched on, note ²)	1Nm
Moving Torque	0.85Nm @ 360°/s
Parallelism rotor to stator	< 0.1 mm
Maximum radial load	1900N
Maximum axial load (note ¹)	475N
Range	Configurable between 0°-360° including continuous rotation

Table 2: Motor mechanical power and mechanical properties.

Environmental Conditions	
Specification	Value
Temperature rated (standard operation with OTP switched on, note ²)	-40°C to +75°C
Humidity range (standard operation)	5% to 90% RH at 40°C
Storage Conditions	-40°C to +75°C
Storage Humidity (%RH non-condensing)	5% - 85%

Table 3: Motor environmental capabilities.

¹ When either the axial or radial load is zero, the other axis can be loaded more.

² The motor is not rated to draw 24W continuously, maximum continuous power is dependent on installation and duty cycle. It is recommended that Over Temperature Protection (OTP) is on to protect the drive if the motor is used at high temperatures or high powers.

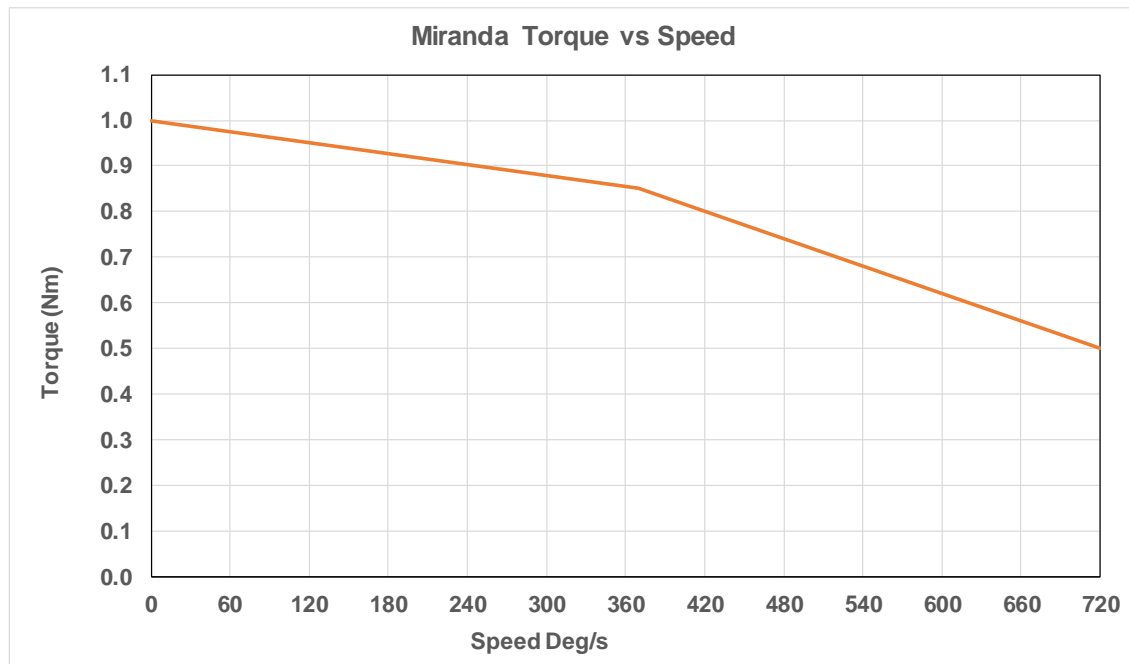


Figure 1: Torque vs speed curve of the Miranda motor at 24V DC input

Note that the speed is limited by software to 720 deg/s

5. Product Breakdown

5.1. Product Drawing – Miranda Continuous Rotation Motor

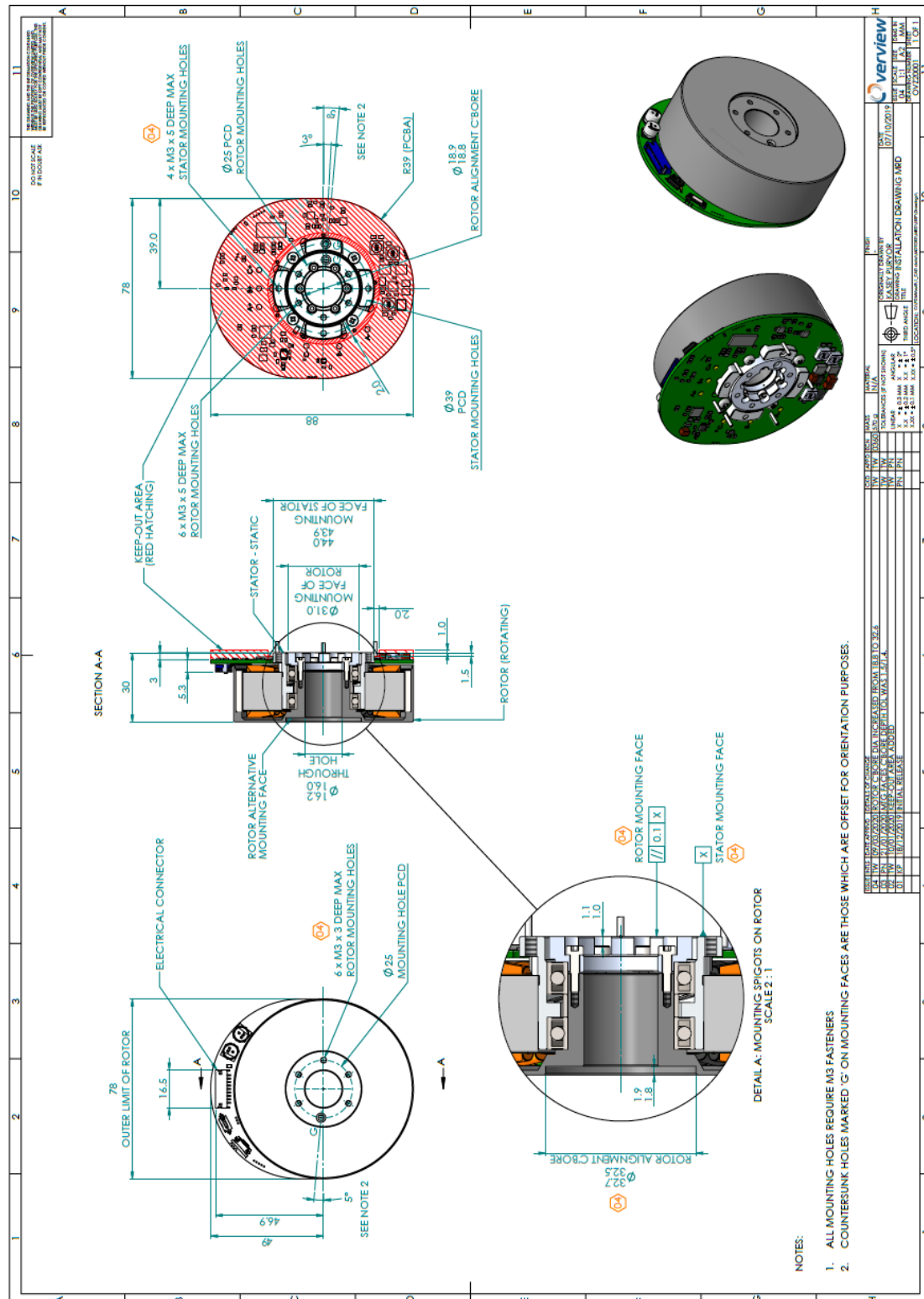


Figure 2: Miranda Continuous Rotation Motor

6. Installation / Connection

6.1. Printed Circuit Board Assembly (PCBA)

Figure 3 below highlights the position of the J2 connector and a switch SW1 on the PCBA for interfacing with external systems.

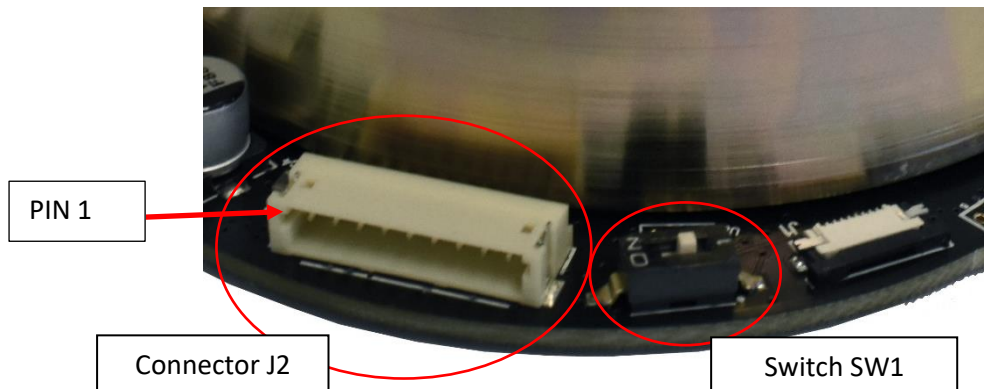


Figure 3: Drive View with J2 & Switch SW1 highlighted

Connector J2 JST part no S9B-ZR-SM4A-TF(LF)(SN)	
Pin	Signal
1	24V Power (1000mA Max Total)
2	24V Power (1000mA Max Total)
3	Power Ground (0V)
4	Power Ground (0V)
5	Address select pin 0x29 (Tilt) = 0V 0x28 (Pan) = Open (not connected)
6	Signal Ground (0V)
7	SCL (I2C Clock Signal 3V3)
8	Signal Ground (0V)
9	SDA (I2C Data Signal 3V3)

Table 4: J4 Connector details

SW1 - Switch		
I ² C Address		
Position	Description	Address
ON	Tilt	0x29
OFF (pin 5 open)	Pan	0x28
OFF (pin 5 grounded)	Tilt (externally selected)	0x29

Table 5: SW1 Switch details

6.2. Mechanical

The two mounting faces of the rotor have six M3 threaded holes provided for mechanical interfacing with the motor. The stator has four M3 threaded holes on the PCB side provided for mechanical interfacing with the motor. The maximum thread depth of the M3 screws in the rotor to avoid damage to the mechanical parts shall be 3mm and in the stator shall be 5mm. Appropriate thread engagement calculations should be carried out to prevent stripping of threads during installation.

7. Software Set-up

This section describes the use of the I2C interface to the Servotorq BLDC Integrated Servo Motor family. The message protocol for devices on the bus is given, followed by a detailed description of the available motor commands. The commands listed in this document apply to firmware versions 9.8.137 and upwards.

7.1. Physical Layer

A standard two wire I²C 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 the idle voltage. Miranda BLDC motors use an I²C high voltage of 3.3V. The SCL signal is always generated by the master and the BLDC motor(s) will always be slaves on the bus. SDA can be controlled by either the master or the slave being communicated to.

7.2. 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.

7.3. 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

7.3.1. 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 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: 0
2	Address bit 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 7.5.

Data Bytes.

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

7.3.2. Data Request Message

To request data from the 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: 0
2	Address bit 1: 0
1	Address bit 0: 1/0
0	Write/Request bit: 0 (write)

Command Byte. See section 7.5 for a detailed description of request commands.

Deliver Requested Data Message

STX	Control Byte	ACK	Data Byte 0	ACK 0	Data Byte n	NACK n	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: 0
2	Address bit 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.

7.4. Startup

To allow the motor firmware to be upgraded remotely, the device can be run in In-Field Programming (IFP) mode. This mode will be activated if the "\HOLD LAUNCH" command (0xF0) is received in the first 500ms after power up. Figure 1 shows an outline of timings of the startup procedure. More information about the IFP mode can be found in section 7.7.

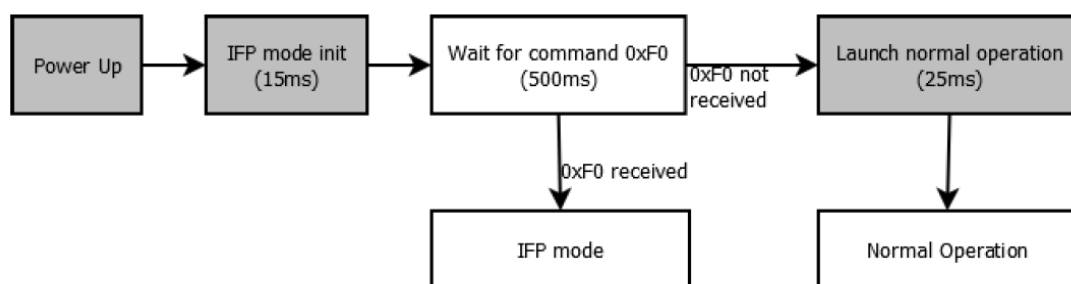


Figure 1: BLDC motor Startup procedure.

Grey boxes indicate when the motor may not be guaranteed to respond to I²C messages.

7.5. Commands

7.5.1. Summary

Code	Description	Message Type	# data bytes
0x01	Reset	Data write	0
0x02	Calibration complete	Data read	1
0x03	Is motor moving	Data read	1
0x04	Return current location	Data read	2
0x05	Goto absolute location	Data write	2
0x06	Goto relative location	Data write	2
0x07	Travel at velocity	Data write	2
0x08	Set max acceleration	Data write	2
0x09	Goto absolute location in set time	Data write	3
0x0A	Goto relative location in set time	Data write	3
0x0B	Get max acceleration	Data read	2
0x0C	Set tuning algorithm proportional gain	Data write	2
0x0D	Get tuning algorithm proportional gain	Data read	2
0x0E	Set tuning algorithm integral gain	Data write	2
0x0F	Get tuning algorithm integral gain	Data read	2
0x10	Set tuning algorithm differential gain	Data write	2
0x11	Get tuning algorithm differential gain	Data read	2
0x12	Set first endstop distance from mechanical endstop	Data write	2
0x13	Set mechanical motor range	Data write	2
0x15	Set Over Temperature Protection max temp	Data write	2
0x16	Get Over Temperature Protection max temp	Data read	2
0x19	Set continuous	Data write	2
0x1A	Get continuous	Data read	2
0x1B	Get firmware version	Data read	4
0x1C	Wake up	Data write	0
0x1D	Set sleep on power up mode	Data write	1
0x1E	Get encoder position	Data read	2
0x23	Save settings to flash	Data write	0
0x24	Reload factory defaults	Data write	0
0x2F	Get sleep on power up mode	Data read	1
0x30	Is sleeping	Data read	1
0x40	Goto relative position 360	Data write	3
0x41	Goto relative position at speed	Data write	4
0x42	Goto absolute position at speed	Data write	4
0x43	Set use Low Pass Filter	Data write	1
0x46	Set tuning algorithm Kc gain	Data write	2
0x47	Get tuning algorithm Kc gain	Data read	2
0x4A	Get first endstop distance	Data read	2
0x4B	Get mechanical motor range	Data read	2
0x4C	Set Ud filter factor	Data write	1
0x4D	Get Ud filter factor	Data read	1
0x4E	Set use hall sensor	Data write	1
0x4F	Get use hall sensor	Data read	1
0x50	Get use Low Pass Filter	Data read	1
0x51	Set current controller set point	Data write	2
0x52	Get current controller set point	Data read	2
0x53	Set use Turbo mode	Data write	1
0x54	Get use Turbo mode	Data read	1
0x56	Set init method	Data write	1

Code	Description	Message Type	# data bytes
0x57	Get init method	Data read	1
0x58	Set position filter factor	Data write	4
0x59	Get position filter factor	Data read	4
0x5E	Goto absolute pos in time ms	Data write	4
0x5F	Goto relative pos in time ms	Data write	5
0x75	Set phase alignment current limit	Data write	2
0x76	Get phase alignment current limit	Data read	2
0x83	Set Dynamic Trajectory mode	Data write	1
0x84	Get Dynamic Trajectory mode	Data read	1
0xFE	Get program state	Data read	1

7.5.2.Detailed Description

0x01 RESET

Purpose	Resets motor
Message Type	Data write message
Notes	Equivalent to power cycling the motor, wait 25ms before issuing new I ² C commands
Command	0x01
Number of data bytes	0

0x02 CALIBRATION COMPLETE

Purpose	Checks if motor is calibrated and ready for normal operation
Message Type	Data read message
Notes	Velocity and position moves sent before calibration is complete will be ignored.
Command	0x02
Number of data bytes	1
Data 0	0x00 - calibration in progress 0x01 - calibration completed successfully 0x02 - calibration failed

0x03 IS MOTOR MOVING

Purpose	Returns movement status of motor
Message Type	Data read message
Command	0x03
Number of data bytes	1
Data 0	8bit signed value 0xff: moving anticlockwise 0x00: stopped 0x01: moving clockwise Else - Error

0x04 RETURN CURRENT LOCATION

Purpose	Returns the current target rotor location
Message Type	Data read message
Notes	Range 0 - 0xFFFF in motor counts (65536 counts per revolution) The position returned is the clockwise distance from the first mechanical endstop in motor counts. This is the controller setpoint and will differ from true location if the rotor is externally disturbed. See also 0x43. 1 degree ~ 182 motor counts Resolution ~ 0.005 degrees
Command	0x04
Number of data bytes	2
Data 0	MSB of position
Data 1	LSB of position

0x05 GOTO ABSOLUTE LOCATION

Purpose	Moves to an absolute position
Message Type	Data write message
Notes	Range 0 - 0xFFFF in motor counts (65536 counts per revolution). The position specified is the clockwise distance from the first mechanical endstop in motor counts. 1 degree ~ 182 motor counts Resolution ~ 0.005 degrees
Command	0x05
Number of data bytes	2
Data 0	MSB of position
Data 1	LSB of position

0x06 GOTO RELATIVE LOCATION

Purpose	Moves 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 ~ 0.005 degrees NB. The move will stop at the software endstop if the supplied range is too large to execute. The maximum relative move is ± 180 degrees. See also 0x40.
Command	0x06
Number of data bytes	2
Data 0	MSB of position
Data 1	LSB of position

0x07 TRAVEL AT VELOCITY

Purpose	Moves 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 ~ 0.05degrees/second Min velocity 4 counts/second
Command	0x07
Number of data bytes	2
Data 0	MSB of speed
Data 1	LSB of speed

0x08 SET MAX ACCELERATION

Purpose	Sets the maximum acceleration
Message Type	Data write message
Notes	acceleration range 1 (minimum) -256(max) default value 64 == 2746 degrees/s ² Maximum recommended value depends on load.
Command	0x08
Number of data bytes	2
Data 0	MSB of acceleration
Data 1	LSB of acceleration

0x09 GOTO ABSOLUTE LOCATION IN SET TIME

Purpose	Moves to an absolute position in a given time
Message Type	Data write message
Notes	Range 0 - 0xFFFF in motor counts (65536 counts per revolution) 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

0x0A GOTO RELATIVE LOCATION IN SET TIME

Purpose	Moves 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 degrees.
Command	0x0A
Number of data bytes	3
Data 0	MSB of position
Data 1	LSB of position
Data 2	Time

0x0B GET MAX ACCELERATION

Purpose	Get the current maximum acceleration setting
Message Type	Data read message
Notes	
Command	0x0B
Number of data bytes	2
Data 0	MSB of acceleration
Data 1	LSB of acceleration

0x0C SET TUNING ALGORITHM PROPORTIONAL GAIN

Purpose	Set the gain level of the 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 - $x:2^{11}$
Command	0x0C
Number of data bytes	2
Data 0	MSB of proportional gain
Data 1	LSB of proportional gain

0x0D GET TUNING ALGORITHM PROPORTIONAL GAIN

Purpose	Get the gain level of the proportional control stage of the tuning PID loop
Message Type	Data read message
Notes	16bit unsigned fixed point (0-0xFFFF) 0xFFFF - Error Range $0 < x < 32$ Resolution - 0.005 Value to send - $x:2^{11}$
Command	0x0D
Number of data bytes	2
Data 0	MSB of proportional gain
Data 1	LSB of proportional gain

0x0E SET TUNING ALGORITHM INTEGRAL GAIN

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

0x0F GET TUNING ALGORITHM INTEGRAL GAIN

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

0x10 SET TUNING ALGORITHM DIFFERENTIAL GAIN

Purpose	Set the gain level of the differential control stage of the tuning PID loop
Message Type	Data write message
Notes	16bit unsigned fixed point (0-0xFFFF) Range $0 < x < 128$ Resolution - 0.005 Value to send - $x:2^9$
Command	0x10
Number of data bytes	2
Data 0	MSB of differential gain
Data 1	LSB of differential gain

0x11 GET TUNING ALGORITHM DIFFERENTIAL GAIN

Purpose	Get the gain level of the differential control stage of the tuning PID loop
Message Type	Data read message
Notes	16bit unsigned fixed point (0-0xFFFF) Range $0 < x < 128$ Resolution - 0.005 Value to send - $x:2^9$
Command	0x11
Number of data bytes	2
Data 0	MSB of differential gain
Data 1	LSB of differential gain

0x12 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	Range 0 - 0xFFFF in motor counts (65536 counts per revolution) 1 degree ~ 182 motor steps Resolution ~ 0.005degrees Command 0x23 must be subsequently called if this value is to be saved to non-volatile memory.
Command	0x12
Number of data bytes	2
Data 0	MSB of distance
Data 1	LSB of distance

0x13 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	<p>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.</p> <p>Range 0 - 0xFFFF in motor counts (65536 counts per revolution) 1 degree ~ 182 motor steps Resolution ~ 0.005degrees</p> <p>If in hardware endstop mode (command 0x19 = 0x00), first endstop pos + range can't be greater than 360 degrees. This restriction does not apply in "continuous with sw endstops" mode (command 0x19 = 0x02).</p> <p>Command 0x23 must be subsequently called if this value is to be saved to non-volatile memory.</p>
Command	0x13
Number of data bytes	2
Data 0	MSB of motor range
Data 1	LSB of motor range

0x15 SET OVER TEMPERATURE PROTECTION MAX TEMP

Purpose	Set the temperature trigger level for OTP to send the motor to sleep
Message Type	Data write message
Notes	Temperature in integer degrees C. e.g. 0x0057 = 87degC
Command	0x15
Number of data bytes	2
Data 0	MSB of max temp
Data 1	LSB of max temp

0x16 GET OVER TEMPERATURE PROTECTION MAX TEMP

Purpose	Get the temperature trigger level for OTP to send the motor to sleep
Message Type	Data read message
Notes	Temperature in integer degrees C. e.g. 0x0057 = 87degC
Command	0x16
Number of data bytes	2
Data 0	MSB of max temp
Data 1	LSB of max temp

0x19 SET CONTINUOUS

Purpose	Select whether pan motor is to perform continuous (slip ring) or limited range motion
Message Type	Data write message
Notes	16bit unsigned int Send 0x0001 for continuous motion, 0x0000 for limited motion. From firmware version 5900 onwards, send 0x0002 for continuous rotation with software endstops. When changing between limited motion and any other mode, if not in sleep mode, saves settings to flash (0x23) automatically and performs a software reset, forcing re-calibration. Wait 2000ms before issuing new I ² C commands. If in sleep mode, 0x23 must be called manually to save the change to non-volatile memory. Invalid input values for data1 are changed to 0x00. See also 0x30 (IS SLEEPING).
Command	0x19
Number of data bytes	2
Data 0	0x00
Data 1	0x00 or 0x01, bool on/off

0x1A GET CONTINUOUS

Purpose	Get boolean value indicating whether continuous or limited motion is default
Message Type	Data read message
Notes	16bit unsigned int 0x0000 limited range 0x0001 unlimited range (slipring) For firmware revision 5900 onwards: 0x0002 unlimited range but with software endstops
Command	0x1A
Number of data bytes	2

0x1B GET FIRMWARE VERSION

Purpose	Read firmware version
Message Type	Type Data read message
Notes	Data 0: (u8) ver major Data 1: (u8) ver middle Data 2-3: (u16) ver minor
Command	0x1B
Number of data bytes	4

0x1C WAKE UP

Purpose	Wake motor from low power state
Message Type	Data write message
Command	0x1C
Number of data bytes	0

0x1D SET SLEEP ON POWER UP MODE

Purpose	Set motor to sleep on power up, or begin calibration immediately
Message Type	Data write message
Notes	<p>Data 0: 0 - calibrate immediately, 1 - wait for WAKE UP command.</p> <p>If not sleeping, saves settings to ash (0x23) automatically and performs a software reset. Wait 2000ms before issuing new I²C commands.</p> <p>If sleeping, 0x23 must be called to save the new setting to non-volatile memory. See also 0x30 (IS SLEEPING).</p> <p>All non-zero input values for data 0 are treated as 0x1.</p>
Command	0x1D
Number of data bytes	1

0x1E GET ENCODER POSITION

Purpose	Returns the current encoder position
Message Type	Data read message
Notes	<p>Range 0 - 0xFFFF in motor counts (65536 counts per revolution)</p> <p>The position returned is the clockwise distance from the first mechanical endstop in motor counts. This is the actual position of the motor and will differ from the controller setpoint if the motor is disturbed.</p> <p>1 degree ~ 182 motor counts</p> <p>Resolution ~ 0.005 degrees</p>
Command	0x1e
Number of data bytes	2
Data 0	MSB of position
Data 1	LSB of position

0x23 SAVE SETTINGS TO FLASH

Purpose	Save the exiting PID control values, endstop and sleep mode settings to non-volatile FLASH sector
Message Type	Data write message
Notes	Motor will reset after the command, wait 2000ms before issuing new I ² C commands.
Command	0x23
Number of data bytes	0

0x24 RELOAD FACTORY DEFAULTS

Purpose	Restore default settings in non-volatile ash sector, including PID parameters, endstop range and presence, and sleep mode settings
Message Type	Data write message
Notes	Motor will reset after the command, wait 2000ms before issuing new I ² C commands.
Command	0x24
Number of data bytes	0

0x2F GET SLEEP ON POWER UP MODE

Purpose	Get whether motor sleeps on power up, or begins calibration immediately based on value currently stored in flash
Message Type	Data read message
Notes	Data 0: 0 - calibrates immediately, 1 - waits for WAKE UP command
Command	0x2F
Number of data bytes	1

0x30 IS SLEEPING

Purpose	Get current sleep state (sleeping/awake)
Message Type	Data read message
Notes	Data 0: 0 - in normal operation 1 - low power mode From firmware revision 6058: 2 - calibrated low power mode
Command	0x30
Number of data bytes	1

0x40 GOTO RELATIVE POSITION 360

Purpose	Move to relative position with input range -360 to +360 degrees
Message Type	Data write message
Notes	Range 0 - 0xFFFF in motor counts (65536 counts per revolution)
Command	0x40
Number of data bytes	3
Data 0	Sign bit: 0x00 clockwise, 0x01 anticlockwise
Data 1	MSB of move
Data 2	LSB of move

0x41 GOTO RELATIVE POSITION AT SPEED

Purpose	Move to relative position at maximum speed specified
Message Type	Data write message
Notes	<p>Range 0 - 0xFFFF in motor counts (65536 counts per revolution)</p> <p>Direction set by velocity.</p> <p>Velocity: signed int16</p> <p>0xA001 - max anticlockwise (540 degrees/second)</p> <p>0 - stopped</p> <p>0x5FFF - max clockwise (540 degrees/second)</p> <p>Velocities above 0x5FFF or below 0xA001 will be limited to 540 degrees/s</p> <p>1 count/second ~ 0.022 degrees/second</p> <p>Min velocity 0.1 degrees/second (5 counts/second).</p> <p>If the requested move is beyond an endstop, move to the nearest endstop in the specified direction.</p>
Command	0x41
Number of data bytes	4
Data 0	MSB of move
Data 1	LSB of move
Data 2	MSB of velocity
Data 3	LSB of velocity

0x42 GOTO ABSOLUTE POSITION AT SPEED

Purpose	Move to absolute position at maximum speed specified
Message Type	Data write message
Notes	<p>Move: Range 0 - 0xFFFF in motor counts (65536 counts per revolution)</p> <p>Velocity: signed int16</p> <p>0xA001 - max anticlockwise (540 degrees/second)</p> <p>0 - stopped</p> <p>0x5FFF - max clockwise (540 degrees/second)</p> <p>Velocities above 0x5FFF or below 0xA001 will be limited to 540 degrees/s. 1 count/second ~ 0.022 degrees/second</p> <p>Min velocity 0.1 degrees/second (5 counts/second).</p> <p>When in continuous rotation mode, the velocity sign determines the move direction. When in endstop mode, the velocity sign is ignored, and the direction required to successfully reach the destination is automatically chosen.</p>
Command	0x42
Number of data bytes	4
Data 0	MSB of move
Data 1	LSB of move
Data 2	MSB of velocity
Data 3	LSB of velocity

0x43 SET USE LOW PASS FILTER

Purpose	Set whether the motor applies a low pass filter to incoming encoder readings
Message Type	Data write message
Notes	<p>The LPF reduces noise in position readings.</p> <p>Turning it on/off can affect controller tuning parameters.</p>
Command	0x43
Number of data bytes	1
Data 0	<p>0x00 - disabled</p> <p>0x01 – enabled</p>

0x46 SET TUNING ALGORITHM KC GAIN

Purpose	Set the gain level of the integral saturation 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 - $x:2^{11}$
Command	0x46
Number of data bytes	2
Data 0	MSB of proportional gain
Data 1	LSB of proportional gain

0x47 GET TUNING ALGORITHM KC GAIN

Purpose	Get the gain level of the integral saturation stage of the tuning PID loop
Message Type	Data read message
Notes	16bit unsigned fixed point (0-0xFFFFE) 0xFFFF - Error Range $0 < x < 32$ Resolution - 0.005 Value to send - $x:2^{11}$
Command	0x47
Number of data bytes	2
Data 0	MSB of proportional gain
Data 1	LSB of proportional gain

0x4A GET FIRST ENDSTOP DISTANCE

Purpose	Read current setting for distance between mechanical endstop and first software endstop
Message Type	Data read message
Notes	See 0x12 for scaling
Command	0x4A
Number of data bytes	2
Data 0	MSB of distance
Data 1	LSB of distance

0x4B GET MECHANICAL MOTOR RANGE

Purpose	Read current setting for distance between first and second software end-stops
Message Type	Data read message
Notes	See 0x13 for scaling
Command	0x4B
Number of data bytes	2
Data 0	MSB of distance
Data 1	LSB of distance

0x4C SET UD FILTER FACTOR

Purpose	Adjust filter for controller derivative term
Message Type	Data write message
Notes	Sets the bias towards existing reading as a power of 2. Higher setting more aggressively filters new data. Firmware default of 0x01.
Command	0x4C
Number of data bytes	1
Data 0	Power of 2 to use in filter

0x4D GET UD FILTER FACTOR

Purpose	Get current Ud filter power of 2 setting
Message Type	Data read message
Notes	See 0x4C
Command	0x4D
Number of data bytes	1
Data 0	Power of 2 currently used in filter

0x4E SET USE HALL SENSOR

Purpose	Set whether the motor uses the hall sensor during the homing routine
Message Type	Data write message
Notes	If disabled, when in a system with endstops present, then after searching for the first mechanical endstop the motor moves to the first software end-stop position. If enabled, homing always finishes with the motor position determined by the hall effect sensor.
Command	0x4E
Number of data bytes	1
Data 0	0x00 - disabled 0x01 – enabled

0x4F GET USE HALL SENSOR

Purpose	Get whether the motor uses the hall sensor during the homing routine
Message Type	Data read message
Notes	See 0x4E
Command	0x4F
Number of data bytes	1
Data 0	0x00 - disabled 0x01 - enabled

0x50 GET USE LOW PASS FILTER

Purpose	Get whether the motor applies a low pass filter to incoming encoder readings
Message Type	Data read message
Notes	See 0x43
Command	0x50
Number of data bytes	1
Data 0	0x00 - disabled 0x01 - enabled

0x51 SET CURRENT CONTROLLER SET POINT

Purpose	Set the 12-bit target for the current controller
Message Type	Data write message
Notes	0x0700 is 1A 0x0200 is 0.16A
Command	0x51
Number of data bytes	2
Data 0	MSB of set point
Data 1	LSB of set point

0x52 GET CURRENT CONTROLLER SET POINT

Purpose	Get the 12-bit target for the current controller
Message Type	Data read message
Notes	See 0x51
Command	0x52
Number of data bytes	2
Data 0	MSB of set point
Data 1	LSB of set point

0x53 SET USE TURBO MODE

Purpose	Set whether the motor uses turbo mode during velocity and position-at-speed moves
Message Type	Data write message
Notes	If enabled, velocity input with command 0x07 is doubled, giving max velocity of 720 deg/s. In addition, the max velocity cap of 540 deg/s for goto-pos-at-speed is changed to 720 deg/s.
Command	0x53
Number of data bytes	1
Data 0	0x00 - disabled 0x01 - enabled

0x54 GET USE TURBO MODE

Purpose	Get whether the motor uses turbo mode during velocity and position-at-speed moves
Message Type	Data read message
Notes	See 0x53.
Command	0x54
Number of data bytes	1
Data 0	0x00 - disabled 0x01 - enabled

0x56 SET INIT METHOD

Purpose	Set phase alignment method used on startup
Message Type	Data write message
Notes	Call save settings to commit to NVM
Command	0x56
Number of data bytes	1
Data 0	0x00 - standard approach with potential for large moves 0x01 - phase interrogation approach, to reduce the maximum move size

0x57 GET INIT METHOD

Purpose	Get whether the motor uses turbo mode during velocity and position-at-speed moves
Message Type	Data read message
Notes	See 0x56.
Command	0x57
Number of data bytes	1
Data 0	0x00 - standard approach with potential for large moves 0x01 - phase interrogation approach, to reduce the maximum move size

0x58 SET POSITION FILTER FACTOR

Purpose	Set the IQ24-format filter used to smooth trajectory demands
Message Type	Data write message
Notes	0x01000000 is equivalent to a filter factor of 1.0 Valid range of 0x00000001 – 0x00ffffff. Higher means more aggressive filtering on trajectories. Default of 0x00FD27D2 = 0.988888888
Command	0x58
Number of data bytes	4
Data 0	Filter MSB
Data 1	...
Data 2	...
Data 3	Filter LSB

0x59 GET POSITION FILTER FACTOR

Purpose	Get the IQ24-format filter used to smooth trajectory demands
Message Type	Type Data read message
Notes	See 0x58
Command	0x59
Number of data bytes	4

0x5E GOTO ABSOLUTE POS IN TIME MS

Purpose	0.01 second resolution on timed absolute moves
Message Type	Data write message
Notes	Range 0 - 0xFFFF in motor counts (65536 counts per revolution) Move time in multiples of 0.01s 0 - 0xFFFF, e.g. 10.00 seconds == 0x03e8 If time/distance > 2 ⁷ , where distance is in motor counts and time in ms, a smooth trajectory cannot be guaranteed due to minimum velocity limitations.
Command	0x5E
Number of data bytes	4
Data 0	Move MSB
Data 1	Move LSB
Data 2	Time MSB
Data 3	Time LSB

0x5F GOTO RELATIVE POS IN TIME MS

Purpose	0.01 second resolution on timed relative moves
Message Type	Data write message
Notes	<p>Range 0 - 0xFFFF in motor counts (65536 counts per revolution) Move time in multiples of 0.01s 0 - 0xFFFF, e.g. 10.00 seconds == 0x03e8 Direction 0x00 clockwise, 0x01 anti-clockwise If time/distance > 2⁷, where distance is in motor counts and time in ms, a smooth trajectory cannot be guaranteed due to minimum velocity limitations.</p>
Command	0x5F
Number of data bytes	5
Data 0	Move MSB
Data 1	Move LSB
Data 2	Time MSB
Data 3	Time LSB
Data 4	Direction

0x75 SET PHASE ALIGNMENT CURRENT LIMIT

Purpose	Set the 11-bit limit for current used in phase alignment
Message Type	Data write message
Notes	<p>Current limit which is only applied to phase alignment calibration stage, before current control is active. Default 0x0800 is 1.3A 0x0400 is 1A 0x0180 is 0.16A</p>
Command	0x75
Number of data bytes	2
Data 0	MSB of current limit
Data 1	LSB of current limit

0x76 GET PHASE ALIGNMENT CURRENT LIMIT

Purpose	Get the 11-bit limit for current used in phase alignment
Message Type	Data read message
Notes	See 0x75
Command	0x76
Number of data bytes	2
Data 0	MSB of current limit
Data 1	LSB of current limit

0x83 SET DYNAMIC TRAJECTORY MODE

Purpose	Dynamic Trajectory Mode allows the motor to accept new move commands while in the middle of a current move, and recalculate desired velocity profile from a moving start. Without Dynamic Trajectory Mode (default), the motor comes to a stop in order to calculate the new trajectory.
Message Type	Data write message
Notes	0x00: Dynamic Trajectory Mode OFF (default) 0x01: Dynamic Trajectory Mode ON
Command	0x83
Number of data bytes	1

0x84 GET DYNAMIC TRAJECTORY MODE

Purpose	Dynamic Trajectory Mode allows the motor to accept new move commands while in the middle of a current move, and recalculate desired velocity profile from a moving start. Without Dynamic Trajectory Mode (default), the motor comes to a stop in order to calculate the new trajectory.
Message Type	Data read message
Notes	0x00: Dynamic Trajectory Mode OFF (default) 0x01: Dynamic Trajectory Mode ON
Command	0x84
Number of data bytes	1

0x95 SET CURRENT CONTROLLER GAINS KP AND KI

Purpose	Set the gains for current controller
Message Type	Data write message
Notes	Sets both Kp and Ki with the same command. Defaults 0x0100 and 0x0100
Command	0x95
Number of data bytes	4
Data 0	Kp MSB
Data 1	Kp LSB
Data 2	Ki MSB
Data 3	Ki LSB

0x96 SET CURRENT CONTROLLER GAINS KP AND KI

Purpose	Get the gains for current controller
Message Type	Type Data read message
Notes	See 0x95
Command	0x96
Number of data bytes	4

0x97 SET USE CURRENT CONTROLLER

Purpose	The MRD current controller is used to limit the motor power usage over a range of temperatures.
Message Type	Data write message
Notes	0x00: Current controller OFF 0x01: Current controller ON (default)
Command	0x97
Number of data bytes	1

0x98 GET USE CURRENT CONTROLLER

Purpose	The MRD current controller is used to limit the motor power usage over a range of temperatures.
Message Type	Data read message
Notes	See 0x97
Command	0x98
Number of data bytes	1

0x99 SET USE OVER TEMPERATURE PROTECTION

Purpose	Turn on or off OTP
Message Type	Data write message
Notes	The OTP feature will send the motor to sleep if its internal temperature goes above the threshold set with command 0x15 (default 87degC). The motor will remain sleeping until manually reset/woken, even if temperature reduces. 0x00: OTP OFF 0x01: OTP ON (default)
Command	0x99
Number of data bytes	1

0x9A GET USE OVER TEMPERATURE PROTECTION

Purpose	Get current usage of OTP function
Message Type	Data read message
Notes	See 0x99
Command	0x9a
Number of data bytes	1

0x9B GET INTERNAL TEMPERTURE READING

Purpose	Read the temperature from the internal DSP sensor
Message Type	Data read message
Notes	The value from this sensor is used by the Over Temp Protection feature. Temp is in integer degrees C e.g. 0x0030 = 48degC
Command	0x9B
Number of data bytes	2
Data 0	MSB of temperature
Data 1	LSB of temperature

0xFE GET PROGRAM STATE

Purpose	Query run status of motor firmware
Message Type	Data read message
Notes	This command will produce a response in both IFP and normal running modes.
Command	0xFE
Number of data bytes	1
Data 0	0x00 - Motor drive system started 0x01 - Infield programming mode, launch timeout active 0x02 - Infield programming mode, launch timeout halted, device will stay in IFP mode until launched by user

7.6. Recommended Configurations

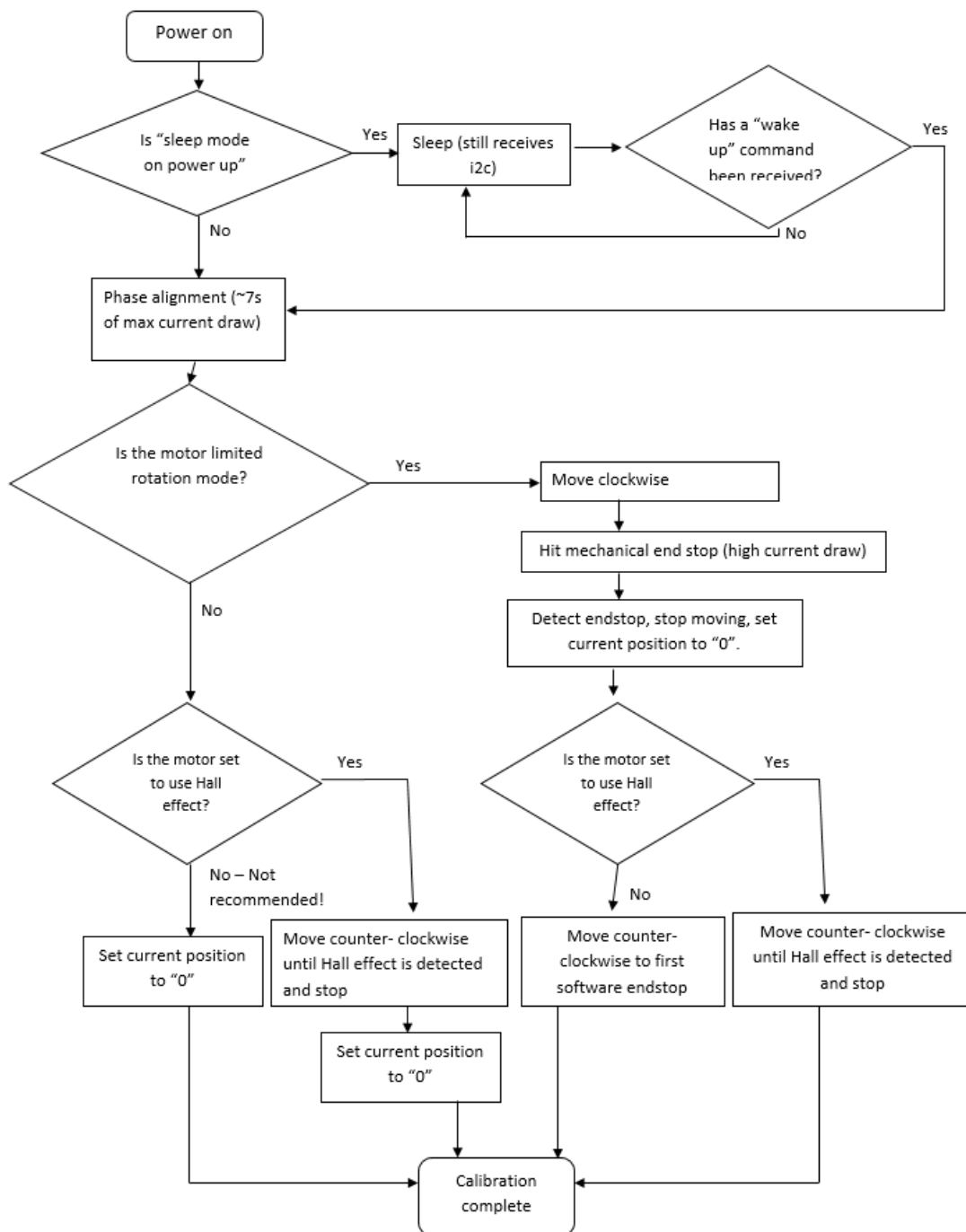
Parameter	I ² C command	I ² C data					
		Firmware defaults (v5.8.161 or earlier)	Firmware defaults (v5.8.162 or later)	24W Unloaded Benchtop usage	*20/26W Rated Load – Pan Rotational Inertia = 0.03112kg m ²	*4/4W Rated Load – Pan Rotational Inertia = 0.0027kg m ²	*20/26W Rated Load – Tilt Rotational Inertia = 0.00794kg m ²
Kp	0x0C	0x0100	0x0100	0x0100	0x1000	0x0600	0x0800
Ki	0x0E	0x0005	0x0005	0x0005	0x0001	0x0040	0x0001
Kd	0x10	0x0100	0x0100	0x0100	0x4000	0x1000	0x1800
Max Acceleration	0x08	0x0040	0x0040	0x0040	0x00c0	0x00c0	0x00c0
Ud filter factor	0x4c	0x01	0x01	0x01	0x04	0x00	0x01
Low pass filter	0x43	0x00	0x00	0x00	0x00	0x00	0x00
Kc	0x46	0x0266	0x0266	0x0266	0x0100	0x0100	0x0266
Pos filter factor	0x58	0x00fd27d2	0x00fd27d2	0x00fd27d2	0x00f33333	0x00f33333	0x00f33333
Current controller	0x97	0x01	0x01	0x01	0x01	0x01	0x01
Current Kp and Ki	0x95	0x01000100	0x01000100	0x01000100	0x00a00021	0x0040c000	0x00a00021
Cur. cont. set point	0x51	0x0700	0x0700	0x0700	0x0400	0x0140	0x0400
Continuous	0x19	0x0000	0x0000	0x0001	0x0001	0x0001	0x0000
Use hall effect	0x4e	0x00	0x00	0x01	0x01	0x01	0x00
Sleep on power up	0x1d	0x00	0x01	0x01	0x01	0x01	0x01
Phase alignment current limit	0x75	0x0800	0x0800	0x0400	0x03A0	0x0180	0x03A0
Init method	0x56	0x00	0x00	0x01	0x01	0x01	0x01
Dynamic Trajectory Mode	0x83	0x00	0x00	0x01	0x01	0x01	0x01
Turbo mode	0x53	0x00	0x00	0x01	0x01	0x01	0x01
Over Temp Protection mode	0x99	0x01	0x01	0x01	0x01	0x01	0x01
OTP set point	0x15	0x57	0x57	0x57	0x57	0x57	0x57

*The Rated Load parameters are configured so that the motor does not draw more than the specified power under normal operation/during initialisation e.g. 20W during normal operation and 26W during initialisation.

The motor start-up and homing routine is controlled by 3 settings for each motor:

- “Sleep on power up” mode
- “Continuous” mode
- “Use hall effect” mode. In this document the legacy term “hall effect” is used, even though the homing signal is created by an optical device for the MRD motor.

The behaviour of the motor and impact of each of the settings can be seen in this flow chart:



Note that the “clockwise” and “counter clockwise” directions describe the movement of the rotor relative to the PCBA, when viewed from the side of the motor that the rotor is mounted on.

For use in a standard pan-tilt gimbal system with a slipping + continuous motion in pan, and no slipping + mechanical end stop in tilt, the recommended settings are:

Pan: Continuous and Use hall effect.

Tilt: Not continuous and Not use hall effect.

During the “phase alignment” process the motor will draw max current. The “sleep on power up” setting can be used along with staggered “wake up” I²C commands to ensure that only one motor is in this process at a time, in situations where power supply limitations apply.

7.7. Firmware Update Mode

In this section we describe the ability to update the motor firmware remotely using the Infield Programming (IFP) mode. When in IFP mode the same protocol applies for I²C data read and write messages as in section 2, except the commands used are in the range 0xF0 upwards.

7.7.1. Firmware Image File

BLDC motor firmware is distributed as a 32KB binary file. The layout of the file is as follows:

Byte Address	Contents
0x0000 - 0x0001	Magic Number 0xA5A5
0x0002 - 0x0003	Uint16 Major version
0x0004 - 0x0005	Uint16 Middle version
0x0006 - 0x0007	Uint16 Minor version
0x0008 - 0x7FFB	Program data
0x7FFC - 0x7FFD	Magic Number 0xA5A5
0x7FFE - 0x7FFF	16-bit LRC

7.7.2. Programming Procedure

Below is pseudo-code that would be run on a host to program the BLDC motor using the I²C bus.

```
void I2C_write (u8 command , u8* tx_data , u8 n_data_bytes );
void I2C_read (u8 command , u8* rx_data , u8 n_data_bytes );
int fw_upgrade ()
{
    u8 rx_data [4];
    u8 tx_data [4];
    // reset ( if already powered on ), otherwise power on , then interrupt launch
    I2C_write (0 x01 , tx_data , 0);
```

```
delay_ms (250) ; // max wait 500 ms , min wait 15 ms
// " hold " command
I2C_read (0 xF0 , rx_data , 1);
if( rx_data [0] != 0) return -1;
// " prepare to send " command
I2C_read (0 xF1 , rx_data , 1);
if( rx_data [0] != 0) return -1;
// send data chunk by chunk
{
    int i_c = 0;
    u8 * p_d = BLDC_Drive_APP ; // application image data
    const int chunk_size = 4;
    int chunks = BLDC_Drive_APP_ARRAY_LEN / chunk_size ;
    for ( i_c = 0; i_c < chunks ; i_c ++ )
    {
        I2C_write (0 xF2 , p_d , chunk_size ); // load chunk to device ram
        I2C_read (0 xF3 , rx_data , 1); // save chunk to device flash
        if( rx_data [0] != 0) return -1;
        p_d += chunk_size ;
    }
}
// check image integrity in device memory
I2C_read (0 xF4 , rx_data , 1);
if( rx_data [0] != 0) return -1;
// finished , now launch
I2C_write (0 xF5 , tx_data , 0);
return 0;
}
```

Important points to note:

- The "prepare to send" command (0xF1) must be sent before the application is written to the device. This erases the current application, and the device holds the SCL line low while this operation is in progress.
- Data is transferred to the device in 4-byte chunks with command 0xF2
- Command 0xF3 is used to save each chunk to non-volatile memory. The device holds the SCL line low until this is completed (5ms).
- The whole update procedure takes approximately 5 seconds from start to finish.

7.7.3. IFP Commands Summary

Code	Description	Message type	# data bytes
0xF0	Hold launch	Data read	1
0xF1	Prepare to send	Data read	1
0xF2	Send data chunk	Data write	4
0xF3	Save data chunk	Data read	1
0xF4	Verify program	Data read	1
0xF5	Launch	Data write	1
0xF6	Get ifp mode version	Data read	2
0xFE	Get program state	Data read	1

7.7.4. IFP Commands in Detail

0xF0 HOLD LAUNCH

Purpose	Hold in IFP mode
Message Type	Data read message
Notes	If received within 500ms following power on or a firmware reset, halt launch and stay in reprogramming mode
Command	0xF0
Number of data bytes	1
Data 0	0x00 - hold successful

0xF1 PREPARE TO SEND

Purpose	Prepare device to receive new firmware image
Message Type	Data read message
Notes	This erases the current firmware from ash memory. The device will hold SCL low during this operation (approximately 3 seconds) and only return the status byte once this process is finished.
Command	0xF1
Number of data bytes	1
Data 0	0x00 - device ready

0xF2 SEND DATA CHUNK

Purpose	Send next data chunk to device
Message Type	Data write message
Notes	Send the next 4 bytes of the firmware image to the device.
Command	0xF2
Number of data bytes	4

0xF3 SAVE DATA CHUNK

Purpose	Commit last data chunk sent to ash memory
Message Type	Data read message
Notes	Writes the last data sent to the device to non-volatile memory. The device will hold SCL low until the write operation is finished.
Command	0xF3
Number of data bytes	1
Data 0	0x00 - Write successful 0xFF - Write failed

0xF4 VERIFY PROGRAM

Purpose	Verify image currently stored in the device ash memory
Message Type	Data read message
Notes	
Command	0xF4
Number of data bytes	1
Data 0	0x00 - Image verification OK 0x01 - LRC failed 0x02 - Image not a valid program

0xF5 LAUNCH

Purpose	Launch into normal operation
Message Type	Data write message
Notes	
Command	0xF5
Number of data bytes	1

0xF6 GET IFP MODE VERSION

Purpose	Read the version of the infield update program currently running
Message Type	Data read message
Notes	NB this is NOT the same as 0x1B \Get firmware version".
Command	0xF6
Number of data bytes	2
Data 0	UInt8 Major version
Data 1	UInt8 Minor version

0xFE GET PROGRAM STATE

Purpose	Query run status of motor firmware
Message Type	Data read message
Notes	
Command	0xFE
Number of data bytes	1
Data 0	0x00 - Motor drive system started 0x01 - Infield programming mode, launch timeout active 0x02 - Infield programming mode, launch timeout halted, device will stay in IFP mode until launched by user