

# TC410 Telescope Drive

Technical manual, Revision 2, March 2006



## Table of Contents

1. Introduction.....	3
1.1. Disclaimer.....	3
1.2. Terminology.....	3
1.2.1. Axes.....	3
1.2.2. Hard and soft limits.....	3
1.3. Specifications.....	3
2. Connections.....	5
2.1. Location of connectors.....	5
2.2. Power inlet.....	5
2.3. Motors.....	5
2.4. Ethernet.....	6
2.5. EIA-232.....	6
2.6. Joystick.....	7
2.7. Analogue/Digital Autoguider.....	8
2.8. Digital Autoguider.....	8
2.9. Multi-function bus.....	9
2.9.1. Encoders.....	11
2.9.2. Joystick and autoguider.....	12
2.9.3. Limits.....	12
3. Motors.....	13
3.1. 4-lead motors.....	13
3.2. 6-lead motors.....	14
3.2.1. Half coil configuration.....	14
3.2.2. Full coil configuration.....	14
3.3. 8-lead motors.....	14
3.3.1. Parallel configuration.....	15
3.3.2. Series configuration.....	15
4. Remote Monitor And Control.....	16
4.1. Introduction.....	16
4.2. Control philosophy.....	16
4.3. Introduction to Modbus.....	16
4.4. Modbus Transport Layer.....	16
4.5. Motion control.....	17
4.5.1. Slew to target.....	17
4.5.2. Direct speed control.....	17
4.5.3. Position synchronize.....	17
4.5.4. All stop.....	17
4.5.5. Velocity control flags.....	17
4.6. Data formats.....	18
4.6.1. Drive types.....	18
4.6.2. Phase current.....	18

4.6.3. Position and Motion.....	18
4.6.4. Encoders.....	19
4.6.5. Gearing.....	20
4.6.6. Joystick position.....	20
4.6.7. Joystick and Autoguider modes.....	20
4.6.8. Temperatures.....	21
4.6.9. Coordinates.....	21
4.7. Modbus registers.....	21
4.7.1. Coils.....	21
4.7.2. Discrete Inputs.....	21
4.7.3. Input Registers.....	23
4.7.4. Holding Registers.....	25

## Index of Tables

Table 1: Electrical specifications.....	4
Table 2: Supported motor characteristics.....	4
Table 3: Mechanical specifications.....	4
Table 4: Environmental specifications.....	4
Table 5: Operating characteristics.....	4
Table 6: Motor connector pin out.....	5
Table 7: EIA-232 pin out.....	6
Table 8: Joystick pin out.....	7
Table 9: Analogue autoguider pin out.....	8
Table 10: Digital autoguider pin out.....	8
Table 11: Multifunction bus pin out by pin.....	10
Table 12: Multifunction bus pin out for encoders.....	11
Table 13: Multifunction bus pin out for joystick and autoguider.....	12
Table 14: Multifunction bus pin out for limits.....	12
Table 15: Velocity control priorities.....	17
Table 16: Relationship between position, velocity and acceleration data formats.....	18
Table 17: Modbus Coil registers.....	22
Table 18: Modbus Discrete Inputs - power control.....	22
Table 19: Modbus Discrete Inputs - peripheral senses.....	22
Table 20: Modbus Discrete Inputs - power supply alarms.....	22
Table 21: Modbus Discrete Inputs - limit senses.....	23
Table 22: Modbus Discrete Inputs - movement flags.....	23
Table 23: Modbus Input Registers - general status.....	25
Table 24: Modbus Input Registers - axis 1 status.....	25
Table 25: Modbus Input Registers - axis 2 status.....	25
Table 26: Modbus Holding Registers - general.....	26
Table 27: Modbus Holding Registers - axis 1 configuration.....	27
Table 28: Modbus Holding Registers - axis 2 configuration.....	28
Table 29: Modbus Holding Registers - axis 1 status and control.....	28
Table 30: Modbus Holding Registers - axis 2 status and control.....	29
Table 31: Modbus Holding Registers - actions.....	29

# 1. Introduction

## 1.1. Disclaimer

This manual is provided in good faith and is not guaranteed to be a complete and error-free document.

This manual covers only the TC410 model, not its predecessor the TC400 (which is a significantly different design from a control point of view).

## 1.2. Specifications

*Table 1: Electrical specifications*

<b>Power supply</b>	230V AC 45-55Hz
<b>Power consumption</b>	160VA max (peak) <sup>(1)</sup>
<b>Standby power consumption</b>	2W <sup>(2)</sup>
<b>Line protection</b>	1A slow blow fuse
<b>Low voltage protection</b>	10A fast flow fuse

*Table 2: Supported motor characteristics*

<b>Motor type</b>	2/4-phase stepping <sup>(3)</sup>
<b>Phase voltage</b>	1.4V – 7.0V
<b>Phase current</b>	1.0A – 5.0A DC/RMS
<b>Phase inductance</b>	500µH minimum
<b>Step angle</b>	no restriction

*Table 3: Mechanical specifications*

<b>Dimensions</b>	482x80x150mm
<b>Weight</b>	5kg
<b>Mounting method</b>	2U rack mounting
<b>Motor cable type</b>	4 twisted pairs + earth
<b>Motor cable specification</b>	5A/conductor
<b>Motor cable length</b>	10m max

*Table 4: Environmental specifications*

<b>Operating temperature</b>	0°C - 50°C
<b>Storage temperature</b>	-25°C – 100°C
<b>Operating humidity</b>	5% - 95% non-condensing
<b>Noise level</b>	<sup>4</sup>

*Table 5: Operating characteristics*

<b>Maximum motor speed</b>	20000 micro steps/second
<b>Microsteps/full step</b>	10/16/256 <sup>(5)</sup>

1 Typical power consumption is less than 50W RMS when driving 2x 3-amp motors

2 Motor control system shut down, Ethernet communication available

3 Motors are always driven in bipolar/2-phase mode

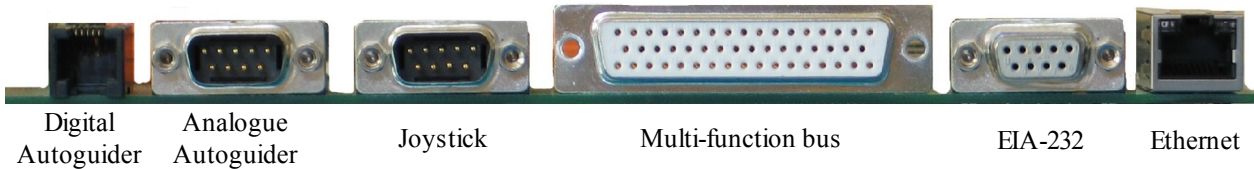
4 All fans are temperature controlled and are only activated when internal operating temperature exceeds set thresholds.

5 Gecko drive is 10 microsteps/step, IMS is 16 microsteps/step at high speed and 256 microsteps/step at low speed.

## 2. Connections

### 2.1. Location of connectors

Illustration 1: Rear panel connectors



### 2.2. Power inlet

Power inlet is a standard IEC inlet to be connected only to 230-250V.

### 2.3. Motors

To allow the drive to be located away from the telescope, the motor connectors carries power for both motors in addition to an earth connected directly to the utility earth.

Pin	Function
1	Axis 2 A
2	Axis 2 B
3	Axis 2 C
4	Axis 2 D
5	Axis 1 A
6	Axis 1 B
7	Axis 1 C
8	Axis 1 D
9	Earth

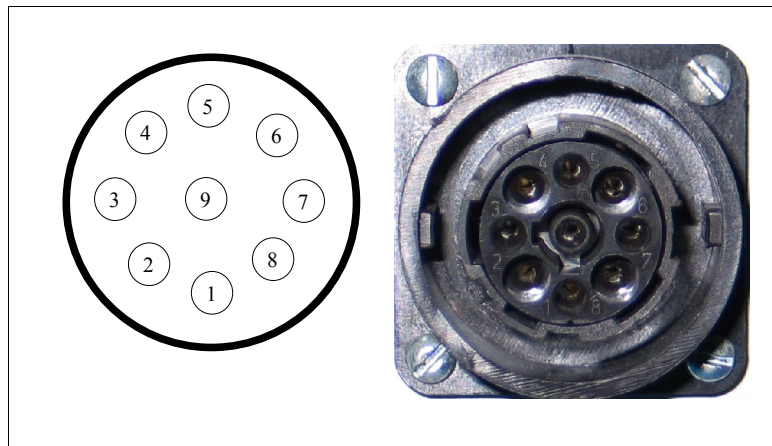


Table 6: Motor connector pin out

The designations A, B, C and D are the four motor control lines. A and B form a pair (and should be twisted together in the cable) as do C and D.

It is highly recommended that metallic parts of the telescope be earthed with the available earth connection for reasons of safety in the event of a malfunction.

The connections to the motors should be wired using high-current twisted pair cabling. The TC410 uses switching techniques for motor current control and twisted pairs greatly reduce the electromagnetic emissions that would otherwise be generated by a long cable carrying such signals. Shielding may also be used (but not required) in which case the shield must only be drained (to

earth) at the drive's end of the cable. This is so that any fault currents are conducted only by the dedicated earth cable and not the shield.

See section 3 (page 12) for further information about wiring the motors themselves.

## 2.4. Ethernet

The TC410 should operate in a LAN environment where a DHCP server is available to automatically assign an IP address to the drive when it starts. The drive should be connected to a “live” network before powering it on otherwise acquisition of an IP address may not occur.

Because TCP connections must be made to the drive in order to communicate with it, it is necessary to know what IP address has been assigned to the drive. The recommended way to do this is to instruct the DHCP server to assigned a fixed address to the drive's MAC address. It is also useful to assign a DNS name to this address (if a local DNS server is available) so that the drive can be referenced by name.

Further details on network administration is outside the scope of this manual.

The Ethernet connection is the standard RJ-45 Ethernet scheme.

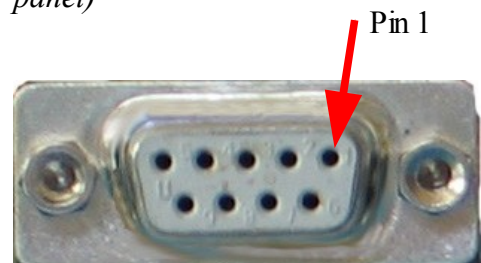
## 2.5. EIA-232

Standard 3-wire EIA-232 with DCE wiring (e.g. uses a straight-through cable to connection to a computer).

Pin	Function
1	Not connected
2	Transmit (out)
3	Receive (in)
4	Not connected
5	Signal ground
6	Not connected
7	Not connected
8	Not connected
9	Not connected

Table 7: EIA-232 pin out

Illustration 2: DE9F EIA-232 connector (view into drive rear panel)



Data rate and format is fixed at 9600 N81:

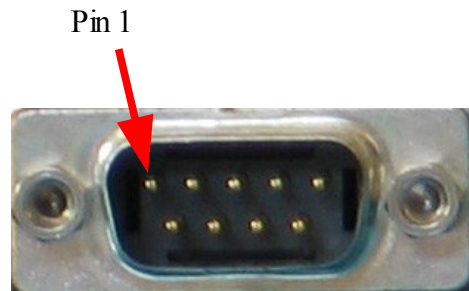
- 9600 baud
- 8 data bits
- 1 stop bit
- No parity

## 2.6. Joystick

The main hand-held control for the TC410 is a dual-axis analogue joystick. In principle it can be any device that generates the required analogue voltage (including a switched resistor digital joystick).

Pin	Function
1	Button 0
2	Button 1
3	Button 2
4	Button 3
5	Voltage reference
6	Signal ground
7	Axis 1 input
8	Axis 2 input
9	Joystick detect

*Illustration 3: DE9M Joystick connector (view into drive rear panel)*



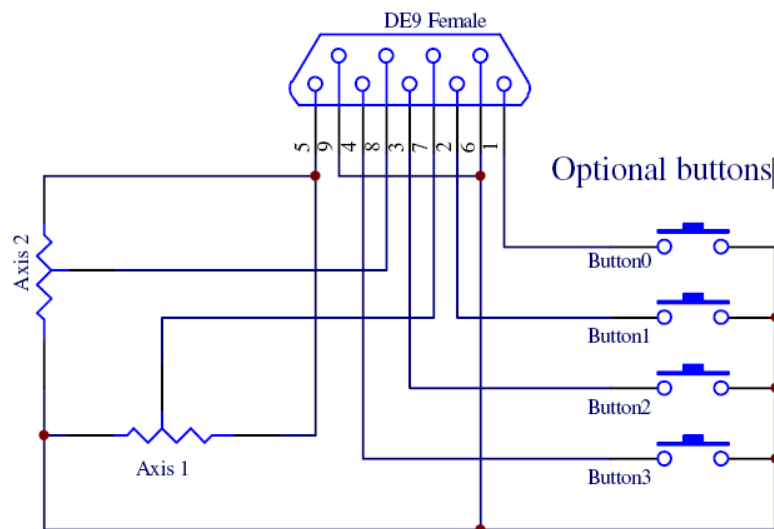
*Table 8: Joystick pin out*

The button inputs are not presently used.

Voltage reference is 5V with current limiting. The joystick should not draw more than 100mA from this pin.

Voltage reference and signal ground define the range of voltage that should be presented on pins 7 and 8 with the nominal dead zone being the voltage half way between these (i.e. 2.5V).

The TC410 detects the joystick by the presence of 0V on pin 9. Pin 9 should be connected to the signal ground (pin 6) to achieve this.



*Illustration 4: Joystick schematic*

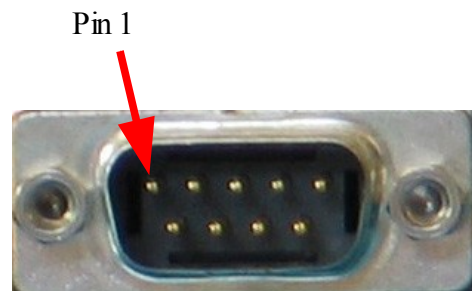
## 2.7. Analogue/Digital Autoguiders

The analogue autoguiders are designed to allow fine analogue control independently of the main joystick. As such its operation is mostly the same as the joystick except that movement scale is much smaller.

Pin	Function
1	North
2	South
3	East
4	West
5	Voltage reference
6	Signal ground
7	Axis 1 input
8	Axis 2 input
9	Autoguiders detect

Table 9: Analogue autoguiders pin out

Illustration 5: DE9M Autoguiders connector (view into drive rear panel)



The 4 direction inputs on pins 1-4 are active-low. i.e. a voltage of 0V should be applied to effect a movement in the desired direction.

Voltage reference is 5V with current limiting. The autoguiders/joystick should not draw more than 100mA from this pin.

Voltage reference and signal ground define the range of voltage that should be presented on pins 7 and 8 with the nominal dead zone being the voltage half way between these (i.e. 2.5V).

The TC410 detects the autoguiders by the presence of 0V on pin 9. Pin 9 should be connected to the signal ground (pin 6) to achieve this.

The autoguiders wiring is identical to the joystick except that the joystick's 4 buttons become digital direction controls.

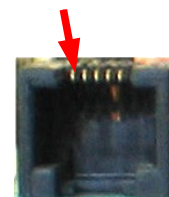
## 2.8. Digital Autoguiders

This is a "standard" digital autoguiders input that is compatible with normal 4-quadrant CCD autoguiders. The connector is an RJ11 with pin out as follows:

Pin	Function
1	Voltage reference
2	Signal ground
3	West
4	South
5	North
6	East

Table 10: Digital autoguiders pin out

Illustration 6: RJ11 autoguiders socket



The 4 direction inputs on pins 1-4 are active-low. i.e. a voltage of 0V should be applied to effect a movement in the desired direction.

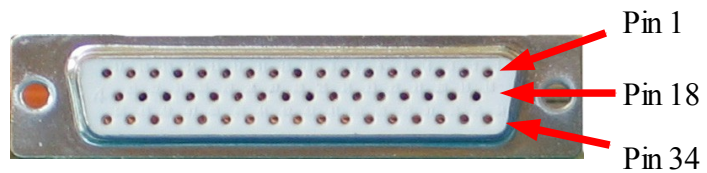
Voltage reference is 5V with current limiting. The autoguider/joystick should not draw more than 100mA from this pin.

## 2.9. Multi-function bus

This connector is intended as the main connection for data signals between the drive and the telescope. It duplicates some of the functions on other connectors as well as providing the only connection for the encoders and limit inputs.

Normally this would be connected to a breakout junction very close to the telescope itself into which can be connected the joystick, autoguider, encoders and limit switches.

*Illustration 7: DD50 multifunction socket*



<b>Pin</b>	<b>Function</b>	<b>Pin</b>	<b>Function</b>
1	Axis 2 encoder D15	26	Axis 1 encoder D10
2	Axis 2 encoder D6	27	Axis 1 encoder D1
3	Axis 2 encoder D12	28	Axis 2 -ve limit
4	Axis 2 encoder D3	29	Axis 1 +ve limit
5	Axis 2 encoder D9	30	Autoguider axis 1
6	Axis 2 encoder D0	31	Autoguider south
7	Axis 1 encoder D14	32	Joystick axis 2
8	Axis 1 encoder D5	33	Joystick button 0
9	Axis 1 encoder D11	34	Axis 2 encoder D7
10	Axis 1 encoder D2	35	Axis 2 encoder D13
11	Axis 1 encoder D8	36	Axis 2 encoder D4
12	Axis 2 +ve limit	37	Axis 2 encoder D10
13	Autoguider detect	38	Axis 2 encoder D1
14	Autoguider west	39	Axis 1 encoder D15
15	Autoguider north	40	Axis 1 encoder D6
16	Joystick axis 1	41	Axis 1 encoder D12
17	Signal ground	42	Axis 1 encoder D3
18	Axis 2 encoder D14	43	Axis 1 encoder D9
19	Axis 2 encoder D5	44	Axis 1 encoder D0
20	Axis 2 encoder D11	45	Axis 1 -ve limit
21	Axis 2 encoder D2	46	Autoguider axis 2
22	Axis 2 encoder D8	47	Autoguider east
23	Axis 1 encoder D7	48	Joystick detect
24	Axis 1 encoder D13	49	Joystick button 1
25	Axis 1 encoder D4	50	Voltage reference

*Table 11: Multifunction bus pin out by pin*

The total current that can be supplied from the voltage reference pin is 500mA.

The functions of the various connections are described in the following sections.

### 2.9.1. Encoders

The TC410 supports optional absolute Gray scale encoders up to 16 bits resolution. Incremental encoders are not supported.

<b>Axis 1 Encoder</b>	<b>Pin</b>	<b>Axis 2 encoder</b>	<b>Pin</b>
Axis 1 encoder D0	<b>44</b>	Axis 2 encoder D0	<b>6</b>
Axis 1 encoder D1	<b>27</b>	Axis 2 encoder D1	<b>38</b>
Axis 1 encoder D2	<b>10</b>	Axis 2 encoder D2	<b>21</b>
Axis 1 encoder D3	<b>42</b>	Axis 2 encoder D3	<b>4</b>
Axis 1 encoder D4	<b>25</b>	Axis 2 encoder D4	<b>36</b>
Axis 1 encoder D5	<b>8</b>	Axis 2 encoder D5	<b>19</b>
Axis 1 encoder D6	<b>40</b>	Axis 2 encoder D6	<b>2</b>
Axis 1 encoder D7	<b>23</b>	Axis 2 encoder D7	<b>34</b>
Axis 1 encoder D8	<b>11</b>	Axis 2 encoder D8	<b>22</b>
Axis 1 encoder D9	<b>43</b>	Axis 2 encoder D9	<b>5</b>
Axis 1 encoder D10	<b>26</b>	Axis 2 encoder D10	<b>37</b>
Axis 1 encoder D11	<b>9</b>	Axis 2 encoder D11	<b>20</b>
Axis 1 encoder D12	<b>41</b>	Axis 2 encoder D12	<b>3</b>
Axis 1 encoder D13	<b>24</b>	Axis 2 encoder D13	<b>35</b>
Axis 1 encoder D14	<b>7</b>	Axis 2 encoder D14	<b>18</b>
Axis 1 encoder D15	<b>39</b>	Axis 2 encoder D15	<b>1</b>
Voltage reference	<b>50</b>	Voltage reference	<b>50</b>
Signal ground	<b>17</b>	Signal ground	<b>17</b>

*Table 12: Multifunction bus pin out for encoders*

### 2.9.2. Joystick and autoguider

The joystick and autoguider connections are functionally identical to what appears on the dedicated connectors and indeed they are parallel wired. Therefore each of these input devices must only be connected via one path at a time. e.g. do not connect a joystick to the dedicated DE9 connector if a joystick is connected via the DD50 multifunction bus.

<b>Joystick</b>	<b>Pin</b>	<b>Autoguider</b>	<b>Pin</b>
Axis 1	<b>16</b>	Axis 1	<b>30</b>
Axis 2	<b>32</b>	Axis 2	<b>46</b>
Button 0	<b>33</b>	North	<b>15</b>
Button 1	<b>49</b>	South	<b>31</b>
Button 2	-	East	<b>47</b>
Button 3	-	West	<b>14</b>
Detect	<b>48</b>	Detect	<b>13</b>
Voltage reference	<b>50</b>	Voltage reference7	<b>50</b>
Signal ground	<b>17</b>	Signal ground	<b>17</b>

*Table 13: Multifunction bus pin out for joystick and autoguider*

### 2.9.3. Limits

The limit inputs are active-low. The voltage reference can be used if required but normally it is sufficient to use a microswitch to created a close-contact circuit between an input and the signal ground.

<b>Function</b>	<b>Pin</b>
Axis 1 +ve	<b>29</b>
Axis 1 -ve	<b>45</b>
Axis 2 +ve	<b>12</b>
Axis 2 -ve	<b>29</b>
Voltage reference	<b>50</b>
Signal ground	<b>17</b>

*Table 14: Multifunction bus pin out for limits*

## 3. Motors

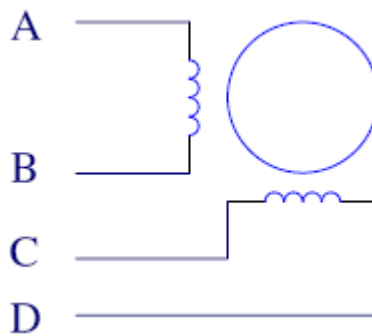
The TC410 can drive a wide range of stepper motors with the specified range of phase currents. It operates in the bipolar mode which means it can directly drive 4-wire motors. 6-wire and 8-wire motors can each be driven in two ways by a bipolar drive.

For all motor configurations, all four connections to the drive (A, B, C and D) must be connected. It only matters that one coil is connected to A/B and the other coil is connected to C/D. It doesn't matter which coil is connected to A/B and which to C/D. Furthermore, for the A/B coil it doesn't matter which wire is connected to A and which is connected to B. Different permutations simply determine which direction the motor will rotate – and this can be reversed in software.

When determining the operating current for a motor it is important to know which configuration the motor's specified current refers to and to make appropriate adjustments if the motor is operated in a different configuration.

See section 4.7.2 on page 18 for the maximum motor current that can be used.

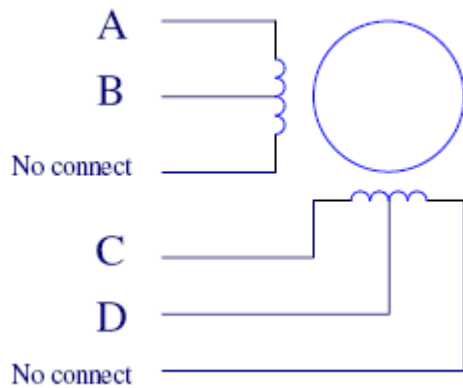
### 3.1. 4-lead motors



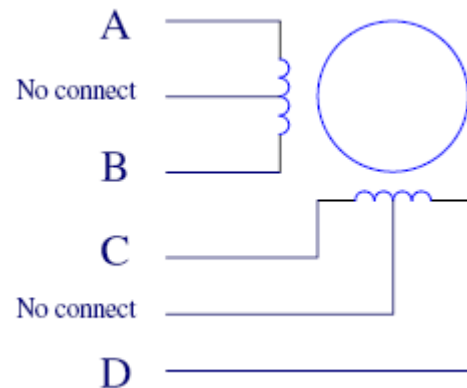
*Illustration 8: 4-lead motor connection*

The simplest type of motor to connect since there is only one way to wire them. When configuring the current for these motors, the value printed on the motor should be used.

### 3.2. 6-lead motors



*Illustration 9: 6-lead half coil configuration*



*Illustration 10: 6-lead full coil configuration*

There are two ways to connect a 6-wire motor to a bipolar drive – each has its advantages in different situations.

#### 3.2.1. Half coil configuration

In this configuration only half of the coil is used and the other half is left “floating”. Because of this, the inductance is only half of the motor's specified (full coil) inductance and therefore lower torque. However the motor will be more stable at higher speeds.

This is the recommended configuration of 6-wire motors for driving telescopes.

When configuring the current for these motors, the motor's specified (per-phase) current should be used because this is the same as its intended unipolar configuration.

#### 3.2.2. Full coil configuration

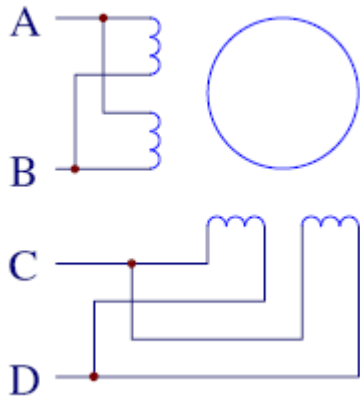
In this configuration the whole coil is used and the centre tap is left floating. Because this is equivalent to operating 2 coils in series (compared to the half coil configuration), a lower current must be used to operate at the same power and to dissipate the same amount of heat. This give a higher torque for the same power at the expense of poorer performance at high speed.

This configuration is not recommended for driving telescopes.

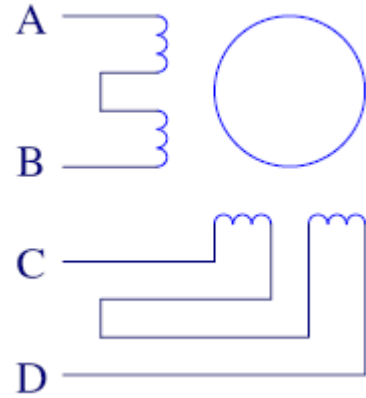
When configuring the current for these motors, multiply the specified per-phase current rating by 0.7.

### 3.3. 8-lead motors

8-wire motors cannot easily be wired without a datasheet because without knowing the polarity of each coil it is possible to wire them in anti-parallel or anti-series. Consult the motor's datasheet for correctly wiring these motors.



*Illustration 11: 8-lead parallel configuration*



*Illustration 12: 8-lead series configuration*

### **3.3.1. Parallel configuration**

This configuration gives the motor better torque at higher speeds but less torque at lower speeds. It is similar to the 6-wire half coil configuration.

This is the recommended configuration of 8-wire motors for driving telescopes.

8-wire motors may specify their current ratings in different ways and it is important to interpret this correctly. If the per-phase current is specified then configure 1.4 times this value. If the parallel or bipolar current is specified then configure use this value directly.

### **3.3.2. Series configuration**

This configuration gives high torque at low speeds but poor performance at high speed. This is not recommended for driving telescopes.

If the per-phase current is specified then configure 0.7 times this value. If the parallel or bipolar current is specified then configure 0.5 times this value.

## 4. Remote Monitor And Control

### 4.1. Introduction

TC410 communication uses an industrial standard called Modbus. This protocol is available over EIA-232 or Ethernet. It is a query/response protocol which can be used monitor the status of the drive, control it and configure it. Because Modbus is designed for multi-drop buses, the TC410 emits no unsolicited information. i.e. it cannot produce notification messages, it can only be polled.

The TC410 is an “open architecture” drive. This means it is not designed to control any particular type of telescope (indeed it not even be a telescope). It only requires that the motors be stepper motors – the step size, gearing and motor currents are configurable. Therefore, an application communicating with a TC410 needs to be able to configure the drive as well and monitor and control it when in normal operation.

Drive configuration is stored in internal non-volatile memory and needs to be configured at least once to match the physical properties of the telescope being driven.

### 4.2. Control philosophy

The TC410 is not designed to be a “smart” stand-alone telescope drive but rather a relatively “dumb” motion controller under the control of a central automation controller. Because of this there is no concept of real time or coordinate systems and virtually all astronomy functionality is handled externally. This permits a high degree of versatility at the expense of pushing a lot of control complexity out to a control computer.

### 4.3. Terminology

#### 4.3.1. Axes

Because the TC410 is not specifically designed for any particular type of telescope (although it works best with equatorial mounts), mount agnostic terms are used for driven axes. The terms “Axis 1” and “Axis 2” are used which, on an equatorial telescope would refer to the primary (right ascension) and secondary (declination) axes respectively.

#### 4.3.2. Hard and soft limits

The TC410 has two types of limits which prevent the telescope from being driven past certain limits. Typically these are needed to ensure the telescope cannot be manually driven past travel points for reasons such as cable reach and pier collisions.

*Hard limit:* a limit determined by a physical sensor such as a microswitch. This is a physical electrical input to the drive.

*Soft limit:* a limit determined by the axis being a soft-programmable angle away from its centre position.

*Positive limit:* A limit (hard or soft) encountered when moving in the direction of (mathematically) increasing raw axis position.

*Negative limit:* A limit (hard or soft) encountered when moving in the direction of (mathematically) decreasing raw axis position.

The physical direction associated with positive and negative limits depends on motor wiring and the motor reversing option.

#### **4.4. Introduction to Modbus**

Modbus is a standard generic protocol characterized by a small set of commands which allow the device properties to be read and set using an array of virtual registers.

The Modbus specifications are freely available from [www.modbus.org](http://www.modbus.org). The “Modbus Application Protocol” describes the application layer which defines the transport-independent part of the protocol including what is meant by the different types of registers that following sections assume the reader is familiar with.

#### **4.5. Modbus Transport Layer**

Modbus defines several methods for transporting the application protocol over different types of media. The primary ones are Modbus-RTU, Modbus-ASCII and Modbus-TCP. The TC410 uses Modbus-ASCII over both EIA-232 *and* Ethernet. This has more overhead than the binary protocols but has no timing constraints.

Because EIA-232 and TCP are both single-drop end-to-end connections, the TC410 does not have a configurable address like a normal Modbus device. It responds to the reserved address 255 which acts like a broadcast to a single device.

When operating over Ethernet, the master first establishes a TCP connection to the drive on port 10001 and from then the TCP socket can be treated exactly like a serial connection.

#### **4.6. Motion control**

##### **4.6.1. Slew to target**

To slew to a target position requires some computation. First the required object position must be converted to axis raw positions (as determined by the step size and zero position of the axis) at the time of expected arrival. Once the distance from the current position to the target position is known, the estimated travel time needs to be calculated and this time added to the current hour angle (to calculate the hour angle at estimated time of arrival).

When the raw axis positions are known, they can be written to the target position registers (see 4.8.4.4 and 4.8.4.5). Then the slew is started by setting the “slew to target” bit in the velocity control register (see 4.8.4.1).

##### **4.6.2. Direct speed control**

The remote device can control the motion directly in the same way that the joystick or autoguider can. To do this, set the “remote control” bit in the velocity control register (see 4.8.4.1) then write velocities to the target velocity registers (4.8.4.4 and 4.8.4.5).

### 4.6.3. Position synchronize

To update the position of the telescope without physically moving it (i.e. to resynchronize the reported position with the actual position) simply write the correct position to the position registers.

For each axis the most significant bits must be written first since the full 32-bit position is only updated when the least significant bits are written (i.e. the MSBs are held in a temporary location until all 32-bits are available). When using the Modbus write multiple register command, this ordering occurs naturally.

### 4.6.4. All stop

A special “all stop” command is available (see 4.8.4.6) which ensures the telescope comes to a halt as quickly as possible. This is equivalent to clearing any slew to target operation and zeroing the target velocities.

### 4.6.5. Velocity control flags

The velocity control flags (see 4.8.4.1) determine which of the possible motion control sources are actually allowed to cause motion. They are, in decreasing order of priority:

Priority level	Control source	Idle state
1	Slew to target	Slewing completed
2	Remote direct speed control	Must be explicitly disabled
3	Joystick	Centre dead zone
4	Analogue autoguider	Centre dead zone
5	Digital autoguider	No directions asserted
6	Tracking	No idle state

*Table 15: Velocity control priorities*

When any particular source is “active” (i.e. in its non-idle state), all sources of lower priority are ignored until that source goes idle again.

With the exception of the remote speed control, all other sources can be left enabled since they can automatically enable on demand by virtue of their idle state. When a remote device enables remote speed control (by setting the appropriate bit in the velocity control register) it has near-exclusive control – even when its axis velocities are zero.

## 4.7. Data formats

### 4.7.1. Drive types

The TC410 supports two types of stepper motor drive – “Gecko” and “IMS” – reported in the axis status (see sections 4.8.3.2 and 4.8.3.3). The main difference between them is that the Gecko divides motor steps in 10 microsteps and the IMS into 16 microsteps. This knowledge is required to compute the angular step size of the motor because the TC410 returns position in motor steps.

### 4.7.2. Phase current

The motor phase currents are soft-programmable to allow a range of motors to be used. Phase currents are specified in deciamps and are the DC current that one winding draws with its specified DC voltage applied across it. It is very important not to overdrive the motor by using a higher than specified current because the motor will overheat with very little torque advantage. It is OK to use less than specified current but microstepping will become erratic if less than 50% of the rated current is used.

The minimum motor current is 10 deciamps and the maximum is 30 deciamps for IMS drive types and 50 deciamps for Gecko drive types.

### 4.7.3. Position and Motion

Position and motion information is not translated to units that a human might find useful – in fact the internal representation is simply exposed. The way the position, velocity and acceleration relate to each is shown by this diagram:

<i>Unit</i>	<i>40-47</i>	<i>32-39</i>	<i>24-31</i>	<i>16-23</i>	<i>8-15</i>	<i>0-7</i>
Reported position						
Internal position						
Velocity						
Acceleration						

*Table 16: Relationship between position, velocity and acceleration data formats*

This is further described in the following subsections.

#### 4.7.3.1 Axis position

The TC410 does not have any knowledge of sky co-ordinates – it doesn't even know about angular position. Positions are returned in raw microstep units of  $1/10^{\text{th}}$  of a motor step or  $1/16^{\text{th}}$  of a motor step depending on the drive type (see 4.7.1). A device requesting position information must be aware of the number of steps per revolution of the final drive in order to calculate the angular position.

As shown above, the position is kept internally to 48-bit resolution with the reported position being the 24 most significant bits. The 24 least significant bits correspond to fractions of a microstep.

The (visible) position is an unsigned 24-bit number with the nominal centre of travel for the axis being located at 0x800000. The exact physical position of this centre is not important to the TC410 since the mapping of this number to an actual sky position is determined externally. On an equatorial fork mounted telescope the axis 1 centre would ideally be at zero hour angle and axis 2 centre on the zenith. When performing a coordinate synchronization the external computer would compute the ideal axis positions for a star's position (e.g. mapping axis 1 to 0x800000 for a star exactly on the meridian) and write this to the position registers.

On power-up, the positions for both axes are set to 0x800000 and, in the absence of encoders would require a synchronisation procedure with human intervention. If encoders are available, these can be read to find the physical orientation of the axes and use these to pre-set the position registers.

The target position is the same format as the reported position and is used to auto-slew. This would normally be set prior to enabling “slew to target” in the velocity control.

#### 4.7.3.2 Axis velocity

Velocity is represented as a 32-bit signed number. The maximum velocity is  $\pm 2^{24}-1$  ( $\pm 0xFFFFFFFF$ ) where  $2^{24}$  would correspond to a position change of 20,000 units/second (or 1 step per 50 $\mu$ s). An alternative way to think of this is that a velocity of  $v$  represents a change in (visible) position of  $v/2^{24} * 20000$  every second. The maximum motor speed is 20000 microsteps/second.

This representation is used for current velocity, target velocity and tracking velocity.

The maximum speed is represented in the same manner as the velocity described above but negative values are not allowed.

#### 4.7.3.3 Axis acceleration

Acceleration rate is represented as an 8-bit unsigned number. The acceleration value is added to bits 8-15 of the 24-bit velocity every 2ms. e.g. an acceleration value of 1 would increment the velocity by 0x000100 every 2ms – a change of 0x01F400 every second or a 0-max in 131 seconds. A typical acceleration value of 0x1A corresponds to a velocity change of 0x001A00/2ms = 0x32C800/s which is 0 to 0xFFFFFFFF in about 5 seconds.

#### 4.7.4. Encoders

Encoder values are 16-bit unsigned number either in Gray code (raw) or binary (cooked). A 14-bit absolute encoder will map to the 14 most significant bits on the 16-bit value. The angular position will be  $n/65536 * 360^\circ$  (19.78” resolution).

To map the encoder position to the motor position, an encoder reading is remembered for the point where the motor position is zero. This is the “encoder zero position”.

Information also stored where whether the encoder is “reversed” with respect to the motor direction. i.e. if an increase in axis position results in a decrease in encoder position (in a mathematical sense) then the encoder is “reversed”. In this case, the two's complement of the cooked encoder position is used instead of the value itself.

#### 4.7.5. Gearing

The TC410 does not itself use gearing information but it can store the mechanical configuration for the purposes of providing all information required to compute actual axis angles. The 4 items available for this are:

- Motor steps/revolution (n)
- Drive type (determines the microstep size – see 4.7.1)
- Main gearing ratio (m1/d1) – usually the worm and wheel ratio
- Secondary gearing ratio (m2/d2) – the gear ratio between the motor and the worm shaft

The final full step size (in degrees) of an axis is then  $360/n * m1/d1 * m2/d2$ . The reported position step size is then values divided by 10 (for Gecko drive) or 16 (for IMS drive).

For example: For a motor with 200 steps/rev ( $n = 200$ ), a worm gear of 1:285 ( $m1=1, d1=285$ ) and secondary gearing of 16:40 ( $m2=16, d2=40$ ) has a full step size of  $360/200 * 1/285 * 16/40 = 9^\circ$ .

### 4.7.6. Joystick position

The raw joystick position is an 8-bit unsigned number with 128 nominally in the centre of its movement. The actual joystick centre position may be specified explicitly as any non-zero 8-bit number but more usefully this is set to zero which means the TC410 determines it automatically by sampling the joystick on start-up or whenever the joystick is first detected.

### 4.7.7. Joystick and Autoguider modes

The analogue input devices (joystick and autoguider) may operate in different modes as defined in section 4.8.4.1. The mode is defined by an 8-bit number defined as follows:

- bit 7 set: logarithmic mode
- bit 7 clear: linear mode

In logarithmic mode, bits 0-6 are ignored and a companding algorithm is used to map the linear deflection of the input device onto a usable speed range of some 42dB. This allows a typical speed range of 5"/second down to 1"/second.

In linear mode, the lower 6 bits define a shift value which works like a gear selector. The analogue input is treated as a 7-bit unsigned deflection (0 is stop, 127 is maximum speed). This is then mapped to the 7 most significant bits of the 24-bit velocity (as defined in in 4.7.3.2). The actual speed used is this number shifted right by the shift value.

<i>Shift value</i>	23						16									8										0
0	█	█	█	█	█	█																				
1		█	█	█	█	█	█																			
8								█	█	█	█	█	█	█												
12															█	█	█	█	█	█						
17																					█	█	█	█	█	█

*Illustration 13: Mapping of joystick linear speed ranges*

Note that only shift values in the range 0-17 are actually useful; other values will cause no perceptible motion to occur at any deflection value.

The maximum speed for any given shift value (s) is the maximum speed divided by 2<sup>s</sup>.

### 4.7.8. Temperatures

All temperatures (both reported and threshold settings) are in units of 1/10<sup>th</sup> degree C. For example 20.5°C is represented as 205 (decimal) or 0x00CD. -5.7°C is represented as -57 (decimal) or 0xFFC7.

### 4.7.9. Coordinates

As the TC410 does not use angular measurement for most of its operations, this only applies to the geographical location in section 4.8.4.1 (page 25). Since even this is not actually used by the drive the format somewhat non-rigid. For the purpose of future compatibility it is recommended that a coordinates be stored in the TC410 the preferred format of milliarcseconds. This allows angles up

to  $\pm 596^\circ$  to be stored to milliarcsecond resolution (about 30mm when used for latitude and longitude).

#### 4.7.10. Mount information

Information about the telescope mount is not needed for operation of the drive itself but some meta information can be held in the drive's non-volatile memory. This could easily be held externally by the controlling computer.

The *mount type* is simply an 8-bit number which can be used to store an enumerated value indicating the type of mount. A driver needs to know this because, for example, driving a German mount requires a different slewing algorithm from a fork mount. The following values should be used as "standard" types with custom types being enumerated from 0x10 onwards:

0. None/unknown
1. Equatorial fork
2. Alt-azimuth fork
3. German

The mount X and Y values are a pair of 32-bit numbers intended for axis alignment calibration. For example, when computing a transformation matrix from sky coordinate to telescope coordinates the direction of the primary axis can be obtained effectively providing the polar misalignment. Although these values could be used to hold this information in any form, a suggested format is the amount of rotation in milliarcseconds of a Cartesian coordinate system from the sky (where Z is the celestial pole) to the telescope (where Z is the primary axis). For a telescope intended to be polar aligned this then relates directly to a polar alignment error.

This information can be used both for axis skew correction and for helping perform polar alignment.

### 4.8. Modbus registers

#### 4.8.1. Coils

Coils are the Modbus terminology for single-bit controls (typically relays). When reading a coil, its current state is read.

Address	Description
0x0000	Power-on/standby

*Table 17: Modbus Coil registers*

#### 4.8.2. Discrete Inputs

Discrete inputs are read-only digital state information represented by a single bit.

##### 4.8.2.1 Power control

Address	Description
0x0000	Manual power-on selected (1=asserted)
0x0001	Remote power-on selected (1=asserted)
0x0002	Main motor power enabled (1=enabled)
0x0003	System fan is running (1=running)

*Table 18: Modbus Discrete Inputs - power control*

#### 4.8.2.2 Peripheral senses

Address	Description
0x0008	Joystick detected (1=detected)
0x0009	Analogue autoguider detected (1=detected)
0x000A	Axis 1 encoder detected (1=detected)
0x000B	Axis 2 encoder detected (1=detected)

*Table 19: Modbus Discrete Inputs - peripheral senses*

#### 4.8.2.3 Power supply alarms

Address	Description
0x0010	Power supply shut down alarm (1=shut down)
0x0011	Power supply over current alarm (1=alarm)
0x0012	Power supply over voltage alarm (1=alarm)
0x0013	Power supply under voltage alarm (1=alarm)

*Table 20: Modbus Discrete Inputs - power supply alarms*

#### 4.8.2.4 Limit senses

The limit senses are also available as an input register. See section 4.3.2 on page 15 for the meanings of these limits. A value of “1” for these flags means that the limit reached.

Address	Description
0x0018	Axis 1 hard positive limit
0x0019	Axis 1 hard negative limit
0x001A	Axis 2 hard positive limit
0x001B	Axis 2 hard negative limit
0x001C	Axis 1 soft positive limit
0x001D	Axis 1 soft negative limit
0x001E	Axis 2 soft positive limit
0x001F	Axis 2 soft negative limit

*Table 21: Modbus Discrete Inputs - limit senses*

### 4.8.2.5 Movement flags

The movement flags are also available as an input register. Descriptions are for “1” state.

Address	Description
0x0020	Telescope is auto-slewing to a target
0x0021	Remote control active (direct velocity control via Modbus)
0x0022	The paddle is causing movement (not supported)
0x0023	The analogue joystick is causing movement
0x0024	The analogue autoguider is causing movement
0x0025	The digital autoguider is causing movement
0x0026	Unused
0x0027	Movement by tracking

Table 22: Modbus Discrete Inputs - movement flags

### 4.8.3. Input Registers

Input registers are read-only 16-bit quantities representing enumerated states or digitized analogue data.

#### 4.8.3.1 General status

Address	Description		
0x0000	System temperature (see 4.7.8/p20)		
0x0001	Power supply current sense (milliamps)		
0x0002	Power supply voltage sense (millivolts)		
0x0003	Slew flags	Bit 0	axis 1 is slewing to target
		Bit 1	axis 1 is past deceleration point
		Bit 2	axis 1 slew direction is negative
		Bit 3	unused (0)
		Bit 4	axis 2 is slewing to target
		Bit 5	axis 2 is past deceleration point
		Bit 6	axis 2 slew direction is negative
		Bit 7	unused (0)
0x0004	Limit flags (bits 0-7 map directly to discrete inputs 0x18-0x1F). See 4.3.2/p15.	Bit 0	Axis 1 hard positive limit
		Bit 1	Axis 1 hard negative limit
		Bit 2	Axis 2 hard positive limit
		Bit 3	Axis 2 hard negative limit
		Bit 4	Axis 1 soft positive limit
		Bit 5	Axis 1 soft negative limit
		Bit 6	Axis 2 soft positive limit
		Bit 7	Axis 2 soft negative limit

Address	Description		
0x0005	Movement flags (bits 0-7 map directly to discrete inputs 0x20-0x27)	Bit 0	Telescope is auto-slewing to a target
		Bit 1	Remote control active
		Bit 2	Paddle movement
		Bit 3	Analogue joystick movement
		Bit 4	Analogue autoguider movement
		Bit 5	Digital autoguider movement
		Bit 6	Unused
		Bit 7	Tracking movement
0x0006	Power supply flags (bits 0-3 map directly to discrete inputs 0x00-0x03 and bits 4-7 map directly to discrete inputs 0x10-0x13).		
0x0007	Raw data from joystick and autoguider (digital parts)	Bit 0	Joystick button 0 (0=depressed)
		Bit 1	Joystick button 1 (0=depressed)
		Bit 2	Joystick detect (0=present)
		Bit 3	Autoguider north (0=drive)
		Bit 4	Autoguider south (0=drive)
		Bit 5	Autoguider east (0=drive)
		Bit 6	Autoguider west (0=drive)
		Bit 7	Autoguider detect (0=present)
0x0008	Raw ADC value for joystick axis 1		
0x0009	Raw ADC value for joystick axis 2		
0x000A	Raw ADC value for analogue autoguider axis 1		
0x000B	Raw ADC value for analogue autoguider axis 2		
0x000C	Signed temperature of axis 1 motor controller (see 4.7.8/p20)		
0x000D	Signed temperature of axis 2 motor controller (see 4.7.8/p20)		
0x000E	Miscellaneous flags	Bit 0	Axis 1 motor controller fan is running
		Bit 1	Axis 2 motor controller fan is running

Table 23: Modbus Input Registers - general status

#### 4.8.3.2 Axis 1 status

See section 4.7 on page 17 for information about drive types, velocities and encoders.

Address	Description
0x0010	Drive type (0=none, 1=Gecko, 2=IMS) (see 4.7.1/p17)
0x0011	Current velocity (16 most significant bits) (see 4.7.3.2/p19)
0x0012	Current velocity (16 least significant bits) (see 4.7.3.2/p19)
0x0013	Raw encoder value (Gray scale) (see 4.7.4/p19)
0x0014	Cooked encoder value (binary) (see 4.7.4/p19)

Table 24: Modbus Input Registers - axis 1 status

### 4.8.3.3 Axis 2 status

See section 4.7 on page 17 for information about velocities and encoders.

Address	Description
0x0020	Drive type (0=none, 1=Gecko, 2=IMS) (see 4.7.1/p17)
0x0021	Current velocity (16 most significant bits) (see 4.7.3.2/p19)
0x0022	Current velocity (16 least significant bits) (see 4.7.3.2/p19)
0x0023	Raw encoder value (Gray scale) (see 4.7.4/p19)
0x0024	Cooked encoder value (binary) (see 4.7.4/p19)

Table 25: Modbus Input Registers - axis 2 status

### 4.8.4. Holding Registers

Holding registers are read/write 16-bit values typically used to hold configuration information.

#### 4.8.4.1 General registers

Address	Description
0x0000	Velocity control (see 4.6.5/p17).
	Bit 0 Slew to target
	Bit 1 Remote control (via Modbus)
	Bit 2 Reserved
	Bit 3 Joystick control
	Bit 4 Analogue autoguider control
	Bit 5 Digital autoguider control
	Bit 6 Reserved
Bit 7 Tracking	
0x0001	Joystick mode (see 4.7.7/p20)
0x0002	Autoguider mode (see 4.7.7/p20)
0x0003	System fan start temperature (see 4.7.8/p20)
0x0004	System fan stop temperature (see 4.7.8/p20)
0x0005	Site latitude (most significant bits) (see 4.7.9/p20)
0x0006	Site latitude (least significant bits) (see 4.7.9/p20)
0x0007	Site longitude (most significant bits) (see 4.7.9/p20)
0x0008	Site longitude (least significant bits) (see 4.7.9/p20)
0x0009	Reserved
0x000A	Mount type (see 4.7.10/p21)
0x000B	Reserved
0x000C	Mount X hi (see 4.7.10/p21)
0x000D	Mount X lo (see 4.7.10/p21)
0x000E	Mount Y hi (see 4.7.10/p21)
0x000F	Mount Y lo (see 4.7.10/p21)

Table 26: Modbus Holding Registers - general

#### 4.8.4.2 Axis 1 configuration

Address	Description
0x0010	Tracking velocity (16 most significant bits) (see 4.7.3.2/p19)
0x0011	Tracking velocity (16 least significant bits) (see 4.7.3.2/p19)
0x0012	Negative direction soft limit (16 most significant bits only) (4.3.2/p15)
0x0013	Positive direction soft limit (16 most significant bits only) (4.3.2/p15)
0x0014	Acceleration rate (see 4.7.3.3/p19)
0x0015	Deceleration rate (see 4.7.3.3/p19)
0x0016	Maximum speed (see 4.7.3.2/p19)
0x0017	Joystick ADC centre value (see 4.7.6/p20)
0x0018	Motor phase current (see 4.7.2/p18)
0x0019	Encoder zero position (see 4.7.4/p19)
0x001A	Reserved
0x001B	Steps per motor revolution (usually 200 – see 4.7.5/p19)
0x001C	Main gear multiplier (see 4.7.5/p19)
0x001D	Main gear divisor (see 4.7.5/p19)
0x001E	Secondary gear multiplier (see 4.7.5/p19)
0x001F	Secondary gear divisor (see 4.7.5/p19)
0x0020	Park position (16 most significant bits only)

*Table 27: Modbus Holding Registers - axis 1 configuration*

#### 4.8.4.3 Axis 2 configuration

Note that these registers have the same meaning as their axis 1 counterparts.

Address	Description
0x0030	Tracking velocity (16 most significant bits) (see 4.7.3.2/p19)
0x0031	Tracking velocity (16 least significant bits) (see 4.7.3.2/p19)
0x0032	Negative direction soft limit (16 most significant bits only) (4.3.2/p15)
0x0033	Positive direction soft limit (16 most significant bits only) (4.3.2/p15)
0x0034	Acceleration rate (see 4.7.3.3/p19)
0x0035	Deceleration rate (see 4.7.3.3/p19)
0x0036	Maximum speed (see 4.7.3.2/p19)
0x0037	Joystick ADC centre value (see 4.7.6/p20)
0x0038	Motor phase current (see 4.7.2/p18)
0x0039	Encoder zero position (see 4.7.4/p19)
0x003A	Reserved
0x003B	Steps per motor revolution (usually 200 – see 4.7.5/p19)
0x003C	Main gear multiplier (see 4.7.5/p19)
0x003D	Main gear divisor (see 4.7.5/p19)
0x003E	Secondary gear multiplier (see 4.7.5/p19)
0x003F	Secondary gear divisor (see 4.7.5/p19)
0x0040	Park position (16 most significant bits only)

*Table 28: Modbus Holding Registers - axis 2 configuration*

#### 4.8.4.4 Axis 1 status and control

These are items that change during normal operation but can be updated for remote control. See section 4.6 on page 16 for an overview motion control.

Address	Description
0x0050	Position (16 most significant bits) (see 4.7.3.1/p18)
0x0051	Position (16 least significant bits) (see 4.7.3.1/p18)
0x0052	Target position (16 most significant bits) (see 4.7.3.1/p18)
0x0053	Target position (16 least significant bits) (see 4.7.3.1/p18)
0x0054	Target velocity (16 most significant bits) (see 4.7.3.2/p19)
0x0055	Target velocity (16 least significant bits) (see 4.7.3.2/p19)

*Table 29: Modbus Holding Registers - axis 1 status and control*

#### 4.8.4.5 Axis 2 status and control

See section 4.6 on page 16 for an overview motion control.

Address	Description
0x0060	Position (16 most significant bits) (see 4.7.3.1/p18 and 4.6.3/p17)
0x0061	Position (16 least significant bits) (see 4.7.3.1/p18 and 4.6.3/p17)
0x0062	Target position (16 most significant bits) (see 4.7.3.1/p18 and 4.6.1/p16)
0x0063	Target position (16 least significant bits) (see 4.7.3.1/p18 and 4.6.1/p16)
0x0064	Target velocity (16 most significant bits) (see 4.7.3.2/p19 and 4.6.2/p16)
0x0065	Target velocity (16 least significant bits) (see 4.7.3.2/p19 and 4.6.2/p16)

*Table 30: Modbus Holding Registers - axis 2 status and control*

#### 4.8.4.6 Actions

These are write-only registers that perform an action rather than setting a state. Reading from these locations is undefined.

Address	Description
0x0070	All stop (see 4.6.4/p17)
0x0071	Load factory default configuration (like factory reset but not saved)
0x0072	Save current configuration to non-volatile memory
0x0073	Load configuration from non-volatile memory

*Table 31: Modbus Holding Registers - actions*