Tritex II Parameters Manual

Identification and Usage

Exlar Corporation

8/14/2013

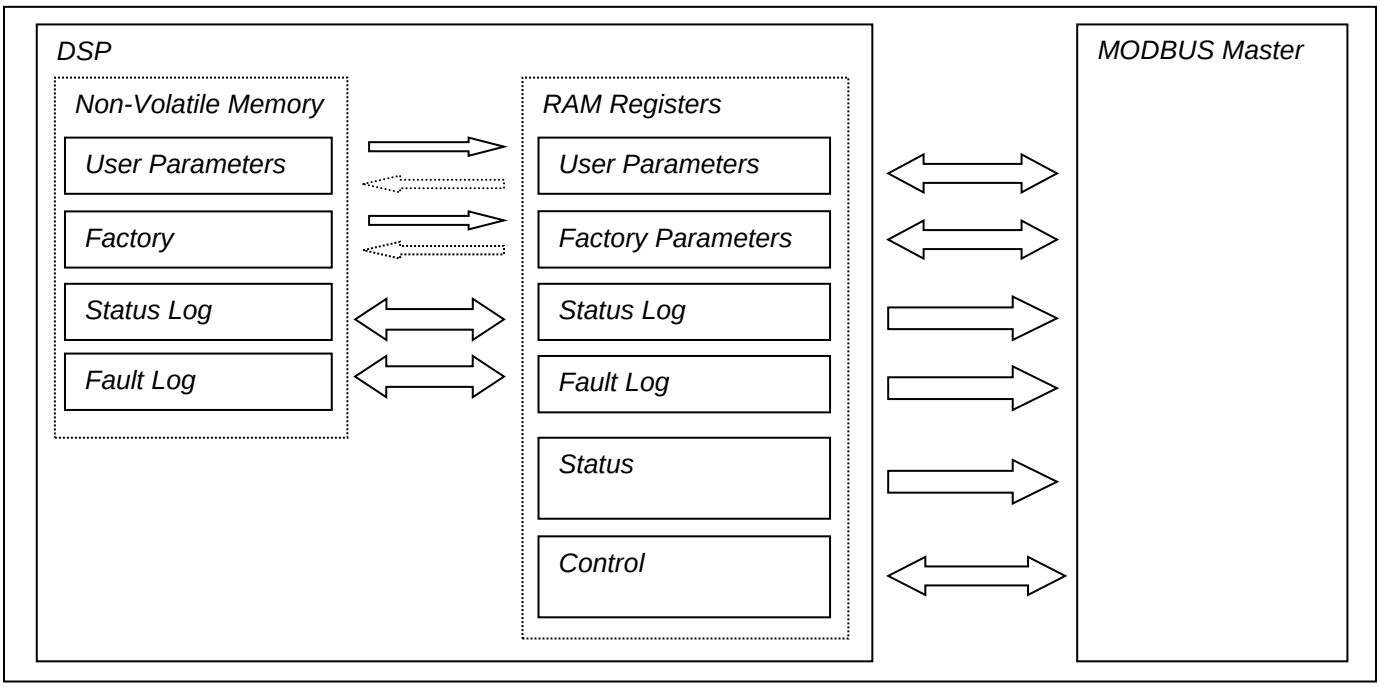
This manual provides a detailed description of the MODBUS registers used for set-up, status monitoring, and operational control of the Tritex drive. The information presented may be used by customers implementing their own MODBUS Master interface to the drive or connecting a MODBUS Master MMI (Man-Machine Interface) device to the drive. The register descriptions will also be helpful for selecting the data to be transferred between the drive and a PLC (Programmable Logic Controller) or other device using the EtherNet/IP protocol.

Table of Contents

Memory Organization

In general, all *Tritex* register data may be considered to exist in *RAM* (*Random Access Memory*) where it is available for reading, writing, or both reading and writing. (Internally, some registers may be mapped directly to *ROM (Read-Only Memory)* or *DSP* registers or other areas but these exceptions are transparent to the user.) Within the *MODBUS* protocol, registers that are read-only are referred to as *INPUT* registers and registers that are both readable and writable are referred to as *HOLDING* registers.

The drive maintains parameter and other internally logged data in non-volatile memory so that the data values are saved between power-ups. During start-up initialization, all non-volatile memory data is copied to *RAM* where it is made available to the *MODBUS* interface. After start-up, all parameter *RAM* blocks will have been initialized from the data in their non-volatile memory image block. The non-volatile images for the *Status Log* and *Fault Log* are automatically updated as necessary when the *RAM* data changes. The writing of parameter register data, however, modifies only the current values in *RAM*. Future power-ups will again initialize the *RAM* parameter blocks to their original values retained in non-volatile memory. The copying (saving) of parameter *RAM* blocks to non-volatile memory is carried out only when specified through direct control commands.



User Parameters

User parameters are stored as a block in non-volatile memory. The block contains a *CRC* (*Cyclic Redundancy Checksum*) word to guarantee data integrity. At power-up, the user parameter block is validated and copied to its runtime location in *RAM* where all parameters are available for both reading and writing through their individual *MODBUS* registers.

Identification and Configuration

Identification / Configuration Parameters Register Table

ID	Name	Type*	Description
5000	DriveName	STR16	User drive name
5100	Options	FLAGS	Configuration option flags
5101	PowerUpDelay	UINT16	Power-up delay [0.01 SEC]

See Appendix A for details on data Types

DriveName
DriveName is a sixteen *ASCII* character string used only for display purposes. It is available to the user to provide a descriptive name for the drive. *DriveName* is displayed for informational purposes in the *Tritex* user interface software and in the interface software's network connection status information.

Options
Miscellaneous user configuration options are enabled through bit flags of the *Options* word as specified by the following bitmap table.

							STUP	**DMD**	**PLPCT**	**PLP**	**PLM**	**DIR**		**TE**	**AE**
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

AE *Auto Enable*
If *Auto Enable* is set, the drive continually forces the *Enable Maintained* input event active in the *InputEvents.Mode* register. The drive enabling action is exactly the same as it would be if the *Enable Maintained* input event were assigned to a continually active digital input.

TE *Auto Teach Enable*
If *Auto Teach Enable* is set, the drive continually forces the *Teach Enable* input event active in the *InputEvents.Mode* register. The drive's teach enabling functionality will be exactly the same as it would be if the *Teach Enable* input event were assigned to a continually active digital input.

The *TE* option eliminates the requirement for a *Teach Enable* input but should be used with caution since the individual move teach functionality is no longer safeguarded by the additional requirement of first activating the *Teach Enable*.

DIR *Reverse Drive Polarity*
The *DIR* option reverses the internal direction polarities of drive current, velocity, and distance. By convention, the drive always equates positive current with positive velocity and positive direction. If the *DIR* flag is set, this positive motion is defined to be counter-clockwise rotor movement for rotoary actuators (as viewed looking towards the actuator from the shaft end), or retract shaft motion for linear actuators. If the *DIR* flag is not set (default), positive motion is defined as clockwise rotor movement or linear shaft extension.

If *DIR* is modified, the *Homed* output status event will be cleared since the definition of all absolute positions has also effectively been modified. The drive may need to be re-homed to establish a new absolute position reference frame before motion is allowed.

PLM *Position Limit Minus Enable*

PLP *Position Limit Plus Enable*
The *PLM* and *PLP* options enable the negative and positive *Position Limits*, respectively. Position limit functionality is described in the *Position Limits* section.

PLPCT *Position Limit Percent Enable*
The *PLPCT* option enables the drive's position limits to be calculated automatically as a percent of the *Analog Position Mode* range of position. Position limit functionality is described in the *Position Limits* section.

DMD *Dedicated Move Disable*
The *Dedicated Move Disable* option will set the *EMOVE* flag in the *Disables* register (forcing the drive to become disabled) when a *Dedicated Move* terminates and the *At Dedicated Move Position* output event becomes active.

STUP *Startup Required*
The STUP option requires that *Startup Complete* is active before allowing operation in any mode. No motion commands other than *Startup* will be acted upon.

PowerupDelay
The *PowerupDelay* register specifies a time delay period at start-up during which the drive will remain idle before initializing the system. During this idle time the digital outputs will not be driven and the communication channels will not be available. Once the delay time has expired, the drive will execute its normal start-up routine. *PowerupDelay* is specified in 0.01 second units.

Drive Limits

Registers described in this section control various global parameters used by the drive at runtime. Parameter registers specific to the drive's position limits are covered separately in the *Position Limits* section.

Drive Limits Parameters Register Table

ID	Name	Type*	Description
5108	Ipeak	UCUR16	User current command limit [9.7 AMPS]
5110	StopAccel	UACC32	Stop acceleration [12.20 RPS/S]
5112	InPositionWindow	UPOS32	In position window width [16.16 REVS]
5114	MaxFollowingError	UPOS32	Position error limit [16.16 REVS]
5116	InPositionTime	UINT16	Time to in-position [ms]
5117	MaxFollowingErrorTime	UINT16	Time to in-position [0.01 SEC]
5129	InCurrentLimitTime	UINT16	In current limit hysteresis time [0.1 SEC]

****See Appendix A for details on data Types***

Ipeak
The maximum current command allowed under normal operation is set to the minimum of the factory limits *Ipeak* parameter and the user's *Ipeak* parameter. The maximum allowable current command may be lowered during *Host Mode* operation to the value specified in *Host.Current* and also when operating outside of the drive's position limits, moving, or jogging if specific current limit options are enabled.

StopAccel
The drive uses the acceleration limit specified in *StopAccel* when decelerating to zero velocity due to an active *STOP* input event. *StopAccel* is also used when aborting motion for other internal reasons such as an active hard fault which has been selected to stop the drive without disabling. *StopAccel* should normally be set to the highest value that will safely stop motion without undue stress on the drive or machine mechanics.

InPositionWindow
InPositionWindow sets the maximum position error allowed at zero velocity before the *In Position* output status event can be activated. The *In Position* output status event will be active whenever command velocity is zero, the absolute value of position error (command position – feedback position) is less than *InPositionWindow*, and the *InPositionTime* criteria (below) has been met.

During *Analog Position* and *Analog Velocity* modes of operation, the velocity command signal will usually have some small amount of *noise* even when the drive's position appears to be constant. The non-zero velocity command may keep *the In Position* output status event from activating.

Note that since the absolute value of position error is used, the 'window' is actually twice the position width specified by *InPositionWindow.*

InPositionTime
InPositionTime sets the hysteresis time for the *In Position* output status event. The time is specified in milliseconds. An internal countdown timer is continually reinitialized to the *InPositionTime* value while the command velocity is non-zero or the absolute value of position error is greater than InPositionWindow (above). The *In Position* output status event will be active whenever the timer value is zero.

MaxFollowingError
MaxFollowingError sets the maximum position error allowed before a *Following Error* fault is activated. Whenever the absolute value of position error (command position – feedback position) is greater than *MaxFollowingError* and the *MaxFollowingErrorTime* criteria (below) has been met the *Following Error* fault will be active.

When operating in one of the current command modes, the positioning loop is not active and position command is continually set to the value of position feedback. Since position error is always zero during current command modes, a *Following Error FAULT* will never be generated.

MaxFollowingErrorTime
MaxFollowingErrorTime sets the hysteresis time for the *Following Error* fault. The time is specified in hundredths of a second. An internal countdown timer is continually reinitialized to the *MaxFollowingErrorTime* value while the absolute value of position error remains less than *MaxFollowingError*. A *Following Error* fault will be activated if the timer values counts down to zero (i.e. the absolute value of position error has remained greater then *MaxFollowingError* for at least *MaxFollowingErrorTime*).

InCurrentLimitTime
InCurrentLimitTime sets the amount of time the drive must be current limiting before triggering the *In Current Limit* output status event.

Fault Control

The registers described in this section are used to control the drive's behavior upon the occurrence of a *FAULT* event. The specific faults that are monitored by the drive are described in the *FAULT* data type. ***(See Appendix B for the definition of the FAULT data type bitmap.)***

Fault Control Register Table

ID	Name	Type*	Description
5102	FaultDisables	FAULT	Disabling FAULTs
5103	FaultWarnings	FAULT	Warning FAULTs
5104	FaultStop	FAULT	Stopping FAULTs
5105	FaultDedicatedMove	FAULT	Dedicated move FAULTs
5130	FaultLogFaults	FAULT	Fault log FAULT enables
5139	FaultDisables2	FAULT	Disabling FAULTs for 2nd 16
5140	FaultWarnings2	FAULT	Warning FAULTs for 2nd 16
5141	FaultStop2	FAULT	Stopping FAULTs for 2nd 16
5142	FaultDedicatedMove2	FAULT	Dedicated move FAULTs for 2nd 16
5138	FaultLogFaults2	FAULT	Fault log FAULT enables for 2nd 16
5132	FaultResetDelay	UINT16	Fault auto-reset delay [0.01 SEC]
5133	FaultLogDelay	UINT16	Delay before logging faults [0.01 SEC]
5143	Low_VoltageDC_Warn_On	INT16	Voltage at which Low voltage warning goes on
5144	Low_VoltageDC_Warn_Off	INT16	Voltage at which Low voltage warning goes off

See Appendix A for details on data Types

FaultDisables and FaultDisables2
Faults selected in *FaultDisables/FaultDisables2* will cause the drive to disable when the fault occurs. The disabling may be delayed in order for the drive to stop under controlled conditions if the fault is also set in *FaultStop*. Faults selected in *FaultDisables/FaultDisables2* are also referred to as *'hard faults'* and will require a rising edge of the *Reset Faults*, *Enable Momentary*, or *Enable Maintained* input events to reset the fault and re-enable the drive. The *Faulted* output status event will be active while a hard fault is active.

CAUTION
Removing faults from the *FaultDisables* register may lead to permanent damage of the drive or actuator. It is up to the user to take appropriate action on any faults that do not automatically disable the drive.

FaultWarnings and FaultWarnings2
Faults selected in *FaultWarnings/FaultWarnings2* will cause the *Warning* output status event to become active. The warning indication will be automatically removed if the fault later becomes inactive. The drive is not disabled (unless the fault is also selected in *FaultDisables/FaultDisables2*) and continues normal operation. Faults selected in *FaultWarnings* are also referred to as *'soft faults'*. Soft faults should be monitored by the user so that appropriate action may be taken.

FaultStop and FaultStop2
Faults selected in *FaultStop/FaultStop2* will cause the drive to do a controlled stop of all motion. If the drive is operating in a current command mode it will first be forced into an internal position control mode. Drive motion is forced to zero velocity using the acceleration specified in the *StopAcceleration* register. Once motion has been stopped, the drive will disable if the fault has also been selected in the *FaultDisables/FaultDisables2* register. If the fault has not been selected in *FaultDisables/FaultDisables2*, the drive will hold position and remain enabled. While the drive remains enabled with the fault pending it will continually attempt an automatic fault reset at the rate specified in the *FaultResetDelay* register. The fault may also be reset at any time with the *Reset Faults* input event.

FaultDedicatedMove and FaultDedicatedMove2
Faults selected in *FaultDedicatedMove/FaultDedicatedMove2* will automatically execute the *Dedicated Move* when the fault occurs. If the drive is operating in a current command mode it will first be forced into the internal position control mode. Once the *Dedicated Move* has completed, the drive will disable if the fault has also been selected in the *FaultDisables/FaultDisables2* registers. If the fault has not been selected in *FaultDisables/FaultDisables2*, the drive will hold position and remain enabled. While the drive remains enabled with the fault pending it will continually attempt an automatic fault reset at the rate specified in the *FaultResetDelay* register. The fault may also be reset at any time with the *Reset Faults* input event.

FaultLogFaults and FaultLogFaults2
Faults selected in *FaultLogFaults/FaultLogFaults2* will be logged in the Fault Log when they occur. Faults may be selected for logging even if they are not selected in any other fault action register. The logging of a new fault is delayed by time period specified in *FaultLogDelay* to avoid filling the fault log with faults that may occur regularly when power is removed.

FaultLogDelay
FaultLogDelay sets the time delay between the occurrence of a fault and logging of the fault in the *Fault Log*. *FaultLogDelay is specified in 0.01 second units.*

FaultResetDelay
The *FaultResetDelay* sets the rate at which automatic fault resets are attempted for faults that have been selected in the *FaultStop* or *FaultDedicatedMove* registers but are not selected in the *FaultDisables* register. *FaultResetDelay* is specified in 0.01 second units. The automatic fault reset may be appropriate to recover automatically from an analog input *Loss of Signal* when the signal returns to the valid operating range or to continue motion execution using *Move Maintained Initiate* input events.

Low_VoltageDC_Warn_On
Low_VoltageDC_Warn_On is the level at which the voltage must drop in order for the Low DC warning to go on. By default, it is set to zero which disables it.

Low_VoltageDC_Warn_Off
Low_VoltageDC_Warn_Off is the level at which the voltage must rise to in order for the warning to go away. For proper operation, this value must be greater than *Low_VoltageDC_Warn_Off* and the difference should be great enough to prevent the warning from flickering on and off. By default, it is set to zero.

Position Limits

Position limits are software monitored travel limits with special features for control outside of the travel limits. The limits may be individually enabled and, once enabled, will be monitored in all operating command modes. The positioning algorithms in the controller will *'look ahead'* to anticipate a position limit being reached and reduce speed, if necessary. Once a position limit has been passed, reduced current commands may be put into effect for various application control options.

In addition to the parameters listed in this section, the *Configuration.Options* parameter contains the following flags that govern the behavior of the drive's position limits.

PLM - Position Limit Minus Enable
PLP - Position Limit Plus Enable
PLPCT - Position Limit Percent Enable

Position Limits Registers Table

ID	Name	Type*	Description
5118	PlimitMinus	POS32	S/W (-) position limit [16.16 REVS]
5120	PlimitPlus	POS32	S/W (+) position limit [16.16 REVS]
5122	PlimitPercentMinus	UINT16	S/W (-) position limit percent [1.15 %]
5123	PlimitPercentPlus	UINT16	S/W (+) position limit percent [1.15 %]
5124	PlimitVelocity	UVEL32	Position limit velocity limit [8.24 RPS]
5126	PlimitIfoldback	UCUR16	Position limit foldback current [9.7 AMPS]
5127	PlimitIpeak	UCUR16	Position limit peak (seating) current [9.7 AMPS]
5128	PlimitIpeakTime	UINT16	Position limit seating time current [ms]

****See Appendix A for details on data Types***

PlimitMinus
PlimitPlus
The *PlimitMinus* and *PlimitPlus* set the negative and positive position limits, respectively. The drive always attempts to reduce speed to the position limit velocity limit before moving on to or past an enabled position limit. These registers are not used if current limit positions have been selected to be a percentage of the analog input range.

PlimitPercentMinus
PlimitPercentPlus
These registers specify the position limits as a percent of the analog input range. The values are only used if the percentage of range option has been selected, in which case the *PlimitMinus* and *PlimitPlus* values are ignored. Specifying positions as a percentage of range avoids having to change the limits when the position range of the analog input is modified for *Analog Position* command mode.

PlimitVelocity
The *PlimitVelocity* register sets the maximum velocity command allowed while outside of the position limits. A zero value will force the drive to a stop until motion is commanded in a direction opposite that of the active limit. A non-zero value will force to drive to continue in the direction of the active limit with the position limit foldback current in effect.

The PlimitVelocity register will not be used if the operating mode is a current control mode while the limit is active.

PlimitIfoldback
The *Plimitfoldback* register sets the maximum current command allowed while outside of the position limits. The reduced current command limits are in effect only for motion in the direction that activated the position limit – full current is available to move in the direction opposite that of the limit (and to decelerate the drive towards zero speed).

PlimitIpeak
PlimitIpeakTime
The *PlimitIpeak* and *PlimitIpeakTime* registers set a maximum current and time to command this maximum current after a current limit is hit. A higher momentary peak peak may be useful, for example, as a momentary *seating* current. Once the specified time period has elapsed, the maximum allowable current command will drop back to the foldback current value. If a momentary limit is not required it may be set to the same value as the foldback current.

Position limits can only become active if they have been enabled and the absolute position reference frame has been established.

Ethernet Parameters

Ethernet parameter registers are used by a *Tritex* drive that has an optional *Ethernet* communications module installed. Most *Ethernet* module parameters are configured at the factory when the option board is installed and are preconfigured to control the internal *MODBUS Master* serial channel used between the option board and the drive. The few parameters that require customer configuration are described in this section.

Ethernet Parameters Register Table

ID	Name	Type*	Description
5150	IP	UINT32	IP (Internet Protocol) Address
5152	Subnet	UINT32	Subnet Address
5154	Gateway	UINT32	Gateway Address

See Appendix A for details on data Types

IP
Subnet
Gateway
These registers specify the *IP*, *Subnet*, and *Gateway* addresses used when an *Ethernet* option board is present. The *Ethernet MODBUS Master* reads the values at start-up, if needed.

MODBUS Parameter Registers

The *MODBUS* registers configure the *RS485* serial communications channel.

MODBUS Parameters Register Table

ID	Name	Type*	Description
5300	Flags	FLAGS	Modbus Serial Channel A options
5301	AxisId	UINT16	Modbus Serial Channel A axis identifier
5302	Baud	BAUD	Modbus Serial Channel A baud identifier
5303	RxDelay	UINT16	Modbus Serial Channel A extra RX delay
5304	TxDelay	UINT16	Modbus Serial Channel A extra TX delay

See Appendix A for details on data Types

Flags

The *Flags* register selects the hardware framing options as specified in the following bitmap table.

													S2	**NP**	**OP**
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

OP - Parity selection
0 = EVEN parity
1 = ODD parity

NP - Parity bit enabling
0 = parity ENABLED
1 = parity DISABLED

When parity is DISABLED, the serial channel will automatically be set for two stop bits to keep the standard *MODBUS* frame at ten bits (1 start, 8 data, 2 stop), regardless of the **S2** option selection.

S2 - Stop bit selection
0 = one stop bit
1 = two stop bits

When two stop bits are selected, the serial channel will always transmit two stop bits which will result in an 11-bit (non-standard *MODBUS*) frame if parity is also enabled. This option may normally be left at 0 (one stop bit), since an extra stop bit is automatically added when parity is disabled.

AxisId

The *AxisId* specifies the *MODBUS* axis identifier used in *MODBUS* commands to and from the drive.

Baud
The value set for the serial channel's baud rate must be a value from the *BAUD Enumerations Table*. Baud rates above 38400 bps are specified with a value equal to 1/150th of the desired baud rate.

BAUD Enumerations Table

Value	Baud (bps)	Description
4800	4800	baud = value
9600	9600	
19200	19200	
38400	38400	
32	4800	baud = value * 150
64	9600	
128	19200	
256	38400	
384	57600	
512	76800	
625	93750	
768	115200	
1250	187500	
1536	230400	
3072	460800	
6144	921600	

RxDelay
Standard *MODBUS* communication protocol requires that serial characters be transmitted back-to-back and specifies that a frame idle time of 1.5 character times is an end-of-command indication. *RxDelay* may be used to extend the frame idle time required before assuming an end-of-command condition. Extra delay may be useful for a slow *MODBUS Master* that cannot guarantee back-to-back character transmission. The value is specified in milliseconds.

TxDelay
Standard *MODBUS* communication protocol requires a minimum 3.5 character time frame idle delay between transmissions. *TxDelay* may be used to extend the frame idle time between the receipt of the command and the transmission of the response. Extra delay may be useful for a slow *MODBUS Master* or an *RS485* device that requires extra time to tri-state the bus after transmitting. The value is specified in milliseconds.

The *RxDelay* and *TxDelay* registers (unlike *Flags*, *Baud* and *AxisID*) become effective as soon as they are modified. This allows the new values to be tested before committing them to non-volatile memory. It is recommended that changes to *RxDelay* and *TxDelay* be tested before saving parameters to avoid communication issues that may occur at start-up with untested parameters.

Tuning

The drive's proprietary motion control algorithms have been designed to minimize the user's tuning requirements and provide for adequate performance with default tuning parameter values. The overall inertia (*KJ*) may need adjustment for larger system inertia variations. If necessary, the proportional (*KP*) and integral (*KI*) terms may then be individually adjusted for optimal performance. Feed-forward (*KFF*) and damping (*KD*) terms may normally be ignored and left at their default settings of zero. The drive will recalculate all necessary internal run-time gain scaling factors as necessary when changes are made to any tuning parameter.

Tuning Parameters Register Table

ID	Name	Type*	Description
5400	KJ	UINT16	Inertia gain [6.10]
5402	KP	UINT16	Position loop bandwidth [8.8 1/s]
5403	KI	UINT16	Velocity integral time constant [8.8 1/s]
5404	KFF	UINT16	Feed forward velocity scale [0.16]
5405	KD	UINT16	Velocity damping [0.16]

See Appendix A for details on data Types

KJ

KJ controls the overall inertia gain of the system. In general, default tuning values for *KP* and *KI* should provide reasonable loop performance and the *KJ* term should be adjusted first for the system inertia. Changes to *KJ* will linearly affect the internal runtime values for both the proportional (*KP*) and integral (*KI*) terms.

KP

KP controls the overall response of the position loop. Larger values increase will increase positional stiffness and reduce positional error. *KP* values that are too high may cause instability.

KI

KI controls the rate of integration of the positional error. Higher *KI* values will add stiffness and improve positional holding accuracy at a stop. *KI* values that are too high may cause overshoot and instability.

KFF

KFF provides open loop feed-forward velocity that sums with the velocity command generated from the positional error signal. *KFF* tends to reduce position error and may be a better alternative than higher *KP* and *KI* settings which may cause instability.

KD

KD is used to reduce velocity gain proportional to the speed. Increasing *KD* will smooth out higher velocity instabilities but will increase run-time position error. *KD* should not normally be used in combination with *KI* (integral gain) as the two terms may tend to fight each other.

Command Mode

The command mode parameter registers select the operational command modes available during normal drive operation. Normal operation, for the purposes of command mode selection, means that the drive is enabled, not stopped or in a fault condition, homing requirements (if any) have been satisfied, and no host controller has overridden the command mode by specifying a direct command mode of its own.

The specific operating modes available to the drive are defined by the *OPMODE* data type. ***(See Appendix B for the definition of the OPMODE data type and its enumeration values.)***

Operation Mode Parameters Register Table

ID	Name	Type*	Description
5106	DefaultCommandMode	OPMODE	Main command mode selection
5107	AltCommandMode	OPMODE	Alternate command mode selection

***See Appendix A for details on data Types**

DefaultCommandMode
The operational command mode specified by the *DefaultCommandMode* register will be in effect during normal drive operation while the *Alternate Mode* input event is inactive. While the *DefaultCommandMode* register is being used for the source of the drive's operational command mode, the *Default Mode* output status event will be active.

AltCommandMode
The operational command mode specified by the *AltCommandMode* register will be used instead of the *DefaultCommandMode* while the *Alternate Mode* input event is active during normal drive operation. While the *AlttCommandMode* register is being used for the source of the drive's operational command mode, the *Alternate Mode* output status event will be active.

The command operating mode actually in effect at any moment is available in the *Command.Mode* status register.

Jog Control

The *Jog* parameters control the profile motion commanded by the drive while the *Jog Positive* or *Jog Negative* input events are active. Active jog input events are normally accepted when the command mode is *Digital Inputs* and other move motion is inactive. The jog input events may optionally be allowed to override a command mode's active motion through the *DMO* and *AMO* option flags.

Jog Parameters Register Table

ID	Name	Type*	Description
6020	Options	FLAGS	Option flags
6022	SlowVelocity	UVEL32	Jog command target velocity 1
6024	FastVelocity	UVEL32	Jog command target velocity 2
6026	Acceleration	UACC32	Jog acceleration limit

See Appendix A for details on data Types

Options

General jogging options are enabled through bit flags of the *Options* word as specified by the following bitmap table.

ILIMIT													**AMO**	**DMO**
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DMO

The *DMO* option allows the *Jog Positive* and *Jog Negative* input events to override the command mode in effect when the *Default Operating Mode* is active. Jog events will not be given priority over an active *Home* or *Dedicated Move*.

AMO

The *AMO* option allows the *Jog Positive* and *Jog Negative* input events to override the command mode in effect when the *Alternate Operating Mode* is active. Jog events will not be given priority over an active *Home* or *Dedicated Move*.

ILIMIT

The *ILIMIT* option may be set to limit the maximum current command allowed while jog motion is active to the value for the *PlimitIfoldback* register (see *Position Limits*). Limiting the current command while jogging may be useful when the drive may be jogged into a hard stop.

SlowVelocity

The *SlowVelocity* register specifies the absolute value of the velocity commanded while jog is active and the *Jog Fast* input event is inactive. The drive will ramp the velocity command to the *SlowVelocity* using the *Acceleration* rate.

FastVelocity

The *FastVelocity r*egister specifies the absolute value of the velocity commanded while jog is active and the *Jog Fast* input event is active. The drive will ramp the velocity command to the *FastVelocity* using the *Acceleration* rate.

The 'slow' and 'fast' designations are simply naming conveniences to distinguish between the two velocities. The default values and the *Tritex* user interface software will normally have the lower speed value set in the 'slow' parameter. There is no actual requirement, however, that the 'slow' velocity be less than the 'fast' velocity and these registers may be set to any velocity values. The 'fast' and 'slow' velocities are simply the command velocity targets set for jog motion when the *Jog Fast* input event is active or inactive, respectively.

Acceleration

The *Acceleration* register specifies the absolute value of the acceleration rate used to achieve the velocity commanded while jog mode is active. The *Acceleration* rate is also used to achieve zero velocity when the jog input events become inactive while jogging. Once the jog input events become inactive, jog mode remains active until the command velocity has been ramped to zero.

***Home operation**:*
The Home function can be initiated from the following methods:
1. The Home Input function from either an assigned digital input
3. Home button on the Control Page
4. From a host using Modus address control.
5. Automatically Home when the actuator is enabled, if the Homed status in not active.

The motion parameters for the *Home* move are defined by the *MOVE* data type.
(See Appendix B for the definition of the MOVE data type registers and MOVE profile motion.)

Home Parameters Register Table

ID	Name	Type*	Description
6000	Options	FLAGS	Home option flags
6002	Position1	POS32	Home reference position 1
6004	Position2	POS32	Home reference position 2
6406	Move	MOVE	Home motion parameters

****See Appendix A for details on data Types***

Options
Miscellaneous general homing options are enabled through bit flags of the *Options* register.

AUTO		**OM_ALT**	**OM_DEF**												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

AUTO
The *AUTO* option enables automatic *Home* move execution whenever the drive is enabled and the absolute position reference frame has not yet been established.

OM_ALT
The *OM_ALT* option requires the absolute position reference frame to have been established before the *Alternate Operating Mode* is allowed to become active.

OM_DEF
The *OM_DEF* option requires the absolute position reference frame to have been established before the *Default Operating Mode* is allowed to become active.

Position1
Position1 defines the absolute reference frame position set upon completion of a *MOVE* with the *REF1* flag set in the *Move.Options* parameter.

Position2
Position2 defines the absolute reference frame position set upon completion of a *MOVE* with the *REF2* flag set in the *Move.Options* parameter.

Move
The *Move* registers define the profile motion for the home in the common move format defined by the *MOVE* data type.
(See Appendix B for the definition of the MOVE data type and specific parameter IDs.)

Dedicated Move Control

The *Dedicated Move* is designed to provide a method, available from any mode, to move to a specified position. The *Dedicated Move* has priority in all operating modes and will override any active motion other than an active *Home* move. The *Dedicated Move* is assumed to be set-up as a move to absolute position and the absolute position reference frame must be defined before a *Dedicated Move* will be allowed to execute. The motion parameters for the Dedicated Move are defined by the *MOVE* data type.

(See Appendix B for the definition of the MOVE data type registers and MOVE profile motion.)

In addition to the dedicated move registers described below, the operation of the dedicated move is also controlled by the *DMD* option in the *Configuration.Options* register which may be set to force the drive to become disabled upon completion of the move.

Dedicated Move Parameters Register Table

ID	Name	Type*	Description
6388	Emove	MOVE	Dedicated (emergency) move motion motion parameters

See Appendix A for details on data Types

Emove

The *Emove* registers define the profile motion for the dedicated move in the common move format defined by the *MOVE* data type. The move parameters for the dedicated move will normally be set-up to execute a move to an absolute position.

Move Control

The *Tritex II* is capable of storing and controlling up to 16 move profiles. Moves may be execute through the *Move Maintained* and *Move Momentary* input events when the drive's operational command mode is Digital I/O. The motion parameters for a move is defined by the *MOVE* data type.

(See Appendix B for the definition of the MOVE data type registers and MOVE profile motion.)

Move Parameters Register Table

ID	Name	Type*	Description
6100	Move.0	MOVE	Move 0 motion parameters
6118	Move.1	MOVE	Move 1 motion parameters
6136	Move.2	MOVE	Move 2 motion parameters
6154	Move.3	MOVE	Move 3 motion parameters
6172	Move.4	MOVE	Move 4 motion parameters
6190	Move.5	MOVE	Move 5 motion parameters
6208	Move.6	MOVE	Move 6 motion parameters
6226	Move.7	MOVE	Move 7 motion parameters
6244	Move.8	MOVE	Move 8 motion parameters
6262	Move.9	MOVE	Move 9 motion parameters
6280	Move.10	MOVE	Move 10 motion parameters
6298	Move.11	MOVE	Move 11 motion parameters
6316	Move.12	MOVE	Move 12 motion parameters
6334	Move.13	MOVE	Move 13 motion parameters
6352	Move.14	MOVE	Move 14 motion parameters
6370	Move.15	MOVE	Move 15 motion parameters

***See Appendix A for details on data Types**

Analog Motion Control

An analog input may be set up as a position command, a velocity command or a current Command. Each command mode uses its own set of parameters to specify the desired motion.

Analog Position Control

Analog Position Control provides position control proportional to the analog input value, the input is continuously updated and scaled to provide an Analog Command position, if the *Analog Position Control* is active from Default, Alternate or Host Modes the drive will position to the command while following the Velocity and Acceleration limits as specified on the page. The Acceleration Limit is also used as the deceleration limit.

Analog Position Control Parameters Register Table

ID	Name	Type*	Description
7101	Channel	UINT16	Analog input channel
7102	Minimum	POS32	Position (minimum)
7104	Maximum	POS32	Position (maximum)
7106	Velocity	UVEL32	Positioning velocity limit
7108	Acceleration	UACC32	Positioning acceleration limit
7188	ModbusCtrl	UINT16	Modbus Control of position

See Appendix A for details on data Types

Channel
The *Channel* register selects the analog input to be used for monitoring the position command. It can also be set to be controlled via Modbus address 7190.

Minimum
Maximum
The *Minimum* and *Maximum* registers specify the target absolute positions corresponding to analog input's minimum and maximum values, respectively. These registers define the absolute position range of the analog control input.

Velocity
The *Velocity* register specifies the absolute value of the maximum command velocity used during the profile.

Acceleration
The *Acceleration* register specifies the absolute value of the maximum command acceleration rate used during the profile.

Modbus Ctrl
The *Modbus Ctrl* register specifies the velocity if the Channel is set to '2'.

Analog Velocity Control

Analog Velocity Control provides velocity control proportional to an analog input value. The input is continuously updated and scaled to provide an Analog Command velocity. If the *Analog Velocity Control* is active from Default, Alternate or Host Modes the drive will operate as velocity control following the Acceleration limit as specified on the page. The Acceleration Limit is also used as the deceleration limit. In Analog Velocity Mode, position control is the responsibility of the users control system.

Analog Velocity Control Parameters Register Table

ID	Name	Type	Description
7117	Channel	UINT16	Analog input channel
7118	Minimum	VEL32	Velocity (minimum)
7120	Maximum	VEL32	Velocity (maximum)

7102	Acceleration	UACC32	Acceleration limit
7189	Modbus Ctrl	UINT16	Analog Velocity

Channel
The *Channel* register selects the analog input to be used for monitoring the velocity command. It can also be set to be controlled via Modbus address 7189.

Minimum
Maximum
The *Minimum* and *Maximum* registers specify the command velocities corresponding to analog input's minimum and maximum values, respectively. These registers define the speed range of the analog control input.

Acceleration
The *Acceleration* register specifies the absolute value of the maximum command acceleration rate used to ramp to the analog signal's commanded velocity.

Modbus Ctrl
The *Modbus Ctrl* register specifies the velocity if the Channel is set to '2'.

Analog Current Control

Analog Current Control provides a current command proportional to an analog input value. If the Analog Current Control is active from Default, Alternate or Host Modes the drive will operate as torque / force control. In *Analog Current* control velocity and position control are the responsibility of the users control system.

Analog Torque Control Parameters Register Table

ID	Name	Type*	Description
7133	Channel	UINT16	Analog input channel
7134	Minimum	CUR16	Current (minimum)
7135	Maximum	CUR16	Current (maximum)
7190	Modbus Ctrl	UINT16	Current

****See Appendix A for details on data Types***

Channel
The *Channel* register selects the analog input to be used for monitoring the current command. It can also be set to be controlled via Modbus address 7190

Minimum
Maximum
The *Minimum* and *Maximum* registers specify the target current commands corresponding to analog input's minimum and maximum values, respectively. These registers define the current command range of the analog control input.

Modbus Ctrl
The *Modbus Ctrl* register specifies the current if the Channel is set to '2'.

Velocity Override

VelocityOverride Allows the velocity of a move to be controlled via an analog input or Modbus address 7180. If a move has enabled velocity override, it's velocity will be equal to the value set times the percent of range at which the analog input or Modbus address is currently at.

Analog Torque Control Parameters Register Table

ID	Name	Type*	Description
7185	Channel	UINT16	Analog input channel
7186	Minimum	INT16	Minimum percentage
7187	Maximum	INT16	Maximum percentage
7180	Modbus Ctrl	UINT16	Modbus override input

See Appendix A for details on data Types

Channel
The *Channel* register selects the analog input to be used for setting the velocity override. It can also be set to be controlled via Modbus address 7180

Minimum
Maximum
The *Minimum* and *Maximum* registers specify the minimum and maximum percentage that the override value can attain.

Modbus Ctrl
The *Modbus Ctrl* register specifies the override level if the Channel is set to '2'.

Digital Input Assignments

The drive is controlled through an *input event* system which is organized into input event groups. Each event group may contain up to sixteen specific events within the group. Drive control is achieved through activation of input events that are assigned to the eight hardware digital inputs of the drive. Each digital input may be individually mapped to an input event by assigning the input event group and the specific event bitmap of the input event(s) within the group that is to be activated when the digital input is activated. The input event bitmaps for each group are, for convenience, given their own data type since the maps are used for the definition of multiple registers (input event bitmaps are used for digital input assignment, direct control, and status registers).

(See Appendix B for the definition of the IEG_XXX input event group data type bitmaps.)

Digital Input Parameters Register Table

ID	Name	Type*	Description
7000	Input.Polarities	FLAGS	Input polarity bitmap
7002	Input1.GroupMap	IEG_XXX	Input 1 group bitmap
7003	Input1.Group	IGROUP	Input 1 group
7004	Input2.GroupMap	IEG_XXX	Input 2 group bitmap
7005	Input2.Group	IGROUP	Input 2 group
7006	Input3.GroupMap	IEG_XXX	Input 3 group bitmap
7007	Input3.Group	IGROUP	Input 3 group
7008	Input4.GroupMap	IEG_XXX	Input 4 group bitmap
7009	Input4.Group	IGROUP	Input 4 group
7010	Input5.GroupMap	IEG_XXX	Input 5 group bitmap
7011	Input5.Group	IGROUP	Input 5 group
7012	Input6.GroupMap	IEG_XXX	Input 6 group bitmap
7013	Input6.Group	IGROUP	Input 6 group
7014	Input7.GroupMap	IEG_XXX	Input 7 group bitmap
7015	Input7.Group	IGROUP	Input 7 group
7016	Input8.GroupMap	IEG_XXX	Input 8 group bitmap
7017	Input8.Group	IGROUP	Input 8 group

****See Appendix A for details on data Types***

Polarities

								IN8	**IN7**	**IN6**	**IN5**	**IN4**	**IN3**	**IN2**	**IN1**
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

IN1
Active low Input 1 polarity

IN2
Active low Input 2 polarity

IN3
Active low Input 3 polarity

IN4
Active low Input 4 polarity

IN5
Active low Input 5 polarity

IN6
Active low Input 6 polarity

IN7
Active low Input 7 polarity

IN8
Active low Input 8 polarity

GroupMap
The *GroupMap* specifies the particular event(s) within the selected input *Group*. An active digital input will activate all specified input events. The bitmap definition for each input group data type is specified in *Appendix 2*.

Group
The group identifies the input event group for the desired input event group type as specified in the following enumeration table.

IGROUP Enumerations

Value	Associated bitmap Type*	Description
0	IEG_MODE	Enable, mode input events
1	IEG_MOTION	General motion input events (jog, home, dedicated move…)
2	IEG_MOVE_LEVEL	Individual move maintained (level) initiates
3	IEG_MOVE_EDGE	Individual move momentary (edge) initiates
4	IEG_MOVE_TEACH	Individual move teach position
5	IEG_MOVE_SELECT	Binary selects
6	IEG_MOVE_SWITCH	Move (feed) switches (level and edge)

****See Appendix A for details on data Types***

Digital Output Assignments

The drive maintains status flags, organized by group, that provide information on the current internal operational status of the drive. The individual bit flags in theses status words are referred to as *Output Status Events* and the state of any flag may be mapped to one of the drive's digital output or LED indicators to provide status feedback. Each output event group may contain up to sixteen specific events. The output event bitmaps for each group are, for convenience, given their own data type since the maps are used for the definition of multiple registers (output event bitmaps are used for digital output assignment, direct control, and status registers).

(See Appendix B for the definition of the OEG_XXX output event group data type bitmaps.)

Digital Output Parameters Register Table

ID	Name	Type*	Description
7001	Output.Polarities	FLAGS	Output polarity bitmap
7018	Output1.GroupMap	OEG_XXX	Output 1 group bitmap
7019	Output1.Group	OGROUP	Output 1 group
7020	Output2.GroupMap	OEG_XXX	Output 2 group bitmap
7021	Output2.Group	OGROUP	Output 2 group
7022	Output3.GroupMap	OEG_XXX	Output 3 group bitmap
7023	Output3.Group	OGROUP	Output 3 group
7024	Output4.GroupMap	OEG_XXX	Output 4 group bitmap
7025	Output4.Group	OGROUP	Output 4 group
7034	RedLed.GroupMap	OEG_XXX	Red LED group bitmap
7035	RedLed.Group	OGROUP	Red LED group
7036	Grn.GroupMap	OEG_XXX	Grn LED group bitmap
7037	Grn2.Group	OGROUP	Grn LED group
7040	Yel1.GroupMap	OEG_XXX	Yel1 LED group bitmap
7041	Yel1.Group	OGROUP	Yel1 LED group
7042	Yel2.GroupMap	OEG_XXX	Yel2 LED group bitmap
7043	Yel2.Group	OGROUP	Yel2 LED group

****See Appendix A for details on data Types***

Polarities

The polarity flags are used to select the active state of the hardware output.

			YEL2	**YEL1**		**GRN**	**RED**					**OUT4**	**OUT3**	**OUT2**	**OUT1**
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

OUT1
Active low Output 1 polarity

OUT2
Active low Output 2 polarity

OUT3
Active low Output 3 polarity

OUT4
Active low Output 4 polarity

RED
Active low Red LED polarity

GRN
Active low Green LED polarity

YEL1
Active low Yellow 1 LED polarity

YEL2
Active low Yellow 2 LED polarity

GroupMap
The *GroupMap* specifies the particular event(s) within the selected output event group. The output will be active when any specified bit of the G*roup* is active.

Group
The group identifies the output event group for the desired output event group type as specified in the following enumeration table.

OGROUP Enumerations

Value	Associated bitmap Type*	Description
0	OEG_STATUS	General drive status output events
1	OEG_MOTION	General motion status output events
2	OEG_CONTROL	Internal control status output events
3	OEG_MOVE_ACTIVE	Active move status output events
4	OEG_MOVE_IN_POSITION	Move in-position status output events

****See Appendix A for details on data Types***

Analog Input Channel Parameters Register Table

CH 1 ID	CH 2 ID	Name	Type*	Description
7200	7230	Options	FLAGS	Options
7201	7231	Bandwidth	UINT16	Filter bandwidth
7202	7232	Mode1UserLow	INT32	Mode 1 user low calibration
7204	7234	Mode1UserHigh	INT32	Mode 1 user high calibration
7206	7236	Mode1AdcLow	INT32	Mode 1 ADC low calibration
7208	7238	Mode1AdcHigh	INT32	Mode 1 ADC high calibration
7210	7240	Mode2UserLow	INT32	Mode 2 user low calibration
7212	7242	Mode2UserHigh	INT32	Mode 2 user high calibration
7214	7244	Mode2AdcLow	INT32	Mode 2 ADC low calibration
7216	7246	Mode2AdcHigh	INT32	Mode 2 ADC high calibration
7218	7248	RangeMimimum	INT32	Minimum value of useable range
7220	7250	RangeMaximum	INT32	Maximum value of useable range
7222	7252	FaultTripLow	INT32	Low fault trip value
7224	7254	FaultTripHigh	INT32	High fault trip value

See Appendix A for details on data Types

Options

												HFLT	LFLT	ORDER2	MODE2
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MODE2

The *MODE2* option selects the Mode2 calibration values to be used at runtime when calculating the analog input value. Some analog input channels may be fed from two physically different hardware inputs with different offsets and scales. This option allows both hardware inputs to be calibrated on a single input channel.

ORDER2

If *ORDER2* is set, the drive will filter the input value using a second order filter algorithm. If *ORDER2* is not set a first order filtering algorithm is used. A second order filter will usually be more responsive but may also lead to 'ringing'. A first order filter may settle to a quieter final value but will be less responsive, settle to its final value at a slower rate, and have trouble actually reaching the exact input value as bandwidth is reduced.

LFLT

The *LFLT* flag enables low range fault checking. A *LOSS OF SIGNAL* fault will be generated if the value of the input variable falls below the level specified by the *RangeMinimum* parameter (below).

HFLT

The *HFLT* flag enables high range fault checking. A *LOSS OF SIGNAL* fault will be generated if the value of the input variable rises above the level specified by the *RangeMaximum* parameter (below).

Bandwidth

The *Bandwidth* register sets the signal tracking frequency of the digital filter used to smooth the analog input signal. The value is a fixed point 8.8 number specifying the signal tracking frequency in Hertz.

Mode1UserLow
Mode1UserHigh
Mode1AdcLow
Mode1AdcHigh
Mode2UserLow
Mode2UserHigh
Mode2AdcLow
Mode2AdcHigh

RangeMinimum
RangeMaximum

FaultTripLow
FaultTripHigh

Analog Output Channel Parameters Register Table

CH 1 ID	CH 2 ID	Name	Type*	Description
7400	7416	Options	FLAGS	Options
7401	7417	Bandwidth	UINT16	Filter bandwidth
7402	7418	VariableID	UINT16	Output variable ID
7403	7419	VariableFlags	UINT16	Output variable flags
7404	7420	CalLow	UINT16	DAC low calibration offset
7405	7421	CalHigh	UINT16	DAC high calibration offset
7406	7422	VarMinimum	INT32	Minimum variable value
7408	7424	VarMaximum	INT32	Maximum variable value

See Appendix A for details on data Types

Options
The *Options* register selects various options available for the analog output channel.

														ORDER2	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

ORDER2
If *ORDER2* is set, the drive will filter the output value using a second order filter algorithm. If *ORDER2* is not set a first order filtering algorithm is used. A second order filter will usually be more responsive but may also lead to 'ringing'. A first order filter may settle to a quieter final value but will be less responsive, settle to its final value at a slower rate, and have trouble actually reaching the true end point values as bandwidth is reduced.

Bandwidth
The *Bandwidth* register specifies the bandwidth, in Hertz, of the filter used to smooth the analog output signal..

VariableID
The *VariableID* register specifies the *MODBUS* identifier of the variable whose value is to be monitored on the analog output channel.

VariableFlags
The *VariableFlags* registers provide additional information about the type of variable being monitored on the analog output channel.

reserved															**D**
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

D
The *D* flag indicates that the variable is a double word (32 bit) variable. The VariableID register specifies the starting register of the value to monitor. If the D flag is set, the starting register is assumed to be the low word of a 32-bit value. If the D flag is not set, the starting register is assumed to be a 16-bit value.

CalLow
DAC low calibration offset

CalHigh
DAC high calibration offset

VarMinimum
VarMaximum
Minimum and maximum variable values defining the range over which the analog output is scaled.

Mapped Parameters

The mapped parameter register tables provide a method for user to organize the *MODBUS* registers that it needs to read and write into contiguous blocks. The *MappedRead* and *MappedWrite* parameter registers each contain a list MODBUS register identifiers. At run time, a *MODBUS Master* may use the *MODBUS* read/write multiple register commands to read or write contiguous blocks of data from the *MappedReadValue* and *MappedWriteValue* tables. These value tables do hold actual data values but refer instead to the values stored at the registers specified in the corresponding *MappedRead* or *MappedWrite* parameter tables. Considerable runtime communications overhead is saved since the individual registers do not have to be individually read or written single *MODBUS* commands.

Mapped Register Parameter Table

ID	Name	Type*	Size	RunTime Table	Description
8000	MappedRead	UINT16	100	MappedReadValue	Mapped read MODBUS register IDs [0..99]
8200	MappedWrite	UINT16	100	MappedWriteValue	Mapped write MODBUS register IDs [0..99]

See Appendix A for details on data Types

MappedRead
Each element of the *MappedRead* table specifies the *MODBUS* identifier of the register that will is indirectly read or written through the corresponding element of the *MappedReadValue* table. Unused table values should be set to zero.

MappedWrite
Each element of the *MappedWrite* table specifies the *MODBUS* identifier of the register that will is indirectly read or written through the corresponding element of the *MappedWriteValue* table. Unused table values should be set to zero.

Mapped Register Value Table

ID	Name	Type*	Size	Description
8400	MappedReadValue	UINT16	100	Runtime indirect values for MappedRead IDs
8600	MappedWriteValue	UINT16	100	Runtime indirect values for MappedWrite IDs

See Appendix A for details on data Types

MappedReadValue
Reading or writing values in the *MappedReadValue* table will read or write the register specified in the corresponding element of the MappedRead parameter register table. If the *MappedRead* parameter value (i.e. register id) is zero, a read will return a zero and a write will have no effect.

MappedWriteValue
Reading or writing values in the *MappedWriteValue* table will read or write the register specified in the corresponding element of the MappedWrite parameter register table. If the *MappedWrite* parameter value (i.e. register id) is zero, a read will return a zero and a write will have no effect.

The *MappedRead/MappedWrite* and *MappedReadValue/MappedWriteValue* tables work in an identical manner. There is no actual requirement that the *read* tables be used only for reading registers and the *write* tables for writing registers. A *MODBUS Master* is free to read or write from either table and may read and write from the same table. Any starting register and number of multiple registers to be read or written may be specified so long as the range of registers is within the table being used.

An *EtherNet/IP MODBUS Master* uses the read tables only for reading, the write tables only for writing, and will read or write the full table with a single *MODBUS* command.

Factory Parameters

Factory parameters are stored as a block in non-volatile memory. The block contains a CRC (Cyclic Redundancy Checksum) word to guarantee data integrity. At power-up, the factory parameter block is validated and copied to its runtime location in RAM where all parameters are available for both reading and writing through their individual MODBUS identifiers.

Factory Identification Parameters

Factory Identification Parameters Register Table

ID	Name	Type*	Description
9000	PartNumber	STR16	Factory part number
9016	SerialNumber	STR16	Factory serial number

See Appendix A for details on data Types

PartNumber
The *PartNumber* register specifies the part number assigned to the drive by the factory during commissioning.

SerialNumber
The *SerialNumber* register specifies the serial number assigned to the drive by the factory during commissioning.

Factory Actuator Parameters

Factory Actuator Parameters Register Table

ID	Name	Type*	Description
9100	Model	STR16	Model name
9116	Options	FLAGS	Option flags
9117	EcyclesPerRev	UINT16	Poles / 2
9118	R	UINT16	Resistance [8.8 ohms L-L]
9119	L	UINT16	Inductance [8.8 mH L-L]
9120	J	UINT32	Inertia [0.32 kg-m^2]
9122	KT	UINT16	KT [6.10 Nm/AMP]
9123	FeedbackDevice	ENUM	Position feedback device type
9124	StepsPerRev	UINT32	Encoder steps/rev

See Appendix A for details on data Types

Model
The *Model* specifies the factory name for the actuator.

Options
The Options register is used to enable various actuator options as specified in the word bitmap.

													EAB	**PTBL**	**TEMP**
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

EAB
Allows position feedback encoder wiring flexibility by performing an internal signal swap of the encoder A dn B channels.

PTBL
Enables use of the factory parameters position correction table. This flag will normally be enabled when the drive uses linear Hall sensors for position feedback.

TEMP
The *TEMP* option indicates that the actuator utilizes a thermistor for temperature fault detection. If enabled, the drive will monitor the actuator's temperature for a fault (or warning) condition using the factory actuator temperature limit. If disabled, the fault input from the actuator is monitored a digital fault input and the factory actuator temperature trip level parameter is ignored.

EcyclesPerRev
EcyclesPerRev specifies the electrical cycles per actuator revolution, or one-half the number of motor electrical poles.

R
R specifies the line-to-line winding resistance in fixed-point units of 8.8 ohms.

L
L specifies the line-to-line winding inductance in fixed-point units of 8.8 milli-henrys.

J
J specifies the actuator's rotor intertia in fixed-point units of 0.32 kg-m^2.

KT
KT specifies the actuator's torque constant in fixed-point units of 6.10 Nm / amp.

FeedbackDevice
FeedbackDevice is an enumerated value specifying the type of the actuator's position feedback device and should be one of the following values:

0 - Analog Hall
The *Analog Hall* feedback device uses magnetic linear Hall analog inputs to generate pseudo-sinusoidal analog signals in quadrature from which a position angle may be determined.

1 - Analog Hall Absolute
The *Analog Hall Absolute* uses the same angular feedback method as the Analog Hall, but also incorporates battery backed monitoring functionality to track position change while the drive is powered down so that the absolute position of the drive is always tracked.

2 - Incremental Encoder
The Incremental Encoder position feedback device utilizes an incremental encoder with UVW commutation tracks.

StepsPerRev
StepsPerRev specifies the steps per actuator revolution from an encoder position device. *StepsPerRev* is used only if the *FeedbackDevice* has been selected to Encoder.

CAUTION
StepsPerRev is specified in steps/rev, NOT encoder lines. For standard quadrature encoding, steps/rev = 4 * encoder lines

Factory Limits Parameters

Factory Limits Parameters Register Table

ID	Name	Type*	Description
9200	LowVoltageTripLevel	UVOLT16	Low voltage fault trip level
9201	HighVoltageTripLevel	UVOLT16	High voltage fault trip level
9202	BoardTempTripLevel	BTMP16	PCB temperature trip level
9203	Itrip	UCUR16	Current fault trip level
9204	Ipeak	UCUR16	Peak command current
9205	Icontinuous	UCUR16	Continuous current rating
9206	IcTimeConstant	UINT16	Continuous current time constant
9209	ActuatorTempTripLevel	ATMP16	Actuator temperature fault trip level
9210	PwmModulation	INT16	PWM modulation factor
9211	ShuntHigh	UINT16	High shunt level (shunt ON)
9212	ShuntLow	UINT16	Low shunt level (shunt OFF)

See Appendix A for details on data Types

LowVoltageTripLevel
HighVoltageTripLevel
BoardTempTripLevel
Itrip
Ipeak
Icontinuous
IcTimeConstant
ActuatorTempTripLevel
PwmModulation
ShuntHigh
ShuntLow

Factory Calibration Parameters

Factory Calibration Parameters Register Table

ID	Name	Type*	Description
9300	Options	FLAGS	Options / option board flags
9301	VbusScale	INT16	DC Bus voltage scale
9302	BoardTempOffset	INT16	PCB temperature offset
9303	BoardTempScale	INT16	PCB temperature scale
9304	PsineInZero	INT16	Linear Hall sine zero offset
9305	PsineInScale	INT16	Linear Hall sine scale
9306	PcosineInZero	INT16	Linear Hall cosine zero offset
9307	PcosineInScale	INT16	Linear Hall cosine scale
9308	RphaseOffset	INT16	R phase current sensor offset
9309	RphaseScale	INT16	R phase current sensor scale
9310	SphaseOffset	INT16	S phase current sensor offset
9311	SphaseScale	INT16	S phase current sensor scale
9315	Eoffset	INT16	Electrical angle offset
9320	BrakeReleaseDelay	UINT16	Brake release delay [0.01s]
9321	BrakeEngageDelay	UINT16	Brake engage delay [0.01s]
9323	TharmonicMag	INT16	Torque harmonic magnitude [2..14AMPS]
9328	ActuatorTempOffset	INT16	Actuator temperature offset [13.3 DEG]
9329	ActuatorTempScale	INT16	Actuator temperature scale [13.3 DEG/FULLSCALE]

See Appendix A for details on data Types

Options
The *Options* register is used to enable various factory options as specified by the following table.

IA4	SCIB													TH
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

IA4
Analog I/O option board.

SCIB
Modbus serial communications enabled on on SCI-B.

TH
Torque harmonics enable

VbusScale
BoardTempOffset
BoardTempScale
PsineInZero
PsineInScale
PcosineInZero
PcosingInScale
RphaseOffset
RphaseScale
SphaseOffset
SphaseScale
Eoffset
BrakeReleaseDelay
BrakeEngageDelay
TharmonicMag
ActuatorTempOffset
ActuatorTempScale

Factory Tuning Parameters

Factory Tuning Parameters Register Table

ID	Name	Type*	Description
9400	Options	FLAGS	Option flags
9401	HallBW	UINT16	Position angle tracking bandwidth
9402	HallDamping	UINT16	Position angle tracking damping
9403	IloopBW	UINT16	Current loop bandwidth [HZ]
9405	VbusBW	UINT16	Bus voltage filter bandwidth [HZ]

See Appendix A for details on data Types

Options
The *Options* register enables various internal tuning and algorithm options as specified by the following table.

															SV
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SV
Enables use of a *space vector* voltage control algorithm in the current loop as opposed to pure sinusoidal voltage *PWM* commands to the actuator phases. The space vector algorithm allows for more efficient use of the available DC bus voltage, resulting in more available torque at any given speed. The *space vector* option is normally enabled.

HallBW
The bandwidth of the position tracking converter in rad/s.

HallDamping

IloopBW
The bandwidth of the current loop in HZ.

VbusBW
The bandwidth of the DC Bus filter in HZ.

Factory Position Correction Table

Factory Position Correction Register Table

ID	Name	Type*	Description
9500	PosCorrectionTable	INT16[64]	Position correction factors table

See Appendix A for details on data Types

Factory Commutation Table

The commutation table is set at the factory for an actuator using encoder position feedback. The encoder device contains UVW commutation tracks. The tracks are usually Grey scale encoded (only one bit can change at a time). The table contains the angle, in electrical degrees, for the center of the UVW pattern and enables the drive to establish the initial electrical within +/- 30 degrees at startup (the exact angle is not known until a change is observed to the UVW pattern). Illegal table values (normally 000 and 111) should have the high bit (Bit 15) set in the angle.

Factory Commutation Register Table

ID	Name	Type*	UVW Electric Angle
9600	0	UINT16	0 0 0 pattern angle
9601	1	UINT16	0 0 1 pattern angle
9602	2	UINT16	0 1 0 pattern angle
9603	3	UINT16	0 1 1 pattern angle
9604	4	UINT16	1 0 0 pattern angle
9605	5	UINT16	1 0 1 pattern angle
9606	6	UINT16	1 1 0 pattern angle
9607	7	UINT16	1 1 1 pattern angle

See Appendix A for details on data Types

Factory Communication Parameters

The serial channel used internally for communications with the Ethernet/IP Modbus Master uses the *MODBUS* parameters shown in the table below. The parameter descriptions are identical to those used for the user Modbus parameter registers.

Factory Communication Parameters Register Table

ID	Name	Type*	Description
9700	Flags	FLAGS	Modbus Serial Channel B flags
9701	AxisId	UINT16	Modbus Serial Channel B axis identifier
9702	Baud	UINT16	Modbus Serial Channel B baud identifier
9703	RxDelay	UINT16	Modbus Serial Channel B extra RX delay
9704	TxDelay	UINT16	Modbus Serial Channel B extra TX delay

See Appendix A for details on data Types

Status Registers

All status registers are read-only (*MODBUS* input registers) and provide feedback information on the runtime operation of the drive. Status registers normally exist in *RAM* and their values are not saved in non-volatile memory between drive power-ups.

System Status Registers

This section describes general status registers used to monitoring overall drive operational status.

System Status Register Table

ID	Name	Type*	Description
4	Disables	FLAGS	Drive disabling sources
5	Faults	FAULT	All active faults
6	HardFaults	FAULT	Active disabling faults
7	SoftFaults	FAULT	Active non-disabling faults
8	HallBattery	UVOLT16	Absolute Hall board battery voltage [11.5 V]
10	BoardTemp	BTMP32	PCB temperature [11.21 DEG C]
14	ActuatorTemp	ATMP32	Actuator temperature [13.19 DEG C]
24	Faults32	FAULT	All active faults (32 faults)
26	HardFaults32	FAULT	Active disabling faults (32 faults)
28	SoftFaults32	FAULT	Active non-disabling faults (32 faults)

Disables
The Disables status register is a bitmapped value indicating all currently active disabling sources in the drive. When all disabling sources are removed (register value is zero), the drive will become enabled.

															LVW
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

INT	**PU**	**FLASH**	**RESET**		**COMMS**						**MODE**	**EMOVE**	**FLT**	**HOST**	**HWENA**
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

HWENA
The drive cannot enable due to an inactive *Enable* input event. This disabling source will clear on the rising edge of a *Momentary Enable* input event, an active *Maintained Enable* input event, or an active *AE* (*Auto Enable*) flag in *Configuration.Options*.

HOST
The drive cannot enable while the Host.Disables register is non-zero.

FLT
The drive is disabled due to an active FAULT that has been selected as ‘hard’ fault which disables the drive.

EMOVE
The drive has been disabled upon completion of a *Dedicated Move* with the *DMD* (*Dedicated Move Disable*) option selected in *Configuration.Options*. Disabling upon completion of a *Dedicated Move* is not considered a fault and the *EMOVE* flag is not cleared on the rising edge of the *Reset Faults* input event but instead is cleared only on the rising edge of *ENABLE*, leaving the *EMOVE* flag active as an indicator of why the drive disabled.

MODE
The drive’s currently active *Command Mode* is the *Disabled* mode. The *MODE* flag will clear automatically when the drive’s command mode is anything other than the *Disabled* mode.

COMMS

The drive has lost communications after it had been established. This flag does NOT automatically clear if the communications is reestablished. This flag can be disabled from the *Comms Faults* tab in the *System Setup* page.

RESET
The drive is temporarily disabled internally while restarting from a *Restart Drive* system command. The *RFESET* flag will not be active during normal operation.

FLASH
The drive is disabled internally while firmware is being upgraded. The *FLASH* flag will not be active during normal operation.

PU
The drive is temporarily disabled internally during startup initialization. This disabling indicator is used internally at startup to keep the drive disabled until internal filtered values have stabilized and the drive is ready for normal operation. The *PU* flag will not be active during normal operation.

INT
The drive is disabled for unspecified internal reasons. The *INT* flag will not be active during normal operation.

Faults and Faults2
The *Faults/Faults2* registers latches all fault conditions observed whether or not the faults have been selected to generate hard or soft faults. Flags set in this register indicate that the criteria required for the fault condition has been satisfied. The *Faults* register is reset to zero on the rising edge of the *Enable Momentary*, *Enable Maintained*, or *Reset Faults* input events.

HardFaults and HardFaults2
The *HardFaults/Faults2* registers latches all fault conditions that have been selected in *FaultDisables/FaultsDisables2* to disable the drive. The *Faults/Faults2* registers are reset to zero on the rising edge of the *Enable Momentary*, *Enable Maintained*, or *Reset Faults* input events

SoftFaults and SoftFaults2
The *SoftFaults/SoftFaults2* registers indicates all fault conditions that are currently active and have been selected in the FaultWarnings register. The *Fault Warning* output event will be active while any flag is active in *SoftFaults/SoftFaults2*. *SoftFaults/SoftFaults2* flags are dynamic and will reset automatically when the fault condition is removed.

HallBattery
The *HallBattery* register monitors the battery supply voltage from the Absolute Hall board. The Absolute Hall board's nominal voltage is 3.7V full-scale. A value of less than 2.8V will set the *Hall Battery* fault flag (*HB)* flag in the *Faults* register. The register is updated only when the actuator's position feedback device is selected to be Absolute Hall feedback – for other position feedback devices the voltage will always be zero and the *Hall Battery* fault flag is not updated.

BoardTemp
The *BoardTemp* register monitors the *PCB* (*Printed Circuit Board*) temperature. A value greater than *BoardTempTripLevel* in the *Factory Limits Register Table* will set the *Board Temperature* fault flag (*BT*) in the *Faults* register.

ActuatorTemp
The *ActuatorTemp* register monitors actuator temperature from the actuator's temperature fault input. A value greater than *ActuatorTempTripLevel* in the *Factory Limits Register Table* will set the *Actuator* T*emperature* fault flag (*AT*) in the *Faults* register. *ActuatorTemp* is updated only when the temperature fault input from the actuator is configured as a thermistor input. When the actuator's temperature fault input is configured as a digital switch input, the *ActuatorTemp* register will always be zero and the *Actuator Temperature* fault flag will be updated from the status of the digital switch input.

BoardTemp and ActuatorTemp are 32-bit filtered values. In general, only the high word of these registers (ID + 1) will need to be monitored - the low word provides extra resolution used internally by the filtering process.

Digital Input Status Register Table

ID	Name	Type*	Description
100	Inputs	FLAGS	Active input status
102	HwInputs	FLAGS	Hardware input status (before inhibits/polarity)
110	InputEvents.Mode	IEG_MODE	Mode input events
111	InputEvents.Motion	IEG_MOTION	Motion input events
112	InputEvents.MoveLevel	IEG_MOVE_LEVEL	Move Maintained input events
113	InputEvents.MoveEdge	IEG_MOVE_EDGE	Move Momentary input events
114	InputEvents.MoveTeach	IEG_MOVE_TEACH	Move Teach Position input events
115	InputEvents.MoveSelect	IEG_MOVE_SELECT	Binary Select input events
116	InputEvents.MoveSwitch	IEG_MOVE_SWITCH	Move Switches input events
118	LatchedInputEvents[8]	IEG_x	Latched input event status
126	RisingInputEvents[8]	IEG_x	Rising edge input event status
134	FallingInputEvents[8]	IEG_x	Falling edge input event status

See Appendix A for details on data Types

Inputs
The *Inputs* register indicates the active state of the digital inputs. The active input state is the logical exclusive or of the *HwInputs* register (below) and the *Inputs.Polarity* parameter register.

HwInputs
The HwInputs register reflects the hardware status of the digital inputs.

InputEvents.x
The input event registers reflect the active state of all input event groups. Input events are activated when a digital input assigned to the event is active in the Inputs register (if the input is not inhibited by a host controller) or when the event is directly set active by the host through the host input event registers.

LatchedInputEvents
RisingInputEvents
FallingInputEvents
These registers are used internally to track edge sensitivity of the inputs and are updated during each background scan of the controller.

Digital Output Status Register Table

ID	Name	Type*	Description
101	Outputs	FLAGS	Active output status
103	HwOutputs	FLAGS	Hardware output status (after inhibits/polarity)
104	OutputEvents.Status	OEG_STATUS	Drive Status output events
105	OutputEvents.Motion	OEG_MOTION	Motion output events
106	OutputEvents.Control	OEG_CONTROL	Control output events
107	OutputEvents.MoveActive	OEG_MOVE_ACTIVE	Move Active output events
108	OutputEvents.MoveInPosition	OEG_MOVE_IN_POS	Move In-Position output events

See Appendix A for details on data Types

Outputs
The *Outputs* register reflects the state of any output status events assigned to the hardware outputs and LEDs.

HwOutputs
Unless an output is inhibited by a host controller, the HwOutputs refelects the logical exclusive or of the Outputs register and the Outputs.Polarity parameter register combined with any direct outputs set by a host controller in Host.Outputs. When an output is inhibited, only the contribution from Host.Outputs will be reflected. The HwOutputs word directly maps to the actual state of the hardware digital outputs and LEDs.

OutputEvents.x
The output event registers reflect the active state of all output status event groups. Digital outputs assigned to an output status event will be active in the Outputs register when the associated output status event is active.

Analog Input Status Registers

The registers in the *Analog Input Status Registers Table* provide runtime information on the state of *Analog Input Channel 1* and *Analog Input Channel 2*.

Analog Input Status Registers Table

ID		Name	Type*	Description
CH 1	CH 2			
221	241	Instantaneous	UINT16	Unfiltered ADC input (0x0000..0xFFF0)
222	242	Filtered	INT32	Filtered ADC input (0..1) [2.30]
224	244	UserFiltered	INT32	Filtered input [6.26 user units]
226	246	UseableRange	INT32	Fraction of useable range (0..1) [2.30]
228	248	dydx	INT32	dy/dx scale factor [10.22]
230	250	B	INT32	y-axis intercept offset [6.26 user units]
232	252	RangeMinimum	INT32	Minimum useable range [2.30 ADC]
234	254	RangeMaximum	INT32	Maximum useable range [2.30 ADC]

See Appendix A for details on data Types

Instantaneous
The *Instantaneous* register provides the unfiltered most recent value from the ADC.

Filtered
The *Filtered* register provides the raw filtered value from the ADC. The value is a fixed-point 2.30 value in the range 0 to 1.

UserFiltered
The *UserFiltered* register scales the raw *Filtered* value (above) in the user range specified by the *RangeMinimum* and *RangeMaximum* registers.

UseableRange

dydx
b
For runtime efficiency, internal scale and offset values are calculated by the controller to convert runtime analog input values to user units from the two-point calibration values. The *dydx* (scale) and b (*offset*) values are calculated at start-up and whenever the analog input parameter values are modified.

RangeMinimum
RangeMaximum
The *RangeMinimum* and *RangeMaximum* registers specify the minimum and maximum values, respectively, that the final *UserFiltered* input value will be scaled into. The values are fixed-point 2.30 values.

Analog Output Status Registers

The registers in the *Analog Output Status Registers Table* provide runtime information on the state of the values being monitored on *Analog Ouput Channel 1* and *Analog Output Channel 2*.

Analog Output Status RegistersTable

ID		Name	Type*	Description
CH 1	CH 2			
700	704	Filtered	INT32	Filtered output variable
702	706	RawDAC	UINT16	DAC output [0x0000..0xFFFF]
703	707	Fraction	UINT16	Fraction of variable range

See Appendix A for details on data Types

Filtered
The register(s) being monitored on the output are filtered in the 32-bit *Filtered* register based on the filter parameters set for the analog channel. If monitoring a single (16-bit) register, the 32-bit filtered value will be left justified.

RawDAC
RawDAC uses the analog output's calibration values to scale the calculated Fraction into the range 0x0000 to 0xFFFF using the basic formala: *RawDAC* = calLow + ((0xFFFF – calHigh) – calLow) * *Fraction* / 0x8000. A *RawDAC* value of 0x0000 maps to the minimum *DAC* output and a value of 0xFFFF maps to the maximum *DAC* output. (The actual value written to the *DAC* will depend on the hardware specific *DAC* circuitry and resolution used for the particular analog output channel.

Fraction
Fraction uses the minimum and maximum values set-up for the analog output channel to scale the *Filtered* value into the range 0x0000 to 0x8000 using the basic formula: *Fraction* = (*Filtered* – minimum) / (maximum – minimum).

Analog Command Status Registers

The *Analog Command Status Registers Table* specifies the registers that monitor the analog input commands for the *Analog Position*, *Analog Velocity*, and *Analog Torque* operating modes. These status registers are updated at a *1 millisecond* rate regardless of the drive's operating mode. The analog input channel source and operational range of the analog command signals are set-up in the *Analog Motion Control* parameters section.

Analog Command Status Registers Table

ID	Name	Type*	Description
200	AnalogPositionTarget	POS32	Analog Position Mode command
202	AnalogVelocityTarget	VEL32	Analog Velocity Mode command
204	AnalogCurrentTarget	CUR32	Analog Current Mode command

See Appendix A for details on data Types

AnalogPositionTarget
Position command to the position loop while the drive's operating mode is Analog Position Mode.

AnalogVelocityTarget
Velocity command to the position loop while the drive's operating mode is Analog Velocity Mode.

AnalogCurrentTarget
Current (torque) command to the current loop while the drive's operating mode is Analog Current Mode.

Position and Velocity Status Registers

The status registers in this section provide position and velocity command, feedback, and error status.

ID	Name	Type*	Description
344	Vfeedback	VEL32	Feedback velocity
346	Vcommand	VEL32	Command velocity
348	Verror	VEL32	Velocity error
350	VerrorMin	VEL32	Minimum velocity error
352	VerrorMax	VEL32	Maximum velocity error
356	Vdisplay	VEL32	Filtered display velocity
378	Pfeedback	POS32	Feedback position
380	Pcommand	POS32	Command position
382	Perror	POS32	Position error
384	PerrorMin	POS32	Minimum position error
386	PerrorMax	POS32	Maximum position error

See Appendix A for details on data Types

Vfeedback
The *Vfeedback* register monitors the feedback velocity.

Vcommand
The *Vcommand* register monitors the command velocity.

Verror
The *Verror r*egister monitors the velocity error and is equal to the difference between *Vcommand* and *Vfeedback*.

VerrorMin
VerrorMax
VerrorMin and *VerrorMax* monitor the minimum and maximum values of *Verror*, respectively, that have been observed since diagnostics values were last reset.

Pfeedback
The *Pfeedback* register monitors the absolute feedback position.

Pcommand
The *Pcommand* register monitors the absolute command position.

Perror
The *Perror r*egister monitors the positional following error and is equal to the difference between *Pcommand* and *Pfeedback*.

PerrorMin
PerrorMax
PerrorMin and *PerrorMax* monitor the minimum and maximum values of *Perror,* respectively, that have been observed since diagnostics values were last reset.

Current Status Registers

The registers in this section provide feedback on the state of the current loop and *PWM* phase voltages.

ID	Name	Type*	Description
502	Eangle	REV32	Electrical angle
504	Esine	INT32	Sin(Eangle)
506	Ecosine	INT32	Cos(Eangle)
508	IR	CUR32	R phase current feedback
510	IS	CUR32	S phase current feedback
512	IT	CUR32	T phase current feedback
520	IqFeedback	CUR32	Q (torque) leg current
526	Imagnitude	CUR32	Current vector magnitude
564	Icontinuous	UCUR32	Continuous current
566	Idisplay	CUR32	Filtered display current
568	VbusI	VOLT32	Instantaneous bus voltage
570	Vbus	VOLT32	Filtered bus voltage

See Appendix A for details on data Types

Eangle
The *Eangle* register monitors the electrical being used for commutation.

Esine
Ecosine
The *Esine* and *Ecosine* registers monitor the sine and cosine of the electrical angle, respectively, in 2.30 fixed point units.

IR
IS
IT
The *IR*, *IS*, and *IT* registers monitor the actuator's phase current feedbacks.

IqFeedback
The *IqFeedback* register monitors the current (torque) leg of the current feedback vector.

Imagnitude
The *Imagnitude* register monitors the value of the overall current vector magnitude.

Icontinuous
The *Icontinuous* register monitors the value of the average current being used by the drive.

Idisplay
The *Idisplay* register provides a more stable, filtered value of the *IqFeedback* which is used for display purposes.

VbusI
The *VbusI* register provides the unfiltered value of the most recent voltage present on the DC Bus.

Vbus
The *Vbus* register provides the filtered value of the voltage present on the DC Bus.

Command Status Registers

Registers in this section reflect the final internal command values based on the active operational command mode and command limits that are in effect.

Command Register Table

ID	Name	Type*	Description
30	SubMode	FLAGS	Active command sub-mode
31	Mode	OPMODE	Active command mode
34	PlimitMinus	POS32	Active position limit (-)
36	PlimitPlus	POS32	Active position limit (+)
38	PlimitVelocity	INT32	Active position limit velocity
40	IlimitMinus	CUR32	Active current limit (-)
42	IlimitPlus	CUR32	Active current limit (+)
44	Vlimit	INT32	Active velocity limit
46	Alimit	UACC32	Active acceleration limit
50	Ptarget	POS32	Active position target
52	Vtarget	INT32	Active velocity target
54	Id	CUR32	Direct current command
56	Iq	CUR32	Quadrature current command
58	Vd	VOLT32	Direct voltage command
60	Vq	VOLT32	Direct quadrature voltage command

See Appendix A for details on data Types

SubMode
The *SubMode* register provides additional command mode information for the main command mode specified in the *Mode* register. Currently, *SubMode* values are used only for *Digital IO* command mode as specified by the bitmap table. For all other modes, the *SubMode* register will always be zero.

UNHOMED	BKDLY								MOV			JOG_NEG	JOG_POS	PAUSE	STOP
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

STOP
The *STOP* flag indicates the drive is holding due to a stop condition.

PAUSE
The *PAUSE* flag indicates the drive is holding due to a paused move condition.

JOG_NEG
JOG_POS
The *JOG_NEG* and *JOG_POS* flags indicate that the *JOG NEGATIVE* or *JOG POSITIVE* input events are active.

MOV
The MOV field encodes a currently active move.
0 – none
3 – Dedicated Move to Position active
4 – Home Move active
5 – Standard Move (0..15) active

BKDLY
The *BKDLY* flag indicates the drive is enabled but brake holding delay is active.

UNHOMED
The *UNHOMED* flag indicates the drive is enabled but the absolute position reference frame must be defined before normal operation can occur.

Mode
The *Mode* register specifies the currently active command operational mode.

PlimitMinus
PlimitPlus
The PlimitMinus and PlimitPlus registers provide the values of the negative and positive position limits, respectively, that are currently in effect. These values are available whether or not either position limit is actually enabled.

IlimitMinus
IlimitPlus
The *IlimitMinus* and *IlimitPlus* registers provide the values of the negative and positive current limits that are in effect.

Vlimit
Alimit
The *Vlimit* and *Alimit* registers provide the values of command velocity and acceleration that are in effect.

Ptarget
Vtarget
The *Ptarget* and *Vtarget* registers provide the position and velocity targets used for all positional control modes of operation.

Id
Iq
The *Id* and *Iq* registers provide the direct and quadrature (torque) current commands to the current control loop. The Id command will always be zero for normal operational modes and is used only in the current and voltage locking modes.

Vd
Vq
The *Vd* and *Vq* registers provide the direct and quadrature *PWM* voltage commands.

System Diagnostics Register Table

ID	Name	Type*	Description
576	MaxImagnitude	UCUR32	Maximum current vector magnitude observed
578	MaxVbus	VOLT32	Maximum DC bus voltage observed
580	MinLoopTime	UINT32	Minimum control loop execution time [SYSCLK]
582	MaxLoopTime	UINT32	Maximum control loop execution time [SYSCLK]
584	AvgLoopTime	UINT32	Average control loop execution time [SYSCLK]
16	MinScanTime	UINT32	Minimum background scan time [SYSCLK]
18	MaxScanTime	UINT32	Maximum background scan time [SYSCLK]
20	AvgScanTime	UINT32	Average background scan time [SYSCLK]

See Appendix A for details on data Types

MaxImagnitude
MaxVbus
The current loop controller tracks the maximum current vector magnitude and DC bus voltage values observed since power-on. If the values observed exceed the maximum stress values in the *Status Log*, the *Status Log* values are updated and the *Status Log* is re-saved to non-volatile memory. The *IDIAG* control bit in the *System Command* register may be used to reset the maximum values observed and capture new data.

MinLoopTime
MaxLoopTime
AvgLoopTime
The current loop controller tracks its minimum and maximum execution times. The execution times are run through an internal single stage filter to track the average execution time. All times are tracked at the system clock rate of *100 MHZ*. The loop execution time also includes the time required to execute the position and velocity loop controllers. The *IDIAG* control bit in the *System Command* register may be used to reset the minimum and maximum times observed and capture new data. The drive's loop execution times are normally used internally for test and validation.

MinScanTime
MaxScanTime
AvgScanTime
The drive's background scan routine tracks its minimum and maximum execution times. The execution times are run through an internal single stage filter to track the average execution time. All times are tracked at the system clock rate of *100 MHZ*. The *IDIAG* control bit in the *System Command* register may be used to reset the minimum and maximum times observed and capture new data. Background scan information is normally used internally for test and validation.

The major functions executed by the background scan include:

- Monitoring system commands and internal recalculations on change of parameter values
- Updating hardware input status registers
- Updating input event status registers
- Updating fault and disable status registers
- Determination and updating of the operational command mode
- Updating output status event registers
- Updating hardware output status registers
- Updating analog outputs
- Monitoring of position *Teach* events
- Handling MODBUS Channel 1 commands
- Handling MODBUS Channel 2 commands

ADC Status Registers

Access to the *DSP*'s internal *ADC* (*Analog to Digital Conversion) Module* values is provided through the registers in the *ADC Status Register Table*. These registers are normally used internally for testing and validation.

ADC Status Register Table

ID	Name	Type*	Description
900	ADC.0	UINT16	DSP ADC channel 0 (0x0000..0xFFF0)
901	ADC.1	UINT16	DSP ADC channel 1 (0x0000..0xFFF0)
902	ADC.2	UINT16	DSP ADC channel 2 (0x0000..0xFFF0)
903	ADC.3	UINT16	DSP ADC channel 3 (0x0000..0xFFF0)
904	ADC.4	UINT16	DSP ADC channel 4 (0x0000..0xFFF0)
905	ADC.5	UINT16	DSP ADC channel 5 (0x0000..0xFFF0)
906	ADC.6	UINT16	DSP ADC channel 6 (0x0000..0xFFF0)
907	ADC.7	UINT16	DSP ADC channel 7 (0x0000..0xFFF0)
908	ADC.8	UINT16	DSP ADC channel 8 (0x0000..0xFFF0)
909	ADC.9	UINT16	DSP ADC channel 9 (0x0000..0xFFF0)
910	ADC.10	UINT16	DSP ADC channel 10 (0x0000..0xFFF0)
911	ADC.11	UINT16	DSP ADC channel 11 (0x0000..0xFFF0)
912	ADC.12	UINT16	DSP ADC channel 12 (0x0000..0xFFF0)
913	ADC.13	UINT16	DSP ADC channel 13 (0x0000..0xFFF0)
914	ADC.14	UINT16	DSP ADC channel 14 (0x0000..0xFFF0)
915	ADC.15	UINT16	DSP ADC channel 15 (0x0000..0xFFF0)

See Appendix A for details on data Types

Factory Identification Registers

Registers in this section are stored in *ROM* (*Read-Only Memory*) and are attributes of the firmware code flashed into the *DSP*. (Note that these registers are an exception to normal 'status' registers which are stored in *RAM*.)

System Identification Register Table

ID	Name	Type*	Description
9910	AppIdentifier	UINT16	Application identifier
9911	AppVersion	UINT16	Application firmware version [0.01 revision units]
9916	BootIdentifier	UINT16	Boot code identifier
9917	BootVersion	UINT16	Boot code firmware version [0.01 revision units]

See Appendix A for details on data Types

AppIdentifier
The *AppIdentifier* register specifies a factory specific product identifier used to distinguish between product families. Standard products will have values in the range 0 to 127. The value will always be two for standard *Tritex II* firmware. Customized firmware for special applications may will have values in the range 128 to 255.

AppVersion
The *AppVersion* register specifies the firmware revision of the application code. The value is specified in units of 0.01 revision units. For example, a value of 203 would indicate firmware *Revesion 2.03* of the product type specified by *AppIdentifier*.

BootIdentifier
The *BootIdentifier* register specifies a factory specific value used internally to identify the boot-strapping firmware resident in the DSP. This special firmware block is used to flash the application code into the DSP.

BootVersion
The *BootVersion* register specifies the firmware revision of the boot-strapping DSP code. The value is specified in units of 0.01 revision units.

Host Control

All registers in the *Host Register Table* are read-write (*MODBUS* holding registers). They are initialized to zero at power-up and are thereafter never modified by the drive's internal operation. These registers are used by a *MODBUS Master* that desires direct control over various features of the drive's runtime operation.

Host Register Table

ID	Name	Type*	Description
4302	Disables	FLAGS	Disable drive flags
4303	CommandMode	OPMODE	Mode of operation
4304	Position	POS32	Target position
4306	Velocity	VEL32 / UVEL32	Velocity command / limit
4308	Acceleration	UACC32	Acceleration limit
4310	Current	CUR16 / UCUR16	Current command / limit
4311	Voltage	VOLT16	Voltage command
4312	InputInhibits	FLAGS	Hardware input inhibits
4313	OutputInhibits	FLAGS	Hardware output inhibits
4314	Outputs	FLAGS	Direct outputs
4315	FactoryTest	UINT16	Reserved for factory test and internal testing
4316	InputEvents.Mode.	IEG_MODE	Input Event Groups
4317	InputEvents.Motion	IEG_MOTION	
4318	InputEvents.MoveLevel	IEG_MOVE_LEVEL	
4319	InputEvents.MoveEdge	IEG_MOVE_EDGE	
4320	InputEvents.MoveTeach	IEG_MOVE_TEACH	
4321	InputEvents.MoveSelect	IEG_MOVE_SELECT	
4322	InputEvents.MoveSwitch	IEG_MOVE_SWITCH	

See Appendix A for details on data Types

Disables
The *Disables* register provides an easy method for a host controller to force the drive to be disabled. Any non-zero value will cause the *HOST* flag in the *Status.Disables* register to be set and disable the drive. A zero value will clear the *HOST* flag in the *Status.Disables* register and the drive may become enabled.

CommandMode
CommandMode allows the host to override the active drive command mode (the default or alternate mode). Any mode specified other than *Disabled Mode* will have priority and take effect immediately. The *Disabled Mode* (0) is used to indicate normal command mode operation and may be set by the host to relinquish direct control over the drive's command mode.

When a host specifies an operating mode (other than *Disabled*) , the DefaultModeActive and AlternateModeActive status flags will remain INACTIVE, even if the mode specified by the host matches the mode selected by the default or alternate operating mode parameters.

Position
The *Position* register is used only when the drive's operating mode is *Host Position Mode*. It specifes the absolute target position for a 'simplified' move profile which utilizes the *Host.Velocity* and *Host.Acceleration* registers as limiting motion values to acquire the target position. Additionally, the *Host.Current* register is *always* used as the maximum current command allowed. The move profile is a simple absolute move without the secondary motion or termination options available to a standard move. The move has no dedicated move active or in-position status flags associated with it, though the general In Position status will become active when the target position is acquired.

Velocity
The *Velocity* register sets the limiting profile velocity used to achieve position in the *Host Position* operating mode and the target velocity for the *Host Velocity* operating mode. For the *Host Position* operating mode, the velocity specified should be an unsigned (*UVEL32*) value. When the *Velocity* register specifies the velocity target for the *Host Velocity* command mode, the value may be a signed (*VEL32*) value.

Acceleration
The *Acceleration* register sets the maximum acceleration that will be commanded in both the *Host Position* and *Host Velocity* modes of operation.

Current
The *Current* register sets the current command used in the *Host Current* and *Host Current Lock* modes of operation, and the maximum current that will be commanded in the *Host Position* and *Host Velocity* modes of operation. For the *Host Position* and *Host Velocity* operating modes, the maximum current value specified should be an unsigned (*UCUR16*) value. When the *Current* register specifies the current command for the *Host Current* and *Host Current Lock* command modes, the value may be a signed (*CUR16*) value.

Voltage
The *Voltage* register sets the voltage command used in the *Host Voltage* and *Host Voltage Lock* modes of operation.

InputInhibits
Digital inputs one through eight may be inhibited by setting the corresponding flag in *InputInhibits*. While an input is inhibited it cannot activate any input event assigned to it.

								II8	**II7**	**II6**	**II5**	**II4**	**II3**	**II2**	**II1**
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

II1 - inhibit digital input 1
II2 - inhibit digital input 2
II3 - inhibit digital input 3
II4 - inhibit digital input 4
II5 - inhibit digital input 5
II6 - inhibit digital input 6
II7 - inhibit digital input 7
II8 - inhibit digital input 8

OutputInhibits
The drive's hardware outputs and LEDs normally reflect the status of any output status event assigned to them. When the output or LED is inhibited, any output status event assigned will not affect the output. This feature allows the host to take direct control of the hardware outputs (through *Host.Outputs*) and is normally used only for test purposes.

				LI4	**LI3**	**LI2**	**LI1**					**OI4**	**OI3**	**OI2**	**OI1**
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

OI1 - inhibit digital output 1
OI2 - inhibit digital output 2
OI3 - inhibit digital output 3
OI4 - inhibit digital output 4
LI1 - inhibit LED output 1
LI2 - inhibit LED output 2
LI3 - inhibit LED output 3
LI4 - inhibit LED output 4

Outputs
The *Outputs* register allows the host to directly control the status of the hardware outputs and LEDs. Outputs are normally also activated through the output events assigned to the digital outputs. If the host desires exclusive control, it may either 'un-assign' the digital output(s) or inhibit the hardware event(s) from activating the outputs through the *OutputInhibits* register.

				L4	**L3**	**L2**	**L1**					**O4**	**O3**	**O2**	**O1**
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

OI1 - set digital output 1
OI2 - set digital output 2
OI3 - set digital output 3
OI4 - set digital output 4
LI1 - set LED output 1
LI2 - set LED output 2
LI3 - set LED output 3
LI4 - set LED output 4

FactoryTest
This register is used for multiple purposes during drive test and commissioning. It is not used internally during normal drive operation and is available as a general purpose storage register.

InputEvents.x
The input event group maps give the host direct input into the input event system. Any flags set in these registers will activate the associated input event. Events are normally also activated through the input events assigned to the digital inputs. If the host desires exclusive control, it may either 'un-assign' the digital input(s) or inhibit the hardware input(s) through the *InputInhibits* register.

System Command Register Table

ID	Name	Type*	Description
4000	SecurityKey	UINT16	Security key
4001	Command	FLAGS	System command / response flags

See Appendix A for details on data Types

SecurityKey

The *SecurityKey* register is used to protect the drive from accidental or reckless system commands written to the *Command* register that may corrupt drive data and render the drive unusable. System commands for which security is required will check the *SecurityKey* before performing the requested action. If the key value is not *0x5AA5*, the command will be ignored and the *ERROR* flag will be set in the *Command* register. It is recommended that *SecurityKey* be written to zero (or any value other than *0x5AA5*) after the secure command is performed to remove the risk of future errors.

Although the *SecurityKey* operation is documented here, its use is ***NOT*** expected to be required by the user. The writing of *SecurityKey* is normally required only by the factory during the commissioning of the drive or by the *Tritex* user interface software for special operations. The factory should be consulted before attempting any operation involving the *SecurityKey*.

Command

The Command register is used to execute various internal system functions within the drive.

ERROR	IF	IU	FLSH[1]							ILOG[1]	HUDT	IDIAG	SVFP[1]	SVUP	RESTART
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

[1]Requires *SecurityKey*

RESTART

The *RESTART* flag forces a software power-on restart of the drive.

SVUP

The *SVUP* flag saves the current runtime values of all user parameters to non-volatile memory. The runtime values will be initialized to this state during a future power-on or software restart condition.

SVFP

The *SVFP* flag saves the current runtime values of all factory parameters to non-volatile memory. The runtime values will be initialized to this state during a future power-on or software restart condition. This command requires that the *SecurityKey* be set before the requested action is performed and is not intended for normal user use.

IDIAG

The *IDIAG* flag will resets various internal diagnostic variables to their initial values so that new values will be captured. The default reset values are variable dependent. If the variable is monitoring the minimum value of another variable, the reset value will be the maximum value. Similarly, if monitoring the maximum value of another variable, the reset value will be the minimum value. Variables monitoring counters or other variables will be reset to zero. Most variables affected are non-writable status variables and the *IDIAG* command is the only method available to directly modify their value.

The following internal variables are reset through the IDIAG command flag:
Status.Position.MinError, Status.Position.MaxError, Status.Velocity.MinError, Status.Velocity.MaxError, Status.Position.SineMin, Status.Position.SineMax, Status.Position.CosineMin, Status.Position.CosineMax, Status.Iloop.MaxImagnitude, Status.Iloop.MaxVbus, Status.Iloop.MinScanTime, Status.Iloop.MaxScanTime, Status.Iloop.MinExecutionTime, Status.Iloop.MaxExecutionTime, Status.HrSystem.IdLatchedValue, Status.HrSystem.IdLatchedCount.

HUDT
The *HUDT* flag forces a re-read of the Absolute Hall Position Feedback status variables which are not continually monitored by the system. This command is a convenience for internal factory testing and not intended for the user.

ILOG
The *ILOG* flag initializes the non-volatile status and fault logs to initial values and saves the initial values to non-volatile memory. This command requires that the *SecurityKey* be set before the requested action is performed and is not intended for normal user use.

FLSH
The *FLSH* flag forces the drive to exit normal drive operation and start execution of the firmware's *BOOT* code. This command requires that the *SecurityKey* be set before the requested action is performed and is not intended to be used during normal operation. The *BOOT* code contains the algorithms for reprogramming *DSP* code blocks and is not normally modified during the firmware upgrade process. The *FLSH* command is used during a firmware upgrade procedure to force the *DSP* to execute from *BOOT* code while reprogramming the application code blocks.

IU
The *IU* flag will initialize all user parameter registers to default values. Only the *RAM* image of the user parameters are initialized – the new values are *NOT* saved to non-volatile memory.

IF
The *IF* flag will initialize all factory parameter registers to default values. Only the *RAM* image of the user parameters are initialized – the new values are *NOT* saved to non-volatile memory.

ERROR
The *ERROR* flag is set if any of the specified actions cannot be executed or if an error occurred during execution of a command and the command failed to complete properly. The *ERROR* flag, once set, is never cleared internally. The user may read the value of the *Command* register before overwriting it with a new value to verify that the value is zero and the requested operation has succeeded.

Scope Control Registers

All registers in the *Scope Control Register Table* are read-write and are initialized to zero at start-up. Once the Scope Control Register Table has been properly configured and the *Control* register written to start data acquisition, Scope Control Register values must not be changed – changing any word in the table will clear the *ACTIVE* flag in the *Status* register of the *Scope Status Register Table*, halting any active scope data acquisition.

Scope Control Register Table

ID	Name	Type*	Description
4100	Channel1.ID	UINT16	Channel 1 variable id
4101	Channel1.Flags	FLAGS	Channel 1 variable flags
4102	Channel2.ID	UINT16	Channel 2 variable id
4103	Channel2.Flags	FLAGS	Channel 2 variable flags
4104	Channel3.ID	UINT16	Channel 3 variable id
4105	Channel3.Flags	FLAGS	Channel 3 variable flags
4106	Channel4.ID	UINT16	Channel 4 variable id
4107	Channel4.Flags	FLAGS	Channel 4 variable flags
4108	Trigger.ID	UINT16	Trigger variable id
4109	Trigger.Flags	FLAGS	Trigger variable flags
4110	Trigger.Level	INT32	Trigger level
4112	PreTrigger	UINT16	Pre-trigger buffer size [UPDATES]
4114	UpdateRate	UINT16	Scope update rate [SYS_TICKS]
4115	Control	FLAGS	Control command flags

See Appendix A for details on data Types

Channel1.ID
Channel2.ID
Channel3.ID
Channel4.ID
The *Channel ID* registers specify the *MODBUS* identifier of the variable to be monitored on the corresponding digital scope channel. The size (number of 16-bit words) of the channel variable's data collected in each scope update record is specified in the associated *ChannelX.Flags* register.

Trigger.ID
The *Trigger.ID* register specifies the *MODBUS* identifier of the variable whose value will be used to start (*trigger*) the collection of data from enabled scope channels. The size (number of 16-bit words) of the trigger variable's data is specified in the associated *Trigger.Flags* register. A valid Trigger.ID is required only if the *TR* or *TF* flag is set in the *Control* word.

Channel1.Flags
Channel2.Flags
Channel3.Flags
Channel4.Flags
Trigger.Flags
The variable *Flags* registers provide additional information about the type of variable assigned to the channel or trigger variables as specified in the following table.

														D
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

D
The *D* flag indicates that the variable is a double word (32-bit) variable. Every scope update record will require two words of storage for every double word channel variable enabled in the Control register and a single word of storage for every single word channel variable enabled in the Control.

Trigger.Level
Trigger.Level specifies the value of the *Trigger.ID* variable at which the scope will begin data acquisition of the channels selected in the *Control* word. *Trigger.Level* is used only if the *TR* or *TF* flag is set in the *Control* word. The size of *Trigger.Level* is determined by the *D* flag in *Trigger.Flags*. The default *Trigger.Level* value is zero and does not need to be specified specified unless scope data acquisition is being triggered and the default value of zero is unacceptable.

PreTrigger
The value of *PreTrigger* specifies the maximum number of scope data record acquisitions (updates) that will be saved in the scope data buffer before the scope's trigger criteria was met. When the scope triggers, there will be *PreTrigger* number of data acquisition records immediately available in the buffer. If the scope has not been active long enough to acquire all of the *PreTrigger* records, the early record values will be zero. The default *PreTrigger* value is zero and does not need to be specified unless scope data acquisition is being triggered and data values before the trigger point are desired.

UpdateRate
UpdateRate is automatically set to the default value of 100 SYS_TICKS (10 ms) whenever its current value is zero and any data is written to the *Scope Control Register Table*. *UpdateRate* will therefore always contain a valid non-zero data acquisition period and does not need to be specified unless the default period is unacceptable.

Control
The *Control* word is used to start and stop scope data acquisition, specifying the data channels to monitor and whether or not the acquisition should be started on the rising or falling edge of the trigger. The *Control* word should normally be the last control value written to the *Scope Control Register Table*.

SS										TR	TF	CH4	CH3	CH2	CH1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CH1
CH2
CH3
CH4
The *CHX* flags enable the individual scope channels for data acquisition. *CHX* flags will be automatically cleared whenever any data is changed in the *Scope Control Register Table* and the associated *ChannelX.ID* does not specify a valid *MODBUS* identifier.

TR
When the *TR* (rising edge trigger) flag is set, data acquisition will begin when the current value of the *Trigger.ID* variable transitions from a value less than or equal to *Trigger.Level* to a value greater than *Trigger.Level*. The *TR* flag is cleared automatically whenever any data is changed in the *Scope Control Register Table* and *Trigger.ID* is not a valid *MODBUS* identifier.

TF
When the *TF* (falling edge trigger) flag is set, data acquisition will begin when the current value of the *Trigger.ID* variable transitions from a value greater than or equal to *Trigger.Level* to a value less than *Trigger.Level*. The *TF* flag is cleared automatically whenever any data is changed in the *Scope Control Register Table* and *Trigger.ID* is not a valid *MODBUS* identifier.

SS
The *SS* flag is used to start and stop scope data record acquisition.

Scope Status Registers

All registers in the *Scope Status Register Table* are read-only. After configuring and starting data acquisition through the *Scope Control Register Table*, the status registers may be used to monitor the scope's progress and to retrieve the acquired data.

Scope Status Register Table

ID	Name	Type*	Description
400	Status	FLAGS	Scope status flags
401	RecordCount	UINT16	# of records captured
402	MaxRecords	UINT16	# of buffer records
403	RecordSize	UINT16	Size of single update record
404	BufferSize	UINT16	Size of data buffer
405	Timestamp	UINT16	Time of data acquisition [SYS_TICKS]
406	Channel1	INT32	Last Channel 1 value
408	Channel2	INT32	Last Channel 2 value
410	Channel3	INT32	Last Channel 3 value
412	Channel4	INT32	Last Channel 4 value
414	Trigger	INT32	Current trigger value

See Appendix A for details on data Types

Status

ACTIVE	**FULL**									**TR**	**TF**				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

ACTIVE
The *ACTIVE* flag indicates that the scope data acquisition is active. The flag is cleared when data is written to the *Scope Control Register Table* SCOPE_STATUS_ACTIVE

FULL
The *FULL* flag indicates that the data acquisition buffer is full. The flag is cleared when scope acquisition is started (*Control* register written with the *SS* flag set). The flag is set when the scope has been triggered and the *RecordCount* is equal to *MaxRecords*.

TR
The *TR* flag indicates that data acquisition was triggered on the rising edge of the trigger variable. The flag is cleared when scope acquisition is started (*Control* register written with the *SS* flag set) and set when the *TR* (rising edge trigger) flag is set in the *Control* register and the value of the *Trigger.ID* variable transitions from a value less than or equal to *Trigger.Level* to a value greater than *Trigger.Level*.

TF
The *TF* flag indicates that data acquisition was triggered on the rising edge of the trigger variable. The flag is cleared when scope acquisition is started (*Control* register written with the *SS* flag set) and set when the *TF* (falling edge trigger) flag is set in the *Control* register and the value of the *Trigger.ID* variable transitions from a value greater than or equal to *Trigger.Level* to a value less than *Trigger.Level*.

RecordCount
RecordCount specifies the number of data records available for reading from the data buffer. *RecordCount* is initialized to zero when scope acquisition is started (*Control* register written with the *SS* flag set) and is set to the *PreTrigger* value when the scope is triggered. RecordCount is incremented on each data record acquisition (every *UpdateRate* ticks of SYS_CLOCK) after triggering.

MaxRecords
MaxRecords specifies the total number of data records that the data buffer can hold. *MaxRecords* is initialized when scope acquisition is started (*Control* register written with the *SS* flag set). If *RecordSize* is zero, *MaxRecords* is set to zero, otherwise the value of *MaxRecords* is calculated as: *MaxRecords* = *BufferSize* / *RecordSize.*

RecordSize
RecordSize specifies the size, in 16-bit words, of each data acquisition update record. *RecordSize* is updated automatically from the *CHX* enable flags in the *Control* register and the data sizes specified for the channel variables in the *ChannelX.Flags* registers whenever any data is changed in the *Scope Control Register Table.*

BufferSize
BufferSize specifies the total size, in 16-bit words, of the scope data buffer. The *TritexII* buffer size is normally *2K* (2048) words.

Timestamp
Timestamp is a free-running counter incremented every data acquisition update (i.e. every *UpdateRate* SYS_CLOCK ticks). When the scope is not being used in trigger mode, *Timestamp* provides a relative indication of the acquisition time of the current data record.

Channel1
Channel2
Channel3
Channel4
The channel registers provide the latest values observed on the scope data channels. The channel data is valid only if the associated *ChannelX.ID* value specifies a valid MODBUS variable.

Trigger
If the scope is active and *Trigger.ID* specifies a valid MODBUS register, *Trigger* will be updated at the *UpdateRate* with the current value of the *Trigger.ID* variable.

Logs

The drive automatically maintains data logs in non-volatile memory to track status information that may change at run-time. Data integrity of each log is maintained in a manner similar to that used for parameter data and is validated at power-up. Unlike parameter data, however, log information is not considered *critical* and therefore no fault or warning condition is indicated if a log is found to be corrupt at start-up. If a data validation error occurs the log will simply be initialized to a default starting state. (Note that a data log error should not occur after the factory commissioning of a drive - logged data should remain valid for the life of the drive.)

Data log registers, like other status information, are read-only. Data is updated (written) internally when necessary and the new data is automatically saved to non-volatile memory. Logs may be initialized default starting values through specific control commands.

Status Log

The *Status Log* maintains information on drive usage and worst-case stress values observed in non-volatile memory. Data changes are automatically updated in non-volatile memory as necessary.

Status Log Register Table

ID	Name	Type*	Description
734	Flags	FLAGS	Status log flags
735	PowerUpCount	UINT16	Number of times drive has powered up
736	PwmRunTime	UINT32	Total time PWM has been active [MINUTES]
738	AbsPositionOffset	POS32	Absolute position offset
740	MaxCurrent	UCUR32	Max current magnitude observed
742	MaxVbus	UVOLT32	Max bus voltage observed
744	MaxBoardTemp	INT32	Max PCB temperature observed [11.21 DEG C]
746	MaxActuatorTemp	INT32	Max actuator temperature observed [11.21 DEG C]

See Appendix A for details on data Types

Flags

														STC	UNHOMED
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

UNHOMED

This flag is used only when the *Absolute Hall Feedback* option is selected for the actuator's position feedback device to indicate that the absolute position reference frame has not yet been established. The flag is set when a new reference frame is established (at the completion of a move that establishes a reference frame or upon the rising edge of a *Define Home* input event). The flag is cleared when a move that will establish a new reference upon completion is initiated, and also under certain fault conditions that indicate the loss of reference frame (*Position Tracking* fault, e.g.).

STC Startup Complete

This flag saves and restores the *Startup Complete* status between power-ups. The flag is set (and the *Status Log* is resaved to non-volatile memory) when a move with the *Set Startup Complete* option finishes execution. The flag is cleared when *Startup* is initiated, on the rising edge of *Reset Startup Complete*, and when the direction polarity flag (*DIR*) in configuration options is modified.

PowerUpCount
The *PowerUpCount* tracks the number of times that the drive has powered-up.

PwmRunTime
PwmRunTime tracks the total time the drive's *PWM* (actuator phase voltage) circuitry has been active. The *PwmRunTime* is, effectively, the total time that the drive has been enabled with addition of any *PWM* time required for brake activation and deactivation.

AbsPositionOffset
AbsPositionOffset is saved in the *Status Log* only when the *Absolute Hall Feedback* option is selected for the actuator's position feedback and specifies the offset between the free-running count read from the *Absolute Hall Feedback* board and the absolute reference frame.

MaxCurrent
MaxCurrent tracks the maximum magnitude of the current feedback vector observed.

MaxVbus
MaxCurrent tracks the maximum *DC Bus* voltage observed.

MaxBoardTemp
MaxBoardTemp tracks the maximum *PCB* (*Printed Circuit Board*) temperature observed.

MaxActuatorTemp
MaxActuatorTemp tracks the maximum actuator thermistor temperature observed for actuators that have thermistor temperature feedback. The value is not updated and will be a constant zero for actuators that use only digital switch temperature fault indicators.

Fault Log

The *Fault Log* maintains information of *FAULT* occurrences in non-volatile memory. Data changes are automatically updated in non-volatile memory as necessary.

Fault Log Register Table

ID	Name	Type*	A	N	S	Description
782	FaultCount	UINT16	RO	16	FL	
782	FaultCount.0	UINT16	RO	1	FL	Count of faults observed for FAULT.0
783	FaultCount.1	UINT16	RO	1	FL	Count of faults observed for FAULT.1
784	FaultCount.2	UINT16	RO	1	FL	Count of faults observed for FAULT.2
785	FaultCount.3	UINT16	RO	1	FL	Count of faults observed for FAULT.3
786	FaultCount.4	UINT16	RO	1	FL	Count of faults observed for FAULT.4
787	FaultCount.5	UINT16	RO	1	FL	Count of faults observed for FAULT.5
788	FaultCount.6	UINT16	RO	1	FL	Count of faults observed for FAULT.6
789	FaultCount.7	UINT16	RO	1	FL	Count of faults observed for FAULT.7
790	FaultCount.8	UINT16	RO	1	FL	Count of faults observed for FAULT.8
791	FaultCount.9	UINT16	RO	1	FL	Count of faults observed for FAULT.9
792	FaultCount.10	UINT16	RO	1	FL	Count of faults observed for FAULT.10
793	FaultCount.11	UINT16	RO	1	FL	Count of faults observed for FAULT.11
794	FaultCount.12	UINT16	RO	1	FL	Count of faults observed for FAULT.12
795	FaultCount.13	UINT16	RO	1	FL	Count of faults observed for FAULT.13
796	FaultCount.14	UINT16	RO	1	FL	Count of faults observed for FAULT.14
797	FaultCount.15	UINT16	RO	1	FL	Count of faults observed for FAULT.15
12502	FaultCount.16	UINT16	RO	1	FL	Count of faults observed for FAULT.16
12503	FaultCount.17	UINT16	RO	1	FL	Count of faults observed for FAULT.17
12504	FaultCount.18	UINT16	RO	1	FL	Count of faults observed for FAULT.18
12505	FaultCount.19	UINT16	RO	1	FL	Count of faults observed for FAULT.19
12506	FaultCount.20	UINT16	RO	1	FL	Count of faults observed for FAULT.20
12507	FaultCount.21	UINT16	RO	1	FL	Count of faults observed for FAULT.21
12508	FaultCount.22	UINT16	RO	1	FL	Count of faults observed for FAULT.22
12509	FaultCount.23	UINT16	RO	1	FL	Count of faults observed for FAULT.23
12510	FaultCount.24	UINT16	RO	1	FL	Count of faults observed for FAULT.24
12511	FaultCount.25	UINT16	RO	1	FL	Count of faults observed for FAULT.25
12512	FaultCount.26	UINT16	RO	1	FL	Count of faults observed for FAULT.26
12513	FaultCount.27	UINT16	RO	1	FL	Count of faults observed for FAULT.27
12514	FaultCount.28	UINT16	RO	1	FL	Count of faults observed for FAULT.28
12515	FaultCount.29	UINT16	RO	1	FL	Count of faults observed for FAULT.29
12516	FaultCount.30	UINT16	RO	1	FL	Count of faults observed for FAULT.30
12517	FaultCount.31	UINT16	RO	1	FL	Count of faults observed for FAULT.31
798	PowerUpCount	UINT16	RO	16	FL	
798	PowerUpCount.0	UINT16	RO	1	FL	Power-up count at last FAULT.0 fault
799	PowerUpCount.1	UINT16	RO	1	FL	Power-up count at last FAULT.1 fault
800	PowerUpCount.2	UINT16	RO	1	FL	Power-up count at last FAULT.2 fault
801	PowerUpCount.3	UINT16	RO	1	FL	Power-up count at last FAULT.3 fault
802	PowerUpCount.4	UINT16	RO	1	FL	Power-up count at last FAULT.4 fault
803	PowerUpCount.5	UINT16	RO	1	FL	Power-up count at last FAULT.5 fault
804	PowerUpCount.6	UINT16	RO	1	FL	Power-up count at last FAULT.6 fault
805	PowerUpCount.7	UINT16	RO	1	FL	Power-up count at last FAULT.7 fault
806	PowerUpCount.8	UINT16	RO	1	FL	Power-up count at last FAULT.8 fault
807	PowerUpCount.9	UINT16	RO	1	FL	Power-up count at last FAULT.9 fault
808	PowerUpCount.10	UINT16	RO	1	FL	Power-up count at last FAULT.10 fault
809	PowerUpCount.11	UINT16	RO	1	FL	Power-up count at last FAULT.11 fault
810	PowerUpCount.12	UINT16	RO	1	FL	Power-up count of last FAULT.12 fault
811	PowerUpCount.13	UINT16	RO	1	FL	Power-up count at last FAULT.13 fault

812	PowerUpCount.14	UINT16	RO	1	FL	Power-up count at last FAULT.14 fault
813	PowerUpCount.15	UINT16	RO	1	FL	Power-up count at last FAULT.15 fault
12518	PowerUpCount.16	UINT16	RO	1	FL	Power-up count at last FAULT.16 fault
12519	PowerUpCount.17	UINT16	RO	1	FL	Power-up count at last FAULT.17 fault
12520	PowerUpCount.18	UINT16	RO	1	FL	Power-up count at last FAULT.18 fault
12521	PowerUpCount.19	UINT16	RO	1	FL	Power-up count at last FAULT.19 fault
12522	PowerUpCount.20	UINT16	RO	1	FL	Power-up count at last FAULT.20 fault
12523	PowerUpCount.21	UINT16	RO	1	FL	Power-up count at last FAULT.21 fault
12524	PowerUpCount.22	UINT16	RO	1	FL	Power-up count at last FAULT.22 fault
12525	PowerUpCount.23	UINT16	RO	1	FL	Power-up count at last FAULT.23 fault
12526	PowerUpCount.24	UINT16	RO	1	FL	Power-up count at last FAULT.24 fault
12527	PowerUpCount.25	UINT16	RO	1	FL	Power-up count at last FAULT.25 fault
12528	PowerUpCount.26	UINT16	RO	1	FL	Power-up count at last FAULT.26 fault
12529	PowerUpCount.27	UINT16	RO	1	FL	Power-up count at last FAULT.27 fault
12530	PowerUpCount.28	UINT16	RO	1	FL	Power-up count of last FAULT.28 fault
12531	PowerUpCount.29	UINT16	RO	1	FL	Power-up count at last FAULT.29 fault
12532	PowerUpCount.30	UINT16	RO	1	FL	Power-up count at last FAULT.30 fault
12533	PowerUpCount.31	UINT16	RO	1	FL	Power-up count at last FAULT.31 fault
814	PwmTime	UINT16	RO	32	FL	
814	PwmTime.0	UINT32	RO	2	FL	PWM run time at last FAULT.0 fault
816	PwmTime.1	UINT32	RO	2	FL	PWM run time at last FAULT.1 fault
818	PwmTime.2	UINT32	RO	2	FL	PWM run time at last FAULT.2 fault
820	PwmTime.3	UINT32	RO	2	FL	PWM run time at last FAULT.3 fault
822	PwmTime.4	UINT32	RO	2	FL	PWM run time at last FAULT.4 fault
824	PwmTime.5	UINT32	RO	2	FL	PWM run time at last FAULT.5 fault
826	PwmTime.6	UINT32	RO	2	FL	PWM run time at last FAULT.6 fault
828	PwmTime.7	UINT32	RO	2	FL	PWM run time at last FAULT.7 fault
830	PwmTime.8	UINT32	RO	2	FL	PWM run time at last FAULT.8 fault
832	PwmTime.9	UINT32	RO	2	FL	PWM run time at last FAULT.9 fault
834	PwmTime.10	UINT32	RO	2	FL	PWM run time at last FAULT.10 fault
836	PwmTime.11	UINT32	RO	2	FL	PWM run time at last FAULT.11 fault
838	PwmTime.12	UINT32	RO	2	FL	PWM run time at last FAULT.12 fault
840	PwmTime.13	UINT32	RO	2	FL	PWM run time at last FAULT.13 fault
842	PwmTime.14	UINT32	RO	2	FL	PWM run time at last FAULT.14 fault
844	PwmTime.15	UINT32	RO	2	FL	PWM run time at last FAULT.15 fault
12534	PwmTime.16	UINT32	RO	2	FL	PWM run time at last FAULT.16 fault
12536	PwmTime.17	UINT32	RO	2	FL	PWM run time at last FAULT.17 fault
12538	PwmTime.18	UINT32	RO	2	FL	PWM run time at last FAULT.18 fault
12540	PwmTime.19	UINT32	RO	2	FL	PWM run time at last FAULT.19 fault
12542	PwmTime.20	UINT32	RO	2	FL	PWM run time at last FAULT.20 fault
12544	PwmTime.21	UINT32	RO	2	FL	PWM run time at last FAULT.21 fault
12546	PwmTime.22	UINT32	RO	2	FL	PWM run time at last FAULT.22 fault
12548	PwmTime.23	UINT32	RO	2	FL	PWM run time at last FAULT.23 fault
12550	PwmTime.24	UINT32	RO	2	FL	PWM run time at last FAULT.24 fault
12552	PwmTime.25	UINT32	RO	2	FL	PWM run time at last FAULT.25 fault
12554	PwmTime.26	UINT32	RO	2	FL	PWM run time at last FAULT.26 fault
12556	PwmTime.27	UINT32	RO	2	FL	PWM run time at last FAULT.27 fault
12558	PwmTime.28	UINT32	RO	2	FL	PWM run time at last FAULT.28 fault
12560	PwmTime.29	UINT32	RO	2	FL	PWM run time at last FAULT.29 fault
12562	PwmTime.30	UINT32	RO	2	FL	PWM run time at last FAULT.30 fault
12564	PwmTime.31	UINT32	RO	2	FL	PWM run time at last FAULT.31 fault
846	RecentFault	UINT16	RO	40	FL	
846	RecentFault.0.ID	FAULT	RO	1	FL	FAULT ID of most recent fault
847	RecentFault.0.PowerUpCount	UINT16	RO	1	FL	Power up count of most recent fault
848	RecentFault.0.RunTime	UINT32	RO	2	FL	PWM run time of most recent fault
850	RecentFault.1.ID	FAULT	RO	1	FL	
851	RecentFault.1.PowerUpCount	UINT16	RO	1	FL	

852	RecentFault.1.RunTime	UINT32	RO	2	FL	
854	RecentFault.2.ID	FAULT	RO	1	FL	
855	RecentFault.2.PowerUpCount	UINT16	RO	1	FL	
856	RecentFault.2.RunTime	UINT32	RO	2	FL	
858	RecentFault.3.ID	FAULT	RO	1	FL	
859	RecentFault.3.PowerUpCount	UINT16	RO	1	FL	
860	RecentFault.3.RunTime	UINT32	RO	2	FL	
862	RecentFault.4.ID	FAULT	RO	1	FL	
863	RecentFault.4.PowerUpCount	UINT16	RO	1	FL	
864	RecentFault.4.RunTime	UINT32	RO	2	FL	
866	RecentFault.5.ID	FAULT	RO	1	FL	
867	RecentFault.5.PowerUpCount	UINT16	RO	1	FL	
868	RecentFault.5.RunTime	UINT32	RO	2	FL	
870	RecentFault.6.ID	FAULT	RO	1	FL	
871	RecentFault.6.PowerUpCount	UINT16	RO	1	FL	
872	RecentFault.6.RunTime	UINT32	RO	2	FL	
874	RecentFault.7.ID	FAULT	RO	1	FL	
875	RecentFault.7.PowerUpCount	UINT16	RO	1	FL	
876	RecentFault.7.RunTime	UINT32	RO	2	FL	
878	RecentFault.8.ID	FAULT	RO	1	FL	
879	RecentFault.8.PowerUpCount	UINT16	RO	1	FL	
880	RecentFault.8.RunTime	UINT32	RO	2	FL	
882	RecentFault.9.ID	FAULT	RO	1	FL	FAULT ID of least recent fault
883	RecentFault.9.PowerUpCount	UINT16	RO	1	FL	Power up count of least recent fault
884	RecentFault.9.RunTime	UINT32	RO	2	FL	PWM run time of least recent fault

See Appendix A for details on data Types

FaultCount
The total number of times a particular fault/warning has been logged is tracked in the *FaultCount* array. The bit number of the fault in the *FAULT* type specifies a particular fault's index in the *FaultCount* array.

PowerUpCount
The drive's power-up-count at the last occurrence of a particular fault/warning is tracked in the *PowerUpCount* array. The bit number of the fault in the *FAULT* type specifies a particular fault's index in the *PowerUpCount* array.

PwmTime
The total time the drive's *PWM* circuitry (phase voltage) has been active at last occurrence of a particular fault/warning is tracked in the *PwmTime* array. The bit number of the fault in the *FAULT* type specifies a particular fault's index in the *PwmTime* array.

RecentFault.N.ID
RecentFault.N.PowerUpCount
RecentFault.N.RunTime
The ten most recent faults/warnings that have been logged are available in the *RecentFault* array. The array is kept in chronological order with *RecentFault.0* being the most recent and *RecentFault.9* the least recent. Each entry contains the drive's power-up-count and *PWM* active time at the occurrence of the fault specified by the *FAULT* bitmap in the record's *ID*. To reduce redundant entries and preserve more useful fault history data, a new *RecentFault* entry will not be created for a fault whose data would have exactly matched the most recent record.

Appendix A - Types

This appendix defines the various data types used in the *Register Tables*.

All *MODBUS* registers contain sixteen bits of data. Data that requires more than sixteen bits uses multiple consecutive registers to store the data value. Integer data requiring more than sixteen bits is stored in *Little Endian* word order – the least significant data word is stored first (at the lowest register *ID*). All integer data is stored in two's complement format.

The drive's *DSP* controller does not directly support floating-point math. For efficiency, the drive uses fixed-point numeric representation for real numbers. A fixed-point number specifies the number of binary bits used to represent the integer part of the value and the number of binary bits used to represent the fractional part of the value. A fixed-point format of *2.14*, for example, specifies a 16-bit word that uses two bits to represent the integer part and fourteen bits to represent the fractional part. A normal integer (in the range 0 to 65535) uses all 16 bits to represent the integer part of the value and could be specified as a *16.0* fixed-point number.

Specifying a particular data type for a register is a simple convenience to better describe the format and internal use of the data. All data could be typed only as *N-bit* (*N* = 16, 32, 128, etc) or *N-word* (*N* = 1, 2, etc) describing the number of bits or registers required to store the data. Specifying a data type of *UINT16* (instead of just *'16'*) is more descriptive and carries the additional information that the register is a non-negative integer in the range of 0 to 65535. Taking this one step further, describing a register as type *UCUR16* instead of the more generic *UINT16* specifies that the value should be interpreted as an unsigned 9.7 fixed-point amperage value.

The structured data type is used for data items composed of multiple elements of different type and for data of a specific format that appears multiple times in the *Register Tables*. For each structured data type, the number of 16-bit words required to store a single item of the type is shown along with a base data type, if applicable.

Simple Types

The table below specifies the simple register data types used in the *Register Tables*. Simple data types are used for numeric data types of 16 or 32 bits. The fixed-point format, units, and numeric ranges are also specified if applicable.

Simple Data Types

Type	Format	Units	Range		Resolution	Description
			min	max		
FLAGS						Bit-mapped data
ENUM						Integer range of data
UINT16	16.0		0	65535	1	Unsigned 16-bit integer
INT16	16.0		-32768	32767	1	Signed 16-bit integer
UINT32	32.0		0	4294967295	1	Unsigned 32-bit integer
INT32	32.0		-2147483648	2147483647	1	Signed 32-bit integer
POS32	16.16	REVS	-32768.0	32767.9999847412109375	0.0000152587890625	32-bit position/distance
UPOS32	16.16	REVS	0.0	65535.9999847412109375	0.0000152587890625	Unsigned 32-bit position/distance
UPOS16	0.16	REVS	0	0. 9999847412109375	0.0000152587890625	16-bit rev position
VEL32	8.24	REVS/S	-128.0	127.99999994039535522460937 5	0.0000000596046447753906 25	32-bit signed velocity
VEL16	8.8	REVS/S	-128.0	127.99609375	0.00390625	16-bit signed velocity
UVEL32	8.24	REVS/S	0	255.99999994039535522460937 5	0.0000000596046447753906 25	32-bit unsigned velocity
UVEL16	8.8	REVS/S	0	255.99609375	0.00390625	16-bit unsigned velocity
ACC32	12.20	REVS/S/S	-2048.0	2047.99999904632568359375	0.00000095367431640625	32-bit signed acceleration
UACC32	12.20	REVS/S/S	0	4095.99999904632568359375	0.00000095367431640625	32-bit unsigned acceleration

CUR32	9.23	AMPS	-256.0	255.99999988079071044921875	0.000000119209289550781250	32-bit signed current
CUR16	9.7	AMPS	-256.0	255.9921875	0.0078125	16-bit signed current
UCUR32	9.23	AMPS	0	511.99999988079071044921875	0.000000119209289550781250	32-bit unsigned current
UCUR16	9.7	AMPS	0	511.9921875	0.0078125	16-bit unsigned current
VOLT32	11.21	VOLTS	-1024.0	1023.999999523162841796875	0.000000476837158203125	32-bit signed voltage
VOLT16	11.5	VOLTS	-1024.0	1023.96875	0.03125	16-bit signed voltage
UVOLT32	11.21	VOLTS	0	2047.999999523162841796875	0.000000476837158203125	32-bit unsigned voltage
UVOLT16	11.5	VOLTS	0	2047.96875	0.03125	16-bit unsigned voltage

STR16

Type	Base Type	Words	Description
STR16	UINT16	16	16 character string

The *STR16* type is used to hold an ASCII character string. Each word encodes a printable ASCII character in the low eight bits (the high eight bits of each word are unused). The STR16 data type is used to hold displayable text strings used for identification and informational purposes only.

FAULT

Type	Base Type	Words	Description
FAULT	FLAGS	2	Bit-mapped fault flags

The *FAULT* type is a bit-mapped word with each bit representing a specific fault that may occur in drive. The *FAULT* type appears in multiple *Register Tables* and is used for both parameter set-up and status information.

RS	**FP**	**UP**	**HB**	**IH**	**COM**	**LOS**	**AT**	**BT**	**FE**	**VHIGH**	**VLOW**	**MV**	**POS**	**IC**	**IPK**
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

															LVW
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

IPK – Peak Current
A *Peak Current* fault occurs when the instantaneous magnitude of the current feedback vector exceeds the maximum allowable value. The maximum value is set at the factory and is available in the *Itrip* parameter of the *Factor Limits Register Table*. *Peak Current* faults may be caused by a short circuit condition, sudden decelerations (hitting a hard stop), or poor tuning that results in unstable control loop operation.

IC – Continuous Current
A *Continuous Current* fault occurs when the absolute value of the average feedback current of the drive exceeds the drive's continuous current rating. The continuous current rating of the drive is set at the factory and is available in the *Icontinuous* parameter of the *Factor Limits Register Table*.

POS – Position Tracking
A *Position Tracking* fault is generated internally when the drive cannot satisfactorily monitor the drive's feedback position. The possible causes of a *Position Tracking* fault will vary with type of position feedback device being used on the actuator. This fault will clear the *Homed* output status event and the drive's absolute position reference frame may need to be reestablished.

MV – Move Termination
A move completes (terminates) upon occurrence of an input event, a current feedback threshold value, or completion of the travel distance to its new position. Any or all of these move termination options may be specified in the move set-up parameters to generate a *Move Termination* fault.

VLOW – Low DC Bus Voltage
A *Low DC Bus Voltage* fault occurs when the drive is enabled and the DC bus voltage falls below the minimum allowable voltage. The minimum voltage is set at the factory and is available in the *LowVoltageTripLevel* parameter of the *Factor Limits Register Table*. This fault is conditional on the drive being enabled since the drive's logic (and *DSP*) may remain powered while voltage is removed from the DC bus.

VHIGH – High DC Bus Voltage
A *High DC Bus Voltage* fault occurs whenever the DC bus voltage rises above the maximum allowable voltage. The maximum voltage is set at the factory and is available in the *HighVoltageTripLevel* parameter of the *Factory Limits Register Table*. *High DC Bus Voltage* faults may occur when bus voltage is elevated from the motor regeneration energy. Possible solutions are slower accelerations or other methods of storing or shunting the energy by connecting an external braking resister. Note that, unlike the *Low DC Bus Voltage* fault, a *High DC Bus Voltage* fault may occur even if the drive is disabled.

FE – Positional Following Error
A *Positional Following Error* fault occurs when the absolute value of the difference between drive's commanded and feedback positions remains greater than the maximum allowable following error for a specified amount of time. The time period allows for some hysteresis and avoids spurious faults during rapid accelerations or momentarily high current loading due to load change or stiction. The maximum error and its associated time period are user parameters described in the *Limits Register Table*.

BT – Board Temperature
A *Board Temperature* fault occurs when the *PCB* (*Printed Circuit Board*) temperature exceeds maximum allowable value. The maximum value is set at the factory and is available in the *BoardTempTripLevel* parameter of the *Factor Limits Register Table*. Board Temperature faults will occur if the actuator is continuously operated above its power rating.

AT – Actuator Temperature
The drive monitors an analog signal from the actuator which is configured at the factory as the input from either a thermistor or a simple (active high) switch. If the input is configured as a thermistor, an *Actuator Temperature* fault is generated if the measured value exceeds the maximum allowable actuator temperature. The maximum value is set at the factory and is available in the *ActuatorTempTripLevel* parameter of the *Factor Limits Register Table*. If the input is configured as a simple switch, an *Actuator Temperature* fault is generated when the measured value exceeds approximately 2.4V.

LOS – Loss of Signal
A *Loss of Signal* fault is generated when an analog input detects a value that is below is low trip value or above its high trip value and the offending low or high trip detection option is enabled. If multiple analog input signals are configured to generate *Loss of Signal* faults, there is no direct way to determine which input was the source of the fault. The analog input signal values may be observed (read) if necessary.

IH – Hardware Overcurrent
A *Hardware Overcurrent* fault occurs when current sensing circuitry detects extremely high current on the DC bus. The threshold trip value is greater than current feedback values expected under normal operating conditions and may be an indication that the actuator phase windings have been shorted. The *Hardware Overcurrent* fault cannot be masked by removing this bit from the hard fault mask. A hardware overcurrent fault will always be generated if detected.

COM –Communications
A *Communications* fault occurs when a communication error occurs that has been selected to cause this fault. The selection is done via the Comms Faults Setup tab.

HB – Absolute Hall Board Battery
An *Absolute Hall Board Battery* fault will occur only if the drive is using an Absolute Hall position feedback device and the battery level of the device falls below 2.8V. The Absolute Hall position feedback circuitry requires the battery to maintain feedback position monitoring while the drive is powered down and the *Absolute Hall Board Battery* fault is an indication that the feedback board's battery requires replacement.

UP – User Parameters
If the user parameter data block in non-volatile memory appears invalid at start-up, a *UserParameters fault* is generated and the user parameters in *RAM* are initialized to default values. A *UserParameters* fault cannot be masked by removing this bit from the hard fault mask and cannot be reset through either a fault reset or the rising edge of ENABLE. The fault will be automatically cleared when user parameters are saved to non-volatile memory.

FP – Factory Parameters
If the factory parameter data block in non-volatile memory appears invalid at start-up, a *FactoryParameters fault* is generated and the factory parameters in *RAM* are initialized to default values. A *FactoryParameters* fault cannot be masked by removing this bit from the hard fault mask and cannot be reset through either a fault reset or the rising edge of ENABLE. The fault will be automatically cleared when factory parameters are saved to non-volatile memory.

RS – Restart
A *Restart* fault indicates that the drive has internally restarted without powering down. This is an abnormal occurrence and should not occur unless the drive could not properly execute code. The *Restart* fault cannot be masked by removing this bit from the hard fault mask and cannot be reset through either a fault reset or the rising edge of ENABLE. A power cycle will be required to reset the drive.

LVW – Low DC Bus Voltage Warning
A *Low DC Bus Voltage* warning occurs when the drive is enabled and the DC bus voltage falls below the warning voltage as set by the user. This fault is conditional on the drive being enabled since the drive's logic (and *DSP*) may remain powered while voltage is removed from the DC bus. The warning will be automatically cleared when the DC Bus Voltage goes above the warning voltage off value which is also set by the customer.

OPMODE

Type	Base Type	Words	Description
OPMODE	ENUM	1	Command operating modes

OPMODE is an enumerated value type specifying a particular drive command operating mode. The type is used for the default and alternate operating mode parameters, the host direct operating control mode, and the current operating mode status registers. Valid *OPMODE* values and their corresponding drive operational command mode are specified in the table below.

OPMODE Enumerations

Value	Operational Command Mode
0	Disabled Mode
1	Digital Inputs Mode
2	Analog Position Mode
3	Analog Velocity Mode
4	Analog Current Mode
5	Host Position Mode[1]
6	Host Velocity Mode[1]
7	Host Current Mode[1]
8	Current Lock Mode[1,2]
9	Voltage Lock Mode[1,2]

10	Voltage Mode[1,2]

[1] Not assignable through user interface

[2] Factory test and calibration modes

Disabled Mode
The *Disabled Mode* will force the drive to become disabled. The *Disabled Mode*, when assigned to *Host.CommandMode*, is also used to indicate that *Host.CommandMode* is no longer active.

Digital Inputs Mode
The *Digital Inputs Mode* allows for the execution of the sixteen *Move* profiles through *Move Momentary* and *Move Maintained* input events.

Analog Position Mode
Analog Position Mode provides position control proportional to an analog input value. While *Analog Position Mode* is active, the drive continually attempts to achieve the absolute position command specified from the analog input within the velocity and acceleration limits set in the *Analog Position Control* registers.

Analog Velocity Mode
Analog Velocity Mode provides velocity control proportional to an analog input value. While *Analog Velocity Mode* is active, the drive continually attempts to achieve the velocity command specified from the analog input within the acceleration limit set in the *Analog Velocity Control* registers.

Analog Current Mode
Analog Current Mode provides current (torque) control proportional to an analog input value. While *Analog Current Mode* is active, the drive continually attempts to achieve the current command specified from the analog input.

Host Position Mode
While the command mode is *Host Position Mode* the drive will be forced into an internal position control mode which will continually attempt to achieve the absolute position specified by the *Host.Position* register using the *Host.Velocity*, *Host.Acceleration*, and *Host.Current* limits.

Host Velocity Mode
While the command mode is *Host Velocity Mode* the drive will be forced into an internal position control mode which will continually attempt to achieve the velocity specified by the *Host.Velocity* register using the *Host.Acceleration*, and *Host.Current* limits.

Host Current Mode
While the command mode is *Host Current Mode* the drive will be forced into an internal current control mode which will continually attempt to achieve the current feedback specified by the *Host.Current* register.

Host Poistion, *Host Velocity*, and *Host Current* operating modes require an active host controller to be useful and are not normally available for assignment to user operating modes through the *Tritex* user interface software.

Current Lock Mode
The *Current Lock Mode* is used for rotor alignment during factory commissioning. The drive will not track the electrical feedback angle but will instead force the electrical angle specified by the *Eoffset* register using the *Host.Current* register for the phase current command value.

Voltage Lock Mode
The *Voltage Lock Mode* is used for test and calibration of the actuator phase currents. The drive will not track the electrical feedback angle but will instead force the electrical angle specified by the *Eoffset* register using the *Host.Voltage* register for the phase voltage command value.

Voltage Mode
The *Voltage Mode* is used for internal factory test and calibration. The actuator phases are directly commanded with sinusoidal voltage commands of the magnitude specified by the *Host.Voltage* register. The current control loop is disabled while direct voltage mode is active.

Current Lock, *Voltage Lock*, and *Voltage* modes are not normally available for assignment to user operating modes through the *Tritex* user interface software. They are usually specified only in host control mode and are used during factory commissioning for test and alignment.

Input Event Groups

This section defines the input event flags in each of the input event status groups.

Input Event Group Types

Type	Base Type	Words	Description
IEG_MODE	FLAGS	1	Enable, mode input events
IEG_MOTION	FLAGS	1	General motion input events (jog, home, dedicated move…)
IEG_MOVE_LEVEL	FLAGS	1	Individual move maintained (level) initiates
IEG_MOVE_EDGE	FLAGS	1	Individual move momentary (edge) initiates
IEG_MOVE_TEACH	FLAGS	1	Individual move teach position
IEG_MOVE_SELECT	FLAGS	1	Binary selects
IEG_MOVE_SWITCH	FLAGS	1	Move (feed) switches (level and edge)

IEG_MODE

RESET	**BKOV**	**TSEL**	**TENA**	**H2**	**H1**			**ALT**						**EL**	**EE**
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

EE – Enable Momentary
The rising edge of the *Enable Momentary* input will reset any active hard faults. If all hard faults are successfully cleared the drive will become enabled. The drive be disabled while the input is inactive. After the occurrence of a hard fault, a new rising edge will be required to re-enable the drive (compare with *Enable Maintained*, below).

EL – Enable Maintained
The Enable Maintained operates in a manner similar to the Enable Momentary input. Like the momentary enable, the rising edge of Enable Maintained will reset faults. A new rising edge, however, is not required after resetting faults through the RESET input – the drive will automatically enable whenever no hard faults are pending and Enable Maintained is active.

ALT – Alternate Operating Mode
Sets the operational command mode of the drive to the mode specified in the *Alternate Operating Mode* register. While the *ALT* input event is inactive, the operational command mode of the drive is set to the mode specified in the *Default Operating Mode* register.

H1 – Define Home Position 1
The drive's absolute position is set to *Home.Position1* on the rising edge of the *H1* input, establishing the absolute reference frame and setting the *Homed* output status flag.

H2 – Define Home Position 2
The drive's absolute position is set to *Home.Position2* on the rising edge of the *H2* input, establishing the absolute reference frame and setting the *Homed* output status flag.

TENA – Move Teach Enable
While the *TENA* input is active, move teach functionality will is enabled. Move positions may be set either through *IGROUP 4* individual move teach inputs, or by selecting the move number on the binary select inputs of *IGROUP 5* and then setting the *TSEL* flag (below). Teach functionality may also be enabled automatically at drive power-up through the *TE* flag of the *Configuration.Options* register, eliminating the requirement for a dedicated input.

TSEL – Teach Selected Move Position

The rising edge of the *TSEL* input with teach functionality enabled (*TENA* active) will cause the current absolute position of the drive to be set in the position parameter of the move number selected on the binary select inputs of *IGROUP 5*.

BKOV – Brake Override
The BKOV input forces the actuator brake relay to be activated, releasing the brake. This input is honored whether or not the drive is enabled and overrides the normal brake behavior.

RESET – Fault Reset
The rising edge of the *RESET* input will clear any faults active. If all pending 'hard' faults are successfully cleared, the drive will become enabled if the Maintained Enable input is active or a rising edge of the Momentary Enable is observed. *RESET* allows faults to be cleared independently of the drive enable inputs (*Enable Momentary* or *Enable Maintained*).

IEG_MOTION

	STR	**STU**	**MVE**	**MVL**	**EMOVE**		**HOME**		**JOGF**	**JOGN**	**JOGP**	**PAUSE**	**STOP**		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

STOP – Stop Motion
The *STOP* input event forces the drive to decelerate using STOP deceleration and hold position. The drive will be forced into a position mode of operation if necessary.

PAUSE – Pause Move
If any move is executing, the PAUSE input event will force the drive to decelerate to zero command velocity using the current motion's deceleration and hold position until PAUSE is removed. If no move is active, the PAUSE input event has no effect.

JOGP – Jog Positive
JOGN – Jog Negative
The *JOGP* and *JOGN* input events will begin jog motion. If both input events are active, the drive will command zero velocity and hold position.

JOGF – Jog Fast
While the *JOGF* input event is active, the target velocity for an active jog will be set the value in the *Jog.FastVelocity* register. While *JOGF* is inactive, the value in the *Jog.SlowVelocity* register is used.

HOME– Initiate Home Move
The rising edge of the HOME input event initiates the Home Move.

EMOVE – Initiate Dedicated Move (Emergency Move)
The EMOVE input event provides maintained move initiate functionality for the Dedicated Move in the same manner as the individual move maintained input events.

MVL – Initiate Maintained (level sensitive) Binary Select Move
The *MVL* input executes the move number specified on the binary select inputs while the *MVL* event remains active in the same manner that the individual Move Maintained initiates operate.

MVE – Initiate Momentary (edge sensitive) Binary Select Move
Te move number specified on the binary select inputs will be executed on the rising edge of the *MVL* input event in the same manner that the individual *Move Momentary* initiates operate. The move number does not need to be maintained on the binary select inputs after the move has begun execution.

STU – Startup
Resets *Startup Complete* and initiates *Move 15*. The *Startup* function is the only motion command accepted while *Startup Complete* is inactive and *Startup Required* has been selected as a configuration option. A move in the move sequence started through *Move 15* will normally have the *Set Startup Complete* option flag selected which will cause *Startup Complete* to be set when the move finishes.

STR – Reset Startup Complete
The rising edge of *STR* will resets the *Startup Complete* flag. If *Startup Required* has been selected as a configuration option, no motion other than *Startup* motion will be allowed until *Startup Complete* is again set (normally through the execution of *Startup*).

IEG_MOVE_LEVEL

L15	L14	L13	L12	L11	L10	L9	L8	L7	L6	L5	L4	L3	L2	L1	L0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The *IEG_MOVE_LEVEL* input events will execute the move specified by the highest active event bit while the bit remains active. Any next move that has been set for the move is ignored when moves are executed with the maintained (level) input events. A move initiated through a maintained input event has priority over (and will interrupt and override) an executing move that was initiated with a momentary input event.

IEG_MOVE_EDGE

E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The *IEG_MOVE_EDGE* input events will execute the move specified on the rising edge of the input event. The event does not need to remain active. A move must be initiated with a momentary input event in order for it to execute another move automatically upon completion. A move initiated through a maintained input event has priority over (and will interrupt and override) a move initiated with a momentary input event.

IEG_MOVE_TEACH

M15	M14	M13	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	M0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The rising edge of an *IEG_MOVE_TEACH* input event will store the drive's current absolute position in the move's position register. The teach mode must be enabled and an absolute position reference frame must be active before move positions may be taught.

IEG_MOVE_SELECT

								B7	B6	B5	B4	B3	B2	B1	B0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The B0 through B7 select a move number that for the binary select move initiate and move teach input events. Each flag provides a 2^n binary value towards the final move number.

IEG_MOVE_SWITCH

E8	E7	E6	E5	E4	E3	E2	E1	L8	L7	L6	L5	L4	L3	L2	L1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

The switch group specifies input events that may be selected to terminate move segments. The L1 through L7 input events provide maintained (level) input event functionality. The E1 through E8 input events provide momentary (edge) input event functionality. Switch inputs may be selected as move termination events.

Output Event Groups

This section defines the output status event flags in each of the output event status groups.

Output Event Group Types

Type	Base Type	Words	Description
OEG_STATUS	FLAGS	1	General drive status
OEG_MOTION	FLAGS	1	General motion status
OEG_CONTROL	FLAGS	1	Internal control status
OEG_MOVE_ACTIVE	FLAGS	1	Active move status
IEG_MOVE_IN_POSITION	FLAGS	1	Move in-position status

OEG_STATUS

			STC					ALTM	DEFM	FW	WARN	FLT	RDY	HOMED	ENA
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

ENA – Enabled
The *ENA* flag will be active when no flags are set in the Disables status word and the drive becomes enabled.

HOMED – Homed
The *HOMED* flag is set when a move completes that establishes an absolute reference frame or a rising edge of the *Define Home 1* or *Define Home 2* input event is observed.

READY – Ready
The *READY* flag is set when both the *ENA* and *HOMED* flags are active indicating that the drive is homed and enabled.

FLT – Faulted
The *FLT* flag is latched when hard fault is active. The flag is cleared on the rising edge of *ENA* or the rising edge of a *Reset Faults* input event.

WARN – Warning
The WARN flag is set while a soft fault is active. The flag is cleared automatically when the *FAULT* contributing to the warning is removed.

FW – Fault or Warning
The *FW* is the logical ‘or’ of the *FLT* and *WARN* flags.

DEFM – Default Mode of Operation
The *DEFM* flag will be active while the default mode of operation is active.

ALTM – Alternate Mode of Operation

The *ALTM* flag will be active while the alternate mode of operation is active.

STC – Startup Complete
The *STC* flag will be active when Startup has completed. It follows the state of the *Startup Complete* flag in the *Status Log*.

OEG_MOTION

EP	H2P	H1P	IP		SMV	MV	EMV		ST	HMV	JM	JP	J	PAUSED	STOPPED
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

STOPPED – Stopped
The *STOPPED* flag is active while the *STOP* input event is active.

PAUSED – Paused
The *PAUSED* flag is active while any move is active and the *PAUSE* input event is active. While active the drive will decelerate to a stop using the current motion's acceleration parameter and hold position until the *PAUSE* input event becomes inactive.

J – Jogging
JP – Jogging (+)
JM – Jogging (-)
The *J* flag will be active while the drive is executing jog motion. If the direction of motion is positive, the *JP* flag will be active, if negative the *JM* flag will be active.

While both the Jog Positive and Jog Negative input events are active, the J, JP, and JM output status events will all be active and the drive will decelerate to zero velocity and hold position.

HMV – Homing Active
The *HMV* flag is active while the Home move is executing.

ST – Startup Active
The *ST* flag is active while the Startup function is executing.

EMV – Dedicated (emergency) Move Active
The *EHMV* flag is active while the Dedicated Move is executing.

MV – Move Active
The *MV* flag is active while any move (including the Home and Dedicated Move) is executing.

SMV – Secondary Move Active
The S*MV* flag is active while the secondary segment of any move (including the Home and Dedicated Move) is executing.

IP – In Position
The *IP* flag is active when the drive is enabled, a position control mode is active, the command velocity is zero, and the absolute value of position error has remained less than the maximum allowable position error value for at least the time specified by the position error window time parameter.

H1P – At Home.Position1
The *H1P* flag is active when the *IP* is active and the commanded position is *Home Position 1*.

H2P – At Home.Position2
The *H2P* flag is active when the *IP* is active and the commanded position is *Home Position 2*.

EP – At Dedicated Move Position
The *H1P* flag is active when the *IP* is active and the commanded position is the position specified for the final segment of the *Dedicated Move*.

OEG_CONTROL

		PWM	BKDLY	AUP	ILS	PLM	PLP	PL	ILM	ILP	IL	VL	IRATED	BRKR	SHNT
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SHNT – Shunt Active
The *SHNT* flag is active while the DC Bus shunt relay is active.

BRKR – Brake Release Active
The *BRKR* flag is active while the Brake Release relay is active.

IRATED – Over Rated (continuous) Current
The *IRATED* is active whenever the magnitude of the current feedback vector is greater than the drive's continuous current rating.

VL – Voltage Limited
The *V*L flag is active when the drive command requires more current than can be delivered from the DC Bus and the controller is being voltage limited.

IL – Current Limited
ILP – Current Limited (+)
ILM – Current Limited (-)
The *IL* flag is set whenever the position controller requires more current than can be delivered due the current command being limited. If the drive requires more positive current than can be delivered, the *ILP* flag will be set. If the drive requires more negative current than can be delivered, the *ILM* flag will be set.

PL – Position Limit Active
PLP – Position Limited (+)
PLP – Position Limited (-)
The *PL* flag is set whenever the position command has moved past an enabled position limit. If the drive has passed the enabled positive limit, the *PLP* flag will be set. If the drive has passed the enabled negative limit, the *PLM* flag will be set.

ILS – Seating Current Limit Active
The *ILS* flag is set on the rising edge of *PL* (position limit active) and remains active for the time specified for current limit seating.

AUP – Analog Unipolar Input 1 (0 – 10V)
The *AUP* flag is set when *Analog Input 1* has been selected for unipolar operation.

BKDLY – Brake Delay Active
The BKDLY flag will be active while the drive is controlling actuator phase voltages (*PWM* is active) but the drive is not enabled. This occurs with a non-zero brake delay.

PWM – PWM Active
The PWM flag is set when the *PWM* circuitry becomes active controlling the actuator phase voltages. The *PWM* may be active before the drive indicates that it is enabled if there is non-zero brake delay.

OEG_MOVE_ACTIVE

M15	M14	M13	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	M0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

A flag in the *OEG_MOVE_ACTIVE* output status event group is active while the corresponding Move is executing.

OEG_MOVE_IN_POS

M15	**M14**	**M13**	**M12**	**M11**	**M10**	**M9**	**M8**	**M7**	**M6**	**M5**	**M4**	**M3**	**M2**	**M1**	**M0**
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

A flag in the *OEG_MOVE_IN_POS* output status event group is active when the *IP* flag is active in the *OEG_MOTION* register and the command position matches the final segment position of the corresponding move. Note that multiple flags may be active.

MOVE

The *MOVE* type defines the standard motion registers used for the Move, Home, and Dedicated Move motion profiles.

MOVE Structure Table

Offset	Name	Type	Size	Description
0	Options	FLAGS	1	Move option control flags
1	CurrentLimit	UCUR16	1	Move current limit
2	Acceleration	UACC32	2	Move acceleration
4	TerminationSwitch	IEG_MOVE_SWITCH	1	Move termination switches
5	NextMove	UINT16	1	Next move
6	Primary.Options	FLAGS	1	Primary motion control flags
7		UINT16	1	reserved
8	Primary.Position	POS32	2	Primary motion position/distance
10	Primary.Velocity	UVEL32	2	Primary motion velocity
12	Secondary.Options	FLAGS	1	Secondary motion control flags
13		UINT16	1	reserved
14	Secondary.Position	POS32	2	Secondary motion position/distance
16	Secondary.Velocity	UVEL32	2	Secondary motion velocity

Deceleration

Deceleration can be set for a move and will be used if the motion returns to zero velocity. The Modbus address for Decel 0 is 6500. The Modbus address for all moves is 6500 + Move# *2. It is of the same type as Acceleration. If the deceleration value is set to zero, the acceleration value will be used in its place.

Options

General options for the move are set through the bitmapped fields of the *Options* parameter as specified in the following table.

STC	**APMAX**	**APMIN**								**REF2**	**REF1**		**WAIT**	**NEXT**	**SM**
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

SM – Secondary Motion Enable

If secondary motion is not enabled, the move will execute only the primary motion segment.

NEXT – Auto-Execute Next MOVE

The *MOVE* specified in *NextMove* is will be executed when the move completes if the *NEXT* flag is set in the *Options* register.

WAIT – Wait for In-Position Active

This flag may be set to require the IN_POSITION status flag to be active before the MOVE is considered complete. The MOVE remains active until the in-position conditions are met.

REF1 – Set Absolute Reference Position 1
REF2 – Set Absolute Reference Position 2

If the *REF1* (*REF2*) flag is set, the absolute position of the drive will be set to *HOME.Position1* (*HOME.Position2*) upon completion of the *MOVE*, establishing a new absolute position reference frame. (If both *REF1* and *REF2* are set, *REF1* has priority and the *REF2* flag is ignored.) The establishment of the reference frame will be the final action taken in completion of the *MOVE* and will not occur until the *In-Position* criteria has been met if selected by the *WAIT* flag. Once an absolute reference frame has been established, the drive is considered to be *Homed* (*see also DefineHome*)

APMIN – Set Minimum Analog Position

Sets the *Analog Position Control* mode's minimum position to the current position.

APMAX – Set Maximum Analog Position

Sets the *Analog Position Control* mode's maximum position to the current position.

STC – Set Startup Complete
Sets the *Startup Complete* flag in the *Status Log* and saves both the *Status Log* and *User Parameters* to non-volatile memory.

CurrentLimit
CurrentLimit is used by both the primary and secondary motion segments if enabled by the *Primary.Options* or *Secondary.Options ILIMIT* flag. The absolute value of the maximum current command allowed for a motion segment with the *ILIMIT* option enabled will be limited to the value specified by the MOVE's *CurrentLimit*.

Acceleration
Acceleration sets the maximum rate of change of velocity command allowed for the entire move motion.

TerminationSwitch
An active input event in the *IEG_MOVE_SWITCH* group will terminate the move segment if the TSF option is selected in the move segment's options.

NextMove
The NextMove register specifies the zero-based move number of a move that is to

Primary.Options
Secondary.Options
Options for both the primary and secondary motion segments are set their respective options bitmap register.

	FP	**FS**	**FI**		**TSF**	**TSR**	**TI**			**VO**	**ILIMIT**		**TYPE**		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

TYPE
The *TYPE* field encodes the type of motion profile executed by the motion segment.

0 Absolute Position
The motion segment will move to the specified position in the absolute reference frame.

1 Incremental Distance
The motion segment will move the specified distance from the segment's starting position.

2 Unlimited Positive Direction
Motion will execute in the positive direction until a termination condition is observed.

3 Unlimited Negative Direction
Motion will execute in the negative direction until a termination condition is observed.

4 Absolute Positive
The motion segment will move to the specified position in the absolute reference frame if the direction of motion is positive, otherwise the drive will stop.

5 Absolute Negative
The motion segment will move to the specified position in the absolute reference frame if the direction of motion is negative, otherwise the drive will stop.

6 (reserved)
7 (reserved)
These motion profile types are reserved.

ILIMIT
Limits the maximum current command during the motion segment to the MOVE's CurrentLimit parameter. When the move segment completes, the normal drive current limit is restored.

VO
If selected, the index velocity will be controlled via the analog velocity override option.

TI
Terminates the motion segment if the absolute value of feedback current reaches the MOVE's CurrentLimit. If a TSR or TSF option (below) has also been enabled to terminate the motion segment on a digital switch input, both current limit AND switch termination options must be fulfilled before the motion segment terminates.

TSR
Terminates the motion segment if any digital input switch specified in the MOVE's TerminationSwitch bitmap becomes active. A momentary (edge-sensitive) input switch becomes active when a rising edge of the input is observed AFTER the move segment is started. A maintained (level-sensitive) input switch fulfills the termination condition whenever it is observed to be ON during the move segment. If the ILIMIT option (above) has also been enabled to terminate the motion segment on a current limit condition, both current limit AND switch termination options must be fulfilled before the motion segment terminates.

TSF
Terminates the motion segment if any digital input switch specified in the MOVE's TerminationSwitch bitmap becomes inactive. A momentary (edge-sensitive) input switch becomes inactive when a falling edge of the input is observed *after* the move segment is started. A maintained (level-sensitive) input switch fulfills the termination condition whenever it is observed to be OFF during the move segment. If the *ILIMIT* option (above) has also been enabled to terminate the motion segment on a current limit condition, both current limit AND switch termination options must be fulfilled before the motion segment terminates.

FI
Generates a *MoveTermination* fault if the TI option flag is set and is the reason for the move segment's termination.

FS
Generates a *MoveTermination* fault if either the TSR or TSF option flag is set and is the reason for the move segment's termination. This option may be

FP
Generates a MoveTermination fault if the move segment has terminated due to reaching the target position (absolute moves) or completing the target distance (incremental moves). This option is normally used in conjunction with either the TI, TSR, or TSF options when either current or switch termination is expected to occur before achieving the move segment position (absolute moves) or completing the move segment distance (incremental moves).

Primary.Position
Secondary.Position
Specifies the position or distance for move segment. This register is not required for motion segments specified as *Unlimited Positive Direction* or *Unlimited Negative Direction* motion types in the segment's option word.

Primary.Velocity
Secondary.Velocity
Specifies the absolute value of the maximum velocity that will be commanded for the motion segment.

Appendix B – Modbus Register IDs

This appendix provides the full *MODBUS* register map ordered by register *MODBUS* identifier. The register map is divided for convenience into the following major sections:

- Status and Control Registers
- User Parameter Registers
- Mapped Value Registers
- Factory Parameter Registers
- Factory Identification Registers

Each table gives the register's *MODBUS* identifier (in both decimal and hexadecimal), name, data type, read-write accessibility, data size (number of 16-bit words), storage location, and brief description. The abbreviations used in the read-write accessibility (*Access*) and storage location (*Storage*) fields are defined in the tables below.

Access Definitions	
Abbreviation	Description
RO	Data is read-only (input register)
RW	Data is read-write (holding register)

Storage Definitions	
Abbreviation	Description
RAM	Data is stored in RAM
ROM	Data is stored in flash read-only memory
UP	User parameter data in RAM initialized from and storable to non-volatile storage mirror
FP	Factory parameter data in RAM initialized from and storable to non-volatile storage mirror
SL	Status log data in RAM initialized from and auto-stored to non-volatile storage mirror
FL	Fault log data in RAM initialized from and auto-stored to non-volatile storage mirror
NONE	Data has no storage behind it (used for indirect mapped data)

Status and Control Registers Table

See Appendix A for details on data Types

ID	ID (hex)	Name	Type	Access	Size	Storage	Description
2	0x0002	System Status	UINT16	RO	30	RAM	System status registers
2	0x0002		UINT16	RO	2	RAM	reserved
4	0x0004	Disables	FLAGS	RO	1	RAM	Disabling flags (internal)
5	0x0005	Faults	FAULT	RO	1	RAM	Fault status flags (all active)
6	0x0006	HardFaults	FAULT	RO	1	RAM	Active hard fault status flags
7	0x0007	SoftFaults	FAULT	RO	1	RAM	Active soft fault status flags (warnings)
8	0x0008	HallBattery	UVOLT16	RO	1	RAM	ABS Hall Board battery voltage [11.5 V]
9	0x0009		UINT16	RO	1	RAM	reserved
10	0x000A	BoardTemp	INT32	RO	2	RAM	Filtered PCB temperature [11.21 DEG C]
12	0x000C		UINT16	RO	2	RAM	reserved
14	0x000E	ActuatorTemp	INT32	RO	2	RAM	Filtered actuator temperature [11.21 DEG C]
16	0x0010	MinScanTime	UINT32	RO	2	RAM	Minimum background scan time [SYSCLK]
18	0x0012	MaxScanTime	UINT32	RO	2	RAM	Maximum background scan time [SYSCLK]
20	0x0014	AvgScanTime	UINT32	RO	2	RAM	Average background scan time [SYSCLK]
22	0x0016	Hs_temp	INT32	RO	2	RAM	Filtered hea tsink temperature
24	0x0018	Faults32	UINT32	RO	2	RAM	Fault status flags (all active) - 32 faults
26	0x001A	Hardfaults32	UINT32	RO	2	RAM	Active hard fault status flags – 32 faults
28	0x001B	Softfaults32	UINT32	RO	2	RAM	Active soft fault status flags – 32 faults
30	0x001E	Command Status	UINT16	RO	64	RAM	Command status registers
30	0x001E	Command.SubMode	FLAGS	RO	1	RAM	Active command sub-mode
31	0x001F	Command.Mode	OPMODE	RO	1	RAM	Active command mode
32	0x0020	Command.Flags	FLAGS	RO	1	RAM	Internal command mode flags
33	0x0021	Command.Move	UINT16	RO	1	RAM	Active move number
34	0x0022	Command.PlimitMinus	POS32	RO	2	RAM	Active position limit (-)
36	0x0024	Command.PlimitPlus	POS32	RO	2	RAM	Active position limit (+)
38	0x0026	Command.PlimitVelocity	INT32	RO	2	RAM	Active position limit velocity
40	0x0028	Command.IlimitMinus	CUR32	RO	2	RAM	Active current command limit (-)
42	0x002A	Command.IlimitPlus	CUR32	RO	2	RAM	Active current command limit (+)
44	0x002C	Command.Vlimit	INT32	RO	2	RAM	Active velocity limit
46	0x002E	Command.Alimit	UACC32	RO	2	RAM	Active acceleration limit
48	0x0030	Command.MoveFlags	FLAGS	RO	1	RAM	Active position limit (+)
49	0x0031		UINT16	RO	1	RAM	reserved
50	0x0032	Command.Ptarget	POS32	RO	2	RAM	Active position target
52	0x0034	Command.Vtarget	INT32	RO	2	RAM	Active velocity target
54	0x0036	Command.Id	CUR32	RO	2	RAM	Direct current command
56	0x0038	Command.Iq	CUR32	RO	2	RAM	Quadrature current command
58	0x003A	Command.Vd	VOLT32	RO	2	RAM	Direct voltage command
60	0x003C	Command.Vq	VOLT32	RO	2	RAM	Quadrature voltage command
62	0x003E	D_liimit	UACC32	RO	2	RAM	Active deceleration limit
100	0x0064	I/O Status	UINT16	RO	42	RAM	Digital I/O status registers
100	0x0064	Inputs	FLAGS	RO	1	RAM	Active input status
101	0x0065	Outputs	FLAGS	RO	1	RAM	Active output status
102	0x0066	HwInputs	FLAGS	RO	1	RAM	Hardware inputs (before inhibits/polarity)
103	0x0067	HwOutputs	FLAGS	RO	1	RAM	Hardware outputs (after inhibits/polarity)
104	0x0068	OutputEvents.0	OEG_STATUS	RO	1	RAM	Active output status events
105	0x0069	OutputEvents.1	OEG_MOTION	RO	1	RAM	
106	0x006A	OutputEvents.2	OEG_CONTROL	RO	1	RAM	
107	0x006B	OutputEvents.3	OEG_MOVE_ACTIVE	RO	1	RAM	
108	0x006C	OutputEvents.4	OEG_MOVE_IN_POS	RO	1	RAM	
109	0x006D		UINT16	RO	1	RAM	reserved
110	0x006E	InputEvents.0	IEG_MODE	RO	1	RAM	Active input events
111	0x006F	InputEvents.1	IEG_MOTION	RO	1	RAM	
112	0x0070	InputEvents.2	IEG_MOVE_LEVEL	RO	1	RAM	
113	0x0071	InputEvents.3	IEG_MOVE_EDGE	RO	1	RAM	
114	0x0072	InputEvents.4	IEG_MOVE_TEACH	RO	1	RAM	
115	0x0073	InputEvents.5	IEG_MOVE_SELECT	RO	1	RAM	
116	0x0074	InputEvents.6	IEG_MOVE_SWITCH	RO	1	RAM	
117	0x0075		UINT16	RO	1	RAM	reserved
118	0X0076	LatchedInputEvents	UINT16	RO	8	RAM	Latched input event status
126	0x007E	RisingInputEvents	UINT16	RO	8	RAM	Rising edge input event status
134	0x0086	FallingInputEvents	UINT16	RO	8	RAM	Falling edge input event status
200	0x00C8	Analog Command Status	UINT16	RO	6	RAM	Analog command mode status registers

ID	ID (hex)	Name	Type	Access	Size	Storage	Description
200	0x00C8	AnalogPositionTarget	POS32	RO	2	RAM	Target position for analog position mode
202	0x00CA	AnalogVelocityTarget	VEL32	RO	2	RAM	Target velocity for analog velocity mode
204	0x00CC	AnalogCurrentTarget	CUR32	RO	2	RAM	Target current for analog current mode
220	0x00DC	Analog Input 1	UINT16	RO	20	RAM	Analog Input 1 status registers
220	0x00DC		UINT16	RO	1	RAM	reserved
221	0x00DD	AdcInstantaneous	UINT16	RO	1	RAM	Unfiltered ADC [0x0000..0xFFF0)
222	0x00DE	AdcFiltered	INT32	RO	2	RAM	Filtered ADC (0..1) [2.30 ADC]
224	0x00E0	UserFiltered	INT32	RO	2	RAM	Filtered input [6.26 user units]
226	0x00E2	UseableRange	INT32	RO	2	RAM	Fraction of useable range ([0..1) [2.30]
228	0x00E4	Dudx	INT32	RO	2	RAM	du/dx scale factor [10.22]
230	0x00E6	B	INT32	RO	2	RAM	y-axis intercept offset [6.26 user units]
232	0x00E8	RangeMinimum	INT32	RO	2	RAM	Minimum useable range (0..1) [2.30 ADC]
234	0x00EA	RangeMaximum	INT32	RO	2	RAM	Maximum useable range (0..1) [2.30 ADC]
236	0x00EC		UINT16	RO	4	RAM	reserved
240	0x00F0	Analog Input 2	UINT16	RO	20	RAM	Analog Input 2 status registers
240	0x00F0		UINT16	RO	1	RAM	reserved
241	0x00F1	AdcInstantaneous	UINT16	RO	1	RAM	Unfiltered ADC [0x0000..0xFFF0)
242	0x00F2	AdcFiltered	INT32	RO	2	RAM	Filtered ADC (0..1) [2.30 ADC]
244	0x00F4	UserFiltered	INT32	RO	2	RAM	Filtered input [6.26 user units]
246	0x00F6	UseableRange	INT32	RO	2	RAM	Fraction of useable range ([0..1) [2.30]
248	0x00F8	Dudx	INT32	RO	2	RAM	du/dx scale factor [10.22]
250	0x00FA	B	INT32	RO	2	RAM	y-axis intercept offset [6.26 user units]
252	0x00FC	RangeMinimum	INT32	RO	2	RAM	Minimum useable range (0..1) [2.30 ADC]
254	0x00FE	RangeMaximum	INT32	RO	2	RAM	Maximum useable range (0..1) [2.30 ADC]
256	0x0100		UINT16	RO	4	RAM	reserved
260	0x0104	Analog Input 3 (reserved)	UINT16	RO	20	RAM	Analog Input 3 status registers
280	0x0118	Analog Input 4 (reserved)	UINT16	RO	20	RAM	Analog Input 4 status registers
340	0x0154	Velocity Status	UINT16	RO	48	RAM	Velocity/Position loop status registers
340	0x0154	Vcontrol	CUR32	RO	2	RAM	Controller current command out [9.23 AMPS]
342	0x0156	RevAngleLast	INT32	RO	2	RAM	Last position rev angle [0.32 REV]
344	0x0158	VfeedbackUser	VEL32	RO	2	RAM	Feedback velocity [8.24 RPS]
346	0x015A	VcommandUser	VEL32	RO	2	RAM	Command velocity [8.24 RPS]
348	0x015C	VerrorUser	VEL32	RO	2	RAM	Velocity error [8.24 RPS]
350	0x015E	VerrorMinUser	VEL32	RO	2	RAM	Minimum velocity error observed [8.24 RPS]
352	0x0160	VerrorMaxUser	VEL32	RO	2	RAM	Maximum velocity error observed [8.24 RPS]
354	0x0162		UINT16	RO	2	RAM	reserved
356	0x0164	Vdisplay	VEL32	RO	2	RAM	Filtered display velocity [8.24 RPS]
358	0x0166		UINT16	RO	2	RAM	reserved
360	0x0168	Vfeedback	VEL32	RO	2	RAM	Internal velocity f/b [0.32 REV/UPDATE]
362	0x016A	Vcommand	VEL32	RO	2	RAM	Internal velocity cmd [0.32 REV/UPDATE]
364	0x016C		UINT16	RO	14	RAM	reserved
378	0x017A	Pfeedback	POS32	RO	2	RAM	Feedback position [16.16 REVS]
380	0x017C	Pcommand	POS32	RO	2	RAM	Command position [16.16 REVS]
382	0x017E	Perror	POS32	RO	2	RAM	Position error [16.16 REVS]
384	0x0180	PerrorMin	POS32	RO	2	RAM	Min position error observed [16.16 REVS]
386	0x0182	PerrorMax	POS32	RO	2	RAM	Max position error observed [16.16 REVS]
388	0x0184	Execution_time_min	UINT32	RO	1	RAM	
389	0x0185	Execution_time_max	UINT32	RO	1	RAM	
400	0x0190	Scope Status	UINT16	RO	16	RAM	Digital oscilloscope status registers
400	0x0190	Status	FLAGS	RO	1	RAM	Scope status flags
401	0x0191	RecordCound	UINT16	RO	1	RAM	Number of records in buffer
402	0x0192	MaxRecords	UINT16	RO	1	RAM	Number of records buffer can hold
403	0x0193	RecordSize	UINT16	RO	1	RAM	Size of single record [WORDS]
404	0x0194	BufferSize	UINT16	RO	1	RAM	Total buffer size [WORDS]
405	0x0195	Timestamp	UINT16	RO	1	RAM	Free-running timestamp [UPDATES]
406	0x0196	Channel1	INT32	RO	2	RAM	Last Channel 1 value
408	0x0198	Channel2	INT32	RO	2	RAM	Last Channel 2 value
410	0x019A	Channel3	INT32	RO	2	RAM	Last Channel 3 value
412	0x019C	Channel4	INT32	RO	2	RAM	Last Channel 4 value
414	0x019E	Trigger	INT32	RO	2	RAM	Current trigger level
500	0x01F4	Current Loop Status	UINT16	RO	87	RAM	Current loop status registers
500	0x01F4	RawEangle	INT32	RO	2	RAM	Raw electrical angle from position rev angle
502	0x01F6	Eangle	INT32	RO	2	RAM	Electrical angle (commutation)
504	0x01F8	Esine	INT32	RO	2	RAM	SIN(Eangle)
506	0x01FA	Ecosine	INT32	RO	2	RAM	COS(Eangle)
508	0x01FC	IR	CUR32	RO	2	RAM	R phase current [9.23 AMPS]
510	0x01FE	IS	CUR32	RO	2	RAM	S phase current [9.23 AMPS]
512	0x0200	IT	CUR32	RO	2	RAM	T phase current [9.23 AMPS] (calculated)
514	0x0202	Id	CUR32	RO	2	RAM	D leg current (static frame) [9.23 AMPS]
516	0x0204	Iq	CUR32	RO	2	RAM	Q leg current (static frame) [9.23 AMPS]

ID	ID (hex)	Name	Type	Access	Size	Storage	Description
518	0x0206	IdFeedback	CUR32	RO	2	RAM	D leg current (rotating frame) [9.23 AMPS]
520	0x0208	IqFeedback	CUR32	RO	2	RAM	Q leg current (rotating frame) [9.23 AMPS]
522	0x020A	Idcommand	CUR32	RO	2	RAM	D leg current cmd [9.23 AMPS]
524	0x020C	IqCommand	CUR32	RO	2	RAM	Q leg current cmd [9.23 AMPS]
526	0x020E	Imagnitude	UCUR32	RO	2	RAM	Magnitude of current vector [9.23 AMPS]
528	0x0210	VdCommand	VOLT32	RO	2	RAM	D leg voltage cmd (rotating frame) [11.21 V]
530	0x0212	VqCommand	VOLT32	RO	2	RAM	Q leg voltage cmd (rotating frame) [11.21 V]
532	0x0214	Vd	VOLT32	RO	2	RAM	D leg voltage cmd (static frame) [11.21 V]
534	0x0216	Vq	VOLT32	RO	2	RAM	Q leg voltage cmd (static frame) [11.21 V]
536	0x0218	Vr	INT32	RO	2	RAM	R phase voltage cmd [1.31 full scale]
538	0x021A	Vs	INT32	RO	2	RAM	S phase voltage cmd [1.31 full scale]
540	0x021C	Vt	INT32	RO	2	RAM	T phase voltage cmd [1.31 full scale]
542	0x021E	Qharmonic	CUR32	RO	2	RAM	Q harmonic feed-forward [9.23 AMPS]
544	0x0220	IdError	INT32	RO	2	RAM	D leg controller error
546	0x0222	IqError	INT32	RO	2	RAM	Q leg controller error
548	0x0224	Tr	INT32	RO	2	RAM	R phase duty cycle [0.0x7FFFFFFF]
550	0x0226	Ts	INT32	RO	2	RAM	S phase duty cycle [0.0x7FFFFFFF]
552	0x0228	Tt	INT32	RO	2	RAM	T phase duty cycle [0.0x7FFFFFFF]
554	0x022A	PwmBasePeriod	UINT16	RO	1	RAM	Base period divisor [TBCLK counts]
555	0x022B	PwmPeriod	UINT16	RO	1	RAM	Modulated period divisor [TBCLK counts]
556	0x022C		UINT16	RO	4	RAM	reserved
560	0x0230	Vmagnitude	VOLT32	RO	2	RAM	Magnitude of voltage command vector
562	0x0232		UINT16	RO	2	RAM	reserved
564	0x0234	Icontinuous	UCUR32	RO	2	RAM	Filtered continuous current
566	0x0236	Idisplay	CUR32	RO	2	RAM	Filtered IqFeedback for display
568	0x0238	VbusInstantaneous	VOLT32	RO	2	RAM	Instantaneous bus voltage
570	0x023A	Vbus	VOLT32	RO	2	RAM	Filtered bus voltage
572	0x023C	PowerInstantaneous	INT32	RO	2	RAM	Instantaneous power
574	0x023E	Power	INT32	RO	2	RAM	Filtered power
576	0x0240	MaxImagnitude	UCUR32	RO	2	RAM	Maximum Imagnitude observed
578	0x0242	MaxVbus	VOLT32	RO	2	RAM	Maximum Vbus observed
580	0x0244	MinExecutionTime	UINT16	RO	1	RAM	Minimum execution time [SYSCLK]
581	0x0245	MaxExecutionTime	UINT16	RO	1	RAM	Maximum execution time [SYSCLK]
582	0x0246	Shunt_energy_available	INT32	RO	2	RAM	Shunt energy available
584	0x0248	Shunt_on	INT16	RO	1	RAM	Shunt on point
585	0x0249	Shunt_off	INT16	RO	1	RAM	Shunt off point
586	0x250	High_voltage_trip	INT16	RO	1	RAM	Runtime high voltage trip point
600	0x0258	Position Status	UINT16	RO	36	RAM	Position status registers
600	0x0258	RevAngle	INT32	RO	2	RAM	Actuator rev angle
602	0x025A	HallGain	INT32	RO	2	RAM	Tracking filter accel gain
604	0x025C	HallDamping	INT32	RO	2	RAM	Tracking filter damping gain
606	0x025E	HallError	INT32	RO	2	RAM	Tracking filter error
608	0x0260	HallVelocity	INT32	RO	2	RAM	Tracking filter velocity
610	0x0262	HallRevAngle	INT32	RO	2	RAM	Tracking rev angle
612	0x0264	SineIn	INT32	RO	2	RAM	SIN(input position angle)
614	0x0266	CosineIn	INT32	RO	2	RAM	COS(input position angle)
616	0x0268	SineOut	INT32	RO	2	RAM	SIN(estimated position angle)
618	0x026A	CosineOut	INT32	RO	2	RAM	COS(estimated position angle)
620	0x026C	SineMin	INT16	RO	1	RAM	Minimum SineIn observed
621	0x026D	SineMax	INT16	RO	1	RAM	Maximum SineIn observed
622	0x026E	CosineMin	INT16	RO	1	RAM	Minimum CosineIn observed
623	0x026F	CosineMax	INT16	RO	1	RAM	Maximum CosineIn observed
624	0x0270	EncoderCount	INT32	RO	2	RAM	Free-running count from encoder inputs
626	0x0272	UVW	UINT16	RO	1	RAM	UVW commutation position (bit 0 = U)
627	0x0273	PosTrackingError	INT16	RO	1	RAM	Position tracking error
628	0x0274	PosTableCorrection	INT32	RO	2	RAM	Position feedback correction from table
630	0x0276	Eoffset	INT32	RO	2	RAM	Electrical angle offset
632	0x0278	AbsHallPosition	INT16	RO	1	RAM	Absolute Hall board position
633	0x0279	AbsHallStatus	FLAGS	RO	1	RAM	Absolute Hall board status flags
634	0x027A	AbsHallRevision	UINT16	RO	1	RAM	Absolute Hall board revision
635	0x027B	Status	FLAGS	RO	1	RAM	Position status flags
700	0x02BC	Analog Output 1 Status	UINT16	RO	4	RAM	Analog Output 1 status registers
700	0x02BC	AnalogOutput.1.Vfiltered	INT32	RO	2	RAM	Filtered output variable
702	0x02BE	AnalogOutput.1.DacOutput	UINT16	RO	1	RAM	DAC output variable
703	0x02BF	AnalogOutput.1.Fraction	UINT16	RO	1	RAM	Fraction of variable
704	0x02C0	Analog Output 2 Status	UINT16	RO	4	RAM	Analog Output 2 status registers
704	0x02C0	AnalogOutput.2Vfiltered	INT32	RO	2	RAM	Filtered output variable
706	0x02C2	AnalogOutput.2.DacOutput	UINT16	RO	1	RAM	DAC output variable
707	0x02C3	AnalogOutput.2.Fraction	UINT16	RO	1	RAM	Fraction of variable
732	0x02DC	Status Log	UINT16	RO	32	SL	Status Log

ID	ID (hex)	Name	Type	Access	Size	Storage	Description
732	0x02DC	CRC	UINT16	RO	1	SL	Internal CRC check word
733	0x02DD		UINT16	RO	1	SL	reserved
734	0x02DE	Flags	FLAGS	RO	1	SL	Status log flags
735	0x02DF	PowerUpCount	UINT16	RO	1	SL	Number of times drive has powered up
736	0x02E0	PwmRunTime	UINT32	RO	2	SL	Total time PWM has been active
738	0x02E2	AbsPositionOffset	POS32	RO	2	SL	Absolute position offset
740	0x02E4	MaxCurrent	UCUR32	RO	2	SL	Max current magnitude observed
742	0x02E6	MaxVbus	UVOLT32	RO	2	SL	Max bus voltage observed
744	0x02E8	MaxBoardTemp	INT32	RO	2	SL	Max PCB temperature observed
746	0x02EA	MaxActuatorTemp	INT32	RO	2	SL	Max actuator temperature observed
748	0x02EC		UINT16	RO	16	SL	reserved
764	0x02FC	Fault Log	UINT16	RO	1	FL	Fault Log
764	0x02FC	CRC	UINT16	RO	1	FL	Internal CRC check word
765	0x02FD		UINT16	RO	1	FL	reserved
766	0x02FE		UINT16	RO	16	FL	reserved
782	0x030E	FaultCount.0	UINT16	RO	1	FL	Count of faults observed for FAULT.0
783	0x030F	FaultCount.1	UINT16	RO	1	FL	Count of faults observed for FAULT.1
784	0x0310	FaultCount.2	UINT16	RO	1	FL	Count of faults observed for FAULT.2
785	0x0311	FaultCount.3	UINT16	RO	1	FL	Count of faults observed for FAULT.3
786	0x0312	FaultCount.4	UINT16	RO	1	FL	Count of faults observed for FAULT.4
787	0x0313	FaultCount.5	UINT16	RO	1	FL	Count of faults observed for FAULT.5
788	0x0314	FaultCount.6	UINT16	RO	1	FL	Count of faults observed for FAULT.6
789	0x0315	FaultCount.7	UINT16	RO	1	FL	Count of faults observed for FAULT.7
790	0x0316	FaultCount.8	UINT16	RO	1	FL	Count of faults observed for FAULT.8
791	0x0317	FaultCount.9	UINT16	RO	1	FL	Count of faults observed for FAULT.9
792	0x0318	FaultCount.10	UINT16	RO	1	FL	Count of faults observed for FAULT.10
793	0x0319	FaultCount.11	UINT16	RO	1	FL	Count of faults observed for FAULT.11
794	0x031A	FaultCount.12	UINT16	RO	1	FL	Count of faults observed for FAULT.12
795	0x031B	FaultCount.13	UINT16	RO	1	FL	Count of faults observed for FAULT.13
796	0x031C	FaultCount.14	UINT16	RO	1	FL	Count of faults observed for FAULT.14
797	0x031D	FaultCount.15	UINT16	RO	1	FL	Count of faults observed for FAULT.15
798	0x031E	PowerUpCount.0	UINT16	RO	1	FL	Power-up count at last FAULT.0 fault
799	0x031F	PowerUpCount.1	UINT16	RO	1	FL	Power-up count at last FAULT.1 fault
800	0x0320	PowerUpCount.2	UINT16	RO	1	FL	Power-up count at last FAULT.2 fault
801	0x0321	PowerUpCount.3	UINT16	RO	1	FL	Power-up count at last FAULT.3 fault
802	0x0322	PowerUpCount.4	UINT16	RO	1	FL	Power-up count at last FAULT.4 fault
803	0x0323	PowerUpCount.5	UINT16	RO	1	FL	Power-up count at last FAULT.5 fault
804	0x0324	PowerUpCount.6	UINT16	RO	1	FL	Power-up count at last FAULT.6 fault
805	0x0325	PowerUpCount.7	UINT16	RO	1	FL	Power-up count at last FAULT.7 fault
806	0x0326	PowerUpCount.8	UINT16	RO	1	FL	Power-up count at last FAULT.8 fault
807	0x0327	PowerUpCount.9	UINT16	RO	1	FL	Power-up count at last FAULT.9 fault
808	0x0328	PowerUpCount.10	UINT16	RO	1	FL	Power-up count at last FAULT.10 fault
809	0x0329	PowerUpCount.11	UINT16	RO	1	FL	Power-up count at last FAULT.11 fault
810	0x032A	PowerUpCount.12	UINT16	RO	1	FL	Power-up count of last FAULT.12 fault
811	0x032B	PowerUpCount.13	UINT16	RO	1	FL	Power-up count at last FAULT.13 fault
812	0x032C	PowerUpCount.14	UINT16	RO	1	FL	Power-up count at last FAULT.14 fault
813	0x032D	PowerUpCount.15	UINT16	RO	1	FL	Power-up count at last FAULT.15 fault
814	0x032E	PwmTime.0	UINT32	RO	2	FL	PWM run time at last FAULT.0 fault
816	0x0330	PwmTime.1	UINT32	RO	2	FL	PWM run time at last FAULT.1 fault
818	0x0332	PwmTime.2	UINT32	RO	2	FL	PWM run time at last FAULT.2 fault
820	0x0334	PwmTime.3	UINT32	RO	2	FL	PWM run time at last FAULT.3 fault
822	0x0336	PwmTime.4	UINT32	RO	2	FL	PWM run time at last FAULT.4 fault
824	0x0338	PwmTime.5	UINT32	RO	2	FL	PWM run time at last FAULT.5 fault
826	0x033A	PwmTime.6	UINT32	RO	2	FL	PWM run time at last FAULT.6 fault
828	0x033C	PwmTime.7	UINT32	RO	2	FL	PWM run time at last FAULT.7 fault
830	0x033E	PwmTime.8	UINT32	RO	2	FL	PWM run time at last FAULT.8 fault
832	0x0340	PwmTime.9	UINT32	RO	2	FL	PWM run time at last FAULT.9 fault
834	0x0342	PwmTime.10	UINT32	RO	2	FL	PWM run time at last FAULT.10 fault
836	0x0344	PwmTime.11	UINT32	RO	2	FL	PWM run time at last FAULT.11 fault
838	0x0346	PwmTime.12	UINT32	RO	2	FL	PWM run time at last FAULT.12 fault
840	0x0348	PwmTime.13	UINT32	RO	2	FL	PWM run time at last FAULT.13 fault
842	0x034A	PwmTime.14	UINT32	RO	2	FL	PWM run time at last FAULT.14 fault
844	0x034C	PwmTime.15	UINT32	RO	2	FL	PWM run time at last FAULT.15 fault
846	0x034E	RecentFault.0.ID	FAULT	RO	1	FL	FAULT ID of most recent fault
847	0x034F	RecentFault.0.PowerUpCount	UINT16	RO	1	FL	Power up count of most recent fault
848	0x0350	RecentFault.0.RunTime	UINT32	RO	2	FL	PWM run time of most recent fault
850	0x0352	RecentFault.1.ID	FAULT	RO	1	FL	FAULT ID of most recent fault
851	0x0353	RecentFault.1.PowerUpCount	UINT16	RO	1	FL	Power up count of most recent fault
852	0x0354	RecentFault.1.RunTime	UINT32	RO	2	FL	PWM run time of most recent fault

ID	ID (hex)	Name	Type	Access	Size	Storage	Description
854	0x0356	RecentFault.2.ID	FAULT	RO	1	FL	
855	0x0357	RecentFault.2.PowerUpCount	UINT16	RO	1	FL	
856	0x0358	RecentFault.2.RunTime	UINT32	RO	2	FL	
858	0x035A	RecentFault.3.ID	FAULT	RO	1	FL	
859	0x035B	RecentFault.3.PowerUpCount	UINT16	RO	1	FL	
860	0x035C	RecentFault.3.RunTime	UINT32	RO	2	FL	
862	0x035E	RecentFault.4.ID	FAULT	RO	1	FL	
863	0x035F	RecentFault.4.PowerUpCount	UINT16	RO	1	FL	
864	0x0360	RecentFault.4.RunTime	UINT32	RO	2	FL	
866	0x0362	RecentFault.5.ID	FAULT	RO	1	FL	
867	0x0363	RecentFault.5.PowerUpCount	UINT16	RO	1	FL	
868	0x0364	RecentFault.5.RunTime	UINT32	RO	2	FL	
870	0x0366	RecentFault.6.ID	FAULT	RO	1	FL	
871	0x0367	RecentFault.6.PowerUpCount	UINT16	RO	1	FL	
872	0x0368	RecentFault.6.RunTime	UINT32	RO	2	FL	
874	0x036A	RecentFault.7.ID	FAULT	RO	1	FL	
875	0x036B	RecentFault.7.PowerUpCount	UINT16	RO	1	FL	
876	0x036C	RecentFault.7.RunTime	UINT32	RO	2	FL	
878	0x036E	RecentFault.8.ID	FAULT	RO	1	FL	
879	0x036F	RecentFault.8.PowerUpCount	UINT16	RO	1	FL	
880	0x0370	RecentFault.8.RunTime	UINT32	RO	2	FL	
882	0x0372	RecentFault.9.ID	FAULT	RO	1	FL	FAULT ID of least recent fault
883	0x0373	RecentFault.9.PowerUpCount	UINT16	RO	1	FL	Power up count of least recent fault
884	0x0374	RecentFault.9.RunTime	UINT32	RO	2	FL	PWM run time of least recent fault
886	0x0376		UINT16	RO	6	FL	reserved
900	0x0384	DSP Analog Input Channels	UINT16	RO	16	RAM	DSP ADC inputs
900	0x0384	ADC0	UINT16	RO	1	RAM	DSP ADC Channel 0
901	0x0385	ADC1	UINT16	RO	1	RAM	DSP ADC Channel 1
902	0x0386	ADC2	UINT16	RO	1	RAM	DSP ADC Channel 2
903	0x0387	ADC3	UINT16	RO	1	RAM	DSP ADC Channel 3
904	0x0388	ADC4	UINT16	RO	1	RAM	DSP ADC Channel 4
904	0x0389	ADC5	UINT16	RO	1	RAM	DSP ADC Channel 5
906	0x038A	ADC6	UINT16	RO	1	RAM	DSP ADC Channel 6
907	0x038B	ADC7	UINT16	RO	1	RAM	DSP ADC Channel 7
908	0x038C	ADC8	UINT16	RO	1	RAM	DSP ADC Channel 8
909	0x038D	ADC9	UINT16	RO	1	RAM	DSP ADC Channel 9
910	0x038E	ADC10	UINT16	RO	1	RAM	DSP ADC Channel 10
911	0x038F	ADC11	UINT16	RO	1	RAM	DSP ADC Channel 11
912	0x0390	ADC12	UINT16	RO	1	RAM	DSP ADC Channel 12
913	0x0391	ADC13	UINT16	RO	1	RAM	DSP ADC Channel 13
914	0x0392	ADC14	UINT16	RO	1	RAM	DSP ADC Channel 14
915	0x0393	ADC15	UINT16	RO	1	RAM	DSP ADC Channel 15
4000	0x0FA0	System Command	UINT16	RW	2	RAM	System control registers
4000	0x0FA0	SecurityKey	UINT16	RW	1	RAM	Security key
4001	0x0FA1	Command	FLAGS	RW	1	RAM	System command / response flags
4002	0x0FA2	Maped Write Monitor	UINT16	RW	4	RAM	Mapped write diagnostics registers
4002	0x0FA2	MonitorID	UINT16	RW	1	RAM	MODBUS ID of mapped variable to monitor
4003	0x0FA3	MonitorExpectedValue	UINT16	RW	1	RAM	Expected value to be written variable
4004	0x0FA4	MonitorLatchedValue	UINT16	RW	1	RAM	Last unexpected value written
4005	0x0FA5	MonitorCount	UINT16	RW	1	RAM	Count of unexpected values written
4100	0x1004	Scope Control	UINT16	RW	16	RAM	Digital oscilloscope control registers
4100	0x1004	Channel1.Variable	UINT16	RW	1	RAM	Channel 1 variable ID
4101	0x1005	Channel1.Flags	FLAGS	RW	1	RAM	Channel 1 variable flags
4102	0x1006	Channel2.Variable	UINT16	RW	1	RAM	Channel 2 variable ID
4103	0x1007	Channel2.Flags	FLAGS	RW	1	RAM	Channel 2 variable flags
4104	0x1008	Channel3.Variable	UINT16	RW	1	RAM	Channel 3 variable ID
4105	0x1009	Channel3.Flags	FLAGS	RW	1	RAM	Channel 3 variable flags
4106	0x100A	Channel4.Variable	UINT16	RW	1	RAM	Channel 4 variable ID
4107	0x100B	Channel4.Flags	FLAGS	RW	1	RAM	Channel 4 variable flags
4108	0x100C	TriggerVariable	UINT16	RW	1	RAM	Trigger variable ID
4109	0x100D	TriggerFlags	FLAGS	RW	1	RAM	Trigger variable flags
4110	0x100E	TriggerLevel	INT32	RW	2	RAM	Trigger level
4112	0x1010	PreTrigger	UINT16	RW	1	RAM	Pre-trigger buffer size [UPDATES]
4113	0x1011		UINT16	RW	1	RAM	Reserved
4114	0x1012	UpdateRate	UINT16	RW	1	RAM	Update rate [SYS_TICKS]
4115	0x1013	Control	FLAGS	RW	1	RAM	Control command flags
4200	0x1068	Comms Diagnostics	UINT16	RW	8	RAM	Communication status registers
4200	0x1068	SCIA_errors	FLAGS	RW	1	RAM	SCI-A (RS485) error flags
4201	0x1069	SCIB_errors	FLAGS	RW	1	RAM	SCI-B (Ethernet) error flags
4202	0x106A	NetworkStatus	FLAGS	RW[2]	1	RAM	EXPORT network status

ID	ID (hex)	Name	Type	Access	Size	Storage	Description
4203	0x106B	ModuleStatus	FLAGS	RW[2]	1	RAM	EXPORT module status
4204	0x106C	MacAddress1	UINT16	RW[2]	1	RAM	MAC address 1 (set by XPORT)
4205	0x106D	MacAddress2	UINT16	RW[2]	1	RAM	MAC address 2 (set by XPORT)
4206	0x106E	MacAddress3	UINT16	RW[2]	1	RAM	MAC address 3 (set by XPORT)
4207	0x106F		UINT16	RW	1	RAM	Reserved
4308	0x10D4	Ethernet Address Mirror	UINT16	RW	6	RAM	Ethernet parameter mirror
4208	0x10D4	IP	UINT32	RW[1]	2	RAM	IP address (set by XPORT)
4210	0x10D6	SUBNET	UINT32	RW[1]	2	RAM	SUBNET address (set by XPORT)
4212	0x10D8	GATEWAY	UINT32	RW[1]	2	RAM	GATEWAY address (set by XPORT)
4300	0x10CC	Host Control	UINT16	RW	24	RAM	Host control registers
4300	0x10CC		UINT16	RW	2	RAM	reserved
4302	0x10CE	Disables	FLAGS	RW	1	RAM	Host disables
4303	0x10CF	CommandMode	UINT16	RW	1	RAM	Host direct command mode
4304	0x10D0	Position	POS32	RW	2	RAM	Host target position
4306	0x10D2	Velocity	VEL32	RW	2	RAM	Host velocity command / velocity limit
4308	0x10D4	Acceleration	UACC32	RW	2	RAM	Host acceleration limit
4310	0x10D6	Current	CUR16	RW	1	RAM	Host current command / current limit
4311	0x10D7	Voltage	VOLT16	RW	1	RAM	Host voltage command
4312	0x10D8	InputInhibits	FLAGS	RW	1	RAM	Host hardware input inhibits
4313	0x10D9	OutputInhibits	FLAGS	RW	1	RAM	Host hardware output inhibits
4314	0x10DA	Outputs	FLAGS	RW	1	RAM	Host direct outputs
4315	0x10DB	FactoryTest	UINT16	RW	1	RAM	Reserved for factory and internal testing
4316	0x10DC	IEG_MODE	UINT16	RW	1	RAM	Host Input Event Groups
4317	0x10DD	IEG_MOTION	UINT16	RW	1	RAM	
4318	0x10DE	IEG_MOVE_LEVEL	UINT16	RW	1	RAM	
4319	0x10DF	IEG_MOVE_EDGE	UINT16	RW	1	RAM	
4320	0x10E0	IEG_MOVE_TEACH	UINT16	RW	1	RAM	
4321	0x10E1	IEG_MOVE_SELECT	UINT16	RW	1	RAM	
4322	0x10E2	IEG_MOVE_SWITCH	UINT16	RW	1	RAM	
4323	0x10E3		UINT16	RW	1	RAM	reserved

[1] These registers are initialized from their corresponding user parameters at start-up. Writing these registers will cause the new value to be copied to the corresponding user parameter and ALL user parameters will be re-saved to non-volatile memory.

[2] These registers are normally written by the Ethernet/IP MODBUS master.

User Parameters Registers Table

ID	ID (hex)	Name	Type	Access	Size	Storage	Description
5000	0x1388	Identification	UINT16	RW	16	UP	
5000	0x1388	DriveName	STR16	RW[1]	16	UP	User drive name
5100	0x13EC	System Configuration	UINT16	RW	45	UP	Configuration option flags
5100	0x13EC	Options	FLAGS	RW	1	UP	Configuration option flags
5101	0x13ED	PowerUpDelay	UINT16	RW	1	UP	Power-up delay [0.01 sec]
5102	0x13EE	FaultDisables	FAULT	RW	1	UP	Output variable identifier to monitor on channel
5103	0x13EF	FaultWarnings	FAULT	RW	1	UP	DAC low calibration offset
5104	0x13F0	FaultStop	FAULT	RW	1	UP	DAC high calibration offset
5105	0x13F1	FaultDedicatedMove	FAULT	RW	1	UP	Minimum variable value
5106	0x13F2	DefaultCommandMode	OPMODE	RW	1	UP	Main command mode selector
5107	0x13F3	AltCommandMode	OPMODE	RW	1	UP	Alternate command mode selector
5108	0x13F4	Ipeak	UCUR16	RW	1	UP	User peak current command limit
5109	0x13F5		UINT16	RW	1	UP	reserved
5110	0x13F6	StopAccel	UACC32	RW	2	UP	Stop acceleration [12.20 RPS/S]
5112	0x13F8	InPositionWindow	UINT32	RW	2	UP	In position window width [16.16 REVS]
5114	0x13FA	MaxFollowingError	POS32	RW	2	UP	Following error limit [16.16 REVS]
5116	0x13FC	InPositionTime	UINT16	RW	1	UP	Time to in position [ms]
5117	0x13FD	MaxFollowingErrorTime	UINT16	RW	1	UP	Time to following error fault [0.01 sec]
5118	0x13FE	PlimitMinus	POS32	RW	2	UP	S/W position limit (-) [16.16 REVS]
5120	0x1400	PlimitPlus	POS32	RW	2	UP	S/W position limit (+) [16.16 REVS]
5122	0x1402	PlimitPercentMinus	UINT16	RW	1	UP	S/W position limit percent (-) [1.15 %]
5123	0x1403	PlimitPercentPlus	UINT16	RW	1	UP	S/W position limit percent (+) [1.15 %]
5124	0x1404	PlimitVelocity	UVEL32	RW	2	UP	Position limit velocity limit
5126	0x1406	PlimitIfoldback	UCUR16	RW	1	UP	Position limit fold-back current limit
5127	0x1407	PlimitIpeak	UCUR16	RW	1	UP	Position limit peak (seating) current
5128	0x1408	PlimitIpeakTime	UINT16	RW	1	UP	Position limit seating time [ms]
5129	0x1409	InCurrentLimitTime	UINT16	RW	1	UP	In current limit hysteresis time [0.1 sec]
5130	0x140A	FaultLogFaults	FAULT	RW	1	UP	Fault log enables

ID	ID (hex)	Name	Type	Access	Size	Storage	Description
5131	0x140B		UINT16	RW	1	UP	reserved
5132	0x140C	FaultResetDelay	UINT16	RW	1	UP	Fault auto-reset delay [0.01 sec]
5133	0x140D	FaultLogDelay	UINT16	RW	1	UP	Delay before logging faults [0.01 sec]
5134	0x140E	Comms_power_upt_time	UINT16	RW	1	UP	Comms fault power-up delay
5135	0x140F		UINT16	RW	1	UP	Reserved
5136	0x1410	Comms_fault_timeout_a	UINT16	RW	1	UP	Comms fault ch a cmd timeout
5137	0x1411	Comms_fault_timeout_b	UINT16	RW	1	UP	Comms fault ch b cmd timeout
5138	0x1412	Fault_log_faults2	UINT16	RW	1	UP	Second group of 16 fault enables
5139	0x1413	Fault_disables2	UINT16	RW	1	UP	Secong group of 16 fault drive disable flags
5140	0x1414	Fault_warnings2	UINT16	RW	1	UP	Second group of 16 fault drive warning flags
5141	0x1415	Fault_stop2	UINT16	RW	1	UP	Second group of 16 fault drive stop flags
5142	0x1416	Fault_dedicated_move2	UINT16	RW	1	UP	Second group of 16 fault dedicated move flags
5143	0x1417	Low_VoltageDC_Warn_On	INT16	RW	1	UP	Voltage at which Low voltage warning goes on
5144	0x1418	Low_VoltageDC_Warn_Off	INT16	RW	1	UP	Voltage at which Low voltage warning goes off
5150	0x141E	Ethernet	UINT16	RW	10	UP	
5150	0x141E	IP	UINT32	RW[1]	2	UP	IP (Internet Protocol) address
5152	0x1420	Subnet	UINT32	RW[1]	2	UP	Subnet address
5154	0x1422	Gateway	UINT32	RW[1]	2	UP	Gateway address
5156	0x1424		UINT16	RW	4	UP	reserved
5300	0x14B4	Modbus	UINT16	RW	6	UP	
5300	0x14B4	Flags	FLAGS	RW[1]	1	UP	Serial Channel A options
5301	0x14B5	AxisId	UINT16	RW[1]	1	UP[1]	Serial Channel A axis identifier
5302	0x14B6	Baud	BAUD	RW[1]	1	UP	Serial Channel A baud identifier
5303	0x14B7	CmdDelay	UINT16	RW	1	UP	Serial Channel A extra RX delay
5304	0x14B8	FrameDelay	UINT16	RW	1	UP	Serial Channel A extra TX delay
5305	0x14B9		UINT16	RW	1	UP	reserved
5400	0x1518	Tuning	UINT16	RW	6	UP	
5400	0x1518	KJ	UINT16	RW[1]	1	UP	Inertia gain
5401	0x1519		UINT16	RW	1	UP	reserved
5402	0x151A	KP	UINT16	RW[1]	1	UP	Position loop bandwidth
5403	0x151B	KI	UINT16	RW[1]	1	UP	Velocity integral time constant
5404	0x151C	KFF	UINT16	RW[1]	1	UP	Feed forward velocity scale
5405	0x151D	KD	UINT16	RW[1]	1	UP	Velocity damping
6000	0x1770	Home	UINT16	RW	10	UP	
6000	0x1770	Home.StartupOptions	FLAGS	RW	1	UP	Home option flags
6001	0x1771		UINT16	RW	1	UP	reserved
6002	0x1772	Home.Position1	POS32	RW	2	UP	Home reference position #1
6004	0x1774	Home.Position2	POS32	RW	2	UP	Home reference position #2
6006	0x1776		UINT16	RW	4	UP	reserved
6020	0x1784	Jog	UINT16	RW	8	UP	
6020	0x1784	Jog.Options	FLAGS	RW	1	UP	Jog option flags
6021	0x1785		UINT16	RW	1	UP	reserved
6022	0x1786	Jog.SlowVelocity	UVEL32	RW	2	UP	Jog slow velocity
6024	0x1788	Jog.FastVelocity	UVEL32	RW	2	UP	Jog fast velocity
6026	0x178A	Jog.Acceleration	UACC32	RW	2	UP	Jog acceleration
6100	0x17D4	Move.0	MOVE	RW	18	UP	Move 0 parameters
6100	0x17D4	Move.0.Options	FLAGS	RW	1	UP	Options
6101	0x17D5	Move.0.CurrentLimit	UCUR16	RW	1	UP	Current limit
6102	0x17D6	Move.0.Acceleration	UACC32	RW	2	UP	Acceleration
6104	0x17D8	Move.0.TerminationSwitch	UINT16	RW	1	UP	Termination switch
6105	0x17D9	Move.0.NextMove	UINT16	RW	1	UP	Next move
6106	0x17DA	Move.0.Primary.Options	FLAGS	RW	1	UP	Primary motion options
6107	0x17DB		UINT16	RW	1	UP	reserved
6108	0x17DC	Move.0.Primary.Position	POS32	RW	2	UP	Primary position/distance
6110	0x17DE	Move.0.Primary.Velocity	UVEL32	RW	2	UP	Primary velocity
6112	0x17E0	Move.0.Secondary.Options	FLAGS	RW	1	UP	Secondary motion options
6113	0x17E1		UINT16	RW	1	UP	reserved
6114	0x17E2	Move.0.Secondary.Position	POS32	RW	2	UP	Secondary position/distance
6116	0x17E4	Move.0.Secondary.Velocity	UVEL32	RW	2	UP	Secondary velocity
6118	0x17E6	Move.1	MOVE	RW	18	UP	Move 0 parameters
6118	0x17E6	Move.1.Options	FLAGS	RW	1	UP	Options
6119	0x17E7	Move.1.CurrentLimit	UCUR16	RW	1	UP	Current limit
6120	0x17E8	Move.1.Acceleration	UACC32	RW	2	UP	Acceleration
6122	0x17EA	Move.1.TerminationSwitch	UINT16	RW	1	UP	Termination switch
6123	0x17EB	Move.1.NextMove	UINT16	RW	1	UP	Next move
6124	0x17EC	Move.1.Primary.Options	FLAGS	RW	1	UP	Primary motion options
6125	0x17ED		UINT16	RW	1	UP	reserved
6126	0x17EE	Move.1.Primary.Position	POS32	RW	2	UP	Primary position/distance
6128	0x17F0	Move.1.Primary.Velocity	UVEL32	RW	2	UP	Primary velocity
6130	0x17F2	Move.1.Secondary.Options	FLAGS	RW	1	UP	Secondary motion options

ID	ID (hex)	Name	Type	Access	Size	Storage	Description
6131	0x17F3		UINT16	RW	1	UP	reserved
6132	0x17F4	Move.1.Secondary.Position	POS32	RW	2	UP	Secondary position/distance
6134	0x17F6	Move.1.Secondary.Velocity	UVEL32	RW	2	UP	Secondary velocity
6136	0x17F8	Move.2	MOVE	RW	18	UP	Move 0 parameters
6136	0x17F8	Move.2.Options	FLAGS	RW	1	UP	Options
6137	0x17F9	Move.2.CurrentLimit	UCUR16	RW	1	UP	Current limit
6138	0x17FA	Move.2.Acceleration	UACC32	RW	2	UP	Acceleration
6140	0x17FC	Move.2.TerminationSwitch	UINT16	RW	1	UP	Termination switch
6141	0x17FD	Move.2.NextMove	UINT16	RW	1	UP	Next move
6142	0x17FE	Move.2.Primary.Options	FLAGS	RW	1	UP	Primary motion options
6143	0x17FF		UINT16	RW	1	UP	reserved
6144	0x1800	Move.2.Primary.Position	POS32	RW	2	UP	Primary position/distance
6146	0x1802	Move.2.Primary.Velocity	UVEL32	RW	2	UP	Primary velocity
6148	0x1804	Move.2.Secondary.Options	FLAGS	RW	1	UP	Secondary motion options
6149	0x1805		UINT16	RW	1	UP	reserved
6150	0x1806	Move.2.Secondary.Position	POS32	RW	2	UP	Secondary position/distance
6152	0x1808	Move.2.Secondary.Velocity	UVEL32	RW	2	UP	Secondary velocity
6154	0x180A	Move.3	MOVE	RW	18	UP	Move 0 parameters
6154	0x180A	Move.3.Options	FLAGS	RW	1	UP	Options
6155	0x180B	Move.3.CurrentLimit	UCUR16	RW	1	UP	Current limit
6156	0x180C	Move.3.Acceleration	UACC32	RW	2	UP	Acceleration
6158	0x180E	Move.3.TerminationSwitch	UINT16	RW	1	UP	Termination switch
6159	0x180F	Move.3.NextMove	UINT16	RW	1	UP	Next move
6160	0x1810	Move.3.Primary.Options	FLAGS	RW	1	UP	Primary motion options
6161	0x1811		UINT16	RW	1	UP	reserved
6162	0x1812	Move.3.Primary.Position	POS32	RW	2	UP	Primary position/distance
6164	0x1814	Move.3.Primary.Velocity	UVEL32	RW	2	UP	Primary velocity
6166	0x1816	Move.3.Secondary.Options	FLAGS	RW	1	UP	Secondary motion options
6167	0x1817		UINT16	RW	1	UP	reserved
6168	0x1818	Move.3.Secondary.Position	POS32	RW	2	UP	Secondary position/distance
6170	0x181A	Move.3.Secondary.Velocity	UVEL32	RW	2	UP	Secondary velocity
6172	0x181C	Move.4	MOVE	RW	18	UP	Move 0 parameters
6172	0x181C	Move.4.Options	FLAGS	RW	1	UP	Options
6173	0x181D	Move.4.CurrentLimit	UCUR16	RW	1	UP	Current limit
6174	0x181E	Move.4.Acceleration	UACC32	RW	2	UP	Acceleration
6176	0x1820	Move.4.TerminationSwitch	UINT16	RW	1	UP	Termination switch
6177	0x1821	Move.4.NextMove	UINT16	RW	1	UP	Next move
6178	0x1822	Move.4.Primary.Options	FLAGS	RW	1	UP	Primary motion options
6179	0x1823		UINT16	RW	1	UP	reserved
6180	0x1824	Move.4.Primary.Position	POS32	RW	2	UP	Primary position/distance
6182	0x1826	Move.4.Primary.Velocity	UVEL32	RW	2	UP	Primary velocity
6184	0x1828	Move.4.Secondary.Options	FLAGS	RW	1	UP	Secondary motion options
6185	0x1829		UINT16	RW	1	UP	reserved
6186	0x182A	Move.4.Secondary.Position	POS32	RW	2	UP	Secondary position/distance
6188	0x182C	Move.4.Secondary.Velocity	UVEL32	RW	2	UP	Secondary velocity
6190	0x182E	Move.5	MOVE	RW	18	UP	Move 0 parameters
6190	0x182E	Move.5.Options	FLAGS	RW	1	UP	Options
6191	0x182F	Move.5.CurrentLimit	UCUR16	RW	1	UP	Current limit
6192	0x1830	Move.5.Acceleration	UACC32	RW	2	UP	Acceleration
6194	0x1832	Move.5.TerminationSwitch	UINT16	RW	1	UP	Termination switch
6195	0x1833	Move.5.NextMove	UINT16	RW	1	UP	Next move
6196	0x1834	Move.5.Primary.Options	FLAGS	RW	1	UP	Primary motion options
6197	0x1835		UINT16	RW	1	UP	reserved
6198	0x1836	Move.5.Primary.Position	POS32	RW	2	UP	Primary position/distance
6200	0x1838	Move.5.Primary.Velocity	UVEL32	RW	2	UP	Primary velocity
6202	0x183A	Move.5.Secondary.Options	FLAGS	RW	1	UP	Secondary motion options
6203	0x183B		UINT16	RW	1	UP	reserved
6204	0x183C	Move.5.Secondary.Position	POS32	RW	2	UP	Secondary position/distance
6206	0x183E	Move.5.Secondary.Velocity	UVEL32	RW	2	UP	Secondary velocity
6208	0x1840	Move.6	MOVE	RW	18	UP	Move 0 parameters
6208	0x1840	Move.6.Options	FLAGS	RW	1	UP	Options
6209	0x1841	Move.6.CurrentLimit	UCUR16	RW	1	UP	Current limit
6210	0x1842	Move.6.Acceleration	UACC32	RW	2	UP	Acceleration
6212	0x1844	Move.6.TerminationSwitch	UINT16	RW	1	UP	Termination switch
6213	0x1845	Move.6.NextMove	UINT16	RW	1	UP	Next move
6214	0x1846	Move.6.Primary.Options	FLAGS	RW	1	UP	Primary motion options
6215	0x1847		UINT16	RW	1	UP	reserved
6216	0x1848	Move.6.Primary.Position	POS32	RW	2	UP	Primary position/distance
6218	0x184A	Move.6.Primary.Velocity	UVEL32	RW	2	UP	Primary velocity
6220	0x184C	Move.6.Secondary.Options	FLAGS	RW	1	UP	Secondary motion options

ID	ID (hex)	Name	Type	Access	Size	Storage	Description
6221	0x184D		UINT16	RW	1	UP	reserved
6222	0x184E	Move.6.Secondary.Position	POS32	RW	2	UP	Secondary position/distance
6224	0x1850	Move.6.Secondary.Velocity	UVEL32	RW	2	UP	Secondary velocity
6226	0x1852	Move.7	MOVE	RW	18	UP	Move 0 parameters
6226	0x1852	Move.7.Options	FLAGS	RW	1	UP	Options
6227	0x1853	Move.7.CurrentLimit	UCUR16	RW	1	UP	Current limit
6228	0x1854	Move.7.Acceleration	UACC32	RW	2	UP	Acceleration
6230	0x1856	Move.7.TerminationSwitch	UINT16	RW	1	UP	Termination switch
6231	0x1857	Move.7.NextMove	UINT16	RW	1	UP	Next move
6232	0x1858	Move.7.Primary.Options	FLAGS	RW	1	UP	Primary motion options
6233	0x1859		UINT16	RW	1	UP	reserved
6234	0x185A	Move.7.Primary.Position	POS32	RW	2	UP	Primary position/distance
6236	0x185C	Move.7.Primary.Velocity	UVEL32	RW	2	UP	Primary velocity
6238	0x185E	Move.7.Secondary.Options	FLAGS	RW	1	UP	Secondary motion options
6239	0x185F		UINT16	RW	1	UP	reserved
6240	0x1860	Move.7.Secondary.Position	POS32	RW	2	UP	Secondary position/distance
6242	0x1862	Move.7.Secondary.Velocity	UVEL32	RW	2	UP	Secondary velocity
6244	0x1864	Move.8	MOVE	RW	18	UP	Move 0 parameters
6244	0x1864	Move.8.Options	FLAGS	RW	1	UP	Options
6245	0x1865	Move.8.CurrentLimit	UCUR16	RW	1	UP	Current limit
6246	0x1866	Move.8.Acceleration	UACC32	RW	2	UP	Acceleration
6248	0x1868	Move.8.TerminationSwitch	UINT16	RW	1	UP	Termination switch
6249	0x1869	Move.8.NextMove	UINT16	RW	1	UP	Next move
6250	0x186A	Move.8.Primary.Options	FLAGS	RW	1	UP	Primary motion options
6251	0x186B		UINT16	RW	1	UP	reserved
6252	0x186C	Move.8.Primary.Position	POS32	RW	2	UP	Primary position/distance
6254	0x186E	Move.8.Primary.Velocity	UVEL32	RW	2	UP	Primary velocity
6256	0x1870	Move.8.Secondary.Options	FLAGS	RW	1	UP	Secondary motion options
6257	0x1871		UINT16	RW	1	UP	reserved
6258	0x1872	Move.8.Secondary.Position	POS32	RW	2	UP	Secondary position/distance
6260	0x1874	Move.8.Secondary.Velocity	UVEL32	RW	2	UP	Secondary velocity
6262	0x1876	Move.9	MOVE	RW	18	UP	Move 0 parameters
6262	0x1876	Move.9.Options	FLAGS	RW	1	UP	Options
6263	0x1877	Move.9.CurrentLimit	UCUR16	RW	1	UP	Current limit
6264	0x1878	Move.9.Acceleration	UACC32	RW	2	UP	Acceleration
6266	0x187A	Move.9.TerminationSwitch	UINT16	RW	1	UP	Termination switch
6267	0x187B	Move.9.NextMove	UINT16	RW	1	UP	Next move
6268	0x187C	Move.9.Primary.Options	FLAGS	RW	1	UP	Primary motion options
6269	0x187D		UINT16	RW	1	UP	reserved
6270	0x187E	Move.9.Primary.Position	POS32	RW	2	UP	Primary position/distance
6272	0x1880	Move.9.Primary.Velocity	UVEL32	RW	2	UP	Primary velocity
6274	0x1882	Move.9.Secondary.Options	FLAGS	RW	1	UP	Secondary motion options
6275	0x1883		UINT16	RW	1	UP	reserved
6276	0x1884	Move.9.Secondary.Position	POS32	RW	2	UP	Secondary position/distance
6278	0x1886	Move.9.Secondary.Velocity	UVEL32	RW	2	UP	Secondary velocity
6280	0x1888	Move.10	MOVE	RW	18	UP	Move 0 parameters
6280	0x1888	Move.10.Options	FLAGS	RW	1	UP	Options
6281	0x1889	Move.10.CurrentLimit	UCUR16	RW	1	UP	Current limit
6282	0x188A	Move.10.Acceleration	UACC32	RW	2	UP	Acceleration
6284	0x188C	Move.10.TerminationSwitch	UINT16	RW	1	UP	Termination switch
6285	0x188D	Move.10.NextMove	UINT16	RW	1	UP	Next move
6286	0x188E	Move.10.Primary.Options	FLAGS	RW	1	UP	Primary motion options
6287	0x188F		UINT16	RW	1	UP	reserved
6288	0x1890	Move.10.Primary.Position	POS32	RW	2	UP	Primary position/distance
6290	0x1892	Move.10.Primary.Velocity	UVEL32	RW	2	UP	Primary velocity
6292	0x1894	Move.10.Secondary.Options	FLAGS	RW	1	UP	Secondary motion options
6293	0x1895		UINT16	RW	1	UP	reserved
6294	0x1896	Move.10.Secondary.Position	POS32	RW	2	UP	Secondary position/distance
6296	0x1898	Move.10.Secondary.Velocity	UVEL32	RW	2	UP	Secondary velocity
6298	0x189A	Move.11	MOVE	RW	18	UP	Move 0 parameters
6298	0x189A	Move.11.Options	FLAGS	RW	1	UP	Options
6299	0x189B	Move.11.CurrentLimit	UCUR16	RW	1	UP	Current limit
6300	0x189C	Move.11.Acceleration	UACC32	RW	2	UP	Acceleration
6302	0x189E	Move.11.TerminationSwitch	UINT16	RW	1	UP	Termination switch
6303	0x189F	Move.11.NextMove	UINT16	RW	1	UP	Next move
6304	0x18A0	Move.11.Primary.Options	FLAGS	RW	1	UP	Primary motion options
6305	0x18A1		UINT16	RW	1	UP	reserved
6306	0x18A2	Move.11.Primary.Position	POS32	RW	2	UP	Primary position/distance
6308	0x18A4	Move.11.Primary.Velocity	UVEL32	RW	2	UP	Primary velocity
6310	0x18A6	Move.11.Secondary.Options	FLAGS	RW	1	UP	Secondary motion options

ID	ID (hex)	Name	Type	Access	Size	Storage	Description
6311	0x18A7		UINT16	RW	1	UP	reserved
6312	0x18A8	Move.11.Secondary.Position	POS32	RW	2	UP	Secondary position/distance
6314	0x18AA	Move.11.Secondary.Velocity	UVEL32	RW	2	UP	Secondary velocity
6316	0x18AC	Move.12	MOVE	RW	18	UP	Move 0 parameters
6316	0x18AC	Move.12.Options	FLAGS	RW	1	UP	Options
6317	0x18AD	Move.12.CurrentLimit	UCUR16	RW	1	UP	Current limit
6318	0x18AE	Move.12.Acceleration	UACC32	RW	2	UP	Acceleration
6320	0x18B0	Move.12.TerminationSwitch	UINT16	RW	1	UP	Termination switch
6321	0x18B1	Move.12.NextMove	UINT16	RW	1	UP	Next move
6322	0x18B2	Move.12.Primary.Options	FLAGS	RW	1	UP	Primary motion options
6323	0x18B3		UINT16	RW	1	UP	reserved
6324	0x18B4	Move.12.Primary.Position	POS32	RW	2	UP	Primary position/distance
6326	0x18B6	Move.12.Primary.Velocity	UVEL32	RW	2	UP	Primary velocity
6328	0x18B8	Move.12.Secondary.Options	FLAGS	RW	1	UP	Secondary motion options
6329	0x18B9		UINT16	RW	1	UP	reserved
6330	0x18BA	Move.12.Secondary.Position	POS32	RW	2	UP	Secondary position/distance
6332	0x18BC	Move.12.Secondary.Velocity	UVEL32	RW	2	UP	Secondary velocity
6334	0x18BE	Move.13	MOVE	RW	18	UP	Move 0 parameters
6334	0x18BE	Move.13.Options	FLAGS	RW	1	UP	Options
6335	0x18BF	Move.13.CurrentLimit	UCUR16	RW	1	UP	Current limit
6336	0x18C0	Move.13.Acceleration	UACC32	RW	2	UP	Acceleration
6338	0x18C2	Move.13.TerminationSwitch	UINT16	RW	1	UP	Termination switch
6339	0x18C3	Move.13.NextMove	UINT16	RW	1	UP	Next move
6340	0x18C4	Move.13.Primary.Options	FLAGS	RW	1	UP	Primary motion options
6341	0x18C5		UINT16	RW	1	UP	reserved
6342	0x18C6	Move.13.Primary.Position	POS32	RW	2	UP	Primary position/distance
6344	0x18C8	Move.13.Primary.Velocity	UVEL32	RW	2	UP	Primary velocity
6346	0x18CA	Move.13.Secondary.Options	FLAGS	RW	1	UP	Secondary motion options
6347	0x18CB		UINT16	RW	1	UP	reserved
6348	0x18CC	Move.13.Secondary.Position	POS32	RW	2	UP	Secondary position/distance
6350	0x18CE	Move.13.Secondary.Velocity	UVEL32	RW	2	UP	Secondary velocity
6352	0x18D0	Move.14	MOVE	RW	18	UP	Move 0 parameters
6352	0x18D0	Move.14.Options	FLAGS	RW	1	UP	Options
6353	0x18D1	Move.14.CurrentLimit	UCUR16	RW	1	UP	Current limit
6354	0x18D2	Move.14.Acceleration	UACC32	RW	2	UP	Acceleration
6356	0x18D4	Move.14.TerminationSwitch	UINT16	RW	1	UP	Termination switch
6357	0x18D5	Move.14.NextMove	UINT16	RW	1	UP	Next move
6358	0x18D6	Move.14.Primary.Options	FLAGS	RW	1	UP	Primary motion options
6359	0x18D7		UINT16	RW	1	UP	reserved
6360	0x18D8	Move.14.Primary.Position	POS32	RW	2	UP	Primary position/distance
6362	0x18DA	Move.14.Primary.Velocity	UVEL32	RW	2	UP	Primary velocity
6364	0x18DC	Move.14.Secondary.Options	FLAGS	RW	1	UP	Secondary motion options
6365	0x18DD		UINT16	RW	1	UP	reserved
6366	0x18DE	Move.14.Secondary.Position	POS32	RW	2	UP	Secondary position/distance
6368	0x18E0	Move.14.Secondary.Velocity	UVEL32	RW	2	UP	Secondary velocity
6370	0x18E2	Move.15	MOVE	RW	18	UP	Move 0 parameters
6370	0x18E2	Move.15.Options	FLAGS	RW	1	UP	Options
6371	0x18E3	Move.15.CurrentLimit	UCUR16	RW	1	UP	Current limit
6372	0x18E4	Move.15.Acceleration	UACC32	RW	2	UP	Acceleration
6374	0x18E6	Move.15.TerminationSwitch	UINT16	RW	1	UP	Termination switch
6375	0x18E7	Move.15.NextMove	UINT16	RW	1	UP	Next move
6376	0x18E8	Move.15.Primary.Options	FLAGS	RW	1	UP	Primary motion options
6377	0x18E9		UINT16	RW	1	UP	reserved
6378	0x18EA	Move.15.Primary.Position	POS32	RW	2	UP	Primary position/distance
6380	0x18EC	Move.15.Primary.Velocity	UVEL32	RW	2	UP	Primary velocity
6382	0x18EE	Move.15.Secondary.Options	FLAGS	RW	1	UP	Secondary motion options
6383	0x18EF		UINT16	RW	1	UP	reserved
6384	0x18F0	Move.15.Secondary.Position	POS32	RW	2	UP	Secondary position/distance
6386	0x18F2	Move.15.Secondary.Velocity	UVEL32	RW	2	UP	Secondary velocity
6388	0x18F4	Emove	MOVE	RW	18	UP	Move 0 parameters
6388	0x18F4	.Emove.Options	FLAGS	RW	1	UP	Options
6389	0x18F5	Emove.CurrentLimit	UCUR16	RW	1	UP	Current limit
6390	0x18F6	Emove.Acceleration	UACC32	RW	2	UP	Acceleration
6392	0x18F8	Emove.TerminationSwitch	UINT16	RW	1	UP	Termination switch
6393	0x18F9	Emove.NextMove	UINT16	RW	1	UP	Next move
6394	0x18FA	Emove.Primary.Options	FLAGS	RW	1	UP	Primary motion options
6395	0x18FB		UINT16	RW	1	UP	reserved
6396	0x18FC	Emove.Primary.Position	POS32	RW	2	UP	Primary position/distance
6398	0x18FE	Emove.Primary.Velocity	UVEL32	RW	2	UP	Primary velocity
6400	0x1900	Emove.Secondary.Options	FLAGS	RW	1	UP	Secondary motion options

ID	ID (hex)	Name	Type	Access	Size	Storage	Description
6401	0x1901		UINT16	RW	1	UP	reserved
6402	0x1902	Emove.Secondary.Position	POS32	RW	2	UP	Secondary position/distance
6404	0x1904	Emove.Secondary.Velocity	UVEL32	RW	2	UP	Secondary velocity
6406	0x1906	Home.0	MOVE	RW	18	UP	Move 0 parameters
6406	0x1906	Home.0.Options	FLAGS	RW	1	UP	Options
6407	0x1907	Home.0.CurrentLimit	UCUR16	RW	1	UP	Current limit
6408	0x1908	Home.0.Acceleration	UACC32	RW	2	UP	Acceleration
6410	0x190A	Home.0.TerminationSwitch	UINT16	RW	1	UP	Termination switch
6411	0x190B	Home.0.NextMove	UINT16	RW	1	UP	Next move
6412	0x190C	Home.0.Primary.Options	FLAGS	RW	1	UP	Primary motion options
6413	0x190D		UINT16	RW	1	UP	reserved
6414	0x190E	Home.0.Primary.Position	POS32	RW	2	UP	Primary position/distance
6416	0x1910	Home.0.Primary.Velocity	UVEL32	RW	2	UP	Primary velocity
6418	0x1912	Home.0.Secondary.Options	FLAGS	RW	1	UP	Secondary motion options
6419	0x1913		UINT16	RW	1	UP	reserved
6420	0x1914	Home.0.Secondary.Position	POS32	RW	2	UP	Secondary position/distance
6422	0x1916	Home.0.Secondary.Velocity	UVEL32	RW	2	UP	Secondary velocity
6500	0x1964	Deceleration	UACC32	RW	36	UP	Decel rate for each move.
6500	0x1964	Decel0	UINT32	RW	2	UP	Decel move 0
6502	0x1966	Decel1	UINT32	RW	2	UP	Decel move 1
6504	0x1968	Decel2	UINT32	RW	2	Up	Decel move 2
6506	0x196A	Decel3	UINT32	RW	2	Up	Decel move3
6508	0x196C	Decel4	UINT32	RW	2	Up	Decel move4
6510	0x196E	Decel5	UINT32	RW	2	Up	Decel move5
6512	0x1970	Decel6	UINT32	RW	2	Up	Decel move6
6514	0x1972	Decel7	UINT32	RW	2	Up	Decel move7
6516	0x1974	Decel8	UINT32	RW	2	Up	Decel move8
6518	0x1976	Decel9	UINT32	RW	2	Up	Decel move9
6520	0x1978	Decel10	UINT32	RW	2	Up	Decal move10
6522	0x197A	Decel11	UINT32	RW	2	Up	Decel move11
6524	0x197C	Decel12	UINT32	RW	2	Up	Decel move12
6526	0x197E	Decel13	UINT32	RW	2	Up	Decel move13
6528	0x1980	Decel14	UINT32	RW	2	Up	Decel move14
6530	0x1982	Decel15	UINT32	RW	2	Up	Decel move15
6532	0x1984	Emove_Decel	UINT32	RW	2	Up	Decel emove
6534	0x1986	Home_Decel	UINT32	RW	2	Up	Decel home
7000	0x1B58	Digital I/O Polarities	UINT16	RW	2	UP	Polarity bitmaps
7000	0x1B58	InputPolarities	FLAGS	RW	1	UP	Reverse input polarity bitmap
7001	0x1B59	OutputPolarities	FLAGS	RW	1	UP	Reverse output polarity bitmap
7002	0x1B5A	Digitial Input Assignments	UINT16	RW	16	UP	
7002	0x1B5A	Input.1.GroupMap	IEG_XXX	RW	1	UP	Input 1 group map
7003	0x1B5B	Input1.Group	IGROUP	RW	1	UP	Input 1 event group
7004	0x1B5C	Input.2.GroupMap	IEG_XXX	RW	1	UP	Input 1 group map
7005	0x1B5D	Input.2.Group	IGROUP	RW	1	UP	Input 1 event group
7006	0x1B5E	Input.3.GroupMap	IEG_XXX	RW	1	UP	Input 1 group map
7007	0x1B5F	Input.3.Group	IGROUP	RW	1	UP	Input 1 event group
7008	0x1B60	Input.4.GroupMap	IEG_XXX	RW	1	UP	Input 1 group map
7009	0x1B61	Input.4.Group	IGROUP	RW	1	UP	Input 1 event group
7010	0x1B62	Input.5.GroupMap	IEG_XXX	RW	1	UP	Input 1 group map
7011	0x1B63	Input.5.Group	IGROUP	RW	1	UP	Input 1 event group
7012	0x1B64	Input.6.GroupMap	IEG_XXX	RW	1	UP	Input 1 group map
7013	0x1B65	Input.6.Group	IGROUP	RW	1	UP	Input 1 event group
7014	0x1B66	Input.7.GroupMap	IEG_XXX	RW	1	UP	Input 1 group map
7015	0x1B67	Input.7.Group	IGROUP	RW	1	UP	Input 1 event group
7016	0x1B68	Input.8.GroupMap	IEG_XXX	RW	1	UP	Input 1 group map
7017	0x1B69	Input.8.Group	IGROUP	RW	1	UP	Input 1 event group
7018	0x1B6A	Digitial Output Assignments	UINT16	RW	32	UP	
7018	0x1B6A	Output.1.GroupMap	OEG_XXX	RW	1	UP	Output 1 group map
7019	0x1B6B	Output.1.Group	OGROUP	RW	1	UP	Output 1 group
7020	0x1B6C	Output.2.GroupMap	OEG_XXX	RW	1	UP	Output 2 group map
7021	0x1B6D	Output.2.Group	OGROUP	RW	1	UP	Output 2 group
7022	0x1B6E	Output.3.GroupMap	OEG_XXX	RW	1	UP	Output 3 group map
7023	0x1B6F	Output.3.Group	OGROUP	RW	1	UP	Output 3 group
7024	0x1B70	Output.4.GroupMap	OEG_XXX	RW	1	UP	Output 4 group map
7026	0x1B71	Output.4.Group	OGROUP	RW	1	UP	Output 4 group
7026	0x1B72		UINT16	RW	8	UP	reserved
7034	0x1B7A	RedLed.GroupMap	OEG_XXX	RW	1	UP	Red LED group map
7035	0x1B7B	RedLed.Group	OGROUP	RW	1	UP	Red LED group
7036	0x1B7C	GrnLed.GroupMap	OEG_XXX	RW	1	UP	GRN LED group map
7037	0x1B7D	GrnLed.Group	OGROUP	RW	1	UP	GRN LED group

ID	ID (hex)	Name	Type	Access	Size	Storage	Description
7038	0x1B7E		UINT16	RW	2	UP	reserved
7040	0x1B80	Yel1Led.GroupMap	OEG_XXX	RW	1	UP	YEL1 LED group map
7041	0x1B81	Yel1Led.Group	OGROUP	RW	1	UP	YEL1 LED group
7042	0x1B82	Yel2Led.GroupMap	OEG_XXX	RW	1	UP	YEL2 LED group map
7043	0x1B83	Yel2Led.Group	OGROUP	RW	1	UP	YEL2 LED group
7044	0x1B84		UINT16	RW	6	UP	reserved
7100	0x1BBC	Analog Position	UINT16	RW	10	UP	
7100	0x1BBC		UINT16	RW	1	UP	reserved
7101	0x1BBD	Channel	UINT16	RW	1	UP	Analog input channel
7102	0x1BBE	Minimum	POS32	RW	2	UP	Position min
7104	0x1BC0	Maximum	POS32	RW	2	UP	Position max
7106	0x1BC2	Velocity	UVEL32	RW	2	UP	Velocity limit
7108	0x1BC4	Acceleration	UACC32	RW	2	UP	Acceleration limit
7116	0x1BCC	Analog Velocity	UINT16	RW	8	UP	
7116	0x1BCC		UINT16	RW	1	UP	reserved
7117	0x1BCD	Channel	UINT16	RW	1	UP	Analog input channel
7118	0x1BCE	Minimum	VEL32	RW	2	UP	Velocity min
7120	0x1BD0	Maximum	VEL32	RW	2	UP	Velocity max
7122	0x1BD2	Acceleration	UACC32	RW	2	UP	Acceleration limit
7132	0x1BDC	Analog Current	UINT16	RW	4	UP	
7132	0x1BDC		UINT16	RW	1	UP	reserved
7133	0x1BDD	Channel	UINT16	RW	1	UP	Analog input channel
7134	0x1BDE	Minimum	CUR16	RW	1	UP	Current min
7135	0x1BDF	Maximum	CUR16	RW	1	UP	Current max
7184	0x1C0C	Velocity Override Alternate	UINT16	RW	4	UP	
7184	0x1C0C	Alternate	UINT16	RW	1	UP	Holds Modbus alternative for Velocity Override.
7184	0x1C10	Velocity Override	UINT16	RW	4	UP	
7184	0x1C10		UINT16	RW	1	UP	reserved
7185	0x1C11	Channel	UINT16	RW	1	UP	Analog input channel
7186	0x1C12	Minimum	CUR16	RW	1	UP	Current min
7187	0x1C13	Maximum	CUR16	RW	1	UP	Current max
7188	0x1C14	Analog Alternates	UINT16	RW	4	UP	
7188	0x1C14	Alt_position	UINT16	RW	1	UP	Holds Modbus alternative for analog position.
7189	0x1C15	Alt_velocity	UINT16	RW	1	UP	Holds Modbus alternative for analog velocity.
7190	0x1C16	Alt_torque	UINT16	RW	1	UP	Holds Modbus alternative for analog torque.
7191	0x1C17	reserved	UINT16	RW	1	UP	Reserved.
7200	0x1C20	AnalogIn.1	UINT16	RW	30	UP	Analog Input 1
7200	0x1C20	AnalogIn.1.Options	FLAGS	RW	1	UP	Options
7201	0x1C21	AnalogIn.1.Bandwidth	UINT16	RW	1	UP	Filter bandwidth
7202	0x1C22	AnalogIn.1.Mode1UserLow	INT32	RW	2	UP	Mode 1 user low calibration
7204	0x1C24	AnalogIn.1.Mode1UserHigh	INT32	RW	2	UP	Mode 1 user high calibration
7206	0x1C26	AnalogIn.1.Mode1AdcLow	INT32	RW	2	UP	Mode 1 ADC low calibration
7208	0x1C28	AnalogIn.1.Mode1AdcHigh	INT32	RW	2	UP	Mode 1 ADC high calibration
7210	0x1C2A	AnalogIn.1.Mode2UserLow	INT32	RW	2	UP	Mode 2 user low calibration
7212	0x1C2C	AnalogIn.1.Mode2UserHigh	INT32	RW	2	UP	Mode 2 user high calibration]
7214	0x1C2E	AnalogIn.1.Mode2AdcLow	INT32	RW	2	UP	Mode 2 ADC low calibration
7216	0x1C30	AnalogIn.1.Mode2AdcHigh	INT32	RW	2	UP	Mode 2 ADC high calibration
7218	0x1C32	AnalogIn.1.RangeMinimum	INT32	RW	2	UP	Minimum value of useable range
7220	0x1C34	AnalogIn.1.RangeMaximum	INT32	RW	2	UP	Maximum value of useable range
7222	0x1C36	AnalogIn.1.FaultTripLow	INT32	RW	2	UP	Fault trip low value
7224	0x1C38	AnalogIn.1.FaultTripHigh	INT32	RW	2	UP	Fault trip high value
7226	0x1C3A		UINT16	RW	4	UP	reserved
7230	0x1C3E	AnalogIn.2	UINT16	RW	30	UP	Analog Input 2
7230	0x1C3E	AnalogIn.2.Options	FLAGS	RW	1	UP	Options
7231	0x1C3F	AnalogIn.2.Bandwidth	UINT16	RW	1	UP	Filter bandwidth
7232	0x1C40	AnalogIn.2.Mode1UserLow	INT32	RW	2	UP	Mode 1 user low calibration
7234	0x1C42	AnalogIn.2.Mode1UserHigh	INT32	RW	2	UP	Mode 1 user high calibration
7236	0x1C44	AnalogIn.2.Mode1AdcLow	INT32	RW	2	UP	Mode 1 ADC low calibration
7238	0x1C46	AnalogIn.2.Mode1AdcHigh	INT32	RW	2	UP	Mode 1 ADC high calibration
7240	0x1C48	AnalogIn.2.Mode2UserLow	INT32	RW	2	UP	Mode 2 user low calibration
7242	0x1C4A	AnalogIn.2.Mode2UserHigh	INT32	RW	2	UP	Mode 2 user high calibration]
7244	0x1C4C	AnalogIn.2.Mode2AdcLow	INT32	RW	2	UP	Mode 2 ADC low calibration
7246	0x1C4E	AnalogIn.2.Mode2AdcHigh	INT32	RW	2	UP	Mode 2 ADC high calibration
7248	0x1C50	AnalogIn.2.RangeMinimum	INT32	RW	2	UP	Minimum value of useable range
7250	0x1C52	AnalogIn.2.RangeMaximum	INT32	RW	2	UP	Maximum value of useable range
7252	0x1C54	AnalogIn.2.FaultTripLow	INT32	RW	2	UP	Fault trip low value
7254	0x1C56	AnalogIn.2.FaultTripHigh	INT32	RW	2	UP	Fault trip high value
7256	0x1C58		UINT16	RW	4	UP	reserved
7260	0x1C5C	AnalogIn.3 (reserved)	UINT16	RW	30	UP	
7290	0x1C7A	AnalogIn.4 (reserved)	UINT16	RW	30	UP	

7400	0x1CE8	AnalogOut.1	UINT16	RW	16	UP	Analog Output 1
7400	0x1CE8	AnalogOut.1.Options	FLAGS	RW	1	UP	Options
7401	0x1CE9	AnalogOut.1.Bandwidth	UINT16	RW	1	UP	Filter bandwidth
7402	0x1CEA	AnalogOut.1.Variable	UINT16	RW	1	UP	Variable ID
7403	0x1CEB	AnalogOut.1.Flags	UINT16	RW	1	UP	Variable flags
7404	0x1CEC	AnalogOut.1.CalLow	UINT16	RW	1	UP	DAC low calibration offset
7405	0x1CED	AnalogOut.1.CalHigh	UINT16	RW	1	UP	DAC high calibration offset
7406	0x1CEE	AnalogOut.1.VarMinimum	INT32	RW	2	UP	Minimum variable value
7408	0x1CF0	AnalogOut.1.VarMaximum	INT32	RW	2	UP	Maximum variable value
7410	0x1CF2		UINT16	RW	6	UP	reserved
7416	0x1CF8	AnalogOut.2	UINT16	RW	16	UP	Analog Output 2
7416	0x1CF8	AnalogOut.2.Options	FLAGS	RW	1	UP	Options
ID	ID (hex)	Name	Type	Access	Size	Storage	Description
7417	0x1CF9	AnalogOut.2.Bandwidth	UINT16	RW	1	UP	Filter bandwidth
7418	0x1CFA	AnalogOut.2.Variable	UINT16	RW	1	UP	Variable ID
7419	0x1CFB	AnalogOut.2.Flags	UINT16	RW	1	UP	Variable flags
7420	0x1CFC	AnalogOut.2.CalLow	UINT16	RW	1	UP	DAC low calibration offset
7421	0x1CFD	AnalogOut.2.CalHigh	UINT16	RW	1	UP	DAC high calibration offset
7422	0x1CFE	AnalogOut.2.VarMinimum	INT32	RW	2	UP	Minimum variable value
7424	0x1D00	AnalogOut.2.VarMaximum	INT32	RW	2	UP	Maximum variable value
7426	0x1D02		UINT16	RW	6	UP	reserved
7432	0x1D08	AnalogOut.3	UINT16	RW	16	UP	Analog Output 3
7432	0x1D0A	AnalogOut.3.Options	FLAGS	RW	1	UP	Options
7433	0x1D0C	AnalogOut.3.Bandwidth	UINT16	RW	1	UP	Filter bandwidth
7434	0x1D0E	AnalogOut.3.Variable	UINT16	RW	1	UP	Variable ID
7435	0x1D10	AnalogOut.3.Flags	UINT16	RW	1	UP	Variable flags
7436	0x1D12	AnalogOut.3.CalLow	UINT16	RW	1	UP	DAC low calibration offset
7437	0x1D14	AnalogOut.3.CalHigh	UINT16	RW	1	UP	DAC high calibration offset
7438	0x1D16	AnalogOut.3.VarMinimum	INT32	RW	2	UP	Minimum variable value
7440	0x1D18	AnalogOut.3.VarMaximum	INT32	RW	2	UP	Maximum variable value
7442	0x1D1A		UINT16	RW	6	UP	reserved
8000	0x1F40	MappedRead	UINT16	RW	100	UP	Mapped MODBUS ID Table 1
8200	0x2008	MappedWrite	UINT16	RW	100	UP	Mapped MODBUS ID Table 2
12500	0x30D4	New_faults_Fault Log	UINT16	RO	1	FL	Fault Log for new 16 faults
12500	0x30D4	CRC	UINT16	RO	1	FL	Internal CRC check word
12501	0x30D5		UINT16	RO	1	FL	reserved
12502	0x30D6	FaultCount.16	UINT16	RO	1	FL	Count of faults observed for FAULT.16
12503	0x30D7	FaultCount.17	UINT16	RO	1	FL	Count of faults observed for FAULT.17
12504	0x30D8	FaultCount.18	UINT16	RO	1	FL	Count of faults observed for FAULT.18
12505	0x30D9	FaultCount.19	UINT16	RO	1	FL	Count of faults observed for FAULT.19
12506	0x30DA	FaultCount.20	UINT16	RO	1	FL	Count of faults observed for FAULT.20
12507	0x30DB	FaultCount.21	UINT16	RO	1	FL	Count of faults observed for FAULT.21
12508	0x30DC	FaultCount.22	UINT16	RO	1	FL	Count of faults observed for FAULT.22
12509	0x30DD	FaultCount.23	UINT16	RO	1	FL	Count of faults observed for FAULT.23
12510	0x30DE	FaultCount.24	UINT16	RO	1	FL	Count of faults observed for FAULT.24
12511	0x30DF	FaultCount.25	UINT16	RO	1	FL	Count of faults observed for FAULT.25
12512	0x30E0	FaultCount.26	UINT16	RO	1	FL	Count of faults observed for FAULT.26
12513	0x30E1	FaultCount.27	UINT16	RO	1	FL	Count of faults observed for FAULT.27
12514	0x30E2	FaultCount.28	UINT16	RO	1	FL	Count of faults observed for FAULT.28
12515	0x30E3	FaultCount.29	UINT16	RO	1	FL	Count of faults observed for FAULT.29
12516	0x30E4	FaultCount.30	UINT16	RO	1	FL	Count of faults observed for FAULT.30
12517	0x30E5	FaultCount.31	UINT16	RO	1	FL	Count of faults observed for FAULT.31
12518	0x30E6	PowerUpCount.16	UINT16	RO	1	FL	Power-up count at last FAULT.16 fault
12519	0x30E7	PowerUpCount.17	UINT16	RO	1	FL	Power-up count at last FAULT.17 fault
12520	0x30E8	PowerUpCount.18	UINT16	RO	1	FL	Power-up count at last FAULT.18 fault
12521	0x30E9	PowerUpCount.19	UINT16	RO	1	FL	Power-up count at last FAULT.19 fault
12522	0x30EA	PowerUpCount.20	UINT16	RO	1	FL	Power-up count at last FAULT.20 fault
12523	0x30EB	PowerUpCount.21	UINT16	RO	1	FL	Power-up count at last FAULT.21 fault
12524	0x30EC	PowerUpCount.22	UINT16	RO	1	FL	Power-up count at last FAULT.22 fault
12525	0x30ED	PowerUpCount.23	UINT16	RO	1	FL	Power-up count at last FAULT.23 fault
12526	0x30EE	PowerUpCount.24	UINT16	RO	1	FL	Power-up count at last FAULT.24 fault
12527	0x30EF	PowerUpCount.25	UINT16	RO	1	FL	Power-up count at last FAULT.25 fault
12528	0x30F0	PowerUpCount.26	UINT16	RO	1	FL	Power-up count at last FAULT.26 fault
12529	0x30F1	PowerUpCount.27	UINT16	RO	1	FL	Power-up count at last FAULT.27 fault
12530	0x30F2	PowerUpCount.28	UINT16	RO	1	FL	Power-up count of last FAULT.28 fault
12531	0x30F3	PowerUpCount.29	UINT16	RO	1	FL	Power-up count at last FAULT.29 fault
12532	0x30F4	PowerUpCount.30	UINT16	RO	1	FL	Power-up count at last FAULT.30 fault
12533	0x30F5	PowerUpCount.31	UINT16	RO	1	FL	Power-up count at last FAULT.31 fault
12534	0x30F6	PwmTime.16	UINT32	RO	2	FL	PWM run time at last FAULT.16 fault
12536	0x30F8	PwmTime.17	UINT32	RO	2	FL	PWM run time at last FAULT.17 fault

12538	0x30FA	PwmTime.18	UINT32	RO	2	FL	PWM run time at last FAULT.18 fault
12540	0x30FC	PwmTime.19	UINT32	RO	2	FL	PWM run time at last FAULT.19 fault
12542	0x30FE	PwmTime.20	UINT32	RO	2	FL	PWM run time at last FAULT.20 fault
12544	0X3100	PwmTime.21	UINT32	RO	2	FL	PWM run time at last FAULT.21 fault
12546	0x3102	PwmTime.22	UINT32	RO	2	FL	PWM run time at last FAULT.22 fault
12548	0x3104	PwmTime.23	UINT32	RO	2	FL	PWM run time at last FAULT.23 fault
12550	0x3106	PwmTime.24	UINT32	RO	2	FL	PWM run time at last FAULT.24 fault
12552	0x3108	PwmTime.25	UINT32	RO	2	FL	PWM run time at last FAULT.25 fault
12554	0x310A	PwmTime.26	UINT32	RO	2	FL	PWM run time at last FAULT.26 fault
12556	0x310C	PwmTime.27	UINT32	RO	2	FL	PWM run time at last FAULT.27 fault
12558	0x310E	PwmTime.28	UINT32	RO	2	FL	PWM run time at last FAULT.28 fault
12560	0x3110	PwmTime.29	UINT32	RO	2	FL	PWM run time at last FAULT.29 fault
ID	ID (hex)	Name	Type	Access	Size	Storage	Description
12562	0x3112	PwmTime.30	UINT32	RO	2	FL	PWM run time at last FAULT.30 fault
12564	0x3114	PwmTime.31	UINT32	RO	2	FL	PWM run time at last FAULT.31 fault

[1] These registers are considered *drive specific* and are not normally written by the Tritex user interface software during a user parameter download operation.

ID	ID (hex)	Name	Type	Access	Size	Storage	Description
8400	0x20D0	MappedReadValues	UINT16	RW	100	NONE	Indirect read/write table values
8400	0x20D0	MappedReadValue.0	UINT16	RW	1	NONE	
8401	0x20D1	MappedReadValue.1	UINT16	RW	1	NONE	
8402	0x20D2	MappedReadValue.2	UINT16	RW	1	NONE	
8403	0x20D3	MappedReadValue.3	UINT16	RW	1	NONE	
8404	0x20D4	MappedReadValue.4	UINT16	RW	1	NONE	
8405	0x20D5	MappedReadValue.5	UINT16	RW	1	NONE	
8406	0x20D6	MappedReadValue.6	UINT16	RW	1	NONE	
8407	0x20D7	MappedReadValue.7	UINT16	RW	1	NONE	
8408	0x20D8	MappedReadValue.8	UINT16	RW	1	NONE	
8409	0x20D9	MappedReadValue.9	UINT16	RW	1	NONE	
8410	0x20DA	MappedReadValue.10	UINT16	RW	1	NONE	
8411	0x20DB	MappedReadValue.11	UINT16	RW	1	NONE	
8412	0x20DC	MappedReadValue.12	UINT16	RW	1	NONE	
8413	0x20DD	MappedReadValue.13	UINT16	RW	1	NONE	
8414	0x20DE	MappedReadValue.14	UINT16	RW	1	NONE	
8415	0x20DF	MappedReadValue.15	UINT16	RW	1	NONE	
8416	0x20E0	MappedReadValue.16	UINT16	RW	1	NONE	
8417	0x20E1	MappedReadValue.17	UINT16	RW	1	NONE	
8418	0x20E2	MappedReadValue.18	UINT16	RW	1	NONE	
8419	0x20E3	MappedReadValue.19	UINT16	RW	1	NONE	
8420	0x20E4	MappedReadValue.2	UINT16	RW	1	NONE	
8421	0x20E5	MappedReadValue.21	UINT16	RW	1	NONE	
8422	0x20E6	MappedReadValue.22	UINT16	RW	1	NONE	
8423	0x20E7	MappedReadValue.23	UINT16	RW	1	NONE	
8424	0x20E8	MappedReadValue.24	UINT16	RW	1	NONE	
8425	0x20E9	MappedReadValue.25	UINT16	RW	1	NONE	
8426	0x20EA	MappedReadValue.26	UINT16	RW	1	NONE	
8427	0x20EB	MappedReadValue.27	UINT16	RW	1	NONE	
8428	0x20EC	MappedReadValue.28	UINT16	RW	1	NONE	
8429	0x20ED	MappedReadValue.29	UINT16	RW	1	NONE	
8430	0x20EE	MappedReadValue.30	UINT16	RW	1	NONE	
8431	0x20EF	MappedReadValue.31	UINT16	RW	1	NONE	
8432	0x20F0	MappedReadValue.32	UINT16	RW	1	NONE	
8433	0x20F1	MappedReadValue.33	UINT16	RW	1	NONE	
8434	0x20F2	MappedReadValue.34	UINT16	RW	1	NONE	
8435	0x20F3	MappedReadValue.35	UINT16	RW	1	NONE	
8436	0x20F4	MappedReadValue.36	UINT16	RW	1	NONE	
8437	0x20F5	MappedReadValue.37	UINT16	RW	1	NONE	
8438	0x20F6	MappedReadValue.38	UINT16	RW	1	NONE	
8439	0x20F7	MappedReadValue.39	UINT16	RW	1	NONE	
8440	0x20F8	MappedReadValue.40	UINT16	RW	1	NONE	
8441	0x20F9	MappedReadValue.41	UINT16	RW	1	NONE	
8442	0x20FA	MappedReadValue.42	UINT16	RW	1	NONE	
8443	0x20FB	MappedReadValue.43	UINT16	RW	1	NONE	
8444	0x20FC	MappedReadValue.44	UINT16	RW	1	NONE	
8445	0x20FD	MappedReadValue.45	UINT16	RW	1	NONE	
8446	0x20FE	MappedReadValue.46	UINT16	RW	1	NONE	
8447	0x20FF	MappedReadValue.47	UINT16	RW	1	NONE	
8448	0x2100	MappedReadValue.48	UINT16	RW	1	NONE	
8449	0x2101	MappedReadValue.49	UINT16	RW	1	NONE	
8450	0x2102	MappedReadValue.50	UINT16	RW	1	NONE	
8451	0x2103	MappedReadValue.51	UINT16	RW	1	NONE	
8452	0x2104	MappedReadValue.52	UINT16	RW	1	NONE	
8453	0x2105	MappedReadValue.53	UINT16	RW	1	NONE	
8454	0x2106	MappedReadValue.54	UINT16	RW	1	NONE	
8455	0x2107	MappedReadValue.55	UINT16	RW	1	NONE	
8456	0x2108	MappedReadValue.56	UINT16	RW	1	NONE	
8457	0x2109	MappedReadValue.57	UINT16	RW	1	NONE	
8458	0x210A	MappedReadValue.58	UINT16	RW	1	NONE	
8459	0x210B	MappedReadValue.59	UINT16	RW	1	NONE	
8460	0x210C	MappedReadValue.60	UINT16	RW	1	NONE	
8461	0x210D	MappedReadValue.61	UINT16	RW	1	NONE	
8462	0x210E	MappedReadValue.62	UINT16	RW	1	NONE	
8463	0x210F	MappedReadValue.63	UINT16	RW	1	NONE	
8464	0x2110	MappedReadValue.64	UINT16	RW	1	NONE	

ID	ID (hex)	Name	Type	Access	Size	Storage	Description
8465	0x2111	MappedReadValue.65	UINT16	RW	1	NONE	
8466	0x2112	MappedReadValue.66	UINT16	RW	1	NONE	
8467	0x2113	MappedReadValue.67	UINT16	RW	1	NONE	
8468	0x2114	MappedReadValue.68	UINT16	RW	1	NONE	
8469	0x2115	MappedReadValue.69	UINT16	RW	1	NONE	
8470	0x2116	MappedReadValue.70	UINT16	RW	1	NONE	
8471	0x2117	MappedReadValue.71	UINT16	RW	1	NONE	
8472	0x2118	MappedReadValue.72	UINT16	RW	1	NONE	
8473	0x2119	MappedReadValue.73	UINT16	RW	1	NONE	
8474	0x211A	MappedReadValue.74	UINT16	RW	1	NONE	
8475	0x211B	MappedReadValue.75	UINT16	RW	1	NONE	
8476	0x211C	MappedReadValue.76	UINT16	RW	1	NONE	
8477	0x211D	MappedReadValue.77	UINT16	RW	1	NONE	
8478	0x211E	MappedReadValue.78	UINT16	RW	1	NONE	
8479	0x211F	MappedReadValue.79	UINT16	RW	1	NONE	
8480	0x2120	MappedReadValue.80	UINT16	RW	1	NONE	
8481	0x2121	MappedReadValue.81	UINT16	RW	1	NONE	
8482	0x2122	MappedReadValue.82	UINT16	RW	1	NONE	
8483	0x2123	MappedReadValue.83	UINT16	RW	1	NONE	
8484	0x2124	MappedReadValue.84	UINT16	RW	1	NONE	
8485	0x2125	MappedReadValue.85	UINT16	RW	1	NONE	
8486	0x2126	MappedReadValue.86	UINT16	RW	1	NONE	
8487	0x2127	MappedReadValue.87	UINT16	RW	1	NONE	
8488	0x2128	MappedReadValue.88	UINT16	RW	1	NONE	
8489	0x2129	MappedReadValue.89	UINT16	RW	1	NONE	
8490	0x212A	MappedReadValue.90	UINT16	RW	1	NONE	
8491	0x212B	MappedReadValue.91	UINT16	RW	1	NONE	
8492	0x212C	MappedReadValue.92	UINT16	RW	1	NONE	
8493	0x212D	MappedReadValue.93	UINT16	RW	1	NONE	
8494	0x212E	MappedReadValue.94	UINT16	RW	1	NONE	
8495	0x212F	MappedReadValue.95	UINT16	RW	1	NONE	
8496	0x2130	MappedReadValue.96	UINT16	RW	1	NONE	
8497	0x2131	MappedReadValue.97	UINT16	RW	1	NONE	
8498	0x2132	MappedReadValue.98	UINT16	RW	1	NONE	
8499	0x2133	MappedReadValue.99	UINT16	RW	1	NONE	
8600	0x2198	MappedWriteValues	UINT16	RW	100	NONE	Indirect read/write table values
8600	0x2198	MappedWriteValue.0	UINT16	RW	1	NONE	
8601	0x2199	MappedWriteValue.1	UINT16	RW	1	NONE	
8602	0x219A	MappedWriteValue.2	UINT16	RW	1	NONE	
8603	0x219B	MappedWriteValue.3	UINT16	RW	1	NONE	
8604	0x219C	MappedWriteValue.4	UINT16	RW	1	NONE	
8605	0x219D	MappedWriteValue.5	UINT16	RW	1	NONE	
8606	0x219E	MappedWriteValue.6	UINT16	RW	1	NONE	
8607	0x219F	MappedWriteValue.7	UINT16	RW	1	NONE	
8608	0x21A0	MappedWriteValue.8	UINT16	RW	1	NONE	
8609	0x21A1	MappedWriteValue.9	UINT16	RW	1	NONE	
8610	0x21A2	MappedWriteValue.10	UINT16	RW	1	NONE	
8611	0x21A3	MappedWriteValue.11	UINT16	RW	1	NONE	
8612	0x21A4	MappedWriteValue.12	UINT16	RW	1	NONE	
8613	0x21A5	MappedWriteValue.13	UINT16	RW	1	NONE	
8614	0x21A6	MappedWriteValue.14	UINT16	RW	1	NONE	
8615	0x21A7	MappedWriteValue.15	UINT16	RW	1	NONE	
8616	0x21A8	MappedWriteValue.16	UINT16	RW	1	NONE	
8617	0x21A9	MappedWriteValue.17	UINT16	RW	1	NONE	
8618	0x21AA	MappedWriteValue.18	UINT16	RW	1	NONE	
8619	0x21AB	MappedWriteValue.19	UINT16	RW	1	NONE	
8620	0x21AC	MappedWriteValue.20	UINT16	RW	1	NONE	
8621	0x21AD	MappedWriteValue.21	UINT16	RW	1	NONE	
8622	0x21AE	MappedWriteValue.22	UINT16	RW	1	NONE	
8623	0x21AF	MappedWriteValue.23	UINT16	RW	1	NONE	
8624	0x21B0	MappedWriteValue.24	UINT16	RW	1	NONE	
8625	0x21B1	MappedWriteValue.25	UINT16	RW	1	NONE	
8626	0x21B2	MappedWriteValue.26	UINT16	RW	1	NONE	
8627	0x21B3	MappedWriteValue.27	UINT16	RW	1	NONE	
8628	0x21B4	MappedWriteValue.28	UINT16	RW	1	NONE	
8629	0x21B5	MappedWriteValue.29	UINT16	RW	1	NONE	
8630	0x21B6	MappedWriteValue.30	UINT16	RW	1	NONE	
8631	0x21B7	MappedWriteValue.31	UINT16	RW	1	NONE	
8632	0x21B8	MappedWriteValue.32	UINT16	RW	1	NONE	
8633	0x21B9	MappedWriteValue.33	UINT16	RW	1	NONE	

ID	ID (hex)	Name	Type	Access	Size	Storage	Description
8634	0x21BA	MappedWriteValue.34	UINT16	RW	1	NONE	
8635	0x21BB	MappedWriteValue.35	UINT16	RW	1	NONE	
8636	0x21BC	MappedWriteValue.36	UINT16	RW	1	NONE	
8637	0x21BD	MappedWriteValue.37	UINT16	RW	1	NONE	
8638	0x21BE	MappedWriteValue.38	UINT16	RW	1	NONE	
8639	0x21BF	MappedWriteValue.39	UINT16	RW	1	NONE	
8640	0x21C0	MappedWriteValue.40	UINT16	RW	1	NONE	
8641	0x21C1	MappedWriteValue.41	UINT16	RW	1	NONE	
8642	0x21C2	MappedWriteValue.42	UINT16	RW	1	NONE	
8643	0x21C3	MappedWriteValue.43	UINT16	RW	1	NONE	
8644	0x21C4	MappedWriteValue.44	UINT16	RW	1	NONE	
8645	0x21C5	MappedWriteValue.45	UINT16	RW	1	NONE	
8646	0x21C6	MappedWriteValue.46	UINT16	RW	1	NONE	
8647	0x21C7	MappedWriteValue.47	UINT16	RW	1	NONE	
8648	0x21C8	MappedWriteValue.48	UINT16	RW	1	NONE	
8649	0x21C9	MappedWriteValue.49	UINT16	RW	1	NONE	
8650	0x21CA	MappedWriteValue.50	UINT16	RW	1	NONE	
8651	0x21CB	MappedWriteValue.51	UINT16	RW	1	NONE	
8652	0x21CC	MappedWriteValue.52	UINT16	RW	1	NONE	
8653	0x21CD	MappedWriteValue.53	UINT16	RW	1	NONE	
8654	0x21CE	MappedWriteValue.54	UINT16	RW	1	NONE	
8655	0x21CF	MappedWriteValue.55	UINT16	RW	1	NONE	
8656	0x21D0	MappedWriteValue.56	UINT16	RW	1	NONE	
8657	0x21D1	MappedWriteValue.57	UINT16	RW	1	NONE	
8658	0x21D2	MappedWriteValue.58	UINT16	RW	1	NONE	
8659	0x21D3	MappedWriteValue.59	UINT16	RW	1	NONE	
8660	0x21D4	MappedWriteValue.60	UINT16	RW	1	NONE	
8661	0x21D5	MappedWriteValue.61	UINT16	RW	1	NONE	
8662	0x21D6	MappedWriteValue.62	UINT16	RW	1	NONE	
8663	0x21D7	MappedWriteValue.63	UINT16	RW	1	NONE	
8664	0x21D8	MappedWriteValue.64	UINT16	RW	1	NONE	
8665	0x21D9	MappedWriteValue.65	UINT16	RW	1	NONE	
8666	0x21DA	MappedWriteValue.66	UINT16	RW	1	NONE	
8667	0x21DB	MappedWriteValue.67	UINT16	RW	1	NONE	
8668	0x21DC	MappedWriteValue.68	UINT16	RW	1	NONE	
8669	0x21DD	MappedWriteValue.69	UINT16	RW	1	NONE	
8670	0x21DE	MappedWriteValue.70	UINT16	RW	1	NONE	
8671	0x21DF	MappedWriteValue.71	UINT16	RW	1	NONE	
8672	0x21E0	MappedWriteValue.72	UINT16	RW	1	NONE	
8673	0x21E1	MappedWriteValue.73	UINT16	RW	1	NONE	
8674	0x21E2	MappedWriteValue.74	UINT16	RW	1	NONE	
8675	0x21E3	MappedWriteValue.75	UINT16	RW	1	NONE	
8676	0x21E4	MappedWriteValue.76	UINT16	RW	1	NONE	
8677	0x21E5	MappedWriteValue.77	UINT16	RW	1	NONE	
8678	0x21E6	MappedWriteValue.78	UINT16	RW	1	NONE	
8679	0x21E7	MappedWriteValue.79	UINT16	RW	1	NONE	
8680	0x21E8	MappedWriteValue.80	UINT16	RW	1	NONE	
8681	0x21E9	MappedWriteValue.81	UINT16	RW	1	NONE	
8682	0x21EA	MappedWriteValue.82	UINT16	RW	1	NONE	
8683	0x21EB	MappedWriteValue.83	UINT16	RW	1	NONE	
8684	0x21EC	MappedWriteValue.84	UINT16	RW	1	NONE	
8685	0x21ED	MappedWriteValue.85	UINT16	RW	1	NONE	
8686	0x21EE	MappedWriteValue.86	UINT16	RW	1	NONE	
8687	0x21EF	MappedWriteValue.87	UINT16	RW	1	NONE	
8688	0x21F0	MappedWriteValue.88	UINT16	RW	1	NONE	
8689	0x21F1	MappedWriteValue.89	UINT16	RW	1	NONE	
8690	0x21F2	MappedWriteValue.90	UINT16	RW	1	NONE	
8691	0x21F3	MappedWriteValue.91	UINT16	RW	1	NONE	
8692	0x21F4	MappedWriteValue.92	UINT16	RW	1	NONE	
8693	0x21F5	MappedWriteValue.93	UINT16	RW	1	NONE	
8694	0x21F6	MappedWriteValue.94	UINT16	RW	1	NONE	
8695	0x21F7	MappedWriteValue.95	UINT16	RW	1	NONE	
8696	0x21F8	MappedWriteValue.96	UINT16	RW	1	NONE	
8697	0x21F9	MappedWriteValue.97	UINT16	RW	1	NONE	
8698	0x21FA	MappedWriteValue.98	UINT16	RW	1	NONE	
8699	0x21FB	MappedWriteValue.99	UINT16	RW	1	NONE	

Factory Parameters Register Table

ID	ID (hex)	Name	Type	Access	Size	Storage	Description
9000	0x2328	Identification	UINT16	RW	32	FP	Factory identification parameters
9000	0x2328	PartNumber	STR16	RW	16	FP	Factory part number
9016	0x2338	SerialNumber	STR16	RW	16	FP	Factory serial number
9100	0x238C	Actuator	UINT16	RW	26	FP	Actuator parameters
9100	0x238C	Model	STR16	RW	16	FP	Model name
9116	0x239C	Options	FLAGS	RW	1	FP	Option flags
9117	0x238D	EcyclesPerRev	UINT16	RW	1	FP	Poles / 2
9118	0x238E	R	UINT16	RW	1	FP	Resistance [8.8 ohms L-L]
9119	0x238F	L	UINT16	RW	1	FP	Inductance [8.8 mH L-L]
9120	0x2390	J	UINT32	RW	2	FP	Inertia [0.32 KG-M^2]
9122	0x2392	KT	UINT16	RW	1	FP	KT [6.10 NM/AMP]
9123	0x2393	FeedbackDevice	UINT16	RW	1	FP	Position feedback device type
9124	0x2394	StepsPerRev	UINT32	RW	2	FP	Encoder steps/rev
9200	0x23F0	Limits	UINT16	RW	14	FP	Factory limits parameters
9200	0x23F0	LowVoltageTripLevel	UVOLT16	RW	1	FP	Low voltage fault trip level
9201	0x23F1	HighVoltageTripLevel	UVOLT16	RW	1	FP	High voltage fault trip level
9202	0x23F2	BoardTempTripLevel	INT16	RW	1	FP	PCB board temperature trip level
9203	0x23F3	Itrip	UCUR16	RW	1	FP	Current fault trip level
9204	0x23F4	Ipeak	UCUR16	RW	1	FP	Peak command current
9205	0x23F5	Icontinuous	UCUR16	RW	1	FP	Continuous current rating
9206	0x23F6	IcTimeConstant	INT16	RW	1	FP	Continuous current time constant
9207	0x23F7		UINT16	RW	1	FP	reserved
9208	0x23F8		UINT16	RW	1	FP	reserved
9209	0x23F9	ActuatorTempTripLevel	INT16	RW	1	FP	Actuator temperature fault trip level
9210	0x23FA	PwmModulation	INT16	RW	1	FP	PWM modulation factor
9211	0x23FB	ShuntHigh	UINT16	RW	1	FP	High shunt level (shunt ON)
9212	0x23FC	ShuntLow	UINT16	RW	1	FP	Low shunt level (shunt OFF)
9213	0x23FD		UINT16	RW	1	FP	reserved
9300	0X2454	Initialization	UINT16	RW	30	FP	Initialization / calibration parameters
9300	0X2454	Options	FLAGS	RW	1	FP	Option and option board flags
9301	0X2455	VbusScale	INT16	RW	1	FP	DC Bus voltage scale
9302	0X2456	BoardTempOffset	INT16	RW	1	FP	PCB board temperature offset
9303	0X2457	BoardTempScale	INT16	RW	1	FP	PCB board temperature scale
9304	0X2458	PsineInZero	INT16	RW	1	FP	Peak command current
9305	0X2459	PsineInScale	INT16	RW	1	FP	Continuous current rating
9306	0X245A	PcosineInZero	INT16	RW	1	FP	Continuous current time constant
9307	0X245B	PcosineInScale	INT16	RW	1	FP	reserved
9308	0X245C	RphaseOffset	INT16	RW	1	FP	reserved
9309	0X245D	RphaseScale	INT16	RW	1	FP	Actuator temperature fault trip level
9310	0X245E	SphaseOffset	INT16	RW	1	FP	PWM modulation factor
9311	0X245F	SphaseScale	INT16	RW	1	FP	High shunt level (shunt ON)
9312	0X2460		UINT16	RW	1	FP	reserved
9313	0X2461		UINT16	RW	1	FP	reserved
9314	0X2462		UINT16	RW	1	FP	reserved
9315	0X2463	Eoffset	INT16	RW	1	FP	Electrical angle offset
9316	0X2464		UINT16	RW	1	FP	reserved
9317	0X2465		UINT16	RW	1	FP	reserved
9318	0X2466		UINT16	RW	1	FP	reserved
9319	0X2467		UINT16	RW	1	FP	reserved
9320	0X2468	BrakeReleaseDelay	UINT16	RW	1	FP	Brake release delay [0.01s]
9321	0X2469	BrakeEngageDelay	UINT16	RW	1	FP	Brake engage delay [0.01s]
9322	0X246A		UINT16	RW	1	FP	reserved
9323	0X246B	ThармonicMag	INT16	RW	1	FP	Torque harmonic magnitude [2.14 AMPS]
9324	0X246C		UINT16	RW	1	FP	reserved
9325	0X246D		UINT16	RW	1	FP	reserved
9326	0X246E		UINT16	RW	1	FP	reserved
9327	0X246F		UINT16	RW	1	FP	reserved
9328	0X2470	ActuatorTempOffset	INT16	RW	1	FP	Actuator temperature offset [13.3 DEG]
9329	0X2471	ActuatorTempScale	INT16	RW	1	FP	Actuator temp scale [13.3 DEG/ADFULLSCALE]
9400	0X24B8	Tuning	UINT16	RW	8	FP	Tuning parameters
9400	0X24B8	Options	FLAGS	RW	1	FP	Option flags
9401	0X24B9	HallBW	UINT16	RW	1	FP	Position angle tracking bandwidth
9402	0X24BA	HallDamping	UINT16	RW	1	FP	Position angle tracking damping
9403	0X24BB	IloopBW	UINT16	RW	1	FP	Current loop bandwidth [HZ]
9404	0X24BC		UINT16	RW	1	FP	reserved
9405	0X24BD	VbusBW	UINT16	RW	1	FP	Bus voltage filter bandwidth [HZ]

ID	ID (hex)	Name	Type	Access	Size	Storage	Description
9406	0X24BE		UINT16	RW	1	FP	reserved
9407	0X24BF		UINT16	RW	1	FP	reserved
9500	0X251C	Position Correction Table	INT16	RW	64	FP	Position correction factors table parameters
9600	0X2580	CommutationTable	UINT16	RW	8	FP	UVW commutation table parameters
9700	0X25E4	Communications	UINT16	RW	6	FP	Serial Channel B parameters
9700	0X25E4	Flags	FLAGS	RW	1	FP	Serial Channel B options
9701	0X25E5	AxisId	UINT16	RW	1	FP	Serial Channel B axis identifier
9702	0X25E6	Baud	BAUD	RW	1	FP	Serial Channel B baud identifier
9703	0X25E7	CmdDelay	UINT16	RW	1	FP	Serial Channel B extra RX delay
9704	0X25E8	FrameDelay	UINT16	RW	1	FP	Serial Channel B extra TX delay
9705	0X25E9		UINT16	RW	1	FP	reserved

Factory Identification Register Table

Registers in this table are read-only and are stored in ROM flash memory.

ID	ID (hex)	Name	Type	Access	Size	Storage	Description
9900	0x26AC	Checksums	UINT16	RO	8	ROM	Flash code checksums [sectors A..H]
9908	0x26B4	Sectors	FLAGS	RO	1	ROM	Sectors requiring checksums
9909	0x26B5	Options	FLAGS	RO	1	ROM	Flags
9910	0x26B6	AppIdentifier	ENUM	RO	1	ROM	Application code identifier
9911	0x26B7	AppVersion	UINT16	RO	1	ROM	Application firmware version [0.01 units]
9912	0x26B8		UINT16	RO	2	ROM	reserved
9914	0x26BA	Startup	UINT32	RO	2	ROM	Address of app startup code
9916	0x26BC	BootIdentifier	ENUM	RO	1	ROM	Boot code identifier
9917	0x26BD	BootVersion	UINT16	RO	1	ROM	Boot code firmware version [0.01 units]
9918	0x26BE	DspType	ENUM	RO	1	ROM	DSP identifier
9919	0x26BF		UINT16	RO	13	ROM	reserved