Appendix A: Data Registers

Data registers can be dedicated to a specific purpose, optionally dedicated or continuously available for user data. They can be designated as Read Only or Read & Write. Data registers are 32 bits in length (long word) and numbered from 0 to 255 (Not all registers are implemented – varies by product and code revision). Many are designed to operate as two independent 16 bit data registers with each 16 bit word containing discrete data. Data is refreshed internally every servo cycle (120 microseconds). It can be sent or retrieved serially through a host controller or used internally by programs downloaded into the device.

Dedicated Data Registers (0-10)

This type of data register is dedicated to a specific the device function. the device uses this data extensively for many internal operations. Some data registers contain factory specific data that directly affects the servomotor operation. Modifications to this type of data may cause the servo to operate unexpectedly.

The table below provides information on dedicated data registers 0 through 10. These specific registers are used frequently when programming and operating the device.

Data Register	Type	Dedicated Data Register Description High word = (HW) <¦> Low word = (LW)
0	R/W*	Target Position; calculated position data from trajectory generator
1	R/W	Actual Position; current internal encoder position count
2	R/W	Last Index Position; encoder position count of the last internal index trigger
3 ‡	R	Internal Status Word (ISW) (HW) <¦> Reserved (LW)
4	R/W	Last Trigger Position; encoder position count when last stop condition was satisfied.
5	R/W	Delay Counter; clock ticks for the internal delay counter (1 tick = 120 usec). This is the register used by the Delay (DLY) command.
6 ‡	R	Max Position Error (HW) <¦> Current Position Error (LW)
7 ‡	R	Velocity 1; current vel. 1st filter (HW) <¦> Velocity 2; current vel. 2nd filter (LW)
8	R	Reserved
9 ‡	R	Reserved; (HW) <¦> Torque; current torque value (LW)
10	R/W	Accumulator; calc. results, reg. copy/save buffer, indirect addressing pointer

‡ Data register contains two independent 16 bit data words.

R = Read Only: R/W = Read and Write
R/W* = Read and Write (Write only using CLC command with Offset Target/Position operation)

User Data Registers and Optionally Dedicated Data Registers (11-199)

Data registers 11 through 40 (SilverDust Rev 06 11 through 199) are defined as user data registers by default. These read/write registers can be used by all the device

commands that are associated with user data registers. When the device is programmed to operate in an Input Mode, registers 12 through 18 become dedicated to the Input Mode operation. When Profile Move commands are implemented, registers 20 through 24 become dedicated to the Profile Move operation. Registers 11, 19, and 25 through 40 are always available for user data.

Data Register	Type	Default Use	Optional Dedicated Command Use	Dedicated Use Description
11	R/W	User Data		
12	R/W	User Data	*Input Mode	Input Source Data
13	R/W	User Data	*Input Mode	Input Offset
14	R/W	User Data	*Input Mode	Input Dead band
15	R/W	User Data	*Input Mode	Maximum Scale/Limit
16	R/W	User Data	*Input Mode	Maximum Output Scale
17	R/W	User Data	*Input Mode	Output Offset
18	R/W	User Data	*Input Mode	Output Rate of Change Limit
19	R/W	User Data		
20	R/W	User Data	*Profile Move	Absolute Position
21	R/W	User Data	*Profile Move	Acceleration
22	R/W	User Data	*Profile Move	Velocity
23	R/W	User Data	*Profile Move	Deceleration
24	R/W	User Data	*Profile Move	Offset (pos. from input stop)
25	R/W	User Data		
thru	R/W	User Data		
40 (198)	R/W	User Data		
199	R/W	User Data	*Default mapped I/O	Jump/stop on Mapped I/O

R/W = Read and Write

Registers for Optional Dedicated Command Use:

*Input Modes - Position Input Mode (PIM), Velocity Input Mode (VIM), and Torque Input Mode (TIM).

*Profile Move Commands - Profile Move (PMV), Profile Move Continuous (PMC), Profile Move Override (PMO), and Profile Move Exit (PME).

Dedicated Data Registers (200+)

The table below provides information on dedicated data registers 200+. These registers are utilized for advanced operations, complex programming, troubleshooting, and factory specific settings.

Data Reg	Type	Dedicated Data Register Description High word = (HW) <¦> Low word = (LW)
200	R/W	External Encoder Position; total count value from external encoder signals
201	R/W	External Index Position; count value of last external index trigger
202		Reserved
203	R	Reserved <¦> Internal Status Word 3 (IS3)
204	R	Target Acceleration (calculated internally by trajectory generator)
205	R	Target Velocity (calculated internally by trajectory generator)
206 ‡	R/W	Closed Loop Holding Torque (HW) <¦> Closed Loop Moving Torque (LW) Register based version of TQL command.
207 ‡	R/W	Open Loop Holding Torque (HW) <¦> Open Loop Moving Torque (LW) See 206 above
208 ‡	R/W	Error Limit Moving (HW) <¦> Error Limit Holding (LW) Register based version of ERL command.
209 ‡	R	Sense Mask; shaft rotation dir for +data (HW) <¦> IO Status Word (IOS) (LW) The direction Sense Mask (HW) defines the direction of shaft rotation (clockwise or counter clockwise) in relation to positive data parameters of position and velocity.
210 ‡	R	Program Buffer Size (HW) <¦> Program Buffer Start (LW)
211 ‡	R/W	Kill Motor Conditions (HW) <¦> Kill Motor States (LW) May be used to determine the cause of a triggered Kill Motor operation. These registers are written internally whenever the Kill Motor operation is activated. They may be overwritten to zero to make conditional testing of a triggering event easier.
212 ‡	R	Analog Input 1 (HW) <¦> Analog Input 2 (LW) Contains the filtered ADC readings for Analog Input1 (HW) and Analog Input 2 (LW). Allows these inputs to be monitored from the serial port without program involvement or stopping program operation.
213 ‡	R	Analog Input 3 (HW) <¦> Analog Input 4 (LW); raw data Same as 212, but for Analog Input 3 (HW) and Analog Input 4 (LW).
214 ‡	R	Driver Voltage (HW) <¦> Processor Temp (LW); raw data Contains the filtered ADC readings for the Main V+ drive voltage (HW) and the Controller/Processor temperature (LW). Temperature data is in a raw format and requires scaling for degree C output.
215 ‡	R	Process Volt (HW) <¦> Driver Temp (LC) Contains filtered ADC readings for HC processor voltage (HW) and HC driver temperature (LW). HC processors and drivers are used in the SilverNugget N3. Note that the calibration for the processor power is different than that for the driver power. The HC driver temperature does not follow the same scaling equation as for the processor temperature.

		The HC driver temperature can be approximated using Temp (centigrade) = (ADC value-2230)/228. Calculation is accurate to ± 3C between 5C and 100C.
216 ‡	R	Max Driver Voltage; raw data (HW) <¦> Drive Cal (counts/volt) (LW) Maximum driver voltage in ADC counts (HW) and Drive V+ calibrate data in ADC counts/volt (LW). The CAI command stored in the factory memory block initializes the data in this register
217 ‡	R/F	Max Driver Temp (HW) <¦> Processor Volt Cal (counts/volt) (LW) Maximum driver temperature in ADC counts for HC series (HW) and Processor V+ Calibration for the HC series, counts/volt (LW). Data is initialized by factory block.
218 ‡	R/W	Reserved
219 ‡	R	GroupID:Unit ID (HW) <¦> Reserved (LW)
220 ‡ thru 226 ‡	R/W	DIF I/O Filter Data for all seven I/O lines. DIF I/O Line Filter Constant; 0 = no filter (HW) <¦> DIF I/O Line Count (LW). [Note: If data is positive, I/O line is considered high, if negative, low. Counter will count up with high levels, and down with low levels, jumping to +/- filter count when it crosses zero count value (hysteresis).]
227	R/F	Reserved
228 ‡	R/W	Reserved
229 ‡	R/F	STEP4 <¦> Reserved STEP4 is set by the Initialization Wizard based on the encoder's counts/rev (CPR). STEP4 = Encoder CPR/50 To calculate encoder CPR within a program (QCP) read STEP4 from the upper word of reg 229 and multiply by 50.
230 ‡	R/F	Reserved
231	R/F	Reserved
232 ‡	R	Reserved
233	R/F	Reserved

The following registers are valid for the SilverDust only. They are not valid for the SilverNugget. The Rev column indicates the firmware revision the register first became available.

Data Reg	Type	Rev	Dedicated Data Register Description High word = (HW) <|> Low word = (LW)
234	R	04	Encoder CPR (HW) <|> Encoder Modulo Position (LW) CPR is the counts per revolution of internal encoder Modulo Position - rotary location: zero = index, count is modulo CPR; This is only valid after the index has been found at least one time.
235	R/F	04	Reserved
236	R	04	Internal Status Word 2 (IS2) (HW) <|> Internal Status Word 4 (IS4) (LW)
237	R/W	04	Reserved
238	R/W	04	Extended IO Word (XIO) de-bounced input (HW) <|> XIO output driver enable Debounced XIO input values <|> XIO open collector output transistors enabled Input bits 0 to 15 correspond to IO 101 to 116; 0 for low, 1 for 1.5v or higher Output bits 0 to 15 correspond to IO 101 to 116; 0 = transistor off, 1=transistor on
239	R/W	06	Reserved
240	R/W	06	Kill Motor Cond IS2 <|> XIO
241	R/W	06	Max Motor Temp in 1/16 ^{o}C <|> Motor Temp in 1/16 ^{o}C Available in SilverDust IG/IGB with sensor equipped motors. Max=0 disables the over temperature check
242	R/W	06	Reserved
243	R/W	08	Command Error <|> Trajectory State Stores the cause of a command error when one occurs. Save trajectory state as well incase the Command Error was as a result of changing registers used by the trajectory generator (such as the VIM command). Command Errors 1. Not enough writable registers, invalid register 2. Not able to perform requested motion in requested time 3. No motion was pre-calculated (or since cleared) 4. Command prohibited in current state 5. Invalid Program Buffer location 6. Unable to perform action 7. Invalid Selection: Mode, sub-command, I/O bit 8. Parameter out of range 9. Internal Error - bad calibration data 10. Invalid command number fetched from buffer 11. Stack Space Error - excess calls or returns 12. Bad EEPROM Access

			Trajectory States: 0x81 ;Not enough writable registers, invalid register 0x84 ;Command prohibited in current state • 0x88 ;Parameter out of range
244	R/W	08	Count Up Timer (ms) 32 bit free running up counting millisecond timer. Increments once per millisecond from power application. Automatically rolls over. Value is user writable.
245	R/W	08	Count Down Timer (ms) 32 bit down counting millisecond timer. Sets a flag bit is IS2 when it reaches 0. Stops counting when it reaches zero. Value is user writable.
246‡	R	25	CAN_ERR_REG|CAN_STATE CAN_ERR_REG same as CAN object 1001h Lower 8 bits of CAN_STATE are CAN NMT state, upper bits reserved. See CANopen User Manual
247‡	R/C	25	CANESR|CANGSR 2406 hardware registers 7106h:7107h – (see CANopen Manual for details). Hardware status registers for the CAN subsystem. They are Read/Clear or Read Only (see below). Read/Clear indicates that writing a 1 to the designated bit clears the bit. CANESR = CAN Error Status Register 1 = indicated error has occurred Bit 8 = Form Error Flag (RC) Bit 7 = Bit Error Flag (RC) Bit 6 = Stuck at Dominant (RC) Bit 5 = CRC Error (RC) Bit 4 = Stuff Error (RC) Bit 3 = Acknowledge Error (RC) Bit 2 = Bus-Off Status (0=normal operation) (RO) Bit 1 = Error Passive mode (0=normal) (RO) Bit 0 = Warning Status (1=at least one error counter reached 96) (RO) RC = read, write a 1 to clear, RO = Read only, writes to bit are ignored CANGSR = CAN Global Status Register Bit 5 = SMA = Suspend Mode Acknowledge (0=normal) Bit 4 = CCE = Change Configuration Enable (0=normal) Bit 3 = PDA = Power Down Mode Ack. (0=normal) Bit 2 = Reserved Bit 1 = RM = Receive mode = CAN module is receiving a frame Bit 0 = TM = Transmit mode = CAN module is transmitting a frame.

248	R/W	25	Thread 2 Accumulator Thread 2 local copy of Register 10 (Allows access to Thread 2 register 10 via Serial/CAN
249‡	R/W	25	Cmd Err LOADADD | Reserved Copy of EEPROM load address prior to Command Error Recovery command (so Command Error Recovery routine knows the original Command Error)
250‡	R/W	25	Thread 2 LOADADD | Reserved Thread 2 EEPROM load address.
251‡	R	25	Thread 2 Program Buffer Start | Size Start of Program Buffer for Thread 2 Size of Program Buffer for Thread 2
252‡	R	25	Reserved

‡ Data register contains two independent 16 bit data words.

R = Read Only: R/W = Read and Write; R/F = Read/Factory Writable (SilverDust Rev 06)

Note: Use caution when writing to 200 level data registers as some retain factory specific data. Changing the data in specific registers may cause operation problems with the device. Some registers labeled R/F may be user Read Only; these will eventually be set to user Read Only.

Appendix B: Conversion Data

Inertia - To convert from A to B, multiply by the constant in table

A \ B	oz-in^2	oz-in-s^2	lb-in^2	lb-in-s^2	N-m-s^2	g-cm^2	kg-m^2	kgf-m-s^2
oz-in^2	1	$2.59*10^{-3}$	$6.25*10^{-2}$	$1.6188*10^{-4}$	$1.8289*10^{-5}$	182.9	$1.8289*10^{-5}$	$1.86*10^{-6}$
oz-in-s^2	386.09	1	24.131	$6.25*10^{-2}$	$7.0612*10^{-3}$	$7.0612*10^{4}$	$7.0612*10^{-3}$	$7.2*10^{-4}$
lb-in^2	16	$4.1441*10^{-2}$	1	$2.5901*10^{-3}$	$2.9262*10^{-4}$	2926.2	$2.9262*10^{-4}$	$2.9839*10^{-5}$
lb-in-s^2	6177	16	386.09	1	0.11298	$1.1298*10^{6}$	0.11298	$1.1521*10^{-2}$
N-m-s^2	$5.4678*10^{4}$	141.62	3417.4	8.8512	1	$1*10^{7}$	1	0.10197
g-cm^2	$5.4678*10^{-3}$	$1.4162*10^{-5}$	$3.4174*10^{-4}$	$8.8512*10^{-7}$	$1*10^{-7}$	1	$1*10^{-7}$	$1.0197*10^{-8}$
kg-m^2	$5.4678*10^{4}$	141.62	3417.4	8.8512	1	$1*10^{7}$	1	0.10197
kgf-m-s^2	$5.3621*10^{5}$	1388.8	$3.3513*10^{4}$	86.801	9.8067	$9.8067*10^{7}$	9.8067	1

Power - To convert from A to B, multiply by the constant in table

A \ B	Watt	HP	N-m-RPS	oz-in -RPM	ft-lb-RPM	ft-lb/sec	N-m/sec
Watt	1	$1.341*10^{-3}$	0.1592	1352	7.042	0.7375	1
HP	745.7	1	118.7	$1.0083*10^{6}$	5251.4	549.93	745.7
N-m-RPS	6.283	$8.426*10^{-3}$	1	8496	44.25	4.634	6.283
oz-in -RPM	$7.396*10^{-4}$	$9.918*10^{-7}$	$1.177*10^{-4}$	1	$5.208*10^{-3}$	$5.454*10^{-4}$	$7.396*10^{-4}$
ft-lb-RPM	0.142	$1.904*10^{-4}$	$2.26*10^{-2}$	192	1	0.1047	0.142
ft-lb/sec	1.356	$1.818*10^{-3}$	0.2158	1833	9.549	1	1.356
N-m/sec	1	$1.341*10^{-3}$	0.1592	1352	7.0423	0.7375	1

Torque - To convert from A to B, multiply by the constant in table

A \ B	ft-lb	in-lb	oz-in	N-m	kgf-m	kgf-cm	gf-cm
ft-lb	1	12	192	1.3558	0.13825	13.825	$1.3825*10^{4}$
in-lb	$8.333*10^{-2}$	1	16	0.113	$1.1521*10^{-2}$	1.1521	1152.1
oz-in	$5.2083*10^{-3}$	$6.25*10^{-2}$	1	$7.0615*10^{-3}$	$7.2006*10^{-4}$	$7.2006*10^{-2}$	72.006
N-m	0.73757	8.8509	141.61	1	0.10197	10.197	$1.0197*10^{4}$
kgf-m	7.2331	86.798	1388.8	9.8067	1	100	$1*10^{5}$
kgf-cm	$7.2331*10^{-2}$	0.86798	13.888	$9.8067*10^{-2}$	$1*10^{-2}$	1	1000
gf-cm	$7.2331*10^{-5}$	$8.6798*10^{-4}$	$1.3888*10^{-2}$	$9.8067*10^{-5}$	$1*10^{-5}$	$1*10^{-3}$	1

Additional Conversion Data

Length 1 inch = 0.0254 meters

Mass 1 ounce = 0.02835 kilograms

Velocity 1 revolution/second (rps) = 60 revolutions/minute (rpm)

Temperature °F = [°C • (9/5)] + 32

Appendix C: Command Error Codes

Cmd Err	Hex	Thread#	Description
1	0x01	1	Invalid register or not enough writable registers
2	0x02	1	Not able to perform requested motion in requested time
3	0x03	1	No motion was pre-calculated
4	0x04	1	Command prohibited in current state
5	0x05	1	Invalid command buffer location
6	0x06	1	Unable to perform action because already active
7	0x07	1	Invalid Selection: Mode , subcommand, I/O bit
8	0x08	1	Parameter out of range
9	0x09	1	Internal error or bad calibration data
10	0x0A	1	Invalid command number found in command buffer
11	0x0B	1	Stack space problem - underflow/overflow
12	0x0C	1	Bad EEPROM access
13	0x0D	1	CAN Dictionary location not writable
14	0x0E	1	CAN Dictionary location not readable
15	0x0F	1	No such Data Dictionary entry
16	0x10	1	CAN Dictionary internal consistency problem (please report to factory)
17	0x11	1	Thread 2 only command trying to execute in thread 1
18	0x12	1	CAN Dictionary - Write not successful due to system state
19	0x13	1	CAN Dictionary - byte count invalid
65	0x41	2	Invalid register or not enough writable registers
66	0x42	2	Not able to perform requested motion in requested time
67	0x43	2	No motion was pre-calculated
68	0x44	2	Command prohibited in current state
69	0x45	2	Invalid command buffer location
70	0x46	2	Unable to perform action because already active
71	0x47	2	Invalid selection: Mode-Subcommand-I/OBit
72	0x48	2	Parameter out of range
73	0x49	2	Internal error or bad calibration data
74	0x4A	2	Invalid command number found in command buffer
75	0x4B	2	Stack space problem - underflow/overflow
76	0x4C	2	Bad EEPROM access
77	0x4D	2	CAN Dictionary location not writable
78	0x4E	2	CAN Dictionary location not readable
79	0x4F	2	No such Data Dictionary entry
80	0x50	2	CAN Dictionary internal consistency problem (please report to factory)
81	0x51	2	Thread 2 only command trying to execute in thread 1
82	0x52	1	CAN Dictionary - Write not successful due to system state
83	0x53	1	CAN Dictionary - byte count invalid
129	0x81	0	Invalid register or not enough writable registers
132	0x84	0	Command prohibited in current state
136	0x88	0	Parameter out of range