ANIMATICS®

ANIMATICS Corporation

SmartMotor™ Class 5 Product Overview

www.animatics.de

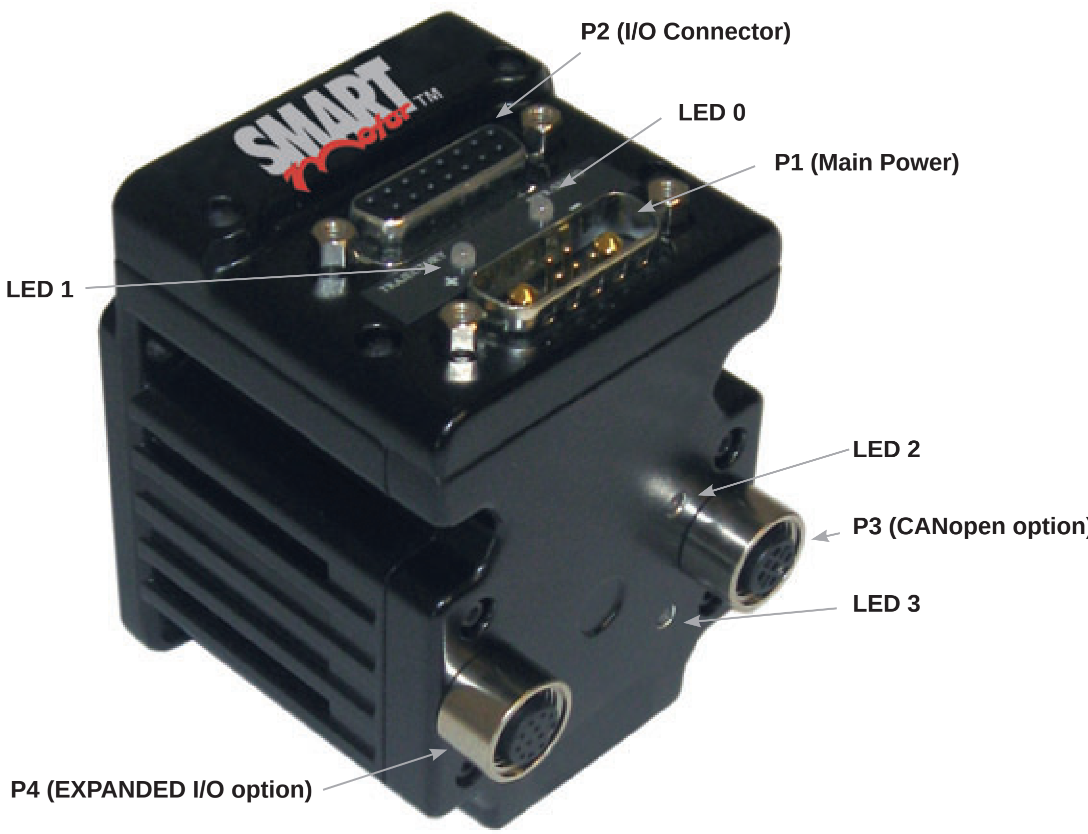
Animatics GmbH | In der Neuen Welt 8/G5-6 | D-87700 Memmingen | Tel. +49 8331 9278355 | Fax +49 8331 9278356 | anfrage@animatics.com

Contents

Class 5 refers to a new series of products consisting of the existing Class 4 motor line and hardware interface with a greatly improved processing capability. All Class 5 motors have the same cables and interconnectivity as existing Class 4 motors. They have the same I/O and serial port capabilities allowing the same cables, power supplies and interface adapters to be used. This enables users to maintain familiarity while gaining net performance and programmability advancements.

Here is a short list of features:

- Faster Processor, 7 to 10 Times Faster than Existing Class 4 Motors
- Faster RS232 / RS485 Communications Speeds, up to 115.2KBaud
- COMBITRONIC™ Transparent Communications over CAN Bus (optional)
- Multi-Port Simultaneous Communications, RS-232/RS-485/CAN Bus
- Enhanced Trap Mode and Sine Mode Commutation
- Higher Frequency PID update rate: down to 62 μsec sample period
- Expanded Math Function Capability
- Floating Point Math
- Modulo Encoder Count Capability
- 8 Level Priority Stacked User Definable Interrupts
- 4 User Definable Independent Timers
- DE/Dt: Rate of Change of Following Error Limit
- Increased I/O Interrupt Assignments
- Software Programmable Limits can be set to trigger interrupts w/o fault
- Enhanced Parameter and Function Based Syntax
- Increased System Status Bit Registers for Advanced Diagnostics
- Optional On-board Expanded I/O : 10 Channels 24VDC Isolated Assignable as Inputs or Outputs



COMBITRONIC™ Communications:

Animatics Corporation has introduced a significant advancement in Integrated Motor Technology. The term "Distributed Control" is used when programmable drives or intelligent Integrated Servos work in conjunction with a Central Control or PLC. This leaves the PLC less burdened with full machine control. In concept, this is good, but still leaves inter-communications of machine subsystems with much to be desired.

Class 5 SmartMotors™ have opened the door to communications between separate motors allowing multiple masters to coexist on the same machine with full deterministic arbitration and no loss of data packets. Through a combination of standard CANopen and extended Combitronic™ protocol, the result is the ability to write a program in one motor that transparently accesses and modifies data and parameters in any other motor on the network. All communications are achieved over the Combitronic™ CAN interface. No user setup is required other than setting of the node address and ensuring matched baud rates. The SmartMotor™ handles everything else. As a result, the user may have conditional command structures in one motor that poll and compare parameters from multiple other motors within one line of code.

Suppose Motor 1 requires data from two other motors.

Example code:

```
IF (PA:2>PA:3)&(VA:4<2000)
     PRINT("Motor 2 has passed Motor 3 and Motor 4 has slowed down")
ENDIF
```

In the above example, Motor 1 is comparing the actual positions (PA) between Motor 2 and Motor 3 while checking the actual speed (VA) of Motor 4. The processor handles all communications via Combitronic™ protocol transparent to the user. As a result, the user is free to fully control multiple motors across the network from a single program within a single motor.

At any time, an additional motor can be added into the network without causing issues with the network. Even further, the extended 29Bit addressing allows non-Animatics CAN Bus products to coexist on the same CAN network without causing communications errors.

Multi-Port Simultaneous Communications, RS-232/RS-485/CAN Bus

Class 5 SmartMotors allow 3 port communications at the same time.

Com(0): RS-232 Primary Port
Com(1): RS-485 Secondary Port
Com(2): CANopen

Class 5 CANopen complies with CIA402 specifications.

Additionally, the motors meet the CANopen IP spec (Interpolation Protocol)
When run in contouring mode via CANopen, an X-Y 2-axis system can be updated every 800 microseconds.
This equates to Extremely fast high resolution paths for advanced CNC applications.
When in Interpolation Protocol Mode, a 2-axis path may have Position points along that path to within 800 microsecond time intervals.

Let's suppose the following:
SM23165DT SmartMotor running CANopen has a 4000 count resolution connected to a 2mm ball screw.
A 2-axis system running 150mm/sec can hold to a coordinated path resolution of 0.12millimeters or .004 inches.
At a more moderate machining speed of 20mm/sec, the system would hold to a tolerance of 0.016mm or .00006 inch resolution. This capability opens the door to EDM precision machining and laser cutting CNC applications.
Animatics Corporation now offers SMNC CNC software package to directly run the SmartMotors over CAN providing a plug and play solution ready for customers.

Enhanced Trap Mode and Sine Mode Commutation:

The motors can be operated with encoder-based commutation that allows for a more precise alignment and association of rotor to stator magnetic phases. The result is a smooth, quiet rotation with very low cogging. As a result, much slower commanded speeds may be achieved with little speed fluctuation.

Higher Frequency PID Update Rate:
User selectable PID update rate defaults to 125 microseconds. Optionally it may be decreased or increased. The faster 62.5 microsecond update rate allows for smoother high speed operation and faster accel/decel correction under varying load conditions.

Expanded Math Function Capability:

Class 5 SmartMotors includes:

- Added boolean operators such as Exclusive OR and Modulo,
- Trigonometry functions, SIN, COS, TAN, ASIN, ACOS, ATAN
- Absolute Value
- IEEE-754 Single Precision Floats
- Commutative and Associative math operations are allowed with up to 128 characters on the right side of an equal sign.

DE/Dt: Rate of Change of Following Error Limit
DEL provides the safest fast-means to fault a motor on sudden changes in load or detection of human interference.

The purpose of this Limit is to act as a look ahead on following error. Instead of just triggering on a raw following error of how far behind in a move the motor may be, the processor is looking at how fast that following error changes.

Example:
If DE/Dt equals commanded velocity, then the motor just hit a hard stop.
Normally, the motor would have to continue applying torque until the normal following error is exceeded. However, If DEL (DE/Dt limit) is set to target velocity (VT), then the controller would error out immediately upon hitting a hard stop without any wind-up whatsoever.

DEL={value} sets the error limit. DEL defaults to 2147483647.
The processor looks at the absolute value of the rate of change of following error. Therefore you can place a limit on both acceleration and deceleration.

Refers to the dynamic rate of change of following error. This results in an instant release of energy and safer operation and less chance of damage to equipment or injury to machine operators. Under normal servo control following error limits, if the load collides against an object, the motor will not fault until the following error limit is reached. As a result, current and torque applied will increase until that condition is met. By adding an additional derivative limit on following error, the servo will fault out within microseconds of contact with the object.

Increased I/O Interrupt Assignment
ANY I/O point can be assigned to trigger interrupts on either rising or falling edge.
The interrupts in turn, can call their respective subroutines to run user code as needed.
I/O triggered interrupts calls respond within 2 PID cycles or typically less than 1 millisecond

Software Programmable Limits Can Be Set to Trigger Interrupts w/o Fault

The software travel limits now have a mode command allowing them to be passive or active travel limit triggers.
When in Active Mode, detection of travel limits will result in an over travel fault.
When in passive mode, detection of an over travel condition will not cause a motor fault. As a result, they can be used for high speed position-based triggers. This allows them to be used as programmable Cam switches.

```
SLP=10000      'Set Positive Travel limit
SLN=-10000     'Set Negative Travel Limt
RSLN     or     x=SLN
RSLP     or     x=SLP
SLE            'enable soft limits
SLD            'disable soft limits
SLM(0)         'Make a soft limit only trigger the flag, but not cause a fault.
SLM(1)         'Make a soft limit trigger the flag and cause a fault.(default mode)
```

Enhanced Parameter and Function Based Syntax

Most commands have been upgraded to support passing parameters to them via variables or equations.

Example:

GOSUB(*value*) Call a specific subroutine

```
GOSUB(x)  'call subroutine "x"
```

GOTO(*value*) Jump to a specific location

```
GOTO(x)   'jump to lable C{x}
```

OS(*value*) Turn on a specific output

```
OS(x)     'Set output "x" (Turn On)
```

OR(*value*) Turn off a specific output

```
OR(x)     'Reset Output "x" (Turn Off)
```

RIN(*value*) Report a specific variable

```
RIN(x)    'Report State of Input "x"
```

Increased System Status Bit Registers for Advanced Diagnostics

There are now multiple System Status bit registers.

Each are in a parameter format allowing user access and reporting via variables and expressions.

Example:

```
x=B(0,1)     'assign status word 0 , bit 1 to the variable x.
x=B(y,z)     'assign status word y , bit z to the variable x.
```

Optional On-board Expanded I/O : 10 channels 24VDC Isolated Assignable as Inputs or Outputs

The expanded on-board I/O consists of 10 channels of I/O points. Each may be programmable as inputs or outputs.

When assigned as outputs, they are sourcing (PNP) and short circuit protected.

6 channels can handle 350mAmps.
4 channels can handle 150mAmps

(See end of document for specific connector pinouts)

This section of the document covers an overview of the Motion Commands.

Note: All Modes of operation require a "G" command to begin Motion

1.0 Standard Modes of Operation

```
MP              'Initiate Position Mode
MV              'Initiate Velocity Mode
MT              'Initiate Torque Mode (open Loop)
```

1.1 Electronic Gear Modes

Mode-Step-Ration: Following Step and Direction:

The following commands will set up inputs zero (A) and one (B) as Step and Direction input

```
MS0             'Initiate and zero counter, but do not follow

MSR             'Calculate Mode Step Ratio and prepare to follow
```

Mode-Follow-Ration: Following and External Standard TTL Quadrature Encoder:

The following commands will set up inputs zero (A) and one (B) as Phase A and B Quadrature Inputs

```
MF0             'Initiate and zero counter, but do not follow
MFR             'Calculate Mode Follow Ratio and prepare to follow
```

Setting a ratio for either Step Ratio (MSR) and Mode Follow Ratio(MFR) modes listed above:

```
MFMUL={expression}   'Assign Incoming counts Multiplier
MFDIV={expression}   'Assign Incoming counts Divisor

RMFMUL          'Report Multiplier
RMFDIV          'Report Divisor

x=MFMUL         'Assign Multiplier to the variable "x"
x=MFDIV         'Assign Divisor to the variable "x"

MFA(value)      'Accel over value master distance. Default is zero (off)


MFD(value)      'Decel over value master distance. Default is zero (off)
```

For MFD, MFA, x <= 0 turns it off so that ratio is reached instantly.

```
MFX             'Slow down based on MFD.

MFSLEW(value) 'Stay at slew for value distance, then decel.
```

Range is: 0 - (2^31 - 1). Value x= -1 turns it off. Default is that this feature is off
Any combination of MFA, MFD, MFMUL, MFDIV is valid.
G starts the ramp up according to above. Further G commands are ignored until motor reaches a stop again.

1.2 Starting and Stopping All Modes

```
G        'Go, Initiates all buffered modes of operation
X        'Decelerate to a Stop at present deceleration rate
S        'Stop at fastest rate possible
OFF      'Turn the amplifier OFF
```

1.3 Reporting the MODE

```
RMODE    'Report Mode of operation
x=MODE   'Assign Mode of Operation to the variable "x"
```

Returned values for RMODE or x=MODE:

-3	STEP/Direction	(MSR command)
-2	Mode Follow	(MFR command)
0	Reserved	
1	Position Mode	(MP command, default on startup)
2	Reserved	
3	Velocity Mode	(MV command)
4	TORQUE	(MT command)
5	Reserved	
6	Reserved	
7	Interpolation Mode	(Profiling via CAN Open only)

2.0 Fault Actions

```
FSA(fault type #, fault mode#)
```

Set it and forget it. If you want to return to defaults, set it again.

Fault type: the type of fault to set a mode on:

0 – All types of fault.
1 – Hardware travel limits.
2+ – reserved.

Fault mode:

0 - default action (MTB)
1 - Servo off
2 - X command (Decelerate to a stop)

Example

```
FSA(0,0) 'Default, sets all types of fault to result in MTB
```

2.1 Brake Control Commands

```
BRKSRV                    'Brake Servo, Engage the brake when the drive
                           is not active
```

Fault actions taken are as outlined in FSA command

```
BRKTRJ                    'Brake Trajectory
```

Typically, any time Trajectory is not in progress, the brake will engage automatically.
Fault actions taken are as outlined in FSA command

```
BRKENG   'Manually Engage the brake
BRKRLS   'Manually Release the brake
```

3.0 Move Generation Setup:

3.1 Target Position Assignments:

```
PT=expression            'Set the absolute target position
PRT=expression           'Set the relative target position
```

Reports:

```
RPA                      'Report present Actual Position
x=PA                     'Assign present Actual Position to "x"

RPC                      'Report present Commanded Position
x=PC                     'Assign present Commanded Position to "x"

RPT                      'Report present target Position
x=PC                     'Assign present target Position to "x"

RPRT                     'Report present Relative target Position
x=PRT                    'Assign present Relative target Position to "x"
```

Origins and Offsets:

```
O=expression             'Set Origin, Set present position to some value
OSH(value)               'Origin Shift of Position Counter on the fly
```

3.2 Target Velocity and Torque Assignments:

```
VT=expression            'Set the velocity target for a move
```

Note: Used in both MV and MP modes of operation.

```
T=expression             'Set commanded Torque while in MT mode
```

Note: Range: -30000 to 30000, +/-0 to 100% PWM open loop

Reports:

```
RVA                      'Report Actual Velocity
x=VA                     'Assign Actual Velocity to "x"

RVC                      'Report Commanded Velocity
x=VC                     'Assign Commanded Velocity to "x"

RVT                      'Report Target Velocity
x=VT                     'Assign Target Velocity to "x"

RT                       'Report Commanded Torque
x=T                      'Assign Commanded Torque to "x"
```

3.3 Target Acceleration and Declaration Assignments:

```
AT=expression  'Set the Acceleration Target for a move
DT=expression  'Set the Deceleration Target for a move
ADT=expression 'Set the Accel/Decel at once for a move
```

Note: Used in both MV and MP modes of operation.

Reports:

```
RAA            'Report Actual Acceleration
x=AA           'Assign Actual Acceleration to "x"

RAC            'Report Commanded Acceleration
x=AC           'Assign Commanded Acceleration to "x"

RAT            'Report Target Acceleration
x=AT           'Assign Target Acceleration to "x"

RDT            'Report Target Deceleration
x=DT           'Assign Target Deceleration to "x"
```

4.0 Following Error Limits
Position error:

Assignments:

```
EL=expression  'Set Maximum allowable Following Error limit
DEL=expression 'Set Maximum allowable Derivative Error limit
```

Reports:

```
REL            'Report Commanded Following Error Limit
x=EL           'Assign Commanded Following Error Limit to "x"

REA            'Report Actual Following Error
x=EA           'Assign Actual Following Error to "x"
RDEL           'Report Commanded Derivative Error Limit
x=DEL          'Assign Commanded Derivative Error Limit to "x"
```

5.0 Temperature Limit

Setting:

```
TH=expression  'Set Maximum allowable Thermal limit (Degrees C)
```

Reporting:

```
RTH            'Report Maximum allowable Thermal limit
x=TH           'Assign Maximum allowable Thermal limit to "x"
```

6.0 Software Programmable Over Travel Limits

Setting:

```
SLP=expression 'Set Positive Over Travel Limit Position
SLN=expression 'Set Negative Over Travel Limit Position
```

Reporting:

```
RSLP           'Report Positive Over Travel Limit Position
x=SLP          'Assign Positive Over Travel Limit Position

RSLN           'Report Negative Over Travel Limit Position
x=SLN          'Assign Negative Over Travel Limit Position
```

Control of:

```
SLE            'Enable Software Travel Limits
SLD            'Disable Software Travel Limits
SLM(0)         'Limit only trigger the flag, but not cause a fault.
SLM(1)         'Limit trigger the flag and cause a fault. (default mode)
```

7.0 Modulo Position Counter Function

```
PML=expression 'Sets the Position Modulo Limit Wrap value.
```
Also resets the modulo counter to 0.
Range: 1000 to 2000000000

```
RPML           'Report Position Modulo Limit

RPMA           'Report the current modulo counter.
```

Always in range 0 to (PML-1) inclusive.

```
PMT=expression 'Sets Position Modulo Target
```

Note: Shaft takes shortest path to target meaning I may more backwards through zero

```
RPMT           'Report the most recent setting of PMT.
```

8.0 Phase Offset Function

```
PON=expression 'Sets Phase Offset Numerator.
```
Default=0 Range: -32768 to 32767

```
POD=expression 'Sets Phase Offset Denominator.
```
Default=0 Range: -32768 to 32767

Note: Phase Offset function is enabled when both values become non-zero

```
RPON           'Report Phase Offset Numerator

RPOD           'Report Phase Offset Denominator

RPA            'Reports shaft position + phase value

RCTR(0)        'Reports what the main encoder is actually doing to motor base

RCTR(1)        'Reports actual external encoder phase value
```

9.0 Commutation Invert Function

```
MINV(0)        'Default, no invert, shaft rotates normally

MINV(1)        'Invert Commutation, shaft rotates opposite direction
```

This section of the document covers both local and expanded I/O.

Local I/O for Class 5 consists of Ports A thru G (I/O 0 thru I/O 6)

Expanded I/O refers to the additional 10 Channels (16 thru 25) of optional I/O on a 12 pin 12mm connector

Please See the end of this document for full electrical data on I/O

- is the IO Bit Number
m - is the mask value of which bits are effected

The following is an I/O map for Class 5 motors:

Discrete I/O General Commands:

1.0 Input Commands: # below refers to individual Input channels where 0 thru 6 for local I/O and 16 thru 25 for expanded I/O

Configuring I/O to be General Purpose Inputs:

EIGN(#)	Assign a single I/O point as General Use Input
EIGN(W,0,m)	Assign a masked Word-sized set of local I/O as General Use inputs at once.
EIGN(W,0,12)	Assign inputs 2 and 3 as general use inputs at once (disabling Over-Travel Limits)
EIGN(W,0)	Assign ALL local I/O as general use inputs

Note: "W" defines it as a Word (16 bits), "0" after the "W" means the first 16 I/O, in this case, I/O 0 thru 6.
IF EIGN(W,1) was issued, All expanded I/O 16 thru 25 would be set to General use inputs.

Assigning I/O Status to a variable. The following applies regardless if the I/O using used as an input or output.
Additionally, they can be used in IF or WHILE conditions as well

x=IN(#)	Assign the state of a specific I/O to a variable (x in this case)
x=IN(W,0)	Assign the state of the first Word of local I/O to the variable x
x=IN(W,0)&m	Assign the state of the first Word of local I/O as masked by "m" to the variable x
Examples:	
x=IN(W,0)&15	Assign the state of the the lowest 4 I/O of the first Word of local I/O as masked by 15 to the variable x
x=IN(W,1)&31	Assign the state of the the lowest 5 I/O of the first Word of expanded I/O as masked by 31 to the variable x

Reporting I/O Status:

RIN(#)	Report the state of a I/O
RIN(W,0)	Report the first Word of local I/O
RIN(W,1)	Report the first 16 channels of expanded I/O

1.1 Over Travel Limit Configuration:

EILN	Set Port C (I/O-2) as Negative Over Travel Limit
EILP	Set Port D (I/O-3) as Positive Over Travel Limit

1.2 Index Capture Commands:

EIRE	Set I/O 6 to capture External-Encoder's current value. Internal encoder will be captured via its own index pulse.
EIRI	Set I/O 6 to capture Internal-Encoder's current value. Internal encoder ignores its own index pulse.

1.3 Configure a Given Input to Cause a G Command to be Issued When Active:

EISM(6)	Issue (G) when local input 6 goes low. (Presently only channel 6 is supported)

Animatics Corporation
tel: 408.748.8721 • fax: 408.748.8725 • www.animatics.com

Analog Input Commands:

2.0 Local I/O: "#" must be between 0 and 6 for Ports A thru G

x=INA(A,#)	Raw Analog Reading: 10-Bit resolution spanned over signed 16-Bit range.
RINA(A,#)	Will report 0 to 32765 for 0 to 5VDC where # is for local channels from 0 to 6
x=INA(V1,#)	Scaled Voltage Reading in millivolts directly 3456 would be 3.456VoltsDC
RINA(V1,#)	Will report 0 to 5000 for 0 to 5VDC where # is for local channels from 0 to 6

2.1 Expanded I/O: "#" must be between 16 and 25 for Connector Pins 1-10

INA(A,#) Raw analog reading: ~9-Bit Resolution where 0-32765 is referenced to 0-42VDC

Note:

Maximum input voltage is 32VDC, so maximum value would be ~24964

INA(V,#) Scaled Voltage Reading in millivolts directly 3456 would be 3.456VoltsDC

3.0 Output Commands:

OS(#)	SET a Single Output to logic level 1	(5VDC for local I/O, 24VDC for expanded I/O)
OS(W, #, m)	SET Multiple Outputs at once, Bit masked by "m"	
OS(W,0,12)	Set I/O 2 and 3 as outputs (Disables travel limits while setting them high at the same time)	
OS(W,0)	Set ALL local I/O (0-6) to 5VDC	
OS(W,1)	Set ALL expanded I/O (16-25), to 24VDC	
OR(#)	RESET a Single Output to zero	(Sinking for local I/O, High Impedance for Expanded I/O)
OR(W, #, m)	RESET Multiple Outputs as once, Word masked by "m"	
OUT(#)=	If argument to the right of “=” is true then set I/O on(1), else off(0)	

Example:

OUT(3)=a&b will set Output 3 to logic 1 if both a and b are not zero, else it will be set to a zero.

OUT(s, #)= Set the I/O group to a value to the right of the “=” .

Example:

OUT(W,0)=0 will set all local I/O 0 thru 6 to zero volts.

OUT(W,1)=1 will set all expanded I/O 16 thru 25 to 24VDC.

3.1 Output Condition and Fault Status Reports:

Note: Fault Condition Commands apply to Expanded I/O only, not local I/O

X=OC(#)	Individual Output status, bit is 1 if Output is being driven. (likewise ROC(W,#))
X=OC(W,#)	Block Output status, bit is 1 if Output is being driven. (likewise ROC(W,#))
X=OF(#)	returns the present fault state for that IO, where: 0 = no Fault , 1 = over current, 2 = shorted
X=OF(W, #)	returns bit mask of present Faulted IO points. Where # is the 16Bit word #. If value is greater than zero then I/O fault status flag (status word 3) is set.
X=OF(L, #)	returns bit mask Fault Latched for IO points, Where # is the 16Bit word #. A read of a 16bit word will attempt to clear the latch.

3.2 Outputs Configuration List:

EOBK(#) Configure a given output to control an external Brake

Note:

Class 5 I/O Cross Reference

ANIMATICS®

I/O Configuration Commands

I/O	Connector -Pin#	Report Command	Assign I/O as Output		
			Set High	Reset Low	Set by Value
Local I/O on DB-15 Connector					
0	P2-1	EIGN(0)	OS(0)	OR(0)	OUT(0)=expression
1	P2-2	EIGN(1)	OS(1)	OR(1)	OUT(1)=expression
2	P2-3	EIGN(2)	OS(2)	OR(2)	OUT(2)=expression
3	P2-4	EIGN(3)	OS(3)	OR(3)	OUT(3)=expression
4	P2-5	EIGN(4)	OS(4)	OR(4)	OUT(4)=expression
5	P2-6	EIGN(5)	OS(5)	OR(5)	OUT(5)=expression
6	P2-7	EIGN(6)	OS(6)	OR(6)	OUT(6)=expression
7	N/A	N/A	OS(7)	OR(7)	OUT(7)=expression
Expanded I/O on M12 12-Pin Connector					
16	P4-1	EIGN(16)	OS(16)	OR(16)	OUT(16)=expression
17	P4-2	EIGN(17)	OS(17)	OR(17)	OUT(17)=expression
18	P4-3	EIGN(18)	OS(18)	OR(18)	OUT(18)=expression
19	P4-4	EIGN(19)	OS(19)	OR(19)	OUT(19)=expression
20	P4-5	EIGN(20)	OS(20)	OR(20)	OUT(20)=expression
21	P4-6	EIGN(21)	OS(21)	OR(21)	OUT(21)=expression
22	P4-7	EIGN(22)	OS(22)	OR(22)	OUT(22)=expression
23	P4-8	EIGN(23)	OS(23)	OR(23)	OUT(23)=expression
24	P4-9	EIGN(24)	OS(24)	OR(24)	OUT(24)=expression
25	P4-10	EIGN(25)	OS(25)	OR(25)	OUT(25)=expression

I/O Report and Assign Status Commands

I/O	Connector -Pin#	Report Command	Assign to a Variable	Report by Status Word	Assign by Status Word
Local I/O on DB-15 Connector					
0	P2-1	RIN(0)	x=IN(0)	RB(16,0)	x=B(16,0)
1	P2-2	RIN(1)	x=IN(1)	RB(16,1)	x=B(16,1)
2	P2-3	RIN(2)	x=IN(2)	RB(16,2)	x=B(16,2)
3	P2-4	RIN(3)	x=IN(3)	RB(16,3)	x=B(16,3)
4	P2-5	RIN(4)	x=IN(4)	RB(16,4)	x=B(16,4)
5	P2-6	RIN(5)	x=IN(5)	RB(16,5)	x=B(16,5)
6	P2-7	RIN(6)	x=IN(6)	RB(16,6)	x=B(16,6)
7	N/A	RIN(7)	x=IN(7)	RB(16,7)	x=B(16,7)
Expanded I/O on M12 12-Pin Connector					
16	P4-1	RIN(16)	x=IN(16)	RB(17,16)	x=B(17,16)
17	P4-2	RIN(17)	x=IN(17)	RB(17,17)	x=B(17,17)
18	P4-3	RIN(18)	x=IN(18)	RB(17,18)	x=B(17,18)
19	P4-4	RIN(19)	x=IN(19)	RB(17,19)	x=B(17,19)
20	P4-5	RIN(20)	x=IN(20)	RB(17,20)	x=B(17,20)
21	P4-6	RIN(21)	x=IN(21)	RB(17,21)	x=B(17,21)
22	P4-7	RIN(22)	x=IN(22)	RB(17,22)	x=B(17,22)
23	P4-8	RIN(23)	x=IN(23)	RB(17,23)	x=B(17,23)
24	P4-9	RIN(24)	x=IN(24)	RB(17,24)	x=B(17,24)
25	P4-10	RIN(25)	x=IN(25)	RB(17,25)	x=B(17,25)

I/O Analog Input Commands

I/O	Connector -Pin#	Report as Raw Analog	Assign as Raw Analog	Report Analog in milliVolts	Assign Analog in milliVolts
Local I/O on DB-15 Connector					
0	P2-1	RINA(A,0)	x=INA(A,0)	RINA(V1,0)	x=INA(V1,0)
1	P2-2	RINA(A,1)	x=INA(A,1)	RINA(V1,1)	x=INA(V1,1)
2	P2-3	RINA(A,2)	x=INA(A,2)	RINA(V1,2)	x=INA(V1,2)
3	P2-4	RINA(A,3)	x=INA(A,3)	RINA(V1,3)	x=INA(V1,3)
4	P2-5	RINA(A,4)	x=INA(A,4)	RINA(V1,4)	x=INA(V1,4)
5	P2-6	RINA(A,5)	x=INA(A,5)	RINA(V1,5)	x=INA(V1,5)
6	P2-7	RINA(A,6)	x=INA(A,6)	RINA(V1,6)	x=INA(V1,6)
7	N/A	RINA(A,7)	x=INA(A,7)	RINA(V1,7)	x=INA(V1,7)
Expanded I/O on M12 12-Pin Connector					
16	P4-1	RINA(A,16)	x=INA(A,16)	RINA(V1,16)	x=INA(V1,16)
17	P4-2	RINA(A,17)	x=INA(A,17)	RINA(V1,17)	x=INA(V1,17)
18	P4-3	RINA(A,18)	x=INA(A,18)	RINA(V1,18)	x=INA(V1,18)
19	P4-4	RINA(A,19)	x=INA(A,19)	RINA(V1,19)	x=INA(V1,19)
20	P4-5	RINA(A,20)	x=INA(A,20)	RINA(V1,20)	x=INA(V1,20)
21	P4-6	RINA(A,21)	x=INA(A,21)	RINA(V1,21)	x=INA(V1,21)
22	P4-7	RINA(A,22)	x=INA(A,22)	RINA(V1,22)	x=INA(V1,22)
23	P4-8	RINA(A,23)	x=INA(A,23)	RINA(V1,23)	x=INA(V1,23)
24	P4-9	RINA(A,24)	x=INA(A,24)	RINA(V1,24)	x=INA(V1,24)
25	P4-10	RINA(A,25)	x=INA(A,25)	RINA(V1,25)	x=INA(V1,25)

Class 5 Encoder Related Commands

Class 5 has 2 independent encoder counters with trap (capture) capabilities allowing high speed capture of either internal or external encoder counters triggered via either Internal index pulse or external Input 6 respectively.
In the following commands:

If x=0 It refers to the Motor's Internal Encoder.
If x=1 It refers to any external encoder tied to the encoder A and B Inputs of the motor

Commands:
Bi (x) Rising Edge Index Report, Indicates Position has been captured
Bj (x) Falling Edge Index Report, Indicates Position has been captured

For Both Bi(x) and Bj(x):

a. They will be Set to 1 when a index register update has occurred
b. Remains true until a Zi(0) command (Zero Index)

Example:

```
IF Bi(0)==1
    PRINT("Internal Index has been recently captured",#13)
ENDIF
```

Bx(x) Real Time state, Detected=1 and OFF=0 of the trigger (trap) input (Input 6)

```
IF Bx(0)==1
    PRINT("The Read Head sees the Internal Index Mark",#13)
ENDIF
```

CTR(x) Present value of the A quad B counter Note: CTR(0) is the same as PA

```
x=0
RCTR(x)     'Report Internal Encoder Position
x=1
RCTR(x)     'Report External Encoder Position
```

I(x) Present value of the Index/Trap register will remain unchanged once Bi(0)=1 until Ai(x) has been executed.

```
RI(0) 'Report where the rising Edge of the Internal Index was detected
```

J(x) Present value of the Index/Trap register will remain unchanged once Bj(0)=1 until Aj(x) has been executed.

```
RJ(0) 'Report where the falling Edge of the Internal Index was detected
```

Ai(x) Re-Arms the Index/Trap register for Rising Edge respectively.

Once the command is issued, the processor is set to capture and record encoder position the next time the rising edge of index mark is detected

Aj(x) Re-Arms the Index/Trap register for Falling Edge respectively.

Once the command is issued, the processor is set to capture and record encoder position the next time the falling edge of the index mark is detected

Aij(x) Re-Arm a capture where Rising Edge captured then Falling Edge captured.
(The Falling edge of a High Pulse will trigger a position capture)

Aji(x) Re-Arm a capture where Falling Edge captured then Rising Edge captured.
(The Rising Edge of a Low Pulse will trigger a position capture)

Serial Ports

There are two serial ports in the Class 5 motor:

Com0: an RS-232 port that is commonly used for communication to a host.

This port is open and ready to communicate in command mode upon motor boot-up. The default Baud rate is 9600, but this can be modified in the EE. This must be done with care as an mistake in this setup can render a motor unable to communicate.

Com1: an RS-485 port that is available through the 15-pin expansion port. By default, this port is off and must be enabled either by a user program OCHN command, or an OCHN through another active command manager such as com0, or CANOpen encapsulated commands.

OCHN(TYPE, CHANNEL, PARITY, BITRATE, STOPBITS, DATABITS, SPEC, TIMEOUT)

```
OCHN(RS2,0,N,9600,1,8,C,1000) 'Default Boot-up Condition for Primary RS232 Port
OCHN(RS4,1,N,9600,1,8,C,1000) 'Open Secondary RS485 Port (via Inputs 4 and 5)
```

Note: (TIMEOUT) is an option parameter. IF No terminating character (space or carriage return) is received within the default time-out of 1000 milliseconds, the associated Communications Error bit will be set.

Setting Motor Serial communications address. This address applies to both RS-232 and RS-485 Ports.

ADDR=***expression*** ***expression*** may be from 1-to 100 The default Serial address is 0 (global address)

Note: For RS-232 Communications where more than one motor is in a serial daisy chain, the ECHO command must be used to insure all data received in one motor will be "ECHO's" out to the next motor down. The default state on boot-up is ECHO_OFF meaning the motor will NOT echo any incoming data to downstream motors.

BAUD(x)=y This allows either COM0 or COM1 to be changed. x is the channel: 0 or 1. y is the baud rate to use.

Baud(x)=	Actual Baud	Percent error
2400	2400.960384	0.040016006
4800	4807.692308	0.16025641
9600	9615.384615	0.16025641
19200	19230.76923	0.16025641
38400	38461.53846	0.16025641
57600	58823.52941	2.124183007
115200	117647.0588	2.124183007

PRINT command examples:

```
PRINT("Hello World",#13)          '#13 is Carriage Return
g=123                             'Set variabel to 123
a=2+128                           '2 will be motor address, 128 is offset for extended
                                   ASCII addressing
PRINT(#a,"GOSUB(",g,")",#13)      'Tell motor 2 to run subroutine 123
x=33                              'Print all Printable ASCII characters
WHILE x<=126        PRINT(#x,#13)       x=x+1 LOOP
```

CAN Bus

The only user settable options are the baud rate and node ID. When changed, the values are written into EEPROM, and the motor must be re-booted for them to take effect. The commands DO NOT need to be re-issued again in a downloaded program or otherwise.

CADDR=***expression*** ***expression*** may be from 1-to 127 The default node address is 63

CBAUD=***expression*** Valid Baud rates are: 1000000, 800000, 500000, 250000, 125000, 100000, 50000, 20000,10000
The default baud rate is 125000

RCBAUD or x=CBAUD reports the can baud rate as one of the values show above.

RCADDR or x=CADDR reports the node ID.

Note: ALL motor syntax and code that uses CAN Bus from one motor to the next is via Animatics Combitronic™ protocol. Communications are fully automatic. All timing, responses and arbitration is taken care of in firmware. The only prerequisite is that each motor have a separate node address and proper shunt resistors are on the line.

1.0 Math Equations/Formula Math:

Syntax allows for multiple communicative and associative operations on the right hand side of the equals sign: a=(b+c/d)*(e-f)/g

1.1 Logical Operators: (These return a value of 0 if false and a value of 1 if true.)

<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
==	Equal to
!=	Not equal to

To check the range of a variable, use the following example: d = (-10 < a) & (a < 10)

It is not valid to write this as a<b<c because the logical operator that occurs first will return a 0 or 1, and will be compared to the remaining value which will not give the expected result. The result 'd' will be 0 if a was outside the range, and 1 if it was within the range.

1.2 Math Operators:

^	Power limited to 4th power and below, on integers only.
*	Multiply
/	Divide – Be careful to not div/0, result will be 0.
+	Add
-	Subtract
%	Modulo (remainder) division. Be careful to not div/0, result will be 0.
|	Bitwise inclusive or
&	Bitwise and
!|	Bitwise exclusive or

1.3 Math Equations Follow a Strict Order of Operations.

Order of operations is (first-highest to last-lowest). Items on the same line described here are in the same priority. Items of the same priority group are evaluated in the order they are parsed. Do not assume left-to-right evaluation.

( )	Parenthesis
^	Power operator
-	Negation
* / %	Multiply, divide, and modulo.
+ -	Add, subtract
< > <= >=	Less than, greater than, less than or equals, greater than or equals
== !=	Equals, not equals
&	Bitwise and
!|	Bitwise xor
|	Bitwise inclusive or

2.0 Math Functions:

Integer Only Math Functions:

SQRT(***x***)	Square Root
ABS(***x***)	Absolute Value

Floating Point Math Functions:

FSQRT(***xf***)	Square Root	
FABS(***xf***)	Absolute Value	
SIN(***xfd***)	Sine	
COS(***xfd***)	Cosine	
TAN(***xfd***)	Tangent	
ACOS(***xf***)	Arc Cosine	where xf is valid from 1.0 to -1.0
ASIN(***xf***)	Arc Sine	where xf is valid from 1.0 to -1.0
ATAN(***xf***)	Arc Tangent	

x - is reduced/truncated to a whole integer
xf - is promoted to a float
xfd- is a value in degrees always promoted to float

Example: `af[0]=SIN(90.0)   'where 90 degrees is 1.570796 radians, will result in 1.0000 in af[0]`

Animatics Corporation
tel: 408.748.8721 • fax: 408.748.8725 • www.animatics.com

3.0 Floating Point Support

In any operation, if the input is integer then the result remains an integer; a result is promoted to a float once the operation includes a float.

Example:

```
af[0]=(a + b)/2 +3.0
```

Note: In the above if a=8 and b=1, then 7.0 would be the result in af[0].

Example:

```
af[0]=(a*1.0 + b)/2 +3.0
```

Note: In the above if a=8 and b=1, then 7.5 would be the result in af[0].

Bring in the bits of a single precision IEEE-754 float literally.

3.1 LFS: Load Float Single Precision

This function converts a long into a floating-point user variable. This is not a simple float value of the integer value. That can be done with af[0]=a. LFS interprets the 32-bits of the integer as the bits of an IEEE-754 single-precision floating-point number. It is promoted to the double-precision stored in the Class 5.

This is the layout for a single float IEEE-754 number:

Sign Bit, s (Bit 31)	Exponent, e (Bit 30-23)							Mantisse, m (Bit 22-0)																						
msb																															lsb
s	e	e	e	e	e	e	e	e	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m	m

Note: If all exponent bits are set to 1, it is considdered+/- infinity and is Not a Number"
The LFS command will treat these cases as overflow and report a value of 0, with a command error.

IEEE-754 does not specify a byte-order when serializing the 32-bits into bytes. The Class 5 memory map is little-endian when loading al[x] registers via the ab[x] overlay; therefore, the LFS command will see these as little-endian. This means that if you are loading the ab[x] array, put the least significant byte of the IEEE-754 value into the lowest ab[x] register of the registers that you are using. You must choose ab[x] registers that are consecutive and aligned with a single al[x] register. i.e. the lowest one must be an index where the index modulo 4 is 0.

```
RLFS(integer)   'Load Float Single Precision
af[0]=LFS(integer)
a=LFS(integer)  'Throws away fractional part

' Load the bits to represent 7.76 as a 32-bit IEEE-754 float:
ab[12]=-20    'EC hex
ab[13]=81     '51 hex
ab[14]=-8     'F8 hex
ab[15]=64     '40 hex

af[0]=LFS(al[3])  ' al[3] maps to ab[12]-ab[15]
Raf[0]        7.760000229
```

3.2 Dump the Literal Bits of an IEEE-754 float

This function is the inverse of LFS. It dumps the bits of an af register into an integer register in order to transmit the literal 32-bits of the floating-point value. Note that the reported value is single-precision IEEE-754. The reverse process of the example above can be used to transmit bytes.

```
af[0]=1.5
al[0]=DFS(af[0])
Ral[0]         1069547520    Report values to terminal screen
Rab[0]         0             Least significant part of the significand.
Rab[1]         0
Rab[2]         -64           Exponent and part of the significand.
Rab[3]         63            Sign and exponent
```

4.0 Convert ASCII to Float

```
RATOF(index)   ' Directly convert and report as float, beginning at array index
af[0]=ATOF(index)
a=ATOF(index) ' Just gets integer portion.
```

Index is an integer such as a, al[5], or 24. It points to a location in the ab[index] array. This location contains ASCII bytes representing a number, which may contain digits, a period, and a leading minus sign. The number string is terminated by any other character.

The most significant decimal digit will be at the lowest address.

Example:

```
ab[20]=45    ' -
ab[21]=49    ' 1
ab[22]=46    ' .
ab[23]=51    ' 3
ab[24]=51    ' 3
ab[25]=0     ' (end - can be anything other than period or digit.)

af[0]=ATOF(20)
Raf[0]          -1.330000000
```

5.0 Convert ASCII Hex to Integer

This function behaves similar to ATOF, in that is uses ASCII bytes found in the ab[0] registers and converts them to a value. The string must always be 8 bytes long. If you have fewer hex digits, you must pad with leading ASCII zeros. The result is an integer. This function is documented here with floating-point utilities, because it may be used in combination with LFS to read hex-encoded float values.

ASCII characters 0-9, a-f, and A-F are accepted in any combination. The conversion is NOT case sensitive. There is no string terminator, it is always assumed to be 8 bytes. There is no leading ‘0x’ string.

The most significant hexadecimal digit will be at the lowest address.

```
RHEX(index)    'Report Hex
a=HEX(index)   'Assign Hex

ab[0]=49       '1'
ab[1]=50       '2'
ab[2]=51       '3'
ab[3]=52       '4'
ab[4]=102      'f'
ab[5]=65       'A'
ab[6]=97       'a'
ab[7]=70       'F'
b=HEX(0)
Rb              305461935
```

Non-volatile Memory EEPROM Command:

EPTR=*expression* — Sets up the pointer for a subsequent VLD or VST command. User-accessible, read-write areas of EE are 0 to 31999 inclusive. Users may read anything from 0 to 31999 inclusive. The upper area from 32000-32767 inclusive contains the Animatics data block information, which is documented separately.

REPTR and variable=EPTR — Shows where the next VST or VLD will take place.

VST (variable, quantity) — Write to EE based on the current state of EPTR. Variable may be a-zzz, ab[n], aw[n], al[n]. Quantity is the count of items to write starting at the variable you supply, if you supply a 32-bit variable and you supply a quantity of 3, then three 32-bit variables are written. If the variable is 16-bit (aw[n]), then the count is in number of 16-bit words. If the variable is a byte (ab[n]), then the count is in bytes. Float values are not yet supported.

VLD (variable, quantity) — Read from EE based on the current state of EPTR. Variable may be a-zzz, ab[n], aw[n], al[n]. Float values are not yet supported.

```
'Record Shaft Position 16 times, once a second
VT=100000          'Set Velocity Target
ADT=1000           'Set Accel/Decel Target
MV                 'Set to Velocity Mode
G                  'Go
i=1
WHILE i<16
      al[i]=PA     'assign Actual Positon to array variable
      i=-i+1
      WAIT=1000    'Wait 1 second (1000 milliseconds)
LOOP
X                  'Decelerate to a stop

EPTR=100           'Set EEPROM pointer
VST(al[1],16)      'Store 16 consecutive variables from above

EPTR=100           'Set EEPROM pointer
VLD(al[1],16)      'Load (recall) 16 consecutive values in to array
```

```
'Typical Start-up Program
EIGN(2)       'Disable Positive Over Travel Limit
EIGN(3)       'Disable Negative Over Travel Limit
ZS            'Clear Travel Limit faults (In case they were not grounded
               on boot-up)
O=0           'Set Origin to zero
MP            'Set to Position Mode
VT=100000     'Set Target Velocity
AT=1000       'Set Target Acceleration
DT=500        'Set Target Deceleration
PT=8000       'Set Target Absolute Commanded Position
G             'Go (Start moving)
TWAIT         'Wait until the move is complete
END     'End the program. All programs must contain at least one END
```

```
'Wait on Input to Go Low
WHILE IN(0)==1 LOOP  'Watch Port A Until It goes low

'Set up Interrupt 4 to trigger on Input Zero to go low (and call C10):
ITR(4,16,0,0,10) '(Int 4, Status Word 16, Bit Zero, Goes to Zero, calls
C10)
EITR(4)           'Enable the specific Interrupt
ITRE              'Enable the Global Interrupt scanner
PAUSE
'Associated subroutine that will be called on Interrupt
C10
'Perform some function when Input zero goes low
RETURNI
```

```
'Wait on Input to Go High
WHILE IN(0)==0 LOOP  'Watch Port A until it goes high

'Set up Interrupt 5 to trigger on Input Zero to go high (and call C11):
ITR(5,16,0,1,11) '(Int 5, Status Word 16, Bit Zero, Goes to 1, calls C11)
EITR(5)           'Enable the specific Interrupt
ITRE              'Enable the Global Interrupt scanner
PAUSE
'Associated subroutine that will be called on Interrupt
C11
'Perform some function when Input zero goes high
RETURNI
```

```
'Check Input While Moving and Perform Some Function
PT=123456                     'Set Target Position
G                             'Go, (Start Moving)
WHILE Bt                      'While Busy Trajectory Bit is on
    IF IN(0)==1               'If Port A goes high
              GOSUB12         'Run Subroutine 12
    ENDIF
LOOP                          'Continue Checking
C12
'Do Something here if needed
RETURN
'Note: An Interrupt could have been used as shown in previous examples.
'Then only a TWAIT would be needed.
```

```
'Check For Errors After a Move
PT=123456                     'Set Target position
G                             'Go, (Start Moving)
TWAIT                         'Hold While Busy Trajectory Bit is on
IF Bo==1                      'If Motor Off Bit comes is on
    GOSUB1                    'Run Subroutine 1    (see below)
ENDIF
```

```
 'Example Error Handler Routine (see above)

C1  'Start of Subroutine 1
    IF Be               'Checking for error status bits
        PRINT("Position Error",#13)
    ENDIF
    IF Bh
        PRINT("Over Temp Error",#13)
    ENDIF
    IF Ba(0)
        PRINT("Over Current Error",#13)
    ENDIF
RETURN  'Return to line just below GOSUB1
```

```
'-----------------------------------------------------------
'Typical Start up Program w/Fault handling
EIGN(2)      'Disable Positive Over Travel Limit
EIGN(3)      'Disable Negative Over Travel Limit
ZS           'Clear Travel Limit faults
O=0          'Set Origin to zero

'Set up a Fault Handler Interrupt (Int Zero) to trigger if Drive Ready
goes low (And call C0):
ITR(0,0,0,0,0)   '(Int 0, Status Word 0, Bit Zero, Goes to Zero, calls
C0)
EITR(0)      'Enable the specific Interrupt
ITRE         'Enable the Global Interrupt scanner
             'Setting up variables for Home routine
vv=100000    'Home Velocity
aa=100       'Home Accel
tt=3000      'Home Torque
hh=1000      'Home Offset Distance (from hard stop)
rr=-1        'Home Direction GOSUB5        'Call Home-to-Hard-Stop Routine
PAUSE        'Prevent Code Execution from ENDing.
END
'-----------------------------------------------------------------------
C0
'Place Any and All Fault code handlers here
RETURNI  'Note: RETURNI with I on the end meaning return from interrupt
'-----------------------------------------------------------------------
'Home Routine (Home to Hard Stop)
C5
IF q==-1 RETURN ENDIF
q=0
VT=vv*rr                 'Set Home Velocity
ADT=aa                   'Set Home Accel
MV                       'Set to Velocity Mode
ZS                       'Clear any prior errors
T=tt*rr                  'Preset Torque Values
G                        'Begin move towards hard stop
WHILE ABS(EA)<ee LOOP    'Loop, While Position Error within limit
MTB                      'Mode Torque Break to stop
MT                       'Switch to Torque Mode in case bounce off of
                          hard stop
WAIT=50                  'Wait 50mseconds
O=hh*rr                  'Set Origin to home offset
MP PT=0 G TWAIT          'Set Motor to Zero
q=123
RETURN
'----------------------------------------------------------------------
```

```
'Pulse Output On at a Given Position
x=20000                    'Set Position to trigger on during a move
PT=50000                   'Set commanded position to move to
G                          'Start move
WHILE Bt                   'While moving
    IF PA>x                'If present position exceeds "x"
        OUT(1)=1           'Set Port B to 5VDC
        WAIT=10            'Wait 10 milliseconds
        OUT(1)=0           'Set Port B to Zero Volts
        BREAK              'Break Out of Loop
    ENDIF
LOOP
TWAIT                      'wait for move to complete
```

```
'Change Speed During Move on Input Signal
O=0
PT=50000                   'Set commanded position to move to
G                          'Start move
WHILE Bt                   'While moving
IF IN(0)==0                'If Port A is grounded
        VT=VT+200000 'Increase speed
        G
        BREAK              'Break Out of Loop
ENDIF
LOOP
TWAIT                      'wait for move to complete
```

```
'Turn Output on If Real Time Position Error Exceeds user Amount

O=0
PT=50000                   'Set commanded position to move to
e=50                       'Set Desired Trigger Point
EL=1000                    'Set Max allowable Position Error to 1000
                           'Set Port B as an Output
OUT(1)=1                   'Set it to 5VDC
G                          'Start move
WHILE Bt                   'While moving
 IF EA>e                   'If Real Time Position Error exceeds e
    OUT(1)=0               'Set Port B to Zero
 ENDIF
LOOP
```

ANIMATICS®

Class 5 Motion Command Quick Reference

		Absolute Position	Relative Position	Velocity	Accel and Decel Together	Accel	Decel	Following Error	DE/Dt Derivitive Error Limit	Over Speed Limit
Report	Actual	RPA	RPRA	RVA	RAA	RAA		REA	RDEA	
Report	End Target	RPT	RPRT	RVT	RAT	RAT	RDT	REL	RDEL	RVL
Report	Commanded	RPC	RPRC	RVC	RAC	RAC				
Assign	End Target	PT=	PRT=	VT=	ADT=	AT=	DT=			
Assign	Command	N/A	N/A	N/A	N/A	N/A	N/A	EL=	DEL=	VL=

CLASS 4 PLS2		CLASS 5		
Mode/Motion Commands		**Mode/Motion Commands**		
F=2	'Invert Commutation	MINV(1)	'Invert motor commutation	
Not assigning F=2 will keep default Direction		MINV(0)	Default Motor Commutation Direction	
MP	'Mode Position (Default)	MP	'Mode Position (Default)	
V=x	'Set Target Velocity	VT=x	'Set Target Velocity	
A=x	'Set Target Accel./Decel.	ADT=x	'Set Target Accel./Decel.	
		AT=x	'Set Target Accel only	DT follows AT until DT is assigned
		DT=x	'Set Target Decel only	
P=x	'Set Target Absolute Position	PT=x	'Set Target Absolute Position	
D=x	'Set Target Relative Position	PRT=x	'Set Target Relative Position	
X	Decel to stop at rate of "A"	X	'Decel to stop at rate of "AT"	Note: If AT not assigned , Fast stop at "S" rate
S	'Stop at fast fixed rate	S	'Stop at fast fixed rate	
MV	'Mode Velocity	MV	'Mode Velocity	
Reporting Motion and Position Values		**Reporting Motion & Position Values**		**8000**
O=0	'Set present position to zero	O=0	'Set present position to zero	
		OSH(x)	'Shift present position by "x" counts, may be used on the fly	
RP	'Report Actual Position	RPA	'Report Actual Position	
x=P Rx	'Report Target Position	RPT	'Report Target Position	
		RPC	'Report Commanded Position	
RV	'Report Commanded Velocity	RVC	'Report Commanded Velocity	
x=V Rx	'Report Target Velocity	RVT	'Report Target Velocity	
RA	Report Target Acceleration	RAT	'Report Target Acceleration	
E=x	'Set Max Following Error Limit	EL=x	'Set Max Following Error Limit	
RE	'Report Error Limit	REL	'Report Error Limit	
		DEL=x	Set Derivative Error Limit	
		RDEL	Report Derivative Error Limit	
		RDEA	Report Actual Derivative Error	

Animatics Corporation
tel: 408.748.8721 • fax: 408.748.8725 • www.animatics.com

ANIMATICS®

CLASS 4 PLS2		CLASS 5		
Open Loop Mode		**Open Loop Mode**		
MT	'Mode Torque (No G required)	MT	'Mode Torque (G will re-enter)	
T=x	'Assign Torque Value, 0 to +/-1023	T=x	'Assign Torque Value, 0 to +/-32765 Note: Torque Mode requires a "G" command in Class 5	
Slave/Following Modes:		**Slave/Following Modes:**		
Step and Direction Input Modes:		**Step and Direction Input Modes:**		
MS0	'Set Ports A and B as Step/Dir	MS0	'Set Ports A and B as Step/Dir	
	'and zero out CTR (ext. counter)		'and zero out CTR(1) (ext. counter)	
'Note: Also zero's out CTR register		'Note: Also zero's out CTR register		
MS	Mode Step at 1:1 (No G required)	No Equivalent in Class 5		
MFMUL=x	Set Incoming pulse multiplier	MFMUL=x	Set Incoming pulse multiplier	RMFMUL Report MFMUL
MFDIV=x	Set Incoming Pulse Divisor	MFDIV=x	Set Incoming Pulse Divisor	RMFDIV Report MFDIV
MSR	'Calculate Mode Step Ratio	MSR	'Calculate Mode Step Ratio	
G	'MSR takes effect, Drive Activates	G	'MSR takes effect, Drive Activates	
RCTR	Report External Counter Value	RCTR(1)	Report External Counter Value	
Encoder Following Mode		**Encoder Following Mode**		
MF0	'Set Ports A and B as Encoder Input	MF0	'Set Ports A and B as Encoder Input	
'Note: Also zero's out CTR register		'Note: Also zero's out CTR register		
MF1	Mode Follow 1:4 (No G required)			
MF1	Mode Follow 1:2 (No G required)			
MF4	Mode Follow 1:1 (No G required)			
MFMUL=x	Set Incoming pulse multiplier	MFMUL=x	Set Incoming pulse multiplier	
MFDIV=x	Set Incoming Pulse Divisor	MFDIV=x	Set Incoming Pulse Divisor	
		MFA(x)	accel over x master distance. Default is zero (off)	
		MFD(x)	decel over x master distance. Default is zero (off)	
		MFX	Slow down based on MFD.	
		MFSLEW(x)	Stay at slew for “x” distance, then decel.	
MFR	'Calculate Mode Follow Ratio	MFR	'Calculate Mode Follow Ratio	
G	'MFR Takes effect, Drive Activates	G	'MFR Takes effect, Drive Activates	
RCTR	Report External Counter Value	RCTR(1)	Report External Counter Value	

CLASS 4 PLS2		CLASS 5	
Program Flow		**Program Flow**	
RUN	'Start Program Execution	RUN	'Start Program Execution
Note: No changes to any values in Flash		Note: No changes to any values in Flash	
RUN?	'Wait at this point for RUN command	RUN?	'Wait at this point for RUN command
GOTO#	'GOTO C#	GOTO(x)	'GOTO C# where x can be any number 0-999
GOSUB#	Call subroutine #	GOSUB(x)	Call subroutine "x"
IF {condition is true}		IF {condition is true}	
ELSEIF {condition is true}		ELSEIF {condition is true}	
ELSE {if no other condition above is true}		ELSE {if no other condition above is true}	
ENDIF	End if IF code structure	ENDIF	End if IF code structure
Note: only two operands allowed		Class 5 allows complex comparisons:	
Example:		**Example:**	
IF a<b	If a is less than b	IF (a<b)&(c<d)	If a is less than b and c is less than d
		IF ABS(EA)>x	If the absolute value of position error is greater than x
WHILE {condition is true}		WHILE {condition is true}	
LOOP	Execute any code between WHILE and LOOP when true	LOOP	Execute any code between WHILE and LOOP when true
Example:		**Example:**	
WHILE a<b	While a is less than b	WHILE a<b	While a is less the b careful not to add this too
BREAK	break out of WHILE loop	BREAK	break out of WHILE loop
Example:		**Example:**	
WHILE a>b IF c>d BREAK ENDIF LOOP		WHILE a>b IF c>d BREAK ENDIF LOOP	

CLASS 4 PLS2		CLASS 5	
Math Functions		**Math Functions**	
+	Addition	**+**	Addition
-	Subtraction	**-**	Subtraction
*****	Multiplication	*****	Multiplication
/	Division	**/**	Division
		%	Modulo (returns remainder of division)
		ABS(***x*** **)**	Absolute Value
		^	Raise to Power
		Integer Only:	
		SQRT(***x*** **)**	Square Root
		ABS(***x*** **)**	Absolute Value
		Floating Point:	
		FSQRT(***xf*** **)**	Square Root
		FABS(***xf*** **)**	Absolute Root
		SIN(***xfd*** **)**	Sine
		COS(***xfd*** **)**	Cosine
		TAN(***xfd*** **)**	Tangent
		ACOS(***xfd*** **)**	Arc Cosine
		ASIN(***xfd*** **)**	Arc Sine
		ATAN(***xfd*** **)**	Arc Tangent
		See notes in Variables Section for more information	

NOTE:

x - is reduced/truncated to a whole integer

xf - is promoted to a float

xfd - is a value in degrees always promoted to float

CLASS 4 PLS2		CLASS 5	
Comparison Operators		**Comparison Operators (may be used as Logical Assignment)**	
>	Greater Than	>	Greater Than
<	Less Than	<	Less Than
==	Equal To	==	Equal To
!=	Not Equal To	!=	Not Equal To
>	Greater Than	>	Greater Than
>=	Greater than or Equal To	>=	Greater than or Equal To
<=	Less then or Equal To	<=	Less then or Equal To
|	Logical OR	|	Logical OR
&	Logical AND	&	Logical AND
		!|	Exclusive OR

Class 5 Command Comparison

CLASS 4 PLS2		CLASS 5	
Variables	**32 bit signed integer**	**Variables**	**32 bit signed integer**
a thru z		a thru z	
aa thru zz		aa thru zz	
aaa thru zzz		aaa thru zzz	
Array Access from aa to zzz		Array Variables (Dedicated)	
ab[0] thru ab[200]	8 Bit Byte array access	ab[0] thru ab[200]	8 Bit Byte array access
aw[0] thru aw[100]	16 Bit Word array access	aw[0] thru aw[100]	16 Bit Word array access
al[0] thru al[50]	32 Bit Long array access	al[0] thru al[50]	32 Bit Long array access
Note: All array variables overlap aa thru zzz in Class 4 motors!!!			
		Floating Point Array Variables:	
		af[0] thru af[7]	
		The floating point array variables meet IEEE-754 specification. So they are a true floating point where the location of the decimal point can vary with the exponent from very tiny i.e. approximately 1x10^-300 to very large approximately 1x10^+300. Class 5 can do basic operations at 64-bit precision: +, -, *, /. BUT, the trig functions are only being calculated with 32-bit precision	

CLASS 4 PLS2		CLASS 5	
All I/O commands refer to local I/O unless otherwise noted			
Configuring Ports as general Inputs:		**Configuring Ports as general Inputs:**	
UAI	'set User port A as Input	EIGN(0)	'set User port 0 as Input
UBI	'set User port B as Input	EIGN(1)	'set User port 1 as Input
UCI	'set User port C as Input	EIGN(2)	'set User port 2 as Input
UDI	'set User port D as Input	EIGN(3)	'set User port 3 as Input
UEI	'set User port E as Input	EIGN(4)	'set User port 4 as Input
UFI	'set User port F as Input	EIGN(5)	'set User port 5 as Input
UGI	'set User port G as Input	EIGN(6)	'set User port 6 as Input
Configuring Ports as general Outputs:		**Configuring and Setting Ports as general Outputs:**	
UAO	'set User port A as Output	OUT(0)=	'set/reset Port 0 Output state
UBO	'set User port B as Output	OUT(1)=	'set/reset Port 1 Output state
UCO	'set User port C as Output	OUT(2)=	'set/reset Port 2 Output state
UDO	'set User port D as Output	OUT(3)=	'set/reset Port 3 Output state
UEO	'set User port E as Output	OUT(4)=	'set/reset Port 4 Output state
UFO	'set User port F as Output	OUT(5)=	'set/reset Port 5 Output state
UGO	'set User port G as Output	OUT(6)=	'set/reset Port 6 Output state

CLASS 4 PLS2		CLASS 5	
Resetting Output Logic States:		**Configuring and Resetting Output Logic States:**	
UA=0	'set User port A to 0VDC	OR(0)	'set User port 0 to 0 VDC
UB=0	'set User port B to 0VDC	OR(1)	'set User port 1 to 0 VDC
UC=0	'set User port C to 0VDC	OR(2)	'set User port 2 to 0 VDC
UD=0	'set User port D to 0VDC	OR(3)	'set User port 3 to 0 VDC
UE=0	'set User port E to 0VDC	OR(4)	'set User port 4 to 0 VDC
UF=0	'set User port F to 0VDC	OR(5)	'set User port 5 to 0 VDC
UG=0	'set User port G to 0VDC	OR(6)	'set User port 6 to 0 VDC
Setting Output Logic States:		**Configuring and Setting Output Logic States:**	
UA=1	'set User port A to 5VDC	OS(0)	'set User port 0 to 5 VDC
UB=1	'set User port B to 5VDC	OS(1)	'set User port 1 to 5 VDC
UC=1	'set User port C to 5VDC	OS(2)	'set User port 2 to 5 VDC
UD=1	'set User port D to 5VDC	OS(3)	'set User port 3 to 5 VDC
UE=1	'set User port E to 5VDC	OS(4)	'set User port 4 to 5 VDC
UF=1	'set User port F to 5VDC	OS(5)	'set User port 5 to 5 VDC
UG=1	'set User port G to 5VDC	OS(6)	'set User port 6 to 5 VDC

CLASS 4 PLS2		CLASS 5	
Set ports A and B as counter inputs (just counting):		**Set ports A and B as counter inputs (just counting):**	
MS0	'(A) Step and (B) Direction Input	MS0	'(A) Step and (B) Direction Input
MF0	'(A) and (B) as Enc. Input	MF0	'(A) and (B) as Enc. Input
CLASS 4 PLS2		**CLASS 5**	
Limit Switch Control (Ports C and D) Default State		**Limit Switch Control (Ports C and D) Default State**	
UCP	'Set Port C as Pos. Travel Limit	EILP	'Set Port C as Pos. Travel Limit
UDM	'Set Port D as Neg. Travel Limit	EILN	'Set Port D as Neg. Travel Limit
CLASS 4 PLS2		**CLASS 5**	
Go (G) command assignment		**Go (G) command assignment**	
UG	'Set Port G to "G" command	EISM(6)	'Set Port G to "G" command
CLASS 4 PLS2		**CLASS 5**	
Brake Control		**Brake Control**	
BRKI	'Default, operates internal brake	'EOBK(0-6 are valid),	
BRKC	'Redirect control to Port C	EOBK(2)	'Redirect control to Port C
BRKG	'Redirect control to Port D	EOBK(6)	'Redirect control to Port D

CLASS 4 PLS2		CLASS 5	
Report I/O Signal Levels		**Report I/O Signal Levels**	
Digital returns 0 or 1 for 0 or 5 VDC		Digital returns 0 or 1 for 0 or 24 VDC	
RUA	'Report Port A Digital	RIN(0)	'Report Port 0 Digital
RUB	'Report Port B Digital	RIN(1)	'Report Port 1 Digital
RUC	'Report Port C Digital	RIN(2)	'Report Port 2 Digital
RUD	'Report Port D Digital	RIN(3)	'Report Port 3 Digital
RUE	'Report Port E Digital	RIN(4)	'Report Port 4 Digital
RUF	'Report Port F Digital	RIN(5)	'Report Port 5 Digital
RUG	'Report Port G Digital	RIN(6)	'Report Port 6 Digital
Group Reporting of all Local I/O		Group Reporting of all Local I/O	
RU	'Report all 7 I/O (7 bit)	RIN(B,0)	'Report all 10 I/O (10 bit)
CLASS 4 PLS2		**CLASS 5**	
Analog Input Commands		**Analog Input Commands**	
Analog returns 0-1023 for 0 to 5 VDC		Analog returns 0-32736 for 0 to 24 VDC	
RUAA	'Report Port A Analog	RINA(V1,0)	'Report Port 0 Analog
RUBA	'Report Port B Analog	RINA(V1,1)	'Report Port 1 Analog
RUCA	'Report Port C Analog	RINA(V1,2)	'Report Port 2 Analog
RUDA	'Report Port D Analog	RINA(V1,3)	'Report Port 3 Analog
RUEA	'Report Port E Analog	RINA(V1,4)	'Report Port 4 Analog
RUFA	'Report Port F Analog	RINA(V1,5)	'Report Port 5 Analog
RUGA	'Report Port G Analog	RINA(V1,6)	'Report Port 6 Analog
CLASS 4 PLS2		**CLASS 5**	
Assign analog value to a variable		**Assign analog scaled voltage to a variable**	
x=UAA	'Assign Port A analog to "x"	x=INA(V1,0)	'Assign Port 0 analog to "x"
x=UBA	'Assign Port B analog to "x"	x=INA(V1,1)	'Assign Port 1 analog to "x"
x=UCA	'Assign Port C analog to "x"	x=INA(V1,2)	'Assign Port 3 analog to "x"
x=UDA	'Assign Port D analog to "x"	x=INA(V1,3)	'Assign Port 4 analog to "x"
x=UEA	'Assign Port E analog to "x"	x=INA(V1,4)	'Assign Port 5 analog to "x"
x=UFA	'Assign Port F analog to "x"	x=INA(V1,5)	'Assign Port F analog to "x"
x=UGA	' Assign Port G analog to "x"	x=INA(V1,6))	'Assign Port 6 analog to "x"

I. Timers – 4 Independent Count-down Timers with Millisecond Resolution

Each timer is a count-down timer from set point to zero in milliseconds.
Their purpose is to trigger interrupts after they time out.

1) Setting a timer:

```
TMR(x,t)        'Set Timer "x" for "t" milliseconds
```

Where:

x refers to one of the four timers: 0, 1, 2, or 3
t is the initial value signed +/-2147483648 of the timer

2) Reporting a timer present value

```
RTMR(x)         'Report timer "x" (present time left in milliseconds)
```

3) Reporting Timer Status bits (bit set to 1 when timer times out)
Timer bits are in Status Word 4 i.e. RB(4,0) would report Timer 0

II. User Program Interrupts – 8 total

Interrupts allow automatic triggering of subroutine calls based on any status bit available.
They may be triggered on rising or falling edge trigger of a state change of the status bits.
Once triggered, the subroutine is called immediately.

1) Overview

a. One Interrupt scanner for the operating system is shared by different User Tasks; therefore an interrupt event can jump to another Task even if it wasn't scheduled to run next. This would include a task that maybe asleep.
NOTE: Present Firmware has only One User Programmable Task.

b. When an Interrupt has occurred, it shall cause the user program to jump to a label (1-999) within the program of any task. The User shall need to be careful not to duplicate label within different tasks.

c. The global interrupt flag is NOT disabled when an interrupt is entered.

d. At any time the Interrupt scanner maybe enabled or disabled. It may be enable within an interrupt subroutine label.

e. If an Interrupt is redefined while it is enabled, it will be disable at the time the ITR(xx) command is run. If the re-definition is in error the ITR(x,…) will remain undefined and not enabled.

f. Trying to enable an interrupt that is not defined has no action.

g. The identifier in the ITR() command (0-7) is also the priority level. 0 is the highest priority.

h. An interrupt cannot be interrupted by an interrupt of the same ID/priority.
WARNING !!!!
If a second interrupt comes in, it is logged but not acted upon until the first finishes.

i. **WARNING !!!!** If an interrupt of higher priority comes in, it interrupts the current interrupt. When it is done, execution resumes in the lower priority interrupt.

The WAIT statement has separate timers; one for the main loop and 1 for each of the 8 interrupt IDs. Since an interrupt of the same ID cannot ever interrupt itself, this is sufficient. The wait timers are always counting down at a 1 millisecond tick even if that thread has been interrupted.

2) Interrupt Instructions and Commands

ITR (X, STATUS_WRD#, BIT#, S, LABEL#) defines the interrupt

Where:

X is the interrupt number 0-7

STATUS_WRD# 0-6, 16 and 17 presently

BIT# 0 to 15 is the Status Word bit from which the interrupt event occurs.

S is the state in which causes the interrupt, if 0, then a transition of the bit from 1 to 0 will cause the interrupt.

LABEL# is the instruction label of the start of the interrupt subroutine.

```
ITRE        'Global Interrupt Scanner Enable

ITRD        'Global Interrupt Scanner Disable

EITR(x)     'Individual Interrupt Enable where:
            'x is the interrupt number 0-7

DITR(x)     'Individual Interrupt disable where:
            'x is the interrupt number 0-7

RETURNI     'Interrupt Return
```

When executed, program flow will return from the interrupt to the program label saved on the interrupt stack.

WARNING !!!!

Take care in use of RETURN vie RETURNI commands! Do NOT place a RETURN in front of RETURNI or vice versa. This can cause unexpected program flow to occur!

Note: A program MUST BE running for the global interrupt scanner to continue operation.
If END is issued, the global interrupt scanner will be disabled and interrupts will cease to operate.
There are three options to maintaining a program running to keep form reaching an END commands.

```
WHILE 1 LOOP
END
'or
C1   GOTO1
END
'or
PAUSE    'code execution will pause here until
         'RESUME is issued from the serial port.
END
```

This document covers Status Words for Class 5 Motors. A Status Word is a 16-Bit Indicator of various parameters in the SmartMotor™.

Each Status Word is accessible from the Motor Polling Window

Commands

Reports:

```
RW(sw)              'Report status word

or

x=W(sw)             'Assign Status word to the variable "x"



RB(sw,b)            'Report Stauts Bit

or

x=B(sw,b)           'Assign Status word to the variable "x"
```

Assignments:

```
Z(sw,b)             'Clears/zeros a status word bit
```

Where:

sw – is the status word #

b – is the bit #

Definitions:

Fault : a bit that will cause motor to stop when asserted. Motor goes into MTB mode. Users must clear the bit, and then command motor to start again.

Historical: bit sticks until cleared by user, but does not cause motor to stop.

Indicator: simply states the current state based on motor condition.

Status Word:0					SW(0) Primary Fault/Status Indicator
Bit	**Value**	**Type**		**Clear**	**Description**
0	1	Indicator			Drive ready – No faults exist and enough bus voltage
1	2	Indicator	Bo		Motor is off
2	4	Indicator	Bt		Trajectory in progress
3	8	Fault			Servo Bus Voltage Fault, set on Regen fault, or while running with Low Bus.
4	16	Historical	Ba	Za	Peak Over-current occurred
5	32	Fault	Bh		Excessive temperature, requires 5 deg C below TH setting and user clear of this bit.
6	64	Fault	Be	Ze	Excessive position error
7	128	Fault	Bv	Zv	Velocity limit
8	256	Indicator			Real-time temperature limit
9	512	Fault			First Derivative (DE/Dt) of position error over limit
10	1024	Indicator			Hardware Right (+) Over Travel Limit Enabled
11	2048	Indicator			Hardware Left (-) Over Travel Limit Enabled
12	4096	Historical	Br	Zs	Right (+) Over Travel Limit
13	8192	Historical	Bl	Zl	Left (-) Over Travel Limit
14	16384	Indicator	Bp		Right (+) Over Travel Limit Active
15	32768	Indicator	Bm		Left (-) Over Travel Limit Active
Status Word:1					**SW(1) Index Registration and Software Over Travel Limits**
Bit	**Value**	**Type**			**Description**
0	1	Indicator			Arming Bit for Rise Capture of Encoder 0
1	2	Indicator			Arming Bit for Fall Capture of Encoder 0
2	4	Historical	Bi(0)		Rising Edge Capture on Encoder 0
3	8	Historical	Bj(0)		Falling Edge Capture on Encoder 0
4	16	Indicator			Arming Bit for Rise Capture of Encoder 1
5	32	Indicator			Arming Bit for Fall Capture of Encoder 1
6	64	Historical	Bi(1)		Rising Edge Capture on Encoder 1
7	128	Historical	Bj(1)		Falling Edge Capture on Encoder 1
8	256	Indicator	Bx(0)		Capture Input State 0
9	512	Indicator	Bx(1)		Capture Input State 1
10	1024	Indicator			Software Over Travel Limits Enabled
11	2048	Indicator			Software Over Travel Limit Mode: [0: Don't Stop] [1: Fault will occur MTB issued, Default]
12	4096	Historical	Brs	Zrs	Software Positive Over Travel Limit Occurred
13	8192	Historical	Bls	Zls	Software Negative Over Travel Limit Occurred
14	16384	Indicator	Bps		Software Positive Over Travel Limit Active
15	32768	Indicator	Bms		Software Negative Over Travel Limit Active

Status Word:2					SW(2) PID State, Brake Control State, Move Generation Indicators
Bit	Value	Type			Description
0	1	Indicator			Com Channel 0 (RS-232) General Error, Use RCHN(0) to get full status
1	2	Indicator			Com Channel 1 (RS-485) General Error, Use RCHN(1) to get full status
2	4	Indicator			Reserved
3	8	Indicator			Reserved
4	16	Indicator			CAN Bus General Error
5	32	Indicator			Reserved
6	64	Indicator			Reserved
7	128	Indicator			Reserved
8	256	Indicator			Reserved
9	512	Indicator			Animatics Data Block Checksum Error
10	1024	Indicator			Program RUNNING !
11	2048	Indicator			Trace In Progress (Currently in Alpha Test)
12	4096	Histroical			EEPROM Write Buffer Overflow, Last Write to EEPROM exceeded buffer and was denied
13	8192	Indicator			EEPROM Busy (Write In Progress)
14	16384	Historical	Bs	Zs	Command Syntax Error NOTE !!!! See ERRC Command Errors Info
15	32768	Indicator	Bk		Main Program Checksum Error, Program is Corrupt and cannot run
Status Word:3					**SW(3) PID State, Brake Control State, Move Generation Indicators**
Bit	Value	Type			Description
0	1	Historical			Position Error has Exceeded Software Limit
1	2	Indicator			Torque Saturation, Drive is Running at 100% PWM
2	4	Indicator			Voltage Saturation, Max Bus Voltage !
3	8	Historical	Bw	Zw	Wrap around Occurred, Position wrapped through +/- 2^31
4	16	Indicator			KG (Gravitational Offset Gain) enabled
5	32	Historical			Low Bus Voltage
6	64	Historical			Regen/Shunt Fault and/or High Bus Voltage (Class 9)
7	128	Historical		ZS	I/O Fault Latch
8	256				
9	512				
10	1024				
11	2048				
12	4096	Indicator			Brake Asserted
13	8192	Indicator			Brake OK, Is Internally present or configured to an external I/O point
14	16384	Indicator			"G" command has been configured to an Input
15	32768	Indicator			Velocity target reached

Status Word:4					SW(4) PID State, Brake Control State, Move Generation Indicators
Bit	Value	Type			Description
0	1	Indicator			Timer 0 Running: (Not timed out yet)
1	2	Indicator			Timer 1 Running: (Not timed out yet)
2	4	Indicator			Timer 2 Running: (Not timed out yet)
3	8	Indicator			Timer 3 Running: (Not timed out yet)
4	16				
5	32				
6	64				
7	128				
8	256				
9	512				
10	1024				
11	2048				
12	4096				
13	8192				
14	16384				
15	32768				

Status Word:5					SW(5) PID State, Brake Control State, Move Generation Indicators
Bit	Value	Type			Description
0	1	Indicator			Interrupt 0 Enabled
1	2	Indicator			Interrupt 1 Enabled
2	4	Indicator			Interrupt 2 Enabled
3	8	Indicator			Interrupt 3 Enabled
4	16	Indicator			Interrupt 4 Enabled
5	32	Indicator			Interrupt 5 Enabled
6	64	Indicator			Interrupt 6 Enabled
7	128	Indicator			Interrupt 7 Enabled
8	256				
9	512				
10	1024				
11	2048				
12	4096				
13	8192				
14	16384				
15	32768	Indicator			Interrupt Event Scanner Enabled

Status Word: 6					SW(6) PID State, Brake Control State, Move Generation Indicators
Bit	Value	Type			Description
0	1	Indicator			Running Standard Trapezoidal Mode (Direct From Hardware Commutation)
1	2	Indicator			Running Enhanced Trapezoidal Mode (Encoder Position Emulated Commutation)
2	4	Indicator			Running Sinusoidal Commutation
3	8				
4	16				
5	32				
6	64				
7	128				
8	256				Commutation Calibration OK (should =1 after first detection of internal index mark)
9	512	Indicator			TOB (Trajectory Overshoot Braking) Enabled
10	1024				Commutation Is Inverted (MINV(1) has been issued)
11	2048				MTB (Mode Torque Brake) is active
12	4096				
13	8192				
14	16384				
15	32768				
Status Word: 16			**Connector**		**SW(16) On Board Local I/O Status**
Bit	Value	I/O	Pin	Port	Description
0	1	0	1	A	
1	2	1	2	B	
2	4	2	3	C	
3	8	3	4	D	
4	16	4	5	E	
5	32	5	6	F	
6	64	6	7	G	
7	128	7	-	-	Internal I/O point may be set for purposes of triggering Interrupts
8	256				
9	512				
10	1024				
11	2048				
12	4096				
13	8192				
14	16384				
15	32768				

Animatics Corporation
tel: 408.748.8721 • fax: 408.748.8725 • www.animatics.com

1.0 Serial Port Errors

RCHN(n) or =CHN(n) where n is the id of the serial port.
Returns an integer. Only the lower 4 bits indicate the state of the error.

bit	Error
1	Overflow error
2	Frame error
4	n/a
8	parity error

i.e. a frame error only is the number 2, a frame error and a parity error is a decimal 10.

2.0 Clearing Errors

ZS Clears the following status bits all at once:

Left and right hard limit history fault.
Left and right soft limits history fault.
Wraparound position history.
Command error history.
Over current history.
Over temp Error (as long as the 5 degree cooling has taken place first)
Position error limit fault.
Position error first derivative positive limit fault.
Velocity error fault.
Low bus voltage fault. Clears it, though in Class 5 this feature not implemented yet.
Clears index captures rise and fall for internal and external capture.
Can0error history cleared.

Z(status word, status bit) Clears specific status word bits. Not all bits can be cleared this way.

```
IF B(0,13)              'If Historical Left Limit Bit is set
       Z(0,13)          'Clear that specific bit
ENDIF

IF B(x,y)               'Generic Bit checking and clearing
       Z(x,y)           'Not recommended as a general rule though
ENDIF
```

Individual bits with explicit commands include:

```
Zl     'Clear left hard limit fault
Zr     'Clear right hard limit fault
Zw     'Clear wrap bit history
Zs     'Clear command error
Ze     'Clear position error
Zh     'Clear over temp history fault
Za     'Clear over current history
Zls    'Clear left soft limit history/fault
Zrs    'Clear right soft limit history/fault
```

SmartMotor Part Number		
Continuous Torque	2.50	in-lb
	40	oz-in
	0.28	N-m
Peak Torque	4.00	in-lb
	64	oz-in
	0.45	N-m
	181	Watt

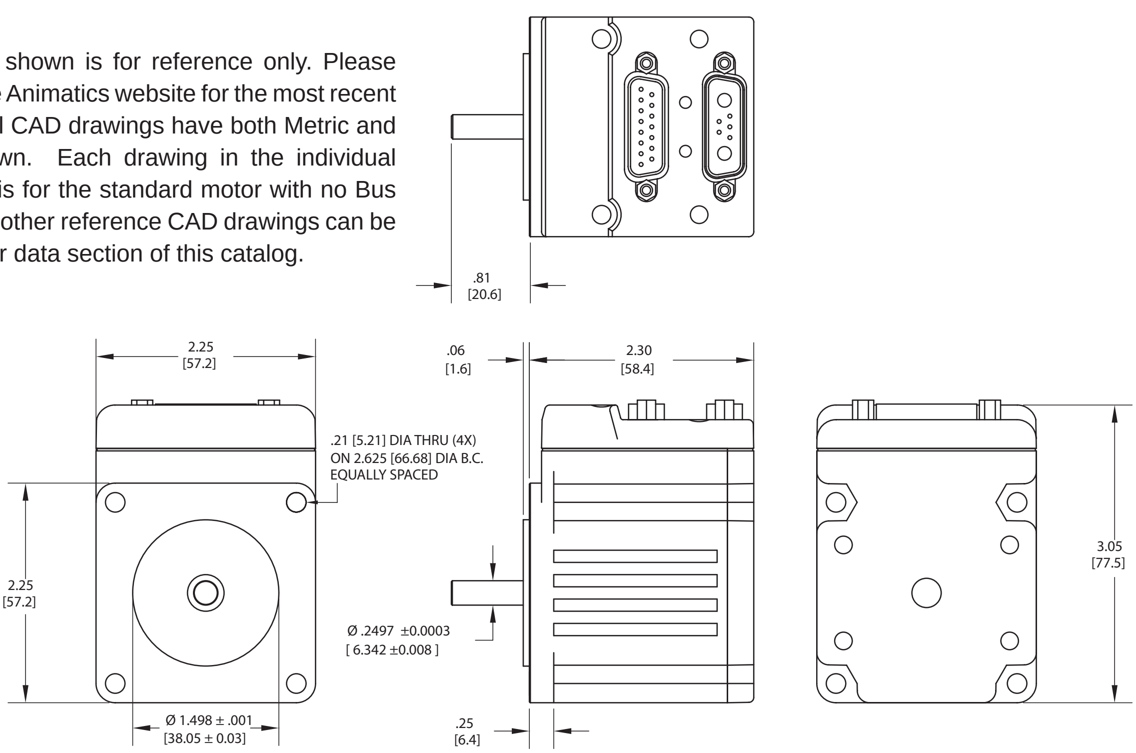
The following section covers individual motor data for each Class 5 SmartMotor™.

Data Tables

The data table shows **maximum** sustained Torque, Power and Current. Note that the continuous torque numbers are typically over a given RPM range, however the Peak Torque is always at Zero-RPM stall and will not be available at higher RPMs.

CAD Drawings

Each CAD drawing shown is for reference only. Please consult factory or the Animatics website for the most recent drawing revision. All CAD drawings have both Metric and Standard units shown. Each drawing in the individual motor table section is for the standard motor with no Bus or Brake options. All other reference CAD drawings can be found after the Motor data section of this catalog.



All SmartMotor data and specifications are subject to change without notice. Please consult the factory for the latest updates.

Understanding Animatics Torque Curves

Each Set of Torque curves depicts limits of both Continuous and Peak torque for the given SmartMotor™ over their full range speed.

Peak Torque Curve:

The Peak Torque Curve is derived from dyno testing and is the point at which peak current limit hardware settings of the drive prevent further torque in an effort to protect drive stage components.

Continuous Torque Curve:

The continuous Torque Curve is also derived from dyno testing, but is instead the point at which the temperature rises from an ambient of 25ºC to the designed thermal limit.

For example, the motor will be placed on the dyno tester and set to operate at 1000 RPM continuously with the load slowly increased until the controller reaches its maximum sustained thermal limit. This limit is either 70ºC or 85ºC depending on the model number. All PLS2 SmartMotors are set to 85ºC.

The far lower right side of the curve is limited by supply voltage. This is the point at which Back EMF suppresses any further speed increase. Higher supply voltages will shift the zero torque point of the curves further to the right.

Ambient Temperature Effects on Torque Curves and Motor Response:

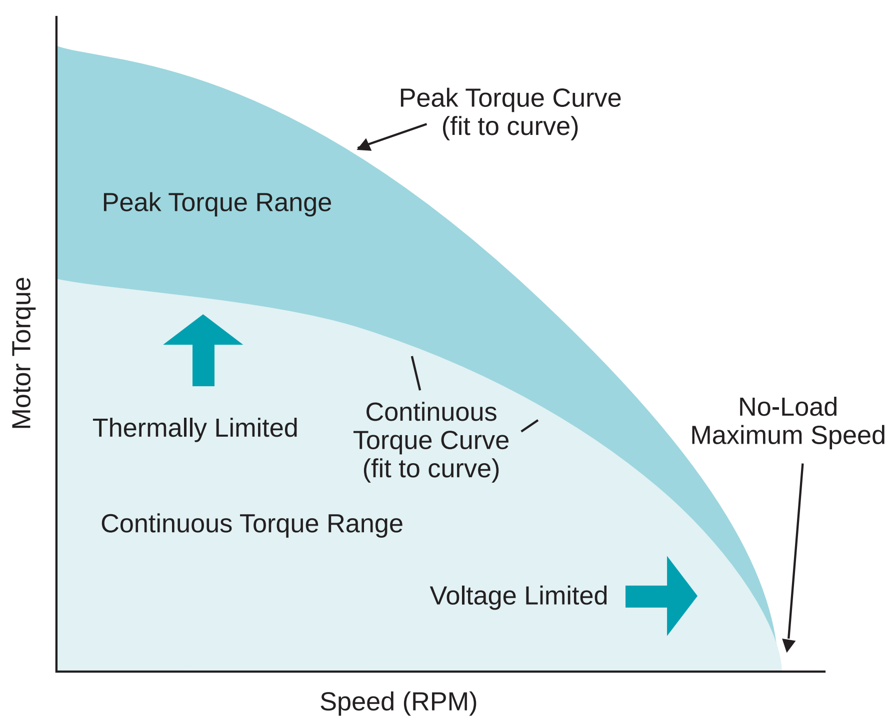
If the motor is operated in an environment greater than 25ºC, then it will reach its thermal limit faster for the same given load thereby further limiting continuous torque.

Therefore; any given motor torque curve MUST BE linearly de-rated for a given ambient temperature from 25ºC to 70ºC, 85ºC for all PLS2 SmartMotors.

Supply Voltage Effects on Torque Curves and Motor Response:

Higher voltages have two-fold effects on torque curves. As mentioned above, raising voltage will shift the curve to the right. It will also allow higher current into the drive. However, Torque curves depict Torque at a given velocity.

If you double supply voltage, the motor can sustain twice the original velocity. But since acceleration is the differential of velocity, it can achieve 4 times the original acceleration. This is useful for high speed indexing and fast start/stop motion.



All Torque Curves in this catalog also have SHAFT OUTPUT Power Curves overlaid on them as well.

Power can be found by the following equation:

Power (kW) = Torque (N.m) x Speed (RPM) / 9.5488

For any given mechanical system being moved by a SmartMotor™, it is ideal to insure the motor is running within its optimum performance range. This can be achieved via proper mechanical system design by adjusting one of the following as it may apply:

- Gear Reduction
- Belt Reduction
- Lead Screw Pitch
- Pinion Gear diameter

Example 1: (Rotary Application)

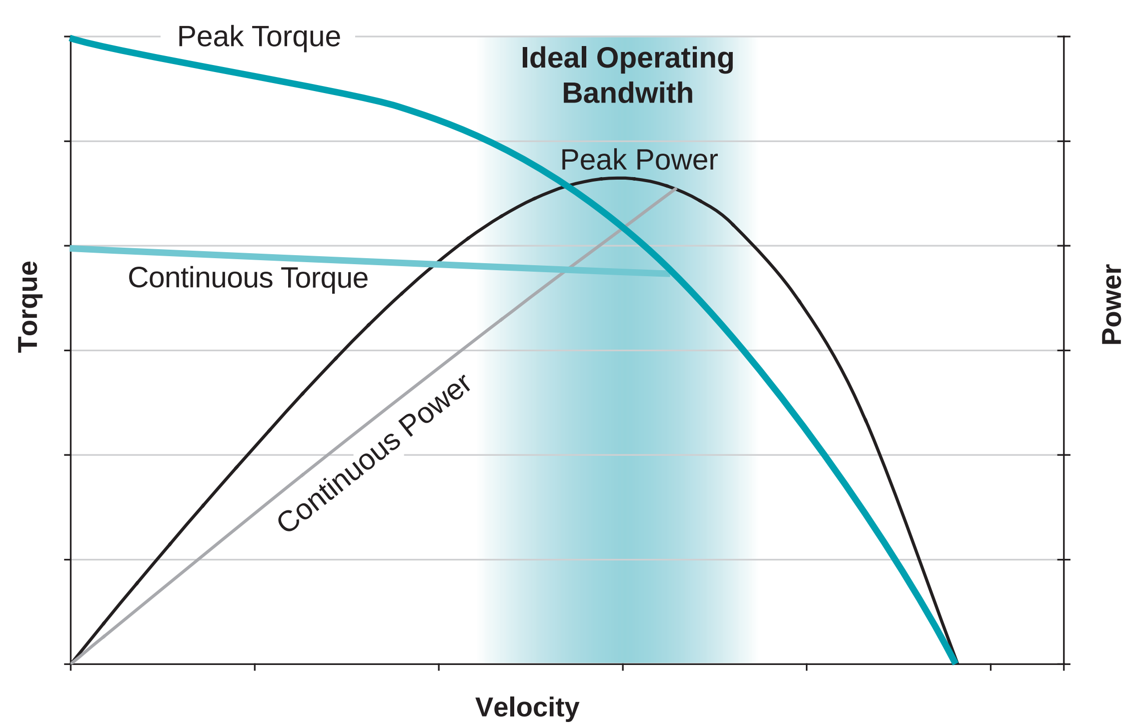
Suppose you have a load that requires 300 RPM at the output of a gear head. Suppose the optimum speed range for the motor is 2100 RPM.

Divide the optimum operating speed by the load speed to get the ideal gear reduction. In this case : 2100 RPM / 300 RPM=7. So a 7:1 gear reduction would allow the motor to operate in its most efficient range.

Example 2 (Linear Application)

Suppose you need to run at 100mm/second via a ball screw and the motor has an ideal range of 3000RPM. 3000RPM/60= 50 Rotations per second. 100mm/sec divided by 50RPS is 2mm per rotation.

So an ideal pitch would be 2mm.



Considerations when using torque curves for motor sizing:

For any given product model number, there may be variations of as much as +/-10%.

The following diagram depicts data points collected from dyno testing of a given model motor. A best-fit torque curve is created from these data points and is then de-rated to at least 5% below the worst case data points. The de-rated curve is what is advertised. This means that within any given model number, EVERY motor sold will perform at or better than the advertised torque. Theoretically, ALL motors should be no less than 5% better than advertised and may be better than 20% higher.

The diagram shows motor loading in 4 areas.

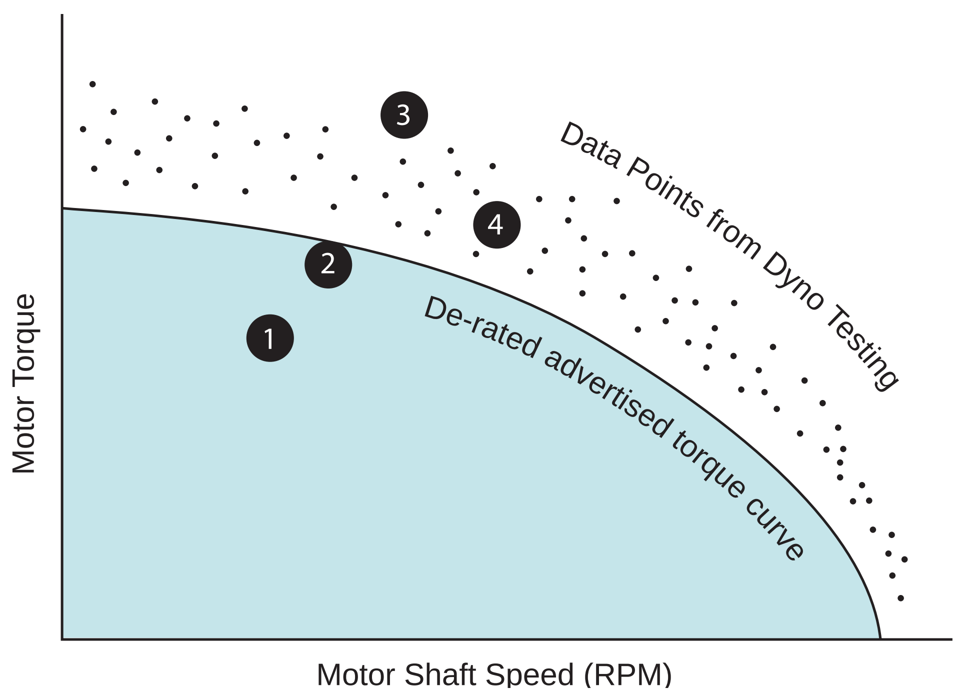
1. This is ideal and depicts a load within the normal operating range of the motor. The motor should operate well and have no problems for many years.
2. The load is very close to the operating limit. The motor will run quite warm as compared to Point 1.
3. The load exceeds the advertised level and exceeds +10% expected range of possible torque capabilities. In this case, the motor will most likely either overheat quickly and fault out or immediately get a position error because it simply does not have enough power to support the load demand.

⚠ WARNING

4. The load exceeds the advertised operating limit of the motor. However, due to data scatter and de-rating, there may be some motors that will work and others that do not.

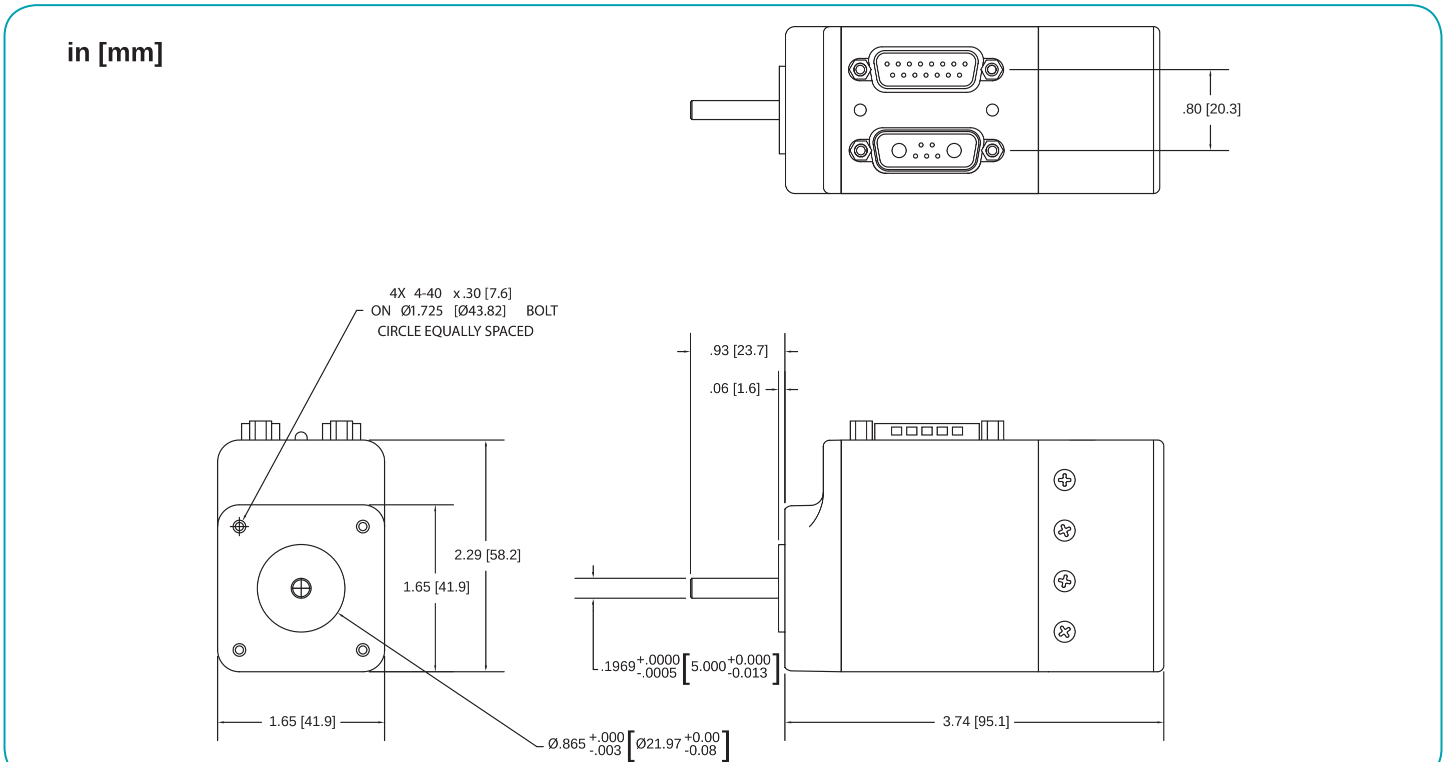
Why? Because it is in the area of +/-10% variation expected in motors for a given size. This can become a major problem. Imagine designing a machine that operates in this range. Then you replicate that machine with many of them running on a production floor. One day, a motor at the lower end of the +/-10% expected variation would be placed on a new machine and that motor would get spurious drive faults. It would appear as though the motor is malfunctioning because… "all the other motors work just fine". This is unfortunate because, in reality, all motors were undersized and operating outside of their advertised limits.

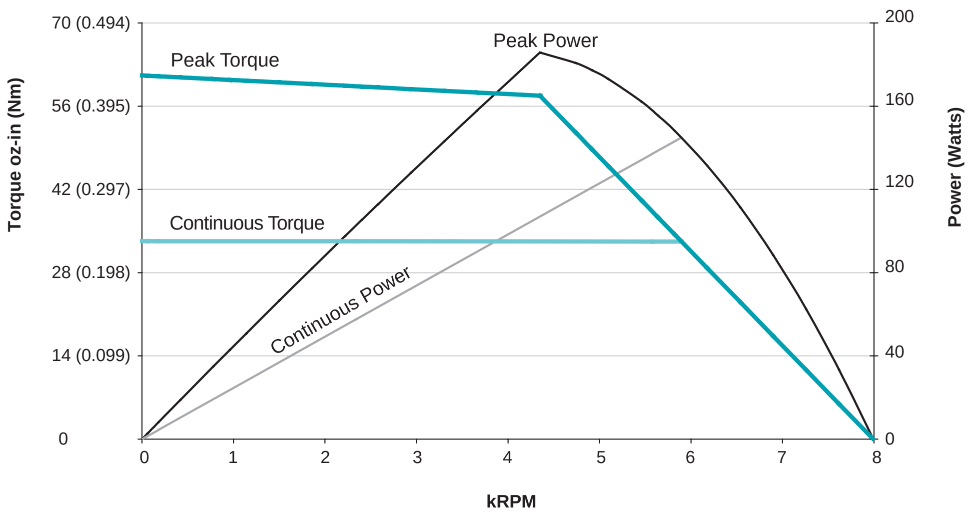
This is why it is important to properly calculate load torque to ensure the correct motor is designed into the application. Never assume that without proper load calculation and motor sizing, that testing of one motor means all of that size may work. This is simply not the case. **Try to keep operating conditions below the advertised limits to ensure reliable long-life operation.**



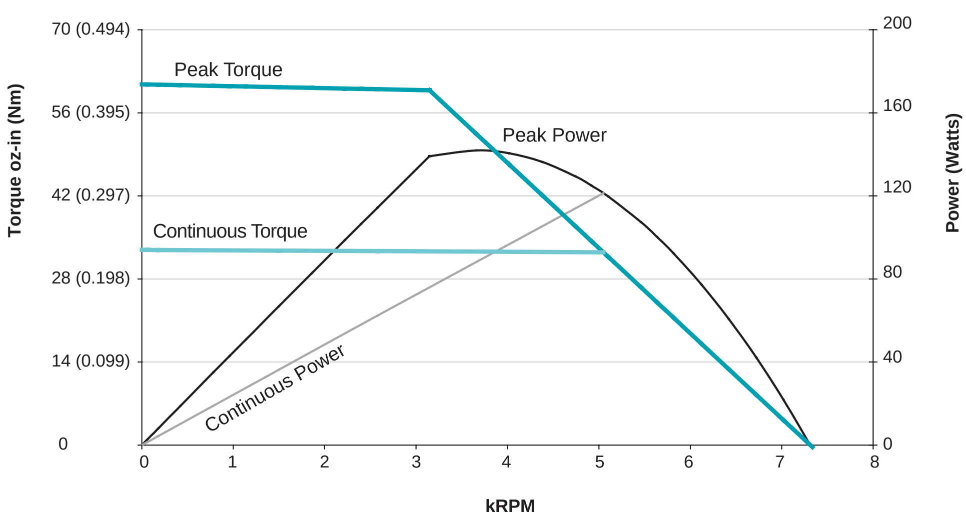
SM17205D		
Continuous Torque	2.08	in-lb
	33	oz-in
	0.24	N-m
Peak Torque	3.82	in-lb
	61	oz-in
	0.43	N-m
Nominal Continuous Power	145	Watt
No Load Speed	7,900	RPM
Continuous Current @ Nominal Power	4.5	Amps
Voltage Constant	5.5	V/kRPM
Winding Resistance	1.8	ohms
Encoder Resolution	4,000	Counts/Rev
Rotor Inertia	0.00026	oz-in-sec^2
	0.184	10^{-5} Kg-m^2
Weight	1.2	lb
	0.55	kg
Shaft Diameter	0.197	in
	5.00	mm
Shaft, Radial Load	7	lb
	3.18	kg
Shaft, Axial Thrust Load	3	lb
	1.36	kg
EtherNet Available		
DeviceNet Available		
ProfiBus Available		
CanOpen Available		

Animatics SmartMotor™ SM17205D (No Options) CAD Drawing

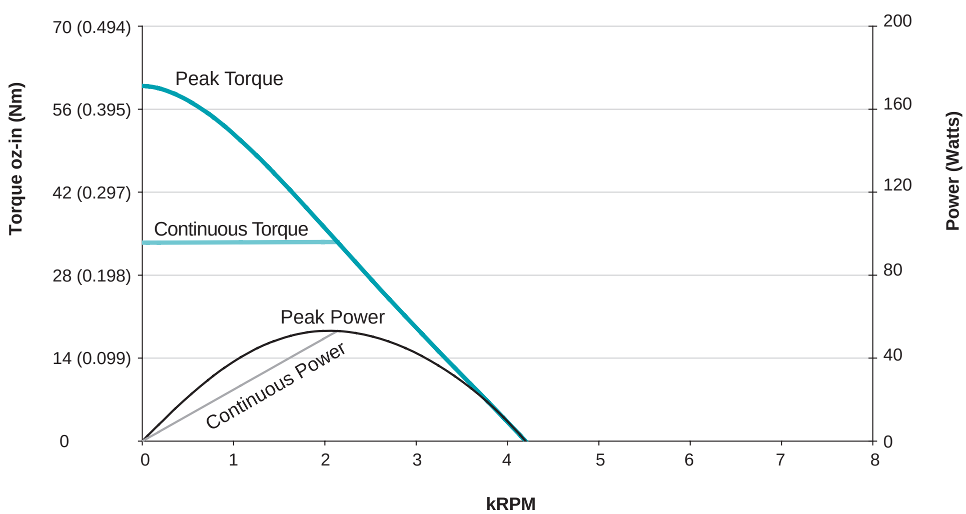




SM17205D
at 48 VDC
at rise to 85°C



SM17205D
at 42 VDC
at rise to 85°C



SM17205D
at 24 VDC
at rise to 85°C

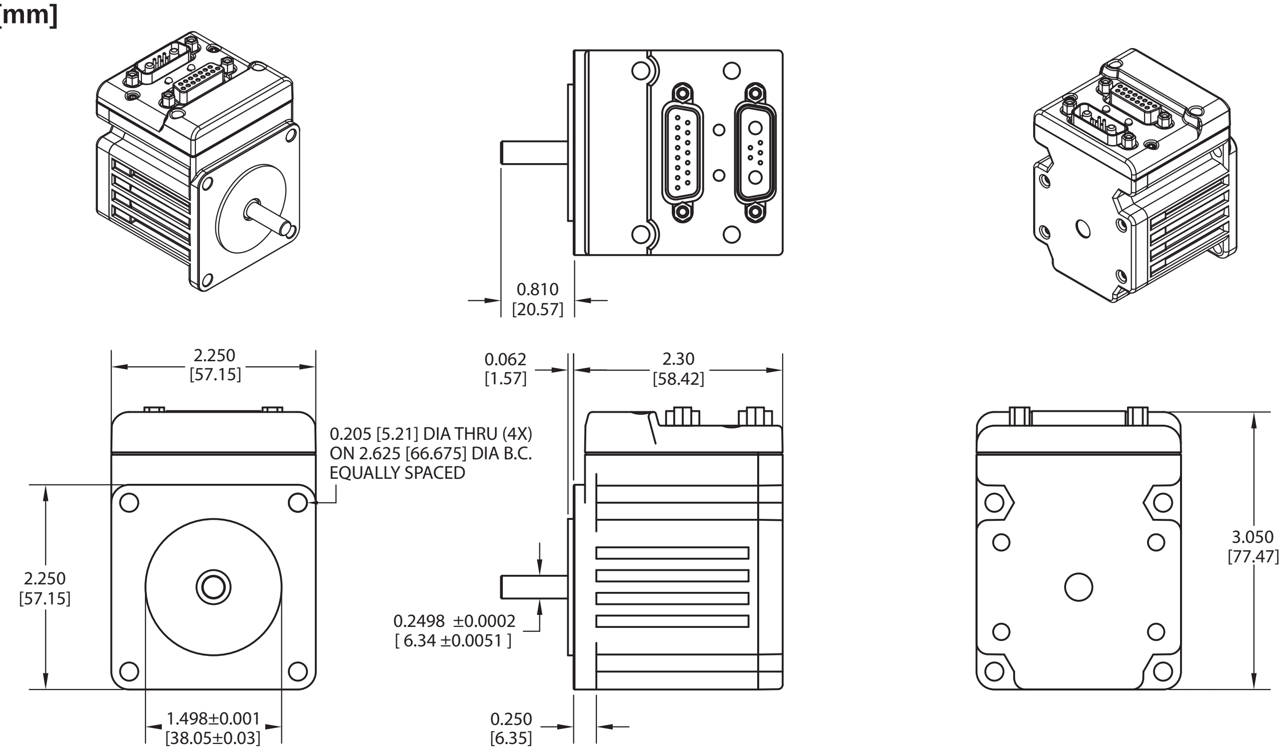
All Torque curves based on 25°C ambient.

Motors were operated in Trap-Commutation Mode.

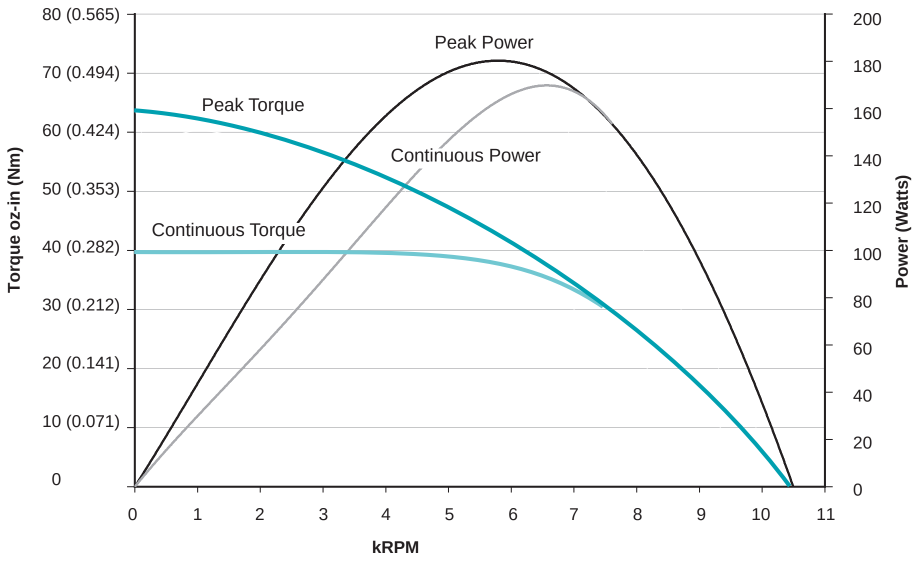
Please consult factory for Sine-Commutation Torque Curves.

SM23165D		
Continuous Torque	2.50	in-lb
	40	oz-in
	0.28	N-m
Peak Torque	4.00	in-lb
	64	oz-in
	0.45	N-m
Nominal Continuous Power	181	Watt
No Load Speed	10,400	RPM
Continuous Current @ Nominal Power	5.0	Amps
Voltage Constant	4.45	V/kRPM
Winding Resistance	1.0	ohms
Encoder Resolution	4,000	Counts/Rev
Rotor Inertia	0.00099	oz-in-sec^2
	0.699	10^{-5} Kg-m^2
Weight	1.0	lb
	0.45	kg
Shaft Diameter	0.250	in
	6.35	mm
Shaft, Radial Load	7	lb
	3.18	kg
Shaft, Axial Thrust Load	3	lb
	1.36	kg
EtherNet Available		
DeviceNet Available		
ProfiBus Available		
CanOpen Available	Yes	

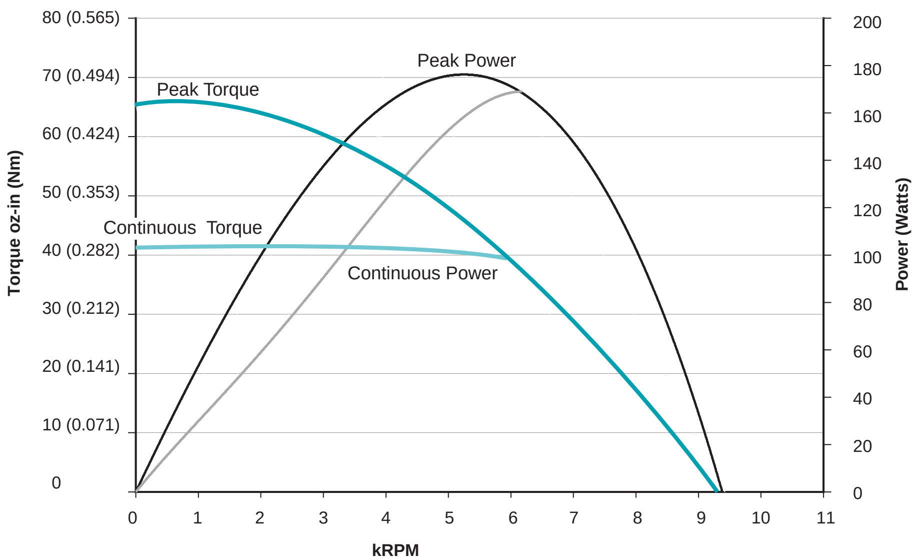
Animatics SmartMotor™ SM23165D (No Options) CAD Drawing



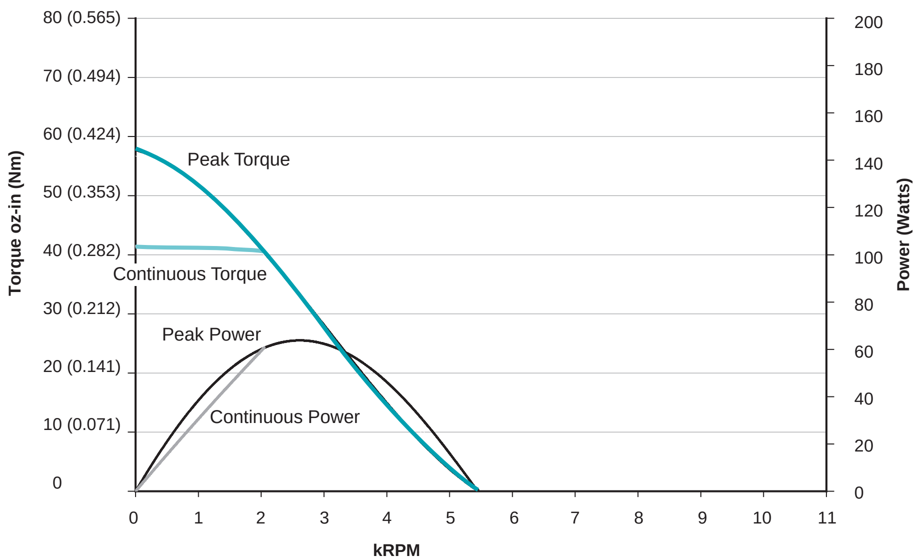
Animatics Corporation
tel: 408.748.8721 • fax: 408.748.8725 • www.animatics.com



SM23165D
at 48 VDC
at rise to 85°C



SM23165D
at 42 VDC
at rise to 85°C



SM23165D
at 24 VDC
at rise to 85°C

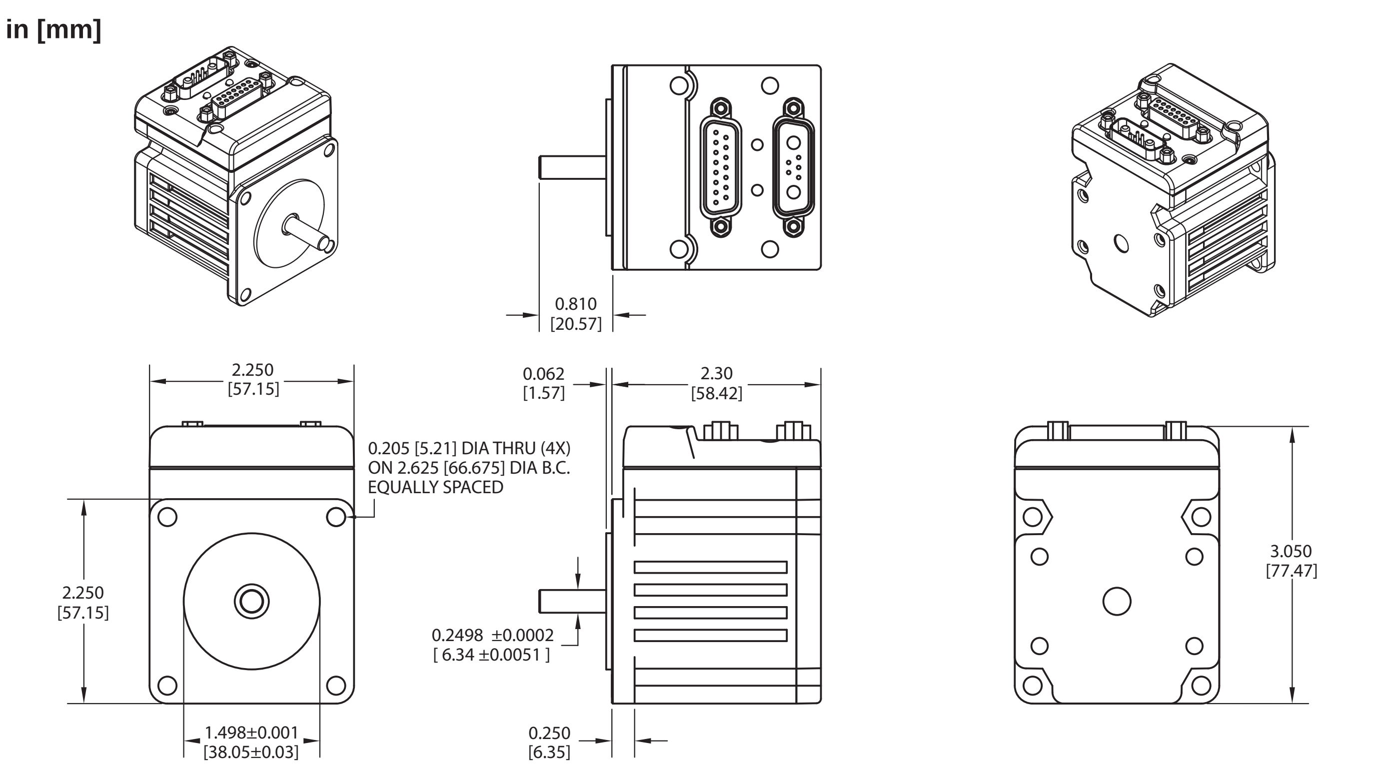
All Torque curves based on 25°C ambient.

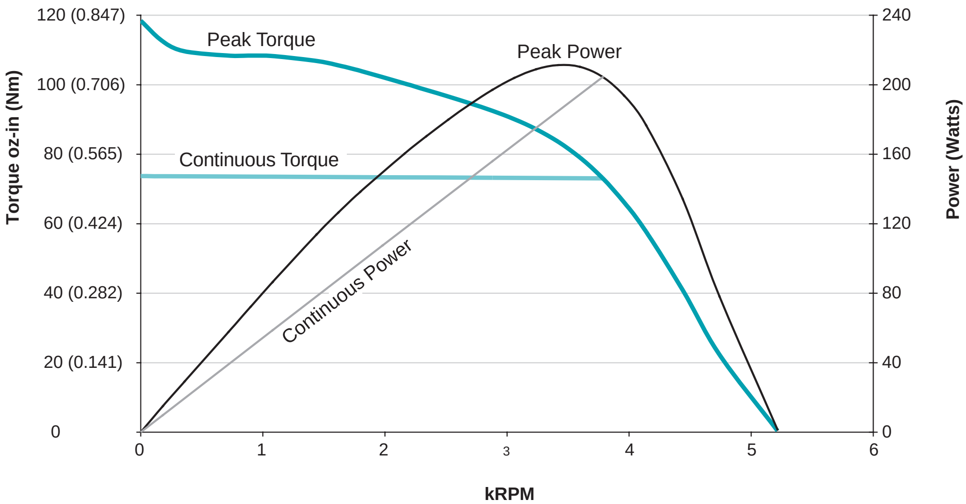
Motors were operated in Trap-Commutation Mode.

Please consult factory for Sine-Commutation Torque Curves.

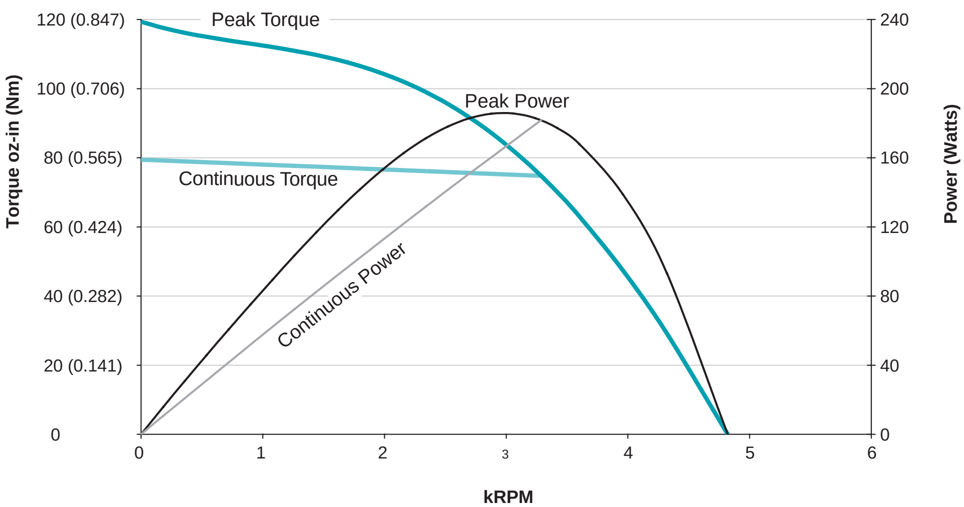
SM23165DT		
Continuous Torque	4.61	in-lb
	74	oz-in
	0.52	N-m
Peak Torque	7.40	in-lb
	118	oz-in
	0.84	N-m
Nominal Continuous Power	204	Watt
No Load Speed	5,200	RPM
Continuous Current @ Nominal Power	6.1	Amps
Voltage Constant	9.08	V/kRPM
Winding Resistance	0.7	ohms
Encoder Resolution	4,000	Counts/Rev
Rotor Inertia	0.001	oz-in-sec^2
	0.706	10^{-5} Kg-m^2
Weight	1.3	lb
	0.59	kg
Shaft Diameter	0.250	in
	6.35	mm
Shaft, Radial Load	7	lb
	3.18	kg
Shaft, Axial Thrust Load	3	lb
	1.36	kg
EtherNet Available		
DeviceNet Available		
ProfiBus Available		
CanOpen Available	Yes	

Animatics SmartMotor™ SM23165DT (No Options) CAD Drawing

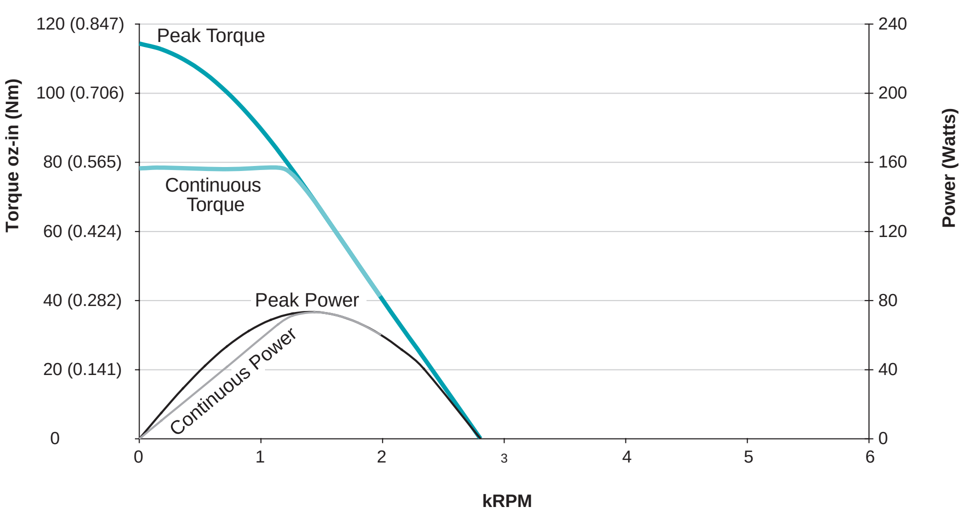




SM23165DT
at 48 VDC
at rise to 85°C



SM23165DT
at 42 VDC
at rise to 85°C



SM23165DT
at 24 VDC
at rise to 85°C

All Torque curves based on 25°C ambient.

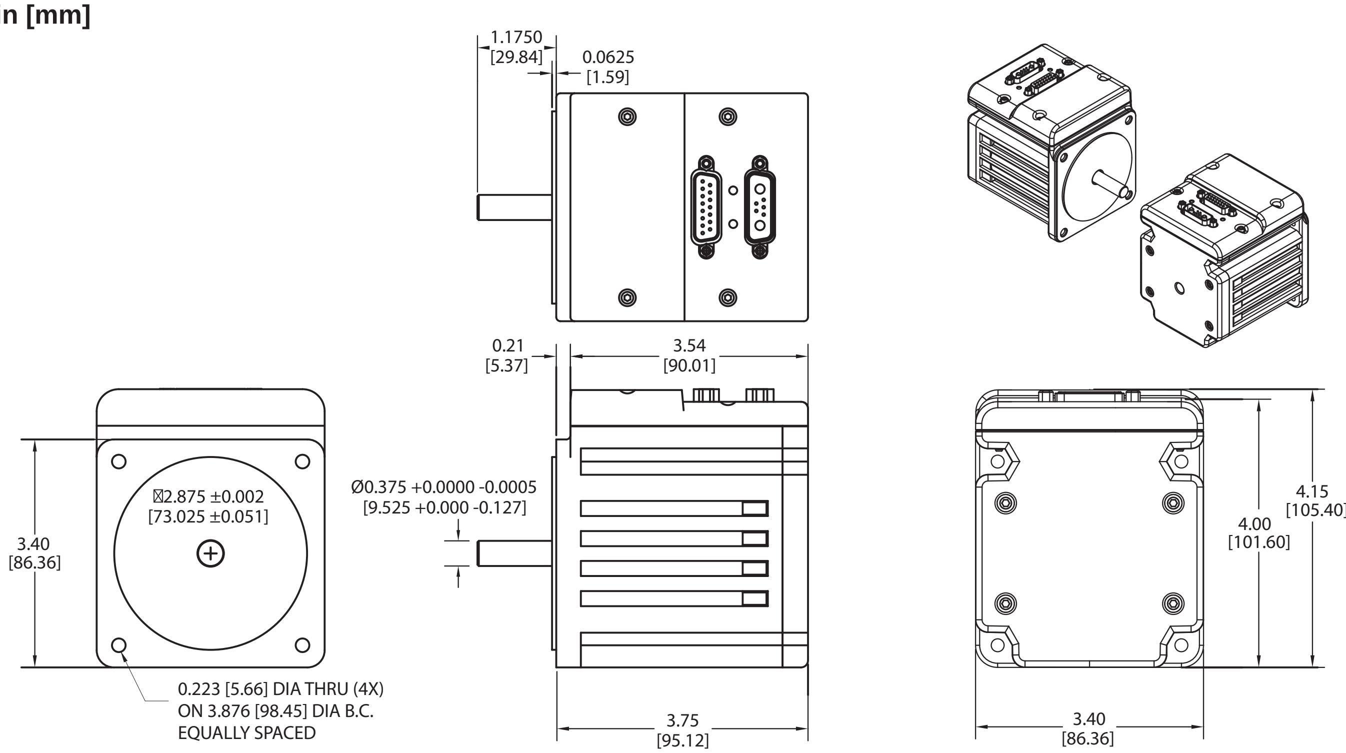
Motors were operated in Trap-Commutation Mode.

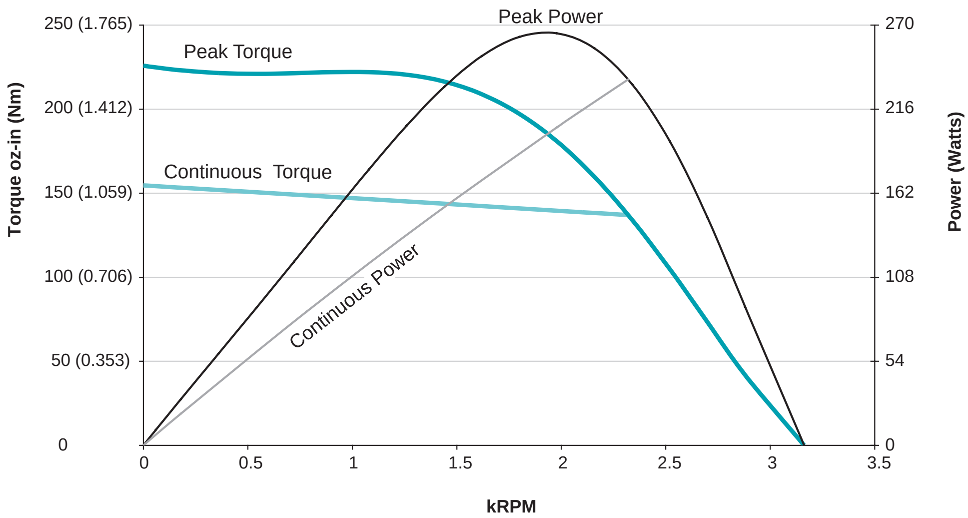
Please consult factory for Sine-Commutation Torque Curves.

SM34165D		
Continuous Torque	9.67	in-lb
	155	oz-in
	1.09	N-m
Peak Torque	14.12	in-lb
	226	oz-in
	1.60	N-m
Nominal Continuous Power	235	Watt
No Load Speed	3,100	RPM
Continuous Current @ Nominal Power	7.4	Amps
Voltage Constant	15.5	V/kRPM
Winding Resistance	0.6	ohms
Encoder Resolution	8,000	Counts/Rev
Rotor Inertia	0.014	oz-in-sec^2
	9.890	10^{-5} Kg-m^2
Weight	5.0	lb
	2.27	kg
Shaft Diameter	0.375	in
	9.53	mm
Shaft, Radial Load	15	lb
	6.80	kg
Shaft, Axial Thrust Load	3	lb
	1.36	kg
EtherNet Available		
DeviceNet Available		
ProfiBus Available		
CanOpen Available		

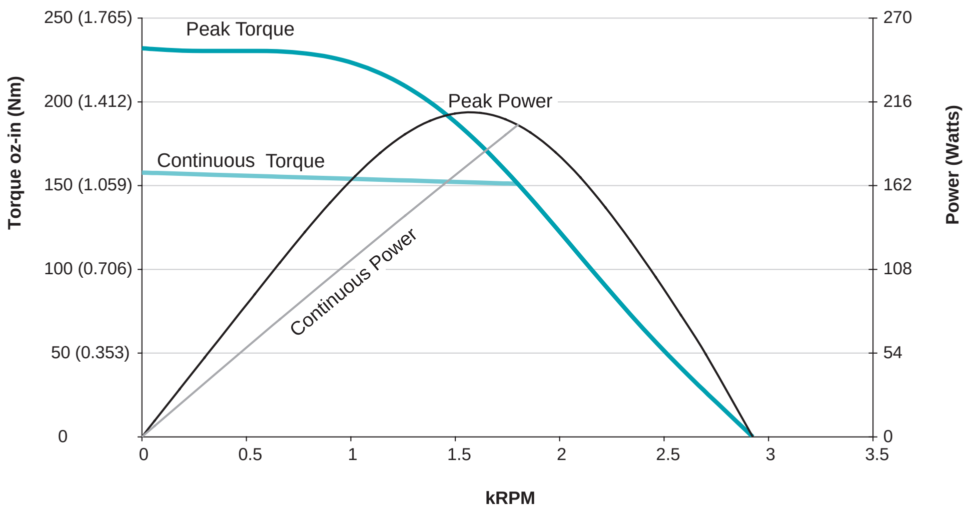
Animatics SmartMotor™ SM34165D (No Options) CAD Drawing

in [mm]

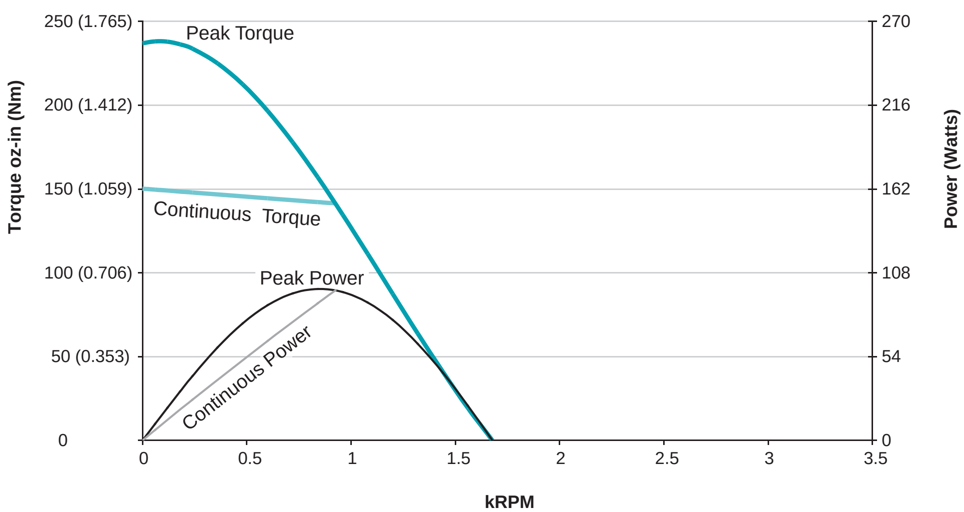




SM34165D
at 48 VDC
at rise to 85°C



SM34165D
at 42 VDC
at rise to 85°C



SM34165D
at 24 VDC
at rise to 85°C

All Torque curves based on 25°C ambient.

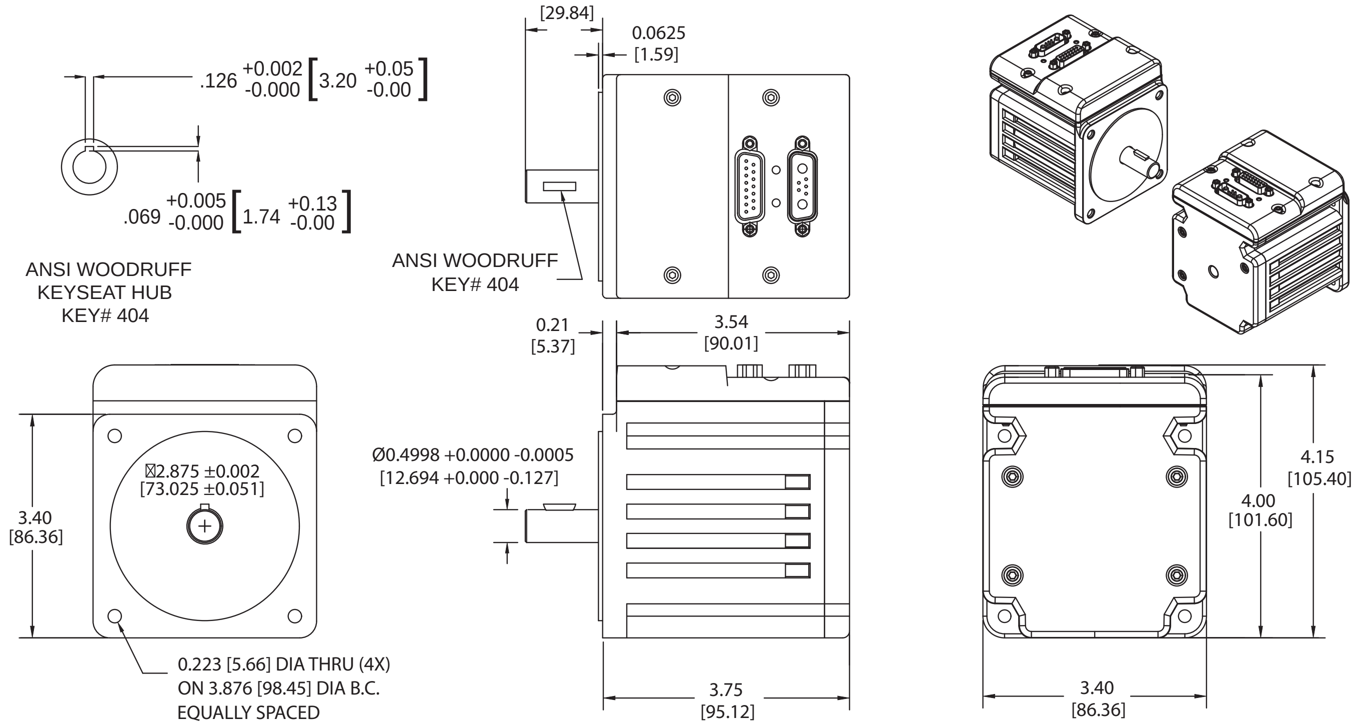
Motors were operated in Trap-Commutation Mode.

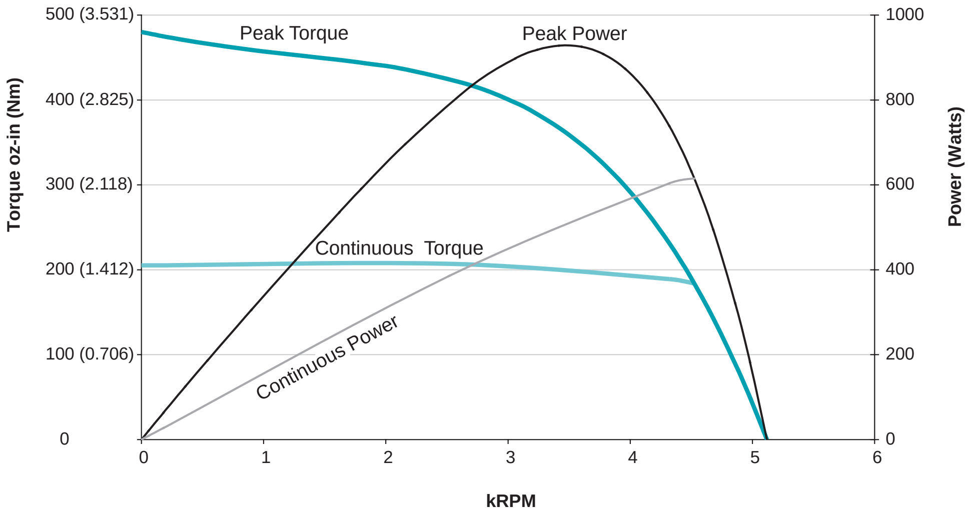
Please consult factory for Sine-Commutation Torque Curves.

SM34165DT		
Continuous Torque	12.83	in-lb
	205	oz-in
	1.45	N-m
Peak Torque	30.00	in-lb
	480	oz-in
	3.39	N-m
Nominal Continuous Power	615	Watt
No Load Speed	5,100	RPM
Continuous Current @ Nominal Power	15.5	Amps
Voltage Constant	8.9	V/kRPM
Winding Resistance	0.06	ohms
Encoder Resolution	8,000	Counts/Rev
Rotor Inertia	0.0142	oz-in-sec^2
	10.031	10^{-5} Kg-m^2
Weight	5.5	lb
	2.49	kg
Shaft Diameter	0.500	in
	12.70	mm
Shaft, Radial Load	30	lb
	13.61	kg
Shaft, Axial Thrust Load	3	lb
	1.36	kg
EtherNet Available		
DeviceNet Available		
ProfiBus Available		
CanOpen Available		

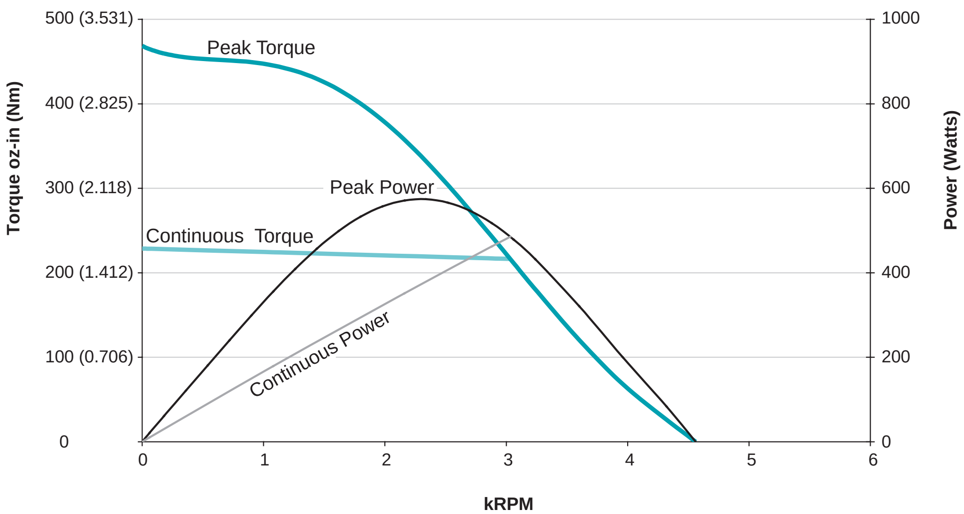
Animatics SmartMotor™ SM34165DT (No Options) CAD Drawing

in [mm]

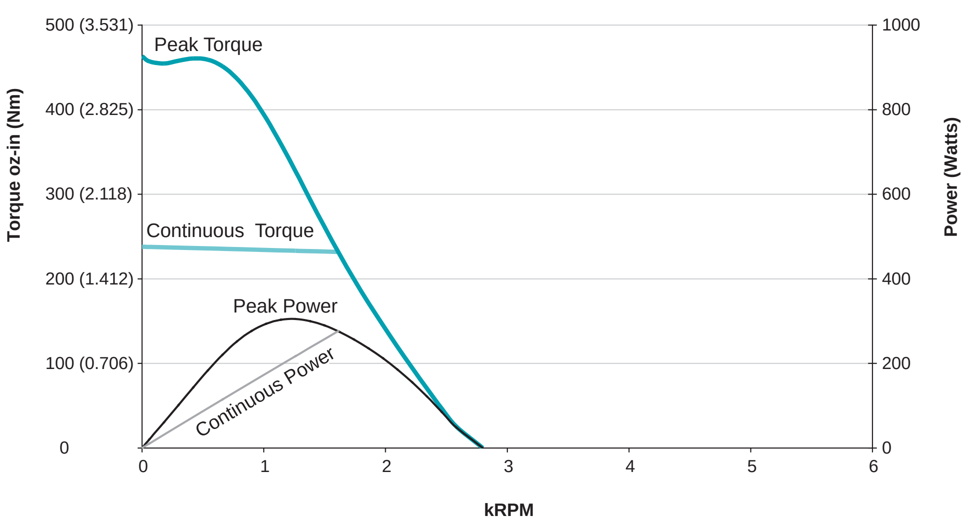




SM34165DT
at 48 VDC
at rise to 85°C



SM34165DT
at 42 VDC
at rise to 85°C



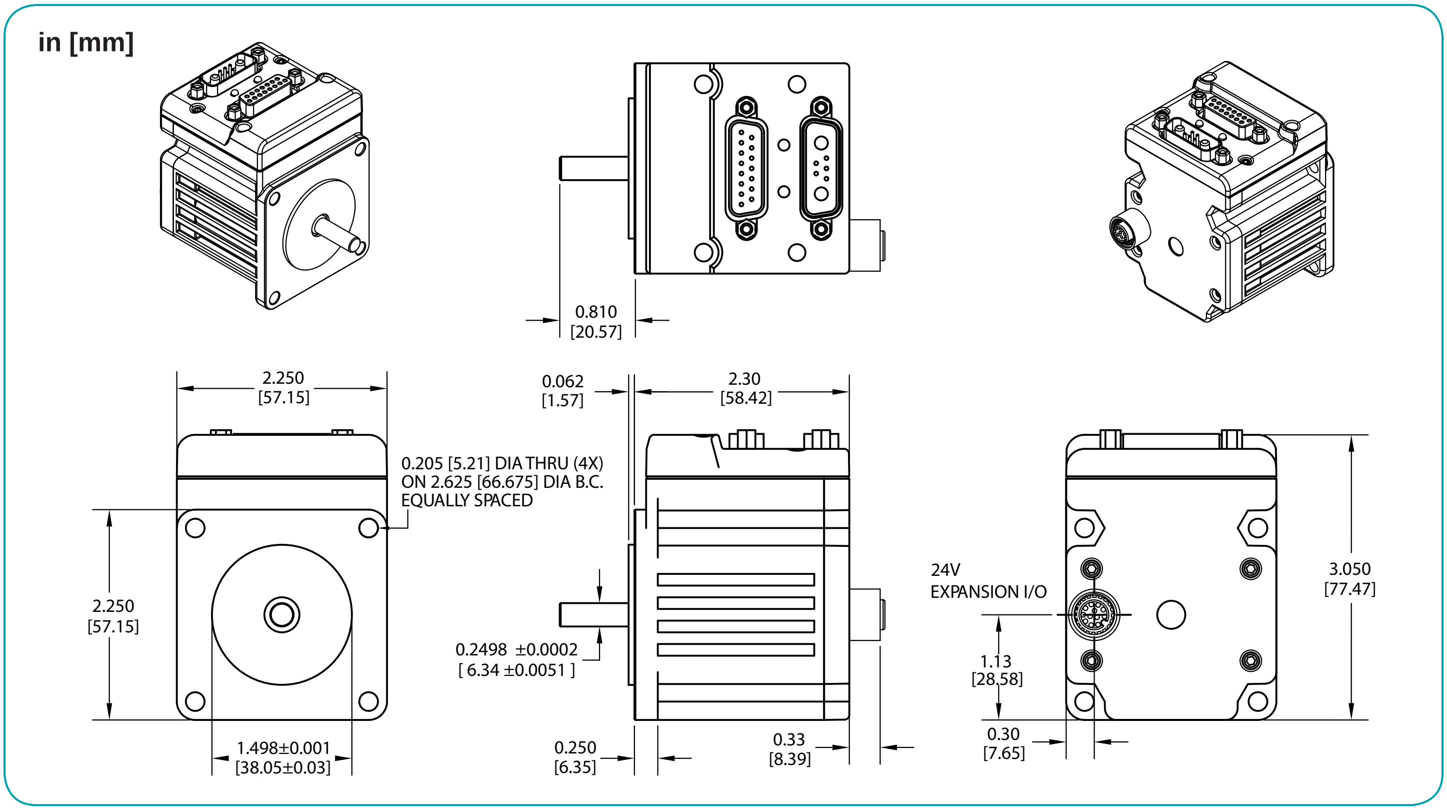
SM34165DT
at 24 VDC
at rise to 85°C

All Torque curves based on 25°C ambient.

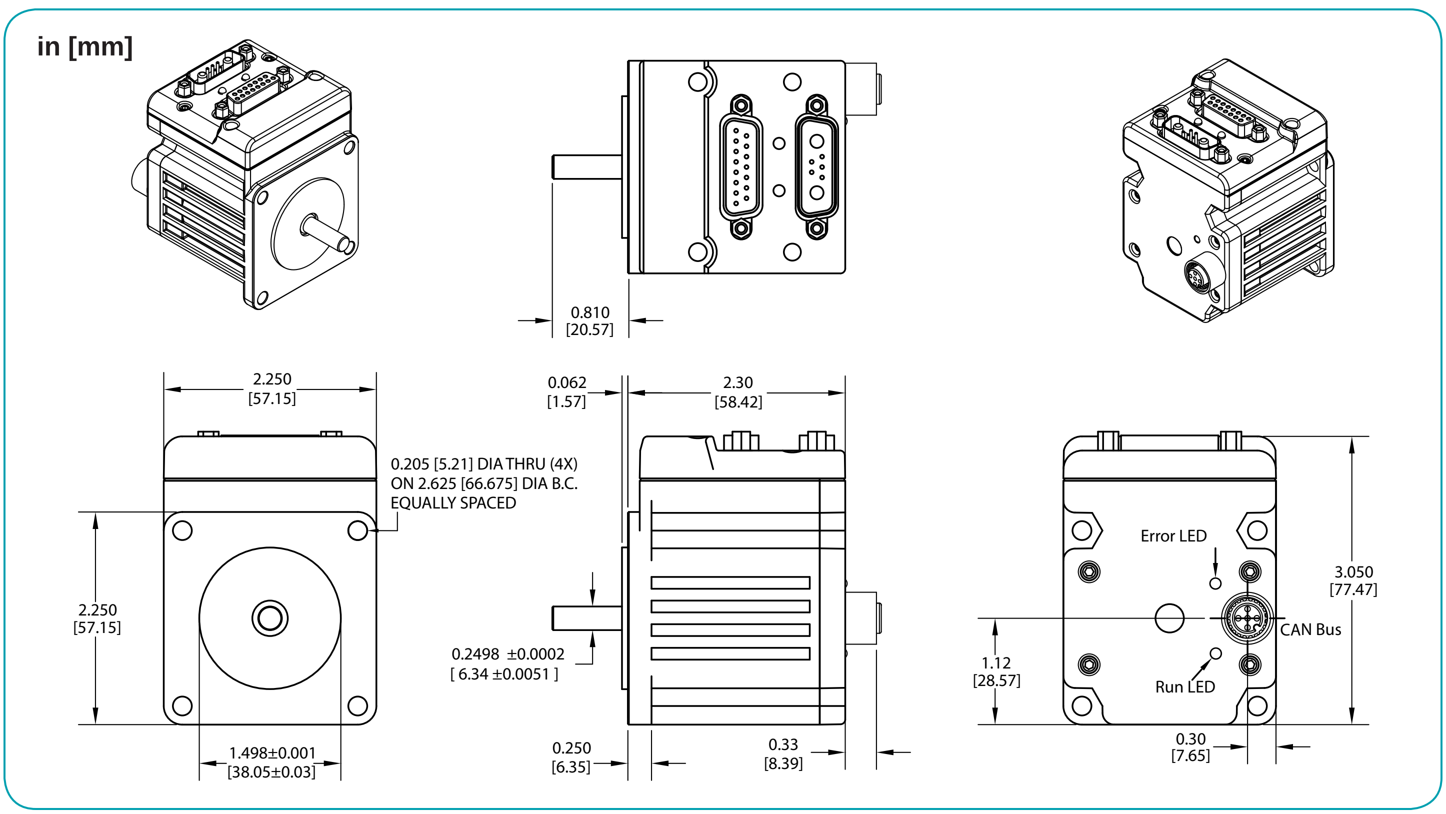
Motors were operated in Trap-Commutation Mode.

Please consult factory for Sine-Commutation Torque Curves.

Animatics SmartMotor™ SM23165D/DT-AD1

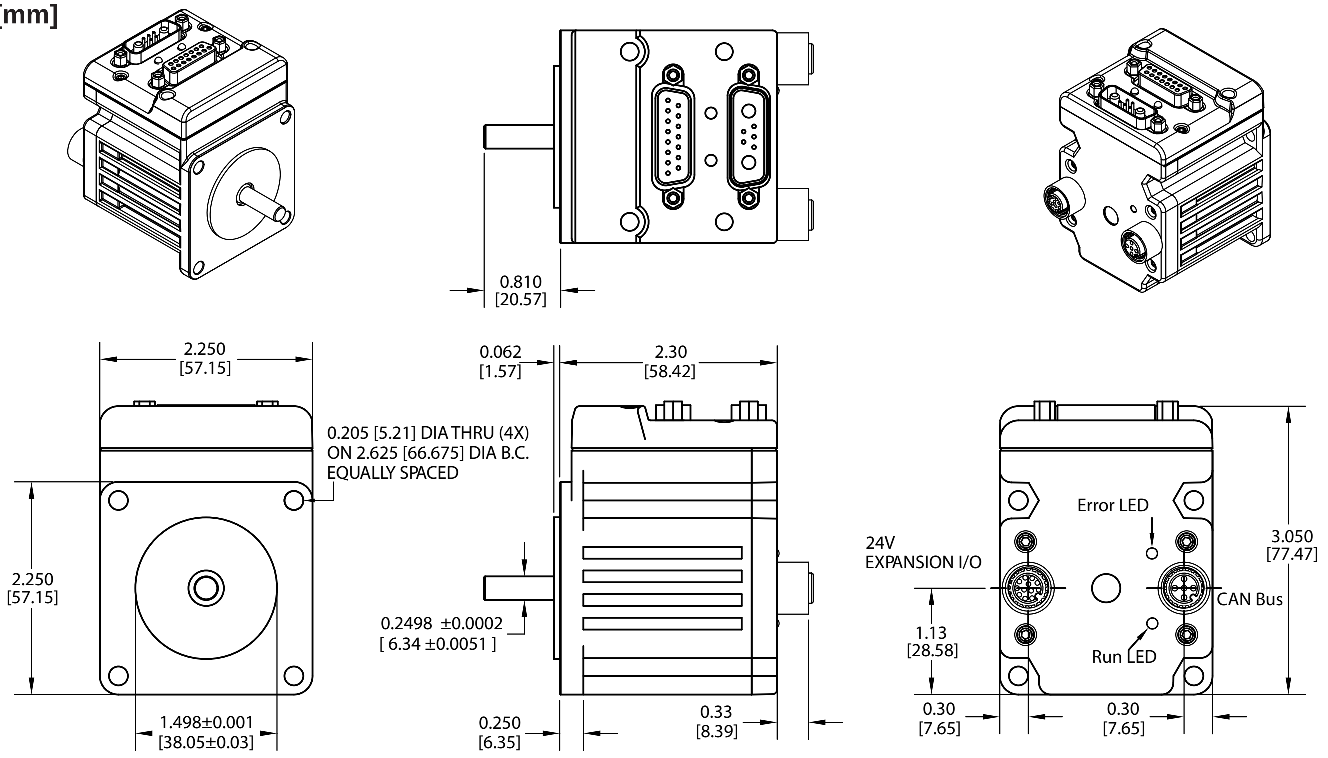


Animatics SmartMotor™ SM23165D/DT-C



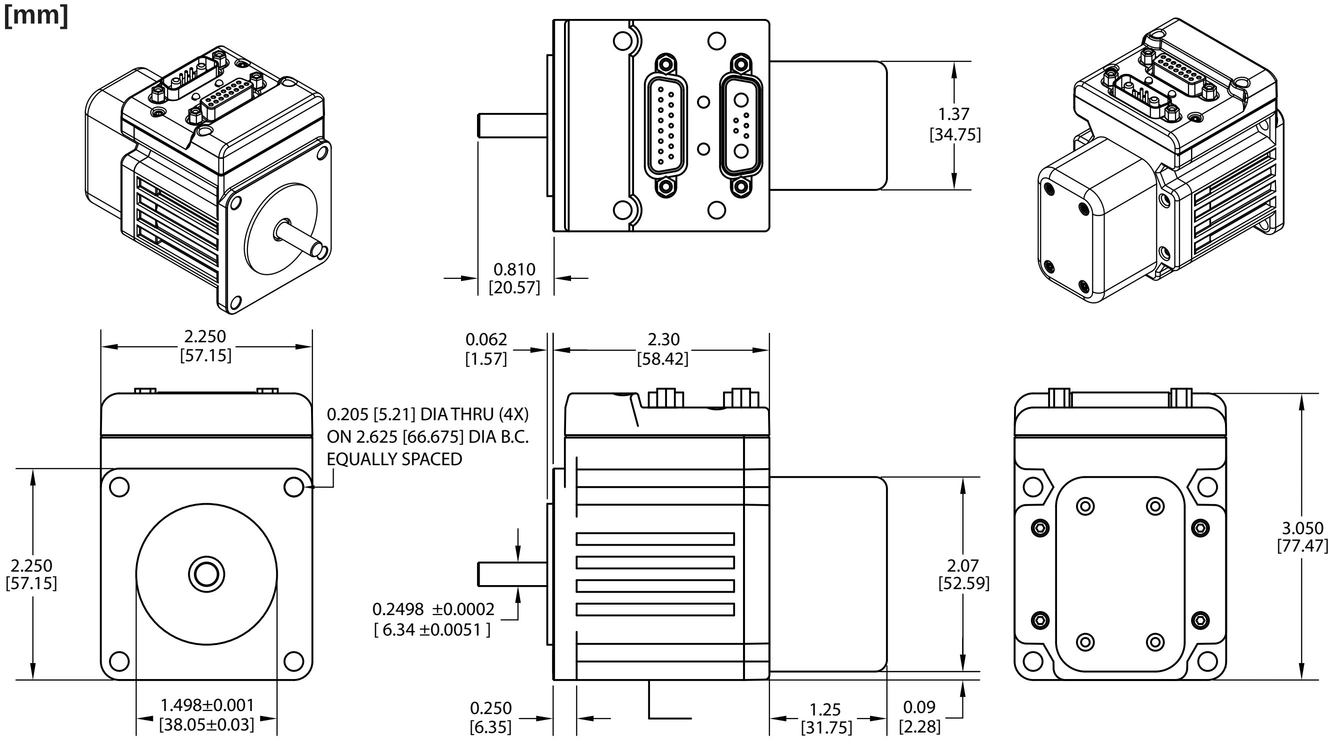
Animatics SmartMotor™ SM23165D/DT-C-AD1

in [mm]

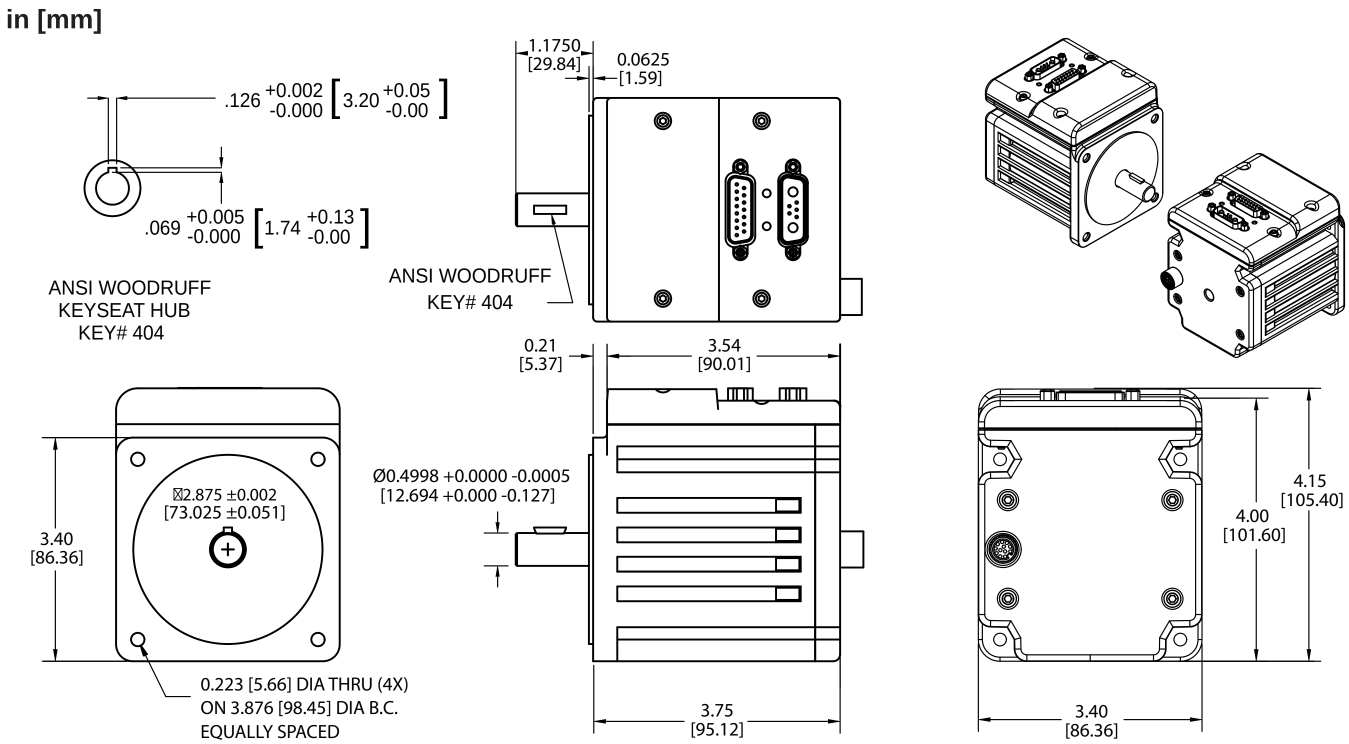


Animatics SmartMotor™ SM23165D/DT-BRK

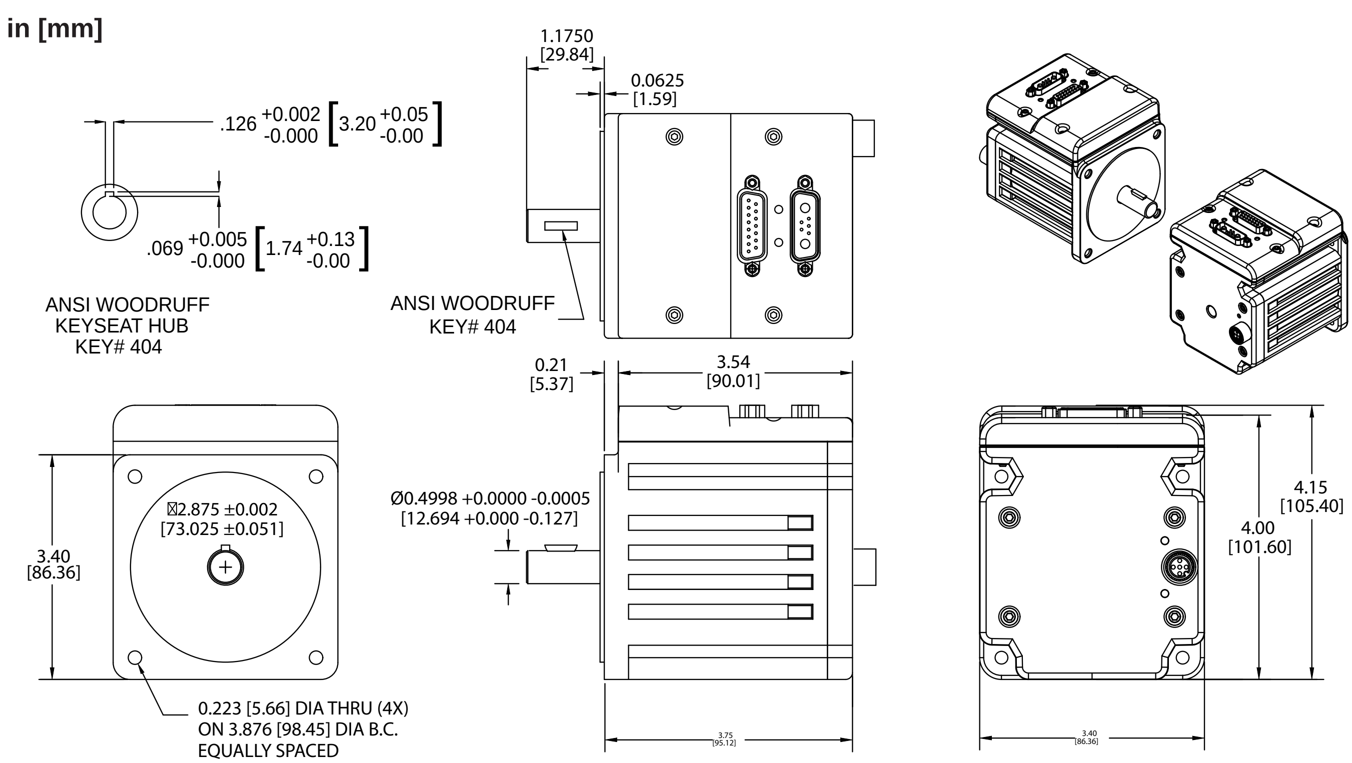
in [mm]



Animatics SmartMotor™ SM34165DT-AD1

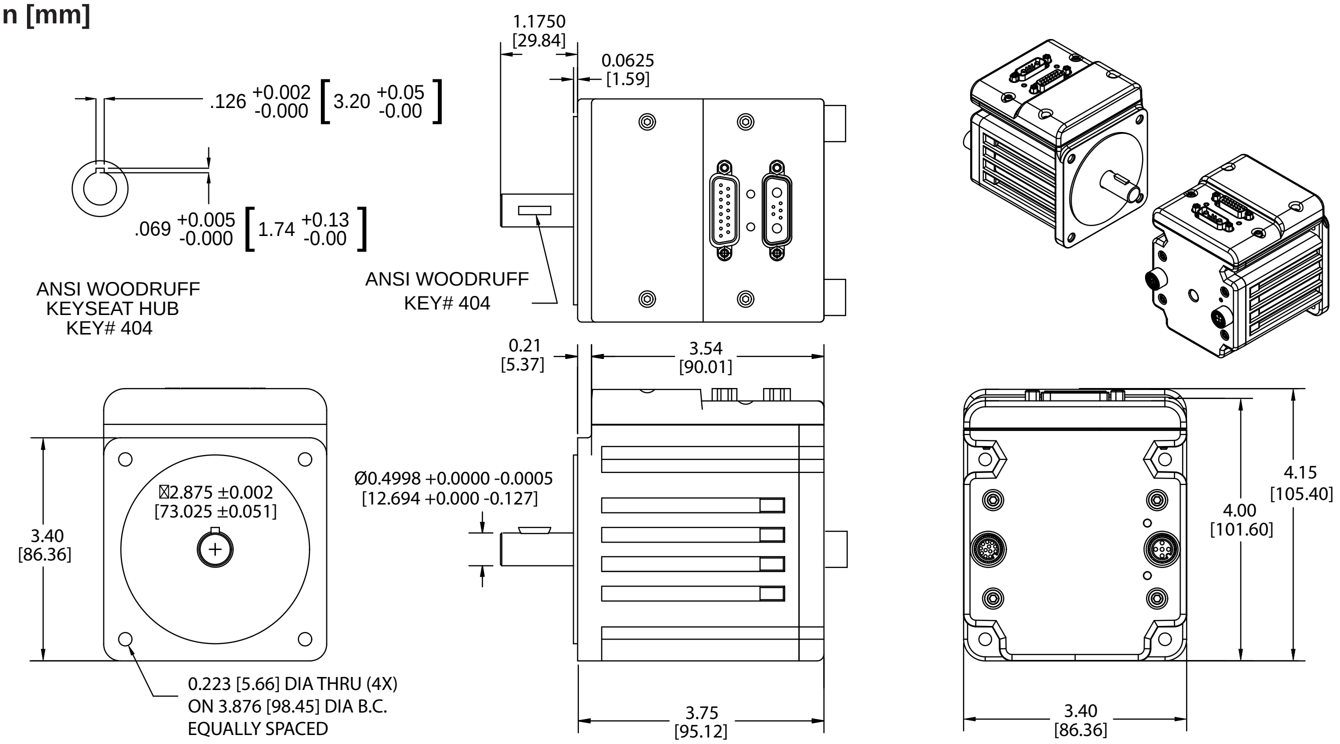


Animatics SmartMotor™ SM34165DT-C



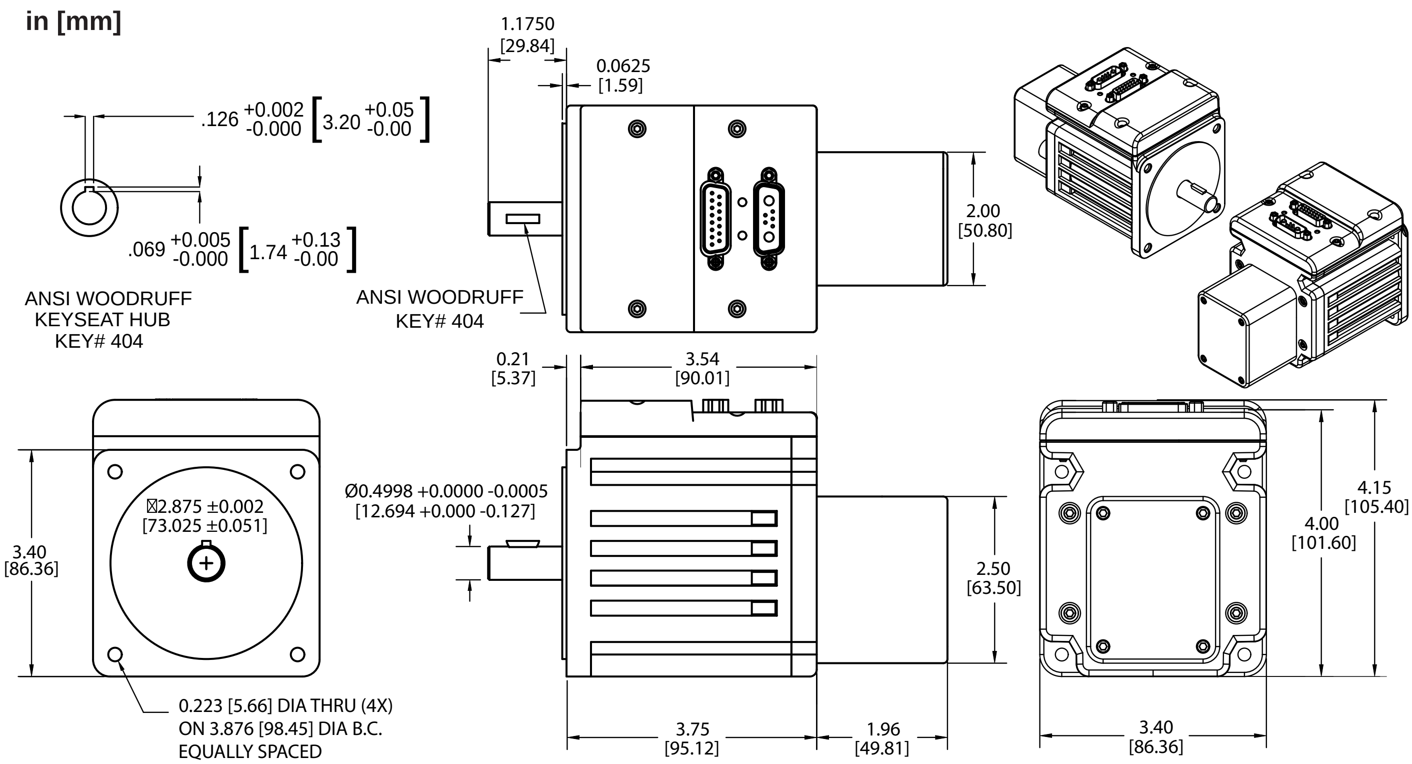
Animatics SmartMotor™ SM34165DT-C-AD1

in [mm]

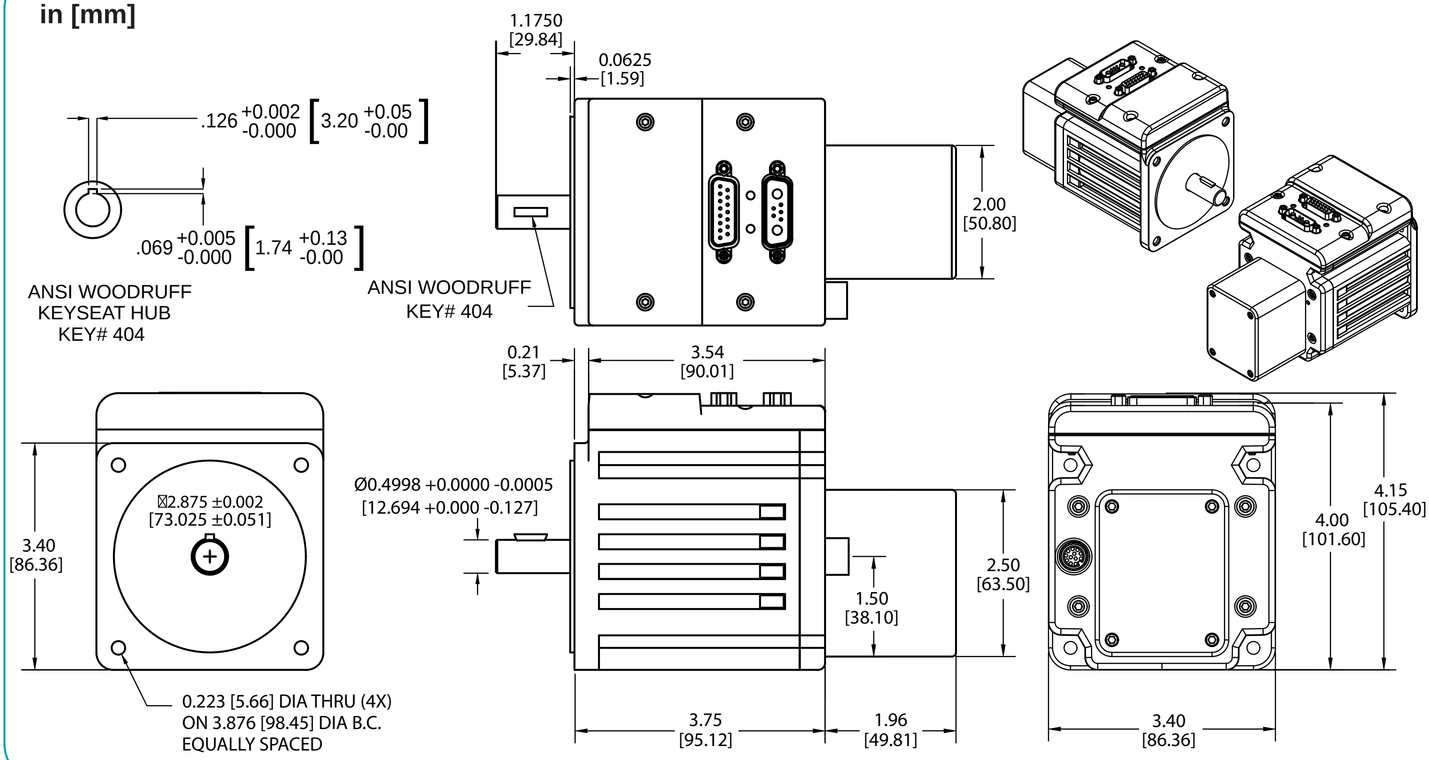


Animatics SmartMotor™ SM34165DT-BRK

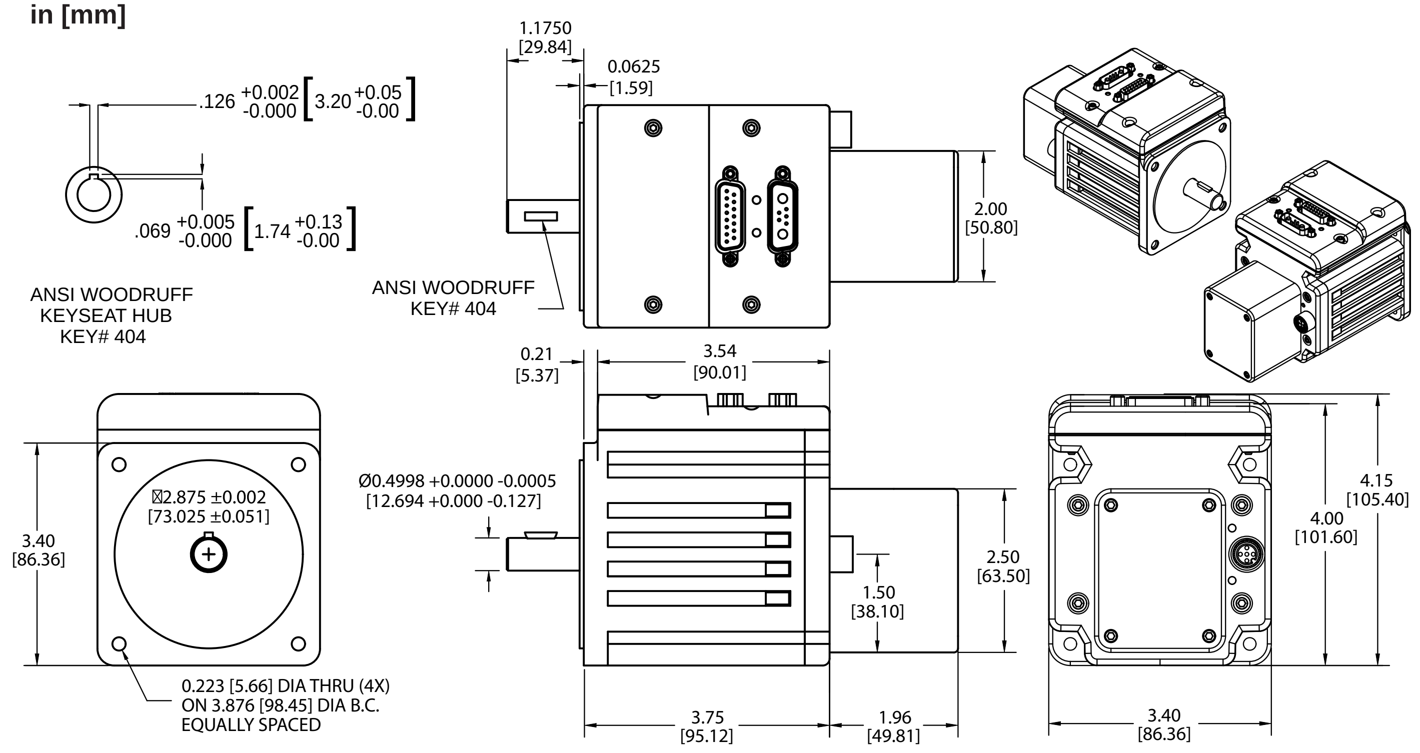
in [mm]



Animatics SmartMotor™ SM34165DT-BRK-AD1



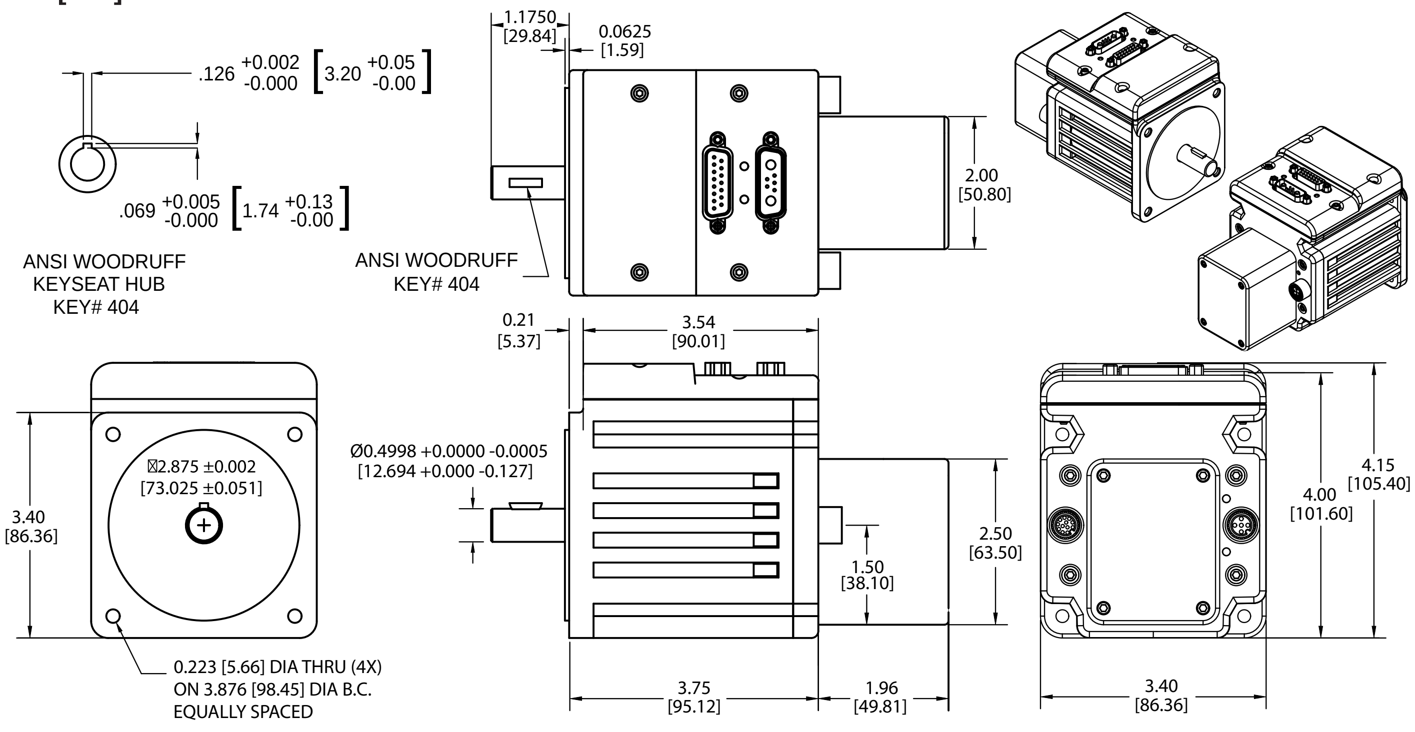
Animatics SmartMotor™ SM34165DT-BRK-C



Animatics Corporation
tel: 408.748.8721 • fax: 408.748.8725 • www.animatics.com

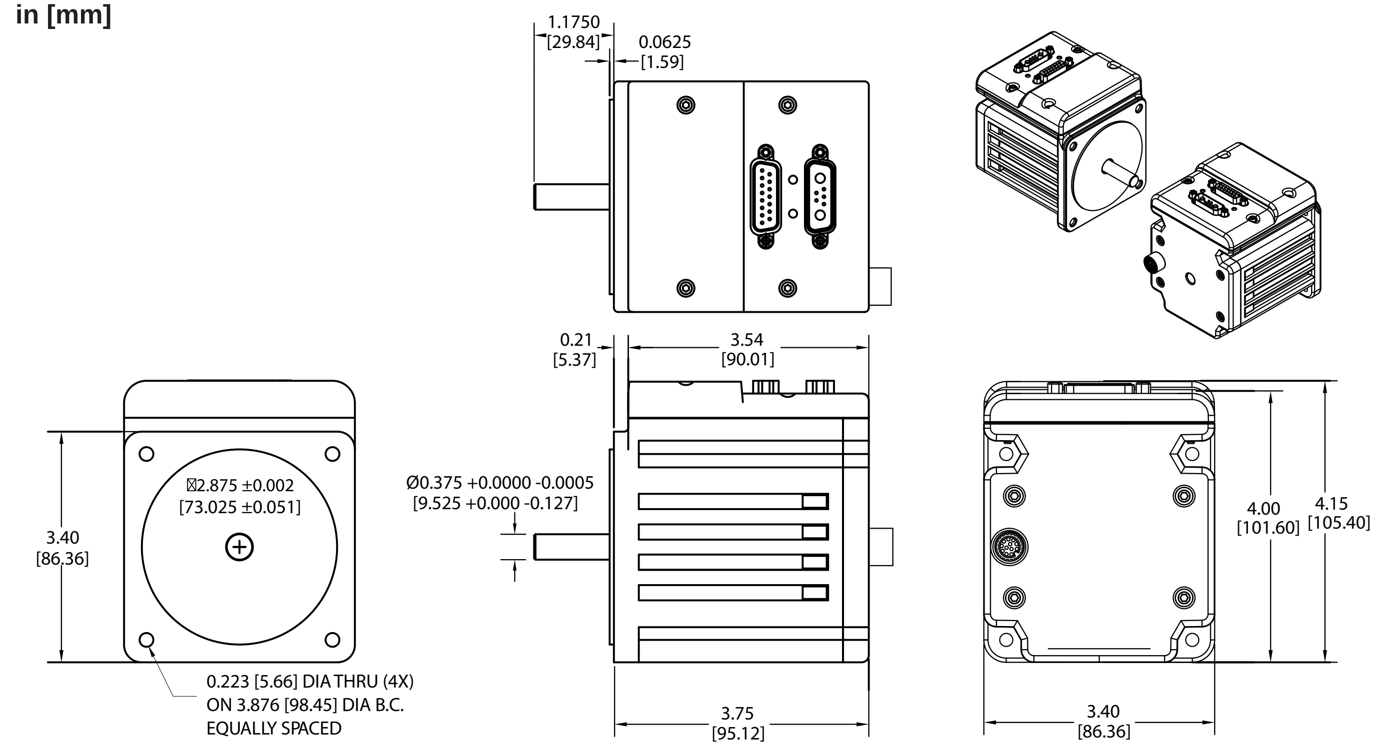
Animatics SmartMotor™ SM34165DT-BRK-C-AD1

in [mm]

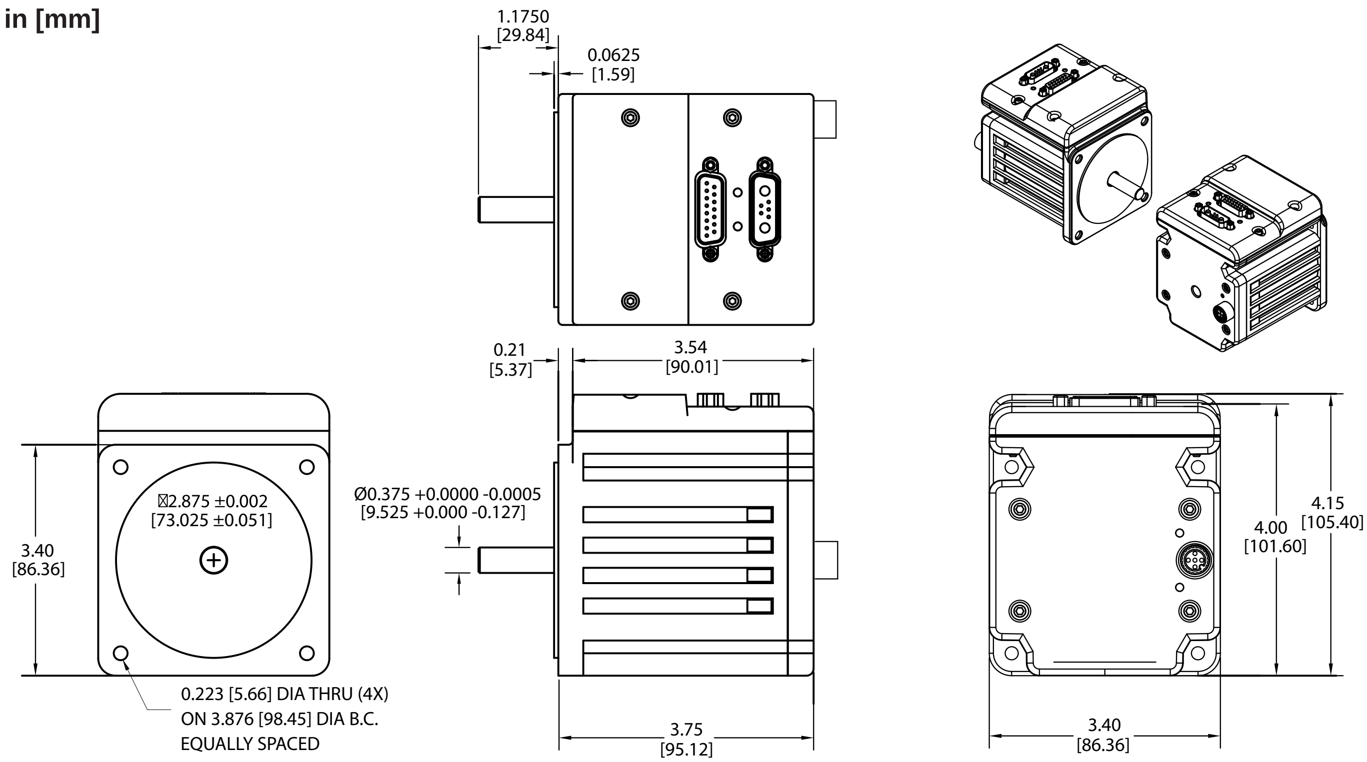


Animatics SmartMotor™ SM34165D-AD1

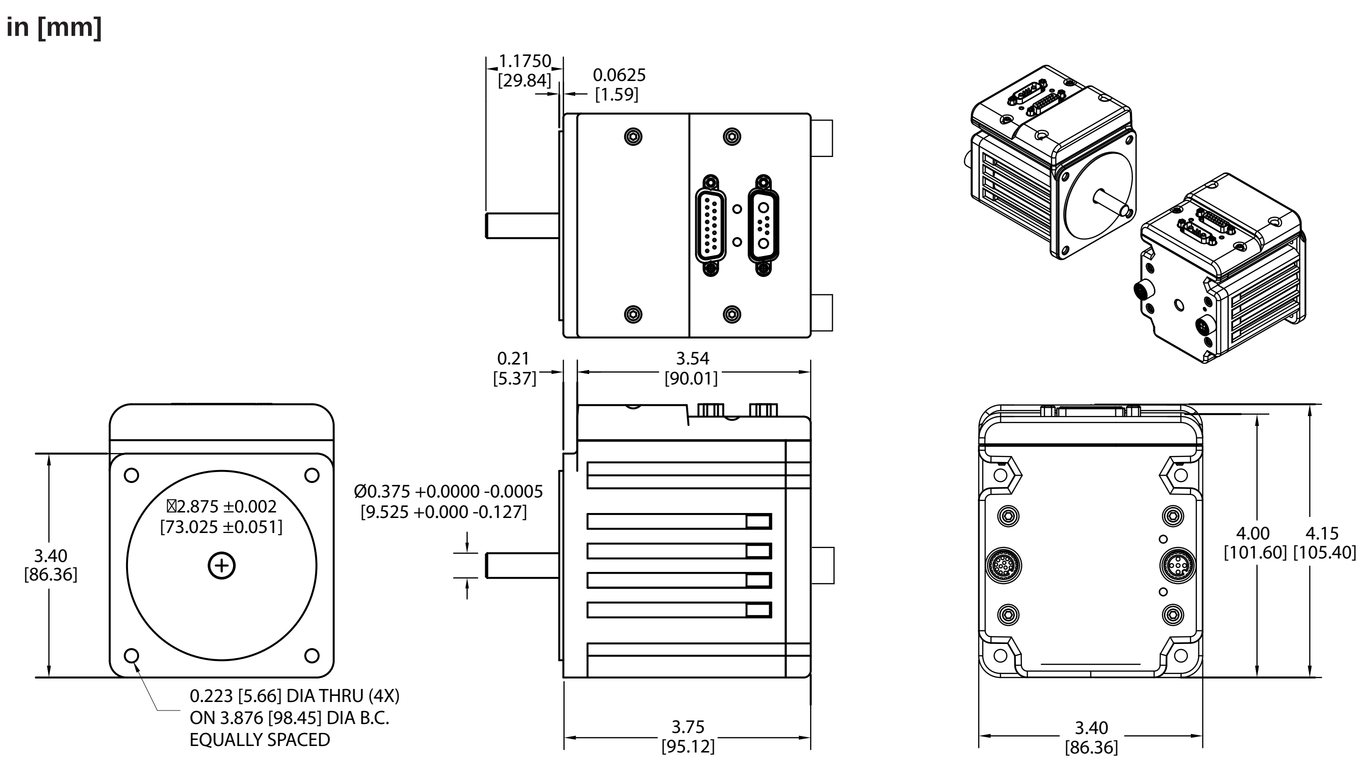
in [mm]



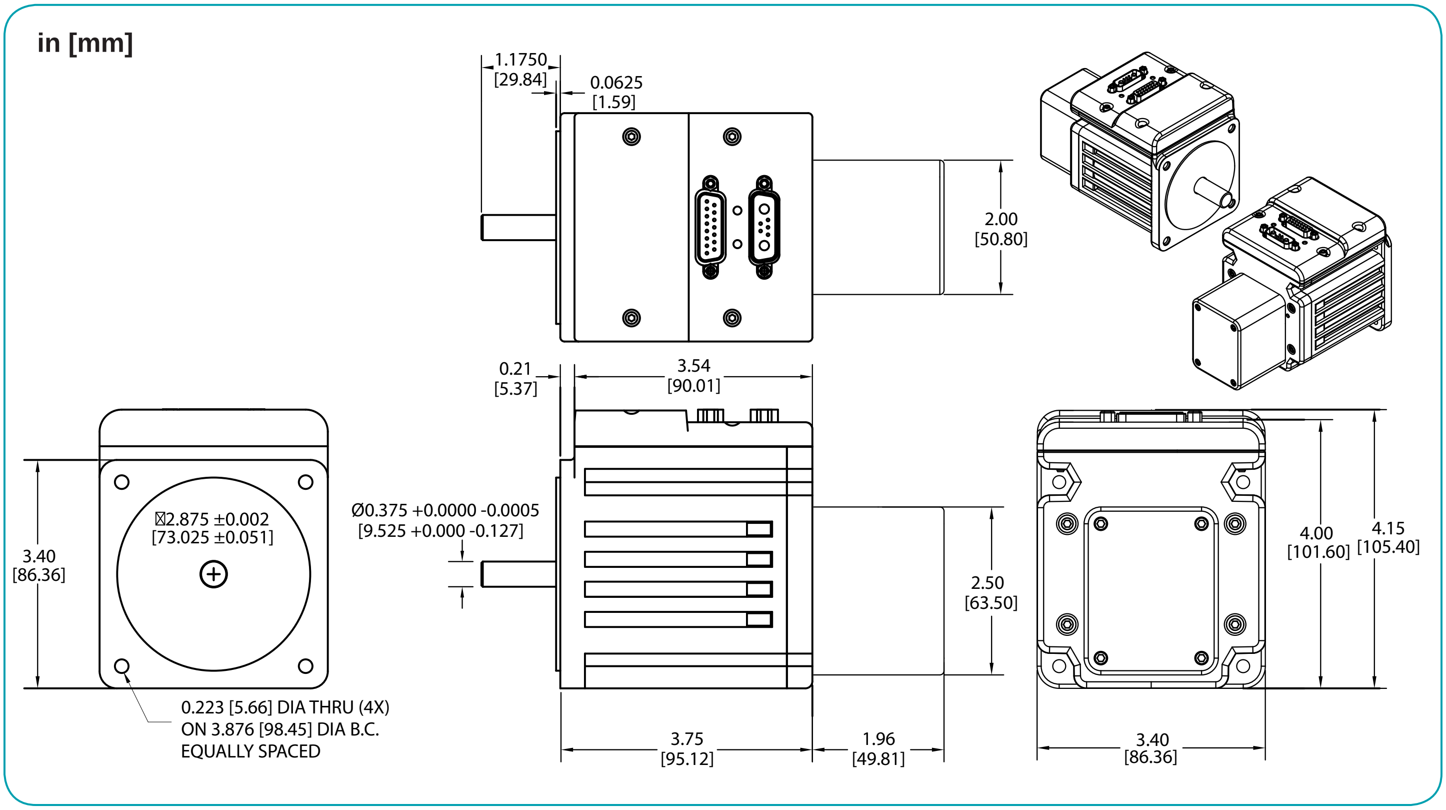
Animatics SmartMotor™ SM34165D-C



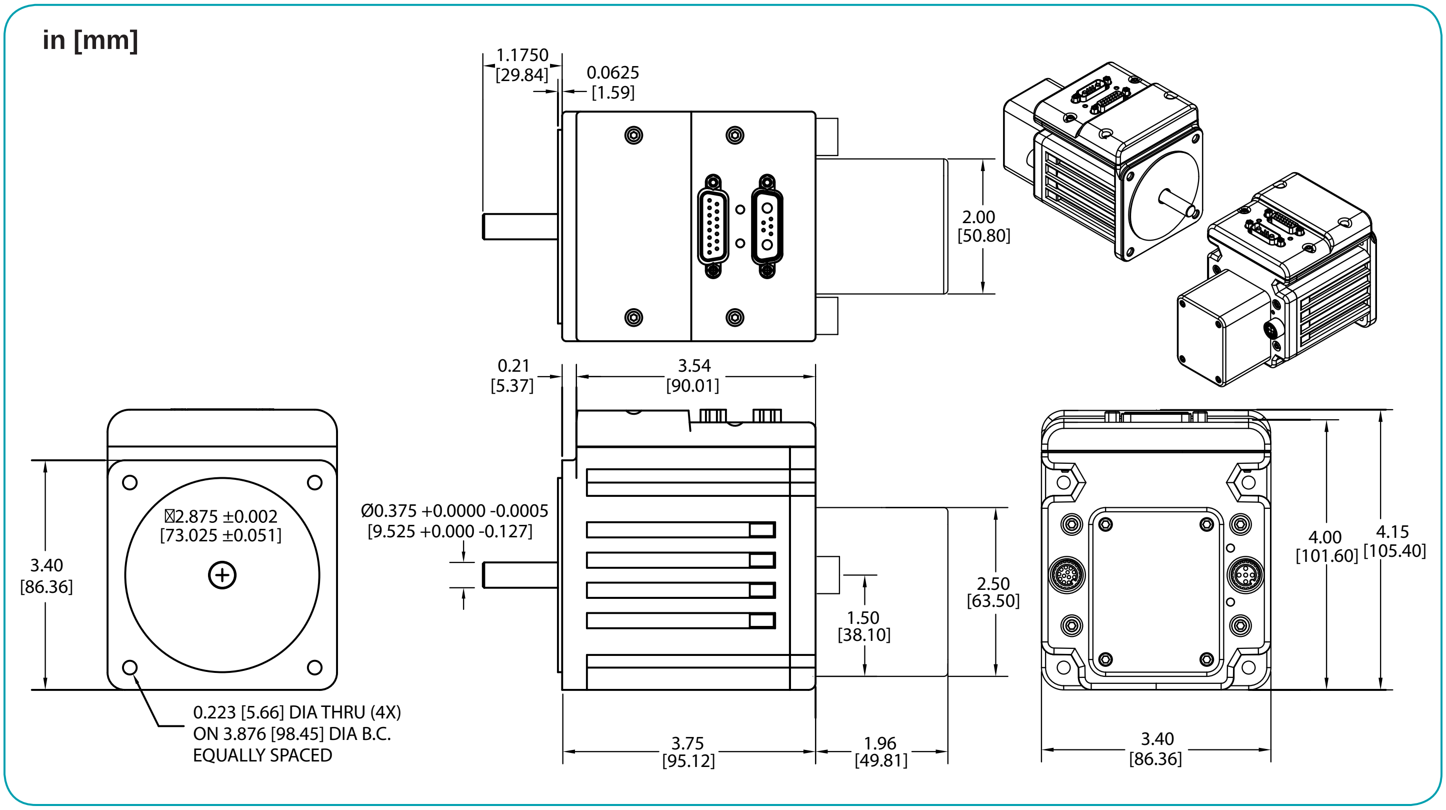
Animatics SmartMotor™ SM34165D-C-AD1



Animatics SmartMotor™ SM34165D-BRK

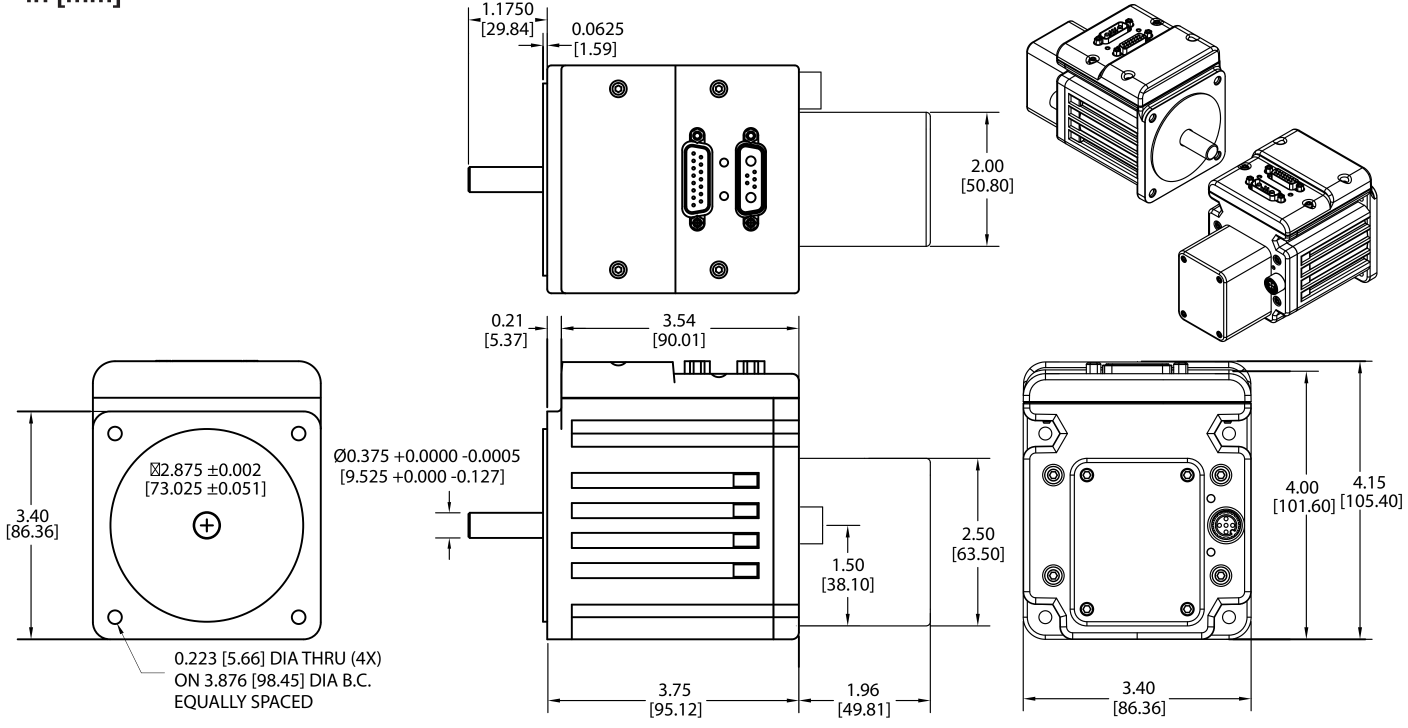


Animatics SmartMotor™ SM34165D-BRK-C-AD1



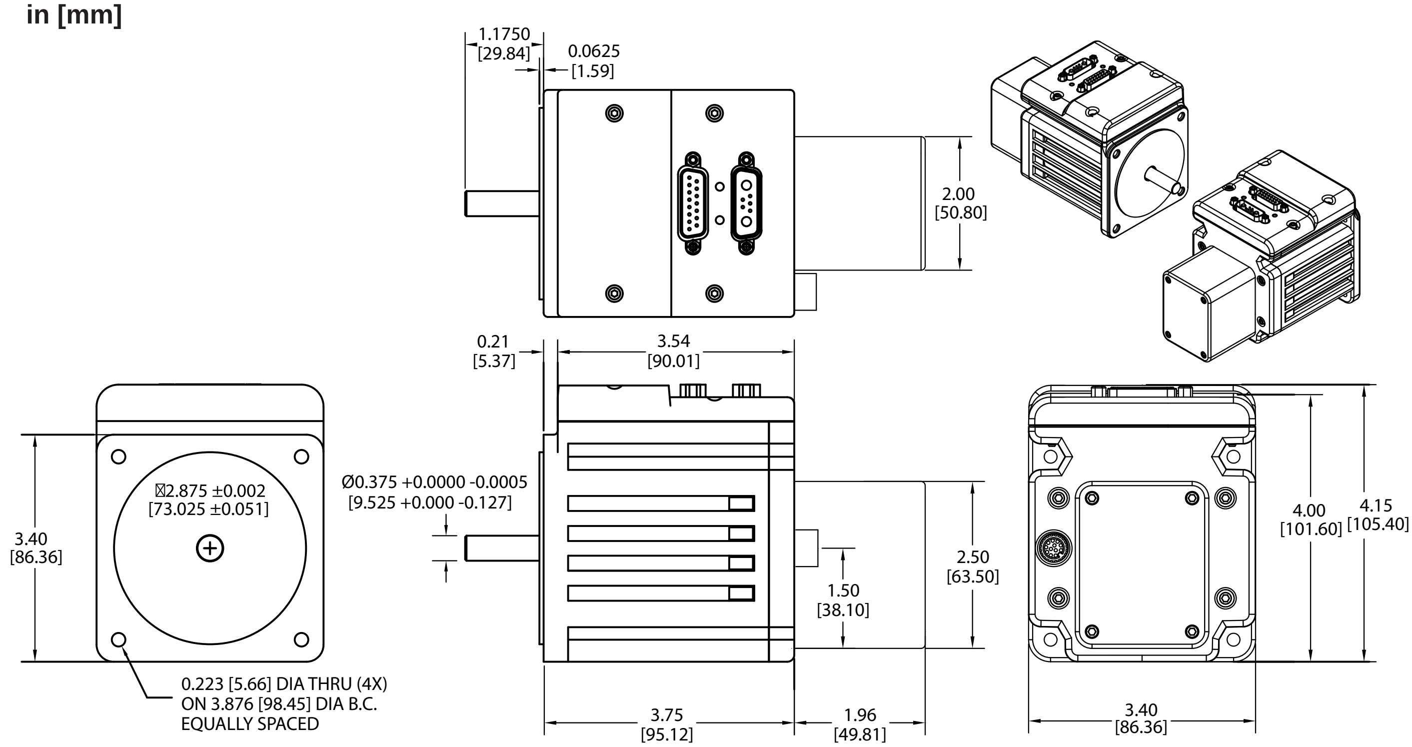
Animatics SmartMotor™ SM34165D-BRK-C

in [mm]



Animatics SmartMotor™ SM34165D-BRK-AD1

in [mm]

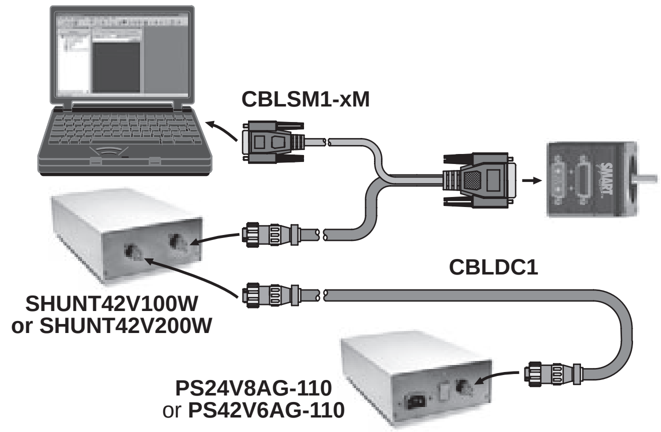


Introduction to Connectivity

Power:

Each SmartMotor™ is operated from 24 to 48VDC. Some of the larger SmartMotors™ can draw high current.

It is highly recommended to use heavy gage wire to connect the larger motors. As a result, the “Add-A-Motor” is recommended for the 17 and 23 frame series only.



Communications:

Each SmartMotor™ has a primary RS-232 serial port and a secondary RS-485 port by re-assignment of ports E and F of the 7 I/O points. Up to 100 SmartMotors may be separately addressed and are identifiable on either RS-232 or RS-485.

The most common and cost effective solution is typically RS-232 serial communications. Under this structure, each motor is placed in an electrical serial connection such that the transmit line of one motor is connected to the receive line of the next. Each motor will be set to “echo” the incoming data to the next motor down with approximately 1 millisecond propagation delay. There is no signal integrity loss from one motor to the next, which results in highly reliable communications.

The following cables/devices are used for RS-232 and Power connectivity:

CBLPWRCOM2-xM	Power and communications cable with flying leads or in conjunction with DIN-RS232 8 channel isolated communications board
CBLSM1-xM	Power and Communications cable with DB-9 serial connector and power supply connector that fits our enclosed power supplies
CBLSM1-DEMO	Testing cable used with our PWR116 “laptop” type power supply
CBLSM1-x-y-z	Custom length multi-drop RS-232 daisy chain cable

The following cables are used for RS-485 and Power connectivity:

RS485-ISO	Converts primary RS-232 to isolated RS-485 (Note: uses Port G I/O pin)
CBLSM2-x-y-z	Custom multi drop isolated RS-485 (multiple RS485-ISO adapters)

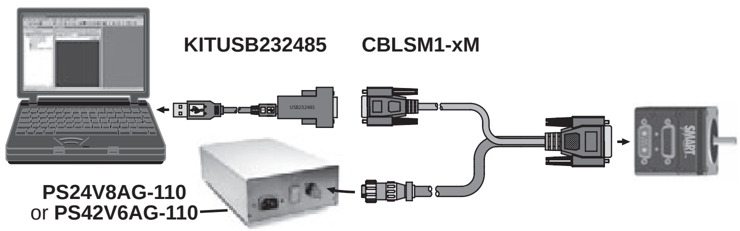
Interfacing with I/O devices:

Each SmartMotor™ has 7 TTL level user-configurable I/O. Each can be used as either inputs or outputs. The following is a quick review of I/O interfacing connectivity options:

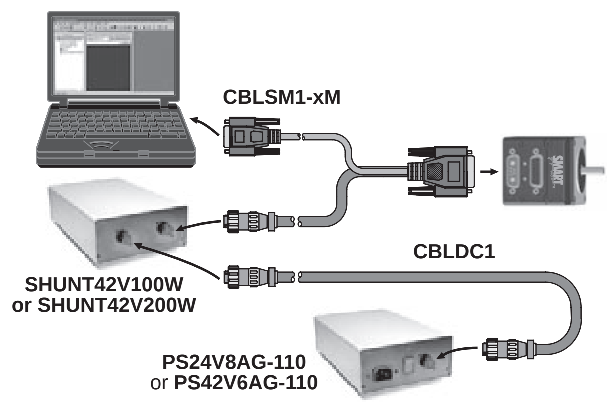
CBLIO5V-xM	Direct connection to 5VTTL I/O
CBLIO5V-xM via OPTO2	24VDC DC isolation and conversion of 5V signals
CBLIO5V-xM via DINIO7	Motor Breakout board to industry standard OPTO relays
CBLIO-ISO1-xM	Isolated 24VDC logic conversion cable

The following pages are a roadmap to motor connectivity. These pages show the physical layout of how cables are used including power, communications and I/O interconnection.

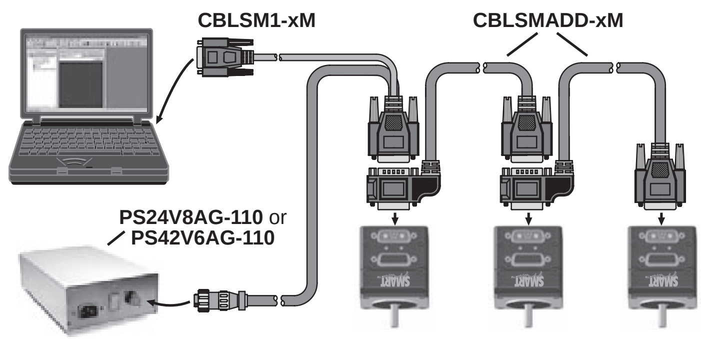
RS-232 Communications Using USB Adapter



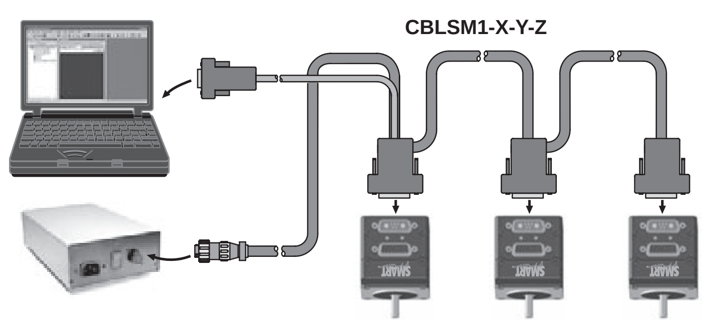
RS-232 Communications with Power Supply & Protective Shunt



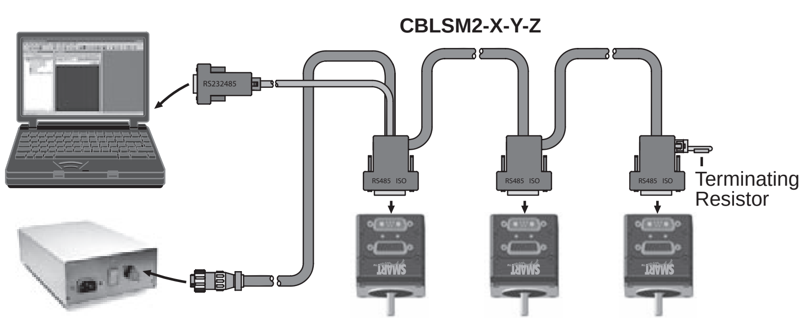
RS-232 Multidrop using Add-A-Motor™ Cables

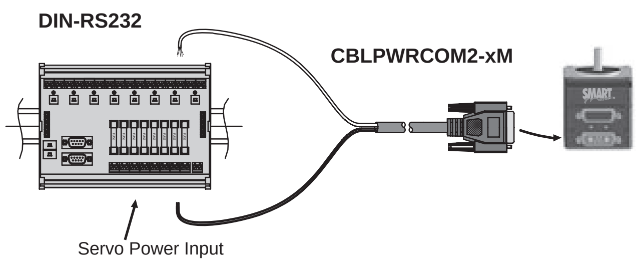


RS-232 Multidrop using Custom Order Cable



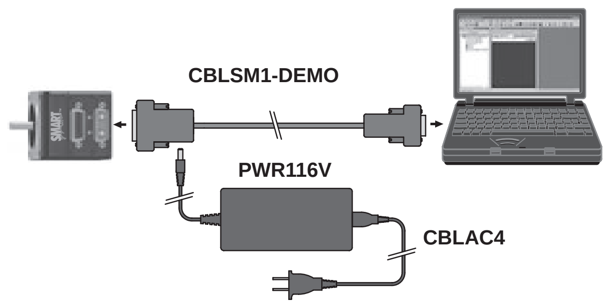
RS-485 Isolated Communications Using Custom Order Cable

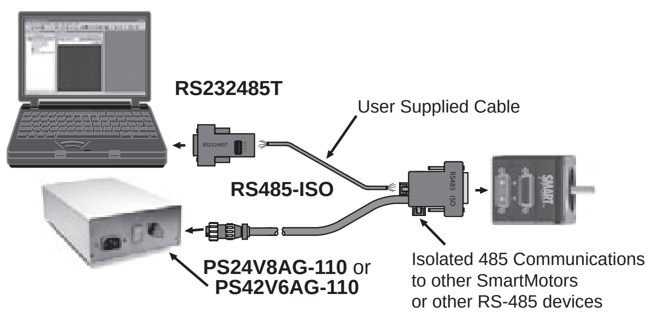




Isolated RS-232 Communications for up to 8 SmartMotors™

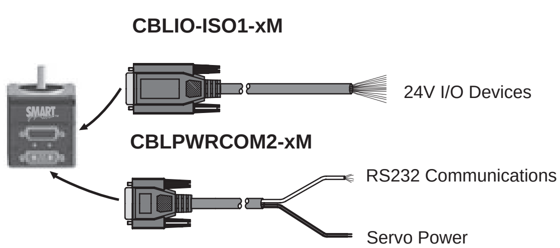
Demonstration & Development Configurations

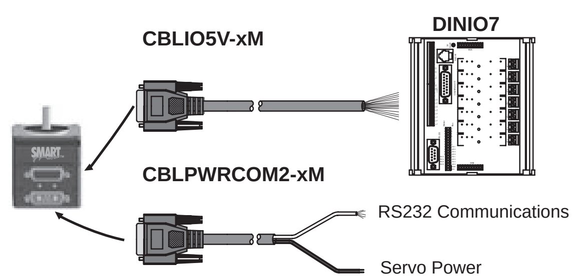


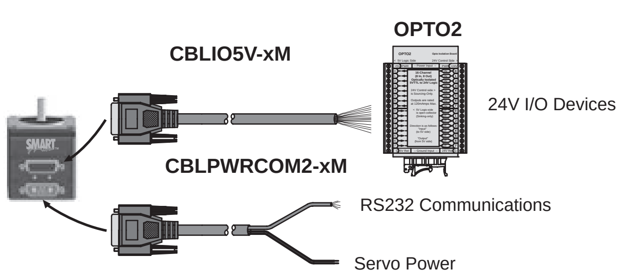


Isolated RS-485 Communications

Interfacing with 24V I/O Devices



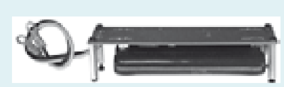




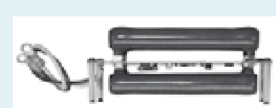
Recommended Power Supply

PS42V20AF series

Recommended Shunts



SHUNT42V100W-OF



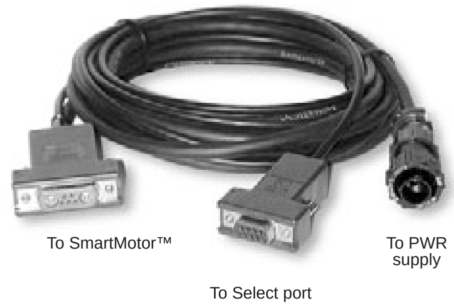
SHUNT42V200W-OF

CBLSM1-3M

Power and Communications Cable for Main 7W2 Connector on Animatics SmartMotor™

CBLSM1 series is the main power and communications cable consisting of a 7W2 main motor connector split out to a pre-wired RS-232 DB-9 connector to plug directly into any standard PC serial port.

The power is split off and has a connector that plugs into our enclosed frame power supplies on page 88.



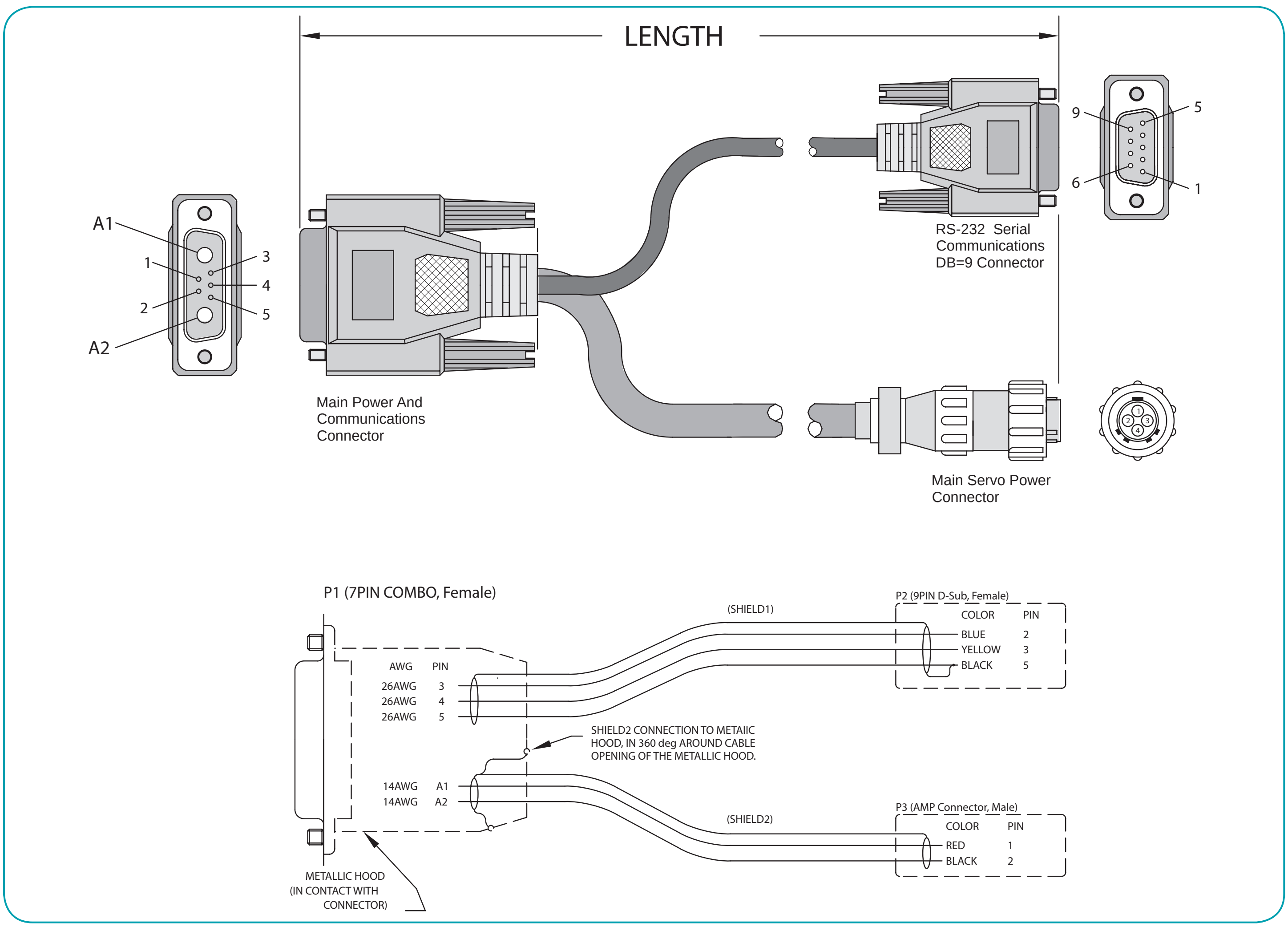
Standard Length

Part Number	Length
CBLSM1-3M	3 meters
CBLSM1-10M	10 meters

Custom Length

Part Number	Length
CBLSM1-x	x (in feet)

Note: Communications Shield is connected at the DB-9 end, but NOT the Motor end. The power cable is connected at the motor connector shell but electrically isolated from the any internal electronic components.



CBLSMADD-xM (Animatics™ "Add-A-Motor™" Cable)

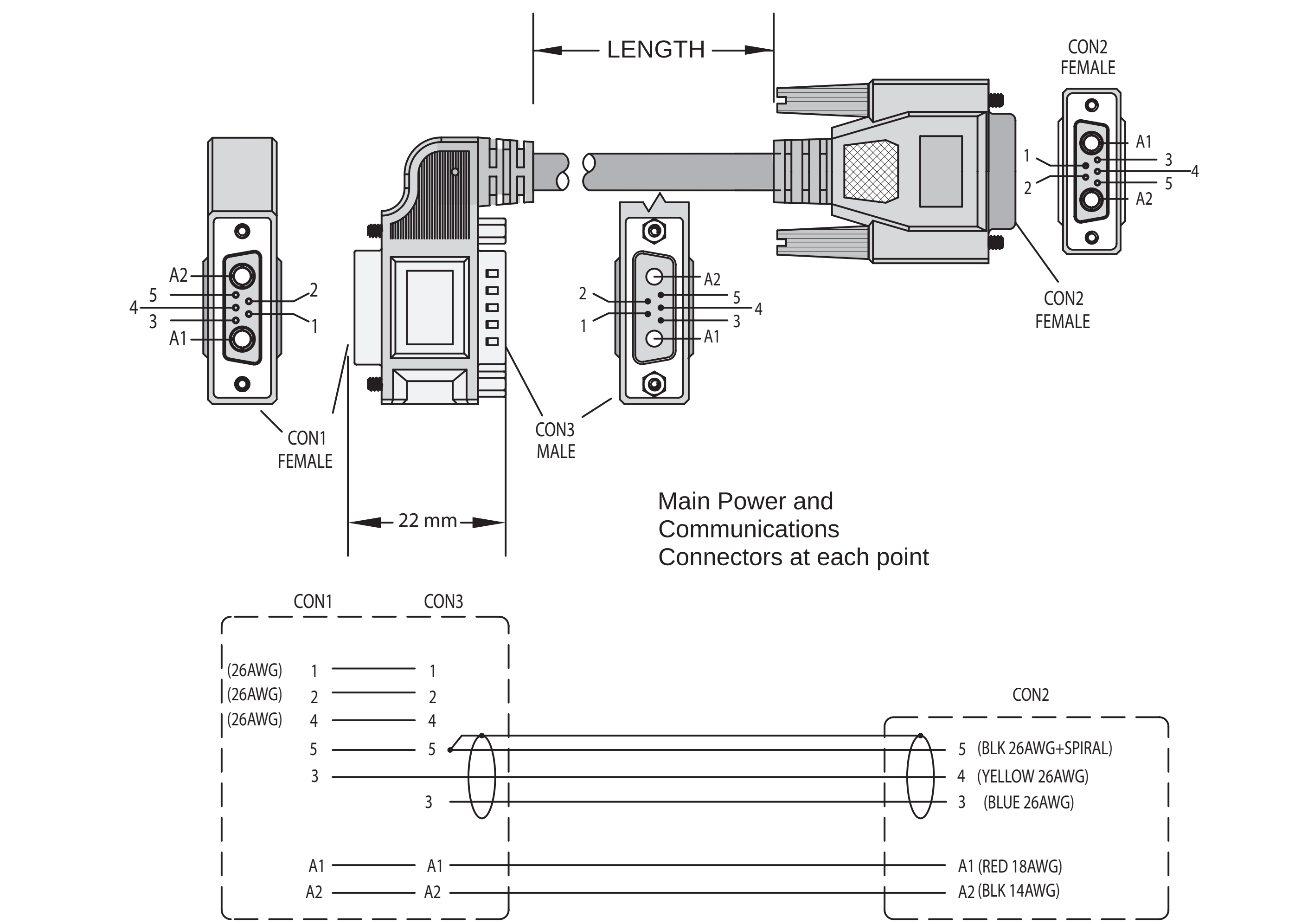
Power and Communications Daisy Chain Cable for networking Power and Communications to multiple Animatics SmartMotors.

CBLSMADD series is the main power and communications cable consisting of a feed-thru 7W2 main motor connector split out to a single second motor 7W2 connector.

The cable is designed to allow ease of connection to multiple motors in a single RS-232 serial daisy chain network. The Main Power Ground wire is of a larger gauge to decrease noise emissions at the ground-plane level. The RS-232 Communications lines are internally shielded from the power lines.

Part Number	Length
CBLSMADD-0.3	0.3 meters
CBLSMADD-1.0	1 meters
CBLSMADD-3.0	3 meters
CBLSMADD-7.5	7.5 meters

Note: Due to gauge of the main power lines, it is not recommended to use the "Add-A-Motor" cables with the larger 34 frame SmartMotors. If there is just one 34 frame SmartMotor in a system design, then it should be the first motor in the chain so as to minimize voltage drop effects over the entire network.

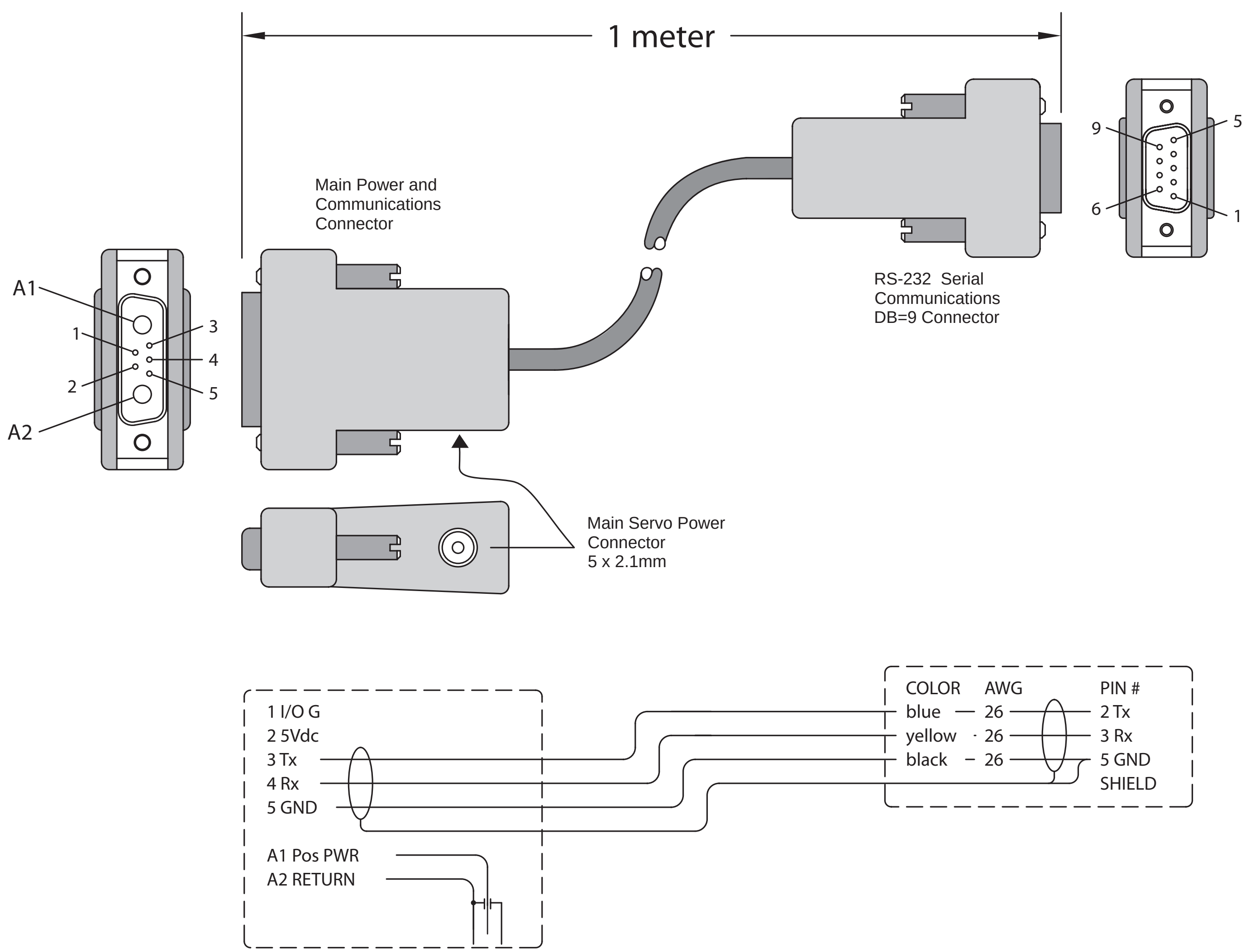


CBLSM1-DEMO

Training/Testing Power and Communications Cable for Main 7W2 Connector on Animatics SmartMotor™.

CBLSM1-Demo cable is only available in a fixed length of ~1 Meter. Similar to the CBLSM1-xM series, it consists of a 7W2 main motor connector split out to a pre-wired RS-232 DB-9 connector to plug directly into any standard PC serial port. The power to the motor is provided via a single 5mm diameter 2.1 mm center pin DC connector.

This connector accepts our PWR116V 24VDC power supply.



CBLSM1-X-Y-Z (Animatics Custom Build-to-Order)

Custom Multi-Motor Power and Communications Daisy Chain Cable for networking Power and Communications to Multiple Animatics SmartMotors.

These cables are made to order where:

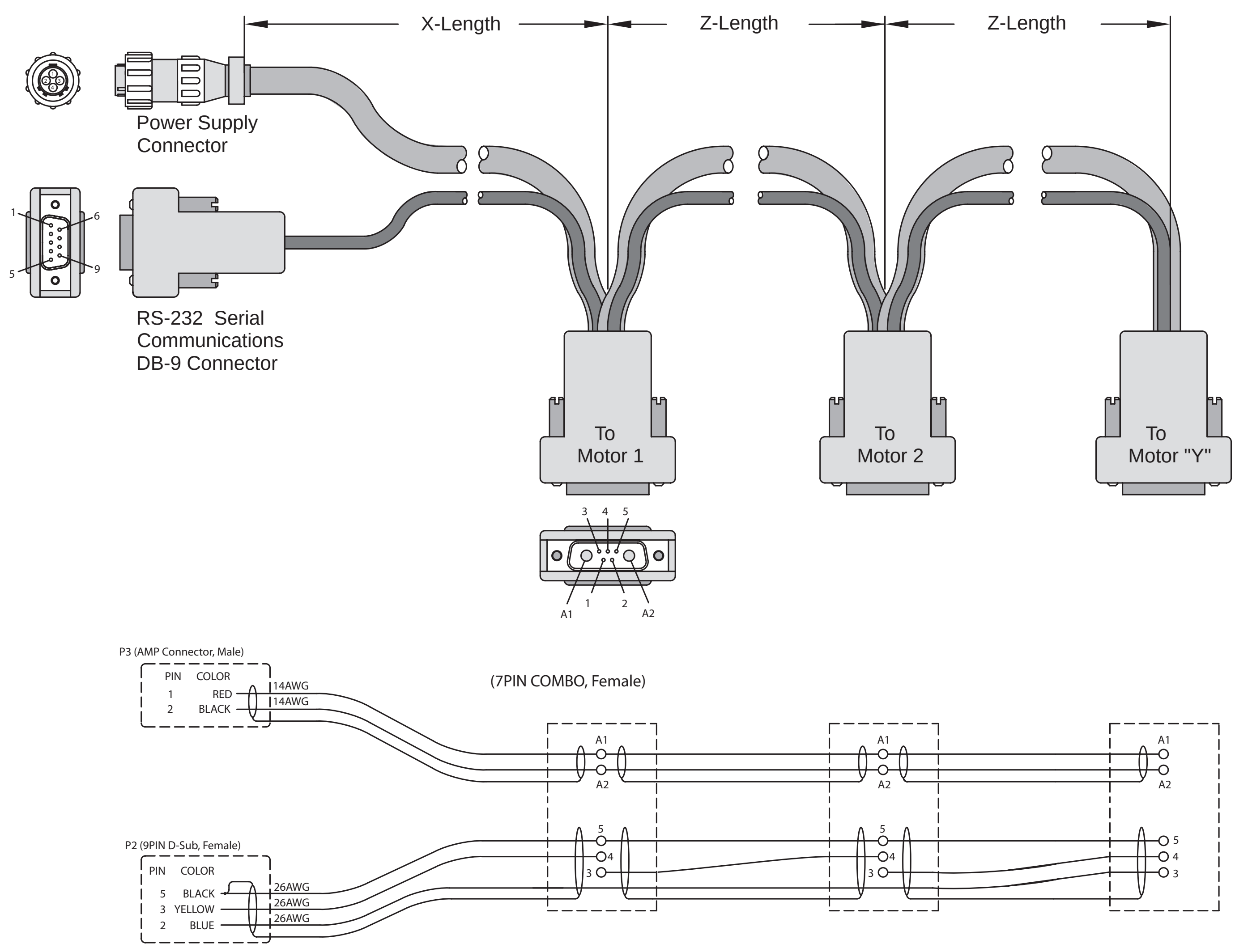
X = cable length in Feet from the first motor to the Power and Serial connectors
Y = Number of Motors
Z = distance in feet from one motor to the next

Note: This part numbering system does not allow for different length between each motor daisy chain network.
The RS-232 Communications lines are in a separate shielded cable from the main power cable for optimum noise immunity.

Example:

CBLSM1- 10- 3- 5 would give you a 3-motor cable with 10 feet to the first motor and 5 feet between each motor.

Example of 3-Motor CBLSM1-X-Y-Z shown



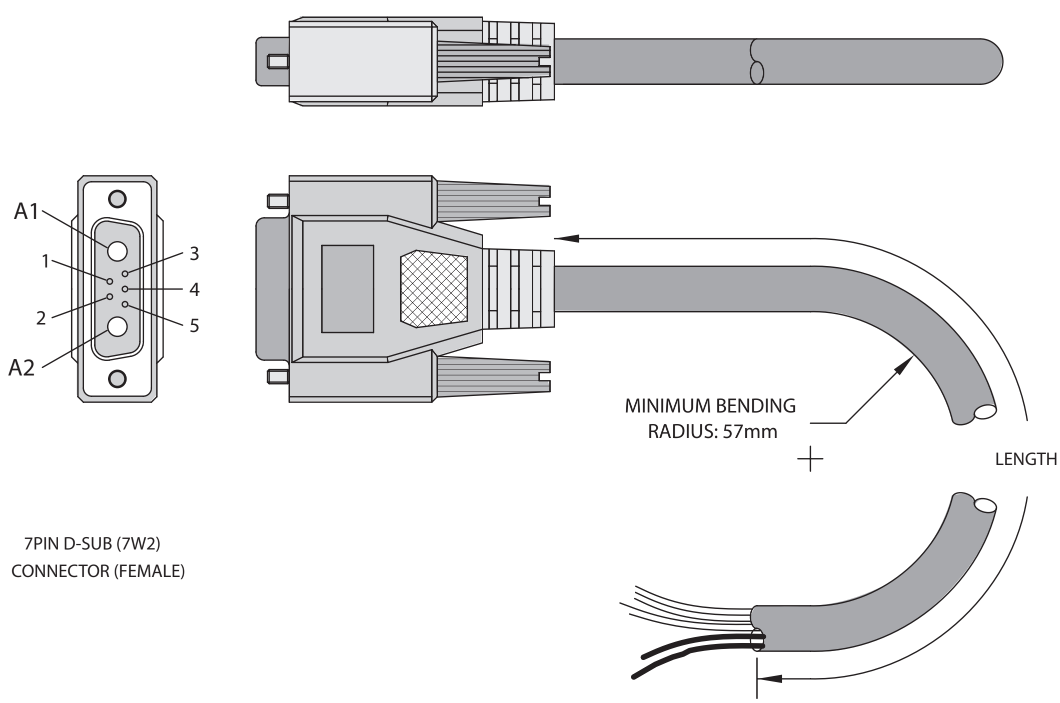
CBLPWRCOM2-xM

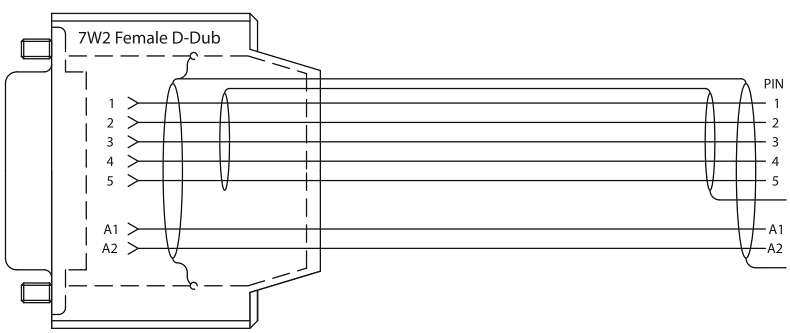
Power and Communications Cable (Flying Leads) for Main 7W2 Connector on Animatics SmartMotor™.

CBLPWRCOM2 series a Power and communications cable consisting of a 7W2 main motor connector with communications internally shielded from power and a full shield over entire length terminating at a metal jacket inside the over-molded connector.

Part Number	Length
CBLPWRCOM2-3M	3 meters
CBLPWRCOM2-5M	5 meters
CBLPWRCOM2-10M	10 meters

Note: Communications Shield is connected at the DB-9 end, but NOT the Motor end. The power cable is connected at the motor connector shell but electrically isolated from the any internal electronic components.





PIN	FUNCTION	COLOR	AWG
1	G-SYN	WHITE	26
2	5Vdc	RED	26
3	RS232 Tx	BLUE	26
4	RS232 Rx	YELLOW	26
5	SIG GND	BLACK	26
	SHIELD1		
A1	PWR	RED	14
A2	PWR GND	BLACK	14
	SHIELD2		

CBLSM2-X-Y-Z (Custom Build-to-Order)

Isolated RS-485 Multi-Drop Custom Cable

This cable makes use of the RS232485 converter at the host and a single RS485-ISO adapter at each motor.

The adapters have power hard wired and RS-485 wired together via jumper cables with a 4 pin G-grid Molex connect at each end.

As a result, it is easy to add or remove a given motor on the bus for setup and troubleshooting.

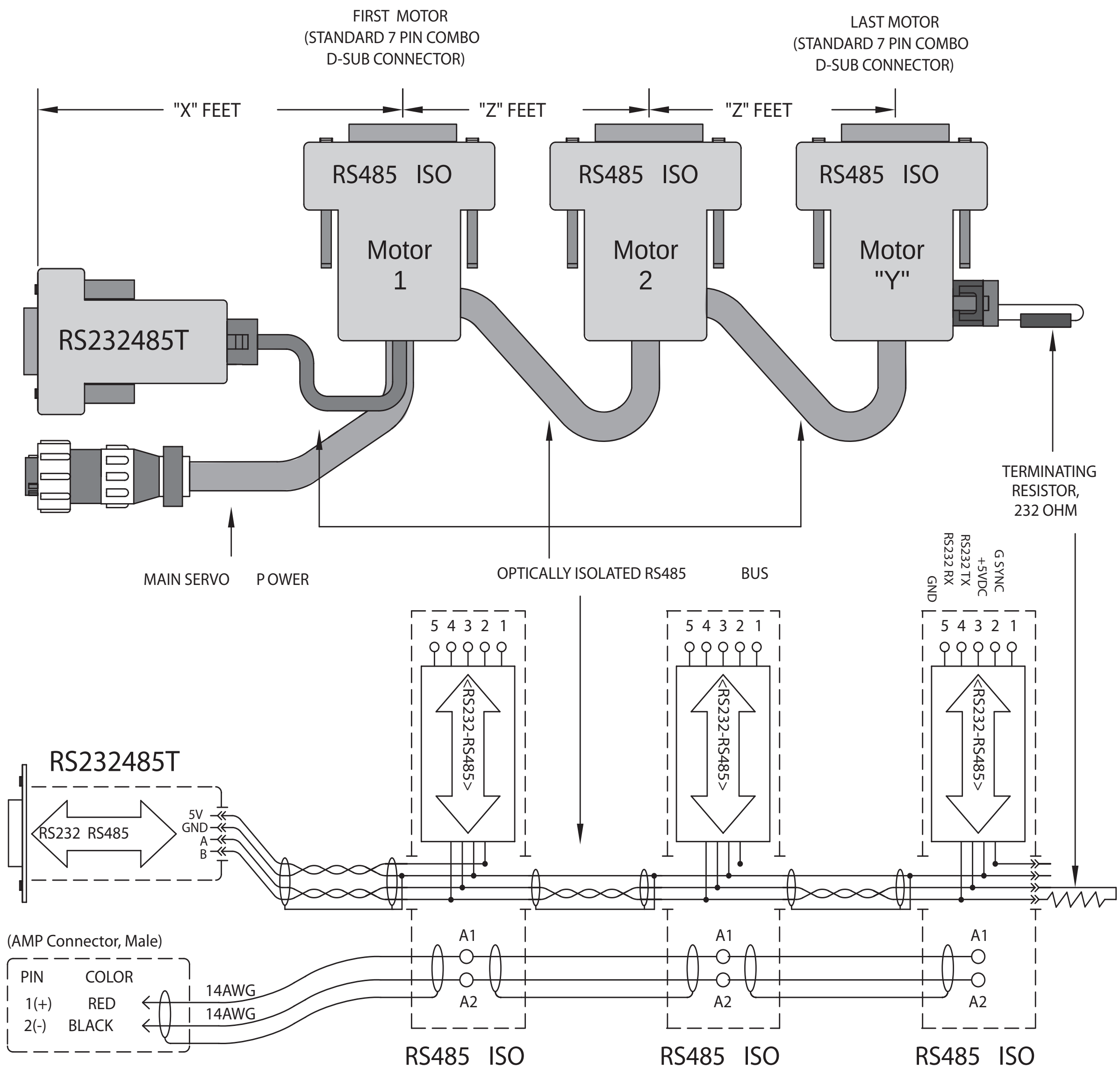
Since the RS485-ISO adapters are used, the entire Bus is isolated and shielded for maximum noise immunity in electrically harsh environments.

These cables are made to order where:

X = cable length in Feet from the first motor to the Power and Serial connectors
Y = Number of Motors
Z = distance in feet fro one motor to the next

Note: This part numbering system does not allow for different length between each motor daisy chain network.

Example:
CBLSM2- 10- 3- 5 would give you a 3-motor cable with 10 feet to the first motor and 5 feet between each motor



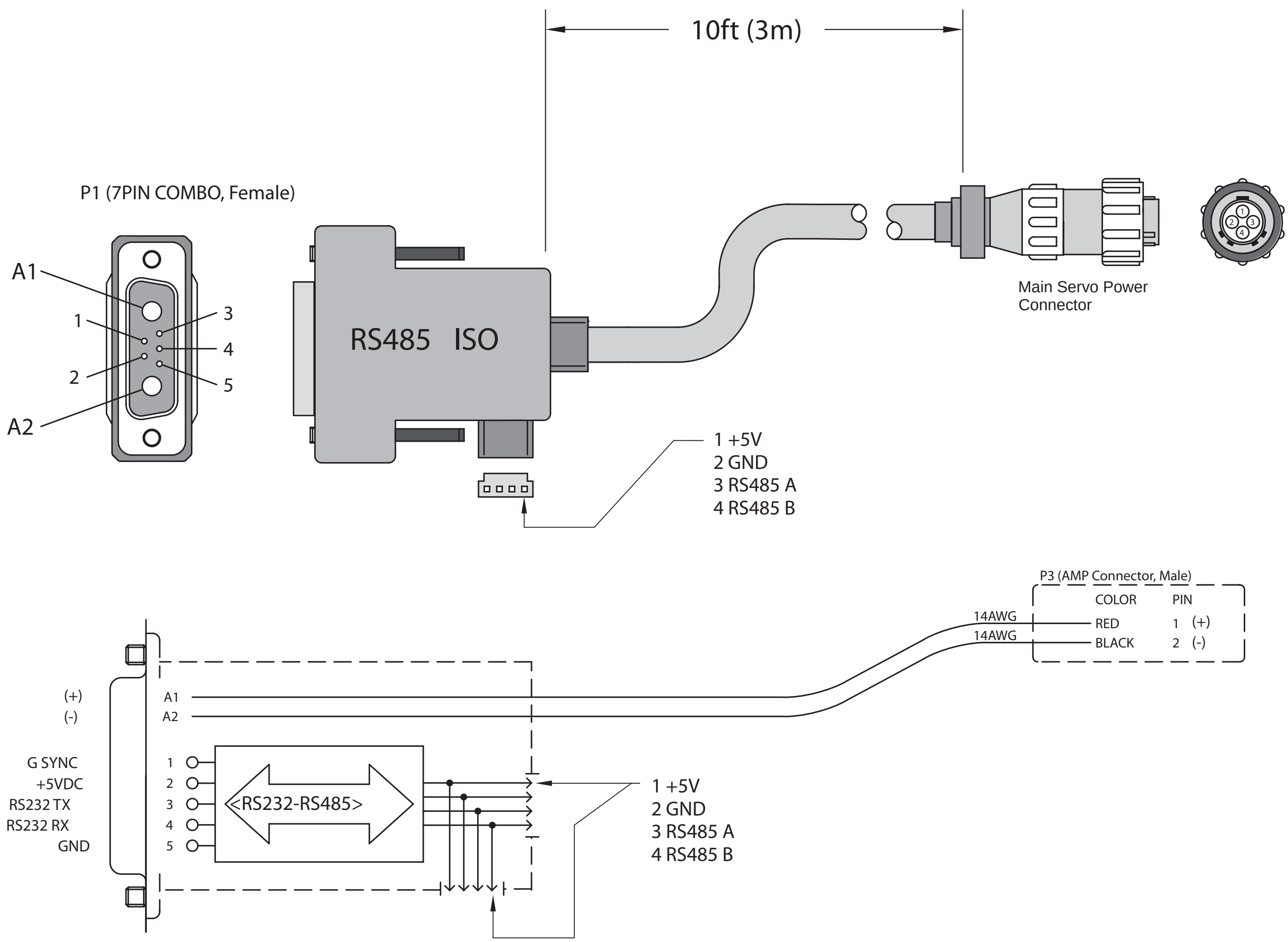
RS485-ISO

The RS485-ISO adapter provides electrically isolated conversion from RS-232 on the main 7W2 connector to RS-485. The adapter comes standard with a 10 Foot power cable and two parallel 4 pin Molex RS-485 connectors.

RS-485 provides improved noise immunity over cable lengths of up to 1,000 ft (305m). It also allows you to operate a network of up to 100 SmartMotors in parallel, rather than daisy-chaining the communications from one motor to the next.

The adapter draws power and ground from the SmartMotor main connector (pins 2 and 5). It does require the use of the main connector G-Synch line (pin 1) for Read-Write control of the 485 transceiver.

The RS485-ISO communications adapter can be ordered with or without the power cable (no cable P/N: RS485ISO-NOCBL).



RS232485T

RS232485T is a non-isolated RS-232 to RS-485 communications adapter. It requires no drivers because it is hardware based only. The DB-9 connector can be plugged directly into a standard PC Serial port allowing the user to easily connect to and communicate with RS-485 devices.

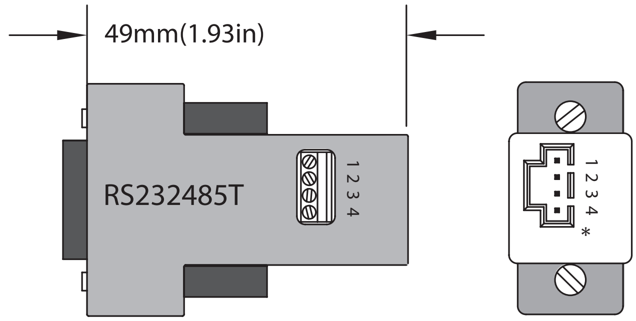
Note: The 4-pin molex connector is designed to match the RS485ISO adapters below.

The connector can be removed to allow direct screw terminal connection as well.

Includes two KITDC3

(Proper RS-485 biasing must be used)

PIN#	FUNCTION:
1	5VDC
2	GND
3	RS485 A
4	RS485 B



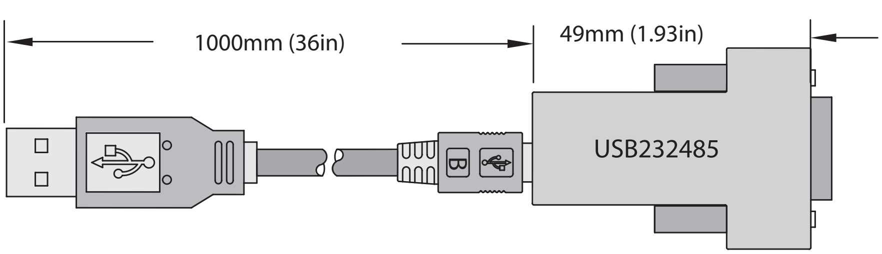
KITUSB232485

USB232485 is a non-isolated USB to RS-232 and RS-485 adapter. Available with Windows 98/2K/XP/Vista drivers.

This adapter plugs into standard USB port and provides either RS-232 or RS-485 communications. It is provided with ~1Meter standard USB cable.

Note: This is a single Port Device, it does not facilitate the use of both RS-232 AND RS-485 at the same time.

(Proper RS-485 biasing must be used in accordance with manual.)



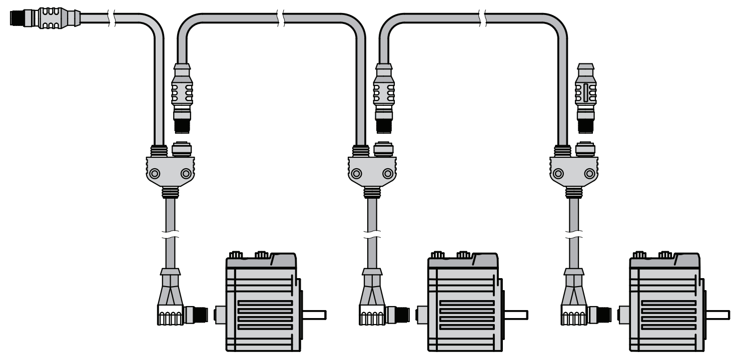
PIN#	FUNCTION:
1	RS485 A
2	RS232 Rx
3	RS232 Tx
4	+5Vdc
5	GND
9	RS485 B

CAN Bus Y Cables

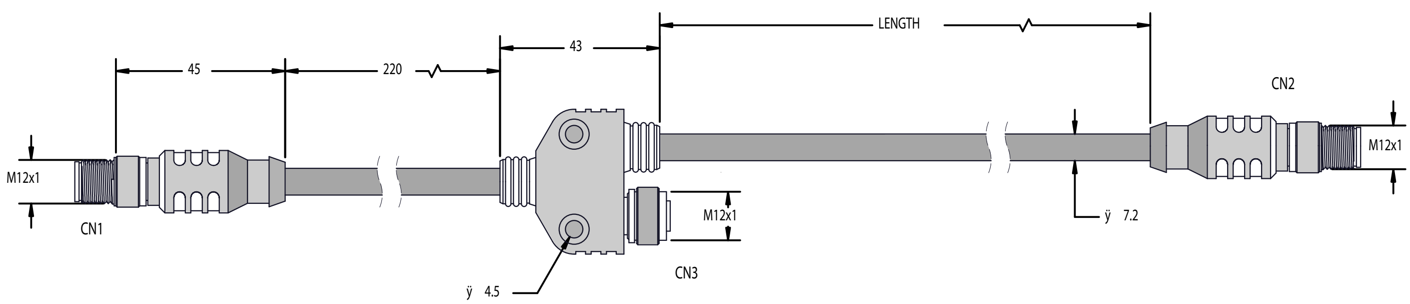
The CAN Y1 series is meant for CAN Bus communications connectivity between two motors with addition of a T-connector in line. Rated to >1MB data transmission rate

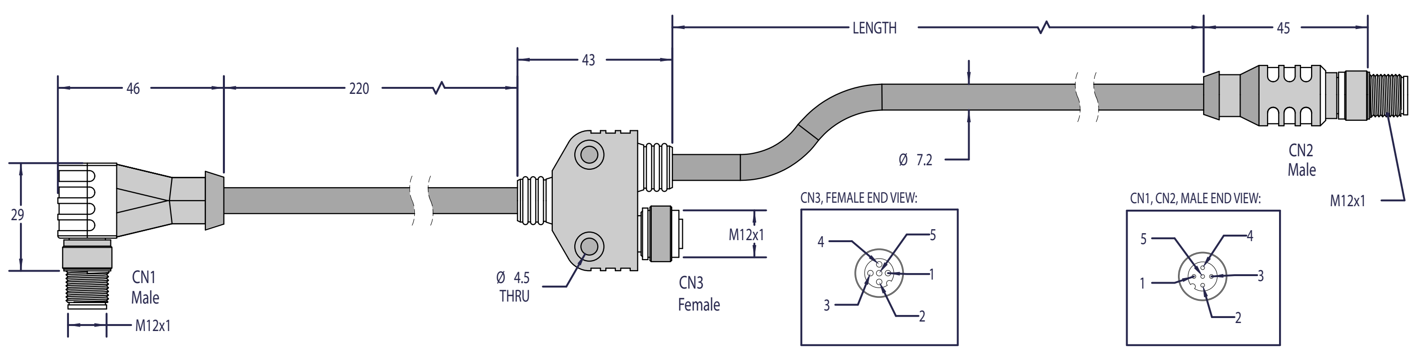
This allows for cascading of multiple SmartMotors™ and the addition of a shunt resistor when required.

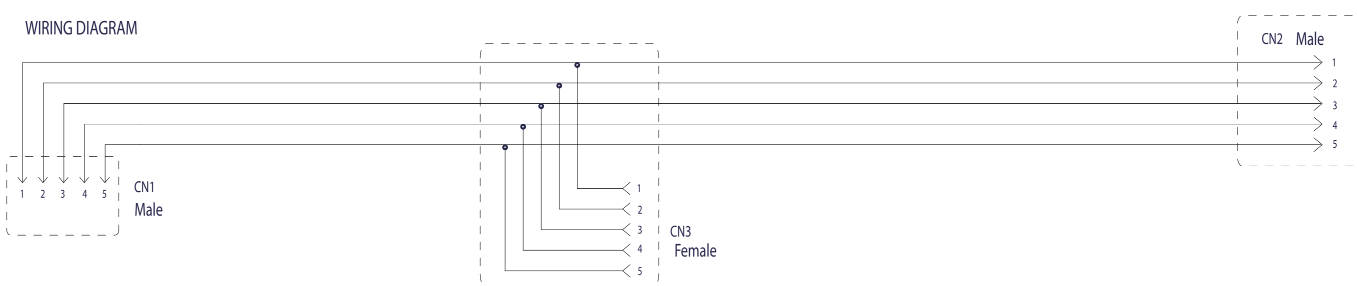
CBLIP series cables are sealed M12 threaded connector
Brass pins w/Gold plating, Maximum 4.0A 250V
Foil shield with Black PVC jacket ~7.4mm Dia.



Strait Connector Part Number	Length	Right Angle Connector Part Number	Length
CBLIP-CAN-Y1-0.5M	0.5 meters	CBLIP-CAN-Y1-0.5MRA	0.5 meters
CBLIP-CAN-Y1-1M	1 meters	CBLIP-CAN-Y1-1MRA	1 meters
CBLIP-CAN-Y1-3M	3 meters	CBLIP-CAN-Y1-3MRA	3 meters





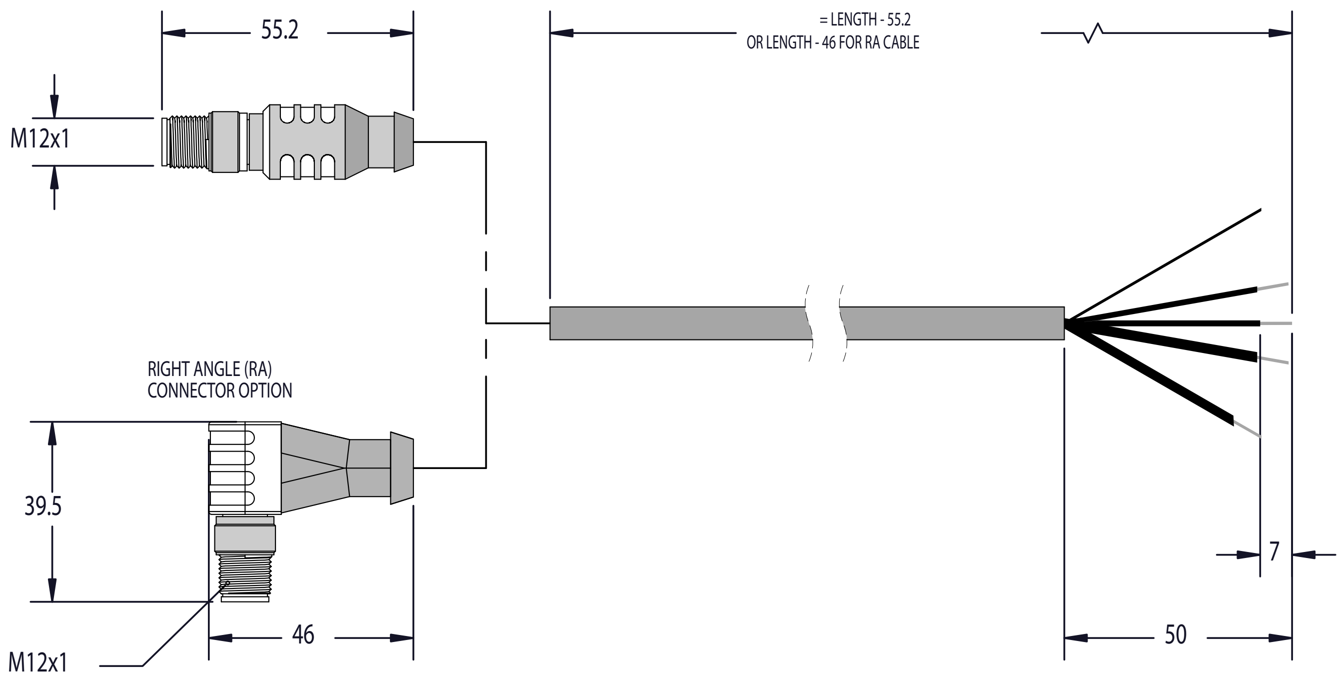


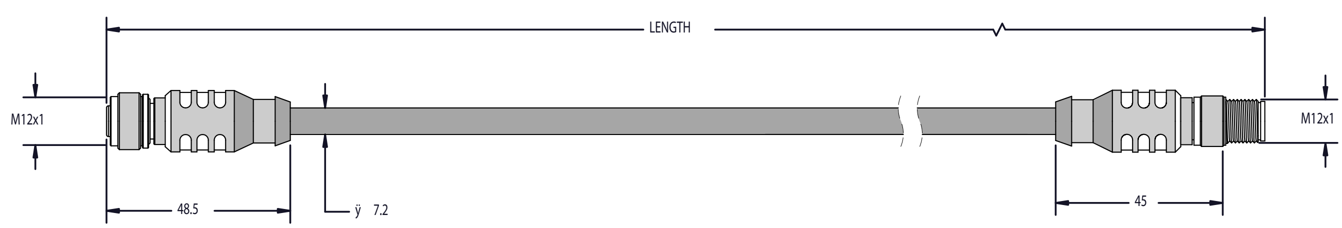
CAN Bus Flying Lead Cables

The CAN FL1 cables are standard CAN Bus communications cables rated to >1MB data transmission rate

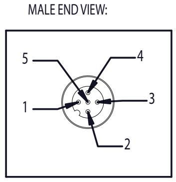
CBLIP series cables are sealed M12 threaded connector Brass pins w/Gold plating, Maximum 4.0A 250V
Foil shield with Black PVC jacket ~7.4mm Dia.

Part Number	Description	Length
CBLIP-CAN-FL-1M	Flying Lead Strait Connector	1 meter
CBLIP-CAN-FL-3M	Flying Lead Strait Connector	3 meters
CBLIP-CAN-Y1-3MRA	Flying Lead Right Angle Connector	3 meters
CBLIP-CAN-EXT-1M	CAN Bus Extension Cable	1 meter
CBLIP-CAN-EXT-2M	CAN Bus Extension Cable	2 meters
CBLIP-CAN-EXT-3M	CAN Bus Extension Cable	3 meters





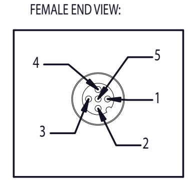
MALE END VIEW:



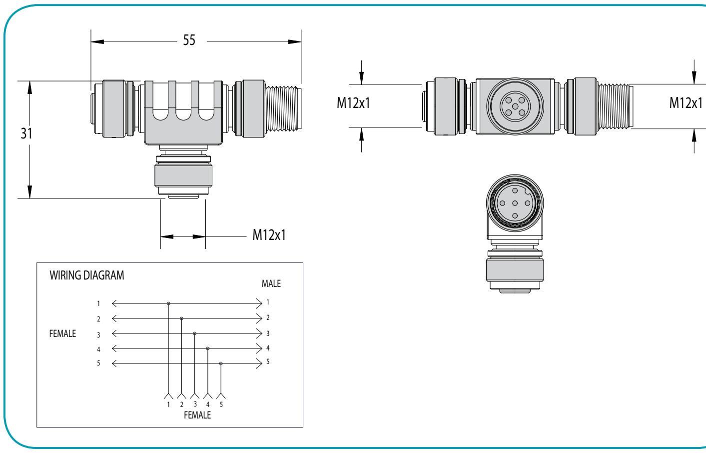
CONNECTIONS

PIN #	NAME	COLOR
1	SHIELD	BARE
2	+ VOLTAGE	RED
3	- VOLTAGE	BLACK
4	CAN_H	WHITE
5	CAN_L	BLUE

FEMALE END VIEW:



"T" CONNECTOR FEMALE-FEMALE-MALE CBLIP-T-FFM

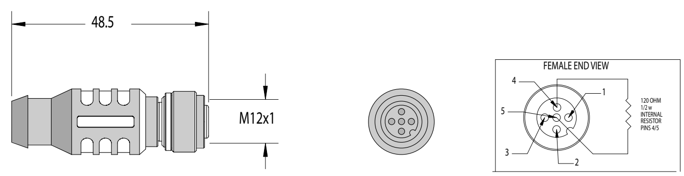


“T connectors may be used in place of Y cables an in conjunction with CAN Bus extension cables.

Shunt resistors MUST BE used to allow proper biasing of CAN Bus cables.

It is suggested to have at least one at furthest end of bus. Two may be required at opposing ends.”

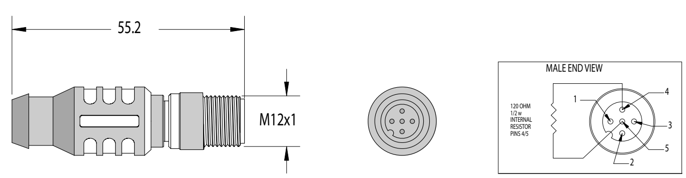
Female Terminating Resistor, 120 Ohms CBLIP-TRF120



“Shunt resistors MUST BE used to allow proper biasing of CAN Bus cables.

It is suggested to have at least one at furthest end of bus. Two may be required at opposing ends.”

Male Terminating Resistor, 120 Ohms CBLIP-TRM120



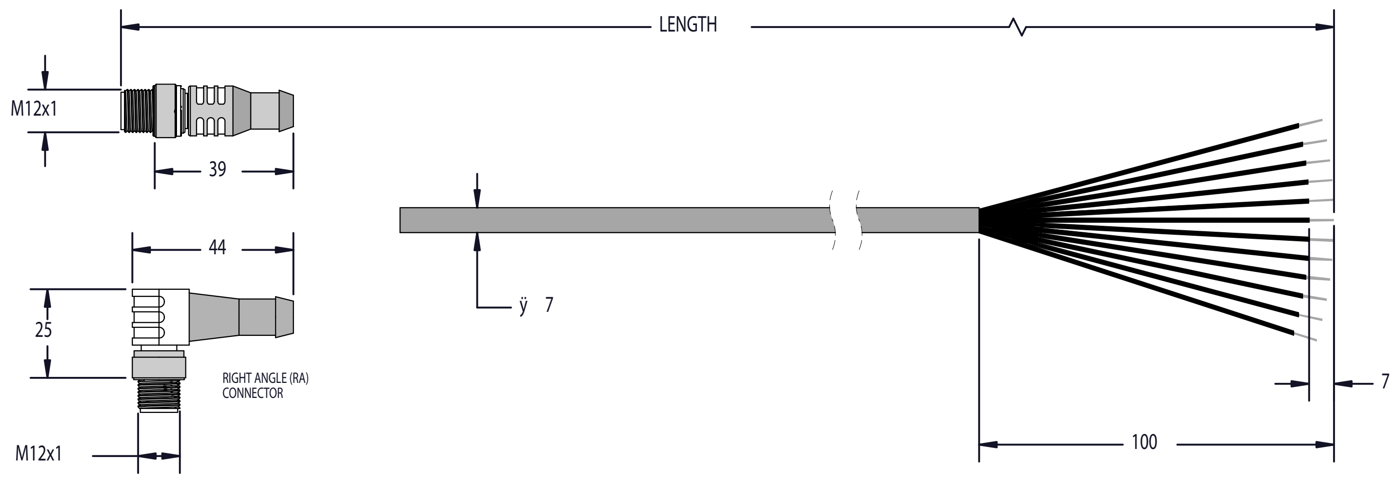
“Shunt resistors MUST BE used to allow proper biasing of CAN Bus cables.

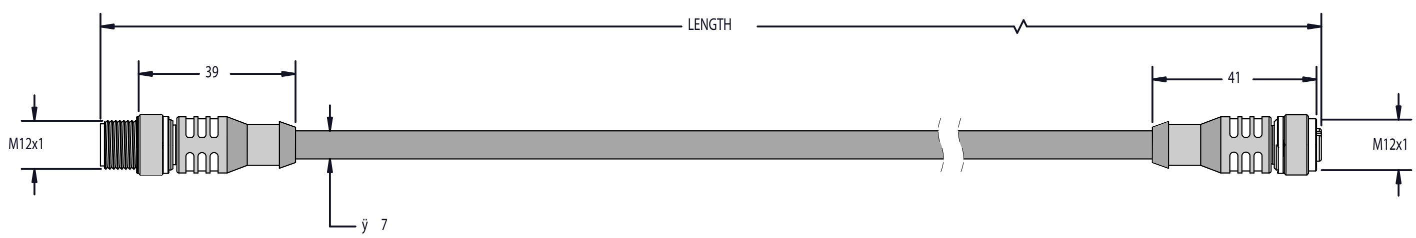
It is suggested to have at least one at furthest end of bus. Two may be required at opposing ends.”

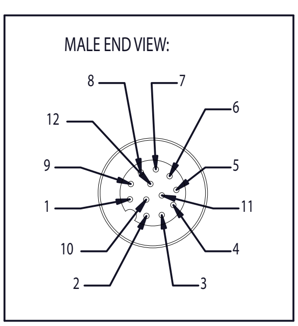
The Expanded I/O cables are for use with the -AD1 expanded I/O option on all Class 5 SmartMotors™. Each cable is a 12 conductor shielded cable. @ conductors are for +24VDC I/O power. The other 10 are for 10 channels of I/O.

CBLIP series cables are sealed M12 threaded connector Brass pins w/Gold plating, Maximum 4.0A 250V
Foil shield with Black PVC jacket ~7.4mm Dia.

Part Number	Description	Length
CBLIP-IO-FL-1M	Flying Lead Strait Connector	1 meter
CBLIP-IO-FL-3M	Flying Lead Strait Connector	3 meters
CBLIP-IO-FL-1MRA	Flying Lead Right Angle Connector	1 meter
CBLIP-IO-FL-3MRA	Flying Lead Right Angle Connector	3 meters
CBLIP-IO-EXT-1M	I/O Extension Cable	1 meters
CBLIP-IO-EXT-2M	I/O Extension Cable	2 meters
CBLIP-IO-EXT-3M	I/O Extension Cable	3 meters

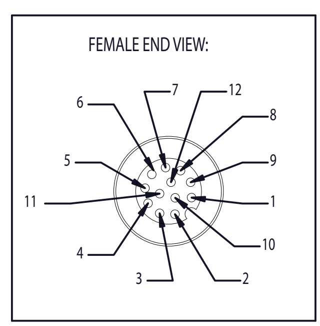






CONNECTIONS

PIN #	NAME
1	GREEN
2	VIOLET
3	PINK
4	WHITE
5	TAN
6	GRAY
7	BLACK
8	ORANGE
9	RED
10	YELLOW
11	BROWN
12	BLUE



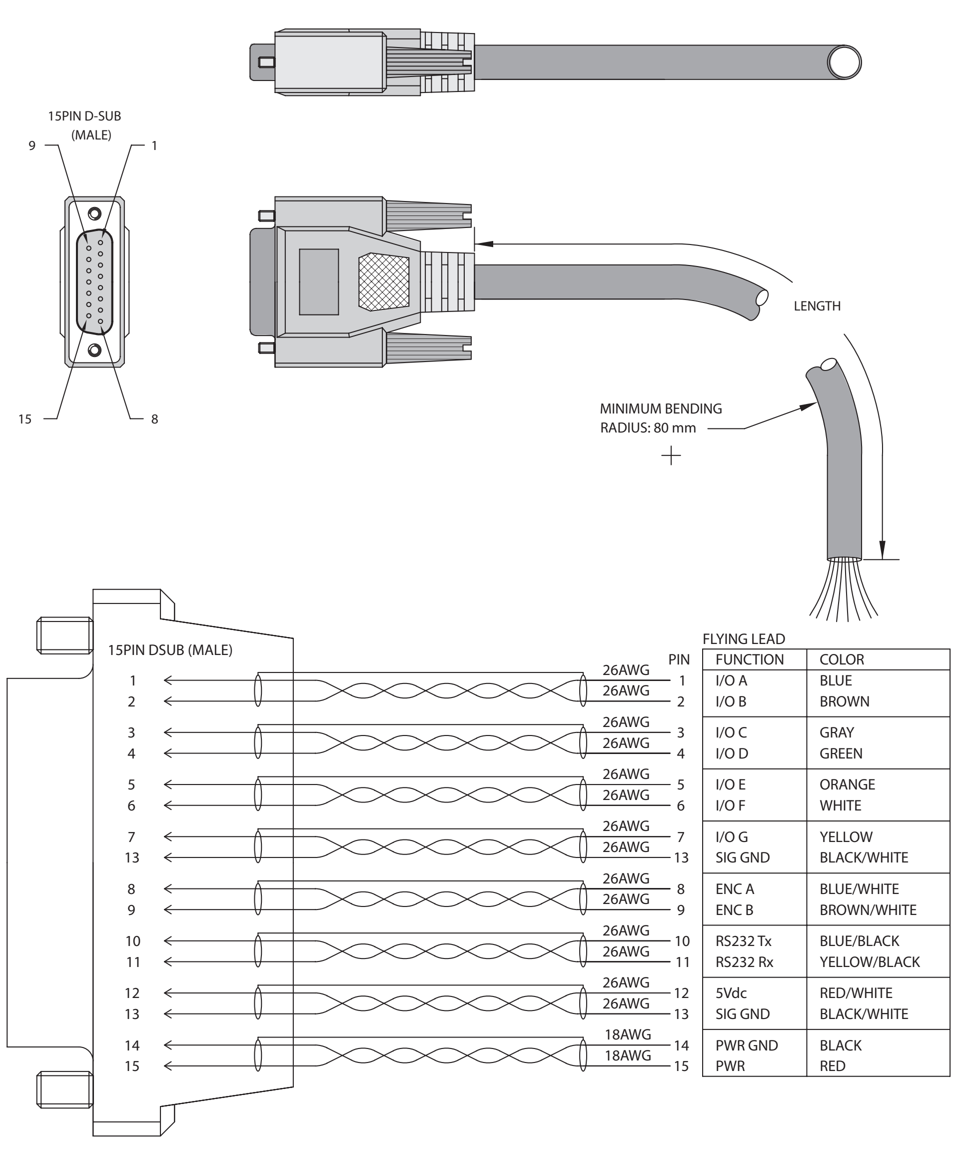
Main I/O connector Cable (Flying Leads) for DB-15 Connector on Animatics SmartMotor™.

CBLIO5V series is for all 5VTTL I/O, communications, Encoder output, and Control Power input (when needed for –DE option Motors).

The cable is organized in separately shielded twisted pairs to provide better noise immunity and lower emissions.

Part Number	Length
CBLIO5V-3M	3 meters
CBLIO5V-5M	5 meters
CBLIO5V-10M	10 meters

Note: The shields DO NOT have electrical contact with each other or the connector shell. This allows for proper grounding in the control cabinet or at termination point determined by user thereby eliminating ground loops.



PIN	FUNCTION	COLOR
1	I/O A	BLUE
2	I/O B	BROWN
3	I/O C	GRAY
4	I/O D	GREEN
5	I/O E	ORANGE
6	I/O F	WHITE
7	I/O G	YELLOW
13	SIG GND	BLACK/WHITE
8	ENC A	BLUE/WHITE
9	ENC B	BROWN/WHITE
10	RS232 Tx	BLUE/BLACK
11	RS232 Rx	YELLOW/BLACK
12	5Vdc	RED/WHITE
13	SIG GND	BLACK/WHITE
14	PWR GND	BLACK
15	PWR	RED

The CBLIO-ISO1 cable provide optically isolated 24VDC I/O Interface to the controller.

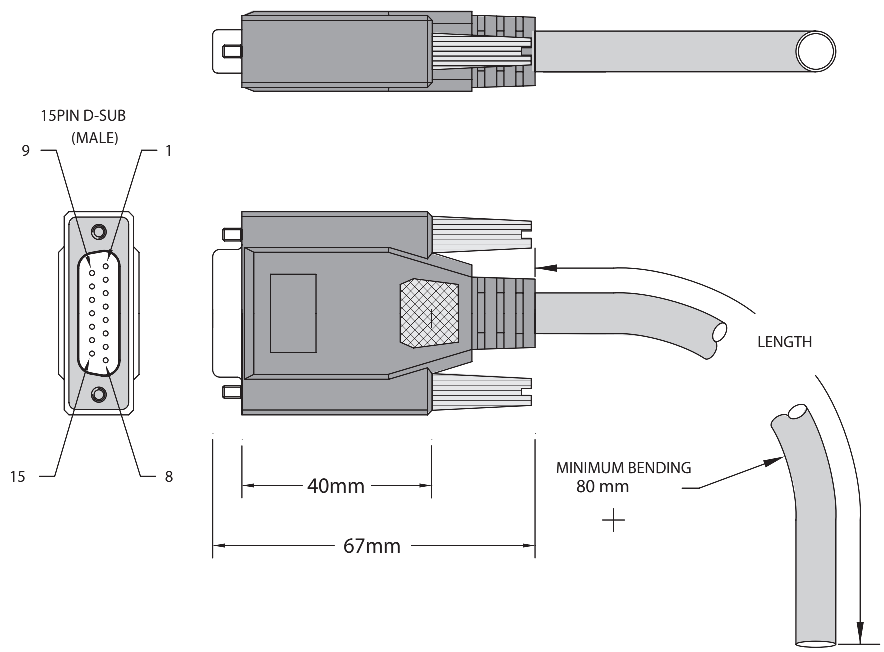
The cable is user configurable as 4 inputs and 3 outputs OR 5 inputs and 2 outputs.

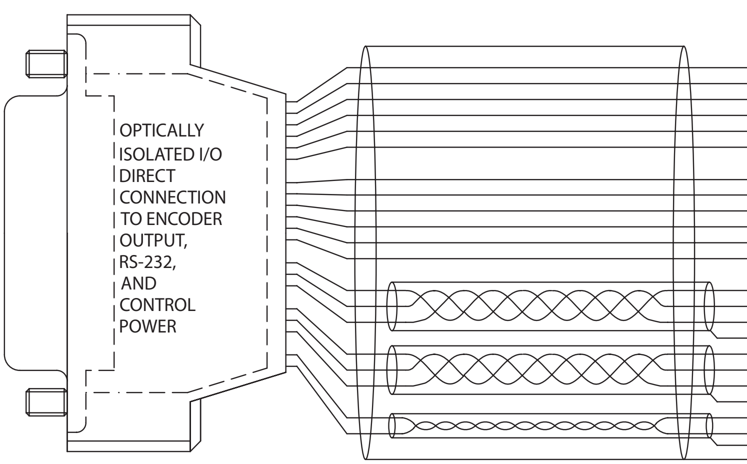
Additionally, this cable provides direct connection to:

RS-232 Primary Communications Port (Ch. 0)
Encoder Output
Control Power Input

Part Number	Length
CBLIO-ISO1-3M	3 meter
CBLIO-ISO1-5M	5 meter
CBLIO-ISO1-10M	10 meter

It can be used with standard or –DE option Animatics SmartMotors™





FUNCTION:	COLOR:	AWG
INPUT COMMON A-G	WHITE	26
INPUT A	BLUE	26
INPUT B	YELLOW	26
INPUT C	GREEN	26
INPUT D	VIOLET	26
INPUT G	ORANGE	26
OUTPUT E (+)	ORANGE/WHITE	26
OUTPUT E (-)	ORANGE/BLACK	26
OUTPUT F (+)	YELLOW/WHITE	26
OUTPUT F (-)	YELLOW/BLACK	26
OUTPUT G (+)	VIOLET/WHITE	26
OUTPUT G (-)	VIOLET/BLACK	26
RS232 Rx	GREEN/BLACK	26
RS232 Tx	GREEN/WHITE	26
RS232 Common	BROWN	26
Enc. A Out	BLUE/WHITE	26
Enc. B Out	BLUE/BLACK	26
Enc. Common	BROWN	26
CONTROL PWR IN (-)	BLACK	26
CONTROL PWR IN (+)	RED	26

SHIELD AND DRAIN WIRES DO NOT CONTACT CIRCUIT BOARD OR CONNECTOR SHELL

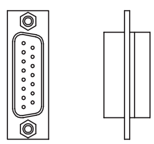
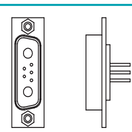
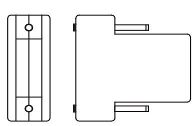
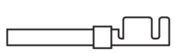
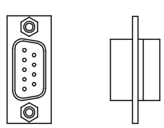
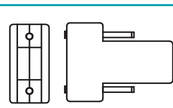
SMDEVPACK-D

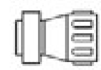
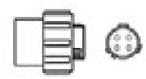
SMDEVPACK-D is the introductory development package for the Animatics SmartMotor™. It is highly recommended for first time users and developers alike.

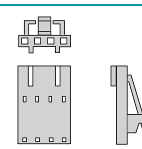
SMDEVPACK-D includes:

- UG-SM: Animatics SmartMotor User's Guide
- CD-SMI: SMI (Smart Motor Interface) software CD
- CBLSM1-3M: Power-Communications Cable
- KITSMDC3: D-Sub Connector kit
- KITDC1: DC power supply connector kit

NOTE: Connector kits above include all parts below on this page with exception of KITDC3

KITSMDC3		
Part	**Description**	**Qty.**
CN132	Connector, Male, 15 Pin D-sub NOTE: for I/O Connector 	1
P104	Pin Contact Contact Size 20 Crimp Type, 24-20 AWG	20
CN141	Connector, 7W2 Combination 7 Pin DB-15 Shell Size 2 	1
CN142	Socket Contact Solder Cup 14 AWG for 7W2 Combo Connector 	2
CN149	Connector Hood For DB-15 Shell Size, EMI/Magnetic Shielded 	1
P121	Pin Contact, Female Contact Size 20 Crimp Type, 24-20 AWG 	20
CN121V	Connector, Female 9 Pin D-Sub 	1
CN161	Connector Hood For DB-9 Shell Size EMI/Magnetic Shielded 	2

KITDC1		
Part	**Description**	**Qty.**
CN114	Connector Shroud/Hood 	1
CN119	4-Pin Connector NOTE: For enclosed Power Supplies & Shunts 	1
P102	Pin, Crimp Type, Male 16-18AWG 	4

KITDC3		
Part	**Description**	**Qty.**
KITDC3	4-Pin Molex Communication Connector Kit NOTE: Includes 4 female crimp pins 	1

Follow these easy rules to assure success in using the Animatics SmartMotor™ to maximize your system's reliability. Please take a minute to see that your system design and implementation passes the test. Keep this page handy to document your settings and send it along with any motors returned for inspection or repair.

1. Power supply selection is very important.

☐ Provide for a means to keep the SmartMotor's voltage below 48 VDC by…

- ◯ operating at 48 VDC or less as nominal or,
- ◯ using a shunt near the motor or,
- ◯ adding a shunt to a switching power supply or,
- ◯ operating at 48 VDC or less and adding a shunt for a vertical application.

2. Proper electrical interfacing is essential.

☐ Refrain from creating any ground loops with the communications by…

- ◯ isolating the ground prong of the host PC for a single motor application or,
- ◯ isolating the motor's power supply for a single motor application or,
- ◯ using a communication isolation product to protect each axis or,
- ◯ operating only smaller motors at low power, like short SM23s or smaller.
- ◯ this does not apply when employing no serial communications at all.

☐ Refrain from creating any ground loops with the SmartMotor's I/O by…

- ◯ using the main or 5V power at the motor to operate any sensors or,
- ◯ using an opto-coupler to interface to the inputs or outputs or,
- ◯ using an I/O isolation product or,
- ◯ operating only smaller motors at low power, like short SM23s or smaller.
- ◯ this does not apply when employing no I/O connections at all.

3. Properly sizing an Animatics SmartMotor™ for the application is critical.

☐ Determine that the motor selected has the torque to handle the friction.

☐ Determine that the motor selected has the torque to support any vertical component of the load.

☐ Determine that the motor selected has the torque to accelerate the load.

☐ Determine that the motor's rotor inertia is properly matched to the load.

4. Considering the thermal environment for the SmartMotor is important.

☐ Consider the ambient temperature and avoid applications above 70°C ambient.

☐ Maximize the heat sinking capability of the motor's mount to any extent possible.

5. Proper mechanical and environmental implementation is needed.

☐ Assure motor shaft loading is within axial and radial limits.

☐ Be certain that the motor does not get exposed to fluids or excessive moisture.

☐ Insure relative humidity is <30% and non condensing.

⚠ Noise Filtering may be necessary

Ports A, B, C, D, and G are all classified as high speed input.

1. **Ports A and B:** Ports A and B may be configured as Phase A and B encoder input or Step and Direction input. To ensure proper operation when following external encoders, it is strongly advised to use Line Driver encoders or encoders with true push-pull drive capability. This allows up to 1.5MHz input frequency. Open Collector output encoders will not work above 20KHz or so typically due the inability to drive input capacitance well enough.
2. **Ports C and D:** Both Ports C and D default as Over Travel Limit inputs. They can be triggered by a negative edge transition as fast as 5 microseconds allowing optimum fail safe detection of over-travel. However, as a result, in noisy systems, it may require capacitive filtering to prevent false triggers from static or other induced noise.
3. **Port G:** Port G defaults as the "sync" or synchronous trigger input. This means any time it goes low, the processor issues a "G" command. The purpose of this is to allow pre-condition setup of motion profiles and then via hardware input the user may synchronize multiple motor/axis starts at the exact same time. However, similar to the Over Travel limit inputs, Port G can detect input pulses down to 5 microseconds. To ensure no false triggers, some system installs may require capacitive filtering to avoid false triggers.
   In all cases above, typical filtering is via 10 to 100 picofarad ceramic capacitors tied from the input pin to ground. This is only a recommendation; system characteristic may vary depending on surrounding equipment.

Which is better, Linear or Switcher Supplies?

Since Servo Motors are inductive they may run highly dynamic motion profiles. As a result, their current demand can vary widely. Surge currents from stand-still to maximum load may be extremely high, yet steady state current demand over time may be relatively mild. As a result, proper care should be taken when selecting power supplies.

Animatics offers two basic types of power supplies.

The chart to the right gives a brief comparison of the two types of supplies.

	Linear	Switcher
AC Input	Field selectable (120/240VAC)	Universal 90-240VAC
Power Factor Corrected	No	Yes
Relative Size	Big and bulky	Lightweight
Cooling	Ambient convection	Fan cooled
Surge Capacity	400%	5%
Voltage Regulation	15% Drop over range	0%, Fixed
Shunt Required ? [1]	Occasionally, but not typically	In most cases, highly recommended!

[1] **See Shunt Section for more information!**

As seen in the graph to the right, Linear (Unregulated) supplies can handle large surge current loads. This is because Linear Supplies typically contain large output capacitors to handle those surges well.

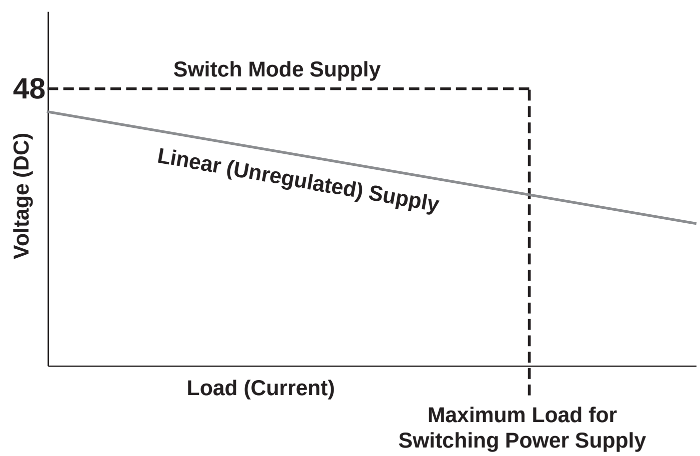
Voltage regulation: Switchers are highly regulated supplies. They will maintain fixed voltage until they reach maximum load and then will "crowbar" to zero volts to protect the output stages. Linear supplies will slowly drop in output voltage while supplying more and more current.

This is the most fundamental difference between switchers and unregulated supplies.

Even though a switcher cannot handle the higher current surges, if it can output as much current as you would expect for a given servo application, then they will actually help the servo accelerate much faster because system voltage will be maintained at maximum level.

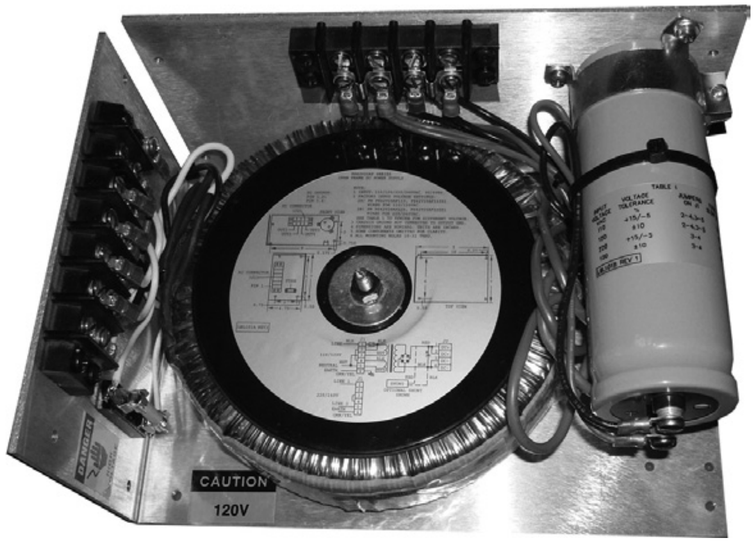
However, if your servo application requires surge currents in excess of 50 Amps or more, the switchers may not be cost effective. Getting 50 amps from an Animatics 20 Amp supply is easy. Getting 50 Amps from Animatics switchers would require placing multiple units in parallel, so it may not be cost effective to do so.

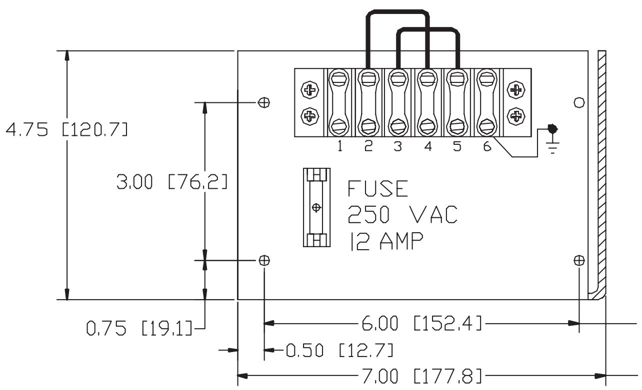
Voltage Drop Comparison

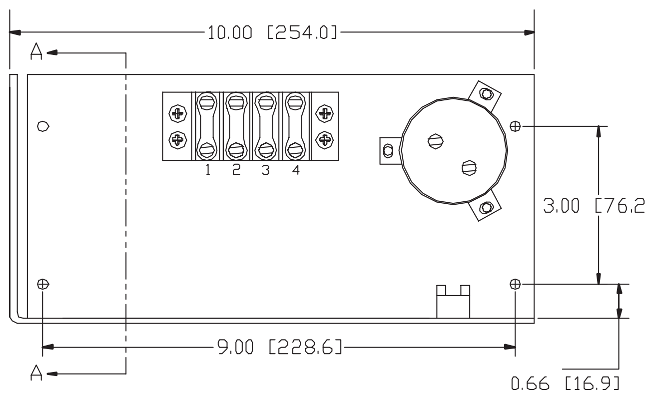


Power Supplies:

- Linear Unregulated
- AC Input, DC output
- Screw Terminal Access
- Toroid Transformer for lower EMI







All sizes are given in inches, sizes in brackets are in mm

Part Number	Input Voltage and Frequency	No Load Output Voltage	Full Load Output		Nominal Wattage	Shunt	Weight(Nom.)
			Voltage	Current			
PS42V20AF110	120VAC 50/60Hz	44VDC	35VDC	20 Amps	680 W		16.5 lbs (7.5kg)
PS42V20AF220	240VAC 50/60Hz	44VDC	35VDC	20 Amps	680 W		16.5 lbs (7.5kg)
PS42V20AF110-S1	120VAC 50/60Hz	44VDC	35VDC	20 Amps	680 W	100 W	17 lbs (7.7kg)
PS42V20AF220-S1	240VAC 50/60Hz	44VDC	35VDC	20 Amps	680 W	100 W	17 lbs (7.7kg)
PS42V20AF110-S2	120VAC 50/60Hz	44VDC	35VDC	20 Amps	680 W	200 W	17 lbs (7.7kg)
PS42V20AF220-S2	240VAC 50/60Hz	44VDC	35VDC	20 Amps	680 W	200 W	17 lbs (7.7kg)

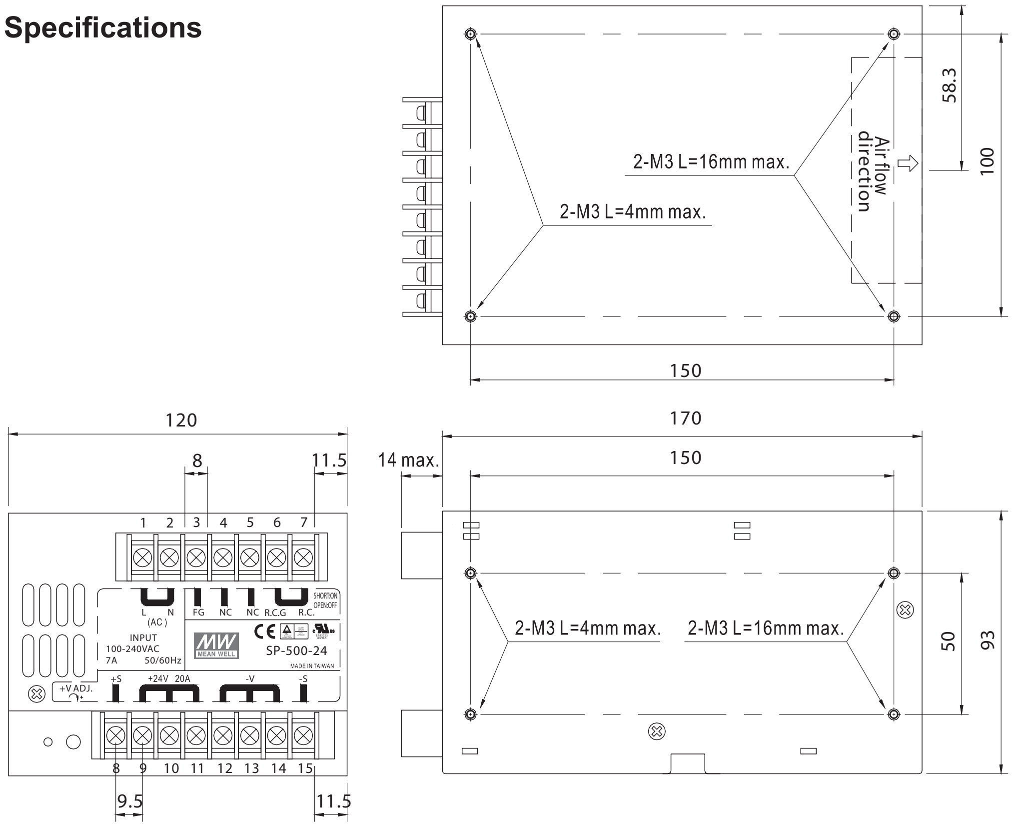
PFC500W-48 Features

- Universal AC input / Full range
- Built-in active PFC function, PF>0.95
- Protections: Short circuit / Overload/ Over voltage / Over temperature
- Forced air cooling by built-in DC fan
- Built-in cooling Fan ON-OFF control
- Built-in remote ON-OFF control
- Built-in remote sense function
- Fixed switching frequency at 110KHz
- 2 years warranty

OUTPUT	DC VOLTAGE	48V
	RATED CURRENT	10A
	CURRENT RANGE	0 ~ 10A
	RATED POWER	480W
	RIPPLE & NOISE (max) Note.2	3--mVp-p
	VOLTAGE ADJ. RANGE	41 ~ 56V
	VOLTAGE TOLERANCE Note.3	± 1.0%
	LINE REGULATION	± 0.5%
	LOAD REGULATION	± 0.5%
	SETUP, RISE TIME	1500ms, 50ms at full load
	HOLD UP TIME (Typ.)	24 ms at full load
INPUT	VOLTAGE RANGE Note.5	88 ~ 264VAC 124 ~ 370VDC
	FREQUENCY RANGE	47 ~ 63Hz
	POWER FACTOR (Typ.)	PF>0.95/230VAC PF>0.95/115VAC at full load
	EFFICIENCY(Typ.)	87%
	AC CURRENT (Typ.)	7A/115VAC 3.5A/230VAC
	INRUSH CURRENT (Typ.)	18A/115VAC 36A/230VAC
PROTECTION	OVER VOLTAGE	57.6 ~ 67.2V
FUNCTION	REMOTE CONTROL	RC+/RC-: Short = power on ; Open = power off
	WORKING TEMP.	-10 ~ +50°C (Refer to output load derating curve)
	WORKING HUMIDITY	20 ~ 90% RH non-condensing
ENVIRON-MENT	STORAGE TEMP., HUMIDITY	-20 ~ +85°C, 10 ~ 95% RH
SAFETY & EMC (Note 4)	SAFETY STANDARDS	UL60950-1, TUV EN60950-1 approved
	WITHSTAND VOLTAGE	I/P-O/P:3KVAC I/P-FG:1.5KVAC O/P-FG:0.5KVAC
	ISOLATION RESISTANCE	I/P-O/P, I/P-FG, O/P-FG:100M Ohms/500VDC
	EMI CONDUCTION & RADIATION	Compliance to EN55022 (CISPR22) Class B
	HARMONIC CURRENT	Compliance to EN61000-3-2,-3
	EMS IMMUNITY	Compliance to EN61000-4-2,3,4,5,6,8,11; ENV50204, light industry level, criteria A
OTHERS	MTBF	133.4K hrs min. MIL-HDBK-217F (25°C)
	DIMENSION	170*120*93mm (L*W*H)
NOTE	1. All parameters NOT specially mentioned are measured at 230VAC input, rated load and 25°C of ambient temperature. 2. Ripple & noise are measured at 20MHz of bandwidth by using a 12" twisted pair-wire terminated with a 0.1uf & 47uf parallel capacitor. 3. Tolerance: includes set up tolerance, line regulation and load regulation. 4. The power supply is considered a component which will be installed into a final equipment. The final equipment must be re-confirmed that it still meets EMC directives. 5. Derating may be needed under low input voltages. Please check the derating curve for more details.	

Animatics Corporation
tel: 408.748.8721 • fax: 408.748.8725 • www.animatics.com

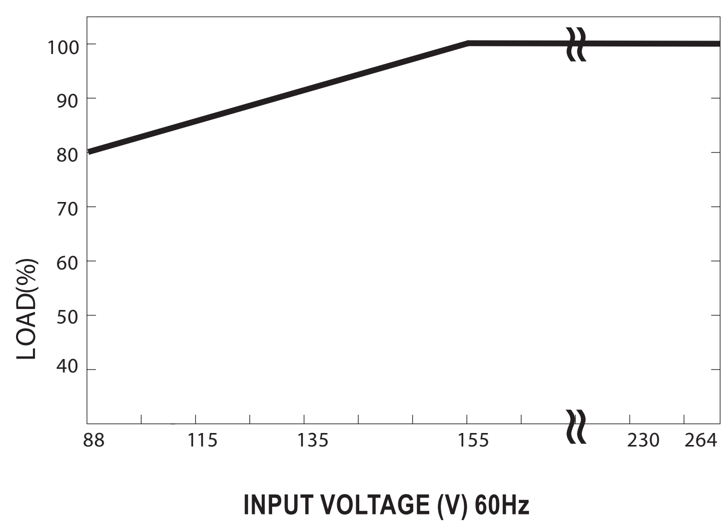
Mechanical Specifications



Terminal Pin No. Assignment

Pin No.	Assignment	Pin No.	Assignment
1	AC/L	7	R.C.
2	AC/N	8	+S
3	FG ⏚	9~11	DC OUTPUT +V
4,5	NC	12~14	DC OUTPUT -V
6	R.C.G	15	-S

Output Derating vs. Input Voltage



⚠ WARNING

Enclosed Switch Mode DC Power Supply

PFC1500W-48 ANIMATICS®

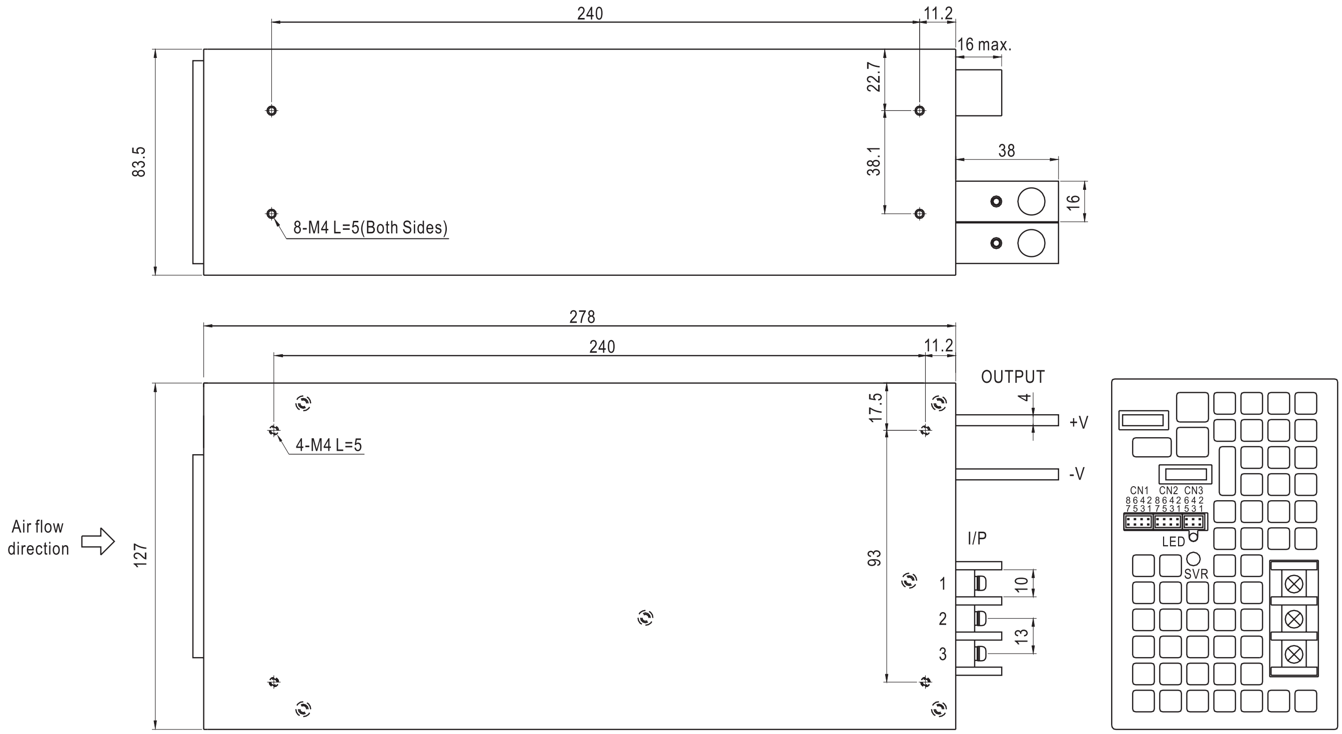
PFC1500W-48 Features:

- Universal AC input / Full range
- AC input active surge current limiting
- Built-in active PFC function, PF>0.95
- Protections: Short circuit / Overload/ Over voltage / Over temperature
- Forced air cooling by built-in DC fan
- Built-in cooling fan ON-OFF control
- Built-in remote ON-OFF control
- Built-in remote sense function
- 2 year warranty

NOTE: Multiple units may be paralleled for additional power

OUTPUT	DC VOLTAGE	48V
	RATED CURRENT	32A
	CURRENT RANGE	0 ~ 32A
	RATED POWER	1536W
	RIPPLE & NOISE (max) Note.2	200mVp-p
	VOLTAGE ADJ. RANGE	43 ~ 56V
	VOLTAGE TOLERANCE Note.3	± 1.0%
	LINE REGULATION	± 0.5%
	LOAD REGULATION	± 0.5%
	SETUP, RISE TIME	1500ms, 100ms at full load
	HOLD UP TIME (Typ.)	16 ms at full load
INPUT	VOLTAGE RANGE Note.5	88 ~ 264VAC 124 ~ 370VDC
	FREQUENCY RANGE	47 ~ 63Hz
	POWER FACTOR (Typ.)	0.95/230VAC 0.98/115VAC at full load
	EFFICIENCY(Typ.)	91%
	AC CURRENT (Typ.)	17A/115VAC 8A/230VAC
	INRUSH CURRENT (Typ.)	30A/115VAC 60A/230VAC
	LEAKAGE CURRENT	<2.0mA/240VAC
PROTECTION	OVERLOAD Note.5	105 ~ 135% rated output power
		Protection type: Constant current limiting unit will shut down o/p voltage after 5 sec. Re-power to recover
	OVER VOLTAGE	57.6 ~ 67.2V
		Protection type: Shut down o/p voltage, recover automatically after temperature goes down
	OVER TEMPERATURE	95°C ± °5C detect on heatsink of power transistor
		Protection type: Shut down o/p voltage, recovers automatically after temperature goes down
FUNCTION	AUXILIARY POWER (AUX)	12V@0.1A (Only for Remote ON/OFF control)
	REMOTE CONTROL	RC+/RC-: Short = power on ; Open = power off
	WORKING TEMP.	-10 ~ +50°C (Refer to output load derating curve)
	WORKING HUMIDITY	20 ~ 90% RH non-condensing
ENVIRON-MENT	WORKING TEMP.	-20 ~ +70°C (Refer to output load derating curve)
	WORKING HUMIDITY	20 ~ 90% RH non-condensing
	STORAGE TEMP., HUMIDITY	-40 ~ +85°C, 10 ~ 95% RH
	TEMP. COEFFICIENT	± 0.5%/°C (0 ~ 50 °C)
	VIBRATION	10 ~ 500Hz, 2G 10min./1cycle, 60min. each along X,Y,Z axis
SAFETY & EMC (Note 4)	SAFETY STANDARDS	UL60950-1, TUV EN60950-1 approved
	WITHSTAND VOLTAGE	I/P-O/P:3KVAC I/P-FG:1.5KVAC O/P-FG:0.5KVAC
	ISOLATION RESISTANCE	I/P-O/P, I/P-FG, O/P-FG:100M Ohms/500VDC
	EMI CONDUCTION & RADIATION	Compliance to EN55022 (CISPR22)
	HARMONIC CURRENT	Compliance to EN61000-3-2,-3
	EMS IMMUNITY	Compliance to EN61000-4-2,3,4,5,6,8,11; ENV50204, light industry level, criteria A
OTHERS	MTBF	62.6K hrs min. MIL-HDBK-217F (25°C)
	DIMENSION	278*127*83.5mm (L*W*H)
	PACKING	2.6Kg: 6PCS/16.6Kg/1.54CUFT
NOTE	1. All parameters NOT specially mentioned are measured at 230VAC input, rated load and 25°C of ambient temperature. 2. Ripple & noise are measured at 20MHz of bandwidth by using a 12” twisted pair-wire terminated with a 0.1uf & 47uf parallel capacitor. 3. Tolerance: includes set up tolerance, line regulation and load regulation. 4. The power supply is considered a component which will be installed into a final equipment. The final equipment must be re-confirmed that it still meets EMC directives. 5. Derating may be needed under low input voltages. Please check the derating curve for more details.	

Mechanical Specifications



AC Input Terminal Pin No. Assignment

Pin No.	Assignment
1	FG ⏚
2	AC/N
3	AC/L

Control Pin No. Assignment(CN1,CN2) : HRS DF11-8DP-2DS or equivalent

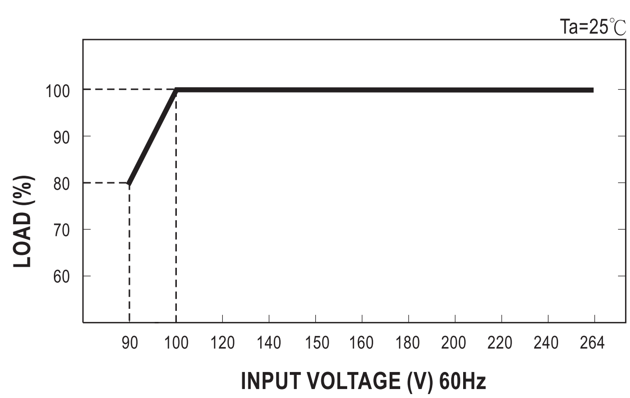
Pin No.	Assignment	Pin No.	Assignment	Mating Housing	Terminal
1	RCG	4	TRIM	HRS DF11-8DS or equivalent	HRS DF11-**SC or equivalent
2	RC2	6	LS(Current Share)		
3,5,7	-S	8	+S		

RCG: Remote ON/OFF Ground
RC2: Remote ON/OFF
-S : -Remote Sensing
TRIM: Adjustment of Output Voltage
LS: Load Share
+S: +Remote Sensing

Control Pin No. Assignment(CN3) : HRS DF11-6DP-2DS or equivalent

Pin No.	Assignment	Pin No.	Assignment	Mating Housing	Terminal
1	P OK GND	4	AUXG	HRS DF11-6DS or equivalent	HRS DF11-**SC or equivalent
2	P OK	5	RC1		
3	RCG	6	AUX		

P OK GND: Power OK Ground
P OK: Power OK Signal
RCG: Remote ON/OFF Ground
AUXG: Auxiliary Ground
RC1: Remote ON/OFF
AUX: Auxiliary Output



⚠ WARNING

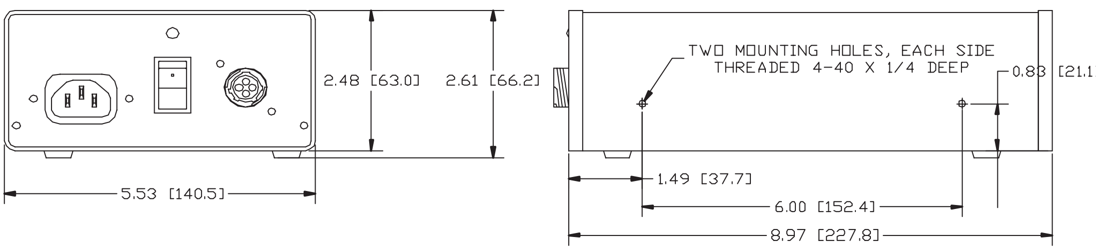
The switcher supplies have an adjustable output trim pot. The output voltage MUST BE adjusted to <=48VDC.

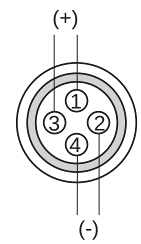
Enclosed DC Power Supplies

- Enclosed linear unregulated power supplies
- PC-type AC power cord
- 4 pin AMP connector on output
- Internally fused on both primary and secondary side
- Toroid transformer for minimal voltage drop and minimal EMI

Includes AC Power Cord and KITDC1 connector kit (see page 80)

Part Number	Input Voltage	Hz AC	No Load Voltage	Full Load		Nominal Wattage	Weight (Nom.)
				Voltage	Current		
PS24V8AG-110	110VAC	60	25 VDC	19 VDC	8 Amps	152 W	6.5 lbs (3 kg)
PS42V6AG-110	110VAC	60	46 VDC	38.7 VDC	6.5 Amps	251 W	7 lbs (3.2 kg)
PS42V6A-220CE	220 VAC	50-60	46 VDC	38.7 VDC	6.5 Amps	251 W	7 lbs (3.2 kg)





* All sizes are given in inches, sizes in brackets are in mm

NOTE: Either pair of Power Pins can handle full load rating

PWR116V

Enclosed Laptop Type Power Supply

This Power Supply connects directly to:

- CBLSM1-DEMO
- SmartBox™
- SmartBoxBCD™

It is ideal for desktop testing of SmartMotors™ and will easily run an unloaded SmartMotor for programming and evaluation testing.

Input:	120-240VAC 50/60Hz
Output:	24VDC, 2.1Amps, 50 Watts
Connector Type:	2.1x5mm coax DC Power Connector
Cable Length	~1meter

RoHS/CE Certified

CBLAC4

AC Power Cable for PWR116V Power Supply above.

Standard 2 prong US AC plug

~1 Meter length

Animatics Corporation offers several shunt options for use with DC input servo motors.

Shunts are needed to protect the servo controller and drive stages from over voltage.

Over voltage sources originate from the following:

- Back EMF due to back driving the motors
- Sudden or hard decelerations
- Hard stop crashes (immediate deceleration to zero speed)
- Vertical load drops

The shunts actually add an additional load to the DC bus automatically when voltage exceeds the trigger level by connecting large load resistors across the bus. Trigger voltage is typically 49.5VDC. As a result, the shunts will work with any of the supplies we offer.

WARNING

The switcher supplies have an adjustable output trim pot. If used with our shunts, the output voltage MUST BE adjusted to <=48VDC to insure the shunts do not stay gated on.

The Real story about Back EMF:

Generally speaking, back EMF is the voltage generated in a motor when it spins. This voltage is typically proportional to speed. However, this is a general rule. The truth is that the back EMF voltage is proportional to the rate of change of magnetic flux in the windings of the stator. As a result, constant speeds produce constant and predictable voltages. However, sudden changes due to decelerations or hard stop crashes cause an immediate change in magnetic flux or even a total instantaneous collapse. As a result, voltages can go 5 to 10 times higher than spinning the motor at its maximum speed.

For this reason alone it is highly recommended to use a shunt in all vertical load applications or any case where the motors could be stopped quickly or back driven suddenly.

We offer both open frame and enclosed shunts in 100Watt and 200Watt capacities. The shunts are all automatic and get their power from the DC bus they are attached to. They simply need to be placed in parallel with the DC bus.

WARNING

1. Shunts cannot be placed in parallel with each other to increase capacity. The shunt with the slightly lower trigger voltage will trigger first while the other shunt never triggers at all. Please consult factory for information on how to deal with larger shunt requirements.
2. Shunts should always be placed between the motor input and any disconnect or e-stop relay to insure protection of the motor when power is not applied or e-stop relay contacts are open.

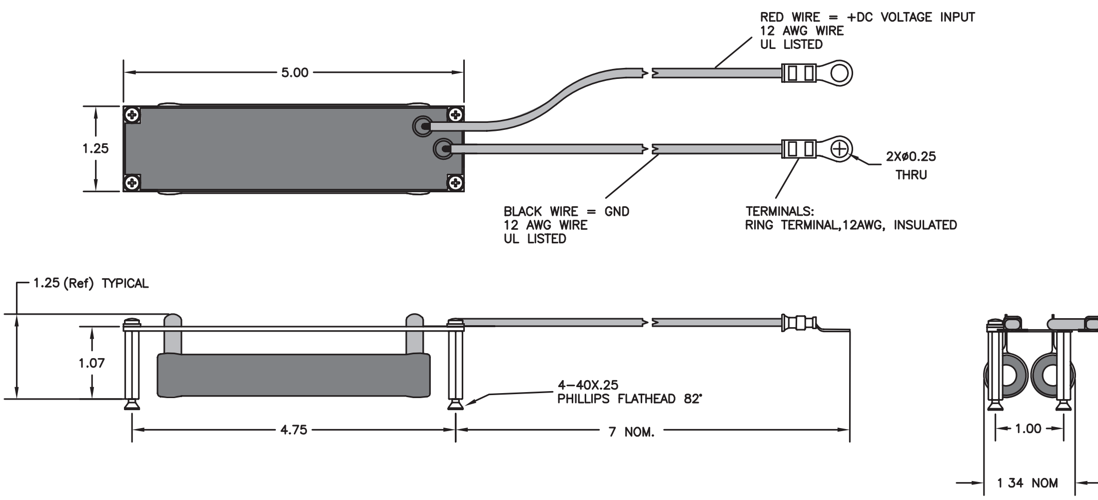
SHUNT42V100WOF and SHUNT42V200WOF

- Can be used with Power supplies that have an output of 48VDC or less
- Automatic Gate-On when Voltage Exceeds 49.5VDC
- Easy direct parallel connection to power supply

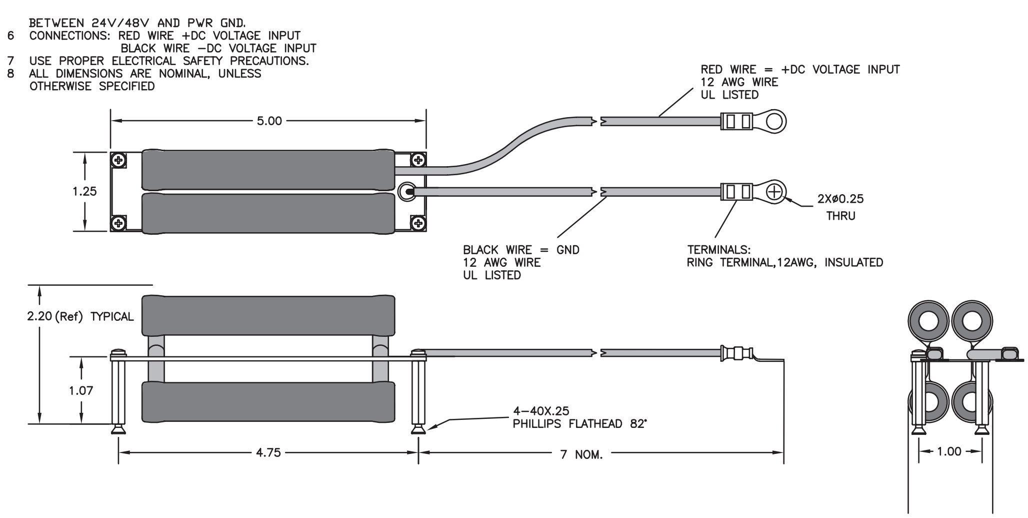
SHUNT42V100W-OF

SHUNT42V200W-OF

Part Number	TRIGGER VOLTAGE	DROP OUT VOLTAGE	CURRENT DRAW WHEN GATED ON	WATTS	EFFECTIVE BUS LOAD
SHUNT42V100WOF	49.5VDC RISING	48.5VDC FALLING	4 AMPS	100W	12.5 OHMS
SHUNT42V200WOF	49.5VDC RISING	48.5VDC FALLING	8 AMPS	200W	6.25 OHMS



SHUNT42V100WOF



SHUNT42V200WOF

Note: **Any time an E-Stop switch is placed on the DC power line to the motor, a Shunt MUST BE installed between the E-Stop switch and the motor connector to ensure protection against over voltage!**

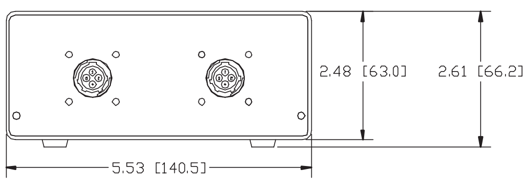
⚠ **Warning ! If the shunt is connected to an adjustable power supply, the output voltage MUST BE set at or below 48VDC. If the output voltage is sustained above the trip point of the shunt, over heating and damage may result.**

- Enclosed Shunt
- Matching 4 pin AMP connector to enclosed power supply.
- Automatically gate on at >=49.5VDC
- Powered from DC bus
- May be connected in parallel with any supply <=48VDC.

Part Number	TRIGGER VOLT-AGE	DROP OUT VOLT-AGE	CURRENT DRAW WHEN GATED ON	WATTS	EFFECTIVE BUS LOAD
SHUNT42V100W	49.5VDC RISING	48.5VDC FALLING	4 AMPS	100W	12.5 OHMS
SHUNT42V200W	49.5VDC RISING	48.5VDC FALLING	8 AMPS	200W	6.25 OHMS

Includes connector kits

Use with cable CBLDC1 below



* All sizes are given in inches, sizes in brackets are in mm

Note: **Any time an E-Stop switch is placed on the DC power line to the motor, a Shunt MUST BE installed between the E-Stop switch and the motor connector to ensure protection against over voltage!**

CBLDC1

Power Supply Cables

Part Number	Connection	Cable Type	Connector Type(s)	Length(s)
CBLAC1	AC Line Cord for power supply	Power	N/A	6 ft (1.8m)
CBLDC1	DC Cable for Enclosed Shunt	DC	4-Pin AMP	1.5ft (0.45M)
CBLSMYPWR-T	Multiple SM - power supply	Y	4 Pin AMP	2 ft (0.61m)

Moment Of Inertia:

A basic understanding of Moment of Inertia serves well in ensuring proper motor sizing. It is one thing to look at static points on torque curves, but it is altogether different when considering the dynamic aspects of loads being accelerated at high rates.

The Inertial mass of an object is a measure of its resistance to a change in its velocity.

The Moment of Inertia of an object is at a point of reference of rotation, which is at the pivot point or axis of rotation.

The Moment of Inertia can therefore be thought of as a measure of the resistance to any change in rotational speed.

For linear systems, the rate of change of speed, (acceleration) is proportional to the force applied. Double the mass and the force needs to be doubled for the same acceleration. Similarly for rotational systems, the angular acceleration of the load is proportional to the torque applied. Double the Moment of Inertia and the torque needs to be doubled for the same angular acceleration. Moment of Inertia is therefore a measure of a load's resistance to angular speed change; of how much effort (torque) is required to cause acceleration or deceleration.

Matching Motor To Load:

A common rule of thumb for SmartMotor™ sizing is that the load should have no more than 10 times the Moment of Inertia of the motor rotor that is driving it. This gives a good starting point and typically allows for safe sizing over a wide range of applications.

Since a rotating load wants to maintain the same velocity, then when a motor attempts to accelerated or decelerate the load, it must overcome the Moment of Inertia of that load by applying enough torque to accelerate it or decelerate it.

It takes more torque to change speed than it does to maintain a given speed.

In the same manner, for the motor to slow down a load, the load's Moment of Inertia will keep the motor going the same speed and will, in effect, back-drive the motor turning it into a generator.

In extreme cases, this can result in over-voltage damage to the Drive stage.

How to Improve Moment of Inertia Ratio Between Motor and Load :

Adding gear reduction to a motor gives it more leverage to prevent back driving and also gives it a better advantage in accelerating a load up to speed.

For any given change in gear reduction, you get a proportional change in speed and static torque but you get a squared change in acceleration and dynamic rate of change of torque. The result is that by adding gear ratio you gain a squared decrease in the ratio of Moment of Inertia between motor and load.

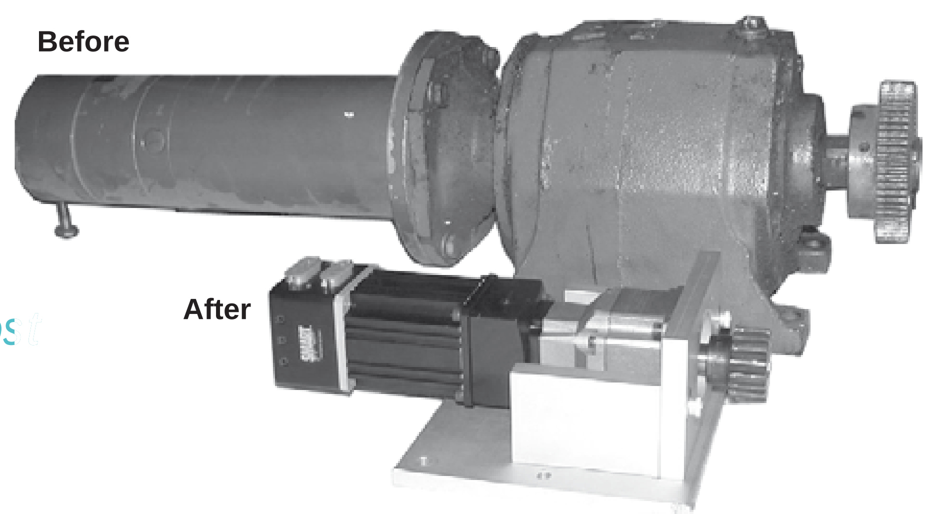
Therefore the motor has a greater advantage in both accelerating and decelerating the load. It adds protection against damage to the system as a whole.

Lower System Cost

To give an idea of how much effect you get from additional gear reduction, take a look at the example below. This is an actual photo of the before-and-after drive system of a given application. The larger motor with low gear reduction and larger pulley was replaced by the smaller Animatics SmartMotor™ with much higher gear reduction and smaller pulley. The result was a smoother operating machine with higher resolution and better acceleration, increasing throughout and improving quality.

Optimize gear reduction to improve load dynamics and motor efficiency & reduce system cost

Before



After

All units are precision ground planetary gear sets capable of sustained servo input speed. They can be ordered by themselves, or pre-mounted to the SmartMotors™ prior to shipment.

Each gearhead has a non-captive input pinion gear. This means the pinion is mounted onto the motor shaft and the gearhead is then mounted onto the motor flange.

Torque throughput on in-line (straight) gearheads are limited by input pinion diameters.

Typically the 7:1 ratio single-stage and 28:2 ratio two-stage gearheads have the higher torque ratings.

10:1 and 100:1 gearhead input pinions are very small and great care should be taken not to exceed maximum torque ratings for those gear ratios.

All right angle gearhead torque levels are limited by the right angle beveled gear sets. This is why all gear ratios show the same torque limits within that series.

All gearheads are limited to a maximum of 5000 RPM input pinion speed. This limit is due to differential speed across the input pinion bearings and lubrication flow. Exceeding 5000 RPM for any sustained amount of period will GREATLY decrease the life of the gearhead and will not be covered by warranty. However, for typical servo applications, there is no issue with reaching 5000RPM on each machine cycle peak speeds.

Please consult the factory for axial and radial load specifications. Load ratings are speed dependant and are charted across curves.

All specifications are subject to change without notice. Please consult the factory or website for latest data and CAD drawings.

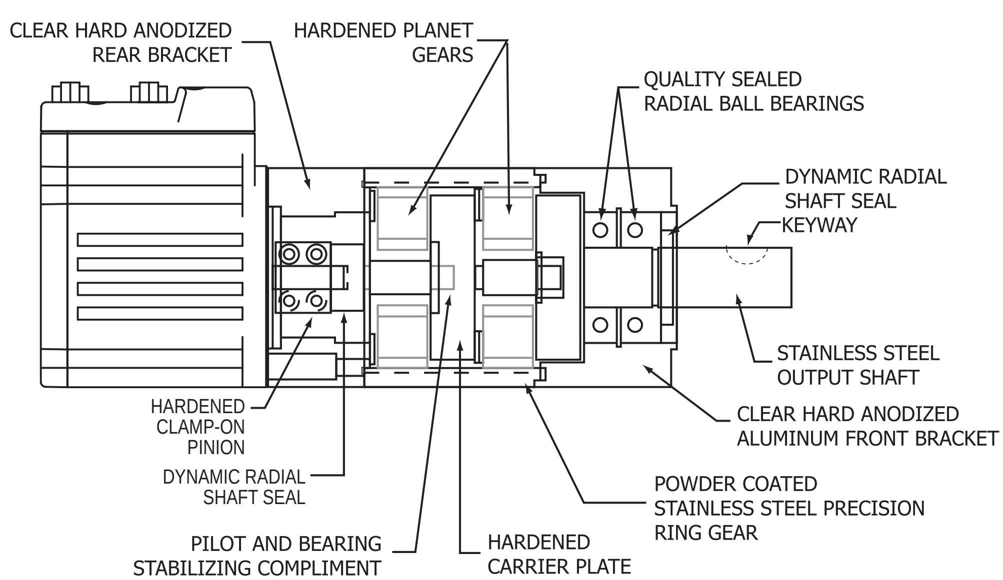
Gearhead Series	Backlash (arc-minutes)	
	Single Stage	Two Stage
High Performance (S)	3	7
OEM Series (SP)	6	10
Right Angle (RAP)	7	11

Note: These are the low backlash values.

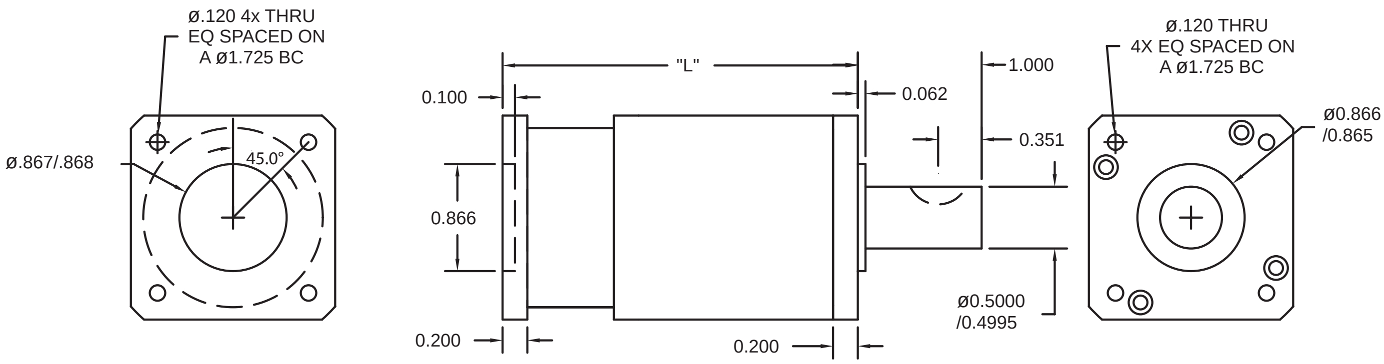
Animatics provides three series of gearheads.
The above chart is a quick reference to backlash specs.

Each gearhead is shipped with appropriate mounting hardware, fasteners, Allen key and pinion gap gauge.

2 Stage Gearheads Shown



DIMENSION "L"
Single Stage = 2.869±.015
Double Stage = 3.738±.015



NOTE: Dimensions in Inches

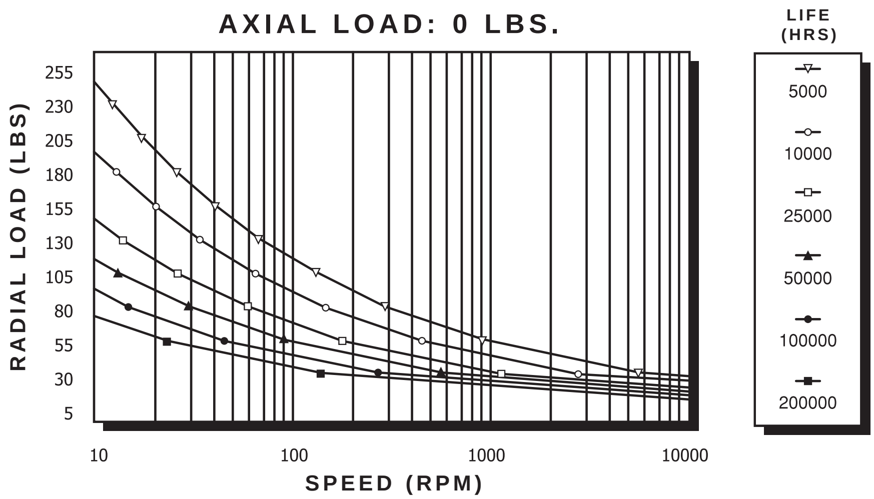
Part #	Ratio	Continuous output torque at 1500 rpm input (in-lbs)	Continuous output torque at 3500 rpm input (in-lbs)	Continuous output torque at 5000 rpm input (in-lbs)	Gearhead intertia at input (lb-in-sec^2)
Single Stage					
GH17P3	3:1	197	140	115	3.25×10^{-5}
GH17P4	4:1	177	136	116	1.60×10^{-5}
GH17P5.5	5.5:1	157	129	113	1.10×10^{-5}
GH17P7	7:1	143	122	110	9.56×10^{-6}
GH17P10	10:1	113	101	93	8.36×10^{-6}
Double Stage					
GH17P16	16:1	211	194	182	1.59×10^{-5}
GH17P22	22:1	216	201	193	1.10×10^{-6}
GH17P28	28:1	218	207	199	9.54×10^{-6}
GH17P40	40:1	220	212	207	8.35×10^{-6}
GH17P49	49:1	158	154	152	9.44×10^{-6}
GH17P55	55:1	183	177	175	8.31×10^{-6}
GH17P70	70:1	160	156	154	8.30×10^{-6}
GH17P100	100:1	122	120	119	8.29×10^{-6}

General Specifications

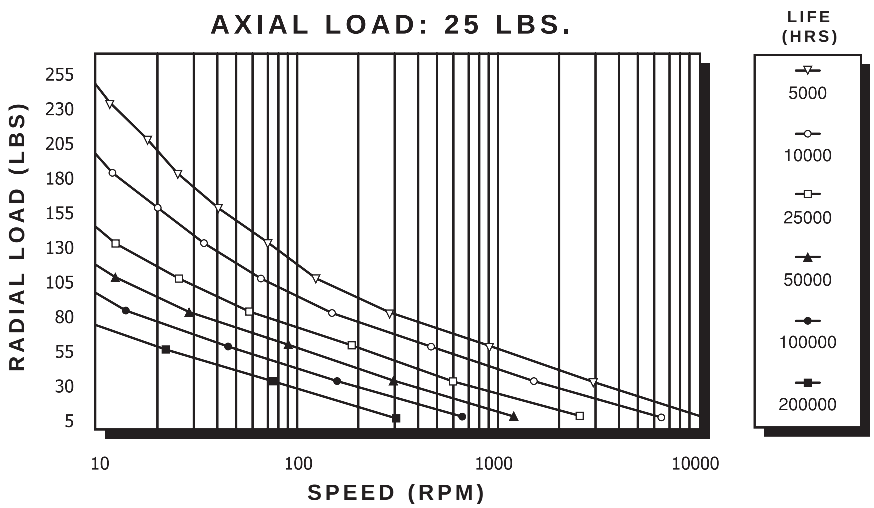
Construction Type	Ratio	Standard Backlash (arc-minutes)	Low Backlash (arc-minutes)	Efficiency	Weight (lbs)	Maximum Tested Input rpm
Single Stage	3:1 to 10:1	6	3	90%	1.14	5000
Double Stage	16:1 to 100:1	10	7	85%	1.62	5000

PEAK TORQUE: 15% above continuous rating. **NOTE**: Repeated peak torque loading may cause failure.

AXIAL LOAD: 0 LBS.



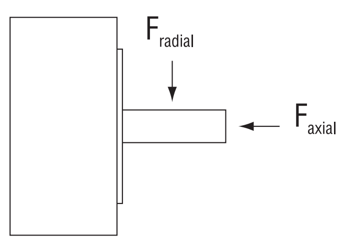
AXIAL LOAD: 25 LBS.



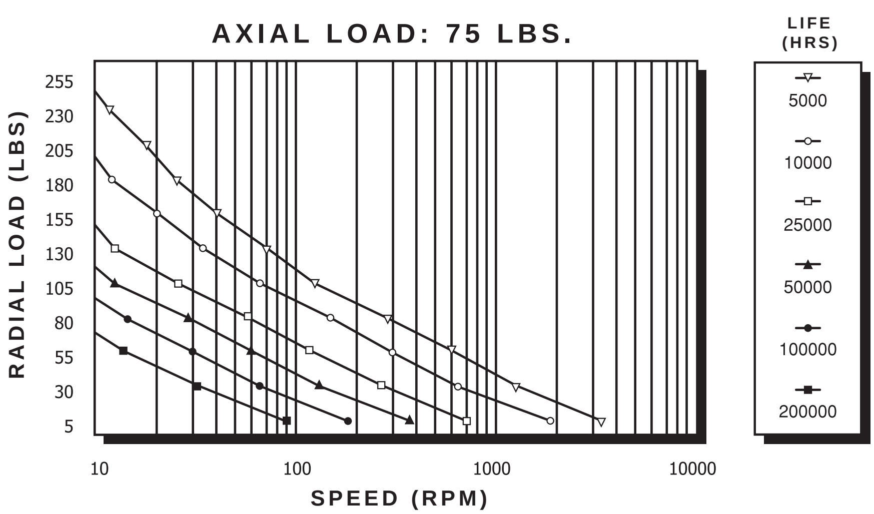
SPEED (RPM) refers to the gearheads output shaft speed.

LIFE (HRS) = (# of lifetime revolutions) ÷ (60 x rpm)

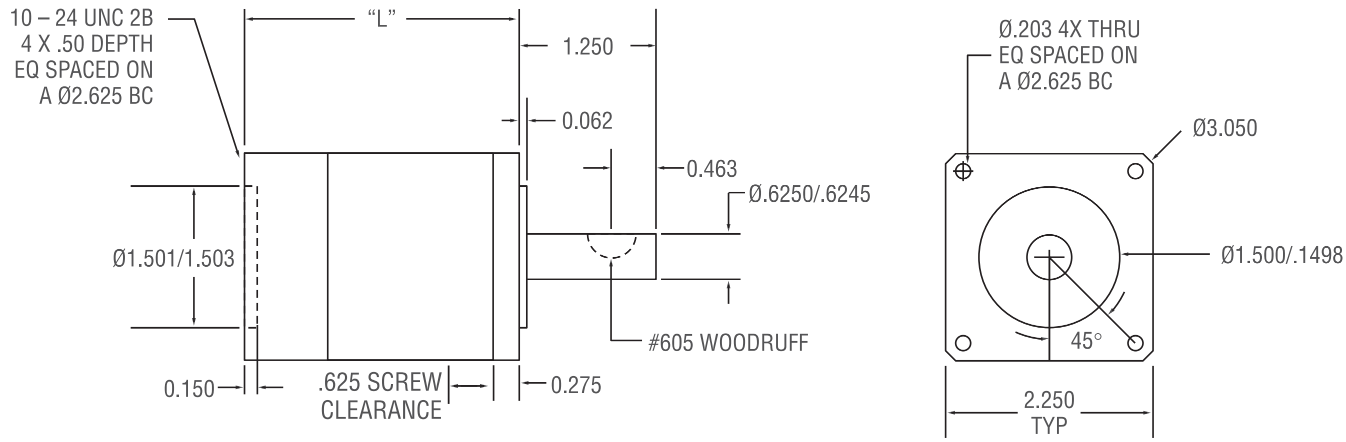
F_{RADIAL} is calculated at 1/2 the shaft length.



AXIAL LOAD: 75 LBS.



DIMENSION "L"
Single Stage (3:1 to 10:1) = 3.146± .015
Double Stage (16:1 to 100:1) = 4.249 ± .015



NOTE: Dimensions in inches

Part #	Ratio	Continuous output torque at 1500 rpm input (in-lbs)	Continuous output torque at 3500 rpm input (in-lbs)	Continuous output torque at 5000 rpm input (in-lbs)	Gearhead intertia at input (lb-in-sec^2)
Single Stage					
GH23P3	3:1	442	292	232	1.22 x 10^{-4}
GH23P4	4:1	410	294	242	5.24 x 10^{-5}
GH23P5.5	5.5:1	373	288	247	2.65 x 10^{-5}
GH23P7	7:1	344	279	245	1.93 x 10^{-5}
GH23P10	10:1	277	238	215	1.35 x 10^{-5}
Double Stage					
GH23P16	16:1	521	463	427	5.32 x 10^{-5}
GH23P22	22:1	536	490	460	2.70 x 10^{-5}
GH23P28	28:1	545	506	481	1.96 x 10^{-5}
GH23P40	40:1	553	525	506	1.36 x 10^{-5}
GH23P49	49:1	400	385	375	1.90 x 10^{-5}
GH23P55	55:1	460	443	432	1.34 x 10^{-5}
GH23P70	70:1	404	393	385	1.33 x 10^{-5}
GH23P100	100:1	308	303	298	1.33 x 10^{-5}

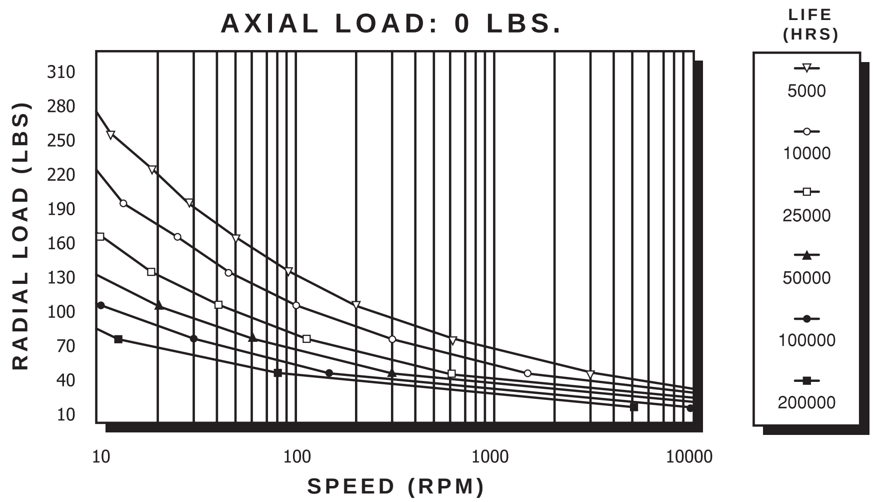
General Specifications

Construction Type	Ratio	Standard Backlash (arc-minutes)	Low Backlash (arc-minutes)	Efficiency	Weight (lbs)	Maximum Tested Input rpm
Single Stage	3:1 to 10:1	6	3	90%	2.29	5000
Double Stage	16:1 to 100:1	10	7	85%	3.42	5000

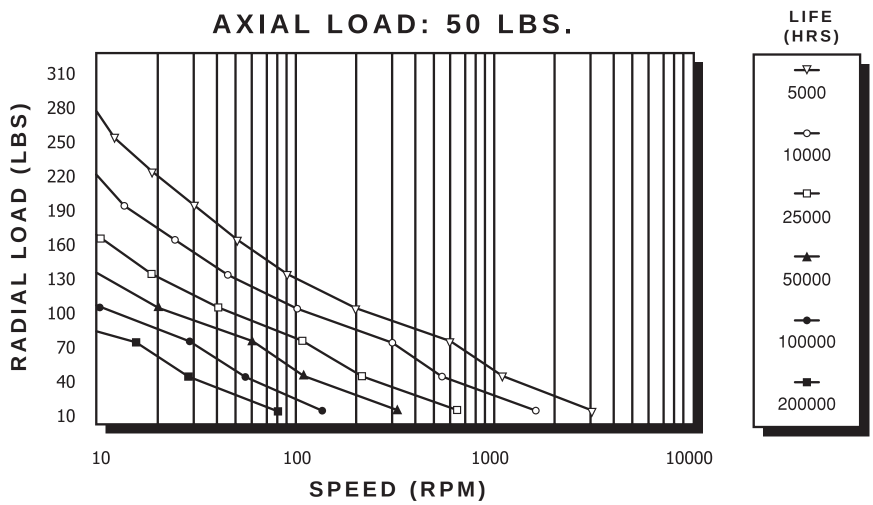
PEAK TORQUE: 15% above continuous rating. **NOTE:** Repeated peak torque loading may cause failure.

Animatics Corporation
tel: 408.748.8721 • fax: 408.748.8725 • www.animatics.com

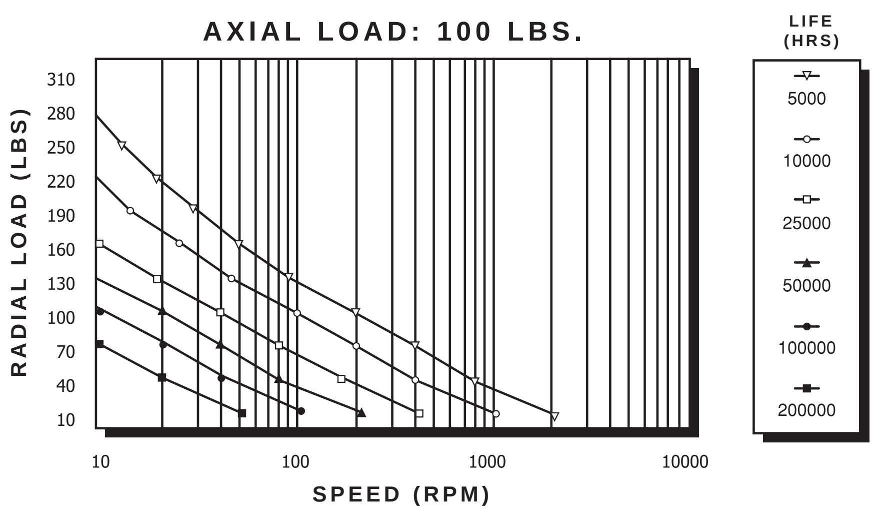
AXIAL LOAD: 0 LBS.



AXIAL LOAD: 50 LBS.



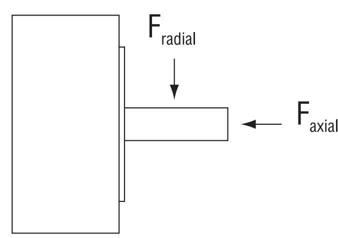
AXIAL LOAD: 100 LBS.



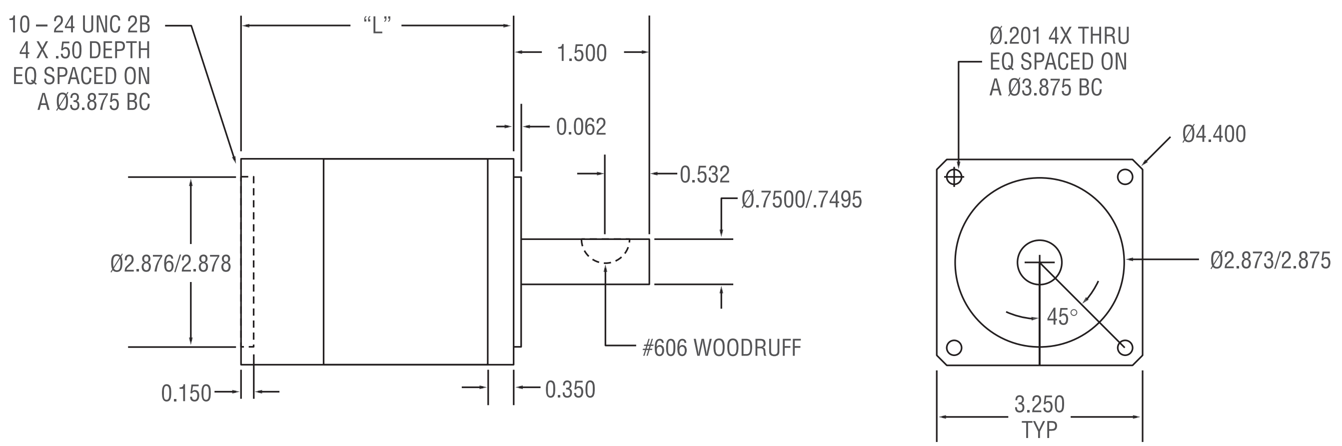
SPEED (RPM) refers to the gearheads output shaft speed.

LIFE (HRS) = (# of lifetime revolutions) ÷ (60 x rpm)

F_{RADIAL} is calculated at 1/2 the shaft length.



DIMENSION "L"
Single Stage (3:1 to 10:1) = 4.063± .015
Double Stage (16:1 to 100:1) = 5.431 ± .015



NOTE: Dimensions in inches

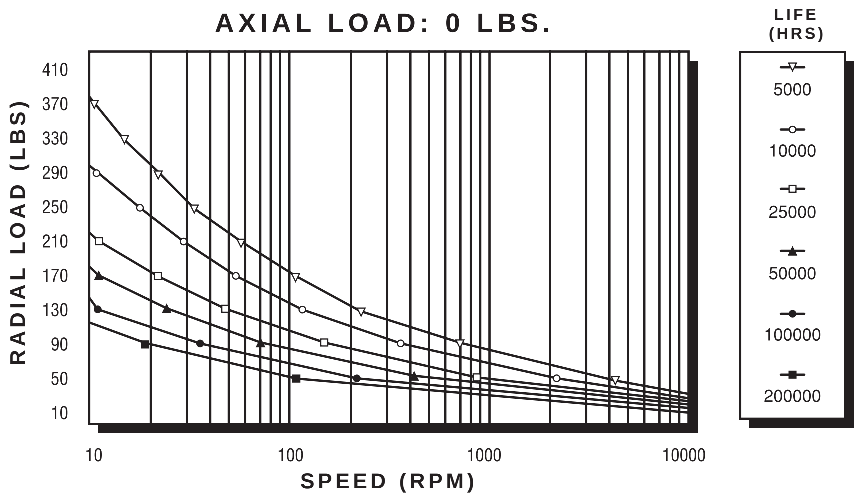
Part #	Ratio	Continuous output torque at 1500 rpm input (in-lbs)	Continuous output torque at 3500 rpm input (in-lbs)	Continuous output torque at 5000 rpm input (in-lbs)	Gearhead intertia at input (lb-in-sec^2)
Single Stage					
GH34P3	3:1	1010	615	475	6.77 x 10^{-4}
GH34P4	4:1	972	643	513	2.77 x 10^{-4}
GH34P5.5	5.5:1	913	657	543	1.51 x 10^{-4}
GH34P7	7:1	859	653	554	1.11 x 10^{-4}
GH34P10	10:1	707	575	505	7.90 x 10^{-5}
Double Stage					
GH34P16	16:1	1350	1145	1027	2.86 x 10^{-4}
GH34P22	22:1	1401	1234	1133	1.55 x 10^{-4}
GH34P28	28:1	1432	1293	1203	1.11 x 10^{-4}
GH34P40	40:1	1469	1362	1293	8.04 x 10^{-5}
GH34P49	49:1	1067	1010	971	1.11 x 10^{-4}
GH34P55	55:1	1228	1165	1123	7.94 x 10^{-5}
GH34P70	70:1	1081	1040	1010	7.90 x 10^{-5}
GH34P100	100:1	827	805	790	7.87 x 10^{-5}

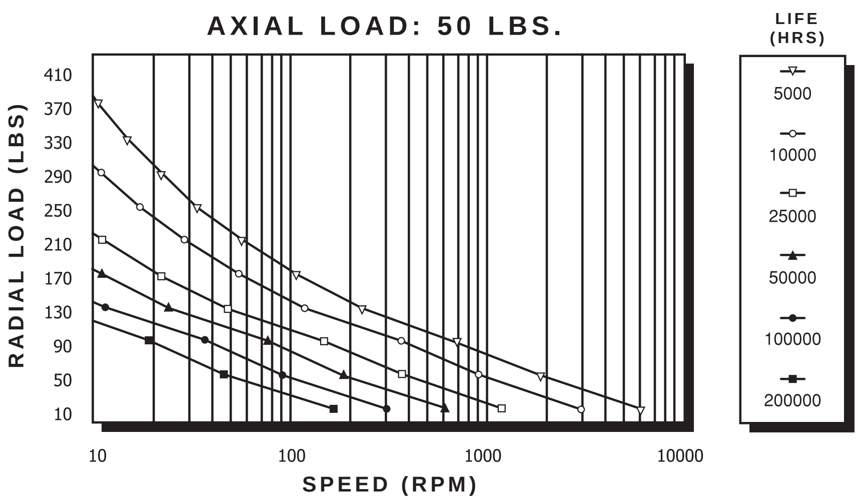
WARNING - Please read when using with SM3416DT-PLS2 Series Motor:
All gearheads above come standard with 3/8 inch diameter input shaft. For Gearheads which require 0.5 inch shaft input, please add "-0.5" to part number. Example: GH34P3-0.5 will give you a half inch input shaft diameter.

General Specifications

Construction Type	Ratio	Standard Backlash (arc-minutes)	Low Backlash (arc-minutes)	Efficiency	Weight (lbs)	Maximum Tested Input rpm
Single Stage	3:1 to 10:1	6	3	90%	5.67	5000
Double Stage	16:1 to 100:1	10	7	85%	8.41	5000

PEAK TORQUE: 15% above continuous rating. **NOTE:** Repeated peak torque loading may cause failure.

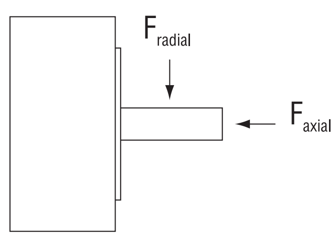


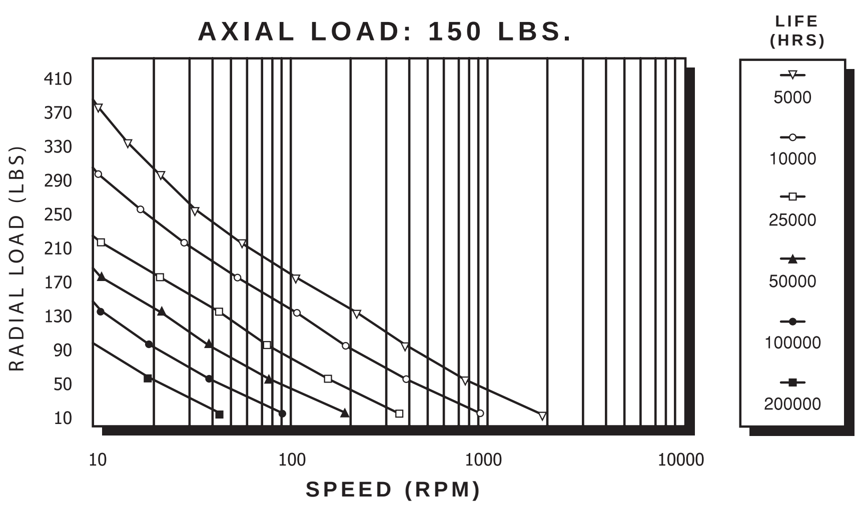


SPEED (RPM) refers to the gearheads output shaft speed.

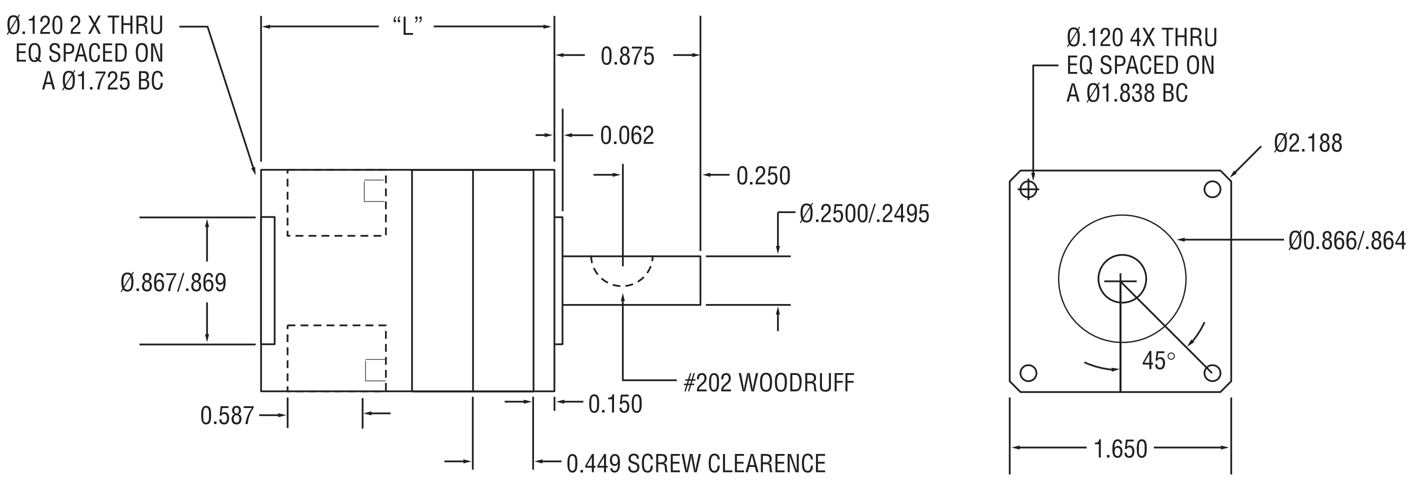
LIFE (HRS) = (# of lifetime revolutions) ÷ (60 x rpm)

F_{RADIAL} is calculated at 1/2 the shaft length.





DIMENSION "L"
Single Stage (4:1 to 10:1) = 2.17± .015
Double Stage (16:1 to 100:1) = 2.78 ± .015



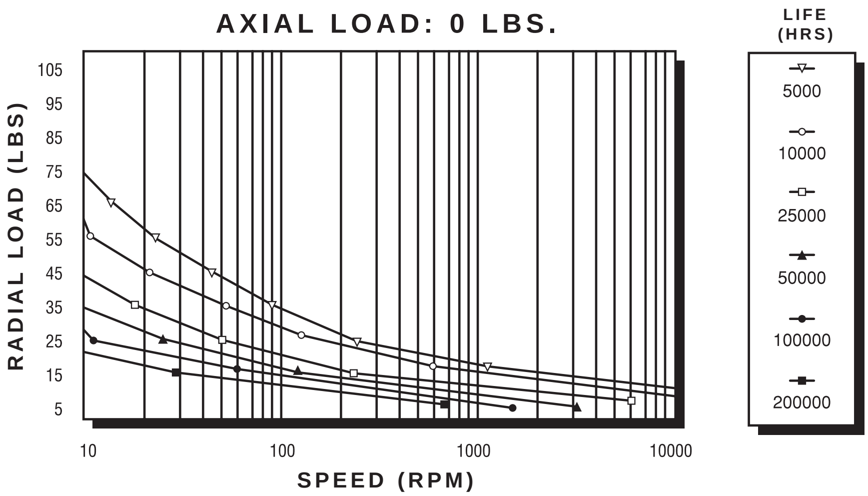
NOTE: Dimensions in inches

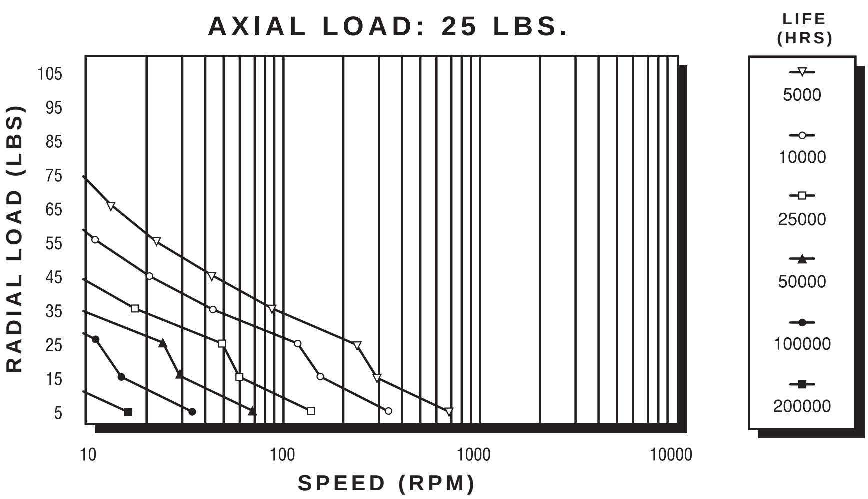
Part #	Ratio	Continuous output torque at 1500 rpm input (in-lbs)	Continuous output torque at 3500 rpm input (in-lbs)	Continuous output torque at 5000 rpm input (in-lbs)	Gearhead intertia at input (lb-in-sec^2)
Single Stage					
GH17SP004	4:1	75	57	49	1.28 x 10^{-5}
GH17SP007	7:1	60	51	46	7.65 x 10^{-6}
GH17SP010	10:1	48	43	39	6.69 x 10^{-6}
Double Stage					
GH17SP016	16:1	81	74	70	1.27 x 10^{-5}
GH17SP028	28:1	83	79	76	7.63 x 10^{-6}
GH17SP049	49:1	61	59	58	7.55 x 10^{-6}
GH17SP070	70:1	61	60	59	6.64 x 10^{-6}
GH17SP100	100:1	47	46	46	6.63 x 10^{-6}

General Specifications

Construction Type	Ratio	Standard Backlash (arc-minutes)	Low Backlash (arc-minutes)	Efficiency	Weight (lbs)	Maximum Tested Input rpm
Single Stage	4:1 to 10:1	12	6	95%	0.59	5000
Double Stage	16:1 to 100:1	16	10	90%	0.88	5000

PEAK TORQUE: 15% above continuous rating. **NOTE:** Repeated peak torque loading may cause failure.

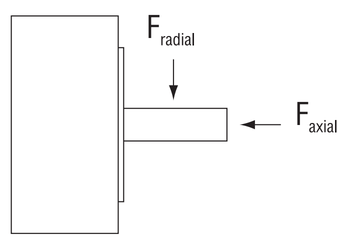


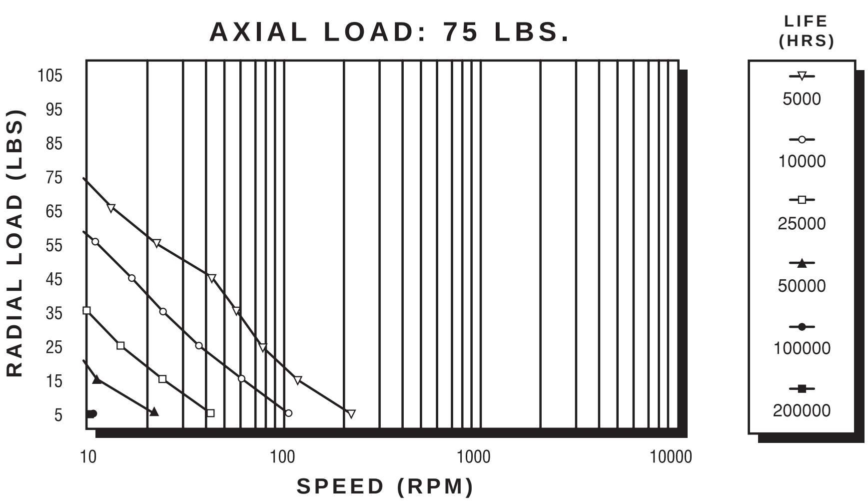


SPEED (RPM) refers to the gearheads output shaft speed.

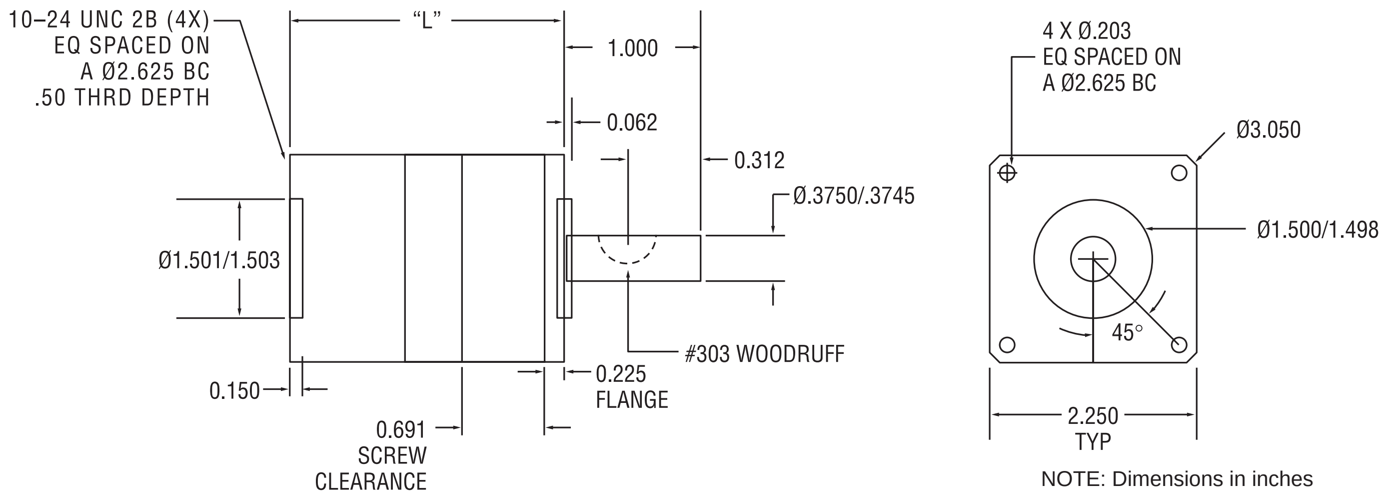
LIFE (HRS) = (# of lifetime revolutions) ÷ (60 x rpm)

F_{RADIAL} is calculated at 1/2 the shaft length.





DIMENSION "L"
Single Stage (4:1 to 10:1) = 2.436± .015
Double Stage (16:1 to 100:1) = 3.029 ± .015



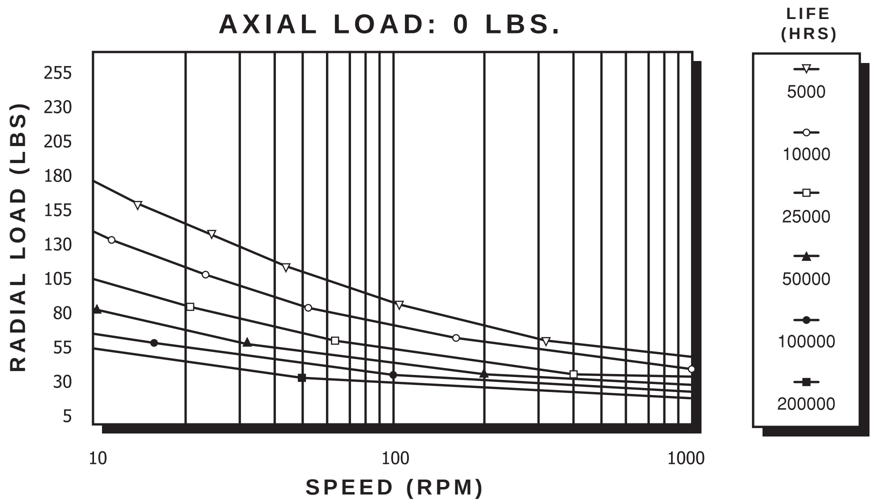
NOTE: Dimensions in inches

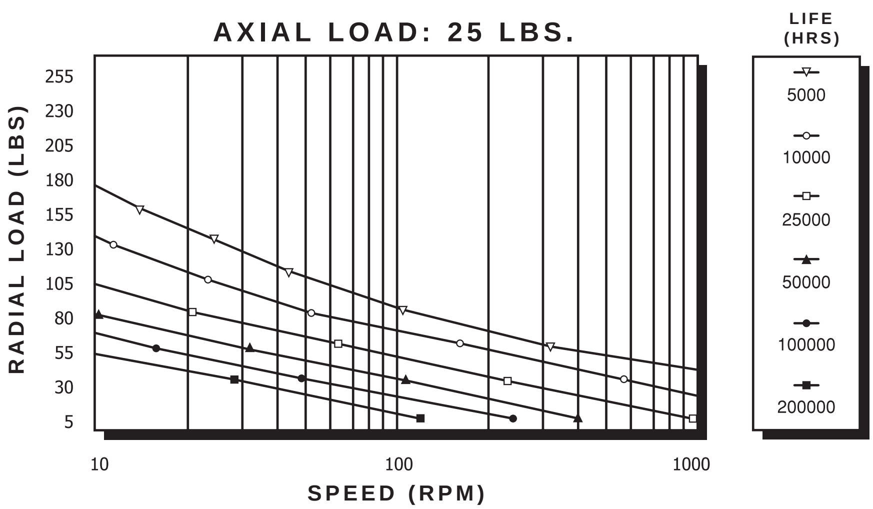
Part #	Ratio	Continuous output torque at 1500 rpm input (in-lbs)	Continuous output torque at 3500 rpm input (in-lbs)	Continuous output torque at 5000 rpm input (in-lbs)	Gearhead intertia at input (lb-in-sec^2)
Single Stage					
GH23SP004	4:1	185	133	109	4.19 x 10^{-5}
GH23SP007	7:1	155	126	110	1.54 x 10^{-5}
GH23SP010	10:1	125	107	97	1.08 x 10^{-5}
Double Stage					
GH23SP016	16:1	214	190	175	4.26 x 10^{-5}
GH23SP028	28:1	223	208	197	1.57 x 10^{-5}
GH23SP049	49:1	164	158	154	1.52 x 10^{-5}
GH23SP070	70:1	166	161	158	1.06 x 10^{-5}
GH23SP100	100:1	126	124	122	1.06 x 10^{-5}

General Specifications

Construction Type	Ratio	Standard Backlash (arc-minutes)	Low Backlash (arc-minutes)	Efficiency	Weight (lbs)	Maximum Tested Input rpm
Single Stage	4:1 to 10:1	12	6	95%	1.55	5000
Double Stage	16:1 to 100:1	16	10	90%	1.95	5000

PEAK TORQUE: 15% above continuous rating. **NOTE:** Repeated peak torque loading may cause failure.

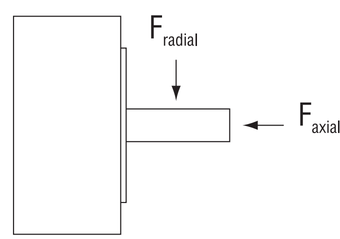


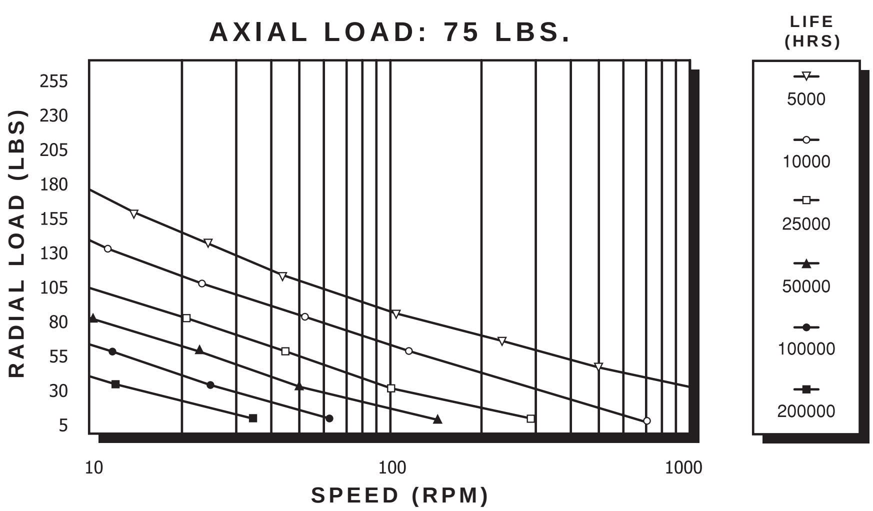


SPEED (RPM) refers to the gearheads output shaft speed.

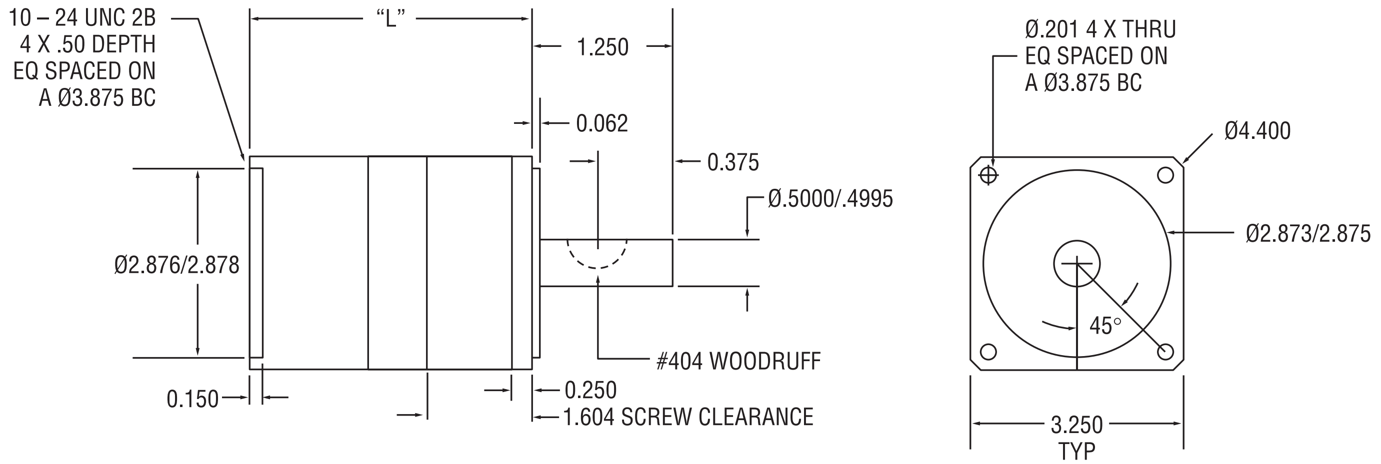
LIFE (HRS) = (# of lifetime revolutions) ÷ (60 x rpm)

F_{RADIAL} is calculated at 1/2 the shaft length.





DIMENSION "L"
Single Stage (4:1 to 10:1) = 3.229± .015
Double Stage (16:1 to 100:1) = 4.087 ± .015



NOTE: Dimensions in inches

Part #	Ratio	Continuous output torque at 1500 rpm input (in-lbs)	Continuous output torque at 3500 rpm input (in-lbs)	Continuous output torque at 5000 rpm input (in-lbs)	Gearhead intertia at input (lb-in-sec^2)
			Single Stage		
GH34SP004	4:1	529	350	279	1.28×10^{-4}
GH34SP007	7:1	467	355	301	7.65×10^{-5}
GH34SP010	10:1	384	313	275	6.69×10^{-5}
			Double Stage		
GH34SP016	16:1	667	566	508	1.27×10^{-4}
GH34SP028	28:1	670	639	595	7.63×10^{-5}
GH34SP049	49:1	528	499	480	7.55×10^{-5}
GH34SP070	70:1	534	514	499	6.64×10^{-5}
GH34SP100	100:1	409	398	391	6.63×10^{-5}

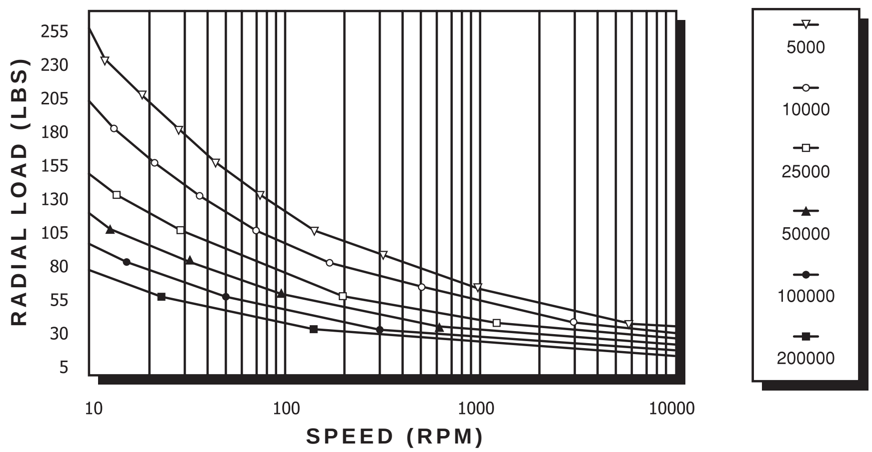
WARNING - Please read when using with SM3416DT-PLS2 Series Motor:
All gearheads above come standard with 3/8 inch diameter input shaft. For gearheads which require 0.5 inch shaft input, please add "-0.5" to part number. Example: GH34SP004-0.5 will give you a half inch input shaft diameter.

General Specifications

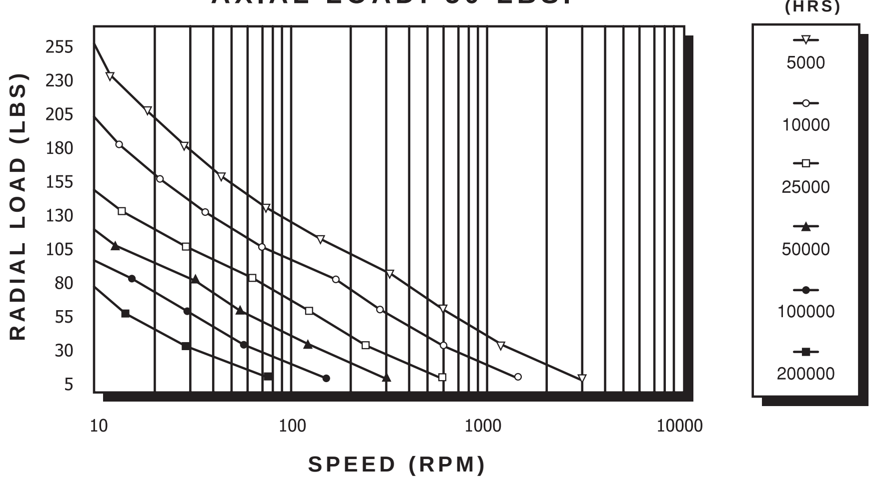
Construction Type	Ratio	Standard Backlash (arc-minutes)	Low Backlash (arc-minutes)	Efficiency	Weight (lbs)	Maximum Tested Input rpm
Single Stage	4:1 to 10:1	12	6	95%	3.67	5000
Double Stage	16:1 to 100:1	16	10	90%	5.10	5000

PEAK TORQUE: 15% above continuous rating. **NOTE:** Repeated peak torque loading may cause failure.

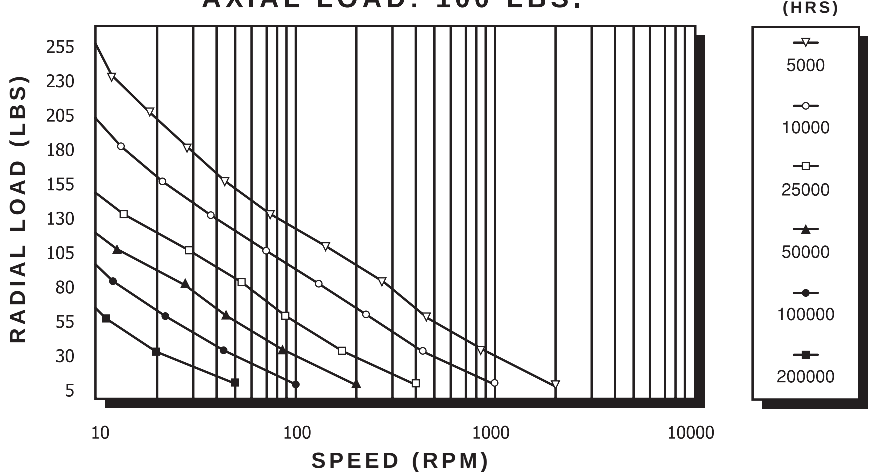
AXIAL LOAD: 0 LBS.



AXIAL LOAD: 50 LBS.



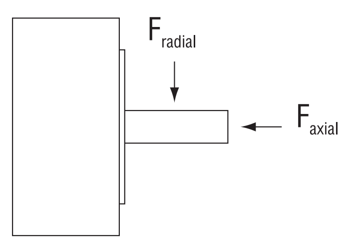
AXIAL LOAD: 100 LBS.

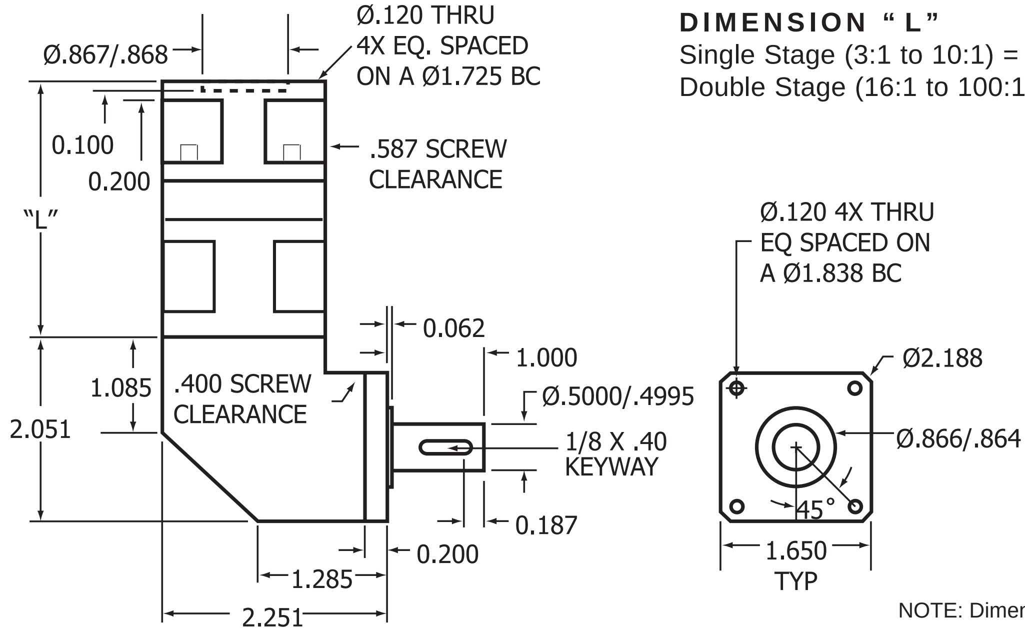


SPEED (RPM) refers to the gearheads output shaft speed.

LIFE (HRS) = (# of lifetime revolutions) ÷ (60 x rpm)

F_{RADIAL} is calculated at 1/2 the shaft length.





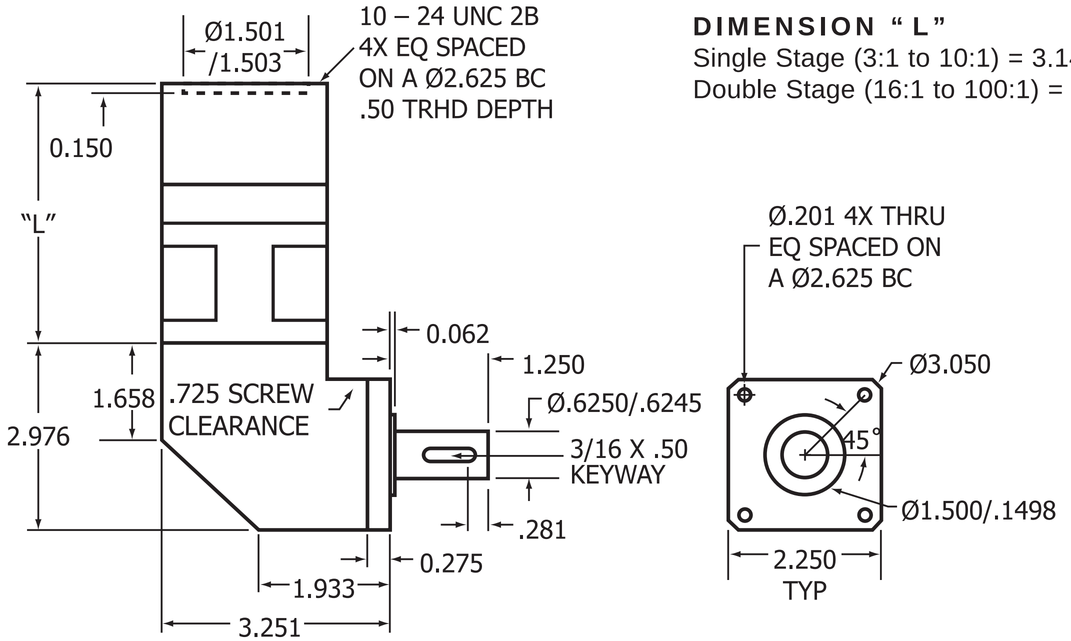
DIMENSION "L"
Single Stage (3:1 to 10:1) = 2.869 ± .015
Double Stage (16:1 to 100:1) = 3.738 ± .015

NOTE: Dimensions in inches

Part #	Ratio	Continuous output torque at 1500 rpm input (in-lbs)	Continuous output torque at 3500 rpm input (in-lbs)	Continuous output torque at 5000 rpm input (in-lbs)	Gearhead intertia at input (lb-in-sec²)
Single Stage					
GH17RAP3	3:1	80	80	80	4.96×10^{-5}
GH17RAP5.5	5.5:1	80	80	80	1.61×10^{-5}
GH17RAP7	7:1	80	80	80	1.27×10^{-5}
GH17RAP10	10:1	80	80	80	9.90×10^{-6}
Double Stage					
GH17RAP16	16:1	100	100	100	1.65×10^{-5}
GH17RAP22	22:1	100	100	100	1.13×10^{-5}
GH17RAP55	55:1	100	100	100	8.36×10^{-6}
GH17RAP100	100:1	100	100	100	8.31×10^{-6}

Construction Type	Ratio	Standard Backlash (arc-minutes)	Low Backlash (arc-minutes)	Efficiency	Weight (lbs)	Maximum Tested Input rpm
Single Stage	3:1 to 10:1	10	7	90%	1.96	5000
Double Stage	16:1 to 100:1	14	11	85%	2.44	5000

PEAK TORQUE: 15% above continuous rating. **NOTE:** Repeated peak torque loading may cause failure.

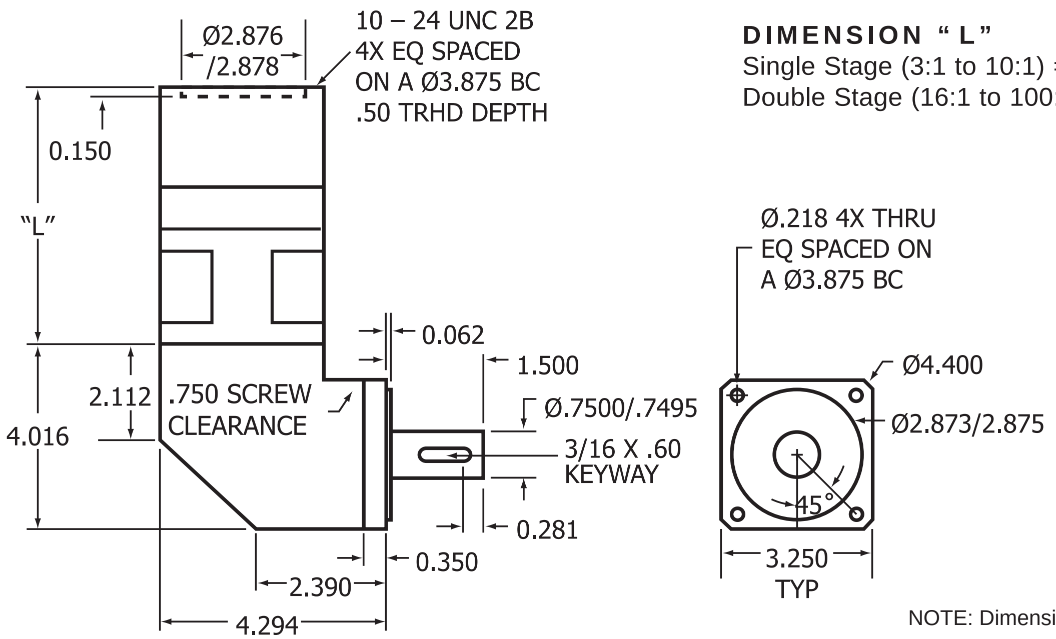


DIMENSION "L"
Single Stage (3:1 to 10:1) = 3.146 ± .015
Double Stage (16:1 to 100:1) = 4.249 ± .015

Part #	Ratio	Continuous output torque at 1500 rpm input (in-lbs)	Continuous output torque at 3500 rpm input (in-lbs)	Continuous output torque at 5000 rpm input (in-lbs)	Gearhead intertia at input (lb-in-sec^2)
Single Stage					
GH23RAP3	3:1	280	280	232	2.94 x 10^{-4}
GH23RAP5.5	5.5:1	280	280	247	7.77 x 10^{-5}
GH23RAP10	10:1	277	238	215	2.90 x 10^{-5}
Double Stage					
GH23RAP16	16:1	350	350	350	5.93 x 10^{-5}
GH23RAP22	22:1	350	350	350	3.02 x 10^{-5}
GH23RAP55	55:1	350	350	350	1.39 x 10^{-5}
GH23RAP100	100:1	308	303	298	1.35 x 10^{-5}

Construction Type	Ratio	Standard Backlash (arc-minutes)	Low Backlash (arc-minutes)	Efficiency	Weight (lbs)	Maximum Tested Input rpm
Single Stage	3:1 to 10:1	10	7	90%	4.87	5000
Double Stage	16:1 to 100:1	14	11	85%	6.00	5000

PEAK TORQUE: 15% above continuous rating. **NOTE:** Repeated peak torque loading may cause failure.



DIMENSION "L"
Single Stage (3:1 to 10:1) = 4.063 ± .015
Double Stage (16:1 to 100:1) = 5.431 ± .015

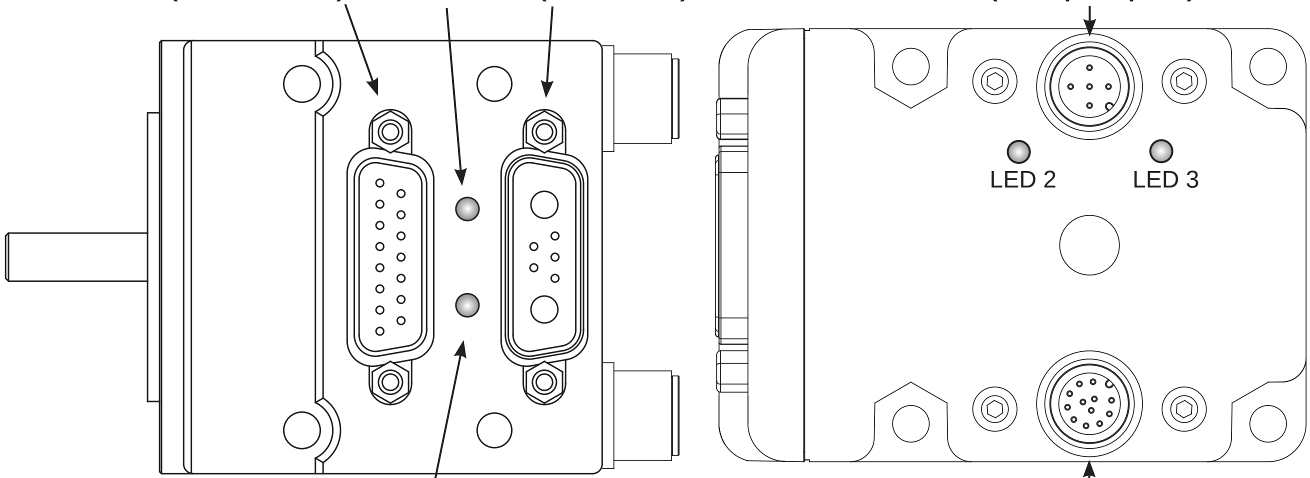
NOTE: Dimensions in inches

Part #	Ratio	Continuous output torque at 1500 rpm input (in-lbs)	Continuous output torque at 3500 rpm input (in-lbs)	Continuous output torque at 5000 rpm input (in-lbs)	Gearhead intertia at input (lb-in-sec^2)
Single Stage					
GH34RAP3	3:1	525	525	475	1.20 x 10^{-3}
GH34RAP5.5	5.5:1	525	525	525	3.08 x 10^{-4}
GH34RAP10	10:1	525	525	505	1.26 x 10^{-4}
Double Stage					
GH34RAP16	16:1	656	656	656	3.05 x 10^{-4}
GH34RAP22	22:1	656	656	656	1.65 x 10^{-4}
GH34RAP55	55:1	656	656	656	8.10 x 10^{-5}
GH34RAP100	100:1	656	656	656	7.92 x 10^{-5}

WARNING - Please read when using with SM3416DT-PLS2 Series Motor:
All gearheads above come standard with 3/8 inch diameter input shaft. For gearheads which require 0.5 inch
shaft input, please add "-0.5" to part number. Example: GH34RAP3-0.5 will give you a half inch input shaft diameter.

Construction Type	Ratio	Standard Backlash (arc-minutes)	Low Backlash (arc-minutes)	Efficiency	Weight (lbs)	Maximum Tested Input rpm
Single Stage	3:1 to 10:1	10	7	90%	11.89	5000
Double Stage	16:1 to 100:1	14	11	85%	14.62	5000

PEAK TORQUE: 15% above continuous rating. **NOTE:** Repeated peak torque loading may cause failure.



LED Status Power-up:

with no program

the travel limit inputs are not grounded:

LED0 will be solid RED indicating the motor is in a fault state due travel limit fault.

LED1 will be off

LED Status Power-up:

with no program

and the travel limits are hard wired to ground:

LED0 will be solid red for 500mseconds and then begin flashing Green.

LED1 will be off

LED Status Power-up:

with a program that only disables travel limits and nothing else

LED0 will be solid red for 500mseconds and then begin flashing Green.

LED1 will be off

LED0: Drive Status

OFF	:No Power
Solid Green	:Drive On
Flashing Green	:Drive Off
Flashing Red	:Watchdog Fault
Solid Red	:Major Fault
Alt. Red/Green	:In Boot Load, Needs Firmware

LED1: Trajectory Status

OFF	:Not Busy
Solid Green	:Drive On, Trajectory In Progress

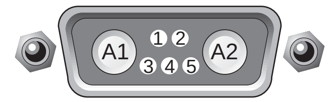
LED2 CAN Bus Network Fault (Red LED)

Off	:No Error
Single Flash.	:At least One Error exceeded Limit
Double Flash	:Heartbeat or Guard Error
Solid	:Busy Off State

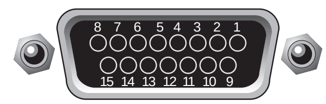
LED3: CAN Bus Network Status (Green LED)

Blinking	:Pre-Operational State, (during boot-up)
Solid	:Normal Operation
Single	:Device is in Stopped State

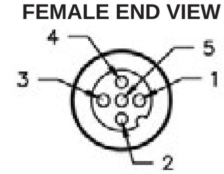
PIN	MAIN POWER	Specifications:		P1
1	I/O – 6 "G" command or GP	25mAmpSink 25mAmp source 10Bit 0-5VDC A/D	Redundant connection on I/O connector	**7W2 Combo D-sub Connectors**
2	RS-232 Transmit	50mAmps Max		
3	RS-232 Transmit	Com(0)	115.2KBaud Max	
4	RS-232 Transmit	Com(0)	115.2KBaud Max	
5	RS-232 Ground			
A1	Main Power: +20-48VDC			
A2	Ground			



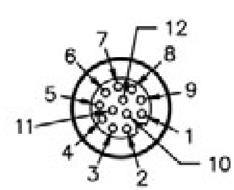
PIN	I/O CONNECTOR (5VTTL I/O)	Specifications:		P2
1	I/O – 0 GP or Enc. A or Step Input	25mAmpSink 25mAmp source 10Bit 0-5VDC A/D	1.5MHz max as Enc or Step input	**P2 DB-15 D-sub Connector**
2	I/O – 1 GP or Enc. B or Dir. Input	25mAmpSink 25mAmp source 10Bit 0-5VDC A/D	1.5MHz max as Enc. Or Dir. Input	
3	I/O – 2 Positive Over Travel or GP	25mAmpSink 25mAmp source 10Bit 0-5VDC A/D		
4	I/O – 3 Negative Over Travel or GP	25mAmpSink 25mAmp source 10Bit 0-5VDC A/D		
5	I/O – 4 GP or RS-485 A Com(1)	25mAmpSink 25mAmp source 10Bit 0-5VDC A/D	115.2KBaud Max	
6	I/O – 5 GP or RS-485 B Com(1)	25mAmpSink 25mAmp source 10Bit 0-5VDC A/D	115.2KBaud Max	
7	I/O – 6 "G" command or GP	25mAmpSink 25mAmp source 10Bit 0-5VDC A/D	Redundant connection on Main Power Connector	
8	Phase A Encoder Output			
9	Phase B Encoder Output			
10	RS-232 Transmit Com(0)		115.2KBaud Max	
11	RS-232 Transmit Com(0)		115.2KBaud Max	
12	+5VDC Out	50mAmps Max		
13	Ground			
14	Ground			
15	Main Power: +20-48VDC	IF-DE Option, Control Power separate from Main Power		



PIN	GND-I/O	Connection:	P3
1	NC	NC	**M12 5-PIN FEMALE END VIEW**
2	NC	NC	
3	GND_CAN	Isolated CAN ground	
4	CAN-H	1M Baud max	
5	CAN-L	1M Baud max	



PIN	Isolated 24VDC I/O Connector	Max Load (sourcing)	P4
1	IO – 16 GP	150mAmps	**M12 5-PIN FEMALE END VIEW**
2	IO – 17 GP	150mAmps	
3	IO – 18 GP	150mAmps	
4	IO – 19 GP	150mAmps	
5	IO – 20 GP	300mAmps	
6	IO – 21 GP	300mAmps	
7	IO – 22 GP	300mAmps	
8	IO – 23 GP	300mAmps	
9	IO – 24 GP	300mAmps	
10	IO – 25 GP	300mAmps	
11	+24Volts Input		
12	GND-I/O		



Animatics Corporation
tel: 408.748.8721 • fax: 408.748.8725 • www.animatics.com

Power & Encoder		
Drive Power:	24-48VDC	
Control Power:	24-48VDC (must be supplied separately when DE option is ordered)	
Expanded I/O:	24VDC (must be supplied)	
Commutation:	Trapezoidal	
	Enhanced Trapezoidal based on Encoder Position	
	Sinusoidal	
Encoder Resolution	23 Frame: 4000 (Class 5)	
	34 Frame: 8000 (Class 5)	

Processor/Clocks:		
Processor Clock Speed:	32MHz	
PWM Switching Frequency:	16KHz	
CPU Regulator Frequency:	140KHz +/-10% load dependant	
Drive Stage Regulator:	100MHz	
PID Update Rates:		
PID1	16KHz	62.5 µsec update rate
(Default) PID2	8KHz	125 µsec update rate
PID4	4KHz	250 µsec update rate
PID8	2KHz	500 µsec update rate

Programming:	
Code:	Command Interpretive Text Based
Program:	32K Program/32K Data Storage
Subroutines:	up to 1000
Stack Pointers:	10 deep for GOSUB RETURNs
	10 Deep for Nested SWITCH statements
	10 Deep for Nested WHILE Statements
	3 Deep for Nested Trig Functions

Communications:		
RS-232:	2400, 4800, 9600, 19200, 38400, 57600, 115200 Baud	9600 default
RS-485:	2400, 4800, 9600, 19200, 38400, 57600, 115200 Baud	9600 default
(Optional) CAN Bus:	20K, 50K, 125K, 250K, 500K, 800K, 1M Baud	125000 default

On Board I/O	
7 Channels:	5V CMOS Logic, Configurable as Inputs/Outputs/10-Bit Analog Inputs
Input Impedance:	5K pull-up, must drive down w/>=1mAmps to pull low
As Analog Input:	0-5VDC 10Bit A/D
Output Sourcing:	25mAmps
Output Sinking:	25mAmps
Note:	Max. 150mAmps total all outputs driven
Ports A and B as Enc. Input:	1MHz maximum input frequency
DB-15 Pin#	I/O Signal
1	Port A
2	Port B
3	Port C
4	Port D
5	Port E
6	Port F
7	Port G
8	Enc A Out
9	Enc B Out

Optional Expanded I/O:	Isolated, Requires 24VDC at the connector	
10 Channels:	Configurable as Inputs/Outputs, 10-Bit Analog Inputs	
Input Impedance:	100K	
12 Pin Connector Pin #	Max Load	
1	I/O 0	150mAmp max load
2	I/O 1	150mAmp max load
3	I/O 2	150mAmp max load
4	I/O 3	150mAmp max load
5	I/O 4	300mAmp max load
6	I/O 5	300mAmp max load
7	I/O 6	300mAmp max load
8	I/O 7	300mAmp max load
9	I/O 8	300mAmp max load
10	I/O 9	300mAmp max load
11	24Volts In	
12	GND-I/O	

Reference Key:

- is the IO Bit Number

m - is the mask value of which bits are effected

W - defines it as a word (16 bits)

expression - a number, variable or math expression up to 128 characters

value - a number, variable or math expression with one operand

constant - means a fixed integer

Communication Commands:

ADDR=*expression*	Set motor's serial communications address. Applies for both RS232 and RS485
BAUD (x)=*y*	This allows for COM0 or COM1 to be changed, x is the channel (0 or 1) and y is baud rate
CADDR=*expression*	Set CAN address, can be different from serial address, default is 63
CBAUD=*expression*	Set CAN baud rate, default is 125000
CCHN (*RS2,0***)**	Close communication channel command
ECHO	Must be used to insure all data received in one motor will be echoed to next motor
ECHO_OFF	Default, turn communication's echo off
GETCHR	Get the next character from channel 0
GETCHR1	Get the next character from channel 1
LEN	Number of characters in channel 0 buffer
LEN1	Number of characters in channel 1 buffer
RCADDR	Reports CAN address
RCBAUD	Reports CAN baud rate
RCHN (0)	Report channel 0 error bits
RCHN (1)	Report channel 1 error bits
SILENT	Ignore print commands to channel 0 from user program
SILENT1	Ignore print commands to channel 1 from user program
SLEEP	Ignore commands for channel 0 except the WAKE command
SLEEP1	Ignore commands for channel 1 except the WAKE command
STDOUT=0	Sets internal report commands to RS232 (default)
STDOUT=1	Sets internal report commands to RS485
TALK	Enable prints for channel 0 from user program
TALK1	Enable prints for channel 1 from user program
WAKE	Wake for channel 0
WAKE1	Wake for channel 1

Program Flow Commands:

CASE *expression*	Switch case statement
C*constant*	Subroutine label, e.g. C10 for subroutine 10, must have a RETURN for each C label
DEFAULT	Default action for switch case statement
DITR (*x***)**	Individual interrupt disable where x is interrupt # 0-8
ELSEIF *expression*	Used for IF statements to test another condition, if expression is true, then execute code
END	End program execution
ENDIF	End statement for IF code structures
ENDS	Command for end of switch case statement
GOSUB (*value***)**	Call a subroutine, value up to 999
GOTO (*value***)**	Jump program execution to a label, value up to 999
IF *expression*	Conditional Test, expression can be multiple math operations
ITR (*x, status_wrd#, bit#, s ,label#***)**	Interrupt setup
ITRD	Global interrupt scanner disable
ITRE	Global interrupt scanner enable
LOOP	Loop command for while loops
PAUSE	Pause program execution, used for interrupts
RESUME	Resume program execution
RETURN	Return from subroutine
RETURNI	Return from interrupt
RUN	Start program execution
RUN?	Wait at this point for RUN command before pro gram starts to execute
STACK	Reset the stack for GOSUB commands
SWITCH *expression*	Switch case statement
TWAIT	Wait for trajectory to complete, only used in pro gram
WAIT=*expression*	Set wait time in milliseconds
WHILE *expression*	
LOOP	While loop format

I/O Commands:

EIGN (*#***)**	Assign a single I/O point as general use input
EIGN (W,0)	Assign all local I/O as general use inputs
EIGN (W,0,12)	Assign inputs 2 and 3 as general use inputs at once (disabling over-travel limits)
EIGN (W,*0***,***m***)**	Assign a masked word-sized set of local I/O as general use inputs at once
EILN	Set port C (I/O-2) as negative over travel limit
EILP	Set port D (I/O-3) as positive over travel limit
EIRE	Set I/O 6 to capture external encoder's current value
EIRI	Set I/O 6 to capture internal encoder's current value
EISM (6)	Issue (G) when local input 6 goes low
EITR (*x***)**	Individual interrupt enable where x is interrupt # 0-7
EOBK (*#***)**	Configure a given output to control an external brake
IN (*#***)**	x=IN(#), assign the state of a specific I/O to a vari- able (x in this case)
IN (W,0)	x=IN(W,0), assign the state of the first word of local I/O to the variable x
INA (A,*#***)**	x=INA(A,#), raw analog reading: 10 bit resolution spanned over signed 16 bit range

INA (V,#)	x=INA(V,#), scaled voltage from supply of analog input value for a given I/O defined by #
INA (V1,#)	x=INA(V1,#), scaled 0-5 VDC reading in millivolts directly, 3456 would be 3.456 VDC
OC (#)	x=OC(#), individual output status, bit 1 if output is being driven
OC (W,#)	x=OC(W,#), block output status, bit 1 if output is being driven
OF (#)	x=OF(#), returns present fault state for I/O defined by #
OF (L,#)	x=OF(W,#), returns bit mask fault latched for I/O points
OF (W,#)	x=OF(W,#), returns bit mask of present faulted I/O points
OR (*value*)	Reset output (turn off)
OS (*value*)	Set output (turn on)
OUT(#)=*expression*	If argument to the right of "=" is true, then set I/O defined by # on(1), else off(0)

Math Commands:

-	Subtract
!|	Bitwise exclusive OR
!=	Not equal to
%	Modulo (remainder) division
&	Bitwise AND
*****	Multiply
/	Divide
^	Power limited to 4th power and below, integers only
|	Bitwise inclusive OR
+	Add
<	Less than
<=	Less than or equal to
==	Equal to
>	Greater than
>=	Greater than or equal to
ABS (*value*)	Absolute Value
ACOS (*value*)	Arc Cosine
ASIN (*value*)	Arc Sine
COS (*value*)	Cosine
FABS (*value*)	Floating point absolute value
FSQRT (*value*)	Floating point square root
RANDOM	Random number variable, 2^31-1 unsigned
RRANDOM	Report a random number, 2^31-1 unsigned
SIN (*value*)	Sine
SQRT (*value*)	Square Root
TAN (*value*)	Tangent
TMR (*x,t*)	Sets timer x for t milliseconds

Motion Commands:

ADT=*expression*	Set the accel/decel at once for a move
Ai (0)	Re-arms the index/trap register for rising edge for internal encoder
Ai (1)	Re-arms the index/trap register for rising edge for external encoder
Aij	
Aj (0)	Re-arms the index/trap register for rising edge for internal encoder
Aj (1)	Re-arms the index/trap register for rising edge for external encoder
Aji	
AMPS=*expression*	Current limit value. 0-1023
AT=*expression*	Set the acceleration target for a move
ATAN (*value*)	Arc Tangent
BREAK	Break out of while loop
BRKENG	Manually Engage the brake
BRKRLS	Manually Release the brake
BRKSRV	Brake Servo, engage the brake when the drive is not active (default)
BRKTRJ	Brake Trajectory
CTR (0)	Present value of internal encoder, same as PA command
CTR (1)	Present value of external encoder
DEL=*expression*	Set maximum allowable derivative error limit
DT=*expression*	Set the deceleration target for a move
EL=*expression*	Set maximum allowable following error limit
ENC1	Enable external encoder for servo
ENCO	Enable internal encoder for servo
F	Set tuning values
G	Go, initiates all buffered modes of operation
KA=*expression*	Feed forward gain
KD=*expression*	Derivative gain coefficient
KG=*expression*	Gravity offset
KI=*expression*	PID integral gain
KL=*expression*	PID integral limit
KP=*expression*	PID proportional gain
KS=*expression*	Differential sample rate
KV=*expression*	Velocity feed forward gain
MDB	Enable TOB when in one of the 2 trapezoidal modes
MDE	Set motor to enhanced trapezoidal mode commun tation by using encoder
MDS	Set motor to sine mode commutation
MDT	Set motor to trapezoidal mode communtation using hall sensors (default mode)
MFA (*value*)	Accel over *value* master distance. Default is zero (off)
MFD (*value*)	Decel over value master distance. Default is zero (off)
MFDIV=*expression*	Assign Incoming counts Divisor
MFMUL=*expression*	Assign Incoming counts Multiplier
MFO	Initiate and zero counter, but do not follow
MFR	Calculate Mode Follow Ratio and prepare to follow

MFR	Calculate Mode Follow Ratio and prepare to follow
MFSLEW (*value*)	Stay at slew for *value* distance, then decel
MFX	Slow down based on MFD
MINV (0)	Default, no invert, shaft rotates normally
MINV (0)	Default motor commutation direction
MINV (1)	Invert commutation, shaft rotates opposite direction
MINV (1)	Invert motor commutation from default
MP	Initiate Position Mode
MS0	Initiate and zero counter, but do not follow
MSR	Calculate Mode Step Ratio and prepare to follow
MT	Initiate Torque Mode (Open Loop)
MTB	Enable mode torque brake
MV	Initiate Velocity Mode
O=*expression*	Set origin, set present position to some value
OFF	Turn the amplifier off
OSH (*value*)	Origin shift of position counter on the fly
PID1	Set default PID update rate
PID2	Set default PID/2 update rate
PID4	Set default PID/4 update rate
PID8	Set default PID/8 update rate
PML=*expression*	Sets the position modulo limit wrap value
PMT=*expression*	Set position modulo target
POD=*expression*	Sets phase offset denominator
PON=*expression*	Sets phase offset numerator
PRT=*expression*	Set the relative target position
PT=*expression*	Set the absolute target position
S	Instantly stop motor
SLD	Disable software travel limits
SLE	Enable software travel limits
SLM (0)	Make a soft limit only trigger the flag, but not cause a fault
SLM (1)	Make s soft limit trigger the flag and cause a fault (default mode)
SLN=*expression*	Set negative software travel limit
SLP=*expression*	Set positive software travel limit
T=*expression*	Set commanded torque while in MT mode
TH=*expression*	Set maximum allowable thermal limit (degrees C)
THD=*expression*	Sets the number of cycle delays before soft over current foldback
VT=*expression*	Set the velocity target for a move
X	Decelerate to a stop at present deceleration rate

Status Commands:

Ba	*Over current bit, status word 0, bit 4* status word 1, bit 3
Be	*Excessive position error, status word 0, bit 6*
Bh	Excessive temperature occurred, status word 0, bit 5
Bi (0)	Rising Edge Capture on Encoder 0 (internal), status word 1, bit 2
Bi (1)	Rising Edge Capture on Encoder 1 (external), status word 1, bit 6
Bj (0)	Falling Edge Capture on Encoder 0 (internal),
Bj (1)	Falling Edge Capture on Encoder 1 (external), status word 1, bit 7
Bk	Main program checksum error, program is corrupt and cannot run, status word 2, bit 15
Bl	Left (-) over travel limit, status word 0, bit 13
Bls	Left (-) over travel software limit occurred, status word 1, bit 13
Bm	Left (-) over travel limit active, status word 0, bit 15
Bms	Left (-) over travel software limit active, status word 1, bit 15
Bo	Motor is off, status word 0, bit 1
Bp	Right (+) over travel limit active, status word 0, bit 14
Bps	Right (+) over travel software limit active, status word 1, bit 14
Br	Right (+) over travel limit, status word 0, bit 12
Brs	Right (+) over travel software limit occurred, status word 1, bit 12
Bs	Command Syntax error note, status word 2, bit 14
Bt	Trajectory in progress, status word 0, bit 2
Bv	Velocity limit, status word 0, bit 7
Bw	Wrap around occurred, position wrapped through +/- 2^31, status word 3, bit 3
Bx (0)	Hardware index input probe state for internal en coder, status word 1, bit 8
Bx (1)	Hardware index input probe state for external en coder, status word 1, bit 9
CLK=*expression*	Hardware clock variable
ERRC	Get most recent command error code
ERRW	Where/Who commanded most recent error
FSA (*fault type #,fault mode #*)	FSA(0,0) is default, sets all types of faults to result in MTB
RAA	Report actual accel/decel
RAC	Report commanded accel/decel
RAT	Report target acceleration
Ra	Report value of variable 'a'
Rab[0]	Report value of ab[0]
Raf[0]	Report floating point value of af[0]
Ral[0]	Report value of al[0]
Raw[0]	Report value of aw[0]
REPTR	Reports EEPROM pointer value
RCKS	Report Checksum
RB (*sw,b*)	Report status bit, b, from status word, sw
RCTR (0)	Report present value of internal encoder
RCTR (1)	Report present value of external encoder
RDEA	Report actual derivative error
RDEL	Report commanded derivative error limit
RDT	Report target deceleration
REA	Report actual following error
REL	Report commanded following error limit

RI (0)	Report where the rising edge of the internal index was detected
RI (1)	Report where the rising edge of the external index was detected
RIN (*#*)	Report the state of a I/O
RIN (*value*)	Report state of Input
RIN (W,0)	Report the first word of local I/O
RINA (A,*#*)	Reports analog input value for a given I/O defined by
RINA (V,*#*)	Reports voltage level (scaled from supply) of ana log input value for a given I/O defined by #
RINA(V1,*#*)	Reports voltage level (scaled 0-5 VDC) of analog input value for a given I/O defined by #
RJ (0)	Report where the falling edge of the internal index was detected
RJ(1)	Report where the falling edge of the external index was detected
RMFDIV	Report Divisor
RMFMUL	Report Multiplier
RMODE	Report mode of operation
RPA	Report present actual position
RPC	Report present commanded position (difference between present position and position error)
RPMA	Report the current modulo counter
RPML	Report position modulo limit
RPMT	Report the most recent setting of PMT (position modulo target)
RPOD	Report phase offset denominator
RPON	Report phase offset numerator
RPRA	Report actual relative position
RPRC	Report commanded relative position
RPRT	Report present relative target position
RPT	Report present target position
RRES	Report encoder resolution of motor
RSLN	Report value of negative software limit
RSLP	Report value of positive software limit
RSP	Report sampling rate and firmware version
RSP1	Report firmware revision date
RTH	Report maximum allowable thermal limit
RTMR (*x*)	Report timer x (present time left in milliseconds)
RT	Report commanded torque
RVC	Report commanded velocity
RVT	Report target velocity
RUIA	Reports current (Amps=UIA/100)
RUJA	Reports bus voltage (Volts=UJA/10)
RVA	Report actual velocity
RW (*value*)	Report status word
Z (*sw,b*)	Clears/zeros status word bits
Za	Reset over current bit
Ze	Reset position error bit
Zh	Reset over temperature bit
Zl	Reset left(-) historical limit bit
Zls	Reset left(-) software historical limit bit
Zr	Reset right(+) historical limit bit
Zrs	Reset right(+) software historical limit bit
Zs	Reset syntax error bit
ZS	Clear all errors, reset system latches to power up state
Zw	Reset wraparound bit

Variable Commands:

a=*expression*	Variable, 32 bit signed integers, a-z, aa-zz, aaa-zzz, 78 total variables
ab[*x*]=*expression*	Array variables, 8 bit byte arrays, x can be 0-200
af[*x*]=*expression*	Floating point array variables, x can be 0-7
al[*x*]=*expression*	Array variables, 32 bit long arrays, x can be 0-50
aw[*x*]=*expression*	Array variables, 16 bit word arrays, x can be 0-100
EPTR=*expression*	EEPROM pointer, non-volatile memory, use before VLD and VST commands
VLD (*variable,quantity*)	Load values from EEPROM to variables starting at EPTR location
VST(*variable,quantity*)	Store values to EEPROM from variables starting at EPTR location

Other Commands:

LOCKP	Physically disable program (EEPROM) upload
UPLOAD	Upload the program

OCHN (*RS2,0,*N,*9600,*1,8,*C,1000*) Default: (RS232,chan=0, no parity,9600 baud,1 stopbit, 8 databits, command,1000 ms timeout)

PRINT ("Hello World",#13) Print command to say "Hello World", see print section for more detailed examples

PRINT1 ("Hello World",#13) Print command to say "Hello World" on channel 1, see print section for more detailed example

ANIMATICS®

Animatics Corporation
3200 Patrick Henry Drive
Santa Clara, CA, 95054
Tel: +1 408.748.8721

www.animatics.com