

# Brushless IDEA™ Drive

Communication Manual

PBL4850E



**Haydon**  
Motion Solutions



[www.haydonkerk.com](http://www.haydonkerk.com)

All Rights Reserved

3-2015

## Table of Contents

Revision History .....	4
Introduction.....	5
IDEA Drive Communications Basics .....	6
Commands .....	7
<i>Abort</i> .....	8
<i>Check Password</i> .....	8
<i>Comment</i> .....	8
<i>Configure Encoder</i> .....	8
<i>Configure Input Interrupts</i> .....	9
<i>E-Stop</i> .....	9
<i>Execute Program</i> .....	10
<i>Go At Speed</i> .....	10
<i>Goto</i> .....	11
<i>Goto If</i> .....	11
<i>Goto Sub</i> .....	11
<i>Index</i> .....	12
<i>Interrupt on Position</i> .....	13
<i>Jump N Times</i> .....	13
<i>Label</i> .....	13
<i>Move To Position</i> .....	14
<i>No-op</i> .....	14
<i>Program</i> .....	15
<i>Read Control Gain</i> .....	15
<i>Read Current Position and Speed</i> .....	15
<i>Read Encoder Settings</i> .....	16
<i>Read Executing</i> .....	16
<i>Read Faults</i> .....	16
<i>Read Feedback Configuration</i> .....	17
<i>Read Firmware Version</i> .....	17
<i>Read IO</i> .....	17
<i>Read Max Current</i> .....	18
<i>Read Motor Parameters</i> .....	18
<i>Read Move Profile</i> .....	18
<i>Read Moving</i> .....	18
<i>Read Program Names</i> .....	19
<i>Read Startup Program</i> .....	19
<i>Recall Program</i> .....	19
<i>Remove Password</i> .....	19
<i>Remove Program</i> .....	20
<i>Restore Factory Defaults</i> .....	20
<i>Return</i> .....	20
<i>Return To</i> .....	20
<i>Run Program</i> .....	21
<i>Set Control Loop Gain</i> .....	21
<i>Set Feedback Configuration</i> .....	21

<i>Set Motor Parameters</i> .....	22
<i>Set Move Profile</i> .....	22
<i>Set Outputs</i> .....	22
<i>Set Password</i> .....	23
<i>Set Position As</i> .....	23
<i>Set Startup Program</i> .....	23
<i>Software Reset</i> .....	23
<i>Stop</i> .....	24
<i>Wait For Move</i> .....	24
<i>Wait Time</i> .....	25

## Revision History

<b>Date</b>	<b>Description</b>
October 2010	Initial release
January 2011	Added "Execute Program" command.
May 2011	Corrected response from Program command
September 2011	Added information about faults Added Read Moving command Updated configure encoder command Alphabetized commands
December 2011	Corrected configure encoder example
April 2013	Corrected program description Corrected table of contents
March 2015	Revised manual for brushless drive

## Introduction

The communication structure was initially developed for the IDEA stepper motor drive. With the introduction of the brushless drive to the IDEA lineup, it was our intention to maintain the communication structure for the brushless version. In this manner, our users which are familiar with programming the stepper drives have a smooth transition into the brushless environment. As the two driving methods vary greatly, the commands for the brushless drive may not use some of the parameters implemented in the stepper version.

## IDEA Drive Communications Basics

The IDEA drive line of products are commanded through the use of an Ascii based language developed by Haydon Kerk. Each command consists of a character identifying the command, followed by between 0 and 12 parameters separated by commas, and then followed by a carriage return. One difference between this language and those used by competing products is that each motion command encapsulates all parameters needed by the move; there are no parameters to set before a move command is issued. While this makes manual entry of commands into a terminal cumbersome, this is not the intended use of the language. Creation of these commands can be done simply in the software of the controller used to command the drives.

The IDEA drive adheres to a master/slave communications model. The master controller initiates all communications. If information is required from the drive, as in the case of requesting the drive's current position, the controller first sends the command requesting the drive's position, then the drive responds with the requested information, enclosed by several characters to identify the response. The extra characters can then be parsed, and the response read.

ONE MAJOR DIFFERENCE BETWEEN USING THIS COMMAND SET TO CONTROL THE DRIVE, AND USING THE IDEA DRIVE USER INTERFACE IS, THERE ARE NO PROTECTIONS WHEN USING THE COMMAND LANGUAGE. THE USER INTERFACE ENSURES THAT BASED UPON THE MOTOR PARAMETERS ENTERED, NO IMPROPER VALUES ARE SENT TO THE DRIVE; WITH THIS COMMAND SET, IT IS THE RESPONSIBILITY OF THE USER TO ENSURE THAT NO DAMAGE IS DONE TO THE DRIVE, MOTOR, OR OTHER EQUIPMENT THROUGH THE INCORRECT USE OF COMMANDS.

The parameters for serial communication are as follows:

Bits per Second: 57600

Data bits: 8

Parity: none

Stop Bits: 1

Flow Control: None

## Commands

The following describes the commands that make up the IDEA drive communications language, as well as the format for any response required from the drive. When quotation marks are present, the text in between the quotation marks is the important string, and the quotation marks themselves should not be included. When [cr] is shown, it is referring to the Ascii carriage return character, not to be confused with a line feed character. When [parameter] is shown, where parameter is the name of a parameter, it is representing some variable with that name, and the brackets will not be part of the string.

The contexts listed below indicate when each command can be used. Realtime commands can only be executed by direct command to the drive, such as requesting the current position. Program commands can only be a part of a program, and are generally branching or similar commands, such as Goto. Realtime/Program commands can be used anytime, and are generally motion related commands, such as Index. For further explanation of the commands, refer to the IDEA drive users' manual.

The following commands are used to configure the drive per the connected motor and will have an impact on drive performance. These parameters are saved to non-volatile drive memory and will only need to be set each time a new motor is used with the drive.

- Set Motor Parameters
- Set Feedback Configuration
- Set Control Loop Gain
- Set Move Profile Type

<u>Command</u>	<u>Symbol</u>	<u>Context</u>	<u>Arguments</u>	<u>Response</u>
<i>Abort</i>	A	Realtime/Program	none	None
<u>Description</u>	This command causes the drive to immediately stop, and ends the execution of any programs.			
<u>Arguments</u>	<u>Argument Description</u>			<u>Valid Values or Range</u>
none				
<u>Example</u>	You want to stop all drive activity.			
<u>Command</u>	"A" followed by a carriage return.			

<u>Command</u>	<u>Symbol</u>	<u>Context</u>	<u>Arguments</u>	<u>Response</u>
<i>Check Password</i>	c	Realtime	Password	"`cYES[cr]`c#[cr]" or "`cNO[cr]`c#[cr]"
<u>Description</u>	This command checks to see if a password is the correct password.			
<u>Arguments</u>	<u>Argument Description</u>			<u>Valid Values or Range</u>
Password	The password in question.			A string, exactly 10 characters long
<u>Example</u>	You want to check if the password is "password ".			
<u>Command</u>	"cpassword " followed by a carriage return.			

<u>Command</u>	<u>Symbol</u>	<u>Context</u>	<u>Arguments</u>	<u>Response</u>
<i>Comment</i>	C	Program	Comment	None
<u>Description</u>	This command creates a comment in the program.			
<u>Arguments</u>	<u>Argument Description</u>			<u>Valid Values or Range</u>
Comment	A string, must be exactly 10 characters long.			
<u>Example</u>	You want to add a comment that says "Extend 1in".			
<u>Command</u>	"CExtend 1in" followed by a carriage return.			

<u>Command</u>	<u>Symbol</u>	<u>Context</u>	<u>Arguments</u>	<u>Response</u>
<i>Configure Encoder</i>	z	Realtime/Program	0, 0, 0, 0, Encoder CPR, 1	None
<u>Description</u>	This command configures the encoder.			
<u>Arguments</u>	<u>Argument Description</u>			<u>Valid Values or Range</u>
N/A	Must be zero			0
N/A	Must be zero			0
N/A	Must be zero			0
N/A				0
Encoder Resolution	The resolution of the encoder being used in cycles per revolution			50 to 25000
N/A	Must be a value of 1			1
<u>Example</u>	You have an encoder with a cycles per revolution value of 1000			
<u>Command</u>	"z0,0,0,0,1000,1" followed by a carriage return.			



<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Configure Input Interrupts</i>	<b>i</b>	Program	Input1 config, input2 config, input3 config, input4 config, input1 destination, input2 destination, input3 destination, input4 destination, input1 priority, input2 priority, input3 priority, input4 priority	None
<b>Description</b>	This command is used to configure the interrupt settings for the inputs.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
Config	What kind of interrupt the input should be. 1 for Falling edge, 2 for rising edge, 3 for both edges, 0 for disabled.			0,1,2,3
Destination	The address of the subroutine that should handle the interrupt.			0 to 87036, multiples of four only.
Priority	The priority of the interrupt; lower numbered priorities are handled first.			0 to 4
<b>Example</b>	You want to set a rising edge interrupt on input 2, whose destination is address 512 and priority is 1, and all other input interrupts disabled.			
<b>Command</b>	"i0,2,0,0,0,512,0,0,4,1,4,4" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>E-Stop</i>	<b>E</b>	RealTime/Program	Decel Current, Hold Current, Delay Time	None
<b>Description</b>	This command stops the motor without decelerating.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
Decel Current	The maximum peak current, in milliamps, used to stop the motor.			0 to 6500
Hold Current	The maximum peak current, in milliamps, for after the motor has stopped.			0 to 5656
Delay Time	Time, in milliseconds, of the transition from the deceleration current to the hold current			10 - 300
<b>Example</b>	You wish to immediately stop the motor with a maximum decel current of 2.0 Apeak, then apply a maximum hold current of 1.0 Apeak to keep the load at position after 50 milliseconds.			
<b>Command</b>	"E2000,1000,50" followed by a carriage return			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Execute Program</i>	<b>m</b>	Realtime	Program name	None
<b>Description</b>	This command begins the execution of a program without changing the state of the outputs or motor.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
Program Name	The name of the program to run.			A string, exactly 10 characters long
<b>Example</b>	You want to run a program named "program 1 ", without returning to the default state.			
<b>Command</b>	"mprogram 1 " followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Go At Speed</i>	<b>Q</b>	RealTime/Program	Speed, Start Speed, End Speed, Accel, Decel, Run Current, Hold Current, Accel Current, Decel Current, Delay Time, 1	None
<b>Description</b>	This command accelerates the motor up to a specified speed, with the given parameters.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
Run Speed	The number of encoder counts per second the motor should move at the top speed			0 > to 16777215
Start Speed	Must be a value of zero			0
End Speed	Must be a value of zero			0
Accel Rate	Rate at which the speed should rise from Start Speed to the Run Speed.			0 > to 16777215
Decel Rate	Rate at which the speed should fall from the Run Speed to End Speed.			0 > to 16777215
Run Current	The maximum peak current, in milliamps for the move.			0 > to 5656
Hold Current	The maximum peak current, in milliamps, for after the move has completed.			0 > to 5656
Accel Current	The maximum peak current, in milliamps, for the acceleration portion of the move.			0 > to 6500
Decel Current	The maximum peak current, in milliamps, for the deceleration portion of the move.			0 > to 6500
Delay Time	Time, in milliseconds, of the transition from the deceleration current to the hold current			10 - 300
N/A	Must be a value of 1			1
<b>Example</b>	Desired move backwards at a speed of 3200 counts per second, accelerating at a rate of 40000 counts per second <sup>2</sup> , decelerating at a rate of 100000 counts per second <sup>2</sup> , with a maximum run current of 1.6 Apeak, a maximum accel current of 1.9 Apeak, a maximum decel current of 2.0 Apeak, and changing to a maximum hold current of 1.0 Apeak after a 50 millisecond delay.			
<b>Command</b>	"Q-3200,0,0,40000,100000,1600,1000,1900,2000,50,1" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Goto</i>	<b>G</b>	Program	Destination	None
<b>Description</b>	This command causes the program to continue execution at the specified address.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
Destination	The address of the command that should be run			0 to 86012, multiples of four only. Must be the address of a valid command.
<b>Example</b>	You want to continue execution at address 1024.			
<b>Command</b>	"G1024" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>												
<i>Goto If</i>	<b>L</b>	Program	Destination, Condition	None												
<b>Description</b>	This command causes the program to continue execution at the specified address if the condition is met.															
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>												
Destination	The address of the command that should be run.			0 to 86012, multiples of four only. Must be the address of a valid command.												
Condition	2 bytes indicating which I/O are tested, and the test values for each. The least significant byte corresponds to the inputs, and the most significant byte corresponds to the outputs. For each byte, the least significant nibble represents the condition being tested, a 1 meaning a high input or output, and a 0 representing a low input or output. The more significant nibble decides which of those conditions are to be tested, with a 1 representing an input or output should be tested. The least significant bit corresponds to input1, the next to input 2, and so on.			0 to 65535												
<b>Example</b>	You want to continue execution at address 1024 if input 2 is high.															
Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Total
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	34
<b>Command</b>	"L1024, 34" followed by a carriage return.															

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Goto Sub</i>	<b>S</b>	Program	Destination	None
<b>Description</b>	This command causes the program to execute the subroutine at the given destination.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
Destination	The address of the subroutine that should be run.			0 to 86012, multiples of four only. Must be the address of a valid command.
<b>Example</b>	You want to run a subroutine at address 1024.			
<b>Command</b>	"S1024" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Index</i>	I	RealTime/Program	Distance, Speed, Start Speed, End Speed, Accel, Decel, Run Current, Hold Current, Accel Current, Decel Current, Delay Time, 1	None
<b>Description</b>	This command moves the motor forward or backwards a defined number of counts, with the given parameters.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
Distance	The positive or negative number of counts the motor should move.			-18446744073709551616 to 18446744073709551615
Run Speed	The number of counts per second the motor should move at the top speed.			0 > to 16777215
Start Speed	Must be set to zero			0
End Speed	Must be set to zero			0
Accel Rate	Rate at which the speed should rise from Start Speed to the Run Speed.			0 > to 16777215
Decel Rate	Rate at which the speed should fall from the Run Speed to the End Speed.			0 > to 16777215
Run Current	The maximum peak current, in milliamps for the move.			0 > to 5656
Hold Current	The maximum peak current, in milliamps, for after the move has completed.			0 > to 5656
Accel Current	The maximum peak current, in milliamps, for the acceleration portion of the move.			0 > to 6500
Decel Current	The maximum peak current, in milliamps, for the deceleration portion of the move.			0 > to 6500
Delay Time	Time, in milliseconds, of the transition from the deceleration current to the hold current			50 - 300
N/A	Must be a value of 1			1
<b>Example</b>	Desired move is backwards 9600 counts per second, at a speed of 3200 counts per second, accelerating at a rate of 40000 counts per second <sup>2</sup> , decelerating at a rate of 100000 counts per second <sup>2</sup> , with a maximum run current of 1.6 Apeak, a maximum accel current of 1.9 Apeak, a maximum decel current of 2.0 Apeak, and a maximum hold current of 1.0 Apeak after a delay time of 50 milliseconds.			
<b>Command</b>	"I-9600,3200,0,0,40000,100000,1600,1000,1900,2000,50,1" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Interrupt on Position</i>	T	Program	Position, Destination, Priority	None
<b>Description</b>	This command sets an interrupt to occur at a given position.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
Position	The position where the interrupt should be triggered.			-18446744073709551616 to 18446744073709551615
Destination	The address of the subroutine to be run when the interrupt is triggered.			0 to 86012, multiples of four only. Must be the address of a valid command.
Priority	The priority of the interrupt; lower values are a higher priority.			0 to 4, 10 to disable
<b>Example</b>	You want to set a trip point at position 0, that runs a subroutine at address 1024, and has the highest priority.			
<b>Command</b>	"T0,1024,0" followed by a carriage return			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Jump N Times</i>	J	Program	Destination, Jumps	None
<b>Description</b>	This command causes the program to continue execution at the specified address a specified number of times.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
Destination	The address of the command that should be run.			0 to 86012, multiples of four only. Must be the address of a valid command.
Jumps	The number of times execution should branch to the destination address.			0 to 65535
<b>Example</b>	You want to continue execution at address 1024, and do so 3 times.			
<b>Command</b>	"J1024, 3" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Label</i>	B	Program	Label name	None
<b>Description</b>	This command creates a label in the program.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
Label Name	A string, must be exactly 10 characters long.			
<b>Example</b>	You want to add a label called "Start".			
<b>Command</b>	"BStart " followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Move To Position</i>	<b>M</b>	RealTime/Program	Position, Speed, Start Speed, End Speed, Accel, Decel, Run Current, Hold Current, Accel Current, Decel Current, Delay Time, 1	None
<b>Description</b>	This command moves the motor to a position, with the given parameters.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
Position	The positive or negative position, in counts, the motor should move to.			-18446744073709551616 to 18446744073709551615
Run Speed	The number of counts per second the motor should move at the top speed			0 > to 16777215
Start Speed	Must be a value of zero			0
End Speed	Must be a value of zero			0
Accel Rate	Rate at which the speed should rise from Start Speed to the Run Speed.			0 > to 16777215
Decel Rate	Rate at which the speed should fall from the Run Speed to End Speed.			0 > to 16777215
Run Current	The maximum peak current, in milliamps for the move.			0 > to 5656
Hold Current	The maximum peak current, in milliamps, for after the move has completed.			0 > to 5656
Accel Current	The maximum peak current, in milliamps, for the acceleration portion of the move.			0 > to 6500
Decel Current	The maximum peak current, in milliamps, for the deceleration portion of the move.			0 > to 6500
Delay Time	Time, in milliseconds, of the transition from the deceleration current to the hold current			50 - 300
N/A	Must be a value of 1			1
<b>Example</b>	Desired move is to position 0, at a speed of 3200 counts per second, accelerating at a rate of 40000 counts per second <sup>2</sup> , decelerating at a rate of 100000 counts per second <sup>2</sup> , with a maximum run current of 1.6 Apeak, a maximum accel current of 1.9 Apeak, a maximum decel current of 2.0 Apeak, and changing to a maximum hold current of 1.0 Apeak after a 50 millisecond delay.			
<b>Command</b>	"M0,3200,0,0,40000,100000,1600,1000,1900,2000,50,1" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>No-op</i>	<b>w</b>	Program	none	None
<b>Description</b>	This command is used to insert an extra line in a program.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
none				
<b>Example</b>	This command would be used in a custom user interface.			
<b>Command</b>	"w" followed by a carriage return.			

<i>Program</i>	P	Realtime	(Program Name, Start Location, Length) or none	None or "P[Program size][CR]P#[CR]"
<b>Description</b>	This command starts and ends the process of writing a program.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
Program Name	The name for the program, if it is the same as a program already on the drive, the old program will be removed.			A string; must be exactly 10 characters.
Start Location	The page number where the program should begin. If the program overlaps with any other program, the old program will be deleted. Each page has 1024 bytes of space.			1 to 75
Length	The number of pages the program will take up.			1 to 75
<b>Example</b>	You want to write a program name program 1, on the first page of memory with a program length of less than of 1 page.			
<b>Command</b>	"Pprogram 1 , 1,1" followed by a carriage return. Then followed by the commands that make up the program, each separated by a carriage return, followed by "P" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Read Control Gain</i>	)	Realtime	None	"")[Kps],[Kis],[Kds],[Kpp],[Ktff][cr]`)#[cr]" where [value] is a number.
<b>Description</b>	This command reads the gains of the control loop used to control the motor.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
None				
<b>Example</b>	You want to check settings of the various gains which make up the drives' control loop. Gains are in the following order: Velocity loop proportional gain, velocity loop integral gain, velocity loop derivative gain, positional loop proportional gain, and current loop feed forward gain.			
<b>Command</b>	")" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Read Current Position and Speed</i>	I	Realtime	None	"I[position],[position error],[velocity],[velocity error][cr]I#[cr]"
<b>Description</b>	This command requests the current position of the motor, the position tracking error, the current velocity of the motor, and the velocity tracking error			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
None				
<b>Example</b>	You want to check the position of the encoder, the position error, the current speed of the motor, and the error in speed			
<b>Command</b>	"I" followed by a carriage return.			

<i>Read Encoder Settings</i>	<b>b</b>	Realtime	None	"`b[NA],[NA],[Encoder CPR][cr]` `b#[cr]"
<b>Description</b>	This command requests the encoder configuration of the drive.			
<b>Arguments</b>	<b>Argument Description</b>		<b>Valid Values or Range</b>	
None				
<b>Example</b>	You want to check the encoder settings on the drive.			
<b>Command</b>	"b" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Read Executing</i>	<b>r</b>	Realtime	None	"`rYES[cr]` r#[cr]" or "rNO[cr]` r#[cr]"
<b>Description</b>	This command requests whether the drive is actively running a program.			
<b>Arguments</b>	<b>Argument Description</b>		<b>Valid Values or Range</b>	
None				
<b>Example</b>	You want to check if the drive is executing a program.			
<b>Command</b>	"r" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>				
<i>Read Faults</i>	<b>f</b>	Realtime	None	"`f[value][cr]` f#[cr]" where value represents the errors present. Each bit represents a specific error, as defined below.				
<b>Description</b>	This command requests the error status of the drive.							
<b>Arguments</b>	<b>Argument Description</b>		<b>Valid Values or Range</b>					
None								
<b>Example</b>	You want to check the error status of the drive.							
<b>Command</b>	"f" followed by a carriage return.							
Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Over Speed	Bad Checksum	Current Limit	Loop Overflow	Int Queue Full	Encoder Error	Temperature	Stack Overflow	Stack Underflow



<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Read Feedback Configuration</i>	>	Realtime	None	">[hall cell],[commutation],[encoder],[complement],[Rotor Position][cr]`>#[cr]" where [value] is a number.
<b>Description</b>	This command requests hall cell spacing, commutation direction, encoder direction, and complement setting of the drive.			
<b>Arguments</b>	<b>Argument Description</b>		<b>Valid Values or Range</b>	
None				
<b>Example</b>	You want to check the hall cell spacing, commutation direction, encoder direction, and complement setting of the drive.			
<b>Command</b>	">" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Read Firmware Version</i>	v	Realtime	None	"`v[value][cr]`v#[cr]" where [value] is a number.
<b>Description</b>	This command requests the firmware version of the drive.			
<b>Arguments</b>	<b>Argument Description</b>		<b>Valid Values or Range</b>	
None				
<b>Example</b>	You want to check the firmware version on the drive.			
<b>Command</b>	"v" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>				
<i>Read IO</i>	:	Realtime	none	"`:[value][CR]`#[CR]", Where [value] is a number between 0 and 255, formed from 1 byte, with ones being highs, zeros being lows, the most significant bit corresponding to output4, and the least significant bit corresponding to input1.				
<b>Description</b>	This command requests the status of the inputs and outputs.							
<b>Arguments</b>	<b>Argument Description</b>		<b>Valid Values or Range</b>					
none								
<b>Example</b>	Want to know the status of the input and outputs. For this example, outputs 1 and 2 will be high, and inputs 2, 3, and 4 will be high, all others will be low.							
<b>Command</b>	":#" followed by a carriage return.							
Output4	Output 3	Output 2	Output 1	Input 4	Input 3	Input 2	Input 1	Value
0	0	1	1	1	1	1	0	62

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Read Max Current</i>	j	Realtime	None	"j[value][cr]`j#[cr]" where [value] is a number.
<b>Description</b>	This command requests the maximum current setting of the drive.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
None				
<b>Example</b>	You want to check the maximum current of the drive.			
<b>Command</b>	"j" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Read Motor Parameters</i>	-	Realtime	None	"-[resistance],[inductance],[pole count],[voltage constant][cr]`-#[cr]" where [value] is a number.
<b>Description</b>	This command requests the motor characteristic settings currently implemented on the drive.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
None				
<b>Example</b>	You want to check the motors' resistance, inductance, magnetic pole count, and voltage constant settings currently implemented by the drive. Resistance, inductance, and voltage constant values will be multiplied by 1000.			
<b>Command</b>	"- " followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Read Move Profile</i>	_	Realtime	None	"_[value][cr]`_#[cr]" where [value] is a number.
<b>Description</b>	This command requests the move profile setting of the drive.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
None				
<b>Example</b>	You want to check the move profile setting, trapezoidal or s-curve, of the drive.			
<b>Command</b>	"_ " followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Read Moving</i>	o	Realtime	None	"oYES[cr]`o#[cr]" or "oNO[cr]`o#[cr]"
<b>Description</b>	This command requests whether the drive is moving.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
None				
<b>Example</b>	You want to check if the drive is moving.			
<b>Command</b>	"o" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Read Program Names</i>	N	Realtime	none	"N[program1 name],[start page],[end page][CR] N[program2 name],[start page],[end page][CR] N#[CR]" More programs would have more entries.
<b>Description</b>	This command requests that all program names and addresses be sent.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
none				
<b>Example</b>	You want to know what programs are residing on the drive.			
<b>Command</b>	"N" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Read Startup Program</i>	K	Realtime	none	"K[program name][CR] K#[CR]" If there is no startup program, [program name] will be an empty string.
<b>Description</b>	This command requests the name of the startup program.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
none				
<b>Example</b>	Want to know what program is set to run on power up.			
<b>Command</b>	"K" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Recall Program</i>	@	Realtime	Password, Program Name	The commands that make up the program, unless the password was incorrect, in which case there is no response.
<b>Description</b>	This command requests the program be read back.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
Password	The password for the drive			A string; must be exactly 10 characters.
Program Name	The name of the program to be read back.			A string; must be exactly 10 characters.
<b>Example</b>	Want to read back a program named "program 1" from the drive, with no password.			
<b>Command</b>	"@ ,program 1 " followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Remove Password</i>	q	Realtime	Password	None
<b>Description</b>	This command removes a password.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
Password	The current password			A string, exactly 10 characters long
<b>Example</b>	You want to remove the password "password " .			
<b>Command</b>	"qpassword " followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Remove Program</i>	D	Realtime	Program name	None
<b>Description</b>	This command removes a program.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
Program Name	The name of the program to be deleted.			A string, exactly 10 characters long
<b>Example</b>	You want to remove a program named "program 1 " from the drive.			
<b>Command</b>	"Dprogram 1 " followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Restore Factory Defaults</i>	a	Realtime	None	None
<b>Description</b>	This command removes the drive password and deletes all the programs on the drive.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
None				
<b>Example</b>	You want to remove the password on a drive, but forgot that password.			
<b>Command</b>	"a" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Return</i>	X	Program	none	None
<b>Description</b>	This command returns from a subroutine.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
none				
<b>Example</b>	You want to return from a subroutine to where the subroutine was called from.			
<b>Command</b>	"X" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Return To</i>	V	Program	Destination	None
<b>Description</b>	This command exits a subroutine, branches to a location, and clears all pending interrupts, the return stack and the loop counters.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
Destination	The address to which the program should branch.			0 to 87036, multiples of four only.
<b>Example</b>	You want to exit a subroutine and continue execution somewhere other than where the subroutine was called from, in this case, address 32.			
<b>Command</b>	"V32" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Run Program</i>	Y	Realtime	Program name	None
<b>Description</b>	This command begins the execution of a program, first returning to step 0 and setting all outputs low.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
Program Name	The name of the program to run.			A string, exactly 10 characters long
<b>Example</b>	You want to run a program named "program 1 ", starting from the default state.			
<b>Command</b>	"Yprogram 1 " followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Set Control Loop Gain</i>	(	Realtime	Set Control Gain	None
<b>Description</b>	This command sets the gains of the control loop used to control the motor, affecting motor response.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
Kps	Velocity loop proportional gain			0 to 10000
Kis	Velocity loop integral gain			0 to 10000
Kds	Velocity loop derivative gain			0 to 10000
Kpp	Positional loop proportional gain			0 to 10000
Ktff	Current loop feed forward gain			0 to 10000
<b>Example</b>	You want to set the default gain settings of the drive: Kps = 100, Kis = 100, Kds = 0, Kpp = 20, and Ktff = 0			
<b>Command</b>	"(100,100,0,20,0" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Set Feedback Configuration</i>	<	Realtime	Hall Cell Spacing, Commutation Direction, and Encoder Direction	None
<b>Description</b>	This command sets the hall cell spacing of the motor, sets the commutation direction, encoder direction, and allows users to invert the hall cell logic			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
Hall Cell Spacing	The hall cell spacing of the motor.			213087 for 60° spacing 215642 for 120° spacing
Commutation Direction	The commutation sequence direction to drive the motor.			0 for A→B→C 1 for B→A→C
Encoder Direction	The encoder signal direction			0 for CW (Ch. A leading Ch. B) 1 for CCW (Ch. B leading Ch. A)
Complement Hall Cell Logic	Whether to invert state of hall cells			0 to keep standard hall cell logic 1 to invert the hall cell logic
<b>Example</b>	You want to set the drive for 120° hall cell spacing, with standard commutation, a standard encoder direction, and standard hall cell logic			
<b>Command</b>	<215642,0,0,0" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Set Motor Parameters</i>	+	Realtime	Motors' Resistance, Inductance, Pole Count, and voltage constant	None
<b>Description</b>	This command configures the drive per a specific motor			
<b>Arguments</b>	<b>Argument Description</b>		<b>Valid Values or Range</b>	
Resistance	The resistance, per phase, of the motor (ohms multiplied by 1000)		200 to 100000	
Inductance	The inductance, per phase, of the motor (millihenries multiplied by 1000)		100 to 100000	
Pole Count	The magnetic pole count of the motor.		4,6,8, or 10	
Voltage Constant	The voltage constant of the motor (Volts/rad/sec multiplied by 1000)		1 to 10000	
<b>Example</b>	You want to set a resistance of 2.2 ohms, and inductance of 2.6 mH, a magnetic pole count of 4, and a voltage constant of 0.037 volts/rad/sec.			
<b>Command</b>	"+2200,2600,4,37." followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Set Move Profile</i>	=	Realtime	Password	None
<b>Description</b>	This command sets the move profile type for the drive to implement while performing moves.			
<b>Arguments</b>	<b>Argument Description</b>		<b>Valid Values or Range</b>	
Profile Type	Either trapezoidal or s-curve profile.		0 for Trapezoidal 1 for S-curve	
<b>Example</b>	You want to set the move profile type to trapezoidal.			
<b>Command</b>	"=0" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>				
<i>Set Outputs</i>	○	Realtime/Program	Output Value	None				
<b>Description</b>	This command sets the state of the outputs.							
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>				
Output Value	1 byte indicating which outputs should be set and what they should be set to. The most significant nibble indicates which outputs are being set, and the least significant nibble controls what they are being set to.			0 to 255				
<b>Example</b>	You want to set output 3 high, output 2 low, and want to leave outputs 1 and 4 unchanged.							
Bit 8 = 128	Bit 7 = 64	Bit 6 = 32	Bit 5 = 16	Bit 4 = 8	Bit 3 = 4	Bit 2 = 2	Bit 1 = 1	Total
0	1	1	0	0	1	0	0	100
<b>Command</b>	"O100" followed by a carriage return.							

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Set Password</i>	p	Realtime	Password	None
<b>Description</b>	This command sets a password, if none exists.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
Password	The desired password.			A string, exactly 10 characters long
<b>Example</b>	You want to set the password as "password ".			
<b>Command</b>	"ppassword " followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Set Position As</i>	Z	Realtime/Program	New Position	None
<b>Description</b>	This command adjusts the position counter.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
New Position	The position, in counts, you would like the current position to become.			-18446744073709551616 to 18446744073709551615
<b>Example</b>	After homing, you want to set the current location to 0.			
<b>Command</b>	"Z0" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Set Startup Program</i>	U	Realtime	Program name	None
<b>Description</b>	This command sets a program as the startup program.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
Program Name	The name of the program to start on power up or reset.			A string, exactly 10 characters long
<b>Example</b>	You want to set a program named "program 1 " as the startup program.			
<b>Command</b>	"Uprogram 1 " followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Software Reset</i>	R	Realtime/Program	none	None
<b>Description</b>	This command causes the drive to restart, acts the same as cycling power.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
none				
<b>Example</b>	You want to restart the drive.			
<b>Command</b>	"R" followed by a carriage return.			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Stop</i>	H	RealTime/Program	End Speed, Decel Rate, Run Current, Decel Current, Hold Current, Delay Time, 1	None
<b>Description</b>	This command stops the motor using an optional deceleration ramp.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
End Speed	Must be a value of zero			0
Decel Rate	Rate at which the speed should fall from the current speed to the 0.			0 > to 16777215
Run Current	The maximum peak current, in milliamps			0 > to 5656
Hold Current	The maximum peak current, in milliamps, for after the move has completed.			0 to 5656
Decel Current	The maximum peak current, in milliamps, for the deceleration portion of the move.			0 > to 6500
Delay Time	Time, in milliseconds, of the transition from the deceleration current to the hold current			50 to 300
N/A	Not Applicable			1
<b>Example</b>	You wish to stop the motor, decelerating at a rate of 100000 counts per second <sup>2</sup> , with a maximum run current of 1.6 Apeak, a maximum decel current of 2.0 Apeak, and changing to a maximum hold current of 1.0 Apeak after a delay of 50 milliseconds.			
<b>Command</b>	"H0,100000,1600,2000,1000,50,1" followed by a carriage return			

<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Wait For Move</i>	F	Program	none	None
<b>Description</b>	This command causes the program to delay execution of the next command until the motor has stopped moving.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
none				
<b>Example</b>	You have started a move command and do not want the next command to execute until the move has finished.			
<b>Command</b>	"F" followed by a carriage return.			



<b>Command</b>	<b>Symbol</b>	<b>Context</b>	<b>Arguments</b>	<b>Response</b>
<i>Wait Time</i>	<b>W</b>	Program	Time	None
<b>Description</b>	This command causes the program to delay execution of the next command for a specified time.			
<b>Arguments</b>	<b>Argument Description</b>			<b>Valid Values or Range</b>
Time	The amount of time, in milliseconds, that the command should be delayed.			0 to 65535
<b>Example</b>	You have started a move command and do not want the next command to execute for 1 second.			
<b>Command</b>	"W1000" followed by a carriage return.			