

MDC100-050101USB Programmable Brushless Motor Controller

User's Guide



ANAHEIM AUTOMATION, INC.

4985 East Landon Drive, Anaheim, CA 92807
sales@anaheimautomation.com

P: (714) 992-6990 F: (714) 992-0471
www.anaheimautomation.com

Table of Contents

Section 1: Introduction	2
Description	2
Methods of Communication	2
Baud Rate	3
Status LEDs	3
Electrical Specifications	3
Control Inputs/Outputs	4
Ordering Information	4
Dimensions/Switch Locations	5
Wiring Diagrams	5
Terminal Descriptions	6
Section 2: Functions	7
PI Tuning	8
Section 3: BMC100 Windows Software	9
Installation	9
Getting Started	9
BMC100 Settings and Functions	11
Section 4: Direct Talk Mode	12
COM Port Settings	12
Command Summary	13
Troubleshooting	22
Error Codes	22
Appendix 1: ASCII Table for Direct Mode	25
Copyright	25
Disclaimer	25
Warranty	26
Trademarks	26

Section 1: Introduction

The MDC100-050101USB is a Microcontroller based Programmable Brushless DC Motor Controller with Hall Sensor feedback for accurate speed measurement. It provides flexible, independent control of a Brushless DC motor from computers, or any machine controller with a USB. It is also capable of standalone operation, making it an embedded machine controller. The easy-to-use Windows software, BMC100, can be used to directly control motion of the motor for Real Time Movements through USB Communication or to set up the parameters for stand-alone use.

The MDC100-050101USB has 34 commands, which are easy-to-remember for direct control of a brushless motor. The MDC100-050101USB communicates via USB with PC or PLCs. The MDC100-050101USB communicates via an USB data bus. A windows driver is provided to communicate with the MDC100-050101USB through a virtual comport from the PC. This driver will turn any USB port into a virtual comport, thus enabling simple commands to be sent to the MDC100-050101USB . To use the BMC100 software, the virtual comport driver must be installed. The MDC100-050101USB uses only 2 watts at 24 VDC with no motor running.

Description

The MDC100-050101USB provides accurate control of Motor speed, direction, coast and brake for a Brushless DC Motor. MDC100-050101USB requires the user to set the number of poles in the motor for accurate measurement. It is embedded with a Proportional-Integrator (PI) controller program. Proportional and Integrator Constants, Kp and Ki, can be programmed using the BMC100 Windows Software or in Direct Mode. The controller has two modes through which it can control the speed of the motor: Analog Mode and Digital Mode. In Digital Mode, the controller is very accurate in keeping the motor running at the desired speed. The Analog Mode provides standalone functionality to the controller with a dynamic DC voltage input to control the speed of the motor. An analog voltage input can be used to set the desired speed between the upper and lower programmable limits.

Methods of Communication

There are two methods for sending commands to the MDC100-050101USB. One is to directly talk to the MDC100-050101USB using Direct Talk Mode (Connected to the Controller via a USB Type-B Connector). This is usually used with a computer or PLC (Programmable Logic Controller), where the computer or PLC gives the MDC100-050101USB serial commands. The second way to give commands to the MDC100-050101USB is to use the software program BMC100 for manual control. The BMC100 software should be used when the MDC100-050101USB is the main controller.

Baud Rate

A term used frequently in serial data communications, a “baud” is defined as the reciprocal of the shortest pulse duration in a data word signal, including start, stop, and parity bits. This is often taken to mean the same as “bits per second,” a term that expresses only the number of “data” bits per second. Very often, the parity bit is included as an information or data bit. The MDC100-050101USB accepts a baud rate of 38400 only.

Status LEDs

When powered and operated properly, the PWR/ERR LED will be green. When a communication error occurs, the LED will change to RED, and an error code will be generated in the error code register. To read and clear the error with the software, click on the “Read All” button. To read and clear the error while in “Direct Mode,” use the error code “!” command. Once the error has been read and cleared, the LED will return to green and the error code register will be cleared to 0. Refer to the table on page 22 for a complete list of the error codes. The RX LED will blink when data is being received from the PC. The TX LED will blink when data is being transmitted back to the PC. The BUSY LED will turn on whenever the motor is running and turn off when the motor is stopped. The FAULT LED turns on if the temperature of the controller becomes too high or if too much power is applied or if the current limit is reached. The FAULT LED also turns on when COAST function is applied. In case of current overload, break will be automatically applied to the controller and the motor will not restart until the current limit is met.



Electrical Specifications

Power Requirements:
20-50 VDC

Operating Temperature:
0 to 60 degrees C

Data Format:
Half-Duplex, 1 start bit, 8 data bits, no parity, 1 stop bit

Hall Sensor Power Outputs:
6.25V @ 30mA (Max)

Baud Rate:
38400 Baud, Fixed

Control Inputs/Outputs

Run/Stop: (TB1, Pin 1)

Logic “1” (open) - Motor will not run and if running will come to a stop rapidly.

Logic “0” = Motor will run.

Analog Input Voltage (TB1, Pin 2): 0 to 5VDC

BUSY Output: Open - Drain Type. $V_{ce}(\max) = 40\text{VDC}$, $I_c(\max) = 50\text{mA}$

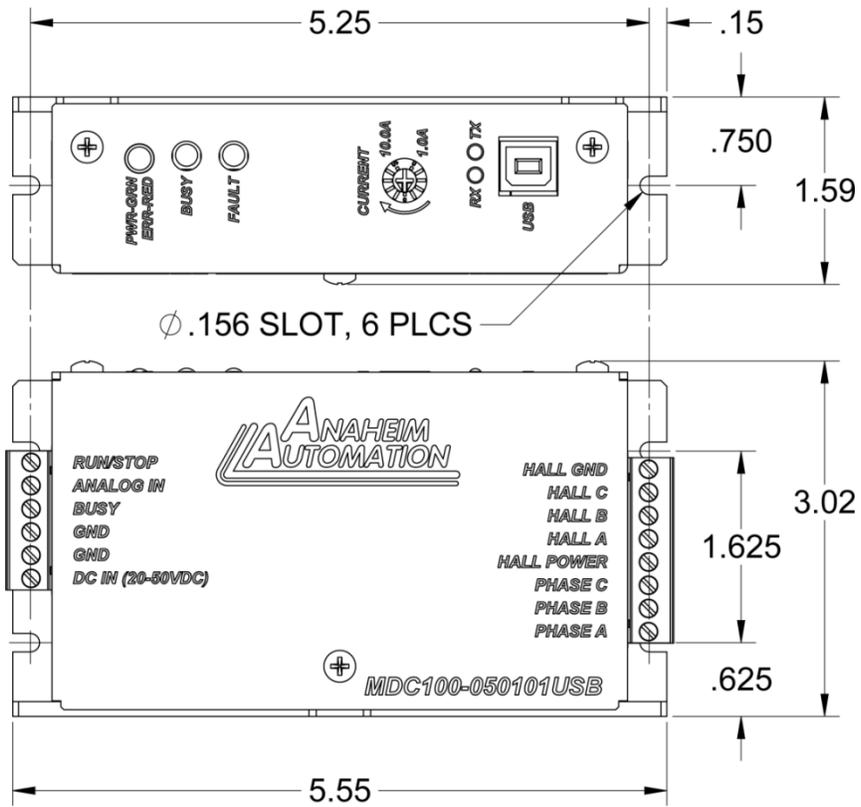
This output will conduct current when active, and be open when not active. Current needs to be limited into this pin. For the MDC100-050101USB Series, this output is a complete output.

Ordering Information

The table below lists a variety of products available from Anaheim Automation, Inc. These products include those covered by this manual, along with supporting cables and devices. We are continually adding new products to our line, so please consult Anaheim Automation, Inc. or its representatives for information on the latest releases.

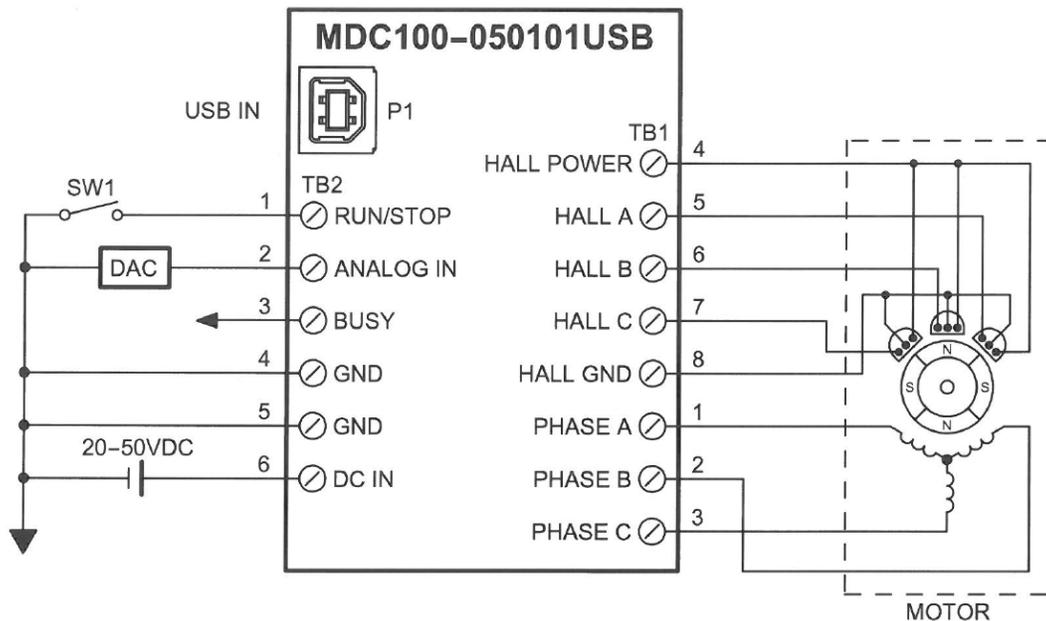
Part Number	Description
MDC100-50101USB	Featured brushless motor controller with hall sensor feedback.
AAUSB-AR-6	USB Communication cable that matches MDC100-050101USB
PSAM 24V 2.7A	DC Power Supply 24V at 2.7 Amps (60 Watts)
PSAM 24V 5.2A	DC Power Supply 24V at 5.2 Amps (125 Watts)
PSAM 24V 8.3A	DC Power Supply 24V at 8.3 Amps (200 Watts)

Dimensions/Inputs Locations



Note: All units are in inches.

Wiring Diagrams



Terminal Descriptions

Terminal Block 1.

Position	Description - Outputs to Motor
1	Phase A
2	Phase B
3	Phase C
4	HALL Power (5VDC)
5	HALL A
6	HALL B
7	HALL C
8	HALL GND

Terminal Block 2.

Position	Description - Inputs
1	Run/Stop
2	Analog Input Voltage
3	Busy
4	GND
5	GND
6	DC Power Supply (20-50VDC)

Section 2: Functions

Run At Given Set Point: This command causes the motor to start spinning at the set point speed. The set point value is in RPM. When the Set Point speed is too high for the motor or the controller to reach or if the load is too much for the motor to handle, the motor will run at the maximum speed possible. The motor will also run at the given set point with RUN/STOP input.

Note: If the motor is started Digitally, it has to be stopped digitally. And if it is started with the RUN/STOP input, it has to be stopped through the RUN/STOP input.

Hard Brake: This command causes the motor to come to an immediate stop. If the motor is moving fast, it will experience a jerk motion.

Coast: This command causes the motor to stop with a constant deceleration and therefore motor will not experience a jerk-like motion.

Go Clockwise: This command changes the direction of the motor to clockwise.

Go Counter-clockwise: This command changes the direction of the motor to counterclockwise.

Analog Input: One can start the motor in Analog mode via the BMC100 software of Analog Run/Stop Input. To set the speed Analog Voltage Input will read a voltage between 0 and +5VDC and based on the "upper and lower" limits of the function, a set point speed will be calculated for the motor to go to between the two limits. For example, if the lower limit is set to 0 RPM and the upper limit is set to 5000 RPM, and the analog input is set at +2.0VDC, then the motor will go to set point of 2000 RPM. Changing the lower limit to 1000 and the voltage to +3.2VDC, the motor will go to set point of 3560 RPM.

See examples below for calculations of the analog inputs.

$$(\text{Upper-Lower}) * (\text{Voltage}/5) = X$$

$$\text{Lower} + X = \text{Frequency}$$

$$\text{Speed(rpm)} = \text{Lower Limit} + ((\text{Upper Limit} - \text{Lower Limit}) * (\text{Analog Voltage}/5))$$

Example1: If Analog Voltage = 2V

$$\text{Speed} = 0 + (5000 - 0) * (2 / 5) = 2000 \text{ RPM}$$

Example2: If Analog Voltage = 3.2V

$$(5000 - 1000) * (3.2 / 5) = 2560$$

$$1000 + 2560 = 3560 \text{ RPM}$$

RUN/Stop Input: Grounding this input will cause the motor to turn on in either Mode and making it High will stop the motor using either Coast or Hard Brake depending on which command was used last. If Coast was used last, it will Coast.

Set Address: This command sets the address of the Controller.

Fetch Address: This command fetches the address of the Controller.

Read Error: This command reads the error. These errors are explained in the error codes section on page number 22.

PI Tuning

The MDC100-050101USB implements an optimized PI controller to control the speed of a brushless motor. The following chart will help you find the optimum settings for your system.

Parameter Increase	Rise Time	Overshoot	Settling Time(Ts)	Steady State Error (Ess)
Ki (0-150)	↑	↓	↑	↓
Kp (0-254)	↑	↑	↑	No Effect
IV (0-1400)	↑	↑	↑	No Effect

Explanation:

Ki is the integrator constant for the closed loop PI controller. This is the most important constant for bringing the running RPM close to the Set Point RPM.

Kp: If the difference between the running RPM and Set Point RPM is too great. the power applied to the motor is increased or decreased by the gain value before the PI controller takes over.

IV (Initial Value): Initial value is the initial power applied to the motor by the controller. If the load requires higher torque, a higher initial value is required to start spinning the load. If the load on the motor is very small and if the Initial Value is really high, the motor may experience a jerk motion.

Section 3: BMC100 Software

The BMC100 software is a handy utility that supports Anaheim Automation's MDC100-050101USB controller. The BMC100 uses a Serial Port and our proprietary Anaheim Automation protocol. This protocol is described in detail in the "Direct Talk Mode" Section. After connecting your PC to the MDC100-050101USB via a USB Cable, the BMC100 software can easily perform the following tasks:

- Exercise and monitor the MDC100-050101USB Controller
- Directly communicate with the MDC100-050101USB Controller

Installation

Software

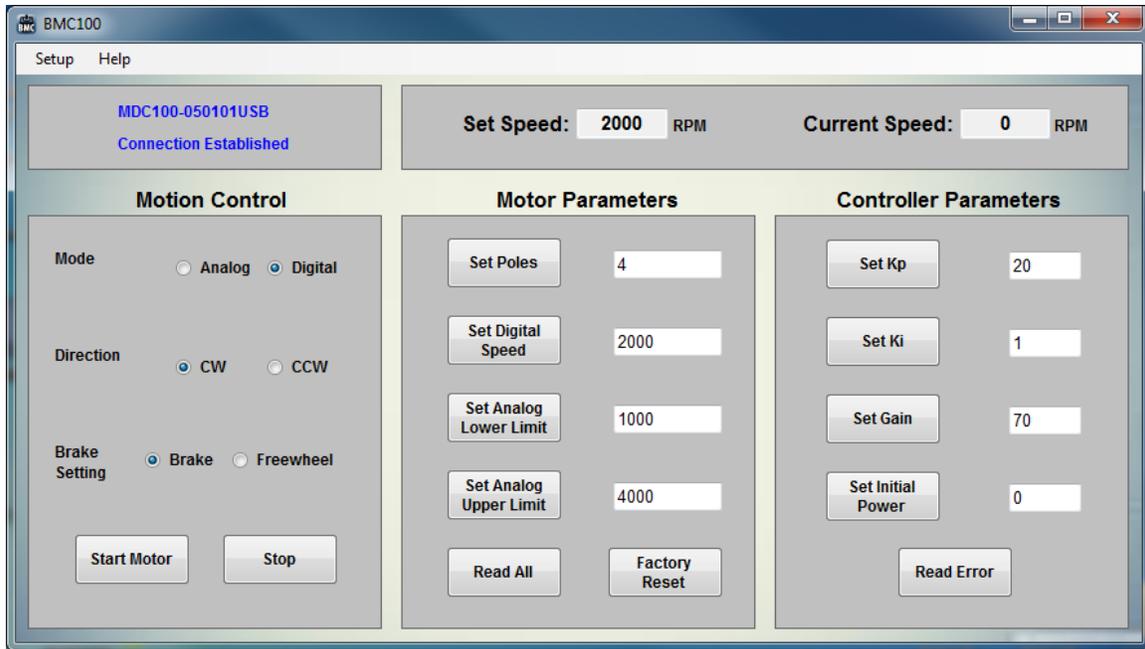
- The BMC100 software is available for download online on our website menu under Download Software. The link to this web page is provided below:
<http://www.anaheimautomation.com/downloads/software/software.php>
- The BMC100 is compatible with all versions of Windows including Windows 8, Windows 7, Windows Vista, and Windows XP.

Windows 8/7/XP/Vista Installation

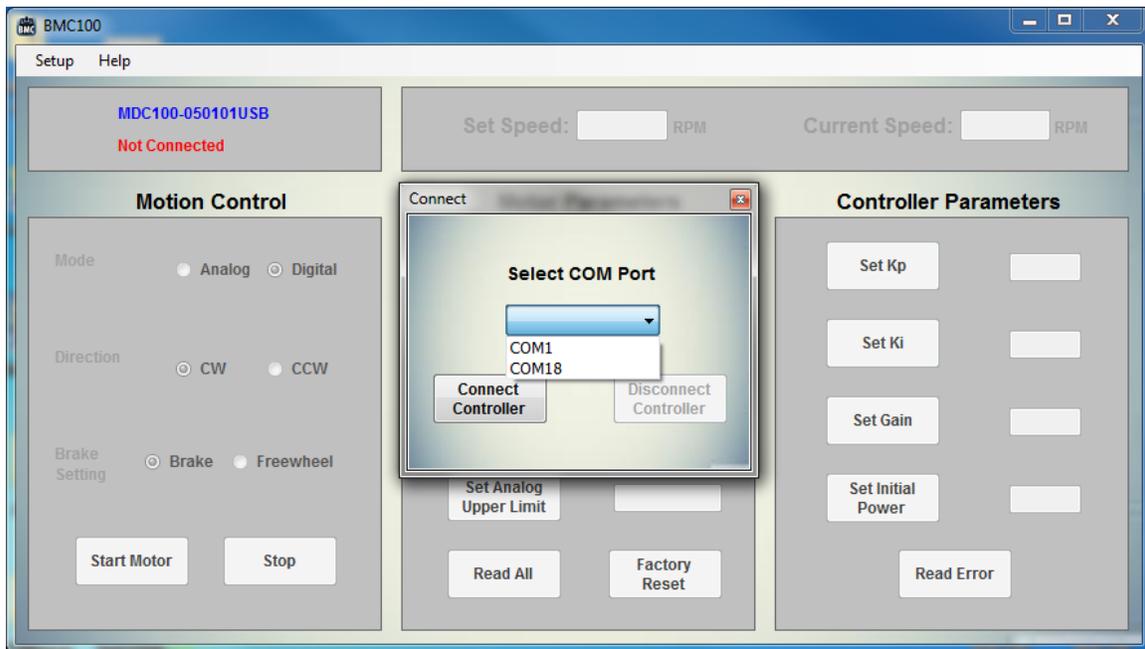
1. <http://www.anaheimautomation.com/downloads/software/software.php>
2. Scroll down to BMC100 Program and Download the .zip file by clicking on the download link. Choose a download destination folder and download the file.
3. Extract the BMC100 folder.
4. Enter the folder and double click setup.exe file and follow the prompts to install the software.

Getting Started

1. Connect MDC100-050101USB to PC via USB.
2. Apply power to the MDC100-050101USB
3. Double click on the BMC100 icon to run the BMC100 Software.
4. Select MDC100-050101USB from the drop down menu and Click Select.
5. The Software will attempt to establish communications with the controller; if a connection is established, the BMC100 GUI will appear as seen in the picture on the following page.



6. If the software did not automatically establish a connection with the controller, the user can manually connect to it after making sure that the controller is powered on and connected to the PC via USB. In order to do this, click on the **Setup** menu and then **Connect**. The connect window will pop up as shown below. Select the appropriate COM port from the dropdown menu and click on **Connect Controller**.



7. Once the connection is made, all the other buttons will be enabled. Also, the default (if the controller is new) or previously programmed parameters will fill the given fields.

Motion Control

Mode	Sets Analog or Digital Mode
Direction	Set Clockwise or Counter Clockwise Direction Looking at Motor Shaft
Freewheel	Let the Motor Coast to a Stop when a Stop Command is Received
Brake	Immediately Stop the Motor when a Stop Command is Received
Start Motor	Run The Motor at the Set Point Speed
Stop	Immediately stops or lets the motor coast to a stop depending on the brake setting

Motor Parameters

Set Poles	This Field Sets the Poles of the Motor
Set Digital Set Point	Sets the Set Point Running Speed in the Digital Mode
Set Analog Min. Limit	Sets the Lower Speed Limit for Analog Mode
Set Analog Max. Limit	Sets the Upper Speed Limit for Analog Mode
Read All	Update all the Parameters
Factory Reset	Set All the Parameters to the Factory Default

These functions are explained in more details in the **Functions** Section.

Controller Parameters

Set Kp	Sets the Proportional Constant
Set Ki	Sets the Integrator Constant
Set Gain	Sets the Gain Constant
Set IV	Sets the Initial Value Constant
Read Error	In Case of a Fault LED (Red) on the Board, Read Error

***Warning: Ignore the FAULT LED followed by a Coast Command as this is a built in hardware feature that can not be replaced.**

In order to fine-tune the controller for your application, look at the **PI Tuning** Section.

The Speed Parameters are constantly updated once the Motor is Started.

In order to disconnect the controller, just click on the "Disconnect Controller" button.

Warning: If the set point speed is too high for the motor to reach, the motor will run at the maximum speed possible. Two possible reasons this could happen are:

1. Higher than possible torque required for the specific load.
2. Given set point speed is more than the limit of the specific motor.

Section 4: Direct Talk Mode

Direct mode is used to directly control motion for real time movements through serial communication. The MDC100-050101USB controller has 34 commands, which are easy to remember for direct control of a Brushless DC motor.

COM Port Settings

Baud Rate: 38400
Data Bits: 8
Parity: None
Stop Bits: 1
Flow Control: Xon/Xoff

NOTE: The unit communicates in half duplex mode, therefore proper setup of hyper terminal is necessary to view characters, if characters are to be echoed back to the screen.

Unit Selection:

In order to communicate with a unit, the @ command followed by the address of the unit must be sent.

NOTE: There should be no spaces between the @ and address select. The factory default address of the unit is 0.

How to select a unit:

@0 (Unit 0 is selected)
@1 (Unit 1 is selected)
@29 (Unit 29 is selected)

Instructions

All instructions require that no spaces be sent between the command and the parameter followed by a carriage return. Carriage Return is ENTER on most Keyboards.

Example:

correct: @0(Command)(carriage return)
incorrect: @ 0 (Command) (carriage return)

Command Summary

Note:

* dash (_) after @ symbol represents the address of the unit. The range for the address is 0-99. All the commands are explained in detail at the end of the list. Other dashes(__) represent a parameter.

**The dash in the specific command (@# __) represents any of the 34 commands above.

More detailed description of this command is in the command explanation below.

@_S	@_V*
@_ ,	@_VM
@_ .	@_VS
@_ +	@_VN
@_ -	@_VX
@_A	@_VDS
@_D	@_VP
@_!	@_VD
@_P__	@_VMO
@_F	@_VST
@_M___	@_VKG
@_[____	@_VKP
@_]____	@_VKI
@_KI__	@_VIV
@_KG__	@_%
@_IV__	@~__
	@#__**

D - Digital Mode.

Format: @0D(Carriage Return)

Description: Engage the Digital Mode.

Example: If the unit is stopped and it is in analog mode, @0D will put it in digital mode. This command followed by @0S and then @0M2500 makes the motor run at 2500 RPM. If the unit is running in Analog Mode, this command doesn't have any effect.

A - Analog Mode.

Format: @0A(Carriage Return)

Description: Engages the Analog Mode.

Example: If the unit is stopped and in digital mode, @0A will put it in analog mode. This command followed by @0S or turning the Analog Brake Switch from OFF to ON makes the motor run at the speed calculated based on Analog Input Voltage and the respective analog min.

RPM and analog max. RPM. The input will read a voltage between 0 and +5VDC, and based on the “upper and lower” limits of the function, a max speed can be obtained based on a calculated frequency between the two points. See examples below for calculations of the analog inputs.

Analog calculations.

$$(\text{Upper-Lower}) * (\text{Voltage}/5) = X$$

$$\text{Lower} + X = \text{Frequency}$$

Example 1: If voltage is 2V and if Upper and Lower Limits are 5000 and 0 respectively.

$$(5000 - 0) * (2 / 5) = 2000$$

$$0 + 2000 = 2000 \text{ RPM}$$

Example 2: If voltage is 3.2V

$$(5000 - 1000) * (3.2 / 5) = 2560$$

$$1000 + 2560 = 3560 \text{ RPM}$$

+ - Clockwise Direction

Format: @0+(Carriage Return)

Description: Makes the motor spin clockwise.

Example: When this command is sent to the controller, the direction of the motor becomes clockwise whether the motor is running or not. If the motor is already running in clockwise direction, there is no change. If it is running in counter clockwise when this command is entered, the motor makes a quick stop and starts running clockwise at the set point speed it was already running at.

- - Counter - Clockwise Direction

Format: @0-(Carriage Return)

Description: Makes the motor spin counter clockwise.

Example: When this command is sent to the controller, the direction of the motor sets to counter clockwise whether the motor is running or not. If the motor is already running in counter clockwise direction, there is no change. If it is running in clockwise when this command is entered, the motor makes a quick stop and starts running counter clockwise at the set point speed it was already running at.

S - Start Motor

Format: @0S(Carriage Return)

Description: Starts the motor in Analog as well as in Digital mode.

Example: When this command is sent, the motor starts running at the set point speed based on other configurations. If the motor is started with @0S in analog mode, it can only be stopped with @0, or @0. and not the input switch. In other words, if the motor is started digitally, it can only be stopped digitally.

, - Coast

Format: @0, (Carriage Return)

Description: Coasts the motor to a stop if it is running in any mode.

Example: When this command is transmitted, if the motor is running, it comes to a stop without any jerk motion. If the motor is not running, no effect is seen with this command. This command also makes the Analog Input Brake a Coast. Therefore after this command is applied, if the motor is restarted and is running in analog mode, and if the analog brake is applied, the motor will coast to a stop.

• - Hard Brake

Format: @0. (Carriage Return)

Description: Engages Hard Brake if motor is running in any mode.

Example: When this command is transmitted, if the motor is running, the controller performs a hard brake. When this brake is applied, the motor comes to an immediate stop. If the motor is running at high RPM this may cause the motor to jerk and move a little. This command also makes the Analog Input Brake a Hard brake. Therefore after this command is applied, if the motor is restarted and is running in analog mode, and after that if the analog brake is applied, the motor performs a hard brake as described.

M - Digital Set Point Speed

Format: @0M(RPM Value)(Carriage Return)

Description: Changes the set point speed of the digital mode to the speed represented by the dashes. It has no effect in Analog Mode.

Example: @0M2600 will set the Digital Mode speed to 2600 revolutions per minute. If the motor is running in digital mode at any speed, the motor will run at 2600 rpm after this command is issued. If the motor is stopped but restarted in digital mode after this command, the set point speed will run at 2600 RPM and the motor will go to 2600 RPM. Factory default is 2000 RPM.

[- Analog Speed Minimum Limit

Format: @0[(RPM Value)(Carriage Return)

Description: Changes the lower boundary of the Analog Mode speed represented by the RPM Value.

Example: @0[600 changes the lower boundary of Analog Mode speed to 600 RPM. @0[1900 changes the lower boundary of Analog Mode speed to 1900 RPM. One condition for this is that the minimum speed has to be lower than or equal to the maximum speed. Therefore, if the maximum speed is assigned to 1800 RPM, @0[1900 will yield an error. If this error happens, no change is made to the minimum speed constant. This error is described in the Error Codes section. This boundary is used in calculating the set point speed for analog mode based on the formula described in the Calculating Analog Speed Section. Factory default is 1000 RPM.

] - Analog Speed Maximum Limit

Format: @0](RPM Value)(Carriage Return)

Description: Changes the upper boundary of the Analog Mode speed represented by the RPM value.

Example: @0]2000 changes the upper boundary of the analog speed to 2000 RPM. @0]900 changes the upper boundary of analog mode speed to 900 RPM. One condition for this is that the minimum speed has to be lower than or equal to the maximum speed. Therefore, if the minimum speed is assigned to 2000 RPM, @0]1900 will yield an error. If this error happens, no change is made to the maximum speed constant. This error is described in more details in the Error Codes section. This boundary is used in calculating the set point speed for analog mode based on the formula described in the Calculating Analog Speed Section. Factory default is 4000 RPM.

P - Set the Number of Poles

Format: @0P(Number of Poles of Motor)(Carriage Return)

Description: Checks and stores the number of poles as represented by the Number of Poles of motor.

Example: This information is crucial for correct RPM calculations. The default number of poles is 4. The options are 2,4,6,8,10,12. Therefore, @0P8 will set the poles parameter to 8. If the current poles parameter is set to 6, @0P5 or @0P99 will throw an error as described in the Error Codes section.

! - Error Code

Format: @0!(Carriage Return)

Description: Returns the error code as described in the Error Codes section.

Example: If the analog max speed is lower than the analog min speed, this command will return 128.

F - Factory Default

Format: @0F(Carriage Return)

Description: @0F restores all the parameters and variables to default. These values are described in the Defaults on Start Up section.

KG - Set Proportional Value

Format: @0KG(Proportional Value)(Carriage Return)

Description: Checks and stores the initial Proportional constant as represented by the Proportional Value.

Example: @0KG70 will make the Proportional constant 70. The range for KG is 0-254. @0KG255 or @0KG1000 will return errors as described in the Error Codes Section. The default value is optimized for a 4 pole Brushless motor with small loads on them. The higher the load, the higher this value should be.

KP - Set Integrator Constant

Format: @0KP(Integrator Value)(Carriage Return)

Description: Checks and stores the Integrator constant as represented by the dashes.

Example: @0KP20 will make the Integrator constant 20. The range for KP is 0-150. @0KP151 or @0KP1000 will return errors as described in the Error Codes Section. The default value is optimized for a 4 pole Brushless motor with small loads on them. The higher the load, the higher this value should be.

IV - Initial Value Constant

Format: @0IV(Proportional Constant Value)(Carriage Return)

Description: Checks and stores the Initial Value constant as represented by the Proportional Constant Value.

Example: @0KI2 will make the Integrator constant 2. The range for this is 0-1400 RPM. @0IV1500 or @0IV10000 will return errors as described in the Error Codes Section. The default value is optimized for a 4 pole Brushless motor with small loads on them. This value depends on the inertia of the load. As the inertia increases, a higher initial force is required to get the load spinning. For optimum results, this value should be increased as the load size or inertia is increased.

VM - Return the Current RPM of the Motor.

Format: @0VM(Carriage Return)

Description: Returns the actual speed the motor is running at.

Example: If motor is running at 1925 RPM, @0VM command will respond with 'M1925'. If motor is stopped, @0VM will respond with 'M0'.

VS - Return the Current Set Point RPM of the Motor.

Format: @0VS(Carriage Return)

Description: Returns the actual set point speed of the motor.

Example: It will return the calculated analog speed if it is in analog mode or the digital set speed if it is in digital mode. If the set point speed is 1630 RPM, @0VS will return 'S1630'

VN - Return the Lower Boundary of the Analog Mode Speed.

Format: @0VN(Carriage Return)

Description: Returns the lower boundary of the analog mode speed.

Example: If the lower analog mode boundary is set to 1000 RPM, @0VN will return 'N1000'

VX - Return the Upper Boundary of the Analog Mode Speed.

Format: @0VX(Carriage Return)

Description: Returns the upper boundary of the analog mode speed.

Example: If the upper analog mode boundary is set to 3000 RPM, @0VN will return 'X3000'

VDS - Return the Set Point Speed of the Digital Mode.

Format: @0VDS(Carriage Return)

Description: Returns the Set Point Speed of the Digital Mode.

Example: If the digital mode speed is set to 2500 RPM. @0VDS will return 'DS2500'. It will do this in analog mode too but in analog mode, this won't be the actual current set point of the point of the motor. In analog mode, it will still return the digital mode speed.)

VP - Return the User Set Number of Poles For the Motor.

Format: @0VP(Carriage Return)

Description: Returns the user set number of poles for the motor.

Example: @0VP(Carriage Return). If the poles parameter is set to 6, @0VP will return 'P6'

VD - Return the Current Direction of the Motor.

Format: @0VD(Carriage Return)

Description: Returns the current direction of the motor.

Example: If the direction is set to counter clockwise, @0VD will return 'D-' or if the direction is clockwise, it will return 'D+' '+' for clockwise, '-' would be for counter clockwise.

VMO - Returns the Mode.

Format: @0VMO(Carriage Return)

Description: Returns the Mode of the controller.

Example: Factory default is A for analog mode. If the mode is set to Analog, @0VMO will return 'MOA.' If the mode is set to Digital, @0VMO will return 'MOD'

VST - Return Configuration of Stop Bit.

Format: @0VST(Carriage Return)

Description: Returns the current configuration for stop bit. Factory default is '.' for a hard stop.

Example: If the stop bit parameter is set to coast, @0VST will return 'ST,' and if the stop bit parameter is set to hard brake, @0VST will return 'ST.'.

VKG - Return the Gain Constant

Format: @0VKG(Carriage Return)

Description: @0VKG returns the current Gain Constant for the motor. Factory default is 70. The range for KG is 0-254.

Example: If the Gain Value parameter is set to 50, @0VKG will return 'KG50'.

VKP - Return the Proportional Constant

Format: @0VKP(Carriage Return)

Description: Returns the current proportional constant for the motor. Factory default is 20. The range for Kp is 0-150.

Example: If the KP parameter is set to 20, @0VKG will return 'KP20.'

VKI - Return the Integrator Constant

Format: @0VKI(Carriage Return)

Description: Returns the current Integrator constant for the motor. Factory default is 1. The range for KI is 1-9.

Example: If the KI parameter is set to 1, @0VKG will return 'KI1.'

VIV - Return the Initial Value Constant

Format: @0VIV(Carriage Return)

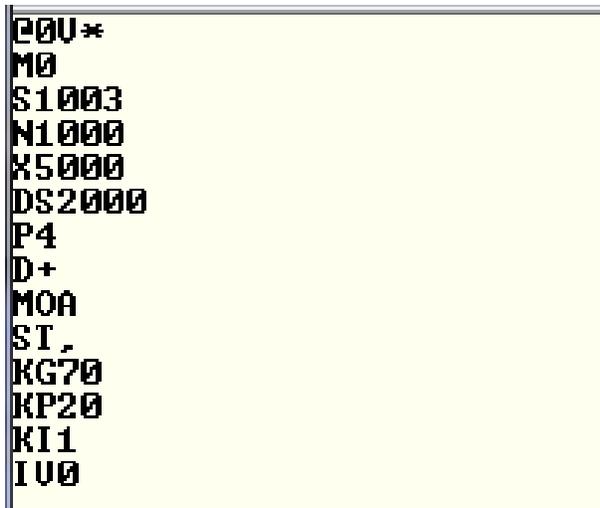
Description: Returns the current Initial constant for the power applied to the motor. The factory default is 0. The range for this is 0-1400. The higher the inertia of the load, the higher this value should be.

Example: If the IV parameter is set to 150 , @0VIV will return 'IV150.'

V* - Returns All the Values Mentioned Above.

Format: @0V*(Carriage Return)

Description: An example of this can be seen in the picture below:



```
@0V*
M0
S1003
N1000
X5000
DS2000
P4
D+
MOA
ST.
KG70
KP20
KI1
IU0
```

% - Fetches the Address of the Controller.

Format: @%(Carriage Return)

Description: Fetches the address of the controller.

Example: If the address is 0, @%(Carriage Return) will return '0'. If the address is 29, @% will return '29'.

~ - Sets the Address of the Unit.

Format: @~[Address Value](Carriage Return)

Description: Sets the address followed by "~." Range for the address is 0-99.

Example: @~10 will set the address to 10. @~100 will give an error as mentioned in the error codes section.

- Match any Address

Format: @#[Command](Carriage Return)

Description: @# matches any address and executes the command that follows "#." Therefore, any command followed by @# is executed no matter what the profile is.

Example: @#S will start the motor. @#P6 will set the number of poles to 6.

Troubleshooting

Problem:

Cannot establish communications with the MDC100-050101USB.

Possible Solutions:

- 1) Make sure the MDC100-050101USB controller has power. Is the Green LED on?
- 2) Check the USB connections.
- 3) Check for loose cable connections on the MDC100-050101USB
- 4) Was the software installed successfully?
- 5) Go to Setup | Com Port Settings and verify COM port and baudrate settings.
- 6) If problems still exist, contact Anaheim Automation Tech Support.

Anaheim Automation, Inc. Tech Support:	4985 E. Landon Drive Anaheim, CA, 92807 phone: (714) 992-6990 fax: (714) 992-0471 www.anaheimautomation.com
---	--

Problem:

There is no power to the MDC100-050101USB controller.

Possible Solutions:

- 1) Is the MDC100-050101USB controller connected to the appropriate power supply?
- 2) Check for any blown fuses in line with the MDC100-050101USB controller.
- 3) If problems still exist, contact Anaheim Automation, Inc. Tech Support.

Error Codes

1	BCMND	ERR,0	;Invalid command
2	BRANGE	ERR,1	;Invalid # of characters for the command
4	BTIME	ERR,2	;While motor is running, certain instructions can't be performed
8	BPOLES	ERR,3	;Poles have to be even numbers in 2-12 range
16	BPRANGE	ERR,4	;Bad Integral range. Range is 0-150
64	BGRANGE	ERR,6	;Bad Proportional range. Range is 10-254
128	BRPMRANGE	ERR,7	;RPM min> RPM max and RPM min<100
256	BIVRANGE	ERR2,0	;Initial Value range is 0-1400.

Examples of each error codes:

If the Address is not correct for the controller, it will not display any error as it will not be just ignoring the command. If any of the following errors occur, no command is executed.

All the following examples assume that the address of the controller is 0. These examples trigger the given ERROR CODE.

1. Invalid Command.

@0G or @0k @0# @03

2. Invalid # of characters for the specified command. Too many characters in the command.

@0S01, @0M234234, @0P123 @~100.

4. While motor is running, following instructions can't be performed. They are all checked for.

These will be triggered only if the motor is running.

@0P__ @0A @0D @0F and @~__ . If any of these also had too many characters, it will actually return error code 2.

8. Poles have to be even numbers in 2-12 range.

@0P1 @0P99 @0P14 @0P

16. Bad Integrator Constant (Kp) range. Range is 0-150

@0KP151 @0KP999.

Note: @0KP1000 gives ERROR code 2.

64. Bad Proportional range. Range is 10-254

@0KG9 @0KG255 @0KG999.

Note: @0KG1000 gives ERROR code 2.

128. Analog RPM min > Analog RPM max will give ERROR code 128.

If Analog max is set at 4000

@0[4001 will give this error.

Note: @0[10000 gives ERROR code 2

If Analog min is set at 1000,

@0]999 will give this error.

Note: @0]10000 gives ERROR code 2

256. Bad initial value Range. Initial Value range is 0-1400.

@0IV1401 @0IV9999.

Note: @0IV10000 will give ERROR code 2.

Appendix 1: ASCII Table for Direct Mode

ASCII Symbol	Hex Value	ASCII Symbol	Hex Value	ASCII Symbol	Hex Value
0	30	J	4A	#	23
1	31	K	4B	\$	24
2	32	L	4C	%	25
3	33	M	4D	"	27
4	34	N	4E	(28
5	35	O	4F	+	2B
6	36	P	50	,	2C
7	37	Q	51	-	2D
8	38	R	52	.	2E
9	39	S	53	:	3A
A	41	T	54	;	3B
B	42	U	55	@	40
C	43	V	56	[5B
D	44	W	57]	5D
E	45	X	58	^	5E
F	46	Y	59	{	7B
G	47	Z	5A	}	7D
H	48	Carriage Return	0D	~	7E
I	49	!	21		

Copyright

Copyright 2014 by Anaheim Automation, Inc.. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of Anaheim Automation, Inc., 4985 E. Landon Drive, Anaheim, CA, 92807. The only exception to this would be use of the program examples in this manual.

Disclaimer

Though every effort has been made to supply complete and accurate information in this manual, the contents are subject to change without notice or obligation to inform the buyer. In no event will Anaheim Automation, Inc. be liable for direct, indirect, special, incidental, or consequential damages arising out of the use or inability to use the product or documentation.

Limited Warranty

All Anaheim Automation, Inc. products are warranted against defects in workmanship, materials and construction, when used under Normal Operating Conditions and when used in accordance with specifications. This warranty shall be in effect for a period of twelve months from the date of purchase or eighteen months from the date of manufacture, whichever comes first. Warranty provisions may be voided if the products are subjected to physical damage or abuse.

Anaheim Automation, Inc. will repair or replace at its option, any of its products which have been found to be defective and are within the warranty period, provided that the item is shipped freight prepaid, with RMA (return material authorization), to Anaheim Automation's plant in Anaheim, California.

Trademarks

Control Link and Driver Pack are registered trademarks of Anaheim Automation, Inc. IBM PC is a registered trademark of International Business Machines, Inc.