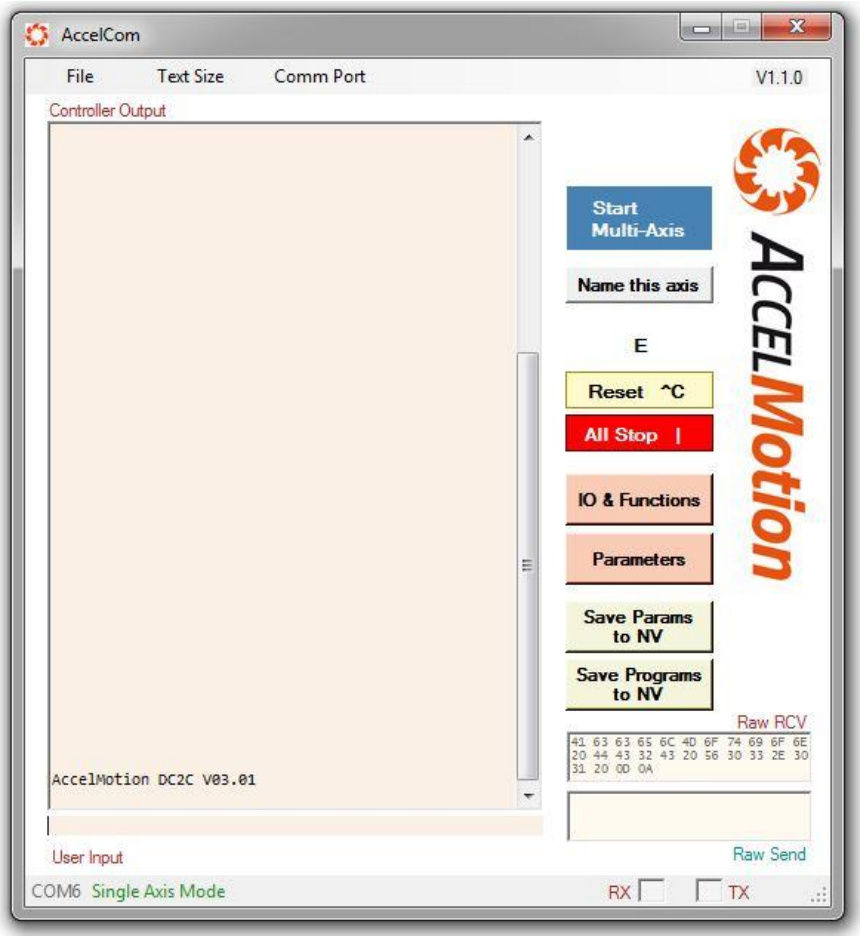
AccelCom

Terminal Program



USER GUIDE

AccelMotion

TABLE OF CONTENTS

AccelCom Features

The **AccelCom** terminal program allows easy communication to all AccelMotion controller/driver units from a standard Windows PC (supports WindowsXP, WindowsVista, WIndows7, WIndows8 and WIndows10). It is designed specifically to support the DC Series of driver/controllers.

AccelCom *features are helpful in the debug and programming of DC-series products, but it is not required – it can be substituted by simple terminal programs such as TeraTerm, RealTerm or even Putty.*

AccelCom provides a console window allowing direct communication with driver/controller units on an AM Communications Bus, as well as helper buttons which select **Single-Axis** and **Multi-Axis** modes, and adjustment screens for controller operating parameters as well as a screen for Input/Output function settings and current I/O values.

The **AccelCom** terminal provides**:**

- Standard terminal (console) functions, with window showing user data typed in and responses back from the DC series units
  - User input window provides character-by-character feedback for each command entered (Note: DC-series controllers utilize line-by-line echoing)
  - All functions of DC-series controllers can be easily evaluated, and programs can be debugged and finalized from the terminal window only, if desired
  - Configurable text size, for a wide range of screen sizes
- Built-in functionality optimized for the DC series of controller/driver units, including:
  - Automatic communications setup including baud-rate and comms settings
  - Simple selection of COM port (either standard or virtual port support)
  - Optimized I/O and Parameter support for the DC Series
    - Visual feedback of current parameters for a selected axis controller
    - Visual feedback of current I/O functions and data for an axis controller
- Status Bar provides real-time status of operating modes, COM port, and RX and TX data
- Convenient function buttons, allow:
  - Entry into **Single-Axis mode** or **Multi-Axis Mode**
  - **Renaming** the current controller/driver (when in Single-Axis Mode)
  - Quick **reset** function (^C), resets the controller completely
  - Quick b function (|), stops all motion and all programs immediately
  - **Input/Output function** setup, as well as reading and writing **I/O** signal values
  - Modification of a units operating **parameters**, in visual format
- Raw ASCII send and receive buffer screens show data for evaluation or debug

Overview

Quick Start

The AccelCom terminal is a Windows application and is provided in standard Windows installer format. Download it from our website `accelmotion.com`, and install it; depending upon your configuration you may need to install other Microsoft Windows resources for AccelCom to operate.

1) Install AccelCom
2) Identify the serial port to be used on the host PC – both hardware serial ports and virtual ports (such as those used on PCI-to-serial and ethernet-to-serial converters) can be used
   a) Hardware Port (ex: COM1, COM2): DC-series controllers require a RS485 converter between a PC RS232 serial port and the AM Bus RS485 serial bus – such converters are available in the electronics marketplace, or consult with technical support at AccelMotion for more information
   b) Virtual serial port: we recommend a reliable USB-to-RS485 converter such as the *AccelMotion CI-200*
3) If a serial converter is used then plug it into the PC and load any required drivers
4) Connect the converter output into the AM Comms Bus two-wire cable
   a) be sure to connect a ground wire between the converter signal ground and the DC-series controller power supply ground (see DC-series unit user manual for more information)
5) Plug the DC-series units needed for your application into the shared AM Comms bus – we recommend plugging in *one at a time* to make sure all are individually connected correctly to the AM comms bus, and that each controller is given a different multi-axis name
6) Start AccelCom (or another terminal, such as TeraTerm)
7) Power up the DC-series controller
8) Press the orange **Start Single-Axis** button to connect to a single controller – in **Single-Axis** mode the controller will respond with its name and version: `AccelMotion DC2C V2.2`

9) Standard commands can now be typed on the User Input line, for transmission to the controller. For example, one of the first tasks for a new unit is to provide a 'name' for the unit. The name acts as an address for commands and responses whenever multiple units are controlled on the same AM Comms Bus. See naming and programming chapters in your DC-series product User Guide.

Example Commands

a) Name the unit: either type a **Control-N** (^N or hex 0E), or press the "**Name this axis**" button; a dialog box will open where the user can type a letter A-Z or a-z, selecting up to 52 units

b) Examine the parameters: type **X**, to see a text output of the controller and driver operating parameters such as velocity, acceleration, driver current, temperature, etc. for this unit – this can also be achieved by pressing the **Parameters** button

c) Set I/O functions: Utilize the **U** command or press the **I/O & Functions** button to change input ports into fixed function inputs, such as Home, Go, Stop, Limits, and Jog inputs – once changed, these functions are direct and require no additional programming, and if stored in NV memory their function becomes permanent.

d) Read general-purpose inputs: Utilize the A or N command to read inputs, or press the **I/O & Functions** button to read all input values (including fixed function inputs)

e) Modify general-purpose outputs: Utilize the w or command to write to outputs, or press the **I/O & Functions** button to write to any output ports available.

f) add program lines using the **P** command, then run the program with the **G** command – for more information on writing and entering programs, examining program memory, and saving programs to non-volatile memory, see the User Guide for your DC-series controller(s)

Before review of this guide AccelMotion strongly recommends review of the DC-series User Guide for your chosen driver/controller model, and in particular:

Single- and Multi-axis Modes
Host-based and internal Programs
I/O types and Fixed-function Input Ports
Non-volatile memory and Program Organization,
Muti-axis Command Addressing (names, and naming)

These concepts are outside the scope of this guide

Reference

AccelCom™ is a terminal program with additional features for AccelMotion DC-series products, and is made available without charge from AccelMotion. AccelMotion recommends its use, but users can easily operate DC-series and other AccelMotion controllers with various scripting languages (such as LabView™, VB, Python, Perl, etc.), and debug with simple terminals such as TeraTerm, RealTerm, etc.

AccelCom can provide a beneficial experimentation and debugging environment for the development of programs and motion routines, both for host-driven control as well as autonomous control. It is not intended as a general-purpose control system for controlling one or more AccelMotion driver/controllers – for this purpose scripting tools such as LabView are recommended.

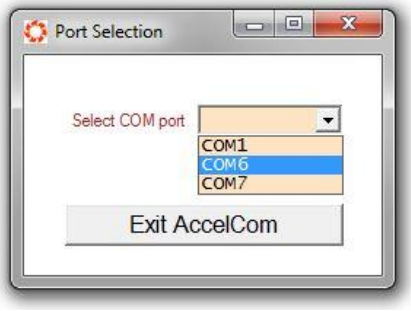
Installation

AccelCom is installed from a standard Microsoft™ installer window, and is tested for operation under Windows Vista™, Windows7™, Windows8™, and Windows10™. The installer may request that additional Windows components be installed to support the application.

Starting AccelCom

AccelCom provides a simple terminal program which is used to communicate to one or more DC-series driver/controller units, document programs and parameters, and provide a limited back-up process.

AccelCom is designed to automatically set the proper communications format and data rate for DC-series controllers and so does not require any comms configuration by the user. *It is required* to select a COM port which is connected to the AM Comms Bus populated by the DC-series controller(s), and this is the first dialog box which appears when AccelCom is opened.



Select the Port which is currently connected to the AM Bus (through a RS-485 converter). This port can be a fixed hardware port (using a RS232 to RS485 converter) or a USB or Ethernet interface which uses 'virtual com port' installed by their driver. If your Virtual COM port does not appear, make sure that proper drivers have been installed.

Once a port is selected, the main communication screen appears and no other comms setup is required. The main screen allows the COM port to be changed via the **Comm Port Menu**, if needed.

Also on the main screen is a **Text Size Menu**, allowing the selection of different text size on the terminal window to better match the available screen size and readability.

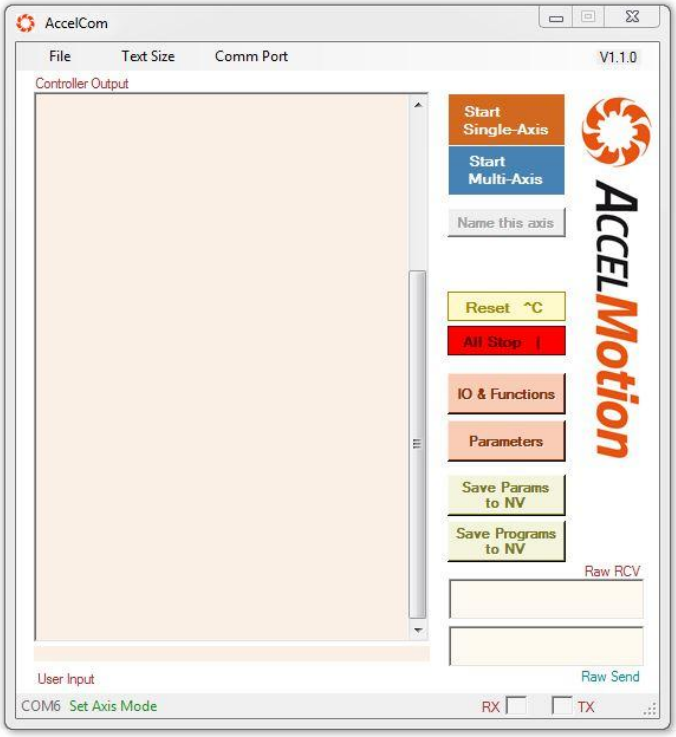
AM Bus Communications Modes and the Idle View

The main screen provides most of the controls needed by a user for controlling DC units. Two AM Bus modes of communication are available for DC units: Single-axis mode and Multi-axis mode.

Single-axis mode is used to directly address a single DC-series unit without need for addressing (inherently addressing that single unit). This mode allows easy configuration and operation since no command addressing is required. *This mode is required to* ***name*** *each DC unit* – if any DC-unit will ever be required to operate alongside other units on the same AM bus then it will require a unique name (address); a single small or capital letter which identifies the commands intended for each unit.

Multi-axis mode is used to control any number of units by adding the name prefix to each command – an example would be a move positive command, identified by the command character '+', which in single-axis mode would simply be "`+10000`" for a 10,000 step move, but in multi-axis mode the command would be "`Y+10000`" when targeted at the unit named "Y". Responses to each command are also prefixed by the unit responding. See more on this in the DC-series User Guides.

Idle mode is the starting condition for the terminal, while the AM bus communications mode is unknown. The view (right) shows that in idle mode most buttons are disabled and that the two main modes can be selected with individual buttons – Orange for Single-Axis and Blue for Multi-Axis.

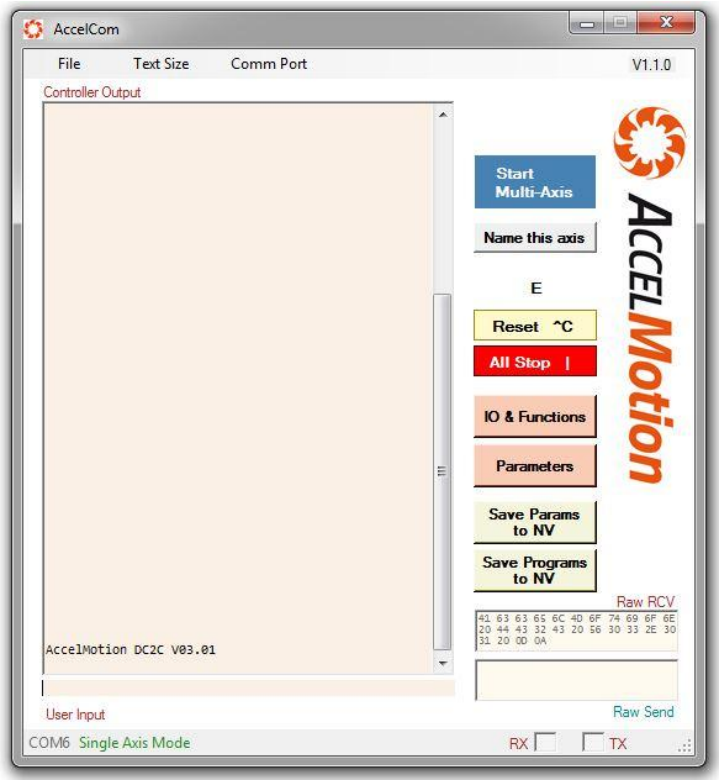


Some common features to all modes are:

- Menus (top), allowing:
  - downloading of parameters and programs, and editing of downloadable files
  - adjustment of text size, allowing selection of different text size to increase readability
  - changing comm port, to allow multiple AM bus operation
- Status bar (bottom) shows selected COM port, operation mode labels and tips, and RX and TX status
- Raw receive and transmit data boxes (lower right) show user commands in raw ASCII format
- The user input field, on the lower left
- The Output 'console' field on the left above the under input field
- A set of control buttons to implement commands graphically instead of using command line
  - **Reset** implements ^C reset, stopping all motion and programs and entering idle mode
  - **All Stop** implements **|** (pipe) command, stopping individual unit motion and program
  - **IO & Functions** opens a new window which allow the updating of I/O outputs, modification of input fixed-functions, and displays input values as long as it is open (these changes graphically implement the **X**, **U**, **A** and **w** commands); it provides a graphic view of I/O functions and data.
  - **Parameters** button opens a new window for the graphic display and modification of the most common motion and control parameters – it utilizes data derived from command queries (see DC series command structure in user guide)

Most main screen features are permanently located and are simply enabled or disabled as different modes are entered and exited.

Single-Axis Mode

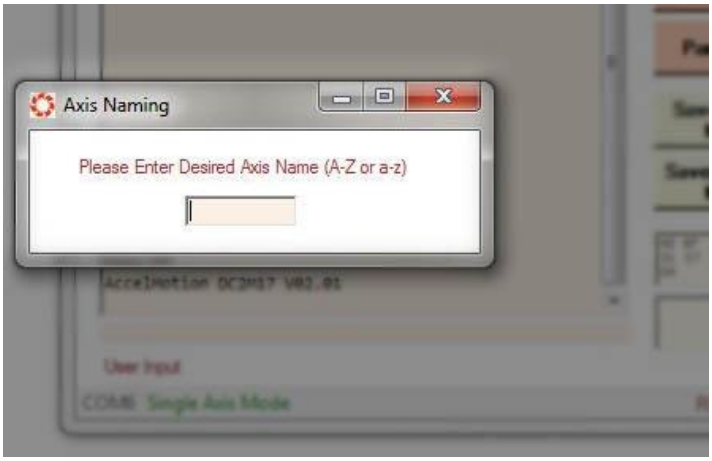


When working with a single DC unit, the **Single-axis mode** is the easiest way to communicate and configure a controller, because the multi-axis address (name) prefix is not needed. AccelMotion recommends that experimentation and checkout of motion control routines and autonomous programs be done in this mode.

A second purpose for **single-axis mode** is to name each unit – see the DC-unit User Guide to describe this name requirement and how it is implemented.

One difference in this mode is the provided identification of the attached unit (there should only be one unit on the AM bus in this mode) – note the ID line at the bottom of the screen (right) showing a DC2C 2-amp unit. The software also queries the attached unit for its current **Name** and displays it just above the yellow Reset button.

Note that in this mode buttons are available and active, the Single-axis button is disabled, and the status line shows the current mode.



A button specific to the Single-Axis mode is now visible called "**Name this axis**". This button provides an automated method to rename the attached unit – use any upper- or lower-case letter as a name (up to 52 units). Note also that the current axis name is labeled just above the reset button.

Some commands provide a more verbose response (or are valid) only in **Single-axis mode** – an example is the **X** command, providing a textual list of current parameters with a single command. Parameter settings can also be queried by sending each parameter command *without a new value*; this is interpreted as a request for the current parameter setting. An example: "`V10000`" would normally set the V parameter (slew velocity) to 10000 steps-per-second, but simply sending "`V`" to a unit with the current V value of 3500 SPS causes a response of "`V3500`". These commands can be entered from the command line (User Input) and responses show in the Output window.

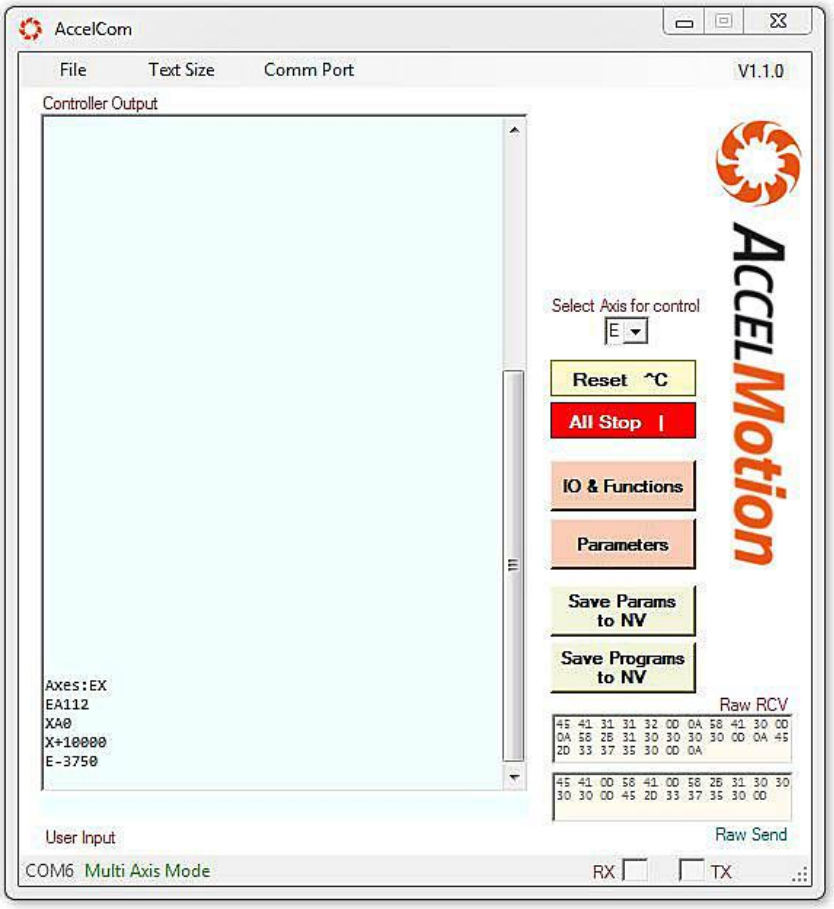
Finally, the Save buttons: The **Save Params** saves the current Parameter settings (plus I/O functions) into permanent non-volatile (NV) memory, similar to the S0 command; and the Save Programs button saves the current main program memory into NV memory, similarly to the S1 command.

Multi-Axis Mode

When one or more units must be controlled from the same host communication interface, Multi-axis mode is required. DC-units (and AccelCom software) start in an idle mode, and can be commanded to single-axis mode or multi-axis mode from idle, but also multi-axis mode is reachable easily from single-axis mode with a single global command ^P (control P). See DC-series User Guides for more info. In this mode responses to the host are only sent when paired with a host command.

Since unit naming is only valid in Single-axis mode, the naming button is unavailable in Multi-axis mode. The axis name label is replaced by a 'pull-down' field containing the names of all units on the AM Bus, allowing the **All-Stop**, **IO & Functions** and **Parameters** buttons to be targeted to a specific named unit. Note on the figure below that two units are connected, names **E** and **X**, and the pull-down selector can target either of those names for access to I/O and Parameter data (see these functions later in guide).

Note: the text output "`Axes: EX`" is provided by the AccelCom software directly – it's a "unit scan" operation, where each and every possible unit name is requested to provide a response (the AccelCom software uses a **c** command, responding with an ID message) to signal it is connected. This scan can be implemented by a host script, but is unneeded in most cases (unless you expect unknown unit counts).



Once the unit scan is complete, and the Axis List is populated with the captured unit names, the user can then either type multi-axis commands on the command line (such as EV4000, or Ew5), or open one of the **I/O** or **Parameter** dialog boxes to modify parameters or change I/O features or values.

Changing the current control axis pull-down will change the context of the **I/O** and **Parameter** dialog boxes (more on this later), affecting only the specific axis/unit selected.

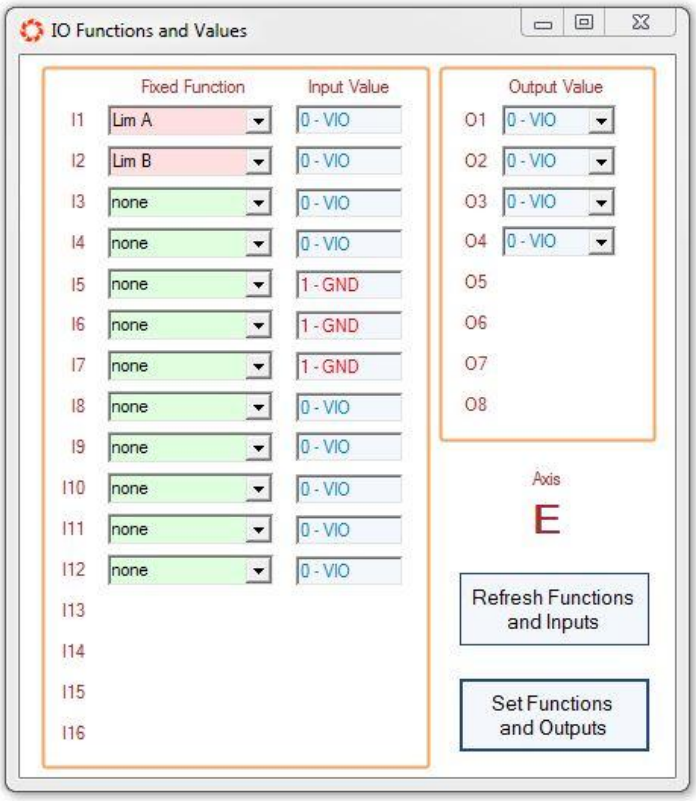
The selected control axis also affects the target of the **All-Stop** button (but remember the **RESET^C** button is global so affects all units).

Save Params and **Save Programs** buttons each affect only the unit selected on the current axis pull-down, similarly to a typed-in command such as "`ES0`" or "`XS1`".

Note that the bottom status bar shows "Multi-Axis Mode".

IO & Functions

The **IO & Functions** button opens a new IO & Functions dialog box that allows the user to examine and change a selected axis input port function settings, output port values, and to examine all input port values – see the User Guide for the specific unit to better understand these features and their operation.



Inputs

Each input port in a DC-series unit can be used as a general-purpose input value for conditional branching instructions (Command **L**), but each can also be configured to provide a hard control function to the controller, such as GO, STOP, HOME, JOG+, JOG–, or Limit+ and Limit–. The I/O fixed function section, on the far left side, allows the user to assign a fixed function to any input port, and to then inspect the functions. Once a fixed function is assigned and set, the input port is labeled by function and the label background is red to mark the assignment.

To modify a fixed function: **Pull-down the fixed function field on the specific inputs**, and select the fixed functions required. Once all functions are assigned, press the **Set Functions** button to write the updated functions to the unit – this works in Single- or Multi-Axis mode. The set button will also refresh the function fields to show the final settings.

Note: it is illegal to set the same fixed function to more than one input port at one time, and for some DC controllers the HOME function must only be assigned to port 1 (see specific User Guides).

For manually updated functions, press the **Refresh Functions and Inputs** button to display the latest.

To read the inputs: Open the **IO Functions and Values** screen and inspect the center column of values, which are the current input values of all ports (even if they have a specific function assigned and in use). Press the **Refresh** button in the lower right to update all the contents of the screen as needed.

Outputs

On the left side of the **IO Functions and Values** screen is the Outputs Value section, where the current status of each output port is shown. Press the **Refresh** button to update the outputs status.

To change the Output Port values, pull down the selector for each output port to change, and select the different value desired. When all values are selected, press the **Set Functions and Outputs** button to modify the outputs.

To lock the Functions and Outputs, be sure to save the Parameters with the **Save Parameters** button. IO Functions and Outputs are considered part of the Parameters and are saved along with the Parameters.

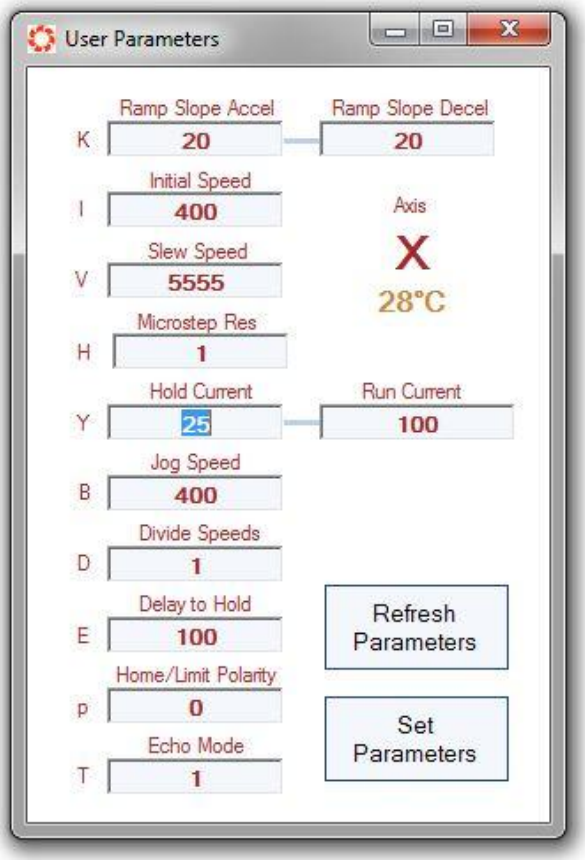
Parameters

The operating parameters of a DC controller/driver include all motion settings, communications settings, input port fixed functions, and the current output port values. These can be modified and set up via text commands, but the AccelCom terminal provides a graphical interface for inspecting and modifying these values in both single- and multi-axis modes.

To open the **Parameters** Dialog Box, press the **Parameters** button on the main screen. The window shows the available parameters for modification and review, starting with motion and driver values on the top, moving to lesser used parameters toward the bottom.

I/O fixed functions normally shown on the text Parameter command **X** are handled by the **IO and Functions** screen, shown elsewhere.

The Parameters window also shows the name of the selected controller unit (axis name) and the current temperature of the driver (where applicable).



To modify a parameter: click on the field to modify it and change it to the desired value. Press the **Set Parameters** button to write the values to the Unit.

Any **Parameter** field value exceeding the minimum or maximum allowed values will be substituted with either the min, max, or a default value, depending upon the specific unit and the parameter.

To update the latest Parameter values: press the **Refresh Parameters** button, to update all parameter values shown for the selected axis.

To view Parameters for another unit: simply change the **Control Axis field** on the main screen, and choose another unit in the pull-down – the **Parameter** and **IO and Functions** screens will update automatically with the new information for that unit.

Menu Items – Settings and Upload/Download

The File menu contains three functions that support the programming and data maintenance of DC controllers. These are:

- **Edit...** to create downloadable command files or programs
- **Download** allows file of text commands to be communicated to selected DC-series system unit
- **Memory Image Upload** moves the entire memory contents of the selected unit into a host file

Edit...

Edit menu item opens a basic text editor, which allows the user to create and edit command files, or review and modify memory image files. The editor can create new files, open existing files, and save changes. These files are used to document command strings or Programs, in the .SMC format.

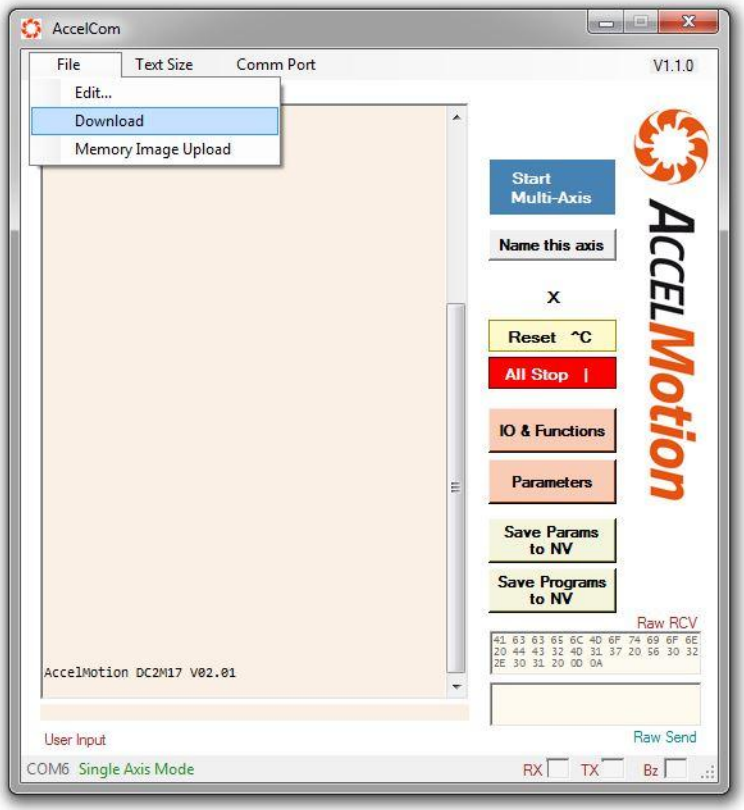
See **File Formats** section below for more information on file types and usage.

Download

This menu item opens any text or .SMC file and transmits it to the selected unit – in single-axis mode the file is transmitted with no modifications, and in multi-axis mode the targeted address is prefixed onto each line before transmission. The multi-axis prefix is the **selected control axis** from main screen.

For this reason, all program files created on AccelCom or other text editors should be created without address name prefixes – they will be downloaded correctly to the selected unit in any mode.

Note also that uploaded **Memory Image** files are stored in .SMC format as well and can also be downloaded to units.



Memory Image Upload

This menu item requests the selected axis unit to reply with the current contents of its main memory (not NV), which is stripped of any address prefix (in the case of multi-axis mode) and then stored in a special upload-ready form. All DC-series units are memory-compatible and so can utilize the same programs and memory-based settings from unit to unit.

Notes and Tips – File and Data Formats – Features for Programming

When attached to one or more DC-series units, the AccelCom editor can be used to create programs and program segments, as well evaluate motion parameters (such as speed, power, acceleration, etc.). Below are notes and tips on the features of DC units, as well as how AccelCom can be used to evaluate and develop motion control programs and settings.

Commands and Programs – Immediate and Program Mode

Commands can provided to a DC-series controller by a host in real-time in **Immediate Mode**, or commands can be placed in the units program memory and executed in **Program Mode**. **Immediate Mode** commands are run at the time they are received. Scripts (programs) running on the host can query the controller unit for conditions such as input values, system status, motor location, motion in-progress, etc. and these conditions are returned as responses for use in conditional command ordering.

For **Program Mode**, the commands and programming reside inside the unit, so the command set is larger and the format is expanded. Added instructions allow conditional execution or "branching" where different code can be run depending upon real-time conditions in the controller. To facilitate branching an addressing strategy is used: each program line commands + and command parameters are stacked into memory in order of execution. Note that command instructions vary in the number and size of command parameters, so to make best use of limited memory each instruction takes up only the number of instruction spaces needed (see command reference for these sizes). Finally, because each command instruction is variable in size, the address for each program segment start or conditional-branch target is specific to the location and size of earlier commands. For convenience DC-series units provide these addresses as commands are entered in **Program Entry Mode**.

Command Modes – what is your requirement?

If your system already uses a host, or consists of multiple axes which must be controlled together, then it's recommended to use **Immediate Mode** commands in **Multi-axis** communications; programming in the host. If so, concepts such as internal memory, addressing, conditional branching, etc. don't really matter, as you won't use these features. Even in this case, testing program segments in internal memory as a method to debug host-based scripts is a possibility, and could be useful.

But, if your system needs only information from each axis to control that axis, and has no over-arching control requirement or existing host, then it is recommended to consider writing internal programs for each unit/axis and store those programs in the unit's non-volatile (NV) memory.

Issues with programming for "host-less" internal programs

Programming best practices include the documentation and protection of your system scripts and settings. In a host-based architecture it is recommended to design and store all programs in the host system, utilizing standard computer storage, backup and documentation systems. But, In the internal program case, instead it is recommended to take the following steps:

1) design the general program flow for the system, with motion commands and events which: start and stop, change positions, actuate mechanical elements, and sense external inputs
2) connect the hardware to the DC-series controller, and test each function (motors, limits, actuators, sensors, etc.) with AccelCom
3) determine subsections of the program, and test each one individually where possible – sometimes it's much easier to enter the codes quickly into the **P** program entry mode to test an individual solution.

4) Now, begin to implement the formal program on the host in a text editor using the results from your empirical testing (the AccelCom editor can be used), implementing the code segments already tested and connecting them together in the memory -- It is recommended to:

   a) segregate phases of operation into sections of code, jumping to each with a **G** Go command

   b) start each discrete section at a new address, leaving a little space between the new and old

   c) start loops at the new known 'round' address, so that the loop target address is easily determined without counting command sizes – if this isn't easy to do, then simply download the program into memory as see what address is allocated for the loop start, and then go back and annotate your text file with the correct address. NOTE: if you change the code, by adding, changing, or removing commands then your addresses could change and must be checked.

5) By writing the code on the host and uploading it to the unit each time for testing, your program will be stored and protected on the host storage and documentation system. If the program is modified, more systems are created, or if a controller must be replaced, this master version of the program will be available. Note: Application programs are important intellectual property so must be stored and protected!

Unified Memory Space and Memory Images

DC-series devices were created with unit-to-unit compatibility as well as memory-mapped parameters and settings – this combination allows a simple memory image to hold both parameters and programs together, and thus sharing images from unit to unit is not only possible but useful.

The main memory address space of DC-series controllers is broken into two address regions: 0-199 and 256-1023 are program space, and 200-254 are memory-mapped parameter/settings space. A file created from the **Image Upload** menu item contains the value of each memory location, so it stores the *whole context of the unit* – this is an excellent way to simply 'back-up' a unit, or to even transfer exact settings and programs between compatible units.

> To back up a unit: First make sure that all current main memory contains the final version of the memory (or NV memory), by either a ^C reset or a C0 command; next click File, Memory Image Upload, and enter the file name – now you have the backup
>
> To restore a unit, or transfer to a compatible unit: First download the backup file by clicking File, Download, and select the proper file; Once the download completes, be sure to save the main memory into NV memory so that it is permanently available. *Note that transferring parameters between units with different setting ranges (such as max current, current range, steps-per-second maximum speeds, etc.) may require adjustments to fit to that particular model.*

Command strings and program segments should be fully compatible among all models in the DC-series.

File Formats and Tools

The **Memory Image Upload** is a raw memory dump image file. It is good for storing copies of final configurations, but is not the best tool for documenting, writing or modifying programs or command strings. It is stored in a format that is easily re-downloaded, even though the data format is memory-value-oriented, rather than command-oriented. The format of each line in the file is:

"<\> <targetaddress> <space> <targetvalue>"

The **.SMC data file** (used by Download and Edit... menu item) utilizes a "command-string" format, where each line in the download file consists of a DC-series Command and its command parameters. Note that the programs are entered and appear in each line of the file as if it were typed to the controller by keyboard entry. Downloaded commands in immediate mode are immediately executed, and commands entered in **Program Entry Mode** (entering and exiting the mode with a P command) are loaded into system program memory. When using the Download command, this flexibility allows system parameter setup and program creation with the same file – an example of a program file follows (in black) with some comments added for the reader:

```
     C2         'clear all parameters to factory default
     C3         'clear all program memory to blank (erase all programs)

     V2000      'set specific V setting
     Y20 80     'set specific Y setting
     I400       'set specific I setting
     U2 8       'set input 2 to limitA
     U3 3       'set input 3 to GO
     P          'starts Program Mode – commands following go into memory
0    +1000      'move + 1000
5    W0         'wait for completion of move
8    w4         'output3 enabled
10   +10        'begin loop here, by slowly moving forward
15   L10 1      'loop back to 10 if sensor not active, fall through if active
19   w0         'output 3 disabled
               •
               •   <-- rest of the program entered here
               •
128  P          'ends Program Mode – following commands executed

     S0         'stores all parameters into non-volatile memory
     S1         'stores all program entries into non-volatile memory
```

This program starts by clearing the parameter and program memory. Then, it directly sets parameter settings **V**, **Y**, **I** and some **U** fixed functions, before starting Program Entry Mode with the **P** command. Note the orange addresses shown for reference only – these are provided by the controller to show where the next line will go in memory, and will show up as a response from the unit but are not part of the .SMC data file format. Finally, the S0 and S1 commands lock the parameters and program memory into non-volatile (NV) flash memory, so it will be permanent. **The parameters and program will be loaded at power-on** and program will start when GO input (input 3) is activated, with a switch or sensor.

ASCII Character Code

Ctrl	Char	Dec	Hex	Code		Dec	Hex	Char		Dec	Hex	Char		Dec	Hex	Char
^@		00	00	NUL		32	20			64	40	@		96	60	`
^A	☺	01	01	SOH		33	21	!		65	41	A		97	61	a
^B	☻	02	02	STX		34	22	“		66	42	B		98	62	b
^C	♥	03	03	ETX		35	23	#		67	43	C		99	63	c
^D	♦	04	04	EOT		36	24	$		68	44	D		100	64	d
^E	♣	05	05	ENQ		37	25	%		69	45	E		101	65	e
^F	♠	06	06	ACK		38	26	&		70	46	F		102	66	f
^G	•	07	07	BEL		39	27	‘		71	47	G		103	67	g
^H	◘	08	08	BS		40	28	(		72	48	H		104	68	h
^I	○	09	09	HT		41	29	)		73	49	I		105	69	i
^J	◙	10	0A	LF		42	2A	*		74	4A	J		106	6A	j
^K	♂	11	0B	VT		43	2B	+		75	4B	K		107	6B	k
^L	♀	12	0C	FF		44	2C	,		76	4C	L		108	6C	l
^M	♪	13	0D	CR*		45	2D	-		77	4D	M		109	6D	m
^N	♫	14	0E	SO		46	2E	.		78	4E	N		110	6E	n
^O	☼	15	0F	SI		47	2F	/		79	4F	O		111	6F	o
^P	►	16	10	DLE		48	30	0		80	50	P		112	70	p
^Q	◄	17	11	DC1		49	31	1		81	51	Q		113	71	q
^R	↕	18	12	DC2		50	32	2		82	52	R		114	72	r
^S	‼	19	13	DC3		51	33	3		83	53	S		115	73	s
^T	¶	20	14	EC4		52	34	4		84	54	T		116	74	t
^U	§	21	15	NAK		53	35	5		85	55	U		117	75	u
^V	▬	22	16	SYN		54	36	6		86	56	V		118	76	v
^W	↨	23	17	ETB		55	37	7		87	57	W		119	77	w
^X	↑	24	18	CAN		56	38	8		88	58	X		120	78	x
^Y	↓	25	19	EM		57	39	9		89	59	Y		121	79	y
^Z	→	26	1A	SUB		58	3A	:		90	5A	Z		122	7A	z
^[	←	27	1B	ESC		59	3B	;		91	5B	[		123	7B	{
^\	∟	28	1C	FS		60	3C	<		92	5C	\		124	7C	|
^]	↔	29	1D	GS		61	3D	=		93	5D	]		125	7D	}
^^	▲	30	1E	RS		62	3E	>		94	5E	^		126	7E	~
^_	▼	31	1F	US		63	3F	?		95	5F	_		127	7F	

* On recent keyboards generated by Enter↵ key

Revision Log

July 11, 2018 – Pre-Release Revision 0.7

Contact AccelMotion

Email: support@accelmotion.com
Phone: 512-212-7300

AccelMotion Systems 3051 N Hwy 183, Unit 4 Liberty Hill, TX 78642 www.accelmotion.com