Migrating from the AMS DCB-241 / 242 or DCB-261 / 262 to the AccelMotion DC2C

The AMS DCB-241/242 and DCB-261/DCB-262 are classic driver controllers made by AMS. For previous users of these models, this guide can help allow an easy upgrade to the DC2C or other DC-series products.

This document outlines the similarities and differences, for benefit of engineers and programmers implementing the transition.

Compatibilities and Similarities

The AccelMotion DC2C is based on Advanced Micro Systems product technology, and thus the general structure of programming and console commands is very similar. Many aspects work the same, including:

- All stepping commands ***+, –, R+, R–, M***
- Parameter setups for motion, such as commands ***V, I, K, D, B***
- Support and action commands like: ***<space>*** to start single-axis mode, ***<ESC>***, ***Control-C***, ***@*** (soft stop), ***G*** (go), ***P*** (program mode), ***Q*** (query), ***F*** (home), ***J*** (jump-loop), ***W*** (wait), ***X*** (examine), ***Z*** and ***O*** (set and read position), ***^*** (moving), and location 192 for auto start on power up

Features to Review – Strongly Suggested

There are features which have different usage between the DCB- and DC-series, and these are described below. The first section shows features *strongly recommended for review*, and the lower section shows features which would optionally be reviewed only if they are used in the existing application being migrated.

Output Drive Current setting

When setting max drive current levels on the DCB-26X a potentiometer was used for Drive Current. To set Hold Current the ***E command*** is used, and sets the percent of move current used during hold (non-drive) periods. Instead, the DC2C uses the ***Y command*** to allow the setting of both Drive and Hold current programmably – **the *E command* is no longer used.** See ***Y Command*** on page 59 of the DC2C user manual – the values are percents of the stated max current, so Y50 100 would set hold current to 50% (1A) and run current to 100% (2A).

Microstepping and Ratios (H and D)

The DCB-26X and DC2C microstepping resolutions are both set with the ***H Command*** – but the breakdown of how the parameter value corresponds to the resolution is different. Check the DC2C manual page 41 for the ***H*** values.

Program memory (main and non-volatile)

The DCB-series provides 2048 bytes of program memory, but the DC2C supports 1024.

Ports and Fixed Functions (such as Go, Stop, Limits, Jog, and Home)

The DCB-series provides a selection of fixed function inputs like Go, Jog, Limits, etc. and a few general-purpose I/Os, but if a function isn't used then that signal and input to the controller are simply wasted and can't be used for anything else. On the other hand, the **DC2C implements fixed functions by allowing the user to associate each needed function with <u>any</u> available input port**, using the ***U command***. Once required fixed functions are allocated the remaining input ports can be used as general-purpose inputs (which can be used to affect program flow in conditional statements using the ***L command***, or can be polled by the host using the ***A command***).

This sharing method allows a balance of fixed and user functions, for various applications with differing needs.

To upgrade, decide which fixed functions were used on the DCB system and create a series of ***U command*** statements that will associate those functions with the desired inputs on the DC2C (see DC2C User Guide pg. 54).

An example would be a DCB-series system which uses GO, LIMIT+, LIMIT-, and STOP, but did not use HOME, JOG+ or JOG-; it was determined that GO would input on input port1, STOP on port 2, L+ on port 3 and L- on port 4; the following commands are constructed for the DC2C setup script:

U1 3	←	associates GO (type 3) with port I1
U2 4	←	associates STOP (type 4) with port I2
U3 8	←	associates Limit+ (type 8) with port I3
U4 9	←	associates Limit- (type 9) with port I4

This example DC2C configuration would leave six ports (I5 - 12) available for any general-purpose I/O input.

Input and Output Port Reads and Writes

The DCB-26X controls the input and output ports through the ***A command***. The DC2C separates the input and output ports completely, and utilizes different commands for reading inputs (***A***) and setting outputs (***N*** and ***w***).

Input ports

The DC2C ***A command*** reads input ports similar to the DCB-261, but does not require a “A129” postfix code. Simply send “**A**”, and the response is a decimal number from 0 – 4095 encoding all 12 inputs values, *even if they have some fixed function attached to their port.*

If the DCB used Read Limits ***] Command*** to access fixed function values, then the ***A command*** can be used instead to give the same access.

Example:

In a system with some ports in use, send “A”, receive response “A177” meaning active status (ground) on ports 8, 6, 5, and 1. From our earlier example this would also mean that the (fixed function) GO button (on port I1) was active.

Output Ports

The DC2C can modify output port values two ways: the ***small-w command*** changes all output values at one time (similar to the old A command) with a single output value, or the new ***N Command*** changes only one output port value at a time without affecting any other output values.

Examples:

- send “w14” which turns on outputs 4, 3, and 2, simultaneously
- send “w0” which clears all outputs, simultaneously
- send “N2 1” which sets output 2 to a 1 (active), and changes no other output value
- send “N1 0” which clears output 1 to a 0 (inactive), and changes no other output value

Wait, for DCB-24X only

The W Command (wait) on the DCB-241 or DCB-242 were specified in milliseconds and not in 10 millisecond tics. Be sure to adjust wait values for DCB-24X units when moving to DC2C, which uses tics like the DCB-26X units.

Features to Review only if Explicitly Used

Review the following table of specific DCB-series functions or commands to see if they are used in your application. If so, review the suggestions and reference the paragraphs of additional data below the table as well as the user manuals for the older product as well as the DC2C.

If this Command or feature used	Description of issue between DCB and DC-series	Suggested modification
A Command*	***A command*** no longer changes outputs	Use ***N*** or ***w command*** for outputs
	A command reads inputs without "A129"	Use ***A command*** directly
L Command*	Loop function condition codes different	Review and modify ***L*** condition codes
Go Table*	Go Table not supported	Translate to ***L Command*** branches
Automatic Go*	Program segments at location 1600 auto-start at power up	Move segment to location 192
T, k, i*	Trip point function not provided on DC-series	Reorganize moves or use delays
w command	***w command*** (pre-energize) is not used	Move without pre-energize
C Command	***C Command*** codes are different	Review ***C*** Codes
S Command	***S Command*** codes are different	Review ***S*** Codes
b Command	***b command*** sets older unneeded drive option	No longer needed – remove
E Command	Previously used to set hold current percentage	***Y command*** for all current settings
] Command	] read limit switch commands are unneeded	Use A to read any input (even fixed)
| Command	***| Command*** (pipe) no longer needed	In program, use ***P*** (end) In host control, use ***@*** Soft Stop
Position counter	O and Z refer to position counter of 24-bit size	DC-series uses 32-bit position counter
Normally-closed switch for limits	***E*** or ***l command*** moved to ***p command***	Translate to ***p Command*** setting
Normally-closed switch for HOME	Change inherent method to ***p command***	Use ***p Command*** setting

* these commands are only valid in internal programming and are not used under host control

Loop on Condition/Port, general

The DC2C *L Command* Loop-on-port is used only for internal programming. Loop Command works on DC-series just like in the DCB-series, using an attribute to change execution flow to a new address when a condition occurs.

Condition values and polarities for DC2C are different from the DCB-series, so the programmer should review the locations and condition codes for all loops.

The DC2C **does** support the L2048 special mode which jumps back to the loop start address while waiting.

Go Table (G2048)

The older G Command special case (G2048) was used in the DCB-series to allow a fixed pattern of execution to be distributed to an array of program address locations – this function is not supported on the DC2C due to the fixed nature of the addresses. Instead the L Command can easily allow conditional programming branching on any input, output or status value, to construct a matrix table easily.

Any existing Go Table must be complied into a number of L Command branches which correspond to the number of table entries used – refer to the user manual or contact `support@AccelMotion` for help

Automatic Go on Power-up (location 1600)

The DCB-series product can auto-start a program at power-up/reset – this is accomplished by starting a program segment at location 1600. The DC-series operates similarly, but the program segment must start at 192.

Trip Functions (T, k, and i)

The DCB-26X Trip Point functionality is not implemented on the DC-series. AccelMotion suggests that all motion commands be implemented explicitly such that location detectors are not necessary. Possible workarounds are multiple motions commands, or wait commands after motion starts for approximate delays.

An example might be a to achieve a specific distance or time within a motion before signaling an output relay, light, or other output. Where the DCB unit might set up a T Command, with a trip location for the delay, a DC-series device could use time for the delay:

```
0       +100000       ←  start large move
5       W100          ←  wait 100 tics or 1 second from start of move (arbitrary example)
8       N7 1          ←  output 1 second delayed from start but during move
10      W0            ←  special “wait for motion complete” delay
13      N8 1          ←  other outputs as needed after motion stopped
.
.                     ←  continued program
```

w Command (pre-energize)

Because the DC-series has advanced driver, there is no need for pre-energizing to drive levels. w command is reused on DC-series products to modify output port data, so carefully review any prior use of this command.

C Command and S Command

C Command and *S Command* modify the parameters for motion and control, as well as program space, in both the main system memory as well as the non-volatile memory. *C Command* (clear) only clears or resets the main memory contents, and the *S Command* (save) only saves the main memory contents to non-volatile (NV) memory. Any permanent change to NV memory must occur furst in main memory then moved to NV via the *S command*.

Note that the data values which correspond to clearing, factory reset, and saving are different for the DC2C (review DC2C User Guide). Examples: **C2** resets parameters to factory settings, and **S0** moves parameters to NV.

b Command – Fast/Slow decay

b command is not used or needed in DC-series products. DC-series drivers utilize automatic fast/slow decay settings to optimize power and accuracy in all situations.

Read Limits Command]

If the DCB used Read Limits ***] Command*** to access fixed function values, then the ***A command*** can be used instead to give the same access.

| Command (pipe) – Selective Termination

This command was used on DCB-26X to allow two kinds of aborts: program end, when placed in program, and individual unit stop command when used by host (in single- and multi-axis modes). For program abort in DC-series, simply end each program segment with no command (by ending program entry with a P, leaving an empty space). For unit stop with DC-series, use the @ soft stop command which stops only the addressed controller.

Location/Position Counter – O and Z

The DC2C uses a 32-bit position counter, and the DCB series used a 24-bit counter. This may not be of concern, but is mentioned here for completeness. The ***O*** and ***Z Commands*** each are affected by this value.

Using Normally-closed Limit switches

The DCB-26X configures for normally-closed limit switches via the (lower-case L) ***l command***, which sets limit polarity (as well as some output port configurations). These functions are now moved to the ***p command*** – review the p command user guide to determine the correct polarity (normally-closed or normally-open) data value.

The DCB24X utilizes the ***E Command*** for setting polarity, and should be migrated to the ***p command.***

Using Normally-closed HOME switch

The DCB-series provides for normally-closed HOME switch by modifying the automatic-home capability to allow reversed operation. The DC-series improves the HOME functionality by providing for automatic high-accuracy homing in *normally-closed and normally-open mode*, once polarity is set by the ***p command*** – review the p command user guide to determine the correct polarity (normally-closed or normally-open) data value.

E Command (Hold Current)

When setting max drive current levels on the DCB-26X a potentiometer is used for Drive Current. To set Hold Current the ***E command*** is used, and sets the percent of move current used during hold (non-drive) periods. Instead, the DC2C uses the ***Y command*** to allow the setting of both Drive and Hold current programmably – **the *E command* is no longer used.**

See ***Y Command*** on page 59 of the DC2C user manual – the values are percents of the stated max current, so Y50 100 would set hold current to 50% (1A) and run current to 100% (2A).

Hardware

Mechanical

The DC2C is smaller in size than the DCB-24X and DCB-26X family so should fit in the same space easily. The connectors for the DC2C are on the front of the unit for simplified ease of connection.

Electrical/Functional Comparison

The DC2C electrical specifications match or exceed the DCB241 and DCB261/262 products from AMS. The table below shows the comparison of the DC2C and the replaced models.

		DCB241		DCB261		DCB262		DC2C		
Characteristic	**Value**	Min	Max	Min	Max	Min	Max	**Min**	**Max**	**Unit**
Supply Voltage	V_{MM}	24	40	24	40	24	40	**10**	**40**	V (DC)
Reference Voltage for Input/Output	V_{IO}	0	28	0	28	0	28	**0**	**36**	V
Output Drive Current/Phase	i_{phase}	-	1.2	-	1	-	2	**-**	**2**	A
Output Current adjustment		Potentiometer fixed		Potentiometer fixed		Potentiometer fixed		**Programmable real-time**		

DC-series New Features of High Value

The DC2C utilizes a user-replaceable fuse for electrical protection of equipment.

The DC2C has an internal diagnostic system which monitors input voltage, I/O voltage, and temperature, and inherent systems to protect against overcurrent and undervoltage. All faults generated from these monitored values can be queried and cleared from the host.

The DC2C has a front-facing status LED, where the DCB-series did not. This status indicator can be helpful for indicating existence of power, stepping in progress, system faults, and program operation in progress.

N Command allows direct control of individual output ports irrespective of other port settings, whereas the existing ***w command*** modifies all outputs at one time – both methods have their place.

{ Command reads machine status values of temperature, VMM (main) supply voltage, and VIO supply voltage.

} Command reads machine fault values of temperature, overtemperaure, overvoltage of VMM (main) supply voltage, overvoltage of VIO supply voltage, and driver reported fault (for overcurrent).