= = = DPCL MANUAL = = =

This is a guide for the DUO Portable command language (DPCL).

= = = SYNTAX = = =

A program contains commands separated by newlines.
A command contains a command name and arguments separated by spaces.
A command name contains exactly three letters.
An argument may be one of the following.

- Number literal: digits 0 through 9.
- Simple variable address: dollar sign followed by a number.
- Complex variable address: two dollar signs followed by a number.
- Null terminated string literal: characters enclosed by quotation marks.
- List of number literals: numbers separated by spaces enclosed by parentheses.

Longs and floats occupy 4 bytes of space.
File handles occupy 1 byte of space.
The pixel buffer starts at address 11000 and is 816 bytes long.

= = = NUMBER OPERATIONS = = =

NWB [byte A] [byte B]
NWL [long A] [long B]
NWF [float A] [float B]
Store B in A.
NLB [byte A] [long B]
NFB [byte A] [float B]
Convert B to byte and store in A.
NBL [long A] [byte B]
NFL [long A] [float B]
Convert B to long and store in A.
NBF [float A] [byte B]
NLF [float A] [long B]
Convert B to float and store in A.
NBS [string A] [byte B]
NLS [string A] [long B]
NFS [string A] [float B]
Convert B to string and store in A.

= = = LIST OPERATIONS = = =

LWB [byte list A] [long B] [byte list C]
LWL [long list A] [long B] [long list C]
LWF [float list A] [long B] [float list C]

Store list C with length B in A.
LEB [byte A] [long B] [byte list C] [byte list D]
LEL [byte A] [long B] [long list C] [long list D]
Determine whether list C and D with length B have the same values. Store the result in A.
LFB [long A] [byte B] [long C] [byte list D]
LFL [long A] [long B] [long C] [long list D]
Find element B in list D with length C. Store the index in A. If B was not found, store -1 in A.
LSB [byte list A] [long B] [byte C]
LSL [long list A] [long B] [long C]
LSF [float list A] [long B] [float C]
Stock C in each element of list A with length B.

= = = STRING OPERATIONS = = =

SWR [string A] [string B]
Store B in A.
SEQ [byte A] [string B] [string C]
Determine whether B and C are equal. Store the result in A.
SFN [long A] [string B] [string C]
Find substring B in string C. Store the index in A.
SLB [byte A] [string B]
SLL [long A] [string B]
Store length of string B in A.
SSB [string A] [string B] [byte C] [byte D]
SSL [string A] [string B] [long C] [long D]
Store substring of B from index C inclusive to index D exclusive in A.
SCN [string A] [string B]
Concatenate string B to string A.
SNB [byte A] [string B]
Convert string B to byte and store in A.
SNL [long A] [string B]
Convert string B to long and store in A.
SNF [float A] [string B]
Convert string B to float and store in A.
SDL [byte A] [string B]
Store number of elements in space delimited string B in A.
SDS [string A] [string B] [byte C]
Store element from space delimited string B with index C in A.

= = = ARITHMETIC OPERATIONS = = =

AAB [byte A] [byte B] [byte C]
AAL [long A] [long B] [long C]
AAF [float A] [float B] [float C]
Add B and C and store the result in A.

ASB [byte A] [byte B] [byte C]
ASL [long A] [long B] [long C]
ASF [float A] [float B] [float C]
Subtract C from B and store the result in A.
AMB [byte A] [byte B] [byte C]
AML [long A] [long B] [long C]
AMF [float A] [float B] [float C]
Multiply B and C and store the result in A.
ADB [byte A] [byte B] [byte C]
ADL [long A] [long B] [long C]
ADF [float A] [float B] [float C]
Divide B by C and store the result in A.
ARB [byte A] [byte B] [byte C]
ARL [long A] [long B] [long C]
ARF [float A] [float B] [float C]
Divide B by C and store the remainder in A.

= = = MATHEMATICAL OPERATIONS = = =

MIB [byte A]
MIL [long A]
Increment A.
MDB [byte A]
MDL [long A]
Decrement A.
MRB [byte A] [byte B]
MRL [long A] [long B]
Generate random number between 0 inclusive and B exclusive and store in A.
MSN [float A] [float B]
Compute the sine of B and store the result in A.
MCS [float A] [float B]
Compute the cosine of B and store the result in A.
MTN [float A] [float B]
Compute the tangent of B and store the result in A.
MAS [float A] [float B]
Compute the inverse sine of B and store the result in A.
MAC [float A] [float B]
Compute the inverse cosine of B and store the result in A.
MAT [float A] [float B]
Compute the inverse tangent of B and store the result in A.
MA2 [float A] [float B] [float C]
Compute angle to the position (B, C) and store the result in A.
MPW [float A] [float B] [float C]
Raise B to the C power and store the result in A.
MSR [float A] [float B]
Compute the square root of B and store the result in A.
MLN [float A] [float B]

Compute the natural log of B and store the result in A.
MLG [float A] [float B]
Compute the log base 10 of B and store the result in A.
MAB [float A] [float B]
Compute the absolute value of B and store the result in A.

= = = BITWISE AND BOOLEAN OPERATIONS = = =

BNB [byte A] [byte B]
BNL [long A] [long B]
Compute the bitwise NOT of B and store the result in A.
BOB [byte A] [byte B] [byte C]
BOL [long A] [long B] [long C]
Compute the bitwise OR of B and C, and store the result in A.
BAB [byte A] [byte B] [byte C]
BAL [long A] [long B] [long C]
Compute the bitwise AND of B and C, and store the result in A.
BIB [byte A] [byte B]
BIL [long A] [long B]
Compute the Boolean inverse of B and store the result in A.
BLB [byte A] [byte B] [byte C]
BLL [long A] [long B] [byte C]
Bit shift B to the left by amount C and store the result in A.
BRB [byte A] [byte B] [byte C]
BRL [long A] [long B] [byte C]
Bit shift B to the right by amount C and store the result in A.

= = = COMPARISON OPERATIONS = = =

CEB [byte A] [byte B] [byte C]
CEL [byte A] [long B] [long C]
Determine whether B and C are equal, and store the result in A.
CGB [byte A] [byte B] [byte C]
CGL [byte A] [long B] [long C]
CGF [byte A] [float B] [float C]
Determine whether B is greater than C, and store the result in A.
CLB [byte A] [byte B] [byte C]
CLL [byte A] [long B] [long C]
CLF [byte A] [float B] [float C]
Determine whether B is less than C, and store the result in A.

= = = FLOW OPERATIONS = = =

FRD [long A]
Store the current command address in A.
FWR [long A]

Set the current command address to A.
FIB [byte A]
FIL [long A]
Execute the following branch if value A is not zero.
FNB [byte A]
FNL [long A]
Execute the following branch if value A is zero.
FEN
End the preceding branch.
FWB [byte A]
FWL [long A]
Execute the following branch while value A is not zero.
FBR
Terminate the current while branch.
FSB [long A]
Declare the following block as a subroutine and store a reference to the subroutine in A.
FCL [long A]
Call subroutine with reference A.

= = = KEY OPERATIONS = = =

KRD [byte A]
Determine which key is being pressed and store the result in A.
KIP [byte A] [byte B]
Determine whether key B is being pressed and store the result in A.

= = = TIME OPERATIONS = = =

TWR [long A]
Set the current time in milliseconds to A.
TRD [long A]
Store the current time in milliseconds in A.
TSB [byte A]
TSL [long A]
Sleep A milliseconds.

= = = DISPLAY OPERATIONS = = =

DCL
Clear the display.
DPX [byte A] [byte B] [byte C]
Draw pixels C at horizontal position A and vertical position B.
DIM [byte A] [byte B] [long C] [byte list D] [byte E]
Draw pixels D with length C at horizontal position A and vertical position C with width E.
DST [byte A] [byte B] [string C]

Draw string C at horizontal position A and vertical position B.
DNB [byte A] [byte B] [byte C]
DNL [byte A] [byte B] [long C]
DNF [byte A] [byte B] [float C]
Draw number C at horizontal position A and vertical position B.
DPT [byte A] [byte B] [byte C]
Draw point in the pixel buffer with color C at position (A, B).
DLN [byte A] [byte B] [byte C] [byte D] [byte E]
Draw line in the pixel buffer with color E from position (A, B) to (C, D).
DRC [byte A] [byte B] [byte C] [byte D] [byte E]
Draw rectangle in the pixel buffer with color E at position (A, B) with width C and height D.
DBF
Redraw the pixel buffer.

= = = PROMPT OPERATIONS = = =

PST [string A]
Prompt with starting string A and store the result in A.
PSL [byte A] [byte B] [string list C]
Prompt selection from options C with amount B and store the result in A. The strings in list C must have start indexes which are multiples of 17.
PFL [byte A]
Prompt file and store the handle in A.
PKY [byte A]
Prompt key and store the result in A.
PNB [byte A]
PNL [long A]
PNF [float A]
Prompt number and store the result in A.

= = = EXCHANGE PIN OPERATIONS = = =

XSM [byte A] [byte B]
Set pin A to mode B. If B is zero, the mode is input. If B is one, the mode is output.
XDR [byte A] [byte B]
Digital read pin B and store the result in A.
XDW [byte A] [byte B]
Digital write value B to pin A.
XAR [long A] [byte B]
Analog read pin B and store the result in A. The result will be between 0 and 1023 inclusive.

= = = REPOSITORY OPERATIONS = = =

RNF [byte A]

Store number of files in A.

ROI [file handle A] [byte B]

Find file with index B, and store handle in A.

RON [file handle A] [string B]

Find file with name B, and store handle in A.

RCR [file handle A] [string B] [long C]

Create file with name B and size C, and store handle in A.

RRD [byte list A] [long B] [file handle C] [long D]

Read B bytes at index D from file C, and store the data in A.

RWR [file handle A] [long B] [long C] [byte list D]

Write C bytes from D into file A at index B.

RGN [string A] [file handle B]

Store name of file B in A.

RSN [file handle A] [string B]

Set name of file A to B.

RGS [long A] [file handle B]

Store size of file B in A.

RSS [file handle A] [long B]

Set size of file A to B.

RDL [file handle A]

Delete file A.

RRN [file handle A]

Run file A.