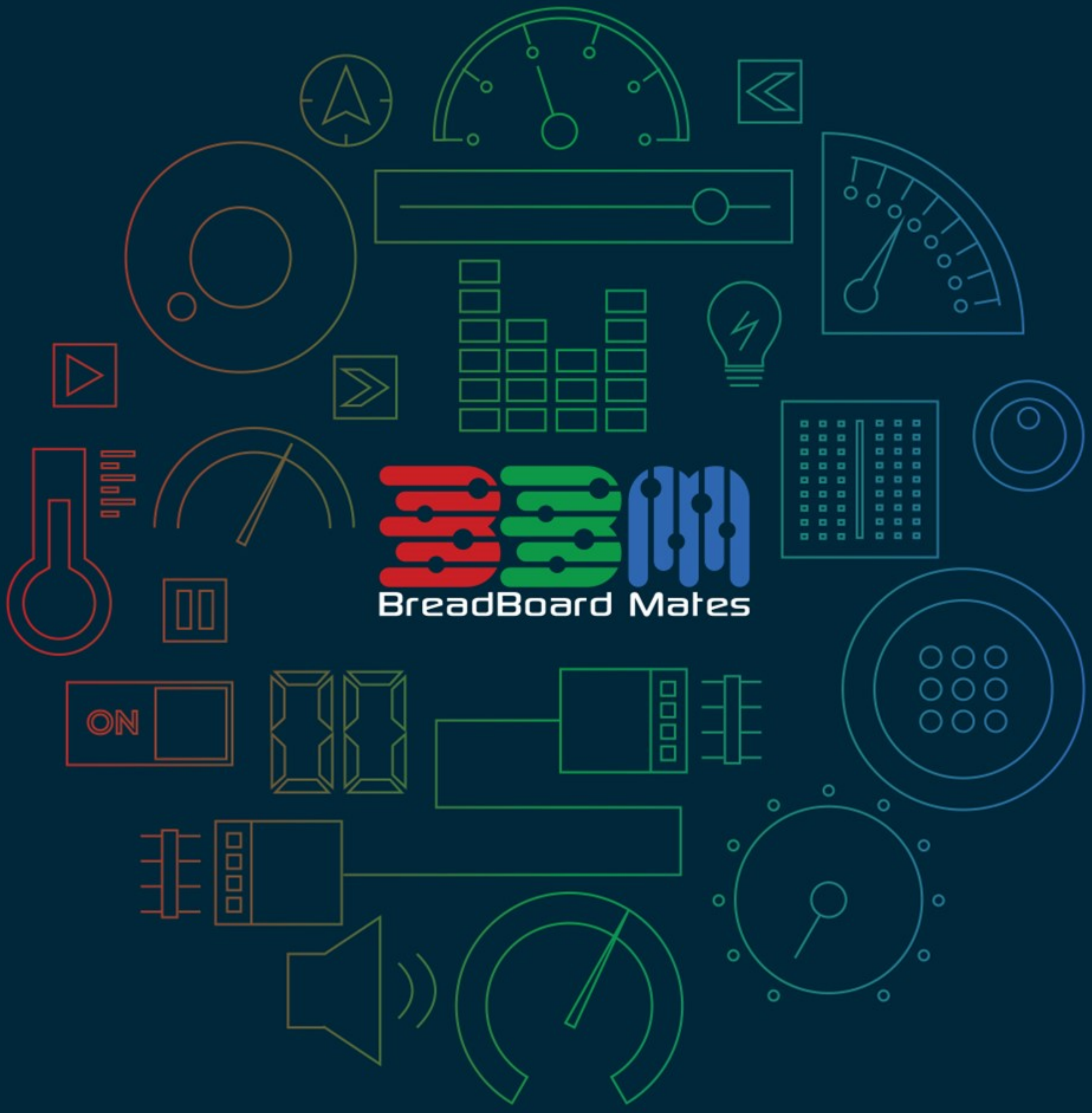


Mates Controller Protocol

REFERENCE MANUAL

Content may change at any time. Please refer to the resource centre for latest documentation.

Contents

Introduction 4

Boot Sequence 4

Command Summary 4

Basic Commands 5

Set Page 5

Get Page 6

Set Backlight 7

System Reset 8

Common Widget Commands 9

Set Widget Value 9

Get Widget Value 10

Set Widget Parameter 11

Get Widget Parameter 12

Special Widget Commands 13

Set 32-bit Widget Value 13

Update Text Area 14

Clear Print Area 15

Set Print Area Color 16

Append to Print Area 17

Append to Scope 18

Update Dot Matrix 19

Touch Input Commands 20

Get Number of Button Events 20

Get Number of Swipe Events 21

Get Next Button Event 22

Get Next Swipe Event 23

Swipe Value Reference 23

General Purpose I/O Commands 25

Set Pin Mode 25

Set Pin State 26

Get Pin State 27

Introduction

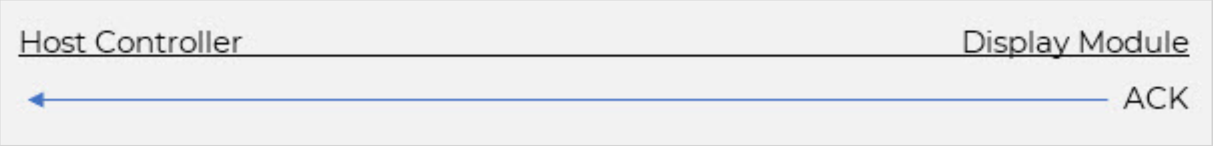
Mates Studio's Commander and Architect environments are designed to create user interfaces for Breadboard Mates' display products with the purpose of using these with the user's preferred host controller.

Architect and Commander projects utilizes the same simple Serial Command protocol allowing any host controller to communicate with the display modules. The protocol features commands including, but not limited to, updating, and reading widget value, changing backlight level, and changing widget color parameters.

Boot Sequence

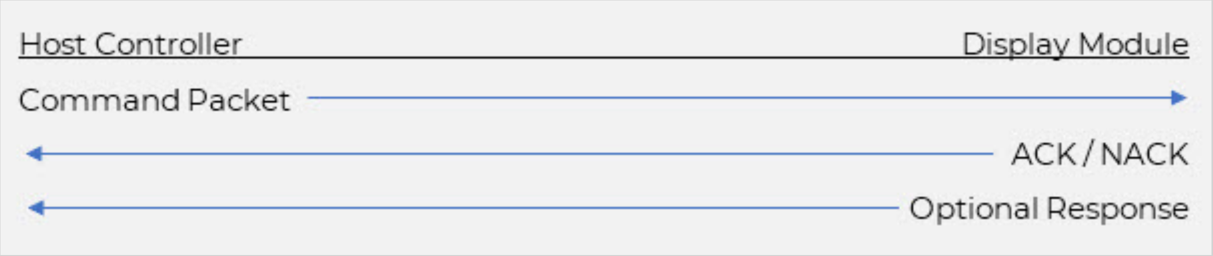
During boot of Architect and Commander projects, the display starts by performing its initial setup which includes mounting external storage devices, displaying initial page *Page0* and initializing UART for receiving commands and transmitting replies.

After setting up everything required, the display sends a ACK `0x06` signifying that the display is ready to accept commands from the host controller.



Command Summary

The command protocol features a simple data exchange format between the host controller and the display module. All commands come from the host controller. After receiving a command from the host, the display performs the appropriate actions and replies to the host controller appropriately. The display's reply always starts with an ACK `0x06` followed by an appropriate response as required, or a NACK `0x15` if the command fails.



Basic Commands

Commands for controlling the display's basic functionality are included and discussed in this section. These commands include activating a specific page, querying the active page, setting backlight level, and performing a soft reset.

Set Page

Sets the active page shown by the project

Parameters	Type	Description
Command	Command	`0x0000`
Index	16-bit Integer	Specifies the target page to activate

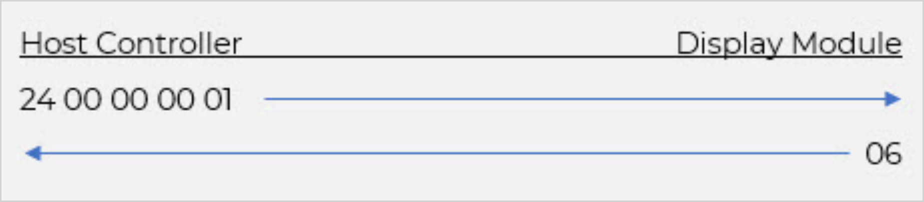
Response

None

Example

Success

Successfully set the active page to Page1 [0x0001]

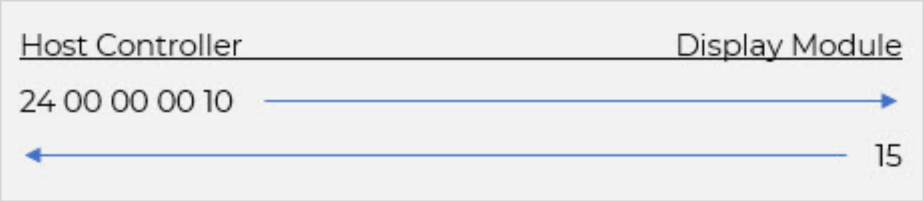


24 00 00 00 01

06

Failed

Failed attempt to set the active page to Page16 [0x0010]



24 00 00 00 10

15

Get Page

Queries the active page

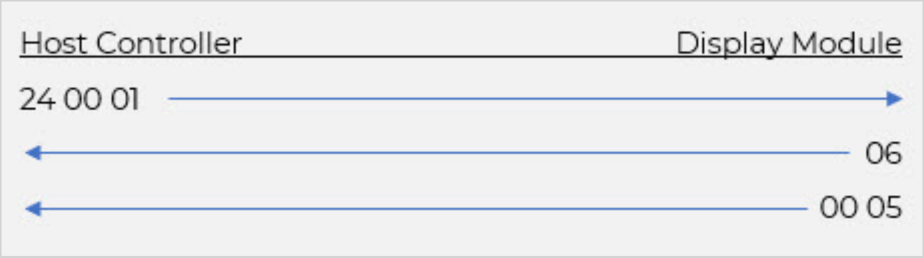
Parameters	Type	Description
Command	Command	`0x0001`

Response

Active Page Index

Example

Queried the active page and got Page5 [0x0005]



Set Backlight

Sets the backlight level of the display module

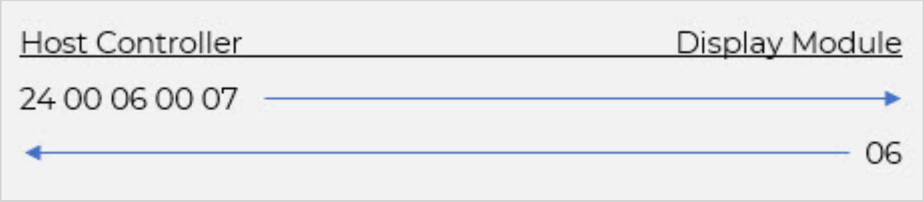
Parameters	Type	Description
Command	Command	`0x0006`
Level	16-bit Integer	Specifies the target backlight level [0 to 15]

Response

None

Example

Successfully set backlight level to 7 [0x0007]



System Reset

Performs a software reset

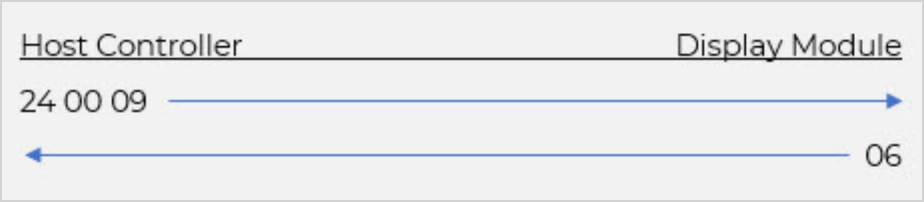
Parameters	Type	Description
Command	Command	`0x0009`

Response

None

Example

Performs a software reset and waits until the display is ready



Note

The ACK from the display module is the same acknowledgement received during a boot sequence. This signifies that the display is ready to receive and process commands.

Common Widget Commands

Most of Mates Studio widgets hold a 16-bit integer value which can be set and queried by the host controller. Most widgets also include the feature to change and read certain color parameters during runtime. Commands related to these features are included and discussed in this section.

Note

1. To check widget compatibility, please refer to the Mates Widgets Compatibility section of the Graphics Editor documentation.
2. This function is not applicable to *Int32* and *Float* LedDigits

Set Widget Value

Update the target widget to the specified value

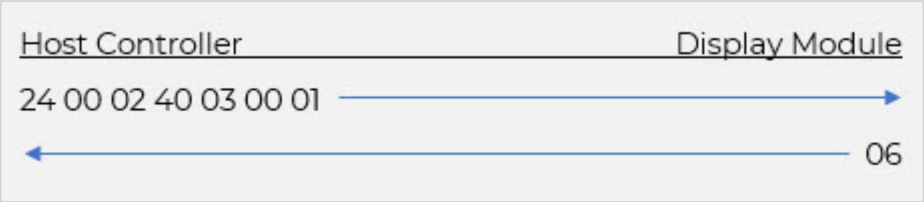
Parameters	Type	Description
Command	Command	`0x0002`
Type	8-bit Integer	Specifies the type of target widget
Index	8-bit Integer	Specifies the index of target widget
Value	16-bit Integer	Specifies the new value

Response

None

Example

Successfully turns ON [0x0001] Media Led [0x40] index 3 [0x03]



```
24 00 02 40 03 00 01
06
```

Get Widget Value

Queries the value of the target widget

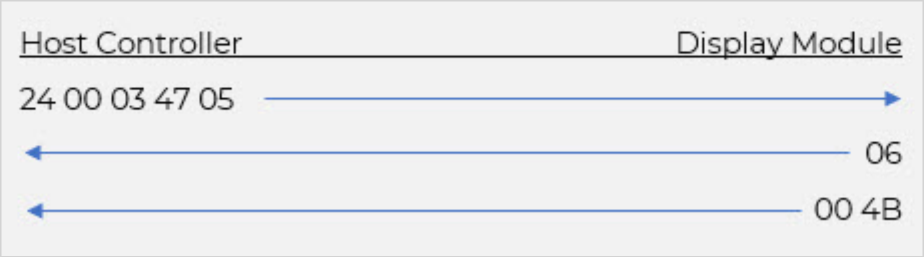
Parameters	Type	Description
Command	Command	`0x0003`
Type	8-bit Integer	Specifies the type of target widget
Index	8-bit Integer	Specifies the index of target widget

Response

Value of the specified widget

Example

Queried the value of Media Gauge B [0x47] index 5 [0x05] and got 75 [0x004B]



Set Widget Parameter

Sets the specified widget's parameter to a new value

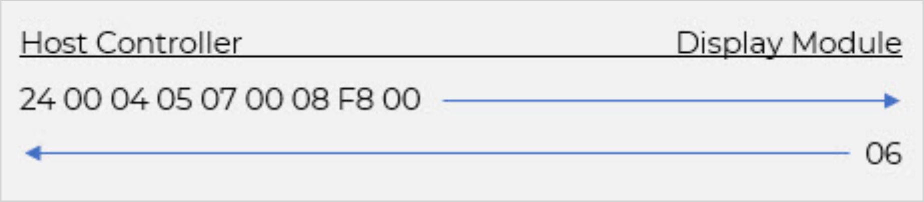
Parameters	Type	Description
Command	Command	`0x0004`
Type	8-bit Integer	Specifies the type of target widget
Index	8-bit Integer	Specifies the index of target widget
Parameter	16-bit Integer	Specifies the target parameter
Value	16-bit Integer	Specifies the new value

Response

None

Example

Successfully changes the Segment Color On [0x0008] parameter of Led Digits [0x05] index 7 [0x07] to RED [0xF800]



```
24 00 04 05 07 00 08 F8 00
06
```

Get Widget Parameter

Queries the parameter value of the target widget

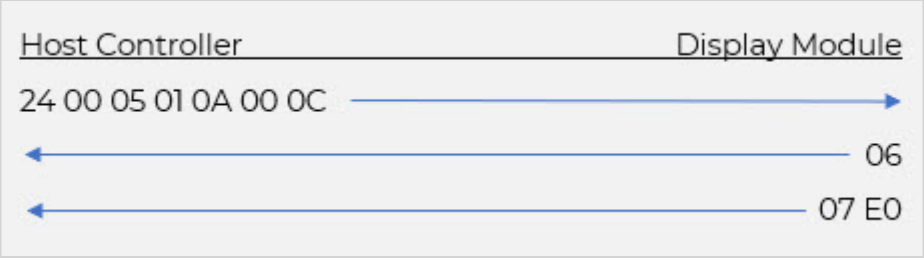
Parameters	Type	Description
Command	Command	`0x0005`
Type	8-bit Integer	Specifies the type of target widget
Index	8-bit Integer	Specifies the index of target widget
Parameter	16-bit Integer	Specifies the target parameter

Response

Value of the specified widget parameter

Example

Queried the value of Ruler Gauge [0x01] index 10 [0x0A] Partition 1 Color [0x000C] and got GREEN [0x07E0]



24 00 05 01 0A 00 0C

06

07 E0

Special Widget Commands

Some of Mates Studio widgets hold a 32-bit integer value or string, instead of a 16-bit integer value, which can be set by the host controller by utilizing special commands. Some widgets include unique features that adds more versatility to projects.

Set 32-bit Widget Value

Update the target widget to the specified value

Parameters	Type	Description
Command	Command	`0xFFFC`
Type	8-bit Integer	Specifies the type of target widget
Index	8-bit Integer	Specifies the index of target widget
Value	32-bit Value	Specifies the new float or long value

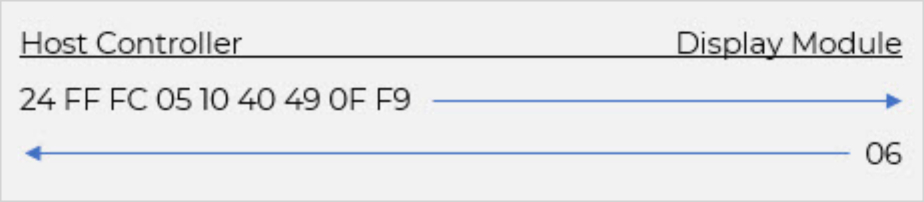
Response

None

Example

32-bit Float

Sets the value of a Float-mode Led Digits [0x05] index 16 [0x10] to 3.1416 [0x40490FF9]

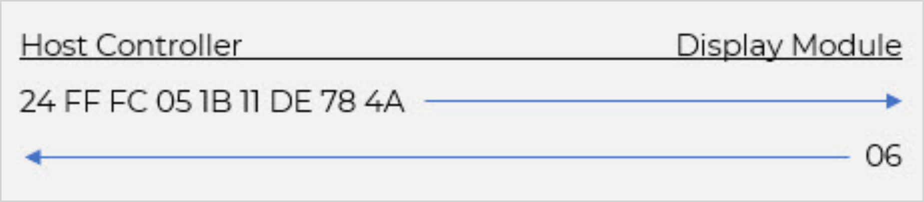


24 FF FC 05 10 40 49 0F F9

06

32-bit Integer

Sets the value of a Int32-mode Led Digits [0x05] index 27 [0x1B] to 299792458 [0x11DE784A]



24 FF FC 05 1B 11 DE 78 4A

06

Note

This command is only applicable to *Int32* and *Float* LedDigits

Update Text Area

Update the Text Area with the specified string

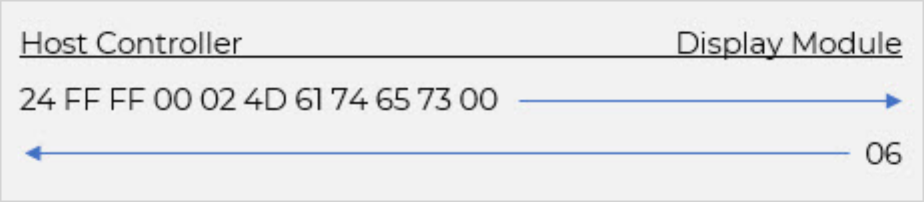
Parameters	Type	Description
Command	Command	`0xFFFF`
Index	16-bit Integer	Specifies the index of target Text Area
Text	ASCII	Specifies the new (null terminated) string value

Response

None

Example

Sets the value of a Text Area index 2 [0x0002] to "Mates" [0x4D61746573]



```
24 FF FF 00 02 4D 61 74 65 73 00
06
```

Clear Print Area

Clear the specified Print Area

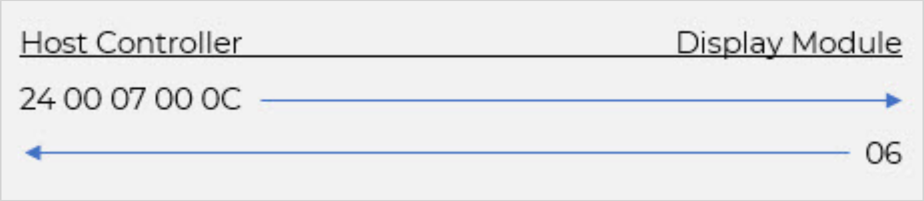
Parameters	Type	Description
Command	Command	`0x0007`
Index	16-bit Integer	Specifies the index of target Print Area

Response

None

Example

Clears Print Area index 12 [0x000C]



Set Print Area Color

Sets the color to use when appending to the specified Print Area

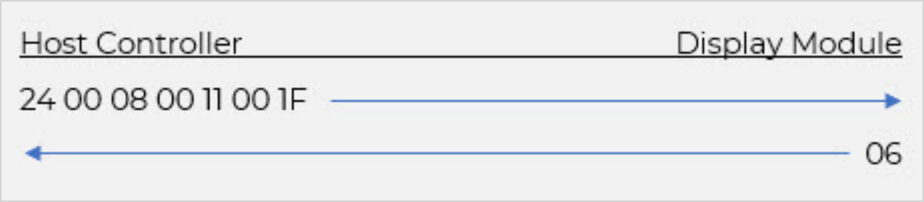
Parameters	Type	Description
Command	Command	`0x0008`
Index	16-bit Integer	Specifies the index of target Print Area
Color	RGB565	Specifies the new 16-bit color value

Response

None

Example

Sets the print color for Print Area index 17 [0x0011] to BLUE [0x001F]



```
24 00 08 00 11 00 1F
06
```

Append to Print Area

Append an array of characters or bytes to the Print Area

Parameters	Type	Description
Command	Command	`0xFFFE`
Index	16-bit Integer	Specifies the index of target Print Area
Count	16-bit Integer	Specifies number of characters or bytes to write
Data	8-bit Array	Specifies the character or byte array to write

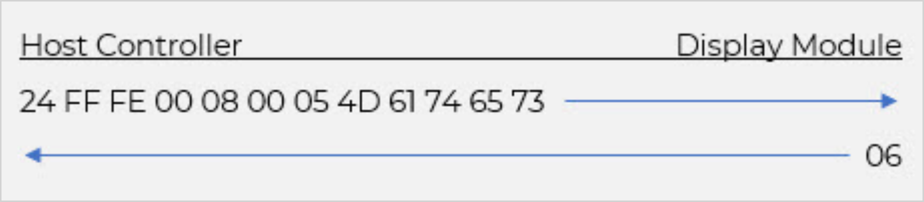
Response

None

Example

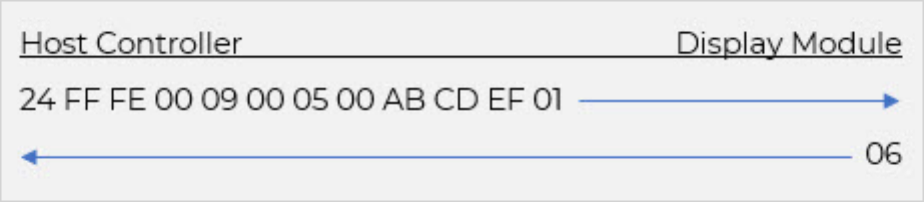
ASCII Mode

Append "Mates" [0x4D61746573] to an ASCII-mode Print Area index 8 [0x0008]



Hex Mode

Append 00 AB CD EF 01 [0x00ABCDEF01] to an HEX-mode Print Area index 9 [0x0009]



Append to Scope

Append new set of values to the specified Scope widget

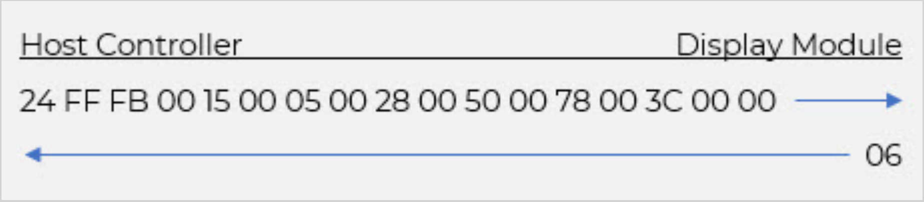
Parameters	Type	Description
Command	Command	`0xFFFB`
Index	16-bit Integer	Specifies the index of target Scope
Count	16-bit Integer	Specifies number of integers to write
Data	16-bit Array	Specifies the 16-bit data array to write

Response

None

Example

Append 40 [0x0028], 80 [0x0050], 120 [0x0078], 60 [0x003C] and 0 [0x0000] to Scope index 21 [0x0015]



24 FF FB 00 15 00 05 00 28 00 50 00 78 00 3C 00 00

06

Update Dot Matrix

Update the Dot Matrix with the specified string

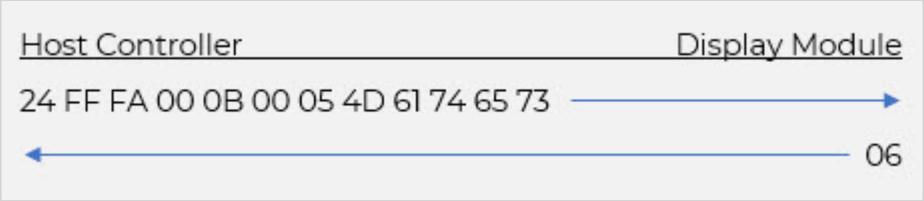
Parameters	Type	Description
Command	Command	`0xFFFA`
Index	16-bit Integer	Specifies the index of target Dot Matrix
Count	16-bit Integer	Specifies number of characters to write
Data	Character Array	Specifies the character array to write

Response

None

Example

Update "Mates" [0x4D61746573] to Dot Matrix index 11 [0x000B]



Touch Input Commands

Commands for handling select touch events such as button presses, and simple swipe actions are included and discussed in this section.

Get Number of Button Events

Queries the number of unread button events recorded by the module

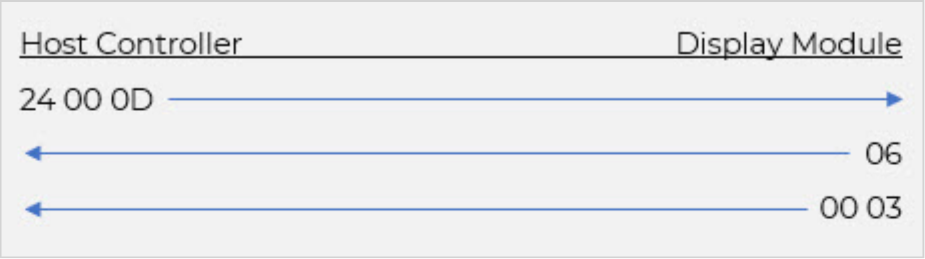
Parameters	Type	Description
Command	Command	`0x000D`

Response

Number of unread button events

Example

Queried the number of button events and got 3 [0x0003]



Get Number of Swipe Events

Queries the number of unread swipe events recorded by the module

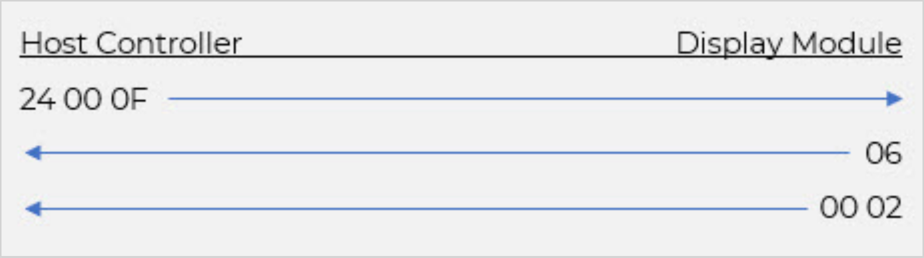
Parameters	Type	Description
Command	Command	`0x000F`

Response

Number of unread swipe events

Example

Queried the number of swipe events and got 2 [0x0002]



Get Next Button Event

Queries the next unread button event

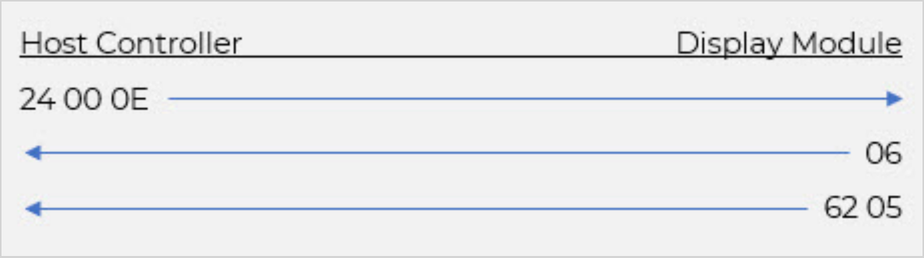
Parameters	Type	Description
Command	Command	`0x000E`

Response

Button ID [High Byte: Type, Low Byte: Index]

Example

Queried the next button event and got MediaButton5 [0x6205]



Get Next Swipe Event

Queries the next unread swipe event

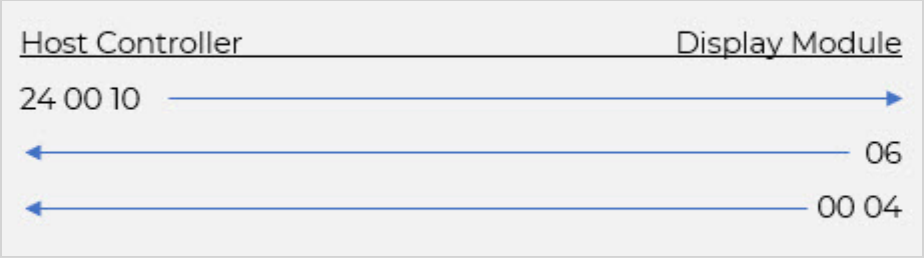
Parameters	Type	Description
Command	Command	`0x0010`

Response

Swipe Information

Example

Queried the next swipe event and got West [0x0004]



Swipe Value Reference

Swipe events can be detected as North, South, East and West.

Direction	Value
North	0b0001
South	0b0010
East	0b0100
West	0b1000

This command always returns both vertical and horizontal directions and therefore can be used to detect diagonal swipes.

The table below lists the suggested swipe flags that can be used for swipe handling.

Event	Value	Condition
MATES_SWIPE_NORTH	0b0001	From bottom to top
MATES_SWIPE_SOUTH	0b0010	From top to bottom
MATES_SWIPE_EAST	0b0100	From left to right
MATES_SWIPE_WEST	0b1000	From right to left
MATES_SWIPE_VERT	0b0011	only done vertically
MATES_SWIPE_HORZ	0b1100	only done horizontally
MATES_SWIPE_TLBR	0b0110	From top left to bottom right
MATES_SWIPE_TRBL	0b1010	From top right to bottom left
MATES_SWIPE_BLTR	0b0101	From bottom left to top right
MATES_SWIPE_BRTL	0b1001	From bottom right to top left

Here are the conditional statement examples for each of the suggested event flags

Event	Usage
MATES_SWIPE_NORTH	`(event & MATES_SWIPE_NORTH) == MATES_SWIPE_NORTH`
MATES_SWIPE_SOUTH	`(event & MATES_SWIPE_SOUTH) == MATES_SWIPE_SOUTH`
MATES_SWIPE_EAST	`(event & MATES_SWIPE_EAST) == MATES_SWIPE_EAST`
MATES_SWIPE_WEST	`(event & MATES_SWIPE_WEST) == MATES_SWIPE_WEST`
MATES_SWIPE_VERT	`(event & MATES_SWIPE_VERT) != 0`
MATES_SWIPE_HORZ	`(event & MATES_SWIPE_HORZ) != 0`
MATES_SWIPE_TLBR	`(event & MATES_SWIPE_TLBR) == MATES_SWIPE_TLBR`
MATES_SWIPE_TRBL	`(event & MATES_SWIPE_TRBL) == MATES_SWIPE_TRBL`
MATES_SWIPE_BLTR	`(event & MATES_SWIPE_BLTR) == MATES_SWIPE_BLTR`
MATES_SWIPE_BRTL	`(event & MATES_SWIPE_BRTL) == MATES_SWIPE_BRTL`

General Purpose I/O Commands

Commands for basic General Purpose I/O functions

Set Pin Mode

Sets the mode of operation of the specified pin

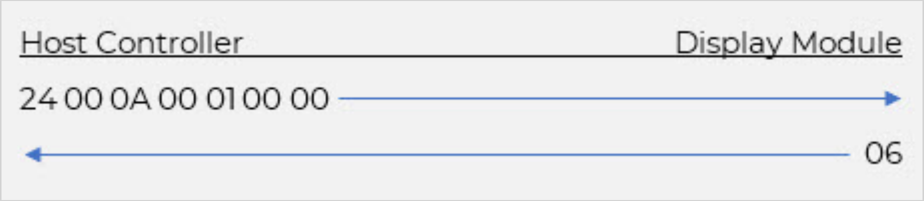
Parameters	Type	Description
Command	Command	`0x000A`
Pin	16-bit Integer	Specifies the GPIO pin number
Mode	16-bit Integer	Specifies the mode of operation 0: Output, 1: Input

Response

None

Example

Successfully sets pin 1 as an **output** pin.



Set Pin State

Sets the state of the pin previously set as output

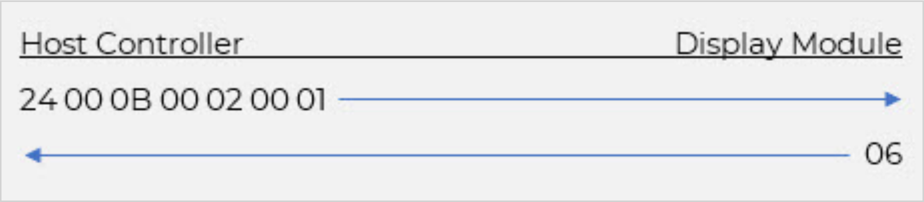
Parameters	Type	Description
Command	Command	`0x000B`
Pin	16-bit Integer	Specifies the GPIO pin number
Value	16-bit Integer	Specifies the state of the pin 0: Low, 1: High

Response

None

Example

Successfully sets pin 2 to HIGH.



Get Pin State

Queries the state of the pin previously set as input

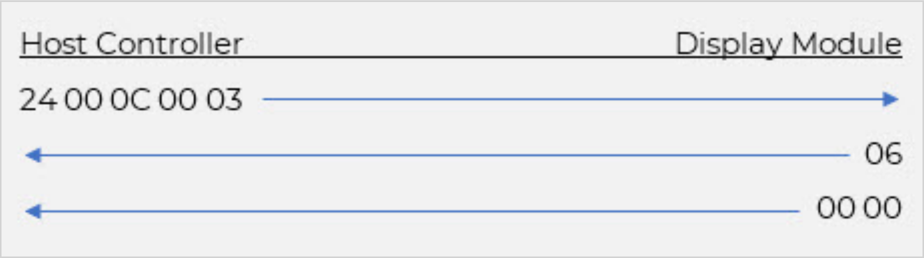
Parameters	Type	Description
Command	Command	`0x000C`
Pin	16-bit Integer	Specifies the GPIO pin number

Response

State of the pin, 0: Low or 1: High

Example

Queried the value of pin 3 and got LOW [0x0000]


Host Controller
Display Module
24 00 0C 00 03
06
00 00