



OPTIGRID MOTORIZATION
USB HID CLASS
INTERFACE INSTRUCTION

Prepared by:
Mark Knudson,
Software Engineer
20 APRIL 2007

Approved by:
Steve Mounnarat,
Director of Engineering

Version	Authorized	Date	Version	Authorized	Date
A	Mounnarat	16NOV06	D		
B		20APR07	E		
C			F		



This document outlines the command protocols used for a device Optigrd using a USB HID class.

HID Class.....	3
Commands Defined	3
Commands Detailed.....	4
1. USB_focusFindHome.....	4
2. USB_focusFindLimit.....	4
3. USB_focusZero.....	4
4. USB_focusGetCount.....	4
5. USB_focusSetCount	4
6. USB_focusSetAutoHome	5
7. USB_focusGetStatus.....	5
8. USB_gridGetVoltage.....	5
9. USB_gridSetVoltage.....	6
10. USB_gridSetControl	6
11. USB_gridGetControl	6
12. USB_getStoredData.....	6
13. USB_SetStoredData.....	7
14. USB_LCDdisplayErrorMessage.....	7
15. USB_LCDdisplayErrorMessage2.....	7
16. USB_LCDdisplayclearErrorMessage	8
17. USB_gridGetActualVoltage	8
18. USB_resetTarget.....	8
19. USB_setMotorManual	8
20. USB_getMotorManual.....	8
21. USB_enterCodedownloadMode	9

HID Class

This section describes the interface to the USB driver following a HID class.

Commands Defined

All packets are HID report # 0

Therefore the first byte sent for each packet is 0x00

Text Command	Hex Command
FOCUS_FIND_HOME	0x00
FOCUS_ZERO	0x01
FOCUS_GET_COUNT	0x02
FOCUS_SET_COUNT	0x03
SET_FOCUS_AUTO_HOME	0x04
GET_FOCUS_STATUS	0x05
GRID_GET_VOLTAGE	0x06
GRID_SET_VOLTAGE	0x07
SET_GRID_VOLTAGE_CONTROL	0x08
GET_GRID_VOLTAGE_CONTROL	0x09
GET_STORED_DATA	0x0A
SET_STORED_DATA	0x0B
LCD_DISPLAY_DATA	0x0C
LCD_DISPLAY_DATA1	0x0C
LCD_DISPLAY_DATA2	0x10
LCD_DISPLAY_DATA_RESET	0x0D
GRID_GET_ACTUAL_VOLTAGE	0x0E
RESET_OPTIGRID_AMPLIFIER	0x0F
FOCUS_FIND_HOME	0x11
SET_MOTOR_MANUAL	0x12
GET_MOTOR_MANUAL	0x13
EMBEDDED_CODE_DOWNLOAD	0xFF

All packets are 19 bytes long in both directions.

Byte 0

Byte 1 Command code

Byte 2 Receive number of reply data bytes + 2 (1 command echo, 1 length byte) bytes.

Byte 3 through 18 Are command optional data bytes. Unused bytes are undefined.

Commands Detailed

1. USB_focusFindHome

Send 0x00 0x00 0x00

Receive 0x00 0x00 0x02

// Move the focus servo to the home sensor position and then moves away until the 1st encoder index is encountered.

// At this point the encoder count is set to zero.

2. USB_focusFindLimit

Send 0x00 0x11 0x00

Receive 0x00 0x11 0x02

// Move the focus servo to the limit sensor position.

3. USB_focusZero

Send 0x00 0x01 0x00

Receive 0x00 0x01 0x02

// Set the focus encoder count to zero (at the current position).

HID Report 0 indicator

4. USB_focusGetCount

Send 0x00 0x02 0x02

Receive 0x00 0x02 0x04 R3 R4

// Reports the current focus encoder count.

R3 LS byte

R4 MS byte

5. USB_focusSetCount

Send 0x00 0x03 0x00 S3 S4

Receive 0x00 0x03 0x02

// Move the focus to an encoder count.

S3 LS byte

S4 MS byte

6. USB_focusSetAutoHome

Send 0x00 0x04 0x00 S3

Receive 0x00 0x04 0x02

// Turn on or off auto-home on power up. Valid inputs are AUTO_HOME and MANUAL_HOME.

S3 Auto home value

7. USB_focusGetStatus

Send 0x00 0x05 0x06

Receive 0x00 0x05 0x08 R3 R4 R5 R6 R7 R8

// If the focus position has been achieved, values are IN_POSITION and NOT_IN_POSITION

// If the focus motor is stalled, values are STALLED and NOT_STALLED

// The state of the home switch, values are HOME_SWITCH_ACTIVE and HOME_SWITCH_INACTIVE

// The state of the limit switch, values are LIMIT_SWITCH_ACTIVE and LIMIT_SWITCH_INACTIVE

// If Auto-home is enabled/disabled, values are AUTO_HOME and MANUAL_HOME

R3 Initialized state

R4 Position status

R5 Stalled status

R6 Home switch status

R7 Limit switch status

R8 Auto home status

R9 Piezo cable error (0 indicates ok)

R10 Focus cable error (0 indicates ok)

R11 External DAC select

8. USB_gridGetVoltage

Send 0x00 0x06 0x02

// Count is sw limited to -32768 through +32767, however the mechanics may not have this range.

// If the focus servo mechanism has been initialised (homed), values are INITIALISED and NOT_INITIALISED

Receive 0x00 0x06 0x04 R3 R4

// Reports the voltage setting. Range 0-150V.

// Note:- this value is just an echo on the setting set value resolution is 37mV

R3 LS byte

R4 MS byte

9. USB_gridSetVoltage

Send 0x00 0x07 0x00 S3 S4

Receive 0x00 0x07 0x02

// Note:- this value is set by a 12bit D-A converter so the resolution is 37mV

S3 LS byte

S4 MS byte

10. USB_gridSetControl

Send 0x00 0x08 0x01 S3

Receive 0x00 0x08 0x03 R3

S3 Control value

R3 Control value

// Sets the state of the external grid control input, values are

EXTERNAL_GRID_CONTROL=1 and INTERNAL_GRID_CONTROL=0

11. USB_gridGetControl

Send 0x00 0x09 0x01

Receive 0x00 0x09 0x03 R3

R3 Control value

// Reports the state of the external grid control input, values are

EXTERNAL_GRID_CONTROL=1 and INTERNAL_GRID_CONTROL=0

12. USB_getStoredData

Send 0x00 0x0A 0x07

Receive 0x00 0x0A 0x09 R3 R4 R5 R6 R7 R8 R9

// Reports various data stored in EEPROM:-

// The unit serial number, range 0- .

// The software version number, range 0-255.

// The hardware version number, range 0-255.

// The day of manufacture, range 1-31.

// The month of manufacture, range 1-12.

// The year of manufacture, range 2000-2255.

R3 LS byte Serial number

R4 MS byte Serial number

R5 SW version

R6 HW version

R7 Manufacture date day

R8 Manufacture date month

R9 Manufacture date year since 2000

13. USB_SetStoredData

Send 0x00 0x0B 0x00 S3 S4 S5 S6 S7 S8 S9

Receive 0x00 0x0B 0x02

// Sets various data stored in EEPROM:-
// The unit serial number, range 0-65535 .
// The software version number, range 0-255.
// The hardware version number, range 0-255.
// The day of manufacture, range 1-31.
// The month of manufacture, range 1-12.
// The year of manufacture, range 2000-2255.

S3 LS byte Serial number

S4 MS byte Serial number

S5 SW version

S6 HW version

S7 Manufacture date day

S8 Manufacture date month

S9 Manufacture date year since 2000

14. USB_LCDdisplayErrorMessage

Send 0x00 0x0C 0x00 S3 S4 S5 S6 S7 S8 S9 S10 S11 S12 S13 S14 S15 S16 S17 S18

Receive 0x00 0x0C 0x02

// Displays an error message on the LCD display line 1.
// This data is not retained through a power cycle.
// Any normal operation LCD messages generated by the embedded software these are cleared.
// Maximum string length 16 characters.
// Characters displayed are limited by the type of LCD display
// string must be null terminated if less than 16 bytes

15. USB_LCDdisplayErrorMessage2

Send 0x00 0x10 0x00 S3 S4 S5 S6 S7 S8 S9 S10 S11 S12 S13 S14 S15 S16 S17 S18

Receive 0x00 0x10 0x02

// Displays an error message on the LCD display line 2.
// This data is not retained through a power cycle.
// Any normal operation LCD messages generated by the embedded software these are cleared.
// Maximum string length 16 characters.
// Characters displayed are limited by the type of LCD display
// string must be null terminated if less than 16 bytes

16. USB_LCDdisplayclearErrorMessage

Send 0x00 0x0D 0x00

Receive 0x00 0x0D 0x02

// clears the error message and restores the normal operation LCD display

17. USB_gridGetActualVoltage

Send 0x00 0x0E 0x02

Receive 0x00 0x0E 0x04 R3 R4

// Note:- this value is read by a 10bit A-D converter so the resolution is 150mV

// This is multiplied to 12-bit in the target

R3 LS byte

R4 MS byte

// Reports the voltage applied to the piezo actuator. Range 0-150V.

18. USB_resetTarget

Send 0x00 0x0F 0x00

Receive 0x00 0x0F 0x02

// This function is used to force the target to re-boot.

19. USB_setMotorManual

Send 0x00 0x12 0x00 S3

Receive 0x00 0x12 0x02

// Sets the Motor Control Mode

S3 Control value

S3 = 1 for Motorized.

S3 = 0 for Manual.

20. USB_getMotorManual

Send 0x00 0x13 0x01

Receive 0x00 0x13 0x03 R3

// Gets the Motor Control Mode

R3 Control value

R3 = 1 for Motorized.

R3 = 0 for Manual.



21. USB_enterCodedownloadMode

Send 0x00 0xFF 0x00

Receive 0x00 0xFF 0x02

// This function is used to force the target into download mode.

// The target will set a flag and then re-boot, the boot code will test this flag, clear it and enter download mode.

// The Microchip download utility can then be used.

// To exit the download without an upgrade the target should be power cycled.

// The code upgrade will use Microchip's "bootloader". See DS51526A page 24.