GPUSB Software Application Developers Guide

Shoestring Astronomy – Doug Anderson
11-Oct-2005

INTRODUCTION

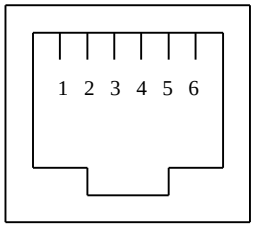
The GPUSB is a USB device intended for driving ST-4 compatible autoguide ports as found on many telescope mounts. The GPUSB is a low-speed USB device of the Human Interface Device (HID) class. The GPUSB utilizes Interrupt Transfers to guarantee no more than 10 millisecond latency for timing critical applications. The benefit of the HID class is that the USB drivers have already been written and are included with all versions of the Windows operating system that support USB devices (Win98SE, Win2000, WinXP). Thus, user installation of the device is simple. Shoestring Astronomy provides the third-party software developer with a dynamic link library (DLL) that further simplifies communication to the GPUSB.

ELECTRICAL INTERFACES

The GPUSB has two connectors, and a bi-color LED.

USB Connector – This is an attached cable (a requirement of the USB-IF for low-speed USB devices) with a USB Type A male connector at the end. This connector will plug into any PC USB port or USB hub port.

Autoguide port connector – This is an RJ-12 modular jack. It is pinned out as follows:



Pin 1 – No connection
Pin 2 – Common
Pin 3 – RA+
Pin 4 – Dec+
Pin 5 – Dec-
Pin 6 – RA-

The jack is driven by optoisolators so that the telescope mount can be electrically isolated from the PC and GPUSB to help eliminate ground loops that can cause electrical interference. The optoisolators function similar to relays. When an output is not asserted, the optoisolator will be in a high impedance state. When the output is asserted, the output will be in a low impedance state to the common terminal, capable of driving loads up to 30mA.

The function calls supplied with this developers kit allow any combination of direction control lines to be turned on. However, in practice, only one RA line and one Dec line should be turned on at a time. The result of turning both RA or Dec lines on simultaneously is indeterminate and varies considerable amongst mount manufacturers. Furthermore, because of variations in mounts and cabling, the names of these direction control lines might not be the actual direction the mount moves in. Therefore, the application software should be written to allow the user to change the mapping of these control lines.

There is a bi-color LED on the GPUSB that is completely under software control. Thus, the application developer is at full liberty to use it as is seen fit.

When the GPUSB is first plugged into a USB port, it initializes itself. The power-up state is LED on and red, and all direction control lines are deasserted.

INCLUDE AND LIBRARY FILES

Shoestring Astronomy provides a development kit with several files you will need for your application development. The DLL file is ShoestringGPUSB_DLL.dll. To support C++ development, a header and library file, ShoestringGPUSB_DLL.h and ShoestringGPUSB_DLL.lib, are included. For Visual Basic support, a module file ShoestringGPUSB_DLL.bas is included.

FUNCTION CALLS

The DLL includes these functions.

boolean GPUSB_Open()
Opens the GPUSB if it is present, and prepares it for use. The most likely cause for unsuccessful initialization is that the GPUSB has not been plugged into the PC.
function returns:

TRUE	Operation successful
FALSE	Operation unsuccessful

boolean GPUSB_Close()
Closes the GPUSB after use is complete. The most likely cause for unsuccessful closure is that the GPUSB has been unplugged from the PC.
function returns:

TRUE	Operation successful
FALSE	Operation unsuccessful

boolean GPUSB_Reset()
Resets GPUSB to its power-up default state. The most likely cause for unsuccessful closure is that the GPUSB has been unplugged from the PC.
function returns:

TRUE	Operation successful
FALSE	Operation unsuccessful

boolean GPUSB_RAMAssert ()
Asserts the RA- control line on the autoguide connector. The most likely cause for failure is that the GPUSB has been unplugged from the PC.
function returns:

TRUE	Operation successful
FALSE	Operation unsuccessful

boolean GPUSB_RAMDeassert ()
Deasserts the RA- control line on the autoguide connector. The most likely cause for failure is that the GPUSB has been unplugged from the PC.
function returns:

TRUE	Operation successful
FALSE	Operation unsuccessful

boolean GPUSB_RAPAssert ()
Asserts the RA+ control line on the autoguide connector. The most likely cause for failure is that the GPUSB has been unplugged from the PC.
function returns:

TRUE	Operation successful
FALSE	Operation unsuccessful

boolean GPUSB_RAPDeassert ()
Deasserts the RA+ control line on the autoguide connector. The most likely cause for failure is that the GPUSB has been unplugged from the PC.
function returns:

TRUE	Operation successful
FALSE	Operation unsuccessful

boolean GPUSB_DecMAssert ()
Asserts the Dec- control line on the autoguide connector. The most likely cause for failure is that the GPUSB has been unplugged from the PC.
function returns:

TRUE	Operation successful
FALSE	Operation unsuccessful

boolean GPUSB_DecMDeassert ()
Deasserts the Dec- control line on the autoguide connector. The most likely cause for failure is that the GPUSB has been unplugged from the PC.
function returns:

TRUE	Operation successful
FALSE	Operation unsuccessful

boolean GPUSB_DecPAssert ()
Asserts the Dec+ control line on the autoguide connector. The most likely cause for failure is that the GPUSB has been unplugged from the PC.
function returns:

TRUE	Operation successful
FALSE	Operation unsuccessful

boolean GPUSB_DecPDeassert ()
Deasserts the Dec+ control line on the autoguide connector. The most likely cause for failure is that the GPUSB has been unplugged from the PC.
function returns:
TRUE Operation successful
FALSE Operation unsuccessful

boolean GPUSB_AllDirDeassert ()
Deasserts all four direction control lines on the autoguide connector. The most likely cause for failure is that the GPUSB has been unplugged from the PC.
function returns:
TRUE Operation successful
FALSE Operation unsuccessful
Deasserts all four direction control lines on the autoguide connector. The most likely cause for failure is that the GPUSB has been unplugged from the PC.

boolean GPUSB_LEDOn()
Energizes the LED with the current color. The most likely cause for failure is that the GPUSB has been unplugged from the PC.
function returns:
TRUE Operation successful
FALSE Operation unsuccessful

boolean GPUSB_LEDOff()
De-energizes the LED. The most likely cause for failure is that the GPUSB has been unplugged from the PC.
function returns:
TRUE Operation successful
FALSE Operation unsuccessful

boolean GPUSB_LEDRed()
Sets the current LED color as red. The most likely cause for failure is that the GPUSB has been unplugged from the PC.
function returns:
TRUE Operation successful
FALSE Operation unsuccessful

boolean GPUSB_LEDGreen()
Sets the current LED color as green. The most likely cause for failure is that the GPUSB has been unplugged from the PC.
function returns:
TRUE Operation successful
FALSE Operation unsuccessful

boolean GPUSB_SetAll(int ram, int rap, int decm, int decp, int led)
Allows shutter, focus, and LED states to all be set simulataneously. The most likely cause for failure is that the GPUSB has been unplugged from the PC.
function returns:
TRUE Operation successful
FALSE Operation unsuccessful
ram can be set to:
GPUSB_RAM_ASSERTED
GPUSB_RAM_DEASSERTED
rap can be set to:
GPUSB_RAP_ASSERTED
GPUSB_RAP_DEASSERTED
decm can be set to:
GPUSB_DECM_ASSERTED
GPUSB_DECM_DEASSERTED
decp can be set to:
GPUSB_DECP_ASSERTED
GPUSB_DECP_DEASSERTED
led can be set to:
GPUSB_LED_ON_RED
GPUSB_LED_ON_GREEN
GPUSB_LED_OFF_RED
GPUSB_LED_OFF_GREEN

boolean GPUSB_RAMStatus(int *status)
Checks the current status of the RA- control line. The most likely cause for failure is that the GPUSB has been unplugged from the PC. This function may take up to 300 milliseconds to execute.
function returns:
TRUE Operation successful
FALSE Operation unsuccessful
status returns:
GPUSB_RAM_ASSERTED
GPUSB_RAM_DEASSERTED

boolean GPUSB_RAPStatus(int *status)
Checks the current status of the RA+ control line. The most likely cause for failure is that the GPUSB has been unplugged from the PC. This function may take up to 300 milliseconds to execute.
function returns:
TRUE Operation successful
FALSE Operation unsuccessful
status returns:
GPUSB_RAP_ASSERTED
GPUSB_RAP_DEASSERTED

boolean GPUSB_DecMStatus(int *status)
Checks the current status of the Dec- control line. The most likely cause for failure is that the GPUSB has been unplugged from the PC. This function may take up to 300 milliseconds to execute.
function returns:

TRUE	Operation successful
FALSE	Operation unsuccessful

status returns:
GPUSB_DECM_ASSERTED
GPUSB_DECM_DEASSERTED

boolean GPUSB_DecPStatus(int *status)
Checks the current status of the Dec+ control line. The most likely cause for failure is that the GPUSB has been unplugged from the PC. This function may take up to 300 milliseconds to execute.
function returns:

TRUE	Operation successful
FALSE	Operation unsuccessful

status returns:
GPUSB_DECP_ASSERTED
GPUSB_DECP_DEASSERTED

boolean GPUSB_LEDStatus(int *status)
Checks the current status of the LED. The most likely cause for failure is that the GPUSB has been unplugged from the PC. This function may take up to 300 milliseconds to execute.
function returns:

TRUE	Operation successful
FALSE	Operation unsuccessful

status returns:
GPUSB_LED_ON_RED
GPUSB_LED_ON_GREEN
GPUSB_LED_OFF_RED
GPUSB_LED_OFF_GREEN

boolean GPUSB_Status(int *ram_status, int *rap_status, int *decm_status, int *decp_status, int *led_status,)

Checks the current status of all control lines simultaneously. The most likely cause for failure is that the GPUSB has been unplugged from the PC. This function may take up to 300 milliseconds to execute.

function returns:

TRUE	Operation successful
FALSE	Operation unsuccessful

ram_status returns:
GPUSB_RAM_ASSERTED
GPUSB_RAM_DEASSERTED

rap_status returns:
GPUSB_RAP_ASSERTED
GPUSB_RAP_DEASSERTED

decm_status returns:
GPUSB_DECM_ASSERTED
GPUSB_DECM_DEASSERTED

decp_status returns:
GPUSB_DECP_ASSERTED
GPUSB_DECP_DEASSERTED

led_status returns:
GPUSB_LED_ON_RED
GPUSB_LED_ON_GREEN
GPUSB_LED_OFF_RED
GPUSB_LED_OFF_GREEN