Stellaris USB Host Joystick

Contents

USB Joystick with Stellaris

This is a sample project showing how to use standard HID Joystick with the Stellaris Evalbot (http://www.ti.com/tool/ek-evalbot).

Prerequisites

HID standard allow designing devices with different amount of analog and digital(buttons) inputs. Therefore every HID device will have different report frame that includes all specific inputs. Keyboard and Mouse have fixed report frame size (for example mouse report always consist at least 3 byes: byte 0 with active buttons, byte 1 with x movement and byte 2 with y movement). As a result keyboard and mouse are easier to handle. Different joystick will have different report frames. There are 2 ways for solving this issue:

1. Reading HID Report Descriptor
2. Writing code for specific joystick

In first case host device is "learning" how report packet look-like using Report Descriptor that host can request after basic USB enumeration. This is how your PC acts. This way is complicated and require detailed study of USB HID documentation (http://www.usb.org/developers/hidpage/)

In second case we are omitting Report Descriptor parsing and assuming that connected device has known report packet. Programmer must know what bits are responsible for what function by reverse engineering or manufacturer documentation.

In this project report structure was reverse engineered using PC USB analyser with software HID parser.

Logitech Extreme 3D Pro

http://www.logitech.com/en-gb/product/extreme-3d-pro?crid=719

- idVendor: 0x046D
- idProduct: 0xC215

Report descriptor (packet) structure		
Byte.Bit	**Function**	**Description**
0.7	X7	analog value X LSB
0.6	X6	
0.5	X5	
0.4	X4	
0.3	X3	
0.2	X2	
0.1	X1	
0.0	X0	
1.7	Y5	analog value Y LSB
1.6	Y4	
1.5	Y3	
1.4	Y2	
1.3	Y1	
1.2	Y0	
1.1	X9	analog value X MSB
1.0	X8	
2[4:7]	H[1:4]	hatswitch buttons
2.3	Y9	analog value Y MSB
2.2	Y8	
2.1	Y7	
2.0	Y6	
3[0:7]	Rz	analog value yaw (rotation)
4[0:7]	B[1:8]	Buttons 1 to 8
5[0:7]	T	analog throttle
6[0:7]	B[9:12]	Buttons 9 to 12

Step 1 - Mouse demo

Mouse and keyboard are most common USB HID (Human Interface Device) devices. Joystick is also HID class device and HID driver from StellarisWare is a good starting point for custom Joystick driver.

There are no USB examples in StellarisWare for Evalbot but we can modify example for board with similar chip.

1. Download most recent StellarisWare release (http://www.ti.com/lsds/ti/microcontroller/arm_stellaris/code_examples.page)
2. Below you will find sample Evalbot HID mouse host project based on `...\StellarisWare\boards\ek-lm3s9d92\usb_host_mouse`
   File:Evalbot Mouse.zip
3. Build the project. It should compile without any errors or warnings. Resolve any issues with the paths (ex. StellarisWare directory)

If you connect Evalbot to the PC you should be able to open terminal (115200 8N1, DTR on). If you connect mouse to the Evalbot and move it you will see following sequence:

```
Host Mode.

Event Connected

  ^ Mouse Connected

Pos: 1, 0  Buttons: 000
Pos: 11, 0  Buttons: 000
Pos: 11, -3  Buttons: 000
[...]
```

Additionally LEDs on Ethernet socket will toggle on mouse buttons press.

Evalbot can be powered using either batteries or USB device port.

Step 2

With working mouse project user can modify sub-driver (usbhhidmouse.c) to get the joystick report data.

1. Copy content of linked resource usbhhidmouse.c into new file usbhhidjoy.c.
2. Repeat for header file.
3. Replace word mouse to joy in both files.
4. In usbhhidjoy.c change `USBHMS_REPORT_SIZE` value to 7. **This is a hardware specific value.**
5. In USBHJoyOpen function replace `USBH_HID_DEV_MOUSE` to `USBH_HID_DEV_NONE`. Label is a bit misleading and should be `USBH_HID_DEV_OTHER`. HID Joysticks will respond with HID subclass value of 0 instead of 2 (mouse) or 1 (keyboard).
6. Disable boot protocol (simple protocol for mouse and keyboard support in PC BIOS) support by setting boot value to 0 in function `USBHHIDSetProtocol(pUSBHJoy->ulJoyInstance, 0);`. The function is located in `USBHjoyInit`.

7. Write UpdateJoyState function. **This function is hardware specific and will most likely not work with any other joystick.** User needs to understand report data of his/her joystick and write appropriate "decoding" function.
8. In main.c replace `g_ulMouseInstance` with `g_ulJoyInstance` and `USBHMouseOpen` with `USBHJoyOpen`.
9. In main.c replace `USBHMouseInit(g_ulMouseInstance);` with `USBHJoyInit(g_ulJoyInstance);`

Sample project: File:Evalbot Joystick.zip

Additional considerations

- Demo project does not check USB device VID and PID. This should be done to avoid wrong decoding of unsupported joystick report data.
- This demo does not support events like mouse demo does.

References

- http://en.wikipedia.org/wiki/USB_human_interface_device_class
- http://www.usb.org/developers/hidpage/
- http://www.frank-zhao.com/cache/hid_tutorial_1.php

{{

1. switchcategory:MultiCore=
   - For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum
   - For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum

Please post only comments related to the article **Stellaris USB Host Joystick** here.

Keystone=

- For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum
- For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum

Please post only comments related to the article **Stellaris USB Host Joystick** here.

C2000=*For technical support on the C2000 please post your questions on The C2000 Forum. Please post only comments about the article **Stellaris USB Host Joystick** here.*

DaVinci=*For technical support on DaVincoplease post your questions on The DaVinci Forum. Please post only comments about the article **Stellaris USB Host Joystick** here.*

MSP430=*For technical support on MSP430 please post your questions on The MSP430 Forum. Please post only comments about the article **Stellaris USB Host Joystick** here.*

OMAP35x=*For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article **Stellaris USB Host Joystick** here.*

OMAPL1=*For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article **Stellaris USB Host Joystick** here.*

MAVRK=*For technical support on MAVRK please post your questions on The MAVRK Toolbox Forum. Please post only comments about the article **Stellaris USB Host Joystick** here.*

*For technical su please post you questions at http://e2e.ti.com Please post onl comments abou article **Stellaris Host Joystick***

}}

Links

Amplifiers & Linear
Audio
Broadband RF/IF & Digital Radio
Clocks & Timers
Data Converters
DLP & MEMS
High-Reliability
Interface
Logic
Power Management
Processors

- ARM Processors
- Digital Signal Processors (DSP)
- Microcontrollers (MCU)
- OMAP Applications Processors

Switches & Multiplexers
Temperature Sensors & Control ICs
Wireless Connectivity