View Full Version : Problem with Teensy USB HID / Joystick implementation ?

Chris Veigl 11-01-2018, 12:06 PM

I made a special joystick for a boy with a disability using TeensLC and it works nicely on the PC,
however when i attach it to the XBOX adaptive controller
https://www.xbox.com/de-DE/xbox-one/accessories/controllers/xbox-adaptive-controller

the joystick does not work at all. (i tested other devices including gamepads and also USB composite devices - they all worked).

I cloned the device-/configuration- and HID-descriptors of working devices (by modifying the teensyduino source code accordingly)
but that did not solve the issue.

i tracked the problem down into the send-routine of the joystick implementation:

```
int usb_flipjoystick_send(void)
{
uint32_t wait_count=0;
usb_packet_t *tx_packet;

while (1) {
if (!usb_configuration) {
return -2;
}
if (usb_tx_packet_count(FLIPJOYSTICK_ENDPOINT) < TX_PACKET_LIMIT) {
tx_packet = usb_malloc();
if (tx_packet) break;
}
if (++wait_count > TX_TIMEOUT || transmit_previous_timeout) {
transmit_previous_timeout = 1;
return -1;
}
yield();
}

transmit_previous_timeout = 0;
memcpy(tx_packet->buf, usb_flipjoystick_data, 3);
tx_packet->len = 3;
usb_tx(FLIPJOYSTICK_ENDPOINT, tx_packet);

return 0;
}
```

this always runs into the timeout - it seems that usb_tx() does not succeed in sending the packets !

i tried the same with teensy 3.1 and also teensy 2++ (using an AVR MCU) - same problem:
works on PC, does not work on the XBOX adaptive controller !

BTW: the HID joystick implementation in Arduino for the Leonardo/Pro Micro devices works fine...

any ideas what could be wrong or what we could try to solve the issue ?

thanks,
chris

gdsports 11-04-2018, 01:30 AM

See my reply on this thread about the MSF-XINPUT library which might help.

https://forum.pjrc.com/threads/53786-Xbox-Adaptive-Controller-with-Teensy-HID-Joystick

Chris Veigl 11-04-2018, 10:16 AM

Thank you for your reply!

Unfortunately, i have already tried all suggested strategies (implement single joystick device, mimick working

joystick HID descriptors, implement XINPUT device)
- and all of them failed.

I think that the problem is a small difference in how the USB interrupts are handled / the usb_tx() function is implemented in the teensyduino code base ...

steve.mcgie 11-06-2018, 04:41 PM

GD, thanks for reviving this thread!

Modifying the drivers is unfortunately a bit above my level, if you know what I mean. I trust the XInput author over myself on this, for sure.

Unfortunately, I couldn't get XInput working either. I couldn't get "XInput" to show up in the "USB Type" menu, even after copying over all the modified Teensyduino files. And in fact, doing so briefly bricked Teensyduino so that all of the options other than "Board" were greyed out. If anyone reading this tries installing XInput, be sure to back up the originals of those files!

I suspect that this may have something to do with me using a significantly older version (IDE 1.0.6, Teensyduino 1.35). But I've broken some of my other code by updating the IDE before, so I'm hesitant to try that.

In any case, Chris' linked thread doesn't make it sound too optimistic, even if I did get XInput working as intended.

Thanks both of you for your help, but in the end, it sounds like the Adaptive Controller just doesn't like to play nicely with Teensy. Which is a real shame. If there's any device that Microsoft should make open-ended for the DIY community, the Adaptive is it.

Chris Veigl 11-06-2018, 05:05 PM

Yes - maybe a conflict caused by the older Teensyduino version!
backup of the Teensyduino files before modifying is a good advice indeed :)
I got the XINPUT USB device code (and menu selection) working nicely, and the USB device appears and works on the PC,
but (as said) no luck on the XBox Adaptive controller also with an XINPUT device implemented using Teensyduino's USB routines ....

it sounds like the Adaptive Controller just doesn't like to play nicely with Teensy. Which is a real shame. If there's any device that Microsoft should make open-ended for the DIY community, the Adaptive is it.

I suspect a small issue so that either the Teensyduino's USB device implementation - or the XBOX adaptive controller's USB host implementation - behaves slightly different from the USB standard. Anyhow: all other USB joysticks and game controllers I tried worked with the adaptive controller - so I suppose the problem _could_ be fixed on the Teensyduino side...

gdsports 11-07-2018, 01:17 AM

I now have an XAC but no USB joysticks. I can confirm the following do not work: MSF_XINPUT, teensyLC joystick, and arduinojoysticklibrary on SAMD. I will try the arduino joystick library on a pro micro 32u4.

I hacked the teensy code to remove the teensy serial port so the teensy joystick has only 1 interface but this still does not work. Could be slight differences in the HID report descriptor.

gdsports 11-07-2018, 05:14 AM

The Teensy LC and probably Teensy 3 work on the Xbox Adaptive Controller (XAC) using the modified Teensyduino 1.44 files are in the zip file. The changes creates a USB Type Joystick which does not include keyboard, mouse, or serial. Auto upload does not work but upload can be manually started by clicking the Teensy program/reset button.

The example program Teensy | USB Joystick | Complete works but change all references from Serial to Serial1. This program is handy because it sends axes movement messages without having to connect hardware joysticks. Very handy on a crowded desk. The XAC is larger than I expected.

I am still not sure why the triple interface Keyboard, Mouse, Joystick USB Type fails.

15107

Chris Veigl 11-07-2018, 01:39 PM

The Teensy LC and probably Teensy 3 work on the Xbox Adaptive Controller (XAC) using the modified Teensyduino 1.44 files are in the zip file. The changes creates a USB Type Joystick which does not include keyboard, mouse, or serial.
15107

Wow - very cool! This really amazes me because

1) i replicated the configuration-, device- and HID-report descriptors from a working device (which is a single joystick, no serial port) without success,
2) another mouse/keyboard/joystick composite device works fine with the XAD - so this can't be the reason ....

however, in the zip you provided, the usb_desc.c file is identical with the original version of teensyduino 1.44
- could it be that you put the wrong file into that .zip ?

many thanks + cheers!
chris

gdsports 11-07-2018, 07:20 PM

Oops, that should have been usb_desc.h, not usb_desc.c. Attached is an update zip file.
15108

Chris Veigl 11-08-2018, 03:08 PM

Hi GD!
i can confirm that the Joystick-only HID version works with the XBox Adaptive Controller! how nice!
thanks for your help :)

i am still looking for the reason why the composite device with mouse/keyboard/joystick does not work.
in fact, for my application i need the serial CDC device also - so the challenge continues ... but the working joystick is a huge step forward !

best regards,
chris

gdsports 11-09-2018, 10:32 PM

Chris, I am glad to hear the new Joystick working out.

I sketched out an IMU to joystick program for a head, hand, or finger controlled joystick. I have to step away from XAC for a while so I am posting the code in case someone wants to give it a try.

```
/*
* Proof of concept head controlled joystick. Not tested with gameplay.
*
* Read quaternions from the BNO055, convert to heading, pitch, roll. Then
* convert roll and pitch to USB joystick X,Y to Xbox Adaptive Controller.
* Should work with other BNO055 boards. tindie.com has dime size BNO055
* boards. Could be mounted on a hand or finger instead of head.
* Set USB Type: Joystick

Connections to Adafruit BNO055 breakout board and Teensy LC
===========
Connect SCL to 19 SCL0
Connect SDA to 18 SDA0
Connect Vin to 3V
Connect GND to GND

TODO Do not send joystick USB data if x and y are not changing. Keep track
of last x and last y. See if send_now() checks for no change. If so,
the sketch does not have to worry about this.

TODO Needs sensitivy adjust. Currently large IMU movement, -180..180
degrees, is required to get full range of joystick movement. Probably should
default to -15..15 or something. Use 2 potentiometers connected to
analog inputs for X and Y sensitivity adjustment.

TODO Try capacitive touch inputs for zero force "buttons". Adafruit sells
conductive fabric strips as a better looking alternative to copper tape or
wire.

Public domain code based on Adafruit and PJRC example programs.
*/

#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BNO055.h>
#include <utility/imumaths.h>
```

```
#include <elapsedMillis.h>

#define DEBUG_IMU (0)

/* Set the delay between fresh samples. The BNO055 produces at most 100
* updates per second. */
#define BNO055_SAMPLERATE_DELAY_MS (10)

Adafruit_BNO055 bno = Adafruit_BNO055();

elapsedMillis imuElapsed;

void setup(void)
{
Serial1.begin(115200);
Serial1.println(F("\nOrientation Sensor Raw Data Test")); Serial1.println();

// configure the joystick to manual send mode. This gives precise
// control over when the computer receives updates, but it does
// require you to manually call Joystick.send_now().
Joystick.useManualSend(true);

Serial1.println("Begin Complete Joystick Test");
/* Initialise the sensor */
if (!bno.begin())
{
/* There was a problem detecting the BNO055 ... check your connections */
Serial1.print(F("BNO055 not found, Check your wiring or I2C ADDR!"));
while (1) delay(1);
}

bno.setExtCrystalUse(true);
/* Display the current temperature */
delay(1000);
int8_t temp = bno.getTemp();
Serial1.print(F("Current Temperature: "));
Serial1.print(temp);
Serial1.println(F(" C"));
Serial1.println();
}

void joy_loop()
{
// Other joystick stuff such as checking buttons.
//
// Because setup configured the Joystick manual send,
// the computer does not see any of the changes yet.
// This send_now() transmits everything all at once.
Joystick.send_now();
}

void imu_loop()
{
// Thanks to gammaburst @ forums.adafruit.com for this conversion.
imu::Quaternion q = bno.getQuat();
// flip BNO/Adafruit quaternion axes to aerospace: x forward, y right, z down
float temp = q.x(); q.x() = q.y(); q.y() = temp; q.z() = -q.z();
q.normalize();

// convert aerospace quaternion to aerospace Euler, because BNO055 Euler data is broken
float heading = 180/M_PI * atan2(q.x()*q.y() + q.w()*q.z(), 0.5 - q.y()*q.y() - q.z()*q.z());
float pitch = 180/M_PI * asin(-2.0 * (q.x()*q.z() - q.w()*q.y()));
float roll = 180/M_PI * atan2(q.w()*q.x() + q.y()*q.z(), 0.5 - q.x()*q.x() - q.y()*q.y());
heading = heading < 0 ? heading+360 : heading;

#if DEBUG_IMU
Serial1.print(F("Heading,Pitch,Roll: "));
Serial1.print(heading); // heading, nose-right is positive
Serial1.print(F(" "));
Serial1.print(pitch); // pitch, nose-up is positive
Serial1.print(F(" "));
Serial1.print(roll); // roll, leftwing-up is positive
Serial1.println(F(""));
#endif

// Convert pitch to joystick y axis, roll to joystick x axis
// Joystick.X, .Y expect values between 0..1023
// Convert -180..180 to 0..1023
int x = ((round(roll) + 180) * 1024) / 360;
int y = ((round(pitch) + 180) * 1024) / 360;
```

```
Joystick.X(x);
Joystick.Y(y);
#if DEBUG_IMU
Serial1.print(F("x,y "));
Serial1.print(x);
Serial1.print(',');
Serial1.println(y);
#endif
}

void loop(void)
{
if (imuElapsed > BNO055_SAMPLERATE_DELAY_MS) {
imuElapsed = 0;
imu_loop();
joy_loop();
}
}
```

gdsports 11-25-2018, 12:41 AM

The Microsoft Xbox Adaptive Controller (XAC) ignores the hat switch on the Logitech Extreme 3D Pro flight stick. The joystick splitter project is one way to solve this problem using Arduino compatible boards.

Joystick X,Y maps to the left thumbstick
Hat 8-way switch maps to the right thumbstick
4 top buttons map to A, B, X, Y
Front trigger maps to right bumper
Side trigger maps to left bumper

https://github.com/gdsports/xac-joystick-splitter

15220
15221

gdsports 12-02-2018, 08:57 AM

The attached file has patches for Teensyduino 1.44 to add USB types Joystick and Joystick + Serial. The Serial is USB CDC ACM. The Microsoft Xbox Adaptive Controller works with both options. The Joystick + Serial option is new.
15273

gdsports 12-02-2018, 11:46 PM

The Microsoft Xbox Adaptive Controller (XAC) ignores all USB HID devices except for joysticks. This project converts USB HID mouse messages into USB HID joystick messages. This allows the use of USB mice, track balls, and some touchpads.

https://github.com/gdsports/xac-mouse2joy

gdsports 12-05-2018, 07:18 PM

The attached file has patches for Teensyduino 1.44 to add USB types Joystick and Joystick + Serial. The Serial is USB CDC ACM. The Microsoft Xbox Adaptive Controller works with both options. The Joystick + Serial option is new.
NEW: The Joystick + Serial option was only available for Teensy 3.6. The option is now available for all Teensy 3.x and LC.

15299

electron 07-20-2019, 08:56 AM

Hello, i am trying the joystick_teensy_20181205 and it's not work witch XAC.
When I put a value in Joystick.X or Joystick.Yy, do I have to write a command to send the value to the USB port or not ?. Could you show me a file? Cordially. Christian

gduck24 08-01-2019, 07:38 PM

This thread seems close to what I want to do so I thought I would ask here. I hacked a gamepad together that uses a teensy and analog joystick to work with my XIM on xbox. The code below works as a joystick with the XIM, but I want to directly plug into the xbox one and play with this as a keyboard and analog. I didnt write this code, only slightly modified it so I dont know what i am doing. I understand the xbox can recognize composite devices, which makes sense because my mouse has keyboard keys and the xbox recognizes the mouse and its

keyboard keys.

Is it possible to modify this code below to make my device work on an xbox but still keep the analog joystick function?

```
void setup()
{
Joystick.useManualSend(true);

pinMode(0, INPUT_PULLUP);
pinMode(1, INPUT_PULLUP);
pinMode(2, INPUT_PULLUP);
pinMode(3, INPUT_PULLUP);
pinMode(4, INPUT_PULLUP);
pinMode(5, INPUT_PULLUP);
pinMode(6, INPUT_PULLUP);
pinMode(7, INPUT_PULLUP);
pinMode(8, INPUT_PULLUP);
pinMode(9, INPUT_PULLUP);
pinMode(10, INPUT_PULLUP);
pinMode(11, INPUT_PULLUP);
pinMode(12, INPUT_PULLUP);
pinMode(24, INPUT_PULLUP);
pinMode(25, INPUT_PULLUP);
pinMode(26, INPUT_PULLUP);
pinMode(27, INPUT_PULLUP);
pinMode(28, INPUT_PULLUP);
}
void loop() {

Joystick.button(1, !digitalRead(2));
Joystick.button(2, !digitalRead(3));
Joystick.button(3, !digitalRead(4));
Joystick.button(4, !digitalRead(5));
Joystick.button(5, !digitalRead(6));
Joystick.button(6, !digitalRead(7));
Joystick.button(7, !digitalRead(8));
Joystick.button(8, !digitalRead(9));
Joystick.button(9, !digitalRead(10));
Joystick.button(10, !digitalRead(11));
Joystick.button(11, !digitalRead(12));
Joystick.button(12, !digitalRead(24));
Joystick.button(13, !digitalRead(25));
Joystick.button(14, !digitalRead(26));
Joystick.button(15, !digitalRead(27));
Joystick.button(16, !digitalRead(28));

Joystick.X(analogRead(0));
Joystick.Y(analogRead(1));

if(touchRead(18) > 3000) Joystick.button(17, 1);
else Joystick.button(17, 0);
if(touchRead(19) > 3000) Joystick.button(18, 1);
else Joystick.button(18, 0);

Joystick.send_now();
}
17104
```