**Steps to update Arduino files for Xinput Fight Stick project:**

Notice on this section <ARDUINO\_INSTALL> means the folder in which Arduino is installed. The default folder in Windows 10 is “C:\Program Files (x86)\Arduino” but user may change it during installation, so in these instructions replace <ARDUINO\_INSTALL> with the path in which Arduino is installed.

1. Open a text editor as administrator and edit <ARDUINO\_INSTALL>\hardware\teensy\avr \boards.txt
2. Search for this line:

*teensyLC.menu.usb.disable=No USB*

1. Above that line insert the next 3 lines:

teensyLC.menu.usb.xinput=[MSF] XINPUT Controller!

teensyLC.menu.usb.xinput.build.usbtype=USB\_XINPUT

teensyLC.menu.usb.xinput.fake\_serial=teensy\_gateway

1. Using windows explorer, open this folder: <ARDUINO\_INSTALL>\hardware\teensy\avr\cores\teensy3
2. Copy to that folder the following files:

MSF-XINPUT-master\MSF\_XINPUT\Teensyduino Files that were edited\hardware\teensy\avr\cores\teensy3\usb\_xinput.c

MSF-XINPUT-master\MSF\_XINPUT\Teensyduino Files that were edited\hardware\teensy\avr\cores\teensy3\usb\_xinput.h

1. Open a text editor as administrator and edit <ARDUINO\_INSTALL>\hardware\teensy\avr\cores\teensy3\usb\_inst.cpp
2. Search for these 3 lines:

*#ifdef USB\_DISABLED*

*usb\_serial\_class Serial;*

*#endif*

1. Above those lines add the next 3 lines:

#ifdef USB\_XINPUT

usb\_xinput\_class XInput;

usb\_serial\_class Serial;

#endif

1. Open a text editor as administrator and edit <ARDUINO\_INSTALL>\hardware\teensy\avr\cores\teensy3\WProgram.h
2. Search for this line:

*#include "usb\_undef.h"*

1. Above that line insert the next line:

#include "usb\_xinput.h"

1. Open a text editor as administrator and edit <ARDUINO\_INSTALL>\hardware\teensy\avr\cores\teensy3\usb\_desc.h
2. Search for these lines 4 lines:

*#endif*

*#ifdef USB\_DESC\_LIST\_DEFINE*

*#if defined(NUM\_ENDPOINTS) && NUM\_ENDPOINTS > 0*

1. Above those lines insert these lines:

#elif defined(USB\_XINPUT)

#define DEVICE\_CLASS 0xFF

#define DEVICE\_SUBCLASS 0xFF

#define DEVICE\_PROTOCOL 0xFF

#define DEVICE\_VERSION 0x0114

#define DEVICE\_ATTRIBUTES 0xA0

#define DEVICE\_POWER 0xFA

#define VENDOR\_ID 0x045e

#define PRODUCT\_ID 0x028e

#define MANUFACTURER\_NAME {'©','M','i','c','r','o','s','o','f','t'}

#define MANUFACTURER\_NAME\_LEN 10

#define PRODUCT\_NAME {'C','o','n','t','r','o','l','l','e','r'}

#define PRODUCT\_NAME\_LEN 10

#define EP0\_SIZE 8

#define NUM\_ENDPOINTS 6

#define NUM\_USB\_BUFFERS 24

#define NUM\_INTERFACE 4

#define XINPUT\_INTERFACE 0

#define XINPUT\_RX\_ENDPOINT 2

#define XINPUT\_RX\_SIZE 8

#define XINPUT\_TX\_ENDPOINT 1

#define XINPUT\_TX\_SIZE 20

#define CONFIG\_DESC\_SIZE 153

#define ENDPOINT1\_CONFIG ENDPOINT\_TRANSIMIT\_ONLY

#define ENDPOINT2\_CONFIG ENDPOINT\_RECEIVE\_ONLY

#define ENDPOINT3\_CONFIG ENDPOINT\_TRANSIMIT\_ONLY

#define ENDPOINT4\_CONFIG ENDPOINT\_RECEIVE\_ONLY

#define ENDPOINT5\_CONFIG ENDPOINT\_TRANSMIT\_AND\_RECEIVE

#define ENDPOINT6\_CONFIG ENDPOINT\_TRANSIMIT\_ONLY

1. Open a text editor as administrator and edit <ARDUINO\_INSTALL>\hardware\teensy\avr\cores\teensy3\usb\_desc.c
2. Search for these lines 4 lines:

*1, // iManufacturer*

*2, // iProduct*

*3, // iSerialNumber*

*1 // bNumConfigurations*

1. Above those lines there is a line in this format (where M, N, P, Q are hexadecimal digits (0 to F) ):

*0xMN, 0xPQ, // bcdDevice*

1. And above the “*0xMN, 0xPQ, // bcdDevice*” line there is a line like this:

*LSB(PRODUCT\_ID), MSB(PRODUCT\_ID), // idProduct*

1. Above the “*0xMN, 0xPQ, // bcdDevice* “ line and below the “*LSB(PRODUCT\_ID), MSB(PRODUCT\_ID), // idProduct*” line insert the following lines:

#ifdef DEVICE\_VERSION // bcdDevice

LSB(DEVICE\_VERSION), MSB(DEVICE\_VERSION),

#else

1. Above the “*1, // iManufacturer* ” line and below the “*0xMN, 0xPQ, // bcdDevice*” line insert the following line:

#endif

1. Now search for this line:

*static uint8\_t config\_descriptor[CONFIG\_DESC\_SIZE] = {*

1. Inside that bracket search for a line like this:

*0xC0, // bmAttributes*

1. Above that line insert the following lines:

#ifdef DEVICE\_ATTRIBUTES

DEVICE\_ATTRIBUTES, // bmAttributes

#else

1. Below that line insert this line:

#endif

1. Inside the same bracket search for a line like this:

*50, // bMaxPower*

1. Above that line insert the following lines:

#ifdef DEVICE\_POWER

DEVICE\_POWER, // bMaxPower

#else

1. Below that line insert this line:

#endif

1. Now search for this line:

*#define CONFIG\_DESC\_SIZE*

1. Above that line insert this line:

#ifndef XINPUT\_INTERFACE

1. Below that line insert this line:

#endif

1. Now search for these lines:

*};*

*// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*// String Descriptors*

*// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

1. Make sure that the closing bracket corresponds to the bracket that was opened on the line “*static uint8\_t config\_descriptor[CONFIG\_DESC\_SIZE] = {* “ If is not please search for the next closing bracket “};” that is below that line.
2. Above the closing bracket line insert the following lines:

#ifdef XINPUT\_INTERFACE

//Interface 0

9, //bLength (length of interface descriptor 9 bytes)

4, //bDescriptorType (4 is interface)

0, //bInterfaceNumber (This is interface 0)

0, //bAlternateSetting (used to select alternate setting. notused)

2, //bNumEndpoints (this interface has 2 endpoints)

0xFF, //bInterfaceClass (Vendor Defined is 255)

0x5D, //bInterfaceSubClass

0x01, //bInterfaceProtocol

0, //iInterface (Index of string descriptor for describing this notused)

//Some sort of common descriptor? I pulled this from Message Analyzer dumps of an actual controller

17,33,0,1,1,37,129,20,0,0,0,0,19,2,8,0,0,

//Endpoint 1 IN

7, //bLength (length of ep1in in descriptor 7 bytes)

5, //bDescriptorType (5 is endpoint)

0x81, //bEndpointAddress (0x81 is IN1)

0x03, //bmAttributes (0x03 is interrupt no synch, usage type data)

0x20, 0x00, //wMaxPacketSize (0x0020 is 1x32 bytes)

4, //bInterval (polling interval in frames 4 frames)

//Endpoint 2 OUT

7, //bLength (length of ep2out in descriptor 7 bytes)

5, //bDescriptorType (5 is endpoint)

0x02, //bEndpointAddress (0x02 is OUT2)

0x03, //bmAttributes (0x03 is interrupt no synch, usage type data)

0x20, 0x00, //wMaxPacketSize (0x0020 is 1x32 bytes)

8, //bInterval (polling interval in frames 8 frames)

//Interface 1

9, //bLength (length of interface descriptor 9 bytes)

4, //bDescriptorType (4 is interface)

1, //bInterfaceNumber (This is interface 1)

0, //bAlternateSetting (used to select alternate setting. notused)

4, //bNumEndpoints (this interface has 4 endpoints)

0xFF, //bInterfaceClass (Vendor Defined is 255)

0x5D, //bInterfaceSubClass (93)

0x03, //bInterfaceProtocol (3)

0, //iInterface (Index of string descriptor for describing this notused)

//A different common descriptor? I pulled this from Message Analyzer dumps of an actual controller

27,33,0,1,1,1,131,64,1,4,32,22,133,0,0,0,0,0,0,22,5,0,0,0,0,0,0,

//Endpoint 3 IN

7, //bLength (length of ep3in descriptor 7 bytes)

5, //bDescriptorType (5 is endpoint)

0x83, //bEndpointAddress (0x83 is IN3)

0x03, //bmAttributes (0x03 is interrupt no synch, usage type data)

0x20, 0x00, //wMaxPacketSize (0x0020 is 1x32 bytes)

2, //bInterval (polling interval in frames 2 frames)

//Endpoint 4 OUT

7, //bLength (length of ep4out descriptor 7 bytes)

5, //bDescriptorType (5 is endpoint)

0x04, //bEndpointAddress (0x04 is OUT4)

0x03, //bmAttributes (0x03 is interrupt no synch, usage type data)

0x20, 0x00, //wMaxPacketSize (0x0020 is 1x32 bytes)

4, //bInterval (polling interval in frames 4 frames)

//Endpoint 5 IN

7, //bLength (length of ep5in descriptor 7 bytes)

5, //bDescriptorType (5 is endpoint)

0x85, //bEndpointAddress (0x85 is IN5)

0x03, //bmAttributes (0x03 is interrupt no synch, usage type data)

0x20, 0x00, //wMaxPacketSize (0x0020 is 1x32 bytes)

64, //bInterval (polling interval in frames 64 frames)

//Endpoint 5 OUT (shares endpoint number with previous)

7, //bLength (length of ep5out descriptor 7 bytes)

5, //bDescriptorType (5 is endpoint)

0x05, //bEndpointAddress (0x05 is OUT5)

0x03, //bmAttributes (0x03 is interrupt no synch, usage type data)

0x20, 0x00, //wMaxPacketSize (0x0020 is 1x32 bytes)

16, //bInterval (polling interval in frames 16 frames)

//Interface 2

9, //bLength (length of interface descriptor 9 bytes)

4, //bDescriptorType (4 is interface)

2, //bInterfaceNumber (This is interface 2)

0, //bAlternateSetting (used to select alternate setting. notused)

1, //bNumEndpoints (this interface has 4 endpoints)

0xFF, //bInterfaceClass (Vendor Defined is 255)

0x5D, //bInterfaceSubClass (93)

0x02, //bInterfaceProtocol (3)

0, //iInterface (Index of string descriptor for describing this notused)

//Common Descriptor. Seems that these come after every interface description?

9,33,0,1,1,34,134,7,0,

//Endpoint 6 IN

7, //bLength (length of ep6in descriptor 7 bytes)

5, //bDescriptorType (5 is endpoint)

0x86, //bEndpointAddress (0x86 is IN6)

0x03, //bmAttributes (0x03 is interrupt no synch, usage type data)

0x20, 0x00, //wMaxPacketSize (0x0020 is 1x32 bytes)

16, //bInterval (polling interval in frames 64 frames)+

//Interface 3

//This is the interface on which all the security handshaking takes place

//We don't use this but it could be used for man-in-the-middle stuff

9, //bLength (length of interface descriptor 9 bytes)

4, //bDescriptorType (4 is interface)

3, //bInterfaceNumber (This is interface 3)

0, //bAlternateSetting (used to select alternate setting. notused)

0, //bNumEndpoints (this interface has 0 endpoints ???)

0xFF, //bInterfaceClass (Vendor Defined is 255)

0xFD, //bInterfaceSubClass (253)

0x13, //bInterfaceProtocol (19)

4, //iInterface (Computer never asks for this, but an x360 would. so include one day?)

//Another interface another Common Descriptor

6,65,0,1,1,3

#endif // XINPUT\_INTERFACE

1. Now search for this line:

*void usb\_init\_serialnumber(void)*

1. Above that line add these lines:

struct usb\_string\_descriptor\_struct usb\_string\_xinput\_security\_descriptor = {

0xB2, 0x03, 0x58, 0x00, 0x62, 0x00, 0x6F, 0x00, 0x78, 0x00, 0x20, 0x00, 0x53, 0x00, 0x65, 0x00,

0x63, 0x00, 0x75, 0x00, 0x72, 0x00, 0x69, 0x00, 0x74, 0x00, 0x79, 0x00, 0x20, 0x00, 0x4D, 0x00,

0x65, 0x00, 0x74, 0x00, 0x68, 0x00, 0x6F, 0x00, 0x64, 0x00, 0x20, 0x00, 0x33, 0x00, 0x2C, 0x00,

0x20, 0x00, 0x56, 0x00, 0x65, 0x00, 0x72, 0x00, 0x73, 0x00, 0x69, 0x00, 0x6F, 0x00, 0x6E, 0x00,

0x20, 0x00, 0x31, 0x00, 0x2E, 0x00, 0x30, 0x00, 0x30, 0x00, 0x2C, 0x00, 0x20, 0x00, 0xA9, 0x00,

0x20, 0x00, 0x32, 0x00, 0x30, 0x00, 0x30, 0x00, 0x35, 0x00, 0x20, 0x00, 0x4D, 0x00, 0x69, 0x00,

0x63, 0x00, 0x72, 0x00, 0x6F, 0x00, 0x73, 0x00, 0x6F, 0x00, 0x66, 0x00, 0x74, 0x00, 0x20, 0x00,

0x43, 0x00, 0x6F, 0x00, 0x72, 0x00, 0x70, 0x00, 0x6F, 0x00, 0x72, 0x00, 0x61, 0x00, 0x74, 0x00,

0x69, 0x00, 0x6F, 0x00, 0x6E, 0x00, 0x2E, 0x00, 0x20, 0x00, 0x41, 0x00, 0x6C, 0x00, 0x6C, 0x00,

0x20, 0x00, 0x72, 0x00, 0x69, 0x00, 0x67, 0x00, 0x68, 0x00, 0x74, 0x00, 0x73, 0x00, 0x20, 0x00,

0x72, 0x00, 0x65, 0x00, 0x73, 0x00, 0x65, 0x00, 0x72, 0x00, 0x76, 0x00, 0x65, 0x00, 0x64, 0x00,

0x2E, 0x00

};

1. Now search for this line:

*{0x0303, 0x0409, (const uint8\_t \*)&usb\_string\_serial\_number, 0},*

1. Below that line insert these lines:

#ifdef XINPUT\_INTERFACE

{0x0304, 0x0409, (const uint8\_t \*)&usb\_string\_xinput\_security\_descriptor, 0},

#endif

1. Open a text editor as administrator and edit <ARDUINO\_INSTALL>\hardware\teensy\avr\cores\teensy3\usb\_serial.h
2. Search for this line:

*#if F\_CPU >= 20000000*

1. Just to the right of the last 0 add this:

&& (defined(USB\_SERIAL) || defined(USB\_SERIAL\_HID))

1. Now search for this line:

*#if (defined(CDC\_STATUS\_INTERFACE) && defined(CDC\_DATA\_INTERFACE)) || defined(USB\_DISABLED)*

1. At the beginning of that line add two slashes “//”, the line should look like this:

//#if (defined(CDC\_STATUS\_INTERFACE) && defined(CDC\_DATA\_INTERFACE)) || defined(USB\_DISABLED)

1. Now search for this line:

*#endif // CDC\_STATUS\_INTERFACE && CDC\_DATA\_INTERFACE*

1. At the beginning of that line add two slashes “//”, the line should look like this:

//#endif // CDC\_STATUS\_INTERFACE && CDC\_DATA\_INTERFACE