

If this is your first visit, be sure to check out the [FAQ](#) by clicking the link above. You may have to [register](#) before you can post: click the register link above to proceed. To start viewing messages, select the forum that you want to visit from the selection below.

How To Build Your Own USB Joystick

Like 15 people like this. Be the first of your friends.

by [FlightSim.Com Staff](#)

Published on 10-28-2004 10:00 AM

How To...Build A DIY USB Joystick

By Mindaugas Milasauskas (28 October 2004)

Introduction

I have started developing this joystick controller inspired by the need of a simple DIY joystick which could overcome the limitations of the gameport and deliver greater accuracy and flexibility in the possible designs. I realized the need for such device because I like to play IL2 Sturmovik. This game has obvious demand for a high quality joystick unless you are happy being dead meat. Most mass market joysticks have lots of flaws. And there are a lot of guys who make joysticks of their own design. Many of these designs are much better than mass production ones. For most of these designs the controller is needed. Using the gameport would ruin all the efforts of high precision mechanics. There are many of these discussions on IL2 game forums. The obvious demand was for USB joystick controller so I pursued in this direction. And here it is...

Investigation

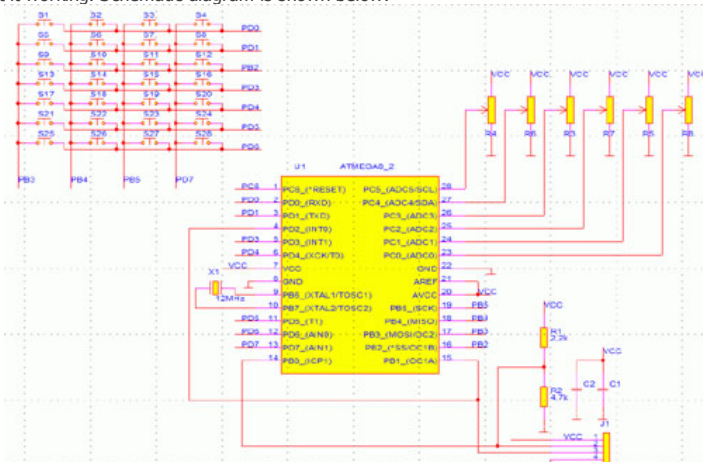
The best way looked to find a suitable USB enabled controller and implement application on it. But after looking for various possibilities it was obvious that it is not easy to get such a device for reasonable price in reasonable time for DIY amateur. The other possibility was to take some generic microcontroller and implement USB protocol on it as well as application level procedures. Looking across the Internet I found Igor Cesko page <http://www.cesko.host.sk/>.

Igor has made good investigation and big work in implementing USB protocol on microcontroller. So I decided to take his design as a basis and implement everything what is needed for joystick on top. I chose ATmega8 as it was cheap and easy to get component for the design. Cheers, Igor! ☺

Next it was time for investigating HID firmware implementation. After that it was also needed to dig a bit deeper in USB specifications as implementation of USB interrupts was not included in Igor's device. After intensive evening work for couple of weeks I managed to get this controller working.

Hardware

Hardware component basis is very minimal for this controller so most of DIY electronics hobbyists even with limited experience could repeat it and build the super-joystick by themselves. The main brain is ATmega8 microcontroller in DIP package with 12 Mhz quartz resonator. Few resistors and capacitors are everything what is needed to get it working. Schematic diagram is shown below:



The connection of controller to joystick sensors and buttons is very simple. In this schematic it is shown with

Tag Cloud

737 777 a2a a319 a320 ade aerosoft afs design airbus aircraft
airline alabeo atc australia beechcraft boeing canadian xpress
carenado cessna cockpit crash curtiss jenny danish airfields x
drzewiecki design flightsoft fly-in flysimware ford tri-motor project
freeware fs2crew fs2004 fsx just flight landing lionheart
creations md-11 mega airport megascenery megascenery earth
night 3d night environment nuvecta p3d payware piper pmdg
prepar3d problem rex rex4 ron blehm saitek sale scenery sketchup
sky song soundworks taburet treeline north america tss turbine sound
studios ultra res cities vatsim vidan design virtual virtual airline
virtual airlines where in the world wilco X-plane x-plane 10

Recent Forum Activity



Captain Sim L-1011

Thread Starter: windqaz

Does anyone know if the Captain Sim L-1011 will work in FS9? Thanks Syd
Last Post By: [HardCub](#) Today, 01:29 PM



Strange problem, autopilot is automatically engaged at the beginning of every flight

Thread Starter: sidfadc

Hi guys, First post woohoo! Anyway I'm a new convert to FSX and I absolutely love it. I'm having the problem as per the title. It wasn't always...
Last Post By: [Iredbarron](#) Today, 01:25 PM



MSI Afterburner

Thread Starter: usb777

Just what will the MSI Afterburner do for my performance?
Last Post By: [usb777](#) Today, 01:07 PM



"senior member" ???

Thread Starter: vflight2

I was perusing some of the forums last week and happened to run across a few signatures that bore the designation, "SENIOR MEMBER". This is a...
Last Post By: [Ptrcam](#) Today, 12:58 PM



Earth Simulations ceased trading?

Thread Starter: aeroone

It seems that Earth Simulations has ceased trading. That's a shock. :(
Last Post By: [TightGit](#) Today, 12:55 PM



Ethiopian WOI Updated Aircraft Fleet?

Thread Starter: AidenMiller

Hey guys, So I'm a WOI (world of ai) user, and Ethiopian Airline is one of my favorite airlines. Does anyone know where I can find updated WOI...
Last Post By: [StringBean](#) Today, 12:53 PM



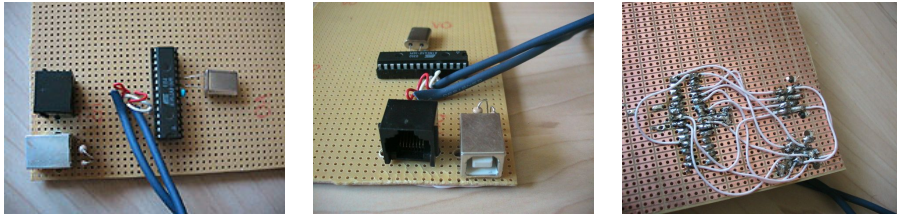
Captain Sim 757 Autothrottles

Thread Starter: wwhteside

You guys ever get the A/T to work on the CS 757. Ive not had any luck with their support system. Mine just go to full and never respond to any...
Last Post By: [PPSFA](#) Today, 12:19 PM

potentiometers but it may be also some other axis position sensors – magnetic, optic and other. I am also working on connecting this controller to my UR Gear modified helmet. I will also show some alternative hardware+software joystick configurations. ADC inputs in this design take the voltages from 0 to 5 volts as full range. But smaller range is possible. Hat switch uses buttons PD7-PD3, PD7-PD4, PD7-PD5, PD7-PD6. Capacitors used here are filtering ones and usually have value of 0.1 μ F but any similar ones should do as well.

The controller has no PCB design ready and you will have to create your own. Or maybe I'll do it at some later time when I have some spare time which I never have. 😊 My piece is made on generic drilled PCB with several wires connecting the components:



One connector is for USB, another is for ISP programming adapter which is shown below. I know it doesn't look very nice but it does its job. I would be very thankful to anyone who could build the PCB and would send one piece to me. 😊 (It's a joke)

Software

All the needed software is in the microcontroller. No additional Windows drivers needed. Thanks to HID class. The development and assembly was done using AVR Studio 4. Most of basic USB protocol handling parts are left from Igor's firmware. Additional routines were added: HID specific descriptor handling, Interrupt In handling, joystick data acquiring from ADCs and buttons and stuffing this data to 8 byte long reports. Some unneeded routines were removed.

The software version published here is for one of possible joystick configuration. This is for 6 axis, 8 position hat switch, 24 buttons joystick:

- 2 x 10 bit ADC inputs for X, Y axis
- 1 x 8 bit ADC input for Throttle axis
- 1 x 8 bit ADC input for Rudder axis
- 1 x 8 bit ADC input for Slider axis
- 1 x 8 bit ADC input for Dial axis
- 1 x 8 position Hat switch (Hat uses 4 button switches)
- 24 buttons

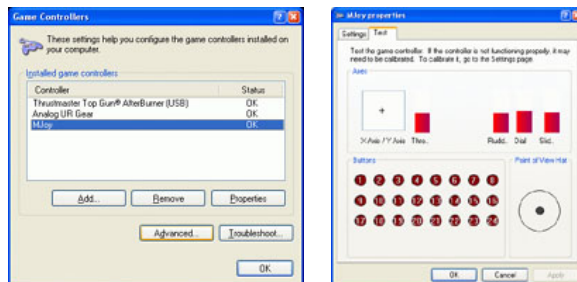
It uses almost all data input resources available on ATmega8 DIP packaged chip. If you have ATmega16 or greater you can have even more axes and buttons. The AVR Studio 4 project file with source code and assembled binary for this controller version is here:

mjoy_v1.1.zip

There may be huge amount of other possible configurations. If you need custom configuration you could build your own. In next section I'll describe how.

Installation

Installation is accomplished by simply plugging in the device into the USB port of the computer. The OS should find the device and install it as a joystick. You should find your USB joystick among gaming devices. It should appear as "MJoy":



It was tested on Windows 2000 and Windows XP computers. This controller does not have automatic calibration function. But it may be implemented in future. For now you will need to calibrate this joystick using standard Windows joystick calibration procedure.

What is needed to build custom joystick configuration

I won't dig deep into Igor's implementation as you can read his documentation and investigate by your own. I will try to describe my part and hope it will help you to build your own joystick configurations. For that you will have to get yourself a bit familiar with AVR microcontroller assembly instructions. Programming in higher level languages is not suitable as the Igor's code contains some time-tight routines. The possible configurations limit is 64 bits for one report which you should use to stuff all the data from your custom controller. It may be 64 buttons console for flight controls either 8 x 8 bit axes analog controls for propeller pitch, flaps and trimmers or anything in between. You choose.

Another possibility is to use several reports with different IDs to send virtually as much information as you wish. The drawback is that the lowest report query interval limit is 10ms. In case you have several reports sent back in cyclic manner the overall device poll rate would decrease. In the implementation presented here there are 2 reports which are sent back cyclically. As a result this joystick poll period of 20ms.

HID USB device Enumeration

All USB devices start operating with Enumeration. After getting enumeration information host operating system knows which drivers to load to be able to use the device. For HID devices Windows have drivers built-in and no special drivers needed. All that is needed is that device return the descriptors which conform to HID firmware specifications.

In addition to minimal generic USB device descriptors additional descriptors are needed for minimal HID device: HIDDescriptor and ReportDescriptor. All these descriptors are at the end of source file.

ReportDescriptor is the main structure telling what kind of joystick it is and what are its inputs. If you want to build your own configuration you will have to familiarize yourself with this structure and understand it. There are specifications and some examples from which you should be able to learn how to get what you want. Please familiarize yourself with HID documentation part about Report descriptor to be able to create meaningful structure. The HID tool has example files of some commonly used descriptors to help you out with this.

In this implementation it has defined that it uses 2 reports with IDs 1 and 2. The size of the report data returned back in each of them is defined by JoystickReport1Size and JoystickReport2Size values. These values state the size of each report.

In this case ReportID1 has 8 bytes. First byte carries report ID. The next 7 bytes carry information about axes and Hatswitch. ReportID2 has 4 – 1 for ID + 3 for 24 buttons. These values are used in ProcJoystickRequest routine which is discussed later.

After editing ReportDescriptor you should update the ReportDescriptorSize value Equal to the size of ReportDescriptor structure in bytes. So pay attention to it when you edit ReportDescriptor. Making an error would result your joystick not to appear in joysticks list. HID test tool is also useful for this calculating descriptor size. It can tell you what is the size of this descriptor when you add all the fields in it. But at the end I got used to recalculate it manually on the fly.

There is also some important field in EndpointDescriptor – polling interval. You can define your own value in range from 10 to 255. It will define the polling frequency. With default 10 milliseconds value controller is polled 100 times in a second. But also remember how many reports you have to calculate your joystick polling period.

If you are building your custom configuration also update the fields VendorUSBID, DeviceUSBID and DeviceVersion. You may also want to change the name of the device string but please leave the copyright credits in VendorStringDescriptor. This is for me and for Igor.

Joystick Data Acquiry and Report Formatting

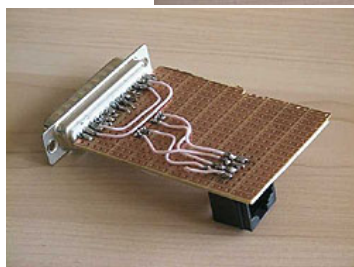
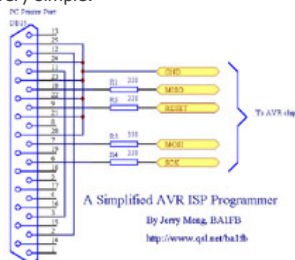
All joystick input data processing is done in ProcJoystickRequest routine. You would also have to modify it to correspond to ReportDescriptor if you want to build custom configuration joystick. This routine is called in spare time between the data polls. The complexity of it might affect the poll time. If it is too complex you might need to increase the polling period as described above. Also it is worth to note that some ADC and button inputs preparation instructions are in reset routine. So if you are changing some pins behavior, look in there. One important parameter there is ADC prescaler division factor. It defines how fast ADC conversion goes. In this design it is 64 which gives 187 kHz make prescaler value smaller to make A-D conversion go faster. But this might decrease the accuracy of ADC. Recommended ATmega guideline is up to 200 kHz for accurate results. In presented version the software reads values from all ADC inputs and reads buttons matrix using all other spare pins of ATmega8. Two branches process Report1 and Report2 which are defined in ReportDescriptor. They are selected in cyclic order. There are procedures which read 10 or 8 bit ADC inputs. The input parameter for them is carried in temp0 which is the number of Port C pin to read. Other procedure reads a row of buttons formed using port D and port B pins. All the results are written into the JoystickBuffer using the structure defined in ReportDescriptor.

Programming The Controller

The controller was programmed using very simple programmer. I like using simple tools for simple tasks 😊.

Programmer Hardware

Hardware is very simple:



It has a bit dirty design but it works for me and I don't think something much more is needed.

Programmer Software

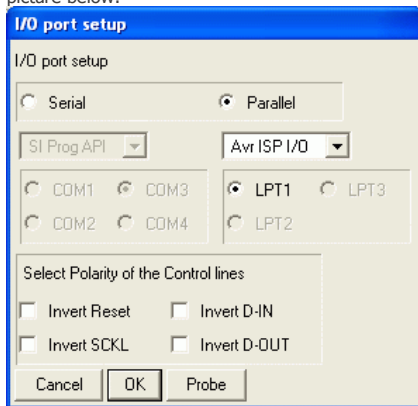
I use PonyProg to program the controller. You can find the latest version on this website:

<http://www.flightsim.com/vbfs/content.php?2112-How-To-Build-Your-Own-USB-Joystick>

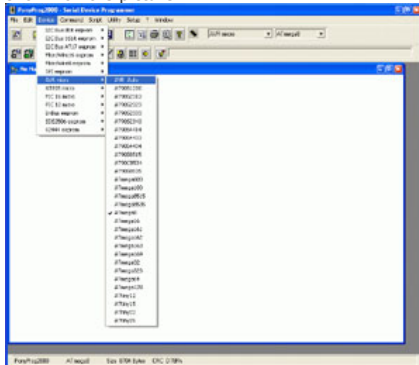
<http://www.lancos.com/prog.html>

Programming Procedure

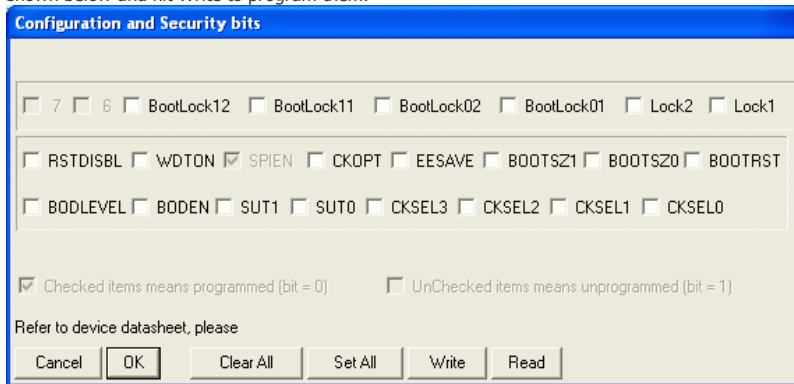
1. Connect the controller to USB port on your computer. This is needed to provide +5V power supply. Connect the programmer to LPT1 port on your computer. Connect the programmer to your controller (I use simple LAN patch cord).
2. Start PonyProg program. On first start it proposes to make calibration to adjust itself to your PC speed. Let it do it. Alternatively use Setup->Calibration from menu.
3. Make setup of programmer interface using Setup->Interface Setup... Make settings according to picture below.



4. Select ATmega8 chip as the device you want to program from menu Device->ATmega8 as shown on the picture.



5. Open mjoy.hex file using menu File->Open Program (FLASH) File.
6. Program the chip using Command->Write Program (FLASH) menu.
7. Program the chip configuration bits using Command->Security and Configuration bits. Set flags as shown below and hit Write to program them.



8. Close PonyProg, disconnect the programmer and replug your new megajoystick to USB port.

Licence and Copyrights

All information in this published document and all source code is free for non commercial use. All presented programs and source code are under GPL licence (for non commercial use).

Copyright © Mindaugas Milasauskas 2004, Lithuania, www.mindaugas.com

Copyright © Ing. Igor Cesko 2003, Slovakia, www.cesko.host.sk, cesko@internet.sk

References:

1. Igor Cesko "USB to RS232" controller
2. USB Documents

3. [USB Specifications](#)
4. [HID Web page](#)
5. [HID Specifications](#)
6. [HID Descriptor Tool](#) - good for creating your own Report Descriptors
7. [USB Sniffer v1.8](#) - good for troubleshooting
8. [Atmel AVR microcontrollers](#) -
9. [AVR Studio 4](#) - good tool for programming and debugging AVR microcntrrollers
10. [Simple AVR ISP programmer hardware](#)
11. [PonyProg](#) - good AVR programming software
12. [MJoy in vkb.sukhoi.ru](#) - this article in Russian

Enjoy and please send your comments to me.

Regards,
[Mindaugas Milasauskas](#)

Visit the MJoy project home page for latest information, include v1.2 software with autocalibration

Categories: [2004](#), [FS2004](#)

-- FlightSim ▼	Select Language ▼	Contact Us FlightSim.Com Archive Privacy Statement Top
--------------------------------	-------------------------------------	--

All times are GMT -4. The time now is 01:36 PM.
Powered by [vBulletin®](#) Version 4.2.2
Copyright © 2015 vBulletin Solutions, Inc. All rights reserved.
Image Gallery provided by [DragonByte Gallery \(Pro\)](#) - vBulletin Mods & Addons Copyright © 2015 DragonByte Technologies Ltd.
Shoutbox provided by [vBShout \(Pro\)](#) - vBulletin Mods & Addons Copyright © 2015 DragonByte Technologies Ltd.
NavTabs provided by [vBNavTabs \(Pro\)](#) - vBulletin Mods & Addons Copyright © 2015 DragonByte Technologies Ltd.
Copyright © 1996 - 2015 FlightSim.Com Inc. All Rights Reserved