

LUFA & Studio5 Beginner's Guide

Part 1: Compiling the VirtualSerial Demo

Introduction

This is a step by step guide to compiling the VirtualSerial demo in the LUFA USB library using Studio5. Please keep in mind that this guide is “a way of doing it”, not “the way of doing it”.

LUFA (Lightweight USB Framework for AVR), written and maintained by Dean Camera, “is an open-source complete USB stack for the USB-enabled Atmel AVR8 and (some of the) AVR32 microcontroller series.” It is available at: <http://www.fourwalledcubicle.com/LUFA.php>.

Studio5 (Atmel's AVR Studio 5) is an IDE (Integrated Development Environment) for AVR microcontrollers that includes the GCC 'C' compiler. It is available from Atmel at: <http://www.atmel.com/tools/ATMELAVRSTUDIO.aspx>

The GCC Compiler Manual can be downloaded from: <http://gcc.gnu.org/onlinedocs/gcc-4.6.3/gcc.pdf>

The GNU C Library Manual can be downloaded from: <http://www.gnu.org/software/libc/manual/pdf/libc.pdf>

The GNU Make Manual can be downloaded from: <http://www.gnu.org/software/make/manual/>

This guide uses Studio5 version 5.1.208 and LUFA-120219.

Caveat Emptor

I am not an expert in AVR, Studio5, or LUFA. In fact, I'm a rank beginner in both Studio5 and LUFA. I'm writing this because being able to compile the demo represents the first rung of the ladder and I hope this guide helps the next person get to that first rung quicker and with less frustration.

No warranty expressed or implied. YMMV.

Before We Start

You should already have downloaded and installed Studio5 as well as downloaded the LUFA zip file and extracted the files. This guide is based on LUFA-120219 from LUFA-120219.zip. If you extracted the files and couldn't find the documentation, that's because it's not included. You need to download the documentation separately.

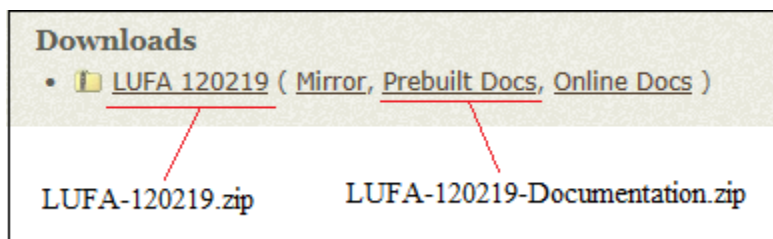


Figure 1: From www.fourwalledcubicle.com/LUFA.php - Scroll to bottom of page.

Once the documentation zip file is downloaded and extracted, navigate to the file index.html in the directory: \documentation\html\. Open the file using a browser and you'll see the main page with a table of contents in the left hand pane.

As shown in figure 2, a Getting Started section is included under the Related Pages group. As shown in figure 3, under Building as a Linkable Library, it is recommended that the library source files be compiled with the project. This is the approach that will be followed in this guide.

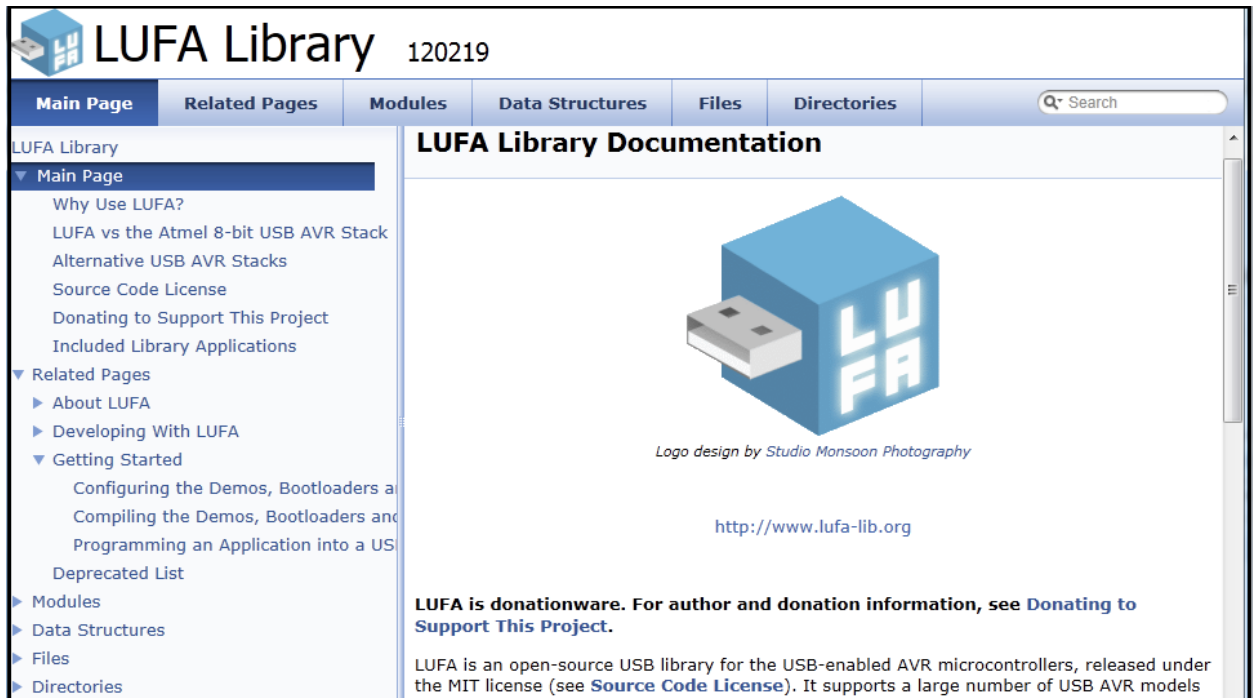


Figure 2: LUFA Documentation via index.html

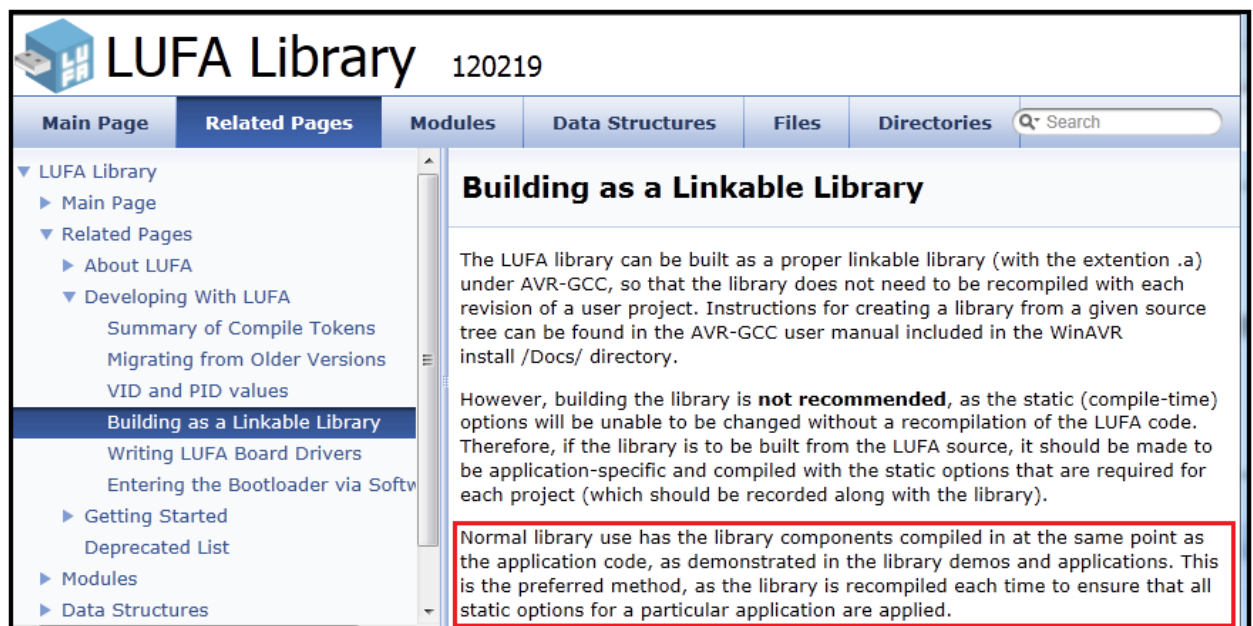


Figure 3: LUFA Documentation - Compile library source files with project.

Step 1: Import the Studio 4 Project VirtualSerial.aps

Use File, Import, AVR Studio 4 Project

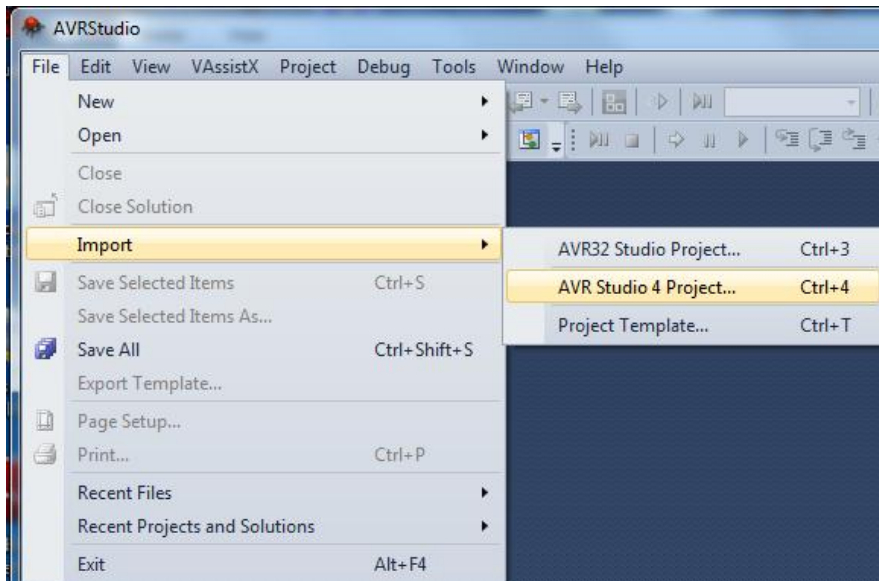


Figure 4: Import an AVR Studio 4 Project

Select the project VirtualSerial.aps located at:

LUFA\LUFA-123219\LUFA-120219\ Demos\ Device\Class Driver\VirtualSerial\

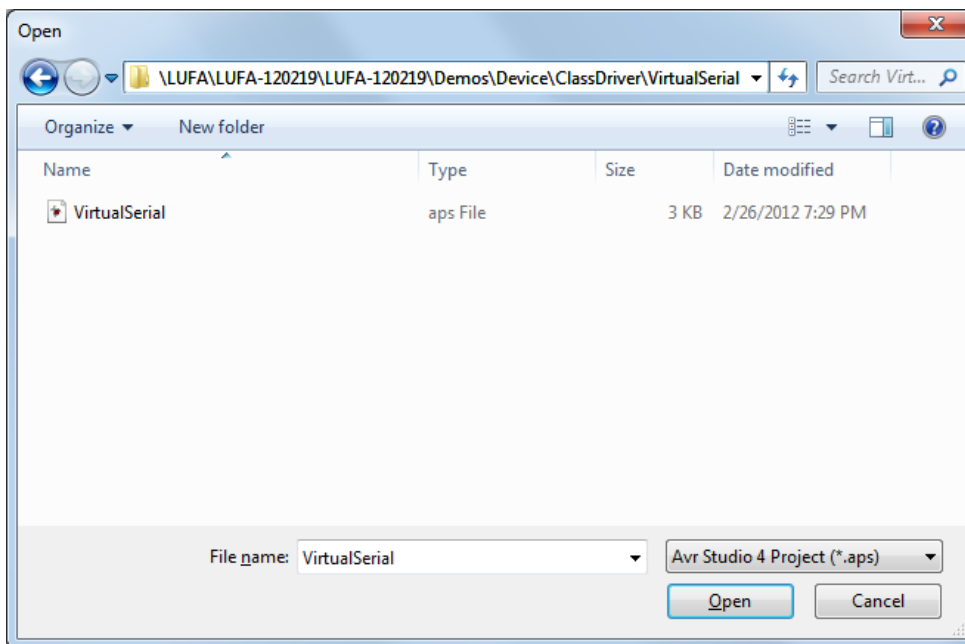


Figure 5: Select the Project File VirtualSerial.aps

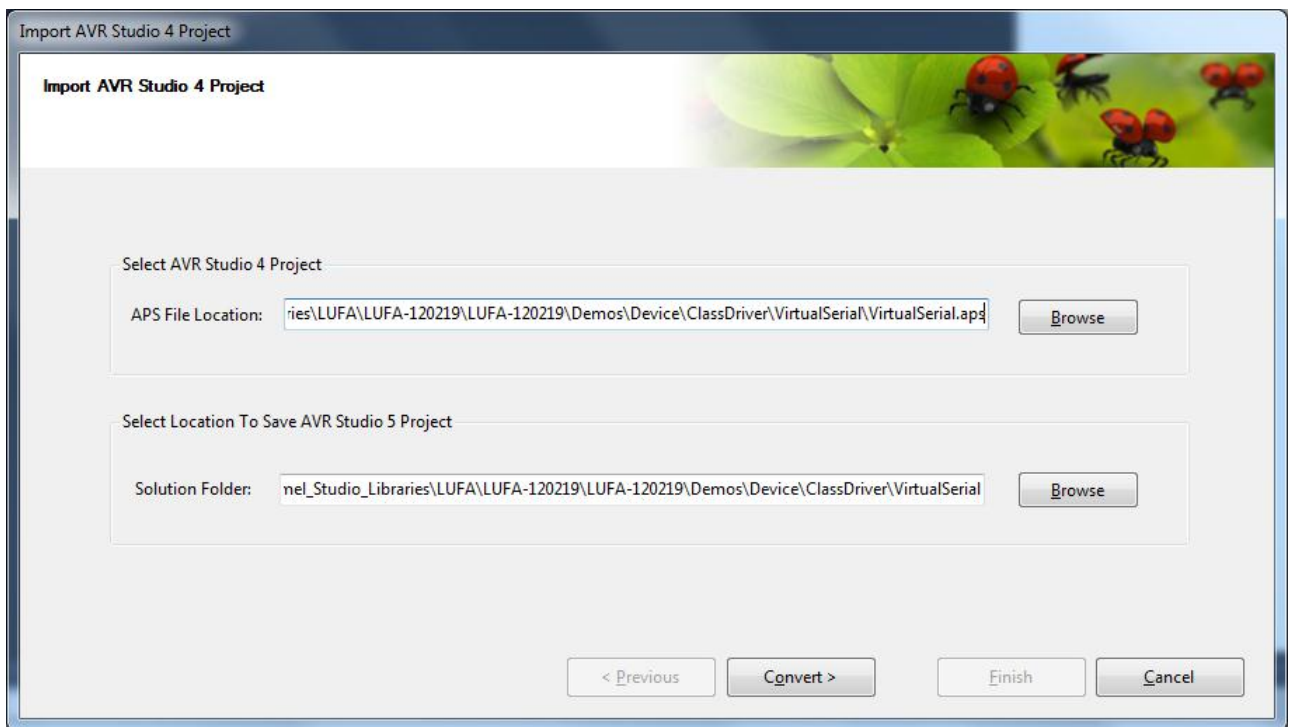


Figure 6: By default, the Studio5 project location is the location of the Studio 4 project file.

Notice that by default, the conversion will put the Studio5 Project in the same folder as the demo Project File.

Since we will be modifying the project source files to suit our own project requirements, we will specify a different directory for the Studio5 Project. Under “Select Location To Save AVR Studio 5 Project”, click Browse and navigate to where you want to locate the project. Click the “Make New Folder” button and name the new folder. This guide uses LUFA-Test2.

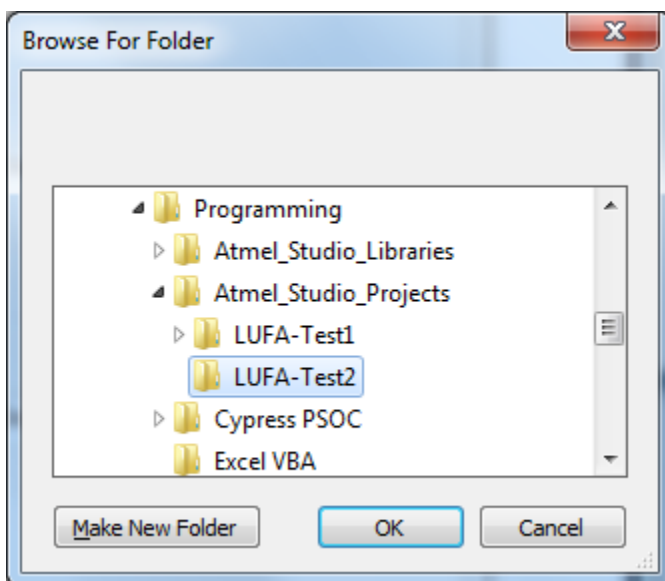


Figure 7: Make a new folder for the Studio5 Project – LUFA-Test2

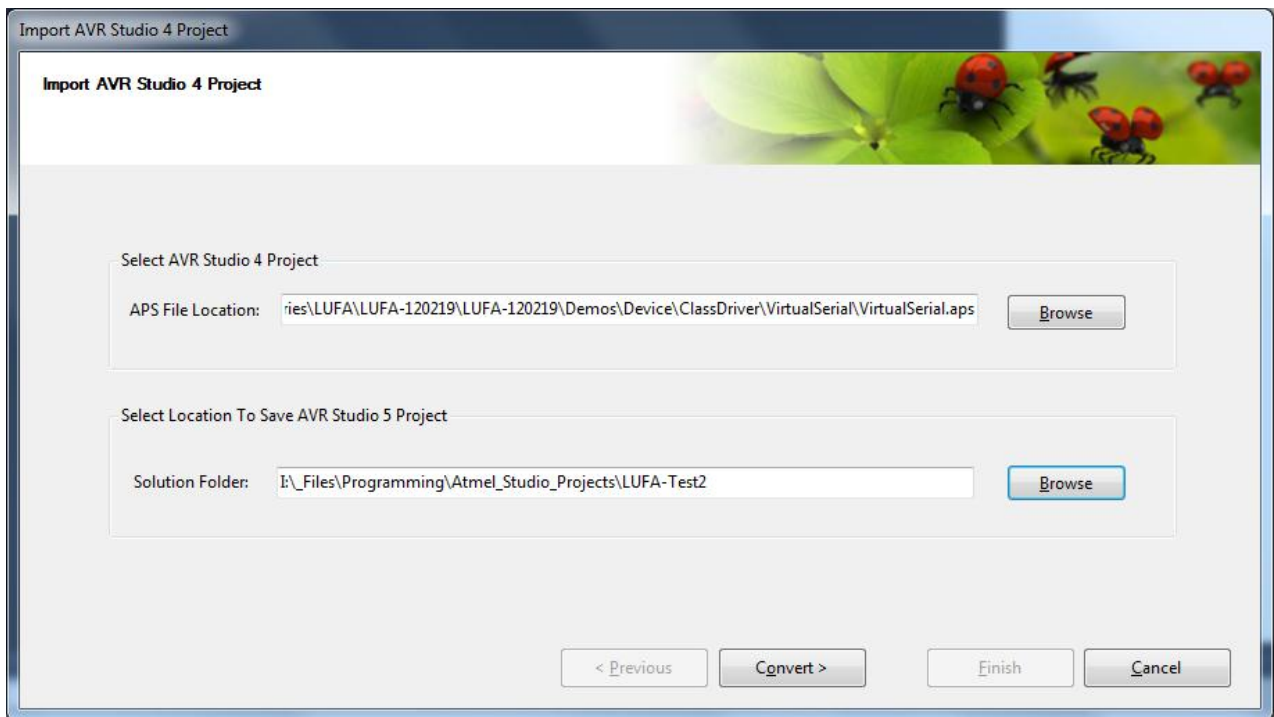


Figure 8: The two separate directories are shown and ready to Convert

Click “Convert” and then “Yes” to copy the files, then “Finish”. The Import function copies the files from the Studio4 project to the directory specified for the Studio5 project and creates a new Studio5 project file.

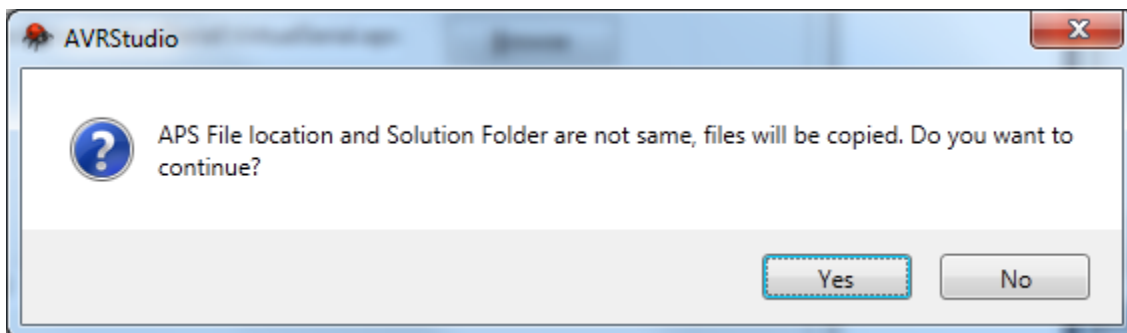


Figure 9: Import function will copy files. Click Yes.

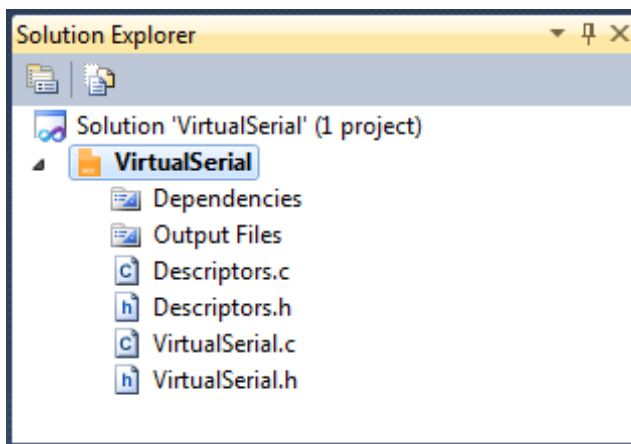


Figure 10: Solution Explorer shows the files in the new Studio5 project.

Step 2: Make Changes to the Studio5 Project Properties

Selecting the makefile that the Studio5 project will use.

Figure 11 shows the files in the new Studio5 project. Notice that the makefile for the VirtualSerial Demo has been copied to the project folder.

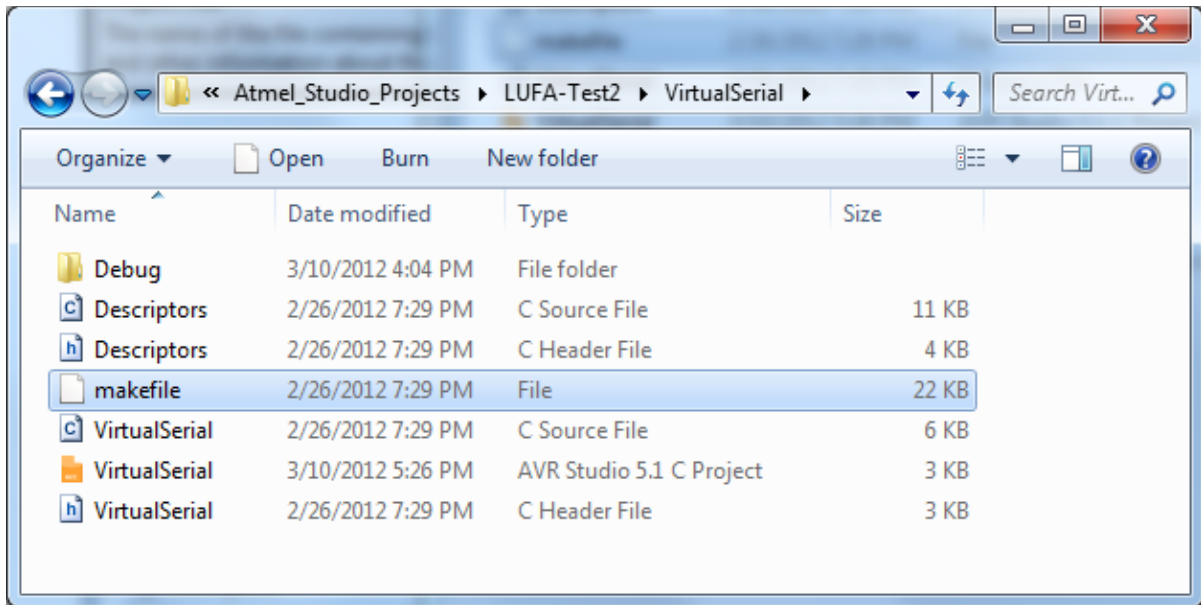


Figure 11: Though not shown in the Solution Explorer (Figure 10), the makefile was copied to the new folder.

To open the Project Properties Window:

In Solution Explorer, right click on the VirtualSerial project and then click on Properties in the popup menu.

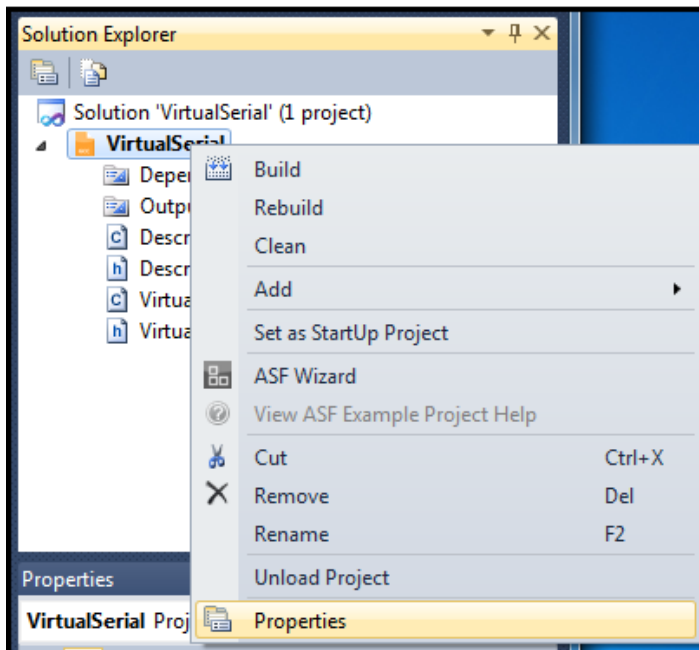


Figure 12: Right click on VirtualSerial and then click on Properties.

Studio5 now displays the Project Properties widow. Note the menu on the left hand side.

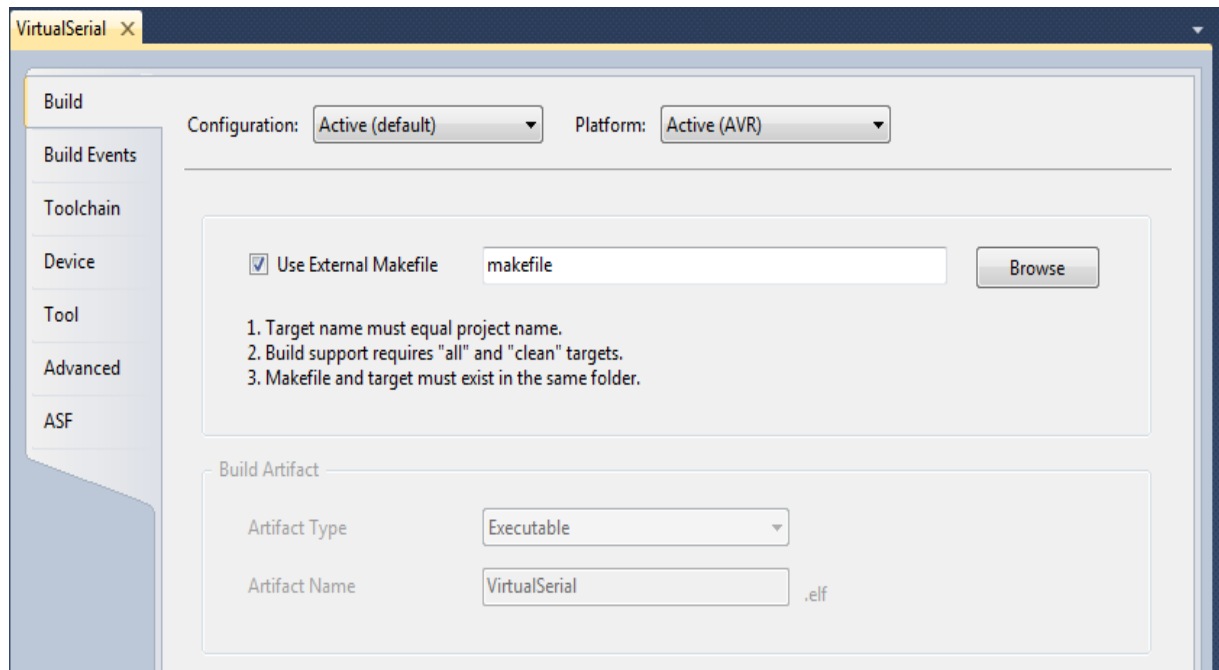


Figure 13: VirtualSerial Project Properties - Note that "Use External Makefile" is already checked.

Click the Browse button and you will see that the project is using the makefile in the demo folder.

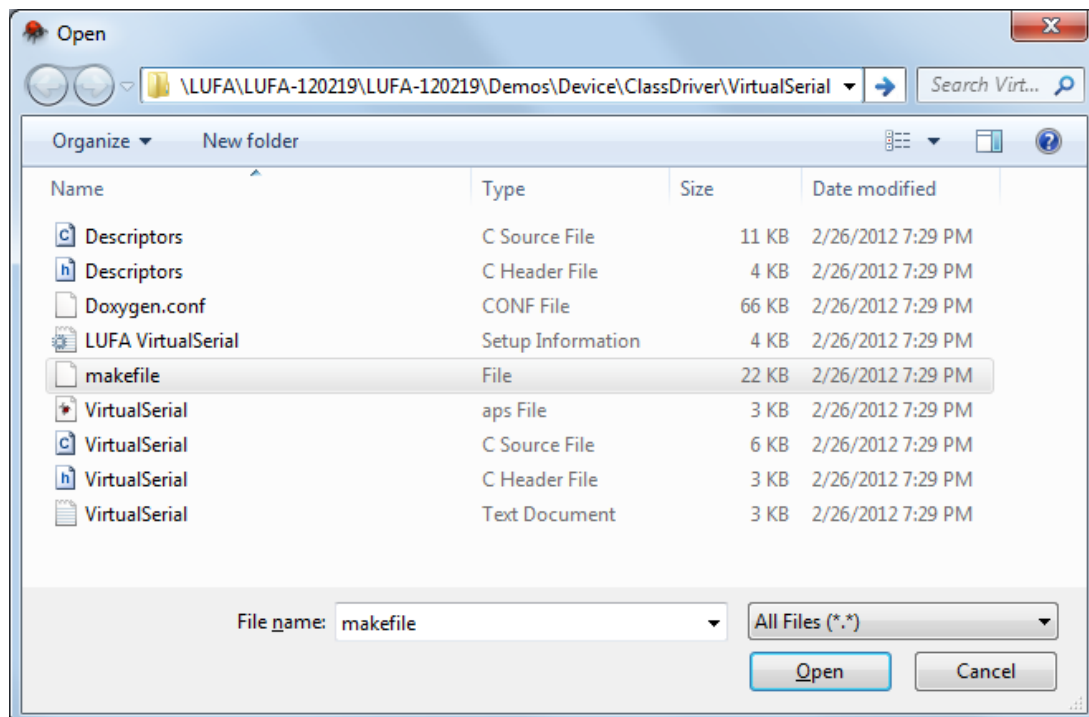


Figure 14: By default, the project properties selects the makefile located in the LUFA VirtualSerial Demo folder.

By default, the original makefile in the LUFA demo folder is selected. Because we'll be making changes to the makefile, we want to use the makefile that is located in the new Studio5 project folder.

Navigate to the makefile in the Studio5 project, and click open.

Figure 15 shows that the makefile in the Studio5 project is now specified.

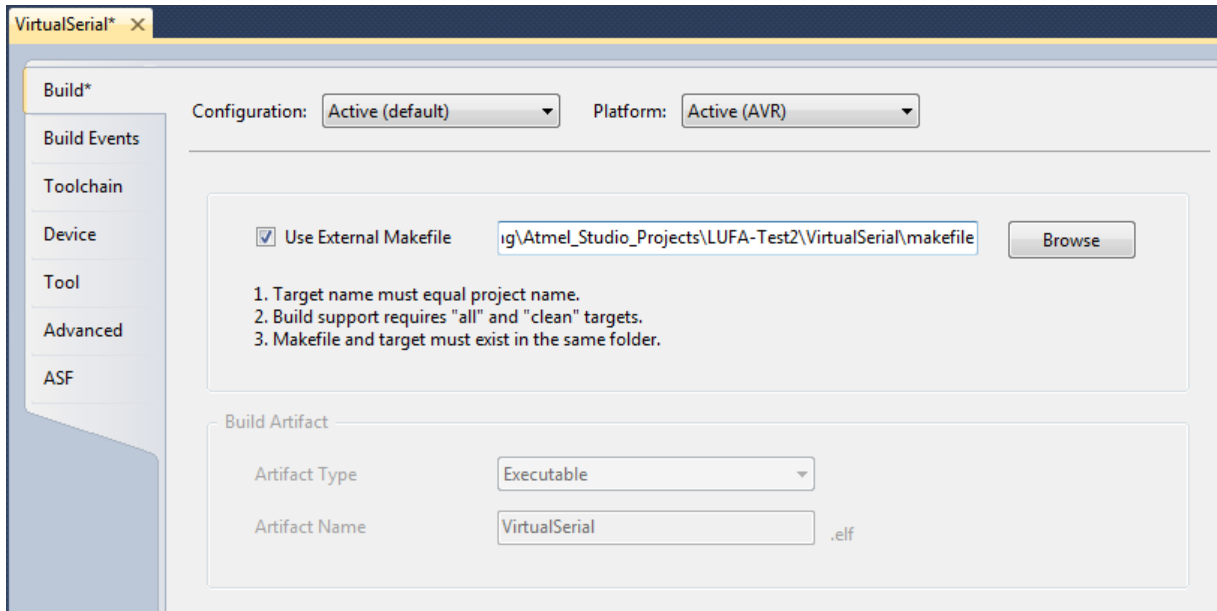


Figure 15: The makefile in the Studio5 project is now specified.

Selecting the device for the Studio5 project.

The makefile specifies the at90sub1287.

```
# MCU name
MCU = at90usb1287
```

Figure 16: The makefile specifies an AT90USB1287 microcontroller.

The Project Properties, under Device, shows that the default device is the ATmega128.

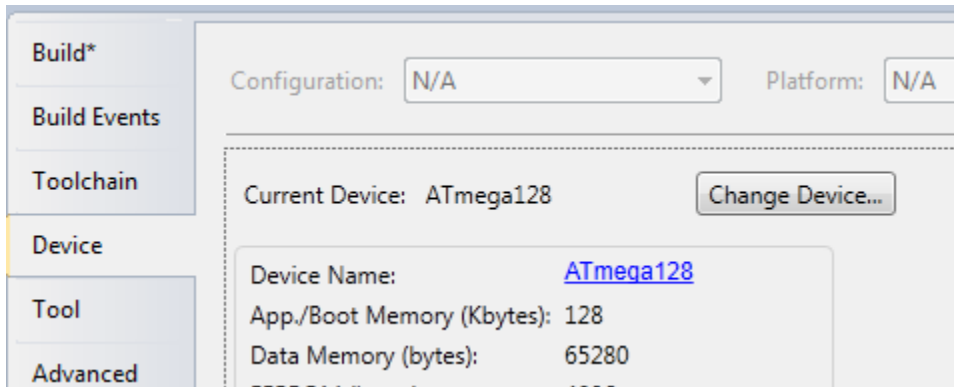


Figure 17: In the project properties under "Device", the default device is the ATmega128.

Click on the “Change Device” button and select the AT90USB1287.

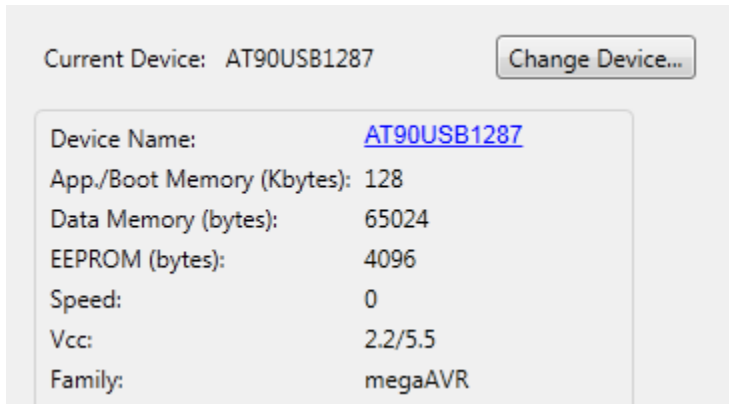


Figure 18: The correct device is now shown.

Step 3: Make Changes to the Makefile

Open the makefile in the Studio5 project. (I used Notepad to open it.)

The makefile specifies a path for locating LUFA library files.

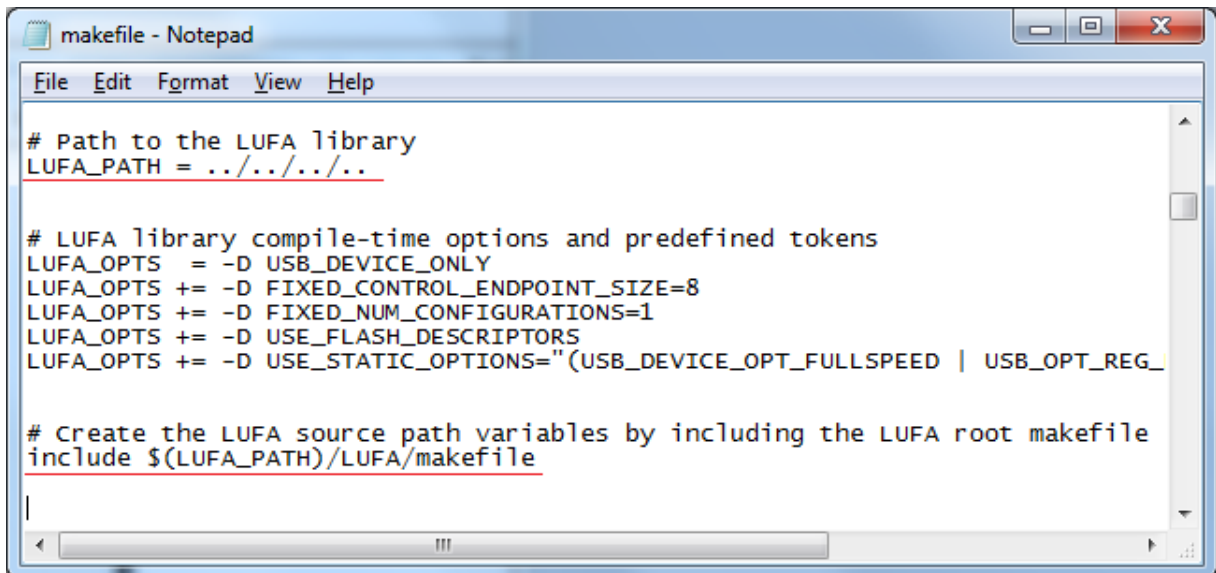


Figure 19: The LUFA_PATH needs to be changed to reflect the changed project folder.

The original makefile shows

```
LUFA_PATH = ../../../../..
```

and then

```
include $(LUFA_PATH)/LUFA/makefile
```

which translates to: backup four folders, open folder LUFA, and include file “makefile”.

This was for the original situation where both the project makefile and the library makefile were in the LUFA library folders. Now, however, the project makefile is in the Studio5 project folder while the library makefile is still in the LUFA library folder.

The directory structure that I'm working with is shown in figure 20.

```
Folder: I:\_Files\Programming\Atmel_Studio_Libraries\LUFA\LUFA-120219\LUFA-120219\Demos\Device\ClassDriver\VirtualSerial
File: makefile <Makefile for the demo project>

Folder: I:\_Files\Programming\Atmel_Studio_Libraries\LUFA\LUFA-120219\LUFA-120219\LUFA
File: makefile <Makefile for the LUFA library itself>

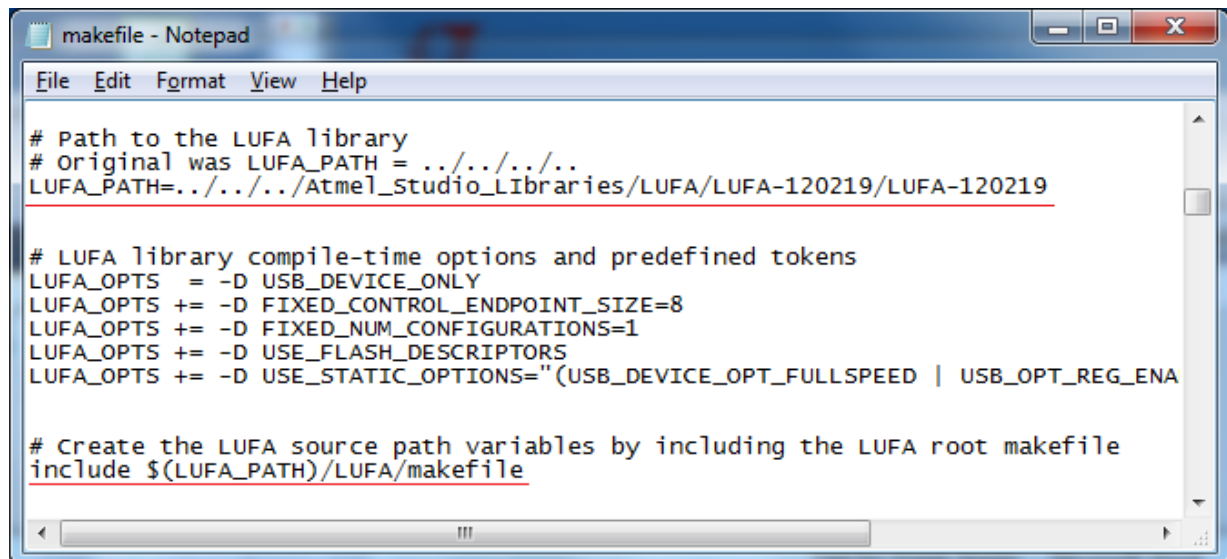
Folder: I:\_Files\Programming\Atmel_Studio_Projects\LUFA-Test2\VirtualSerial
File: makefile <Makefile for the Studio5 test project, which is a copy of the makefile of the demo project>
```

Figure 20: Directory structure showing the locations of the makefiles.

LUFA_PATH should be changed to: backup three folders, open folders "Atmel_Studio_Libraries", "LUFA", "LUFA-120219", and "LUFA-120219". This translates to:

```
LUFA_PATH=../../..../Atmel_Studio_Libraries/LUFA/LUFA-120219/LUFA-120219
```

Avoid using direct path specifications because, as I understand it, GCC Make doesn't like the ":" used by Windows (i.e. C:\). Figure 21 shows the new entry for LUFA_PATH.



```
File Edit Format View Help

# Path to the LUFA library
# Original was LUFA_PATH = ../../../../..
LUFA_PATH=../../..../Atmel_Studio_Libraries/LUFA/LUFA-120219/LUFA-120219

# LUFA library compile-time options and predefined tokens
LUFA_OPTS = -D USB_DEVICE_ONLY
LUFA_OPTS += -D FIXED_CONTROL_ENDPOINT_SIZE=8
LUFA_OPTS += -D FIXED_NUM_CONFIGURATIONS=1
LUFA_OPTS += -D USE_FLASH_DESCRIPTOR
LUFA_OPTS += -D USE_STATIC_OPTIONS="(USB_DEVICE_OPT_FULLSPEED | USB_OPT_REG_ENA

# Create the LUFA source path variables by including the LUFA root makefile
include $(LUFA_PATH)/LUFA/makefile
```

Figure 21: The new LUFA_PATH in the project makefile.

Save the make file and exit Notepad, and then in Studio5, under File, click Save All.

Step 4: Build the Project

Before building, we first must clean.

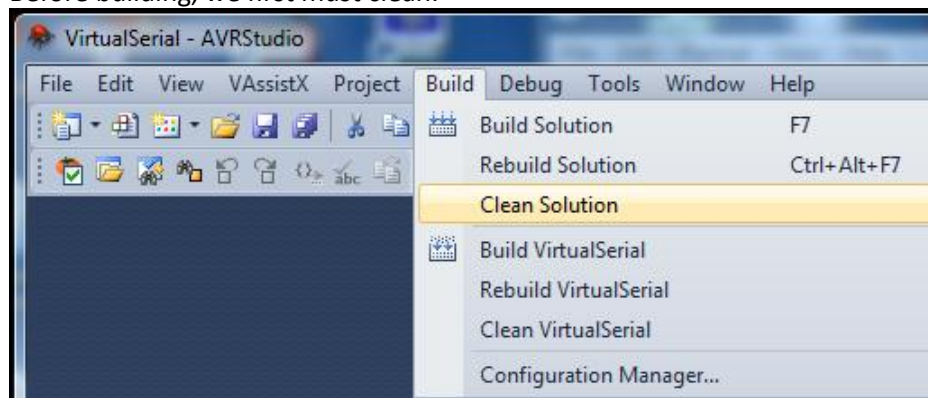


Figure 22: Clean the Solution before building.

Now we are ready to build.

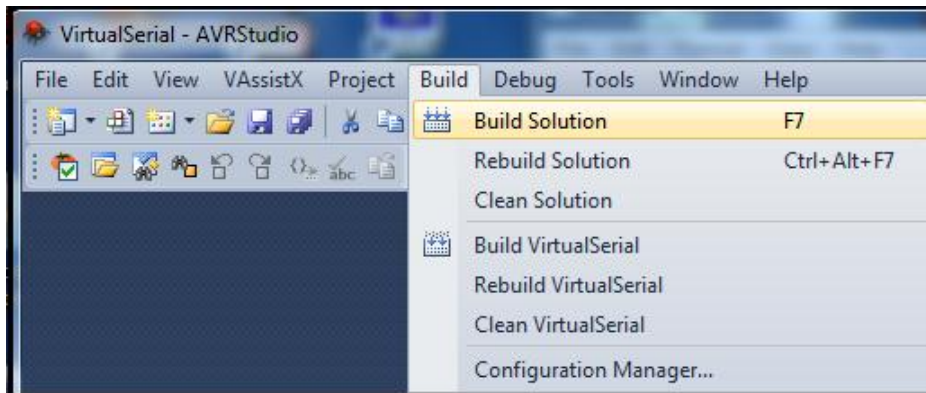


Figure 23: Build the Solution.

Finally, the result.

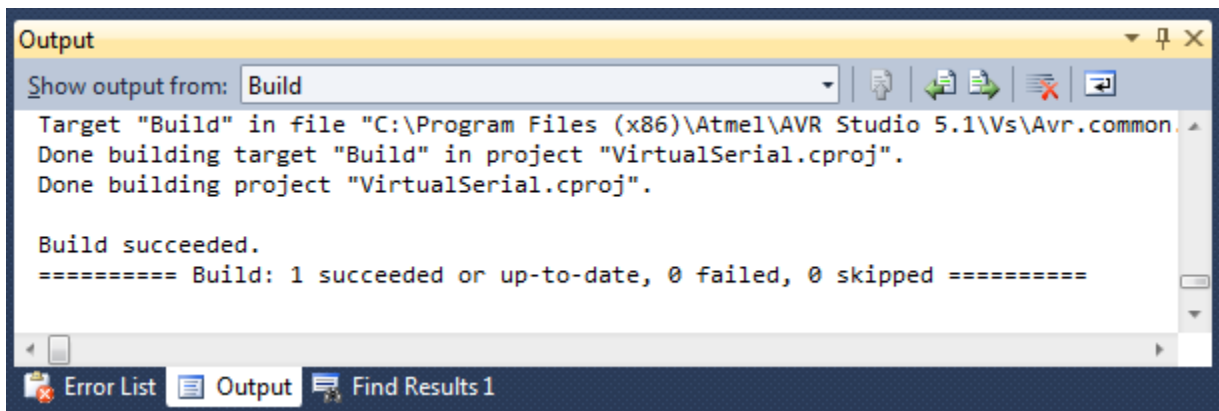


Figure 24: Build was successful.

While the build was successful, I don't know if the demo actually works because I don't have the board it was designed for.

Onward and Upward

Part 2, the second rung of the ladder, will be modifying the Studio5 project to work with the AVR AT90USB162 microcontroller in the OLIMEX162 board.