

Piksi Developer Getting Started Guide

This is a guide to getting started developing for the Swift Navigation Piksi GPS Receiver. It will provide an overview of how to install and use the tools necessary to build and debug code for the onboard STM32F4 microcontroller, update the configuration for the onboard Spartan 6 FPGA, and use various other tools we have developed for interacting with the device.

This guide was tested on Linux (Ubuntu 12.10 and 14.01 32-bit, kernel version 3.5.0-17), OS X (10.6 - 10.9), and Windows 7 (SP1) installations.

If anything in this guide is incorrect or unclear, please email our [mailing list \(https://groups.google.com/forum/#!forum/swiftnav-discuss\)](https://groups.google.com/forum/#!forum/swiftnav-discuss).

Contents

Building the Piksi firmware

Flashing the firmware to the Piksi

Using the Piksi

Updating to the latest firmware version

Updating the SwiftNAP (FPGA) firmware

Ubuntu Notes

- Installing FTDI USB drivers

- Adding USB udev rules

OS X Notes

- USB Serial communications

 - FTDI Drivers

 - OS X

Building the Piksi firmware

1. Clone the piksi_firmware repository from Github:

```
$ git clone https://github.com/swift-nav/piksi_firmware.git
```

2. Before starting development, you will need to install some dependencies in your development environment. You may either (i) bootstrap those dependencies into your native development environment (either OS X or Ubuntu/Linux), or (ii) provision a virtual machine.

```
$ cd ./piksi_firmware
# Option 1 (OS X and Ubuntu): Run bootstrapping script.
$ sudo bash setup.sh -x install
# Option 2 (All platforms): Provision a VM using VirtualBox (https://www.virtualbox.org/wiki/Downloads) and Vagrant (http://www.vagrantup.com/downloads.html).
$ vagrant up
```

Installation instructions for Windows, including additional details about setting up development dependencies, are at:

[HOW-TO: Setting up developer tools for Piksi.](#)

3. Initialize Git submodules via the following commands:

```
$ git submodule init
$ git submodule update
```

Enter the directory and build the firmware:

```
$ cd piksi_firmware
$ make
```

If the build was successful, there will be firmware binary files under `piksi_firmware/build/` in a variety of formats (ELF, Intel HEX, binary).

Flashing the firmware to the Piksi

The next step after successfully building the Piksi firmware is to flash it to the on-board microcontroller. This can be done in two ways, the simplest is to use the built in bootloader which allows the firmware to be flashed over USB with no additional tools. Please refer to our guide on using the bootloader for flashing the firmware:

[HOW-TO: Flashing the Piksi firmware using the bootloader](#)

The second method is to use a JTAG programmer. Every JTAG programmer is different so please refer to the documentation available for your particular programmer. At Swift Navigation we use and highly recommend the [Black Magic Probe](http://www.blackmagic.co.nz/main/blackmagic) (<http://www.blackmagic.co.nz/main/blackmagic>) JTAG programmer.

[HOW-TO: Flashing and debugging the Piksi firmware using Black Magic Probe and gdb](#)

Using the Piksi

Now you have the firmware built and flashed to the Piksi it should be ready for use. Please refer to the user guide for instructions on running the Piksi for the first time.

[Piksi User Getting Started Guide](#)

Updating to the latest firmware version

We recommend becoming familiar with the [git](http://git-scm.com/) (<http://git-scm.com/>) version control system which we use for all of our projects.

To use git to update to the latest version, from the `piksi_firmware` directory, run:

```
$ git pull origin master
$ git submodule init
$ git submodule update
```

Then build the firmware again to compile new, updated binaries:

```
$ make clean
$ make
```

Updating the SwiftNAP (FPGA) firmware

From time to time Swift Navigation may offer updates to the SwiftNAP firmware as we further develop the SwiftNAP FPGA design. Please refer to SwiftNAP section of the firmware update guide.

HOW-TO: Flashing the Piksi firmware using the bootloader: Updating the SwiftNAP FPGA firmware

This guide will help you set up the required software for building the Piksi firmware and running the Python tools from source.

As of November 2014, there are a few options:

- **Normal usage.** If you're only using the Piksi console, download easy-to-use binary installers (Windows and OS X) are here (http://downloads.swiftnav.com/piksi_console/) and follow the instructions at the [Piksi User Getting Started Guide](#).
- **Development (native and VM).** To install development tools for your platform (OS X and Ubuntu), you may either bootstrap the required dependencies to your OS or provision a separate VM.

```
$ git clone https://github.com/swift-nav/piksi_firmware.git
# Option 1 (OS X and Ubuntu): Run the bootstrapping script.
$ bash setup.sh -x install
# Option 2 (All platforms): The Vagrant file is currently used for testing installation setup.sh, but can also be used to
provision a development VM using VirtualBox (https://www.virtualbox.org/wiki/Downloads) and Vagrant (http://www.vagrantup.com/downloads.html):
$ vagrant up
```

The setup.sh script will support Windows in the future.

If you encounter any difficulties in completing these instructions, please email our [mailing list](https://groups.google.com/forum/#!forum/swiftnav-discuss) (<https://groups.google.com/forum/#!forum/swiftnav-discuss>).

Ubuntu Notes

The setup tools have been tested on Ubuntu 12.10 and 14.01 Please note the special instructions below if using using a version less than 12.04.

Installing FTDI USB drivers

As of Ubuntu 11.10, kernel 3.0.0-19, all FTDI devices are supported by the kernel in Virtual Com Port mode. If you are using an earlier version of the kernel, you may need to install the FTDI VCP drivers yourself : <http://www.ftdichip.com/Drivers/VCP.htm>.

Adding USB udev rules

We have to create a rules file in `/etc/udev/rules.d` to allow programs to interact with Piksi over USB without root permissions.

```
$ sudo tee /etc/udev/rules.d/99-piksi.rules <<EOF
ATTRS{idProduct}=="6014", ATTRS{idVendor}=="0403", MODE="666", GROUP="plugdev"
ATTRS{idProduct}=="8398", ATTRS{idVendor}=="0403", MODE="666", GROUP="plugdev"
EOF
```

OS X Notes

USB Serial communications

FTDI Drivers

The Piksi has an on-board FTDI USB serial converter which is used for host communications. This converter can be used in two ways. The first is to install the standard drivers from FTDI which make the Piksi appear as a regular serial port. They can be found at <http://www.ftdichip.com/Drivers/VCP.htm>. This is the recommended solution.

OS X

NOTE: If you are running Mac OS X 10.9+ (Mavericks) you MUST install the official driver from the FTDI website. The built in driver is buggy and WILL NOT WORK with Piksi

Some users are now reporting that the OS X drivers will continue to claim the device even after installing the official FTDI VCP driver. If you experience crashes or lock-ups while using Piksi you can try forcibly unloading the Apple driver and loading the FTDI one:

```
$ sudo kextunload /System/Library/Extensions/IOUSBFamily.kext/Contents/PlugIns/AppleUSBFTDI.kext
$ sudo kextload /System/Library/Extensions/FTDIUSBSerialDriver.kext
```

Retrieved from "http://docs.swiftnav.com/w/index.php?title=Piksi_Developer_Getting_Started_Guide&oldid=22819"

This page was last edited on 28 January 2016, at 03:13.