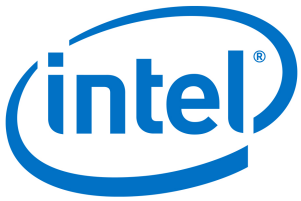


Updating Arduino 101 Firmware

User Guide

March 2016

Table of Contents

Tables

1 Introduction

1.1 Prerequisites

This firmware is intended to be built only on 64 bit Ubuntu systems. If your native machine is not running a 64 bit Ubuntu operating system, you can still perform the firmware building process using an Ubuntu 14 64 bit OS in a virtual machine with 15GB of HDD space allocated. It is advised that you do not use a live USB or CD. Doing so will result in some steps failing, as it would not be possible to download the required packages.

If you have not configured the Arduino101 board yet, follow the steps in the link below before downloading the source code. Verify that you can successfully run a blink sketch to ensure the DFU is operating correctly. This is important, as it will provide you with the platform to flash the binaries.

https://www.arduino.cc/en/Guide/Arduino101

§

2 Installation and Setup

2.1 Downloading the source code

Visit *https://downloadcenter.intel.com/download/25832* and download the tar-ball located there. When the download is complete, move it into the folder where the tar-ball is saved, and execute the following commands:

```
$ tar -xf arduino101_firmware_source-v1.tar.bz2
$ cd arduino101_firmware_source-v1
$ project_directory=$(pwd)/arduino101_firmware/projects/arduino101
```

2.2 Installing prerequisite packages

Verify that packages from all repositories are up to date:

```
$ sudo apt-get update
```

Ensure that you have all the required packages before compiling. As the target suggests, this is the only required the first time you compile:

```
$ sudo make one_time_setup -C $project_directory
```

This installs the following packages:

```
gawk wget git-core diffstat unzip texinfo
gcc-multilib build-essential chrpath libsdl1.2-dev xterm
libqtgui4:i386 libtool libc6:i386
libc6-dev-i386 autoconf libtool pkg-config gperf flex bison
```

2.3 Building images

Enter this command each time you modify the code to update the images:

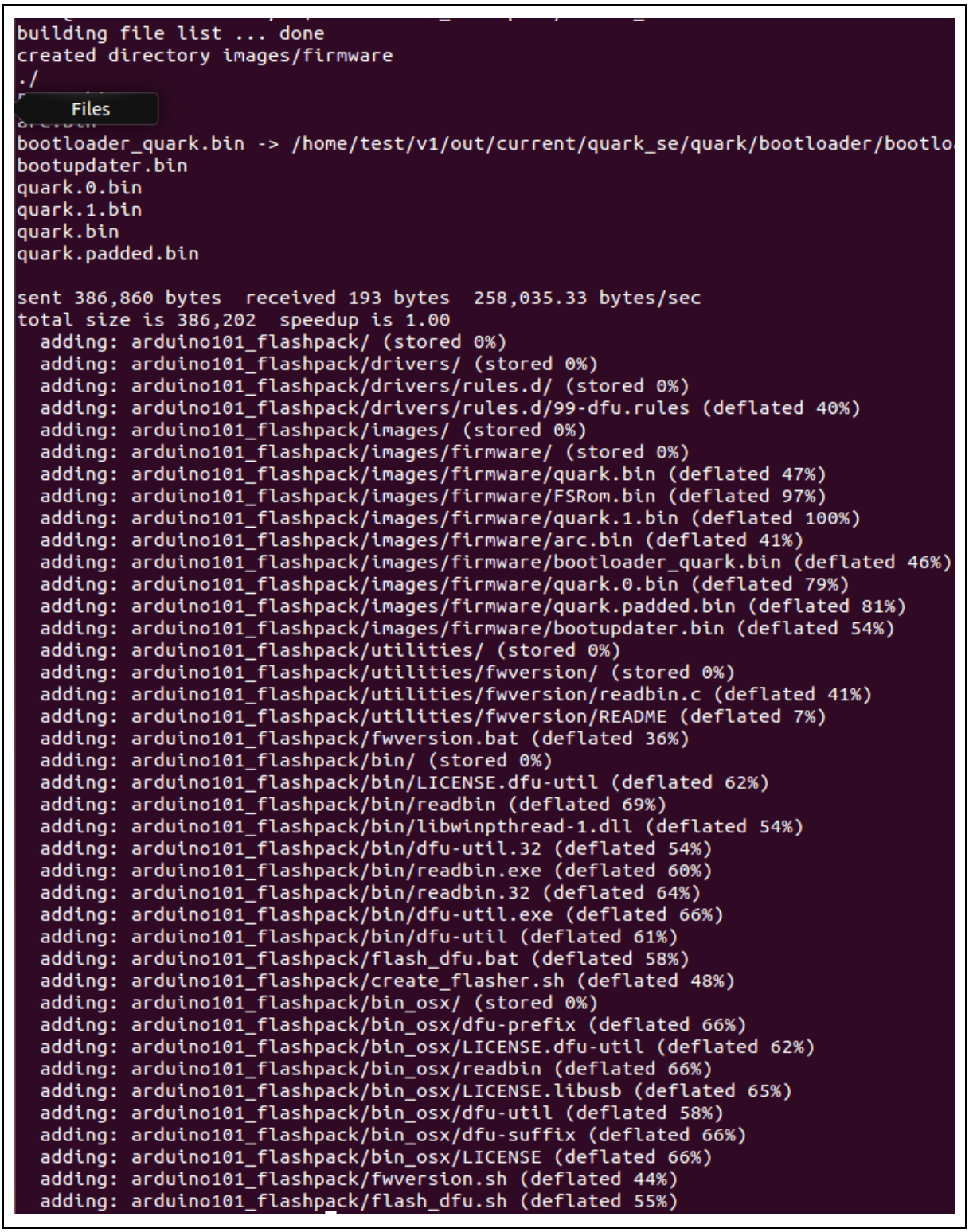
```
$ make clean setup image -C $project_directory
```

2.4 Creating flashpack.zip

This command creates "flashpack.zip" which is used for flashing the board:

```
$ arduino101_flashpack/create_flasher.sh
```

Figure 1 Creating flashback.zip



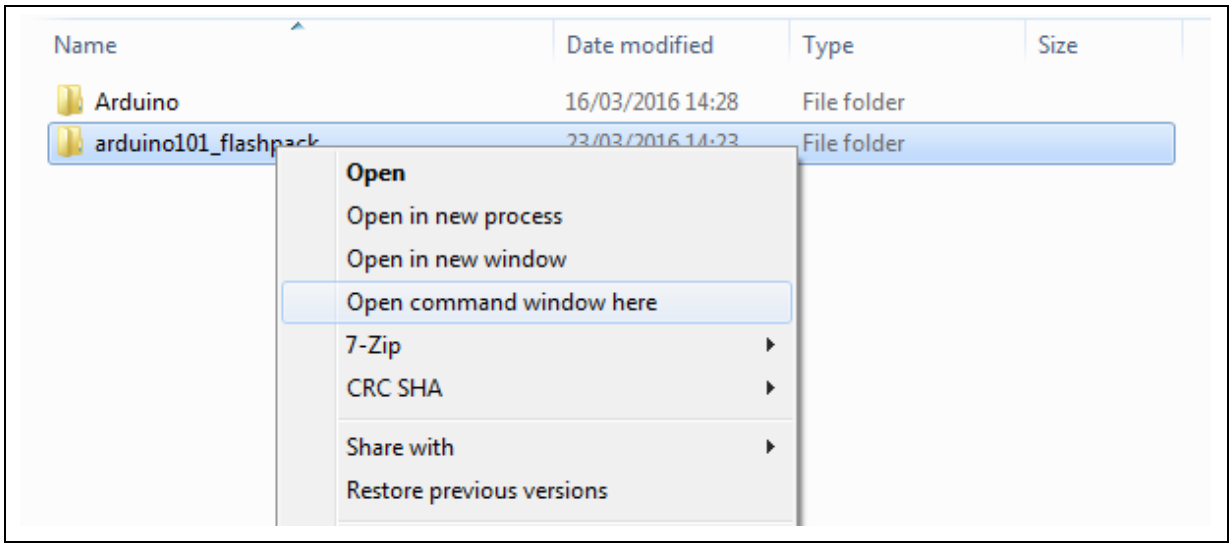
2.5 Flashing images to the board

Ensure the board is connected via USB, move the .zip file to the machine where the Arduino IDE is installed, and extract there.

2.5.1 Windows

Shift+Ctrl+right click mouse on extracted folder and click “Open command window here”.

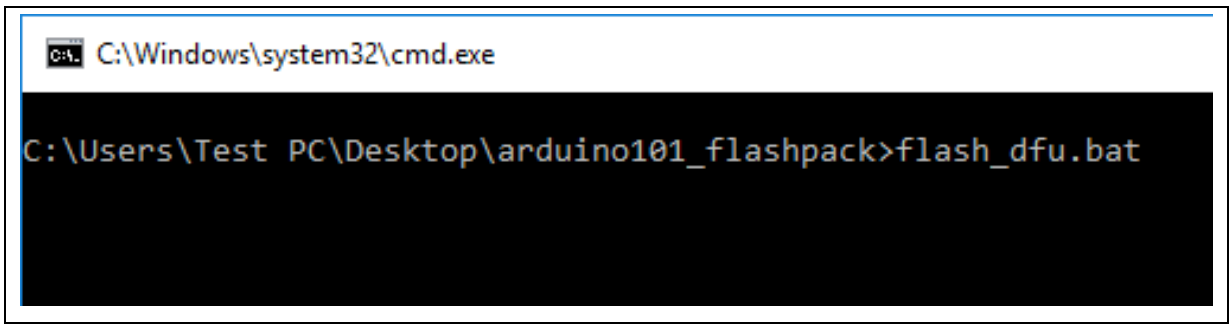
Figure 2 Flashing images to the board with Windows– open command window



Execute command:

```
flash_dfu.bat
```

Figure 3 Flashing images to the board with Windows – execute command

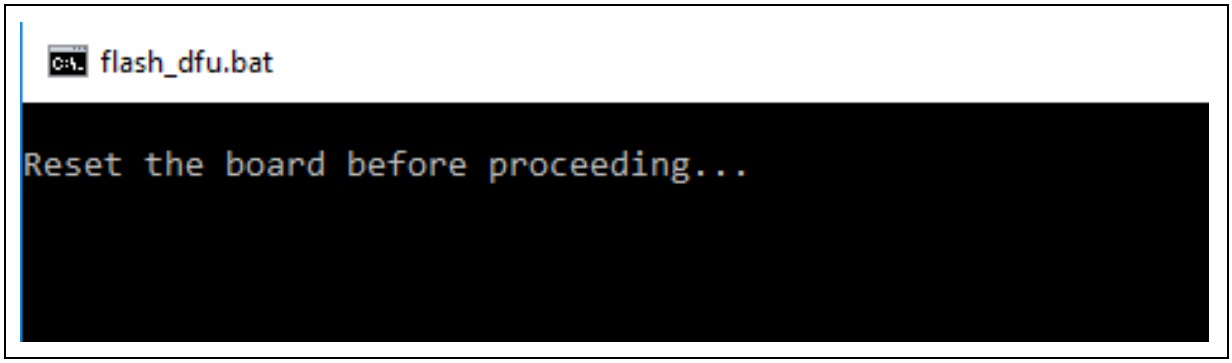


2.5.2 Linux/Mac

```
$ cd arduino101_flashpack
$ ./flash_dfu.sh
```

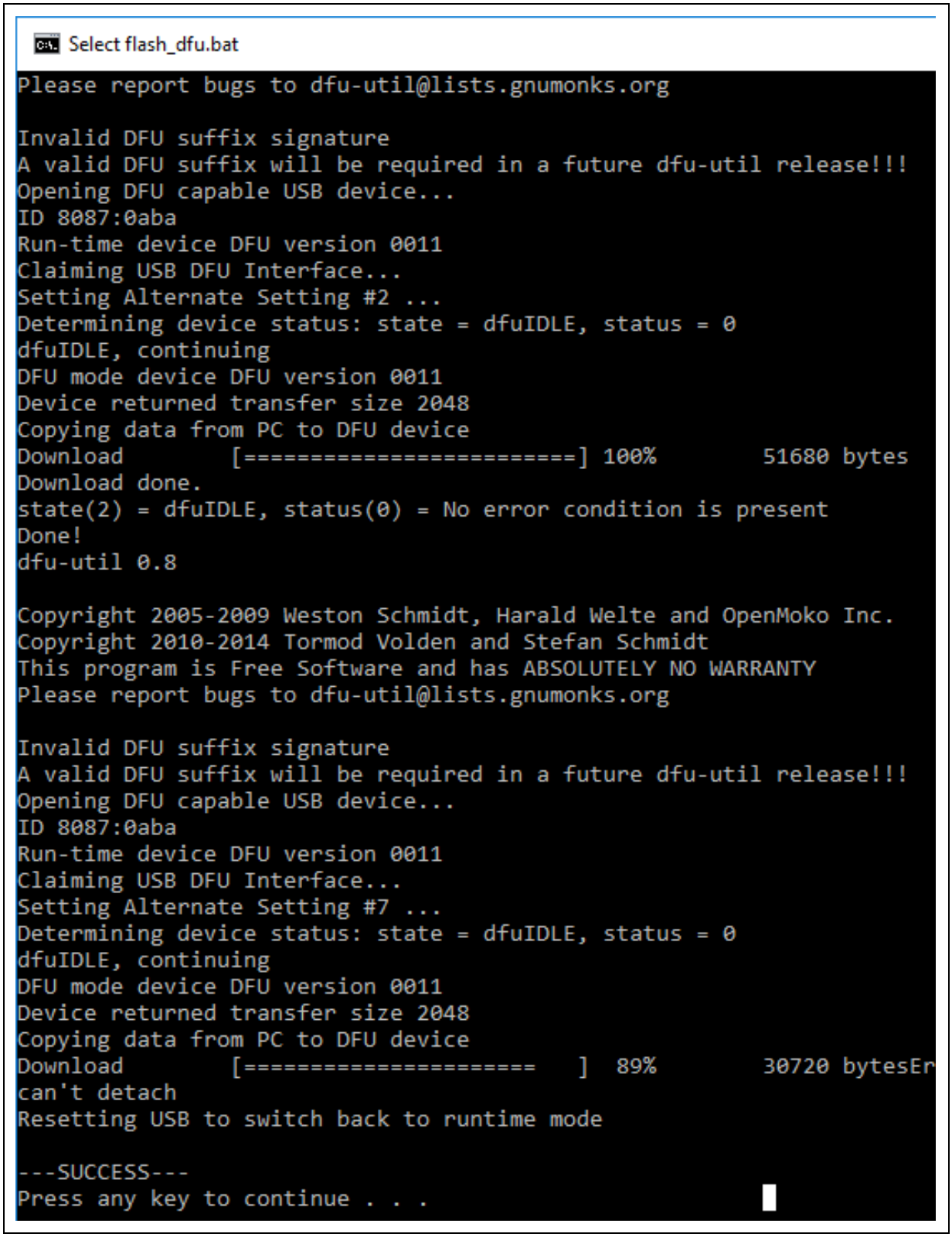
Press the reset button on the board to begin the flash process.

Figure 4 Flashing images to the board with Linux/Mac – reset the board



The figure below shows an example of a successful flash.

Figure 5 Flashing images to the board with Linux/Mac – successful flash



§