



## **Ajile Suite Software Users Guide**

Ajile Light Industries ©, Ottawa, Canada, 2018

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Overview of Features . . . . .	1
1.2	Software Installation . . . . .	2
1.2.1	Obtaining Ajile Software . . . . .	2
1.2.2	Software Installation in Windows . . . . .	2
1.2.3	Software Installation in Ubuntu . . . . .	7
1.3	Running the Examples . . . . .	11
1.3.1	Running the Examples in the GUI . . . . .	11
1.3.2	Running the Examples with the SDK . . . . .	13
1.4	Upgrading Ajile Software and Firmware . . . . .	15
1.4.1	Upgrading Ajile Firmware . . . . .	15
1.4.2	Upgrading Ajile Software . . . . .	16
<b>2</b>	<b>Project Model Overview</b>	<b>18</b>
2.1	Projects . . . . .	18
2.2	Components . . . . .	18
2.3	Images . . . . .	20
2.4	Sequences, Sequence Items, Frames . . . . .	20
2.5	Lighting . . . . .	21
2.6	Triggers . . . . .	21
<b>3</b>	<b>Projects</b>	<b>25</b>
3.1	Project Members . . . . .	25
3.2	Creating New Projects . . . . .	25
3.2.1	Creating New Projects in the GUI . . . . .	25
3.2.2	Creating New Projects in the SDK . . . . .	26
3.3	Saving and Opening Projects . . . . .	28
3.3.1	Saving and Opening Projects in the GUI . . . . .	28
3.3.2	Saving and Opening Projects in the SDK . . . . .	28
<b>4</b>	<b>Components</b>	<b>31</b>
4.1	Component Members . . . . .	31
4.2	Initializing Components . . . . .	31
4.2.1	Reading Components Kits from File . . . . .	31
4.2.2	Retrieving Components from the Hardware . . . . .	33
4.2.3	Creating Custom Components . . . . .	35
4.3	Configuring Components . . . . .	35
4.3.1	Configuring Components in the GUI . . . . .	35
4.3.2	Configuring Components in the SDK . . . . .	36
<b>5</b>	<b>Images</b>	<b>37</b>
5.1	Image Members . . . . .	37
5.2	Image Data Format . . . . .	37
5.3	Creating DMD Images . . . . .	40

5.3.1	Creating DMD Images in the GUI . . . . .	40
5.3.2	Creating DMD Images in the SDK . . . . .	41
<b>6</b>	<b>Sequences</b>	<b>45</b>
6.1	Sequence, Sequence Item and Frame Members . . . . .	45
6.1.1	Sequence Members . . . . .	45
6.1.2	Sequence Item Members . . . . .	45
6.1.3	Frame Members . . . . .	45
6.2	Sequence Structure . . . . .	48
6.3	Creating Sequences . . . . .	49
6.3.1	Creating Sequences in the GUI . . . . .	50
6.3.2	Creating Sequences in the SDK . . . . .	51
6.4	Adding Sequence Items and Frames . . . . .	51
6.4.1	Adding Sequence Items and Frames in the GUI . . . . .	52
6.4.2	Adding Sequence Items and Frames in the SDK . . . . .	52
6.5	Modifying Sequence Item and Frame Parameters . . . . .	53
6.5.1	Modifying Sequence Item and Frame Parameters in the GUI . . . . .	53
6.5.2	Modifying Sequence Item and Frame Parameters in the SDK . . . . .	54
6.6	Verifying Sequences . . . . .	56
6.6.1	Verifying Sequences in the GUI . . . . .	56
6.6.2	Verifying Sequences in the SDK . . . . .	56
<b>7</b>	<b>Lighting</b>	<b>59</b>
7.1	Lighting Introduction . . . . .	59
7.1.1	Lighting Controller Overview . . . . .	59
7.1.2	Lighting Control Software Overview . . . . .	59
7.2	Lighting Members . . . . .	60
7.2.1	LED Property Members . . . . .	60
7.2.2	LED Setting Members . . . . .	61
7.3	LED Settings Detailed Description . . . . .	61
7.4	Configuring Component LED Properties . . . . .	63
7.4.1	Configuring LED Properties in the GUI . . . . .	63
7.4.2	Configuring LED Properties in the SDK . . . . .	64
7.5	Configuring LED Settings per Frame . . . . .	65
7.5.1	Configuring LED Settings in the GUI . . . . .	65
7.5.2	Configuring LED Settings in the SDK . . . . .	66
<b>8</b>	<b>Triggers</b>	<b>68</b>
8.1	Device State Outputs . . . . .	69
8.2	Device Control Inputs . . . . .	69
8.3	Trigger Members . . . . .	69
8.3.1	External Trigger Setting Members . . . . .	69
8.3.2	Trigger Rule Members . . . . .	69
8.3.3	Frame Trigger Setting Members . . . . .	69
8.4	Trigger Timing . . . . .	72
8.5	Trigger Rule Structure . . . . .	72
8.6	Configuring Trigger Settings . . . . .	76
8.6.1	Configuring Trigger Settings in the GUI . . . . .	76
8.6.2	Configuring Trigger Settings in the SDK . . . . .	76
8.7	Creating Trigger Rules . . . . .	78
8.7.1	Creating Trigger Rules in the GUI . . . . .	79
8.7.2	Creating Trigger Rules in the SDK . . . . .	79
8.8	Per Frame Trigger Settings . . . . .	82
8.8.1	Per Frame Trigger Settings in the GUI . . . . .	82
8.8.2	Per Frame Trigger Settings in the SDK . . . . .	83

<b>9</b>	<b>System Control</b>	<b>85</b>
9.1	Connecting to the Device . . . . .	85
9.1.1	Connecting to the Device in the GUI . . . . .	85
9.1.2	Connecting to the Device in the SDK . . . . .	86
9.2	Loading Projects . . . . .	87
9.2.1	Loading Projects in the GUI . . . . .	87
9.2.2	Loading Projects in the SDK . . . . .	87
9.3	Running Sequences . . . . .	89
9.3.1	Running Sequences in the GUI . . . . .	89
9.3.2	Running Sequences in the SDK . . . . .	89
9.4	Device Status Information . . . . .	90
9.4.1	Device State . . . . .	90
9.4.2	Sequence Status . . . . .	91
9.5	Streaming Sequences . . . . .	93
9.5.1	Running Streaming Sequences in the SDK . . . . .	93
<b>10</b>	<b>Color and Grayscale Display</b>	<b>97</b>
10.1	Displaying Color/Grayscale as a List of Bitplanes . . . . .	98
10.1.1	Splitting Multi-Bit Images into Bitplanes . . . . .	98
10.1.2	Displaying Bitplanes of n-Bit Images . . . . .	98
10.1.3	Grayscale Display: Frame Time Control Only . . . . .	99
10.1.4	Grayscale Display: Frame Time and LED Power Control . . . . .	99
10.1.5	Grayscale Display Optimization . . . . .	100
10.1.6	Color Display . . . . .	102
10.2	Displaying Color and Grayscale Images . . . . .	102
10.2.1	Displaying Color and Grayscale Images in the GUI . . . . .	103
10.2.2	Displaying Color and Grayscale Images in the SDK . . . . .	104
10.3	Creating High Bit-Depth (>8-bit) Color and Grayscale Sequence Items . . . . .	108
10.4	Optimizing the Output Linearity of Color and Grayscale Images . . . . .	108
10.5	Optimizing Color and Grayscale for Human Display . . . . .	109

# Chapter 1

## Introduction

Ajile has developed a suite of hardware and software products that cooperatively support fast image creation and projection by Ajile DMD structured light projectors as well as coordinated fast image capture by Ajile smart cameras. These products have broad applicability in areas such as machine vision, automated inspection, vision testing and spectroscopy.

The Ajile suite is designed to make management of light a simpler process. Users can begin with an easy to use graphical user interface (GUI) to quickly and easily create projects which create, project and capture images. The GUI workflow will not only give one familiarity with the project structure and the way in which components interact, but will also enable users to create and run fully functional experiments to accomplish meaningful tasks.

For greater levels of control, flexibility and power, one can make use of the Ajile software development kit (SDK) which exposes in a programmatic form the same object-oriented project structure and overall workflow which are available in the GUI, available in either Python or C++ programming languages.

The tight integration of both hardware and software within a single coherent suite makes many tasks straightforward which were previously either extremely difficult or even impossible with other existing solutions.

### 1.1 Overview of Features

The following list provides an overview of some of the more interesting, unique, or otherwise nice to have features which are found in the Ajile software suite. These features along with many others will be described in detail throughout the documentation.

- Object-oriented project model describes images, sequences, frames and their relationships with one another.
- Projects created by the GUI and/or SDK, then loaded and run directly by the controller hardware.
- Control of nearly every imaging parameter (e.g. frame time, lighting settings, region of interest, etc.) on a frame by frame (per frame) basis.
- Tight synchronization of components and external devices by graphically specifying trigger rules in software which are then evaluated in hardware with nanoseconds of latency.
- Images preloaded, stored and run from controller memory, or streamed continuously from a PC over a number of available interfaces.
- Symmetric view of both projectors (which consume/display images) and cameras (which produce/-capture images). Both types of components follow the same overall project model.

- Projects easily ported between GUI and SDK applications.
- SDK applications can be run on a host PC, or run directly on the embedded Linux controller with minimal code changes.
- Embedded image processing algorithms which run in FPGA hardware but configured in software allow smart image pipelines for processing or generating images at high speeds.

## 1.2 Software Installation

Currently supported operating systems for the Ajile GUI and SDK include Windows 7 / 8.1 / 10 and Ubuntu Linux 14.04 / 16.04. Other distributions of both Windows and Linux may be possible but are not officially supported and so their operation cannot be guaranteed.

### 1.2.1 Obtaining Ajile Software

To obtain the Ajile software suite packages, go to the Ajile downloads section of the Ajile website at <http://ajile.ca/downloads> and select the files for your operating system of choice. You will need to enter in the login username and password which were supplied to you.

For Windows installations, download the most recent .msi installation package (e.g. `ajile_installer_win64_2017-07-11_1.0-4.exe`). For Ubuntu installations, download the most recent .tar.gz archive for your Linux distribution (e.g. for Ubuntu 16.04, `ajile_suite_linux64_16.04_x86_64_2017-07-11_1.0-4.tar.gz`).

### 1.2.2 Software Installation in Windows

#### Installing Prerequisites

To install the Ajile software in Windows we first need to install a few prerequisites which allow the software to run. Prior to installing the Ajile software please install all Windows updates by opening the Windows Update tool in Windows, checking for new updates, and selecting and installing all updates which are recommended by Windows. Note that you may need to restart your computer several times to complete the updates and each time after a restart you should again check for updates in the Windows Update tool as new updates may become available once the previous updates have been installed.

#### Installing Ajile Software

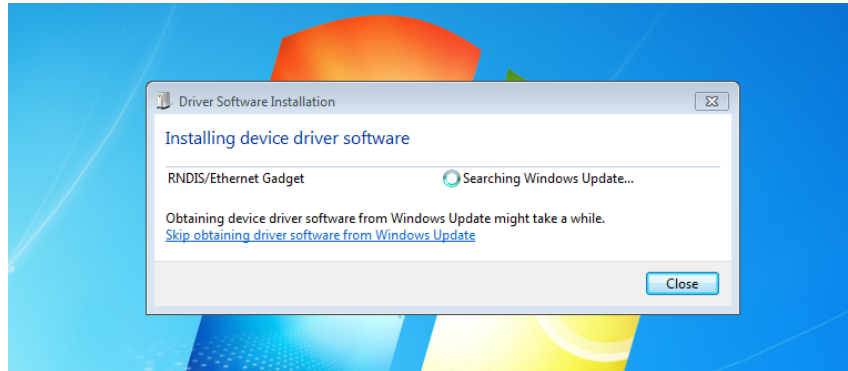
Once all Windows updates have been installed, double-click and open the obtained Ajile software package which is a .msi Windows installer (i.e. `ajile_installer_win64_xxx.exe`). This will open an installer program. Follow the prompts to install the Ajile GUI and Ajile SDK Driver to the Program Files directory into the Ajile directory (e.g. to `C:\Program Files (x86)\Ajile`). The installer also creates a shortcut in the start menu for the Ajile GUI.

#### Installing USB Ethernet Driver

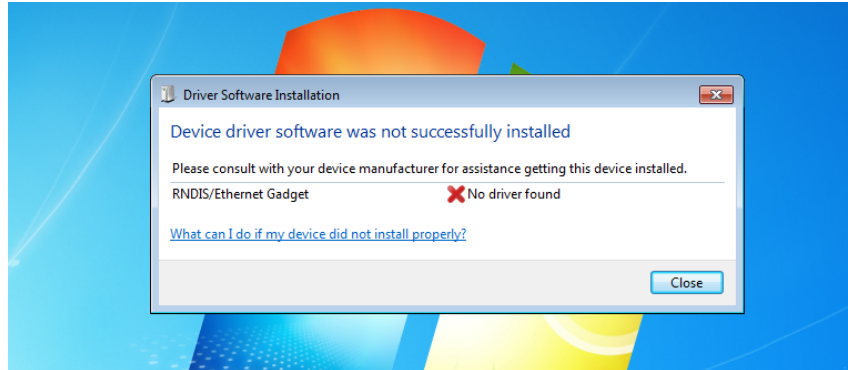
The Ajile software drivers make use of Ethernet over USB for connecting to USB 2 devices. As of release 1.0-4.9 of the Ajile firmware (released February 2018) Ajile devices include a DHCP server so that when connecting a USB cable to the device the PC will automatically get its IP address assigned and so no further configuration of the TCP settings is required.

The default USB Ethernet IP address of the device is **192.168.200.1**. To connect to the USB 2 Ajile device, follow these steps:

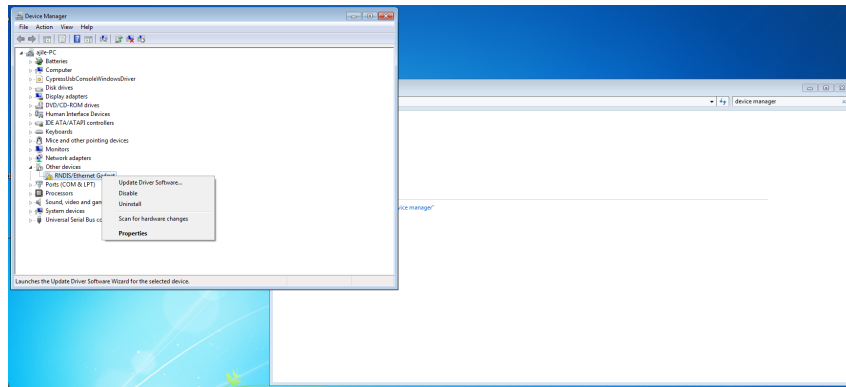
1. Power on the Ajile device by plugging in its power adapter and wait for it to initialize (you should see the LEDs on the device illuminated).
2. Connect the USB 2 cable from the PC into the Ajile device.
3. A Windows Driver Software Installation dialog may pop up indicating that it is installing device driver software, similar to the following:



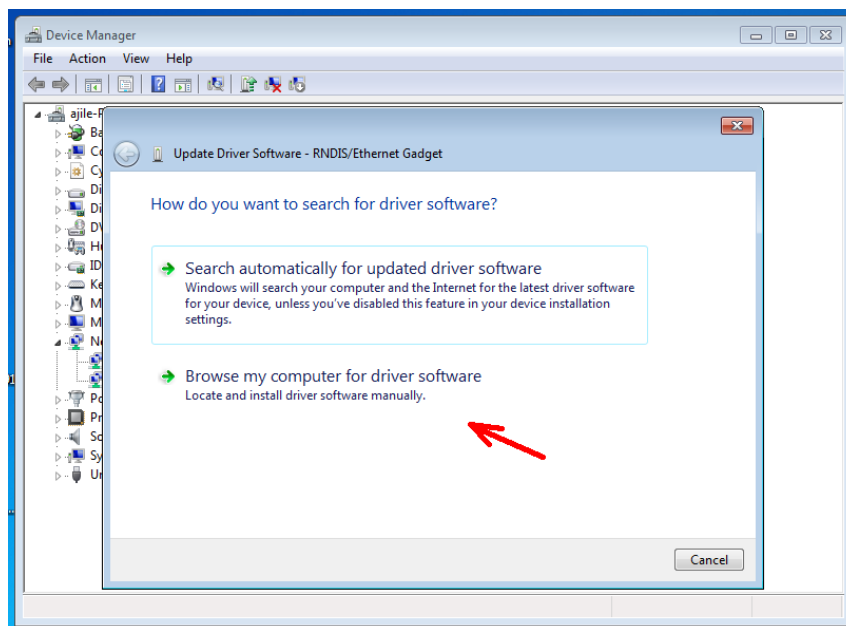
4. If the Driver Software Installation was successful, great, you can skip to step 11. Otherwise you may see something similar to the following if the automatic installation failed:



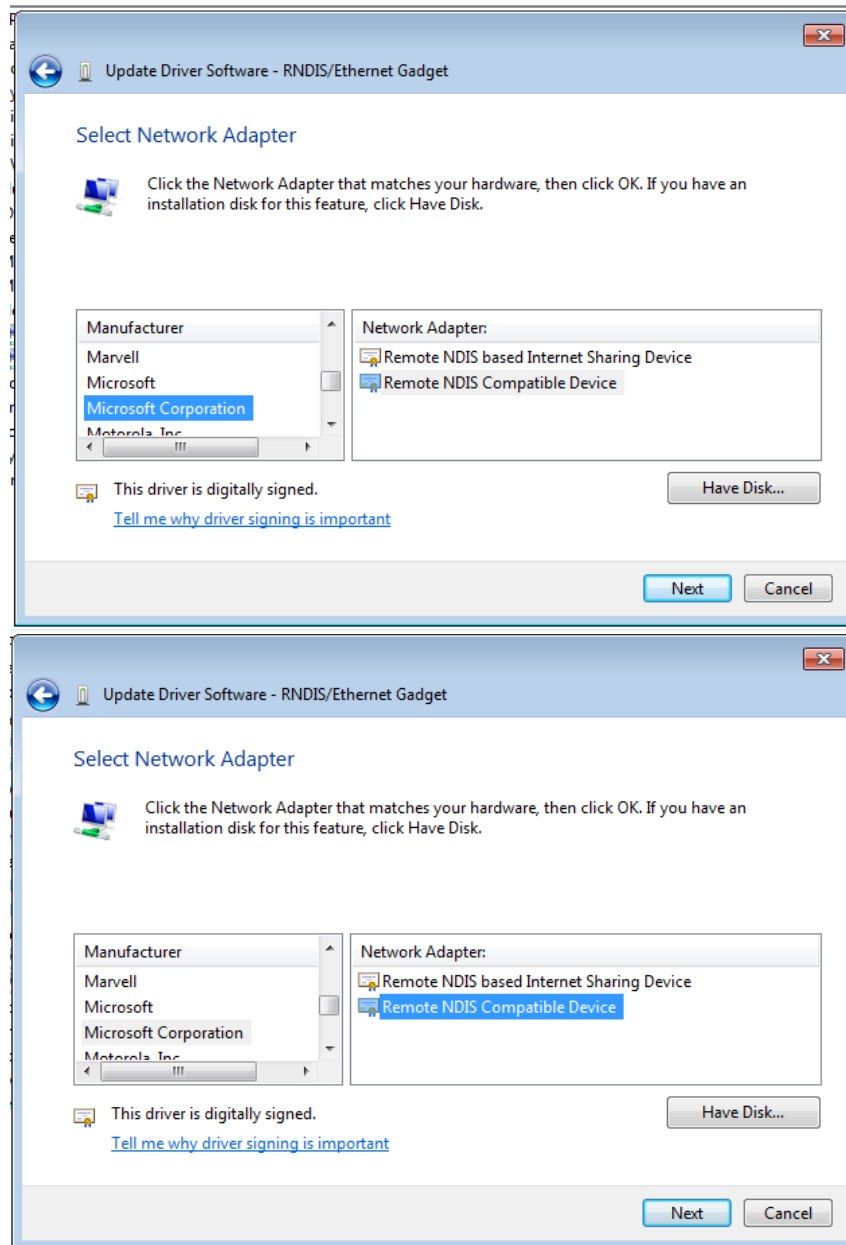
5. We need to manually install the USB Ethernet Gadget using the Windows Device Manager. Open the Device Manager (use Windows Search to find it).
6. Locate the RNDIS/Ethernet Gadget in the Device Manager, right click on it, and select Properties:



7. Select **Browse my computer for driver software** for how to search for driver software.



8. Select **Let me pick from a list....**
9. Under the Select Network Adapter dialog, select **Microsoft** as the Manufacturer and the **Remote NDIS Compatible Device** and the Network Adaptor, then click **Next**.



10. The driver should be successfully installed so we can close the Device Manager.
11. The USB 2 Ethernet Gadget installation should now be complete and you should be able to communication with the Ajile USB 2 device.

## Starting the GUI

To run the Ajile GUI, find the Ajile folder in the start menu and select the Ajile GUI application. Alternately, use the Windows search tool to search for Ajile GUI and select it from the Windows search. This will launch the Ajile GUI and it will be ready for use.

## Loading SDK Libraries

### Python Libraries

#### *Installing Python and NumPy*

To use the Ajile SDK for Python in Windows, you will first need to install Python and the NumPy package for Python. Currently supported versions of Python are Python 2.7.x and 3.x, 64-bit versions only. Download and install the latest 64-bit version of either Python 2.7.x or 3.x from <http://www.python.org>.

Next, the NumPy package must be installed in your Python distribution since it is used as the main tool to pass images between Python and the Ajile SDK. Details for installing NumPy can be found at <http://www.python.org>. The easiest way to install NumPy however is with Python pip tool. If your Python executable is installed at 'C:\Python\python.exe', then you can install NumPy from the command line terminal with:

```
C:\> c:\Python\python.exe -m pip install numpy
```

#### *Installing the Ajile Python Library*

With Python and NumPy installed, the Ajile Python libraries are located at 'C:\Program Files (x86)\Ajile\AjileDriver\lib\_python' by default for Python 2.7.x, and at 'C:\Program Files (x86)\Ajile\AjileDriver\lib\_python3' for Python 3.x. To use them with your existing Python2.7 (or Python3) installation, copy the files `_ajiledriver.pyd` and `ajiledriver.py` to your Python installation folder at %PYTHON\_HOME%\Lib\site-packages. Also copy all the .dll libraries from 'C:\Program Files (x86)\Ajile\AjileDriver\lib\_thirdparty' to %PYTHON\_HOME%\Lib\site-packages. These are third party libraries (Pthreads and OpenCV) which the AjileDriver uses and needs to load. Once this is done you should be able to load and use the ajiledriver from you Python programs. To test it try:

```
C:\> python
>>> import ajiledriver
>>> project = ajiledriver.Project("Test")
>>> print project.Name()
Test
```

If you get similar output and the name of the Project is printed then your Python installation was successful.

### C++ Libraries

The Ajile C++ libraries are located at 'C:\Program Files (x86)\Ajile\AjileDriver\lib' by default. In addition the include headers are located at 'C:\Program Files (x86)\Ajile\AjileDriver\include'. For building C++ applications using the Ajile SDK we use Visual Studio 2015. In addition we use the freely available CMake (<https://cmake.org/>) to generate Visual C++ projects that use the Ajile SDK. It is possible to use the include files and library files with other build tools, but we show an example here based on CMake and Visual C++ 2015.

To build with CMake we first need a CMakeLists.txt file. The following is a minimal CMakeLists.txt file which uses the AjileDriver.

```
CMAKE_MINIMUM_REQUIRED(VERSION 2.8)
PROJECT(project_test)
SET(AJILEDRIIVER_HOME "C:/Program Files (x86)/Ajile/AjileDriver/")
ADD_DEFINITIONS(-DWIN32_LEAN_AND_MEAN)
SET(SOURCE main.cpp)
INCLUDE_DIRECTORIES(
    ${CMAKE_CURRENT_BINARY_DIR}
    ${CMAKE_CURRENT_SOURCE_DIR}
    ${AJILEDRIIVER_HOME}/include)
ADD_EXECUTABLE(project_test ${SOURCE})
TARGET_LINK_LIBRARIES(
    project_test
    ${AJILEDRIIVER_HOME}/lib/ajiledriver.lib )
```

A minimal main.cpp source file to accompany it would be as follows:

```
#include <ajile/AJObjects.h>
#include <iostream>
void main() {
    aj::Project project("Test");
    std::cout << project.Name() << endl;
}
```

You can create the preceding CMakeLists.txt file and main.cpp file into the same directory, then run CMake and point the source directory where you created those files (i.e. run Configure followed by Generate, see Figure 1.1). You can then open the generated Visual Studio solution and build it. This will output an executable, project\_test.exe, into the Release or Debug directory. If you try to run the executable you will likely get an error indicating that the ajiledriver.dll library is missing. To fix this problem, copy the file 'C:\Program Files (x86)\Ajile\AjileDriver\lib\ajiledriver.dll' as well as all the .dll libraries from 'C:\Program Files (x86)\Ajile\AjileDriver\lib\_thirdparty' to the location of your executable (e.g. in Release or Debug). (Note that you can install the .dll libraries to a system directory instead of the executable location.) You should then be able to run the executable in a console window and see the name of the project displayed, see Figure 1.2 which shows a console window running the created program alongside a Windows Explorer window which shows all files in the same directory.

### 1.2.3 Software Installation in Ubuntu

#### Installing Prerequisites

The following prerequisite packages are required to install the Ajile Software Suite:

- libboost-serialization-dev
- libstdc++6
- libgcc1
- libc6
- libopencv-core2.4
- libopencv-highgui2.4
- libopencv-imgproc2.4
- libpython2.7
- python-pyside

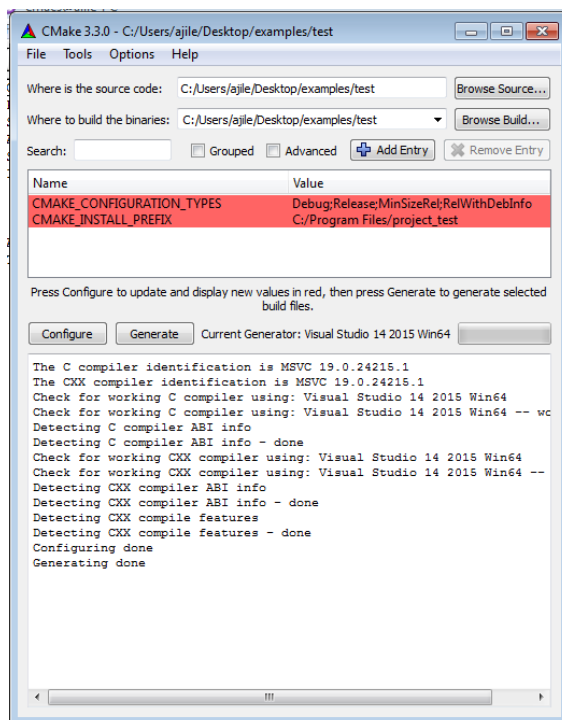


Figure 1.1: Screenshot of CMake output.

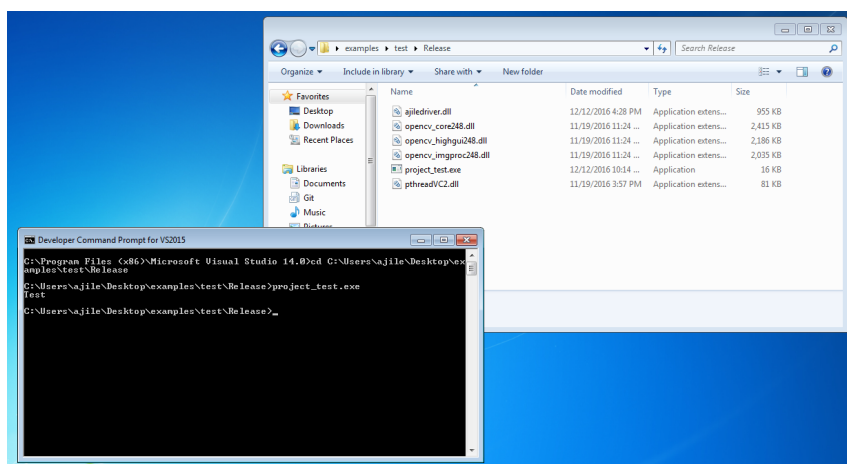


Figure 1.2: Screenshot of Running a test program.

- python-numpy
- python-opencv
- python-pil
- python-enum34

You can install the required prerequisites with the following commands:

```
sudo apt-get upgrade
sudo apt-get install libboost-serialization-dev libstdc++6 libgcc1 libc6 libopencv-core2.4
libopencv-highgui2.4 libopencv-imgproc2.4 libpython2.7 python-pyside python-numpy python-
opencv python-pil python-enum34
```

## Installing Ajile Software

To install the Ajile software in Ubuntu you will need to first extract the files from the .tar.gz archive which you downloaded. This will extract a number of .deb Debian packages in the same directory as the archive. Each of these packages needs to be run and installed, which will install the Ajile software into the proper system directories. The steps to install the Ajile software in Ubuntu is therefore as follows:

1. Open a new Terminal window to get a system command line interface. On Ubuntu systems this may be called Terminal in the Application → Accessories menu.
2. Change directories to the location where you downloaded the .tar.gz archive. For example if you download the file to the /Downloads directory, the command to type in the terminal is `cd /Downloads`.

3. Extract the files from the .tar.gz archive with

```
tar xvf ajile_suite_linux64_xxx_xxxx.tar.gz
```

where `xxx_xxxx` should be replaced with the characters in the file you just downloaded.

4. Install the Debian packages from the archive with the command

```
sudo dpkg -i libajiledriver_1.0-4.deb python-ajiledriver1.0-4.deb ajile-gui_1.0-4.deb
ajiledriver-doc_1.0-4.deb
```

Note that the version numbers of 1.0–4 may be different in your downloaded files, so please change them to match the file names in the archive that you downloaded.

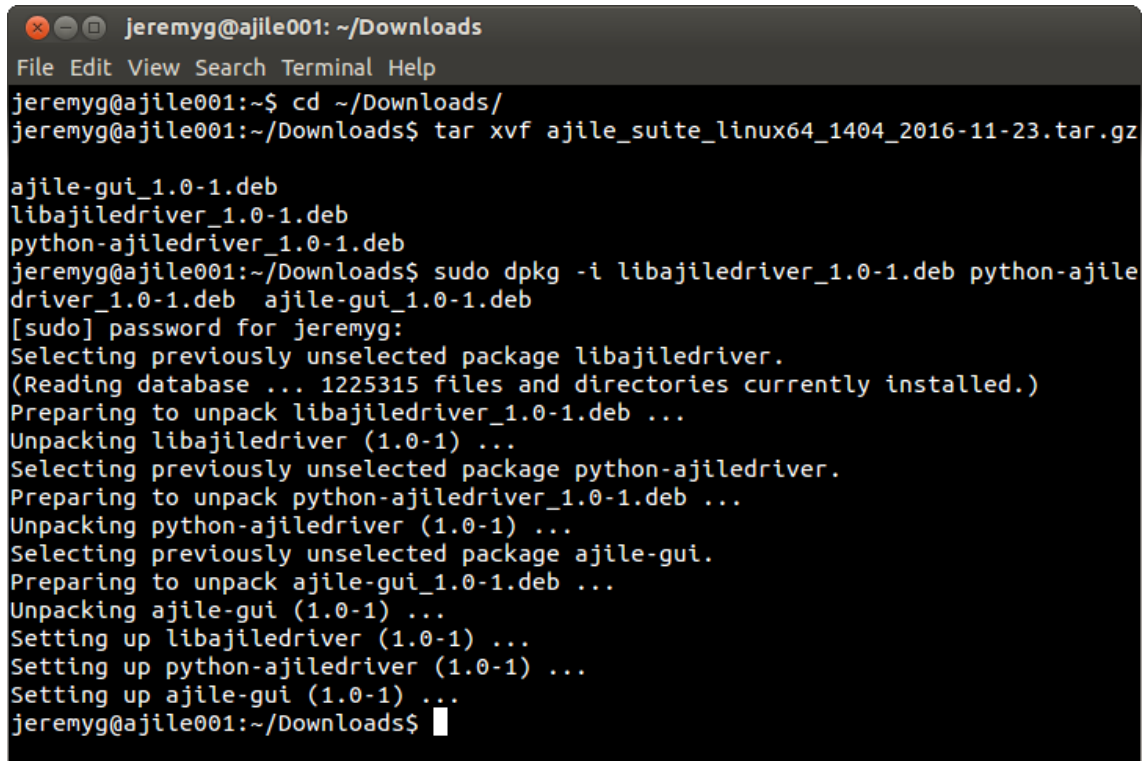
See Figure 1.3 for commands and the typical output that will be seen when installing the Debian packages from the command line. Once completed, the Ajile Driver will be installed with both the C++ and Python interfaces as well as the Ajile GUI.

## Installing USB Ethernet Driver

The Ajile software drivers make use of Ethernet over USB for connecting to USB 2 devices. As of release 1.0-4.9 of the Ajile firmware (released February 2018) Ajile devices include a DHCP server so that when connecting a USB cable to the device the PC will automatically get its IP address assigned and so no further configuration of the TCP settings is required.

The default USB Ethernet IP address of the device is **192.168.200.1**. To connect to the USB 2 Ajile device, follow these steps:

1. Power on the Ajile device by plugging in its power adapter and wait for it to initialize (you should see the LEDs on the device illuminated).
2. Connect the USB 2 cable from the PC into the Ajile device.



```
jeremyg@ajile001: ~/Downloads
File Edit View Search Terminal Help
jeremyg@ajile001:~$ cd ~/Downloads/
jeremyg@ajile001:~/Downloads$ tar xvf ajile_suite_linux64_1404_2016-11-23.tar.gz

ajile-gui_1.0-1.deb
libajiledriver_1.0-1.deb
python-ajiledriver_1.0-1.deb
jeremyg@ajile001:~/Downloads$ sudo dpkg -i libajiledriver_1.0-1.deb python-ajile
driver_1.0-1.deb ajile-gui_1.0-1.deb
[sudo] password for jeremyg:
Selecting previously unselected package libajiledriver.
(Reading database ... 1225315 files and directories currently installed.)
Preparing to unpack libajiledriver_1.0-1.deb ...
Unpacking libajiledriver (1.0-1) ...
Selecting previously unselected package python-ajiledriver.
Preparing to unpack python-ajiledriver_1.0-1.deb ...
Unpacking python-ajiledriver (1.0-1) ...
Selecting previously unselected package ajile-gui.
Preparing to unpack ajile-gui_1.0-1.deb ...
Unpacking ajile-gui (1.0-1) ...
Setting up libajiledriver (1.0-1) ...
Setting up python-ajiledriver (1.0-1) ...
Setting up ajile-gui (1.0-1) ...
jeremyg@ajile001:~/Downloads$
```

Figure 1.3: Screenshot of installing the Ajile software Debian packages.

3. The PC should automatically detect a new USB Ethernet Gadget connection and have a network interface connected which will be assigned an IP address by the Ajile device. To confirm this you may follow the remaining steps.
4. Open a new Terminal window to get a system command line interface.
5. Type the command `lsusb` in the Terminal window and press Enter to list the USB devices connected to the PC. One of the entries should read something similar to *Netchip Technology, Inc. Linux-USB Ethernet/RNDIS Gadget*, see Figure 1.4.
6. Type the command `ifconfig`. You should see a device such as `usb0` with an IP address of 192.168.200.20.

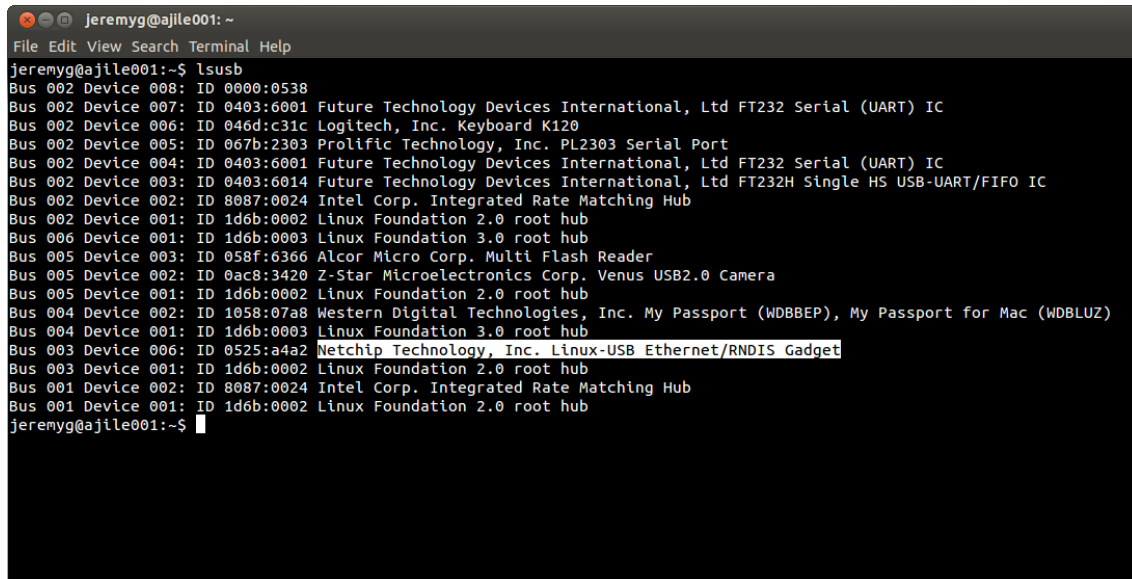
The USB Ethernet device should now be connected. If we type the command `'ping 192.168.200.1'` (the IP address of the Ajile device) then we should get a response.

## Starting the GUI

To start the Ajile GUI, enter the following command at the Terminal prompt:

```
ajile-gui
```

You can also start the Ajile GUI with Run Application dialog by pressing Alt-F2 in Ubuntu and entering the command `ajile-gui`.



```
jeremyg@ajile001:~  
File Edit View Search Terminal Help  
jeremyg@ajile001:~$ lsusb  
Bus 002 Device 008: ID 0000:0538  
Bus 002 Device 007: ID 0403:6001 Future Technology Devices International, Ltd FT232 Serial (UART) IC  
Bus 002 Device 006: ID 046d:c31c Logitech, Inc. Keyboard K120  
Bus 002 Device 005: ID 067b:2303 Prolific Technology, Inc. PL2303 Serial Port  
Bus 002 Device 004: ID 0403:6001 Future Technology Devices International, Ltd FT232 Serial (UART) IC  
Bus 002 Device 003: ID 0403:6014 Future Technology Devices International, Ltd FT232H Single HS USB-UART/FIFO IC  
Bus 002 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub  
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
Bus 006 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub  
Bus 005 Device 003: ID 058f:6366 Alcor Micro Corp. Multi Flash Reader  
Bus 005 Device 002: ID 0ac8:3420 Z-Star Microelectronics Corp. Venus USB2.0 Camera  
Bus 005 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
Bus 004 Device 002: ID 1058:07a8 Western Digital Technologies, Inc. My Passport (WDBBEP), My Passport for Mac (WDBLUZ)  
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub  
Bus 003 Device 006: ID 0525:a4a2 Netchip Technology, Inc. Linux-USB Ethernet/RNDIS Gadget  
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
Bus 001 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub  
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
jeremyg@ajile001:~$
```

Figure 1.4: Screenshot of listing the USB device from the terminal.

## Loading SDK Libraries

The Debian packages install the C++ Ajile driver library to `/usr/lib`, the C++ Ajile driver headers to `/usr/lib/ajile`, and the Python Ajile driver to `/usr/lib/python2.7/dist-packages`. On most systems these paths are automatically found by the C++ compiler (GDB) and by the Python 2.7 interpreter and you should be ready to start designing your own software using the Ajile SDK.

## 1.3 Running the Examples

The Ajile Software Suite installation comes with a number of example programs/projects so that you can get started right away with running your Ajile devices. Most of the examples can be run in the GUI, in Python and in C++. In this section we will see how to run the GUI examples, and how to install the Python/C++ examples to a suitable location and build/run them. For demonstration purposes we show how to load and run a DMD project which displays a generated checkerboard pattern and its inverse. Other examples can be run in the same way.

### 1.3.1 Running the Examples in the GUI

When you first open the Ajile GUI you will be presented with the 'Start' screen which shows a number of icons to get you started. Click on 'Example Projects' to open a new example project, see Figure 1.5.

Next, the list of available example projects will be shown in a new dialog box. The projects are categorized starting with the device type (DMD, Camera, etc.) then with their subcategories. For this tutorial select the 'DMD Binary Checkerboard Patterns' example then click Next, see Figure 1.6.

Clicking Next will bring up a New Project dialog. Select the AJD-4500 kit and enter any Project Name and Working Path (or leave them to the defaults) then click OK, see Figure 1.7. See Chapter 3 for more details on creating Projects and Chapter 4 for information on selecting Components (i.e. kits).

This will create a new project using the checkerboard pattern example project. You can view or even edit the project using the Ajile GUI. To run the checkerboard pattern sequence on the connected DMD device

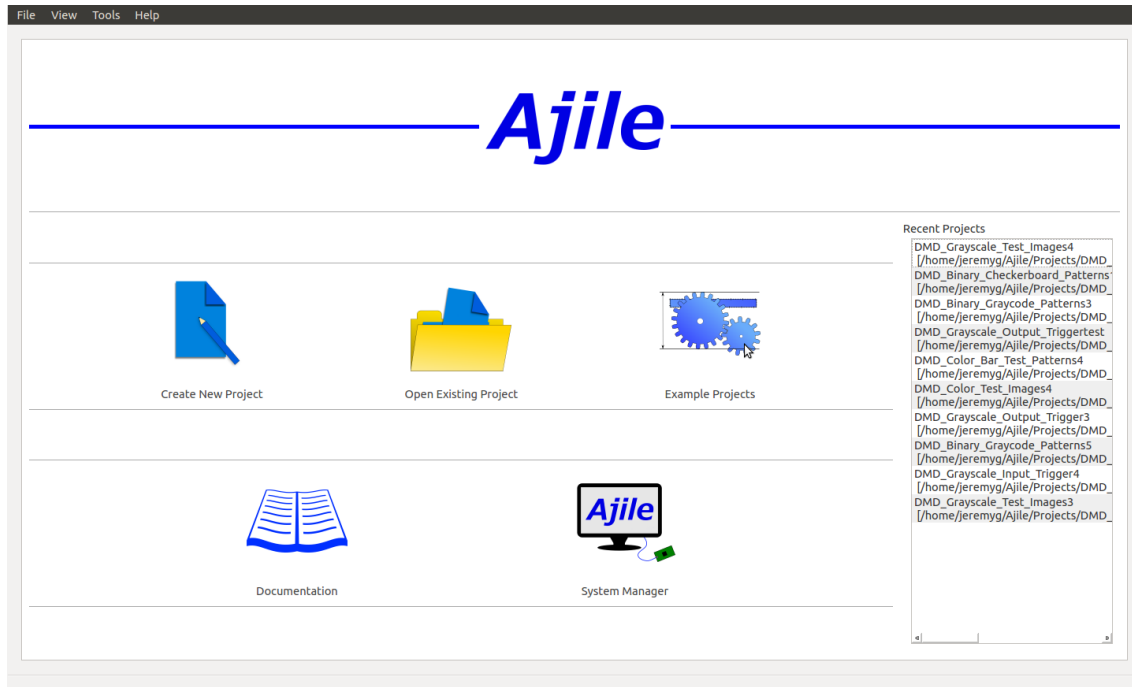


Figure 1.5: Screenshot of the Start Window, where we click on Example Projects to open the examples.

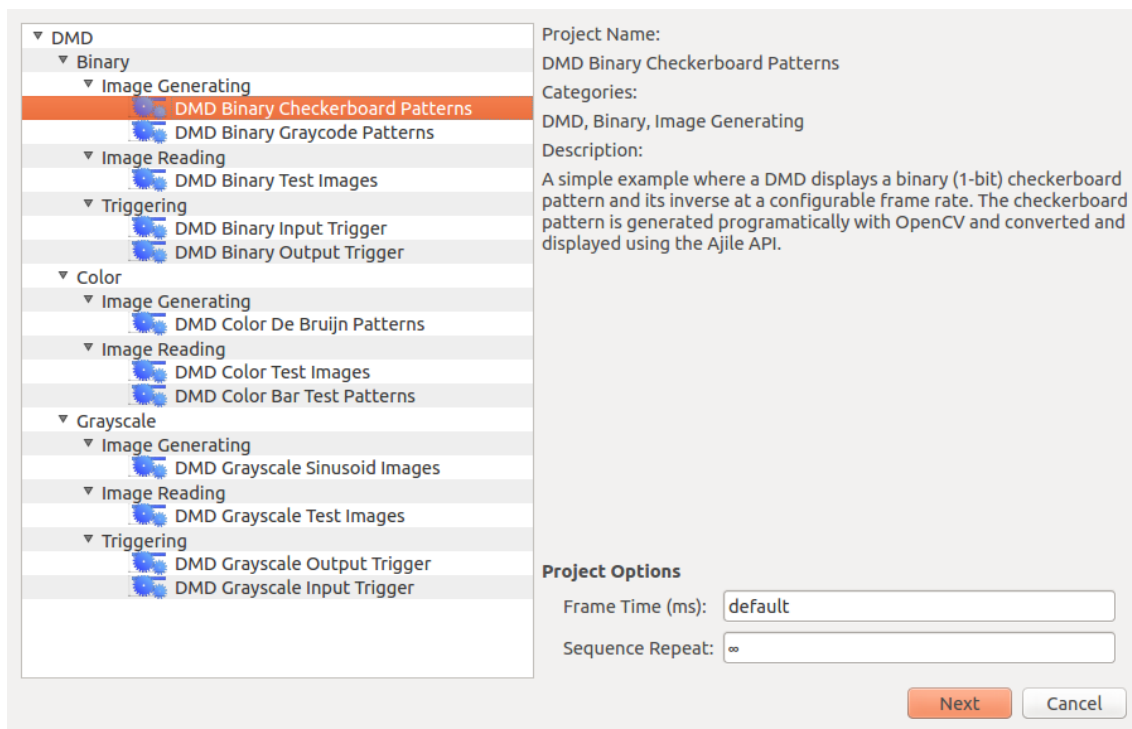


Figure 1.6: Screenshot of the Example Projects Dialog where we can select an example project to load.

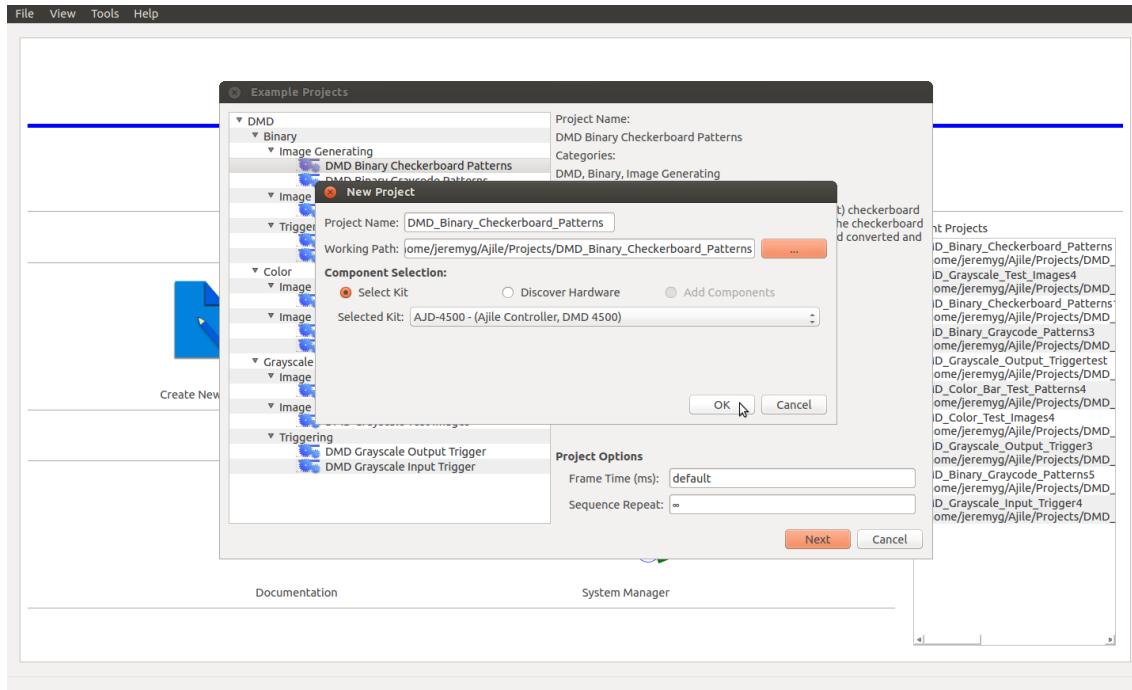


Figure 1.7: Screenshot of the creating a new Example Project.

we use the Run Environment. Click on the Run Environment button on the left Menu buttons to open in, see Figure 1.8, number 1. Next we must connect to the hardware device. This is done by clicking on the 'Connect to HW' button, see Figure 1.8, number 2. If successful you should see the connected components and their properties displayed. If not then you will likely need to configure your connection settings, see Chapter 9 for details.

When we are connected to the hardware device we load the project by clicking on the 'Load' button, see Figure 1.8, number 3. The load should be nearly instantaneous, then the 'Run' button will appear. Click the 'Run' button (see Figure 1.8, number 4) and the checkerboard pattern should be displayed by the device. Finally, click the 'Stop' button to stop the sequence from being run.

### 1.3.2 Running the Examples with the SDK

The Ajile SDK examples are available in both Python and C++. On Windows they are located at 'C:\Program Files (x86)\Ajile\Documentation\examples' and on Ubuntu Linux they are located at '/usr/share/doc/ajiledriver-doc/examples/'. The first step is to therefore copy the 'examples' directory to a location which is user-writable, e.g. on Windows copy 'C:\Program Files (x86)\Ajile\Documentation\examples' to your Desktop or on Linux copy '/usr/share/doc/ajiledriver-doc/examples/' to your home directory '~'.

#### Running the Examples in Python

Assuming that the Ajile Python libraries are installed correctly as above it is straightforward to run the Python examples. Open a Command Prompt (Windows) or Terminal (Linux) and change directories to the examples directory:

**On Windows:**

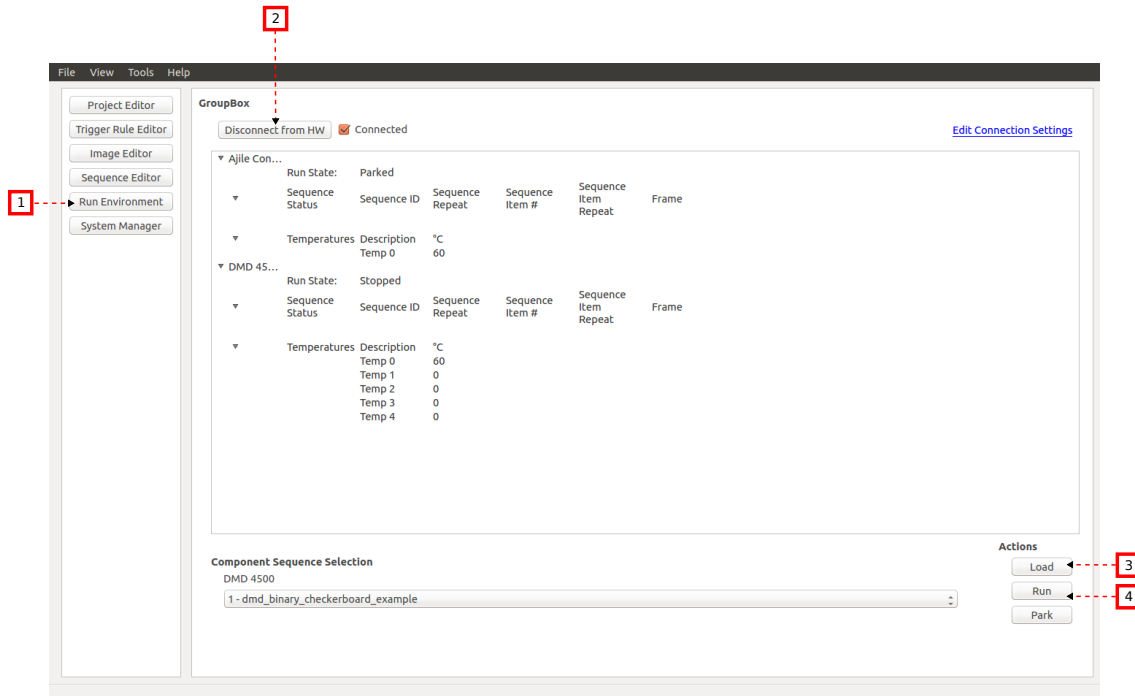


Figure 1.8: Screenshot of running an example project.

```
> cd C:\Users\ajile\Desktop\examples\dmd_binary_checkerboard\python
> python dmd_binary_checkerboard_example.py
```

#### On Linux:

```
$ cp -R /usr/share/doc/ajiledriver-doc/examples/ ~/
$ cd ~/examples/dmd_binary_checkerboard/python/
$ python dmd_binary_checkerboard_example.py
```

This should result in a checkerboard pattern being displayed by the connected DMD device.

#### Running the Examples with C++

Running the examples in C++ has one additional step over the Python examples which is that they must be compiled first. To do this we use CMake. Assuming that you have CMake installed in the system path you will run the following commands to generate the CMake project:

#### On Windows:

```
> cd C:\Users\ajile\Desktop\examples\dmd_binary_checkerboard\cpp
> cmake .
```

In Windows this creates a Visual Studio Solution .sln. Open the `dmd_binary_checkerboard_example.sln` file with Visual Studio and build it. If we build it in Release mode the executable will be located in the Release subfolder. Finally, to run the compiled example do the following:

```
> cd C:\Users\ajile\Desktop\examples\dmd_binary_checkerboard\cpp\Release
> dmd_binary_checkerboard_example.exe
```

If you get an error message which says that you have missing .dll files, copy the .dll files from 'C:\Program Files (x86)\Ajile\AjileDriver\lib' and 'C:\Program Files (x86)\Ajile\AjileDriver\lib\_thirdparty' to the Release directory, as was explained in Section 1.2.2.

**On Linux:**

```
> cd ~/examples/dmd_binary_checkerboard/cpp
> cmake .
```

In Linux this creates a Makefile. You can build the program then with make, then run the program:

```
> make
> ./dmd_binary_checkerboard_example
```

This should result in a checkerboard pattern being displayed by the connected DMD device.

## 1.4 Upgrading Ajile Software and Firmware

Ajile regularly releases new software and firmware versions which improve on performance, add new features and fix certain bugs from previous versions. For nearly all new software releases it will be required to first upgrade your device firmware before upgrading your PC software. The steps to upgrading your Ajile software are as follows:

1. Upgrade device firmware using currently installed PC software. For example, if software version 3 is installed on the PC and firmware release 3 is installed on the device and we want to upgrade the software/firmware to release 4, use the installed release 3 to first upgrade the device firmware, since it is compatible with the currently running firmware.
2. Restart Ajile device.
3. Uninstall currently installed software on the PC.
4. Install new software release (e.g. software release 4) on the PC. Connect to the device and run projects as usual.

The firmware and software upgrade steps are detailed below.

### 1.4.1 Upgrading Ajile Firmware

To upgrade the Ajile device firmware, first download the most recent firmware release from <http://ajile.ca/downloads>. The downloaded filename will be of the format BOOT\_DMD\_release\_X.ajb, where X is the release version number.

Next open the Ajile GUI and open the System Manager (which is available from the Start screen or from the Navigation side bar.) With the System Manager open, physically connect the Ajile device which will receive the firmware upgrade, and connect to the device with the 'Connect to HW' button (see Section 9.1 for details on connecting.) When you are connected to the device you will see a screen similar to Figure 1.9 where the currently connected device firmware version is shown.

With the device connected, select the downloaded firmware image (i.e. BOOT\_DMD\_release\_X.ajb) with the 'Select File' button. Finally, click the 'Update Software' button. The system will ensure that the selected firmware image is valid for the target device, then a dialog will appear to confirm whether or not to proceed with the firmware upgrade as in Figure 1.10. Click 'Yes' then a progress indicator bar will appear, shown in Figure 1.11. The firmware update can take around 2 minutes to complete so do not close the GUI or power off the device during this procedure. The progress percentage will increment during this update. If the progress percentage stays at 0% for 2 minutes or more then there was likely a communication error when sending the firmware image to the device. If this happens power cycle the device then reconnect and try again, or if using USB 3.0 try again but connecting using USB 2 instead.

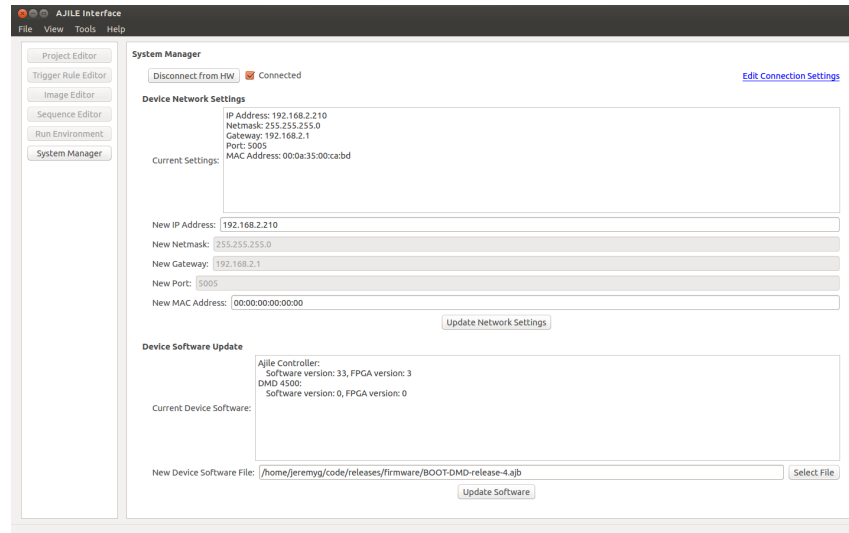


Figure 1.9: Screenshot of connecting with the System Manager.

If neither of these work then please contact Ajile support. Finally, when the firmware upgrade completes a new prompt will appear and you will be instructed to power cycle the device. Do so and proceed with the PC software upgrade detailed in the next section.

## 1.4.2 Upgrading Ajile Software

Upgrading the Ajile PC software involves first uninstalling the existing Ajile software from the system, then installing the newly downloaded software release (from <http://ajile.ca/downloads>) in the same way as described in Section 1.2.

To uninstall the Ajile software suite in Windows, use the ‘Add or remove programs’ Windows utility, select the ‘Ajile Software Suite’, and click the ‘Uninstall’ button (note that there may be minor variations depending on your Windows version). When the uninstall completes you can re-install the newly downloaded software release on the PC.

To uninstall the Ajile software suite in Ubuntu Linux we use the apt-get command from the command line. Open a Terminal window to get a system command line interface and enter the following command:

```
sudo apt-get remove ajiledriver-doc ajile-gui python-ajiledriver libajiledriver
```

When complete the Ajile software suite will be removed from the system, after which you can re-install the newly downloaded software release.

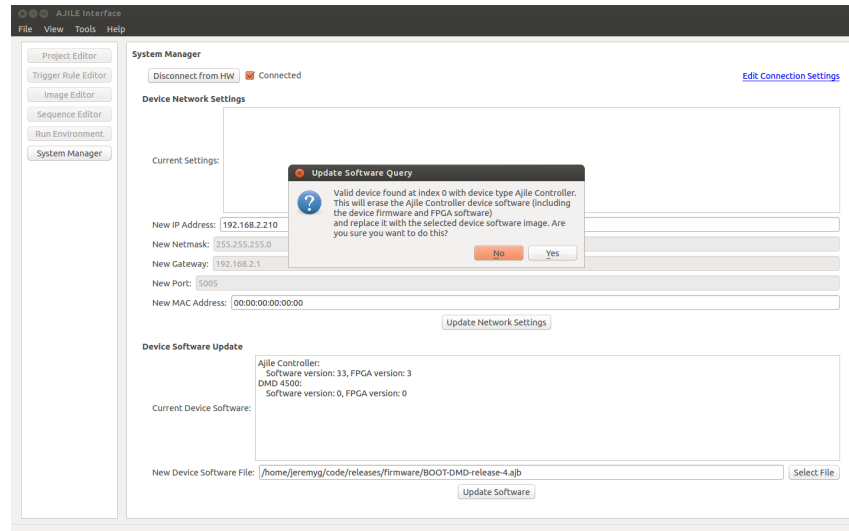


Figure 1.10: Screenshot of a valid firmware image file.

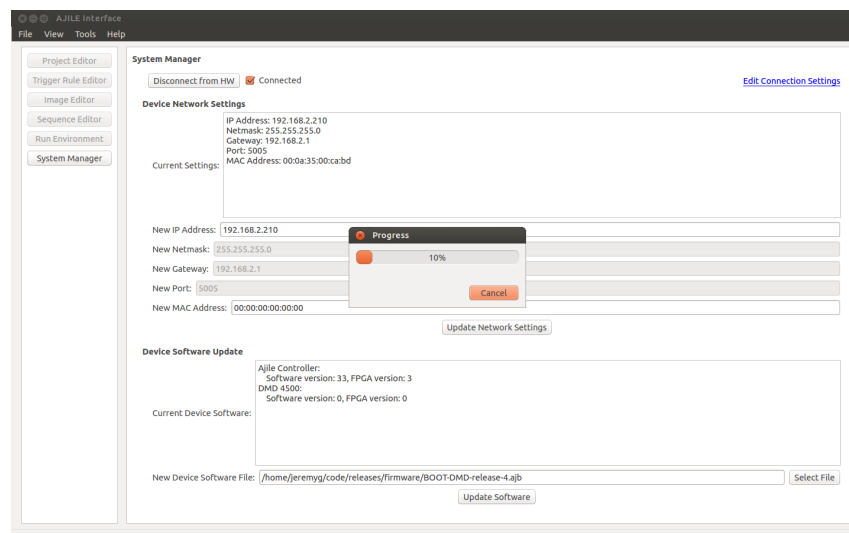


Figure 1.11: Screenshot of firmware upgrade progress.

Please visit <https://ajile.ca/support/>

and [log in](#) or [register](#)

to download full document text

