Myriad™ 2 Development Kit (MDK)

Getting Started Guide

v3.81 / June 2018

Copyright and Proprietary Information Notice

Intel® Movidius™
2200 Mission College Blvd
M/S SC11-201
Santa Clara, CA 95054
http://www.movidius.com/

Revision History

Date	Version	Description
June 2018	3.81	Updated section 2.3.1, Windows + Cygwin with note on Cygwin precedence.
April 2018	3.8	Added section 2.3.2.1, Pre-requisites and installation steps.
November 2017	3.7	Updated section 2.3.1, Windows + Cygwin.
October 2017	3.6	Updated section 2.1, Package contents.
August 2017	3.5	Added section 1.4, Documentation searching.

Table of Contents

1 Introduction

1.1 Overview

This document serves as an introduction to the Myriad™ Development Kit (MDK), and contains all the information necessary to get it up and running.

1.2 Target audience

The Getting Started is for customers of the MDK who intend to develop embedded applications using one of the Movidius® chips.

The MDK is a technical product requiring advanced knowledge about hardware and embedded firmware systems. Significant effort has been made to provide overviews, background information, and test applications. It is recommended that a typical user of this product have the following experience:

- Experience with C programming language, especially with the use of C pointers.
- Experience with embedded hardware architectures.
- Experience with embedded multi-threading OSes and/or interrupt driven, bare metal systems.
- Experience with the typical peripherals involved with the Computer Vision technology, like the cameras and the sensors.
- Experience with cross compilation development environments.
- Knowledge of Unix-like build systems helpful, but not required.

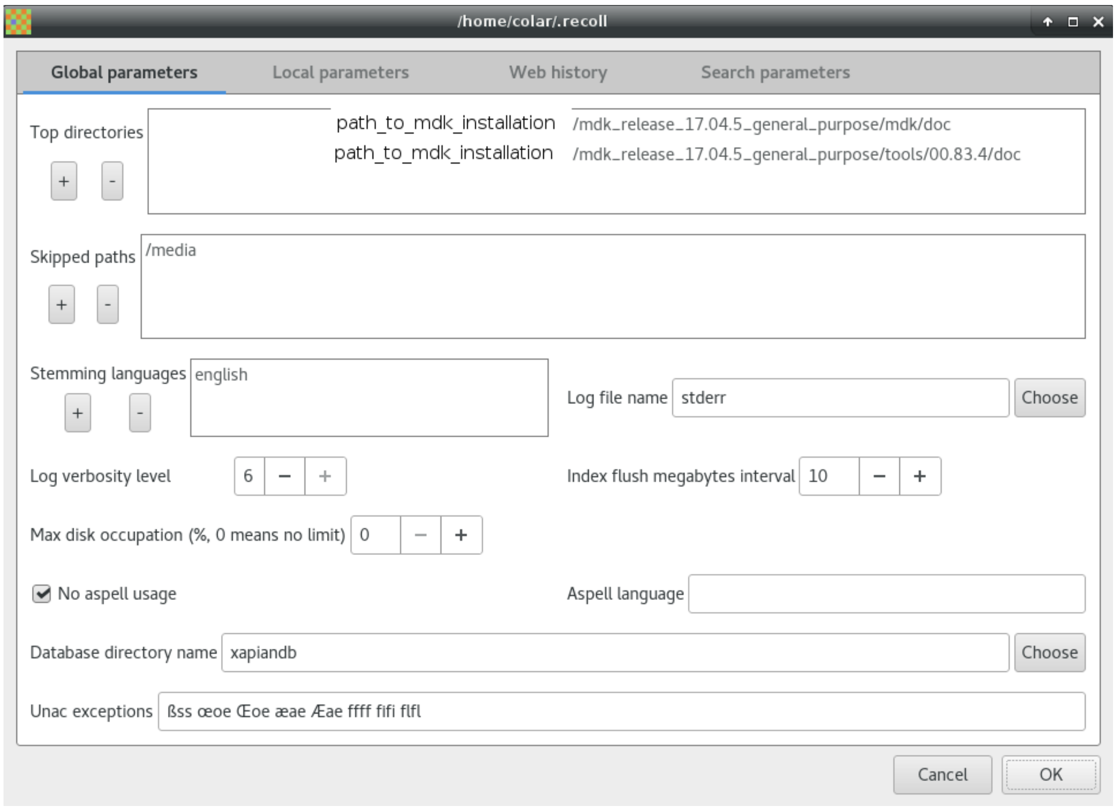
1.3 MDK structure

The following table shows some directories and their contents in the MDK Movidius development kit.

Directory	Contents
`./tools`	Movidius tools – compiler, linker, etc. `MV_TOOLS_DIR` should point here (white spaces not allowed)
`./mdk`	Root directory of software development environment
`./mdk/examples`	Simple example applications
`./mdk/common`	Directory holding drivers and building code
`./mdk/regression/unittest`	Unit test framework
`./mdk/packages`	MDK libraries
`./mdk/resources`	Data files required by examples and the unit test framework

1.4 Documentation searching

The MDK documentation is a collection of searchable PDF files present in the MDK archive. While users may use individual files for all their documentation searching needs, it has been observed that use of software performing offline documentation indexing is highly helpful in development. Good results were observed by internal users using Recoll (https://en.wikipedia.org/wiki/Recoll). Recoll is available for many operating systems and many Linux OS versions have it in their package managers. Just specify the paths to the folders containing the MDK documentation (see screenshot below) and you're ready to go.



Top directories: path_to_mdk_installation/mdk_release_17.04.5_general_purpose/mdk/doc
path_to_mdk_installation/mdk_release_17.04.5_general_purpose/tools/00.83.4/doc

Skipped paths: /media

Stemming languages: english

Log file name: stderr

Log verbosity level: 6

Index flush megabytes interval: 10

Max disk occupation (%, 0 means no limit): 0

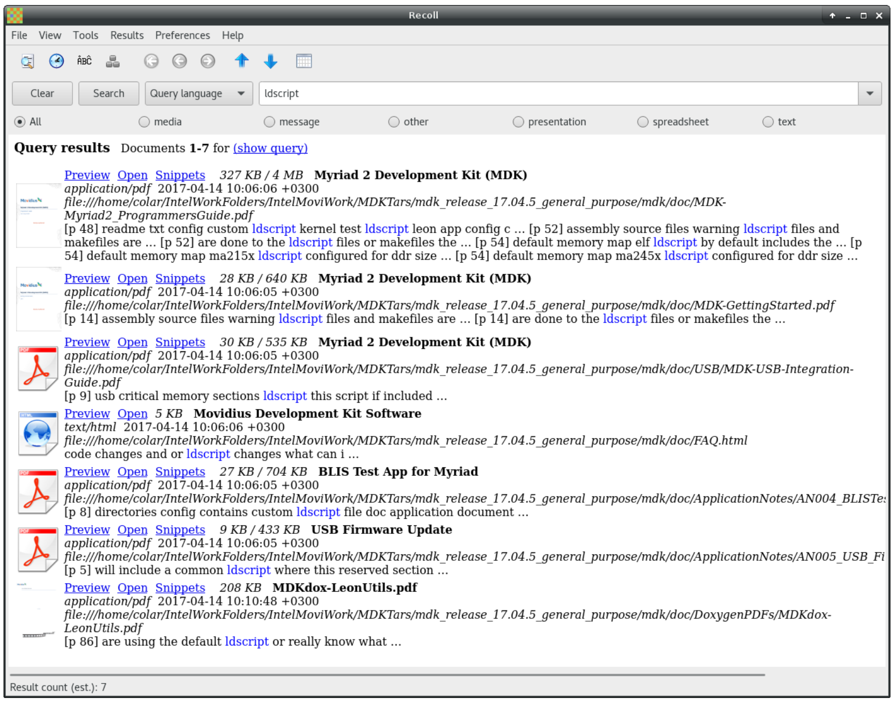
[✓] No aspell usage

Aspell language:

Database directory name: xapiandb

Unac exceptions: ßss œoe Œoe æae Æae ﬀff ﬁfi ﬂfl

The advantages brought by these types of indexing software are significant. For example, if searching for "ldscript" to find information regarding linker scripts usage in the MDK, we find links from all of the development documentation:

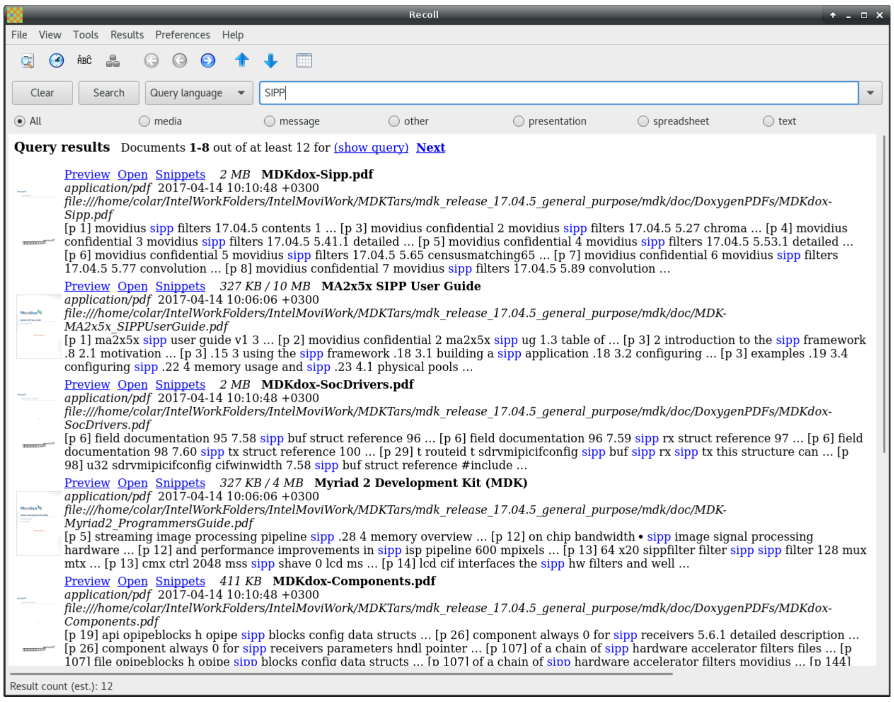


```
Query results  Documents 1-7 for (show query)

Preview Open Snippets  327 KB / 4 MB  Myriad 2 Development Kit (MDK)
application/pdf 2017-04-14 10:06:06 +0300
file:///home/colar/IntelWorkFolders/IntelMoviWork/MDKTars/mdk_release_17.04.5_general_purpose/mdk/doc/MDK-
Myriad2_ProgrammersGuide.pdf
[p 48] readme txt config custom ldscript kernel test ldscript leon app config c ... [p 52] assembly source files warning ldscript files and
makefiles are ... [p 52] are done to the ldscript files or makefiles the ... [p 54] default memory map elf ldscript by default includes the ... [p
54] default memory map ma215x ldscript configured for ddr size ... [p 54] default memory map ma245x ldscript configured for ddr size ...
Preview Open Snippets  28 KB / 640 KB  Myriad 2 Development Kit (MDK)
application/pdf 2017-04-14 10:06:05 +0300
file:///home/colar/IntelWorkFolders/IntelMoviWork/MDKTars/mdk_release_17.04.5_general_purpose/mdk/doc/MDK-GettingStarted.pdf
[p 14] assembly source files warning ldscript files and makefiles are ... [p 14] are done to the ldscript files or makefiles the ...
Preview Open Snippets  30 KB / 535 KB  Myriad 2 Development Kit (MDK)
application/pdf 2017-04-14 10:06:05 +0300
file:///home/colar/IntelWorkFolders/IntelMoviWork/MDKTars/mdk_release_17.04.5_general_purpose/mdk/doc/USB/MDK-USB-Integration-
Guide.pdf
[p 9] usb critical memory sections ldscript this script if included ...
Preview Open 5 KB  Movidius Development Kit Software
text/html 2017-04-14 10:06:06 +0300
file:///home/colar/IntelWorkFolders/IntelMoviWork/MDKTars/mdk_release_17.04.5_general_purpose/mdk/doc/FAQ.html
code changes and or ldscript changes what can i ...
Preview Open Snippets  27 KB / 704 KB  BLIS Test App for Myriad
application/pdf 2017-04-14 10:06:05 +0300
file:///home/colar/IntelWorkFolders/IntelMoviWork/MDKTars/mdk_release_17.04.5_general_purpose/mdk/doc/ApplicationNotes/AN004_BLISTe
[p 8] directories config contains custom ldscript file doc application document ...
Preview Open Snippets  9 KB / 433 KB  USB Firmware Update
application/pdf 2017-04-14 10:06:05 +0300
file:///home/colar/IntelWorkFolders/IntelMoviWork/MDKTars/mdk_release_17.04.5_general_purpose/mdk/doc/ApplicationNotes/AN005_USB_Fi
[p 5] will include a common ldscript where this reserved section ...
Preview Open Snippets  208 KB  MDKdox-LeonUtils.pdf
application/pdf 2017-04-14 10:10:48 +0300
file:///home/colar/IntelWorkFolders/IntelMoviWork/MDKTars/mdk_release_17.04.5_general_purpose/mdk/doc/DoxygenPDFs/MDKdox-
LeonUtils.pdf
[p 86] are using the default ldscript or really know what ...

Result count (est.): 7
```

Similarly, if searching for SIPP:



```
Query results  Documents 1-8 out of at least 12 for (show query)  Next

Preview Open Snippets  2 MB  MDKdox-Sipp.pdf
application/pdf 2017-04-14 10:10:48 +0300
file:///home/colar/IntelWorkFolders/IntelMoviWork/MDKTars/mdk_release_17.04.5_general_purpose/mdk/doc/DoxygenPDFs/MDKdox-
Sipp.pdf
[p 1] movidius sipp filters 17.04.5 contents 1 ... [p 3] movidius confidential 2 movidius sipp filters 17.04.5 5.27 chroma ... [p 4] movidius
confidential 3 movidius sipp filters 17.04.5 5.41.1 detailed ... [p 5] movidius confidential 4 movidius sipp filters 17.04.5 5.53.1 detailed ...
[p 6] movidius confidential 5 movidius sipp filters 17.04.5 5.65 censusmatching65 ... [p 7] movidius confidential 6 movidius sipp filters
17.04.5 5.77 convolution ... [p 8] movidius confidential 7 movidius sipp filters 17.04.5 5.89 convolution ...
Preview Open Snippets  327 KB / 10 MB  MA2x5x SIPP User Guide
application/pdf 2017-04-14 10:06:06 +0300
file:///home/colar/IntelWorkFolders/IntelMoviWork/MDKTars/mdk_release_17.04.5_general_purpose/mdk/doc/MDK-
MA2x5x_SIPPUserGuide.pdf
[p 1] ma2x5x sipp user guide v1 3 ... [p 2] movidius confidential 2 ma2x5x sipp ug 1.3 table of ... [p 3] 2 introduction to the sipp framework
.8 2.1 motivation ... [p 3] .15 3 using the sipp framework .18 3.1 building a sipp application .18 3.2 configuring ... [p 3] examples .19 3.4
configuring sipp .22 4 memory usage and sipp .23 4.1 physical pools ...
Preview Open Snippets  2 MB  MDKdox-SocDrivers.pdf
application/pdf 2017-04-14 10:10:48 +0300
file:///home/colar/IntelWorkFolders/IntelMoviWork/MDKTars/mdk_release_17.04.5_general_purpose/mdk/doc/DoxygenPDFs/MDKdox-
SocDrivers.pdf
[p 6] field documentation 95 7.58 sipp buf struct reference 96 ... [p 6] field documentation 96 7.59 sipp rx struct reference 97 ... [p 6] field
documentation 98 7.60 sipp tx struct reference 100 ... [p 29] t routeid t sdrvmipicifconfig sipp buf sipp rx sipp tx this structure can ... [p
98] u32 sdrvmipicifconfig cifwinwidth 7.58 sipp buf struct reference #include ...
Preview Open Snippets  327 KB / 4 MB  Myriad 2 Development Kit (MDK)
application/pdf 2017-04-14 10:06:06 +0300
file:///home/colar/IntelWorkFolders/IntelMoviWork/MDKTars/mdk_release_17.04.5_general_purpose/mdk/doc/MDK-
Myriad2_ProgrammersGuide.pdf
[p 5] streaming image processing pipeline sipp .28 4 memory overview ... [p 12] on chip bandwidth • sipp image signal processing
hardware ... [p 12] and performance improvements in sipp isp pipeline 600 mpixels ... [p 13] 64 x20 sippfilter filter sipp sipp filter 128 mux
mtx ... [p 13] cmx ctrl 2048 mss sipp shave 0 lcd ms ... [p 14] lcd cif interfaces the sipp hw filters and well ...
Preview Open Snippets  411 KB  MDKdox-Components.pdf
application/pdf 2017-04-14 10:10:48 +0300
file:///home/colar/IntelWorkFolders/IntelMoviWork/MDKTars/mdk_release_17.04.5_general_purpose/mdk/doc/DoxygenPDFs/MDKdox-
Components.pdf
[p 19] api opipeblocks h opipe sipp blocks config data structs ... [p 26] component always 0 for sipp receivers 5.6.1 detailed description ...
[p 26] component always 0 for sipp receivers parameters hndl pointer ... [p 107] of a chain of sipp hardware accelerator filters files ... [p
107] file opipeblocks h opipe sipp blocks config data structs ... [p 107] of a chain of sipp hardware accelerator filters movidius ... [p 144]

Result count (est.): 12
```

The results will show both drivers documentation from the doxygen generated files as well as documentation from the programmer guides related to SIPP files.

1.5 Other documents

1.5.1 Movidius.org

Other relevant documents may be downloaded from the Download section on www.movidius.org. Please register with the site first in order to get access to these. Registering may be done by accessing: https://www.movidius.org/authorisation/register.

1.5.2 Movidius documentation map

Please visit our documentation map to get an overview of the available MDK documentation with short descriptions and links:

./MDK-Software-Documentation.html

1.6 Supported hardware platforms

The MDK Software package offers support for the following platforms:

1. Myriad 2 SoC
   - Myriad 2 MA2150 Silicon on MV0182 Board (R4 E1+).
   - Myriad 2 MA2150 movisim.
   - Myriad 2 MA2450 Silicon on MV0212 Board (R0, R1).
   - Myriad 2 MA2450 movisim.

2. MV0182 Board
   - Revisions: R2, R3, R4 and R5.
   - For details about the differences between revisions R2, R3, R4 and R5, please see chapters 1.7-1.10 of the MV0182 User Manual (MV0182-UserManual.pdf).

3. MV0212 Board
   - Revisions: R0,R1.
   - For details about the R5 revision board, please see the MV0212 User Manual (MV0212-UserManual.pdf).

For an overview of the Myriad™ 2 product family, SDK please refer to the MDK Myriad™ 2 Programmer's Guide.

2 Quick start installation guide

2.1 Package contents

- Reference Myriad™ 2 Development Board (MV0212)
- Power supply
- Power Monitor Board (MV198)
- Dual Sony IMX208 2MP Sensor board (MV200xxx)
- Sony IMX378 12MP Sensor (MV0249)
- Olimex JTAG Debugger
- Cables
- Tripod
- Spudger
- QuickStart Installation Guide

2.2 MV0212 board notes

- When connecting the board power will turn on.
- It is possible to turn off the board both by unplugging the power supply or by pressing the B4 (“Off – 10s”) button for 10 seconds.
- If in a state where power is plugged but the board is turned off through the pressing of the B4 button, it is possible to turn it back on again by unplugging the power supply and plugging it back again.
- Before starting any debugging sessions, please make sure the Olimex JTAG dongle is firmly connected to the USB cable going to the PC as well as to the ribbon cable extending to the MV0212 board. Please make sure the ribbon cable from the JTAG dongle is also firmly pressed into the MV0212 JTAG socket JP4.
- Please follow the rest of the instructions in this QuickStart Guide to complete installation for various platforms.

2.3 Supported development platforms

2.3.1 Windows + Cygwin

Supported Windows platforms:

- Windows 7 (64-bit) with Cygwin (32-bit) version 2.9.0 and 32 & 64 bit version of Visual Studio 2015 Redistributable Package Update 3 RC.
- Windows 8 (64-bit) with Cygwin (32-bit) version 2.9.0 and 32 & 64 bit version of Visual Studio 2015 Redistributable Package Update 3 RC.

NOTE: Visual Studio 2015 Redistributable Package can be installed via 'vcredist_x86.exe' at https://www.microsoft.com/en-us/download/confirmation.aspx?id=52685. **Please select both 32 & 64 bit version of the redistributable.**

NOTE: Please note that the MDK build systems use commands that exist on both Windows and Cygwin. Cygwin commands should take precedence, and this can be achieved either by using the Cygwin shell or by having the Cygwin/bin path at the beginning of the Windows `path` variable.

Supported Cygwin Environment:

- Latest 32-bit Cygwin. **Please use 32-bit Cygwin v2.9.0** (http://www.cygwin.com/setup-x86.exe).
- Bash terminal. **Please use the bash terminal, not mintty or other flavors**. See FAQ at https://www.movidius.org/support/faqs/movidebug-wont-run.
- Mandatory Cygwin packages: `binutils`, `gcc`, `g++`, `sed`, `grep`, `awk`, `egrep`, `xargs`, `cut`, `sort`, `make`, `bash`. When installing, this means choosing "Install" for: Base, Devel, Shells, Python and interpreters.

WARNING: Relative to which mirror was chosen for installation, the default icon placed on the user's desktop might be mintty. To compensate for this you can just go into the Cygwin folder installation and run the `.bat` file that is in there. That starts a Windows command prompt with the Cygwin bash environment enabled which is similar to using the Cygwin bash terminal. It is advisable to replace your desktop created shortcut with a shortcut to this file. CTRL+SHIFT and dragging the said `.bat` file to desktop should be enough.

NOTE: If the mentioned `.bat` file does not exist, a Windows shortcut may be created to point to [Cygwin installation path]`\usr\bin\bash.exe` (or any other path where `bash.exe` was installed inside your Cygwin installation folder). After being created, its properties should be changed. Its target will probably be "[somepath]\`bash.exe`". This should be changed to "[somepath]\`bash.exe –login`".

2.3.2 Linux

Platforms supported:

- CentOS: Red Hat Enterprise Linux WS 5.3 and greater (64-bit).
- Ubuntu 16.04 LTS (64 bit) with ia32-libs package (recommended).
- Ubuntu 14.04 LTS (64 bit) with ia32-libs package.

2.3.2.1 Pre-requisites and installation steps

- Install GNU make and the GNU development tools.
- Install kconfig-frontends.
- Optional: Set-up **vim** to recognize the Next Generation Build System.

These steps are detailed below.

2.3.2.1.1 Install GNU make tools

The MDK build system is based on GNU make. GNU make is available in packages named something like `base-devel`.

2.3.2.1.2 Install `kconfig-frontends`

MDK uses a slightly modified version of `kconfig-frontends`. Accordingly, one is required not to use the version supplied with one's Linux distribution, but instead to use the following.

`kconfig-frontends` needs to be compiled and for this it uses GNU `autotools`, which should be installed using the packages of one's particular Linux distribution:

- **Ubuntu Linux**

  ```
  $ sudo apt-get install automake autoconf flex bison libncurses-dev libtool autoconf-archive
  ```

- **Fedora Linux**

  ```
  $ sudo dnf install automake autoconf flex bison ncurses-devel libtool qt-devel
  ```

- **Arch Linux**

  ```
  $ sudo pacman -Sy automake autoconf flex bison ncurses qt4
  ```

NOTE: The Qt packages are needed only if you plan to use the Qt version of the front-end.

Finally, clone and compile `kconfig-frontends`:

```
$ git clone https://github.com/movidius/kconfig-frontends.git
$ cd kconfig-frontends
# if you don't have a script called 'configure' then run this:
$ ./bootstrap
# then configure the frontends
$ ./configure
   # inspect ./configure output and if you find problems
   # then install the missing libraries, then redo ./configure until
no error
   #
   # look-up this line and confirm you've got your frontends:
   #
   # configure: - frontends          : conf mconf nconf qconf
$ make
$ sudo make install
$ # NOTE this will replace an existing kconfig-frontends on your
system
```

2.4 JTAG dongle driver installation

The JTAG dongle to be used should be `ARM-USB-TINY` (https://www.olimex.com/Products/ARM/JTAG/ARM-USB-TINY/).

NOTE: Some dongles may be called `ARM-USB-TINY-H`. These are also valid.

2.4.1 Linux

The drivers for the JTAG dongle are already included in this package, under the **tools** directory and they are automatically handled. So no need to install them from the ftdichip.com or olimex siteweb site.

In order to run the `moviDebugServer`, the user must be either logged in as root, or an additional rule needs to be inserted into the `/etc/udev/rules.d/` folder in order to allow other users access to the device.

In order to add this rule, create the file `/etc/udev/rules.d/93-olimex.rules` containing, for ARM-USB-TINY-H, the following:

```
SUBSYSTEM=="usb", ATTRS{idProduct}=="002a",
ATTRS{idVendor}=="15ba",GROUP="users", MODE="0666"
```

Make sure your user is also in the “users” group by checking:

```
groups [your user name]
```

Alternatively, use `GROUP=”[`Some other group your user is part of`]”` in the rules file.

NOTE: In case your dongle is ARM-USB-TINY, the rules are slightly different:

```
SUBSYSTEM=="usb", ATTRS{idProduct}=="0004",
ATTRS{idVendor}=="15ba",GROUP="users", MODE="0666"
```

2.4.2 Windows

For windows, installation guidelines are similar to generic FTDI devices which are found here:

https://www.olimex.com/Products/ARM/JTAG/ARM-USB-TINY/

The actual drivers to install are included in the release pack at the following location:

```
./tools/00.xx.yy.z/win32/drivers/install
```

NOTE: If switching from a ARM-USB-TINY dongle to a ARM-USB-TINY-H dongle on the same PC, in Windows, the existing FTI drivers should be uninstalled first and then, when the ARM-USB-TINY-H dongle is connected the user should go again through the driver installation process (in some cases Windows will do that automatically).

NOTE: **Windows 8**: The Olimex JTAG driver is not digitally signed, so by default Windows prevents the installation process for security reasons. The drivers is safe to install, so in order to do that the driver signature verification needs to be disabled during the installation as described here.

2.5 Installation procedure

Unzip the release archive.

- **Windows:**
  Set the tools environment variable `MV_TOOLS_DIR` (white spaces not allowed) to point to the tools

directory. For example: "`MV_TOOLS_DIR = C:\WORK\Movidius\mdk\tools`" if the archive was extracted to "`C:\WORK\Movidius`"

NOTES:

- The variable points to the directory that contains the tool version subdirectory
- When using Cygwin, there are two further rules:
  - this must be defined as a DOS path
  - if the DOS drive is "`X:\...`" then the `X` volume must be mounted as `/cygdrive/x` in Cygwin

- **Linux:**
  For setting the Linux `MV_TOOLS_DIR` (white spaces not allowed) environment variable, please refer to your own Linux distribution environment variable setting instructions. Common ways of setting these environment variables are using either .profile or .bashrc from your user's home folder, but please check your distribution's instruction for a definite answer on the best way to set this.

If using Windows, ensure JTAG dongle drivers are installed correctly (see previous section).

Make sure the JTAG parallel cable is connected properly both in the JTAG dongle and in the MV0182 or MV0212 board connector, and the keys are correct.

Ensure that `/tmp` exists and is writable.

Check the installation by typing:

```
cd ./mdk/common/utils
bash ./checkinstall.sh
```

To proceed further, the message displayed should be: **"Everything should be all right with your MDK installation."**

Ensure that the board is correctly powered and debug cables correctly attached (See MV0182 or MV0212 development board User manual).

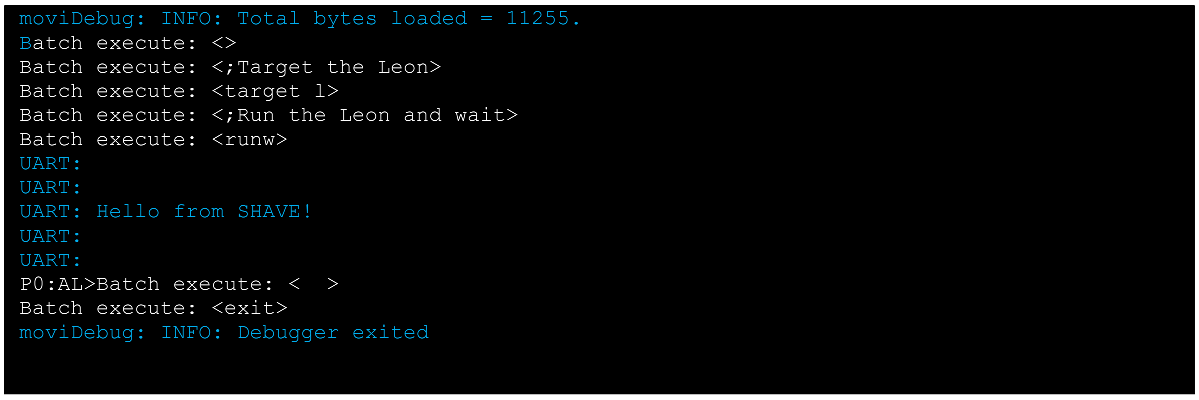
Open a terminal window where MDK was extracted and type the following commands:

```
cd ./mdk/examples/Progressive/000_HelloWorld_BareMetal
make start_server
```

Open a new terminal window where MDK was extracted and type the following commands:

```
cd ./mdk/examples/Progressive/000_HelloWorld_BareMetal
make all run
```

Check that you observe the following output:



```
moviDebug: INFO: Total bytes loaded = 11255.
Batch execute: <>
Batch execute: <;Target the Leon>
Batch execute: <target l>
Batch execute: <;Run the Leon and wait>
Batch execute: <runw>
UART:
UART:
UART: Hello from SHAVE!
UART:
UART:
P0:AL>Batch execute: <  >
Batch execute: <exit>
moviDebug: INFO: Debugger exited
```

NOTES:

1. If using dual boot Windows/Linux systems and sharing the MDK installation or if using virtual machines for the Linux machine, the partitions mounted with the MDK installation need to be mounted using the

"exec" option. Please refer to the manual of mount and of `fstab`.

2. If there are any problems running `moviDebugServer` or `moviDebug` in either Windows or Linux, please refer to the Movidius.org FAQ section: https://www.movidius.org/support/faqs/category/sample-faqs.

   Note for internal Movidius users: the sample FAQ link does not work if your user account type is support or admin type. Please just browse to the FAQ section from the main login page and check there.

2.6 Using Eclipse as IDE

Movidius also provides an Eclipse based IDE named moviEclipse that can be used to develop Myriad™ applications. A copy of the IDE can be downloaded from the following URL:

https://www.movidius.org/downloads/eclipse-bundle

There two versions available, one for Windows 64-bit and one for Linux 64-bit. Download the one that matches your OS and unpack it into a folder of your choice.

To function properly, moviEclipse needs a system with at least the following characteristics:

CPU	2.00GHz dual core processor.
RAM	2GB free memory.
DISK	1GB free disk space.
OS	Windows 7 64-bit.
	Linux 64-bit with GTK+ 2.18.0 or GTK+3.0.0 or greater.
Additional software	Latest Myriad™ MDK and tools in working condition. See previous chapters for details.

NOTE: **Windows + Cygwin**: Please make sure you add your Cygwin's bin directory ([Cygwin installation path]\bin) at the beginning to your system's "Path" environment variable. Doing so make it possible to use the cygwin shell commands in the Windows command line, which is needed for the moviEclipse IDE.

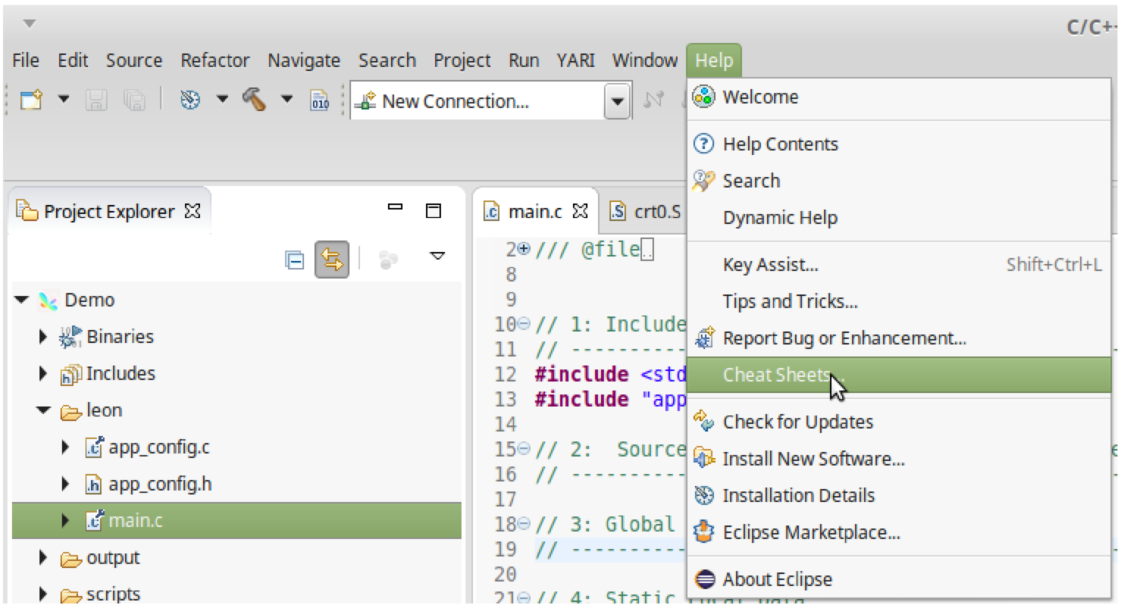
2.6.1 Using moviEclipse for the first time

If you are new to moviEclipse you should first follow the provided cheat sheet pages.

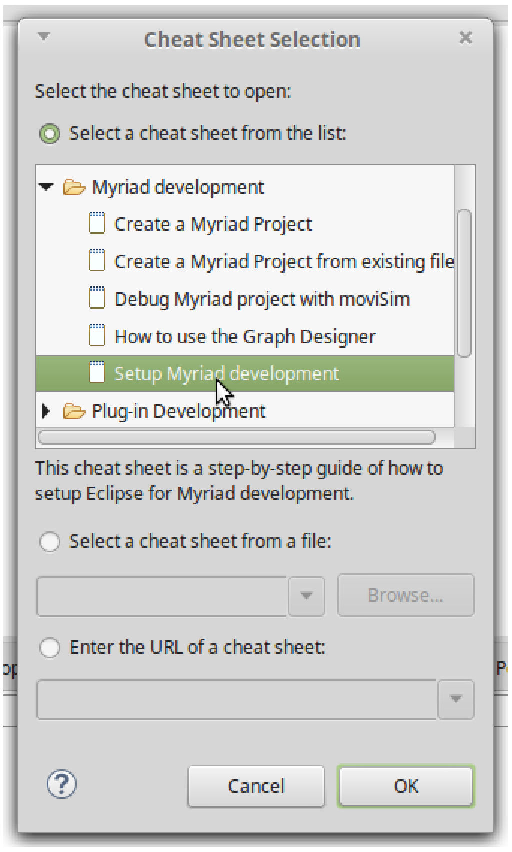
Currently, several cheat sheets are provided:

- Setup the Myriad environment.
- Creating Myriad projects.
- Debugging Myriad projects.
- Using the Graph Designer.

The cheat sheets can be found into the Help menu as follows:



Selecting the menu item shown in the image above will bring up the cheat sheet window, where the desired cheat sheet can be selected, from the Myriad™ development group:



Cheat Sheet Selection

Select the cheat sheet to open:

[✓ Select a cheat sheet from the list:]

- Myriad development
  - Create a Myriad Project
  - Create a Myriad Project from existing file
  - Debug Myriad project with moviSim
  - How to use the Graph Designer
  - Setup Myriad development
- Plug-in Development

This cheat sheet is a step-by-step guide of how to setup Eclipse for Myriad development.

Select a cheat sheet from a file:

Enter the URL of a cheat sheet:

After selecting the desired cheat sheet, a step by step walk-through guide will open that will help you to perform the steps described. We recommend to start with the “Setup Myriad development” cheat sheet.

2.6.2 Additional help

MoviEclipse includes a comprehensive help which can be found by executing the menu item “Help \ Help contents”. It includes not only the help relevant for Myriad™ development but also, for example, the help provided by Eclipse for general Eclipse usage and C/C++ development. If you are new to Eclipse please take your time to browse the provided help to see what's available.

3 Software overview

3.1 Introduction

The MDK comprises generic source codes which includes both drivers and components, and some example applications. This section provides a brief description of the example applications and the re-usable components included in the MDK.

3.2 Example Applications

Sample applications are included in MDK release pack to give developers a starting point for programming Myriad™ platform. The best starting point for detailed overview of examples is:

```
mdk/examples/examples.html
```

There are five categories of examples: Demo, HowTo, Progressive and SIPP. Find below a short description of them. Each example's folder contains a `Readme.txt` with detailed description.

Directory	Type of example
`Examples/Progressive`	The “progressive” category contains examples provided for the purpose of training. They are numbered in the recommended order to be examined. They evolve from simple to complex.
`Examples/HowTo`	This category features examples that show how to achieve various MDK software development purposes. There are specific subgroups of examples showing RTEMs, C++, and DragonBoard™ integration.
`Examples/Demo`	This category features different demo applications from Movidius with various functionality.
`Examples/Sipp/Opipe`	This category contains Opipe software component based examples which use the SIPP hardware underneath. For more information about Opipe please refer to MDK Programmer's Guide.
`Examples/Sipp/SippFw`	This category contains SIPP Framework based examples which use the SIPP hardware underneath. For more information about SIPP Framework please refer to the MA21x5x SIPP User Guides.
`Examples/Evaluation`	This category contains a set of evaluation examples highlighting the key

Directory	Type of example
	features and characteristics of the Myriad™ 2 platform.

Within the above categories the examples are organized into sub directories based on which Myriad 2 platform they support:

- Examples in the root folder of the `Progressive/HowTo/Demo/Sipp` categories are the cross platform examples, which can be built and run on all available Myriad™ 2 platforms. These examples are defaulted for MA2150, please refer to the examples' Readme.txt regarding how to build them to another platform.
- Examples placed under a specific ma2150/ma2450/ma2x5x directory indicates that these examples supports only this specific platform.

3.3 Make targets

3.3.1 make help

Displays a help message containing a short description of all `make` targets.

3.3.2 make all

Used to build a target after changes were done to C or assembly source files.

WARNING: `ldscript` files and `Makefiles` are not covered by "`make all`". If changes are done to the `ldscript` files or `Makefiles`, the build needs to be cleaned first before running "`make all`".

3.3.3 make run

Build everything and run it on a target via `moviDebug`. `MoviDebugServer` needs to be started previously for this target to work correctly. Alternatively, `moviSim` may also be started instead of `moviDebugServer` if the user wishes to run on the simulator using `moviDebug`.

3.3.4 make start_server

Starts `moviDebugServer` required to run `moviDebug`. This may alternatively be called with: "`make srv`".

3.3.5 make start_simulator

Starts `moviSim` in quiet mode for testing with the simulator.

3.3.6 make start_simulator_full

Starts `moviSim` in verbose mode (full log). Recommended usage: make `start_simulator_full > results.log`. Traps all verbose log in "`results.log`" file.

3.3.7 make debugi

Starts `moviDebug` for interactive use.

3.3.8 make debug

This debug target differs from the run target in that it strips out any call to exit in the run script. This allows the user to do interactive debugging once the execution has stopped or alternatively to hit CTRC+C during the `runw` session and to start checking the state of execution.

3.3.9 make report

Creates a graphical report of memory utilization. The output is an html file which can be opened in any browser. Application has to be built previously.

3.3.10 make clean

Clean build files.

3.3.11 make load

Builds application and loads target elf into debugger but doesn't start execution (allows setting breakpoints in advance).

3.3.12 make flash

Program SPI flash of the MV0182 and MV0212 boards with your current `mvcmd`. For more details regarding how to flashing an application please refer to section 3.6.

3.3.13 make flash_erase

Erase SPI flash of the MV0182 and MV0212 boards. For more details regarding how to flashing an application please refer to section 3.6.

3.3.14 make show_tools

Displays tools paths.

3.4 Drivers

Drivers are located under the mdk/common/drivers/myriad2 directory and grouped as follows:

- brdDrivers – Drivers for specific boards, e.g. MV0182, MV0212.
- icDrivers – Drivers for specific IC on a board.
- socDrivers – Drivers for peripherals and HW blocks on the Myriad 2 SoC.
- system – Base system routines, e.g. crt0.S, trap handler.

The drivers are further grouped based on if they are bare-metal or OS (RTEMS) drivers.

Drivers are automatically included in the build system, so no extra makefile inclusion is needed to use them.

3.5 Components

Components are located under the mdk/common/components directory and selectively included in projects through makefile. For detailed description of each components see the header file comments within each component.

Component name	Short description
`Board182`	Basic PCB HGL functionality.
`JpegM2Encoder`	JPEG encoder functionality.
`camGeneric`	Generic Camera driver.
`PatternGenerator`	Pattern generator component.
`sampleProfiler`	Sampling profiler component.
`LeonIPC`	LeonOS – LeonRT inter processor messaging.
`BLIS`	Shave BLIS Linear Algebra library.
`imageWarp`	Shave image warping algorithm.
`KernelLib/MvCV`	Movidius Computer Vision kernel library.
`KernelLib/MvISP`	Movidius ISP kernel library.
`KernelLib/LAMA`	Movidius Liner Algebra kernel library.
`Opipe`	MA2x5x SIPP HW based ISP pipeline component.

3.6 Flashing an Application

The make system has a built-in utility for flashing an application. Running ‘`make flash`’ in an application directory causes that application to be flashed so that it may be run without a debugger.

Points to note:

- In order to boot from flash make sure the boot configuration DIP switches of the MV0182 and MV0212 boards are set to the onboard SPI Flash device, which is the default configuration of the board (Switch1..8: 00001100). Switch2 is the BOOT_CONTROL switch which must be low even for programming the SPI Flash. For more details regarding boot configuration please refer to MV0182 and MV0212 User Manuals.
- Do not have any `printf` in the application to be flashed. Building using ‘`make clean all NO_PRINT=YES`’ ensures this is the case. Alternatively, `-DNO_PRINT` as a C option in the Makefile can be used.
- Flashing erases the entire flash, and then program minimum number of blocks required to write the desired image. The image is then read back and validated.
- ‘`make flash_erase`’ erases the flash. Recommended prior to interactive debugging.