



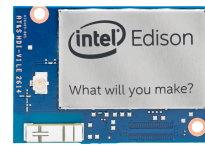
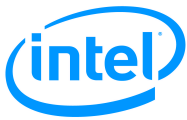
Intel[®] Edison Tutorial: Introduction to OPKG



Table of Contents

Introduction.....	3
List of Required Materials and Equipment.....	3
OPKG.....	4

Revision history		
Version	Date	Comment
1.0	6/14/2016	Modified old tutorial version



Introduction

In this tutorial, you will:

1. Learn about OPKG and package managers

List of Required Materials and Equipment

1. 1x Intel Edison Kit
2. 2x USB 2.0 A-Male to Micro-B Cable (micro USB cable)
3. 1x powered USB hub **OR** an external power supply
4. 1x personal computer
5. Access to a network with an internet connection



OPKG

To quote Wikipedia “A package manager or package management system is a collection of software tools that automates the process of installing, upgrading, configuring, and removing computer programs for a computer's operating system in a consistent manner. A package manager deals with packages, distributions of software and data in archive files. Packages contain metadata, such as the software's name, description of its purpose, version number, vendor, checksum, and a list of dependencies necessary for the software to run properly. Upon installation, metadata is stored in a local package database. Package managers typically maintain a database of software dependencies and version information to prevent software mismatches and missing prerequisites. They work closely with software repositories, binary repository managers, and app stores.

Package managers are designed to eliminate the need for manual installs and updates. This can be particularly useful for large enterprises whose operating systems are based on Linux and other Unix-like systems, typically consisting of hundreds or even tens of thousands of distinct software packages.”

If you have used Ubuntu before, you may be familiar with **apt-get**. The Yocto image does not have **apt-get** but instead has something called **opkg** which stands for **Open PackaGe Management**. The two package management systems work similarly, but **opkg** may not have some programs that **apt-get** has, and vice-versa.

In this tutorial, we will use **opkg** to install **git** and then use **git** to download an online code repository.

1. Access the shell on your Intel Edison. For more information, please refer to the document labelled *Intel Edison Tutorial – Introduction, Shell Access and SFTP*.
2. Print out all available packages (installed or not). Press **ctrl-C** to stop.
\$ opkg list

```
root@edison:~/tutorial3_examples# opkg list
alsa-conf-base - 1.0.27.2-r0
alsa-lib - 1.0.27.2-r0
alsa-states - 0.2.0-r4
alsa-utils-alsactl - 1.0.27.2-r0
alsa-utils-alsamixer - 1.0.27.2-r0
alsa-utils-amixer - 1.0.27.2-r0
alsa-utils-aplay - 1.0.27.2-r0
ap-mode-toggle - 0.1-r6
attr - 2.4.47-r0
attr-dev - 2.4.47-r0
autoconf - 2.69-r11
automake - 1.14-r0
```

Figure 1: List of all available packages on opkg

3. Print out all installed packages.



\$ opkg list-installed

```
root@edison:~/tutorial3_examples# opkg list-installed
alsa-conf-base - 1.0.27.2-r0
alsa-lib - 1.0.27.2-r0
alsa-states - 0.2.0-r4
alsa-utils-alsactl - 1.0.27.2-r0
alsa-utils-alsamixer - 1.0.27.2-r0
alsa-utils-amixer - 1.0.27.2-r0
alsa-utils-aplay - 1.0.27.2-r0
ap-mode-toggle - 0.1-r6
attr - 2.4.47-r0
attr-dev - 2.4.47-r0
autoconf - 2.69-r11
automake - 1.14-r0
base-files - 3.0.14-r86
base-files-dev - 3.0.14-r86
```

Figure 2: List of packages currently installed on the Intel Edison through opkg

4. Issue the below command to install **git**.

\$ opkg install git

Notice the error message.

5. To resolve this error, you must update the list of package sources so that **opkg** knows where to download various packages from. To do this, issue the below command:

\$ vi /etc/opkg/base-feeds.conf

Press [i] once you have opened the file in Vi.

6. Please type the following. Do **not** copy and paste it. For more information on why copy and pasting fails on some programs such as PuTTY, please refer to the document labelled *Intel Edison Tutorial – Introduction to Vim*.

```
src/gz all http://repo.opkg.net/edison/repo/all
src/gz edison http://repo.opkg.net/edison/repo/edison
src/gz core2-32 http://repo.opkg.net/edison/repo/core2-32
```

7. Exit **vi** by pressing “**esc**” followed by “**:wq**” and then “**enter**”.
8. Issue the below command to update the list of packages **opkg** has access to.

\$ opkg update



```
root@edison:~/tutorial3_examples# opkg update
Downloading http://iotdk.intel.com/repos/1.1/iotdk/all/Packages.
Updated list of available packages in /var/lib/opkg/all.
Downloading http://iotdk.intel.com/repos/1.1/iotdk/x86/Packages.
Updated list of available packages in /var/lib/opkg/x86.
Downloading http://iotdk.intel.com/repos/1.1/iotdk/i586/Packages.
Updated list of available packages in /var/lib/opkg/i586.
Downloading http://iotdk.intel.com/repos/1.1/intelgalactic/Packages.
Updated list of available packages in /var/lib/opkg/iotkit.
```

Figure 3: Output from issuing the command `opkg update` successfully

9. We can check if **git** is available now. Issue the following command:

```
$ opkg list | grep git\ -
```

NOTE: there is a space between the `\` and the `-` characters.

(`grep` is used to filter out all other entries except the one that contains `"git -"`).

```
root@edison:~/tutorial3_examples# opkg list | grep git\ -
git - 1.9.0-r0 - Distributed version control system Distributed version control system.
perl-module-config-git - 5.14.3-r1 - perl module config-git perl module config-git
```

Figure 4: Testing if see if git is available to download from `opkg`

10. `$ opkg install git`

```
root@edison:~/tutorial3_examples# opkg install git
Installing git (1.9.0-r0) to root...
Downloading http://iotdk.intel.com/repos/1.1/iotdk/i586/git_1.9.0-r0_i586.ipk.
Configuring git.
```

Figure 5: Standard output during the installation process of `git`

11. **git** has been successfully installed.
12. You can learn more **opkg** commands at <http://wiki.openwrt.org/doc/techref/opkg>.
13. Issue the below command to clone a **bitbucket** repository with **git**.

```
$ git clone https://chrisIHbaek@bitbucket.org/chrisIHbaek/tutorial3_example.git
```

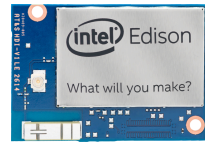
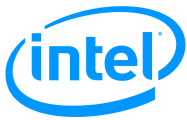
14. List all files and directories.

```
$ ls
```

15. A new “directory” named `tutorial3_example` is added. It is a cloned repository of https://chrisIHbaek@bitbucket.org/chrisIHbaek/tutorial3_example.git.

```
$ cd tutorial3_example
```

16. Verify that the folder contains a file named **README**.



17. It is very common during development to **lose files or progress** due to hardware failures, or human errors. As such, using **git** is very strongly recommended. It allows you to have an online repository for any code or files that you develop on your Intel Edison for both **version control** and **backup** purposes.

For more information regarding **git** please refer to the below link:

<https://www.atlassian.com/git/tutorials/>