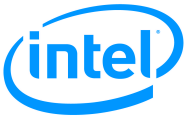
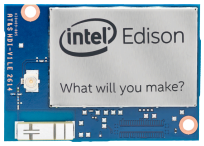




Intel® Edison Tutorial: Introduction to Vim

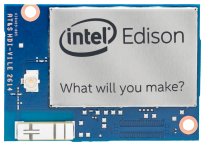
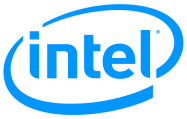
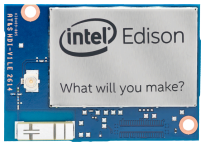


Table of Contents





Introduction

In this tutorial, users will:

1. Install Vim using the opkg package management system
2. Use the Vim editor to open and edit a simple C code source file
3. Learn about selected Vim features such as copying and pasting

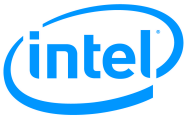
Prerequisite Tutorials

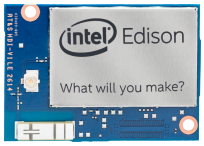
Users should ensure they are familiar with the documents listed below before proceeding.

1. Intel Edison Tutorial – Introduction, Linux Operating System Shell Access
2. Intel Edison Tutorial – Introduction to Linux
3. Intel Edison Tutorial – Introduction to Opkg

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





Introduction

What Is Vim?

Vim is a highly configurable text editor built to enable efficient text editing. It is an improved version of the vi editor distributed with most UNIX systems. For more information, refer to the link shown below.

http://www.vim.org/

Note: The cursor is the solid rectangular white block seen in the Vim editor.

Installation

1. Access the shell program on the Intel Edison. For more information, refer to the document labelled ***Intel Edison Tutorial – Introduction, Linux Operating System Shell Access and SFTP***.

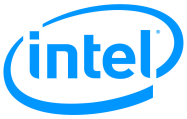
2. Ensure the Intel Edison is connected to a network that provides access to the internet. For more information, refer to the document labelled ***Intel Edison Tutorial – Introduction to Opkg***.

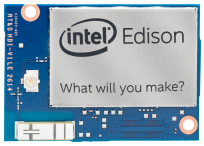
3. To install Vim, use the **opkg install** command.

   $ opkg install vim

```
root@ucla_iot:~# opkg install vim
Installing vim (7.4.481-r0) on root.
Downloading http://repo.opkg.net/edison/repo/core2-32/vim_7.4.481-r0_core2-32.ipk.
package vim suggests installing diffutils
Configuring vim.
update-alternatives: Linking //bin/vi to /usr/bin/vim
```

Figure 1: Standard output from opkg on successful install of Vim





Vim Basics

This section will show users how to use the Vim editor to write a basic C code source file.

Some users may habitually press **ctrl-S** at regular intervals as a method to ensure that their work is saved to their local machine. However, in Linux, this keystroke suspends a process. This means that if a user presses **ctrl-S** while using Vim, Vim will become frozen. To move the application from the suspended state to the running state, press **ctrl-Q**.

1. To open a file, issue the **vi** command.

   $ vi hello_world.c

Figure 2: Issuing the command to open a file in Vim

The **vi** command attempts to open a file associated with the name **hello_world.c** in the Vim editor. If no file is associated with the name **hello_world.c**, Vim will create a file and associate it with the name **hello_world.c**.

If the command is issued and processed correctly, the terminal should show a screen similar to the figure shown below.

Figure 3: hello_world.c file successfully opened in the Vim editor

2. The Vim editor enters the **command mode** when first opening a file.

   To begin modifying the file contents, press [I]. This will cause Vim to enter the **insert mode**.

Figure 4: The Vim editor should display --INSERT-- in the bottom left corner

3. Once Vim is in **insert mode**, users may edit the file contents.

   Type the following into Vim. Notice how the editor provides syntax highlighting.

Figure 5: Adding text to a file using the Vim editor

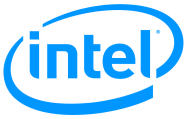
4. To save the file, press [Esc] followed by [Shift] + [:], [W], [Q].
5. Ensure the string ":wq" is displayed in the bottom left corner. If the string ":wq" is not displayed, repeat step 4 after pressing [Esc] two times. If this still does not resolve the issue, press **ctrl-C** twice, then press [Esc] twice, and then repeat step 4.

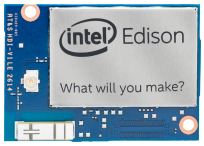
Figure 6: Save and quit the file

If the **ctrl-S** keystroke is issued, the terminal screen will become frozen. To resume regular operation, issue the **ctrl-Q** keystroke.

6. Users will be returned to the shell prompt after exiting the Vim editor.

Figure 7: The terminal session should return to the shell prompt after exiting the Vim editor





Tips and Tricks

Undo and Redo

1. To undo, press [Esc] and press [U].
2. To redo, press [Esc] and press [Ctrl]+[R].

Copy and Pasting

Issuing the **ctrl-C** and **ctrl-V** keystrokes will create signals in the Linux Operating System. This may differ from the copy and paste functionality that some users may be familiar with. As such, Vim provides an alternative method to copy and paste. However, the clipboard is only valid for sessions within the Vim editor. To address this, examine the subsection labelled Externally – Windows or Externally – Mac to copy and paste text externally to and from the Vim editor.

Internally

1. Open the file labelled **hello_world.c**

Figure 8: Opening an existing file with the Vim editor

2. Use the arrow keys to move the cursor to the line containing the below text.

   printf("Hello World\n");

   Note: Do not enter press [i]. Do not switch the Vim editor from the command mode. If the Vim editor is not in command mode, press [Esc] to enter the command mode.

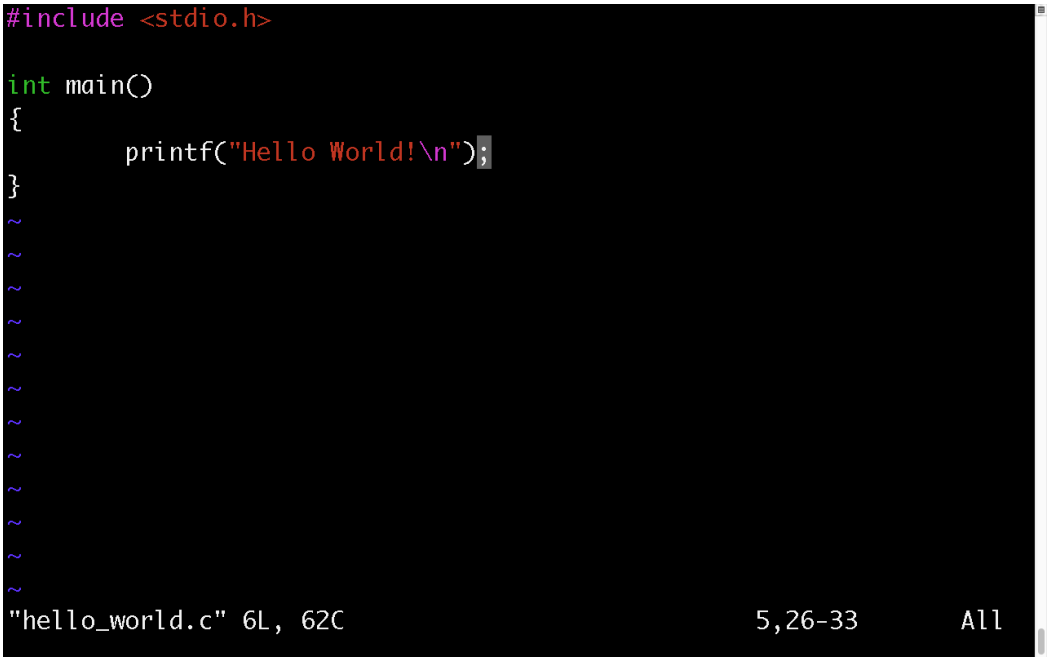
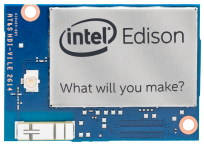


Figure 9: Vim editor showing the correct placement of the cursor

3. To *copy or yank the current line*, press [Y] twice. There is no visual feedback to indicate a successful copy operation.



4. To ***p**aste the contents of the copy buffer to the current line*, press [P] once.

```
#include <stdio.h>

int main()
{
        printf("Hello World!\n");
        printf("Hello World!\n");
}
~
~
~
~
~
~
~
~
~
~
~
~
                                                6,2-9         All
```

Figure 10: The contents of the file after pasting the copied line

5. To ***d**elete the current line*, press [D] twice. This operation is known as **cut** in other editors.

```
#include <stdio.h>

int main()
{
        printf("Hello World!\n");
}
~
~
~
~
~
~
~
~
~
~
~
~
~
                                                6,1           All
```

Figure 11: File contents performing the delete command on a line

6. Press [P] **four** times. Notice how the content last placed in the copy buffer is pasted.

```
#include <stdio.h>

int main()
{
        printf("Hello World!\n");
        printf("Hello World!\n");
        printf("Hello World!\n");
        printf("Hello World!\n");
        printf("Hello World!\n");
}
~
~
~
~
~
                                                5,1-8         All
```

Figure 12: Pasting a line multiple times in Vim

7. Edit the file such that it has the same contents as the figure shown below.

```
#include <stdio.h>

int main()
{
        printf("Hello World! 0\n");
        printf("Hello World! 1\n");
        printf("Hello World! 2\n");
        printf("Hello World! 3\n");
        printf("Hello World! 4\n");
}
```

Figure 13: Inserting numbers after each Hello World

8. To copy a collection of consecutive lines to the copy buffer, users must move the cursor to the desired line. Users will then issue the following keystrokes: [Y], number of lines below current line, [↓] (down arrow key).

   For example, if a user would like to copy a total of 3 lines, the user would need to press [Y], [2], [↓]. This will copy the current and following two lines to the copy buffer.

9. Move the cursor to the printf statement with the argument “Hello World! 0\n”.
10. Press [Y], [2], [↓].
11. Move the cursor to the semicolon of the printf statement with the argument “Hello World! 4\n”.

12. Press [P].

    The contents of the file should match the figure shown below.

```
#include <stdio.h>

int main()
{
        printf("Hello World! 0\n");
        printf("Hello World! 1\n");
        printf("Hello World! 2\n");
        printf("Hello World! 3\n");
        printf("Hello World! 4\n");
        printf("Hello World! 0\n");
        printf("Hello World! 1\n");
        printf("Hello World! 2\n");
}
~
~
~
~
~
3 more lines                                    10,2-9        All
```

Figure 14: File contents after copying and pasting 3 lines

13. Edit the file such the text contents match the figure shown below.

```
#include <stdio.h>

int main()
{
        printf("Hello World! 0\n");
        printf("Hello World! 1\n");
        printf("Hello World! 2\n");
        printf("Hello World! 3\n");
        printf("Hello World! 4\n");
        printf("Hello World! 5\n");
        printf("Hello World! 6\n");
        printf("Hello World! 7\n");
}
~
~
~
~
~
3 more lines                                    12,28-35      All
```

Figure 15: File contents before deleting multiple lines at once

14. Multiple lines may be deleted following instructions below.
15. Move the cursor to the printf statement with the argument “Hello World! 3\n”.
16. Press [D], [2], [↓].

The below image shows what should happen after the keystrokes of step 13.

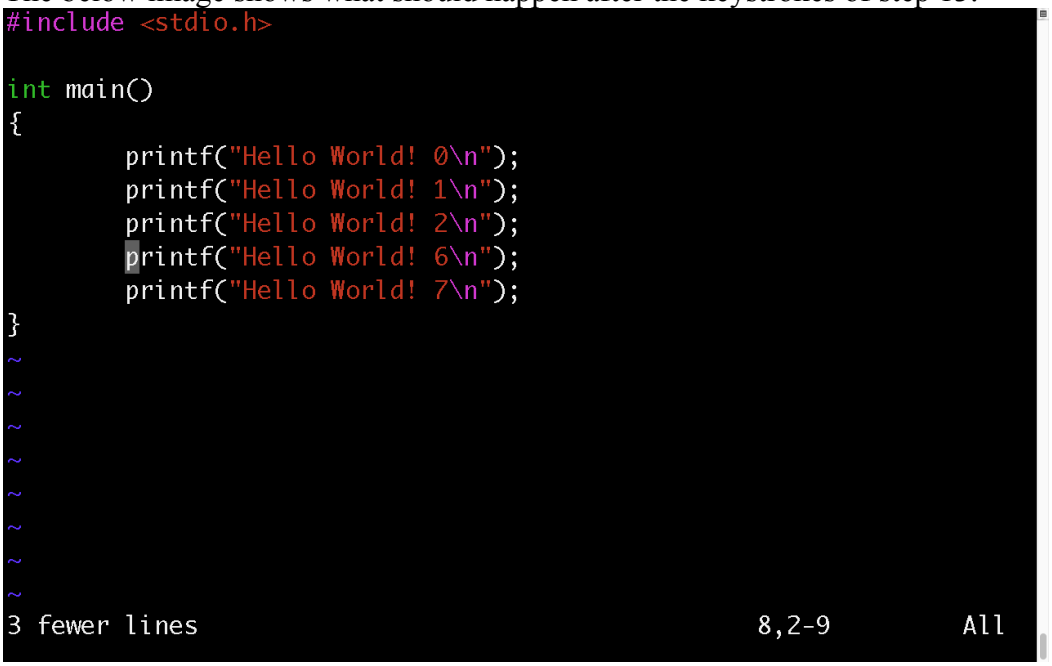


Figure 16: File contents after deleting multiple lines at once

Externally – Windows

Users familiar with the Windows operating system may have used **ctrl-C** and **ctrl-V** to copy and paste text. However, these keystrokes generate **interrupts** on many operating systems, including Linux. As such, pressing **ctrl-C** or **ctrl-V** may cause the Vim editor or current running process to terminate. The steps outlined below will ensure successful copy-paste functionality when attempting to copy text from an online reference and paste it into file open in the Vim editor.

1. Open a web browser on a personal computer and navigate to any website providing some example C code.
2. Select the C code text provided online with the mouse.
3. Press the right mouse button.
4. Clicking the **copy** option from the resulting drop down menu.

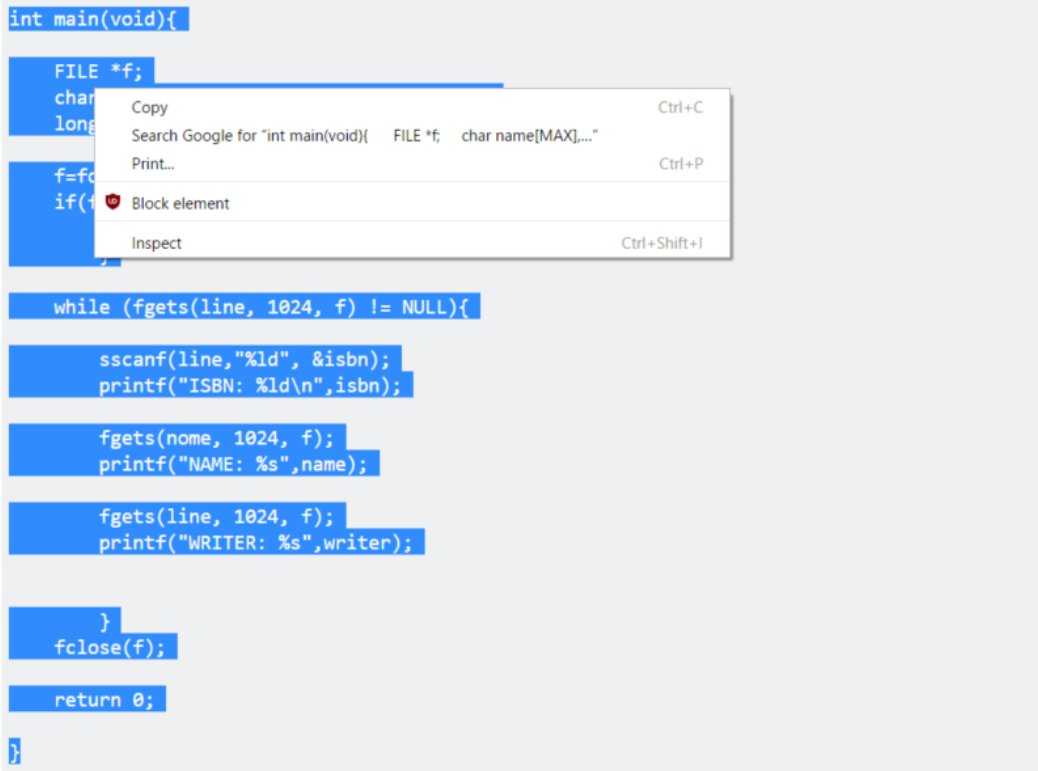


Figure 17: Online source that you may wish to copy and paste into Vim

5. Access the shell program on the Intel Edison.

   NB: PuTTY may still have issues with this method. For best results, use MobaXterm.

6. Open a new file called "hello_world_external.c".

   $ vi hello_world_external.c

7. Press [I] to enter insert mode.
8. Right click where the cursor is.

   Users accessing the shell through the PuTTY program may skip step 9 as the text will be pasted into the Vim editor immediately after clicking the right mouse button.

9. Click **paste** from the resulting drop down menu.

10. The pasted text may have indentation issues.

```
int main(void){

        FILE *f;
            char name[MAX], writer[MAX], line[MAX];
                long isbn;

                    f=fopen("path.txt","r");
                        if(f == NULL){
                                        return 0;
                                                }

                                while (fgets(line, 1024, f) != NULL){

                                                sscanf(line,"%ld", &isbn);
                                                        printf("ISBN: %ld\n",isbn);

                                                                fgets(nome, 1024, f);
                                                                        printf("NAME: %s",name);

                                                                                fgets(line, 1024, f);
                                                                                        printf("WRITER: %s",writer);


                                                                                                }

                                        fclose(f);

                                            return 0;

}
```

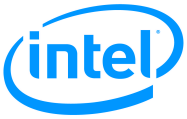
Figure 18: Pasting text from online sources may cause indentation issues

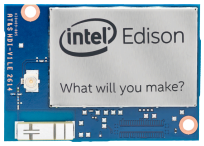
11. To fix this, press [Esc].
12. Move the cursor to the top of the file.
13. Ensure the editor remains in **command-mode.**

    Press [9] four times
    Press [=] two times

    This series of keystrokes instructs the Vim editor to automatically indent the next 9999 lines.

Figure 19: File contents after issuing the automatic indent command to Vim





Externally – Mac

Copy and pasting text between the Intel Edison and online sources works as expected.

Press ⌘**-C** to copy and press ⌘**-V** to paste.

Ensure the Vim editor is in insert mode before attempting to paste code.

Follow the instructions outlined below to resolve indentation issues.

1. To examine a file with indentation issues, examine Figure 18
2. Open the file with indentation issues using Vim.
3. Press [Esc].
4. Move the cursor to the top of the file.
5. Ensure the editor remains in **command-mode.**

   Press [9] four times
   Press [=] two times

   This series of keystrokes instructs the Vim editor to automatically indent the next 9999 lines.

6. The file should now have correct indentation. Examine Figure 19 for more details.