

# Lesson Proper for Week 4

## OPEN SOURCE CUSTOM ROUTER FIRMWARE

**Router Firmware**

It is the preinstalled, embedded software that manages the control of routing protocols, administrative features and the router's security mechanism.

**Firmware**

Used to operate the router and works as an operating system by providing an interface, protocol configuration and security settings. It allows the router to be configured and customized according to the network operating environments.

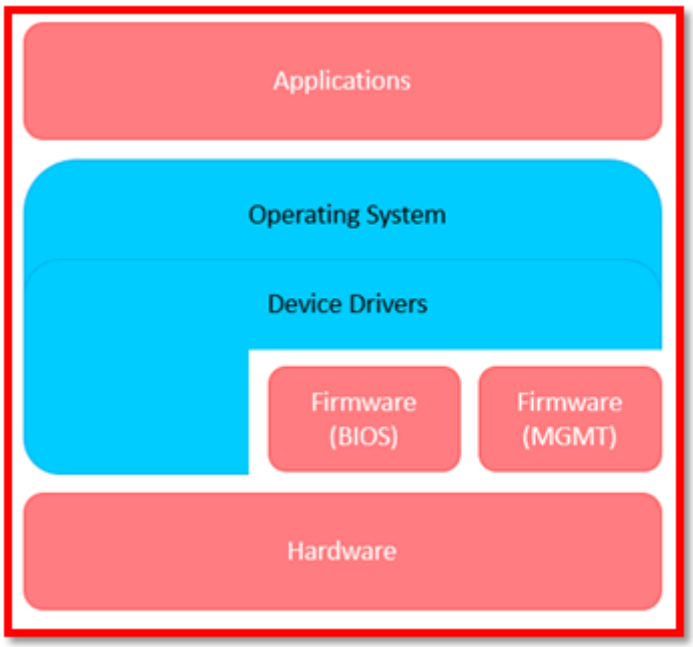
A software program typically stored in the flash ROM (Read-only-Memory) of a hardware device which provides instructions on how the machine should operate.

***Origin of the term***

Ascher Opler coined the term “firmware” in a 1967 Datamation article. Originally, it meant the contents of a writable control store (a small specialized high speed memory), containing microcode that defined and implemented the computer’s instruction set, and that could be reloaded to specialize or modify the instructions that the central processing unit (CPU) could execute.

As originally used, firmware contrasted with hardware (the CPU itself) and software (normal instructions executing on a CPU). It was not composed of CPU machine instructions, but of lower-level microcode involved in the implementation of machine instructions. It existed on the boundary between hardware and software; thus the name “firmware”.

**FIRMWARE, DEVICE DRIVER AND SOFTWARE**




| Firmware | Device Driver | Software |
|---|---|---|
| A type of software program that enables device functionalities without the need for installing additional software | A type of software program that enables the communication between an OS and the hardware | A broad term used to define a set of instructions that enables a device to function in specific ways |
| Types of firmware include BIOS, EFI (Extensible Firmware Interface), etc. | A device driver is hardware-specific. For instance, printer driver, graphics driver, etc. | Types of software include application software, shareware, system software, etc. |
| A firmware gives life to a hardware | A device driver ensures the smooth functioning of a device | A software adds functionality to a device |
| A firmware is not meant for user interaction | A device driver is not meant for user interaction | A software is meant for user interaction |

## EXAMPLES OF FIRMWARE

Examples of devices containing firmware are embedded systems:

§ Traffic lights

§ Consumer appliances

§ Digital watches

§ computers,

§ computer peripherals,

§ mobile phones, and

§ digital cameras

**The firmware contained in these devices provides the control program for the device.*

## TYPES OF FIRMWARE

### 1. BIOS (Basic Input/Output)

After pressing the power button to turn on, the computer will boot into the **BIOS.** It can interact with the hardware and check for any errors, then signal to another program called Bootloader, do the task of waking the sleeping operating system inside the hard drive and sending it. Into temporary data memory (Random Access Memory – RAM).

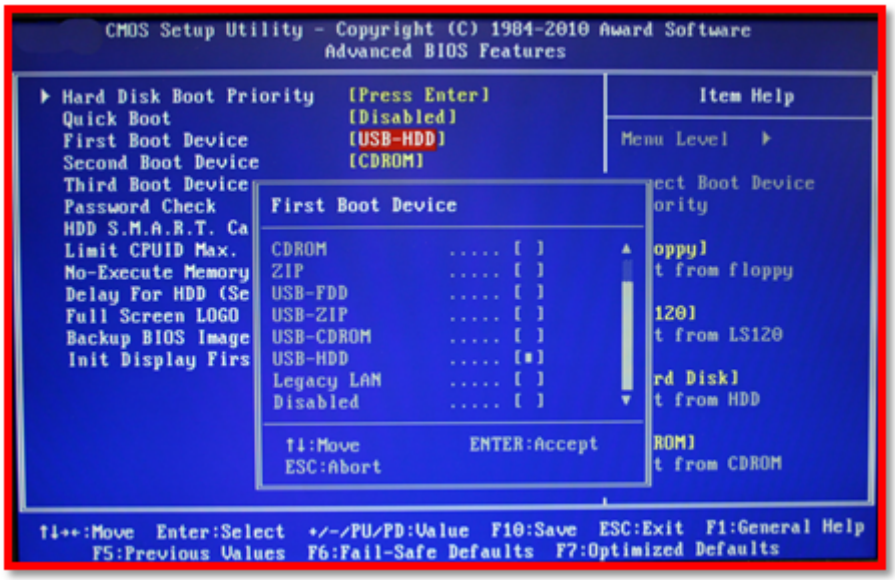
The main task of the **BIOS** is to handle the components of the computer hardware and ensure that these components work properly.

The BIOS's primary function is to handle the system setup process including driver loading and operating system booting.

*The 4 functions of BIOS*

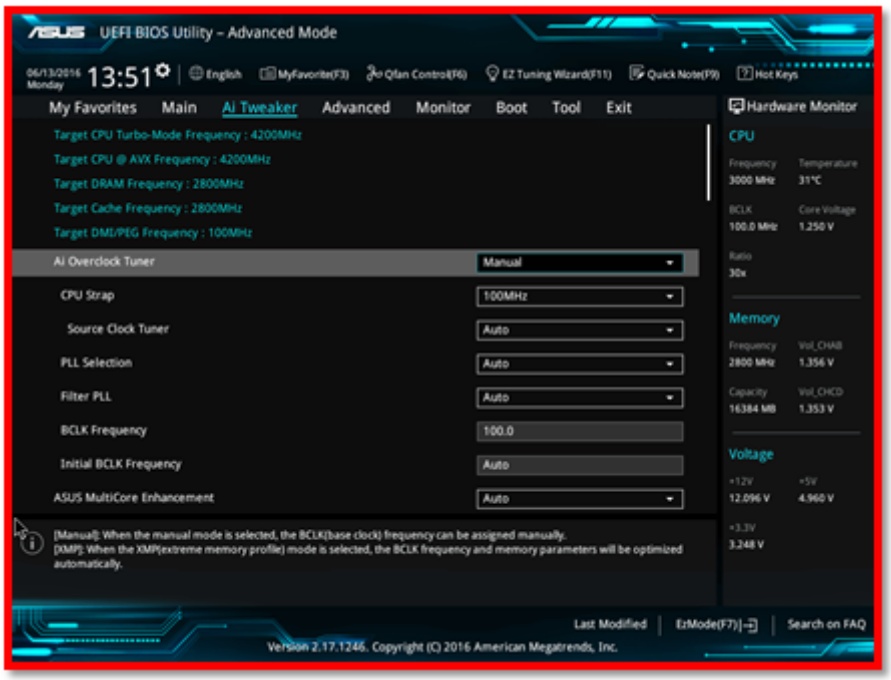
- Power-on self-test (POST). This tests the hardware of the computer before loading the OS.
- Bootstrap loader. This locates the OS.

- Software/drivers. This locates the software and drivers that interface with the OS once running.
- Complementary metal-oxide semiconductor (CMOS) setup.




**2. EFI (Extensible Firmware Interface)**

Is a software protocol specification that is responsible for communicating between the operating system and system firmware, used by the CPU to boot hardware and bypass the Bootloader. Sometimes EFI is also known as UEFI (Stands for Unified Extensible Firmware Interface) and certain advantages over the BIOS.




**ADVANTAGES OF UEFI**

UEFI provides many significant enhancements over BIOS, including the following:

o **Boot mode.** Microsoft Windows users can run 32-bit UEFI or 64-bit UEFI, although experts recommend that the OS bit mode and the firmware bit mode should be the same to avoid communication issues during runtime.

o **Drives.** UEFI supports boot drives of 2.2 TB and higher capacities, including drives with theoretical capacity of 9.4 zettabytes.

o **Drivers.** UEFI supports discrete drivers, whereas BIOS drive support is stored in read-only memory, which necessitates tuning it for compatibility when drives are swapped out or changes are made.

o **Graphical user interface (GUI).** UEFI enables new modules to be added to the GUI more easily, including device drivers for motherboard hardware and attached peripheral devices.

o **Multiple OS support.** Whereas BIOS allows a single boot loader, UEFI lets users install loaders for Debian-based Ubuntu and other Linux variants, along with Windows OS loaders, in the same EFI system partition.

o **Security.** Secure Boot is a UEFI protocol for Windows 8 or later Windows versions. Secure Boot makes a system's firmware the root of trust to verify device and system integrity. The goal is to prevent hackers from installing rootkits in the time between bootup and handoff to the OS. Secure Boot also enables an authorized user to configure networks and troubleshoot issues remotely, something a BIOS administrator must be physically present to do.

◄ Preliminary Activity for Week 4 Jump to...

Analysis, Application, and Exploration for Week 4 ►

## Navigation

Home
Dashboard
Site pages
My courses
Capstone Project 1
Network Attacks: Detection, Analysis & Counter...
Participants
General
01 The Home Router
02 External Vulnerability Scanning - Shodan, Qual...
03 Internal Vulnerability Scanning
04 Open Source Custom Router Firmware
Preliminary Activity for Week 4
**Lesson Proper for Week 4**
Analysis, Application, and Exploration for Week 4
Generalization for Week 4

## Fair Warning

**NOTICE**: Please be reminded that it has come to the attention of the Publishing Team of eLearning Commons that learning materials published and intended for ***free use only by students and faculty members within the eLearning Commons network were UNLAWFULLY uploaded in other sites without due and proper permission***.

**PROSECUTION**: Under Philippine law (Republic Act No. 8293), copyright infringement is punishable by the following: Imprisonment of between 1 to 3 years and a fine of between 50,000 to 150,000 pesos for the first offense. Imprisonment of 3 years and 1 day to six years plus a fine of between 150,000 to 500,000 pesos for the second offense.

**COURSE OF ACTION**: Whoever has maliciously uploaded these concerned materials are hereby given an ultimatum to take it down within 24-hours. Beyond the 24-hour grace period, our Legal Department shall initiate the proceedings in coordination with the National Bureau of Investigation for IP Address tracking, account owner identification, and filing of cases for prosecution.

## Activities

Assignments
Forums
Quizzes
Resources