Embedded Systems Architecture

BIOS
UEFI

M.Eng. Mariusz Rudnicki
BIOS – BASIC INPUT/OUTPUT SYSTEM

BIOS – System BIOS, ROM BIOS, PC BIOS:
- a type of firmware used during the booting process (power-on startup);
- first software which is running when the system is powered on;

BIOS:
- initializes and tests the system hardware components;
- loads a boot loader or an operating system from a mass memory devices;
- provides an abstraction layer for the hardware;
  - a consistent way for application programs and operating systems to interact with the:
    - keyboard,
    - mouse;
    - display,
    - and other input/output devices;
Variations in the system hardware are hidden by the BIOS from programs that use BIOS services instead of directly accessing the hardware – abstraction layer.

**MS-DOS (PC DOS):**
- relied on BIOS services for disk, keyboard, and text display functions.

**MS Windows NT, Linux, and other protected mode operating systems**
- in general ignore the abstraction layer provided by the BIOS and do not use it after loading, instead accessing the hardware components directly.
BIOS – BASIC INPUT/OUTPUT SYSTEM

- Every BIOS implementation is specifically designed to work with a particular computer or motherboard model.

- Originally, BIOS firmware was stored in a ROM chip on the PC motherboard.

- In modern computer systems, the BIOS contents are stored on flash memory. This allows easy updates to the BIOS firmware; new features can be added or bugs can be fixed, but it also creates a possibility for the computer to become infected with BIOS rootkits.

- EEPROM chips are advantageous because they can be easily updated by the user – this has the risk that an improperly executed or aborted BIOS update could render the computer or device unusable;
BIOS – BASIC INPUT/OUTPUT SYSTEM

BIOS viruses:
- CIH (Chen Ing Hau) – Chernobyl accident - it was able to erase flash ROM BIOS content;
- BIOS virus – technique presented by John Heasman – ACPI replacement;
- Persistent BIOS infection – insert malicious code into the decompression routines in the BIOS, allowing for nearly full control of the PC at start-up.

- To avoid these situations, more recent BIOSes use a "boot block"; a portion of the BIOS which runs first and must be updated separately and this code verifies if the rest of the BIOS is intact (using hash checksums or other methods).

- Boot block detects any corruption in the main BIOS – warn the user that a recovery process must be initiated by booting from removable media.

- Backup BIOS (sometimes referred to as DualBIOS boards).
User interface:
- Originally, BIOS had no interactive user interface;
- Error codes and messages were displayed on the screen,
- Or coded series of sounds were generated to signal errors.

Mid-1990s „BIOS configuration utility” or "BIOS setup utility“ - allowed the user to set system configuration options;

System startup:
- after system has just been powered up,
- or the reset button was pressed ("cold boot"), the full power-on self-test (POST) is run;
- key combination CTRL+ALT+DEL was pressed ("warm boot"), a special flag value is stored in nonvolatile BIOS memory BIOS startup code detects this flag and does not run the POST.
Features present in the BIOS setup utility typically include:

- Configuring the hardware components, including setting their various operating modes and frequencies (for example, selecting how the storage controllers are visible to the operating system, or overclocking the CPU);
- Setting the system clock;
- Enabling or disabling system components;
- Selecting which devices are potential boot devices, and in which order booting from them will be attempted;
- Setting various passwords, such as:
  - a password for securing access to the BIOS user interface functions itself and preventing malicious users from booting the system from unauthorized portable storage devices,
  - a password for booting the system, or a hard disk drive password that limits access to it and stays assigned even if the hard disk drive is moved to another computer.
BIOS – BASIC INPUT/OUTPUT SYSTEM

IDE Configuration
- OnBoard PCI IDE Controller: [Both]
  - Primary IDE Master: [Hard Disk]
  - Primary IDE Slave: [Not Detected]
  - Secondary IDE Master: [Not Detected]
  - Secondary IDE Slave: [Not Detected]

Hard Disk Write Protect: [Disabled]
IDE Detect Time Out (Sec): [35]
ATA(PI) 60Pin Cable Detection: [Host & Device]

Configure advanced CPU settings
- Manufacturer: Intel
- Brand String: Intel(R) Pentium(R) M processor 1600M
- Frequency: 600MHz
- FSB Speed: 400MHz
- Cache L1: 32 KB
- Cache L2: 1024 KB
- Ratio Status: Locked
- Ratio Actual Value: 6
- Intel(R) SpeedStep(tm) tech. [Maximum Performance]
  - Limit CPU clock ratio: [255]
BIOS – BASIC INPUT/OUTPUT SYSTEM

- Boot Settings Configuration
  - Quick Boot: [Enabled]
  - Quiet Boot: [Enabled]
  - AddOn ROM Display Mode: [Force BIOS]
  - Bootup Num-Lock: [Off]
  - PS/2 Mouse Support: [Auto]
  - Parity Check: [Enabled]
  - Boot To OS/2: [No]
  - Wait For 'F1' If Error: [Enabled]
  - Hit 'DEL' Message Display: [Enabled]
  - Interrupt 19 Capture: [Disabled]
  - Primary Display Preference: [Add-On Card]
  - Boot Failure Retry Prompt: [Enabled]

- Select Power-on state for Numlock.

- Chipset
  - Tundra Universe PCI-to-UMEbus bridge
    - UMEbus properties
      - System Controller State: [Auto]
      - Fast DTACK# Filtering: [Disabled]

- PCI properties
  - UMEbus Access Window Size: [512MB]
BIOS limitations were unacceptable for the larger server platforms.

- In 1998 - was founded Intel Boot Initiative - later renamed to EFI.

- 2005, Intel ceased development of the EFI specification at version 1.10, and contributed it to the Unified EFI Forum.

- 2007 – Version 2.1 of the UEFI (Unified Extensible Firmware Interface) specification was released. It added:
  - cryptography,
  - network authentication,
  - and the User Interface Architecture (Human Interface Infrastructure in UEFI).

- 2013 – UEFI specification, version 2.4, was approved.
UEFI - Unified Extensible Firmware Interface

Operating system

Extensible Firmware Interface

Firmware

Hardware
UEFI - Unified Extensible Firmware Interface

- UEFI – specification that defines a software interface between an operating system and platform firmware.
- UEFI – replaces the Basic Input/Output System (BIOS) firmware interface.
- Most UEFI firmware images provide legacy support for BIOS services. UEFI can support:
  - remote diagnostics,
  - repair of computers, even without another operating system.
- Unified EFI Forum manages the UEFI specification.
UEFI - Unified Extensible Firmware Interface

- Interface defined by the EFI specification includes data tables of:
  - platform information;
  - boot and runtime services that are available to the OS loader and OS;
- UEFI firmware provides several technical advantages over a traditional BIOS system:
  - CPU-independent architecture;
  - CPU-independent drivers;
  - Flexible pre-OS environment, including network capability;
  - Modular design;
  - Ability to boot from large disks (over 2 TB) with a GUID Partition Table (GPT – Globally Unique Identifier).
UEFI - Unified Extensible Firmware Interface

- Processor compatibility;
  - Itanium,
  - x86,
  - x86-64,
  - ARM (AArch32),
  - ARM64 (AArch64).
- Disk device compatibility;
  - new partitioning scheme: GUID Partition Table (GPT);
    - GPT is free from many of the limitations of MBR. In particular, the MBR limits on the number and size of disk partitions (up to 4 primary partitions per disk, up to 2 TB ($2 \times 2^{40}$ bytes) per disk) are relaxed.
    - GPT allows for a maximum disk and partition size of 8 ZB ($8 \times 2^{70}$ bytes).
UEFI - Unified Extensible Firmware Interface

- **Booting:**
- **UEFI booting:**
  - Unlike BIOS, UEFI does not rely on a boot sector;
  - Instead defines a boot manager as part of the UEFI specification;
  - When a computer is powered on, the boot manager checks the boot configuration and, based on its settings, loads and executes the specified operating system loader or operating system kernel. The boot configuration is a set of global-scope variables stored in NVRAM, including the boot variables that indicate the paths to operating system loaders or kernels.
  - Operating system loaders can also be automatically detected by an UEFI implementation, what enables easy booting from removable devices such as USB flash drives.
  - Booting UEFI systems from GPT-partitioned disks is commonly called **UEFI-GPT booting**.
UEFI - Unified Extensible Firmware Interface

- **Booting:**
- **CSM booting:**
  - Compatibility Support Module (CSM) which provides legacy BIOS compatibility;
  - support booting in legacy BIOS mode from MBR-partitioned disks;
  - booting is performed in the same way as on legacy BIOS-based systems, by ignoring the partition table and relying on the content of a boot sector.
UEFI - Unified Extensible Firmware Interface

- **Booting:**
  - Network booting:
    - Network protocols:
      - IPv4 and IPv6 – Internet Protocol;
      - UDP – User Datagram Protocol;
      - DHCP – Dynamic Host Configuration Protocol;
      - TFTP – Trivial File Transfer Protocol;
  - PXE – Preboot eXecution Environment: support for boot images remotely stored on Storage Area Networks (SANs):
    - Internet Small Computer System Interface (iSCSI);
    - Fibre Channel over Ethernet (FCoE);
UEFI - Unified Extensible Firmware Interface

- **Booting:**
- Network booting:
- PXE – Preboot eXecution Environment – describes a standardized client-server environment that boots a software retrieved from a network;
UEFI - Unified Extensible Firmware Interface

- initially downloaded and run Network Bootstrap Program (NBP) must be built relying on a client;
- firmware layer providing a hardware independent standardized way to interact with the surrounding network booting environment;
- availability and subjection to standards are a key factor required to guarantee the network boot process system interoperability.
UEFI - Unified Extensible Firmware Interface

- PXE environment:
  - relies on a combination of industry-standard Internet protocols, namely UDP/IP, DHCP and TFTP – why this?;
    - easily implemented in the client's NIC firmware;
    - because of above standardized small-footprint PXE ROMs;
    - small size of PXE firmware images;
    - low use of resources;
  - PXE standard to be identically implemented on a wide variety of systems: single-board computers (SBC) and system-on-a-chip (SoC) computers.
UEFI - Unified Extensible Firmware Interface

- PXE environment:
- DHCP is used to provide the appropriate client network parameters and specifically the location (IP address) of the TFTP server hosting;
- To initiate a PXE bootstrap session the DHCP component of the client's PXE firmware broadcasts a DHCPDISCOVER packet containing PXE-specific options to port 67/UDP (DHCP server port),
- it asks for the required network configuration and network booting parameters;
- PXE-specific options identify the initiated DHCP transaction as a PXE transaction;
- After parsing a PXE enabled DHCP server DHCPOFFER, the client will be able to set its own network IP address, IP Mask, etc., and to point to the network located booting resources, based on the received TFTP Server IP address and the name of the NBP.
UEFI - Unified Extensible Firmware Interface

- **PXE environment:**
  - The client next transfers the NBP into its own random-access memory (RAM) using TFTP, possibly verifies it (i.e. UEFI Secure Boot), and finally boots from it;
  - NBPs are just the first link in the boot chain process and they generally request via TFTP a small set of complementary files in order to get running a minimalistic OS executive;
    - WindowsPE;
    - basic Linux kernel+initrd;
  - Small OS executive is alive it loads its own fully capable network drivers, a full TCP/IP stack,
  - The rest of transfers for booting or installing a full OS are performed not by TFTP but at this point using more robust transfer protocols like HTTP, CIFS, NFS, etc.
UEFI - Unified Extensible Firmware Interface

- Network booting:
- classic DHCP server providing IP address, IP mask, etc. to all booting DHCP clients – port 67/UDP;
- proxyDHCP server providing TFTP server IP address and name of the NBP only to PXE identified booting clients – port 4011/UDP.
**Network booting:**

- Projects have included PXE support:
  - All the major Linux distributions;
  - HP OpenVMS on Itanium;
  - Microsoft Remote Installation Services (RIS);
  - Microsoft Windows Deployment Services (WDS);
  - Microsoft Deployment Toolkit (MDT);
  - Microsoft System Center Configuration Manager (SCCM);
  - Syslinux PXELINUX;
  - gPXE/iPXE;
- Above-mentioned projects, are able to boot/install more than one OS, work under a "Boot Manager - Boot Loader" paradigm.

- Initial NBP is a Boot Manager able to retrieve its own configuration and deploy a menu of booting options;
- The user selects a booting option;
- An OS dependent Boot Loader is downloaded and run in order to continue with the selected specific booting procedure.
Network booting:

Similar enviroments:

- Apple – Boot Server Discovery Protocol (BSDP);
  - The OS X Server includes a system tool called NetBoot.
  - A NetBoot client uses BSDP to dynamically acquire resources that enable it to boot a suitable OS;
  - BSDP is crafted on top of DHCP using vendor-specific information to provide the additional NetBoot functionality not present in standard DHCP;
- The protocol is implemented in client firmware;
- the client obtains an IP address via DHCP then discovers boot servers using BSDP;
UEFI - Unified Extensible Firmware Interface

- BSDP server responds with boot information consisting of:
  - A list of bootable operating system images
  - The default operating system image
  - The client’s currently selected operating system image (if defined)
- client chooses an operating system from the list and sends a message to the server indicating its selection;
- selected boot server responds supplying the boot file and boot image;
- and any other information needed to download and execute the selected operating system.

- Boot Information Negotiation Layer (BINL) – key component of the Remote Installation Services (RIS) and Windows Deployment Services (WDS) strategies.
UEFI - Unified Extensible Firmware Interface

- PXE Error:
  - Init/Boot/Loader Codes;
  - ARP Codes;
  - BIOS and BIS Codes;
  - TFTP/MTFTP Codes;
  - BOOTP/DHCP Codes;
  - UNDI Codes;
  - Bootstrap and Discovery Codes;
  - Miscellaneous Codes;
  - BaseCode/UNDI Loader Codes;
UEFI - Unified Extensible Firmware Interface

- **Booting:**
- **Secure boot:**
  - The UEFI 2.2 specification adds a protocol known as *secure boot*:
  - it can secure the boot process;
  - by preventing the loading of drivers or OS loaders that are not signed with an acceptable digital signature;
  - During time the secure boot is enabled, “setup” mode allows a public key (Platform Key – PK) to be written to the firmware.
  - “User” mode – only drivers and loaders signed with the platform key can be loaded by the firmware;
  - “Key Exchange Keys” (KEK) can be added to a database stored in memory to allow other certificates to be used, but they must still have a connection to the private portion of the Platform key;
  - “Custom” mode, where additional public keys can be added to the system that do not match the private key.
UEFI - Unified Extensible Firmware Interface

- References:
  - [http://www.uefi.org/specifications](http://www.uefi.org/specifications):  
    - UEFI Specification Version 2.4 (Errata B)
    - ACPI Specification Version 6.0
  - Preboot Execution Environment (PXE) Specification Version 2.1