Firmware Settings and Menus

Daniel Maslowski
Agenda

- Introduction
- History
- Modern Firmware Interfaces
- Ideas for Open Source Firmware
Introduction
Hello, I am Daniel :) 

Work and education
- IT security and computer science
- software engineer
- infrastructure and web
- apps, UIs, ecommerce

Open Source contributions
- hardware and firmware
- operating systems
- software distributions
- reverse engineering
- Fiedka the Firmware Editor
Fiedka

Fiedka is a graphical firmware editor app\(^1\).

\(^1\)https://fiedka.app/
User Interfaces are Critical

Navy Reverting DDGs Back to Physical Throttles, After Fleet Rejects Touchscreen Controls

By: Megan Eckstein
August 9, 2019 10:46 AM

User Interface Design

Figure 4. John S McCain SCC. (Drawing from IBNS technical manual; color added by NTSB)
History
Early Firmware and Interfaces

Early Firmware and Interfaces

**BIOS**
- first compatible commercial implementation by Phoenix Technologies\(^3\)
- sparked the IBM PC compatible computer

---

\(^3\)https://historyofinformation.com/detail.php?entryid=3846

\(^4\)http://www.firmworks.com/www/ofw.htm
Early Firmware and Interfaces

BIOS
- first compatible commercial implementation by Phoenix Technologies
- sparked the IBM PC compatible computer

Open Firmware (IEEE 1275)
- first non-proprietary boot firmware for different processors and buses
- Forth interpreter as UI

---

4 http://www.firmworks.com/www/ofw.htm
Early Firmware and Interfaces

BIOS
- first compatible commercial implementation by Phoenix Technologies\(^3\)
- sparked the IBM PC compatible computer

Open Firmware (IEEE 1275)
- first non-proprietary boot firmware for different processors and buses\(^4\)
- Forth interpreter as UI

EFI
- Human Interface Infrastructure (HII)\(^5\)
- standardized protocol and data structures for building forms

\(^3\)https://historyofinformation.com/detail.php?entryid=3846
\(^4\)http://www.firmworks.com/www/ofw.htm
Open Firmware Interfaces

User Interface

Open Firmware
(Resides in ROM on CPU Board)

Device Interface

Client Interface

Client Program
(Operating System or its loader)

Expansion Bus

Disk

Controller

Network

Monitor

Keyboard
Open Firmware Interactive Environment

Welcome to Open Firmware, the system time and date is: 19:52:42 02/17/2014
To continue booting, type "mac-boot" and press return.
To shut down, type "shut-down" and press return.

0 > dev /aliases .properties
name    aliases
hd      /pci@f4000000/ata-6@d/disk@0
        /pci@f2000000/mac-io@17/ata-3@20000/disk@0
        /pci@f2000000/usb@1b,1
        /pci@f2000000/usb@1b
        /pci@f2000000/usb@1a

0 > _

image originally from https://www.morphos-team.net/guide/usb-boot
see also https://www.youtube.com/watch?v=u9OMOHl73IE
Visual BIOS

Oh my word! This @VisualBIOS thing is amazing! When did this happen?! Why was I not informed?
#nextunitofcomputing
Modern Firmware Interfaces
NUI vs TUI vs GUI

NUI: User interface that applies to embedded devices mostly, where interactive access is not necessary.

TUI: Textual user interface available even in non-graphical environments, such as via serial console.

GUI: Graphical user interface most suitable for end users, can support accessibility.
NUI vs TUI vs GUI

NUI
No user interface - applies to embedded devices mostly, where interactive access is not necessary.
NUI vs TUI vs GUI

NUI
No user interface - applies to embedded devices mostly, where interactive access is not necessary.

TUI
Textual user interface - available even in non-graphical environments, such as via serial console.
NUI vs TUI vs GUI

NUI
No user interface - applies to embedded devices mostly, where interactive access is not necessary.

TUI
Textual user interface - available even in non-graphical environments, such as via serial console.

GUI
Graphical user interface - most suitable for end users, can support accessibility.
Open Source Implementations

---

7https://zirblazer.github.io/htmlfiles/coreboot.html?ver=123#chapter-3
Open Source Implementations

coreboot

- nvramtool (for OS), nvramcui (payload)
- coreinfo (payload)
- corevantage, coreboot-configurator (GUIs)

---

7https://zirblazer.github.io/htmlfiles/coreboot.html?ver=123#chapter-3
Open Source Implementations

**coreboot**
- nvramtool (for OS), nvramcui (payload)
- coreinfo (payload)
- corevantage, coreboot-configurator (GUIs)

**LinuxBoot**
- shell
- Heads
- webboot and boot menu (TUI)

---

7 [https://zirblazer.github.io/htmlfiles/coreboot.html?ver=123#chapter-3](https://zirblazer.github.io/htmlfiles/coreboot.html?ver=123#chapter-3)
Open Source Implementations

coreboot

- nvramtool (for OS), nvramcui (payload)\(^7\)
- coreinfo (payload)
- corevantage, coreboot-configurator (GUIs)

LinuxBoot

- shell
- Heads
- webboot and boot menu (TUI)

U-Boot

- interactive command interface

\(^7\)https://zirblazer.github.io/htmlfiles/coreboot.html?ver=123#chapter-3
Open Source Implementations

**coreboot**
- `nvramtool` (for OS), `nvramcui` (payload)
- `coreinfo` (payload)
- `corevantage`, `coreboot-configurator` (GUIs)

**LinuxBoot**
- `shell`
- `Heads`
- `webboot` and `boot menu` (TUI)

**U-Boot**
- Interactive command interface

**Tianocore / EDK2**
- UEFI Shell
- Setup Browser (interactive menu, TUI)

---

7https://zirblazer.github.io/htmlfiles/coreboot.html?ver=123#chapter-3
Graphical Firmware User Interfaces
UI Features

The UI has clickable elements, but mostly, simple text.
UI Features

The UI has clickable elements, but mostly, simple text.

Informative
- hard component info: DRAM, CPU, …
- soft component info: firmware itself, ucode, …
- hardware monitor
- QR code: link to the manual
- date/time, internationalization
UI Features

The UI has clickable elements, but mostly, simple text.

Informative
- hard component info: DRAM, CPU, …
- soft component info: firmware itself, ucode, …
- hardware monitor
- QR code: link to the manual
- date/time, internationalization

Settings
- clock adjustments
- boot media / source, order, default
- Secure Boot key provisioning

Note: screenshot taken from within the UI, stored to USB drive
UI Features

The UI has clickable elements, but mostly, simple text.

Informative
- hard component info: DRAM, CPU, …
- soft component info: firmware itself, ucode, …
- hardware monitor
- QR code: link to the manual
- date/time, internationalization

Settings
- clock adjustments
- boot media / source, order, default
- Secure Boot key provisioning

Note: screenshot taken from within the UI, stored to USB drive
EFI variables

$ xxd /sys/firmware/efi/efivars/SMBIOSELOG000-c3eeae98-23bf-412b-*

00000000: 0700 0000 0000 0000 0060 0160 0000 0000 ............`\`....
00000010: 0000 0001 0890 1901 0100 0108 0002 0000 ................
00000020: 0000 0000 0890 1901 0100 0118 0002 0000 ................
00000030: 0000 0000 0890 1901 0100 0236 0002 0000 ................ 6....
00000040: 0000 0000 0890 1901 0100 0302 0002 0000 ................
00000050: 0000 0000 0890 1901 0100 0035 0010 0000 ................ 5....
00000060: 0000 0000 0890 1901 0100 0035 0002 0000 ................ 5....
00000070: 0000 0000 0890 1901 0100 0042 0002 0000 ...........B.....
00000080: 0000 0000 0890 2006 2808 3720 0002 0000 ...... (.7 ....
EFI variables

$ xxd /sys/firmware/efi/efivars/SMBIOSELOG000-c3eeae98-23bf-412b-*
00000000: 0700 0000 0000 0000 0060 0160 0000 0000 ...........`.`.....
00000010: 0000 0001 0890 1901 0100 0108 0002 0000 ................
00000020: 0000 0000 0890 1901 0100 0118 0002 0000 ................
00000030: 0000 0000 0890 1901 0100 0236 0002 0000 ................ 6.....
00000040: 0000 0000 0890 1901 0100 0302 0002 0000 ................
00000050: 0000 0000 0890 1901 0100 0035 0010 0000 ............... 5.....
00000060: 0000 0000 0890 1901 0100 0035 0002 0000 ............... 5.....
00000070: 0000 0000 0890 1901 0100 0042 0002 0000 ............B.....
00000080: 0000 0000 0890 2006 2808 3720 0002 0000 ......(.7.....

Can we create or do we have a parser and a viewer for this?
coreboot nvramtool

dump coreboot tables: nvramtool -d

coreboot table at physical address 0x76b42000:
  signature: 0x4f49424c (ASCII: LBIO)
  header_bytes: 0x18 (decimal: 24)
  header_checksum: 0x4d99 (decimal: 19865)
  table_bytes: 0x7d4 (decimal: 2004)
  table_checksum: 0x18b9 (decimal: 6329)
  table_entries: 0x2c (decimal: 44)

CMOS_OPTION_TABLE record at physical address 0x76b42018:
  tag: 0xc8 (decimal: 200)
  size: 0x294 (decimal: 660)
  data:
  ...

Could this be more intuitive?
coreboot nvramtool

dump coreboot tables: nvramtool -d

coreboot table at physical address 0x76b42000:
  signature: 0x4f49424c (ASCII: LBIO)
  header_bytes: 0x18 (decimal: 24)
  header_checksum: 0x4d99 (decimal: 19865)
  table_bytes: 0x7d4 (decimal: 2004)
  table_checksum: 0x18b9 (decimal: 6329)
  table_entries: 0x2c (decimal: 44)

CMOS_OPTION_TABLE record at physical address 0x76b42018:
  tag: 0xc8 (decimal: 200)
  size: 0x294 (decimal: 660)
  data:

... Could this be more intuitive?
Star Labs coreboot-configurator

strongly inspired by or reworked copy of corevantage invoking nvramtool
System76 Firmware Info in Pop!_OS

System Firmware

System76 Lemur Pro (temp10)
2022-01-06_C730492

Changelog

2022-01-06_C73e482
Updated Intel microcode, enabled TPM measured boot, added CMOS option to configure IME mode, fixed Thunderbolt on Linux 5.13+, fixed UEFI boot options getting erased by CMOS option

2021-07-20_04c2809
Sync GPU and CPU fans, enable fan interpolation, fix missed keys and responsiveness in early boot, add scroll lock to keyboard layout

2021-04-07_236914e
Fix fan max keeping fan on when in S0iK, report all keys as released when lid is closed, and update microcode

2021-02-04_be04aea
Clear NVRAM when CMOS battery is removed and fix NVRAM compating

2020-12-06_0e80285
Original firmware
Ideas for Open Source Firmware
LinuxBoot

Simple

Add a splashscreen image, e.g., using the `fbsplash` command in u-root.
LinuxBoot

Simple
Add a splashscreen image, e.g., using the `fbsplash` command in u-root.

Advanced
Render an image around the TUI, possibly like `fbcondecor`.
# Appendix A

## Conventions for IFR to HTML Translation

Table A-2 defines suggested translations between IFR and HTML.

<table>
<thead>
<tr>
<th>IFR</th>
<th>HTML</th>
</tr>
</thead>
<tbody>
<tr>
<td>String in <code>form operand</code></td>
<td>Both <code>&lt;title&gt;</code> and <code>&lt;h1&gt;</code></td>
</tr>
<tr>
<td>Subtitle</td>
<td><code>&lt;h3&gt;</code></td>
</tr>
<tr>
<td>Text</td>
<td>Standard text</td>
</tr>
<tr>
<td>One-of</td>
<td>Either radio button or drop down</td>
</tr>
<tr>
<td>Checkbox</td>
<td>Single selection check box</td>
</tr>
<tr>
<td>Numeric</td>
<td>Text input sized to fit the maximum number of digits in the number along with JavaScript or equivalent validation</td>
</tr>
<tr>
<td>Password</td>
<td>No recommendation</td>
</tr>
<tr>
<td>Go-to</td>
<td><code>&lt;a href...&gt;</code></td>
</tr>
</tbody>
</table>
HII Concepts

HII: Key Concepts

- localization
- setup browser
- input sources
- forms & strings

Simulator

Encourage a “walk up and use” (WUU) user interface. Most applications are designed to be used repeatedly. User interface designers must trade off learnability for usability. The goal of WUU applications is to be instantly usable without a learning curve or other documentation.

Design characteristics include the following:

- A simplified interface.
- Continual display of both keys and context-sensitive help, rather than having the user ask for it.
- Minimal shortcuts (most people become confused by more than one method for doing things).
- An interface that is analogous to a common interface. At this time, a generic web browser is probably the most universal non-proprietary interface.

User Experience (UX) \(^{10}\)

Encourage a “walk up and use” (WUU) user interface. Most applications are designed to be used repeatedly. User interface designers must trade off learnability for usability. The goal of WUU applications is to be instantly usable without a learning curve or other documentation.

Encourage a “walk up and use” (WUU) user interface. Most applications are designed to be used repeatedly. User interface designers must trade off learnability for usability. The goal of WUU applications is to be instantly usable without a learning curve or other documentation.

Design characteristics include the following:

- A simplified interface.
- Continual display of both keys and context-sensitive help, rather than having the user ask for it.
- Minimal shortcuts (most people become confused by more than one method for doing things).
- An interface that is analogous to a common interface. At this time, a generic web browser is probably the most universal nonproprietary interface.

---

UEFI Configuration Namespace\textsuperscript{11}

Approach
- Form \{Builder, Generator\}
- schemas defined by spec
can be implemented in Fiedka
- Fiedka is based on Electron, i.e., a web browser with OS interfacing

\textsuperscript{11}https://uefi.org/namespace_instructions
IPC and RPC

We can build a local interface only, using IPC, or be more lax and provide a remote API for RPC.
Notes on Security and Safety

Principle of Least Privilege (PoLP)
Interfaces should guard from full access. Restricted access prevents accidents and compromise.

Robustness
Configuration means (user) input. Input must be validated.

Define fallbacks for resilience.
Notes on Security and Safety

Principle of Least Privilege (PoLP)
Interfaces should guard from full access.
Restricted access prevents accidents and compromise.
Notes on Security and Safety

**Principle of Least Privilege (PoLP)**
Interfaces should guard from full access.
Restricted access prevents accidents and compromise.

**Robustness**
Configuration means (user) input.
Input *must* be validated.
Define fallbacks for resilience.
Awareness

*Remember: User interfaces are critical!*
Awareness

*Remember: User interfaces are critical!*

Pick a user interface that fits the need, even if it seems old-fashioned.
Thanks!
Questions?