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

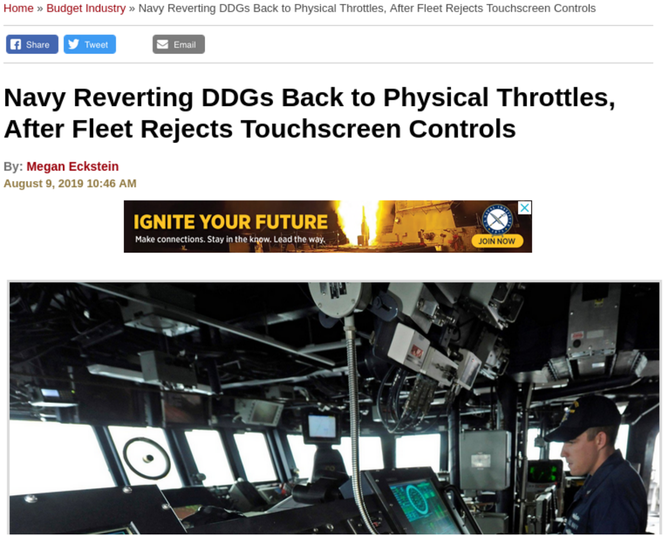
- hardware and firmware
- operating systems
- software distributions
- reverse engineering
- Fiedka the Firmware Editor

Fiedka

Fiedka is a graphical firmware editor app[1].

[1] https://fiedka.app/

User Interfaces are Critical[2]



Home » Budget Industry » Navy Reverting DDGs Back to Physical Throttles, After Fleet Rejects Touchscreen Controls

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

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

[2] https://news.usni.org/2019/08/09/navy-reverting-ddgs-back-to-physical-throttles-after-fleet-rejects-touchscreen-controls

User Interface Design

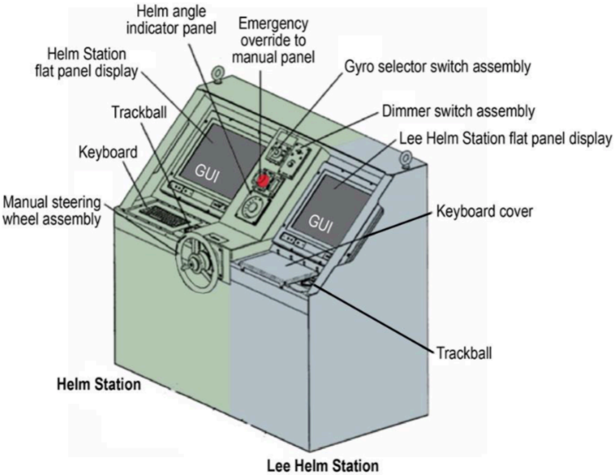

Helm angle indicator panel
Emergency override to manual panel
Helm Station flat panel display
Gyro selector switch assembly
Trackball
Dimmer switch assembly
Lee Helm Station flat panel display
Keyboard
GUI
Manual steering wheel assembly
GUI
Keyboard cover
Trackball
Helm Station
Lee Helm Station

Figure 4. *John S McCain* SCC. (Drawing from IBNS technical manual; color added by NTSB)

History

Early Firmware and Interfaces

[3] https://historyofinformation.com/detail.php?entryid=3846

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[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

[3] https://historyofinformation.com/detail.php?entryid=3846

[4] http://www.firmworks.com/www/ofw.htm

[5] https://www.intel.com/content/dam/www/public/us/en/documents/reference-guides/efi-human-interface-infrastructure-specification-v09.pdf

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

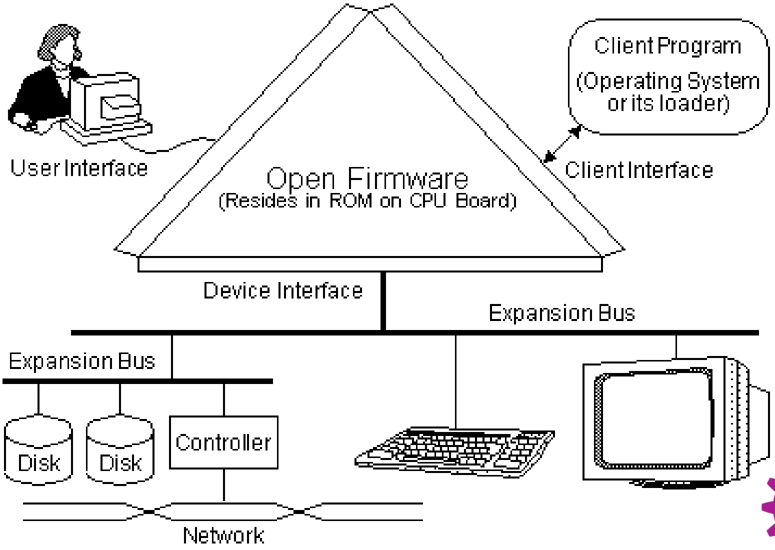
- Human Interface Infrastructure (HII)[5]
- standardized protocol and data structures for building forms

[3] https://historyofinformation.com/detail.php?entryid=3846

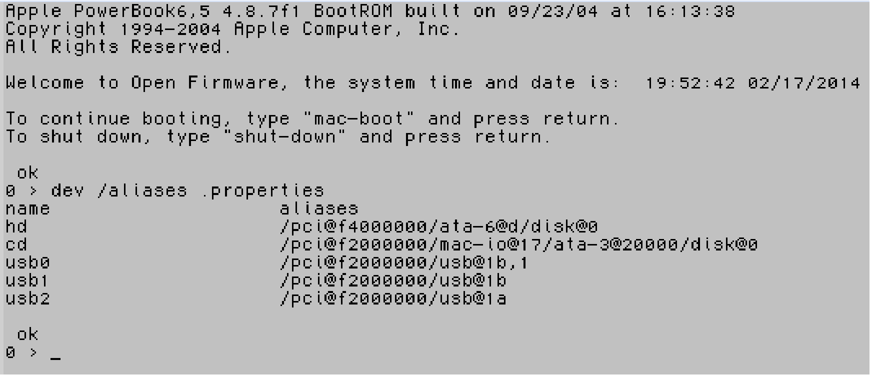
[4] http://www.firmworks.com/www/ofw.htm

[5] https://www.intel.com/content/dam/www/public/us/en/documents/reference-guides/efi-human-interface-infrastructure-specification-v09.pdf

Open Firmware Interfaces



Open Firmware Interactive Environment

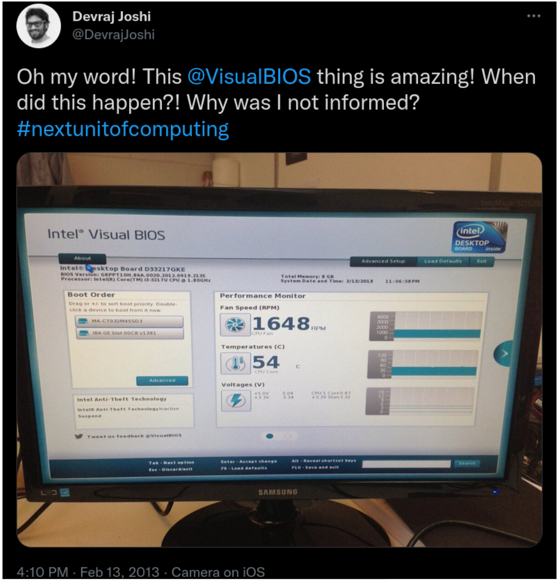


```
Apple PowerBook6,5 4.8.7f1 BootROM built on 09/23/04 at 16:13:38
Copyright 1994-2004 Apple Computer, Inc.
All Rights Reserved.

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.

 ok
0 > dev /aliases .properties
name                    aliases
hd                      /pci@f4000000/ata-6@d/disk@0
cd                      /pci@f2000000/mac-io@17/ata-3@20000/disk@0
usb0                    /pci@f2000000/usb@1b,1
usb1                    /pci@f2000000/usb@1b
usb2                    /pci@f2000000/usb@1a

 ok
0 > _
```

image originally from https://www.morphos-team.net/guide/usb-boot

see also https://www.youtube.com/watch?v=u9OMOHl73IE

Visual BIOS[6]



Devraj Joshi
@DevrajJoshi

Oh my word! This @VisualBIOS thing is amazing! When did this happen?! Why was I not informed? #nextunitofcomputing

4:10 PM · Feb 13, 2013 · Camera on iOS

[6] https://twitter.com/DevrajJoshi/status/301710041109639169

Modern Firmware Interfaces

NUI vs TUI vs GUI

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

[7] 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)

[7] 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)

[7] 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)[7]
- coreinfo (payload)
- corevantage, coreboot-configurator (GUIs)

LinuxBoot

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

U-Boot

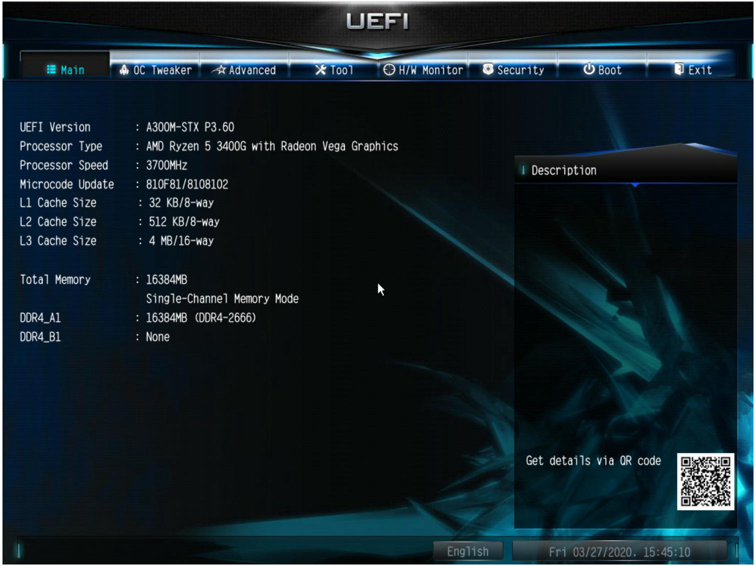
- interactive command interface

Tianocore / EDK2

- UEFI Shell
- Setup Browser (interactive menu, TUI)

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

Graphical Firmware User Interfaces



```
UEFI Version        : A300M-STX P3.60
Processor Type      : AMD Ryzen 5 3400G with Radeon Vega Graphics
Processor Speed     : 3700MHz
Microcode Update    : 810F81/8108102
L1 Cache Size        : 32 KB/8-way
L2 Cache Size        : 512 KB/8-way
L3 Cache Size        : 4 MB/16-way

Total Memory        : 16384MB
                      Single-Channel Memory Mode
DDR4_A1             : 16384MB (DDR4-2666)
DDR4_B1             : None
```

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

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:
...
```

coreboot nvramtool

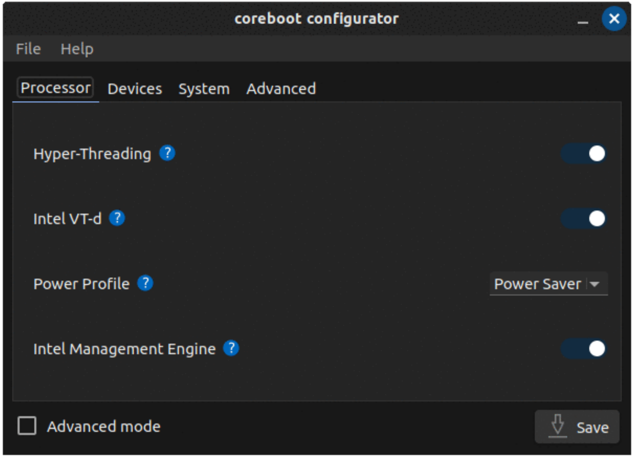
dump coreboot tables: nvramtool -d

```
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`



coreboot configurator

Hyper-Threading ?

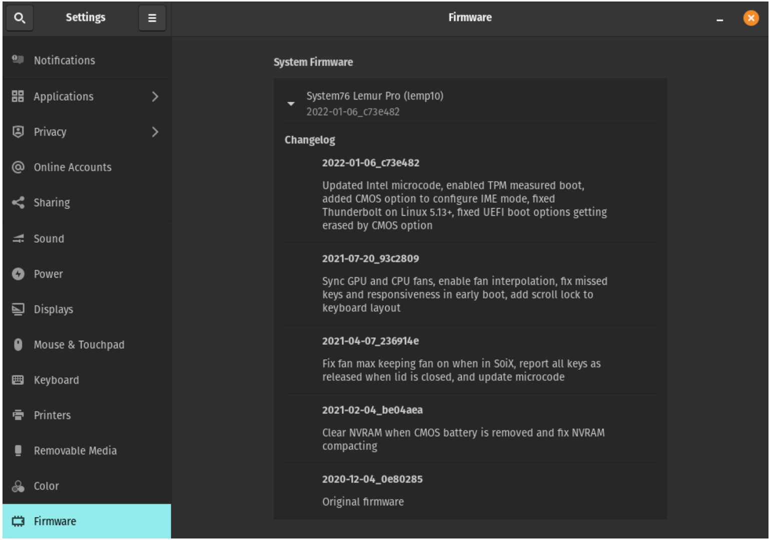
Intel VT-d ?

Power Profile ? Power Saver

Intel Management Engine ?

□ Advanced mode

System76 Firmware Info in Pop!_OS



System Firmware

System76 Lemur Pro (lemp10)
2022-01-06_c73e482

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_93c2809

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 S0iX, 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 compacting

2020-12-04_0e80285

Original firmware

Ideas for Open Source Firmware

LinuxBoot

Simple

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

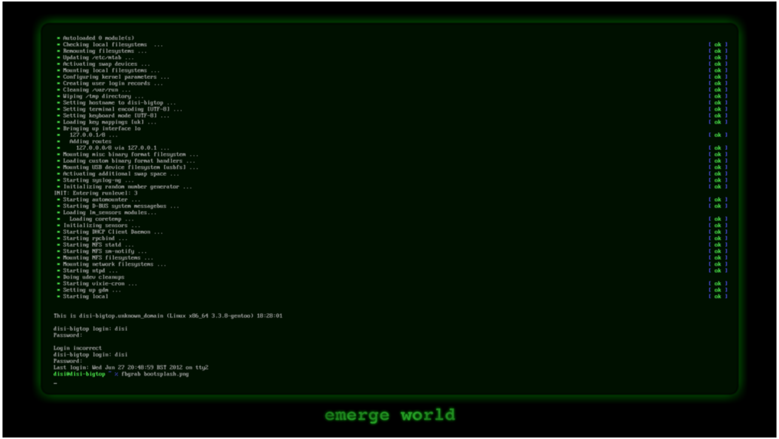
LinuxBoot

Simple

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

Advanced

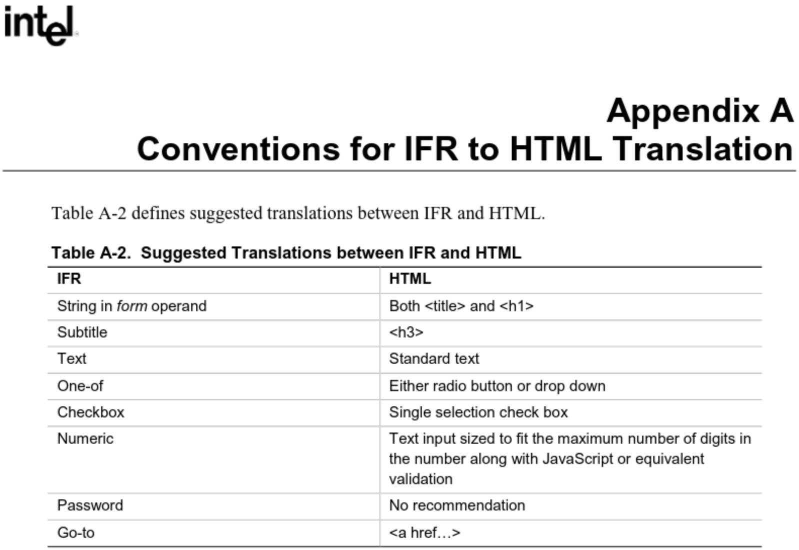
Render an image around the TUI, possibly like `fbcondecor`.



```
 * Autoloaded 0 module(s)
 * Checking local filesystems  ...                          [ ok ]
 * Remounting filesystems ...                                [ ok ]
 * Updating /etc/mtab ...                                    [ ok ]
 * Activating swap devices ...                               [ ok ]
 * Mounting local filesystems ...                            [ ok ]
 * Configuring kernel parameters ...                         [ ok ]
 * Creating user login records ...                           [ ok ]
 * Cleaning /var/run ...                                     [ ok ]
 * Wiping /tmp directory ...                                 [ ok ]
 * Setting hostname to disi-bigtop ...                       [ ok ]
 * Setting terminal encoding [UTF-8] ...                     [ ok ]
 * Setting keyboard mode [UTF-8] ...                         [ ok ]
 * Loading key mappings [uk] ...                             [ ok ]
 * Bringing up interface lo
 *   127.0.0.1/8 ...                                         [ ok ]
 *   Adding routes
 *     127.0.0.0/8 via 127.0.0.1 ...                         [ ok ]
 * Mounting misc binary format filesystem ...                [ ok ]
 * Loading custom binary format handlers ...                 [ ok ]
 * Mounting USB device filesystem [usbfs] ...                [ ok ]
 * Activating additional swap space ...                      [ ok ]
 * Starting syslog-ng ...                                    [ ok ]
 * Initializing random number generator ...                  [ ok ]
INIT: Entering runlevel: 3
 * Starting automounter ...                                  [ ok ]
 * Starting D-BUS system messagebus ...                      [ ok ]
 * Loading lm_sensors modules...
 *   Loading coretemp ...                                    [ ok ]
 * Initializing sensors ...                                  [ ok ]
 * Starting DHCP Client Daemon ...                           [ ok ]
 * Starting rpcbind ...                                      [ ok ]
 * Starting NFS statd ...                                    [ ok ]
 * Starting NFS sm-notify ...                                [ ok ]
 * Mounting NFS filesystems ...                              [ ok ]
 * Mounting network filesystems ...                          [ ok ]
 * Starting ntpd ...                                         [ ok ]
 * Doing udev cleanups
 * Starting vixie-cron ...                                   [ ok ]
 * Setting up gdm ...                                        [ ok ]
 * Starting local                                            [ ok ]


This is disi-bigtop.unknown_domain (Linux x86_64 3.3.8-gentoo) 18:28:01

disi-bigtop login: disi
Password:

Login incorrect
disi-bigtop login: disi
Password:
Last login: Wed Jun 27 20:48:59 BST 2012 on tty2
disi@disi-bigtop ~ % fbgrab bootsplash.png
_
```

emerge world

Back to HII…



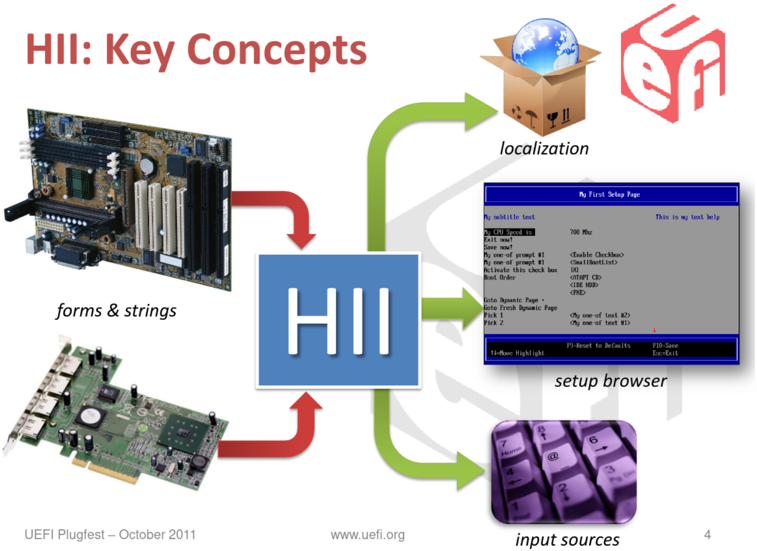
Appendix A
Conventions for IFR to HTML Translation

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

Table A-2. Suggested Translations between IFR and HTML

IFR	HTML
String in *form* operand	Both <title> and <h1>
Subtitle	<h3>
Text	Standard text
One-of	Either radio button or drop down
Checkbox	Single selection check box
Numeric	Text input sized to fit the maximum number of digits in the number along with JavaScript or equivalent validation
Password	No recommendation
Go-to	<a href…>

HII Concepts[8]



HII: Key Concepts

forms & strings

HII

localization

setup browser

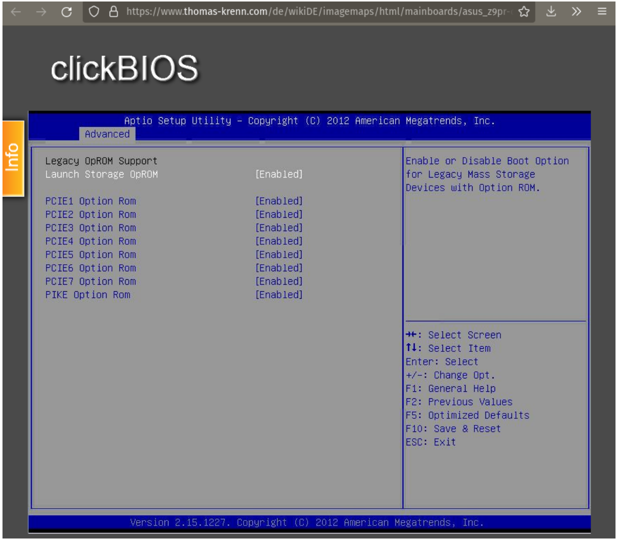
input sources

```
                     My First Setup Page

My subtitle text                            This is my text help

My CPU Speed is         700 Mhz
Exit now!
Save now!
My one-of prompt #1     <Enable Checkbox>
My one-of prompt #1     <SmallBootList>
Activate this check box [X]
Boot Order              <ATAPI CD>
                        <IDE HDD>
                        <PXE>
Goto Dynamic Page ▸
Goto Fresh Dynamic Page
Pick 1                  <My one-of text #2>
Pick 2                  <My one-of text #1>

                 F9-Reset to Defaults    F10-Save
 ↑↓=Move Highlight                       Esc=Exit
```

UEFI Plugfest – October 2011   www.uefi.org   4

[8] https://uefi.org/sites/default/files/resources/UEFI_Plugfest_2011Q4_P4_Intel.pdf

Simulator[9]


https://www.thomas-krenn.com/de/wikiDE/imagemaps/html/mainboards/asus_z9pr-
clickBIOS
Info

```
Aptio Setup Utility – Copyright (C) 2012 American Megatrends, Inc.
    Advanced

Legacy OpROM Support                              Enable or Disable Boot Option
Launch Storage OpROM          [Enabled]           for Legacy Mass Storage
                                                  Devices with Option ROM.
PCIE1 Option Rom              [Enabled]
PCIE2 Option Rom              [Enabled]
PCIE3 Option Rom              [Enabled]
PCIE4 Option Rom              [Enabled]
PCIE5 Option Rom              [Enabled]
PCIE6 Option Rom              [Enabled]
PCIE7 Option Rom              [Enabled]
PIKE Option Rom               [Enabled]

                                                  →←: Select Screen
                                                  ↑↓: Select Item
                                                  Enter: Select
                                                  +/-: Change Opt.
                                                  F1: General Help
                                                  F2: Previous Values
                                                  F5: Optimized Defaults
                                                  F10: Save & Reset
                                                  ESC: Exit

Version 2.15.1227. Copyright (C) 2012 American Megatrends, Inc.
```

[9] https://www.thomas-krenn.com/de/wikiDE/imagemaps/html/mainboards/asus_z9pr-d12_4l/pcie_slot_option_rom_configuration.php

User Experience (UX)[10]

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.*

[10] https://www.intel.com/content/dam/www/public/us/en/documents/reference-guides/efi-human-interface-infrastructure-specification-v09.pdf

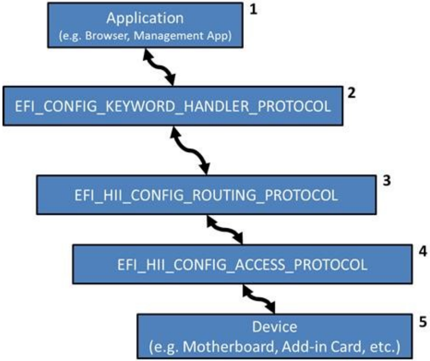
User Experience (UX)[10]

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.
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.

[10] https://www.intel.com/content/dam/www/public/us/en/documents/reference-guides/efi-human-interface-infrastructure-specification-v09.pdf

UEFI Configuration Namespace[11]



Application
(e.g. Browser, Management App) 1

EFI_CONFIG_KEYWORD_HANDLER_PROTOCOL 2

EFI_HII_CONFIG_ROUTING_PROTOCOL 3

EFI_HII_CONFIG_ACCESS_PROTOCOL 4

Device
(e.g. Motherboard, Add-in Card, etc.) 5

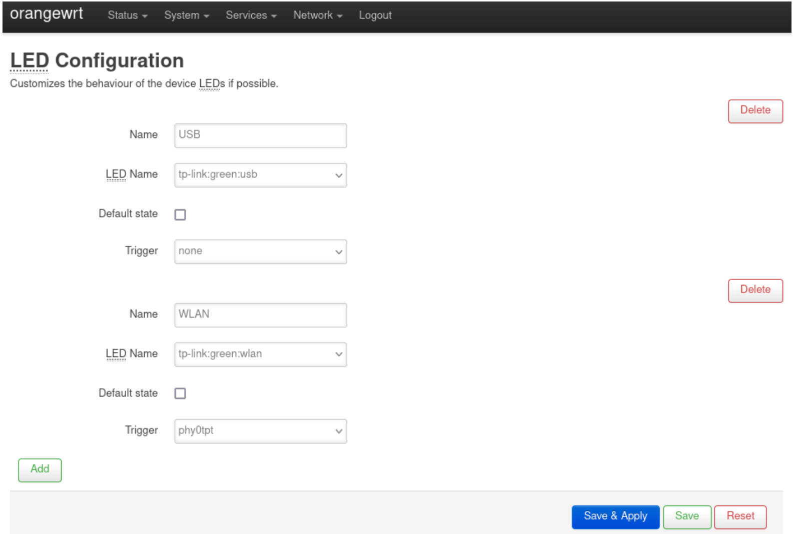
Approach

- 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

[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.



LED Configuration

Customizes the behaviour of the device LEDs if possible.

Name: USB

LED Name: tp-link:green:usb

Default state: □

Trigger: none

Name: WLAN

LED Name: tp-link:green:wlan

Default state: □

Trigger: phy0tpt

Notes on Security and Safety

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?