Nerves Training Cheat Sheet

Terminology

host The computer on which you are editing source code, compiling, and assembling firmware

target The computer that runs the compiled code such as a Raspberry Pi Zero, Beaglebone, or other board.

toolchain The tools required to build code for the target, such as compilers, linkers, binutils, and C runtime

system A lean Buildroot-based Linux root filesystem that has been customized and cross-compiled for a particular target

firmware bundle A single file that contains the system, application, configuration and anything else needed to initialize and update the non-volatile storage of a device

firmware image A large binary file generated from the firmware bundle that is a bit-for-bit image of a devices non-volatile memory

Installation

https://hexdocs.pm/nerves/installation.html

MacOS

```
$ brew update
$ brew install fwup squashfs coreutils
```

Linux (Debian/Ubuntu)

```
$ sudo apt install ssh-askpass squashfs-tools
```

All platforms

Update/install asdf

```
$ asdf install erlang 20.0
$ asdf install elixir 1.4.5
$ asdf global erlang 20.0
$ asdf global elixir 1.4.5

$ mix local.hex
$ mix local.rebar
$ mix archive.install \
  https://github.com/nerves-project/
   archives/raw/master/nerves_bootstrap.ez
```

or if you just need to update the archive

```
$ mix local.nerves
```

Training files

All files for training are available on the test network (See your table tent for WiFi SSID/password).

Go to ftp://192.168.11.1 and download nerves_dl.zip. Unzip in your home directory. It puts files in *.nerves/dl*.

Nerves basics

Create a new project

```
$ mix nerves.new hello nerves
$ cd hello nerves
$ export MIX TARGET=<mix target>
$ mix deps.get
$ mix firmware
```

Burn a MicroSD card

```
$ mix firmware.burn
```

Update using nerves_firmware_ssh

```
$ mix firmware.push hostname
```

Connecting to the target

Nerves sends the iex prompt over a virtual serial port on the USB cable. It shows up as a device like /dev/tty.usbmodem or /dev/ttyACM0.

screen

```
$ screen /dev/tty<device>
```

Exit screen with CTRL+a, CTRL+\

picocom

```
$ picocom /dev/tty<device>
```

Exit picocom with CTRL+a, CTRL+x

Creating the example projects

Creating the starter nerves_init_gadget project

```
$ mix nerves.new starter
$ cd starter
$ export MIX TARGET=rpi0
```

Add the following deps to mix.exs:

```
{:nerves_init_gadget, "~> 0.2"}
```

Configure bootloader to start :nerves_init_gadget in config.exs:

```
config :bootloader,
  init: [:nerves_runtime,
         :nerves_init_gadget],
  app: :starter
```

Add your ssh key or keys to config.exs:

```
config :nerves_firmware_ssh,
  authorized_keys: [
    "ssh-rsa AAAAB3N…",
    "ssh-rsa another one",
  ]

$ mix deps.get
$ mix firmware
$ mix firmware.burn
```

Training Git repositories

Base URL:

```
https:// bitbucket.org/fhunleth/
```

Repositories:

```
nervestraining-starter.git
nervestraining-basic_io.git
nervestraining-barometer.git
nervestraining-mysystem.git
nervestraining-webui.git
nervestraining-nerves_cam.git
```

Useful IEx commands

Install Nerves helpers	`use Nerves.Runtime.Helpers`
	`Nerves.Runtime.Helpers .install`

Run a Linux command	`cmd("ps") or`
	`:os.cmd('ps') |> IO.puts`

Reboot	`reboot! or`
	`Nerves.Runtime.reboot`

Shell	`<CTRL+G>s sh<Enter>c 1`

Show as hex	`hex(1234) or`
	`inspect(1234, base: :hex)`

Hostname	`:inet.gethostname`
Logging level	`Logger.configure level: :warn`
Reverse search	`<CTRL+R>text to find`
Refresh line	`<CTRL+L>`

Binary pattern matching

```
<<variable::[modifiers-]type[-size()], ...>>
```

Except for binaries, size is specified in bits.

Types

`integer, float, bits, bitstring, binary, bytes, utf8, utf16, utf32`

Modifiers

`signed,` **`unsigned`**`, little,` **`big`**`, native`

```
iex> h <<>>
```

Built with Nerves

Nerves Training Cheat Sheet

Networking

Runtime

```
Nerves.Network.setup "wlan0", \
    ssid: "nerves1", key_mgmt: :"WPA-PSK",\
    psk: "nervestraining"
```

Defaults

```
config :nerves_network, :default,
  wlan0: [
    ssid: "nerves1",
    psk: "nervestraining,
    key_mgmt: :"WPA-PSK"
  ]
```

Check settings

```
Nerves.Network.status "wlan0"

SystemRegistry.match(:_)
|> get_in([:state, :network_interface,\
          "wlan0"])
```

nerves.system.shell

```
$ ELIXIR_ERL_OPTIONS="+Bc" mix \
    nerves.system.shell
```

Modify Linux configuration

```
/nerves/build > make linux-menuconfig
/nerves/build > make linux-savedefconfig
/nerves/build > cp build/linux-*/defconfig \
    ../env/custom_rpi0/linux-4.4.defconfig
(on Linux) > cp build/linux-*/defconfig \
    ../../../linux-4.4.defconfig
```

Modify Buildroot configuration

```
/nerves/build > make menuconfig
/nerves/build > make saveconfig
```

Erlang distribution

Add this to your config/config.exs

```
config :nerves_init_gadget,
  node_name: "nerves"
```

Add runtime_tools to your mix extra_applications

```
def application(_target) do
 [mod: {Starter.Application, []},
 extra_applications:[:logger,:runtime_tools]]
end
```

Remsh

```
$ iex --name me@0.0.0.0 \
      --cookie <from rel/vm.args> \
      --remsh nerves@nerves.local
```

Custom Buildroot package example

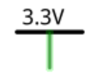
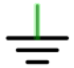
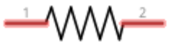
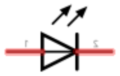
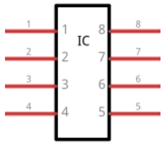
https://bitbucket.org/fhunleth/helloworld

http://buildroot.org/manual.html

Troubleshooting

1. If compilation doesn't work, check your MIX_TARGET setting. For training, MIX_TARGET=rpi0 most of the time.
2. Is the USB cable plugged into the USB port and not the PWR port?
3. Did you run *mix release*? Try *mix firmware* instead.
4. Did you swap boards with someone and the ssh keys no longer match?
5. Check connections. Maybe something is lose or plugged into the wrong slot.
6. Unplug and replug the USB cable. Check that something changes on your laptop. On Linux, run *dmesg*. On Mac, check if the /dev/tty file disappears and appears.
7. Check that the MicroSD card didn't pop out.

Common symbols on schematics

	Positive terminal of the power supply. Sometimes labelled VCC or VDD. "3V3" means 3.3V.
	Ground terminal of the power supply. Often abbreviated GND.
	Resistor
	Light emitting diod
	Integrated circuit. Similar symbol is used for connectors and other parts with lots of pins.

Nerves Training Bill of Materials

July 21, 2017

Part Name	Quantity	Vendor	Notes
Raspberry Pi Zero W	1	MicroCenter, Adafruit, etc.	
GrovePi Zero	1	Dexter Industries, Amazon	
Grove - Red LED	1	Seeed Studios	Can substitute any digital output sensor like a buzzer
Grove - Light Sensor v1.2	1	Seeed Studios	Can substitute any analog input sensor
Grove - Barometer Sensor (BMP280)	1	Seeed Studios	
Raspberry Pi Camera V2	1	MicroCenter, Adafruit, etc.	A version 1 camera works fine as well
Raspberry Pi Zero to Camera cable	1	MicroCenter, Adafruit, etc.	The cameras come with a ribbon cable for the other Raspberry Pis. The Zero requires a special adapter cable.
MicroSD card	1	Almost anywhere	Minimum 4 GB. Class 10 cards and faster are nice, but not required
MicroUSB cable	1	Monoprice, etc.	
40 pin GPIO headers	1	Pololu	The usual route is to buy an 80 pin breakaway header and break it in half
3/4" M-F standoffs 4-40 for GrovePi	4	Pololu	If imperial isn't available, substitute an 18.6mm or longer M2.5 standoff
3/8" M-F standoffs 4-40 for front plate	4	Pololu	
4-40, 1/4" machine screws	4	Pololu	Substitute M2.5 screws
2-56, 3/8" machine screws	8	Pololu	Substitute M2 screws
2-56, hex nuts	8	Pololu	
4-40, hex nuts	4	Pololu	
Lasercut front plate	1	Custom	These were custom made for the class. The design is available if you'd like to make your own.
3/8" x 5/8" labels	1	Any office supply store	Optional component for covering the "PWR IN" connector to avoid plugging into it accidentally
5" x 3" x 3" gift box	1	Amazon	Hold parts and final assembly

Built with Nerves