Terminology

host	The computer on which you are editing source code, compiling, and assembling firmware
target	The platform for which your firmware is built (for example, Raspberry Pi, Raspberry Pi 2, or Beaglebone Black)
toolchain	The tools required to build code for the target, such as compilers, linkers, binutils, and C runtime
system	A lean Buildroot-based Linux distribution that has been customized and cross-compiled for a particular target
assemble	The process of combining system, application, and configuration into a firmware bundle
firmware bundle	A single file that contains an assembled version of everything needed to burn firmware
firmware image	Built from a firmware bundle and contains the partition table, partitions, bootloader, etc.

Installation

MacOS

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

Linux

```
$ sudo apt-get install ssh-as kpass squash fs- tools
```

All platforms

```
$ mix local.hex
$ mix local.r ebar
$ mix archiv e.i nstall https: //g ith ub.c om /ne rve s-p roj ect /ar chi ves /ra w/m ast er/ ne
r ves _bo ots tra p.ez`
```

Updating

```
$ mix local.n erves
```

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

Tutorial projects

Pi Camera

```
git clone https: //f hun let h@b itb uck et.o rg /fh unl eth /ne
 rve s_c am.git
```

Official targets

Target	Nerves System	mix
Raspberry Pi Zero	nerves_system_rpi0	rpi0
Raspberry Pi A, B	nerves_system_rpi	rpi
Raspberry Pi 2	nerves_system_rpi2	rpi2
Raspberry Pi 3	nerves_system_rpi3	rpi3
All BeagleBones	nerves_system_bbb	bbb
Lego EV3	nerves_system_ev3	ev3
Linkit Smart	nerves_system_linkit	linkit

Nerves basics

Create a new project

```
$ mix nerves.new hello_ nerves
```

Build firmware bundle

```
$ cd hello_ nerves
$ export MIX_TA RGE T=<mix target>
$ mix deps.get
$ mix firmware
```

Burn a MicroSD card

```
$ mix firmwa re.burn
```

Update using nerves_firmware_http

```
$ mix firmwa re.push hostname [--target <mix target
>]
```

Connecting to the target

Most Nerves systems provide an IEx prompt over a serial port or UART. TTY emulators like `screen` and `picocom` can access it. Check the system for baud rate (normally 115200).

screen

```
$ screen /dev/t ty< dev ice> 115200
```

Exit screen with CTRL+a, CTRL+\

picocom

```
$ picocom -b 115200 /dev/t ty< dev ice>
```

Exit picocom with CTRL+a, CTRL+x

Useful IEx commands

Run a Linux command	`:os.cm d('ps') |> IO.puts`
Reboot	`Nerves.Ru nti me.r eboot`
Shell	`<CT RL+ g>s sh< Ent er>c 1`