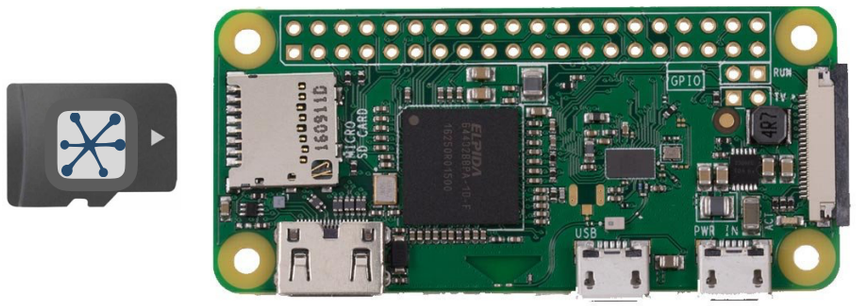
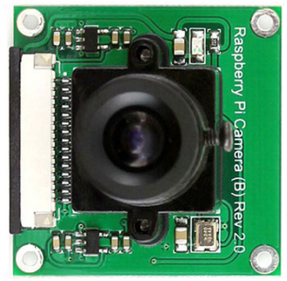
ELIXIRCONF™ 2017

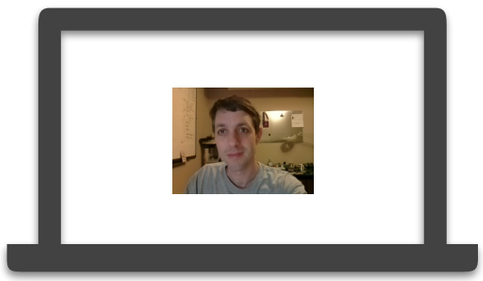
Nerves Training
Part 6: Streaming video

Goal





USB w/ virtual serial port
and virtual Ethernet



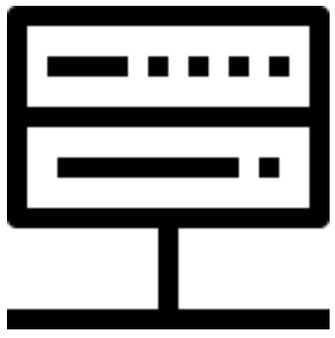
Streaming video

- Lots of options and tradeoffs
- High latency protocols
  - Movies and live “TV” content (i.e., Netflix)
    - Can support CDNs for distribution
    - Apple HLS, Microsoft Smooth Streaming, Adobe HDS
- Low latency protocols
  - Video conferencing and webcam streaming
  - Usually optimized for delivery to a small number of viewers

Low latency streaming

- Video encoding
  - H.264 - almost ubiquitous now
  - Motion JPEG - still around
- Transport
  - RTP - Real-time Transport Protocol (UDP)
  - WebRTC - stream to browsers
  - HTTP

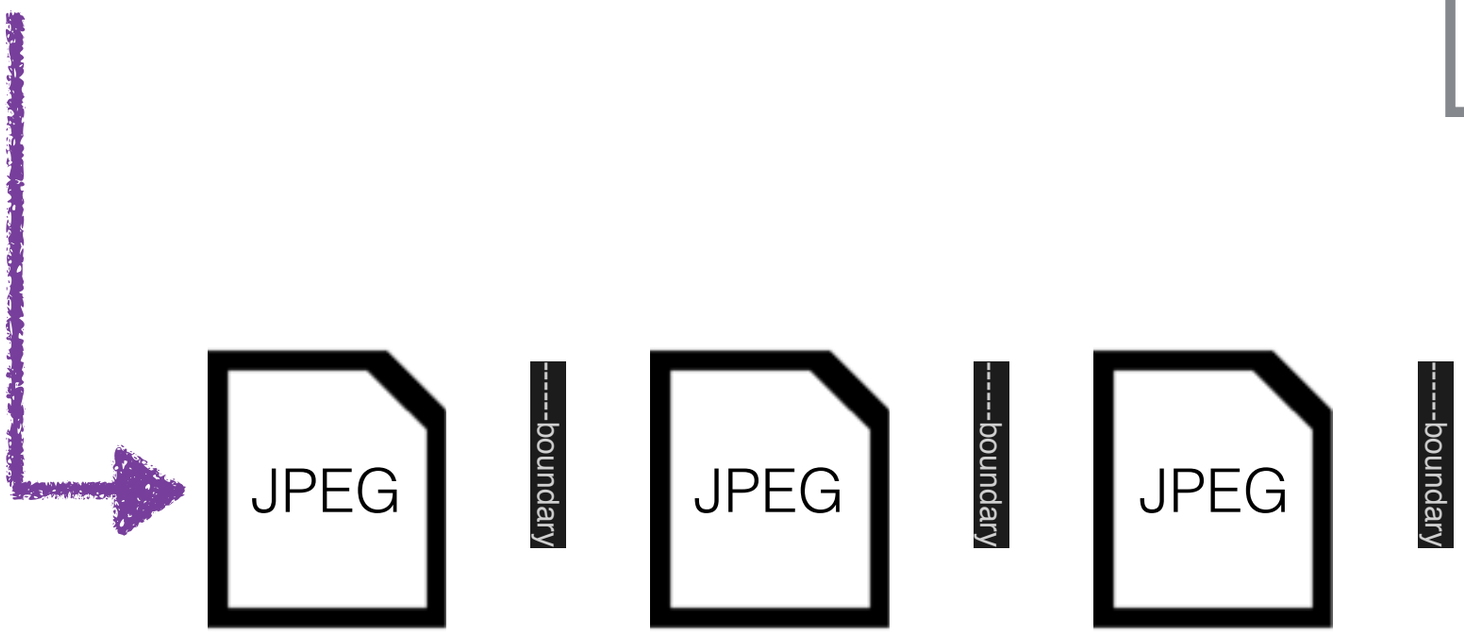
MJPEG over HTTP



```
<html>

  <head>
    <title>Video!</title>
  </head>

  <body>
    <img src="video.mjpg" />
  </body>
</html>
```

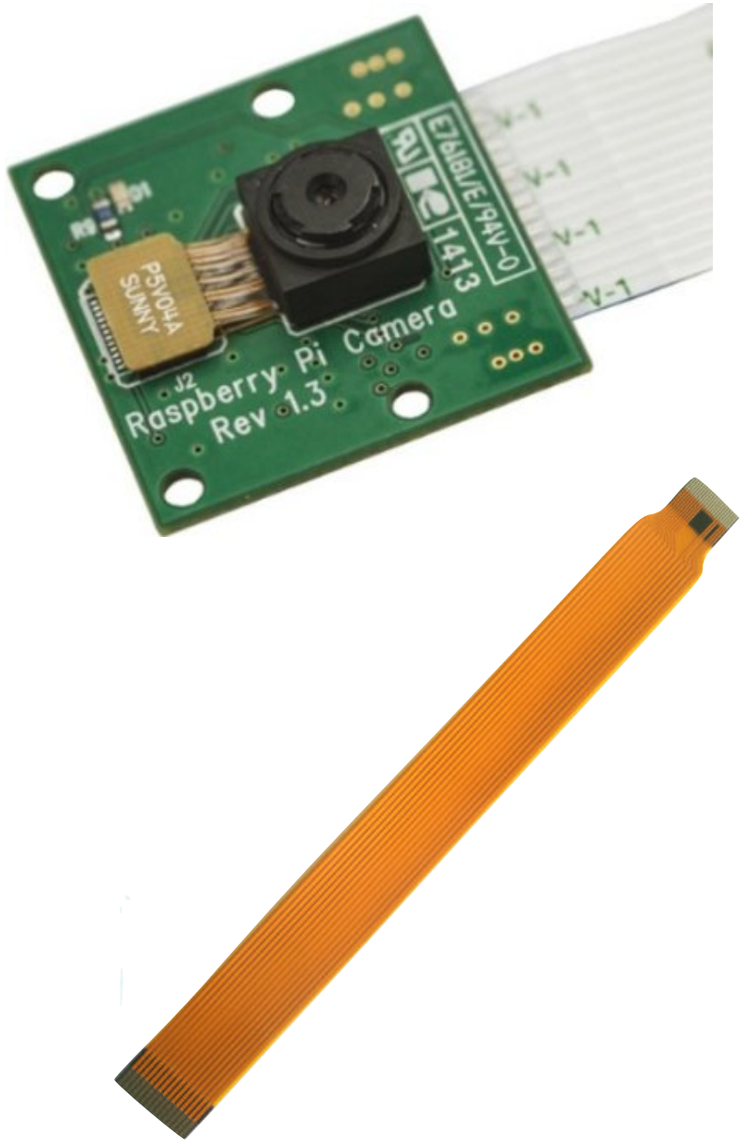


MJPEG over HTTP

- Pros
  - Not much code
  - It's just JPEGs
- Cons
  - High bandwidth for low resolution
  - No timing information
  - Head of line blocking

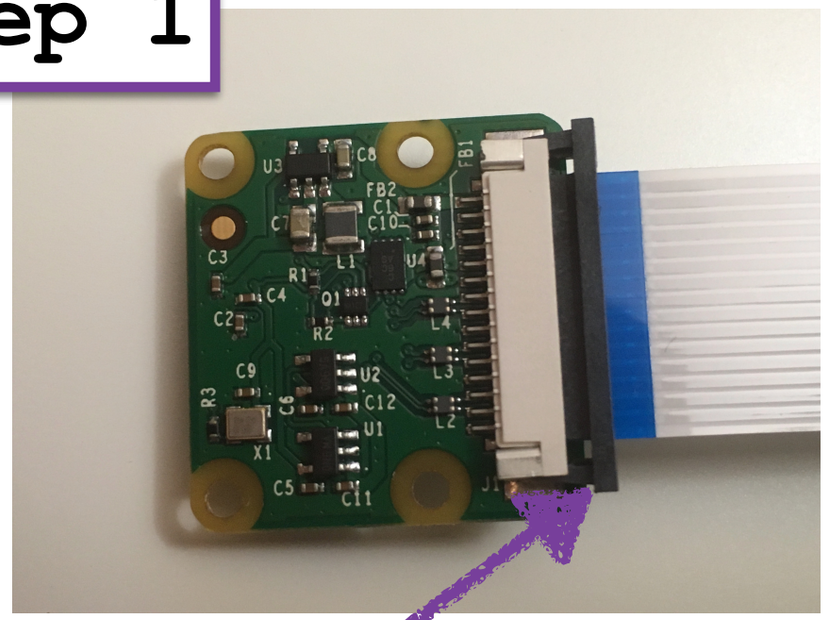
Attach the camera

- Raspberry Pi Camera V2
- Raspberry Pi Zero to camera cable
- Two 2-56 3/8" screws
- Two 2-56 hex nuts



Attach the camera

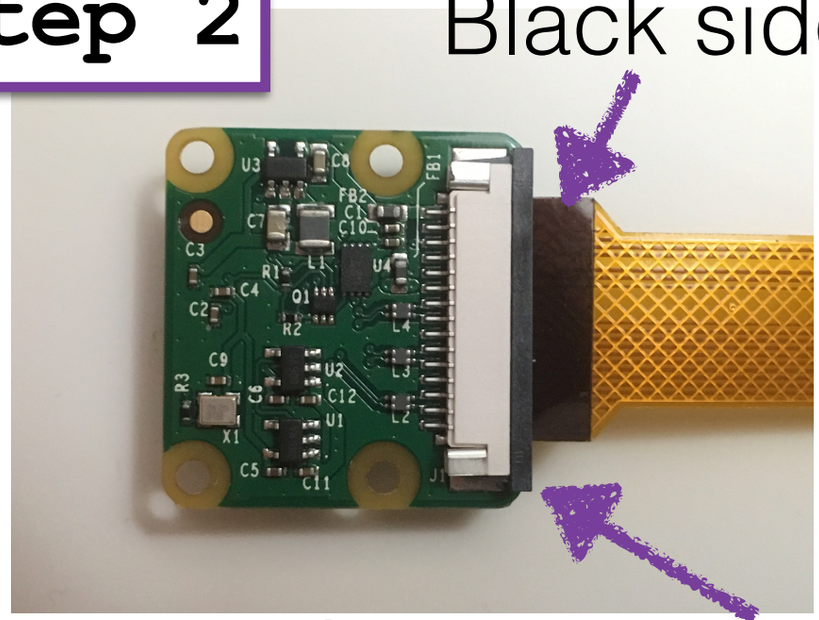
Step 1



Loosen connector and remove cable

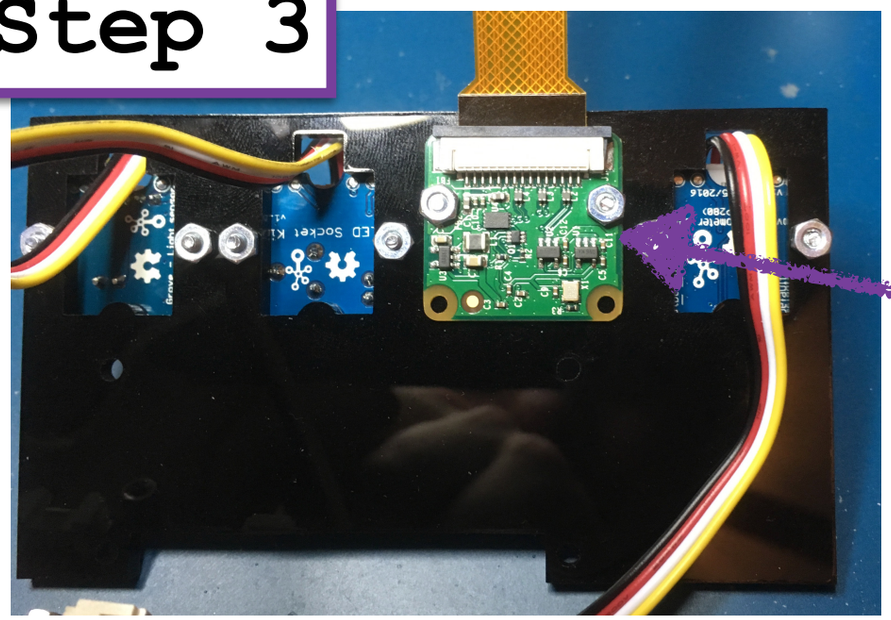
Step 2

Black side



Insert snuggly and close connector

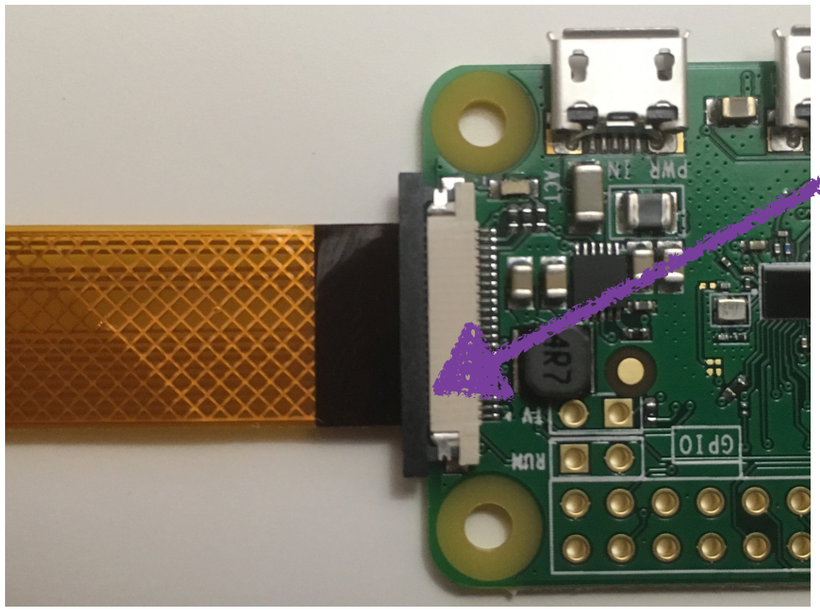
Step 3



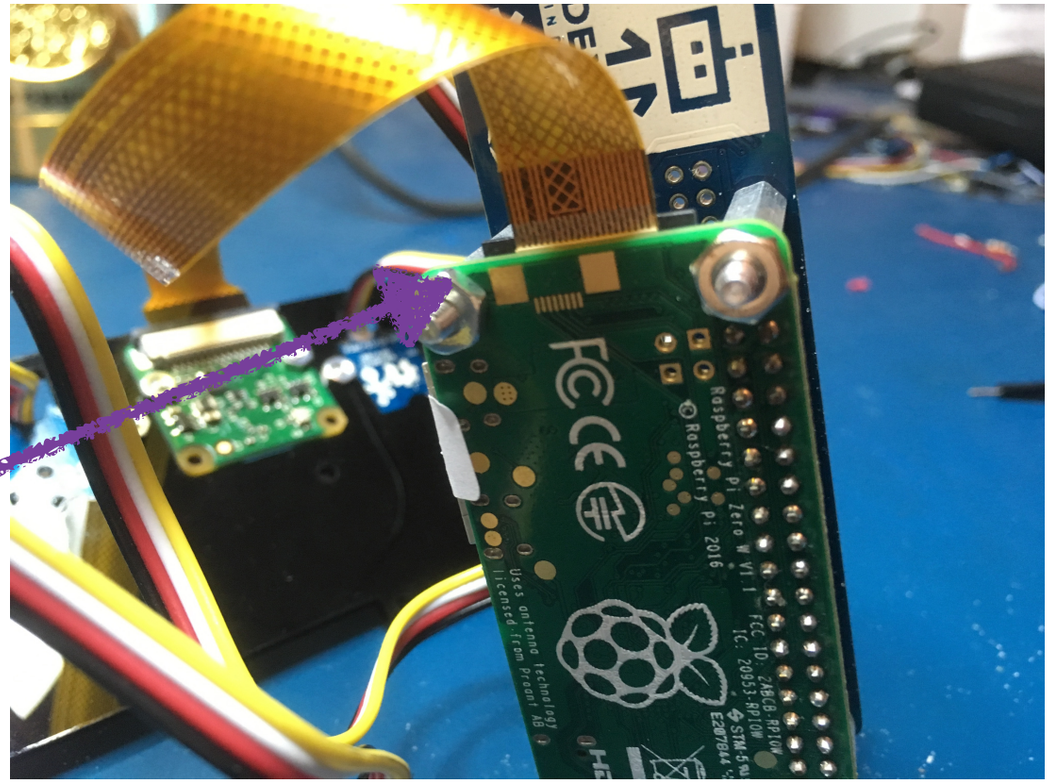
Carefully fasten to front plate with 2-56 screws

Connect to the RPi Zero W

Remove the LED (Port D3) and Barometer (I2C Port) connectors



Gently loosen black plastic piece and insert ribbon cable

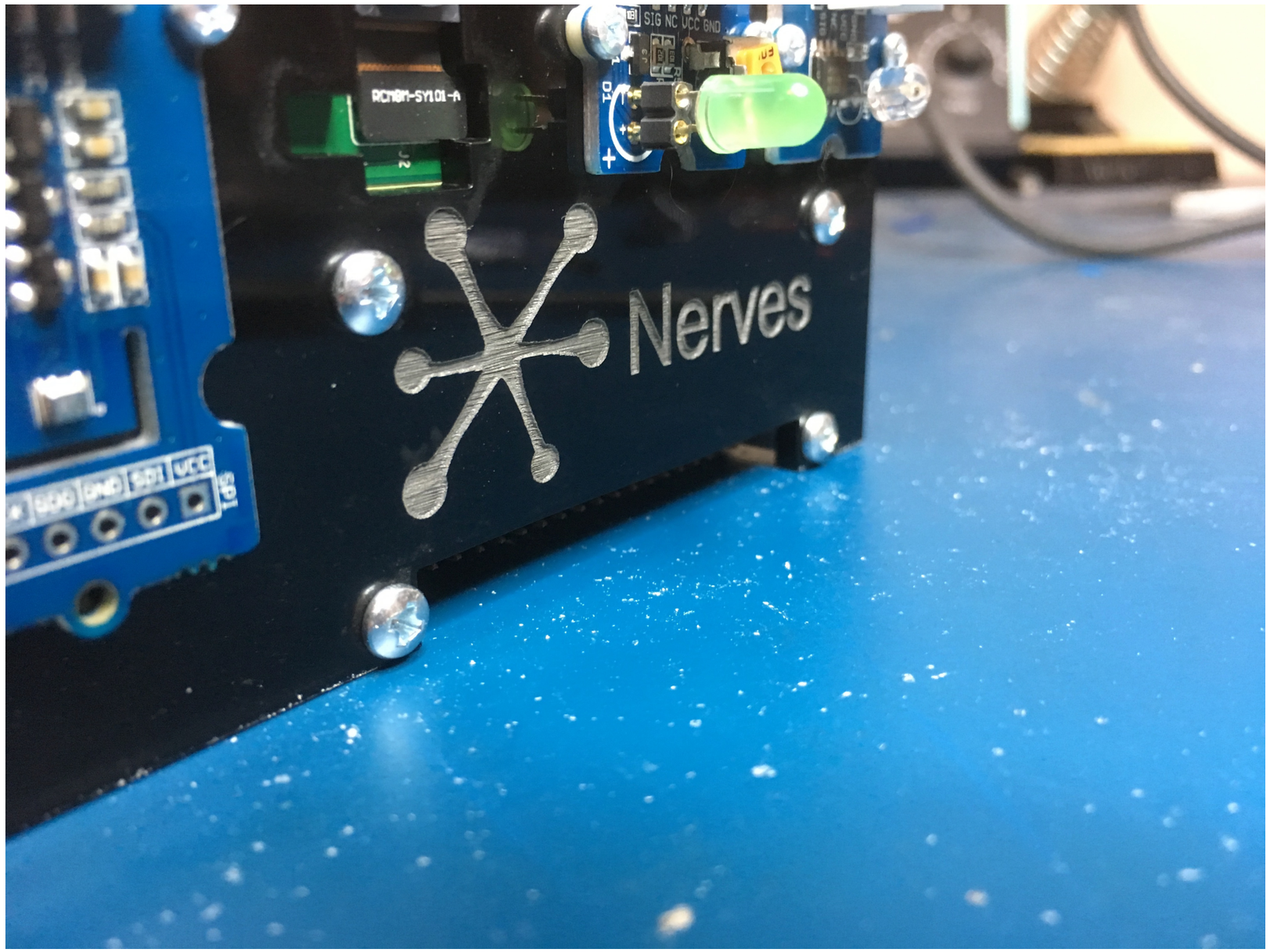


Close connector so that ribbon cable is locked

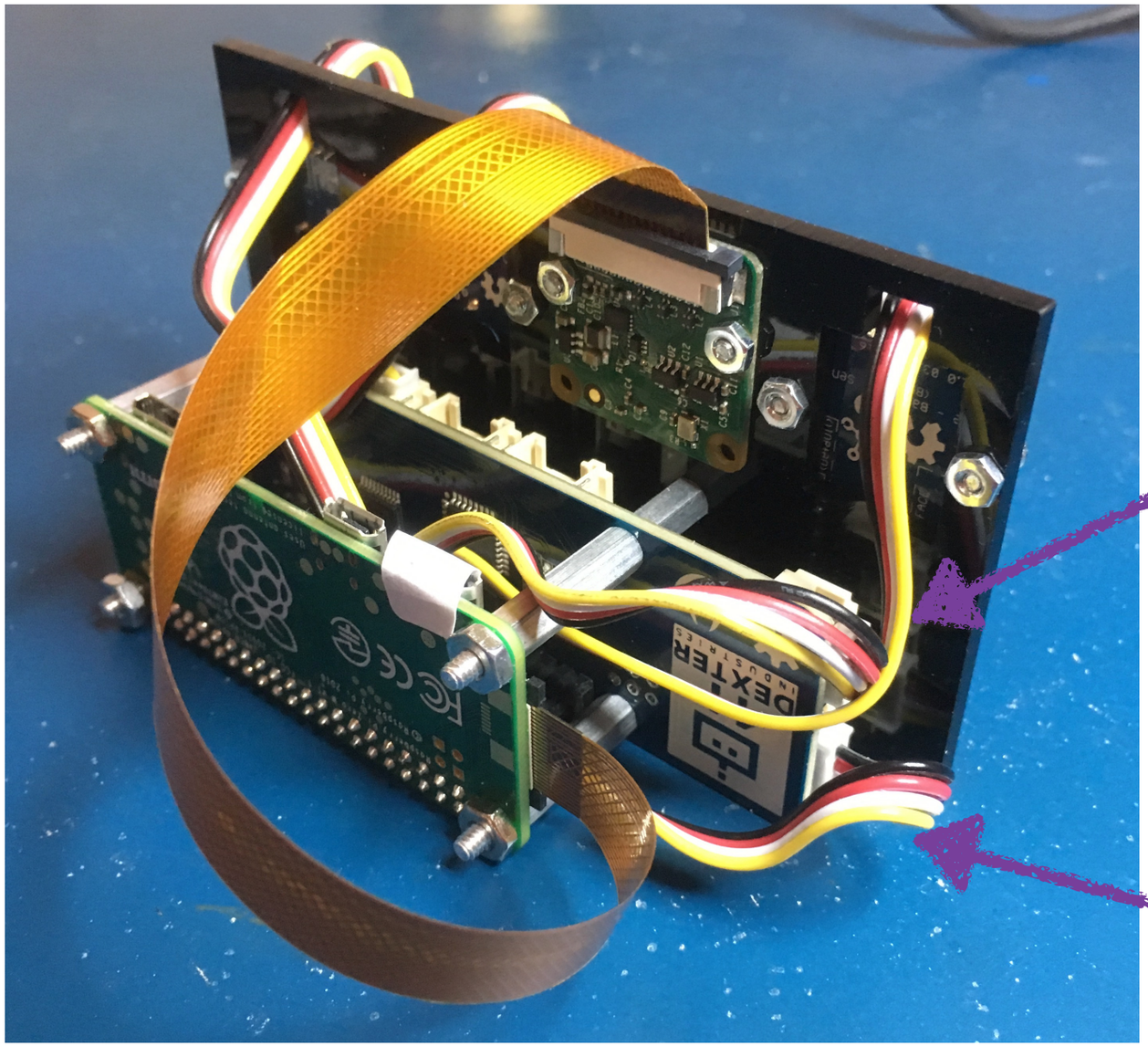
Attach the faceplate to the GrovePi Zero/RPi Zero W

- Four 4-40 screws

Attach the faceplate



Reconnect wires and tuck them between GrovePi



LED goes to Port D3 (top one)

Barometer goes to the I2C Port (bottom one)

Create a new nerves_init_gadget project

```
$ cd <workspace>
$ mix nerves.new nerves_cam
$ cd nerves_cam

add nerves_init_gadget to mix.exs and config.exs
copy ssh key config to config/config.exs
```

Skip typing

```
$ git clone \
https://bitbucket.org/fhunleth/nervestraining-nerves_cam.git \
nerves_cam
git checkout step1
```

Add picam and plug to mix.exs

```
def deps(target) do
  [ system(target),
    {:nerves_runtime, "~> 0.3.0"},
    {:bootloader, "~> 0.1"},
    {:nerves_init_gadget, github: “fhunleth/nerves_init_gadget", branch: "master"},
    {:picam, "~> 0.1"},
    {:cowboy, "~> 1.0.0"},
    {:plug, "~> 1.0"}
  ]
end
```

Skip typing

```
$ git clone \
https://bitbucket.org/fhunleth/nervestraining-nerves_cam.git \
nerves_cam
git checkout step2
```

lib/nerves_cam/application.ex

```
# Define workers and child supervisors to be supervised
children = [
  worker(Picam.Camera, []),
  Plug.Adapters.Cowboy.child_spec(:http, NervesCam.Router, [], [port: 80]),
]
```

Skip typing

```
$ git checkout step2
```

lib/nerves_cam/router.ex

```
defmodule NervesCam.Router do
  use Plug.Router

  plug :match
  plug :dispatch

  get "/" do
    markup = """
    <html>
    <head><title>Picam Video Stream</title></head>
    <body>
      Coming soon!
    </body>
    </html>
    """
    conn
    |> put_resp_header("Content-Type", "text/html")
    |> send_resp(200, markup)
  end
end
```

Skip typing

```
$ git checkout step2
```

Try it out

```
$ mix firmware

$ mix firmware.push nerves.local
   -or-
$ ./upload

Open http://nerves.local/ up in your browser
```

Are we going to take pictures yet?

lib/nerves_cam/router.ex

```
get "/" do
   markup = """
   <html>
   <head><title>Picam Video Stream</title></head>
   <body>
     <img src="video.mjpg" />
   </body>
   </html>
   """
   conn
   |> put_resp_header("Content-Type", "text/html")
   |> send_resp(200, markup)
end

forward "/video.mjpg", to: NervesCam.Streamer
```

```
$ git checkout step3
```

lib/nerves_cam/streamer.ex

```
defmodule NervesCam.Streamer do
  import Plug.Conn
  @behaviour Plug

  def init(opts), do: opts
  def call(conn, _opts) do
    conn
    |> put_resp_header("Age", "0")
    |> put_resp_header("Cache-Control", "no-cache, private")
    |> put_resp_header("Pragma", "no-cache")
    |> send_picture
  end

  defp send_picture(conn) do
    jpg = Picam.next_frame
    conn
    |> put_resp_header("Content-Type", "image/jpeg")
    |> send_resp(200, jpg)
  end
end
```

```
$ git checkout step3
```

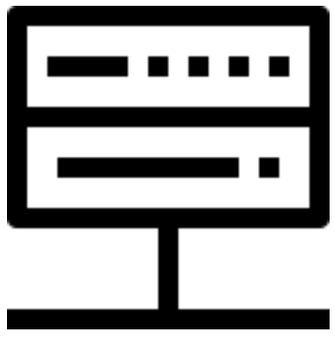
Try it out

```
$ export MIX_TARGET=rpi0
$ mix firmware

Connect the Raspberry Pi Zero and ping it
$ mix firmware.push nerves.local

Wait for the Raspberry Pi to reboot
Open a browser and try loading http://nerves.local/
```

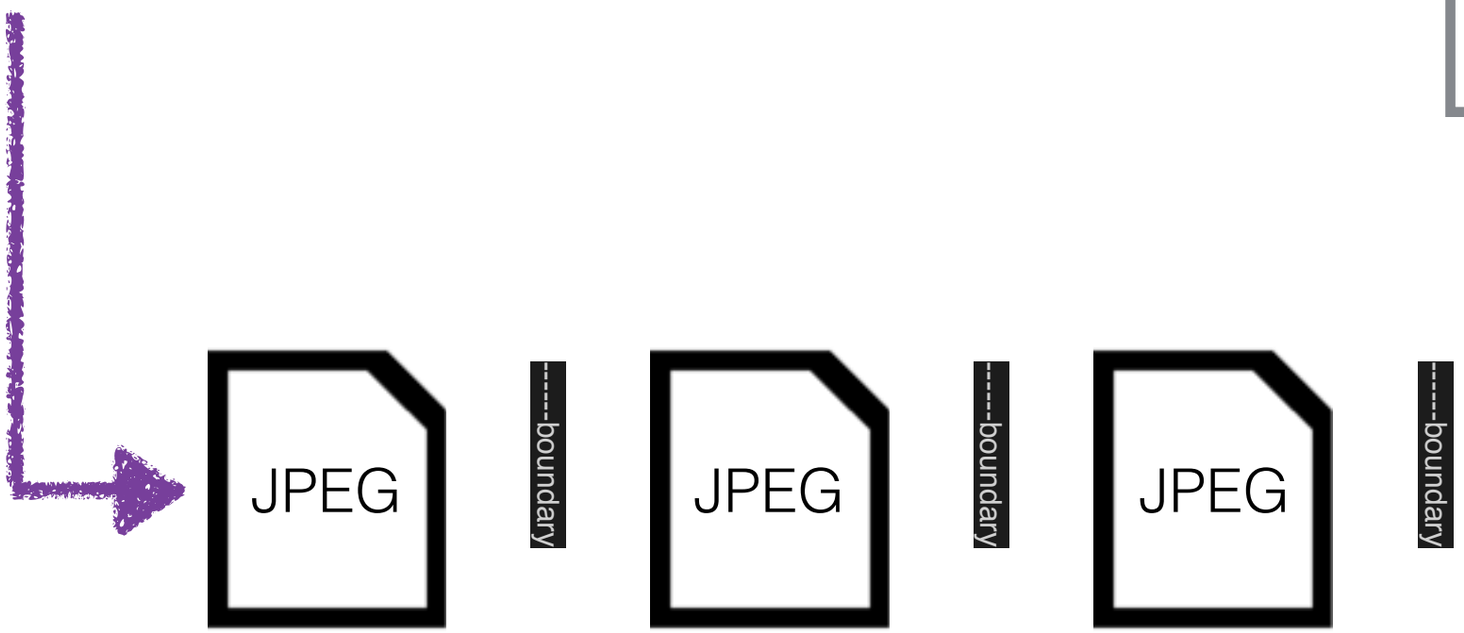
MJPEG over HTTP



```
<html>

  <head>
    <title>Video!</title>
  </head>

  <body>
    <img src="video.mjpg" />
  </body>
</html>
```



lib/nerves_cam/streamer.ex

```
defmodule NervesCam.Streamer do
  import Plug.Conn

  @behaviour Plug
  @boundary "w58EW1cEpjzydSCq"

  def init(opts), do: opts

  def call(conn, _opts) do
    conn
    |> put_resp_header("Age", "0")
    |> put_resp_header("Cache-Control", "no-cache, private")
    |> put_resp_header("Pragma", "no-cache")
    |> put_resp_header("Content-Type", "multipart/x-mixed-replace; boundary=#{@boundary}")
    |> send_chunked(200)
    |> send_pictures
  end

  defp send_pictures(conn) do
    send_picture(conn)
    send_pictures(conn)
  end

  defp send_picture(conn) do
    jpg = Picam.next_frame()
    size = byte_size(jpg)
    header = "------#{@boundary}\r\nContent-Type: image/jpeg\r\nContent-length: #{size}\r\n\r\n"
    footer = "\r\n"
    with {:ok, conn} <- chunk(conn, header),
         {:ok, conn} <- chunk(conn, jpg),
         {:ok, conn} <- chunk(conn, footer),
      do: conn
  end
end
```

```
$ git checkout step4
```

Try it out

```
$ export MIX_TARGET=rpi0
$ mix firmware

Connect the Raspberry Pi Zero and ping it
$ mix firmware.push nerves.local

Wait for the Raspberry Pi to reboot
Open a browser and try loading http://nerves.local/
```

Things to try

```
Connect over the serial port using picocom or screen

iex> Picam.set_img_effect(:sketch)
:ok
iex> h Picam.set_img_effect
...
iex> Picam.set_col_effect({128,128})
:ok
```

```
Check out the docs at
https://hex.pm/packages/picam
```

ELIXIRCONF™ 2017

Nerves Training
Part 6: Streaming video