GPS Tracker

Nerves

Yann Very

@yannvery

Topics

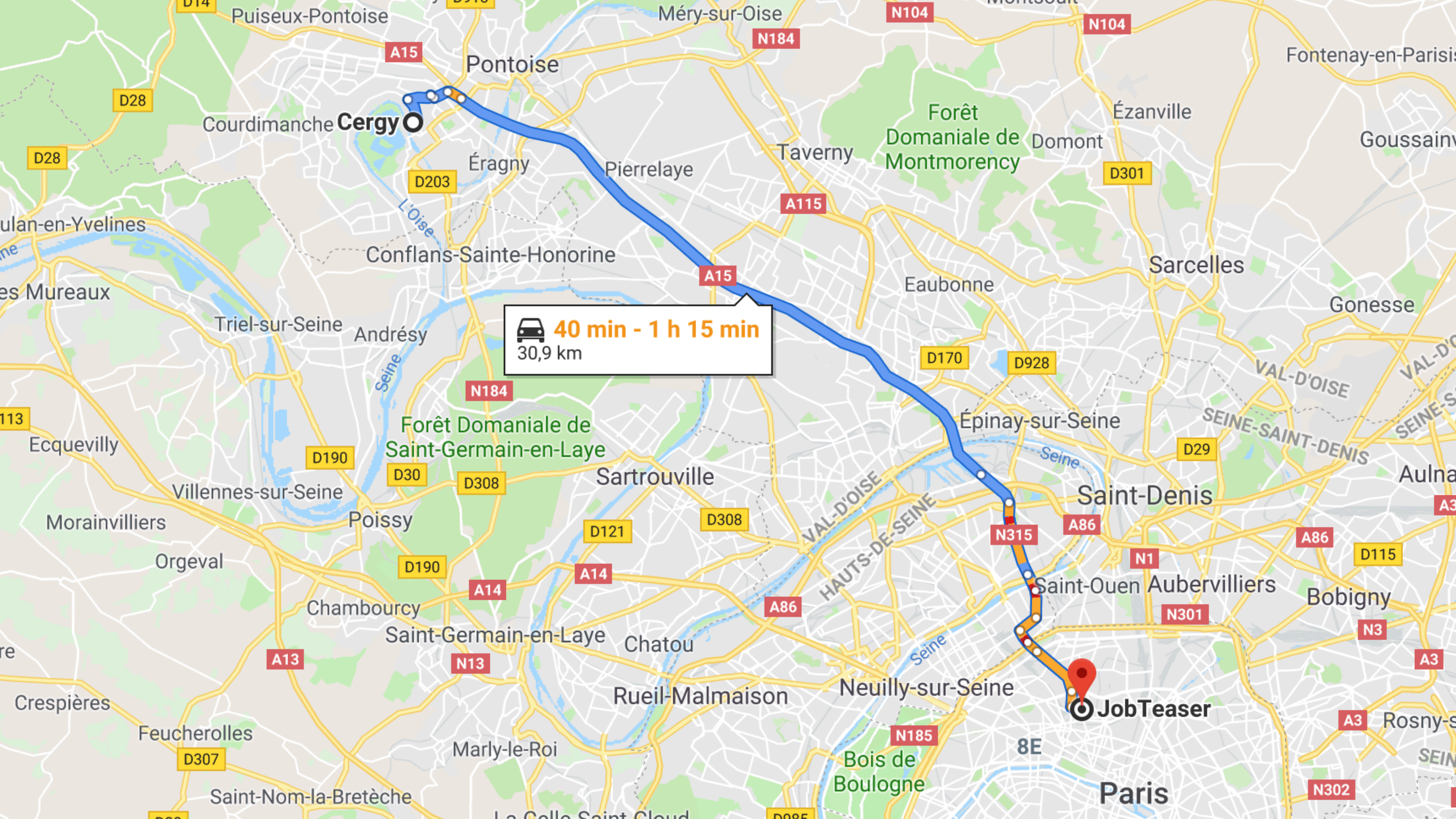
Context

Nerves

GPS / Hardware

Implementation

Why?


Cergy
JobTeaser
40 min - 1 h 15 min
30,9 km
Puiseux-Pontoise
Méry-sur-Oise
Pontoise
Courdimanche
Éragny
Pierrelaye
Taverny
Forêt Domaniale de Montmorency
Domont
Ézanville
Fontenay-en-Parisis
Goussainv
L'Oise
Conflans-Sainte-Honorine
Eaubonne
Sarcelles
Gonesse
Triel-sur-Seine
Andrésy
Seine
Forêt Domaniale de Saint-Germain-en-Laye
Épinay-sur-Seine
VAL-D'OISE
SEINE-SAINT-DENIS
Ecquevilly
Villennes-sur-Seine
Sartrouville
Saint-Denis
Morainvilliers
Poissy
HAUTS-DE-SEINE
Orgeval
Saint-Ouen
Aubervilliers
Bobigny
Chambourcy
Saint-Germain-en-Laye
Chatou
Rueil-Malmaison
Neuilly-sur-Seine
Crespières
Feucherolles
Marly-le-Roi
Bois de Boulogne
8E
Paris
Saint-Nom-la-Bretèche
Rosny-s
A15
N184
N104
D28
D203
A115
D301
D170
D928
D29
N315
A86
N1
D115
N301
N3
A3
D190
D30
D308
D121
A14
A13
N13
N185
N302
D307

OHMAGIF.COM

OHMAGIF.COM

gobee.bike

Core Feature

- Make a GPS tracker
- Emit my position
- Provide an html page with my current position

Where to begin?

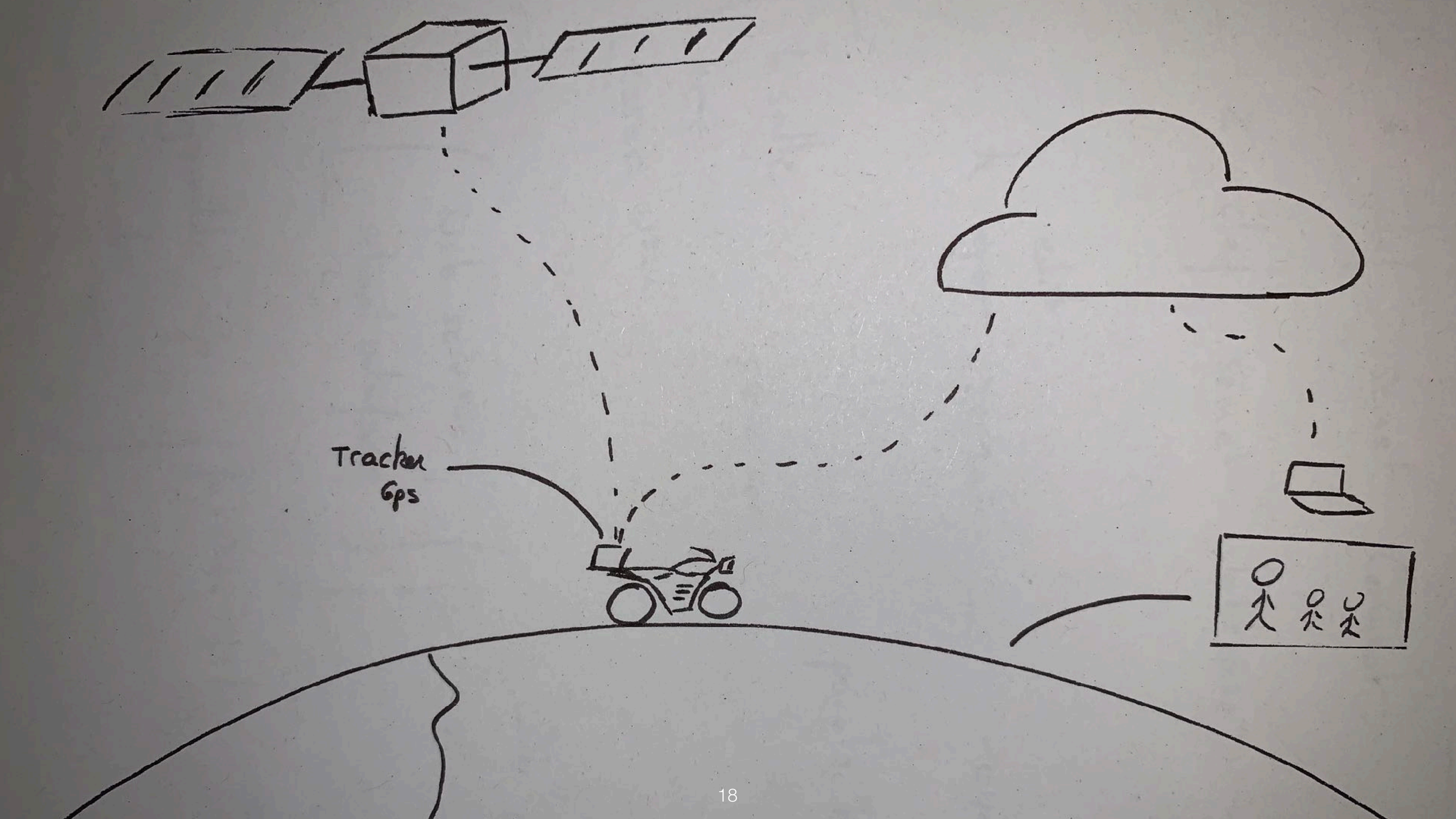
9.1 AH(20HR)/190 CCA
ELECTROLYTE FILLED
MADE IN JAPAN

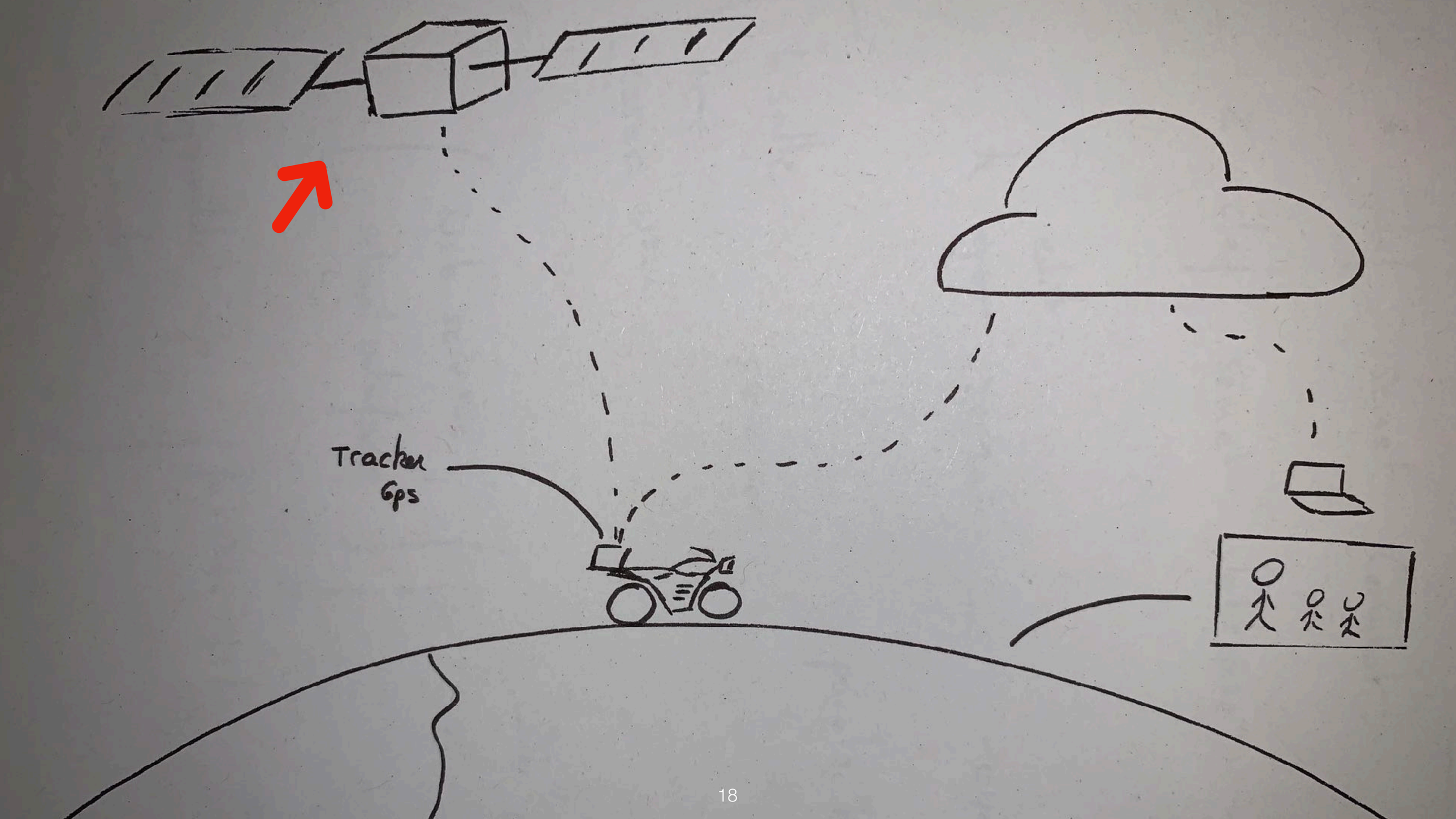
9.1 AH(20HR)/190 CCA

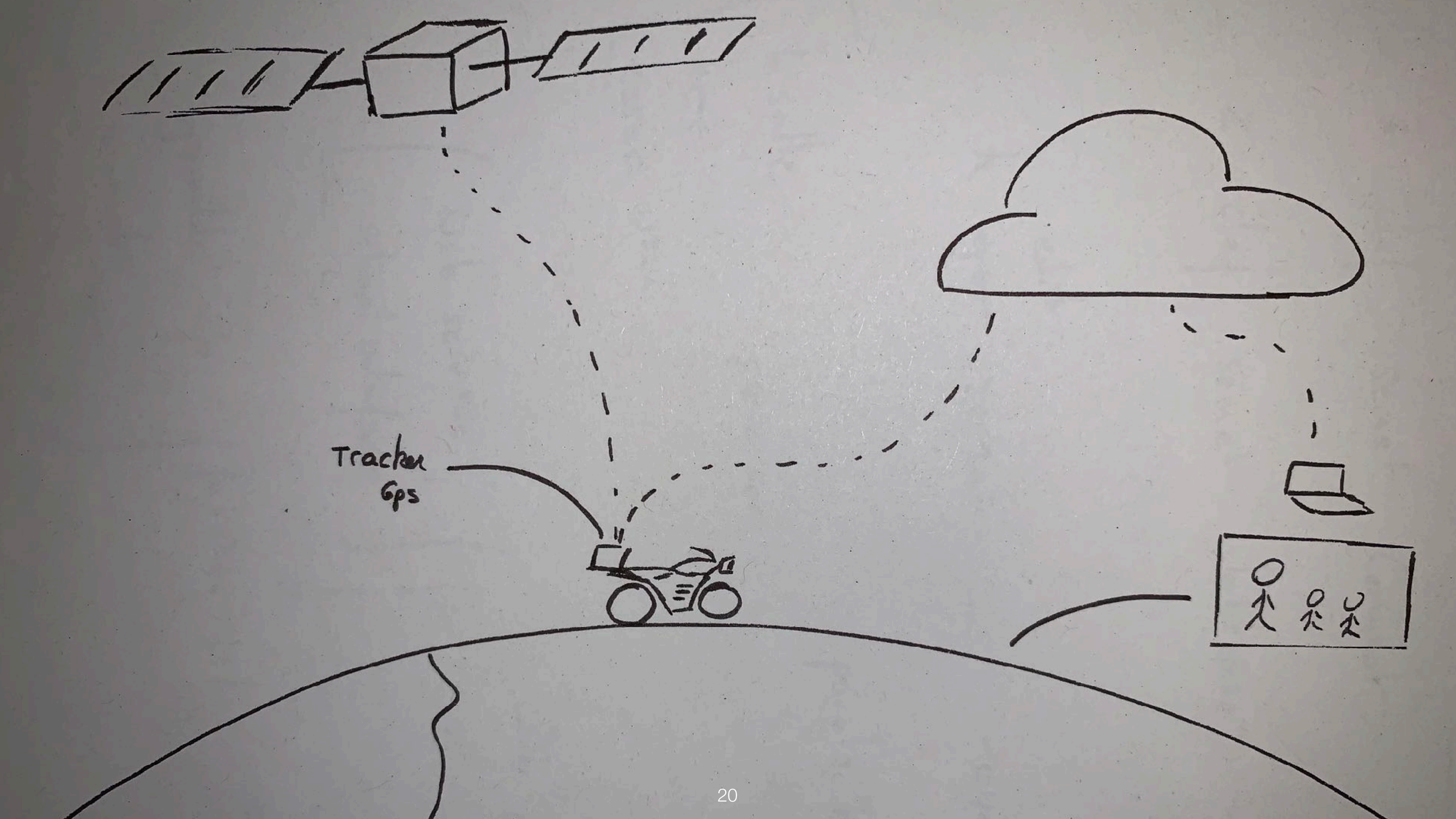
BS21-010
A
Poire
9.1 Ah(20HR)/190 CCA
ELECTROLYTE FILLED
MADE IN JAPAN

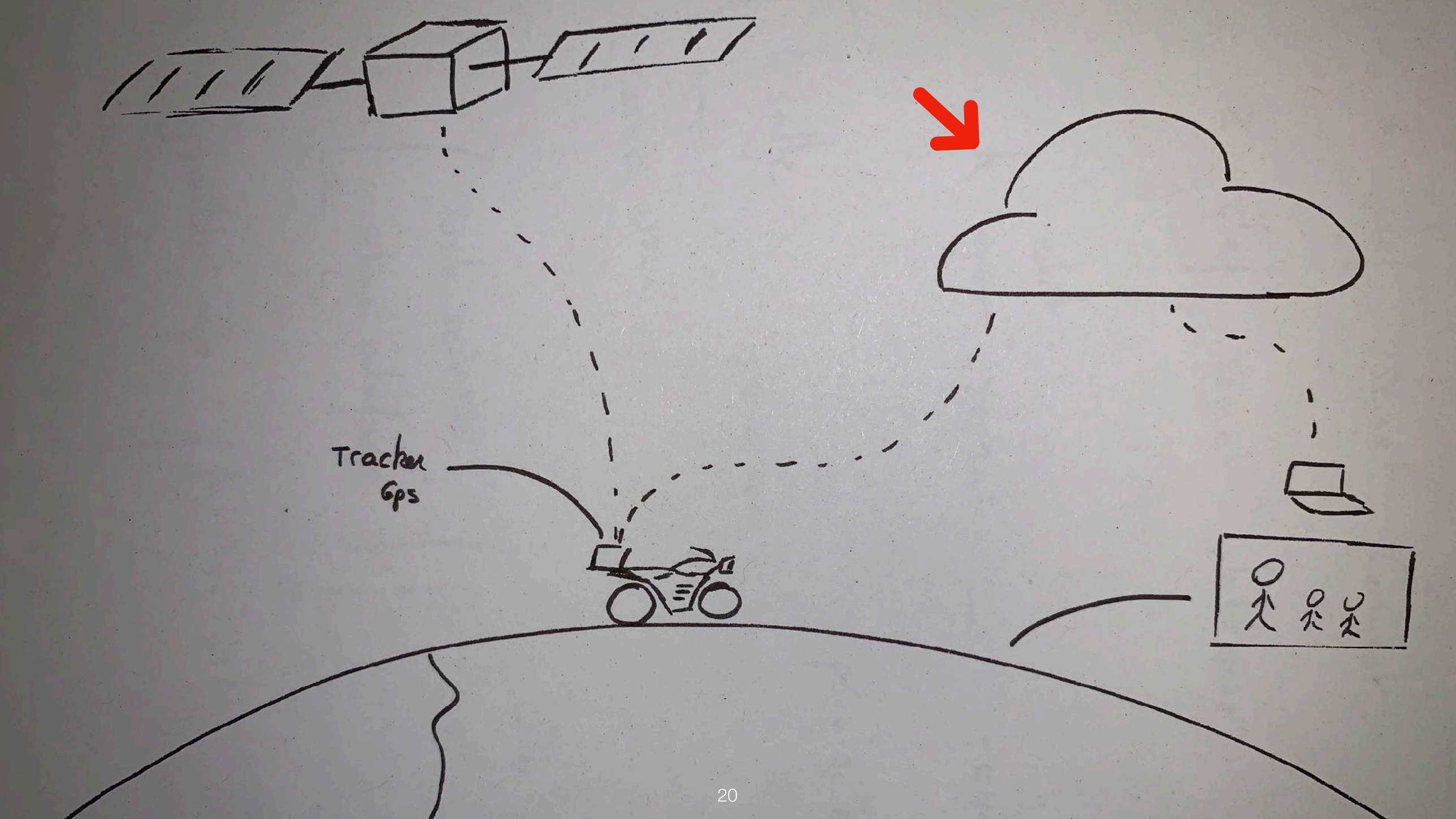
Identifying the unknown

By making a plan

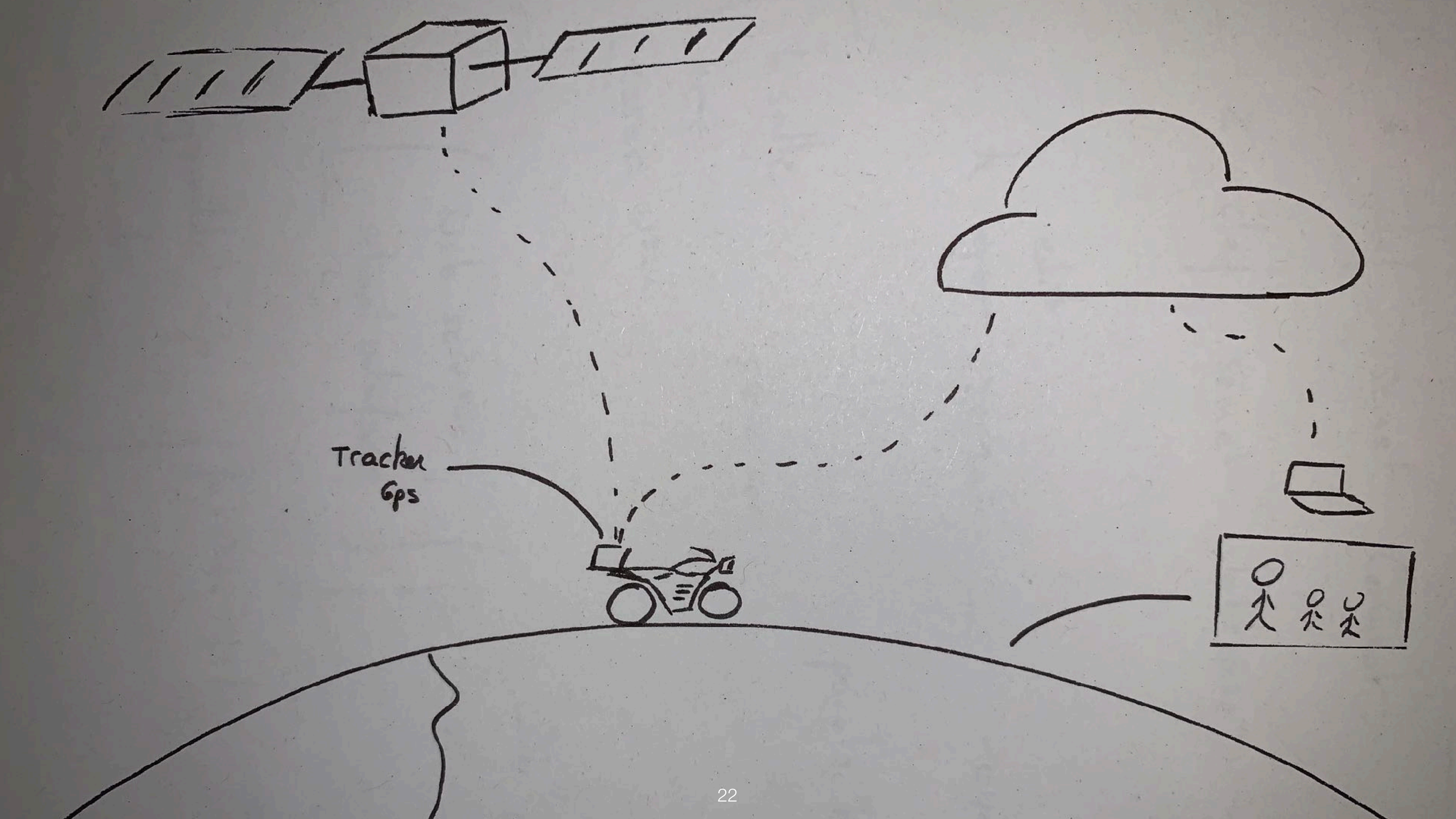

Tracker
Gps

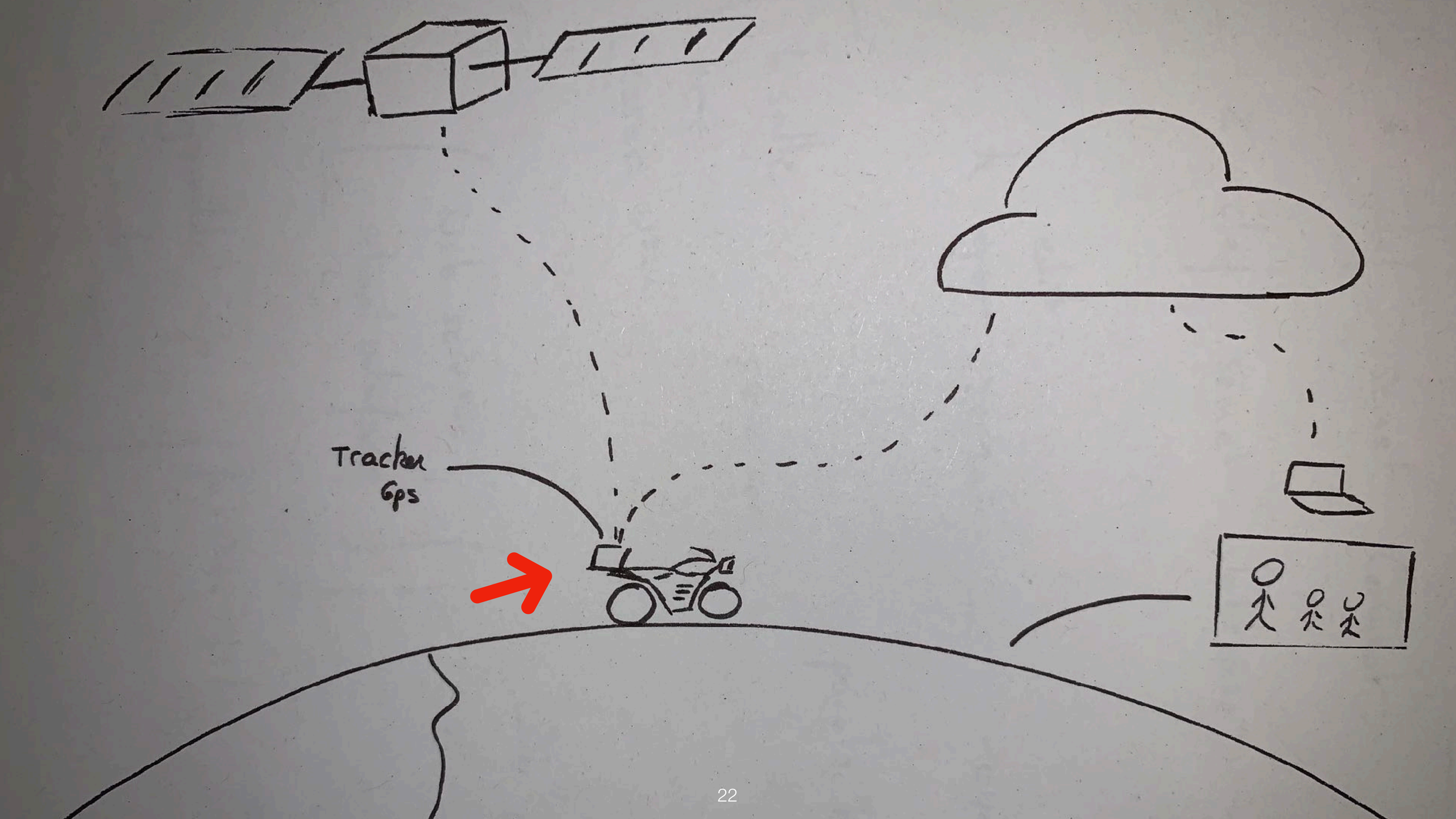

Tracker
Gps

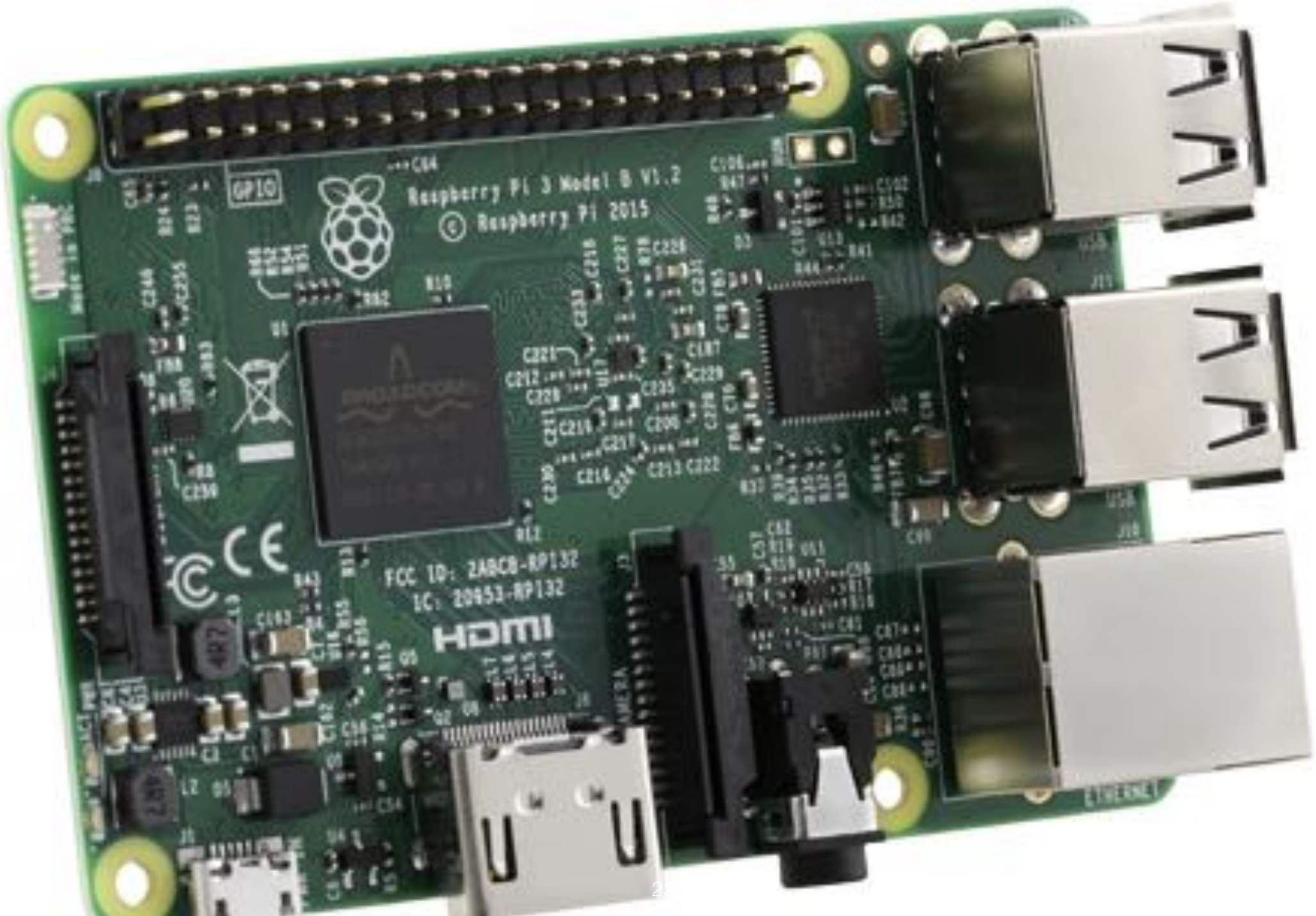

Tracker
Gps

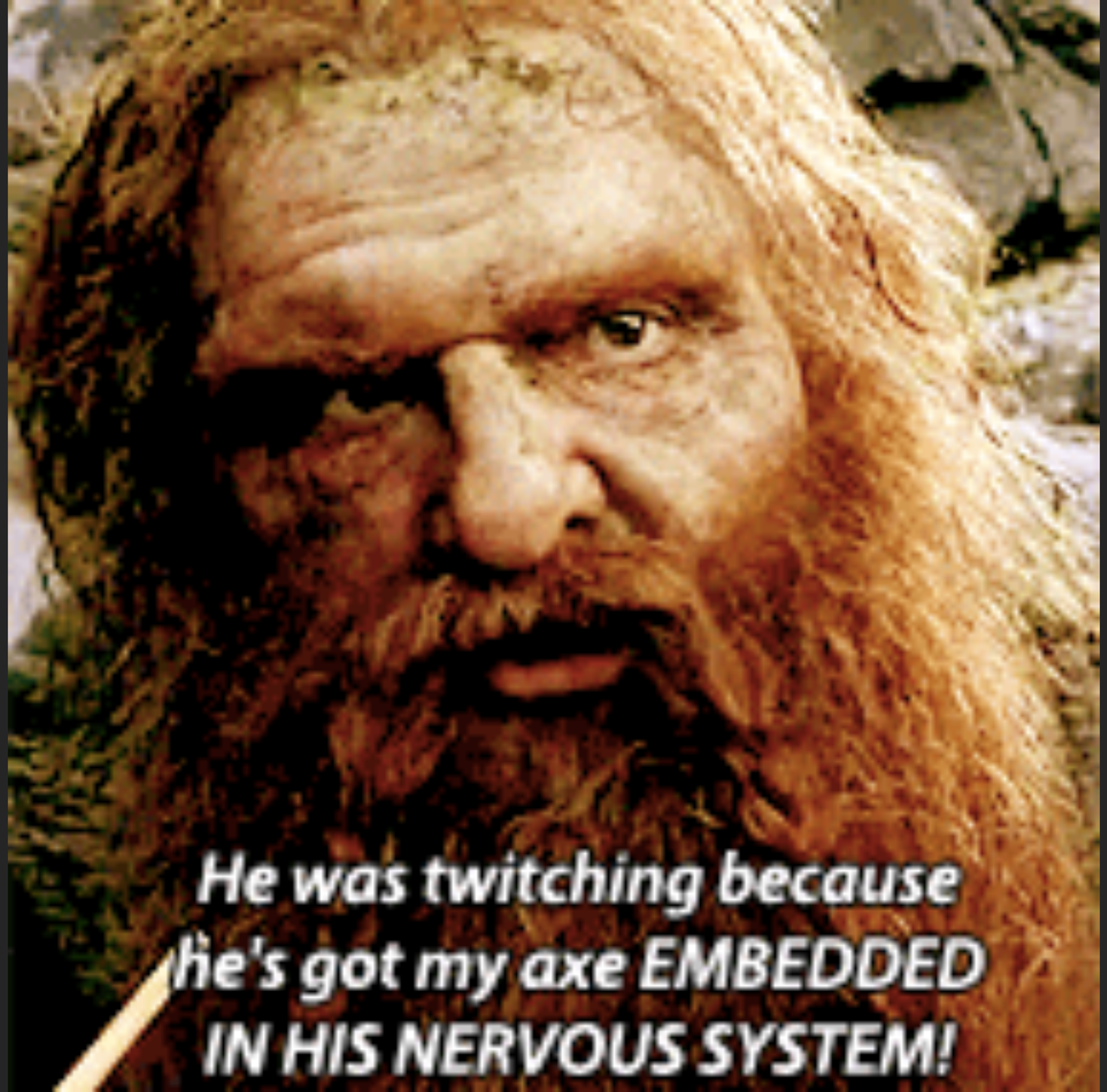

Tracker
Gps

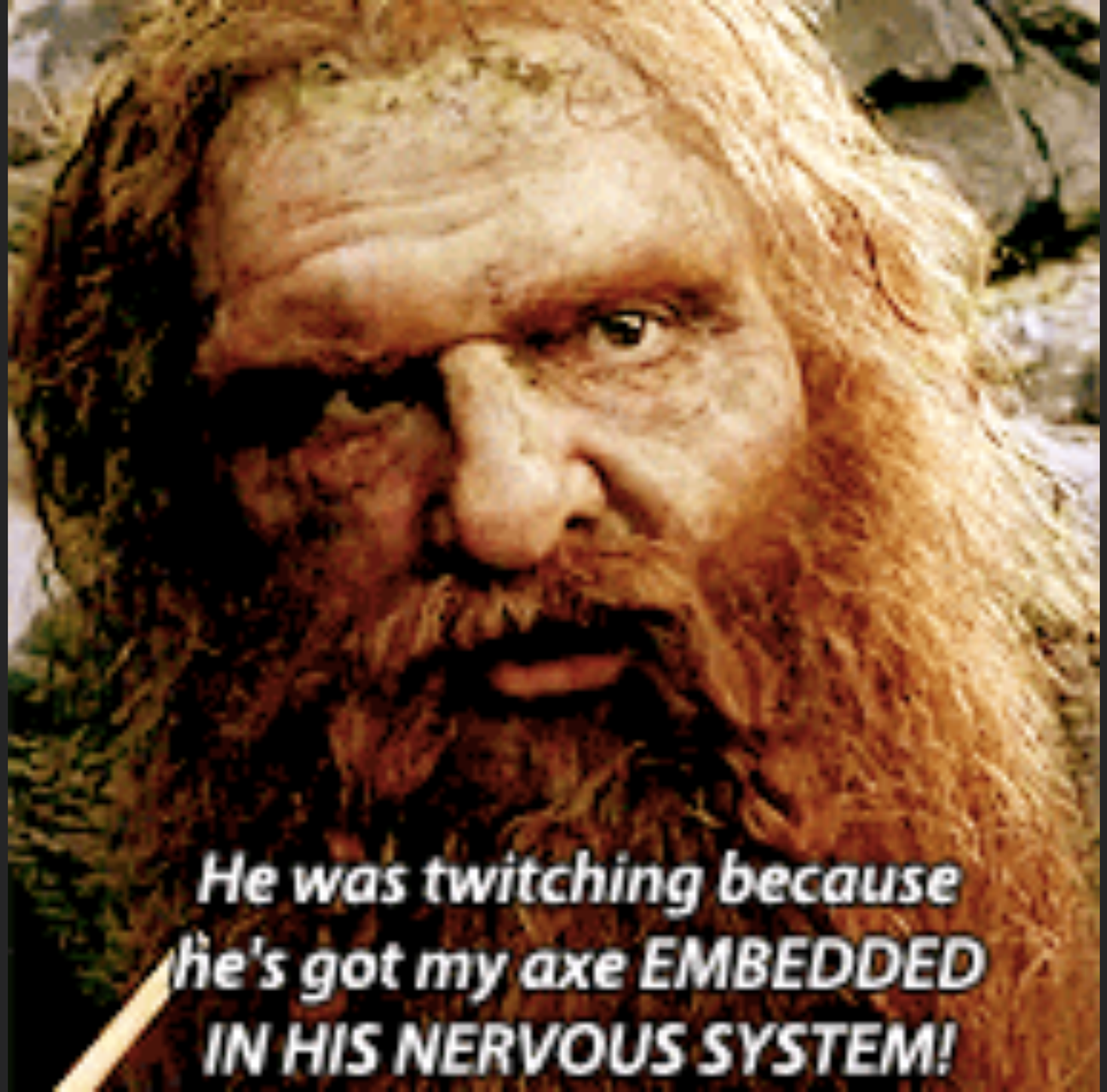
A web app


Tracker
Gps


Tracker
Gps


Raspberry Pi 3 Model B V1.2
© Raspberry Pi 2015
GPIO
FCC ID: 2ABCB-RPI32
IC: 20953-RPI32
HDMI


He was twitching because
he's got my axe EMBEDDED
IN HIS NERVOUS SYSTEM!


He was twitching because
he's got my axe EMBEDDED
IN HIS NERVOUS SYSTEM!

Built with Nerves

nerves-project / nerves

Watch 52 | Star 1,200 | Fork 96

Code | Issues 20 | Pull requests 1 | Projects 0 | Wiki | Security | Insights

Craft and deploy bulletproof embedded software in Elixir http://nerves-project.org

elixir nerves embedded

802 commits | 3 branches | 54 releases | 59 contributors | Apache-2.0

Contributors

This project exists thanks to all the people who contribute.

Nerves

Platform

Framework

Tooling

Platform

Supported Targets and Systems

TARGET	SYSTEM	TAG
Raspberry Pi A+, B, B+	nerves_system_rpi	`rpi`
Raspberry Pi Zero and Zero W	nerves_system_rpi0	`rpi0`
Raspberry Pi 2	nerves_system_rpi2	`rpi2`
Raspberry Pi 3 B, B+	nerves_system_rpi3	`rpi3`
BeagleBone Black, BeagleBone Green, BeagleBone Green Wireless, and PocketBeagle.	nerves_system_bbb	`bbb`
Generic x86_64	nerves_system_x86_64	`x86_64`

Framework

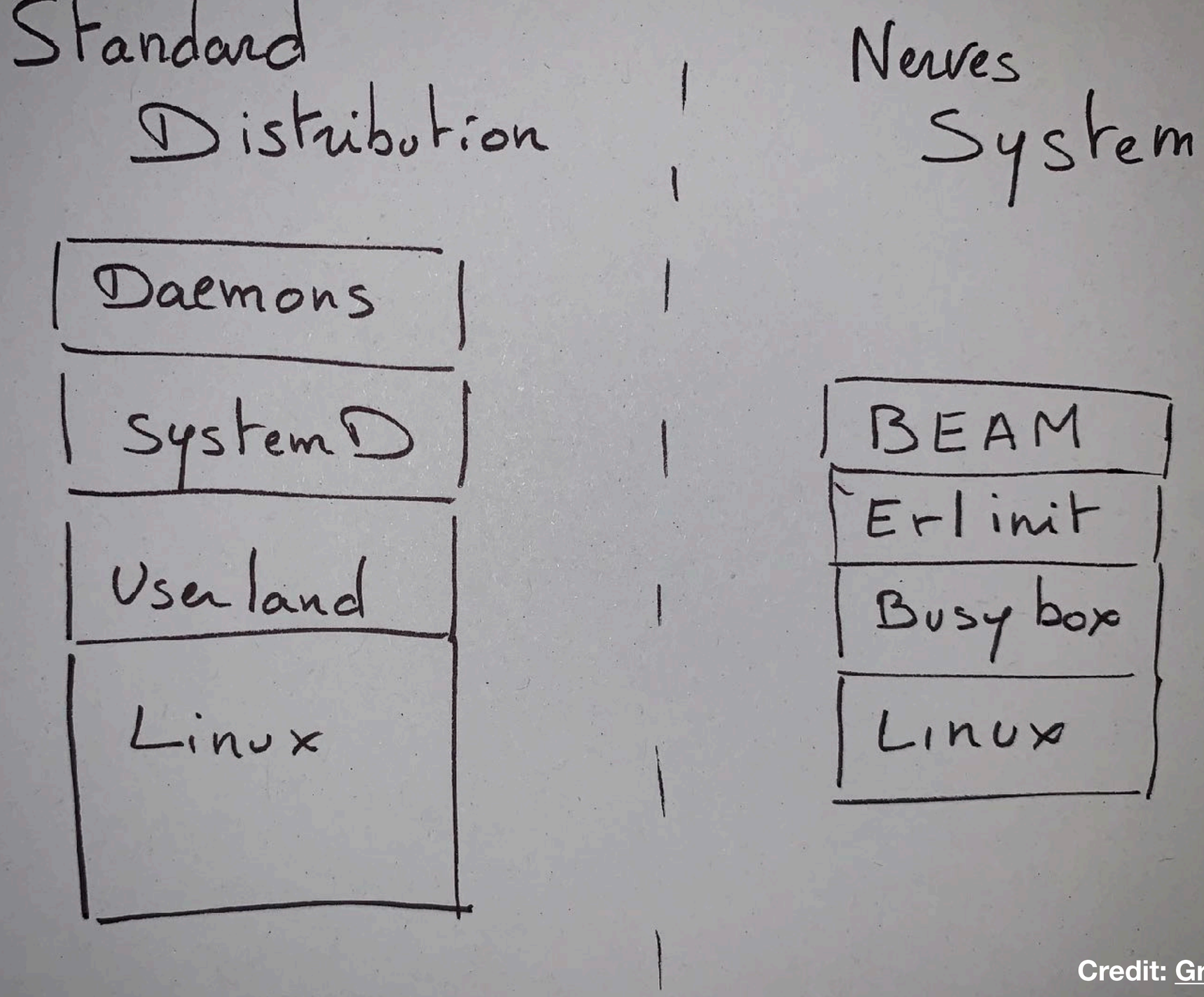
Ready to Go Elixir libraries:

- Network
- Interacting with the hardware

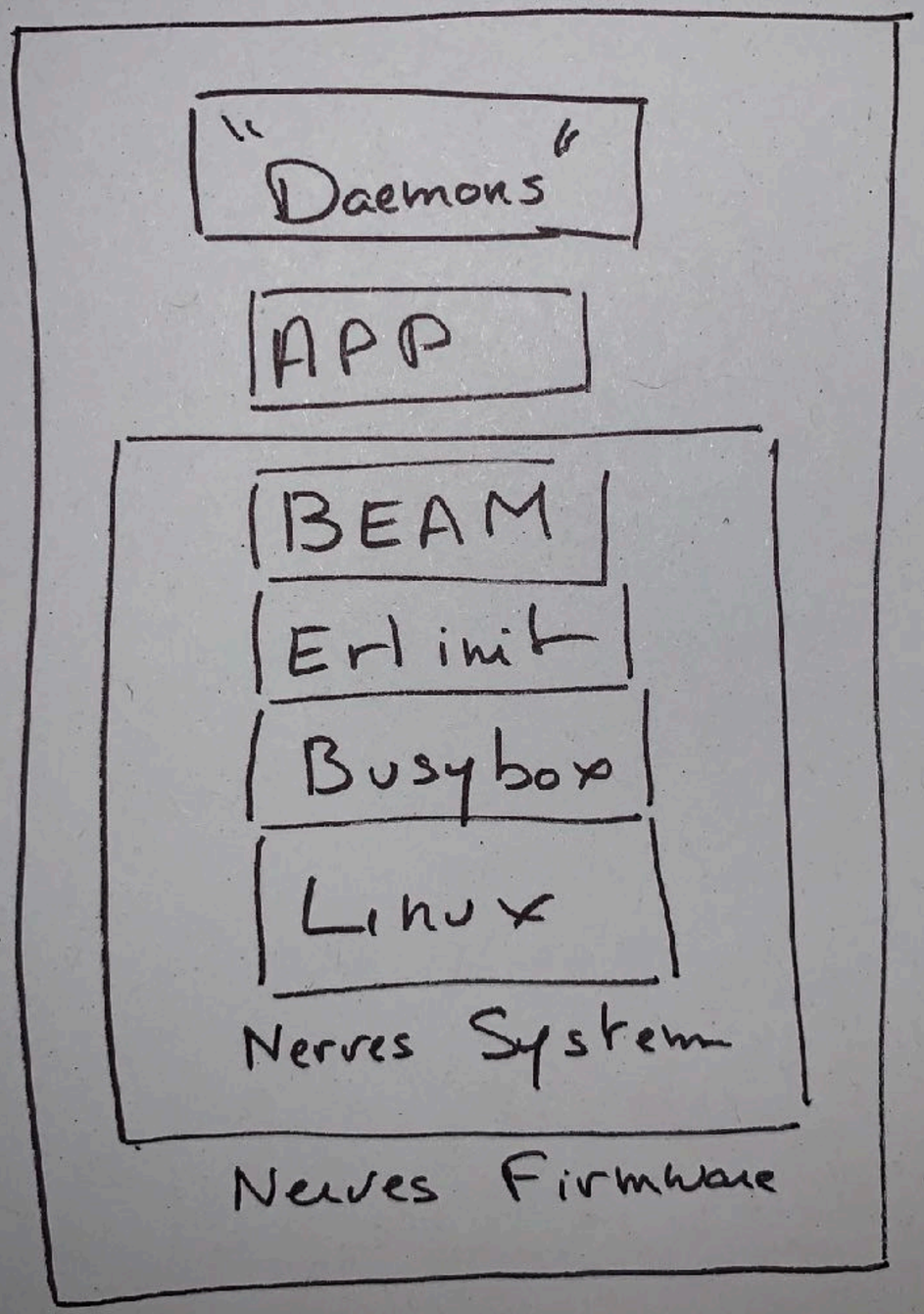
Tooling

CLI which allows to:

- Create a project
- Build firmware
- Configure a release



Credit: Greg Mefford



Credit: Greg Mefford

Create a Nerves Project

```
# Create a Nerves project
mix nerves.new gps_tracker
```

Create a Nerves Project

```
# Create a Nerves project
mix nerves.new gps_tracker

# Update project dependencies
cd gps_tracker
MIX_TARGET=rpi3 mix deps.get
```

Create a Nerves Project

```
# Create a Nerves project
mix nerves.new gps_tracker

# Update project dependencies
cd gps_tracker
MIX_TARGET=rpi3 mix deps.get

# Create a firmware
mix firmware
```

Create a Nerves Project

```
# Create a Nerves project
mix nerves.new gps_tracker

# Update project dependencies
cd gps_tracker
MIX_TARGET=rpi3 mix deps.get

# Create a firmware
mix firmware

# Create a bootable SD card
mix firmware.burn
```

nerves-project / **nerves_examples** Nerves Examples

Watch 17 | Star 211 | Fork 79

Code | Issues 2 | Pull requests 2 | Projects 0 | Wiki | Insights

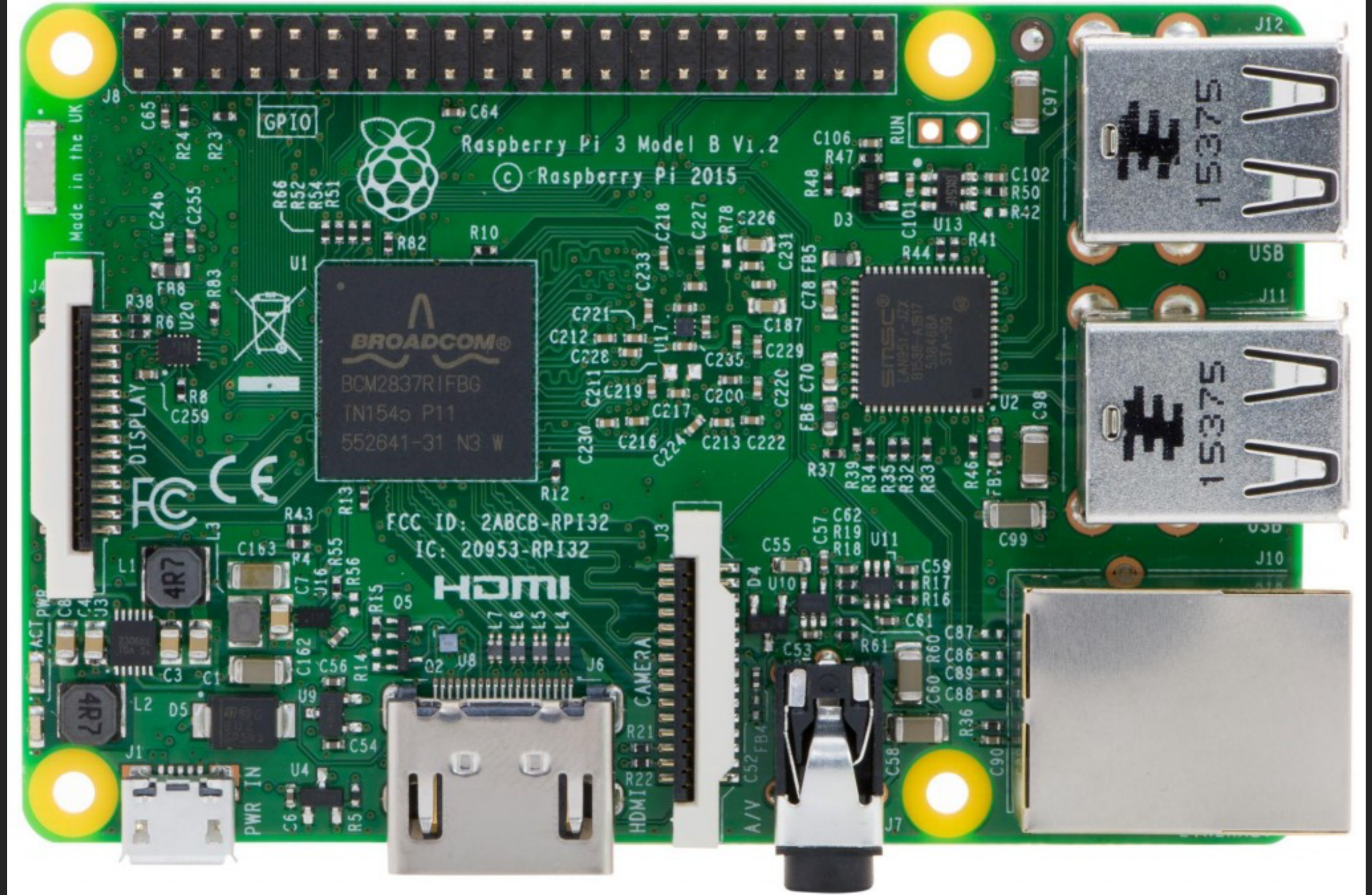
A couple small examples to demonstrate using Nerves

262 commits | 5 branches | 0 releases | 36 contributors

Branch: master | New pull request | Create new file | Upload files | Find File | Clone or download

GregMefford and fhunleth Consistently assume rpi0 target by default — Latest commit d9a21ca on Feb 21

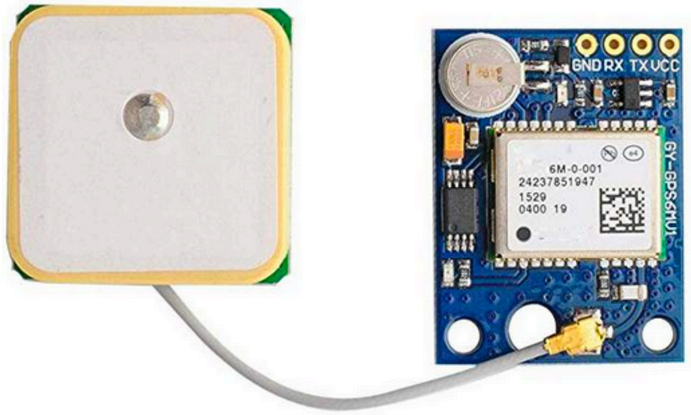
.circleci	Upgrade to Elixir 1.8 in CI	3 months ago
blinky	Consistently assume rpi0 target by default	3 months ago
hello_gpio	Consistently assume rpi0 target by default	3 months ago
hello_leds	Consistently assume rpi0 target by default	3 months ago
hello_network	Consistently assume rpi0 target by default	3 months ago
hello_phoenix	Update for Elixir 1.8 and Nerves 1.4	3 months ago
.gitignore	Synchronize various files with latest template	7 months ago
CODE_OF_CONDUCT.md	Add links to contribution guides	9 months ago
CONTRIBUTING.md	Add links to contribution guides	9 months ago
ISSUE_TEMPLATE.md	Add links to contribution guides	9 months ago
README.md	Update for Elixir 1.8 and Nerves 1.4	3 months ago


Raspberry Pi 3 Model B V1.2
© Raspberry Pi 2015
GPIO
Made in the UK
BROADCOM
BCM2837RIFBG
TN1545 P11
552641-31 N3 W
FCC ID: 2ABCB-RPI32
IC: 20953-RPI32
HDMI
DISPLAY
CAMERA
PWR IN
A/V
USB
RUN
ACT PWR

GPS

GTPA010
GPS Breakout
adafruit
ARDUINO
UNO
GPS ANTENNA
GPS-1-ABI-1-V1
2.7 cm
GPS MODULE
PN 30-2207
5Hz GPS/GLONASS
CAN 2.0B Output
AEMnet
Green: Fix Acquired
Red: No Signal
uBlox GPS Module
NEO-6 GPS

Neo 6m GPS



DIYmall 6M GPS Module with EEPROM for MWC/AeroQuad with Antenna for Arduino Flight Control Aircraft

★★★★☆ 25

$16.50

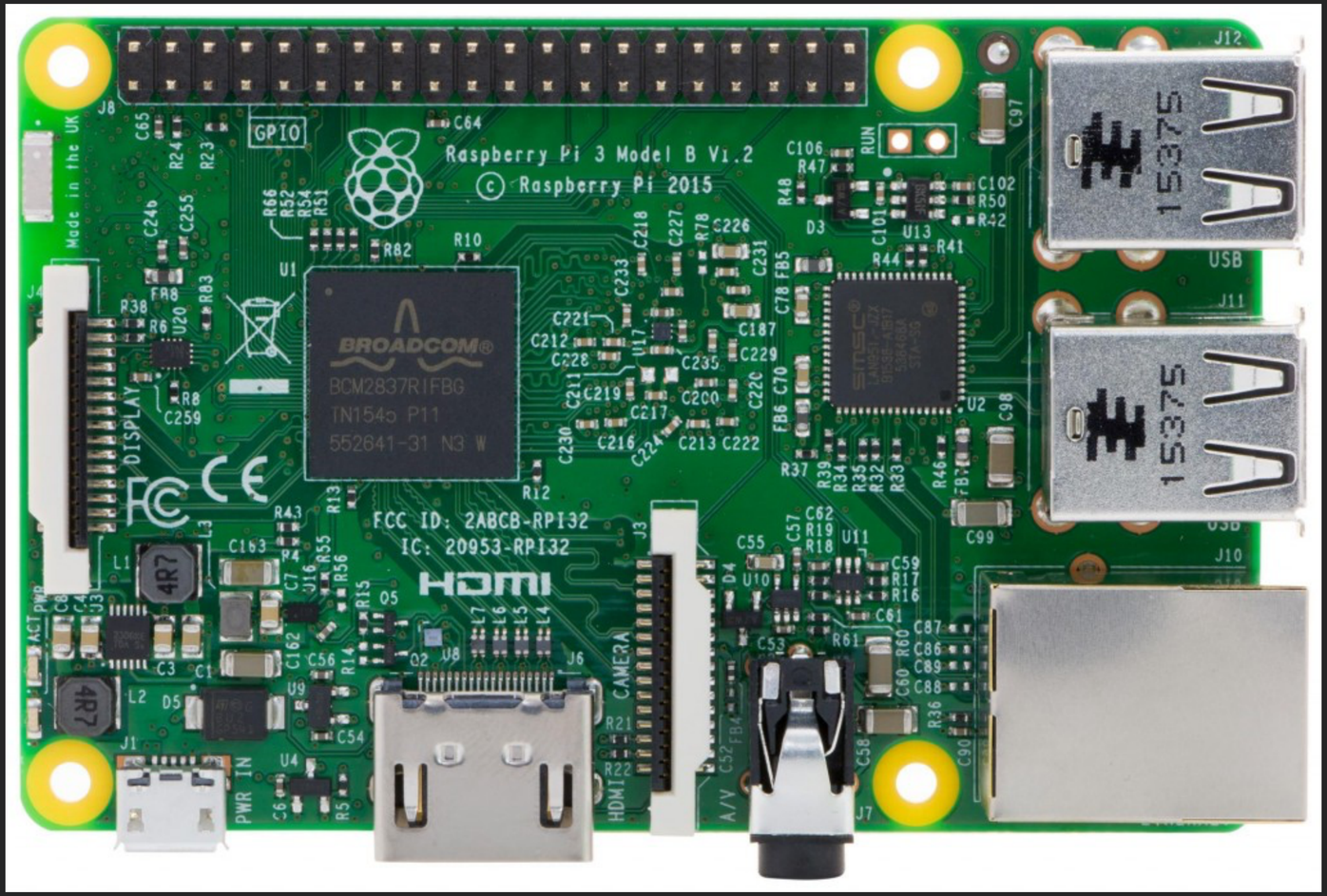
Ca marche pas !!
RTFM !!

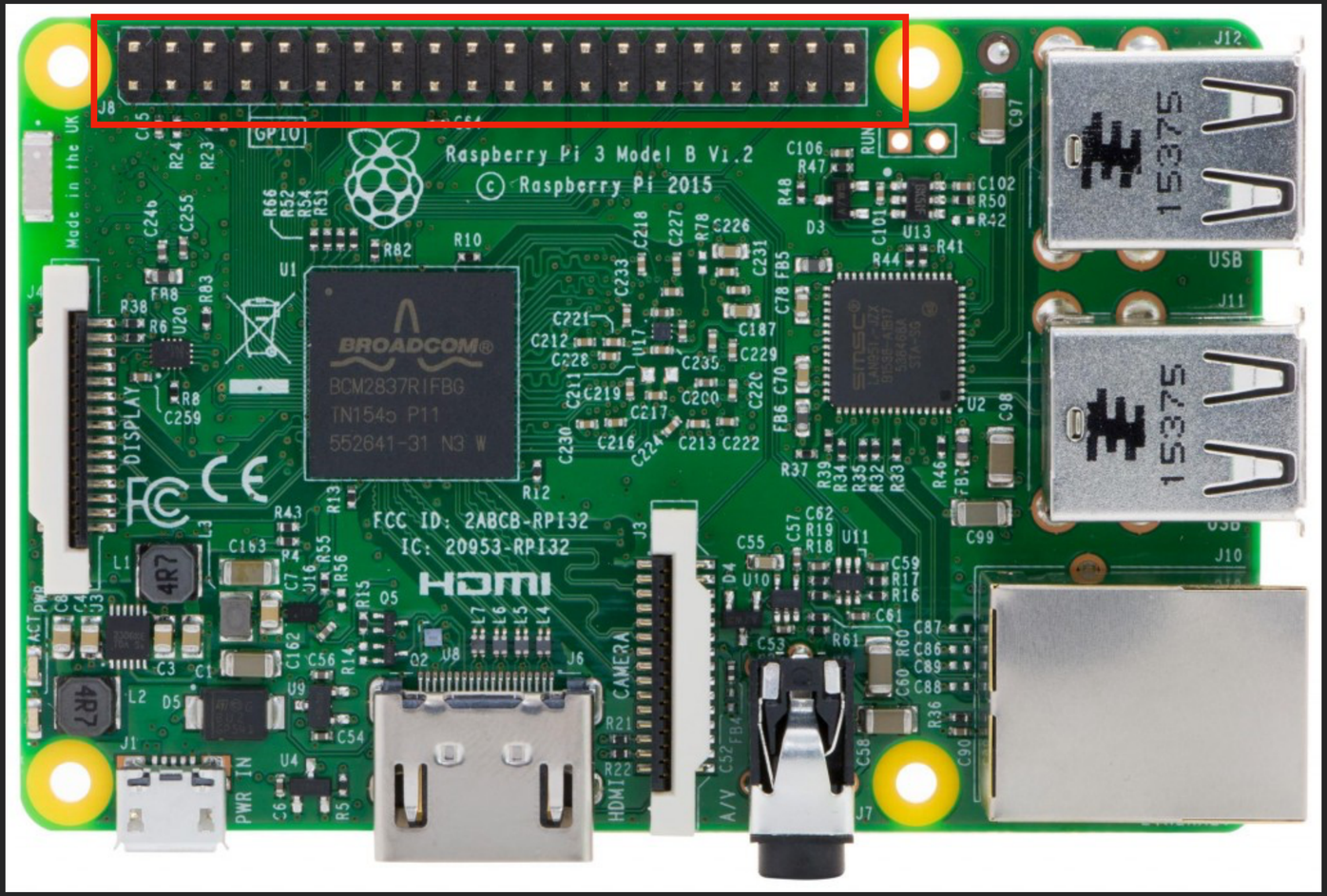
https://www.terraelectronica.ru/pdf/show?pdf_file=%2Fz%2FDatasheet%2FU%2FUART+GPS+NEO-6M+User+Manual.pdf

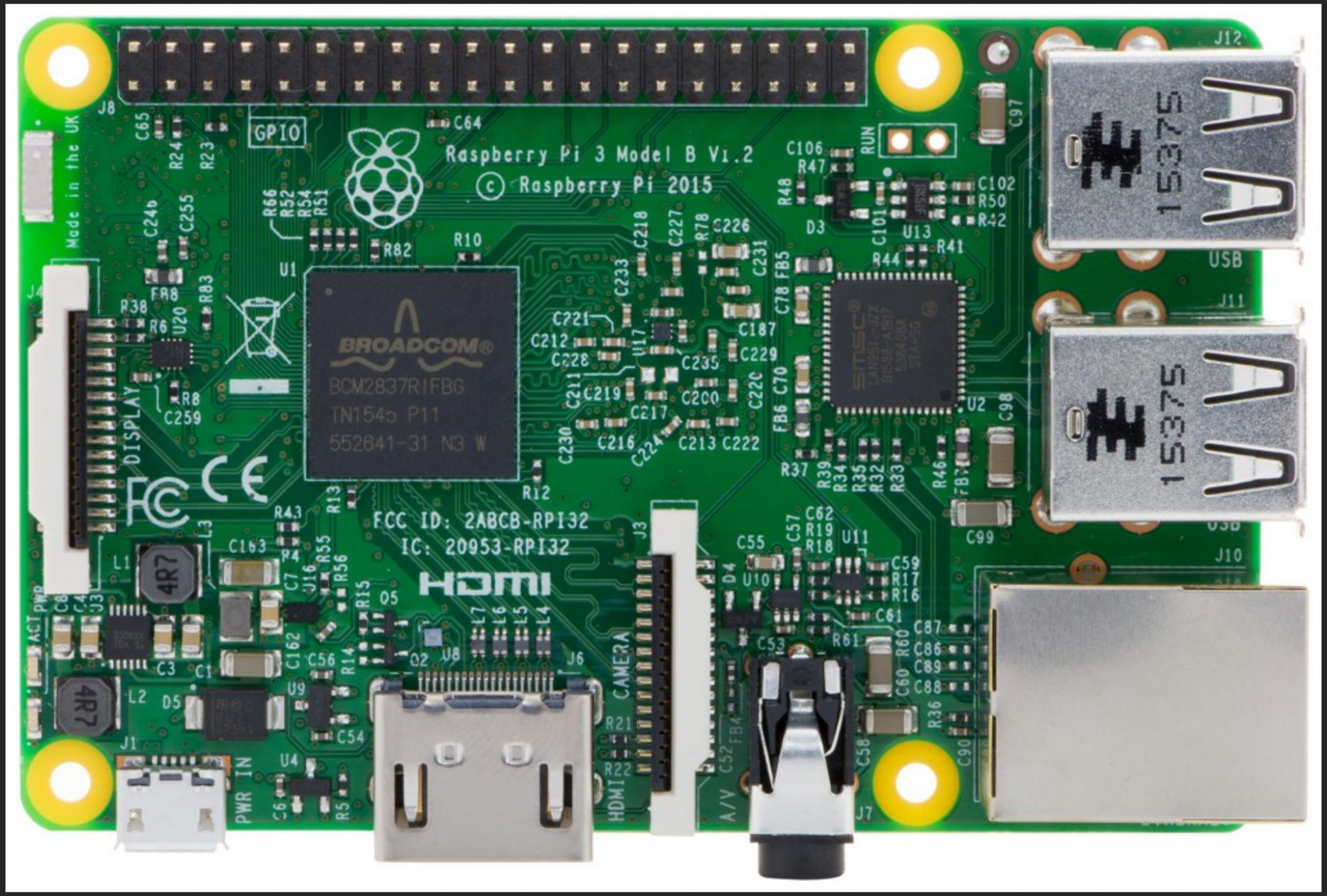
How to use

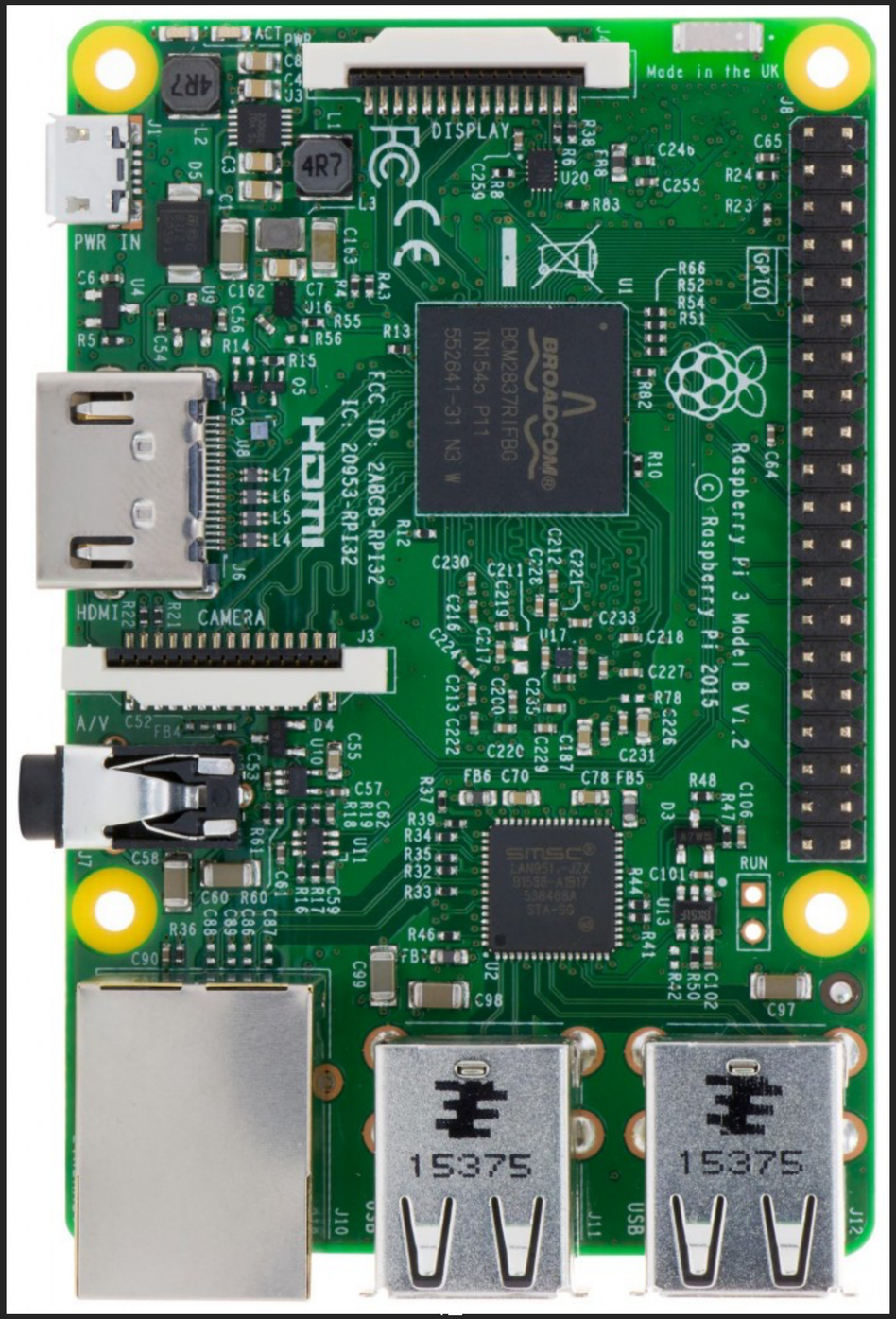
1. Connect the UART GPS NEO-6M module to a serial module. FT232 is applied as the serial module in this document.
   The connection between the GPS module and the serial module is listed as the table below.

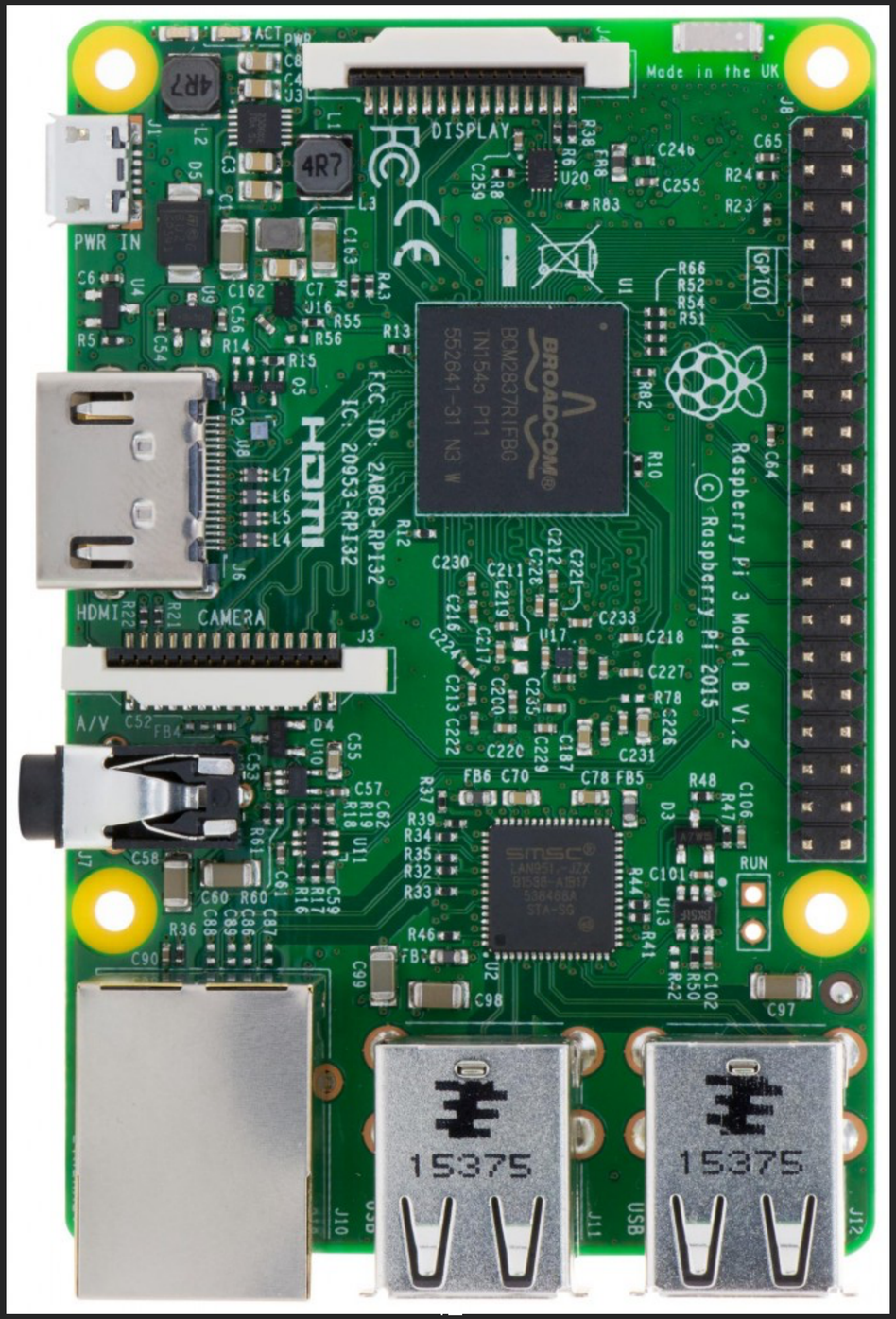
UART GPS NEO-6M module pins	Serial module pins
VCC	3.3V/5V
GND	GND
TXD	RX
RXD	TX
PSS*	
* PPS should be connected to the clock pulse output (CPOUT) of a MCU. However, this pin is unconnected, in the case that the GPS module is connected to a computer.	


GPIO
Raspberry Pi 3 Model B V1.2
© Raspberry Pi 2015
BROADCOM
BCM2837RIFBG
FCC ID: 2ABCB-RPI32
IC: 20953-RPI32
HDMI
DISPLAY
CAMERA
USB
PWR IN
Made in the UK


Raspberry Pi 3 Model B V1.2
© Raspberry Pi 2015
BROADCOM
BCM2837RIFBG
FCC ID: 2ABCB-RPI32
IC: 20953-RPI32
HDMI
GPIO
DISPLAY
CAMERA
PWR IN
USB
Made in the UK

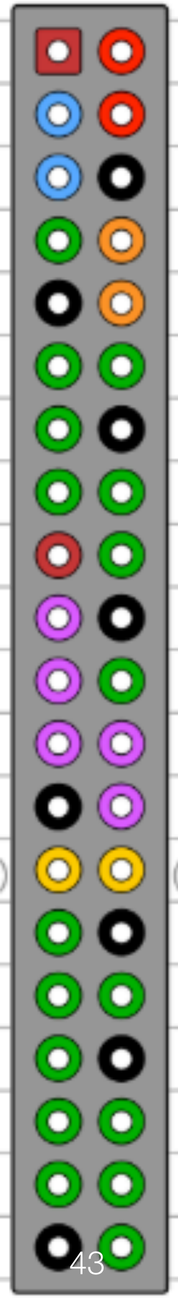

GPIO
Raspberry Pi 3 Model B V1.2
© Raspberry Pi 2015
BROADCOM
BCM2837RIFBG
FCC ID: 2ABCB-RPI32
IC: 20953-RPI32
HDMI
DISPLAY
CAMERA
USB
PWR IN
Made in the UK


Made in the UK
DISPLAY
PWR IN
GPIO
HDMI
CAMERA
A/V
RUN
USB
Raspberry Pi 3 Model B V1.2
© Raspberry Pi 2015
FCC ID: 2ABCB-RPI32
IC: 20953-RPI32
BROADCOM
BCM2837IFBG
15375
15375


Made in the UK
DISPLAY
PWR IN
GPIO
HDMI
CAMERA
A/V
RUN
USB
Raspberry Pi 3 Model B V1.2
© Raspberry Pi 2015
FCC ID: 2ABCB-RPI32
IC: 20953-RPI32
BROADCOM
BCM2837IFBG
15375
15375

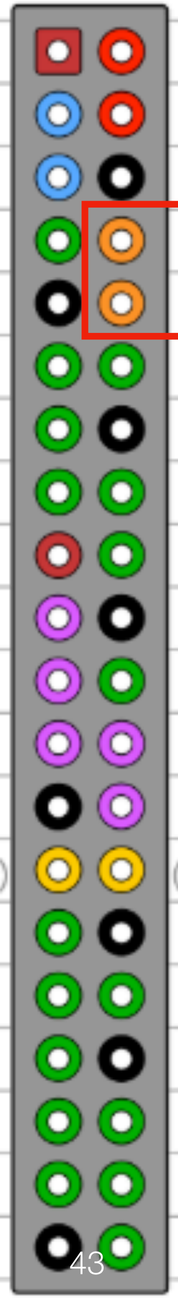
Raspberry Pi 3 GPIO Header

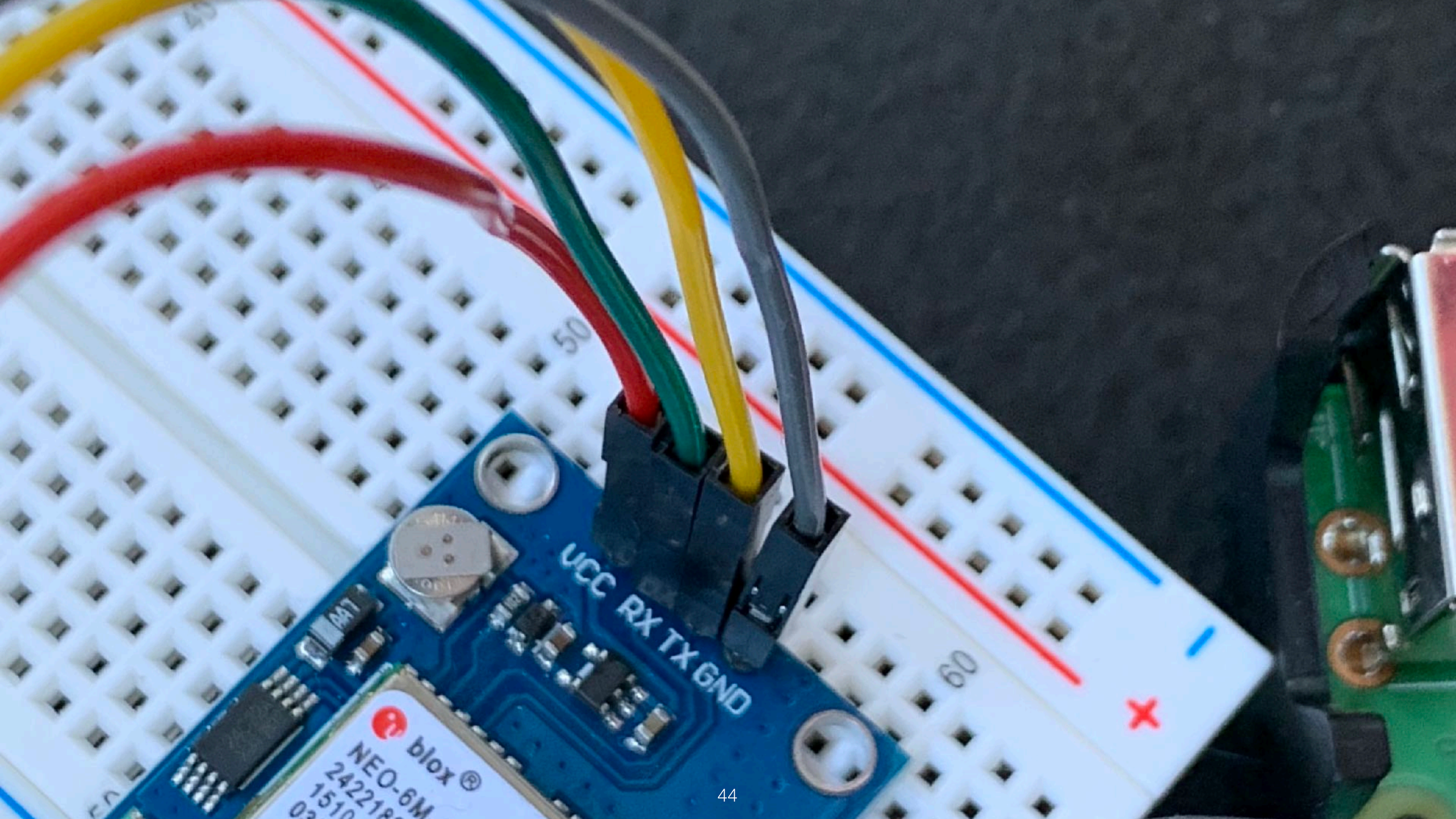
Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I²C)	DC Power 5v	04
05	GPIO03 (SCL1 , I²C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24
25	Ground	(SPI_CE1_N) GPIO07	26
27	ID_SD (I²C ID EEPROM)	(I²C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40



Raspberry Pi 3 GPIO Header

Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I²C)	DC Power 5v	04
05	GPIO03 (SCL1 , I²C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24
25	Ground	(SPI_CE1_N) GPIO07	26
27	ID_SD (I²C ID EEPROM)	(I²C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40




VCC RX TX GND
blox
NEO-6M
50
60

UART MODULE

```
# mix.exs file - Update dependency

defp deps do
  [
    # …
    # UART
    {:circuits_uart, "~> 1.3"}

  ]
end
```

4GIFs
.com

4GIFs
.com

Process Elixir

Process Elixir

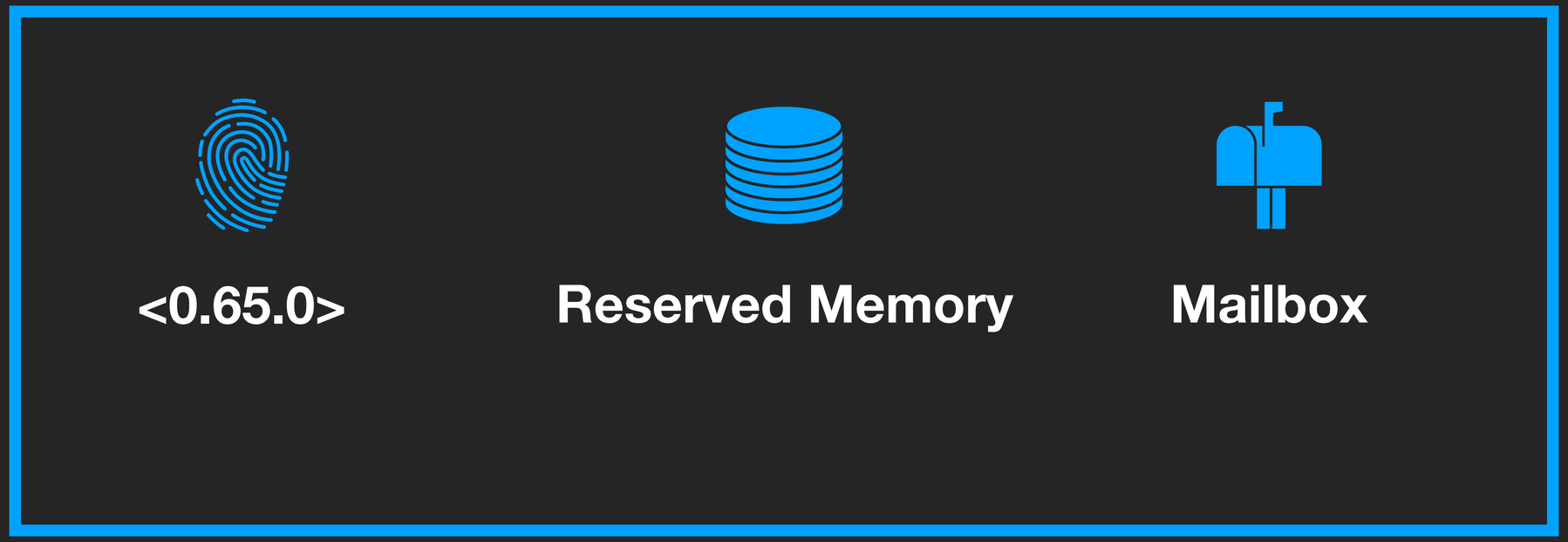
<0.65.0>

Process Elixir

<0.65.0>

Reserved Memory

Process Elixir



UART MODULE

UART MODULE

```
# Start UART GenSserver (process)
{:ok, uart} = Circuits.UART.start_link()
```

UART MODULE

```
# Start UART GenSserver (process)
{:ok, uart} = Circuits.UART.start_link()

# Ask to the UART process to open the serial port 0
Circuits.UART.open(uart, "ttyAMA0", speed: 9600, active: true)
```

UART MODULE

```
# Start UART GenSserver (process)
{:ok, uart} = Circuits.UART.start_link()

# Ask to the UART process to open the serial port 0
Circuits.UART.open(uart, "ttyAMA0", speed: 9600, active: true)

# UART sent informations to the current process
flush()
```

UART MODULE

```
# Start UART GenSserver (process)
{:ok, uart} = Circuits.UART.start_link()

# Ask to the UART process to open the serial port 0
Circuits.UART.open(uart, "ttyAMA0", speed: 9600, active: true)

# UART sent informations to the current process
flush()

{:circuits_uart, “ttyAMA0",
"$GPGGA,064036.289,4836.53750,N,00740.93730,E,1,04,3.2,200.2,M,,,,0000*0E"}
```

NMEA Data

```
"$GPGGA,064036.289,4836.53750,N,00740.93730,E,1,04,3.2,200.2,M,,,,0000*0E"
```

NMEA Data

```
"$GPGGA,064036.289,4836.53750,N,00740.93730,E,1,04,3.2,200.2,M,,,,0000*0E"
```

$GPGGA	**Global Positioning System Fix Data**
064036.289	**Time**
4836.53750,N	**Latitude 48 deg 36.537’ N**
00740.93730,E	**Longitude 07 deg 40.937’ E**
1	**Quality**
04	**Number of satellites being tracked**
200	**Altitude, Meters, above mean sea level**

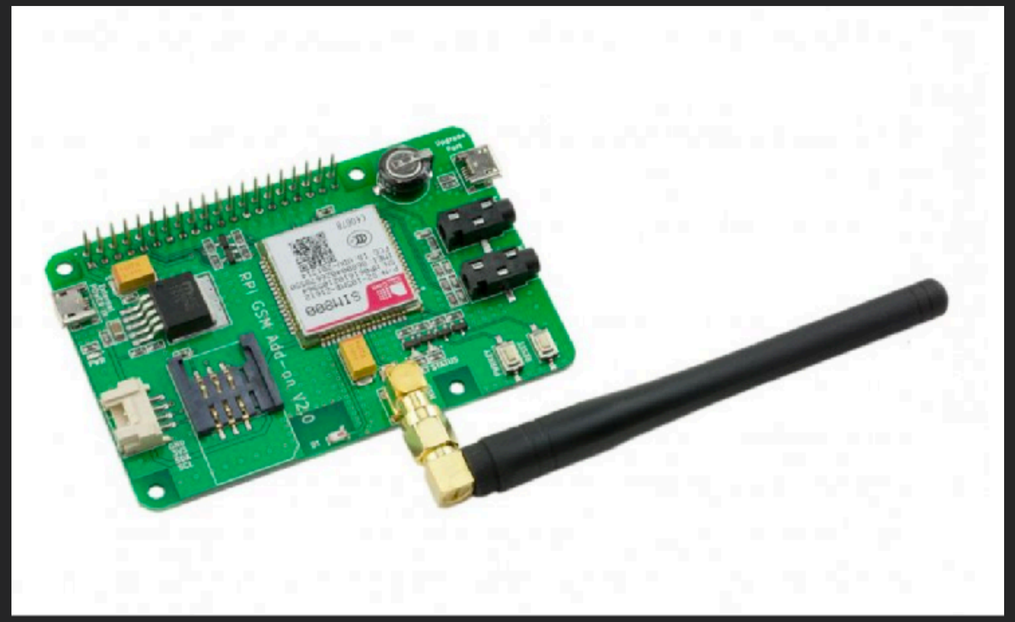
NMEA Module

```
defmodule GpsTracker.Nmea do
  def parse("$GPGGA," <> data) do
    data
    |> String.split(",")
    |> extract_time()
    |> extract_latitude()
    |> extract_longitude()
    |> compute_coordinates()
  end
end
```

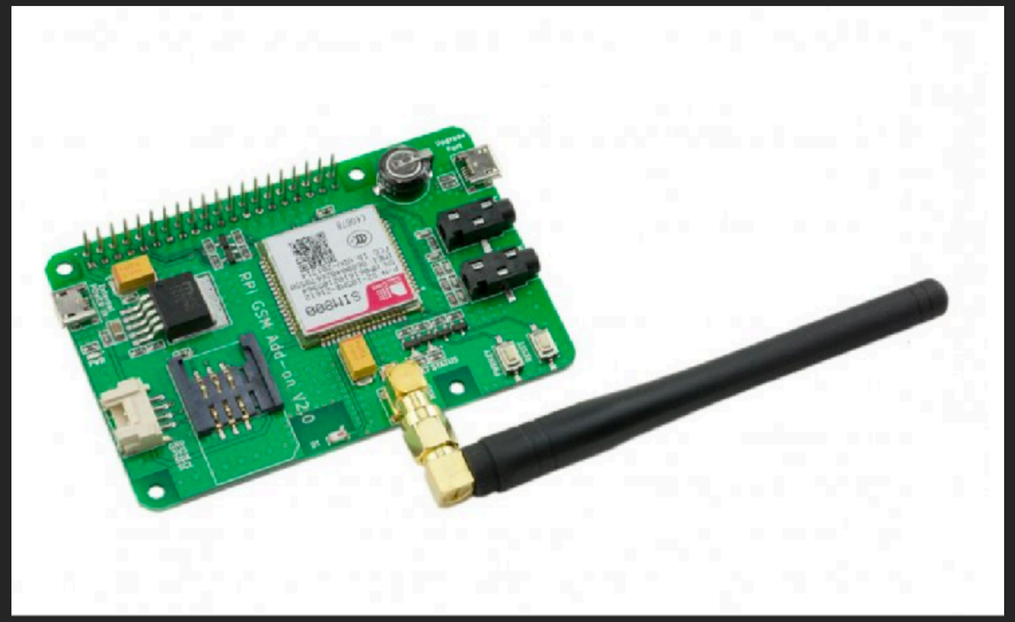
```
iex> GpsTracker.Nmea.parse(
    “$GPGGA,064036.289,4836.53750,N,00740.93730,E,1,04,3.2,200.2,M,,,,0000*0E"
  )

  {
    :ok,
    %{time: "064036.289", latitude: 48.608958333333, longitude: 7.682288333333}
  }
```

How to send position?

GSM Board



GSM Board



Sigfox

Wireless Chip

Nerves Network

Nerves Network

```
# mix.exs file - Update dependency

defp deps do
  [
    # …
    # Wifi
    {:nerves_network, "~> 0.5”}

  ]
end
```

Nerves Network

```
# config/config.exs
config :shoehorn,
  init: [:nerves_runtime, :nerves_init_gadget, :nerves_network],
  app: Mix.Project.config()[:app]

# …
config :nerves_network, :default,
  wlan0: [
    networks: [
      [
        ssid: "Iphone de Yann",
        psk: "Foobar",
        key_mgmt: “WPA-PSK”
      ]
    ]
  ]
```

```elixir
defmodule GpsTracker.Transpondeur do
  use GenServer
  # Some Other Stuff here

  def handle cast({:emit, position}, endpoint) do
    {:ok, json} = Poison.encode(position)

    HTTPotion.post(endpoint,
          body: json,
          headers: ["User-Agent": “Beacon", "Content-Type": "application/json"]
        )

    {:noreply, endpoint}
  end
end
```

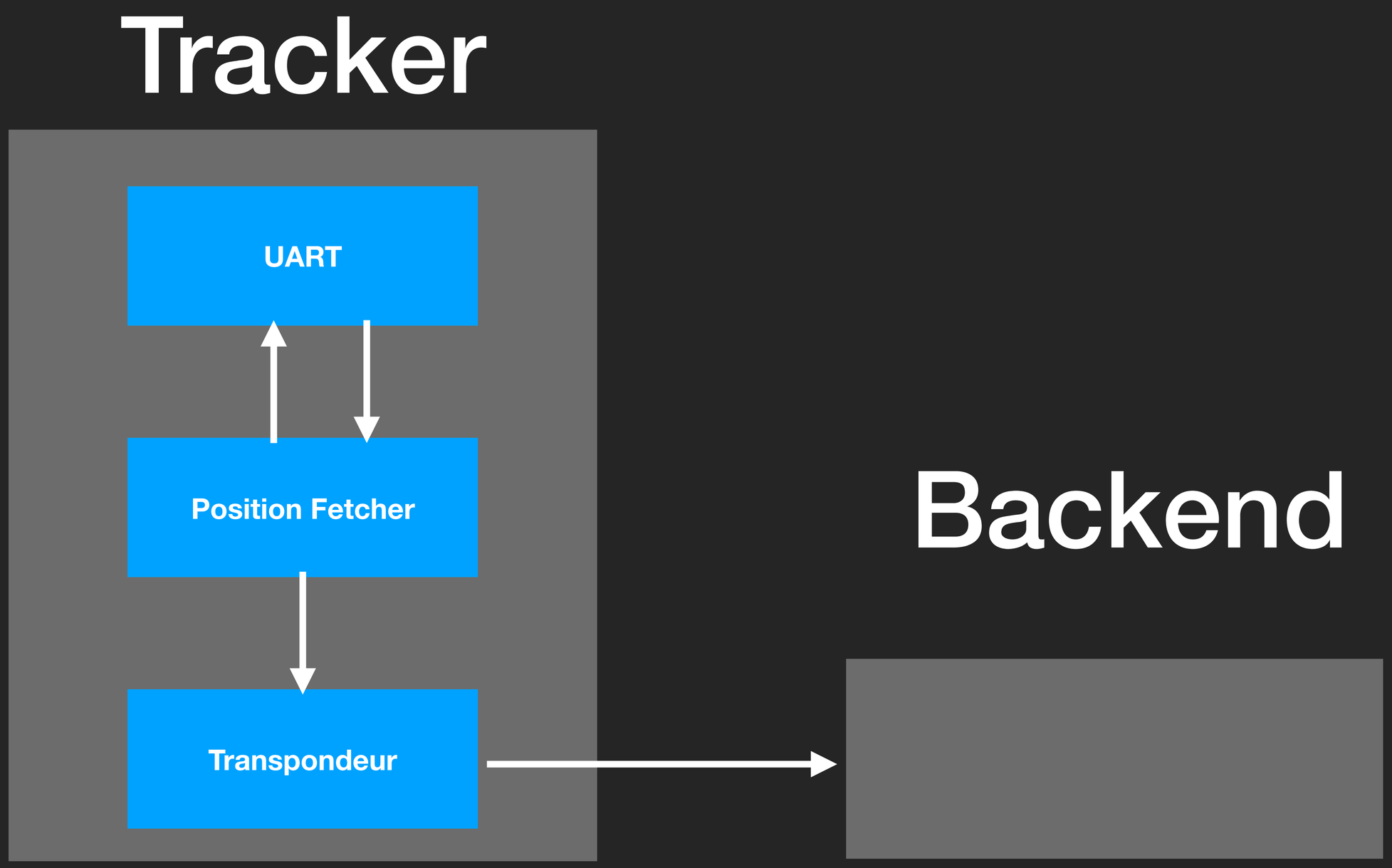
```elixir
defmodule GpsTracker.Transpondeur do
  use GenServer
  # Some Other Stuff here

  def handle_cast({:emit, position}, endpoint) do
    {:ok, json} = Poison.encode(position)

    HTTPotion.post(endpoint,
          body: json,
          headers: ["User-Agent": “Beacon", "Content-Type": "application/json"]
        )

    {:noreply, endpoint}
  end
end
```

```
defmodule GpsTracker.Transpondeur do
  use GenServer
  # Some Other Stuff here

  def handle_cast({:emit, position}, endpoint) do
    {:ok, json} = Poison.encode(position)

    HTTPotion.post(endpoint,
          body: json,
          headers: ["User-Agent": “Beacon", "Content-Type": "application/json"]
        )

    {:noreply, endpoint}
  end
end
```

```elixir
defmodule GpsTracker.Transpondeur do
  use GenServer
  # Some Other Stuff here

  def handle cast({:emit, position}, endpoint) do
    {:ok, json} = Poison.encode(position)

    HTTPotion.post(endpoint,
          body: json,
          headers: ["User-Agent": “Beacon", "Content-Type": "application/json"]
        )

    {:noreply, endpoint}
  end
end
```

```elixir
defmodule GpsTracker.Transpondeur do
  use GenServer
  # Some Other Stuff here

  def handle_cast({:emit, position}, endpoint) do
    {:ok, json} = Poison.encode(position)

    HTTPotion.post(endpoint,
          body: json,
          headers: ["User-Agent": “Beacon", "Content-Type": "application/json"]
        )

    {:noreply, endpoint}
  end
end
```

```elixir
defmodule GpsTracker.Transpondeur do
  use GenServer
  # Some Other Stuff here

  def handle_cast({:emit, position}, endpoint) do
    {:ok, json} = Poison.encode(position)

    HTTPotion.post(endpoint,
          body: json,
          headers: ["User-Agent": “Beacon", "Content-Type": "application/json"]
        )

    {:noreply, endpoint}
  end
end
```


Tracker
UART
Position Fetcher
Transpondeur
Backend

Backend

Leaflet

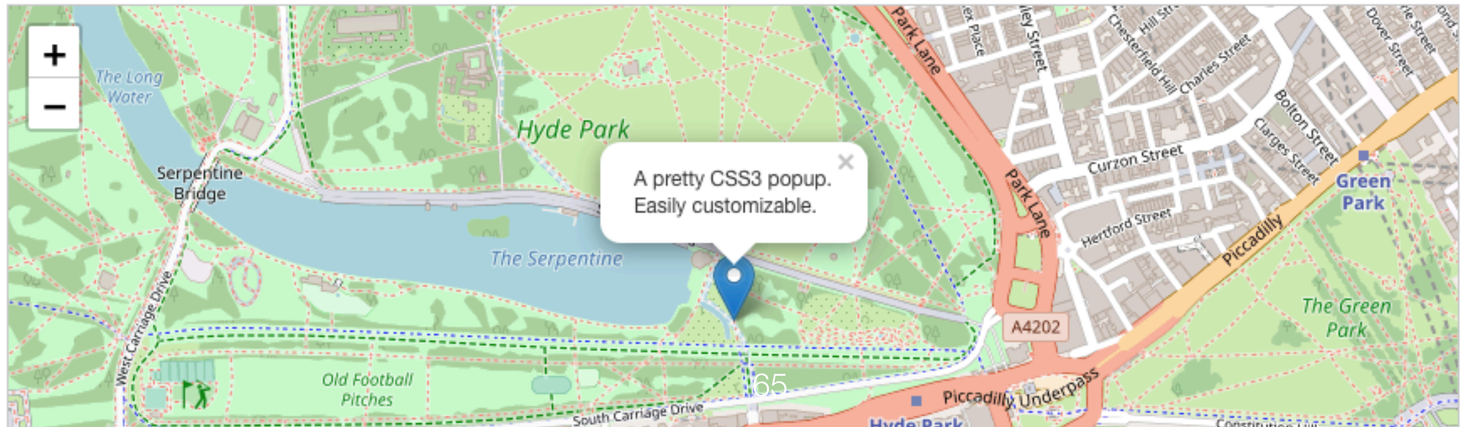
an open-source JavaScript library
for mobile-friendly interactive maps

Overview Tutorials Docs Download Plugins Blog

May 8, 2019 — Leaflet 1.5.1 has been released!

Leaflet is the leading open-source JavaScript library for mobile-friendly interactive maps. Weighing just about 38 KB of JS, it has all the mapping features most developers ever need.

Leaflet is designed with *simplicity*, *performance* and *usability* in mind. It works efficiently across all major desktop and mobile platforms, can be extended with lots of plugins, has a beautiful, easy to use and well-documented API and a simple, readable source code that is a joy to contribute to.



Leaflet

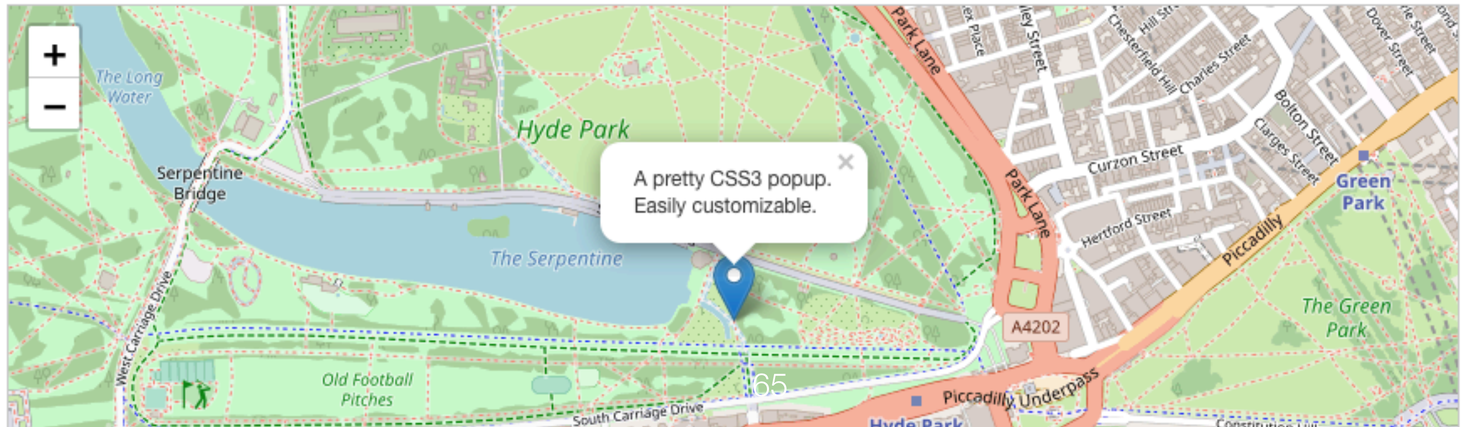
an open-source JavaScript library
for mobile-friendly interactive maps

Overview Tutorials Docs Download Plugins Blog

May 8, 2019 — Leaflet 1.5.1 has been released!

Leaflet is the leading open-source JavaScript library for mobile-friendly interactive maps. Weighing just about 38 KB of JS, it has all the mapping features most developers ever need.

Leaflet is designed with *simplicity*, *performance* and *usability* in mind. It works efficiently across all major desktop and mobile platforms, can be extended with lots of plugins, has a beautiful, easy to use and well-documented API and a simple, readable source code that is a joy to contribute to.



A1
GRAY'S INN RD
OLD ST
Mile End Park
A4200
Whitechapel
NEW RD
City of London
CABLE ST
A1203
Hello world!
I am a popup.
ondon
River Thames
SALTER RD
's Park
LONG LN
A2
Southwark Park
NEW KENT RD
MILLBANK
Leaflet | Map data © OpenStreetMap contributors, CC-BY-SA, Imagery © Mapbox

GeoJson

```
{ "type": “Feature",
        "geometry": {
          "type": "Point",
          “coordinates": [102.0, 0.0]
         }
}
```

API EndPoint

```
defmodule GpsServerWeb.Api.PositionController do
  use GpsServerWeb, :controller


  def create(conn, params) do
    {:ok, position} = GpsServer.create_position(params)
    {:ok, geo_point} =
       position
       |> GpsServer.Position.to_geojson()
       |> Jason.encode()

     conn
     |> put_resp_content_type("application/json")
     |> Plug.Conn.resp(201, geo_point)
     |> send_resp()
  end

end
```

Feedback

It’s easy

How Long ?

Hard to Debug

No SSH access

Firmware Upgrade

Hard to Debug

N

Nerves Init Gadget

ware Upgrade

Material

Material

Raspberry pi 3

Material

Raspberry pi 3

SD Card reader

Material

Raspberry pi 3

SD Card reader

GPS MODULE

Material

Raspberry pi 3

SD Card reader

GPS MODULE **X2**

Material

Raspberry pi 3

SD Card reader

GPS MODULE **X2**

Material

Raspberry pi 3

SD Card reader

GPS MODULE **X2**

Wires

Material

Raspberry pi 3

SD Card reader

GPS MODULE **X2**

Wires

Soldering Iron kit

Material

Raspberry pi 3

SD Card reader

GPS MODULE **X2**

Wires

Soldering Iron kit

1 battery

Now What ?

Now What ?

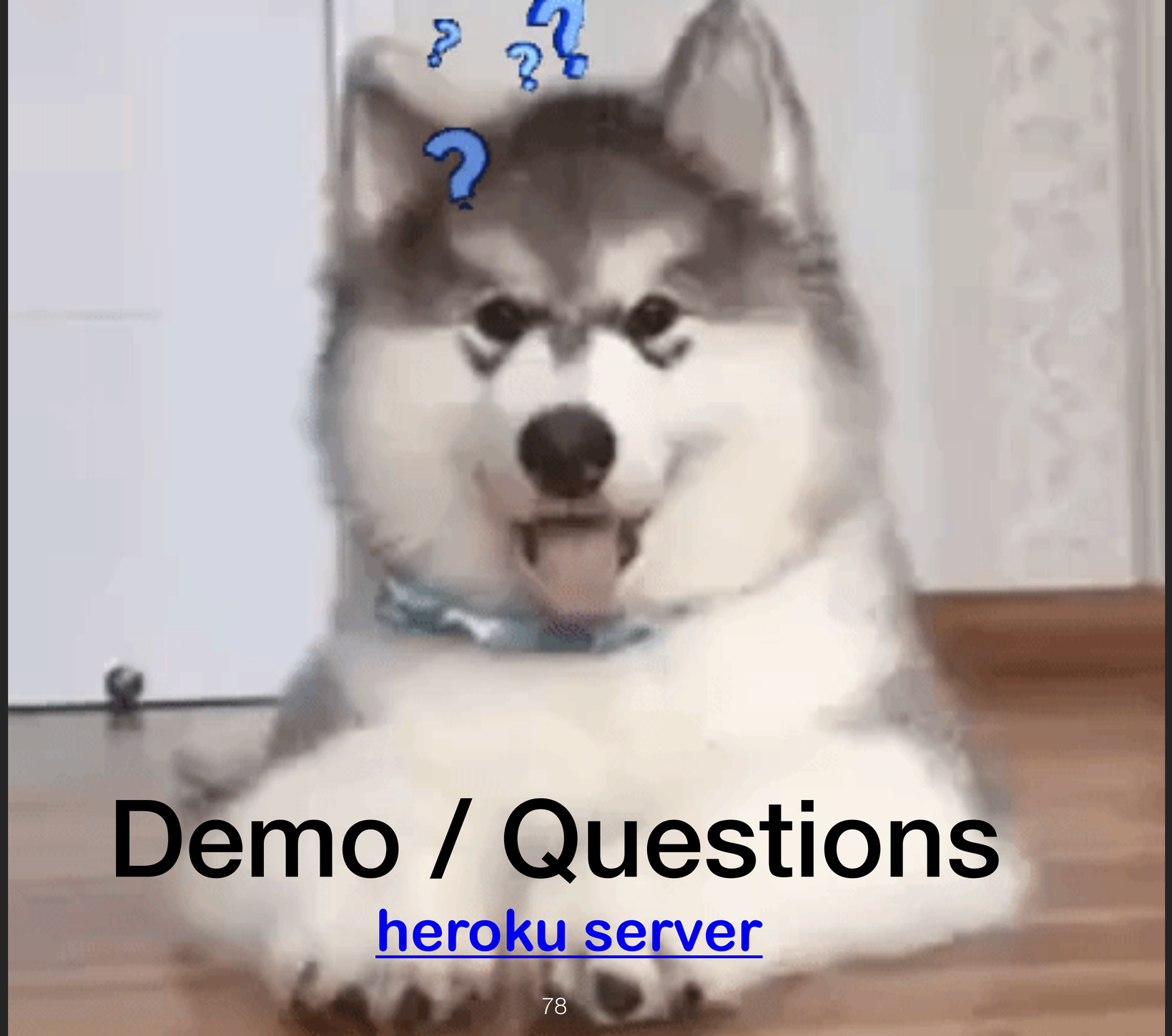
Future - Tracker

Optimizations

- power consumption
- Data consumption
- Space

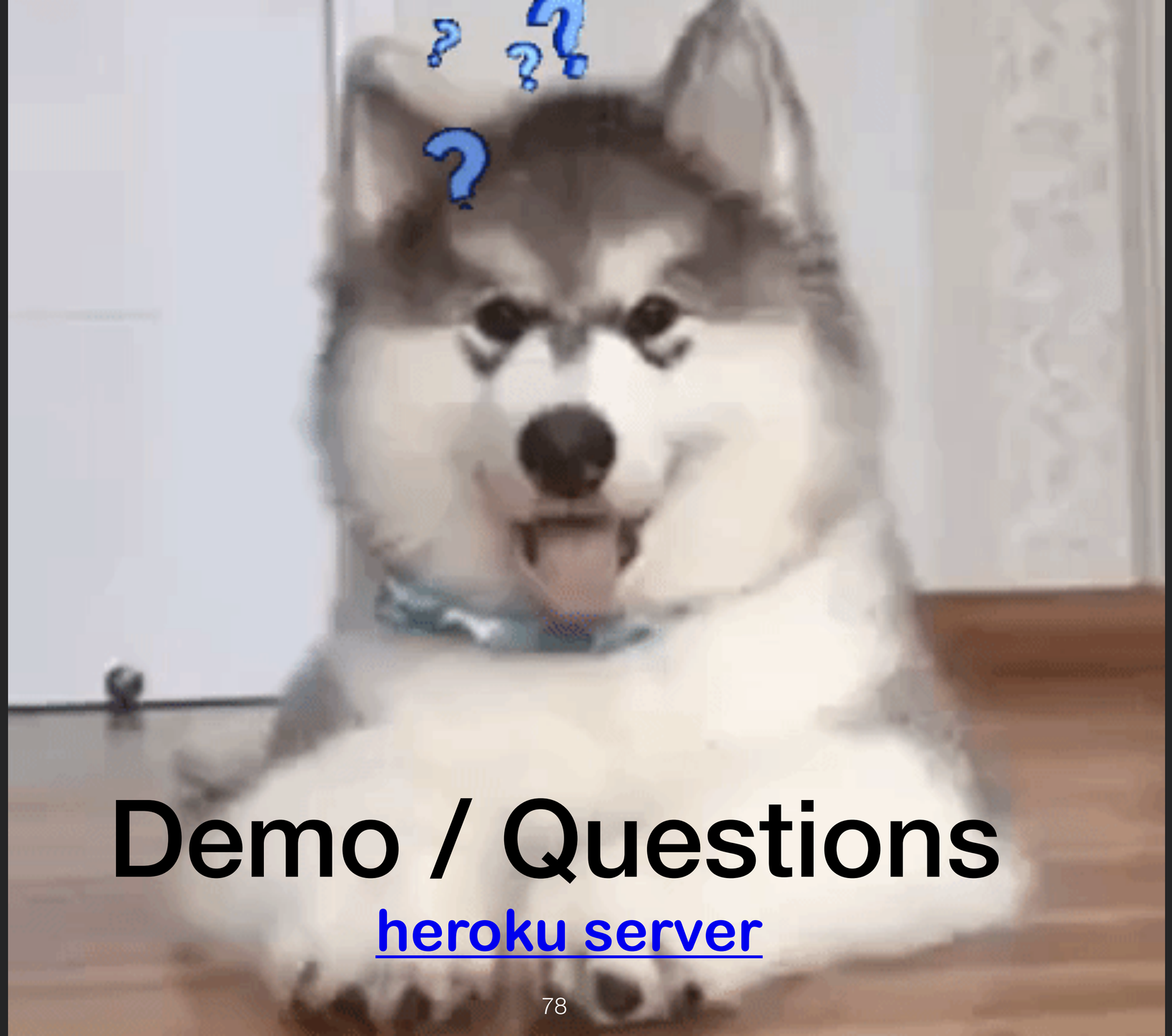
Future - Backend

- Login
- History
- Notification



Demo / Questions

heroku server



Demo / Questions

heroku server

git@github.com:yannvery/gps_tracker
git@github.com:yannvery/gps_server