How to Teach Threading to a Dolphin

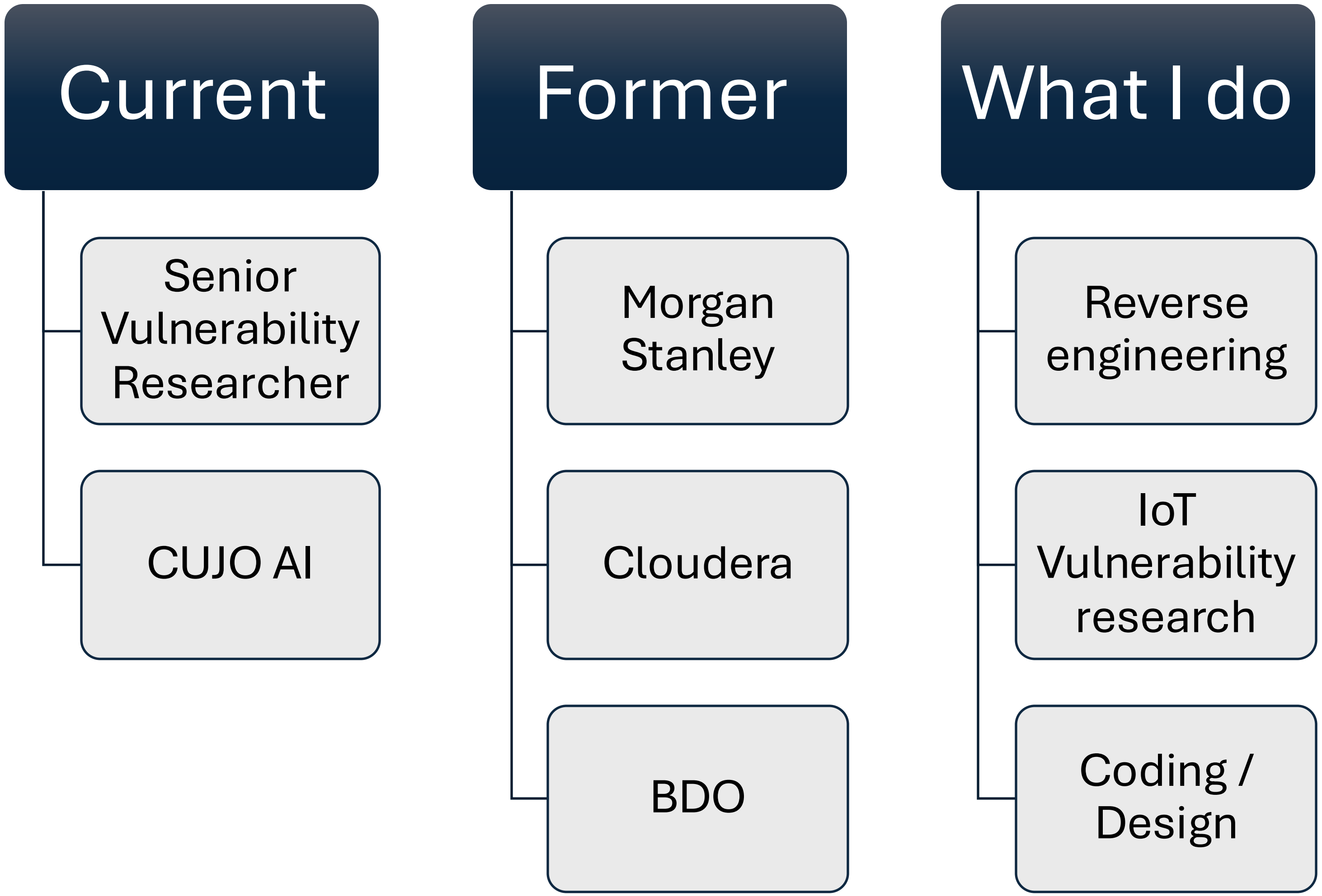
Misuse of Home IoT Networks

András Tevesz

Hacktivity
Budapest 2024

CUJOAI

About Me.
Who am I?



Current

- Senior Vulnerability Researcher
- CUJO AI

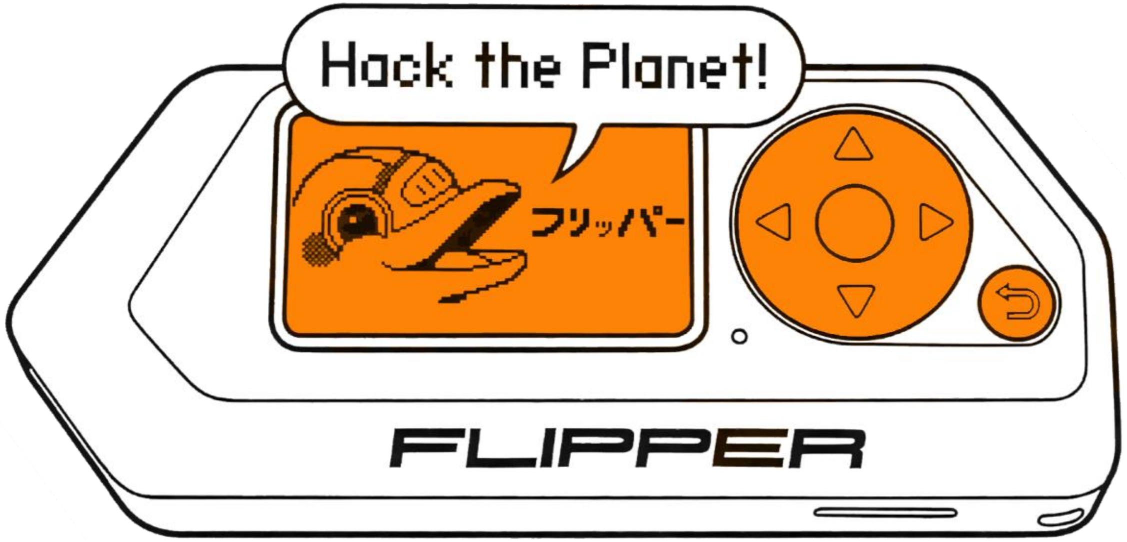
Former

- Morgan Stanley
- Cloudera
- BDO

What I do

- Reverse engineering
- IoT Vulnerability research
- Coding / Design

What is this presentation about?
Agenda



THREAD

Where did this come from?

1

I conducted a research project on Thread for CUJO AI.

2

During the research, I encountered challenges with devices, SDKs, and changing codebases.

3

I wanted to understand how the network connection could be monitored and, if necessary, blocked.

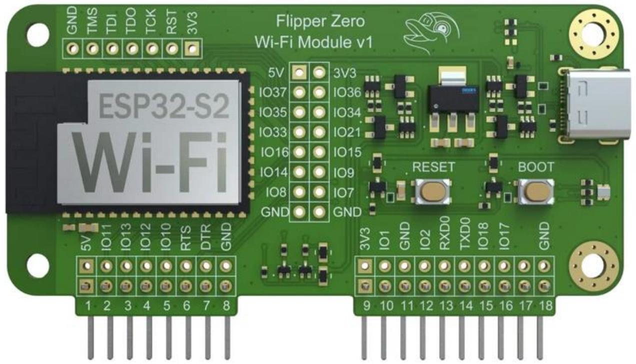
4

I found that there is no device on the market to easily interact with Thread.

What's in it for you?

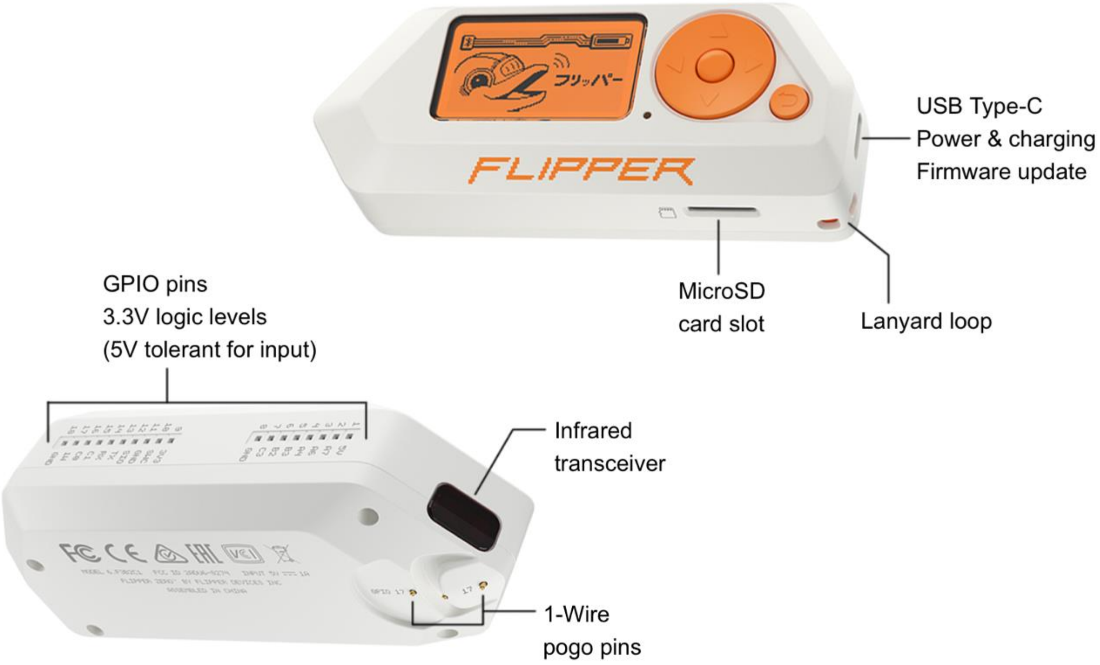
- Basic understanding of the Thread protocol
- Basic understanding of the Flipper Zero and its GPIO capabilities
- Hacking
- Opportunity to win an amazing Thread radio and some stickers

Flipper Zero Multi-tool device for geeks


Flipper Zero
Wi-Fi Module v1
ESP32-S2
Wi-Fi
GND TMS TDI TDO TCK RST 3V3
5V IO37 IO35 IO33 IO16 IO14 IO8 GND
3V3 IO36 IO34 IO21 IO15 IO9 IO7 GND
RESET
BOOT
5V IO11 IO13 IO12 IO10 RTS DTR GND
3V3 IO1 GND IO2 RXD0 TXD0 IO18 IO17 GND
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16 17 18

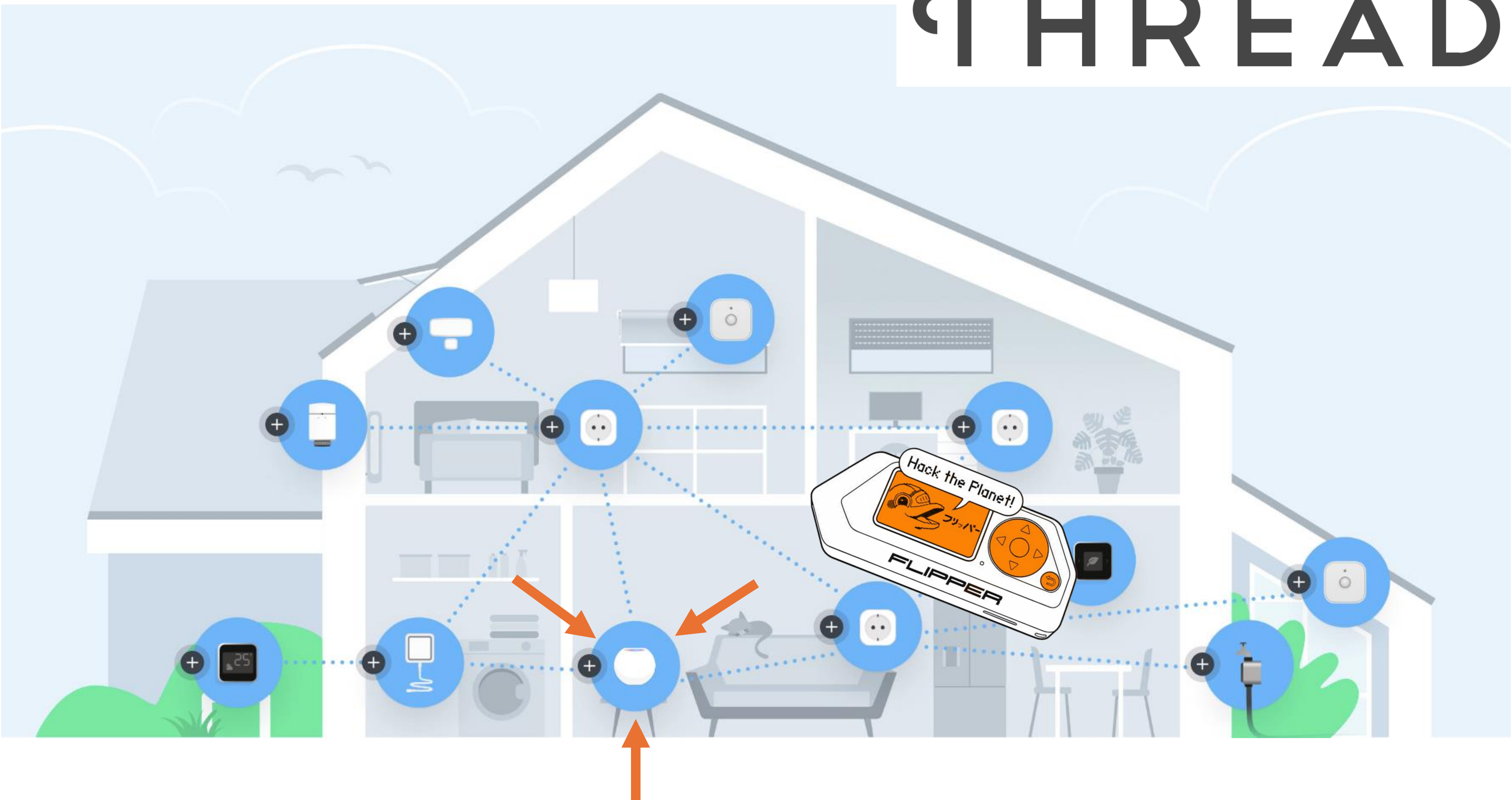
- 125 kHz **RFID**
- Sub 1 GHz Transceiver
- **NFC** High-frequency proximity cards
- **Bluetooth**
- **Infrared** Transceiver
- MicroSD card
- **USB**
- **GPIO**

- SPI, UART, I2C to USB converter
- Flashing and debugging tools



Banned in Brazil...

THREAD



Zigbee
~
Thread +Matter

	ZIGBEE	THREAD
Application layer	✓	Matter
Network layer	✓	✓
Radio	2.4 GHz + 868, 915 MHz	2.4 GHz only
IPv6-based	🚫	✓
Single point of failure	One single Coordinator	Elected Leader role
Supported devices	65,000	250 devices per border router

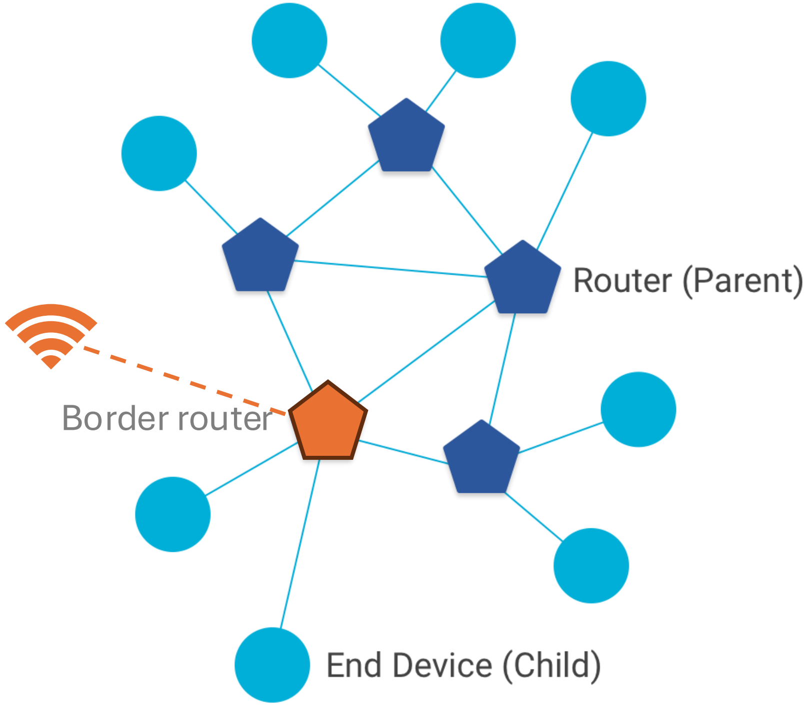
But who uses Thread?

THREAD
CERTIFIED
PRODUCTS
BUILT ON
THREAD
THREAD
CERTIFIED
COMPONENT
BUILT ON
THREAD
BORDER
ROUTER
BUILT ON
THREAD
REQUIRES
BORDER
ROUTER

Thread version 1.[1..4]

NRF 52840 SOC

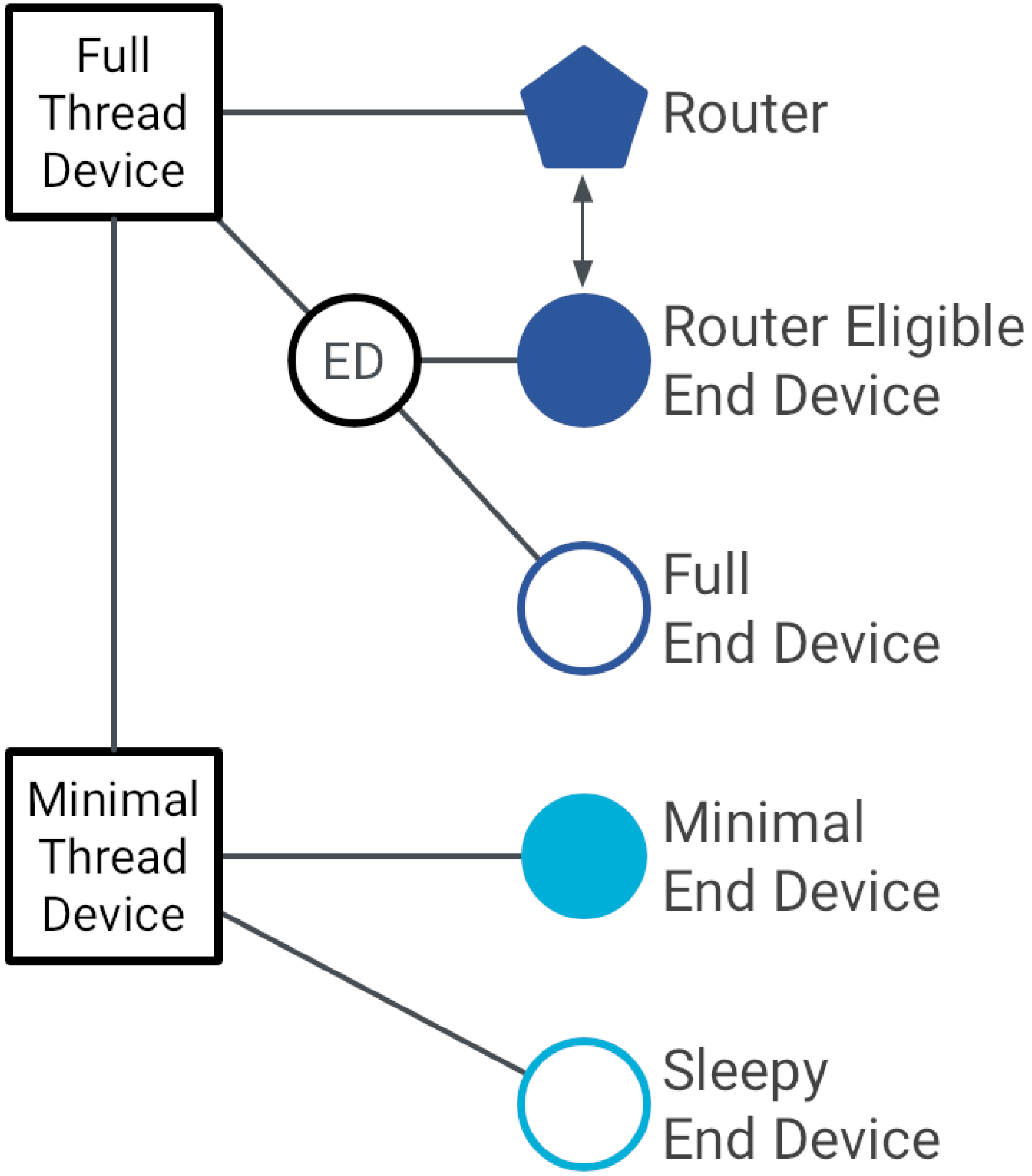
Node Roles



Thread Roles	A node in the Thread network:
Router	- forward packets for other devices - accepts joiners - keeps radio on
Border Router	- relays between Thread and non-Thread - act as a gateway
End Devices ED	- communicates with a single router - does NOT forward packets - can disable its radio

Device Types

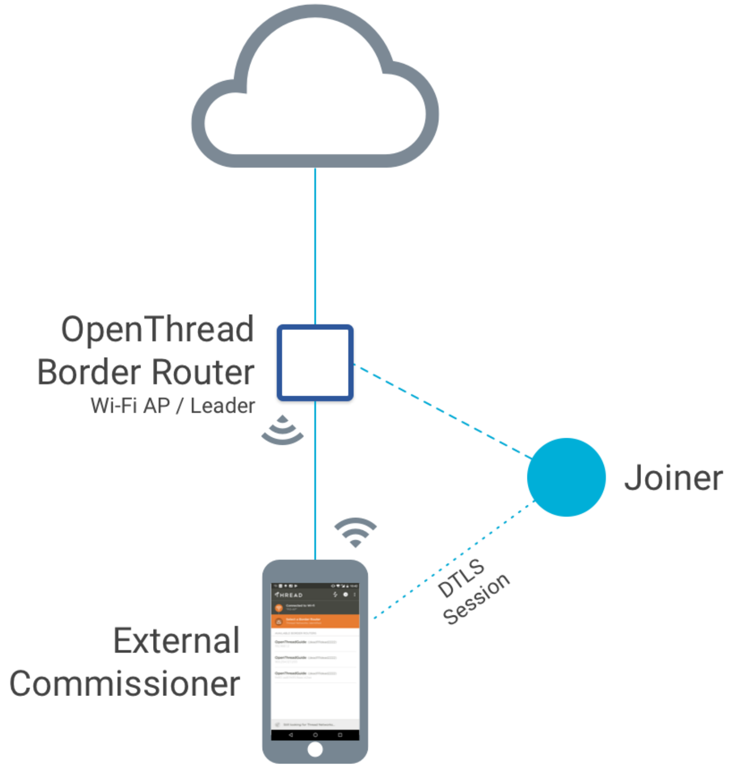
Thread Device Types	
Full Thread Device **FTD**	It can be a **Router** and an **End device** (radio always on)
Minimal Thread Device **MTD**	Its always an **End device,** communicates with its parent
Minimal End Device **MED**	Keeps transceiver always on
Sleepy End Device **SED**	Wakes up occasionally to receive from its parent
Synchronized Sleepy End device **SSED**	Only transmits in a specified time interval

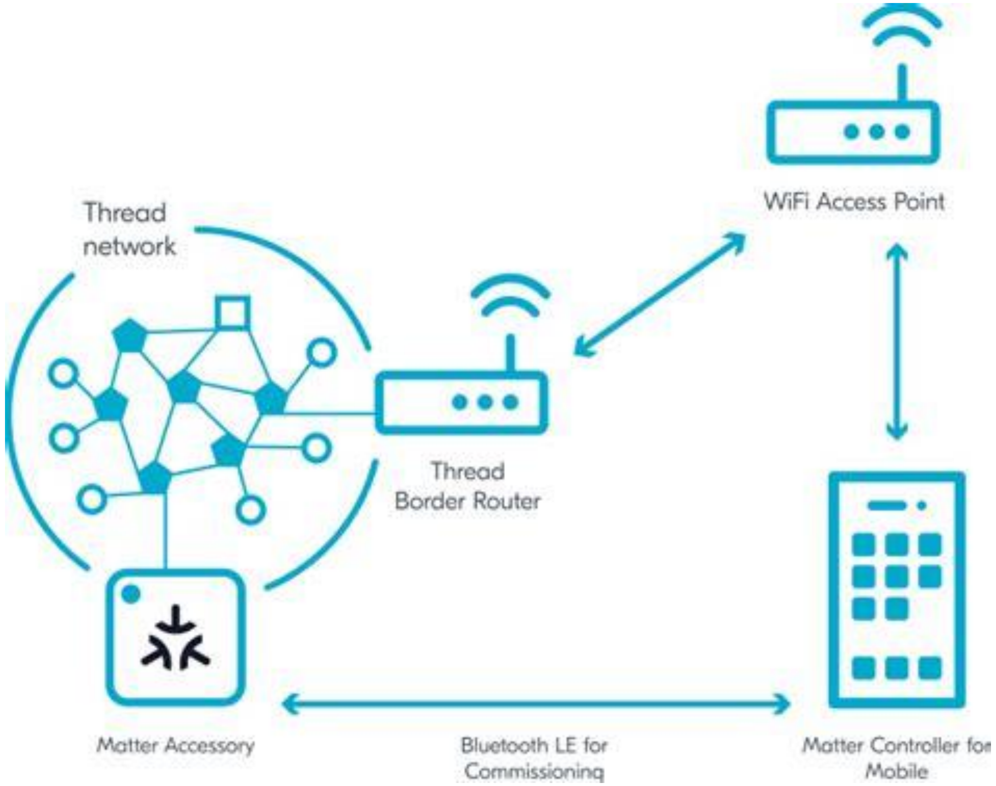


Thread Commissioning

Commissioner	Authenticates the Joiner
Commissioner Candidate	A commissioner who could be promoted by leader
Joiner	A device who wants to join to the Thread Network
Border Router	Gateway between Thread and non-Thread Networks
Border Router Agent	Accepting commissioner candidates and relays between the network and the Commisioner
Backbone Router	Device roaming and multicast forwarding, with Thread Domains
Leader	Maintains Thread network configuration promote candidates, ensures only one commisioner

Ok, but how can we connect to the network?





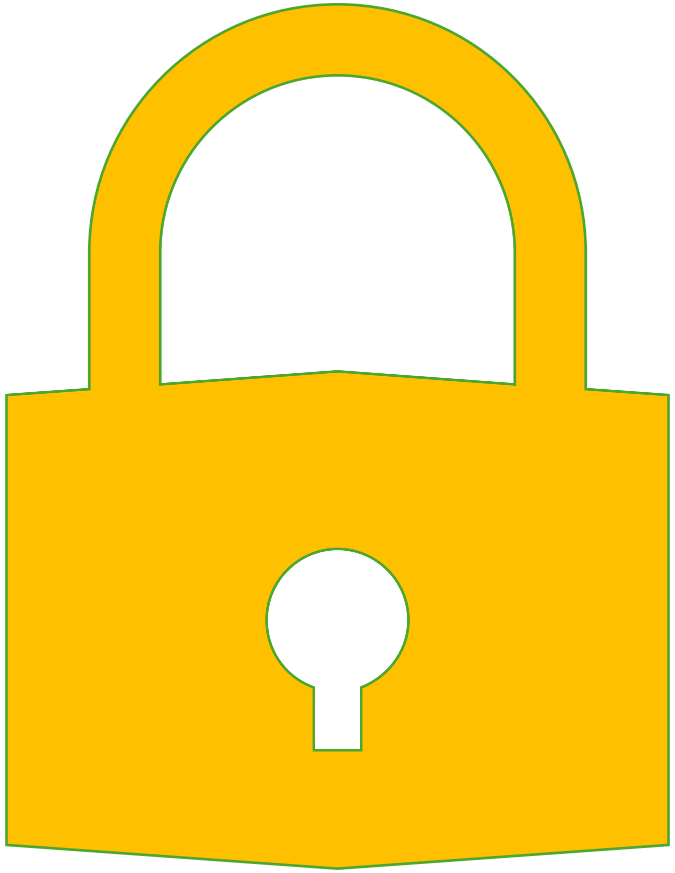
How Matter should work

Is there another way to connect?

You know the **joiner** password, but it needs a joiner window to be usable

Use a leaked **dataset**

Use a known **network key** and the **PAN**



How do we get the PAN id and what is it?

2-byte Personal Area Network ID (PAN ID)

uart# ot scan -> otLinkActiveScan

```
| PAN  | MAC Address          | Ch | dBm | LQI |
| 9749 | ee9afe59d77e515e  | 11 | -60 | 128   |
| e948 | 9273124c7a125bc8 | 25 | -61 | 128   |
| e948 | 866d554cead1f46f   | 25 | -57 | 152   |
```

uart# ot discover -> otThreadDiscover

```
| Network Name      | Extended PAN        | PAN  | MAC Address          |
| AMZN-Thread-9749 | f23dd4876455b41f   | 9749 | ee9afe59d77e515e  |
| MyHome44015048  | 555c7d90aea746ca | e948 | 767d9c53c6dfb1bd |
| MyHome44015048  | 555c7d90aea746ca | e948 | 866d554cead1f46f   |
| MyHome44015048  | 555c7d90aea746ca | e948 | 9273124c7a125bc8 |
```

Thread dataset

$ python3 tlv-parser.py

```
0e080000000000010000000300001235060004001fffe0
0208a1fce8946f2f9b1d0708fd505ff6fd1b325b0510e674
46d4e450ad76cd3ad5472530d410030f4f70656e546872
6561642d656539370102ee97041042743e8b67c06353c
d038520a0ab8b7f0c0402a0f7f8
```

```
t: 14 (ACTIVETIMESTAMP), l: 8, v: 0x0000000000010000
t:  0 (CHANNEL), l: 3, v: 0x000012
t: 53 (CHANNELMASK), l: 6, v: 0x0004001fffe0
t:  2 (EXTPANID), l: 8, v: 0xa1fce8946f2f9b1d
t:  7 (MESHLOCALPREFIX), l: 8, v: 0xfd505ff6fd1b325b
t:  5 (NETWORKKEY), l: 16, v: 0xe67446d4e450ad76cd3ad5472530d410
t:  3 (NETWORKNAME), l: 15, v: b'OpenThread-ee97'
t:  1 (PANID), l: 2, v: 0xee97
t:  4 (PSKC), l: 16, v: 0x42743e8b67c06353cd038520a0ab8b7f
t: 12 (SECURITYPOLICY), l: 4, v: 0x02a0f7f8
```

Thread network keys

default Open Thread

- 11112233445566778899DEAD1111DEAD
- 1234c0de7ab51234c0de7ab51234c0de
- 00112233445566778899aabbccddeeff

https://github.com/simenkid/ot-ctl/blob/main/index.js

- e947a2e6b08b8cfefa6961b5c3943928
- 89722adb7ef02054ec73111c337ec6a9

https://docs.gl-inet.com/iot/en/thread_board_router/gl-s200/openthread_border_router_codelabs/

- e67446d4e450ad76cd3ad5472530d410

Pre-Shared Key for the Commissioner (PSKc)

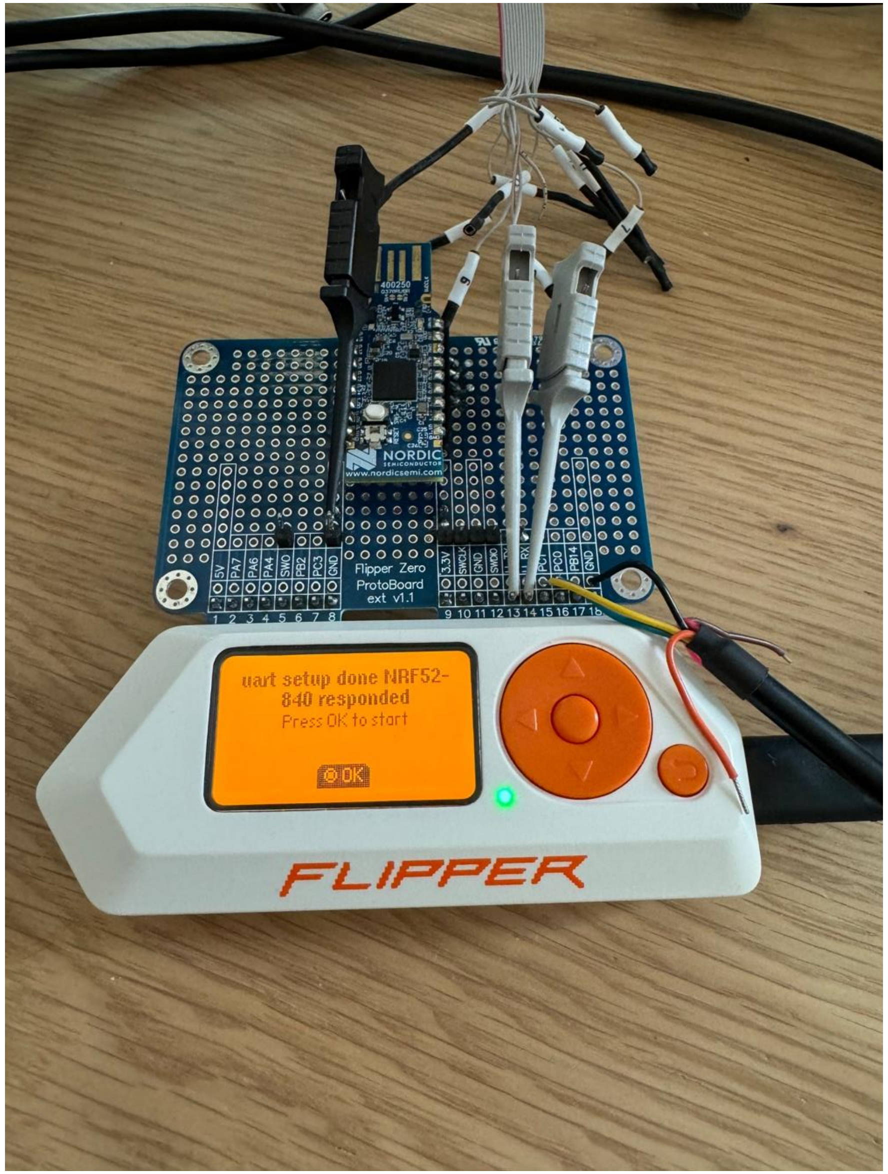
```
./pskc commissioner-credential extpanid network-name
```

```
./pskc J01NME 1234AAAA1234BBBB MyOTBRNetwork
```

```
ee4fb64e9341e13846bbe7e1c52b6785
```

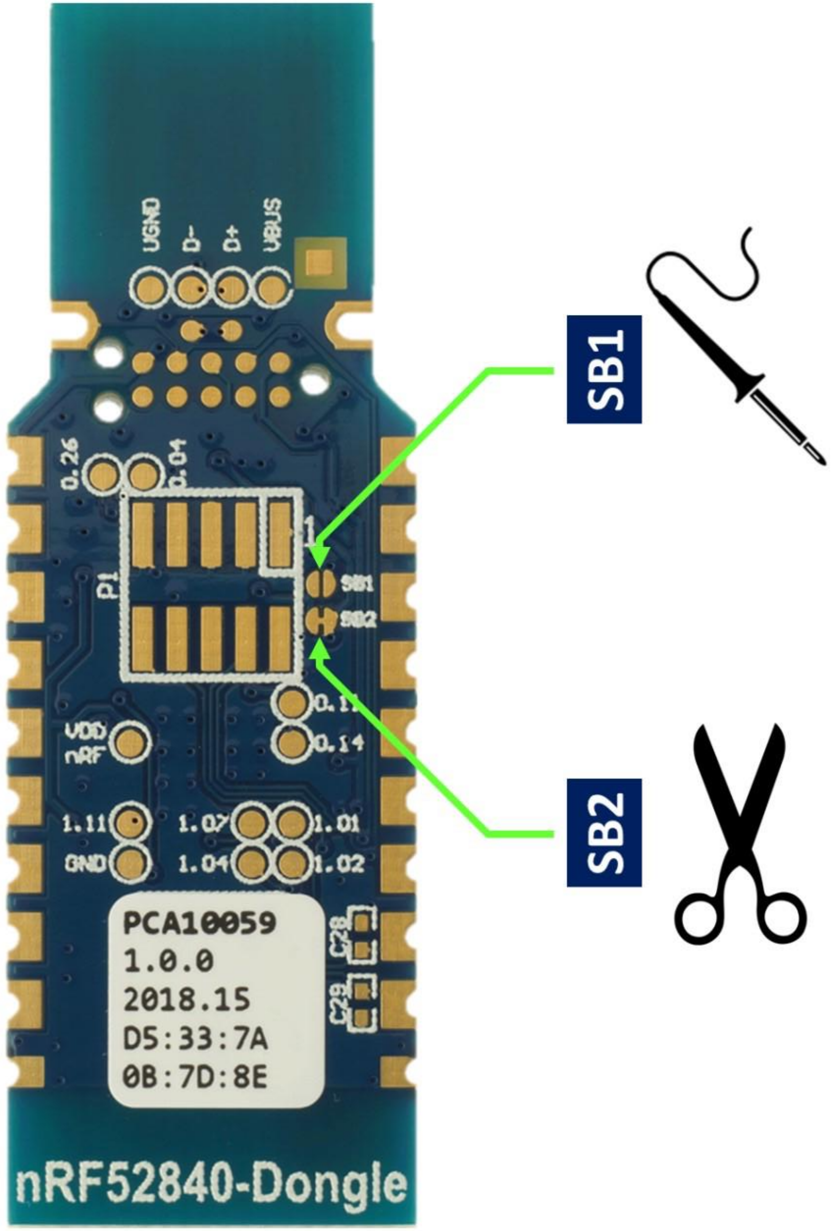
Zephyr Firmware app

ThreadFlipper

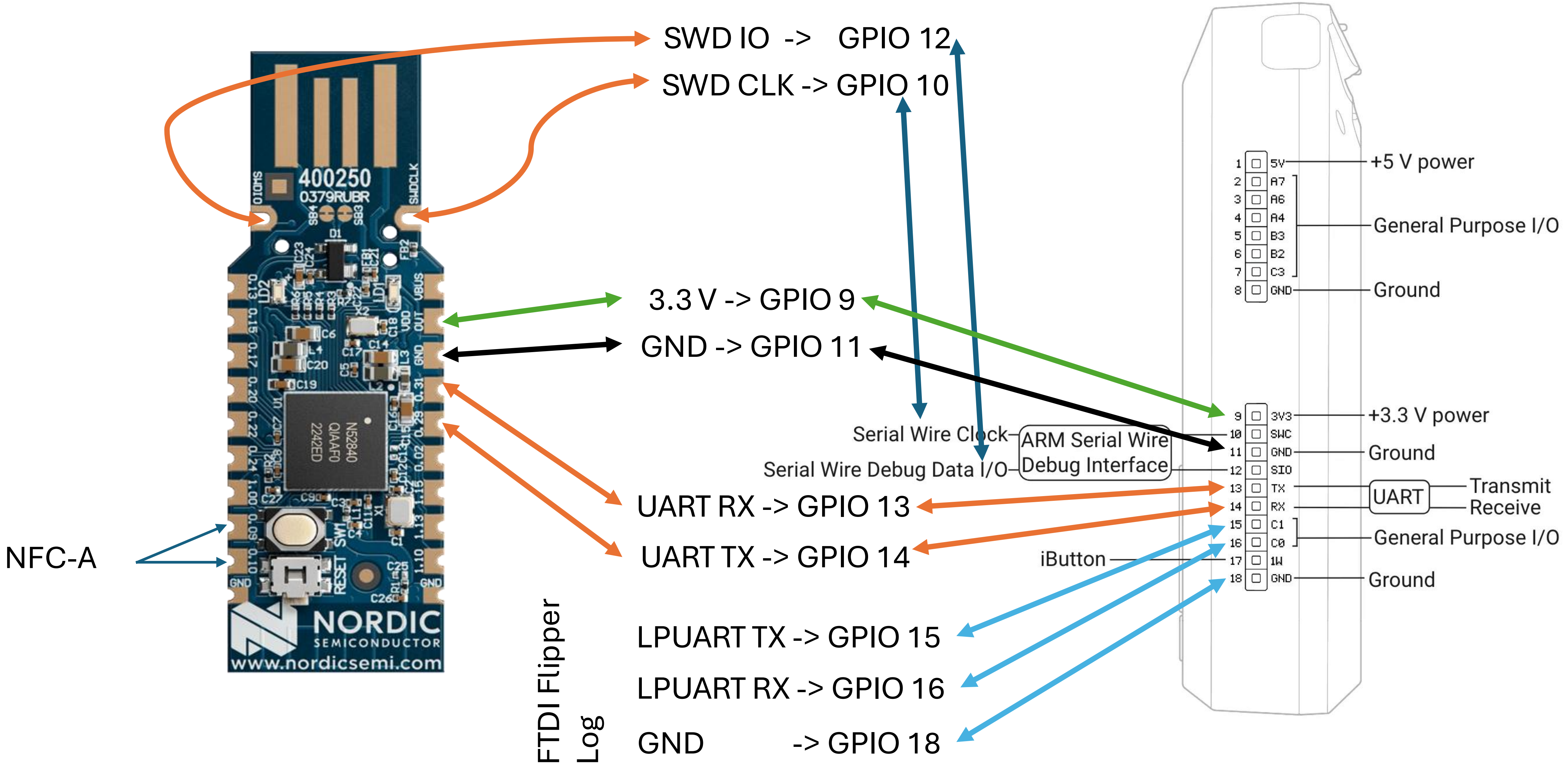

uart setup done NRF52-840 responded
Press OK to start
OK

Enable the external power supply through the VDDOUT pin

External regulated 1.8–3.6 V (max 50 mA) is supported



How to wire our new board



SWD IO -> GPIO 12

SWD CLK -> GPIO 10

3.3 V -> GPIO 9

GND -> GPIO 11

UART RX -> GPIO 13

UART TX -> GPIO 14

NFC-A

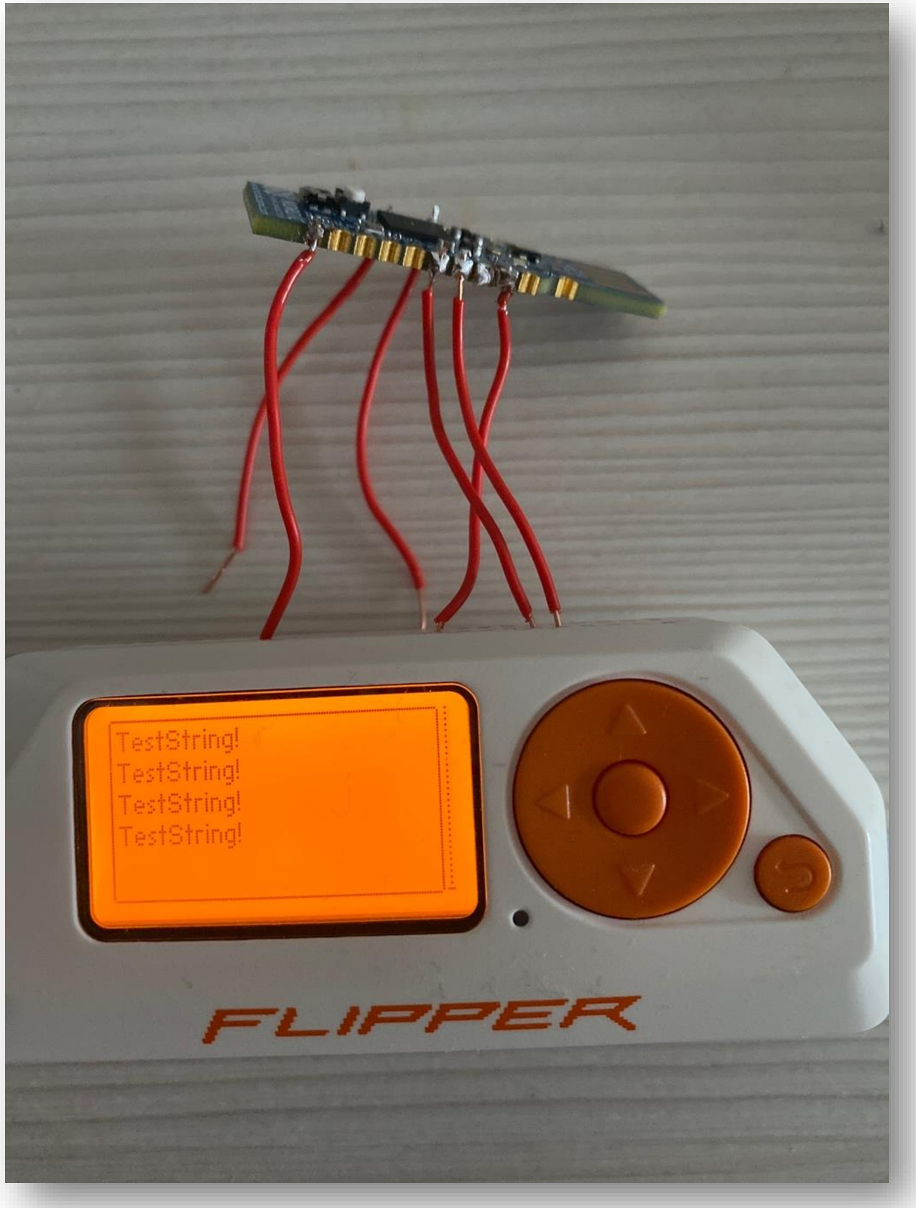
FTDI Flipper Log

LPUART TX -> GPIO 15

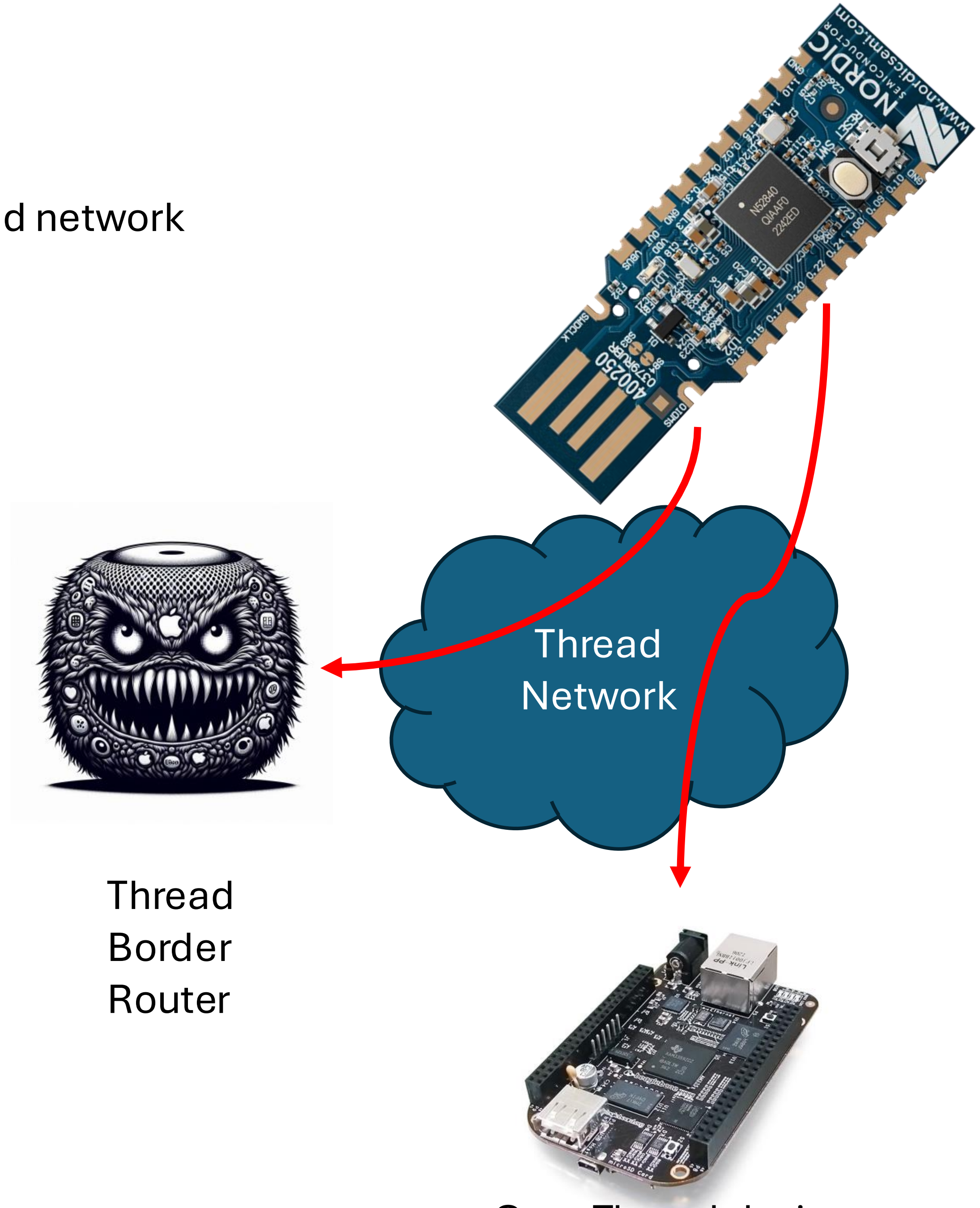
LPUART RX -> GPIO 16

GND -> GPIO 18

DEMO



- Thread Discover
- Connect to Thread network

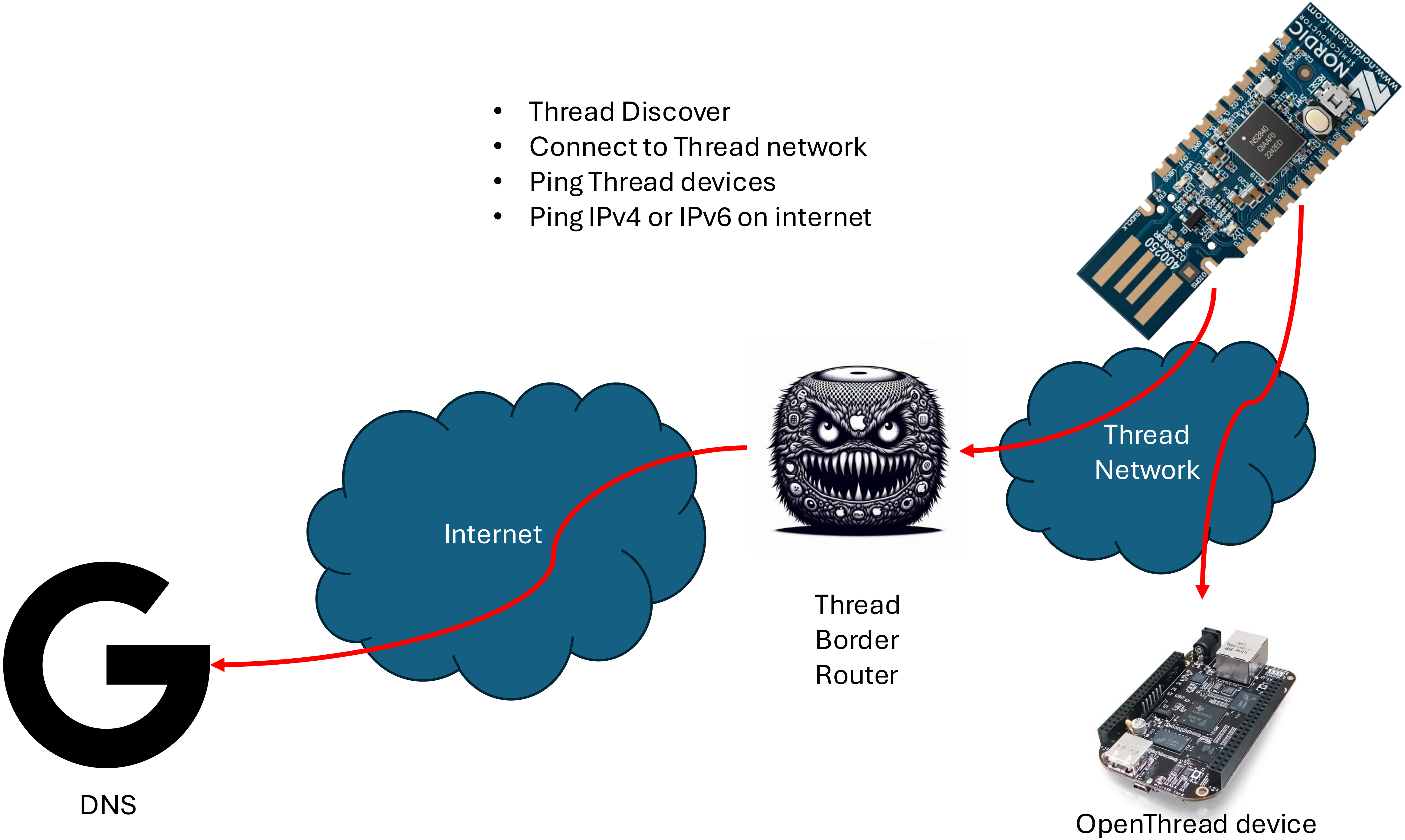


Thread Network

Thread
Border
Router

OpenThread device

- Thread Discover
- Connect to Thread network
- Ping Thread devices
- Ping IPv4 or IPv6 on internet



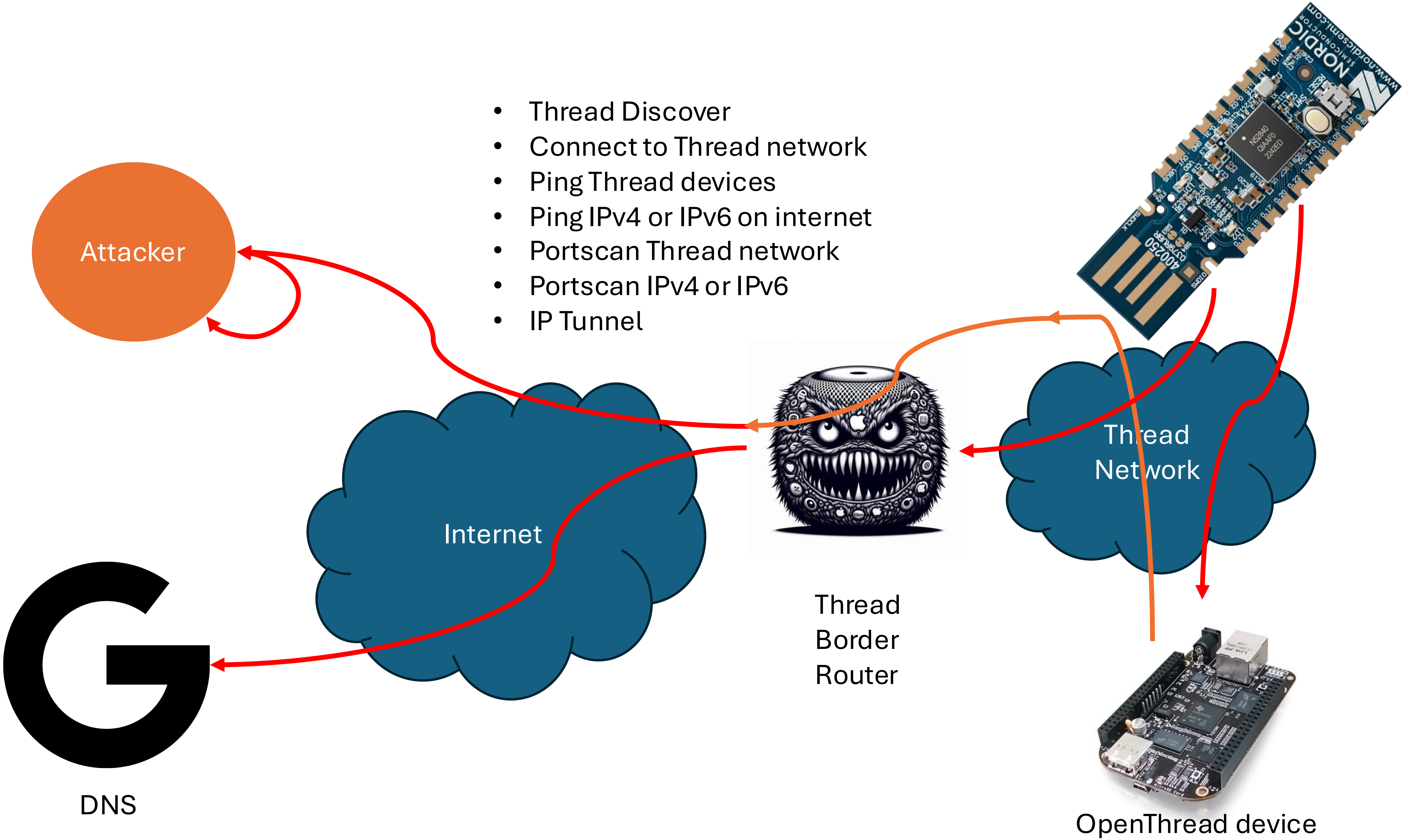
Internet

Thread Network

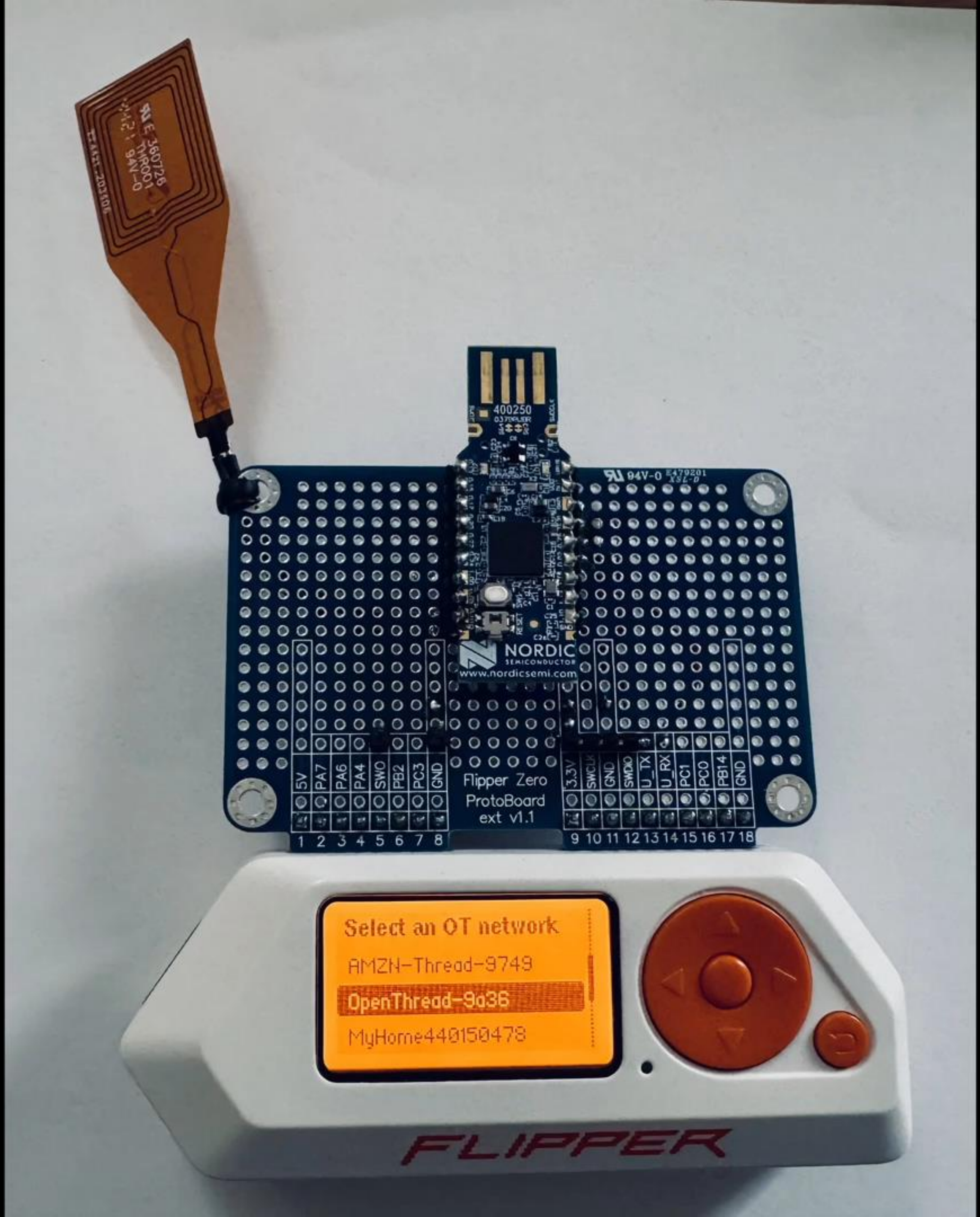
Thread Border Router

DNS

OpenThread device



- Thread Discover
- Connect to Thread network
- Ping Thread devices
- Ping IPv4 or IPv6 on internet
- Portscan Thread network
- Portscan IPv4 or IPv6
- IP Tunnel

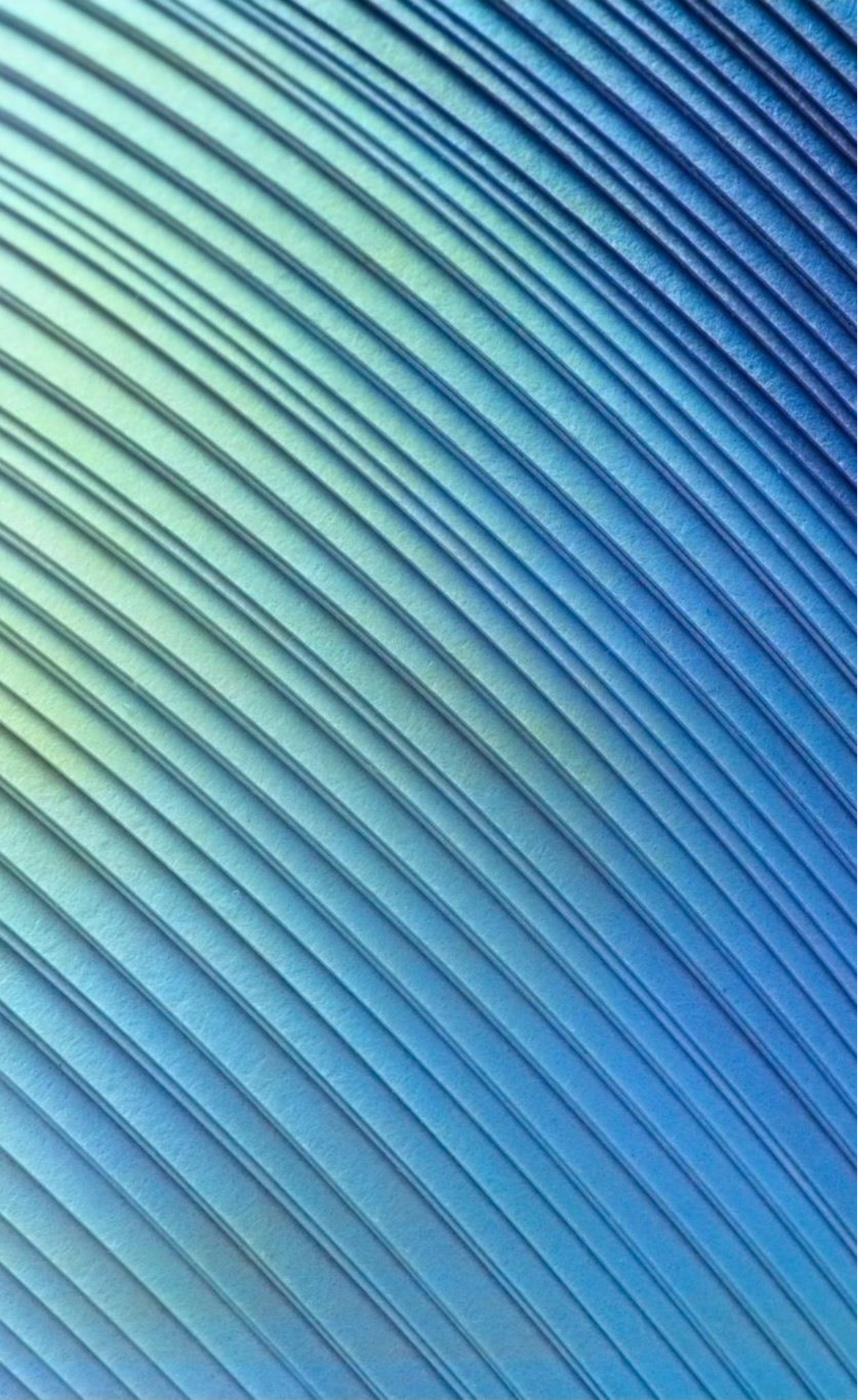


ThreadFlipper
Demo

We are open sourcing the projects

- Open source the NRF firmware app
- Open source the Flipper Zero JS scripts

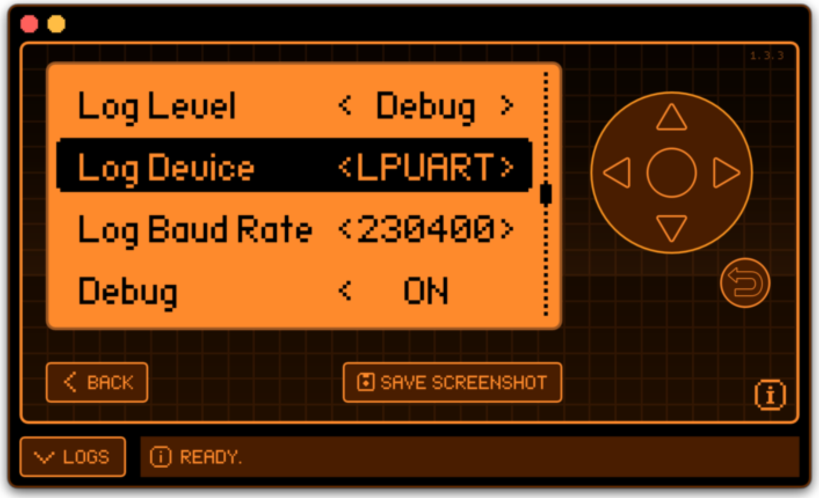
https://github.com/getCUJO/ThreatIntel/tree/master/Research_materials/ThreadFlipper



Whats next ?

Native APP	Finish the native Flipper Zero app not just JS script
SWD	Integrate SWD and support automated flashing of firmware images
NFC	Integrate an NFC antenna .
Protection	Add some protection to the PCB (reverse polarity, voltage regulator, hotplug support)
Matter	Support Matter, enable Thread key extraction
5V	Use the 5v power from Flipper Zero with a voltage regulator to provide more juice for thread

Challenges



Log Level < Debug >

Log Device <LPUART>

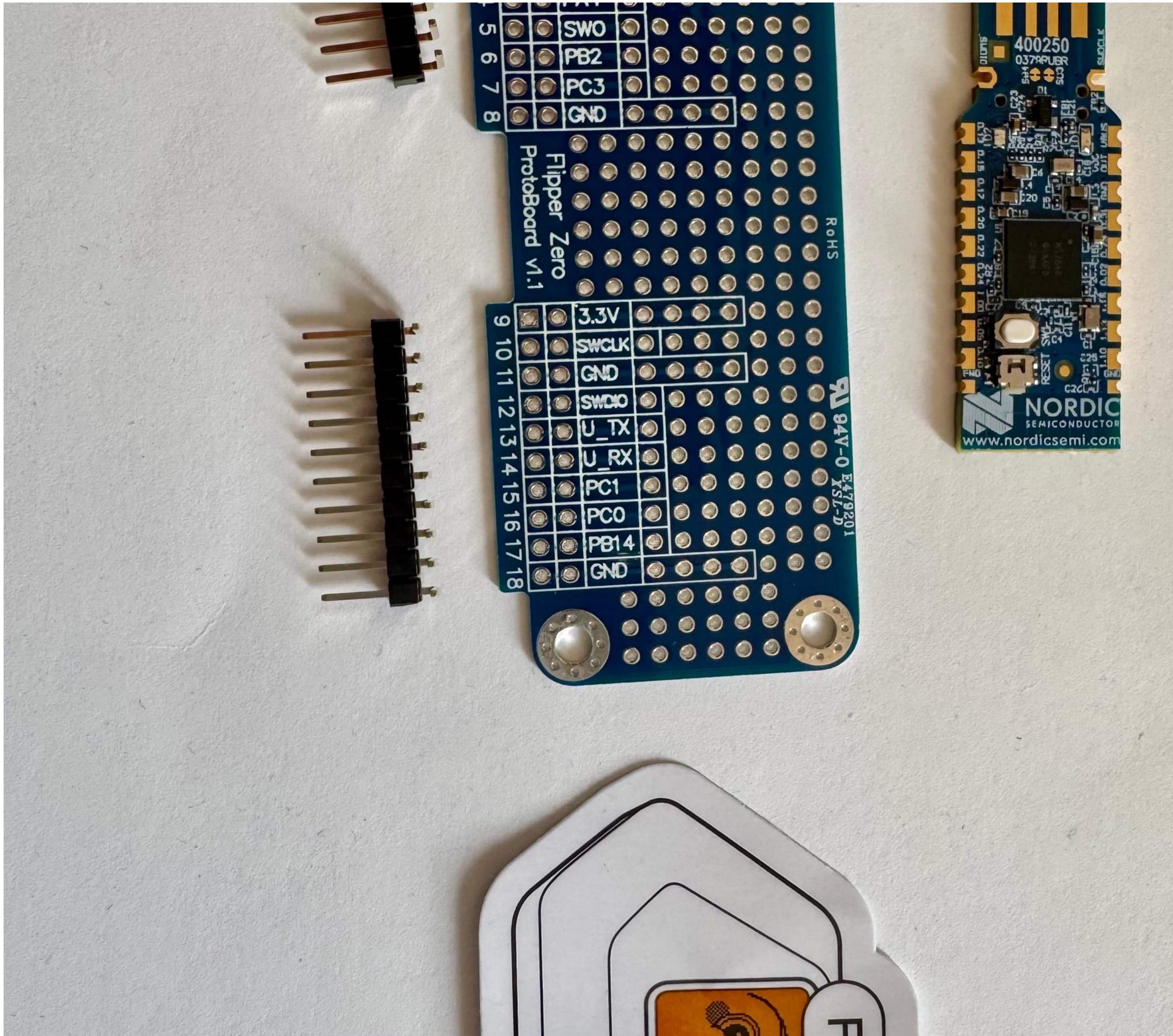
Log Baud Rate <230400>

Debug < ON

- Debugging a Flipper App with a connected Thread board via a WI-Fi extension board is impossible as they use the same UART IO ports. Moving to LPUART will not help, as you will lose the Flipper Logs
- Jumper Wires can be used to connect only SWD pins for the WI-FI extension debugger
- There is no documentation explaining how the esp32 Blackmagic debugger uses the SWD pins
- Flipper with debug mode enabled is prone to getting stuck in a pre-boot breakpoint without a screen
- Flipper support JS uses a lib called mJS (50k JS with 1k RAM), which lacks useful JS functions. The stock firmware does not support features like storage in JS
- Firmware development with Zephyr is hard, with all the possible and conflicting CONFIG parameters
- Manually set the SEGGER JLink Voltage detection to 3.3V otherwise, it SWD will fail
- SEGGER JLink might help to recover from a seemingly bricked flipper (8x times)
- Adding your own pins for the SWD port supports JLink SWD debug

What protocol was Thread's predecessor?

Your questions?

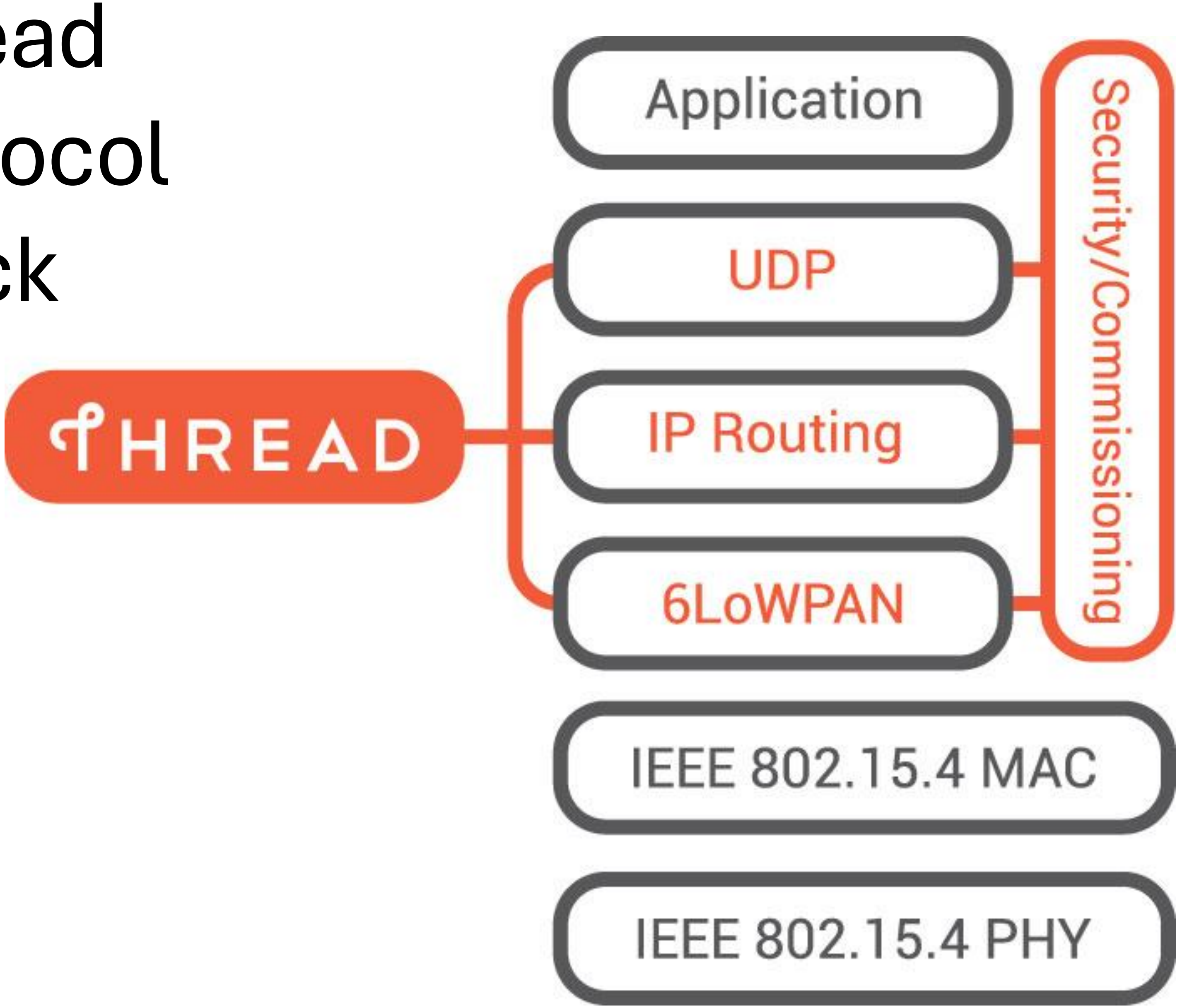


So Long, and Thanks for All the Fish!

Appendings

https://docs-be.nordicsemi.com/bundle/ug_nrf52840_dongle/attach/nRF52840_Dongle_User_Guide_v2.1.1.pdf?_LANG=enus https://docs-be.nordicsemi.com/bundle/ps_nrf52840/attach/nRF52840_PS_v1.11.pdf?_LANG=enus	nRF 52840 Dongle guide, leds, pins
https://www.nordicsemi.com/Products/Development-hardware/nrf52840-dongle https://www.nordicsemi.com/Products/Development-hardware/nRF52840-DK	nRF 52840 Dongle and Development Kit sites
https://threadgroup.org https://github.com/openthread/openthread	OpenThread reference
https://flipperzero.one https://docs.flipper.net/development/hardware/modules-blueprints	Flipper Zero development
https://momentum-fw.dev/	Flipper Zero firmware with proper JS support

Thread Protocol Stack



COAP, MQTT, Matter

TCP and UDP + DTLS

MLE routing / Distance Vector Routing

Low Power IPv6 – packet fregmentation

Message and connection level

Packet sending over radio