LASAGNA [concept]

Low power Autonomous System for Adaptive Generalised Naval Assistance

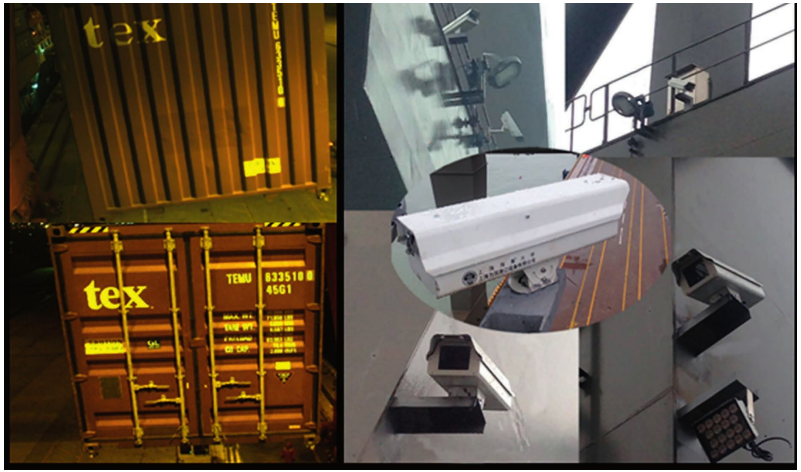
Alessio Cappellini, Dario Petrillo
May 27th, 2022

The challenge

Ports currently track ships and trucks, but individual containers are harder to track.

Solutions with cameras exist, but they can't track containers hidden behind others

We want to track containers and optimize their transit in a port

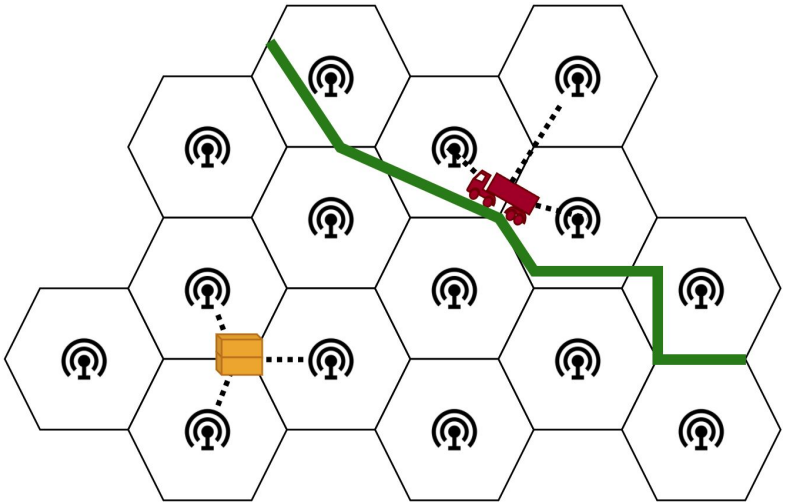


Computer vision systems in a port

Our solution

It is designed for port operators helping to identify containers that are shipped to the wrong place but also needlessly long routes.

How? Fitting containers with a small, cheap and reusable beacon, and listen for messages with receivers spread throughout the port area.



LASAGNA [architecture]

Low power Autonomous System for Adaptive Generalised Naval Assistance

Alessio Cappellini, Dario Petrillo
May 27th, 2022

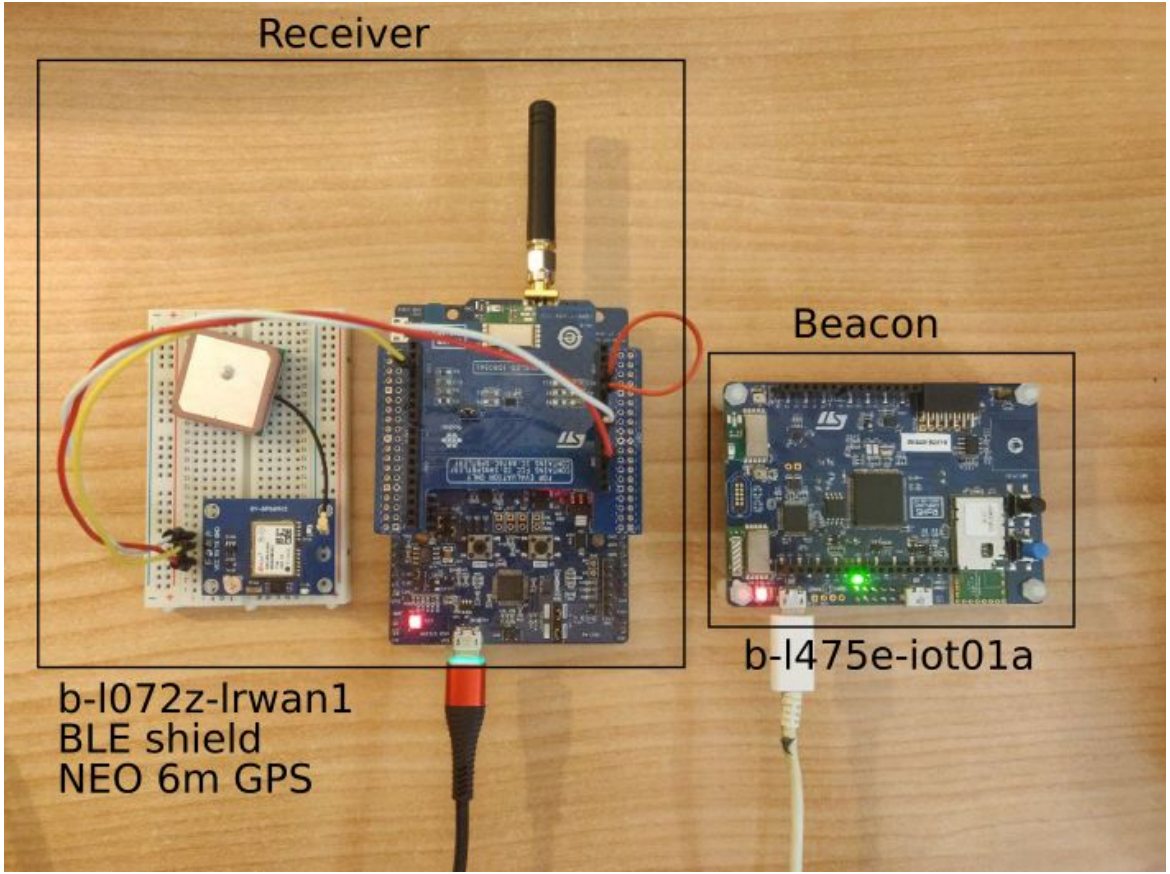
Hardware setup

Beacon:

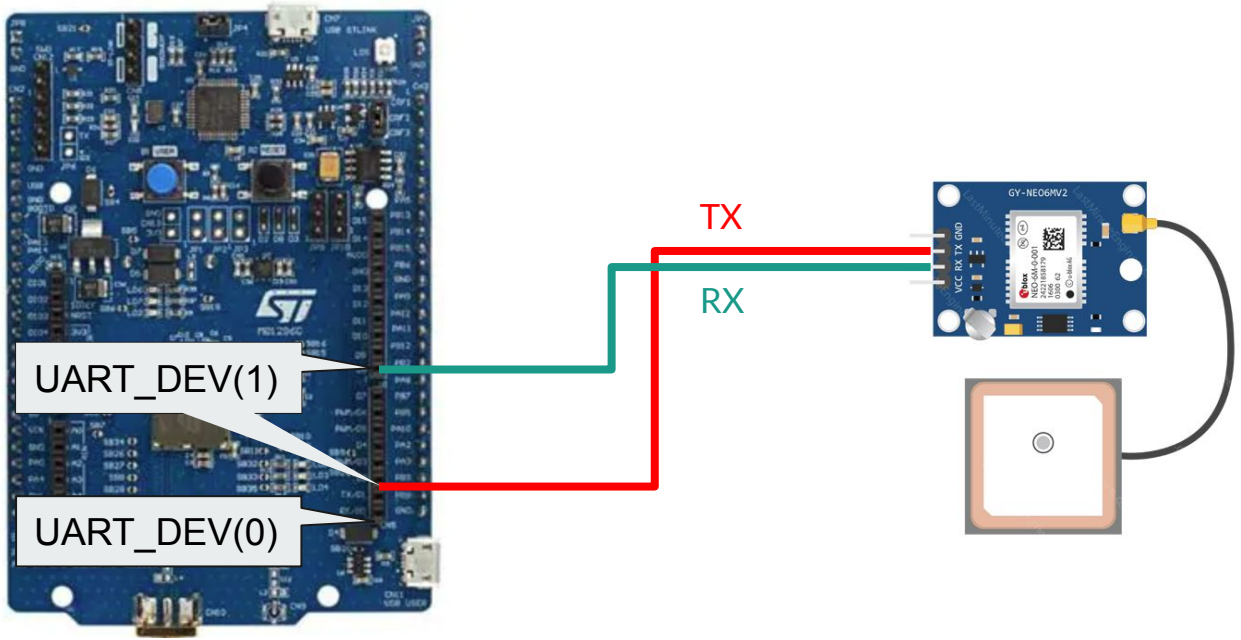
- L475e discovery with onboard BLE module

Receiver:

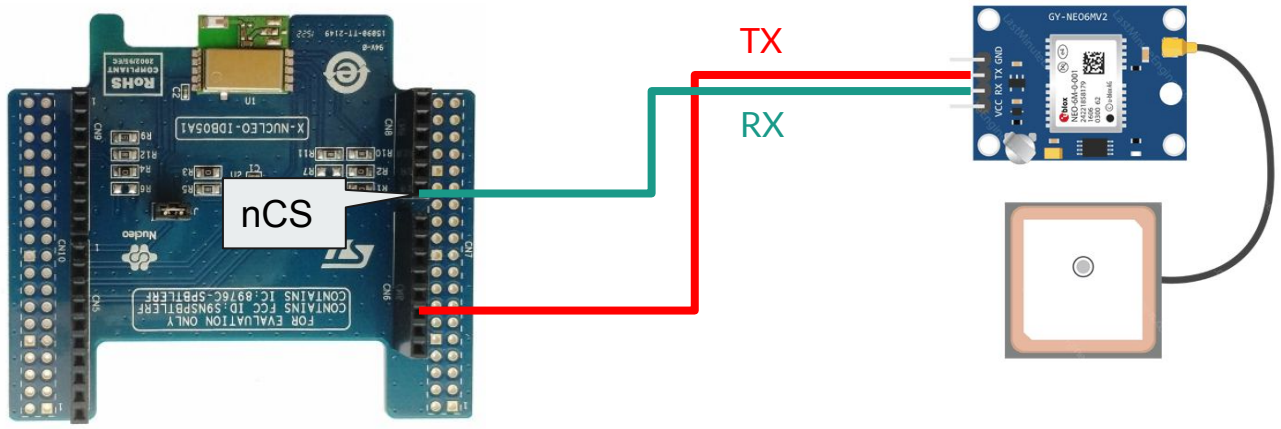
- L072 LRWAN1 board
- BLE shield
- NEO-6M GPS module



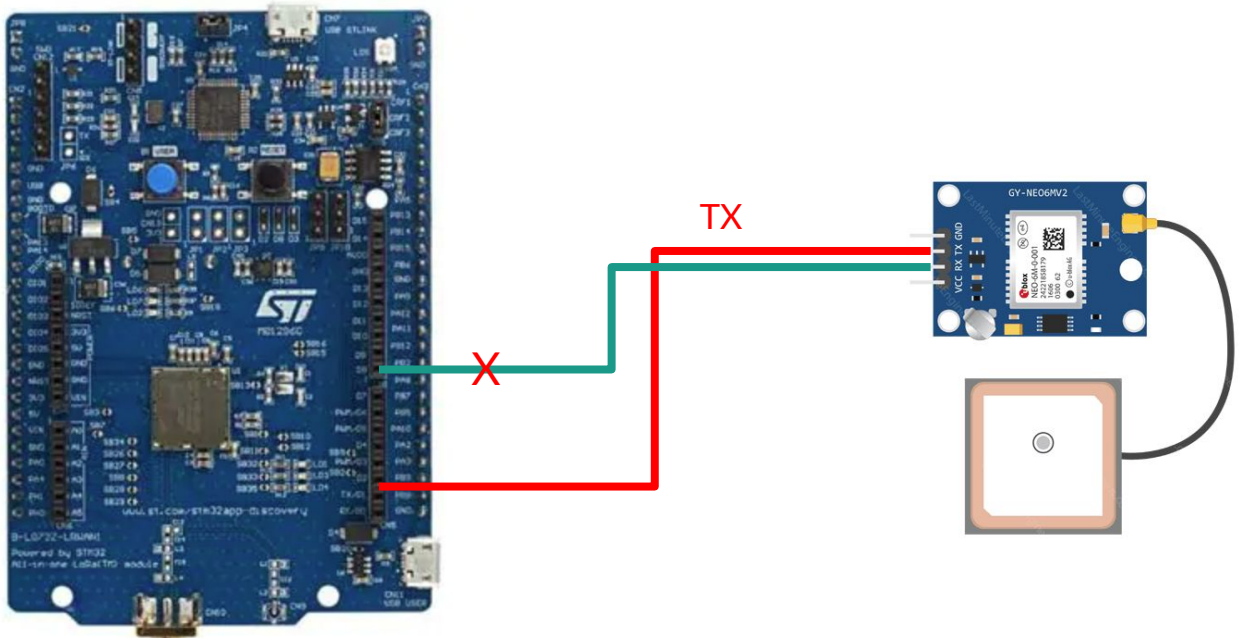
On the receivers: GPS



On the receivers: GPS



On the receivers: GPS



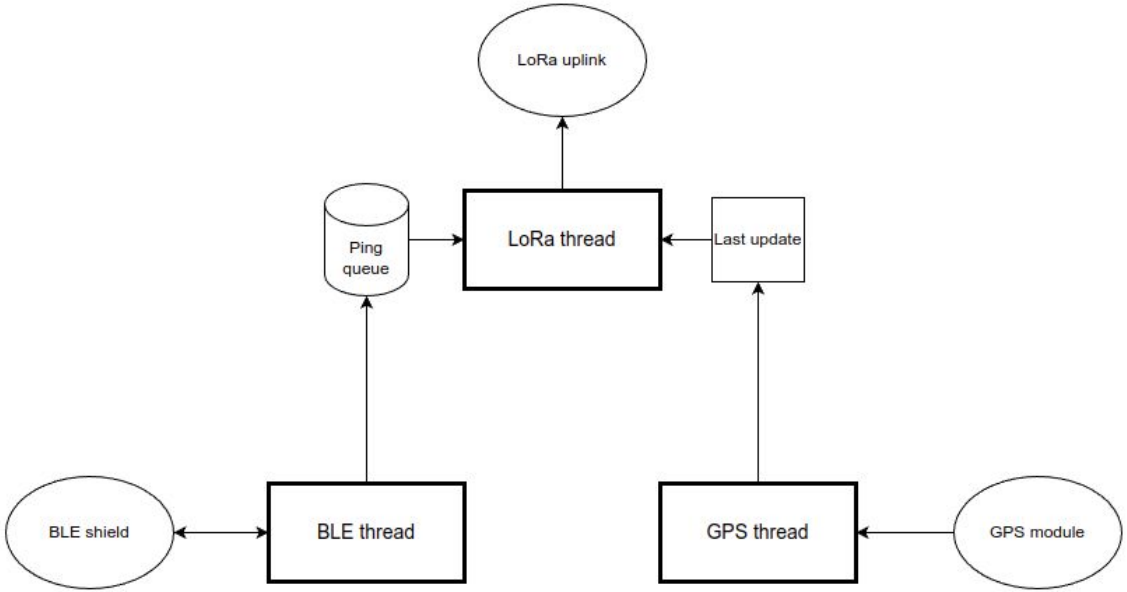
On the receivers: GPS

```
uart_init(UART_DEV(1), 9600, uart_rx_cb, NULL);
gpio_init(GPIO_PIN(PORT_A, 9), GPIO_IN);

void uart_rx_cb(void *arg, uint8_t data) {
    (void) arg;
    isrpipe_write(&isrpipe, &data, 1);
}
```

```
uint8_t* uart_read_line(uint8_t* buf, size_t
buf_size) {
    size_t read_bytes = 0;
    while (read_bytes < buf_size - 1) {
        uint8_t x;
        if (isrpipe_read(&isrpipe, &x, 1) != 1) {
            return NULL;
        }
        [...]
    }
}
```

Receiver software



Receiver software

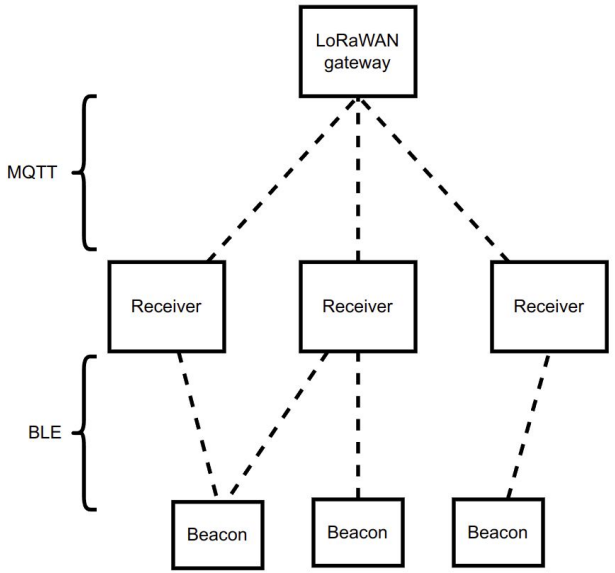
We contributed back some of the resulting code and fixes, both in PRs, and in separate libraries where full integration with RIOT OS would have been too complicated.

Bluetooth driver: SPBTLE-RF-RIOT

PRs: SPI buses C++ bugfixing

Network

- BLE pings contain identification data
  - We send beacon IDs via Eddystone
  - Once packets reach receivers, they are tagged with timestamp and location data
  - Receivers aggregate data and forward the resulting packets via LoRa
- Receivers communicate and store tagged data via MQTT



LASAGNA [evaluation]

Low power Autonomous System for Adaptive Generalised Naval Assistance

Alessio Cappellini, Dario Petrillo
May 27th, 2022

Beacon lifetime

Table 2: Average dwell times at major European container terminals (in days)

Characteristics	**Bremen**	**Hamburg**	**Rotterdam**	**Antwerp**	**La Spezia**	**Gioia Tauro**
Import dwell vessel – truck	6,4	6,4	6,4	6,4	7,4	7,4
Export dwell truck – vessel	4,6	4,6	4,6	4,6	5,6	5,6
Import dwell vessel – train	6,5	6,5	6,5	6,5	7,5	7,5
Export dwell train – vessel	4,7	4,7	4,7	4,7	5,7	5,7
Import dwell vessel – barge	4,1	4,1	4,1	4,1	5,1	5,1
Export dwell barge – vessel	4,3	4,3	4,3	4,3	5,3	5,3
Transshipment dwell	-	-	-	-	-	5,3

Beacon lifetime

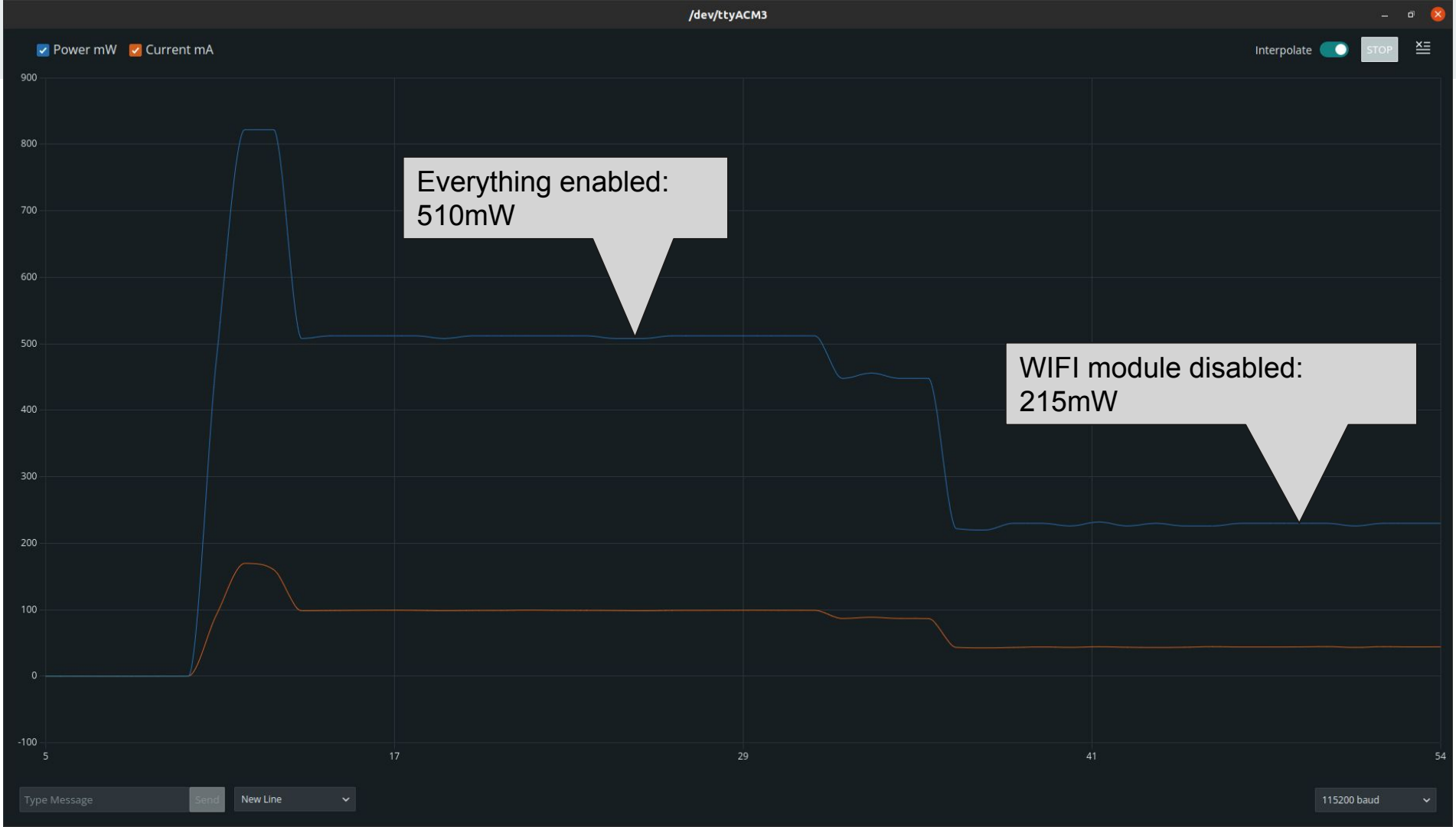
Table 4.1: Comparison of Cargo Dwell Time in Major African Ports

Type of dwell time	Durban	Mombassa	Douala	Lagos	Port said
	No. of days	No. of days	No. of days	No. of days	No. of days
Operational	2	5	5	5	1
Transactional	1	3	5	4	1
Storage	1	3	9	7	3
Total	**4**	**11**	**19**	**16**	**5**

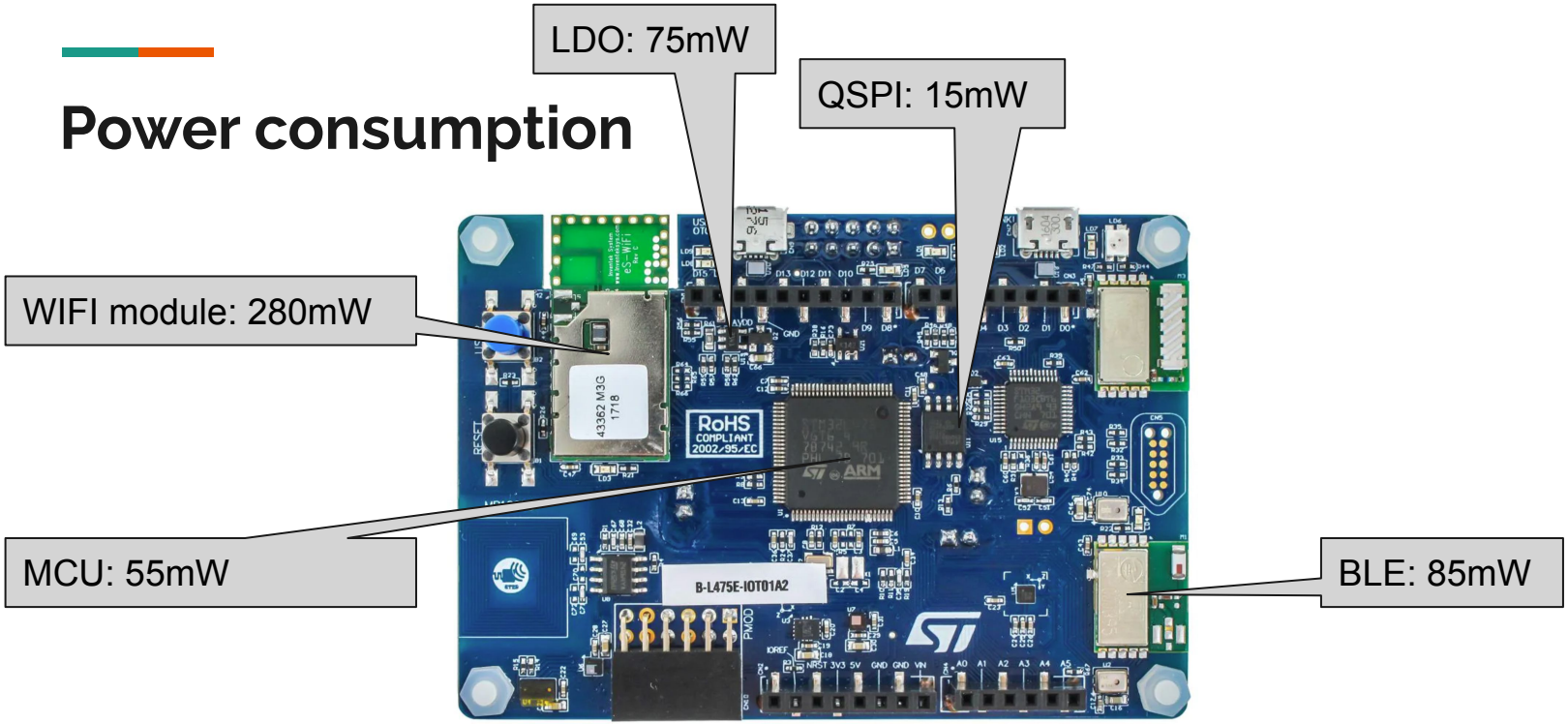
Source: Extracted from the World Bank study on cargo dwell time in African ports.

Beacon lifetime

- Given these results, we target a lifetime of 20-30 days to handle most situations
- With a typical 3xAA battery setup, we have ~7500mWh at our disposal. This means that, on average, the beacon must consume no more than 35-50mW


/dev/ttyACM3
Power mW
Current mA
Interpolate
STOP
900
800
700
600
500
400
300
200
100
0
-100
5
17
29
41
54
Everything enabled:
510mW
WIFI module disabled:
215mW
Type Message
Send
New Line
115200 baud

Power consumption



Beacon lifetime

Even with all unused peripherals disabled, the consumption is still too high: about 215mW

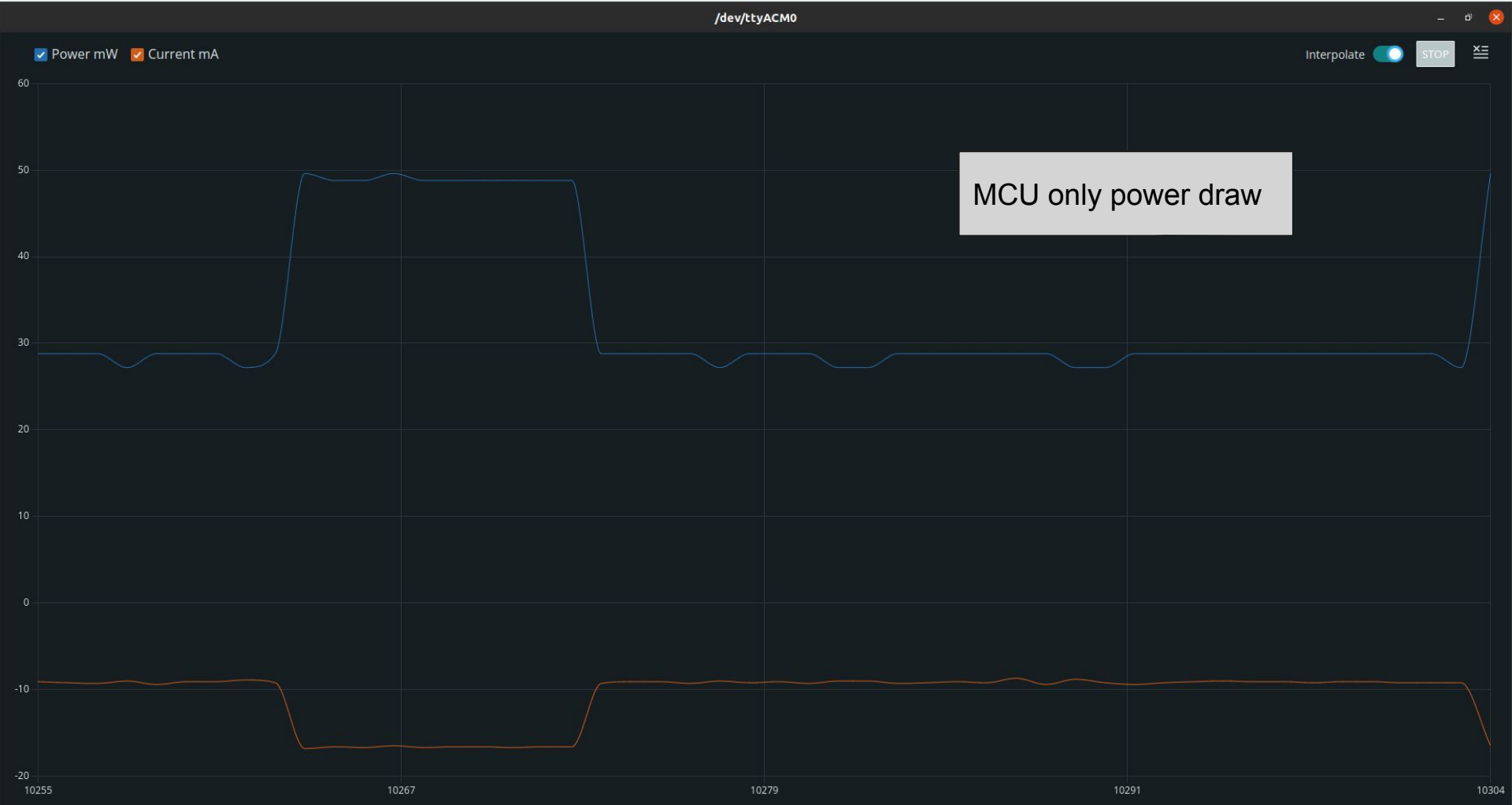
- ~75mW can be attributed to the linear regulator performing the 5v -> 3.3v conversion, and could be avoided with a better (switching) regulator
- A combined ~140mW are still required for the MCU and BLE module, and are clearly too high.

  Due to limited time constraints, we did not explore this, but it should be lowered by having both sleep when not in use

Duty cycle

To get down to a 35mW consumption, it should suffice to have a 25% duty cycle.

Unfortunately, When testing with it, we measured that we only save ~45mW in the idle periods. We did not have time to test this further, but we suspect that we are either not properly stopping the cpu, or that the BLE module continues to use a lot of power even while in reset.


/dev/ttyACM0
Power mW
Current mA
Interpolate
STOP
MCU only power draw
60
50
40
30
20
10
0
-10
-20
10255
10267
10279
10291
10304

LASAGNA [final remarks]

Low power Autonomous System for Adaptive Generalised Naval Assistance

Alessio Cappellini, Dario Petrillo
May 27th, 2022

Future work

Clearly, some additional work is required for a full system:

- A proper data visualization interface
- Data analysis for localization algorithms
- Deeper power consumption optimizations on the beacon

Additionally, the bluetooth driver currently doesn't support Peripheral and Central BLE GAP roles. These are not required for our project, which only uses Broadcaster and Observer, but would make the library more useful in future projects.

Considerations

We spent the majority of the time on writing device drivers and debugging hardware problems, as well as reading and analyzing datasheets and reference implementations.

In the end, we managed to have all the components connected and functioning, but obviously this took time away from a more thorough evaluation of the resulting system.

We believe that the results are nonetheless reasonable, especially given the time constraints, but that to reach a fully functioning state more time would be required.

Blog posts

https://www.hackster.io/lasagna/interfacing-with-spbtle-rf-on-riot-os-4664f8

https://www.hackster.io/lasagna/upgrading-the-firmware-on-a-i-nucleo-lrwan1-board-8b33cb