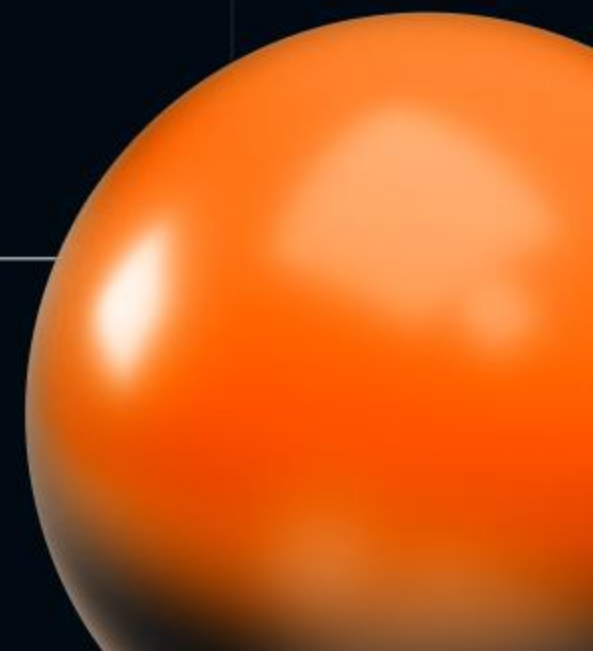
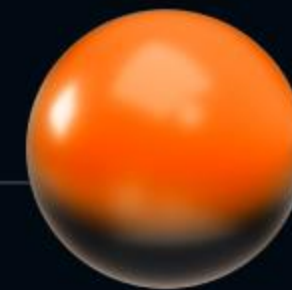


QA Revolution: Современные подходы и технологии в embedded тестировании



**Константин
Волков**

Flipper Devices



HEISENBUG





Волков Константин

- Работал в Embedded сфере разработчиком
- Писал свой асинхронный тестовый фреймворк в финтехе. React + Fastapi
- Работал как SDET в highload
- Обучаю автоматизации тестирования
- Веду тестированием в Flipper Devices и автоматизирую все что можно





Содержание:

Flipper Zero	4
Создаем свое API	7
О hardware баге	18
Кому писать тесты	38
Планы на будущее и выводы	44



Мы Flipper Devices

И мы любим создавать крутые проекты.
Мне в голову пришла гениальная идея и
наша компания как раз идеально подходит
для ее воплощения в жизнь



FLIPPER



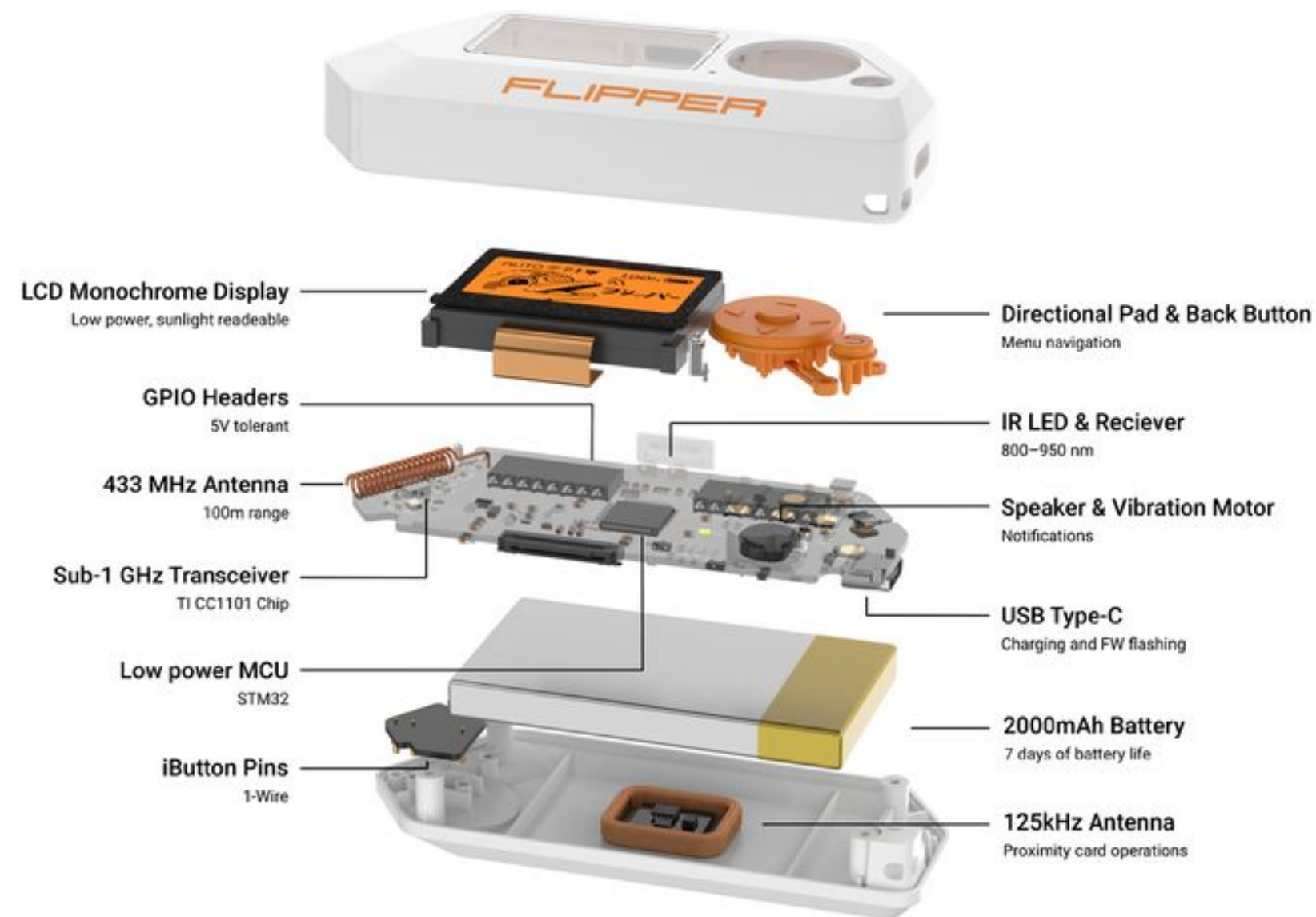
Что такое Flipper Zero

Flipper Zero — портативное устройство для пентестеров и хакеров, который с виду напоминает игрушку. Гаджет позволяет анализировать данные в радиопотоках, эмулировать ID в системах контроля доступа, перехватывать команды пультов дистанционного управления и многое другое.





Flipper Zero



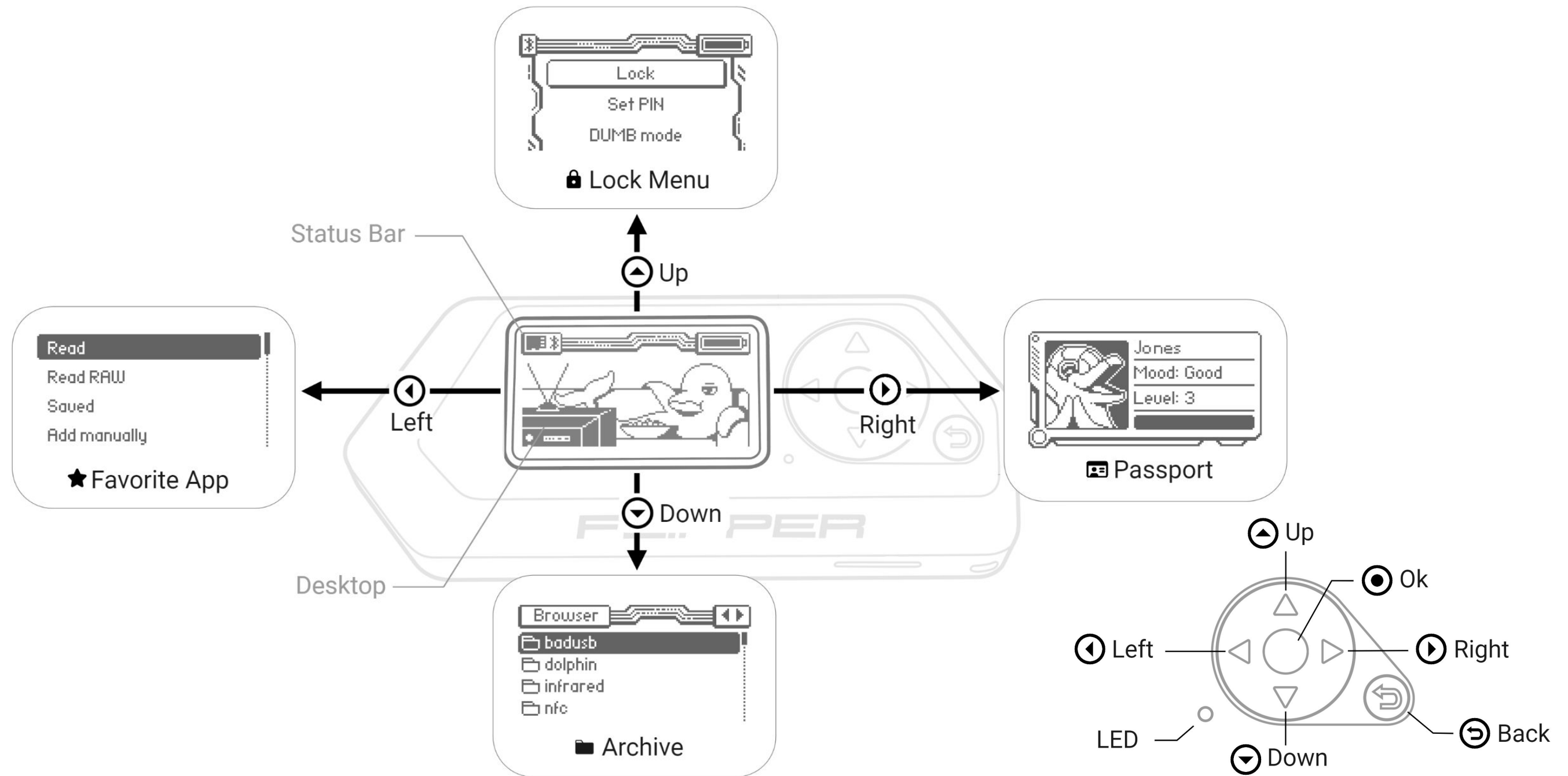
FLIPPER



Что делать если нет API?



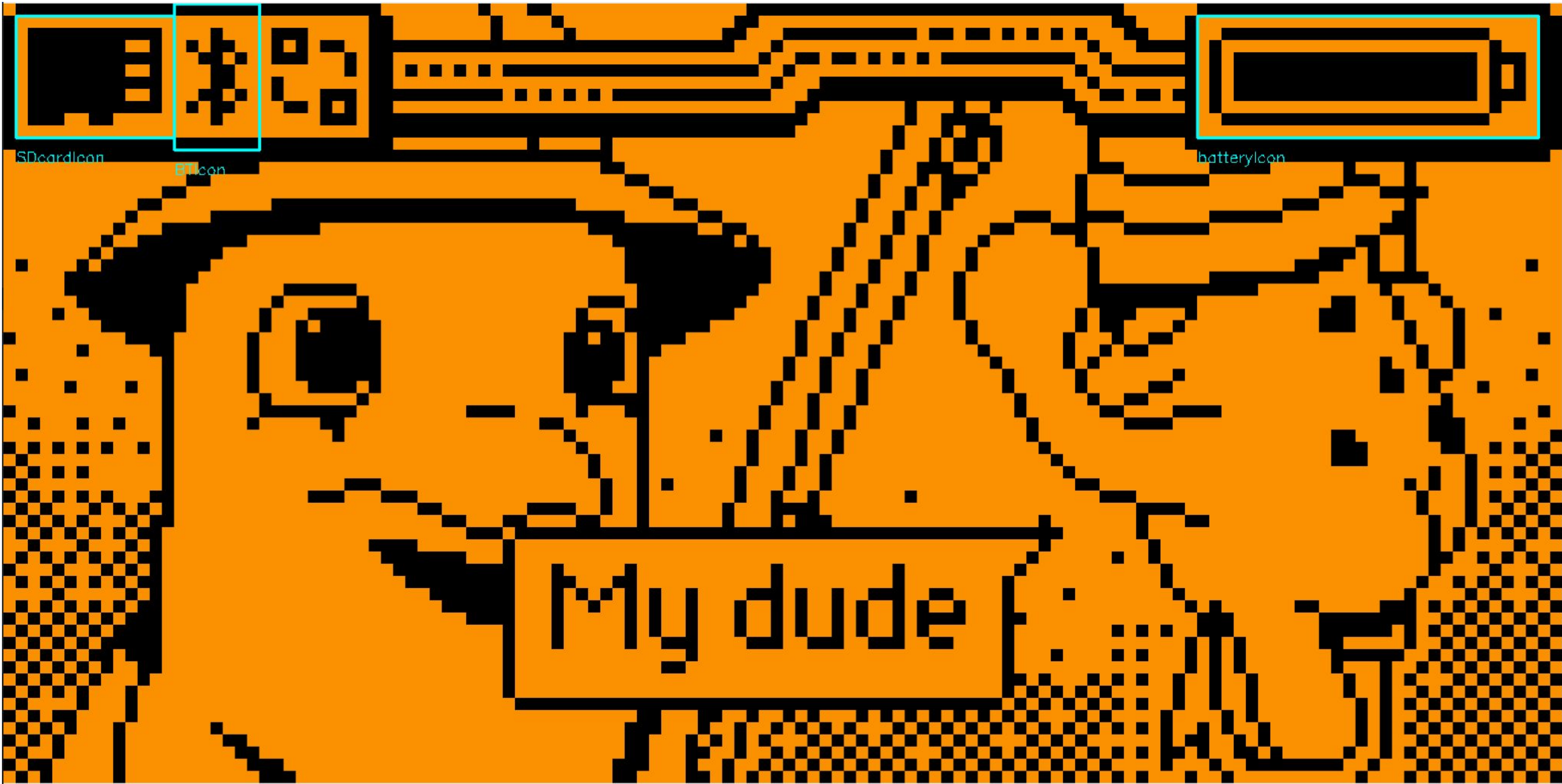
Управление





Сделаем свое API с OpenCV







И как мне писать тесты?

Pytest + allure, транспорт в виде protobuf через USB/BLE.

Фикстуры для доступа к флипперу и различным устройствам

```
@pytest.mark.smoke
def test_read(self, nav):
    nav.nfc.go_into()
    nav.go_to("Read")
    nav.press_ok()
    state = nav.get_current_state()
    assert "ReadingNFC" in state, "NFC Reading failed"
    nav.go_to_main_screen()
```


FLIPPER





Тесты со станком

```
@pytest.mark.bench
class TestNfcBench(BaseCase):
    def test_read_mifare_classic_1k_card(self, nav, gator, reader_nfc):
        with allure.step("Delete previous card, if possible"):
            nav.delete_file("NFC", "Bench_nfc_0")
        with allure.step("Go to NFC"):
            nav.nfc.go_into()
        with allure.step("Swim to NFC card"):
            gator.swim_to(-10.0, -10.0, 15000)
        with allure.step("Read a card"):
            nav.go_to("Read")
            nav.press_ok()
```

FLIPPER





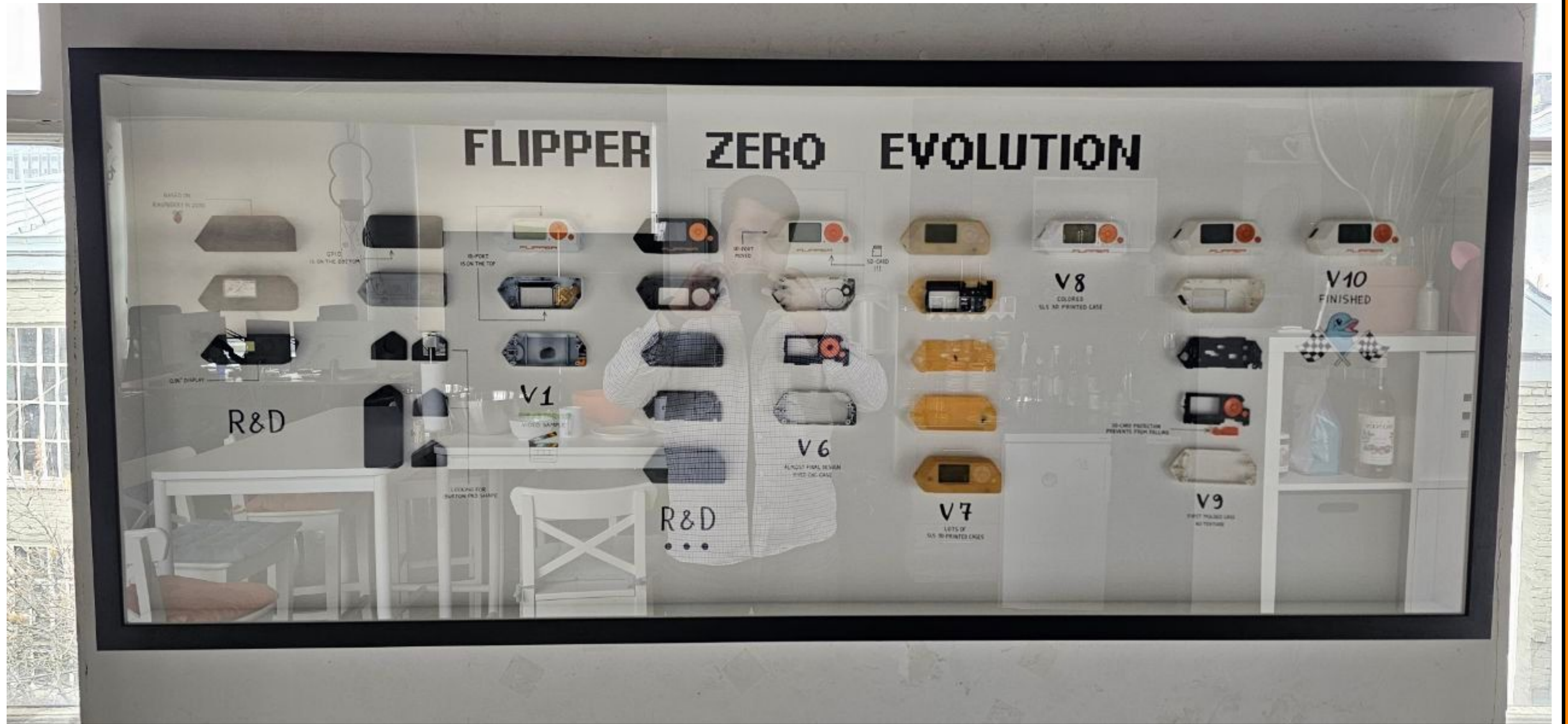
Не забываем:

**Если автотесты не
запускаются автоматически,
то у вас нет автотестов!**

*согласно опросу 2х человек в офисе



<https://heisenbug.ru/talks/21e3024ab21c4d6b92be5665b36ceb72/>



FLIPPER





Падают автотесты или история hardware бага



FLIPPER



Max Andreev 7:03 PM

<https://github.com/flipperdevices/flipperzero-firmware/actions/runs/5431866648/jobs/9878513245?pr=2824>

Упало.

`ERROR:root:Flipper not found!` (edited)



10 replies Last reply 3 months ago

FLIPPER



Увеличил таймауты

FLIPPER



Enterprise sleep

FLIPPER



Retry attempts

FLIPPER



Поиск устройства

FLIPPER

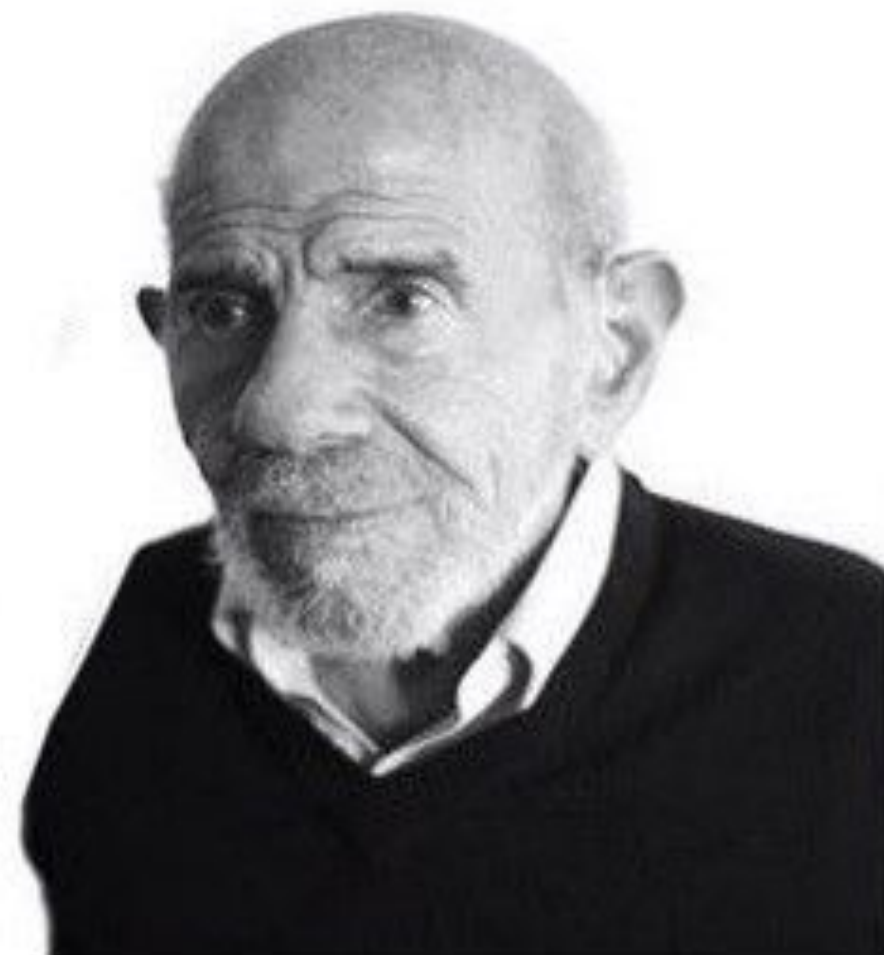


Рефактор




НЕ ПОМОГЛО

— Жак Фреско —






 **Monitoring** APP 3:55 PM


Host PROBLEM notification


Host: monitoring-01.lan.flipp.dev
IP: 172.30.1.68
State: DOWN

Additional Info :
(No output on stdout) stderr: /usr/lib/nagios/plugins/check_ping: 1: ELF: not found

 Nagios notification: Wed Sep 13 12:55:08 UTC 2023



 **monitoring-bot** APP 3:36 PM

 **Github runner: ERROR | flipper-docker-qa** 🔥

App: flipper-docker-qa
Runner: github-runner-04-flip_Teharg
Source:
Log level: 3

Device 002F00144D48500920373831 not found!

[Show more](#)





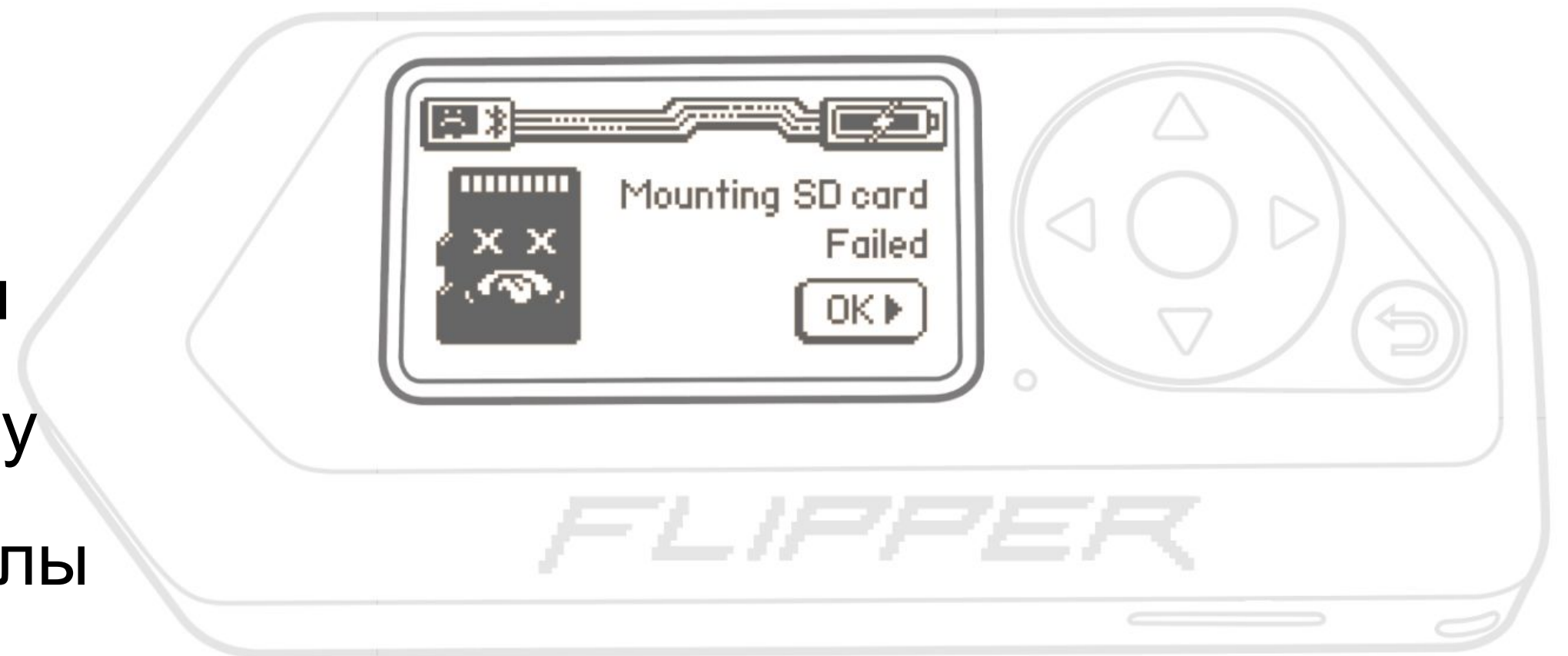
Как подготовить устройство:

Найти устройство

Прошить программатором

Отформатировать SD карту

Залить необходимые файлы





Device management

Health Check

Update Firmware

Additional
equipment

Disable device

Wipe storage

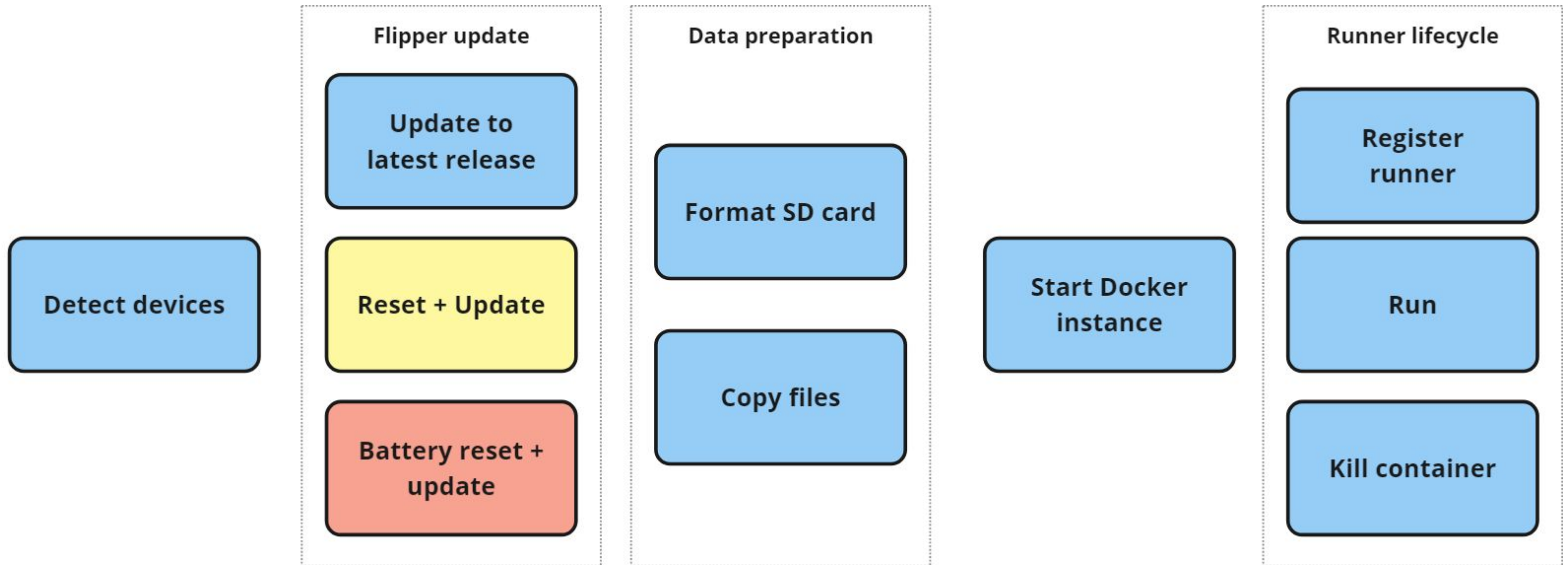
Prepare test data

FLIPPER

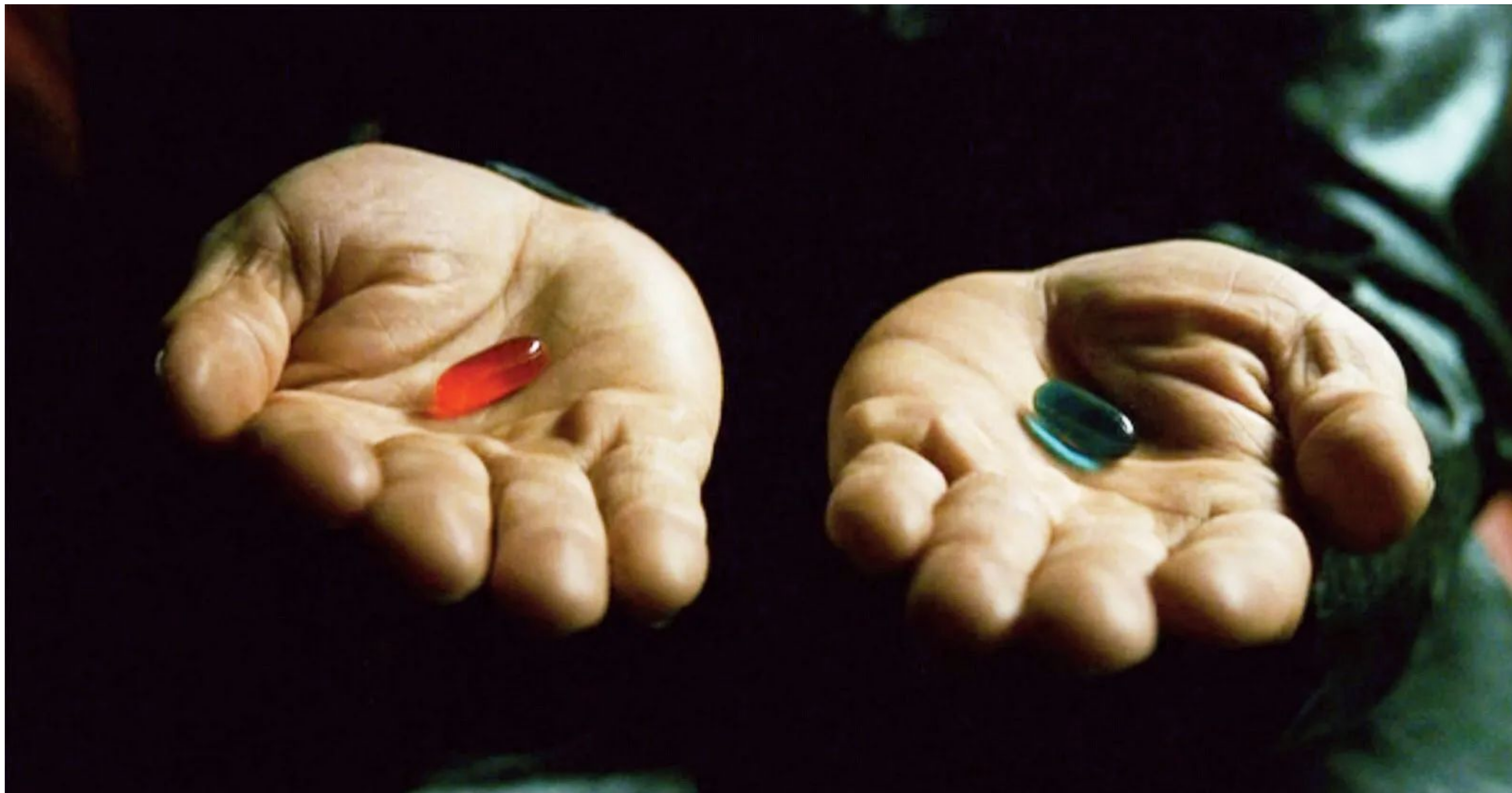


**Think outside the box, but within
the container.**

*Народная мудрость



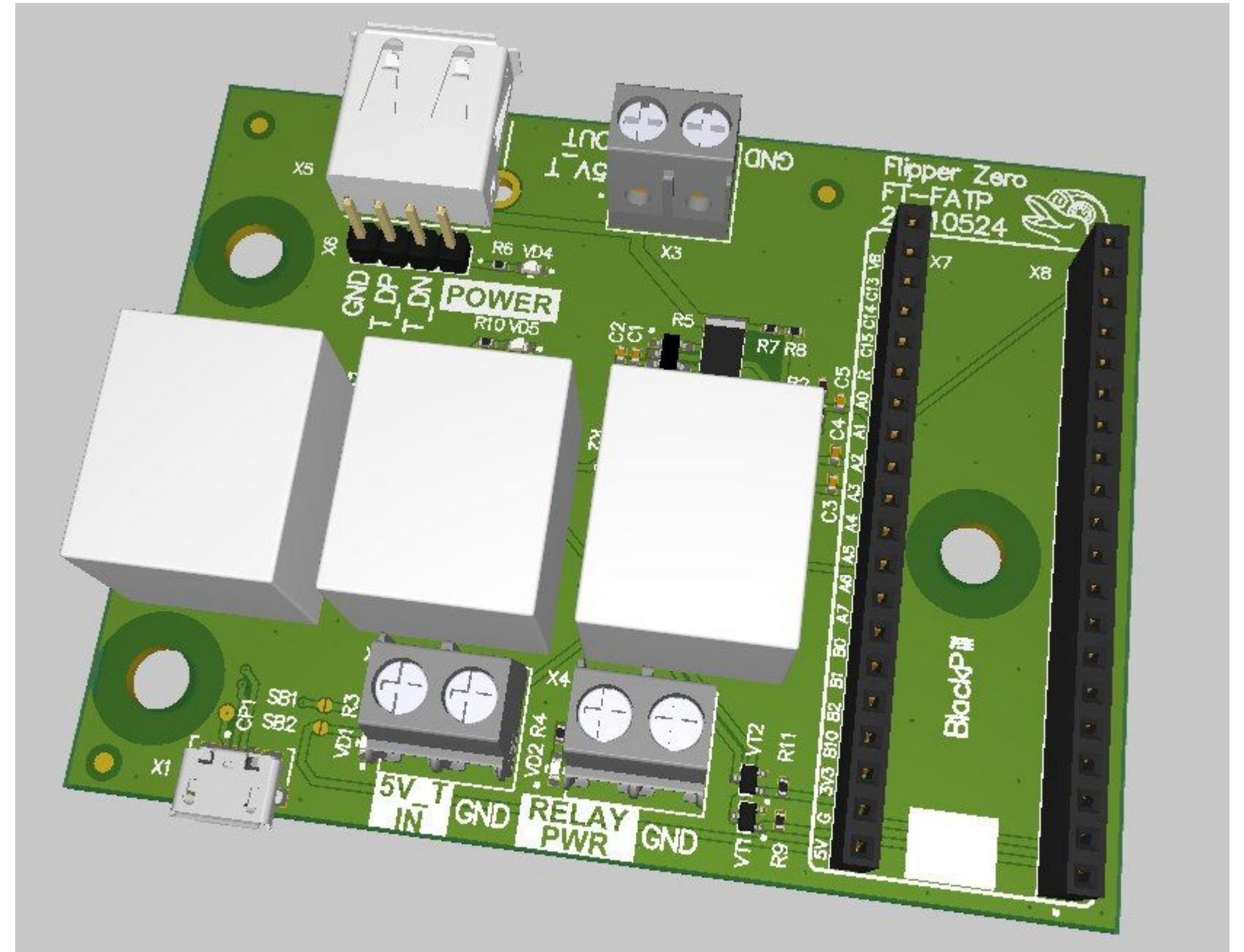
FLIPPER



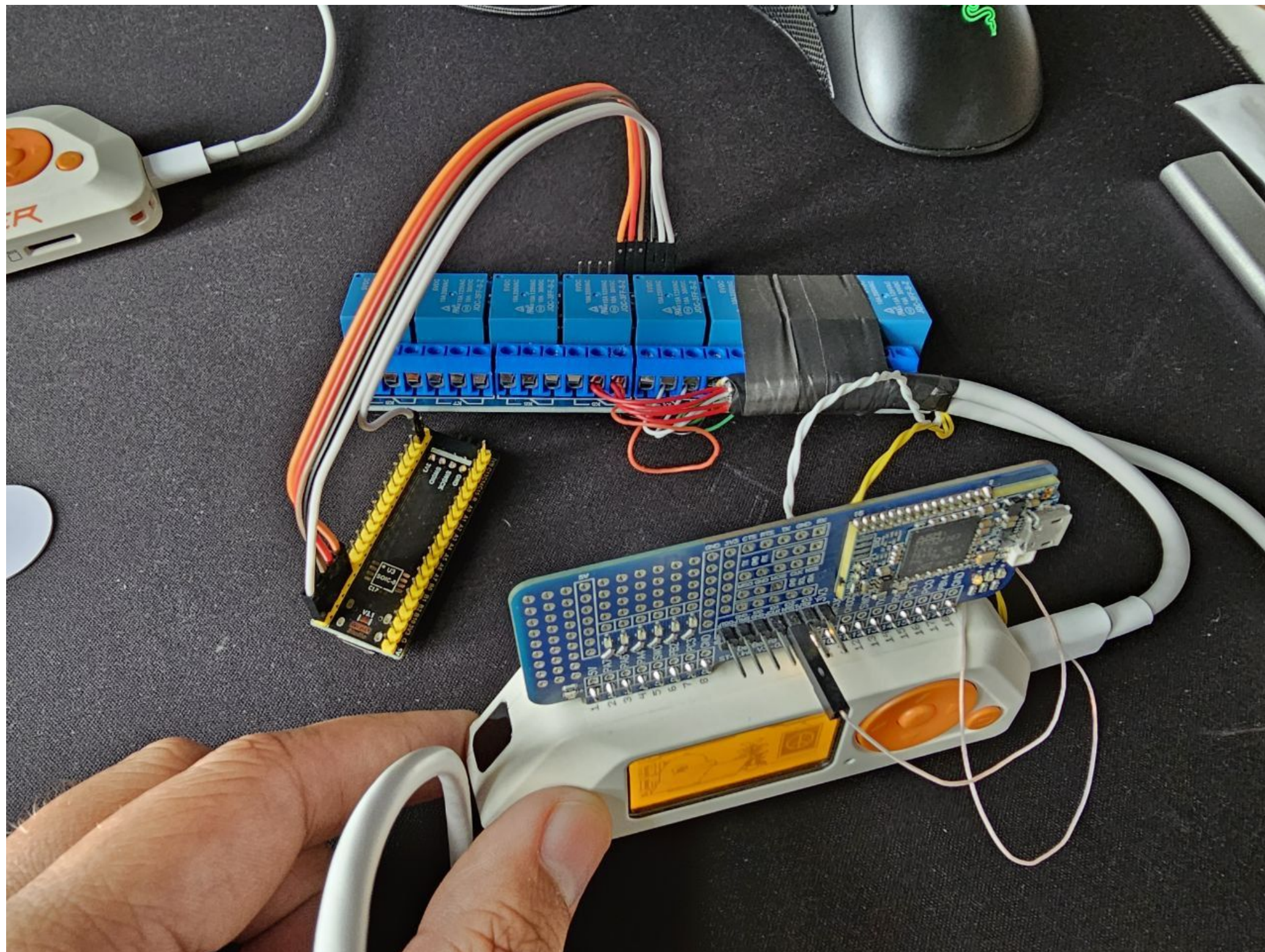


Black Pill:

- Отключает питание батареи
- Отключает питание USB
- Отключает DATA линию USB
- Физически нажимает кнопки



FLIPPER



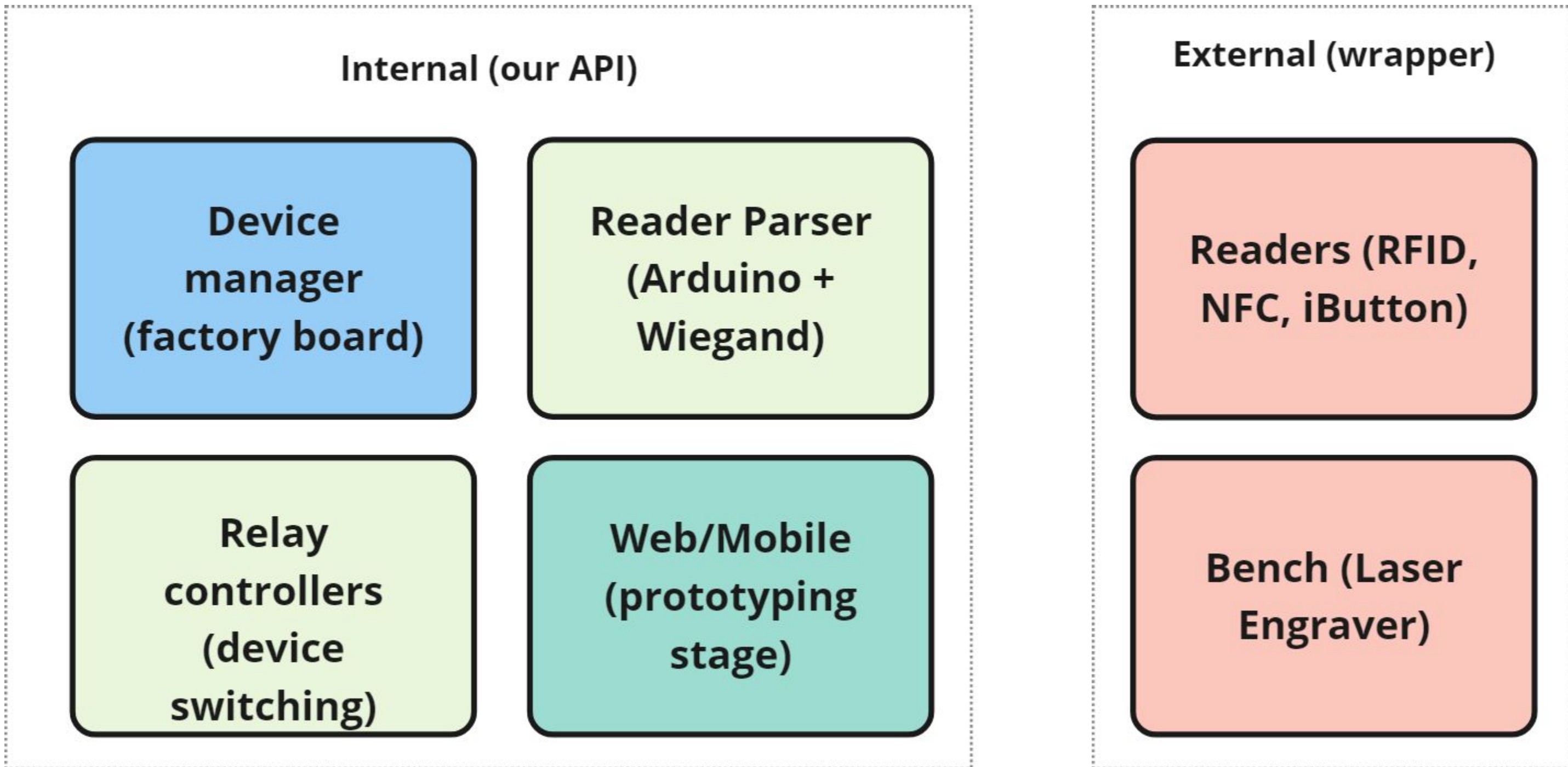


The culprit

```
[1056813.872041] usb 1-3: USB disconnect, device number 101
[1056827.893960] usb 1-3: new full-speed USB device number 102 using xhci_hcd
[1056828.045131] usb 1-3: New USB device found, idVendor=0483, idProduct=5740, bcdDevice= 1.00
[1056828.045138] usb 1-3: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[1056828.045143] usb 1-3: Product: Flipper Nyroplos
[1056828.045146] usb 1-3: Manufacturer: Flipper Devices Inc.
[1056828.045150] usb 1-3: SerialNumber: flip_Nyroplos
[1056828.047072] cdc_acm 1-3:1.0: ttyACM0: USB ACM device
[1056883.330315] usb 1-3: USB disconnect, device number 102
[1056884.100972] usb 1-3: new full-speed USB device number 103 using xhci_hcd
[1056884.255902] usb 1-3: New USB device found, idVendor=0483, idProduct=5740, bcdDevice= 1.00
[1056884.255906] usb 1-3: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[1056884.255909] usb 1-3: Product: Flipper Nyroplos
[1056884.255911] usb 1-3: Manufacturer: Flipper Devices Inc.
[1056884.255913] usb 1-3: SerialNumber: flip_Nyroplos
[1056884.257538] usb 1-3: can't set config #1, error -71
```



А какие у нас есть API/CLI/clients?





Личное мнение:

Если есть возможность что-то автоматизировать, то делайте это сразу. Это будет полезно.



А кому же писать тесты?
Или 600 тестов за одну неделю



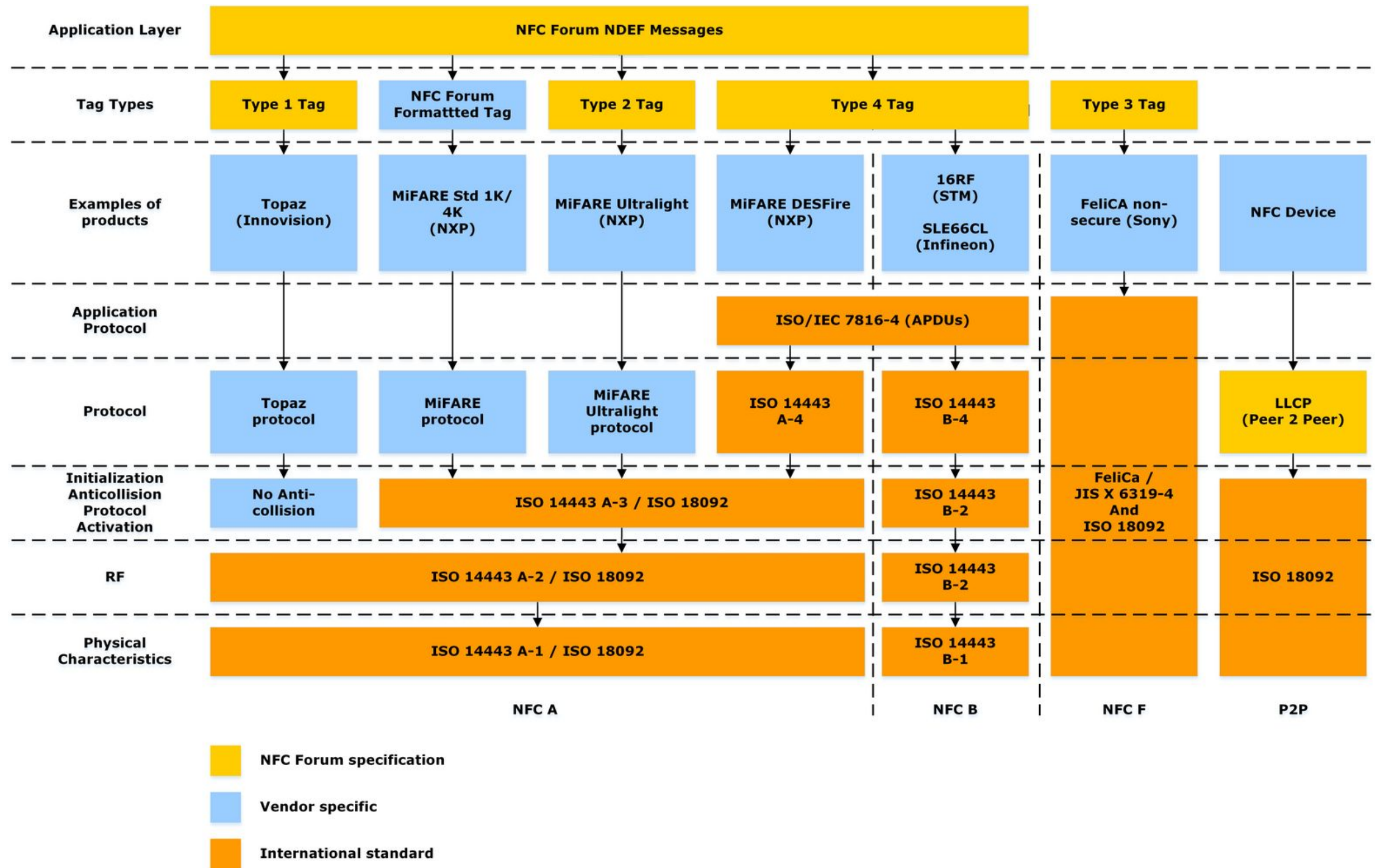
NFC refactor

+38,245 -67,884 

0 / 501 files viewed

[Review in codespace](#)

[Review changes](#) ▼



FLIPPER



А может не надо
автоматизировать?



158		Set the block to a value block	hf mf value --blk 1 -k ffffffff --set 1000	Value set success
159		Return ACs to the test state	hf mf wrbl --blk 3 -k ffffffff -d ffffffff5d278a69ffffffff	Write success
160		Increment + transfer with key A	hf mf value --blk 1 -k ffffffff --inc 200	Increment fail
161		Increment + transfer with key B	hf mf value --blk 1 -b -k ffffffff --inc 100	Increment fail
162		Decrement + transfer with key A	hf mf value --blk 1 -k ffffffff --dec 50	Decrement fail
163		Decrement + transfer with key B	hf mf value --blk 1 -b -k ffffffff --dec 350	Decrement fail
164		Restore + transfer with key A	hf mf value --blk 1 -k ffffffff --res --transfer 1	Transfer fail
165		Restore + transfer with key B	hf mf value --blk 1 -b -k ffffffff --res --transfer 1	Transfer fail
166	5D25AA	Disable the whole sector	hf mf wrbl --blk 3 -k ffffffff -d ffffffff7d25a869ffffffff	
167		Read with key A	hf mf rdbl --blk 1 -k ffffffff	Read fail
168		Read with key B	hf mf rdbl --blk 1 -b -k ffffffff	Read fail
169		Write with key A	hf mf wrbl --blk 1 -k ffffffff -d 11223344556677889900112233445566	Write fail
170		Write with key B	hf mf wrbl --blk 1 -b -k ffffffff -d 00112233445566778899123456123456	Write fail
171		Set value with key A (write)	hf mf value --blk 1 -k ffffffff --set 1000	Value set fail
172		Set value with key B (write)	hf mf value --blk 1 -b -k ffffffff --set 500	Value set fail
173		Set ACs to allow for writing	hf mf wrbl --blk 3 -k ffffffff -d ffffffff7f078869ffffffff	Write success
174		Set the block to a value block	hf mf value --blk 1 -k ffffffff --set 1000	Value set success
175		Return ACs to the test state	hf mf wrbl --blk 3 -k ffffffff -d ffffffff7d25a869ffffffff	Write success
176		Increment + transfer with key A	hf mf value --blk 1 -k ffffffff --inc 200	Increment fail
177		Increment + transfer with key B	hf mf value --blk 1 -b -k ffffffff --inc 100	Increment fail
178		Decrement + transfer with key A	hf mf value --blk 1 -k ffffffff --dec 50	Decrement fail
179		Decrement + transfer with key B	hf mf value --blk 1 -b -k ffffffff --dec 350	Decrement fail
180		Restore + transfer with key A	hf mf value --blk 1 -k ffffffff --res --transfer 1	Transfer fail
181		Restore + transfer with key B	hf mf value --blk 1 -b -k ffffffff --res --transfer 1	Transfer fail



```
(  
    "hf 14a raw -ack 3026",  
    "30 30 30 30 35 41 00 00 00 AA AA BD 84 00 26 FF [ A8 56 ]",  
) ,  
(  
    "hf 14a raw -ack 3027",  
    "35 42 00 00 00 AA AA BD 84 00 26 FF 10 05 00 00 [ 5F D8 ]",  
) ,  
("hf 14a raw -ac A22A000050000", "0A"),  
]  
  
@pytest.mark.parametrize("command, expected_result", test_data_ascii_counter_0_pwd)  
def test_ascii_counter_0_pwd(self, px, command, expected_result):  
    px_result = px.execute(command)  
  
    result = px_result[0]  
  
    assert expected_result == result
```



О чем не поговорили:

- Тест framework v2
- Тестирование новых устройств
- Метрики в hardware (не на производстве)
- Flipper + WEB / Mobile
- Стоимость поддержки



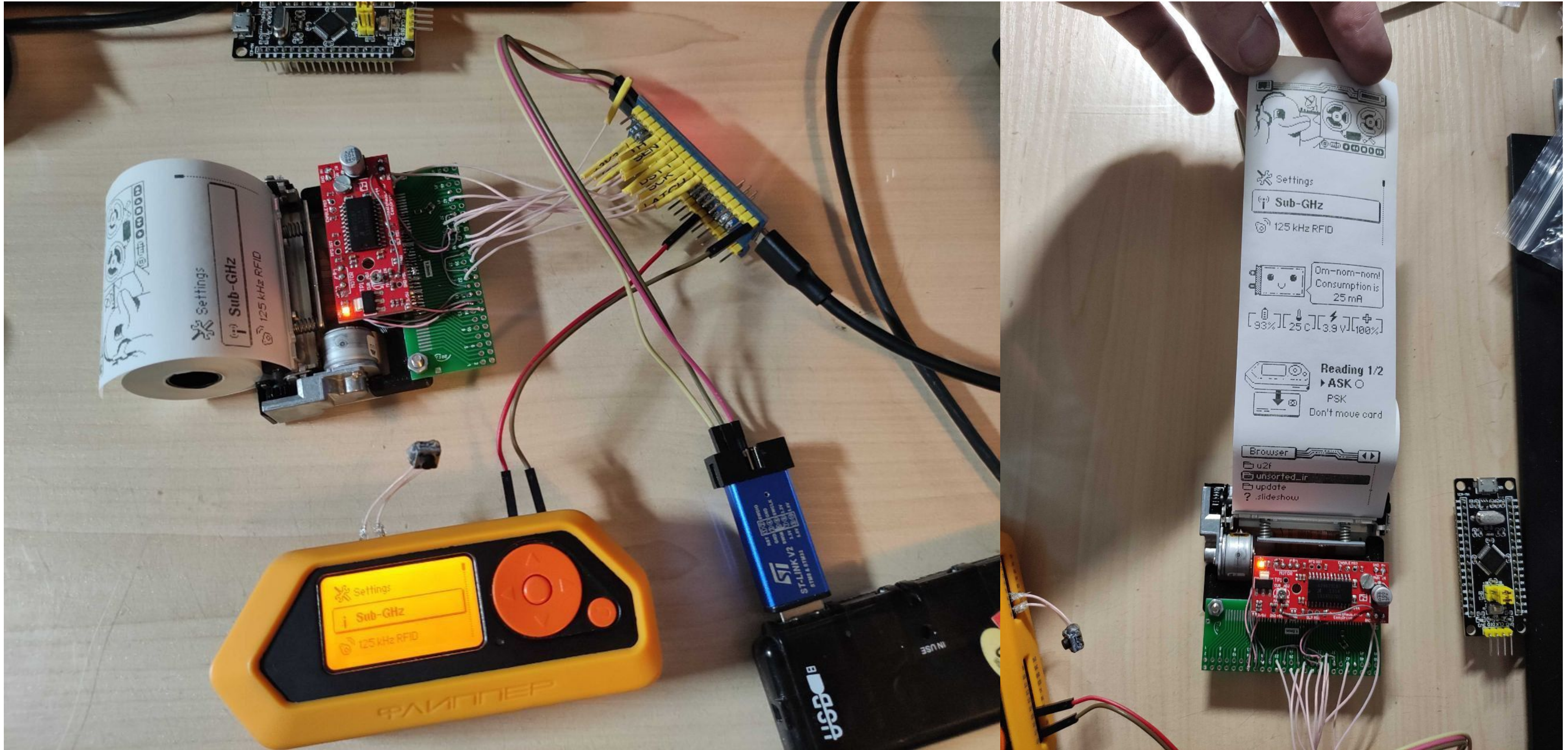
Выводы:

- Автоматизировать можно что угодно.
- Решать проблемы необходимо по мере поступления и не откладывать, техдолг убивает.
- Нужно создать ситуацию где писать тесты легко, кто их будет писать уже не важно.
- Нужно решить, а нужна ли 100% стабильность

FLIPPER



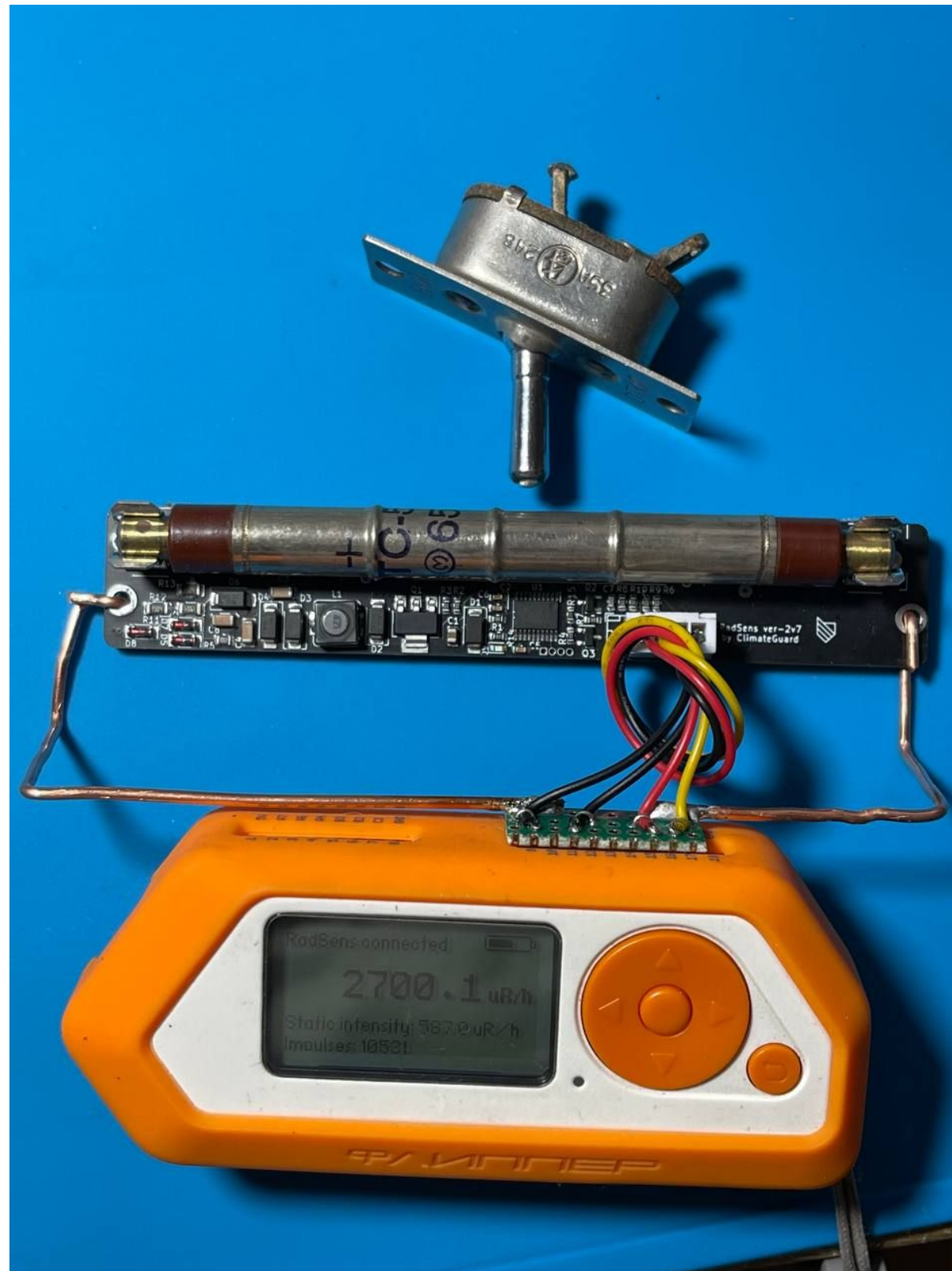
FLIPPER



FLIPPER



FLIPPER



FLIPPER





Приходите пообщаться

