Flipper Zero MISSING English Manual

translation/arrangement by baudlink

2min

Basics

Congratulations, you have become the owner of an unusual device FlipperZiro is an electronic multi-tool for passionate people who love technology, electronics, programming and explore the physical world around.

There are many functions hidden in our device that can be used in different ways, but we believe that you will not use Flipper for evil deeds.

We hope you share our freedom of information values. That's why firmware And electronic circuits Flippers are completely open, and we are waiting for you to share your developments with the community and develop the project together with us.

Flipper Zero started out as a bold idea and was born thanks to the support of tens of thousands of people onKickstarter . Nothing would have been possible without you. Thanks!

Here you will find instructions for users and developers of the device FlipperZero . If you can't find answers to your questions, ask a question Community

Basics

Sub GHz

RFID 125 kHz

Reading

Editing

Adding manually

Emulation

Bulk entry

List of RFID readers

NFC

Reading

Editing

Adding manually

Emulation

List of NFC readers

Types of NFC cards

infrared port

Reading

Editing

Emulation

Brute force by dictionary

Known IR protocols

GPIOs and Modules

WiFi Module

iButton

Reading

Editing

Adding manually

Emulation

Bulk entry

Types of iButton Keys

Development

Applications

Iron

infrared port

RFID

Firmware

Mechanics

File system

iButton

External modules

CLI Console

Community

Kickstarter
habr.com
Discord
Forum
Blog

For developers

Developer Program
Github

Partners

Neuron Hackerspace
Design Heroes
Slozhno.Media

About

Contacts
company
Careers
press kit
privacy policy
FAQ

2min

Basics

Congratulations, you have become the owner of an unusual device FlipperZiro is an electronic multi-tool for passionate people who love technology, electronics, programming and explore the physical world around.

There are many functions hidden in our device that can be used in different ways, but we believe that you will not use Flipper for evil deeds.

We hope you share our freedom of information values. That's why firmware And electronic circuits Flippers are completely open, and we are waiting for you to share your developments with the community and develop the project together with us.

Flipper Zero started out as a bold idea and was born thanks to the support of tens of thousands of people onKickstarter . Nothing would have been possible without you. Thanks!

Here you will find instructions for users and developers of the device FlipperZero . If you can't find answers to your questions, ask a question Community

Basics

Sub GHz

Permitted frequencies

Known Protocols

List of receivers

RFID 125 kHz

Reading

Editing

Adding manually

Emulation

Bulk entry

List of RFID readers

NFC

Reading

Editing

Adding manually

Emulation

List of NFC readers

Types of NFC cards

infrared port

Reading

Editing

Emulation

Brute force by dictionary

Known IR protocols

GPIOs and Modules

WiFi Module

iButton

Reading

Editing

Adding manually

Emulation

Bulk entry

Types of iButton Keys

Development

Applications

infrared port

RFID

Firmware

Mechanics

Iron

File system

iButton

External modules

CLI Console

Community

Kickstarter
habr.com
Discord
Forum
Blog

For developers

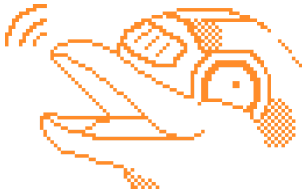
Developer Program
Github

Partners

Neuron Hackerspace
Design Heroes

About

Contacts
company



13min

First run

Switching on

The flipper in the box is in transport mode. This mode only supplies power to the microcontroller's internal clock. Read more in the section nutrition . Press the button to turn on the device

Hold down the button to turn on Flipper

If the device does not turn on

Flipper's battery may be completely discharged, then it will not be able to turn on. Plug in the USB charging cable and it will turn on automatically when the cable is connected.
If the Flipper does not turn on after the battery is charged, try resetting it by simultaneously pressing + .

Reboot

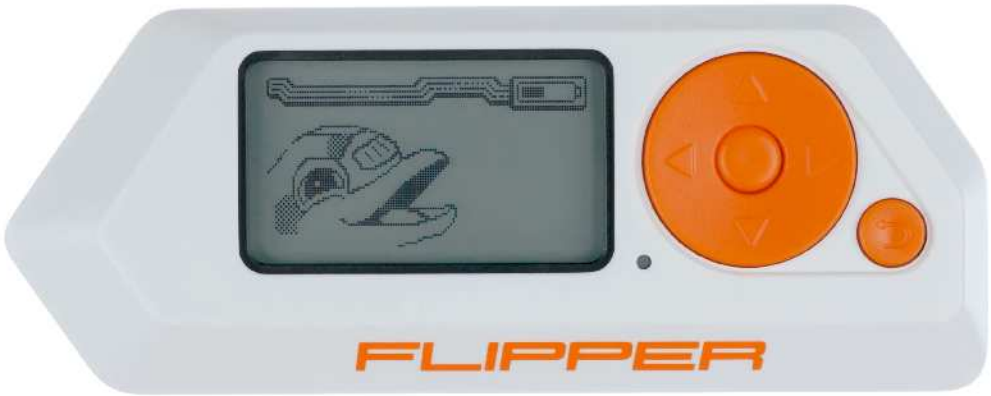
Flipper's firmware is currently in beta and is extremely unstable. It has a lot of bugs and the device may freeze during use. It's not scary, just restart Flipper if it freezes.

Clamp at the same time + to reload Flipper

> ### Reboot is different
>
> Flipper can be rebooted into different modes, including recovery mode. Read sectionreboot to find out what options there are for reloading.

Installing an SD card

Keys, cards, remotes, databases are stored on the SD card. It is also needed to update the firmware, so it is important to install the SD card BEFORE updating the firmware. The flipper supports SD cards up to 128GB, we recommend using an SD card no larger than 16GB or 32GB.



Use high quality SD cards

The flipper works with the SD card in SPI mode instead of the more popular SDIO. Therefore, it is important
Use high quality branded SD cards. Counterfeit cheap SD cards may not work well in this mode.

If the SD card does not work

After installing the SD card, a mount error message may appear on the screen of the device:

This error may occur for the following reasons:

- The memory card has a file system other than FAT32 or exFAT.
- There is no file system on the memory card.
- The memory card is damaged.

To fix the problem:

.Format the memory card according to the instructions:Format SD card . All data on the memory card will be deleted.

.On the main screen of the device, press to close the message.

.If the error message persists, replace the SD card with a new one.

SD card needed for update

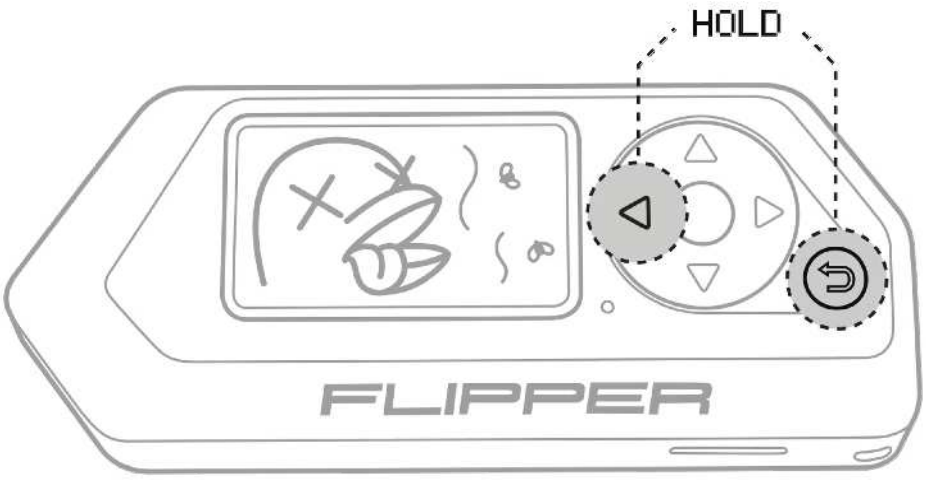
The firmware update will not work if the memory card is damaged

Firmware and database update

Flipper's firmware is in beta and is being actively developed. Every day it changes. Therefore, we recommend immediately updating the firmware and internal databases to the latest version. Read more in the section Firmware update

21min

Reboot



The Flipper firmware is updated regularly. We are working to ensure that the device works without failures. In rare cases, when working with Flipper, users encounter freezes and bugs. This is fine. For

to restore the device, please reboot.

Normal reboot

Key combination ◀ + ↩ hardware restarts the processor and performs a normal reboot. Even when the operating system is not responding, a hardware reset will still work.

Recovery Mode (DFU)

In this mode, Flipper activates the bootloader and is detected in the computer as a DFU device. When you update the firmware, Flipper goes into recovery mode on its own. To exit this mode, reboot.

> You may need to manually enable this mode when the Flipper's main firmware is broken and the device won't boot.

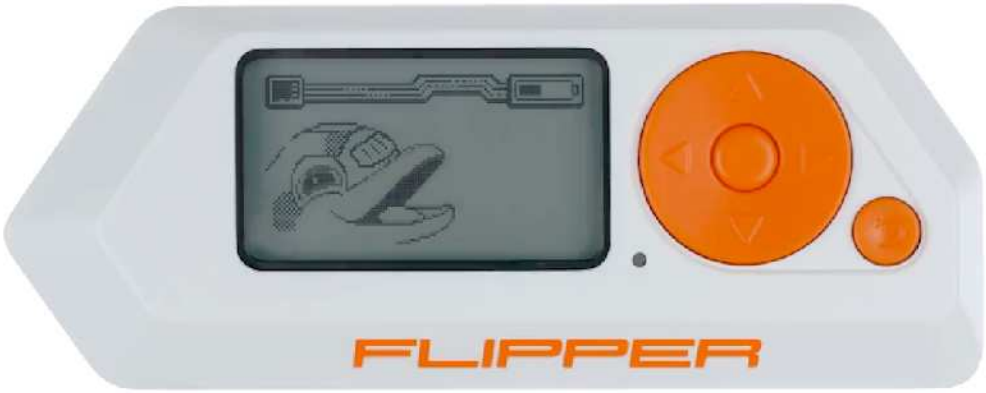
To reboot into DFU mode:

. Hold down at the same time ◀ + ↩ как during normal reboot

. let go ↩ but don't let go ◀ until the blue LED lights up

.You will see a message on the screen that DFU mode is enabled

.Restore the firmware according to the instructions:firmware recovery



Full recovery mode

If the bootloader is damaged, rebooting into update mode will not work. There will be no image on the Flipper screen and the status LED will not work. In this case, only full recovery mode will help restore the device.

To activate full recovery mode:

.Disconnect USB Flipper

.Press the button ● + ↩ and hold down for 30 seconds

.Connect the device to the computer

.Restore the firmware according to the instructions:firmware recovery

> If the bootloader and main firmware are corrupted, you will have to act blindly. There will be no image on the Flipper screen and the status LED will not light - this is normal.

You can find out how the Flipper download process is arranged in the sectiondownload steps (todo)

Reboot from the menu

You can reload Flipper from the menu, which can be useful when working through qFlipper.

To reboot from the menu:

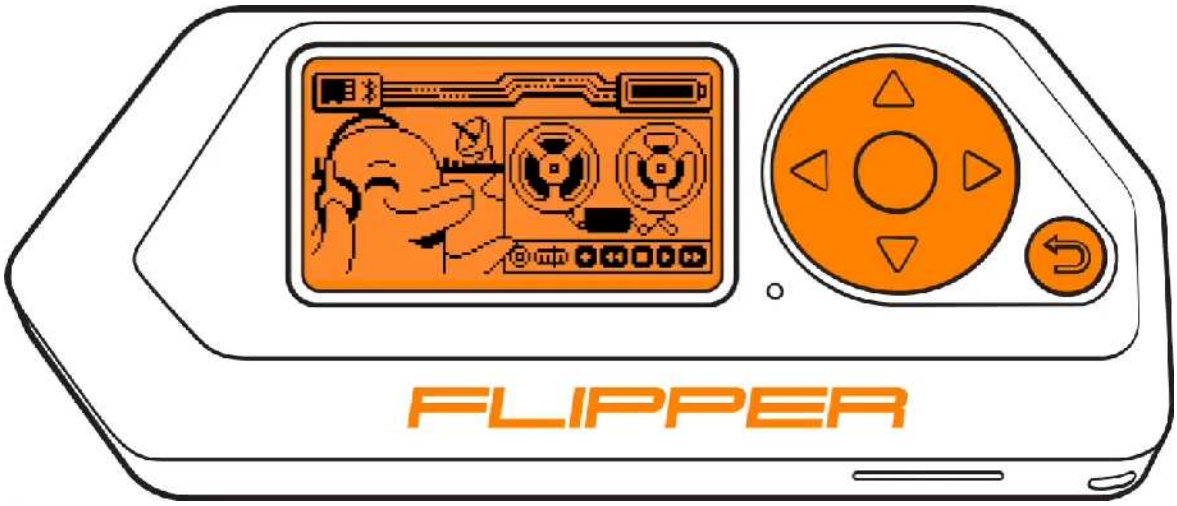
.Open menu

.Select Settings

.Go to Power section

.Select Reboot

.In Reboot type select Flipper OS



Restarting the device through the menu

Reboot from Console

Flipper can be reloaded via[Console (CLI)]

To reboot via the console:

.Connect Flipper to PC

.Open Putty

.Connect to the flipper COM port with a speed of 115200

.Enter command `reboot` and press Enter.

After the reboot, a message will appear saying that the connection was lost.

Community

Kickstarter
habr.com
Discord
Forum
Blog

For developers

Developer Program
Github

Partners

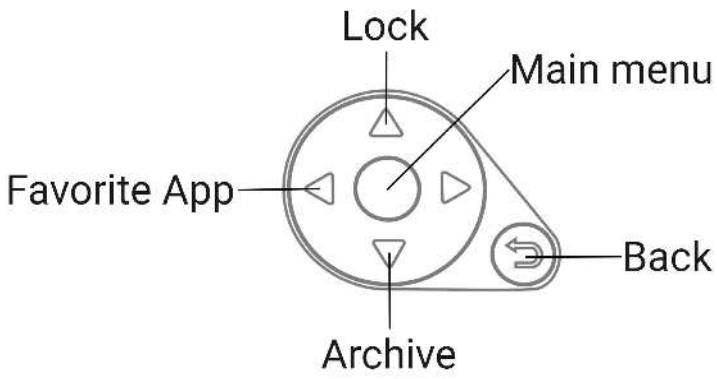
Neuron Hackerspace
Design Heroes
Slozhno.Media

About

Contacts
company
Careers
press kit
privacy policy
FAQ

5min

Control



The buttons on the main screen of the flipper do the following:

- - launchesFavorite app
- - opens the menuKeyboard blocking AndBlocking using a PIN code
- - opensarchive
- - opensMenu
- - Back

Main menu

!!! Picture of the main menu with captions

Settings

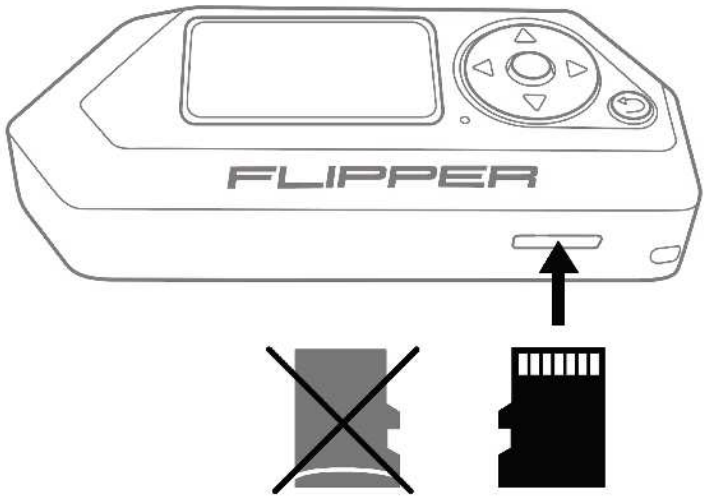
Settings menu





21min

SD card settings



Install the SD card with the contacts facing up

Keys, cards, remotes, databases are stored on the SD card. It is also needed to update the firmware, so it is important to install the SD card BEFORE updating the firmware. The flipper supports SD cards up to 128GB, we recommend using an SD card no larger than 16GB or 32GB.

The flipper works with the SD card in SPI mode (instead of the standard SDIO), so it is important to use high quality branded SD cards. Counterfeit cheap SD cards may not work well in this mode.

Formatting an SD card

Formatting the card may be required after installing a new SD card if the file system is not recognized. To do this, follow these steps:

.Open menu

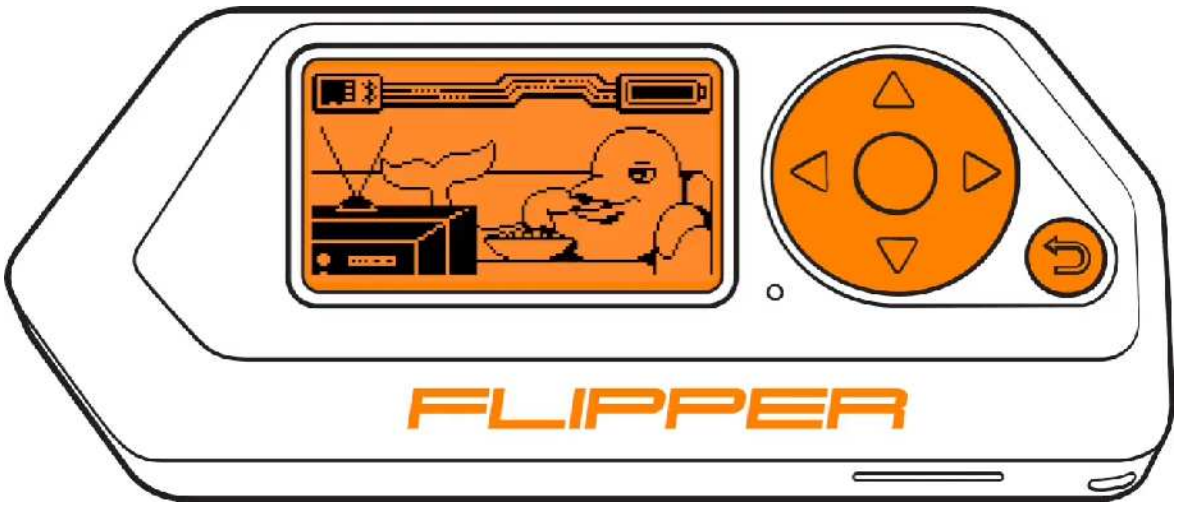
.Select Settings

.Open storage settingsStorage

.Select the format of the FormatSD Card

.The next window will warn you that all data will be deleted after formatting. Ignore it and click to continue.

.If successful, SD cardformatted will appear



Formatting an SD card

If formatting failed

When formatting, a message may appear

Cannot format SD Card. It means that formatting failed. Try formatting again on the PC, or replace the SD card.

Speed Check

To check the speed of the SD card on the flipper, follow these steps:

.Open menu

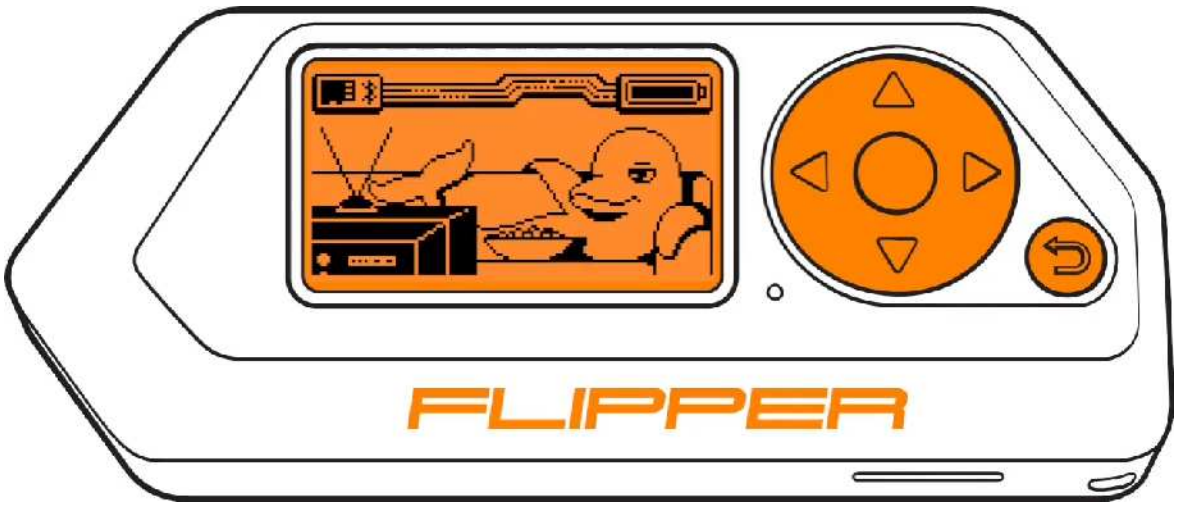
.Select Settings

.Open storage settingsStorage

.Select BenchmarkSD Card

.After the test is completed, the results will be displayed on the screen.

.To return press

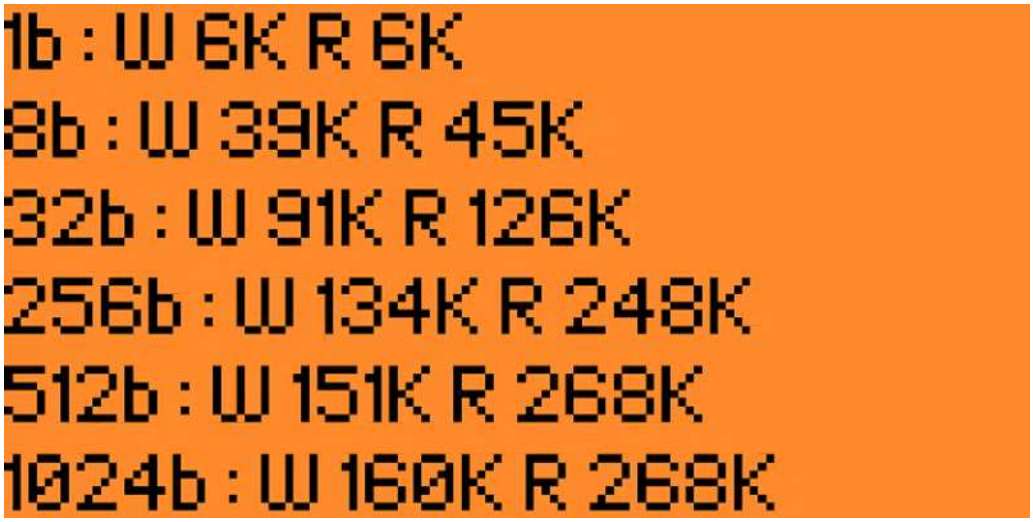


Checking the speed of the SD card

Speed Test Results

When the speed test is completed, an image will appear with

results:



Each line looks like this: `1b: W 6K R 6K` where

- `1b:` is the size of the block for which testing took place. 1b - 1 bit, 8b-8 bits and so on.
- `W6K` — write speed in kilobits.
- `R6K` — reading speed in kilobits.

Removing the SD card

Safe removal of an SD card stops all writing and reading processes, thus avoiding damage to the data on the card during the removal or replacement process.

To unmount:

.Open menu

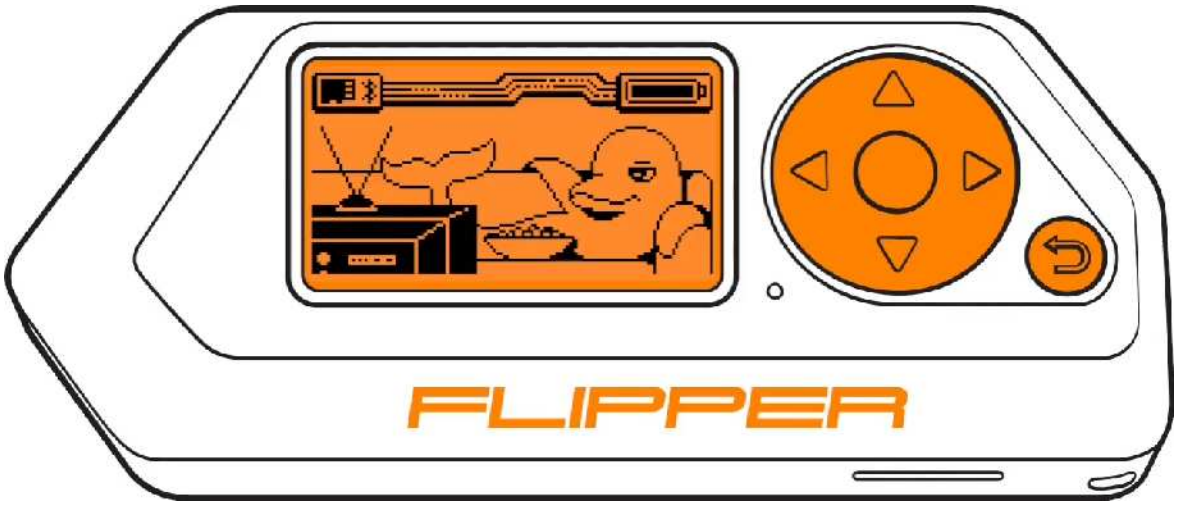
.Select Settings

.Open storage settingsStorage

.SelectUnmount

.Confirm by pressing the button to the right

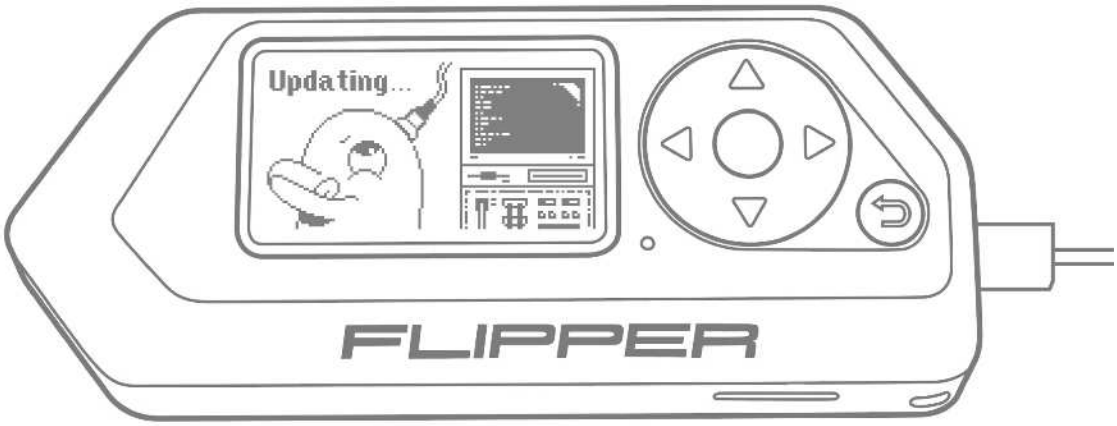
.Remove the memory card from the device



Safe removal of the SD card via the menu

13min

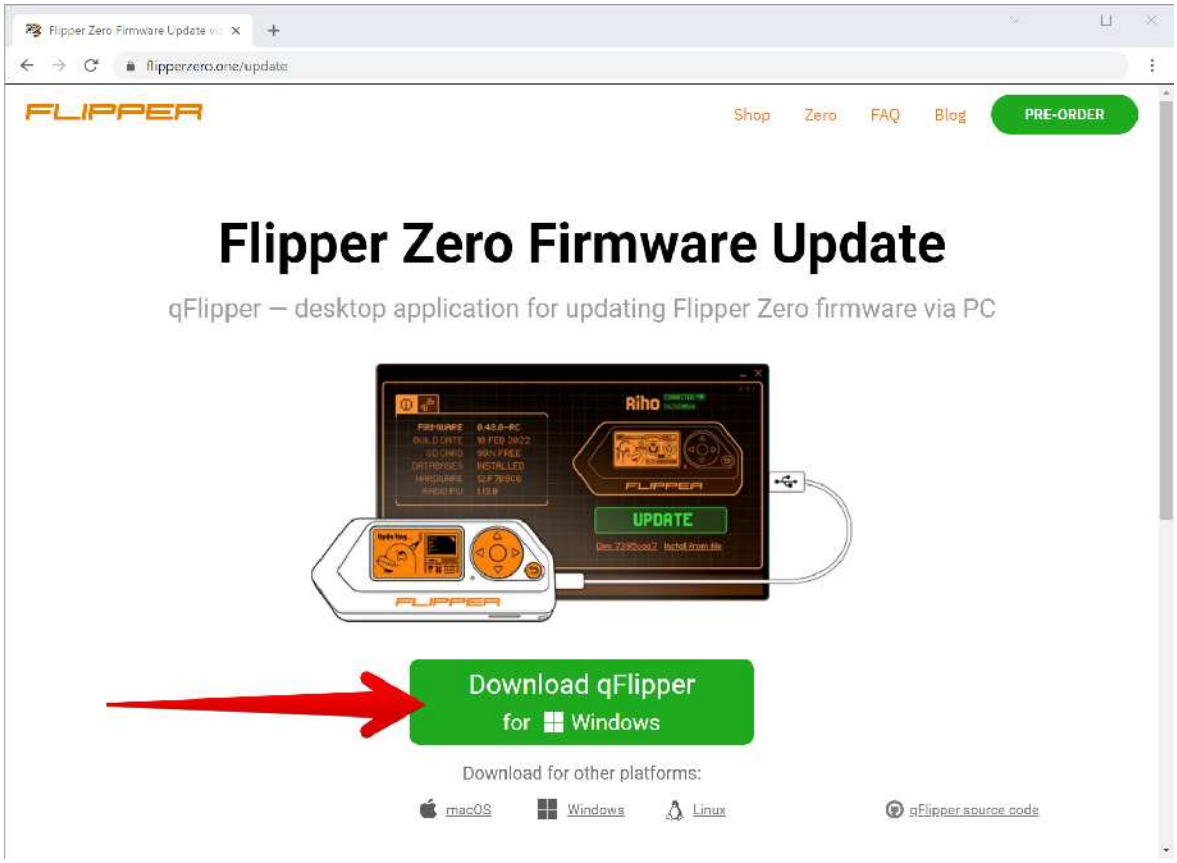
Firmware update



Firmware for Flipper Zero is in beta and is getting better every day. Therefore, it is important to constantly update the firmware on the device.

Installing qFlipper

- Download the program from the siteFlipperZeroUpdatePage .
- Select and download the installation file for your OS. (supported versions of Windows 10 and 11).
- Run the setup file and follow the instructions.



qFlipper download page

Firmware update in qFlipper

The flashing process is automatic, the qFlipper program itself creates backup copies of data from the device before updating.

Check the SD card

The Flipper must have an SD card installed in order to properly update the firmware. Part of the data

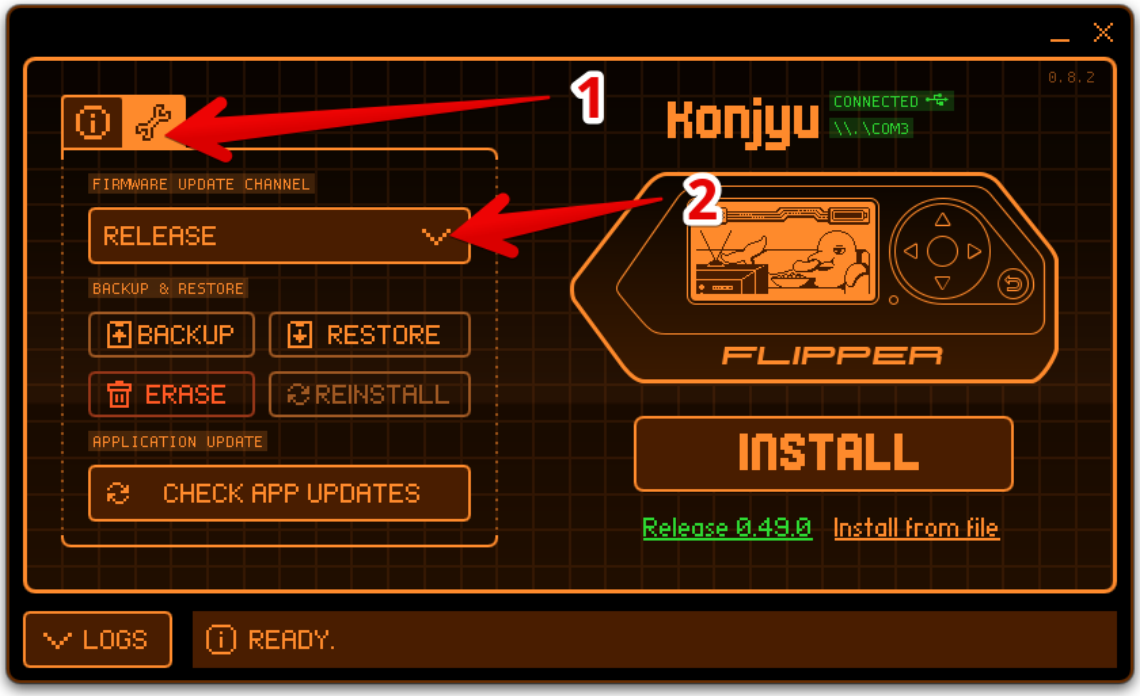
stored on it.

Firmware update channels

To select an update channel in qFlipper:

.Click on the icon

.Select the required version under FIRMWARE UPDATE CHANNEL



Selecting a firmware branch in qFlipper

qFlipper has three channels for updating the firmware:

.RELEASE- main update channel, firmware fully tested.
Recommended for installation.

.RELEASE-CANDIDATE- pre-release versions of firmware have a small number of bugs.

.DEVELOPMENT- firmware versions for developers. In them

new features are being tested, so they may contain many bugs and corrupt data on the device. Recommended for advanced users only.

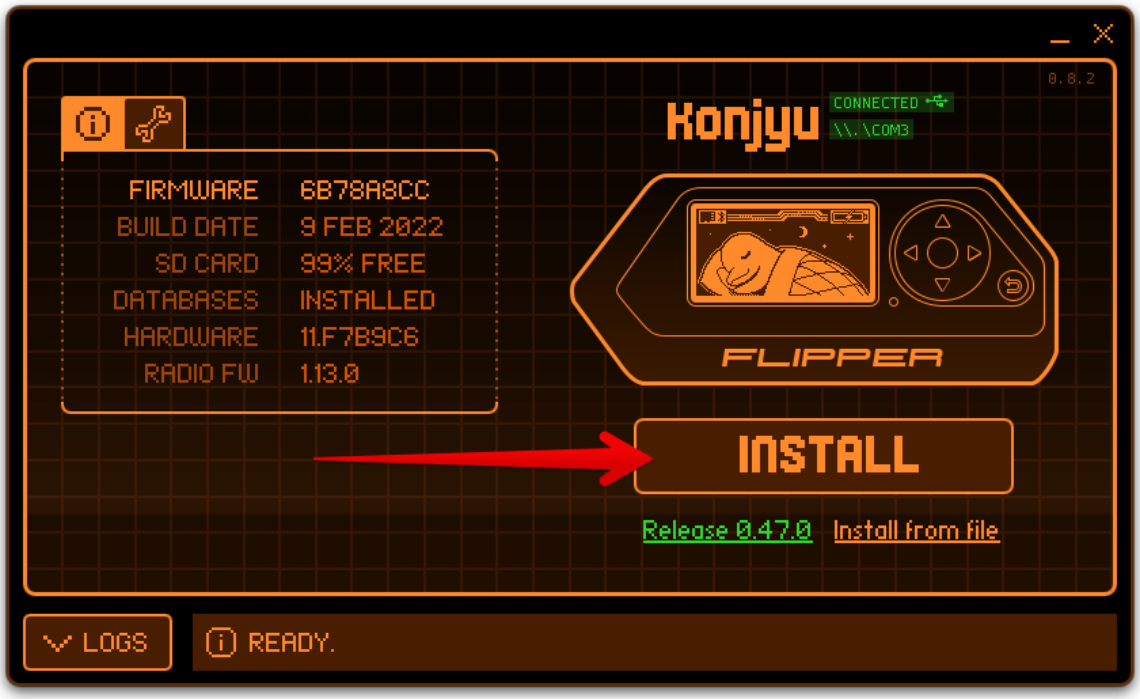
Device update

To update the firmware:

.Run qFlipper

.Connect the device to the computer via USB

.Press the INSTALL button (the button is active when there are firmware updates available).



qFlipper window

After installation is complete, click CONTINUE.

Community

Kickstarter
habr.com
Discord
Forum
Blog

For developers

Developer Program
Github

Partners

Neuron Hackerspace
Design Heroes
Slozhno.Media

About

Contacts
company
Careers
press kit
privacy policy
FAQ

7 min

Sub GHz

This is the operating range for a wide class of devices and access control systems, such as remote controls for garage doors, barriers. Most remotes operate at 315 MHz, 433.92 MHz, 868.35 MHz.

Operating frequencies vary by country

Depending on the region of delivery, Flipper's firmware supports different operating frequencies. For more information on the permitted bands in your country, please see the link below.

Permitted frequencies

The flipper receives/sends digitally modulated signals and is not designed to read, write, or play back voice data over the radio.

Software features out of the box

- **Receiving a signal at the selected frequency** - the receiving frequency is selected. Flipper waits for a signal, determines a familiar protocol and offers to save
- **Scanning signals on the main frequencies** - monitors the radio on the most popular frequencies: 315 MHz, 433.92 MHz, 868.35 MHz (if the frequency is allowed for use in the delivered country). Receives signal when available and continues scanning
- **Signal transmission** - send the stored signal at the touch of a button
- **Generation and decryption of dynamic data** - if you have created your own dynamic protocol, you can add it to the code and receive / send dynamic signals

For hardware support of operating frequencies, 3 antenna matching paths are used. Each matching path has its own characteristic, and guarantees operation at software-available frequencies.

When writing your own firmware, it is possible that the TI CC1101 chip supports the desired frequency, but it is impossible to work with the signal due to a weak receive or send signal. This is due to the frequency characteristics of the antenna, and occurs at the boundaries of the TI CC1101 bands: 300 MHz, 348 MHz, 387 MHz, 464 MHz. Later, the frequency characteristics of antennas can be found in the **Development section.**

Chip operating frequencies TI CC1101 [MHz]	Software supported frequencies [MHz]
300-348	315
387-464	433.075; 433.920; 434.775; 438.900

Transceiver TI CC1101

To work with Sub-1 GHz devices, Flipper has a built-in radio module - TI CC1101 chip. It supports both transmitting and receiving digital signals on 3 frequency bands: 300-348MHz, 387-464MHz, 779-928MHz

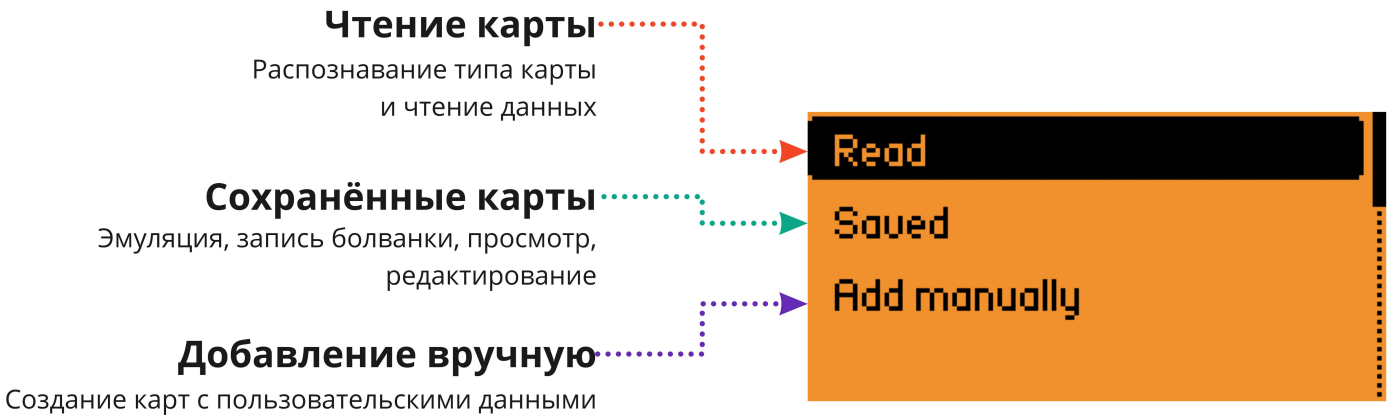
Information about the hardware capabilities of the chip can be found in the **Datasheet** .

About Omin

Permitted frequencies

3min

RFID 125 kHz



Start menu overview 125 kHz RFID

RFID 125kHz is a technology for low frequency contactless radio methods used in intercoms, office passes, pet tracking, etc. They are usually used in systems that do not require special security.

At the moment, Flipper canto read , save,Em atwhether Rovate Andwrite down on a blockheadki the following types of charts:

- EM Marin- EM4100, EM4102.
- HID - H10301.
- Motorola- Indala.

About 6min

Notable IR protocols

We invite everyone to participate in adding well-known IR protocols! With the help of the community, we want to build a comprehensive database of IR protocols that anyone can use.

At the moment, Flipper knows the following IR protocols for TVs, media centers, etc.:

- **NEC**
- **NECext**
- **RC6**
- **Samsung q32**

Protocols for air conditioners have not yet been entered into the firmware. Once entered, they will be here:**Kon ditsioners**

^1—IF=F=EF=J
FROM OOCS

Q® C? G

frequency from 30kHz to 40kHz and filling 25-35%. The exception is Bang & Olufsen with a carrier frequency of 455kHz.

Protocol name	Notes
SIRCS	Sony
NEC	NEC with 32 bits, 16 address + 8 + 8 command bits, Pioneer, JVC, Toshiba, NoName etc.
NEC16	NEC with 16 bits (incl. sync)
NEC42	NEC with 42 bit
SAMSUNG	Samsung
SAMSUNG32	Samsung32: no sync pulse at bit 16, length 32 instead of 37
SAMSUNG48	air conditioner with SAMSUNG protocol (48 bits) LG
LGAIR	air conditioner
MATSUSHITA	Matsushita
TECHNICS	Technics, similar to Matsushita, but 22 instead of 24 bits
KASEIKYO	Kaseikyo (Panasonic etc)
PANASONIC	Panasonic (Beamer), start bits similar to KASEIKYO
MITSU_HEAVY	Mitsubishi-Heavy Aircondition, similar timing as Panasonic beamer
RECS80	Philips, Thomson, Nordmende, Telefunken, Saba
RC5	Philips etc
DENON	Denon, Sharp
RC6	Philips etc
APPLE	Apple, very similar to NEC
RECS80EXT	Philips, Technisat, Thomson, Nordmende, Telefunken, Saba

NUBERT	Nubert
BANG_OLUFSEN	Bang & Olufsen
GRUNDIG	Grundig
NOKIA	Nokia
SIEMENS	Siemens, eg Gigaset
FDC	FDC keyboard
RCCAR	R.C. Car
JVC	JVC (NEC with 16 bits)
RC6A	RC6A, eg Kathrein, XBOX
NIKON	Nikon
ENWIDO	Ruwido, eg T-Home Mediareceiver
IR60	IR60 (SDA2008)
KATHREIN	Kathrein
NETBOX	Netbox keyboard (bitserial)
lego	LEGO Power Functions RC
THOMSON	Thomson
BOSE	BOSE
A1TVBOX	Al TV Box
ORTEK	ORTEK-Hama
TELEFUNKEN	Telefunken (1560)
ROOMBA	iRobot Roomba vacuum cleaner
RCMM32	Fujitsu-Siemens (Active remote control)
RCMM24	Fujitsu-Siemens (Active keyboard)
RCMM12	Fujitsu-Siemens (Active keyboard)
SPEAKER	Another loudspeaker protocol similar to Nubert
MERLIN	Merlin (Pollin 620 185)
PENTAX	Pentax camera
FAN	FAN (ventilator), very similar to NUBERT, but last bit is data bit instead of stop bit
S100	very similar to RC5, but 14 instead of 13 data bits
ACP24	Stiebel Eltron ACP24 air conditioner
VINCENT	Vincent
SAMSUNGAH	SAMSUNGAH
IRMP16	IRMP specific protocol for data transfer, eg between two microcontrollers via IR
GREE	Gree climate
RCII	RC II Infra Red Remote Control Protocol for FM8
METZ	METZ
ONKYO	Like NEC but with 16 address + 16 command bits

About 5min

receiver list B

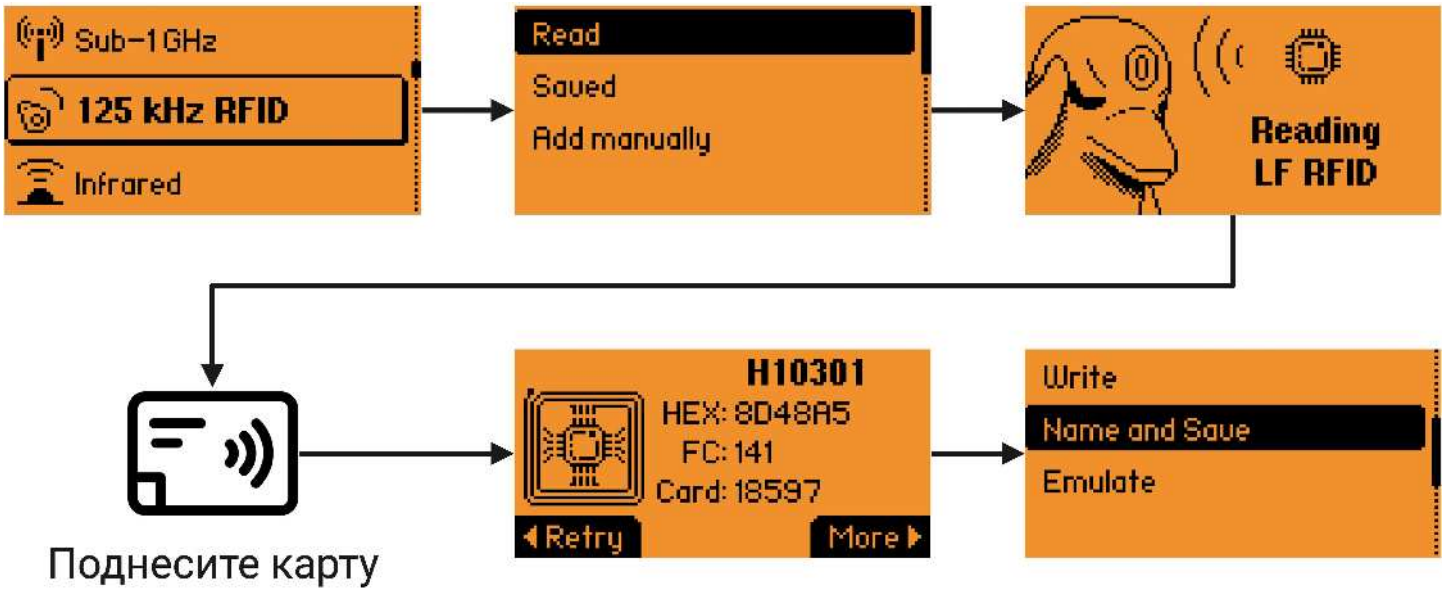
Name	Remote controller	Description
 Name:**RX Multi 433/868** Work with flipper and with different consoles.	Support all popular protocols.	Universal receiver RX-MULTI. Has 2 channels 433-EOOmHz.**(inst Raction)**
 Support all popular protocols. Gant,		The universal two-channel receiver allows you to record remote controls of the following brands: BFT, An-Motors, CAME, Doorhan, and others on 433 MHz.**(inst Rction)**

htu z Works with flipper and with different consoles.		
 Name:**Nice Flox2** Verified since Nice remote control FLO and Flipper. (Signal sent from flipper - Nice FLO 12 Bit Key:0x00000555)	 Name: **Nice FLO2**	External single channel The NICE FLOX1 receiver (433.920 MHz) is designed to work with NICE FLO and VE series key fobs operating at a frequency of 433 MHz with a static code. The receiver has one channel with relay exit,[inst Raction)
 Llnonni IUA' **Came**	 Name: **sate TOP44RGR**	Two-channel universal external receiver CAME RE432M (433.92 MHz) with memory for up to 50 remote controls management. Compatible with Top, Tam, Twin and Atomo series CAME key fobs with

flipper not maybe for now read signal from remote control.		
 Name: **GATE-RX** Accepts signal from flipper. (Signal sent from flipper - Gate tx 24bit 02ED0D)	 Name: **GATE-TX**	GATE-RX is a standalone device designed for remote reading of GATE-TX key fobs. Operating frequency: 433.92MHz. **{inst Raction)**
 Name: **Doorhan DHRE-one** Accepts signal from	 Name: **doorhan transmitter 4**	Single channel external Radio receiver Doorhan DHRE-1 (433.92 MHz) with a dynamic code. The receiver is equipped with a non-volatile memory chip for storing 1000 key fob codes., **{inst Raction)**

1min

Reading

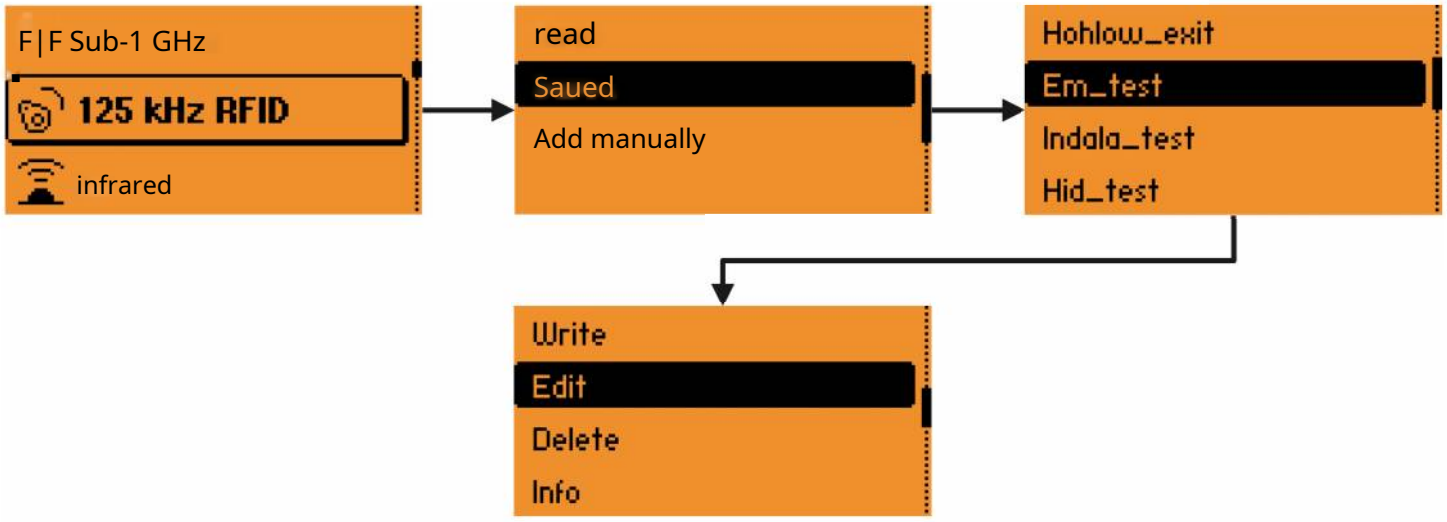


Card reading menu and available operations after reading

To read the low-frequency card, you must go to `125 kHz RFID —> Read` and attach a label to the back side. Flipper define protocols and display its name and chart ID. In one pass, Flipper reads in turn all known types of protocols that differ in modulation. Therefore, reading can last a few seconds.

About 9min

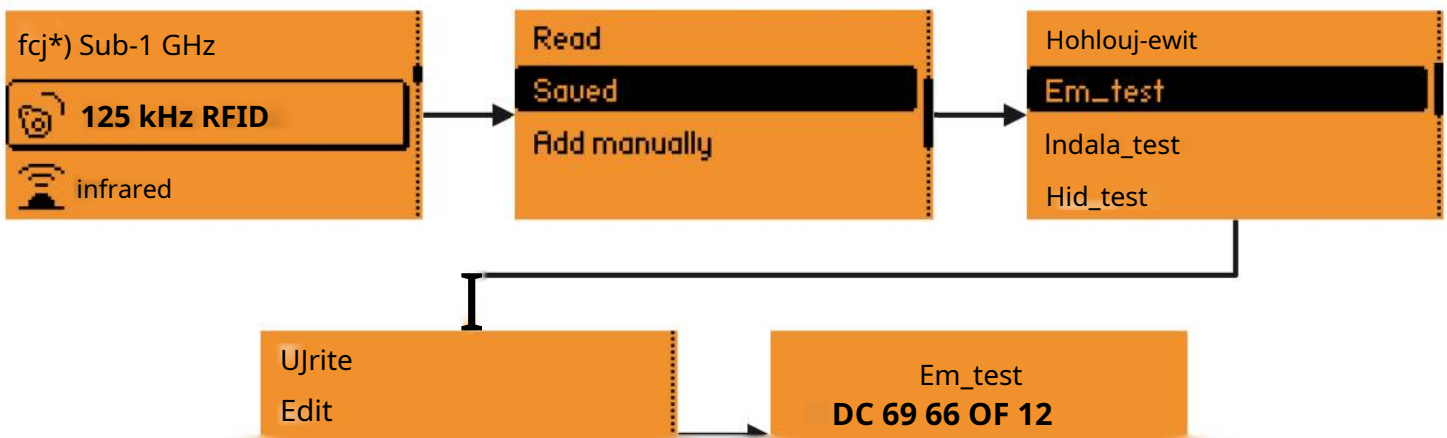
Editing



Edit Saved Label Menu

To change the unique number or rename the saved label, go to 125 kHz RFID -> Saved -> [CARD NAME] -> Edit. You can delete a card from the menu125 kHz RFID -> Saved -> [CARD NAME] -> Delete.

Map Information



Detailed information about the map is available in the menu `125 kHz RFID -> Saved -> [CARD NAME] -> Info`

This screen displays:

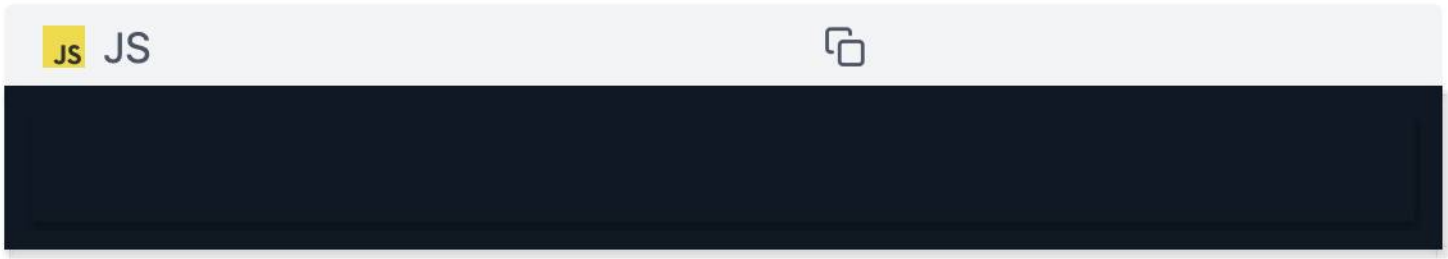
- Name
- Unique code in hexadecimal format
- Unique code in text format
- Card type

Editing a map using a PC

LF RFID cards are saved on the SD card in the directory `/lfrfid`. Using a PC, you can view or change the data of the saved maps.

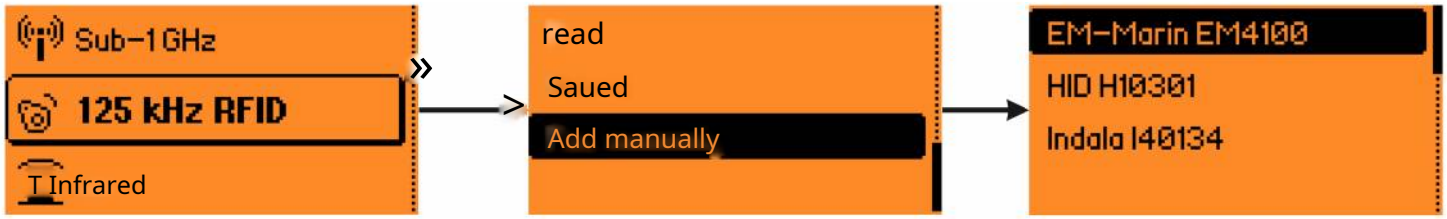
The card storage format contains:

- **Transfer protocol -**EAT4100, H10301,140134
- **Data -**5 bytes for EM-Magip, and 3 bytes each for H10301 and 140134



Oh imin

Adding manually

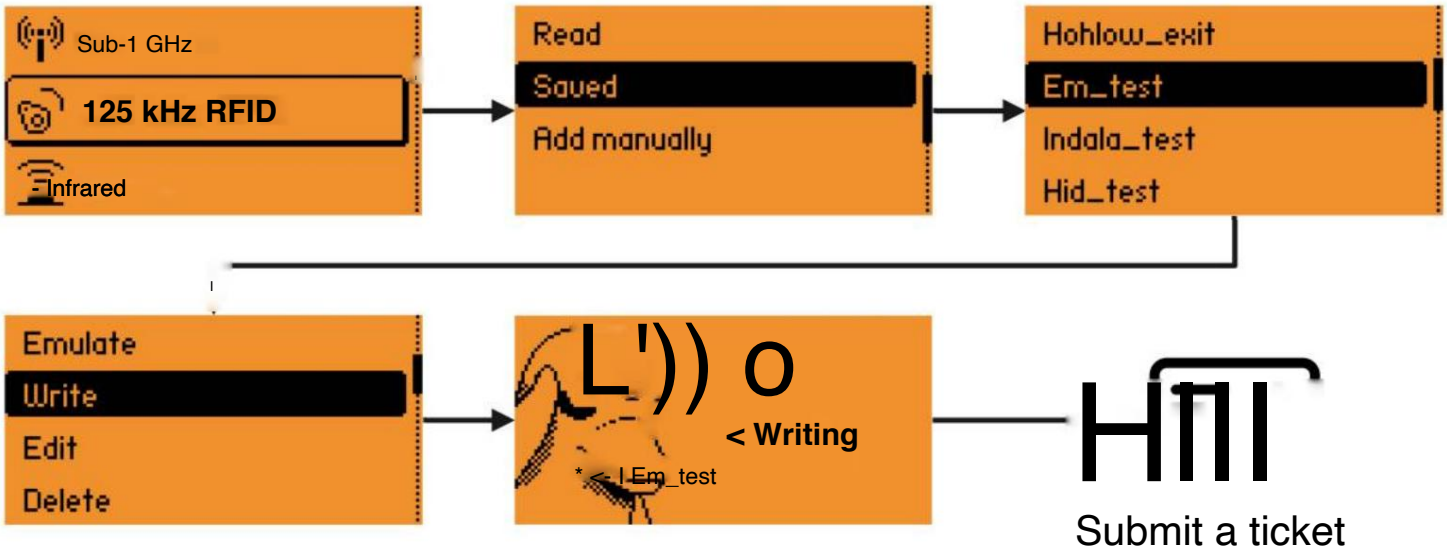


Menu for manually adding a map. Protocols supported EM4100, H10301 and 140134

To manually add an ID card, go to the menu125 kHz RFID -> Add manually -> [PROTOCOL NAME]and enter a unique code.

ÿ Imin

Recording on a blank



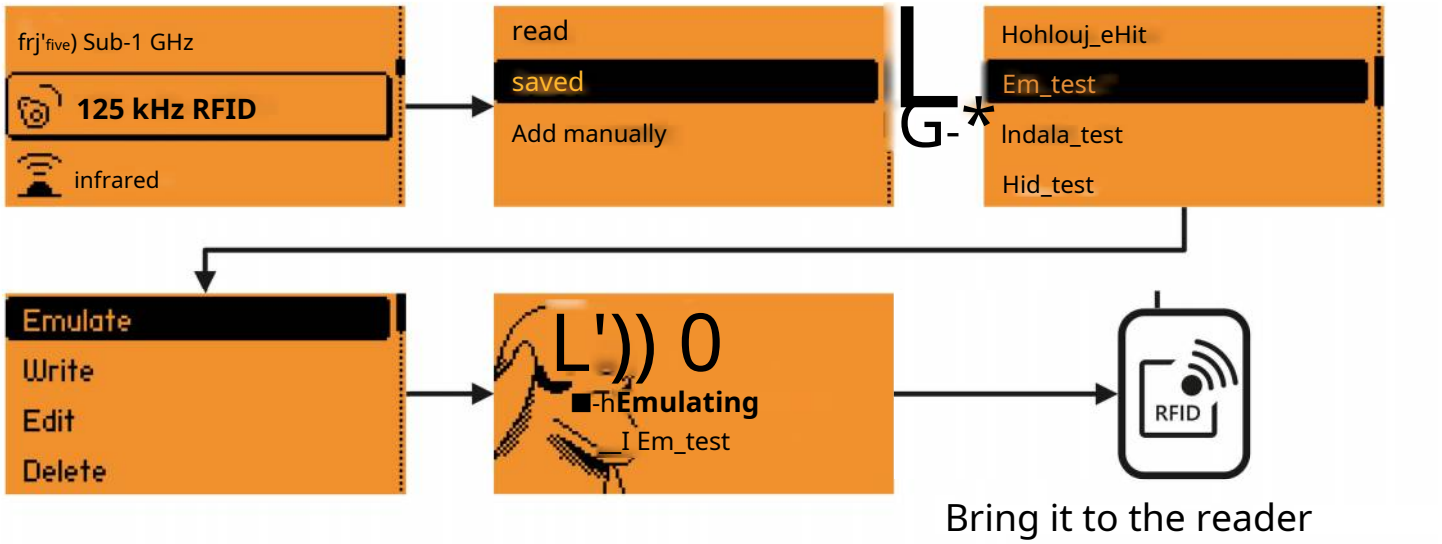
Menu for writing a unique code to a disc

To write a unique code to a blank, you need to go to 125 kHz RFID -> Saved -> [CARD NAME] -> Write .

There are different types of blanks: supporting different protocols and bypassing different checks of readers. The most popular blanks are T5577. They work with all the protocols that Flipper knows: EM4100, HID26, 140134.

Oh imin

Emulation



Stored label emulation menu

To emulate the card, go to the menu125 kHz RFID -> Saved -> [CARD NAME] -> Emulate

Some readers have protection against duplicate keys, overwriting the disc before reading, thereby changing the correct unique code. When emulated by Flipper, the reader will not be able to overwrite the unique code, so there will be no such problems.

10min

Types RFID readers

A photo	Name	Protocols reader	Flipper's work
	Iron Logic Matrix-II	EM-Marin	EM4100
	Iron Logic Matrix - III	EM-Marin, HID	EM4100, HID26
	Iron Logic ÿP Z-2	EM-Marin	EM4100
	Iron Logic Matrix V	EM-Marin	EM4100
	Smartec ST CE010EM	EM-Marin	EM4100
	Smartec ST PR011EM	EM-Marin	EM4100

	Parsec PR EH05	EM-Marin, HID	EM4100, HID26
	Parsec PR EH03	EM-Marin, HID	EM4100, HID26
	Proximus TM / W-3	EM-Marin	EM4100
	Bolid Proxy-3A	EM-Marin, HID	EM4100 - sometimes not triggered, HID26
	ELEKTA STEEL	EM-Marin	EM4100
	I lose CL05.1	EM-Marin, HID	EM4100, HID26
	Prox NeoPlus	EM-Marin, HID	EM4100, HID26
	ELTIS DP400-RDC16	EM-Marin	EM4100
	VISIT	EM-Marin	EM4100

	Proxway PW 309	EM-Marin	EM4100120671961221913 5
	HID OMNIKEY 5427 CK	EM-Marin, HID, Indala	EM4100, HID26, I40134 - does not work
	Proxmark RDV4	EM-Marin, HID, Indala	EM4100, HID26, I40134
	So many TS RDR-E	EM-Marin	EM4100
	CODOS RD–1100 M	EM-Marin, HID	Not verified
	CIFRAL CCD 2094ÿ	EM-Marin	EM4100
	Gate-Reader EH	EM-Marin, HID	EM4100, HID26 - does not work
	ikey TMD-5R	EM-Marine, HID, Indala	EM4100, HID26, I40134

	motorola asr 505	Indala	I40134
	Indala/motorola ASR 505/10022	Indala	I40134
	PROX Ltd I Reader	Indala	I40134 - does not work

6min

NFC

Overview of the NFC start menu

NFC, aka RFID 13.56 MHz, are contactless radio tags used in travel cards, bank cards, and electronic devices. The 13.56 MHz tags have a layered architecture and can have cryptography, authentication, and two-way exchange for enhanced security.

Low-level NFC architecture

Supported low-level operations:

- Reading - recognizes the typecards and reads low-level identifiers
- Daddition to RUch atYu
- Em atla cand I - primitive ACS sometimes possible fromBury with low-level identifiers

On the other hand, high-level protocols can only be read.
This allows you to explore the surrounding charts. The open code
allows you to write your own programs to work with charts.

Operations withMifare Ultrali ght - travel passesYes, intercom keychains:

- Reading

Operations withEMV – banCurrent charts:

- Reading

Work with theMifare Classic - pending

Work with theNFC NDEF - pending

More details about the NFC Architecture, the Standards and Protocols
used can be found in the section Once Rwork

ÿ Omin

Mifare Ultralight

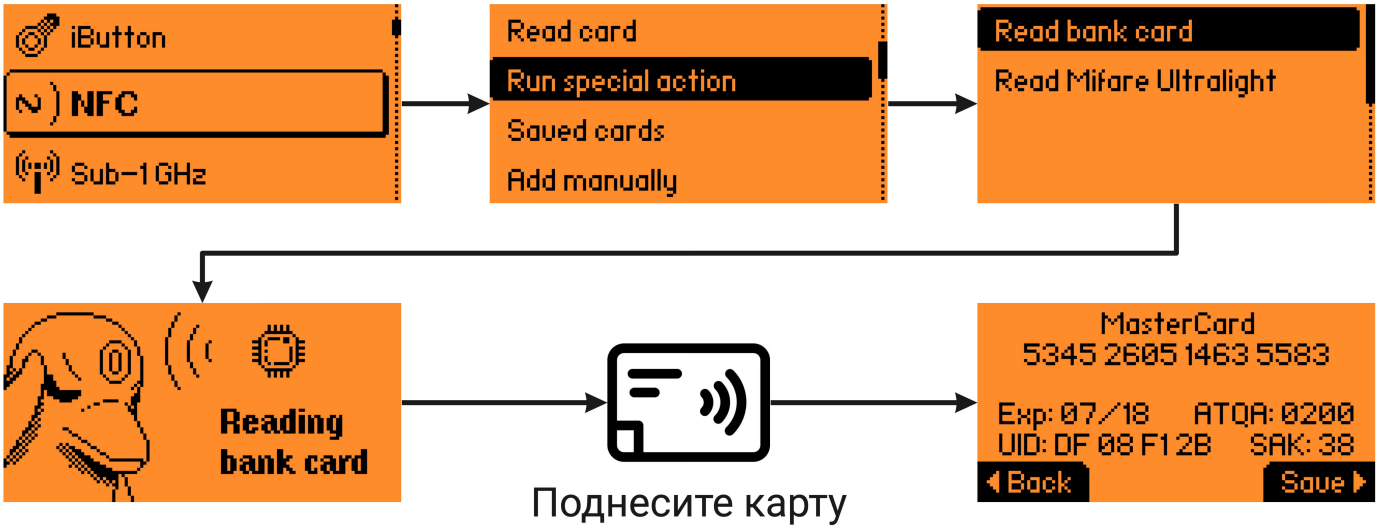
Mifare is a family of contactless smart cards that have their own different high-level protocols. Mifare Ultralight is the simplest card type in the family. In the basic version, it does not use cryptographic protection and has only 64 bytes of internal memory. The flipper supports reading Mifare Ultralight. Such tags are sometimes used as intercom key fobs, passes and travel cards. For example, Moscow transport tickets "Single" and "90 minutes" are made just on the basis of Mifare Ultralight cards.

Operation is now availableReading **Mifare Ultralight**

1min

Reading EMV bank cards

Information about the type of card on the link **EMV bank cards**



Menu for reading EMV bank cards

To read EMV bank cards, go to `NFC -> Run special action -> Read bank card` . From them you can subtract PAN (16 digits on the front of the card) and exp. date, as well as low-level UIDs, SAK, ATQA. `Select Save` to save the label to the SD card .

Community

Kickstarter
Habr.com
Discord
Forum
Blog

For developers

Developer Program
Github

Partners

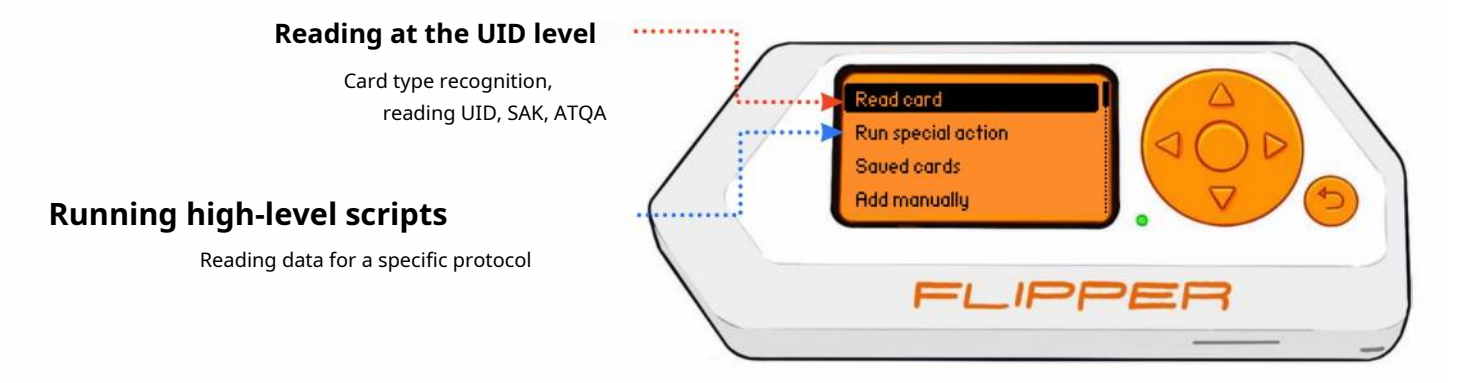
Neuron Hackerspace
Design Heroes
Slozhno.Media

About

Contacts
Company
Careers
Press kit
Privacy Policy
FAQ

About 14min

Reading



NFC reading is divided into two types - low-level and high-level

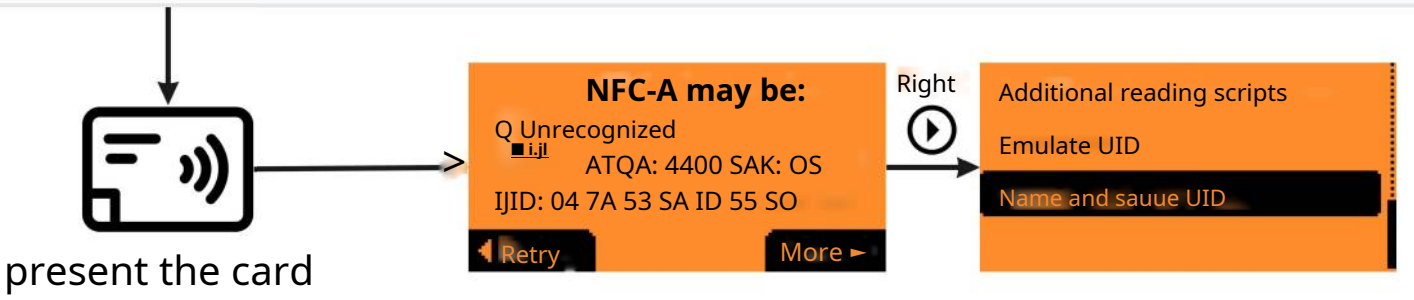
Reading 13.56 MHz labels in Flipper can be divided into 2 parts:

- **Low Level -**primary readingUID, SAK and ATQA. Based on this data, Flipper tries to guess what high-level protocol the card is running on. This guess cannot be 100% accurate, it's just a guess.

- **High Level -**reading data from card memory using specific high-level protocol:

  aboutReadingMifare Ultrali ght
  aboutReading HIM bank accounts rt
  o Read Mifare Classic - pending o Read NFC NDEF - pending

Low level reading

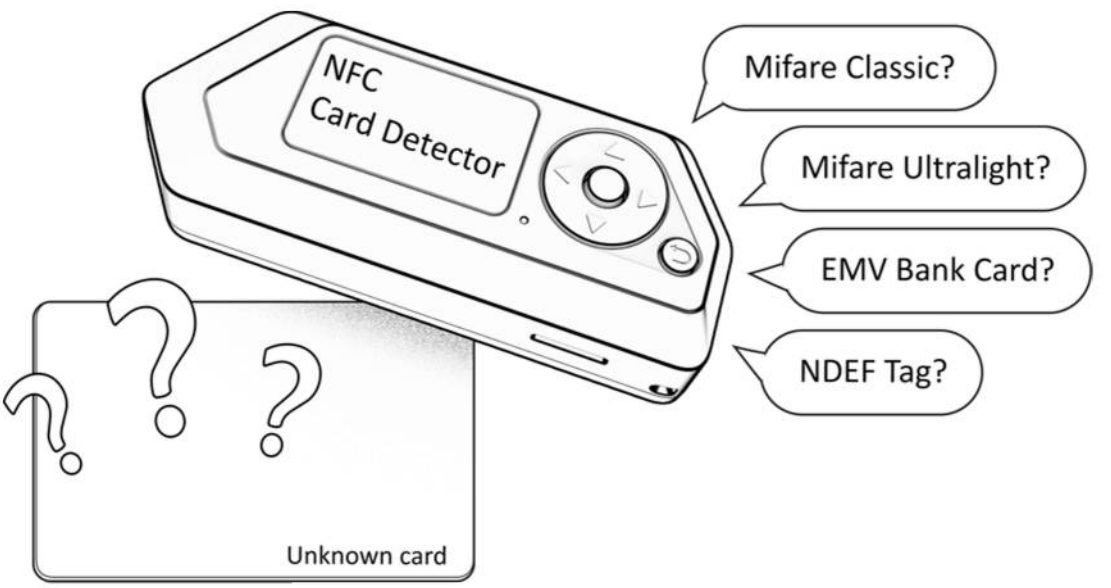


Low-level reading outputs: NFC type, intended protocol high level and low level identifiers

To read the map, go to the menuNFC -> Read card. The flipper will read the UID, ATQA, SAK and output the name of the proposed high-level protocol. If the protocol is not recognized, Unrecognized will be output. When the protocol is recognized, you will be prompted to proceed to high-level reading. To save the label, you need to go to the menuMore -> Name and save UID.

Virtual cards (such as the Apple Router) use a dynamic UID.

Purpose of low-level reading



Low-level reading recognizes the card type:

1. reads low level data

   about[ISO 14443-A data](#)

2. defines a low-level standard:

   about NFC-A - ISO 14443-A

   about NFC-B - ISO 14443-B

   about NFC-F - Felica

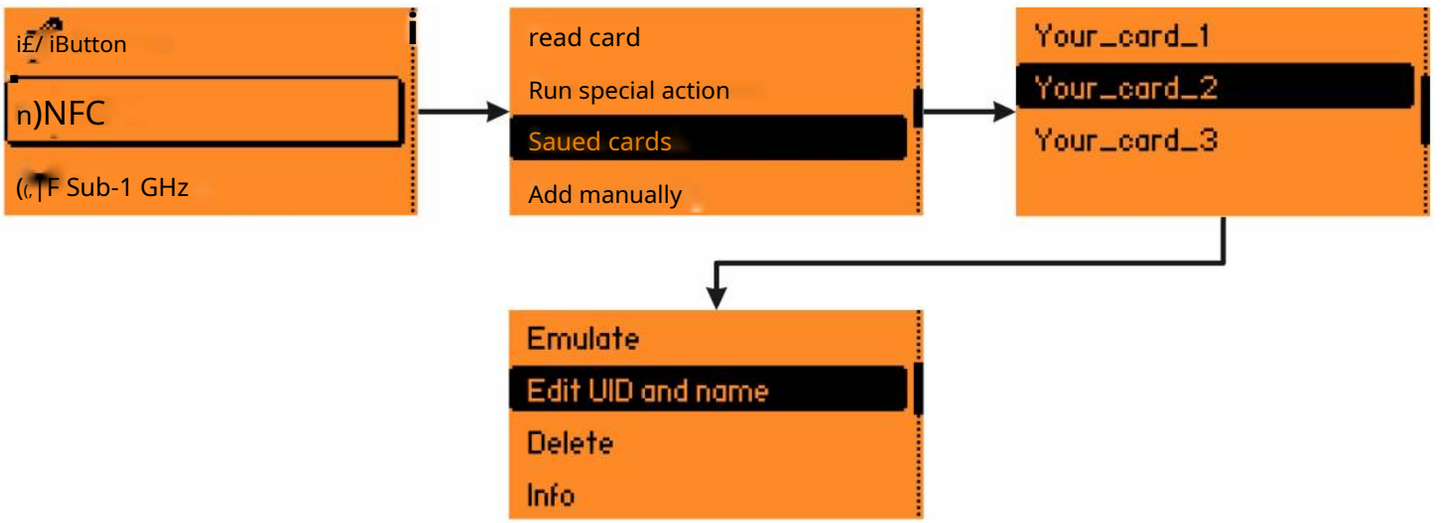
   about NFC-V - P2P

3. makes an assumption about the type of high-level protocol

At the moment, from low-level implementations, Flipper can only work with ISO 14443-A: read data, add manually and emulate. Other types of NFC (B, F, V) Flipper can recognize. In the future, we plan to add work with ISO 14443-B.

About 4min

Editing



Map editing menu

Flipper can edit the UID and name of the saved map. For this you need to go toNFC -> Saved cards -> [CARD NAME] -> Edit UID and name. To delete a card, go toNFC -> Saved cards -> [CARD NAME] -> Delete.

Editing a mark using a PC

NFC tags are stored on the SD card in the directory /nfc. Using a PC, you can view or change the data of the read cards.

Example of Mifare Ultralight type MFOUL11 data on SD card

Mifare Ultralight has different implementations, which may differ in memory space and data stack. The implementation of MF0UL11 has

An example of the Moscow transport card "Single", with the implementation of MF0UL11.

linux

```

```

PpimpnauuhiY FMVidptr

linux th

1min

Adding manually

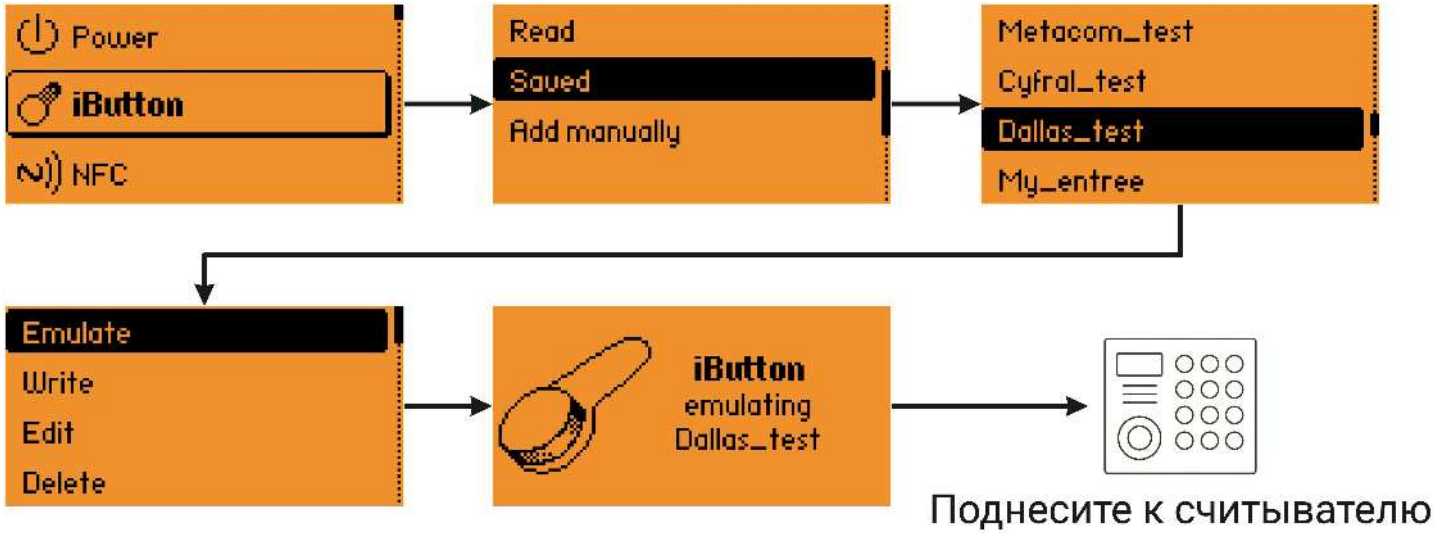
Menu for manually adding a key. Cyfral protocols are supported, Dallas, Metacom

To manually add a key, go to `iButton -> Add manualy` and choose its type:Dallas ,C yfral or Metakom .

The video shows an example of creating a new Cyfral key from 2 bytes.

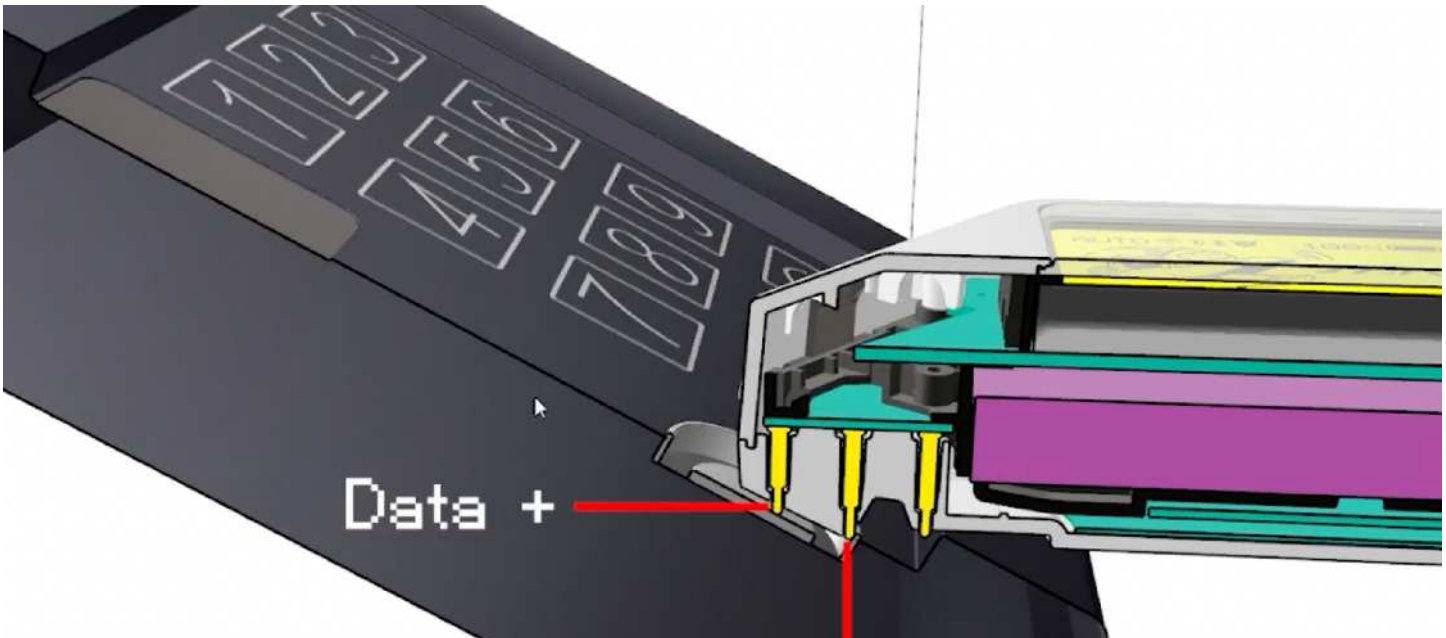
2min

Emulation



Saved Key Emulation Menu iButton

To start emulation, you need to go to the menu `iButton -> Saved-> [KEY NAME] -> Emulate` and bring the Flipper with the contacts к reader. Different pins are used to work with the iButton key and reader.



In emulation mode, Flipper transmits a specific ID and emulates only one predefined key protocol. So that Flipper will only open a specific intercom that knows this key.

It is impossible to sort through several keys at once in this mode, so it is impossible to unambiguously make sure whether the intercom has read our key, and it is impossible to know for sure the delay after reading the wrong key. Therefore, for a house, office, dacha, basement, it will be necessary to choose the specific key every time.

Not all intercoms recognize emulation

Some reader manufacturers use their own command sequences to prevent emulators from working. Therefore, we cannot guarantee that emulation will be supported with all doorphones.

About 4min

Types of NFC cards

Flipper now supports maps:

- Mifare Ultrali ght
- HIM bank cards

It is planned to add support for:

- Mifare Classic
- NFCNDEF

List of encountered cards

Map name	Notes
Cipurse transport cards	
German Identification Card	
Machine Readable Travel Document	
Felica	
FIDO and FIDO2	
Jooki	
LEGIC	
LTO Cartridge Memory	
MIFARE Classic	

MIFARE Desfire	
SEOS	
ST25TA	
thin film	
TOPAZ	
Waveshare NFC ePaper	

ÿ 3min

ИЗВlocal protocols

The transmitted code is:

- **static** - the same code is sent every time
- **dynamic -** the code changes every transmission. Usually the change algorithm is unknown, but there are exceptions

Dynamic protocols have internal encryption keys. They are are set by the manufacturer and are not disclosed.

Protocols are not tied to a specific frequency and modulation type. The same protocol may use different frequency and modulation depending on the manufacturer.

Protocol name	Stat./Din.
Princeton	Stat.
Bytec	Stat.
Tantos-Proteus	Stat.
GSN	Stat.
Nice Flo 12/24 bit	Stat.
CAME 12/24 bit	Stat.
Gate TX	Stat.

Taac_sin	Din.
Nice Flor-S	Din.
KeeLoq	Din.
DoorHan	Din.
AM-Motors	Din.
Stilmatic	Din.
HCS101	Din.
Alligator D-810, Alligator D-930	Din.
Alligator S-750RS	Din.
Alligator S-275	Din.
Alligator NS, NS-105, NS-205, NS-305, NS-405, NS-505, NS-605	Din.
Alligator M-550, M-500	Din.
Alligator L330	Din.
Pantera SLR-5100	Din.
Pantera CLK-355	Din.
Pantera SLK-2i, SLK-2i/3i/4i/5i/7i, SLK-25SC	Din.
Pantera CL-500, CL400, CL600	Din.
Pantera XS-1500, XS-2000, XS-1000, XS-1700, XS-100, XS-110	Din.
Pantera XS-2600, XS-2700	Din.
Jaguar JX-1000, XS-2700	Din.

iz-yu iu, bL-ÿ, U- / UU, u-yuu, 5- / UU	
Guard RF-311A	Din.
Septah A-90	Din.
Sheriff ZX-600	Din.
Sheriff APS-35 PRO	Din.
Sheriff APS-25 PRO	Din.
Sheriff APS-2400	Din.
Sheriff ZX-925, ZX-900, ZX-910, APS-75	Din.
Mongoose 800C, IQ-215	Din.
Mongoose 7000 RF, AMG-850C	Din.
Leopard LS50/10	Din.
Partisan RX-1	Din.
APS 3000, 2550, 2450	Din.
APS 2300, 2500, 2000,1500,1000, 500	Din.
StarLine	Din.
Cenmax ST-5A, Cenmax Vigilant V-5A	Din.
Cenmax ST-7A, Cenmax Vigilant V-7A	Din.
KGB FX-5	Din.
Tomahawk 9030, TW-9030, TW-7010, TW-9020, TZ 7010, TZ-9020, TZ-9030, HI, H2	Din.
Tomahawk Z5, Z3, ÿÿ, X5	Din.
rl inn ÿ A 7 ÿ A	P m

StatLine B6, B9	Din.
Harpoon BS-2000	Din.
Jaguar EZ-Beta	Din.

4min

List of NFC readers

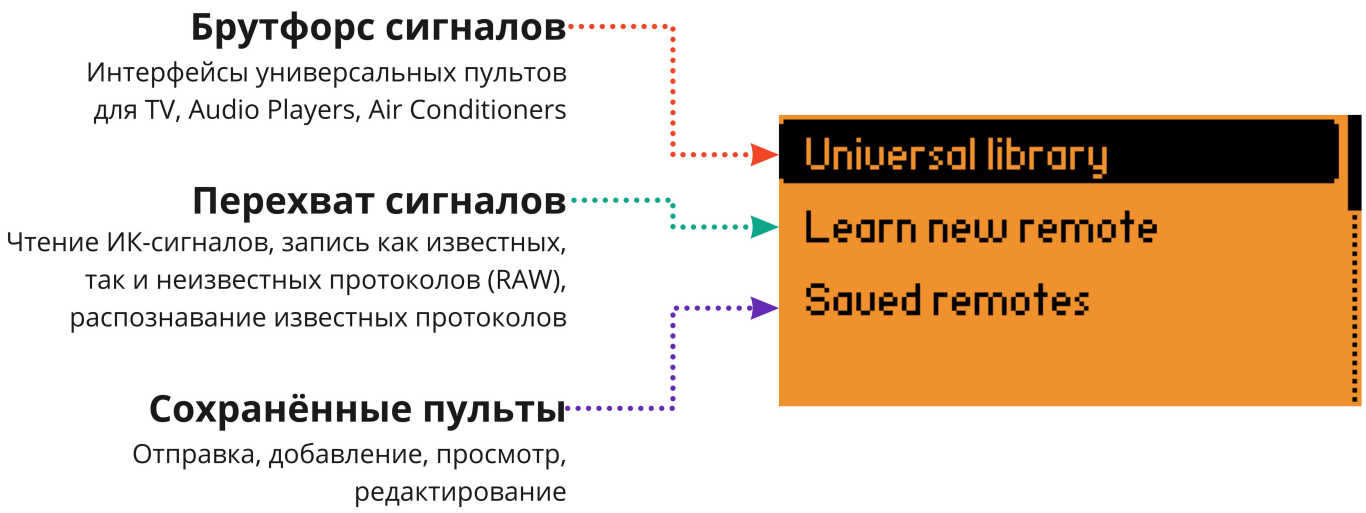
A photo	Name	Reader protocols
	Wisenet 5427CK	Mifare Classic, Mifare DESfire, iClass, iClass SE
	Proxmark3 RDV4	Cipurse transport Cards, German Identification Card, Machine Readable Travel Document, FeliCa, FIDO, FIDO2, Beverage, HID iClass, LEGIC, LTO Cartridge Memory, MIFARE Plus, MIFARE Classic, MIFARE Ultralight, MIFARE Desfire, SEOS, ST25TA, Thinfilm, TOPAZ, Waveshare NFC ePaper
	Nedap uPASS Access	Mifare Classic, HID iClass

	Supreme Xpass D2 XPD2-MDB	Mifare Classic, Mifare DESFire, Mifare Plus
		
	Smartec ST PR040MF	Mifare Classic
	Smartec ST PR140MK	Mifare Classic
	Parsec PNR-P15	Mifare Classic 1K/4K, Mifare Plus 2K/4K, Mifare ID
	Iron Logic Matrix-II	Mifare Ultralight, Mifare Classic 1K/4ÿ, Mifare ID
	Iron Logic Matrix-III	Mifare Ultralight, Mifare Classic 1K/4ÿ, Mifare ID

	Iron Logic ÿP-Z 2MF	Mifare Ultralight, Mifare Classic 1K/4ÿ, Mifare ID

4min

infrared port



Infrared start menu overview

TV remotes, air conditioners, music players often control devices via infrared. Flipper Zero allows you to control IR devices and perform:

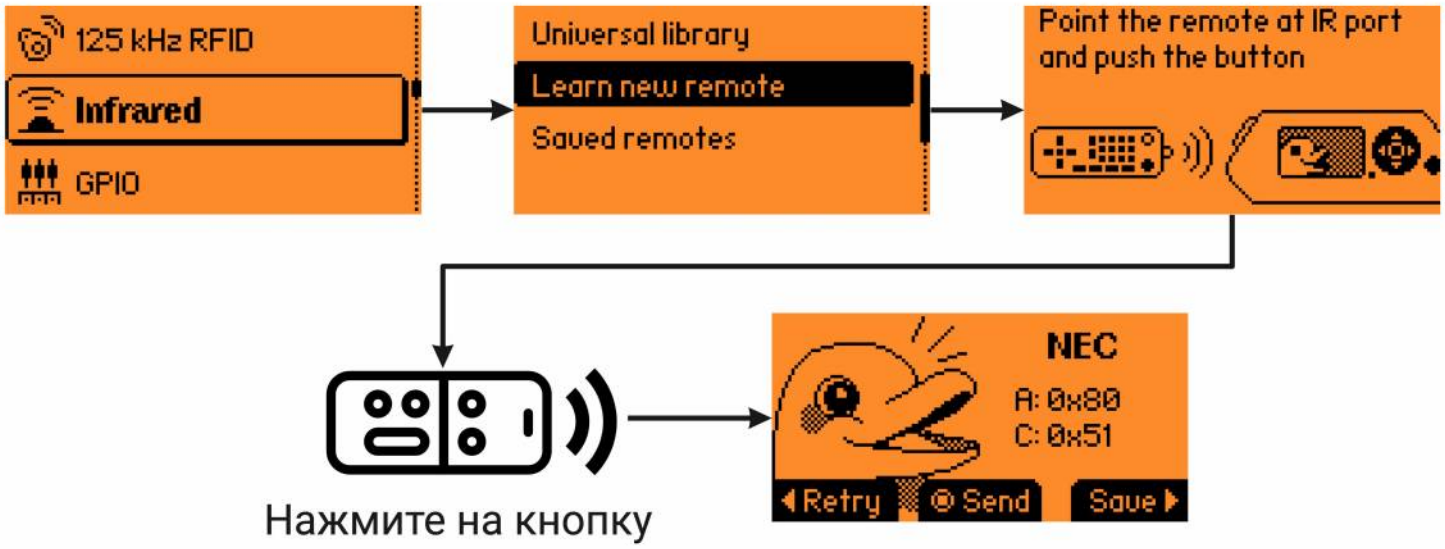
- B RUT fabout Rwords R
- You Pe Rsignal capture
- Otp Rave atsignal
- R atexact dadditiono dovp atltov Adding Your Protocols -
- Process Description add later

The flipper supports the reception and transmission of any IR code, even unknown to it, such as encoding and decoding the IR signal is performed by software. Documentation can be found FamousIR-p Rotohols

About 3min

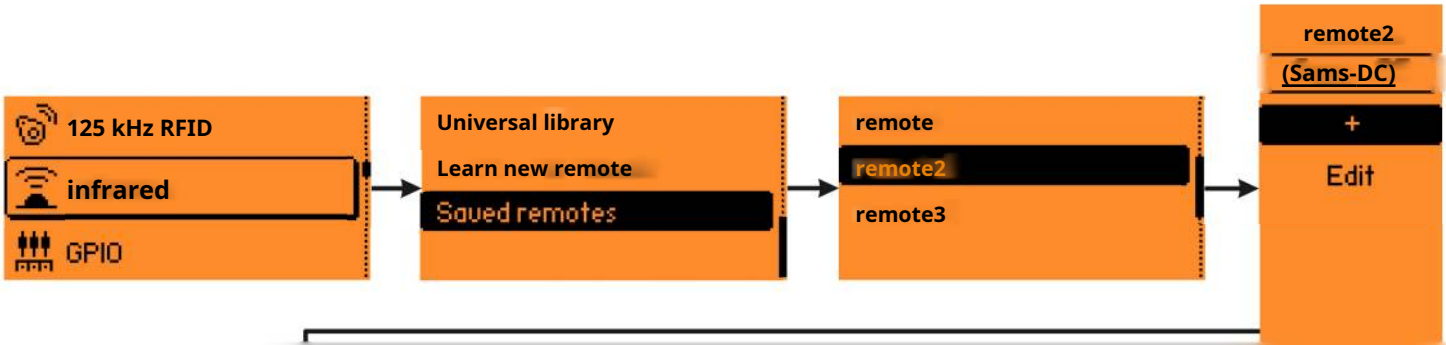
Reading

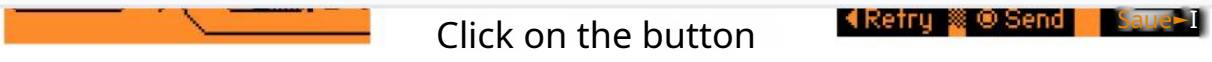
The sensitivity of the Flipper's infrared receiver allows you not to bring the remote control close to the Flipper's infrared port. You can intercept the signal while standing on the side, between the remote control and the device.



Menu for reading IR signal into a new remote control

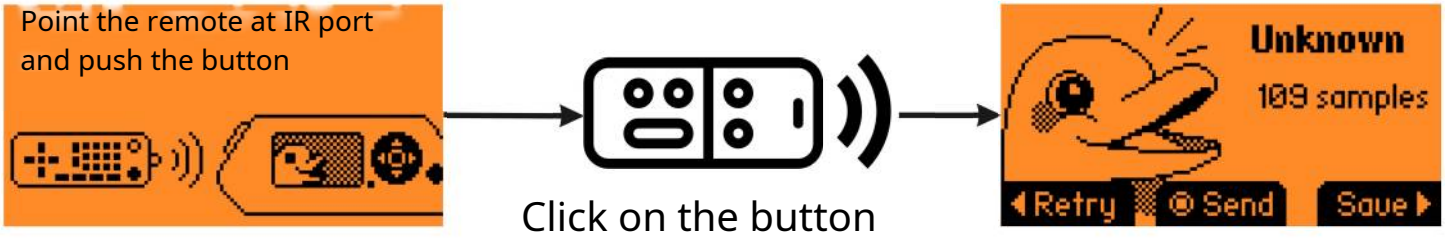
To read the IR signal, you need to go to the menuInfrared -> Learn new remote, after which Flipper will wait for the IR signal. The read signal can be sent or saved. When saving a waveform from the menuInfrared -> Learn new remote, a new virtual remote will be created.





Menu for signal interception to an existing remote control

To save the IR signal to an existing virtual remote control, you need to go to the menuInfrared -> Saved remotes -> [REMOTE NAME] -> +and read the IR signal. There can be an unlimited number of signals (buttons) in one remote control.



Reading a RAW signal

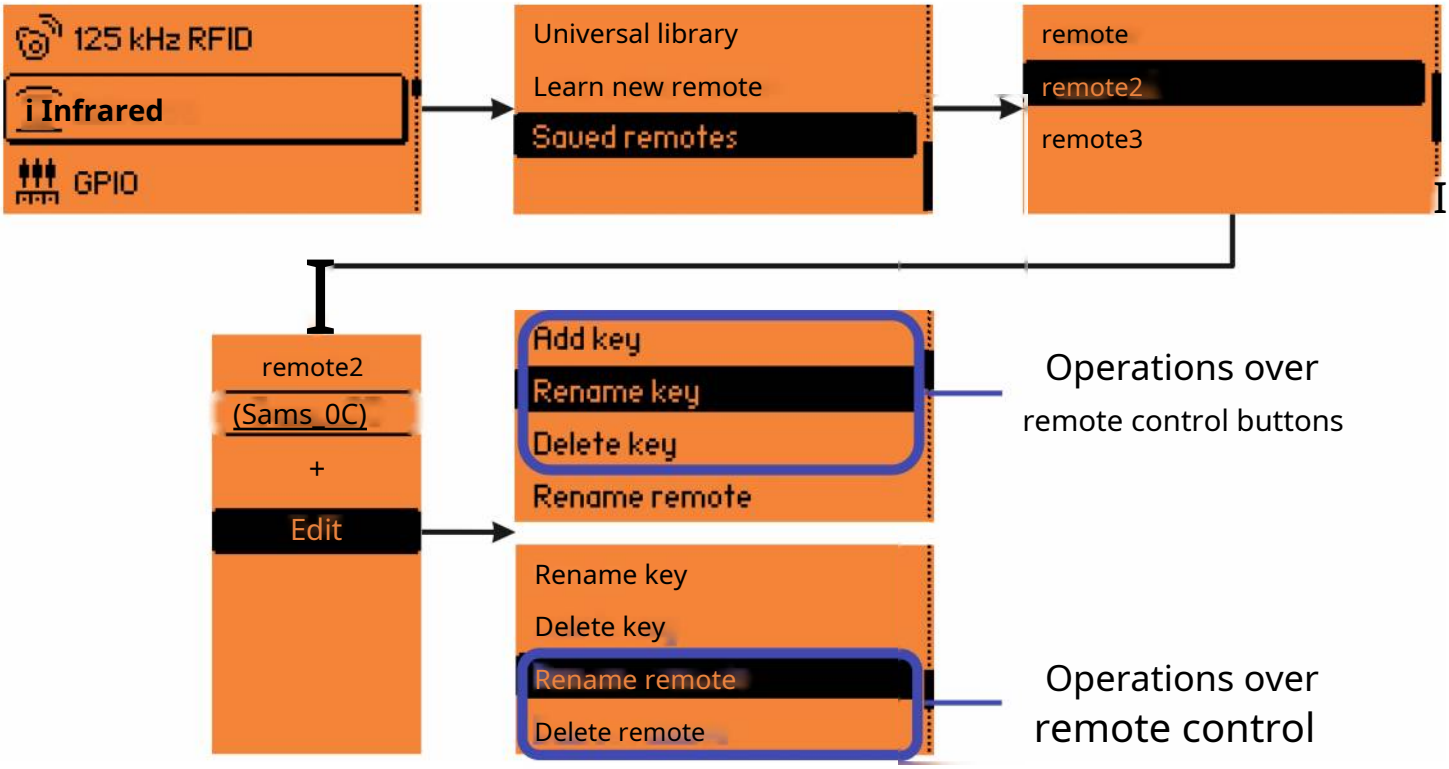
Flipper defines IR protocols on its own. If the IR protocol is unknown to Flipper, then the read signal can be used in RAW (raw) format - it can also be saved and sent.

Directory on SD card

When using an SD card, Flipper saves remotes in the /irda directory.

About 13min

Editing



Menu for editing signals and consoles

To edit the saved signal, go to the menu Infrared -> Saved remotes -> [REMOTE NAME] -> Edit, from where the signal can be renamedRename keyor delete Delete key.

Remotes that store IR signals can also be renamed or deleted. This is done from the same menuInfrared -> Saved remotes -> [REMOTE NAME] -> Edit, pointsRename remoteorDelete remoterespectively.

Editing the remote using a PC

/ivaa/universal. and with the help of nk you can view, add or change the remote control. Information about the storage format of a particular protocol is stored in the section **Development**

An example of a console file with intercepted signals

The format of the intercepted Samsung32 signal contains:

- **Button name**
- **Signal Format —**Samsung32
- Protocol specific data (may differ from protocol to protocol)
    - **Address —**1 byte
    - **Command code —**1 byte

linux

An example of a universal remote control file for

signals, and the "button name" of the universal remote control, you can record the desired number of codes.

Text

An example of a remote control file with a RAW signal

The format of the captured RAW signal contains:

- **Button name**
- **Signal Format —**RAW
- **Carrier frequency**[Hz] - to send supports values 10..54 kHz. The receiver uses 38 kHz for reading.
- **Duty cycle [%] —**for sending supports values 1..100%. For recording Flipper always writes 33 as it is used more often (photodetector does not share information about Duty Cycles)

- **Intercepted signal**[µs] - time ranges the presence / absence of a signal with alternation. The first number is the presence of IR pulses, the second is the absence, and so on.

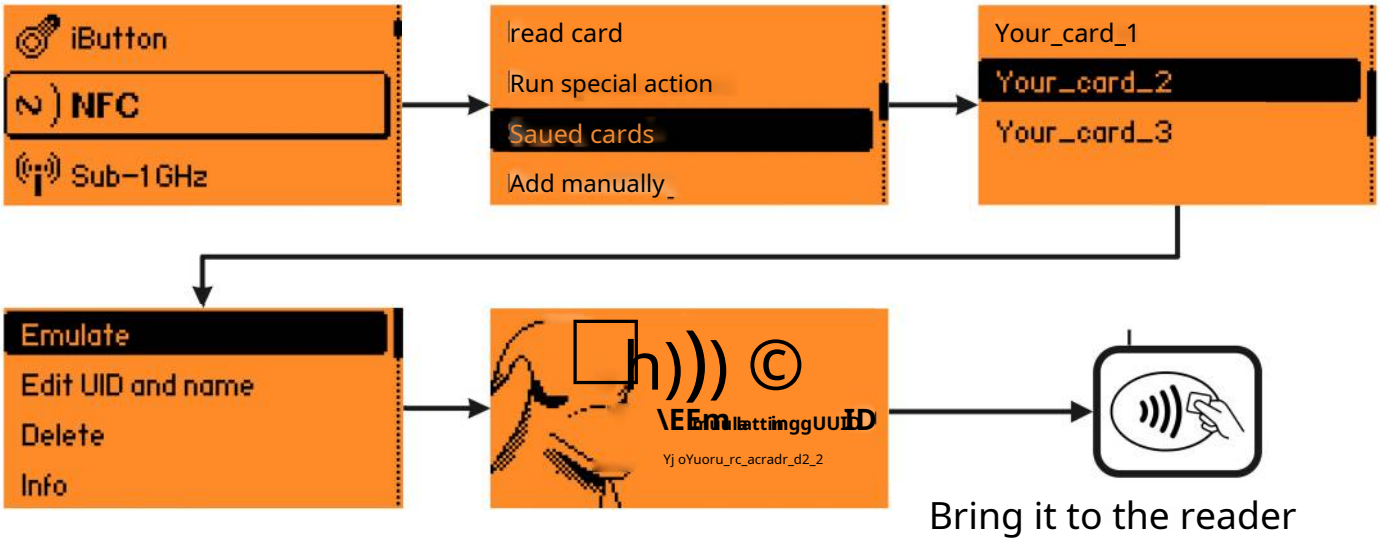
linux

Oh imin

Emulation

Works only for IS014443-A

So far, Flipper can only emulate ISO 14443-A parameters: UID, SAK, ATQA .



Stored label emulation menu

Flipper can emulate ISO 14443-A low-level data by responding to the desired UID, SAK, and ATQA. To emulate the label, you need to go to the menuNFC -> Saved cards -> [CARD NAME] -> Emulate .

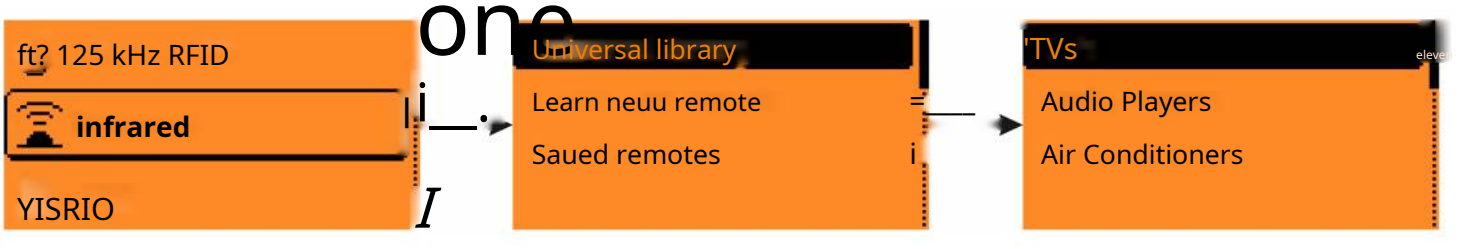
About 5min

Brute force by dictionary

Flipper Zero can be used as a universal remote control for any TV, air conditioner or media center. In this mode, Flipper iterates over a dictionary with known codes from different manufacturers, stored on the SD card. The larger the dictionary, the longer it will take to wait until the enumeration of all signals is over.

Dictionary update

If there are no dictionaries on the SD card, you won't be able to enable brute force! Dictionary files are downloaded to the SD card when updating the firmware via the qFlipper desktop application or added manually.

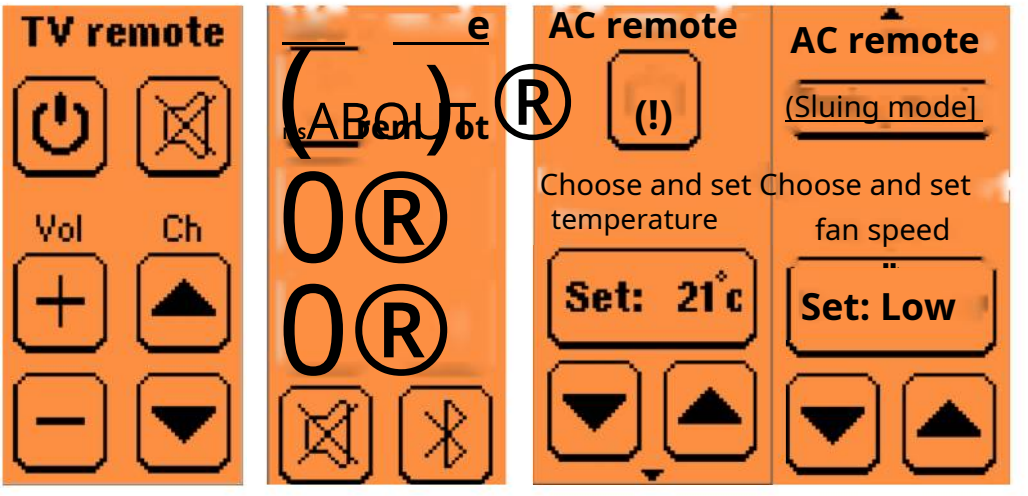


Menu of universal remotes

For brute force, 4 types of universal remotes are used, and dictionaries for them are located in the directories:

- TVs -ZirdaZuniversalZtv.i
- Honeyianentry -Zirda/universal/mc.ir
- Air conditioners-ZirdaZuniversaIZac.i

vertical - it's more convenient
hold the device in your hand, directing the infrared port towards the receiver. The appearance of the interfaces is shown below:



The appearance of universal remotes: TV, media center, air conditioner

During the enumeration of the dictionary, the execution status is displayed on the screen, and the enumeration itself can be interrupted at any time.

1min

Air conditioners

Expected

Software interface not added to firmware

The universal TV remote is selected in the menu Infrared -> Universal library -> Air Conditioners . It contains the buttons for power on/off and the mode that changes the direction of the air flow (SWING), switching the temperature and power of operation.

IR remote controls from air conditioners, unlike the others, do not transmit the command of the pressed button, but as a whole all the parameters of the air conditioner that are visible on the remote control screen. That is, they always send ALL air conditioner settings in one large package.

1min

media centers

Expected

Software interface not added to firmware

The universal TV remote is selected in the menu Infrared -> Universal library -> Audio Players . He can manage power on, sound, track playback and pairing via Bluetooth.

2min

TVs

The universal remote for TVs is located in the menu Infrared -> Universal library -> TV's . It can control power on, sound and change channels.

In the future, we plan to store dictionaries with IR codes in a separate database available to all users. Temporarily, a dictionary of popular TV shutdown signals will be stored as a list on this page.

List of commands to turn off TVs

JS JS

6min

Notable IR protocols

We invite everyone to participate in adding well-known IR protocols! With the help of the community, we want to collect a comprehensive database of IR protocols that anyone can use.

At the moment, Flipper knows the following IR protocols for TVs, media centers, etc.:

- NEC
- NECext
- RC6
- Samsung32

Protocols for air conditioners have not yet been entered into the firmware. Once entered, they will be here: Air conditioners

List of the most common protocols

All of these protocols, with the exception of BANG_OLUFSEN (Bang & Olufsen), operate on a carrier modulation basis, with frequency from 30Hz to 40Hz and 25-35% duty cycle. The exception is Bang & Olufsen with a carrier frequency of 455Hz.

Name of the protocol	Embryos
SIRCS	Sony
NEC	NEC with 32 bits, 16 address + 8 + 8 command bits, Pioneer, JVC, Toshiba, NoName etc.
NEC16	NEC with 16 bits (incl. sync)
NEC42	NEC with 42 bits
SAMSUNG	Samsung
SAMSUNG32	Samsung32: no sync pulse at bit 16, length 32 instead of 37
SAMSUNG48	air conditioner with SAMSUNG protocol (48 bits)
LGAIR	LG air conditioner
MATSUSHITA	Matsushita
TECHNICS	Technics, similar to Matsushita, but 22 instead of 24 bits
KASEIKYO	Kaseikyo (Panasonic etc)
PANASONIC	Panasonic (Beamer), start bits similar to KASEIKYO
MITSU_HEAVY	Mitsubishi-Heavy Aircondition, similar timing as Panasonic beamer
RECS80	Philips, Thomson, Nordmende, Telefunken, Saba
RC5	Philips etc.
DENON	Denon, Sharp
RC6	Philips etc.
APPLE	Apple, very similar to NEC
RECS80EXT	Philips, Technisat, Thomson, Nordmende, Telefunken, Saba

NUBERT	Nubert
BANG_OLUFSEN	Bang & Olufsen
THOROUGH	Thorough
NOKIA	Nokia
SIEMENS	Siemens, eg Gigaset
FDC	FDC keyboard
RCCAR	RC Car
JVC	JVC (NEC with 16 bits)
RC6A	RC6A, e.g. Kathrein, XBOX
NIKON	Nikon
RUWIDO	Ruwido, e.g. T-Home Mediareceiver
IR60	IR60 (SDA2008)
KATHREIN	Catherine
NETBOX	Netbox keyboard (bitserial)
LEGO	LEGO Power Functions RC
THOMSON	Thomson
BOSE	BOSE
A1TVBOX	A1 TV Box
avalanche	ORTEK - Hama
TELEFUNKEN	Telefunken (1560)
ROOMBA	iRobot Roomba vacuum cleaner
RCMM32	Fujitsu-Siemens (Activy remote control)
RCMM24	Fujitsu-Siemens (Activy keyboard)
RCMM12	Fujitsu-Siemens (Activy keyboard)
SPEAKER	Another loudspeaker protocol, similar to Nubert
MERLIN	Merlin (Pollin 620 185)
PENTAX	Pentax camera
FAN	FAN (ventilator), very similar to NUBERT, but last bit is data bit instead of stop bit
S100	very similar to RC5, but 14 instead of 13 data bits
ACP24	Stiebel Eltron ACP24 air conditioner
VINCENT	Vincent
SAMSUNGAH	SAMSUNG AH
IRMP16	IRMP specific protocol for data transfer, e.g. between two microcontrollers via IR
GREE	Gree climate
cool the	RC II Infra Red Remote Control Protocol for FM8
METZ	METZ
ONKYO	Like NEC but with 16 address + 16 command bits

21min

WiFi Module

What Wi-Fi Module Can Do

Flipper Zero Wi-Fi Module is a native wireless/wired debugger and programmer for Flipper Zero built around the ESP32-S2 module and the open source Black Magic Probe tool. At the moment he is able to flash and debug various
microprocessors and controllers (including the one used in Flipper Zero) via Wi-Fi or USB, and work as a USB-UART adapter.

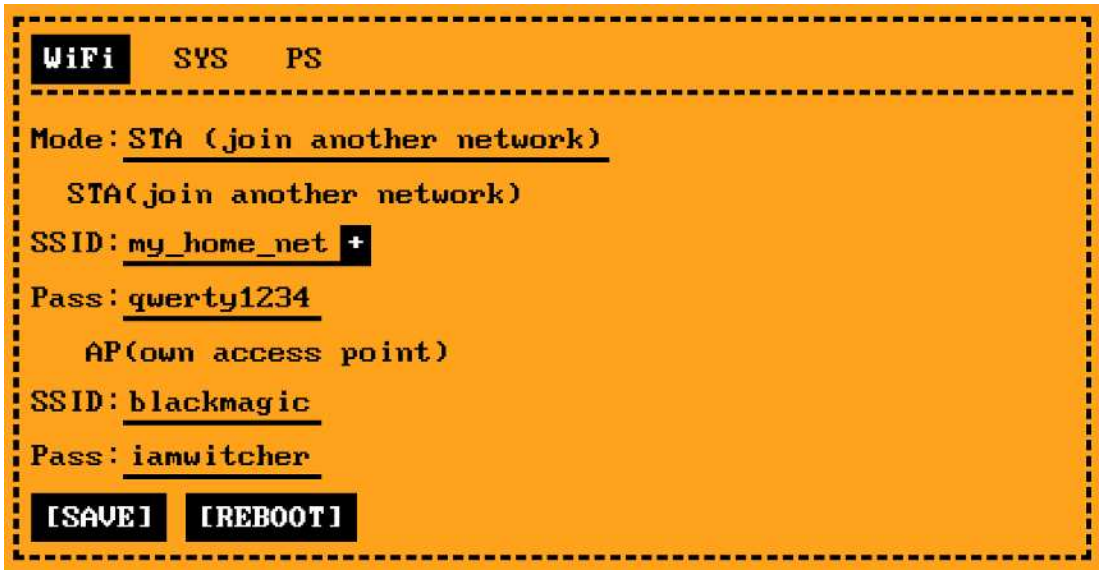
Terms

STA - the mode in which the Wi-Fi module connects to an existing network.
AP - the mode in which the Wi-Fi module creates its own
network
SSID - name of the Wi-Fi network SWD -
firmware and debugging protocol JTAG -
firmware and debugging protocol
GDB - debugger program, usually consists of a server and a client, in our case, the Wi-Fi module acts as a server

UART - Data Transfer Protocol USB - Data
Transfer Protocol

via WiFi

1. Power up the module via USB or by plugging it into the Flipper Zero expansion port.
2. Connect to the blackmagic Wi-Fi network that appears with the password iamwitcher.
3. Go to 192.168.4.1
4. Set up the device connection for your network. Select the STA mode, and in the STA section, configure the SSID and password for your network. SSID can be selected from the list by clicking on . Save the settings and restart the module.
5. The device is now available on your network at blackma gic.local if your OS supports protochol mDNS. If this protocol is not supported, you will have to find out the IP address yourself.



Example setup for connecting the debugger to an existing network my_home_net with password qwerty1234

via USB

1. Connect the module using the USB connector
2. The first port that appears is for communicating with GDB

server

The next port is for reading UART

Debugging with a module

via WiFi

```
BLACKMAGIC=BMP_IP_ADDRESS:2345 make blackmagi c
```

- Where `BMP_IP_ADDRESS` - IP address of the module

via USB

```
BLACKMAGIC=/dev/BMP_SERIAL make blackmagi c
```

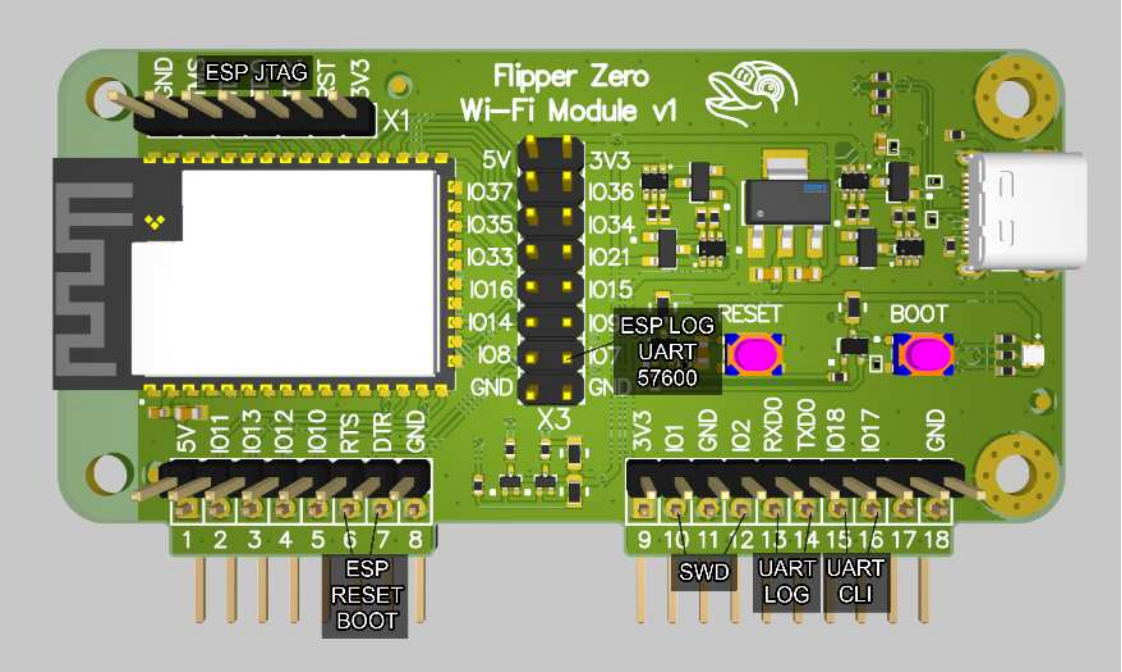
- Where `/dev/BMP_SERIAL` - port advertised by the module
- On MacOS, be sure to open cu(calling unit) for example /dev/cu.usbmodemblackmagic1

Module firmware update

1. Download the firmware you are interested in fromse Rve Rupdates
2. InstallP ython 3
3. Install PIP
   ```
   python3 -m ensurepip --upgrade
   ```
   ```
   python3 -m pip install --upgrade pip
   ```
4. Install ESPTool
   ```
   pip3 install esptool
   ```
5. Connect the module using the USB connector
6. Hold down the BOOT button on the module, briefly click on the RESET button, release BOOT. The system will have a COM port intended for firmware
7. Execute the command from the flash.command file (it comes with the firmware) substituting the port that appeared in the previous step instead of (PORT), for example:

```
esptool.py -p COM12 -b 460800 --before default_reset
--after hard_reset --chip esp32s2 write_flash --
flash_mode dio --flash_freq 80m --flash_size 4MB
0x1000 bootloader.bin 0x10000 blackmagic.bin 0x8000
partition-table.bin
```

Module interfaces



Interfaces that the module uses

6min

iButton

iButton start menu overview

iButton is the general name for a dongle in a metal “tablet” form factor. It is often erroneously called a "magnetic" key, but this is incorrect. Inside the iButton is a full-fledged microchip that works according to a digital protocol. Often used as an intercom key.

Appearance of the iButton key

Operations available for iButton:

- reading – intercom role (1-Wire Master)

Flipper uses different pins for reading and emulating

Supported key protocols:

- Dallas
- C yfral
- Metakom

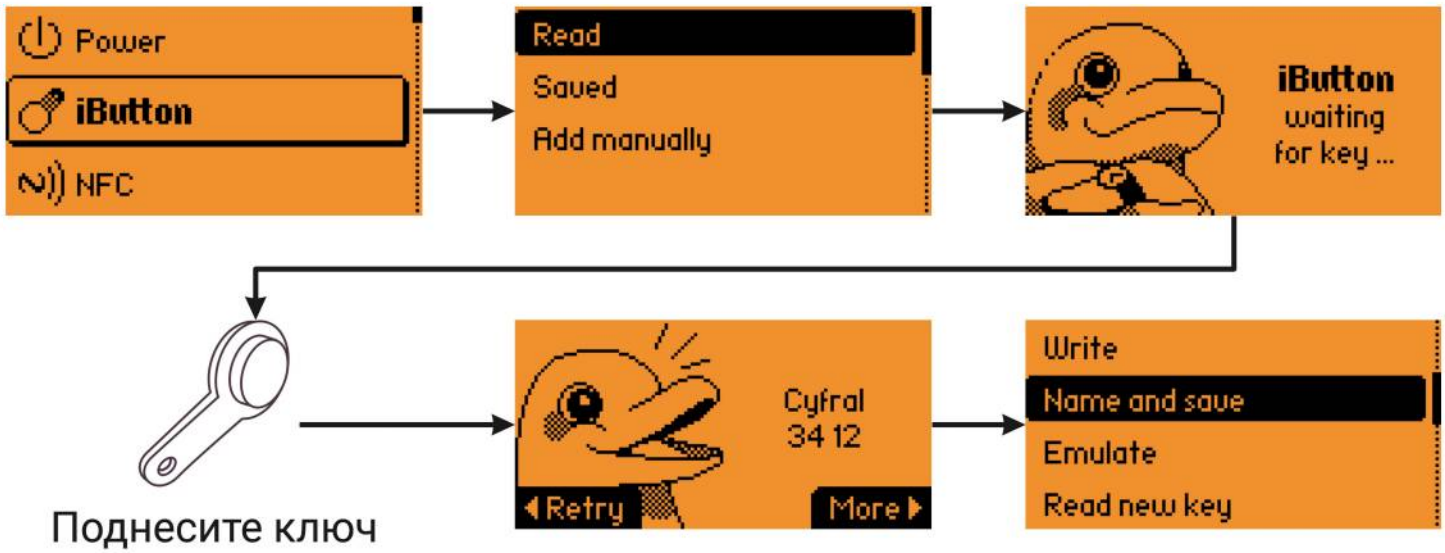
More details about the principles of iButton operation, hardware implementation, transfer protocols, etc. can be found in the section Once Rwork .

Not all "tablets" are intercom keys!

In this form factor, there are not only simple keys with ID, but also climate sensors, storage devices
cryptographic keys.
These devices look the same as intercom keys, but they are not.

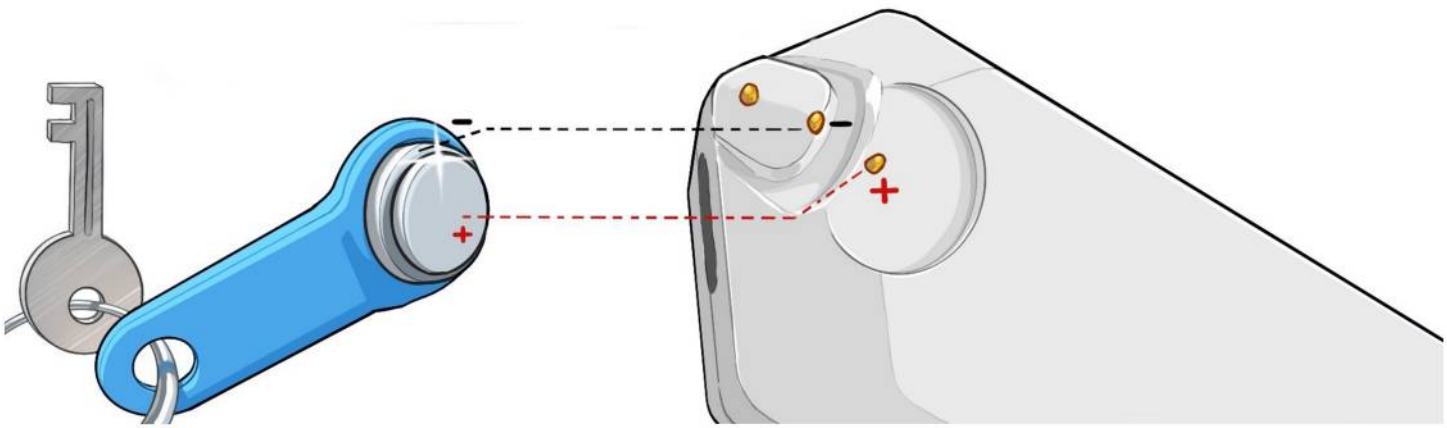
About 6min

Reading



iButton key reading menu

To read iButton you need to go toiButton -> Readand bring the key to the Flipper's contact pad. On a successful read, the type and key data are output. To save the key after reading, go to the menuMore -> Name and save.



Flipper contact pad for working with iButton keys

Read errors

Key data structure**Dallas** has the fields you need write it down in a special way. When writing incorrect values, the following errors are possible:

- **Incorrect checksum —**mistake inCRC
- **Wrong family code —**whenfamily code is different from 0x01, Flipper swears that this is not an iButton key

CRC byte Error

Family-code-byte Error

Possible errors when reading Dallas keys: invalid CRC byte — CRC ERROR; Family-code byte is not equal to 0x01 - THIS IS NOT A KEY

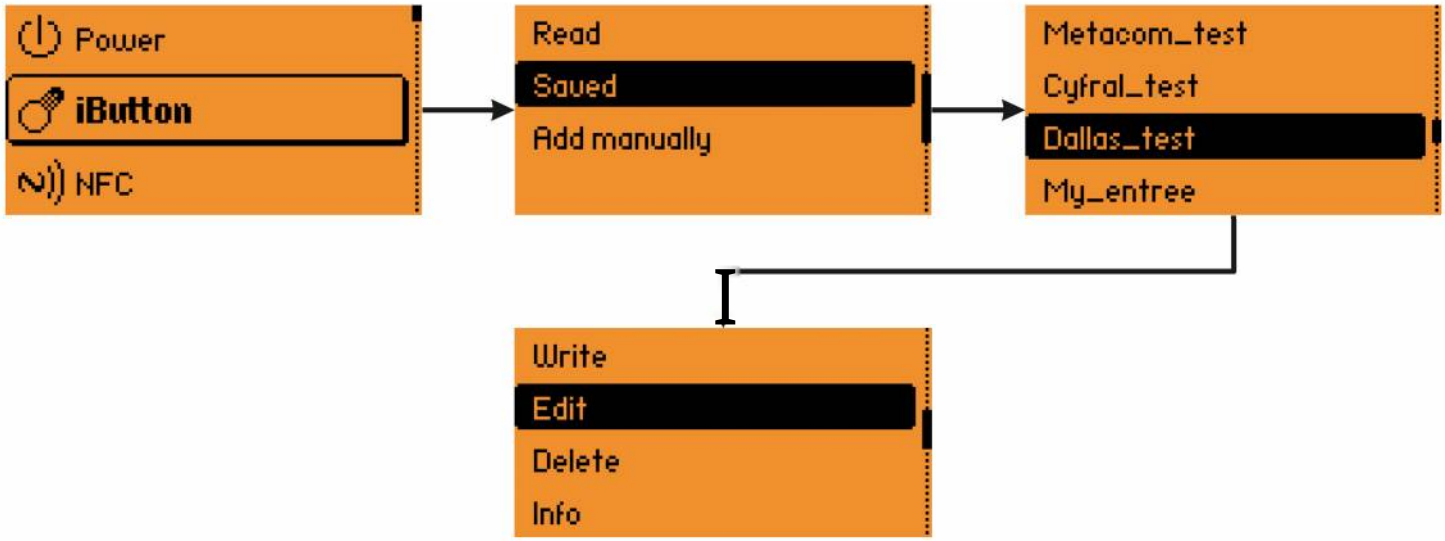
To avoid errors, you need to use the Family code equal to 0x01 and the CRC-byte offered by Flipper in the error message (Expected

Some keys may not be read or read with errors due to the following factors:

- This is a type of key unknown to Flipper
- The key is probably damaged
- It's not the intercom key

About 6min

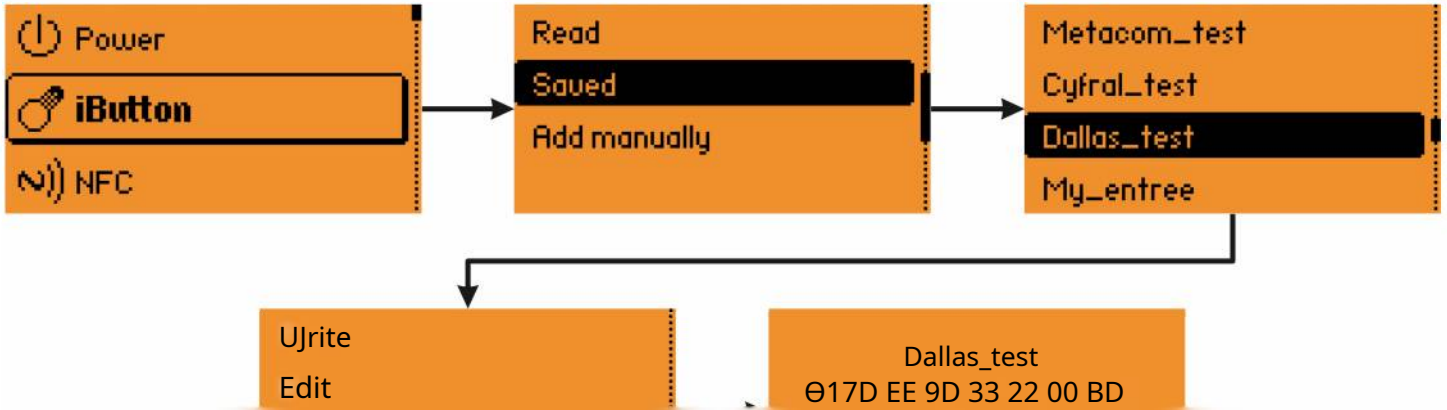
Редактирование



Menu for changing the name and key data

To edit the name and data of the saved key, go to the menuiButton -> Saved -> [KEY NAME] -> Edit. To delete a key, go toiButton -> Saved -> [KEY NAME] -> Delete.

Key Information



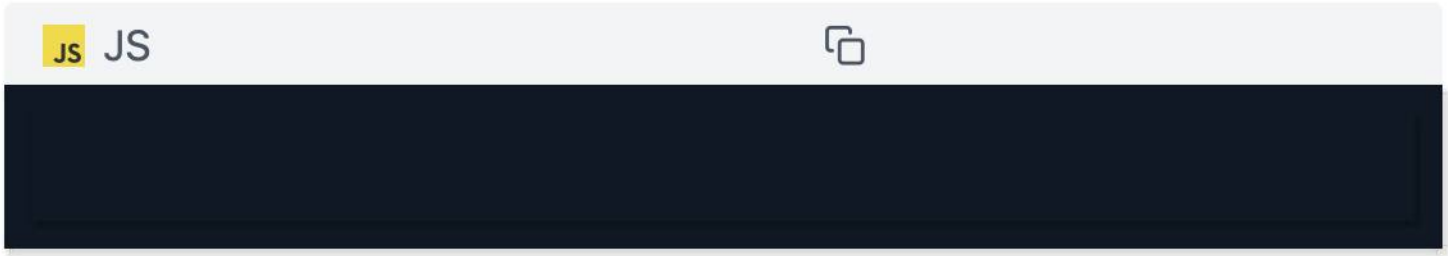
Information about the key is available in the menu`iButton -> Saved -> [KEY NAME] -> Info`.

Editing a key using a PC

iButton keys are stored on the SD card in the directory `/ibutton`. Using a PC, you can view or change the data of the stored keys.

The key storage format contains:

- **Key type -**"D" Dallas, "C" Cyfral, "M" Metakom
- **Data**-each key type has its own data structure. More about it can be found in the section**Development**



Oh imin

Adding manual

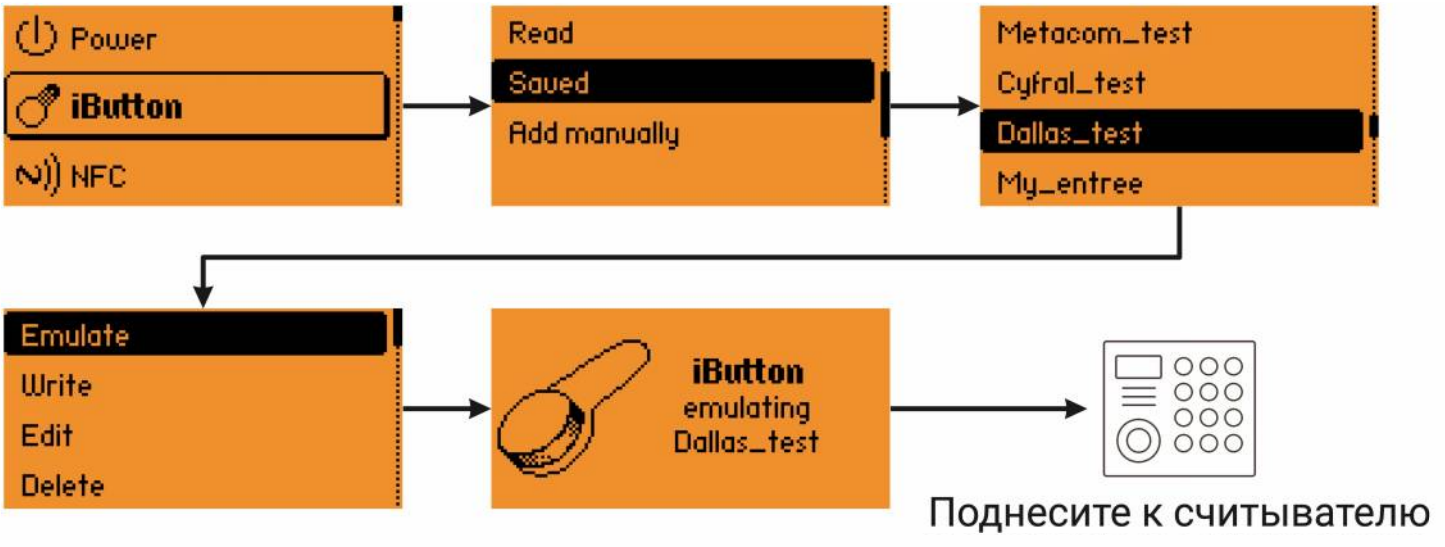
Menu for manually adding a key. Cyfral protocols are supported, Dallas, Metacom

To manually add a key, go toiButton -> Add manualand choose its type:**Dallas . Cyfral** or**Metakom .**

The video shows an example of creating a new Cyfral key from 2 bytes.

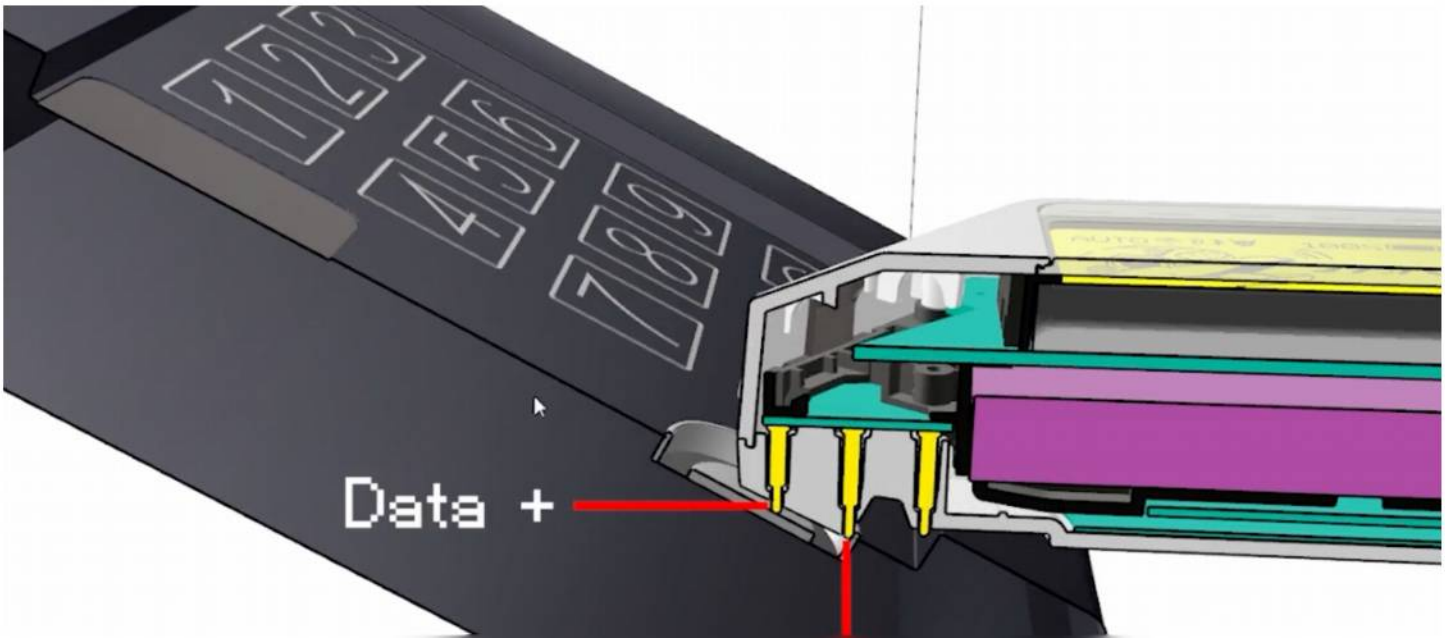
About 2min

Emulation



Saved iButton key emulation menu

To start emulation, go to the menu`iButton -> Saved -> [KEY NAME] -> Emulate`and bring Flipper with contacts to the reader. To work with the key and the iButton reader, different outputs are used.



In emulation mode, Flipper transmits a specific ID and emulates only one predefined key protocol. So Flipper will only open to a specific intercom that knows that key.

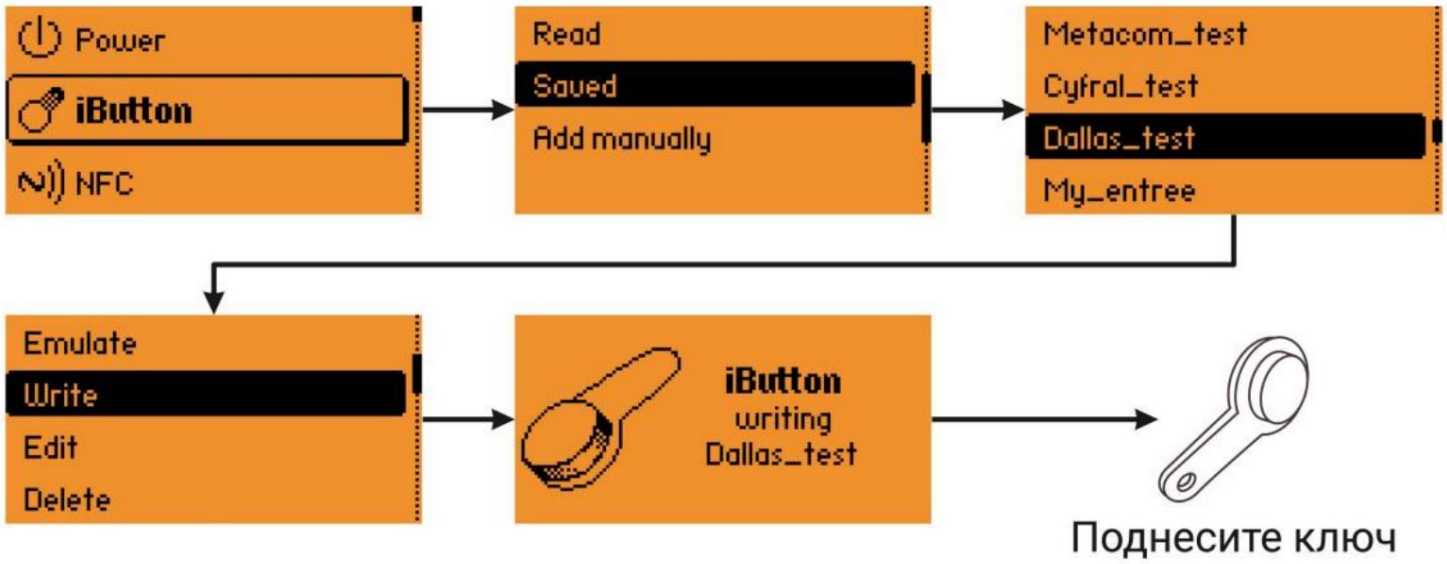
It is impossible to sort through several keys at once in this mode, since it is impossible to be sure whether the intercom has read our key, and it is impossible to know for sure the delay after reading the wrong key. Therefore, for a home, office, cottage, basement, you will need to select a specific key from the menu each time.

Not all intercoms recognize emulation

Some reader manufacturers use their own command sequences to prevent emulators from working. Therefore, we cannot guarantee support for emulation with all intercoms.

About 6min

Запись на болванку

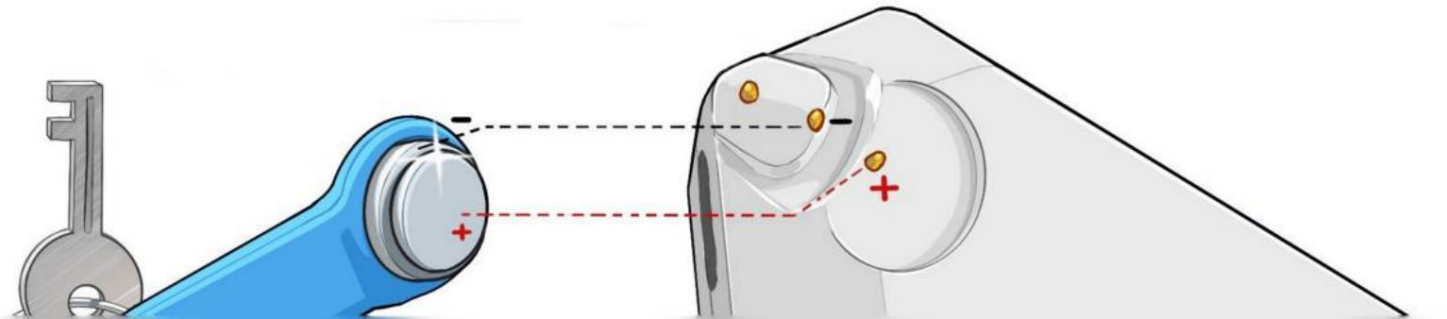


Menu for writing the iButton key to a blank

To write a key to a disc, you need to go to the iButton -> Saved -> Key_foo_bar -> Write menu and bring the key to the Flipper's pad.

There are different types of pigs: some only support one transfer protocol, others support all protocols.

We recommend using RW1990, TM2004, TM01C blanks supported by Flipper.



The nuances of recording on a disc

The recording process has nuances depending on the model of the intercom,
and the type of blank:

Recording a disc may require increased voltage - the RW2000 requires 8 volts to write.
Some blanks require finalization - after finalization, the key can no longer be overwritten.

Some intercoms take advantage of this by trying to overwrite the key before reading to avoid fake keys.

Features of Vizit intercoms

new dimisrins (handle zai iiikki, even ki iaiikii analogue of the non-writable key DS1990A, working only with original DS1990 keys manufactured by Dallas
Semiconductor, such as DS1990C.

To create copies of the key to the Vizit intercoms, it is recommended to use a special blank TM08-Vizit, TM-08 Vizit (ÿ2)
or TM-08 Vizit F.

Table of supported formats blanks

FLIP-iButtonblankstypes...

Oh imin

Types of iButton keys

iButton—this is just the physical form factor of the key and reader: round tablet with two contacts. At the same time, seemingly identical iButton keys can work on different protocols and contain different microcircuits.

This section contains different types of keys with descriptions. We try to collect here the most complete database of keys.

Name	Description
	The most popular iButton key. Not rewritable. Company keys have laser engraved with key ID. This key ID can be entered manually on flipper according to the format (link)

RW1990, RW1990.1	

Writable keys (blanks)

0min

Applications

4min

File storage

So, create your own application for Flipper Zero. Let's see where and how it should store its files.

1. All files required by the application must be stored inside the application folder.
2. Files generated by the application have their own extension (ibutton => .ibtn, RFID -> .rfid) and are stored in the application's folder root.
3. Files used by the application (application assets such as pictures that your application displays) are stored in the assets subfolder.

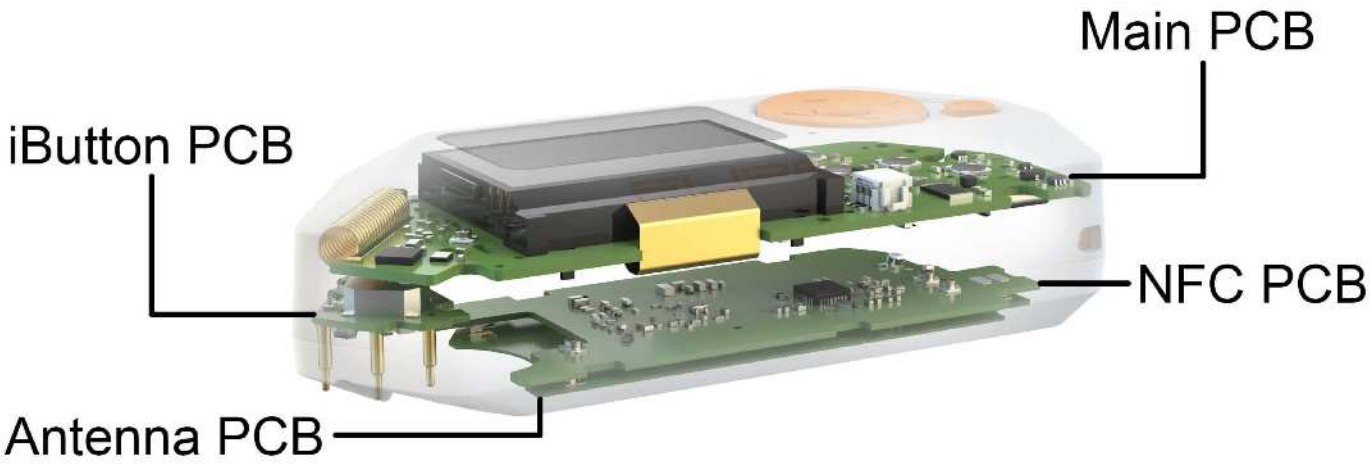
For example, let's take a look at the "Infrared" application:

- Application folder: `irda`.
- Remotes are saved in a folder, for `irda` with extension `.ir`, example `irda/My_tv.ir`.
- Databases are in a subfolder `assets`, for example `assets/tv.ir`.

7 minutes

iron

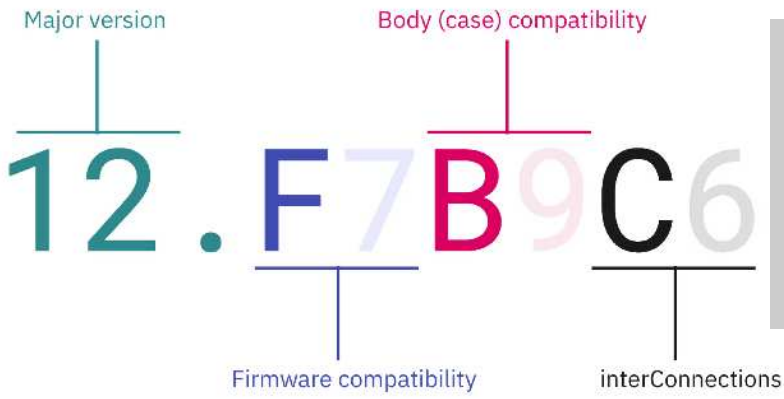
Schematicdiagrams Flipper Zeroversion12.F7B9C6



Flipper Zero consists of 4 boards

Here are the schematicdiagramsofall FlipperZeroboards.Theywill be useful for module developers and for low-level debugging. The diagrams are published as a reference and do not constitute accuratemanufacturing documentation.

How to read the hardware version

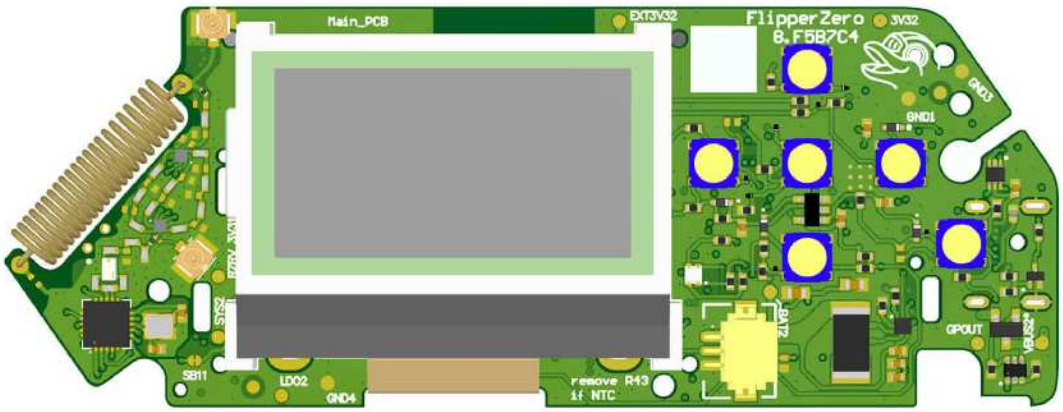


What is the iron version of Flipper Zero formed from

- Thefirstdigitbeforethedot is the serial number of the version. Incrementedwith eachmodification.
- F <number> - Firmware Target. Firmware compatible. It is incremented only in cases where the new hardware revision is no longer compatible with the old firmware.
- B<number>—Body, compatible with plastic body. Incremented if the new boards are no longer compatible with the old chassis.
- C<number>—Interconnect. Compatibility with cables connecting boards. Incremented if the pinout of the loops

Main fee

Main board with display, buttons and screen.



Main fee

PDF file (Main board)

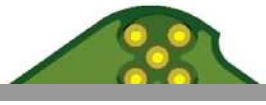
FLPZ_12.F7B9C6_Main_S...

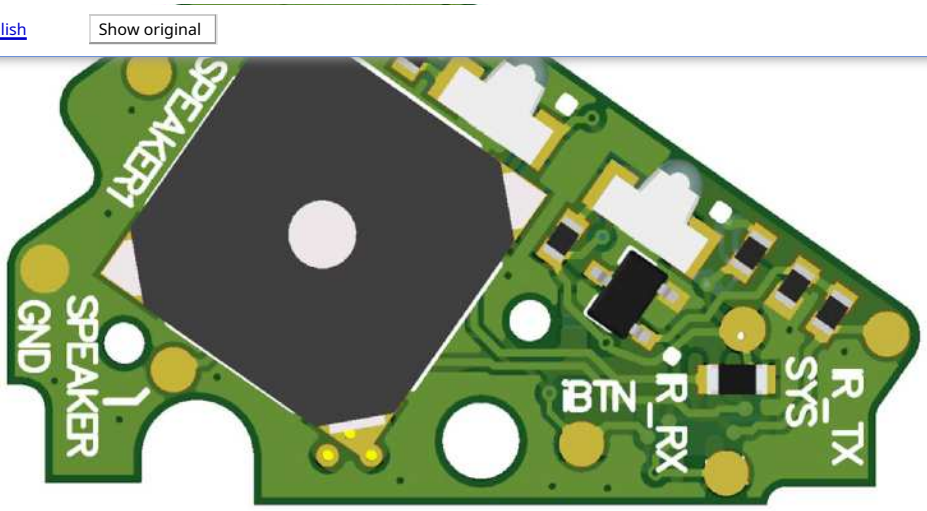
Live view (Main board)

Interactive Altium Schematic Viewer. Switch pages in the upper left corner.

iButton board

The iButton board contains iButton pogo pins, piezo dynamics, and an infrared port.





iButton board

PDF File (iButton Board)

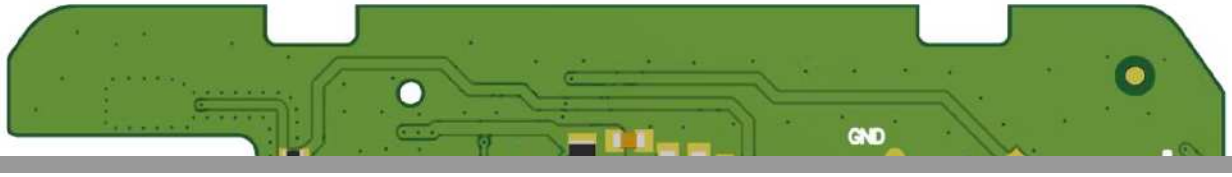
FLPZ_12.F7B9C6_iButton...

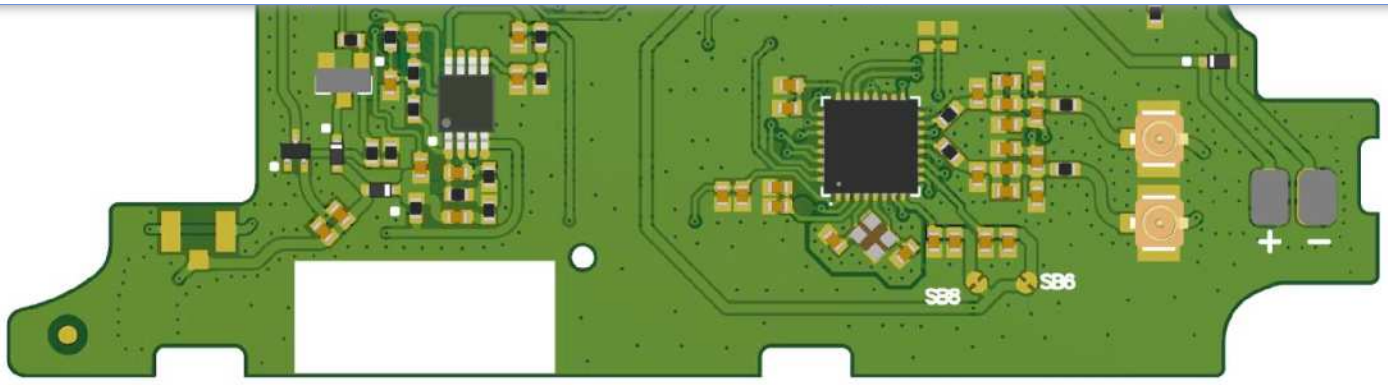
Live view (iButton board)

Interactive Altium Schematic Viewer. Switch pages in the upper left corner.

NFC card

The board contains 125 Hz RFID and NFC modules.





NFC card

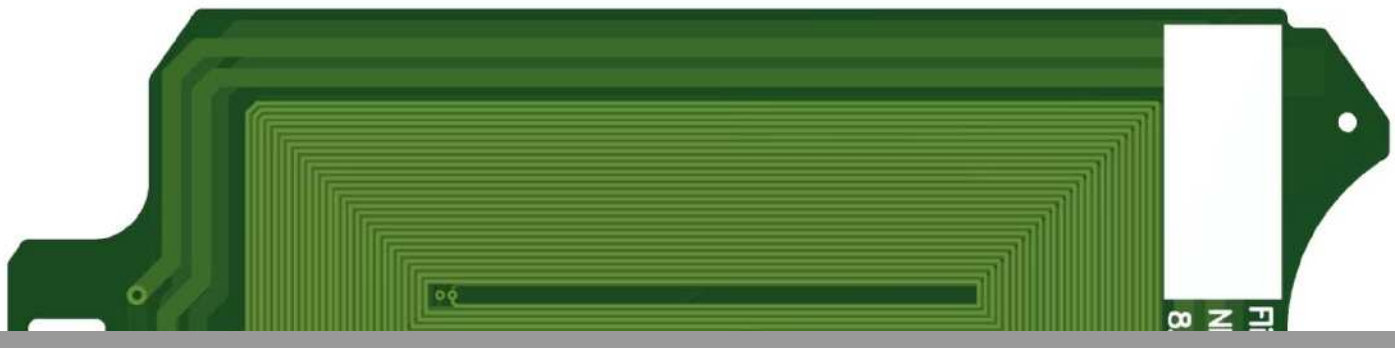
PDF file (NFC card)

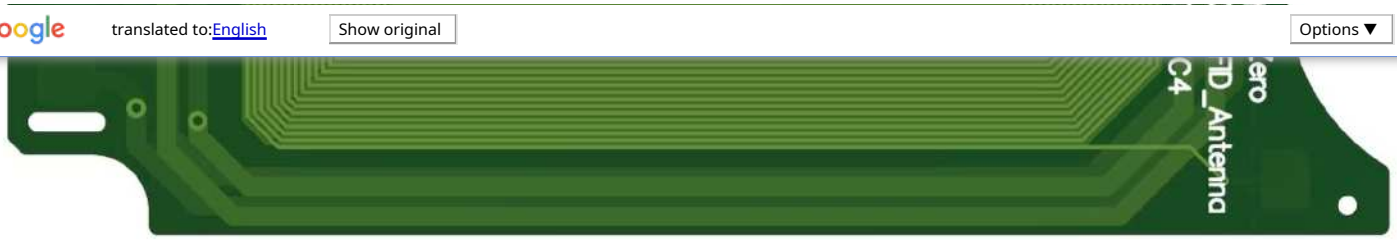
FLPZ_12.F7B9C6_NFC_Sc...

Live view (NFC card)

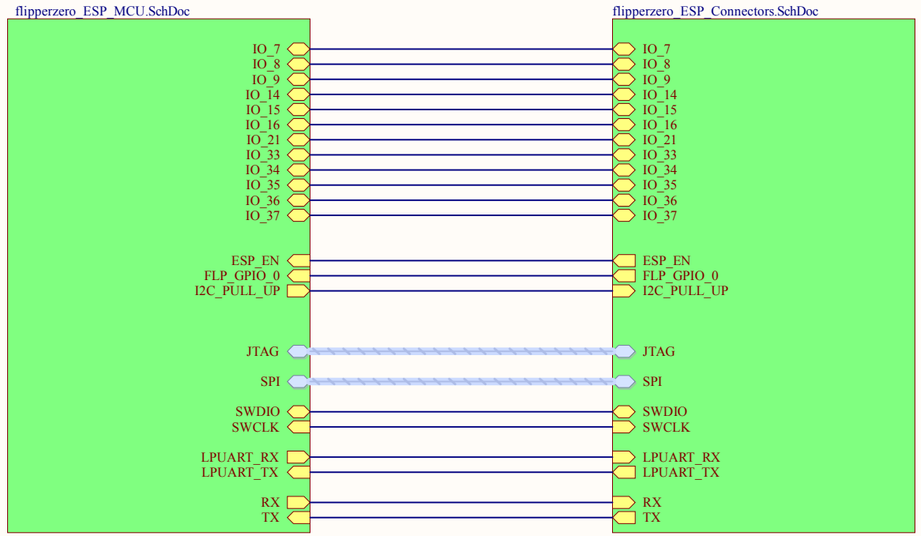
Interactive Altium Schematic Viewer. Switch pages in the upper left corner.

RFID antenna board





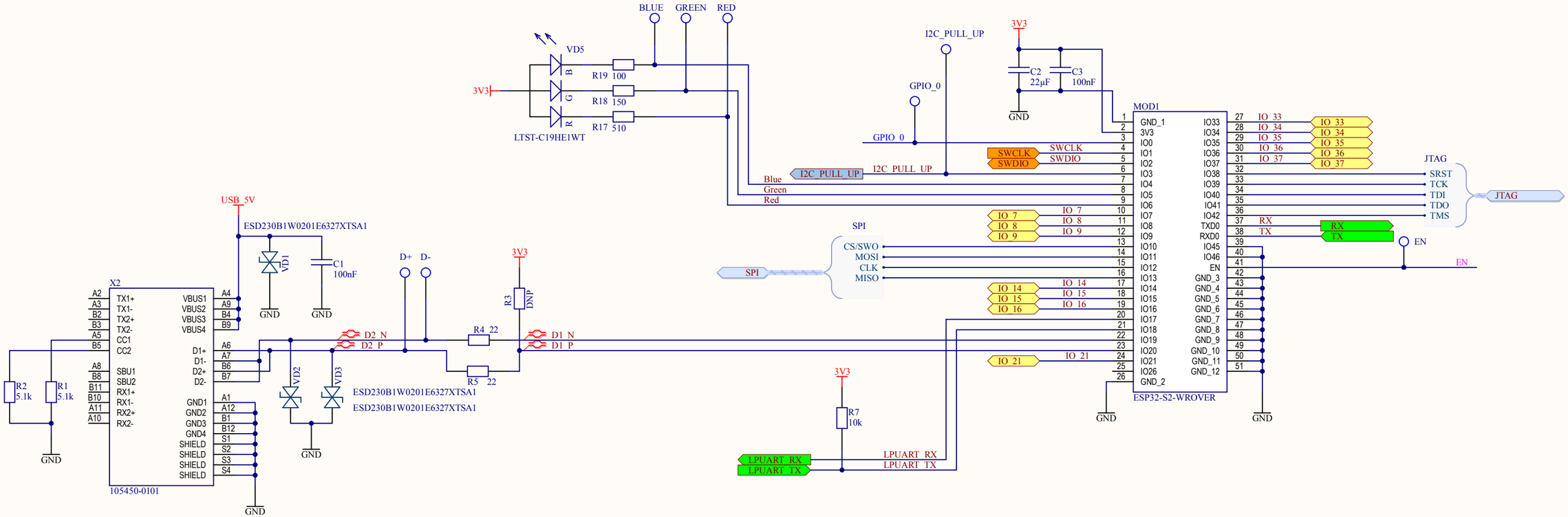
RFID 125khz and NFC Antenna Board

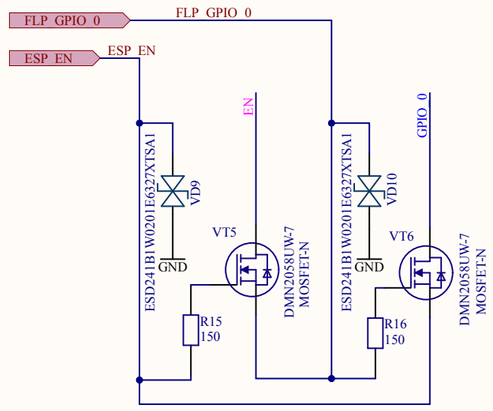


flipperzero_ESP_Power.SchDoc

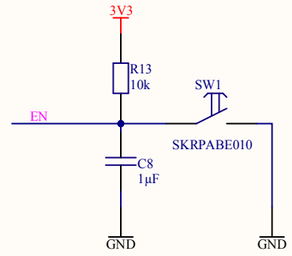
フリッパーデバイス

Title:		*Flipper Devices Inc.* *2803 Philadelphia* *Pike Suite B #551* *Claymont, DE* *19703, USA*
Project:	Rev.:	
Date: 12/10/2021	Size: A3	Sheet 1 of 4

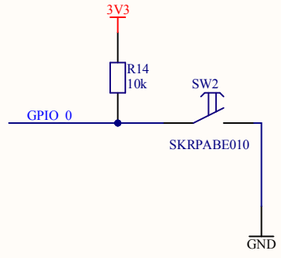




RESET

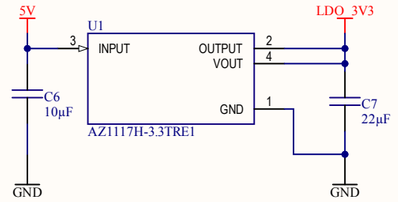


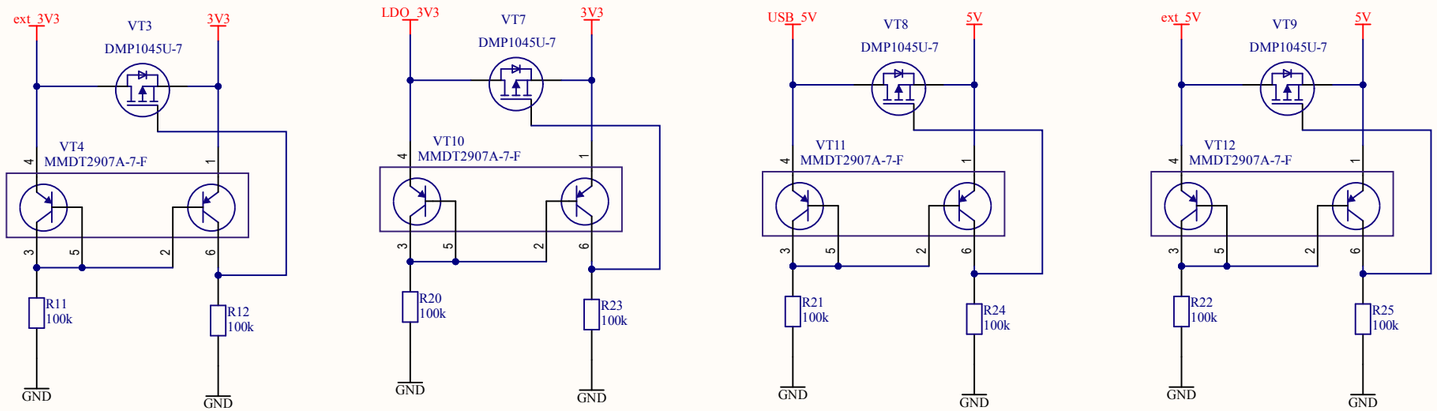
BOOT



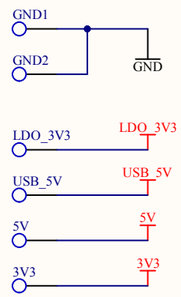
フリッパーデバイス

Title:		Flipper Devices Inc. 2803 Philadelphia Pike Suite B #551 Claymont, DE 19703, USA
Project:	Rev.:	
Date: 12/10/2021	Size: A3	Sheet 2 of 4



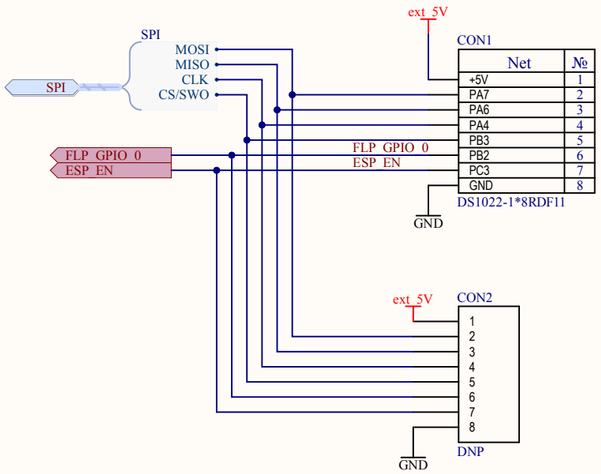


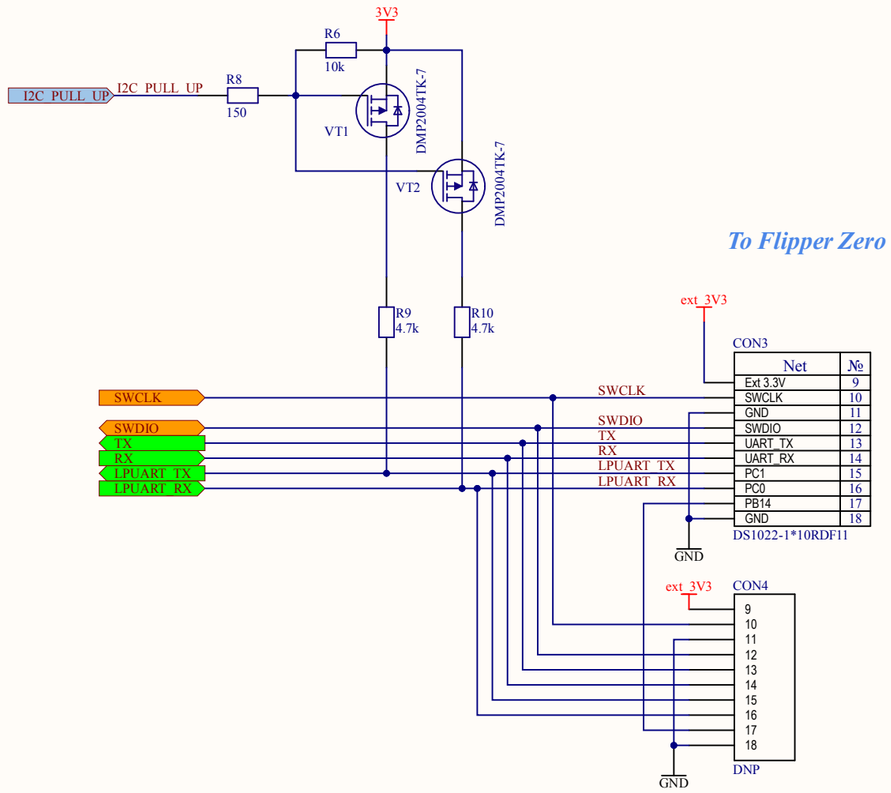
Testpoints

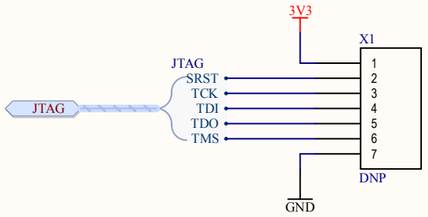


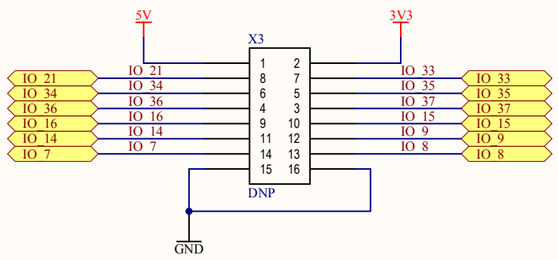
フリッパーデバイス

Title:		*Flipper Devices Inc.*
Project:	Rev.:	*2803 Philadelphia Pike Suite B #551*
Date: 12/10/2021 Size: A3	Sheet 3 of 4	*Claymont, DE 19703, USA*









フリッパーデバイス

Title:		*Flipper Devices Inc.* *2803 Philadelphia* *Pike Suite B #551* *Claymont, DE* *19703, USA*
Project:	Rev.:	
Date: 12/10/2021	Size: A3	Sheet 4 of 4

2min

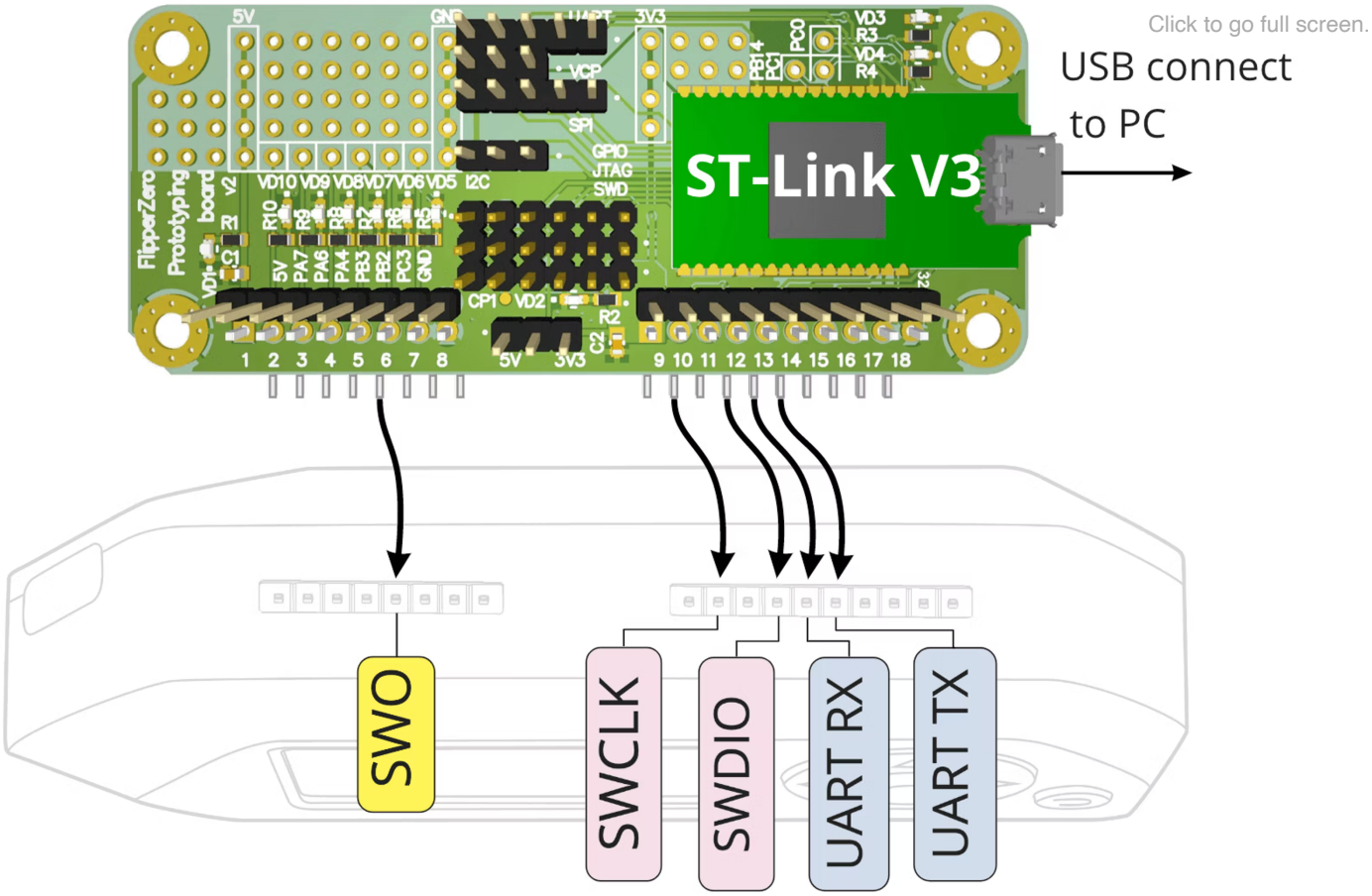
Hardware

Flipper Ziro's UIR port has a special dark eye - it blocks interference from visible light and transmits infrared radiation from remotes. This helps isolate the useful IR signal and remove visible light glare. It is this filter that we used to see in all infrared ports. Behind him are the receiver and transmitter elements:

- ₽ Re dayu SCHeIR light dand about ds - to increase power used immediately 3
- Ғeating Rreceiver TSOP-75338 - fancy IR receiver, filtering and, if necessary, amplifying the signal

5min

ST-LinkV3 debugger



A debug board for advanced developers who need in-circuit debugging of their programs. Built on the basis of the usual **ST-Link V3 Mini** , **it differs only in form factor** and ease of connection. Additionally, unused ST-Link and GPIO Flipper Zero interfaces are brought to the board.

Debug board is not needed for Flipper Zero firmware

You can update firmware, develop and upload your firmware to Flipper Zero via USB without a development board! A debug board is needed for in-circuit debugging of running programs, for example, through GDB, OpenOCD. If you don't know exactly how to use it, you don't need this board.

Specifications

- ST-Link V3 Mini for flashing and in-circuit debugging

- Flipper Zero unused GPIO pins for debugging and breadboarding

Scheme and BOM

Project sources in Altium Designer

github.com/Flipper-Zero/flipperzero-devboard-stlinkv3

Partners

Neuron Hackerspace
Design Heroes
Slozhno.Media

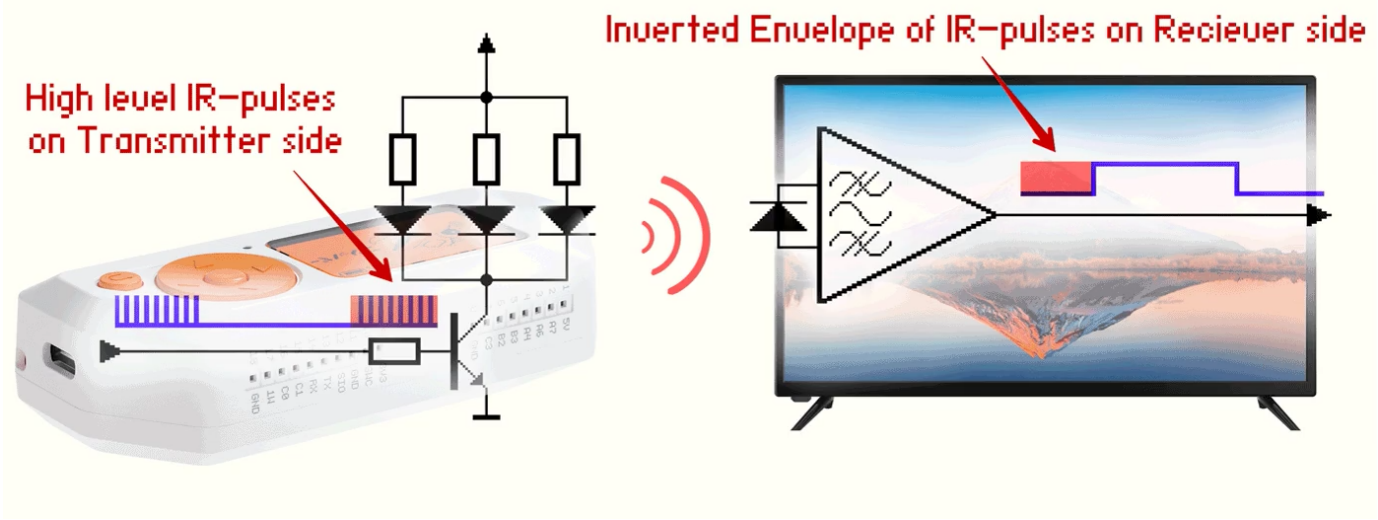
About

Contacts
Company
Careers
Press kit
Privacy Policy
FAQ

5min

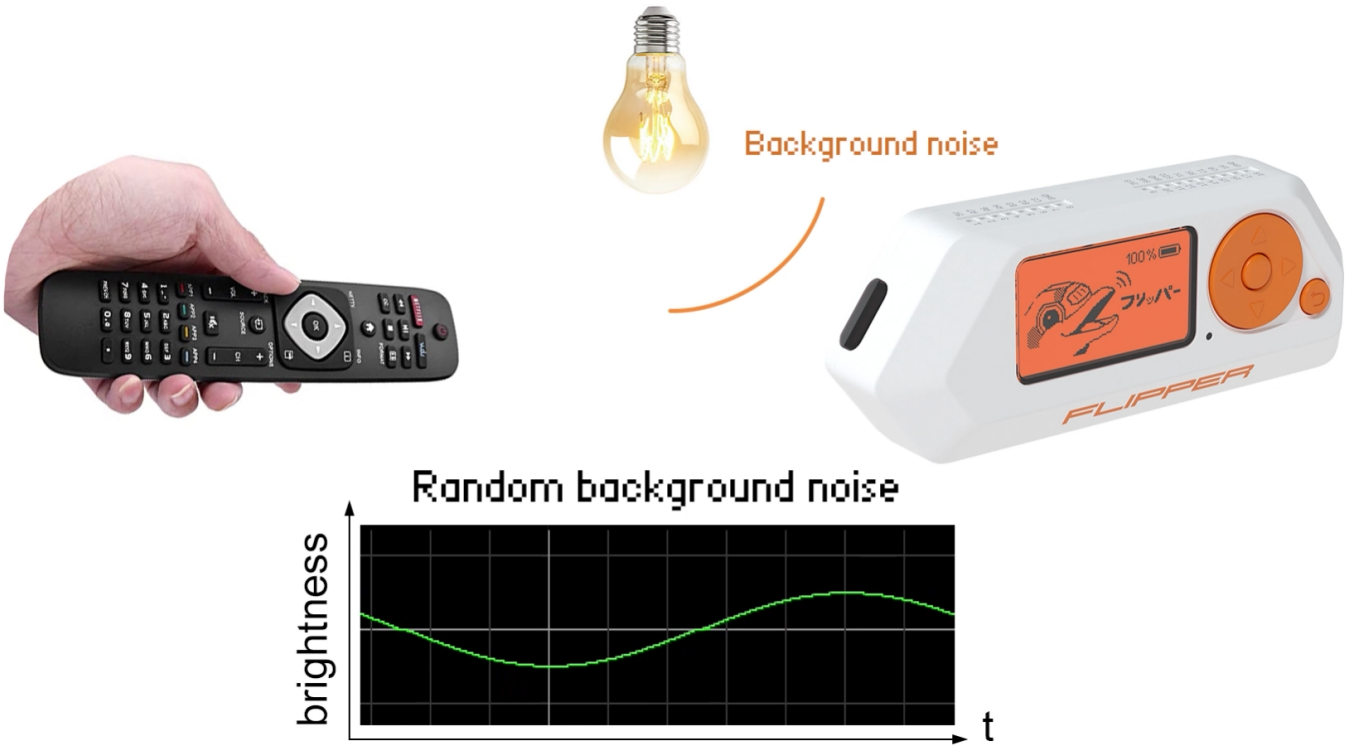
Principle of operation

Infrared radiation is electromagnetic radiation invisible to humans with a wavelength of 0.7 to 1000 microns. Household remote controls use an IR signal for data transmission and operate in the radiation range of 0.75..1.4 microns. The microcontroller in the remote control flashes an infrared LED at a certain frequency - this is how the digital signal becomes an IR signal.



As with remotes, data from Flipper Zero is transmitted in bursts. In the receiver, from the received packets of pulses, the demodulator forms envelopes (meanders) and outputs them to the output. Often the digital signal at the output of the receiver is an inverted envelope. Typically, receivers have a dark photofilter that passes radiation at the right wavelength to filter out interference.

Signal modulation

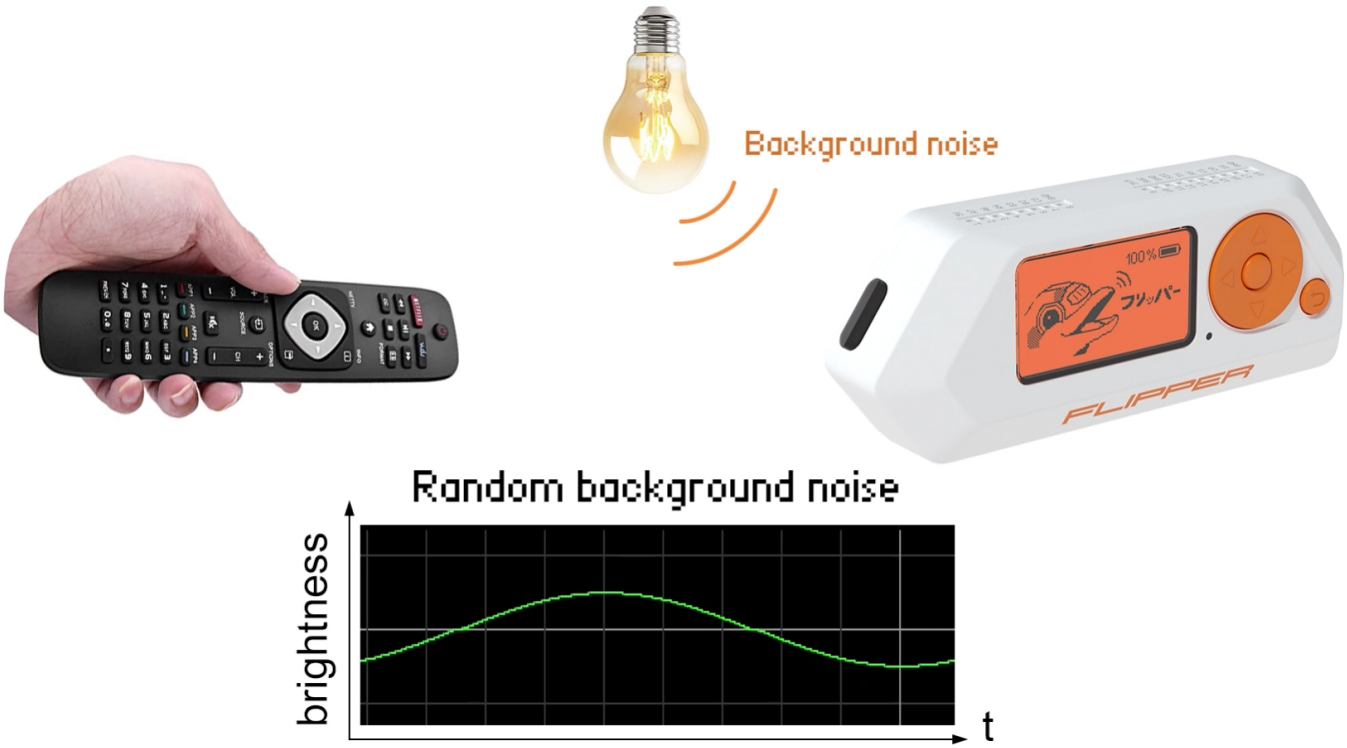


On the IR receiver side, there is almost always background noise because there are many IR emitting objects around, such as ordinary lighting lamps. Therefore, the total signal from the noise and the useful signal arrives at the receiver.

- Noise in the IR range is created by many light sources, since the source of IR radiation is generated heat. Therefore, the background noise will be random. In the gif above, for clarity, it is depicted as a sine wave.
- The useful signal is packets of IR pulses sent by the remote control. An ideal burst of pulses looks like a flat square wave. But such a signal can be seen only in the complete absence of noise. In reality, the meander will always be superimposed on the noise and summed up with it.

Frequency modulation allows you to distinguish between an IR signal with data and noise. When the useful IR signal flashes at a certain frequency, then the IR frequency ripples are visible against the background of non-pulsating radiation. Thus, the photodetector can judge the presence of a signal and distinguish it from light.

The Flipper Zero IR receiver is designed to demodulate a signal with a carrier frequency of 38 kHz. Most consoles operate on a carrier frequency of 36-38 kHz.

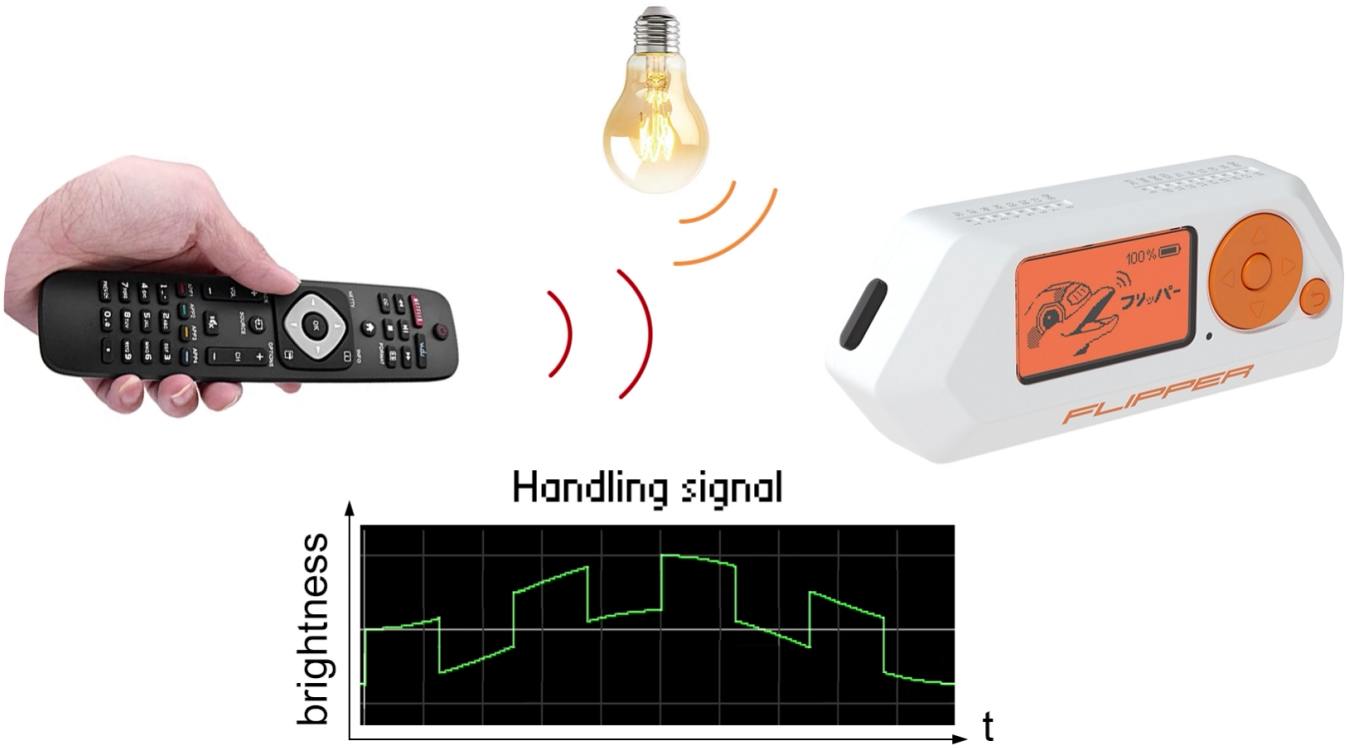


On the IR receiver side, there is almost always background noise because there are many IR emitting objects around, such as ordinary lighting lamps. Therefore, the total signal from the noise and the useful signal arrives at the receiver.

- Noise in the IR range is created by many light sources, since the source of IR radiation is generated heat. Therefore, the background noise will be random. In the gif above, for clarity, it is depicted as a sine wave.
- The useful signal is packets of IR pulses sent by the remote control. An ideal burst of pulses looks like a flat square wave. But such a signal can be seen only in the complete absence of noise. In reality, the meander will always be superimposed on the noise and summed up with it.

Frequency modulation allows you to distinguish between an IR signal with data and noise. When the useful IR signal flashes at a certain frequency, then the IR frequency ripples are visible against the background of non-pulsating radiation. Thus, the photodetector can judge the presence of a signal and distinguish it from light.

The Flipper Zero IR receiver is designed to demodulate a signal with a carrier frequency of 38 kHz. Most consoles operate on a carrier frequency of 36-38 kHz.

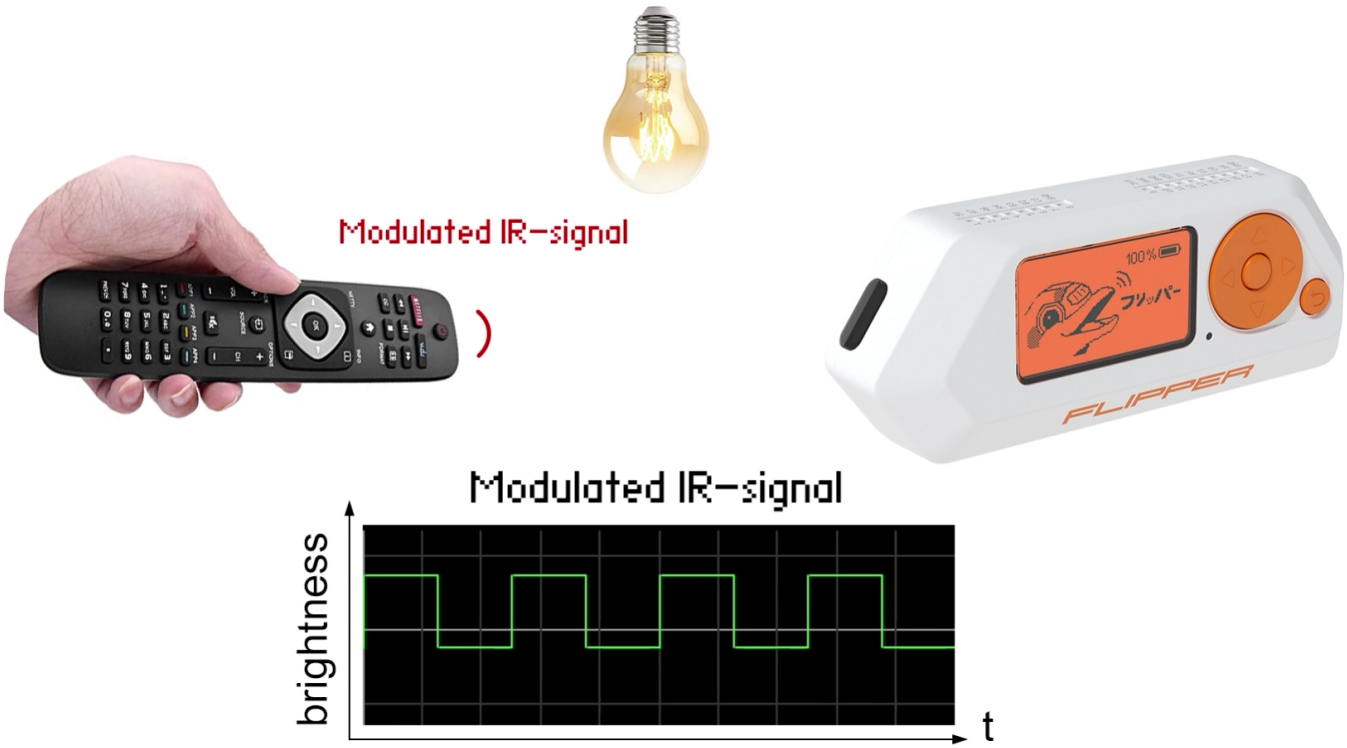


On the IR receiver side, there is almost always background noise because there are many IR emitting objects around, such as ordinary lighting lamps. Therefore, the total signal from the noise and the useful signal arrives at the receiver.

- Noise in the IR range is created by many light sources, since the source of IR radiation is generated heat. Therefore, the background noise will be random. In the gif above, for clarity, it is depicted as a sine wave.
- The useful signal is packets of IR pulses sent by the remote control. An ideal burst of pulses looks like a flat square wave. But such a signal can be seen only in the complete absence of noise. In reality, the meander will always be superimposed on the noise and summed up with it.

Frequency modulation allows you to distinguish between an IR signal with data and noise. When the useful IR signal flashes at a certain frequency, then the IR frequency ripples are visible against the background of non-pulsating radiation. Thus, the photodetector can judge the presence of a signal and distinguish it from light.

The Flipper Zero IR receiver is designed to demodulate a signal with a carrier frequency of 38 kHz. Most consoles operate on a carrier frequency of 36-38 kHz.



On the IR receiver side, there is almost always background noise because there are many IR emitting objects around, such as ordinary lighting lamps. Therefore, the total signal from the noise and the useful signal arrives at the receiver.

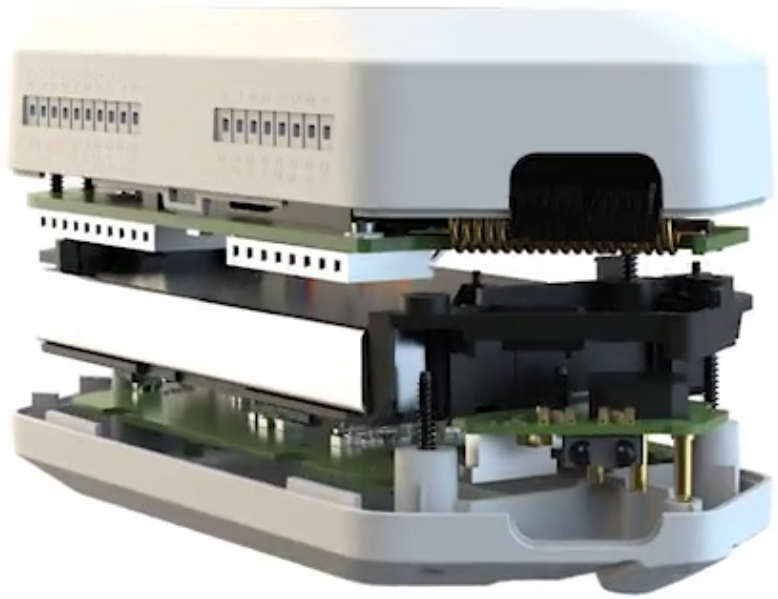
- Noise in the IR range is created by many light sources, since the source of IR radiation is generated heat. Therefore, the background noise will be random. In the gif above, for clarity, it is depicted as a sine wave.
- The useful signal is packets of IR pulses sent by the remote control. An ideal burst of pulses looks like a flat square wave. But such a signal can be seen only in the complete absence of noise. In reality, the meander will always be superimposed on the noise and summed up with it.

Frequency modulation allows you to distinguish between an IR signal with data and noise. When the useful IR signal flashes at a certain frequency, then the IR frequency ripples are visible against the background of non-pulsating radiation. Thus, the photodetector can judge the presence of a signal and distinguish it from light.

The Flipper Zero IR receiver is designed to demodulate a signal with a carrier frequency of 38 kHz. Most consoles operate on a carrier frequency of 36-38 kHz.

5min

Hardware

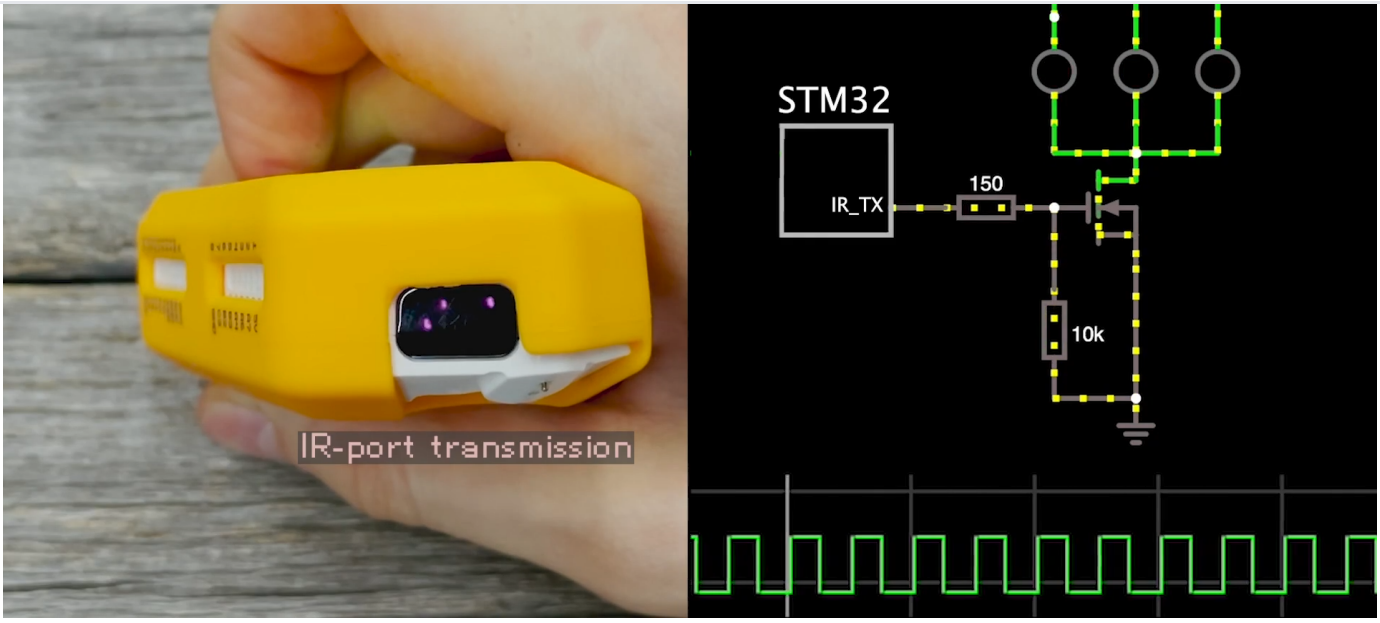


Flipper Zero's IR port has a special dark window - it blocks visible light interference and transmits IR radiation from remotes. This helps to isolate the useful IR signal and remove visible light glare. It is this filter that we are used to seeing in all infrared ports. Behind it are already located the elements of the receiver and transmitter:

- **Transmitting IR LEDs** - 3 are used at once to increase the power
- **Photodetector TSOP-75338** - a sophisticated IR receiver that filters and, if necessary, amplifies the signal

5min

Transmitting IR LEDs



The transmission of the IR signal is directly controlled by the Flipper STM32 microcontroller. Through an external transistor, it sends pulses to the LEDs - so the digital signal becomes an IR signal. To increase the power of the IR transmitter, 3 LEDs are used at once instead of one.

To increase the pulse power of the transmitter (transmission range), data is transmitted in bursts of pulses, and not in whole meanders. Due to this, the average power decreases or remains the same, and therefore the energy consumption decreases or remains the same.

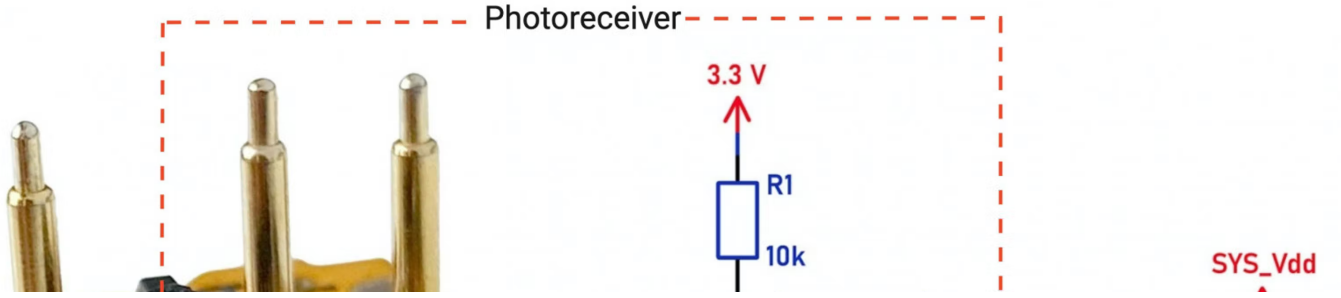
Basically, transmitters operate with carrier frequencies of 30..50 kHz. This range of carrier frequencies during the development of the first transmitters had the lowest level of interference for the available element base. Not to be confused with the frequency of the IR radiation itself, which corresponds to a wavelength of 940 nm (318.93 THz).

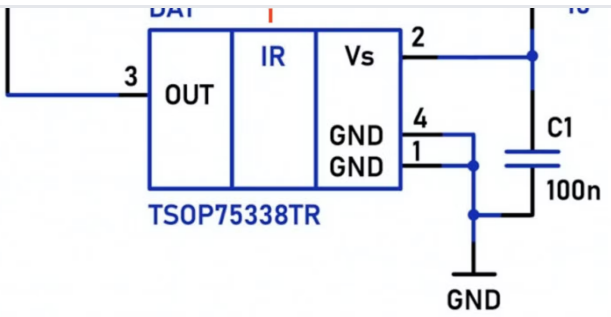
5min

Photodetector TSOP-75338

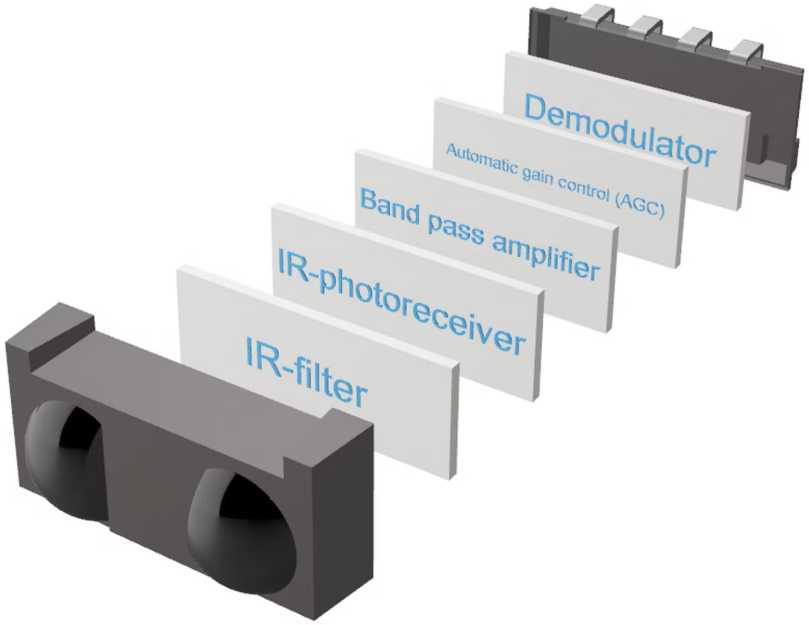
TSOP-75338 photodetector used in Flipper Zero to receive IR signal

Flipper's IR receiver, the TSOP-75338 chip, filters the signal and keeps it at the same logic level, amplifying if necessary. Therefore, the TSOP-75338 is able to receive a weak signal from discharged remotes or reflected from walls. This simplifies the software processing of the signal on the processor side.





The power circuit of the TSOP-75338 photodetector has an RC filter. It is needed, since the microcontroller creates interference on the power lines, due to which the digital signal at the output of the photodetector may not correspond to the received signal. A diode is used to match the levels of the receiver-TSOP and the STM32 microcontroller. At the TSOP output, the STM32 microcontroller already processes the digital signal.



Functionally, the TSOP-75338 IR receiver includes:

- IR filter
- IR photodetector
- Amplifier with a filter for a specific carrier frequency
- Amplifier with automatic control
- Demodulator-detector extracting the envelope

carrier frequency

The TSOP-75338 is designed to operate at a carrier frequency of 38 kHz. He does not know how to recognize the carrier frequency and duty cycle when receiving a signal. When saving the signal, these parameters are set by default: carrier frequency 38 kHz, duty cycle 33%. To transmit a signal with other parameters, you need to generate a RAW format signal. Information about the type of RAW signal can be found **HERE**

ÿ 3min

Изв---notable protocols

The transmitted code is:

- **static** - the same code is sent every time
- **dynamic -** the code changes every transmission. Usually the change algorithm is unknown, but there are exceptions

Dynamic protocols have internal encryption keys. They are are set by the manufacturer and are not disclosed.

Protocols are not tied to a specific frequency and modulation type. The same protocol may use different frequency and modulation depending on the manufacturer.

Protocol name	Stat./Din.
Princeton	Stat.
Bytec	Stat.
Tantos-Proteus	Stat.
GSN	Stat.
Nice Flo 12/24 bit	Stat.
CAME 12/24 bit	Stat.
Gate TX	Stat.

Taac_sin	Din.
Nice Flor-S	Din.
KeeLoq	Din.
DoorHan	Din.
AM-Motors	Din.
Stilmatic	Din.
HCS101	Din.
Alligator D-810, Alligator D-930	Din.
Alligator S-750RS	Din.
Alligator S-275	Din.
Alligator NS, NS-105, NS-205, NS-305, NS-405, NS-505, NS-605	Din.
Alligator M-550, M-500	Din.
Alligator L330	Din.
Pantera SLR-5100	Din.
Pantera CLK-355	Din.
Pantera SLK-2i, SLK-2i/3i/4i/5i/7i, SLK-25SC	Din.
Pantera CL-500, CL400, CL600	Din.
Pantera XS-1500, XS-2000, XS-1000, XS-1700, XS-100, XS-110	Din.
Pantera XS-2600, XS-2700	Din.
Jaguar JX-1000, XS-2700	Din.

iz-yu iu, bL-ÿ, U- / UU, u-yuu, 5- / UU	
Guard RF-311A	Din.
Septah A-90	Din.
Sheriff ZX-600	Din.
Sheriff APS-35 PRO	Din.
Sheriff APS-25 PRO	Din.
Sheriff APS-2400	Din.
Sheriff ZX-925, ZX-900, ZX-910, APS-75	Din.
Mongoose 800C, IQ-215	Din.
Mongoose 7000 RF, AMG-850C	Din.
Leopard LS50/10	Din.
Partisan RX-1	Din.
APS 3000, 2550, 2450	Din.
APS 2300, 2500, 2000,1500,1000, 500	Din.
StarLine	Din.
Cenmax ST-5A, Cenmax Vigilant V-5A	Din.
Cenmax ST-7A, Cenmax Vigilant V-7A	Din.
KGB FX-5	Din.
Tomahawk 9030, TW-9030, TW-7010, TW-9020, TZ 7010, TZ-9020, TZ-9030, HI, H2	Din.
Tomahawk Z5, Z3, ÿÿ, X5	Din.
rl inn ÿ A 7 ÿ A	P m

StatLine B6, B9	Din.
Harpoon BS-2000	Din.
Jaguar EZ-Beta	Din.

5min

Send button command

Here we consider the protocols that send the code of the pressed button. This is how TVs, media centers, etc. work. Usually, the transmitted data includes:

- management team
- device address
- verification information
- any other service information

There are IR protocols that are trying to become universal for several types of equipment. The most famous are the formats: RC6 and NEC.

But more often, hardware manufacturers use their own IR protocols, even within the same types of equipment (eg TVs). Therefore, often remotes from different manufacturers, and sometimes from different models of the same manufacturer, cannot work with other devices of the same type.

At the moment, Flipper knows the following IR protocols for TVs, media centers, etc.:

- **NEC**
- **NECext**
- **RC6**
- **Samsung32**

5min

NEC

Describe what the signal consists of.

Say what encoding of a bit of information

Show what the signal looks like (general view for everyone, the oscillogram is not suitable, since people often do not have it)

What data is passed (Addr, Cmd)

What is the template for entry in the dictionary of universal remotes

The NEC IR protocol contains a short command to be sent and a retry code to be sent if the button is still pressed. Both the command and the repeat code have the same preamble at the beginning.

A command in NEC, in addition to the preamble, consists of an address byte and a command-number byte. By the command-number, the device understands what exactly needs to be done. The bytes of the address and number-command are duplicated with inverse values to check the integrity of the transmission. At the end of the command, there is an additional stop bit.

The **repeat code** after the preamble contains a logical "1" - stop bit.

Logical “0” and “1” in the NEC protocol are determined by intervals: first, a burst of pulses is transmitted, after which there is a pause that sets the value of the bit.

5min

NECext

Describe what the signal consists of.

Say what encoding of a bit of information

Show what the signal looks like (general view for everyone, the oscillogram is not suitable, since people often do not have it)

What data is passed (Addr, Cmd)

What is the template for entry in the dictionary of universal remotes

5min

RC6

Describe what the signal consists of.

Say what encoding of a bit of information

Show what the signal looks like

What data is being transferred

What is the template for entry in the dictionary of universal remotes

5min

Samsung32

Describe what the signal consists of.

Say what encoding of a bit of information

Show what the signal looks like

What data is being transferred

What is the template for entry in the dictionary of universal remotes

5min

Transfer all settings

Remote controls from air conditioners, unlike other equipment, do not transmit the code of the pressed button, but all the parameters of the air conditioner that are visible on the screen of the remote control. That is, they always send ALL the data displayed on the remote control screen.

Expected

So far, there are no IR protocols for air conditioners in Flipper's firmware

2min

Debugging

Here are the technical solutions that we used during the development and debugging of the infrared port in Flipper Zero.

Arduino IRMP

Library of IR protocols and instructions for assembling an IR analyzer

Silver Bullet

Infrared probe for oscilloscope

AnalysIR

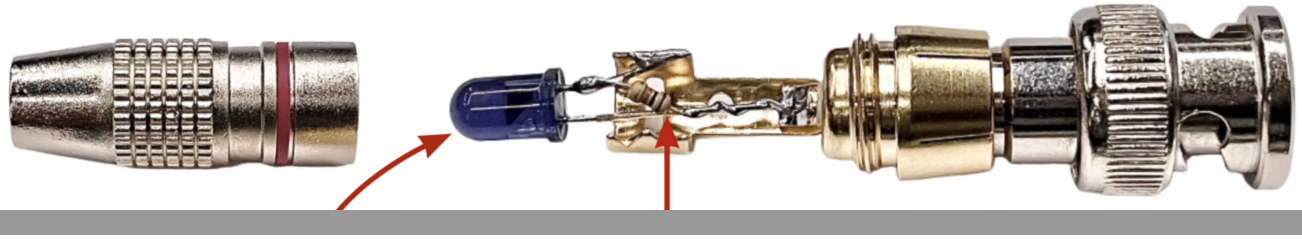
Recommended PC software for IR signal analysis

5min

Sliver Bullet

Oscilloscope recording remote IR signal with Silver-bullet

To capture IR pulses on an oscilloscope, we use a homemade probe **Silver Bulletinvented** by the author of the **AnalysIR** program . This is a conventional IR LED and a resistor soldered into an RCA audio plug, which is connected via a BNC->RCA adapter to the oscilloscope. Assembles in five minutes.



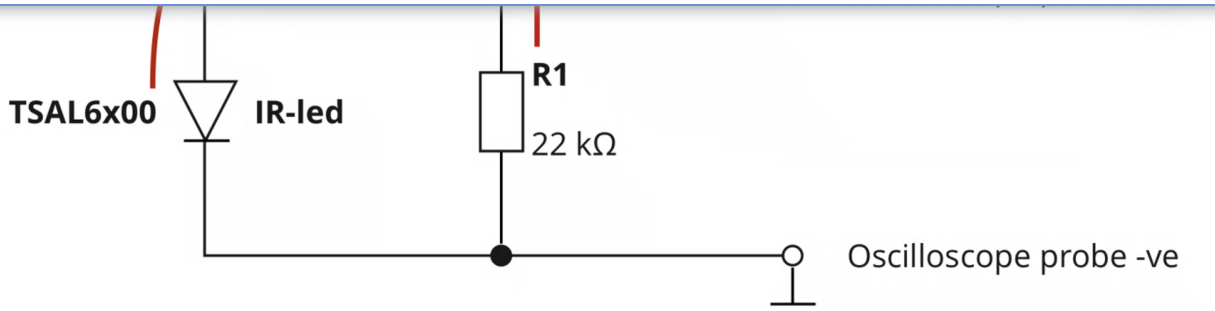
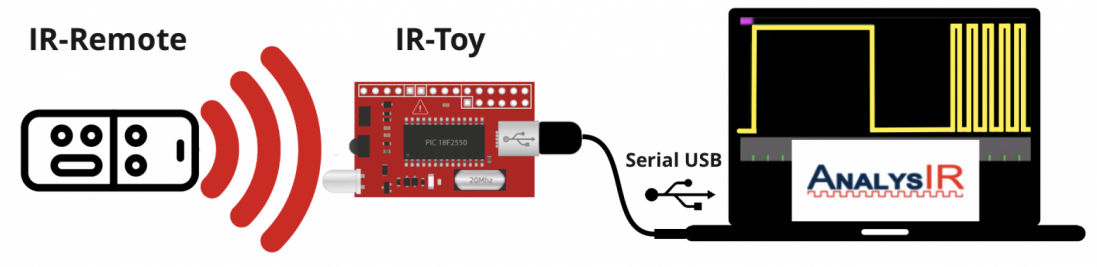


Diagram of the probe for capturing the IR signal on the oscilloscope

When the IR light from the remote control hits the probe's IR LED, a small current flows through it. This current creates a voltage difference across the LED pins, which is clearly visible on an oscilloscope. To get a clear signal on the oscilloscope, it is important that the transmitter leans closely against the probe.

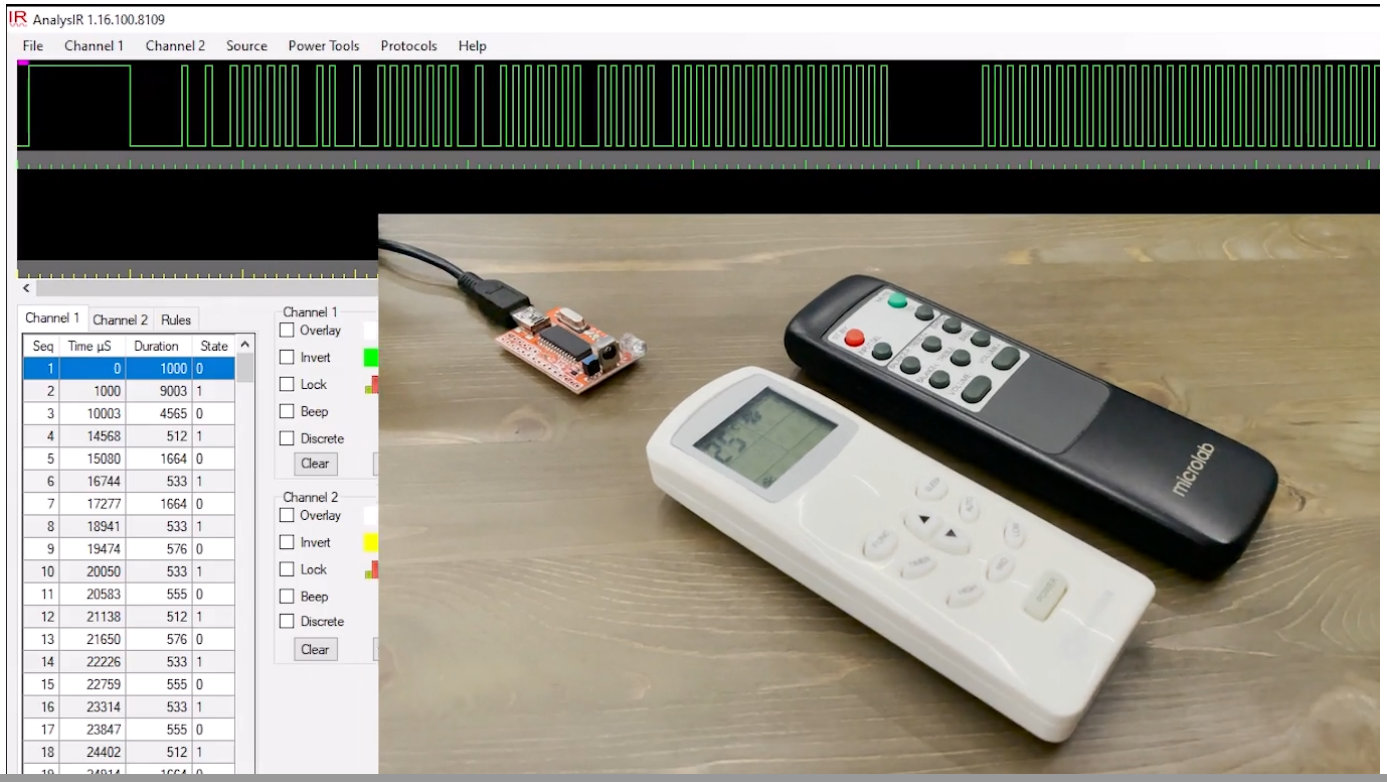
5min

AnalysIR



Scheme of using the AnalysIR program with the IR-Toy equipment

a receiver. List of supported receivers: **AnalysIR.pdf** .



signal. The program calculates the delays and durations of bursts of pulses - all this is recorded in a log and helps to analyze unknown IR protocols. AnalysIR knows more than 100 IR protocols and can automatically recognize them.

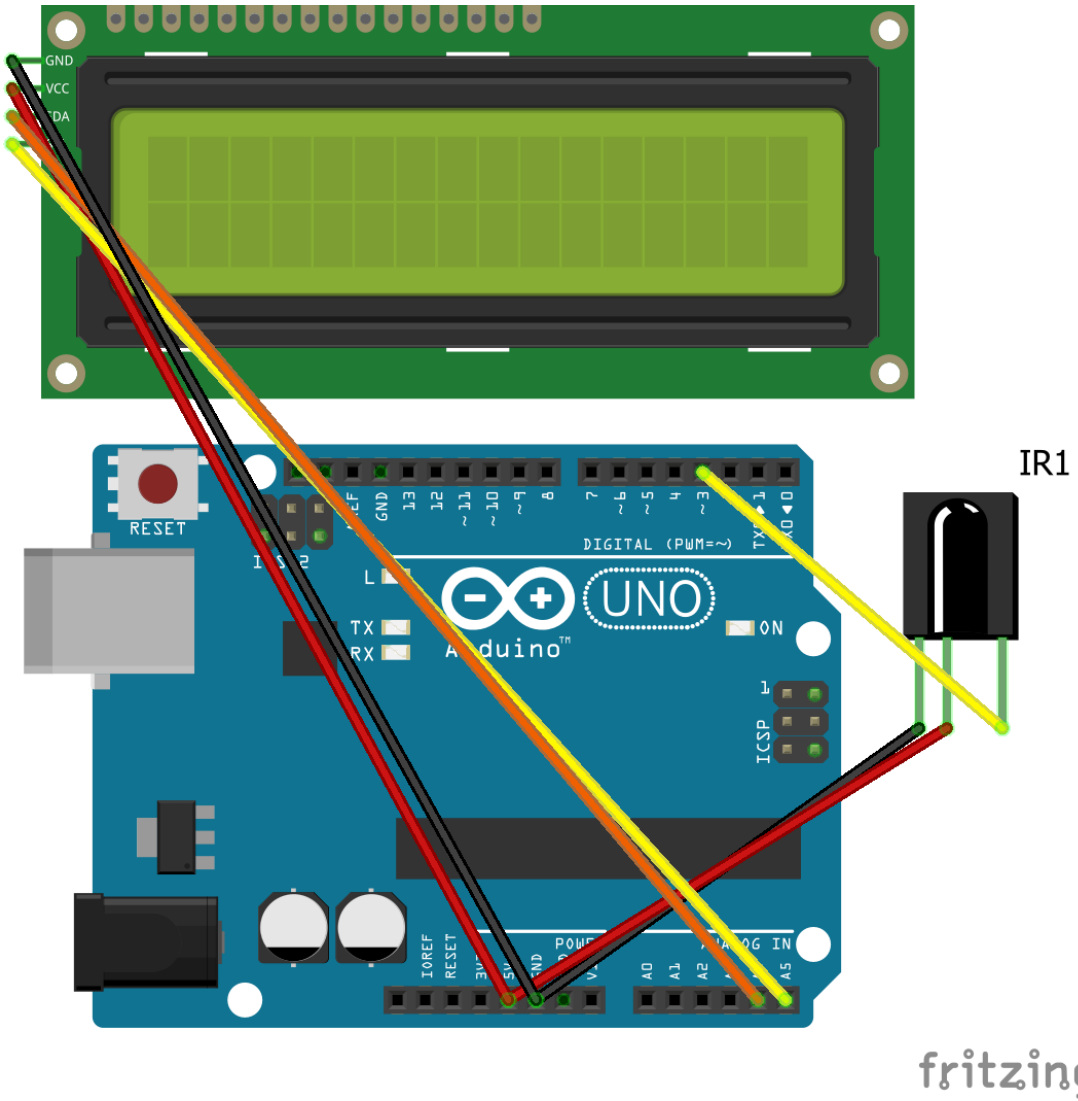




5min

Arduino IRMP

To quickly test and debug IR protocols, we use the **IRMP** library from Arduino. On **the github** you can find instructions on how to assemble a device for analyzing IR protocols.



Schematic diagram of an IR protocol analyzer based on Arduino IRMP

You cannot unconditionally trust the assembled analyzer. If the IR protocol is unknown, then the Arduino IRMP analyzer can recognize it as the Siemens protocol. You can use your photo

5min

File system

Flipper's file storage consists of internal and external memory. Access to this memory is implemented by the storage service using the following directories:

- **/int -** access to internal Flash memory
- **/ext -** access to the external memory of the SD card
- **/any -** SD card if available, otherwise Flash memory. Used in applications that store little data.

The storage service is used to copy and move files between directories.

CLI

The storage service is available on the command line via the `storage` command . Errors that occur are displayed using a single pattern `Storage error: <error message>\r\n` . All error messages are in "applications/storage/filesystem-api.c". The <path> argument can be entered with or without quotes (eg "/ext/System Volume Information"). `The storage` subcommands are documented below .

info

`storage info <path>` - Gets information about the selected storage. The <path> argument must be either "/int" or "/ext". Output syntax: `Label: <name>\r\nType: <type>\r\n<memory size>KB total\r\n<free memory>KB free\r\n` , data is in kilobytes.

```
>: storage info /int
Label: Unknown
Type: LittleFS
136 KB total
124 KB free

>: storage info /ext
Label: FLIPPER SD
Type: FAT16
1992192 KB total
1992191 KB free

# SD card missing
>: storage info /ext
Storage error: filesystem not ready

>:
```

`storage format <path>` - formats storage <path>. Requests confirmation of an action. Formatting is only implemented for /ext.

```
>: storage format /int
Storage error: function not implemented

>: storage format /ext
Formatting SD card, all data will be lost. Are you sure (y/n)?
Cancelled.

>: storage format /ext
Formatting SD card, all data will be lost. Are you sure (y/n)?
Formatting, please wait...
SD card was successfully formatted.

>:
```

list

`storage list <path>` - displays a list of files and directories located in <path>. Синтаксис вывода: `\t[тип]<имя> <размер>b\r\n`. Исключение - пустая папка `\t Empty`) Тип `[D]` для папок, `[F]` для файлов. Размер в байтах указывается только для файлов.

```
>: storage list /
        [D] int
        [D] ext
        [D] any

>: storage list /int
        [F] notification.settings 24b

>: storage list /ext
        [D] 123
        [F] 123.txt 10b

>: storage list /ext/123
        Empty

# Сд карта отсутствует
>: storage list /ext
Storage error: internal error

>:
```

remove

`storage remove <path>` – удаляет файл или директорию <path>. Директория должна быть пустой.

```
>: storage remove /ext/123
Storage error: access denied

>: storage remove /ext/123/file.txt

>: storage remove /ext/123

>:
```

read

`storage read <path>` – выводит в консоль байтовый размер и данные файла <path>. Синтаксис вывода: `Size: <размер файла>\r\n<данные>\r\n`.

```
>: storage read /ext/newfile.txt
Size: 40
12345678901234567890123456789012345678901234567890

>: storage read /ext/not_existing_file.txt
Storage error: file/dir not exist

>:
```

read_chunks

`storage read_chunks <path> <chunk_size>` – выводит в консоль байтовый размер и данные, блоками размером <chunk_size> байт. Синтаксис вывода: `Size: <размер>\r\n`, затем на каждый блок `Ready?\r\n<ожидание любого символа><данные>`.

Размер блока не больше 512 байт

Флиппер использует оперативную память, чтобы вычитать блок из файла и отправить его. Не используйте большие блоки, разумный размер блока - 512 байт.

```
>: storage read_chunks /ext/newfile.txt 9
Size: 40

Ready?
123456789
Ready?
012345678
Ready?
901234567
Ready?
890123456
```

```
>: storage read_chunks /ext/nfile.txt 9
Storage error: file/dir not exist

>:
```

write

`storage write <path>` – добавляет текст из консоли в файл <path>. Остановка ввода производится нажатием Ctrl+C.

```
>: storage write /ext/irda
Just write your text data. New line by Ctrl+Enter, exit by Ctrl+C.
Storage error: access denied

>: storage write /ext
Just write your text data. New line by Ctrl+Enter, exit by Ctrl+C.
Storage error: invalid name/path

>: storage write /ext/clean_file.txt
Just write your text data. New line by Ctrl+Enter, exit by Ctrl+C.
12345

>: storage write /ext/clean_file.txt
Just write your text data. New line by Ctrl+Enter, exit by Ctrl+C.
67890

>: storage read /ext/clean_file.txt
Size: 10
1234567890

>:
```

write_chunk

`storage write_chunk <path> <chunk_size>` – добавляет текст из консоли в файл <path>. Останавливается после записи <chunk_size> байт. Перед отправкой данных требуется дождаться вывода `Ready\r\n`.

Размер блока не больше 512 байт

Флиппер использует оперативную память, чтобы вычитать блок из файла и отправить его. Не используйте большие блоки, разумный размер блока - 512 байт.

```
>: storage write_chunk /ext/irda 5
Storage error: access denied
```

```
Ready

>: storage write_chunk /ext/clean_file.txt 5
Ready

>: storage read /ext/clean_file.txt
Size: 10
1234567890

>:
```

copy

`storage copy <from_path> <to_path>` – копирует файл или папку из <from_path> в <to_path>. Исходная папка должна быть пуста.

```
>: storage copy /ext/dir /ext/dir2

>: storage copy /ext/dir /ext/dir2
Storage error: file/dir already exist

>: storage copy /ext/file.txt /int/file.txt

>: storage copy /ext/file.txt /int/file.txt
Storage error: file/dir already exist

>:
```

rename

`storage rename <from_path> <to_path>` – переименовывает файл или директорию из <from_path> в <to_path>.

```
>: storage rename /ext/dir /ext/dir3

>: storage rename /ext/dir /ext/dir3
Storage error: file/dir not exist

>: storage rename /ext/file.txt /int/file.txt

>: storage rename /ext/file.txt /int/file.txt
Storage error: file/dir not exist

>:
```

mkdir

```
>: storage mkdir /ext/dir

>: storage mkdir /ext/dir
Storage error: file/dir already exist

>:
```

md5

`storage md5 <path>` – подсчитывает md5 хеш файла <path>.

```
>: storage md5 /ext/file.txt
8666683506aacd900bbd5a74ac4edf68

>: storage md5 /ext/file1.txt
Storage error: file/dir not exist

>: storage md5 /ext/dir
Storage error: file/dir not exist

>:
```

stat

`storage stat <path>` – выводит информацию о директории, файле или хранилище <path>. Синтаксис вывода: для корневой директории `Storage\r\n` ; для "/ext", "/int" или "/any" `Storage, <полный размер>KB total, <доступно памяти>KB free\r\n` , данные указаны в килобайтах. Директория выводит сообщение `Directory\r\n` ; файл выводит `File, size: <размер файла>b\r\n` .

```
>: storage stat /
Storage

>: storage stat /ext
Storage, 1992448KB total, 1992128KB free

>: storage stat /int
Storage, 264KB total, 252KB free

>: storage stat /ext/file.txt
File, size: 1b

>: storage stat /ext/dir
Directory

>:
```

Реализация удаленных вызовов через CLI находится в библиотеке "scripts/flipper/storage.py". Ее использует консольная утилита "scripts/storage.py".

Общие рекомендации – после отправления команды дождаться приглашение ввода `>:`. При завершении работы команды тоже дождаться приглашения ввода `>:`, чтобы в буффере порта не было лишних данных.

Папки и файлы могут иметь не-ASCII совместимое имя. Такие файлы/папки нельзя использовать, даже если они видны в списке файлов.

Код

Файловое API Флиппера для написания приложений находится в "applications/storage/storage.h". Для получения ссылки на API используется метод `furi_record_open`. После завершения работы с API объект ссылки нужно закрыть `furi_record_close`:

C++

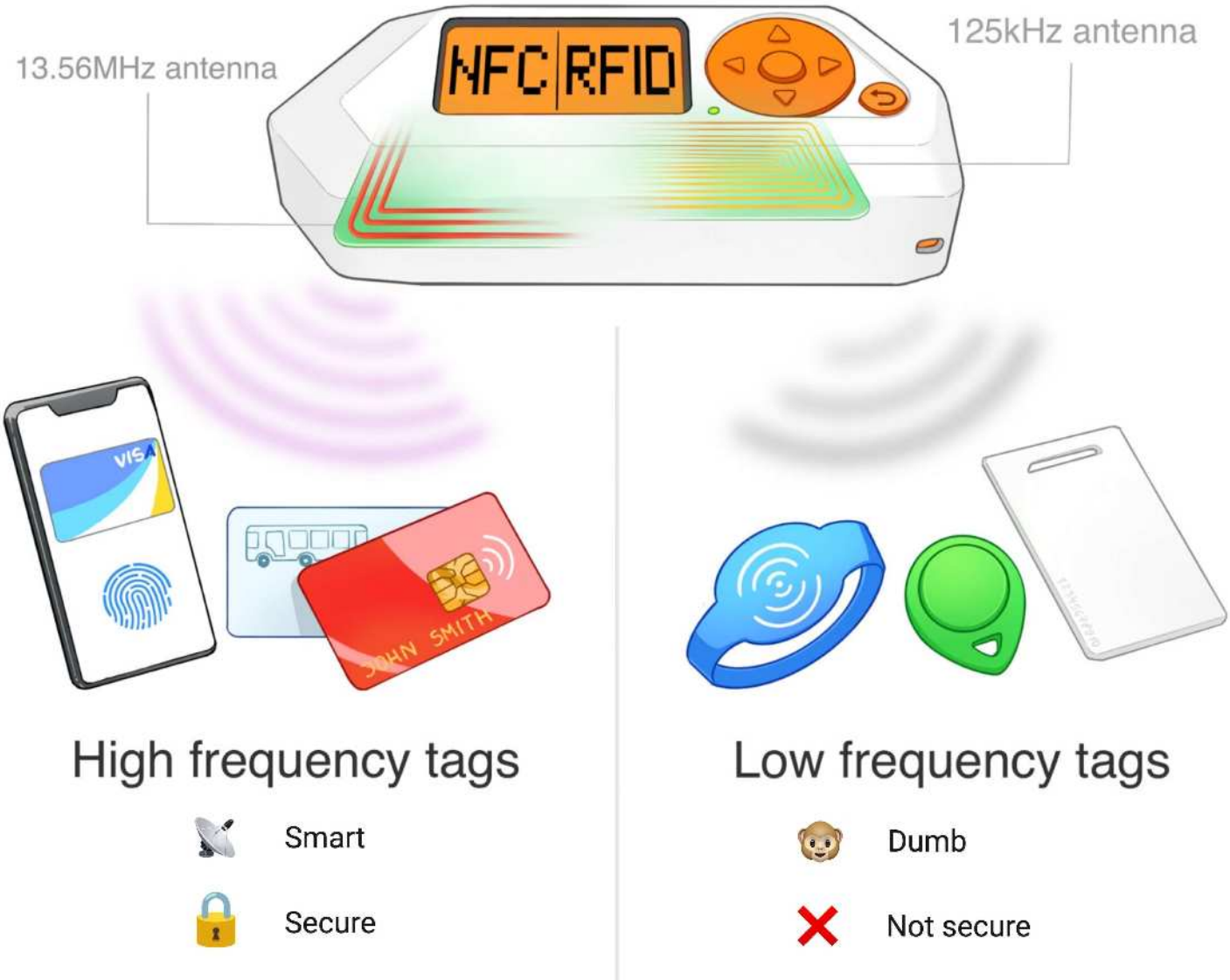
```
Storage* fs_api = furi_record_open("storage");
// do some work
furi_record_close("storage");
```

6min

RFID

RFID is a technology for contactless radio methods used everywhere: in intercoms, payment cards, travel cards, office passes, pets, cars, etc. There are two main types of RFID methods that we use in everyday life:

- Low frequency
- High frequency



Low-frequency (Low Frequency: 125 Hz) - have a long reading range. Unsafe and stupid. Used in primitive access control systems: intercoms, office passes, gym memberships.

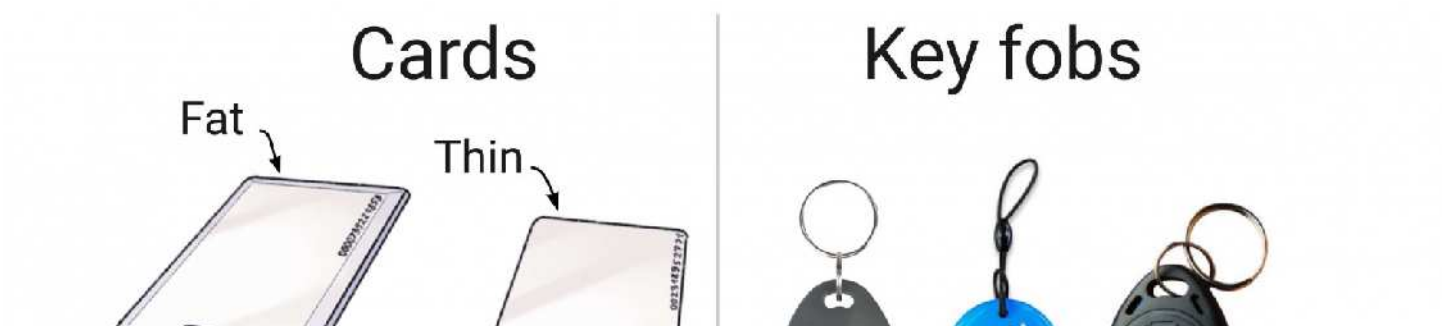
High frequency (High Frequency: 13.56 MHz) - have a shorter operating range compared to low-frequency ones, and may have complex protocols, means of encryption, authentication, cryptography. Used in contactless bank cards, travel tickets, security passes.

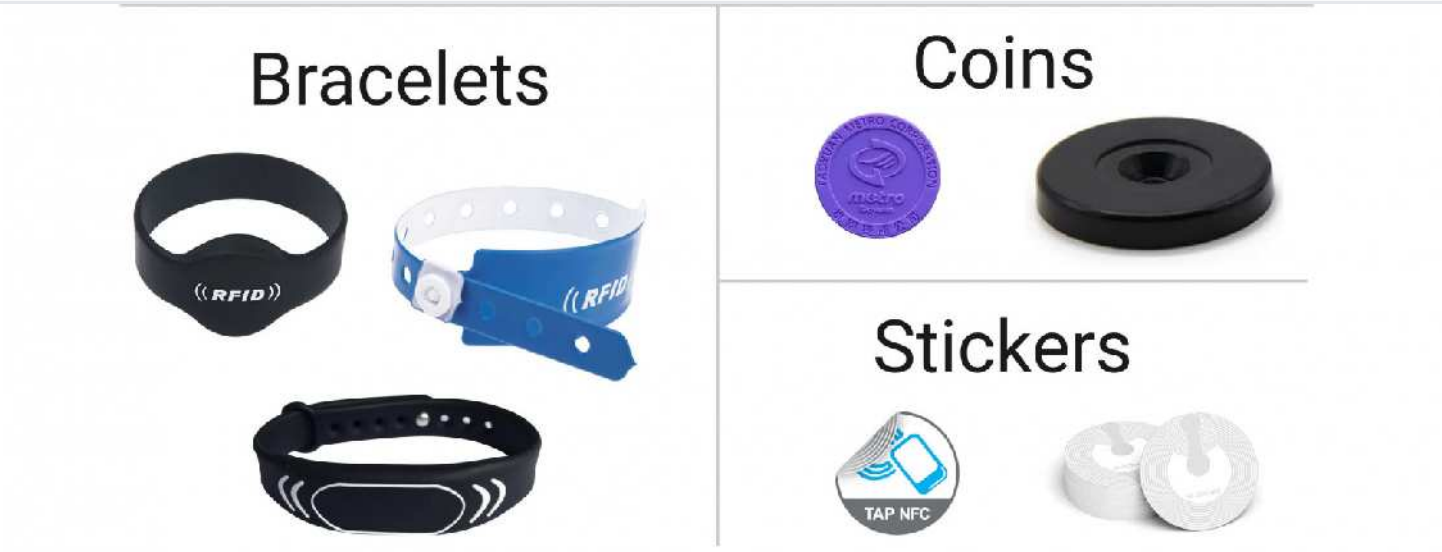
External differences between LF and HF

The appearance of the RFID method can be completely different: thick / thin cards, key chains for intercoms, bracelets, rings, coins, and even stickers. But by one appearance it is impossible to unequivocally say that the metho works at a certain frequency and according to a certain protocol.

visually similar

Often manufacturers of RFID keyfobs use the same plastic housing for both high frequency and low frequency methods. Because of what, two outwardly identical labels can operate at different frequencies.



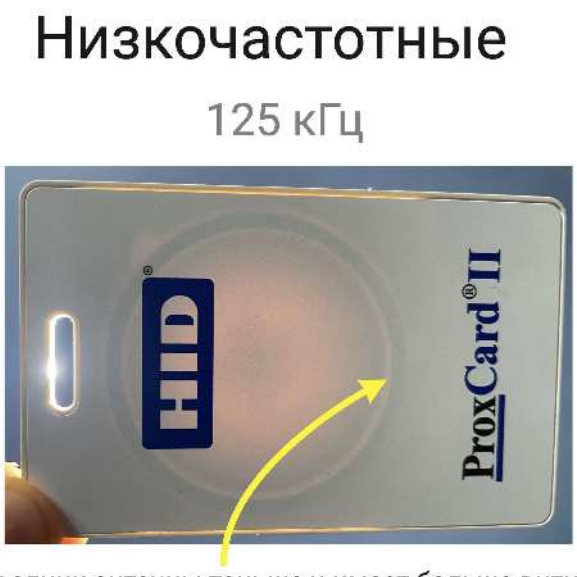


Labels of different forms-factors

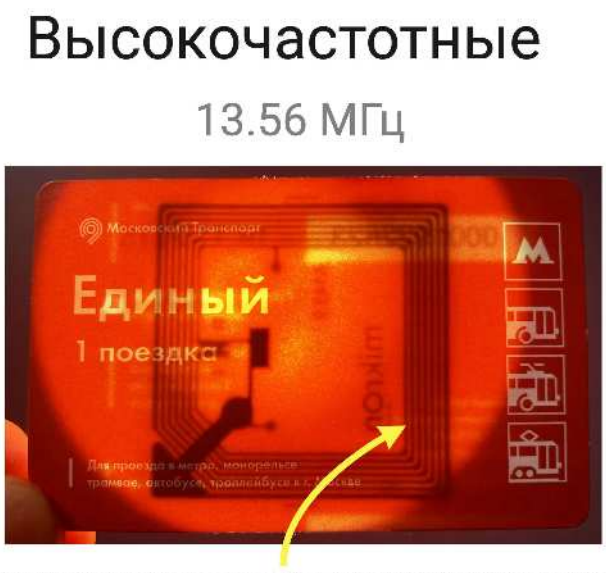
The easiest way to understand the RFID tag in a wide range is by the type of antenna. To see the antenna inside the RFID card, you can illuminate it with a flashlight:

- For low-frequency methods (125 Hz), the antenna is made of very thin wire, literally thinner than a hair, and a huge number of turns. Therefore, such an antenna looks like a solid bar of metal.

- High frequency cards (13.56 MHz) have much fewer turns and thicker wire or tracks. So that gaps are visible between the screws.

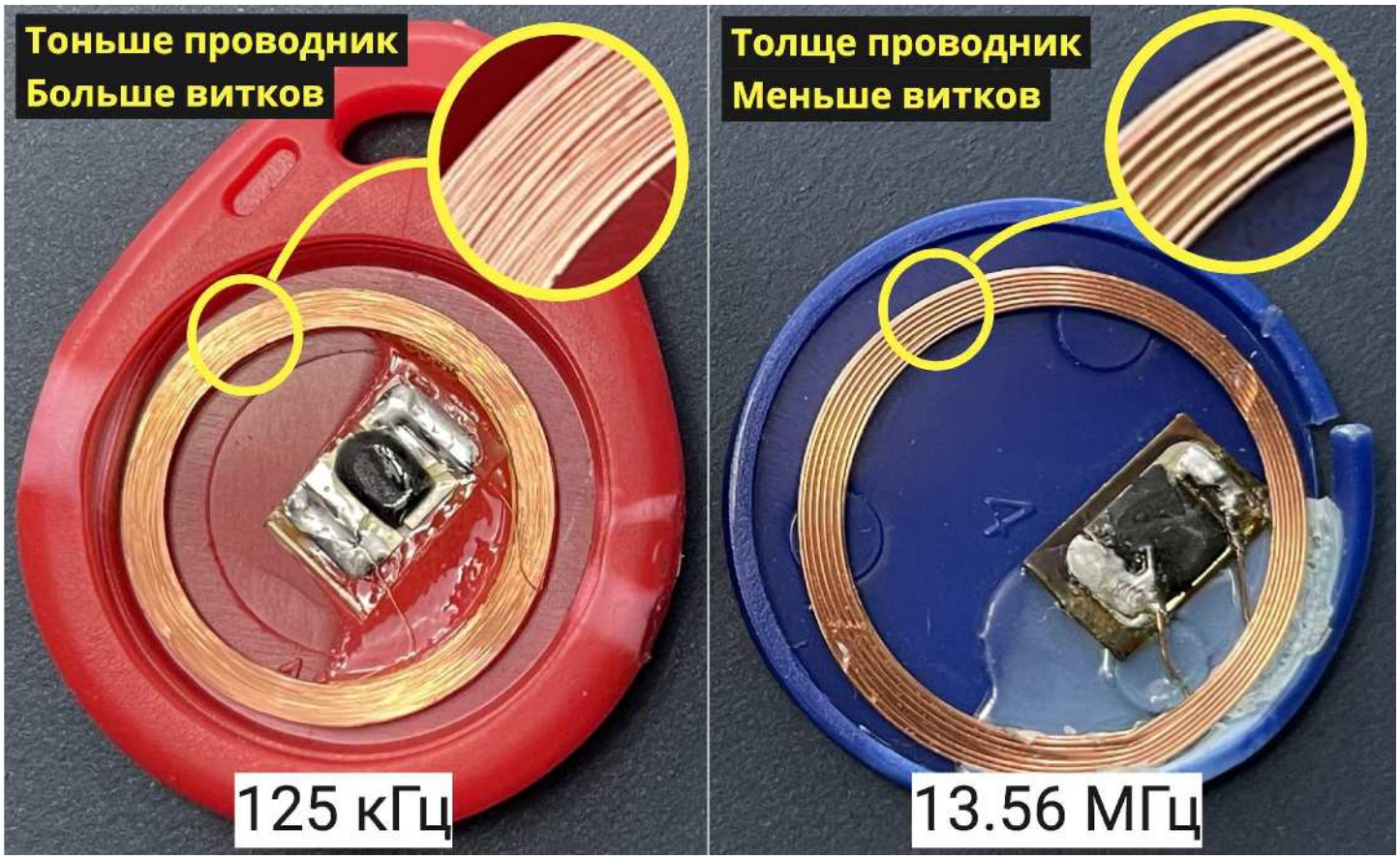


Проводник антенны тоньше и имеет больше витков



Проводник антенны толще и имеет меньше витков

Determining the type of antenna for charts using a flashlight



Difference between wire antennas inside an RFID keyfob

8min

Low Frequency 125kHz

LF RFID Features:

- Range - long range of passive The method is achieved at the expense of low frequency. There are EM-Marin and HID card readers that work at a distance of about a meter. They are often used in car parks.

- Primitive protocol - due to the bottomAt any baud rate, these methods only transmit the short ID. Often, authentication and data protection tools are not used. As soon as the card enters the reader's field of action, it starts transmitting its identifier.

- Low security - lay downHow to copy and read data

Low frequency protocols

Popular types of 125 kHz protocols:

- EM-Marin - usually made on the basis of chipsEM4100 and EM4102 using ASK and FSK modulations. Most common in the CIS

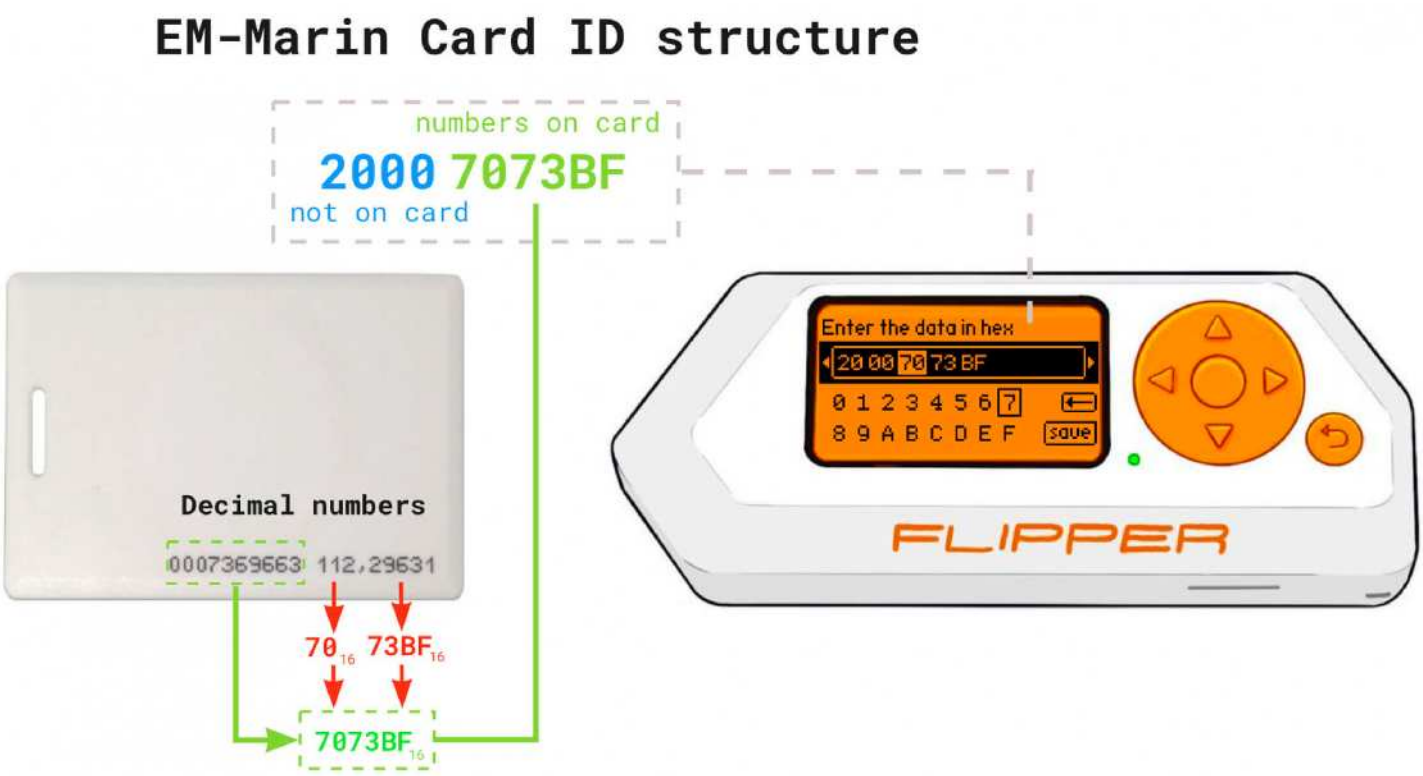
- HID Prox is a popular implementation26-bit protocol H10301 (aka HID26) using FSK modulation. Most common in the west, but also found in the CIS
- Indala is a popular implementation26-bit I40134 protocol using PSK modulation. Modern manufacturers

they use him. But in real life, Indala is still going strong.

There are many more low-frequency protocols, Flipper supports the most common. Flipper can read, save, emulate and overwrite labels.

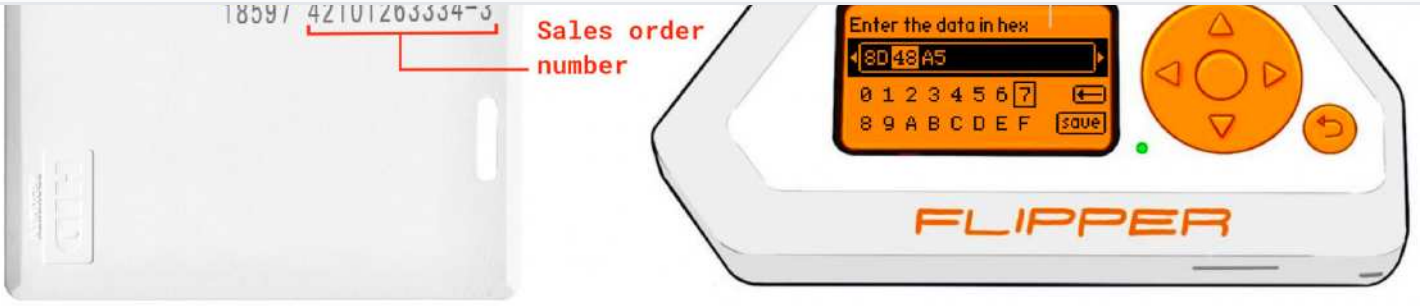
How to use numbers on charts

Low-frequency labels store short ID cards, a few bytes long. Often the card ID is written on its body - it can be added manually to the Flipper. Data in Flipper is stored in hexadecimal format.



Unique EM-Marin Code on Chart and Flipper

The EM4100 unique code consists of 5 bytes and can be written as numbers on the card in several formats at once: decimal text. But on EM-Marin cards, not all 5 bytes are usually written, but only the lower 3 bytes. The remaining 2 bytes will have to be sorted out if it is not possible to read the chart.



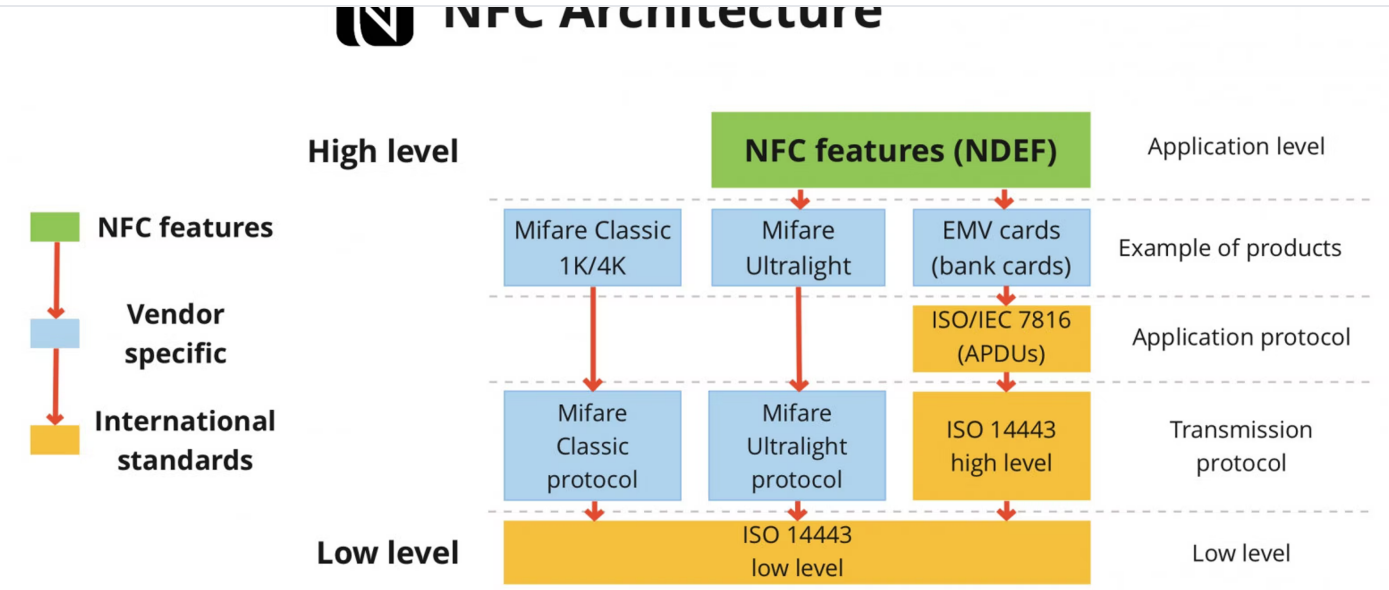
HID26 unique code on card and when read by Flipper

The numbers on the HID26 and Indala cards indicate the lot number and ID card. It is impossible to fully recognize 3 bytes of the unique code by these numbers, only 2 bytes in decimal form are written on the card: Card ID.

5min

NFC High Frequency 13.56 MHz

13.56 MHz high-frequency tags consist of a whole stack of standards and protocols - this entire stack is usually called NFC technology, which is not always correct. The main part of the protocols is based on the ISO 14443 standard - this is a basic set of physical and logical layer protocols on which high-level protocols stand, and based on which alternative low-level standards, such as ISO 18092, are created.



Simplified NFC architecture

Simplified, the NFC architecture looks like this: a transport protocol is implemented on the low-level base of ISO 14443, it is chosen by the manufacturer. For example, NXP came up with its own high-level transport protocol for Mifare cards although at the link level, Mifare cards are based on the ISO 14443-A standard (there may be slight dif general it is).

The flipper is able to interact both with the low level of ISO 14443 protocols, and with the data transfer protocols of Mifare Ultralight and EMV bank cards. We are currently working on adding support for Mifare Classic and NFC NDEF protocols.

More documents on NFC

ISO 14443-A data

1min

ISO 14443-A data

The main ones are 3 values: UID, SAK, ATQA.

All RF cards based on ISO 14443-A have a unique chip identifier - UID. This is the serial number of the card, similar to the MAC address of the network card. UID is 4, 7 and very rarely 10 bytes long. The UID is not read-protected and is not secret, sometimes it is even written on the card.

In reality, there are many ACSs that use UIDs to authorize access. This occurs even when RFID tags are cryptographically secure. In terms of security, this is not much different from stupid low-frequency 125-Hz cards.

SAK and ATQA - card type and manufacturer. It can be assumed from them which high-level protocol is applied.

0min

Mifare Ultralight

Mifare is a family of contactless smart cards that have their own different high-level protocols. Mifare Ultralight is the simplest type of card in the family. In the basic version, it does not use cryptographic protection and has only 64 bytes of internal memory. The flipper supports reading Mifare Ultralight. Такиеметки sometimes use intercom key fobs, passes and travel cards. For example, the Moscow transport tickets "Single" and "90 minutes" were made on the basis of Mifare Ultralight charts.

Operation is now availableReading Mifare Ultrali ght

3min

EMV bank cards

EMV (Europay, Mastercard, and Visa) is an international set of standards
bank cards.

These are full-fledged smart cards with complex data exchange protocols, support for asymmetric encryption. In addition to reading the UID, complex data can be exchanged with a bank card, including pulling out the full card number (16 digits on the front side of the card), the expiration date of the card, sometimes the name of the owner, and even the history of the last orders.

The EMV standard has different high-level implementations, so the data that can be obtained from the charts may differ. CVV (3 digits on the back of the card) can never be counted.

Operation is now available Reading bank EMV RT

Tokenized VS regular bank cards

Apple Pay, Google Pay and other *Pay technologies allow you not to transfer bank card data to the store: PAN, expiration date, owner's name. Instead, digital tokens are used. To work with the token, you need a one-time cryptographic signature, which is generated only on the payer's phone. Outside the phone, creating such a cryptogram will not work.

welded cards are safer

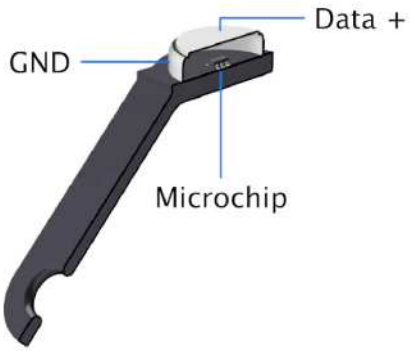
When reading tokenized cards, you cannot get PAN, actions and the owner of the card, so they are not transferred. For For each payment, a token is transferred with a one-time cryptographic signature.

Advantages of tokenized bank cards over regular ones

6min

iButton

iButton is the general name for a dongle in a metal “tablet” form factor. It is often erroneously called a "magnetic" key, but this is incorrect. Inside the iButton is a full-fledged microchip that works according to a digital protocol.



Sectional structure of the iButton key

iButton area structure

Not all "tablets" are intercom keys!

In the "tablet" form factor, there are not only electronic keys, but also climate sensors, devices for storing cryptographic keys. These devices sometimes look like intercom keys, but they are not.

P Roff 1-Wire

Applies to Dallas Keys

Dallas Keys

Dallas key data structure

Metakom keys

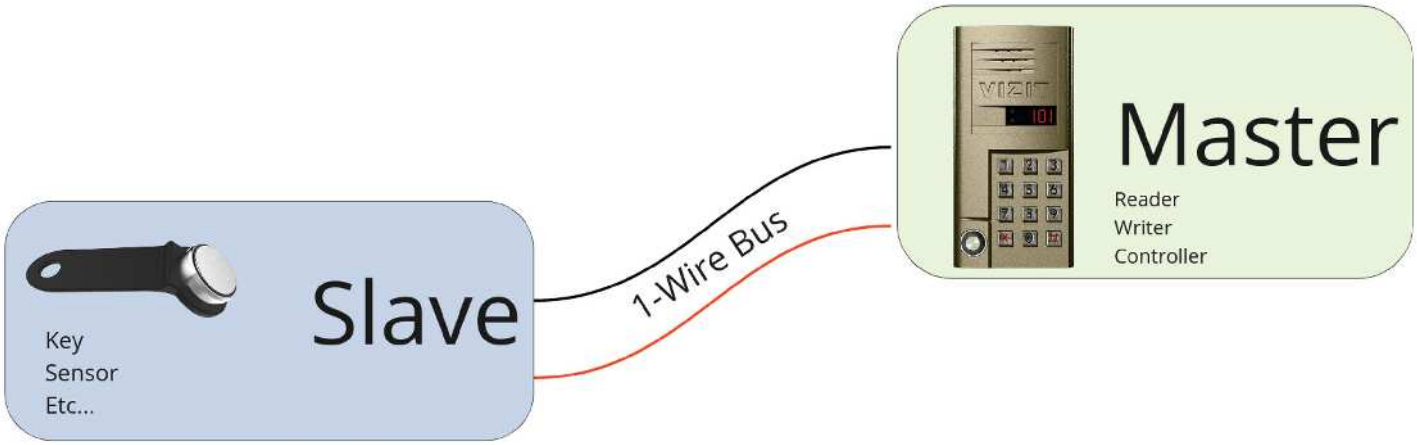
Metakom key data structure

C keys yfral

Cyfral key data structure

16min

Protocol 1-Wire



The 1-wire protocol always has a master and a slave.
slave

Intercom keys use only one contact to receive and transmit data using the 1-Wire protocol in Master-Slave mode. The Master device always initiates communication and the Slave follows its instructions. When the key (Slave) is connected to the intercom (Master), the chip inside the key receives power from the intercom. After that, the key is initialized and the intercom asks for its ID.

Flipper can work as a thief in Master and Slave modes. When reading a key Flipper acts as a reader, that is, it works as a Master.
When emulating a key, Flipper is assumed to be a key, that is, it works in Slave mode.

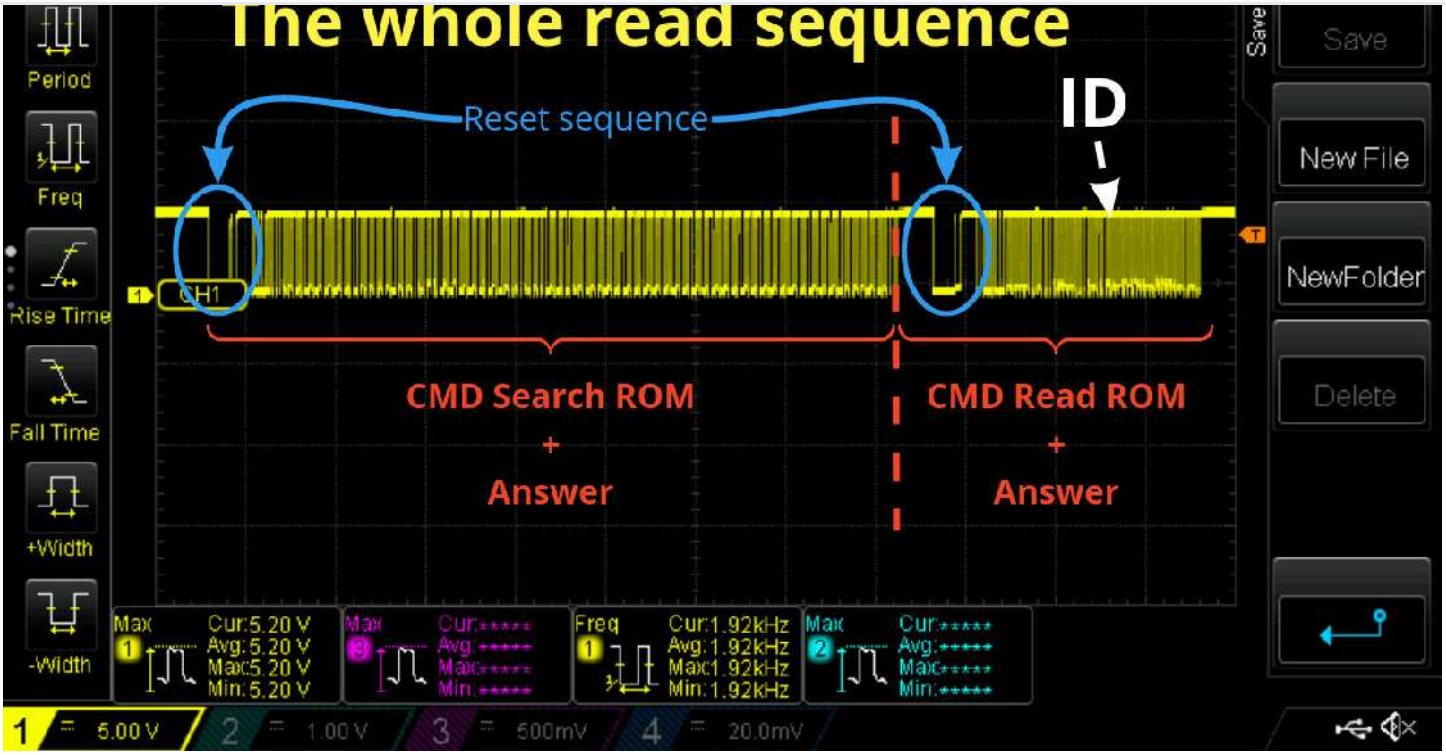
Looking at 1-Wire through an oscilloscope

The transmission line is arranged according to the principle of "mounting AND" and can have one of two states: logical "0" and logical "1". Devices (keys and intercom) have internal transistors, which pull the line to zero at the right time. The entire transmission line goes to the state of logical "0" if any of the devices has transferred it to zero, i.e. if the intercom pulled the line to zero, the key knows about it, and vice versa.

By pulling the voltage and holding the levels, 1-wire has 4 primitives for working on our bus:

1. reset pulse (RESET)
2. Presence impulse (PRESENCE)
3. send bit 0

4. sending bit 1 and also reading bit



Reading the Dallas key on Flipper consists of the Search ROM and Read commands.

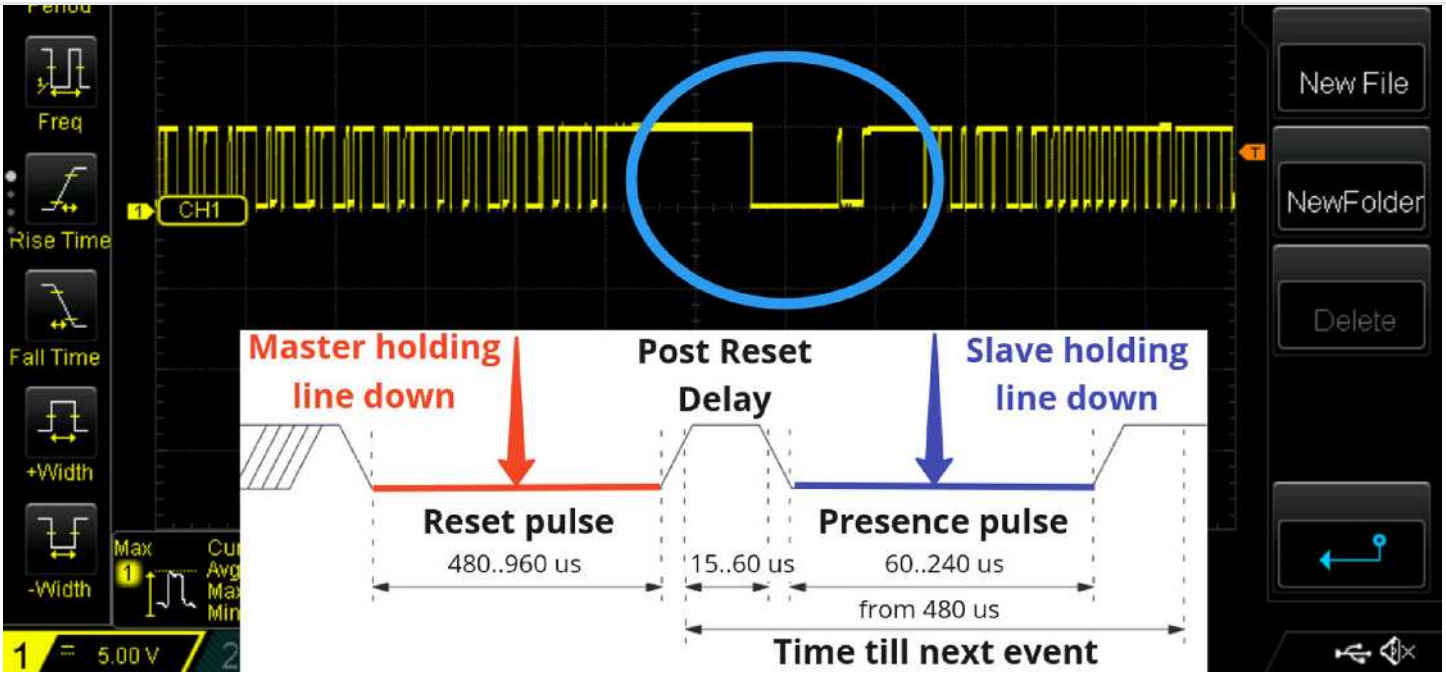
ROM. Each command has a Reset sequence

Reading a key on Flipper Zero works like this: the search command checks for the presence of a key, and then the ID is read. This is done to avoid accidental matches with other Cyfral/Metakom keys that may accidentally match the timings required.

The oscillogram shows a long signal of 2 commands, where each consists of:

- Command initializations:
  - reset impulse
  - Impulse of presence
- Sending a command to a Slave Replying to
- a received command by a Slave

Command initialization



Reset sequence - command initialization. Consists of Reset Pulse and Presence Pulse. Reset Impulse - level lowers
master. Impulse of Presence - the level lowers the Slave

Initialization (reset sequence) consists of two pulses:

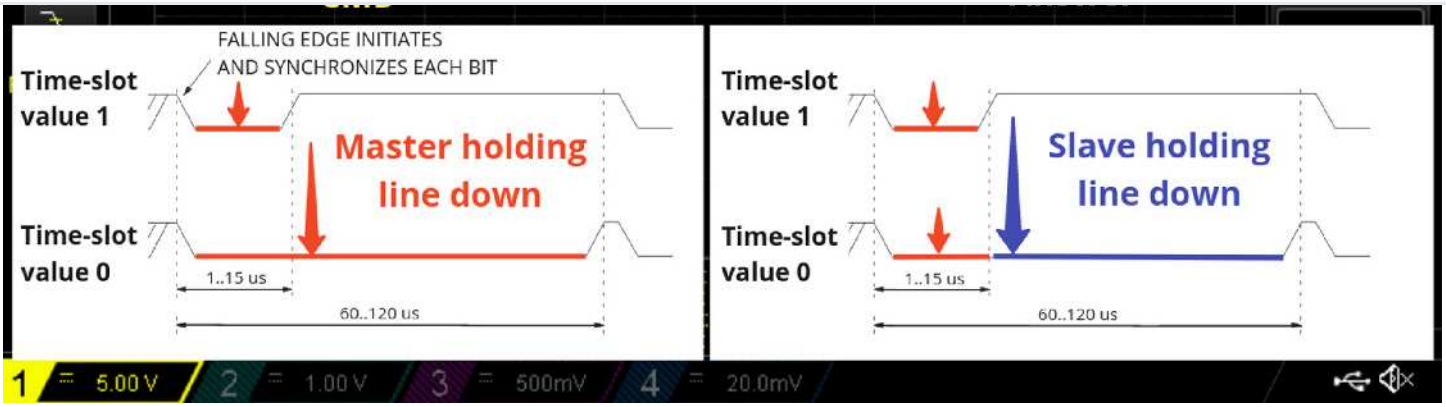
1. Reset pulse
2. Presence pulse

For the Reset Impulse, the Master (intercom) pulls the ground line.

For Impulse of Presence - the line κ to the ground is pulled up by the Slave (κ key).

Some doorphones do not give a reset pulse, so there is a reset signal between the keys and the doorphone.

Sending a command and receiving a response



Read ID command with response. The information bit time slot consists of 2 parts: the timing and the bit value. In COMMANDS, the Master is responsible for the voltage levels in both sections. IN RESPONSE to the read command, the Master is responsible for synchronization, the master is responsible for the bit value slave

After the command is initialized, information is exchanged:

- Sending a command to a Slave
- Slave's response to a command

The exchange of information is carried out by time slots: one time slot for the exchange of one bit of information. Data is transmitted bit by bit, starting with the least significant bit of the least significant byte.
Synchronization of Master and Slave occurs in one step: Master (intercom) pulls the line to a low level. Next, Master or Slave measures the voltage on the line and writes a bit of information (Master - when reading the key, Slave - when writing the key). The time ranges for read and write commands are the same.

Each time slot is synchronized independently, so the transmission of information can be suspended without causing an error. It is important that all signals have specific time ranges that must be respected! It happens that intercom manufacturers run away and use their time delays.

5min

Dallas Keys

The intercom reads 8 bytes (64 bits) of information from the iButton to decide whether to open the door or not.

The data structure of these 8 bytes is as follows:

- 1 byte - Family Code, for iButton it is always 0x01

- 6 bytes - key serial number 1 byte -
- CRC checksum

The family code for Dallas keys is always 0x01. If you have thisIf the code is different, then most likely it is not the key from the intercom.

The serial number is in some cases engraved on the key, but may:

-

value of characters backwards in an incomprehensible place

The original iButton has an ID engraved on it, but its format is slightly different from the one in Flipper: first comes the family code, then the inverted serial number, then

checksum

The picture above shows a non-obvious example of ID engraving on the original iButton. In it, bytes must be read from right to left, the checksum is written on the left, and the family code is on the right.

3min

Metakom keys

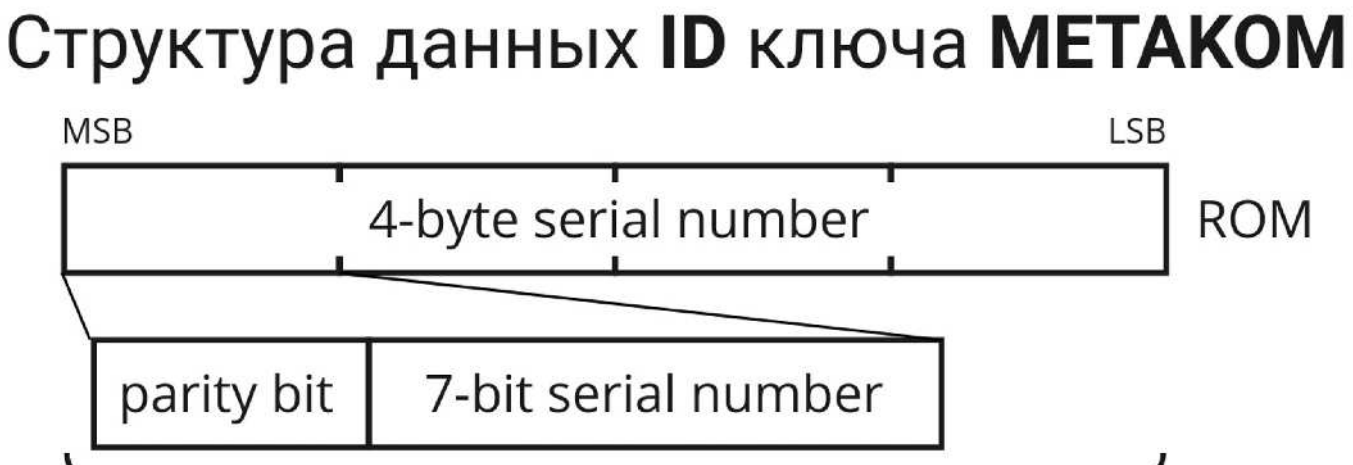
Datasheet — technicalAny information on the operation of the protocol

Unlike Dallas switches, they work not by voltage, but by current. Metakom keys do not accept commands - when power is applied to the key, it immediately starts sending ID endlessly due to a change in resistance. Thus, the logical levels are determined by the resistance of the switch. The Metakom key sends 4 bytes: each byte ends with a parity bit.

Metakom has 3 transmission primitives: Clock Bit, Bit 0, Bit 1.

The structure of the message looks like this:

- Clock Bit
- 4 bytes of information, where each byte contains:
  - 1 parity bit
  - 7 data bits



2min

Cyfral keys

Datasheet — technicalAny information on the operation of the protocol.

Unlike Dallas switches, they work not by voltage, but by current. Cyfral keys do not accept commands - when power is applied to the key, it immediately starts sending ID endlessly due to a change in resistance.

Logic levels in Cyfral, as well as in Dallas, have time limits: if the resistance remains low at about 50 ms, this is a logical “0”, if 100 ms is a logical “1”.

Cyfral cyclically sends 9 nibbles (1 nibble = 4 bits): 1 start and 8 IDs. Nibble can have a total of 4 values for the ID and one value for the start word. All other entries are incorrect.

Ниббл	Значение
1110	0
1101	1
1011	2
0111	3
0001	Стартовое слово

in 2 bytes (Total 8 nibbles ID. 4 nibbles = 16 states = 1 byte of information). Cyfral does not have any control amounts. If you want to make sure the integrity of the ID - read the key again, but you want five.

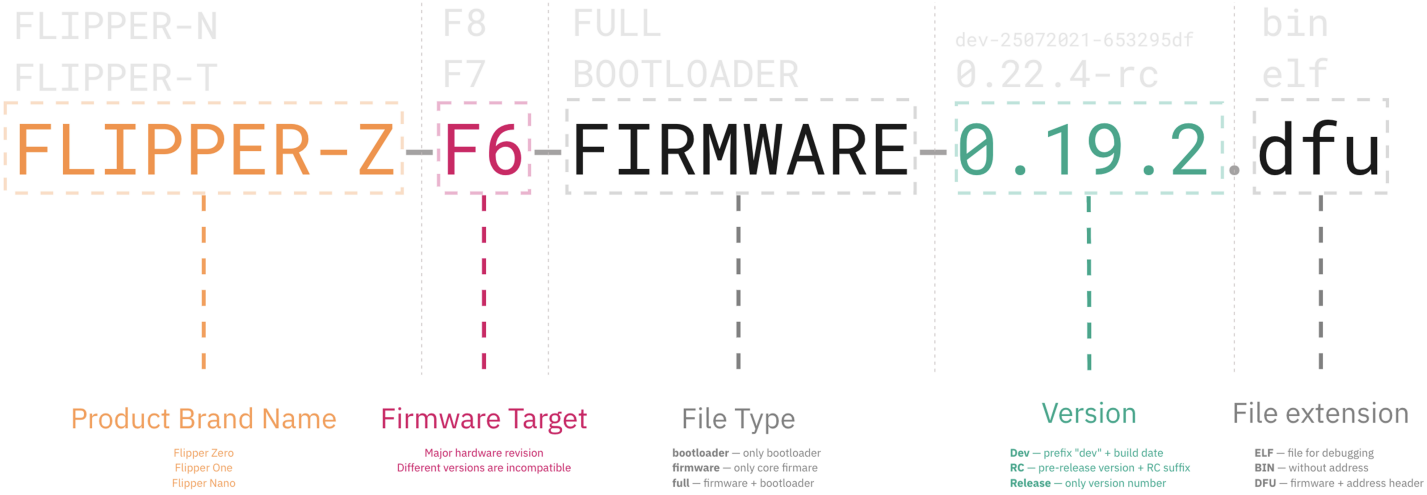
Cyfral data structure

Since the signal is analog, reading the signal requires an ADC or comparator. It is easiest to use a comparator whose output is low or high voltage.

1min

Firmware

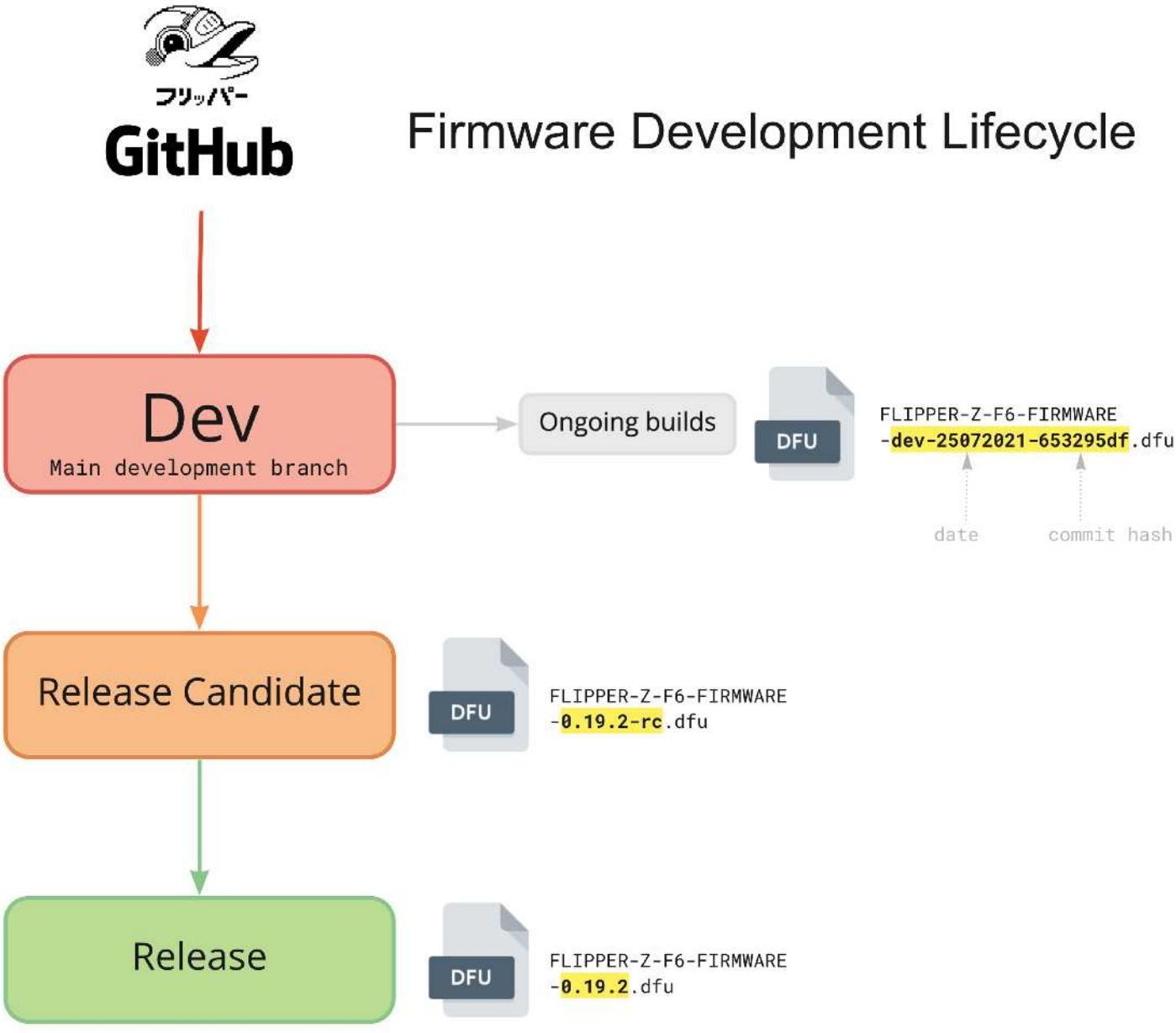
Firmware files



Firmware filename structure

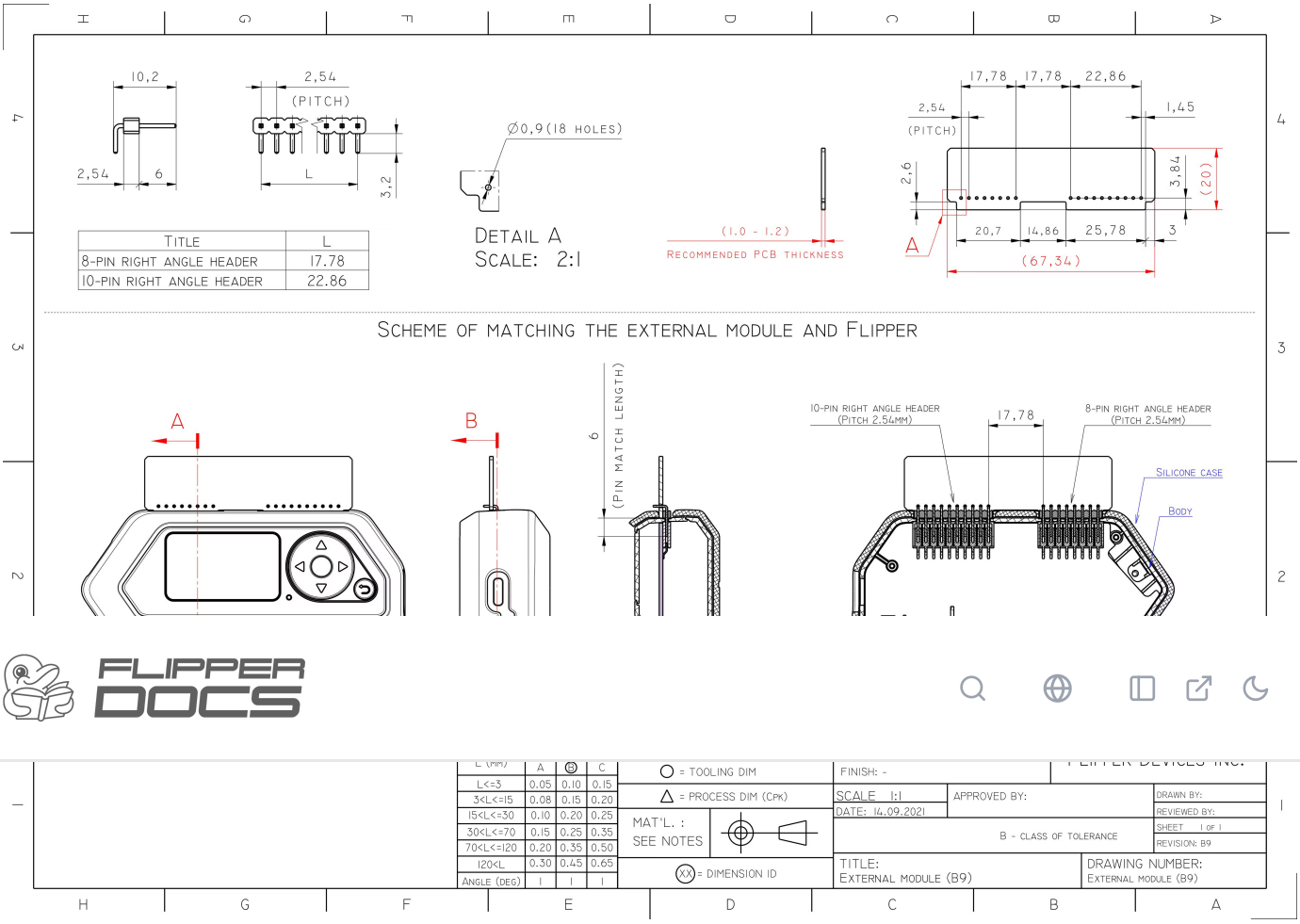
0min

Жизненный цикл релизов прошивки

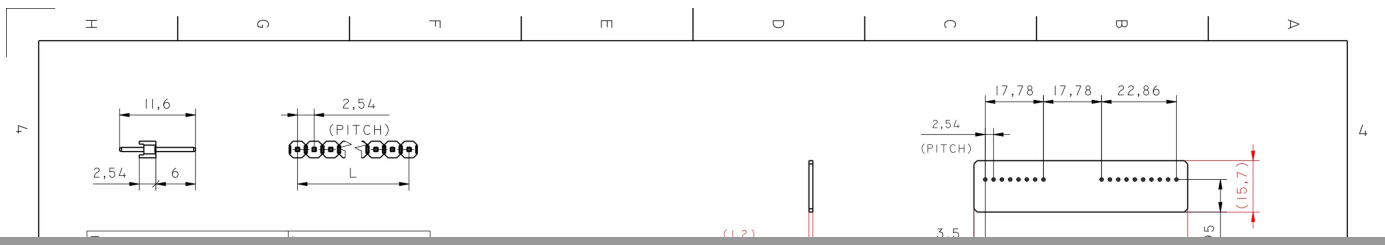


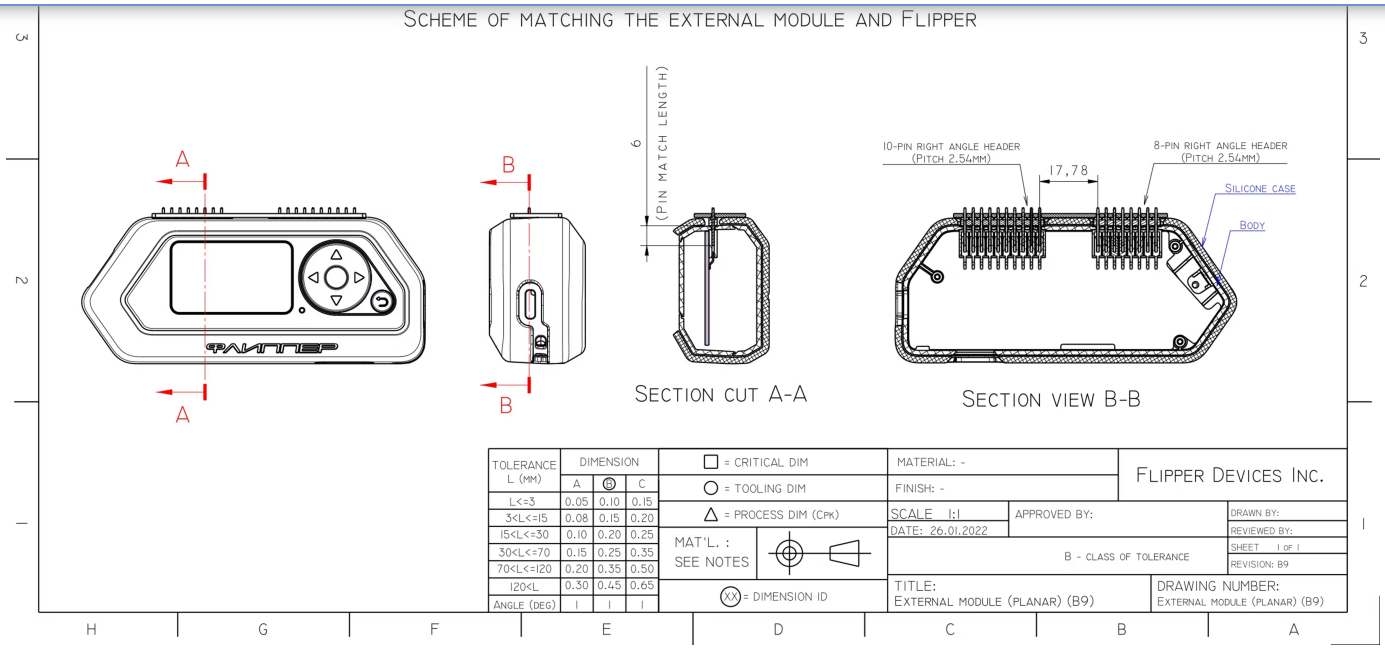
7 minutes

externalmodules



Plugin drawing for Flipper Zero





Drawing of aminiature plug-in for Flipper Zero

Drawing

xternalmodule.pdf

xternalmodulesmall.pdf

circuit board outline

xternalmodulePCB outl...

xternalmodulePCB outl...

xternalmodulesmall PC...

3D models

xternalmodulePCB.stp

xternalmodulePCB (as...

xternalmodulesmall PC...

xternalmodulesmall PC...

5min

How to make a plug-in for Flipper Zero

Flipper Zero on the side of the case has a GPIO comb, which provides for the connection of expansion modules. This makes it possible to design external boards that can be connected directly to the Flipper and access various IO pins of the **STM32WB55RGV6TR** microcontroller .

GPIO comb

The power pins (+3.3 V and +5 V) are input in relation to external connected devices. Thus, modules can be powered directly from Flipper.

The signal pins are connected to the IO pins of the microcontroller through 51 Ohm resistors and allow operation with voltage levels up to +5 V. Information on the pinout and functionality of the pins is located in the **GPIO and Modules** section . By default, the signal pins, except for pin 17, are floated * . Pin 17 is the iButton pin, and is pulled up to +5V.

```
* Note
When Flipper is in DFU mode, the microcontroller pins change their
state according to table 184 section 68 of AN2606 Application note
STM32 microcontroller system memory boot mode .
```

All GPIO pins of the comb are ESD protected. You can see the complete diagram of connecting the GPIO pins of the comb (connectors X8, X9 of the MAIN_PCB board) to the micontroller in the **Hardware** section .

Electrical Requirements

Powered by +3.3V (pin 9) and GND (pins 8, 11, 18)

- Maximum load 1200 mA. The output is enabled by default (Flipper's MicroSD card is also powered from it).
- When the firmware is rebooted, as well as when mounting the MicroSD card, the voltage on this pin is temporarily turned off.

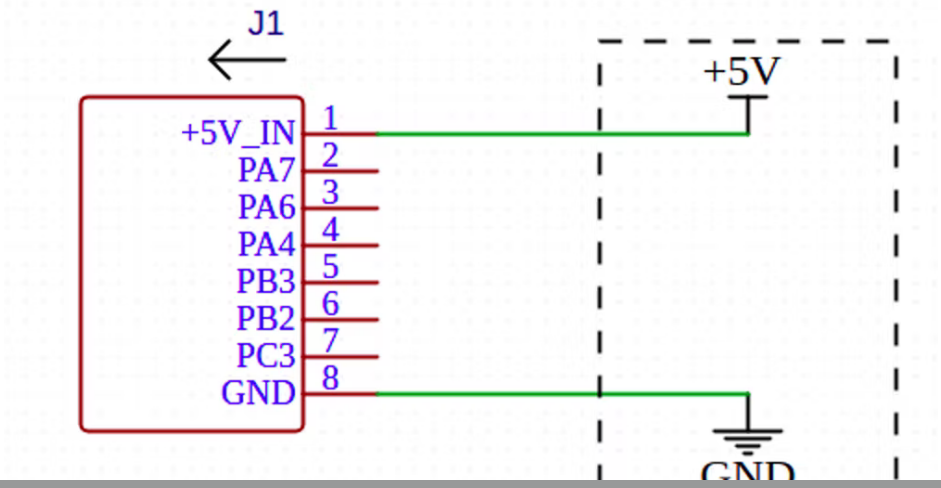
Powered by +5 V (pin 1) and GND (pins 8, 11, 18)

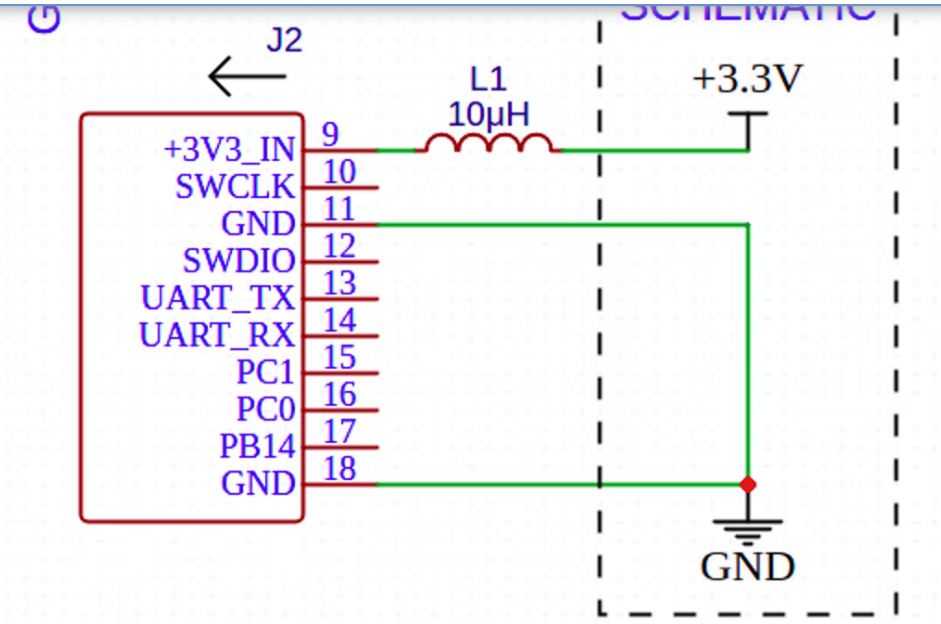
- The maximum load is 1000 mA when the Flipper is powered from the built-in battery. The output is not active by default, you need to enable it through the "Power" menu
- If Flipper is connected to USB, then the power on this output is taken directly from USB. Thus, there is no current limit on the Flipper side, be careful!

Connecting modules with a large capacitive load.

When connecting modules with a capacitive load of more than 330 uF, the sd card may not work correctly, namely, the inability to use files from the media.

To avoid this problem, it is necessary to add a 10uH inductor to the +3.3V output.





Mechanical requirements

For correct docking of an external module with a GPIO comb, you must adhere to **the specification**

We recommend using **EasyEDA plug-in project** as a source and modifying the board according to your wishes. This will save some time for drawing the outline and arranging the connectors in accordance with the specification. For compact modules, you can use the **miniature plug** -in project .

Examples of ready-made plug-ins

1. once
2. two
3. three

0min

Mechanics

0min

CLI Console