

**3257.82**

Рейтинг

RUVDS.comVDS/VPS-хостинг. Скидка 15% по коду **HABR15**[Подписаться](#)**CodeName33**

1 ноя 2024 в 12:01

«Удалённо» управляем компьютером с доступом в BIOS



Средний



9 мин



49K

Блог компании RUVDS.com, C#*, DIY или Сделай сам, GitHub*, Open source*

[Тutorial](#)

Итак, о чём это? Сейчас для удалённого управления компьютером есть великое множество программ на любой цвет, вкус и запах. Но что, если мы хотим пойти немного дальше, и наши требования к удалённому управлению становятся немного жёстче:

- Мы хотим иметь возможность не только работать в операционной системе, но и заходить в биос, или вообще эту систему переустановить.
- По тем или иным причинам, компьютер не может быть подключён к сети, но управлять им мы от этого меньше не хотим, а рядом у нас есть компьютер, который в сети находиться может.

Но как это сделать? Вот этим мы тут и будем заниматься...

■ Как я к этому пришёл

Иногда мне приносят разное железо с просьбами переустановить винду/почистить вирусы и т. д. А я что? Я ж программист простой: мне приносят и просят сделать — я делаю. Но порой не очень удобно подключать к этому всему отдельную мышь/клавиатуру и монитор, а бывает, что там идёт долгий процесс, не требующий особого вмешательства, но периодически надо сделать пару кликов, что я мог бы сделать удалённо с работы, будь у меня такая возможность, и сэкономить кучу времени вечером. Я знаю, что такое удалённое управление бывает на серверных материнских платах, но в последний раз мне приносили сервер, чтобы я переустановил на нём Windows никогда, или даже ещё раньше.

В какой-то момент у меня возникла в голове идея: есть же недорогие устройства видеозахвата USB-HDMI, а ещё есть ESP32 S2/S3, которые умеют эмулировать USB. А что нам ещё надо? Изображение с компьютера мы можем получить, клавиатуру/мышь можем проэмулировать. Может быть, такие проекты даже уже есть, но когда мне в голову приходит идея, которая кажется мне интересной, я:



Поискать, вдруг кто-то
уже сделал что-то
подобное



Изобрести свой
велосипед

Ну что ж, проекту быть, и для него нам потребуется следующий минимум:



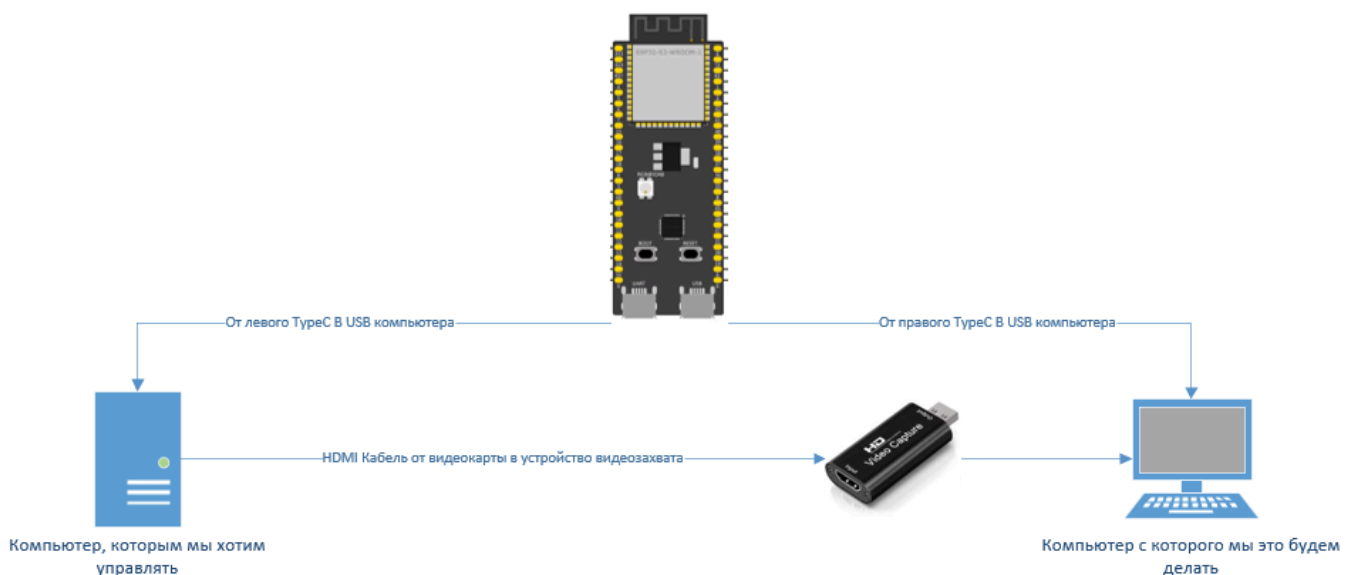
- HDMI-плата захвата видео
- Плата ESP32 S3 — у неё сразу есть 2 TypeC разъёма, что упростит нам жизнь
- HDMI-кабель
- 2 кабеля USB TypeA — TypeC
- Компьютер с Windows, стоящий рядом

С железом условно всё, и если нам не нужно управлять аппаратной перезагрузкой/включением компьютера, то нам даже не придётся ничего паять. А если нужно, то всему своё время...


Итак, возможно, когда я написал пункт «Компьютер с Windows, стоящий рядом», я кого-то очень сильно огорчил. И я согласен, что решение не самое оптимальное, если бы всё работало под Linux, да ещё выводить всё в Web, то можно было бы взять Малинку/Апельсинку и... Но нет. Хотя, может и да, ведь проект открытый, и если у кого-нибудь будет время, желание и умение, то он может переделать под Линукс мой проект, сделанный на .Net, но пока я всё основное время работаю под осью одной из корпораций зла, проект только под Винду.

■ Приступаем к работе

Ладно, начинаем. Схема подключения простая:



Теперь, когда у нас всё подключено, что дальше? Дальше пишем ПО.

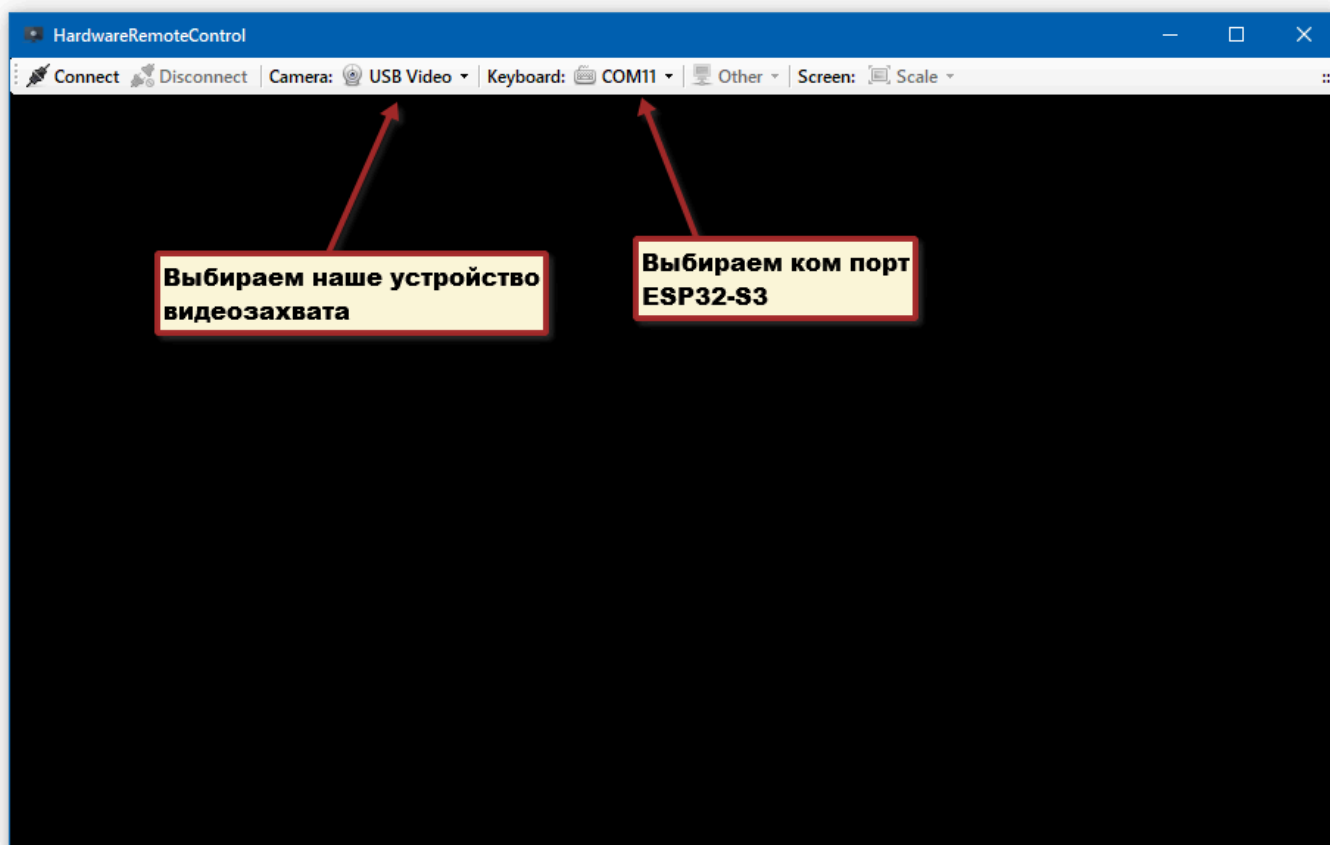


Какнибудь
напишу



Все, написал

ПО написано, что делаем дальше? Для начала надо залить прошивку на нашу ESP32 S3. Убеждаемся, что драйверы ком-порта у нас установлены (откуда их брать, обычно указывает продавец этой самой платы). Для заливки прошивки я использовал Arduino IDE, тем более что скетч написан именно в нём. В этой статье я не буду подробно описывать процесс настройки Arduino IDE на работу с платами ESP32 и прошивки — его можно найти, например, везде, и он довольно прост. Дальше перейдём непосредственно к ПО для удалённого управления, а тут всё ещё проще:



Подключаем всё, включаем компьютер, которым мы хотим управлять, и в программе нажимаем кнопку подключения. И если все звёзды на небе сошлись, то мы видим изображение с монитора компьютера, у нас есть мышь и клавиатура и они работают.

■ Немного о коде

Прежде чем продолжить проект, добавив к нему ещё пару крутых фиш, я предпочту немного замедлиться и погрузиться в код, который был написан. И если со стороны всё выглядит как вжух и готово, то на деле было не совсем так. Начиная этот проект, я стал искать библиотеки для .Net для работы с устройствами видеозахвата. И первое, что мне попалось, было **OpenCvSharp**. Я проверил, что эта библиотека работает с веб-камерами и другими подобными устройствами и выдаёт изображение на WinForms. Отказался я от неё потому, что не нашёл у неё возможности нормально перечислить список всех камер с их именами для выбора в меню, а городить это отдельно не очень хотелось, и пока я ещё не сильно привык к этой библиотеке, я стал смотреть другие.

Следующей была **AForge**. В ней список камер перечислялся нормально, с именами и фамилиями, и, казалось бы, всё было хорошо, но нет. Когда я тестировал программу на реальном устройстве видеозахвата, всё было хорошо, но когда отлаживался на виртуальной веб-камере от OBS Studio, картинку я не получал. При этом ошибку тоже. Я не получал ничего, я был просто проигнорирован и видел, как происходит ничего. Мне не понравилось. Я бы подумал, что что-то не так с самой камерой у OBS, но в OpenCvSharp всё прекрасно работало. И я стал копать... И вот щетина на моём лице уже выросла, вся

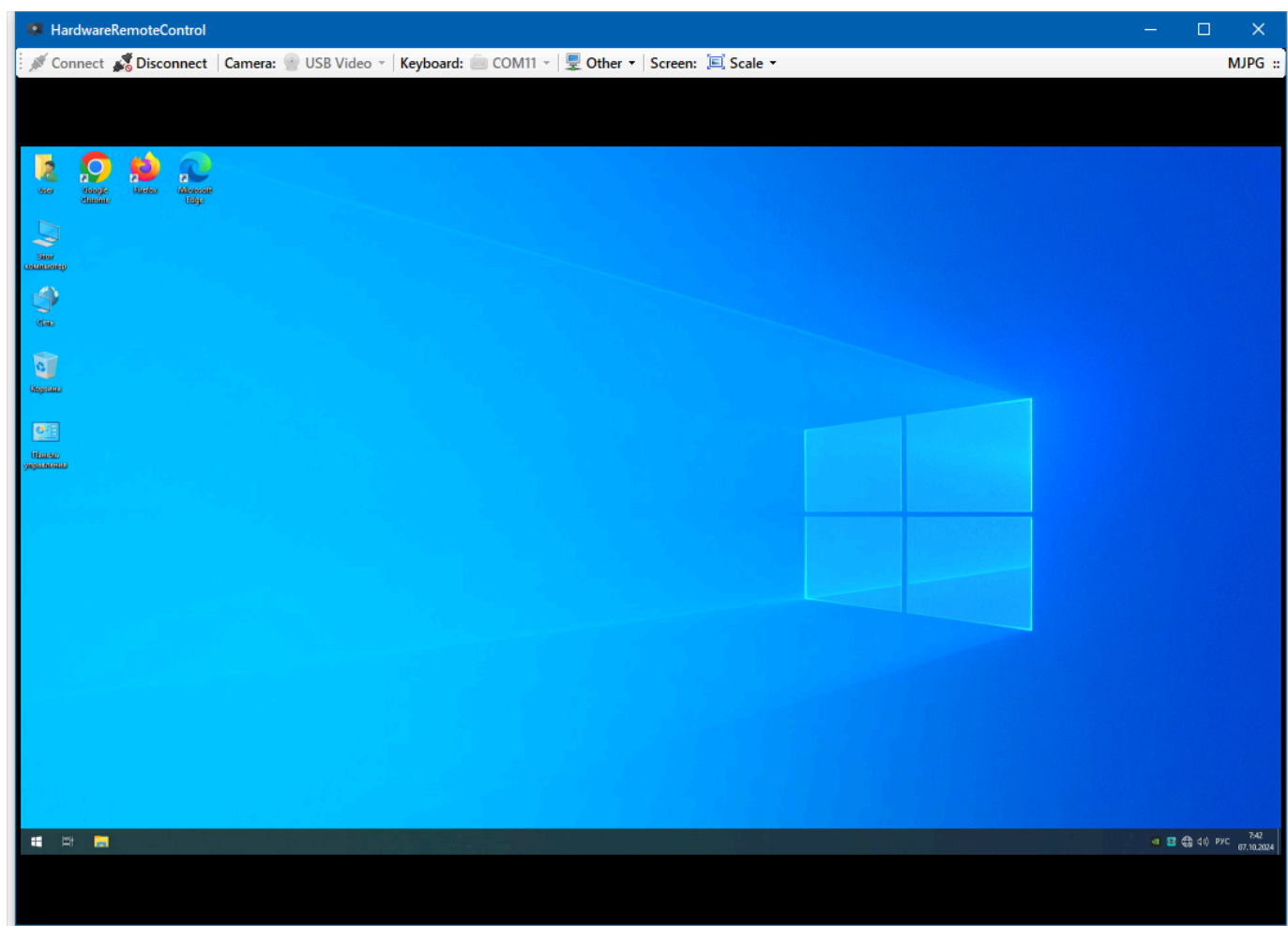
водка была выпита, медведь доиграл свою арию на балалайке, но ничего по-прежнему не работало. Но вскоре проблема вскрылась. AForge принудительно запрашивает у камеры формат RGB24, и если устройство его поддерживает, то всё ОК. А если нет, тады ой. Камера от OBS поддерживала формат NV12, но AForge не умела его декодировать, как и многие другие форматы. Можно было оставить эту библиотеку и пойти к следующей, но это уже было делом принципа, я потратил на поиск этой особенности слишком много времени, и раз уж исходный код AForge открыт, я решил его поправить. В очередной раз я убедился, насколько ChatGPT (и ему подобные) делают нашу программистскую жизнь легче.

▸ Магия XXI века

Вот так легко и просто ChatGPT сгенерировал мне функцию перекодировки YUY2 в RGB, также я сделал для NV12, YUYV и I420. Если бы я писал это сам, я бы потратил на это, может быть, весь день, пока разбирался в этих форматах пикселей и отлаживал баги. А тут код написан за минуту и после визуальной проверки на отсутствие явной лажи уже работал в проекте, сразу же после компиляции. Ладно, если нужна бочка дёгтя на вашу ложку мёда, я вам её таки дам: так хорошо бывает не всегда, иногда он генерит нерабочий код, иногда рабочий, но неэффективный. Но можно попросить его ещё раз? и часто у него получается лучше.

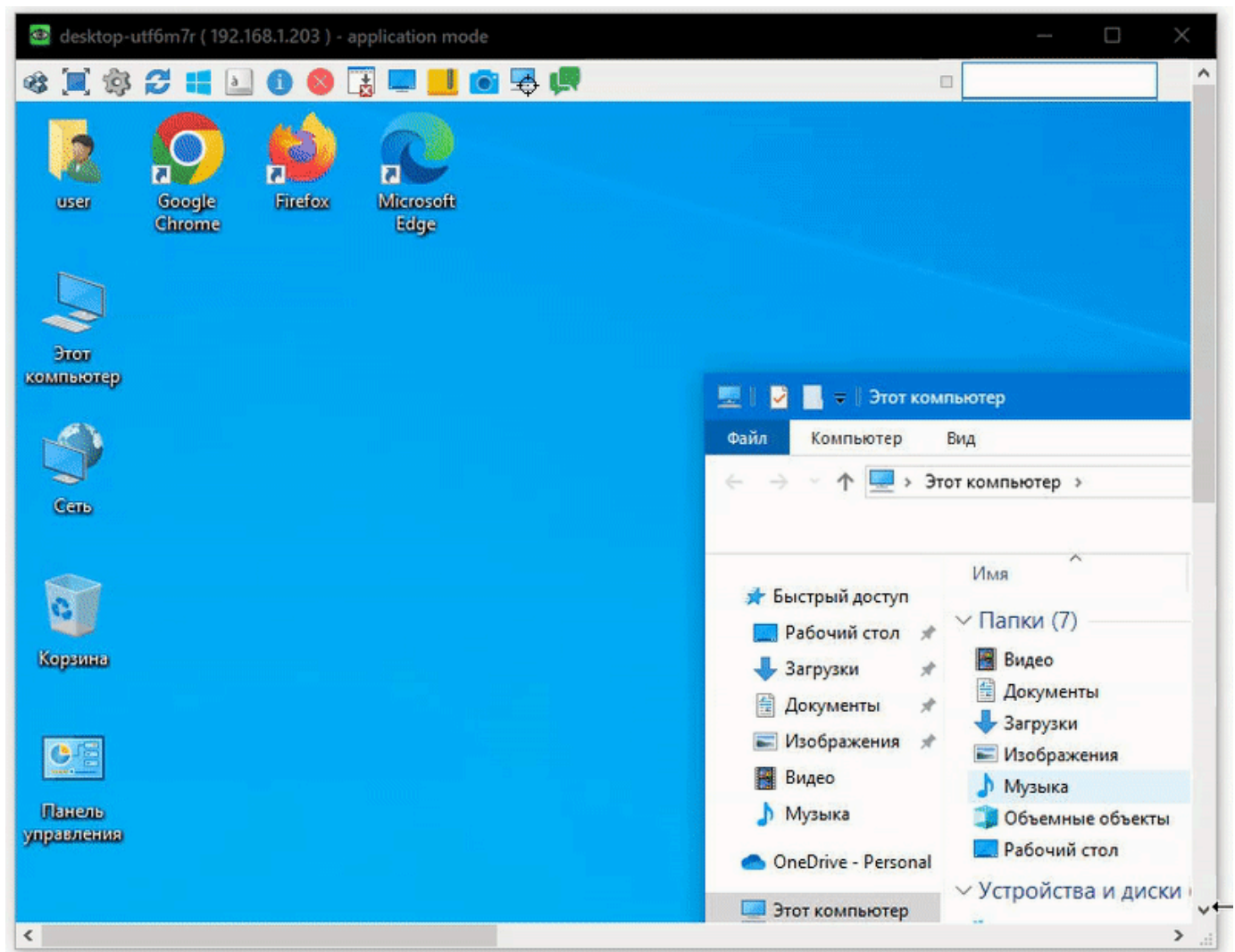
■ Немного про экран

При удалённом управлении часто бывает, что размеры удалённого экрана превышают размеры окна, в котором мы работаем. И самый простой вариант — это обычное пропорциональное растягивание/сжатие картинки под размер рабочей области окна.



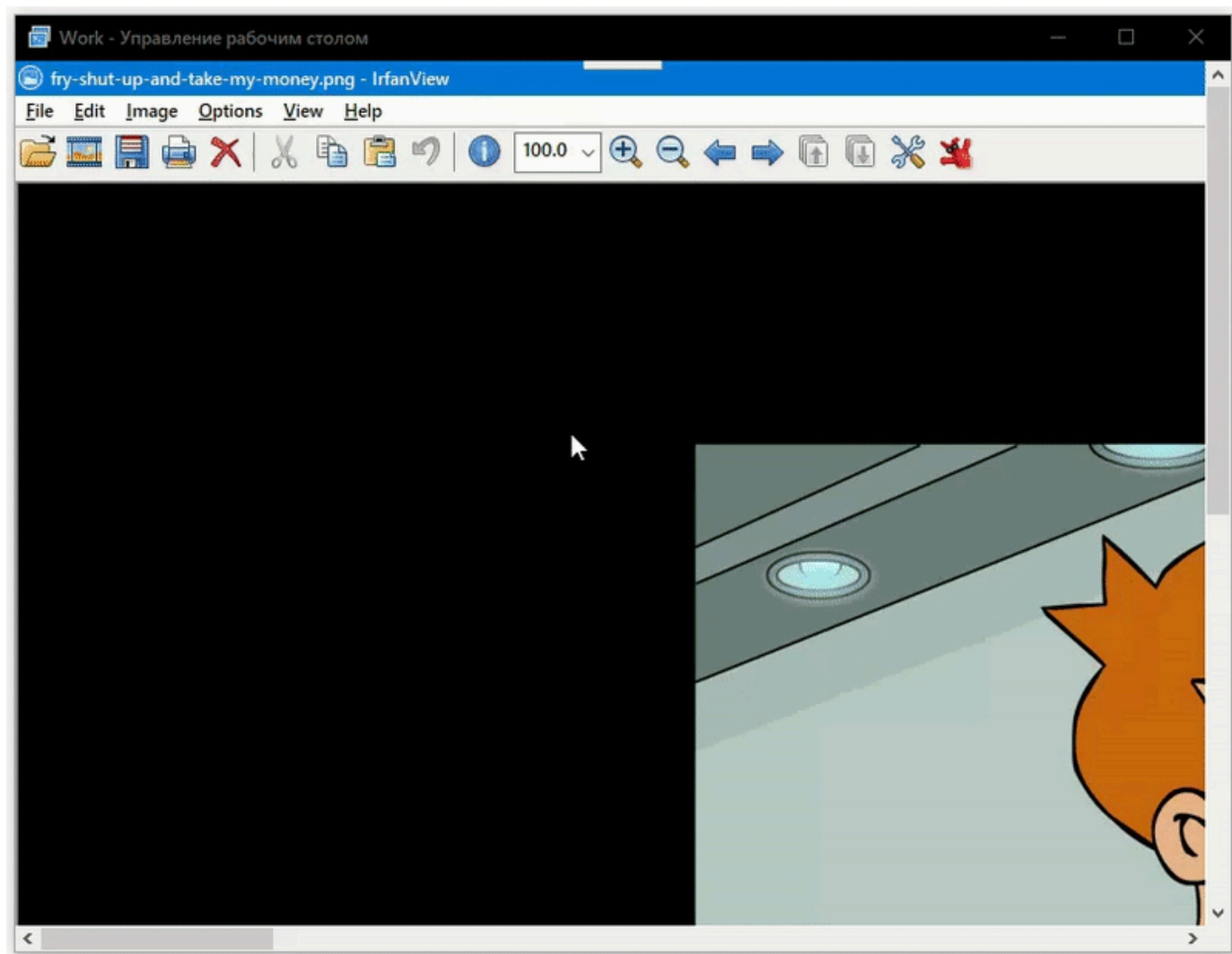
Но, бывают ситуации, когда хочется видеть всё в масштабе 1 к 1, а удалённый экран больше нашего. И я отметил для себя 4 разных варианта, один из которых и реализовал:

1. Просто скроллбары по краям, которые нужно скроллить мышкой вручную. Это не очень удобно.

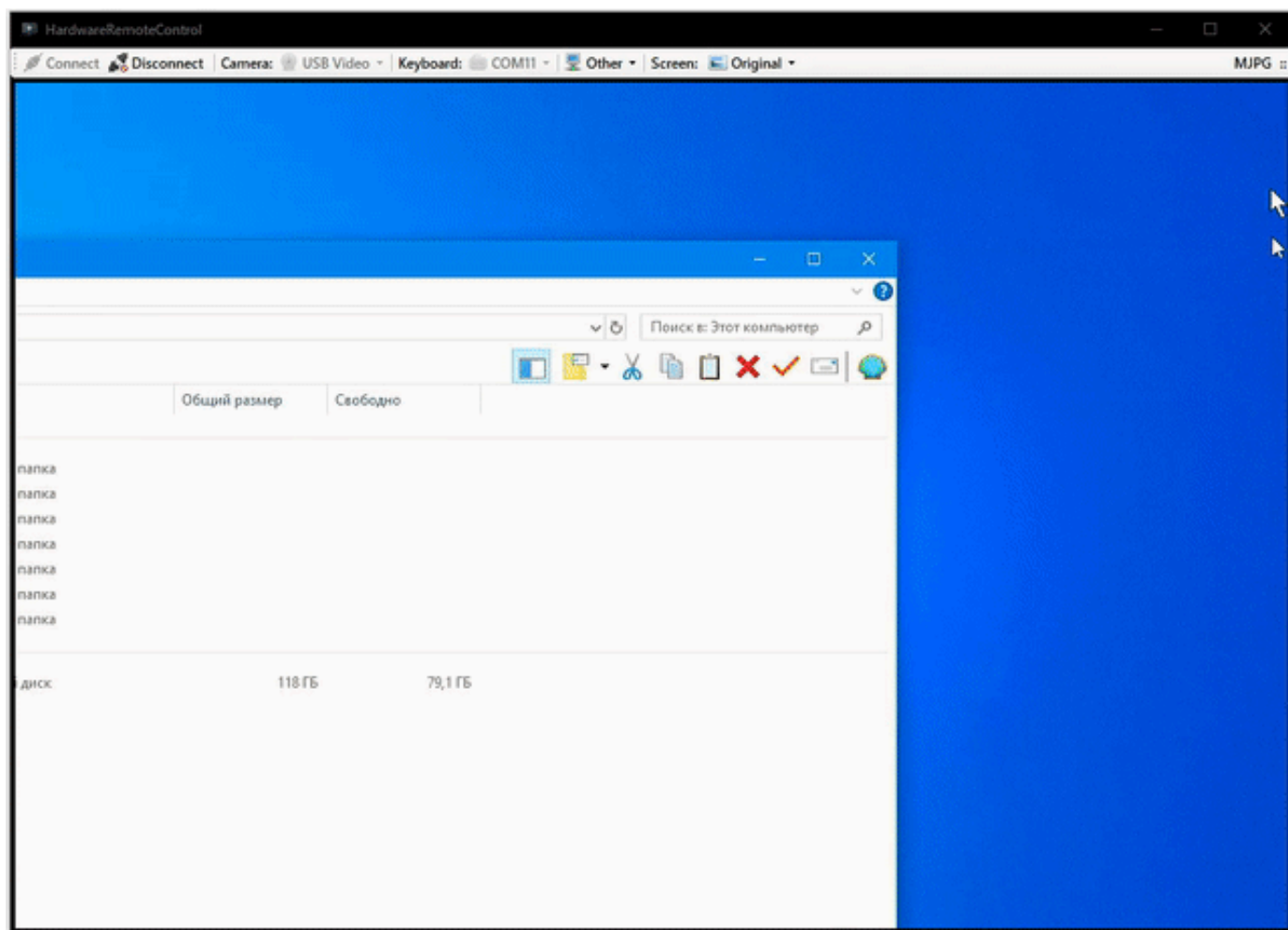


2. Стилль RAdmin — когда мы подводим мышку к краю окна и ведём её дальше, экран начинает скроллиться, а движение мышки блокируется. Уже лучше, но мне не нравится, что в этом случае при проскролливании нужно блокировать движении мыши.

3. Стилль Aspia — когда мы подводим мышку к краю окна, окно начинает само скроллиться, перемещение мышки при этом не блокируется.



4. По мере того, как мы ведём мышью от одного края нашего окна к другому, экран сам проскроллится к этому краю. Поначалу не привычно, но потом вполне удобно. Этот вариант мне понравился больше, и именно его я и реализовал.



Эмуляция устройства ввода

С ESP32 S3 программа взаимодействует через ком порт. Она просто отправляет ей команды (KeyDown, KeyUp, MouseDown, MouseUp, MouseMove). Для мышки была выбрана эмуляция устройства с абсолютным позиционированием курсора, там передаются координаты x и y в пределах от 0 до 32768. Таким образом мне не нужно думать, какое разрешение на удалённом компьютере, всё будет работать само. С клавиатурой оказалось немного сложнее — получаемые коды клавиш нельзя было просто передать один в один в класс USBHIDKeyboard, точнее можно, но со своими приколами, которые местами все портили. Но можно было передавать сырые USBHID-коды, в которые нажатые клавиши надо было сконвертировать. Этот путь я и выбрал. Дальше эти нажатия/отжатия уже отправляются на устройство и эмулируются на конечной системе. Я не стал заморачиваться с перехватом особых спец-клавиш типа CapsLock, но на сегодняшний день у меня нет сценариев, где это могло бы потребоваться.

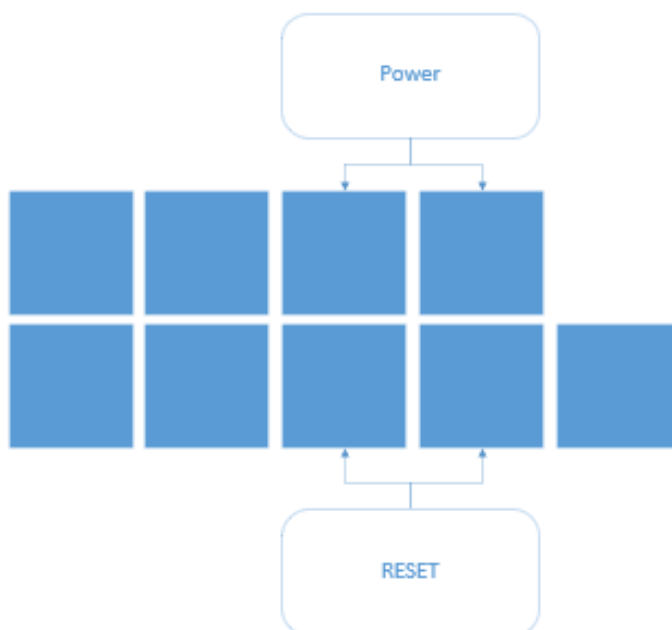
Переходим к проверке:



Заходим в биос, загружаемся в ОС, проверяем, как работает мышь и клавиатура

■ А теперь сделаем это ещё лучше!

В процессе создания всего этого безобразия я решил, что его можно сделать ещё безобразней! А именно: мне внезапно может потребоваться перезагрузить компьютер, если он завис наглухо. Или выключить, а то чего он тут работает? А потом включить потому, а то чего это он не работает? Для этого надо замкнуть соответствующие пины на материнской плате. Это разъём Fpanel и нам нужны вот эти ребята:



У меня дома валялось пара реле с управлением от 3 вольт, и я подключил его управление к пинам ESP32 S3, а замыкание к пинам материнской платы и... естественно, ничего не заработало, потому что нельзя подключать реле к пинам ESP32 напрямую, они не дают такой ток, чтобы сработала катушка реле, но:

Дяденька, я, на самом деле, не настоящий электронщик, я этот паяльник нашёл!

Ладно, у меня валялись ещё IFR3205. Не надо на меня так смотреть, я понимаю, что использовать их для включения реле — это дикая дичь, они были рождены летать, а не ползать. Но ничего не выйдет, потому что, я сказал «ползать» и они поползли! Я не буду выкладывать схему подключения этого безобразия, потому что мне стыдно. Проще взять готовые к подключению напрямую реле типа этих и не париться:



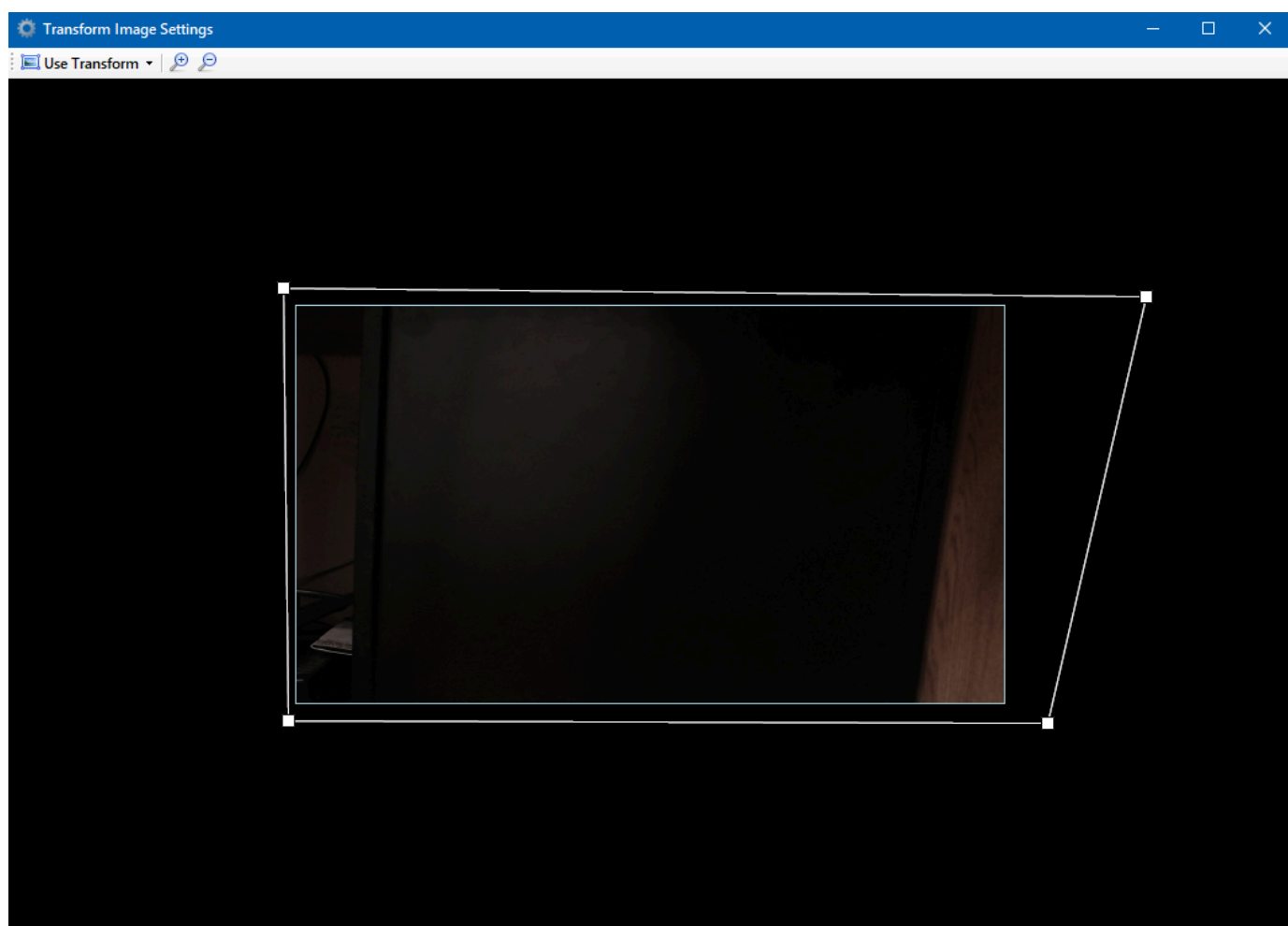
Я использовал пины 17 и 18, но если нужны другие, можно поменять это в скетче.

Итак, теперь у меня заработали кнопки перезагрузки и включения удалённого компьютера. Также я добавил возможность ввода текста из буфера обмена эмуляцией нажатия этих клавиш (как в HyperV). Программа даже умеет переключать раскладку, если видит русские буквы в тексте. Главное, чтобы изначально раскладка на удалённой системе всегда была выбрана английской, а то будет всё наоборот. И вроде бы всё было закончено. Но тут мне пришла в голову ещё одна безумная идея...

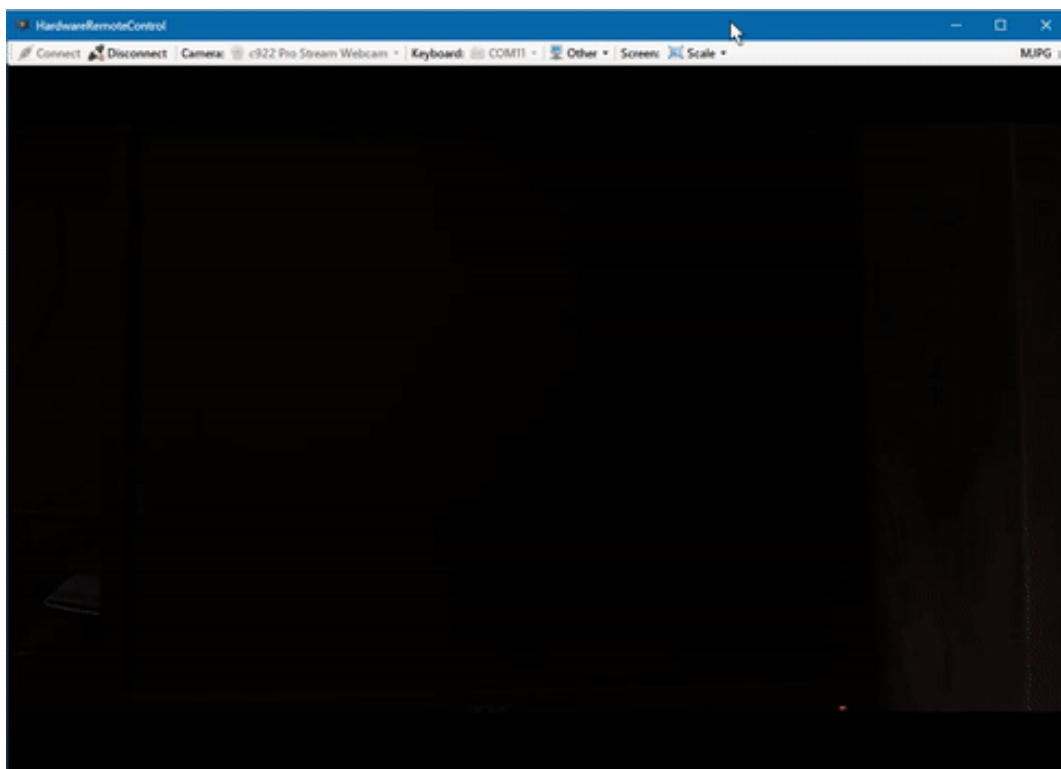
А как насчёт реальной камеры?

Да, ведь мне могут принести ноутбук или моноблок, у которого может и не быть второго видеовыхода, а если и быть, то не факт, что он будет выводить туда стартовую загрузку.

Поэтому я могу просто поставить камеру перед экраном и брать изображение с него. Но! Я же никогда не смогу поставить камеру настолько идеально ровно, чтобы экран был чётко в кадре, не вылезая из него и не оставляя лишнего по краям. А если экран не будет точно заполнять кадр, то сойдутся и координаты мыши. Поэтому мы будем натягивать ~~свою~~ ~~на~~ ~~на~~ изображение с камеры на наш виртуальный экран. Здесь мне было откровенно лень. Я попросил ChatGPT сгенерить мне функции для Perspective Image Distortion. После N попыток он привёл меня к библиотеке [Emgu.CV](https://emgu.com/Emgu.CV/), которая делала это достаточно быстро, и я накидал редактор для этого растягивания.



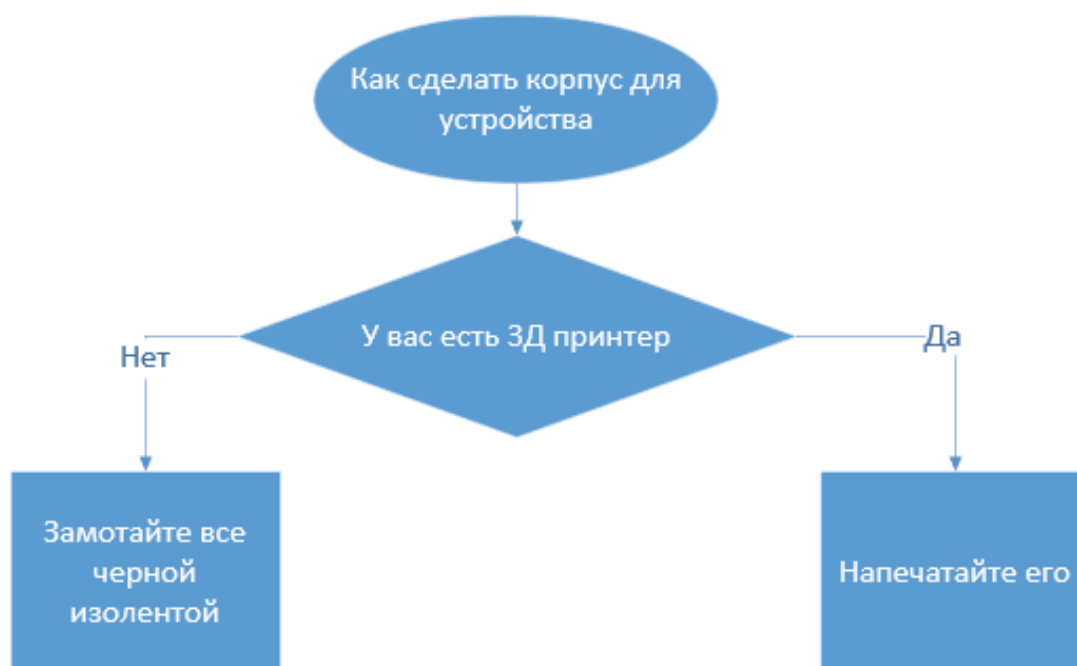
Теперь мы можем натянуть картинку с камеры на наш экран, и мышка ездит от угла к углу достаточно точно. Качество изображения, конечно, прекрасно настолько, что от работы с такой картинкой возникает неременное желание куда-нибудь выйти, например, в какое-нибудь ближайшее окно. Но тут уже всё зависит от качества камеры. Главное, возможность есть.



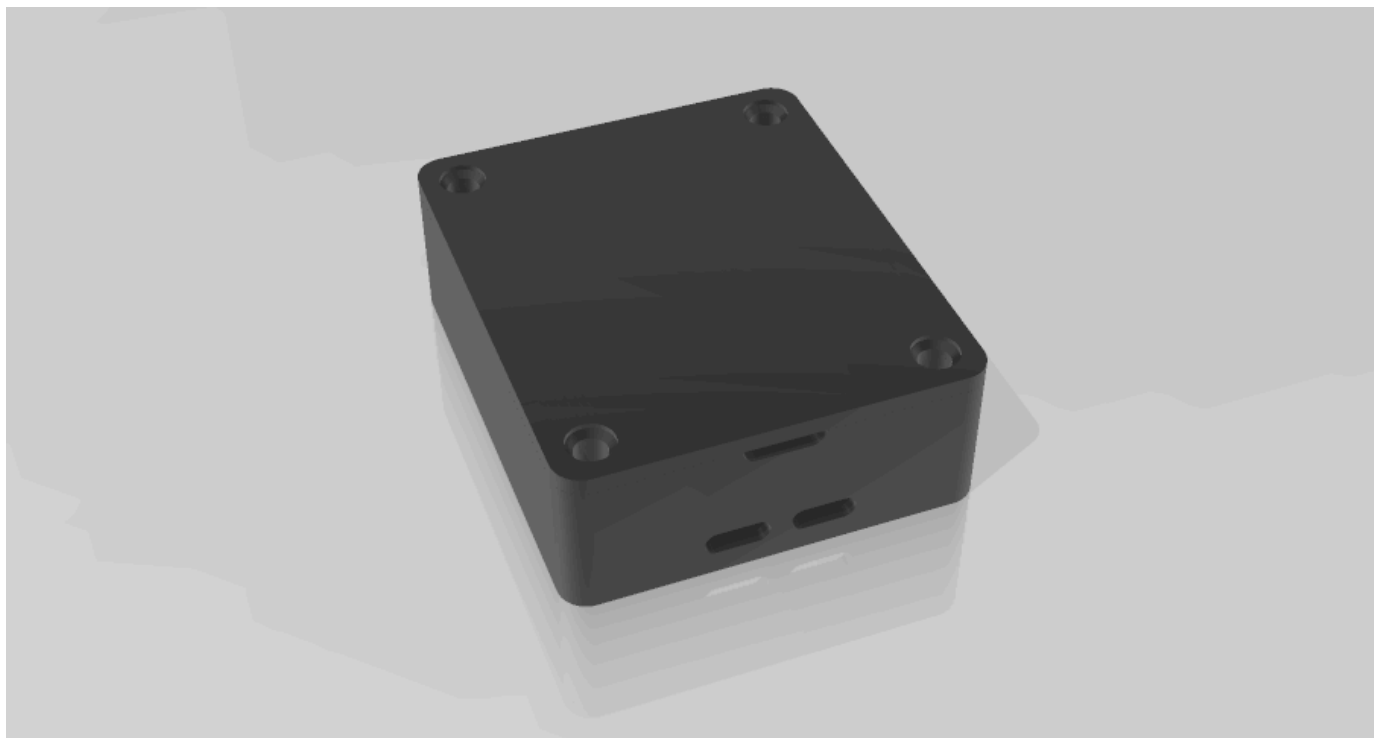
Готовы поработать так на удалёнке весь день?)

■ Бонус

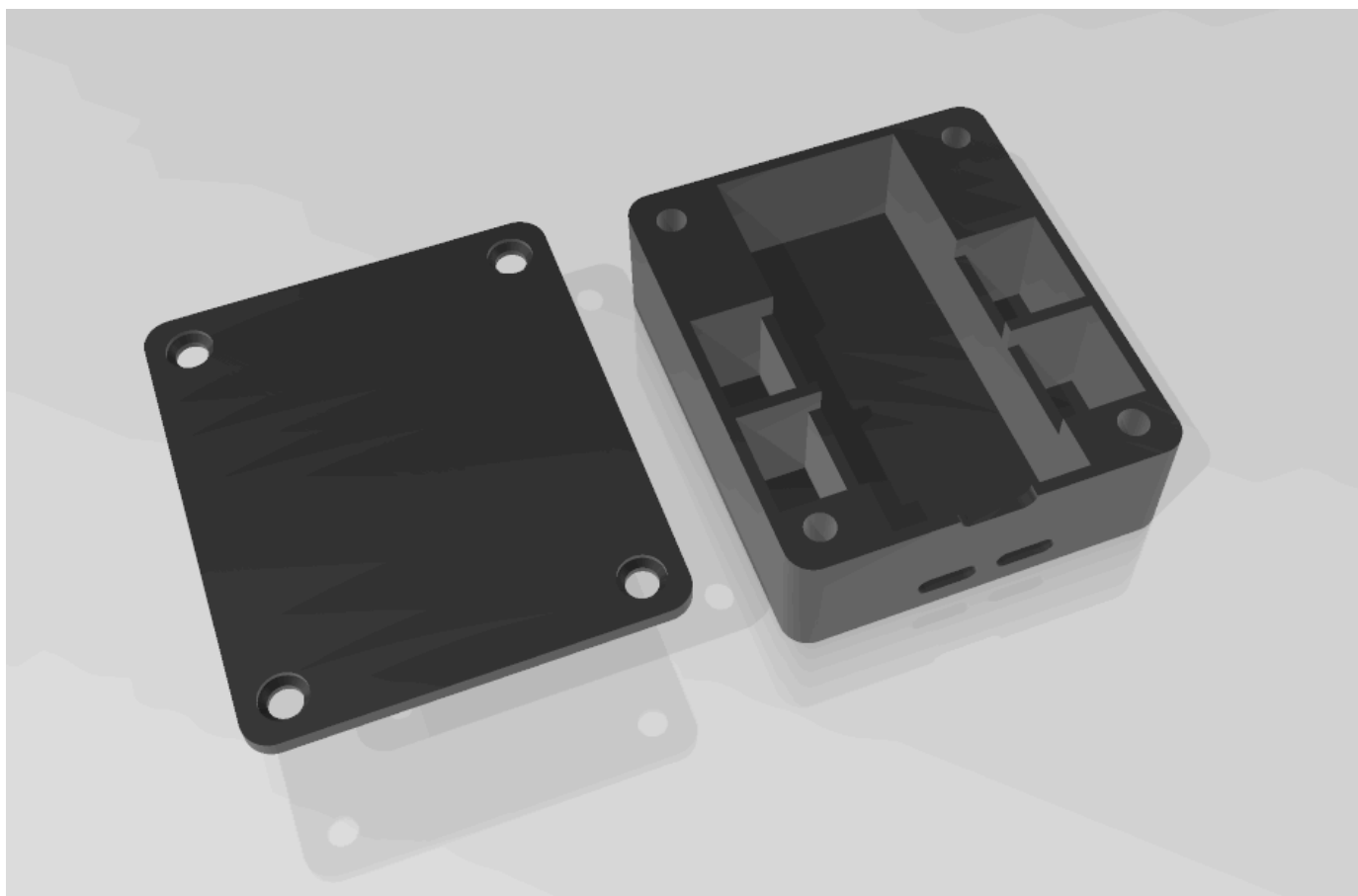
Не люблю я, когда устройство не похоже на устройство. У меня давно сложилась простая схема для создания корпусов:



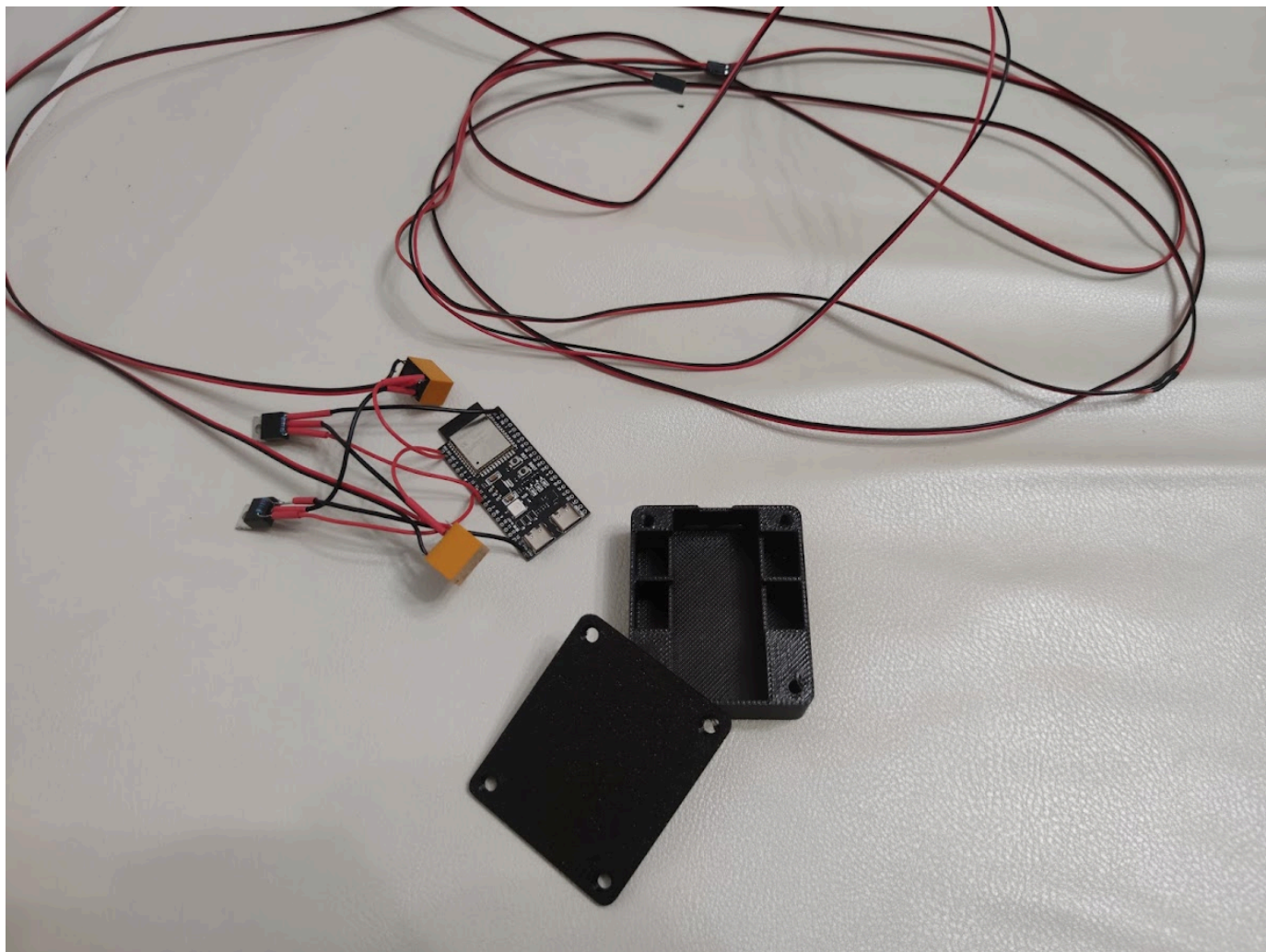
А ещё у меня есть 3D-принтер! Поэтому проектируем такую коробочку:



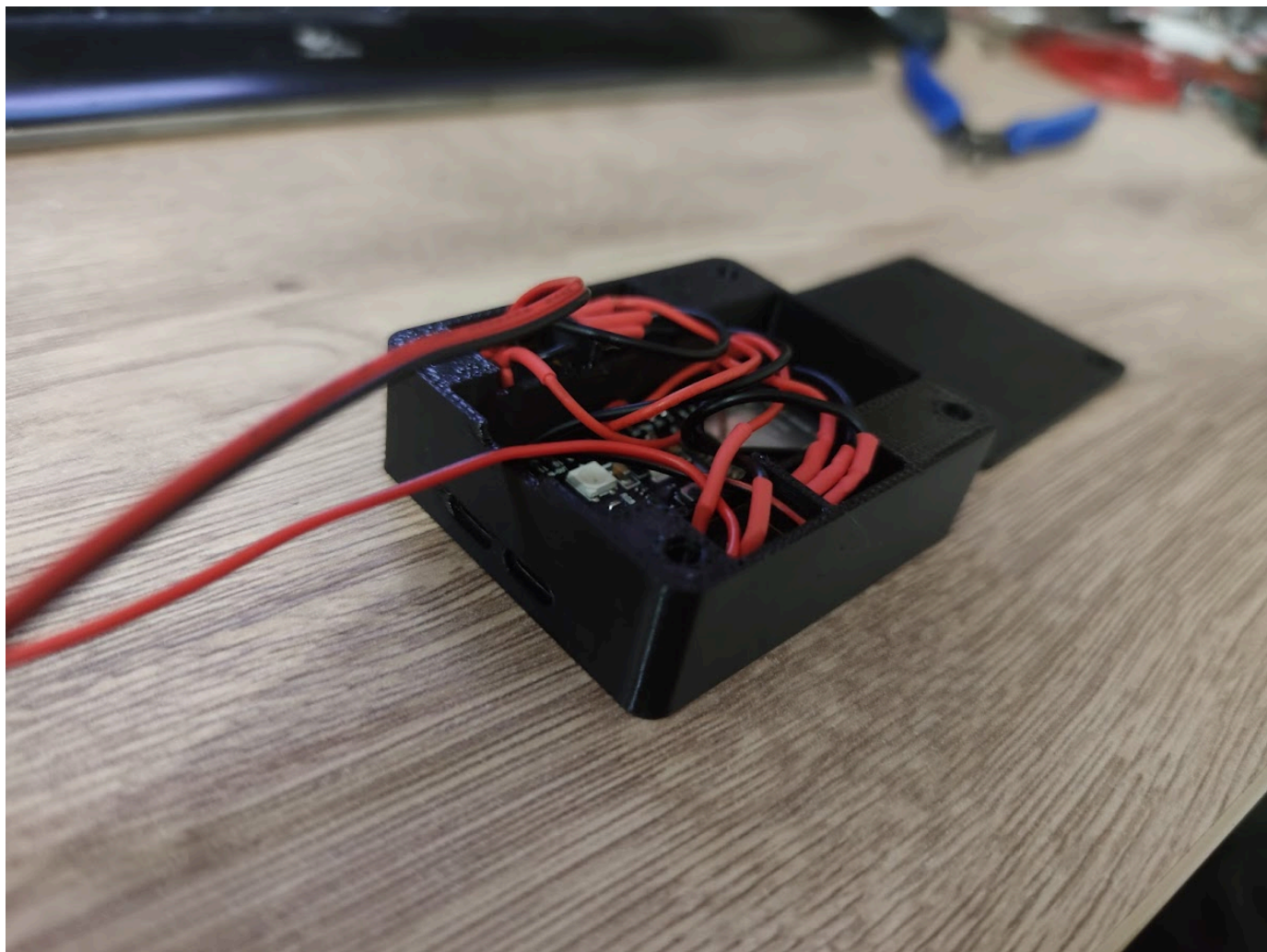
Я делал под свои реле, но это OpenSCAD и там легко можно задать размеры и количество для своих.



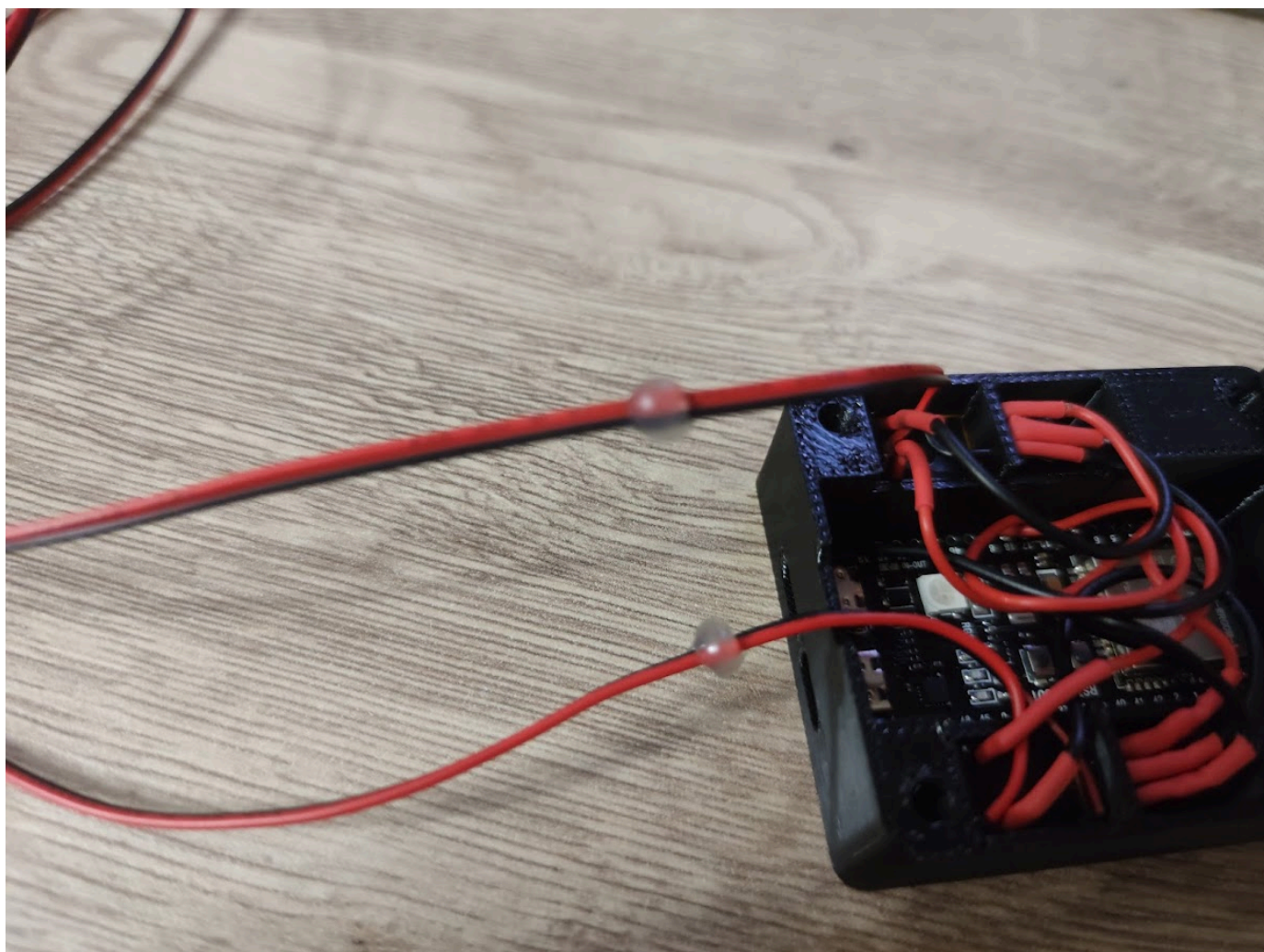
Я всегда делаю корпуса под болтики для компьютерных вентиляторов, они хорошо вкручиваются и крепко держатся в пластике и не нужно впаивать гайки



Воплощаем это в пластике

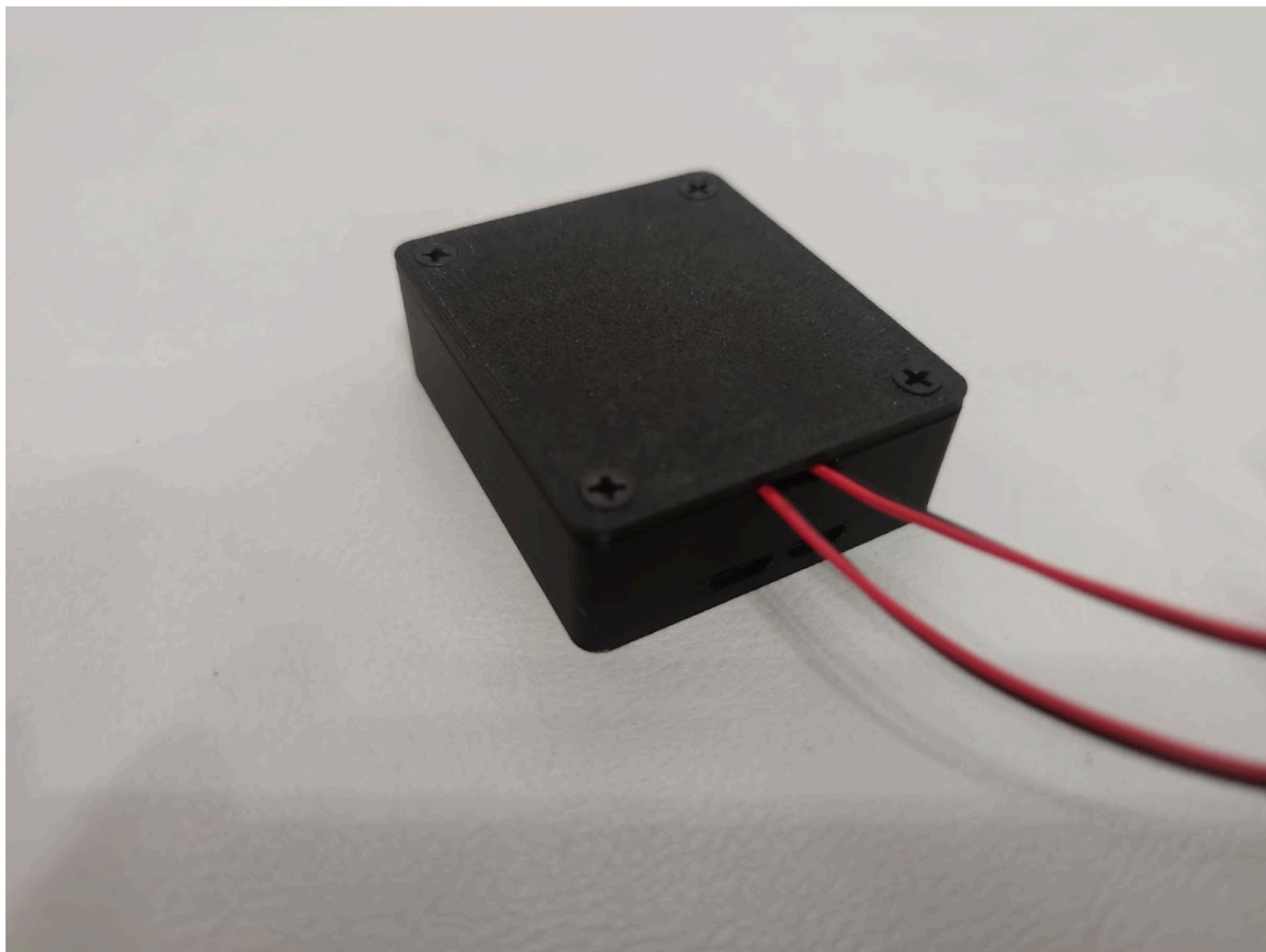


Теперь собираем всё вместе



Термоклеем делаем блямбы на проводах, чтобы не оторвать пайку, если дёрнуть за них

А вот и устройство. Во всяком случае похоже на устройство. Если бы я не знал, как выглядят устройства, я бы подумал, что это, возможно, оно.



Кажется, это устройство

■ Дайте мне это немедленно!

Как обычно, всё, что я сюда выкладываю — [MIT](#), поэтому делайте с этим, что хотите, кроме модифицированной AForge, там [GPL/LGPL](#), но я все исходники выложил и чист перед ними)

Страница проекта: [тут](#).

Но на всякий случай предупреждаю, что я особо не сижу на GitHub, редко туда захожу, не часто там отвечаю, просто выложил и забыл. А сюда пишите, отвечу)

© 2024 ООО «МТ ФИНАНС»

■ [Telegram-канал со скидками, розыгрышами призов и новостями IT](#) 

Дарим панель управления **//ispmanager**

Закажите VPS-сервер на Linux и пользуйтесь панелью бесплатно до конца года

Теги: [ruvds_статьи](#), [DIY](#), [удаленное управление компьютером](#), [remote control](#), [esp32 s3](#)

◆ +142

🔖 260



💬 83

Редакторский дайджест



Присылаем лучшие статьи раз в месяц

Электронпочта



RUVDS.com

VDS/VPS-хостинг. Скидка 15% по коду **HABR15**

Telegram ВКонтакте X



95

0.2

Карма

Рейтинг

@CodeName33

Пользователь

Подписаться



💬 Комментарии 83

Публикации

ЛУЧШИЕ ЗА СУТКИ

ПОХОЖИЕ



Exosphere

15 часов назад

Ещё 10 ошибок авторов Хабра



11 мин



3K



+91



33



62



vital_pavlenko

21 час назад

Больше нет входа в IT. Только выход

🕒 2 мин 👁 74K

📌 +47

🔖 76

💬 310



duran-duran

21 час назад

Трамплин в интернет: как мы ускорили запуск Яндекс Браузера

🕒 6 мин 👁 3.2K

📌 +42

🔖 9

💬 29



Nickmob

16 часов назад

Введение в Angie: краткая история и отличия от Nginx

👉 Простой 🕒 7 мин 👁 2.7K

Ретроспектива

📌 +37

🔖 28

💬 6



Myskat_90

20 часов назад

Распределённый инференс и шардирование LLM. Часть 1: настройка GPU, проброс в Proxmox и настройка Kubernetes

👉 Сложный 🕒 14 мин 👁 1.7K

Тutorial

📌 +33

🔖 46

💬 0



slava_rumin

15 часов назад

Мое производство электроцитов приносит 40 млн в год. Спасибо нейросетям и СССР за конструкторскую школу

👉 Простой 🕒 14 мин 👁 20K

Интервью

📌 +28

🔖 40

💬 48



AlexeyNadezhin

22 часа назад