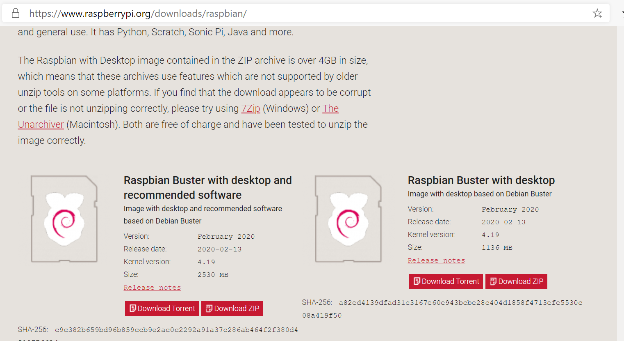
*Разделы:*

1. *Установка среды на малину.*
2. *Подготовка среды.*
3. *Установка необходимых программ и библиотек.*
4. *Настройка репозитория.*
5. *Настройка работы скриптов автоматизации.*
6. **Установка среды.**

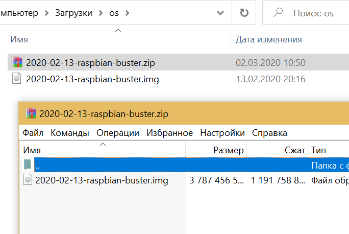
1) Скачиваем Raspbian. Качаем версию с рабочим столом. Учебный софт нам не понадобится, всё необходимое мы установим сами.

<https://www.raspberrypi.org/downloads/raspbian/>



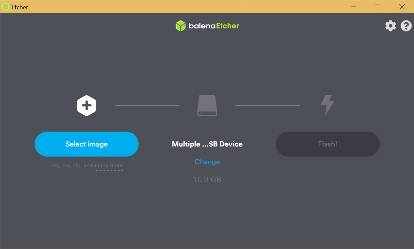
лучше торрент

<https://downloads.raspberrypi.org/raspbian_latest.torrent>

Полученный архив распаковываем, получаем образ формата “.img”

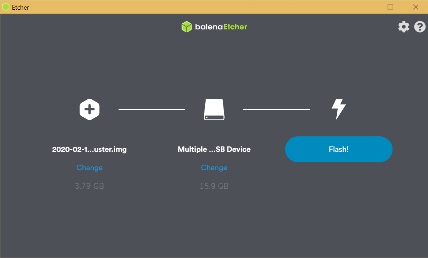
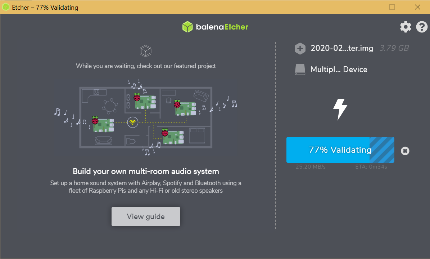
2) Скачайте и запустите Etcher. Это портативная программа.

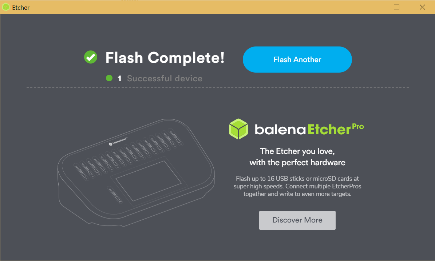
<https://www.balena.io/etcher/>



Выбираем файл образа кнопкой “Select Image”

Запишем через Etcher образ Raspbian на SD карту (при помощи внешнего кард-ридера или встроенного, как у меня на ноутбуке)





Успешно завершена запись образа на карту памяти.

Вставьте SD карту в Raspberry Pi (подключение на оборотной стороне). Убедитесь, что у вас есть в наличии адаптер питания не менее 5V=2A и USB-кабель, поддерживающий этот ток (не менее важно!).



Это минимально допустимое питание для полноценной работы платы! Подключите питание к плате. Загорятся индикаторы питания и доступа к SD карте.



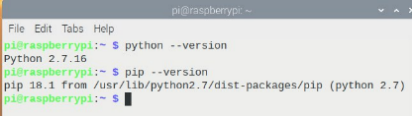
Ожидайте две перезагрузки, пока не встретите экран приветствия.



Готово. Система Raspbian установлена!

1. **Подготовка среды**

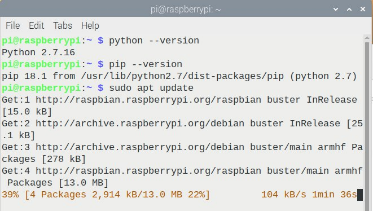
Т.к. мы собираемся использовать современный питон, а не устаревший 2.7, нам необходимо «переучить» Raspbian понимать команды питона так, чтобы вызывался новый интерпретатор.



Вот что мы имеем изначально.

Для начала удалим старый питон.

Выполним обновление репозиториев командой sudo apt update

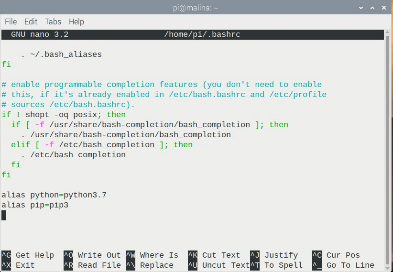


Установите *Python 3.7* командой

sudo apt install python3.7-dev

Откройте файл командой

sudo nano ~/.bashrc



В конец файла вставьте строки

alias python=python3.7

alias pip=pip3

Сохраните файл сочетанием **Ctrl+S**

Выйдите из программы *nano* сочетанием **Ctrl+X**



То, что должно выйти в итоге.

1. **Установка программ и библиотек**

Для поиска нужных библиотек можно пользоваться командой:

sudo apt-cache search {keyword}

**Установка Python:**

sudo apt install python3

sudo apt install python3-pip

Python Essentials:

pip install sip

pip install Lib

pip install matplotlib

pip install numpy

pip install Pillow

pip install picamera

Установка корректной версии OpenCV для работы с изображениями, т.к., «the latest version of openCV doesn't work with RPi.»:

pip3 uninstall opencv-python

pip3 install opencv-python==3.4.6.27

Для загрузки Tensorflow Lite модели:

curl -O http://storage.googleapis.com/download.tensorflow.org/models/tflite/coco\_ssd\_mobilenet\_v1\_1.0\_quant\_2018\_06\_29.zip

unzip coco\_ssd\_mobilenet\_v1\_1.0\_quant\_2018\_06\_29.zip -d ${DATA\_DIR}

rm coco\_ssd\_mobilenet\_v1\_1.0\_quant\_2018\_06\_29.zip

Для установки библиотеки PyQt5 введите команды:

sudo apt-get update

sudo apt-get install qt5-default pyqt5-dev pyqt5-dev-tools

sudo apt-get install python3-pyqt5

sudo apt-get install libqt5quickwidgets5

Для установки *OpenCV:*

sudo apt-get install python3-opencv

sudo pip install opencv-python

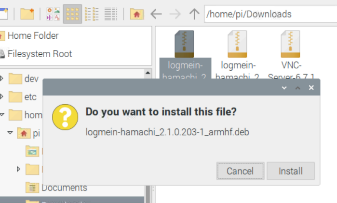
Рекомендую обновить VNC Server на Малине:

<https://www.realvnc.com/en/connect/download/vnc/raspberrypi/>

Для широкого подключения нужно настроить VPN с Малиной. Воспользуйтесь для этого LogMeIn Hamachi – <https://vpn.net>

Установите специфичную для процессора Raspberry Pi версию (ARM HF) по ссылке:

<https://vpn.net/installers/logmein-hamachi_2.1.0.203-1_armhf.deb>

Raspberry умеет устанавливать пакеты Debian прямо из проводника. Поэтому зайдите в папку Downloads и запустите скачанный по ссылке пакет

Настройте VPN сервер на Raspberry Pi. Для этого в терминале введите команду

sudo Hamachi create <network\_id> [password]

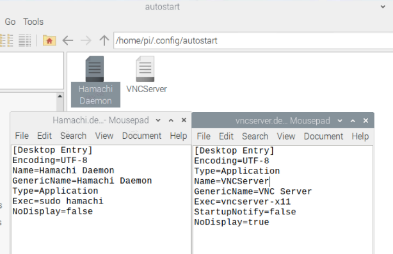
Вместо **<network\_id>** введите имя вашей сети, а вместо **[password]** – пароль к сети.

Для автоматизации запуска VPN сети и VNC сервера, достаточно создать ярлыки с расширением *.desktop* и поместить их по адресу /home/pi/.config/autostart/

Как делать ярлыки, объяснять не буду, достаточно знать, что нужно запускать команды

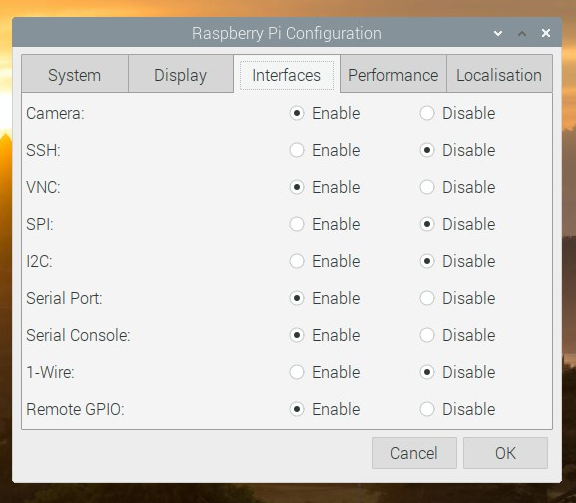
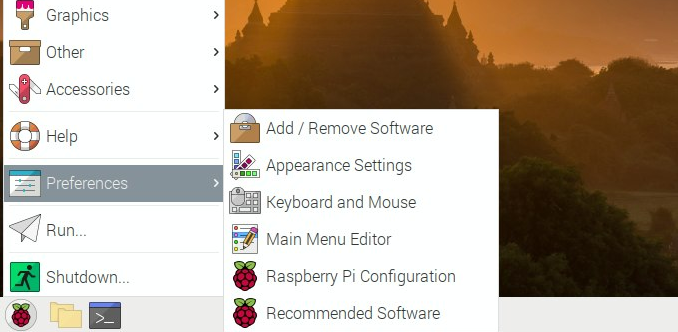
sudo hamachi login – для VPN сети

sudo vncserver-x11 – для VNC сервера



пример ярлыков на автозапуск

Альтернативно можно установить на автозапуск VNC Сервер через настройки Raspbian:

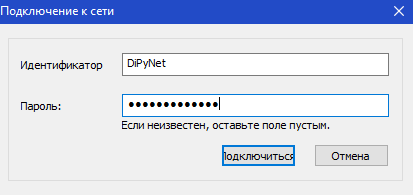
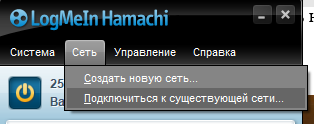


Перезагрузите Raspberry Pi для применения конфигурации



Удалённое подключение к вашей Raspberry Pi нужно осуществлять следую-щим образом:

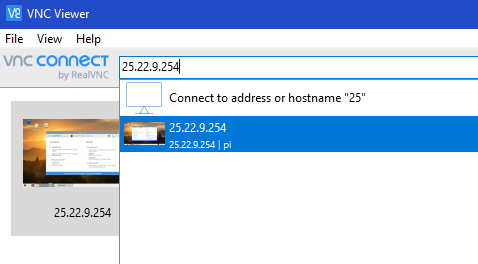
1. На компьютере-клиенте установите аналогичный VPN клиент. Зайдите в вашу VPN сеть.

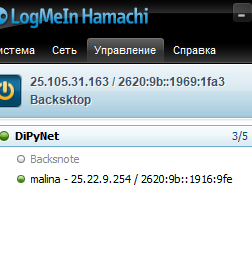


Пример для LogMeIn Hamachi

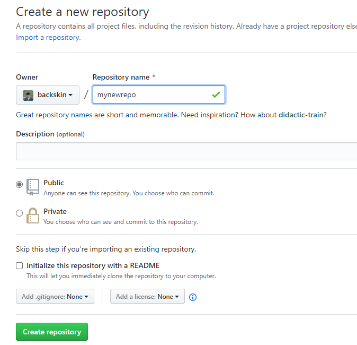
1. Установите VNC-клиент (VNC Viewer) на удалённой стороне по ссылке:

<https://www.realvnc.com/en/connect/download/viewer/>

1.  Проверьте VPN IP-адрес Raspberry и подключитесь по нему через VNC Viewer:



1. **Настройка вашего репозитория**

Создайте репозиторий с вашим проектом на GitHub:

<https://github.com/new>

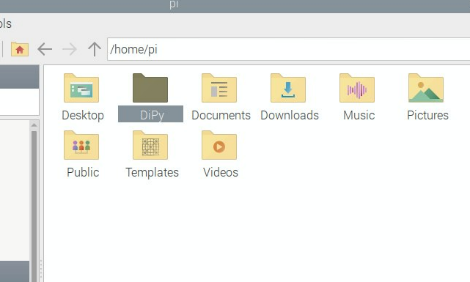
Задайте предпочтительное имя репозитория

В моём случае это – *DiPy*

Затем, на Raspberry Pi, откройте терминал и введите команду:

git clone https://github.com/*YOUR-USERNAME*/*YOUR-REPOSITORY*

Вместо **YOUR-USERNAME** введите имя вашей учётной записи, а вместо **YOUR-REPOSITORY** – наименование созданного репозитория.



У вас появится папка с содержимым вашего репозитория. Рекомендуется редактировать репозиторий исключительно удалённо. Для того, чтобы вовремя обновлять содержимое папки, воспользуйтесь командами:

cd PATH/TO/REPO\_FOLDER

git pull

1. **Автоматизация процессов**

Напишем автоматизированный скрипт на Python, логгирующий температуру в файл:

from gpiozero import CPUTemperature

from time import sleep, strftime, time

import matplotlib.pyplot as plt

def writeLog(cpu):

logname = "CPUtemp\_"+strftime("%Y%m%d")+".log"

log = open("/home/pi/DiPy/resources/templogs/"+logname, "a")

temp = cpu.temperature

print(temp)

log.write("{0} | {1} 'C\n".format(strftime("%H:%M:%S"),str(temp)))

log.close()

def main():

cpu = CPUTemperature()

while True:

writeLog(cpu)

sleep(10)

if \_\_name\_\_ == "\_\_main\_\_":

main()

Создадим

[Unit]

Description=Example systemd service.

[Service]

Type=simple

ExecStart=/bin/bash /usr/bin/test\_service.sh

[Install]

WantedBy=multi-user.target

Скопируем в /etc/systemd/system/

sudo cp myservice.service /etc/systemd/system/myservice.service

sudo chmod 644 /etc/systemd/system/myservice.service

sudo systemctl start myservice

sudo systemctl status myservice

sudo systemctl enable myservice

После перезагрузки для проверки работы фоновой службы введите ещё раз команду:

sudo systemctl status myservice