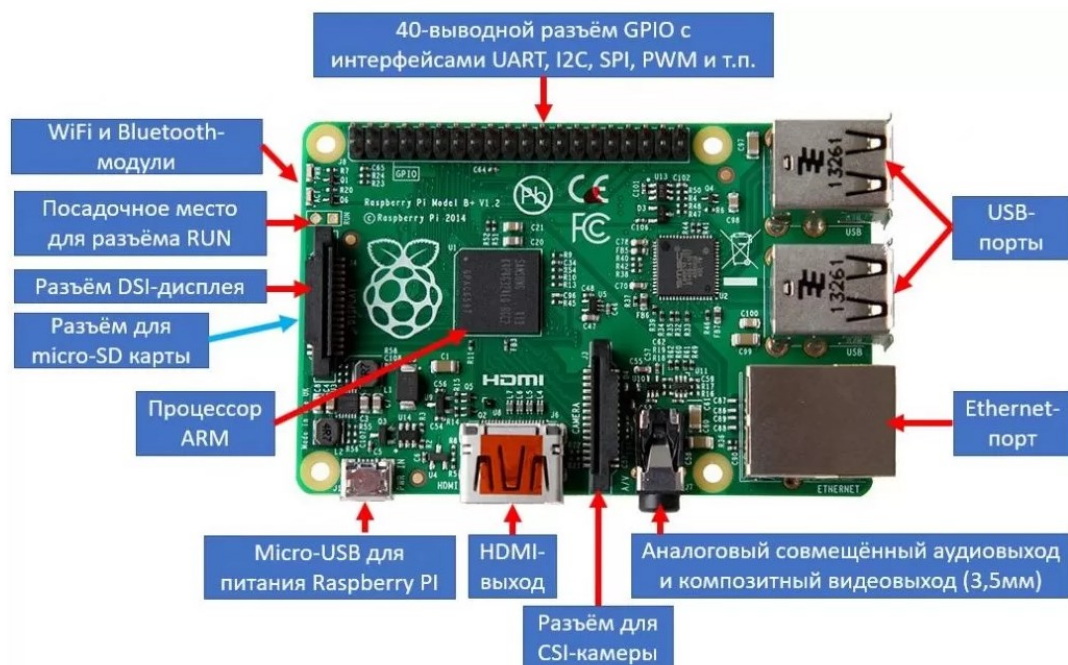


## Лабораторная работа № 3.3

### Создание системы видеонаблюдения с функцией распознавания движения на Raspberry Pi

#### Теоретическая часть:

Raspberry Pi – это миниатюрный одноплатный компьютер, который с лёгкостью поместится на ладони взрослого человека. Несмотря на свои скромные размеры, плата имеет высокую производительность, что позволяет ей выйти на один уровень со стационарными ПК. Изначально Raspberry Pi была разработана, как учебное пособие по информатике. Но сама идея оказалась настолько удачной, что за несколько лет мини-компьютер стал популярен в очень широких кругах. С течением времени Raspberry Pi пережила несколько модификаций, каждая из которых отличалась от предшественника каким-либо параметром. Такой подход позволил регулировать стоимость изделия в зависимости от потребностей пользователя, что также положительно сказалось на популярности устройства. Вся линейка Raspberry Pi применяет процессоры с APM-архитектурой, которая зарекомендовала себя с лучшей стороны. На рисунке №1 показан внешний вид одной из популярных плат Raspberry Pi B+.



На сегодняшний день (период 2012-2019гг.) существует 11 разновидностей Raspberry Pi. Последние версии оснащены беспроводными WiFi и Bluetooth модулями, расширяющими границы применения мини-пк в области Ethernet-технологий. Ниже приведена сравнительная таблица, в которой отражены особенности каждой модификации с указанием некоторых технических данных.

Модификация	Процессор	Тактовая частота	Количество ядер	Объем ОЗУ	Количество GPIO	Количество USB	Поддержка Ethernet	Поддержка WiFi	Поддержка Bluetooth	Год выпуска
B	ARM1176JZ-F	700 МГц	1	512 МБ	26	2	√			2012
A	ARM1176JZ-F	700 МГц	1	256 МБ	26	1				2013
B+	ARM1176JZ-F	700 МГц	1	512 МБ	40	4	√			2014
A+	ARM1176JZ-F	700 МГц	1	256 МБ	40	1				2014
2B	ARM Cortex-A7	900 МГц	4	1 ГБ	40	4	√			2015
Zero	ARM1176JZ-F	1 ГГц	1	512 МБ	40	1				2015
3B	Cortex-A53 (ARM v8)	1,2 ГГц	4	1 ГБ	40	4	√	802.11n	4.1	2016
Zero W	ARM1176JZ-F	1 ГГц	1	512 МБ	40	1		802.11n	4.0	2017
3B+	Cortex-A53 (ARM v8)	1,4 ГГц	4	1 ГБ	40	4	√	802.11n	4.2	2018
3A+	Cortex-A53 (ARM v8)	1,4 ГГц	4	512 МБ	40	1		802.11n	4.2	2018
4B	Cortex-A72 (ARM v8)	1,5 ГГц	4	1, 2, 4 ГБ	40	4	√	802.11n	5.0	2019

Как видно из вышеприведенной таблицы, даже самая младшая модель в линейке имеет вполне серьёзные характеристики, учитывая то, что это одноплатный компьютер размером чуть больше кредитной карты.

На рисунке №2 изображена последняя на момент написания статьи модификация Raspberry Pi 4B, запущенная в продажу в июне 2019г. Она оснащена дополнительным графическим процессором VideoCore VI (OpenGL ES 3.x), а также аппаратным декодером 4Kp60 для воспроизведения HEVC видео. Два порта microHDMI с возможностью пропускать сигнал до 4K, позволяют подключить одновременно два монитора.

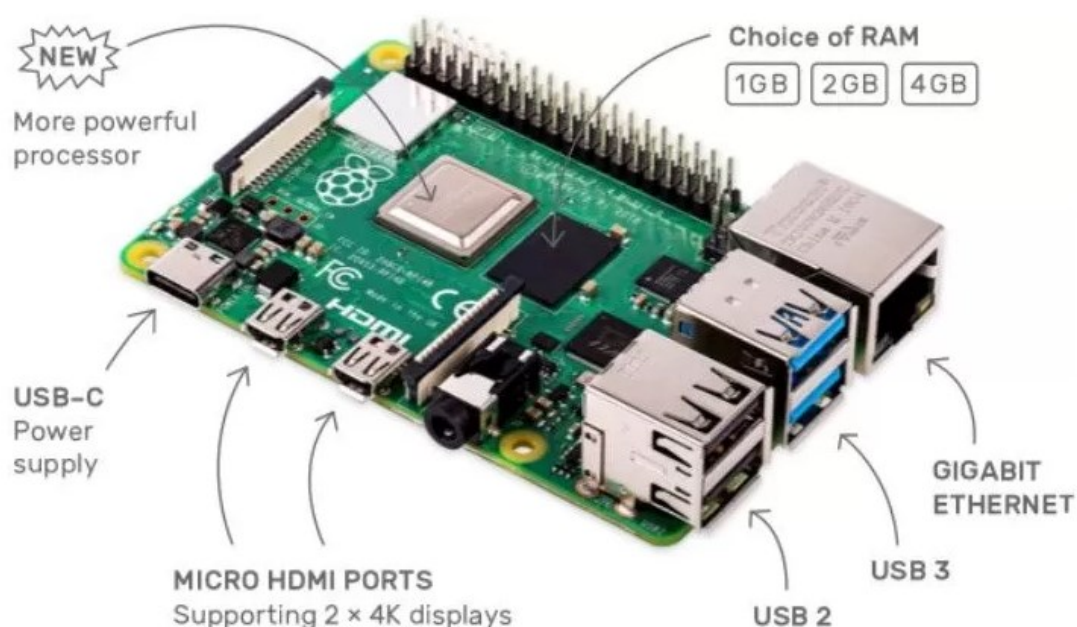


Рисунок №2 – внешний вид Raspberry Pi 4B

Основной отличительной чертой Raspberry Pi от обычных компьютеров, является наличие программируемых портов ввода-вывода GPIO. С помощью них можно управлять различными устройствами и принимать телеметрию с различного рода датчиков.

## Операционные системы

Хоть Raspberry Pi внешне может напомнить нам Arduino, он всё-таки использует кардинально другой метод функционирования. Данная плата, как и обычный ПК, работает под управлением одной из специализированных операционных систем. В зависимости от области применения или личных симпатий, каждый может выбрать для себя свою. Ниже приведён перечень наиболее популярных «операционок» для Raspberry Pi с их кратким описанием.

Raspbian – данная операционная система в 2015 году была представлена как основная для Raspberry Pi. Она по максимуму оптимизирована для процессоров с APM-архитектурой и достаточно активно продолжает развиваться. Основой операционной системы является Debian GNU/Linux. Среда рабочего стола состоит из LXDE (среда для UNIX и других POSIX-совместимых систем типа Linux и BSD), а также менеджера окон Openbox (бесплатный менеджер для X Window System). В состав дистрибутива входят: программа компьютерной алгебры Mathematica; модифицированная версия Minecraft Pi; урезанная версия Chrome.

Debian – операционная система с открытым исходным кодом. В состав Debian входит более 59000 пакетов уже скомпилированного ПО. Система использует ядро Linux или FreeBSD. В стандартный дистрибутив включены: среда рабочего стола GNOME с набором наиболее популярных программ, таких как Firefox, LibreOffice, Evolution, и прочий набор для работы с мультимедиа. Также есть возможность установки образов с используемыми средами рабочих столов KDE, Xfce, LXDE, MATE и Cinnamon.

Ubuntu – система основана на Debian GNU/Linux. По популярности Ubuntu занимает первое место среди дистрибутивов Linux, предназначенных для web-серверов. В состав дистрибутива входят: программа для просмотра Интернет; офисный пакет, программы для коммуникации и т.д.

Fedora – эта операционная система основана на дистрибутиве Linux от известной фирмы Red Hat. В состав дистрибутива входят LibreOffice, Mozilla Firefox, а также другое ПО, которое можно дополнительно установить через Центр Приложений GNOME.

Arch Linux – это свободно распространяемый дистрибутив GNU/Linux общего назначения. Особенностью данной системы является отсутствие графического установщика, что может изрядно потренировать навыки ярых исследователей Linux.

Gentoo Linux – один из популярных дистрибутивов GNU/Linux с гибкой технологией управления пакетами. В системе предусмотрена возможность максимальной оптимизации под конкретное аппаратное решение. Алгоритм управления пакетами даёт возможность легко реализовать как рабочую станцию, так и сервер.

RISC OS – операционная система специально разрабатывалась для процессоров с архитектурой ARM. Особенности ядра RISC OS позволяют системе производить ускоренный запуск за счёт хранения данных в ПЗУ. Такой подход также помогает защитить данные при различного рода сбоях и влияния вредоносного ПО.

OpenELEC – это программный комплекс для организации домашнего кинотеатра под управлением GNU/Linux.

OSMC – ещё один комплекс для реализации домашнего кинотеатра.

В сети Интернет, помимо перечисленных операционных систем, можно найти ещё множество модификаций для самых различных предназначений. Но так как Raspbian является основной средой для Raspberry Pi, то в дальнейшем будем опираться именно на неё.

Для установки операционной системы необходимо подключить к плате Raspberry Pi минимальный набор периферии, а именно: монитор, клавиатуру и мышь. Далее, необходима SD-карта с записанным образом Raspbian. Именно с неё и будет производиться установка.

Для того, чтобы записать образ на карту памяти, её необходимо вставить в компьютер и отформатировать в системе FAT32. Сделать это можно как стандартными средствами Windows, так и сторонними программами, например – SD Memory Card Formatter. После чего, скачиваем дистрибутив операционной системы с официального сайта Raspberry. Для неопытных пользователей, доступна упрощённая версия установщика NOOBS. По окончании загрузки, архив необходимо распаковать в корень карты памяти. На этом подготовительный этап окончен.

Вставляем карту памяти в плату Raspberry Pi (клавиатура, мышь и монитор уже подключены) и подаём питание через разъём micro-USB. Начинается установка Raspbian, которая длится порядка 10 минут. В это время от пользователя практически ничего не требуется кроме самых простых и интуитивно понятных действий, таких как выбор языка, ввод пароля и т.п. На завершающем этапе появиться меню, в котором можно выбрать тип пользовательского интерфейса (консольный или графический). Выбираем графический и завершаем установку нажатием кнопки Finish. Система попросит перезагрузиться и как следствие запуститься уже в более привлекательном виде.

## Работа с GPIO

Как уже говорилось ранее, основной отличительной чертой Raspberry Pi от обычного ПК, является наличие на плате портов общего назначения GPIO (General-purpose input/output). Пользователю доступна возможность управления этими выводами, а это значит, что к Raspberry Pi можно подключать дисплеи, кнопки, датчики, реле и прочие электронные модули, которыми можно манипулировать на своё усмотрение.

Внешне GPIO выполнен в виде двухрядной штыревой колодки с шагом 2,54мм, которая расположена на краю платы. Ранние модели, такие как B и A содержат 26 выводов, а более современные – 40. На рисунке №3 показан внешний вид портов общего назначения для платы Raspberry Pi 3B+ с указанием нумерации выводов.



Рисунок №3 – выводы GPIO



Как видно из вышеприведенного рисунка, на колодке помимо самих GPIO выведены штырьки с напряжениями 3,3V, 5V, а также выводы GND. Некоторые GPIO имеют альтернативные функции, назначение которых указано в синих блоках. К тому же нельзя нарушать нагрузочные способности порта, чтобы не вывести Raspberry Pi из строя.

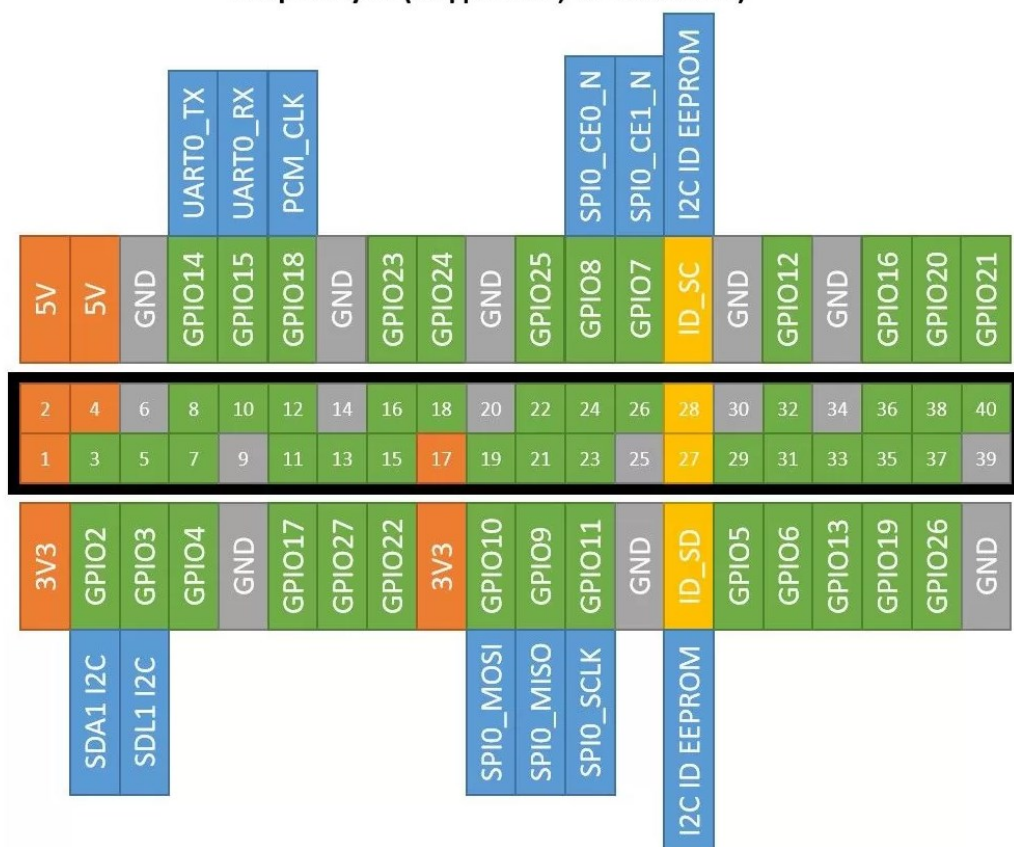


Рисунок №4 – распиновка GPIO в платах Raspberry Pi

Следует помнить, что GPIO работает с напряжением 3,3V и максимальным током нагрузки 50mA на один вывод. Это означает, что любое превышение указанных параметров негативно скажется на работоспособности платы, поэтому лучше использовать гальваническую развязку между GPIO и внешним исполнительным устройством. То же самое относится и ко входным цепям, к которым применяются резистивные делители и всевозможные преобразователи уровней. На рисунке №5 показан пример правильного и неправильного подключения базовых элементов.

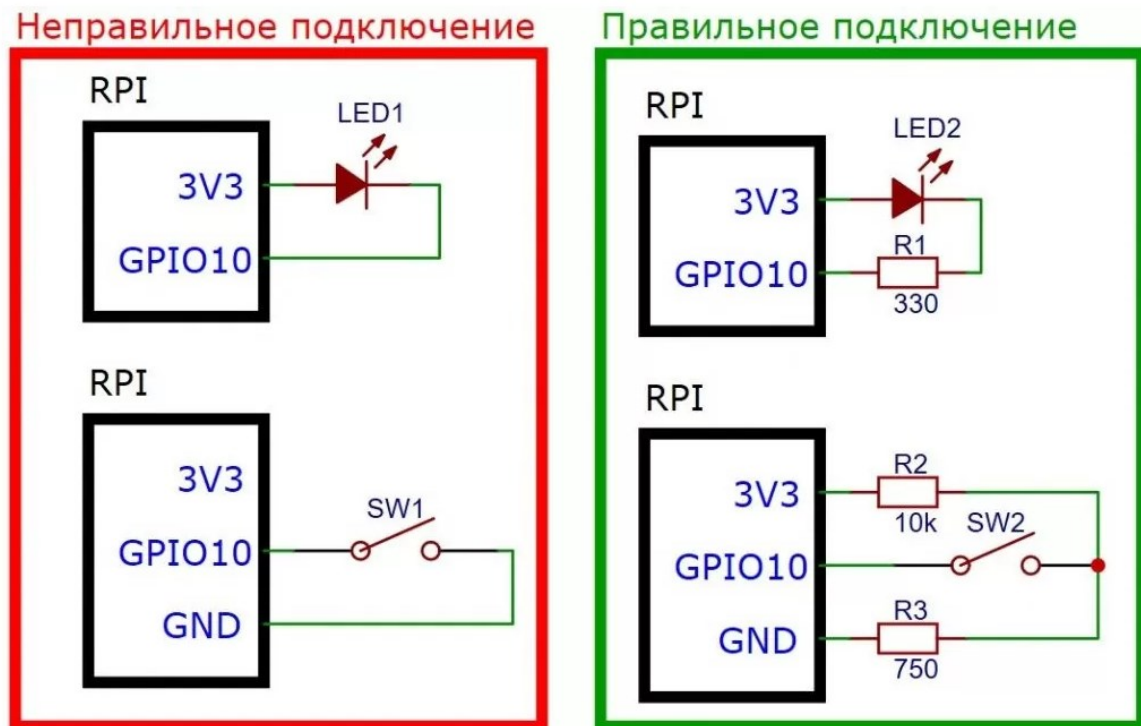


Рисунок №5 – примеры правильного и неправильного подключения

В левой части рисунка прямое подключение светодиода приведёт к превышению максимально допустимого тока 50mA. Это, в свою очередь, выведет GPIO10 из строя. В правой части рисунка добавлен ограничительный резистор, который будет удерживать ток в допустимых рамках. Что касается кнопки, то может возникнуть ситуация, когда GPIO10 ошибочно будет сконфигурирован на выход, и её нажатие приведёт к прямому соединению 3,3V и GND. При добавлении резисторов R2 и R3 все выводы будут гарантировано защищены от перегрузок. Исходя из вышеизложенного можно сделать вывод, что мелочится в элементах защиты не стоит, так как работоспособность Raspberry Pi гораздо важнее кучки дешёвых резисторов.

## Практическая часть:

Используя Raspberry Pi и камеру, в этой лабораторной работе мы сделаем систему видеонаблюдения, которая имеет множество настраиваемых параметров и функцию определения движений.

В данном проекте на Raspberry Pi установлена ОС Raspbian. Для создания системы видеонаблюдения используется Motion - это бесплатное программное обеспечение с широкими возможностями настройки для создания систем видеонаблюдения. Можно настроить видеонаблюдение для системы безопасности, наблюдением за птицами или

домашними животными, создавать таймлапс видео и многое другое. Основные возможности Motion:

- Запись видео или фотографий при обнаружении активности или через регулярные промежутки времени;
- Просмотр прямой трансляции с камер;
- Запуск скриптов при обнаружении активности;
- Регистрация активности в базе данных;
- Полностью настраиваемые маски для задания областей на изображении, в которых определяется движение или для которых нужно обеспечить режим конфиденциальности;
- Полная поддержка tls (https) с аутентификацией для веб-доступа и видеопотока.

В качестве источника видеопотока в данном проекте будут выступать камеры Raspberry Pi видеофайлы и сервер в качестве которого выступит компьютер.

### Подключение RPI к компьютеру

- 1) Карта памяти с загруженной ОС уже вставлена в RPI
- 2) Подключите желтый шнур Lan в разъем RPI и заднюю панель компьютера



- 3) Достаньте камеру RPI из коробки, и подключите шлейфом в нужный разъем, для этого

Поднимите чёрный боковые рамки разъема отмеченного на картинке и подключите туда камеру

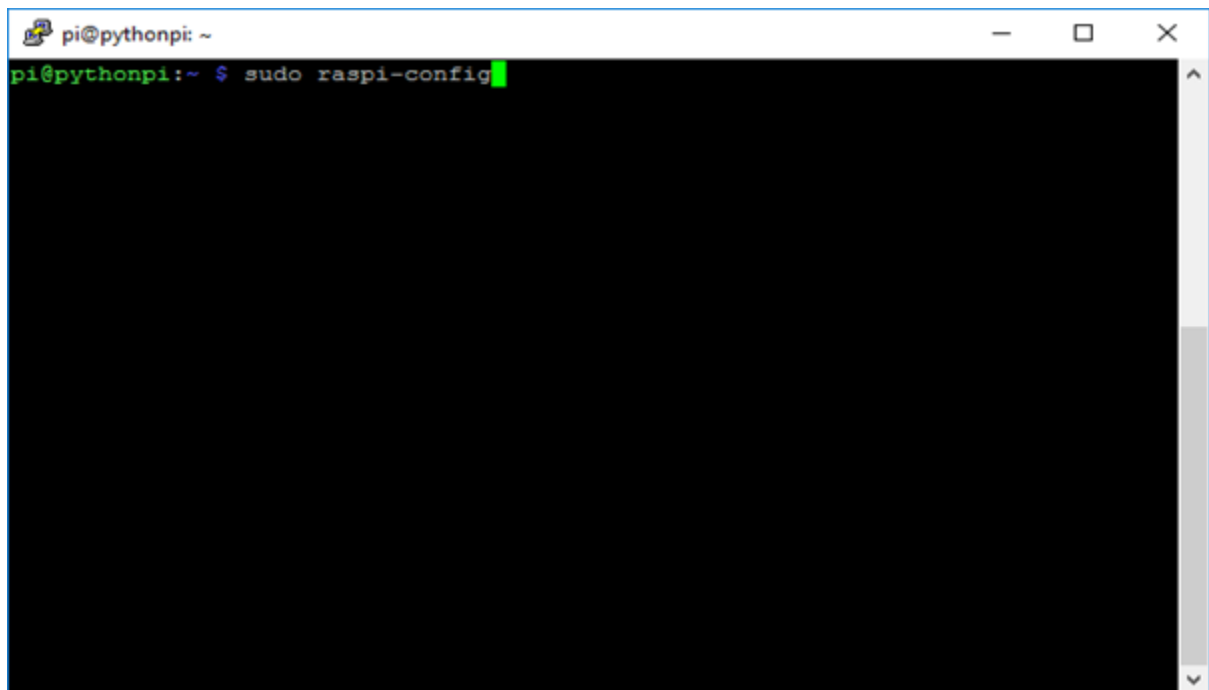


Подключите питание к микрокомпьютеру в разъем отмеченный на картинке

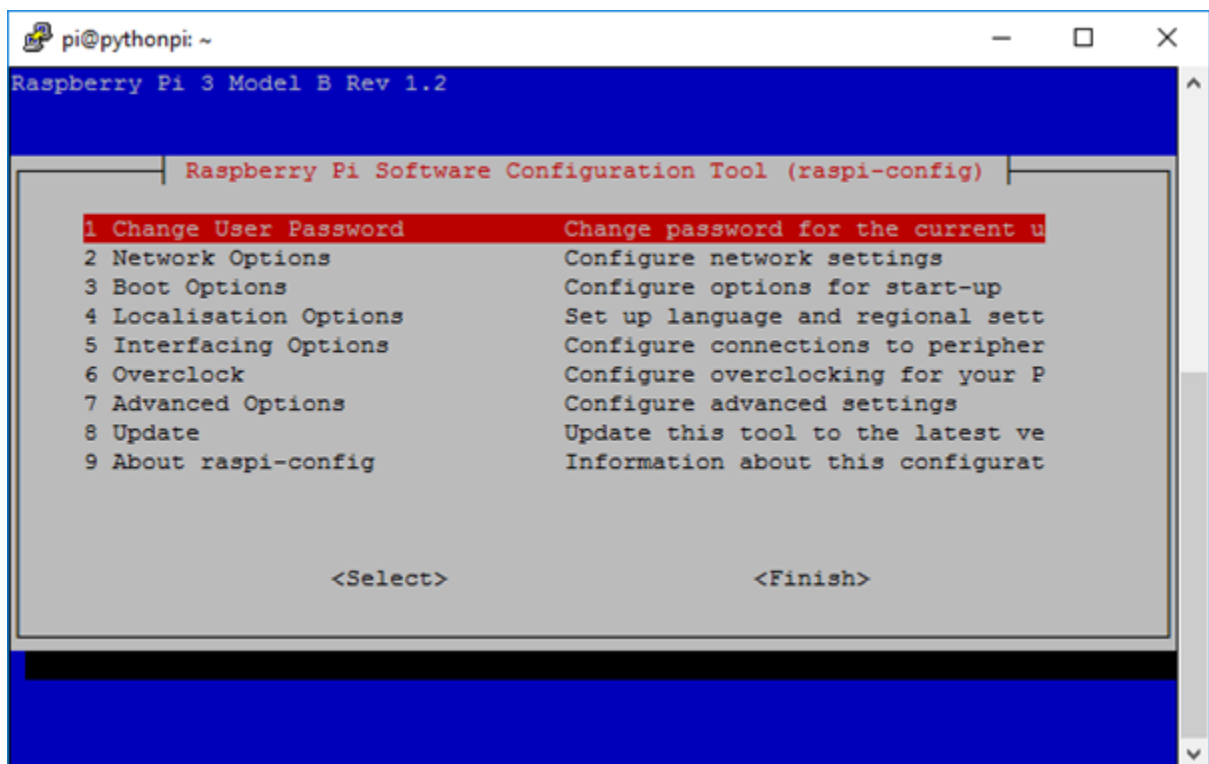


- 4) Проверьте подключение камеры, индикатор на ней должен гореть красным, если этого не произошло, проверьте подключение камеры в разъёме
- 5) Ждём секунд 10-15 и подключаем через Lan к компу напрямую
- 6) Открываем приложение putty на компьютере
- 7) Подключаемся через Putty SSH по адресу rasp40 (40 это номер вашего набора. Необходимо поменять на свой номер)  
 Login: pi  
 Password: rbcrrbcrbc  
 Настроиваем VNC сервер:  
 Для этого наберём в командной строке `sudo raspi-config` и нажмём enter:

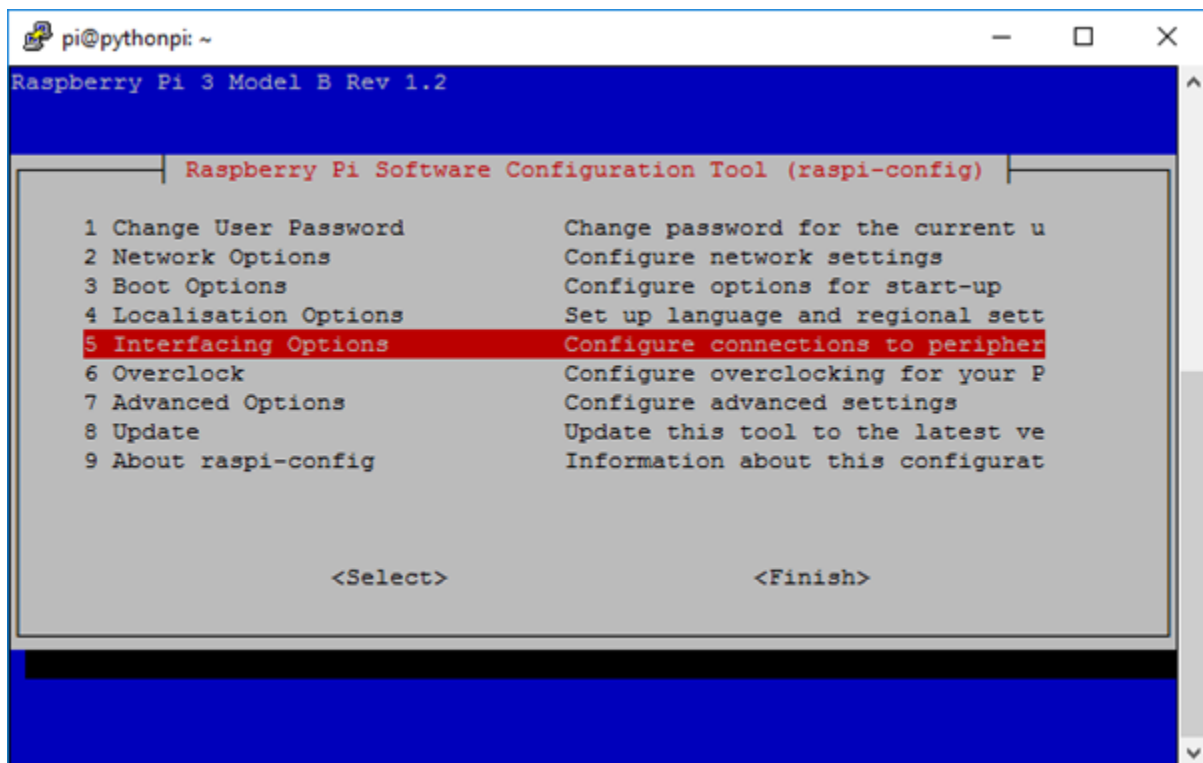




Появится меню настроек Raspberry:



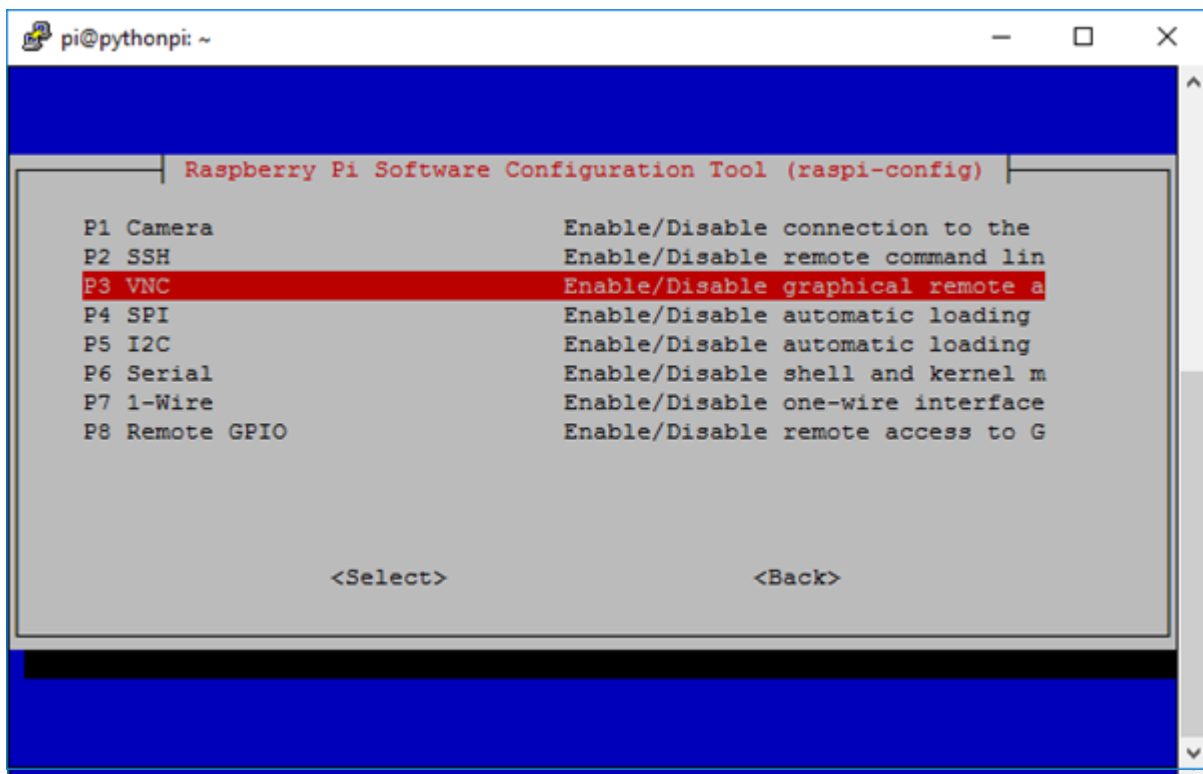
Клавишей "вниз" курсора клавиатуры выбираем пункт 5 Interfacing Options:



Нажимаем enter, откроется меню включения/выключения интерфейсов, выбираем P3 VNC, снова нажимаем enter:

Меню задаст вопрос о включении/выключении VNC сервера, отвечаем <Да>:

Далее нажимаем <OK>:



Нажимаем стрелку клавиатуры вправо, нажимаем <Ok>:

Мы снова в главном меню, нажимаем два раза стрелку вправо на клавиатуре, нажимаем <Finish>:



На вопрос о перезагрузке отвечаем <Yes>:

## Установка и настройка Motion

Прежде всего убедитесь, что Raspberry Pi подключена к интернету через локальную сеть или Wi-Fi.

Запустите консоль и обновите систему выполнив команду:

```
sudo apt-get update
```

Установка Motion:

```
sudo apt-get install motion
```

Чтобы демон Motion всегда работал, нужно отредактировать файл /etc/default/motion и прописать в нём start\_motion\_daemon=yes

```
sudo nano /etc/default/motion
```

pi@raspberrypi: ~

```
GNU nano 2.2.6                               File: /etc/default/motion

# set to 'yes' to enable the motion daemon
start_motion_daemon=yes
```

Для сохранения файла нажмите "CTRL + X", затем «Y» и Enter.

Теперь нужно установить права доступа для каталога (/var/lib/motion/), в котором Motion сохраняет все видеозаписи и файлы изображений. Motion нужно установить владельцем этого каталога. Для этого выполним следующую команду:

```
sudo chown motion:motion /var/lib/motion/
```

Если данные права не будут установлены, Вы получите ошибку, когда проверите статус сервиса Motion с помощью команды:

```
sudo service motion status
```

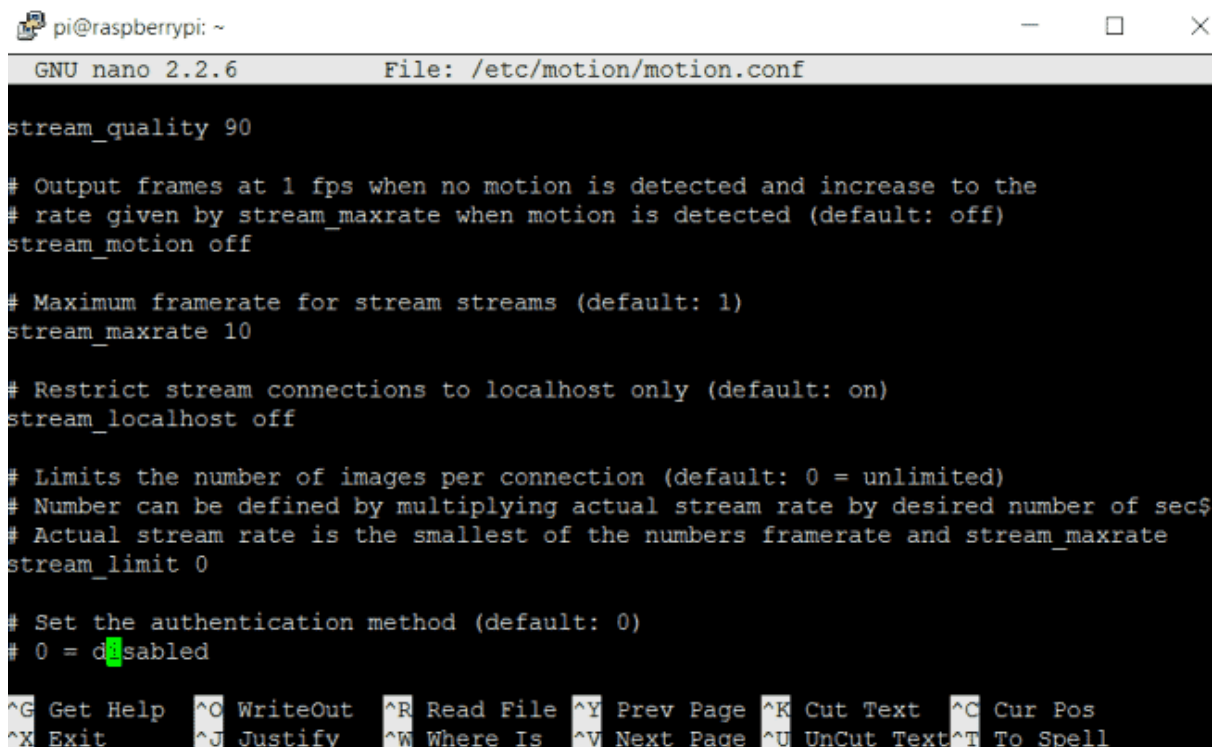
```
pi@raspberrypi: /var/lib/motion
pi@raspberrypi:/var/lib/motion: $ sudo service motion status
● motion.service - LSB: Start Motion detection
   Loaded: loaded (/etc/init.d/motion)
   Active: active (exited) since Sat 2017-02-04 06:12:49 UTC; 1min 7s ago
   Process: 1301 ExecStop=/etc/init.d/motion stop (code=exited, status=0/SUCCESS)
   Process: 1347 ExecStart=/etc/init.d/motion start (code=exited, status=0/SUCCESS)

Feb 04 06:13:08 raspberrypi motion[1367]: [1] [ERR] [ALL] put_picture: Can't write picture to file /var/lib/motion/0
Feb 04 06:13:08 raspberrypi motion[1367]: Thread is going to finish due to this fatal error:
Feb 04 06:13:08 raspberrypi motion[1367]: [1] [NTC] [ALL] motion_loop: Thread exiting
Feb 04 06:13:08 raspberrypi motion[1367]: [1] [NTC] [STR] stream_stop: Closing motion-stream listen socket & active m
Feb 04 06:13:08 raspberrypi motion[1367]: [1] [NTC] [STR] stream_stop: Closed motion-stream listen socket & active m
Feb 04 06:13:08 raspberrypi motion[1367]: [1] [NTC] [VID] vid_close: Closing video device /dev/video0
Feb 04 06:13:09 raspberrypi motion[1367]: [0] [NTC] [ALL] main: Threads finished
Feb 04 06:13:09 raspberrypi motion[1367]: [0] [NTC] [STR] httpd_run: motion-httpd - Finishing
Feb 04 06:13:09 raspberrypi motion[1367]: [0] [NTC] [STR] httpd_run: motion-httpd Closing
Feb 04 06:13:09 raspberrypi motion[1367]: [0] [NTC] [STR] motion_web_control: motion-httpd thread exit
Feb 04 06:13:10 raspberrypi motion[1367]: [0] [NTC] [ALL] main: Motion terminating
```

По умолчанию Motion настроен на потоковую передачу только в localhost, т.е. видеопоток будет доступен на самой Raspberry Pi, но не будет доступен по сети. Если нужно, в файле конфигурации (/etc/motion/motion.conf) это можно перенастроить, задав "off" для параметра stream\_localhost. Команда для редактирования файла настроек:

```
sudo nano /etc/motion/motion.conf
```





```
pi@raspberrypi: ~
GNU nano 2.2.6 File: /etc/motion/motion.conf

stream_quality 90

# Output frames at 1 fps when no motion is detected and increase to the
# rate given by stream_maxrate when motion is detected (default: off)
stream_motion off

# Maximum framerate for stream streams (default: 1)
stream_maxrate 10

# Restrict stream connections to localhost only (default: on)
stream_localhost off

# Limits the number of images per connection (default: 0 = unlimited)
# Number can be defined by multiplying actual stream rate by desired number of seconds
# Actual stream rate is the smallest of the numbers framerate and stream_maxrate
stream_limit 0

# Set the authentication method (default: 0)
# 0 = disabled

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Параметром `stream_port` задаётся порт для потокового видео:

```
# The mini-http server listens to this port for requests (default: 0 = disabled)
stream_port 8081
```

Для сохранения файла нажмите "CTRL + X", затем «Y» и Enter.

Если в файле настроек производились изменения а Motion уже запущен, его нужно перезапустить командой:

```
sudo service motion restart
```

Первоначальная настройка закончена. Запускаем Motion с помощью команды:

```
sudo /etc/init.d/motion start
```

В браузере откройте страницу с адресом Вашей Raspberry Pi и портом 8081:

```
localhost:8081
```

При открытии страницы в браузере Вы увидите прямую трансляцию с веб-камеры.

Вы всегда можете запустить, остановить, перезапустить и получить статус сервиса Motion, для этого используются следующие четыре команды:

```
sudo /etc/init.d/motion start
sudo /etc/init.d/motion stop
sudo service motion restart
sudo service motion status
```

Иногда для устранения неполадок можно перезагрузить Raspberry Pi, используя команду:

```
sudo reboot
```

Когда всё заработало, можно переходить к дальнейшей настройке. В файле конфигурации все настройки очень хорошо прокомментированы, с указанием их значений по умолчанию.

Помимо получения прямой трансляции, Motion записывает и сохраняет видеофайлы всякий раз, когда обнаруживает движение. Вы можете настроить частоту кадров, качество и разрешение (ширина и высота) изображения, изменив следующие параметры:

```
# Image width (pixels). Valid range: Camera dependent, default: 352
```

```
width 640
```

```
# Image height (pixels). Valid range: Camera dependent, default: 288
```

```
height 480
```

```
# Maximum number of frames to be captured per second.
```

```
# Valid range: 2-100. Default: 100 (almost no limit).
```

```
framerate 5
```

Как уже было упомянуто, при обнаружении движения начинается запись видео. Следующими параметрами настраивается, сколько изображений сохранять до и после обнаружения движения:

```
# Specifies the number of pre-captured (buffered) pictures from before motion
```

```
# was detected that will be output at motion detection.
```

```
# Recommended range: 0 to 5 (default: 0)
```

```
pre_capture 0
```

```
# Number of frames to capture after motion is no longer detected (default: 0)
```

```
post_capture 0
```

Очень важный параметр - это picture\_output, ему можно устанавливать значения "on", "off", "first" или "best":

- "off" - отключает запись изображения, это значение по умолчанию;
- "on" - обычное выделение;
- "first" - сохраняет только первое изображение, в котором замечено движение;
- "best" - сохраняет изображение с наиболее измененными пикселями. Требуется немного больше ресурсов от процессора в сравнении с "first".

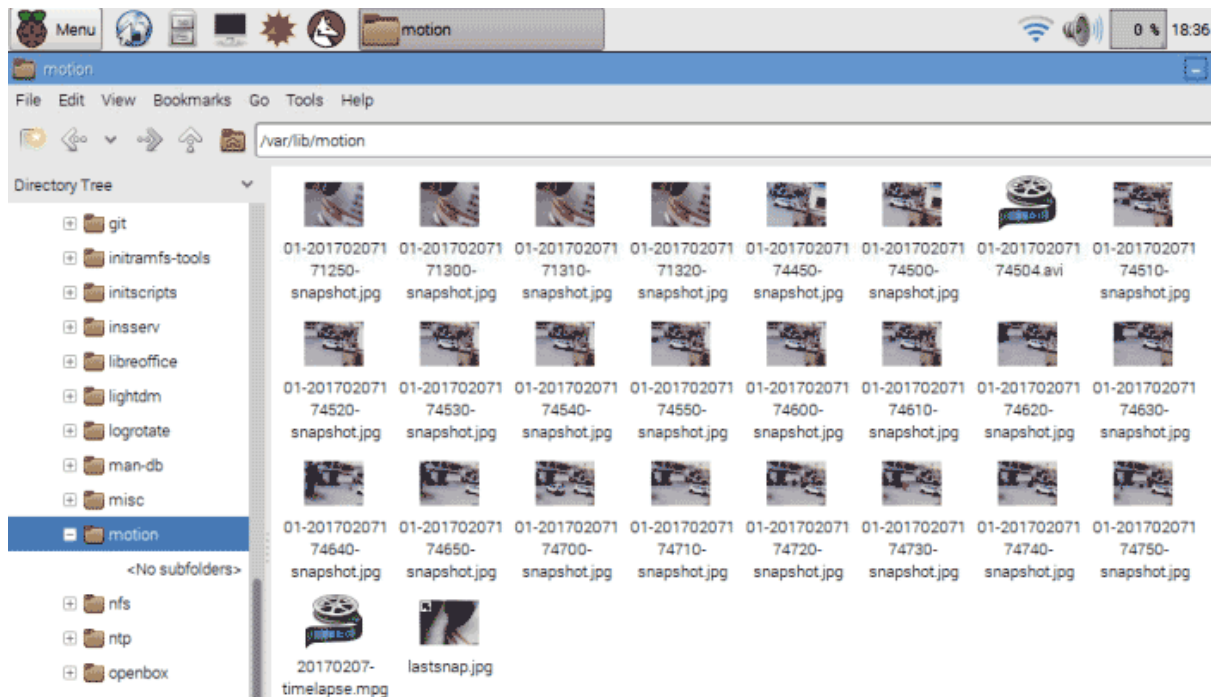
Вам необходимо отключить данную функцию, поэтому переключите этот параметр на off

Motion по умолчанию делает слишком много снимков, когда обнаруживает движение, и Вы получите очень много файлов при каждом даже небольшом движении. Поэтому было бы хорошо или отключить (off) или установить в "best".

```
output_pictures best
```

Для сохранения снимков через определенные промежутки времени, настройте параметр:

```
# Make automated snapshot every N seconds (default: 0 = disabled)
snapshot_interval 0
```



Параметр `target_dir` задаёт путь, по которому сохраняются снимки и видео:

```
# Target base directory for pictures and films
# Recommended to use absolute path. (Default: current working directory)
target_dir /var/lib/motion
```

Не забудьте установить права доступа, установив Motion в качестве владельца этого каталога, как объяснялось ранее.

По умолчанию Motion записывает видео непрерывно. Вместо одного большого файла можно разбивать видео на несколько частей определённой длительности. Максимальная продолжительность (в секундах) задаётся следующим параметром:

```
# Maximum length in seconds of a movie
# When value is exceeded a new movie file is created. (Default: 0 = infinite)
max_movie_time 10
```

Также можно настроить, какое количество пикселей или кадров должны измениться для срабатывания триггера обнаружения движения:

```
# Threshold for number of changed pixels in an image that
# triggers motion detection (default: 1500)
```

```
threshold 1500
# Picture frames must contain motion at least the specified number of frames
# in a row before they are detected as true motion. At the default of 1, all
# motion is detected. Valid range: 1 to thousands, recommended 1-5
minimum_motion_frames 5
```

По умолчанию Motion сохраняет журналы событий в «syslog», местоположение можно перенастроить следующим параметром:

```
# Use a file to save logs messages, if not defined stderr and syslog is used. (default: not
defined)
logfile /tmp/motion.log
```

Не забудьте сохранить и перезапустить сервис Motion после изменения любых настроек. Чтобы сохранить файл motion.conf, нажмите клавиши "CTRL + X", затем "Y", а затем Enter.

Рассмотрена только очень небольшая часть основных настроек, ещё можно настраивать яркость, контрастность, битрейт, триггеры, аутентификацию по паролю и много других параметров. Если интересно, изучайте файл конфигурации и экспериментируйте с настройками.

## **Альтернативный вариант сделать тоже самое:**

Необходимо создать файл питона:

```
sudo touch /home/camera.py
```

Далее необходимо установить библиотеку:

```
sudo apt-get install pip
```

```
pip install picamera
```

Заходим в файл:

```
sudo nano /home/camera.py
```

Заполняем файл следующим кодом:

```
import time
import picamera

#Создаем объект камеры
camera = picamera.PiCamera()
#Устанавливаем разрешение кадра
camera.resolution = (720, 576)
#Устанавливаем длительность видео (в секундах)
```



```
video_length = 15
#Путь к папке для сохранения видео
output_folder = "/path/to/folder/"
#Генерируем имя файла на основе даты и времени
filename = time.strftime("%Y-%m-%d_%H-%M-%S") + ".h264"
#Полный путь к файлу
output_path = output_folder + filename
#Запускаем запись видео
camera.start_recording(output_path)
#Ожидаем заданное количество секунд
time.sleep(video_length)
#Останавливаем запись видео
camera.stop_recording()
#Закрываем объект камеры
camera.close()
```

В этом файле необходимо заменить "/path/to/folder/" на путь к папке, в которой будут сохраняться видео.

Далее необходимо запустить скрипт через команду:

```
python3 camera.py
```

В вашей указанной папке должен появиться файл с видео

## Усложняем задачу:

- 1) Сделать сохранение папки на локальный компьютер в определенную папку
- 2) Развернуть ftp сервер (или предпочтительнее собственное облако, например, Nextcloud) на локальном компьютере или на том же raspberry pi и отправлять сохраненную папку туда
- 3) Добавить функцию уведомления о сохранении нового файла в уже имеющегося телеграмм бота

\*\*\*

- 4) Создать из 30 секундных видео файлов стримовый поток и вывести его на веб страницу