

ООО «Прикладная робототехника»

Офис: Москва, Угрешская улица, 2С35, офис 510

Email: support@robotgeeks.ru

URL: www.robotgeeks.ru

Телефон: +7 (495) 142-02-35



Инструкция по установке программного обеспечения

Конструктор программируемых моделей инженерных систем



Артикул: AR-DEK-STR

ООО «Прикладная робототехника»

Москва, 2021 год

Описание набора:

Образовательный набор «Конструктор программируемых моделей инженерных систем. Расширенный " предназначен для проведения учебных занятий по электронике и схемотехнике с целью изучения наиболее распространенной элементной базы, применяемой для инженерно-технического творчества учащихся и разработки учебных моделей роботов. Набор позволяет учащимся на практике освоить основные технологии проектирования робототехнических комплексов на примере учебных моделей роботов, а также изучить основные технические решения в области кибернетических и встраиваемых систем.



Конструктор программируемых моделей инженерных систем позволяет учащимся наглядно изучать основы программирования на разных стадиях подготовки:

- 1) программирование в блочно-графических средах Scratch и Blockly
- 2) программирование в текстовом редакторе Arduino IDE с использованием инструментариев языка C
- 3) программирование с использованием средств разработки языка JavaScript
- 4) программирование с использованием средств разработки языка Python
- 5) программирование с использованием средств разработки ROS

Представленные выше средства программирования являются свободно распространяемыми и предоставляют возможность скачивания с соответствующих ресурсов и применения в некоммерческих целях и образовательном процессе. Возможность какого-либо другого применения указана в лицензионном соглашении к вышеуказанным продуктам. Внимание: при использовании свободно распространяемых программных продуктов обязательно следуйте требованиям лицензионного соглашения.

Программирование в блочно-графических средах Scratch и Blockly

Конструктор программируемых моделей инженерных систем в состав включает программируемый контроллер, выполненный в соответствии с архитектурой открытой аппаратной и программной платформы Arduino, и соответствует контроллеру Arduino Mega 2560. В связи с этим становится возможным использовать все инструментарии и средства доступные для программирования данной модели контроллера.

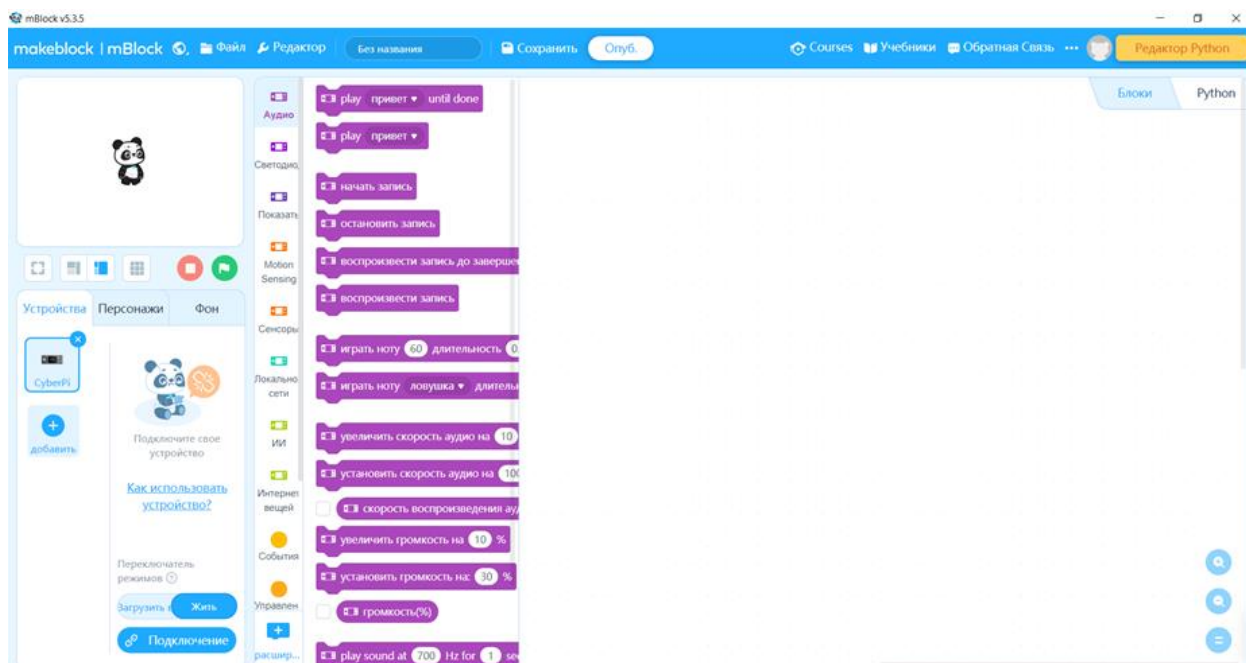
Одним из распространенных средств разработки программного обеспечения для контроллеров Arduino являются среды разработки блочно-графического типа Scratch и Blockly. Существует множество различных свободно распространяемых редакторов для разработки программ с использованием Scratch и Blockly, например – ArduBlock, mBlock и др.

Инструкция по установке среды mBlock:

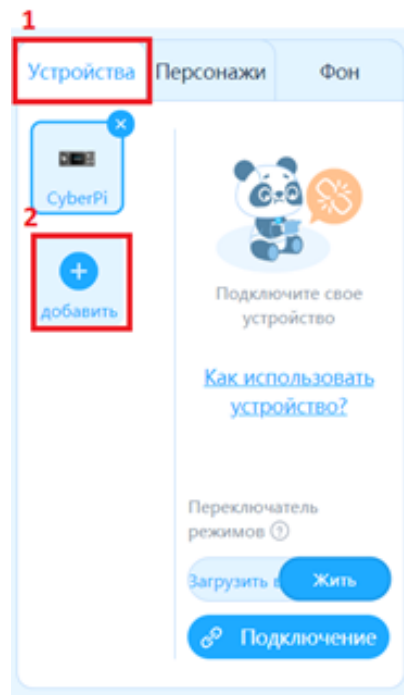
1) Для скачивания приложения mBlock необходимо воспользоваться ссылкой <https://mblock.makeblock.com/en-us/download/>.

Помимо версии приложения для Mac и Windows доступна также веб-версия программы mBlock web version, которая доступна и для Linux. Далее будет использоваться версия приложения для ОС Windows.

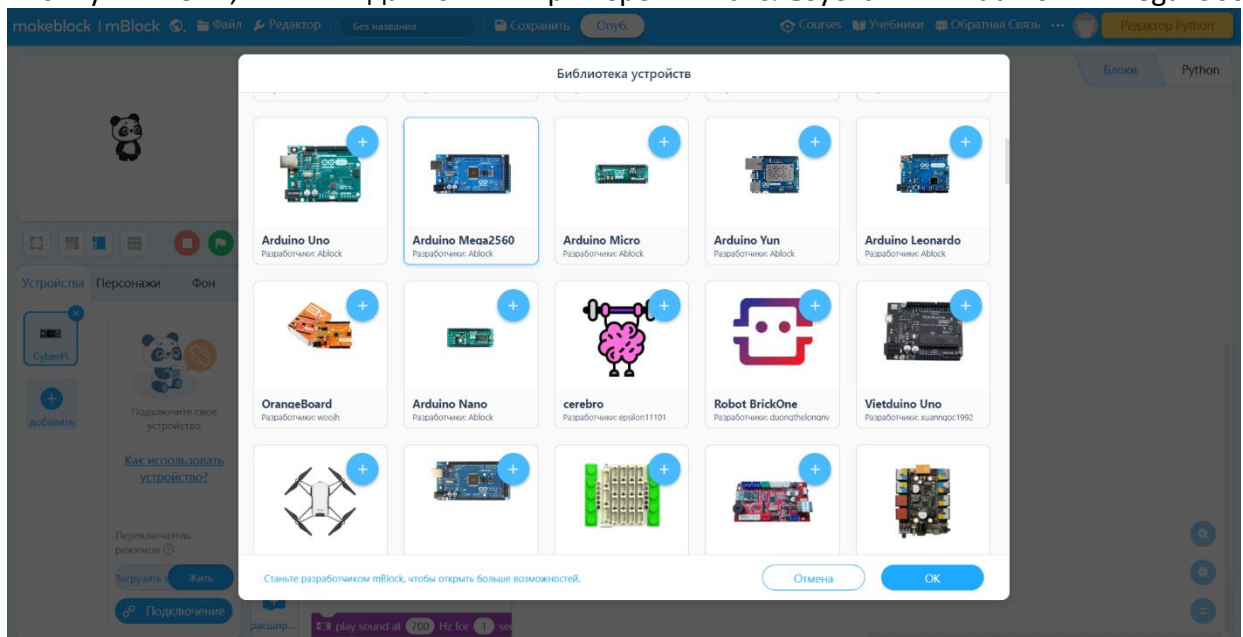
2) После установки и запуска приложения откроется следующее окно:



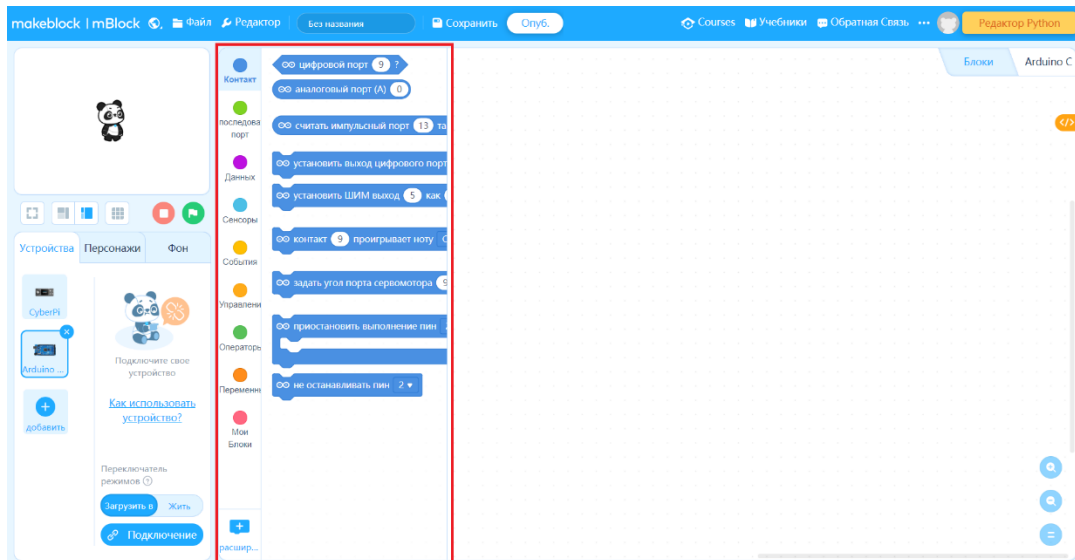
Для работы с Arduino переходим во вкладку «Устройства» и нажимаем кнопку «Добавить»:



В открывшемся окне «Библиотека устройств» выбираем нужную плату и нажимаем кнопку «ОК», в данном примере используется Arduino Mega2560:



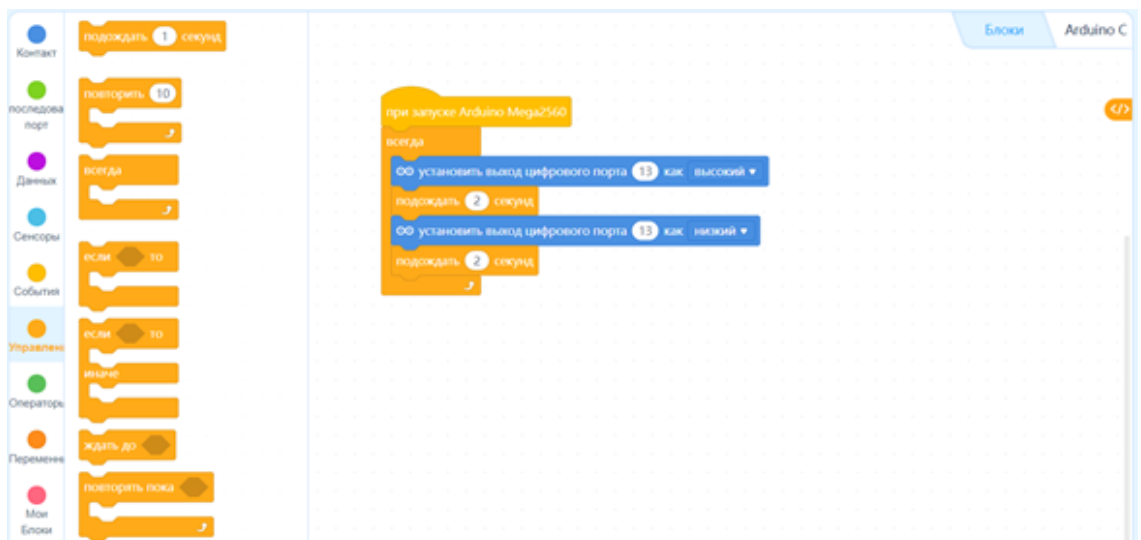
После выбора соответствующей платы главное окно приложения будет иметь следующий вид:



В выделенном столбце представлены библиотека блоков, доступных для выбранной платы. Каждый из блоков имеет описание команды, за выполнение которой он отвечает, а также различные параметры, например номер порта или время задержки.

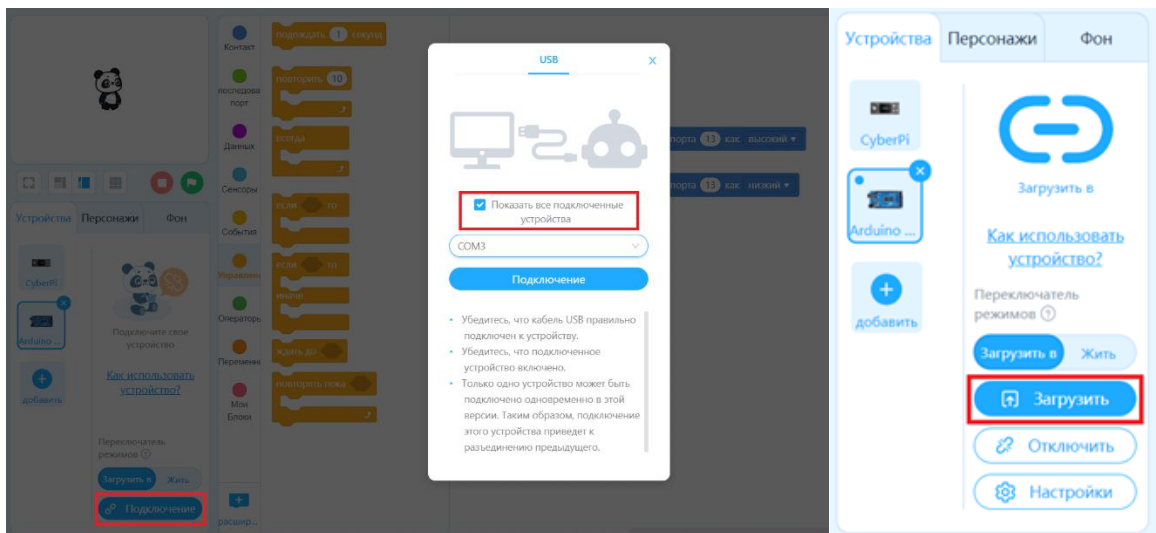
Пример программы для Arduino в mBlock:

В качестве примера рассмотрим моргание светодиодом на плате. Для написания этой программы в рабочую область, расположенную справа от библиотеки блоков, необходимо поместить следующие блоки:



В результате выполнения данной программы светодиод на плате должен мигать с интервалом 2 секунды.

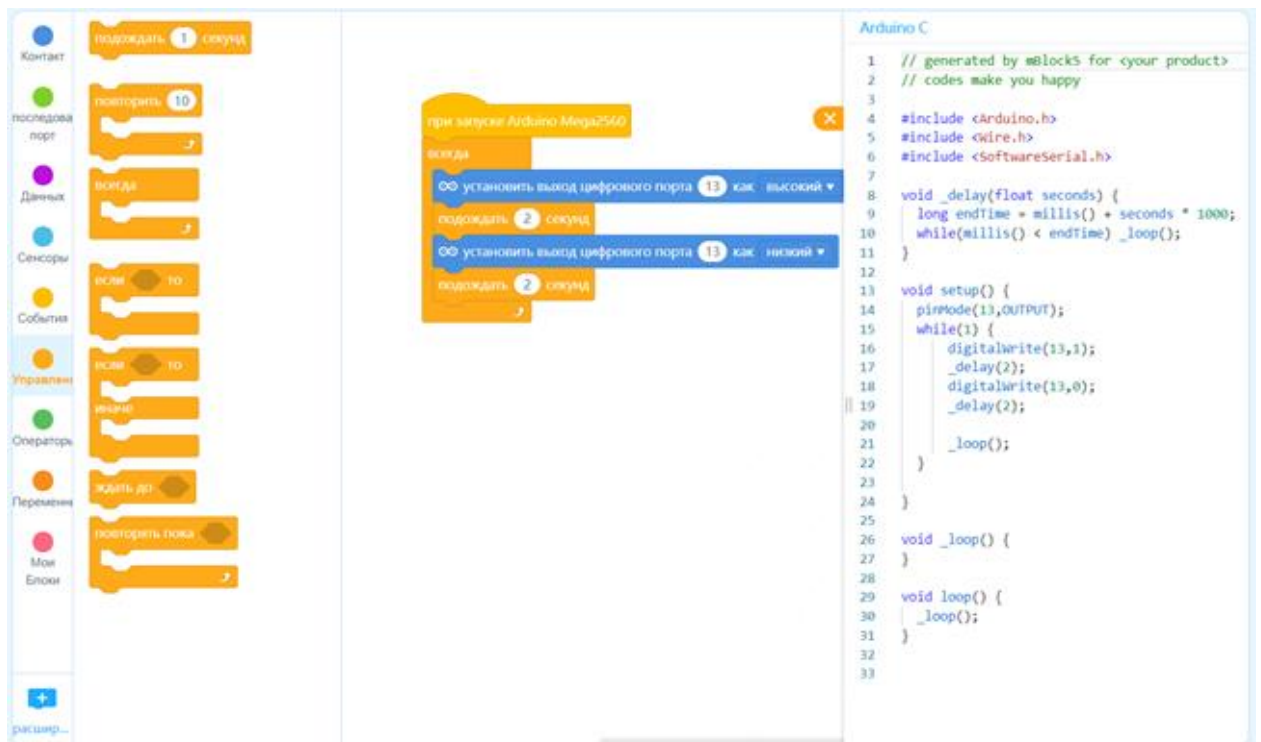
Для загрузки составленной программы на плату необходимо во вкладке «Устройства» нажать на кнопку «Подключение» и в появившемся окне активировать пункт «Показать все подключенные устройства», приложение автоматически определит порт, к которому в данный момент подключена плата:



После нажатия кнопки «Подключение» во вкладке «Устройства» нажимаем кнопку «Загрузить»:

После успешного окончания загрузки программы наблюдаем на плате мигание светодиода с интервалом 2 секунды.

Для получения доступа к сгенерированному С-коду нажимаем кнопку «</>», расположенную справа от рабочей области:



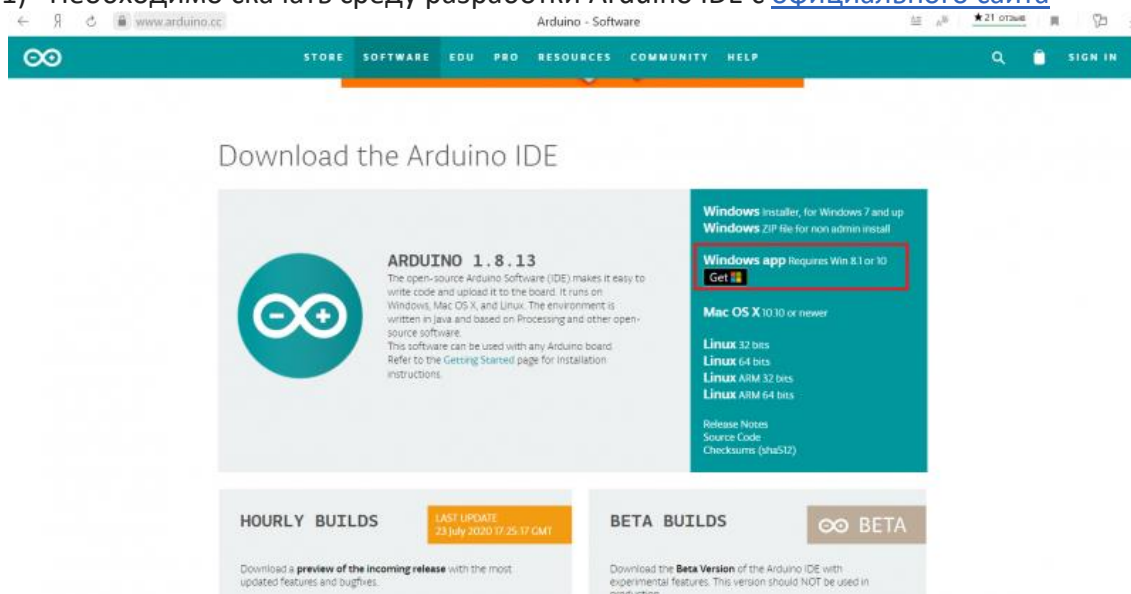
Программирование в текстовом редакторе Arduino IDE с использованием инструментариев языка C

Среда разработки Arduino IDE является свободно распространяемой средой, доступной для скачивания с официального сайта - <https://www.arduino.cc/en/software>.

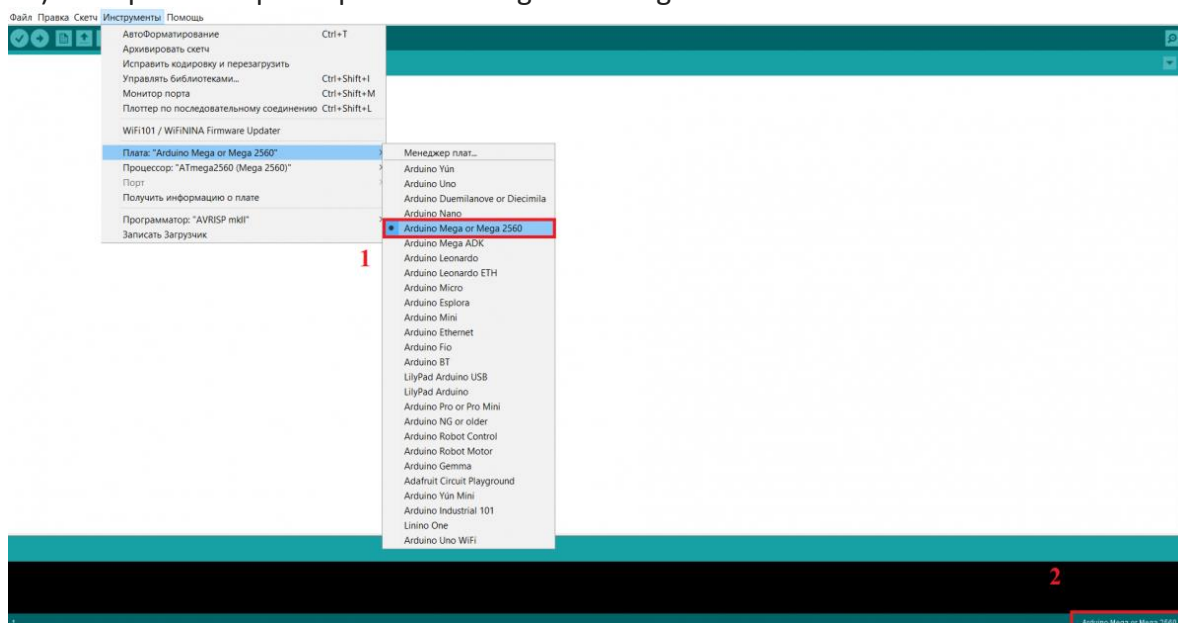
Среда разработки Arduino IDE предназначена для работы с линейкой различных контроллеров Arduino, а также устройств, выполненных в соответствии с основными требованиями спецификации и архитектуры данных контроллеров. В состав набора Конструктор программируемых моделей инженерных систем входят два таких устройства – Контроллер программируемых моделей инженерных систем и Универсальный вычислительный модуль DXL-IoT с набором периферийных плат.

Установка среды разработки Arduino IDE:

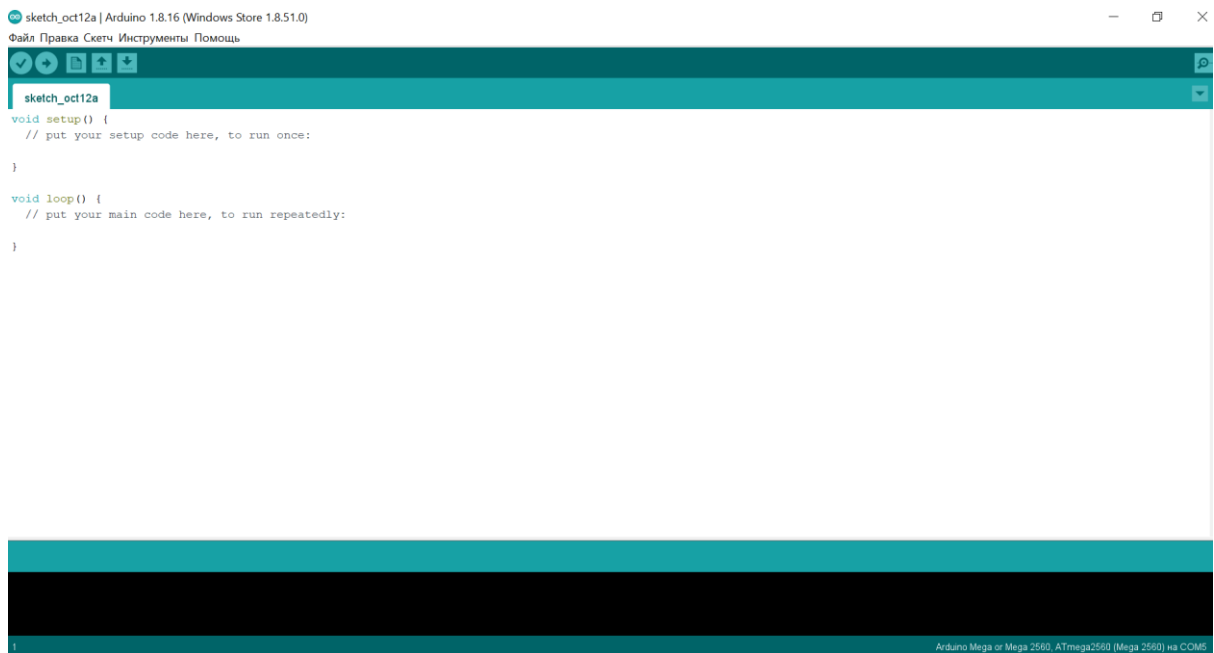
1) Необходимо скачать среду разработки Arduino IDE с [официального сайта](https://www.arduino.cc/en/software)



2) Выбрать контроллер Arduino Mega или Mega 2560 как показано ниже



3) В открывшемся рабочем пространстве можно писать код на языке C/C++

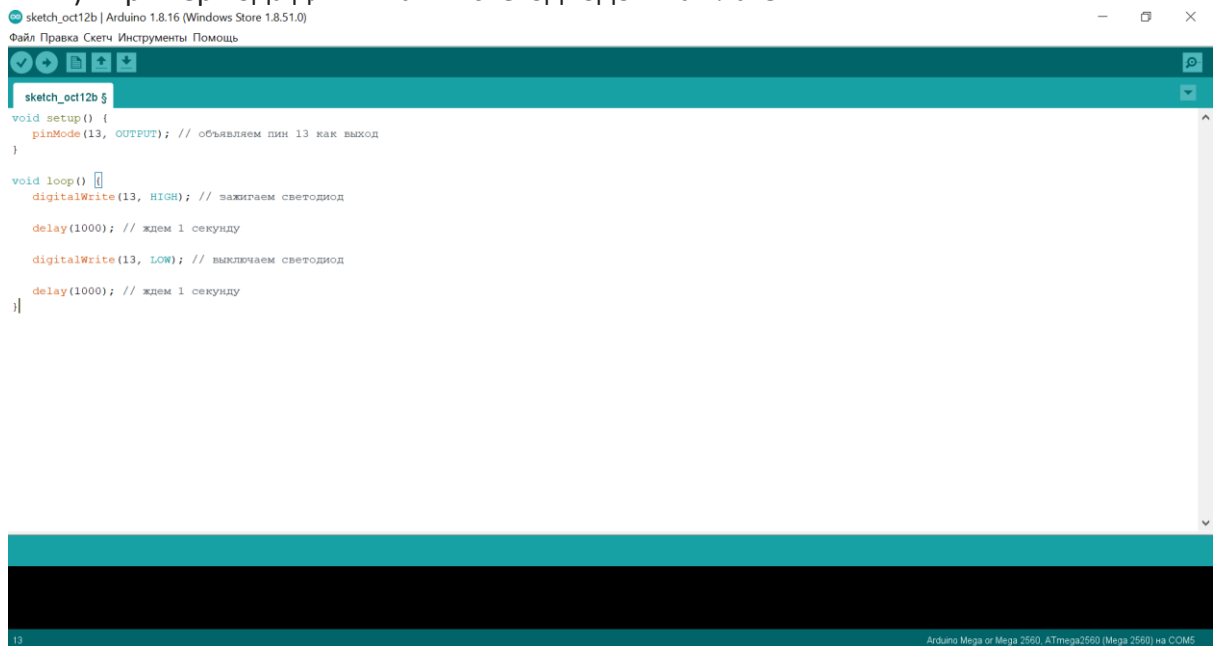


The screenshot shows the Arduino IDE interface. The title bar reads "sketch_oct12a | Arduino 1.8.16 (Windows Store 1.8.51.0)". The menu bar includes "Файл", "Правка", "Скетч", "Инструменты", and "Помощь". The toolbar contains icons for file operations and search. The main editor area is titled "sketch_oct12a" and contains the following code:

```
void setup() {  
  // put your setup code here, to run once:  
  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  
}
```

The status bar at the bottom indicates "Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) на COM5".

4) Пример кода для мигания светодиодом на плате:



The screenshot shows the Arduino IDE interface with a code example for LED blinking. The title bar reads "sketch_oct12b | Arduino 1.8.16 (Windows Store 1.8.51.0)". The menu bar includes "Файл", "Правка", "Скетч", "Инструменты", and "Помощь". The toolbar contains icons for file operations and search. The main editor area is titled "sketch_oct12b" and contains the following code:

```
void setup() {  
  pinMode(13, OUTPUT); // объявляем пин 13 как выход  
}  
  
void loop() {  
  digitalWrite(13, HIGH); // зажигаем светодиод  
  delay(1000); // ждем 1 секунду  
  digitalWrite(13, LOW); // выключаем светодиод  
  delay(1000); // ждем 1 секунду  
}
```

The status bar at the bottom indicates "Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) на COM5".

Программирование с использованием средств разработки языка JavaScript

Язык программирования JavaScript является одним из наиболее популярных решений для разработки web-приложений, а также решений в области технологий «Интернет вещей».

Для разработки решений в области технологий «Интернет вещей» применяется множество различных решений, выпускаемых производителями специализированного оборудования. Однако подобные решения, как правило, очень дорогие и сложные для применения в рамках образовательного процесса.

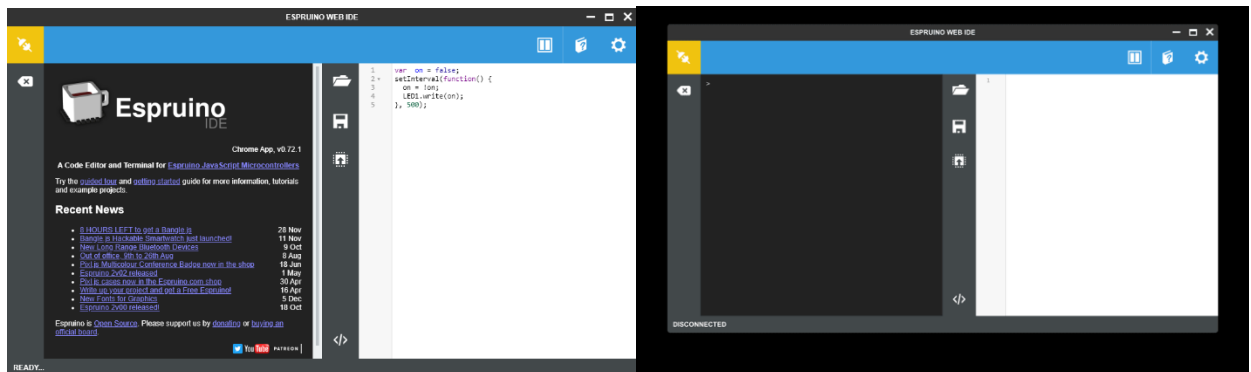
Одним из наиболее распространенных решений для разработки простых и дешевых устройств, поддерживающих технологии «Интернета вещей» и программируемые с помощью языка программирования JavaScript, являются устройства на базе микроконтроллеров серии ESP32.

В состав набора Конструктор программируемых моделей инженерных систем входит подобный контроллер – Универсальный вычислительный модуль DXL-IoT, представляющий собой контроллер на базе микроконтроллера ATmega 2560 и ESP32.

Для программирования контроллера DXL-IoT на базе микроконтроллера ESP32 применяется среда разработки Espruino на базе языка программирования JavaScript.

Установка среды разработки Espruino:

- 1) Скачайте дистрибутив для установки среды Espruino с официального сайта <https://www.espruino.com/ESP32>
Прошейте контроллер в соответствии с инструкцией на сайте.
- 2) Установите среду разработки Espruino Web IDE



- 3) Осуществите настройки во вкладке COMMUNICATIONS и установите скорость 115200
- 4) Подключение к контроллеру может осуществляться беспроводным образом и посредством интерфейса USB

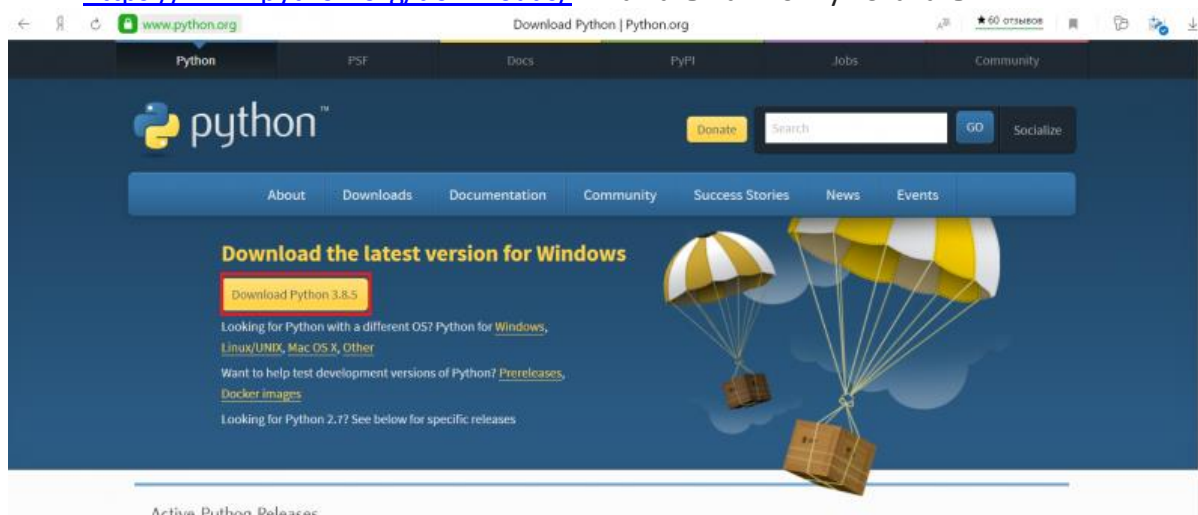
Программирование с использованием средств разработки языка Python

Язык программирования Python является одним из наиболее распространенных многофункциональных языков программирования, применяемых в различных задачах, связанных с математическим моделированием, работой с оборудованием, написание веб-приложений и т.п.

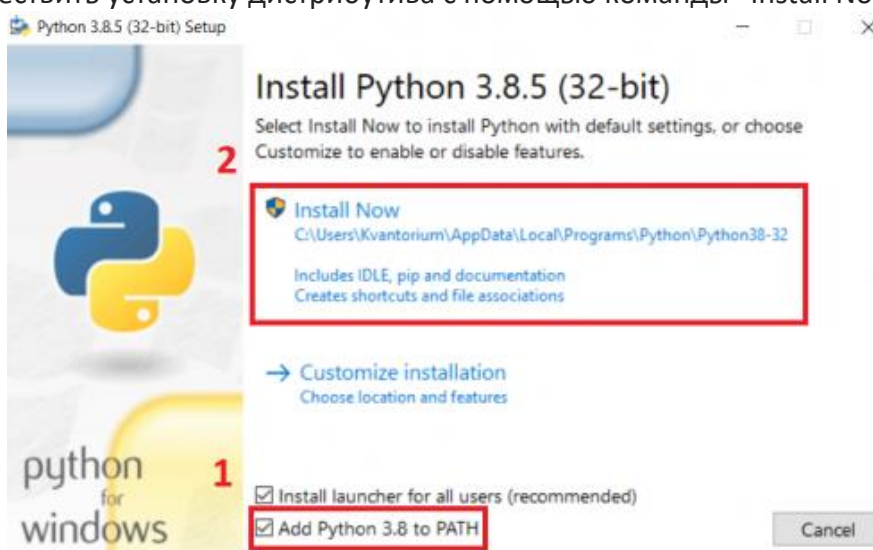
В состав набора Конструктор программируемых моделей инженерных систем входит – Контроллер программируемых моделей инженерных систем и Модуль Технического зрения TrackingCAM, обеспечивающие возможность разработки программного обеспечения с помощью языка программирования Python.

Использование Python для работы с контроллером программируемых моделей инженерных систем:

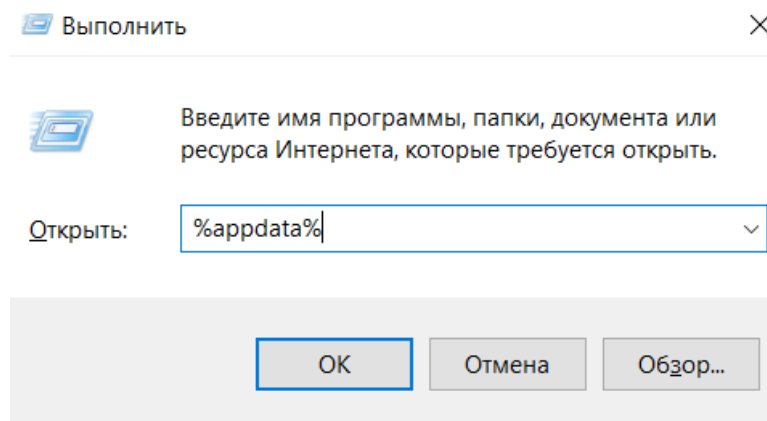
- 1) Установить Python на компьютер. Для этого нужно перейти на официальный сайт <https://www.python.org/downloads/> и нажать на кнопку "Скачать":



Осуществить установку дистрибутива с помощью команды "Install Now":



- 2) Нажать комбинацию клавиш Win+R, после чего откроется командная строка, в ней необходимо написать команду %appdata%, после чего нажать "OK",



После выполнения команды необходимо открыть саму папку AppData и открыть путь к папке Scripts, т.е. путь будет выглядеть следующим образом:

C:\...\AppData\Local\Programs\Python\Python38-32\Scripts

Необходимо скопировать получившийся путь к папке "Scripts".

- 3) Вызвать командную строку:

Win+R -> написать команду cmd и нажать "OK"

После открывается адресная строка, в которой надо написать следующее:

cd C:\...\AppData\Local\Programs\Python\Python38-32\Scripts

(т.е. путь к папке "Scripts") -> нажать "Enter"

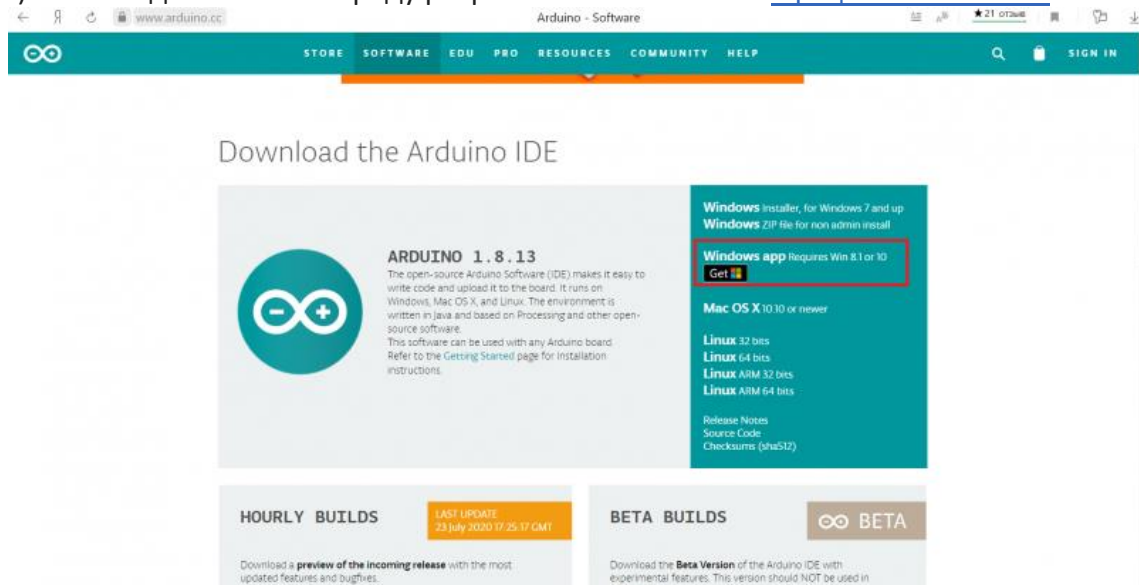
```
C:\Users\... >cd C:\...\AppData\Local\Programs\Python\Python38-32\Scripts
```

После необходимо ввести команду:

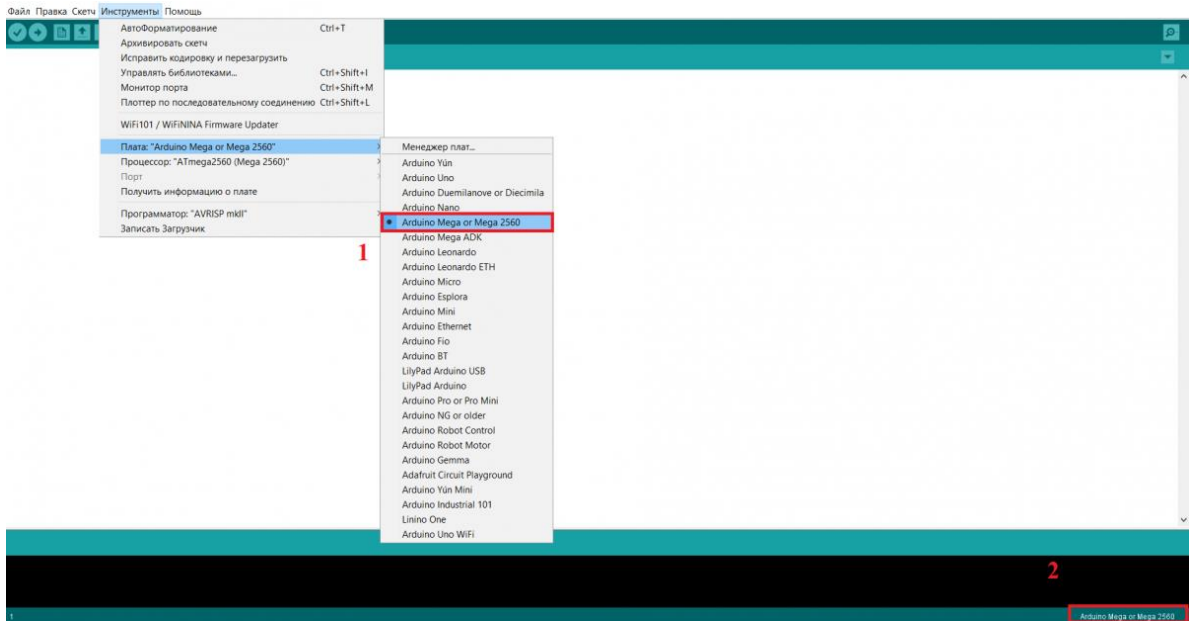
pip install pyserial -> нажать "Enter"

Таким образом, средства языка программирования Python на компьютере установлены, можно приступать к дальнейшей работе.

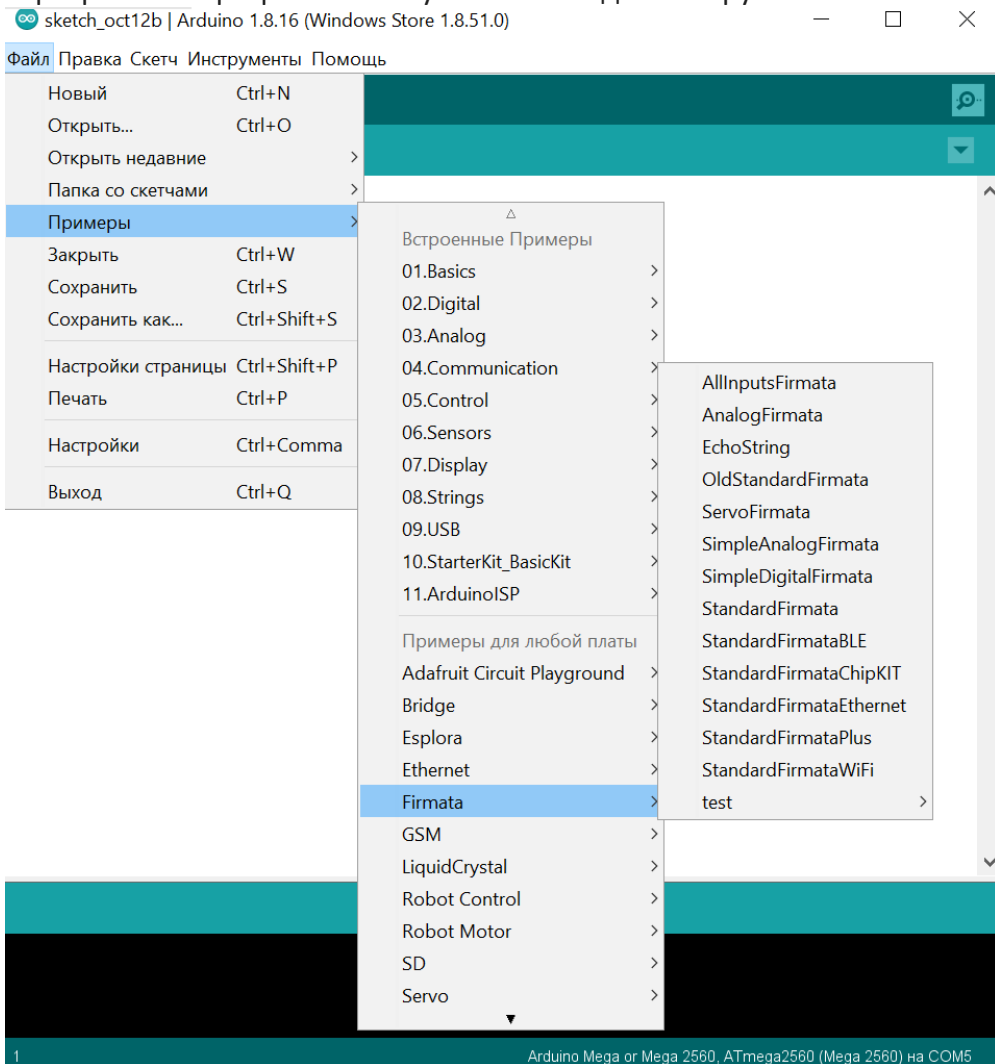
- 4) Необходимо скачать среду разработки Arduino IDE с [официального сайта](https://www.arduino.cc)



5) Выбрать контроллер Arduino Mega или Mega 2560 как показано ниже



6) Для разработки программы на Python необходимо загрузить скетч Firmata:



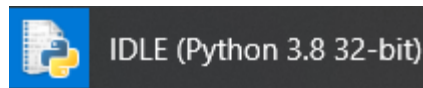
- 7) После выполнения действий откроется новое окно, в нем необходимо будет повторить шаг 5

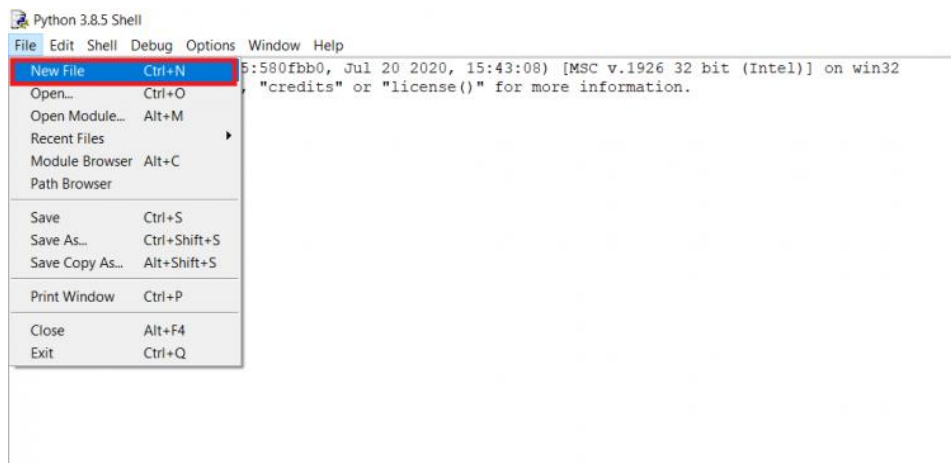


- 8) После завершения загрузки вы не заметите никакой активности на Arduino. Для дальнейшей работы потребуется программа, которая может взаимодействовать с платой через последовательное соединение. Для работы с протоколом Firmata в Python необходим пакет pyFirmata, который необходимо установить с помощью команды:

```
$ pip install pyfirmata
```

- 9) После установки приложения, заходим в ранее установленную программу IDLE(Python ...) или ее аналог.





10) Пример разработки программы на языке Python:

```
Python
1 import pyfirmata
2 import time
3
4 board = pyfirmata.Arduino('/dev/ttyACM0')
5
6 while True:
7     board.digital[13].write(1)
8     time.sleep(1)
9     board.digital[13].write(0)
10    time.sleep(1)
```

В строках 7-8 включается светодиод на 1 секунду

В строках 9-10 светодиод выключается на 1 секунду

Использование Python для работы с модулей технического зрения TrackingCAM:

Модуль технического зрения TrackingCam v3 - это миниатюрный компьютер, который также можно использовать как контроллер для устройств интернета вещей. Для этого в первую очередь необходимо подключиться напрямую к системе камеры с помощью протокола ssh.

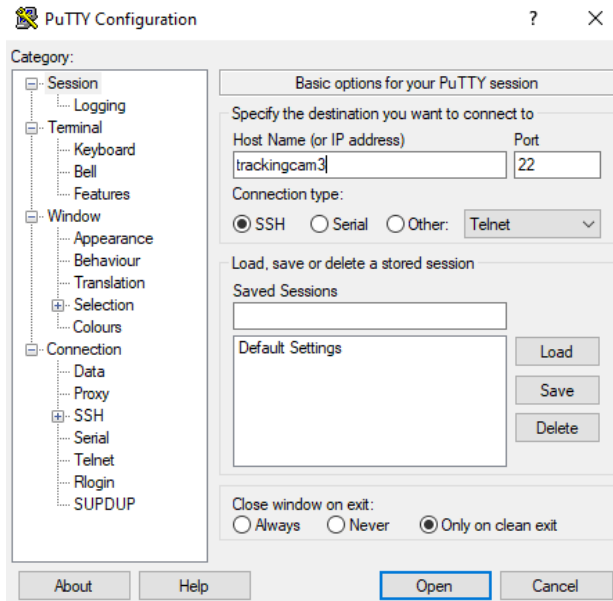
При использовании ОС Windows это можно сделать установив программу Putty, реализующую удаленный терминал доступа к компьютеру.

Скачать и установить Putty можно по ссылке:

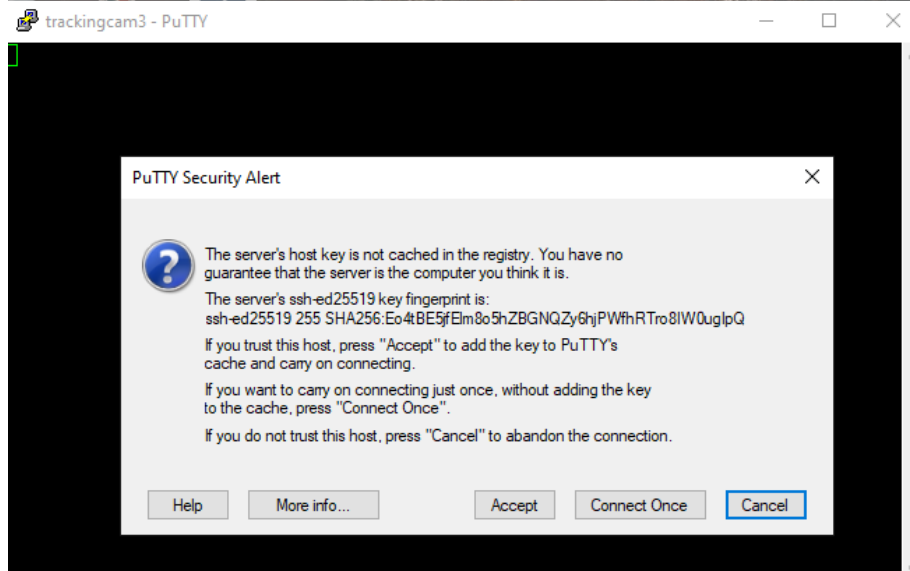
<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

После установки необходимо запустить программу и подключить камеру к компьютеру при помощи USB кабеля.

В окне Host Name (or IP address) необходимо указать: trackingcam3



После этого нажать кнопку open. В выпавшем окне нажать кнопку accept.



Теперь необходимо войти в систему, имя пользователя root, пароль 12345.

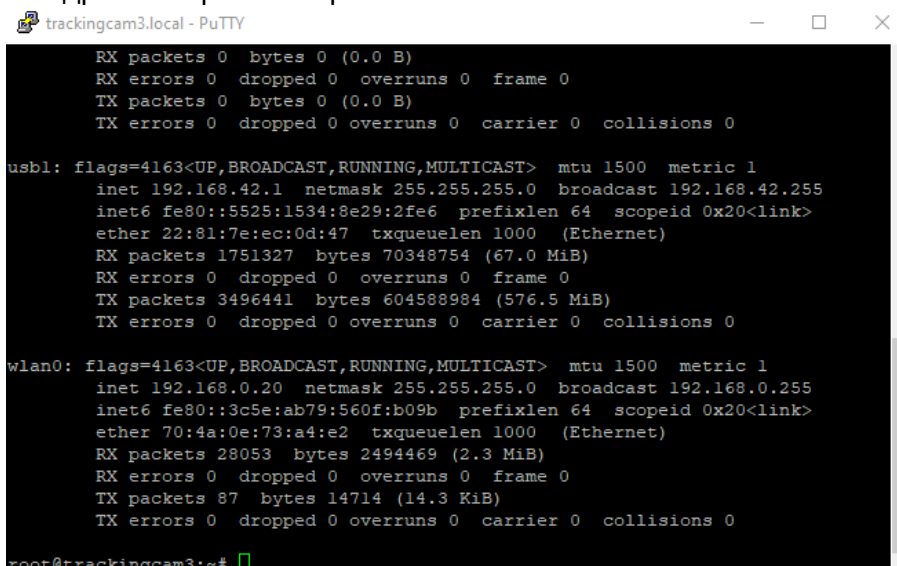


Открывается терминал, через который можно взаимодействовать с камерой.

Для того, чтобы реализовать алгоритм общения на TrackingCam необходимо написать скрипт на языке python3. Рассмотрим пример скрипта, реализующего соединение по протоколу UDP:

```
import socket
UDP_IP_ADDRESS = "127.0.0.1"
UDP_PORT_NO = 6789
UDP_CLIENT_ADRESS = "127.0.1.1"
serverSock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
serverSock.bind((UDP_IP_ADDRESS, UDP_PORT_NO))
udp_broadcast=(UDP_CLIENT_ADRESS, self.UDP_PORT_OUT)
while True:
    data, addr = serverSock.recvfrom(1024)
    print ("Message: ", data)
    serverSock.sendto(data.encode(), udp_broadcast)
```

В данном случае скрипт реализует общение с устройством по адресу UDP_CLIENT_ADRESS, а UDP_IP_ADRESS - это адрес камеры в конкретной сети. Для того, чтобы узнать IP адрес камеры в конкретной сети можно воспользоваться командой *ifconfig*:



```
trackingcam3.local - PuTTY
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

usb1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500 metric 1
inet 192.168.42.1 netmask 255.255.255.0 broadcast 192.168.42.255
inet6 fe80::5525:1534:8e29:2fe6 prefixlen 64 scopeid 0x20<link>
ether 22:81:7e:ec:0d:47 txqueuelen 1000 (Ethernet)
RX packets 1751327 bytes 70348754 (67.0 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 3496441 bytes 604588984 (576.5 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500 metric 1
inet 192.168.0.20 netmask 255.255.255.0 broadcast 192.168.0.255
inet6 fe80::3c5e:ab79:560f:b09b prefixlen 64 scopeid 0x20<link>
ether 70:4a:0e:73:a4:e2 txqueuelen 1000 (Ethernet)
RX packets 28053 bytes 2494469 (2.3 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 87 bytes 14714 (14.3 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@trackingcam3:~#
```

Здесь usb1 - это соединение компьютера и камеры по USB (оно имитирует Ethernet, здесь адрес камеры 192.168.42.1); wlan0 - соединение камеры с общей сетью по wi-fi и в нем адрес 192.168.0.20.

В цикле программа принимает от устройства некоторое сообщение, затем выводит его на экран и отправляет это же сообщение обратно устройству.

Для того чтобы написать программу можно воспользоваться текстовым редактором nano.

Для создания файла используется команда nano "имя файла", например:
nano udp.py

Далее необходимо набрать программу и выйти из редактора сохранив ее, это можно сделать при помощи сочетания клавиш **ctrl + o** и затем **ctrl + x**.

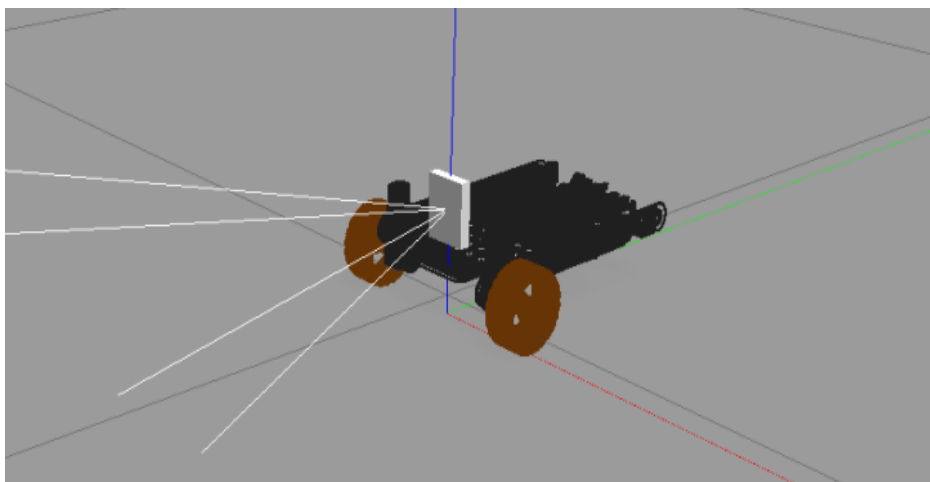
Для запуска программы используется команда python3 "имя файла" в данном случае:

python3 udp.py

Программирование с использованием средств разработки ROS

В состав набора Конструктор программируемых моделей инженерных систем входит программное обеспечение для моделирования алгоритмов управления мобильного робота, оснащенного модулем технического зрения TrackingCAM.

Модель мобильного робота с камерой tracking_cam 3 может быть симулирована в среде Gazebo с использованием фреймворка ROS.



Для использования данной системы необходимо скачать образ системы с предустановленным ПО по ссылке:

https://drive.google.com/file/d/189bw4nz9f0FjmXNQ_pzb8vSciKIYu_HV/view?usp=sharing

После чего установить образ на ПК при помощи программы Oracle VM VirtualBox, что позволит запустить виртуальную машину с образом Ubuntu.

После входа в систему необходимо открыть терминал (сочетание клавиш `ctrl+alt+t`) и выполнить команду: **`roslaunch kpms_gazebo kpms.launch`**.

```
sim@ubuntu: ~/kpms
[master] killing on exit
shutting down processing monitor...
... shutting down processing monitor complete
done
sim@ubuntu:~/kpms$ roslaunch kpms_gazebo turtlebot3_world.launch
... logging to /home/sim/.ros/log/c4b7cdb0-2cf9-11ec-902a-a7b2a234bce2/roslaunch-ubuntu-14301.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

macro: in-order processing became default in ROS Melodic. You can drop the option.
started roslaunch server http://ubuntu:33743/

SUMMARY
=====
PARAMETERS
* /gazebo/enable_ros_network: True
* /robot_description: <?xml version="1...
* /roscdistro: noetic
* /rosversion: 1.15.8
* /use_sim_time: True
```

После чего будет запущен симулятор Gazebo с моделью мобильного робота, оснащенной камерой.

В симуляторе запускается точная копия робота TurtleBot3, включая все топики реального робота. Чаще всего использоваться будут следующие топики:

- odom: топик в который публикуется колесная одометрия робота, то есть его перемещение от точки старта рассчитанное, исходя из показаний датчиков оборотов, установленных в приводах робота. Так как в симулятора Gazebo также симулируется трение, показания в данном топике со временем будут накапливать ошибку, и отличаться от реального перемещения робота. В данный топик публикуются сообщения типа: nav_msgs/Odometry.
- cmd_vel: топик который управляет скоростью движения робота, подробнее его работа рассмотрена в следующем пункте.

Управление моделью робота осуществляется при помощи задания необходимой линейной и угловой скорости движения. В топик cmd_vel, публикуется сообщение типа geometry_msgs/Twist, которое состоит из двух векторов:

linear: - линейная скорость (м/с)

x - по оси X

y - по оси Y

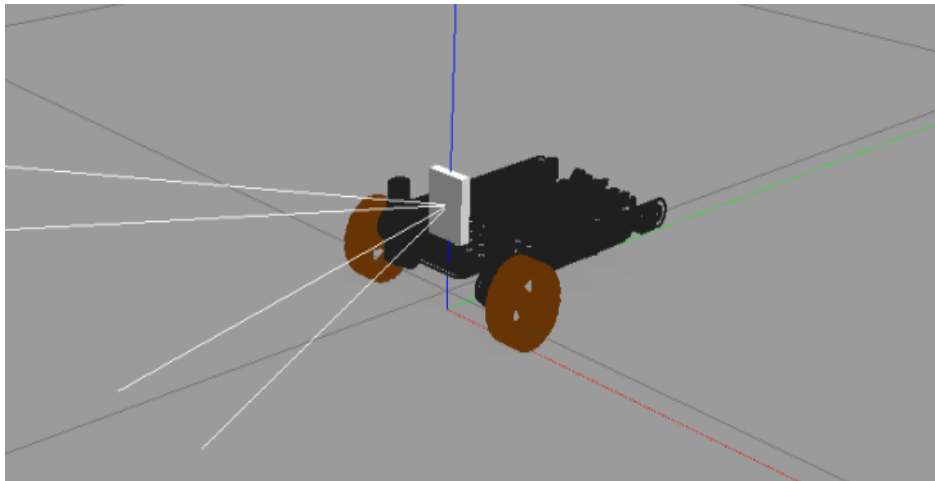
z - по оси Z

angular: - угловая скорость (скорость вращения вокруг своей оси, рад/с)

x - вокруг оси x

y - вокруг оси Y

z - вокруг оси Z



Оси координат модели робота, расположены следующим образом:
Здесь синяя ось - ось Z, красная - ось X, зеленая - ось Y.

Поскольку робот имеет дифференциальную колесную базу, он может перемещаться только вдоль оси X и вокруг оси Z.

Пример команды для прямолинейного перемещения:

msg:

linear:

x = 0.1

y = 0

z = 0

angular:

x = 0

y = 0

z = 0

Пример команды для разворота на месте:

msg:

linear:

x = 0

y = 0

z = 0

angular:

x = 0

y = 0

z = 0.1

Пример команды для движения по дуге:

msg:

linear:

x = 0.1

y = 0

z = 0

angular:

x = 0

y = 0

z = 0.1

Программирование робота можно реализовать в среде разработки Arduino IDE. Чтобы запустить Arduino IDE необходимо в новом терминале выполнить команду: **arduino**