

IoTik Studio

ИНСТРУКЦИЯ

Виктор Фадеев | ООО "MgBot" | 06.04.2020

Расположение ресурсов

Все ресурсы необходимые для начала работы с IoTik Studio, на данный момент, располагаются в [Google Drive](#). Также ниже рассоложен QR код, дублирующий ссылку:



В данной папке содержатся:

- Драйвера для работы с микропроцессором ESP32 (установлен в ЙОТик v2.0)
- Последняя версия прошивки для ЙОТик v2.0 (файл [Firmware.zip](#))
- Установочный файл IoTik Studio (файл [IoTik Studio Installer.exe](#))
- А также примеры диаграмм различных проектов (папка [Примеры](#))

Прямая ссылка на скачивание инсталлятора является обновляемой и по ней, на данный момент, выкладываются обновления. Доступна по коду:



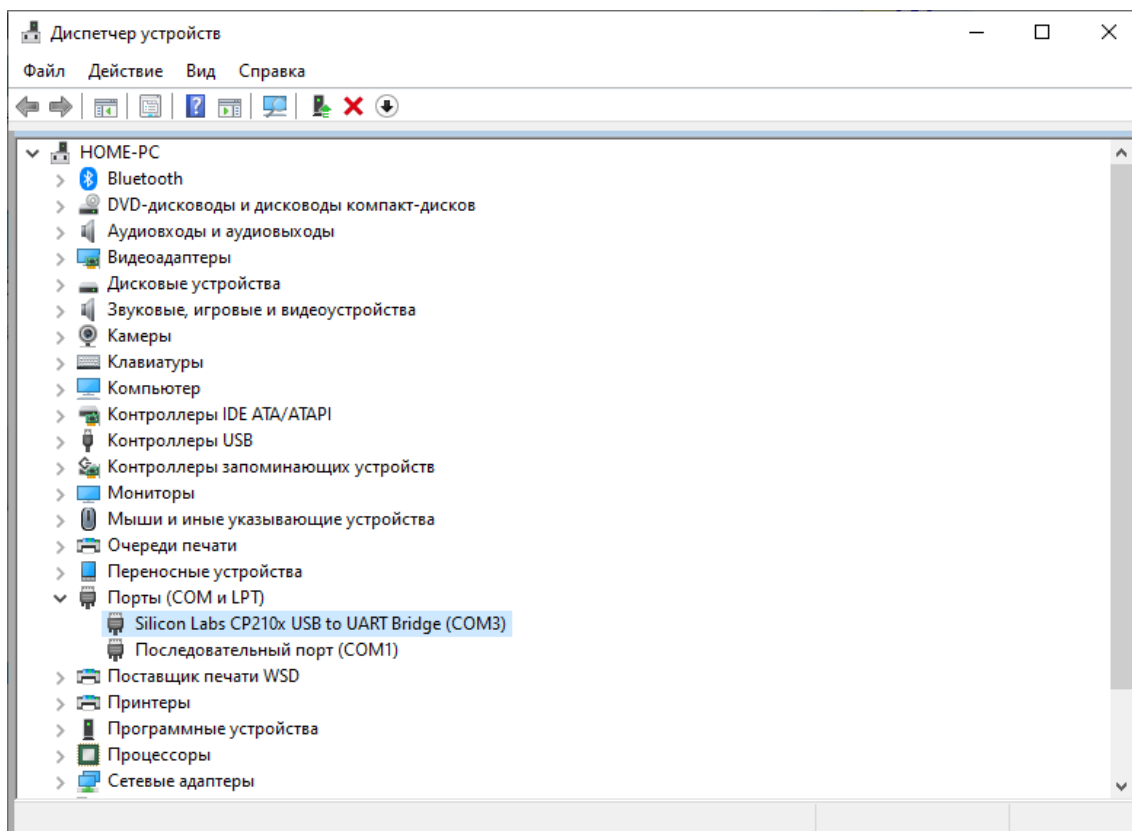
Также для загрузки прошивки понадобится Python, установить его можно с официального сайта www.python.org, или по прямой ссылке [Python 3.8.0](#).

Настройка компьютера

Для первоначальной загрузки прошивки необходимо установить Python. При установке, обязательно нужно поставить галочку “Add Python to PATH”. Как на рисунке ниже:



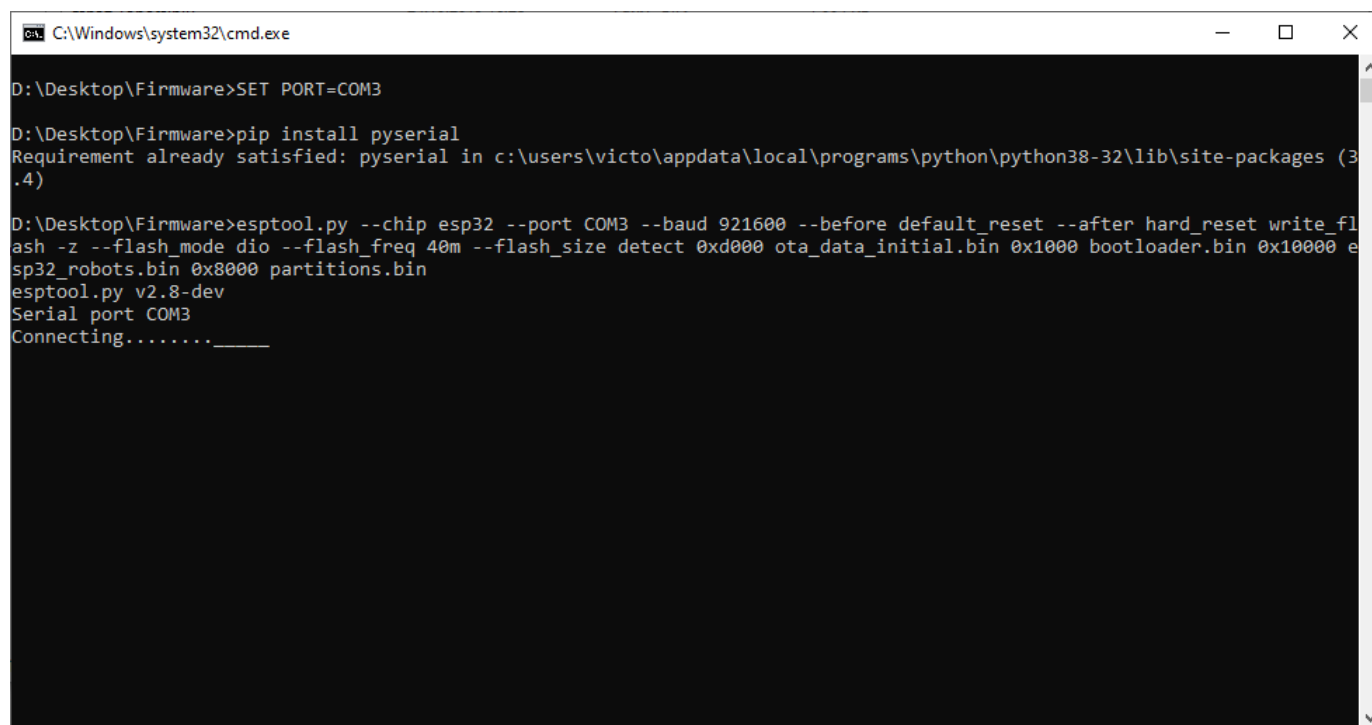
Кроме того, при первом использовании ЙоТик-а на новом компьютере необходимо установить драйвер. Использование драйвера по умолчанию может приводить к некорректной работе. В папке “Драйвера” находятся 2 варианта драйвера, в большинстве случаев достаточно установить только [CP210x Universal Driver.zip](#). В случае неполадок следует использовать [Pololu - CP2102.zip](#). После установки в диспетчере устройств должна появиться строка “Silicon Labs CP210x USB to UART Bridge (COM3)”.



Загрузка прошивки

После установки можно загрузить прошивку на плату. Данную процедуру достаточно провести один раз для поддержки IoTik Studio контроллером. В случае использования IoTik Studio совместно с Arduino IDE, необходимо прошивать контроллер перед каждой сессией в IoTik Studio. На прошлом шаге в диспетчере задач был указан COM порт, по умолчанию COM3. Если у вас другой порт его следует запомнить.

Прошивка находится в архиве [Firmware.zip](#), загрузите его и распакуйте. Если ваш COM порт не совпадает с COM3, тогда откройте файл flash.bat правой кнопкой для изменения и перепишите COM3 на ваше значение. После сохранения подключите ЙоТик и запустите flash.bat двойным щелчком. Начнется загрузка прошивки, для подтверждения нажмите и подержите 1~2 сек. клавишу flash на плате (ближайшая к USB порту). Запрос подтверждения:



```
C:\Windows\system32\cmd.exe

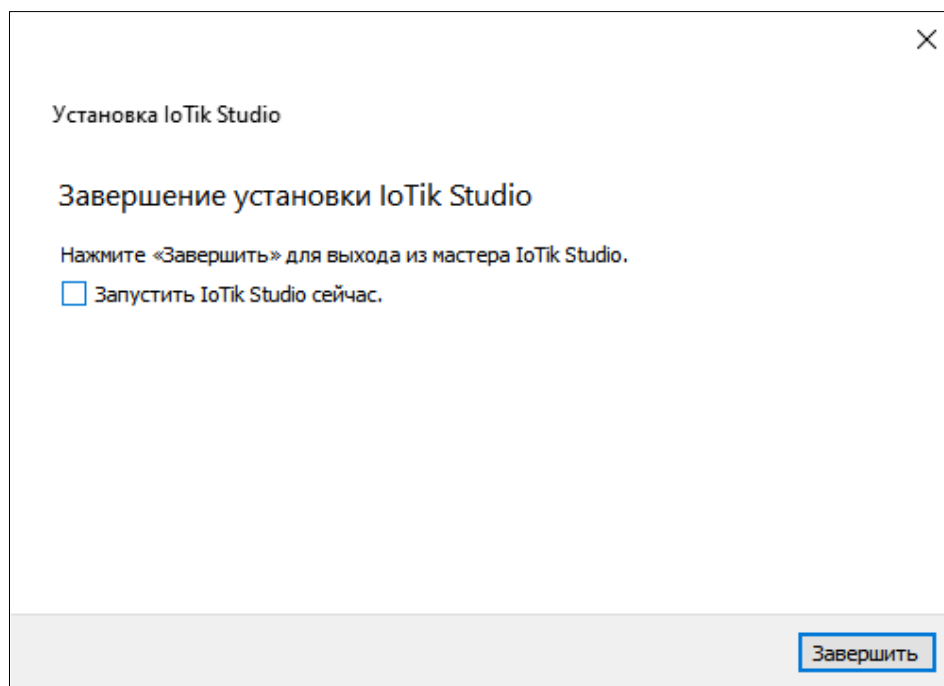
D:\Desktop\Firmware>SET PORT=COM3

D:\Desktop\Firmware>pip install pyserial
Requirement already satisfied: pyserial in c:\users\victo\appdata\local\programs\python\python38-32\lib\site-packages (3.4)

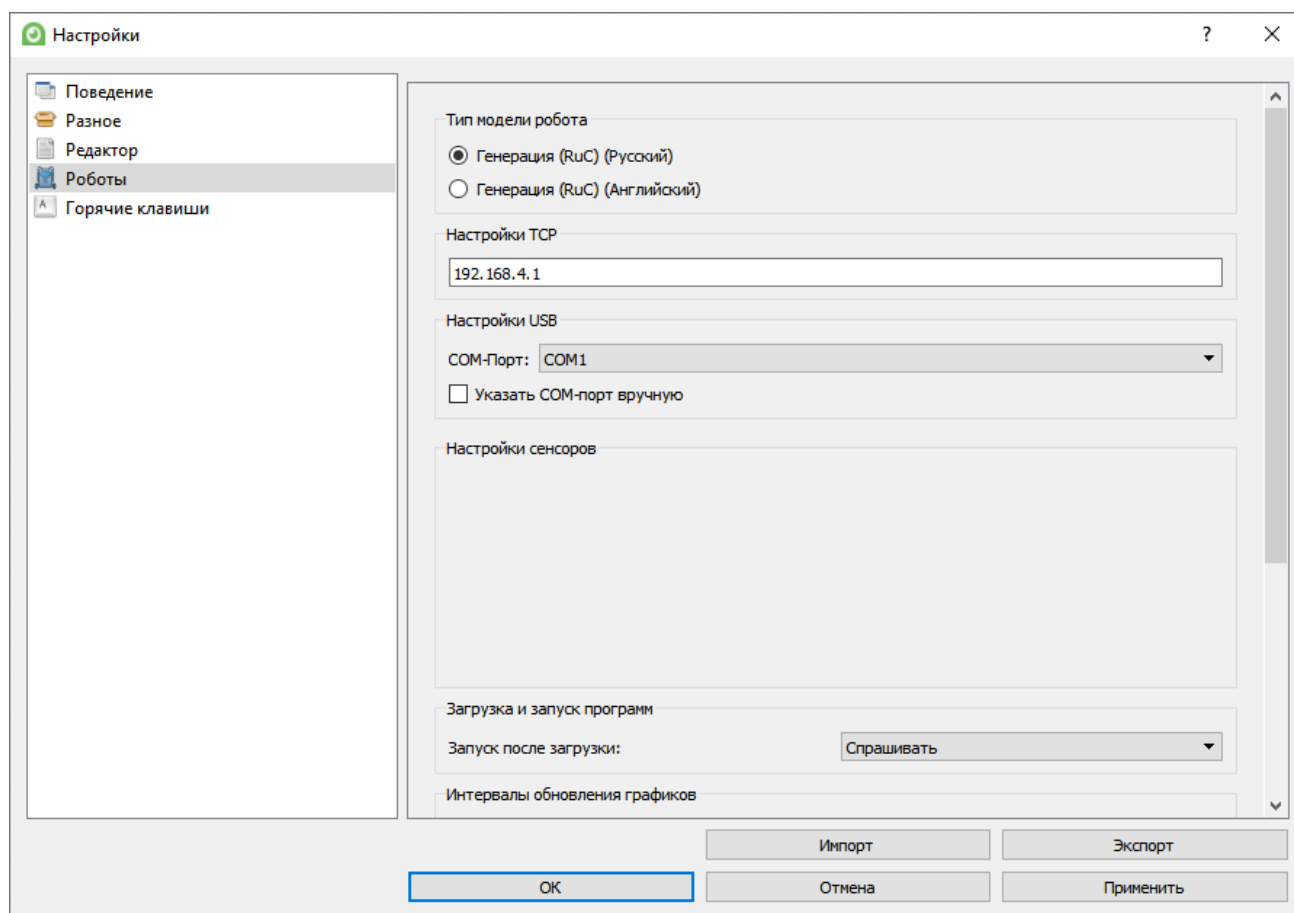
D:\Desktop\Firmware>esptool.py --chip esp32 --port COM3 --baud 921600 --before default_reset --after hard_reset write_flash -z --flash_mode dio --flash_freq 40m --flash_size detect 0xd000 ota_data_initial.bin 0x1000 bootloader.bin 0x10000 esp32_robots.bin 0x8000 partitions.bin
esptool.py v2.8-dev
Serial port COM3
Connecting....._____
```

Установка IoTik Studio

Для установки IoTik Studio следуйте инструкциям установщика. В конце установки предпочтительно снять галочку “Запустить IoTik Studio сейчас”.

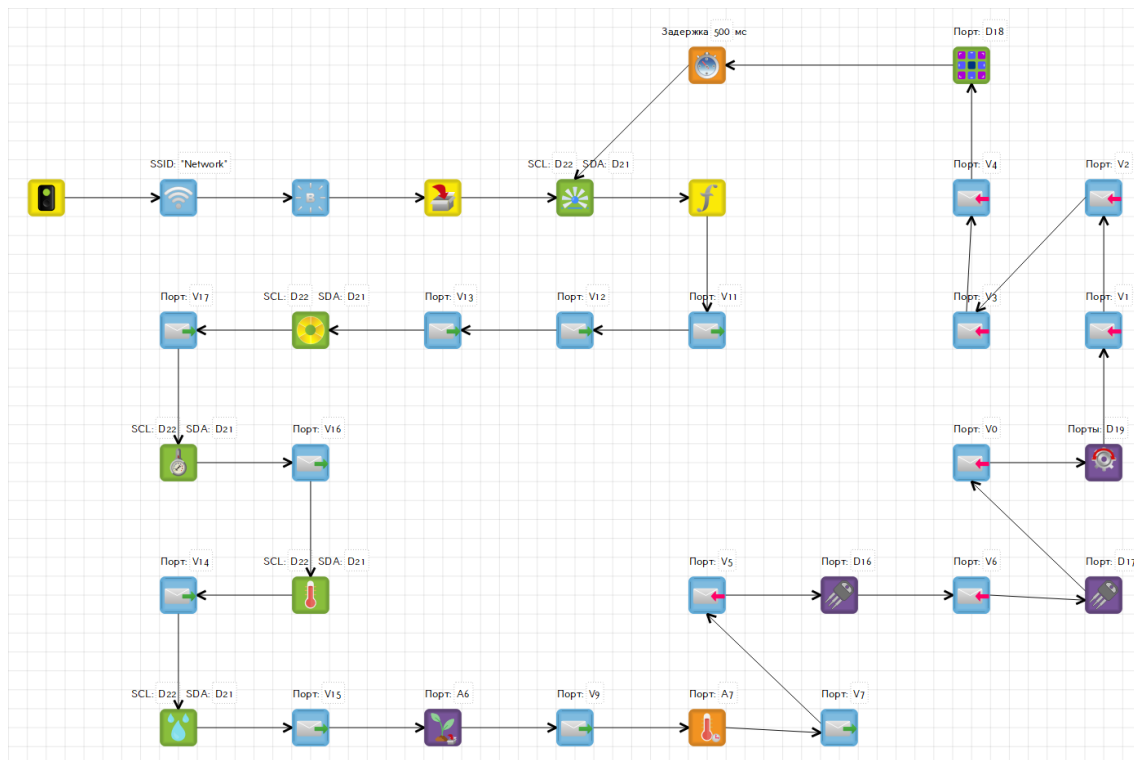


Для программирования под ЙоТик выберите в настройках предпочитаемый язык генерации кода (по умолчанию русский).



Пример “Теплица”

Загрузите из папки примеров файл проекта Greenhouse.qrs, а также вспомогательный файл Greenhouse.c. Открыв проект вы увидите следующую диаграмму:

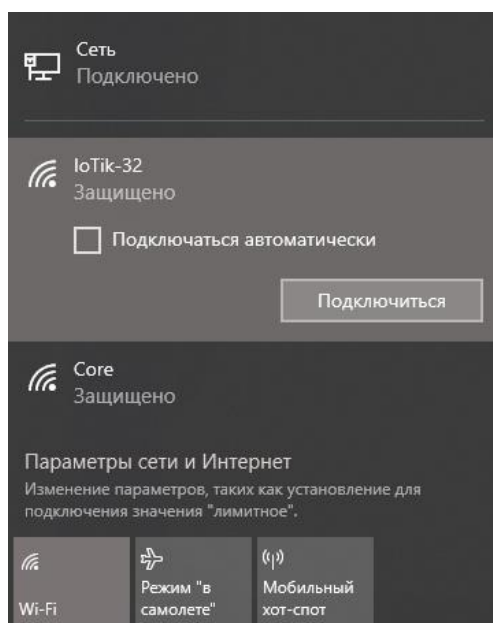


Измените параметры первых блоков: Wi-Fi и Blynk, указав данные своей сети (поддерживаются только сети 2,4 ГГц) и авторизационный токен. После чего нажмите кнопку генерации кода на верхней панели. Если вы хотите отправлять значения аналогового датчика температуры в Blynk и управлять цветом матрицы, внесите следующие изменения (или скопируйте код из Greenhouse.c):

```
47     pressure = getdigsensor(PRESSURE, { D22, D21 });
48     blynk_send(V16, pressure);
49     temperature = getdigsensor(HEATING, { D22, D21 });
50     blynk_send(V14, temperature);
51     humidity = getdigsensor(HUMIDITY, { D22, D21 });
52     blynk_send(V15, humidity);
53     moisture = getdigsensor(SOIL, A6);
54     blynk_send(V9, moisture);
55     while (!(getdigsensor(TEMPERATURE, A7) > 0)) {
56         t_sleep(10);
57     }
58     blynk_send(V7, soil);
59     fan = blynk_receive(V5);
60     setvoltage(D16, fan);
61     pump = blynk_receive(V6);
62     setvoltage(D17, pump);
63     window = blynk_receive(V0);
64     setmotor(D19, window);
65     red = blynk_receive(V1);
66     green = blynk_receive(V2);
67     blue = blynk_receive(V3);
68     bright = blynk_receive(V4);
69     matrix_1[0] = '0';
70     matrix_1[1] = 255 * bright / 100;
71     matrix_1[2] = 255 * bright / 100;
72     matrix_1[3] = 255 * bright / 100;
73     setsignal(MATRIX, { D18 }, matrix_1);
74     t_sleep(500);
75 }
76 return 0;
77 }
78 }
79 }
80 }
```

```
45     blynk_send(V17, light);
46     pressure = getdigsensor(PRESSURE, { D22, D21 });
47     blynk_send(V16, pressure);
48     temperature = getdigsensor(HEATING, { D22, D21 });
49     blynk_send(V14, temperature);
50     humidity = getdigsensor(HUMIDITY, { D22, D21 });
51     blynk_send(V15, humidity);
52     moisture = getdigsensor(SOIL, A6);
53     blynk_send(V9, moisture);
54     soil = getdigsensor(TEMPERATURE, A7);
55     blynk_send(V7, soil);
56     fan = blynk_receive(V5);
57     setvoltage(D16, fan);
58     pump = blynk_receive(V6);
59     setvoltage(D17, pump);
60     window = blynk_receive(V0);
61     setmotor(D19, window);
62     red = blynk_receive(V1);
63     green = blynk_receive(V2);
64     blue = blynk_receive(V3);
65     bright = blynk_receive(V4);
66     matrix_1[0] = '0';
67     matrix_1[1] = red * bright / 100;
68     matrix_1[2] = green * bright / 100;
69     matrix_1[3] = blue * bright / 100;
70     setsignal(MATRIX, { D18 }, matrix_1);
71     t_sleep(500);
72 }
73 return 0;
74 }
```

После внесения всех изменений подключитесь к Wi-Fi “IoTik-32”, с паролем “0123456789”.



Нажмите кнопку загрузки прошивки (со значком Wi-Fi). Для сброса программы перезагрузите контроллер. Для загрузки новой программы заново откройте меню выбора Wi-Fi сети.

P.s. В случае ошибки при работе или неправильной загрузки, просто повторите ее перезапустив контроллер.

НАСТРОЙКА BLYNK

Для управления проектом “Теплица” используется мобильное приложение Blynk. В нем необходимо сконфигурировать следующий интерфейс, стоимостью в 4300 ед. энергии (мин. 3000):



Данный интерфейс использует: 6 x “Labeled Value”, 3 x “Value Display”, 1 x “Styled Button”, 1 x “Button”, 1 x “Slider”, 1 x “Vertical Slider”, 1 x “zeRGBa”. В целях экономии можно заменить все “Labeled Value” на “Value Display”, а “Styled Button” на “Button”, снизив стоимость до 3000 ед. энергии.

При создании проекта укажите в качестве контроллера “ESP32 Dev Board”, а способа связи “Wi-Fi”. Затем добавьте все необходимые блоки на панель и разместите их аналогично схеме. Все элементы используют шрифт максимального размера для улучшения видимости, кроме датчика BME280 (атмосферное давление, температура и влажность). Также все показания принимаются в режиме “PUSH”.

Далее представлен список сопоставления виртуальных портов реальным датчикам:

- Датчик света, порт V17, формат вывода “/pin/ lx”
- Датчик атмосферного давления, порт V16, формат вывода “/pin/ mm”
- Датчик температуры воздуха, порт V14, формат вывода “/pin/ C°”
- Датчик влажности воздуха, порт V15, формат вывода “/pin/ % RH”
- Емкостный датчик влажности почвы, порт V9, формат вывода “Humidity: /pin/ %”
- Емкостный датчик температуры, порт V7, формат вывода “Temperature: /pin/ C°”
- Ультрафиолетовый датчик, параметр UV-A, порт V11
- Ультрафиолетовый датчик, параметр UV-B, порт V12
- Ультрафиолетовый датчик, параметр UV-I, порт V13

Ниже расположен аналогичный список для элементов управления:

- Питание вентилятора, порт V5, значения 0/255, режим “SWITCH”, стиль “OUTLINE”
- Питание насоса, порт V6, значения 0/255, режим “SWITCH”
- Механизм управления окном, порт V0, угол от -90° до 90°
- Яркость матрицы, порт V4, от 0% до 100%
- Цвет матрицы, режим “SPLIT”, порты V1, V2, V3