

Сбор, передача и визуализация данных при помощи устройств IoT

Содержание



Введение в платформу ESP32 Pico kit и датчик BMP280



Описание задачи



Установка ESP32 Arduino



Настройка сервиса InitialState



Визуализация сенсорных данных

ESP32 Pico kit



ESP32 —SoC с поддержкой Wi-Fi и Bluetooth, разработчик Espressif Systems.



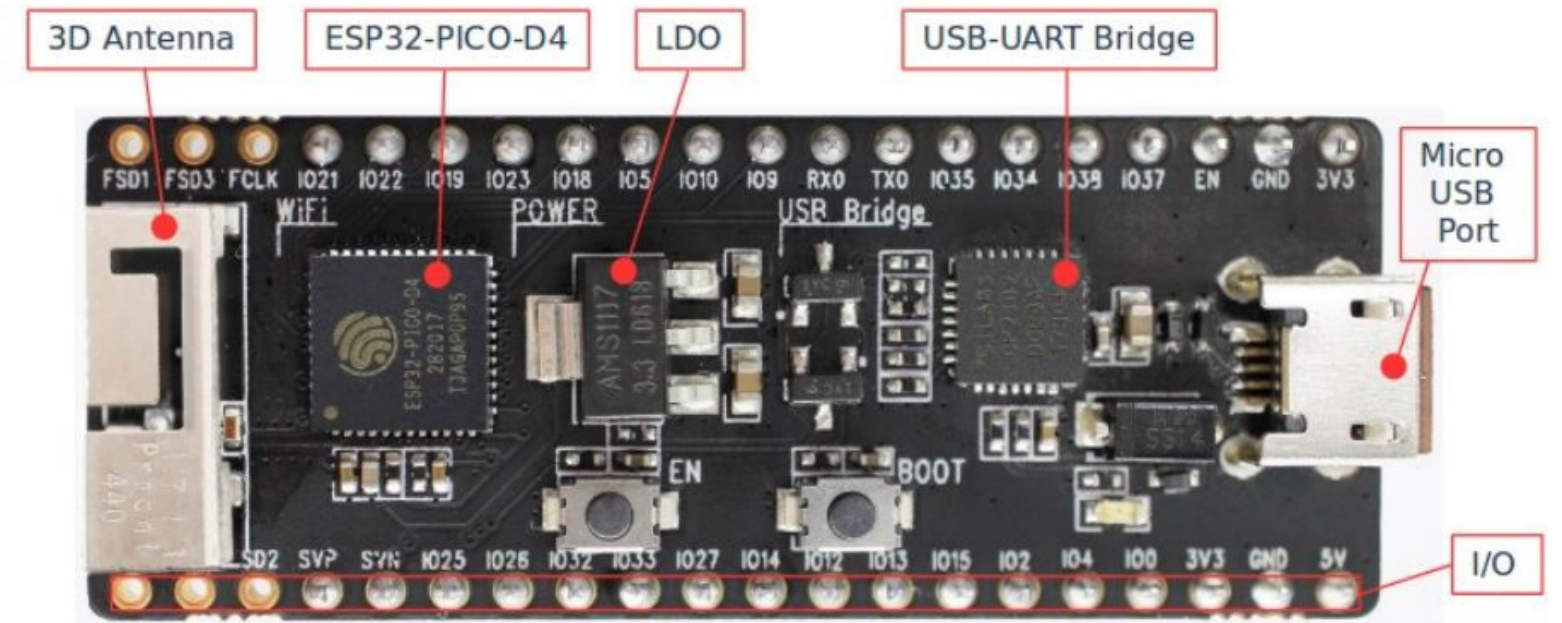
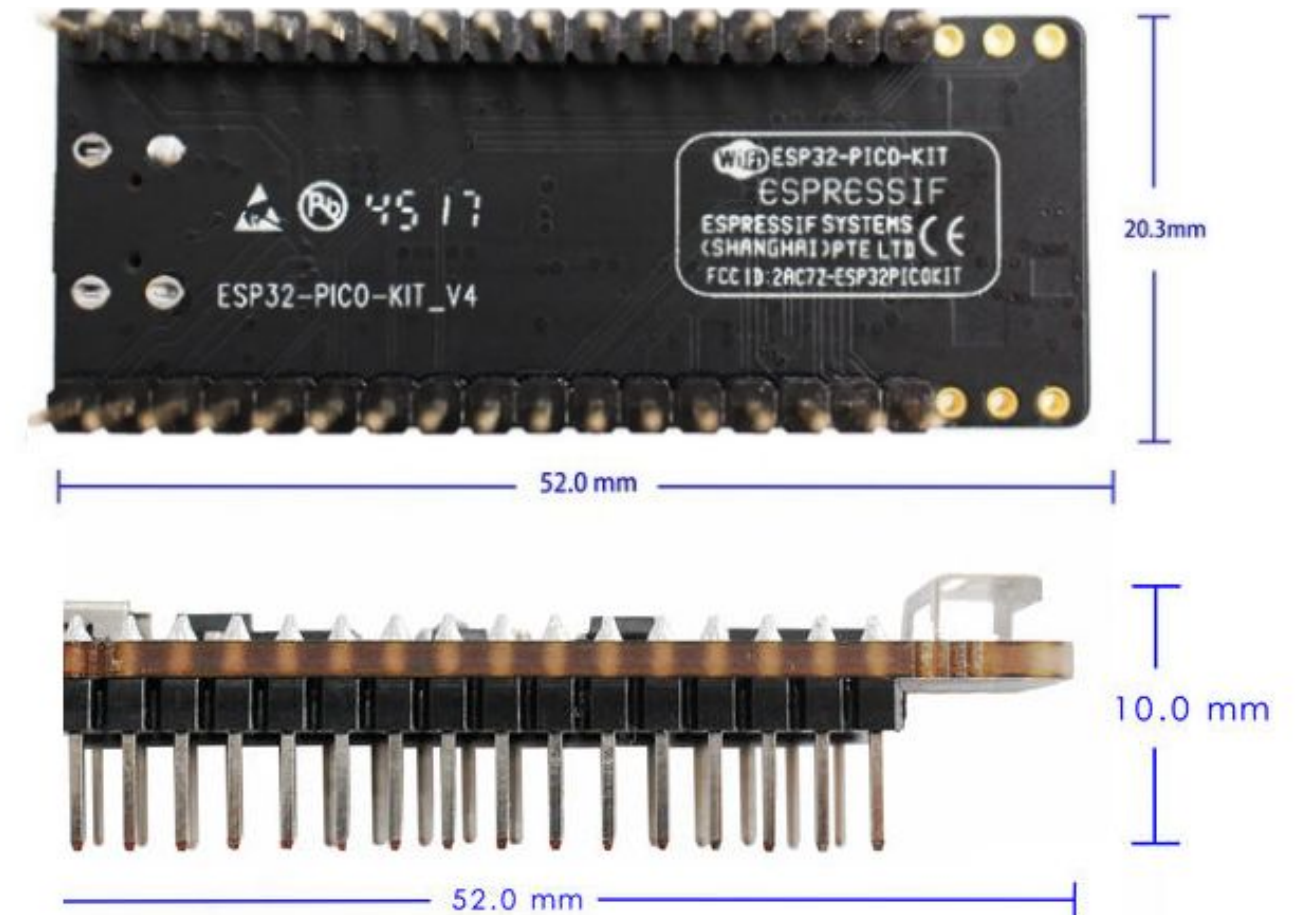
Напряжение питания 5 В или 3.3 В



WiFi и BT/BLE

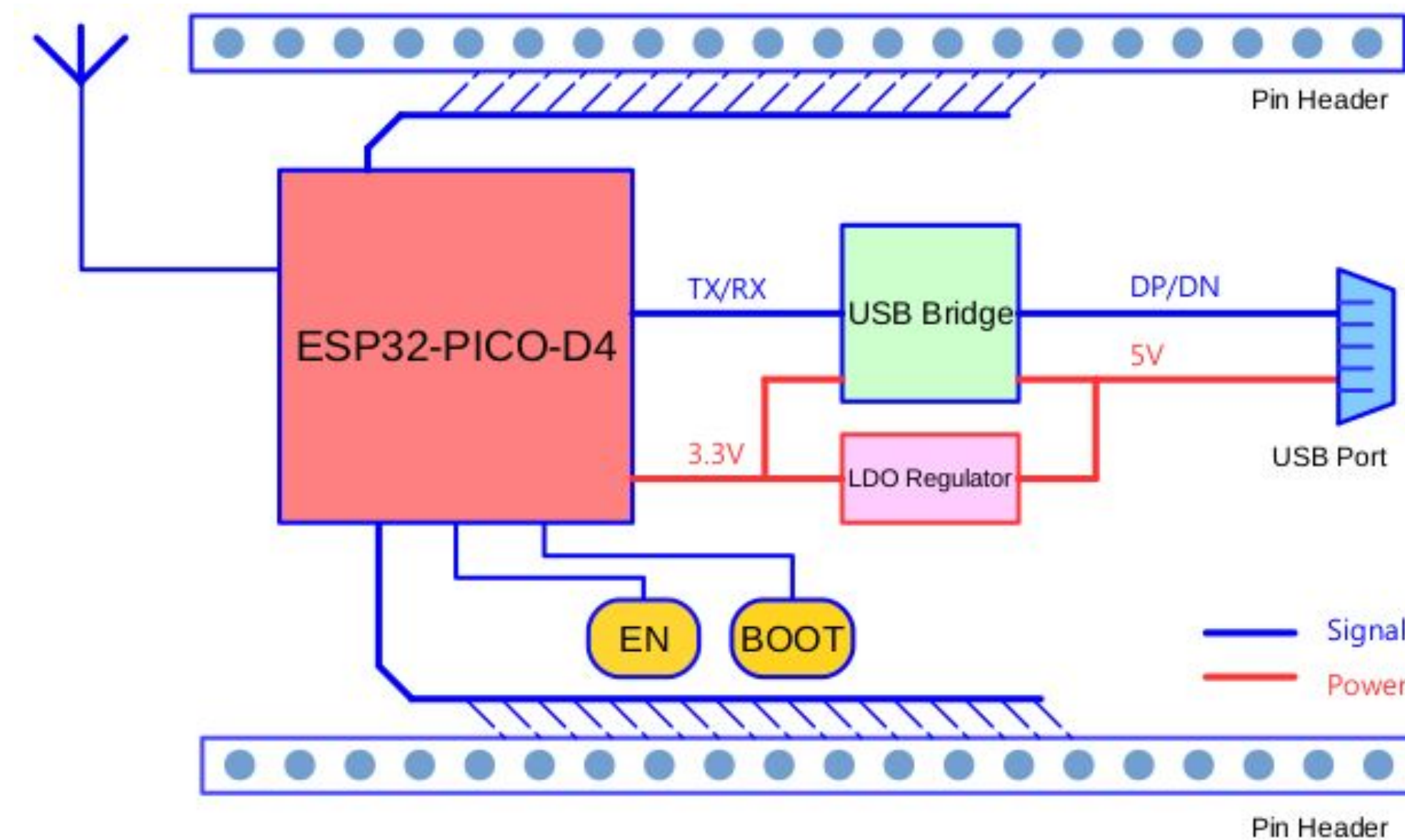


GPIO, ADC, DAC, SPI, SD, I2C, I2S, PWM, UART



Изображение: demo-djiudu.readthedocs.io

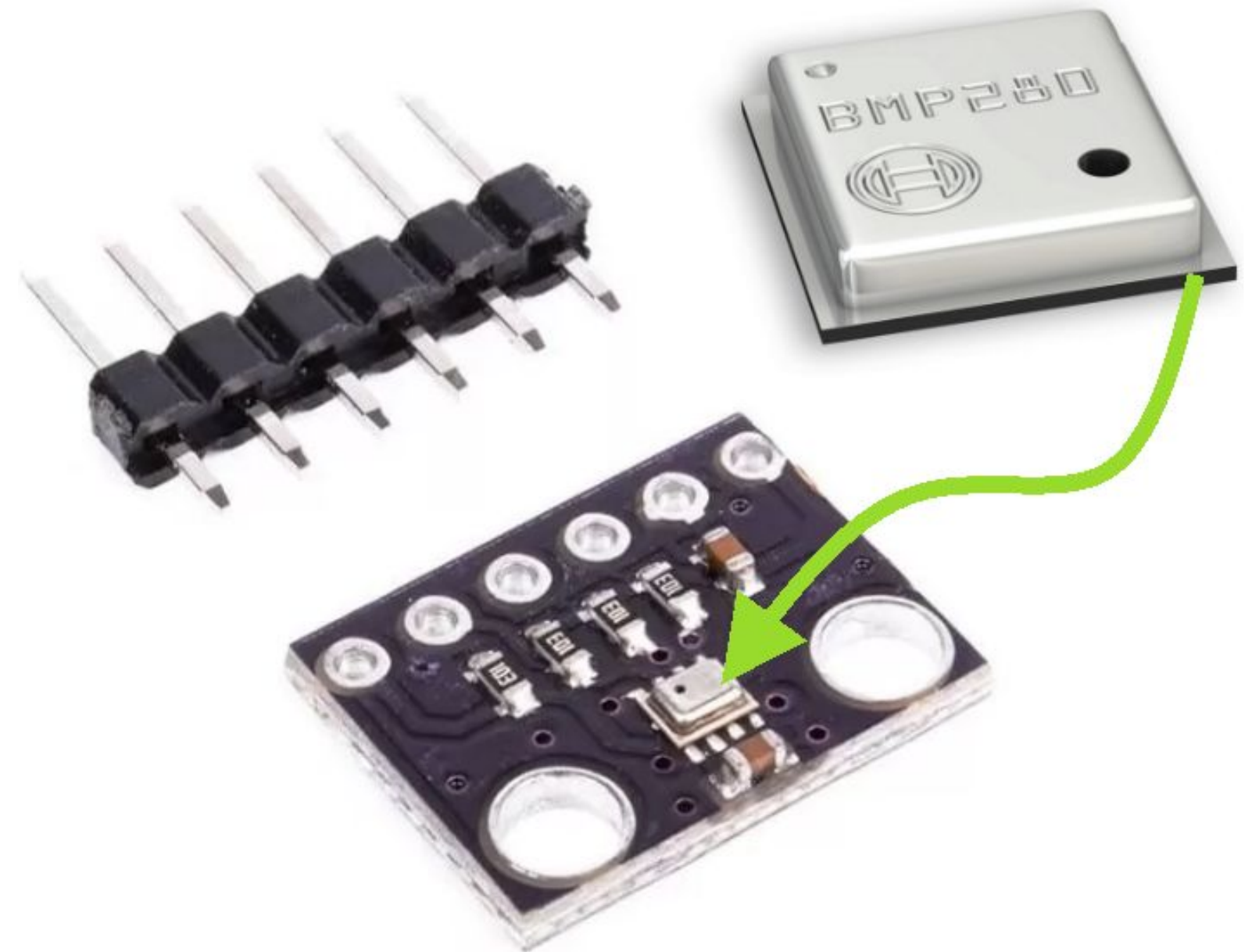
Функциональная схема платы ESP32 Pico kit



Изображение: demo-dijiudu.readthedocs.io

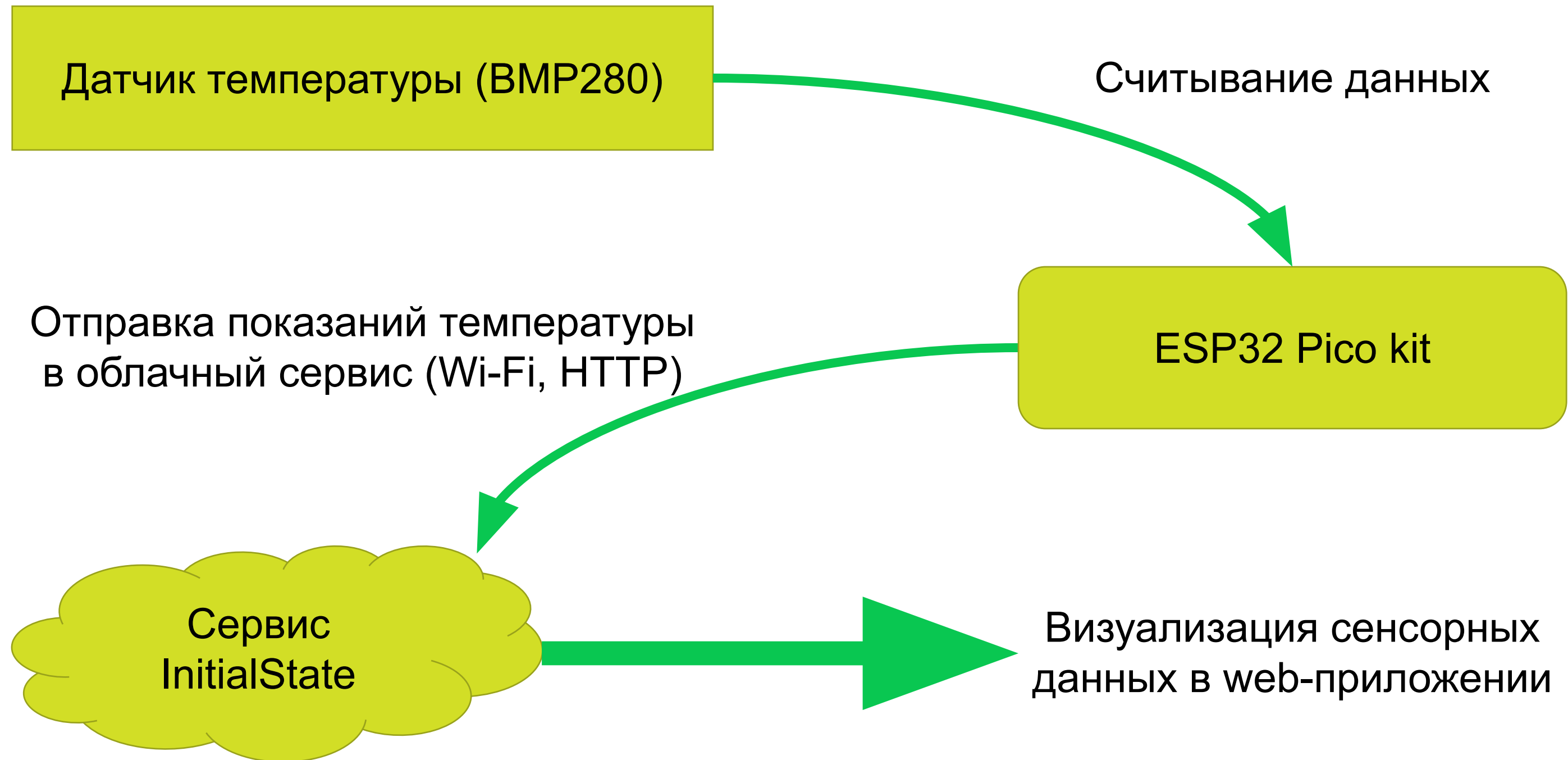
Датчик BMP280

- BMP280 — это цифровой датчик от Bosch Sensortec.
- Датчик способен измерять атмосферное давление и температуру окружающей среды.
- Специально разработан для мобильных приложений, где малый размер и низкое энергопотребление очень важны (ток потребления в рабочем режиме: **2.7мкА** при частоте опроса 1 Гц).
- Поддерживает 2 последовательных интерфейса обмена данными (**SPI и I2C**).
- Напряжение питания: **1.71 – 3.6В**.
- Диапазон измерения атмосферного давления: **300гПа – 1100гПа (± 0.12 гПа)**.
- Диапазон измерения температуры: **-40°C ... +85°C (± 0.01 °C)**.



Изображение: 3d-diy.ru и datasheets [BMP280](#)

Схема проекта



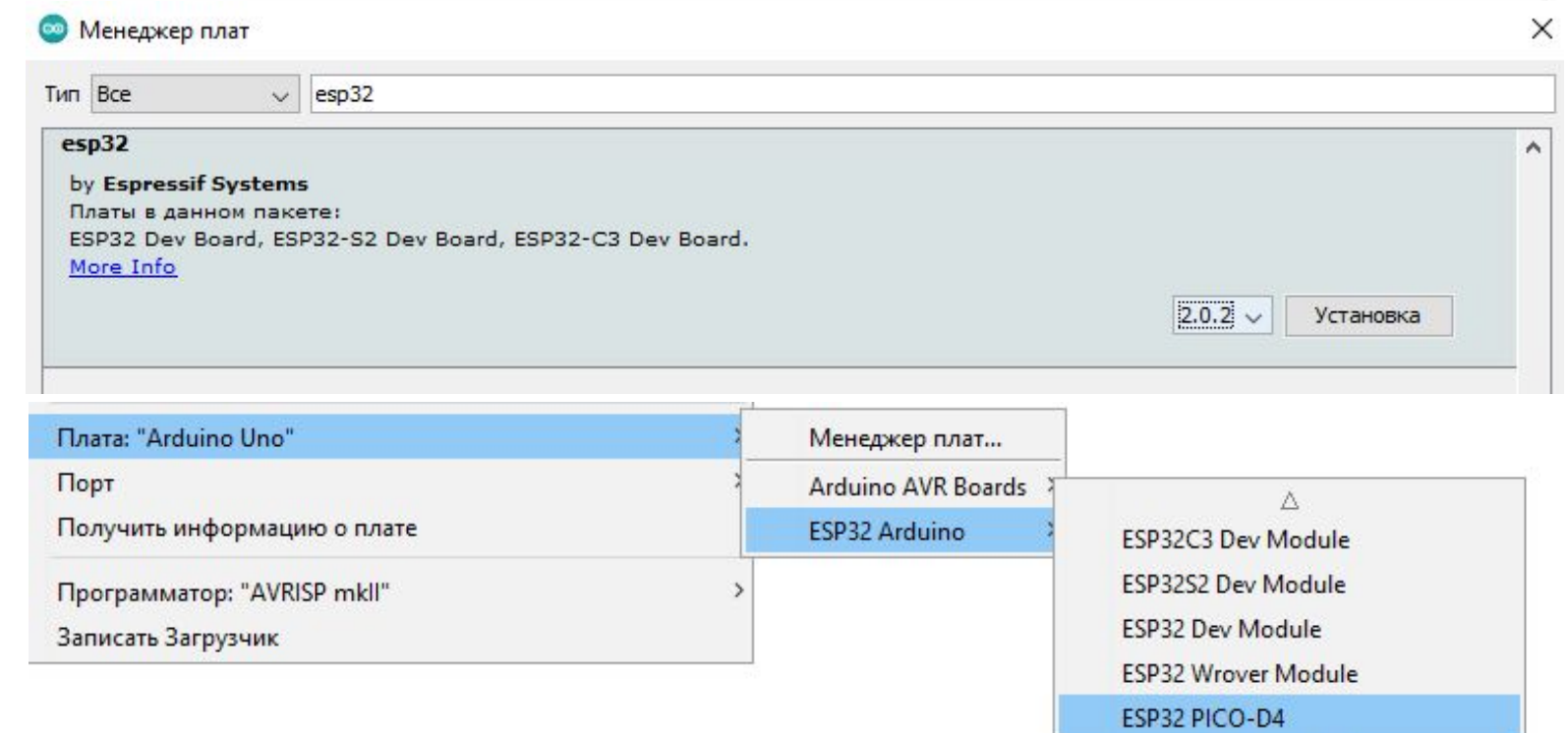
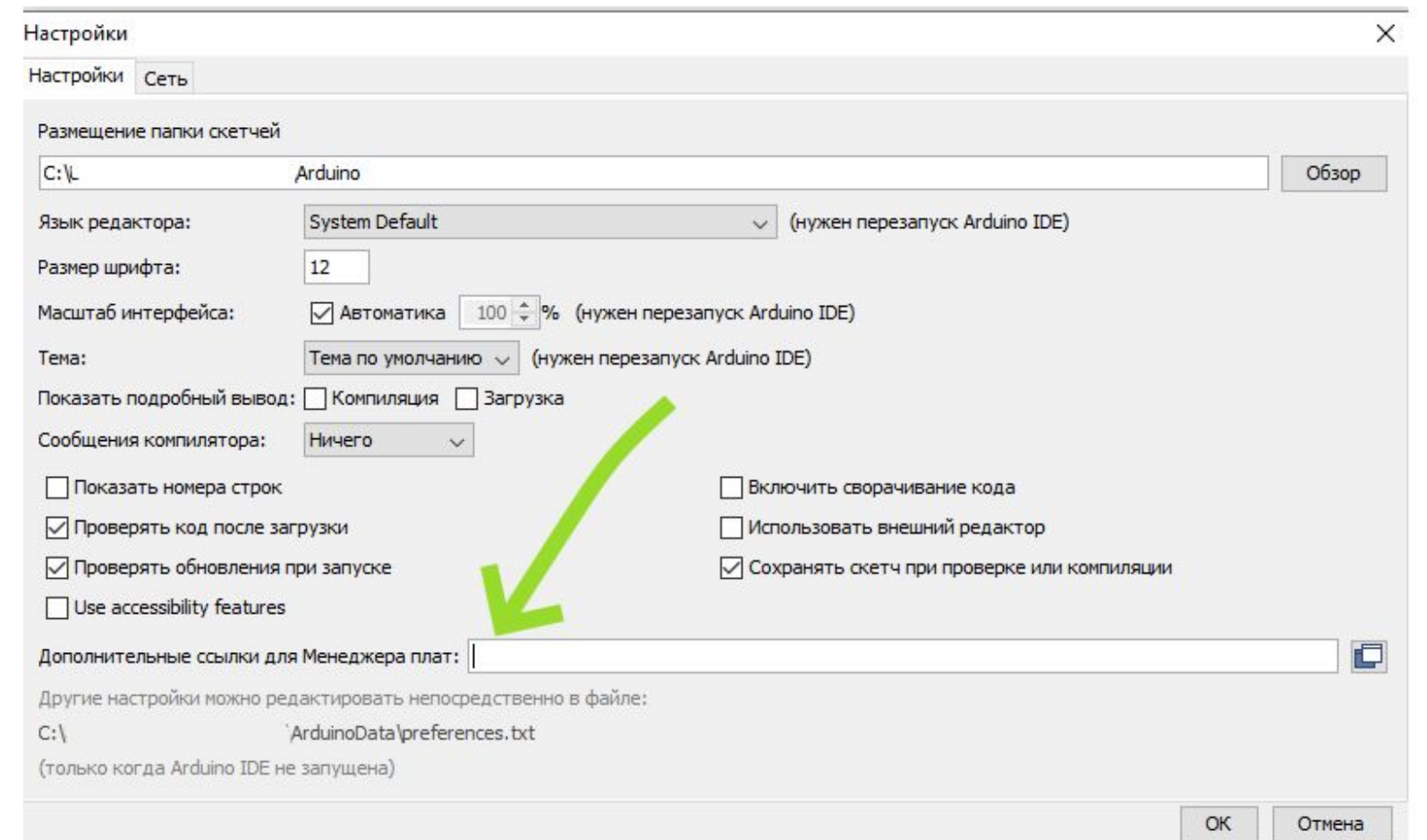
Установка ESP32 Arduino

- Установите текущую версию Arduino IDE версии 1.8.5 или выше (arduino.cc или Microsoft Store).
- Запустите Arduino и откройте окно настроек.
- Введите ссылку на стабильный выпуск в поле «*Additional Board Manager URLs / Дополнительные URL-адреса диспетчера плат*».

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

- Откройте Менеджер плат из меню Инструменты > Плата и установите платформу ESP32 v1.0.4 (v2.0.2).
- В списке плат найдите и выберите ESP32 Pico kit (PICO-D4).
- Подключите USB кабелем ESP32 к ПК
- Задайте COM порт (Инструменты > Порт > COM5)
- Перезапустите среду разработки Arduino IDE.

Arduino ESP32 старше версии v2.x.x нельзя использовать в Windows 8.x x86 (32-разрядная версия), Windows 7 или более ранних версиях.



1 Пример “Hello world!” (Blink)

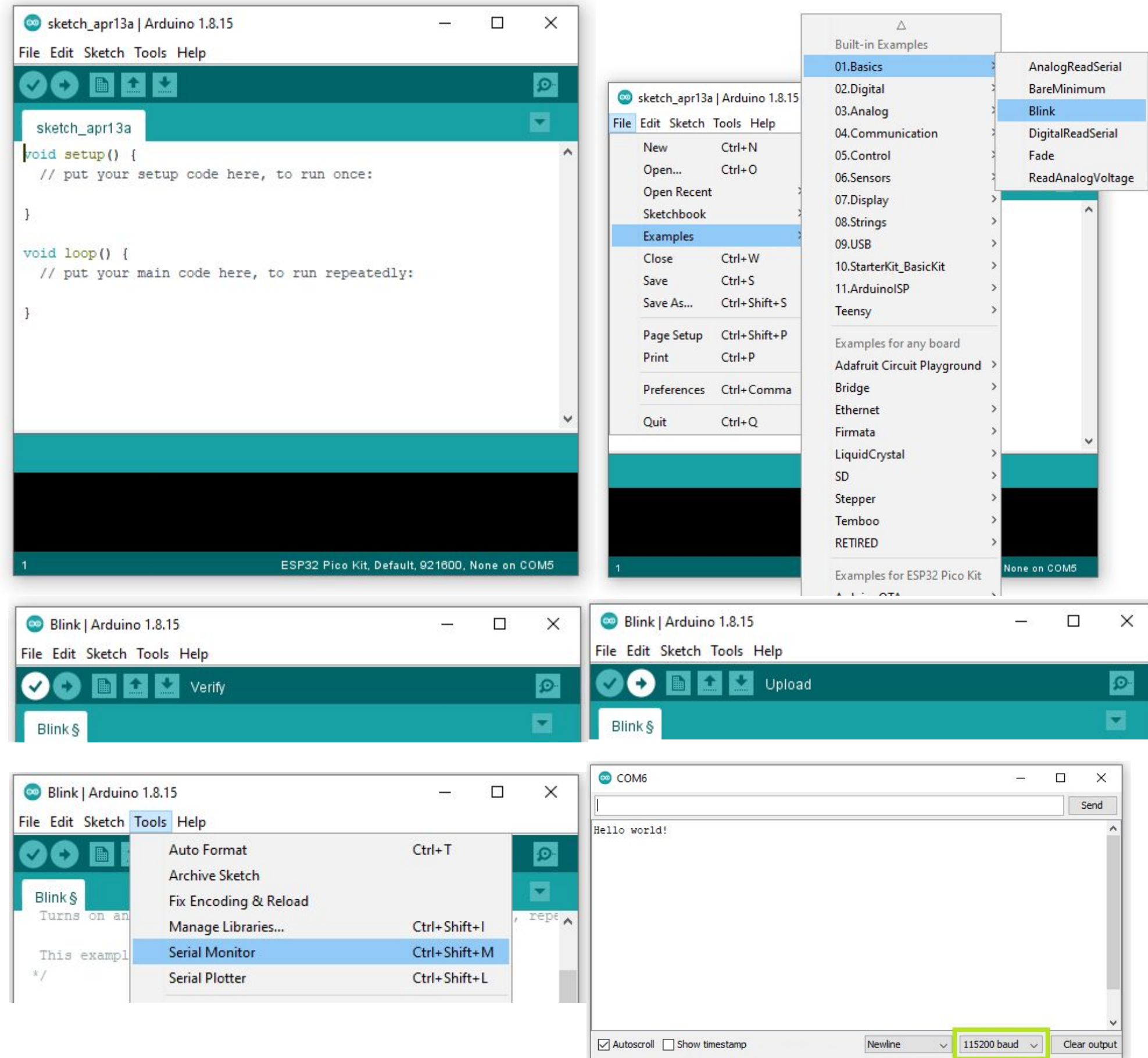
- Откройте Arduino IDE
- Откройте пример Blink
- Добавьте код настройки UART интерфейса в функцию setup (выполняется 1 раз)

```
Serial.begin(115200);
```

- Добавьте код передачи сообщения в функцию loop (выполняется в цикле)

```
Serial.println("Hello world!");
```

- Скомпилируйте и Загрузите прошивку в ESP32 kit
- Запустите Монитор принимаемых по UART сообщений
- Проверьте соответствие заданной скорости передачи данных (115200)
- Наблюдайте сообщения от ESP32

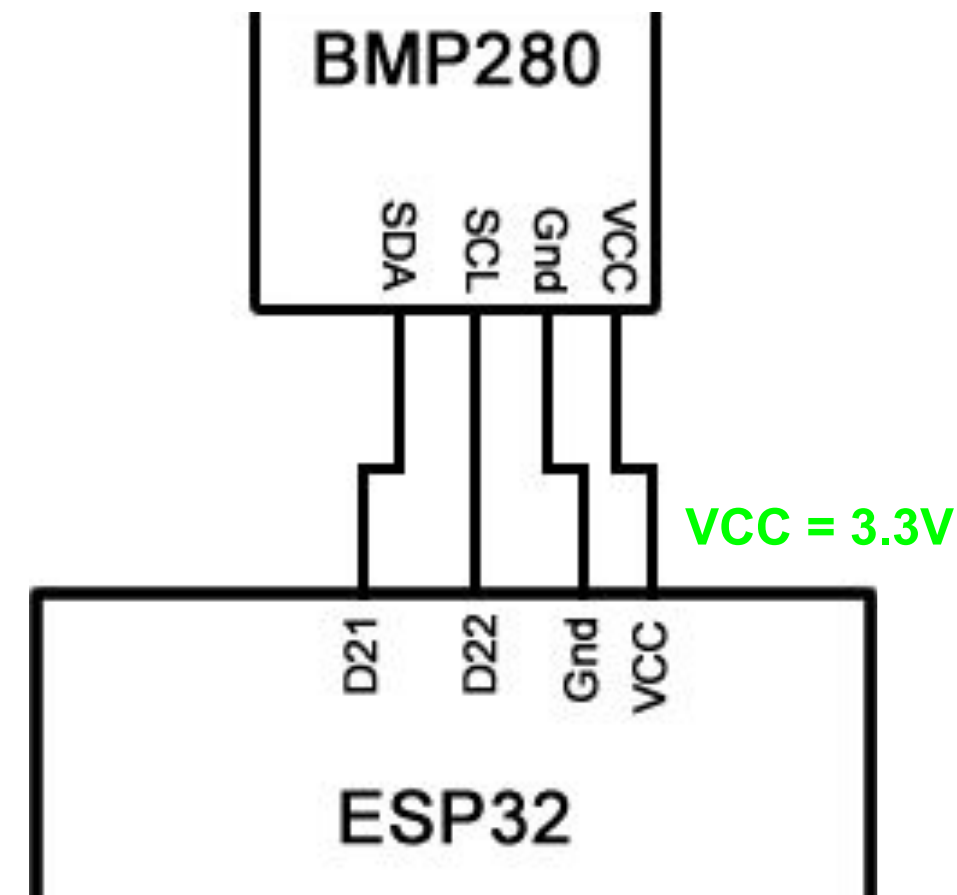


2 Подключение и использование датчика BMP280

- Подключите датчик по интерфейсу I2C (!VCC = 3.3V!, не 5V): VCC, GND, SDA, SCL.
- В Arduino IDE откройте диспетчер библиотек («Инструменты -> Управлять библиотеками...»).
- Установите библиотеку «Adafruit bmp280» v2.6.2 со всеми зависимостями
- Сделайте откат(установку) библиотеки «Adafruit BusIO» до версии v1.9.6
- Откройте меню «Файл -> Примеры -> Adafruit BMP280 Library -> bmp280test».
- Задайте адрес датчика на шине I2C

```
status = bmp.begin(BMP280_ADDRESS_ALT,  
BMP280_CHIPID);
```

- Скомпилируйте и загрузите скетч на отладочную плату.
- Откройте монитор порта (Ctrl+Shift+M), проверьте скорость соединения в правом нижнем углу (9600).
- Проверьте, что данные о температуре и давлении отображаются.

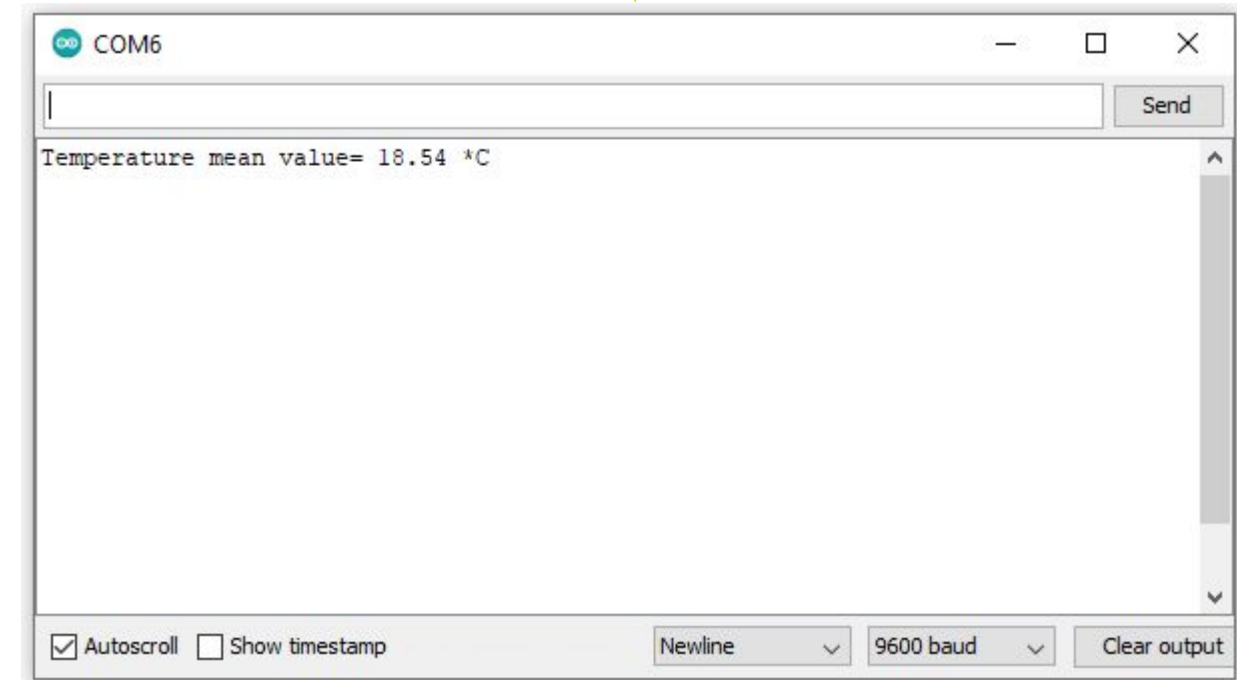
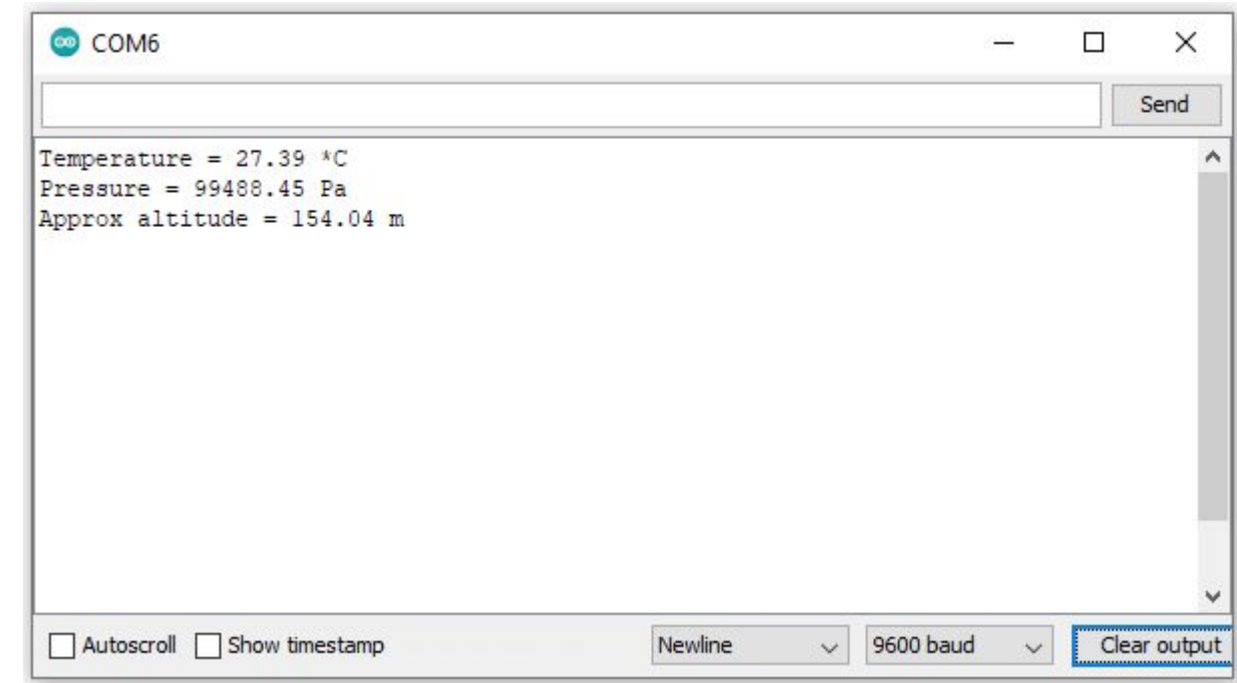


Изображение: xtronical.com



3 Подключение и использование датчика BMP2803

- Модернизируйте код примера, оставив отображение только данных о температуре.
- Реализуйте хранение данных трех предыдущих значений температуры.
- Реализуйте расчет и вывод средней температуры по трем измерениям в течение минуты.



Отправка данных в облако. Настройка сервиса InitialState

- Перейдите на сайт initialstate.com.
- Зарегистрируйтесь и зайдите в панель управления.
- В панели управления создайте новое приложение(1).
- При создании приложения укажите его название(2) и необходимые ключи (3).

New Stream Bucket ×

Name

New Stream Bucket ← 2

☐ Light Theme

Endpoint URL

`https://groker.init.st/api/events?acc` 📄

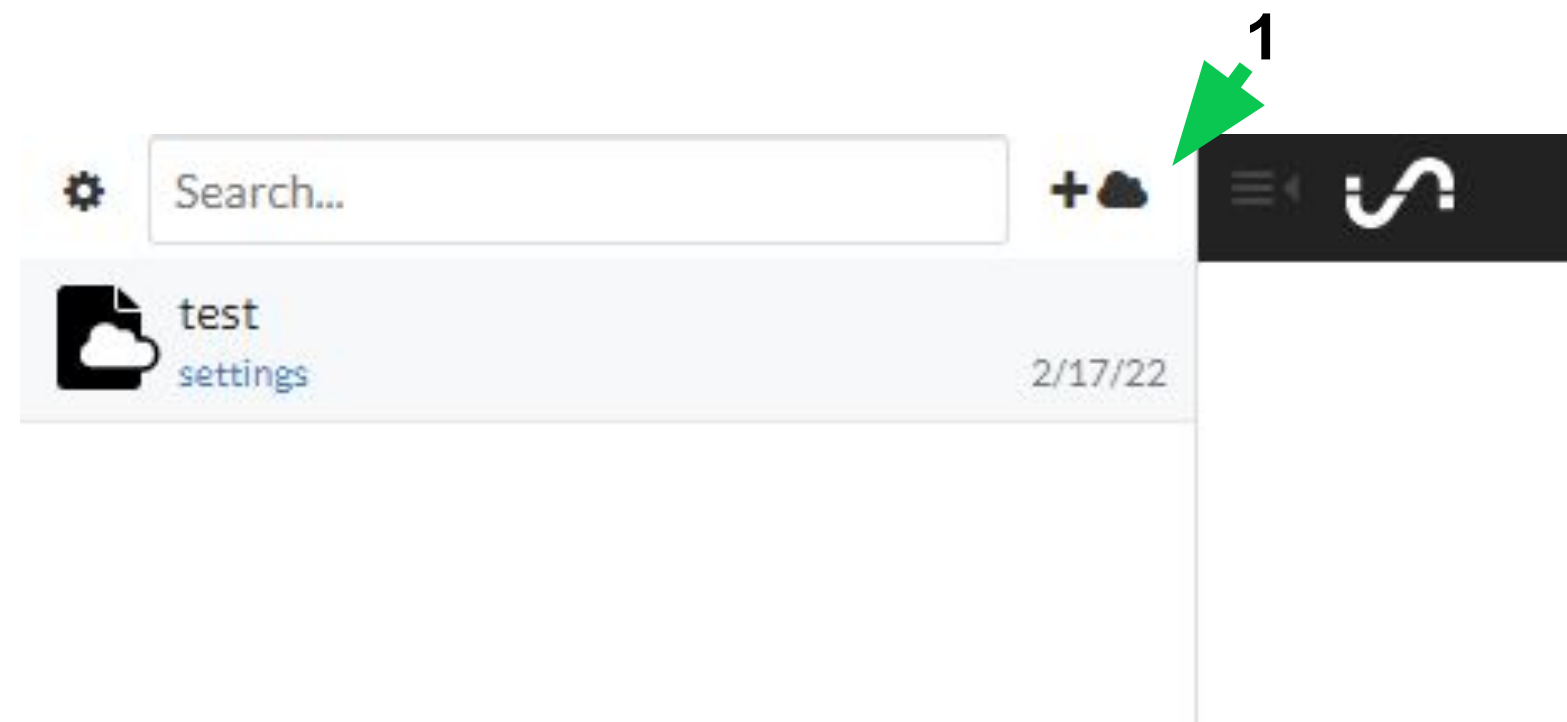
☒ Configure Endpoint Keys

Bucket Key

Lh ← 3

Access Key

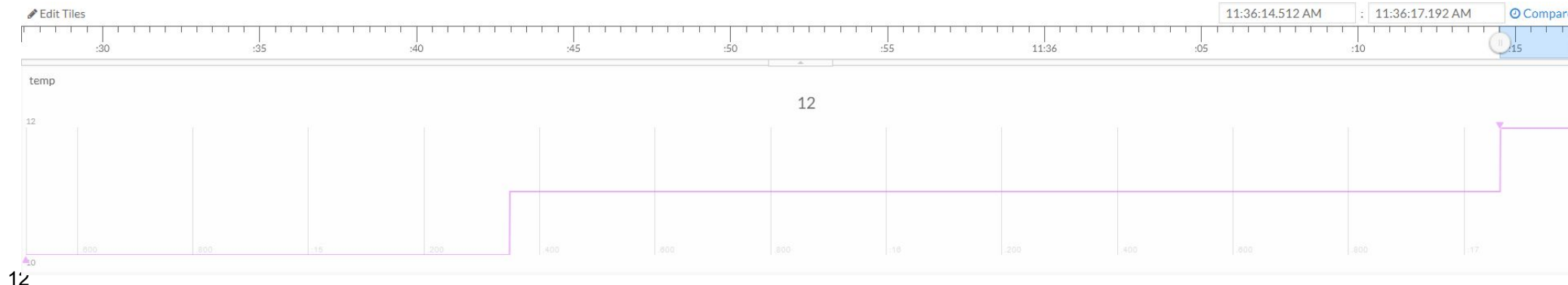
Create A New Access Key ▼ ← 3



Отправка данных в облако.

Проверка работы сервиса InitialState

- Перейдите в настройки приложения и скопируйте значение поля API Endpoint.
- Добавьте к копии значения из поля API Endpoint строку `&NameKey=Value`, где NameKey – ключ(имя переменной), а Value значение. (для редактирования текста можно использовать любой текстовый редактор/блокнот)
- Полностью URL для GET запроса будет иметь вид `https://groker.init.st/api/events?accessKey=accessKeyValue&bucketKey=bucketKeyValue&temperature=10`
- В любом удобном браузере с помощью сайта www.reqbin.com отправьте получившийся GET запрос
- Проверьте, что значение корректно отобразилось на платформе InitialState.
- Для удобства проверки, активируйте опцию «Allow Read Latest Value API», для просмотра текущего значения переменной.



Settings

Data Triggers Sharing

Name

test

☒ Light Theme

Bucket Key

V.

Access Key

ist_ 5n1 8A

API Endpoint

https://groker.init.st/api/events?ac

☐ Allow Read Latest Value API

Send Data Point

Key Value

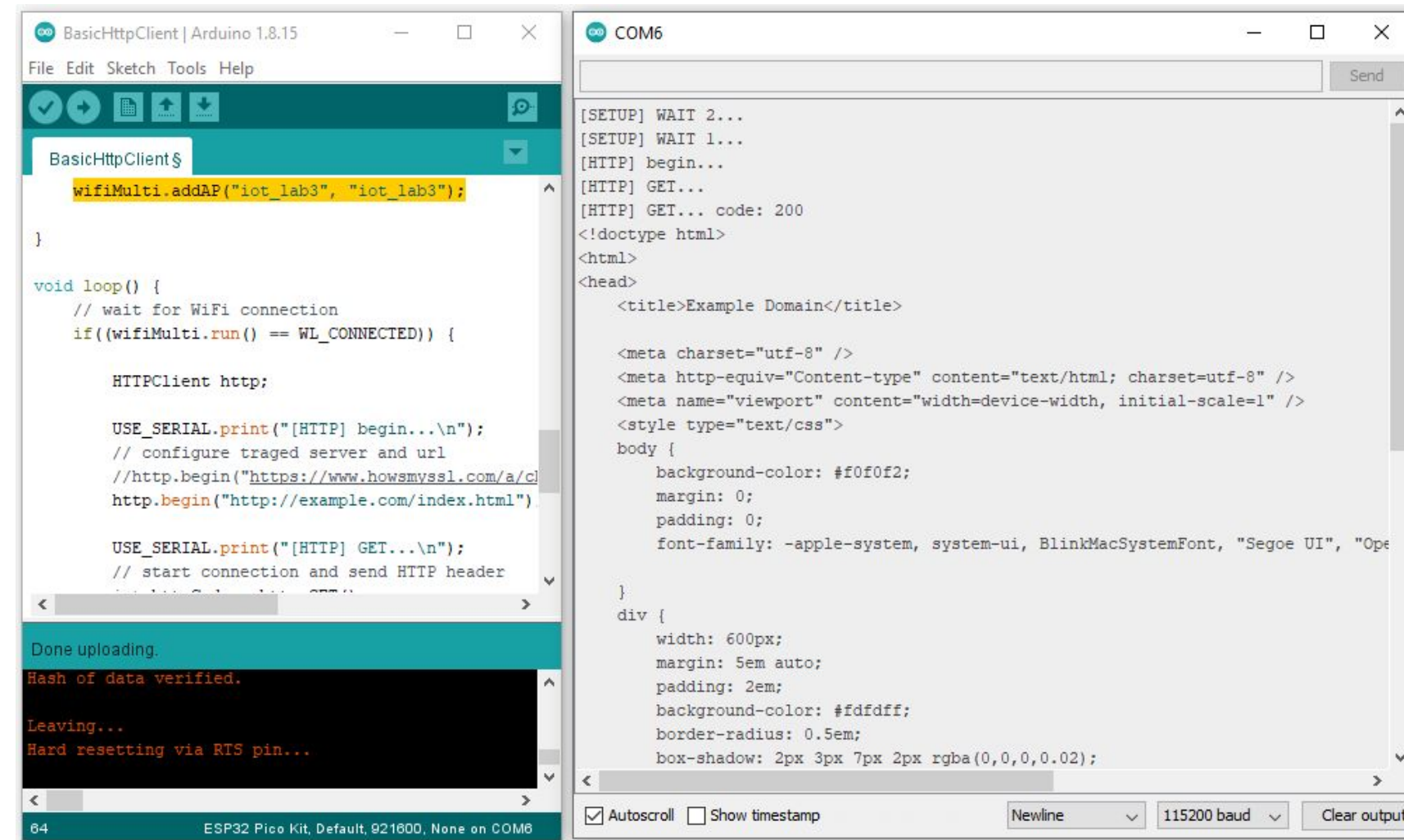
Advanced Settings

Import View Duplicate Remove

4 Отправка данных в облако.

Установка Wi-Fi соединения и передача GET запроса с помощью ESP32

- Откройте меню «Файл -> Примеры -> HTTPClient -> BasicHttpClient».
- Внутри функции `setup()` найдите строку:
`wifiMulti.addAP("SSID", "PASSWORD");`
- Передайте в метод `wifiMulti.addAP()` в качестве аргументов имя и пароль Wi-Fi сети.
`iot_lab3 / iot_lab3`
- Внутри функции `loop()` найдите строку:
`http.begin(*client, "URL");`
- Укажите в качестве аргумента для метода `begin` http URL для тестирования (`http://example.com/index.html`).
- Скомпилируйте и загрузите скетч на отладочную плату.
- Откройте монитор порта (Ctrl+Shift+M), проверьте скорость соединения в правом нижнем углу (115200).
- Проверьте, что в монитор выводится код веб-страницы.



The screenshot displays the Arduino IDE interface. The left pane shows the 'BasicHttpClient' sketch with the following code:

```
BasicHttpClient$
wifiMulti.addAP("iot_lab3", "iot_lab3");

}

void loop() {
  // wait for WiFi connection
  if((wifiMulti.run() == WL_CONNECTED)) {

    HTTPClient http;

    USE_SERIAL.print("[HTTP] begin...\n");
    // configure target server and url
    //http.begin("https://www.howsmyssl.com/a/c/grade");
    http.begin("http://example.com/index.html");

    USE_SERIAL.print("[HTTP] GET...\n");
    // start connection and send HTTP header
    http.GET();
  }
}
```

The right pane shows the serial monitor output for COM6:

```
[SETUP] WAIT 2...
[SETUP] WAIT 1...
[HTTP] begin...
[HTTP] GET...
[HTTP] GET... code: 200
<!doctype html>
<html>
<head>
  <title>Example Domain</title>

  <meta charset="utf-8" />
  <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <style type="text/css">
    body {
      background-color: #f0f0f2;
      margin: 0;
      padding: 0;
      font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans", sans-serif;
    }
  </style>
  <div>
    <h1>Example Domain</h1>
    <p>This domain is for use in illustrative examples in documents. You may use this domain in literature to represent a fictitious web page.
    </p>
  </div>
</html>
```

The bottom status bar indicates the board is 'ESP32 Pico Kit, Default, 921600, None on COM6'.

Фрагмент кода страницы <http://example.com/index.html>

5 Отправка данных в облако.

Передача данных о температуре на облачную платформу

- При помощи значения поля API Endpoint и информации из предыдущих пунктов настройте передачу данных температуры с ESP32 на платформу InitialState.

- <https://groker.init.st/api/>
vs
- <http://insecure-groker.initialstate.com/api/>
*используйте этот адрес для ESP32

Результат

- concatenate.c как пример объединения строк на Си



Thx

Исходный код заданий

