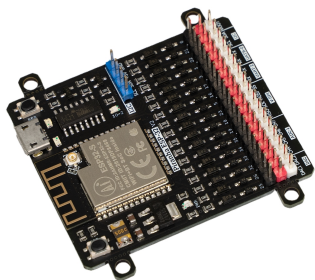


# Платформа Piranha ESP32 v2



## Общие сведения

Используйте [платформу Piranha ESP32 v2](#) на модуле Espressif ESP32-S для создания устройств на базе интернета вещей. Например, плата станет отличным контролером умного дома, веб-сервером для отображения показаний датчиков или мозгом робота управляемого по Wi-Fi или Bluetooth.

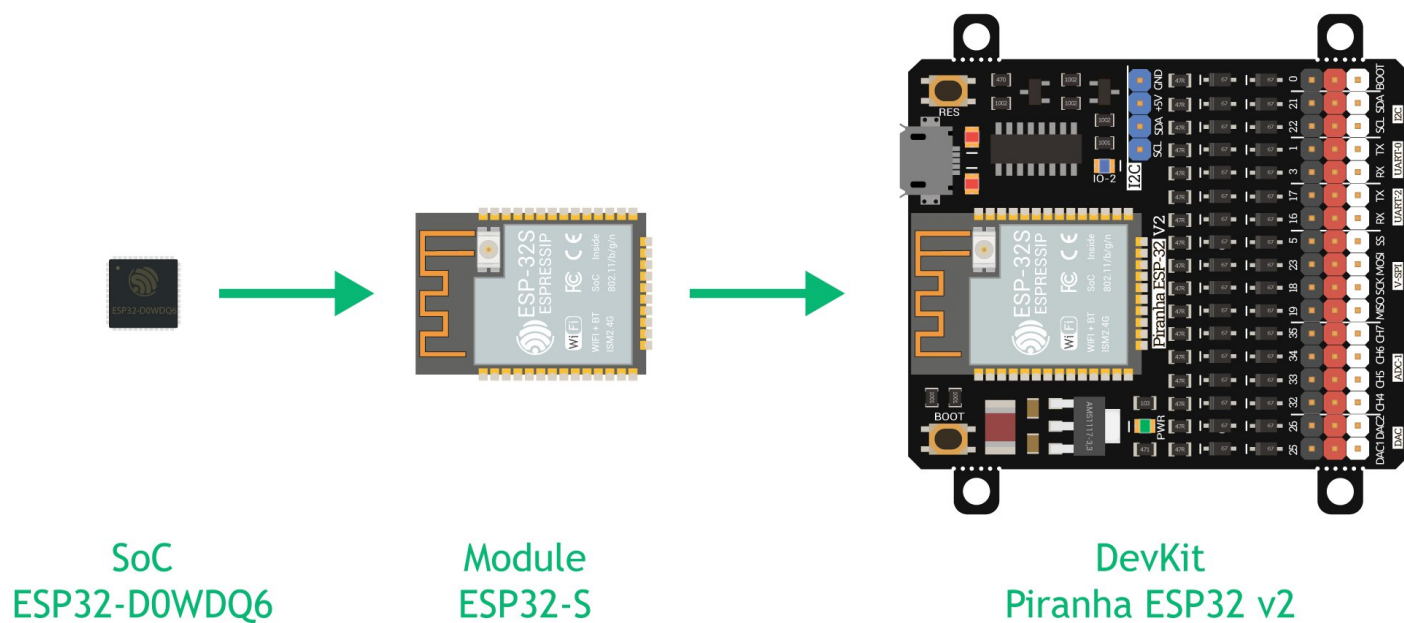
## Предыдущие версии

Платформа Piranha ESP32 v2 пришла на замену v1. Во второй версии мы внесли следующие изменения:

- Аналоговые входы могут работать параллельно с Wi-Fi
- Крепкий и надёжный micro-USB порт
- Разъём для подключение внешней антенны.

## Подробности

Платформа для разработки Piranha ESP32 v2 выполнена на модуле [ESP32-S](#) с чипом [ESP32-D0WDQ6](#). Кристалл SoC ESP32-D0WDQ6 работает на тактовой частоте до 240 МГц с поддержкой беспроводных технологий Wi-Fi 802.11 b/g/n и Bluetooth v4.2 с BLE.



На плате расположена гребёнка PLS в стиле G-V-S для подключения внешних [модулей](#) и [сенсоров](#):

- Всего контактов общего назначения: 17
- Контакты ввода-вывода GPIO: 15
- Контакты ввода GPI: 2

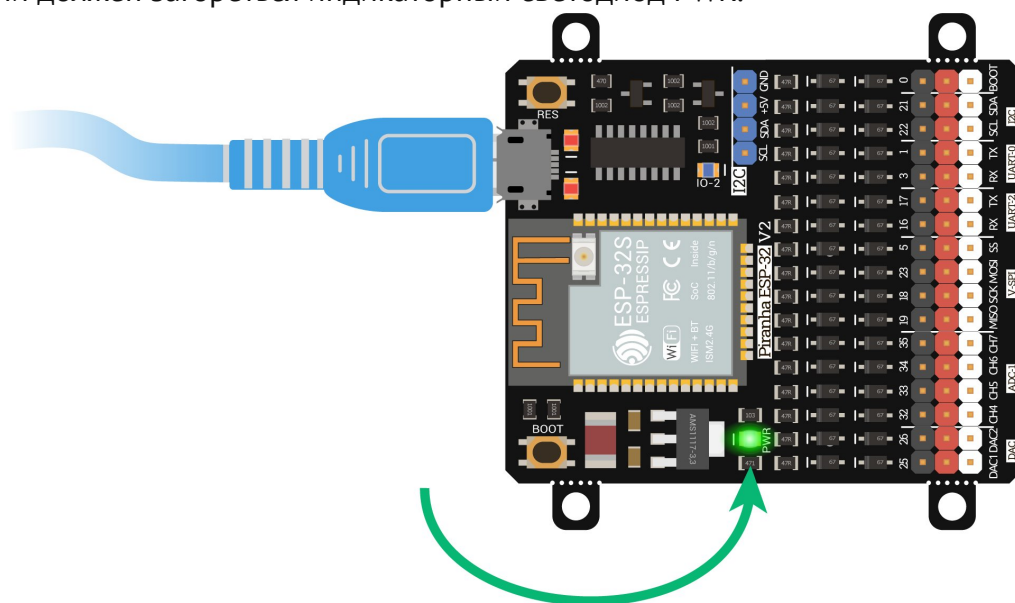
- Контакты с ШИМ: 15 с разрядностью до 16 бит (по умолчанию 8 бит)
- Контакты с АЦП: 6 с разрядностью до 16 бит (по умолчанию 14 бит)
- Контакты с ЦАП: 2 с разрядностью 8 бит
- Возможность сконфигурировать интерфейсы: 3×UART, 3×SPI, 2×I<sup>2</sup>C и 2×I<sup>2</sup>S

На плате также распаяна электронная обвязка, которая даёт всем пинам общего назначения совместимость с 5 вольтовой логикой.

Все подробности по контактам и интерфейсам платформы читайте в текущей статье в разделе [распиновка](#).

## Подключение и настройка

1. Подключите платформу Piranha ESP32 v2 к компьютеру через [кабель micro-USB](#). При успешном подключении должен загореться индикаторный светодиод PWR.



2. [Скачайте и установите среду программирования Arduino IDE.](#)
3. По умолчанию IDE настроена только на AVR-платы. Для работы с платформой Piranha ESP32 v2 – [добавьте в менеджере плат поддержку платформ ESP32 Arduino.](#)
4. Если при подключении контроллера Piranha ESP32 v2 в Arduino IDE не появляется новый COM-порт, установите драйвер USB-UART преобразователя для [Windows 7](#) или [Windows 10](#).
5. На этом настройка закончена, смело [переходите к примерам работы](#).

## Примеры работы

### Маячок

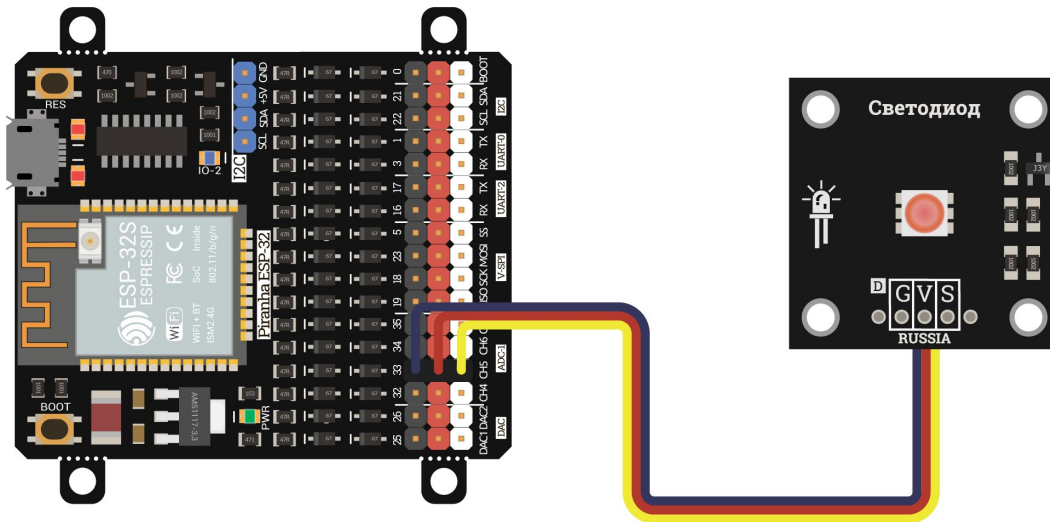
Для начала сделаем привычный «Hello, world» в DIY-мире – соберем маячок.

#### Что понадобится

- 1× [Piranha ESP32 v2](#)
- 1× [Кабель USB \(A – Micro USB\)](#)
- 1× [Светодиод \(Трета-модуль\)](#)

## Схема устройства

Подключите светодиод к Piranha ESP32 v2 к сигнальному пину 33 через трёхпроводной шлейф.



## Исходный код

Прошейте платформу Piranha ESP32 v2 кодом ниже.

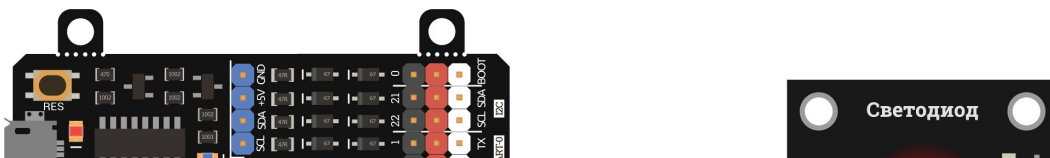
```
// GPIO пин, к которому подключен светодиод
constexpr int PIN_LED = 33;

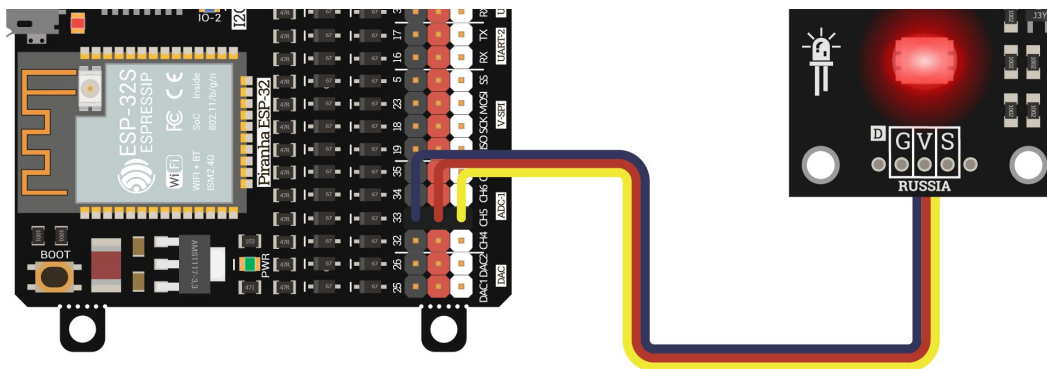
void setup() {
    // Настраиваем пин со светодиодом в режим выхода
    pinMode(PIN_LED, OUTPUT);
}

void loop() {
    // Зажигаем светодиод
    digitalWrite(PIN_LED, HIGH);
    // Ждём 1000 мс
    delay(1000);
    // Гасим светодиод
    digitalWrite(PIN_LED, LOW);
    // Ждём 1000 мс
    delay(1000);
}
```

## Результат

После прошивки, светодиод начнёт мигать раз в секунду.





## Интернет часы

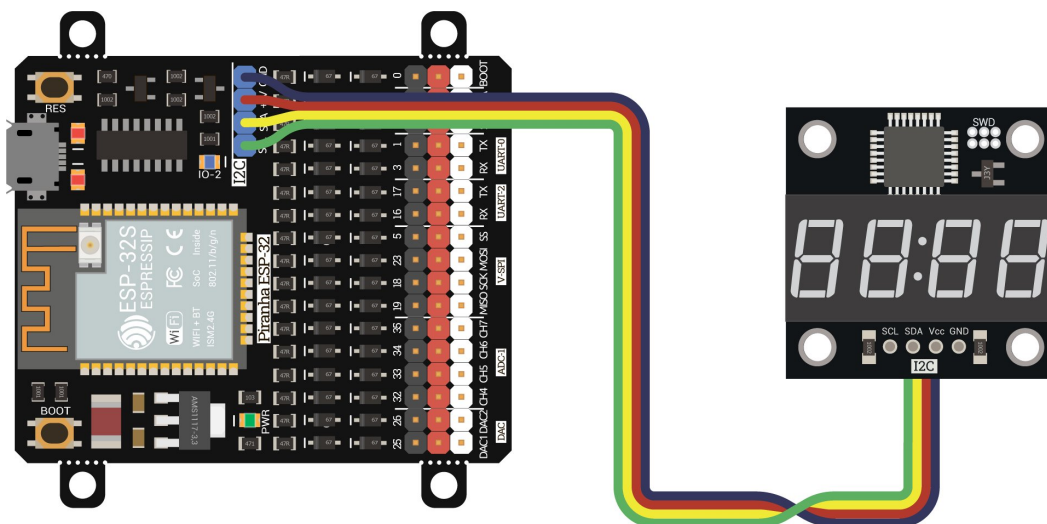
Мигать светодиодом можно на любой ардуинке. Давайте соберём что то связано с IoT, например интернет часы.

### Что понадобится

- 1× [Piranha ESP32 v2](#)
- 1× [Кабель USB \(A – Micro USB\)](#)
- 1× [Четырёхразрядный LED-индикатор Flash-I<sup>2</sup>C \(Тема-модуль\)](#)

### Схема устройства

Подключите LED-индикатор Flash-I<sup>2</sup>C к Piranha ESP32 v2 к сигнальным пинам I<sup>2</sup>C – SDA и SCL через четырёхпроводной шлейф.



### Исходный код

Прошейте платформу Piranha ESP32 v2 кодом ниже.

```
// Подключаем библиотеки для Piranha ESP32
#include <WiFi.h>
#include "time.h"
#include "sntp.h"
// Подключаем библиотеку для LED-индикатора Flash-I2C
#include <iarduino_I2C_4LED.h>
```

```

// Создаём объект для работы с LED-индикатором
// Передаём I2C по умолчанию 0x09
iarduino_I2C_4LED display4LED(0x09);

// Задаём точку доступа Wi-Fi сети и пароль
const char* ssid      = "your ssid";
const char* password  = "your password";

// Задаём сервера для получения времени
const char* ntpServer1 = "pool.ntp.org";
const char* ntpServer2 = "time.nist.gov";

// Устанавливаем временную зону TZ_Europe_Moscow
// Найти вашу временную зону можно тут:
// https://github.com/esp8266/Arduino/blob/master/cores/esp8266/TZ.h
const char* time_zone = "MSK-3";

void setup() {
  // Открываем Serial-порт
  Serial.begin(9600);

  // Инициализируем четырехразрядный LED-индикатор
  display4LED.begin();

  // Подключаемся к Wi-Fi сети
  Serial.printf("Connecting to %s ", ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  // Выдаём сообщение об успешном подключении
  Serial.println("\nConnected");

  // Устанавливаем временную зону
  configTzTime(time_zone, ntpServer1, ntpServer2);
}

void loop() {
  // Создаём объект для получения доступа к данным времени
  struct tm timeinfo;
  // Пытаемся обновить данные времени
  if(!getLocalTime(&timeinfo)){
    Serial.println("No time available...");
    return;
  }
  // Получаем текущее время в часах и минутах

```

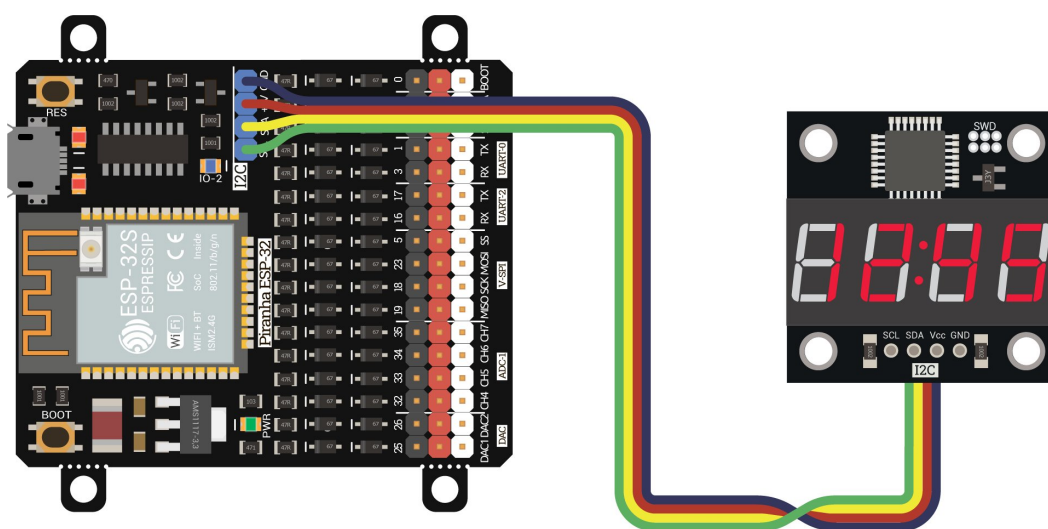
```

int hour = timeinfo.tm_hour;
int minute = timeinfo.tm_min;
// Выводим время на LED-индикатор
display4LED.print(hour, minute, TIME);
// Дублируем вывод времени в Serial-порт
Serial.printf("Time now %d:%d\n", hour, minute);
// Ждём 1 секунду
delay(1000);
}

```

## Результат

После прошивки, ESP32 выйдет в сеть → запросит время у сервера по NTP → отразит его на четырехразрядном LED-индикаторе. Для отладки программы время также дублируется в консоль.



## Элементы платы

Индикаторные  
светодиоды

Преобразователь  
USB-UART CH340

Порт  
micro-USB

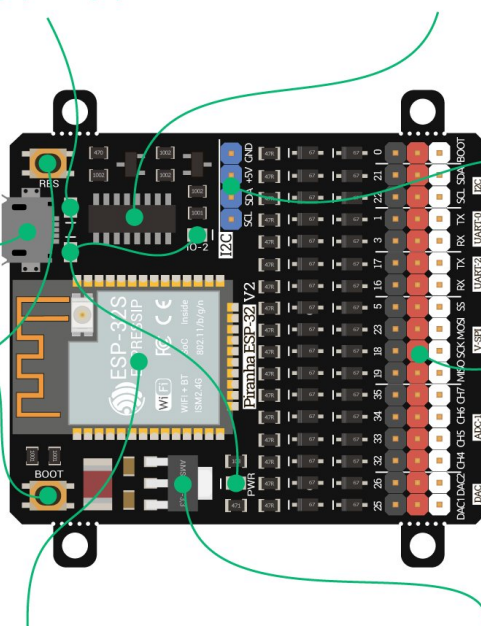
Контактная  
гребёнка I<sup>2</sup>C

Кнопки  
управления

Контактная  
гребёнка G-V-S

Модуль

Преобразователь



## Модуль ESP32-S

Платформа для разработки Piranha ESP32 v2 выполнена на модуле [ESP32-S](#).



Под металлической крышкой модуля [ESP32-S](#) расположена система на кристалле SoC [ESP32-D0WDQ6](#) с тактовой частотой до 240 МГц, Flash-память на 4 МБ и другая электронная обвязка. Также в кристалл ESP32-D0WDQ6 встроен датчик температуры, датчик Холла и контроллер сенсорного ввода. Поддерживаются беспроводные стандарты связи Wi-Fi 802.11 b/g/n (2,4 ГГц) и протокол Bluetooth v4.2 с BLE.

## Преобразователь USB-UART CH340

Преобразователь USB-UART на чипе [CH340](#) обеспечивает связь модуля ESP-12S с USB-портом компьютера. При подключении к ПК – Piranha ESP32 v2 определяется как виртуальный COM-порт.

Если при подключении контроллера Piranha ESP32 v2 в Arduino IDE не появляется новый COM-порт, установите драйвер USB-UART преобразователя для [Windows 7](#) или [Windows 10](#).

## Индикаторные светодиоды

Имя	Цвет	Назначение
IO-2	Синий	Индикатор питания платформы.
PWR	Зелёный	Пользовательский светодиод на GPIO33 пине микроконтроллера. При задании высокого уровня светодиод загорается, при низком – гаснет.
TX	Красный	Мигает при прошивки и отправление данных из Piranha ESP32 v2 в USB.
RX	Красный	Мигает при прошивки и получение данных из USB в Piranha ESP32 v2.

## Кнопка RESET



Кнопка RESET служит для ручного сброса питания — аналог кнопки RESET обычного компьютера.

## Кнопка BOOT

Кнопка BOOT служит для ручного перевода модуля в режим прошивки:

1. Зажмите кнопку BOOT;
2. Нажмите и отпустите кнопку RESET;
3. Отпустите кнопку BOOT.

При прошивки платформы через Arduino IDE специальная схема на плате сбрасывает микроконтроллер автоматически и никаких манипуляций с кнопкой проделывать не нужно. Однако кнопка всё же будет полезна, если вы хотите специально перевести модуль в режим прошивки или когда автоматический сброс не работает.

## Порт micro-USB

Разъём micro-USB предназначен для прошивки и питания платформы Piranha ESP32 v2 по USB. Для коммуникации понадобится [кабель Micro USB](#).

## Преобразователь питания

Понижающий преобразователь питания [AMS1117-3.3](#) с выходом 3,3 вольта, обеспечивает питание модуля ESP32-S. Максимальный выходной ток составляет 500 мА.

## Контактная гребёнка G-V-S

На плате расположена гребёнка PLS в стиле G-V-S для подключения внешних [модулей](#) и [сенсоров](#). Каждый контакт общего назначения подключен к электронной обвязки, которая даёт всем пинам общего назначения совместимость с 5 вольтовой логикой.

Все подробности по контактам и интерфейсам платформы читайте в текущей статье в разделе [распиновка](#).

## Контактная гребёнка I<sup>2</sup>C

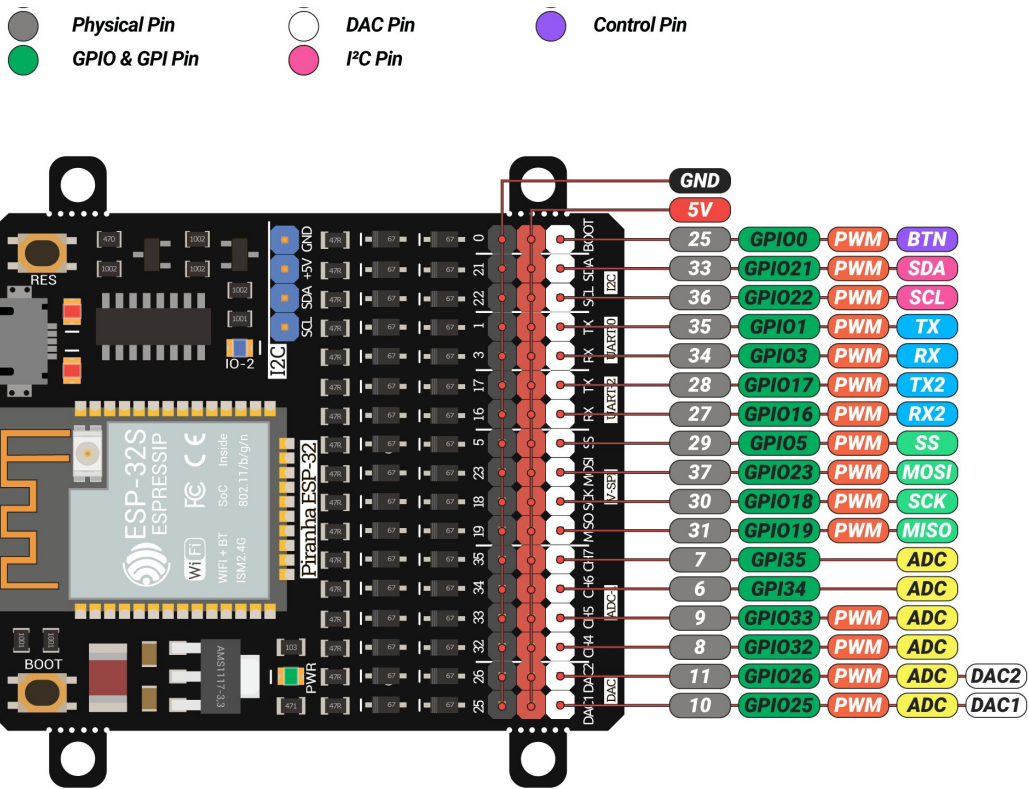
На плате расположена гребёнка PLS на 4 контакта для подключения внешних [модулей](#) и [сенсоров](#) по интерфейсу I<sup>2</sup>C. Контактная гребёнка I<sup>2</sup>C, дублирует пины питания и сигналов интерфейса I<sup>2</sup>C с общей колодки G-V-S.

Все подробности по контактам и интерфейсам платформы читайте в текущей статье в разделе [распиновка](#).

## Распиновка

# Piranha ESP32 v2





Все пины ввода-вывода совместимы с 5 вольтовой логикой.



Аппаратные интерфейсы UART, SPI, I<sup>2</sup>C, I<sup>2</sup>S проставлены на иллюстрации по умолчанию. Модуль ESP32 позволяет сконфигурировать на любых пинах ввода-вывода аппаратные интерфейсы : 3x UART, 3x SPI, 2x I<sup>2</sup>C и 3x I<sup>2</sup>S.

## Пины питания

Имя	Описание
5V	Пин питания с напряжением 5 вольт.
GND	Пин земли.

## Пины ввода-вывода

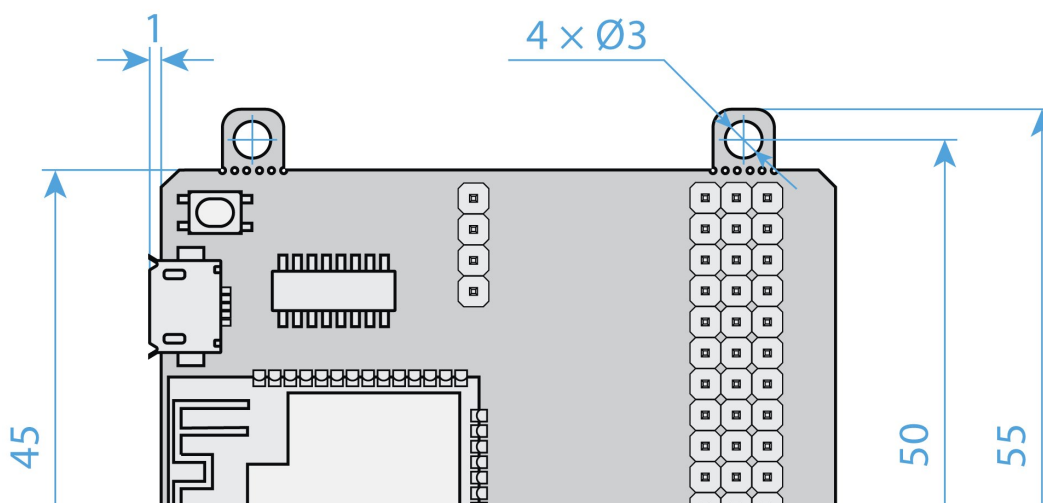
Все пины GPIO/GPI совместимы с 5 вольтовой логикой.

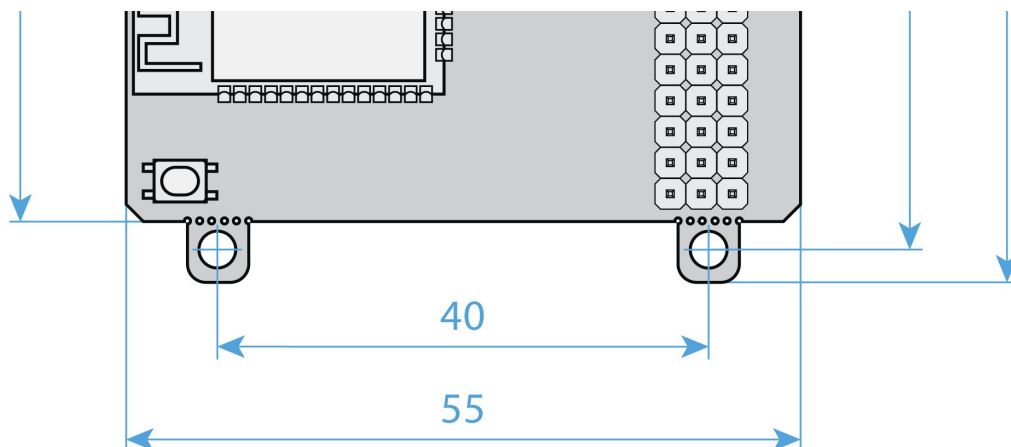
Категория	Количество пинов	Имена пинов	Описание
Пины ввода-вывода GPIO	15	GPIO0, GPIO1, GPIO3, GPIO5, GPIO16–GPIO19, GPIO21–GPIO23, GPIO25,	Пины могут быть настроены на вход или на выход. Логический уровень выходной единицы – 3,3 В, нуля – 0 В. На вход можно подавать сигнал до 5 В.

		GPIO26, GPIO32 и GPIO33.	
Пины ввода GPI	2	GPI34 и GPI35.	Пины могут быть настроены только на вход. На вход можно подавать сигнал до 5 В.
ШИМ / PWM	15	GPIO13, GPIO15 – GPIO19, GPIO21, GPIO22, GPIO25, GPIO26, GPIO32 и GPIO33.	Позволяет выводить аналоговое напряжение в виде ШИМ-сигнала из цифровых значений. Разрядность ШИМ по умолчанию установлена в 8 бит, но можно поднять до 16 бит.
АЦП / ADC	6	GPIO25, GPIO26, GPIO32, GPIO33, GPI34 и GPI36.	Позволяет представить аналоговое напряжение в виде цифровом виде. Диапазон входного напряжения от 0 до 3,3 В. Можно подавать сигнал и до 5 В, однако микроконтроллер будет его воспринимать как 3,3 В. Разрядность АЦП по умолчанию установлена в 14 бит, но можно поднять до 16 бит.
ЦАП / DAC	2	GPIO25 и GPIO26.	Позволяет выводить аналоговое напряжение в чистом виде. Разрядность ЦАП установлена в 8 бит и не меняется.

Каждый пин ввода-вывода GPIO платформы можно сконфигурировать на аппаратные интерфейсы: 3×UART, 3×SPI, 2×I<sup>2</sup>C и 2×I<sup>2</sup>S.

## Габаритный чертёж





## Характеристики

- Модель: Piranha ESP32 v2
- Интерфейс USB-UART: CH340
- Процессор: ESP32-S
- Стандарт Wi-Fi: 802.11 b/g/n
- Стандарт Bluetooth: BLE v4.2 BR/EDR
- Тактовая частота: 240 МГц
- Flash-память: 4 МБ
- SRAM-память: 520 КБ
- Входное напряжение питания: 5 В
- Напряжение логических уровней: 3,3 В с толератностью к 5 В
- Контакты общего назначения:
  - Всего контактов: 17
  - Контакты ввода-вывода GPIO: 15
  - Контакты ввода GPI: 2
  - Контакты с ШИМ: 15 с разрядностью до 16 бит (по умолчанию 8 бит)
  - Контакты с АЦП: 6 с разрядностью до 16 бит (по умолчанию 14 бит)
  - Контакты с ЦАП: 2 с разрядностью 8 бит
  - Возможность сконфигурировать интерфейсы: 3×UART, 3×SPI, 2×I<sup>2</sup>C и 2×I<sup>2</sup>S