**Инструкция по работе с ESP8266**

**Содержание**

1. [Описание микроконтроллера Esp8266 и отладочных плат на его основе](#_1._Описание_микроконтроллера)
   1. [Введение, о микроконтроллерах и отладочных платах](#_1.1._Введение,_о)
   2. [Характеристики микроконтроллера Esp8266EX](#_1.2._Характеристики_микроконтроллер)
2. [Установка и настройка среды разработки](#_2._Установка_и)
   1. [Настройка Visual Studio Code с фреймворком Platformio](#_2.1._Настройка_Visual)
   2. [Настройки среды Eclipse](#_2.2._Настройки_среды)
3. [Основы работы с ESP8266](#_3._Основы_работы)

# 1. Описание микроконтроллера Esp8266 и отладочных плат на его основе

## 1.1. Введение, о микроконтроллерах и отладочных платах

*Микроконтроллер* (далее МК) – это, своего рода, маленький компьютер со своим ядром, АЛУ, контроллером прерываний, оперативной памятью для хранения данных (ОЗУ, SRAM), которая сохраняет данные только во время выполнения программы и постоянной Flash-памятью для хранения кода программы (ПЗУ, ROM) и прочего.

*Отладочная плата* – это печатная плата с МК. На ней расположено всё так, чтобы пользователю было удобно работать с МК. Ведь, например, сам МК esp8266EX выглядит вот так:

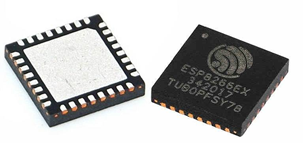


Рис. 1.1.1 – МК Esp8266EX

Согласитесь, работать с ним в таком виде, мягко говоря, не удобно. Для того и нужны отладочные платы, чтобы вывести все выводы МК на удобные для использования пины, а также подключить к нему удобный интерфейс (например, miniUsb, microUsb или Type-c) для связи с компьютером. Например, отладочная плата «WEMOS D1 mini V 3.0.0» с этим МК выглядит вот так:



Рис. 1.1.2 – Отладочная плата Wemos d1 mini v3.0.0

Здесь и сама микросхема Esp8266EX, внешняя Flash-память программ, интерфейс microUsb, куча удобных выводов, к которым можно припаять гребенки и много ещё всего (usb-ttl преобразователь или USB-UART преобразователь соединяет два интерфейса. Он преобразует то, что пришло по USB в другой интерфейс общения, понятный МК – UART, чтобы МК спокойно мог принимать программу, пришедшую к нему с компьютера).

На базе МК ESP8266EX существует целая куча отладочных плат, отличающихся способами подключения, объёмами внешней памяти и прочим. Вот некоторые из них:



Рис. 1.1.3 – Виды отладочных плат на базе МК ESP8266EX. Внутри каждой такой платы сидит именно этот МК, поэтому встроенными возможностями все они не отличаются.

Я же использую маленькую отладочную плату с удобным microUsb разъёмом – «WEMOS D1 mini». У меня есть две платы, «Wemos d1 mini v2» и «wemos d1 mini v3.0.0»:



Рис. 1.1.4. – Версия 2 слвева и версия 3.0.0 справа

Могу порекомендовать брать либо версию 3.0.0 или выше, либо версию pro. У меня версия 2 (которая с этим железным коробом) работала нестабильно. Теперь к характеристикам самого МК, на базе которого сделана любая плата этой серии.

## 1.2. Характеристики микроконтроллера ESP8266EX

Все характеристики ESP8266EX можно посмотреть на [сайте](https://espressif.com) производителя в его [datasheet](https://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf) (технической документации). Здесь перечислю основные:

|  |  |
| --- | --- |
| Разрядность | 32 битная |
| Процессор | Tensilica Xtensa L106 |
| Максимальная тактовая частота | 80/160 МГц (его можно разогнать до 160 МГц, но стабильная работа тогда не будет гарантирована. |
| Напряжение питания | 3 Вольта для самого МК и 5 вольт для всех отладочных плат с usb интерфейсом |
| Flash память программ (ПЗУ, ROM) | У этого МК нет внутренней памяти программ (программы выполняются из внешней Flash-spi памяти. У всех плат она разная. У обоих моих плат она 4 Мбайта. |
| Память данных (ОЗУ, SRAM) | У всех плат одинаковая, т.к. встроена внутрь МК и равна 80 Кбайтам. Однако, библиотеки по настройке этого МК также используют её и пользователю остаётся около 50 кбайт. |
| Порты ввода/вывода | 14 штук (использовать можно только 11) |
| Аналоговый вход | 1, разрядность 10-бит |
| Интерфейсы | SPI – 4 шт, I2C – 1, I2S – 2, UART - 2 |
| Режимы WiFi | Клиент, Программная точка доступа, Клиент+Программная точка доступа (station, softAP, station+softAP) (поддерживается только Wifi с 2,4 ГГц,Wifi 5G не поддерживается) |
| Защита | WPA-PSK, WPA2-PSK |
| Шифрование | WEP , TKIP, AES |
| Протоколы WiFi | 802.11 b/g/n |
| Технологии | STBC, 1x1 MIMO, 2x1 MIMO |

# 2. Установка и настройка среды разработки

В общем, существует множество различных сред разработки для КМ типа esp8266. Самыми популярными можно выделить следующие:

* Arduino IDE
* Visual Studio Code с фраймворком (PlatformIO или ESP-IDF)
* Eclipse + SDK

Про Ардуино писать здесь не буду, т.к. порог вхождения туда ничтожный, а уроков по нему тьма. В общем, в нём можно только использовать библиотеки, что не даёт никакого понимания о том, как работает МК (можно сравнить с SPL в МК STM32, где напрямую работают с регистрами). Опишу установку в VScode и eclipse.

***Увы, но ни одна из этих платформ не поддерживает отладку ESP8266 (я не нашёл в них полноценных отладчиков для этого МК.***

## 2.1. Настройка Visual Studio Code с фреймворком Platformio

Первое, что нужно для программирования любого микроконтроллера (далее МК) – это среда разработки (или IDE). Я буду использовать Visual Studio Code, хотя можно пользоваться и любой другой средой – Arduino IDE, VS, Eclipse и т.д.

Скачать VScode можно с [официального сайта](https://code.visualstudio.com/).

После установки VScode, можно установить одно полезное расширение, добавляющее русский интерфейс. Для этого сбоку слева нажмите на кнопку «Расширения» и в поисковой строке введите «russian». Выберите расширение с иконкой глобуса и установите его. Так весь интерфейс станет русским.

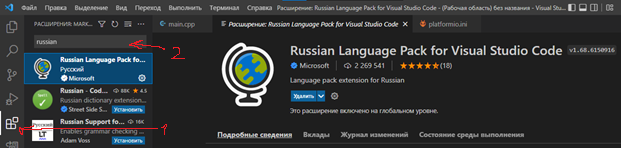


Рис. 1.1.1 – Установка расширения, русифицирующего VSCode

Второе расширение, которое уже является обязательным – это экосистема для разработки на множестве различных МК, в том числе на esp8266.

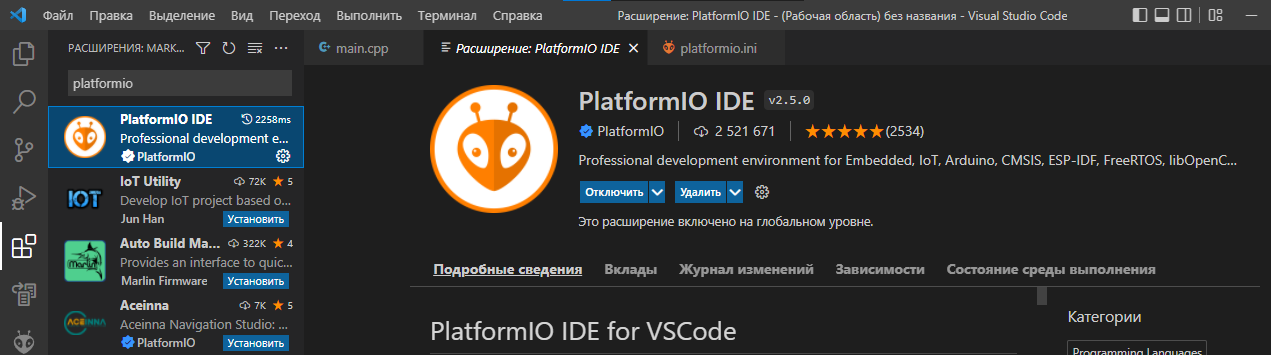


Рис. 1.1.2 – Установка расширения Platformio

Вместе с этим расширением также автоматически установится и расширение C/C++, отвечающее за работу с этими языками программирования. После этого вас, вероятно, попросят перезапустить VSCode.

После того, как PlatformIO установится, надо установить платформу для того МК, над которым мы будем работать. После установки у вас в нижнем левом углу появится значок домика – это главная страница PlatformIO. Жмём её:

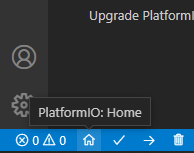


Рис. 1.1.3 – Переход на главную страницу PlatformIO

Теперь выбираем раздел с платформами, в нём выбираем раздел встроенные (embedded) и устанавливаем платформу для esp8266, которая называется «Espressif 8266».

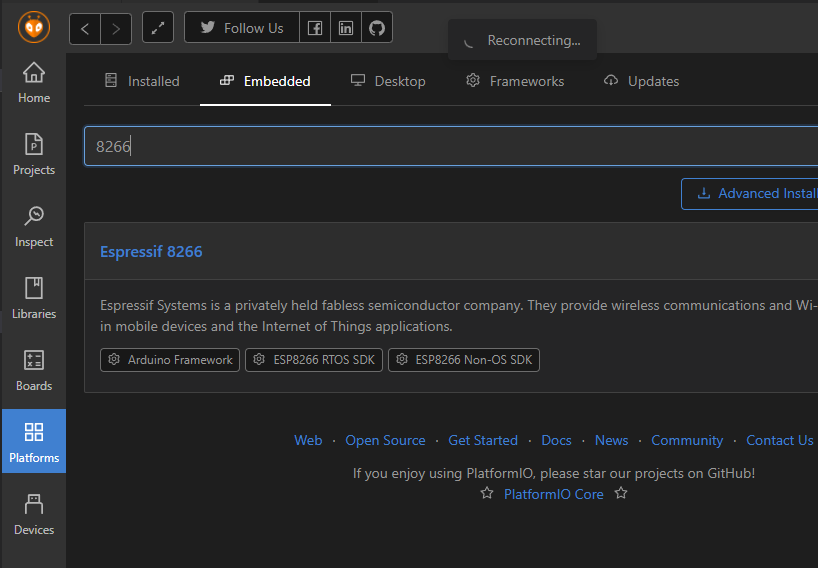


Рис. 1.1.2 – Установка платформы для МК

После этого можно вернуться на домашнюю страницу и создать первый проект:

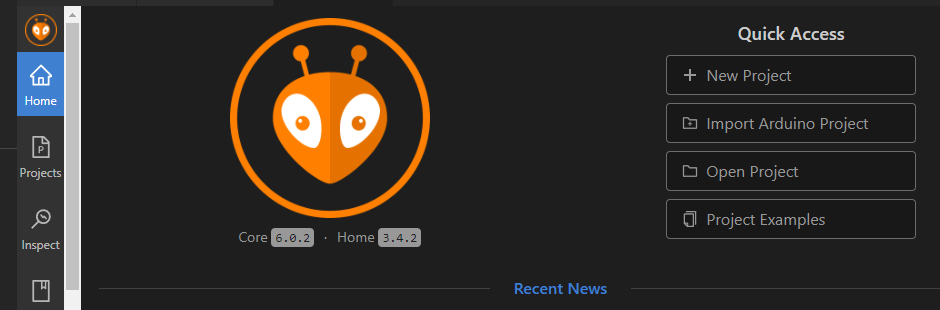


Рис. 1.1.3 – Создание проекта

При создании проекта надо указать его имя, плату, которую собираетесь программировать (у меня это Wemos D1 mini) и используемый фраемворк. Также, если снять галку, то можно указать, где именно создавать проект.

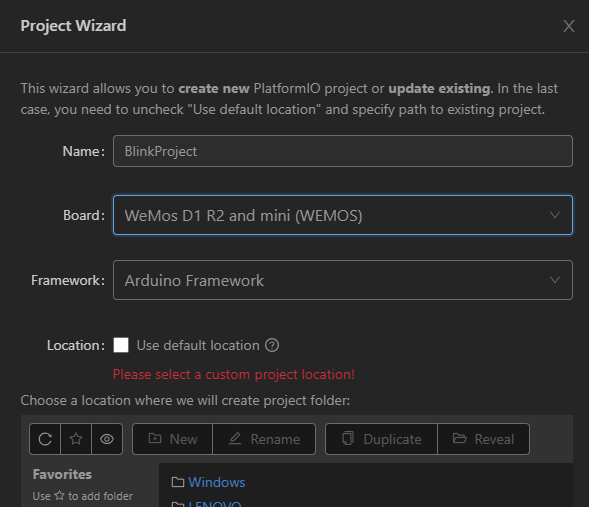


Рис. 1.1.4 – Создание проекта

После того, как проект будет создан, в левой части отображается его структура:

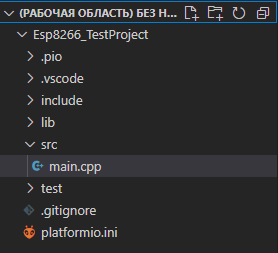


Рис. 1.1.5 –Структура проекта

В папке src уже находится главный файл main.cpp. Также важным файлом является platformio.ini, в котором находятся настройки платформы. Рекомендую сразу указать скорость заливки прошивки в МК (upload\_speed), а также скорость UART (115200):

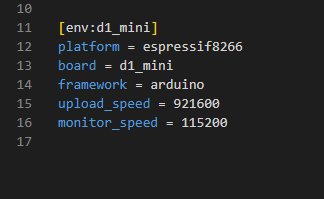


Рис. 1.1.6 – Настройки платформы

Теперь всё готово для написания кода.

Хочу заметить один важный момент. При подключении вашей платы к ПК, на ней должен начать мигать светодиод. Также, проверьте, что диспетчер устройств видит вашу плату (при заливке прошивки может возникнуть ошибка, говорящая, что-такого-то COM-порта не существует). Если в диспетчере устройств платы нет, то это говорит о том, что в вашем USB проводе нет проводов для передачи данных, а есть только провода питания. Смените провод и проверьте, что ПК видит ваше устройство в диспетчере устройств в Com-портах (в моём случае там отображался USB-UART мост CH340, установленный на плате).

## 2.2. Настройки среды Eclipse

Скачать Eclipse можно [тут](https://www.eclipse.org/downloads/).

# 3. Основы работы с ESP8266

3.1. Зашивка МК по воздуху – технология OTA (Over the air)