1. Arduino IDE

Arduino IDE – программа для написания и загрузки прошивки в плату, скачать можно с официального сайта (рисуноки 1,2) вот здесь *(Внимание! Перевод языка страницы в браузере ломает кнопки!)*. Перед загрузкой вам предложат пожертвовать на развитие проекта, можно отказаться и нажать **JUST DOWNLOAD** (только скачать).

Рисунок 1 – Как скачать IDE

Рисунок 2 – Скачивание среды

Для работы рекомендуется компьютер с **Winodws 7** или выше, либо **Linux/MacOS**

- Если у вас **Windows XP**, придётся установить версию 1.6.13, более свежие версии будут очень сильно тормозить или не будут работать вообще. Есть ещё одна проблема: некоторые библиотеки не будут работать на старых

версиях Arduino IDE, также не будет работать поддержка плат семейства esp8266, поэтому крайне рекомендуется обновить свой компьютер до Windows 7 или выше

- Установка на Linux из системного репозитория – читать тут
- Установка на MacOS – читать тут

Arduino Windows app
Не рекомендуется устанавливать Arduino Windows app из магазина приложений Windows, так как с ней бывают проблемы
Другие версии
Не устанавливайте старые версии IDE, если нет на то весомых причин, а также beta и hourly-билды

Java

Для старых версий Arduino IDE, а также для некоторых других программ, понадобится пакет **Java JRE**. Скачать можно с официального сайта для своей операционной системы.

Установка

Arduino IDE устанавливается как обычная программа, запускам и жмём далее далее далее…

Драйвер

Во время установки Arduino IDE программа попросит разрешения установить драйвера от неизвестного производителя, нужно согласиться на установку всего предложенного (рисунок 3).

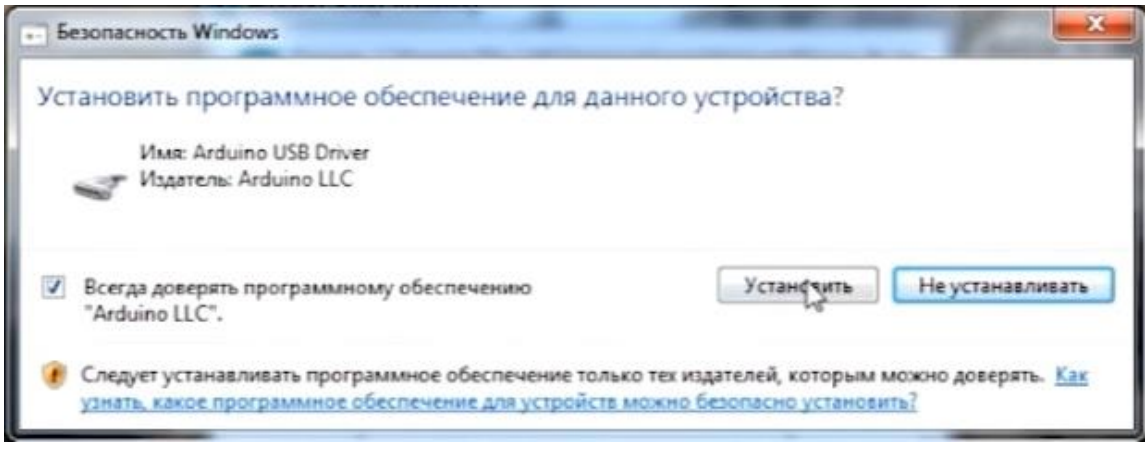

Рисунок 3 – Установка дополнительных драйверов

Обновление

Перед установкой новой версии нужно удалить старую. Ни в коем случае не удаляйте папку установленной IDE из Program Files, удалять нужно через "***Установка и удаление программ***", либо запустив файл *uninstall.exe* из папки с установленной программой. Иначе установщик откажется устанавливать новую программу, так как в системе остались следы от старой. Решение этой проблемы описано в видео ниже. Вкратце о том, как удалить IDE вручную:

Удаляем папки:

- Папка с программой
  - *C:\Program Files (x86)**Arduino***\ (64-битная версия Windows)
  - *C:\Program Files**Arduino***\ (32-битная версия Windows)
- Папка со скетчами и библиотеками
  - *Документы**Arduino***\
- Папка с настройками и дополнительными "ядрами" плат
  - *C:\Пользователи (или Users)\Ваш_пользователь\AppData\Local**Arduino15***\

Удаляем следы из реестра (рисунок 4):

- Открыть *редактор системного реестра*:
  - Windows 10: *Пуск/**regedit***
  - Предыдущие: *Пуск/Выполнить/**regedit***
  - Инструкция для всех Windows
- В открывшемся окне: *Правка/Найти…*
  - В окне поиска пишем **arduino\uninstall**
  - Поиск
- Удаляем найденный параметр (см. скриншот ниже)
- На всякий случай *Правка/Найти далее*
- Удаляем и так далее, пока не удалим все найденные параметры с **arduino\uninstall**
- После этого можно запускать установщик и устанавливать новую программу

Другие проблемы

- Если перестала запускаться Arduino IDE – удаляем файлик **preferences.txt** из *C:\Пользователи (или Users)\Ваш_пользователь\AppData\Local\Arduino15*

Портативная версия

Вместо полной установки программы можно скачать архив с уже “установленной”, на странице загрузки он называется **Windows ZIP file**. Вот прямая ссылка на 1.8.13. Распаковав архив, получим портативную версию Arduino IDE, которую можно скинуть на флешку и использовать на любом компьютере без установки программы. Но понадобится установить драйвер CH341 для китайских плат, а также драйверы из папки с программой Arduino IDE (подробнее в следующем уроке). Возможно понадобится установить **Java**.

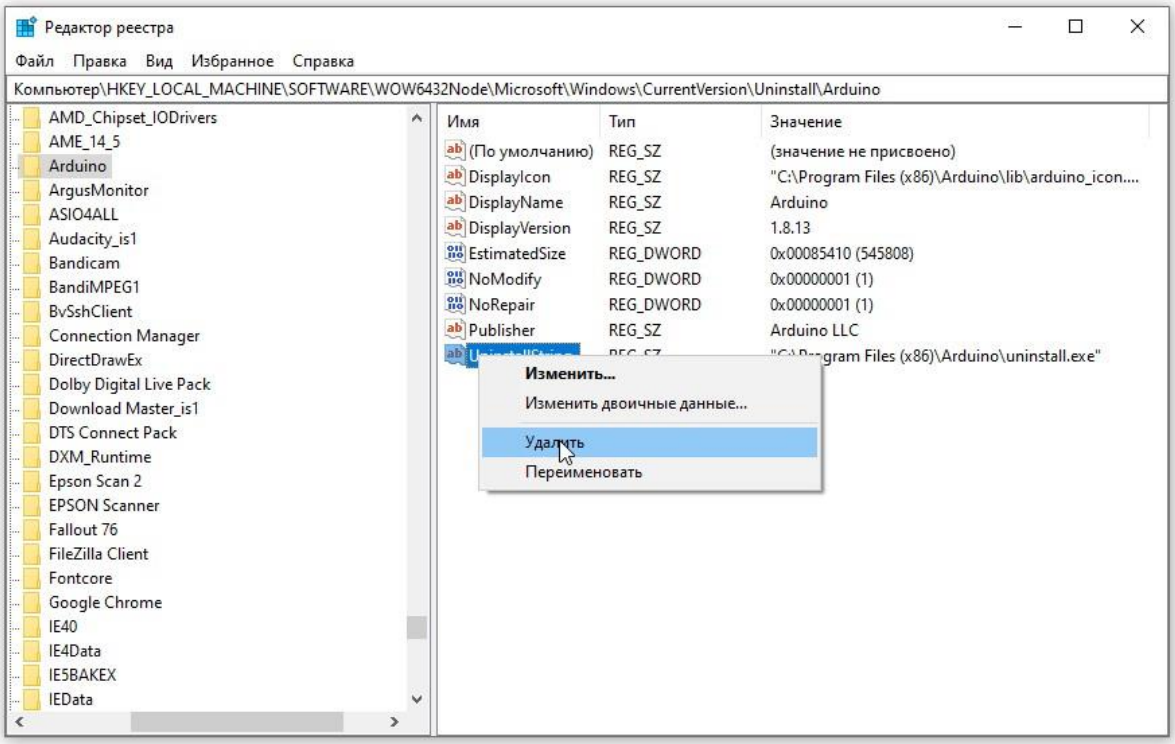

Рисунок 4 – Удаление программы из реестра

Работа на смартфоне

Писать и загружать прошивку через смартфон тоже можно, понадобится смартфон на Android и приложение ArduinoDroid. Также для тренировки и удобного редактирования скетчей можно использовать CppDroid, но загружать в плату она не умеет.

Осмотр платы

Перед подключением к компьютеру рекомендуется провести визуальный осмотр платы на предмет дефектов пайки компонентов. Что можно встретить (в порядке рисунков):

- Замкнутые пины (вроде бы паяются китайцами вручную)
- Непripаянная нога компонента
- “Торчащие” вверх или под углом компоненты – резисторы и конденсаторы, припаянные только с одной стороны
- Компоненты со смещением
- пропой между ногами компонента

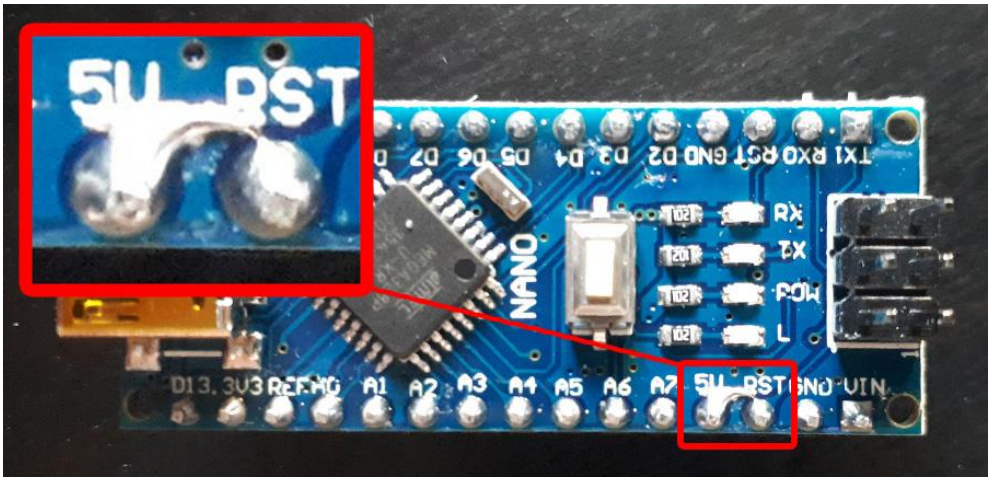

Рисунок 5 – Пропой между ногами компонентов

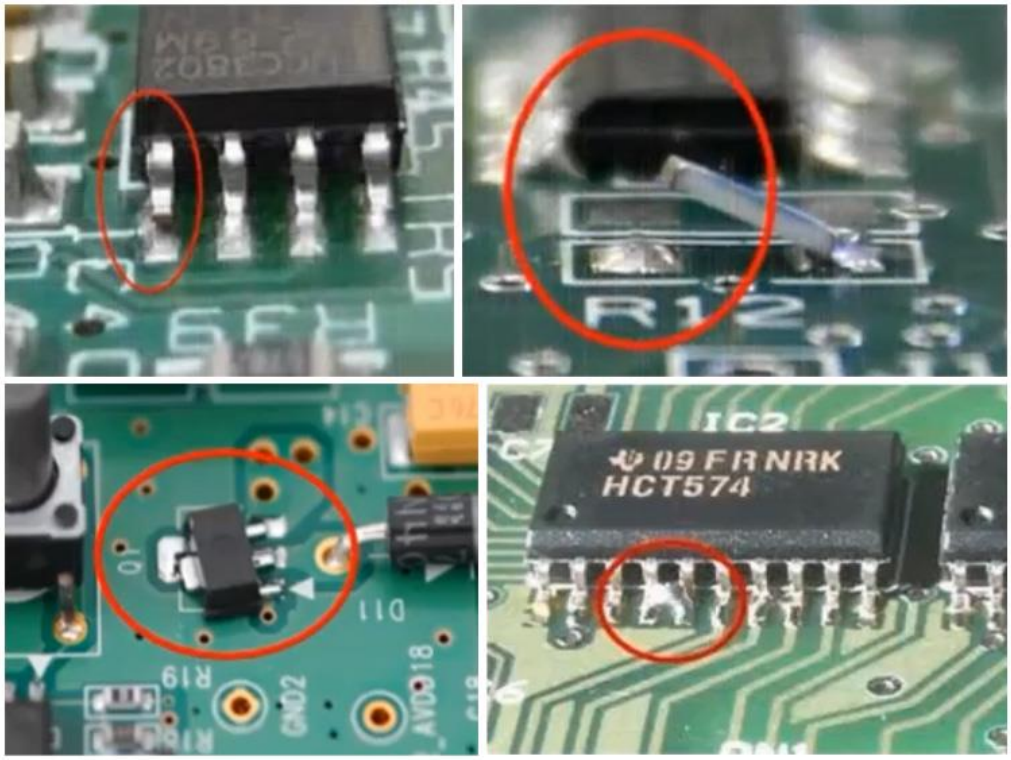

Рисунок 6 – Другие виды дефектов

Плату с обнаруженным дефектом не рекомендуется подключать к компьютеру! Всё можно исправить паяльником, если не умеете сами – попросите того, кто умеет.

Реакция на подключение питания

Как понять, что плата работает корректно? На примере Nano/Uno:

- При подключении USB загорается и горит светодиод **PWR**
- Если плата новая и на ней прошит загрузчик (он обязан быть прошит) – *однократно* мигает светодиод **L**
- *Примечание: светодиоды могут быть любого цвета*
- На новой плате прошито “мигание светодиодом”, поэтому светодиод **L** продолжит мигать один или два раза в секунду в зависимости от версии загрузчика
- При нажатии на кнопку сброса (**RESET**, единственная кнопка на плате) должен однократно мигнуть светодиод **L**, сигнализируя о завершении работы загрузчика.

Если ваша плата ведёт себя иначе – скорее всего это заводской брак, если плата новая, или кривые руки – если плата уже паялась и или куда-то подключалась.

Драйвер USB контроллера

CH341

В своих проектах я использую “Ардуино-совместимые” китайские платы, у которой для подключения по USB используется контроллер **CH340/CH341**. Чтобы он распознавался компьютером, нужно установить драйвер.

Windows

Скачать драйвер можно по ссылке:

- FTP сайта
- GitHub кита
- Яндекс.Диск
- Сайт driverslab

Распаковываем архив и запускаем файл

- **SETUP.EXE** (для 32-х разрядной системы)
- DRVSETUP64/**SETUP64.EXE** (для 64-х разрядной системы).

В появившемся окошке нажимаем **INSTALL** (рисунок 7).

Если во время установки Arduino IDE вы по какой-то причине пропустили установку драйверов, то их можно установить вручную из папки с программой, расположенной по пути.

- *C/Program Files/Arduino/drivers* (для 32-х разрядной системы)
- *C/Program Files (x86)/Arduino/drivers* (для 64-х разрядной системы).

Запустить файл (рисунок 8).

- ***dpinst-x86.exe*** (для 32-х разрядной системы)
- ***dpinst-amd64.exe*** (для 64-х разрядной системы)

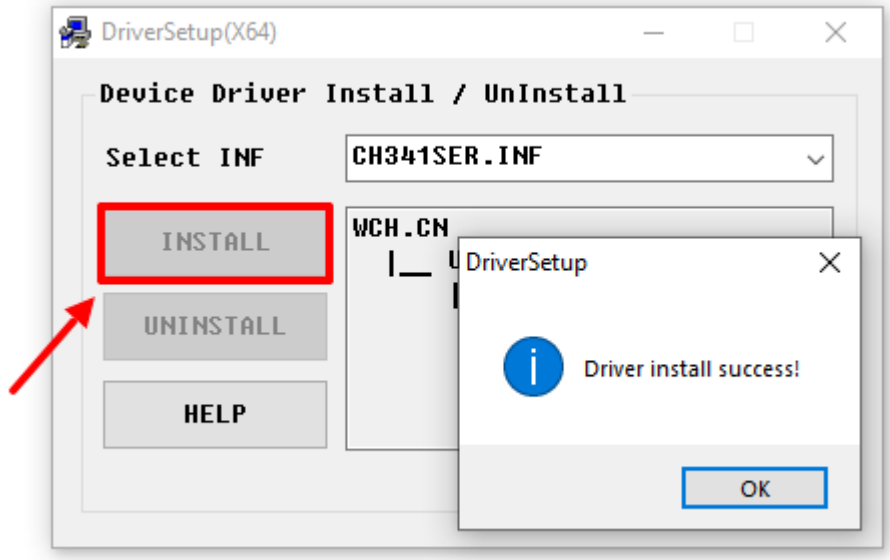

Рисунок 7 – Установка драйвера CH340/CH341

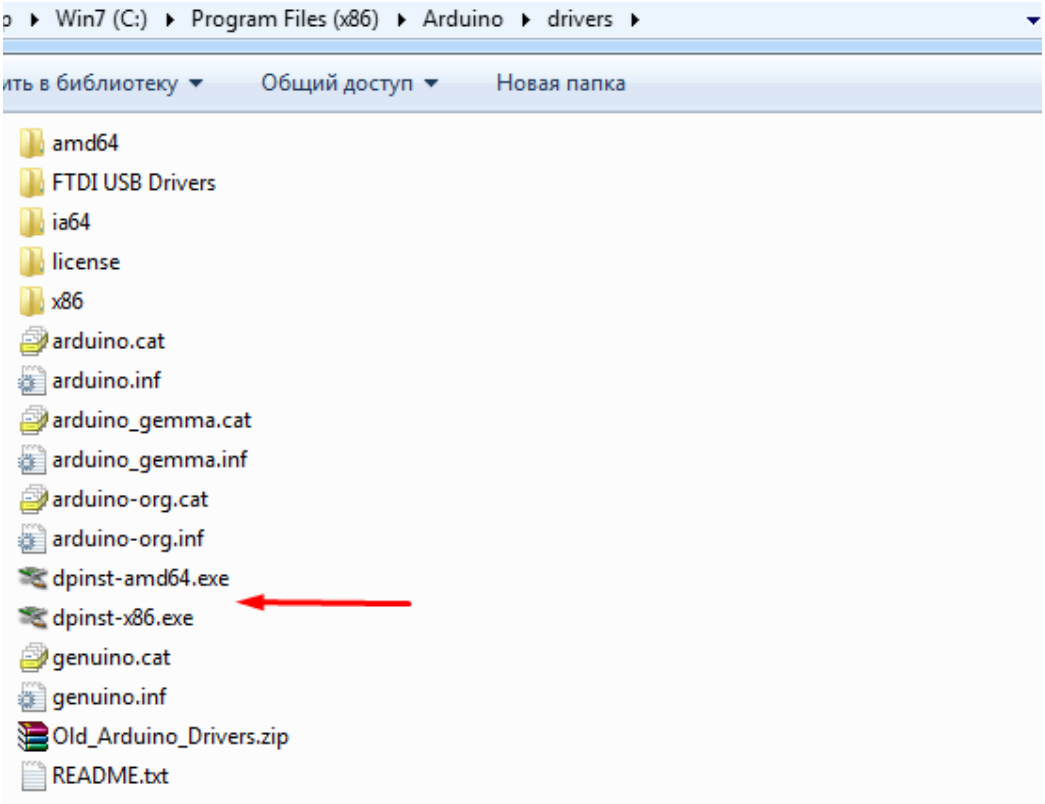

Рисунок 8 – Установка драйверов Arduino

Mac OS

Драйвер CH341 для Mac можно скачать по ссылке с моего сайта, либо со страницы источника. Если у вас будут какие-то проблемы с OSX Sierra и выше, читайте вот эту статью.

Linux Mint

В Linux уже встроен необходимый драйвер, но Arduino IDE может отказаться с ним работать: Linux определяет ардуинку как устройство ttyUSB*, обычно это ttyUSB0 (это можно узнать командой **dmesg** в терминале), то есть в системе появляется интерфейс */dev/ttyUSB0*. Чтобы с ним работать, нужны права доступа. Читать и писать на устройство */dev/ttyUSB0* имеет пользователь root и пользователи группы dialout. Работы с правами суперпользователя лучше избегать, поэтому следует занести своего пользователя в группу dialout. Это можно сделать следующей командой (обратите внимание, команда whoami в обратных кавычках)

sudo usermod -a -G dialout `whoami`

После этого нужно перелогиниться. Дальше запускаем Arduino IDE и в меню «Инструменты/Порт» ставим галочку напротив */dev/ttyUSB0*.

Linux Arch

Вся информация по работе с IDE на данной ОСи есть вот в этой статье.

FT232

На оригинальных Arduino Nano стоит USB контроллер производства FTDI – **FT232**, драйвер для всех версий ОС можно скачать с официального сайта (прямая ссылка на инсталлятор для Windows). Некоторые *очень редкие* китайцы паяют на свои Наны поддельные FTDI контроллеры, которые буквально выходят из строя после некоторых обновлений Windows. Если вам достался такой экземпляр– подробности по ситуации читайте здесь. Как восстановить контроллер и сделать рабочий драйвер – читайте здесь.

CP2102

На некоторые Arduino-совместимые платы китайцы ставят контроллер USB **CP2102**. Драйвер на него в большинстве случаев уже есть в системе (на Linux точно есть), если не работает – скачать можно с официального сайта.

- Прямая ссылка на драйвер для Windows всех версий

- Прямая ссылка на драйвер для Mac OS

Подключение платы

Плата подключается к компьютеру по USB, на ней должны замигать светодиоды. Если этого не произошло:

- Неисправен USB кабель
- Неисправен USB порт компьютера
- Неисправен USB порт Arduino
- Попробуйте другой компьютер, чтобы исключить часть проблем из списка
- Попробуйте другую плату (желательно новую), чтобы исключить часть проблем из списка
- На плате Arduino сгорел входной диод по линии USB из-за короткого замыкания, устроенного пользователем при сборке схемы
- Плата Arduino сгорела полностью из-за неправильного подключения пользователем внешнего питания или короткого замыкания

Компьютер издаст характерный сигнал подключения нового оборудования, а при первом подключении появится окошко “Установка нового оборудования”. Если этого не произошло:

- См. предыдущий список неисправностей
- Кабель должен быть data-кабелем, а не “зарядным”
- Кабель желательно втыкать напрямую в компьютер, а не через USB-хаб
- Не установлены драйверы Arduino (во время установки IDE или из папки с программой), вернитесь к установке.

В списке портов (*Arduino IDE/Инструменты/Порт*) появится новый порт, обычно **COM3**. Если этого не произошло:

- См. предыдущий список неисправностей
- Некорректно установлен драйвер на USB контроллер Arduino
    - Переверните плату и найдите “узкую” микросхему. Если на ней написано **CH341** – ставим драйвер по инструкции выше
    - Если написано **FT232R** – опять же инструкция выше
    - Если ничего не написано – открываем “Диспетчер устройств”, смотрим блок “Другие устройства”. Если при подключении платы к компьютеру там появляется ***FT232R USB UART*** – смотрим инструкцию выше

- Если список портов вообще неактивен – драйвер Arduino установлен некорректно, вернитесь к установке
- Возникла системная ошибка, обратитесь к знакомому компьютерщику или экзорцисту

Выбор и настройка платы

- Выбираем соответствующую плату в *Инструменты* \ *Плата* \ (рисунок 9).
- В микроконтроллер китайских плат зашит “старый” загрузчик, поэтому выбираем *Инструменты\Процессор\ATmega328p (Old Bootloader)*, рисунок 10. Если вам по какой-то причине пришлют платы с новым загрузчиком – прошивка не загрузится (будет минутная загрузка и ошибка), можно попробовать сменить пункт *Процессор* на ***ATmega328p***
- Теперь выбираем порт, к которому подключена плата рисунок 11. **COM1** – в большинстве случаев системный порт, у вас должен появиться ещё один (обычно COM3).

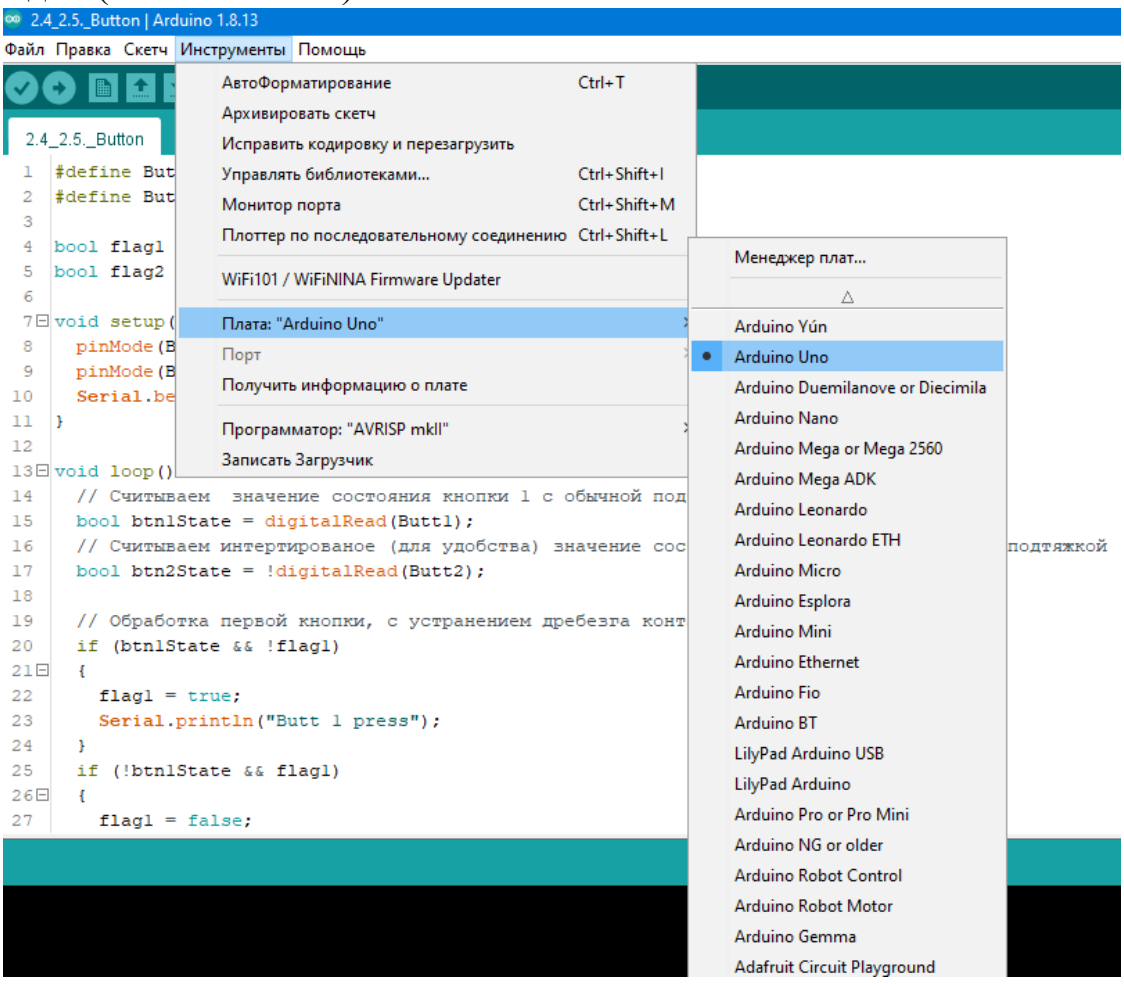

Рисунок 9 – Выбор платы Arduino

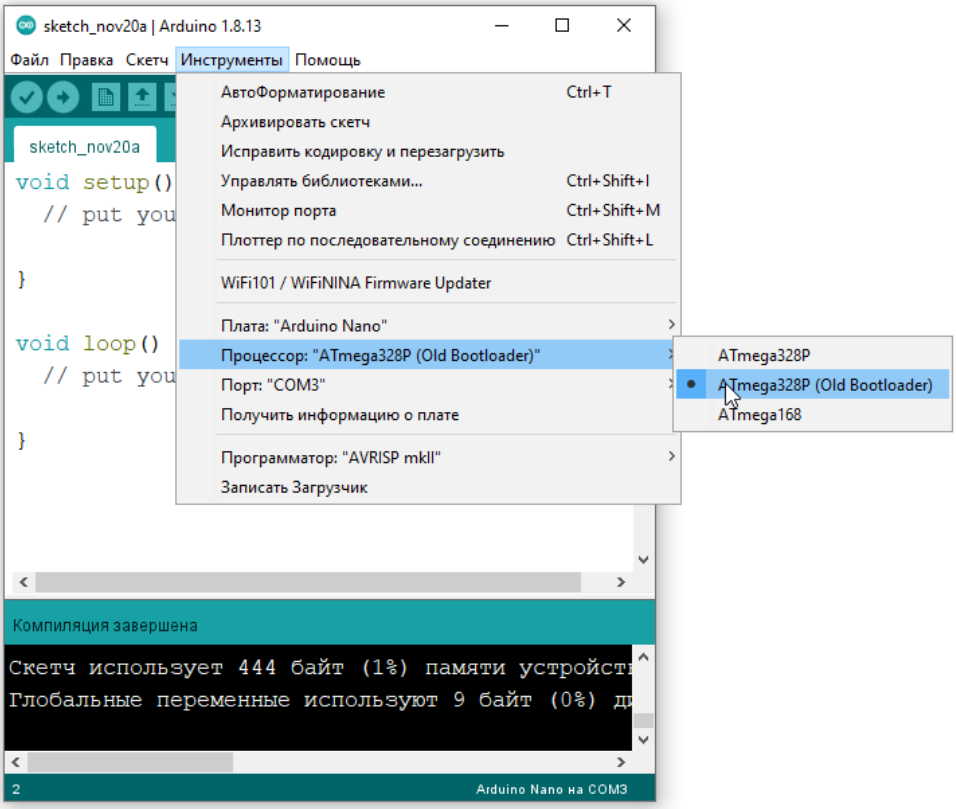

Рисунок 10 – Выбор процессора и загрузчика

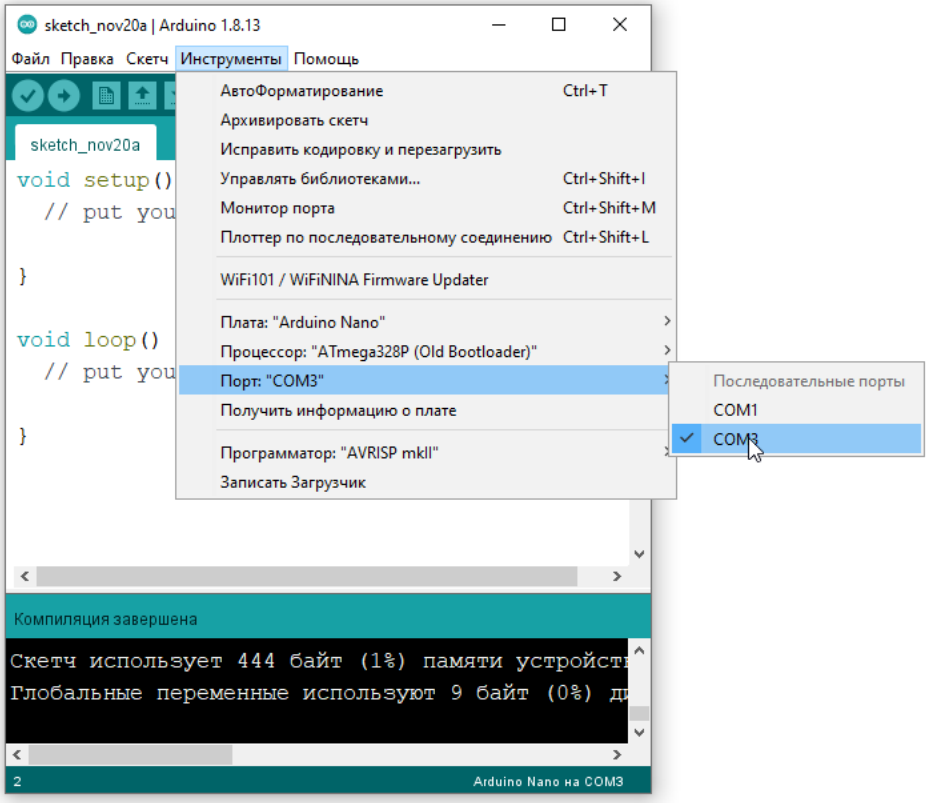

Рисунок 11 – Выбор COM-порта

Загрузка прошивки

“Загрузка” прошивки происходит в два этапа – компиляция и непосредственно загрузка в микроконтроллер. Компиляция – проверка кода на наличие ошибок, её можно запустить, нажав кнопку с символом галочки в верхнем меню программы. Компилировать код можно даже не подключая плату к компьютеру. При нажатии на кнопку с **символом стрелочки** начнётся компиляция, а затем загрузка скомпилированного кода в плату.

Вставьте следующий код с полной заменой содержимого в IDE и загрузите его. Должен начать мигать светодиод **L** на плате, это означает что все программы настроены верно и можно переходить к работе!

```
void setup() {
pinMode(LED_BUILTIN, OUTPUT);
}
void loop() {
digitalWrite(LED_BUILTIN, 0);
delay(300);
digitalWrite(LED_BUILTIN, 1);
delay(300);
}
```

Установка библиотек

Библиотека – несколько файлов с кодом, облегчающим работу с датчиками и другими модулями. К моим проектам библиотеки идут в архиве (об этом ниже). Рассмотрим все способы загрузки и установки библиотек.

Менеджер библиотек

Большинство Ардуино-библиотек можно установить автоматически из встроенного в программу менеджера библиотек (рисунок 12):

- *Скетч/Подключить библиотеку/Управлять библиотеками…*
- Комбинация клавиш **Ctrl+Shift+I**

Нужную библиотеку можно найти в поиске по названию и нажать **Установка**, библиотека будет автоматически установлена в папку с библиотеками. Arduino IDE проверяет обновления библиотек при запуске и

предложит обновиться, если найдёт обновления.

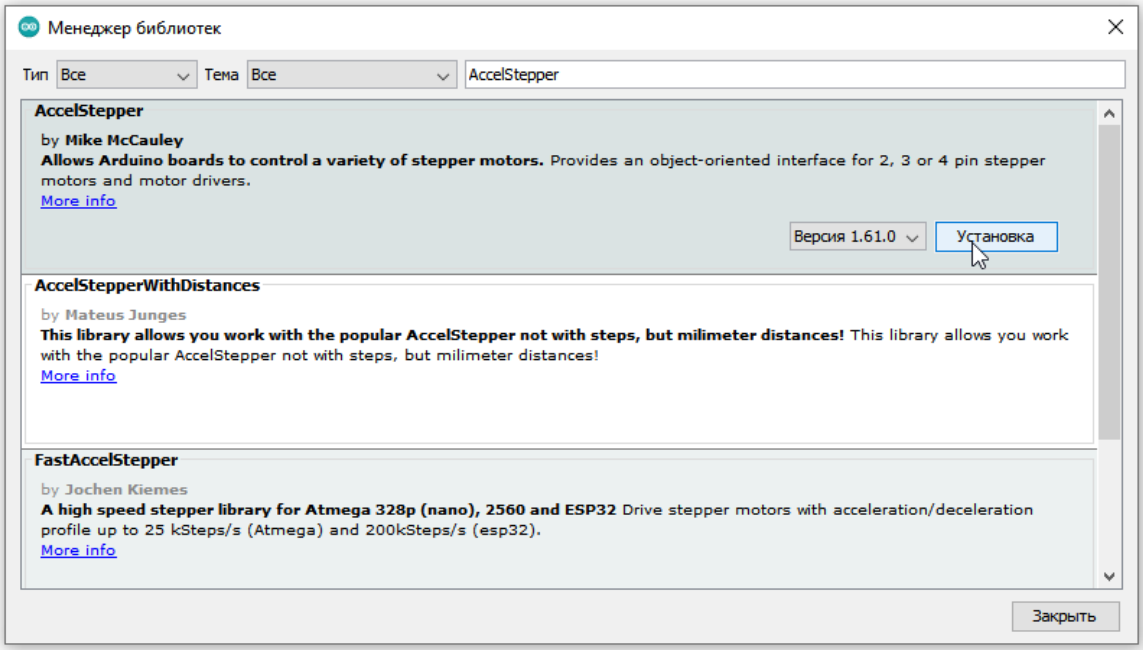

Рисунок 12 – Окно менеджера библиотек

Скачивание с GitHub

Не все существующие библиотеки есть в менеджере библиотек и скачать их можно только с GitHub (рисунок 13). Есть два способа: скачать весь репозиторий и скачать релиз. Весь репозиторий со всеми “лишними” служебными файлами можно скачать одним архивом вот так, нажав ***Code/Download ZIP*** (рисунок 13).

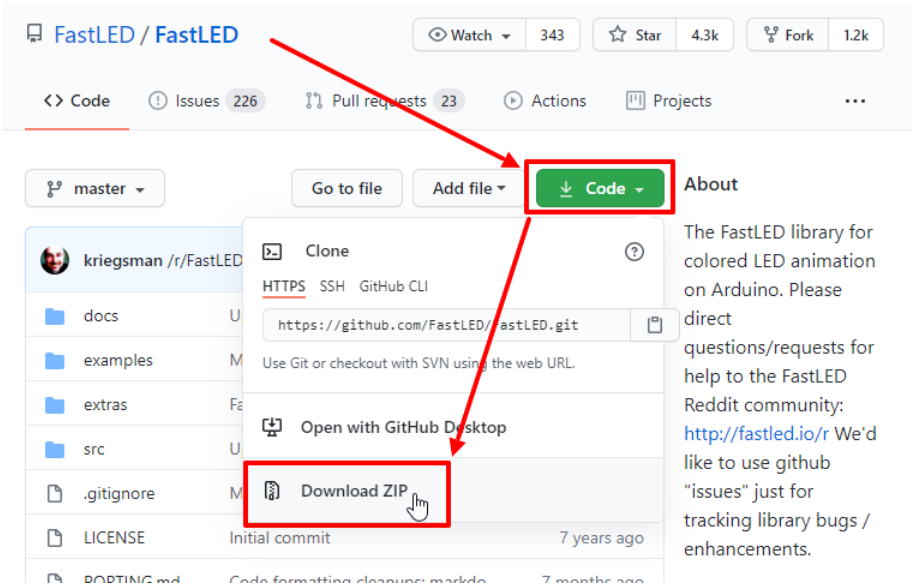

Рисунок 13 – Скачивание библиотек с GitHub

Если у библиотеки есть релизы – справа будет отмечен последний (свежий) релиз. Нажимаем на него (рисунок 14):

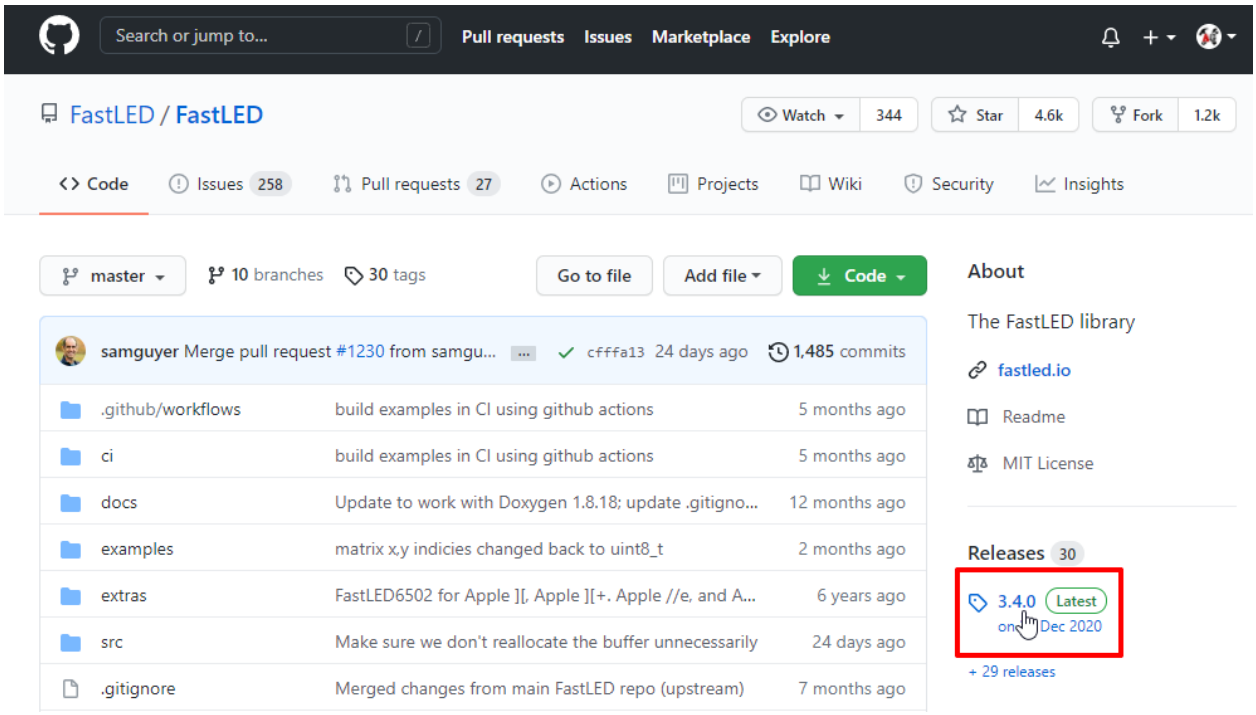

Рисунок 14 – Скачивание последнего релиза репозитория

И в новом окне нажимаем **Source code (zip)** (рисунок 15) – начнётся загрузка архива. Скачивание релиза более предпочтительно, так как содержит только файлы библиотеки.

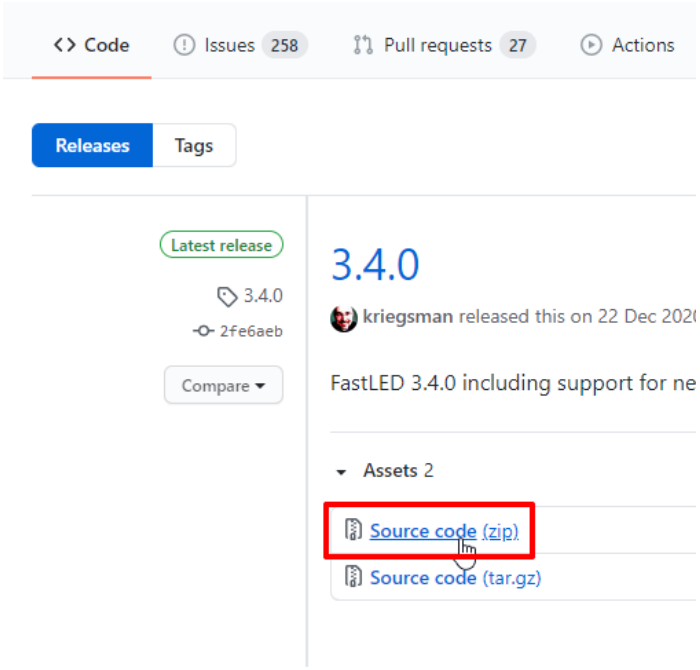

Рисунок 15 – Пояснение к скачиванию релиза репозитория

В обоих случаях библиотека скачается как .zip архив.

Автоматическая установка

Скачанный .zip архив можно установить в автоматическом режиме через *Скетч/Подключить библиотеку/Добавить .ZIP библиотеку…* В открывшемся окне выбрать скачанный архив, библиотека будет установлена по указанному в настройках пути.

Ручная установка

Для начала нужно распаковать архив (стандартный архиватор Windows или WinRAR). Чтобы Arduino IDE смогла использовать библиотеку, нам нужно положить её туда, где программа будет её искать. Таких мест три (*на примере Windows*):

- *Документы/Arduino/libraries/*
- *Папка с программой/libraries/*
    - *C/Program Files/Arduino/libraries/* (Windows 32)
    - *C/Program Files (x86)/Arduino/libraries/* (Windows 64)
    - В портативной версии IDE желательно держать библиотеки в *Папка с программой/libraries*

Рекомендуется держать все библиотеки в одном месте, чтобы не было путаницы. **Если у вас возникли с этим проблемы – устанавливайте в *Документы/Arduino/libraries/***. На рисунке 16 показана установка скачанной с GitHub библиотеки в папку с программой и в документы.

“Голые” МК

Для начала рекомендуется изучить вот эти два урока: первый и второй. У проектов на базе голого микроконтроллера есть два варианта:

- Если проект основан на ATmega328 (Arduino Nano/Mini) и на плате есть источник тактирования на 16 МГц (резонатор), то микроконтроллер можно просто перепаять с Arduino и загружать прошивку через внешний USB-TTL переходник, как на Arduino Pro Mini. Либо загрузить прошивку, и потом перепаивать – всё будет работать.

- Если источника тактирования нет – так делать нельзя! Сначала нужно настроить МК на внутреннее тактирование, подключив ISP программатор к плате Arduino и выбрав внутренний источник тактирования в настройках ядра.

- Если используется новый микроконтроллер (или припаянный китайцами) – он по умолчанию настроен на внутреннее тактирование и его можно паять на плату в любом случае. Загрузить прошивку можно только при помощи ISP программатора. Также можно прошить загрузчик и в дальнейшем загружать прошивку через USB-TTL преобразователь.

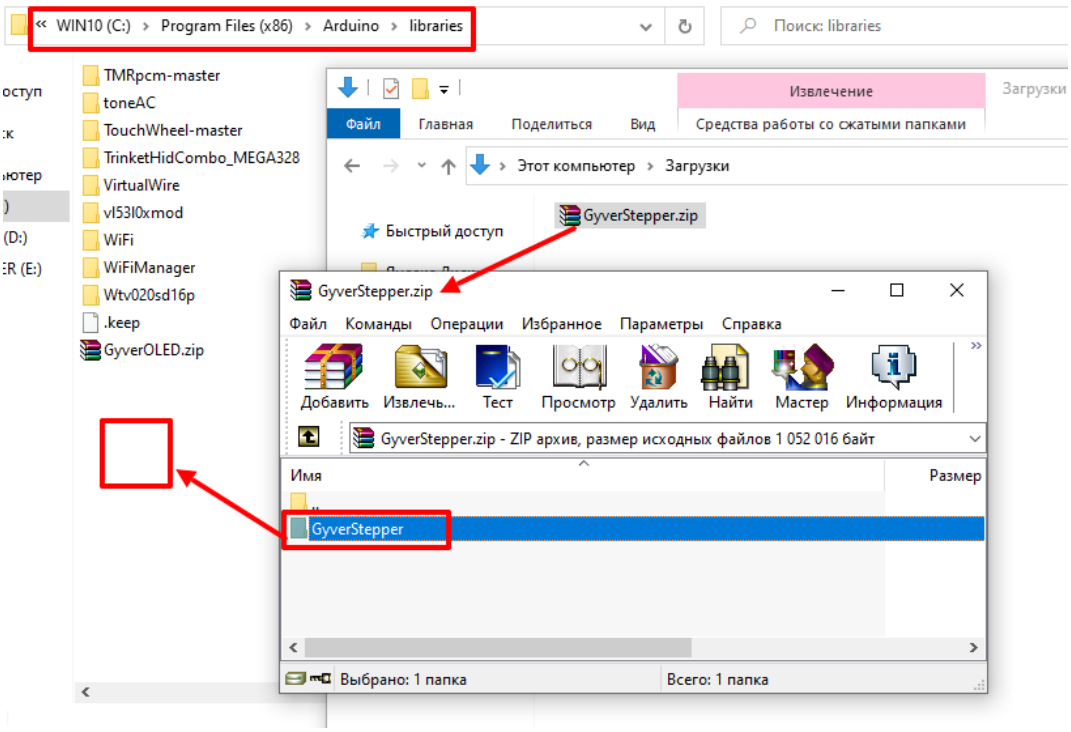

Рисунок 16 – Распаковка библиотеки

ESP8266

ESP8266 – микроконтроллер с WiFi на борту, на его базе сделаны платы Wemos D1 mini, NodeMCU и другие. Китайские платы Wemos и NodeMCU подключаются к компьютеру по USB при помощи бортового USB-TTL преобразователя, причём китайцы паяют **CH340** или **CP2102** (функционально не имеют отличий), обычно это даже указано на странице товара. Установка драйверов разобрана выше на этой странице.

Также для работы с esp нужно добавить поддержку плат в Arduino IDE:

- Запустить Arduino IDE, перейти в *Файл/Настройки/*

- В окошко “Дополнительные ссылки…” Вставить **http://arduino.esp8266.com/stable/package_esp8266com_index.json**
- Нажать ОК
- Перейти в *Инструменты/Плата/Менеджер плат…* Начать вводить в поиске “ESP”. Выбрать и установить **ESP8266 boards**
- Теперь в списке плат *Инструменты/Плата/* появится семейство плат на esp8266! Выбираем соответствующую своей плате конфигурацию.
- Выбираем порт, к которому подключена плата.

Заметка для **NodeMCU**. Перед началом загрузки нужно ввести плату в режим прошивки. Подключить к компьютеру, выбрать появившийся порт для загрузки. Зажать кнопку *Flash*. Кликнуть по кнопке *Reset*. Отпустить кнопку *Flash*. И только после этого нажать стрелочку в программе для загрузки прошивки.

Digispark

Digispark – плата на базе ATtiny85, загрузка в которую может производиться через бортовой USB. Для работы с Digispark нужно добавить поддержку плат в Arduino IDE:

- Запустить Arduino IDE, перейти в *Файл/Настройки/*
- В окошко “Дополнительные ссылки…” Вставить
  - **http://digistump.com/package_digistump_index.json**
  - или **https://raw.githubusercontent.com/digistump/arduino-boards-index/master/package_digistump_index.json**
- Нажать ОК
- Перейти в *Инструменты/Плата/Менеджер плат…* Начать вводить в поиске “Digispark”. Выбрать и установить **Digistump AVR Boards**
- Теперь в списке плат *Инструменты/Плата/* появится семейство плат Digispark! Выбираем первую **Digispark (Default – 16.5mhz)**
- Также нужно установить драйвера, скачать можно на официальном GitHub проекта (в разделе *Релизы*, вот прямая ссылка на архив), либо с моего FTP. Драйвера есть для Win, MacOS и Linux.
- Пользователям Linux читать здесь

- Прошивка загружается следующим образом: **ПЛАТУ НЕ ПОДКЛЮЧАЕМ, ПОРТ НЕ ВЫБИРАЕМ**, нажимаем ***загрузка***, ждём компиляции. Появится надпись *“подключите плату”.* Втыкаем плату в USB и прошивка загружается. Почему так? Дигги имеет на борту свой USB интерфейс, поэтому работает напрямую.

Ошибки компиляции

Возникает на этапе сборки и компиляции прошивки. Ошибки компиляции вызваны проблемами **в коде прошивки**, то есть проблема сугубо программная. Слева от кнопки “загрузить” есть кнопка с галочкой – проверка. Во время проверки производится компиляция прошивки и выявляются ошибки, если таковые имеются. Ардуино в этом случае может быть **вообще не подключена к компьютеру**.

- В некоторых случаях ошибка возникает при наличии **кириллицы** (русских букв) в пути к папке со скетчем. Решение: завести для скетчей отдельную папочку в корне диска с **английским названием**.

- В чёрном окошке в самом низу Arduino IDE можно прочитать **полный текст ошибки** и понять, куда копать

- В скачанных с интернета готовых скетчах часто возникает ошибка с описанием <название файла>.h no such file or directory. Это означает, что в скетче используется библиотека **<название файла>**, и нужно положить её в *Program Files/Arduino/libraries/*. Ко всем моим проектам всегда идёт папочка с использованными библиотеками, которые нужно установить. Также библиотеки всегда можно поискать в гугле по **<название файла>.**

- При использовании каких-то особых библиотек, методов или функций, ошибкой может стать неправильно выбранная плата в “*Инструменты/плата*“. **Пример**: прошивки с библиотекой *Mouse.h* или *Keyboard.h* компилируются только для **Leonardo** и **Micro**.

- Если прошивку пишете вы, то любые синтаксические ошибки в коде будут подсвечены, а снизу в чёрном окошке можно прочитать более детальное описание, в чём собственно косяк. Обычно указывается строка, в которой сделана ошибка, также эта строка подсвечивается красным.

- Иногда причиной ошибки бывает **слишком старая**, или **слишком новая** версия Arduino IDE. Читайте комментарии разработчика скетча.

• Ошибка **недостаточно свободного места** возникает по вполне понятным причинам. Оптимизация: статическая память – память, занимаемая кодом (циклы, функции). Динамическая память занята переменными.

Частые ошибки в коде, приводящие к ошибке компиляции

- **…no such file or directory** – компилятор не может найти файл, который используется в коде. Чаще всего это библиотека, которую не установили или установили неправильно
- **expected ‘,’ or ‘;’** – пропущена запятая или точка запятой на предыдущей строке
- **stray ‘\320’ in program** – русские символы в коде
- **expected unqualified-id before numeric constant** – имя переменной не может начинаться с цифры
- **… was not declared in this scope** – переменная или функция используется, но не объявлена. Компилятор не может её найти
- **redefinition of …** – повторное объявление функции или переменной
- **storage size of … isn’t known** – массив задан без указания размера

Ошибки загрузки

Возникает на этапе, когда прошивка собрана, скомпилирована, в ней нет критических ошибок, и производится загрузка в плату по кабелю. Ошибка может возникать как по причине неисправностей железа, так и из-за настроек программы и драйверов.

• Если неправильно выбран COM порт – прошивка не загрузится с ошибкой *avrdude: ser_open(): can’t open device*. Вернитесь к пункту “Выбор и настройка платы” этого урока и убедитесь в том, что выбор порта активен и при подключении платы появляется новый.

• Большинство проблем при загрузке, вызванных “зависанием” ардуины или загрузчика, лечатся **полным отключением Ардуины от питания**. Потом вставляем USB и по новой прошиваем.

• Причиной ошибки загрузки может быть неправильно выбранная плата в “*Инструменты/Плата”*, а также неправильно выбранный процессор в “*Инструменты/Процессор”*.

• Если это Arduino Nano – попробуйте оба, *Old* и не *Old*.

- Если у вас открыт **монитор COM порта в другом окне Arduino IDE** или плата **общается через COM порт с другой программой** (Ambibox, HWmonitor, SerialPortPlotter и т.д.), то вы получите ошибку загрузки, потому что порт занят. Отключитесь от порта или закройте другие окна и программы.

- Если у вас задействованы пины RX или TX – **отключите от них всё**! По этим пинам Arduino общается с компьютером, в том числе для загрузки прошивки.

- Если в описании ошибки встречается *bootloader is not responding* и *not in sync*, из-за ошибок записи мог "слететь" загрузчик, его можно попробовать прошить заново.

- Если все пункты из этого списка проверены, а загрузчик прошить не удаётся – микроконтроллер скорее всего необратимо повреждён, то есть сгорел.

Предупреждения

Помимо ошибок, по причине которых проект вообще не загрузится в плату и не будет работать, есть ещё предупреждения, которые выводятся оранжевым текстом в чёрной области лога ошибок. Предупреждения могут появиться даже тогда, когда выше лога ошибок появилась надпись "**Загрузка завершена**". Это означает, что в прошивке нет несовместимых с жизнью ошибок, она скомпилировалась и загрузилась в плату. Что же тогда означают предупреждения? Чаще всего можно увидеть такие:

- **# Pragma message…….** – сообщения с директивой Pragma обычно выводят библиотеки, сообщая о своей версии или каких-то настройках. *Это даже не предупреждение, а просто вывод текста в лог*.

- **Недостаточно памяти, программа может работать нестабильно** – чуть выше этого предупреждения обычно идёт информация о задействованной памяти. **Память устройства** можно добивать до 99%, ничего страшного не случится. Это флэш память и во время работы она не изменяется. А вот **динамическую память** желательно забивать не более 85-90%, иначе реально могут быть непонятные глюки в работе, так как память постоянно "бурлит" во время работы. **НО**. Это зависит от скетча и в первую очередь от количества локальных переменных. Можно написать такой код, который будет стабильно работать при 99% занятой SRAM памяти.

- Предупреждения о несовместимых типах данных. Компилятор не смог привести один тип к другому и сообщает о потенциальных ошибках в ходе выполнения программы. В большинстве случаев ничего плохого не случится, но лучше найти проблемную строку и помочь компилятору преобразовать тип.

Частые вопросы

- **Ардуину можно прошить только один раз?** Нет, несколько десятков тысяч раз, всё упирается в ресурс flash памяти. А он довольно большой.

- **Как стереть/нужно ли стирать старую прошивку при загрузке новой?** Память автоматически очищается при прошивке, старая прошивка автоматически удаляется.

- **Можно ли записать две прошивки, чтобы они работали вместе?** Нет, при прошивке удаляются абсолютно все старые данные. Из двух прошивок нужно сделать одну, причём так, чтобы не было конфликтов.

- **Можно ли "вытащить" прошивку с уже прошитой Ардуины?** Теоретически можно, но только в виде нечитаемого машинного кода, в который преобразуется прошивка на C++ при компиляции, т.е. вам это НИКАК не поможет, если вы не имеете диплом по низкоуровневому программированию.

- Зачем это нужно? Например есть у нас прошитый девайс, и мы хотим его "клонировать". В этом случае да, есть вариант сделать дамп прошивки и загрузить его в другую плату **на таком же микроконтроллере**.

- Если есть желание почитать код – увы, прошивка считывается в виде бинарного машинного кода, превратить который обратно в читаемый Си-подобный код обычному человеку не под силу

- Вытащить прошивку, выражаясь более научно – сделать дамп прошивки, можно при помощи ISP программатора, об этом можно почитать здесь

- Снять дамп прошивки можно только в том случае, если разработчик не ограничил такую возможность, например записав **лок-биты**, запрещающие считывание Flash памяти, или вообще **отключив SPI** шину. Если же разработчик – вы, и есть желание максимально защитить своё устройство от копирования – гуглите про лок-биты и отключение SPI