

## Работа со средой Arduino IDE

Среда разработки Arduino IDE состоит из следующих компонентов (рисунок 1):

- Текстовый редактор кода программы (1);
- Область сообщений (2);
- Консоль (3);
- Панель инструментов (4);
- Панель с часто используемыми командами (5).

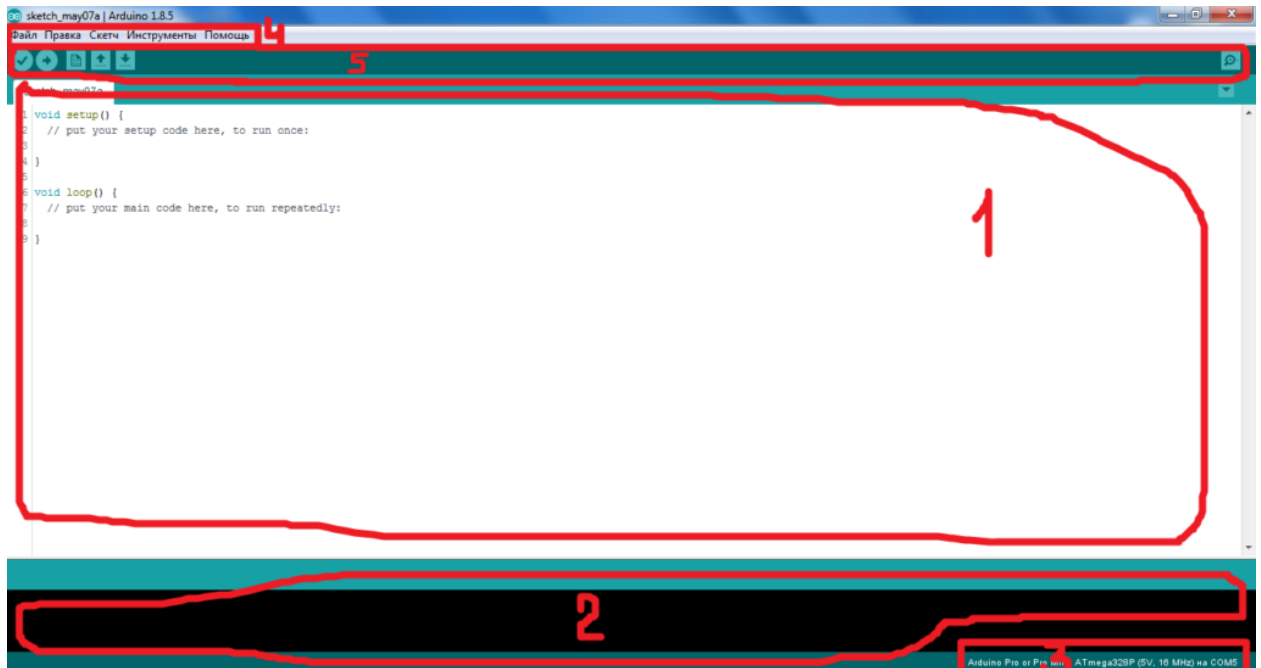


Рисунок 1 – Среда разработки Arduino IDE

Написанная в среде ArduinoIDE программа называется «скетч». Сама программа пишется в текстовом редакторе. В области сообщений выводятся ошибки и пояснения. В консоли отображается полный отчёт о компиляции программы, различные ошибки, подсказки и другая полезная информация. Панель инструментов представляет собой набор полезных функций при разработке программы. На панели с часто используемыми командами всего несколько кнопок, но они являются самыми основными при разработке программы.

### ***Язык программирования Arduino:***

Фактически нет особого языка программирования Arduino. Модули Arduino программируются на C/C++. Особенности программирования сводятся к тому, что существует набор библиотек, включающий некоторые функции (pinMode, digitalWrite и т.д.) и объекты (Serial), которые значительно облегчают процесс написания программы.

## Настройка среды Arduino IDE:

Открыть настройки среды (рисунок 2) можно из пункта меню **Файл -> Настройки** или с помощью комбинации клавиш **Ctrl + ,**

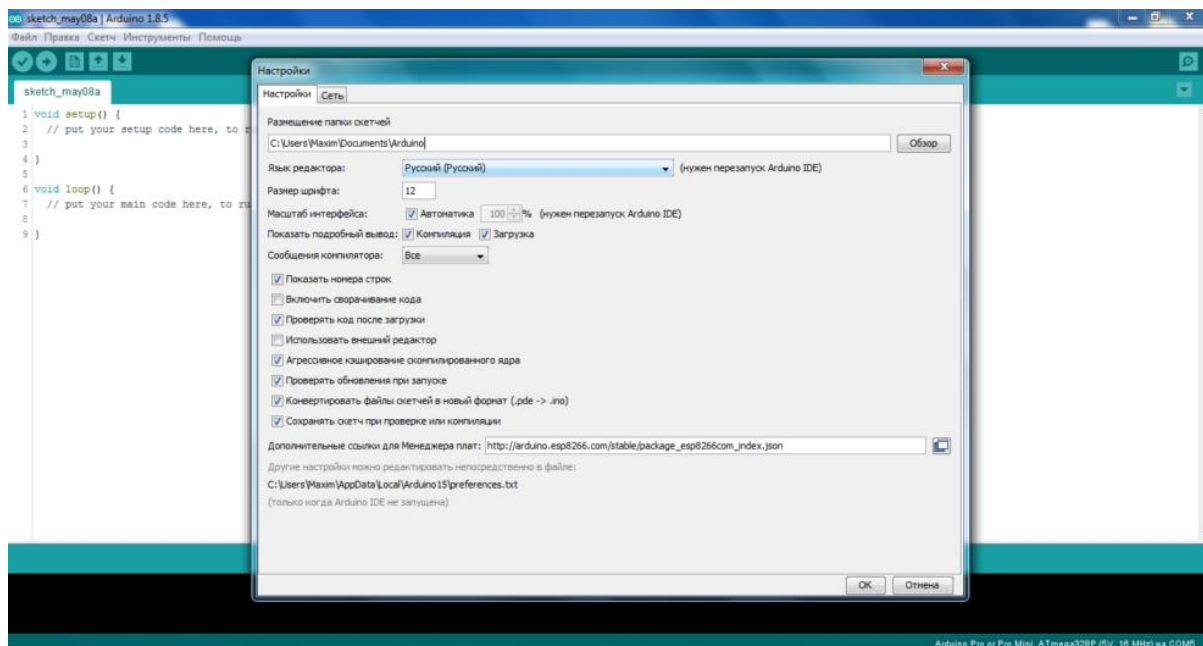


Рисунок 2 – Окно настройки среды

В меню настроек можно изменить путь сохранения программ по умолчанию, изменить размер шрифтов и т.д.

## Различные примеры:

Одной из особенностей Arduino IDE является довольно обширная база различных примеров, что очень удобно для начинающих. Открыть пример можно из пункта меню **Файл -> Примеры** (рисунок 3).

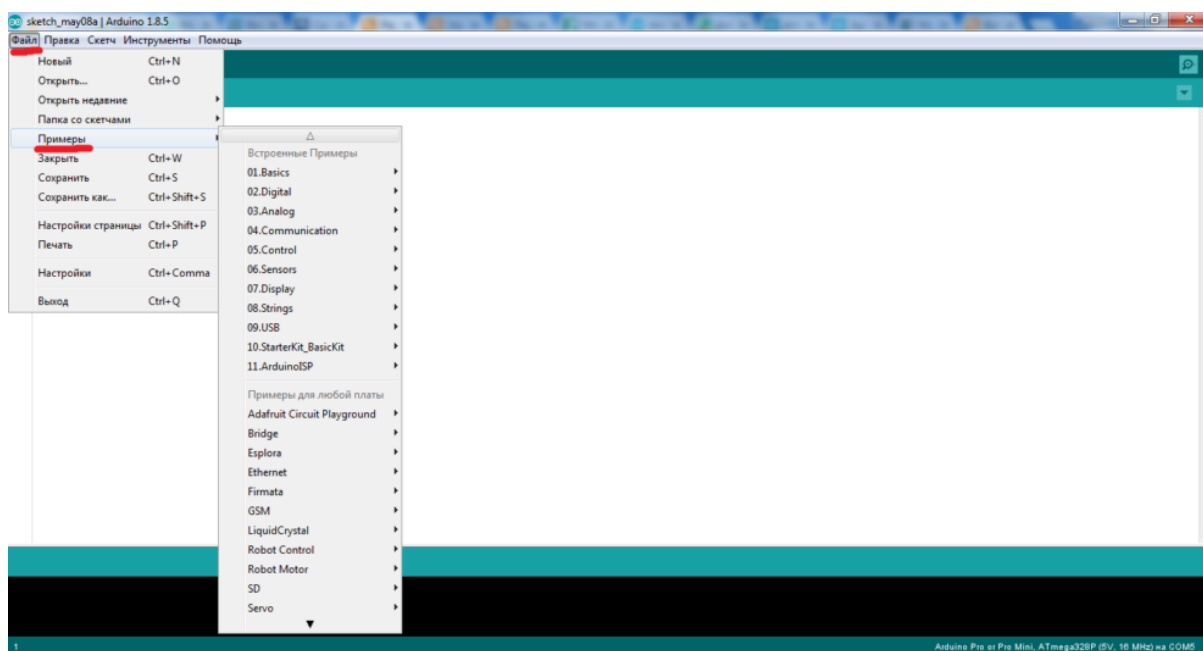


Рисунок 3 – Открытие базы примеров

### ***Создание новой программы:***

Итак, с настройками среды мы разобрались. Теперь можно создать новую программу. Это можно сделать несколькими способами:

- из пункта меню **Файл -> Новый** (рисунок 4);

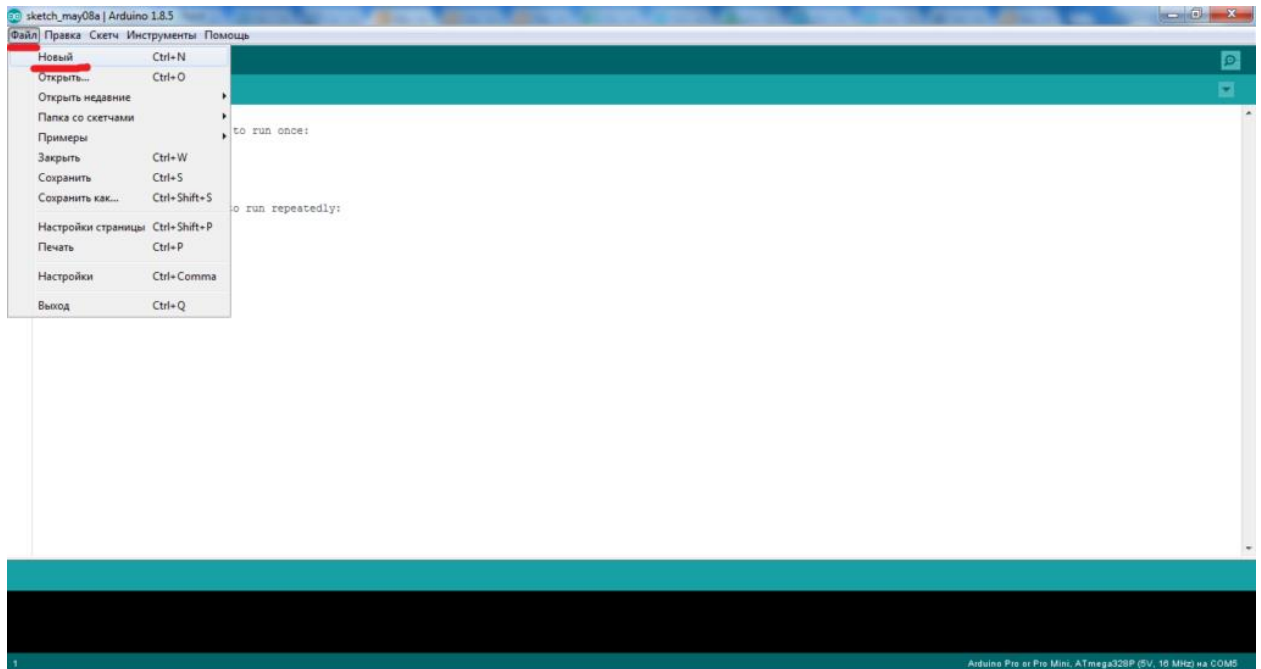


Рисунок 4 – Создание нового файла

- при помощи комбинации клавиш **Ctrl + N**;
- из панели с часто используемыми командами (рисунок 5).



Рисунок 5 – Создание нового файла

### ***Сохранение программы:***

Допустим мы написали программу. Далее нужно сохранить. Сделать это можно несколькими способами:

- из пункта меню **Файл -> Сохранить** (рисунок 6);

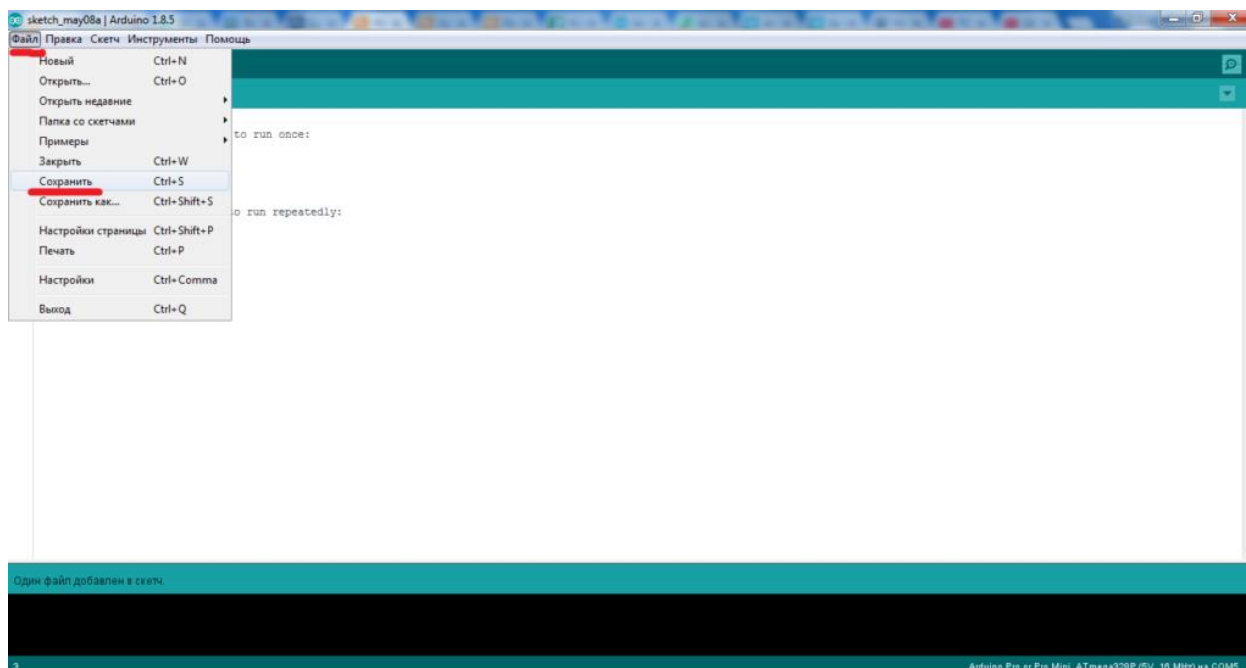


Рисунок 6 – Сохранение файла

- при помощи комбинации клавиш **Ctrl + S**;
- из панели с часто используемыми командами (рисунок 7)



Рисунок 7 – Сохранение скетча

Теперь нужно только ввести имя вашей программы (*оно не должно содержать русских символов!*) и выбрать место, куда её сохранить. Сохранённая программа автоматически помещается в одноимённую папку, которая создаёт сама Arduino IDE.

### **Открытие программы:**

Теперь мы научились создавать и сохранять программу. Но как теперь открыть сохранённую программу? Для этого в Arduino IDE предусмотрено несколько способов:

- из пункта меню **Файл -> Открыть** (рисунок 8);

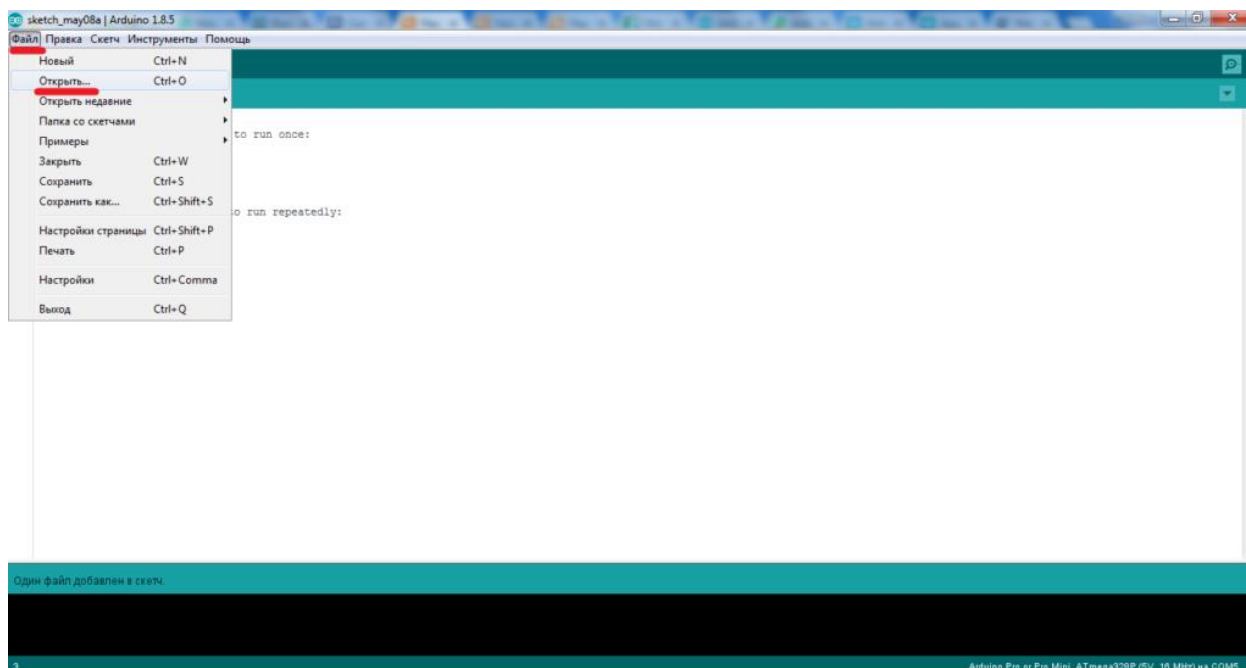


Рисунок 8 – Открытие файла

- при помощи комбинации клавиш **Ctrl + O**;
- из панели с часто используемыми командами (рисунок 9).



Рисунок 9 – Открытие скетча

В открывшемся окошке нужно выбрать папку, в которой находится нужная программа.

### ***Редактирование текста программы:***

Удобный редактор текста программы очень важен при разработке какой-либо программы. В Arduino IDE он довольно неплох, однако уступает конкурентам, таким как Eclipse, Visual Studio и т.д. Однако, его вполне достаточно. В редакторе присутствуют все основные команды, необходимые при редактировании кода. Они находятся в меню **Правка**. Для самых часто используемых команд (копировать, ставить и т.д.) существуют комбинации, способствующие быстрому доступу к нужной команде правки, что очень удобно. Другими отличительными особенностями встроенного редактора кода являются возможность копирования кода для форумов и в html формате, что позволяет делиться Вашими программами, сохраняя наглядность разметки в виде BB кодов или html разметки соответственно.

Давайте остановимся на основных командах, необходимых для редактирования программы (рисунок 10):

- Копировать. Пункт меню **Правка** -> **Копировать** или комбинация клавиш **Ctrl + C**;

- Вставить. Пункт меню *Правка* -> *Вставить* или комбинация клавиш **Ctrl + V**;
- Вырезать. Пункт меню *Правка* -> *Вырезать* или комбинация клавиш **Ctrl + X**;
- Выделить всё. Пункт меню *Правка* -> *Выделить всё* или комбинация клавиш **Ctrl + A**;
- Найти. Пункт меню *Правка* -> *Найти* или комбинация клавиш **Ctrl + F**;
- Отменить. Пункт меню *Правка* -> *Отменить* или комбинация клавиш **Ctrl + Z**;
- Вернуть. Пункт меню *Правка* -> *Вернуть* или комбинация клавиш **Ctrl + Y**;
- Форматировать текст программы. Пункт меню *Инструменты* -> *АвтоФорматирование* (рисунок 11) или комбинация клавиш **Ctrl + T**. Эта команда позволяет исправить неточности в разметке программы и привести её в более читабельный вид.

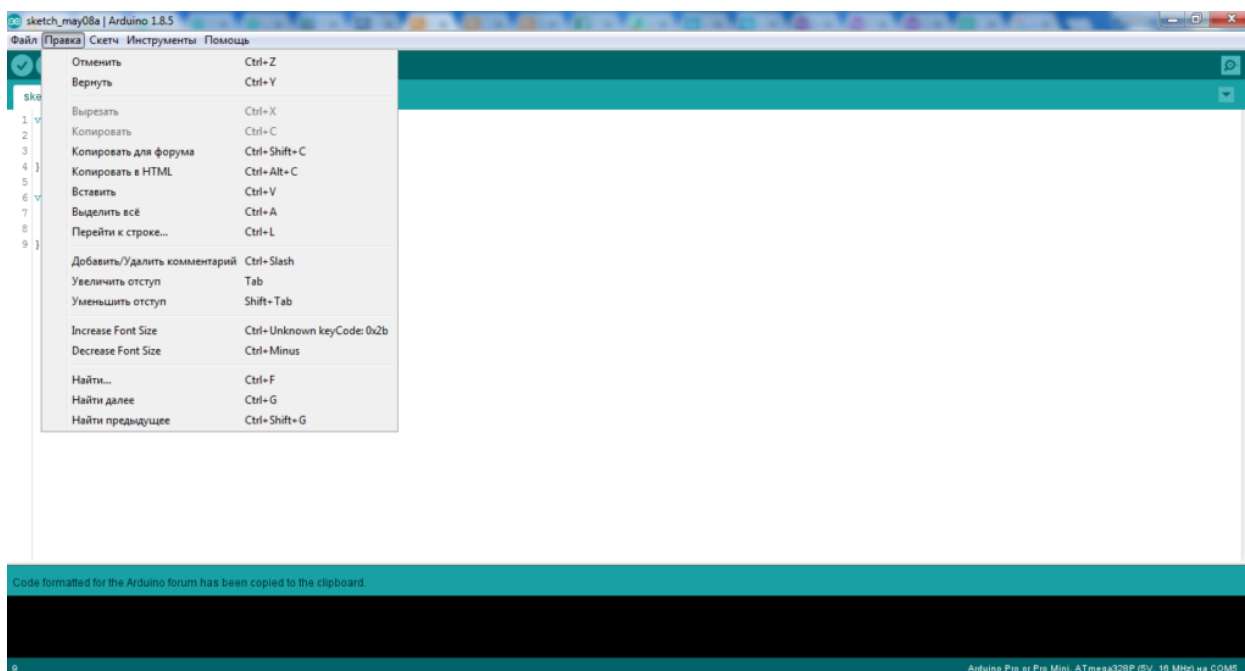


Рисунок 10 – Окно «правка»

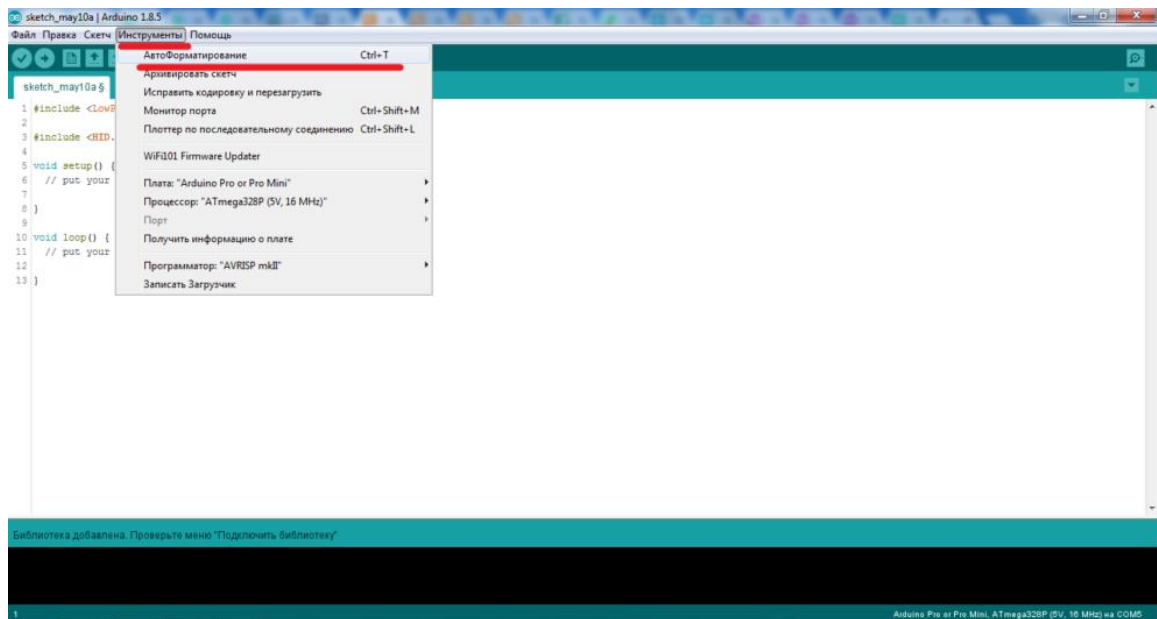


Рисунок 11 – Автоформатирование кода

### **Подключение библиотеки:**

Давайте разберёмся, что же такое библиотека. **Библиотека** — это набор функций, предназначенных для того, чтобы максимально упростить работу с различными модулями, датчиками и т.д. Например, библиотека *LowPower* позволяет легко управлять режимами энергосбережения модулей Arduino. Существует огромное количество различных модулей и датчиков. Но как ими управлять? Для этого разработаны специальные библиотеки, которые значительно облегчают работу. Но перед тем, как использовать дополнительные библиотеки, необходимо установить и подключить их. А как подключить библиотеку, спросите Вы? Есть несколько способов: через среду Arduino IDE. Для этого перейдите в меню **Скетч Подключить библиотеку** (рисунок 12).

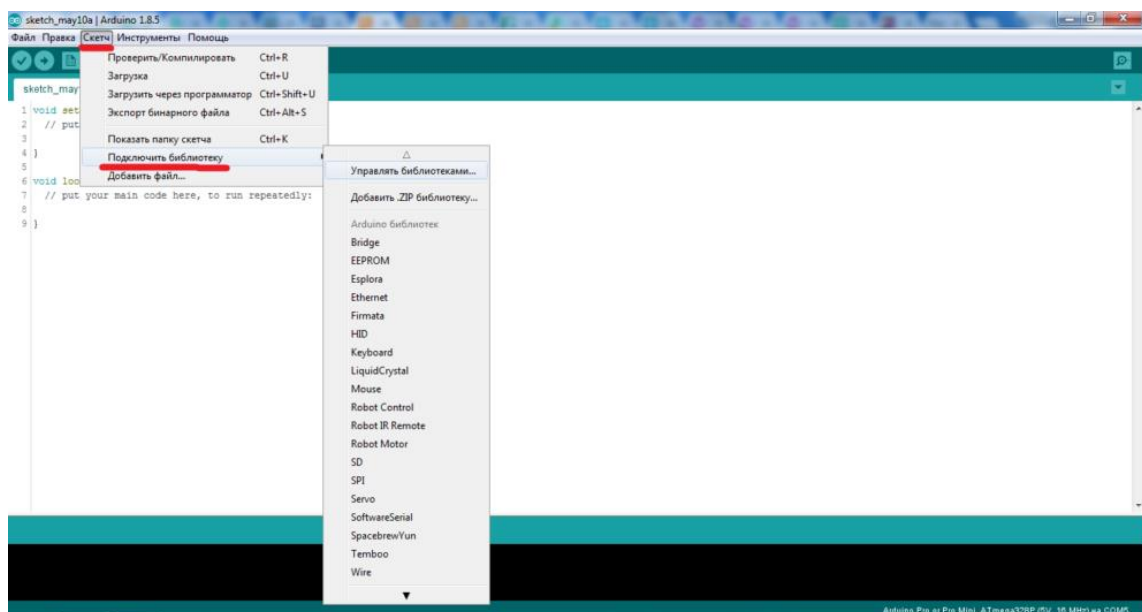


Рисунок 12 – Подключение библиотек

Если нужная библиотека есть в списке, то необходимо просто кликнуть на неё мышкой и она автоматически подключится к Вашей программе. А что же делать, если нужной библиотеки нет в списке? В таком случае перейдите во вкладку **Скетч** -> **Подключить библиотеку** -> **Управление библиотеками**. Перед Вами откроется следующее окошко (рисунок 13):

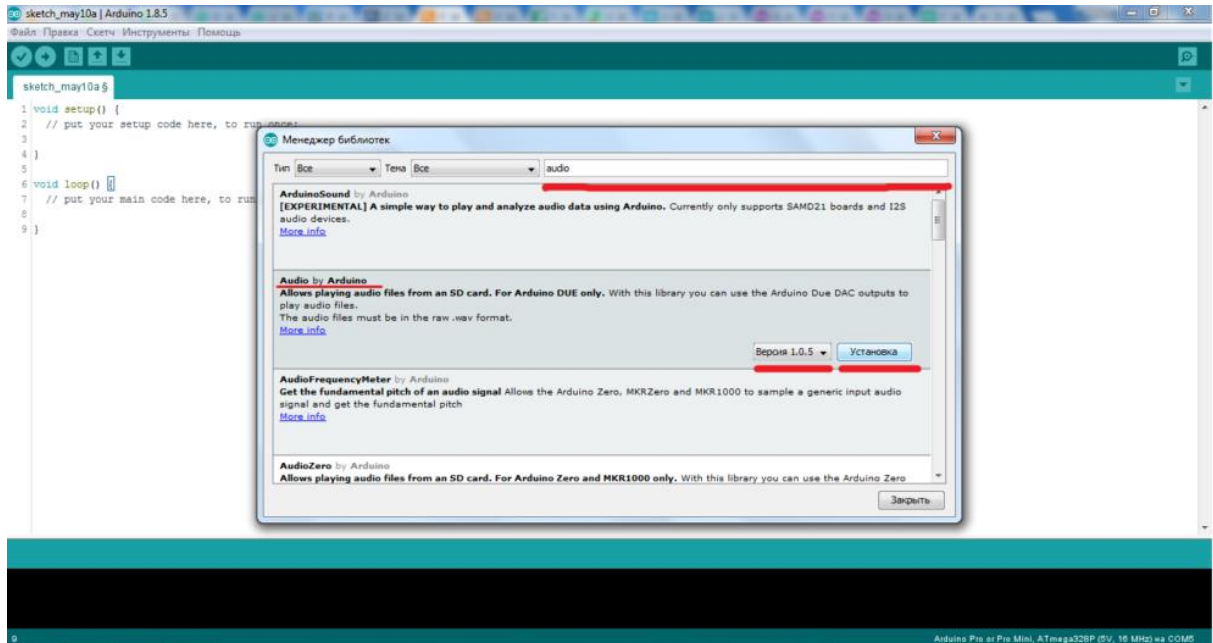


Рисунок 13 – Окно управления библиотеками

В правом верхнем углу необходимо ввести название нужной библиотеки. Далее, из списка надо выбрать нужную библиотеку, её версию и нажать кнопку **установить**. Обратите внимание, что есть возможность отсортировать результаты поиска (вкладки **Тип** и **Тема**). Теперь она появится в списке установленных библиотек и её можно подключить через меню **Скетч** -> **Подключить библиотеку**.

-Предварительно скачав нужную библиотеку её также можно подключить к своей программе. Если она находится в zip архиве, то её можно подключить следующим способом. Для этого нужно перейти в меню **Скетч** -> **Подключить библиотеку** -> **Подключить .ZIP библиотеку** и в открывшемся окошке выбрать zip архив с библиотекой. Теперь эта библиотека появится в списке установленных и её можно подключить через меню **Скетч** -> **Подключить библиотеку**. Если у Вас открыта Arduino IDE, то её нужно перезапустить, чтобы изменения были применены.

-Подключить библиотеку можно и без использования Arduino IDE. Для этого необходимо её скачать и скопировать папку с библиотекой по следующему пути: **X:\Пользователи\<Имяпользователя>\Документы\Arduino\libraries** (рисунок 14).



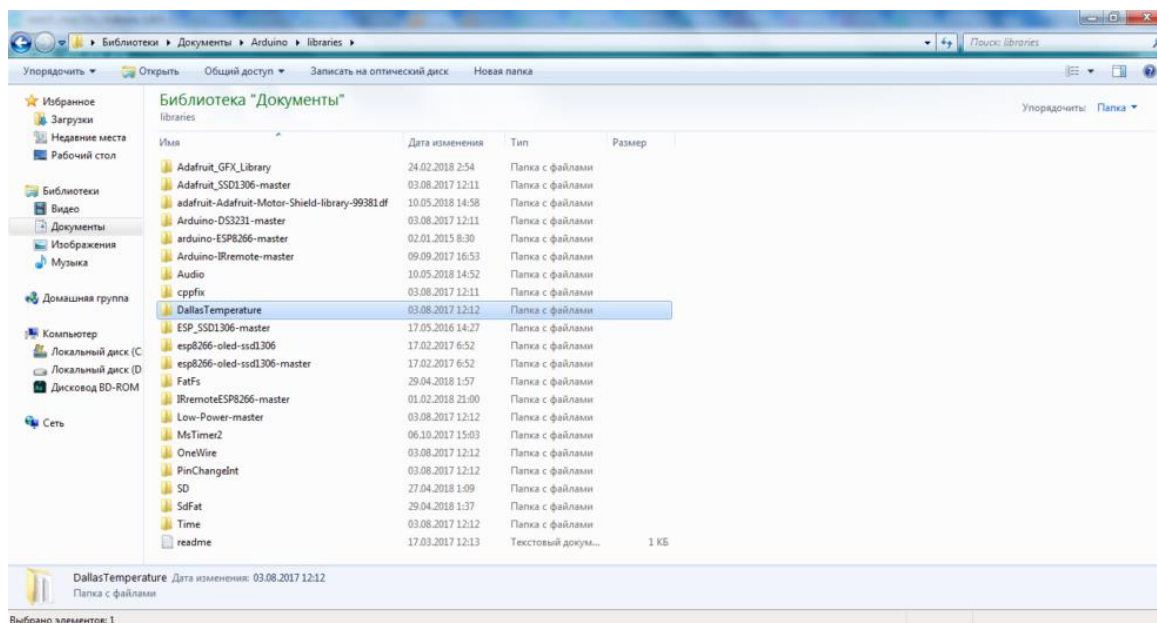


Рисунок 14 – Добавление библиотек

Если у Вас открыта Arduino IDE, то её нужно перезапустить, чтобы изменения были применены. Теперь эта библиотека появится в списке установленных и её можно подключить через меню **Скетч -> Подключить библиотеку**. Однако можно и вручную в программе подключить библиотеку. Для этого перед функцией [setup\(\)](#) нужно прописать следующую конструкцию `#include <имя заголовочного файла.h>` или `#include «имя заголовочного файла.h»`.

В каждой библиотеке есть различные примеры использования функционала библиотеки. Имя заголовочного файла можно найти там. Дорогой читатель, рекомендуем внимательно изучать примеры, ведь любая, даже большая программа, состоит из кусочков простых примеров.

### **Выбор платы:**

Существует довольно большое количество модулей Arduino. Загружать написанную программу нужно именно в тот тип модуля, который подключён к компьютеру. Выбрать модуль можно в меню **Инструменты -> Плата** (рисунок 15).

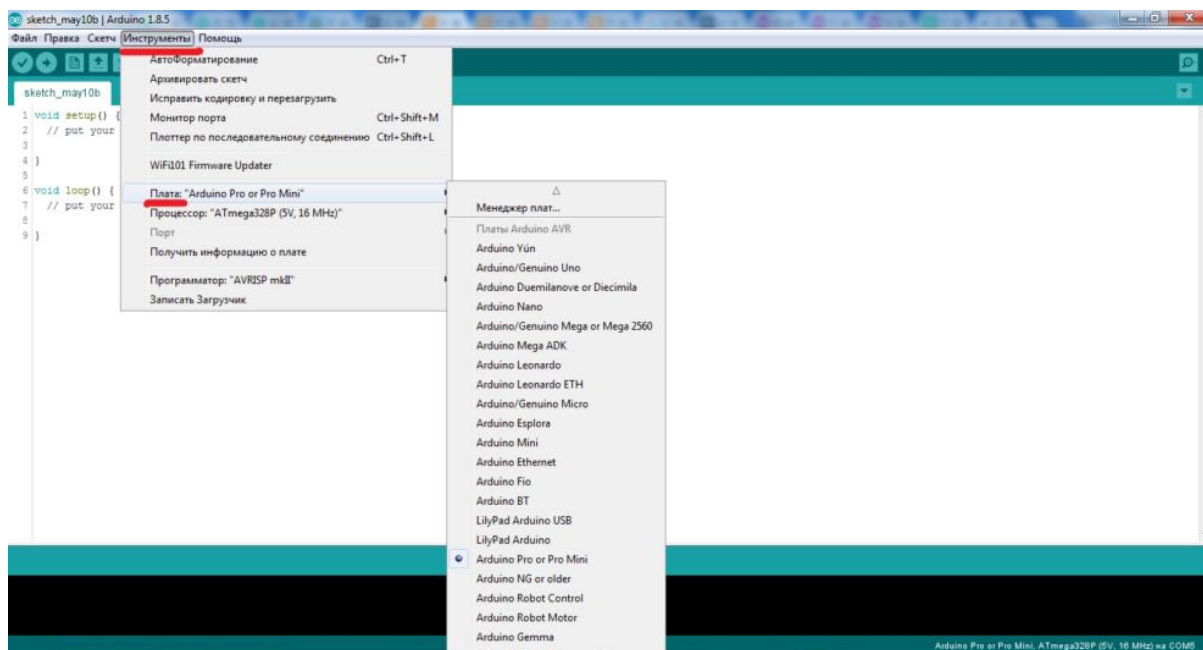


Рисунок 15 – Выбор платы

Далее необходимо выбрать тип контроллера, который установлен на модуле Arduino (на каждом контроллере есть маркировка). Это можно сделать в меню **Инструменты -> Процессор** (рисунок 16).

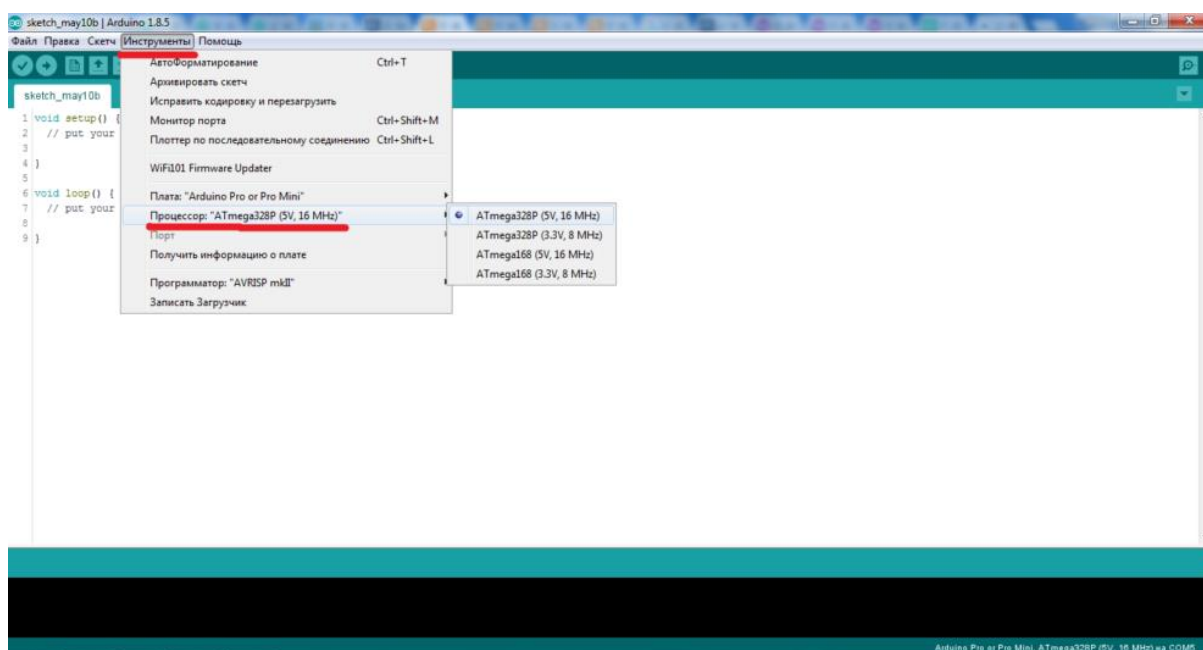


Рисунок 16 – Выбор процессора

### ***Компиляция программы:***

Теперь, когда выбран конкретный модуль Arduino, можно переходить к компиляции написанной программы. Давайте разберёмся, что же такое компиляция. Если говорить простым языком и касательно среды Arduino IDE, то компиляция — это перевод написанной в IDE программы в эквивалентную,

но в машинных кодах. Программа записывается в микроконтроллер именно в машинных кодах, а не в том виде, в котором она написана в IDE. Компиляция также помогает найти ошибки в программе, т.к. компиляция не будет выполнена, если в программе есть ошибки.

С понятием компиляции мы разобрались. А как теперь открыть скомпилировать написанную программу? Для этого в Arduino IDE предусмотрено несколько способов:

- из пункта меню **Скетч -> Проверить/Компилировать**;

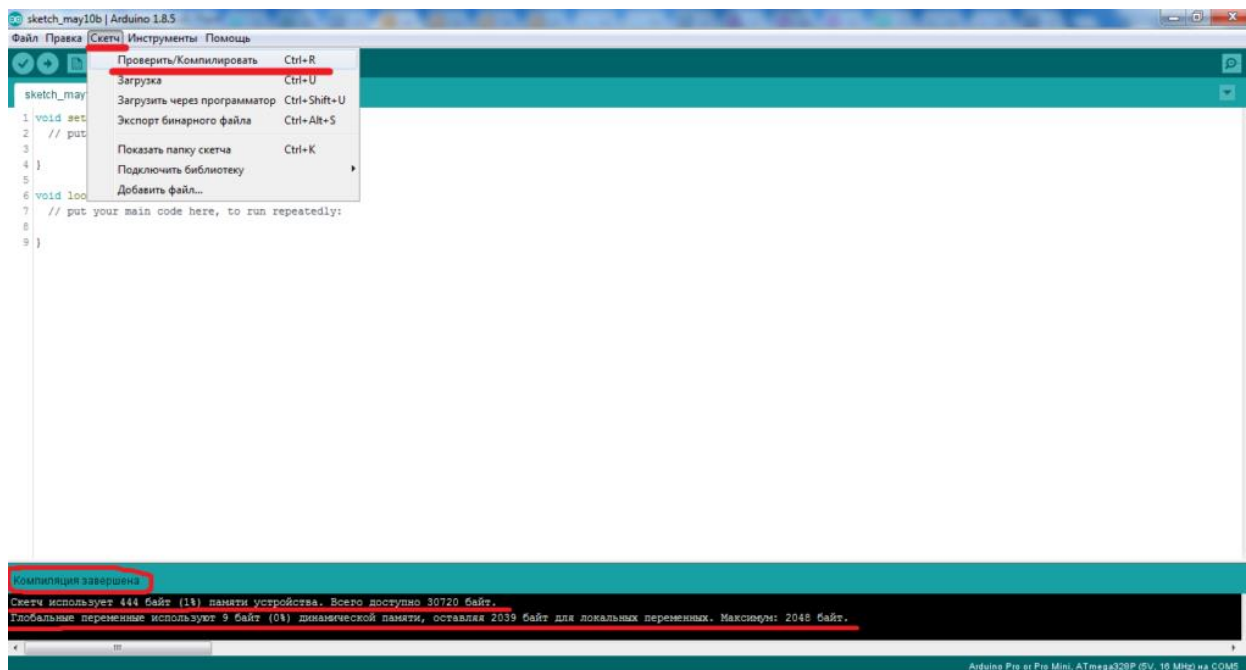


Рисунок 17 – Компиляция программы

- при помощи комбинации клавиш **Ctrl + R**;
- из панели с часто используемыми командами



Рисунок 18 – Компиляция программы

После успешной компиляции будет выведено сообщение об этом. Также в области сообщений можно найти информацию о том, сколько памяти занимает написанная программа. Если в программе есть ошибки, тогда в области сообщений будет выведено сообщение с указанием конкретной строки и ошибки в ней.

### **Выбор программатора:**

Пункт меню **Инструменты -> Программатор** используется для выбора аппаратного программатора, если программирование модуля или микроконтроллера осуществляется **не** при помощи встроенного USB-

последовательного соединения. Как правило, эта команда используется довольно редко, однако может пригодиться, например, при записи загрузчика в новый микроконтроллер.

Т.к. в модулях Arduino уже есть свой встроенный программатор, то в качестве программатора в меню **Инструменты** -> **Программатор** нужно оставить стандартный **AVRISP mkII** (рисунок 19).

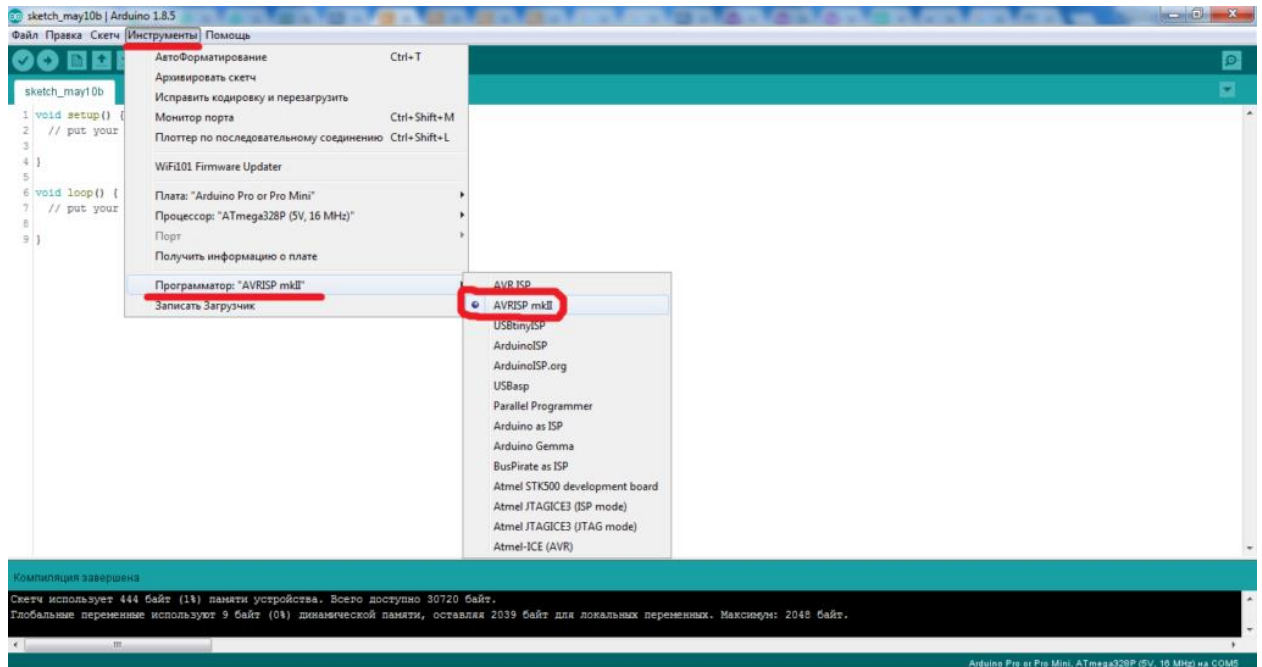


Рисунок 19 – Выбор программатора

### **Загрузка программы:**

Теперь, когда почти со всеми элементами среды Arduino IDE мы разобрались, можно приступить к финальному этапу — загрузке программы в модуль Arduino.

Прежде чем загружать программу, нужно выбрать порт, к которому подключён Ваш модуль Arduino. Выбрать его можно в меню **Инструменты** -> **Порт** (рисунок 20).

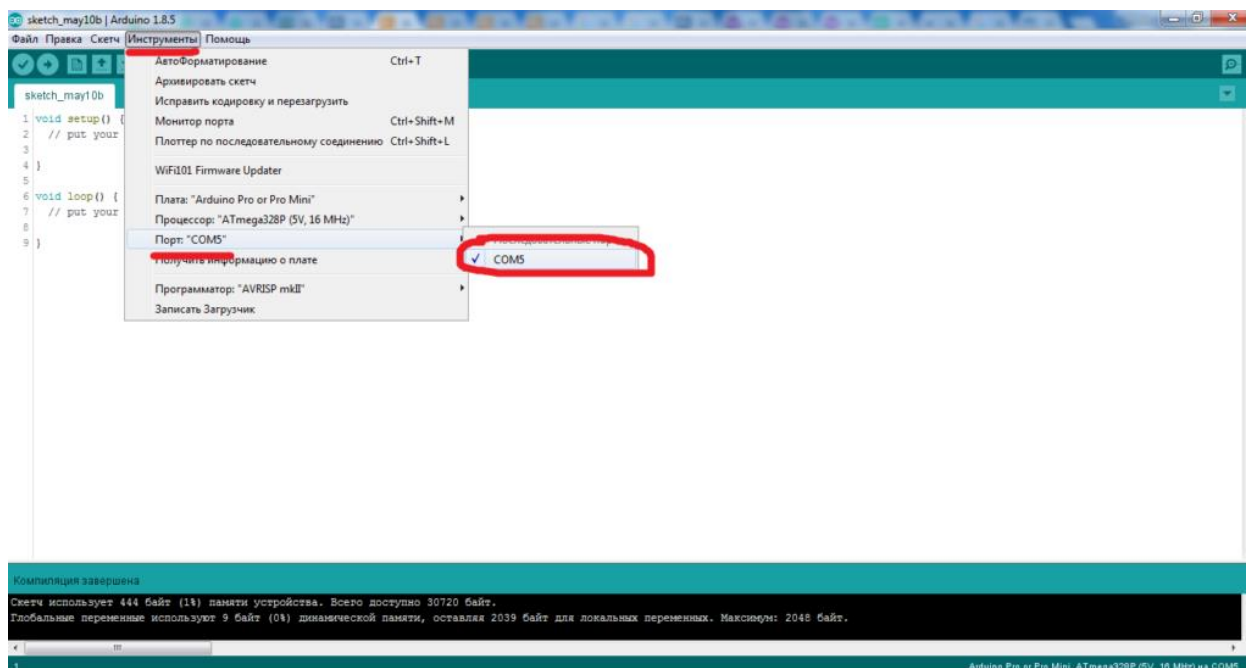


Рисунок 20 – Выбор порта для загрузки

Это не обязательно должен быть COM5, как на рисунке. Имя порта у Вас, скорее всего, будет другим (COM3, COM 10 и т.д.).

Когда выбран соответствующий модуль Arduino, нужный порт, процессор и программатор, можно приступать к загрузке программы.

Для этого в Arduino IDE предусмотрено несколько способов:

- из пункта меню **Скетч -> Загрузка** (рисунок 21);

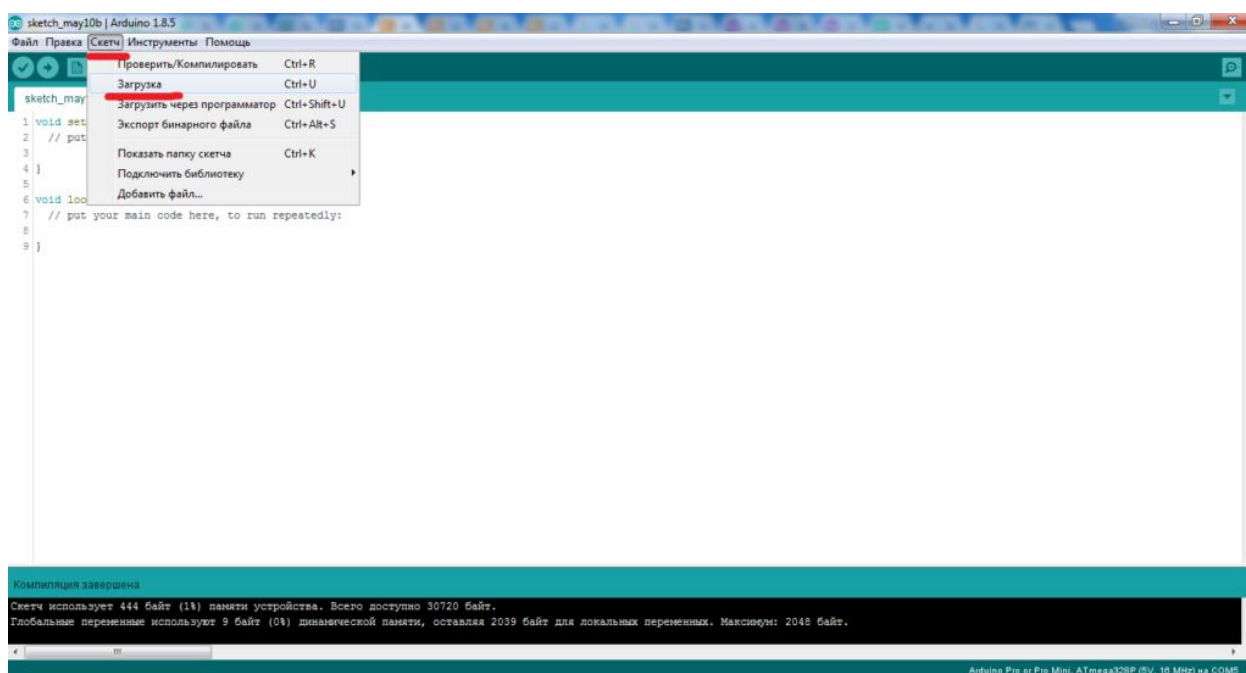


Рисунок 21 – Загрузка скетча

- при помощи комбинации клавиш **Ctrl + U**;
- из панели с часто используемыми командами (рисунок 22).



Рисунок 22 – Загрузка скетча

В процессе загрузки программы первым этапом код будет скомпилирован, а только потом, если не возникло ошибок в ходе компиляции, будет записан в модуль Arduino. При успешной загрузке программы в области сообщений появится сообщение об этом (рисунок 23).

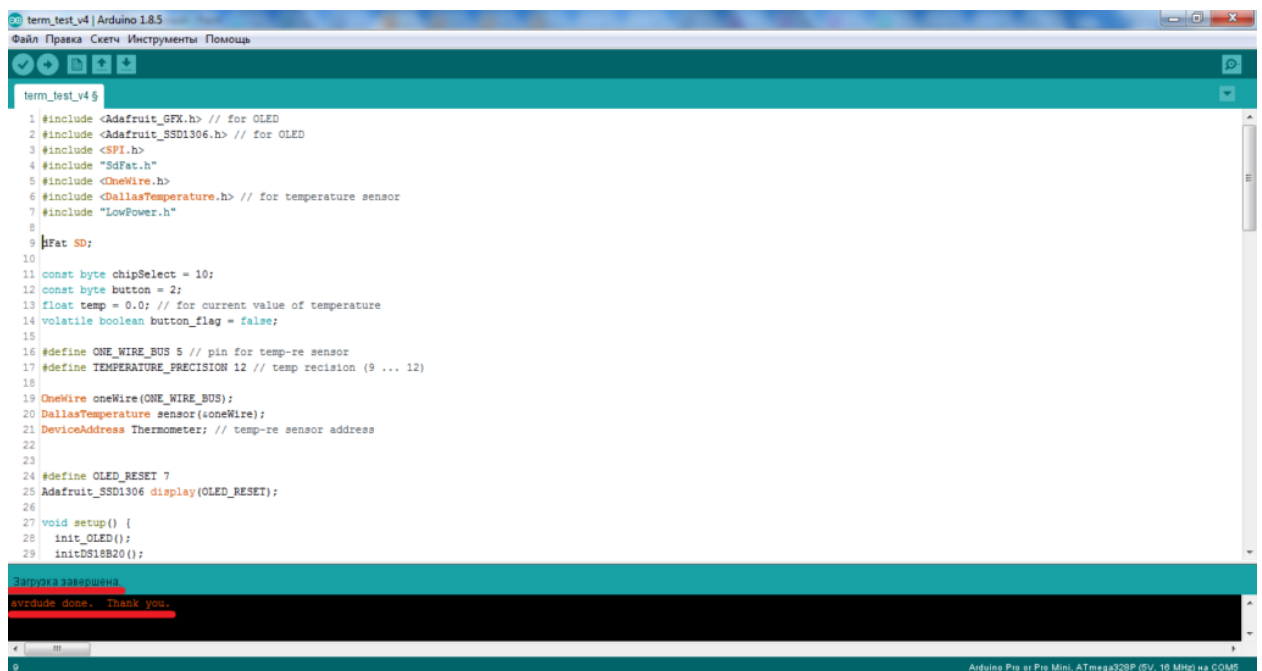


Рисунок 23 – Сообщение об успешной загрузке

Рекомендуем включить подробный вывод информации при компиляции и загрузке программы, это зачастую помогает выявить тип ошибки при компиляции или загрузке программы. Для этого в меню **Файл -> Настройки** (рисунок 24) установить соответствующие галочки.

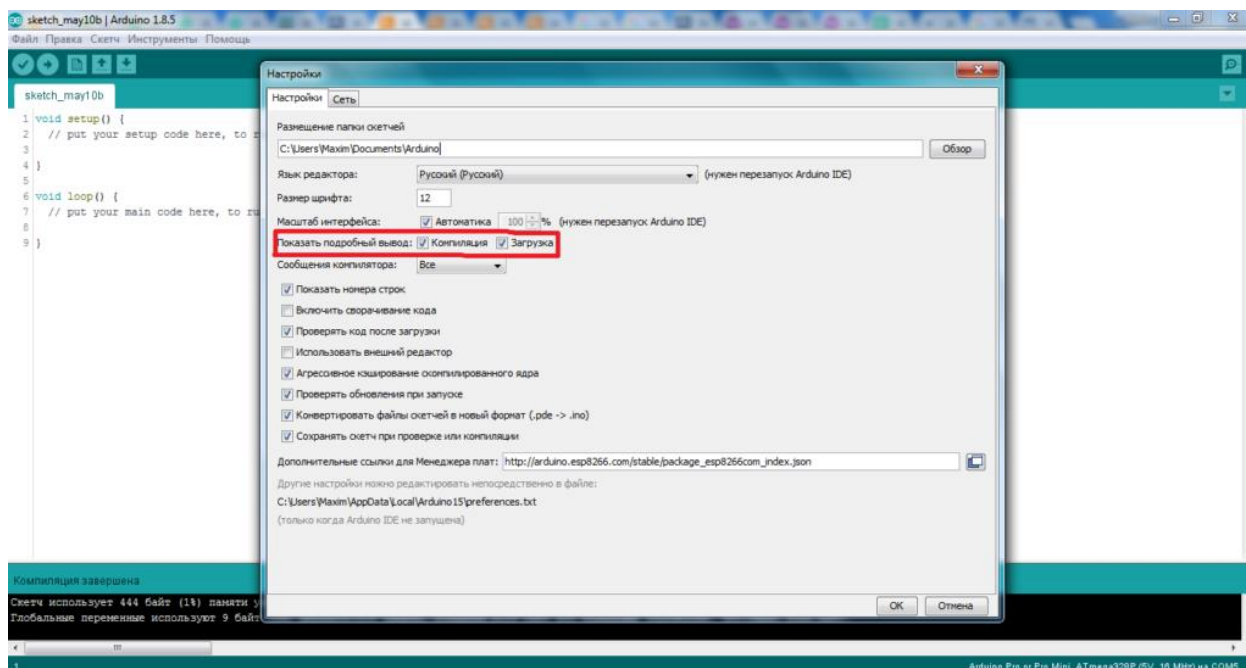


Рисунок 24 – Включение подробного отчёта компиляции

### *Монитор последовательного порта:*

Между Arduino и компьютером можно обмениваться данными через последовательный порт (он же интерфейс UART). Монитор последовательного порта может использоваться как для вывода отладочной информации от модуля Arduino, так и для других целей. Через него можно как отправлять данные в модуль Arduino, так и получать данные от него. Не забудьте выбрать порт, к которому подключён модуль Arduino, иначе монитор последовательного порта не откроется! При его открытии модуль Arduino перезагрузится!

Открыть окно монитора последовательного порта можно несколькими способами:

- из пункта меню **Инструменты -> Монитор порта** (рисунок 25);



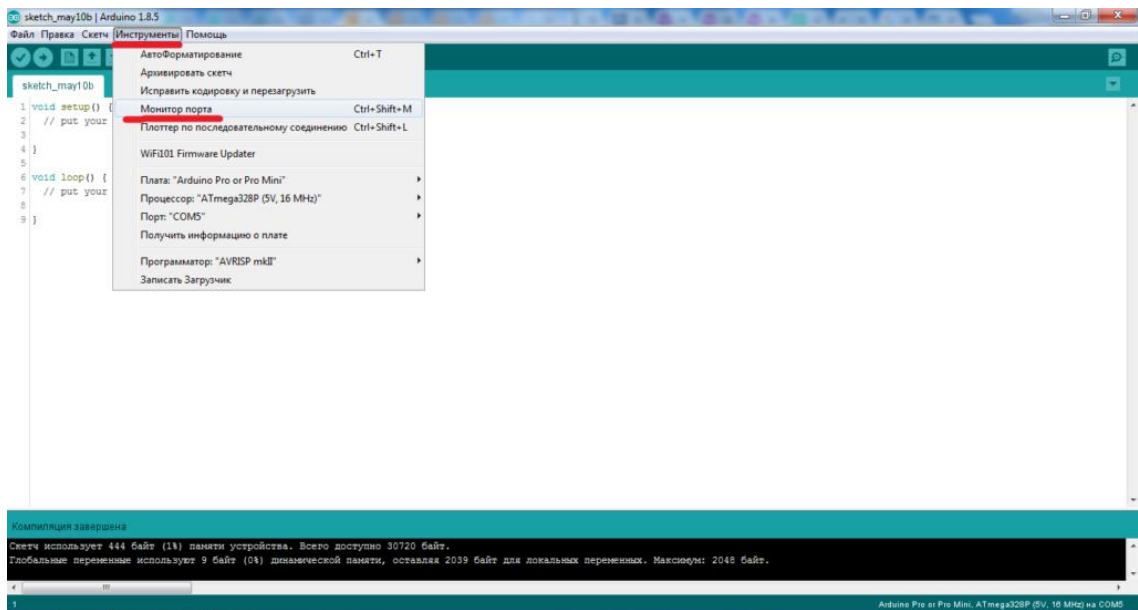


Рисунок 25 – Открытие последовательного порта

- при помощи комбинации клавиш ***Ctrl + Shift + M***;
- из панели с часто используемыми командами (в правом верхнем углу, рисунок 26)



Рисунок 26 – Значок последовательного порта

После открытия монитора последовательного порта появится следующее окошко как на рисунке 27:

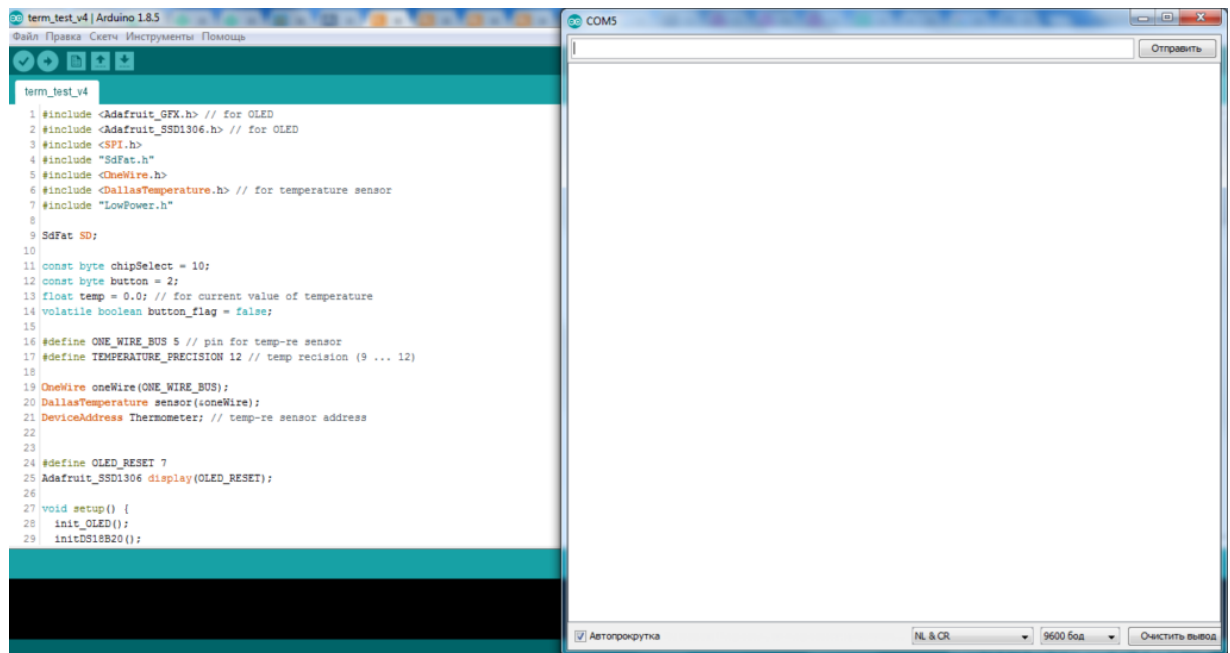


Рисунок 27 – Окно последовательного порта

В самом низу этого окошка можно изменить скорость работы порта (***она должна совпадать с той, которая указана в программе!***) и очистить окно.