Начало работы с ARDUINO

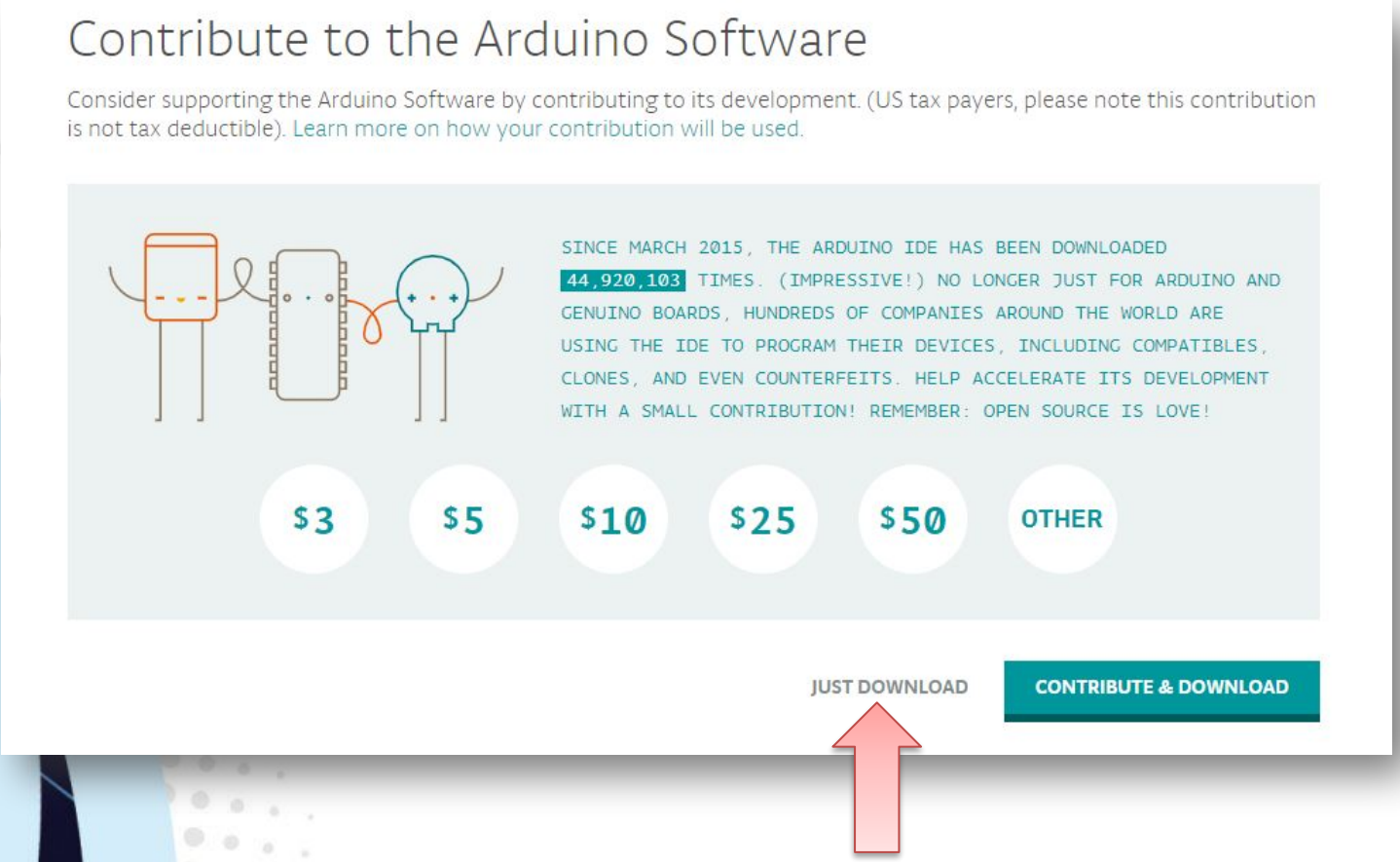
Загрузка и установка программного обеспечения (ПО) и драйверов

1. Качаем и устанавливаем актуальную версию Arduino IDE с сайта разработчика:

https://www.arduino.cc

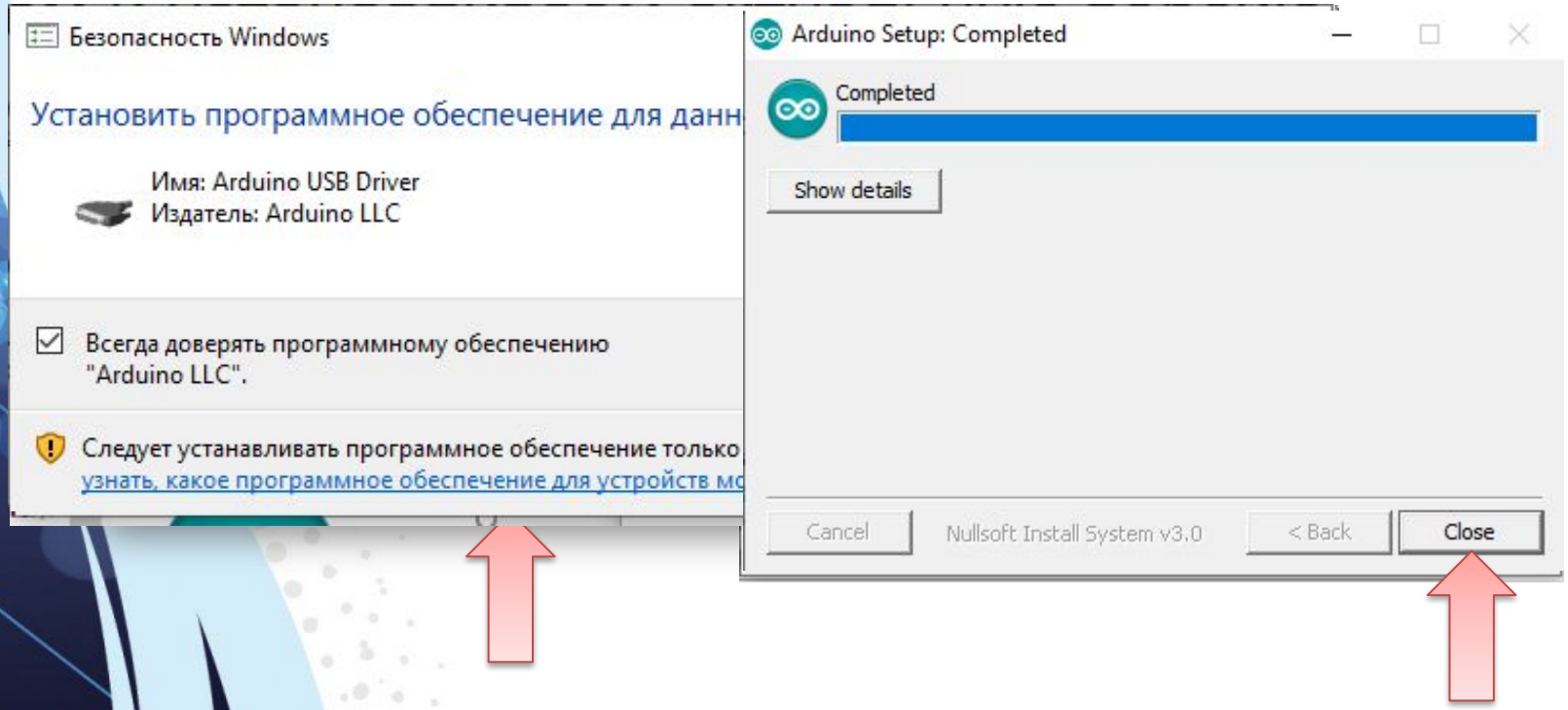
Загрузка и установка программного обеспечения (ПО) и драйверов

1. Качаем и устанавливаем актуальную версию Arduino IDE с сайта разработчика:



Загрузка и установка программного обеспечения (ПО) и драйверов

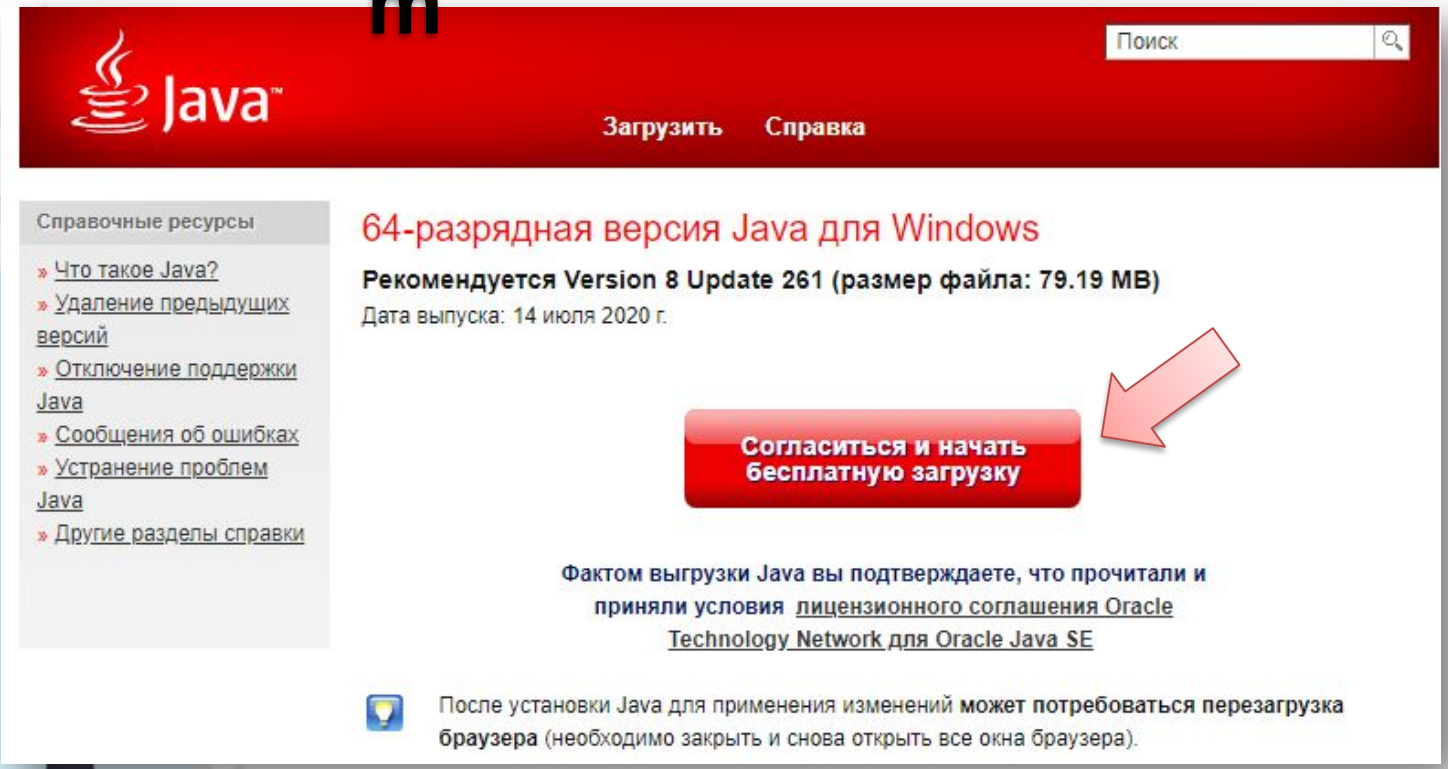
1. Качаем и устанавливаем актуальную версию Arduino IDE с сайта разработчика:



Загрузка и установка программного обеспечения (ПО) и драйверов

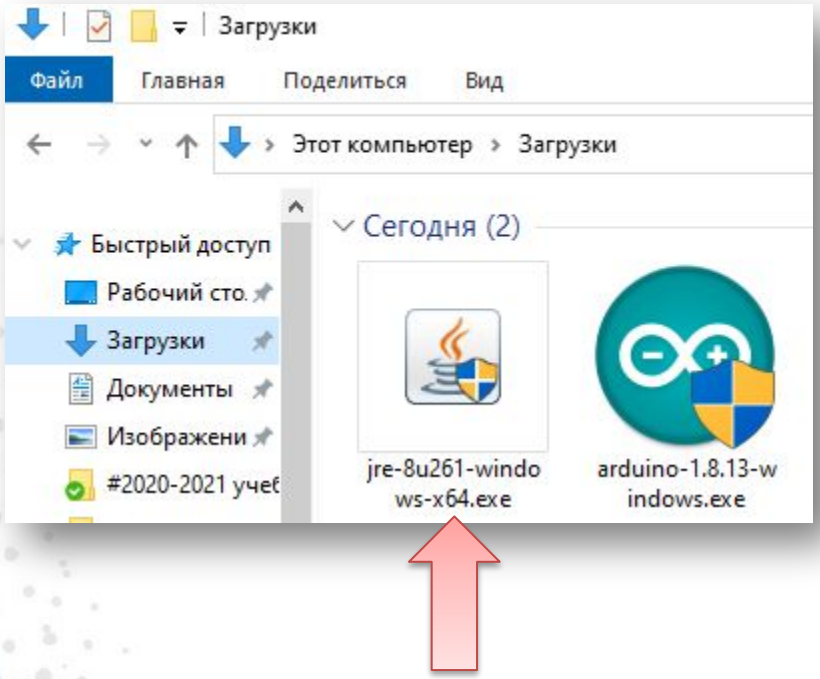
2. Качаем и устанавливаем JRE (Java Runtime Environment) с сайта разработчика:

https://www.java.com



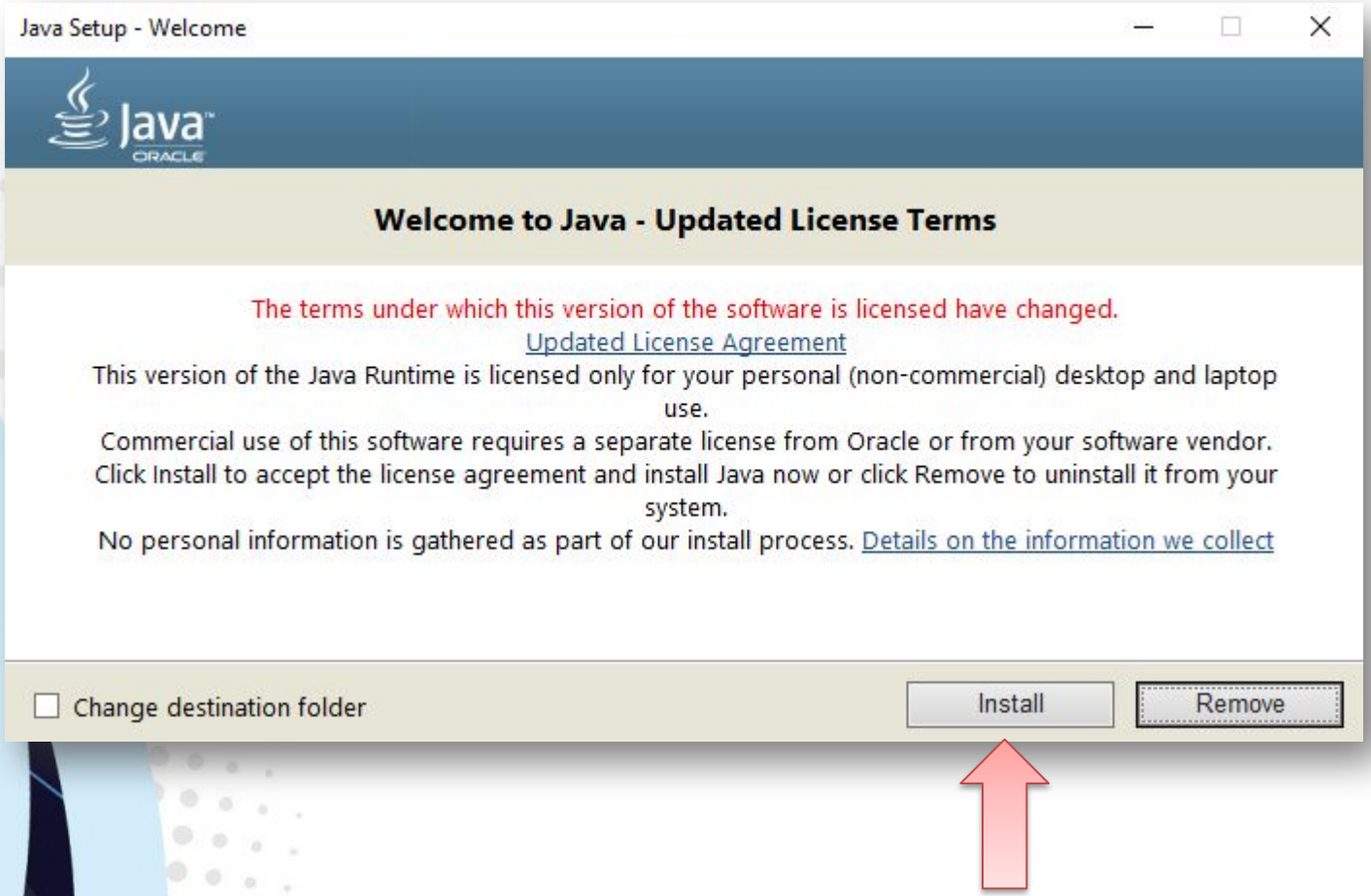
Загрузка и установка программного обеспечения (ПО) и драйверов

2. Качаем и устанавливаем JRE (Java Runtime Environment) с сайта разработчика:



Загрузка и установка программного обеспечения (ПО) и драйверов

2. Качаем и устанавливаем JRE (Java Runtime Environment) с сайта разработчика:



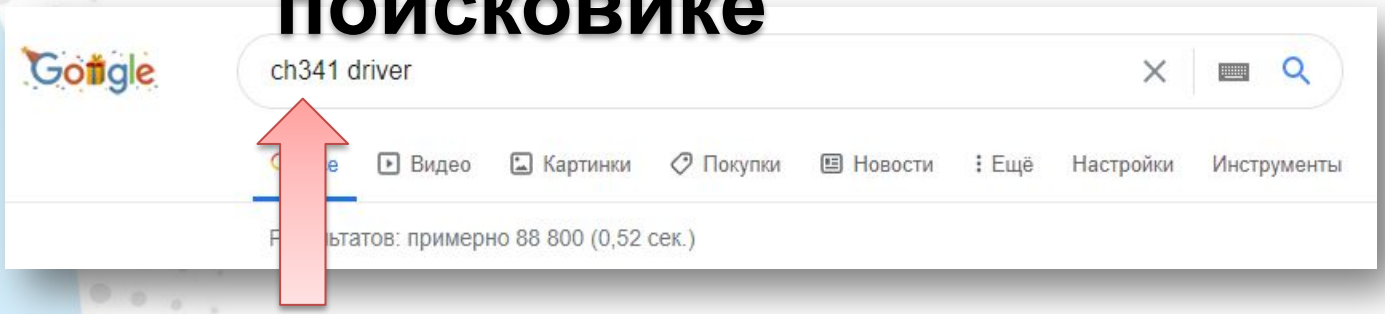
Загрузка и установка программного обеспечения (ПО) и драйверов

3. При установке Arduio IDE должны автоматически поставиться необходимые для работы драйверы.

На большинстве китайских Arduino-плат стоит контроллер интерфейса USB CH341, для работы с ним нужен отдельный специальный драйвер.

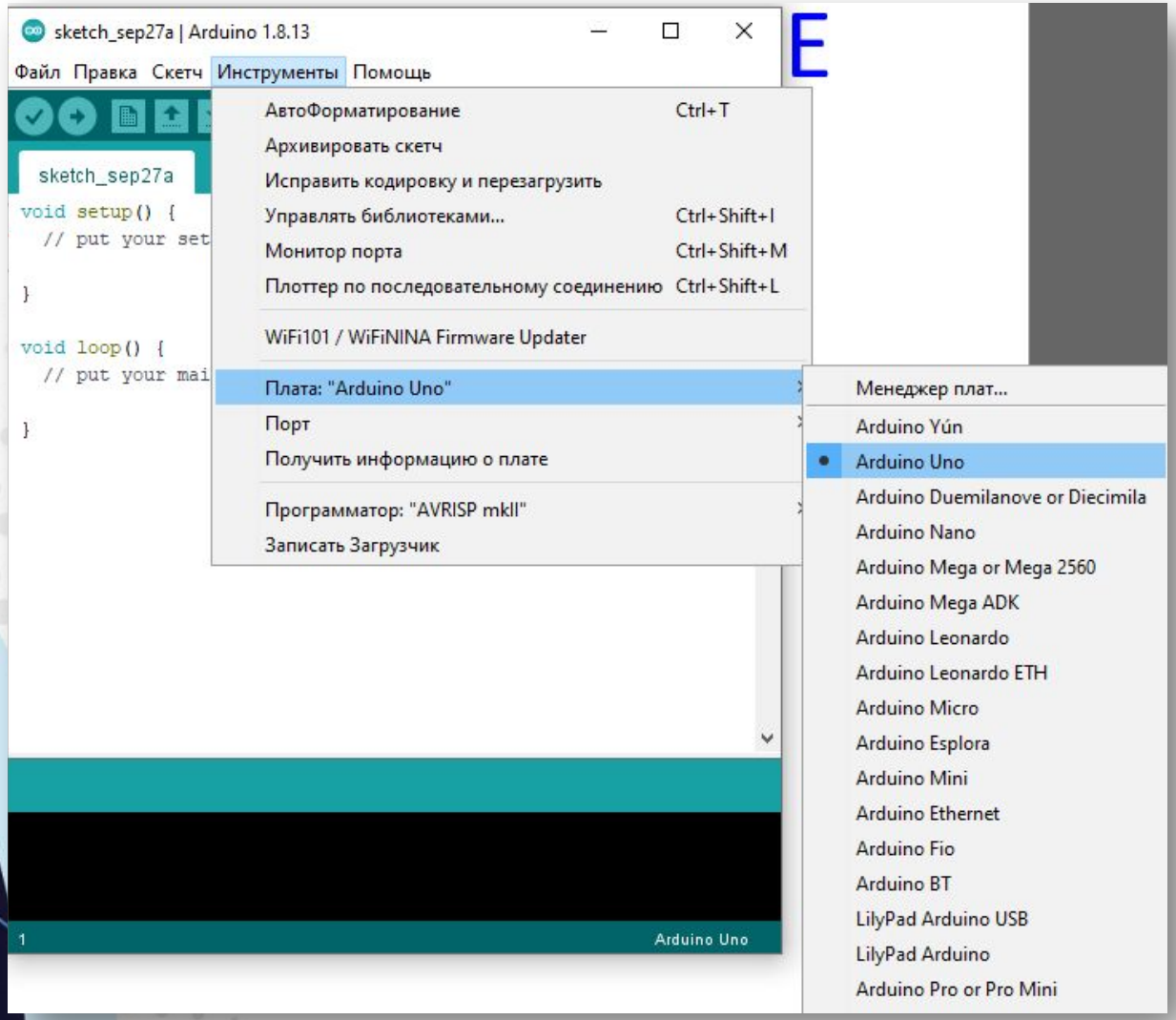
(*Китайский контроллер ничем не хуже, он просто дешевле :)*

ищем в поисковике



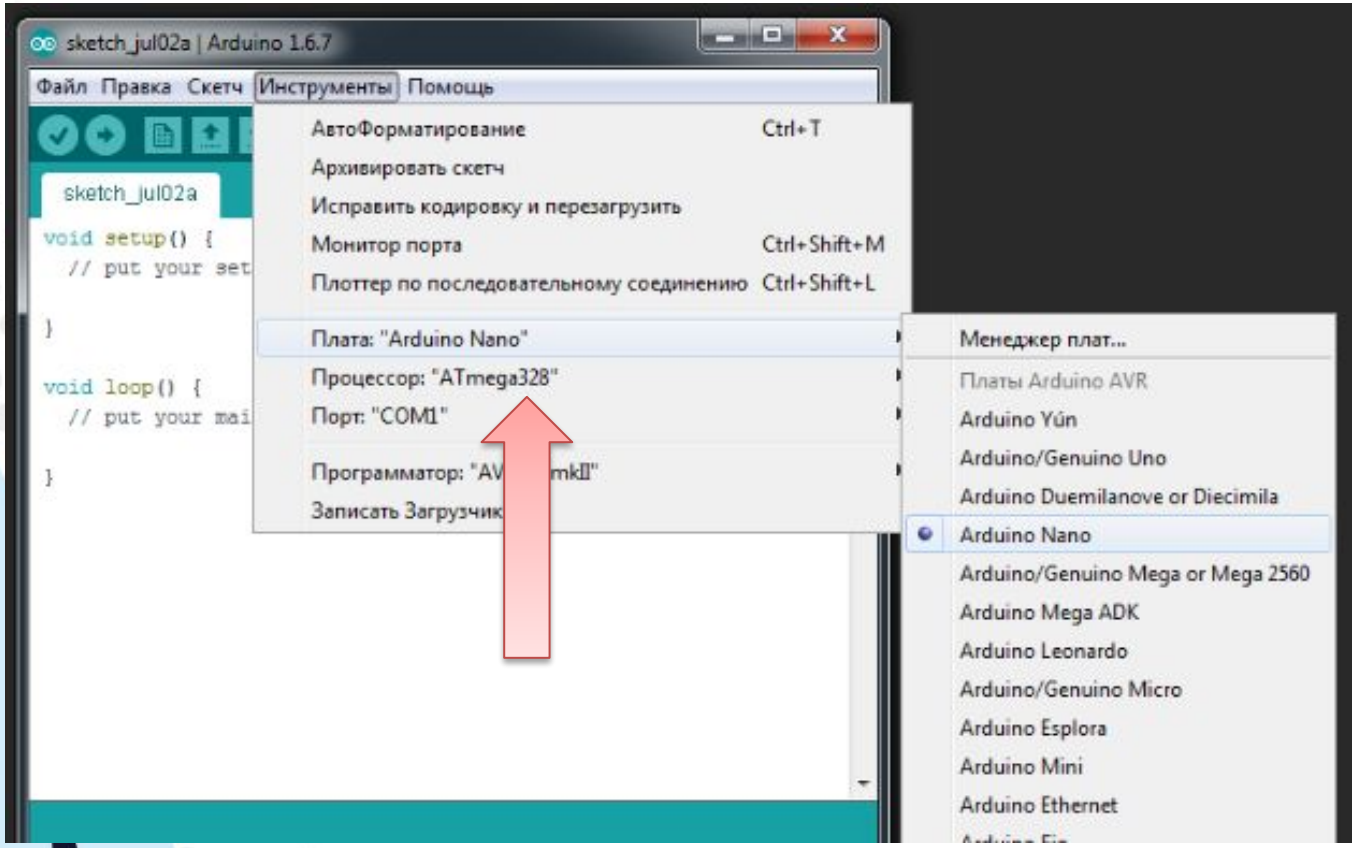
Настраиваем Arduino IDE

Запустить Arduino IDE, выбрать плату (Инструменты\плата\”ваша плата”)



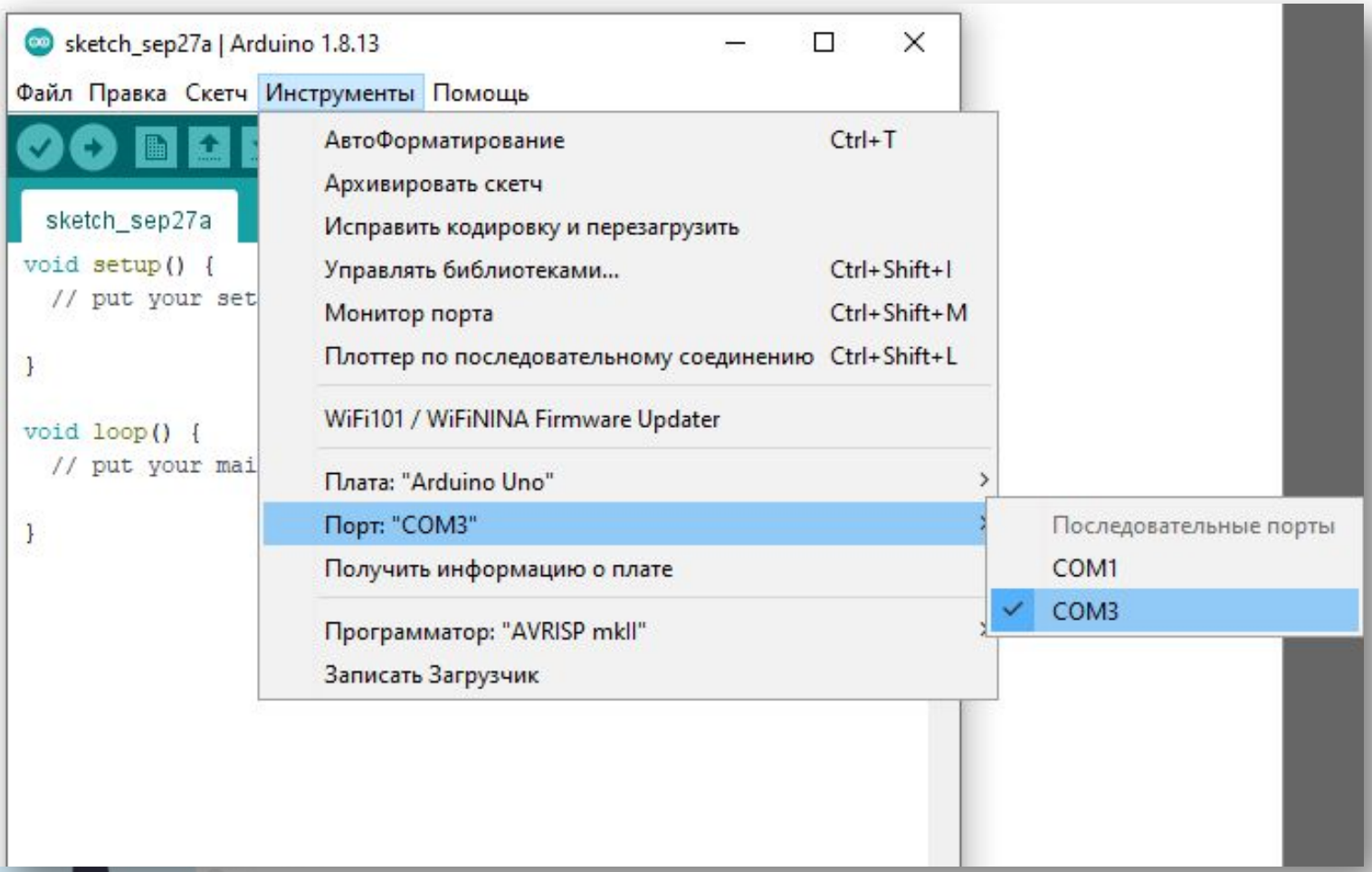
Настраиваем Arduino IDE

Выбрать модель платы/микроконтроллера (загрузчика) Инструменты\Процессор\”Ваша модель”. (для Nano)



Настраиваем Arduino IDE

Выбрать порт: инструменты\порт\”COM отличный от COM1, например COM3, COM5…”

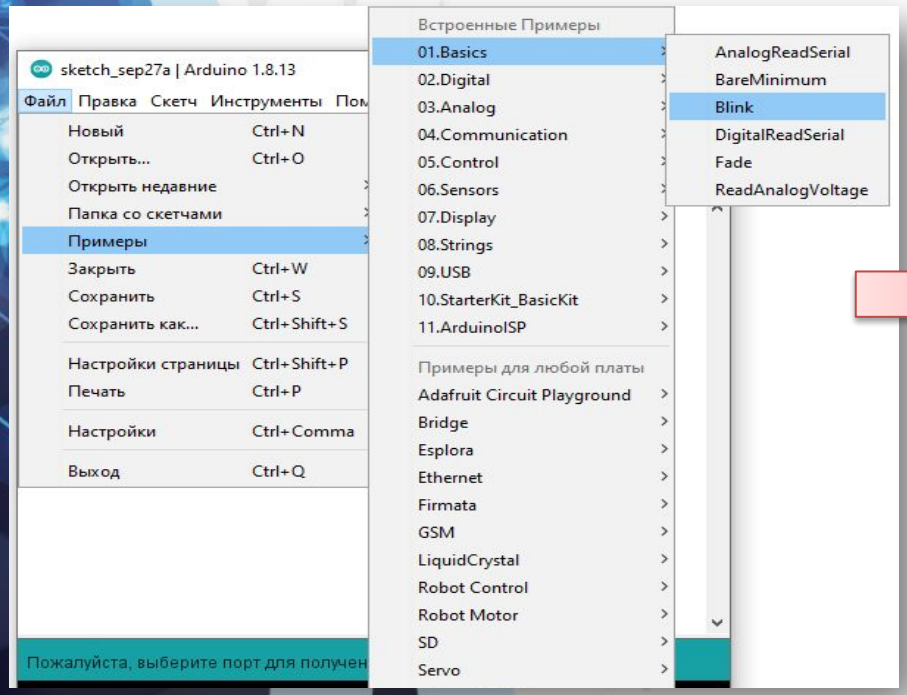


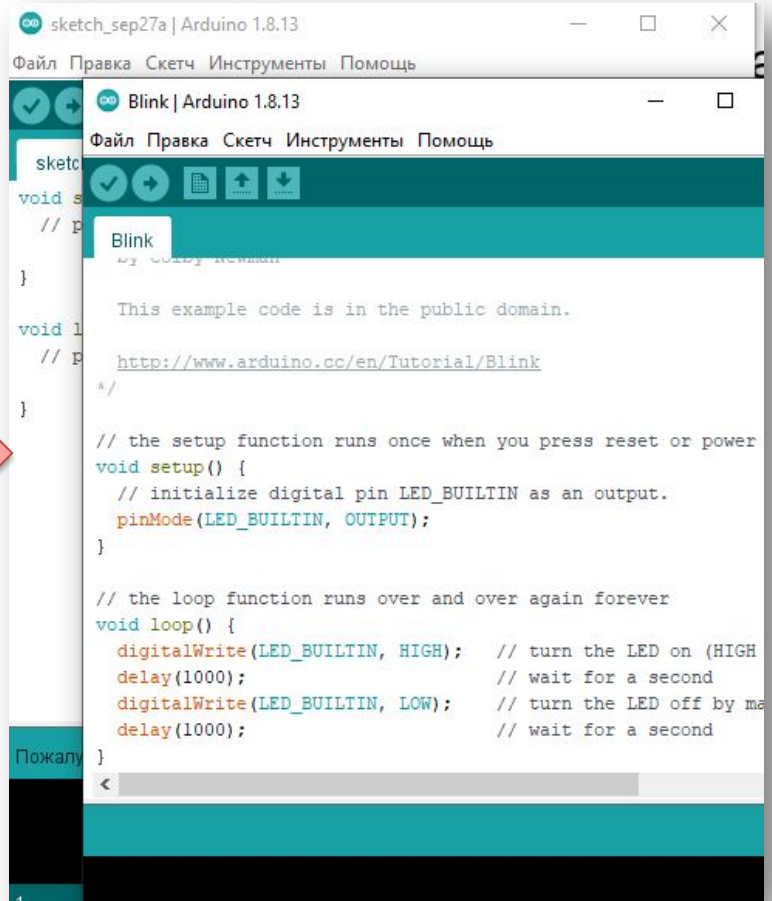
Настраиваем Arduino IDE

ВНИМАНИЕ! Как только достанете Arduino из пакетика, сразу прошейте в неё скетч с миганием светодиода (blink.ino)

Таким образом вы узнаете, что Ардуина рабочая

(на тот случай, когда после сборки/пайки она перестанет работать и прошиваться)





Ответы на 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 будет автоматически удалена.

Можно ли записать две прошивки, чтобы они работали вместе?

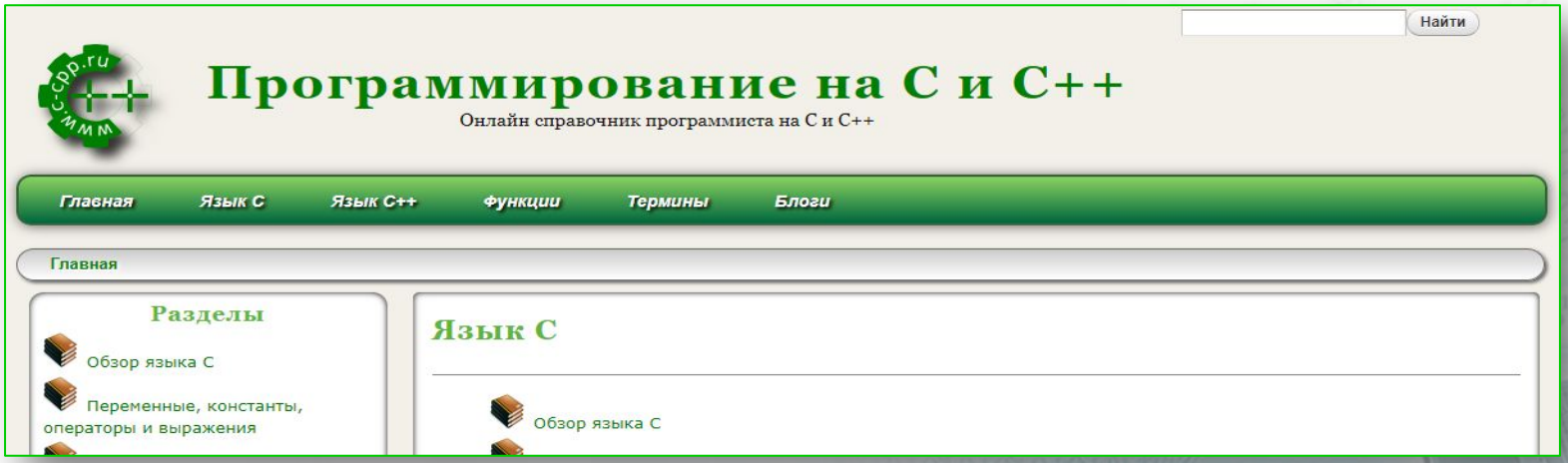
- Нет, при прошивке удаляются абсолютно все старые данные.

Программирование микроконтроллера:

Общение с микроконтроллером происходит в письменном виде на языке программирования, язык это очень чёткий, строгий и имеет свой синтаксис и некоторые нормы оформления.

Официальной средой разработки является Arduino IDE, где программирование осуществляется на языке **C++** – одном из самых популярных и мощных языков.

Справочник по языку для любознательных:



http://www.c-cpp.ru/book

Видеоурок:

6 мин.

Видео прилагается к презентации

Конспект 1.0:

myFirstSketch | Arduino 1.6.7

Файл Правка Скетч Инструменты Помощь

myFirstSketch §

```
11 // подключаются библиотеки, например библиотека серво
12
13 #include <Servo.h>
14
15 // объявляются переменные
16
17 int lolokek;
18
19 void setup() {
20   // всё находящееся здесь выполнится 1 раз при запуске Ардуин
21   // настраиваем датчики и саму Ардуино, готовимся к
22   // выполнению основной программы
23 }
24
25 void loop() {
26   // бесконечный цикл, всё выполняется сверху вниз
27   // идёт чтение значений с датчиков, их обработка
28
```

Загрузка завершена.

```
Скетч использует 450 байт (1%) памяти устройства. Всего доступно
Глобальные переменные используют 9 байт (0%) динамической памяти,
```

27 Arduino Nano, ATmega328 on COM3

Конспект 1.0

// Однострочный комментарий
/* Многострочный комментарий
*/ (комментарии не занимают память)
#include - Подключить файл (библиотеку)
void setup() {} - Всё находящееся внутри {}
будет выполнено 1 раз при загрузке Ардуино
void loop() {} - Всё находящееся внутри {}
бесконечно повторяется сверху вниз

Комментарии:

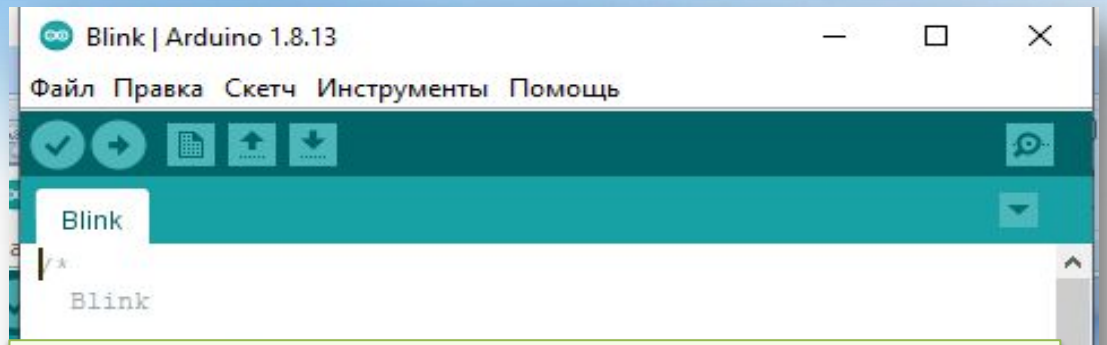
Однострочный комментарий:

// однострочный комментарий

// компилятор меня игнорирует =(

Многострочный комментарий:

/* Многострочный комментарий */



Несколько раз включает светодиодный индикатор на одну секунду, а затем выключается на одну секунду.

Большинство Arduino имеют встроенный светодиод, которым вы можете управлять. На UNO, MEGA и ZERO он подключен к цифровому выводу 13, на MKR1000 - к выводу 6. LED_BUILTIN установлен на правильный вывод светодиода независимо от того, какая плата используется.

Если вы хотите знать, к какому выводу подключен встроенный светодиод на вашем Arduino

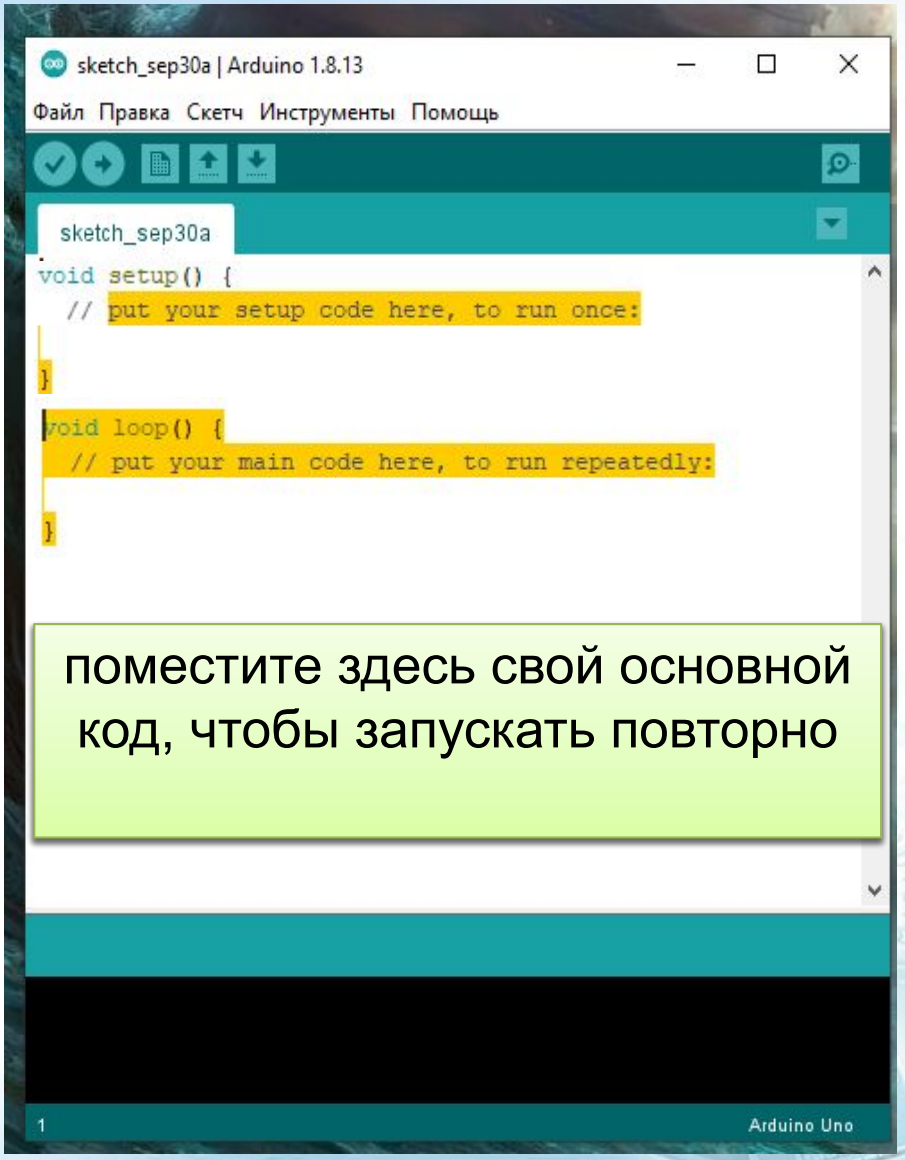
модель, проверьте технические характеристики вашей платы по адресу:

Комментарии:

При запуске Arduino IDE даёт нам заготовку в виде двух обязательных функций: **setup** и **loop**

Код в блоке setup()
 выполняется один раз при каждом запуске микроконтроллера. Код в блоке **loop()**
 выполняется “по кругу” на всём протяжении работы микроконтроллера, начиная с момента завершения выполнения **setup()**

.



Видеоурок:

myFirstSketch | Arduino 1.6.7

Файл Правка Скетч Инструменты Помощь

myFirstSketch §

```
1  boolean bybool = 1;
2  boolean azaza = 0;
3  boolean logicLOLo = true; // инвертируем !logicLOLO
4
5  char mysymbol = 'a'; // хранится в виде числа согласно ASCII
6
7  byte lolbyte = 150;
8
9  int myvalue 22000;
10
11 unsigned int lolokek = 50000;
12
13 long BIGBIG = 123423425;
14
15 float superTen = 10.25
16
17
18 void setup() {
19   // put your setup code here, to run once:
```

Загрузка завершена.

```
Скетч использует 450 байт (1%) памяти устройства. Всего доступно
Глобальные переменные используют 9 байт (0%) динамической памяти,
```

15 Arduino Nano, ATmega328 on COM3

Конспект 1.1

ТИПЫ ДАННЫХ ARDUINO

объявление "тип данных" "название";

или "тип данных" "имя" = "значение";

Название	Вес	Диапазон	Особенность
boolean	1 байт	0 или 1	Логическая переменная, может принимать значения **true** (1) и **false** (0)
char	1 байт	-128... 127	Хранит номер символа из таблицы символов ASCII
byte	1 байт	0... 255	
int	2 байта	-32 768... 32 767	
unsigned int	2 байта	0... 65 535	
word	2 байта	0... 65 535	То же самое, что unsigned int
long	4 байта	-2 147 483 648... 2 147 483 647	- 2 миллиарда... 2 миллиарда
unsigned long	4 байта	0... 4 294 967 295	0... 4 миллиарда...
float	4 байта	-3.4028235E+38... 3.4028235E+38	Хранит числа с плавающей точкой (десятичные дроби). **Точность: 6-7 знаков**
double	4 байта		То же самое, что float

Видеоурок:

myFirstSketch | Arduino 1.6.7

Файл Правка Скетч Инструменты Помощь

myFirstSketch §

```
7  byte lolbyte = 150;
8
9  int myvalue 22000;
10
11 unsigned int lolokek = 50000;
12
13 long BIGBIG = 123423425;
14
15 float superTen = 10.25;
16 float opopo = 10.0;
17
18
19 void setup() {
20   int celoe_chislo = 10;
21   float des9tka = (float)celoe_chislo / 2.314;
22
23 }
24
25 void loop() {
```

Загрузка завершена.

```
Скетч использует 450 байт (1%) памяти устройства. Всего доступно
Глобальные переменные используют 9 байт (0%) динамической памяти,
```

Конспект 1.2

ОСОБЕННОСТИ FLOAT

1) Присваивать только значение с точкой, даже если оно целое (10.0)

2) Делить тоже только на числа с точкой, даже если они целые (переменная / 2.0)

3) При делении целочисленного типа с целью получить число с плавающей точкой, писать (float) перед вычислением!

P.S. Операции с числами типа float занимают гораздо больше времени, чем с целыми! Если нужна высокая скорость вычислений, лучше применять всякие хитрости =)

Синтаксис:

Тела функций заключаются в фигурные скобки **{ }**

Каждая команда заканчивается точкой с запятой **;**

Метод применяется к объекту через точку **.**

Пример: ***Serial.begin();***

Вызов функции или метода всегда заканчивается скобками **()**

Пример: ***loop()***

Разделитель десятичных дробей – точка **.**

Пример: ***0.25***

У запятой тут другое применение

Одиночный символ заключается в одиночные кавычки **'a'**

Строка и массив символов заключается в двойные кавычки **"строка"**

Синтаксис:

Регистр имеет значение, т.е. большая буква отличается от маленькой.

Пример: переменные val и Val – не одно и то же.