Лабораторная работа 1
Подключение светодиода к микроконтроллеру.

Цель работы: приобрести практические навыки по подключению светодиода к плате Ардуино.

Последовательность выполнения работы:
Изучить теоретические сведения, приведенные в лабораторной работе.
Собрать схемы на макетной плате для приведенных примеров.
Запрограммировать микроконтроллер согласно тексту, указанному в примере.
Выполнить задание для самостоятельной работы.

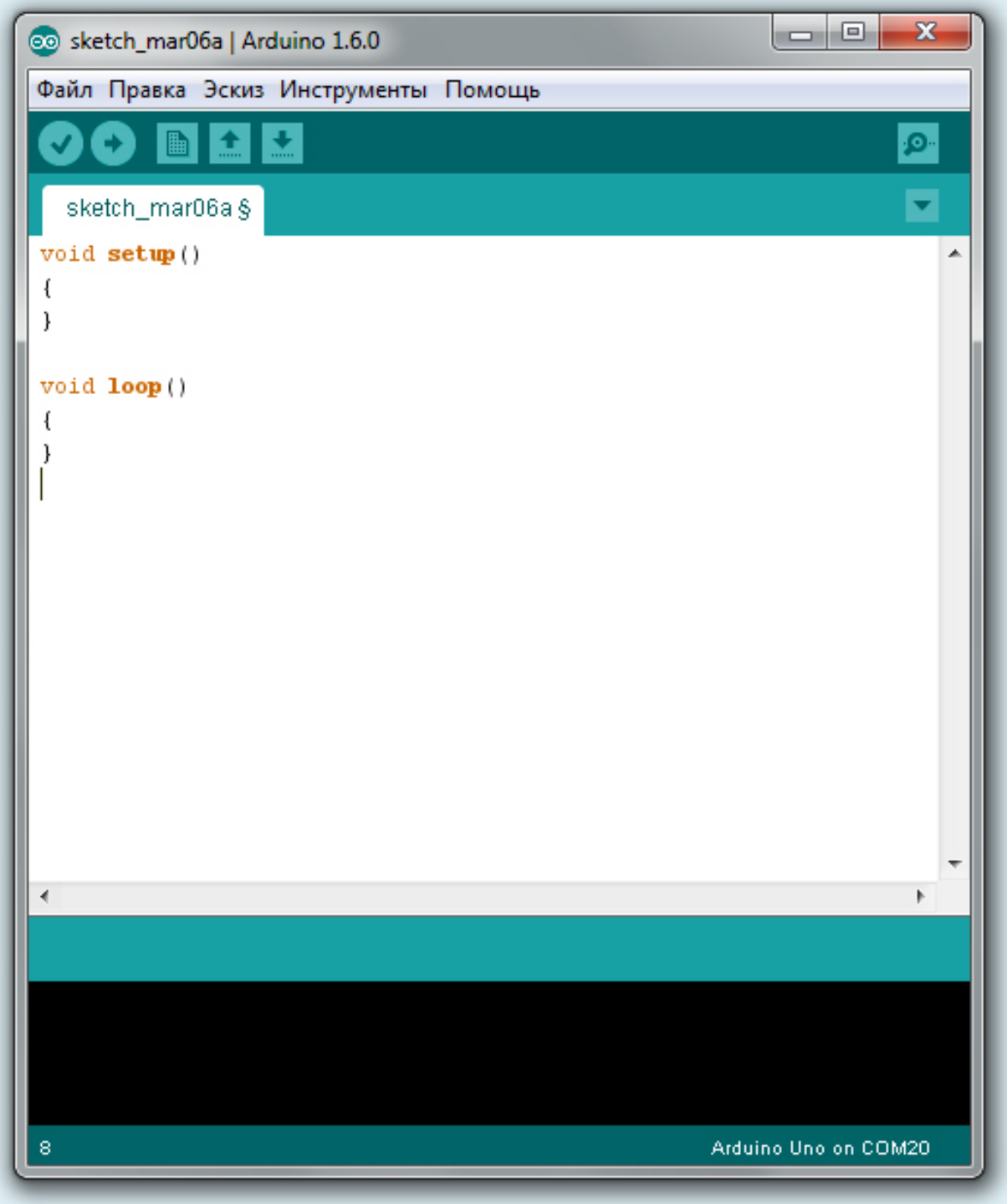
Содержание отчета:
Название лабораторной работы, ее цель.
Задание на лабораторную работу (по варианту).
Схемы подключения к микроконтроллеру.
Программный код для скетчей.
Вывод о проделанной работе.

Теоретические сведения

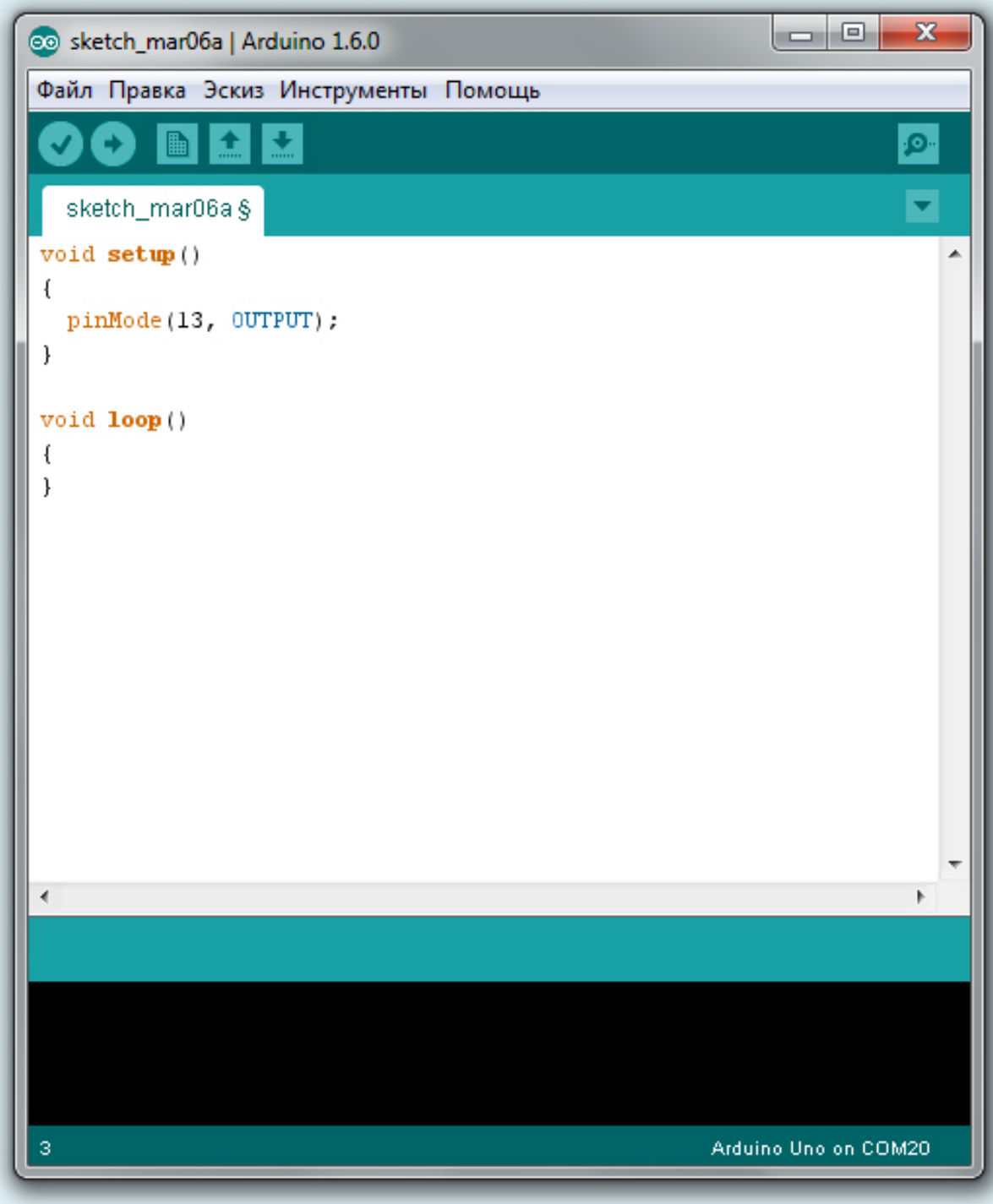
Обьявление переменных в ARDUINO IDE

boolean	Переменная с таким типом является логической. Она может принимать всего два значения, либо **true(истина)**, либо **false(ложь)**
char	Переменная с таким типом является знаковой и длинной в один байт. Может принимать значение от **-127** до **+127**, либо принимать значение символа из таблицы ASCII. Например символ **'1'** примет значение **48**. Более подробное описание символов мы затронем когда будем работать со строками.
unsigned char	То же самое что и **char** за одним исключением. Слово **unsigned** говорит о том, что значение данной переменной может принимать только в положительном диапазоне. То есть от **0** до **255** и никаких отрицательных.
int	Переменные с таким типом являются знаковыми, но в отличии от **char** имеют длину в два байта. Значения, которые можно хранить в данной переменной, лежат в диапазоне от **-32768** до **+32767**.
unsigned int	Этот тип переменной указывает тоже на двубайтовое значение, но как и **unsigned char** имеет строго положительное значение и лежит в диапазоне от **0** до **65535**.
byte	Данный тип переменной похож на **unsigned char** и принимает значения в пределах от **0** до **255**.
float	Данный тип переменной дает возможность храить значения с плавающей точкой. Под эту переменную будет выделено 4 байта. Диапазон значений может лежать в пределах от **-3.4028235E+38** до **3.4028235E+38**.

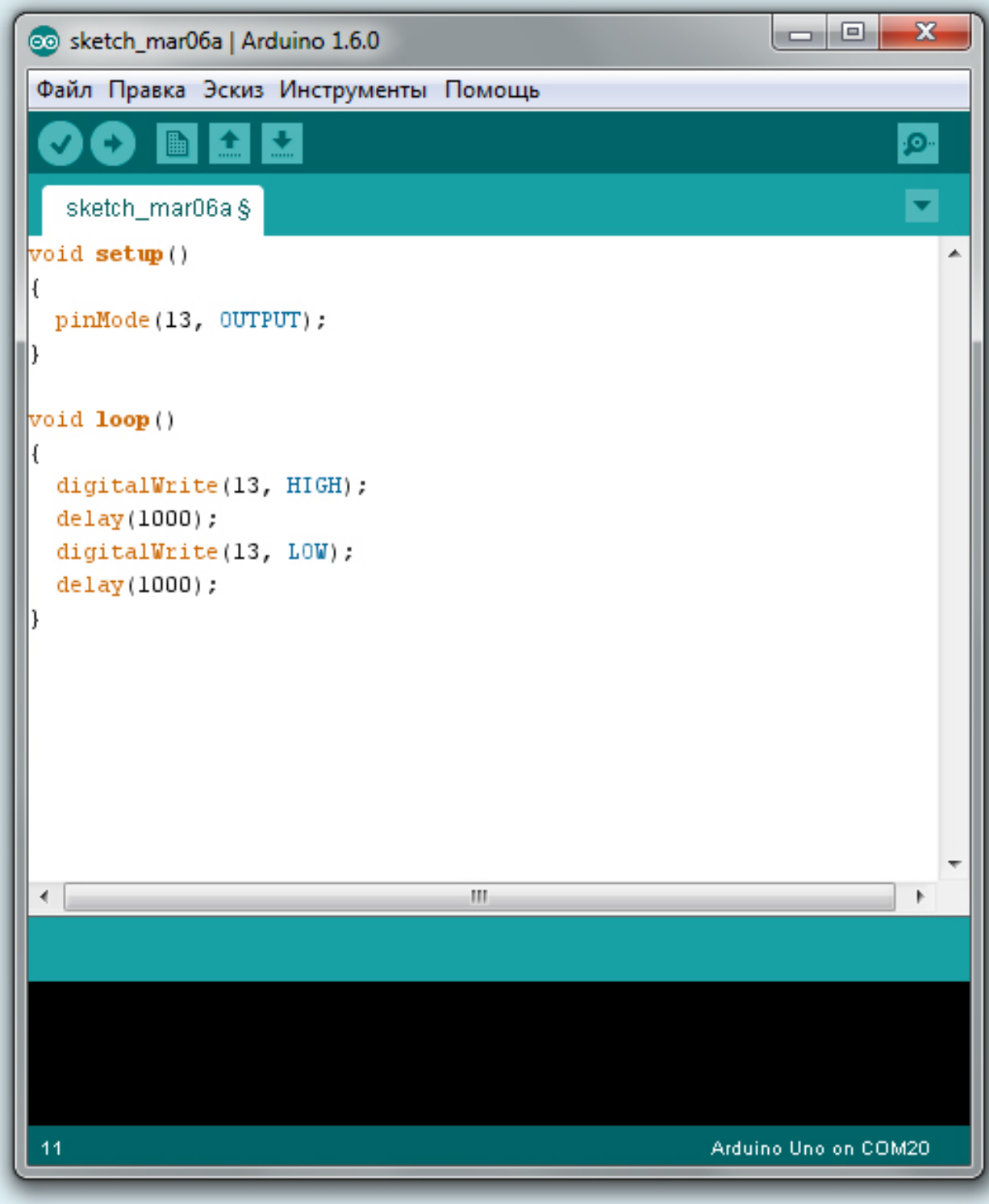
Вот мы и разобрались с типами переменных. Далее давайте взглянем на имена самих переменных. Имена можно задавать разнообразными способами. Например **R84Hhj7I** тоже может быть именем переменной. Но заметьте, такое название просто ужас. Поэтому рекомендую задавать человекопонятные имена, например **LEDdiod**. Сразу понятно, что в данной переменной лежит значения состояния светодиода, либо ноль, либо единица. Но есть одно ограничение, нельзя начинать имя переменной с цифры. **5LEDdiod** будет являться ошибкой. Это правило тянется из языка С, я не пробовал проверять на **arduino**, но надеюсь что оно и здесь работает. Так же необходимо запомнить что имя **LED** и **led** это не одно и тоже. Регистр тоже учитывается.
Так, с именами разобрались. Давайте, опираясь на полученные знания, попробуем написать первую программу. Что она будет делать? А ничего, просто микать светодиодом установленным на плате. На чистом поле текстового редактора пишем следующий код.


sketch_mar06a | Arduino 1.6.0
Файл Правка Эскиз Инструменты Помощь
sketch_mar06a §
void setup()
{
}
void loop()
{
}
8
Arduino Uno on COM20

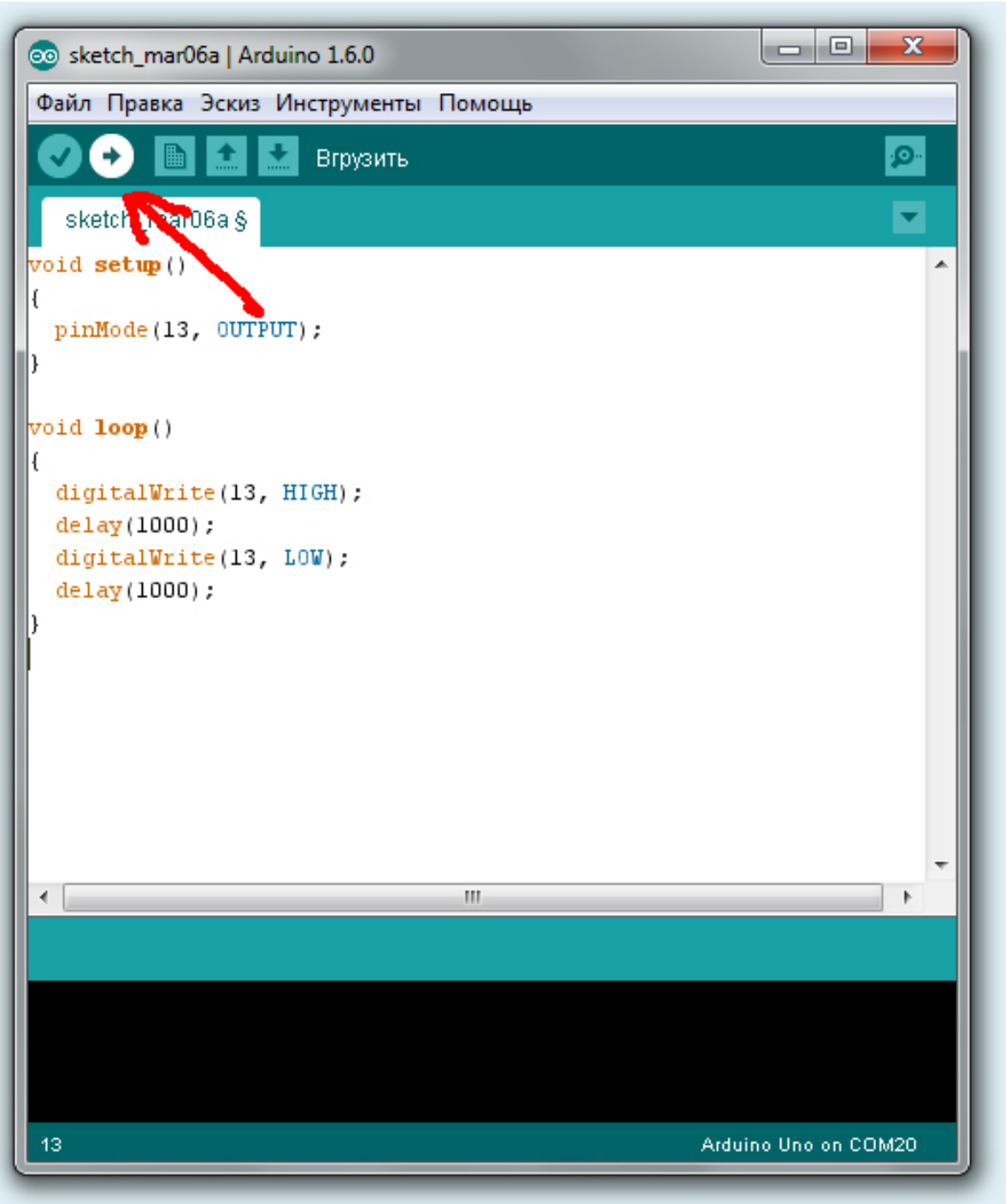
Что мы собственно здесь написали. А написали мы следующее, сначала мы объявили функцию **setup()**, а за ней функцию **loop()**. Объяснять что такое функция, пока очень рановато, поэтому просто копируем написанное, но на одном моменте я все же подробно остановлюсь. Дело в том что язык программирования для **arduino** это не С. И даже я больше скажу, что его даже нельзя назвать языком. Больше подходит конфигуратор-разработчик. Сейчас сами все поймете. Дело в том, что для программирования микроконтроллера необходима знание его архитектуры и умение обращаться с регистрами разной периферии, а в **arduino** это не нужно. Для работы той или иной периферии вам нужно лишь вызвать ту или иную функцию и передать по надобности какие-либо параметры, а уже в зависимости от периферии функция может вернуть какое-то значение. То есть если бы был графический интерфейс, то все программирование свелось бы к выбору разных галочек и подпунктов меню. Отсюда на первых парах вам нужно просто запомнить, что любая программа для **arduino** должна начинаться именно с объявления этих двух функций. А теперь давайте разберемся для чего эти функции нужны. Первая с названием **setup()** нужна для начальной конфигурации программы. Данная функция вызывается один раз при первом старте программы и на протяжении всего периода работы основной программы больше не вызывается. Зачем так сделано? Дело в том что чаще всего при первом старте программы, необходимо проинициализировать периферию. Если это делать в основном цикле, то процессор будет выполнять не нужную работу. Поэтому нужно пока запомнить, что код написанный в теле функции **setup()** выполнится один раз. А вот основная программа, которая будет выполнятся на протяжении всей работы микроконтроллера, записывается в тело функции **loop()**. Давайте подведем итоги. Для начальной настройки и выполнении разово некого кода заполняем тело функции **setup()**. Для работы основной программы заполняем тело функции **loop()**. От слов к делу. Для того чтобы помигать светодиодом, нам необходимо посмотреть где он подключен. На плате arduino UNO что у меня, он подключен к **13** ножке выводов **DIGITAL**. А теперь смотрите. Сначала в функции **setup()** мы настроим ножку, а затем в функции **loop()** будем включат и выключать светодиод. Как это делать. Цифровая ножка микроконтроллера может выполнять всего два действия, либо выводить значение **1** или **0**, либо читать значение переданное на ножку тоже **1** или **0**. В данный момент, так как нам нужно зажигать светодиод, то вывод должен быть настроен как выход. Пишем в теле функции setup *(тело любой функции обременяется фигурными скобками)* следующую строку.



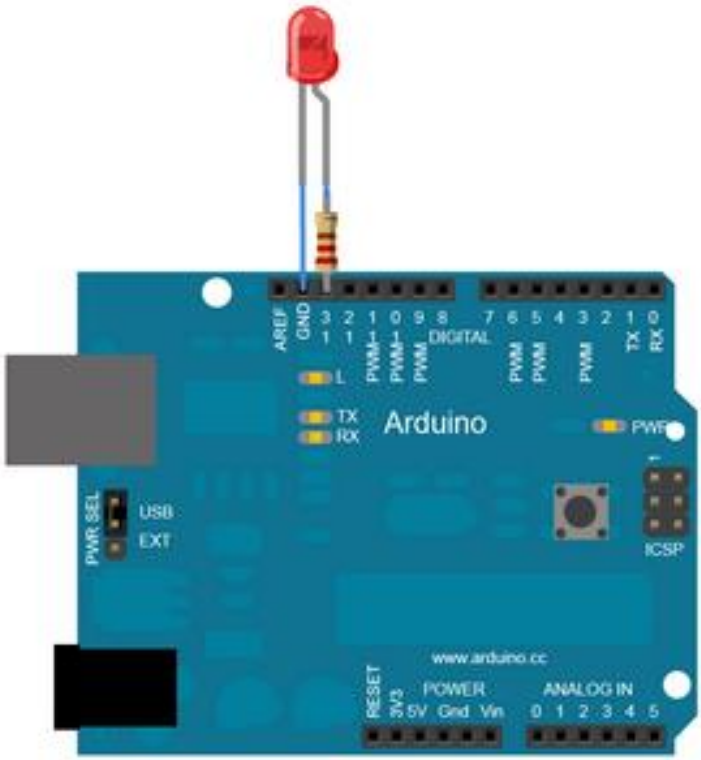
Что собственно делает эта строчка. Задача функции **pinMode()** является настройка пина на вход или на выход. Для того чтобы данная функция поняла что ей делать, она принимает два аргумента. Первый аргумент отвечает за номер пина. В нашем случае это **13**, а второй аргумент *(все аргументы передаваемые функциям пишутся через запятую)* это направление. В нашем случае это выход **OUTPUT**. в конце функции как можно заметить стоит точка с запятой. Данная точка с запятой говорит о том что текущая коменда закончилась. Следует запомнить как таблицу умножения, что каждая команда ОБЯЗТЕЛЬНО должна заканчиваться точкой с запятой. Вот таким образом мы настроили вывод. Теперь переходим к телу функции **loop()** и напишем четыре строки.



Что эти строчки делают. Первая **digitalWrite(13, HIGH);** это функция которая выводит в пин некое значение. Для того чтобы она понимала какое значение и куда ей выводить мы снова передаем два аргумента. Первый это номер пина, а второй **HIGH** то есть **1**. Как только эта функция отработает, на ножке **13** появится логическая единица *(логическая единица всегда равна напряжению питания МК. Чаще всего это 5 вольт или 3,3 вольта)*. Следующая строка **delay(1000);** это пауза. Данной функции в качестве аргумента передается значение времени в миллисекундах. В нашем случае это **1000** мс или **1** секунда. Теперь смотрите что происходит. Сначала функция **digitalWrite(13, HIGH);** выводит на ножку логическую единицу. Затем функция **delay(1000);** останавливает работу программы на **1** секунду *(она не останавливает в прямом смысле, а просто заставляет процессор таптаться на одном месте ничего не делая 1 секунду)*, а затем выполняется функция **digitalWrite(13, LOW);**. Как не трудно догадаться она выводит на ножку логический ноль. А затем снова останавливается на **1** секунду. Так как функция **loop()** является циклической, то после паузы программа возвращается к началу тела функции **loop()** и все начинается с начала. Вот и все, наша первая программа написана. Давайте ее загрузим в нашу **arduino**. Для этого нажмите на кнопку **"Вгрузить"**.



Если программа не сохранена, то перед загрузкой вам будет предложено ее сохранить. С учетом того что это первая пробная программа, то ее можно не сохранять и отказаться. Если все же захочется оставить на память, то сохраняйте в любом удобном для вас месте. После сохранения или отказа, программа загрузится в **arduino** и вы увидите как замигает светодиод. Ниже я для наглядности прикрепил видео работы программы.



Для связи с внешними элементами в контроллере Arduino UNO существуют 14 цифровых выводов. Каждый вывод может быть определен программой как вход или выход.

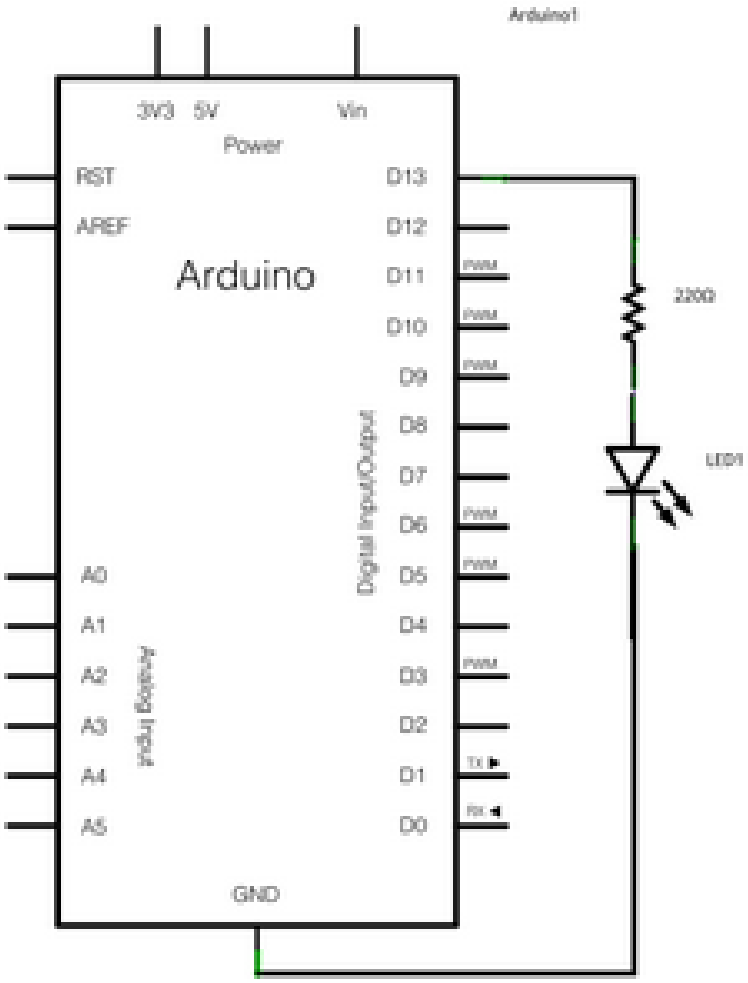
У цифрового выхода есть только два состояния высокое и низкое. Высокое состояние соответствует напряжению на выходе порядка 5 В, низкое состояние – 0 В. Выход допускает подключение нагрузки с током до 40 мА.

Когда вывод определен как вход, считав его состояние, можно определить уровень напряжения на входе. При напряжении близком к 5 В (реально более 3 В) будет считано высокое состояние, соответствующее константе HIGH. При напряжении близком к 0 (менее 1,5 В) будет считано низкое состояние, или константа LOW.

Светодиод мы должны подключить к выводу, определив его как выход, а кнопка подключается к выводу с режимом вход.

Светодиод подключается через резистор, ограничивающий ток. Вот типичная схема.

Собрать нижеприведенную схему:



Пример

```
int ledPin = 13;                  // Светодиод, подключенный к вход/выходу 13
void setup()
{
  pinMode(ledPin, OUTPUT);       // устанавливает режим работы - выход
}

void loop()
{
  digitalWrite(ledPin, HIGH);    // включает светодиод
  delay(1000);                    // ждет секунду
  digitalWrite(ledPin, LOW);     // выключает светодиод
  delay(1000);                    // ждет секунду
}
```

Директива #define.

Во всех примерах для функций ввода/вывода мы указывали аргумент pin, определяющий номер вывода, в виде конкретного числа - константы. Мы помнили, что константа 12 это номер вывода кнопки, а 13 – номер вывода светодиода. Гораздо удобнее работать с символьными именами. Для этого в языке С существует директива, связывающая идентификаторы с константами, выражениями.

Директива #define определяет идентификатор и последовательность символов, которая подставляется вместо идентификатора, каждый раз, когда он встречается в тексте программы.

В общем виде она выглядит так:

#define имя последовательность_символов

Если в наших программах мы напишем:

#define LED_PIN 13 // номер вывода светодиода равен 13

то каждый раз, когда в программе встретится имя LED_PIN, при трансляции вместо него будет подставлены символы 13. Функция включения светодиода выглядит так:

digitalWrite(LED_PIN, HIGH);

Окончательный вариант программы с использованием #define.

Резистор рассчитывается по формуле I = Uвыхода – Uпадения на светодиоде / R.

Uвыхода = 5 В, Uпадения на светодиоде можно принять равным 1,5 В (более точно указывается в справочнике). Получается, то в нашей схеме ток через светодиод задан на уровне 10 мА.

Можете выбрать любой вывод, но я предлагаю для простоты соединений использовать светодиод, установленный на плате. Тот самый, который мигал в первом тестовом примере. Он подключен к цифровому выводу 13. В этом случае дополнительный светодиод к плате подключать не надо.

Кнопку подключаем к любому другому выводу, например, 12. Аппаратная часть схемы подключения кнопки должна обеспечивать уровни напряжений 0 В при нажатой кнопке и 5 В при свободной. Это можно сделать простой схемой.

ЗАДАНИЕ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Сгенерировать сигнал SOS с помощью светодиода
2. Сгенерировать сигнал Вашего имени с помощью светодиода
3. С помощью трёх светодиодов симитировать работу светофора

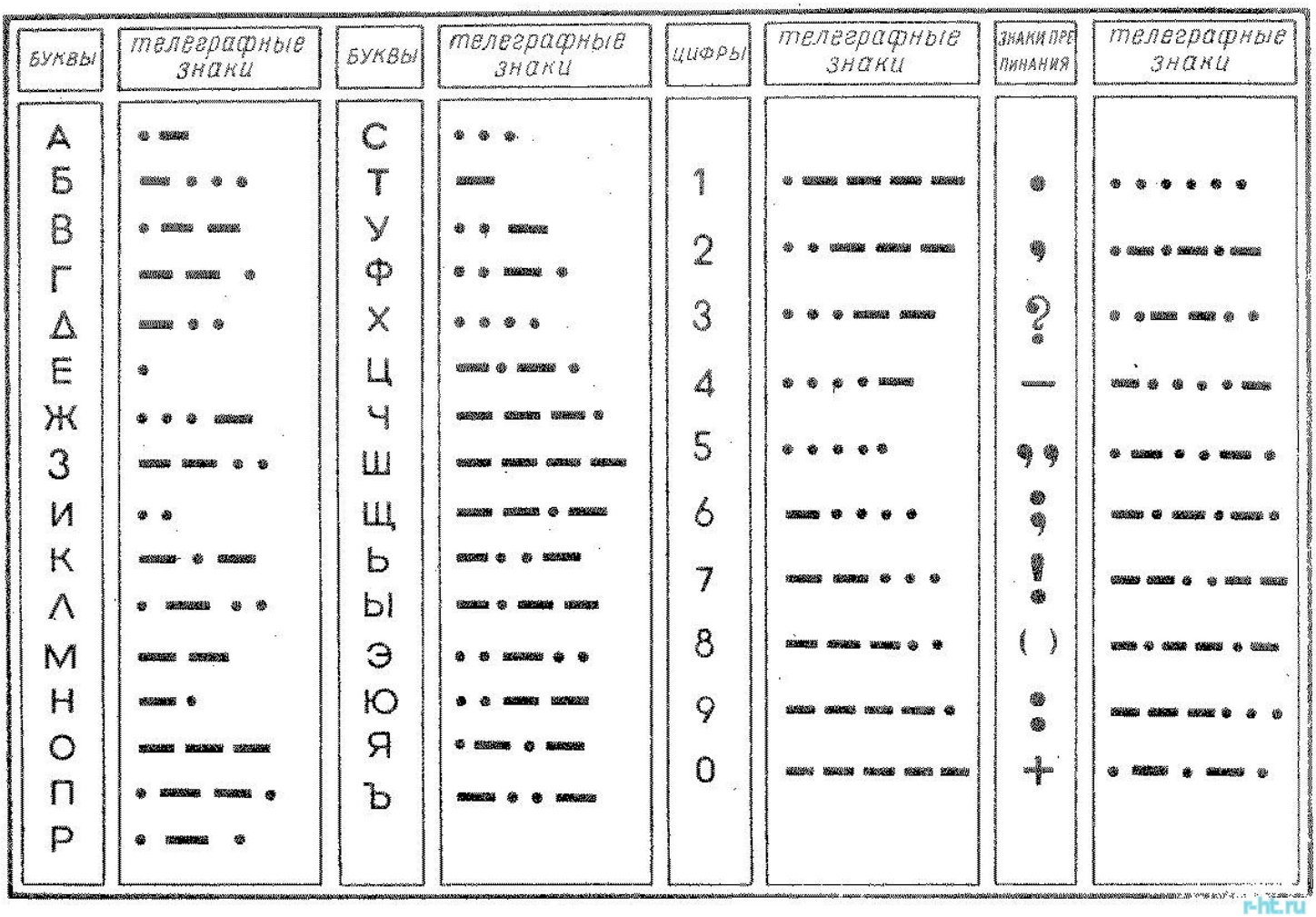
ФРАЗЫ, ЗАШИФРОВАННЫЕ АЗБУКОЙ МОРЗЕ

В азбуке Морзе буквы обозначаются точками и тире. Точка – короткий сигнал.
Тире – длинный, эквивалентный трем точкам.

БЛГ	Благодарю	-••• •-••--•	ДСВ	До свидания	-•• ••• •--
ЗДР	Здравствуйте	--••-•••-•	СПБ	Спасибо	••• •--• -•••
SOS		••• ––– •••	88	Любовь и поцелуй *(в шутку передают оператору-женщине)*	---•• ---••

Для ускорения радиообмена используются аббревиатуры – «Q-коды»

QRR	Вас понял	--•-•-• •-•	QRO	Увеличьте мощность передатчика	--•-•-••---
QRQ	Передавайте быстрее	--•- •-• --•-	QRL	Я занят, прошу не мешать	--•-•-• •-•••
QBS	У вас антенна залеплена птичьим пометом?	--•--••• •••	QEW	Плохо слышу вас, уши забиты	--•-• •--
QRC	Осторожно, в эфире болтун	--•- •-• -•-•			


Буквы
Телеграфные знаки
Буквы
Телеграфные знаки
Цифры
Телеграфные знаки
Знаки препинания
Телеграфные знаки
А Б В Г Д Е Ж З И К Л М Н О П Р
С Т У Ф Х Ц Ч Ш Щ Ь Ы Э Ю Я Ъ
1 2 3 4 5 6 7 8 9 0
( )