

Управляем работой светодиодов

Рассмотрим более подробно скетч, который мы заранее подготовили. Для этого откроем [файл 1_1_1.ino](#) (нужно скачать архив и разархивировать его).

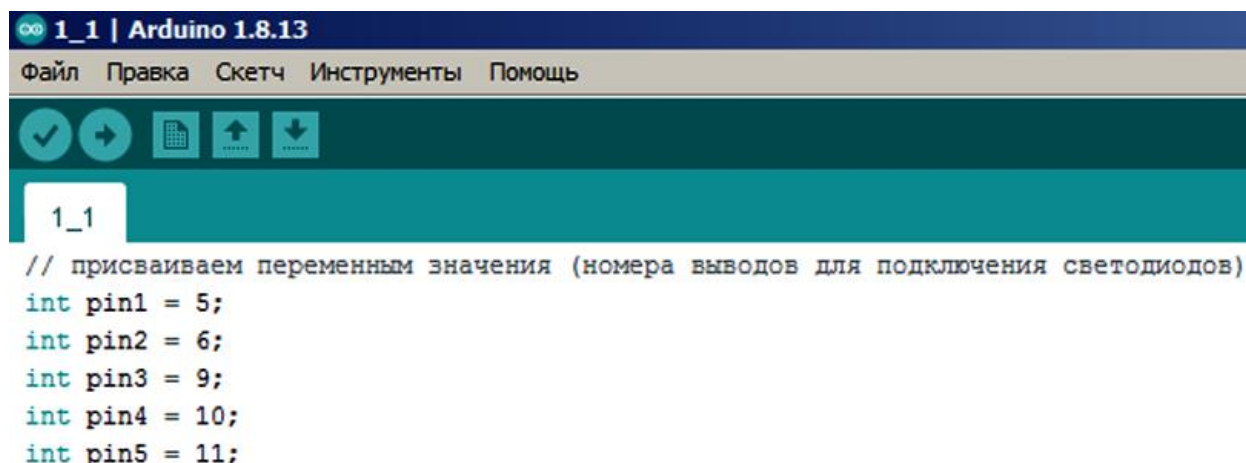
В окне среды программирования Arduino IDE мы видим текст программы с комментариями.

Но, прежде чем мы приступим к подробному анализу кода, посмотрим, как он работает для этого нажимаем на кнопку **Проверить (V)** или **Скетч – Проверить/Компилировать**, затем на кнопку **Загрузить (→)** или **Скетч – Загрузить**.

Наблюдаем мигание светодиодов в заданной последовательности на представленном видео.

Рассмотрим подробно код программы.

Введем 5 переменных, соответствующих выбранным нами выводам для подключения светодиодов. Присваиваем переменным pin1 – pin5 значения (номера выводов для подключения светодиодов: 5, 6, 9, 10, 11).



```
1_1
// присваиваем переменным значения (номера выводов для подключения светодиодов)
int pin1 = 5;
int pin2 = 6;
int pin3 = 9;
int pin4 = 10;
int pin5 = 11;
```

В блоке **void setup()** функция **pinMode()** — определяет режим работы указанных выводов на выход. Её достаточно выполнить один раз. Назначаем выводы 5, 6, 9, 10, 11 (переменные pin1 – pin5) работать в режиме ВЫХОД (OUTPUT), т.е. в режиме источника напряжения, поскольку в данном случае микроконтроллер будет управлять светодиодом.

```
void setup()
{
  /* указываются команды для предварительной настройки микроконтроллера,
  которые он выполнит только один раз в момент подключения платы к источнику питания
  или сразу после перезагрузки */

  /* настраиваем выводы подключения светодиодов на работу в режиме выхода (OUTPUT),
  т.е. в режиме источника напряжения*/
  pinMode(pin1, OUTPUT);
  pinMode(pin2, OUTPUT);
  pinMode(pin3, OUTPUT);
  pinMode(pin4, OUTPUT);
  pinMode(pin5, OUTPUT);
}
```

Обратите внимание, что каждой открывающейся фигурной скобке { всегда соответствует закрывающаяся }. Фигурные скобки обозначают границы логически завершенного фрагмента кода.

Каждую строку программы внутри цикла необходимо завершать символом «;».

Также полезно снабжать код программы комментариями. Они могут быть, как однострочными, которые начинаются //, так и многострочными – заключаются между /* */.

Рассмотрим функции в бесконечном цикле **void loop()**

Функция `digitalWrite(pin, value)` не возвращает никакого значения и принимает два параметра:

- `pin` — номер цифрового порта, на который мы отправляем сигнал
- `value` — значение, которое мы отправляем на порт. Для цифровых портов значением может быть HIGH (высокое, единица) или LOW (низкое, ноль).

Константы: INPUT, OUTPUT, LOW, HIGH, пишутся заглавными буквами, иначе компилятор их не распознает и выдаст ошибку.

Рассмотрим на примере фрагмента нашей программы.

Функция **digitalWrite()** определяет, какой сигнал подается на указанный вывод. Установим значение сигнала на выводах для переменных `pin1`, `pin2`, `pin4`, `pin5`, как «0» (**LOW**) (напряжение 0 В) и светодиоды, подключенные к этим выводам платы, погаснут. А для переменной `pin3` установим значение сигнала на выводе как «1» (**HIGH**) (напряжение 5 В) – светодиод загорится.

Функция **delay()** останавливает выполнение программы на 500 миллисекунд (0,5 с), т.е. в течение этого времени четыре светодиода выключены, а 1 включен и никаких действий больше не происходит.

```
void loop()
{
  /* указываются команды, которые будут бесконечно выполняться,
  пока плата подключена к источнику питания.
  Микроконтроллер будет выполнять код, дойдёт до последней строки
  и снова начнёт с первой («loop» с англ. — петля, цикл) */

  /* выключаем светодиоды, подавая на вывод логический 0 (LOW),
  величина напряжения равна 0 В */
  digitalWrite(pin1, LOW);
  digitalWrite(pin2, LOW);
  /* включаем светодиод, подавая на вывод логическую 1 (HIGH),
  величина напряжения равна 5 В */
  digitalWrite(pin3, HIGH);
  digitalWrite(pin4, LOW);
  digitalWrite(pin5, LOW);
  /* пауза 0,5 с (500 мс), т.е. остановка программы на 0,5 с (500 мс),
  после чего работа программы продолжается*/
  delay(500);
}
```

После паузы 0,5 с выполняется следующий фрагмент программы, в результате которого светодиоды, подключенные к выводам pin1 и pin5 по-прежнему выключены, светодиод, подключенный к выводу pin3 тоже гаснет, но загораются светодиоды, подключенные к выводам pin2 и pin4.

```
digitalWrite (pin1, LOW);  
digitalWrite (pin2, HIGH);  
digitalWrite (pin3, LOW);  
digitalWrite (pin4, HIGH);  
digitalWrite (pin5, LOW);  
delay(500);
```

После очередной паузы 0,5 с помимо светодиода, подключенного к выводу pin3 гаснут светодиоды, подключенные к выводам pin2 и pin4, а зажигаются светодиоды, подключенные к выводам pin1 и pin5.

```
digitalWrite (pin1, HIGH);  
digitalWrite (pin2, LOW);  
digitalWrite (pin3, LOW);  
digitalWrite (pin4, LOW);  
digitalWrite (pin5, HIGH);  
delay(500);
```

Через 0,5 с выполняется следующий (последний в этом цикле) фрагмент программы, в результате которого все светодиоды гаснут. Через 1 с. процедура loop выполняется снова (с первой строки цикла).

```
digitalWrite (pin1, LOW);  
digitalWrite (pin2, LOW);  
digitalWrite (pin3, LOW);  
digitalWrite (pin4, LOW);  
digitalWrite (pin5, LOW);  
/* пауза 1 с (1000 мс), т.е. остановка программы на 1 с (1000 мс),  
   после чего процедура loop выполняется снова (с первой строки цикла) */  
delay(1000);  
}
```

Таким образом, цикл **void loop()** повторяется бесконечно, пока плата Ардуино подключена к источнику питания.