

# ЁЛОЧНАЯ МИГАЛКА НА ARDUINO КАК СРЕДСТВО ОТ БОЯЗНИ МИКРОКОНТРОЛЛЕРОВ

При программировании ARDUINO UNO питается от USB-порта компьютера. Такое же питание возможно и при работе, например, от ЗУ для сотового телефона.

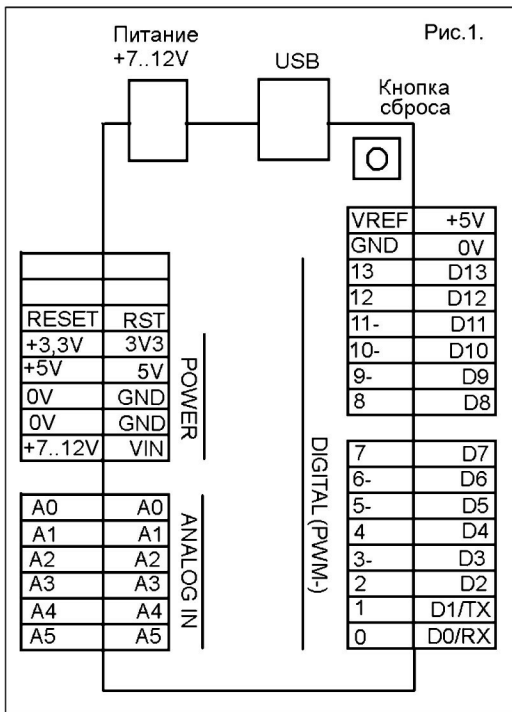
Будучи сам человеком пенсионного возраста, знаю многих коллег по канифольному дыму, категорически отказывающихся даже пытаться что-то сделать на микроконтроллере. Аргумент обычно один, - «Это для молодежи, а мы старики...»

Впрочем, не так страшен черт, как его... Если вы уже освоили персональный компьютер, хотя бы на начальном уровне, можете открыть файл, закрыть, сохранить, написать письмо и т.д., то «ARDUINO» вам поможет.

Здесь будем рассматривать ARDUINO UNO. Это небольшая печатная плата, на которой расположен микроконтроллер ATMEGA328, а так же вся его «обвязка», необходимая для его работы, включая USB-программатор и источник питания. Стоимость ARDUINO UNO лежит в пределах от 200 до 1000 рублей, в зависимости от места продажи. Конечно, самое дешевое на радиорынке и китайском интернет-посылторге «Aliexpress». Бывают и наборы - конструкторы, в которых кроме собственно ARDUINO UNO есть платы с различными датчиками, реле и прочим.

На рисунке 1 схематически показана плата ARDUINO UNO. В таком положении как она нарисована, - наверху два разъема. Один для подключения источника питания. Напряжения питания микроконтроллера 5V, поэтому напряжение с этого разъема поступает на 5-вольтовый стабилизатор имеющийся на плате. Рядом сверху разъем USB для подключения к компьютеру. Через него и происходит программирование. Кабель как для принтера. Затем кнопка сброса.

Слева два разъема. На разъем «POWER» выведены цепи питания и сброса.



Ниже разъем «ANALOG IN», - порты аналоговые входы. Их шесть.

Справа два разъема под общим названием «DIGITAL (RWM-)». Контакты 0 и 1 (D0, D1) это приемник и передатчик для связи. Остальные. От D2 по D13 - цифровые входы/выходы. Интересно то, что порты, отмеченные (-), такие как D3, D5, D9, D10, D11 могут работать и как выходы с широтно-импульсной модуляцией, широту импульсов на которых можно программно регулировать 255-ю степенями.

Ну и два последних контакта - GND и +5V (напряжение с выхода встроенного стабилизатора 5V).

Еще есть на плате светодиоды. Один служит для индикации подачи питания и горит всегда, два других индицируют работу приема и передачи. Они мигают, например, при записи программы, поступающей от компьютера. И еще один светодиод - он подключен к порту D13.

Для работы с ARDUINO UNO нужен персональный компьютер с установленной программой ARDUINO. Программу раздают бесплатно на официальном сайте [www.arduino.ru](http://www.arduino.ru). Зайдите на этот сайт, и почитайте там, затем перейдите по выделенной синим цветом ссылке: «*бесплатного скачивания*». Попадете на англоязычную страницу. Выберите там «*WINDOWS installer*», а затем нажимайте мышкой на надпись: «*JUST DOWNLOAD*». После этого на ваш компьютер загрузится файл *arduino-1.6.11-windows.exe* или другой версии (версии обновляются).

Подключите плату ARDUINO UNO к компьютеру через разъем USB (при этом внешний блок питания не нужен).

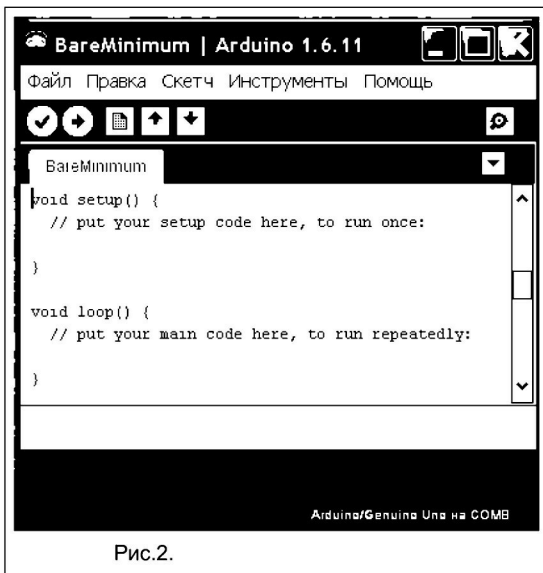


Рис.2.

Затем нужно запустить файл *arduino-1.6.11-windows.exe* точно так же, как при установке любой другой программы. Согласиться с условиями лицензии, и далее соглашаться со всем, что предлагают. В конце установки WINDOWS

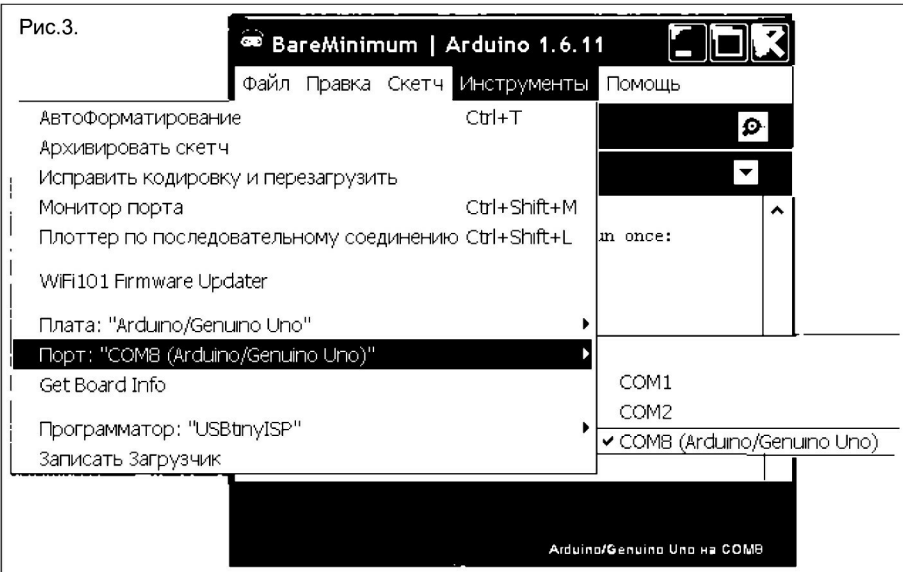


Рис.3.

два-три раза ргнется на то, что, мол, программный продукт они не тестировали. При этом нужно нажать «*все равно продолжить*».

Что касается аппаратных требований, могу сказать на личном опыте, программа одинаково хорошо работает и на старом «Пентиуме» с 1-ядерным процессором 1 ГГц, ОЗУ 256 МБ и WINDOWS XP, и на новом 4-ядерном планшете Prestigio MultiPad PMP881 с WINDOWS 8. Главное чтобы был хотя бы один разъем USB.

Для товарищей, по разным причинам не имеющих доступа к интернету, хочу заметить, что загрузочный файл *arduino-1.6.11-windows.exe* есть и на диске DVD#22 (на котором архивы журнала «Радиоинженер»), но только на выпущенном после сентября этого года.

Запустите программу (на рабочем столе ярлык «ARDUINO»). Через некоторое время появится окошко, показанное на рисунке 2. Выберите «Инструменты» и затем отметьте галочкой порт COM, на котором у вас оказалась «ARDUINO» (рис.3), в моем случае это порт COM8.

Посмотрите что в строке «Плата», там должно быть «*Arduino/Genuino Uno*», если что-то другое перейдите на строку «Плата» и в выпадающем меню выберите «*Arduino/Genuino Uno*».

Хочу заметить, что плата ARDUINO UNO

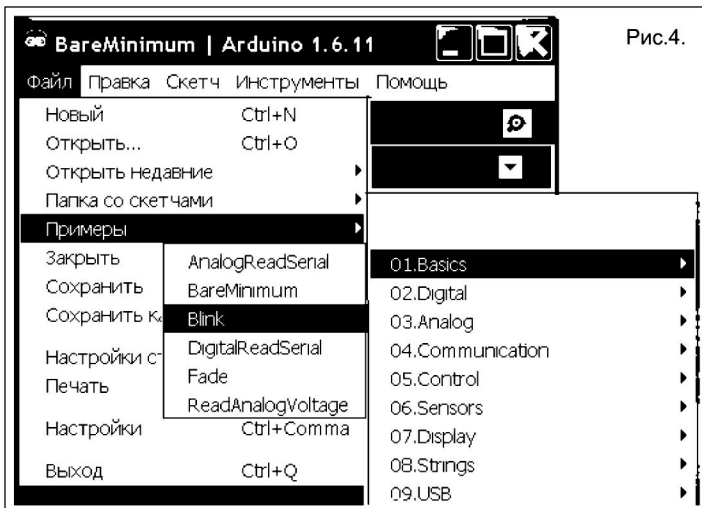


Рис.4.

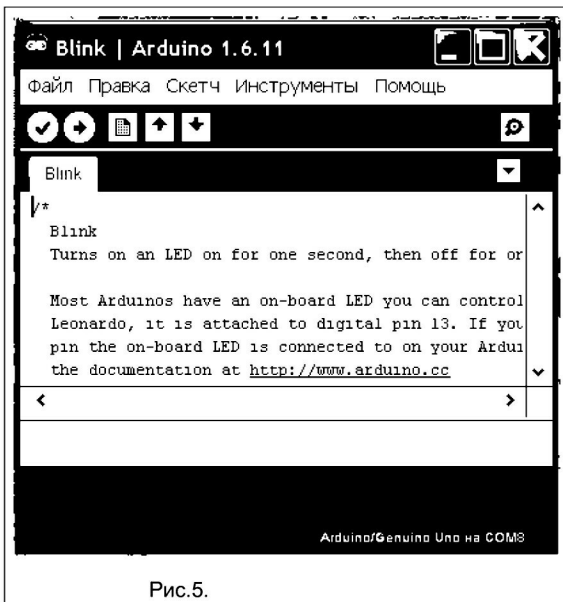


Рис.5.

идет уже сразу с демонстрационной программой, которая мигает светодиодом, подключенным к порту D13. А в программе есть много примеров. Можно посмотреть программу вызывающую это мигание. Для этого перейдите: *Файл - Примеры -*

01.Basics - Blink (рис.4). Появится окошко с программой (рис.5). В нем будет текст программы Blink.

То, что там написано в строках от /\* до \*/ не столь важно. Это информация о программе, о её разработчике и т.д. Важно вот что:

```
// the setup function runs once
when you press reset or power
the board
void setup() {
  // initialize digital pin 13 as an
  output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again
forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on
(HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(13, LOW); // turn the LED off
by making the voltage LOW
  delay(1000); // wait for a second
}
```

Здесь дано указание на то, с каким портом работаем:

```
// initialize digital pin 13 as an output.
pinMode(13, OUTPUT);
```

Порт 13, и то, что он работает как выход (OUTPUT).

Далее идет сама программа:

```
// the loop function runs over and over again
forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on
(HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(13, LOW); // turn the LED off
by making the voltage LOW
  delay(1000); // wait for a second
}
```

Здесь указывается состояние порта 13 ноль (LOW) или единица (HIGH), а так же сколько времени длится каждое состояние. Вот здесь:

*digitalWrite(13, HIGH);* - установить порт 13 в состояние HIGH (логическая единица).  
*delay(1000);* - продолжительность 1 секунда (значение в миллисекундах, поэтому 1000 = 1 сек.)  
*digitalWrite(13, LOW);* - установить порт 13 в состояние LOW (логический ноль).  
*delay(1000);* - продолжительность 1 сек .

То есть, светодиод, подключенный к порту 13 мигает с полупериодом в одну секунду.

Теперь давайте изменим параметры мигания этого светодиода. Для этого нужно данный пример сохранить. В окне (рис.5) выбираем «Файл - Сохранить как» и сохраняем его куда-нибудь, куда предложит программа или по своему усмотрению.

Затем открываем сохраненный файл. Для начала изменим периодичность мигания. Для этого изменим значения в строках:

```
delay(1000);
например, на: delay(500);
```

Затем, перейдите «Скетч - Загрузка». Через некоторое время на плате замигают светодиоды TX и RX, и потом светодиод, подключенный к порту 13 будет мигать вдвое быстрее.

УРРРААА! ЗАРАБОТАЛО!!!!

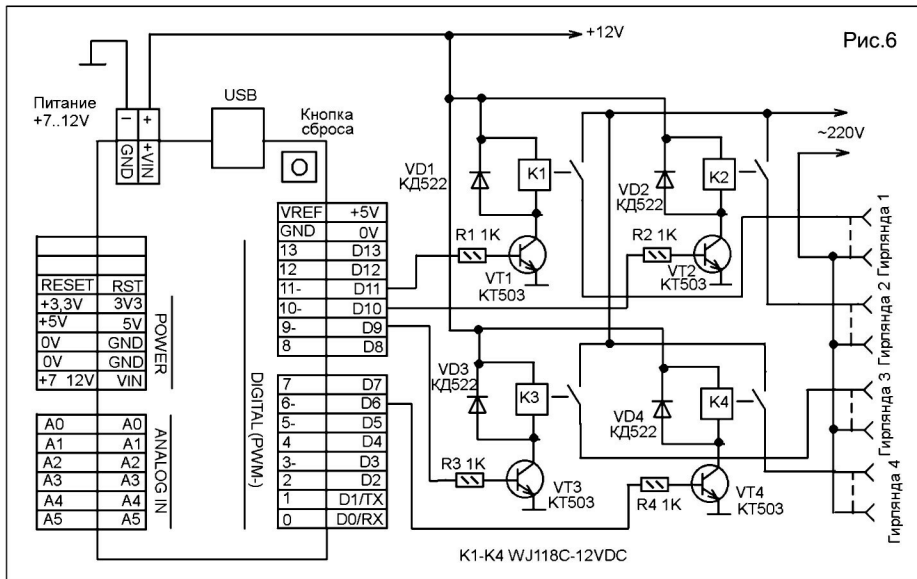
Кстати, все, что в программе за знаком «//» это комментарии, пояснения, и на работу программы они не влияют.

### Переключатель ёлочных гирлянд.

Сначала рассмотрим схему, она на рис.6. Для управления гирляндами взят самый простой вариант с транзисторными ключами на VT1-VT4 и реле K1-K4. Можно сделать ключи на тиристорах, симисторах, оптопарах, мощных полевых транзисторах и т.д.

Схема предназначена для управления четырьмя гирляндами. Для управления выбраны порты D11, D10, D9, D6. Впрочем, количество гирлянд можно увеличить до 11-ти, используя так же порты D2, D3, D4, D5, D7, D8, D12. Но здесь сделано только на четыре гирлянды.

Текст программы приводится ниже. Программу писал я, потому и комментарии на русском языке.



/\*  
Blink4  
\*/

// задание выходов

```
void setup() {
  pinMode(11, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(6, OUTPUT);
}
```

// рабочий цикл

```
void loop() {
  digitalWrite(11, HIGH); // на 11 единица
  digitalWrite(10, LOW); // на 10 ноль
  digitalWrite(9, LOW); // на 9 ноль
  digitalWrite(6, LOW); // на 6 ноль
  delay(1000); // время шага 1 сек.
  digitalWrite(11, HIGH); // на 11 единица
  digitalWrite(10, HIGH); // на 10 единица
  digitalWrite(9, LOW); // на 9 ноль
  digitalWrite(6, LOW); // на 6 ноль
  delay(1000); // время шага 1 сек.
  digitalWrite(11, HIGH); // на 11 единица
  digitalWrite(10, HIGH); // на 10 единица
  digitalWrite(9, HIGH); // на 9 единица
  digitalWrite(6, LOW); // на 6 ноль
}
```

```
delay(1000); // время шага 1 сек.
digitalWrite(11, HIGH); // на 11 единица
digitalWrite(10, HIGH); // на 10 единица
digitalWrite(9, HIGH); // на 9 единица
digitalWrite(6, HIGH); // на 6 единица
delay(1000); // время шага 1 сек.
digitalWrite(11, LOW); // на 11 ноль
digitalWrite(10, HIGH); // на 10 единица
digitalWrite(9, HIGH); // на 9 единица
digitalWrite(6, HIGH); // на 6 единица
delay(1000); // время шага 1 сек.
digitalWrite(11, LOW); // на 11 ноль
digitalWrite(10, LOW); // на 10 ноль
digitalWrite(9, HIGH); // на 9 единица
digitalWrite(6, HIGH); // на 6 единица
delay(1000); // время шага 1 сек.
digitalWrite(11, LOW); // на 11 ноль
digitalWrite(10, LOW); // на 10 ноль
digitalWrite(9, LOW); // на 9 ноль
digitalWrite(6, HIGH); // на 6 единица
delay(1000); // время шага 1 сек.
}
```

То есть, скрупулезно, по шагам записываем состояние каждого выхода. И длительность каждого шага.

Шагов может быть великое множество. Просто подставляете далее и далее вот такое «четверостишие»:

```
digitalWrite(11, LOW);  
digitalWrite(10, LOW);  
digitalWrite(9, LOW);  
digitalWrite(6, HIGH);  
delay(1000);
```

в котором соответственно вашему желанию записываете для каждого вывода «LOW» или «HIGH», и продолжительность шага. Как видите, все очень просто. Комментарии (то, что после «//» писать не обязательно).

Главное, не забыть, что в конце программы на последней строчке должна быть фигурная скобка.

После того как программа введена, загрузить её в микроконтроллер («Скетч - Загрузка»).

Чтобы было проще, можно взять имеющуюся программу Blink (которую сохраняли) и переделать.

Вместо:

```
pinMode(13, OUTPUT);
```

написать:

```
pinMode(11, OUTPUT);  
pinMode(10, OUTPUT);  
pinMode(9, OUTPUT);  
pinMode(6, OUTPUT);
```

а далее, вместо:

```
digitalWrite(13, HIGH); // turn the LED on  
(HIGH is the voltage level)  
delay(1000); // wait for a second  
digitalWrite(13, LOW); // turn the LED off  
by making the voltage LOW  
delay(1000); // wait for a second
```

написать целую «поэму» «четверостишей» вроде:

```
digitalWrite(11, LOW);  
digitalWrite(10, LOW);  
digitalWrite(9, LOW);  
digitalWrite(6, HIGH);  
delay(1000);
```

При этом не потерять в конце фигурную скобку, не потерять точки с запятыми, и не наделать других ошибок, вроде лишних и недостающих пробелов. При том, за

количество пробелов в строке после знака « ; » можно не опасаться, а после « // » вообще может быть что угодно.

Если выходы сделать на мощных транзисторных ключах и управлять низковольтными гирляндами постоянного тока, то можно задавать еще и яркость. Для этого *digitalWrite* меняем на *analogWrite* и значение задаем числом от 0 до 255, например, так :

```
analogWrite (11, 255);  
analogWrite (10, 100);  
analogWrite (9, 30);  
analogWrite (6, 0);  
delay(1000);
```

Почему программа может не заработать?

1. Если программа не записывается в микроконтроллер, это скорее всего связано с тем, что выбран не тот порт. После установки и первого запуска программы почему-то в моем случае был указан порт COM1. Но плата ARDUINO UNO установлена была на порт COM8. Нужно как показано на рисунке 3, проверить на каком порту плата, и выбрать из выпадающего списка именно тот порт, возле которого написано «*Arduino/Genuino Uno*». У меня это COM8, у вас может быть другой COM.

2. Неправильно выбрана плата. Нужно точно так же, как показано на рис.3, посмотреть какая плата. Посмотрите в строке «Плата», там должно быть «*Arduino/Genuino Uno*», если что-то другое перейдите на строку «Плата» и в выпадающем меню выберите «*Arduino/Genuino Uno*».

3. Нарушен синтаксис программы. Где-то потеряли точку с запятой, где-то лишний пробел, где-то пробела не хватает, где-то букву пропустили. Это мелочи, но очень важные. В таком случае, ниже на черном фоне красными буквами будет указано на ошибку, а в тексте программы будет выделено место, где эта ошибка.

*Каравкин В.*