**#1 Эксперимент “SOS-маячок”. Управление светодиодом**

В этом эксперименте мы будем управлять частотой мигания светодиода, и даже создадим прототип аварийного маячка, который будет передавать сигнал бедствия SOS, используя код азбуки Морзе.

**Перед выполнением эксперимента прочтите:**

* [Понятие электричества](http://eschool.by/courses/arduino-%D0%BD%D0%B0%D1%87%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9-%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C/lessons/%D0%BF%D0%BE%D0%BD%D1%8F%D1%82%D0%B8%D0%B5-%D1%8D%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%B8%D1%87%D0%B5%D1%81%D1%82%D0%B2%D0%B0/);
* [Основные законы электричества](http://eschool.by/courses/arduino-%D0%BD%D0%B0%D1%87%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9-%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C/lessons/%D0%BE%D1%81%D0%BD%D0%BE%D0%B2%D0%BD%D1%8B%D0%B5-%D0%B7%D0%B0%D0%BA%D0%BE%D0%BD%D1%8B-%D1%8D%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%B8%D1%87%D0%B5%D1%81%D1%82%D0%B2%D0%B0/);
* [Управление электричеством](http://eschool.by/courses/arduino-%D0%BD%D0%B0%D1%87%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9-%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C/lessons/%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5-%D1%8D%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%B8%D1%87%D0%B5%D1%81%D1%82%D0%B2%D0%BE%D0%BC/);
* [Входы и выходы Arduino. Цифровой и аналоговый сигналы](http://eschool.by/courses/arduino-%D0%BD%D0%B0%D1%87%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9-%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C/lessons/%D0%B2%D1%85%D0%BE%D0%B4%D1%8B-%D0%B8-%D0%B2%D1%8B%D1%85%D0%BE%D0%B4%D1%8B-arduino-%D1%86%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%BE%D0%B9-%D0%B8-%D0%B0%D0%BD%D0%B0%D0%BB%D0%BE%D0%B3%D0%BE%D0%B2%D1%8B%D0%B9/);
* [Принципиальные схемы](http://eschool.by/courses/arduino-%D0%BD%D0%B0%D1%87%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9-%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C/lessons/%D0%BF%D1%80%D0%B8%D0%BD%D1%86%D0%B8%D0%BF%D0%B8%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5-%D1%81%D1%85%D0%B5%D0%BC%D1%8B/);
* [Быстрая сборка схем](http://eschool.by/courses/arduino-%D0%BD%D0%B0%D1%87%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9-%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C/lessons/%D0%B1%D1%8B%D1%81%D1%82%D1%80%D0%B0%D1%8F-%D1%81%D0%B1%D0%BE%D1%80%D0%BA%D0%B0-%D1%81%D1%85%D0%B5%D0%BC/);
* [Резистор](http://eschool.by/courses/arduino-%D0%BD%D0%B0%D1%87%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9-%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C/lessons/%D1%80%D0%B5%D0%B7%D0%B8%D1%81%D1%82%D0%BE%D1%80/);
* [Светодиод.](http://eschool.by/courses/arduino-%D0%BD%D0%B0%D1%87%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9-%D1%83%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C/lessons/%D1%81%D0%B2%D0%B5%D1%82%D0%BE%D0%B4%D0%B8%D0%BE%D0%B4/)

 **СБОРКА УСТРОЙСТВА НА МАКЕТНОЙ ПЛАТЕ**

**Для сборки нашего первого устройства нам понадобятся следующие компоненты:**

* Arduino Uno;
* USB-кабель для подключения к компьютеру;
* Беспаечная макетная плата;
* 1 светодиод любого цвета;
* 1 резистор номиналом 220 Ом;
* 2 соединительных провода.

**Соберите устройство по следующей схеме**:

|  |
| --- |
| Схема устройства на макетной плате |
| http://eschool.by/wp-content/uploads/2017/10/Ardu_lessons_starter_1_1.jpg |
| Принципиальная схема устройства |
| http://eschool.by/wp-content/uploads/2017/10/Ardu_lessons_starter_1_2.jpg |

**Обратите внимание**: длинная ножка светодиода должна быть подключена к 13 пину Arduino, так как именно с этого пина мы будем подавать на светодиод высокий сигнал +5 В (HIGH). Если вы перепутаете полярность светодиода и подключите его длинной ножкой к пину GND, он может выйти из строя!

Запустите Arduino IDE, зайдите в меню [инструменты] и выберите пункт [ArduBlock]. Теперь пришло время написать наш первый алгоритм, который будет управлять нашим светодиодом. Если мы подадим на 13-й пин высокий сигнал (HIGH), то он подаст на наш светодио напряжение в 5 В. Такое напряжение является через чур большим для светодиода, поэтому последовательно с ним, мы включили резистор номиналом 220 Ом. В итоге, напряжение на светодиоде с 5 В, упадёт до 2-2,3 В, что и требуется для штатной работы светодиода.

**ПРОГРАММИРОВАНИЕ**

Любая программа для Arduino имеет 2 основных раздела:

* Раздел “Установка”;
* Раздел “цикл”.

При запуске микроконтроллера, сначала однократно выполняются инструкции из раздела [Установка], а затем начинают выполняться инструкции из раздела [цикл]. При этом, инструкции, находящиеся в разделе [цикл] будут выполняться постоянно в зацикленном режиме. помните это. Первая задача:

Давайте заставим наш светодиод мигать – зажигаться на 1 секунду и гаснуть на 2 секунды. Таким образом, 1 цикл работы нашего светодиода будет занимать 3 секунды, 1 из которых он будет зажжён и потушен в течение оставшихся 2-х секунд.

Зайдите в окно ArduBlock и в разделе [Управление] найдите и перетащите на экран блок [Программа]:



Раздел [Установка], нам пока не понадобится. Весь алгоритм будет находится в разделе [цикл].

Зажигать светодиод будем с помощью блока [уст. цифровой порт] из раздела [Порты]. Перетащите данный блок в раздел [цикл] программы и настройте значения данного блока. Управлять будем 13-м цифровым пином, и пошлём на него высокий сигнал (HIGH):



Теперь можете нажать кнопку [Загрузить в Arduino]:



Программа может попросить сохранить файл с вашим алгоритмом, соглашайтесь на это, заодно вы можете создать папку в удобном для вас месте, где вы будете хранить ваши алгоритмы. После успешного сохранения, ваш код будет загружен в микроконтроллер и светодиод загорится! Давайте теперь потушим его. Для этого, добавим ещё один блок [уст. цифровой порт] к уже имеющемуся. Но, не забудьте поменять значение сигнала с высокого (HIGH) на низкий (LOW), чтобы отключить напряжение на пине:



Загрузите полученный алгоритм в Arduino. Обратите внимание: светодиод не погас! На самом деле, наш светодиод включается и выключается очень-очень быстро. На столько быстро, что наш глаз не успевает этого замечать, и видит постоянно включённый светодиод. Для того, чтобы светодиод начал мигать, необходимо задать паузу – остановить программу после включения и выключения светодиода. Если нам необходимо, чтобы светодиод горел 1 секунду, а затем был потушен в течении 2 секунд, то мы должны поставить паузы после наших команд в 1 и 2 секунды соответственно. Для этого мы будем использовать блок [задержка в миллисекундах] из раздела [Управление]. Давайте вставим блоки задержки после  команд включения и выключения светодиода:



Обратите внимание, блок задержки имеет параметр – время задержки в миллисекундах. Для того, чтобы получить задержку в 1 секунду, мы должны установить значение 1000 миллисекунд, а для задержки в 2 секунды – значение 2000 миллисекунд. Теперь, Arduino будет зажигать светодиод, ждать 1 секунду, затем тушить светодиод и ждать 2 секунды. Затем, всё повториться сначала. Загрузите алгоритм в Arduino.

Если вы всё сделали правильно, то светодиод начнёт мигать!

Видеоплеер

00:00

00:05

Давайте попробуем составить алгоритм посложнее.

Составим алгоритм, который заставит наш светодиод подавать сигнал SOS, используя код символов азбуки Морзе.

А вы знакомы с азбукой Морзе? В 1838 году Сэмюэль Морзе предложил кодировку, которая могла позволить передавать информацию по телеграфной линии. Особенность линии была в том, что она работала по цифровому принципу: на одном конце линии, человек замыкал контакт и подавал на линии высокий сигнал (HIGH), на другом конце линии, был подключён динамик, который преобразовывал электрический сигнал, посланный по линии в звуковой сигнал. По сути, замыкающий контакты человек мог лишь изменять время замыкания контактов и паузы между сигналами. Если вместо динамика к телеграфу подключить обычную лампочку, то она бы начала мигать в такт посылаемому сигналу, и человек на принимающей стороне смог бы расшифровать его. Для обозначения символов в азбуке Морзе используются **точки** • и **тире —**. При этом, договорились, что длительность тире будет равна длительности 3-х точек. Пауза между элементами одного символа составляет длительность 1 точки, а между отдельными буквами – 3 точки, а между словами – 7 точек. Ниже, представлены кодировка английского алфавита, представленная с помощью азбуки Морзе:



Таким образом, для того, чтобы закодировать сигнал бедствия SOS, необходимо подать следующую комбинацию знаков: • • •   **— — —   • • •**.

Попробуем перенести этот код в ArduBlock. Наша задача: включать и выключать светодиод, выдерживая нужные интервалы включения и выключения. Поскольку, за единицу длительности, в азбуке Морзе принята точка, то и длительность остальных символов будет зависеть от неё. Давайте зададим длительность точки в 300 миллисекунд (или 0,3 секунды). Составим схему нашего алгоритма (для удобства, паузы будем обозначать символом **×**):

вкл. светодиод (300 мс)  •

выкл. светодиод (300 мс) ×

вкл. светодиод (300 мс)  •

выкл. светодиод (300 мс)  ×

вкл. светодиод (300 мс)  •

выкл. светодиод (900 мс)  × – пауза между буквами

вкл. светодиод (900 мс) —

выкл. светодиод (300 мс) ×

вкл. светодиод (900 мс) —

выкл. светодиод (300 мс) ×

вкл. светодиод (900 мс) —

выкл. светодиод (900 мс) × – пауза между буквами

вкл. светодиод (300 мс) •

выкл. светодиод (300 мс) ×

вкл. светодиод (300 мс) •

выкл. светодиод (300 мс) ×

вкл. светодиод (300 мс) •

выкл. светодиод (2100 мс) × – длинная пауза между словами

Алгоритм будем писать в разделе [цикл], таким образом наш сигнал “SOS” будет автоматически повторяться. Вот, что у нас должно получиться:



Наш алгоритм получился достаточно длинным и однообразным. На следующем занятии вы узнаете как сделать его значительно короче. Тем не менее наша программа работает. Если вы всё сделали правильно, то ваш светодиод должен мигать настоящий “SOS” на морзянке:

Видеоплеер

00:01

00:09

**ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОГО ВЫПОЛНЕНИЯ**

**Задание 1**:

Создайте алгоритм, который заставит светодиод мигать слово “HELLO” на морзянке.

**Задание 2\***:

Создайте алгоритм, в ходе которого, в начале работы, светодиод коротко мигает 3 раза, а затем передаёт слово “HELLO” на морзянке. *Подсказка: часть алгоритма нужно будет записать в разделе*[Установка]*блока*[Программа]*.*

Начало формы

ЗАКОНЧЕН

Конец формы