

1

Эксперимент 2.

Бегущий огонек на 8-ми светодиодах

В данном эксперименте мы создадим “бегущий огонек” на ленте из 8-ми светодиодах, научимся создавать собственные функции для сокращения кода скетча.

В эксперименте мы будем использовать следующие компоненты:

- Плата Arduino UNO – 1;
- Кабель USB;
- Плата прототипирования – 1;
- Светодиод – 8;
- Резистор 220 Ом – 8;
- Провода ММ – 5.

Переключатели на плате Arduino+WiFi установите следующим образом:

ПЕРЕКЛЮЧАТЕЛЬ						
1	2	3	4	5	6	7
OFF	OFF	ON	ON	OFF	OFF	OFF

Собираем схему согласно рис. 2.1

Приступим к написанию программы (скетча).

В процедуре `setup()` настроим режим работы контактов (пинов), к которым подключены светодиоды, как `OUTPUT` (выход)

```
void setup() {  
  pinMode(5, OUTPUT);  
  pinMode(6, OUTPUT);  
  pinMode(7, OUTPUT);  
  pinMode(8, OUTPUT);  
  pinMode(9, OUTPUT);  
}
```



Рис. 2.1. Схема соединений для «бегущего огня» на 8 светодиодах

Алгоритм основного цикла следующий:

1. “гасим” все 8 светодиодов;
 2. включаем первый светодиод;
 3. включаем в скетче продолжительную паузу, чтобы увидеть горение одного светодиода;
 4. повторяем шаги 1-3 для второго, третьего и далее до восьмого светодиода;
 5. далее возврат в начало цикла loop().
- Сокращенный код цикла loop() представлен в листинге 2.1.

Листинг 1.1

```
void loop() {  
  // выключить 8 светодиодов  
  digitalWrite(5, LOW);  
  digitalWrite(6, LOW);  
  digitalWrite(7, LOW);  
  digitalWrite(8, LOW);  
  digitalWrite(9, LOW);  
  digitalWrite(10, LOW);  
  digitalWrite(11, LOW);  
  digitalWrite(12, LOW);  
  // включить 1 светодиод
```

```
digitalWrite(5, HIGH);  
// пауза 1000 мсек  
delay(1000);  
// выключить 8 светодиодов  
...  
// включить 2 светодиод  
digitalWrite(6, HIGH);  
// пауза  
delay(1000);  
// выключить 8 светодиодов  
...  
// включить 3 светодиод  
digitalWrite(7, HIGH);  
// пауза  
delay(1000);  
.....  
.....  
// выключить 8 светодиодов  
...  
// включить 8 светодиод  
digitalWrite(12, HIGH);  
// пауза  
delay(1000);
```

```
}
```

Скачать данный скетч целиком можно на сайте Arduino-kit по ссылке

https://arduino-kit.ru/scetches/exp_02_01.

Загрузим данный скетч на плату Arduino. Вы должны наблюдать бугущий огонек, перемещающийся по светодиодам с 1 по 8.

При том, что скетч работает верно, у него есть большой недостаток – он очень громоздкий. Рассмотрим, как можно код значительно уменьшить с помощью функций. Функция – это именованная последовательность операций. Причиной создания функции является необходимость выполнять одинаковое действие несколько раз. В цикле loop() восемь раз встречается совершенно одинаковый код для выключения 8 светодиодов. Создадим функцию ledoff() (выключить светодиоды). Функция создается за скобками функций setup() и loop(). Создание функции:

```
void ledsoff() {  
    // код функции  
    digitalWrite(5, LOW);  
    digitalWrite(6, LOW);  
    digitalWrite(7, LOW);  
    digitalWrite(8, LOW);  
    digitalWrite(9, LOW);  
    digitalWrite(10, LOW);  
    digitalWrite(11, LOW);  
    digitalWrite(12, LOW);  
}
```

```
}
```

И тогда цикл loop() нашего скетча сократится следующим образом (листинг 2.2)

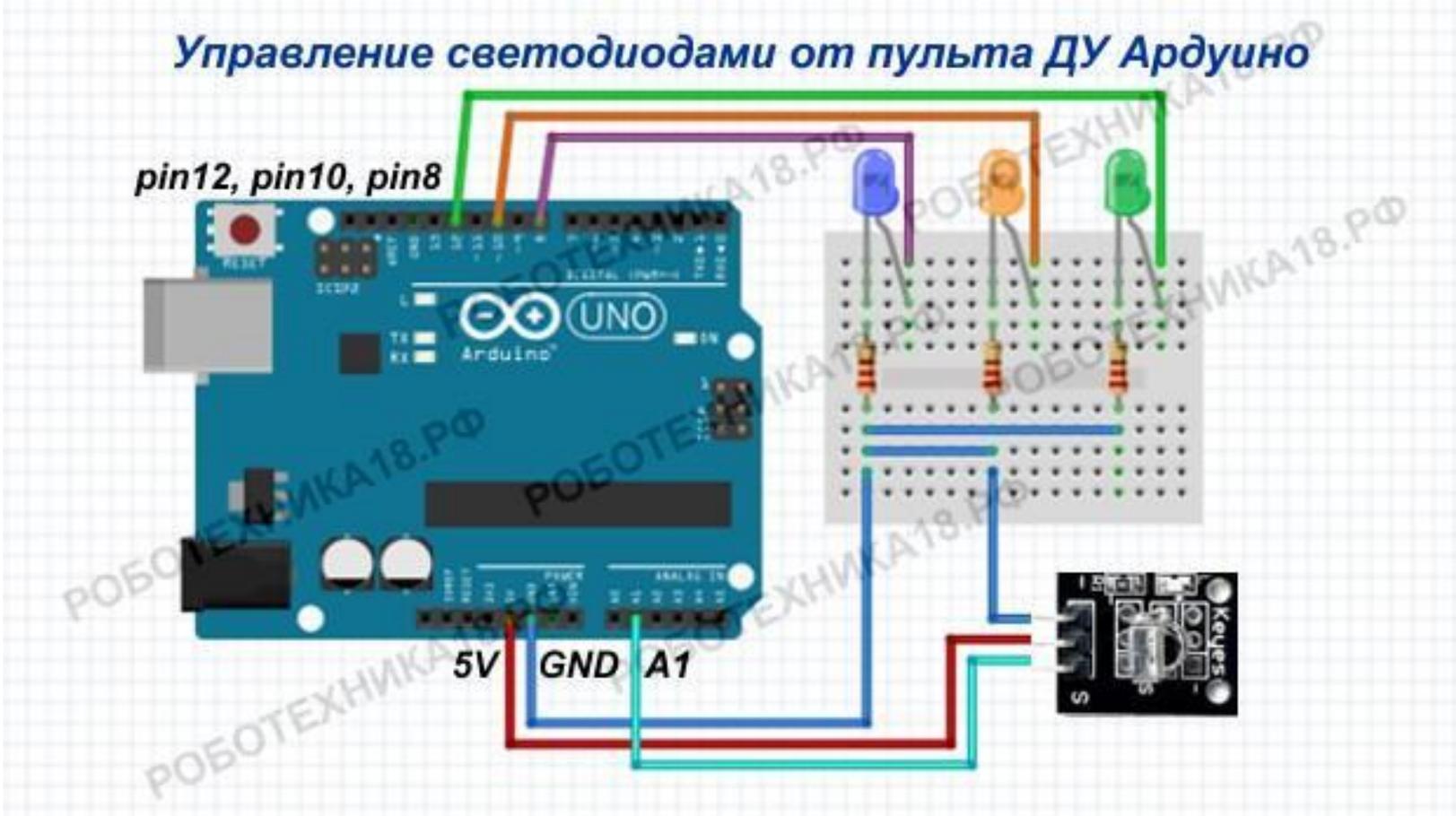
2

Включение и выключение светодиода от пульта Ардуино



- Управление светодиодом от пульта дистанционного управления Ардуино — мини проект, на основе которого можно сделать дистанционное управление освещением в помещении или [елочную гирлянду](#) на адресной светодиодной ленте. Рассмотрим схему сборки данного проекта, подключение светодиодов к плате, а также программу для включения и выключения светодиодов Arduino с помощью пульта ДУ.

Ардуино управление светодиодам от ИК пульта



```
#include "IRremote.h"
```

```
IRrecv irrecv(A1);  
decode_results results;
```

```
// пины подключения светодиодов
```

```
#define G 12
```

```
#define R 10
```

```
#define B 8
```

```
// коды сигнала кнопок пульта ду
```

```
#define B1 16718055
```

```
#define B2 16718055
```

```
#define B3 16718055
```

```
#define B4 16718055
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
  irrecv.enableIRIn();
```

```
  pinMode(G, OUTPUT);
```

→ ↻ 🔒 роботехника18.рф/светодиод-пульт/

```
pinMode(R, OUTPUT);  
pinMode(B, OUTPUT);
```

```
}
```

```
void loop() {
```

```
  if (irrecv.decode(&results)) {  
    Serial.println(results.value);
```

```
    if (results.value == B1) { digitalWrite(G, HIGH); }
```

```
    if (results.value == B2) { digitalWrite(R, HIGH); }
```

```
    if (results.value == B3) { digitalWrite(B, HIGH); }
```

```
    if (results.value == B4) {
```

```
      digitalWrite(G, LOW); digitalWrite(R, LOW); digitalWrite(B, LOW);
```

```
    }
```

```
    irrecv.resume();
```

```
  }
```

```
}
```

Пояснения к коду:

- 1** нажмите кнопку на пульте дистанционного управления, вы увидите код на последовательном мониторе, нажмите кнопку несколько раз, чтобы убедиться, что у вас правильный код;
- 2** коды указаны в программе с помощью инструкции #define.

3

Тема: Умный дом



Информация

- «**Умный дом**» (англ. smart home) — это комплекс устройств, которые самостоятельно выполняют рутинные задачи по дому для удобства человека. Системы поддержания микроклимата, умные лампочки, сигнализация, системы управления мультимедиа устройствами — это всё умный дом. А термин «**умное здание**» (англ. smart house) относится к организации целым строением, например, многоквартирным домом.

Что такое технология «Умный дом»

- Что такое технология «Умный дом»
- Умный дом – это система автоматизации управления устройствами в доме или квартире. Первые появления данной технологии зафиксированы в середине прошлого века. Сегодня smart home получает всё большее распространение среди потребителей, желающих избавиться себя от постоянного контроля за техническим состоянием помещения. Рассмотрим технологию умного дома, ее сильные и слабые стороны.
- Умный дом представляет собой совокупность трех направлений:
 - повышение уровня безопасности;
 - улучшение комфорта жизни человека;
 - эффективное потребление ресурсов.

Что входит в экосистему технологии «Умный дом» (smart home)

- **Контроллер** (хаб, шлюз) — устройство, выполняющее функцию «мозгового центра». Контроллер можно запрограммировать на приём и обработку сигналов, управление различными устройствами по проводам или дистанционно.
- **Датчики** (сенсоры) — устройства для сбора информации о состоянии помещения (сведения о температуре и влажности воздуха, присутствии людей в комнатах, протечках воды, состоянии освещённости в помещении и многом другом).
- **Актуаторы** (исполнители) — это агрегаты, которые под управлением контроллера выполняют различные действия — включают свет, управляют питанием техники, открывают и закрывают шторы, перекрывают воду и делают многое другое.
- **Устройства управления** — при их помощи владелец узнает о состоянии системы, может настроить режим работы или отдать команду для устройств. В современных системах органом управления все чаще становится смартфон с установленным приложением.

История развития систем Smart Home

- Первые попытки домашней автоматизации в современном понимании появились в середине XX века. Наиболее известными были «Дом с кнопками» (1950 г) американского инженера Матиаса и компьютер Echo IV (1966 г) американского инженера Сазерленда, который мог регулировать работу домашней климатической техники, включать и выключать некоторые приборы и распечатывать списки покупок.
- Официальной датой рождения системы Smart Home считается 1978 год. Шотландская компания Pico Electronics разработала в 1975 году первый стандарт управления домашними устройствами — X10. Создатели предусмотрели беспроводное управление устройствами на радиочастоте 433 МГц. Система позволяла включать приборы, менять яркость освещения в комнатах и получать данные о состоянии приборов.
- Начало XXI века стало временем бурного развития Smart Home. Возможности технологии значительно расширились — из крайне ограниченного набора функций, системы превращаются в многофункциональные и масштабные. Интеллектом может обладать не только дом, но и целое здание, аэропорт или спортивный стадион. Многие специалисты характеризуют рынок систем Smart Home, как наиболее перспективный.

Как работает «Умный дом» (Smart Home)



4

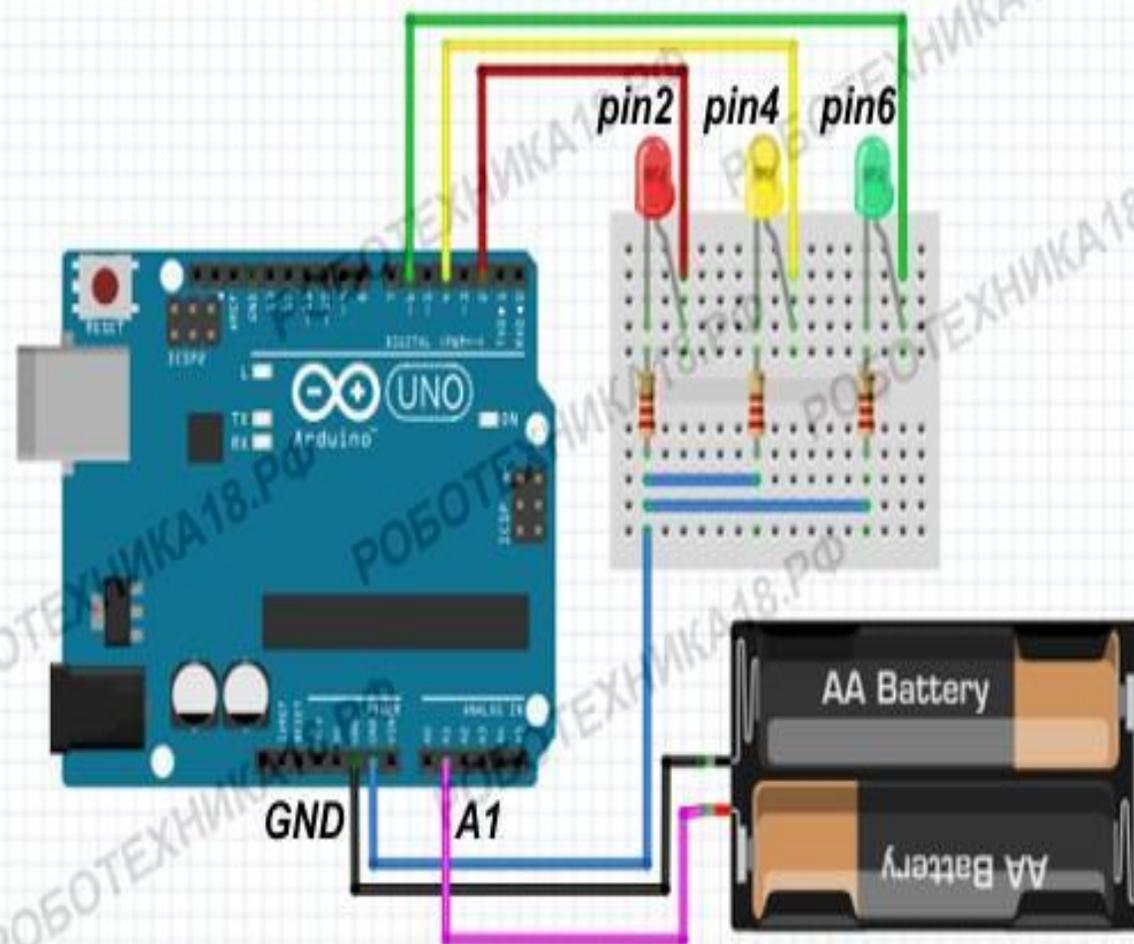
Тема; Тестер батареек на Ардуино с LCD 1602



Информация

- Тестер батареек и аккумуляторов на Arduino Uno — один из самых простых проектов для которого потребуются минимум деталей. Представим два варианта данного устройства — на светодиодах для индикации заряда батарей, а также с использованием LCD дисплея для вывода информации. Подробная схема сборки проекта и программа для тестера на микроконтроллере Ардуино представлена далее на странице.
- Первым делом необходимо предупредить, что на аналоговые порты плат Ардуино не следует подавать напряжение более 5 Вольт. Согласно [описанию Arduino Uno](#), данного производителем, при большем напряжении микроконтроллер может выйти из строя. Если вам необходимо будет проверить заряд аккумулятора с напряжением более 5 Вольт, то можно использовать делитель напряжения на резисторах.

Тестер батереек на Ардуино со светодиодами



- **Для этого проекта потребуются:**
- Arduino Uno / Arduino Nano / Arduino Mega;
- три светодиода и три резистора;
- диод;
- LCD дисплей;
- макетная плата;
- провода «папа-папа», «папа-мама».

```
интересные проекты x Arduino тестер батар x Новая вкладка x Arduino управление x
роботехника18.рф/тестер-батареек-на-ардуино/

#define RED 2 // красный светодиод
#define YELLOW 4 // желтый светодиод
#define GREEN 6 // зеленый светодиод
#define TESTER A1 // порт для подключения батарейки

int data;
float voltage;

void setup() {
  Serial.begin(9600); // запускаем серийный монитор порта

  pinMode(RED, OUTPUT);
  pinMode(YELLOW, OUTPUT);
  pinMode(GREEN, OUTPUT);

  pinMode(TESTER, INPUT);
  analogWrite(TESTER, LOW);
}

void loop() {
  data = analogRead(A1); // считываем данные с порта A1
  voltage = data * 0.0048; // переводим значения в вольты

  Serial.print(voltage); // выводим напряжение на монитор
```

```
интересные проекты x Arduino тестер батар x Новая вкладка x Arduino управление x
роботехника18.рф/тестер-батареек-на-ардуино/

Serial.print(voltage); // выводим напряжение на монитор
Serial.println(" Volts");

if ( voltage >= 1.5 ) {
  digitalWrite(GREEN, HIGH);
  digitalWrite(YELLOW, LOW);
  digitalWrite(RED, LOW);
}

else if ( voltage < 1.5 && voltage > 1.3 ) {
  digitalWrite(GREEN, LOW);
  digitalWrite(YELLOW, HIGH);
  digitalWrite(RED, LOW);
}

else if ( voltage <= 1.3 ) {
  digitalWrite(GREEN, LOW);
  digitalWrite(YELLOW, LOW);
  digitalWrite(RED, HIGH);
}

delay(500);
}
```

