

Практическая работа 11

Кнопочный переключатель

В этом эксперименте мы добавляем порцию яркости светодиоиду одной кнопкой и убавляем другой.

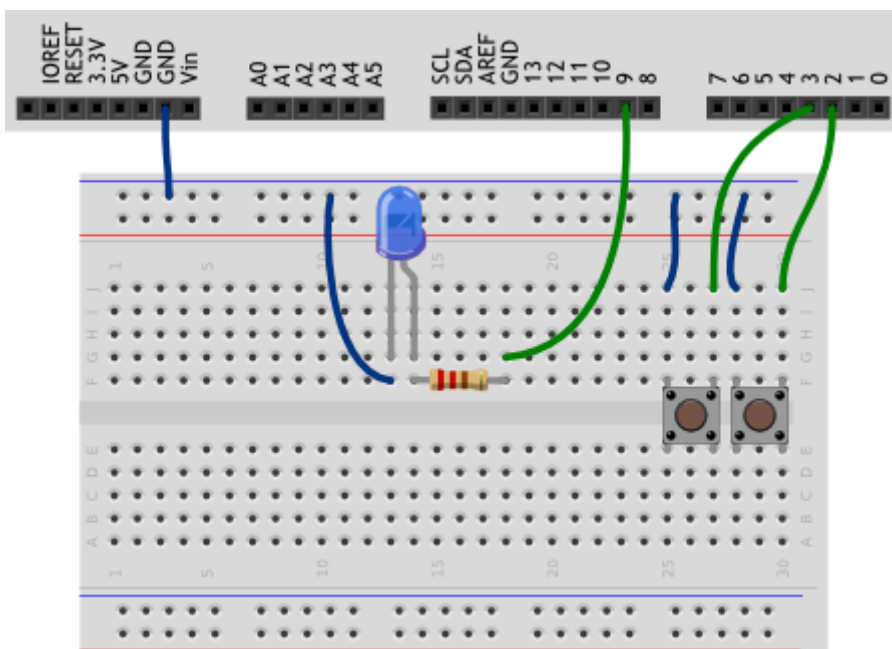
Задание 1. Ответить на вопросы

1. Что такое ШИМ?
2. К каким пинам платы можно подключать устройства, поддерживающие ШИМ?

Задание 2. Список деталей для эксперимента

1. 1 плата Arduino Uno
2. 1 беспаячная макетная плата
3. 2 тактовых кнопки
4. 1 резистор 220 Ом
5. 1 светодиод
6. 7 проводов «папа-папа»

Схема на макетной плате:



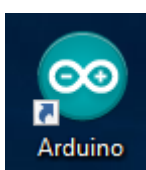
1. Зарисуйте принципиальную схему установки.

Обратите внимание

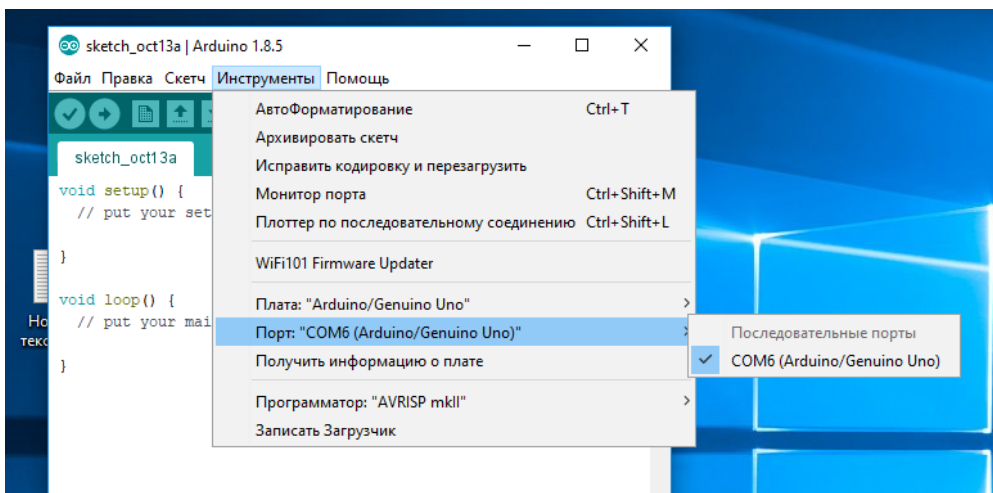
✓ Если вы переделываете схему из схемы предыдущего эксперимента, обратите внимание, что на этот раз нам нужно подключить светодиод к порту, поддерживающему ШИМ.

Задание 3. Программирование микроконтроллера

1. Запустите приложение



2. Убедитесь, что выбран нужный порт



3. Наберите в редакторе кода следующий код программы:

```
Файл Правка Скetch Инструменты Помощь
Файл Правка Скetch Инструменты Помощь
sketch_nov20a §
#define PLUS_BUTTON_PIN    2
#define MINUS_BUTTON_PIN   3
#define LED_PIN            9

int brightness = 100;
boolean plusUp = true;
boolean minusUp = true;

void setup()
{
  pinMode(LED_PIN, OUTPUT);
  pinMode(PLUS_BUTTON_PIN, INPUT_PULLUP);
  pinMode(MINUS_BUTTON_PIN, INPUT_PULLUP);
}

void loop()
{
  analogWrite(LED_PIN, brightness);
  // реагируем на нажатия с помощью функции, написанной нами
  plusUp = handleClick(PLUS_BUTTON_PIN, plusUp, +35);
  minusUp = handleClick(MINUS_BUTTON_PIN, minusUp, -35);
}
// Собственная функция с 3 параметрами: номером пина с кнопкой
// (buttonPin), состоянием до проверки (wasUp) и градацией
// яркости при клике на кнопку (delta). функция возвращает
// (англ. return) обратно новое, текущее состояние кнопки
boolean handleClick(int buttonPin, boolean wasUp, int delta)
{
  boolean isUp = digitalRead(buttonPin);
  if (wasUp && !isUp) {
    delay(10);
    isUp = digitalRead(buttonPin);
    // если был клик, меняем яркость в пределах от 0 до 255
    if (!isUp)
      brightness = constrain(brightness + delta, 0, 255);
  }
  return isUp; // возвращаем значение обратно, в вызывающий код
}

38 Arduino/Genuino Uno на COM7
```

Пояснения к коду

- ✓ Мы можем пользоваться не только встроенными функциями, но и создавать собственные. Это обоснованно, когда нам нужно повторять одни и те же действия в разных местах кода или, например, нужно выполнять одни и те же действия над разными данными, как в данном случае: обработать сигнал с цифровых портов 2 и 3.
- ✓ Определять собственные функции можно в любом месте кода вне кода других функций. В нашем примере, мы определили функцию после loop.
- ✓ Чтобы определить собственную функцию, нам нужно:
- ✓ Объявить, какой тип данных она будет возвращать. В нашем случае это boolean. Если функция только выполняет какие-то действия и не возвращает никакого значения, используйте ключевое слово void
- ✓ Назначить функции имя — идентификатор. Здесь действуют те же правила, что при именовании переменных и констант. Называть функции принято в том же стиле как Переменные.
- ✓ В круглых скобках перечислить передаваемые в функцию параметры, указав тип каждого. Это является объявлением переменных, видимых внутри вновь создаваемой функции, и только внутри нее. Например, если в данном эксперименте мы попробуем обратиться к wasUp или isUp из loop() получим от компилятора сообщение об ошибке. Точно так же, переменные, объявленные в loop, другим функциям не видны, но их значения можно передать в качестве параметров.
- ✓ Между парой фигурных скобой написать код, выполняемый функцией
- ✓ Если функция должна вернуть какое-то значение, с помощью ключевого слова return указать, какое значение возвращать. Это значение должно быть того типа, который мы объявили
- ✓ Так называемые глобальные переменные, т.е. переменные, к которым можно обратиться из любой функции, обычно объявляются в начале программы. В нашем случае — это brightness.

- ✓ Внутри созданной нами функции handleClick происходит всё то же самое, что в эксперименте «Кнопочный переключатель».
- ✓ Поскольку при шаге прироста яркости 35 не более чем через восемь нажатий подряд на одну из кнопок значение выражения `brightness + delta` выйдет за пределы интервала `[0, 255]`. С помощью функции `constrain` мы ограничиваем допустимые значения для переменной `brightness` указанными границами интервала.
- ✓ В выражении `plusUp = handleClick(PLUS_BUTTON_PIN, plusUp, +35)` мы обращаемся к переменной `plusUp` дважды. Поскольку `=` помещает значение правого операнда в левый, сначала вычисляется, что вернет `handleClick`. Поэтому когда мы передаем ей `plusUp` в качестве параметра, она имеет еще старое значение, вычисленное при прошлом вызове `handleClick`.
- ✓ Внутри `handleClick` мы вычисляем новое значение яркости светодиода и записываем его в глобальную переменную `brightness`, которая на каждой итерации `loop` просто передается в `analogWrite`.

Задание 4. Ответьте на следующие вопросы

1. Что необходимо для определения собственной функции?
2. Что означает ключевое слово `void`?
3. Как ведет себя программа при упоминании одной переменной с разных сторон от оператора присваивания `=`?

Задание 5. Самостоятельно измените существующую программу и схему

1. Доработайте код таким образом, чтобы шаг изменения яркости настраивался в одном месте.
2. Создайте еще одну функцию и переделайте код так, чтобы одна функция отвечала за отслеживание нажатий, а другая — за вычисление яркости светодиода и возвращала его в `analogWrite`.