

Работа с цифровым выходом

Задание 1. Blink (Мигание встроенным светодиодом)

Указания. Изучите устройство платы Arduino UNO, расположение и назначение её контактов, разъёмов, кнопок. Умейте показать на плате: разъём USB, разъём питания, кнопку RESET, микроконтроллер ATmega328, цифровые входов-выходов 0-13, аналоговые входы A0-A5, выводы 5B, GND, светодиоды ON, L, RX, TX. Умейте подключать плату Arduino UNO к компьютеру. Изучите общую структуру программы в среде Arduino IDE; назначение функций: void setup(), void loop(), порядок выполнения функций, порядок проверки и загрузки программы, сохранения и открытия скетча. Изучите правила работы с цифровыми выходами, функции pinMode(), digitalWrite(), delay().

Подключите плату Arduino UNO к компьютеру и запрограммируйте следующую функциональность:

- При включении питания встроенный светодиод (L) бесконечно мигает (включается и выключается) с частотой 1 Герц (1 раз в секунду).
- При программировании используйте инструкции digitalWrite() и delay().

Составьте принципиальную электрическую схему собранного устройства.

Задание 2. Blink (Мигание внешним светодиодом)

Указания. Изучите устройство светодиода, правила подключения светодиода в электрическую цепь, устройство макетной платы, подключение макетной платы к питанию и размещение элементов на макетной плате.

Создайте и запрограммируйте электрическую цепь, состоящую из микроконтроллера Arduino UNO, макетной платы, светодиода с ограничивающим резистором, обладающую следующей функциональностью:

- При включении питания светодиод бесконечно мигает (включается и выключается) с частотой 1 Герц (1 раз в секунду).
- При программировании используйте инструкции digitalWrite() и delay().

Составьте принципиальную электрическую схему собранного устройства.

Задание 3. Светофор

Создайте и запрограммируйте электрическую цепь, состоящую из микроконтроллера Arduino UNO, макетной платы, трёх светодиодов (зелёного, желтого, красного) с ограничивающими резисторами, обладающую следующей функциональностью:

- При включении питания вначале на 1 секунду загорается зелёный светодиод, потом на 1 секунду желтый, потом на 1 секунду красный. Данная последовательность включений бесконечно повторяется.

Составьте принципиальную электрическую схему собранного устройства.

Использование переменных в программе

Задание 4. Blink с ускорением

Указания. Изучите понятие переменной, типы переменных в среде разработки Arduino IDE, способы объявления и инициализации переменных, различия между глобальными и локальными переменными, арифметические операции с целыми и вещественными типами.

Создайте и запрограммируйте электрическую цепь, состоящую из микроконтроллера Arduino UNO, макетной платы, светодиода с ограничивающим резистором, обладающую следующей функциональностью:

- При включении питания светодиод бесконечно мигает (включается и выключается) с увеличением частоты. Т.е. мигает всё быстрее и быстрее. Время включения и выключения должно плавно уменьшаться от 1 секунды до нуля, каждый раз на 20%.
- При программировании используйте инструкции digitalWrite() и delay(), сохраняйте значение временной задержки в глобальной переменной типа int.

Вывод информации в консоль

Задание 5. Вывод информации в консоль. Serial.println()

Указания. Изучите возможность вывода информации в монитор порта с помощью инструкции Serial.print(), Serial.println().

Дополните программный код задания 2 выводом значения переменной (временной задержки) в консоль (монитор порта). Для вывода переменной в консоль используйте инструкцию Serial.println(). Перед выводом значения выведите подсказку ("d =").

Пример вывода:

d = 1000

d = 800

Использование условной инструкции в программе

Задание 6. Циклический Blink с ускорением

Указания. Изучите операции сравнения, булевский тип `bool`, синтаксис условной инструкции `if, if-else`.

Создайте и запрограммируйте электрическую цепь, состоящую из микроконтроллера Arduino UNO, макетной платы, светодиода с ограничивающим резистором, обладающую следующей функциональностью:

1. Светодиод мигает (включается и выключается) с увеличением частоты. Т.е. мигает всё быстрее и быстрее. Время включения и выключения должно плавно уменьшаться от 1 секунды до нуля, каждый раз на 20%.
2. Как только временная задержка станет равной нулю, установите задержку в 1 секунду и вернитесь к пункту 1.
3. Бесконечно повторяйте пункты 1 и 2.
4. При программировании функциональности используйте инструкцию `if`.

Задание 7*. Blink с ускорением и замедлением (дополнительное)

Создайте и запрограммируйте электрическую цепь, состоящую из микроконтроллера Arduino UNO, макетной платы, светодиода с ограничивающим резистором, обладающую следующей функциональностью:

1. При включении питания светодиод мигает (включается и выключается) с увеличением частоты. Т.е. мигает всё быстрее и быстрее. Время включения и выключения должно плавно уменьшаться от 1 секунды до 10 ms, каждый раз на 20%.
2. Как только временная задержка станет равной 10 ms, светодиод начинает мигать с уменьшением частоты (медленнее), каждый раз время включения и выключения увеличивается на 20%.
3. Как только временная задержка превысит 1 секунду, вернитесь к пункту 1.
4. Бесконечно повторяйте пункты 1, 2, 3.

Рекомендация: при программировании объявите две глобальные переменные `d` – временная задержка и `k` – коэффициент изменения задержки. Первоначально `d = 1000`, `k = 0.8`.

Работа с текущим временем

Задание 8. Вывод текущего времени. Функция `millis()`.

Указания. Изучите функцию `millis()`, тип возвращаемого ей значения.

Напишите программу для микроконтроллера, которая с интервалом 300мс выводит в консоль надпись «Current time: » и значение текущего времени (количество миллисекунд с момента начала выполнения текущей программы). Надпись «Current time: » и текущее время должны находиться в одной строке. Пример вывода:

Current time: 0

Current time: 300

Current time: 600

Задание 9. Мигание светодиодом без использования `delay()`

Указания. Изучите способ программирования событий по триггеру (события ожидают своего времени выполнения).

Создайте и запрограммируйте электрическую цепь, состоящую из микроконтроллера Arduino UNO, макетной платы, светодиода с ограничивающим резистором, обладающую следующей функциональностью:

- При включении питания светодиод бесконечно мигает (включается и выключается) с частотой 1 Герц (1 раз в секунду).
- В программном коде нельзя использовать инструкцию `delay()`.
- Временную задержку необходимо реализовать с использованием переменной-триггера (событие ожидает своего времени).

Задание 10*. Мигание тремя светодиодами без `delay()`

Создайте и запрограммируйте электрическую цепь, состоящую из микроконтроллера Arduino UNO, макетной платы, трёх светодиодов с ограничивающими резисторами, обладающую следующей функциональностью:

- При включении питания первый светодиод бесконечно мигает (включается и выключается) с частотой 1 Герца (1 раз в секунду), второй светодиод бесконечно мигает (включается и выключается) с частотой 3 Герца (3 раза в секунду), третий светодиод бесконечно мигает (включается и выключается) с частотой 5 Герц (5 раз в секунду)
- В программном коде нельзя использовать инструкцию `delay()`, временную задержку необходимо реализовать с использованием переменных – триггеров (события ожидают своего времени).

Использование цикла for

Задание 11. Цикл for.

Указания. Изучите синтаксис цикла for.

Используя цикл FOR, напишите программу для микроконтроллера, которая выводит в консоль пять строчек:

Первая строка - числа от 1 до 20 в одну строчку через пробел.

Вторая строка – числа от 20 до 5 в одну строчку через пробел.

Третья строка – числа кратные 5 от 0 до 100

Четвертая строка – квадраты целых чисел от 0 до 10

Пятая строка – все числа от 0 до 100 которые делятся либо на 3, либо на 5, но не делятся на 6.

Пример вывода:

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5
0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100
0 1 4 9 16 25 36 49 64 81 100
3 5 9 10 15 20 21 25 27 33 35 39 40 45 50 51 55 57 63 65 69 70...
```

Задание 12. Заданное число миганий (for).

Создайте и запрограммируйте электрическую цепь, состоящую из микроконтроллера Arduino UNO, макетной платы, светодиода с ограничивающим резистором, обладающую следующей функциональностью:

- При включении питания светодиод должен мигнуть ровно 7 раз с частотой 2 Гц (2 раза в секунду)
- При программировании используйте цикл FOR

Использование цикла while

Задание 13. Цикл while.

Указания. Изучите синтаксис цикла while.

Выполните задание 11, используя цикл while вместо for.

Задание 14 Заданное число миганий (while).

Выполните задание 8, используя цикл while вместо for.

Работа с цифровым входом

Задание 15. Подключение тактовой кнопки

Указания. Изучите способ подключения тактовой кнопки к микроконтроллеру с помощью внешнего подтягивающего резистора (режим INPUT), инструкцию digitalRead().

Создайте электрическую цепь, состоящую из микроконтроллера Arduino UNO, макетной платы, тактовой кнопки, подтягивающего резистора. Запрограммируйте постоянное чтение состояния тактовой кнопки в переменную k (k = digitalRead(pin)) и вывода значения переменной в монитор порта. При нажатой кнопке в консоль должно выводиться 1, при отжатой 0.

Режим вывода, к которому подключается кнопка, установите – INPUT.

Составьте принципиальную электрическую схему, собранного устройства.

Задание 16. Тактовая кнопка и светодиод

Создайте и запрограммируйте электрическую цепь, состоящую из микроконтроллера Arduino UNO, макетной платы, светодиода с ограничивающим резистором, тактовой кнопки, подтягивающего резистора, обладающую следующей функциональностью:

- Если тактовая кнопка нажата – светодиод горит, если отжата – светодиод выключен.

Составьте принципиальную электрическую схему, собранного устройства.

Задание 17. Подсчет количества нажатий на кнопку

Указания. Изучите эффектдребзга контактов. Как можно подавитьдребзг контактов? Чтожелательно сделать после каждого чтения состояния кнопки?

Создайте и запрограммируйте электрическую цепь, состоящую из микроконтроллера Arduino UNO, макетной платы, тактовой кнопки, подтягивающего резистора (при необходимости), обладающую следующей функциональностью:

- При включении питания в монитор порта выводится число «0» (значение счетчика нажатий).
- Каждый раз при нажатии на тактовую кнопку (сразу после нажатия) происходит увеличение счётчика и вывод значений счётчика в консоль.

Пример вывода:

```
0
1
2
```

Задание 18*. Подключение тактовой кнопки без внешнего подтягивающего резистора.

Указания. Изучите способ подключения тактовой кнопки к микроконтроллеру с помощью внешнего подтягивающего резистора (режим INPUT_PULLUP). Как считывать в переменную не инвертируемое значение?

Выполните задания 15 не используя внешний подтягивающий резистор. Режим вывода, к которому подключается кнопка, установите – INPUT_PULLUP.

Составьте принципиальную электрическую схему, собранного устройства.

Задание 19. Переключение состояния светодиода

Создайте и запрограммируйте электрическую цепь, состоящую из микроконтроллера Arduino UNO, макетной платы, светодиода с ограничивающим резистором, тактовой кнопки, подтягивающего резистора (при необходимости), обладающую следующей функциональностью:

- При включении питания светодиод не горит.
- Каждый раз при нажатии на тактовую кнопку и отпускании – светодиод меняет своё состояние. Если он был включен – выключается, если был выключен – включается.
- Изменение состояния должно происходить только после ОТПУСКАНИИ кнопки (не при нажатии)!

Работа с аналоговым (ШИМ) выходом

Задание 20. Плавное изменение яркости

Указания. Изучите работу ШИМ вывода, инструкцию analogWrite()

Создайте и запрограммируйте электрическую цепь, состоящую из микроконтроллера Arduino UNO, макетной платы, светодиода с ограничивающим резистором, обладающую следующей функциональностью:

- При включении питания светодиод из выключенного состояния плавно увеличивает яркость до своего максимального значения, а затем плавно уменьшает яркость до минимального.
- Увеличение и уменьшение яркость бесконечно повторяется.
- При программировании для управления яркости светодиода используйте ШИМ-сигнал и инструкцию analogWrite().

К каким выводам платы Arduino UNO можно подключить светодиод в этом задании?

Комплексные задания

Задание 21. Управление яркостью светодиода с помощью кнопок

Создайте и запрограммируйте электрическую цепь, состоящую из микроконтроллера Arduino UNO, макетной платы, светодиода с ограничивающим резистором, двух тактовых кнопок с подтягивающими резисторами (при необходимости), обладающую следующей функциональностью:

- При включении питания светодиод не горит.
- Каждый раз при нажатии и отпускании на первую тактовую кнопку, светодиод увеличивает свою яркость на 10% (от максимального значения). Изменение происходит при ОТПУСКАНИИ кнопки!
- Каждый раз при нажатии и отпускании на вторую тактовую кнопку, светодиод уменьшает свою яркость на 10% (от максимального значения). Изменение происходит при ОТПУСКАНИИ кнопки!
- Первая кнопка (при удержании её в нажатом положении более 1 секунды) включает светодиод на максимальную яркость.
- Вторая кнопка (при удержании её в нажатом положении более 1 секунды) выключает светодиод.

Составьте принципиальную электрическую схему устройства.

Задание 22. Параллельный код

Указания. Изучите логические операторы and, or, not

Создать и запрограммировать электрическую цепь, состоящую из микроконтроллера Arduino UNO, макетной платы, трёх тактовых кнопок с подтягивающими резисторами (при необходимости), двух светодиодов красного и зелёного цвета с ограничивающими резисторами, обладающую следующей функциональностью:

- При включении красный светодиод горит, зелёный выключен (доступ запрещён).
- Если комбинация нажатых кнопок соответствует «секретному коду» 1-0-1 (кнопка №1 – нажата, кнопка №2 – отжата, кнопка №3 – нажата), красный светодиод гаснет и загорается зелёный светодиод (доступ разрешён).
- Если комбинация нажатых кнопок не соответствует «секретному коду» 1-0-1, должен гореть красный светодиод.
- В консоль постоянно выводится текущее состояние тактовых кнопок в виде «1-0-1».

Составьте принципиальную электрическую схему, собранного устройства.

ПРИМЕЧАНИЕ. Если данное задание выполняется в среде эмуляции TinkerCAD, вместо тактовых кнопок используйте переключатели с фиксацией.

Задание 23. Последовательный код

Создать и запрограммировать электрическую цепь, состоящую из микроконтроллера Arduino UNO, макетной платы, трёх тактовых кнопок с подтягивающими резисторами (при необходимости), двух светодиодов красного и зелёного цвета с ограничивающими резисторами, обладающую следующей функциональностью:

- При включении красный светодиод горит, зелёный выключен (состояние «доступ запрещён»).
- Если с помощью кнопок будет набрана последовательная комбинация, соответствующая «секретному коду» 1-3-2 (сначала нажимается кнопка №1, потом - кнопка №3, потом - кнопка №2), красный светодиод гаснет и загорается зелёный светодиод (состояние «доступ разрешён»), через 5 секунд схема возвращается в состояние «доступ запрещён».
- Предыстория нажатых кнопок при наборе «секретного кода» не влияет на включение режима «доступ разрешён».
- При состоянии «доступ разрешён», которое длится 5 секунд, схема не реагирует на кнопки.
- В консоль постоянно выводится последовательность нажатых кнопок в виде «1 2 1 3 2 1 ...».

Пример:

- последовательности «1-1-1-1-3-2» и «2-1-3-2» соответствуют «секретному коду»
- последовательности «1-3-3-2» и «1-2-3-2» не соответствуют «секретному коду»

Работа с аналоговым входом

Задание 24. Подключение переменного резистора

Указания. Изучите принцип работы делителя напряжения, устройство переменного резистора, способы работы с аналоговым входом платы Arduino UNO, инструкцию `analogRead()`.

Создать электрическую цепь, состоящую из микроконтроллера Arduino UNO, макетной платы и переменного резистора (потенциометра).

Запрограммируйте постоянное чтение состояния потенциометра (положении ручки) и вывода этого значения в монитор порта.

Составьте принципиальную электрическую схему, собранного устройства.

Задание 25. Управление яркостью светодиода с помощью переменного резистора

Указания. Изучите функции: `constrain()` - для ограничения переменной заданными пределами, `map()` – для пропорционального преобразования значения из одного диапазона в другой.

Часть 1.

Создать и запрограммировать электрическую цепь, состоящую из микроконтроллера Arduino UNO, макетной платы, светодиода ограничивающего резистора, переменного резистора (потенциометра), обладающую следующей функциональностью:

- При перемещении ручки потенциометра в крайнее левое положение – светодиод не горит.
- При перемещении ручки потенциометра из крайнего левого положения в крайнее правое - пропорционально увеличивается яркость светодиода до максимальной.
- Для преобразования диапазона 0...1023 → 0...255 воспользуйтесь функцией `map()`.
- В консоль постоянно выводится показание потенциометра и преобразованное значение для управления яркостью светодиода (два значения в одной строке):

```
pot:0      led:0
pot:100    led:25
pot:1023   led:255
```

Составьте принципиальную электрическую схему, собранного устройства.

Часть 2.

Используя функцию `constrain()`, измените программный код таким образом, чтобы:

- Светодиод НЕ горел при значении потенциометра от 0 до 200
- Светодиод светился с максимальной яркостью при значении потенциометра от 800 до 1023
- При перемещении ручки потенциометра от значения 200 до значения 800 - пропорционально увеличивается яркость светодиода от нуля до максимальной.

Часть 3.

Не изменяя электрическую схему устройства, измените значения в функции `map()` таким образом, чтобы светодиод горел с максимальной яркостью при положении ручки потенциометра в крайнем левом положении, а при перемещении ручки потенциометра из крайнего левого положения в крайнее правое – пропорционально уменьшал яркость до нуля.

ПРИМЕЧАНИЕ. Контролируйте правильность задания с помощью значений, выводимых в консоль, и яркости светодиода.

Задание 26. Подключение фоторезистора

Указания. Изучите устройство и принцип работы фоторезистора, способы подключения фоторезистора к плате Arduino UNO.

Создать электрическую цепь, состоящую из микроконтроллера Arduino UNO, макетной платы и фоторезистора.

Запрограммируйте постоянное чтение состояния фоторезистора и вывода этого значения в монитор порта.

Составьте принципиальную электрическую схему, собранного устройства.

Задание 27. Счетчик посетителей

Создать электрическую цепь, состоящую из микроконтроллера Arduino UNO, макетной платы и фоторезистора.

Запрограммируйте микроконтроллер для подсчета проходящих посетителей. В качестве датчика, обнаруживающего проходящих посетителей, должен выступать фоторезистор. Кратковременное затемнение фоторезистора является фактом обнаружения проходящего мимо датчика человека.

При обнаружении проходящего человека необходимо вывести надпись «Обнаружен посетитель №», увеличить счетчик посетителей и вывести значения счетчика в монитор порта.

Пример вывода:

```
detected visitor №1  
detected visitor №2  
detected visitor №3
```

Задание 28. Индикатор освещенности

Создать и запрограммировать электрическую цепь, состоящую из микроконтроллера Arduino UNO, макетной платы, четырёх светодиодов с ограничивающими резисторами и фоторезистора, обладающую следующей функциональностью:

- При максимальной освещенности фоторезистора светятся все 4 светодиода
- При минимальной освещенности фоторезистора все светодиоды выключены
- При 25% освещенности фоторезистора горит только один светодиод
- При 50% освещенности фоторезистора горят два светодиода
- При 75% освещенности фоторезистора горят три светодиода

Составьте принципиальную электрическую схему, собранного устройства.

Задание 29. Функция map() и нулевая точка

Создать электрическую цепь, состоящую из микроконтроллера Arduino UNO, макетной платы, переменного резистора (потенциометра), тактовой кнопки с подтягивающим резистором (при необходимости).

Напишите программный код, который постоянно читает состояние потенциометра (положении ручки) и выводит ФОРМАТИРОВАННОЕ значения в монитор порта. Форматированное значение соответствует:

«-100» когда ручка потенциометра в самом левом положении

«+100» когда ручка потенциометра в крайнем правом положении

« 0 » когда ручка потенциометра точно в нулевом положении (нулевая точка)

Первоначально нулевая точка соответствует среднему положению ручки потенциометра. При нажатии на кнопку, переопределяется нулевая точка. Положение, которое в настоящее момент находится ручка потенциометра, становится нулевой точкой.

Форматированное значение вывода в нулевой точке равно нулю. При вращении ручки потенциометра по часовой стрелке от нулевой точки – значения плавно увеличиваются от 0 до 100 (крайнее правое положение), при вращении против часовой стрелки от нулевой точки – значения плавно уменьшаются от 0 до -100.

При написании программного кода используйте функцию **map()**