



ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
УПРАВЛЕНИЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ И ПОВЫШЕНИЯ
КВАЛИФИКАЦИИ

Кафедра «Автоматизация производственных процессов»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к проведению практических занятий
по дисциплине

«Микроконтроллеры и микропроцессоры в СУ»

Автор

Быкадор В.С.



Ростов-на-Дону, 2016

Аннотация

Методические указания предназначены для студентов всех форм обучения специальностей 27.03.04, 15.03.04, 15.03.06.

Автор



к.т.н., доцент кафедры «Автоматизация производственных процессов»
Быкадор В.С.

Компьютерную верстку выполнили и принимали участие в проведении экспериментов лабораторного оборудования студент кафедры «Автоматизация производственных процессов» Киселёв Н.В. и студентка кафедры «Автоматизация и математическое моделирование в нефтегазовом комплексе» Остапенко Д.А.



Оглавление

КРАТКАЯ ТЕОРИЯ	4
Оборудование	6
ЛАБОРАТОРНАЯ РАБОТА №1	8
ЛАБОРАТОРНАЯ РАБОТА №2	13
ЛАБОРАТОРНАЯ РАБОТА №3	15
ЛАБОРАТОРНАЯ РАБОТА №4	21
Приложение	24
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	31

КРАТКАЯ ТЕОРИЯ

Микроконтроллеры представляют собой объединение микропроцессора и различных периферийных устройств, таких как таймеры, счетчики, аналого-цифровой преобразователь, интерфейсы RS-232, SPI, I²C и ряд других устройств в одной микросхеме. Применение микроконтроллеров для разработки электронных и автоматических устройств, имеет ряд преимуществ:

1. значительное увеличение скорости разработки устройств;
2. повышение надежности разрабатываемых устройств;
3. уменьшение массогабаритных показателей.

Появление микропроцессоров и особенно микроконтроллеров дало новый импульс в развитии систем автоматического управления за счет возможности несложной реализации различных линейных и главным образом нелинейных законов управления. На сегодняшний день сложно себе представить автоматические устройства без микропроцессорной системы управления. Микропроцессоры, а точнее микроконтроллеры используются в системах контроля бытовой техники, в системах программного управления технологическим оборудованием, в системах управления летательными аппаратами (автопилоты) и многих других.

Дисциплина «Микропроцессоры и микроконтроллеры в системах управления» является прикладной дисциплиной, что предполагает освоение практических методов разработки микропроцессорных систем управления и в первую очередь освоение методов программирования микропроцессорных систем. Тем не менее освоение методов программирования микропроцессорных систем является достаточно трудоемким процессом, особенно на начальных этапах. Это связано как с необходимостью наличия специального оборудования, так и со спецификой программирования микропроцессорных систем. Но совсем недавно была предложена платформа Arduino, позволяющая упростить процесс первоначального освоения разработки микропроцессорных систем, что в дальнейшем способствует переходу к освоению микропроцессорных систем на профессиональном уровне.

Arduino – это средство изучения программирования микроконтроллеров начального уровня. Хорошо адаптирована для выполнения несложных лабораторных работ, практикумов.

Микроконтроллеры и микропроцессоры в СУ

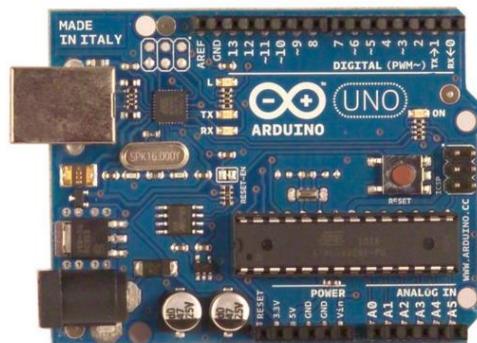


Рис. 1 Arduino UNO

Arduino применяется для разработки элементарных электронных устройств и устройств автоматики с микропроцессорным управлением с возможностью приема сигналов от различных цифровых и аналоговых датчиков и управления различными исполнительными устройствами.

Используемый в лабораторных работах, Arduino Uno (рис. 1) контроллер построен на ATmega328. Uno — самая популярная версия базовой платформы Arduino USB. Uno имеет стандартный порт USB. Arduino Uno во многом схожа с Duemilanove, но имеет новый чип ATmega8U2 для последовательного подключения по USB и новую, более удобную маркировку вход/выходов. Платформа может быть дополнена платами расширения, например, пользовательскими платами с различными функциями. Платформа имеет 14 цифровых вход/выходов (6 из которых могут использоваться как выходы ШИМ (Широтно-Импульсная Модуляция)), 6 аналоговых входов, кварцевый генератор 16 МГц, разъем USB, силовой разъем, разъем ICSP и кнопку перезагрузки. Для работы необходимо подключить платформу к компьютеру посредством кабеля USB, либо подать питание при помощи адаптера AC/DC или батареи.

Программная часть состоит из бесплатной программной оболочки (IDE ARDUINO) для написания программ, их компиляции и программирования аппаратуры. Язык программирования устройств Arduino основан на C/C++.

ОБОРУДОВАНИЕ

Перечислим необходимое оборудование для лабораторных работ:

- 1) Контроллер ARDUINO UNO
- 2) Макетная плата
- 3) Светодиоды
- 4) Резисторы 560 Ом
- 5) Потенциометр 10 кОм
- 6) Пьезоэлемент
- 7) Семисегментный индикатор GNS-3911BU
- 8) Провода-перемычки
- 9) USB-кабель



Рис. 2 Контроллер ARDUINO UNO

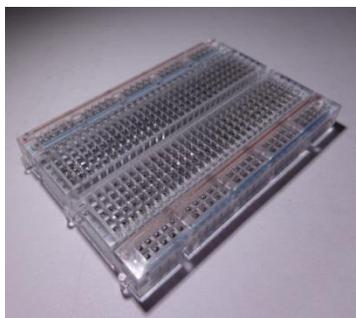


Рис. 3 Макетная плата



Рис. 4 Светодиоды

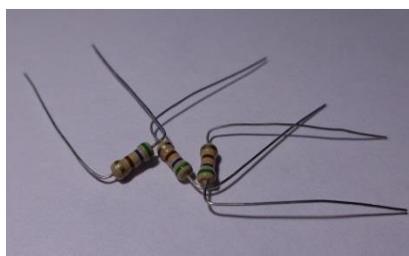


Рис. 5 Резисторы 560 Ом

Микроконтроллеры и микропроцессоры в СУ

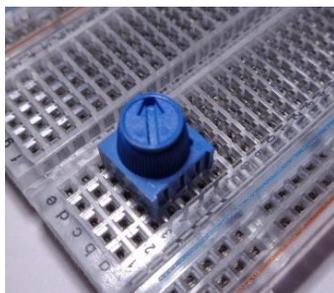


Рис. 6 Потенциометр 10 кОм

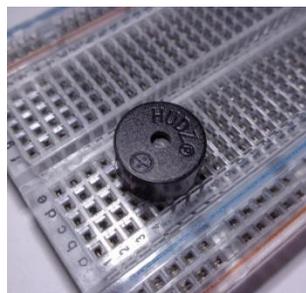


Рис. 7 Пьезоэлемент

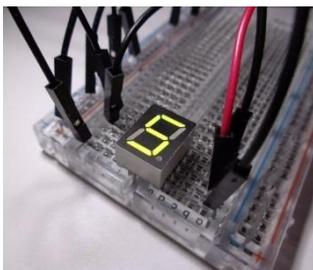


Рис. 8 Семисегментный индикатор GNS-3911BU



Рис. 9 Провода-перемычки



Рис. 10 USB-кабель

ЛАБОРАТОРНАЯ РАБОТА №1

«Управление портами ввода/вывода»

Цель работы:

- 1) Получить знания по структурной организации и технологии разработки программного обеспечения на языке Си для контроллеров ARDUINO.
- 2) Получить навыки программирования цифровых портов контроллера ARDUINO.

Ход работы:

Опыт №1

«Вводное задание»

- 1) По принципиальной электрической схеме определить вывод цифрового порта, к которому подключен светодиод (рис. 1).

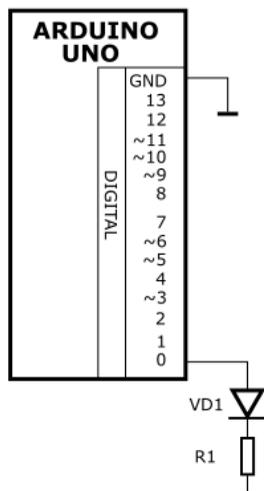


Рис.1 Подключение светодиода к ARDUINO

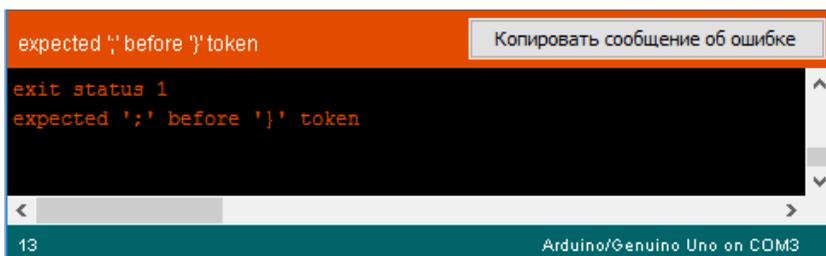
- 2) Выполнить конфигурацию вывода цифрового порта в рутине `setup()`, используя функцию `pinMode(...)` на вывод данных.

Микроконтроллеры и микропроцессоры в СУ

3) Используя функцию задержки `delay(...)` и функцию вывода данных в цифровой порт `digitalWrite()`, создать программу в рутине `loop()`, реализующую периодическое включение и выключение светодиода (мигание) с периодом $T = 1$ с.

4) Выполнить компиляцию созданной программы, используя команду Скетч -> Проверить/Компилировать или комбинацией клавиш *Ctrl+R*.

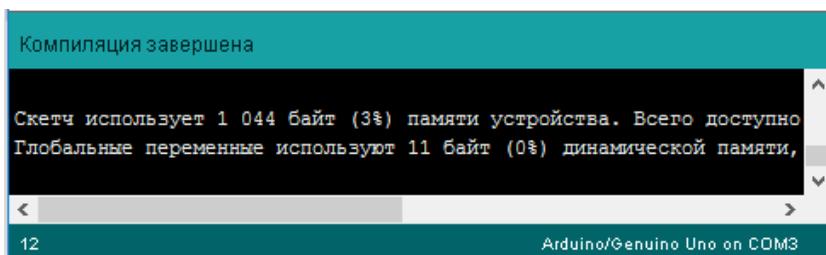
Если в результате компиляции будут выданы ошибки (рис. 2), тогда необходимо их исправить и повторить компиляцию программы.



```
expected ';' before ')' token
exit status 1
expected ';' before ')' token
```

Рис. 2 Пример сообщения, возникающего при ошибке компиляции

Если в результате компиляции не будет выдано ошибок (рис. 3), тогда необходимо выполнить загрузку программы в контроллер ARDUINO, используя команду Скетч -> Загрузить или комбинацией клавиш *Ctrl+U*.



```
Компиляция завершена
Скетч использует 1 044 байт (3%) памяти устройства. Всего доступно
Глобальные переменные используют 11 байт (0%) динамической памяти,
```

Рис. 3 Пример сообщения, возникающего при успешной компиляции

Подключите ARDUINO с помощью кабеля к порту USB компьютера и проверьте правильность программирования цифрового

порта. Если все операции выполнены правильно, вы будете наблюдать мигание светодиода с заданной задержкой.

Опыт №2

«Управление 4-х разрядным семи-сегментным индикатором»

1) По принципиальной электрической схеме определить выводы цифрового порта, к которым подключены светодиоды.

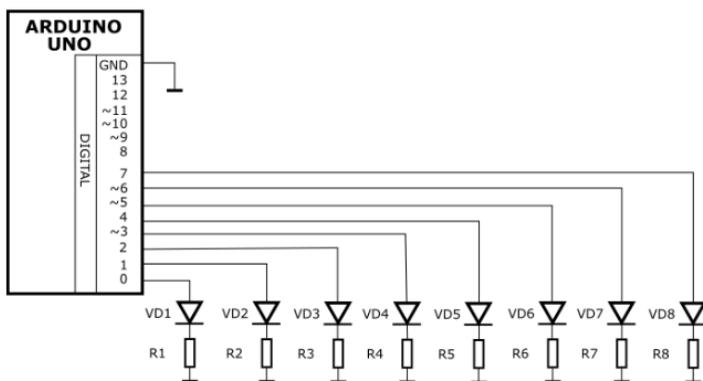


Рис. 4 Подключение 4-х разрядного семисегментного индикатора

2) Выполнить конфигурацию выводов цифрового порта в рутине `setup()`, используя функцию `pinMode(...)` на вывод данных.

3) Используя функцию задержки `delay(...)` и функцию вывода данных в цифровой порт `digitalWrite()`, создать программу в рутине `loop()`, реализующую коммутацию светодиодов согласно заданию (рис. 5, 6) с периодом $T = 1$ с.

4) Выполнить компиляцию и загрузку созданной программы, используя соответствующие команды IDE ARDUINO.

Подключите ARDUINO с помощью кабеля к порту USB компьютера и проверьте правильность программирования цифровых портов. Если все операции выполнены правильно, вы будете

наблюдать мигание светодиодов в режиме бегущей строки с заданной задержкой.

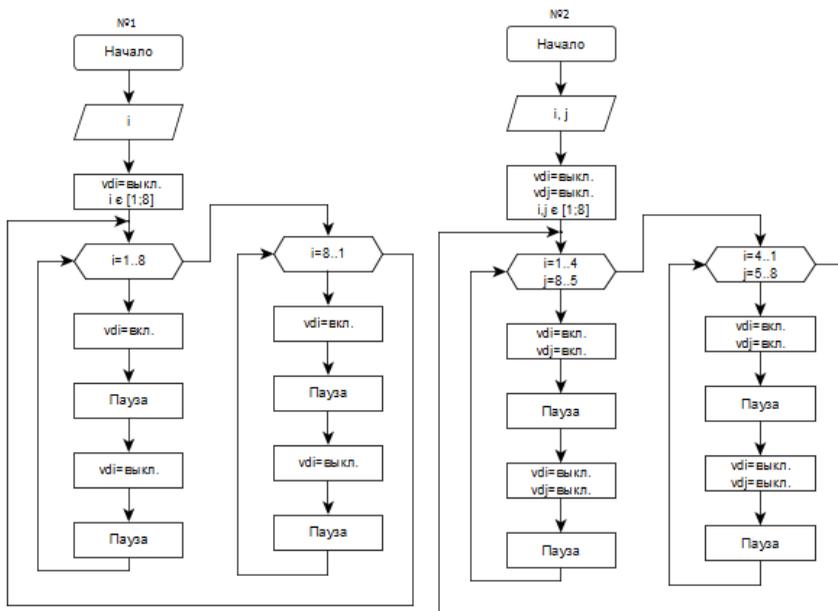


Рис. 5 Алгоритмы заданий для вариантов 1 и 2

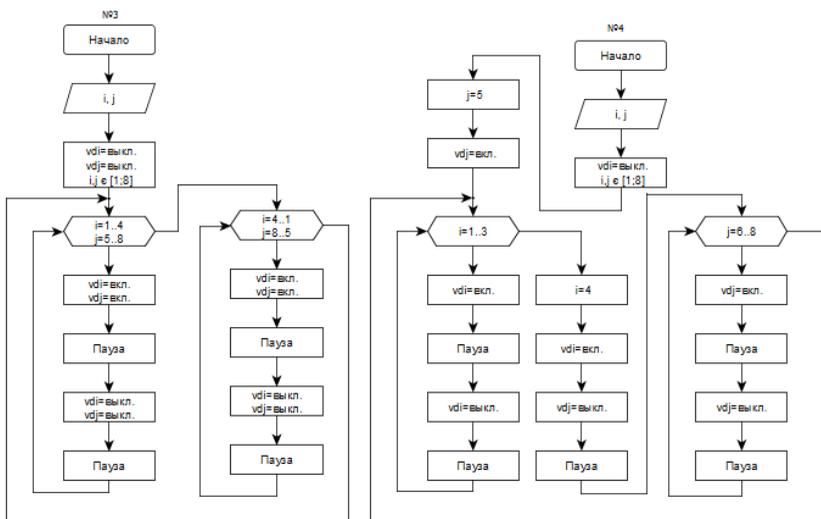


Рис. 6 Алгоритмы заданий для вариантов 3 и 4

Контрольные вопросы:

- 1) Для каких целей необходимы рутины `setup()` и `loop()`?
- 2) Для чего используется функция `pinMode(...)`, какие параметры она принимает и какое значение имеют эти параметры?
- 3) Для чего используется функция `digitalWrite(...)`, какие параметры она принимает и какое значение имеют эти параметры?
- 4) Какие функции используются для реализации программной задержки, какие параметры принимают эти функции?
- 5) Можно ли, используя функцию `digitalWrite(...)` послать в цифровой порт байт данных, если нет, то подумайте, как можно реализовать такую функцию?

ЛАБОРАТОРНАЯ РАБОТА №2

«Управление семисегментным индикатором»

Цель работы:

- 1) Получить навыки разработки программ на языке Си по управлению семиразрядным индикатором.
- 2) Получить навыки использования условного оператора языка Си «*if*» и типа данных – массивы.

Ход работы:

- 1) По принципиальной электрической схеме (рис. 1) определить выводы цифрового порта, к которым подключен семисегментный индикатор.

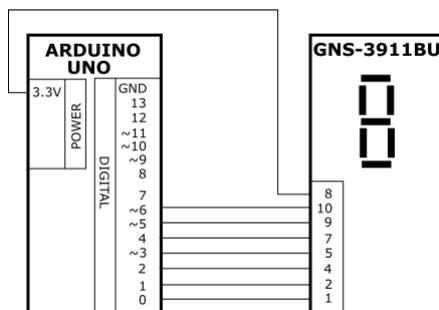


Рис. 1 Подключение семисегментного индикатора к ARDUINO

- 2) По индексации светодиодов (рис. 2) и номерам выводов цифрового порта в семисегментном индикаторе, определить коды отображения символов чисел от 0 до 9.

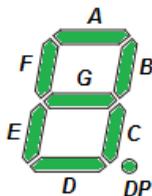


Рис. 2. Индексация светодиодов в семисегментном индикаторе

3) В программе создать массив с кодами чисел. Данный массив потребуется для получения изображения числа на семисегментном индикаторе.

4) По принципиальной электрической схеме семисегментного индикатора (рис. 3), определить тип управляющего сигнала для включения светодиодов индикатора (логический «0» или логическая «1»).

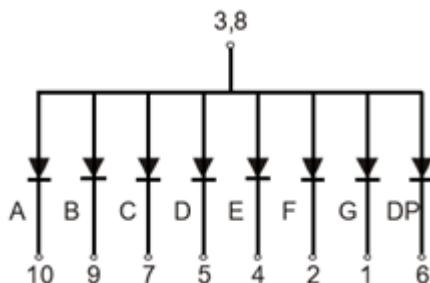


Рис. 3. Принципиальная электрическая схема семисегментного индикатора GNS-3911BU

5) Выполнить конфигурацию выводов цифрового порта в рутине `setup()`.

6) Создать программу в рутине `loop()`, реализующую счетную последовательность, с периодом выставления данных в цифровой порт $T = 1$ с. С пустой диаграммой состояния необходимо обратиться к преподавателю за вариантом.

7) Выполнить компиляцию и загрузку созданной программы.

Контрольные вопросы:

1) Для каких целей, в программе, используется условный оператор `if`?

2) Приведите полный синтаксис условного оператора `if` и поясните для каких целей используется каждый блок?

3) Как на блок схемах обозначается условный оператор `if`?

4) Для организации каких данных используются массивы?

5) Как объявляется и инициализируется массив? Приведите пример.

6) Приведите пример записи в массив и чтение из массива данных из i -ой позиции.

ЛАБОРАТОРНАЯ РАБОТА №3

«Считывание данных с цифрового порта»

Цель работы:

- 1) Получить навыки настройки выводов цифрового порта на ввод данных на языке Си в контроллере ARDUINO.
- 2) Получить навыки считывания данных с выводов цифрового порта на языке Си в контроллере ARDUINO.

Ход работы:

- 1) По принципиальной электрической схеме (рис. 1) определить выводы цифрового порта, к которым подключены кнопки и семисегментный индикатор (семисегментный индикатор подключен также, как и в Л.Р. № 2).

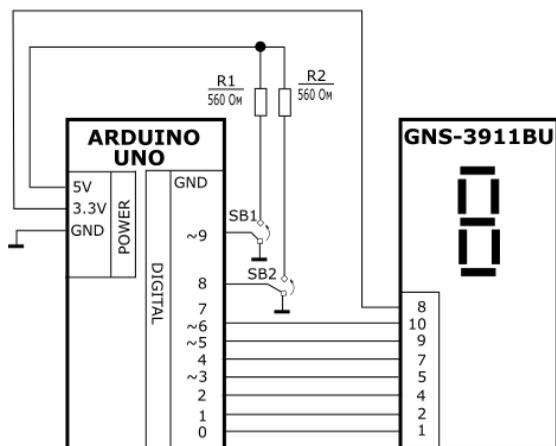


Рис. 1 Принципиальная электрическая схема системы

- 2) Выполнить конфигурацию выводов цифрового порта на ввод данных и инициализацию переменных программы в рутине `setup()`.
- 3) Создать программу в рутине `loop()`, реализующую алгоритм приведенный в задании, согласно варианту.
- 4) Выполнить компиляцию и загрузку созданной программы.

Задание для варианта 1. Согласно нажимаемой комбинации кнопок необходимо выполнить вывод соответствующего числа на семисегментный индикатор. Алгоритм приведен на рис. 2.

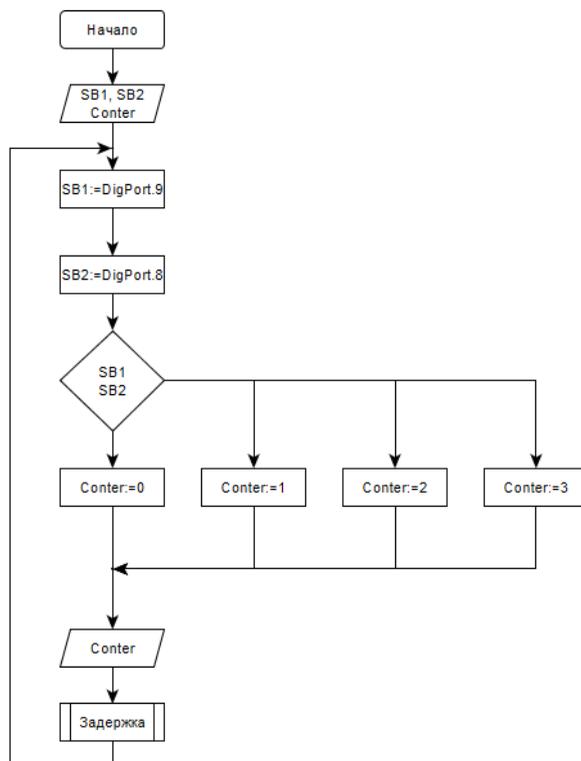


Рис. 2 Задание для варианта 1

Задание для варианта 2. Согласно нажимаемой комбинации кнопок необходимо выполнить инкремент или декремент переменной в ОЗУ контроллера и вывод этого числа на семисегментный индикатор.

Следует обратить внимание на два момента. Во-первых, одновременное нажатие на две кнопки является запрещающей комбинацией – на индикатор должен выводиться символ «Е» (сокр. от англ. «Error» - ошибка). Во-вторых, индикатор позволяет выводить числа от 0 до 9 включительно, поэтому необходимо предусмотреть меры, которые не позволят выводить отрицательные числа, а также числа больше 9. Алгоритм приведен на рис. 3.

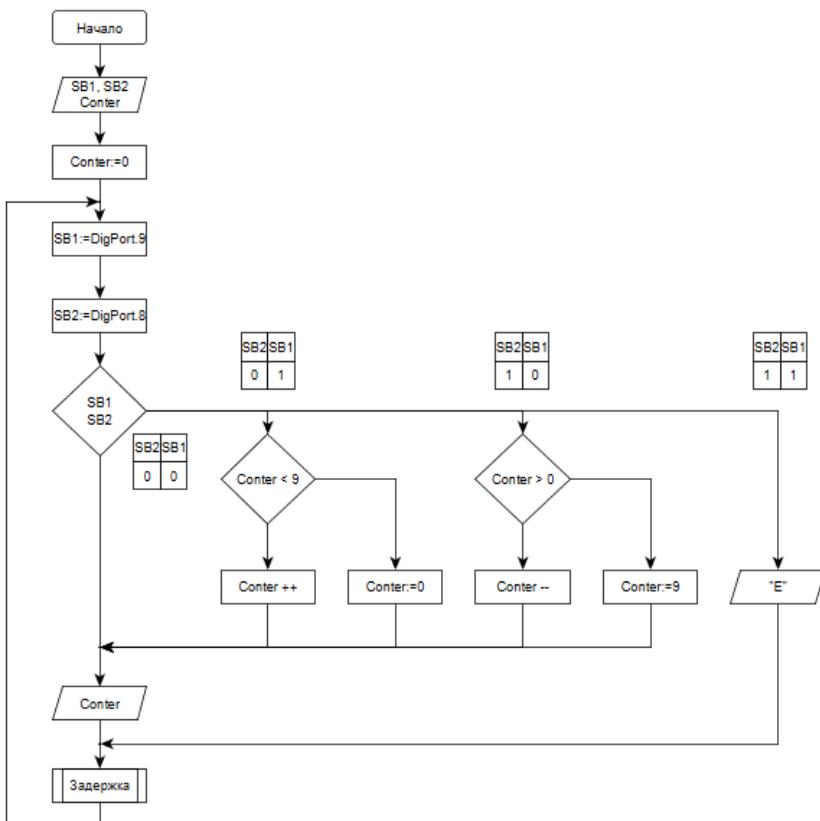


Рис. 3 Задание для варианта 2

Задание для варианта 3. Согласно нажимаемой комбинации кнопок необходимо выполнить инкремент или декремент частоты вывода символа «8» на семисегментный индикатор.

Следует обратить внимание на то, что период вывода символа «8» должен изменяться в пределах от 0 мс. до 3000 мс. Алгоритм приведен на рис. 4.

Микроконтроллеры и микропроцессоры в СУ

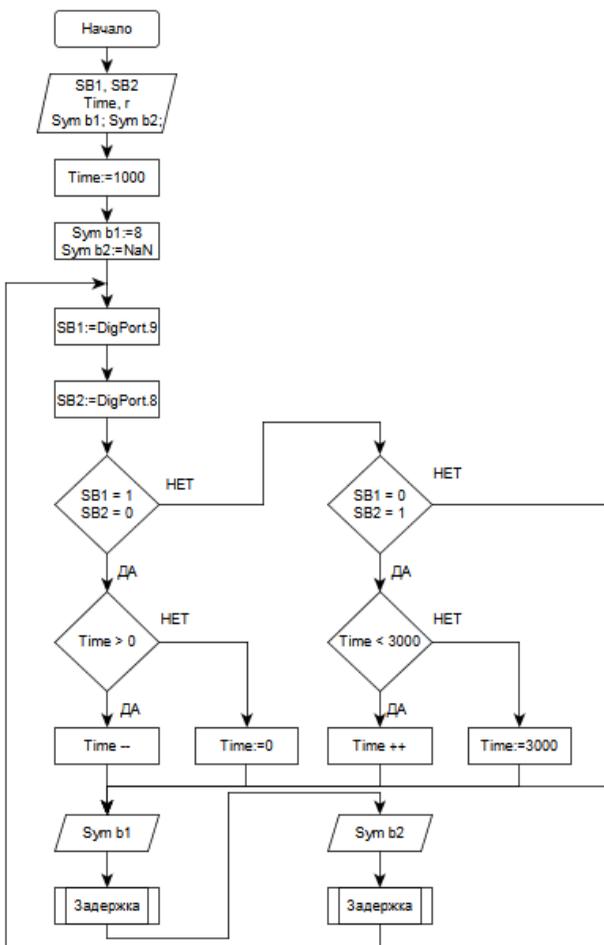


Рис. 4 Задание для варианта 3

Задание для варианта 4. Согласно нажимаемой комбинации кнопок выполняется выбор автоматического режима работы контроллера - «инкрементальный» или «декрементальный». Так, например, кратковременное нажатие на кнопку SB1, приводит к тому, что контроллер в автоматическом режиме выполняет инкремент переменных в ОЗУ и вывод этого значения на семисегментный индикатор.

Следует обратить внимание на два момента. Во-первых, одновременное нажатие на две кнопки является запрещающей комбинацией – переменной должно быть присвоено значение «0». Во-вторых, индикатор позволяет выводить числа от 0 до 9 включительно, поэтому необходимо предусмотреть меры, которые не позволят выводить отрицательные числа, а также числа больше 9. Алгоритм приведен на рис. 5.

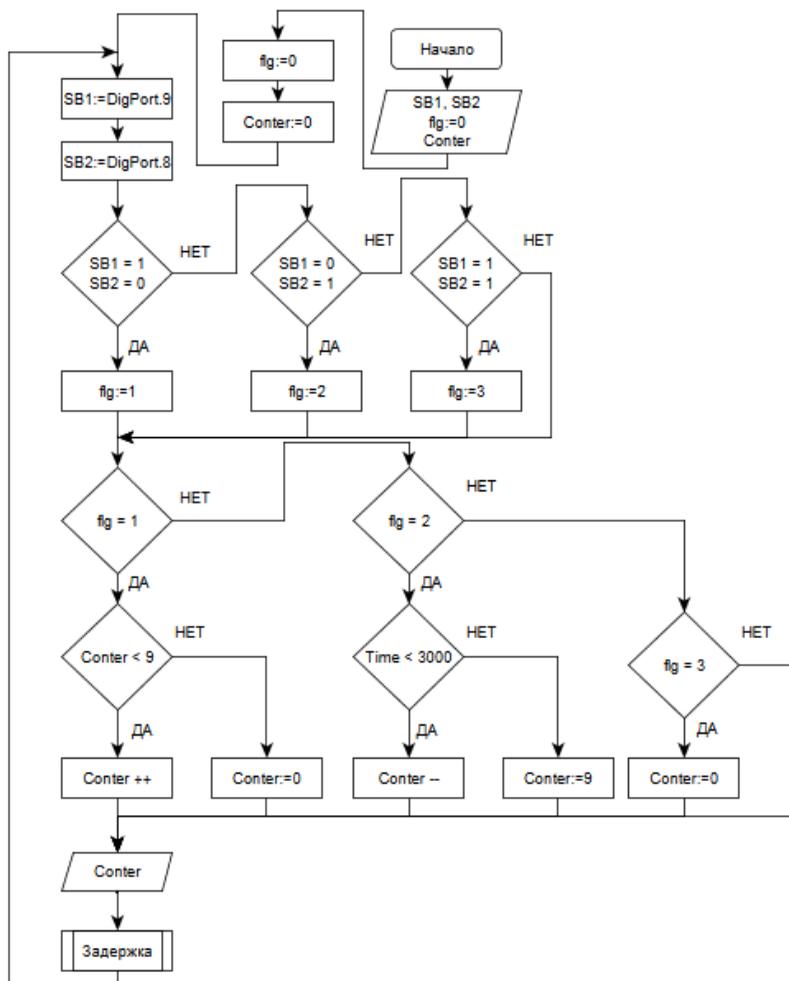


Рис. 5 Задание для варианта 4

Контрольные вопросы:

- 1) Каким образом выполняется инициализация выводов цифрового порта на ввод данных в контроллере ARDUINO?
- 2) Каким образом выполняется считывание данных с выводов цифрового порта контроллера ARDUINO?
- 3) Как можно реализовать считывание байта данных с выводов цифрового порта контроллера ARDUINO и запись байта в ОЗУ контроллера?

ЛАБОРАТОРНАЯ РАБОТА №4

«Считывание данных с аналогового порта»

Цель работы:

Получить навыки считывания и преобразования данных с выводов аналогового порта на языке Си в контроллере ARDUINO.

Необходимые теоретические сведения:

Представление аналогового напряжения в контроллере.

Напряжение на выводе аналогового порта (ANALOG IN) может изменяться от 0 до 5 В. Напряжение с выводов аналогового порта поступает на внутренний аналого-цифровой преобразователь (АЦП) с выхода которого считывается код (функция `analogRead()`), соответствующий входному аналоговому напряжению. Диапазон, считанного функцией `analogRead()`, цифрового кода равен от 0 до 1023, таким образом можно рассчитать дискрету аналогового напряжения $\frac{5В}{1024} \approx 0,0049 В$. Например, если в результате был считан код, равный 289, то умножив его на дискрету, получим значение аналогового напряжения, поданного на вход аналогового порта $289 \cdot 0,0049 \approx 1,41 В$.

Пьезоэлемент.

Пьезоэлемент позволяет преобразовать напряжение, поданное на его входы в механическое перемещение. Если на вход пьезоэлемента подавать напряжение в виде импульсов с переменной частотой, то пьезоэлемент будет формировать звуковые волны разной частоты.

Ход работы:

1) По принципиальной электрической схеме (рис. 1) определить вывод аналогового порта, к которому подключен потенциометр и выводы цифрового порта, к которому подключены пьезоэлемент и семисегментный индикатор (семисегментный индикатор подключен также, как и в Л.Р. № 2).

2) Выполнить конфигурацию выводов цифрового порта на вывод данных и инициализацию переменных программы в рутине `setup()`.

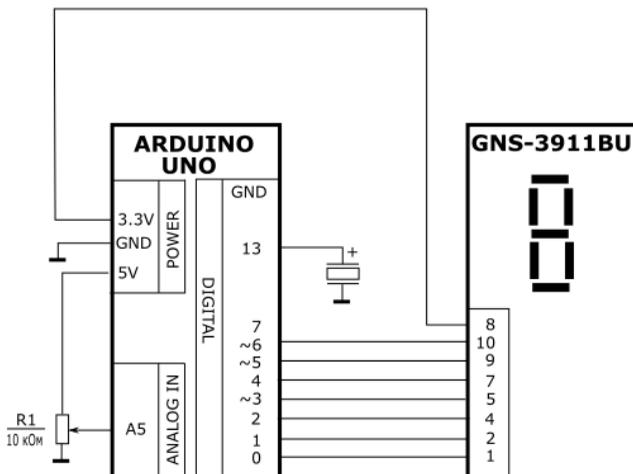


Рис. 1. Принципиальная электрическая схема системы

3) Создать программу в рутине `loop()`, реализующую алгоритм приведенный в задании.

4) Выполнить компиляцию и загрузку созданной программы.

Задание:

1) Циклически выполнять считывание значения напряжения с вывода аналогового порта;

2) Выводит целочисленное значение напряжения на семи-сегментный индикатор;

3) Формировать импульсную последовательность с частотой обратной значению напряжения, для изменения звука, генерируемого пьезоэлементом (рекомендуется использовать функцию `tone()`).

Алгоритм приведен на рис. 2.

Микроконтроллеры и микропроцессоры в СУ

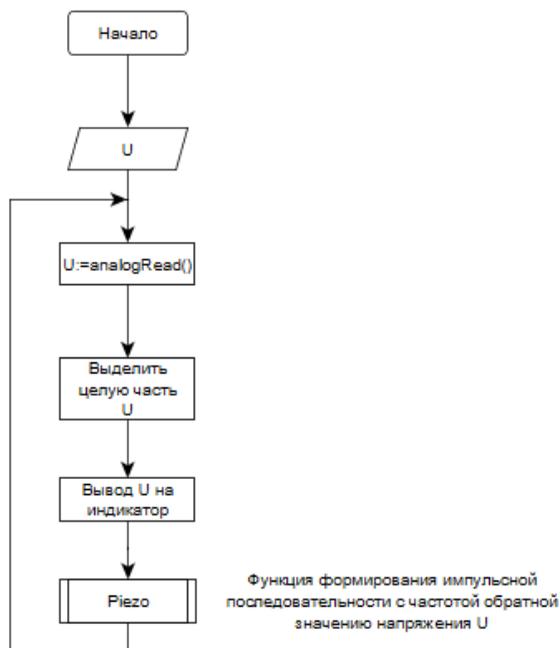


Рис. 2. Алгоритм программы

Контрольные вопросы:

- 1) Каким образом, в контроллере, представляется аналоговое напряжение?
- 2) Рассчитайте дискрету аналогового напряжения, если диапазон изменения напряжения от 0 до 3В, а соответствующий ему диапазон цифрового кода от 0 до 2047?
- 3) Какая функция используется для считывания данных с вывода аналогового порта? Приведите пример.
- 4) Какие параметры импульсного сигнала необходимо варьировать, чтобы изменять громкость и частоту его звучания?

ПРИЛОЖЕНИЕ

КРАТКАЯ СПРАВКА ПО ОСНОВНЫМ ФУНКЦИЯМ ЯЗЫКА СИ В IDE ARDUINO

Приведены основные данные по аутентичным функциям IDE ARDUINO, необходимые для выполнения лабораторных работ по данному курсу.

Полные данные по языку Си, используемому в IDE ARDUINO, приведены по ссылке: www.arduino.cc/en/Reference/

Основные типы данных

№ п.п.	Тип данных	Название типа данных	Размер	Принимаемые значения
1.	<code>boolean</code>	Логический тип	1 бит	{true, false} или {1, 0}
2.	<code>char</code>	Целочисленный знаковый тип	1 байт	-128...+127 или ASCII символы, например, 'A', 'B', 'f' и т.п.
3.	<code>unsigned char</code>	Целочисленный без знаковый тип	байт	0...255
4.	<code>int</code>	Целочисленный знаковый тип	2 байта	-32 768...+32 767
5.	<code>unsigned int</code>	Целочисленный без знаковый тип	2 байта	0...65 535
6.	<code>long</code>	Целочисленный знаковый тип	4 байта	-2 147 483 648... +2 147 483 647
7.	<code>unsigned long</code>	Целочисленный без знаковый тип	4 байта	0...4 294 967 295
8.	<code>float</code>	Вещественный тип	4 байта	-3.4028235E+38... +3.4028235E+38

Микроконтроллеры и микропроцессоры в СУ

Основные функции

Преобразования данных				
Запись	Описание	Синтаксис	Параметр	Пример
<code>int()</code>	Преобразует значение, переданное в функцию как аргумент, к вещественному типу <code>int</code> .	<code>int(x)</code>	<code>x</code> : значение какого-либо типа.	<pre>float a; int b; a = 5.3; b = int(a);</pre>
<code>float()</code>	Преобразует значение, переданное в функцию как аргумент, к целочисленному типу <code>float</code> .	<code>float(x)</code>	<code>x</code> : значение какого-либо типа.	<pre>int a; float b; float c; float d; a = 5; b = 10.4; c = float(a); d = b + c;</pre>
Работы с битами в байте				
<code>bitRead()</code>	Считывание бита из байта.	<code>bitRead(x, n)</code>	<code>x</code> : пер. типа байт, из кот. считывается бит. <code>n</code> : ном. считываемого бита. Мл. бит имеет ном. 0, ст. – 7.	<pre>unsigned char a; boolean b; a = 0b00000101; b = bitRead(a, 2); // b = 1</pre>
<code>bitWrite()</code>	Запись бита в байта на соответствующую позицию.	<code>bitWrite(x, n, b)</code>	<code>x</code> : пер. типа байт, в кот. записывается бит. <code>n</code> : ном. записываемого бита. Мл. бит имеет номер 0, ст. – 7.	<pre>unsigned char a; a = 0b00000101; bitWrite(a, 3, 1); //a= 0b00001101</pre>
Работы с цифровым портом				
<code>pinMode()</code>	Конфигурация вывода цифрового порта на ввод или вывод бита данных.	<code>pinMode(pin, mode)</code>	<code>pin</code> : ном. вывода в цифровом порту. <code>mode</code> : реж. раб. порта. INPUT: ввод. OUTPUT: вывод. INPUT_PULLUP: вв. бита данн. с подкл. внут. подтягивающим резистором.	<pre>boolean pin = 4; pinMode(pin, OUTPUT); // вывод 4 цифрового порта сконфигурирован на выдачу сигнала во внешнюю схему</pre>

Микроконтроллеры и микропроцессоры в СУ

digitalWrite ()	Запись бита (0 или 1) в вывод цифрового порта.	digitalWrite (pin, value)	pin: номер вывода в цифровом порту value: выводимое значение. HIGH: логическая единица. LOW: логический ноль.	boolean pin = 4; pinMode (pin, OUTPUT); // вывод 4 цифрового порта сконфигурирован на выдачу сигнала во внешнюю схему digitalWrite (pin, HIGH); // в вывод 4 цифрового порта логической единицы
digitalRead ()	Чтение бита (0 или 1) с вывода цифрового порта	digitalRead (pin)	pin: номер вывода в цифровом порту.	boolean pin = 4; boolean getBit; pinMode (pin, INPUT); // вывод 4 цифрового порта сконфигурирован на чтение бита getBit = digitalRead (pin);

Работы с аналоговым портом

analogRead ()	Чтение данных с вывода аналогового порта. Контроллер ARDUINO имеет 10-разрядный аналого-цифровой преобразователь (АЦП). Диапазон входного напряжения и разрешающая способность АЦП могут быть изменены функцией analogReference ().	analogRead (pin)	pin: номер вывода в аналоговом порту.	boolean pin = 0; int data; data=analogRead (pin);
tone ()	Формирует меандр (импульсный сигнал с равными по длительности импульсами и паузами) определенной частоты на выводе порта. К выводу порта может быть подключен пьезоэлемент или какой-нибудь другой динамик для воспроизведения звука.	tone (pin, frequency) tone (pin, frequency, duration)	pin: номер вывода в порту frequency: частота сигнала в герцах [Гц] (тип данных unsigned int); duration: длительность звучания в миллсек. (тип данных unsigned long);	boolean pin = 2; tone (pin, 50, 100);

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Соммер У. Программирование микроконтроллерных плат Arduino/Freduino. – СПб.: БВХ – Петербург, 2012. – 256 с. ил. – (Электроника).
2. Болл Стюарт Р. Аналоговые интерфейсы микроконтроллеров. – М.: Издательский дом «Додэка-XXI», 2007. – 360с.: ил. (Серия «Программируемые системы»).
3. Петин В. А. Проекты с использованием контроллера Arduino. - СПб.: БВХ – Петербург, 2014. – 400 с. ил. – (Электроника).
4. Ramos Enrique M., Casto Ciriaco D. Arduino and Kinect Projects. - «Technology in Action», RM, IT, 2010. – 411с.: рс.
5. Patrick J., Gertz E. Atmospheric Monitoring with Arduino. - First Edition, «O’Reilly Media», CA, USA, 2012. – 89с.: рс.