Федеральное государственное бюджетное образовательное учреждение высшего образования «Сибирский государственный университет телекоммуникаций и информатики»

Кафедра вычислительных систем (ВС)

Отчет

по курсовому проекту

по дисциплине «Микропроцессорные системы»

**«Разработка калькулятора»**

**Выполнил:** студент группы ИВ-422

Блинов А.И.

**Проверил:** старший преподаватель

кафедры вычислительных систем (ВС)

Гонцова А.В.

Новосибирск 2018

**Задание на курсовую работу**

Реализовать на базе программируемого микроконтроллера Arduino UNO™ калькулятор.

**Описание среды разработки**

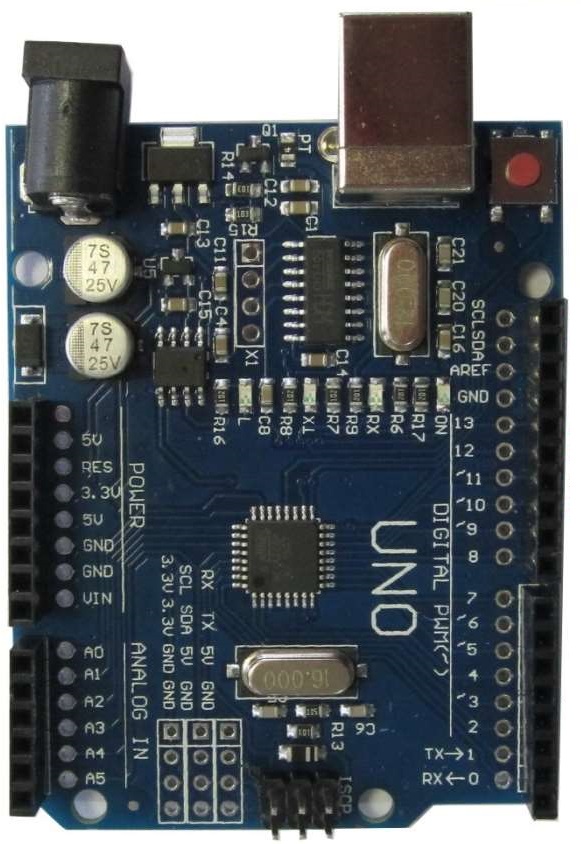
Плата Arduino Uno R3

Рисунок 1 – Плата Arduino Uno R3.

Общие сведения

Arduino UNO R3 выполнен на микроконтроллере ATmega328. У него:

* 14 цифровых портов входа-выхода (6 из них поддерживают режим ШИМ модуляции);
* 6 аналоговых входов;
* частота тактирования 16 МГц;
* USB порт;
* разъем питания;
* разъем внутрисхемного программирования;
* кнопка сброса.

У платы есть все необходимые компоненты для обеспечения работы микроконтроллера. Достаточно подключить USB кабель к компьютеру и подать питание. Микроконтроллер установлен на колодке, что позволяет легко заменить его в случае выхода из строя.

Краткие технические сведения

|  |  |
| --- | --- |
| Тип микроконтроллера | ATmega328P |
| Напряжение питания микроконтроллера | 5 В |
| Рекомендуемое напряжение питания платы | 7 – 12 В |
| Предельно допустимое напряжение питания платы | 6 – 20 В |
| Цифровые входы-выходы | 14 (из них 6 поддерживают ШИМ) |
| Выходы ШИМ модуляции | 6 |
| Аналоговые входы | 6 |
| Допустимый ток цифровых выходов | 20 мА |
| Допустимый ток выхода 3,3 В | 50 мА |
| Объем флэш памяти (FLASH) | 32 кБ (из которых 0,5 кБ используется загрузчиком) |
| Объем оперативной памяти (SRAM) | 2 кБ |
| Объем энергонезависимой памяти (EEPROM) | 1 кБ |
| Частота тактирования | 16 мГц |
| Длина платы | 68,6 мм |
| Ширина платы | 53,4 мм |
| Вес | 25 г |

Таблица 1 – Краткие технические сведения платы Arduino Uno R3.

Питание

Плата UNO может получать питание от USB порта или от внешнего источника. Источник питания выбирается автоматически. В качестве внешнего источника питания может использоваться сетевой адаптер или батарея. Адаптер подключается через разъем диаметром 2,1 мм (центральный контакт – положительный). Батарея подключается к контактам GND и Vin разъема POWER.

Напряжение внешнего источника питания может быть в диапазоне 6 – 20 В. Но рекомендуется не допускать снижение напряжения ниже 7 В из-за нестабильной работы устройства. Также нежелательно повышать напряжение питания более 12 В, т.к. может перегреется стабилизатор и выйти из строя. Т.е. рекомендуемый диапазон напряжения питания 7 – 12 В.

Для подключения питания могут быть использованы следующие выводы:

**VIN**. Питание платы от внешнего источника питания. Не связано с питанием 5 В от USB или выходами других стабилизаторов. Через этот контакт можно получать питание для своего устройства, если плата питается от адаптера.

**5 V**. Выход стабилизатора напряжения платы. На нем напряжение 5 В при любом способе питания. Питать плату через этот вывод не рекомендуется, т.к. не используется стабилизатор, что может привести к выходу микроконтроллера из строя.

**3 V 3**. Напряжение 3,3 В от стабилизатора напряжения на плате. Предельно допустимый ток потребления от этого вывода 50 мА.

**GND**. Выводы заземления.

**IOREF**. На выводе информация о рабочем напряжении платы. Плата расширения может считать значение сигнала и переключиться на режим питания 5 В или 3,3 В.

Память

У микроконтроллера ATmega328 несколько видов памяти:

* 32 кБ флэш (FLASH);
* 2 кБ оперативной памяти (SRAM);
* 1 кБ энергонезависимой памяти (EEPROM).

Входы и Выходы

Каждый из 14 цифровых выводов может быть использован в качестве выхода или входа. Уровень напряжения на выводах 5 В. Рекомендовано вытекающий и втекающий ток каждого вывода ограничивать на уровне 20 мА. Предельно допустимое значение этого параметра составляет 40 мА. Каждый вывод имеет внутренний подтягивающий резистор сопротивлением 20-50 кОм. Резистор может быть отключен программно.

Некоторые выводы могут выполнять дополнительные функции.

1. **Последовательный интерфейс: выводы 0 (Rx) и 1 (Tx).** Используются для приема (Rx) и передачи (Tx) последовательных данных логических уровней TTL. Эти выводы подключены к выводам передачи данных микросхемы ATmega16U2, используемой в качестве моста USB-UART.
2. **Внешние прерывания: выводы 2 и 3.** Эти выводы могут быть использованы как входы внешних прерываний. Программно могут быть установлены на прерывание по низкому уровню, положительному или отрицательному фронту, или на изменение уровня сигнала.
3. **ШИМ: выводы 3, 5, 6, 9, 10, 11.** Могут работать в режиме ШИМ модуляции с разрешением 8 разрядов.
4. **Последовательный интерфейс SPI: выводы 10(SS), 11(MOSI), 12(MISO), 13(SCK).**
5. **Светодиод: вывод 13.** Светодиод, подключенный к выводу 13. Светится при высоком уровне сигнала на выводе.
6. **Интерфейс TWI: вывод A4 или SDA и A5 или SCL.** Коммуникационный интерфейс TWI.
7. У платы Arduino UNO есть **6 аналоговых входов, обозначенных A0-A5**. Разрешающая способность аналогового цифрового преобразования 10 разрядов. По умолчанию, входное напряжение измеряется относительно земли в диапазоне 0-5 В, но может быть изменено с помощью вывода AREF и программных установок.

Еще 2 вывода платы имеют функции:

1. **AREF.** Опорное напряжение АЦП микроконтроллера.
2. **RESET.** Низкий уровень на этом выводе вызывает сброс микроконтроллера.

Связь

Модуль Arduino UNO имеет средства для связи с компьютером, с другой платой UNO или с другими микроконтроллерами. Для этого на плате существует интерфейс UART с логическими уровнями TTL (5 В), связанный с выводами 0 (RX) и 1(TX). Микросхема ATmega16U2 на плате связывает UART интерфейс с USB портом компьютера. При подключении к порту компьютера, появляется виртуальный COM порт, через который программы компьютера работают с Arduino. Прошивка ATmega16U2 использует стандартные драйверы USB-COM и установка дополнительных драйверов не требуется. Для операционной системы Windows необходим соответствующий .inf файл. В интегрированную среду программного обеспечения Arduino (IDE) включен монитор обмена по последовательному интерфейсу, который позволяет посылать и получать с платы простые текстовые данные. На плате есть светодиоды RX и TX, которые индицируют состояние соответствующих сигналов для связи через USB (но не для последовательного интерфейса на выводах 0 и 1).

Микроконтроллер ATmega328 также поддерживает коммуникационные интерфейсы I2C (TWI) и SPI.

Программирование

Контроллер программируется из интегрированной среды программного обеспечения Arduino (IDE). Программирование происходит под управлением резидентного загрузчика по протоколу STK500. Аппаратный программатор при этом не требуется.

Микроконтроллер можно запрограммировать через разъем для внутрисхемного программатора ICSP, не используя, загрузчик. Исходный код программы-загрузчика находится в свободном доступе.

**Автоматический (программный) сброс**

Для того, чтобы не приходилось каждый раз перед загрузкой программы нажимать кнопку сброс, на плате UNO реализована аппаратная функция сброса, инициируемая с подключенного компьютера. Один из сигналов управления потоком данных (DTR) микросхемы ATmega16U2 подключен к выводу сброса микроконтроллера ATmega328 через конденсатор емкостью 0,1 мкФ.  Когда сигнал DTR переходит в низкое состояние, формируется импульс сброса микроконтроллера. Это решение позволяет загружать программу одним нажатием кнопки из интегрированной среды программирования Arduino (IDE).

На модуле UNO существует дорожка, которую можно перерезать для отключения функции автоматического сброса. Дорожка маркирована надписью «RESET-EN». Автоматический сброс также можно запретить, подключив резистор сопротивлением 110 Ом между линией питания 5 В и выводом RESET.

**Защита USB порта от перегрузок**

В плате Arduino UNO линия питания от интерфейса USB защищена восстанавливаемым предохранителем. При превышении тока свыше 500 мА, предохранитель разрывает цепь до устранения короткого замыкания.

Принципиальная схема контроллера Arduino Uno R3

Принципиальная схема контроллера изображена в Приложении 1.

**Подключаемые устройства**

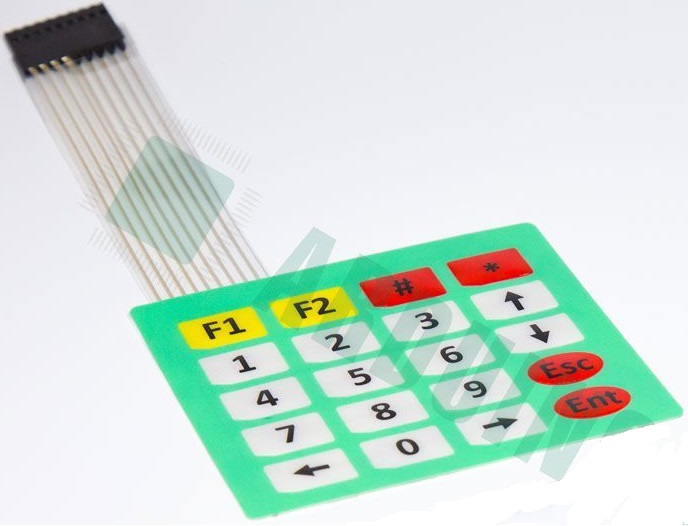
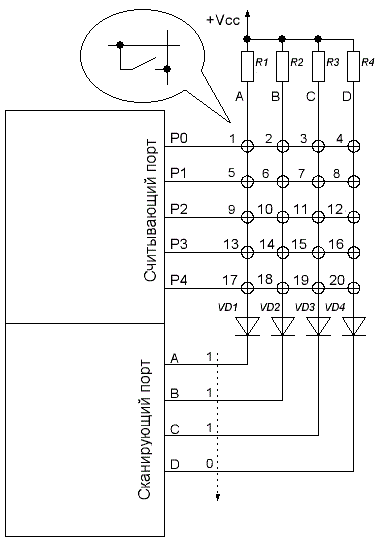
Матричная клавиатура

Рисунок 2 – Матричная клавиатура 5х4.

В моём случае матрица имеет размер 5 \* 4 = 20 кнопок (5 строк P0, P1, P2, P3, P4 и 4 столбца A, B, C, D). Обратим внимание на рисунок 3. Линии P0…P4, являющиеся строками, всегда настроены на ввод в качестве считывающего порта. С них и производится считывание состояния кнопок. Линии A…D настроены на вывод в качестве сканирующего порта. На них в каждый момент времени сигнал низкого уровня (логический ноль) подается только на один столбец кнопок, на остальные подаётся логическая единица.

Рисунок 3 – Схема матричной клавиатуры с 5х4 кнопками.

Если в столбце с логическим нулём не нажата ни одна кнопка, то ноль уходит в землю. И только при нажатии кнопок этого столбца ноль переходит на соответствующую линию строки матрицы (P0…P4) и линия столбца с логическим нулём замыкается.

Таким образом столбцы сканируются по кругу. Зная какой столбец сканируется, и получив на считывающем порту ноль, можно, как по координатам, определить нажатую кнопку.

LCD-Дисплей

Рисунок 4 – LCD-дисплей 16х2 символьный.

В моей работе используется монохромный дисплей, который имеет опциональную подсветку и может отображать 2 строки по 16 символов.

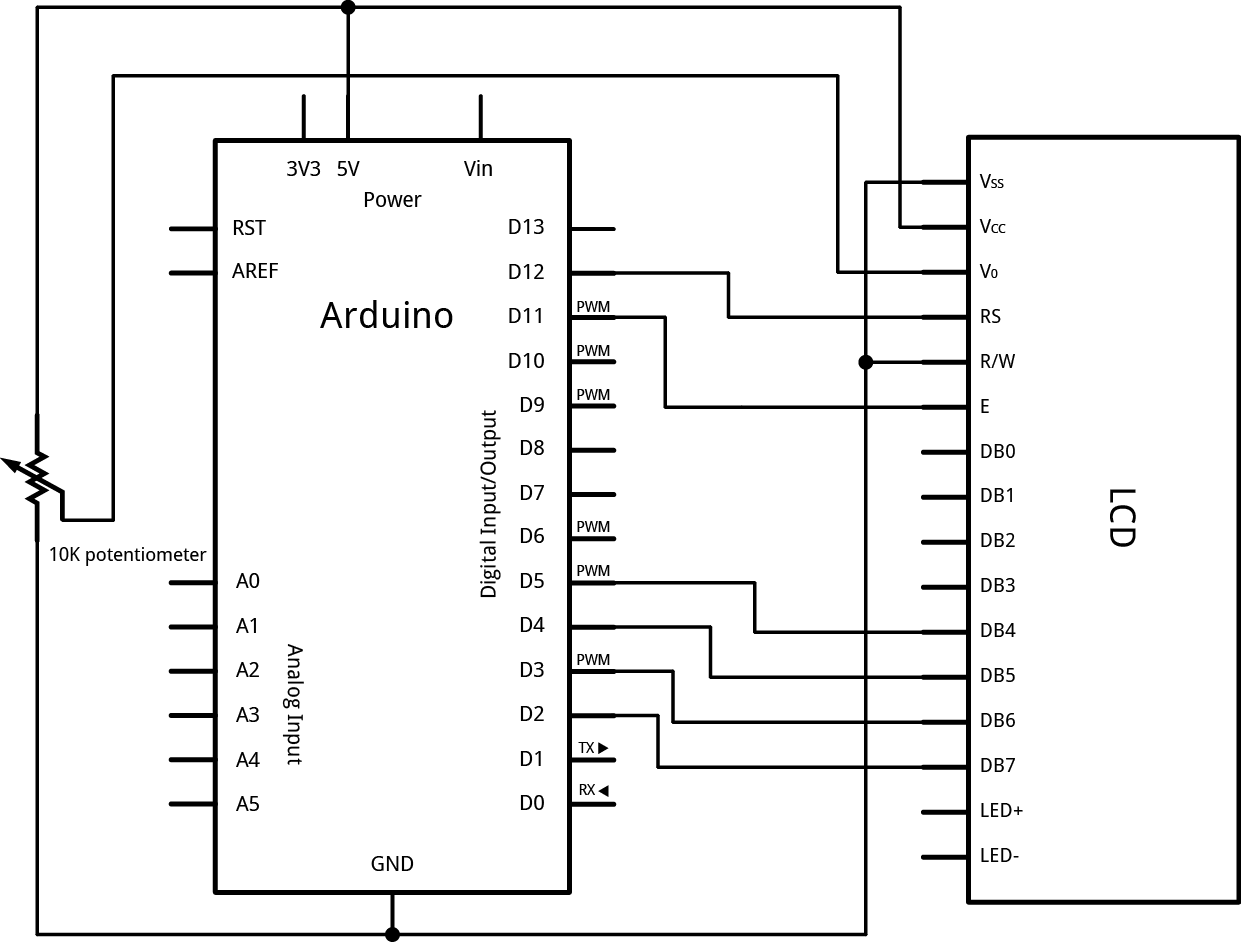
Принципиальная схема подключения данного дисплея к плате Arduino Uno R3 выглядит следующим образом.

Рисунок 5 – Принципиальная схема подключения LCD 1602 к Arduino Uno R3.

Для полноценной работы дисплея подключим необходимые выводы:

**1 (VSS) -** Питание контроллера (-) ⇨ GND

**2 (VDD)** – Питание контроллера (+) ⇨ питание + 5В

**3 (VO)** - Вывод управления контрастом ⇨ средний выход потенциометра. Два оставшихся выхода потенциометра в землю (GND) и на питание (+5B).

**4 (RS)** -  Выбор регистра ⇨ 11 pin Arduino

**5 (R**/**W)** - Чтение/запись ⇨ GND (режим записи)

**6 (Е)** - Еnable (строб по спаду) ⇨ 10 pin Arduino

**7-10 (DB0-DB3)** - Младшие биты 8-битного интерфейса. Не подключены, т.к. использование 8-битного режима на данном дисплее не целесообразно. Для его работы требуется на 4 ноги больше, а выигрыша в скорости практически нет т.к. частота обновления данного дисплея упирается в предел до 10раз в секунду.

**11-14 (DB4-DB7)** - Старшие биты интерфейса ⇨ pin 5, 4, 3, 2 Arduino соответственно.

**15 (A)** -  Анод (+) питания подсветки ⇨ через резистор 220 Ом на питание +5В.

**16 (K)** - Катод (-) питания подсветки ⇨ GND.

Подключение устройств

Все устройства подключены ко своим входам. В приложении 2 представлена макетная схема всего курсового проекта, созданная в программе Fritzing.

**Программная составляющая**

Листинг программы представлен в приложении 3.

Пользователь вводит математическое выражение с ограничением в 24 символа. Предусмотрена защита от некорректного ввода выражения. Например, нельзя ввести после плавающей запятой ещё одну запятую. Нажав на кнопку «С» экран очищается, чтобы заново ввести выражение. Нажав на кнопку «←» стирается последний введённый символ. По нажатию на «=» производится вычисление ответа выражения.

Вычисление производится следующим образом. Массив выражения переводится в выражения вида обратной польской записи (постфиксная нотация) по алгоритму Эдсгера Дейкстры «Сортировочная станция».

Функция return\_case по текущему символу выражения и последнему элементу промежуточного стека возвращает номер случая. Возможные случаи показаны в таблице на рисунке 6. Символом «|» в таблице обозначается «ничего», отсутствие символов.

Рисунок 6 – Таблица определения номера случая.

Функции case\_0, case\_1, case\_2, case\_3 в соответствующем случае выполняют определённые действия:

* Case\_0 – добавление цифр в результирующий стек обратной польской записи.
* Case\_1 – добавление элемента в промежуточный стек.
* Case\_2 – перенос последнего элемента промежуточного стека в результирующий стек обратной польской записи.
* Case\_3 – текущий элемент выражения и последний элемент промежуточного стека удаляются.
* Case\_4 и Case\_5 – остановка, перевод завершён.

Далее, с помощью стека подсчёта, цифры формируются в числа и, по ходу встречи символов операций, считаются.

Функция Pinmode (pin, mode) устанавливает режим работы заданного вход/выхода(pin) как входа или как выхода.

Порты Arduino установленные в режим **INPUT (вход)** находятся в высокоимпедансном состоянии. Это означает то, что порт ввода дает слишком малую нагрузки на схему, в которую он включен. Эквивалентом внутреннему сопротивлению будет резистор 100 МОм подключенный к выводу микросхемы. Таким образом, для перевода порта ввода из одного состояния в другое требуется маленькое значение тока. Это позволяет применять выводы микросхемы для подключения различных датчиков, но не питания.

Порт, установленный в режим выхода — **OUTPUT (выход),** находится в низкоимпедансном состояние. Он может пропускать через себя довольно большой ток, до 40 mA, достаточный для запитывание внешней цепи, например, светодиода. В этом состоянии порт может быть поврежден как замыкании на землю так и на питание 5В. Тока с порта микроконтроллера не достаточно для питания моторов и сервоприводов напрямую.

Функция digitalWrite(pin, value) подает HIGH или LOW значение на цифровой вход/выход (pin).

Если вход/выход (pin) был установлен в режим выход (OUTPUT) функцией pinMode(), то для значения HIGH напряжение на соответствующем вход/выходе (pin) будет 5В и 0В(земля) для LOW.

Если вход/выход (pin) был установлен в режим вход (INPUT), то функция digitalWrite со значением HIGH будет активировать внутренний 20K нагрузочный резистор. Подача LOW в свою очередь отключает этот резистор.

Отладка программы велась с помощью встроенного класса Serial, предназначенного для работы с аппаратными UART контроллерами в Arduino UNO. Он предназначен для управления обменом данными через UART.