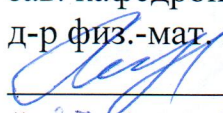


Министерство образования и науки Российской Федерации  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
Физико-Технический Факультет  
Кафедра прикладной газовой динамики и горения (ПГДиГ)

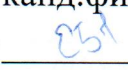
ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК  
Руководитель ООП,  
зав. кафедрой ПГДиГ,  
д-р физ.-мат. наук, профессор  
 Г. Р. Шрагер  
« 27 » 05 2017 г.

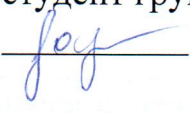
## МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

### ДРАЙВЕР ШАГОВОГО ДВИГАТЕЛЯ НА ОСНОВЕ МИКРОКОНТРОЛЛЕРА ATMEGA8

по основной образовательной программе подготовки магистров  
направление подготовки 15.04.06 – Мехатроника и робототехника

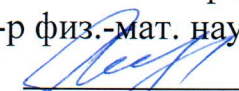
Хотулёв Владимир Александрович

Научный руководитель:  
канд. физ.-мат. наук, доцент  
 Е. И. Борзенко  
« 27 » 05 2017 г.

Автор работы:  
студент группы № 10513  
 В.А. Хотулёв

Томск 2017

Министерство образования и науки Российской Федерации  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
Физико-Технический Факультет  
Кафедра прикладной газовой динамики и горения (ПГДиГ)

УТВЕРЖДАЮ  
Руководитель ООП,  
зав. кафедрой ПГДиГ,  
д-р физ.-мат. наук, профессор  
 Г.Р. Шрагер  
« 27 » 05 2017 г.

**Задание**

на выпускную квалификационную работу магистра  
студента группы № 10513, Хотулёв Владимир Александровича

1. Тема: Драйвер шагового двигателя на основе микроконтроллера Atmega 8.
2. Цель: Создание драйвера управления биполярным шаговым двигателем, на основе микроконтроллера ATmega8.
3. Руководитель научно-исследовательской работы: доцент, канд.физ.-мат.наук, Борзенко Евгений Иванович.
4. Срок сдачи выпускной квалификационной работы:  
на кафедре « 26 » 05 2017 г.,  
в ГЭК « 09 » 06 2017 г.
5. Календарный план проведения выпускной квалификационной работы:

| №» | Название раздела                                                               | Срок исполнения           |
|----|--------------------------------------------------------------------------------|---------------------------|
| 1  | Обзор литературы о микроконтроллерах и способах их применения.                 | 01.09.2016-<br>22.09.2016 |
| 2  | Реализация драйвера шагового двигателя на основе микроконтроллера ATmega8.     | 03.11.2016-<br>14.02.2017 |
| 3  | Тестирование программы и работы драйвера шагового двигателя на основе ATmega8. | 15.03.2017-<br>10.05.2017 |
| 4  | Подготовка отчета по выпускной квалификационной работе.                        | 10.05.2017-<br>01.06.2017 |

6. Рекомендуемая литература для изучения:

1. Реализация в ATmega8 //Radio Aktiv[Электронный ресурс].URL: <http://radioaktiv.ru/shems/digital/prog/27894-usbasp-usb-programmator-dlya-atmel-avr-mikrokontrollerov.html>
2. Использование интерфейса USART микроконтроллеров AVR //Radio Party. [Электронный ресурс]. URL: <http://radioparty.ru/prog-avr/program-c/307-lesson-usart-avr>.
3. AVR. Учебный курс. Передача данных через UART //Easy Electronics. [Электронный ресурс]. URL:<http://easyelectronics.ru/avr-uchebnyj-kurs-peredacha-dannyx-cherez-uart.html>.
4. CP2102 - преобразователь USB-UART // Easy Electronics. [Электронный ресурс]. URL:<http://we.easyelectronics.ru/electro-and-pc/cp2102-preobrazovatel-usb-uart-za-45-rublej.html>.
5. Таблица символов ASCII // Industrial Nets. [Электронный курс]. URL: <http://www.industrialnets.ru/files/misc/ascii.pdf>.
6. Настройка скорости UART // Micri Sin. [Электронный ресурс]. URL: <http://microsin.net/programming/avr/atmega2560-working-with-uart.html>.
7. Bill Earl. All about Stepper Motors [Text] // Adafruit learning system, 2015.
8. Stepper Motor Basics [Text] // Industrial Circuits Application Note – P. 1-6.

Научный руководитель:

доцент кафедры ПГДиГ,

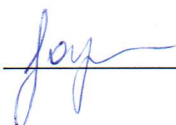
канд. физ.-мат. наук, доцент



Е. И. Борзенко

Задание принял к исполнению:

«01» сентября 2016 г.



Хотулёв В.А.

## ОГЛАВЛЕНИЕ

|                                                            |    |
|------------------------------------------------------------|----|
| ВВЕДЕНИЕ .....                                             | 5  |
| 1 Общая схема .....                                        | 5  |
| 1.1 Шаговый двигатель ПБМТ 200-265-1.....                  | 8  |
| 1.2 Драйвер шагового двигателя .....                       | 15 |
| 1.3 Экран LCD 1602A .....                                  | 16 |
| 1.4 UART Universal Asynchronous Receiver Transmitter ..... | 18 |
| 1.4.1 Аппаратная часть .....                               | 18 |
| 1.4.2 Реализация в Atmega8 .....                           | 19 |
| 1.4.3 Настройка скорости передачи данных по UART .....     | 24 |
| 1.5 Переходник USB-UART CP2102.....                        | 25 |
| 1.6 Программа Terminal 1.9b.....                           | 26 |
| Тестирование.....                                          | 26 |
| ЗАКЛЮЧЕНИЕ.....                                            | 31 |
| Список использованных источников литературы .....          | 32 |
| Приложение.....                                            | 33 |

## ВВЕДЕНИЕ

В настоящее время в системах управления и обработки данных все чаще применяются микроконтроллеры, решающие широкий спектр задач. Микроконтроллеры используются во всех сферах жизнедеятельности человека, устройствах, которые окружают его. Простота подключения и большие функциональные возможности. С помощью программирования микроконтроллера можно решить многие практические задачи аппаратной техники.

Шаговые двигатели играют большую роль в развитии техники и электронике. Шаговые двигатели хорошо используются в принтерах, приводах дисководов, автомобильных приборных панелях, станках ЧПУ и других приложениях, требующих высокой точности позиционирования. В связи с этим управление биполярного шагового двигателя играет важную роль.

Цель работы – разработка драйвера управления биполярным шаговым двигателем, на основе микроконтроллера ATmega8 и модулем L298n, с функцией обмена информацией между микроконтроллером и персональным компьютером. Задачи – организация обмена информацией, путем передачи данных с микроконтроллера ATmega8 по UART каналу на PC при помощи переходника CP2102 подключенному к USB-порту PC, для управления микросхемой L298n и шаговым двигателем.

### 1 Общая схема

Рисунок 1 иллюстрирует общую схему, для передачи данных с микроконтроллера ATmega8, который по UART каналу связывается с персональным компьютером при помощи переходника CP2102 подключенному к USB-порту персонального компьютера. С персонального компьютера передается значение скорости вращения, которое в свою очередь выводится на экран LCD 1602A. Далее согласно заданной скорости подаются сигналы управления на усилитель L298n, а с усилителя подается сигнал на шаговый двигатель ПБМТ 200-265-1.

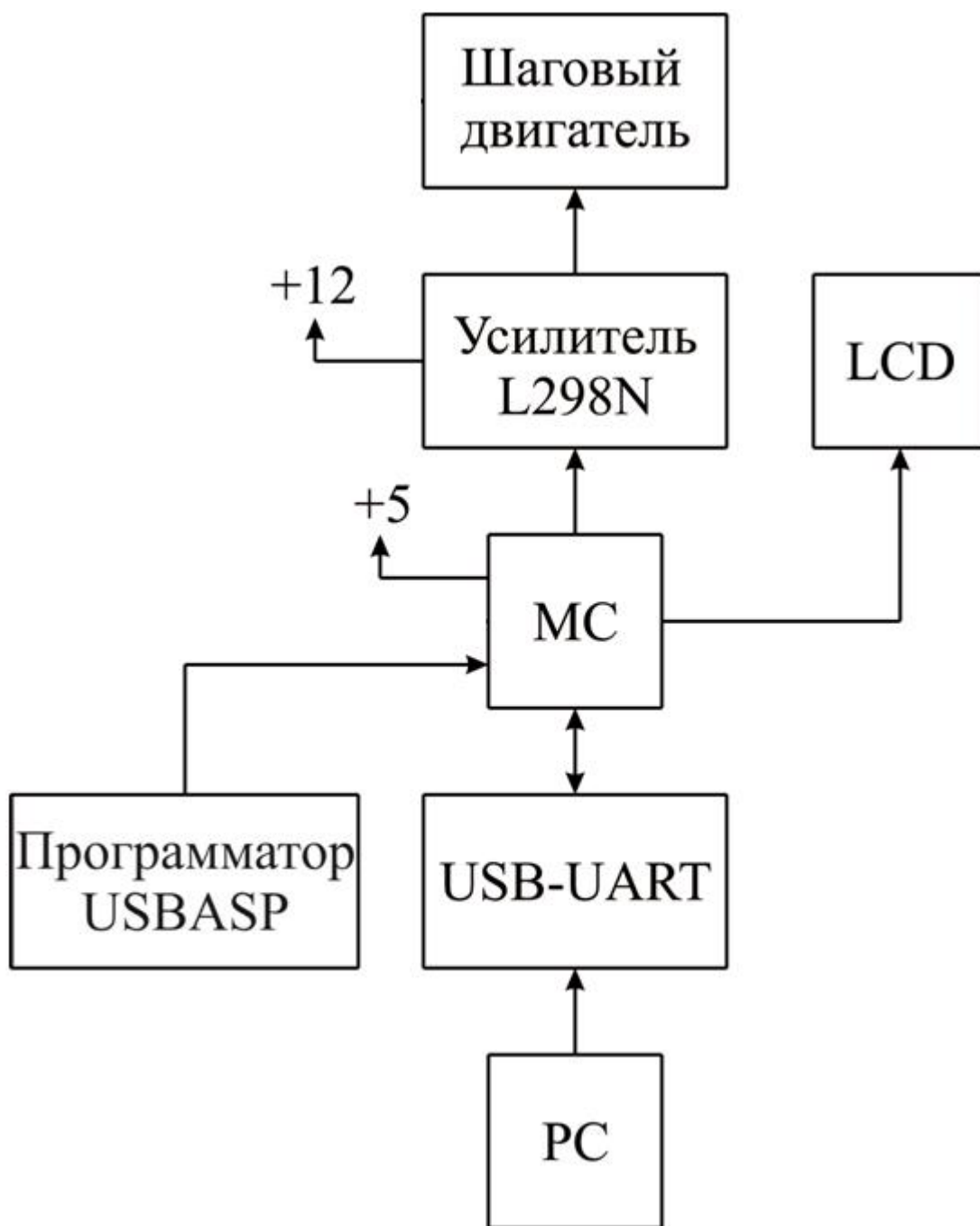


Рисунок 1 – Общая схема

На рисунке 2 – представлен шаблон двухсторонней платы, к которой подключается микроконтроллер ATmega8, LCD модуль, программатор USBASP, мотор и COM порт.

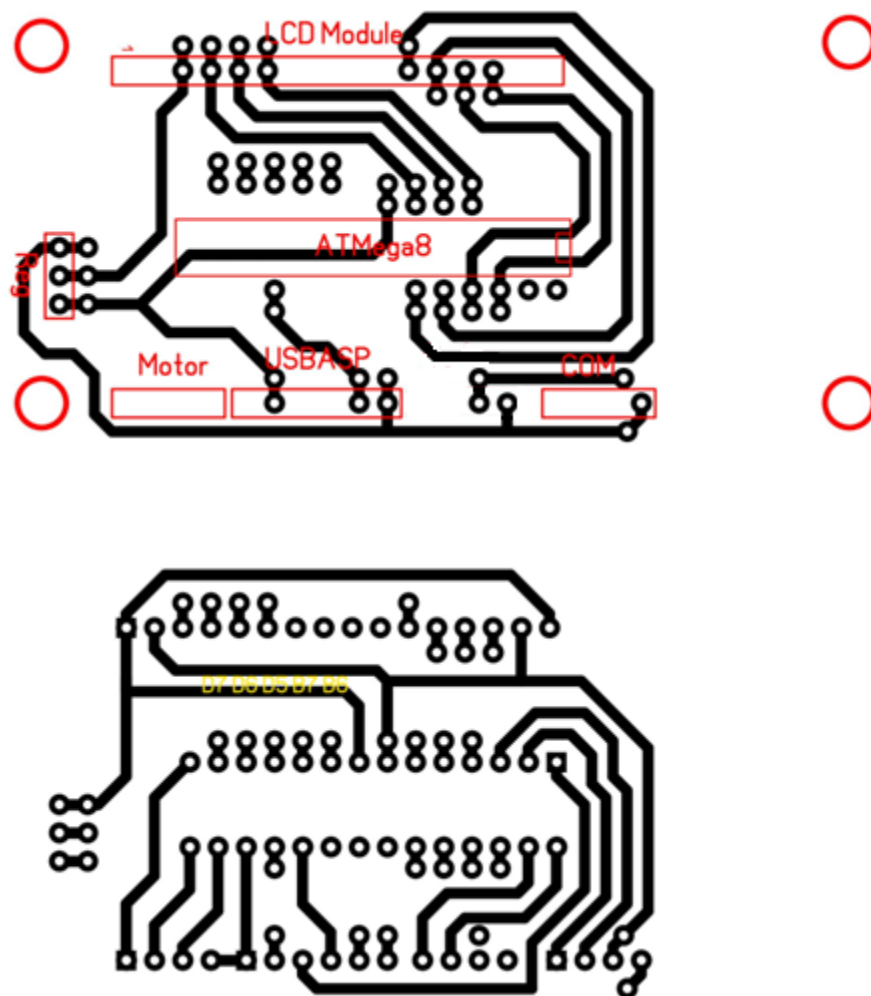


Рисунок 2 – Макетная плата

Создание перчатной макетной платы. Принцип этой технологии заключается в том, чтобы перенести рисунок разводки электрической схемы, а именно дорожки соединений электрических элементов и их расположение на стеклотекстолит. Стеклотекстолит – это слоеный пластик, а основой для него является стеклоткань, имеющая электроизоляционные свойства. Слои состоят из полимеров, таких как – бакелит, эпоксидная смола и т.д. Поверхность текстолита покрыта медной фольгой. Текстолит бывает, как односторонний (медная фольга с одной стороны), так и двухсторонний (медная фольга с обеих сторон). Для того чтобы перевести рисунок на текстолит, нужно использовать фотобумагу. Печатать рисунок нужно именно на лазерном принтере, поэтому технология называется ”лазерной”. Нужно помнить о том, что изображение должно быть зеркальным. Печатается

разводка платы в масштабе 1:1. Во время ЛУТ метода нужно почистить текстолит от грязи, неровностей, а также жирных пятен. Грязь и неровности убираются с помощью наждачной бумаги нулевой зернистости, так как она не портит медную пленку. Жирные пятна убираются с помощью технического спирта или ацетона. Далее на обработанный текстолит накладывается рисунок будущей платы и подвергается термическому воздействию с помощью утюга или ламинатора. Рисунок переводится в течение 4-6 минут. После нанесения рисунка на текстолит нужно “протравить” плату. Это означает что плату надо поместить в раствор хлорного железа. В течение некоторого времени медная фольга сойдет со стеклотекстолита, и останется только медные дорожки, которые были закрашены тоном. После процесса “травления” вся медь сошла со стеклотекстолита, осталась только та, на которой был нанесен тон. Чтобы снять тонер нужно обработать плату ацетоном или наждачной бумагой нулевой зернистости. Далее идет подготовка платы к пайке. Для этого нужно подготовить отверстия для радиоэлементов. Чтобы просверлить отверстия используем бормашину (мини-дрель). Рабочий диаметр сверла составляет 0,9-1 мм.

### 1.1 Шаговый двигатель ПБМТ 200-265-1

Шаговый двигатель - синхронный бесколлекторный электродвигатель с несколькими обмотками, в котором ток, подаваемый в одну из обмоток статора, вызывает фиксацию ротора. Последовательная активация обмоток двигателя вызывает дискретные угловые перемещения (шаги) ротора.

Конструктивно шаговые электродвигатели состоят из статора, на котором расположены обмотки возбуждения, и ротора, выполненного из магнито-мягкого или из магнито-твёрдого материала.

Шаговые двигатели с магнитным ротором позволяют получать больший крутящий момент и обеспечивают фиксацию ротора при обесточенных обмотках.

Шаговые двигатели бывают следующих видов:

Биполярный - 2 обмотки, по одной на каждую фазу, для управления можно использовать схему на 2 Н-моста или один полу-мост с двуполярным питанием;

Униполярный 2 обмотки, каждая с отводом от середины, удобно переключать фазы сменой половинок каждой из обмоток, упрощает схему драйвера (4 ключа), а также использовать как биполярный без использования отводов от обмоток;

С четырьмя обмотками - универсальный, подключив обмотки соответствующим образом можно использовать как биполярный или униполярный двигатель.



Рисунок 3 – Шаговый двигатель ПБМТ 200-265-1

Основные характеристики шагового двигателя ПБМГ 200-265-1:

Тип двигателя: Шаговый, униполярный.

Угол поворота за один шаг: 1.8 градуса.

Количество шагов на полный оборот вала: 200.

Количество выводов: 6.

Количество обмоток: 2 со средней точкой.

Напряжение питания: 12В.

Сопротивление обмоток: 70Ом.

Потребляемый ток: 0,17А.

Одним из компонентов для управления шагового двигателя является Н-мост. Это электронная схема, которая даёт возможность приложить напряжение к нагрузке в разных направлениях. Эта схема очень часто используется в робототехнике и игрушечных машинах, чтобы изменять направление вращения мотора.

При проектировании станков, роботов и других автоматизированных устройств возникает необходимость управлять электродвигателем постоянного тока или же катушками шагового двигателя. Для того, чтобы иметь возможность управлять обмоткой двигателя и заставить его вал вращаться в разные стороны, необходимо выполнять коммутацию с переполюсовкой. Для подобной цели используется так называемый "Н-мост".

Схема работы Н-моста.

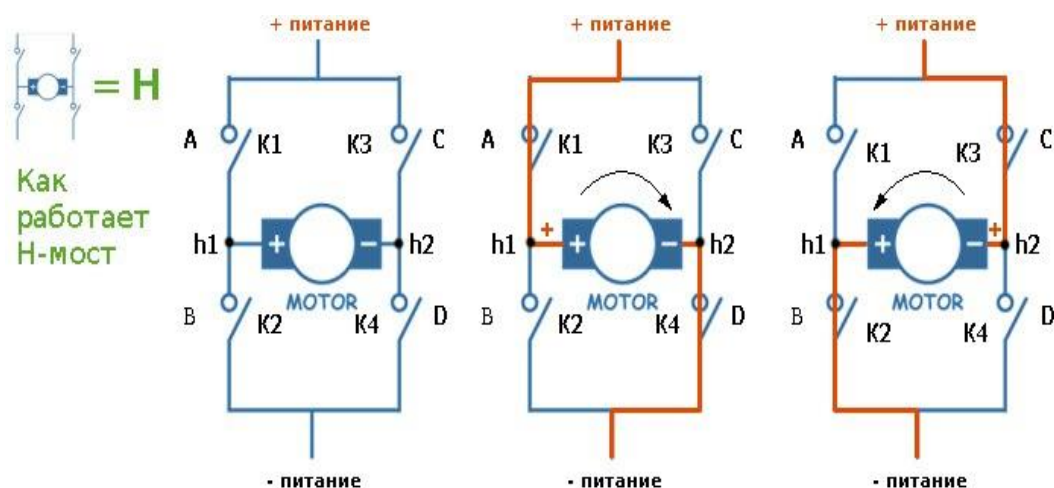


Рисунок 4 – Схема работы Н-моста

K1, K2, K3, K4 - управляемые ключи

A, B, C, D - сигналы управления ключами

Идея этой схемы очень проста: Если ключи K1 и K4 замкнуты, а ключи K2 и K3 разомкнуты, то к точке h1 оказывается приложено напряжение питания, а точка h2 замыкается на общий провод. Ток через нагрузку в этом случае течёт от точки h1 к точке h2. Если сделать наоборот, - ключи K1 и K4 разомкнуть, а ключи K2 и K3 замкнуть, то полярность напряжения на нагруз-

ке изменится на противоположную, - точка h1 окажется замкнута на общий провод, а точка h2 - на шину питания. Ток через нагрузку теперь будет течь от точки h2 к точке h1.

Четыре сигнала управления:

1.  $V_0=1$ ;  $V_1=0$ , то 1 Н-мост ( $K_1$ -  $K_4$  ключи замкнуты);  
 $V_2=V_3=0$ , второй Н-мост отключен;
2.  $V_0=V_1=0$ , первый Н-мост отключен;  
 $V_2=1$ ;  $V_3=0$ , то второй Н-мост ( $K_2$ -  $K_3$  ключи замкнуты);
3.  $V_1=1$ ;  $V_0=0$ , то 1 Н-мост ( $K_1$ -  $K_4$  ключи замкнуты);  
 $V_3=V_2=0$ , второй Н-мост отключен;
4.  $V_1=V_0=0$ , первый Н-мост отключен;  
 $V_3=1$ ;  $V_2=0$ , то второй Н-мост ( $K_2$ -  $K_3$  ключи замкнуты);

Кроме смены полярности, h-мост, в случае управления электродвигателем, добавляет нам и ещё один бонус - возможность закоротить концы обмоток, что ведёт к резкому торможению нашего двигателя. Такой эффект можно получить замкнув одновременно либо ключи  $K_1$  и  $K_3$ , либо ключи  $K_2$  и  $K_4$ .

Для внедрения в нашу плату Н-моста используется микросхема драйвера L 298n.

L298N - представляет собой двойной мостовой драйвер для управления двунаправленными нагрузками с токами до 2А и напряжением от 4.5 V до 46 V. Микросхема разработана для управления реле, соленоидами, двигателями постоянного тока и шаговыми двигателями. L298N имеет TTL совместимые входы. В L298 существует разделение электропитания для логической схемы и для нагрузки, что позволяет подключить нагрузку с меньшим или большим напряжением питания, чем у микросхемы, а также уменьшает помехи. Микросхемы L298N имеют встроенную защиту от перегрева. Выходы микросхемы отключаются при нагреве до температуры около  $+70^{\circ}\text{C}$ .

## Алгоритм управления шаговым двигателем.

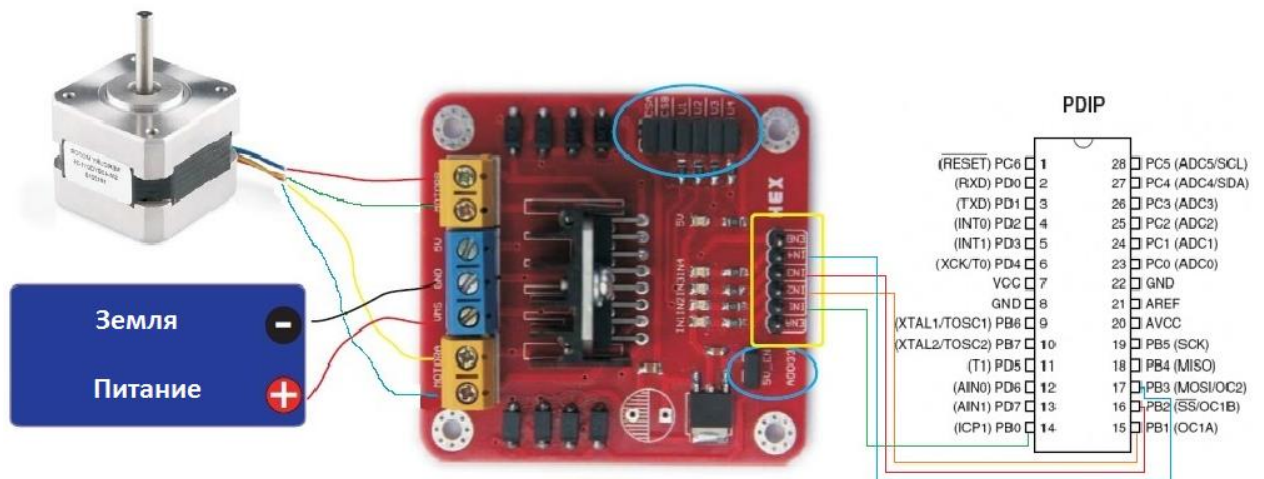


Рисунок 5- подключение шагового двигателя к схеме L298n, которая связывается с микроконтроллером.

Логика управление шаговым двигателем предполагает включение обмоток в определенной последовательности через 2 Н-моста в схеме L298N.

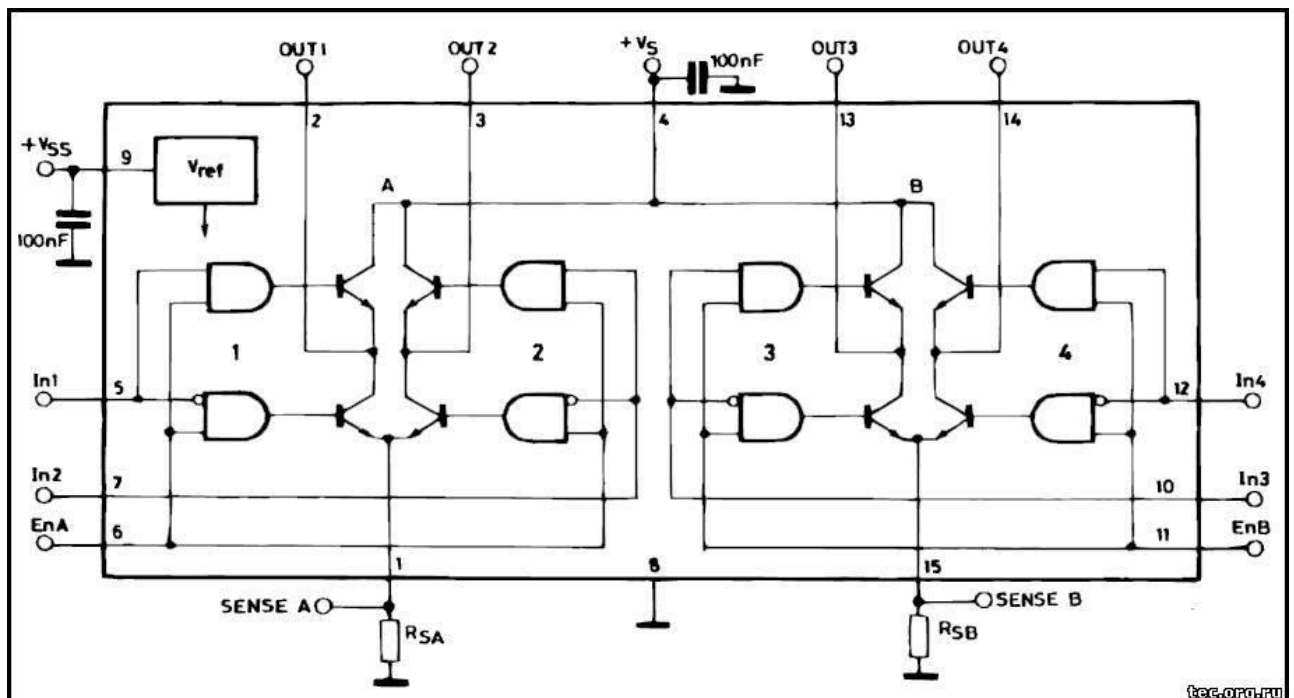


Рисунок 6 – блок-диаграмма мостовой схемы L298N

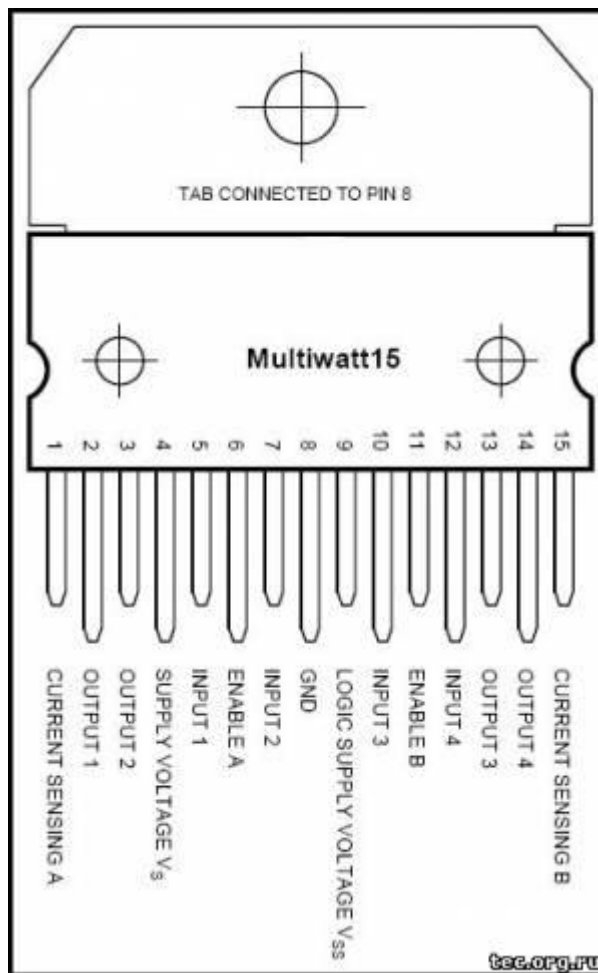


Рисунок 7 – расположение выводов микросхемы L298N

Назначения выводов L298N:

1. CURRENT SENSING A – Контроль тока 1 и 2 каналов;
2. OUTPUT 1 – выход канала 2;
3. OUTPUT 2 – выход канала 3;
4. SUPPLY VOLTAGE  $V_s$  – питание нагрузки;
5. INPUT 1 – вход 1 канала;
6. ENABLE A – разрешение работы 1 и 2 канала;
7. INPUT 2 – вход 2 канала;
8. GND – общий вывод;
9. LOGIC SUPPLY VOLTAGE  $V_{ss}$  – питание логики (+5V);
10. INPUT 3 – вход 3 канала;
11. ENABLE B – разрешение работы 3 и 4 канала;

12. INPUT 4 – вход 4 канала;
13. OUTPUT 3 – выход 3 канала;
14. OUTPUT4 – выход 4 канала;
15. CURRENT SENSING B – контроль тока 3 и 4 каналов;

Таким образом, логика управления шаговым двигателем осуществляется следующим образом:

На вход I1 с микроконтроллера (Port B0) приходит логическая 1, при этом в остальных входах L298n приходит логический ноль, при этом схема активирует обмотки так, что якорь ориентируется относительно первой обмотки. На вход I3 с микроконтроллера (Port B2) приходит логическая 1, при этом в остальных входах L298n логический ноль, при этом схема активирует обмотки так, что якорь ориентируется относительно второй обмотки. Микроконтроллер подает управляющие сигналы переключения обмоток по прерыванию таймера с определенным интервалом. Таким образом, меняя настройки таймера, можно изменять скорость вращения якорем. Такой алгоритм работы МП, позволяет МП работать практически автономно от шагового двигателя (в отличии от алгоритма с использованием функции delay).

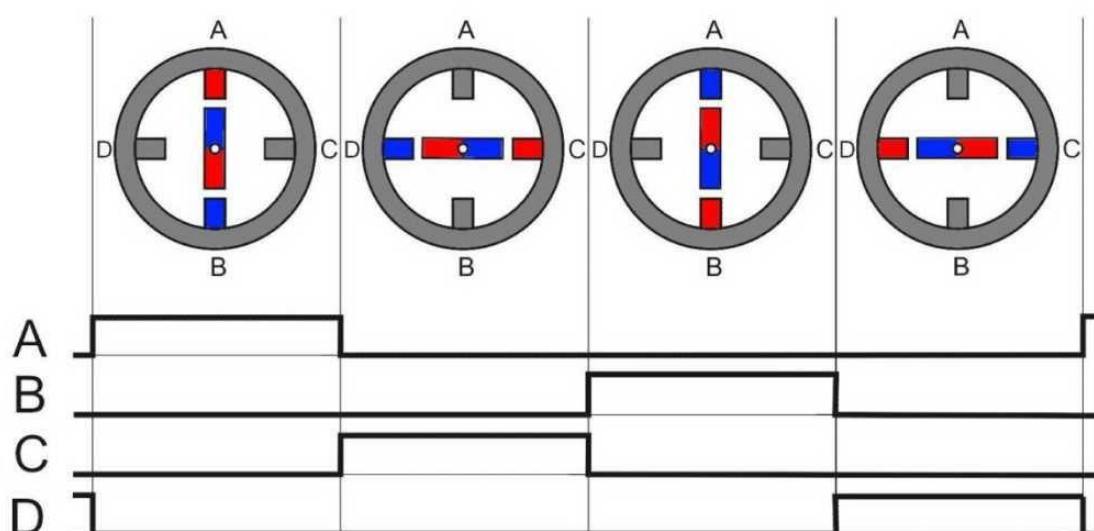


Рисунок 8 – схема управления шагового двигателя

Одна фаза на полный шаг. Точки равновесия ротора совпадают с полюсами статора.

## 1.2 Драйвер шагового двигателя

Интегральная микросхема L298 - это мощный универсальный мостовой драйвер для управления двигателями постоянного тока, шаговыми движками, электромагнитными реле и электромагнитами (соленоидами). В микросхеме содержится два H-моста, выполненных на силовых полевых транзисторах, а также логика совместимая с TTL.

Модуль L298N H-bridge можно использовать для двигателей, напряжение питания которых находится в диапазоне от 5 до 35 вольт. Кроме того, на многих подобных платах есть встроенный 5В регулятор.

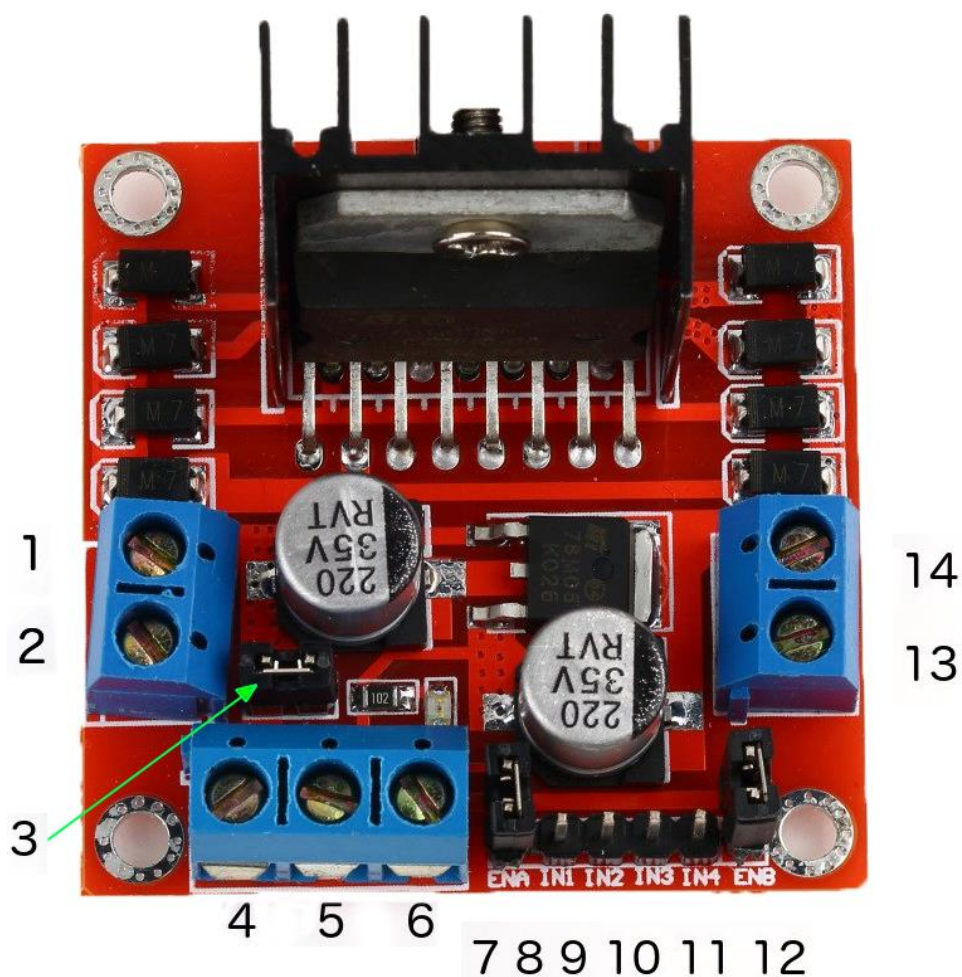


Рисунок 9 – Микросхема L298N и ее распиновка

Ниже приведены разъяснения к рисунку:

1. Для двигателя постоянного тока 1 “+” или для шагового двигателя A+;
2. Для двигателя постоянного тока 1 “-” или для шагового двигателя A-;

3. Коннектор на 12 вольт. Коннектор можно снять, если используем напряжение питания больше 12 вольт;
4. Питания двигателя обеспечивается с этого выхода. Максимальное напряжение питания постоянным током 36 вольт. Если напряжение больше 12 вольт, надо разомкнуть контакты на 3 коннекторе;
- 5.GND – земля;
6. Питание 5 вольт, если коннектор на 12 вольт замкнут ;
7. Коннектор для двигателя постоянного тока 1. Можно подключить к ШИМ-выходу для управления скоростью двигателя постоянного тока;
8. IN1 Управление ключами Н моста;
9. IN2;
10. IN3;
- 11.IN4;
12. Коннектор для двигателя постоянного тока 2. В случае использования шагового двигателя, подключать на него ничего не надо. Можно подключить к ШИМ-выходу для управления скоростью двигателя постоянного тока;
- 13.Двигатель постоянного тока 2 “+” или шаговый двигатель В+;
- 14.Двигатель постоянного тока 2 “-” или шаговый двигатель В- .

### 1.3 Экран LCD 1602A

Размерность дисплеев на контроллере HD44780 может быть различной, управляться они будут одинаково.

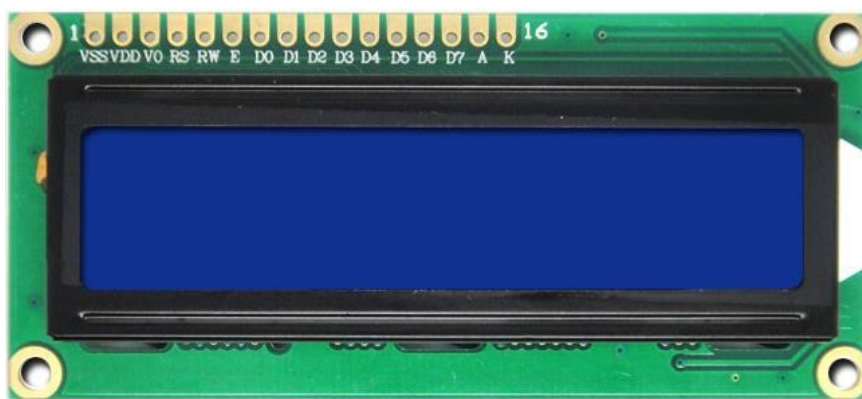


Рисунок 10 – экран LCD 1602A

Самые распространенные размерности 16x02 (т.е. по 16 символов в двух строках) либо 20x04. Разрешение же самих символов - 5x8 точек.

Сам же дисплей может работать в двух режимах:

1. 8-битный режим - для этого используются и младшие и старшие биты (VB0- DB7);
2. 4-битный режим - для этого используются и только младшие биты (VB4- DB7);

Использование 8-битного режима на данном дисплее не целесообразно. Для его работы требуется на 4 вывода больше, а выигрыша в скорости практически нет т.к. частота обновления данного дисплея упирается в предел < 10 раз в секунду. Для вывода текста необходимо подключить выводы RS, E, DB4, DB5, DB6, DB7 к выводам контроллера. Их можно подключать к любым пинами, главное в коде задать правильную последовательность.

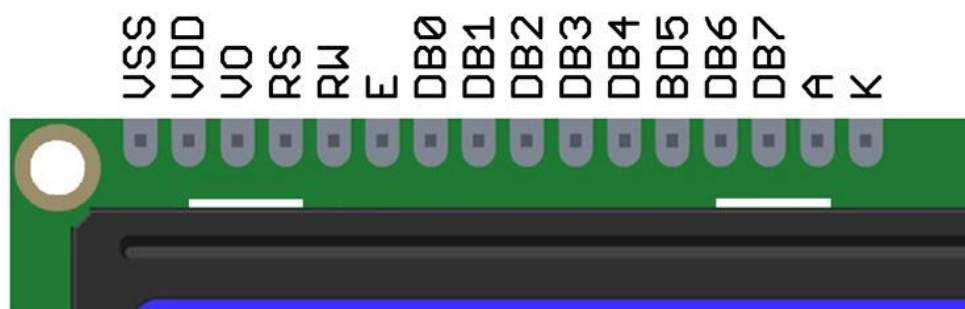


Рисунок 11 – Распиновка LCD 1602A

На дисплее имеется 16pin разъем для подключения. Выводы промаркированы на тыльной стороне платы.

Выводы дисплея:

1. (VSS) - Питание контроллера (-) ;
2. (VDD) - Питание контроллера (+) ;
3. (VO) - Вывод управления контрастом ;
4. (RS) - Выбор регистра ;
5. (R/W) - Чтение/запись ( режим записи при соединении с землей) ;
6. (E) - Enable (строб по спаду) ;
- 7-10. (DB0-DB3) - Младшие биты 8-битного интерфейса ;

- 11-14. (DB4-DB7) - Старшие биты интерфейса ;
- 15. (A) - Анод (+) питания подсветки ;
- 16. (K) - Катод (-) питания подсветки ;

#### 1.4 UART Universal Asynchronous Receiver Transmitter

Микроконтроллеры AVR имеют в своем составе модуль полнодуплексного универсального асинхронного приемопередатчика UART (в семействе MEGA универсальный синхронный/асинхронный приемопередатчик USART).

Через него осуществляется прием и передача информации, представленной последовательным кодом, поэтому модуль UART часто называют также последовательным портом. С помощью этого модуля микроконтроллер может обмениваться данными с различными внешними устройствами. Поток данных, передаваемых по каналу UART, представляет собой совокупность посылок или кадров. Каждый кадр содержит стартовый бит, восемь битов данных и стоповый бит. Стартовый бит имеет уровень логического 0, стоповый - уровень логической 1. Скорость передачи данных может варьироваться в широких пределах, причем высокие скорости передачи могут быть достигнуты даже при относительно низкой тактовой частоте микроконтроллера.

##### 1.4.1 Аппаратная часть

Соединение осуществляется крест-накрест, т.е. приемник с передатчиком, как показано на рисунке 1.

Tx передающая линия (Transmitted Data).

Rx принимающая линия (Received Data).

Модуль стандартного приемопередатчика обеспечивает передачу данных по линиям TxD и RxD с программным и аппаратным управлением потоком данных.

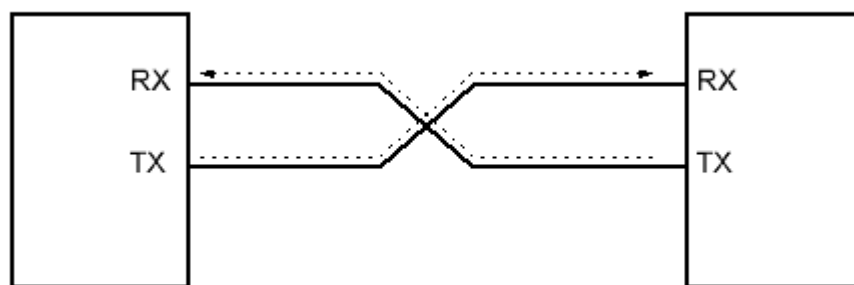


Рисунок 12 – Подключение UART

#### 1.4.2 Реализация в Atmega8

На рисунке 13 представлена схема выводов микроконтроллера Atmega8. Линия Rx соответствует выводу PD0, соответствующему контакту 2 DIP корпуса. Линия Tx соответствует выводу PD1, соответствующему контакту 3 DIP корпуса.

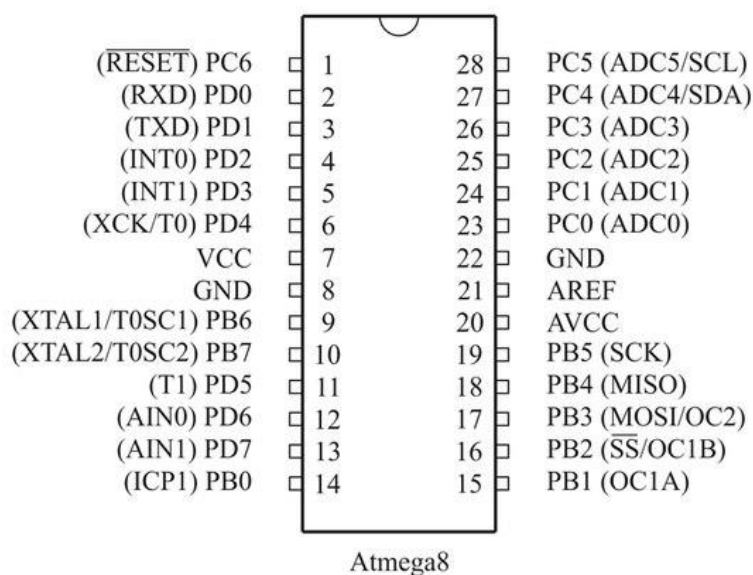


Рисунок 13 – Микроконтроллер Atmega8

Протокол UART ( Universal Asynchronous Receiver-Transmitter) является одним из самых распространённых протоколов передачи данных между устройствами. Стандартный размер послыки при асинхронной передаче данных составляет 10 бит. Если по линии ничего не передаётся, то она находится в состоянии логической единицы. Передача начинается со стартового бита,

который равен 0. Затем следуют 8 бит данных. Завершается передача стоп-битом, который равен 1.

Передача данных в UART осуществляется по одному биту в равные промежутки времени. Этот временной промежуток определяется заданной скоростью UART и для конкретного соединения указывается в бодах, что соответствует количеству бит в секунду. Существует общепринятый ряд стандартных скоростей: 300; 600; 1200; 2400; 4800; 9600; 19200; 38400; 57600; 115200; 230400; 460800; 921600 бод;

Бод единица измерения символьной скорости, количество изменений информационного параметра несущего периодического сигнала в секунду. Для двоичного кода считают что бод это количество бит, переданное в секунду. Кроме этого, бодами выражают полную ёмкость канала, включая служебные символы (биты), если они есть. Эффективная же скорость канала выражается другими единицами, например битами в секунду (бит/с, bps).

Скорость ( $S$ , бод) и длительность бита ( $T$ , секунд) связаны соотношением  $T=1/S$ .

Уровню логической единицы и нуля соответствует уровням TTL:

+5В это логическая 1ца;

0В это логический 0 .

Помимо собственно информационного потока, UART автоматически вставляет в поток синхронизирующие метки, так называемые, стартовый и стоповый биты. При приеме стартовый и стоповые биты удаляются из потока. Обычно стартовый и стоповый биты обрамляют один байт информации (8 бит), однако встречается реализация UART, которые позволяют передавать по 5,6,7,8 или 9 бит. Называемые стартом и стопом биты являются минимальной посылкой. Некоторые реализации UART позволяют вставлять два стоповых бита при передаче, для уменьшения вероятности рассинхронизации приемника и передачи при плотном трафике. Приемник игнорирует второй стоповый бит, воспринимая его как короткую паузу на линии. Пассивным (в отсутствие потока данных) состоянием входа и выхода UART является

ся логическая 1. Стартовый бит всегда логический 0, поэтому приемник UART ждет перепада из 1 в 0 и отсчитывает от него временной промежуток в половину длительности бита (середина передачи стартового бита). Если в этот момент на входе все еще 0, то запускается процесс приема минимальной посылки. Для этого приемник отсчитывает 9 битовых длительностей подряд (для 8-битных данных) и в каждый момент фиксирует состояние входа. Первые 8 значений являются принятыми данными, последнее значение проверочное (стоп-бит). Значение стоп-бита всегда 1, если значение стоп-бита 0, UART фиксирует ошибку.

Бит контроля четности позволяет проверить, не произошел ли сбой при передаче. Битом контроля чётности называют контрольный бит, принимающий значения '0' или '1' и служащий для проверки общей чётности двоичного числа (чётности количества единичных битов в числе).

На рисунке 14 представлена передача байта 10000010.

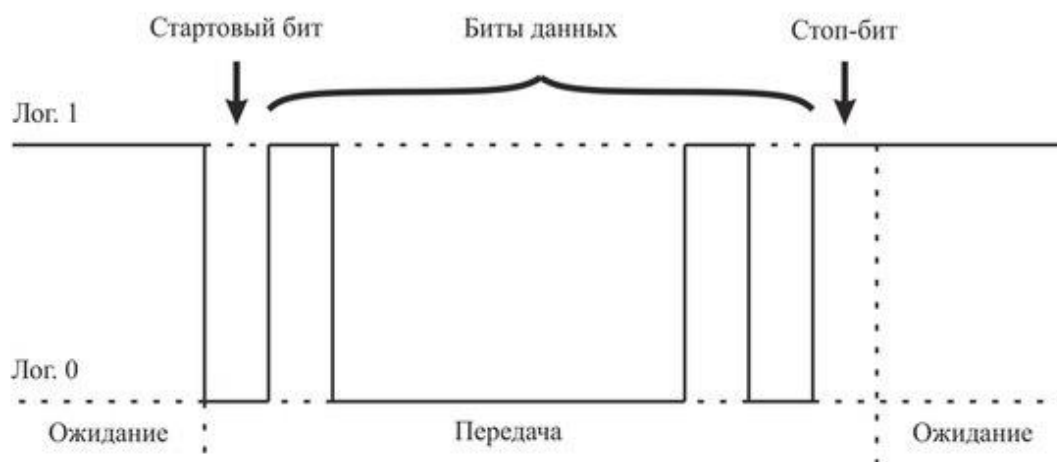


Рисунок 14 – Передача байта

Для настройки и работы с UART в mega8 выделено 5 регистра:

- 1) Регистр управления UCSRB: разрешает/запрещает прерывания от UART, управляет функциями приёмопередатчика;
- 2) Регистр состояния UCSRA;

- 3) Регистр данных UDR - регистр где хранятся посторонние данные, представляет собой 2 регистра, расположенных по одному адресу (1 регистр для передачи, 2 - для приёма данных);
- 4) Регистр UBRR отвечает за скорость передачи данных. Задаются значения младшего (UBBRL) и старшего (UBBRH) байтов регистра;
- 5) Биты регистра UCSRC задают формат кадра ;

Регистром называется функциональный узел, осуществляющий приём, хранение и передачу информации. Регистры состоят из группы триггеров, обычно D. По типу приёма и выдачи информации различают 2 типа регистров:

С последовательным приёмом и выдачей информации — сдвиговые регистры.

С параллельным приёмом и выдачей информации — параллельные регистры. Сдвиговые регистры представляют собой последовательно соединённую цепочку триггеров. Основной режим работы — сдвиг разрядов кода от одного триггера к другому на каждый импульс тактового сигнала.

По назначению регистры различаются на:

аккумулятор используется для хранения промежуточных результатов арифметических и логических операций и инструкций вводавывода;

флаговые хранят признаки результатов арифметических и логических операций;

общего назначения хранят операнды арифметических и логических выражений, индексы и адреса;

индексные хранят индексы исходных и целевых элементов массива;

указательные хранят указатели на специальные области памяти (указатель текущей операции, указатель базы, указатель стека);

сегментные хранят адреса и селекторы сегментов памяти; управляющие хранят информацию, управляющую состоянием процессора, а также адреса системных таблиц.

Регистр UCSRB

|       |       |       |      |      |      |      |      |
|-------|-------|-------|------|------|------|------|------|
| 7     | 6     | 5     | 4    | 3    | 2    | 1    | 0    |
| RXCIE | TXCIE | UDRIE | RXEN | CHR9 | CHR9 | RXB8 | TXB8 |

RXCIE=1: и прерывания разрешены (бит I=1 в регистре SREG) : прерывание по завершению приёма по UART разрешено;

TXCIE=1: и прерывания разрешены (бит I=1 в регистре SREG) : прерывание по завершению передачи по UART разрешено;

UDRIE=1: и прерывания разрешены (бит I=1 в регистре SREG) : прерывание по опустошению регистра данных UART разрешено;

RXEN=1: активация приёмника, вывод D0 становится входом UART;

TXEN=1: активация передатчика, вывод D1 становится выходом UART;

CHR9=1: длина передаваемой посылки становится равной 11 бит (9 бит данных + стартовый бит). + стопбит;

#### Регистр UCSRA

|     |     |      |      |    |    |   |   |
|-----|-----|------|------|----|----|---|---|
| 7   | 6   | 5    | 4    | 3  | 2  | 1 | 0 |
| RXC | TXC | UDRE | RXEN | FE | OR | - | - |

#### Регистр UCSRA информирует о состоянии UART

RXC=1: принятое слово данных перемещено из сдвигового регистра в регистр UDR. После чтения регистра UDR бит RXC сбрасывается в 0;

TXC=1: символ в сдвиговом регистре передан полностью (включая стоп-бит), и из регистра UDR не ожидается новый байт данных;

UDRE=1: содержимое регистра UDR передано в сдвиговый регистр, т.е. приёмопередатчик готов к передаче нового байта. Только после этого в UDR можно записывать новое значение. В противном случае можно потерять байт, который находился в UDR в ожидании отправки в сдвиговый регистр. Бит UDRE сбрасывается при записи данных в регистр UDR. При включении питания бит UDRE устанавливается в лог. 1;

FE=1: ошибка кадрирования. Ошибка кадрирования происходит, если стоп-бит не равен лог.1;

$OR=1$ : один из символов, переданный в UDR из сдвигового регистра не был прочитан перед поступлением следующего символа;

### 1.4.3 Настройка скорости передачи данных по UART

Скорость передачи данных по UART рассчитывается по формуле

$$UBRR = (F/(B*16))-1,$$

F тактовая частота микроконтроллера (например 8МГц);

B требуемая скорость, (например 9600бит/с);

$$UBRR = (8000000 / (9600*16)) - 1 = 51;$$

UBRR значение регистра UBRR (состоит из значений младшего и старшего байтов UBBRH+UBBRL), т.к. этот регистр 16-битный то доступ организован через два 8-ми битных регистра;

UBRRH = 0 - старший байт;

UBRRL = 51 - младший байт;

Если в результате вычисления UBRR получалось не целое число, то в регистр записывается округленное значение, частота обмена при этом будет отличаться от той, что была задана при расчете. Расхождение в частоте может приводить к сбоям при обмене данными, по этому надо задавать такую скорость обмена, что бы ошибки округления были минимальны.

Значения скоростей передачи на передающем и принимающем устройствах должны совпадать. Также совпадать должны форматы передающихся кадров.

Формат кадра задаётся в регистре UCSRC

| UCSRC.0 | UCSRC.1 | UCSRC.2 | Формат кадра               |
|---------|---------|---------|----------------------------|
| 0       | 0       | 0       | Слово данных 5 бит         |
| 1       | 0       | 0       | 6 бит                      |
| 0       | 1       | 0       | 7 бит                      |
| 1       | 1       | 0       | 8 бит (стоит по умолчанию) |
| 1       | 1       | 1       | 9 бит                      |

## 1.5 Переходник USB-UART CP2102

Переходник USB-UART на базе микросхемы CP2102 предназначенный для связи микроконтроллера с ПК через интерфейс UART. Микроконтроллер подключается к микросхеме CP2102 через UART интерфейс. Преобразователь распознается компьютером как устройство, подключенное через COM-порт и взаимодействие с переходником сводится как работа с простыми COM-портами. Может работать с длиной передаваемых данных от 5 до 8 бит с 1-2 "стоповыми" битами. Кроме того могут использоваться биты проверки на четность и нечетность. Диапазон скоростей передачи также значительно расширен: от 300 бит/секунду до 1 Мегабайта/секунду. На рисунке 15 изображено переходник USB-UART CP2102.



Рисунок 15 – Представлен USB-UART CP2102

Для соединения к микроконтроллеру нужны 5 проводов:

1. GND общий провод питания ("Земля");
2. V5.0 (V3.3) в зависимости от используемой платы;
3. RXD асинхронный последовательный вход данных (Receive);
4. TXD асинхронный последовательный выход данных (Transmit);
5. RESET сброс микросхемы. Выход с открытым стоком внутренней цепи сброса или монитора питания. Внешний источник может инициировать си-

стемный сброс переводом входного уровня в низкое состояние на время как минимум 15 мкс;

Основные характеристики:

1. Поддержка интерфейса USB 2.0 ;
2. Стандартный разъем USB-A ;
3. 6 пиновый TTL разъем и соединительные провода ;
4. Напряжение питания: 3.3В и 5В ;
5. Скорость передачи данных по UART: от 300бит/с до 1 Мб/с ;
6. Три светодиода (RX, TX, POWER) ;
7. Буфер обмена TX: 640 б;
8. Буфер обмена RX: 576 б;

Первый бит это направление, а остальные семь битов это модуль скорости.

#### 1.6 Программа Terminal 1.9b

Для связи с AVR и компьютера по UART на компьютере используется программа "Terminal 1.9b". Программа Terminal является монитором COM порта персонального компьютера. С помощью программы можно отправлять и принимать данные через COM порт компьютера. Достоинства программы это гибкая настройка программы под различные режимы работы (различная скорость работы, конфигурация стоп-битов, конфигурация количества бит-данных и многое другое).

#### Тестирование

Управление скоростями шагового двигателя проводилось с компьютера. С программы Terminal 1.9 мы отправляли значение скорости, которое через переходник USB-UART передавалось на шаговый двигатель.

Правильность передачи скорости на шаговый двигатель проверялось с помощью осциллографа. На осциллограмме двоичным кодом, показывается значение текущей скорости.

Из графика видно, что 1bit передается приблизительно в течение 100 мкс, что совпадает с теорией скорости передачи.

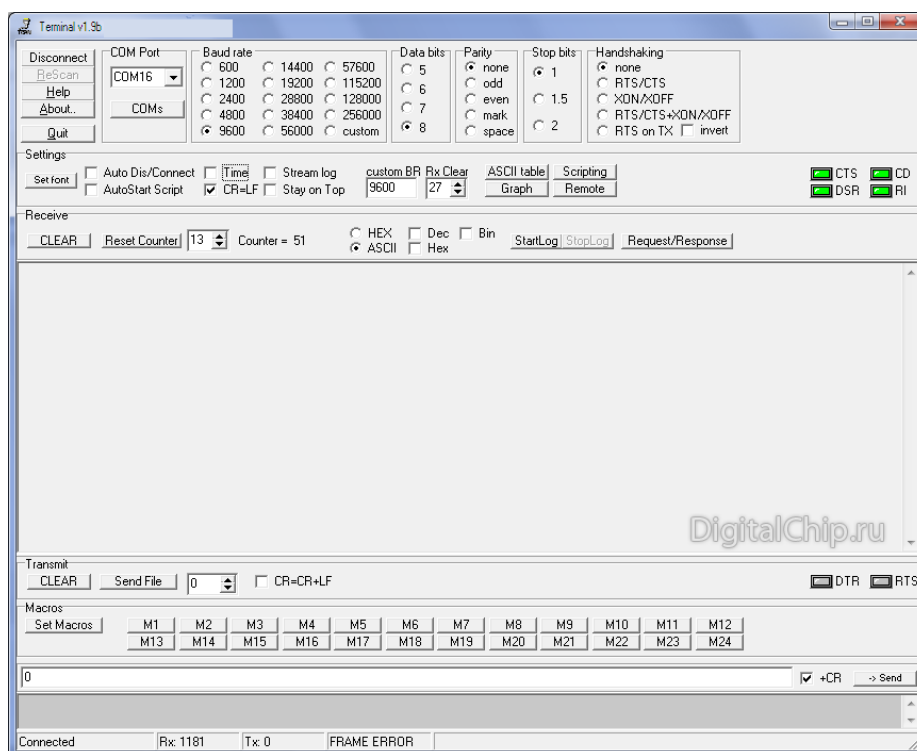


Рисунок 16 - общий вид программы Terminal 1.9b

Визуализация UART протокола.

Из таблицы ASCII – кодов мы можем записать следующие значения скоростей в двоичном коде:

“1” = 48 = 00110001;

“2” = 49 = 00110010;

“3” = 50 = 00110011;

“4” = 51 = 00110100.

На рисунках 1720 представлены состояние линии  $R_x$  со стороны микроконтроллера. Передача осуществляется справа налево: вначале младшие биты потом старшие.

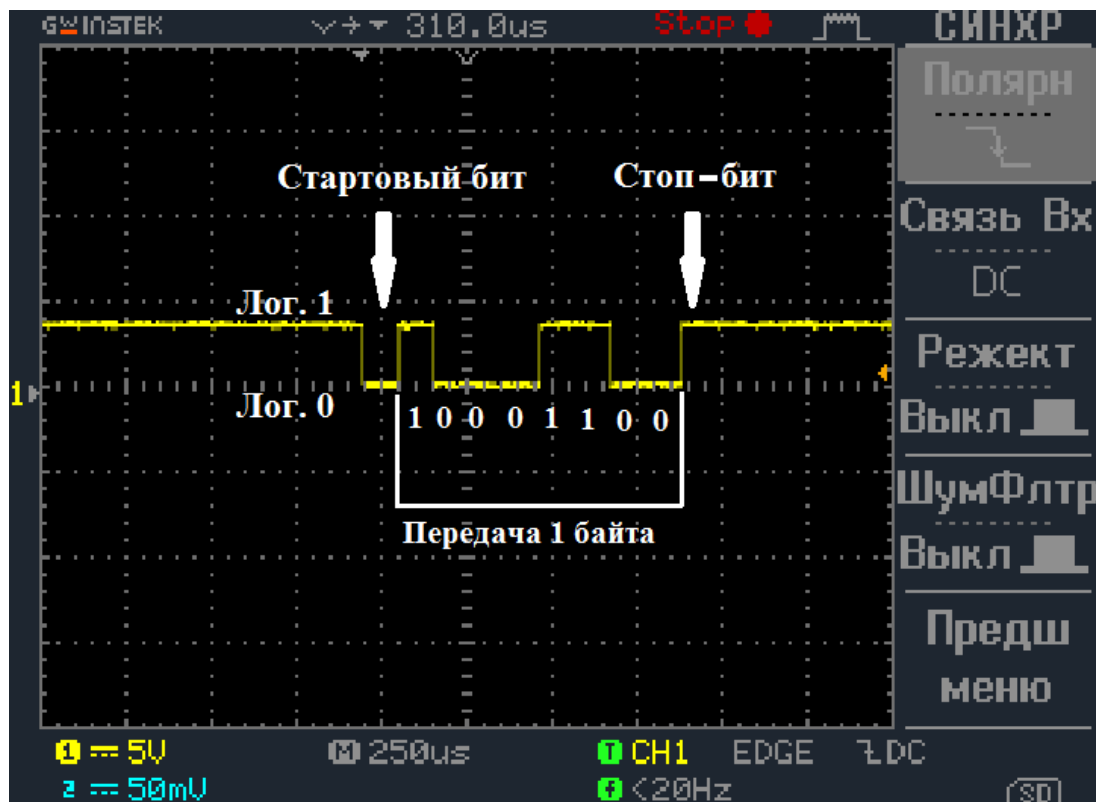


Рисунок 17 – показана диаграмма соответствующая передачи символа “1” с помощью таблицы ASCII-кода ( “1” = 48 = 00110001)

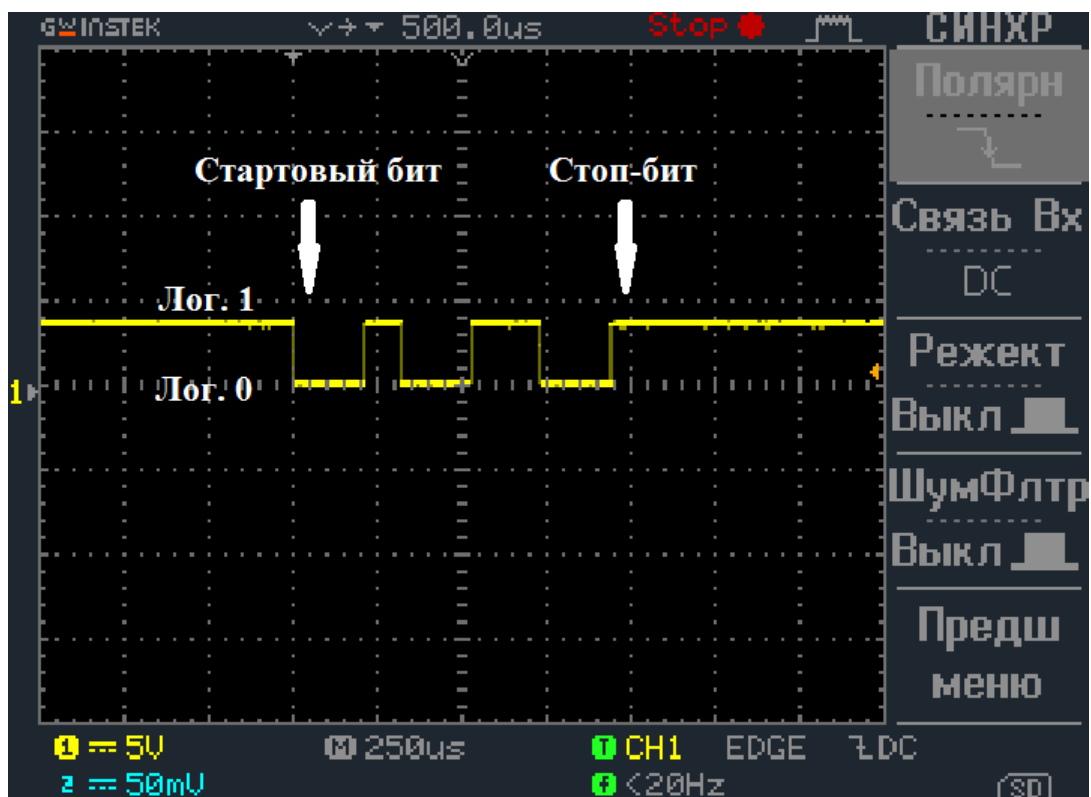


Рисунок 18 – показана диаграмма соответствующая передачи символа “2” с помощью таблицы ASCII-кода ( “2” = 49 = 00110010)

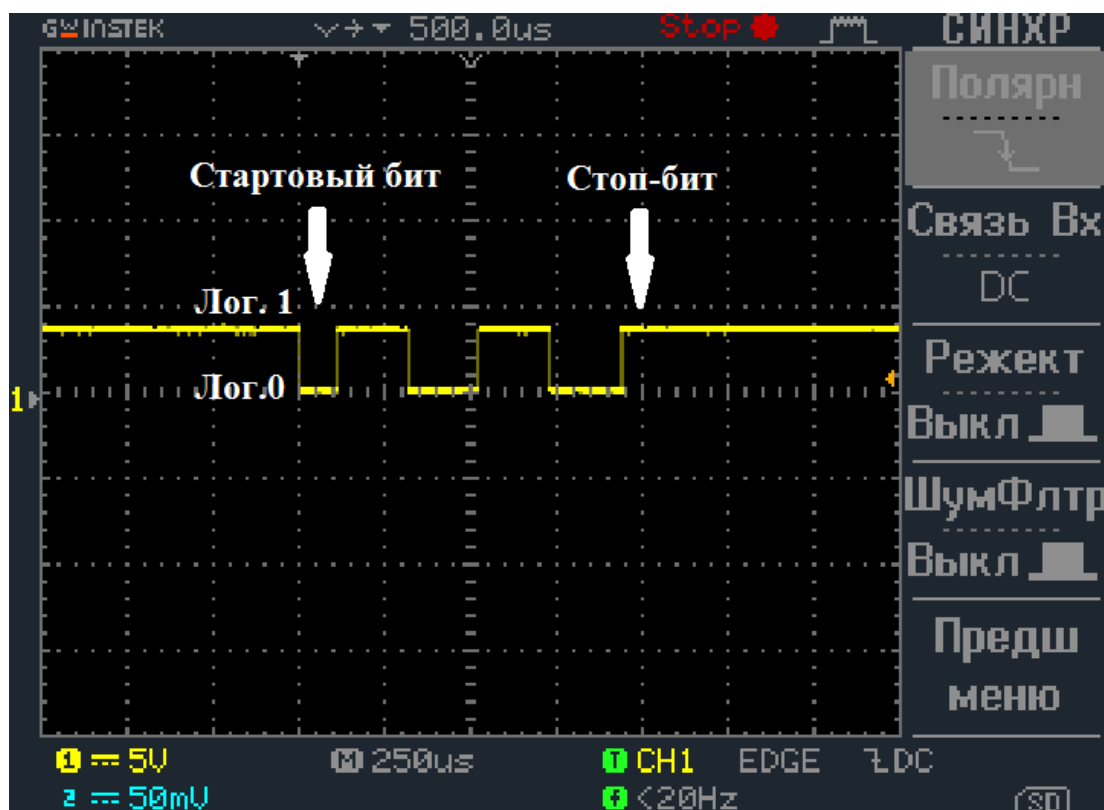


Рисунок 19 – показана диаграмма соответствующая передачи символа “3” с помощью таблицы ASCII-кода ( “3” = 50 = 00110011)

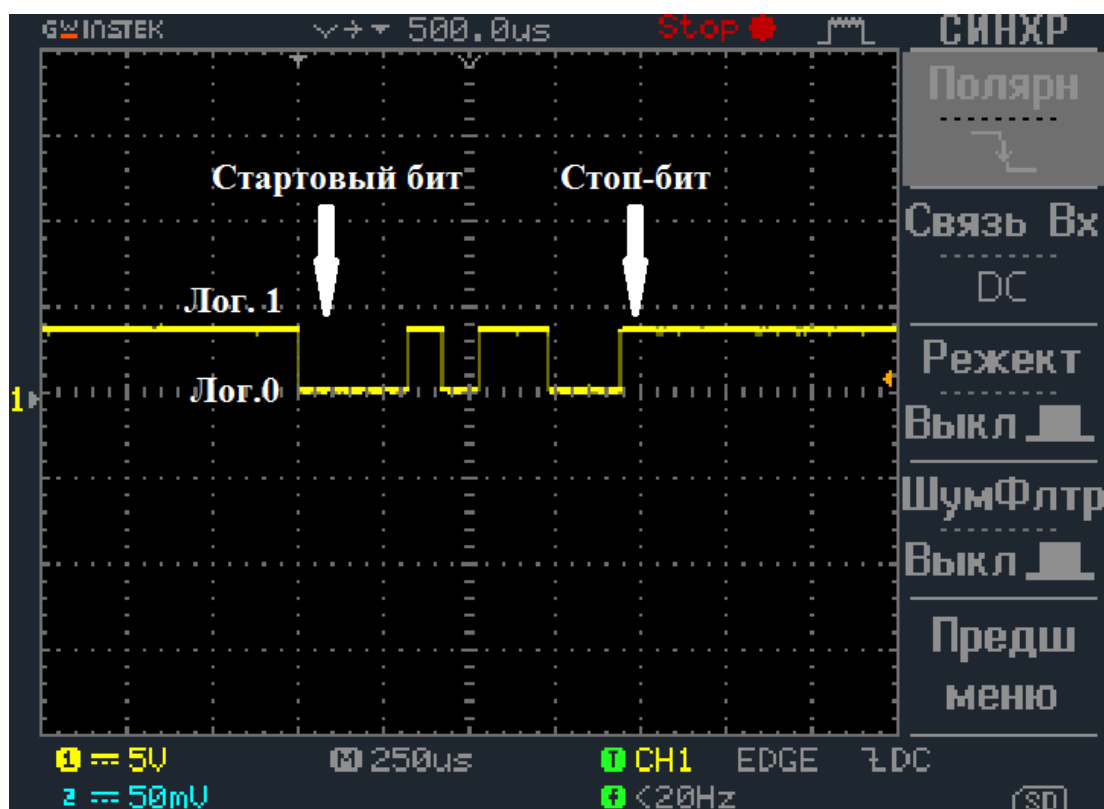


Рисунок 20 – показана диаграмма соответствующая передачи символа “4” с помощью таблицы ASCII-кода ( “4” = 51 = 00110100)

В итоге была собрана схема передачи данных между микроконтроллером Atmega8, шаговым двигателем, микросхемой L298n и выводом информации на LCD экран, при помощи персонального компьютера, с помощью преобразователя CP2102. Для микроконтроллера ATmega8 была написана программа,

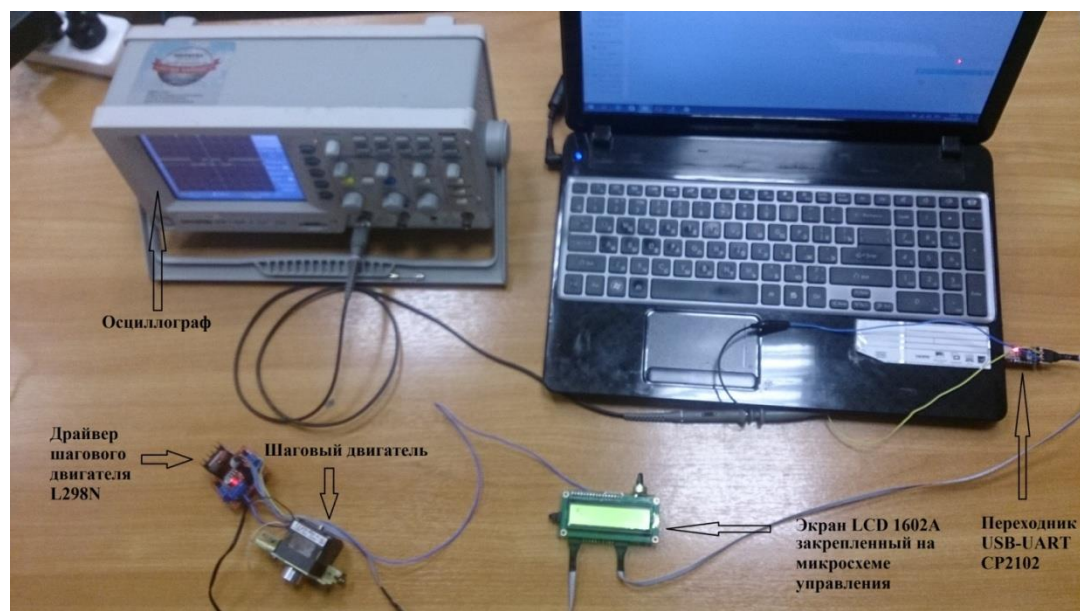


Рисунок 21 – схема работы

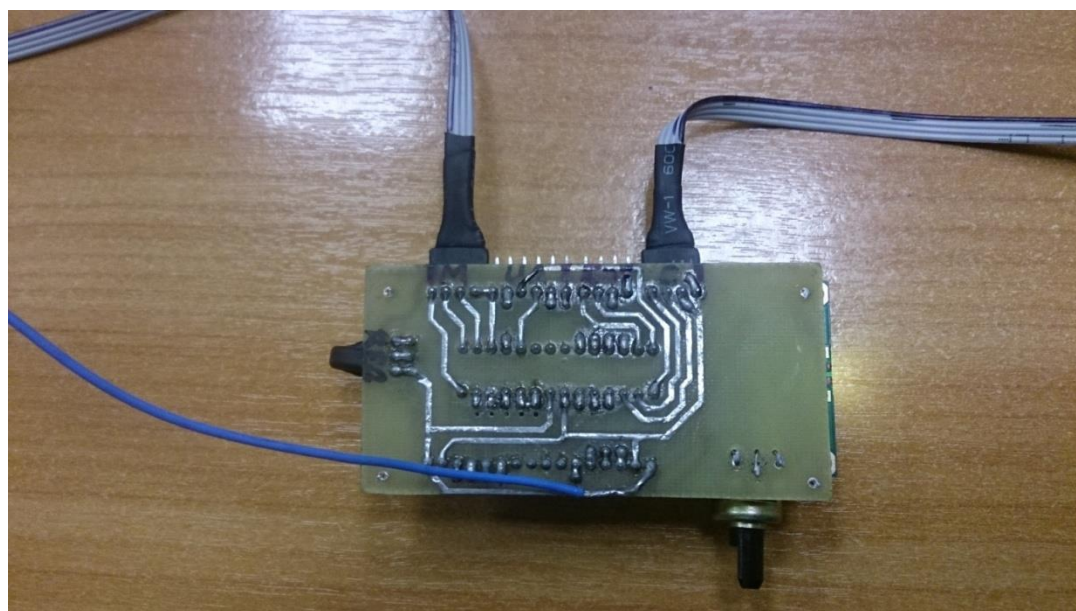


Рисунок 22 – микросхема управления

## ЗАКЛЮЧЕНИЕ

В ходе работы были изучены основы функционирования микроконтроллеров и средств передачи данных между цифровыми устройствами. Была создана схема передачи данных между микроконтроллером Atmega8, шаговым двигателем, микросхемой L298n и выводом информации на LCD экран, при помощи персонального компьютера, с помощью преобразователя CP2102. Была написана программа для микроконтроллера.

## Список использованных источников литературы

- 1.Использование интерфейса USART микроконтроллеров AVR //Radio Party. [Электронный ресурс]. URL: <http://radioparty.ru/prog-avr/program-c/307-lesson-usart-avr> (дата обращения: 20.03.16).
- 2.Реализация в ATmega8 //Radio Aktiv[Электронный ресурс].URL: <http://radioaktiv.ru/shems/digital/prog/27894-usbasp-usb-programmator-dlya-atmel-avr-mikrokontrollerov.html> (дата обращения 24.03.16).
- 3.AVR микроконтроллеры //Doneat Home. [Электронный ресурс]. URL: <http://www.doneathome.ru/archives/966> (дата обращения 30.03.16).
4. AVR. Учебный курс. Передача данных через UART //Easy Electronics. [Электронный ресурс]. URL:<http://easyelectronics.ru/avr-uchebnyj-kurs-peredacha-dannyx-cherez-uart.html>(дата обращения 07.04.16).
5. CP2102 - преобразователь USB-UART // Easy Electronics. [Электронный ресурс]. URL:<http://we.easyelectronics.ru/electro-and-pc/cp2102-preobrazovatel-usb-uart-za-45-rublej.html> (дата обращения 20.04.16).
6. Terminal 1.9b — работа с COM-портом // Digital Chip. [Электронный ресурс]. URL: <http://digitalchip.ru/terminal-1-9b-rabota-s-com-portom> (дата обращения 30.04.16).
- 7.Таблица символов ASCII // Industrial Nets. [Электронный курс]. URL: <http://www.industrialnets.ru/files/misc/ascii.pdf> (Дата обращения 24.05.16).
- 8.Настройка скорости UART // Micri Sin. [Электронный ресурс]. URL: <http://microsin.net/programming/avr/atmega2560-working-with-uart.html> (дата обращения 27.05.16).
- 9.Bill Earl. All about Stepper Motors [Text] // Adafruit learning system, 2015. – P. 28.
- 10.Stepper Motor Basics [Text] // Industrial Circuits Application Note – P. 1-6.

## Приложение

```
#define F_CPU 8000000UL
#include <avr/io.h>
#include <util/delay.h>
#include "lcd.h"
#include <avr/interrupt.h>
int n = 65458;
ISR(TIMER1_OVF_vect)                                //Прерывания по переполнению таймера 1
{
    TCNT1 = n;                                       //Старт счетчика с n
    if(PINB & (1<<1)) {PORTB=0x04;}                //Задаем третью скорость шагового дви-
гателя
    else if(PINB & (1<<3)) {PORTB=0x02;}           //Задаем вторую скорость шагового дви-
гателя
    else if(PINB & (1<<0)) {PORTB=0x08;}           //Задаем четвертую скорость шагового
двигателя
    else {PORTB=0x01;}                             //Задаем первую скорость шагового дви-
гателя
}
int init_UART(void)
//Установка скорости 9600
{
    UBRRH=0; // UBRR=f/(16*band)-1 f=8000000Гц band=9600,
    UBRL=51; // Нормальный асинхронный двунаправленный режим работы
    UCSRA=0b00000000;
    UCSRB=0b00011000; // Разрешен приём и передача по UART
    UCSRC=0b10000110; // 8-битовая посылка
}
int main(void)
{
    char st[2]={'s','\0'};
    char a;
    LCDInit(0); //Установка курсора LCD
    LCDClear(); //Очистка LCD
    LCDWriteString(st); //Вывод на LCD строки
    init_UART(); //Инициализация UART
    DDRB=0x0F;
    PORTB=0x00; //Устанавливаем нули на выходе
    TCCR1A = 0x00;
    TCCR1B = (1<<CS12)|(0<<CS11)|(1<<CS10);
    TIMSK |= (1<<TOIE1);
    TCNT1 = n;
    sei();
    while(1)                                         // Бесконечный рабочий цикл
    {
        if(UCSRA&(1<<RXC)) // Если пришёл байт по UART
        {a=UDR;
        st[0]=a;
        LCDClear(); //Очистка LCD
        _delay_ms(10); //Задержка 50мс
        LCDWriteString(st); //Вывод на LCD строки
        if(a == '1'){n=64754;}; //Установка первой скорости
        }
```

```
if(a == '2'){n=65458;}; //Установка второй скорости
if(a == '3'){n=65497;}; //Установка третьей скорости
if(a == '4'){n=65516;}; //Установка четвертой скорости
}
}
}
```

Поиск заимствований в научных текстах<sup>β</sup>
[\(/index.php/ru/\)](#) [\(/index.php/en/\)](#)

Введите текст:

...или загрузите файл:

Файл не выбран...

Выбрать файл...

Укажите год публикации:

2017 ▼

Выберите коллекции

Все

Рефераты

Авторефераты

Иностранные конференции

PubMed

Википедия

Российские конференции

Иностранные журналы

Российские журналы

Энциклопедии

Англоязычная википедия

Анализировать

 Проверить по расширенному списку коллекций системы Руконтекст (<http://text.rucont.ru/like>)

Обработан файл:

Магистерская Хотулёв Владимир.docx.

Год публикации: 2017.

Оценка оригинальности документа - 88.61%

Процент условно корректных заимствований - 0.0%

Процент некорректных заимствований - 11.39%

Просмотр заимствований в документе

Время выполнения: 21 с.

Документы из базы

Источники заимствования

**1. Универсальный асинхронный приёмопередатчик**  
 (<https://ru.wikipedia.org/wiki?curid=1139057>)

Год публикации: 2016. Тип публикации: статья википедии.

<https://ru.wikipedia.org/wiki?curid=1139057> (<https://ru.wikipedia.org/wiki?curid=1139057>)

Показать заимствования (14)

**2. Учебное пособие: Микроконтроллеры AVR**  
 (<http://www.bestreferat.ru/files/11/bestreferat-141011.docx>)

Год публикации: 2016. Тип публикации: реферат.

<http://www.bestreferat.ru/files/11/bestreferat-141011.docx>
<http://www.bestreferat.ru/files/11/bestreferat-141011.docx>

Показать заимствования (3)



| В списке литературы | Источники Заимствования |
|---------------------|-------------------------|
| 6.9%                | 1.49%                   |

**3. Учебное пособие: Электропривод с шаговым двигателем**  
**(<http://www.bestreferat.ru/files/12/bestreferat-170112.docx>)**

Год публикации: 2016. Тип публикации: реферат.  
<http://www.bestreferat.ru/files/12/bestreferat-170112.docx>  
<http://www.bestreferat.ru/files/12/bestreferat-170112.docx>  
 Показать заимствования (4)

1.42%

**4. Реферат: Разработка программно аппаратного устройства с числовым программным управлением**  
**(<http://www.bestreferat.ru/files/97/bestreferat-220897.docx>)**

Год публикации: 2016. Тип публикации: реферат.  
<http://www.bestreferat.ru/files/97/bestreferat-220897.docx>  
<http://www.bestreferat.ru/files/97/bestreferat-220897.docx>  
 Показать заимствования (3)

1.29%

**5. Учебное пособие: Основы автоматизации эксперимента Герман**  
**(<http://www.bestreferat.ru/files/32/bestreferat-408532.docx>)**

Год публикации: 2016. Тип публикации: реферат.  
<http://www.bestreferat.ru/files/32/bestreferat-408532.docx>  
<http://www.bestreferat.ru/files/32/bestreferat-408532.docx>  
 Показать заимствования (3)

1.03%

**6. Совершенствование лабораторного практикума обучения студентов по направлению подготовки «Электромеханика»**  
**(<http://cyberleninka.ru/article/n/sovershenstvovanie-laboratornogo-praktikuma-obucheniya-studentov-po-napravleniyu-podgotovki-elektromehanika>)**

Авторы: Бондаренко В. И., Орловский И. Л., Пирожок А. В.  
 Год публикации: 2012. Тип публикации: статья научного журнала.  
<http://cyberleninka.ru/article/n/sovershenstvovanie-laboratornogo-praktikuma-obucheniya-studentov-po-napravleniyu-podgotovki-elektromehanika>  
<http://cyberleninka.ru/article/n/sovershenstvovanie-laboratornogo-praktikuma-obucheniya-studentov-po-napravleniyu-podgotovki-elektromehanika>  
 Показать заимствования (3)

0.79%

**7. Реферат: Ответы к Экзамену по Микропроцессорным Системам**  
**(микроконтроллеры микропроцессоры)**  
**(<http://www.bestreferat.ru/files/84/bestreferat-53384.docx>)**

Год публикации: 2016. Тип публикации: реферат.  
<http://www.bestreferat.ru/files/84/bestreferat-53384.docx>  
<http://www.bestreferat.ru/files/84/bestreferat-53384.docx>  
 Показать заимствования (4)

0.78%

**8. AVR микроконтроллер AT90S2333 фирмы Atmel**  
**([http://limej.ru/index.php/home/192-stat/48354-AVR\\_mikrokontroller\\_AT90S2333\\_firmi\\_Atmet.html](http://limej.ru/index.php/home/192-stat/48354-AVR_mikrokontroller_AT90S2333_firmi_Atmet.html))**

Год публикации: 2016. Тип публикации: реферат.  
[http://limej.ru/index.php/home/192-stat/48354-AVR\\_mikrokontroller\\_AT90S2333\\_firmi\\_Atmet.html](http://limej.ru/index.php/home/192-stat/48354-AVR_mikrokontroller_AT90S2333_firmi_Atmet.html)  
[http://limej.ru/index.php/home/192-stat/48354-AVR\\_mikrokontroller\\_AT90S2333\\_firmi\\_Atmet.html](http://limej.ru/index.php/home/192-stat/48354-AVR_mikrokontroller_AT90S2333_firmi_Atmet.html)  
 Показать заимствования (3)

0.63%

**9. Реферат: AVR микроконтроллер AT90S2333 фирмы Atmel**  
**(<http://www.bestreferat.ru/files/15/bestreferat-297315.docx>)**

Год публикации: 2016. Тип публикации: реферат.  
<http://www.bestreferat.ru/files/15/bestreferat-297315.docx>  
<http://www.bestreferat.ru/files/15/bestreferat-297315.docx>  
 Показать заимствования (3)

0.63%

**10. Ответы к Экзамену по Микропроцессорным Системам**  
**(микроконтроллеры микропроцессоры)**  
**([http://limej.ru/index.php/home/136-stat/46956-Otveti\\_k\\_Ekzameni\\_po\\_Mikroprotsessornim\\_Sistemam\\_m.html](http://limej.ru/index.php/home/136-stat/46956-Otveti_k_Ekzameni_po_Mikroprotsessornim_Sistemam_m.html))**

Год публикации: 2016. Тип публикации: реферат.  
[http://limej.ru/index.php/home/136-stat/46956-Otveti\\_k\\_Ekzameni\\_po\\_Mikroprotsessornim\\_Sistemam\\_m.html](http://limej.ru/index.php/home/136-stat/46956-Otveti_k_Ekzameni_po_Mikroprotsessornim_Sistemam_m.html)  
[http://limej.ru/index.php/home/136-stat/46956-Otveti\\_k\\_Ekzameni\\_po\\_Mikroprotsessornim\\_Sistemam\\_m.html](http://limej.ru/index.php/home/136-stat/46956-Otveti_k_Ekzameni_po_Mikroprotsessornim_Sistemam_m.html)  
[http://limej.ru/index.php/home/136-stat/46956-Otveti\\_k\\_Ekzameni\\_po\\_Mikroprotsessornim\\_Sistemam\\_m.html](http://limej.ru/index.php/home/136-stat/46956-Otveti_k_Ekzameni_po_Mikroprotsessornim_Sistemam_m.html)  
 Показать заимствования (3)

0.58%

Значимые оригинальные фрагменты

Дополнительно

Библиографические ссылки

Искать в Интернете

Руководитель ООО *Сбер* *Сергеев Г.Р.*  
Руководитель ВКР *ЕБ* *Борзенко Е.И.*  
Автор работы *Жу* *Хотулев В.А.*