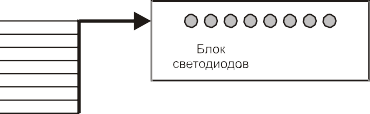
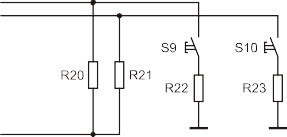
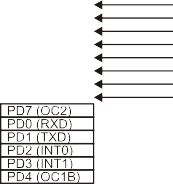
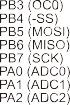
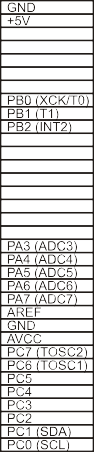
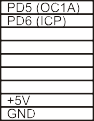
Лабораторная работа № 13



# Использование модуля UART

Цель работы:

1. Изучение принципов работы UART.
2. Программирование работы UART микроконтроллера ATmega8535.
3. Организация передачи данных между лабораторным макетом и пер- сональным компьютером (ПК).

Оборудование и программное обеспечение:

1. Лабораторный макет MK8535 в составе: модуль базовый МБ8535 +

модуль сменный МС01.

1. Персональный компьютер.
2. Кабель соединительный RS-232 «MK8535 – COM-порт ПК».
3. Кабель соединительный для программирования МК.
4. Интегрированная среда программирования CodeVisionAVR.

# Описание лабораторного макета

Функциональная схема лабораторного макета привед на на рис. 80.

е



Рис. 80. Функциональная схема лабораторного макета. Для модуля МС01 показана только та часть схемы, которая используется в данной работе

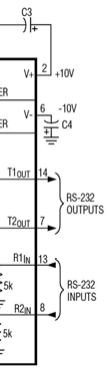
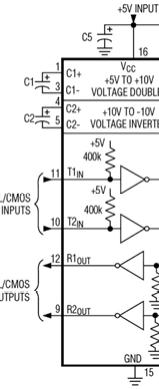


Рис. 79.

Схема разъема-заглушки

Напомним, что уровни сигналов на выводах RxD и TxD, как и на остальных выводах МК, могут принимать значения 0 В и +5 В. Для пре- образования их к уровням стандарта RS-232 в нашем макете использована популярная микросхема MAX232 (рис. 78). Она содержит по два канала преобразователей в обоих направлениях, встроенный источник двуполяр- ного питания ±10 В и нуждается только в стандартном питании +5 В.

Рис. 78. Микросхема MAX232

# Подготовка к работе

**Прием/передача с использовани м стандартных функций**

**е**

В библиотеке **CodeVisionAVR** имеются функции getchar() – ввод одного символа и putchar() – вывод одного символа. Эти функции пред- полагают ввод и вывод через UART.

Функция getcha () возвращает байт, принятый чер з UART (если байт еще не поступил, то функция дожидается его прихода и после этого

r

е

и зададим режимы работы UART (см. рис. 81): разрешить работу прием- ника и передатчика, режим – асинхронный, скорость передачи – 9600 бод, формат – 8 бит, 1 стоп-бит, без контроля четности. Прерывания здесь не используются.

Ниже приведен текст программы для МК. В нем опущены строки, не существенные для данного примера.

# #include <mega8535.h> #include <stdio.h>

завершается). Функция putchar() отправляет байт через U

A

В работе используется разъем-заглушка, распая

RT.

ный так, что

н

# #include <delay.h>

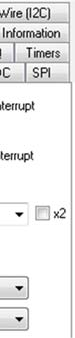
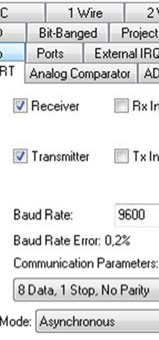
данные TD с выхода передатчика порта RS-232 направляются на вход

RD его же приемника (см. рис. 79).

Исследуем простой пример, где байт, набранный на блоке пере- ключателей S1…S8, будет отправлен по нажатию кнопки S9 на выход RS-232 и через разъем-заглушку верн тся на вход UART. Принятый байт отобразим на блоке светодиодов VD1…VD8. Прием байта будем подтверждать коротким гудком динамика SP1.

в

е



а) б)

Рис. 81. Выбор тактовой частоты (а) и установка режимов мод ля USART (б)

у

Вставим заглушку в разъем RS- 32 на модуле МС01. Создадим прое т в **CodeVisionA R,** все линии пор а A настроим на вывод, порт C –

к

**V**

2

т

# void main(void)

**{**

**unsigned char i;**

**PORTA=0x00; DDRA=0xFF; // Port A initialization PORTB=0x00; DDRB=0x08; // Port B initialization PORTC=0x00; DDRC=0x00; // Port C initialization PORTD=0x00; DDRD=0x02; // Port D initialization**

**// USART initialization**

**// Communication Parameters:**

**// 8 Data, 1 Stop, No Parity**

**// USART Receiver: On**

**// USART Transmitter: On**

**// USART Mode: Asynchronous**

**// USART Baud Rate: 9600 UCSRA=0x00;**

**UCSRB=0x18; UCSRC=0x86; UBRRH=0x00; UBRRL=0x33;**

**while (1)**

**{**

**while (PIND.2 != 0); // ждем нажатия кнопки S9 delay\_ms(20); // задержка 20 мсек**

**// отправить байт с переключателей**

**putchar(PINC);**

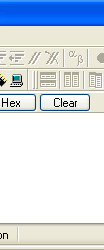
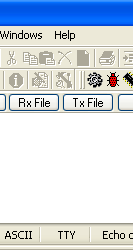
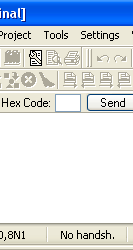
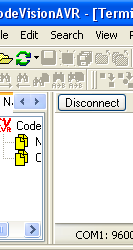
**while (PIND.2 == 0); // ждать отпускания S9 delay\_ms(20); // задержка 20 мсек**

**PORTA = getchar(); // отобразить принятый байт**

на ввод. Линию 3 порта B настроим на вывод. Линии 2 и 3 порта D –

на ввод, а линию 1 (TxD) – на вывод. Выберем тактовую частоту МК

# for (i=150; i>0; i--) // Гудок



ис. 83. Программа Терминал»

**{**

# PORTB.3 = i & 0x01;

**delay\_ms(1);**

**}**

**}**

**}**

# Передача данных между макетом MK8535 и персональным компьютером

В составе персонального компьютера (ПК), как правило, имеется последовательный коммуникационный порт (COM-порт), работающий в стандарте RS-232. Это позволяет устройству на основе микроконтрол- лера AVR обмениваться данными с ПК. Наиболее просто соединить лабо- раторный макет МК8535 с ПК можно с помощью кабеля, распаянного по т.н. «нуль-модемной» схеме (см. рис. 82).



С помощью соединительного кабеля соединим разъем RS-232 на модуле МС-01 и разъем COM-порта ПК. Запустим «Терм нал», выберем и настроим COM-порт. Очевидно, что для успешной передачи данных настройки портов на обеих сторонах должны быть идентичными. Для этого выберем те же настройки, что и на макете МК8535. В главном меню выберем **Settings**→**Te minal** и в окне **Terminal Settings** (рис. 84) уста- новим такие же параметры, как на рис. 81, б.

Р

**r**

«

и

Рис. 82. Схема «нуль-модемного» соединительного кабеля

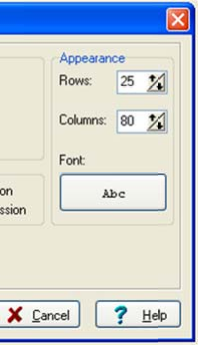


Рис. 84. Настройки COM-порта ПК в программе «Терминал»

Как видно на схеме, кабель симметричный – выход передатчика одной стороны соединен с входом приемника другой стороны и наоборот.

В CodeVisionAVR имеется программа «Терминал» (рис. 83) (вызов из главного меню **Tools**→**Terminal**). Программа позволяет отображать на экране символы и строки, поступающие из COM-порта ПК, а также отправлять символы и строки в COM-порт.

# Задание

1. Создать проект и реализовать рассмотренный выше пример программы.
2. Организовать передачу из «Терминала» в макет одиночного байта и отображение его на светодиодах модуля МС01. В свою очередь, байт, набранный на переключателях модуля МС01, передать нажатием кнопки и отобразить в «Терминале».
3. Библиотечная функция puts() в CodeVisionAVR выводит строку сим- волов через UART. Организовать передачу строки символов, напри- мер, приветствия, от макета в персональный компьютер.

# УПРАВЛЕНИЕ ШАГОВЫМ ЭЛЕКТРОДВИГАТЕЛЕМ

Шаговый электродвигатель (ШД) обычно состоит из многополюс- ного статора с соответствующим числом фазных обмоток и зубчатого ротора из магнитного материала. Для управления поворотом ротора в фазные обмотки подают импульсы тока в нужном порядке, при этом зубцы ротора стремятся притянуться к соответствующим полюсам статора и ротор поворачивается на фиксированный угол. Таким образом, направ- ление и угол поворота ротора ШД однозначно определяется серией фаз- ных импульсов. За один оборот ротор совершает целое число шагов.

Ротор ШД может удерживаться в фиксированном положении, пока через фазную обмотку протекает постоянный ток.

Применение ШД обеспечивает относительную простоту реализа- ции систем позиционирования, а импульсный характер управления удобно согласуется с цифровыми управляющими устройствами.

Основные характеристики ШД:

* Число фаз;
* Угол поворота град./шаг (или число шагов на 1 оборот);
* Крутящий момент (момент удерживания);
* Максимальная частота фазных импульсов.

К недостаткам ШД можно отнести невысокие скорость вращения и момент на валу ротора.

В лабораторной работе используется 4-фазный шаговый двига- тель. Исследуются два режима управления ШД: основной (полношаго- вый) и режим с половинным шагом.

Основной режим управления ШД поясняет диаграмма на рис. 85,

когда импульсы тока подают в фазные обмотки по очереди, в порядке

1 → 2 → 3 → 4 → 1 → 2 → 3 → 4 → 1 → и т.д.

и ротор последовательно принимает фиксированные угловые положе- ния. Видно, что управляющая последовательность циклическая и один цикл составляют 4 шага. Чтобы изменить направление вращения, доста- точно поменять эту последовательность на обратную. На диаграмме также приведены двоичные коды для управления фазными токами.

В этой управляющей последовательности цикл состоит из 8 ша- гов. В те моменты, когда токи протекают одновременно в обеих сосед- них обмотках, зубцы ротора «повисают» между полюсами статора, и образуются дополнительные угловые положения. Угловой шаг в таком режиме уменьшается вдвое, а число шагов на один оборот, соответ- ственно, удваивается. Для смены направления вращения нужно так же изменить фазную последовательность на обратную.



Рис. 85. Временная диаграмма фазных токов в режиме полного шага

Режим половинного шага на том же ШД можно получить, если применить другую последовательность фазных токов:

1 → 1,2 → 2 → 2,3 → 3 → 3,4 → 4 → 4,1 → 1 → 1,2 → 2 → 2,3 → и т.д. Такой режим еще называют режимом с перехватом фаз. Времен-

ная диаграмма показана на рис. 86.



Рис. 86. Временная диаграмма фазных токов в режиме половинного шага