

аналогов в оценке бизнеса. Использование формальной модели бизнес-процессов позволило создать информационную систему, обеспечивающую поиск предприятий-аналогов и расчет мультипликаторов и позволяющую организовать эффективную работу пользователя на завершающем этапе процесса оценки бизнеса.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Калянов Г.Н.* Моделирование, анализ, реорганизация и автоматизация бизнес-процессов/ Г.Н.Калянов, – М: Финансы и статистика, 2007. –240 с.

Корохова Елена Вячеславовна

Федеральное государственное образовательное учреждение высшего профессионального образования «Южный федеральный университет»

E-mail: alen_ko@mail.ru

344090, Ростов-на-Дону, ул. Мельчакова, 10

Тел.: +7(8632)696991

Малышева Ольга Игоревна

Федеральное государственное образовательное учреждение высшего профессионального образования «Южный федеральный университет»

E-mail: olga-vrem@yandex.ru

Korokhova Elena Vyacheslavovna

Federal State-Owned Educational Establishment of Higher Vocational Education «Southern Federal University»

E-mail: alen_ko@mail.ru

10, Melchikova street, Rostov-on-Don, 344090, Russia

Phone: +7(8632)696991

Malysheva Olga Igorevna Federal State-Owned Educational Establishment of Higher Vocational Education «Southern Federal University»

E-mail: olga-vrem@yandex.ru

УДК 681.51(075.8)

М.И. Ледовской

ПРОСТОЙ СПОСОБ ОРГАНИЗАЦИИ ВЗАИМОДЕЙСТВИЯ AVR-КОНТРОЛЛЕРА И ПК

Рассматривается взаимодействие AVR-контроллера и ПК с использованием стенда STK500 и программы Hyper Terminal, а также программирование файлового ввода-вывода через UART микроконтроллера на языке C.

AVR-контроллер; STK500; RS-232; Hyper Terminal; язык C; UART; файловый ввод-вывод.

M.I. Ledovskoj

SIMPLE WAY OF THE ORGANIZATION OF INTERACTION OF THE AVR- CONTROLLER AND THE PERSONAL COMPUTER

Interaction of the AVR-controller and the personal computer with use of stand STK500 and program Hyper Terminal, and also programming of file input-output through UART the microcontroller in language C is examined.

The AVR-controller; STK500; RS-232; Hyper Terminal; language C; UART; file input-output.

В настоящее время микроконтроллерную технику осваивают широкие круги пользователей с различным уровнем начальной подготовки. Как правило, эти пользователи владеют персональным компьютером (ПК) и привыкли общаться с ним посредством клавиатуры и монитора. У такой категории пользователей может возникнуть вполне оправданное желание сделать так, чтобы клавиатура и монитор компьютера стали средством общения с микроконтроллером. Ниже рассматривается простой способ решения данной задачи на примере микроконтроллеров AVR.

Как известно, взаимодействие микроконтроллера и ПК имеет два аспекта: аппаратный и программный [1]. Среди известных аппаратных решений наиболее простым является использование фирменного стенда отладки микроконтроллеров, например STK500, в котором размещается целевой микроконтроллер. В этом случае микроконтроллер взаимодействует с ПК с помощью встроенного приемопередатчика UART, который через интерфейс RS-232 стенда STK500 подключается к разъему COM-порта компьютера (рис.1).

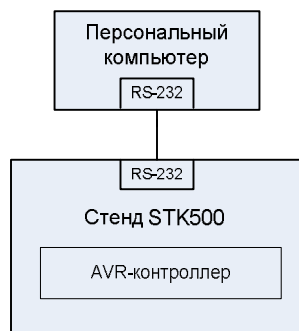


Рис.1. Схема подключения стенда STK500

Программный аспект взаимодействия микроконтроллера и ПК является более сложным. Зачастую на стороне ПК развертывают полноценное Windows-приложение, которое обладает развитым графическим интерфейсом и реализует обмен данными с микроконтроллером через COM-порт. Пример такого приложения содержится в [2]. Однако создание Windows-приложения требует знания объектно-ориентированного программирования и владения соответствующей средой программирования типа Delphi или Visual C++. В тех случаях, когда можно ограничиться простым текстовым интерфейсом, для взаимодействия с микроконтроллером используется стандартная терминальная программа Hyper Terminal, входящая в состав операционной системы Windows XP [1]. Данная программа обеспечивает ввод данных с клавиатуры, обмен данными через COM-порт компьютера и отображение этих данных на мониторе в текстовом формате. Применение программы Hyper Terminal полностью освобождает пользователя от программирования ПК и является самым простым решением.

Что же касается микроконтроллера, то программирование обмена данными с ПК является неизбежностью. Фактически речь идет о программировании ввода/вывода через приемопередатчик UART микроконтроллера, что по своей сути является программированием сетевого обмена данными с ПК через интерфейс RS-232.

Как известно, «быстрый старт» в программировании микроконтроллеров обеспечивает язык С [3]. К тому же многие пользователи имеют некоторый опыт программирования ПК на языке С. Для этих пользователей важно использовать такую технологию программирования ввода/вывода через UART, которая мало чем отличается от программирования ввода/вывода для стандартных периферийных устройств ПК (клавиатуры и монитора). Такой подход обеспечивает быструю и простую программную реализацию ввода/вывода на стороне микроконтроллера.

Возможность указанного подхода обеспечивает концепция файлового ввода/вывода, которую поддерживает язык С [4]. Она представлена функциями файлового ввода/вывода, которые предназначены для работы с неким абстрактным периферийным устройством, называемым потоком. Конкретное периферийное устройство типа UART в рамках данной концепции называется файлом. Концепция «ограждает» пользователя от программирования конкретного периферийного устройства, сводя его работу к программированию абстрактного периферийного устройства с помощью привычных операторов ввода/вывода, применяемых для клавиатуры и монитора компьютера. Пользователю потребуется лишь открыть поток и связать его с конкретным файлом (приемопередатчиком UART). Таким образом, благодаря концепции файлового ввода/вывода, программирование UART практически не отличается от программирования стандартного ввода/вывода для ПК.

Рассмотрим пример программы микроконтроллера AVR ATmega16, в которой применяется концепция файлового ввода/вывода компилятора WinAVR для программирования UART. Программа воспроизводит синусоидальную функцию относительно реального времени. Значения синуса, получаемые в микроконтроллере каждую секунду, выводятся через UART на монитор компьютера. Отсчет времени реализуется в микроконтроллере с помощью таймера/счетчика T/C1, который настроен на выдачу прерывания через одну секунду. Вычисление значений синуса и вывод их через UART происходит в обработчике прерывания от таймера/счетчика. В нем же обеспечивается включение/выключение светодиода для индикации работы таймера/счетчика.

Для работы программы в составе отладочного стенда STK500 порт В микроконтроллера подключается к линейке светодиодов, а выводы PD0 и PD1 порта D – к контактам RxD и TxD интерфейса RS-232 (RxD – линия приема, TxD – линия передачи). Микроконтроллер тактируется от внутреннего RC-генератора с частотой 1МГц. Значение калибровочного байта для частоты 1МГц составляет 0xA5. Ниже приводится листинг программы микроконтроллера.

```
// ioHT.c
#include <avr/io.h>
#include <stdio.h>
#include <stdlib.h>
#include <avr/interrupt.h>
#include <avr/signal.h>
#include <inttypes.h>
#include <pgmspace.h>
#include <math.h>
#define SYSCLK 1000000 // 1МГц
#define TIMER_TCNT1 (0x10000-(SYSCLK/256)) // 1с
#define TxD PD1
volatile float t;
int uart_putchar(char c)
```

```

{
    if (c == '\n') uart_putchar('\r');
    loop_until_bit_is_set(UCSRA, UDRE);
    UDR = c;
    return 0;
}
int uart_getchar(void)
{
    char c;
    loop_until_bit_is_set(UCSRA, RXC);
    c = UDR;
    return c;
}
// Обработчик прерывания таймера/счетчика T/C1
SIGNAL(SIG_OVERFLOW1)
{
    double y, z;
    int i;
    char str[]="";
    TCNT1=TIMER_TCNT1; // Установка таймера/счетчика T/C1 - 1с
    PORTB^=0x01; // Включение/выключение светодиода
    if (t>6.2) t=0;
    y=sin(t);
    z=20+20*y;
    i=(int)ceil(z);
    str[i]='*';
    str[20]='|';
    fprintf(stdout, "t=%3.11f sin(t)=%8.51f %s\n\r",
            t, y, str);
    str[i]=' ';
    t+=0.1;
}
void __init3(void) __attribute__((section(".init3")))
    __attribute__((naked));
void __init3(void)
{
    OSCCAL=0xA5; // Тактовая частота CPU - 1МГц
    DDRB=0x01; // Вывод PB0 - выход
    // Инициализация таймера/счетчика T/C1
    TCCR1A=0;
    TCCR1B=4; // Коэффициент деления частоты CPU -
1/256
    TCNT1=TIMER_TCNT1; // Установка таймера/счетчика T/C1 - 1с
    TIFR=0;
    TIMSK=1<<TOIE1;
    GIMSK=0;
    // Инициализация UART
    DDRD=1<<TxD;
    UBRR1L=25; // Скорость UART - 2400 бод
    UBRR1H=0;
    UCSRC=(1<<URSEL) | (0x3<<UCSZ0);
}

```

```

        UCSRB = (1<<TXEN) | (1<<RXEN);

    }
    int main (void)
    {
        double a, t=0;
        char c=' ';
        fdevopen(uart_putchar, uart_getchar, 0);
        fputs("\n\rВвод вещественного числа в AVR ATmega16:
                \n\r", stdout);

        fscanf(stdin, "%lf", &a);
        fprintf(stdout, "\n\rЭхо-ответ из AVR ATmega16:
                \n\r%10.4lf\n\r", a);
        fputs("\n\rВывод значений синусоиды из AVR ATmega16.
                Продолжить (Y)?\n\r", stdout);
        while (c!='Y') fscanf(stdin, "%c", &c);
        t=0;
        sei(); // Разрешение всех прерываний
        while(1); // Бесконечный цикл ожидания прерываний T/C1
        return 0;
    }

```

В секции инициализации программы выполняется инициализация регистров таймера/счетчика T/C1 и UART, а также настройка порта В на вывод. В начале главной функции `main()` функция `fdevopen()` открывает поток на ввод и вывод, а также связывает его с приемопередатчиком UART посредством функций `uart_putchar()` и `uart_getchar()`. Затем происходит тестирование канала обмена путем ввода произвольного вещественного числа с клавиатуры компьютера в микроконтроллер и ответного вывода на монитор (эхо-ответ). Для этого используются функции файлового ввода/вывода `fputs()`, `fscanf()` и `fprintf()`. После этого выполняется запрос на вывод значений синусоиды из микроконтроллера. В случае подтверждения запроса разрешаются все прерывания, и программа входит в бесконечный цикл ожидания прерываний от таймера/счетчика. При возникновении прерывания автоматически запускается обработчик, где выполняются указанные выше действия. По выходу из обработчика выполнение бесконечного цикла продолжается. Так происходит каждую секунду реального времени.

Для взаимодействия с микроконтроллером через интерфейс RS-232 в компьютере используется терминальная программа Nureg Terminal. Данная программа находится в разделе стандартных программ операционной системы Windows XP и запускается по команде Пуск→Все программы→Стандартные→Communication (Связь)→Nureg Terminal. После первого запуска программы необходимо ввести название подключения и выбрать для него значок. Затем следует настроить программу, выполнив следующие действия:

1. В меню программы выбрать Файл→Свойства.

2. На вкладке **Подключение** задать порт COM1. Затем выбрать опцию **Настроить ...** и в окне **Параметры порта** установить:

- ✓ скорость (бит/с) – 2400;
- ✓ биты данных – 8;
- ✓ четность – Нет;
- ✓ стоповые биты – 1;
- ✓ управление потоком – Аппаратное.

Нажать **Применить** и **ОК**.

3. На вкладке **Параметры** выбрать **Параметры ASCII ...**. В появившемся окне оставить включенными только следующие опции:
 - ✓ отображать введенные символы на экране;
 - ✓ переносить строки, превышающие ширину терминала.
4. В меню выбрать Вид→Шрифт. В появившемся окне задать шрифт **Lucida Console**, начертание **Обычный**, размер 12, набор символов **Кириллический**.
5. В меню выбрать Вид→Согласовать размеры.

Приведенные параметры настройки позволяют использовать программу Hyper Terminal для обмена данными с микроконтроллером ATmega16, в котором размещена приведенная выше программа. После запуска программы Hyper Terminal можно включить стенд STK500 и запустить тем самым программу микроконтроллера. При работе пользователя с программой микроконтроллера создается впечатление, что клавиатура и монитор компьютера являются устройствами ввода/вывода микроконтроллера (рис.2).

В заключение следует отметить, что недостатком файлового ввода/вывода является значительный объем исполняемого кода. Однако данный недостаток можно нивелировать, используя микроконтроллер с высоким быстродействием и достаточным объемом FLASH-памяти.

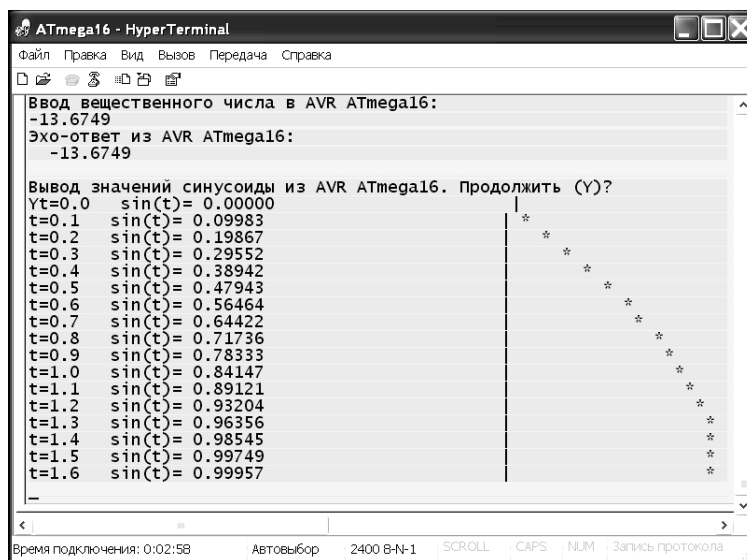


Рис.2. Окно программы Hyper Terminal после запуска программы микроконтроллера

Изложенный способ взаимодействия микроконтроллера и ПК целесообразно использовать для приобретения первоначального опыта работы с микроконтроллерами. Автор опробовал и успешно применил данный способ в учебном процессе в рамках дисциплины «Микропроцессорные контроллеры».

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Трамперт В.* Измерение, управление и регулирование с помощью AVR – микроконтроллеров. – К.: «МК-Пресс», 2006. – 208 с.

2. *Ледовской М.И.* Учебная информационная система для управления тепловыми процессами. Известия ЮФУ. Технические науки. Тематический выпуск: «Компьютерные и информационные технологии в науке, инженерии и управлении». – Таганрог: Изд-во ТТИ ЮФУ, 2008. №11(88). –С.138–143.
3. *Шнак Ю.А.* Программирование на языке С для AVR и PIC микроконтроллеров. – Киев, 2006. – 400 с.
4. *Шилдт Г.* Полный справочник по С. – М.: Изд-во Вильямс, 2004. – 704 с.

Ледовской Михаил Иванович,
Технологический институт федерального государственного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г.Таганроге
E-mail: kafmps@ttpark.ru
347900, г. Таганрог, ул. Петровская, 81
Тел. +7(8634)32-80-25

Ledovskoj Mikhail Ivanovich
Taganrog Institute of Technological – Federal State-Owned Educational Establishment of Higher Vocational Education «Southern Federal University»
E-mail: kafmps@ttpark.ru
81, Petrovskay street, Taganrog, 347900, Russia
Phone: +7(8634) 328025

УДК 621.391

Ю.М. Туляков

ОЦЕНКА СКОРОСТИ ПЕРЕДАЧИ ДАННЫХ ПО РАДИОИНТЕРФЕЙСАМ В ДЕЙСТВУЮЩИХ СИСТЕМАХ СОТОВОЙ СВЯЗИ

Приводится оценка способов и скорости передачи данных в системах сотовой связи, основанных на TDMA технологии, на примере GSM систем. Результаты оценки показывают потенциальные возможности передачи данных в таких системах и полезны при организации межсистемного взаимодействия систем сотовой связи разного типа и поколений.

Временное разделение каналов; глобальная система; мобильной связи; служба коротких сообщений; общая радиослужба; пакетная передача.

J.M. Tulyakov

ESTIMATION OF DATA LINK SPEED ON THE RADIO INTERFACE IN OPERATING SYSTEMS OF CELLULAR COMMUNICATION

The estimation of methods and link speed of the data in the systems of cellular communication based (founded) on TDMA of technology, on example GSM of systems are shown. Results of an estimation show potential possibilities of data transmission in such systems and are useful at the organisation of intersystem interaction of systems of cellular communication of different type and generations.

Time Division Multiple Access; Global System; Mobile Communications; Short Message Service; General Protocol; Radio Service.