

AVR-микроконтроллеры: программные средства

Не будет преувеличением сказать, что огромное число разработчиков микропроцессорных систем в мире, а особенно в Советском Союзе, выросло на 8-разрядных микропроцессорах 8080 и микроконтроллерах MSC 48/51 фирмы INTEL. Несмотря на то что этим архитектурам уже более 20 лет и сама фирма INTEL давно забыла об этих микросхемах, аналоги MSC 51 фирмы INTEL еще продолжают выпускаться многими производителями. Между тем, прогресс не стоит на месте, и новые микропроцессорные архитектуры на основе RISC-ядра постепенно вытесняют классические CISC-системы. Это относится как к мощным универсальным микропроцессорам (яркий пример — совместная разработка IBM и Motorola — RISC-процессор Power PC), так и к 8-разрядным микроконтроллерам. Тем не менее часто в новых разработках закладывается 8051-совместимая микросхема. Причин этому несколько — знакомая, испытанная система команд, богатый архив наработанного программного обеспечения, легкодоступные ассемблеры и компиляторы. Однако вычислительная мощность этих микроконтроллеров для некоторых приложений недостаточна, и тогда приходится искать новые решения. Одно из таких решений — использование AVR-микроконтроллеров фирмы ATMEL. Они выпускаются в корпусах с цоколевкой, идентичной Intel 8051, что позволяет ставить эти микросхемы в уже изготовленные печатные платы.

Николай Королев

korolev@argussoft.ru

Дмитрий Королев

components@argussoft.ru

В процессе выбора элементной базы для создания нового устройства разработчик рассматривает не только технические характеристики того или иного микроконтроллера, но также большое внимание уделяет средствам поддержки — как аппаратным (стартовые наборы, программаторы, внутрисхемные эмуляторы), так и программным (языки низкого и высокого уровня, симуляторы). Естественно, в расчет берется не только удобство работы и функциональные возможности конкретного пакета программ, но и его стоимость. Данная статья посвящена рассмотрению программных средств для разработки устройств с использованием микроконтроллеров семейства AT90S, выпускаемых фирмой ATMEL.

В настоящее время AVR-микроконтроллеры фирмы ATMEL завоевали широкую популярность на российском рынке. И это неудивительно — по количеству моделей в семействе они занимают первое место в мире среди ФЛЭШ-микроконтроллеров и по совокупности своих характеристик опережают большинство аналогичных изделий, занимая одно из первых мест в мире по соотношению цена/производительность. Высокая производительность достигнута не в последнюю очередь благодаря мощному и удобному набору команд, существенно повышающему эффективность кода по сравнению с микроконтроллерами классической архитектуры.

Как станет ясно из нижеизложенного материала, основным инструментом программиста является интегрированная среда разработки (IDE — INTEGRATED DEVELOPMENT ENVIRONMENT) — AVR STUDIO® 3.0. Эта оболочка включает в себя текстовый редактор, менеджер проектов, отладчик и предоставляет следующие возможности:

- создание и редактирование исходного кода на ассемблере;
- символьная отладка в исходных кодах;
- просмотр содержимого ФЛЭШ-ПЗУ, ЭСПЗУ, ОЗУ, регистров и портов ввода/вывода;
- неограниченное число точек прерывания;
- буфер трассировки;
- просмотр и модификация переменных с поддержкой механизма Drag-and-Drop;
- модификация состояния активности выводов портов ввода/вывода;
- загрузка файлов в форматах COFF, UBROF6 и HEX;
- поддержка проектов на языках C, Паскаль и Бейсик;
- совместная работа со всеми внутрисхемными эмуляторами фирмы Atmel.

Менеджер проекта (project manager) объединяет в проект группу файлов и обеспечивает интерфейс для подключения внешнего ассемблера/компилятора. Таким образом, есть возможность писать программу на выбранном языке и компилировать ее выбранным компилятором. Мощный текстовый редактор, входящий в AVR Studio, обеспечивает «бесшовную» стыковку с внешним компилятором и линкером. Исходный код можно редактировать непосредственно в отладочном окне. Поддерживается цветное выделение соответствующих фрагментов исходного кода.

Пользовательский интерфейс специально разработан для облегчения работы пользователя. Инструментальные панели (toolbars) и клавиши быстрого доступа обеспечивают удобный доступ ко всем ресурсам AVR-микроконтроллера. Установка точек прерывания и переключение на окно исходного текста производится одним нажатием кнопки мыши.

Одной из важных характеристик AVR Studio является встроенная поддержка работы с внутрисхемным эмулятором. При запуске программы производится опрос COM-портов компьютера на предмет наличия подключенного эмулятора. Если на каком-либо COM-порте обнаруживается эмулятор (в общем случае допускается совместная работа нескольких эмуляторов), AVR Studio стартует в режиме аппаратной отладки (emulator mode), о чем сообщает надпись «emulator» в нижней части основного окна, иначе активизируется режим симулятора. Пользовательский интерфейс в обоих случаях идентичный.

Эмуляторы AVR ICE 200, ICE PRO и AVR ICE 30 имеют возможность обновления конфигурации. Соответствующее программное обеспечение входит в состав AVR Studio. При инициализации эмулятора происходит проверка текущей версии эмулятора и при необходимости выводится окно с предложением «обновить» версию.

Архитектура AVR-микроконтроллеров спроектирована под компиляторы с языков высокого уровня. В частности, большое количество регистров общего назначения удобно для хранения «регистровых» переменных при написании программы на C. Этому способствует также высокое быстродействие микроконтроллеров (время выполнения команды составляет 100–150 наносекунд) и практически неограниченный объем памяти программ



Рис.1. Окно AVR Studio

(микросхем Atmega 103 имеет ФЛЭШ-ПЗУ программ объемом 128 Кбайт). Фирмы, выпускающие C-компиляторы для микроконтроллеров, не заставили себя ждать и вскоре предложили соответствующие пакеты. Из всего разнообразия наиболее интересны два — первый от шведской фирмы IAR Systems и второй от американской фирмы ImageCraft Creations Inc.

Фирма IAR Systems известна своей плодотворностью, ее продукты поддерживают около двадцати типов микроконтроллеров разных фирм-производителей. В комплект поставки входит среда разработки IAR Embedded Workbench и отладчик IAR C-Spy. Девиз фирмы «Различные архитектуры. Одно решение».

Программа 1

```
#include <io2313.h>
#pragma interrupt_handler timer:5
#define LCD_E (1 << 6)
#define LCD_RS (1 << 5)
#define LCD_RW (1 << 4)
#define lcd_set_e() (PORTD |= LCD_E)
#define lcd_set_rs() (PORTD |= LCD_RS)
#define lcd_set_rw() (PORTD |= LCD_RW)
#define lcd_clear_e() (PORTD &= ~LCD_E)
#define lcd_clear_rs() (PORTD &= ~LCD_RS)
#define lcd_clear_rw() (PORTD &= ~LCD_RW)

// короткая задержка для ЖКИ
void delay(int ticks)
{
    while(ticks--);
}

// строб ЖКИ
void lcd_pulse(void)
{
    lcd_set_e();
    delay(4);
    lcd_clear_e();
    delay(4);
}

// средняя задержка для ЖКИ
void lcd_wait(void)
{
    delay(1000);
}

// отправка байта на ЖКИ
void lcd_send(unsigned char data)
{
    lcd_wait();
    PORTB = data;
    lcd_pulse();
}

// очистка экрана ЖКИ
void clrscr(void)
{
    lcd_clear_rs();
    lcd_clear_rw();
    lcd_send(0x01);
    lcd_wait();
}

// инициализация ЖКИ
void initgraph(void)
{
    DDRB = 0xFF;
    DDRD |= (LCD_E | LCD_RS | LCD_RW);
    lcd_clear_rs();
    lcd_clear_rw();
    lcd_send(0x3C);
    lcd_send(0x3C);
    lcd_send(0x3C);
    lcd_send(0x06);
    lcd_send(0x0C);
}

// перемещение курсора ЖКИ
void gotoz(unsigned char z)
{
    lcd_clear_rs();
    lcd_clear_rw();
    lcd_send(z | 0x80);
}

#define gotoxy(x,y) gotoz((x)|(y)<<6)

// отправка символа на ЖКИ
void putchar(char c)
{
    lcd_clear_rw();
    lcd_set_rs();
    lcd_send(c);
}

// отправка строки
void outtext(char* text)
{
    unsigned char i;
    for(i = 0; text[i] && i < 16; i++)
        putchar(text[i]);
    unsigned char hour = 0,
        minute = 0, second = 0;
}

// вызов 1 раз в секунду
void timer(void)
{
    // текущее время
    clrscr();
    gotoxy(0,0);
    putchar('0'+hour/10);
    putchar('0'+hour%10);
    putchar(':');
    putchar('0'+minute/10);
    putchar('0'+minute%10);
    putchar(':');
    putchar('0'+second/10);
    putchar('0'+second%10);
    // then increment counter
    second++;
    if(second == 60)
    {
        second = 0;
        minute++;
        if(minute == 60)
        {
            minute = 0;
            hour++;
            if(hour == 24)
            {
                hour = 0;
            }
        }
    }
}

// 'main' объявлена как 'int' для совместности с ANSI-C
int main(void)
{
    TIMSK = (1<<6); // set OCIE1A
    TCCR1A = 0;
    TCCR1B = 0x0C; // CTC1, CK/256
    OCR1H = 0x3D; // 400000/256=15625=0x3D09
    OCR1L = 0x09;
    TCNT1H = TCNT1L = 0;
    initgraph();
    timer();
    SREG = 0x80; // SEI
    return 0;
}
```

Такой подход имеет неоднозначную оценку пользователей. С одной стороны — единая среда разработки облегчает переход к новому типу микроконтроллера. С другой стороны, вследствие такой универсальности усложняется настройка опций компиляции и оптимизации — приходится разбираться среди массы возможно неиспользуемых для конкретного микроконтроллера ключей и настроек. Нередко приходится прибегать к прямому редактированию хсл-файлов. Для ознакомления с пакетом предлагается демо-версия, имеющая следующие ограничения: максимальный размер исходного кода на языке C составляет 2 Кбайта, на ассемблере — 1 Кбайт, размер откомпилированного приложения — 2 Кбайта, размер файла для отладки — 2 Кбайта. Кроме того, формат файлов сделан таким, что они читаются только внутри этой демо-версии.

Следует отметить, что цена C-компилятора фирмы IAR Systems составляет в зависимости от приобретаемой конфигурации от 2500 до 3600 долларов США, что является существенным ограничивающим фактором для приобретения его многими российскими фирмами.

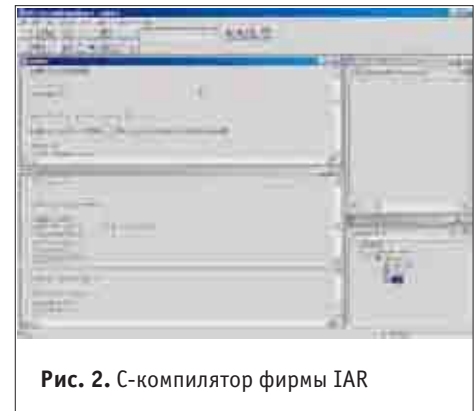


Рис. 2. C-компилятор фирмы IAR

Более интересен для рассмотрения второй из ранее упомянутых C-компиляторов. Фирма ImageCraft пошла принципиально другим путем. Ее оптимизирующий компилятор написан специально для AVR-микроконтроллеров, и, как следствие, он достаточно компактен: дистрибутив имеет объем 2,5 мегабайта, а после установки на диск компилятор занимает немногим более 4 мегабайт. При скромных размерах он имеет достаточно богатый набор характеристик. Интегрированная среда разработки (IDE) функционирует под управлением Windows 95/98/NT, включая в себя текстовый редактор с полноценной поддержкой русского языка и менеджер проекта, который формирует стандартный makefile. Обеспечивается поддержка практически всех AVR-микроконтроллеров, включая еще не выпущенные Atmega161/163 и FPSLIC AT94. Для микросхем семейства Attiny и AT90S1200, не имеющих внутреннего ОЗУ, есть пакет ICStiny-AVR. Опции компиляции можно выбрать из стандартного набора или установить самостоятельно, выбрав в поле «тип микросхемы» — custom device. В результате компиляции формируется стандартный HEX-файл для загрузки в микросхему, файл в формате COFF, который «понимает» AVR Studio, и файл с листингом

в текстовом формате. Компилятор совместим со стандартом ANSI C. Исходный текст может содержать вставки на ассемблере (строки или ассемблерные модули). Ассемблерные директивы могут включать условное выполнение, include-файлы, макросы и текстовые определения.

Библиотеки включают функцию printf, распределение памяти, строковые и математические функции. Также есть специальные функции, такие как доступ к ЭСПЗУ, АЦП и SPI-интерфейсу. При возникновении каких-либо вопросов можно обратиться к достаточно подробно написанному help-файлу или к входящей в комплект книге объемом около 120 страниц. Следует отметить, что есть возможность обратиться за помощью непосредственно на фирму ImageCraft, написав письмо по адресу info@imagecraft.com. Эта возможность была проверена дважды, и авторы подтверждают, что на следующий день они находили в своем почтовом ящике исчерпывающие ответы на заданные вопросы. Еще одно несомненное преимущество работы с программой ICC AVR от ImageCraft — это возможность полноценной работы до приобретения C-компилятора. Последняя демо-версия пакета доступна на сайте производителя программы по адресу <http://www.imagecraft.com/software/index.html>, либо на сайте российского дистрибьютора <http://atmel.argussoft.ru>, причем эта версия максимально дружелюбна: пользователь может работать с исходными текстами неограниченной длины и компилировать файл для любой микросхемы. Единственное ограничение — время работы, 30 дней. Его достаточно для разработки даже крупного проекта. По истечении месяца пользователь должен принять решение: или переформатировать винчестер, чтобы еще раз установить демо-версию, либо отдать 240 долларов и приобрести официальный дистрибутив и право на техническую поддержку.

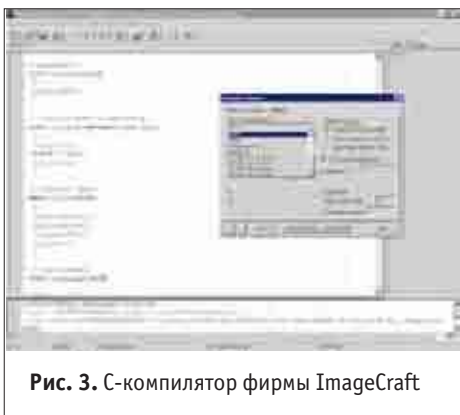


Рис. 3. C-компилятор фирмы ImageCraft

В качестве иллюстрации разработки проекта в C-компиляторе фирмы ImageCraft ниже приводится пример разработки устройства «Цифровые часы с функцией измерения и отображения температуры окружающей среды» на микроконтроллере AT90S2313. Датчиком температуры в данном приборе служит микросхема фирмы Analog Devices AD7817, а устройством отображения — любой стандартный цифробуквенный жидкокристалли-

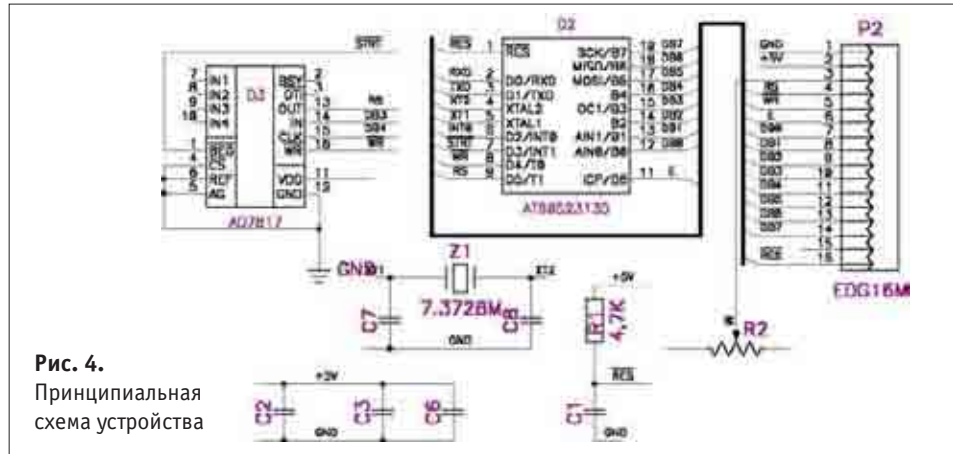


Рис. 4. Принципиальная схема устройства

ческий индикатор с контроллером типа HD44780.

Исходный код программы часов представлен на предыдущей странице (программа 1).

Частота задающего генератора = 4 МГц.

Ниже представлена функция работы с AD7817 (программа 2). Для отображения на второй строке индикатора температуры окружающей среды, следует подключить эту функцию.

Программа 2

```
#define STRT (1 << 3)
#define WR (1 << 4)
#define OUT (1 << 2)
#define IN (1 << 2)
#define CLK (1 << 4)

void strobe0(void)
{
    mcu_nop();
    PORTB &= ~CLK;
    mcu_nop();
}

void strobel(void)
{
    mcu_nop();
    PORTB |= CLK;
    mcu_nop();
}

void strobe(void)
{
    strobe0();
    strobel();
}

byte hc = 0, s = 0, m = 0,
h = 12;
void interrupt timer0_overflow(void)
{
    byte i;
    word temperature;

    PORTB &= ~IN;
    mcu_nop();
    PORTD &= ~WR;
    mcu_nop();

    for(i = 0; i < 8; i++)
        strobe();
    mcu_nop();
    PORTD &= ~STRT;
    mcu_nop();
    PORTD |= STRT;
    for(i = 0; i < 50; i++)
        mcu_nop();
    PORTD |= WR;
    mcu_nop();
    // 10 bits
    temperature = 0;
    for(i = 0; i < 8; i++)
    {
        strobe();
        temperature *= 2;
        if(PIND & OUT)
            temperature++;
    }
}
```