



ADAM-5510

как зеркало современных тенденций автоматизации

Сергей Гусев

Пролог

В начале 1998 года фирма Advantech приступила к выпуску своего очередного продукта – ADAM-5510, продолжая линию интеллектуальных УСО серии ADAM-5000. Ничего революционного, просто к серии базовых блоков ADAM-5000/485 и ADAM-5000/CAN, ориентированных на работу с управляющей машиной верхнего уровня в режиме «вопрос — ответ» по одному из популярных промышленных интерфейсов, добавился еще один — программируемый.

Новорожденный был «гол и нем», его ПЗУ не содержало никакого специального firmware для работы с собственной периферией и для общения по каналу связи. Но задатки у него были хорошие: удвоенные по сравнению со «старшими братьями» объемы ПЗУ и ОЗУ, ROM-DOS 6.22 на борту и хорошее «приданое» в виде набора утилит, библиотек функций для популярных языков программирования с исходными текстами, примерами и документацией.

Но самое главное — ADAM-5510 оставался для пользователя открытым IBM PC совместимым контроллером, гибким и легко программируемым. И уже через пару месяцев «новичок» стал бестселлером, в том числе и в России, оставив позади себя не только «старших братьев», но и потеснив старо-

жилов в сферах, где ранее традиционно применялись PLC.

Секрет успеха

Так что же все-таки определило успех нового изделия? Ведь в истории Advantech уже был пример создания IBM PC совместимого контроллера для модулей ввода-вывода серии ADAM-

4000. Но ADAM-4500 не вызвал такого бурного интереса у пользователей и нашел гораздо меньший круг применения, чем все остальные модули серии ADAM-4000.

Для тех, кто не знаком с этими модулями УСО, позволю себе напомнить об их организации и принципах построения систем на их основе (рис. 1).

Все модули ввода-вывода серии 4000 состоят из микроконтроллера с жестким «защитым» в ПЗУ алгоритмом, коммуникационного интерфейса и собственно УСО. Пропускная способность канала связи на основе RS-485 составляет 1200-115 200 бит/с, что соответствует примерно 5-480 обменам в секунду по протоколу, принятому для модулей этой серии.

Контроллер ADAM-4500 был призван заменить машину верхнего уровня в «цепочке» модулей серии 4000 и выполнять одновременно задачи опроса УСО и управления. Однако такое решение оказалось не очень удачным. Заменить машину верхнего уровня полностью не удалось уже ввиду отсутствия дисковых накопителей для ведения архива, а уменьшить время реакции в цикле управления не получилось, так как все модули ввода-вывода и процессор в модуле ADAM-4500 оказались соединенными «узким» интерфейсом RS-485.

Таким образом, все потенциальные возможности процессора 4500 оказались не раскрыты, и он нашел себе применение только в относи-



Внешний вид ADAM-5510

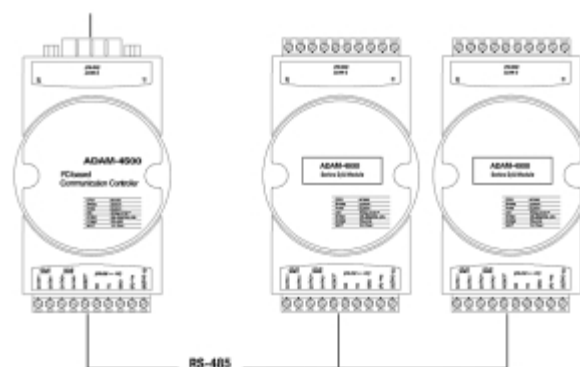


Рис. 1. Управление сетью из модулей ADAM-4000 с помощью контроллера ADAM-4500

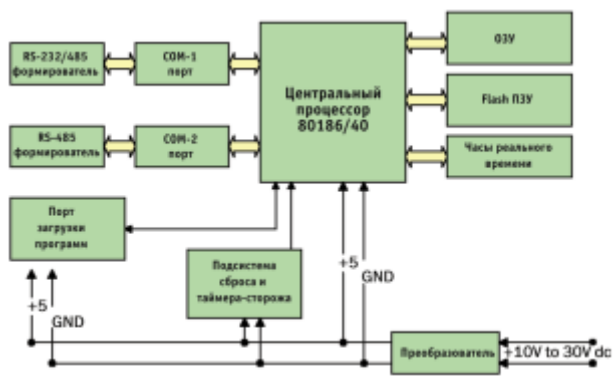


Рис. 2. Структурная схема контроллера ADAM-4500

тельно медленных системах сбора информации и управления.

С серией ADAM-5000 все обстоит совершенно иначе. Базовый блок содержит процессор, аналогичный по всем параметрам процессору ADAM-4500 (рис. 2). Однако он общается с модулями ввода-вывода по локальной шине с пропускной способностью около 10 Мбайт/с. Это значит, что в реальной программе, написанной на языке низкого уровня, скорость опроса одного дискретного входа может достигать 1 МГц с учетом накладных расходов на обслуживание сторожевого таймера и последовательного интерфейса. Таким образом, узким местом в системе управления, собранной на основе ADAM-5000, становится внешний интерфейс. Даже при использовании интерфейса CAN на скорости 500 кбит/с частота сканирования одного канала не превышает 10 кГц.

Появление модуля ADAM-5510 позволило расположить управляющую программу пользователя непосредственно в памяти контроллера, что сократило время реакции в контуре управления виртуально до 5-10 мкс. Этот факт впервые позволил применить системы на базе ADAM-5510 для решения задач «жесткого» реального времени, где ранее применялись «классические» PLC.

Первые впечатления

Что же получает пользователь после приобретения ADAM-5510 (рис. 3)? Это небольшая коробка, содержащая собственно контроллер, руководство

пользователя и дискету с утилитами. Приятно удивляют малый вес изделия (менее 300 г) и габариты (231×110×75 мм). Внутри же, согласно описанию, следующее.

Модуль процессора

- Процессор: 80188-40 МГц.
- Флэш-память: 256 кбайт (170 кбайт доступно для пользователя).
- Операционная система: ROM-DOS (совместимая с MS-DOS 6.22).

- Статическое ОЗУ: 256 кбайт (234 кбайт доступно для пользователя).
- Таймер BIOS, часы-календарь реального времени, сторожевой таймер.
- Последовательный порт COM1: RS-232 (с полным набором сигналов).
- Последовательный порт COM2: RS-485 (DATA+, DATA-, полудуплекс).

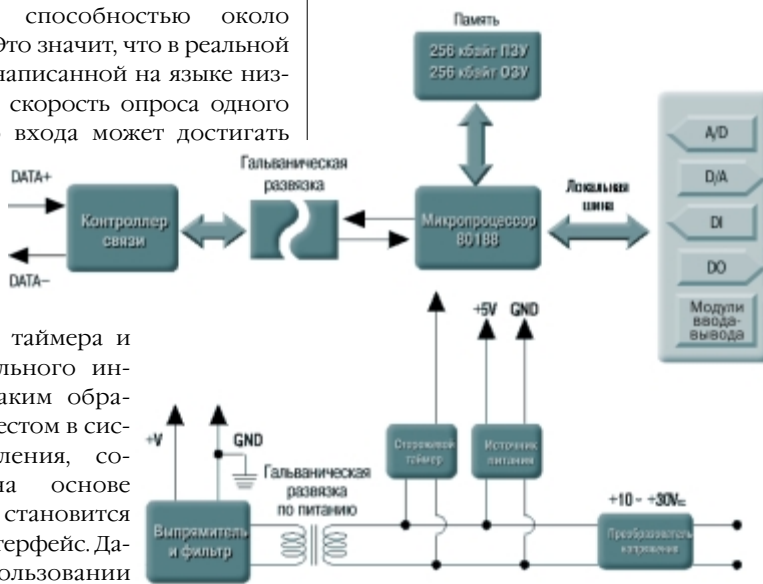


Рис. 3. Структурная схема контроллера на базе ADAM-5510

- Специальный порт консоли: RS-232 (Tx, Rx, GND, используется только для загрузки программ и консольного вывода DOS).

Источник питания

- Входное напряжение нестабилизированное от +10 до +30 В постоянного тока, с защитой от переплюсовки.
- Потребляемая мощность: 2,0 Вт.

Условия эксплуатации

- Рабочий диапазон температур: от минус 10 до +70°C.
- Диапазон температур хранения: от минус 25 до +70°C.
- Относительная влажность: от 5 до 95 % без конденсации.

С чего начать

Для первого включения вам понадобятся модемный кабель и источник питания на +24 В. Соединив модемным кабелем свободный COM-порт своего компьютера с консольным портом и запустив с прилагаемой дискеты программу Adam5510.exe, вы, весьма вероятно, увидите следующую картину (рис. 4).

Для начала вам предстоит указать программе, какой COM-порт вы собираетесь использовать для связи с ADAM-5510. Не волнуйтесь, если при этом ваш манипулятор типа Microsoft Mouse for COM port вдруг перестанет работать. Дело в том, что бедные разработчики программного обеспечения обходятся мышами типа PS/2 и не имеют возможности тестировать свои продукты на устаревших компьютерах.

Далее перейдите к пункту меню Terminal и подайте напряжение питания на ADAM-5510. Через несколько секунд на экране появится примерно следующее (рис. 5).

Вы попадаете в привычную и комфортную среду MS-DOS, где работают

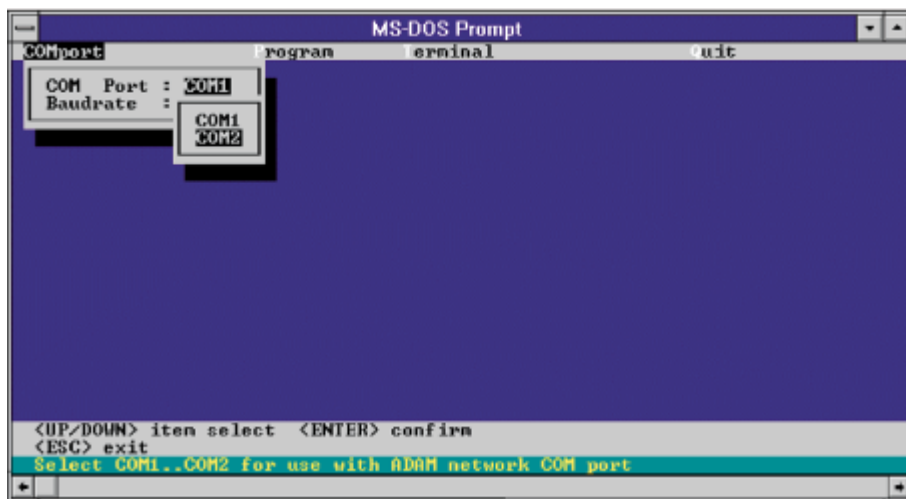


Рис. 4. Выбор COM-порта

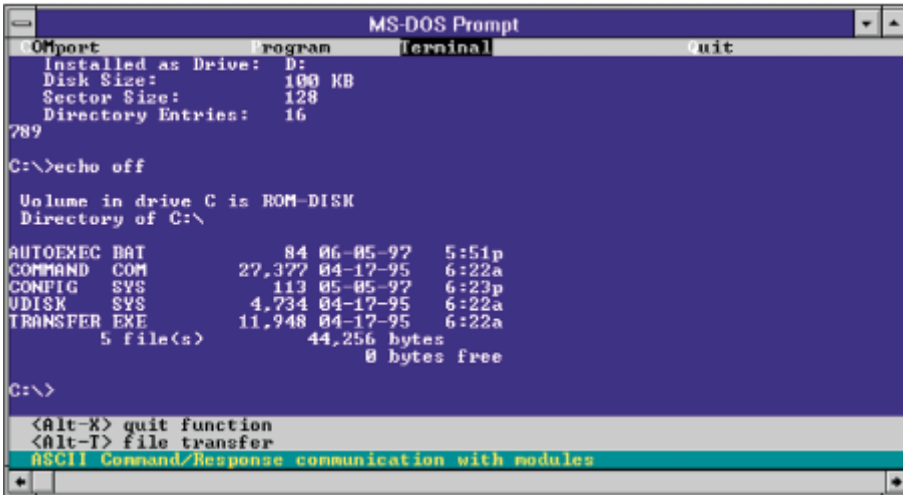


Рис. 5. Вид экрана при работе с ROM-DOS

понятные команды. По умолчанию в системе есть два «диска», один из которых, D:> (емкостью 100 кбайт) образован утилитой VDISK.SYS из основной памяти. Для приложений остается лишь 130 кбайт, но и этого кажется вполне достаточно для большинства задач. Теперь вы готовы приступить к созданию своей Первой Программы. Для этого вам нужно научиться отлаживать созданные шедевры и загружать их в контроллер (считается, что писать на Си или Паскале вы уже умеете). Начнем с простого.

Способы загрузки программ

Как вы, наверно, уже догадались, для загрузки программ в электронный диск в области статической памяти достаточно нажать комбинацию клавиш <Alt-T> и указать имя файла. При этом ваша программа с минимальными затратами усилий оказывается в контроллере и может быть запущена из командной строки DOS. Это хороший способ для первого шага, но следует помнить, что в этом случае в программе необходимо предусмотреть возможность ее завершения по команде с консоли, ибо при нажатии кнопки RESET последует инициализация диска D:> и программа будет уничтожена. Соответственно, запись на диск D:> не подходит для отладки программ, использующих сторожевой таймер в режиме сброса.

Другой способ загрузки — это программирование флэш-памяти (диск C:>). Для этого предназначен специальный пункт меню Program. Перед выполнением этого ответственного шага необходимо знать следующее.

1. Директория ALLFILE, которая находится на поставочной дискете, содержит образцовые файлы, которые будут переноситься на флэш-диск C:>.
2. Файл DEMO-DIS.HEX получается из файлов в директории ALLFILE в результате работы программы ROMDISK.EXE.
3. Файлы ROM-DOS.HEX и ADAMINI.HEX представляют собой образы ROM-DOS 6.22 и мини-BIOS, которые подстегиваются к файлу DEMO-DIS.HEX программой HEXCAT.EXE и упаковываются в файл ADAM-DEM.HEX, который, собственно, и загружается во флэш-память.
4. Все эти процессы начинаются при нажатии кнопки меню «Program» и являются обратимыми вплоть до появления на экране надписи «Now is programming FLASH...».

Таким образом, для записи вашей программы на флэш-диск 5510 ее нужно поместить в директорию ALLFILE и, если необходимо, указать ее в файле autoexec.bat из этой же директории. Если программа требует более 130 кбайт оперативной памяти, необходимо удалить из файла config.sys упоминание о vdisk.sys. Затем запускается программа Adam5510.exe и выбирается пункт меню Program. Надеюсь, вы догадались, что при таком методе программирования диска будет очень полезно иметь резервную копию директории ALLFILE...

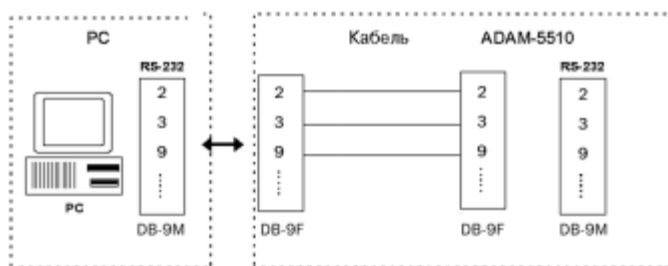


Рис. 6. Соединение PC и ADAM-5510 нуль-модемным кабелем

Позволю себе заметить, что если в процессе подготовки к программированию (после нажатия кнопки меню Program, но до завершения процесса копирования образа в ОЗУ контроллера) нажать клавишу Esc, то процесс программирования прервется, а на экране окажется приглашение:

e8b:_

Не удивляйтесь, нажмите привычно «?» и «Enter», и вы увидите, что это специализированный отладочный монитор типа DEBUG.EXE для DOS, наличие которого в комплекте поставки почему-то нигде не оговорено. С его помощью можно просматривать и модифицировать содержимое памяти, запускать программы и обращаться непосредственно к установленным модулям ввода-вывода. Опытному пользователю этот инструмент может сослужить хорошую службу.

Существует еще один способ загрузки программ в ADAM-5510 для отладки — вообще их туда не загружать, а пользоваться удаленным отладчиком-загрузчиком TDREMOTE.EXE, поставляемым в составе систем программирования фирмы Borland, но об этом отдельно...

Отладка программ

Во-первых, для использования утилиты TDREMOTE необходимо приобрести или изготовить еще один кабель — нуль-модемный — и соединить им COM1 ADAM-5510 и один из COM-портов вашей персоналки (рис. 6).

Во-вторых, необходимо переделать стандартный TDREMOTE в TDADAM с помощью специальной утилиты UPDATE.EXE, поставляемой вместе с контроллером. Успешно переделываются только TDREMOTE из комплекта Borland Turbo C++ v.2.0 и 3.0. Пользователям C++ v.3.1, а также Borland Turbo Pascal v.4.5 и выше, придется поискать TDREMOTE от более младших версий, но далее все будет работать нормально.

Причина неработоспособности стандартного удаленного отладчика на контроллере ADAM-5510 кроется в том, что коммуникационные порты в контроллере находятся в непривычном месте — в области адресов памяти, а не адресов ввода-вывода. Подробнее об этом в следующем разделе.

Особенности аппаратной реализации

Конфигурация контроллера ADAM-5510 несколько отличается от привычной PC-архитектуры. Так как ни один из моду-

лей ввода-вывода не использует прерываний, в системе возможны только два типа прерываний: от таймера-сторожа и от коммуникационных портов (табл. 1).

Таблица 1. Доступные прерывания

Источник прерывания	Адрес вектора
Немаскируемое прерывание (NMI)	02h
Прерывание от COM1	0Ch
Прерывание от COM2	0Eh

Распределение памяти в контроллере показано в табл. 2.

Таблица 2. Карта памяти ADAM-5510

0xF8000 — 0xFFFFF	Отладочный монитор
0xF6C00 — 0xF7FFF	Мини-BIOS
0xCC000	Начальный адрес флэш-диска (около 171 К)
0xC0000	Начальный адрес ROM-DOS (около 48 К)
0x40000 — 0xBFFFF	Не используется
0x00400 — 0x3FFFF	ОЗУ, доступное для программ
0x00000 — 0x003FF	Системная область
0x003F8 — 0x003FF	COM1
0x002F8 — 0x002FF	COM2
0x00070 — 0x00071	Часы реального времени

Как видно из сказанного, программирование устройства требует некоторого навыка и учета аппаратных особенностей. Например, пользователям языка программирования C++ (и аналогичных) придется забыть о функции «delay()» и пользоваться некими эквивалентами (см. функцию «delay1()» из примера, приведенного далее).

Предвижу возмущение опытных программистов, но, поверьте мне, такими же методами пользовались и разработчики библиотек, поставляемых вместе с ADAM-5510. К сожалению, объем данной статьи не позволяет поговорить о библиотеках более подробно. Скажу только одно. Библиотеки хорошо документированы и поставляются вместе с исходными текстами. Пользователю, желающему быстро написать работоспособную программу, они, безусловно, окажут хорошую помощь, а опытному программисту они доставят немало веселых минут (особенно в части функций, работающих с модулями аналогового ввода). Надеюсь, что модификация вами библиотечных функций не станет для вас самоцелью, а лишь поможет лучше узнать специфику оборудования. Иногда полезно помнить, что несколько миллисекунд процессорного времени несоизмеримо в денежном отношении с несколькими рабочими днями квалифицированного программиста.

Здесь я позволю себе привести пример законченной программы, превращающей ADAM-5510 в отказоустойчивый будильник (пример 1). Здесь иллюстрируются способы работы с системным таймером, таймером-сторожем, консолью и устройствами ввода-вывода (в данном случае — модулем релейного вывода ADAM-5060).

К числу курьезных несоответствий документации на библиотеки стоит отнести тот факт, что функция считыва-

ния сетевого адреса, устанавливаемого внешними переключателями, возвращает результат в инверсном коде.

Но не стоит судить строго подобные огрехи в бесплатном (!) и хорошо документированном программном обеспечении. Главное все-таки в ADAM-5510 — это то, что сама аппаратура получилась довольно надежной, сбалансированной и открытой для пользователя. Производительность процессора и скорость локальной шины, к которой подключены модули, позволяет опытному пользователю решать на нем такие задачи, о которых нельзя было и мечтать, работая с системами ADAM предыдущих поколений.

Например, классическая тестовая задача опроса 64 цифровых каналов ввода на предмет подсчета поступающих импульсов (с определением фронта, защитой от дребезга, накоплением полученных значений в 32-разрядных переменных и с обслуживанием сторожевого таймера), написанная на TC++ v.3.0 с использованием поставляемых библиотек, заняла около 10 кбайт кода и позволила делать около 1700 циклов в секунду.

Но что делать, если пользователь не хочет знать ничего из описанного и тем более из неописанного? Если вид программы, написанной на Си, вызывает у него сожаление о бесцельно прожитых годах, а перспектива копаться в чужих исходных текстах — активное нежелание бесцельно потратить время и деньги на зарплату программистам?

Пример 1

```
#include <time.h>
#include <adam5510.h>
#define FALSE 0
#define TRUE 1
#define ESC 27
void delay1(void);
void main(void)
{
    int key;
    struct dostime_t    t;
    wdt_enable();      // Инициализация сторожевого
                      // таймера
    led_init();        // Инициализация порта, управ-
                      // ляющего светодиодами
    if (!io_init())    // Инициализация устройств
                      // ввода-вывода
    {
        printf(«Устройства ввода-вывода не обнаружены\n»);
        exit(0);
    }
    while (TRUE)
    {
        if (kbhit())
            if ((key = getch()) == ESC) break;
        wdt_clear();  // Сброс таймера-сторожа
        _dos_gettime(&t); // Считывание показаний сис-
                          // темного таймера
        printf(«Текущее время:%2d:%02d:%02d\n»,
            t.hour, t.minute, t.second, t.hsecond);
        if (t.hour==8)
        {
            if (t.minute==30) io_set_do(0,0x00f);
            // Если 8:30 - включение
            // «будильника»
            if (t.minute==31) io_set_do(0,0x0000);
            // Если уже 8:31 - выключение
        }
        // Далее следует бессмыслен-
        // ное мигание светодиодами...
        led1(1); led2(0); led3(0); led4(0); delay1();
        led1(0); led2(1); led3(0); led4(0); delay1();
        led1(0); led2(0); led3(1); led4(0); delay1();
        led1(0); led2(0); led3(0); led4(1); delay1();
    }
}
void delay1(void) // А это, извините, задержка...
{
    int i,j;
    for (i=0;i<500;i++)
        for (j=0;j<200;j++);
}
```

Если пользователь привык мыслить в терминах Теории автоматического управления, а сладкие призраки PLC от гигантов рынка автоматизации и мечты о системах сквозного программирования от апологетов МЭК-1131 не становятся реальностью только из-за скудности бюджета и постоянного желания самому заработать хоть немного побольше? В этом случае пользователю будет полезно ознакомиться со следующим разделом.

Ultralogic

Надеюсь, читатели уже достаточно знакомы с этой средой разработки программ, чтобы не задерживаться на ее подробном описании. Остановимся лишь на тех аспектах, которые важны для пользователей ADAM-5510.

Программирующий на Ultralogic может не знать ничего о внутреннем устройстве ADAM-5510. Для него важно понимать, какие модули ввода-вывода присутствуют в его контроллере и на какие именно линии в них приходят те или иные сигналы. Эта информация задается при определении модели и назначении переменных (рис. 7).

После этого достаточно в поле редактора изобразить функциональную схему (в формате МЭК 1131.3), выражающую зависимость выходных переменных от входных, и нажать на кнопку «компиляция проекта». Полученный в результате этого исполняемый модуль будет содержать вашу стратегию, а также поддержку сети на базе RS-485 и, если это нужно, таймера-сторожа. Ваш ADAM-5510 превращается в полный аналог PLC.

Для решения типовых задач управления Ultralogic содержит обширную библиотеку типовых «кубиков», таких как, например, ПИД-регуляторы, нормализаторы ввода сигналов термопар, функции работы с таймером, триггеры, амплитудные ограничители, уловители фронтов, счетчики и т. п. Скорость создания работоспособных приложений при использовании Ultralogic оказывается на порядок выше, чем при программировании на языках класса С или Pascal. Важно, что в состав пакета входит мощный отладчик, который позволяет отладить ваш проект в эмуляторе без загрузки его в ADAM-5510.

Но что самое интересное, скорость выполнения задач, скомпилированных на Ultralogic, существенно превосходит (!) скорость выполнения аналогичных задач, написанных на С++ с использованием поставляемых производителем библиотек. Так, например, описанная здесь тестовая задача по подсчету им-

пульсов на 64 каналах показала производительность около 4500 циклов в секунду. И это на фоне активного обмена по сети!

В чем же тут секрет? Дело в том, что при включении модели ADAM-5510 в Ultralogic программистами была проведена большая работа по оптимизации методов работы с аппаратурой и критическому переосмыслению библиотек, поставляемых производителем. Особых успехов удалось достичь в оптимизации драйверов работы с АЦП.

В результате этих работ программа, написанная на Ultralogic, может получать новые данные с аналоговых входов с интервалом от 10 до 100 миллисекунд (в зависимости от требуемой точности), причем занимаясь в это время другими делами. Для сравнения, при использовании стандартных библиотек цикл ввода с АЦП занимает 0,6 с при полной занятости процессора.

позволяющий подключать сеть из ADAM-5510 к продвинутым SCADA-системам, таким как Genesis фирмы Iconics.

Итоги

Учитывая все сказанное, можно с уверенностью утверждать, что новый программируемый контроллер фирмы Advantech пользуется заслуженной популярностью, так как дает пользователю широчайшие возможности применения, позволяет применять различные методы программирования. Он стал одинаково близок как сторонникам открытых IBM PC совместимых платформ, так и поклонникам PLC-систем, поскольку сократил пропасть, разделяющую эти два класса оборудования. Его хорошо продуманная и сбалансированная аппаратная организация, отлично зарекомендовавшая себя на практике, обилие программного обеспечения, появляюще-

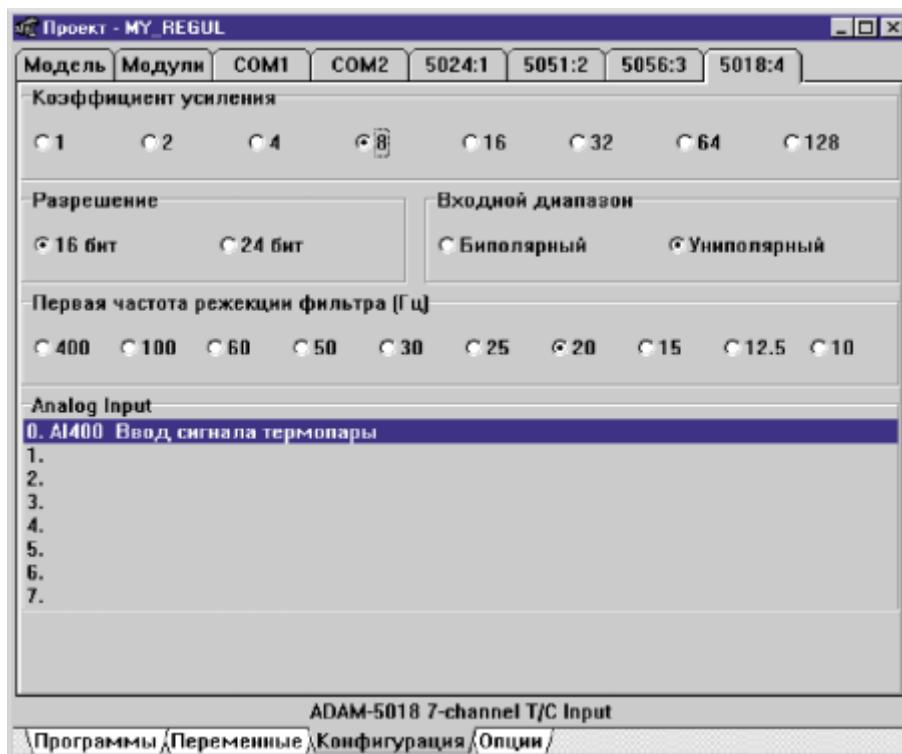


Рис. 7. Экран настройки параметров АЦП

Отдельно стоит упомянуть и удобство подключения ADAM-5510, запрограммированного с помощью Ultralogic, или даже целой сети из них к компьютеру верхнего уровня. Существует специальная программа, оформленная в виде DLL-модуля для связи с самой «народной» SCADA-системой Genie, остающаяся последней только функции визуализации, в то время как решением задач сбора информации и управления занимается непосредственно ADAM-5510. Существует также DDE-сервер,

гося для него, лишний раз свидетельствуют о том, что пользователи, которые взяли его в качестве основы для решения своих задач, сделали правильный выбор. ●