

Разработка и отладка систем на базе микросхем фирмы Xemics

В предыдущей статье («Интеллектуальный датчик с малым энергопотреблением» — «КиТ» № 8'2003) рассказывалось о структуре конкретного устройства, в котором применяются микросхемы фирмы Xemics. Собственно само устройство представляет узкоспециальный интерес для профессионалов в данной области. Но аппаратно-программные средства для разработки более широкого класса устройств на базе микросхем фирмы Xemics интересны более широкому кругу специалистов. Для них и предлагается данная статья.

Сергей Синютин

siniutin@tppark.ru

Любого разработчика при знакомстве с новым микропроцессорным набором после изучения его технических характеристик прежде всего интересует, с помощью каких технических и программных средств можно с ним работать. По поводу программных средств особых проблем обычно не возникает — правилом хорошего тона для всех фирм-производителей стал обязательный выпуск ассемблера, отладчика и интегрированной среды разработки. Но после разработки конкретного устройства на базе микроконтроллера с использованием встроенных периферийных устройств, отладить его на программном эмуляторе практически не возможно. Особенно это касается приложений, работающих с низкоуровневыми аналоговыми сигналами, интерфейсных схем, схем, реализующих обработку прерываний от внешних сигналов. Средства эмуляции внешней среды программных отладчиков недостаточны для профессиональной работы. Поэтому, спроектировав систему, подобную датчику из предыдущей статьи, необходимо решить вопрос, с ее отладкой.

Фирма Xemics предлагает для этого набор демонстрационных плат с возможностью программирования и отладки в комплексе со своими средствами разработки. Но для разработчика (особенно если он работает с 5–10 микроконтроллерами различных типов) приобретение новых недорогих средств, чтобы только «попробовать» новый микроконтроллер, обычно вызывает проблему (по крайней

мере, это трудно объяснить руководству предприятия). Отсюда возникают две задачи:

- разработка достаточно простого программатора;
- разработка универсального стенда для отладки разработок.

Начнем с программатора. Собственно процедура программирования подробно описана в документе **Technical Note TN8000.02 XE88LC01-03-05 programming**, но целью данной статьи является не перевод данного руководства, а руководство к действию, то есть демонстрация, как можно воспользоваться диаграммами и алгоритмами, чтобы быстро получить результат — работающий программатор. Прежде всего, приведем некоторые сокращения, используемые в руководстве (табл. 1).

Фирма применила довольно архаичный метод программирования, использующий источник повышенного напряжения (и оно еще весьма прихотливо изменяется во время программирования!). Вместе с тем, алгоритмы программирования даны достаточно подробно и в нотации, похожей на язык Modula.

Нотация и подсказала автору, как достаточно быстро и легко реализовать их на другом типе контроллера. Реализацию программатора было решено организовать методом «раскрутки». В начале на другом микроконтроллере, для которого имеются инструментальные средства, реализуется первая версия программатора, а в дальнейшем, если будет необходимость, с его помощью можно будет реализовать программатор на базе микроконтроллера фирмы Xemics.

В качестве инструментального контроллера был выбран контроллер фирмы Atmel Atmega8-16AI. Выбор данного контроллера обусловлен наличием простого программатора PonyProg (подключается к LPT-порту ПК и содержит одну микросхему 1533АП5) и системы программирования E-LAB Pascal (www.E-LAB.de).

Собственно синтаксис в системе E-LAB ближе к Modula, что и явилось причиной выбора ее в качестве инструментальной системы программирования. Кроме того, E-LAB — это не просто компиля-

Таблица 1

Сокращение	Описание
MTP	Multiple Time Programmable — Многократное программирование
Crck	Coolrisc clock — Тактовая частота контроллера
Ptck	Peripheral test clock — Тактовая частота периферийных устройств
HV	High Voltage — Повышенное напряжение
Testck	serial clock for shift instruction — Тактовая частота для сдвига команды
Testin	serial input for shift instruction — Вход данных для приема команды
Testout	serial output for shift instruction — Выход данных для чтения результата

Таблица 2

Имя контакта	Наименование	Назначение в режиме программирования	Значение в активном состоянии	Тип сигнала
Vbat	Вход напряжения питания	Vdd	+5 В	Постоянное напряжение
Vss	Общий	Vss — Общий	0 В	Постоянное напряжение
RESET	Цифровой вход < Vbat	Сброс	0 В	Постоянное напряжение
Vreg	Выход напряжения питания	Vreg (подсоединить конденсатор = 1 мкФ)		
OscIn	Цифровой вход < Vbat	Crck	Vss или Vdd	Переменное напряжение
OscOut	Цифровой вход < Vbat	Ptck	Vss или Vdd	Переменное напряжение
VPP	Вход напряжения программирования	HV	Vdd или Vddt или Vddhigh	Переменное напряжение
PortA(O)	Цифровой вход < Vbat	Testin	Vss или Vdd	Переменное напряжение
PortA(1)	Цифровой вход < Vbat	Testck	Vss или Vdd	Переменное напряжение
PortB(O)	Цифровой выход < Vbat	Testout	Vss или Vdd	Переменное напряжение

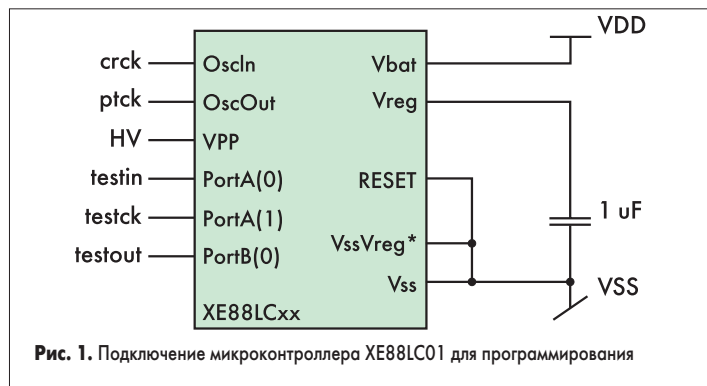


Рис. 1. Подключение микроконтроллера XE88LC01 для программирования

тор, а небольшая ОСРВ, допускающая на таком небольшом кристалле, как Atmega8, такие вещи, как параллельные процессы, многозадачность и т. п.

Демо-версия E-Lab позволяет реализовать все алгоритмы программатора (для Atmega8 демо-версия позволяет запрограммировать весь объем Flash-памяти программ).

В таблице 2 показаны все выходы микроконтроллера XE88LC01, используемые при программировании.

В таблице 3 приведены значения параметров, изменяющихся при программировании. Из таблицы видно, что Vdd должно иметь минимум 3 уровня (5, 7, 11,55 В). Для реализации этого в схему программатора введен стабилизатор с коммутируемым напряжением стабилизации на основе линейного регулятора напряжения LM317.

У такого стабилизатора напряжение стабилизации равно:

$$U = 1.25 \times \left(1 + \frac{R_2}{R_1} \right),$$

где $R_1=249$ Ом, а с помощью изменения R_2 можно получить требуемые напряжения. Коммутировать резисторы, определяющие R_2 , будем транзисторными ключами (хотя более подходящие результаты дает использование специализированных ключей, например ADG661).

Из таблицы 4 видно, что вполне возможно при тактовой частоте микроконтроллера 16 МГц программно реализовать временные диаграммы сигналов для программирования.

Для организации управления напряжением программирования использовались выходы PD4, PD6 и PD7 порта D. Вывод PD5 используется для выключения напряжения программирования. Выводы PC0–PC3 используются для программной последовательной передачи команды в программируемый микроконтроллер и приема информации из него.

Таблица 3

Параметр	Min	Max	Единицы изм.	Примечания
Vdd	4,5	5,5	В	
Vddt	Vdd+2,0	Vdd+2,5	В	
Vddhigh	11,55	11,65	В	Inpr. < 30 мА
Конденсатор Vreg	0,9	1,1	мкФ	
Температура	10	40	°С	Тип. = 25

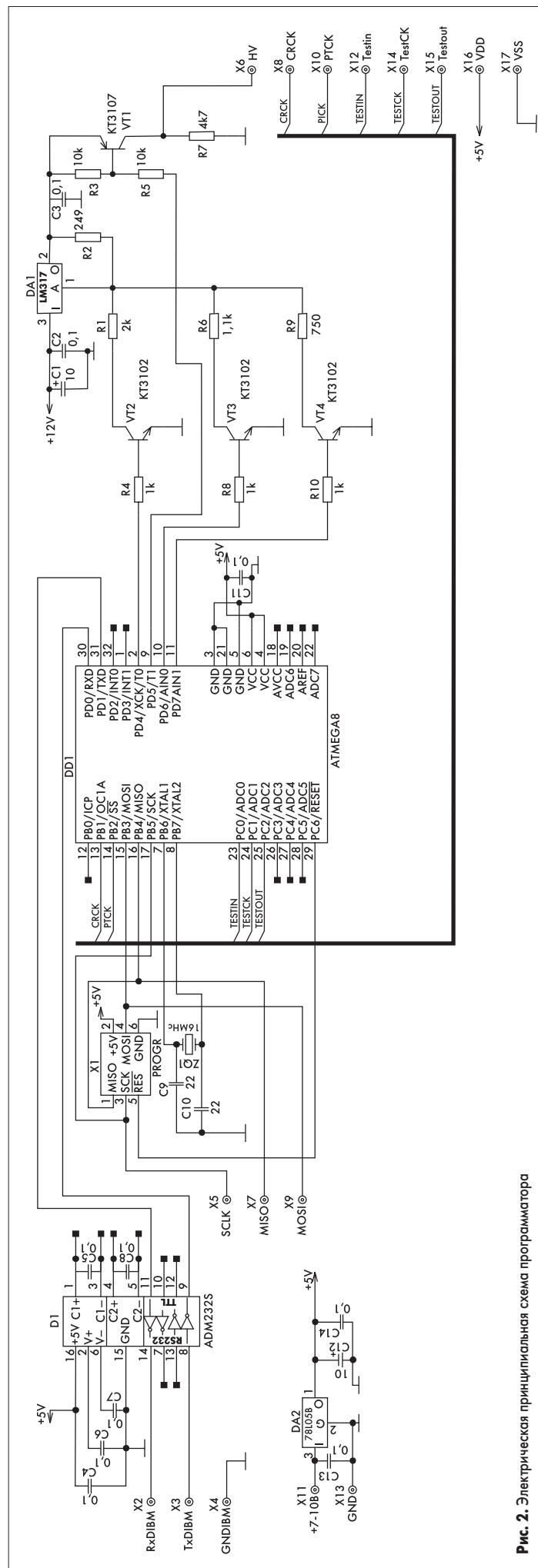


Рис. 2. Электрическая принципиальная схема программатора

Таблица 4

Параметр	Описание	min	max	Ед. изм.
ta	Предустановка Testin относительно фронта testck	50		нс
tb	Задержка Testin stable после фронта testck	50		нс
tr	Время переднего фронта Testck		10	нс
tf	Время заднего фронта Testck		10	нс
th	Длительность Testck при высоком уровне сигнала	125		нс
tl	Длительность Testck при низком уровне сигнала	125		нс
to	Длительность Crck и ptok в высоком/низком уровне	800		нс
trfc	Передние и задние фронты Crck и ptok		10	нс
ttf	Быстрая rick-серия во время стирания и верификации	115	125	нс

Выводы PB1–PB2 используются для организации тактовых серий микроконтроллера и периферии. Связь управляющего контроллера с персональным компьютером осуществляется через интерфейс RS232, для чего применяется преобразователь уровня на микросхеме D1. Схема его включения особенностей не имеет. X1 — разъем программатора Pony-Prog.

Для разработки программного обеспечения воспользуемся алгоритмами из указанного выше технического руководства. В описании алгоритмов вводится понятие *Macro*. На самом деле это — обычная процедура в смысле Pascal или C. Поэтому технология преобразования алгоритмов программирования в программу на E-LAB следующая:

- каждую Macro заменяем процедурой или функцией;
- в основную программу добавляем процедуры инициализации выводов портов, организации тактовых серий и т. п.;
- организуем диалог с персональным компьютером для приема команд и данных и программирования.

Покажем на примере конкретной Macro, как производится такая замена. Например, возьмем следующий фрагмент из руководства по программированию:

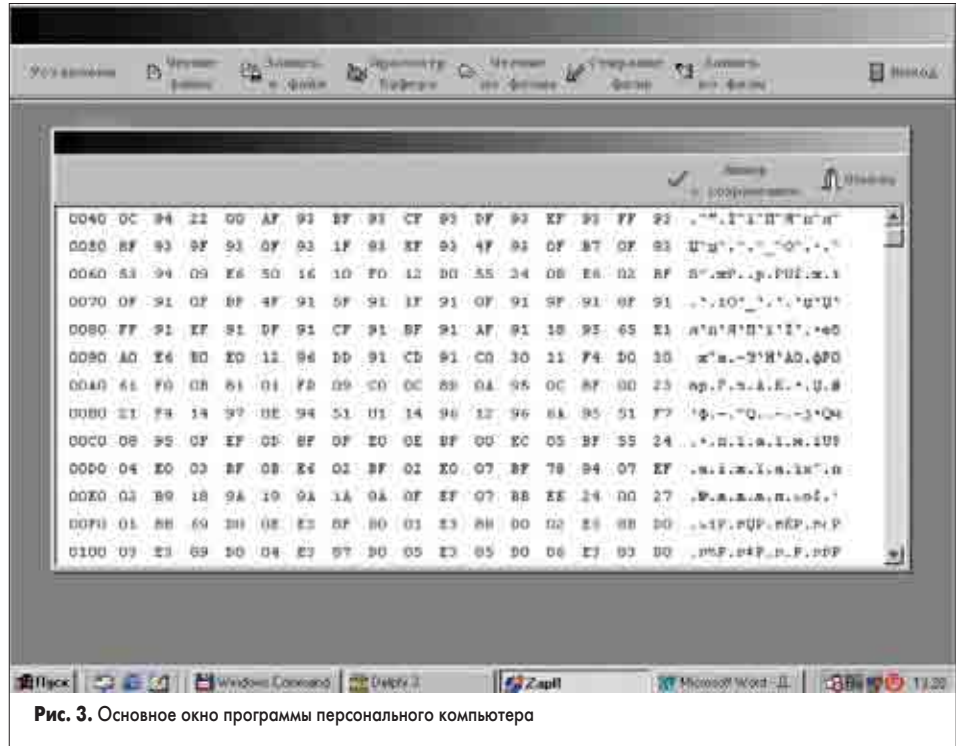


Рис. 3. Основное окно программы персонального компьютера

```
macro signature
shift_instruction(11111111111111111111)
cycle_crck(1)
shift_signature
cycle_crck(1)
end macro
```

В системе E-Lab этот фрагмент будет выглядеть так:

```
procedure signature;
begin
shift_instruction(%11111111111111111111);
cycle_crck(1);
shift_signature;
cycle_crck(1);
end;
```

В общем случае замена Macro на Procedure — вещь довольно рутинная и не представляющая особой сложности. Диалог же с персональным компьютером каждый разработчик может организовать по своему желанию. Автор воспользовался для этого системой

программирования Delphi 5. В данной системе с помощью готовых компонентов достаточно просто и быстро реализуются такие вещи, как ввод файла, редактирование в двоичном и шестнадцатеричном формате, работа с портом RS232. Основное окно программы персонального компьютера показано на рис. 3.

В дальнейшем программа персонального компьютера может развиваться, включая в себя элементы работы с другими объектами стенда (кроме программатора). В частности, для отладки датчика использовались программы работы с преобразователем «емкость-напряжение».

При разработке стенда учитывалась необходимость работы как с микроконтроллером, так и с преобразователем «емкость-напряжение». Причем на начальном этапе работы преобразователь емкости настраивался и программировался через Atmega8, а дан-

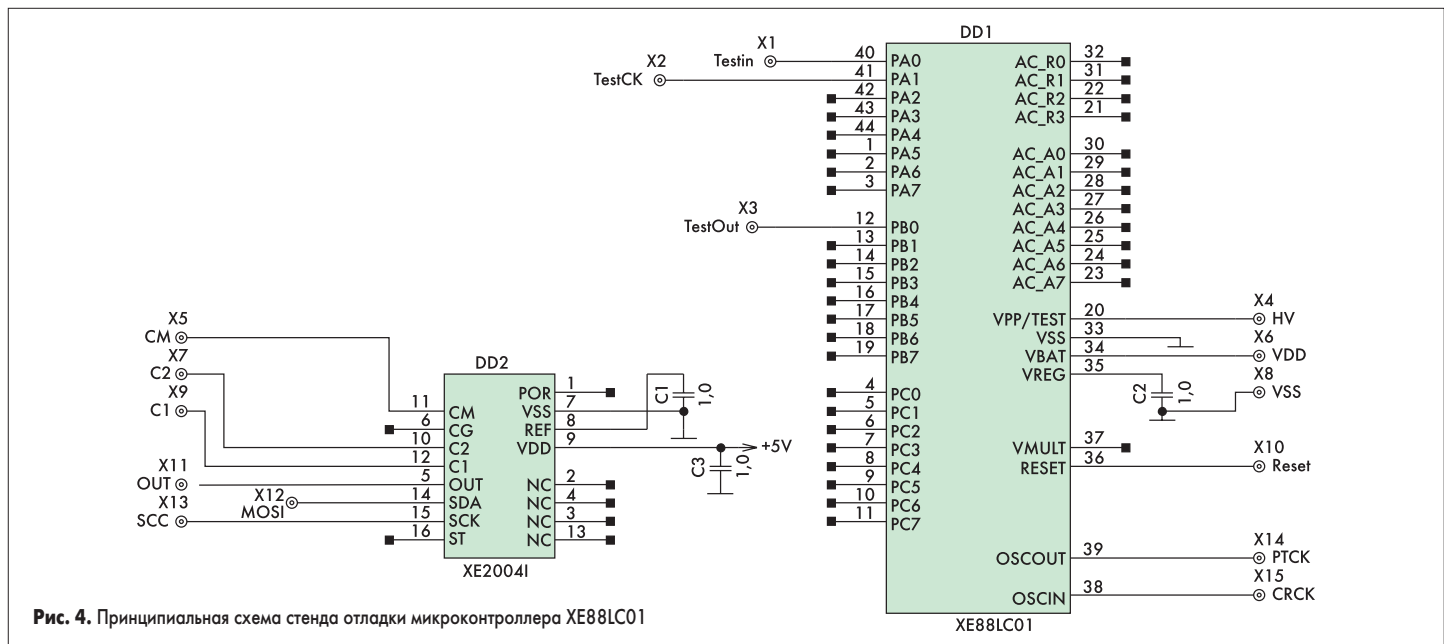


Рис. 4. Принципиальная схема стенда отладки микроконтроллера XE88LC01

ные с него вводились через АЦП Atmega8. Схема стенда показана на рис. 4.

Для простоты здесь не показаны Pin неиспользуемых при программировании ножек, а также аналоговые элементы для ADC и DAC микроконтроллера XE88LC01. Конструктивно программатор выполнен на одной плате со стендом, а необходимые соединения для программирования осуществляются перемычками.

Рассмотренный стенд-программатор не содержит дефицитных деталей, а главное —

«открыт», то есть имеется возможность его расширения как в части программы инструментального микроконтроллера, так и в части программы на ПК, не говоря уже о составе аппаратных средств. Такие расширения будут необходимы, и это ясно видно уже из того, что даже сейчас некоторые контроллеры фирмы Xemics программируются по другим алгоритмам, достаточно хорошо документированным.

Приведенные схемы и методика преобразования алгоритмов в программы для инст-

рументального микроконтроллера являются лишь своеобразным трамплином для быстрого освоения таких многообещающих микроконтроллеров, какими являются изделия фирмы Xemics. ■

Использована информация

1. www.xemics.com
2. www.atmel.com
3. www.e-lab.de
4. www.analog.com