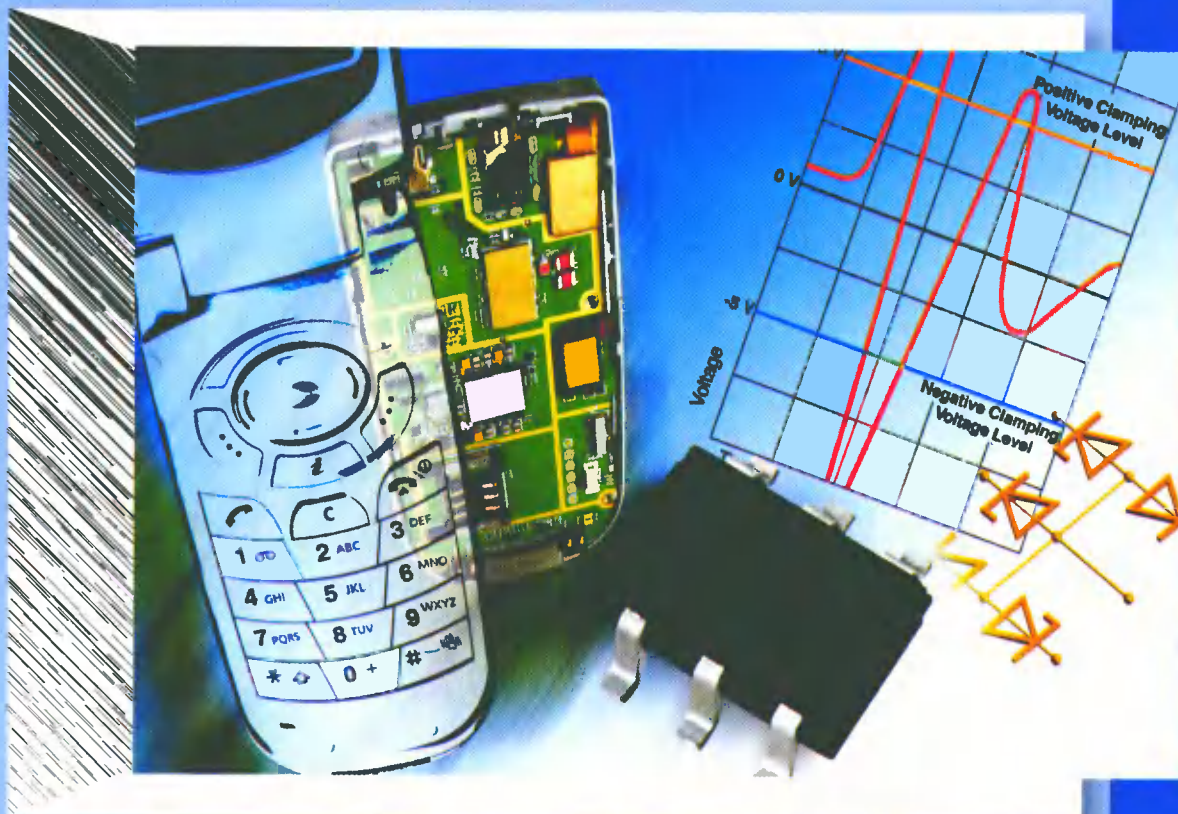
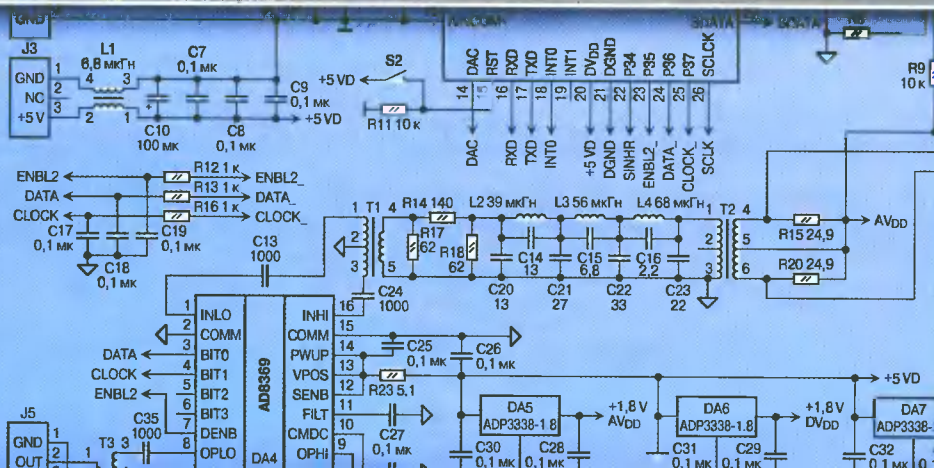




Супервизоры питания и их применение
Активные корректоры коэффициента мощности ON Semiconductor
Низковольтные аналоговые микросхемы Maxim/Dallas
Макетная плата для микропроцессоров PIC16A62x
Преобразователь сигналов дифференциального емкостного датчика
Автономный считыватель штрихкода



Главный редактор

Сергей Бирюков

Зам. главного редактора

Сергей Кузнецов

Редакционная коллегия

Павел Асташкевич
Александр Фрунзе
Виктор Иовчик

Дизайн и верстка

Ирина Чикина

Отдел распространения

(495) 775-16-76
e-mail: sales@dian.ru
Марина Трофимова
Юрий Царев
Сергей Лукин

Отдел рекламы

Юлия Суханова

Адрес редакции:

Москва, Ленинградское ш., д. 18, оф. 342
Для писем: 127015 Москва,
ул. Бутырская, д. 41/47 «ИД Скимен»
тел./факс: (495) 775-16-76
www.dian.ru
e-mail: editor@dian.ru
editordian@mail.ru

Издатель и учредитель ООО "ИД Скимен"

Отпечатано в ОАО Ордена Трудового
Красного Знамени "Чеховский
полиграфический комбинат"
142300 г. Чехов Московской области
Т/ф (501)443-92-17, т/ф (272)6-25-36
E-mail: marketing@chpk.ru

Тираж 5 200 экз.
Заказ № 2 70 4.

Журнал зарегистрирован
в Министерстве РФ по делам печати,
телерадиовещания и средств
массовых коммуникаций
Рег. № ПИ77-5262

Журнал включен в Реферативный
журнал и Базы данных ВИНТИ.
Сведения о журнале ежегодно
публикуются в международной
справочной системе по
периодическим и продолжающимся
изданиям «Ulrich's Periodicals
Directory».

Редакция не несет ответственности
за информацию, приведенную
в рекламных материалах

За содержание статьи
и ее оригинальность несет
ответственность автор

Полное или частичное
воспроизведение материалов
допускается только с разрешения
ООО "ИД Скимен"

Информацию о подписке
см. на последней странице журнала

Содержание

КОМПОНЕНТЫ

- | | |
|---|---|
| В. Дьяконов. Супервизоры питания и их применение | 2 |
| С. Бирюков. Микросхемы для активных корректоров коэффициента мощности NCP1653 и NCP1653A | 4 |
| В. Олейник. Низковольтные аналоговые интегральные микросхемы компании Maxim/Dallas | 7 |

ИСКУССТВО СХЕМОТЕХНИКИ

- | | |
|--|----|
| Б. Шевкопляс. Вероятностная синхронизация в телекоммуникационных системах | 10 |
| В. Дьяконов. Функциональные генераторы | 13 |
| В. Баранов. Работа с микроконтроллерами Atmel | 15 |

СОФТ

- | | |
|---|----|
| О. Петраков. Аналитические расчеты в электронике | 19 |
| О. Вальпа. Среда разработки VisualDSP++ | 22 |
| Р. Брыгин. Национальные особенности технологических сетей... | 24 |
| О. Вальпа. Автономный считыватель штрихкода | 26 |

ПРАКТИКА

- | | |
|--|----|
| Д. Онышко. Использование ЖКИ модулей | 28 |
| А. Маврычев. Схемотехника трехфазных многотарифных электронных счетчиков | 32 |
| А. Николаев. Макетная плата для микропроцессоров PIC16A62x | 35 |
| В. Василенко. Кабельный пробник со звуковой сигнализацией | 38 |
| Ю. Пахоменко. Преобразователь сигналов дифференциального емкостного датчика | 42 |
| Ю. Садиков. Простое радиопереговорное устройство на 27 МГц | 45 |

Новые книги. А. Фрунзе. «Микроконтроллеры фирмы Philips семейства x51» (с. 47).
Подписка-2007 (с. 48).

Супервизоры питания и их применение

В статье описаны микросхемы для микропроцессорных устройств — супервизоры (детекторы) напряжений, которые служат для четкого и точного определения момента снижения питающих напряжений до заданного уровня. Показано, что будучи простыми трехвыводными устройствами, эти микросхемы имеют довольно большие функциональные возможности, которые позволяют применять их и в других интересных и полезных устройствах — источниках электропитания, зарядных устройствах для аккумуляторов, импульсных устройствах и т. д. Описаны результаты исследования микросхем супервизоров и даны рекомендации по их применению.

МИКРОСХЕМА MC34064

Роль точного контроля напряжений питания непрерывно возрастает. Массовое применение устройств с батарейным (в частности аккумуляторным) питанием сделало непрерывный контроль напряжения питания обязательным для многих устройств, например, калькуляторов, карманных компьютеров, MP3-плееров, электронных часов и т. д. Разрядка аккумуляторов ниже определенного уровня губительно сказывается на сроке их работы, также как и перезарядка. Кроме того, многие электронные приборы, даже при сетевом питании, чувствительны к изменению напряжения источника. В первую очередь это относится к таким устройствам, как микропроцессоры, аналого-цифровые и цифро-аналоговые преобразователи, модули памяти и т. д.

Известно огромное число устройств контроля напряжения питания — от банального стрелочного вольтметра до сложных интеллектуальных блоков зарядки аккумуляторов. Нередко точность контроля и температурная стабильность порогов у многих таких устройств оказывалась явно низкой, а их повышение вело к неоправданному усложнению узлов контроля и увеличению потребляемой ими мощ-

ности. Учитывая эту ситуацию, ряд крупных фирм микроэлектронной промышленности приступил к серийному производству специальных микросхем супервизоров напряжения.

Одной из наиболее распространенных микросхем супервизоров напряжения является MC34064/33064, разработанная фирмой Motorola [1]. Она выпускается также фирмами Linfinity Microelectronics, On Semiconductor и др. Микросхема (рис. 1) содержит высокоточный температурно-компенсированный источник опорного напряжения, делитель напряжения R1R2, прецизионный гистерезисный компаратор ГИК с нагрузочным резистором R3 и выходной ключевой транзистор VT с диодом VD.

В микросхеме 21 транзистор и она выпускается во всех наиболее распространенных корпусах для транзисторов и микросхем малой степени интеграции, например в транзисторном корпусе TO226AA и в корпусах восьмивыводных микросхем 751 (SO-8) и 846A (Micro-8).

Основной задачей при разработке новых микросхем было их предельно простое применение по основному назначению (контроль за падением напряжения ниже заданного уровня) и наличие только трех выводов. Это несколько сужает возможные области применения таких массовых микросхем и требует внимательного изучения всех особенностей их работы, что и составляет цель данной статьи.

Прежде всего рассмотрим функциональную схему супервизора (рис. 1, а) более подробно. Ясно, что порог срабатывания задается напря-

жением опорного источника $U_{REF} = 1,2$ В и делителем напряжения R1R2. В технической документации задаются пороги срабатывания и гистерезис, они приведены в табл. 1.

Гистерезис необходим для исключения срабатывания компаратора от случайных быстрых изменений напряжения питания и шумов. Из-за существенной нелинейности входящих в супервизор элементов корректная работа устройства обеспечивается вблизи области срабатывания и далеко за ее пределами — примерно 1...9 В, хотя допустимый диапазон входных напряжений шире — 1...10 В. Максимальная рассеиваемая мощность 520...650 мВт в зависимости от корпуса. Максимальный протекающий в выход ток — 100 мА, диапазон рабочих температур 0...+70 °С для микросхем обычного применения и -40...+85 °С для микросхем в промышленном исполнении.

Статические характеристики

В руководстве по микросхеме MC34064/33064 [1] приведено детальное описание статических характеристик микросхем. Рассмотрим основные из них. Главной является передаточная характеристика, показанная на рис. 2.

Она описывает зависимость выходного напряжения от входного. Нетрудно заметить, что эта характеристика куда сложнее, чем это можно было бы предположить из идеализированного описания микросхемы. Лишь в средней части (в области входных напряжений примерно 1...9 В она соответствует описанию типовой роли прибора.

В области малых напряжений (менее 0,5 В), когда источник опорного напряжения перестает работать, передаточная характеристика имеет характерный выброс с линейным участком, на котором выходное напряжение равно входному, но уже при напряжении 0,5 В выходное напряжение падает практически до нуля и остается таким до увеличения входного напряжения до основного порога около 4,6 В. Далее, вплоть до напряжения чуть больше 9 В, выходное напряжение после скачка снова становится практически равным входному. А при входном напряжении более 9,2 В выходное напряжение скачком уменьшается практически до нуля. Причина подобного поведения не поясняется, но это означает, что микросхема может использоваться

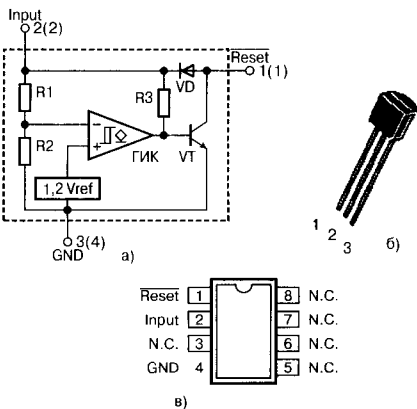


Таблица 1

Параметр	Мин.	Тип.	Макс.
Верхний порог, В	4,5	4,61	4,7
Нижний порог, В	4,5	4,59	4,7
Гистерезис, В	0,01	0,02	0,05

Рис. 1

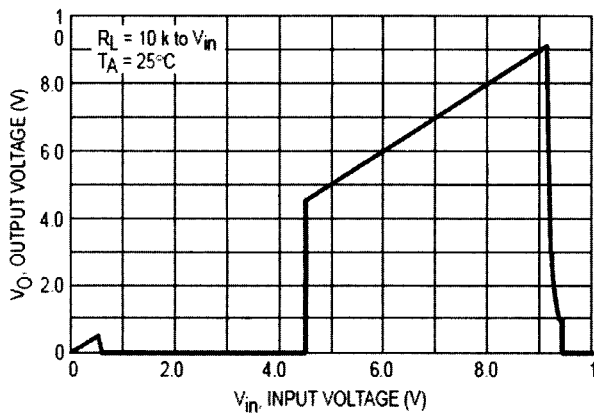


Рис. 2

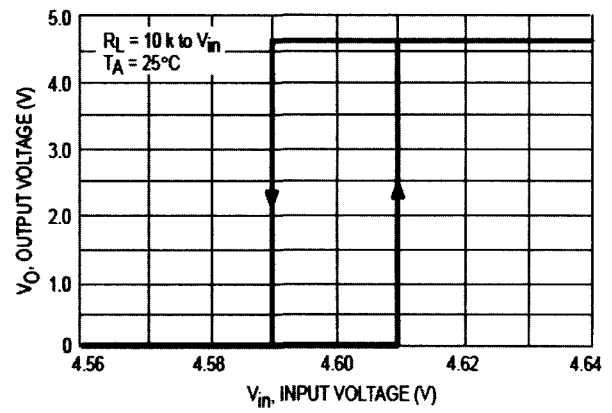


Рис. 3

для двухпорогового контроля. Неясно и то, является ли точное значение 2 для отношения напряжений порогов преднамеренным или случайным обстоятельством.

Исследования показывают, что в малой области главного порога (напряжение около 4,6 В) передаточная характеристика имеет гистерезис, как показано на рис. 3.

При снятии характеристик в статическом режиме ширина петли гистерезиса составляет 20 мВ. Наличие гистерезиса исключает дребезг при переключении, как при нарастании, так и уменьшении контролируемого напряжения, а малая величина гистерезиса делает двойственность

порога (при увеличении и уменьшении напряжения) практически незаметной.

Высокая стабильность порога — отличительное качество микросхем данной серии. На рис. 4 представлены температурные изменения верхнего и нижнего порогов в диапазоне температуры окружающей среды $T_A = -40...+85^\circ\text{C}$, разность порогов уменьшается при понижении температуры.

Интересный вид имеют зависимости входного тока от входного напряжения, представленные на рис. 5 для трех значений температуры окружающей среды, в целом они носят почти линейный характер с небольшим отклонением от линейности в области

малых значений, однако в области порогов (4,6 и 9,2 В) эти зависимости имеют характерные падающие и нестабильные участки, обусловленные регенеративным переключением устройства. В определенных условиях это может порождать паразитные или полезные релаксационные колебания с частотой 1 МГц и выше.

Когда транзистор VT микросхемы открыт, выходное напряжение определяется начальным участком вольтамперной характеристики насыщенного транзистора. На рис. 6 показана зависимость выходного напряжения от втекающего выходного тока для разных значений температуры окружающей среды T_A .

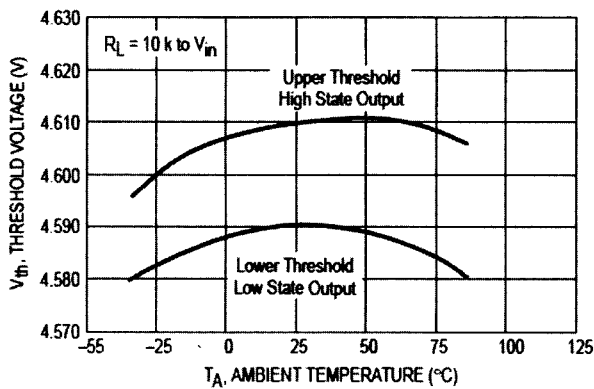


Рис. 4

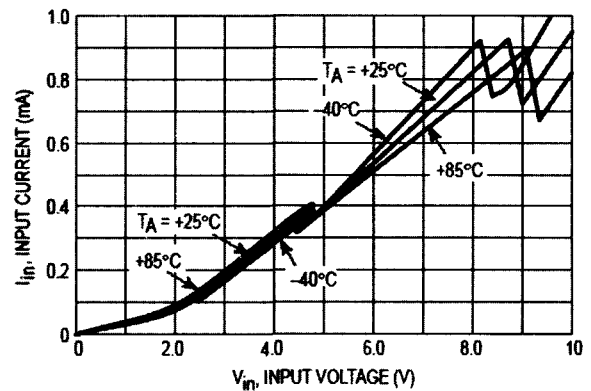


Рис. 5

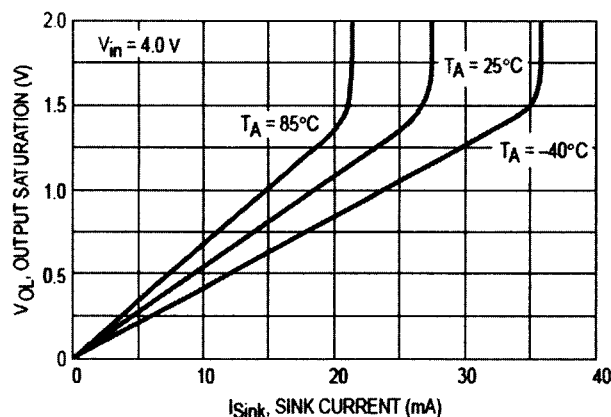


Рис. 6

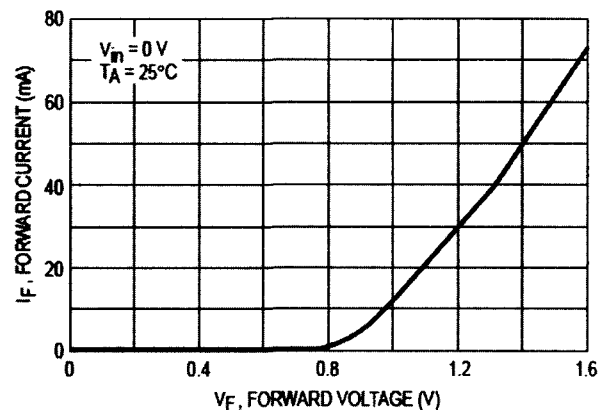


Рис. 7

Максимальное значение выходного тока (до выхода из насыщения) при нормальных условиях составляет около 25 мА, что достаточно для яркого свечения светодиодного индикатора или включения мало-мощного реле.

Для оценки свойств микросхем полезно также знать вольтамперную характеристику диода VD, она показана на рис. 7. Из нее видно, что диод выдерживает ток до 70 мА при прямом падении напряжения на нем 1,6 В.

Статические характеристики супервизоров питания неплохо описывают их применение при медленно изменяющихся входных напряжениях, что характерно для многих стандартных применений таких микросхем. Из них следует, что по основному назначению их можно использовать при высокостабильном пороге около 4,6 В. Использование второго порога 9,2 В в технической документации не оговаривается, но, как показала практика, вполне возможно (максимальное

напряжение питания с запасом взято равным 10 В).

ЛИТЕРАТУРА:

1. MC34064, MC33064. Undervoltage Sensing Circuit. Motorola, Inc., 1996.

Окончание следует

Владимир Дьяконов,
г. Смоленск

Продолжение. Начало — № 9/2006

Микросхемы для активных корректоров коэффициента мощности NCP1653 и NCP1653A

РЕЖИМ ПОВЫШАЮЩЕГО ПОВТОРИТЕЛЯ

Как уже упоминалось в предыдущей части статьи, корректор коэффициента мощности на микросхеме рассматриваемой серии может работать как в режиме стабильного выходного напряжения, так и при выходном напряжении, меняющемся пропорционально входному. Второй вариант называют «Режим повышающего повторителя», он реализуется при условии постоянства тока I_{CONTROL} . Работа АККМ в этом случае проиллюстрирована на рис. 10. Верхняя штриховая прямая соответствует стандартному режиму стабильного выходного напряжения, полусинусоиды U_{IN} — меняющемуся выпрямленному напряжению сети, нижняя ломаная P_{OUT} — скачку потребляемой мощности. Второй сверху график соответствует выходному

напряжению, оно увеличивается при увеличении входного и падает при увеличении потребляемой мощности.

Режим повторителя имеет некоторые преимущества перед стандартным — снижаются требования к дросселю и ключевому транзистору, за счет чего уменьшается его стоимость. В то же время работа преобразователя, подключенного к выходу корректора, усложняется из-за нестабильности его входного напряжения, что увеличивает уже стоимость преобразователя. В результате выбор, что сделать — АККМ с нестабильным выходным напряжением и преобразователь с широким диапазоном входного или стабилизированный корректор и преобразователь, работающий при фиксированном входном напряжении, должен решаться в каждом конкретном случае. Например, при установке в качестве второй ступени источника питания двухтактного преобразователя без стабилизации, что его существенно упрощает, естественным будет использование стабилизированного корректора.

СТАБИЛИЗАЦИЯ ВЫХОДНОГО НАПЯЖЕНИЯ

Рассмотрим работу цепи обратной связи, осуществляющей стабилиза-

цию выходного напряжения. Между выходом корректора и входом обратной связи FB микросхемы (вывод 1) включен резистор R_{FB} (рис. 4). Напряжение на выводе FB мало по сравнению с выходным, поэтому им обычно пренебрегают, и ток обратной связи, втекающий в этот вывод, можно рассчитать по следующей формуле:

$$I_{\text{FB}} = \frac{U_{\text{OUT}} - U_{\text{FB1}}}{R_{\text{FB}}} \approx \frac{U_{\text{OUT}}}{R_{\text{FB}}}. \quad (11)$$

В узле микросхемы, указанном на рис. 4 как Regulation Block, производится сравнение этого тока с опорным значением I_{REF} (типовое значение 200 мкА) и генерация выходного напряжения, управляющего через резистор 300 кОм узлом формирования выходных импульсов. Кроме того, ток I_{FB} используется в цепях защиты АККМ от превышения выходным напряжением допустимой величины и от его снижения ниже заданного значения.

Управление формированием выходных импульсов производится током I_{CONTROL} , идеализированная зависимость которого от тока I_{FB} иллюстрируется графиком на рис. 11.

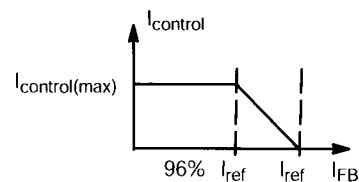


Рис. 11

Зависимость имеет три явно выраженных линейных участка. На первом из них при $I_{\text{FB}} < I_{\text{REF}} \times 96\%$ ток I_{CONTROL} постоянен и имеет макси-

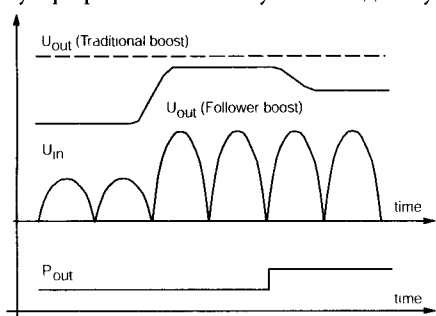


Рис. 10

мальную величину $I_{\text{CONTROL MAX}} = 100 \text{ мкА}$, корректор работает в режиме повторителя. Выходное напряжение АККМ можно рассчитать по следующей формуле:

$$U_{\text{OUT}} = \frac{2R_S R_{\text{VAC}} I_{\text{CONTROL MAX}} U_{\text{REF}} U_{\text{AC}}}{R_M R_{\text{CS}} P_{\text{OUT}}} \approx \frac{U_{\text{AC}}}{P_{\text{OUT}}} \quad (12)$$

Выходное напряжение не стабилизировано, оно прямо пропорционально входному и обратно пропорционально выходной мощности. Такая работа корректора проиллюстрирована на рис. 12.

На втором участке при выполнении условия $I_{\text{REF}} \times 96\% < I_{\text{FB}} < I_{\text{REF}}$ выходное напряжение корректора стабилизируется и находится в пределах 96...100 % от заданной величины. Ток

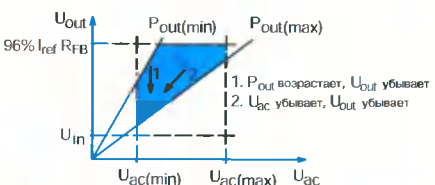


Рис. 12

I_{CONTROL} линейно спадает с увеличением I_{FB} и U_{OUT} :

$$I_{\text{CONTROL}} = \frac{I_{\text{CONTROL MAX}}}{0,04} \left(1 - \frac{I_{\text{FB}}}{I_{\text{REF}}} \right) = \frac{I_{\text{CONTROL MAX}}}{0,04} \left(1 - \frac{U_{\text{OUT}}}{R_{\text{FB}} I_{\text{REF}}} \right) \quad (13)$$

Учитывая, что для этого участка справедлива формула

$$U_{\text{OUT}} = \eta \frac{2R_S R_{\text{VAC}} I_{\text{CONTROL}} U_{\text{REF}} U_{\text{AC}}}{R_M R_{\text{CS}} P_{\text{OUT}}} \quad (14)$$

получаем:

$$U_{\text{OUT}} = \frac{U_{\text{AC}}}{\frac{R_M R_{\text{CS}}}{2R_S R_{\text{VAC}} U_{\text{REF}} I_{\text{CONTROL MAX}}} \times 0,04 \times \frac{1}{\eta \times \frac{P_{\text{OUT}}}{R_{\text{FB}} I_{\text{REF}}}}} \quad (15)$$

В соответствии с (15) выходное напряжение становится равным своей расчетной величине $I_{\text{REF}} R_{\text{FB}}$ лишь тогда, когда P_{OUT} близка к нулю, это максимально возможная величина при работе в этой области изменения

I_{FB} . При увеличении потребляемой мощности или уменьшении входного напряжения выходное напряжение уменьшается, но остается не менее 96 % от своего расчетного значения. Зависимости, иллюстрирующие работу корректора в этом режиме, показаны на рис. 13.

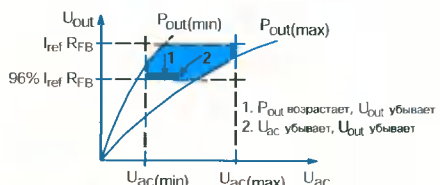


Рис. 13

Если ток обратной связи I_{FB} превысит I_{REF} , ток I_{CONTROL} становится равным нулю и микросхема уменьшает до нуля длительность импульсов, открывающих ключевой транзистор. Выходное напряжение корректора начинает снижаться. Когда оно достигнет номинального значения, работа возобновляется в режиме стабилизации.

ЗАЩИТА ОТ ПРЕВЫШЕНИЯ ВЫХОДНОГО НАПЯЖЕНИЯ

Если выходное напряжение превысит 107 % от требуемой величины

Макро Тим

ООО "МАКРО ТИМ" ПРЕДСТАВЛЯЕТ КОРРЕКТОРЫ КОЭФФИЦИЕНТА МОЩНОСТИ ON SEMICONDUCTOR:

- Различные решения для режима непрерывного тока и режима критической проводимости
- Гибкость, компактность, небольшая стоимость в режиме повышающего преобразователя (Boost)
- Фиксированная частота и возможность синхронизации
- Уменьшение электромагнитных помех
- Низкое потребление тока в дежурном режиме
- Интегрированная в кристалл защита от перегрузки по току, от превышения и понижения напряжения, от обрыва обратной связи

Схема корректора коэффициента мощности на 300 Вт

Наименование	Макс. мощн., Вт	Топология	Режим	Рабочий режим	Цель запуска	OVP*	UVP*	Ограничение тока	Ограничение мощности	Обнаружение броска тока	Выключение	Корпус
NCP1651	250	Flyback	CCM/DCM	Средний ток	●	●	●	●	●	●	●	SO-16
NCP1603	250	Flyback	DCM	Ток	●	●	●	●	●	●	●	SO-16
MC33260	300	Boost	CC	Напряжение	●	●	●	●	●	●	●	PDIP-8, SO-8
MC33262	200	Boost	CC	Ток	●	●	●	●	●	●	●	PDIP-8, SO-8
MC33368	300	Boost	CC	Ток	●	●	●	●	●	●	●	PDIP-16, SO-16
NCP1601	400	Boost	DCM	Напряжение	●	●	●	●	●	●	●	PDIP-8, SO-8
NCP1653	1000	Boost	CCM	Средний ток	●	●	●	●	●	●	●	PDIP-8, SO-8
NCP1650	5000	Boost	CCM/DCM	Средний ток	●	●	●	●	●	●	●	SO-16

* Режим непрерывного тока
* Режим прерывистого тока
* Режим критической проводимости
* Защита от превышения напряжения
* Защита от понижения напряжения

ООО «Макро Тим», 111141, Россия, Москва, Зеленый пр., 2/19
Тел.: (095) 306 0026/4721/4789, факс: (095) 306 0283, e-mail: sales@macrotsam.ru, http://www.macroteam.ru

ны, срабатывает узел Overvoltage Protection (рис. 4) и выдача выходных импульсов блокируется по дополнительной цепи независимо от работы других узлов.

При номинальном выходном напряжении 390...400 В максимальный порог срабатывания микросхемы от превышения выходного напряжения допустимого значения с учетом разброса параметров составляет 444 В, поэтому рабочее напряжение выходного конденсатора C_{BULK} корректора следует выбирать не менее 450 В.

ЗАЩИТА ОТ ПАДЕНИЯ ВЫХОДНОГО НАПРЯЖЕНИЯ

Если ток обратной связи I_{FB} уменьшается до значения менее чем 8 % от I_{REF} , т. е. выходное напряжение падает до 8 % от номинального значения, срабатывает узел Shutdown/UPV (рис. 4) и микросхема выключается, ее ток потребления в этом состоянии уменьшается до 50 мкА. Устройство вновь начинает работать, когда I_{FB} возрастет до 12 % от I_{REF} (рис. 14).

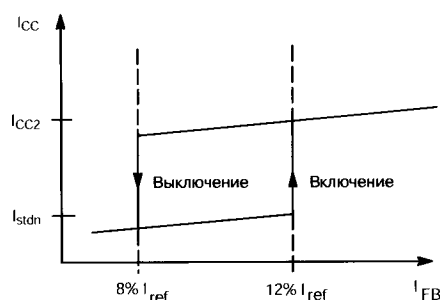


Рис. 14

В нормальном режиме работы повышающего АККМ выходное напряжение всегда выше входного и с большим запасом превышает указанные 8 и 12 %. Защита от слишком низкого входного напряжения может сработать при замыкании выхода корректора, при обрыве цепи вывода FB или при принудительном выключении корректора и переводе его в состояние с низким потреблением путем замыкания вывода 1 на общий провод (вывод 6).

МЯГКИЙ СТАРТ

Широтноимпульсный модулятор микросхемы управляется током $I_{CONTROL}$. В цепи формирования этого тока установлен внешний конденсатор $C_{CONTROL}$, поэтому при включении корректора напряжение на нем плавно нарастает, что обеспечивает плавное увеличение длительности выходных импульсов и мягкий старт.

ЗАЩИТА ОТ ПЕРЕГРУЗКИ ПО ТОКУ

Как уже упоминалось в первой части статьи, резистор R_S в минусовой цепи выпрямленного напряжения сети является токоизмерительным — через него протекает ток дросселя L , при открытом ключевом транзисторе VT1 — его ток, при закрытом VT1 — ток диода VD1. Падение напряжения на этом резисторе имеет отрицательную по отношению к входу CS (вывод 4) полярность. Цепи этого входа построены так, что поддерживают на нем нулевое напряжение (рис. 15). В результате, вытекающий ток этого вывода пропорционален току через резистор R_{CS} , а коэффициент пропорциональности определяется соотношением сопротивлений резисторов R_{CS} и R_S , что зафиксировано формулой (10)).

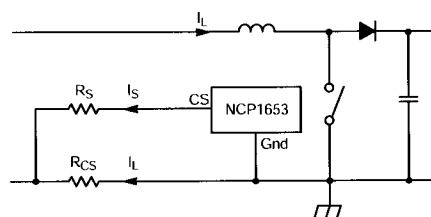


Рис. 15

Через «токовое зеркало» сигнал, пропорциональный току резистора R_{CS} , подается на узел защиты от перегрузки по току Overcurrent Protection и умножитель, обеспечивающий соответствующее функционирование микросхемы.

Узел защиты от перегрузки по току срабатывает, когда ток I_S начинает превышать порог, типовое значение которого составляет 200 мкА (на рис. 4 в предыдущей части статьи выпал знак μ). Используя соотношение (10), несложно получить пороговое значение тока ограничения $I_{L OCP}$:

$$I_{L OCP} = \frac{R_S I_{SOCP}}{R_{CS}} = \frac{R_S}{R_{CS}} \times 200 \text{ мкА.} \quad (16)$$

Если узел защиты от перегрузки срабатывает, выходной импульс микросхемы сразу же обрывается. Нормальная работа корректора возобновляется при снижении тока ниже порога срабатывания.

ПОДАЧА ВХОДНОГО НАПРЯЖЕНИЯ

Ток, пропорциональный среднему значению входного напряжения, подается через вывод 3 (рис. 16). «Токовое зеркало» подключено к этому

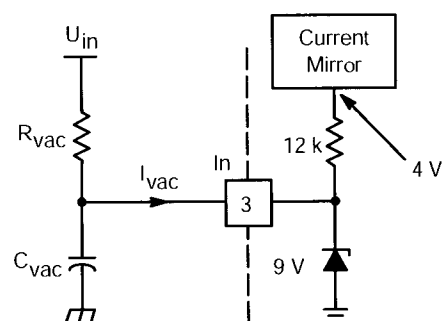


Рис. 16

выводу через резистор сопротивлением 12 кОм, кроме того, к этому входу подключен стабилитрон на 9 В. Важно, чтобы при нормальной работе этот стабилитрон не открывался. Если потребовать, чтобы это условие соблюдалось при входном пиковом напряжении 400 В, можно получить следующее ограничение на сопротивление R_{VAC} снизу:

$$\frac{R_{VAC}}{400 \text{ В} - 9 \text{ В}} > \frac{12 \text{ кОм}}{9 \text{ В} - 4 \text{ В}} \quad (17)$$

$$R_{VAC} > 938 \text{ кОм.}$$

ЗАЩИТА ОТ ПРЕВЫШЕНИЯ ВХОДНОЙ МОЩНОСТИ

Ток I_{VAC} через вход In (вывод 3) пропорционален среднему значению выпрямленного напряжения сети (в предыдущей части неправильно указано — мгновенному), а ток I_S входа CS — потребляемому корректором току. Они через «токовые зеркала» подаются на перемножитель (рис. 4), выходной сигнал которого делится на ток $I_{CONTROL}$ и используется для управления длительностью выходных импульсов, а также подается на узел защиты от превышения входной мощности корректора Overpower Limitation. В этом узле произведение токов I_{VAC} и I_S сравнивается с пороговым значением, равным 3 нА². При превышении указанного порога замыкается ключ на выходе узла формирования тока $I_{CONTROL}$ и этот ток становится равным нулю. Это приводит к уменьшению длительности подаваемых на ключевой транзистор импульсов и ограничению потребляемой корректором мощности.

Продолжение следует

Сергей Бирюков,
г. Москва

Продолжение. Начало — №9/2006

Низковольтные аналоговые интегральные микросхемы компании Maxim/Dallas

ПРЕОБРАЗОВАТЕЛИ ДАННЫХ

АЦП

Низкая потребляемая мощность и высокая скорость обработки сигнала являются решающими факторами для АЦП, работающих в носимой аппаратуре. Обычно указанные требования являются взаимоисключающими, и сохранить их баланс довольно сложно. Компания Maxim/Dallas нашла компромисс, представив на рынке АЦП, имеющие умеренное потребление тока при обработке данных и крайне низкое потребление в выключенном режиме. Результат — ощутимое снижение мощности в узлах, где АЦП не работает постоянно.

Например, последовательный АЦП MAX1115 обеспечивает скорость преобразования сигнала 100 MSPS, имеет внутренний ИОН 2,048 В, работает от источника питания +3 В и потребляет всего 175 мкА, при этом в выключенном режиме ток снижается до 1 мкА. MAX1115 обеспечивает значительное снижение потребляемой мощности в системах, где потребляемая нагрузкой мощность является критичным фактором, а АЦП работает в периодическом режиме (рис. 8).

Одним из таких применений служит измерение уровня принятого сигнала в

сотовых телефонах (индикатор уровня принятого сигнала — RSSI, Received Signal Strength Indicator). Использование АЦП MAX1115 позволяет выполнять оцифровку сигнала с частотой 2 кГц при токе потребления 2 мкА от источника питания 3 В. Общая нерегулируемая ошибка (сумма смещения, интегральной нелинейности и ошибок усиления) составляет менее одного наименее значимого разряда, а гарантированное отношение сигнал/(шум+искажения) (SINAD — Signal-to-Noise And Distortion) не превышает 48 дБ.

ЦАП

Современные ЦАП позволяют выполнять преобразование цифровых сигналов в аналоговые даже в системах с низким напряжением питания. Рассматриваемые ЦАП проектировались из расчета их использования в портативных (носимых) системах, поэтому отличаются исключительно низким током потребления и малой занимаемой площадью. Бюджетный ЦАП MAX5811 является 10-разрядным прибором с выходом по напряжению. Ток потребления указанной ИМС в рабочем режиме составляет 170 мкА, а в выключенном (shutdown mode) снижается до 1 мкА, что идеально подходит для носимых систем с батарейным питанием. Передача данных выполнена с использованием последовательного интерфейса, поэтому ЦАП уместается в корпусе SOT23.

ЦАП MAX5811 работает от однополярного источника 2,7 В, обеспечивая размах выходного напряжения, равный напряжению питания. Указанный ЦАП идеально подходит для недорогих устройств, например, используемых для подстройки напряжения смещения, задания смещения при регулировке источников тока или напряжения и т. п.

ОУ и датчики тока

При питании ОУ от низковольтного источника снижется отношение

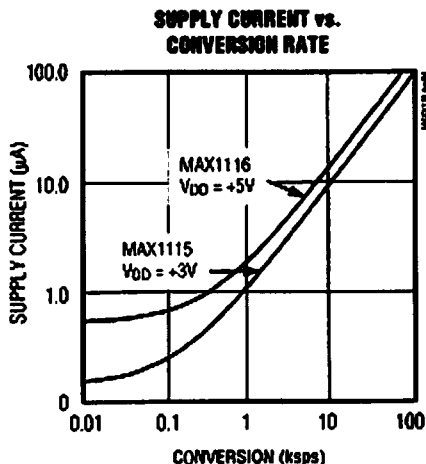
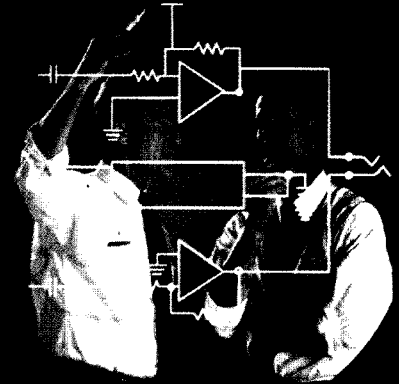


Рис. 8

Стратегические инженерные решения



Продукты компании IC

Аналоговые ИМС

Цифровые ИМС

Системы

Системы

Системы

Системы

Системы

Системы

Системы

Системы

Системы

Системы

Системы

Системы

Системы

Системы

Системы

Системы

Системы

Системы

Системы

Системы

Системы

Системы

Системы

Системы

Связь и телеком. мультимедиа

Энергетика

Транспорт

Торговля

Доступ и защита информации

Помощь клиентам

Медицина

Информационные технологии

RAINBOW TECHNOLOGIES

www.mcs.ru
www.ibutton.ru
www.rainbow.by

Москва (495) 797-8993 info@rainbow.msk.ru
Санкт-Петербург (812) 324 0902 spb@rainbow.msk.ru
Минск (+37517) 249-8273 chip@rainbow.by
Киев (+38044) 520-0479 chip@rainbow.com.ua
Екатеринбург (343) 261-6446 ural@rainbow.msk.ru
Новосибирск (383) 211 9087 nsk@rainbow.msk.ru

DALLAS SEMICONDUCTOR

MAXIM

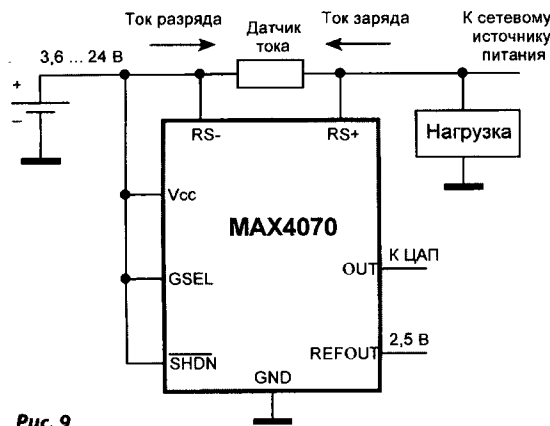


Рис. 9

Таблица 3

Микросхема	Диапазон питающих напряжений, В	Ток потребления мкА (тип./макс.)	Диапазон входных напряжений, В	Размах выходного напряжения, В (тип.)
MAX4036/ MAX4038	1,4...3,6	0,8/1,2	USS...(VDD-0,4)	(USS+0,002)... (UDD-0,002)
MAX4069	2,7...24	100/250	1,35...24	(USS+0,1) ... (UDD-0,09)
MAX4070	3,6...24	100/250	1,35...24	(USS+0,1)... (UDD-0,09)

сигнал/шум (SNR — Signal-to-Noise Ratio) за счет снижения размаха выходного сигнала. Многие низковольтные ОУ обеспечивают размах выходного сигнала от одной шины питания до другой как средство сохранения отношения сигнал/шум. Некоторые ОУ работают при уровнях входного сигнала также от одной шины питания до другой.

При работе ОУ при пониженном напряжении питания не только снижается диапазон напряжений сигнала, но и повышается уровень шумов, что ухудшает соотношение сигнал/шум. Низковольтные усилители проектируются из расчета низкого тока потребления, что приводит к росту уровня шумов усилителя. Кроме того, резисторы обратной связи имеют более высокие сопротивления, что также усиливает уровень шумов.

Следует учитывать и тот факт, что из-за емкостной связи между проводниками печатной платы точки усилителя, имеющие высокое входное сопротивление, значительно проще воспринимают шумы, генерируемые высокоскоростными магистралями передачи данных. Для снижения этого эффекта высокоомные цепи следует выполнять как можно более короткими и разносить в пространстве подальше от цифровых линий передачи высокоскоростных сигналов.

Примечательно, что рассматриваемые низковольтные ОУ имеют малый ток потребления, низкое напряжение смещения и высокую скорость — параметры, которые можно отнести к категории взаимоисключающих. Например, серия ОУ MAX4236A, минимальное рабочее напряжение которой составляет +3 В, имеет ширину полосы усиления 1,7 МГц, напряжение смещения 20 мкВ и ток потребления 350 мкА. Эти параметры делают ОУ серии MAX4236A незаменимыми при проектировании инструментальных

усилителей и низковольтных систем с батарейным питанием.

Семейство ОУ компании Maxim/ Dallas содержат в своем составе усилители с датчиками тока, например, MAX4069, работающий от напряжения +2,7 В (рис. 9).

Подобные токовые усилители сводят к минимуму проблемы соединения с общим проводом, выполняя измерение тока в цепи положительной шины источника питания с помощью специальных измерительных резисторов. Данный токовый измеритель поставляется в малогабаритных 8- или 10-выводных корпусах μ MAX.

В портативных системах, где каждый микроампер, что называется, «на счету», некоторые микроомные низковольтные ОУ могут обеспечить выдающиеся показатели. Так, ОУ MAX4036/MAX4038 и MAX4474, имеющие минимальные напряжения источника питания +1,4 В и +1,8 В соответственно, обеспечивают исключительно низкий ток потребления, не превышающий 1,2 мкА. Размах выходного напряжения — от одной шины питания до другой, при этом входные напряжения могут опускаться ниже минусовой шины питания.

В общем случае, при питании ОУ от низковольтного источника диапазон входных напряжений и размах выходного напряжения сужается. При проектировании низковольтных систем это обстоятельство следует учитывать. В табл. 3 представлены выборочные параметры ОУ, описанных выше.

Компараторы

Подобно низковольтным ОУ, среди низковольтных компараторов имеются высокоскоростные и микроомные изделия, а также компараторы с низким напряжением смещения. Например, компаратор MAX9100 является микроомным, работающим от однополярного источника питания напря-

жением 1...5,5 В, максимальный ток которого составляет 12 мкА. Время задержки срабатывания этой ИМС составляет 3,7 мс, смещение равно 2 мВ, а размах выходного напряжения на 0,3 В менее напряжения питания.

Некоторые задачи, например, мониторинг выходного напряжения источника питания, требуют применения сверхмикроомных ИМС. Компаратор MAX9017A работает от однополярного источника питания 1,8...5,5 В, при этом максимальный ток потребления составляет всего 1,2 мкА (типовое значение), в своем составе компаратор имеет ИОН.

Мониторы питания

Все микропроцессорные системы обычно содержат мониторы питания, обеспечивающие защиту микропроцессора от нештатного функционирования из-за изменения напряжения питания. Простейший монитор представляет собой устройство, выполняющее функцию сброса микропроцессора при подаче на систему напряжения питания. Многие мониторы питания оснащены дополнительными функциями, например, управлением резервной батареей питания, защитой от записи в память, а также использованием сторожевых (watchdog) таймеров мониторинга выполнения программы.

Резервные батареи обеспечивают бесперебойность подачи питания на устройства, для которых подоб-

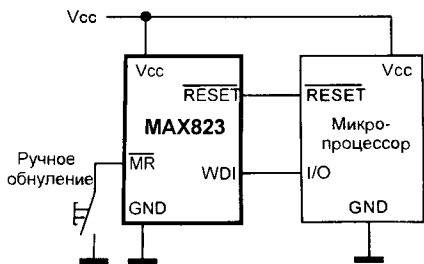


Рис. 10

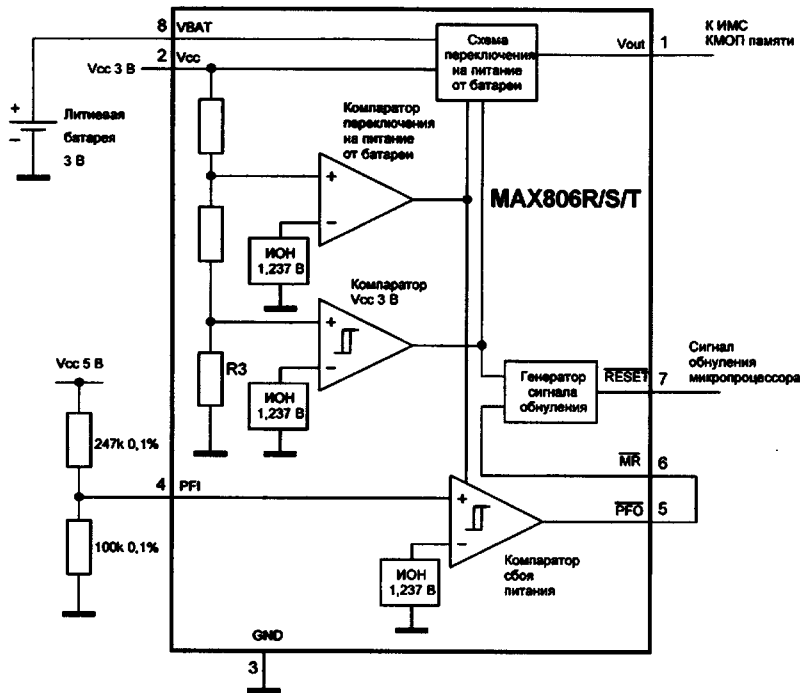


Рис. 11

ная процедура является критичной (например, элементы КМОП памяти, часы реального времени и т. п.). Выполняя постоянное слежение за основным напряжением питания, монитор определяет момент переключения на питание микропроцессора от резервного (автономного) источника. Однако работа систем при пониженном напряжении накладывает определенные ограничения, с которыми не сталкиваются разработчики систем, работающих с изделиями, которые питаются от источников 5 В.

В устройствах с напряжением питания 5 В выполняется простое сравнение U_{CC} с напряжением резервной батареи и переключение осуществляется при снижении напряжения U_{CC} ниже определенного уровня. Но этот подход может привести к ложным переключениям в низковольтных системах — так, например, мониторы питания литиевых батарей определяют батарею как свежую при напряжении на ней 3,6 В, что выше уровня 3 В для напряжения питания в системах с питанием 3,3 В. Мониторы питания Maxim/Dallas позволяют избежать этой проблемы, допуская превышение напряжения резервной батареи над U_{CC} и инициализируя процедуру переключения питания микропроцессора только при снижении значения U_{CC} ниже заданного предела.

ИМС MAX823/MAX824 представляют собой мониторы питания и сторожевые таймеры. Они поставляются в корпусах SC70 и SOT23 (рис. 10).



Рис. 12

Мониторы MAX806R/S/T имеют встроенный узел переключения на резервный источник и выполняют мониторинг напряжений 3 и 5 В для систем с двумя напряжениями питания (рис. 11). Компаратор главного напряжения U_{CC} выполняет мониторинг напряжения 3 В, а компаратор снижения питания PFI (power fail input) — 5 В.

Внутренний узел монитора обеспечивает выдачу сигнала сброса при выходе напряжения 3 В за допустимые пределы. Для напряжения 5 В порог 4,527...4,726 В задается прецизионными резисторами с допуском 0,1 %. Когда напряжение источника питания 5 В опускается ниже указанного порога, компаратор PFO (power fail output) замыкает на общий провод вход сброса MR (manual reset). Та-

ким образом, при выходе любого из источников питания за допустимые границы монитор осуществляет процедуру сброса микропроцессора.

Некоторые из низковольтных мониторов питания производства Maxim/Dallas осуществляют защиту ИМС памяти выдачей сигнала CE (chip enable). Сигнал CE защищает память путем блокирования операций считывания и записи при выходе напряжения питания за установленные границы. MAX792 и MAX820 выполняют выдачу сигнала CE с задержкой 10 нс. Малые времена задержки позволяют работать с более дешевыми (менее быстродействующими) ИМС памяти, так как время задержки распространения сигнала CE занимает меньшую часть цикла обращения к ней. Эти мониторы питания также имеют вывод MR, обеспечивают выполнение процедур сброса при включении, оповещения о

снижении напряжения питания, работают в режиме сторожевого таймера.

Низковольтная ИМС MAX6741 осуществляет мониторинг двух напряжений питания, используемых в системе, при этом ток собственного потребления составляет всего 6 мкА (рис. 12). Все ИМС этого семейства имеют либо двухтактные выходы, либо выходы с открытым стоком, изготавливаются в малогабаритном корпусе SC70 размерами 2×2,1 мм и позволяют выполнять мониторинг напряжений вплоть до 0,488 В.

Окончание следует

Перевод
Владимира Олейника
г. Королев Московской обл.

Вероятностная синхронизация в телекоммуникационных системах

ИЗМЕРЕНИЕ УРОВНЯ ПОТЕРЬ ПАКЕТОВ, ПЕРЕДАВАЕМЫХ МЕЖДУ ДВУМЯ УЗЛАМИ СЕТИ

Прежде всего, отметим, что скорость передачи пакетов — переменная величина, а скорость передачи битов по линии — постоянная. Скорость передачи битов выбирается максимально возможной для данной линии и не изменяется в процессе ее эксплуатации. Она не зависит от характера передаваемой информации. В частности, в паузах между пакетами по линии передаются некоторые заранее оговоренные протоколом обмена последовательности битов. Средняя скорость передачи одинаковых по длине пакетов максимальна, если они следуют друг за другом непрерывно, без пауз. С увеличением средних длительностей пауз скорость передачи пакетов снижается. В частности, эта скорость на некотором отрезке времени может быть нулевой. Длина пакета может быть постоянной или переменной.

Потери пакетов при их передаче по каналу связи могут быть обусловлены двумя основными факторами. Первый — искажения принимаемого сигнала из-за помех, отражений, асимметрии линии и т. п. Второй фактор — кратковременное или длительное ограничение способности многоканального приемника обрабатывать прибывающий из N-го канала поток пакетов, например, в связи с получением более приоритетных потоков из других каналов. В этом случае буферная память N-го канала приемника переполняется или блокируется и вновь поступающие пакеты теряются.

Линии связи обычно обеспечивают высокое качество передачи сигналов, поэтому потери пакетов обусловлены, в основном, вторым фактором — изменяющейся пропускной способностью приемника. Именно о таких потерях и пойдет речь.

Передачик должен регулировать скорость выдачи пакетов так, что-

бы уровень их потерь (если потери имеются) находился около некоторой достаточно низкой отметки. Если уровень лежит ниже этой отметки, это означает, что передатчик может увеличить скорость передачи пакетов (если максимальная скорость еще не достигнута) и, наоборот — при обнаружении заметных потерь передатчик вынужден снизить скорость. Таким образом, передача пакетов осуществляется на скорости, близкой или равной максимально возможной. Альтернативой увеличению скорости выдачи пакетов может служить увеличение их длины.

Как измерить уровень потерь пакетов и снизить его до минимума? Чтобы ответить на эти вопросы, следует различать два типа систем передачи данных.

Первый тип характеризуется тем, что каждый удачно переданный пакет сопровождается ответным пакетом, в котором содержится подтверждение его получения. Альтернативный вариант оповещения предусматривает передачу ответных пакетов только в тех случаях, когда соответствующие пакеты данных не приняты приемником в течение допустимого времени ожидания. Непринятые по тем или иным причинам пакеты передаются повторно один или несколько раз в зависимости от результата передачи.

В таких системах адаптация передатчика к пропускной способности приемника может выполняться так. Если к началу сеанса передачи потока пакетов максимально допустимая скорость их следования неизвестна передатчику, он использует процедуру «мягкого старта» [6]. Она заключается в том, что на первом шаге скорость передачи пакетов выбирается весьма низкой (паузы между пакетами заполняются флаговыми кодами [4]). На втором шаге скорость увеличивается вдвое (паузы между пакетами соответственно уменьшаются) и проверяется уровень потерь

пакетов. О наличии этих потерь передатчик судит по содержанию ответных пакетов, поступающих по обратному каналу со стороны приемника. Если уровень потерь по-прежнему низкий, на третьем шаге скорость вновь увеличивается вдвое и т. д., вплоть до обнаружения заметных потерь пакетов.

После обнаружения таких потерь достигнутая передатчиком скорость уменьшается на 25 % и вновь проводится очередная оценка их уровня. На следующем шаге на основании этой оценки скорость увеличивается или уменьшается на 12,5 % и т. д. В результате последовательного приближения к границе резкого возрастания уровня потерь пакетов передатчик адаптируется к пропускной способности приемника.

Полученные в процессе регулирования статистические данные о зависимости уровня потерь пакетов от скорости их передачи (или длины) могут использоваться при очередном старте сеанса передачи потока пакетов для уменьшения времени сходимости процесса адаптации.

Второй тип предусматривает «безответную» передачу пакетов в том смысле, что в таких системах нет подтверждений получения или неполучения пакетов приемником. Если, к тому же, в системе не предусмотрен мониторинг уровня потерь пакетов, то передатчик лишен возможности оценивать правильность выбора скорости их отправки из-за отсутствия обратной связи.

Чтобы в таких системах оперативно отслеживать изменения пропускной способности приемника, применяют измерительные пакеты [2]. Они периодически посылаются передатчиком в приемник и затем возвращаются к нему по обратному каналу. По имеющимся в них показаниям счетчиков переданных и принятых пакетов передатчик вычисляет уровень потерь. Недостатком такого способа мониторинга уровня потерь является необходимость использования громоздких измерительных пакетов, «засоряющих» трафик.

Предлагаемая система передачи данных (рис. 4) содержит встроенный измеритель уровня потерь пакетов, передаваемых из устройства А в устройство В.

В данном случае измерительные пакеты не нужны, взаимодействие передатчика с приемником, как и в ранее рассмотренном решении, осу-

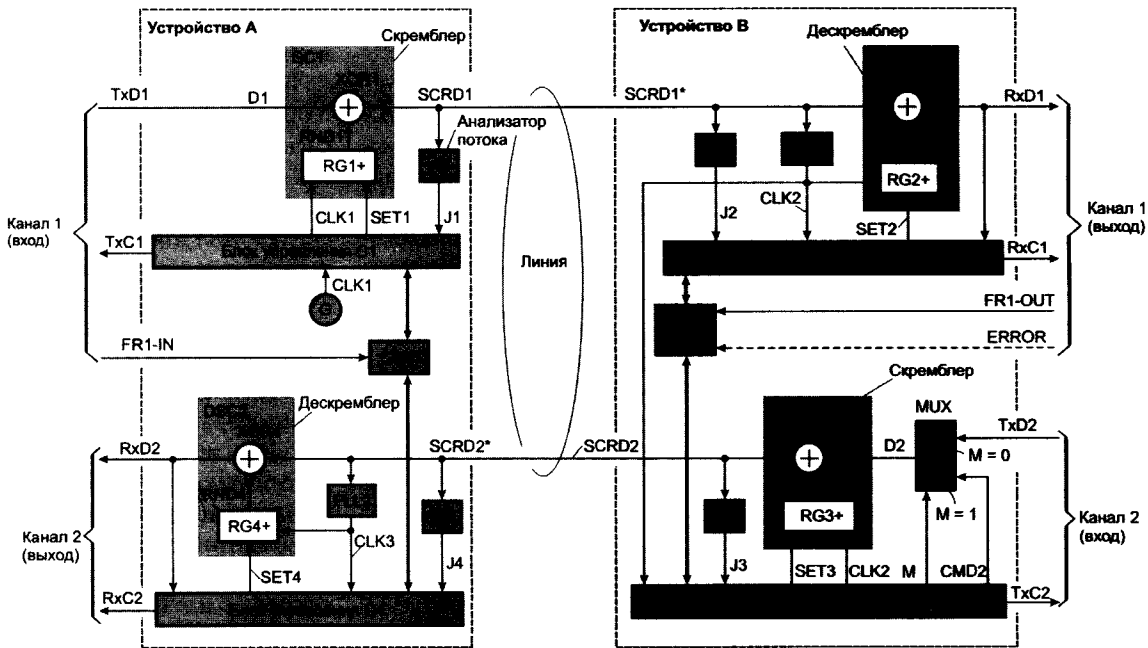


Рис. 4

ществляется на основе вероятностной синхронизации. Эта система по структуре близка к приведенной на рис. 1, поэтому описание общих для обеих систем составных частей далее опущено.

Система на рис. 4 предназначена для измерения уровня потерь пакетов при их передаче из устройства А в устройство В. На рисунке показаны только те элементы, которые имеют прямое отношение к такой передаче, поэтому система выглядит асимметричной. Схему можно привести к симметричному виду, дополнив ее соответствующими элементами для измерения уровня потерь пакетов при их передаче из устройства В в устройство А.

Передаваемые в обе стороны данные упакованы в пакеты (их также называют фреймами, кадрами, ячейками [4]). Формирование и распознавание пакетов осуществляется внешними устройствами (эти устройства на рисунке не показаны). Система передачи данных не изменяет структуру передаваемых пакетов, а лишь ведет их подсчет при пересылке из устройства А в устройство В.

Процессор CPU1 содержит внутренний счетчик переданных в линию пакетов. Этот счетчик устанавливается в нуль сигналом J1 и увеличивает текущее содержимое на единицу всякий раз при получении признака FR1-IN наличия очередного пакета, поступившего на вход канала 1. В качестве счетчика может использоваться ячейка памяти или регистр общего назначения процессора.

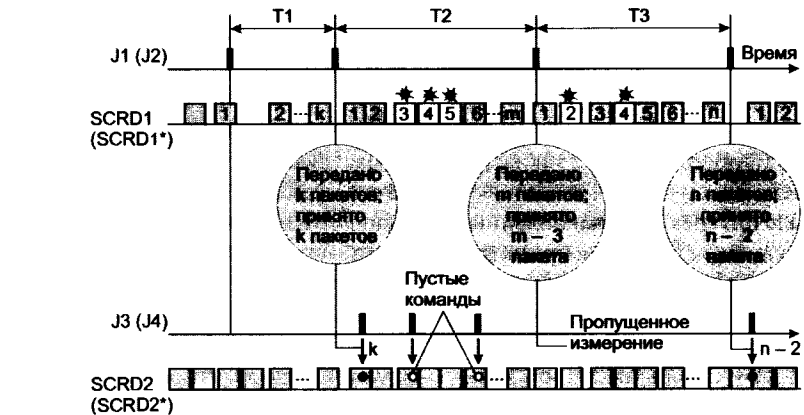


Рис. 5

Процессор CPU2 содержит внутренний счетчик пакетов, успешно принятых с выхода канала 1 внешним устройством. Этот счетчик устанавливается в нуль сигналом J2 и увеличивает содержимое на единицу всякий раз при получении признака FR1-OUT наличия пакета, принятого внешним устройством. Если внешнее устройство отвергло пакет из-за имеющихся в нем ошибок или (что важнее) из-за снижения своей пропускной способности, то пакет теряется и признак FR1-OUT не формируется.

В отсутствие потерь содержимое счетчиков переданных и принятых внешними устройствами пакетов совпадает с точностью до некоторой задержки, которой пренебрегаем для упрощения описания (задержку можно вычислить при проектировании конкретной системы). Потери приводят к тому, что число переданных пакетов превышает число принятых.

Доля потерянных пакетов вычисляется процессором CPU1 в каждом

цикле измерения после получения им сведений о содержимом счетчика принятых пакетов из процессора CPU2. Полученные в отдельных циклах результаты усредняются.

Рассмотрим процесс измерения потерь пакетов более подробно (рис. 5).

Циклы измерения задаются промежутками времени T1, T2, T3 и т. д. между импульсами J1 (J2) вероятностной синхронизации в канале прямой передачи данных. Импульсы J3 (J4) вероятностной синхронизации в канале обратной передачи данных задают моменты передачи команд из устройства В в устройство А. В некоторых из этих команд содержится информация о числе пакетов, принятых и зарегистрированных процессором CPU2 в ближайшем предыдущем цикле измерения.

Как отмечалось ранее, импульсы вероятностной синхронизации возникают спонтанно с некоторой средней частотой. Эта частота выбирается на этапе проектирования системы и

зависит от скорости передачи битов по линии и разрядности кодов, которые отыскиваются анализаторами A1—A4 в скремблированных потоках битов.

Средний период между импульсами J1 (J2) задает среднее число пакетов, передаваемых через систему в каждом цикле измерения. Это число может быть равно, например, 500. Отметим, что в отличие от существующих систем, среди этих пакетов нет служебных, т. е. тех, которые вводились бы специально для обслуживания процесса измерения потерь.

В начале интервала T1 накопленный в счетчике переданных пакетов код запоминается, после чего счетчик устанавливается в нуль. По окончании прохождения текущего пакета к содержимому этого счетчика прибавляется единица. После прохождения через устройство А серии из k пакетов в счетчике накапливается код k . Так как в данном случае потерь пакетов нет, аналогичный счетчик принятых пакетов в процессоре CPU2 также накапливает код k .

В начале интервала T2 накопленные в счетчиках переданных и принятых пакетов коды k запоминаются, после чего эти счетчики устанавливаются в нуль и начинают накопление новых кодов. Первый импульс J3, зарегистрированный после окончания предыдущего интервала измерения T1, вызывает вставку группы командных битов в поток данных SCRD2 (SCRD2*). В этой группе битов, помеченной на рис. 5 крайней левой черной точкой, содержится код k — ранее сохраненный результат измерения числа полученных в интервале T1 пакетов.

Процессор CPU1 считывает принятый в составе команды код k и сверяет его с ранее сохраненным результатом подсчета переданных пакетов. В данном случае число переданных пакетов в точности совпадает с числом принятых. Полученный результат усредняется с предыдущими в некотором «скользящем окне» длительностью, например, 10 с.

В примере, показанном на рис. 5, до момента окончания текущего цикла измерения T2 спонтанно формируются еще два импульса J3, которые в соответствии с программой, выполняемой процессором CPU2, порождают две пустые команды, так как код k уже передан в процессор CPU1. Эти команды помечены белыми точками. Каждая пустая команда может быть

представлена одним битом, например, нулевым.

В измерительном интервале T2 устройство А выдает в линию связи m пакетов. В этот же период, как предполагаем, внешнее устройство, подключенное к устройству В, в течение некоторого времени находится в состоянии переполнения и поэтому не способно принять три пакета — 3, 4 и 5 (эти пакеты помечены на рис. 5 «кляксами»). Пакеты теряются, сигнал FR1-OUT в период их прохождения стабилен, поэтому в счетчике процессора CPU2 накапливается на три единицы меньше (код $m-3$), чем в счетчике процессора CPU1 (код m).

Этот результат, однако, теряется, так как у устройства В в интервале T3 не оказалось «транспортного средства» в виде импульса J3, чтобы передать его в процессор CPU1. Иными словами, один цикл измерения пропущен, что несущественно, так как в усреднении результатов такие циклы не участвуют.

В измерительном интервале T3 передано n пакетов, а принято на два пакета меньше. Код $n-2$ передается в процессор CPU1 и т. д.

Таким образом, процессор устройства А может постоянно оценивать ситуацию и корректировать скорость передачи пакетов или их длину.

Напомним, что речь идет о предотвращении потерь пакетов в связи с перегрузкой приемника. Однако, как уже отмечалось, потери пакетов могут быть также обусловлены искажениями принимаемого сигнала. С увеличением доли таких потерь ошибка адаптации возрастает и приводит к тому, что скорость выдачи пакетов передатчиком неоправданно снижается, так как с его точки зрения приемник перегружен, что не соответствует действительности. Для устранения таких ошибок адаптации при мониторинге потерянных пакетов следует выделять из них те, которые отвергнуты внешним устройством в связи с неправильной контрольной суммой, а не из-за перегрузки приемника.

Сортировку потерянных пакетов можно осуществить при введении в систему дополнительной линии, по которой в процессор CPU2 передается сигнал ERROR (штриховая линия на рис. 4).

Идея сортировки такова. Внешнее устройство, подключенное к устройству В, «знает» причину уничтожения

пакета. Если пакет не принят из-за некорректируемой ошибки (обнаружена неправильная контрольная сумма, и ошибку исправить невозможно), то сигнал FR1-OUT не формируется, так как пакет не принят, но взамен него формируется сигнал ERROR. Этот сигнал поступает в процессор CPU2 и вызывает прибавление единицы к дополнительно введенному в него внутреннему счетчику ошибочных пакетов. Начальная установка и передача содержимого обоих счетчиков процессора CPU2 в устройство А выполняются совместно.

Если пакет не принят из-за перегрузки приемника, то не формируются оба сигнала FR1-OUT и ERROR.

Предположим, что в течение некоторого интервала измерения оправлено 250 пакетов, получено 240, причем в счетчике ошибочных пакетов зафиксировано число 3. Из этих сведений процессор CPU1 заключает, что семь из десяти потерянных пакетов не приняты внешним устройством из-за перегрузки, а оставшиеся три пакета отвергнуты им в связи с наличием в них некорректируемых ошибок. При адаптации скорости в расчет принимаются только семь указанных пакетов.

ЗАКЛЮЧЕНИЕ

В статье рассмотрена задача измерения некоторых параметров канала связи между узлами компьютерной сети. К ним относятся, в частности, задержка распространения сигнала и уровень потерь передаваемых информационных пакетов. При мониторинге этих параметров узлы сети обмениваются служебной информацией. В отличие от известных, предлагаемые системные и схемные решения позволяют уменьшить объем служебной информации, передаваемой по каналам связи, благодаря использованию вероятностной синхронизации [3, 5].

ЛИТЕРАТУРА

6. Пат. США № 7.035.214 В1 <http://www.uspto.gov>.

Борис Шевкопляс,
г. Москва

Окончание. Начало № 9/2006

Функциональные генераторы

Наша и, особенно, зарубежная промышленность выпускает множество функциональных генераторов. Мы рассмотрим наиболее доступные из них. К сожалению, ныне положение таково, что более распространены на нашем рынке зарубежные генераторы, таковы законы конкуренции.

Малогабаритный и легкий прибор FG2CE фирмы Wavetek Meterman (рис. 9) представляет собой типичную бюджетную модель функционального генератора для широкого применения.

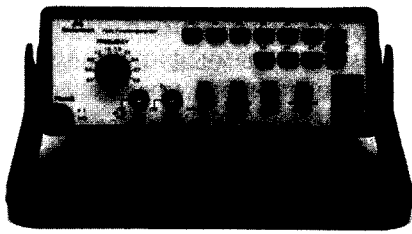


Рис. 9

Диапазон частот 0,3 Гц...3 МГц покрывается семью поддиапазонами. Плавное изменение частоты в 20 с небольшим раз осуществляется ручкой с лимбом. Прибор вырабатывает следующие виды сигналов — синусоидальный (коэффициент гармоник менее 1 % на частотах до 200 кГц), треугольный, прямоугольный, TTL и CMOS. Возможна перестройка частоты до 1:100 при подаче внешнего напряжения 0...10 В. Габариты прибора 292×143×93 мм, вес 2 кг.

Поскольку частота является важным параметром любого измерительного генератора, разумно объединение функционального генератора с простым цифровым частотомером. Это и сделано в приборе FG3CE (рис. 10).

Этот прибор построен на основе ранее описанной модели и имеет иден-

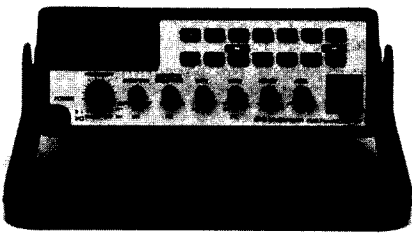


Рис. 10

тичные с ней параметры собственно генератора. Цифровой электронный частотомер прибора построен по счетному принципу и имеет шесть разрядов индикации. Диапазоны частот — 0,3 Гц...3 МГц при контроле частоты генератора и 5 Гц...150 МГц на внешнем входе. Габариты и вес прибора аналогичны приведенным выше для прибора FG2CE.

Южнокорейская компания Metex получила известность на нашем рынке своими бюджетными измерительными приборами и комплексами, например настольными и портативными мультиметрами и портативными осциллографами. Выпускает она и комбинированные измерительные приборы, имеющие широкие возможности в выполнении разнообразных измерений в сочетании с вполне умеренной ценой. Они прекрасно подходят для служб ремонта и сервиса сложной электронной техники, исследовательских лабораторий и даже для радиолюбителей. Такой генератор MSG-9810A автор имеет в своей лаборатории.

Функциональные генераторы и цифровые частотомеры MSG-9802A/9810A/9816A (рис. 11) отличаются только диапазонами частот функционального генератора.

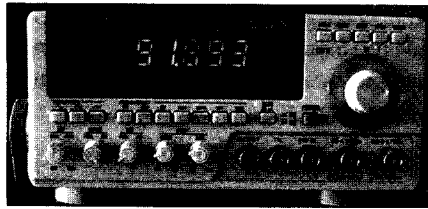


Рис. 11

У прибора MSG-9802A он составляет 0,2 Гц...2 МГц, у MSG-9810A — 1 Гц...10 МГц, а у MSG-9816A — 1 Гц...16 МГц. Этот диапазон разбит на 7 поддиапазонов, выбираемых кнопками под цифровым индикатором. Функциональный генератор формирует сигналы синусоидальной формы, прямоугольные (меандр) и треугольные. Выбор формы сигналов осуществляется кнопками. Размеры прибора 213×80×260 мм (у MSG-9802A — 280×213×110 мм).

Генератор позволяет добавлять к этим сигналам постоянную составля-

ющую и менять соотношение времен нарастания и спада сигналов. Все это позволяет получать множество форм для тестирования электронных устройств и проведения исследования.

Управление генераторами достаточно удобно и осуществляется с помощью кнопок и переменных резисторов с переключателями (выдвигаемые ручки). Частота сигналов меняется с помощью многооборотного резистора, оснащенного ручкой с верньером и шкалой. Сама шкала чисто символическая и установка частоты осуществляется по цифровому частотомеру. Поскольку функциональный генератор аналоговый, искажения формы сигналов, связанные с дискретизацией, отсутствуют.

Автором была практически оценена работа функционального генератора MSG-9810A. Учитывая довольно высокую частоту (10...16 МГц), для этого применялся цифровой широкополосный (250 МГц) осциллограф DS-1250 фирмы EZ Digital, подключенный по интерфейсу USB к компьютеру. Применение такого комплекса позволяет не только наблюдать форму сигналов испытуемого генератора без искажений, но и получить в цифровой форме данные о десятке параметров сигналов по двум каналам.

На рис. 12 показано окно с осциллограммами синусоидального и TTL сигналов на частоте генератора, равной 1 МГц.

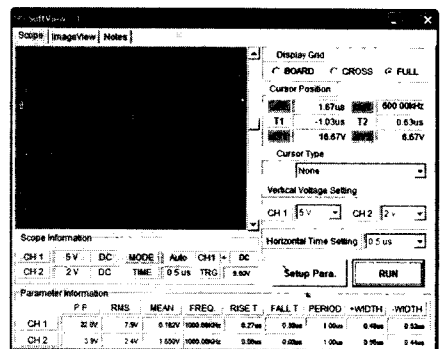


Рис. 12

На синусоиде слабо видны признаки ее дискретизации цифровым осциллографом. Но в целом форма синусоиды весьма близка к идеальной (коэффициент гармоник нормируется на уровне менее 1 % на частоте 1 кГц). Данные измерений различных амплитудных и временных параметров представлены внизу окна.

На рис. 13 и 14 представлены осциллограммы прямоугольного и треугольного сигналов на той же частоте.

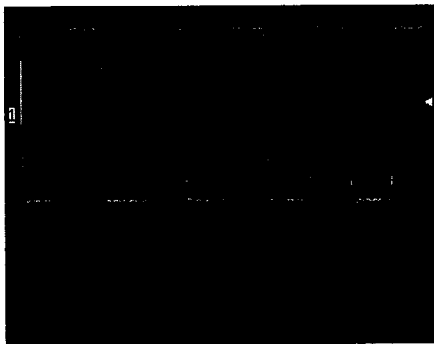


Рис. 13



Рис. 14

Можно сделать вывод о хорошей форме сигналов на частотах порядка 1 МГц. Это оправдывает наличие «низкочастотной» модели MSG-9802A, у которой нижняя граничная частота равна 0,1 Гц и гарантируется хорошая форма сигналов до предельной частоты 2 МГц.

Максимальные частоты даже куда более дорогих функциональных генераторов других фирм редко превосходят 3...5 МГц. Поэтому как немаловажное достоинство приборов Metex MSG-9810A и MSG-9816A можно отметить возможность генерации сигналов с частотой 10 и даже 16 МГц. Это резко расширяет их возможности в тестировании и наладке современных импульсных, радиоприемных и иных устройств.

Разумеется, ждать идеальной формы сигналов на таких частотах не приходится. Это подтверждают осциллограммы (рис. 15) для генератора MSG-9810A на предельной для него частоте 10 МГц.



Рис. 15

Тем не менее, нетрудно заметить, что даже на этой частоте треугольный сигнал имеет удовлетворительную форму со слегка скругленными верхушками. Форма прямоугольных импульсов, увы, далека от идеальной, как на TTL, так и на основном выходе. Для ее улучшения рекомендуется подключать к выходу согласованную нагрузку в 50 Ом.

Как уже отмечалось, у генераторов есть возможность регулировки асимметрии сигналов — отношение длительности нарастания и спада примерно от 1:3 до 3:1 и более (рис. 16).

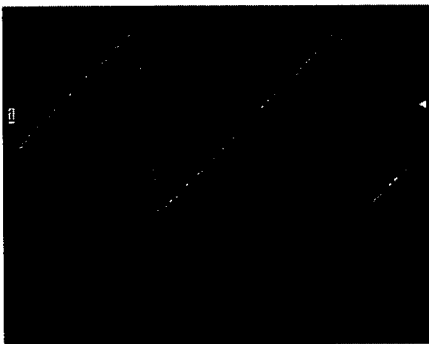


Рис. 16

Введение асимметрии позволяет получать линейно-нарастающие и линейно-спадающие сигналы, а также сигналы прямоугольной формы со скважностью, отличающейся от меандра. Это также расширяет области применения прибора, например, позволяет использовать его в качестве генераторов развертки и запускающих импульсов.

Наряду с основным выходом OUT (амплитуда сигнала 20 В без нагрузки и 10 В на согласованной нагрузке 50 Ом) предусмотрен выход TTL для несимметричных почти прямоугольных импульсов с параметрами, характерными для TTL микросхем. При вытягивании ручки регулировки амплитуды включается аттенюатор, ослабляющий сигнал в 10 раз (20 дБ).

Цифровой частотомер описываемых приборов имеет два канала. Канал А обеспечивает измерение частот до 20 МГц с высокоомным входом (1 МОм, максимальное напряжение от пика до пика 35 В), канал В — частот 20 МГц...2,7 ГГц на 50-омном входе (максимальное напряжение от пика до пика 3 В). Измерение производится подсчетом периодов сигналов в интервалах времени 0,1, 1 и 10 с. Это соответствует разрешению по частоте 10, 1 и 0,1 Гц. Результаты измерения внешней частоты или частоты функционального генератора

отображаются на восьмиразрядном цифровом индикаторе (с белым или красным цветом цифр). Для канала А предусмотрена возможность подачи сигнала с выхода функционального генератора (для контроля его частоты) или с входа СН-А.

Для своих мультиметров и частотомеров, в том числе входящих в комбинированные приборы, компания Metex поставляет программу Bench-View, которая позволяет вводить данные измерения частоты в компьютер и отображать их в табличной и графической форме. Разумеется, это возможно после подключения приборов к компьютеру с помощью прилагаемого интерфейсного кабеля RS-232 и установки программного обеспечения.

Представителем функциональных генераторов высокого класса может быть модель 33220A известной своими высококласными измерительными приборами фирмы Agilent. Внешний прибор представлен на рис. 17.

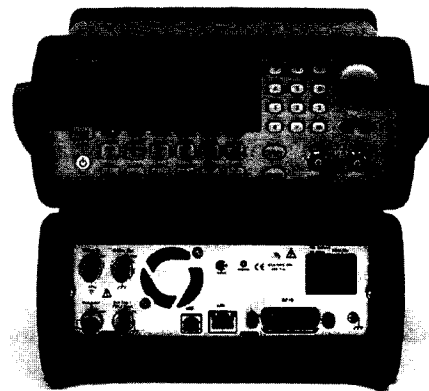


Рис. 17

Прибор имеет диапазон генерации до 20 МГц с возможностью цифровой установки частоты и ее контролем по встроенному цифровому частотомеру. Прибор вырабатывает 11 стандартных форм сигналов. Имеется возможность AM, FM, PM, FSK и PWM модуляции. Принцип формирования сигнала — цифровой. Используется 14-битовое кодирование со скоростью до 50 Мвыб/с при 64 К точек. Предусмотрены линейное и логарифмическое качание частоты и связь с компьютером по интерфейсам USB, GPIB и LAN.

Владимир Дьяконов,
г. Смоленск

Продолжение. Начало — № 8/2006

Работа с микроконтроллерами Atmel

АСЕМБЛИРОВАНИЕ ПРОГРАММЫ

Результат асемблирования программ может выводиться в различных форматах. По умолчанию результат сохраняется в файл *.hex, именно этот формат файла (Intel Intellect 8/MDS) может использоваться большинством программаторов для загрузки программ в память микроконтроллеров. Если формат файла был изменен и вы не можете найти файл *.hex, следует восстановить формат.

Для изменения формата файла, получаемого при асемблировании, откройте меню Project/Assembler Options, в открывшемся окне Assembler Options в окошке HEX Output format: выберите Intel Intellect 8/MDS.

Здесь же на панели AVR Assembler выберите Version 2. Для новых проектов разработчики AVR Studio рекомендуют использовать именно эту версию.

Для асемблирования программы можно воспользоваться меню Build/Build, кнопкой F7, кнопкой Assemble панели инструментов Assembler. По мере освоения AVR Studio вы сможете выбрать наиболее подходящие варианты.

Если у вас другая версия AVR Studio, отличная от версии 4.12 build 462, надо учитывать, что органы управления AVR Studio 4 в обновляемых версиях несколько отличаются, так асемблирование относилось к меню Project/Build, а панель инструментов Assembler отсутствовала.

Итак, для асемблирования нажмем клавишу F7. В результате в окно Build (рис. 6) выводится информация о результатах асемблирования.

Интерес в этом окне представляет следующая информация:

- имя асемблируемого файла программы;
- файлы, включенные в файл программы директивой `.include`;
- использование памяти программ микроконтроллера;
- сообщение о том, что ошибки не обнаружены.

Обнаружение ошибок при асемблировании

Если в программе обнаружены ошибки, содержимое окна Output поможет локализовать их.

```
AVRASM: AVR macro assembler version 1.77.3 (Sep 21 2005 08:43:03)
Copyright (C) 1995-2005 ATMEL Corporation
```

```
Assembling 'C:\avr\Alarm\Alarm.asm'
Including 'D:\Program Files\Atmel\AVR Tools\AvrAssembler\Appnotes\tn2313def.inc'
```

```
Program memory usage:
Code       : 72 words
Constants (dw/db): 0 words
Unused     : 0 words
Total      : 72 words
```

● Assembly complete with no errors.

Рис. 6

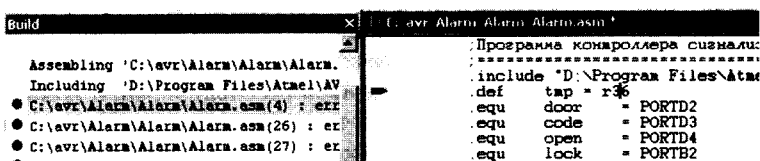


Рис. 7

Введем ошибку в директиву `.def tmp = r16`.

Имя tmp назначим регистру r36, которого не существует в рассматриваемых микроконтроллерах — `.def tmp = r36`. Снова асемблируем программу (клавиша F7). В окне Output появились сведения об обнаруженных ошибках (рис. 7).

Каждой ошибке (error) соответствует сообщение, представленное одной строкой в окне Output.

Например: `C:\avr\Alarm\Alarm.asm(4): error: Register R0-R31 expected`. Это значит, что в файле `C:\avr\Alarm\Alarm.asm` в строке 4 обнаружена ошибка — могут быть использованы регистры r0...r31, а мы попытались использовать r36.

Поскольку в программу директивами `.include` может быть включено несколько файлов, в каждом из которых может обнаружиться ошибка, указание полного имени файла в сообщении не является лишним.

Как быстро найти в программе строку с ошибкой? На приведенном выше рисунке представлен фрагмент окна AVR Studio с окном Build и окном программы Alarm.asm. Нужно выполнить двойной щелчок по сообщению об ошибке в окне Build, тогда против строки с ошибкой в окне программы появится указатель.

Почему обнаружилось так много ошибок, ведь мы ввели только одну? Теперь ошибка обнаруживается в каждой строке, содержащей имя tmp, которое мы попытались присвоить несуществующему регистру. Перед дальнейшей работой устраните ошибку, восстановив наименование регистра r16.

Определение опций симулятора

Вновь асемблируйте программу (F7), не содержащую ошибок.

Для выполнения пошаговой отладки откроем меню Debug/Start Debugging, в окне программы слева от первой команды появится желтая стрелка, индицирующая ход выполнения отладки.

Только теперь открыт доступ к опциям симулятора AVR Studio, для их определения откроем окно Simulator Options (меню Debug/AVR Simulator Options. Заметим, что до начала отладки данная строка отсутствовала в меню Debug).

Здесь можно определить тип микроконтроллера, для которого написана программа; для этого щелкните мышкой по стрелке, расположенной справа в конце окошка Device, в открывшемся списке выберите микроконтроллер ATtiny2313 (рис. 8).

Тип выбранного микроконтроллера должен соответствовать файлу `*def.inc`, включаемому директивой `.include` в программу, поэтому в нашу программу включим файл `tn2313def.inc`.

В окошке Frequency таким же образом выберите частоту тактового генератора микроконтроллера из списка или введите ее вручную. Для нашего устройства эта частота равна 4 МГц и соответствует резонансной частоте кварцевого резонатора, подключенного к микроконтроллеру.

Значение частоты может быть дробным, например, 7.37 МГц, причем поддерживается лишь два

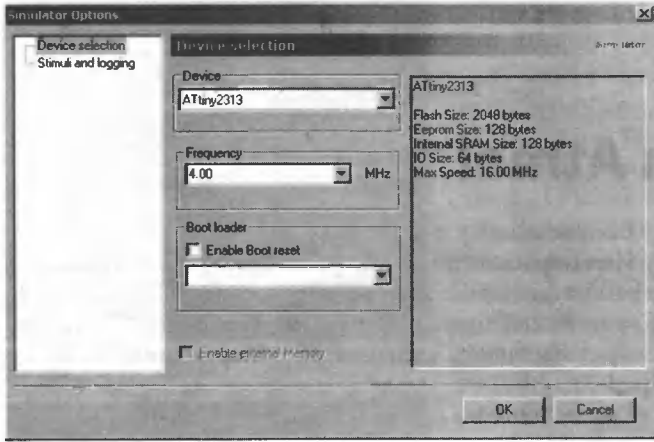


Рис. 8

знака после десятичной точки. Если введенная частота окажется выше допустимой для выбранного устройства, сообщение об этом появится в окне Message после нажатия кнопки OK.

О файлах *def.inc

Мы начали отладку. В окне программы желтая стрелка указателя отладчика установлена против первой команды: `jmp RESET`. Теперь можно просмотреть содержимое файла `tn2313def.inc`. На рис. 9 демонстрируется, как открыть файл `*def.inc` для просмотра.

Для того, чтобы открыть файл `*def.inc`, добавим его в список файлов в окне Project. Для этого щелкнем правой клавишей мышки по списку Source Files, в открывшемся окне — по строке Add Files to Project, в появившемся окне «Открытие файла» найдем и откроем файл — `D:\Program Files\Atmel\AVR Tools\AvrAssembler2\App\notes\tn2313def.inc`.

Двойным щелчком откройте добавившийся к списку Source Files файл `tn2313def.inc`, на экране поверх окна программы появится окно с содержимым интересующего нас файла. Большую часть файла составляют директивы `.equ`, позволяющие присваивать имена константам, например, `.equ PORTB = $18`.

Конечно, можно использовать в программе вместо имени `PORTB` константу `$18`, но удачно подобранные имена значительно облегчают чтение и понимание программ. Это же касается и директивы `.def`, которая используется в программе. Отличие в том, что она позволяет назначить имена регистрам.

Замечу, что одним и тем же константам или регистрам можно присвоить несколько имен в зависимости от их назначения в том или ином блоке программы. Например, в файле `tn2313def.inc` константе 2 назначено несколько имен — `PORTB2`, `DDD2`, `PORTD2` (в файле `tn2313def.inc` есть и директива `.equ PORTD2=2`). В нашей программе директива `.equ door = PORTD2` назначает константе 2 имя `door`, но константа представлена не явно, а одним из присвоенных ей имен `PORTD2`.



Рис. 9

В связи с ограниченным числом регистров в микроконтроллере одни регистры в разных блоках программы могут использоваться для разных целей под разными именами. Так, в одном месте регистр `r18` может использоваться как счетчик циклов (`counter`), в другом — как регистр временного хранения (`store`), поэтому удобно для этого регистра назначить соответствующие имена — `.def store = r18` или `.def cnt = r18`. В окне Program Output после ассемблирования программы появится предупреждение (`warning`) о том, что регистру уже назначено имя.

Поскольку обращение как к `store`, так и к `cnt`, это обращение к регистру `r18`, надо избегать конфликтов при их совместном использовании, ведь изменение содержимого `store` автоматически изменяет содержимое `cnt`.

Векторы прерываний

Еще две директивы нашей программы требуют размещения последующих команд в памяти программ (директива `.cseg`), начиная с нулевого адреса (директива `.org 0`).

С нулевого адреса в микроконтроллере размещаются векторы прерываний. Каждый тип микроконтроллеров AVR имеет индивидуальный набор прерываний (для микроконтроллера ATtiny2313 их 19) и равное им число векторов прерываний.

Подробно работа прерываний будет рассмотрена далее. Здесь заметим, что прерывания могут приостановить нормальный ход программы, при котором за одной командой выполняется следующая, адрес которой находится в счетчике команд микроконтроллера. Для каждого прерывания имеется своя ячейка в памяти программ (вектор прерывания), в эту ячейку записывается команда перехода к обработчику прерывания (к подпрограмме), который должен выполняться при обнаружении прерывания. По окончании обработки прерывания выполнение программы продолжается обычно с той команды, перед выполнением которой произошло прерывание.

За директивой `.org 0` следует команда `jmp RESET`, за которой 18 команд `por` (команда, не выполняющая каких-либо действий). Это значит, что в программе используется лишь вектор прерывания `RESET`, для остальных обработчиков в программе не предусмотрены.

Подпрограммы-обработчики прерываний входят в состав программы микроконтроллера и могут размещаться в любой ее части после блока векторов прерываний.

В предложенном варианте программы только одно прерывание — `RESET`. Команда перехода вместе с адресом подпрограммы, обрабатывающей прерывание `RESET`, хранится в ячейке памяти программ с адресом `0x0000`.

Обработчик этого прерывания выполняется после изменения уровня с НИЗКОГО на ВЫСОКИЙ на контакте `RESET` микроконтроллера. В нашем устройстве это происходит после подачи питания на микроконтроллер или с некоторой задержкой после отпускания кнопки `SA4` (см. описание схемы).

Чтобы обозначить начало обработчика, в программе ставится метка — слово с двоеточием (в нашей программе это метка `RESET`), находящаяся сразу за последним вектором прерывания), а ее адрес надо записать в нулевую ячейку памяти программ. При ассемблировании команды `jmp RESET` в нулевую ячейку памяти запишется код команды `jmp` и адрес, по которому ассемблер разместит команду, помеченную меткой `RESET`.

Итак, в нулевой ячейке хранится адрес начала обработчика RESET; в следующих 18-ти ячейках в строгой последовательности должны храниться адреса оставшихся прерываний микроконтроллера ATtiny2313. Так как мы не используем эти прерывания, вместо адресов помещаются пустые команды пор. Однако в тексте приведенной программы за каждой командой пор следует закомментированная ссылка (перед ней ставится точка с запятой) на подпрограмму обработки прерывания. Так, если понадобится использовать внешнее прерывание INT0, следует убрать первый оператор пор и точку с запятой за ним, тогда в этой строке окажется команда `rjmp INT0`. В программу понадобится добавить метку `INT0`, которая обозначит начало подпрограммы обработки внешнего прерывания. Завершить подпрограмму обработки прерывания надо командой `reti`.

Как узнать, в какой последовательности и сколько прерываний должно быть в блоке?

Такой блок можно найти в полном техническом описании соответствующего микроконтроллера в разделе *Interrupts/Interrupt Vectors*, либо в конце файла `*def.inc`.

Ниже приведен блок векторов прерываний (*Interrupt Vectors*), скопированный из файла `tn2313def.inc` без последних трех строк, оставленных для совместимости с программами, написанными для микроконтроллера AT90S2313:

```

;*****
;**** Interrupt Vectors ****
;*****
.equ INT0addr    = 0x001 ;External Interrupt0
.equ INT1addr    = 0x002 ;External Interrupt1
.equ ICP1addr    = 0x003 ;Input capture interrupt 1
.equ OC1Aaddr    = 0x004 ;Timer/Counter1 Compare Match A
.equ OVF1addr    = 0x005 ;Overflow1 Interrupt
.equ OVF0addr    = 0x006 ;Overflow0 Interrupt
.equ URXC0addr   = 0x007 ;USART0 RX Complete Interrupt
.equ UDRE0addr   = 0x008 ;USART0 Data Register Empty
                    ; Interrupt
.equ UTXC0addr   = 0x009 ;USART0 TX Complete Interrupt
.equ ACIaddr     = 0x00A ;Analog Comparator Interrupt
.equ PCINTaddr   = 0x00B ;Pin Change Interrupt
.equ OC1Baddr    = 0x00C ;Timer/Counter1 Compare Match B
.equ OCOAaddr    = 0x00D ;Timer/Counter0 Compare Match A
.equ OCOBaddr    = 0x00E ;Timer/Counter0 Compare Match B
.equ USI_STARTaddr = 0x00F ;USI start interrupt
.equ USI_OVFaddr  = 0x010 ;USI overflow interrupt
.equ ERDYaddr    = 0x011 ;EEPROM write complete
.equ WDTaddr     = 0x012 ;Watchdog Timer Interrupt

```

Здесь отсутствует лишь самое первое прерывание RESET, размещаемое по нулевому адресу. В этом блоке соблюдается последовательность векторов прерываний, но для размещения в программе его надо откорректировать (см. блок в программе). Обычно эти блоки размещаются в конце файлов `*def.inc`.

Основная программа

Основная программа начинается командой с меткой `RESET`:

Первая пара операторов определяет адрес начала стека в оперативной памяти микроконтроллера. Работу стека мы еще рассмотрим, поэтому сейчас заметим, что стек это область оперативной памяти или оперативного запоминающего устройства микроконтроллера (далее ОЗУ). Стек

предназначен в основном для обслуживания подпрограмм, при занесении в стек нескольких величин адрес каждой последующей величины уменьшается (у первой самый большой адрес, у последней — самый маленький). Поэтому стек обычно располагают в конце ОЗУ, начиная с последней ячейки ОЗУ.

Мы прервались после первого шага отладки — стрелка указателя отладчика установлена против первой команды `rjmp RESET`.

Откройте окно просмотра состояния процессора (щелчок по закладке *I/O View*, находящейся внизу окна, расположенного слева). Далее слова «просмотр состояния» будут опускаться.

В окне *I/O View* указана выбранная нами тактовая частота 4 МГц, в окошках счетчика тактов микроконтроллера (*Cycle Counter*), указателя стека (*Stack Pointer*), счетчика команд (*Program Counter*), времени выполнения (*Stop Watch*) — нули.

На рис. 10 показано окно *I/O View*, в котором отображается состояние процессора во время отладки программы.

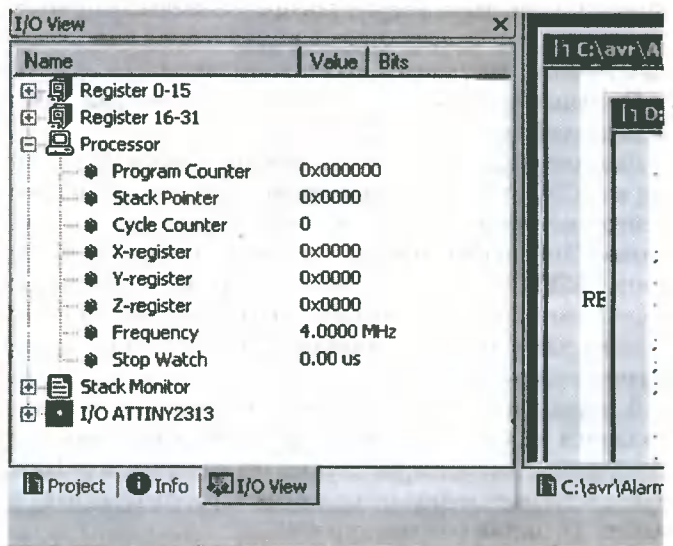


Рис. 10

Откройте окно памяти (меню *View/Memory*, рис. 11).

Кнопкой `8/16` выберите отображение в виде двухбайтных слов, как на рисунке.

Заметьте, что в нулевой ячейке памяти программ находится число `12C0` (правильнее было бы написать `$12C0` или `0x12C0`, что указывает на шестнадцатеричное представление величины, однако вид записи соответствует виду, представленному в описываемом окне). Откроем окно *Disassembler* (рис. 12), выбрав меню *View/Disassembler*.

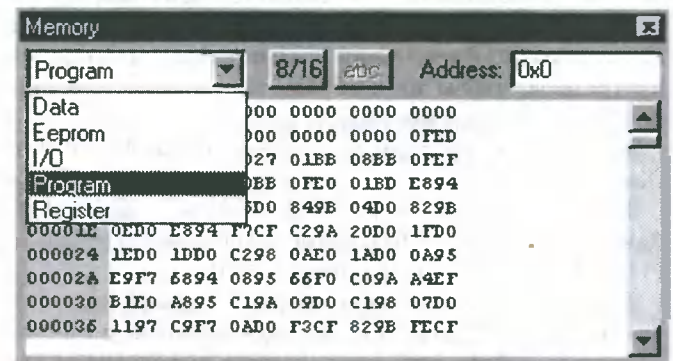


Рис. 11

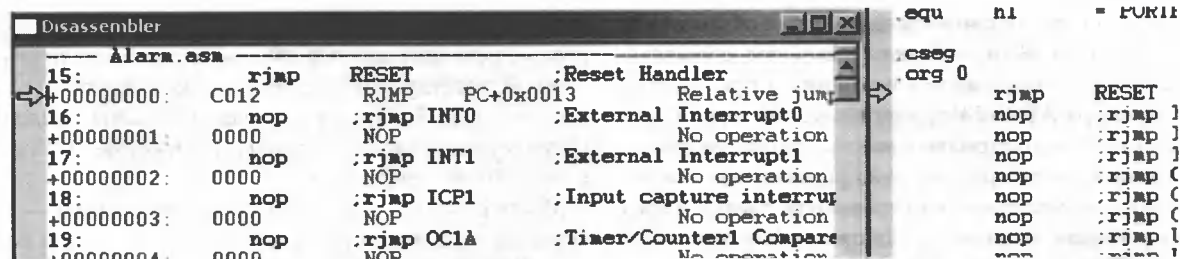


Рис. 12

Открывшееся окно Alarm закрывает собой окно программы Alarm.asm, поэтому его лучше сместить, как это показано на рис. 12.

В первой колонке окна Alarm находится адрес двухбайтной ячейки памяти программ, во второй колонке — код команды, хранящейся в этой ячейке, в третьей — наименование команды, такое же, как и в нашей программе, в четвертой — операнды, приведенные к приемлемому для микроконтроллера виду. Например, вместо имени tmp здесь будет имя регистра R16, вместо метки RESET: — смещение адреса и так далее. В пятой колонке находятся комментарии. Здесь могут быть представлены, например, результаты адреса переходов или константы в шестнадцатеричном, двоичном и десятичном видах.

По существу в этом окне представлен код программы в дизассемблированном виде.

Во второй колонке строки с адресом +00000000: — тот же код C012, только с измененной последовательностью байтов, который находится в нулевой ячейке памяти программ. Это двухбайтный код, соответствующий команде jmp RESET. C0 — код команды jmp, 12 — смещение адреса, которое надо добавить к содержимому счетчика команд для перехода к обработчику прерывания, начинающемуся меткой RESET:.

В результате выполнения этой команды в счетчике окажется адрес 12, для перехода к следующей команде содержимое счетчика увеличится на единицу, в результате следующей выполнится команда, расположенная по адресу 13 (пятая колонка строки).

С адреса 00000001 по адрес 00000012 следуют коды 0000, соответствующие пустым командам пор, и так далее.

Вероятно, вы уже уловили связь между строкой в программе и кодом, исполняемым микроконтроллером.

Новый шаг отладки вызвал бы перемещение указателя отладчика к следующей команде как в окне программы, так и в окне Disassembler. Не многовато для первого шага отладки?

Если материал для вас пока не очень понятен, не расстраивайтесь, просто запомните, что вопрос о такой-то возможности отладчика рассматривался, тогда при необходимости вы сможете вернуться к этой части. Поэтому дать максимум материала по отладке целесообразно именно при рассмотрении простейшей программы, чтобы не возвращаться к нему при рассмотрении более сложных.

Вернемся к программе, закрыв окна Memory и Disassembler.

Выполним второй шаг отладки, нажав клавишу F11. Далее подобный текст будет заменен фразами «выполним F11:» «нажмите F11:» или просто «F11:».

В окне I/O View состояние счетчика стало равным \$13, что соответствует адресу следующей команды, в счетчике тактов — 2, то есть, команда jmp выполнена за два такта микроконтроллера, в счетчике времени — 0,5 мкс,

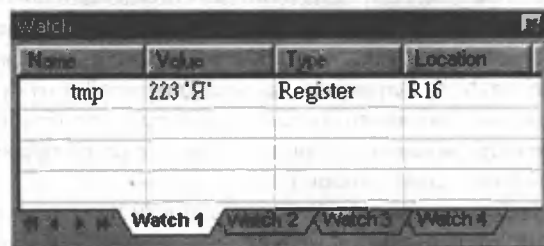


Рис. 13

что соответствует двум тактам микроконтроллера при частоте 4 МГц.

Константа RAMEND в следующей команде хранит адрес последней ячейки ОЗУ. Значение этой константы (\$DF), а также других констант, которые не определены в самой программе директивой .equ, можно найти в файле tn2313def.inc.

Функция low(RAMEND) вычисляет младший байт константы RAMEND. Для определения адреса последней ячейки ОЗУ микроконтроллера ATtiny2313 достаточно одного байта, поэтому указатель стека представлен только одним регистром SPL и используется только младший байт адреса, а старший байт равен нулю.

Краткое описание работы самих команд ищите в разделе Instruction Set Summary полного технического описания микроконтроллера. Подробное описание — в Help AVR Studio, для получения помощи установите указатель мыши на команду и нажмите F1.

Откройте меню View/Watch, щелкните правой кнопкой мыши в появившемся окне, выберите строку Add Item (рис. 13).

В окне появится строка ввода в колонке Watch. Введите в нее имя переменной tmp. Откройте меню View/Register и посмотрите, что находится в регистре r16 — его содержимое соответствует значению переменной tmp, так как переменная tmp хранится в регистре r16.

Удобно одновременно просматривать изменения сразу в нескольких окнах, для этого надо тщательно устанавливать их размер и выбирать область просмотра. Например, окно Watches можно значительно сузить, перемещая влево вертикальные линии, разделяющие колонки. Для этого установите указатель мыши на вертикальную линию между заголовками колонок, например, между Name и Value, при этом должна появиться черная вертикальная полоска со стрелками налево и направо, нажмите левую клавишу и, не отпуская ее, переместите линию влево.

В окне I/O View можно убрать неразвернутые строки, для этого надо нажать правую клавишу мыши и выбрать строку Hide non-expanded. Для восстановления скрытой строки надо нажать правую клавишу мыши и выбрать строку Show all items.

Продолжение следует

Вадим Баранов,
г. Харьков, Украина

Аналитические расчеты в электронике

4. ОСОБЕННОСТИ ВХОДНОГО ЯЗЫКА MAPLE

Мы уже пользовались Maple для расчетов моделей. Примеры были весьма прозрачны для понимания, т. к. входной язык весьма напоминает BASIC, разница только в том, что система команд Maple гораздо мощнее, это язык программирования сверх-высокого уровня. Команды Maple в большинстве случаев соответствуют общеизвестным математическим функциям и операторам, но записанным в виде текстовой строки. В дополнение к этому Maple имеет палитры для ввода команд в классических математических символах. Режим отображения показа команд текст/символ определяется специальной кнопкой. Удобно вводить команды и выражения текстовой строкой, а для проверки правильности ввода преобразовывать ее в символьный вид. Это актуально при вводе сложных выражений, когда легко запутаться во множестве скобок, которые влияют на очередность выполнения математических операций. Текст Maple в отличие от многих других математических программ содержит полную информацию о решаемой задаче и не содержит скрытых настроек. Это очень удобно, когда документ Maple публикуется в печатном издании, так как при этом не происходит потери информации.

В этой главе мы опишем только особенности входного языка Maple. Для более глубокого изучения Maple можно рекомендовать [1]. Кроме этого в Maple есть встроенный справочник по всем командам, правда англоязычный, но вполне понятный.

В примерах по ходу решения задач даются комментарии о выполняемых действиях, поэтому примеры можно использовать как справочник по применению команд. Смысл всех этих команд — указать машине, какие именно математические операции и в каком порядке надо выполнить, а математика для всех одна.

Команда должна заканчиваться точкой с запятой «;». Если затем нажать клавишу Enter, команда будет выполнена и на экране появится ре-

зультат. Если команда заканчивается двоеточием «:», она выполняется, результат запоминается, но не выводится на экран. Результат выполнения предыдущей команды обозначается знаком «%». Комментарий в командной строке отделяется от команды знаком «#».

Структура команд Maple

В общем виде структура команды Maple выглядит так:

```
> имя команды (операнды, [модификаторы]);  
[# комментарий]
```

- Имя команды указывает конкретную математическую операцию, которую следует выполнить. Например, упростить, разложить на множители, дифференцировать, интегрировать, взять предел, построить график и т. д. Имя команды, как правило, должно начинаться со строчной буквы, тогда она будет выполняться. Если команда начинается с заглавной буквы, ее выполнение будет отложено до ввода команды value(), содержащей ссылку на отложенную команду.

- Операнд — переменные или выражения, над которыми выполняется математическая операция. Операндом может быть ссылка на другую команду, результаты исполнения которой станут операндом. В качестве ссылок можно использовать символы %, %, %, которые указывают на результаты предыдущих команд, начало отсчета снизу вверх. Это позволяет быстро организовать передачу данных между командными строками.

- Модификаторы — ключевые слова, которые изменяют действие команды таким образом, чтобы получить результат в форме, удобной для последующего анализа. При отсутствии модификатора используется модификатор по умолчанию.

- # Комментарий — любой текст, поясняющий действие команды. Если знак «#» поставить перед командой, командная строка превращается в комментарий и не выполняется. Это удобно для временной отмены действия команды без ее удаления.

В Maple существуют системные команды, которые управляют вы-

числительным процессом. Покажем некоторые из них.

```
> restart: # сброс всех переменных  
> Digital:=7: # установка точности вычислений, по умолчанию установлено 10 знаков.  
> with(ScientificConstants): # подключение пакета расширения для работы с константами.  
> with(plots): # подключение пакета расширенной графики.
```

Warning, the name changecoords has been redefined

Надо сказать, что параметр Digital в Maple может быть очень большим, я пробовал 100000. Это просто грандиозная цифра! Maple содержит множество пакетов расширения. Они подгружаются по мере необходимости, если стандартных средств для решения задачи не хватает. Команда restart вызывает очистку всех переменных, ее рекомендуется всегда выполнять в начале программы. Сбросить переменную выборочно, например R2, можно следующей записью.

```
> R2:='R2':
```

Арифметика

Сложение 2+3, вычитание 2-3, умножение 2*3, деление 2/3, возведение в степень 2^3.

Порядок выполнения действий:
действие первого уровня — возведение в степень,
действия второго уровня — умножение и деление,
действия третьего уровня — сложение и вычитание.

Действия одинакового уровня выполняются слева направо:

```
> 3+5*3-6/7;
```

```
890  
7
```

```
> evalf(%);  
127.1428571
```

Встроенные функции

Почти все они пишутся с маленькой буквы.

```
> # sin(x); cos(x); exp(x); ln(x); log10(x)  
; log[n](x);  
# sqrt(x); abs(x); n!; signum(x); sin(x);  
cos(x); tan(x);  
# cot(x); arcsin(x); arccos(x); arctan(x)  
; arccot(x);
```

```
# sinh(x); cosh(x); tanh(x); arcsinh(x);
arccosh(x);
# arctanh(x); Heaviside(x); Dirac(x);
floor(x);
```

Упрощение выражений

Функция `simplify` — одна из самых мощных. Эта функция многоцелевая и предназначена для упрощения математических выражений. Она выполняет следующие типовые действия:

- комбинирует подвыражения;
- приводит подобные множители в произведениях;
- приводит подобные члены в суммах;
- использует тождества, содержащие нуль;
- распределяет целочисленные показатели степени в произведениях;
- сокращает `expr` на наибольший общий полиномиальный или иной множитель;
- понижает степень полиномов, если это возможно;
- использует преобразования, способные упростить выражения.

Функция `simplify` возвращает упрощенное выражение `expr` или повторяет его, если Maple не может его упростить.

```
> simplify(expr);
expr
```

Возвращает упрощенное выражение `expr` с учетом параметров с именами `n1`, `n2`, ... (в том числе заданных списком или множеством)

```
# simplify(expr, n1, n2, ...);
```

Возвращает упрощенное выражение с учетом всех условий, представленных равенством или списком равенств.

```
# simplify(expr, assume=prop);
```

Ниже даны примеры применения функции `simplify`:

```
> simplify(4^(1/2)+3);
5
> simplify((x^y)^z+3^(3), power);
(x^y)^2 + 27
> simplify(sin(x)^2+cos(x)*2, trig);
2cos(x) + 1 - cos(x)^2
> simplify(ln(x*y), power, symbolic);
ln(x) + ln(y)
```

Расширение выражений

Порою упрощенное выражение скрывает его особенности, знание которых является желательным. Функция `expand` «расширяет» выражение `expr` и записывается в виде:

```
expand(expr, expr1, expr2 ... exprn)
```

где `expr` — расширяемое выражение, `expr1`, `expr2`, ..., `exprn` — необязательные подвыражения опции. Функция `expand` раскладывает рациональные выражения на простые дроби, полиномы на полиномиальные разложения, она способна раскрыть многие математические функции, такие как `sin`, `cos`, `tan`, `sinh`, `cosh`, `tanh`, `det`, `erf`, `exp`, и т. д. С помощью дополнительных аргументов `expr1`, `expr2`, ..., `exprn` можно задать расширение отдельных фрагментов в `expr`. Примеры применения функции `expand` приведены ниже:

```
> expand((x+2)*(x+3)*(x+4));
x^3 + 9x^2+26x + 24
> expand(sin(2*x));
2sin(x)cos(x)
> expand(sin(x+y));
sin(x)cos(y) + cos(x)sin(y)
> expand((x+1)*(y+z));
xy + xz + ly + lz
> expand((x+1)*(y+z), x+1);
(x+1)y + (x+1)z
```

Комплектование по степеням

Еще одна функция общего назначения, `collect`, служит для комплектования выражения `expr` по степеням указанного фрагмента `x` (в том числе множества, либо списка). Она задается в одной из следующих форм:

```
collect(a, x)
collect(a, x, from, func)
```

Во второй форме этой функции дополнительно задаются параметры `form` (форма) и `func` (функция или процедура). Параметр `form` может иметь два значения — `recursive` (рекурсивная форма) и `distributed` (дистрибутивная форма). Параметр `func` позволяет задать имя функции, по которой будет идти комплектование `expr`.

Примеры применения функции `collect` представлены ниже:

```
> collect(x+x^3-2*x, x);
-x + x^3
> collect(x+2*y+3*x+3*x^3*y, recursive, x);
x(2x + 6y + 3 + x^3y)
> collect(x+2*y+3*x+3*x^3*y, distributive, y);
```

$y(2x + 6y + 3 + 3xy)$

```
> p:=x*y*a*x*y+y*x^2-a*y*x^2+x+a*x;
p:=x^2y^2a+yx^2-ayx^2+x+ax
> collect(p,[x,y], recursive);
(y^2a + (1 - a)y)x^2 + (a + 1)x
```

Пределы, производные, интегралы, решение уравнений

Эти операции традиционно были сложными и трудоемкими. Однако без них математика — не математика, это основные инструменты математического анализа. Maple способен брать весьма сложные интегралы, производные, пределы и решать системы алгебраических и дифференциальных уравнений.

Пределы

```
> limit(sin(x)/x, x=0):=limit(sin(x)/x,
x=0);
```

$$\lim_{x \rightarrow \infty} \frac{\sin(x)}{x} := 1$$

Производные

```
> Diff(sin(x)/x, x)=diff(sin(x)/x, x);
```

$$\frac{d}{dx} \left(\frac{\sin(x)}{x} \right) = \frac{\cos(x)}{x} - \frac{\sin(x)}{x^2}$$

```
> f:=x->sqrt(x)+sin(2*x)+exp(1/
x)+(x^2+3*x)/(x+1);
```

$$f := x \rightarrow \sqrt{x} + \sin(2x) + e^{\left(\frac{1}{x}\right)} + \frac{x^2 + 3x}{x + 1}$$

```
> df:=Diff(f(x), x);
```

$$df := \frac{d}{dx} \sqrt{x} + \sin(2x) + e^{\frac{1}{x}} + \frac{x^2 + 3x}{x + 1}$$

```
> df=value(df);
```

$$\frac{d}{dx} \sqrt{x} + \sin(2x) + e^{\frac{1}{x}} + \frac{x^2 + 3x}{x + 1} = \frac{1}{2\sqrt{x}} + 2 \cos(2x) - \frac{e^{\left(\frac{1}{x}\right)}}{x^2} + \frac{2x + 3}{x + 1} - \frac{x^2 + 3x}{(x + 1)^2}$$

Интегралы

```
> int(sin(x)/x, x):=int(sin(x)/x, x);
```

$$\int \frac{\sin(x)}{x} dx := \text{Si}(x)$$

```
> indefINT := Int( sin(x)*cos(b*x), x );
> indefINT = value( indefINT );
```

$$\int \sin(x) \cos(bx) dx = \frac{1}{2} \frac{\cos((1+b)x)}{1+b} + \frac{1}{2} \frac{\cos((-1+b)x)}{-1+b}$$

```
> int(a*x^n, x):=int(a*x^n, x);
```

$$\int a x^n dx := \frac{x^{(n+1)} a}{n+1}$$

```
> int(x^5*exp(-x), x):=int(x^5*exp(-x),
x);

$$\int x^5 e^{-x} dx :=$$


$$= -(120 + 120x + 60x^2 + 20x^3 + 5x^4 + x^5) e^{-x}$$

```

```
> Int(exp(x^4), x):=int(exp(x^4), x);
```

$$\int e^{(x^4)} dx = -\frac{1}{4}(-1)^{(3/4)} \times$$

$$\times \left(\frac{x(-1)^{(1/4)} \pi \sqrt{2}}{\Gamma\left(\frac{3}{4}\right)(-x^4)^{(1/4)}} - \frac{x(-1)^{(1/4)} \Gamma\left(\frac{3}{4}\right) (-x^4)^{(1/4)}}{(-x^4)^{(1/4)}} \right)$$

РЕШЕНИЕ УРАВНЕНИЙ И НЕРАВЕНСТВ

Основная функция solve

Решение линейных и нелинейных уравнений и неравенств — еще одна важная область математического анализа. Maple имеет мощные средства для такого решения. Так, для решения линейных и нелинейных уравнений в аналитическом виде используется достаточно универсальная и гибкая функция solve(eq, var) или solve({eq1, eq2 ... }, {var, var2 ...}), где eq — уравнение, содержащее функцию ряда переменных, var — переменная, по которой ищется решение. Если при записи eq не используются знак равенства или знаки отношения, считается, что solve ищет корни уравнения eq = 0.

При решении систем уравнений они и список переменных задаются как множества, т. е. в фигурных скобках. При этом и результат решения получается в виде множества. Чтобы преобразовать его к обычному решению, нужно использовать функцию assign, которая обеспечивает присваивание переменным значений, взятых из множества.

Функция solve старается дать решение в аналитическом виде. Это не означает, что ее нельзя использовать для получения численных корней уравнений. Иногда для этого приходится дополнительно использовать функции evalf, convert или allvalues.

```
> eq1:= 2*y+x+3=0;
eq2:= 2*x+y-1=0;
eq1:= 2y + x + 3 = 0
eq2:= 2x + y - 1 = 0
> solve({eq1,eq2},{x,y});
```

$$\left\{ y = \frac{-7}{3}, x = \frac{5}{3} \right\}$$

```
> assign(%);
> x:=evalf(x); y:=evalf(y);
x:=1.66666666667
y:=-2.33333333333
```

```
> restart;
```

```
> eq3:=2*arcsin(x)-arccos(5*x);
eq3:=2arcsin(x) - arccos(5x)
```

```
> x:=solve(eq3,x);
```

$$x := -\frac{5}{4} + \frac{\sqrt{33}}{4}$$

Решение дифференциальных уравнений

Для RLC цепочки, включенной по схеме НЧ фильтра, на которую подается скачок единичного напряжения, с помощью функции dsolve найдем напряжение на выходе.

```
> eq1:= 0=-int((V1(t)-V2(t)),t)/
L+C*diff(V2(t),t)+V2(t)/R;
V1(t):=1;
eq1:=0=
```

$$= -\left(\frac{1}{L} \int V1(t) - V2(t) dt \right) +$$

$$+ C \left(\frac{d}{dt} V2(t) \right) + \frac{V2(t)}{R}$$

```
> dsolve(eq1, V2(t));
```

$$V2(t) = e^{\left(\frac{(L - \sqrt{L^2 - 4R^2LC})t}{2LRC} \right)} _C2 +$$

$$+ e^{\left(\frac{(L + \sqrt{L^2 - 4R^2LC})t}{2LRC} \right)} _C1 + 1$$

Выделение числителя и знаменателя

```
> restart;
> S:=solve(a*x+y+b*x+c*y+e, y);
> y:=S;

$$y := -\frac{bx + e}{ax + c}$$

```

```
> numer(S); # выделение числителя
-bx - e
```

```
> denom(S); # выделение знаменателя
ax + c
```

Операции оценивания

Функция evalf() вычисляет в виде десятичных дробей величины, заданные точно:

```
> evalf(sqrt(2)); # десять знаков по
умолчанию
1.414213562
```

Число знаков можно изменить, если до вычисления выполнить команду Digits:=n, либо добавить в команду оценки опцию

```
> evalf(sqrt(2),15); # 15 знаков
1.41421356237310
```

Если аргументы функций заданы в десятичной форме, оценка происходит автоматически:

```
> sqrt(2.0);
1.414213562
```

```
> sin(1.0);
0.8414709848
```

Родственные команды оценивания: evalm, evalhf, evalb, evala.

Математические константы

```
> evalf(Pi,50); # число Pi
3.141592653589793238462
6433832795028841971693993751
> evalf(exp(1),20); # число e
2.7182818284590452354
```

```
> sqrt(-1); # мнимая единица
> infinity; # бесконечность
```

```
I
∞
> # END
```

ЛИТЕРАТУРА:

1. В. П. Дьяконов. Maple-9 в математике, физике, образовании. — М.: Солон-Пресс, 2004.

Продолжение следует

Олег Петраков,
г. Москва

Среда разработки Visual DSP++

Это заключительная часть цикла публикаций. В ней рассматриваются примеры программ на языке программирования C и их выполнение в среде разработки Visual DSP++.

ПРИМЕРЫ ПРОГРАММ

Кроме ассемблерных, среда разработки Visual DSP++ позволяет транслировать программы, написанные на языке программирования C. Рассмотрим пример простой программы на этом языке, которая вычисляет значения тригонометрической функции синуса для четырех заданных значений аргумента.

Создайте новый проект с именем «C» и подключите к проекту новый файл с именем «c1.c», набрав в нем текст программы, приведенный ниже:

```
#include <math.h>
#define PI 3.14159 /* Число Пи */
int y1,y2,y3,y4;
main()
{
    y1 = sin(0);
    y2 = sin(PI/2.0);
    y3 = sin(PI);
    y4 = sin(PI+PI/2.0);
}
```

Здесь используются стандартные директивы и операторы языка C. В начале программы оператор включения подключает библиотеку математических функций math.h. Далее производится определение символического имени PI. Ниже объявляются целочисленные переменные y1—y4. В данном примере переменные объявлены целочисленными для наглядности представления результата работы программы. Главный цикл программы main состоит из четырех однотипных операций вычисления значений функции синуса для различных значений аргумента. В данном случае аргумент принимает значения, кратные PI/2. Результатом программы будут значения функции синус, которые очевидно должны быть 0, 1, 0 и -1.

Оттранслируйте программу, нажав клавишу F7. Затем выполните ее, нажав клавишу F5. После выполнения программы в правой части экрана откроется окно дизассемблера, в котором будет присутствовать текст программы на языке ассемблера. Этот

текст позволяет увидеть соответствие операторов языка C набору ассемблерных инструкций. Для просмотра результата работы программы необходимо открыть окно памяти данных с помощью команды Memory->Data. По умолчанию в этом окне данные отображаются в шестнадцатеричном виде. Для отображения результатов в знаковом целочисленном виде щелкните на окне правой кнопкой мыши и в открывшемся контекстном меню (рис. 1) выберите формат Select Format->Signed Integer.

В результате данные в окне будут представлены в более привычном виде (рис. 2).

Как видно, результат работы программы оказался абсолютно точен.

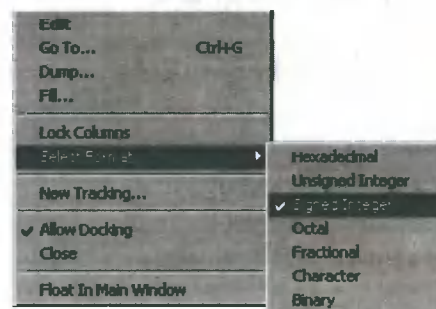


Рис. 1

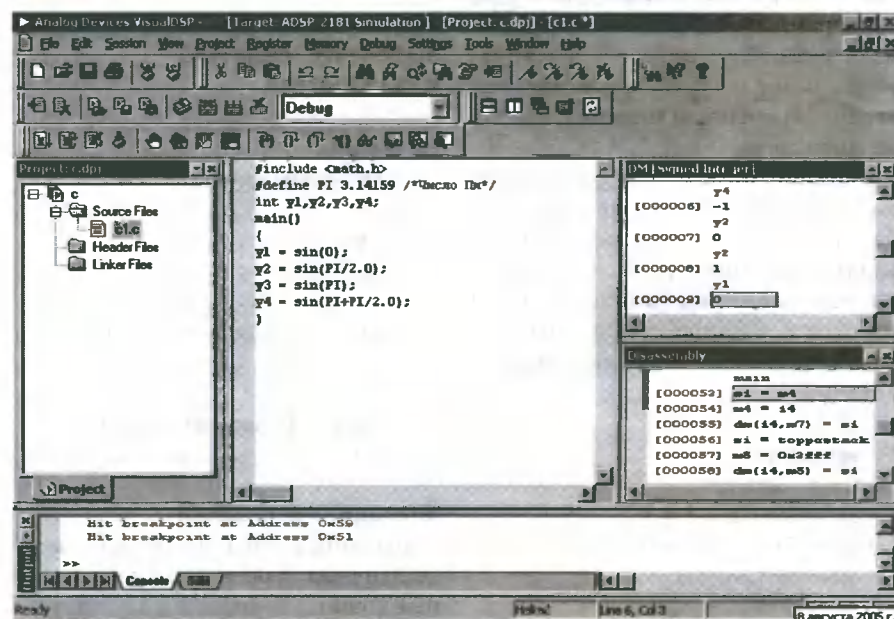


Рис. 2

Заметьте, что благодаря лаконичности записи, присущей языку высокого уровня C, программа для вычисления тригонометрических функций состоит всего лишь из несколько строк. Для написания подобной программы на ассемблере понадобилось бы несколько десятков строк.

Теперь немного усложним программу, изменив ее текст на тот, что приведен ниже:

```
#include <math.h>
#define PI 3.14159 /* Число Пи */
#define D PI/180.0 /* Число Дельта */
#define N 100 /* Число отсчетов */
float y1[N], y2[N];
main()
{
    int i;
    for (i=0; i<N; i++)
    {
        y1[i] = sin(10.0*i*D);
        y2[i] = cos(10.0*i*D);
    }
}
```

Здесь определены два массива y1 и y2 с данными в формате float. Значения этих массивов будут вычисляться в цикле for для N значений аргумента функций синуса и косинуса.

Выполните трансляцию программы и установите точку останова на последней строчке программы, после этого запустите программу на выполнение. Через несколько секунд выполнение программы завершится. Для просмотра результата ее работы откройте новое окно графопостроителя. Для этого выполните команду View->Debug Windows->Plot->New. В

открывшемся окне задайте значения всех полей в соответствии с рис. 3 и нажмите программную кнопку Browse.



Рис. 3

В новом открывшемся окне (рис. 4) найдите и выделите в памяти данных строку с массивом y1, после чего нажмите кнопку OK.

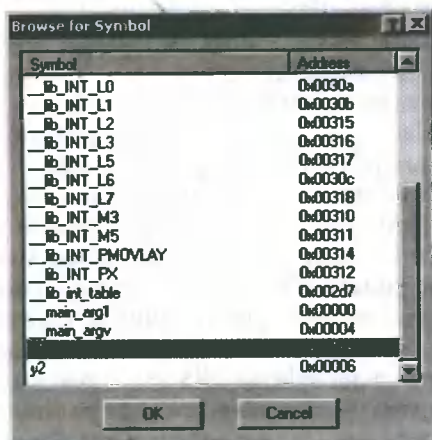


Рис. 4

Нажмите в окне Plot Configuration программную кнопку Add. При этом в поле Data sets появится запись Data set1. Теперь еще раз нажмите программную кнопку Browse и выделите в памяти данных строку с массивом y2, после чего нажмите кнопку OK. Вновь нажмите в окне Plot Configuration программную кнопку Add. После этих действий в поле Data sets появится новая запись Data set2. Убедитесь, что строке Data set1 соответствует адрес y1, а строке Data set2 — адрес y2, выделяя по очереди каждую из строк в окне Plot Configuration. Для обеих строк данные должны иметь тип float. Если это не так, исправьте формат данных на float.

Теперь нажмите в окне кнопку OK разверните окно с графиками результатов работы программы (рис. 5).

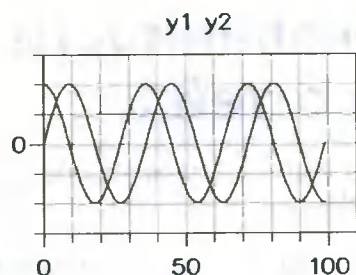


Рис. 5

Как видно из рисунка, графики представляют собой кривые функций синуса и косинуса, полученные в результате работы программы.

С помощью графопостроителя можно увидеть и спектр сигналов. Для этого необходимо щелкнуть правой кнопкой мыши по окну с графиками и в открывшемся контекстном меню выбрать команду Modify Setting. Далее в открывшемся окне Plot Setting открыть закладку Data Processing и выделить в поле Data Process строку FFT Magnitude (рис. 6).

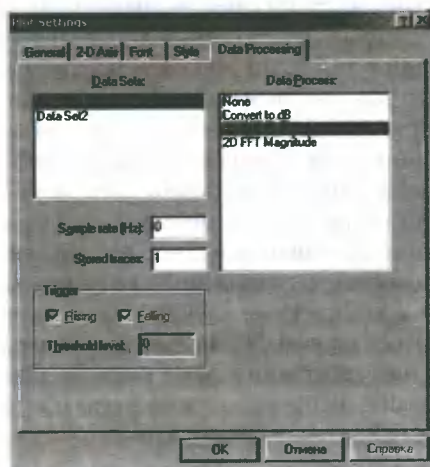


Рис. 6

После этого закрыть окно, нажав кнопку OK. В результате график преобразуется к виду, приведенному на рис. 7.

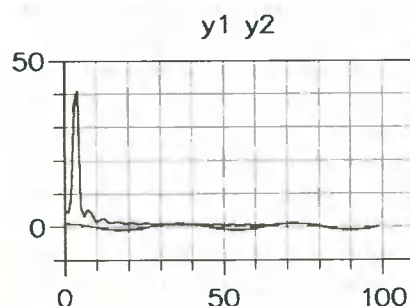


Рис. 7

Здесь отображены одновременно функция косинуса и спектр функции синуса.

Подобным образом можно производить вычисление и других математических функций, для которых существуют готовые библиотеки в трансляторе языка C. Среда разработки Visual DSP++ содержит ряд других примеров, в том числе на языке C, изучив которые можно расширить объем знаний в этой области. Кроме того, встроенная справочная система среды разработки дает большой объем дополнительной информации по многим вопросам, возникающим в процессе программирования.

ЗАКЛЮЧЕНИЕ

Этой статьей заканчивается цикл публикаций, посвященных сигнальному процессору ADSP2181 и средствам его программирования. Данная публикация позволяет пройти путь изучения сигнального процессора от самых основ до самостоятельного программирования и разработки на его основе устройств с цифровой обработкой сигнала. В ней были изложены как схемотехнические решения, так и программные материалы. Однако возможности сигнальных процессоров фирмы Analog Devices, а также средств разработки Visual DSP++ не исчерпываются приведенными в рамках статей материалами и примерами. Это связано не только с краткостью изложения материала, но и с непрерывным развитием как самих процессоров, так и программных средств, для них. Появление на рынке электроники новых процессоров приводит к необходимости постоянного обновления и углубления знаний в этой области. Надеюсь, что изложенный материал уже помог и еще поможет тем людям, которые захотят заняться изучением и применением сигнальных процессоров в своей практике.

Олег Вальпа,
г. Миасс Челябинской обл.

открывшемся окне задайте значения всех полей в соответствии с рис. 3 и нажмите программную кнопку Browse.

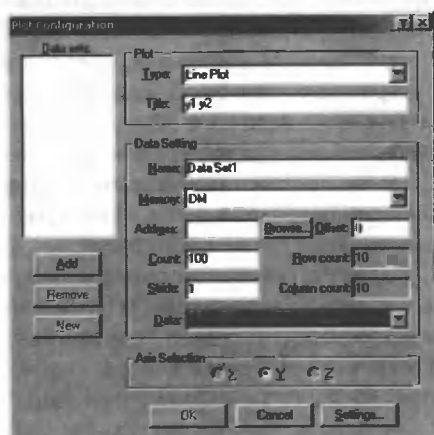


Рис. 3

В новом открывшемся окне (рис. 4) найдите и выделите в памяти данных строку с массивом y1, после чего нажмите кнопку OK.

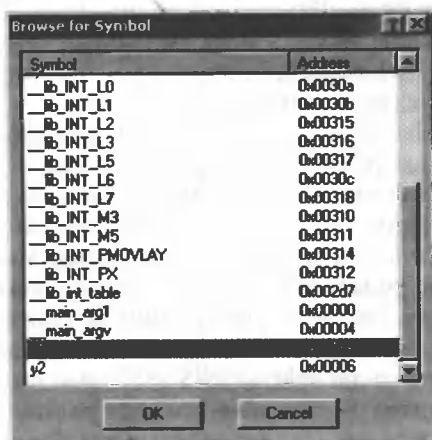


Рис. 4

Нажмите в окне Plot Configuration программную кнопку Add. При этом в поле Data sets появится запись Data set1. Теперь еще раз нажмите программную кнопку Browse и выделите в памяти данных строку с массивом y2, после чего нажмите кнопку OK. Вновь нажмите в окне Plot Configuration программную кнопку Add. После этих действий в поле Data sets появится новая запись Data set2. Убедитесь, что строке Data set1 соответствует адрес y1, а строке Data set2 — адрес y2, выделяя по очереди каждую из строк в окне Plot Configuration. Для обеих строк данные должны иметь тип float. Если это не так, исправьте формат данных на float.

Теперь нажмите в окне кнопку OK разверните окно с графиками результатов работы программы (рис. 5).

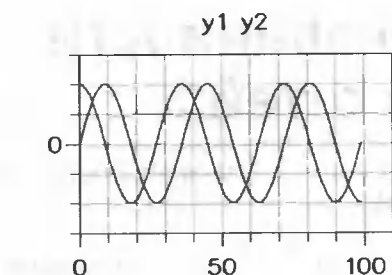


Рис. 5

Как видно из рисунка, графики представляют собой кривые функций синуса и косинуса, полученные в результате работы программы.

С помощью графопостроителя можно увидеть и спектр сигналов. Для этого необходимо щелкнуть правой кнопкой мыши по окну с графиками и в открывшемся контекстном меню выбрать команду Modify Setting. Далее в открывшемся окне Plot Setting открыть закладку Data Processing и выделить в поле Data Process строку FFT Magnitude (рис. 6).

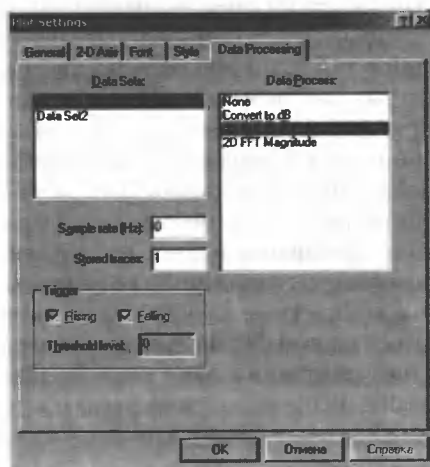


Рис. 6

После этого закрыть окно, нажав кнопку OK. В результате график преобразуется к виду, приведенному на рис. 7.

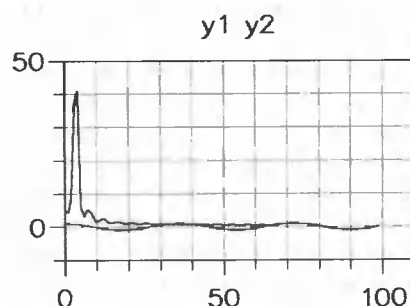


Рис. 7

Здесь отображены одновременно функция косинуса и спектр функции синуса.

Подобным образом можно производить вычисление и других математических функций, для которых существуют готовые библиотеки в трансляторе языка C. Среда разработки Visual DSP++ содержит ряд других примеров, в том числе на языке C, изучив которые можно расширить объем знаний в этой области. Кроме того, встроенная справочная система среды разработки дает большой объем дополнительной информации по многим вопросам, возникающим в процессе программирования.

ЗАКЛЮЧЕНИЕ

Этой статьей заканчивается цикл публикаций, посвященных сигнальному процессору ADSP2181 и средствам его программирования. Данная публикация позволяет пройти путь изучения сигнального процессора от самых основ до самостоятельного программирования и разработки на его основе устройств с цифровой обработкой сигнала. В ней были изложены как схемотехнические решения, так и программные материалы. Однако возможности сигнальных процессоров фирмы Analog Devices, а также средств разработки Visual DSP++ не исчерпываются приведенными в рамках статей материалами и примерами. Это связано не только с краткостью изложения материала, но и с непрерывным развитием как самих процессоров, так и программных средств, для них. Появление на рынке электроники новых процессоров приводит к необходимости постоянного обновления и углубления знаний в этой области. Надеюсь, что изложенный материал уже помог и еще поможет тем людям, которые захотят заняться изучением и применением сигнальных процессоров в своей практике.

Олег Вальпа,
г. Миасс Челябинской обл.

Национальные особенности технологических сетей...

В статье рассмотрены некоторые вопросы проектирования и построения встраиваемых систем.

Многие современные электронные приборы сейчас «разумны», предельно интерактивны, удобны в управлении и насыщены всевозможными функциями — нужными и не очень. Все это стало возможным благодаря системам управления на микроконтроллерах или, как они правильно называются, встраиваемым системам (Embedded Systems). Ядром этих систем, как правило, является микроконтроллер (МК) — настоящий компьютер в одной микросхеме с памятью, интерфейсами и периферией.

Техническая эволюция не останавливается, и сейчас ни потребители, ни разработчики не представляют себе сколько-нибудь сложной системы без возможностей работы в сети. Век обособленных stand-alone устройств закончился, и нужно, чтобы приборы могли «общаться» друг с другом и с «большими» компьютерами. Если в секторе потребительской техники этот вопрос еще как-то решается, в промышленно-технологических и научных областях применения «умных» приборов все не так радужно. И это вполне закономерно — сложные системы требуют сложных решений в сетевой конфигурации. Такие системы в силу своей сложности требуют особого подхода при проектировании и повышенного внимания разработчика.

Попытаемся рассмотреть с точки зрения разработчика или системщика наиболее сложные проблемы построения встраиваемых систем, функции которых так или иначе зависят от эффективности и безошибочности работы в сети. Под сетями будем подразумевать не только компьютерные вычислительные сети, но и все разнообразие технологических сетей.

КАКУЮ ТЕХНОЛОГИЧЕСКУЮ СЕТЬ ВЫБРАТЬ?

На сегодняшний день даже за рубежом не приняты единые стандарты на технологические и промышленные сети, так называемые fieldbus. Ситуация такова, что крупные производители промышленного оборудования продвигают свои коммерческие решения, например, PROFIBUS, ModBus,

LONworks и подобные. Временами корпорации объединяются в консорциумы и пытаются, иногда успешно, придать своим разработкам статус стандарта. В результате эти стандарты в оборудовании используют только сами фирмы-разработчики, и в реальности мы можем наблюдать настоящий технологический «зоопарк». Получается, что общего и действительно рабочего стандарта, аналогичного Ethernet в ЛВС, до сих пор не существует. Часто разработчик, изучив «фирменные» стандарты и примерив их к своему прибору, находит их непригодными или чрезмерно сложными и дорогими в реализации, и начинает изобретать свой «велосипед», т. е. свою, часто ни с чем не совместимую технологическую сеть, свои протоколы и т. п. Все это действительно в том случае, если в техническом задании прямо не указано, что ему требуется обеспечить совместимость с какой-либо стандартной сетью. Если разработчик все-таки использует стандартные протоколы и сети, он должен помнить, что стандарт — не панацея, и в конкретном случае может оказаться малоэффективным средством, а при адаптации к нему «железа» породить непредсказуемые неувязки и ошибки.

КУЛИБИНЫ И ВЕЛОСИПЕДЫ...

Особенно тяжелая ситуация сложилась у российских разработчиков и производителей аппаратуры. На примере наших изобретателей можно наблюдать все многообразие негативных факторов, приводящих в итоге к нестабильным системам, логически не согласованным, разваливающимся при изменении конфигурации или попытке модернизации. Так, промышленные контроллеры-теплосчетчики одной известной фирмы прекрасно работали в конфигурации «один ведущий — много ведомых» по физическим интерфейсам RS-232/485, но когда разработчик, сделав специальный мост RS-232—Ethernet, перешел на технологии UDP/IP по Ethernet, пользователи столкнулись с несовершенством протокола на прикладном уровне или, скорее, его естественную

неуниверсальность. Получалось, что данные от контроллера, передаваемые, практически без изменения в пакетах UDP, путались со штампами времени, в результате чего диспетчерская программа вместо технологических данных показывала какую-то абстракцию, что было вполне предсказуемо — ведь UDP не гарантирует доставку данных и порядок пакетов. Если бы разработчик учел это и хотя бы пронумеровал пакеты от контроллера, такого бы не случилось.

Изобретатели «велосипедов» или технологических сетей (ТС) и протоколов будут всегда, в силу причин, часть которых изложена в статье, потому хочется сказать несколько слов о том, как сделать хороший, добротный велосипед на классических двух колесах, а не чудо техники с одним овальным колесом и тремя звонками...

При разработке ТС всегда стоит брать во внимание физический интерфейс (И) и помнить, что разные типы интерфейсов подразумевают разные области применения, не стоит их смешивать без особой надобности. Например, I²C, SPI — межмодульный интерфейс, CAN удобен для связи с несложными удаленными контроллерами и для расширения основного блока субмодулями сбора данных и управления, а вот использовать его для передачи больших объемов прикладных данных неудобно, здесь подойдет Ethernet. RS-485 и токовая петля — хорошее решение для небольшой сети устройств, а для больших проектов неплохо было бы продумать иерархию.

Придумывая свой протокол, нужно опираться на свойства физического и канального, сетевого уровня интерфейса, не забывая о топологии сети. Для примера — реализовать топологию одноранговой сети на интерфейсе RS-485 и подобном будет проблематично, тут возникает задача арбитража устройств или разделения доступа к среде. Конечно, можно попробовать вариации на тему маркерного кольца или CSMA/CD, но будет ли это оправдано?

В своей реализации протокола надо постараться не забывать про уровни модели BOC. Хотя бы разделите протокол на три уровня — физический, канальный и сетевой прикладной. Кроме четких разделений услуг уровней это даст некоторые степени свободы при совершенствовании устройства, многое встанет на свои места.

При проектировании сложной системы иерархия в коммуникациях просто необходима.

Пусть submodule сбора данных обмениваются с контроллером по своему, возможно упрощенному протоколу, контроллеры с «суперконтроллером» — по каналам технологических сетей, а «суперконтроллер» общается с диспетчерских компьютеров по сетям ЛВС.

Профессионалы могут придумать — дескать Ethernet вкупе с IP не очень хорошо подходит для систем реального времени, но если структура системы действительно хорошо продумана, это несущественно.

Если имеются унаследованные устройства со своим, ни с чем не совместимым интерфейсом или протоколом, хорошим решением будет создание «интеллектуального» прибора — шлюза, преобразующего этот интерфейс во что-то удобное и понятное. Прибор этот при разумном проектировании позволит нивелировать многие недостатки старого устройства. Решение, возможно, не очень изящное и дешевое, зато действенное.

При использовании в качестве транспорта каналов общего пользования (ЛВС на Ethernet, радиолинки, Интернет) не стоит забывать про защиту данных от просмотра, фальсификации и т. п., здесь лучшее решение — проверенная временем криптография. Есть достаточно простые и эффективные алгоритмы, которые неплохо работают даже на восьмьбитных микроконтроллерах. А если учитывать, что современные МК имеют довольно мощные «числодробилки», криптография не слишком усложнит проект.

ДЯДЮШКИНО НАСЛЕДСТВО

В предыдущем примере можно увидеть корни и другой проблемы, с которой часто сталкиваются отечественные разработчики — наследованная аппаратура. Это могут быть серьезные приборы, которые разрабатывали могучие КБ, а также «наколенные» подделки, доставшиеся разработчику от его коллеги-предшественника. Этим устройствам позволено быть несовершенными и устаревшими, но в силу многих причин они часто не имеют достойной замены.

И тут, как назло, возникает задача — адаптировать эти приборы, порою ценой полной переделки, под современные требования заказчика и

связать их с компьютером. Тут и начинается самое интересное — выясняется, что у прибора отсутствуют средства коммуникаций или есть какой-то допотопный GPIB или «стык С2» и совершенно непонятный протокол, а документация полностью отсутствует. Разработчику практически приходится проводить настоящий реинжиниринг внутренностей прибора, чтобы добавить нужную функцию. В результате, разработчику кажется, что ничего хорошего все равно не получится, и он идет самым легким путем — путем минимального вмешательства и вынуждено неуклюжих доработок, лишь бы это все заработало, а дальше будь, что будет, т. е. создает все условия для повторения ситуации. Автору доводилось работать с контроллерами для диспетчеризации объектов ЖКХ, которые по специальной сигнальной линии передавали центру свое состояние, используя для этого «телеметрический» протокол (односторонний, только для приема данных от устройств). Протокол отличался повышенной помехоустойчивостью за счет большой избыточности корректирующих данных и очень маленькой скорости обмена. Но вот заказчику захотелось переделать систему связи — реализовать возможность централизованного опроса контроллеров. И тут стало ясно, что маленькая скорость и большое число лишних данных приводят к падению производительности обновленной системы, в которой в случае ошибок обмена можно просто «переспросить» контроллер. Но уже ничего не поделаешь, контроллеров выпущено очень много и пришлось пойти на некие технические компромиссы в виде гибридного протокола и полуаппаратной схемы опроса. А как бы все упростилось, если бы возможность двустороннего обмена была предусмотрена ранее...

ПРОДУМАННЫЙ ПРОЕКТ — ХОРОШИЙ ПРИБОР

Самый интересный и творческий этап работы разработчика — составление технического задания и проектирование прибора с самого начала. Но это и самая ответственная часть работы — любая ошибка или недоработка, воплощенная в готовой конструкции, порой обходится очень дорого. Часто при проектировании прибора сетевые средства рассматриваются как дополнительная, факультативная функция или ей вообще придают статус — «на

хорошее будущее», т. е. есть порт RS-485 и ладно, потом придумаем, как и чем по нему будем обмениваться. А когда наступает это самое «потом», становится ясно, что интегрировать устройство в сеть и обеспечить его корректную работу практически невозможно. Автор был свидетелем случая, когда разработчик отладил макет блока для управления двигателем постоянного тока, в котором параметры задавались аналоговыми сигналами, но заказчик вспомнил, что неплохо бы управлять блоком по цифровому каналу. Переделки серьезно задержали выпуск готового прибора — разработчик уже и так использовал практически все ресурсы слабенького микроконтроллера, уложив буквально по тактам микропрограмму, к тому же заказчик так и не смог четко сформулировать, как именно он хочет управлять прибором.

Сетевое взаимодействие с самого начала нужно продумывать до мелочей и, учитывая все факторы, проектировать функции и режимы прибора именно в разрезе того, как он будет работать в сети. На этом этапе всегда есть возможность подстраховаться и, разумно планируя «линию жизни» устройства, обеспечить на будущее резерв функциональности пусть даже путем дополнительных технических и материальных затрат. В дальнейшем, это обязательно окупится.

IP И ETHERNET В КАЖДОМ УСТРОЙСТВЕ

Сейчас — время Интернета, а самая распространенная сетевая технология — Ethernet. Так что вполне естественно, что разработчику или заказчику может потребоваться встроить в прибор интерфейс Ethernet и стек протоколов TCP/UDP/IP. Существует несколько технических решений этой задачи:

- взять микросхему Ethernet-контроллера, используемую в сетевых картах для ПК (обычно для шины ISA), затем реализовать программно стек IP в firmware микроконтроллера. При внешней простоте это решение осложняется проблемами с адаптацией и отладкой протоколов стека, трудностями приобретения микросхем;
- купить готовый модуль преобразователя интерфейса RS-232 — Ethernet (X-Port и т. п.), но это решение явно не дешево и, скорее всего, не сможет обеспечить до-

статочной гибкости и всех технических требований;

- наверняка не последнее, но проверенное автором, простое и достаточно дешевое — микросхема W3100A «Fully Hardwired TCP/IP Chip», как называет ее производитель (фирма WIZnet Inc., www.iinchip.com).

Об этом решении хотелось бы рассказать подробнее. Микросхема W3100A представляет собой аппаратный TCP/IP стек, предназначенный для работы в паре с микросхемой физического уровня PHY RTL8201BL в проводных сетях Ethernet. Микросхема легко подключается практически к любому микроконтроллеру по параллельному интерфейсу (шина адреса, шина данных, стробы чтения/записи) и поддерживает несколько режимов синхронизации обмена с МК, а также может работать по шине I²C, несколько не теряя функциональности. Полный UDP/TCP/IP стек, четыре сокета-канала, два буфера по 8 Кб на прием и передачу, поддерживаются субрежимы IP Layer RAW и MAC Layer RAW. Максимальная скорость обмена по сети в канале МК — Ethernet до 6 Мбит/с (зависит от режима обмена с МК) при скоростях Ethernet 10 и 100 Мбит/с. Встроенный ответчик на ICMP (ping) запросы, толерантность входов и выходов к логическим уров-

ням схем с питанием 5 В. Минимум дополнительных элементов и приемлемая цена (около 250 руб., не считая развязывающего трансформатора и разъема RJ45) вместе с микросхемой PHY, существуют варианты трансформатор и разъем, которые, например, производит фирма Speed-Tech, www.stech.com.tw. Что еще надо разработчику? Для любителей макетирования или поклонников модульности фирма WIZnet предлагает также готовые и относительно недорогие платы-модули очень скромных размеров, где уже смонтировано все, что нужно. Конечно, в аппаратном стеке протоколов могут быть ошибки, которые способен исправить только производитель, выпустив новую ревизию чипа, но, как показывает практика, таковых нет. А существующие в текущей версии чипа недочеты можно назвать особенностями реализации.

ВСЕ ТОЛЬКО НАЧИНАЕТСЯ

В области коммуникаций встраиваемых систем российские разработчики как всегда идут своим путем, и все негативные примеры — свидетельство проблем роста, часто связанных с нехваткой нужных специалистов, а не с недостатком профессионализма. Отечественные фирмы, делающие по-настоящему хорошие и серьезные приборы, вынуждены экономить и

часто решают вопросы связи приборов силами штатных инженеров, пусть способных, но не «профильных». В отрасли явно ощущается недостаток людей, способных вести комплексные Embedded проекты.

Автор хотел бы помочь советами и предостеречь от банальных ошибок разработчиков, начинающих работать в этой интересной и сложной области. Как говорится — «предупрежден, значит вооружен».

ЛИТЕРАТУРА:

1. Г. Олссон, Д. Пиани. *Цифровые системы автоматизации и управления*. — С.-Пб.: Невский Диалект, 2001.
2. В. Г. Олифер, Н. А. Олифер. *Компьютерные сети. Принципы, технологии, протоколы*. — С.-Пб.: Издательство «Питер», 2000.
3. В. Б. Бродин, А. В. Калинин. *Системы на микроконтроллерах и БИС программируемой логики*. — М.: Издательство «ЭКОМ», 2002.
4. П. Хоровиц, У. Хилл. *Искусство схемотехники. В 3-х томах: том 2, пер. с англ.* — 4-е изд. — М.: Мир, 1993.

Роман Брыгин,
г. Новосибирск

Автономный считыватель штрихкода

Данная статья содержит информацию о штрихкоде и идею по созданию портативного устройства для его считывания и расшифровки.

В настоящее время штрихкод наносится практически на все производимые товары. Его можно встретить на упаковке товара практически в любом магазине или супермаркете, он позволяет ускорить операции по учету и продаже. С помощью штрихкода зашифрована информация о некоторых наиболее существенных параметрах продукции, например, о стране происхождения, производителе, классе товара и т. д. По нему можно, кроме того, определить законность изготовления продукции.

Как правило, в залах супермаркетов имеются специальные места для считывания штрихкода покупателями

еще до прохождения через кассу, но таких мест обычно немного, и порой их нелегко найти, а изучить товар подробнее требуется до его приобретения. Вот тут на выручку покупателям могли бы прийти портативные считыватели штрихкодов, которые можно было бы носить с собой во время закупок.

Идея автора состоит в предложении разработать портативное автономное устройство для считывания штрихкода. Такое устройство должно иметь небольшие размеры и вес, малое потребление энергии и автономное питание. Считыватель должен иметь жидкокристаллический

дисплей с возможностью отображения на нем алфавитно-цифровой информации. Он должен быть прост и удобен в эксплуатации. За основу считывателя можно взять, например, оптический манипулятор-мышь, доработав его соответствующим образом. Возможно, считыватель можно будет оформить в виде авторучки и т. п. Для упрощения задачи цифры штрихкода можно вводить вручную с помощью кнопок.

Ниже приводится пример конкретного штрихкода и расшифровка информации, которую он несет. Наиболее распространены американский универсальный товарный код UPC и европейская система кодирования EAN. Согласно той или иной системе, каждому виду изделия присваивается свой номер, который содержит обычно 13 цифр (EAN-13).

Рассмотрим штрихкод, изображенный на рис. 1 и имеющий цифровой эквивалент 3000376211205.



Рис. 1

Таблица 1

Код	Страна
00—09	США и Канада
30—37	Франция
380	Болгария
383	Словения
385	Хорватия
400—440	Германия
460—469	Россия и бывший СССР
4605	Латвия
471	Тайвань
489	Гон-Конг (ныне Китай)
45, 49	Япония
50	Великобритания
520	Греция
529	Кипр
535	Мальта
539	Ирландия
54	Бельгия и Люксембург
560	Португалия
569	Исландия
57	Дания
590	Польша
599	Венгрия
600, 601	ЮАР
619	Тунис
64	Финляндия
690	Китай
70	Норвегия
729	Израиль
73	Швеция
740	Гватемала
741	Сальвадор
742	Гондурас
743	Никарагуа
744	Коста-Рика
745	Панама
750	Мексика
759	Венесуэла
76	Швейцария
770	Колумбия
773	Уругвай
775	Перу
779	Аргентина
780	Чили
786	Эквадор
789	Бразилия
80—83	Италия
84	Испания
850	Куба
859	Чехия, Словакия
860	Югославия
869	Турция
87	Нидерланды
880	Южная Корея
885	Таиланд
888	Сингапур
90, 91	Новая Зеландия
955	Малайзия



Рис. 2

30 — две первые цифры означают страну происхождения (изготовителя или продавца) продукта;

00376 — следующие пять — предприятие-изготовитель;

21120 — еще пять — наименование товара, его потребительские свойства, размеры, массу, цвет;

5 — последняя цифра — контрольная, используемая для проверки правильности считывания штрихов сканером.

Согласно европейской системе кодирования EAN-13:

1 — код страны,

2 — код изготовителя,

3 — код товара,

4 — контрольная цифра,

5 — знак товара, изготовленного по лицензии.

Для вычисления контрольной цифры необходимо провести следующие вычисления.

1. Сложить цифры, стоящие на четных местах — $0 + 0 + 7 + 2 + 1 + 0 = 10$.

2. Полученную сумму умножить на 3 — $10 \times 3 = 30$.

3. Сложить цифры, стоящие на нечетных местах, кроме контрольной цифры — $3 + 0 + 3 + 6 + 1 + 2 = 15$.

4. Сложить полученные в пунктах 2 и 3 числа — $30 + 15 = 45$.

5. Отбросить десятки — из 45 получим 5.

6. Из числа 10 вычесть полученную в пункте 5 цифру — $10 - 5 = 5$.

Полученная цифра должна совпадать с контрольной штрихового кода, что говорит о подлинности товара. Если полученная цифра не совпадает с контрольной — это означает, что товар произведен незаконно.

Возможен также вариант, когда для кода страны-изготовителя отводится три знака, а для кода предприятия — четыре. Товары, имеющие большие размеры, могут иметь

короткий код, состоящий из восьми цифр — EAN-8.

Как правило, код страны присваивается Международной ассоциацией EAN. Код страны никогда не состоит из одной цифры.

Нередко на товаре можно увидеть надпись, например, «Сделано в Германии», а код, нанесенный на этикетку, этой стране не соответствует. Для подлинного товара причин может быть несколько:

- фирма была зарегистрирована и получила код не в своей стране, а в той, куда направлен основной экспорт ее продукции;
- товар был изготовлен на дочернем предприятии;
- товар был изготовлен в одной стране, но по лицензии фирмы из другой страны;
- учредителями предприятия являются несколько фирм из различных государств.

В табл. 1. приведены коды некоторых стран.

В Интернете по адресу <http://drako.com.ua> можно скачать свободно распространяемую программу «BarCow» для проверки штрихкодов. Эта программа проверяет подлинность кода и определяет страну изготовителя товара, она поддерживает следующие стандарты — EAN-13, EAN-8, 12-UPC-E и 8-UPC-A. Эта программа может помочь в отладке разрабатываемого считывателя штрихового кода.

Результат работы данной программы при проверке рассмотренного выше штрихкода представлен на рис. 2.

Олег Вальпа,
г. Миасс Челябинской обл.

Использование ЖКИ модулей

В настоящее время модули ЖКИ находят широкое применение в различных устройствах, поскольку они представляют собой готовые и недорогие технические решения. Это позволяет экономить время и ресурсы при проектировании различных устройств. В данной статье описываются некоторые подходы к использованию ЖКИ модулей с кириллицей.

Кодировка английского шрифта полностью соответствует кодировке ANSI. Русифицированная часть содержит только те символы, которые не имеют аналогов в английском шрифте. Таким образом, кодировка русского шрифта не соответствует кодировке ANSI, поэтому при составлении сообщения на русском языке для его вывода на ЖКИ текст

Большинство символьных (алфавитно-цифровых) модулей ЖКИ реализуется на основе контроллера HD44780. Данный контроллер может управлять двумя строками по 40 символов в каждой (для модулей с четырьмя строками по 40 символов используются два однотипных контроллера) при матрице символа 5×7 точек. Существует большая номенклатура ЖКИ модулей, которая по объему выводимой информации делится в соответствии с числом символов и строк на 8×2, 16×1, 16×2, 16×4, 20×1, 20×2, 20×4, 24×2, 40×2, 40×4, правда, можно встретить и другие комбинации символов и строк.

Необходимо отметить, что модули ЖКИ одного и того же формата могут отличаться по размерам печатной платы модуля, расположению контактов для подключения ЖКИ к микроконтроллеру, цветом ЖК панели и др., поэтому при проектировании печатной платы и корпуса устройства необходимо заранее изучить и выбрать подходящий модуль ЖКИ по его конструктивным особенностям.

Основную информацию о модуле ЖКИ можно узнать по его обозначению. Автор широко использовал индикаторы фирмы Sunlike. В табл. 1, 2 показана система обозначений для ЖКИ этой фирмы.

Система обозначений модулей ЖКИ других производителей может отличаться.

Как видно из данного обозначения, модули ЖКИ могут иметь различные наборы символов, в каждом из которых используется английский шрифт

Таблица 1

Обозначение	Поз.
S	1
C	2
1602	3
H	4
S	5
L	6
B	7
—	°
XH	9
—	°
H	9
S	10

Таблица 2

Позиция	Код	Описание
1. Производитель	S	SUNLIKE DISPLAY
2. Тип дисплея	C	Знакосинтезирующий
	B	Графический
	D	Заказной
3. Разрешение	—	Число символов×строк
4. Серия	A—Z	—
5. Тип ЖК панели	N	Позитивный TN
	M	Негативный TN
	F	Позитивный FSTN
	W	Негативный FSTN
	S	Позитивный STN серый
	Z	Позитивный STN серый
	T	Позитивный STN желто-зеленый
	U	Позитивный STN желто-зеленый
	V	Негативный STN синий
	L	LED — желто-зеленый
6. Подсветка, светодиодная — LED, электролюминесцентная — EL, люминесцентная лампа — CCFL	M	LED — янтарный
	N	LED — красный
	O	LED — оранжевый
	R	LED — зеленый
	E	EL — белый
	B	EL — зелено-голубой
	P	EL — зеленый
	C	CCFL — белый
	B	6:00
	T	12:00
7. Угол зрения, °	XH, SH	английский и кириллица
	SO, SA, XA	английский и японский
	SB, XB	английский и европейский
8. Язык шрифта	G	Стандартный
	H	Расширенный
9. Температурный диапазон, °C	B	Черный
	S	Серебристый

Таблица 3

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0				0	1	2	3	4	5	6	7	8	9	A	B	C
1				!	1	A	Q	a	q		G	Y	ш	ц	ш	ш
2				"	2	B	R	b	r		E	б	у	ш	ш	ш
3				#	3	C	S	c	s		J	в	м	ш	ш	ш
4				\$	4	D	T	d	t		Z	г	ь	ш	ш	ш
5				%	5	E	U	e	u		I	ё	э	ш	ш	ш
6				&	6	F	V	f	v		Й	ж	ю	ш	ш	ш
7				'	7	G	W	w	w		Л	з	я	ш	ш	ш
8				<	8	H	X	x	x		П	и	«	ш	ш	ш
9				>	9	I	Y	y	y		У	й	»	ш	ш	ш
A				*	:	J	Z	z	z		Ф	к	«	ш	ш	ш
B				+	:	K	C	c	c		Ч	л	»	ш	ш	ш
C				,	<	L	F	f	f		Ш	м	»	ш	ш	ш
D				-	=	M	J	j	j		Ъ	н	»	ш	ш	ш
E				.	>	N	^	n	n		Ы	п	»	ш	ш	ш
F				/	?	O	-	o	o		Э	т	»	ш	ш	ш

и японские, европейские или русские символы.

В табл. 3 представлены коды символов для русифицированного ЖКИ (здесь строка — это старшая тетрада кода, столбец — младшая).

придется кодировать в соответствии с табл. 3.

Этот перевод может осуществляться программой, защитой в микроконтроллер, который управляет процессом вывода информации на индикатор. Данный подход имеет недостатки, которые заключаются в том, что сам процесс перекодировки будет занимать машинное время, а программа — объем памяти микроконтроллера. В случае перехода на микроконтроллер другого семейства программу нужно будет переделывать.

Для некоторых микроконтроллеров, например для PIC, каждому символу ANSI можно поставить в соответствие его код в формате ЖКИ, причем это соответствие можно оформить отдельным файлом, который будет подключаться к проекту. В процессе компиляции из этого файла для каждого символа будет извлекаться код в формате ЖКИ.

Пример такого файла — symbol.inc

```
;*****
```

```
A      EQU      H'41'
Б      EQU      H'A0'
B      EQU      H'42'
Г      EQU      H'A1'
Д      EQU      H'E0'
E      EQU      H'45'
Е      EQU      H'A2'

Я      EQU      H'B1'
```

В программе вывод на ЖКИ можно оформить следующим образом:

```
MOVLW  П
CALL   LCD
MOVLW  P
CALL   LCD
MOVLW  И
CALL   LCD
MOVLW  В
CALL   LCD
MOVLW  Е
CALL   LCD
MOVLW  Т
CALL   LCD
```

К сожалению, не все компиляторы поддерживают кириллицу в теле программы. Один из лучших компиляторов для микроконтроллеров семейства x51 Keil 3 — не исключение. В этом случае эффективным путем для решения проблемы будет представление каждого выводимого сообщения в виде закодированной в формате ЖКИ таблицы. Перевод сообщений можно осуществлять вручную, но если речь идет о большом числе сообщений, перевод превращается в рутинную работу. Для облегчения процесса составления сообщений на русском языке можно использовать программу, которая будет запускаться на ПЭВМ и преобразовывать сообщения в соответствии с табл. 3. Ниже представлена такая программа, которая написана на языке Delphi.

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes,
  Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, Mask;

type
  TForm1 = class(TForm)
    MaskEdit1: TMaskEdit;
    BitBtn1: TBitBtn;
    MaskEdit2: TMaskEdit;
```

```
Memo1: TMemo;
Memo2: TMemo;
MaskEdit5: TMaskEdit;
Label1: TLabel;
Label2: TLabel;
procedure BitBtn1Click(Sender:
TObject);
procedure Rusconv(symbol:char);
procedure strokaconv(stroka:
string);

private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;
  s, l, l2: string;

implementation

{$R *.DFM}
procedure TForm1.Rusconv(Symbol:char);
begin
  // замена русских символов на их англ.
  аналоги
  if Symbol='А' then s:=inttohex(ord('A'
),2)+'h' else
  if Symbol='Б' then s:=inttohex(ord('B'
),2)+'h' else
  if Symbol='В' then s:=inttohex(ord('C'
),2)+'h' else
  if Symbol='Г' then s:=inttohex(ord('E'
),2)+'h' else
  if Symbol='Д' then s:=inttohex(ord('H'
),2)+'h' else
  if Symbol='Е' then s:=inttohex(ord('K'
),2)+'h' else
  if Symbol='Ж' then s:=inttohex(ord('M'
),2)+'h' else
  if Symbol='З' then s:=inttohex(ord('O'
),2)+'h' else
  if Symbol='И' then s:=inttohex(ord('P'
),2)+'h' else
  if Symbol='Й' then s:=inttohex(ord('T'
),2)+'h' else
  if Symbol='К' then s:=inttohex(ord('X'
),2)+'h' else
  // перевод по таблице
  if Symbol='3' then s:='0A4h' else
  if Symbol='Л' then s:='0A7h' else
  if Symbol='М' then s:='0AEh' else
  if Symbol='Ж' then s:='0A3h' else
  if Symbol='П' then s:='0A8h' else
  if Symbol='Р' then s:='0A1h' else
  if Symbol='Б' then s:='0A0h' else
  if Symbol='У' then s:='0A9h' else
  if Symbol='Д' then s:='0E0h' else
  if Symbol='Ц' then s:='0E1h' else
  if Symbol='И' then s:='0A5h' else
  if Symbol='Ю' then s:='0B0h' else
  if Symbol='Й' then s:='0A6h' else
```

```
if Symbol='0' then s:='0AAh' else
if Symbol='Ч' then s:='0ABh' else
if Symbol='Ш' then s:='0ACH' else
if Symbol='Щ' then s:='0E2h' else
if Symbol='b' then s:='0ADh' else
if Symbol='W' then s:='0AEh' else
if Symbol='b' then s:='62h' else
if Symbol='3' then s:='0AFh' else
if Symbol='Я' then s:='0B1h' else
if Symbol='a' then s:=inttohex(ord('a'
),2)+'h' else
if Symbol='c' then s:=inttohex(ord('c'
),2)+'h' else
if Symbol='e' then s:=inttohex(ord('e'
),2)+'h' else
if Symbol='o' then s:=inttohex(ord('o'
),2)+'h' else
if Symbol='p' then s:=inttohex(ord('p'
),2)+'h' else
if Symbol='y' then s:=inttohex(ord('y'
),2)+'h' else
if Symbol='x' then s:=inttohex(ord('x'
),2)+'h' else
if Symbol='T' then s:='0BFh' else
if Symbol='n' then s:='0BEh' else
if Symbol='ч' then s:='0C0h' else
if Symbol='я' then s:='0C7h' else
if Symbol='и' then s:='0B8h' else
if Symbol='н' then s:='0BDh' else
if Symbol='к' then s:='0BAh' else
if Symbol='д' then s:='0E3h' else
if Symbol='ь' then s:='0C4h' else
if Symbol='б' then s:='0B2h' else
if Symbol='в' then s:='0B3h' else
if Symbol='г' then s:='0B4h' else
if Symbol='ж' then s:='0B6h' else
if Symbol='з' then s:='0B7h' else
if Symbol='й' then s:='0B9h' else
if Symbol='м' then s:='0BCh' else
if Symbol='л' then s:='0BBh' else
if Symbol='ф' then s:='0E4h' else
if Symbol='ц' then s:='0E5h' else
if Symbol='ш' then s:='0C1h' else
if Symbol='щ' then s:='0E6h' else
if Symbol='ь' then s:='0C2h' else
if Symbol='ы' then s:='0C3h' else
if Symbol='э' then s:='0C5h' else
if Symbol='ю' then s:='0C6h' else
showmessage('Don't have symbol '+symbol);
end;

procedure TForm1.strokaconv(stroka:
string);
var i: integer;
begin
  l:=''; l2:='';
  for i:=1 to Length(stroka) do begin
    if ord(stroka[i])<$7f then s:=inttohex
(ord(stroka[i]),2)+'h'
      else Rusconv(stroka[i]);
    l:=l+', '+s; l2:=l2+', 0x'+s;
  end;
  for i:=Length(stroka) to 16-1 do begin
    l:=l+', 20h'; l2:=l2+', 0x20'; end;
```

```

1[1]:=' '; 12[1]:=' ';
end;

procedure TForm1.BitBtn1Click(Sender:
TObject);
begin
//----первая строка-----
strokaconv(maskedit1.text);
memo1.clear;
memo1.lines.text:=' '+maskedit1.text;
memo1.lines.text:=memo1.lines.text+chr(1
3)+' '+maskedit2.text;
memo2.clear;
memo2.lines.text:=' //' +maskedit1.text;
memo2.lines.text:=memo2.lines.
text+chr(13)+' //' +maskedit2.text;
memo1.lines.text:=memo1.lines.
text+chr(13)+' Db ' +1;
memo2.lines.text:=memo2.lines.
text+chr(13)+12;
//----вторая строка-----
strokaconv(maskedit2.text);
memo1.lines.text:=memo1.lines.
text+chr(13)+' Db ' +1;
memo2.lines.text:=memo2.lines.
text+chr(13)+12;
end; end.

```

Алгоритм функционирования данной программы очень прост и заключается в следующем. Программа считывает первую строку сообщения и анализирует каждый символ. Если код текущего символа находится в диапазоне 0...7F, то это латинская буква или специальный символ. Для этого символа запоминается его код. Если же код оказался больше 7F, то символ набран в русской раскладке. Этому символу ставится в соответствие код в формате ЖКИ. Анализ символов реализован с помощью операторов if, аналогичные действия можно осуществить командой case. Далее программа считывает вторую строку сообщения и анализирует ее. Основными элементами данной программы являются компоненты MaskEdit1, MaskEdit2 — для ввода исходного сообщения; Memo1, Memo2 — для получения закодированного сообщения в формате языка ассемблера и C; BitBtn1 — кнопка запуска преобразования. Программу и ее исходный текст можно найти по адресу www.onyx2001.narod.ru/lcd.rar или на сайте журнала по адресу www.dian.ru/programs. На рис. 1 показан внешний вид окна программы.

Программа написана для индикаторов с форматом выводимой информации 16×2. При необходимости ее можно легко адаптировать для индикаторов других форматов и ввести дополнительные функции.

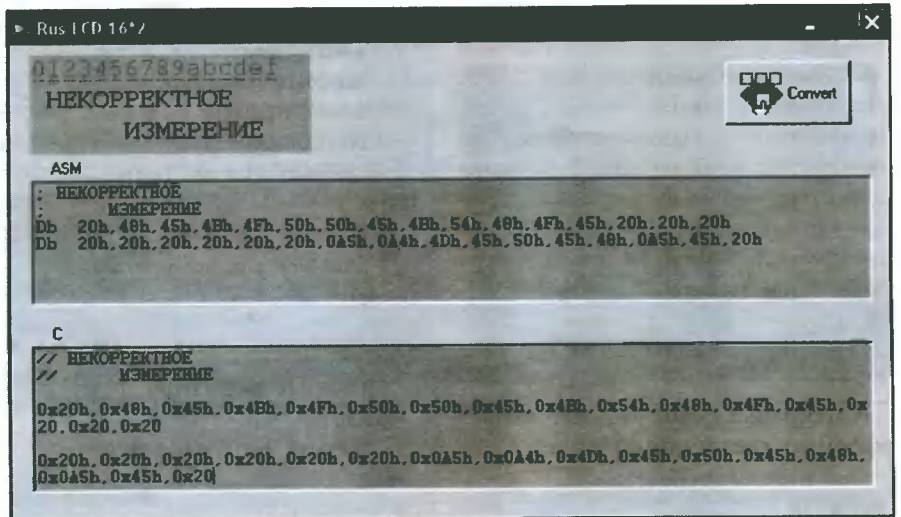


Рис. 1

При написании программного обеспечения для микроконтроллера данную программу можно интегрировать в среду разработки, что позволит запускать ее при необходимости из самой среды. Для этого в Keil 3 нужно в опции Tools/customize tools menu задать название вкладки (menu content), например, LCD converter, а в строке command определить путь к файлу exe конвертора. В результате в опции Tools появится ссылка на LCD converter, как показано на рис. 2.

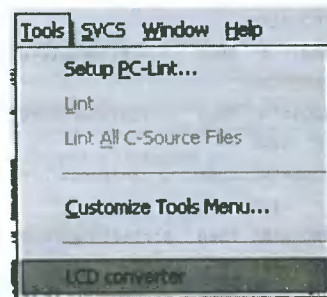


Рис. 2

Теперь для этой программы можно назначить «горячую клавишу». Для этого в опции Edit/configuration нужно выбрать вкладку shortcut key, найти там LCD converter и определить для него комбинацию клавиш.

Суть программы микроконтроллера для такого подхода заключается в последовательном считывании символов из таблицы сообщений и выводе их на ЖКИ. Ниже представлен фрагмент программы для микроконтроллера семейства x51, где осуществляется считывание данных из таблицы и их пересылка в ЖКИ.

```

MOV DPTR, #MESGREADY
CALL DISPLAY

```

```

DISPLAY:      ; подпрограмма вывода на
ЖКИ 16x2
MOV LCD_DATA, #01H ;очистить ЖКИ
CALL W_COM ;запись команды в ЖКИ
MOV CH1, #2 ; число строк
STR1: MOV LCD_DATA, #80H ;позиция
первой строки
CALL W_COM
JMP ST1
ST2: MOV LCD_DATA, #0C0H ;позиция
второй строки
CALL W_COM
ST1: MOV CH2, #16; число символов
TT: CLR A
MOVCA, @A+DPTR; читаем символ из
таблицы
MOV LCD_DATA, A
CALL W_SYM; запись символа в ЖКИ
INC DPTR ;перейти к следующему сим-
волу
DJNZ CH2, TT
DJNZ CH1, ST2
RET

MESGREADY:
; ИДЕТ
; ИЗМЕРЕНИЕ
Db 20h, 20h, 20h, 0A5h, 0E0h, 45h, 54h, 20h, 20h,
20h, 20h, 20h, 20h, 20h, 20h, 20h
Db 20h, 20h, 20h, 20h, 20h, 0A5h, 0A4h, 4Dh, 45h,
50h, 45h, 48h, 0A5h, 45h, 20h, 20h
MESGERROR:
; НЕКОРРЕКТНОЕ
; ИЗМЕРЕНИЕ
Db 20h, 48h, 45h, 48h, 4Fh, 50h, 50h, 45h, 48h,
54h, 48h, 4Fh, 45h, 20h, 20h, 20h
Db 20h, 20h, 20h, 20h, 20h, 20h, 0A5h, 0A4h,
4Dh, 45h, 50h, 45h, 48h, 0A5h, 45h

```

Каждое сообщение имеет свою метку, которую нужно указывать при вызове подпрограммы. В данном случае метка (начало таблицы) конкретного сообщения указывается в регистре DPTR.

В заключение хотелось бы остановиться на возможности программной симуляции работы микроконтроллера с ЖКИ. К пакету Keil 2/3 существует дополнение LCD simulation (HD44780) от третьих фирм, которое позволяет отлаживать программу по обслуживанию ЖКИ. Это дополнение можно найти по ссылке <http://www.c51.de/c51.de/Dateien/uVision2DLLs.php?Spr=EN#LCD>.

Для включения дополнения в среду необходимо файлы LCD_Def.cdb и lcd.dll из архива lcd.zip скопировать в папку c51/bin среды Keil. Затем для lcd.dll нужно подключить интерфейс AGSI (Advanced Generic Simulation Interface), который используется для симуляции аппаратной части устройств. Для этого необходимо в файле tools.ini прописать строку AGSI1=LCD.DLL (LCD simulation). После этого в режиме отладки появится опция Peripherals/LCD-simul. При вызове данной опции надо выбрать из библиотеки нужный индикатор, а также обозначить выводы портов, к которым он подключается к микроконтроллеру. Пример показан на рис. 3.

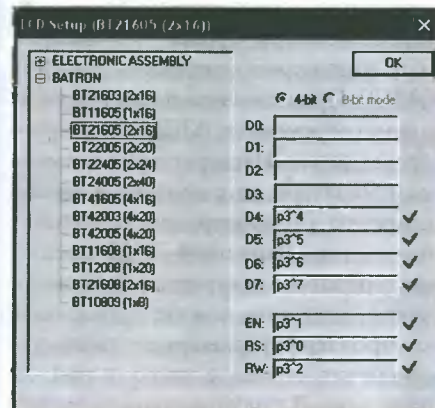


Рис. 3

К сожалению, все индикаторы, которые имеются в библиотеке, не относятся к ЖКИ с русской модификацией. После настройки индикатор доступен для работы. В режиме отладки на панели отображается информация, выводимая на ЖКИ, состояние линий портов, а также внутренние команды ЖКИ (рис. 4).

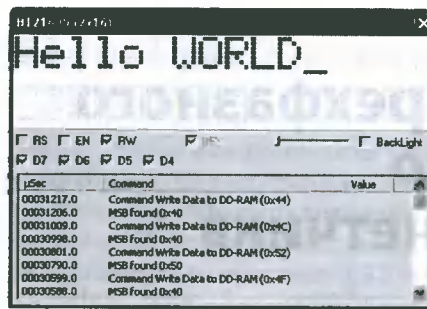


Рис. 4

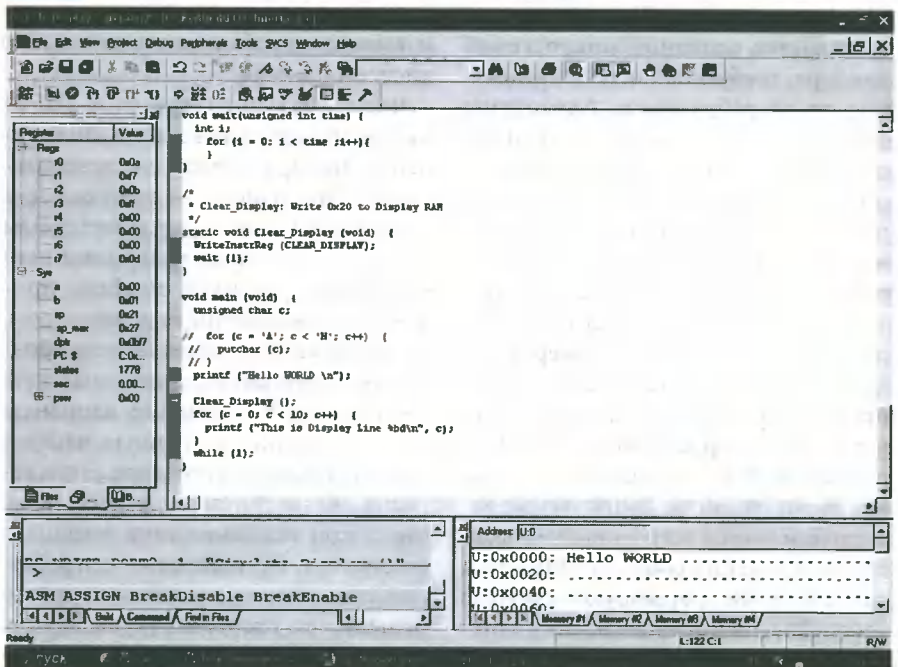


Рис. 5

При настройке ЖКИ оказывается доступным только четырехбитный режим обмена, однако, если сохранить информацию о ЖКИ на диск, а затем прописать все линии портов, к которым он подключается, то окажется доступным и восьмибитный режим.

```
[Version]
Version=0.2
[LCD_0]
Manufacturer=BATRON
OrderNr=BT21605 (2x16)
PortWidth=8
D0_NAME=p0^0
D1_NAME=p0^1
D2_NAME=p0^2
D3_NAME=p0^3
D4_NAME=p0^4
D5_NAME=p0^5
```

```
D6_NAME=p0^6
D7_NAME=p0^7
RS_NAME=p2^2
RW_NAME=p2^1
EN_NAME=p2^3
```

Существует еще один способ для программной симуляции работы с ЖКИ, который предлагают разработчики пакета Keil. Этот способ не использует внешние дополнения, а

заключается в использовании специального скрипта для отладчика (файл apnt_161.pdf на сайте www.keil.com). В этом случае информация, выводимая на ЖКИ, отображается в окне window memoгу как память, которая имеет префикс U (рис. 5).

Рассмотренные подходы могут быть полезными при проектировании различных информационно-вычислительных устройств с использованием ЖКИ-модулей.

ЛИТЕРАТУРА:

1. www.atmel.com.
2. www.keil.com.
3. www.c51.de.

Дмитрий Онышко,
г. Новочеркасск Ростовской обл.

Схемотехника трехфазного многотарифного электронного счетчика

В качестве измерителя используется высокопроизводительный процессор обработки сигналов. Измеритель часто является основной микросхемой прибора, требующей особого внимания от разработчиков. Алгоритмы обработки сигналов рассмотрены в специальной литературе. Разработка собственного измерителя позволяет реализовать практически любые пожелания заказчика по измерению параметров качества электроэнергии.

В настоящее время есть готовые современные недорогие измерители, которые выпускаются серийно. Их производят компания Analog Devices и ряд других фирм, занимающихся разработкой микросхем для счетчиков электроэнергии. Выпускаемые измерители имеют встроенный многоходовый АЦП и процессор обработки сигнала. Они формируют счетные импульсы, некоторые из них имеют встроенные источники опорного уровня, встроенный термометр, монитор питания. Применение готового измерителя существенно сокращает разработку нового прибора.

Примером готового измерителя может служить микросхема ADE7758 производства Analog Devices. Измеритель в той или иной мере содержит все основные функциональные узлы — АЦП на три канала тока и три канала напряжения, измеритель параметров, формирователь импульсных выходов, источник опорного уровня в 30 ppm/°C, встроенный монитор питания. Микросхема обеспечивает функцию калибровки параметров вычислителя, точность вычисления параметров — до 0,1 %.

Управляющий контроллер предназначен для организации внутреннего управления счетчиком, он выполняет все сервисные функции прибора. Контроллер обеспечивает передачу данных из памяти счетчика в ПК и производит обмен информацией при калибровке. Управляющий контроллер выполняет функцию накопительного механизма счетчика. Он формирует срезы мощности и загружает их во внешнюю память, принимает информацию из измерителя, преобразует

полученную цифровую информацию в единицы измерения СИ, осуществляет опрос кнопок управления и выводит необходимые данные на дисплей прибора.

Часы реального времени (ЧРВ) предназначены для ведения внутреннего календаря. Должна обеспечиваться высокая точность — уход не более 3 с в сутки. Существуют микросхемы часов с внутренним калиброванным кварцевым резонатором. Если применять часы с внешним резонатором, необходима программная калибровка, иначе нельзя получить высокую точность. Есть схемные варианты использования встроенных часов в управляющем контроллере. Этот вариант обладает тем же недостатком, что и при использовании внешнего резонатора. Но главное — потребление тока внешними часами гораздо меньше, чем у хороших управляющих контроллеров в спящем режиме. Высокая экономичность внешних часов значительно снижает требования к батарее резервного питания. Примером таких часов может служить микросхема RTC-8564JE фирмы Epson, ее использование позволит сократить время разработки прибора и упростить процесс регулировки и калибровки на этапе производства.

Батарея резервного питания обеспечивает работу часов реального времени при пропадании сетевого напряжения. Емкости батареи должно хватить не менее, чем на 6 лет хранения прибора на складе. Как правило, «экономия» на элементе питания оборачивается в дальнейшем большими проблемами. При анализе конструкций известных счетчиков видно, что разработчики обычно идут по пути применения больших батарей на напряжение 3,6 В емкостью до 1 А·ч. В счетчиках необходимо использовать литиевые элементы с малым током саморазрядки, они также имеют лучшие параметры по соотношению емкость/объем. В отличие от солевых и щелочных элементов, они довольно стабильно «держат» напряжение в течение всего срока службы. После того, как элемент разрядится, напряжение

на нем быстро падает. Литиевые элементы могут работать в достаточно широком диапазоне температур, в котором не могут функционировать солевые и щелочные приборы. Снижение емкости литиевых элементов происходит не более чем на 2...4 % в год, они могут храниться до 10 лет и работать при температуре +45 °C.

Примером могут служить литиевые элементы SL-350 фирмы Sonnenschein, ER14250 фирмы Minamoto, LS14250 фирмы SAFT.

Энергонезависимое внешнее запоминающее устройство (ВЗУ) хранит различные текущие данные. В приборах высокого класса этих данных набирается достаточно много, что требует установки нескольких ВЗУ, в них можно использовать микросхемы AT24C512-10PI-2.7 или аналогичные.

Дисплей, выполненный на основе многоместного семисегментного цифрового индикатора, предназначен для вывода различной информации (параметров электроэнергии, служебных данных). Разработаны специализированные ЖКИ, которые предназначены для установки их в электросчетчики, но в последнее время наблюдается тенденция использования более универсальных приборов — многострочных жидкокристаллических дисплеев (ЖКД). При незначительных отличиях в цене современные ЖКД имеют много преимуществ. Им характерна большая номенклатура по размерам и типам, широкое распространение, наличие подсветки. Варианты ЖКД предусматривают многострочность, большое число символов, имеют встроенный контроллер управления дисплеем, что обеспечивает малое число линий сопряжения. В качестве примера можно привести двустрочные индикаторы с 16-ю символами в строке — индикаторные панели MTC-S16204XRJXS-01 фирмы Microtips Technology или WH1602A-NGG-CP фирмы Winstar.

Основная функция дисплея — выполнять функцию индикатора суммирующего устройства, т. е. индицировать накопленное значение потребленной энергии.

Время измерения суммирующего устройства при максимальной нагрузке должно быть:

- не менее 250 ч на одну единицу старшего разряда;
- не более 15 мин на единицу младшего разряда.

Емкость учета счетного механизма должна быть не менее 1500 ч при максимальной нагрузке.

Дополнительно через систему меню на индикатор может выводиться разная служебная информация.

Кнопки управления предназначены для вывода на дисплей внутренней информации счетчика о многочисленных параметрах контролируемой сети из внутренней памяти управляющего микроконтроллера и внешней памяти. С помощью двух кнопок с использованием короткого или длительного нажатия можно реализовать достаточно большое меню контроля данных.

Интерфейс RS-485 является стандартным для использования во всех трехфазных электросчетчиках. Он обеспечивает передачу внутренней информации из него на персональный компьютер. Интерфейс позволяет обеспечить параллельное соединение информационных выходов до 125 счетчиков с помощью двух проводов и передачу информации на расстояние до 1,5 км на автоматизированные системы коммерческого учета электроэнергии (АСКУЭ). Как правило, нулевой адрес прибора используется для выполнения общей команды для группы счетчиков. Это может быть команда «зафиксировать срез мощности», «установить общее время» и многие другие. Дополнительно к стандартному интерфейсу многие счетчики высокого класса для связи с ПК имеют инфракрасный оптический порт (I-Red). Это облегчает оперативный съем информации непосредственно со счетчика без подключения к нему каких-либо проводов. Некоторые приборы могут иметь и другие распространенные интерфейсы, например CAN.

Сигнал управления блоком отключения нагрузки (БОН) используется в счетчике для управления нагрузкой в фидерной линии электропередачи. БОН позволяет автоматически отключать фидер любого энергопотребителя по каким-либо заранее заданным параметрам потребления электроэнергии. В качестве отключателей электрической энергии могут использоваться различные системы автоматики, например, УЗО (устройство защитного отключения), высоковольтные вакуумные выключатели, управляемые многофазные выключатели ПР88 и другие аналогичные устройства. Функция БОН заложена во многие типы счетчиков. Ответственность за применение этой функции лежит на пользователе прибора.

Формирователь счетных импульсов предназначен для преобразования

формы и частоты импульсов, поступающих от измерителя. Для увеличения надежности счетные импульсы должны иметь достаточно малую длительность. Для работоспособности счетчика со старыми импульсными УСПД длительность импульсов должна быть не менее 100 мс. Формирователь можно выполнить на простейшей ПЛИС, или использовать какие-либо недорогие маломощные микроконтроллеры. По внешнему сигналу «Поверка» формирователь должен перейти в режим, при котором пропорционально увеличивается число выходных импульсов. Одно из обязательных требований к формирователю импульсов — обеспечение эквидистантности (равномерности) последовательности формируемых импульсов.

К счетным импульсам в рабочем режиме предъявляются достаточно жесткие требования, заложенные в ГОСТ 30206-94. Число периодов изменения импеданса выходной цепи передающего устройства N должно быть связано с приращением измеренной энергии W , кВт·ч, зависимостью $N = R \cdot W$, где R — постоянная счетчика, $\text{имп}/(\text{кВт} \cdot \text{ч})$.

Значение постоянной счетчика R должно быть ограничено неравенством: $1111/(K \cdot P_{\text{MAX}}) < R < 12600/P_{\text{НОМ}}$, где P_{MAX} — максимальная мощность, измеряемая счетчиком.

Испытательные выходы счетчика должны обеспечивать проверку порога чувствительности за время, не превышающее 10 мин.

Из всего этого следует, что счетчик обязательно должен иметь режим поверки, который значительно увеличивает число счетных импульсов. Их минимальная длительность и частота появления практически ничем не ограничены. Повышенное число счетных импульсов значительно ускоряет этап калибровки и поверки счетчиков. В конечном итоге это заметно сокращает время и стоимость поверки государственным метрологом.

Индикатор функционирования позволяет наблюдать за функционированием счетчика. Обычно его выполняют на светодиоде. Для его работы удобно использовать импульсные сигналы счетного механизма. Если установить четыре светодиода, можно получить наглядную визуальную информацию о величине нагрузки, характере и величине реактивной составляющей, о направлении тока.

Импульсные выходы электросчетчиков являются некоторым рудимен-

том, оставшимся от индукционных приборов. В индукционных счетчиках учет электроэнергии шел с помощью подсчета импульсов, соответствующих некоторому значению накопленной энергии. Большее число импульсов за фиксированный промежуток времени означает большую скорость накопления энергии. Счетные импульсы предназначены для передачи их на устройство сбора передачи данных (УСПД). В настоящее время в связи с электронным учетом электроэнергии счетные импульсы обычно используются только на этапе производства, при калибровке и поверке счетчика метрологом.

Импеданс выходной цепи передающего устройства счетного механизма должен быть не более 200 Ом в замкнутом состоянии и не менее 50 кОм в состоянии «разомкнуто». Максимальный ток в состоянии «замкнуто» — 30 мА. Допустимое напряжение в разомкнутом состоянии — 24 В. В качестве оптронного ключа можно использовать оптроны TLP627-1 производства фирмы Toshiba или другие аналогичные.

Сигнал включения поверки переводит счетчик в этот режим, его можно включить и по компьютерному интерфейсу, но здесь может повлиять человеческий фактор. Если на производстве работник забудет перевести счетчик в рабочий режим, он может остаться в режиме поверки на линии учета, что приведет к непоправимым последствиям при учете электроэнергии.

Требования по чувствительности счетчика указаны в ГОСТ 30206-94. Так, проверку отсутствия самохода счетчика производят подачей напряжения питания при отсутствии тока в цепи. При этом испытательный выход счетчика не должен создавать более одного импульса за любое время испытаний.

Чувствительность счетчика должна быть такой, чтобы прибор регистрировал показания при токе, равном 0,001 от номинального тока при коэффициенте мощности, равном 1.

Счетчик должен измерять энергию при подаваемой на него мощности не менее рассчитываемой по формуле $P = 0,0025 \times K \times P_{\text{НОМ}}$, где K — класс точности счетчика, $P_{\text{НОМ}}$ — номинальное значение мощности, рассчитанное по номинальным значениям тока и напряжения, кВт.

Рассмотренная схема показывает общую структуру трехфазного элек-

тронного счетчика. В связи с быстрыми темпами развития электроники часть узлов может быть совмещена в едином конструктивном элементе. На первом этапе разработки бывает удобно немного усложнить аппаратную часть прибора для того, чтобы упростить разработку встроенного программного обеспечения. Это значительно сократит время на разработку нового прибора, ускорит выход его на рынок. Все входные и выходные сигналы управления должны иметь оптронную развязку, выдерживающую испытательное напряжение до 4 кВ. В интерфейсе RS-485 необходимо использовать достаточно быстродействующие оптроны для обеспечения возможности работы при больших скоростях обмена с ПК, например 4N33, 4N35 и другие аналогичные. В других цепях, где быстродействие не критично, можно установить более дешевые медленные оптроны.

КОНСТРУКЦИЯ КОРПУСА

Электронные счетчики, как правило, имеют пластмассовый корпус светлого цвета. Корпус состоит из цоколя, кожуха и крышки зажимов. По определению ГОСТ 30206-94, цоколь — это задняя часть счетчика, служащая для его крепления и крепления измерительного элемента, зажимов и кожуха. Кожух — это крышка с передней стороны счетчика, герметизированная по периметру. Крышка зажимов — крышка, закрывающая зажимы счетчика и концы внешних проводов или кабелей, присоединенных к зажимам. Измерительный элемент — это печатная плата, на которой установлены основные функциональные элементы счетчика — дисплей, кнопки управления, индикатор функционирования и др. Цоколь счетчика должен иметь три точки для крепления. Места крепления должны находиться вне полезной площади корпуса счетчика. Нижние крепежные кронштейны обычно изготавливаются вместе с корпусом, с выступом вниз в углах цоколя. Кронштейны должны закрываться крышкой зажимов. Крепление кожуха к цоколю обычно осуществляется сверху на защелках, снизу — двумя винтами с возможностью их пломбирования. Кожух может изготавливаться либо целиком из прозрачного материала, либо из непрозрачного материала с окном, через которое можно наблюдать за индикатором функционирования и считывать показания дисплея. Как правило, на кожухе должна быть пре-

дусмотрена возможность размещения оптоприемника по МЭК 1107-96. В некоторых счетчиках в кожухе делают изолированный отсек с подпружиненными контактами для батареи резервного питания. Типоразмер батареи 1/2 АА. Отсек батареи должен иметь пломбируемую крышку, которая не имеет выступов и не портит внешний вид счетчика. На кожухе наносят фирменные знаки предприятия-изготовителя.

Зажимы представляют собой коммутационные устройства, предназначенные для подключения к счетчику измерительных цепей тока и напряжения, а также вспомогательных слаботоковых цепей. Они должны обеспечивать однозначное раздельное включение цепей тока и напряжения. Зажимы должны обеспечивать подключение до двух медных или алюминиевых проводов с суммарным сечением до 5 мм². Соединения зажимов цепей тока и цепей напряжения должны быть разъемными и размещены в зажимной коробке. Отверстия для зажима проводов цепи тока должны быть диаметром не менее 4,2 мм. Все зажимы цепи напряжения должны быть раздельными и иметь отверстия диаметром не менее 4,2 мм. Диаметр отверстий для зажима проводов цепей тока должен быть больше диаметра отверстий для зажима цепей напряжения.

Число слаботоковых клемм может быть 8—16. Чем больше вспомогательных клемм, тем проще обеспечить большую функциональность счетчика. Способ крепления проводов к зажимам должен обеспечивать надежный и долговечный контакт, исключаящий опасность ослабления соединения или перегрева. Конструкция зажимов должна минимизировать вероятность возникновения коррозии.

Крышка зажимов должна закрывать все зажимы, винты крепления, внешние провода длиной не менее 55 мм. На лицевой стороне крышки зажимов должна быть предусмотрена возможность установки шильдика с данными трансформаторов тока и напряжения, к которым подключен счетчик. С внутренней стороны крышки зажимов должна быть нанесена схема подключения счетчика к электросети.

Корпус счетчика должен предусматривать возможность пломбирования. Все пломбы должны быть доступны для визуального контроля при полнотой собранном и установленном приборе.

Наряду с надежностью и функциональностью счетчика, дизайн корпуса сильно влияет на популярность прибора. Дизайн должен соответствовать лучшим моделям зарубежного производства. К сожалению, до последнего времени в нашей стране уделяется мало внимания к внешнему виду разрабатываемой РЭА.

СТЕНДОВОЕ ОБОРУДОВАНИЕ

При разработке, регулировке и на производстве электронных счетчиков необходимо иметь соответствующее компактное стендовое оборудование. Стенды обеспечивают оперативное формирование тестовых сигналов и проведение контрольных замеров. Стенд для регулировки должен содержать блок напряжений и блок тока. Блок напряжения позволяет изменять напряжение питания счетчика для оперативной проверки работоспособности прибора. Блок тока должен выдавать токи частотой 50 Гц различного номинала во всем рабочем диапазоне счетчика от минимально фиксируемого до максимального рабочего тока. Рабочее место регулировщика и разработчика дополнительно должно быть оборудовано визуальным индикатором наличия импульсных сигналов с выхода счетчика.

На производстве необходимо иметь метрологическую установку с образцовым счетчиком для калибровки собранных приборов и их метрологической поверки. Образцы установок можно посмотреть на сайте концерна «Энергомера».

Для обеспечения надежности изготовленных приборов необходимо иметь стенд прогона с встроенным стендом помех. На стенде прогона изготовленный прибор получает начальную приработку. Стенд помех формирует импульсные провалы напряжения, имитирующие реальные помехи, встречающиеся в электрических сетях. Форма сигнала помех определена в приложении «В» ГОСТ 30206-94. Использование стенда помех позволит обеспечить надежность и помехоустойчивость работы каждого изготовленного прибора.

Электросчетчики работают в достаточно тяжелых климатических условиях, поэтому с ними дополнительно проводят климатические испытания в термокамере, камере холода, камере влаги.

Александр Маврычев,
г. Балахна, Нижегородская обл.

Макетная плата для микропроцессоров PIC16F62х и PIC16F84

Макетная плата предназначена для отладки программ, написанных для микропроцессоров фирмы Microchip PIC16F627, PIC16F628 и PIC16F84 в корпусах DIP. Статья будет интересна не только тем, кто делает первые шаги в увлекательный мир программируемой логики, но и радиолюбителям со стажем. Плата проста в повторении и потребует минимальных денежных вложений при изготовлении.

Когда автор только начинал заниматься разработкой устройств на микроконтроллерах фирмы Microchip, часто приходилось собирать узел из стандартных внешних элементов — цепи сброса MCLR, цепи питания, внешнего кварцевого резонатора или RC-цепочки. Иногда нужно было связать микроконтроллер с компьютером, для чего требовался преобразователь уровней RS-232—TTL. И все это нужно было собирать при разработке каждого нового устройства заново. Перед тем, как начать писать основную программу, приходится писать тестовую, предназначенную для проверки работоспособности микроконтроллера на собранной макетной плате, чтобы быть уверенным в том, что микроконтроллер исправен и правильно работает.

Для отладки микроконтроллеров наиболее часто используются следующие типы макетных плат.

- Универсальная макетная плата, изготовленная из фольгированого стеклотекстолита. Предназначена для отладки любых радиоэлектронных устройств. Она обладает рядом существенных недостатков — самая дешевая стоит не меньше 100 руб. Очень часто при длительном использовании отслаиваются проводники из-за перегрева паяльником, разводка проводников не всегда оптимальна и часто вносит дополнительные помехи в работу устройства, для соединения элементов используется пайка. Автор считает, что оптимальное использование универсальной макетной платы — сборка отлаженных устройств, которые настроены, но нет времени или желания разрабатывать печатную плату.
- Наиболее распространены макетные платы фирмы WISHER. Они позволяют производить быст-

рый монтаж довольно сложных устройств, используя проволочные перемычки (джамперы). Комплект поставки включает в себя клеммные соединители для подачи питания на плату и проволочные перемычки. Довольно хорошее решение, но и ему присущи недостатки. Самая простая макетная плата WB-102+J стоит 190 руб., более удобная для макетирования несложных устройств WB-104-3+J — 380 руб. по прайсу фирмы «Платан». Низкая надежность контактов, особенно это проявляется после того, как пытаешься получить контакт в гнезде, в котором до этого побывал относительно толстый проводник, например, от резистора МЛТ-2. Значительна паразитная емкость между параллельными рядами контактов.

- Разработка радиолюбителей, например [1] — интересная конструкция для PIC16F877, но она предназначена скорее для изучения этого микроконтроллера, чем для макетирования своих устройств, т. к. выводы микроконтроллера жестко распределены для выполнения определенных функций — индикация, ввод-вывод данных, звуковая индикация и т. п. Другая конструкция [2] предназначена для изучения PIC16F84, обладает тем же недостатком, что и первая, но для первоначального освоения этих микроконтроллеров они очень хорошо подходят.
- Интересные макетные платы представлены на сайте «Мега-Электроника» [3] — PIC-IO, PIC-MT, PIC-PG4D-628. Дополнительной информации по их устройству автору не удалось получить. Могут только сказать, что цена их высока — от 702 руб. за PIC-PG4D-628 и до 1091 руб. за PIC-MT.

- Простые макетные платы представлены на сайте [4] для микроконтроллеров PIC16F84, PIC16F873, PIC16F874. Цена \$16...20. Содержат стабилизатор на 5 В, кварцевый резонатор, разъемы для подключения к портам, цепи сброса. Основной минус, а может быть и плюс — это их простота.

- На сайте «Железный Феликс» [5], представлена интересная система SimmStick — макетная плата, состоящая из кроссплаты от старых разъемов для установки модулей памяти SIMM. На них подано питание, сигналы от периферийных разъемов. Макет собирается на отдельной плате в виде модуля и вставляется в разъем SIMM. Основные недостатки — мала плата для макетирования, необходимо применение паяльника для сборки.

После анализа элементов, необходимых для работы микропроцессора, сделан вывод, что макетная плата должна обладать следующими элементами:

- узел формирования напряжения питания + 5 В;
- цепь сброса MCLR и обязательна кнопка сброса;
- цепь подключения резонатора;
- узел связи с компьютером при помощи RS-232;
- цепи подтягивания портов к выводу +5 В или общему проводу;
- внутреннее программирование на плате (т. е. без извлечения микропроцессора из платы, что значительно повышает ресурс панельки для установки микросхемы);
- наличие разъемов и одиночных разъемных соединителей для подключения к портам;
- простота реализации и настройки.

Проанализируем расположение выводов распространенных микропроцессоров серии PIC16F62х и PIC16F84, как наиболее часто применяемых в радиолюбительской практике, и представим их в виде табл. 1.

Раскладка выводов у этих двух типов микропроцессоров одинакова, отличие заключается только в том, что PIC16F62х более насыщен функциональными возможностями — наличие USART, несколько типов тактовых генераторов, модуль аналоговых компараторов, три таймера, модуль захвата и сравнения ШИМ. Таким образом, при разработке макетной платы нужно ориентироваться на микропроцессоры семейства PIC16F62х.

Таблица 1

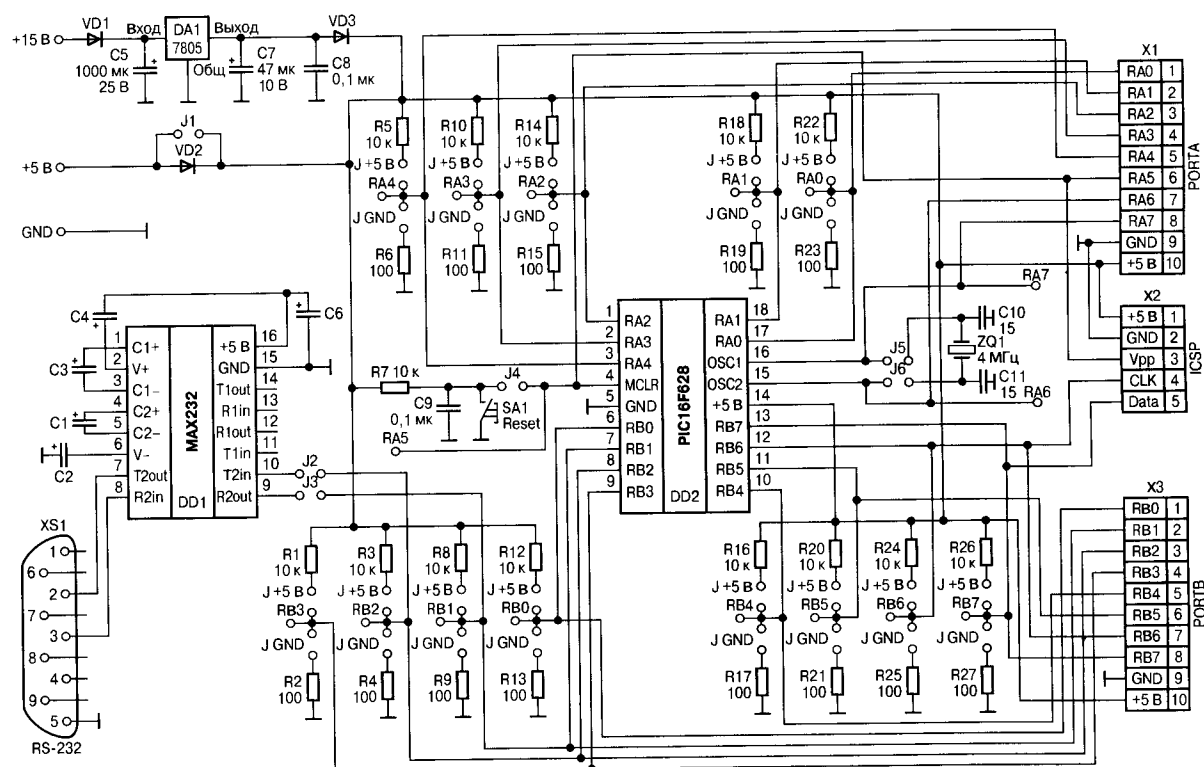
Вывод	PIC16F62х	PIC16F84	Описание вывода
1	RA2/AN2/Vref1	RA2	Двунаправленная линия ввода/вывода (в PIC16F62х аналоговый вход компаратора, выход источника опорного напряжения)
2	RA3/AN3/CPM1	RA3	Двунаправленная линия ввода/вывода (в PIC16F62х аналоговый вход компаратора, выход компаратора)
3	RA4/TOCKI/CPM2	RA4/RTCC	Двунаправленная линия ввода/вывода (в PIC16F62х может использоваться как TOCKI, выход компаратора)
4	RA5/-MCLR/THV	-MCLR	Сигнал сброса микроконтроллера (в PIC16F62х вход напряжения программирования, вход цифрового сигнала)
5	GND	GND	Общий провод
6	RB0/INT	RB0	Двунаправленная линия порта ввода/вывода, вход внешнего прерывания
7	RB1/RX/DT	RB1	Двунаправленная линия порта ввода/вывода, (в PIC16F62х вход приемника USART, линия данных в синхронном режиме)
8	RB2/TX/CK	RB2	Двунаправленная линия порта ввода/вывода, (в PIC16F62х выход передатчика USART, линия тактового сигнала в синхронном режиме)
9	RB3/CCP1	RB3	Двунаправленная линия порта ввода/вывода, (в PIC16F62х вывод модуля CCP)
10	RB4/PGM	RB4	Двунаправленная линия порта ввода/вывода, (в PIC16F62х вход для низковольтного программирования)
11	RB5	RB5	Двунаправленная линия порта ввода/вывода
12	RB6/T1OSO/T1CKI	RB6	Двунаправленная линия порта ввода/вывода (в PIC16F62х выход генератора таймера 1, вход синхронизации при программировании)
13	RB7/T1OSI	RB7	Двунаправленная линия порта ввода/вывода (в PIC16F62х вход генератора таймера 1, вход/выход данных при программировании)
14	VDD	VDD	Питание микроконтроллера
15	RA6/OSC2/CLKOUT	OSC2/CLKOUT	Выход генератора для подключения кварцевого резонатора (в PIC16F62х двунаправленная линия ввода/вывода)
16	RA7/OSC1/CLKIN	OSC1/CLKIN	Вход генератора для подключения кварцевого резонатора (в PIC16F62х двунаправленная линия ввода/вывода)
17	RA0/AN0	RA0	Двунаправленная линия порта ввода/вывода (в PIC16F62х аналоговый вход компаратора)
18	RA1/AN1	RA1	Двунаправленная линия порта ввода/вывода (в PIC16F62х аналоговый вход компаратора)

Определимся с имеющимися портами:

- RA0—RA4 — сделаем возможность подключения вывода порта через резистор к источнику +5 В или к общему проводу;
- RA5—RA7 — просто отводы, они будут использоваться при специальном конфигурировании микропроцессора, у PIC16F84 эти выводы используются по прямому назначению, т. е. они не могут быть портами ввода и вывода;
- RB0—RB7 — сделаем возможность подключения вывода порта к источнику +5 В или к общему проводу, кроме этого выводы RB1 и RB2 через джампер подключим к преобразователю уровней RS-232—TTL.

Принципиальная схема показана на рис. 1.

Питание на макетную плату можно подавать двумя способами. Если имеется стабилизированный источник +5 В, питание подается на вторую сверху клемму (джампер J1 не должен быть подключен, это исключает возможность повреждения от неправильной полярности питания), в этом случае не используется стабилизатор +5 В. Второй способ — если имеется нестабилизированный источник постоянного тока 10...20 В, его положительный вывод соединяется с верхней клеммой, далее напряжение стабилизируется микросхемой DA1



7805 или аналогичной, если есть необходимость подавать +5 В на другие узлы устройства, необходимо установить джампер J1.

Внешние элементы микроконтроллера DD2 стандартны, выбраны согласно документации фирмы Microchip DS40300b, сопротивление резистора R7 должно быть менее 40 кОм, емкость конденсатора C9 — 0,1 мкФ. Джампер J4 включает узел сброса, когда вывод 4 MCLR сконфигурирован как цепь сброса, если джампер не установлен, этот вывод можно использовать как цифровой порт ввода/вывода. При внутрисхемном программировании (ICSP) этот джампер не должен быть установлен, чтобы не влиять на работу программатора. Кнопка SB1 «Reset» перезапускает микроконтроллер.

Для подключения резонатора используются джамперы J5, J6, ZQ1 выбирается с параллельным резонансом, т. к. при его использовании с последовательным резонансом можно получить частоту, не соответствующую той, что указано на нем. Частота и тип кварцевого резонатора выбирается в зависимости от создаваемого устройства, при этом режимы работы генератора микроконтроллера подразделяются на варианты:

- LP — низкочастотный резонатор,
- XT — обычный резонатор,
- HS — высокочастотный резонатор.

От типа резонатора зависит емкость конденсаторов C10 и C11, она определяется согласно табл. 2.

Если будет использоваться встроенный генератор микроконтроллера, можно использовать выводы RA6, RA7 как цифровые для ввода/вывода, при этом, кроме указания необходимой конфигурации в микроконтроллере, необходимо также убрать джамперы J5, J6 для отключения резонатора (при использовании микроконтроллеров серии PIC16F62x).

Подключение любого вывода порта к источнику +5 В осуществляется при помощи джампера J +5 В, к общему проводу — J GND.

Подключиться к выводу любого порта можно либо при помощи специального разъема X1 PORTA или X3 PORTB, либо к специальному лепестку от разъема типа PIMM1 — 66ГЗ-В.

Для внутрисхемного программирования используется разъем X2 ICSP. Выводы 1 и 2 подают питание от программатора, вывод 3 — напряжение 12,5...14 В для перевода микроконтроллера в режим программирования, вывод 4 — тактовые импульсы, вывод 5 — данные.

Для связи с компьютером используется разъем XS1 RS-232, он подключается к микросхеме преобразователя уровней RS-232—TTL DD1 типа MAX232 или аналогичной. Емкость конденсаторов C1—C4, C6 определяется по документации, для MAX232 составляет 1 мкФ. Для наиболее распространенных микросхем преобразователей уровня RS-232—TTL емкость конденсаторов приведена в табл. 3.

Это далеко не полный список возможных замен, при необходимости можно использовать функциональный аналог MAX232 других фирм, например Analog Devices.

Для использования RS-232 нужно замкнуть джамперы J2 и J3, тем самым микросхема DD1 подключается к USART микроконтроллера DD2. Выводы RB1 и RB2 будут использоваться как порты универсального синхронно-асинхронного приемопередатчика. Напоминаю, что USART есть только у микроконтроллеров семейства PIC16F62x, у PIC16F84 его нет, поэтому если необходима связь по RS-232, придется реализовать USART программно.

Микропроцессор DD2 устанавливается в панельку. В качестве DD2

Таблица 3

Микросхема	C1—C4, C6, мкФ
MAX232	1,0
MAX232A	0,1
MAX220	0,1
MAX243	0,1

можно использовать микросхемы PIC16F84, PIC16F627, PIC16F628 и другие, имеющих 18 выводов и такую же раскладку по питанию и портам ввода/вывода.

Настройку устройства необходимо начать с подачи постоянного напряжения питания 10...25 В. При исправных элементах на выходе микросхемы DA1 будет напряжение +5 В, это напряжение проверяется на выводе 14 микроконтроллера DD2. При включенном джампере J4 на выводе 4 будет около 5 В, при нажатии кнопки SA1 — нуль. Далее поочередно подключаем джамперы J +5 В к каждому выводу PORTA и PORTB и проверяем наличие +5 В на специальном лепестке, разъемах X1—X3 и соответствующих выводах микроконтроллера. То же самое делаем и с джампером J GND.

При первом включении макетной платы с загруженной программой в микроконтроллер на выходе RB0 будут прямоугольные импульсы, сигнализирующие о том, что микроконтроллер исправен и работает.

Проверку работы преобразователя можно провести, подключив макетную плату к COM-порту компьютера при помощи простого кабеля, у которого одноименные выводы соединены между собой.

Проверка осуществляется при помощи специальной тестовой программы Test, написанной на языке Delphi. Для начала настраиваем нужный COM-порт компьютера, нажав на кнопку «Настройка COM порта». Выбираем порт, скорость Baud rate, Data bits и Stop bits оставляем по умолчанию, т. к. они связаны со скоростью работы микроконтроллера и частотой его резонатора (в данном случае — 4 МГц). Далее нажимаем на кнопку «Открыть порт» — тем самым программа открывает указанный в программе COM-порт. Хочу заметить, что работать с этим портом программа будет единолично, т. е. другая программа не получит доступ к нему до тех пор, пока эта программа его не освободит. Кстати, все программы, использующие внешние COM-порты, захватывают их в единоличное использование, поэтому,

Таблица 2

Режим	Частота	C10, C11, пФ	Примечание
Керамический резонатор ZQ1			
XT	455 кГц	22...100	Большая емкость увеличивает стабильность генератора, но увеличивается и время запуска. Значения емкости являются оценочными и подбираются опытным путем.
	2 МГц	15...68	
	4 МГц	15...68	
HS	8 МГц	10...68	
	16 МГц	10...22	
Кварцевый резонатор ZQ1			
LP	32 кГц	68...150	Большая емкость увеличивает стабильность генератора, но увеличивается и время запуска. Значения емкости являются оценочными и подбираются опытным путем.
	200 кГц	15...30	
XT	100 кГц	68...150	
	2 МГц	15...30	
	4 МГц	15...30	
HS	8 МГц	150	
	10 МГц	15...30	
	20 МГц	15...30	

если при открытии порта возникла ошибка, прежде всего сделайте так, чтобы другие программы его не использовали.

Далее настраиваем нужный порт либо на прием, либо на передачу, выбрав в пункте «Настройка порта» вариант «Передача данных» для настройки соответствующего порта на выход. Стоит сказать, что на RB1, RB2 установка и прием данных невозможны — они используются для связи с компьютером. Для приема данных из порта надо выбрать пункт «Прием данных». Если выводы порта не будут подключены к источнику +5 В или общему проводу, на выходе результат будет неизвестен.

Для записи данных надо указать число и нажать на кнопку «Записать

в порт А» или «Записать в порт В». Для того, чтобы прочитать информацию из порта, надо выбрать «Прием данных» и нажать кнопку «Принять данные из порта А» или «Принять данные из порта В».

Индикатор Tx показывает, что сейчас идет передача данных в микроконтроллер, а индикатор Rx — о прием.

На сайте журнала www.dian.ru/programs находится архив, в нем — файл для загрузки в микроконтроллер и исходные файлы для MPLAB IDE, исходный текст программы проверки монтажной платы на Delphi. Разводка печатной платы сделана при помощи программы SprintLayout.

ЛИТЕРАТУРА:

1. А. Киселев, Ю. Мартышевский. *Макетная плата для освоения PIC микроконтроллеров фирмы Microchip*. — *Схемотехника*, 2003, № 12 с. 35.
2. В. Федоров. *Стенд для изучения основ работы с микроконтроллерами Microchip*. — *Схемотехника*, 2004, № 9, с. 37.
3. Сайт фирмы «Мега-Электроника». www.megachip.ru.
4. Сайт Brian's PIC resource page. <http://www3.mistral.co.uk/brian.clewer/index.htm>.
5. Робототехнический сайт «Железный Феликс». <http://www.ironfelix.ru/>.

Алексей Николаев,
г. Боготол, Красноярский край

Кабельный пробник со звуковой сигнализацией

В статье рассмотрен усовершенствованный кабельный пробник на основе микроконтроллера AT89C2051.

В настоящее время существует множество различных пробников для прозвонки кабеля на витой паре, который используется при монтаже локальных вычислительных сетей (ЛВС). Как правило, такой кабель содержит четыре пары проводов, свитых с определенным шагом. Дорогие модели приборов позволяют не только выявлять ошибки в разделке жил кабеля, но и измерять его электрические параметры. Однако использовать такие дорогие приборы, видимо, имеет смысл только при сертификации кабельной системы. Для контроля правильности заделки жил кабеля часто используют простые пробники, большинство из которых используют принцип «бегущих огней». Такой пробник состоит из активной части, которая включает в себя генератор «бегущая единица» с источником питания и пассивной части, которая состоит из линейки светодиодов, расположенных в один ряд. Обе части содержат стандартные гнезда RJ-45 и подключаются к противоположным концам тестируемой кабельной линии. По очередности и количеству зажигающихся светодиодов можно судить о правильности заделки концов кабеля, об обрывах и коротких замыканиях в проводах.

Генератор «бегущая единица» может быть реализован множеством способов, в том числе и на микросхемах средней степени интеграции (счетчиках, дешифраторах и т. п.). Однако при таком подходе трудно реализовать режим экономии энергии источника питания (или потребует существенное усложнение прибора). На практике могут быть случаи, когда пробник после окончания тестирования просто забывают выключить. Активная часть может потреблять ток порядка 10 мА и более, т. е. энергия источника питания (гальванических элементов или аккумуляторов) может быть полностью исчерпана в течение нескольких дней. К следующему тестированию может оказаться, что элементы питания не только разряжены, но и потекли. Поэтому пробник должен обладать некоторым «интеллектом», позволяющим сберегать энергию источника питания. Такая энергосберегающая функция может быть легко реализована с помощью микроконтроллера. Здесь возможны два пути. Первый — использовать один из выводов микроконтроллера для управления внешними цепями, через которые подается питание на активную часть пробни-

ка. Второй — использовать энергосберегающие режимы, заложенные в самом микроконтроллере. Один из них (Power Down) отличается тем, что тактовый генератор микроконтроллера останавливается и он переходит в статический режим. В этом случае ток потребления составляет единицы микроампер и, если погашены светодиоды пассивной части либо она отсоединена, пробник может находиться во включенном состоянии довольно длительное время. Именно такой подход и реализован в описываемой ниже конструкции. Пробник автоматически переходит в режим Power Down через 20 мин после отсоединения пассивной части.

Для удобства работы пробник снабжен звуковым излучателем, который сигнализирует о нормальной работе и о переходе в режим Power Down. Основа активной части — микроконтроллер AT89C2051.

Схема пробника изображена на рис. 1.

При замыкании выключателя SA1 на активную часть пробника подается напряжение питания. При этом на вывод 1 микроконтроллера поступает лог. 1 на время, определяемое постоянной времени цепочки $R_{вн}C1$ ($R_{вн}$ — сопротивление внутреннего резистора микроконтроллера между выводом 1 и общим проводом). Лог. 1 на выводе RST микроконтроллера устанавливает в начальное состояние все его регистры и начинается выполнение программы с нулевого адреса. Упрощенная блок-схема программы изображена на рис. 2.

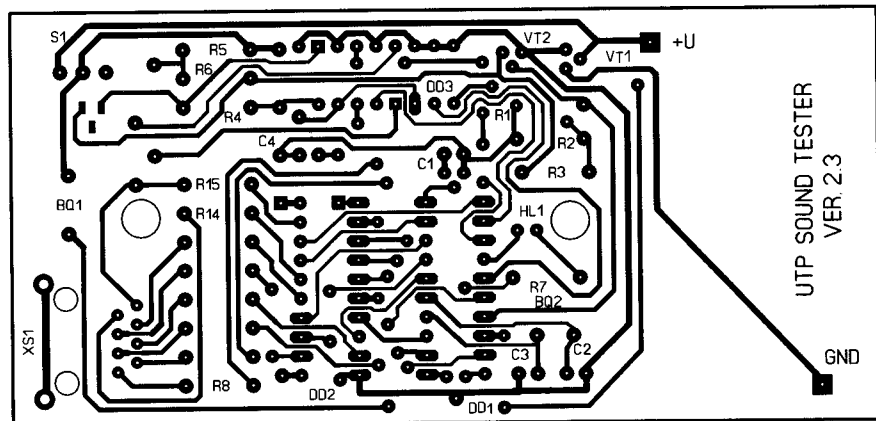


Рис. 3

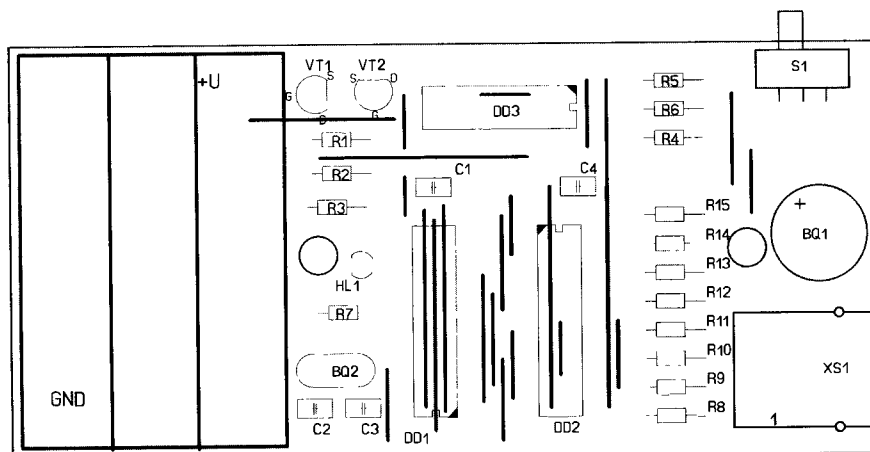


Рис. 4

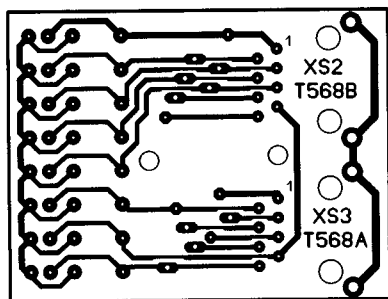


Рис. 5

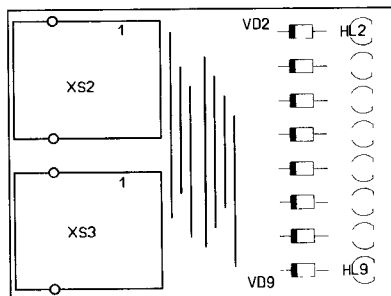


Рис. 6

разряд элемента ниже 1,05 В. Если питающее напряжение в норме, пробник издает короткий звуковой сигнал длительностью 100 мс, если оно ниже нормы, сигнал становится прерывистым (5 раз по 100 мс) в каждом рабочем цикле. В качестве источника сигнала используется пьезокерамический излучатель со встроенным генератором ВА1. Управление излучателем осуществляется через вывод 16 порта P1 микроконтроллера, активный уровень — низкий. Затем осуществляется проверка присоединения пассивной части со светодиодами. Для этого отслеживается напряжение на правом (по схеме) выводе резистора R15, при этом также используется встроенный аналоговый компара-

тор микроконтроллера. Напряжение с этого вывода через канал Y2—Y мультимплексора поступает на инвертирующий вход P1.1 компаратора. На его неинвертирующий вход P1.0 через канал X2—X поступает сигнал с делителя напряжения R2R3. Если напряжение на выходе делителя R2R3 больше, чем на правом выводе резистора R15, выход компаратора устанавливается в лог. 1. Когда пассивная часть присоединена, напряжение на правом выводе резистора R15 меньше, чем сигнал с выхода делителя R2R3. Если пассивная часть отсоединена, напряжение на правом выводе резистора R15 будет близко к напряжению питания и превысит уровень с выхода делителя R2R3, при этом выход компаратора установит-

ся в лог. 0. Управление адресными входами мультимплексора DD3 осуществляется с выходов P1.3 и P1.6 порта P1 микроконтроллера.

Если пассивная часть отсоединена, в регистр R5 микроконтроллера заносится некоторое число, определяющее длительность работы до отключения пробника. В каждом рабочем цикле происходит декремент регистра R5, и когда его содержание станет равным нулю (примерно через 20 мин после отсоединения пассивной части), пробник издает длинный (около 4 с) звуковой сигнал и переходит в режим Power Down. Происходит запись определенного числа в регистр управления мощностью PCON микроконтроллера. Перед записью в регистр PCON происходит запись лог. 0 в разряд P1.5 порта P1 микроконтроллера, а также запись неактивных уровней в остальные разряды портов-выходов микроконтроллера. Ключ на VT2 закрывается, отключая от общего провода нижние плечи резистивных делителей R2R3 и R5R6 и цепь источника опорного напряжения R4VD1. Таким образом, в режиме Power Down ток потребляют только микроконтроллер, буферный регистр DD2 и мультимплексор DD3. Выход из этого режима осуществляется отключением питания выключателем SA1.

В каждом рабочем цикле, как в нормальном режиме, так и в режиме отсчета времени перед переходом в режим Power Down, производится запись в разряды P1.7 (младший разряд), P3.0—P3.5 и P3.7 (старший разряд) портов P1 и P3 таким образом, что получается «бегущая единица» от младшего разряда к старшему. С целью экономии энергии источника питания лог. 1 устанавливается на очередном выводе на 150 мс, после чего на этом же выводе на 250 мс формируется лог. 0. Затем лог. 1 устанавливается на 150 мс на следующем выводе и снова туда же выводится на 250 мс лог. 0. Так происходит со всеми выводами поочередно. Выходы микроконтроллера соединены с входами восьмиразрядного буферного регистра DD2 74HC373. Через токоограничивающие резисторы R8—R15 (и через тестируемый кабель) он управляет светодиодами HL2—HL9. Лог. 1 на выводе DD2 зажигает соответствующий светодиод. Таким образом, длительность одного цикла составляет примерно $(150+250) \times 8 = 3200$ мс, причем

светодиоды светятся меньше, чем половину цикла. Наличие или отсутствие присоединения пассивной части проверяется в каждом рабочем цикле как в нормальном режиме, так и в режиме отсчета времени перед переходом в режим Power Down. В режиме отсчета времени пробник издает прерывистый звуковой сигнал (три раза по 100 мс) в каждом рабочем цикле. После присоединения пассивной части пробник переходит в обычный рабочий режим.

Пассивная часть снабжена двумя разъемами RJ-45 для двух вариантов заделки — 568B (компьютер-концентратор) и 568A (компьютер-компьютер). Это удобнее, чем иметь два разъема в активной части — могут быть ситуации, когда заранее неизвестно, по какому варианту разделан кабель, нужно будет идти к противоположному концу кабельной линии (длина которой может достигать 100 м), чтобы подключить его к другому разъему активной части.

Индикация работы активной части осуществляется с помощью светодиода HL1, управление которым производится с вывода 14 микроконтроллера. Причем также с целью экономии

энергии светодиод включается на 200 мс и на 2000 мс гаснет. Управление разрядом P1.2 производится с помощью внутреннего прерывания от встроенного таймера T0 микроконтроллера.

Перед началом работы следует убедиться в работоспособности пробника, соединив активную и пассивную части гибким и мягким отрезком ка-

беля с запрессованными на его концах вилками RJ-45.

Активная и пассивная части собраны на печатных платах из односторонне фольгированного стеклотекстолита толщиной 1 мм (рис. 3—6) размерами соответственно 114×54 и 50×38 мм.

В активной и пассивной частях применены высокоэффективные

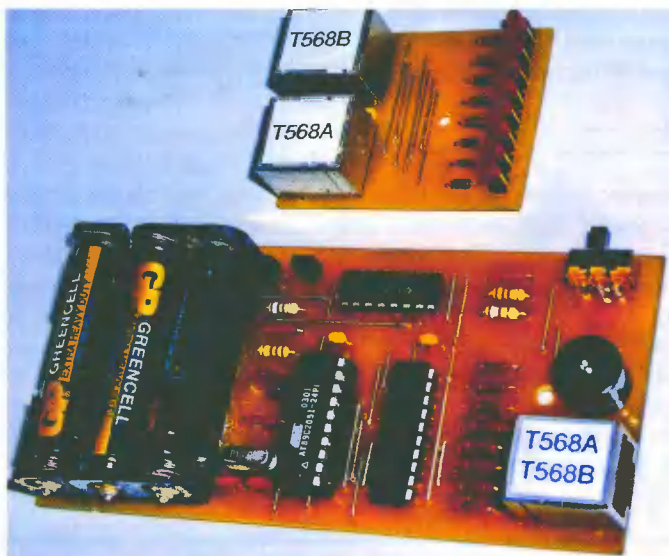


Рис. 7

IOR

Honeywell

MAXIM

DALLAS

EPCOS

MITSUBISHI

ELECTRIC

PHILIPS

Infineon

ROHM

BOURNS

ANALOG

DEVICES

VISHAY

AMP

SICK

DAEA

VISION

CRYDOM

Kingbright

ИНТЕГРАЛЬНЫЕ КЛЮЧИ

Двухрежимные контроллеры тока и напряжения + МОП-транзистор в одном корпусе.

- Токковый режим управления и дополнительный режим управления напряжением
- Защита от перенапряжения, перегрузки по току и температуре
- Работают в квазирезонансном режиме и с переменной частотой (управление скважностью)
- Для импульсных источников питания
- 5-выводные корпуса TO-220/TO-262



Наименование	Напряжение пробоя сток-исток, В	Сопротивление в откр.сост., Ом	Ток	Выходная мощность, Вт
IRIS4007(K)	200	0,4	4	30
IRIS4009(K)	650	8	1,5	30
IRIS4011(K)	650	3,95	2,5	60
IRIS4013(K)	650	1,95	5,1	120
IRIS4015(K)	650	0,9	8	180



светодиоды красного цвета диаметром 3 мм с прямым напряжением 1,8...1,9 В и хорошей яркостью при токе 2...3 мА. Средний потребляемый ток пробника в активном режиме — не более 10 мА, в режиме Power Down — не более 1 мА. Пробник не боится замыканий жил в кабеле. Внешний вид его смонтированных плат показан на рис. 7.

Для питания пробника используются три гальванических элемента ААА. Нех-файл прошивки микроконтроллера приведен в Листинге 1.

ЛИТЕРАТУРА:

1. В. Василенко. Кабельный тестер. — Радиохобби, 2005, № 2.
2. А. В. Фрунзе. Микроконтроллеры? Это же просто! Т 1 и 2. — М.: ИД Скимен, 2002.
3. AT89C2051 8-bit Microcontroller with 2K Bytes Flash. Atmel data sheets.

Виктор Василенко,
Украина, г. Свердловск
Луганской обл.

ЛИСТИНГ 1

```
:03000000020100FA
:0300080002027B73
:18010000758901758ACA758CF5D2A9D2AFD28C75903775B00012024B0F
:18011800A2B640147F06DF0302014FC294120259D29412025902011E83
:18013000C294120259D294120259D293A2B640030201B2C29312024B8B
:18014800A2B6400302011CC29775B000120267D297120260C2971202A2
:180160006775B00112026075B00012026775B00212026075B000120212
:180178006775B00412026075B00012026775B00812026075B0001202F1
:180190006775B01012026075B00012026775B02012026075B0001202B5
:1801A8006775B08012026002013A7E03DE0302023D7DBBDD030201B410
:1801C000408D7F04DF030201D6C294120259D2941202590201C4C29766
:1801D80075B000120267D297120260C29712026775B00112026075B0FF
:1801F0000012026775B00212026075B00012026775B00412026075B07F
:180208000012026775B00812026075B00012026775B01012026075B054
:180220000012026775B02012026075B00012026775B080120260D29374
:18023800A2B60201BBC29412026E75901F75B0007587027A53DAFE22B2
:180250007A217BFADBFEFAFA22120250120250221202591202502212C8
:180268000260120259227A107BA67CFADCFEDBFADAF622C28CCDD0C02D
:18028000E0C002C003C004C005C006C007D29208B8FF0DC2921809898D
:180298000C06D29279007800758CF5758ACAD007D006D005D004D003FF
:0902B000D002D0E0D0D0D28C3293
:00000001FF
```

Преобразователь сигналов дифференциального емкостного датчика

Описывается преобразователь сигналов дифференциально-го емкостного датчика, представлены рекомендации по его расчету, регулированию и применению.

Для преобразования линейных и угловых перемещений в системах автоматизации технологических процессов применяют дифференциальные емкостные датчики, характеризующиеся высокой чувствительностью к преобразуемому параметру, малой инерционностью и незначительным уровнем шумов [1]. Поскольку выходные сигналы таких датчиков отличаются малой мощностью, преобразователи, в которых выходной сигнал датчика формируется в виде напряжения, оказываются весьма чувствительными к влиянию паразитных емкостей монтажа и токов утечки [2—4]. Ослабление влияния монтажных емкостей достигается включением каждой емкости датчика последовательно с источниками противофазных синусоидальных напряжений и токовым входом вторичного преобразователя [5]. Однако практи-

ческая реализация такой схемы характеризуется повышенной сложностью, а также существенной инерционностью, что связано с обеспечением необходимых фазовых соотношений и фильтрацией синусоидальных сигналов. Ниже описан преобразователь сигналов дифференциального емкостного датчика в напряжение постоянного тока, особенностями которого являются применение линейно изменяющихся напряжений возбуждения датчика, реализация дифференциальной структуры для уменьшения аддитивной погрешности преобразования, а также уменьшение мультипликативной погрешности преобразования за счет применения микросхемы с согласованными параметрами резисторов.

В состав измерительного канала (рис. 1) входят дифференциальный емкостный датчик Д со встроенным

устройством контроля УК и вторичный преобразователь ВП, содержащий генератор G противофазных линейно изменяющихся напряжений U_1 , U_2 , демпфирующие резисторы $R_1 = R_2 = R$, электрометрический операционный усилитель ОУ с резистором R_{OC} в цепи обратной связи, устройства выборки и хранения УВХ1, УВХ2, дифференциальный и выходной усилители ДУ и ВУ, а также логический узел ЛУ, управляющий УВХ.

Начальные емкости секций датчика составляют $C_{X1,0} = C_{X2,0} = 1$ нФ, а информативный параметр $|C_{X1} - C_{X2}|$ достигает значения 100 пФ, выходной сигнал при этом ВП изменяется в диапазоне $-10...10$ В.

Паразитная емкость монтажа $C_{ПЗ}$ включена параллельно входу ОУ (рис. 1) и практически не влияет на погрешность преобразования, а влияние емкостей $C_{П1}$ и $C_{П2}$ ослаблено за счет соответствующего выбора сопротивлений демпфирующих резисторов R_1 и R_2 . Генератор G при его реализации по релаксационной схеме формирует напряжения U_1 и $U_2 = -U_1$ (рис. 2) с амплитудой A и частотой $f = U_p / (4AR_G C_G)$, где U_p — пороговое напряжение, R_G , C_G — времязадающие сопротивление и емкость.

При выполнении условия $f \ll \ll 1 / \{4R[\max(C_{X1} + C_{П1}, C_{X2} + C_{П2})]\}$,

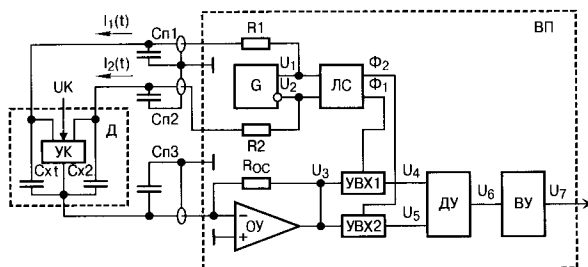


Рис. 1

где $C_{П1}$, $C_{П2}$ — паразитные емкости, форма токов I_1 и I_2 , протекающих через датчик, близка к противофазным прямоугольным импульсам (рис. 2). Амплитудные значения напряжений на выходе ОУ составляют:

$$U_3 \approx \pm 4fAR_{OC}(C_{X1} - C_{X2}) + U_0, \quad (1)$$

где U_0 — смещение нуля, обусловленное параметрами ОУ, токами утечки в цепях датчика и по элементам конструкции. Преобразование сигнала U_3 в напряжения постоянного тока U_4 , U_5 , пропорциональные соответственно положительным и отрицательным полуциклам U_3 осуществляют УВХ1 и УВХ2. Логический узел формирует сигналы управления УВХ Φ_1 и Φ_2 (рис. 2) с активным состоянием лог. 0

в окрестности нулевых значений напряжений U_1 и U_2 , что обеспечивает включение УВХ в режим слежения по окончании переходных процессов токов датчика (рис. 2). С целью компенсации напряжения U_0 выходы УВХ подключены к ДУ, для которого это напряжение является синфазным. При условиях $R_{OC} = K_0 R_G$ и $A = U_p$, выходное напряжение ВП составляет:

$$U_7 \approx 2AK_0K_1K_2K_3(C_{X1} - C_{X2})/C_G, \quad (2)$$

где K_1 , K_2 и K_3 — коэффициенты передачи УВХ, ДУ и ВУ.

Как следует из выражения (2), сигнал U_7 не зависит от абсолютных значений сопротивлений. В частности, при реализации резисторов R_{OC} и R_G на элементах микросхемы

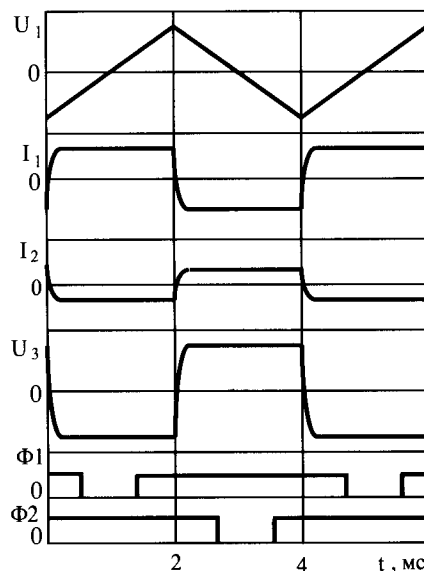


Рис. 2

313HP310 погрешность величины K_0 составляет $\delta K_0 = 2\delta k$, где $|\delta k| \leq 0,035\%$ — погрешность делителя напряжения, составленного из двух произвольно выбранных резисторов микросхемы.

По дискретному сигналу УК инициируется режим функционального контроля при котором УК изменяет информативный параметр датчика,

ИНТЕЛЛЕКТУАЛЬНЫЕ ПАЯЛЬНЫЕ СТАНЦИИ

Паяльные станции с единым микропроцессорным блоком управления (Dig-203A), к которому может быть подключен один из паяльных инструментов: паяльник TechTool, паяльник MicroTool, паяльник PowerTool, термопинцет Pincette-40 или вакуумный термоотсос X-Tool.

- выбор единицы измерения °C/°F
- установка продолжительности режима stand-by
- функция калибровки и регулировки погрешности
- блокировка регулировок пайки посредством цифрового пароля

Dig-2000A-Tech

универсальная паяльная станция
70 Вт, 350°C, разогрев за 12 с

Dig-2000A-ChipTool

паяльная станция с термопинцетом
30 Вт, 350°C

Dig-2000A-PowerTool

паяльная станция для пайки
30 Вт, 350°C, д

Москва, ул. Ивана Франко, д. 40, стр. 2

Тел./факс: (495) 97 000 99

Почта: 121351, Москва, а/я 100

E-mail: platan@aha.ru

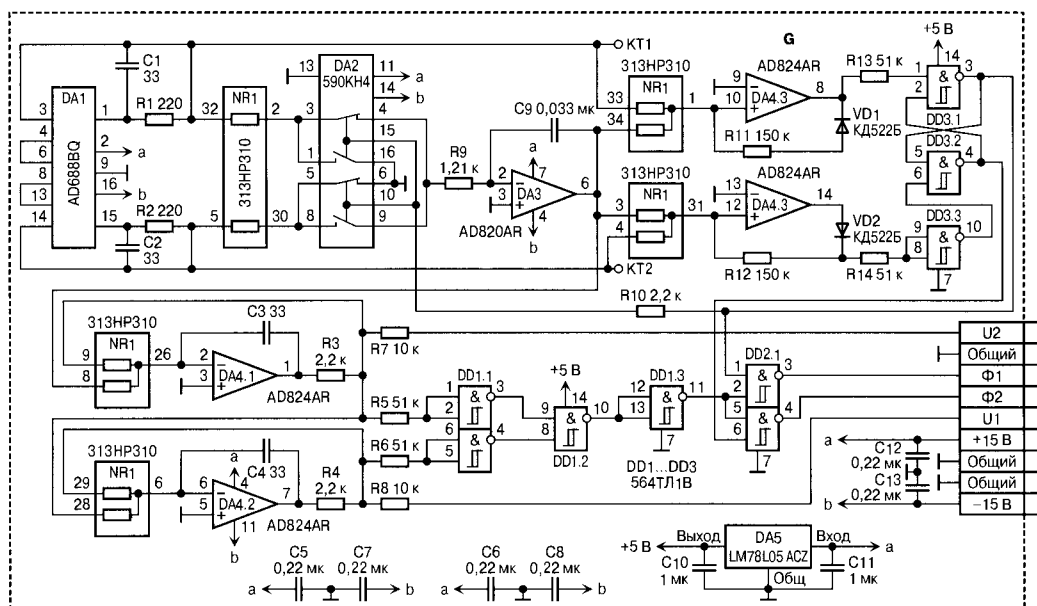


Рис. 3

что приводит к прогнозируемому изменению выходного сигнала ВП.

Поскольку выходное напряжение ВП в значительной степени определяется параметрами сигналов U_1 и U_2 , к реализации генератора предъявляются повышенные требования по точности. В состав генератора (рис. 3) входят источник опорных напряжений $\pm U_0$ DA1, электронный ключ DA2, интегратор DA3, C9, R9 с зарядными резисторами сборки NR1, инвертирующие усилители и пороговые элементы, построенные на основе резисторов сборки NR1 и ОУ DA4, а также логический узел DD1—DD3, питание которого осуществляется от стабилизатора напряжения DA5.

Сигналы датчика поступают на вход нормирующего преобразователя (рис. 4), в состав которого входят преобразователь тока, выполненный на основе резисторов сборки NR1 и ОУ DA6.1, инвертирующие УBX1 DA7.1, C17 и УBX2 DA7.2, C18, дифференциальный усилитель DA8, R27 с фильтром входного сигнала R17—R20, C19, и выходной инвертирующий усилитель, реализованный на основе ОУ DA6.4 с резисторами регулирования масштаба R24 и начального значения R28 выходного напряжения.

Вход усилителя DA6.1 снабжен цепью защиты от повышенного напряжения R15, VD7, VD8. Диодные ограничители VD3—VD6 падения напряжения на ключах DA7.1, DA7.2 способствуют уменьшению токов утечки этих ключей в разомкнутом состоянии.

Как следует из выражения (2), для описываемой схмотехнической реали-

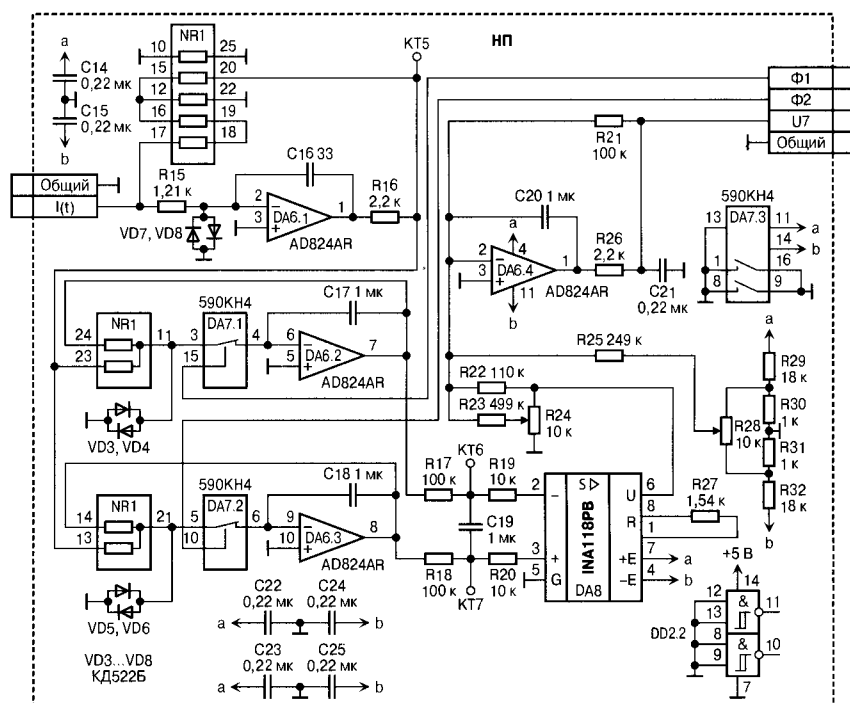


Рис. 4

зации ВП мультипликативная погрешность преобразования составляет:

$$\delta U_7 \approx \delta U_0 + \delta C_9 + 6\delta k + \delta K_2 + \delta R_{21} + s_0 \delta R_{22} + s_1 \delta R_{23}, \quad (3)$$

где δ — символ относительной погрешности, $s_0 = R_{23}/(R_{23} + xR_{22})$, $s_1 = x s_0 R_{22}/R_{23}$ — коэффициенты чувствительности ($0 \leq x \leq 1$ — положение движка резистора R_{24}), $\delta K_2 \approx 0,36 + \delta R_{27}$ — погрешность коэффициента передачи ДУ, %.

Функциональный контроль измерительного канала в процессе его эксплуатации осуществляется с помощью УК (рис. 5), при активизации которого конденсатор C_K в зависимости от полярности напряжения УК контактами реле (K1, K2) подклю-

чается параллельно одной из секций датчика (C_{X1} или C_{X2}), что приводит к изменению информативного параметра датчика.

Регулирование начального значения и масштаба выходного сигнала ВП, а также проверку его параметров на стадии изготовления выполняют с применением имитатора, обеспечивающего воспроизведения сигнала датчика с погрешностью не хуже $\pm 0,5\%$ в 11-ти точках рабочего диапазона. Регулирование ВП в составе измерительного канала заключается в установке начального и максимального значений выходного сигнала ВП, составляющих соответственно $0 \pm 0,01$ и $10 \pm 0,01$ В при равновесном положении и максимальном перемещении

чувствительного органа датчика. С учетом регулирования погрешность ВП определяется преимущественно дополнительной температурной погрешностью $\theta(t_{OS}) = \delta U_7(t_{OS}) - \delta U_7(20^\circ\text{C})$, где величины $\delta U_7(t_{OS})$ и $\delta U_7(20^\circ\text{C})$ вычислены по формуле (3) при значениях температуры окружающей среды t_{OS} и 20°C . При этом дополнительная температурная погрешность коэффициента передачи ДУ составляет $\delta K_2(t_{OS}) - \delta K_2(20) = 0,001 \cdot (t_{OS} - 20) \cdot \delta K_2(20)$.

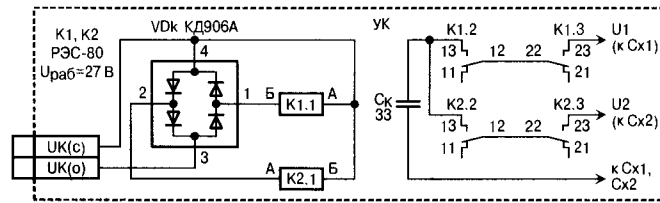
В ВП применены резисторы C2-29B-0,125 (R21—R23, R27 с допуском $\pm 0,1\%$, R17—R20, R29—R32 с допуском $\pm 1\%$), а также резисторы C2-23H-0,125 с допуском $\pm 5\%$. Регулировочные резисторы — C5-2BB-0,5. Конденсатор C9 — K71-7 с допуском $\pm 0,5\%$. Конденсаторы C1—C4, C16, C_к — K10-17a (M750) с допуском $\pm 5\%$, C5—C8, C10—C15, C21—C25 — K10-17a — (H90), остальные — K73-16 на напряжение до 63 В. Вместо резисторов сборки серии 313 могут применяться две сборки 301 HP7 или высокоточные резисторы C2-29B. Операционные усилители DA3, DA4, DA6 могут быть заменены аналогичными, например, микросхемами K544УД15. Преобразователь смонтирован на пла-

Рис. 5

те с двухсторонним расположением элементов. Трассировка цепи общего провода выполнена по «лучевой» схеме, фильтрующие конденсаторы в цепях питания равномерно распределены по поверхности платы, а инвертирующие входы операционных усилителей DA6.2, DA6.3 экранированы.

Синтез ВП удобно выполнять с помощью расчетной методики в виде файла Edt.mcd, в формате MathCad 2000 Pro, размещенного на сайте журнала по адресу www.dian.ru/programs/.

Таким образом, применение описанных выше технических решений и разработанной расчетной методики позволяет реализовать на распространенной элементной базе измерительные каналы с дифференциальными емкостными датчиками, удовлетворяющие весьма высоким требованиям по точности.



ЛИТЕРАТУРА:

1. В. А. Арцюковский. Емкостные дифференциальные датчики перемещения. — Л.: ГЭИ, 1960.
2. А. Вострухин, И. Минаев. Измерительный преобразователь емкости в постоянное напряжение. — Схемотехника, 2003, № 5, с. 2, 3.
3. В. С. Мелентьев. Методы и средства измерения параметров емкостных дифференциальных датчиков. — Датчики и системы, 2005, № 5, с. 36—38.
4. В. Я. Распопов. Измерительные цепи и передаточные функции акселерометров прямого преобразования. Датчики и системы, 2006, № 5, с. 31—34.
5. Е. С. Левишина, П. В. Новицкий. Электрические измерения физических величин. — Л.: Энергоатомиздат, 1983.

Юрий Пахоменков,
г. С.-Петербург

Простое радиопереговорное устройство на 27 МГц

«Мастер КИТ» предлагает набор NF275, позволяющий собрать маломощные радиопереговорные устройства, не требующие сертификации и разрешения.

Мобильный телефон стал необходимым, как воздух, атрибутом каждого цивилизованного человека. «Урбанизация» — скажет житель небольшого городка или деревеньки и будет по-своему прав — есть случаи, когда вполне возможно обойтись без карманного «цветного брата». Чтобы поговорить с соседом по квартирной площадке, приятелем на рыбалке или с другом, живущим в соседнем доме, совсем не обязательно тратить средства на телефонный звонок или отправку sms, достаточно собрать комплект из двух простейших радиостанций и просто нажать на тангенту! Эксплуатация подобной маломощной техники (10 мВт) не требует получения специального разрешения, т. к. радиопереговорные устройства, работающие в по-

лосе радиочастот 26957...27283 кГц с допустимой мощностью излучения передатчика до 10 мВт не требуют сертификации.

Технические характеристики

Напряжение питания, В	9
Ток потребления в режиме приема, мА	30
Ток потребления в режиме передачи, мА	50
Мощность передатчика, мВт	10
Частота излучения, МГц	27,000
Дальность связи при прямой видимости, м	до 150
Размеры печатной платы, мм	60×60

Общий вид приемопередатчиков приведен на рис. 1, а принципиальная схема — на рис. 2.



В режиме передачи динамическая головка выполняет функцию микрофона. Через регулятор громкости VR1 и разделительный конденсатор C9 сигнал поступает на усилитель низкой частоты, выполненный на транзисторах VT2—VT4. Индуктивность первичной обмотки низкочастотного трансформатора T1, включенная в качестве нагрузки транзисторов VT3 и VT4, играет роль модуляторного дросселя. С него усиленный сигнал поступает в высокочастотный тракт.

При нажатии на кнопку SB1 в каскад усиления вводится глубокая положительная обратная связь, узел

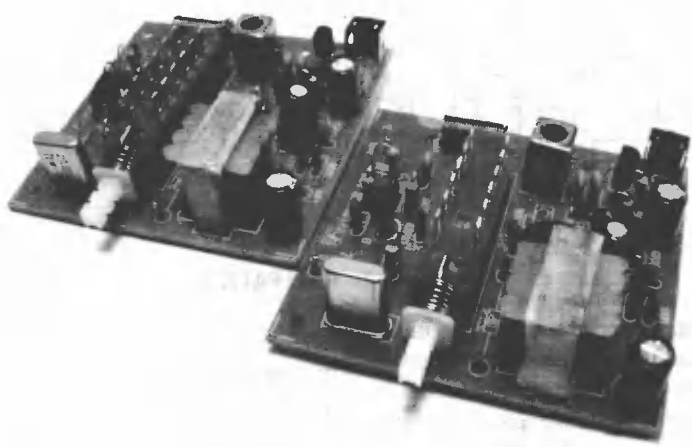


Рис. 1

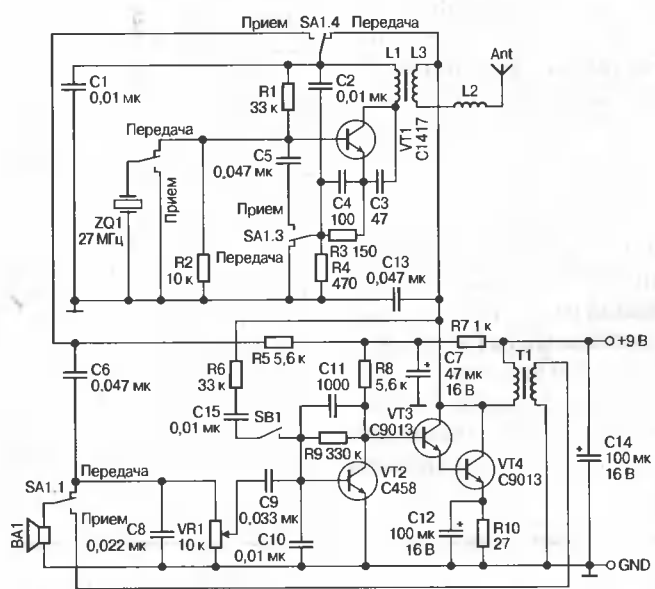


Рис. 2

на транзисторах VT2 и VT3 возбуждается и генерирует сигнал низкой частоты («Вызов»), поступающий в ВЧ-тракт.

Генератор высокой частоты выполнен на транзисторе VT1, он стабилизирован кварцевым резонатором на частоту 27 МГц. Напряжение на генератор, как указывалось выше, подается с первичной обмотки трансформатора T1. Таким образом, сигнал высокой частоты 27 МГц оказывается промодулированным по амплитуде сигналом звуковой частоты. Через высокочастотный трансформатор L1L3 сигнал поступает на удлинительную катушку L2, а далее — в антенну.

В режиме приема каскад на транзисторе VT1 является сверхрегенеративным детектором. С его выхода через конденсатор C6 низкочастотный сигнал поступает на регулятор громкости VR1, с него — на усилитель мощности на транзисторах VT2—VT4. Через согласующий трансформатор T1 усиленный сигнал поступает на динамическую головку BA1.

Приемопередатчик выполнен на печатной плате из фольгированного текстолита размерами 60×60 мм.

Для тех, кто захочет повторить это занимательное устройство самостоятельно, приводим данные электронных компонентов в табл. 1, а на рис. 3 и 4 — рисунки печатной платы.

ВЧ трансформатор выполнен на пластмассовом цилиндрическом каркасе диаметром 3 мм, его первичная обмотка L1 содержит 5 витков провода ПЭВ-0,2, вторичная L3 — 2 витка того же провода. Вторичная обмотка намотана ниже первичной. Трансформатор имеет подстроечный ферритовый сердечник марки 50 ВЧ диаметром 2,5 мм длиной 8 мм и заключен в металлический экран.

Катушка L2 намотана на оправке диаметром около 6...7 мм, число витков около 20.

Первичная обмотка НЧ трансформатора содержит 2000 витков, намотанных проводом ПЭВ-0,05, вторич-

Таблица 1

Позиция	Номинал	Кол.
R1, R6	33 кОм	4
R2	10 кОм	2
R3	150 Ом	2
R4	470 Ом	2
R5, R8	5,6 кОм	4
R7	1 кОм	2
R9	330 кОм	2
R10	27 Ом	2
VT1	C1417	2
VT2	C458	2
VT3, VT4	C9013	4
C1, C2	10 нФ	8
C10, C15	47 пФ	2
C3	47 пФ	2
C4	100 пФ	2
C5, C6, C13	47 нФ	6
C7	47 мкФ/16 В	2
C8	22 нФ	2
C9	33 нФ	2
C11	1000 пФ	2
C12, C14	100 мкФ/16 В	4
VR1	10 кОм	2
L1, L3	ВЧ трансформатор	2
L2	Катушка (или провод для намотки)	2
T1	НЧ трансформатор	2

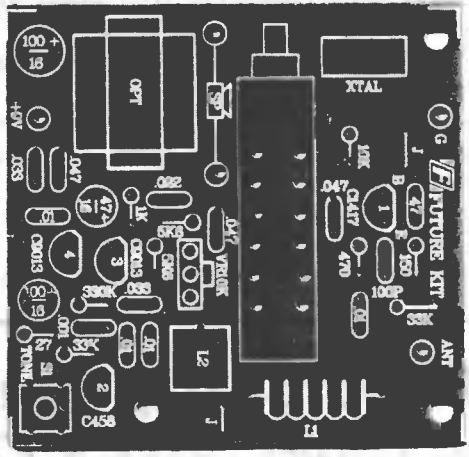


Рис. 3

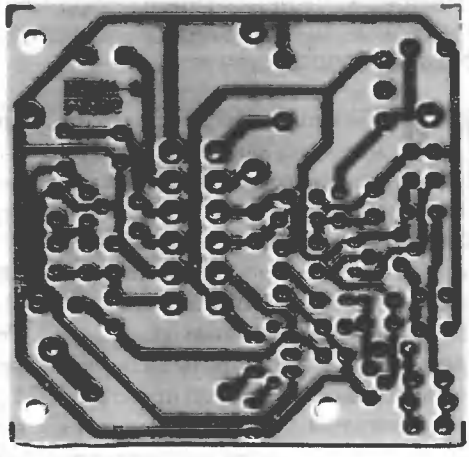


Рис. 4

ная — 40 витков провода ПЭВ-0,2. Трансформатор намотан на сердечнике Ш-образной формы габаритами 18×14×4 мм из трансформаторной стали.

Настройку приемопередатчиков лучше проводить с помощником на

открытой местности, вдали от железобетонных зданий и металлических конструкций.

Расположите платы на некотором расстоянии друг от друга (начните с нескольких метров). Подключите к платам антенны. Оптимальная длина антенны определяется опытным путем. Лучшие результаты будут достигнуты с длиной антенны, равной четверти длины волны излучения (для частоты 27 МГц — около 2,6 м), но, по понятным причинам, для достижения мобильности устройств длину антенн приходится уменьшать до 20...30 см.

Несколько компенсирует вынужденное уменьшение длины антенны катушка L2, меняя число витков которой можно настроить для совместной работы с приемопередатчиком антенну любой длины.

Подключите динамические головки и напряжение питания к обоим платам, соблюдая полярность.

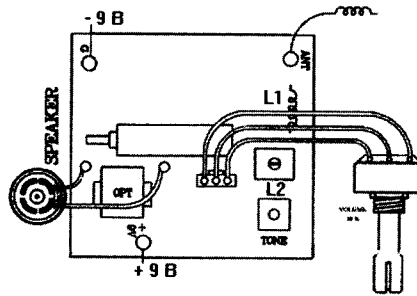


Рис. 5

Переведите переключатель одного из приемопередатчиков в режим «Передача» и говорите в микрофон. Переключатель другого приемопередатчика в это время должен находиться в положении «Прием», и в его динамической головке должна прослушиваться передаваемая речь. Нажмите кнопку SB1 — в приемнике должен раздаться сигнал «Вызов».

Удалите приемник от передатчика на максимальное расстояние, на котором связь еще устанавливается.

В режиме «Передача» медленно вращайте тонкой отверткой (лучше с диэлектрическим жалом) сердечник трансформатора L1L3, а также сдвигайте и раздвигайте витки катушки выходной обмотки. Добившись улучшения, переместите приемник еще дальше и повторите манипуляции. Действуя таким образом, добейтесь максимального радиуса действия, после чего зафиксируйте сердечник и витки катушки несколькими каплями парафина.

Аналогичные настройки проведите и для второй платы приемопередатчика.

В процессе эксплуатации переменными резисторами можно установить желаемый уровень громкости.

Юрий Садиков,
г. Москва

Чтобы сэкономить время и избавить вас от рутинной работы по поиску необходимых компонентов и изготовлению печатных плат, «Мастер КИТ» предлагает набор NF275 «Приемопередатчик 27 МГц». Набор состоит из двух печатных плат, всех необходимых компонентов и инструкции по сборке, настройке и эксплуатации.

Более подробно ознакомиться с ассортиментом нашей продукции можно с помощью «CD-каталога Мастер КИТ-2006» и на сайте www.masterkit.ru, где представлено много полезной информации по электронным наборам и модулям «Мастер КИТ», приведены адреса магазинов, где их можно купить.

Наборы, блоки и модули «Мастер КИТ», а также журналы «Схемотехника» можно купить в магазинах радиодеталей вашего города.

За справками можно обратиться по тел. (495) 234-77-66, по почтовому адресу: МАСТЕР КИТ, А/Я 19, Москва, 109044, Россия.

Желаем вам приятных покупок!

Новые книги

А. В. Фрунзе

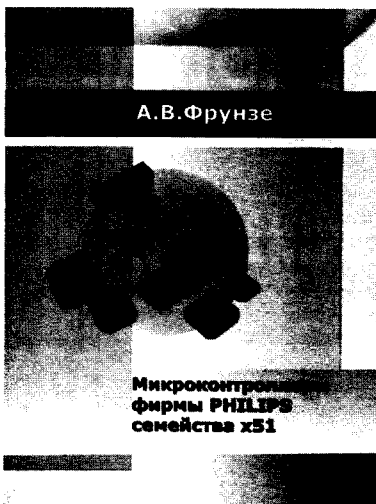
МИКРОКОНТРОЛЛЕРЫ ФИРМЫ PHILIPS СЕМЕЙСТВА x51

Издательский дом «Скимен» выпустил первый том книги Александра Фрунзе «Микроконтроллеры фирмы Philips семейства x51». В этой книге приведен подробный обзор всех выпускаемых фирмой Philips микроконтроллеров, дана информация по доступным отечественному пользователю микроконтроллерам семейства x51, изготавливаемым другими ведущими мировыми производителями. В ней приведена самая свежая на момент выпуска сводная таблица по всем выпускаемым Philips микроконтроллерам, а также терминологический словарь. Далее подробно рассматриваются особенности классических микроконтроллеров семейства x51 — система команд (подробно описаны все команды микроконтроллеров с кодами операций, операндами, примерами действия команд, временем их исполнения) и аппаратные особенности (структура памяти и регистров, арифметическо-логическое устройство, таймеры-счетчики и режимы их работы, система прерываний, стандартный последовательный порт со всеми режимами работы и система управления питанием). Подробно описаны микроконтроллеры с разработанным фирмой Philips оригинальным ядром 80C51+, в том числе микроконтроллеры с флэш-памятью программ, с программируемой матрицей счетчиков, с расширенной памятью программ и данных. Рассмотрены также микроконтроллеры с дополнительными портами ввода/вывода, с аппаратно реализованной шиной I²C и встроенным аналого-цифровым преобразователем. Вся приведенная информация основана на справочных материалах фирмы Philips.

Книга рассчитана на широкий круг инженерно-технических работников, интересующихся последними достижениями микроконтроллерной техники.

Книга формата А4 имеет мягкую обложку и содержит 336 с.

Вы можете подписаться на эту книгу через редакцию. Стоимость подписки — 260 руб., в эту сумму включена пересылка по России. Пример заполнения квитанции для оплаты через банк см. на стр. 56.



ПОДПИСКА — 2007

Журнал «СХЕМОТЕХНИКА»

Редакционная подписка на 2007 г. — 600 рублей.

Через каталог Агентства «РОСПЕЧАТЬ» (красный)

- индекс 80724 — подписка на 1-е полугодие 2007 г.;
- индекс 82117 — подписка на архив журнала за 2004 г. на CD.

Через каталог «ПОЧТА РОССИИ»

- индекс 60192 — подписка на 1-е полугодие 2007 г.
- индекс 60193 — годовая подписка на 2007 г.

Оформить подписку с последующей доставкой в ЛЮБОЙ СТРАНЕ можно через подписное агентство «МК-PERIODICA».

Подробная информация по телефонам в Москве 7+495 681-9345, 681-5715, 681-3322, e-mail: info@periodicals.ru

Подписка на Украине осуществляется также через каталог Подписного агентства KSS (тел: в Киеве 044 - 270-6220, 270-6222), подписной индекс 10540.

Подписка через редакцию по тел. (495)775-16-76, podpiska@dian.ru.

Для подписки через редакцию

- перечислите деньги на наш расчетный счет через Сбербанк по квитанции, либо через почтовое отделение почтовым переводом;
- отправьте квитанцию об оплате (или копию) и свой точный почтовый адрес (индекс обязательно) в редакцию по почтовому адресу: 127015, Москва, ул. Бутырская, д. 41/47, ООО «ИД Скимен», отдел подписки или по факсу (495) 775-16-76, e-mail: podpiska@dian.ru.
- вы можете также подписаться на журнал непосредственно в редакции по адресу: г. Москва, Ленинградское ш., д. 18, офис 342 (метро «Войковская», из первого вагона от центра направо, вперед по ходу поезда до последнего здания перед железной дорогой, вход со стороны д. 16) и на специализированных выставках.

Пример заполнения банковского извещения для подписки на издания ИД «СКИМЕН»:

Извещение	ООО «ИД СКИМЕН» ИНН/КПП 7731195492/773101001 в ОАО «НК банк» в г. Москва Р/с 40702810200000005646 К/с 30101810900000000278 БИК 044579278 Платательщик Адрес (с индексом)	
	Назначение платежа	Сумма
Кассир	Подпись _____ 2006 г.	
Квитанция	ООО «ИД СКИМЕН» ИНН/КПП 7731195492/773101001 в ОАО «НК банк» в г. Москва Р/с 40702810200000005646 К/с 30101810900000000278 БИК 044579278 Платательщик Адрес (с индексом)	
	Назначение платежа	Сумма
Кассир	Подпись _____ 2006 г.	