

Главный редактор

Сергей Бирюков

Зам. главного редактора

Сергей Кузнецов

Редакционная коллегия

Павел Асташкевич
Александр Фрунзе
Виктор Иовчик

Дизайн и верстка

Ирина Чикина

Отдел распространения

(095) 777-12-15
e-mail: sales@dian.ru
Марина Трофимова
Юрий Царев
Сергей Лукин

Отдел рекламы

Юлия Суханова

Адрес редакции:

127015 Москва, ул. Бутырская, д. 41/47
«ИД Скимен»
тел./факс: (095) 777-12-15
www.dian.ru
e-mail: editor@dian.ru
editordian@mail.ru

Издатель и учредитель
ООО «ИД Скимен»

Отпечатано в ОАО «Чеховский
полиграфический комбинат»

Тираж 5 250 экз.
Заказ № 39 6 2.

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

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

- | | |
|--|----|
| С. Бирюков. Микросхемы для импульсных источников питания NCP1000, NCP1001, NCP1002 фирмы ON Semiconductor | 2 |
| О. Николайчук. Анализ SFR-совместимости микроконтроллеров фирмы Silabs | 6 |
| О. Вальпа. Цифровые сигнальные процессоры | 10 |

ИСКУССТВО СХЕМОТЕХНИКИ

- | | |
|--|----|
| А. Фрунзе, А. Фрунзе. Микроконтроллеры? Это же просто! | 12 |
| С. Лозицкий. Уменьшение пульсаций в блоке питания с емкостным балластом | 15 |
| А. Титов. Широкополосный усилитель мощности с защитой от перегрузки | 18 |
| В. Вычужанин. Цифровой комбинированный регулятор на ПЛИС | 21 |

СОФТ

- | | |
|--|----|
| О. Вальпа. Borland C++ Builder 6 для начинающих | 24 |
| А. Сошкин, А. Мамонтов. Подготовка программ для микроконтроллеров семейства x51 в среде µVISION-2 | 27 |
| Г. Кардашев. Компьютерное схемотехническое моделирование электронных устройств | 30 |
| Пакет Acronis | 34 |

ПРАКТИКА

- | | |
|--|----|
| Н. Заец. Устройство подачи звонков по расписанию | 35 |
| Ю. Килиба. ИК датчик препятствия | 38 |
| С. Лузгин. Лампа—часы—термометр на микроконтроллере Atmel ATmega16L-8PI | 40 |
| В. Захаров. Электронная игрушка — автомат световых эффектов | 44 |
| В. Цибин. О керамических конденсаторах | 47 |
| Цифровые часы на основе микроконтроллера HT49R30A-1 | 48 |
| В. Баранов. Изготовление печатных плат для макетов | 50 |
| Ю. Садиков. Автомобильный электронный балласт для переносной лампы | 52 |
| С. Шишкин. Таймер на микроконтроллере с индикатором фирмы Data Vision | 53 |

CD-ROM с журналами «Схемотехника» (3-я стр. обложки).

Новости компании «ЧипЭКСПО» (с. 9). Вниманию читателей (с. 29, 39).
Новые книги. А. Фрунзе. «Микроконтроллеры фирмы Philips семейства x51» (с. 48). Подписка-2005 (с. 56).

На нашей обложке. Владимир Захаров демонстрирует «Волшебную палочку» (см. статью на с. 44).

Микросхемы для импульсных источников питания NCP1000, NCP1001, NCP1002 фирмы ON Semiconductor

В статье рассматриваются микросхемы, служащие для построения сетевых импульсных обратно- и пряموходовых импульсных источников питания и содержащие большинство элементов первичной цепи таких устройств, включая высоковольтный ключ, собранный на паре полевых транзисторов.

Контроллер микросхем [1] работает на фиксированной частоте 100 кГц, для стабилизации выходного напряжения используется широтно-импульсная модуляция (ШИМ, PWM). Встроенный ключ на полевых транзисторах и цепь его запуска обеспечивают длительную работу источников на основе этих микросхем в широком диапазоне входного выпрямленного напряжения сети переменного тока — 85...265 В, что позволяет использовать их во всем мире. Микросхемы обладают всеми необхо-

димыми для них свойствами, такими как защита от перегрева и работы при слишком низком входном напряжении, встроенный источник опорного напряжения, блокировка ограничения выходного тока в момент открывания ключа и многими другими.

Основные свойства источников на этих микросхемах:

- возможность работы в широком диапазоне входных напряжений;
- встроенный ключ на 700 В;
- минимальное число внешних элементов;

- блокировка от работы при слишком низком входном напряжении;
- малый ток потребления при работе без нагрузки;
- защита от выхода из строя оптопары обратной связи;
- отсутствие необходимости в начальной нагрузке;
- защита от перегрева.

Микросхемы оформляются в стандартных восьмивыводных корпусах DIP.

Основные применения:

- зарядные устройства для сотовых телефонов;
- источники питания в сетевой вилке (адаптеры);
- встроенные источники питания маломощных устройств.

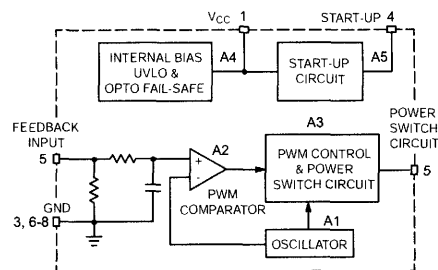


Рис. 1

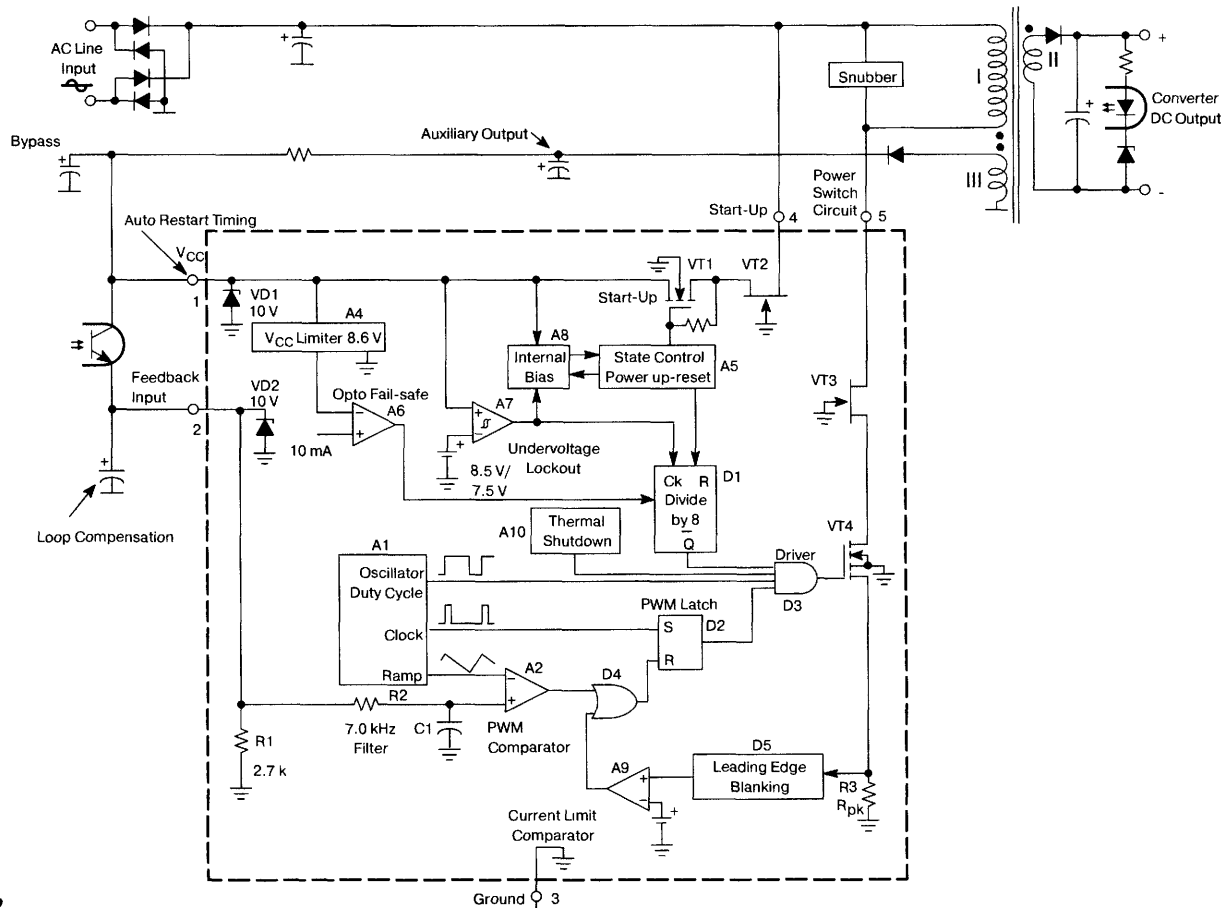


Рис. 2

Упрощенная структура микросхем показана на рис. 1. Микросхемы содержат задающий генератор А1, компаратор ШИМ А2, выходной ключ с цепями управления А3, встроенный источник опорного напряжения и цепь защиты от выхода из строя оптопары А4, цепь начального запуска А5.

Более подробно структура микросхем вместе с типовой схемой включения приведены на рис. 2. Задающий генератор А1 содержит два компаратора, настроенные на пороговые напряжения около 2,7 и 4,1 В. Компараторы управляют включением и выключением двух генераторов тока, один из которых заряжает, другой разряжает времязадающий конденсатор генератора. В результате на конденсаторе формируется пилообразное напряжение с размахом, соответствующим уровням настройки компараторов.

Генератор А1 имеет три выхода. Импульсы на верхнем по схеме выходе имеют коэффициент заполнения 72 % и поступая на вход элемента совпадения D3, управляющего выходным ключом VT3VT4, задают такой же максимальный коэффициент заполнения выходных импульсов микросхемы. Импульсы с выхода Clock генератора устанавливают в начале каждого цикла триггер D2 в единичное состояние, что включает выходной ключ.

Пилообразное напряжение с выхода Ramp генератора А1 поступает на инвертирующий вход компаратора ШИМ А2, на второй его вход через фильтр НЧ R2C1 подается напряжение с входа обратной связи (вывод 2). Компаратор сравнивает эти напряжения и формирует импульсы с коэффициентом заполнения, линейно уменьшающемся при увеличении напряжения на входе 2 микросхемы, что обеспечивает стабилизацию выходного напряжения преобразователя. Напряжение на входе 2 определяется током фотодиода внешней оптопары и сопротивлением встроенного в микросхему резистора R1 с номиналом 2,7 кОм. Типовая зависимость коэффициента заполнения от тока обратной связи приведена на рис. 3.

Интегрирующая цепочка R2C1 снижает уровень высокочастотных шумов на входе компаратора и является основным элементом, обеспечивающим устойчивость цепи обратной связи преобразователя. Стабилитроны VD1 и VD2 служат для защиты соответствующих входов от статичес-

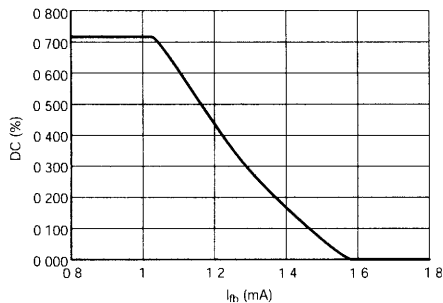


Рис. 3

кого электричества. Постоянная времени этой цепочки составляет около 23 мкс, что соответствует частоте полюса приблизительно 7 кГц.

Диаграммы на рис. 4 иллюстрируют процессы, происходящие в преобразователе при снижении напряжения на входе обратной связи, например, при увеличении тока нагрузки. На верхней паре диаграмм показаны пилообразное напряжение на выходе Ramp генератора А1 и снижающееся напряжение обратной связи. Следующие две диаграммы — это импульсы на верхнем по схеме выходе генератора, определяющие максимальный коэффициент заполнения выходных импульсов микросхемы (Oscillator Duty Cycle), и импульсы, устанавливающие своим фронтом триггер D2 в единичное состояние (Oscillator Clock). На средней диаграмме приведен выходной сигнал компаратора ШИМ А2 (PWM Comparator Output), а на следующей — состояние триггера D2 (PWM Latch Q Output). Результат взаимодействия выходного сигнала триггера с импульсами, ограничивающими максимальный коэффициент заполнения, показан на предпоследней диаграмме. Нижняя диаграмма иллюстрирует форму импульсов тока через силовой ключ и первичную обмотку трансформатора.

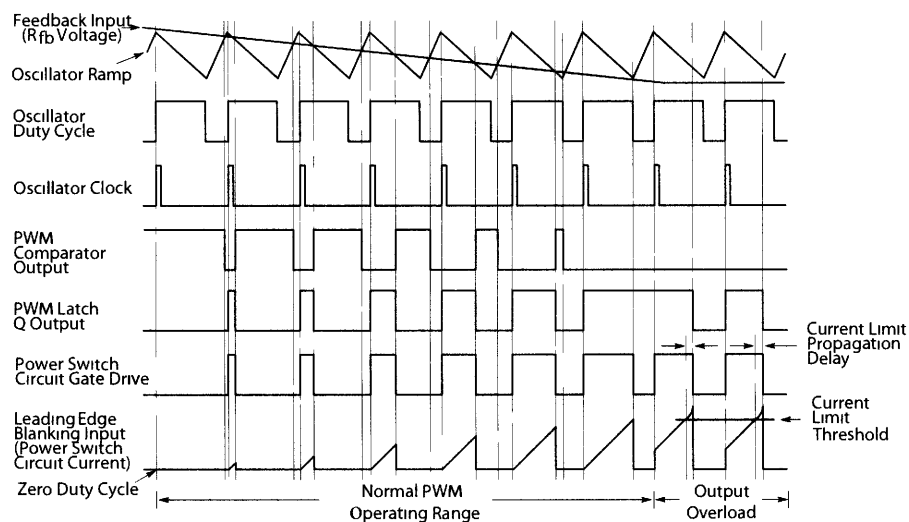


Рис. 4

Пока напряжение на входе обратной связи превышает максимальное напряжение выходного сигнала генератора (первый цикл на диаграммах), происходит пропуск тактов (Zero Duty Cycle). По мере снижения напряжения на выводе 2 увеличиваются коэффициент заполнения выходных импульсов микросхемы и передаваемая в нагрузку мощность, что должно стабилизировать выходное напряжение. Когда напряжение обратной связи становится ниже минимального напряжения «пилы», коэффициент заполнения ограничивается величиной 72 %, стабилизация прекращается, микросхема переходит в режим перегрузки.

Рассмотрим работу блока питания без нагрузки. В этом случае преобразователь начинает пропускать циклы, это типовой вариант работы импульсных источников. Преобразователь включается на несколько циклов, затем следует пауза в его работе. Во время паузы напряжение на выходе преобразователя медленно спадает за счет разрядки выходного конденсатора через ограничительный резистор и светодиод оптрона обратной связи, а микросхема получает питание от конденсатора, подключенного к выводу 1 микросхемы (Bypass). Для нормальной работы микросхемы важно, чтобы постоянная времени разрядки этого конденсатора была больше, чем выходного конденсатора преобразователя. Если работа преобразователя без нагрузки не предполагается, емкость конденсатора, подключенного к выводу 1, может быть относительно небольшой (<10 мкФ).

При работе микросхем для исключения перегрузки осуществляется позиционный контроль тока через выходной ключ. Для осуществления такой

функции транзистор VT4 выполнен с двумя истоками, в цепь одного из них включен токоизмерительный резистор R3. При превышении падения напряжения на этом резисторе пороговой величины срабатывает компаратор A9 и через элемент ИЛИ D4 сбрасывает триггер D2, обрывая импульс тока через выходной ключ.

В момент открывания выходного ключа ток зарядки емкости обмоток трансформатора вместе с приведенным к первичной обмотке током восстановления обратного сопротивления выпрямительного диода вторичной цепи могут значительно превысить нормальный ток через ключ. Для того, чтобы он при этом не выключался, в цепь контроля тока введен узел D5, служащий для подавления сигнала выключения, если его длительность не превышает 220 нс. На это время подавляется сигнал превышения тока и в других случаях, например, при вхождении сердечника трансформатора в насыщение, поэтому следует иметь ввиду, что пороговое значение тока выключения зависит от скорости его нарастания. Оценить увеличение $\Delta I_{\text{ПОР}}$ порогового значения тока можно по формуле:

$$\Delta I_{\text{ПОР}} = dI/dT \times 220 \text{ нс},$$

где $dI/dT = U_1/L_1$, а U_1 и L_1 — постоянное напряжение питания преобразователя и индуктивность первичной обмотки трансформатора соответственно.

Для обеспечения питания в процессе запуска рассматриваемые микросхемы имеют специальные встроенные цепи. После запуска преобразователя они отключаются, и микросхема переходит на питание от дополнительной вторичной обмотки трансформатора (III на рис. 2), что повышает экономичность устройства.

Выпрямленное и сглаженное сетевое напряжение подводится к выводу 4 микросхемы (рис. 2). Полевой транзистор VT2 работает как стабилизатор

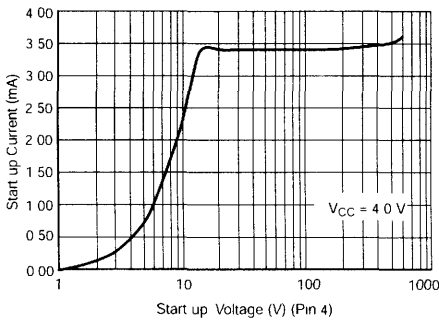


Рис. 5

тока силой около 3 мА. Через ключ на транзисторе VT1 этот ток поступает на вывод 1 (V_{CC}) и заряжает подключенный к нему конденсатор. Зависимость тока через вывод 1 от напряжения на выводе 4 показана на графике на рис. 5. Напряжение на выводе 1 повышается, и когда оно

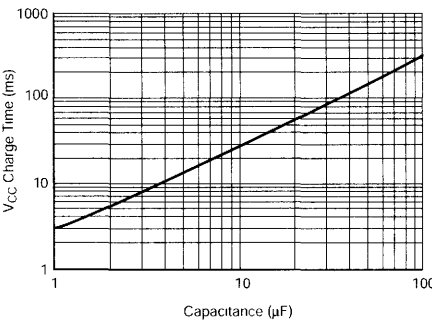


Рис. 6

достигнет уровня 8,5 В, срабатывает компаратор A7 и через узлы A8 и A5 останавливает процесс зарядки конденсатора. Узлы микросхемы начинают получать питание от конденсатора, подключенного к выводу 1. Одновременно переключается триггер D1 и разрешает прохождение импульсов генератора через драйвер D3 на затвор ключевого транзистора VT4, преобразователь начинает работать. Питание на микросхему поступает через выпрямитель с дополнительной

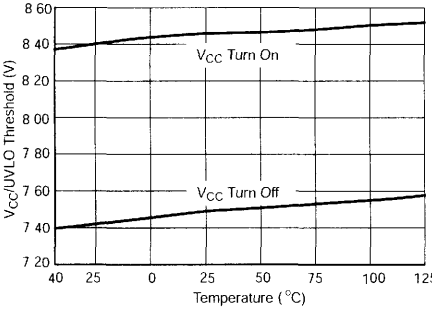


Рис. 7

обмотки III трансформатора. Типовая зависимость времени зарядки конденсатора от 0 до 8,5 В от его емкости приведена на рис. 6.

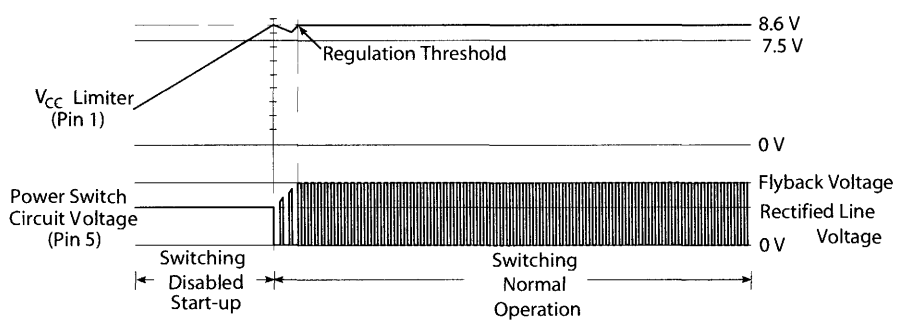


Рис. 8

Если в процессе работы напряжение на выводе 1 снизится до уровня менее 7,5 В, компаратор A7 вновь включит стабилизатор тока зарядки конденсатора, подключенного к выводу 1, и переключит триггер D1 в состояние, запрещающее работу ключа VT3VT4.

Типовые температурные зависимости порогов включения и выключения преобразователя по напряжению на выводе приведены на рис. 7.

При замыкании выхода преобразователя или его перегрузке устройство переходит в режим автозапуска. Условием перехода в этот режим является снижение напряжения на выводе V_{CC} до уровня менее 7,5 В за счет того, что напряжение на обмотке III трансформатора в этом случае падает и не в состоянии поддержать нормальную работу микросхемы. Преобразователь прекращает работу, включается стабилизатор тока зарядки конденсатора на транзисторах VT1, VT2. Конденсатор, подключенный к выводу 1, заряжается до верхнего порогового уровня 8,5 В, но преобразователь работать не начинает, а стабилизатор тока выключается, и конденсатор вновь разряжается до нижнего порогового уровня. Счетчик D1 считает циклы зарядки-разрядки и на восьмом цикле включает преобразователь. Если причина выключения (короткое замыкание, перегрузка) устранена, преобразователь входит в нормальный режим, если нет, то попытки запуска продолжаются.

Режим автозапуска существенно снижает мощность, рассеиваемую микросхемой и преобразователем в целом в аварийных режимах и повышает надежность работы устройства. На рис. 8 приведены временные диаграммы работы в нормальном режиме, на рис. 9 — при перегрузке.

На рис. 8 на первом участке временной диаграммы происходит зарядка конденсатора, подключенного к выводу 1, ключ VT3VT4 закрыт. Далее зарядка конденсатора прекращается, начинает работать преобразователь.

Анализ SFR-совместимости микроконтроллеров фирмы SiLabs

Регистр конфигурации ADC0CF у всех микроконтроллеров находится по одному и тому же SFR адресу 0xBC. Назначение битов этого регистра приведено в табл. 3.

Таблица 3

Название регистра — ADC0CF, ADC Configuration Register (Регистр конфигурации аналого-цифрового преобразователя)									
SFR адрес — 0xBC									
Семейства	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Значение после сброса
F00x, F01x, F02x, F2xx	ADCSC2	ADCSC1	ADCSC0	—	—	AMPGN2	AMPGN1	AMPGN0	0110000b (0x60)
F04x, F12x, F13x	AD0SC4	AD0SC3	AD0SC2	AD0SC1	AD0SC0	AMPGN2	AMPGN1	AMPGN0	11111000b (0xF8)
F06x	AD0SC3	AD0SC2	AD0SC1	AD0SC0	AD0SCAL	AD0GCAL	AD0LCAL	AD0OCAL	11111000b (0xF8)
F30x	AD0SC4	AD0SC3	AD0SC2	AD0SC1	AD0SC0	—	AMPGN1	AMPGN0	11111000b (0xF8)
F31x, F32x, F33x	AD0SC4	AD0SC3	AD0SC2	AD0SC1	AD0SC0	AD0LJST	—	—	11111000b (0xF8)

В первых семействах микроконтроллеров F00x и F01x биты ADCSCx, а в более современных AD0SCx определяют входную частоту для ADC, т. е. косвенно задают время преобразования. Код, устанавливаемый в этих битах, устанавливает коэффициент деления делителя, на вход которого поступает системная тактовая частота.

$$AD0SCx = (SYSCLK/CLK_{ADC0}) - 1.$$

Биты AMPGNx определяют коэффициент усиления входного усилителя:

- 000 — единичное усиление;
- 001 — усиление 2;
- 010 — усиление 4;
- 011 — усиление 8;
- 10X — усиление 16.

В семействе F30x биты AMPGNx также определяют коэффициент усиления входного усилителя, но коды имеют другие значения:

- 00 — усиление 0,5;
- 01 — усиление 1;
- 10 — усиление 2;
- 11 — усиление 4.

AD0(1)SCAL (System Calibration Enable) — бит выбора источника напряжения для калибровки смещения и усиления: 0 — встроенный источник опорного напряжения; 1 — внешний источник.

AD0(1)GCAL (Gain Calibration) определяет состояние калибровки усиления. При чтении: 0 — калибровка завершена или не начиналась, 1 — идет процесс калибровки. При записи: 0 — не действует, 1 — запускает калибровку

усиления, если ADC0(1) находится в режиме энергосбережения.

AD0(1)LCAL (Linearity Calibration) определяет состояние калибровки линейности, аналогично предыдущему биту.

AD0(1)OCAL (Offset Calibration) определяет состояние калибровки смещения, аналогично предыдущим двум битам.

AD0LJST (ADC0 Left Justify Select) — бит выравнивания данных: 0 — вправо; 1 — влево.

Регистр управления ADC0CN у всех микроконтроллеров находится по одному и тому же SFR адресу 0xE8. Назначение битов этого регистра приведено в табл. 4.

Таблица 4

Название регистра — ADC0CN, ADC Control Register (Регистр управления ADC)								
SFR адрес — 0xE8			Значение после сброса — 00000000b (0x00)					
Семейства	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F00x, F01x, F02x, F04x, F12x, F13x, F2xx	ADCEN	ADCTM	ADCINT	ADBUSY	ADSTM1	ADSTM0	ADWINT	ADLIST
F06x	AD0EN	AD0TM	AD0INT	AD0BUSY	AD0CM1	AD0CM0	AD0WINT	—
F30x, F31x, F32x	AD0EN	AD0TM	AD0INT	AD0BUSY	AD0WINT	AD0CM2	AD0CM1	AD0CM0

ADBUSY (AD0(1, 2)BUSY) — бит занятости. При чтении: 0 — преобразование завершено или после сброса не было преобразований. Перепад уровней с высокого в низкий уровень генерирует прерывание, если оно разрешено; 1 — ADC занят преобразованием. При записи: 0 — игнорируется; 1 — запускает преобразование, если ADSTM1, ADSTM0 = 00b.

ADCEN (AD0EN) — бит разрешения работы ADC: 0 — выключен; 1 — включен.

ADCINT (AD0(1,2)INT) — флаг прерывания завершения преобра-

зования, устанавливается в 1 после завершения преобразования.

ADCTM (AD0TM) — бит режима слежения: 0 — слежение выключено до начала преобразования; 1 — режим слежения задается битами ADSTM1, ADSTM0: 00 — слежение начинается с установки ADBUSY и продолжается три периода преобразования; 01 — слежение начинается с переполнения таймера 3 и продолжается три периода преобразования; 10 — слежение производится только тогда, когда внешний вход CNVSTR имеет низкий логический уровень; 11 — слежение

начинается с переполнения таймера 2 и продолжается три периода преобразования.

Бит AD0(1,2)WINT — флаг сравнения оконной функции: 0 — срабатываний не было, 1 — срабатывание оконной функции сравнения произошло.

Биты AD0CM2—AD0CM0 — режимы задания запуска преобразования. Если AD0TM = 0: 000 — преобразование начинается при установке бита ADBUSY; 001 — при перепол-

нении таймера 0; 010 — таймера 2; 011 — таймера 1; 1xx — по перепаду уровня с низкого на высокий на входе CNVSTR. Если AD0TM = 1: 000 — слежение начинается при установке бита AD0BUSY и длится три такта преобразования; 001 — при переполнении таймера 0; 010 — таймера 2; 011 — таймера 1; 1xx — по перепаду уровня с низкого на высокий на входе CNVSTR и также длится три такта преобразования.

Биты ADSTM1, ADSTM0 — режимы задания запуска преобразования: 00 — преобразование начинается при

установке бита ADBUSY; 01 — при переполнении таймера 3; 10 — по перепаду уровней с низкого на высокий; 11 — при переполнении таймера 2.

Бит AD0INT — флаг прерывания завершения преобразования, устанавливается в 1 после завершения преобразования.

ADLIST — бит выравнивания: 0 — регистры ADC0H, ADC0L выравниваются вправо; 1 — влево.

Регистр конфигурации мультимплексора AMX0CF имеется только в некоторых полноформатных семействах (F00x, F01x, F02x, F04x, F12x, F13x). Расположен этот регистр по адресу 0xBA. Назначение битов этого регистра приведено в табл. 5.

Часть битов этого регистра не используется, при чтении из них возвращаются нулевые значения, при записи значения игнорируются. Остальные биты определяют конфигурацию пар входов, например, AIN67IC определяет пару аналоговых входов AIN6 и AIN7. Если бит равен 1 (установлен), входы используются как независимые однополярные. Если бит равен 0 (обнулен), пара выводов работает как дифференциальные входы, причем вход с меньшим номером является неинвертирующим, а вход с большим — инвертирующим. Например, для AIN67IC вход AIN6 — прямой

Таблица 5

Название регистра — AMX0CF, AMUX Configuration Register (Регистр конфигурации аналогового мультимплексора)								
SFR адрес — 0xBA				Значение после сброса — 00000000b (0x00)				
Семейства	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F00x, F01x, F02x, F04x, F12x, F13x	—	—	—	—	AIN67IC	AIN45IC	AIN23IC	AIN01IC

Таблица 6

Название регистра — AMX0SL, AMUX Channel Select Register (Регистр выбора каналов мультимплексора)								
SFR адрес — 0xBB				Значение после сброса — 00000000b (0x00)				
Семейства	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F00x, F01x, F02x, F04x, F06x, F12x, F13x, F2xx, F30x	—	—	—	—	AMXAD3	AMXAD2	AMXAD1	AMXAD0

(неинвертирующий), а вход AIN7 — инверсный (инвертирующий).

Регистр выбора канала мультимплексора AMX0SL имеется в семействах F00x, F01x, F02x, F04x, F06x, F12x, F13x, F2xx и F30x. Расположен этот регистр по адресу 0xBB. Назначение его битов приведено в табл. 6.

Старшие биты не используются, при чтении возвращают 0000b, при записи значение игнорируются. Биты 3:0 представляют собой двоичный адрес от 0000b до 1111b. Коды больше 0111b выбирают температурный датчик. В табл. 7 представлены различные возможные комбинации входов в зависимости от значений битов AMXAD3:0 и AMX0CF3:0.

Малоформатные семейства F31x, F32x и F33x имеют отличный от изложенного выше механизм настройки входов аналогового мультимплексора, она осуществляется не с помощью кодов, а отдельно для каждого положительного (неинвертирующего) и отрицательного (инвертирующего) входов входного усилителя.

Неинвертирующие входы определяются регистром AMX0P с адресом 0xBB (табл. 8). Биты 0...4 регистра AMX0P содержат код выбора положительного входа аналогового мультимплексора в соответствие с табл. 9.

Инвертирующие входы определяются регистром AMX0N с адресом 0xBA (табл. 10). Биты 0...4

International
IOR Rectifier

www.platan.ru
ПЛАТАН

Honeywell
MAXIM
DALLAS
EPCOS
IOR
MITSUBISHI
ELECTRIC
PHILIPS
Infineon
TECHNOLOGIES
ANALOG
DEVICES
VISHAY
AMP
SICK
DATA
VISION
CRYDOM
Kingbright

МИКРОЭЛЕКТРОННЫЕ твердотельные реле

PVN Малое сопротивление в откр. состоянии (100 мОм),
для коммутации больших токов до 4.5 А и малых сигналов

PVT Для телекоммуникационного применения, модели
со встроенной схемой ограничения по току

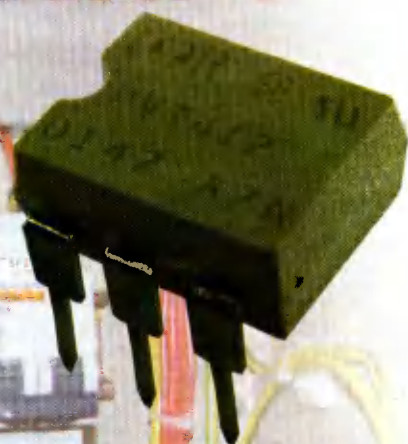
PVY Для низковольтных телекоммуникационных приложений
с высокочастотными сигналами

PVR Переключение аналоговых сигналов до 100 В перем./пост.
тока, полоса до 450 Гц

PVD Переключение аналоговых сигналов до 100 В пост. тока,
полоса до 450 Гц

PVG Выдерживают повышенную нагрузку по току

PVA Переключение аналоговых сигналов до 100 В



Отсутствие дребезгов контактов
Высокое быстродействие,
низкая потребляемая мощность
Выходная развязка: полевой транзистор HEXFET



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

Тел./факс: (095) 744 70 70

Почта: 121351, Москва, а/я 100

E-mail: platan@aha.ru



Украина, Киев
18-21 октября 2005



8-я международная специализированная выставка
электронных компонентов и комплектующих
«Мир электроники 2005»

www.presto.kiev.ua

Титульный спонсор выставки

**ЭЛЕКТРОННЫЕ
КОМПОНЕНТЫ**
украина

Генеральный
информационный спонсор



Выставочный центр "АССО International"
пр-т Победы, 40-б, парк им. А.С. Пушкина

Информационная поддержка:

ИД "Мир электроники" ИД "Электроника" "РАДИОАМАТОР"
"РАДИОХОББИ" "Электроника: НТБ" "РАДИО"
"Компоненты и технологии" Издательство "ЕКОинформ" "ЕСТА"



Оргкомитет выставки – ООО «PrestoExpo»
03062, Украина, г. Киев, ул. Чистяковская, 2, оф. 11
тел/факс: +38 (044) 449-94-76, 443-73-50
e-mail: info@presto.kiev.ua www.presto.kiev.ua

Таблица 7

AMX0CF3:0	AMXAD3:0								1xxx
	0000	0001	0010	0011	0100	0101	0110	0111	
0000	AIN0	AIN1	AIN2	AIN3	AIN4	AIN5	AIN6	AIN7	TEMP SENSOR
0001	+(AIN0) -(AIN1)	—	AIN2	AIN3	AIN4	AIN5	AIN6	AIN7	
0010	AIN0	AIN1	+(AIN2) -(AIN3)	—	AIN4	AIN5	AIN6	AIN7	
0011	+(AIN0) -(AIN1)	—	+(AIN2) -(AIN3)	—	AIN4	AIN5	AIN6	AIN7	
0100	AIN0	AIN1	AIN2	AIN3	+(AIN4) -(AIN5)	—	AIN6	AIN7	
0101	+(AIN0) -(AIN1)	—	AIN2	AIN3	+(AIN4) -(AIN5)	—	AIN6	AIN7	
0110	AIN0	AIN1	+(AIN2) -(AIN3)	—	+(AIN4) -(AIN5)	—	AIN6	AIN7	
0111	+(AIN0) -(AIN1)	—	+(AIN2) -(AIN3)	—	+(AIN4) -(AIN5)	—	AIN6	AIN7	
1000	AIN0	AIN1	AIN2	AIN3	AIN4	AIN5	+(AIN6) -(AIN7)	—	
1001	+(AIN0) -(AIN1)	—	AIN2	AIN3	AIN4	AIN5	+(AIN6) -(AIN7)	—	
1010	AIN0	AIN1	+(AIN2) -(AIN3)	—	AIN4	AIN5	+(AIN6) -(AIN7)	—	
1011	+(AIN0) -(AIN1)	—	+(AIN2) -(AIN3)	—	AIN4	AIN5	+(AIN6) -(AIN7)	—	
1100	AIN0	AIN1	AIN2	AIN3	+(AIN4) -(AIN5)	—	+(AIN6) -(AIN7)	—	
1101	+(AIN0) -(AIN1)	—	AIN2	AIN3	+(AIN4) -(AIN5)	—	+(AIN6) -(AIN7)	—	
1110	AIN0	AIN1	+(AIN2) -(AIN3)	—	+(AIN4) -(AIN5)	—	+(AIN6) -(AIN7)	—	
1111	+(AIN0) -(AIN1)	—	+(AIN2) -(AIN3)	—	+(AIN4) -(AIN5)	—	+(AIN6) -(AIN7)	—	

Таблица 8

Название регистра — AMX0P, AMUX0 Pozitiv Channel Select Register (Регистр конфигурации положительного входа аналогового мультиплексора)								
SFR адрес — 0xBB			Значение после сброса — 0000000b (0x00)					
Семейства	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F31x, F32x, F33x	—	—	—	AMX0P4	AMX0P3	AMX0P2	AMX0P1	AMX0P0

Таблица 10

Название регистра — AMX0N, AMUX0 Negative Channel Select Register (Регистр конфигурации отрицательного входа аналогового мультиплексора)								
SFR адрес — 0xBA				Значение после сброса — 00000000b (0x00)				
Семейства	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
F31x, F32x, F33x	—	—	—	AMX0N4	AMX0N3	AMX0N2	AMX0N1	AMX0N0

регистра AMX0N содержат код выбора отрицательного входа аналогового мультиплексора в соответствии с табл. 11.

Оставшиеся четыре регистра относятся только к одному из семейств. Регистры AMX0PRT и HVA0CN относятся к семейству C8051F04x. Регистры ADC0CCF и ADC0CPTT — к семейству C8051F06x. Очевидно, что

ни о какой совместимости в этом случае речи не идет.

Продолжение следует

ЛИТЕРАТУРА:

1. <http://www.silabs.com>

Олег Николайчук,
г. Кишинев, Молдавия

Таблица 9

Код битов AMX0P4—AMX0P0	Ассоциированная линия ввода/вывода
00000	P1.0
00001	P1.1
00010	P1.2
00011	P1.3
00100	P1.4
00101	P1.5
00110	P1.6
00111	P1.7
01000	P2.0
01001	P2.1
01010	P2.2
01011	P2.3
01100	P2.4
01101	P2.5
01110	P2.6
01111	P2.7
10000	P3.0
10001*	P3.1*
10010*	P3.2*
10011*	P3.3*
10100*	P3.4*
10101...11101	Не используются
11110	Температурный датчик
11111	Напряжение питания U _{DD}

Таблица 11

Код битов AMX0N4—AMX0N0	Ассоциированная линия ввода/вывода
00000	P1.0
00001	P1.1
00010	P1.2
00011	P1.3
00100	P1.4
00101	P1.5
00110	P1.6
00111	P1.7
01000	P2.0
01001	P2.1
01010	P2.2
01011	P2.3
01100	P2.4
01101	P2.5
01110	P2.6
01111	P2.7
10000	P3.0
10001*	P3.1*
10010*	P3.2*
10011*	P3.3*
10100*	P3.4*
10101...11101	Не используются
11110	Опорное напряжение U _{REF}
11111	Общий в однополярном режиме

НОВОСТИ КОМПАНИИ «ЧИПЭКСПО»

Продолжается прием заявок на участие в тематических стендах «Инновационные и инвестиционные проекты российской электроники» и «Импортозамещение и аналоги», которые будут организованы на третьей международной выставке «ChipEXPO-2005».

Приглашаются все российские компании, обладающие инновационными, инженерными и научными разработками в микроэлектронике, электронике и специальных технологиях для представления своей продукции на стендах.

Продолжается прием заявок в сборник «Дайджест инновационных разработок российской электроники». Все российские компании, независимо от того, принимают ли они участие в выставке «ChipEXPO-2005» или нет, могут поместить аннотацию своих изделий, разработок, проектов, идей в дайджесте,

который будет распространяться одновременно с официальным каталогом выставки.

Все участники тематических стендов автоматически участвуют в конкурсе на получение отраслевой премии «Золотой Чип» в номинации «За лучшую инновационную идею» и «За достижения в разработке импортозамещающей продукции и аналогов».

Оформить заявку на участие в составе тематических стендов и в конкурсе «Золотой чип» и разместить аннотацию для включения ее в дайджест «Инновационные разработки российской электроники» можно на сайте www.chipexpo.ru в разделе «Проекты».

Цифровые сигнальные процессоры

В этой статье рассказывается о вычислительных устройствах сигнального процессора и выполняемых с их помощью командах.

ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА

Как уже говорилось ранее, процессор ADSP-2181 содержит три независимых вычислительных устройства (рис. 1).

К ним относятся арифметико-логическое устройство (ALU), умножитель с накопителем (MAC) и устройство сдвига (Shifter). Эти устройства работают с 16-разрядными данными и имеют аппаратную поддержку обмена данными, позволяющую обеспечить многопоточность.

ALU выполняет ряд стандартных арифметических и логических команд, MAC — одноцикловые операции умножения, умножения-сложения и умножения-вычитания. Shifter осуществляет логические и арифметические сдвиги, нормализацию, денормализацию и операцию получения порядка, а также управление форматом данных, разрешая работу с плавающей точкой.

Вычислительные модули размещаются последовательно друг за другом таким образом, чтобы выход одного мог стать входом другого в следующем цикле. Результаты работы вычислительных устройств собираются на 16-разрядную шину R-BUS.

Все три устройства содержат входные и выходные регистры, которые доступны через 16-разрядную шину данных. Команды, выполняемые в них, берут в качестве операндов данные, находящиеся в регистрах ввода, и после выполне-

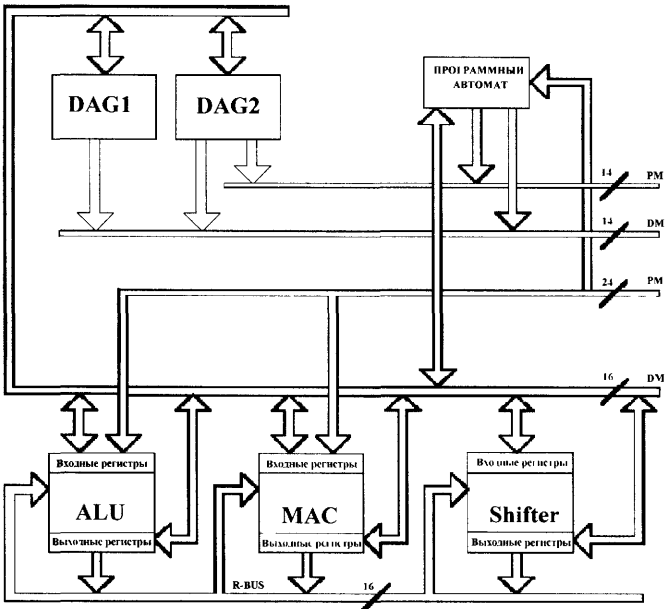


Рис. 1
Таблица 1

Сигнал	Расшифровка	Назначение
AZ	Zero (нуль)	Разряд нулевого результата. Устанавливается в 1, если результат операции равен 0. Является результатом логического сложения всех битов регистра результатов ALU.
AN	Negativ (негатив)	Знаковый разряд результата ALU. Устанавливается в 1, если результат отрицательный.
AV	Overflow (переполнение)	Разряд переполнения. Устанавливается в 1, если происходит переполнение ALU путем переноса единицы в знаковый разряд при сложении чисел.
AC	Carry (перенос)	Разряд переноса. Устанавливается в 1 при сложении беззнаковых чисел, если происходит переполнение ALU путем переноса единицы из старшего разряда.
AS	Sign (знак)	Знаковый разряд. Определяется при операции вычисления абсолютного значения числа ABS. Устанавливается в 1, если операнд имеет отрицательное значение.
AQ	Quotient (частное)	Разряд состояния частного. Формируется только командами DIVS и DIVQ.

ния операции записывают результат в регистры вывода. Регистры являются как бы промежуточным хранилищем между памятью и вычислительным устройством. Шина R-BUS позволяет результату одного вычисления стать операндом для другой операции. Это экономит время благодаря исключению лишних пересылок данных между устройствами и памятью.

АРИФМЕТИКО-ЛОГИЧЕСКОЕ УСТРОЙСТВО ALU

Рассмотрим подробнее ALU. Его структурная схема приведена на рис. 2.

ALU является 16-разрядным устройством с двумя 16-разрядными входными портами данных X и Y и одним выходным портом результата R. ALU генерирует шесть сигналов состояния: нуль (AZ), негатив (AN), перенос (AC), переполнение (AV), знак (AS) и частное (AQ). В конце каждого цикла процессора эти сигналы сохраняются в

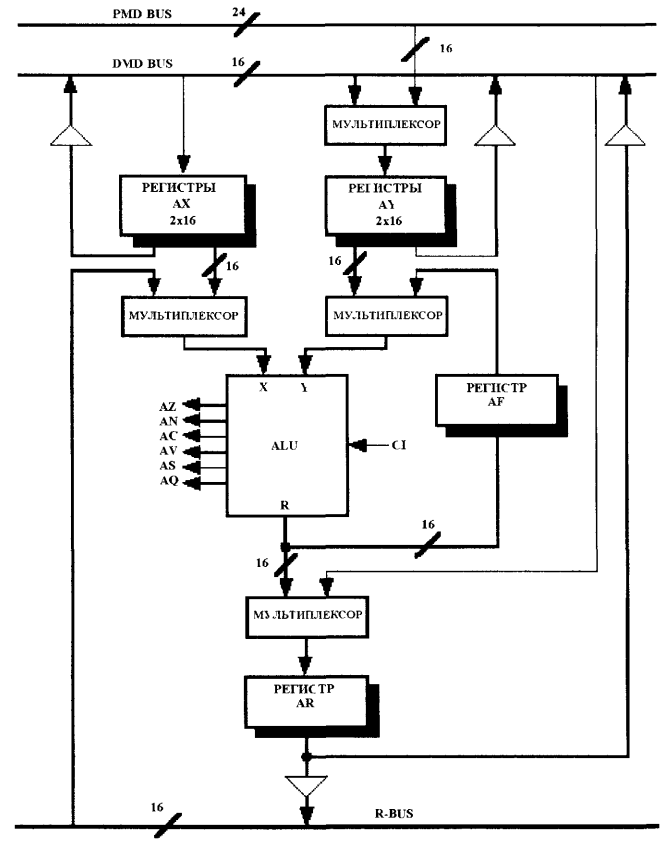


Рис. 2

регистре состояния процессора ASTAT. Описание назначения этих сигналов состояния приведено в табл. 1.

Порт ввода X может принимать данные от двух источников — из блока регистров AX или с шины результата R-BUS. Шина R-BUS обеспечивает обмен данными между всеми вычислительными устройствами. Блок регистров AX состоит из двух регистров: AX0 и AX1. Эти регистры подключены к шине памяти данных DMD-BUS. Система команд позволяет читать содержимое регистра AX и выдавать данные на шину памяти программ PMD-BUS, для этого используется устройство обмена между шинами. Выводы регистров AX0 и AX1 реализованы таким образом, чтобы один из них служил операндом для ALU, а другой в это время мог считываться на шину DMD-BUS.

Порт ввода Y также принимает данные от двух источников — из блока регистров AY и регистра обратной связи AF. Блок регистров AY состоит из регистров AY0 и AY1. Эти регистры доступны для обмена данными с шиной DMD-BUS. Система команд процессора позволяет читать регистры AY0 и AY1 с помощью шины PMD-BUS с использованием устройства обмена между шинами. Выводы регистров AY0 и AY1 реализованы аналогично регистрам AX0 и AX1.

Выход ALU подключен к выходному регистру результата AR и через регистр обратной связи AF — на вход ALU через мультиплексор. Регистр AF позволяет использовать результат вычисления в качестве следующего операнда. Регистр AR имеет выход на шины DMD-BUS и R-BUS. В системе команд предусмотрена возможность чтения регистра AR посредством шины PMD-BUS аналогично двум предыдущим вариантам.

Любой из регистров ALU доступен для записи и чтения в течение цикла процессора (для чтения — в начале цикла, для записи — в конце). Таким образом, новое значение, записанное в конце одного цикла, может быть прочитано лишь в начале следующего. Это позволяет входным регистрам записать в ALU операнд в начале цикла и считать следующий операнд в конце того же цикла. Это также позволяет сохранить содержимое регистра результата в памяти и оперировать со следующим результатом в одном цикле.

ALU имеет альтернативные (теневые) банки регистров AX, AY, AF и AR. На структурной схеме они изображены в виде теней. В конкретный момент времени доступен только один из типов банков. Дополнительный банк может быть использован для автоматического сохранения данных в регистрах ALU при быстром переключении на выполнение подпрограммы. В этом случае экономится время на сохранение текущих данных в памяти процессора.

Выбор главного или альтернативного банка регистров определяется битом 0 регистра режимов процессора MSTAT. Если этот разряд содержит 0, выбран главный банк, если 1, выбирается дополнительный.

ALU выполняет набор стандартных арифметических и логических операций. Из арифметических операций АЛУ выполняет сложение, вычитание, отрицание, инкремент, декремент и вычисление модуля. Эти операции дополнены двумя примитивами, с помощью которых возможна реализация цикла многократного деления. Из логических операций АЛУ выполняет логическое сложение (OR), логическое умножение (AND), «Исключающее ИЛИ» (XOR), логическое отрицание (NOT). Стандартные операции ALU приведены в табл. 2.

В качестве условных записей операндов X, Y и R ALU, указанных в табл. 2, могут выступать следующие регистры:

Таблица 2

Операция	Назначение
R = X+Y	Сложение операндов X и Y
R = X+Y+CI	Сложение операндов X и Y с переносом
R = X-Y	Вычитание операнда Y из X
R = Y-X	Вычитание операнда X из Y
R = X-Y+CI-1	Вычитание операнда Y из X с заемом
R = Y-X+CI-1	Вычитание операнда X из Y с заемом
R = -X	Инверсия операнда X
R = -Y	Инверсия операнда Y
R = X+1	Инкремент операнда X
R = Y+1	Инкремент операнда Y
R = X-1	Декремент операнда X
R = Y-1	Декремент операнда Y
R = PASS X	Результат равен операнду X с установкой флагов регистра статуса ASTAT
R = PASS Y	Результат равен операнду Y с установкой флагов регистра статуса ASTAT
R = 0	Очистка результата
R = ABS X	Результат равен абсолютному значению операнда X (значение по модулю)
R = X AND Y	Логическое умножение операндов X и Y
R = X OR Y	Логическое сложение операндов X и Y
R = X XOR Y	«Исключающее ИЛИ» операндов X и Y
R = NOT X	Логическое отрицание X
R = NOT Y	Логическое отрицание Y

- для X — AX0, AX1, AR, MR0, MR1, MR2, SR0, SR1;
- для Y — AY0, AY1, AF;
- для R — AR и AF.

Регистры MR0, MR1, MR2 принадлежат умножителю MAC, регистры SR0, SR1 — устройству сдвига Shifter. Они доступны ALU по шине данных.

Для обработки чисел с повышенной точностью в ALU используются сигнал переноса CI (carry-in) и бит переноса AC (ALU carry). Операция сложения с переносом предназначена для сложения старших частей чисел, а операция вычитания с заемом — для их вычитания.

Регистр AR может работать в режиме насыщения. В этом режиме он принимает максимально возможное отрицательное или положительное значение, если операция заканчивается переполнением. Эта функция регистра AR становится возможной при установке в единицу разряда 3 регистра MSTAT. В табл. 3 показано содержимое регистра AR ALU в зависимости от флагов регистра состояния ASTAT при включенном режиме насыщения.

Режим блокировки переполнения ALU разрешается при установке в единицу бита 2 регистра MSTAT. В этом режиме при переполнении ALU разряд AV устанавлива-

Таблица 3

AV (переполнение)	AC (перенос)	Содержимое AR
0	0	В соответствии с результатом ALU
0	1	В соответствии с результатом ALU
1	0	0111111111111111 (макс. положительное число)
1	1	1000000000000000 (мин. отрицательное число)

ется в 1, и при последующих операциях остается в этом состоянии. Таким образом, последующие переполнения уже не генерируются. Бит состояния AV можно очистить непосредственно с шины DMD путем записи в регистр состояния ASTAT нуля.

Продолжение следует

Олег Вальпа,
г. Миасс Челябинской обл.

Микроконтроллеры? Это же просто!

Итак, поскольку частное в отличие от произведения можно получать только со старших его цифр, имеют-ся всего две вычислительные схемы деления (рис. 44). Схема 1 (рис. 44, а) аналогична схеме 3 умножения (см. рис. 33), а схема 2 (рис. 44, б) — схеме 4 умножения (см. рис. 34). Очевидно, что схема 1 — это схема, в которой делимое по мере вычисления сдвигается влево (рис. 45), а делитель остается неподвижным. Схема 2 — это схема, в которой делимое неподвижно, а делитель по мере вычисления сдвигается вправо (рис. 46).

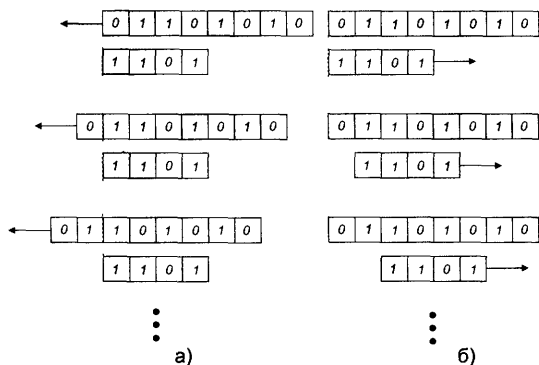


Рис. 44

В схемах на рис. 45 и 46 использованы следующие обозначения: ДМ — делимое; ДЛ — делитель; ЧСТ — частное; ОСТ — остаток (в скобках указаны разрядности данных). Цифры частного формируются по значению признака переноса СУ, устанавливаемого в операциях вычитания делителя из делимого или остатка, и сдвигаются влево. При этом в схеме 1 для получения нового остатка предыдущий остаток сдвигается влево при неподвижном делителе, а в схеме 2 наоборот, при неподвижном остатке делитель сдвигается вправо. Чаще всего схема 1 оказывается проще, поскольку не требует команд сдвига вправо многобайтных данных (в большинстве микроконтроллеров сдвиг вправо сложнее сдвига влево, последний, как мы уже видели, легко реализуется многобайтовым сложением регистров с самими собой). Поэтому все рассматриваемые ниже программы деления выполнены по схеме 1.

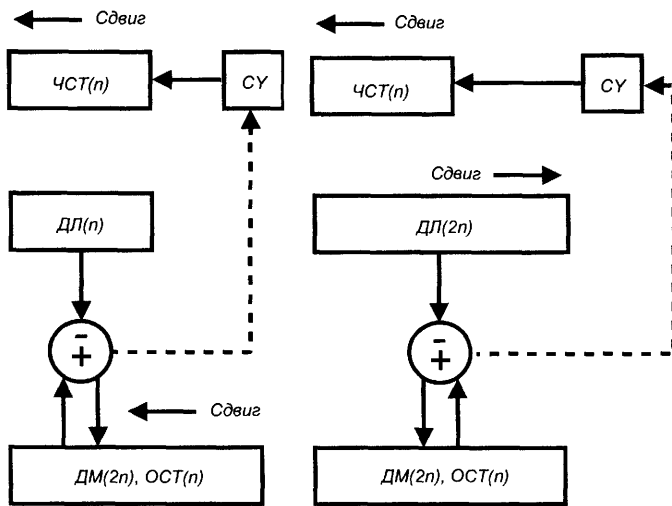


Рис. 45

Рис. 46

В схеме 1 есть два способа получения цифр частного и остатка: деление с восстановлением остатка и деление без восстановления остатка. С первым из них мы уже практически познакомились выше и здесь опишем его формально. Итак, рассмотрим способ деления с восстановлением остатка для класса целых беззнаковых двоичных чисел (он применим и для дробных беззнаковых чисел). Пусть X, Y, Z, Q — целые беззнаковые (X, Y, Q — n -разрядные числа, а Z — $2n$ -разрядные) числа, такие что $X \cdot Y + Q = Z$, где $Q < X$. Тогда справедливы выражения:

$$Z : Y \approx X; \quad Z : Y = (X, Q), \quad (18)$$

где X — неполное частное, Q — остаток. Если $Z < Y$, то $X = 0, Q = Z$; если $Z = Y$, то $X = 1, Q = 0$; если $Z = X \cdot Y$, то $Q = 0$.

Обозначим буквой Q разность между остатком Q_{i-1} и делителем, где $i \in \{1, 2, \dots, n\}$; $Q_0 = Z$, а через x_j — цифру частного X , причем $j = n - i, j \in \{0, 1, \dots, n - 1\}$. Тогда процедуру деления с восстановлением остатка можно записать в виде рекуррентной формулы

$$Q_i, x_{n-i} = \begin{cases} 2Q_{i-1} - Y; & x_{n-i} = 1 \text{ при } 2Q_{i-1} - Y \geq 0; \\ 2Q_{i-1}; & x_{n-i} = 0 \text{ при } 2Q_{i-1} - Y < 0; \end{cases} \quad (19)$$

Эта процедура состоит из n циклов деления, причем в каждом i -м цикле осуществляется сдвиг влево на один двоичный разряд остатка Q_{i-1} , полученного во время выполнения предыдущего $(i-1)$ -го цикла, и вычисление разности между удвоенным остатком и делителем $2Q_{i-1} - Y$. Если разность положительна, то в очередной $(n-i)$ -й разряд частного записывается цифра $x_{n-i} = 1$, а разность становится новым остатком Q_i . Если же разность отрицательна, то в разряд частного записывается $x_{n-i} = 0$, а остаток $2Q_{i-1}$ восстанавливается либо путем сложения $2Q_{i-1} - Y + Y$, либо извлечением из области его сохранения и интерпретируется как новый остаток Q_i . После выполнения n -го цикла деления сформированы все n цифр целой части частного. С целью вычисления дробной части частного и повышения тем самым точности его вычисления мы можем многократно повторять последнюю операцию, вычисляя частное с любой наперед заданной точностью.

Стоит заметить, что в алгоритме (19) вначале производится сдвиг частного влево, а затем возможное вычитание. Это немного отличается от предыдущего описания алгоритма, но удобнее, т. к. такой сдвиг освобождает в регистрах место для следующего бита частного. В свою очередь при этом несколько усложняется вычисление разности и определение разряда частного. Но принципиально замена местами сдвига частного и возможного вычитания ничего не меняет — и в том, и в другом случае деление выполняется корректно.

Еще одно замечание. При выполнении деления $2n$ -разрядного делимого на n -разрядный делитель возможно переполнение n -разрядного формата частного: появление абсолютной величины, не представимой в n -разрядном формате. Это имеет место в том случае, когда число в старших n разрядах делимого больше или равно делителю.

Переполнение необходимо выявлять программным способом и блокировать дальнейшие вычисления вследствие их некорректности. Избежать переполнения можно за счет увеличения разрядности частного до разрядности делимого, но при этом в два раза увеличивается и число циклов деления. Помимо выявления переполнения при делении чисел необходимо определять ситуацию деления на нуль. Заметим, что при анализе переполнения при делении целых чисел можно автоматически выявлять и деление на нуль, как это делается в наших программах.

Теперь давайте посмотрим, как способ деления с восстановлением остатка и с анализом возможности переполнения частного реализован в подпрограмме DIV16.

```
DIV16:
; Подпрограмма деления двух целых беззнаковых чисел
; Вход:
; R7R6R5R4 - 32-разрядное делимое
; R3R2 - 16-разрядный делитель
; Выход:
; R5R4 - 16-разрядное частное от деления (если CY=0)
; R7R6 - 16-разрядный остаток от деления (если CY=0)
; Флаги.
; CY - признак переполнения 16-разрядного частного
; или деление на нуль, т.е. CY=1 - признак ошибки и
; результат в R7R6 и R5R4 не определен
; Используемые регистры:
; A, B
; Требует свободных байт в стеке:
; 2

CLR C
MOV A, R6
SUBB A, R2
MOV A, R7
SUBB A, R3
JC DIV16_0 ; R7R6 < R3R2, значит, частное
; умещается в 16-ти битах

SETB C ; CY=1 - переполнение 16-разрядного
частного или
RET ; деление на нуль
```

```
DIV16_0:
MOV B, #16 ; 16 циклов

DIV16_1:
MOV A, R4
ADD A, R4
MOV R4, A
MOV A, R5
ADDC A, R5
MOV R5, A
MOV A, R6
ADDC A, R6
MOV R6, A
MOV A, R7
ADDC A, R7
MOV R7, A ; R7R6R5R4 сдвинуто влево

JC DIV16_2 ; если переполнение остатка,
; вычесь ДЛ, +1 в частное
```

```
MOV A, R6
SUBB A, R2
MOV R6, A
MOV A, R7
SUBB A, R3
MOV R7, A ; вычли делитель из остатка

JNC DIV16_3 ; +1 в частное, восстанавливать ос-
таток не нужно

MOV A, R6
ADD A, R2
MOV R6, A
MOV A, R7
ADDC A, R3
MOV R7, A ; восстановили остаток

SJMP DIV16_4 ; +0 в частное

DIV16_2:
CLR C
MOV A, R6
SUBB A, R2
MOV R6, A
MOV A, R7
SUBB A, R3
MOV R7, A ; вычли делитель из остатка

DIV16_3:
INC R4 ; добавили 1 в частное

DIV16_4:
DJNZ B, DIV16_1 ; цикл

CLR C ; CY=0 - нет переполнения 16-разрядно-
го частного
RET ; и нет деления на нуль
```

Рис. 47

Вначале осуществляется проверка на переполнение частного. Переполнение возникнет, если старшие два байта делимого будут больше или равны делителю. Поэтому до того, как начать деление, мы сравниваем их. Уточнее, вычитаем из старших двух байт делимого делитель) и переходим непосредственно к операции деления только в случае, если CY = 1. Если это не так, то мы выходим из подпрограммы со знаком “аварийной ситуации” (установленным в 1 флаге переноса), и основная программа, вызвавшая подпрограмму DIV16, должна будет каким-либо образом информировать вас о том, что деление осуществлено некорректно.

Далее все достаточно тривиально. В регистре В организован счетчик циклов — их должно быть 16, коль скоро 32-разрядное число делится на 16-разрядное. Сдвиг влево осуществляется путем сложения содержимого регистров R7R6R5R4 с самими собой. В тем, кто забыл, напомним, что сдвиг влево на один разряд эквивалентен удвоению двоичного числа). Если при вычитании делителя из соответствующего остатка перенос не возник, в соответствующий разряд частного заносится 1, если возник, то заносится 0 и остаток восстанавливается. В ходе выполнения подпрограммы делимое на каждом шаге по биту

освобождает регистры R4R5 (начиная с младшего бита R4), и в этих регистрах накапливается частное. Выглядит это так, как будто частное по мере его определения вытесняет делимое из R4R5.

Способ деления без восстановления остатка определяется выражениями:

$$Q_i = \begin{cases} 2Q_{i-1} - Y, & \text{если } Q_{i-1} \geq 0; \\ 2Q_{i-1} + Y, & \text{если } Q_{i-1} < 0; \end{cases}$$
$$x_{n-1} = \begin{cases} 1, & \text{если } Q_i \geq 0; \\ 0, & \text{если } Q_i < 0; \end{cases} \quad (23)$$

В этой процедуре при получении отрицательной разности $Q_{i-1} = 2Q_{i-2} - Y$ последняя не восстанавливается, как в приведенной выше рекуррентной формуле для деления с восстановлением остатка $Q_{i-1} = 2Q_{i-2}$, а сдвигается в следующем i -ом, цикле влево и к ней прибавляется делитель $Q_i = 2Q_{i-1} + Y$.

Почему это возможно? В схеме с восстановлением остатка при получении в $(i - 1)$ -ом цикле отрицательной разности $Q_{i-1} = 2Q_{i-2} - Y$ мы, во-первых, прибавляли к этой разности делитель Y (восстанавливали остаток, имея в итоге $Q_{i-1} = 2Q_{i-2}$), во-вторых сдвигали влево Q_{i-1} , получая $Q_i = 2Q_{i-1}$, и затем уже в i -ом цикле вычитали этот делитель Y из Q_i , получая $Q_i = 2Q_{i-1} - Y = 2(2Q_{i-2}) - Y$. Фактически мы прибавили к отрицательной разности Q_{i-1} , сдвинули его влево вместе с ней (удвоили) и затем в i -ом цикле вновь вычли. А если к какой-либо величине прибавить константу $2Y$, а затем вычесть Y , это будет эквивалентно тому, чтобы просто один раз прибавить к ней Y . Иными словами, вместо сложения Y с $2Q_{i-2}$, удвоения этой суммы и последующего вычитания из нее Y можно вначале удвоить $2Q_{i-2}$ и прибавить к удвоенному Y . А ведь это и есть алгоритм деления без восстановления остатка, и он, как мы только что убедились, идентичен алгоритму деления с восстановлением остатка. Нижеследующее преобразование еще раз подтверждает это:

$$Q_i = 2Q_{i-1} + Y = 2(2Q_{i-2} - Y) + Y = 2(2Q_{i-2}) - Y. \quad (24)$$

При способе деления с восстановлением остатка в каждом цикле производятся либо одна, либо две операции: вычитание или вычитание и сложение (при восстановлении). При втором способе проводится только одна операция — вычитание или сложение. Поэтому второй способ имеет потенциально большее быстродействие, но это его преимущество проявляется лишь при аппаратных реализациях деления. Программная реализация второго способа сложнее, что связано с необходимостью хранения знака остатка до начала следующего цикла деления с целью определения типа выполняемой операции (сложение или вычитание делителя) и коррекции конечного остатка Q_n в случае, если $Q_{n-1} < 0$ (необходимо вычесть делитель). Вследствие этого деление без восстановления остатка редко используется в программах.

Последовательность действий при делении с восстановлением и без восстановления остатка поясняется на примере деления двоичных чисел $00100000:1111 = (0010, 0010)$ соответственно в табл. 14 и 15, в которых приняты следующие обозначения операций: $\leftarrow Q_i$ — сдвиг влево остатка; $\pm Y$ — суммирование или вычитание делителя; $= Q_i$ — присвоение значения остатку. Заметим, что остаток считается положительным при делении с восстановлением остатка, если в результате вычитания $2Q_{i-1} - Y$ признак переноса $CY = 0$, а при делении без восстановления остатка,

Таблица 14

I	Остаток		x_{4-1}	Тип операции
	CY	Q_i		
1	0	01000000		$\leftarrow Q_0$
		1111		$-Y$
	1	01010000		$=Q_1 < 0$
		1111		$+Y$
	1	01000000	0	$=Q_1 > 0$
2	0	10000000		$\leftarrow Q_1$
		1111		$-Y$
	1	10010000		$=Q_2 < 0$
		1111		$+Y$
	1	10000000	0	$=Q_2 > 0$
3	1	00000000		$\leftarrow Q_2$
		1111		$-Y$
		00010000	1	$=Q_3 > 0$
4	0	00100000		$\leftarrow Q_3$
		1111		$-Y$
	1	00110000		$=Q_4 < 0$
		1111		$+Y$
	1	00100000	0	$=Q_4 > 0$

Таблица 15

I	Остаток		x_{4-1}	Тип операции
	CY	Q_i		
1	0	01000000		$\leftarrow Q_0$
		1111		$-Y$
	1	01010000	0	$=Q_1 < 0$
2	0	10100000		$\leftarrow Q_1$
		1111		$+Y$
	1	10010000	0	$=Q_2 < 0$
3	1	00100000		$\leftarrow Q_2$
		1111		$+Y$
		00010000	1	$=Q_3 > 0$
4	0	00100000		$\leftarrow Q_3$
		1111		$-Y$
	1	00110000	0	$=Q_4 < 0$
		1111		$+Y$
	1	00100000		$=Q_4 > 0$

если признак переноса CY не меняет своего значения между операциями $2Q_{i-1}$ и $2Q_{i-1} \pm Y$, т. е. $CY:1 \rightarrow 1$ или $CY:0 \rightarrow 0$.

Если при выполнении деления по схеме без восстановления остатка в последнем цикле Q_i окажется отрицательным, его нужно восстановить так же, как это делалось в завершающем цикле схемы с восстановлением остатка (сравните последние пять строк обеих таблиц). Это нужно для того, чтобы получить в результате деления неискаженный целочисленный остаток (если делимое не делится нацело).

Как уже упоминалось, при выполнении деления $2n$ -разрядного делимого на n -разрядный делитель возможно переполнение n -разрядного частного. Фактически это означает, что получаемое частное нельзя представить в n -разрядном формате и ему требуется большее число разрядов. Например, при попытке разделить 11110000_2 на 1011_2 вы получите 10101_2 с остатком 1001_2 ($240 : 11 = 21$ и 9 в остатке). Это переполнение имеет место в том случае, когда число в старших n разрядах делимого больше или равно делителю ($1111_2 > 1011_2$).

Как и раньше, проверить работоспособность программ полезно на тестовых наборах данных (табл. 16) или на аналогичных, составленных с использованием стандартного Windows-калькулятора.

Таблица 16

Представление чисел	
Шестнадцатеричное	Десятичное
FFFE0001:FFFF=(FFFF,0000)	4294836225:65535=(65535,0)
FFFE0002:FFFF=(FFFF,0001)	4294836226:65535=(65535,1)
00FE0100:FF00=(00FF,0)	16646400:65280=(255,0)
FFFDFFFF:FFFF=(FFFE,FFFD)	4294836223:65535=(65534,65533)
38E31C75:AAAA=(5555,0003)	954408053:43690=(21845,3)
3FFF0000:7FFF=(7FFE,7FFE)	1073676288:32767=(32766,32766)

Продолжение следует

Александр Фрунзе,
Алексей Фрунзе,
г. Москва

Уменьшение пульсаций в блоке питания с емкостным балластом

Теперь о гипотетической возможности дребезга включения, при котором пиковый ток теоретически может удвоиться [2]. На с. 2 [7] приводится график допустимого пикового тока полупериода синусоиды частотой 60 Гц от числа циклов (полупериодов).

Экстраполяция этого графика в сторону долей цикла дает основание полагать, что при длительности импульса, не превышающей 100 мкс, для диодов серий 1N400х, RL10х, RL20х допустимый пиковый ток $I_{\text{пик доп}}$ как минимум в два раза превышает ток I_{FSM} :

$$I_{\text{пик доп}} \geq 2I_{\text{FSM}} \quad (6)$$

Проведенный в первой части статьи анализ был сделан в предположении, что сеть является цепью с сосредоточенными параметрами и имеет нулевое выходное сопротивление. Однако в действительности все обстоит гораздо сложнее. Выходное

сопротивление сети на частотах превышающих 30...50 кГц. Выше мы полагали, что зарядный ток балластного конденсатора увеличивается скачком (мгновенно). Однако известно, что природа не терпит скачков. По отношению к импульсным сигналам сеть следует рассматривать как длинную линию с распределенными параметрами (конечными величинами погонных индуктивностей, емкостей и активных сопротивлений) [8, 9]. При коротком замыкании «хорошая» сеть может «отдать» очень большой ток, но скорость нарастания этого тока ограничена параметрами длинной линии и в результате балластный конденсатор заряжается быстрее, чем его зарядный ток нарастает до катастрофических величин. Поясним сказанное примером. На рис. 10 изображена схема, в которой длинная линия с потерями Т1 моделирует достаточно мощный сетевой ввод, например, участок линии длиной 250 м

минальном напряжении сети через линии Т2 и Т3 протекают переменные токи соответственно 50 и 2,2 А (действующие значения). Ключ А1 моделирует подключение к сети зарядной цепи R, VD1—VD4 балластного конденсатора С. Параметры длинных линий Т1—Т3 приведены в табл. 3.

На рис. 11 изображены временные диаграммы импульса зарядного тока балластного конденсатора, протекающего через диоды VD3 и VD2, для четырех значений сопротивления ограничительного резистора R.

В табл. 4 приведены максимальные значения импульсов зарядного тока I_{MAX} и времена спада этих импульсов до уровня 2 А, полученные в результате многовариантного анализа схемы на рис. 10 при изменении емкости балластного конденсатора С и сопротивления ограничительного резистора R.

«Зубцы» на временных диаграммах тока обусловлены отражениями волн тока и напряжения от концов линий, нагруженных на несогласованные сопротивления. Обратите внимание на то обстоятельство, что при прочих равных условиях и увеличении емкости балластного конденсатора возрастают амплитуда I_{MAX} и длительность t_z импульса зарядного тока.

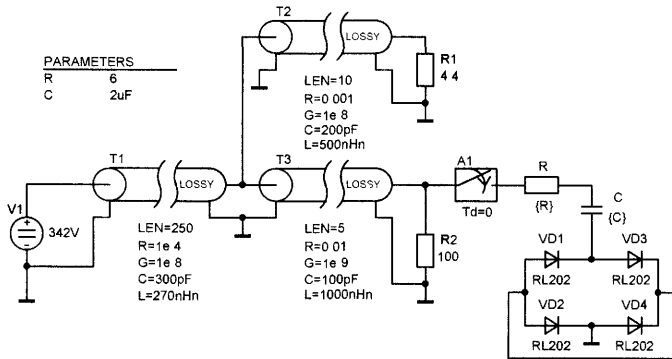


Рис. 10

сопротивление сети на низких частотах (на частоте 50 Гц) определяется следующими факторами:

- длиной линии (удаленностью потребителя от подстанции), сечением провода линии, материалом (удельным сопротивлением) провода, температурой провода;
- сопротивлением контактных соединений в электрических шкафах, соединительных коробках, розетках, защитной аппаратуре и т. п.

Однако главным фактором, существенно уменьшающим пиковый зарядный ток балластного конденсатора, является достаточно высокое выход-

от понижающей подстанции до жилого многоквартирного дома, а линия Т3 — ответвление, например, квартирный сетевой ввод (участок линии длиной 5 м от электрощита в подъезде до розетки в квартире). Линия Т2 и резистор R1 моделируют другие потребители энергии, питающиеся от линии Т1.

Резистор R2 — это внутриквартирные потребители энергии. При но-

Таблица 4

R, Ом	C = 0,47 мкФ		C = 1 мкФ		C = 2 мкФ	
	I_{MAX} , А	t_z , мкс	I_{MAX} , А	t_z , мкс	I_{MAX} , А	t_z , мкс
0	44,1	7,92	54,3	13,2	68,9	23,0
6	25,6	15,3	28,6	29,0	33,0	48,5
12	17,9	19,8	19,2	38,3	21,9	67,9
24	11,0	24,9	11,5	50,3	13,0	96,7

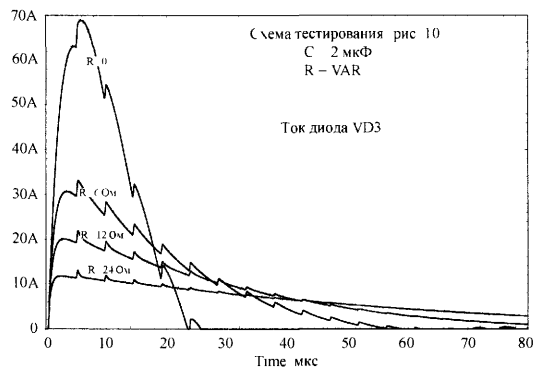


Рис. 11

Таблица 3

Параметр	T1	T2	T3
Длина, м	250	10	5
Погонное сопротивление, Ом/м	0,0001	0,001	0,01
Погонная проводимость утечки, Сим/м	10 ⁻⁸	10 ⁻⁸	10 ⁻⁹
Погонная индуктивность, нГн/м	270	500	1000
Погонная емкость, пФ/м	300	200	100
Волновое сопротивление, Ом	30	50	100

Приведенные результаты наглядно показывают, что сеть является достаточно эффективным демпфером быстрых изменений тока нагрузки: даже при отсутствии ограничительного резистора ($R = 0$) и вариации емкости балластного конденсатора в пределах 0,47...2 мкФ пиковые значения зарядных токов не превышают 44,1...68,9 А. Важно, однако, понимать, что схема, изображенная на рис. 10, является очень грубым приближением к реальной ситуации. Для реальной низковольтной линии передачи электрической энергии характерно следующее:

- к линии подключено большое число потребителей, имеющих различный характер (индуктивный, активный, реже емкостной) нагрузки, более того, конфигурация сети постоянно изменяется, поскольку к ней подключаются одни потребители и отключаются другие;
- различные участки линии обычно выполняются из проводов разных марок и, следовательно, эти участки характеризуются различными величинами распределенных параметров и не согласованы между собой по волновому сопротивлению;
- в разрывы линий включаются защитные устройства и счетчики потребляемой энергии, внутренние сопротивления которых имеют ярко выраженный индуктивный характер и на высоких частотах по величине могут быть соизмеримы с сопротивлением ограничительного резистора БП;
- на высоких частотах (при большой скорости изменения тока в линии) за счет скин-эффекта [10, 11] увеличивается омическое сопротивление линии;
- на параметры участков линии оказывают влияние близко расположенные к ним предметы с достаточно высокой электропроводностью и/или магнитной проницаемостью.

Из сказанного следует, что свойства линий электропередачи на высокой частоте, влияющие на процесс зарядки балластного конденсатора, зависят от большого числа факторов. Кроме того, в реальной схеме БП (см. рис. 8) уменьшению пикового тока через диоды способствуют следующие факторы:

- конечная величина эквивалентного последовательного сопротивления (ESR) на высокой частоте оксидного накопительного

конденсатора СЗ, включенного последовательно с балластным конденсатором и диодами БП;

- паразитные индуктивности и активные сопротивления элементов БП (так, например, сопротивление предохранителя ВП-1-1-0,5А находится в диапазоне значений 0,3...0,4 Ом);
- остаточная индуктивность насыщенного дросселя ДПМ (насыщенный дроссель имеет индуктивность не меньшую, чем воздушный дроссель тех же габаритов и с тем же числом витков);
- подключение БП к сети в момент времени, когда ее напряжение превышает (20...30) В, неизбежно сопровождается образованием между замыкающимися контактами электрической дуги [12], которая имеет конечное сопротивление и, следовательно, также демпфирует процесс зарядки балластного конденсатора.

Количественно оценить степень влияния большинства из перечисленных выше факторов на пиковый ток диодов БП не представляется возможным. Не будем, однако, чрезмерно драматизировать ситуацию, поскольку, к счастью, результатом влияния этих факторов является уменьшение пикового тока диодов.

Подводя итог сказанному и принимая во внимание то обстоятельство, что теоретически за счет дребезга контактов пиковый ток диодов может удвоиться [2], сформулируем рекомендации, следование которым обеспечивает высокую надежность работы БП с емкостным балластным:

- сопротивление R резистора, ограничивающего зарядный ток балластного конденсатора в момент подключения БП к сети, должно отвечать следующему требованию:

$$R \geq \frac{U_M}{I_{FSM}}; \quad (7)$$

- если в БП используется балластный конденсатор, емкость ко-

торого находится в диапазоне значений 0,33...3 мкФ, то в качестве токоограничивающего резистора целесообразно использовать высокочастотные дроссели ДПМ-0,1, ДМЗ-0,1 [6], омическое сопротивление обмоток которых отвечает условию (7). При емкости балластного конденсатора, не превышающей 0,22 мкФ, для ограничения его зарядного тока допустимо использовать популярны резисторы типов С2-33, МЛТ с предельным рабочим напряжением не менее 700 В.

В устройстве управления водоснабжением, выпускаемом мелким серийно нашей фирмой уже более восьми лет, используются диоды КД226А, батарея балластных конденсаторов с суммарной емкостью 4 мкФ и токоограничивающий дроссель ДПМ-0,1-240 ($r \approx 5$ Ом). Случаев отказов диодов не было. А вот ограничительные резисторы, пока мы не перешли на дроссели, отказывали весьма часто.

Вернемся, однако, к схеме на рис. 8. Некоторые результаты двухвариантного анализа этого БП приведены в табл. 5 (тестирование производилось при номинальном напряжении сети).

В этой таблице: R_{LOAD} — сопротивление нагрузки БП; $I_{OUT\ AVG}$ — средний (за период сети) выходной ток БП; $U_{OUT\ AVG}$ — среднее выходное напряжение БП; $U_{OUT\ PP}$ — размах пульсаций выходного напряжения БП.

Компоненты БП, работающего в режиме холостого хода выхода, рассеивают следующие средние мощности (через косую черту указаны значения, соответствующие напряжению сети 220 и 242 В): транзистор VT1 — $P_{VT1\ AVG} = 8,32/1,47$ Вт; резистор R2 — $P_{R2\ AVG} = 8,07/0,85$ Вт; дроссель L1 ($R = 6$ Ом) — $P_{DR\ AVG} = 0,11/0,14$ Вт; микросхема DA1 — $P_{DA1\ AVG} = 77,9/80,1$ мВт; суммарная мощность, рассеиваемая компонентами — $P_{TOTAL\ AVG} = 2,21/2,54$ Вт.

Часто для питания РЭА требуется несколько напряжений. На рис. 12

Таблица 5

$R_{LOAD}, \text{ Ом}$	$R2 = 51 \text{ Ом}$			$R2 \text{ отсутствует}$		
	$I_{OUT\ AVG}, \text{ mA}$	$U_{OUT\ AVG}, \text{ В}$	$U_{OUT\ PP}, \text{ mВ}$	$I_{OUT\ AVG}, \text{ mA}$	$U_{OUT\ AVG}, \text{ В}$	$U_{OUT\ PP}, \text{ mВ}$
∞	0	12,08	0,244	0	12,08	9,868
1000	12,08	12,08	0,238	12,08	12,08	40,54
200	60,41	12,08	0,230	60,19	12,04	218,8
150	80,54	12,08	0,256	79,97	11,99	314,7
120	100,7	12,08	0,494	99,38	11,93	426,4
115	105,0	12,08	1,298	103,5	11,91	454,4
112	107,9	12,08	4,394	106,2	11,89	472,9
110	109,8	12,08	16,34	108,0	11,88	486,1
108	111,7	12,06	66,29	109,9	11,87	500,0
100	116,0	11,60	291,8	118,1	11,81	562,9
90	116,5	10,48	284,9	118,8	10,69	557,9

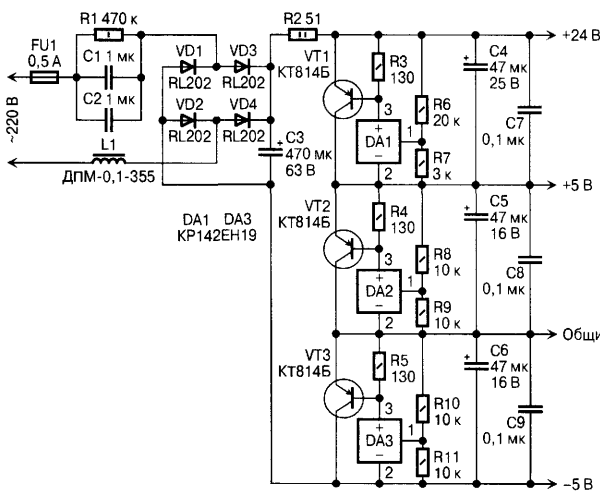


Рис. 12

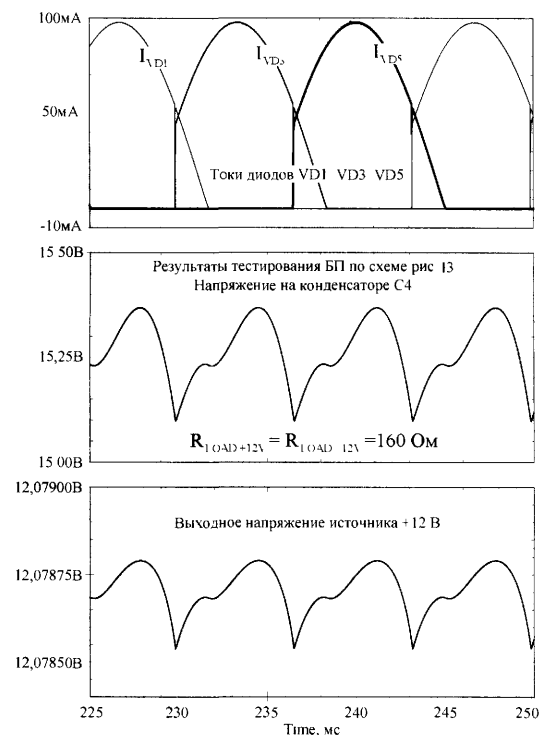


Рис. 14

изображена схема БП, позволяющая получить три стабилизированных напряжения с номинальными уровнями +5 В, -5 В и +24 В.

Очевидно, что изменяя номиналы резисторов R6—R11, можно получить любые уровни выходных напряжений из диапазона значений 2,5...36 В. При расчете БП с несколькими выходными напряжениями необходимо в формулы (2), (3) подставлять величину U_{OUT} , равную сумме всех выходных напряжений (для схемы на рис. 12: $U_{OUT} = 5 + 5 + 24 = 34$ В). Если внешние нагрузки подключены между шиной «Общий» и одной из шин «+5 В», «-5 В», «+24 В», то выходные токи БП должны отвечать следующим условиям:

$$I_{+24В} + I_{+5В} < I_0, \quad (8)$$

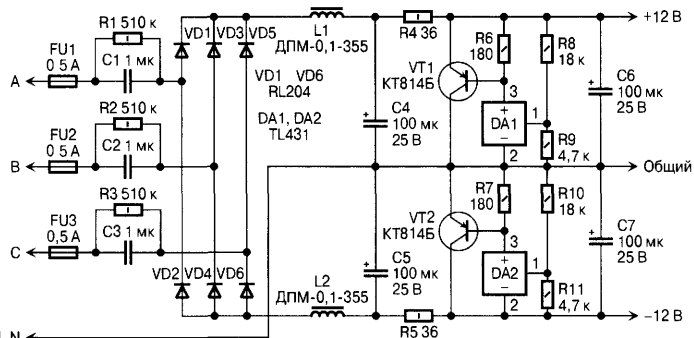


Рис. 13

Таблица 6

R_{LOAD}, Ω	$I_{OUT AVG}, \text{mA}$	$U_{OUT AVG}, \text{B}$	$U_{OUT PP}, \text{mB}$	$U_{C4 PP}, \text{mB}$	$P_{VT1 AVG}, \text{mBT}$
∞	0	12,08	0,128	271	1003
180	67,1	12,08	0,160	271	203
160	75,5	12,08	0,252	271	104
150	80,5	12,08	0,905	270	45,7
140	86,2	12,07	23,1	268	2,13
137	87,5	11,99	49,1	258	0,00

$$I_{-5В} < I_0, \quad (9)$$

где I_0 — максимальный выходной ток, рассчитанный по формуле (2); $I_{+24В}$, $I_{+5В}$, $I_{-5В}$ — выходные токи каналов БП с номинальными уровнями напряжения, равными соответственно +24, +5 и -5 В.

Существует целый ряд приборов промышленной автоматики, например, автоматика управления работой водонапорной башни, автоматика управления работой канализационной станции, устройства защитного (аварийного) отключения трехфазных асинхронных электродвигателей, которые объединяет следующее — все они обычно встраиваются в силовой шкаф управления, в который введены три фазы — А, В, С и нейтраль N промышленной сети 380 В, 50 Гц. В этом случае электропитание устройств автоматики может быть осуществлено от БП, вариант схемы которого изображен на рис. 13.

Работу этого БП поясняют временные диаграммы, изображенные на рис. 14.

На рис. 14 хорошо видно, что частота первой гармоники пульсаций выходного напряжения БП равна 150 Гц, т. е. утроенной частоте сети. Максимальный выходной ток БП рассчитывают по формуле:

$$I_0 [\text{mA}] \approx 1,5 S_{FC} C [\text{мкФ}] \times \frac{62 - 0,2(U_{OUTP} + |U_{OUTM}| + 2U_{FWD})}{1 + 0,0002(R_4 + R_5)}, \quad (10)$$

где С — емкость конденсаторов C1—C3; U_{OUTP} , U_{OUTM} — положительное и отрицательное выходное напряжение БП.

Результаты анализа работы БП при номинальном напряжении сети приведены в табл. 6.

Напряжения на конденсаторах C4, C5 не превышает 15,4 В, а мощность рассеивания резисторов R4, R5 — 282 мВт. Сравнение данных табл. 5 и 6, а также схем рис. 8 и 13 позволяет сделать следующий вывод — важное достоинство БП по рис. 13 заключается в том, что он обеспечивает низкий уровень пульсаций при малой емкости конденсаторов C4, C5 и малом сопротивлении резисторов R4, R5.

ЛИТЕРАТУРА:

- Л. Р. Нейман, К. С. Демирчан. Теоретические основы электротехники. В 2-х томах, том 1. — М.—Л.: Энергия, 1966.
- Л. А. Бессонов. Теоретические основы электротехники. Изд. 6-е, перераб. и доп. Учебник для студентов энергетических и электротехнических вузов. — М.: Высш. школа, 1973.
- М. Л. Волин. Паразитные процессы в радиоэлектронной аппаратуре. — 2-е изд., перераб. и доп. — М.: Радио и связь.
- Е. И. Манаев. Основы радиоэлектроники. Учебное пособие для вузов. — М.: Советское радио, 1976.
- Дж. Барнс. Электронное конструирование: Методы борьбы с помехами. Пер. с англ. — М.: Мир, 1990.

Сергей Лоцицкий,
г. Брянск

Широкополосный усилитель мощности с защитой от перегрузки

Необходимым элементом многих радиотехнических и измерительных систем являются широкополосные усилители мощности, к которым предъявляются требования сохранения работоспособности при неблагоприятных внешних воздействиях. Описание такого усилителя приведено в данной статье.

Технические характеристики усилителя

Полоса рабочих частот, МГц	40...600
Неравномерность АЧХ, дБ	±2
Максимальная выходная мощность, Вт	≥ 6
Коэффициент усиления, дБ	36
Сопrotивление генератора и нагрузки, Ом	50
Напряжение первого источника питания, В	+12...20
Потребляемый ток от первого источника, А	0,5
Напряжение второго источника питания, В	+22...30
Потребляемый ток от второго источника, А	1,8
Габаритные размеры, мм	140×110×3

Принципиальная схема усилителя приведена на рис. 1.

Усилитель содержит четыре предварительных каскада на транзисторах VT2, VT5, VT7, VT9, включенных по схеме с общим эмиттером; выходной каскад на транзисторе VT12, включенном по схеме со сложением напряжений [1]; управляемый стабилизатор напряжения на микросхеме DA1 и транзисторе VT4; два стабилизатора

напряжения на VD6, VT10 и VD7, VT13; цепь защиты от перегрузки по входу на диоде VD1; цепь защиты от рассогласования по выходу на диодах VD8 и VD9; защиту на терморезисторе R27.

Все усилительные каскады работают в режиме класса А с фиксированной рабочей точкой с токами покоя транзисторов VT2, VT5, VT7, VT9, VT12, равными 0,2; 0,3; 0,4; 0,7; 0,7 А соответственно. Стабилизация токов покоя достигается применением цепей активной коллекторной термостабилизации [2]. Установка токов осуществляется подбором номиналов резисторов R7, R13, R23, R32, R37. Увеличение сопротивления этих резисторов приводит к увеличению тока потребления, и наоборот.

В первом каскаде усилителя применена корректирующая цепь пятого порядка [3], при этом одним из элементов корректирующей цепи является индуктивная составляющая входного импеданса транзистора [4]. Указанная цепь обладает большой степенью свободы и используется для подстройки амплитудно-частотной характеристики усилителя после его покаскадной настройки. В промежуточных каскадах использованы

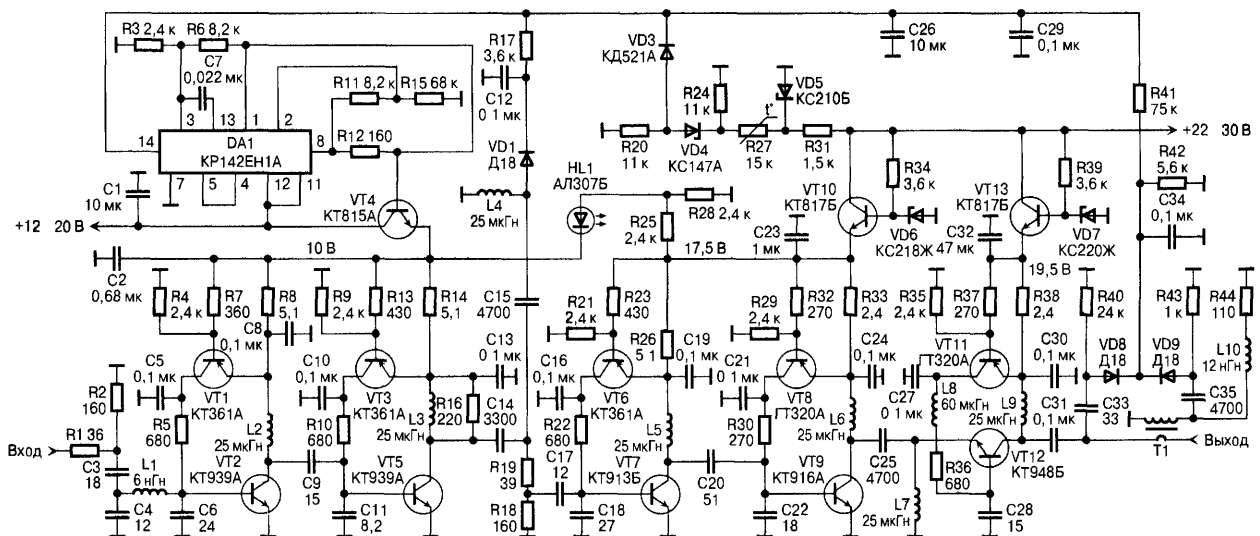
корректирующие цепи третьего порядка [3].

Выходной каскад на транзисторе KT948B выполнен по схеме с общей базой с коэффициентом усиления по напряжению, равным 2,5. Заданный коэффициент усиления устанавливается соответствующим выбором емкости конденсатора C28 [1].

Управляемый стабилизатор напряжения на микросхеме DA1 и транзисторе VT4 используется для осуществления автоматической регулировки усиления усилителя при срабатывании одной из цепей защиты.

При превышении входным сигналом номинального уровня напряжение с выпрямителя на диоде VD1, поступая на управляющий выход микросхемы DA1, уменьшает напряжение питания первых двух каскадов. Это ведет к снижению общего коэффициента усиления и двустороннему ограничению усиливаемого сигнала в этих каскадах. Уровень срабатывания цепи защиты от перегрузки по входу устанавливается подбором резистора R17. Включение светодиода HL1 в этом случае свидетельствует о перегрузке усилителя по входу.

Цепь защиты от рассогласования по выходу препятствует выходу из строя оконечных транзисторов VT9 и VT12 при работе на несогласованную нагрузку. Она состоит из датчика выходного напряжения на диоде VD8 и датчика выходного тока на диоде VD9 и трансформаторе T1 [5]. Датчик выходного тока работает следующим образом. При протекании выходного тока по первичной обмотке во вторичной наводится напряжение, пропорциональное протекающему току. Оно детектируется диодом VD9. Цепь L10R44 служит для выравнива-



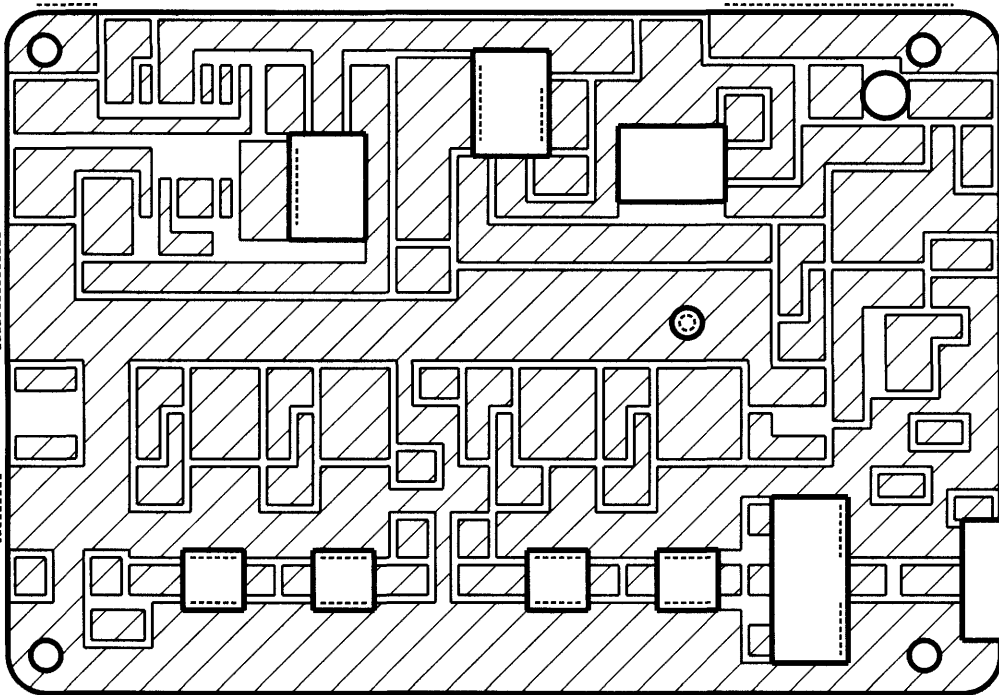


Рис. 2

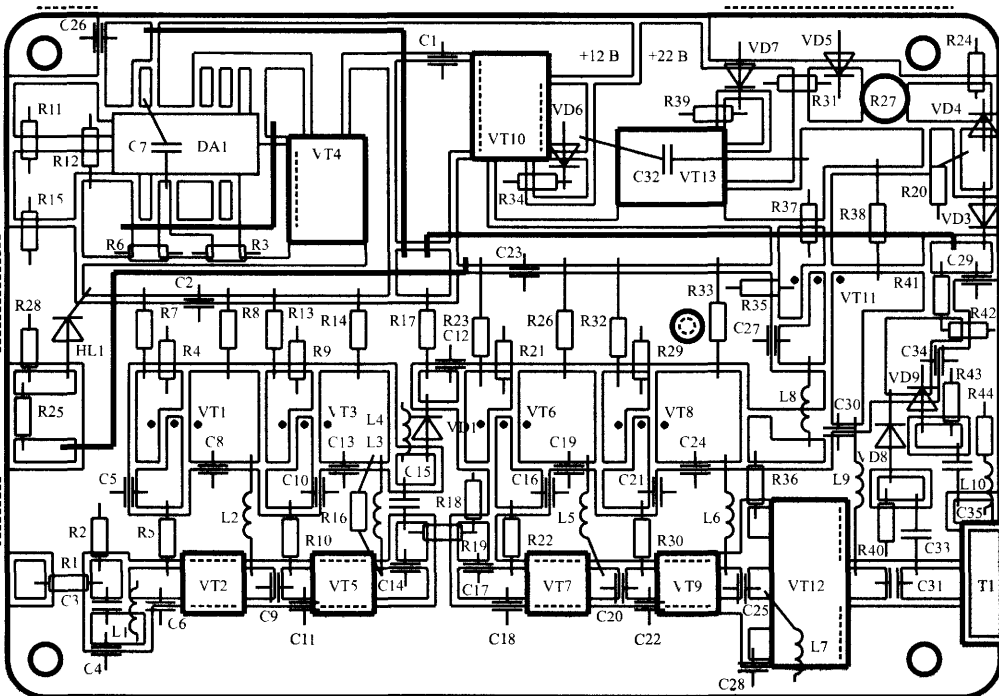


Рис. 3

ния коэффициента передачи датчика выходного тока в полосе рабочих частот усилителя. При работе на нагрузку 50 Ом датчики выходного напряжения и выходного тока выдают одинаковые управляющие напряжения, пропорциональные уровню выходного сигнала, при работе на нагрузку сопротивлением более 50 Ом сигнал управления формируется датчиком выходного напряжения, менее 50 Ом — датчиком выходного тока. В результате использование рассматриваемой системы защиты позволяет при работе на высокоомную нагрузку

реализовать максимальное выходное напряжение, равное максимальному напряжению, развиваемому усилителем на нагрузке 50 Ом. При работе на низкоомную нагрузку максимальный ток в нагрузке равен максимальному току, отдаваемому усилителем в нагрузку 50 Ом. Уровень срабатывания защиты от рассогласования по выходу устанавливается подбором резистора R41.

Срабатывание системы защиты при перегреве усилителя приводит к уменьшению до нуля напряжения питания входных каскадов и вклю-

чению светодиода HL1. В этом случае светодиод будет светиться и при отключении генератора усиливаемых сигналов от входа усилителя. Температура срабатывания цепи защиты устанавливается подбором резистора R24.

Управляемый стабилизатор напряжения на микросхеме DA1 и транзисторе VT4 и два стабилизатора напряжения на транзисторах VT10 и VT13 обеспечивают сохранение технических характеристик усилителя при его работе от аккумуляторных батарей, напряжение которых в процессе эксплуатации изменяется в пределах +12...20 В и +22...30 В. Подача питания усилителя через проходные конденсаторы, как это видно из фотографии (рис. 4), исключает возможность прохождения усиленного сигнала по цепям питания в маломощные каскады, в которой используется рассматриваемый усилитель.

Печатная плата (рис. 2, 3) размерами 130×90 мм изготавливается из фольгированного с двух сторон стеклотекстолита толщиной 2 мм. Штриховыми линиями обозначены места металлизации торцов, что может быть сделано с помощью медной фольги, которая припаивается к нижней и верхней частям платы. Металлизация необходима для устранения паразитных резонансов и соединения с общим проводом нужных участков печатной платы.

Транзисторы VT2, VT5, VT7, VT9, VT12 крепятся к основанию с использованием теплопроводящей пасты, VT4, VT10, VT13 установлены с использованием слюдяных прокладок и теплопроводящей пасты. Особо следует остановиться на креплении транзистора VT12. Транзистор КТ948Б предназначен для работы в схеме с общей базой, однако в данном случае между базой и общей шиной включен конденсатор C28, конструктивно образованный основанием усилителя и корпусом транзистора КТ948Б. Для отвода тепла от транзистора между ним и основанием

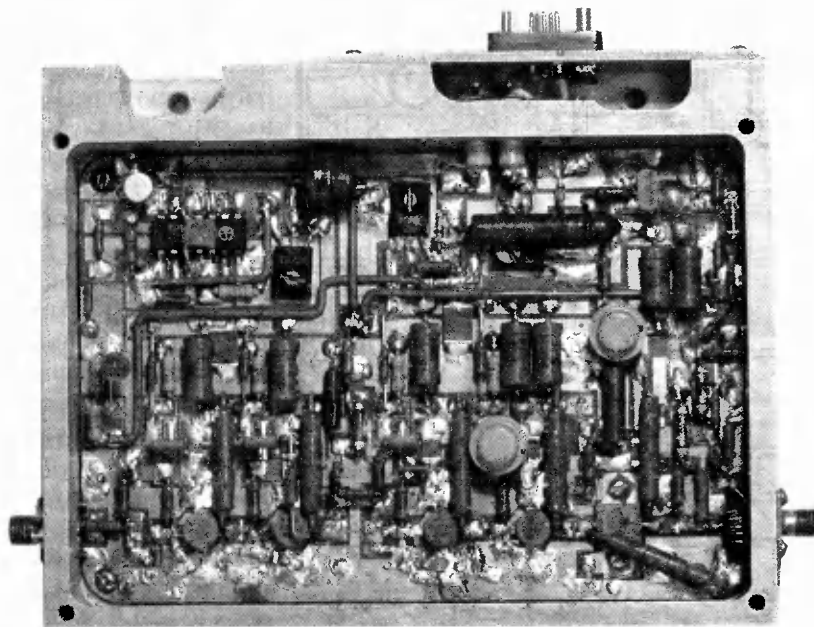


Рис. 4

устанавливается керамическая подложка толщиной 1 мм. Крепление транзистора VT12 осуществляется с помощью винтов, на которые надеты изолирующие шайбы и трубки. Перед установкой крепежных винтов под их головки подкладывают полоски медной фольги с отверстиями. К этим полоскам в дальнейшем припаивают резистор R36 (рис. 4) и дополнительные конденсаторы, входящие в состав C28 и обеспечивающие настройку усилителя на максимальную выходную мощность. При сборке усилителя следует минимизировать длину цепи, связывающей коллектор транзистора VT9 с эмиттером транзистора VT12. Это обусловлено тем, что наличие индуктивной составляющей указанной цепи приводит к неполному сложению сигнальных напряжений, отдаваемых двумя выходными транзисторами (VT9 и VT12). В этом случае потери выходной мощности в области верхних частот полосы пропускания усилителя могут достигать значительной величины.

Трансформатор T1 намотан на сердечнике из феррита M100NH типоразмера 10×6×3. Первичная обмотка — провод, пропущенный сквозь отверстие сердечника, вторичная содержит 5...7 витков провода диаметром 0,5...0,8 мм. Фотография смонтированного усилителя приведена на рис. 4.

При настройке усилителя сначала с помощью резистора R6 устанавливают выходное напряжение стабилизатора,

питающего входные каскады, равным 10 В. Подбором номиналов резисторов R7, R13, R23, R32, R37 устанавливают токи покоя транзисторов VT2, VT5, VT7, VT9, VT12 на 20...30 % меньше заданных. Затем в режиме малого сигнала производится покаскадная регулировка амплитудно-частотной характеристики усилителя. Для этого в качестве нагрузки транзистора VT2 через разделительный конденсатор подключают резистор номиналом 50 Ом. Подбором емкости конденсатора C4 добиваются равномерной АЧХ каскада в области нижних и средних частот полосы пропускания, а подбором емкости конденсатора C6 — выравнивания АЧХ в области верхних частот полосы пропускания. Если этого не удастся достичь, следует уменьшить емкость конденсатора C3 и повторить настройку. Далее к первому каскаду подключают второй и с помощью конденсатора C11 выравнивают АЧХ двух каскадов и т. д. После этого с помощью входной корректирующей цепи осуществляют подстройку АЧХ всего усилителя.

После формирования АЧХ усилителя в режиме малого сигнала проверяют максимальный уровень его выходной мощности в полосе рабочих частот. Введением дополнительного конденсатора, входящего в состав C28, следует добиться выравнивания максимального уровня выходной мощности усилителя в полосе рабочих частот. И, наконец, варьируя

значениями тока покоя транзисторов усилителя необходимо найти значения указанных токов, при которых усилитель отдает в нагрузку максимальную мощность.

Теперь осуществляют настройку узла защиты. В режиме максимальной выходной мощности необходимо подобрать номинал резистора R17, при котором незначительное увеличение входного сигнала будет приводить к включению светодиода HL1, что соответствует срабатыванию защиты от перегрузки по входу. Не впаивая резистор R41 изменением номиналов элементов C33, R40 и L10, R44 настраивают датчики выходного напряжения и выходного тока таким образом, чтобы при работе на нагрузку 50 Ом они выдавали одинаковые управляющие напряжения, пропорциональные уровню выходного сигнала. Затем подбирают номинал резистора R41, при котором в режиме максимальной выходной мощности усилителя начинается слабое свечение светодиода HL1. И, наконец, выбором резистора R24 устанавливают температуру срабатывания цепи защиты от перегрева.

ЛИТЕРАТУРА:

1. А. А. Титов, В. Н. Ильющенко, Б. И. Авдоchenko, В. Д. Обихвостов. Широкополосный усилитель мощности для работы на несогласованную нагрузку. — ПТЭ, 1996, № 2, с. 68, 69.
2. А. А. Титов. Расчет схемы активной коллекторной термостабилизации и ее использование в усилителях с автоматической регулировкой потребляемого тока. — Электронная техника. Сер. СВЧ-техника, 2001, № 2, с. 26—30.
3. А. А. Титов. Расчет межкаскадной корректирующей цепи многокаскадного транзисторного усилителя мощности. — Радиотехника, 1987, № 1, с. 29—31.
4. А. А. Титов, Л. И. Бабак, М. В. Черкашин. Расчет межкаскадной согласующей цепи транзисторного полосового усилителя мощности. — Электронная техника. Сер. СВЧ-техника, 2000, вып. 1, с. 46—50.
5. А. А. Титов, В. Н. Ильющенко. Широкополосный датчик высокочастотного тока. Свидетельство на полезную модель № 32883 Российского агентства по патентам и товарным знакам. Оpubл. 27.09.2003, бюл. № 27.

Александр Титов,
г. Томск

Цифровой комбинированный регулятор на ПЛИС фирмы Altera

Разработан и исследован цифровой комбинированный регулятор, реализованный на программируемой логической интегральной схеме (ПЛИС) фирмы Altera.

Совершенствование технологии производства интегральных полупроводниковых компонентов, обеспечивающее возможность создания высокоэкономичных цифровых устройств обработки и хранения информации, а также появление эффективных средств программирования, оказывает все более существенное влияние на развитие техники измерения и управления. На основе цифровых регуляторов могут быть построены системы управления любых типов. Программное обеспечение подобных систем можно без труда корректировать как в предпусковой период, так и в процессе эксплуатации. Немаловажно и то, что цифровые регуляторы позволяют изменять их параметры в широком диапазоне и способны работать практически с любыми периодами квантования.

Формирование и развитие микропроцессорной техники способствовало широкому использованию при проектировании цифровых систем программируемой логики. Если разработчик систем на основе устройств с жесткой структурой может пользоваться для реализации необходимых функций только аппаратными средствами, а при использовании ЭВМ в большей степени программными средствами, то при построении систем на основе программируемой логики он получает в свое распоряжение полностью как аппаратные, так и программные средства.

Развитием программируемой логики стало появление программируемых логических интегральных схем, представляющих собой платформу для создания реконфигурируемых и высокопроизводительных цифровых систем и устройств с минимальными материальными затратами и сокращенным временем на проектирование.

При разработке децентрализованной системы цифрового управления для АСУ тепломассообменными аппаратами центрального кондиционера решалась задача стабилизации пара-

метров процессов воздухообработки на заданных уровнях в условиях изменения одного из термодинамических параметров воздуха, например, теплосодержания.

Цифровая автоматическая система регулирования (АСР) на ПЛИС обладает особенностями, отличающими ее от аналоговых систем:

- законы управления реализуются в форме алгоритмов, запрограммированных с помощью аппаратных или программных средств;
- обрабатываются квантованные (дискретные во времени) сигналы;
- сигналы могут принимать только определенные дискретные значения вследствие квантования по амплитуде.

При разработке алгоритма управления использовались данные о характере динамики объекта регулирования (ОР) — теплообменного аппарата установки кондиционирования воздуха [1].

Для того, чтобы цифровая АСР обладала свойствами, близкими к свойствам аналоговой АСР, период квантования T_K выбирается достаточно малым.

Во многих случаях при построении алгоритмов управления динамическими процессами исходят из принципов действия хорошо исследованных классических линейных регуляторов: пропорционального (П), пропорционально-интегрального (ПИ) и пропорционально-интегрально-дифференциального (ПИД). Несмотря на то, что наиболее распространенными являются алгоритмы ПИ и ПИД, для цифровых систем регулирования часто используется более простой в реализации пропорциональный закон регулирования.

Алгоритм цифрового П-регулятора, формирующего управляющее воздействие, легко вывести, используя [2]:

$$u_1(k) = K_p e(k), \quad (1)$$

где $e(k) = y(k) - y_0$ — отклонение выходной переменной от заданного значения; K_p — коэффициент передачи П-регулятора; $k = 1, 2, 3, \dots$ — номер периода квантования.

В целях обеспечения минимального динамического отклонения регулируемого параметра, а также минимальной ошибки регулирования, оптимальное численное значение коэффициента передачи K_p определяют в результате моделированием ОР с учетом его реальных параметров [1].

Исследование цифровой одноконтурной АСР на основе П-регулятора, реализующей уравнение (1), в MATLAB 6.5-Simulink показало наличие статической ошибки регулирования, превышающей допустимые пределы, предусмотренные требованиями к АСР центральных кондиционеров. Проведение синтеза оптимального регулятора, дающего максимальные показатели качества регулирования, экономически не оправдано.

Для обеспечения требуемых показателей качества проведен синтез комбинированной АСР. В основу синтеза положены методы, используемые при аналитическом конструировании комбинированных систем регулирования, при этом учитывались ранее определенные реальные динамические свойства технологического оборудования центрального кондиционера [1]. В целях компенсации возмущающих воздействий проведен синтез корректирующего звена АСР на основе использования прямого метода Ляпунова.

Управляющее воздействие u_2 , формируемое корректирующим звеном АСР, в аналоговой форме в соответствии с [3] определяется по формуле:

$$\frac{du_2}{dt} = -p(u_2 - K_0 f - y_0 M). \quad (2)$$

Для малых периодов квантования алгоритм цифрового управления, реализующий (2) и записанный в конечных разностях с учетом приращения управляющего сигнала от корректирующего звена, можно найти по формуле:

$$u_2(k) - u_2(k-1) = Bf(k) + Cy_0, \quad (3)$$

где $A = \frac{1}{1 + T_K p}$; $B = \frac{T_K p K_0}{1 + T_K p}$; $C = \frac{T_K p M}{1 + T_K p}$.

Алгоритм работы всей системы управления при использовании цифрового модуля ОР:

$$u(k-1) = u(k)$$

После отладки компонентов схемы на основе созданных при компиляции выходных файлов осуществлено моделирование с

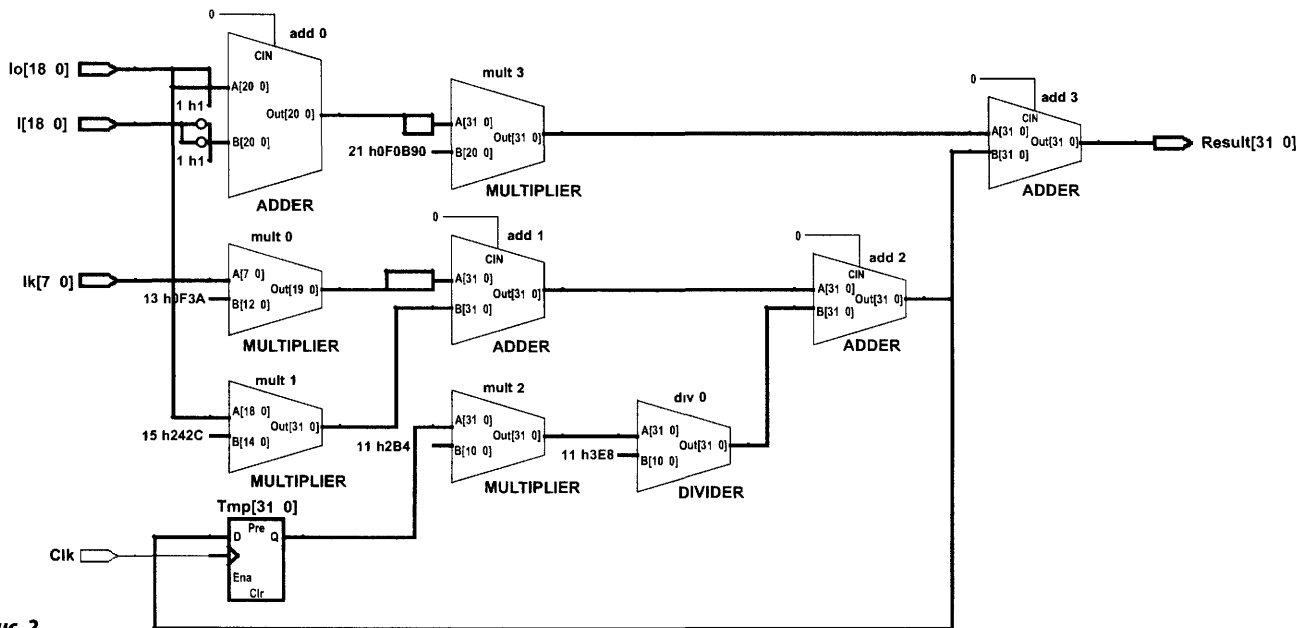


Рис. 2

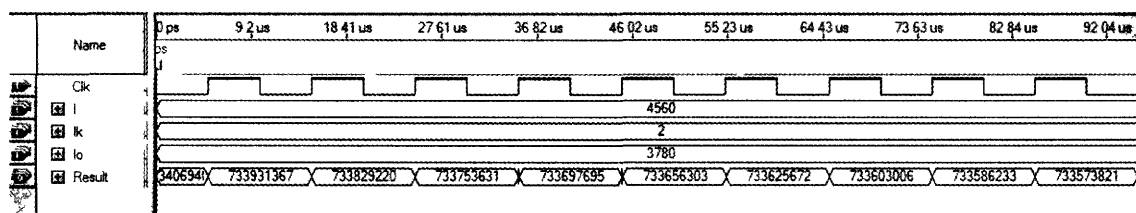


Рис. 3

помощью подсистемы (Simulator) пакета Quartus II 4.1. При этом проверялись логические операции и внутренние временные соотношения проекта. Компиляция выполнена с учетом заданных требований — обеспечения требуемых временных характеристик проекта; увеличения быстродействия; оптимизации используемых ресурсов ПЛИС. В результате компиляции создан файл для программирования и конфигурирования ПЛИС Altera. Полученная схема проекта для конфигурации в ПЛИС приведена на рис. 2.

При компиляции проекта выбрана ПЛИС семейства Cyclone — устройство EP1C4F324C6.

Функциональным моделированием проверена правильность описания и логического функционирования проекта. Временным моделированием проверены временные параметры разработки. Фрагмент файла с результатами моделирования цифрового регулятора в зависимости от кодов входного сигнала и соответствующих коэффициентов показан на рис. 3.

Проверка АСР моделированием в САПР подтвердила правильность ее функционирования.

Широкие функциональные возможности регулятора позволяют использовать его для управления как теплосодержанием, так и температурой, относительной влажностью, влагосодержанием воздуха, давлением, расходом, уровнем жидкости или любой другой физической величиной. При необходимости на входы регулятора могут быть подключены соответствующие датчики с унифицированными цифровыми выходами. В случае использования АСР для управления теплосодержанием или влагосодержанием его входы могут быть соединены с выходами цифрового устройства определения термодинамических параметров воздуха, выполненного также на ПЛИС [6]. В результате образуется комплект регулятора системы кондиционирования воздуха, состоящий из двух тесно сотрудничающих устройств. Параметры регулятора могут быть заданы вручную, либо с помощью автоматической настройки (активной или пассивной).

ЛИТЕРАТУРА:

1. В. В. Вычужанин, В. Б. Медзеновский. Исследование характеристик судового центрального кондиционера. — Холодильная техника, 1984, № 3, с. 28—31.
2. Р. Изерман. Цифровые системы управления. — М.: Мир, 1984.
3. В. В. Вычужанин. Комбинированная автоматическая система регулирования параметров воздуха в кондиционере. — Холодильная техника, 1985, № 9, с. 43, 44.
4. Д. А. Комолов, Р. А. Мяльк, А. А. Зобенко. Системы автоматизированного проектирования фирмы AlteraMAX + plus II и Quartus II. — М.: ИП РадиоСофт, 2002.
5. Е. А. Суворова, Ю. Е. Шейнин. Проектирование цифровых систем на VHDL. — С-Пб.: БХВ-Петербург, 2003.
6. В. В. Вычужанин, С. В. Малюта. Цифровое устройство определения термодинамических параметров воздуха на ПЛИС фирмы Altera. — Схемотехника, 2005, № 4, с. 30—33.

Владимир Вычужанин,
г. Одесса, Украина

Borland C++ Builder 6

для начинающих

В этой статье рассказывается о назначении и применении на практике некоторых методов визуальных компонентов.

МЕТОДЫ КОМПОНЕНТОВ

Наряду с обработчиками событий, Borland C++ Builder 6 позволяет использовать при написании программы множество методов компонентов. Мы уже применяли в рассмотренных программах некоторые из них, например, LoadFromFile, SaveToFile и др. В отличие от обработчиков событий, методы не вызываются в результате каких-либо действий пользователя, а выполняются при достижении программой строки, в которой они записаны. Методы вставляются в программу самим разработчиком по мере необходимости, например для отображения рисунка на экране монитора, работы с файлами и т. д. По сути дела методы являются функциями или подпрограммами. Так же как свойства и события, одни и те же методы могут быть применимы для разных типов компонентов. В табл. 1 приведены краткие описания большинства методов, расположенных в алфавитном порядке.

Рассмотрим более подробно назначение и описание некоторых наиболее часто используемых методов с демонстрацией их применения.

Начнем с создания небольшой программы, которая наглядно покажет нам, как применяются методы Add (добавить), Hide (скрыть), Show (показать) и Delete (удалить). Создайте новое приложение и поместите на его форму компонент ListBox и четыре кнопки Button. Измените размеры компоненты так, как показано на рис. 1.

Задайте свойство названия Caption для формы и кнопок в соответствии с этим же рисунком. Щелкните дважды левой кнопкой мыши по компоненту первой кнопки Button1 и впишите для нее код обработки события OnClick в виде программной строки

```
ListBox1->Items->Add(String(1++));
```

Эта строка содержит метод Add, который позволит добавить в компонент ListBox1 элемент списка, представляющий собой строку, формирующуюся из переменной i с по-

мощью функции String. Каждый раз при вызове данной программной строки с методом Add переменная i будет увеличивать свое значение благодаря тому, что после нее вписан оператор автоинкремента (увеличения значения на единицу) «++». Описание переменной с присвоением ей начального значения в виде строки int I = 0; необходимо вписать в программу в инспекторе объектов сразу после строк с описанием включений #include. В этом случае переменная будет доступна любому обработчику программы. Если описать эту переменную в самом обработчике события, она станет локальной (недоступной другим обработчикам) и будет инициализироваться каждый раз при формировании этого события.

Аналогично сформируйте заготовку кода обработки события OnClick для компонента кнопки Button2 и впишите в него программную строку

```
ListBox1->Hide();
```

Эта строка содержит метод Hide, который скрывает изображение компонента на форме от пользователя программы.

Подобным образом впишите в код обработки события OnClick для компонента кнопки Button3 программную строку

```
ListBox1->Show();
```

Эта строка содержит метод Show, который показывает изображение компонента на форме для пользователя программы.

Наконец, впишите код обработки события OnClick для компонента кнопки Button4 в виде программной строки

```
if(i>0) ListBox1->Items->Delete(--i);
```

Эта строка содержит метод Delete, который удаляет указанный элемент списка из компонента ListBox1. Обратите внимание на то, что в качестве номера удаляемого элемента списка

мы использовали ту же переменную i, которая применялась при добавлении элементов списка в компонент. При этом перед ее использованием она автоматически декрементируется (уменьшает свое значение на единицу) благодаря применению оператора «--». Метод удаления имеет смысл выполнять только в том случае, если элементы в списке есть, т. е. имеется что удалять. Для проверки данного условия в программную строку включен оператор сравнения переменной индекса элементов списка с нулем if(i > 0), который позволяет выполнять удаление элемента из списка только при натуральном значении переменной i, т. е. когда элементы в списке есть. Сохраните созданный проект и запустите приложение на выполнение, после чего проверьте, как оно работает.

Теперь можно рассмотреть некоторые детали методов, которые часто используются в программах.

Метод Add применяется к компонентам, имеющим списки или строки. Метод возвращает при выполнении индекс добавленного элемента. Например, программная строка

```
i = ListBox1->Items->Add(«Новый элемент списка»);
```

присвоит переменной i значение возвращаемого параметра. Индекс первого элемента списка всегда равен нулю. Данный метод увеличивает значение свойства Count (номер элемента) на единицу. Если значение Count равно значению свойства Capacity (размер списка), то увеличивается значение Capacity и перераспределяется память под новые элементы. Напомню, что доступ к свойствам Count и Capacity осуществляется с помощью строк ListBox1->Item->Count и ListBox1->Item->Capacity. Используя данные свойства можно исключить в предыдущей программе переменную i, сохранив при этом все ее функции. Предлагаю читателям проделать это самостоятельно для получения собственного опыта.

Метод Hide делает компонент невидимым, задавая свойству Visible компонента значение False. Если компонент является контейнером для других компонентов, то эти дочерние компоненты также делаются невидимыми. Несмотря на то, что компонент становится невидимым, его свойства и методы остаются доступными в программе.

Таблица 1

Метод	Описание
Add	Добавляет новый элемент в список
Assign	Копирует изображение одного графического объекта в другой
BeginDrag	Используется при перетаскивании компонента
BringToFront	Позволяет изменять последовательность перекрытия компонентов
BrushCopy	Копирует часть изображения битовой матрицы на канву
CanFocus	Определяет, может ли компонент получать сообщения пользователя
ChangeScale	Используется для изменения масштаба компонента
Chord	Рисует заполненную замкнутую фигуру, ограниченную дугой окружности или эллипса и хордой
ClassName	Возвращает имя типа объекта
Clear	Очищает список или текст
ClientToScreen	Преобразует координаты клиентской области в координаты экрана
ContainsControl	Определяет, является ли указанный компонент прямым или косвенным наследником данного оконного компонента
ControlAtPos	Определяет, какой дочерний компонент имеется в указанной позиции
CopyRect	Копирует часть изображения
Delete	Удаляет элемент с указанным индексом из списка
DisableAlign	Временно запрещает выравнивание дочерних компонентов
Dormant	Создает изображение битовой матрицы в памяти
Draw	Рисует графическое изображение в указанную позицию канвы
DrawFocusRect	Рисует изображение прямоугольника в виде, используемом для отображения рамки фокуса, с помощью операции XOR
DisableAlign	Временно запрещает выравнивание дочерних компонентов
Ellipse	Рисует заполненный эллипс или окружность
Expand	Увеличивает емкость списка типа TList
FillRect	Заполняет указанный прямоугольник канвы, используя текущее значение Brush
FindNextControl	Возвращает следующий в последовательности табуляции оконный дочерний компонент
FloodFill	Закрашивает текущей кистью замкнутую область канвы, ограниченную определенным цветом
Focused	Определяет, находится ли оконный элемент в фокусе
FrameRect	Рисует на канве текущей кистью прямоугольную рамку
Free	Освобождает память, занимаемую объектом
GetTabOrderList	Строит список дочерних оконных компонентов в последовательности табуляции
HandleAllocated	Проверяет наличие дескриптора окна компонента
HandleNeeded	Создает дескриптор окна, если до этого он не существовал
Hide	Делает компонент невидимым
IndexOf	Определение первого вхождения в список заданного элемента
Insert	Вставляет элемент в заданную позицию списка
Invalidate	Сообщает Windows о необходимости полностью перерисовать компонент
LineTo	Рисует на канве прямую линию
LoadFromClipboardFormat	Загружает изображение из буфера обмена в формате Clipboard
LoadFromFile	Загружает изображение, хранящееся в файле
LoadFromResourceID	Загружает битовую карту из файла ресурсов по указанному идентификатору
LoadFromResourceName	Загружает битовую карту из файла ресурсов по указанному имени
LoadFromStream	Загружает графическое изображение из указанного потока
Lock	Блокирует канву, запрещая рисовать на ней
Move	Меняет текущую позицию элемента в списке на позицию, заданную в параметрах метода
MoveTo	Устанавливает позицию пера на позицию, заданную в параметрах метода
Pie	Рисует заполненную замкнутую фигуру в виде сегмента окружности или эллипса
Polygon	Рисует на канве заполненную замкнутую фигуру в виде многоугольника
Polyline	Рисует на канве ломаную линию
DisableAlign	Временно запрещает выравнивание дочерних компонентов
Rectangle	Рисует на канве закрашенный прямоугольник
Refresh	Перерисовывает изображение компонента на экране
Remove	Удаляет элемент с заданным значением из списка TList
Repaint	Перерисовывает изображение компонента на экране
ReplaceDockedControl	Встраивает компонент на место другого компонента
RoundRect	Рисует на канве прямоугольную рамку со скругленными углами
SaveToClipboardFormat	Создает копию изображения в формате Clipboard
SaveToFile	Сохраняет графическое изображение в файле
SaveToStream	Сохраняет графическое изображение в потоке
ScaleBy	Масштабирует оконный элемент и все содержащиеся в нем компоненты
ScaleControls	Масштабирует дочерние компоненты оконного элемента, не изменяя масштаба самого элемента
ScreenToClient	Преобразует координаты экрана в координаты клиентской области компонента
ScrollBy	Сдвигает содержимое оконного элемента
SelectFirst	Передаёт фокус дочернему компоненту, первому в последовательности табуляции
SelectNext	Передаёт фокус дочернему компоненту, следующему в последовательности табуляции за указанным компонентом
SendCancelMode	Прерывает модальное состояние элемента управления
SendToBack	Переносит компонент ниже других компонентов
SetBounds	Устанавливает одновременно свойства Left, Top, Width и Height
SetChildOrder	Изменяет позицию компонента в списке дочерних компонентов оконного элемента
SetFocus	Передаёт фокус элементу
SetZOrder	Перемещает компонент на вершину или вниз Z-последовательности
Show	Делает видимым невидимый компонент
StretchDraw	Рисует графическое изображение в указанную прямоугольную область канвы, подгоняя размер изображения под заданную область
TextExtent	Возвращает длину и высоту в пикселях текста, который предполагается написать на канве текущим шрифтом
TextHeight	Возвращает высоту в пикселях текста, который предполагается написать на канве текущим шрифтом
TextOut	Пишет указанную строку текста на канве, начиная с указанной позиции
TextRect	Пишет указанную строку текста на канве, начиная с указанной позиции и обрезая текст, выходящий за пределы указанной прямоугольной области
TextWidth	Возвращает в пикселях длину текста, который предполагается написать на канве текущим шрифтом
TryLock	Блокирует канву, если она не была заблокирована, не разрешая другим нитям многопоточного приложения рисовать на ней
Unlock	Уменьшает на единицу значение свойства LockCount, способствуя тем самым разблокировке канвы, когда LockCount станет равным нулю
Update	Перерисовывает компонент

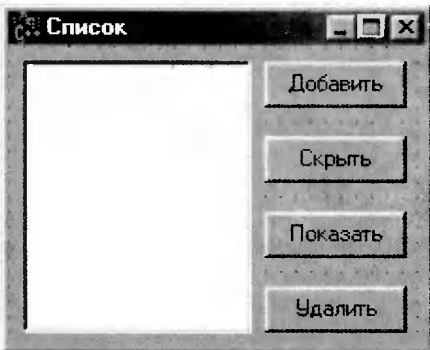


Рис. 1

Метод Show делает видимым ранее невидимый компонент. Он задает значение True свойству Visible компонента и проверяет, является ли видимым родительский компонент.

Метод Delete удаляет из списка элемент с указанным индексом.

Метод CanFocus определяет, может ли компонент получать сообщения пользователя, т. е. может ли он получать фокус. Функция возвращает True, если у компонента и всех его родителей свойства Visible и Enabled установлены в True. В противном случае возвращается значение False.

Метод ChangeScale(M,D) используется для изменения масштаба компонентов. При его применении масштабируются такие свойства компонента, как Top и Left, определяющие его местоположение, а также Width и Height, определяющие его размер.

Параметры метода M и D определяют соотношение новых размеров компонентов к их прежним размерам. Например, чтобы уменьшить размеры компонента до 50 % начального значения, нужно задать M равным 50, а D равным 100. То же самое можно сделать, задав M = 1 и D = 2. Если вы хотите увеличить размер, нужно задать значение M больше, чем значение D. Так, например, программная строка ChangeScale(2,1); увеличит размеры компонентов в два раза.

Следующая группа описываемых методов связана с графическими компонентами и предназначена для рисования. Рассмотрим их применение на примере программы.

Создайте новое приложение и разместите на его форму компонент Image и одну кнопку Button. Установите размеры компонента Image равными 200 единиц по высоте и 300 единиц по ширине с помощью свойства Height и Width. Свойство Caption формы Form1 заполните названием Draw. Задайте свойство Caption компонента Button1, назвав кнопку словом «Рисо-

вать» и расположите компоненты на форме в соответствии с рис. 2.

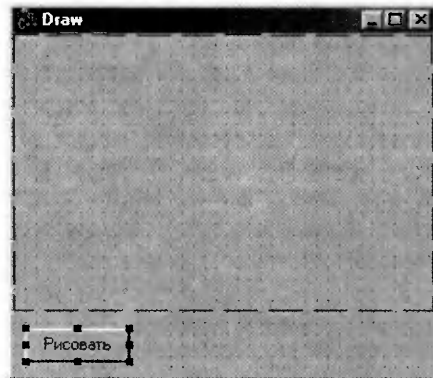


Рис. 2

Объявите целую переменную i с помощью оператора int i; после строки public в файле описаний модуля с расширением h и задайте ее начальное значение с помощью оператора i = 1; в обработчике создания формы Form1.

Сформируйте для кнопки заготовку обработчика события OnClick и вставьте в него следующие программные строки:

```
Image1->Canvas->Pen->Color=c1Black;
Image1->Canvas->Brush->Color=c1Blue;
switch(1){
    case 1: Image1->Canvas->Ellipse(0,0,50,50); break;
    case 2: Image1->Canvas->Chord(50,0,100,50,75,0,100,25); break;
    case 3: Image1->Canvas->Pie(100,0,150,50,125,0,150,25); break;
    case 4: Image1->Canvas->Arc(150,0,200,50,150,0,200,50); break;
    case 5: Image1->Canvas->Rectangle(200,0,250,50); break;
    case 6: Image1->Canvas->RoundRect(250,0,300,50,20,20); break;
    case 7: Image1->Canvas->TextOutA(0,100,"Графический текст"); break;
    case 8: Image1->Canvas->MoveTo(0,200); Image1->Canvas->LineTo(300,0); break;
    case 9: Image1->Canvas->FloodFill(0,0,c1Red,fsBorder);
        Image1->Canvas->Brush->Color=c1White;
        Image1->Canvas->FillRect(Rect(0,0,Image1->Width,Image1->Height));
        i=0;
        break;
    default: break;
}
Button1->Caption="Фигура"+String(++i);
```

Текст этой программы под названием Draw можно найти на сайте журнала по адресу www.dian.ru/programs. Первые две строки данной про-

граммы задают свойства пера (Pen) и кисти (Brush) для канвы (Canvas) рисунка (Image). Далее следует оператор switch, который сравнивает значение переменной i с целыми значениями в диапазоне 1...9. Каждому из этих значений сопоставлена программная строка с применением различных методов графического отображения фигур, заканчивающаяся оператором break для прерывания сравнения. Познакомимся подробнее с этими методами программных строк.

Метод Ellipse(X1, Y1, X2, Y2) рисует эллипс или окружность с помощью текущих параметров пера (Pen). Фигура заполняется текущим значением кисти (Brush). X1, Y1 и X2, Y2 являются координатами верхнего левого и правого нижнего углов прямоугольника, описывающего эллипс. Метод вызывается с помощью строки:

```
Image1->Canvas->Ellipse(X1, Y1, X2, Y2);
```

Операционная система Windows95 накладывает ограничение на параметры этого и последующих методов. Они связаны с тем, что сумма координат прямоугольника не должна превышать значение 32767.

Метод Chord(X1, Y1, X2, Y2, X3, Y3, X4, Y4) рисует фигуру в виде дуги эллипса или окружности с помощью текущих параметров пера (Pen), замкнутую хордой. Фигура заполняется текущим значением кисти (Brush). Координаты X1, Y1 и X2, Y2 определяют прямоугольник, описывающий эллипс. Дуга начинается с точки, являющейся пересечением эллипса с прямой, проходящей через его центр, и точку с координатами X3, Y3. Конечная точка дуги определяется пересечением эллипса с прямой, проходящей через его центр и точку с координатами X4, Y4. Дуга рисуется против часовой стрелки от начальной до конечной точки.

Продолжение следует

Олег Вальпа,
г. Миасс Челябинской обл.

Подготовка программ для микроконтроллеров семейства x51 в среде μ VISION-2

ОТЛАДКА ПРОГРАММЫ

Программная симуляция моделирует большинство особенностей микроконтроллеров семейства x51 и позволяет отладить программу до занесения в конкретную микросхему. Среда μ VISION-2 моделирует разнообразные периферийные устройства, такие как последовательный порт, таймеры, порты ввода/вывода, стандартно присутствующие в микроконтроллерах 8051 (8052), и, кроме того, специфические устройства, присутствующие в некоторых моделях семейства, например аналоговые компараторы.

Запуск отладчика можно осуществить двумя путями, однако сейчас и в дальнейшем для экономии места будем указывать только один.

Переход в режим отладки и возврат обратно осуществляется щелчком мышью по кнопке Start/Stop Debug Session панели инструментов File (левая кнопка на рис. 6).



Рис. 6

Перейдем в режим отладки.

Экран видоизменился, в окне проекта стала активной закладка Regs, на которой отображены регистры контроллера и их значение. Следует отметить, что переход к файловой и справочной системам по-прежнему легко осуществим щелчком мышью по закладкам Files и Books соответственно.

Ввод и вывод дополнительных окон отладчика производится нажатием соответствующих кнопок на панелях инструментов (нажатие на кнопку открывает окно, если оно закрыто, закрывает, если открыто). Во всех окнах щелчком правой кнопки мыши можно вызвать контекстное меню, которые в некоторых случаях не будут описаны в данной статье.

В нижней части экрана выведем дополнительные окна Memory Win-

dow (окно памяти) и Watch&Call Stack Window (окно просмотра и вызова стека — окно переменных) щелчком мыши поочередно по второй и пятой кнопкам панели инструментов «Отладка», показанной на рис. 7.



Рис. 7

В нижней части окна переменных (рис. 8) расположены четыре закладки:

- Locals — локальные переменные текущей функции;
- Watch #1, Watch #2 — определенные пользователем программные переменные;
- Call Stack — уровень вложенности выполняемой команды.

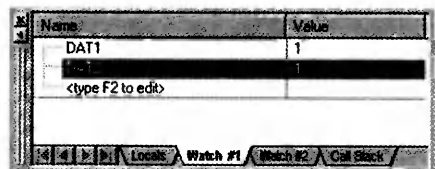


Рис. 8

Для добавления переменных необходимо двойным нажатием мыши выделить строку <type F2 to edit>, ввести имя переменной и нажать Enter. То же самое можно сделать, нажав клавишу F2 на клавиатуре. В качестве переменных также могут быть указаны регистры специальных функций (SFR). Для удаления переменной необходимо щелчком мыши выделить имя переменной и нажать клавишу Delete. Содержимое окна переменных автоматически модифицируется при каждой остановке программы.

Для примера введем указанным выше способом переменные DAT1 и DAT2, результат показан на рис. 8.

В окне памяти на рис. 9 отображается содержимое различных классов и областей памяти.

В строке Address вводится стартовый адрес области памяти, которую необходимо отобразить. Например, введя C:100h и нажав Enter мы уви-

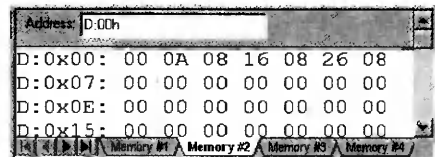


Рис. 9

дим в окне Memory #1 программную область, начиная с адреса 100h, а введя D:00h в окне Memory #2 (рис. 9) мы сможем увидеть область памяти данных микроконтроллера.

Контекстное меню этого окна позволяет выбрать формат представления данных. Щелчок мышью по строке контекстного меню Modify Memory at...(изменить память) вызывает окно Enter Bite(s) at... (ввод байта), в котором производится изменение содержимого ячеек памяти программы и данных.

На рис. 10 показано окно вывода, которое обеспечивает диалог со средой μ VISION-2 с помощью команд.

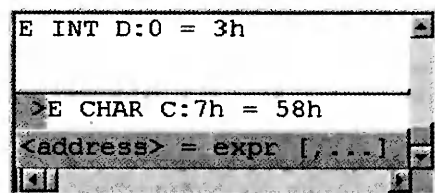


Рис. 10

Более подробно работу с этим окном мы разберем в конце статьи.

Включением режима Workbook Mode через пункт главного меню View→Workbook Mode слегка изменим вид рабочего окна. В результате в нижней части окна появляются дополнительные закладки, которые облегчают работу с файлами.

В рабочем окне при остановленном выполнении программы открыто окно редактора с исходным текстом или при нажатой кнопке Disassembly Window (крайняя слева на рис. 7) показано окно дизассемблера. На рис. 11 в окне дизассемблера выведены одновременно текст исходной программы, код команд и адрес расположения кодов в памяти программ.

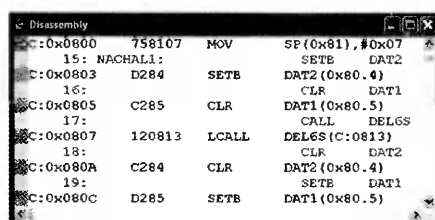


Рис. 11

Работа с окном ассемблера дает дополнительные возможности. Напри-

мер, вызвав через контекстное меню команду Inline Assembly... (встроенное ассемблирование) в открывшемся окне Inline Assembler (рис. 12) в строке Enter New Instruction можно изменить код инструкции.

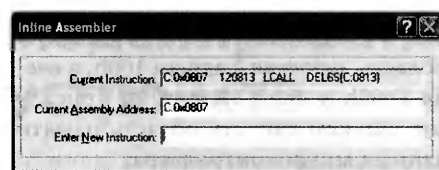


Рис. 12

Это позволяет оперативно исправлять ошибки и производить изменения в программе во время ее отладки.

Среда μ VISION-2 запоминает состояние окна редактора при выходе из него и восстанавливает его в том же виде при повторном обращении к нему. Во время отладки можно использовать большинство возможностей редактора, кроме команд компоновки программы и изменения структуры проекта.

Для отображения состояния внутренних регистров и внешних выводов микроконтроллера среда μ VISION-2 позволяет вывести дополнительные специальные окна, для разных контроллеров они могут отличаться. Вывод осуществляется через пункт главного меню Peripherals (рис. 13).

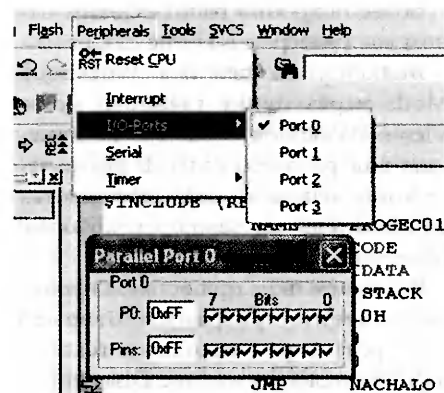


Рис. 13

С помощью команд Peripherals $\rightarrow$ I/O-Ports $\rightarrow$ Port 0 откроем окно Parallel Port 0, оно необходимо нам для наблюдения за изменением состояния выводов в процессе выполнения программы.

Отладка программы обычно разбивается на этапы, без первых двух этапов обойтись невозможно.

Этап 1. Пошаговое выполнение программы.

Этап 2. Расстановка в программе точек останова и пошаговое выполнение программы в тех местах, где возникает несоответствие выполняемого и задуманного алгоритма.

Этап 3. Повторное пошаговое выполнение программы.

Итак, основные подготовительные операции закончены, и мы можем приступить к пошаговому выполнению программы. Для этого щелкнем по кнопке Step into (рис. 14) или нажмем клавишу F11 на клавиатуре, при этом выполняется первая команда нашей программы.

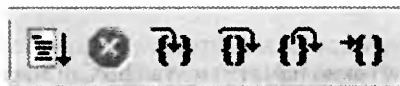


Рис. 14

При выполнении этого шага была выполнена команда JMP NACHALO, в результате чего управление было передано на команду MOV SP, #?STACK-1. Выполняем эту команду. Продолжая пошаговое выполнение программы, мы будем обращать внимание на изменения, происходящие в регистрах и переменных нашей программы.

После выполнения команды SETB DAT2, CLR DAT1 в окне переменных мы увидим изменение значения переменной DAT1 из 1 в 0 (закладка Watch#1). При этом строка переменной DAT1 окрасится в синий цвет, этим выделением обозначаются переменные, которые поменяли свое значение на последнем выполненном шаге. В окне Parallel Port 0 очистится значение пятого бита. Указанная цветовая подсветка имеет место и в других окнах (например, в окне проекта).

На следующем шаге будет выполнена команда CALL DEL10, при этом изменится значение стека, оно увеличится на два и станет равным SP 0x09. В окне переменных (закладка Call Stack) видно, что в данный момент времени выполняется подпрограмма. Выполнив еще три шага, мы окажемся в подпрограмме DEL10 mS. При этом значение стека достигнет величины SP 0x0f, а в окне переменных (закладка Call Stack) мы увидим уровень вложенности нашей подпрограммы.

Дальнейшее выполнение программы потребует большого числа шагов внутри подпрограмм для реализации задержки, равной 1 с. Поэтому мы сократим время выполнения подпрограмм, воспользовавшись кнопкой Step out (вторая справа на рис. 14) или комбинацией клавиш Ctrl+F11 на клавиатуре. Ровно четыре шага потребуются для выхода из подпрограмм, при этом каждый шаг соответствует выполнению текущей подпрограммы.

Итак, мы вернулись в основную программу. Выполнив две очередных

команды CLR DAT2, SETB DAT1, мы увидим, что значения переменных DAT1 и DAT2 в окне переменных (закладка Watch #1) поменялись местами. В окне Parallel Port 0 очистился бит 4 и установился бит 5, т. е. на выходах микроконтроллера поменялись выходные уровни.

Следующая команда повторно обращается к подпрограмме задержки, но вместо пошагового выполнения программы мы можем щелкнуть по кнопке Step over (третья справа на рис. 14) или нажать клавишу F10 на клавиатуре. Подпрограмма будет обойдена.

Выполнив следующую команду JMP NACHAL1, мы вернемся к команде SETB DAT2, которая начинает повторный цикл выполнения наших команд.

После проверки правильности исходной программы можно запустить ее кнопкой Run (первая слева на рис. 14) или клавишей F5 на клавиатуре. В окне Parallel Port 0 мы увидим переключение в противофазе битов 4 и 5 порта 0. Останов программы производится щелчком по кнопке Halt (вторая слева на рис. 14) или клавишей Esc на клавиатуре.

Для удобства отладки программы можно вводить точки останова, они позволяют останавливать программу в необходимых местах для контроля хода ее выполнения и внесения требуемых оперативных изменений.

Ввод точек останова возможен во время редактирования и отладки программы несколькими различными путями:

- щелчком по кнопке Insert/Remove Breakpoint (вставить/удалить точку останова) панели инструментов File (первая слева на рис. 15);



Рис. 15

- щелчком правой кнопкой мыши по строке программы в окне редактора или дизассемблера и выбором в открывшемся контекстном меню команды Insert/Remove Breakpoint;
- через пункт главного меню Debug $\rightarrow$ Insert/Remove Breakpoint;
- через окно вывода.

Следом за кнопкой ввода точек останова (слева направо) расположены следующие кнопки:

- кнопка удаления всех точек останова;
- кнопка, изменяющая состояние точки останова (активное или пассивное);
- кнопка, изменяющая состояние всех точек останова на пассивное.

Для просмотра введенных точек останова удобно использовать окно Breakpoints..., вызываемое через пункт главного меню Debug→Breakpoints.... В этом окне щелчком мыши по окошку, расположенному в начале строки, можно сделать введенную ранее точку останова активной или пассивной и задать дополнительные характеристики точки останова.

Нами рассмотрены основные шаги по отладке программы, но изложение было бы неполным, если бы мы не рассмотрели дополнительные окна отладчика.

Отладчик снабжен двумя окнами последовательного порта, которые обеспечивают последовательный ввод и вывод данных. Их вызов осуществляется командой View→Serial Window #1 (Window #2) или щелчком мышью по кнопке Serial Window #1 панели инструментов «Отладка» (четвертая слева на рис. 7). Это окно позволяет моделировать асинхронный приемопередатчик без аппаратных средств. Последовательный вывод может быть перенаправлен на COM порт компьютера с помощью команды Assign в окне вывода.

В окне выполнения кода показано процентное выполнение кода программы. Эту информацию можно направить в окно вывода с помощью команды Coverage. Поле в левом столбце окна отладчика напротив строк выполненного кода окрашивается в зеленый цвет. Данную особенность можно использовать для определения частей программы, которые к данному моменту не были выполнены. Вызов окна выполнения кода осуществляется командой View Code Coverage Window или щелчком мыши по кнопке с аналогичным названием панели инструментов «Отладка» (третья слева на рис. 7).

В окне карты памяти автоматически отображаются адреса областей памяти, в которых расположена программа. Для каждой области можно определить тип доступа — Read, Write, Execute или их комбинацию. Во время выполнения программы происходит проверка правильности обращения к областям памяти. Вызов окна производится командой Debug Memory Map.

В окне просмотра символов показаны общие символы, локальные символы и регистры специальных функций, используемые в данной программе. В окне Mode можно выбрать тип переменных для отображения, а в окне Mask — тип фильтра. Вызов окна производится командой Debug→Symbol Window.

Мы можем взаимодействовать со средой μVISION-2 посредством ввода команд в окне вывода (рис. 10).

Работу в этом режиме предлагается рассмотреть на следующих примерах: D_D:10h, D:50h — вывод содержимого памяти в указанном сегменте адреса с 10 по 50.

E_char_D:0=_3h — ввод однобайтного целого (3h) в память данных по адресу 0.

В процессе отладки возможен ввод ассемблерных команд в любой сегмент по указанному адресу:

Asm_C:1Ch (нажать кнопку Enter) — установка адреса.

Asm_MOV A,#2 — ввод конкретной команды (команд может быть несколько, они будут вводиться последовательно).

R0=_55h — запись в R0 данного кода.

PC=_C:120h — переход на адрес 120h памяти программ.

В приведенных примерах символ подчеркивания обозначает пробел.

Три кнопки панели инструментов «Отладка», приведенные на рис. 16, позволяют повысить оперативность работы.



Рис. 16

Левая желтая кнопка обеспечивает быстрый возврат к месту выполнения программы, следующая устанавливает режим записи выполненных команд. Правая кнопка позволяет посмотреть предыдущую выполненную команду, данное действие может оказаться удобным для нахождения точки входа в текущую подпрограмму.

Крайняя кнопка слева Reset CPU панели инструментов «Отладка» симулирует аппаратный сброс микроконтроллера в процессе отладки данной программы. Нажатие на эту кнопку возвращает отлаживаемую программу в начало, оставляя при этом содержимое оперативной памяти без изменения.

На панели инструментов «Файл» расположены две кнопки поиска — Find in files (поиск в файлах) и Find (поиск).

Щелчок по кнопке Find (поиск) вызывает появление окна Find (рис. 17).

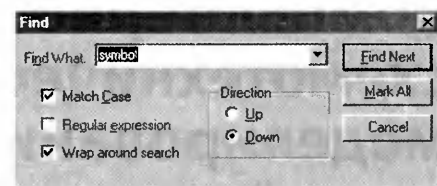


Рис. 17

Режимы поиска могут быть откорректированы установкой следующих флажков.

- Match Case — поиск с учетом регистра;
- Regular expression — поиск целого выражения;
- Wrap around search — циклический поиск.

В разделе Direction задается направление поиска — Up (вверх по тексту) и Down (вниз), Find Next — поиск следующего (предыдущего) совпадения, Mark All — выделять строки.

Описанные приемы отладки программ составляют необходимый и достаточный объем знаний для быстрого перехода из DOS в Windows ориентированную среду проектирования. Даже поверхностный взгляд на предлагаемые средой μVISION-2 возможности говорит об огромных дополнительных ресурсах, обеспечивающих качественную иную стиль работы.

Как и другие Windows ориентированные программы μVISION-2 предлагает пользователю различные варианты путей к конечной цели, которой является рабочая программа. Каждый со временем выберет тот путь, который ему по душе.

Продолжение следует

**Андрей Сошкин,
Андрей Мамонтов,**
г. Москва

Вниманию читателей!

На сайте журнала по адресу <http://www.dian.ru/programs/index> в архиве 2000_1_2_2001_1_Kuznets.zip выложена статья А. Кузнецова «Трансформаторы и дроссели для импульсных источников питания», (Схемотехника, 2000, № 1, 2, 2001, № 1) с учетом уточнений, сделанных автором.

Компьютерное схемотехническое моделирование электронных устройств

В данной статье рассматривается моделирование транзисторных усилителей средствами программ Micro-Cap и Electronics Workbench.

МОДЕЛИРОВАНИЕ ТРАНЗИСТОРНЫХ УСИЛИТЕЛЕЙ ПЕРЕМЕННОГО ТОКА

Электронные усилители являются широко распространенными устройствами, находящими как самостоятельное применение, так применение и в составе многих других устройств. Как правило, источники полезной информации изначально формируют слабые сигналы, недостаточные для выполнения тех или иных функций. В усилителях мощность, напряжение или ток сигнала, подведенные к их входу, увеличиваются на выходе (в нагрузке, подключенной к выходу) за счет управляемого использования источника питания.

Принято классифицировать усилители по различным признакам — усилители тока или напряжения, усилители мощности, а также по частотному диапазону, элементной базе, схемному решению, прикладным функциям и т. д. Первый классификационный признак разделяет все типы усилителей на усилители постоянного тока, усиливающие медленно меняющиеся сигналы, в том числе практически не зависящие от времени, и усилители переменного тока, усиливающие сигналы, меняющиеся во времени.

УСИЛИТЕЛЬНЫЕ КАСКАДЫ НА БИПОЛЯРНЫХ ТРАНЗИСТОРАХ

В основе любого транзисторного усилителя лежит элементарный каскад усиления, выполненный на одном или нескольких транзисторах, не обязательно одинаковых. Такой каскад по отношению к полезному сигналу представляет собой четырехполюсник с входом и выходом, а по отношению к источнику питания в большинстве случаев — двухполюсник.

Существование различных типов транзисторов (биполярных, полевых и МОП-транзисторов), трех возможных схем включения (ОБ, ОЭ и ОК), а также необходимость разделения входных и выходных цепей с источником питания, возможное наличие цепей обрат-

ной связи, цепей управления и защиты приводит к большому разнообразию используемых схем. Еще больше этих вариантов существует для усилителей с двумя и более каскадами усиления.

Наибольшее распространение среди однокаскадных усилителей на биполярных транзисторах получили усилители, собранные по схеме с общим эмиттером.

Схема простейшего усилителя в программе MC показана на рис. 138.

Здесь резистор R1 задает начальное значение (постоянную составляющую тока) базы, а с резистора R2 снимается напряжение на нагрузку R3. Конденса-

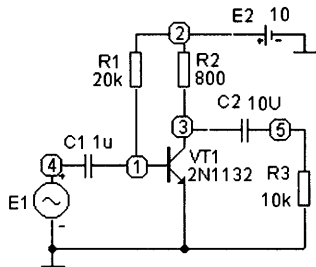


Рис. 138

тор C1 отделяет транзистор и источник постоянного напряжения питания E2 от источника сигнала переменного напряжения E1, а конденсатор C2 аналогично разделяет усилительный каскад и нагрузку. Здесь и далее, если только не оговаривается особо, сигнал принимается гармоническим с частотой 50 Гц и амплитудой 1,4 мВ (действующее значение напряжения приблизительно 1 мВ) и начальной фазой, равной нулю. От выбора сопротивления резисторов R1, R2 и напряжения источника E2 зависит положение рабочей точки и линия нагрузки на выходных характеристиках транзистора и, следовательно-

но, свойства усилителя. Выбор положения рабочей точки очень важен. Если выбрать начальное положение посредине нагрузочной прямой, максимальная амплитуда возможных колебаний будет близка к половине питающего напряжения. Работа усилителя в этом случае происходит без отсечки и называется режимом класса А.

Проведя моделирование переходных процессов (Transient Analysis), получим виртуальные осциллограммы входного и выходного сигналов (рис. 139), из которых видно усиление сигнала и его фазовый сдвиг на этой частоте, обусловленный инвертированием сигнала транзистором, и относительно малой

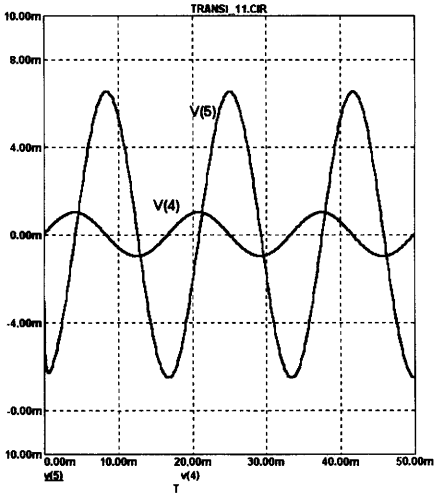


Рис. 139

емкостью разделительных конденсаторов.

Перейдя в режим AC Analysis, получим АЧХ и ФЧХ данного каскада (рис. 140).

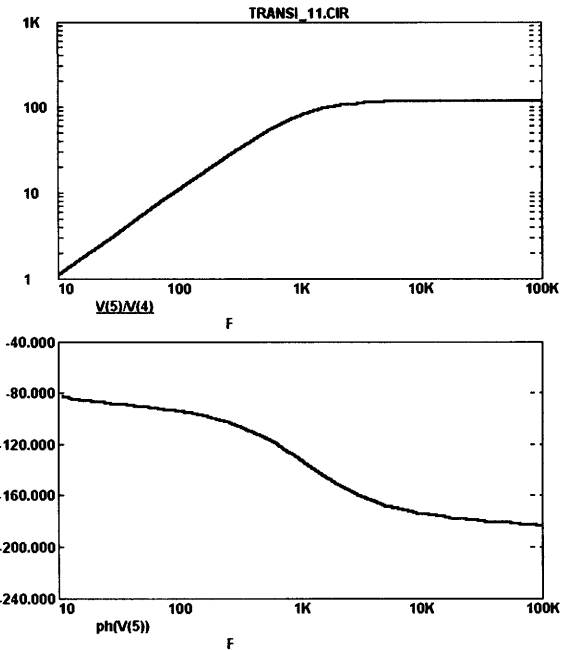


Рис. 140

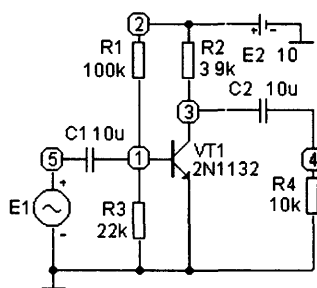


Рис. 141

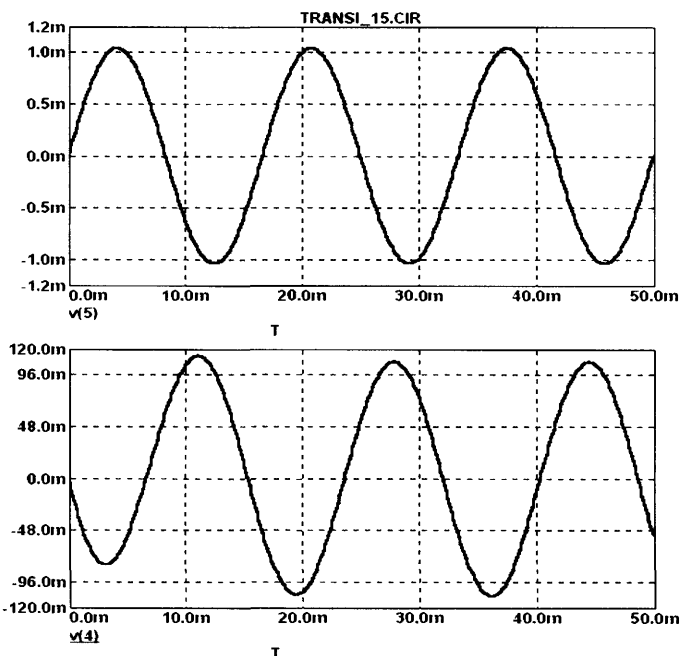


Рис. 142

Другим способом задания режима работы транзистора по постоянному току служит использование делителя напряжения питания на двух резисторах, на рис. 141 это резисторы R1 и R3.

Проведя моделирование аналогично предыдущему, получаем осциллограммы (рис. 142) и частотные характеристики каскада (рис. 143).

В реальных условиях на работу усилителя еще влияет температура, технологический разброс параметров компонентов, наводки и другие дестабилизирующие факторы. В этом отношении каскад по схеме на рис. 141 по сравнению с усилителем по схеме на рис. 138 обладает худшими параметрами по стабильности рабочей точки, в чем нетрудно убедиться, проверив его работу при изменении температуры или напряжения питания.

Свойства каскада могут быть отчасти изменены путем введения обратных связей. Кроме того, целенаправленное введение обратных связей в принципе позволяет управлять режимом работы каскада и даже коренным образом изменить его свойства, например, превратить усилитель в генератор.

Под обратной связью (ОС) подразумевают передачу сигнала от последующих элементов системы к предыдущим (по отношению к направлению распространения полезного сигнала). Для простейшего каскада усиления обратная связь это выходной сигнал (с измененными параметрами), поданный обратно на вход. При положительной обратной связи (ПОС)

для управления (обратная связь по току, напряжению или комбинация). Кроме того, различают ОС по признакам, смысл которых очевиден — внешние и внутренние, полезные и «паразитные».

В усилительном каскаде на рис. 144 в программе EWB резистор R2 играет роль параллельной по напряжению цепи ООС.

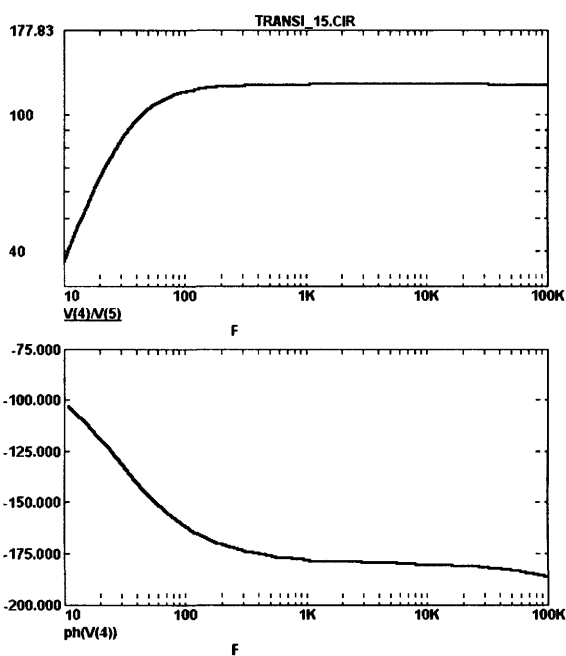


Рис. 143

этот сигнал находится в одной фазе с входным (сдвиг фаз равен нулю) и усиливает его. При отрицательной (ООС) этот сигнал, находясь в противофазе (сдвиг фаз 180°) по отношению к начальному входному сигналу, ослабляет последний. В усилителях используется в основном ООС.

Обратную связь в усилителях классифицируют по знаку (ПОС или ООС), по способу подключения цепи ОС к входу на последовательную и параллельную, а также по виду сигнала, снимаемого

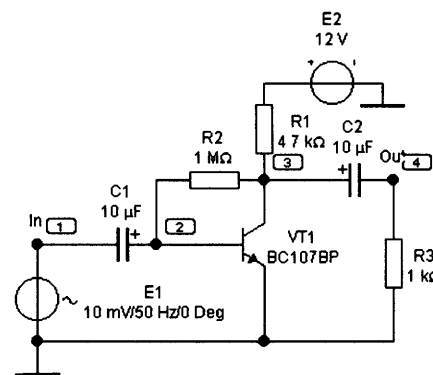


Рис. 144

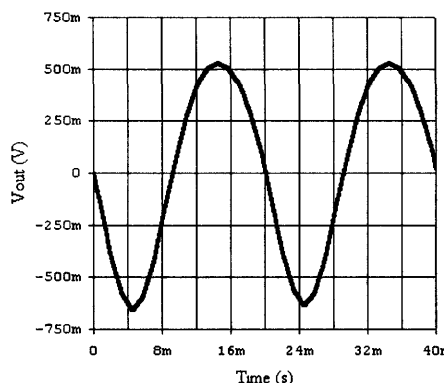
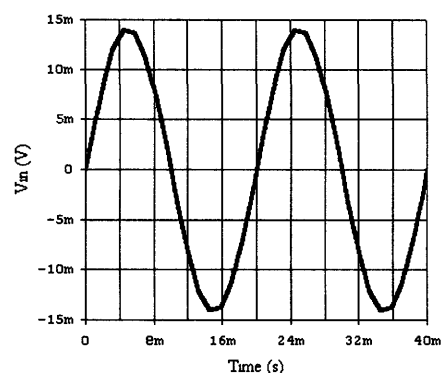


Рис. 145

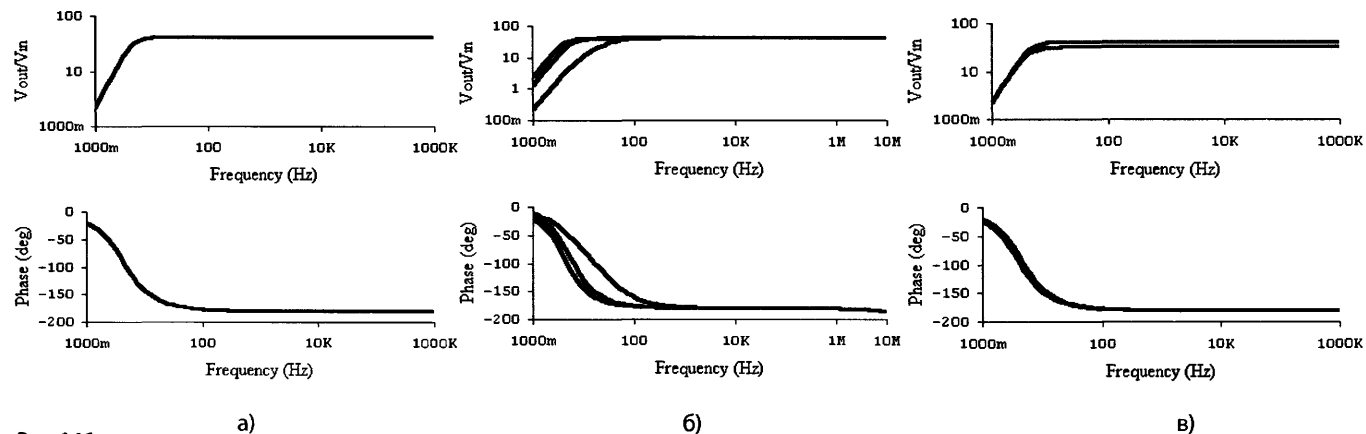


Рис. 146

Результаты стандартного моделирования в режиме исследования переходных процессов (после выполнения команд Analysis > Transient) показаны в виде виртуальных осциллограмм на рис. 145.

Здесь входное напряжение по сравнению с предыдущими схемами увеличено в 10 раз и на осциллограмме выходного сигнала заметно различие в усилении положительной и отрицательной полуволн, связанное с нелинейностью зависимости тока коллектора транзистора от напряжения база-эмиттер.

Частотные характеристики (рис. 146) получены в режиме моделирования на переменном токе (Analysis > AC Frequency).

Толщина линии первой АЧХ (рис. 146, а) связана с тем, что она дополнительно размыта за счет вариации параметра (Parameter sweep). В качестве параметра, изменяемого на порядок, взят обратный ток коллекторного перехода (Saturation current). Следующие характеристики получены при варьировании емкости конденсатора C1 в диапазоне 1...11 мкФ (рис. 146, б) и повышении температуры до 127 °C (рис. 146, в).

Эффективную стабилизацию работы каскада по постоянному току обеспечивает включение в цепь эмиттера резистора R4 при наличии делителя в цепи базы (рис. 147).

В данном случае этот резистор осуществляет последовательную ООС по току.

Выполнив установки анализа в режиме пошагового увеличения температуры Temperature Sweep (рис. 148), получим зависимость напряжения V1 на коллекторе от температуры (рис. 149).

Физически это связано с уменьшением напряжения база-эмиттер транзистора, что увеличивает напряжение на резисто-

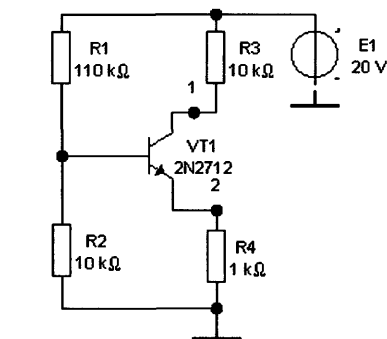


Рис. 147

ре R4 в цепи эмиттера VT1 при почти фиксированном напряжении на его базе. Существенно влияет также увеличение коэффициента передачи тока базы h_{213} при повышении температуры. Поскольку ток коллектора практически равен току эмиттера, то одновременно происходит и рост падения напряжения на резисторе R3.

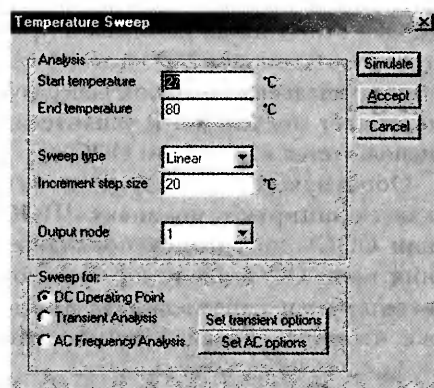


Рис. 148

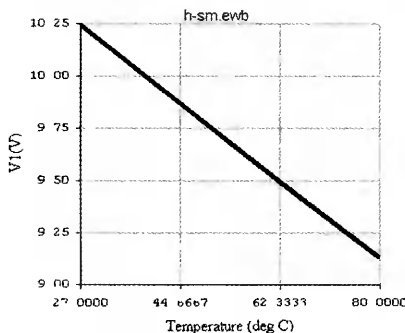


Рис. 149

Характер роста напряжения V2 на резисторе R4 (рис. 150) получен в том же режиме, но при замене выходного узла Output node в окне Temperature Sweep с 1 на 2.

Расположение резисторов R1—R4 на рис. 147 напоминает букву «Н», поэтому подобную схему стабилизации называют также «Н-смещением».

Введение резисторов в цепи ООС приводит к уменьшению усиления каскада. Для устранения этого эффекта используют соответствующее включение конденсаторов. В схеме с Н-смещением эмиттерный резистор просто шунтируют конденсатором, подбирая его емкость в зависимости от требуемой АЧХ.

На рис. 151 эту роль играет конденсатор C3.

Исследование работы каскада иллюстрируется осциллограммами входного V5 и выходного сигнала V6 на рис. 152.

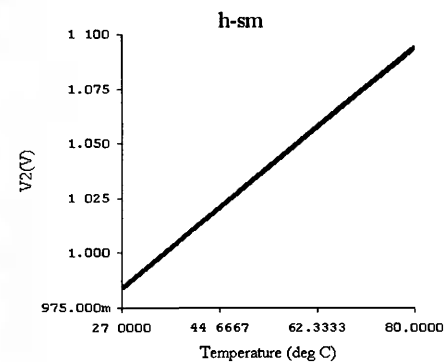


Рис. 150

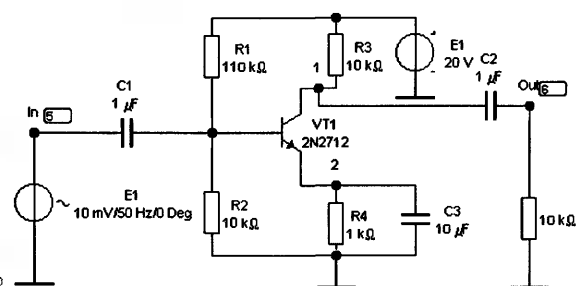


Рис. 151

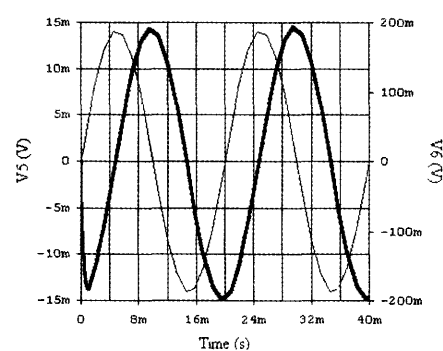


Рис. 152

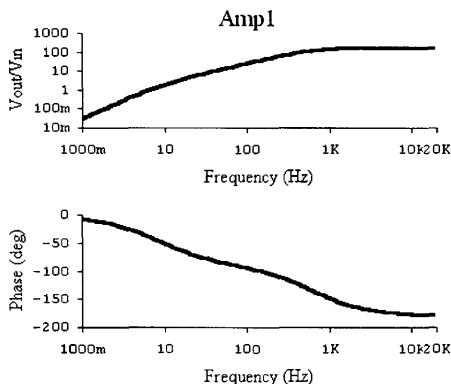


Рис. 153

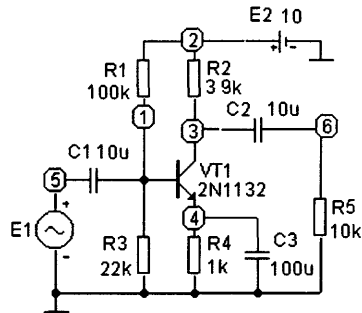


Рис. 154

Здесь сигналу V5 соответствует график, выполненный тонкой линией с масштабом на левой оси ординат, а сигналу V6 — график, выполненный толстой линией с оцифровкой на правой оси. Соответственно АЧХ и ФЧХ каскада приведены на рис. 153.

Схема усилительного каскада в программе МС показана на рис. 154.

Проведем исследование его работы. В режиме Transient Analysis получим виртуальные осциллограммы входного v(5) и выходного v(6) сигналов (рис. 155).

Из этих осциллограмм видно, что сигнал на выходе каскада усилен по отношению к входу, фазовый сдвиг на данной частоте составляет почти 180°.

АЧХ и ФЧХ каскада при различных значениях резистора ОС R4, полученные в режиме АС, показаны на рис. 156.

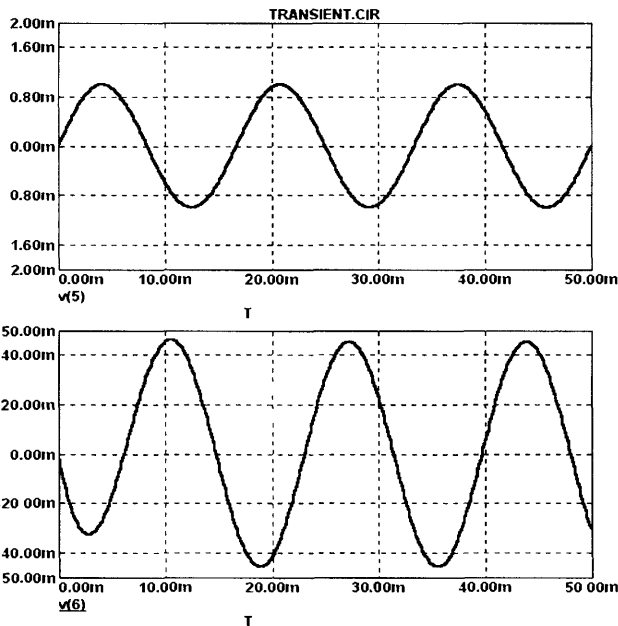


Рис. 155

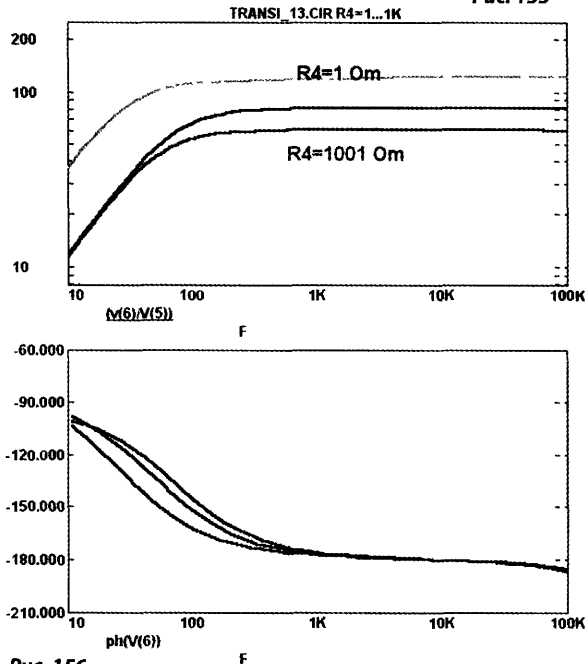


Рис. 156

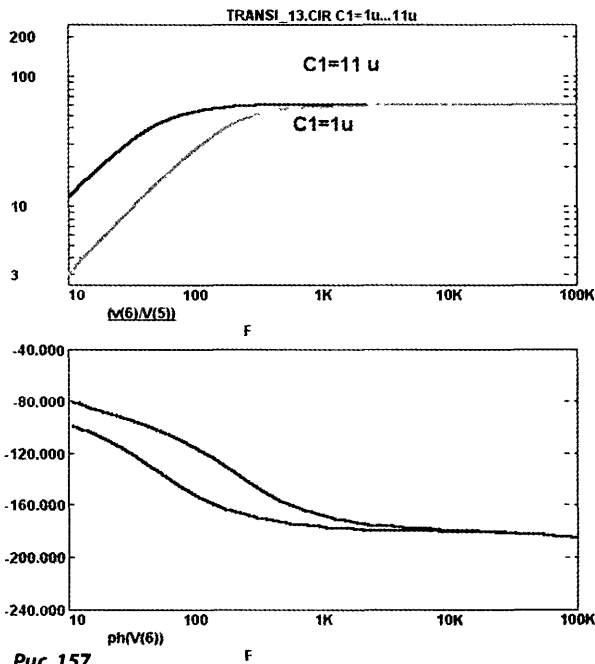


Рис. 157

Из характера АЧХ видно, что каскад имеет практически равномерную характеристику в полосе частот 100 Гц ... 100 кГц.

Влияние изменения емкости входного разделительного конденсатора

C1 на завал АЧХ в области низких частот и фазовые характеристики каскада показаны на графиках на рис. 157, полученных в режиме Sweep-

по этому параметру.

Продолжение следует

Генрих Кардашев,
г. Москва

Пакт Acronis

Письмо в редакцию

Здравствуйте, уважаемая редакция! Прочитал статью А. Фрунзе о восстановлении Windows (2003, № 2) и решил написать письмо. В статье правильно написано, что желательно иметь вторую ОС и с нее загружаться для восстановления первой. У меня у самого стоят две ОС, одна 98SE, другая XP Pro. Единственное, что хотелось бы, это поделиться своим опытом по созданию того, из чего лучше восстанавливать Windows (на мой взгляд, конечно) — из образа диска. Ко мне года два назад попал в руки интересный пакет программ Acronis. После инсталляции и проб нескольких программ я был приятно удивлен его работой, качественной документацией к нему, ну и, конечно, его возможностями.

В эту версию пакета входят восемь программ.

Acronis TrueImage Deluxe позволяет сохранить образ диска в файле архива на жестком диске или на одном из многочисленных сменных носителей, таких как Iomega Zip или Jaz, устройстве записи CD-R(W) или DVD-R(W). Имея точный образ жесткого диска можно в любой момент времени восстановить данные, хранящиеся на диске, в случае системного сбоя, заражения диска вирусами или выхода его из строя.

Acronis RecoveryExpert — программа восстановления разделов жесткого диска, удаленных случайно в результате пользовательской ошибки, программного или аппаратного сбоя, воздействия компьютерного вируса или хакерской атаки. Восстанавливать файлы умеют многие, восстанавливать разделы — только Acronis Recovery Expert.

Acronis PrivacyExpert — программа, обеспечивающая конфиденциальность работы на персональном компьютере, отдельно стоящем или подключенном к локальной или глобальной сети (World Wide Web, WWW).

Acronis PartitionExpert представляет собой менеджер разделов жесткого диска, который позволяет:

- создавать разделы на жестком диске, изменять их размеры и положение без разрушения и потери данных;
- преобразовывать разделы с одной файловой системой в разделы с

другой системой без потери данных на диске;

- копировать (или перемещать) содержимое раздела;
- удалять разделы;
- форматировать, устанавливать метки, присваивать буквы, скрывать и восстанавливать скрытые разделы, устанавливать метки активности раздела и выполнять более редко используемые операции над разделами.

Acronis OS Selector позволяет использовать множество различных операционных систем, установленных на одном компьютере, и переключаться между ними во время загрузки.

Acronis Migrate Easy поможет быстро, легко и надежно установить новый жесткий диск в случае переноса данных со старого диска емкостью до 10 Гбайт на новый диск емкостью 20...30 Гбайт. Понадобится не более нескольких минут на выбор параметров установки жесткого диска и не более одного часа работы самой программы, не требующей внимания пользователя.

Acronis DriveCleanser гарантированно уничтожает данные на выбранных разделах или/и дисках компьютера.

Acronis DiskEditor предоставляет возможность редактировать содержимое секторов жесткого диска.

При фатальных сбоях я пользуюсь Acronis TrueImage Deluxe, на мой взгляд это фантастика, когда за 10...20 мин (в зависимости от скорости компьютера) он восстанавливает физический образ диска, даже не требуя при этом предварительного форматирования. Для этого после инсталляции ОС необходимо почистить реестр, убрать софтовые ошибки и т. д., а затем запускается Acronis TrueImage Deluxe и создает физический образ диска в том месте, где его укажет пользователь.

Хранить его, конечно, надо на отдельном носителе (винчестере, CD и так далее). Можно записать образ, например, диска C на диск D, и в случае серьезного сбоя, перезагрузившись с другой ОС, восстановить на нужный диск. Даже если ОС одна и есть копия, сохраненная не на рухнувшем диске, можно, загрузив оболочку Acronis TrueImage Deluxe с четырех дискет

(при установке программа предлагает сделать свою загрузочную копию) тоже восстановить диск.

Согласитесь, это проще и быстрее, чем переустанавливать всю Windows заново, да и в придачу массу других программ. Ведь вспомнить все, что нужно ставить довольно трудно, особенно когда голова занята совсем не мыслями о восстановлении этих злосчастных Windows. Многих отпугивает слово “физическая копия”, а зря, программисты, создавшие пакет Acronis, не напрасно едят свой хлеб.

Можно, конечно, возразить, что эта проблема касается старых Windows, а вот на XP все гораздо надежнее. Надежнее-то надежнее, но уж сколько раз выходило (и не только у меня), что восстановить из образа гораздо проще, чем часами копаться во внутренностях XP в надежде все-таки восстановить ее работу стандартными средствами.

Тем более, что когда есть конкретная работа, в которой изучение возможностей криво написанных Windows стоит далеко не на первом месте, это совершенно меняет дело. А, как известно, все плохое происходит всегда не вовремя.

Также, когда мне понадобилось выделить на винчестере дополнительный раздел, мне здорово помог Acronis PartitionExpert, переразбив жесткий диск заново, без потери данных на остальных дисках. В общем, попробуйте, конечно, создав копию на всякий случай.

Я давно уже не пользуюсь никакими консолями восстановления и прочими средствами, предусмотренными в XP. Интересно, а сами разработчики Микрософт пользовались хоть раз тем, что создали для восстановления своих Windows. Мне кажется, что нет.

Адрес сайта фирмы-распространителя пакета Acronis <http://www.acronis.ru>.

Кстати, стоимость программ для обычного (не корпоративного) пользователя очень даже невысокая, каждая программа стоит что-то около \$4, а при покупке нескольких предусмотрена скидка. Попробуйте, не пожалеете.

Буду признателен, если моя информация поможет кому-нибудь спасти свои время и силы при очередной полной переустановке Windows и массы программ вместе с ней заново.

С уважением,
Евгений

Устройство подачи звонков по расписанию

Устройство предназначено для автоматической подачи звонков в учебных заведениях. Оно разработано на микроконтроллере PIC16F628 и имеет возможность подачи звонков для 32-х уроков, индикацию текущего времени и времени очередного звонка. Устройство имеет простое управление и возможность просмотра времени подачи всех установленных звонков на ЖК дисплее.

Все установки времени сохраняются в энергонезависимой памяти, а резервное питание сохраняет текущее время и установки коррекции часов и времени звонка в случае отключения сетевого напряжения. Время звучания звонка можно задать в пределах 0...9 с. Суточная коррекция хода часов может быть установлена в пределах 0...±9 мин с дискретностью 1 с.

Устройство имеет следующие режимы индикации:

- текущего времени и времени следующего звонка;
- номера урока и времени начала и окончания урока;
- установки текущего времени и времени звучания звонка;
- коррекции хода часов;
- установки времени начала и окончания урока.

Алгоритм работы программы микроконтроллера показан на рис. 1.

После подачи напряжения питания выполняется инициализация регистров

микроконтроллера и предварительные установки рабочих регистров. Из памяти считываются ранее записанные значения времени начала и окончания первого урока. Далее инициализируются регистры ЖК дисплея.

Через каждые 0,25 с проверяется состояние кнопок управления. Все временные флаги устанавливаются в подпрограмме отсчета времени при прерывании. При частоте кварцевого резонатора 4,096 МГц, машинном цикле, равном 4, коэффициентам деления предварительного делителя 4 и таймера 256 прерывание по переополнению таймера TMR0 будет происходить через 1 мс ($4096000/(4 \times 4 \times 256) = 1000$ Гц).

В начале каждой минуты выполняется сравнение текущего времени с временем включения звонка. Если они совпадают, включается звонок на запрограммированное ранее время. Далее проверяется флаг нажатой кнопки. Поскольку кнопки включены параллельно шине управления ЖК дисплеем, возможна ситуация, когда индикация примет хаотический вид. Для исключения такой ситуации вывод на индикацию выполняется только при отпущенных кнопках. В данном случае кнопки можно было бы подключить к свободным выводам микроконтроллера, но для преемственности программы они подключены параллельно шине управления индикатором.

После передачи значений регистров индикации в ЖК дисплей программа повторяет цикл с ожидания включения флага 0,25 с.

Алгоритм работы подпрограммы определения состояния клавиатуры показан на рис. 2.

Если любая кнопка нажата, устанавливается флаг кнопки и при последующих прохождении подпрограммы проверяется ее отпускание. Если взведен флаг кнопки установки или разряда, инициализируется выбранный разряд или инкрементируется регистр курсора. Поскольку у курсора для различных режимов работы разные начальные установки, они выполняются при смене режима индикации. По значению курсора выбирается разряд для установки.

Если кнопки установки и разряда не нажаты, проверяется состояние кнопки «Режим». Если кнопка «Режим» нажата, инкрементируется регистр режима и по его номеру выбирается необходимый режим индикации. При любом состоянии всех кнопок производятся выбор режима индикации и

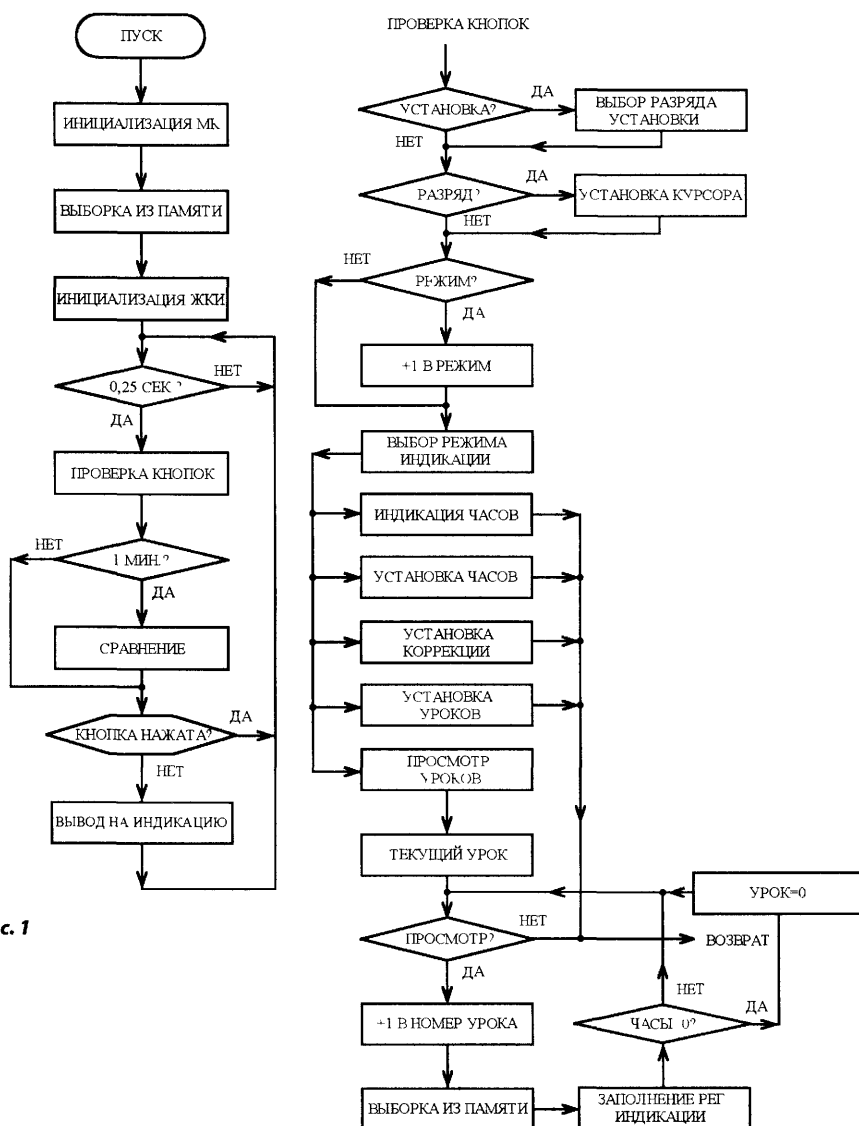


Рис. 1

перезапись в регистры индикации выбранного режима новых или прежних значений. После заполнения регистров индикации программа заканчивает выполнение подпрограммы.

Это справедливо для всех режимов, кроме режима просмотра уроков. В этом режиме сначала индицируется время текущего урока и проверяется состояние кнопки просмотра. Если кнопка не нажата, то выполнение подпрограммы заканчивается. Если кнопка просмотра нажата, инкрементируется регистр просмотра урока и по его номеру из памяти выбирается значение времени урока.

Для того, чтобы номер урока соответствовал адресу записи/считывания, память распределена следующим образом. По адресам 0...31 записываются часы начала урока, 32...63 — минуты начала урока, 64...95 — часы окончания урока, 96...127 — минуты окончания урока. Таким образом, начальный адрес хранения времени урока будет соответствовать номеру урока, уменьшенному на единицу, а адрес каждого последующего значения будет отличаться от предыдущего на 32. Запись в память выполняется в двоичном коде, поэтому после считывания значение перекодируется в двоично-десятичный код. Двоично-десятичные значения переписываются в регистры индикации.

Двоичные значения для записи в память формируются при установке каждого разряда времени урока. После установки любого разряда выполняется запись в память нового значения времени.

Если считанное значение часов начала урока равно нулю, регистр номера урока обнуляется, а последующая выборка из памяти начнется с первого урока. Этим можно пользоваться для отсеечения уроков временно отсутствующей второй смены.

Регистр номера урока для режима просмотра и установки уроков один, поэтому для установки времени урока необходимо сначала найти нужный номер урока при просмотре, затем перейти в режим установки. При установке номер урока не изменяется. Такое усложнение установки принято для исключения случайного изменения времени урока.

Алгоритм работы подпрограммы сравнения показан на рис. 3.

Сравнение выполняется каждую минуту, поэтому для выполнения следующего сравнения сбрасывается флаг одной минуты и проверяется

флаг установки. Сравнение происходит только в режиме индикации часов и времени очередного звонка. Поэтому, если идет установка, подпрограмма сравнения не выполняется.

Если время начала урока предыдущей выборки из памяти равно нулю, значит последний урок закончился и регистр номера урока обнуляется. Таким образом, нулевое значение следующего звонка на индикаторе будет высвечиваться одну минуту. При очередном сравнении номер урока обнулится, и из памяти будет считано время первого урока.

Если часы начала следующего урока не равны нулю, сравнивается текущее время с временем очередного звонка. Если они не равны, подпрограмма не выполняется. При совпадении текущего времени с временем звонка проверяется флаг начала урока. Если было начало урока, регистры индикации заполняются значением времени окончания урока, а флаг начала урока

устанавливается в 1 для дальнейшего сравнения. Включается выход звонка и ожидается окончание времени звучания звонка. После обнуления счетчика секунд звучания звонка звонок выключается.

Поскольку во время звонка программа закликивается на ожидании обнуления счетчика, в это время не выполняется вывод на индикацию, а секундная черта не мигает. После окончания звонка на индикаторе установится время следующего звонка.

Если включен флаг начала урока, инкрементируется регистр номера урока и выполняется считывание из памяти времени данного урока. Считанные значения времени записываются в регистры хранения, а время начала урока переписывается в регистры индикации. При этом флаг начала урока обнуляется, т. е. время начала урока уже записано в регистры индикации. Поскольку не при каждом считывании из памяти необходи-

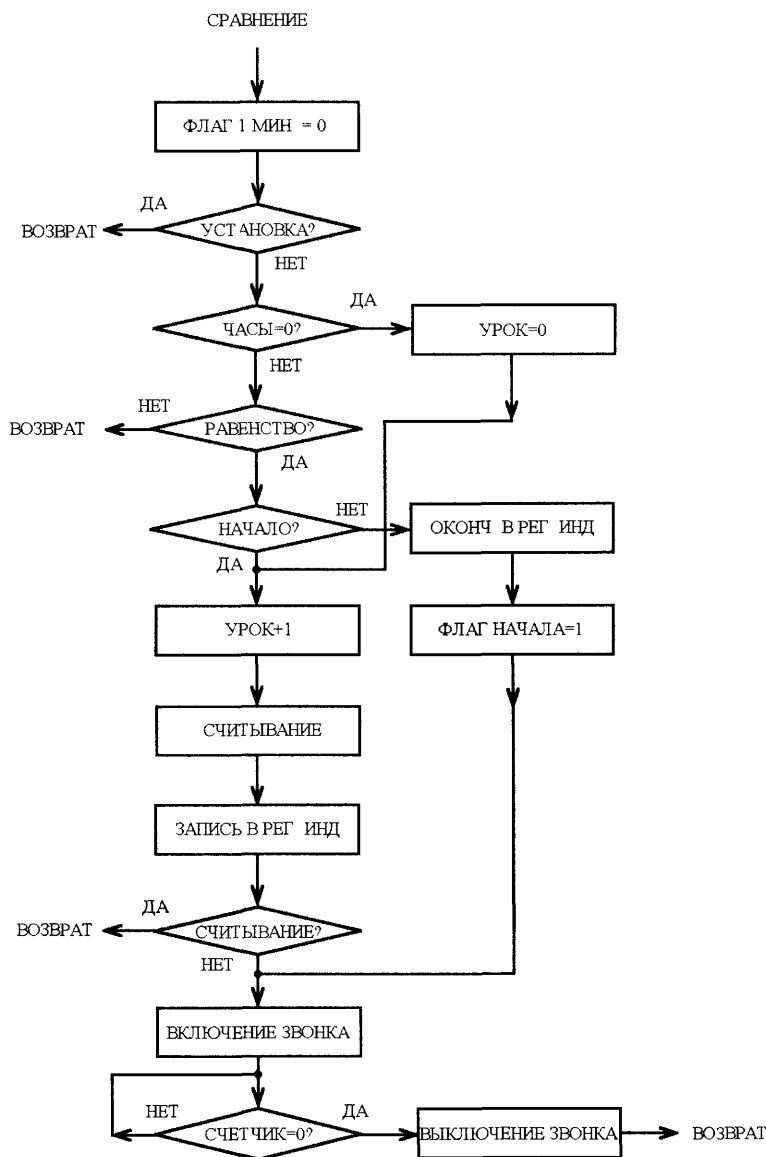


Рис. 3

мо включение звонка, то проверяется флаг считывания. Этот флаг устанавливается при считывании памяти во время включения микроконтроллера и после окончания времени последнего урока. Если флаг считывания выключен, то включается звонок, обрабатывается время звучания звонка, выключается звонок, подпрограмма завершается.

Схема устройства показана на рис. 4. Микроконтроллер DD1 не имеет вывода обнуления, поскольку установлен программный сброс (MCLR) и выполняется он только при подаче напряжения питания. Печатная плата устройства и расположение элементов на ней показаны на рис. 5 и 6 соответственно.

Плата выполнена из односторонне фольгированного стеклотекстолита с одной перемычкой, показанной на рис. 5 штриховой линией.

Устройство смонтировано в унифицированном корпусе ZVI (рис. 7).

Кнопка «Разряд» SB3 и кнопка «Установка» SB2 утоплены в корпус для предотвращения их случайного нажатия. Печатную плату устанавливают в корпус перпендикулярно лицевой стороне, для чего в верхней половине корпуса приклеивают направляющие для платы. ЖК дисплей устанавливают в направляющие передней панели, а в самой панели выбирают отверстие под индикатор и приклеивают плату в двух местах герметиком к индикатору.

В корпусе устанавливают пенал с резервным источником питания. В зависимости от возможной продолжительности отключения напряжения сети выбирают тип элементов питания. Потребляемый от них ток при напряжении 4,5 В равен 1,2 мА. При разрядке элементов резервного питания ниже 4 В индикатор перестает работать, но микроконтроллер нормально функционирует до напряжения 3 В. Для питания устройства от сети используют любой адаптер, имеющий на выходе нестабилизированное постоянное напряжение в диапазоне 9...20 В. Основное требование к адаптеру — отсутствие перегрева при круглосуточной работе.

Налаживание устройства начинают с проверки напряжения питания на выводах панельки под микросхему без микроконтроллера. Вставляют запрограммированный микроконтроллер и подбором резистора R7 добиваются хорошей контрастности

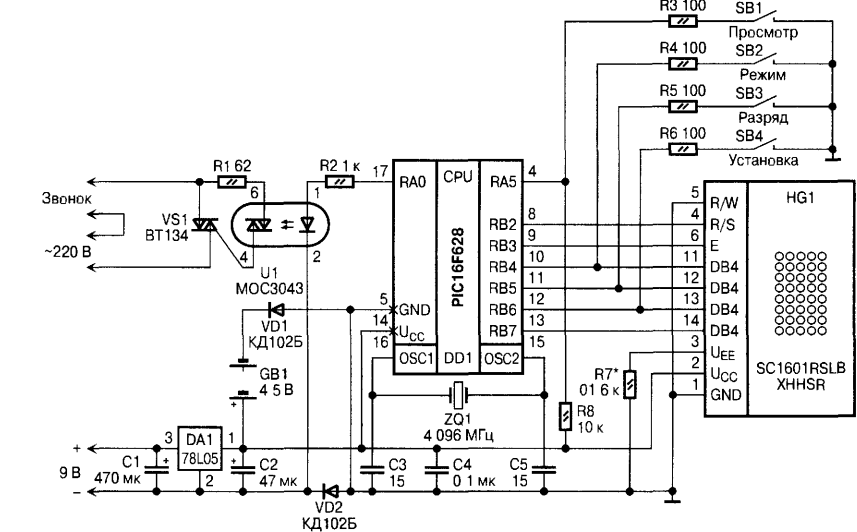


Рис. 4

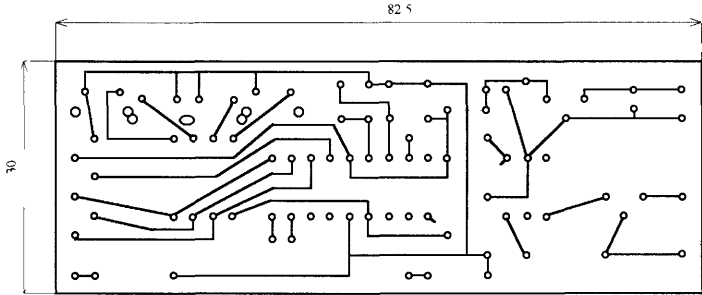


Рис. 5

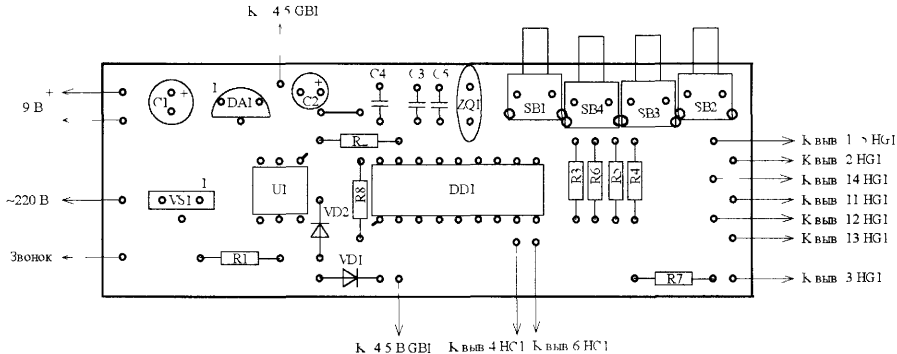


Рис. 6

изображения на дисплее при отключенном сетевом питании. Если применен индикатор, отличный от указанного на схеме, номинал резистора может изменяться в широких пределах. В режиме наладки устройства вместо звонка можно подключить настольную лампу.

После подачи напряжения питания на индикаторе появятся цифры, как показано на рис. 8.

Левые четыре цифры показывают время включения звонка в часах и минутах, правые — текущее время, они разделены мигающей секундной чертой. После нажатия кнопки «Режим» на индикаторе последовательно будут появляться изображения, аналогичные показанным на рис. 9—11.

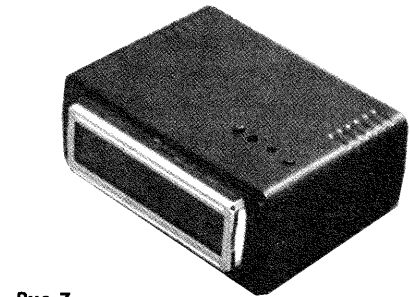


Рис. 7

На рис. 9 показан вид индикации в режиме просмотра.

Индикация означает, что второй урок начинается в 8 час 55 мин и заканчивается в 9 час 40 мин. Нажимая кнопку «Просмотр» можно проконтролировать все уроки. Программно установлено расписание шести уро-

ков. Если необходима переустановка значений времени урока, нужно найти при просмотре необходимый урок и переключить режим нажатием кнопки «Режим» до появления на дисплее аналогичного рис. 9 изображения, но с мигающим курсором.

Номер урока и время, которое индигировалось в режиме просмотра, будет таким же и в режиме установки. Нажимая стержнем шариковой ручки на кнопку «Разряд» (по мигающему курсору под цифрой) выбирают необходимый разряд установки, а нажатием кнопки «Установка» — необходимую цифру в данном разряде. Установка и перемещение курсора происходят после отпускания кнопки. Время удержания кнопки в нажатом состоянии должно быть не менее 0,5 с. Если необходимо установить время седьмого урока, то в режиме просмотра находят седьмой урок с нулевыми показаниями времени и переходят в режим установки. После установки времени урока переходят в режим просмотра и проверяют правильность записи в память. Необходимо следить, чтобы время окончания урока было больше, чем время его начала.

Вряд ли можно поместить 32 урока с переменами в сутки, но если вам требуется давать двойные звонки, то такое большое число установок может пригодиться. Необходимо помнить, что минимальное время между звонками равно одной минуте. При установке время между двойными звонками будет учитываться как урок. Обычно двойные звонки дают перед началом урока, чтобы учащиеся могли успеть занять свои места. Если это так, то фактический номер урока будет равен номеру, который показан на дисплее, деленному на два.

На рис. 10 показана индикация в режиме установки текущего времени и времени звучания звонка (курсор под разрядом десятков часов).

Время звучания звонка устанавливается в секундах. Если время звучания звонка установить равным нулю, звонок не будет звучать. Отключение звонка может потребоваться на время каникул. При инициализации установлено время звучания звонка, равное 1 с. Если в режиме установки часов нажать кнопку «Просмотр», во время отпускания кнопки секунды обнулятся. Если при этом минуты равны 59, обнулятся и минуты, а в значение часов прибавится единица. Данный режим необходим для установки часов по сигналам точного времени.

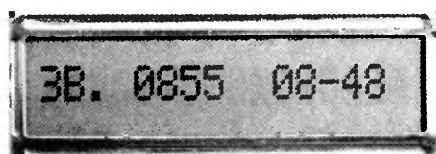


Рис. 8

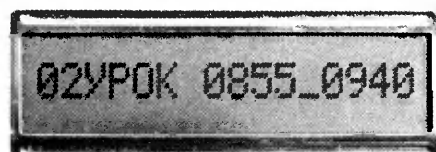


Рис. 9

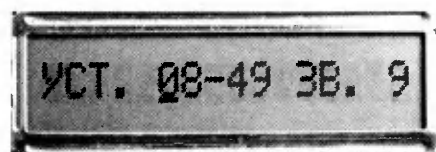


Рис. 10

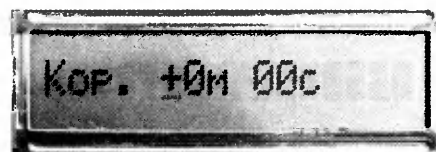


Рис. 11

На рис. 11 показана индикация в режиме установки коррекции, курсор под знаком плюс.

В переписке с читателями журнала выяснилось, что многие испытывают затруднение в приобретении кварцевого резонатора на частоту 4,096 МГц. Поскольку подача звонков «вручную» в настоящее время выполняется в местности, удаленной от крупных городов, то было принято решение сделать суточную коррекцию до ± 9 мин 59 с. Это дает возможность использовать кварцевые резонаторы в интервале частот 4,068...4,124 МГц. Коррекция производится в 00 час 10 мин. Знак коррекции отрицательный, если часы спешат, и наоборот.

Все установки в устройстве желательно выполнять после окончания последнего урока или перед началом первого, когда на индикаторе будет установлено время начала первого урока.

ВНИМАНИЕ! После всех установок необходимо включить режим индикации текущего времени и времени включения звонка. В других режимах звонок включаться не будет.

Файл для программирования микроконтроллера shkola.hex можно найти на сайте журнала по адресу www.dian.ru/programs.

Николай Заец,
п. Вейделевка Белгородской обл.

Известна аналогичная конструкция ИК датчика на микросхемах ТТЛ [1], где в качестве приемника и передатчика используются ИК приемник и светодиод, предназначенные для пультов ДУ бытовой радиоаппаратуры. В предлагаемой конструкции для формирования сигнала передатчика и обработки принятой приемником информации используются микроконтроллеры PIC12F629, что позволяет уменьшить габариты датчика и энергопотребление. Применение микроконтроллеров позволяет избежать проявления «краевого» эффекта, описанного в [1], а также повысить надежность работы двух расположенных рядом датчиков, поскольку датчик реагирует только на «свой» сигнал. Предлагаемое устройство разрабатывалось в качестве бесконтактного датчика наличия заготовки на транспортёрной ленте.

ИК датчик состоит из двух частей — приемника и передатчика. В качестве приемника ИК излучения применена специализированная микросхема TSOP1736. Особенность этой микросхемы в том, что для ее нормальной работы необходимо импульсное ИК излучение с заполнением 36 кГц, на непрерывный ИК сигнал с такой частотой она не реагирует.

Схема ИК передатчика показана на рис. 1. При включении питания он формирует меандр частотой 1 кГц с заполнением 36 кГц. К точности тактовой частоты не предъявляется повышенных требований, поэтому используется внутренний RC генератор частотой 4 МГц, сброс — внутренний. Питание осуществляется от источника постоянного напряжения 24 В, для питания микроконтроллера DD1 использован интегральный стабилизатор DA1. Конденсаторы C1—C3 предназначены для подавления пульсаций питающего напряжения. Диод VD1 служит для защиты от неправильного включения питающего напряжения. Для индикации нормальной работы передатчика используется светодиод HL1. ИК светодиод HL2, чтобы не нагружать стабилизатор, подключен к источнику +24 В и управляется транзисторным ключом VT1. Резистор R3 подбирается в зависимости от типа примененного светодиода так, чтобы ток не превышал максимально допустимого импульсного значения. Например, для AL107Б максимальный ток составляет 100 мА, при этом сопротивление резистора R3 должно быть не менее 240 Ом, также необо-

Предлагаемая конструкция предназначена для контроля положения частей и механизмов станков, подсчета числа пересечений барьера или в качестве барьерного ИК датчика охранной сигнализации.



Схема ИК приемника показана на рис. 2. В микроконтроллере приемника, как и в передатчике, используется внутренний RC генератор частотой 4 МГц, сброс — внутренний. Элементы C1, C2, R1 составляют фильтр для питания ИК приемника DA1. Сигнал с выхода DA1 поступает на вход GP4 микроконтроллера DD1, микроконтроллер измеряет период импульсов и если он составляет $1 \pm 0,1$ мс (частота 1 кГц $\pm 10\%$), то импульсы считаются «своими». При поступлении пяти «своих» импульсов на выходе GP2 микроконтроллера установится лог. 0. Если период отличен от указанной величины, либо временной промежуток между импульсами не равен $0,5 \pm 0,05$ мс, импульс считается «чужим» и на выходе GP2 микроконтроллера установится лог. 1. Также на выходе GP2 микроконтроллера лог. 1 установится при отсутствии импульсов. Для индикации работы приемника используется светодиод HL1. Управление внешней цепью осуществляется транзисторным ключом, состоящим из транзисторов VT1, VT2, ток нагрузки до 1 А.



Все резисторы и конденсаторы, кроме оксидных, для поверхностного монтажа, типоразмер 1206, оксидные конденсаторы цепи 24 В — К50-35 или аналогичные, остальные оксидные — танталовые для поверхностного монтажа. Резистор R3 передатчика состоит из десяти включенных параллельно резисторов сопротивлением 2,7 кОм. Транзисторы BC807 и BC817 (в корпусе SOT23) можно заменить на KT3107, KT3102 соответственно, но при этом ток нагрузки приемника уменьшится до 100 мА. Стабилизатор 7805 в корпусе TO220 без теплоотвода, при этом его перегрев составляет 20...25 °С. Светодиод АЛ107Б можно заменить на любой светодиод ИК диапазона, например АЛ156, АЛ172. В качестве индикаторных светодиодов использованы КИПД-66. Интегральный приемник TSOP1736 можно заменить на любую аналогичную микросхему с частотой модулирующих импульсов 36 кГц. Микроконтроллеры применены в корпусах DIP, т. к. аналогичные микросхемы в корпусах для поверхностного монтажа не дают реального выигрыша в габаритах.

Тексты программ приемника Reciver.asm и передатчика Transm.asm на ассемблере и файлы прошивки контроллеров Reciver.hex и Transm.hex находится на сайте журнала по адресу www.dian.ru/programs.

Юрий Килиба,
г. Великий Новгород

На сайте журнала «Схемотехника» по адресу <http://www.dian.ru/programs/index.html> выложены программы и таблицы прошивки ПЗУ к статье «Частотомер на микроконтроллере АТ89С51» (№ 2, 2002), чертежи печатных плат к статьям «Автомобильный тахометр на К1003ПП1» (2001, № 10), «Сотовый телефон в канале охранной сигнализации» (2002, № 11) и другим статьям из последних номеров, а также содержание (перечень статей в формате Excel) всех вышедших номеров журналов «Схемотехника».

Гонорары выплачиваются авторам, проживающим на территории СНГ.

Лампа-часы-термометр на микроконтроллере Atmel ATmega16L-8PI

В данной статье описано простое multifunctional устройство на микроконтроллере.

Основу устройства составляет микроконтроллер фирмы Atmel ATmega16L-8PI. Микроконтроллер был выбран исходя из следующих его достоинств:

- наличие восьми десятиразрядных АЦП;
- значительный объем Flash-памяти;
- наличие таймера реального времени.

Такое число АЦП позволяет использовать недорогие датчики температуры TMP35—TMP37. В данной разработке используются более дорогие, но и более точные датчики AD592 из-за наличия у них токового выхода, что позволяет обойтись всего двумя соединительными проводами и дает возможность использовать длинную линию связи. Ниже будут приведены схемные решения, позволяющие подключить недорогие датчики. Большой объем Flash-памяти необходим для реализации системы меню и относительно большого числа мелодий.

Основные возможности устройства:

- отсчет времени с индикацией часов, минут, дня недели, числа, месяца, года;
- измерение и отображение температуры от трех независимых датчиков;
- наличие трех будильников с возможностью установки на каждый день недели;
- калибровка хода часов и узлов измерения температуры;
- автоматическое определение високосного года и переход с зимнего времени на летнее и наоборот;
- управление лампой накаливания (плавное зажигание, регулирование мощности, включение по будильникам, выключение или включение по таймеру);
- управление мощной внешней нагрузкой (например, электрообогревателем): регулирование мощности, управление по таймеру, стабилизация температуры комнаты (по внешнему, встроенному датчикам или по среднему значению);
- энергонезависимость часов (в качестве резервного источника

питания используется литиевый элемент питания);

- управление осуществляется пятью кнопками;
- удобное и гибкое меню.

Оригинальным является узел опрерделения перехода через нуль фазы сетевого питающего напряжения без применения оптоизоляторов, тем не менее низковольтная часть устройства гальванически развязана от сетевого напряжения.

Устройство выполнено в виде двух блоков, один из которых (базовый) представляет собой бра с элементами управления и индикации, второй блок — корпус с розетками и установленным в нем симистором на радиаторе для управления электрообогревателем.

Принципиальная схема базовой конструкции приведена на рис. 1.

На элементах R1, R2, VD2—VD5 собран датчик перехода сетевого напряжения через нуль. Переменное напряжение с вторичной обмотки трансформатора питания низковольтной части ограничивается параметрическими стабилизаторами R1VD5 и R2VD4 относительно уровня общего провода и включенными встречно-параллельно диодами между входами встроенного во все микроконтроллеры AVR аналогового компаратора. В момент перехода сетевого напряжения через нулевое значение происходит переключение компаратора, что отслеживается программно, вызывая прерывание. По сравнению с датчиками сети, описанными, например, в [1, 2], данный способ позволяет более точно отследить момент перехода через нуль сетевого напряжения, построен на более дешевых и широко доступных элементах. Недостатком является необходимость наличия микроконтроллера, имеющего в своем составе аналоговый компаратор с двумя информационными входами.

Вывод информации осуществляется с помощью дисплея на жидких кристаллах, имеющего 16 символов в две строки со встроенным контроллером.

Связь с микроконтроллером четырехпроводная, плюс два управляющих сигнала: сигнал данные/команда RS и сигнал строба E. Подстроечный резистор R15 служит для регулировки контраста отображения индикации. Подсветка управляется микроконтроллером через транзистор VT2. Чтение данных из дисплея не предусмотрено, поэтому вывод RW индикатора замкнут на общий провод (запись данных). Поскольку при лог. 0 на шине E шина данных и управляющий вывод находятся в высокоимпедансном состоянии, эти выводы используются для ввода данных в микроконтроллер с пяти кнопок. Для предотвращения искажения выводимой на индикатор информации при одновременном нажатии на кнопки последовательно с ними включены развязывающие диоды VD12—VD16.

На элементах VD6, C1, R6, R7, VT1 и встроенном во входной порт микроконтроллера резисторе выполнен датчик наличия сетевого напряжения питания. Когда сетевое напряжение есть, транзистор VT1 открыт, на выводе прерывания INT0 микроконтроллера — лог. 0. При отключении от сети на выводе INT0 будет лог. 1.

Узел управления лампой накаливания выполнен на элементах U1, VD11, R12, R18, C9, RU2. Способ управления выбран фазоимпульсный, поскольку он обеспечивает плавное изменение яркости лампы. Данный способ регулирования мощности имеет существенный недостаток — создание помех, которые можно услышать, если включить радиоприемник в диапазоне длинных или средних волн. Существуют способы снизить уровень помех, например, при помощи LC-фильтров. В данной конструкции LC-фильтры не используются по конструктивным причинам, но при повторении конструкции рекомендуется установить стандартный дроссель и высоковольтные конденсаторы, бумажные или металлопленочные, в цепь питания лампы. К сожалению, из-за большого тока во втором канале управления затруднительно установить дроссель из-за его значительных габаритов. Впрочем, если мощность силовой нагрузки не регулировать, а просто коммутировать, дроссель может и не понадобиться.

Микроконтроллер с помощью встроенного 16-разрядного таймера с блоком сравнения управляет силовым оптосимистором, нагрузкой которого является лампа накаливания мощностью не более 100 Вт (в авторском

варианте используется лампа фирмы Philips мощностью 60 Вт с отражателем и матовой поверхностью). Второй канал 16-разрядного таймера используется для управления внешней нагрузкой. Принцип регулирования основан на отсчете выдержки времени от точки перехода сетевого напряжения через нуль до включения силовых симисторов. В момент перехода через нуль программно устанавливаются в единицу выходы микроконтроллера, управляющие нагрузкой, тем самым отключая симисторы. После отсчета времени, соответствующего необходимой мощности нагрузки, эти выходы программно сбрасываются, открывая силовые симисторы. Данный способ позволяет регулировать мощность нагрузки в широких пределах. Для цепи управления лампой накаливания реализована плавная подача напряжения питания при включении, что способствует значительному продлению срока службы лампы.

Блок питания собран по классической трансформаторной схеме с интегральным стабилизатором DA2 на 5 В. Резервное питание поступает от литиевого элемента GB1 напряжением 3 В. Для нормальной работы

необходимо выполнение следующих условий: микроконтроллер должен быть рассчитан на напряжение питания 2,7...5,5 В, этому требованию отвечают микроконтроллеры, имеющие в своем обозначении литеру L; диод VD10 должен иметь минимальное падение напряжения (диод Шоттки).

Резисторы R13, R14 образуют делитель для измерения напряжения литиевого элемента питания с помощью АЦП. Литиевый элемент CR2032 имеет емкость 200...250 мА·ч. Ток потребления устройства от элемента питания составляет 200...300 мкА в режиме микропотребления (с отсчетом времени) и 2 мкА при наличии сетевого напряжения. Исходя из указанных значений емкости элемента и максимального тока потребления можно предположить, что отсчет времени в режиме резервного питания сохранится в течение примерно месяца. Однако это было бы верно, если бы микроконтроллер был работоспособен при напряжении питания от 2 В. К сожалению, в данном устройстве при использовании диода Шоттки срок работы резервного питания уменьшается. Впрочем длительная работа от резервного источника питания не имеет смысла, т. к.

происходит только отсчет времени и сохранение различных параметров в ОЗУ, а индикация и остальные периферийные устройства не функционируют. При длительном отключении сетевого напряжения элемент питания на этот срок лучше извлечь из устройства. Несколько продлить срок работы без сетевого напряжения можно, используя вместо диода Шоттки малогабаритное герконовое реле или низковольтный полевой транзистор. Также можно использовать батарею элементов на напряжение 4,5...6 В или экономичный повышающий преобразователь напряжения.

Для звукового оповещения предусмотрен пьезокерамический излучатель BF1, подключенный к порту микроконтроллера через транзистор VT3. Для задания тактовой частоты 4 МГц используется встроенный в микроконтроллер генератор. Для таймера реального времени, реализованного на таймере-счетчике 2 микроконтроллера, установлен кварцевый резонатор ZQ1 на частоту 32768 Гц.

Датчики температуры AD592 с токовым выходом подключаются непосредственно к входам АЦП микроконтроллера. Для преобразования

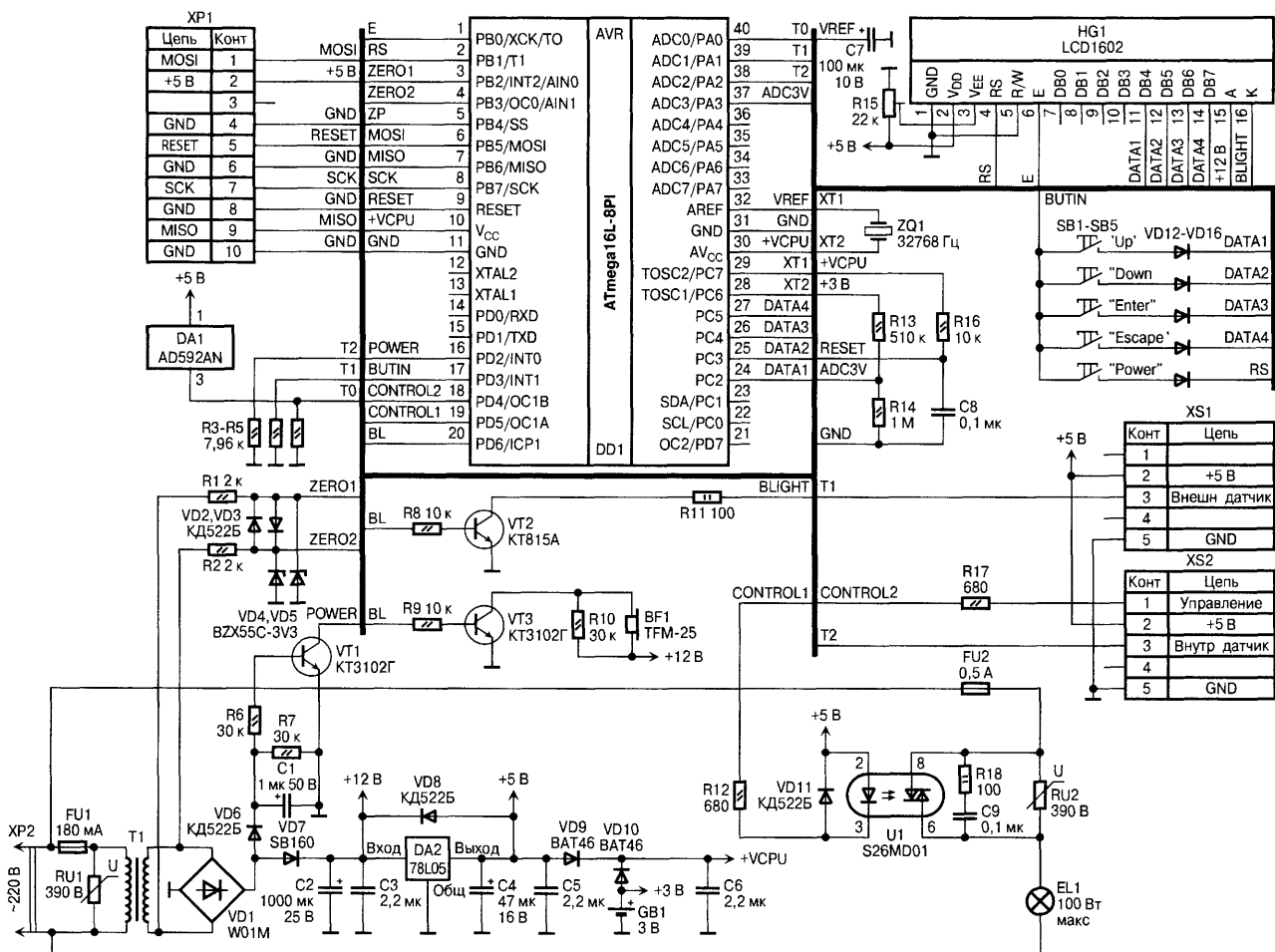


Рис. 1

тока в напряжение служат резисторы R3—R5. Номиналы резисторов рассчитываются по формуле:

$$R = U_{\text{оп}} / I_{\text{макс датч}},$$

где $U_{\text{оп}}$ — опорное напряжение, В; может использоваться как встроенный в микроконтроллер источник опорного напряжения на 2,56 В, так и внешний; $I_{\text{макс датч}}$ — выходной ток датчика температуры при максимальной заданной пользователем температуре. Для повышения точности измерения рекомендуется выбрать верхнюю границу измеряемой температуры на уровне 40...50 °С. Для AD592 выходной ток при температуре 25 °С составляет 298,2 мкА, температурный коэффициент равен 1 мкА/°С.

Датчик температуры DA1, расположенный на плате, изначально предназначался для проверки правильности функционирования узла измерения температуры. Из-за фиксированного расположения проводить его калибровку затруднительно. При желании можно этот датчик не устанавливать, используя для измерения комнатной температуры внешний датчик. Любая систематическая погрешность легко устраняется программной калибровкой.

При установке датчиков температуры с выходом по напряжению (рис. 2) потребуется дополнительный конденсатор, который желательно установить в непосредственной близости к датчику.

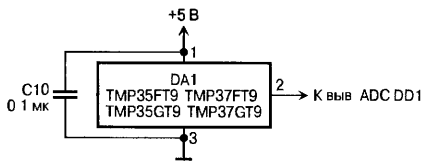


Рис. 2

Подборки резисторов в этом случае не требуется из-за отсутствия таковых. В этом случае при применении внешнего источника опорного напряжения выбор последнего должен происходить в соответствии с напряжением датчика, соответствующим максимальной выбранной температуре.

Блок управления сильноточной нагрузкой показан на рис. 3.

Управляющий сигнал с микроконтроллера открывает оптосимистор U2, который включен в управляющую цепь силового симистора VS1. Двухцветный светодиод HL1 индицирует зеленым свечением наличие сетевого напряжения, а красным — пере-

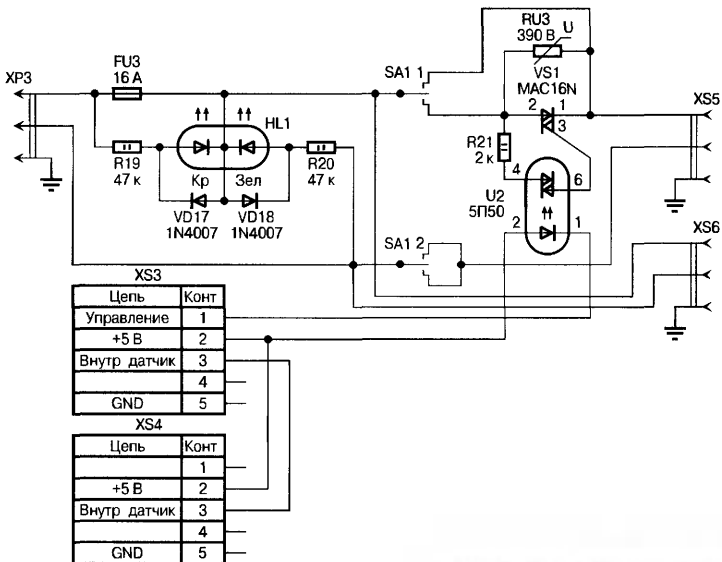


Рис. 3

вание сетевого предохранителя FU3. Трехпозиционным переключателем SA1 выбирается режим включения нагрузки: отключение от сетевого напряжения, включение напрямую, минуя управляющий симистор, либо через симистор с возможностью управления нагрузкой дистанционно. Через блок управления сильноточной нагрузкой транзитом проходит сигнал с датчика температуры. Это удобно, когда требуется измерять температуру вблизи обогревателя, находящегося на значительном удалении от управляющего устройства. Если управление сильноточной нагрузкой не используется, датчик температуры можно подключать прямо к разъему XS2, измеряя температуру внутри помещения.

Устройство выполнено на нескольких платах: на одной располагаются все основные элементы блока управления, на плате, установленной под передней панелью рядом с индикатором, располагаются кнопки управления (рис. 4), плата блока управления сильноточной нагрузкой расположена в отдельном корпусе.

Автор придерживается мнения, что печатные платы лучше всего разводить под имеющиеся корпус и комплектацию. Вот общие рекомендации по расположению элементов и разводке платы. Блокировочные конденсаторы должны устанавливаться вблизи выводов питания микросхем, то же касается и конденсатора в цепи RESET микроконтроллера. Общий провод от таких конденсаторов желательно выделить и развести его отдельно к выводу GND микроконтроллера. Также автор рекомендует для всего, что должно соединяться



Рис. 4

с основной платой проводниками (кнопки, разъемы на корпусе, выносные датчики и т. п.), использовать разъемы. Располагать элементы необходимо, разделяя высоковольтные и низковольтные цепи, стараясь, чтобы они не пересекались друг с другом. После монтажа желательно покрыть высоковольтные цепи на плате и радиодетали, принадлежащие к этим цепям, электроизоляционным лаком в несколько слоев. При монтаже и наладке необходимо соблюдать правила электробезопасности.

Конструкция внешних датчиков произвольна. Желательно датчики герметизировать, чтобы можно было впоследствии калибровать их, опуская в воду. Хорошие результаты дает применение термоусадочных трубок разных диаметров.

Микроконтроллер ATmega16L-8PI можно заменить на более дешевый ATmega8535L-8PI, если не использовать большой набор мелодии, а всего одну или две. Для перехода на этот микроконтроллер придется незначительно переделать программу. Также можно использовать внешний источник опорного напряжения,

который подключается к выводу 32 AREF (нумерация дана для корпуса DIP). В программе предусмотрено использование датчика освещенности на фотодиоде (рис. 5) для автоматического включения подсветки LCD-индикатора.

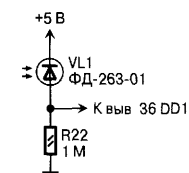


Рис. 5

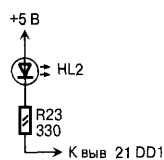


Рис. 6

Полезно будет добавить подсветку кнопки включения лампы (рис. 6).

В программе можно выбрать различные способы подсветки: постоянное свечение светодиода, мигание с частотой пересчета таймера реального времени или отключение подсветки.

Индикатор — любой формата 16 символов в две строки с встроенным контроллером, совместимым с HD44780. При замене может возникнуть проблема несовместимости кодировок русского шрифта, которая устраняется только путем замены подключаемого файла text.inc, в котором необходимо заменить весь текст на кодировку используемого индикатора. У примененного автором индикатора кодировка формата Erson, но более предпочтительной является кодировка ASCII, при которой текст в программу можно вводить напрямую. В случае применения нерусифицированного индикатора его можно русифицировать известными методами [3] или выводить информацию на английском или другом языке.

Звукоизлучатель — любой с достаточной громкостью звучания. В корпусе прибора желательно просверлить отверстия напротив излучателя для улучшения его слышимости. Если громкость все же недостаточна, можно вместо пьезоизлучателя применить малогабаритную динамическую головку. В этом случае резистор R10 можно исключить. Трансформатор — малогабаритный для печатного монтажа, например фирмы Nahn, но подойдет любой на выходное напряжение 7...10 В и ток около 0,5 А. Выпрямительный мост — любой низковольтный на ток 0,5...1 А, VD7 — диод Шоттки на ток 0,5...1 А или любой выпрямительный на тот же ток. Диоды VD9, VD10 — диоды Шоттки с минимальным падением напряжения. Остальные диоды — КД521Б или любые импульсные или выпрямительные. Оптосимистор U1 — S26MD01 или по-

добный на ток от 0,6 А. Необходимо проследить, чтобы симистор не имел узла Zero-cross, который переключает симистор только при переходе через ноль сетевого напряжения. Оптосимистор U2 — 5П50 (отечественный аналог — АОУ163), подойдут импортные MOC3020M—MOC3022M, симистор VS1 — MAC16M, подойдут любые на ток 10...16 А, например отечественные КУ208Г, ТС106-10-6, ТС112-10-6, ТС112-16-6. Резисторы R3—R5 желательно использовать прецизионные C2-29B-0,125; R11, R18—R20 MF-2, ОМЛТ-2; R21 — MF-1, ОМЛТ-1; подстроечный резистор R15 — СП5-2ВБ, СП3-19а; остальные резисторы — MF-0,125, ОМЛТ-0,125. Варисторы RU1—RU3 — JVR20N-391M. Конденсаторы оксидные — SR, керамические — K10-17а, C9 — K73-17 на напряжение 400...630 В. Разъемы для подключения датчиков температуры и кабеля к блоку управления сильноточной нагрузкой — ОНЦ-ВГ-1-3/16Р (В — ответная часть), розетки XS5, XS6 — обычные для открытой проводки. Разъем программирования XP1 зависит от типа применяемого программатора. В авторском варианте это IDC-10M с разводкой выводов, соответствующей программатору STK-200 производства Atmel. Программировать лучше внутрисхемно, но разъем под микроконтроллер предусмотреть все же стоит.

После монтажа радиоэлементов не вставляя микроконтроллер в разъем и включив питание необходимо проверить напряжения +12 В и +5 В. Затем выключить питание, установить микроконтроллер и, подав питание, запрограммировать его. При необходимости можно установить оптимальную контрастность изображения на индикаторе подстроечным резистором R15.

Наладка заключается в калибровке датчиков температуры. В главном меню выбирается пункт «Уст. параметров» — «Калибровка термодатчиков». Затем выбирается тип датчика (встроенный, внешний датчик для измерения уличной температуры и внешний датчик для измерения комнатной температуры). Первая калибровочная константа подбирается при температуре датчика 0 °С (датчик опускается в воду с тающим снегом или льдом), на индикаторе при этом необходимо кнопками «Up» и «Down» установить значение температуры, близкое к нулю. Затем, нажав кнопку «Enter» (константа сохраняется в памяти EEPROM микроконтроллера), необходимо установить темпе-

ратуру по второй произвольной точке, определяемой по образцовому термометру, чем ближе она будет к максимальной границе измеряемой температуры, тем точнее будет произведена калибровка. Достаточную точность можно получить, используя для калибровки температуру человеческого тела.

Система меню, отображаемая индикация и действия кнопок в каждом конкретном случае поясняются в файле mode.doc в приложении к файлу прошивки. Добавление функций устройства и пунктов меню не составит большого труда. Программа, написанная на ассемблере, приведена на сайте редакции www.dian.ru/programs. Число мелодий определяется подключаемым файлом melody.inc, в котором пользователь может сам задать нужные мелодии, пользуясь методикой, приведенной в [4].

При программировании следует обратить внимание на биты Fuse. При неправильном их программировании устройство может не отвечать заявленным требованиям или вообще быть неработоспособным. Так для установки тактовой частоты 4 МГц необходимо выставить биты CKEL0—CKEL3 в 0011. Чтобы обеспечить микропотребление, необходимо запрограммировать следующие биты: BODEN = 1, OCDEN = 1, JTAGEN = 1 (1 — незапрограммированный бит). Остальные биты оставить по умолчанию.

ПЕРЕЧЕНЬ ЛИТЕРАТУРЫ:

1. А. Брякин. Дистанционный регулятор яркости. — Схемотехника, 2004, № 6, с. 42.
2. С. Бирюков. Устройства на микросхемах. Цифровые измерительные приборы, источники питания, любительские конструкции. — М.: «Символ-Р», 2000.
3. http://www.compitech.ru/html.cgi/arhiv/01_06/stat_22.htm
4. М. Белецкий. Музыкальный звонок на 120 мелодий. — Радио, 2004, № 2, с. 33.
5. http://www.atmel.com/dyn/resources/prod_documents/doc2466.pdf
6. <http://data.enerqizer.com/PDFs/cr2032.pdf>
7. <http://www.atmel.ru/Spec/LCD.htm>
8. <http://www.atmel.ru/Articles/Atmel17.htm>

Сергей Лузгин,
г. Екатеринбург

Электронная игрушка — автомат световых эффектов

В статье предлагается простая и забавная детская игрушка, которая основана на световом эффекте. Игрушка представляет собой палочку, при размахивании которой происходит механическая развертка простого изображения.

В наше время в любом киоске, торгующем китайскими игрушками, можно увидеть множество интересных электронных вещей, работа которых основана на различных световых эффектах. Среди них светящаяся бижутерия, мигающие значки, переливающиеся палочки и т. п. Казалось бы, добавить тут уже нечего, но игрушку, которая предлагается вашему вниманию, никто из моих знакомых пока в продаже не встречал.

Устройство представляет собой имитацию «волшебной палочки». Если размахивать ею в темноте, появляется коротенькая надпись или простенький рисунок, висящий в воздухе. Испытания прибора показали, что, несмотря на примитивность отображаемых рисунков, он может позабавить не только детей, но и взрослых. Первые зрители предложили неожиданное рабочее название проекта — «Палочка-ругалочка».

Эскиз конструкции приведен на рис. 1.

Принцип действия устройства очень прост и известен со времен первых телевизоров с механической разверткой. Ряд светодиодов равномерно распределен на рабочем конце «палочки». При размахивании «палочкой» каждый светодиод описывает в пространстве свою траекторию. При движении слева направо микроконтроллер зажигает светодиоды в соответствии с отображае-

мым рисунком, что и обеспечивает механическую развертку изображения. Обратный ход развертки (справа налево) не используется, т. к. из-за неравномерности движения трудно обеспечить совпадение таких «полукладов».

Кадровая синхронизация обеспечивается следующим образом. Микроконтроллер отслеживает моменты изменения направления движения «палочки» с помощью инерционного контактного датчика ускорения. Принцип действия датчика поясняет рис. 2.

Датчик представляет собой грузик, закрепленный на конце пружинки. Грузик является также подвижным контактом датчика. С двух сторон от него закреплены два неподвижных контакта А и В. При резком изменении скорости, которое происходит при

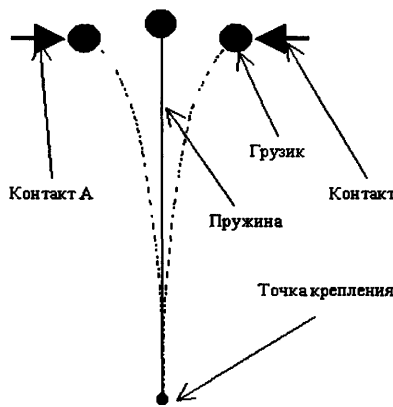


Рис. 2

изменении направления движения «палочки», грузик отклоняется от положения равновесия и замыкает один из контактов. Замыкание контакта А микроконтроллер воспринимает как начало рабочего хода развертки и начинает выдачу информации на светодиоды. Рабочий ход продолжается до замыкания контакта В. Длительность рабочего хода измеряется таймером-счетчиком микропроцессора, и для следующего рабочего хода развертки тактовая частота выдачи информации на светодиоды задается такой, чтобы обеспечить развертку всей картинке с учетом измеренного времени предыдущего рабочего хода.

Кадр изображения хранится в ЕЕ-ПРОМ памяти микроконтроллера в виде массива из 255 байт, младший бит первого байта отвечает за отображение левой верхней точки, старший бит — за левую нижнюю точку раstra. Соответственно, младший бит 255-ого байта соответствует правой верхней точке, а старший бит — правой нижней точке. Если соответствующий бит установлен в 1, то точка засвечена, если в 0 — погашена. Следует заметить, что в начале и в конце рабочего хода развертки из-за неравномерности движения возникают большие геометрические искажения раstra, что в принципе требует формирования кадрового гасящего импульса. Однако в данном устройстве кадровый гасящий импульс не формируется, а просто несколько первых и последних байтов памяти кадра не используются.

Принципиальная схема устройства приведена справа на рис. 3.

Вариант адаптера для загрузки файла изображения в память через COM-порт компьютера показан слева на рис. 3. Использован внутренний RC генератор микроконтроллера с целью



Рис. 1

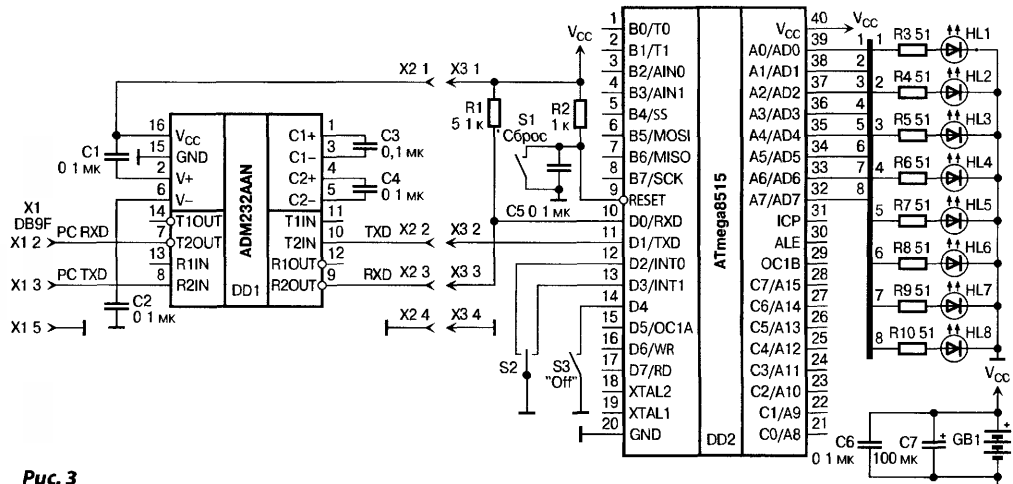


Рис. 3

уменьшения числа навесных элементов. Несмотря на это, эксплуатация показала устойчивую связь через COM-порт. Сразу отмечу, что адаптер не обязателен — изображение можно загружать тем же программатором, которым прошивается рабочая прошивка.

Как видно из рис. 3, основой устройства является микроконтроллер AVR ATmega8515. Вывод изображения обеспечивается сверхяркими светодиодами HL1—HL8. Кнопка S1 «Reset» обеспечивает включение и перевод устройства в режим загрузки изображения с персонального компьютера. Кнопка S3 обеспечивает отключение устройства при его транспортировке. Элемент S2 — это описанный выше датчик ускорения. В состоянии покоя оба его контакта должны быть разомкнуты.

Устройство работает следующим образом.

После нажатия на кнопку S1 микроконтроллер выдает в последовательный порт USART 255 байт содержимого EEPROM. Это предусмотрено для того, чтобы можно было проверить «картинку», ранее загруженную в EEPROM. После этого микроконтроллер переходит в режим ожидания загрузки с компьютера. Если послать байт в последовательный порт, то он запишется в нулевую ячейку EEPROM, с которой начинается отображение. По окончании записи контроллер выдаст в USART записанный байт и байт адреса, по которому он записан, что позволяет проверить правильность записи. Следующий посланный байт будет записан в первую ячейку EEPROM и т. д. После записи 255-го байта адрес записи обнулится, и если передать еще байт, он перезапишет байт в нулевой ячейке. Таким образом, если нажать кнопку «Сброс», а затем передать 255 байт картинки, дожидаясь каждый раз двух ответных байт от микроконтроллера, картинка будет загружена в EEPROM. Важно заметить, что во время программирования устройство нельзя двигать, т. к. при первом же срабатывании датчика S2 программирование будет прервано.

Если картинка уже ранее была загружена в память, то после нажатия кнопки «Сброс» можно сразу начинать размахивать «палочкой».

В начале движения «палочки» слева направо замыкается левый по схеме контакт датчика S2, что вызывает прерывание INT0. Обработчик этого

прерывания разрешает работу таймера T0 микроконтроллера с тактовой частотой $F_{CLC}/256$ и таймера T1 с тактовой частотой F_{CLC} . Подпрограмма обработки прерываний при переполнении T0 ведет подсчет числа переполнений. Полученное к концу рабочего хода развертки двухбайтное число (старший байт — число переполнений, младший — содержимое регистра TCNT0) является константой задержки для следующего цикла развертки. Если «палочку» оставить в покое на время около 0,5 с, это число достигнет 0xFF00 и в этом случае светодиоды HL1—HL8 отключаются, а микроконтроллер командой Sleep переводится в режим микропотребления.

В конце рабочего хода, когда замыкается правый по схеме контакт датчика S2, происходит прерывание INT1, его обработчик останавливает счетчик T0, считывает полученную двухбайтную константу задержки и заносит ее в регистр совпадения A таймера T1. Затем он выдает полученную константу задержки в USART, что предусмотрено с целью диагностики, после этого обработчик вновь запускает таймер T0. Если до наступления прерывания INT0 (т. е. до начала нового рабочего хода развертки) таймер T0 успеет переполниться 255 раз, командой Sleep микроконтроллер переводится в режим микропотребления.

Во время рабочего хода развертки (в интервале времени между вызовом INT0 и INT1) работает таймер T1 с тактовой частотой F_{CLC} . Он настроен на сброс при совпадении A (т. е. при достижении константы задержки, произойдет сброс таймера T1 в 0). Одновременно будет вызвано прерывание по совпадению A таймера T1. Обработчик этого прерывания извлекает очередной байт данных из EEPROM и выводит этот байт в порт A, к которому подключены светодиоды. Таким образом, происходит обновление информации на светодиодах. Возможно, очередной цикл развертки окажется короче предыдущего, тогда к моменту окончания цикла развертки (т. е. к моменту наступления прерывания INT1) будут выданы еще не все байты из EEPROM. В таком случае в обработчике INT1 вывод информации будет остановлен, а все светодиоды погашены.

Как уже упоминалось, если импульсы на входы INT0 и INT1 не поступают в течение 0,5 с, микроконтроллер переводится в режим микропотребле-

ния (Power Down). В рабочем режиме входы INT0 и INT1 настроены на формирование прерывания по спаду. Однако спад на этих входах обнаруживается синхронно с тактовой частотой микроконтроллера, поэтому в режиме Power Down он не может быть обнаружен. По этой причине для того, чтобы микроконтроллер «проснулся» при поступлении отрицательного импульса на входы INT0 или INT1, перед «усыплением» оба входа программируются на прерывание по низкому уровню.

Как видно из описания алгоритма, выключатель питания для данного устройства не обязателен, поскольку при отсутствии активности на входах INT0 и INT1 светодиоды отключаются, а микроконтроллер переходит в режим Power Down. Однако при транспортировке устройства оно будет постоянно срабатывать от вибрации. Чтобы избежать этого, можно нажать на кнопку S3. При обнаружении низкого уровня на входе PD4 все прерывания запрещаются, и командой Sleep микроконтроллер переводится в режим Power Down. Теперь вывести из этого режима его можно только сигналом Reset (кнопка S1). Следует только иметь в виду, что если микроконтроллер в момент нажатия кнопки S3 уже «спал», то нажатие не вызовет никакой реакции, т. к. состояние входа PD4 опрашивается программно, а не по прерыванию. Поэтому чтобы надежно отключить устройство, надо нажать кнопку S3 и потрясти «палочкой», чтобы «разбудить» процессор посредством прерывания INT0 или INT1.

Есть еще особенности, связанные с автоматическим отключением микроконтроллера. После сброса кнопкой S1 микроконтроллер переходит в режим ожидания записи информации от компьютера, при этом он не перейдет в режим Power Down, пока хотя бы раз не будет вызван обработчик INT0 или INT1. Поэтому, если нажать кнопку S1 и аккуратно положить устройство, оно не перейдет в энергосберегающий режим и батареи питания будут разряжаться. А если в процессе программирования его потрясти, то оно через 0,5 с «заснет», и программирование не состоится.

Есть и более простое средство отключения устройства — выключатель питания. Тогда кнопки S1 и S3 не нужны. Если не пользоваться адаптером COM-порта, их можно не ставить и при отсутствии выключателя.

Чтобы получить от игрушки полную отдачу, надо подготовить интересную «картинку» и загрузить ее в электропрограммируемую память микроконтроллера. Образ «картинки», как вы помните, представляет собой 255 байт. В литературе уже многократно описывались простейшие знакогенераторы, их преобразование в двоичный код и другие подобные вопросы, поэтому не стоит на этом подробно останавливаться. Напомню, что если вывести первый байт 0x01, а остальные 254 байта будут нулевыми, то должна гореть одна точка раstra в верхнем левом углу, а если первый байт будет 0x80 — в нижнем левом углу. Однако следует учитывать следующее. Моменты срабатывания контактов датчика S2 не совсем точно соответствуют моментам начала и конца движения «палочки» слева направо, хотя и очень близки по времени. В результате несколько первых и последних байтов могут быть отображены не на своих местах. Именно поэтому несколько первых байт и несколько последних необходимо обнулить и не использовать. Сколько именно — зависит от конкретной конструкции датчика S2. Это легко установить экспериментально, выводя картинки типа вертикальной черты шириной в один пиксель или одной точки. Второй важный момент — движение «палочки» происходит с ускорением, поэтому ближе к правому краю изображение растягивается. Например, в авторском варианте устройства пиксели с правой стороны изображения шире пикселей слева примерно в полтора раза. Этот эффект можно компенсировать, заранее исказив изображение, т. е. сделав его постепенно сжимающимся вправо. Именно с этой целью и был выбран размер изображения 255 байт (а не 64, например).

Для подготовки изображений была написана программа AmbusiveStick, которая позволяет подготовить двоичный файл прошивки EEPROM и зашить его. Главное окно программы показано на рис. 4.

Делается это так. Запустите программу AmbusiveStick.exe. В верхней части окна находится поле, состоящее из прямоугольников. Каждый прямоугольник соответствует пикселю изображения. В нижней части окна видно содержимое файла картинки в шестнадцатеричном виде. Первый щелчок мыши на стилизованном изображении пикселя зажигает его,

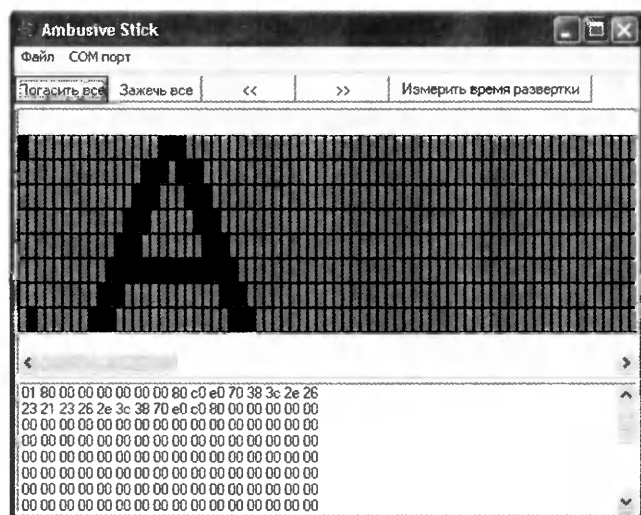


Рис. 4

повторный щелчок гасит. Когда картинка будет готова, нужно сохранить ее (команда «Файл → Сохранить»).

Если вы собрали адаптер (рис. 3), можно подключить его к COM-порту и к микроконтроллеру, затем надо подать питание на микроконтроллер. После этого кратковременно нажать кнопку S1 и вызвать пункт программы AmbusiveStick «Файл → Загрузить в микроконтроллер», будет проведена запись в EEPROM. При этом надо следить, чтобы не сработал датчик S2, иначе микроконтроллер заснет и запись не состоится.

Также программа AmbusiveStick позволяет загрузить файл из контроллера, для чего нужно выбрать пункт меню «Файл → Прочитать из контроллера» и затем кратковременно нажать кнопку S1.

Программа AmbusiveStick использует следующие настройки COM-порта:

- порт COM1;
- скорость 9600 бод;
- без контроля четности;
- два стоп-бита.

Если адаптер подсоединен не к COM1, то в меню выбирайте COM-порт. Остальные параметры должны соответствовать приведенным выше.

Можно прошить файл картинки без адаптера. Для этого следует загрузить его в программу-программатор как двоичный образ EEPROM и прошить в EEPROM микроконтроллера. Автор использовал для этой цели Ponyprog V2.06. Файл нужно загрузить в Ponyprog как двоичный файл (*.bin) образа EEPROM и записать в EEPROM.

Биты FUSE микроконтроллера DD2 следует запрограммировать следующим образом:

- BODEN = 0, BODLEVEL = 1 (т. е. ячейка BODEN запрограммирована, BODLEVEL сброшена), при этом схема BOD включена, порог срабатывания 2,7 В;
- WDTON = 1 (Watchdog таймер отключен);
- CKPOT = 0 (большая амплитуда сигнала тактового генератора);
- CKSEL3...0 = 0100 (работа тактового генератора от внутренней RC-цепочки, частота F_{CLC} — 8 МГц).

Состояние остальных битов FUSE безразлично. Можно оставить и состояние этих битов по умолчанию, но следует иметь в виду, что будет использоваться комбинация CKPOT = 1, CKSEL3...0 = 0001, при этом тактовый генератор микроконтроллера работает от встроенной RC цепочки на частоте 1 МГц. Скорость передачи через USART станет не 9600 бод, а 1200. Если вы используете адаптер COM-порта, это необходимо учесть. По умолчанию работа цепи сброса при снижении питания (BOD) запрещена. Поэтому возрастает вероятность, что при разряде батарей микроконтроллер начнет работать некорректно и испортит содержимое EEPROM, что потребует после смены батарей перезагрузить картинку. В остальном работоспособность устройства сохранится.

Использованные детали широко доступны. Вместо микросхемы DD1 можно использовать другие преобразователи интерфейса RS-232 (например, MAX232). Микроконтроллер ATmega8515 был выбран в основном потому, что он имелся в наличии, а также в расчете на дальнейшее развитие устройства, в частности, на увеличение числа светодиодов. При использовании других микроконтроллеров

роллеров семейства AVR потребуются незначительная модификация программы. Например, для AT90S2313 некоторые регистры имеют другие имена, размер EEPROM всего 128 байт, отсутствует порт А и т. п. Кроме того, для этого микроконтроллера потребуется подключение внешнего кварцевого резонатора.

Светодиоды HL1—HL8 должны быть обязательно ультраяркие, желательно квадратной формы, чтобы их можно было установить вплотную друг к другу. Не забывайте, что максимальная нагрузочная способность выходов DD2 не более 40 мА, и в соответствии с этим выбирайте номиналы R3—R10 для конкретных светодиодов.

В качестве X1 можно использовать разъем от неисправной компьютерной мыши вместе с проводом. Кнопки S1 и S3 — любые миниатюрные.

Датчик S2 в авторском варианте представляет собой плоскую пластинку из нержавеющей стали размерами 60×3 толщиной 0,1 мм, на конце которой наплавлена капля припоя диаметром 6 мм. Собственная частота колебаний датчика около 5 Гц. Наверняка лучше подошла

бы пластинка из бериллиевой бронзы или обрезок часовой пружины, но, как показал опыт, конструкция датчика не критична. Контакты датчика — это две монтажные стойки диаметром 1 мм, впаянные в плату на расстоянии 20 мм друг от друга. Расположить датчик следует так, чтобы грузик находился на самом конце «палочки», где амплитуда колебаний максимальна.

Источник питания устройства GB1 — батарея из трех элементов 1,5 В, соединенных последовательно. Средний ток потребления около 60 мА. Выводы питания следует зашунтировать конденсатором C7 емкостью не менее 100 мкФ. Во время отладки устройства его удобно питать от порта USB компьютера, для чего можно взять кабель USB и разрезать его. Внутри находятся четыре провода. Черный — общий, красный — +5 В, остальные два не используются.

Проект программы AmbusiveStick, исполняемый файл AmbusiveStick.exe, асемблерный файл программы для микроконтроллера AmusiveStick.asm, файл прошивки AmusiveStick.hex и демонстрационный файл прошивки EEPROM heart.bin доступны на сайте

журнала www.dian/programs. Программа AmbusiveStick написана в среде Borland C++ Builder 6, описанию которой посвящен цикл статей [1]. Программа сильно упрощена, поэтому имеет некоторые «странности», например, не предупреждает о необходимости сохранить измененный файл. Программа управления микроконтроллером AmusiveStick.asm написана на Ассемблере в среде AVR Studio 4.

Авторский вариант устройства собран на макетной плате, печатная плата не разрабатывалась.

Хочется отметить, что современная элементная база позволила сделать устройство очень простым и довольно дешевым. Его можно рекомендовать для сборки энтузиастам, делающим первые шаги в освоении AVR.

ЛИТЕРАТУРА:

1. О. Вальна. *Borland C++ Builder 6 для начинающих*. — Схемотехника, 2004, № 6—2005, № 5.
2. А. В. Евстифеев. *Микроконтроллеры AVR семейств Tiny и Mega фирмы Atmel*. — М.: ИД «Додэка-XXI», 2004.

Владимир Захаров,
г. Снежинск Челябинской обл.

О керамических конденсаторах

В статье рассмотрена проблема зависимости емкости керамических конденсаторов от приложенного напряжения.

Однажды в процессе работы над очередным электронным устройством автору пришлось столкнуться с тем, что емкость некоторых керамических конденсаторов находилась в значительной зависимости от приложенного напряжения. Чтобы более точно определить характер этой зависимости, был собран простейший стенд (рис. 1), где измерялась емкость двух последовательно включенных конденсаторов: C_X — исследуемого и C_O — вспомогательного.

При этом на исследуемый конденсатор подавалось напряжение с переменного резистора R1, которое можно было изменять от 0 до 50 В

(номинальное рабочее напряжение исследуемого конденсатора). Поскольку результирующую емкость последовательного соединения можно вычислить по формуле

$$C = C_X \times C_O / (C_X + C_O),$$

при известных $C_{изм}$ и C_O легко найти C_X :

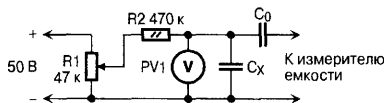
$$C_X = C_O \times C_{изм} / (C_O - C_{изм}).$$

Емкость конденсатора C_O выбирается в диапазоне 0,1...10 C_X , можно использовать, например, конденсаторы типа К73-17. Резисторы R1 и R2 — любого типа. От резистора R1 и вольтметра можно вообще отказаться, если под рукой имеется источник питания с регулировкой и контролем выходного напряжения. В этом случае источник питания может быть

подключен к проверяемому конденсатору через резистор R2.

Таким образом было проверено несколько импортных керамических конденсаторов — аналогов отечественных К10-17 различной емкости с разными группами ТКЕ. Результат оказался неожиданным. Емкость некоторых конденсаторов при увеличении напряжения от 0 до 50 В уменьшалась в 10...15 раз! При напряжении 10 В их емкость уже составляла лишь 35...40 % от номинала. Подобное можно объяснить только очень низким качеством примененного диэлектрика. Конечно, далеко не все импортные конденсаторы имеют такие параметры, но проведенный эксперимент говорит о том, что нужно быть готовым к любым сюрпризам, преподносимым нам иностранными производителями.

Виктор Цибин,
г. Березовский Свердловской обл.



Цифровые часы на основе микроконтроллера HT49R30A-1

В этой статье рассмотрено применение микроконтроллера HT49R30A-1 для разработки цифровых часов. Этот микроконтроллер входит в семейство микросхем фирмы Holtek со встроенными драйверами ЖКИ.

HT49R30A-1 представляет собой недорогой восьмиразрядный однократнопрограммируемый (OTP) контроллер с RISC-архитектурой, разработанный специально для применения в изделиях с большим числом входов/выходов. Его основные характеристики были кратко рассмотрены в статье [1].

На рис. 1 приведена принципиальная схема цифровых часов, состоящих из микроконтроллера HT49R30A-1 и нескольких внешних элементов. При разработке использовались средства проектирования TICE49C-ССАА, HT-IDE3000 и программа моделирования ЖКИ.

В цифровых часах предусмотрены два режима работы — основной и режим установки. В основном режиме на дисплее (рис. 2) отображается текущее время (часы, минуты и секунды). Во втором режиме для установки време-

ни на дисплее используются три кнопки: установки, выбора и увеличения/уменьшения. При нажатии на кнопку

установки SET система переключается из одного режима в другой. Когда система находится в режиме установки, кнопкой выбора SELECT задается знакоместо дисплея, которое будет изменяться (при этом оно мигает). После того, как знакоместо выбрано, кнопкой увеличения/уменьшения UP устанавливается требуемая цифра.

Блок-схема программы часов приведена на рис. 3, ее текст можно найти на сайте журнала по адресу www.dian.ru/programs/.

Действия, производимые подпрограммами:

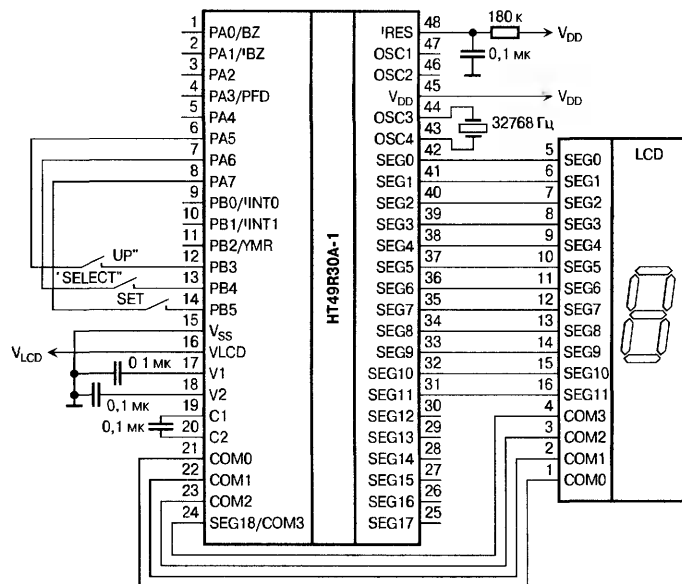


Рис. 1

Новые книги

А. В. Фрунзе

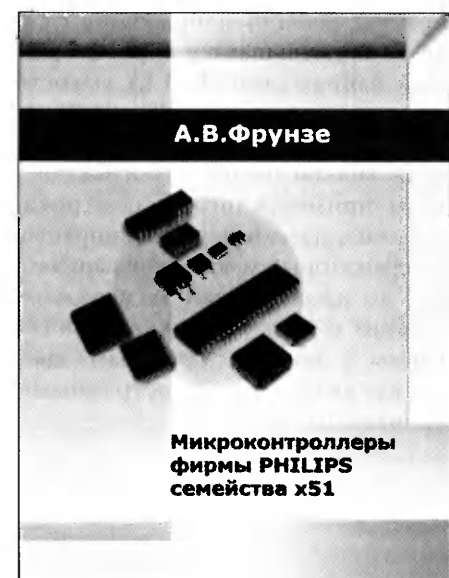
МИКРОКОНТРОЛЛЕРЫ ФИРМЫ PHILIPS СЕМЕЙСТВА x51

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. Наши реквизиты см. на с. 56.



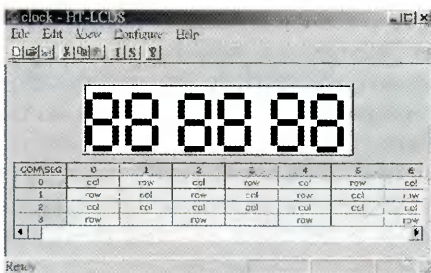


Рис. 2

_TimeCountInt — обработка прерывания по переполнению таймера/счетчика событий. Если значение параметра установлено равным $\frac{1}{4}$ системного такта, подпрограмма будет формировать временной интервал 31 мс.

_TimeBaseInt — обработка прерывания по временной базе. Если значение параметра установлено равным 2^{13} периодов часов реального времени, подпрограмма будет формировать временной интервал 250 мс.

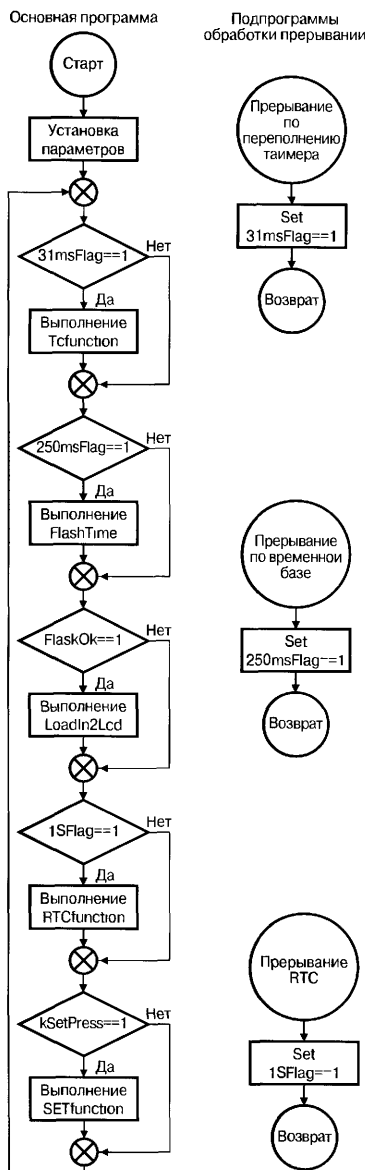


Рис. 3

_RTCInt — обработка прерывания часов реального времени. Если значение параметра установлено равным 2^{15} периодов часов реального времени, подпрограмма будет формировать временной интервал 1 с.

Tcfunction — определение состояния клавиатуры и частоты коммутации ЖКИ.

FlashTime — обновление данных для каждой цифры ЖКИ.

LoadIn2Lcd — запись данных в соответствующую ячейку памяти ЖКИ.

RTCfunction — обновление времени на дисплее.

SETfunction — установка параметров в режиме установки.

Пример индикации, полученной в результате моделирования устройства, представлен на рис. 4.



Рис. 4

В разработанном устройстве, помимо самого микроконтроллера, требуется всего несколько внешних элементов, что позволяет расширить его функции без значительного увеличения стоимости. Например, если необходимо добавить будильник, для этого нужно будет ввести некоторые изменения в программное обеспечение и подключить к выводу PA0 или PA1 зуммер.

ЛИТЕРАТУРА:

1. Спидометр на базе микроконтроллера HT49R30A-1. — Схемотехника, 2005, № 5, с. 39—41.

Liang-Pay Ou,

An MCU-based Digital Clock

Перевод **Николая Раковича**

CHIPSELECT

ПОСТАВКА ЭЛЕКТРОННЫХ КОМПОНЕНТОВ

официальный
дистрибьютор Holtek

поставки
всего спектра
продукции
Holtek

HOLTEK

ВОЗМОЖНОСТЬ
монтажа
CHIP-ON-BOARD,
CHIP-ON-GLASS,
CHIP-ON-CABLE

(095) 267-47-33
(095) 263-96-64
(095) 263-27-37
(095) 101-37-65
(многоканальный)

info@chipselect.ru
www.chipselect.ru

Изготовление печатных плат для макетов

ДВУСТОРОННИЕ ПЛАТЫ. СОВМЕЩЕНИЕ КОНТАКТНЫХ ПЛОЩАДОК

Еще одна проблема — это совмещение контактных площадок при перенесении рисунка на верхнюю (слой TOP) и нижнюю (слой BOTTOM) стороны платы.

Для хорошего совмещения контактных площадок лучше отпечатать оба рисунка — исходный, например, в слое BOTTOM, оставив на нем дорожки и контактные площадки, и зеркально отраженный со слоем TOP. Расстояние между границами рисунков должно быть 20...30 мм. Посередине между двумя рисунками проводится тонкая вертикальная линия, по которой лист будет складываться (линия перегиба), а слева и справа от рисунков, симметрично относительно линии перегиба, размещаются выполненные тонкими линиями метки (кресты или уголки).

Разберем на примере, как организовать такой рисунок в P-CAD.

На рис. 6 показаны слои разведенной платы — TOP (красный), BOTTOM (зеленый) и BOARD (голубой), остальные слои заблокированы (Option/Layers: кнопки Disable/Enable блокируют/разблокируют слои). Границы платы обозначены четырьмя небольшими уголками в слое BOARD. В этом же слое на расстоянии примерно 15 мм справа проведена вертикальная линия перегиба, слева на таком же расстоянии изображен крест.

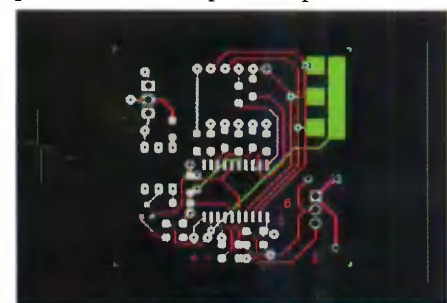


Рис. 6

С этого рисунка при печати получим слой BOTTOM. В этом слое есть залитые площадки Copper Pour, при работе с зеркальным отражением рисунка с ними возникнут дополнительные трудности, кроме того, при печати не понадобится делать дополнительного зеркального отражения нашего рисунка.

Теперь получим рисунок для слоя TOP. Для этого справа от нашего рисунка должно быть достаточно места для его копии.

Выделим блок так, чтобы в него попало все, что изображено на рисунке. Вызовем контекстное меню, щелкнув правой кнопкой мышки, выберем строку Selection Point, определим точку привязки выделенных объектов, установив указатель мышки на нижнюю точку линии перегиба, а затем щелкнем левой кнопкой мышки (рис. 7).

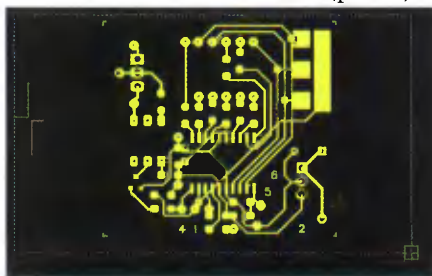


Рис. 7

Скопируем выделенные объекты рисунка (Edit/Copy) и вставим их поверх этого же рисунка (выполнить Edit/Past>From Clipboard, появившийся указатель в виде крестика установить на нижнюю точку линии перегиба и нажать левую кнопку мышки). Теперь у нас точно поверх первой картинке появилась ее копия, из-за их полного совмещения видно только одну картинку.

Нажатие клавиши F вызовет выполнение операции FLIP (зеркальное отображение выделенных объектов относительно точки привязки). Если при этом появится сообщение об ошибке «Item(s) fall outside of workspace», надо расширить рабочую область (Options/Configure, в открывшемся окне на панели Workspace size надо увеличить параметр Width) и повторить попытку.

При успешном завершении операции FLIP слева останется исходный

рисунок с вертикальной линией перегиба, справа — зеркальная копия рисунка, она остается выделенной. Потяните мышкой выделенные объекты влево на два шага сетки до совмещения линии перегиба с ее копией.

Теперь справа надо блокировать слой TOP, слева — слой BOTTOM (рис. 8).

Однако в P-CAD нет возможности блокировать слой лишь для части рисунка. Поэтому на левом рисунке удалим линии в слое TOP, на правом — в слое BOTTOM (напомню, что обработке подвергается копия исходного файла, иначе вам придется восстанавливать его, например, для корректировки).

Начнем с левого рисунка. Выполним Options/Selection Mask, в открывшемся окне Options Selection Mask на закладке Block Selection на панели Items нажмем кнопку Clear All, установим галочку против строк Line, Text и Cu Pour, на панели Select Mode выберем опцию Touching Block, либо Inside Block, а на панели Layers щелкнем мышкой по строке TOP (рис. 9). Закроем окно, нажав кнопку OK. Таким образом, мы разрешили блочное выделение линий, текста и залитых участков только в слое TOP.

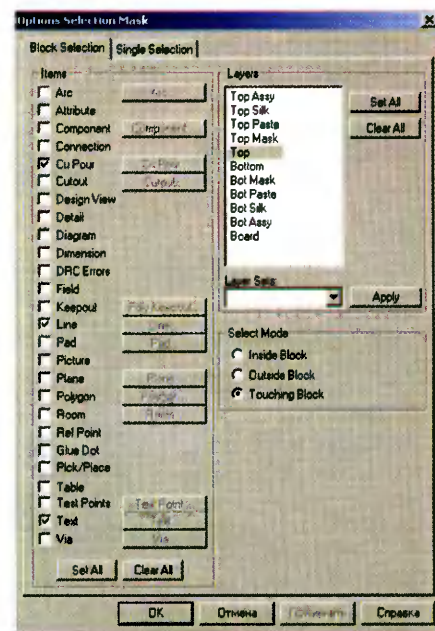


Рис. 9

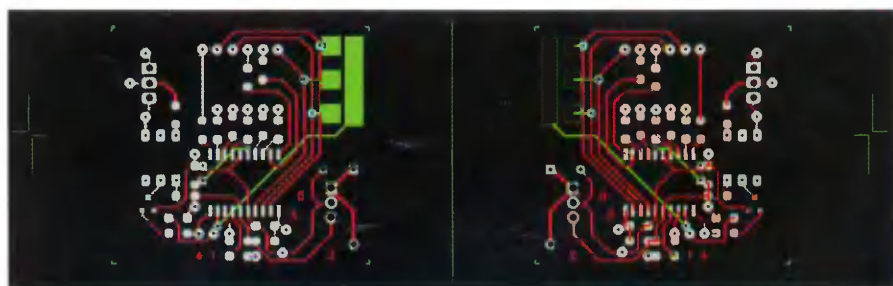


Рис. 8

Блоком выделим левую часть рисунка и нажмем клавишу Del. Синие линии Connections на рисунке слева, возникшие при удалении линий в слое TOP, нам не помешают. Не забудьте восстановить первоначальные свойства блочного выделения (Options/Selection Mask, нажать кнопки Set All на панелях Items и Layers). Аналогичную процедуру повторим для правой части рисунка, только в окне Options Selection Mask на панели Layers щелкнем мышкой по строке BOTTOM и, выделив левый рисунок, нажмем клавишу Del. В результате получим изображение, аналогичное рис. 10.

Подготовим рисунок для печати. Выполнив File/Print, нажмем в открывшемся окне кнопку Setup Print Jobs. В окне Setup Print Jobs установим опции в соответствии с рис. 11 (в окошке Layers несколько слоев выбирается при нажатой кнопке Ctrl), нажмем кнопку Add, потом Close.

В оставшемся на экране окне File Print нажав кнопку Print Preview получим необходимый результат (рис. 12).

Перед переносом рисунка на заготовку платы обязательно просмотрите его, он должен быть зеркальным, проверьте толщину проводников и ширину зазоров. На бумаге огрехи виднее, чем на

Рис. 10

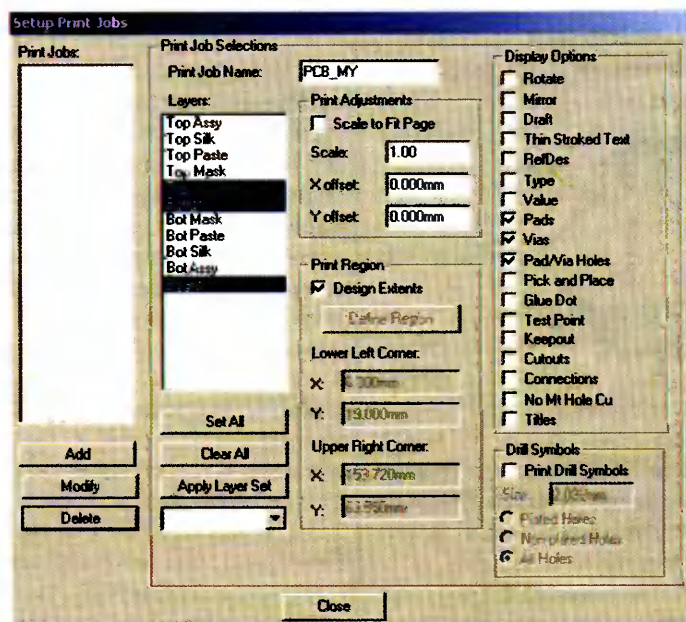


Рис. 11

DECA[®]
SwitchLab

www.platan.ru
ПЛАТАН

ПЕРЕКЛЮЧАТЕЛИ И ИНДИКАТОРНЫЕ ЛАМПЫ



Типы подсветки

- Лампа накаливания 6, 14, 28 В
- Неоновая лампа 110/220 В
- Светодиод 6, 14, 28 В

Типы монтажа

- Монтаж в гнездо
- Пайка на плату

Надежность

- защита IP65
- напряжение изоляции 2000 В
- диапазон температуры -25...+70°C



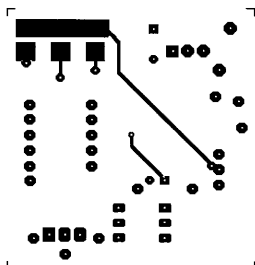
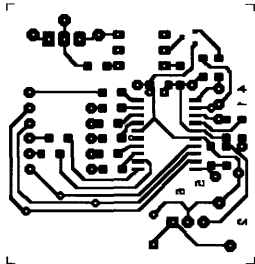


Рис. 12

экране. Часто сэкономленное на контроле время оборачивается довольно длительной черновой работой по устранению пропущенных ошибок.

Обратите внимание на то, что цифры находятся на правом рисунке и зеркально отражены, что свидетельствует об отсутствии ошибок с зеркальностью изображения.

ВЫБОР ШРИФТА ДЛЯ НАДПИСЕЙ

Для того, чтобы надписи перенесли на отпечаток и удовлетворительно получались на плате, надо установить свойства шрифта, представленные на рис. 13.



Рис. 13

Замечу, что шрифты True Type не переносятся на плату.

Окончание следует

Вадим Баранов,
г. Харьков, Украина

Автомобильный электронный балласт для переносной лампы

Предлагаемый набор NK017 позволит автолюбителю собрать простой и надежный электронный балласт для люминесцентной лампы (лампы дневного света). Его отличие от аналогичной конструкции, описанной в предыдущем номере журнала, — более простая схема.

Лампа дневного света (ЛДС) в отличие от стандартной лампы накаливания обладает более высоким КПД, экономичностью и повышенной светоотдачей. Описываемое устройство заменит привычную автомобильную “переноску”, поможет в трудную минуту в дороге и пригодится на отдыхе.

Технические характеристики

Напряжение питания, В	10...15 (тип. 14,4)
Ток потребления, А	1
КПД, %	90
Размеры печатной платы, мм	67×45

Рекомендуемый тип лампы — 11W/21-840 фирмы Osram.

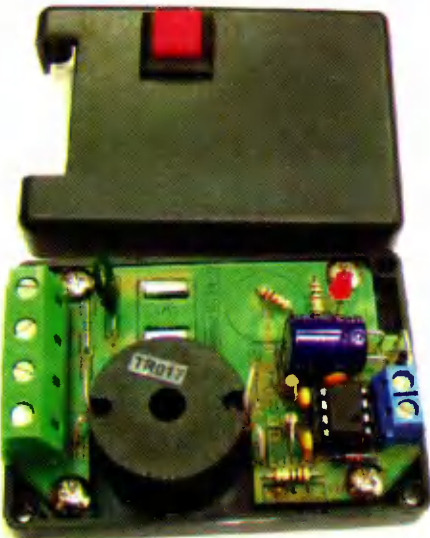


Рис. 1

Общий вид устройства представлен на рис. 1, принципиальная схема — на рис. 2, перечень элементов — в табл. 1.

Электронный балласт выполнен по схеме однотактного обратного преобразователя напряжения. Задающий генератор реализован на таймере NE555 (DA1) в типовом включении. Частота генерации фиксирована и составляет примерно 25 кГц. Ключевой элемент, коммутирующий первичную обмотку трансформатора Т1 — полевой транзистор VT1 (IRF640). Светодиод HL1 индицирует подачу напряжения питания.

В данном устройстве не предусмотрен автоматический поджиг ЛДС, поэтому после подачи питания необходимо замкнуть кнопку SW1 на 0,5...1 с для разогрева нитей лампы.

Таблица 1

Позиция	Наименование	Кол.
C1	0,01 мкФ/630 В	1
C2	470 мкФ/25 В	1
C4	0,1 мкФ	1
C5	0,01 мкФ	1
C6	1000 нФ	1
DA1	NE555	1
HL1	Светодиод Ø3 мм	1
R1	10 Ом	1
R3	1 кОм	1
R4	2,2 кОм	1
R5	10 кОм	1
R6	22 кОм	1
SW1	Кнопка PSW-9AG(9AR)	1
T1	Трансформатор TR017N1	1
VT1	IRF640	1
VD1	1N5817	1
VD3	1N4148	1

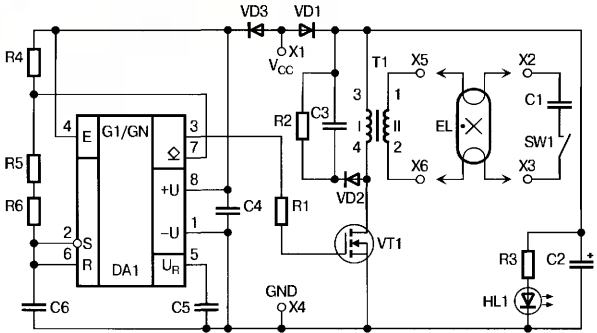


Рис. 2

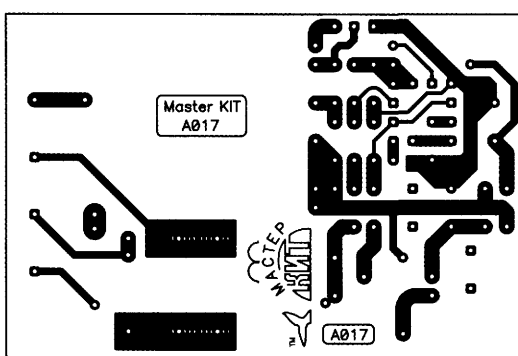


Рис. 3

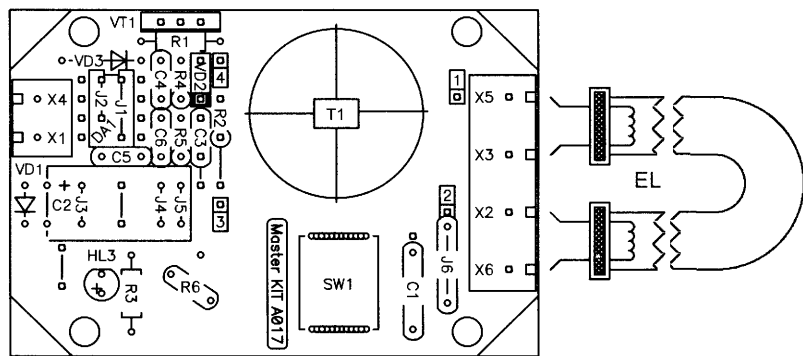


Рис. 4

Источник питания подключают к контактам X1 (+), X4 (-). Для защиты от неправильной полярности установлены диоды VD1 и VD3.

При самостоятельном изготовлении трансформатор T1 наматывают на двухсекционном каркасе проводом ПЭВ-0,3...0,4 виток к витку с послойной изоляцией (можно фторопластовой пленкой или трансформаторной бумагой), собирают в сердечнике из двух ферритовых чашек Ч26 (материал 2000НМ) с зазором 0,05...0,1 мм (чашки склеивают через бумагу толщиной 0,05...0,1 мм)

и пропитывают парафином. Первичная обмотка содержит 30 витков, вторичная — 125.

Устройство собирают на печатной плате размерами 67×45 мм из фольгированного стеклотекстолита (рис. 3), расстановка элементов приведена на рис. 4. Конструкция предусматривает установку платы в корпус BOX-G025, для этого по краям платы выполнены монтажные отверстия под винты 2,5 мм.

На плате предусмотрено место под демпфирующую цепь VD2C3R2 (показана на рис. 2), но опыт изго-

товления устройств показал, что эти элементы не обязательны.

Резисторы R4, R5 и диод VD1 устанавливают перпендикулярно плате, транзистор VT1 — параллельно ей со стороны печатных проводников.

Для удобства подключения к источнику питания и ЛДС на плате предусмотрены посадочные места под клеммные винтовые зажимы.

Правильно собранный электронный балласт не требует настройки.

Юрий Садиков,
г. Москва

Таймер на микроконтроллере с индикатором фирмы Data Vision

Таймеры — одна из самых массовых разработок конструкторов электронных техники, просто невозможно перечислить все применения таймеров в быту, на производстве и в народном хозяйстве. Почти у любого профессионала или начинающего радиолюбителя есть свои разработки по этой теме. Автор предлагает вариант таймера, разработанного на базе микроконтроллера и жидкокристаллического индикатора. Подобные индикаторы в настоящее время — наиболее распространенные устройства отображения информации. Применение ЖКИ позволяет уменьшить потребляемую мощность устройства, еще одна причина массового применения таких индикаторов — большая информативность.

Если в 80—90-х годах прошлого века электронные часы (таймеры или реле времени) строили на обычных или специализированных микросхемах, то сейчас таймеры все чаще разрабатывают на микроконтроллерах. На какие только схемотехнические сложности не шли радиолюбители прошлых лет при разработке различных программируемых управляющих автоматов, а по сути тех же таймеров. Применение микроконтроллеров и ЖК-индикаторов со

встроенными микроконтроллерами в вышеуказанных устройствах, как и в любых других, позволяет значительно упростить их аппаратную часть и при незначительной аппаратной доработке увеличить число исполняемых функций или оперативно изменять их под каждый конкретный случай, внося исправления только в разработанное ранее программное обеспечение.

Принципиальная схема устройства приведена на рис. 1.

Устройство разработано на базе микроконтроллера фирмы Atmel AT89C4051-24PI и двустрочного знаковосинтезирующего жидкокристаллического индикатора DV-16232 FBLV-H/R фирмы Data Vision с встроенным микроконтроллером. К достоинствам подобных индикаторов следует отнести:

- возможность подачи информации по восьми- или четырехразрядной шине;
- встроенный знакогенератор позволяет отображать 192 типа букв, цифр, символов и специальных знаков;
- другие символы можно получить с помощью генератора символов;
- программно доступные команды: очистка индикатора, сдвиг изображения, курсор в начало, сдвиг курсора, вкл./выкл. курсора, чтение/запись данных индикатора, мигание символа и др.;
- компактность позволяет встраивать индикатор в любые устройства и приборы;
- единственный источник питания +5 В;
- низкое энергопотребление;
- широкий диапазон рабочих температур;

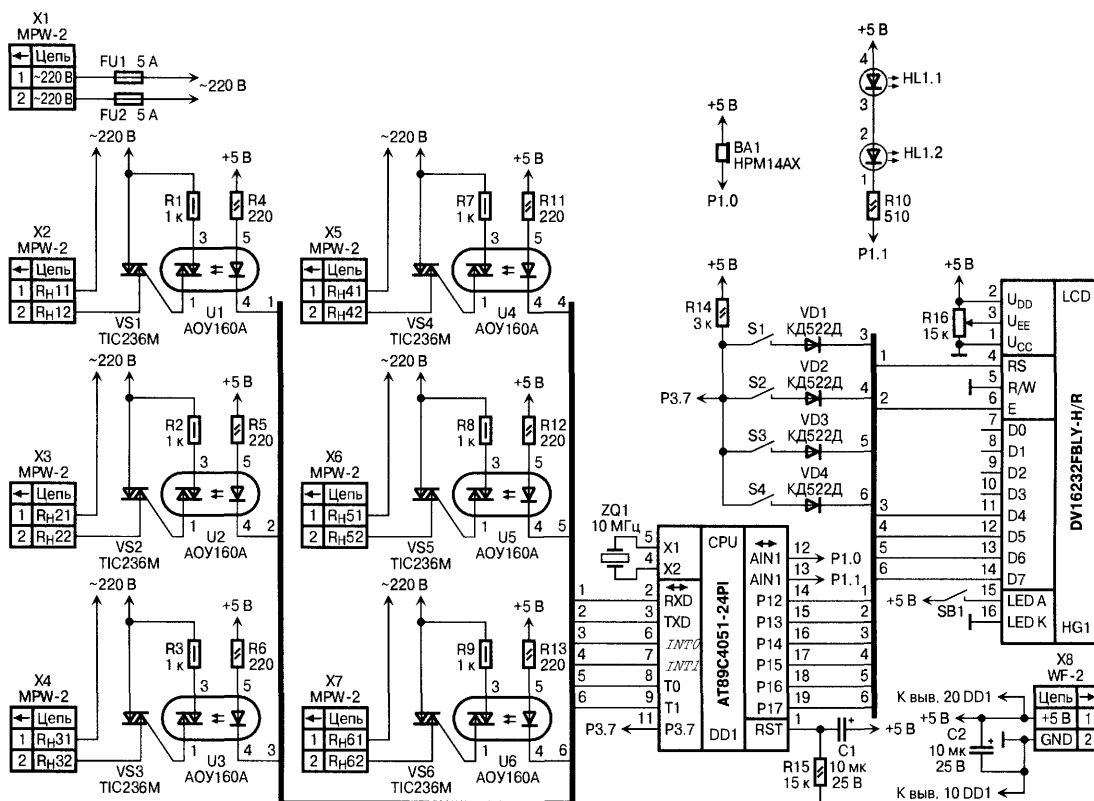


Рис. 1

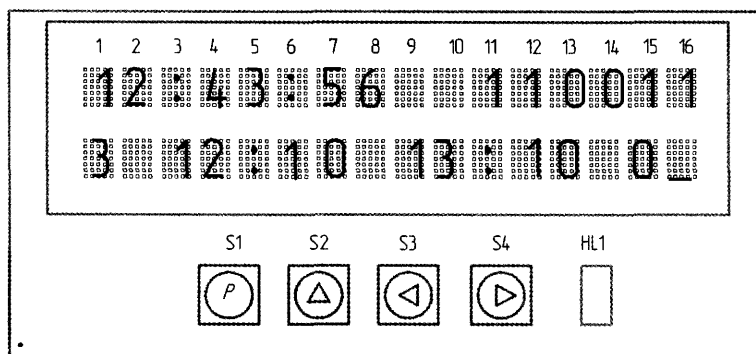


Рис. 2

- встроенные шрифты (англ. или англ./рус.);
- встроенный микроконтроллер (HD44780, KS0076 или аналогичный).

В таймере предусмотрены следующие функции:

- счет реального времени;
- индикация текущего времени в 24-часовом формате в режиме часы—минуты—секунды;
- установка текущего времени и его корректировка;
- установка девяти интервалов времени, в течение которых нагрузки, подключенные к соединителям X2 — X7, включаются в соответствии с заранее заданной программой.

Окончание любого интервала является началом следующего. Границы интервалов в рамках 24-часового формата также программируются. Проще

говоря, в устройстве можно запрограммировать девять будильников, в момент срабатывания которых включаются или выключаются нагрузки, подключенные к устройству. Время включения каждого будильника (часы и минуты) может быть установлено любое. На дисплее ЖК-индикатора HG1 можно одновременно наблюдать текущее время и границы одного из девяти интервалов (время включения и выключения будильника), а также состояние нагрузок (вкл. или выкл.) в данном интервале. Если текущее время совпало с началом какого-либо интервала, на 60 с включается звуковой сигнал с частотой повторения 1 Гц и в порт P3 загружается байт управления нагрузками для данного интервала времени. Предусмотрена подача короткого (длительностью 1 с) звукового сигнала в начале каждого часа и включение на 60 с звукового

сигнала с частотой повторения 1 Гц сразу после подачи напряжения питания на устройство.

Устройство может быть применено в тех случаях, когда необходимо получить эффект присутствия — включать и выключать электроприборы в отсутствие хозяев в квартире или просто включить бытовой электроприбор на определенный промежуток времени, а также для управления электроотопительными и электроосветительными приборами.

В интерфейс устройства (рис. 2) входит клавиатура (кнопки S1—S4), световая полоса HL1 и ЖК-индикатор HG1. Девять программируемых интервалов фактически определяют девять режимов работы устройства — «интервал 1», «интервал 2», ..., «интервал 9». Управление осуществляется с помощью клавиатуры, кнопки которой имеют следующее назначение:

- S1 (P) — выбор режима работы устройства в замкнутом цикле («интервал 1», «интервал 2», ..., «интервал 9»). После подачи питания устройство сразу переходит в режим «интервал 1», каждое нажатие данной кнопки переводит устройство в следующий режим, режимы работ переключаются в порядке возрастания, после «интервал 9» следует «интервал 1»;
- S2 (▲) — увеличение на единицу значения каждого разряда при ус-

тановке текущего времени часов и временных интервалов, а также принудительное выключение звукового и светового сигнала в начале каждого интервала;

- S3 (◀) — выбор разряда при установке значений во всех вышеуказанных режимах, у выбранного разряда устанавливается курсор, при каждом нажатии на данную кнопку курсор сдвигается справа налево на один разряд;
- S4 (▶) — аналогично S3, но курсор сдвигается слева направо.

В каждой строке ЖК-индикатора отображается 16 символов, номера разрядов на рис. 2 сверху отмечены цифрами. Разряды дисплея имеют следующее назначение:

Первая строка:

- разряд 1 — десятки часов текущего времени;
- разряд 2 — единицы часов;
- разряд 3 — символ «:» с периодом включения 1 с во всех режимах, во время корректировки или установки текущего времени символ включен постоянно;
- разряд 4 — десятки минут;
- разряд 5 — единицы минут;
- разряд 6 аналогичен разряду 3;
- разряд 7 — десятки секунд;
- разряд 8 — единицы секунд;
- разряды 9, 10 — пробелы;
- разряды 11—16 отображают состояние нагрузок 1—6 в текущем выбранном режиме: 1 — соответствующая нагрузка включена или будет включена, когда текущее время будет в границах отображаемого интервала, 0 — соответствующая нагрузка отключена.

Вторая строка:

- разряд 1 отображает текущий режим работы устройства, если устройство работает в режиме «интервал 1» в данном разряде индицируется 1, «интервал 2» — 2 и т. д.;
- разряд 2 — пробел;
- разряды 3—7 — начало интервала в отображаемом режиме работы устройства в часах и минутах (через символ «:»);
- разряд 8 — пробел;
- разряды 9—13 — конец интервала;
- разряд 14 — пробел;
- разряд 15 — флаг RAZ, разрешающий включение нагрузок во всех запрограммированных интервалах;
- разряд 16 — курсор.

После подачи питания устройство переходит в режим «интервал 1» (в

первом разряде второй строки дисплея индицируется «1»). Текущее время может быть установлено или откорректировано только в данном режиме. Для этого необходимо кнопками S3 или S4 подвести курсор к изменяемым разрядам текущего времени и кнопкой S2 изменить значение разряда. Для установки начального и конечного значений интервалов необходимо проделать такие же операции. Текущее время и флаг RAZ отображаются во всех режимах. Для программирования подключаемых нагрузок необходимо подвести курсор к нужному разряду и кнопкой S2 изменить его значение. Каждое нажатие кнопки S2 в данном случае инвертирует предыдущее состояние разряда: был 0, будет 1, и наоборот. После подачи питания при инициализации во все разряды 10—15 первой строки заносится 0 (нагрузки отключены). Для разрешения управления нагрузками необходимо флаг RAZ, отображаемый в разряде 15 второй строки на дисплее индикатора, установить в 1. Для этого нужно подвести кнопками S3 или S4 курсор к данному разряду и нажать кнопку S2, при этом каждое нажатие кнопки инвертирует предыдущее состояние разряда. Для оперативного выключения всех нагрузок, подключенных к устройству, необходимо данный разряд установить в нуль. Изменение режима работы (перебор программируемых интервалов времени), как уже упоминалось выше, осуществляется кнопкой S1.

Таким образом, устанавливая показание каждого разряда индикатора, можно оперативно выставить требуемое текущее время, границы интервалов и включаемые нагрузки в данных интервалах. При установке времени в режиме «интервал 1» запрещается отсчет текущего времени. Во всех остальных режимах отсчет текущего времени не запрещается. Изменить флаг RAZ можно только в режиме «интервал 1». Поэтому, например, для оперативного отключения нагрузок, чтобы сбросить флаг RAZ, необходимо войти в режим «интервал 1». Границы интервалов можно перепрограммировать. В приведенном на рис. 2 примере дисплей индикатора отображает текущее время — 12 час, 43 мин 56 с, в режиме работы «интервал 3» запрограммировано включение нагрузок 1, 2, 5, 6. Но нагрузки включены не будут, так как флаг RAZ не установлен. Алгоритм управления устройством получился достаточно простым, понятным и удобным.

Рассмотрим узлы принципиальной схемы (рис. 1). Основной устройством служит микроконтроллер DD1, рабочая частота которого задается генератором с внешним резонатором ZQ1 на 10 МГц.

Первый канал управления нагрузкой собран на симисторе VS1 и оптроне U1. Канал управляется с вывода 2 микроконтроллера DD1. Нагрузка подключается к соединителю X2, остальные — к X3—X7. Сигнал с вывода 13 микроконтроллера через резистор R4 с периодом 1 с включает световую полосу HL1. Микроконтроллер через порт P1 управляет ЖК-индикатором HG1 и клавиатурой (кнопки S1—S4). Резистор R10 — токоограничительный для световой полосы HL1. Как видно из схемы, выводы микроконтроллера DD1 исчерпаны полностью.

Цифровая часть устройства гальванически развязана от сети.

Программное обеспечение микроконтроллера полностью обеспечивает реализацию алгоритма работы устройства. Основная задача «часовой части» программы — формирование точных временных интервалов длительностью 1 с — решена с помощью прерываний от таймера TF0 и счетчика на регистре R0, который подсчитывает число прерываний. Когда число прерываний станет равным определенному значению, текущее время увеличивается на секунду. Корректировка текущего времени происходит каждый час.

Питающее напряжение поступает на плату с соединителя X8. Конденсатор C2 фильтрует пульсации в цепи питания +5 В.

ЛИТЕРАТУРА:

1. В. Б. Бродин, И. И. Шагулин. *Микроконтроллеры. Архитектура, программирование, интерфейс*. — М.: Издательство ЭКОМ, 1999.
2. С. Малахов. *Цифровой таймер на микроконтроллере*. — *Схемотехника*, 2004, № 2, с. 32—34.
3. С. Шишкин. *Часы-термометр*. — *Схемотехника*, 2002, № 9, с. 40—43.
4. А. В. Фрунзе. *Микроконтроллеры? Это же просто! Т. 3* — М.: ООО «ИД Скимен», 2003.

Окончание следует

Сергей Шишкин,
г. Саров Нижегородской обл.

Журнал «СХЕМОТЕХНИКА». Через каталог Агентства «Роспечать» (красный) — индекс 80724 (для РФ), 82177 (стран Балтии и СНГ). Через Объединенный каталог «Пресса России» (зеленый) — индекс 41733. Жители Украины могут подписаться через каталог агентства KSS (044-270-6220, 270-6222) — индекс 10540. Жители Латвии — через каталог агентства PKS (371-7320148).

Журнал «КОМПОНЕНТЫ И ТЕХНОЛОГИИ». Через каталог Агентства «Роспечать» (красный) — индекс 80743. Через Объединенный каталог «Пресса России» (зеленый) — индекс 41734. Жители Украины могут подписаться через каталог агентства KSS (044-270-6220, 270-6222) — индекс 10358. Жители Латвии — через каталог агентства PKS (371-7320148).

Оформить подписку с последующей доставкой можно также в любой стране через фирму МК-Periodica. Подробную информацию Вы можете узнать по телефону в Москве (7+095) 2819345, 2815715, 2813322, e-mail: info@periodicals.su

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 (095) 777-1215, e-mail: podpiska@dian.ru.
- вы можете также подписаться на журнал непосредственно в редакции по адресу: г. Москва, ул. Бутырская, д. 41/47 (метро «Савеловская», во дворе д. 17, вход с торца здания) и на специализированных выставках.

Стоимость подписки через редакцию:

Комплект журнала «Схемотехника» № 1—12 за 2005 г. (стоимость доставки включена) — 528 руб. (РФ)	Комплект журнала «Компоненты и Технологии» № 1—9 за 2005 г. (стоимость доставки включена) — 1080 руб
Комплект журнала «Схемотехника» № 1—12 за 2004 г. (стоимость доставки включена) — 504 руб. (РФ)	Комплект журнала «Компоненты и Технологии» № 1—9 за 2004 г. (стоимость доставки включена) — 900 руб. (РФ)
Комплект журнала «Схемотехника» № 4—7, 10—12 за 2003 г. (стоимость доставки включена) — 475 руб. 20 коп. (РФ)	Вы можете выписать отдельные номера журнала «Компоненты и Технологии» за 2000—2004 г. (стоимость доставки включена) — 100 руб. О наличии номеров узнавайте в редакции по тел. (095) 7771215, email: podpiska@dian.ru
Вы можете выписать отдельные номера журнала «Схемотехника» за 2001—2004 г. (стоимость доставки включена) — 42 руб. О наличии номеров узнавайте в редакции по тел. (095) 7771215, email: podpiska@dian.ru	

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

Извещение	ООО «ИД СКИМЕН» ИНН/КПП 7731195492/773101001 в ОАО КБ «НК банк» в г. Москва Р/с 407028102000000005646 К/с 301018109000000000278 БИК 044579278 Плательщик Адрес (с индексом)	
	Назначение платежа	Сумма
Кассир	Подпись “ ” _____ 2005 г.	
Квитанция	ООО «ИД СКИМЕН» ИНН/КПП 7731195492/773101001 в ОАО КБ «НК банк» в г. Москва Р/с 407028102000000005646 К/с 301018109000000000278 БИК 044579278 Плательщик Адрес (с индексом)	
	Назначение платежа	Сумма
Кассир	Подпись “ ” _____ 2005 г.	