

ETSF
NT
PRESS

ИВ МЕРЖИ

ПРАКТИЧЕСКОЕ РУКОВОДСТВО ПО ЛОГИЧЕСКИМ МИКРОСХЕМАМ И ЦИФРОВОЙ СХЕМОТЕХНИКЕ

- Основные логические элементы
- Схемотехника на основе логических элементов
- Программируемые элементы
- Программирование и программатор для микроконтроллера Basic Stamp 2
- Практические конструкции

**С ПОМОЩЬЮ
РЕГУЛЯТОРВ**

В помощь радиолюбителю

Ив Мержи

**ПРАКТИЧЕСКОЕ РУКОВОДСТВО
ПО ЛОГИЧЕСКИМ
МИКРОСХЕМАМ И ЦИФРОВОЙ
СХЕМОТЕХНИКЕ**

Yves Mergý

**Pour s'initier à
l'électronique logique et
numérique**

ETSF

Ив Мержи

**Практическое руководство
по логическим
микросхемам
и цифровой схемотехнике**

В помощь радиолюбителю

NT Press
Москва

УДК 004.312
ББК 32.85
М52

Подписано в печать 19.04.07. Формат 84x108¹/₃₂. Гарнитура New Baskerville. Печать офсетная. Усл. печ. л. 13,44. Тираж 1500 экз. Заказ № 6495.

М52 Мержи И.

Практическое руководство по логическим микросхемам и цифровой схемотехнике / Ив Мержи; пер. с англ. Ю. Соколов – М. : НТ Пресс, 2007. – 256 с. : ил. – (В помощь радиолюбителю).

ISBN 978-5-477-00265-8

Основным предметом рассмотрения в этой книге будут схемы, построенные на различных основах. Так, помимо схем на «жесткой логике», описываются схемы на программируемой логике и микроконтроллерах PIC фирмы Microchip, а также на менее известном у нас, но всемирно известном микроконтроллере BASIC STAMP фирмы Parallax (его еще называют кратко BS2).

Рассматриваются такие языки программирования, как QBasic и PBasic2; последний специально разрабатывался для BS2. Вы познакомитесь с понятием «булева алгебра» и без труда научитесь самостоятельно создавать интересные конструкции, начиная от самых простых и заканчивая наиболее сложными. В этом поможет доступное описание электронных схем, чертежей печатных плат, размещения радиоэлементов и соединения составных частей, листингов. Также в книге дана схема имитации присутствия людей в квартире, по случайному закону включающая/выключающая свет в квартире и многие другие схемы – на любой вкус. Для читателей, увлекающихся роботостроительством, предлагается простая и интересная конструкция робота.

Эта книга будет интересна как начинающим радиолюбителям, так радиолюбителям-«зубрам», уже собравшим не одну радиолюбительскую конструкцию. А так как она написана интересным и живым языком, то окажется доступной и полезной для студентов технических колледжей и вузов.

УДК 004.312
ББК 32.85

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельца авторских прав.

Материал, изложенный в данной книге, многократно проверен. Но, поскольку вероятность технических ошибок все равно остается, издательство не может гарантировать абсолютную точность и правильность приводимых сведений. В связи с этим издательство не несет ответственности за возможный ущерб любого вида, связанный с применением содержащихся здесь сведений.

ISBN 2-10-005329-9 (фр.)
ISBN 978-5-477-00265-8 (рус.)

© Yves Mergy, 2007
© Издание на русском языке,
перевод на русский язык,
оформление «НТ Пресс», 2007

СОДЕРЖАНИЕ

Скачивайте все данные этой книги из Интернета!	10
Введение	10
Назначение и содержание книги	11
Основы счисления	12
Часть I. Логическая электроника	21
Глава ■ 1	
Два основных семейства микросхем	22
Глава ■ 2	
Основные логические элементы	26
2.1. Логический элемент НЕ, или инвертор	26
2.2. Логический элемент ДА, или буферный повторитель	28
2.3. Логический элемент ИЛИ	29
2.4. Логический элемент ИЛИ-НЕ	31
2.5. Логический элемент И	33
2.6. Логический элемент И-НЕ	35
2.7. Логический элемент ИСКЛЮЧАЮЩЕЕ ИЛИ	36
2.8. Логический элемент ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ	38
2.9. Триггер Шмитта	40
2.10. Прибор для тестирования двухвходовых логических элементов	41

Глава ■ 3

Схемотехника на основе логических элементов 50

- 3.1. Краткое представление содержания главы 3 50
- 3.2. Мультивибраторы на логических элементах 50
- 3.3. Простой одновибратор с управлением от кнопки, генерирующий одиночный положительный импульс 56
- 3.4. Одновибратор с перезапуском 59
- 3.5. RS-триггер 60
- 3.6. D-триггер 62
- 3.7. Двоичные счетчики 65
- 3.8. Десятичный счетчик 70

Глава ■ 4

Логический имитатор на основе микросхем 72

- 4.1. Краткое представление содержания главы 4 72
- 4.2. Принципиальная схема логического имитатора 73
- 4.3. Сборка логического имитатора 76

Глава ■ 5

Программный имитатор на РС 81

- 5.1. Краткое представление содержания главы 5 81
- 5.2. Принципиальная схема программного имитатора 82
- 5.3. Сборка 85

Часть II. Цифровая электроника 92

Глава ■ 6

Программируемые элементы 93

- 6.1. Запоминающие устройства 93
- 6.2. Другие программируемые схемы 98
- 6.3. Микроконтроллер 98

Глава ■ 7

Выбор микроконтроллера 100

Глава ■ 8

Характеристики микроконтроллера BASIC STAMP 2 .. 103

Глава ■ 9

Программирование микроконтроллера BASIC STAMP 2 104

- 9.1. Программное обеспечение 104
- 9.2. Язык программирования PBASIC2 106

Глава ■ 10

Программатор BASIC STAMP 2 на основе элементов в корпусе DIP	125
10.1. Представление	125
10.2. Принципиальная схема	126
10.3. Реализация	128

Глава ■ 11

Программатор BS2 и плата тестирования	133
11.1. Краткое представление содержания главы 11	133
11.2. Принципиальная схема программатора BS2 и платы тестирования	134
11.3. Сборка	138
11.4. Программирование	142

Часть III. От теории к практике 146**Глава ■ 12**

Охранная система	147
12.1. Общее описание	147
12.2. Схема на основе цифровой логики	148
12.3. Схема охранной сигнализации на основе программируемой логики	159
12.4. Схема охранной сигнализации на основе микроконтроллера	169

Глава ■ 13

«Умный» имитатор присутствия	178
13.1. Общее описание	178
13.2. Схема на основе программируемой логики	179
13.3. Схема на основе микроконтроллера BS2	190

Глава ■ 14

Мобильный автономный робот	200
14.1. Общее описание	200
14.2. Сборка робота	201
14.3. Рекомендации по работе устройства	227
14.4. Заключительное слово	228

Приложение ■ 1

Логический зонд	229
Универсальный источник питания	229
Логический зонд	229
Универсальный источник питания	234

Приложение ■ 2

**Технология изготовления печатных плат методом
литографии 240**

Приложение ■ 3

Расположение контактов элементов 243

Справочные материалы 248

 Библиография 248

 Адреса интернет-сайтов 249

 Полезные адреса 250

Предметный указатель 251

Прежде всего я хочу сказать, что эту книгу я посвящаю моей замечательной жене. Благодарю ее за участие в написании книги, а также за ее постоянное терпение и понимание проблем, с которыми приходится сталкиваться на литературно-техническом поприще. А также моим преданным друзьям Бернарду Фишьера (Bernard Fichiera), Паскалю Майо (Pascal Mayeux) и Патрису Огику (Patrice Ogiic), без моральной поддержки которых вряд ли я написал бы эту книгу.

Скачивайте все данные этой книги из Интернета!

Вы можете загрузить все программы и схемы печатных плат этой книги, а также программу BASIC STAMP для программирования микроконтроллеров с французского сайта <http://www.dunod.com>.

Сначала нужно найти книгу (по названию или по автору), затем щелкнуть один раз по надписи **Дополнения в реальном времени** (Complements en ligne) в разделе **Дополнительные ресурсы** (Resources complementaires) на странице, посвященной данной книге. Чтобы зайти в этот раздел, потребуются ввести пароль, соответствующий данной книге.

Вы можете легко распечатать рисунки печатных плат этой книги на прозрачные пленки. Наилучшее качество будет получено при использовании лазерных принтеров. Благодаря точности их печати, можно получить исключительное качество образцов. При использовании струйных принтеров следует тщательно проверить отпечатанную схему на наличие замыканий дорожек между собой, так как микрокапли чернил часто попадают на дорожки схем.

В качестве альтернативного варианта можно посоветовать распечатать сначала схему на бумаге, а затем снять с нее фотокопию на прозрачной пленке. В этом случае нужно изготовить две копии и склеить их для того, чтобы дорожки схемы не оказались прозрачными ультрафиолетовому излучению при экспонировании.

Введение

Назначение и содержание книги

Эта книга предназначена всем, кто желает изучить логику работы цифровых электронных схем и получить базовые знания в этой области. Читатели найдут в книге простые теоретические выкладки, но прежде всего книга представляет интерес с практической стороны. Для написания программ мы применили наиболее распространенный и известный язык программирования *QBASIC*. Этот язык применяется на всех старых компьютерах, в образовании и легко переписывается на другой язык.

Во введении к книге рассматриваются основы счета, необходимые для изучения логической и цифровой электроники. Первая глава посвящается цифровой логике, подробному описанию элементов с таблицами истинности, диаграммами функционирования, примерами программирования и типовыми схемами для изучения наиболее употребительных свойств. Тестовая плата и два симулятора позволяют вам осуществить все необходимые эксперименты.

Вторая глава более подробно останавливается на изучении цифровой электроники. В качестве программируемого микроконтроллера используется современный элемент BASIC STAMP2 фирмы Parallax^{*}, версия которого позволяет использовать элементы поверхностного монтажа и радиоэлементы с двухрядным расположением выводов (DIL - Dual-In-Line), к которым могут относиться не только микросхемы, но и наборы резисторов, диодов и т. д. Методы программирования подробно описаны. Что касается программы, то ее можно бесплатно скачать с сайта в Интернете.

^{*} Parallax является торговой маркой фирмы Parallax Inc. Продукты фирмы выпускаются в США и других странах. - Прим. науч. ред.

С целью закрепления теоретических знаний в третьей главе книги изучаются некоторые практические схемы, использующие полученные теоретические знания: цифровую логику, цифровые программаторы для программирования персональных компьютеров и микроконтроллеров. В любом случае наши схемы можно применить всем, так как они изготавливаются из доступных в продаже и недорогих элементов.

В приложении 1 к книге вы найдете описание логического зонда и схемы блока питания с несколькими выходами. Также в конце книги прилагаются дополнительные материалы для облегчения практических работ и исследований: цоколевка элементов, библиография, ссылки на адреса сайтов в Интернете и многое другое.

Все изученные программы, данные в рамках этого издания и необходимые для изготовления схем, можно загрузить с сайта издателя в Интернете: <http://www.dunod.com>.

Мы надеемся, что книга удовлетворит все ваши ожидания и станет ценным пособием в работе.

Основы счисления

Мы начинаем считать с детства и при счете пользуемся десятичной системой счисления. Чтобы начать изучение электроники, необходимо изучить две основные системы счисления.

Десятичная система счисления

Вспомним некоторые хорошо известные понятия, которые позволят сопоставить десятичную систему с другими системами счисления.

В десятичной системе счет идет от 0 до 9, это система с основанием 10, так как мы используем 10 цифр для счета. Первой цифрой является не 1, а 0. Когда счет доходит до 9, он переходит снова к нулю, при этом разряд десятков увеличивается на 1. Далее снова счет продолжается до 19, после чего следует снова увеличить разряд десятков на 1, чтобы получить число 20. Покажем общий принцип счета, не продолжая это простое объяснение.

0	1	2	3	4	5	6	7	8	9 <ос- нование 10
10	11	12	1	14	1	16	17	18	19
20	21...								

Отсюда можно сделать вывод, что каждый разряд числа представляет собой степень числа 10:

10^0 = единицы	1	например: $9 \times 10^0 = 9$
10^1 = десятки	10	например: $5 \times 10^1 = 50$
10^2 = сотни	100	например: $3 \times 10^2 = 300$
10^3 = тысячи	1000	например: $1 \text{ т } 10^3 = 1000$

и т. д.

В десятичной системе суммарное число единиц, десятков, сотен, тысяч из предыдущего примера будет равно:

$$1359_{10} = 1 \times 10^3 + 3 \times 10^2 + 5 \times 10^1 + 9 \times 10^0 = \\ = 1000 + 300 + 50 + 9$$

Этот способ представления чисел нам знаком из арифметики, а две другие системы счисления, рассматриваемые здесь, также удовлетворяют тем же правилам.

Двоичная система счисления

В двоичной системе счисления (основание системы является не 10, а 2) есть только две цифры – это 0 и 1, которые позволяют считать до бесконечности*.

Выведем соотношение между системой с основанием 2 – двоичной – и системой с основанием 10 – десятичной (табл. 1). Каждая цифра двоичного числа называется *битом*. Каждый бит двоичного числа представляет собой степень числа 2, а не 10. Принцип построения двоичного числа достаточно простой, но двоичные числа занимают больше места и требуют большого терпения при расчетах. Однако, несмотря на все

* В индийском эпосе приведен пример мудрости – изобретателю шахмат индийский раджа предложил выбрать подарок. Изобретатель попросил на каждую следующую клетку шахмат (клеток на шахматном поле 64) положить (начиная с одного зерна на первой клетке) в 2 раза большее количество зерна, чем на предыдущую клетку. Раджа улыбнулся, как ему показалось, глупости изобретателя и пообещал. Мудрец сообщил радже о невозможности дать столько зерна изобретателю шахмат. Это число будет 2^{63} и всего имеющегося количества зерна будет намного меньше даже в масштабе человечества. Желющие могут посчитать в десятичной системе счисления, сколько будет 2^{63} . Это пример 2^{63} , и степень числа относительно небольшая. Так что считать до бесконечности лучше не пробовать. – *Прим. науч. ред.*

неудобства, эта система счисления является основной в цифровой электронике, так как цифровые приборы принимают в большинстве случаев два рабочих состояния: 0 и 1*.

Таблица 1. Представление степени числа два каждого бита в двоичной системе счисления

Разряд	8-й бит Старший бит	7-й бит	6-й бит	5-й бит	4-й бит	3-й бит	2-й бит	1-й бит Младший бит
Степени числа 2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Десятичный эквивалент	128	64	32	16	8	4	2	1

Обозначение старшинства бита важно: первый бит – крайний справа – является младшим, последний – крайний слева – старшим. Так, в следующем примере старший бит – пятый. Порядок счета в двоичных и десятичных кодах идентичный за исключением основания числа. В десятичных кодах – основание 10, в двоичных – 2.

Идентичным будет также позиция цифры относительно других цифр: слева направо – от самой старшей цифры к самой младшей – по их весовым коэффициентам первых четырех разрядов числа. *Двоичный код* называют также *кодом 8421* в соответствии с весовыми коэффициентами первых четырех разрядов числа. Кроме того, его называют *прямым двоичным кодом*. Далее рассматривается пример соответствия двоичного кода 11010_2 и его десятичного эквивалента – 26_{10} .

Бинарное число: 1 1 0 1 0

Соответствие между двоичным и десятичным числом:

. - - - - -> $(0 \times 1) = 00$ -> младший разряд
 - - - - -> $(1 \times 2) = 02$
 - - - - -> $(0 \times 4) = 00$
 - - - - -> $(1 \times 8) = 08$
 - - - - -> $(1 \times 16) = 16$ -> старший разряд

 Общая сумма: = 26

* Имеются цифровые схемы с тремя состояниями: 0, 1 и Z (с большим сопротивлением или иначе называемым высокоимпедансным состоянием). – Прим. науч. ред.

Точно так же, как и с десятичными числами, с двоичными можно производить арифметические операции.

Правило сложения

$$0 + 0 = 0$$

$$1 + 0 = 1$$

$1 + 1 = 10$ Результат этой операции равен «нулю», а не десяти, как можно подумать (то есть 0 и 1 в уме – 1 переносится в следующий (старший) разряд).

Пример:	$\begin{array}{r} 11010 \\ + 00010 \\ \hline 11100 \end{array}$	$\begin{array}{r} 26 \\ + 2 \\ \hline 28 \end{array}$
	==>	

Правило умножения

$$0 \times 0 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

Пример:	$\begin{array}{r} 11010 \\ \times 00010 \\ \hline 00000 \\ 11010. \\ \hline 110100 \end{array}$	$\begin{array}{r} 26 \\ \times 2 \\ \hline 52 \end{array}$
	==>	
	(сдвиг влево на одну позицию относительно множителя)	
	==> эквивалент 52 в десятичной системе счисления	

Дополнение до 1 или обратный двоичный код числа

Дополнение до 1 – простая операция: достаточно проинвертировать все биты, составляющие прямой двоичный код числа, то есть все единицы заменить на нули и наоборот. В результате этой операции образуется *обратный двоичный код числа*.

Эта операция необходима, чтобы ввести операцию дополнения до 2. Затем эти две операции будут использоваться при описании операции вычитания.

Пример:

Давайте возьмем число	110100	(Эквивалент 52 в десятичной системе счисления)
Дополнение до 1 этого числа	001011	(Эквивалент 11 в десятичной системе счисления)

Дополнение до 2 или дополнительный двоичный код числа

Чтобы вычислить дополнение до 2, необходимо к дополнению до 1 прибавить 1.

Таким образом, дополнительный двоичный код числа образуется из обратного кода при добавлении 1 к младшему разряду числа.

Пример:

Давайте возьмем дополнение до 1	001011	(Эквивалент 11 в десятичной системе счисления)
---------------------------------	--------	---

Прибавим 1 к младшему разряду	1

Мы получаем 001100
Эквивалент 12 в десятичной системе счисления образуется путем сложения двух двоичных чисел, аналогично двум десятичным!

Для двоичных чисел выполняется следующее правило: если само число сложить с дополнением к 2 для этого числа, то в результате будет нуль. При этом следует сделать замечание, что единица остается в уме и переходит в старший разряд.

Пример:

Число	110100	(Эквивалент 52 в десятичной системе счисления)
-------	--------	---

Дополнение до 2 предыдущего примера + 001100 (Эквивалент 12 в десятичной системе счисления)

(1)000000 (Эквивалент 64 в десятичной системе счисления)

Мы получаем в результате 0 во всех 6 разрядах (с 0 по 5-ый) и перенос 1 в шестой разряд - так мы сложили 2 двоичных числа, аналогично двум десятичным!

Обратим внимание, как это видно из таблицы 1, счет разрядов идет с нулевого, а не с первого разряда!

Правило вычитания

Чтобы вычесть из одного двоичного числа другое, следует прибавить дополнение до 2, вычисленное для вычитаемого, к уменьшаемому.

Например, пусть из числа 111011 (равно 59 в десятичной системе счисления) нужно вычесть число 110100 (52 в десятичной системе счисления). Для этого прибавим к первому числу число 001100, которое является дополнением до 2 для числа 110100. Операцию вычитания мы заменили на операцию сложения с дополнением до 2 вычитаемого.

Операция:	111011	
	+	001100

Нам дает:	=	(1)000111

Для двоичных чисел нужно всегда очень точно считать число битов. Так, оставляя 1 в уме в старшем бите, получим в результате вычитания число 7 в десятичном исчислении.

Шестнадцатеричная система счисления

В шестнадцатеричной системе счетными являются числа от 0 до 15; как ясно из названия, основанием системы является число 16, так как для счета используется 16 характерных знаков.

Так как для описания всех счетных чисел не хватало цифр, то наиболее простым решением посчитали использовать буквы. Так, число 10 заменяется буквой «А», 11 – «В», 12 – «С», 13 – «D», 14 – «Е», 15 – «F». Когда счет доходит до 15, происходит опять переход к 0, при этом старший разряд увеличивается на 1.

Такая нумерация очень близка десятичной системе, следует лишь поменять основание 10 на 16. Шестнадцатеричная система счисления нашла широкое применения в программировании, вычисления в этой системе занимают меньше места. Каждая цифра шестнадцатеричного числа может быть разложена на двоичное число из четырех битов, которое называют *тетрадой*, или для облегчения расчетов сразу в десятичное число.

Пример:

Возьмем число 4CF в шестнадцатеричной системе счисления	
' ' '→	1111 = 15 × 16 ⁰ = 15 × 1 = 15
' '→	1100 = 12 × 16 ¹ = 12 × 16 = + 192
'→	0100 = 04 × 16 ² = 04 × 16 × 16 = + 1024
	= 1231

Эквивалент шестнадцатеричного числа 4CF в десятичном счислении равен 1231.

Обобщение

Подытожим и запишем в таблицу все, что мы изучили, производя соответствие между тремя системами счисления.

Очевидно, что эта таблица неполная, так как чисел в природе бесконечно. В таблице представлены числа от 0 до 36 (в десятичной системе) и затем от 247 до 255, чтобы показать пример больших чисел (табл. 2).

В качестве заключения по системам счисления покажем присущее обозначение присвоения номеров каждому типу системы счисления при программировании.

Таблица 2. Представление чисел в десятичной, двоичной и шестнадцатеричной системах счисления

Деся- тич- ная	Двоич- ная	16- рич- ная	Деся- тич- ная	Двоич- ная	16- рич- ная	Деся- тич- ная	Двоич- ная	16- рич- ная
0	0000	0	16	10000	10	32	100000	20
1	0001	1	17	10001	11	33	100001	21
2	0010	2	18	10010	12	34	100010	22
3	0011	3	19	10011	13	35	100011	23
4	0100	4	20	10100	14	36	100100	24
5	0101	5	21	10101	15			
6	0110	6	22	10110	16			
7	0111	7	23	10111	17	247	0111	F7
8	1000	8	24	11000	18	248	1000	F8
9	1001	9	25	11001	19	249	1001	F9
10	1010	A	26	11010	1A	250	1010	FA
11	1011	B	27	11011	1B	251	1011	FB
12	1100	C	28	11100	1C	252	1100	FC
13	1101	D	29	11101	1D	253	1101	FD
14	1110	E	30	11110	1E	254	1110	FE
15	1111	F	31	11111	1F	256	1111	FF


- В десятичной системе счисления числу не предшествует никакой особый знак.
Например: 1358;
- в двоичной системе счисления перед числом ставится знак %.
Например: %10101001110;
- в шестнадцатеричной системе счисления перед числом ставится знак \$.
Например: \$54E.

Существуют также другие системы счисления: восьмеричная, (BCD – Binary-Coded Decimal). Мы их здесь не изучаем, чтобы не сильно загружать книгу. Читатели, желающие углубить свои знания в этой области, могут обратиться к книгам, приведенным в библиографии.

Соглашения, принятые в книге

В тексте выделены следующие обозначения:

- *курсивом* выделяются новые термины, на которые дается ссылка в предметном указателе;

 под этой пиктограммой приводятся разного рода примечания, на которые стоит обратить внимание;

 под этой пиктограммой приводятся замечания.

Часть



Логическая электроника

Рассмотрим кратко построение первой части книги. Вначале вы познакомитесь с логической электроникой, которая изучает интегральные микросхемы, позволяющие физически соединить логические функции. Мы будем изучать схемы на «жесткой логике». Начнем изучение с описания двух больших семейств логических схем. После этого перейдем к более детальному изучению основных логических элементов. Мы вам предложим изучение тестового стенда, на котором вы сможете тестировать и анализировать работоспособность схем КМОП с четырьмя входами, также рассмотренных ранее в этой части.

Далее мы перейдем к изучению более сложных комбинационных логических схем, без изучения которых трудно рассматривать практические приложения цифровой логики. В конце первой части мы подробно опишем работу двух логических модельных схем. Первая схема – непрограммируемая. Она позволяет изучить восемь основных функций. Вторая схема является программируемой, и ее производительность гораздо выше.

Цифровая электроника, описывающая программируемые элементы, будет описана во второй части книги.

Глава 1

Два основных семейства микросхем

Интегральные логические микросхемы разделяются на два основных логических семейства (серии): элементы на базе технологии ТТЛ (транзисторно-транзисторная логика) и элементы на базе технологии КМОП (комплиментарная МОП (металл-окисел-полупроводник)-технология).

Для ясности изложения уточним, что в своих опытах мы используем технологию КМОП-микросхем серии CD4000, что связано с большим диапазоном напряжений питания, недорогой ценой и особенно со стандартной цоколевкой выводов элементов. Тем не менее в этой главе мы рассмотрим их основные характеристики и разницу между этими двумя технологиями. Вы сможете легко выбрать самостоятельно элементы для своих схем.

Различия между двумя семействами цифровых микросхем начинаются с рассмотрения технологии их изготовления: ТТЛ-технология основана на применении биполярных транзисторов, в то время как КМОП-технология использует транзисторную технологию на основе полевых транзисторов структуры МОП (металл-окисел-полупроводник), как следует из ее названия.

Ниже приведен список серий схем и их основные характеристики:

- SN74, ТТЛ, стандартная;
- SN74(H), ТТЛ, быстродействующая;

- SN74(L), ТТЛ, маломощная;
- SN74(S), ТТЛ, с диодами Шоттки;
- SN74(LS), ТТЛ, с диодами Шоттки маломощная;
- SN74(AS), ТТЛ, с диодами Шоттки усовершенствованная;
- SN74(ALS), ТТЛ, с диодами Шоттки маломощная усовершенствованная;
- SN74(C), КМОП, с металлическим затвором, цоколевка, совместимая с ТТЛ;
- SN74(HC), КМОП, МДП-структура, быстродействующая цоколевка, совместимая с ТТЛ;
- SN74(HCT), КМОП, МДП-структура, напряжения и цоколевка, совместимые с ТТЛ;
- CD4000, КМОП, с металлическим затвором.

Обозначение логических входов стандартизовано с 1984 года европейским стандартом IEC/ANSI 91-1984. Схемы этой книги удовлетворяют этому стандарту. Но при перечислении основных логических элементов мы познакомим вас также и со старым обозначением элементов, которые применялись до 1984 года.

Для правильного понимания работы логических схем следует запомнить их терминологию. Логические схемы принимают два логических состояния: 0 и 1. Логическим уровням соответствуют определенные диапазоны напряжений. Вход схемы, соединенный с положительным потенциалом, называется «высоким уровнем напряжения», или логической «единицей». Вход схемы, соединенный с «землей», называется «низким уровнем напряжения», или логическим «нулем». Если вход схемы не соединен ни с «землей», ни с высоким уровнем напряжения и имеет потенциал среднего напряжения, он называется «плавающим». Неподключенный вход схемы – источник помех и неустойчивой работы, а в отдельных случаях и отказов в работе. Поэтому незадействованные входы обязательно подключайте к разрешающим потенциалам: напрямую к «земле», к $+V_{\text{питания}}$ (для КМОП-схем) или через резистор номиналом от 1 кОм к $+V_{\text{питания}}$ (для ТТЛ-схем)*. Последний вариант не рекомендуется и должен быть временным

* Другой вариант – объединить входы, но при этом понизится быстродействие цифровой микросхемы. – *Прим. науч. ред.*

состоянием. У каждой из этих технологий, будь то ТТЛ или КМОП, диапазоны напряжений, определяющие эти уровни, различны (рис. 1.1). Важно иметь это в виду при совместном использовании двух данных технологий.

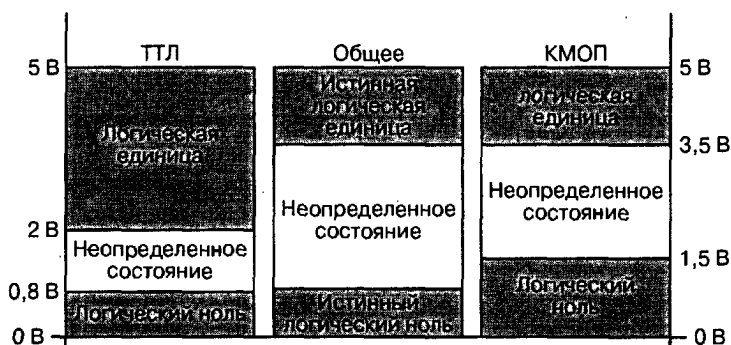


Рис. 1.1. Определение диапазонов входных напряжений разных семейств микросхем

В конце этой главы изучим четыре основных параметра, по которым отличаются технологии логических элементов (табл. 1.1). Читатели, желающие познакомиться более подробно с характеристиками логических схем, могут изучить книги, приведенные в библиографии в конце книги.

Таблица 1.1. Характеристики семейств логических микросхем

Технология	Серия микросхем	Напряжение питания	Частота, МГц	Кэфф. разветвления по выходу	Задержка распространения сигнала, нс
ТТЛ	74	5В ± 5%	35	10	9
	74(H)	5В ± 5%	50	10	6
	74(L)	5В ± 5%	3	20	33
	74(S)	5В ± 5%	125	20	3
	74(LS)	5В ± 5%	45	20	9,5
	74(AS)	5В ± 10%	200	40	1,7
	74(ALS)	5В ± 10%	70	20	4
КМОП	4000	3–15 В	12	50	50
	74(C)	3–15 В	40	>100	8
	74(HC)	3–15 В	40	>100	8
	74(HCT)	5В ± 10%	40	>100	8

- питание: диапазон или максимальное напряжение питания схемы;
- частота: максимальная рабочая частота схемы при напряжении 5 В;
- коэффициент разветвления по выходу (коэффициент нагрузки): максимальное количество входов элементов одного типа, которые могут нагружать выходы микросхемы без нарушения ее нормальной работы;
- время задержки распространения сигнала: интервал времени между входным и выходным импульсом, измеренными на уровне 0,5 от напряжения 5 В.

Глава **2**

Основные логические элементы

2.1. Логический элемент НЕ, или инвертор

Логический элемент НЕ (по-английски NOT) служит для изменения полярности сигнала. Инвертор реализует дополнение логического состояния. Логическая единица на входе инвертора преобразуется на выходе в нуль, а логический нуль на входе – в логическую единицу на выходе (табл. 2.1).

Таблица 2.1. Таблица истинности инвертора

Вход А	Выход S
0	1
1	0

Функцию инвертора можно также реализовать и с помощью других элементов, если они не все используются в схеме. Можно многовходовые логические схемы с инвертированием выхода преобразовать в инвертор двумя способами: первый – объединить все входы. Например, входы А и В объединяют, и схема превращается в инвертор. Второй способ – все входы, кроме первого, подключить к разрешающему потенциалу напряжения. Разрешающий потенциал будет различным в зависимости от семейства микросхем и функции интегральной микросхемы.

На рис. 2.1 показаны условные обозначения инвертора на схемах, а на рис. 2.2 показаны временные диаграммы, поясняющие работу инвертора.

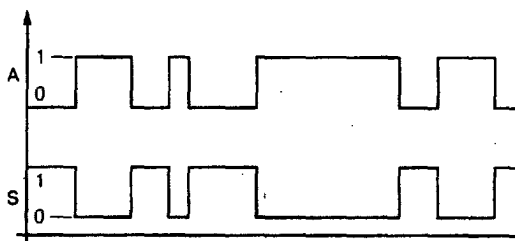


Рис. 2.2. Временная диаграмма, поясняющая работу инвертора

Используемые микросхемы в качестве инверторов

- CD4069, включает 6 инверторов КМОП;
- CD4049, включает 6 инверторов КМОП с повышенной нагрузочной способностью, чем другие КМОП, логические микросхемы и может служить для согласования с ТТЛ-микросхемами при напряжении +5 В.



Цоколевка микросхемы отличается от цоколевки инверторов серии 4000;

- SN7404, включает 6 инверторов ТТЛ;
- SN7606, включает 6 мощных инверторов ТТЛ с открытым коллектором.

Программирование

Команда NOT в QBASIC определяется как инверсия.

Пример 1 (логическая функция):

```

A% = 51
B% = NOT A%
PRINT B% 'На экране появится число -51, в Бейсике
          'инвертирован выход 51

```

Пример 2 (булев оператор):

```

НАЧАЛО:
A% = 8
INPUT "Введите цифру";B%
IF NOT(A%=B%) THEN PRINT "Не равны" ELSE PRINT "Равны"
GOTO НАЧАЛО

```

В зависимости от того, введенная цифра равна 8 или нет, на экране появляется надпись «Равны» или «Не равны».

2.2. Логический элемент ДА, или буферный повторитель

Логический элемент ДА (по-английски BUFFER) не изменяет полярность сигнала. Выход элемента является отражением входа (табл. 2.2). Такой элемент служит для восстановления формы сигнала или, например, как защита информационной шины. Два последовательно соединенных инвертора можно использовать вместо логического элемента ДА; это немного увеличит время распространения сигнала, но в большинстве случаев это не важно.

На рис. 2.3 показаны условные обозначения элемента на схемах, а на рис. 2.4 показаны временные диаграммы, поясняющие работу логического элемента ДА.

Таблица 2.2. Таблица истинности логического элемента ДА

Вход А	Выход S
0	0
1	1

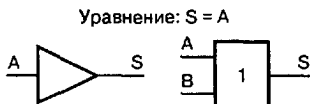


Рис. 2.3. Условные обозначения логического элемента ДА и математическая формула работы инвертора

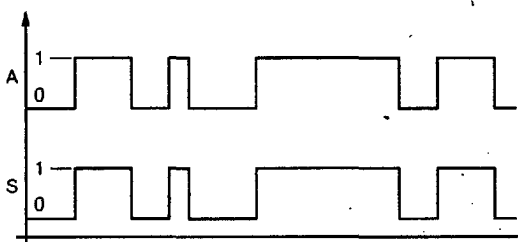


Рис. 2.4. Временная диаграмма для пояснения работы логического элемента ДА

Используемые микросхемы в качестве буферного повторителя

- CD4050, включает 6 усиленных буферных повторителей КМОП и обладает повышенной нагрузочной способностью, чем другие схемы.



Цоколевка микросхемы отличается от цоколевки аналогичной микросхемы серии 4000;

- SN7407, включает 6 мощных буферных повторителей ТТЛ.

Программирование

Команда OUI в QBASIC определяется как равенство. Ввиду того, что эта команда чрезвычайно проста, мы не будем приводить пример. Пример 2 в предыдущем разделе можно привести и здесь, если поменять условие с отрицания на равенство и поменять местами выводимые на экран надписи.

2.3. Логический элемент ИЛИ

Выход *S* логического элемента *ИЛИ* (также его называют *ИЛИ* включающее) по-английски OR, переходит в состояние логической единицы лишь в том случае, если хотя бы один из входов *A* или *B* равен 1 или оба входа одновременно равны 1 (табл. 2.3).

На рис. 2.5 показаны условные обозначения логического элемента ИЛИ на схемах, а на рис. 2.6 показаны временные диаграммы, поясняющие его работу.

Таблица 2.3. Таблица истинности логического элемента ИЛИ

Вход А	Вход В	Выход S
0	0	0
1	0	1
0	1	1
1	1	1

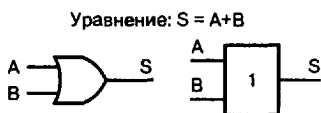


Рис. 2.5. Условные обозначения логического элемента ИЛИ и математическая формула работы элементов

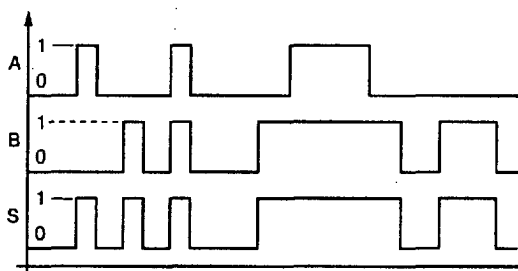


Рис. 2.6. Временная диаграмма, поясняющая работу логического элемента ИЛИ

Используемые микросхемы:

- CD4071, включает 4 логических элемента (ЛЭ) 2ИЛИ технологии КМОП (с 2 входами каждого их четырех ЛЭ);
- CD4075, включает 3 логических элемента 3ИЛИ технологии КМОП (с 3 входами каждого их трех ЛЭ);
- CD4072, включает 2 логических элемента 4ИЛИ технологии КМОП (с 4 входами каждого их двух ЛЭ);
- SN7432, включает 4 логических элемента 2ИЛИ технологии ТТЛ (с 4 входами каждого их двух ЛЭ).

Программирование

Команда OR в QBASIC существует в виде функции и булева оператора. В первом случае программа переводит все нулевые биты в 1. В случае использования булева оператора, программа сравнивает 2 условия и выводит действительный результат, если одно или оба условия выполняются.

Пример 1 (функция):

```
A%=13           'Или 1101 в двоичной форме
B%=6            'Или 0110 в двоичной форме
PRINT A% OR B% 'Результат: 15 в десятичной форме или
                '1111 в двоичной форме
```

Пример 2 (булев оператор):

```
НАЧАЛО:
INPUT "Введите цифру";A%
INPUT "Введите еще одну цифру";B%
IF A%=4 OR B%=9 THEN PRINT "Правильно!" ELSE PRINT "Не
правильно!"
GOTO НАЧАЛО
```

На экране компьютера появится поздравительная надпись, если вы присвоите переменной A% значение 4, либо переменной B% значение 9, либо одновременно присвоите данным переменным правильные значения.

2.4. Логический элемент ИЛИ-НЕ

Выход «S» логического элемента ИЛИ-НЕ, по-английски NOR, переходит в состояние логического нуля в том случае, если хотя бы на одном из входов «А» или «В» или на обоих одновременно будет 1 (табл. 2.4). Работа этого элемента противоположна работе элементу ИЛИ.

Таблица 2.4. Таблица истинности логического элемента ИЛИ-НЕ

Вход А	Вход В	Выход S
0	0	1
1	0	0
0	1	0
1	1	0

Этот элемент (ИЛИ-НЕ) можно составить из двух логических элементов, соединив последовательно выход элемента ИЛИ с входом инвертора.

На рис. 2.7 показаны условные обозначения логического элемента ИЛИ-НЕ на схемах, а на рис. 2.8 показаны временные диаграммы, поясняющие его работу.

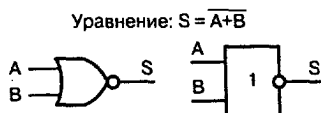


Рис. 2.7. Условные обозначения логического элемента ИЛИ-НЕ и математическая формула работы элемента

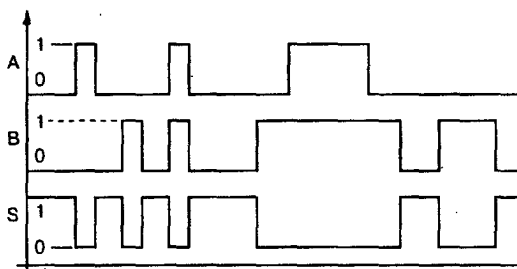


Рис. 2.8. Временная диаграмма, поясняющая работу логического элемента ИЛИ-НЕ

Используемые микросхемы:

- CD4001, включает 4 логических элемента (ЛЭ) 2ИЛИ-НЕ технологии КМОП (с 2 входами каждого их четырех ЛЭ);
- CD4025, включает 3 логических элемента 3ИЛИ-НЕ технологии КМОП (с 3 входами каждого их трех ЛЭ);
- CD4002, включает 2 логических элемента 4ИЛИ-НЕ технологии КМОП (с 4 входами каждого их двух ЛЭ);
- CD4078, включает 1 элемент 8ИЛИ-НЕ технологии КМОП (с 8 входами одного ЛЭ);
- SN7402, включает 4 элемента 2ИЛИ-НЕ технологии ТТЛ (с 2 входами каждого их четырех ЛЭ);
- SN7427, включает 3 элемента 3ИЛИ-НЕ технологии ТТЛ (с 3 входами каждого их трех ЛЭ);
- SN7428, включает 2 элемента 4ИЛИ-НЕ технологии ТТЛ (с 4 входами каждого их двух ЛЭ).

Программирование

Элемент ИЛИ-НЕ широко применяется в логической электронике и практически не применяется в программировании. В QBASIC не существует команды, соответствующей выполнению функции ИЛИ-НЕ. Для создания такой функции в программировании достаточно проинвертировать условия или результат, соответствующие команде ИЛИ.

2.5. Логический элемент И

Выход «S» логического элемента И, по-английски AND, переходит в состояние логической единицы в том случае, если одновременно оба входа «А» и «В» равны 1. Если хотя бы один из входов равен нулю, на выходе элемента также будет нуль (табл. 2.5).

Таблица 2.5. Таблица истинности логического элемента И

Вход А	Вход В	Выход S
0	0	0
1	0	0
0	1	0
1	1	1

На рис. 2.9 показаны условные обозначения логического элемента И на схемах, а на рис. 2.10 показаны временные диаграммы, поясняющие его работу.

Рис. 2.9. Условные обозначения логического элемента И и математическая формула работы элемента

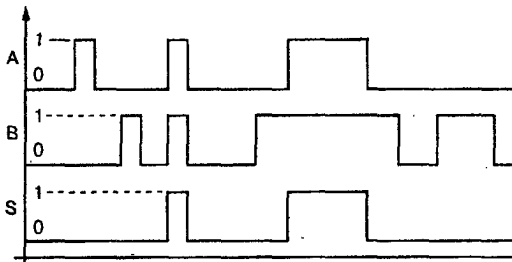


Рис. 2.10. Временная диаграмма, поясняющая работу логического элемента И

Используемые микросхемы:

- CD4081, включает 4 2-входных логических элемента И технологии КМОП;
- CD4073, включает 3 3-входных логических элемента И технологии КМОП;
- CD4082, включает 2 4-входных логических элемента И технологии КМОП;
- SN7408, включает 4 2-входных логических элемента И технологии ТТЛ;
- SN7411, включает 3 3-входных логических элемента И технологии ТТЛ;
- SN7421, включает 2 4-входных логических элемента И технологии ТТЛ.

Программирование

Команда AND в QBASIC представлена в виде функции и булева оператора. В первом случае команда AND используется для создания маски; это прием, который позволяет считывать только нужные биты.

В случае использования булева оператора AND в программе сравниваются два условия, правильный результат получается, только если выполняются оба условия.

Пример 1 (функция):

```
A%=13           'Или 1101 в двоичной форме
B%=6            'Или 0110 в двоичной форме (B% - маска)
PRINT A% AND B% 'Результат: 4 в десятичной форме
                'или 0100 в двоичной форме
```

Мы наложили маску на первый и второй биты числа B%, не используя нулевой и третий; результат показывает, что первый бит числа A% равен нулю, а второй – единице.

Пример 2 (булев оператор):

```
НАЧАЛО:
INPUT "Введите цифру";A%
INPUT "Введите еще одну цифру";B%
IF A%=4 AND B%=9 THEN PRINT "Правильно!" ELSE PRINT "Не
правильно!"
GOTO НАЧАЛО
```

На экране компьютера появится поздравительная надпись, если вы присвоите одновременно переменной A% значение 4, а переменной B% значение 9.

2.6. Логический элемент И-НЕ

Выход «S» логического элемента И-НЕ, по-английски NAND, переходит в состояние логического нуля в том случае, если одновременно оба входа «А» и «В» равны 1. Если хотя бы один из входов равен нулю, на выходе элемента будет единица (табл. 2.6).

Таблица 2.6. Таблица истинности логического элемента И-НЕ

Вход А	Вход В	Выход S
0	0	1
1	0	1
0	1	1
1	1	0

На рис. 2.11 показаны условные обозначения логического элемента И-НЕ на схемах, а на рис. 2.12 показаны временные диаграммы, поясняющие его работу.

Рис. 2.11. Условные обозначения логического элемента И-НЕ и математическая формула работы элемента

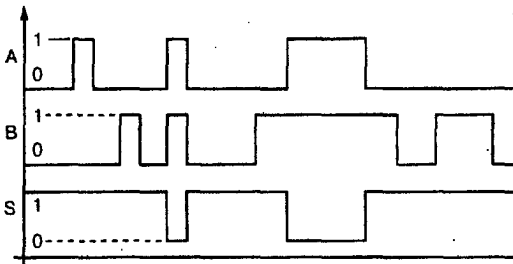
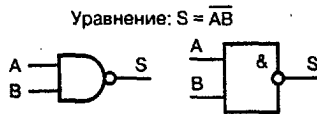


Рис. 2.12. Временная диаграмма, поясняющая работу логического элемента И-НЕ

Используемые микросхемы:

- CD4011, включает 4 логических элемента 2И-НЕ технологии КМОП;
- CD4023, включает 3 логических элемента 3И-НЕ технологии КМОП;
- CD4012, включает 2 логических элемента 4И-НЕ технологии КМОП;
- CD4068, включает 1 логический элемент 8И-НЕ технологии КМОП;
- SN7400, включает 4 логических элемента 2И-НЕ технологии ТТЛ;
- SN7410, включает 3 логических элемента 3И-НЕ технологии ТТЛ;
- SN7420, включает 2 логических элемента 4И-НЕ технологии ТТЛ;
- SN7430, включает 1 логический элемент 8И-НЕ технологии ТТЛ.

Программирование

Команда И-НЕ, точно так же как и ИЛИ-НЕ, широко используется в логической электронике и не применяется в программировании. В QBASIC такой команды не существует. Для создания соответствующей функции в программировании достаточно проинвертировать условия или результат, соответствующие команде И.

2.7. Логический элемент ИСКЛЮЧАЮЩЕЕ ИЛИ

Выход «S» логического элемента *ИСКЛЮЧАЮЩЕЕ ИЛИ*, по-английски XOR, переходит в состояние логической единицы в том и только в том случае, если на один вход «А» или «В» подается 1, а другой – соответственно в 0. В том случае, если на оба входа подается логическая единица, на выходе будет нуль. Точно так же будет, если на оба входа подать нуль (табл. 2.7).

На рис. 2.13 показаны условные обозначения логического элемента ИСКЛЮЧАЮЩЕЕ ИЛИ на схемах, а на рис. 2.14 показаны временные диаграммы, поясняющие его работу.

Таблица 2.7. Таблица истинности логического элемента ИСКЛЮЧАЮЩЕЕ ИЛИ

Вход А	Вход В	Выход S
0	0	0
1	0	1
0	1	1
1	1	0

Рис. 2.13. Условные обозначения логического элемента ИСКЛЮЧАЮЩЕЕ ИЛИ и математическая формула работы элемента

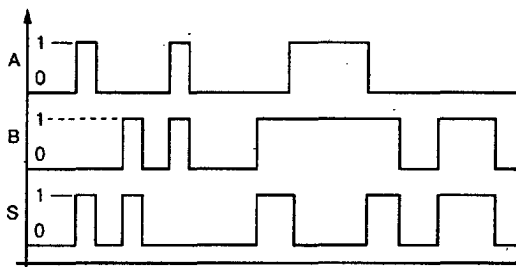
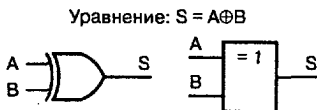


Рис. 2.14. Временная диаграмма, поясняющая работу логического элемента ИСКЛЮЧАЮЩЕЕ ИЛИ

Используемые микросхемы:

- CD4070, включает 4 логических элемента ИСКЛЮЧАЮЩЕЕ ИЛИ технологии КМОП;
- SN74136, включает 4 логических элемента ИСКЛЮЧАЮЩЕЕ ИЛИ технологии ТТЛ.

Программирование

Команда XOR в QBASIC представлена в виде функции и булева оператора. В первом случае команда XOR используется для создания маски, которая инвертирует нужные биты. В случае использования булева оператора XOR в программе сравниваются два условия, правильный результат получается в том и только том случае, если выполняется только одно из условий.

Пример 1 (функция):

```
A%=13          'Или 1101 в двоичной форме
B%=6           'Или 0110 в двоичной форме (B% - маска)
PRINT A% XOR B% 'Результат: 11 в десятичной форме или
                '1011 в двоичной форме
```

Нулевой и третий биты числа A% не принимаются во внимание, они не меняются. Первый и второй биты после действия функции XOR оказались проинвертированными.

Пример 2 (булев оператор):

```
НАЧАЛО:
INPUT "Введите цифру";A%
INPUT "Введите еще одну цифру";B%
IF A%=4 XOR B%=9 THEN PRINT "Правильно!" ELSE PRINT "Не
правильно!"
GOTO НАЧАЛО
```

На экране компьютера появится поздравительная надпись, если вы присвоите переменной A% значение 4, при этом значение B% будет не равно 9, или же если вы присвоите переменной B% значение 9, при этом значение A% будет не равно 4. Во всех других случаях на экране будет появляться надпись «Не правильно».

2.8. Логический элемент ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ

Выход «S» логического элемента ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ, по-английски XNOR, остается в состоянии логического нуля в том и только в том случае, если вход «А» или «В» равен 1, а другой – равен 0.

В том случае, если на оба входа подается логическая единица, на выходе будет единица. Точно так же будет, если на оба входа подать нуль (табл. 2.8).

Таблица 2.8. Таблица истинности логического элемента ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ

Вход А	Вход В	Выход S
0	0	1
1	0	0
0	1	0
1	1	1

На рис. 2.15 показаны условные обозначения логического элемента ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ на схемах, а на рис. 2.16 показаны временные диаграммы, поясняющие его работу.

Рис. 2.15. Условные обозначения логического элемента ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ и математическая формула работы элемента

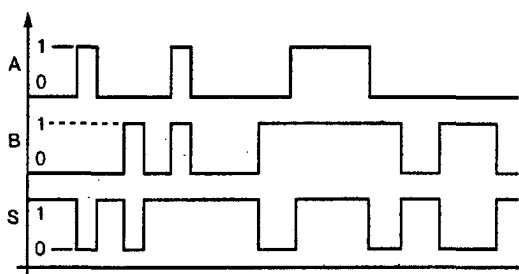
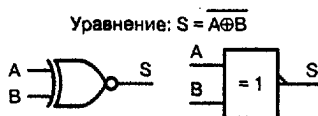


Рис. 2.16. Временная диаграмма, поясняющая работу логического элемента ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ

Используемая микросхема

Используется микросхема CD4077, которая включает 4 логических элемента 2 ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ технологии КМОП.

Программирование

Команда XNOR обычно не используется в программировании, зато она очень распространена в логической электронике. В QBASIC существует булев оператор EQU, действие которого совпадает с таблицей истинности элемента ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ. Этот оператор изменяет результаты предыдущего примера программирования на противоположные (результат ложный тогда и только тогда, когда оба условия одновременно являются истиной).

2.9. Триггер Шмитта

Пороговый вентиль, называемый также *триггером Шмитта*, работает следующим образом. При подаче на вход триггера порогового напряжения напряжение на выходе скачком переходит из одного состояния в другое. Этот элемент существует в форме элемента И-НЕ или инвертора, он часто используется в логической электронике для создания схем, предназначенных для перехода из низкого состояния в высокое без промежуточных состояний и обратно (генераторы, замедлители). В обозначения на схемах внутрь квадратика добавляется маленький символ, который означает, что это триггер Шмитта (рис. 2.17).

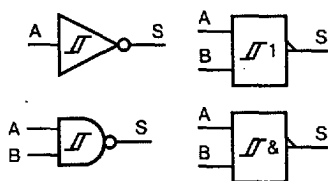


Рис. 2.17. Обозначения триггера Шмитта на схемах

Используемые микросхемы:

- CD4093, включает 4 триггера Шмитта с двумя логическими элементами 2И-НЕ технологии КМОП;
- CD40106, включает 6 триггеров Шмитта с инвертором технологии КМОП;
- SN74132, включает 4 триггера Шмитта с двумя логическими элементами 2И-НЕ технологии ТТЛ;
- SN7414, включает 6 триггеров Шмитта с инвертором технологии ТТЛ.

2.10. Прибор для тестирования двухходовых логических элементов

В конце этой главы, в которой изучаются 9 основных логических элементов, мы вам предлагаем изготовить первую схему. Это очень полезное устройство, предназначенное не только для тестирования логических микросхем КМОП серии 4000 с 4 логическими элементами в корпусе. Вторым назначением схемы является индикация различных функций, которые они выполняют (рис. 2.18).

Принципиальная схема тестера двухходовых логических элементов

Схема на рис. 2.18, 2.19, которая может вначале показаться слишком трудной, состоит из простых и идентичных блоков.

Она предназначена для тестирования микросхем КМОП, состоящих из 4 логических элементов. Мы должны изучить логическое состояние на каждом из 8 входов, которые в состоянии покоя соединены с «землей». 8 рабочих кнопок переводят при необходимости соответствующие входы в состояние логической «1».

Микросхемы 74НС540, обозначенные на схеме как С11, С12, С13, состоят из 8 инверторов и узла управления по входу (выводы 1.19), которые в нашем случае достаточно соединить с «землей», чтобы активировать микросхему. Каждый отдельный логический элемент при тестировании использует 2 инвертора из 8 (смотрите схему в рамке на рис. 2.19). 2 инвертора используются по следующей причине. Пока кнопка не нажата, сигнал инвертируется один раз; красная подсветка двухцветного светодиода обозначает, что на входе инвертора состояние «0». Когда кнопка нажата, светодиод светится зеленым цветом, это сигнализирует о том, что на вход инвертора подается логическая единица. Далее сигнал с выхода второго инвертора после двойного инвертирования восстанавливает свою первоначальную полярность и подается на вход тестируемой микросхемы. Токоограничительный резистор номиналом 1 кОм ограничивает ток светодиода до приемлемого для данной микросхемы.

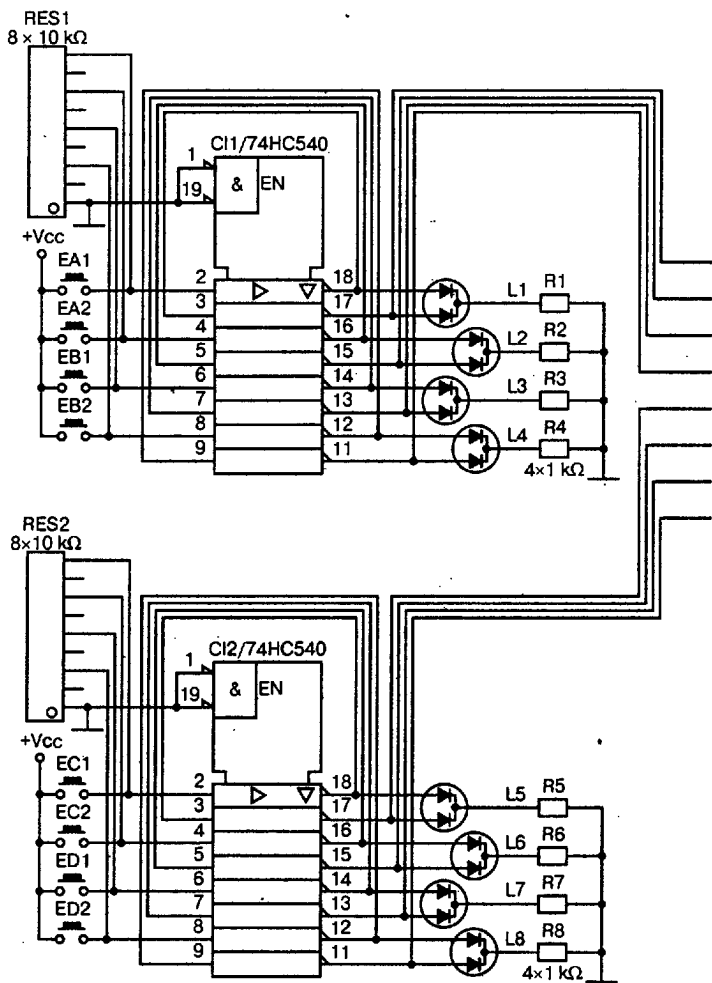


Рис. 2.18. Принципиальная схема тестера двухвходовых логических элементов

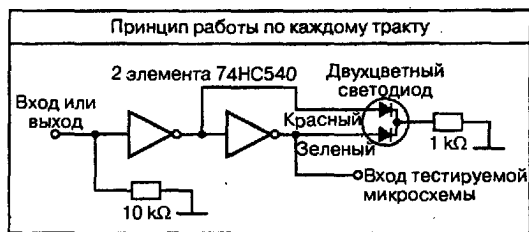
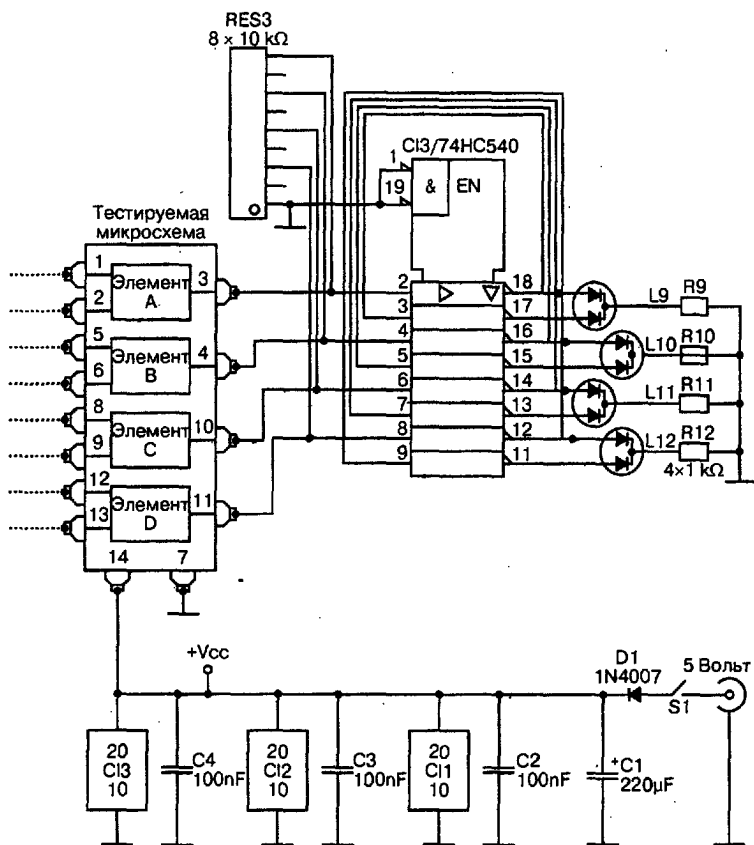


Рис. 2.18. (продолжение) Принципиальная схема тестера двухвходовых логических элементов

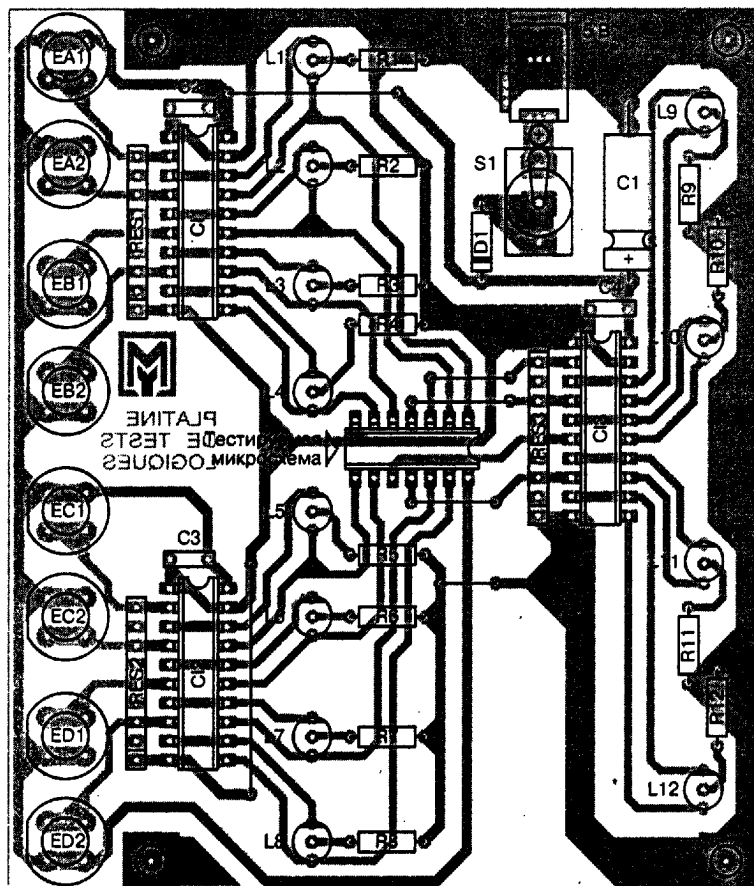


Рис. 2.19. Рисунок печатной платы

Восемь таких блок-схем, включающих в себя микросхемы C11 и C12, резисторные сборки RES1 и RES2, светодиоды L1-L8 и резисторы R1-R8, образуют схему индикации и подачи сигналов на тестируемый образец.

Выходная схема функционирует точно таким же образом. Сигнал с каждого выхода инвертируется два раза и подается на светодиод. Выходной блок состоит из микросхемы C13, набора резисторов RES3, светодиодов L9-L12 и сопротивлений R9-R12.

Схема питается напряжением 5В. Возможно питать схему от батареи 4,5 В, в этом случае эксперименты можно проводить автономно. Выключатель S1 отключает питание. Его использование необязательно. Диод D1 защищает схему от неправильного подключения аккумуляторной батареи – так называемая защита от дурака. Конденсатор C1 сглаживает низкочастотные пульсации напряжения питания, конденсаторы C2, C3, C4 – высокочастотные пульсации микросхем C11, C12 и C13.

Сборка тестера двухходовых логических элементов

На рис. 2.20 показан чертеж печатной платы для травления. Травление можно осуществлять любым известным вам способом. В библиографии, расположенной в конце книги, вы найдете источники, описывающие методику травления.

Установка элементов не представляет особой трудности, но требует внимания и аккуратности. Начните с пайки проводов в изоляции шести связывающих перемычек, которые получились в результате машинной разводки (вернее – машинной недоразводки) печатных проводников, чтобы не забыть это сделать потом. Продолжайте монтаж, формируя выводы элементов. Запаяйте сначала резисторы, потом диод (не перепутайте полярность – будете тем самым «дураком», если перепутайте), панельки для микросхем C11, C12 и C13, соблюдая установку по ключу (короткая сторона микросхемы с выемкой), резисторные сборки; керамические конденсаторы; кнопки; разъем для подключения питания; электролитический конденсатор и светодиоды устанавливаются также, соблюдая полярность; выключатель (по желанию) и, наконец, панельку для установки тестируемой схемы. Светодиоды перед пайкой выровняйте по высоте. Панельку для тестируемой микросхемы можно изготовить из 14 залуженных штырей, расположенных на определенной высоте*.

* Делается в крайнем случае, когда не нашли маленькие, но обычно их легко купить и «самодельным» быть не нужно. – *Прим. науч. ред.*

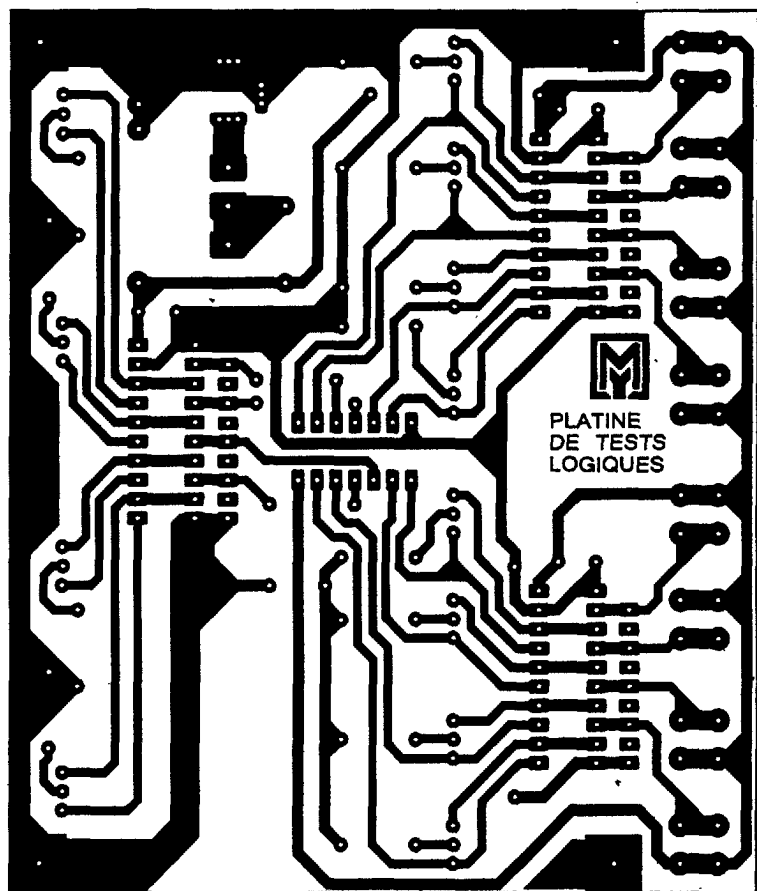


Рис. 2.20. Внешний вид платы тестирования логических элементов

Включение и работа установки

После сборки прибора обязательно проверьте правильность размещения и номиналы элементов. При необходимости используйте лупу для просмотра возможных «холодных паяк», выявления некачественных паяк или замыканий соседних печатных проводников, произошедших в результате пайки. Проверьте качество пайки элементов. При прозвонке в режиме омметра между шинами питания и «земли» при отсутствии

радиоэлементов не должно быть короткого замыкания, иначе его необходимо найти и устранить.

Если вы не уверены в правильности сборки, подключите напряжение питания к плате, не вставляя микросхемы в держатели. С помощью вольтметра проконтролируйте напряжения питания микросхем, вы должны получить примерно 4,5 В, если сборка качественная. В противном случае напряжение будет меньше, и это признак некачественной сборки.

Пользоваться этим прибором несложно. Он позволяет проверять и тестировать все двухходовые КМОП микросхемы по 4 элемента в корпусе микросхемы серии 4000, то есть:

- -CD4071 4-элемента ИЛИ;
- CD4001 4-элемента ИЛИ-НЕ;
- CD4081 4-элемента И;
- CD4093 4-элемента И-НЕ;
- CD4070 4-элемента ИСКЛЮЧАЮЩЕЕ ИЛИ;
- CD4077 4-элемента ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ.

Список компонентов:

Резисторы с 5-процентным допуском:

- R1-R12: 1 кОм (коричневый, черный, красный);
- RES1-RES3: резистивная сборка 8 × 10 кОм.

Конденсаторы:

- C1: 220-470 пФ / 16 В (электролитический с аксиальными выводами);
- C2-C4: 0,1 мкФ керамические.

Полупроводники:

- L1-L12: двухцветные светодиоды с 3 выводами;
- D1: диод 1N4007;
- CI1-CI3: ИМС 74НС540.

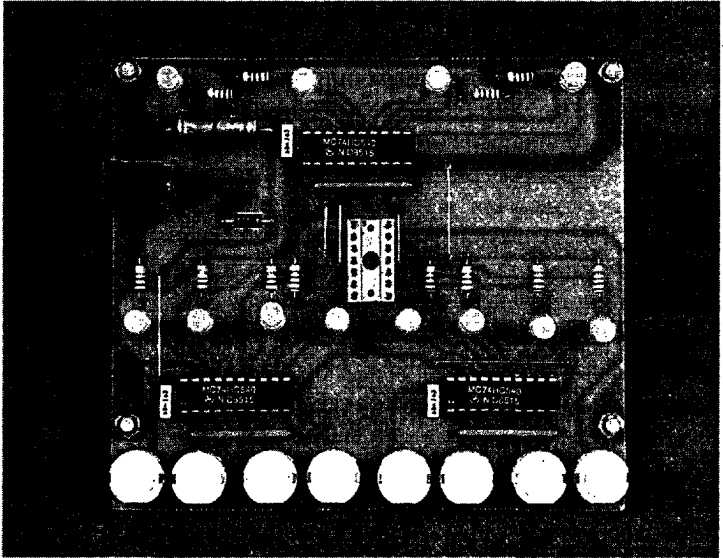


Рис. 2.21. Деталь: держатель тестируемой микросхемы

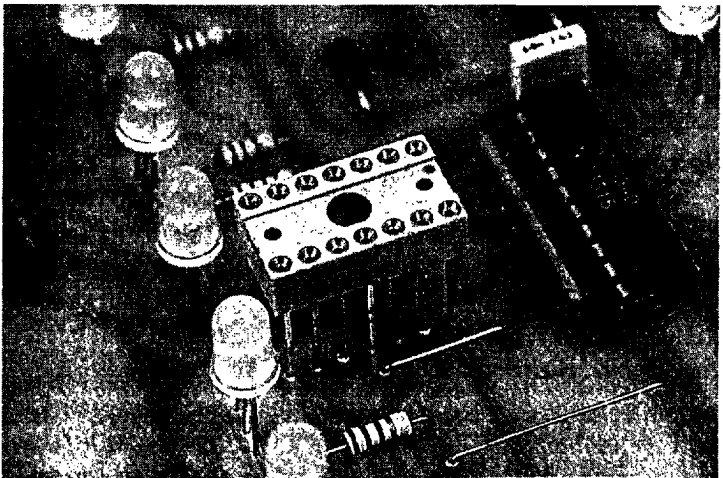


Рис. 2.22. Установка элементов на печатной плате

Прочие элементы:

- 1 держатель микросхемы с 14 выводами;
- 3 держателя 20-выводных микросхем;
- 1 однополюсный выключатель (по желанию);
- 8 кнопочных переключателей;
- 1 разъем подключения питания диаметром 2,1 мм;
- болты и стойки диаметром 3 мм.

Выключите схему, отключив выключатель или вытащив штеккер питания из разъема. Затем вставьте тестируемую микросхему КМОП и включите выключатель или снова подключите штеккер к разъему. Нажмите на кнопки EA1 и EA2, чтобы увидеть результат на выходе SA. Проведите сравнение с помощью таблиц истинности. Три оставшихся элемента должны функционировать тем же образом от соответствующих кнопок.

Глава **3**

Схемотехника на основе логических элементов

3.1. Краткое представление содержания главы 3

До этого момента мы изучали функционирование основных логических элементов. В этой главе будет показано, как, комбинируя логические элементы между собой, можно получить более сложные схемы, реализующие определенные задачи, такие как, генерация звуковой частоты, световые эффекты, формирование калиброванных одиночных импульсов, замедлители сигнала, основанные на скорости заряда-разряда конденсатора. Мы также изучим в этой главе применение микросхем с большой плотностью размещения логических элементов, образующие триггеры, счетчики, а также другие независимые электронные функции.

3.2. Мультивибраторы на логических элементах

Основные положения

Существует много способов создания мультивибраторов, генерирующих импульсы. Опишем способы, которые мы будем применять наиболее часто.

Схема, изображенная на рис. 3.1, является самым простым вариантом мультивибратора импульсов. Она реализуется с помощью одного из шести триггеров Шмитта с инвертором CD40106 или с помощью одного из четырех триггеров Шмитта с двумя логическими элементами 2И-НЕ CD4093, подключенного как инвертор (когда два его входа соединены между собой).

Рассмотрим работу этой схемы. Она основана на заряде-разряде конденсатора С через резистор R. При подаче импульса напряжения на вход 3 элемента N1 конденсатор С начинает заряжаться через сопротивление R. После того как напряжение на нем достигает порогового значения, то есть на входах элемента присутствует логическая единица, выход переходит в состояние нуля. Конденсатор начинает разряжаться. Когда напряжение на нем достигнет нижнего порогового уровня, происходит переход выхода в «1» и процесс повторяется. Приблизительная формула, позволяющая рассчитать период повторения импульсов, пишется следующим образом:

$$T(\text{сек}) = 0,4 \times R(\text{Ом}) \times C(\text{Ф}).$$

Пример: $0,4 \times 100000 \times 0,000001 = 0,04 \text{ с}$

Чтобы упростить расчеты, следует применять следующие единицы измерения:

$$T(\text{сек}) = 0,4 \times R(\text{МОм}) \times C(\text{мкФ}).$$

Пример: $0,4 \times 0,1 \times 1 = 0,04 \text{ с}$

Ниже приведены две формулы, позволяющие рассчитать частоту импульсов:

$$F(\text{Гц}) = \frac{1}{2T};$$

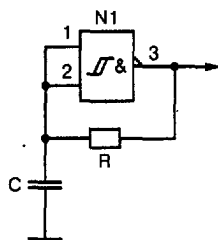


Рис. 3.1. Схема мультивибратора на одном триггере Шмитта с логическим элементом 2И-НЕ

Пример: $\frac{1}{2 \times 0,4} = 12,5 \text{ Гц}$

$$F (\text{Гц}) = \frac{1}{0,8 \times R(\text{МОм}) \times C(\text{мкФ})}$$

Пример: $\frac{1}{0,8 \times 0,1 \times 1} = 12,5 \text{ Гц}$

Как мы видим, мультивибратор вырабатывает импульсы фиксированной частоты. Изменения диапазона частот можно добиться подключением конденсатора различной емкости, как видно из рис. 3.2. Окончательная «точная» настройка частоты осуществляется переменным сопротивлением АЖ. Постоянный резистор R вместе с конденсатором C служит для ограничения частоты колебаний сверху, то есть, «грубой» настройки частоты.

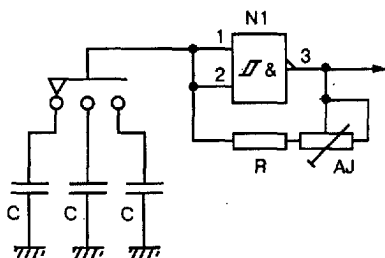


Рис. 3.2. Схема мультивибратора на одном триггере Шмитта с логическим элементом 2И-НЕ на 3 диапазона частот импульсов

Вторая схема мультивибратора составлена из двух инвертирующих элементов (ИЛИ-НЕ, И-НЕ) и содержит дополнительный резистор. Его схема изображена на рис. 3.3.

Следует иметь в виду, что его работа менее стабильна, чем работа предыдущей схемы. Для некоторых значений конденсатора C1 колебания не могут возникать. Кроме того, номинал сопротивления R2 должен не менее чем в 10 раз превосходить номинал R1. Чтобы изучить принцип работы

мультивибратора-автогенератора упрощенным способом, рассмотрим состояние схемы, когда на выходе элемента N2 присутствует логический нуль. Разряжаясь, конденсатор через резистор R2 подает на входы элемента N1 логический «0». Тогда на выходе N1 и соответственно на входах элемента N2 будет «1». Далее конденсатор заряжается через резистор R1 до тех пор, пока напряжение на нем не достигнет верхнего порогового уровня N1. Это напряжение переключает элемент N1, и на входе элемента N2 появляется логический «0». Конденсатор вновь начинает разряжаться через сопротивление R1 до тех пор, пока N1 не переключится. Далее цикл повторяется. Приблизительная формула, позволяющая рассчитать период повторения импульсов, переходов высокого уровня в низкий и наоборот, запишется следующим образом:

$$T(\text{сек}) = 1,1 \times R2(\text{Ом}) \times C1(\text{Ф});$$

Пример: $1,1 \times 470000 \times 0,0000022 = 1,1374 \text{ с.}$

Чтобы упростить расчеты, следует применять следующие единицы измерения:

$$T(\text{сек}) = 1,1 \times R2(\text{МОм}) \times C1(\text{мкФ});$$

Пример: $1,1 \times 0,47 \times 2,2 = 1,1374 \text{ с.}$

Ниже приведены две формулы, позволяющие рассчитать частоту повторения импульсов:

$$F(\text{Гц}) = \frac{1}{2T};$$

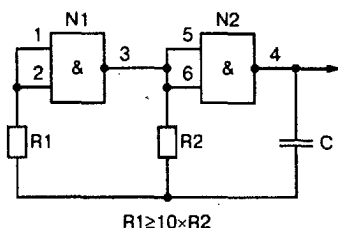


Рис. 3.3. Схема мультивибратора на двух элементах И-НЕ

Пример: $\frac{1}{2 \times 1,1374} = 0,439 \text{ Гц}$

$$F (\text{Гц}) = \frac{1}{2,2 \times R2 (\text{МОм}) \times C1 (\text{мкФ})}$$

Пример: $\frac{1}{2,2 \times 0,47 \times 2,2} = 0,439 \text{ Гц}$.

Приложения

В двух предложенных ниже схемах мультивибраторов из-за своей простоты используются схемы на одном элементе – триггере Шмитта с логическим элементом 2И-НЕ, включенным как инвертор. Схема мультивибратора аналогична изображенной на рис. 3.1 и управляет работой ключа на транзисторе Т1, в коллекторной нагрузке которого светодиод.

На низкой частоте, например 1 Гц, светодиод мигает один раз в секунду. С помощью транзистора Т1 сигнал с выхода малоомощного элемента N1 усиливается и управляет нагрузкой – светодиодом, как показано на рис. 3.4.

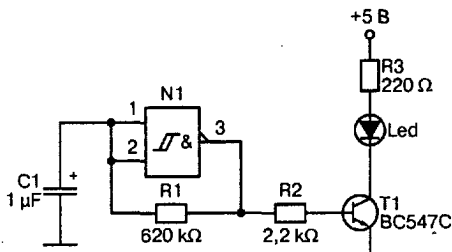


Рис. 3.4. Схема генератора на одном триггере Шмитта с логическим элементом 2И-НЕ, управляющая «миганием» светодиода

На более высокой частоте, например 3200 Гц, легко образуется однотонный звук, вырабатываемый пьезоизлучателем с помощью схемы мультивибратора на одном триггере Шмитта с логическим элементом 2И-НЕ, объединенным в инвертор (рис. 3.5 а). Используя два мультивибратора, выходы которых подключены к пьезоизлучателю, можно получить прерывистое звучание на частоте 3200 Гц, модулированное частотой 100 Гц, например, как в схеме на рис. 3.5 б. Таким образом, пьезоизлучатель на рис. 3.5 а будет звучать непрерывно, а на рис. 3.5 б – прерывисто с частотой модуляции 100 Гц*.

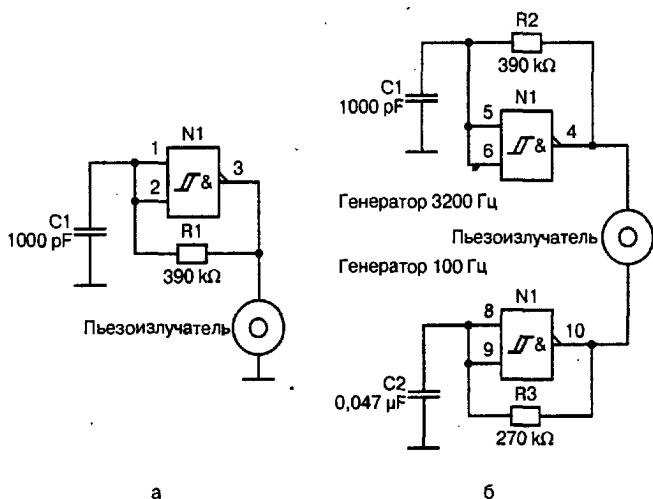


Рис. 3.5. Звуковые генераторы на логических элементах с использованием пьезоизлучателей

* Пьезоизлучатель звучит в узком диапазоне частот в отличие от динамика. – Прим. науч. ред.

3.3. Простой одновибратор с управлением от кнопки, генерирующий одиночный положительный импульс

Основные положения

Это устройство позволяет сформировать импульс определенной длительности, полярность которого зависит от использу-

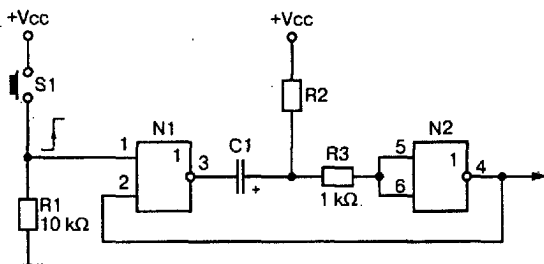


Рис. 3.6. Схема одновибратора, генерирующего одиночный положительный импульс

емых элементов. На рис. 3.6 показана схема одновибратора, генерирующего одиночный положительный импульс.

В этой схеме используются два из четырех логических элемента ИЛИ-НЕ N1 и N2 микросхемы CD4001. В состоянии покоя выход (вывод 4 элемента N1) находится на низком уровне. Следовательно, на низком уровне будет находиться соединенный с ним также вход 2 элемента N1. Вход 1 также находится на низком уровне из-за резистора R1, подключенного к «земле». Тогда выход N1 согласно таблице истинности элемента ИЛИ-НЕ (табл. 2.4) находится на высоком уровне, он соединен с отрицательным выводом конденсатора C1. Конденсатор C1 находится в разряженном состоянии, так как его положительный вывод соединен с положительным напряжением (уровень

«1») через резистор R2. Выход элемента остается в состоянии «0», так как на его входах присутствует логическая «1».

Нажатие на кнопку S1 подключает вход 1 к источнику питания +Vcc и заставляет переключиться элемент N1 в другое состояние, на выходе 3 появляется логический «0». После этого конденсатор C1 быстро заряжается через сопротивление R2 и переводит входы элемента N2 в состояние логического «0». До тех пор, пока конденсатор не зарядится через сопротивление R2, такое состояние схемы будет поддерживаться автоматически за счет того, что на выходе N2, а следовательно, и на входе 2 N1 будет логическая «1», даже если отпустить кнопку S1. Сопротивление R3 необязательно, но все же необходимо – оно защищает схему в случае, если емкость C1 будет слишком большой.

Длительность импульса T на выходе элемента N2 определяется времязадающими элементами – конденсатором C1 и резистором R2. Ниже приведена формула, позволяющая приближенно рассчитать длительность импульса T:

$$T(\text{сек}) = 0,7 \times R2(\text{Ом}) \times C1(\Phi);$$

Пример: $0,7 \times 220000 \times 0,00000068 = 0,10472 \text{ с.}$

Чтобы упростить расчеты, следует применять следующие единицы измерения:

$$T(\text{сек}) = 0,7 \times R2(\text{МОм}) \times C1(\text{мкФ});$$

Пример: $0,7 \times 0,22 \times 0,68 = 0,10472 \text{ с.}$

В этом примере нажатие на кнопку S1 будет генерировать одиночный импульс длительностью, примерно равной 0,1 с, каким бы долгим ни было время нажатия на кнопку. Мы рассмотрели схему, которая позволяет избежать нежелательных пачек импульсов (дребезга контактов) при нажатии на кнопку за счет формирования импульса длительностью, превышающей время дребезга контактов. Временная диаграмма на рис. 3.7 конкретизирует данное объяснение.

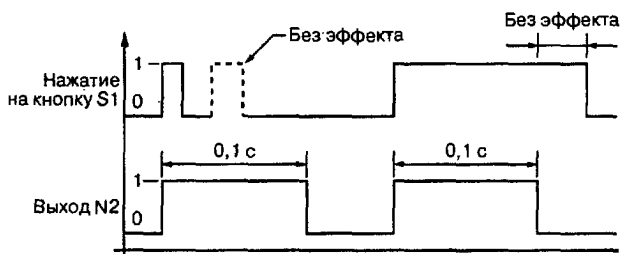


Рис. 3.7. Временная диаграмма, объясняющая работу одновибратора, генерирующего одиночный положительный импульс

Принцип, позволяющий получить инверсную логику, то есть импульс уровня «0» при исходном постоянном состоянии «1», показан схематически на рис. 3.8. Мы не будем разьяснять работу данной схемы, очень похожую на предыдущую. Но есть и существенные отличия: используются схемы 2И-НЕ вместо 2ИЛИ-НЕ, R2 подключен к «земле» вместо +Vcc и кнопка не в верхнем, а в нижнем «плече» входной схемы. Формулы расчета остаются аналогичными формулам для одиночного положительного импульса.

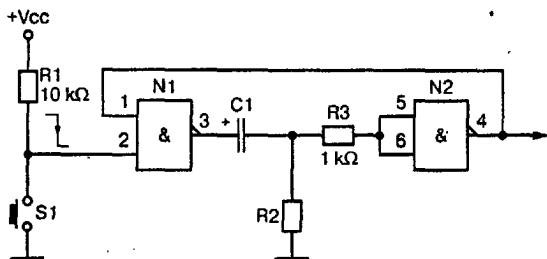


Рис. 3.8. Одновибратор с управлением от кнопки, генерирующий одиночный отрицательный импульс

3.4. Одновибратор с перезапуском

Основные положения

Данное устройство позволяет расширить импульс дополнительным нажатием на кнопку. В противоположность предыдущей схеме, если управляющий импульс подается до завершения длительности импульса, одновибратор продолжает формировать импульс. На рис. 3.9 представлена схема одновибратора.

Это устройство может быть реализовано на любом из следующих трех логических элементов: ИЛИ-НЕ, И-НЕ, инвертор. В состоянии покоя (кнопка не нажата) обе обкладки конденсатора C соединены с положительным потенциалом $+V_{cc}$, поэтому на выходе элемента $N1$ присутствует логический «0». При нажатии на кнопку $S1$ на входе элемента $N1$ подается напряжение 0 В, конденсатор быстро заряжается. Разряд конденсатора может начаться только после того, как отпустить кнопку $S1$. При этом на выходе $N1$ устанавливается логическая «1». Импульс будет продолжаться до тех пор, пока на входах $N1$ не установится нижний порог напряжения в результате разряда конденсатора. Если нажать снова на кнопку до окончания следования импульса, импульс продолжится, так как конденсатор C вновь зарядился.

Приблизительная формула, позволяющая рассчитать время T импульса после отпускания кнопки, совпадает с формулой для одновибратора с простым эффектом:

$$T(\text{сек}) = 0,7 \times R(\text{Ом}) \times C(\Phi);$$

Пример: $0,7 \times 470000 \times 0,0000022 = 0,7238 \text{ с.}$

Чтобы упростить расчеты, следует применять следующие единицы измерения:

$$T(\text{сек}) = 0,7 \times R(\text{МОм}) \times C(\text{мкФ}).$$

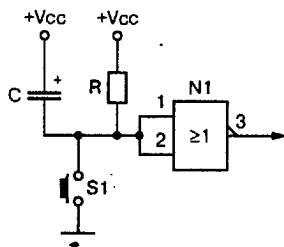


Рис. 3.9. Схема одновибратора с перезапуском

Пример: $0,7 \times 0,47 \times 2,2 = 0,7238$ с.

Эта схема реагирует на каждое нажатие на кнопку. Она может также служить интегратором для определения недостающих импульсов в цикле: при их отсутствии выходной вывод переходит в состояние «0». Диаграмма на рис. 3.10 помогает понять принцип действия.

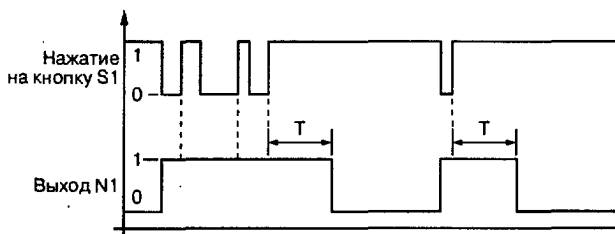


Рис. 3.10. Временная диаграмма, поясняющая работу одновибратора с множественным действием

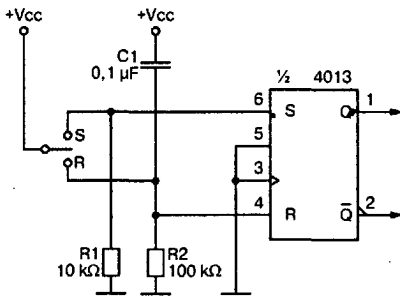
3.5. RS-триггер

Основные положения

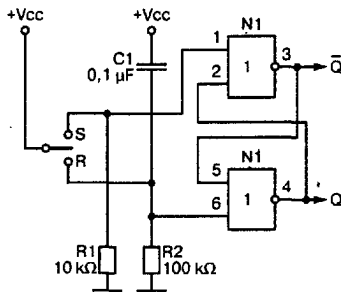
RS-триггер – устройство для хранения информации: положительный импульс на вход S устанавливает на выходе Q логическую единицу «1», а на выходе \bar{Q} – «0». Положительный импульс на входе R изменяет состояние выходов на противоположное. Это устройство занимает половину корпуса CD4013, как это показано на рис. 3.11 (а)*. Его можно легко изготовить комбинацией двух логических элементов ИЛИ-НЕ, как это показано на рис. 3.11 (б). Конденсатор C1 служит для установки в первоначальное нулевое состояние, чтобы избежать случайного состояния, обусловленного только временем переключения элементов под действием напряжения. Сопротивления R1 и R2 в состоянии покоя поддерживают входы заземленными.

* CD4013 состоит из двух D-триггеров, один из которых не задействован, а другой переведен в RS-триггер за счет подключения входов D и C к «земле». – Прим. науч. ред.

Чтобы понять читателю принцип работы RS-триггера, в таблице 3.1 представлена таблица истинности. Следует особо рассмотреть два важных состояния: случай, когда на входы подаются нули, не влияет на состояние выходов. Случай, в котором на оба входа подается единица, приведет к тому, что на выходах установится одинаковый потенциал, что на практике является невозможным. Старайтесь запрещенные состояния не практиковать, что является признаком недостаточной опытности радиолюбителя, и не подавать на R- и S-входы два одинаковых потенциала – единицы в одно и то же время.



(a)



(б)

Рис. 3.11. RS-триггер на $\frac{1}{2}$ D-триггера (а) и на двух элементах ИЛИ-НЕ (б)

Таблица 3.1. Таблица истинности RS-триггера

Состояние R	Входы		Выходы	
	R	S	Q	\bar{Q}
Без изменений	0	0	Неизменно	
Переход в 1	0	1	1	0
Переход в 0	1	0	0	1
Запрещено	1	1	Запрещено	

Используя схему на рис. 3.12, можно легко получить RS-триггер с инверсными значениями сигналов, реализуемый с помощью логических элементов И-НЕ.

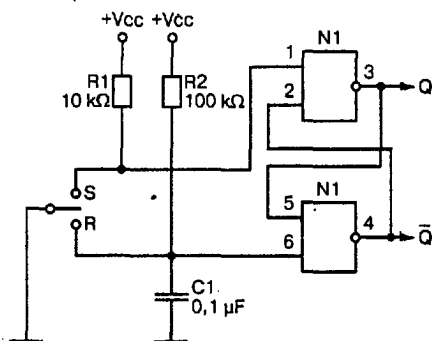


Рис. 3.12. Инверсный RS-триггер на двух элементах И-НЕ

3.6. D-триггер

Основные положения

D-триггер по-другому называется защелкой: простой положительный импульс на вход С устанавливает на выходе Q единицу, а на выходе \bar{Q} — ноль при условии, что на входе D имеется логическая единица. Если на входе D установлен логический ноль, тогда при подаче синхроимпульса С выходы триггера установятся инверсно относительно предыдущего варианта установки потенциала на входе D. В любом из вариантов работа D-триггера как защелки переключения из одного состояния в другое происходит при подаче положительного фронта синхроимпульса С.

D-триггер также располагается в половине корпуса CD4013, как это показано на рис. 3.13.

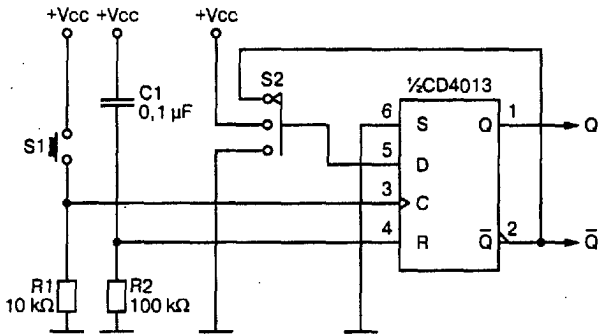


Рис. 3.13. Схема D-триггера на $\frac{1}{2}$ CD4013

Для того, чтобы избежать случайного состояния, обусловленного только временем переключения элементов под действием напряжения, на выходе интегрирующей цепи из конденсатора C1 и резистора R2 при включении питания формируется положительный импульс, которым гарантированно устанавливается D-триггер в нуль. Сопротивление R1 в состоянии покоя поддерживает вход C заземленным.

Вход D с помощью 3-позиционного переключателя S2 можно соединить с 1, 0 или с выходом \bar{Q} . В последнем случае каждый положительный фронт импульса на вход C переключает выходы в инверсное состояние относительно предыдущего. В таком включении D-триггера он становится делителем частоты на 2.

Чтобы избежать неопределенного состояния триггера, вход C следует соединять с одновибратором, управляемым от кнопки. Если кнопка не нажата, то вход C через резистор R1 подключен к «земле».

-- Сконструировать аналогично представленному D-триггеру можно с помощью 4 элементов И-НЕ согласно схеме на рис. 3.14. Самые наблюдательные из вас заметят, что в этой схеме можно выделить блок инверсного RS-триггера.

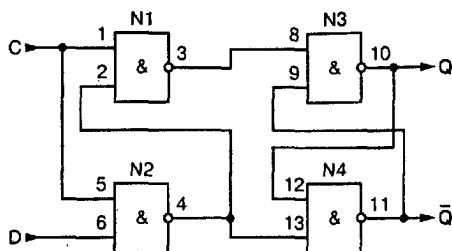


Рис. 3.14. Схема D-триггера на основе логических элементов И-НЕ

В табл. 3.2 представлена таблица истинности D-триггера и RS-триггера, содержащиеся в половине схемы CD4013 при разном включении: в качестве D-триггера R- и S-входы должны быть подключены к «земле», а в качестве RS-триггера – аналогично D- и C-входы подключаются к «земле». И последний вариант – работа D-триггера как счетного будет происходить, если R- и S-входы подключены к «земле», вход D соединен с выходом \bar{Q} и переключение выходов Q и \bar{Q} происходит по положительному фронту импульса на входе C.

Таблица 3.2. Таблица истинности D-триггера и RS-триггера, входящих в состав микросхемы CD4013

Состояние	Входы				Выходы	
	R	S	D	C	Q	\bar{Q}
Переход в 0	1	0	X*	X	0	1
Переход в 1	0	1	X	X	1	0
Запрещенное	1	1	X	X	Неопределенное	
Переход в 0	0	0	0	Нарастающий фронт импульса	0	1
Переход в 1	0	0	1	Нарастающий фронт импульса	1	0
	0	0	X	0	Без изменений	
	0	0	X	1	Без изменений	
	0	0	X	Спадающий фронт импульса	Без изменений	

* X – безразличное состояние

3.7. Двоичные счетчики

Основные положения

Двоичные счетчики состоят из нескольких триггеров, таких как D-триггеры в счетном режиме или на другом типе триггера, соединенных синхронно или асинхронно.

В синхронном счетчике входной импульс действует одновременно на входы всех триггеров, связанных между собой с помощью логических элементов. Сложность этих элементов зависит от уровня счета. Время переключения определяется временем задержки самого медленного триггера. Синхронные двоичные счетчики можно использовать при больших частотах переключений.

В асинхронном двоичном счетчике входной импульс действует на первый триггер, выход которого соединен со вторым триггером и т. д. до самого последнего триггера. Время переключения сигнала становится большим, так как оно определяется прохождением сигнала через все триггеры. Асинхронные счетчики используются на низких частотах переключений.

Каждый триггер делит частоту входных тактовых импульсов на 2, то есть при подсчете числа импульсов на выходе триггера можно использовать двоичные числа. Линейки из двух триггеров-делителей делят начальную частоту на 4, из 3 – на 8, из 4 – на 16 и так далее кратно 2. Можно составить n -разрядов двоичных счетчиков с общим коэффициентом деления 2^n .

Схема с применением CD4040

CD4040 – 12-разрядный асинхронный счетчик, позволяющий делить частоту с делителями от 2 до 2^{12} или 4096 – с выхода 0 (вывод 9)... до выхода 11 (вывод 1). Схема на рис. 3.15 показывает один из примеров применения микросхемы. На этой схеме импульсы частотой 2 Гц вырабатываются мультивибратором на RC-цепи и делятся выбранным делителем.

Триггер Шмитта N1 с логическим элементом 2И-НЕ, объединенным в инвертор, резистор R1, переменное сопротивление-потенциометр AJ1 и конденсатор C1 образуют мультивибратор на RC-цепи, состоящей из конденсатора C1 и суммы постоянного резистора R1 и потенциометра AJ1. С выхода мультивибратора на вход счетчика подаются импульсы частотой 2 Гц. Эту частоту можно точно настроить с помощью потенциометра AJ1. Интегрирующая цепь, состоящая из конденсатора C2 и резистора R2, автоматически сбрасывает счетчик в 0 каждый раз при включении напряжения питания. Кнопка S1 выполняет это действие вручную.

Данную схему можно использовать в качестве генератора временных интервалов, рассчитанного на время от 1 с до 30 с небольшим минут. Неиспользуемые входы (но не выходы!) микросхем следует заземлить.

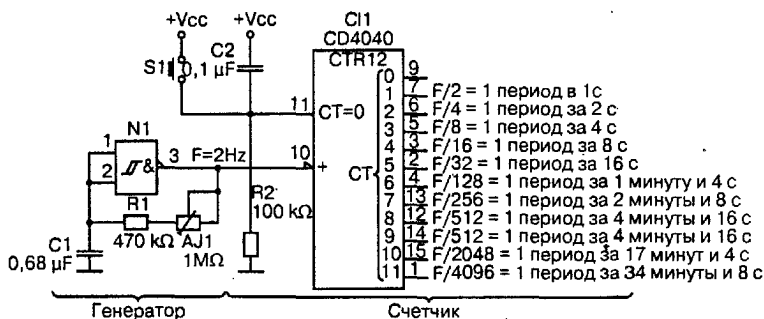


Рис. 3.15. Применение микросхемы CD4040 – 12-разрядного асинхронного счетчика

Схема с применением CD4060

Микросхема CD4060 отличается от предыдущей схемы CD4040 большим коэффициентом деления – 2^{14} вместо 2^{12} и наличием встроенного (не внешнего) генератора, к которому идут внешние времязадающие цепи, подключаемые ко входам CX, RX и RCX по схеме RC-генератора (схема в рамке), так и кварцевого генератора – резонатора с «обвязкой» (основная схема). Генератор импульсов подключен ко входу схемы счетчика-делителя. Количество выводов микросхем CD4060 и CD4040

одинаково – по 16 выводов. Поэтому выходы разрядов Q0–Q2 и Q10 не выведены наружу. Выведены выходы Q3–Q9 и Q11–Q13, первый (Q3) является делителем основной частоты на 16, последний (Q13) является делителем частоты на 16384 или 2^{14} . На рис. 3.16 представлена схема звуковой сигнализации со своеобразным звучанием: частота 1,024 кГц, модулированная частотой 128 Гц звучит с прерыванием в 2 Гц*. Генератор импульсов водключен ко входу схемы счетчика-делителя.

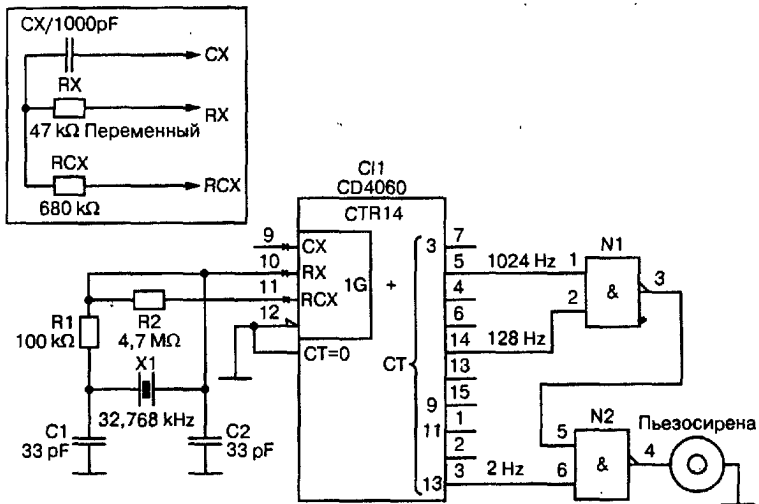


Рис. 3.16. Применение микросхемы CD4060 в качестве звукового сигнализатора

Для увеличения стабильности генератор импульсов собран на кварцевом резонаторе X1 частотой 2^{15} или 32,768 кГц и дополнительных элементах (R1, R2 и C1, C2). В менее требовательных случаях к стабильности частоты возможно также собрать его с помощью RC-цепи, показанной в верхней части схемы в рамке. Тогда формула расчета частоты импульсов запишется следующим образом:

$$F = \frac{1}{2,3 \times RX \times CX} \text{ и } RCX = (2 - 6)RX;$$

* Звучание своеобразной лягушки-квакушки. – Прим. науч. ред.

Подставляя в формулу для определения F числовые значения R_X и C_X , получим:

$$F = \frac{1}{2,3 \times 0,012 \times 0,001} = 36231 \text{ Гц}$$

Сигнал с выходов Q4 и Q7 микросхемы С11 подается на входы логического элемента И-НЕ N1. Далее полученный выходной модулированный выходной сигнал поступает на вход элемента И-НЕ N2 и комбинируется с сигналом частотой 2 Гц, который подается с выхода Q13. На выходе элемента N2, таким образом, присутствует двухчастотный сигнал (1024 Гц и 128 Гц). Такое звучание будет происходить только два раза в секунду, так как сигнал с выхода Q13 CD4060 открывает N2 с такой частотой. Неиспользуемые входы микросхем следует заземлить.

Схема с применением CD4029

Микросхема CD4029 является 4-разрядным реверсивным счетчиком, то есть работать как на сложение, так на вычитание. Кроме того, она может работать как двоичный реверсивный счетчик, так и двоично-десятичный реверсивный счетчик и содержит также 4 входа предустановки. Схема на рис. 3.17 представляет схему реверсивного счетчика с преобразователем двоично-десятичного кода в 7-сегментный код. Выходы сегментного кода через токоограничительные резисторы R4-R10 подключены к соответствующим сегментам 7-сегментного индикатора, который индицирует цифры от 0 до 9 в виде комбинации зажигаемых сегментов a...g.

При нажатии на кнопку S1 на выходе одновибратора вырабатывается положительный импульс. Одновибратор построен на двух элементах ИЛИ-НЕ. Его работа была описана ранее. Ключ S2 позволяет выбирать направление счета: вывод U (U_p – вверх) – на сложение, вывод D (Down – вниз) – на вычитание. Интегрирующая RC-цепь, составленная из конденсатора C2 и резистора R3, обнуляет счетчик при подаче напряжения. Вывод 9 микросхемы CD4029 заземлен при двоично-десятичном счете (от 0 до 9) и подключен к $+V_{cc}$ через резистор от 10 кОм – тогда счетчик работает как двоичный.

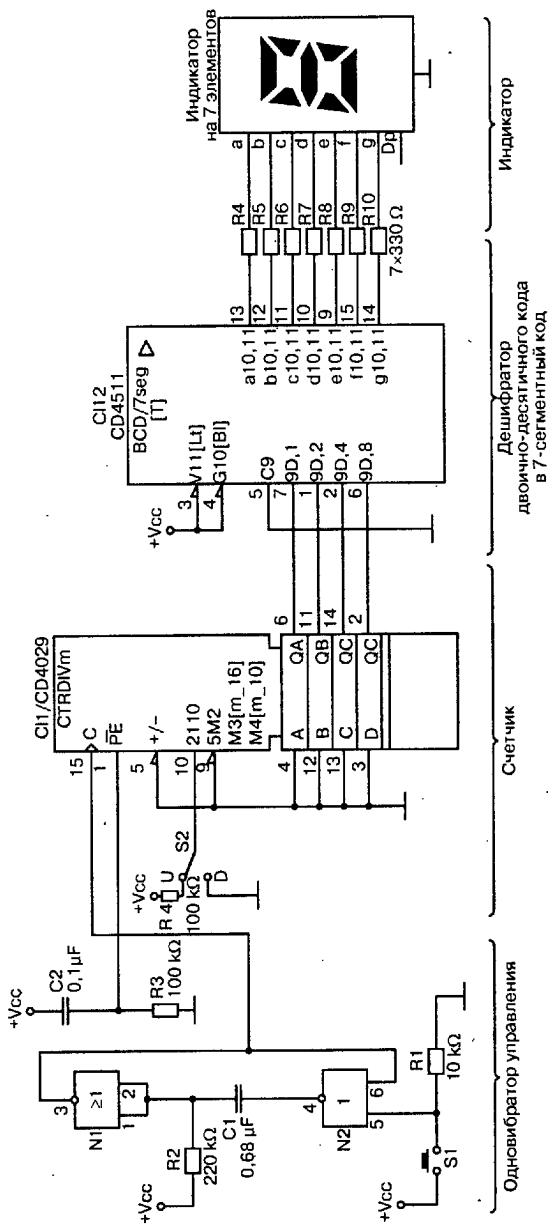


Рис. 3.17. Применение микросхем CD4029 и CD4511 совместно с 7-сегментным индикатором

Также незадействованные входы необходимо подключить к «земле». Четыре двоичных выхода разрядов соединены с дешифратором CD4511, в функцию которого входит в соответствии с двоично-десятичным кодом подавать питание на сегменты индикатора для отображения соответствующей коду цифры от 0 до 9.

3.8. Десятичный счетчик

Основные положения

Десятичный счетчик функционирует таким же образом, что и другие счетчики, работающие в одну сторону на сложение. Разница заключается в том, что он имеет 10 дешифрованных выходов (0–9). Он состоит из 5-каскадного счетчика Джонсона и дешифратора на 10 выходов, который преобразует код Джонсона в единичный сигнал только на одном из выходов: при появлении на входе импульса счетчик увеличивает свое состояние на один и появляется на следующем выходе. Таким образом, одновременно активен только один выход.

Схема с применением CD4017

Микросхема CD4017 представляет собой десятичный счетчик Джонсона (Johnson) с дешифратором на 10 выходов. Схема на рис. 3.18 показывает пример использования, в которой реле срабатывает после определенного числа импульсов и останавливается после следующего действия.

Одновибратор, схему которого вы уже хорошо знаете, управляет по входу разрешения (вывод 14) работой счетчика С11. Цепь из конденсатора С2 и резистора R3 осуществляют сброс счетчика в нулевое состояние при включении питания. Мы выбрали выход 6 (вывод 5) для управления реле. Реле включится после 6 нажатий на кнопку S1. Выход 7 (вывод 6) через диод D1 переводит счетчик в нулевое состояние. Усиление тока для включения обмотки реле осуществляет транзистор T1, управляемый с выхода 6 счетчика-делителя через ограничивающий ток базы резистор R4. Резистор R5

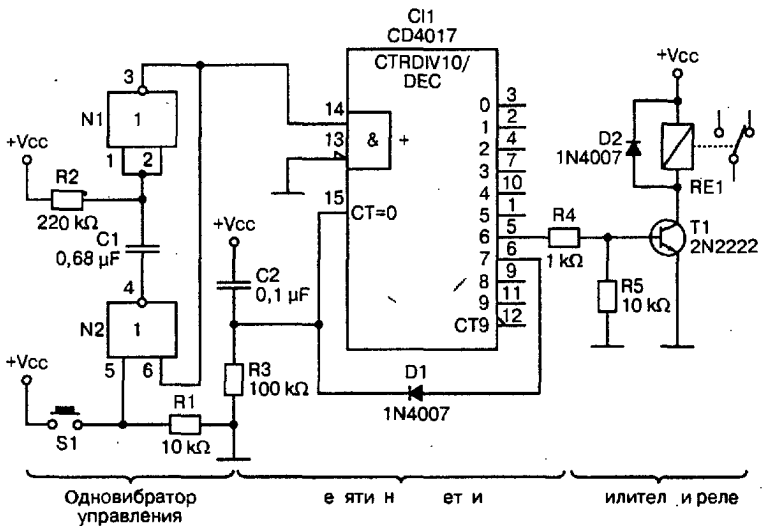


Рис. 3.18. Применение микросхемы CD4017 для управления реле

задает параметры транзистора T1 по постоянному току. Диод D2 защищает T1 от токов пробоя, возникающих в обмотке реле при переключении транзистора из одного состояния в другое.

Глава 4

Логический имитатор на основе микросхем

4.1. Краткое представление содержания главы 4

Познакомившись с логическими схемами, вы уже в состоянии изготовить устройство, способное имитировать до восьми логических функций на основе трех микросхем КМОП. Чтобы не усложнять эту работу, число входов ограничено до двух

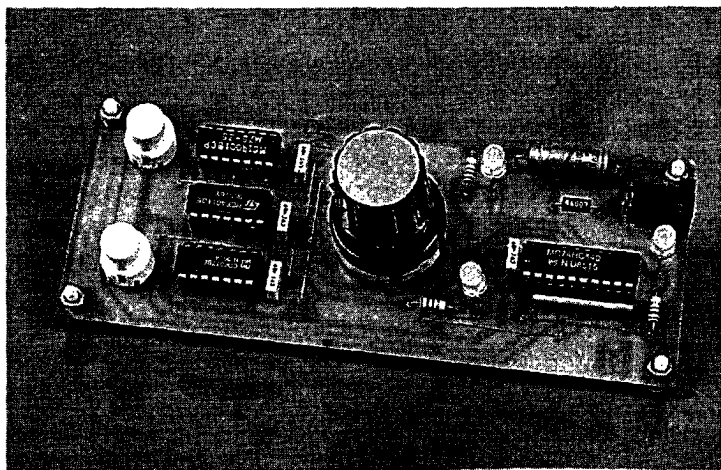


Рис. 4.1. Логический имитатор на основе микросхем

для каждого выхода. Состояние входов, изменяемое с помощью кнопок, отображается на светодиодах. Результат выхода также подается на светодиод. Это самостоятельное устройство является ценным учебным материалом по закреплению знаний таблиц истинности основных логических схем.

4.2. Принципиальная схема логического имитатора

Вначале рассмотрим, каким образом реализуются 8 логических функций (рис. 4.2):

- функция НЕ образуется на основе элемента ИЛИ-НЕ N2, где оба входа которого соединены между собой;
- функция ДА образуется двойной инверсией от НЕ (элемент N2) и второго инвертора на элементе ИЛИ-НЕ N1, объединенного по входам;
- функция ИЛИ-НЕ образуется просто с помощью соответствующей ей элемента ИЛИ-НЕ N4;
- функция ИЛИ образуется инверсией от ИЛИ-НЕ с помощью элемента ИЛИ-НЕ N3;
- функция И-НЕ образуется прямо от элемента И-НЕ N6;
- функция И образуется инверсией от И-НЕ N6 с помощью элемента И-НЕ N5;
- функция ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ образуется прямо с помощью элемента N9;
- функция ИСКЛЮЧАЮЩЕЕ ИЛИ образуется инверсией от ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ N9 с помощью элемента N7.

Как вы уже заметили, в схеме логического имитатора используются только три микросхемы. Элементы N1–N4 выполняют функцию ИЛИ-НЕ, используя микросхему CD4001, элементы N5–N8 выполняют функцию И-НЕ с помощью микросхемы CD4011, и, наконец, элементы N9–N12 выполняют функцию ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ на микросхеме CD4077. При установке независимых микросхем для каждой функции потребовалось бы девять микросхем вместо четырех. Мы же использовали свободные элементы как инверторы, тем самым сократили число необходимых микросхем до приемлемых четырех.

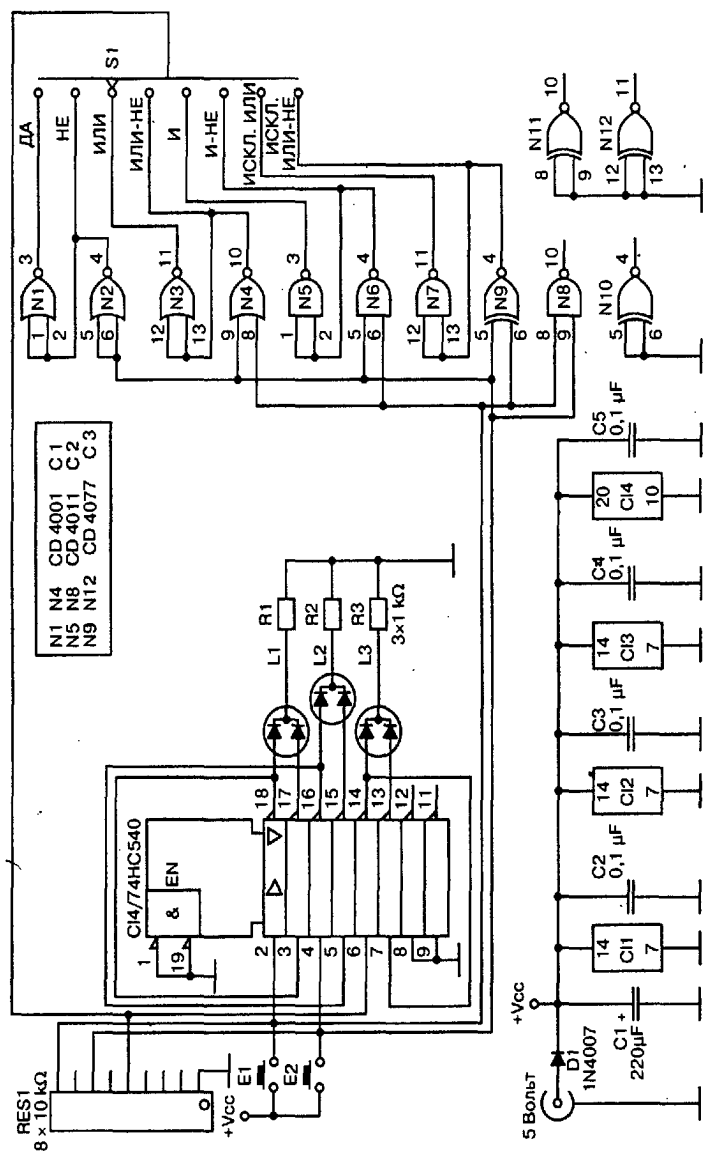


Рис. 4.2. Принципиальная схема логического имитатора

Входы E1 всех элементов связаны между собой, чтобы ими можно было управлять параллельно и всеми сразу. То же самое сделано и для входов E2. В состоянии покоя входы соединены с «землей» через резисторную сборку RES1. При нажатии кнопок на входы подается логическая единица, соответствующая напряжению $+V_{cc}$.

С помощью кнопки S1 можно выбрать требуемую логическую функцию. Результат выхода отображается на светодиоде L3. Его работа идентична схеме прибора для тестирования логических элементов, описанной ранее. Индикация входных схем использует этот же принцип на светодиодах L1 и L2.

Микросхема 74HC540 (CI4) включает восемь инверторов, из которых нами будут использованы шесть и контрольные выводы. Контрольные выводы 1 и 19 соединены с «землей», также как и неиспользуемые входы 8 и 9. Каждая схема индикации использует по два инвертора. Рассмотрим схему, активирующую светодиод L1. Если кнопка E1 нажата, то на первый элемент НЕ (вывод 2) подается логическая единица. На его выходе (вывод 18) будет логический ноль, и на положительном выводе – аноде красного светодиода, соединенного с выводом 18, будет также ноль, а, будучи заземленными отрицательными выводами – катодами всех трех светодиодов, между анодом и катодом светодиода L1 не будет разности напряжения, и светодиод не будет светиться. Тогда сигнал подается на вход второго инвертора (вывод 3). На его выходе (вывод 17) будет высокое напряжение, которое подается на анод зеленого светодиода L1. Светодиод загорается, показывая уровень логической единицы на входе. Если отпустить кнопку E1, вход первого инвертора подключится к «земле» и ситуация поменяется на противоположную. Зеленый светодиод погаснет, и загорится красный, показывая уровень логического нуля на входе. Резисторы R1–R3 ограничивают соответственно токи светодиодов L1–L3 до допустимых значений для данной микросхемы.

Напряжение питания находится в пределах от 4,5 до 5 В. Диод D1 защищает схему от неправильного подключения питания – так называемой переполюсовки, или по-другому защиты от дурака (радиолюбители тоже шутят), конденсатор C1 – фильтр от низкочастотных флюктуаций напряжения питания. Конденсаторы C2–C5 фильтруют соответственно

напряжение питания микросхем С11–С14 по высокой частоте. Неиспользуемые входы микросхем С12 и С13, как всегда, будут заземлены.

4.3. Сборка логического имитатора

На рис. 4.3 показан чертеж печатной платы имитатора. Схема содержит большое количество перемычек (их 9) в связи с трудностями по трассировке платы между пластинами поворотного переключателя функций.

Если вы решили изготовить рисунок платы с помощью клейкой бумаги и ручки, то будьте терпеливы и убедитесь, что на плате нет никаких дефектов. Наилучшим методом является, безусловно, фотолитография. В библиографии в конце книги вы найдете конкретные источники по технике литографии.

Монтажная схема представлена на рис. 4.4. Начните монтаж с пайки девяти перемычек.

Детали следует запаивать на плату в очередности, определяемой размерами и хрупкостью элементов. Придерживайтесь следующего порядка: резисторы, диод, держатели микросхем, резисторная сборка, керамические конденсаторы, кнопки управления, электролитический конденсатор, разъем подключения питания, светодиоды, выравненные по высоте; и, наконец, поворотный переключатель функций, рассчитанный на 12 позиций и отрегулированный на 8 позиций с помощью специальной пластины под гайкой, выполняющей роль фиксатора.

Включение и использование

Возьмите за абсолютную аксиому – никогда не включайте новую плату без предварительной проверки. Короткое замыкание дорожек может мгновенно разрушить элементы в прямом смысле – резисторы почернеют и разломаются. А в другом случае – это касается неправильно подключенных полярных конденсаторов – могут взрываться и даже летать. Основная масса компонентов выйдет из строя, но внешне это будет незаметно. Тем сложнее искать неисправность из-за непростительной поспешности. Проверьте положение и полярность элементов (касается диодов и полярных конденсаторов), тип и номинал каждой детали в соответствии с

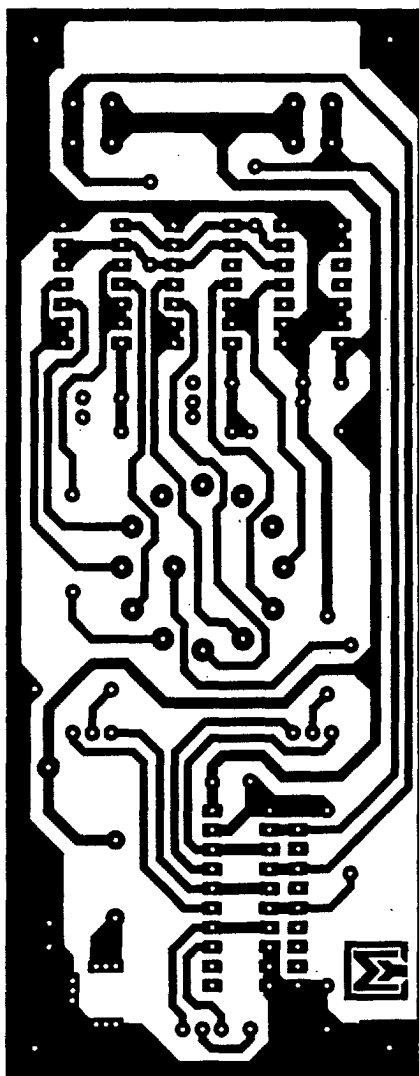


Рис. 4.3. Чертеж печатной платы имитатора

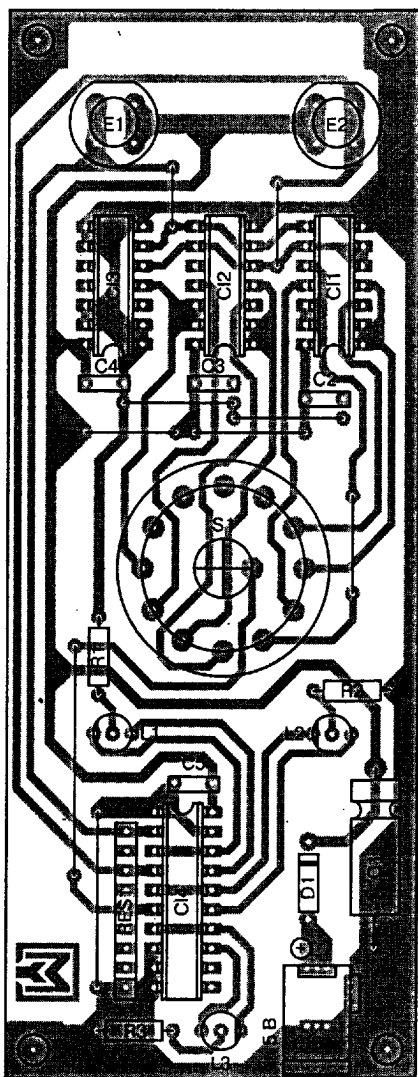


Рис. 4.4. Монтаж элементов на плате

электрической схемой. Только после этого можете подключить питание к схеме, не вставляя микросхемы в держатели. Далее при помощи цифрового мультиметра в режиме вольтметра проверьте напряжение питания на выводах держателей микросхем (красный провод «+» щупа мультиметра подсоедините к выводу 14 микросхем С11–С13 и к выводу 20 С14; черный провод «-» щупа подсоедините к выводу 7 микросхем С11–С13 и к выводу 10 С14), оно должно быть 4,5 В. При этом два входных светодиода должны гореть красным цветом.

После этого отключите напряжение питания и вставьте микросхемы. С помощью переключателя выберите любую логическую функцию. Светодиод L3 является индикатором выхода. Зеленое свечение светодиода указывает на высокий уровень, красное свечение сообщает о низком уровне. Кнопки управления и переключатель можно использовать в любое время для имитации логических функций. При проведении экспериментов сравнивайте данные практических опытов с изученными таблицами истинности.

Список компонентов:

Резисторы с 5-процентным допуском:

- R1-R3: 1 кОм (коричневый, черный, красный);
- RES1: резисторная сборка 8 × 10 кОм.

Конденсаторы:

- C1: 220–470 пФ/16 В (полярный электролитический с аксиальными выводами);
- C2–C5: 0,1 мкФ неполярные керамические.

Полупроводники:

- L1–L3: двухцветные светодиоды с 3 выводами;
- D1: диод 1N4007;
- С11: ИМС CD4001;

- CI2: ИМС CD4011;
- CI3: ИМС CD4077;
- CI4: ИМС 74НС540.

Прочие элементы:

- 3 держателя микросхемы с 14 выводами;
- 1 держатель микросхемы с 20 выводами;
- 1 переключатель поворотный для микросхемы на 14 позиций;
- 1 однополюсный выключатель (по желанию);
- 2 контактные кнопки;
- 1 разъем подключения питания диаметром 2,1 мм;
- болты и стойки диаметром 3 мм.

Глава 5

Программный имитатор на PC

5.1. Краткое представление содержания главы 5

Имитатор, который мы изучили в прошлой главе, ограничен двумя входами для каждого выхода и восемью функциями. Любое расширение возможностей аппаратных имитаторов, как правило, приводит к усложнению схемы, увеличению количества микросхем и, как следствие этого, ведет к повышению себестоимости. При использовании персонального компьютера (PC), выполняющего те же функции, но программно, наше устройство может стать более эффективным (рис. 5.1). В новой схеме присутствует четыре входа, что выражается в некотором усложнении устройства, но и приводит к лучшему пониманию логики. Кроме того, число логических функций в имитаторе зависит только от программы управления и не потребует новых микросхем. Мы увидим, что в программе QBASIC можно создать программу, эффективно управляющую установкой. Лучше использовать последние версии языка программирования WINDOWS. К восьми основным функциям мы добавили специальную, соединяющую в себе множественное И совместно с ИЛИ.

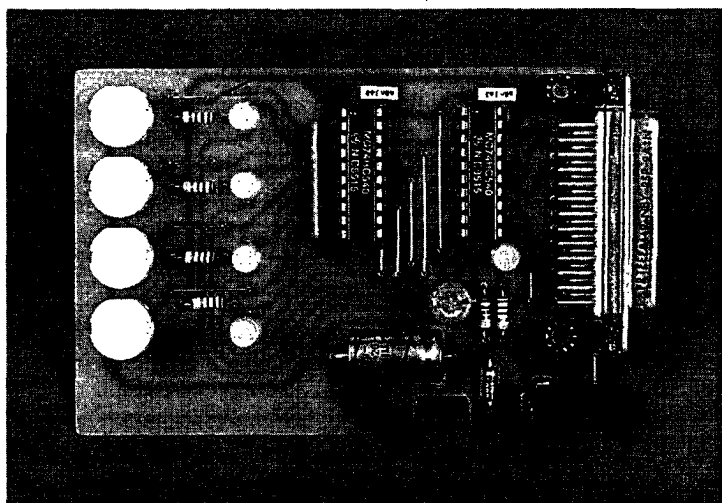


Рис. 5.1. Логический имитатор, управляемый с ПК

5.2. Принципиальная схема программного имитатора

Большая разница между схемой на рис. 5.2 и схемой логического имитатора, рассмотренного в предыдущей главе, заключается в обработке данных при нажатии кнопок. В данном устройстве – программном имитаторе – большинство элементов предназначено для индикации состояния входов и выхода.

Мы будем использовать четыре входных канала (ERROR, SELECT, PAPER END, ACK) и один выход (D0) параллельного порта. Каждый блок индикации работает на основе двух инверторов, интегрированных в микросхему 74HC540. Этот принцип применялся в предыдущих схемах. Рассмотрим работу индикатора кнопки E1. В состоянии покоя вход 8 инвертора заземлен с помощью резистивной сборки RES1. Тогда на выходе (вывод 12) будет логическая единица, которая подастся на анод красного светодиода L1. Светодиод загорается под действием положительной разности потенциалов между

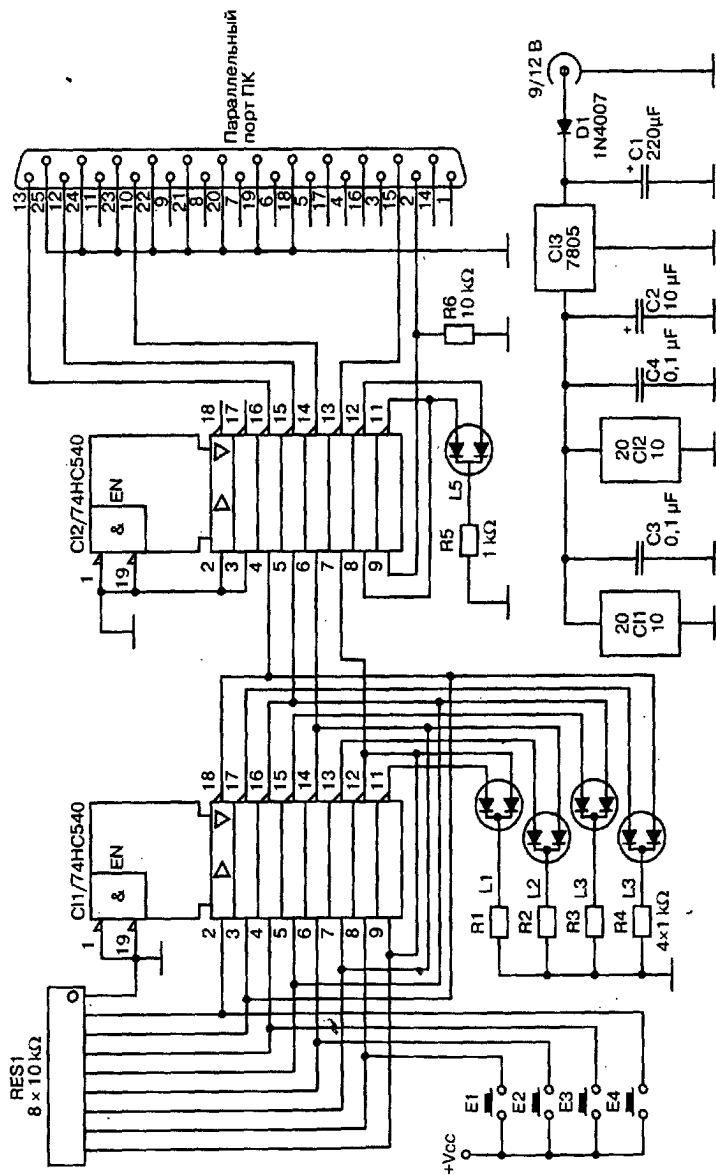


Рис. 5.2. Принципиальная схема программного имитатора

анодом и катодом, что свидетельствует о наличии логического нуля на входе схемы. Кроме того, с выхода инвертора сигнал высокого уровня подается на вход второго инвертора (вывод 9), инвертируется в сигнал с низким уровнем и прикладывается к аноду зеленого светодиода L1. Также с выхода 12 сигнал логической единицы подается на вход 7 инвертора второй микросхемы 74НС540 и устанавливает нуль на его выходе (вывод 13) и соединенном с ним выводе 15 разъема DB25 параллельного порта ПК, называемого каналом ERROR.

При нажатии на кнопку E1 на вход 8 инвертора подается логическая единица, красный светодиод гаснет, загорается зеленый светодиод, и на канале ERROR устанавливается логическая единица. Точно такой же принцип используется и для других трех кнопок E2–E4, а также для светодиода L5, управляемого выходом D0 параллельного порта. Два инвертора микросхемы CI2 не используются, и их входы (выводы 2 и 3) заземляются. Резистор R6 играет ту же роль, что и резистивная сборка RES1: он заземляет выход 9 инвертора CI2 канала D0 параллельного порта, когда плата отсоединена от компьютера. Резисторы R1–R5, установленные между катодами и «землей», ограничивают ток светодиодов L1–L5 до номинальных значений, которые меньше предельно допустимых для них.

Из соображений безопасности напряжение питания +5 В подается на питание микросхем через стабилизатор напряжения 7805, который со своего выхода вырабатывает напряжение +5 В. На вход стабилизатора поступает повышенное напряжение +9 или +12 В от источника внешнего напряжения. В самом деле на параллельный порт компьютера PC нельзя подать напряжение больше 5 В. Диод D1 защищает вход стабилизатора от переплюсовки по питанию – уже известная «защита от дурака». Конденсатор C1 осуществляет первичную фильтрацию напряжения перед подачей его на стабилизатор напряжения CI3. На его выходе вырабатывается стабилизированное напряжение 5 В. Конденсатор C2 фильтрует это напряжение перед подачей на микросхемы. Конденсаторы C3 и C4 развязывают микросхемы CI1 и CI2 от высокочастотных помех по питанию.

5.3. Сборка

На рис. 5.3 показан чертеж печатной платы. Так эта плата будет подключаться к компьютеру, следует отнестись к ее изготовлению с большой тщательностью.

Имитатор соединяется с ПК через разъем DB25, и других трудностей в данной схеме не предвидится. Поэтому обязательно точное согласование логических сигналов имитатора и ПК, иначе ПК выйдет из строя. Второе требование – сначала включается питание, затем подается сигнал. При выключении порядок следования обратный.

Вы можете изготовить печатную плату любым способом на ваш выбор. Самым легким является метод фотолитографии. Вы можете найти в библиографии в конце книги материалы по технике литографии.

Монтажная схема представлена на рис. 5.4. Начните монтаж с пайки десяти перемычек.

Компоненты следует запаивать на плату в следующей очередности: от малых по размерам к большим и по возрастанию

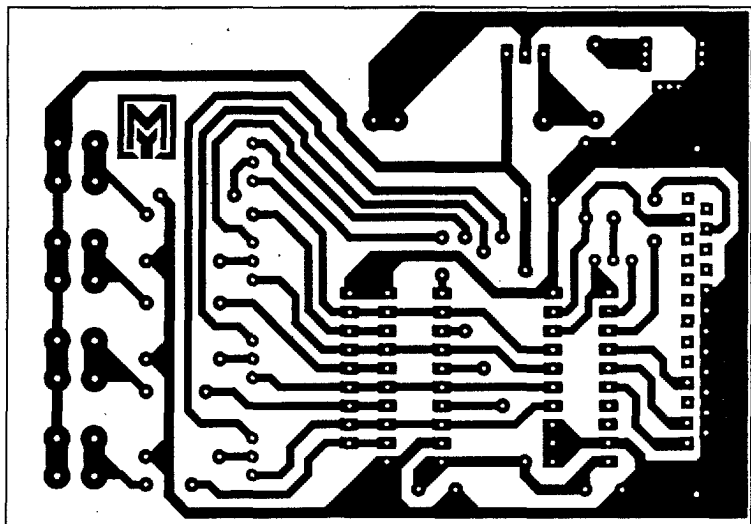


Рис. 5.3. Чертеж разводки печатной платы

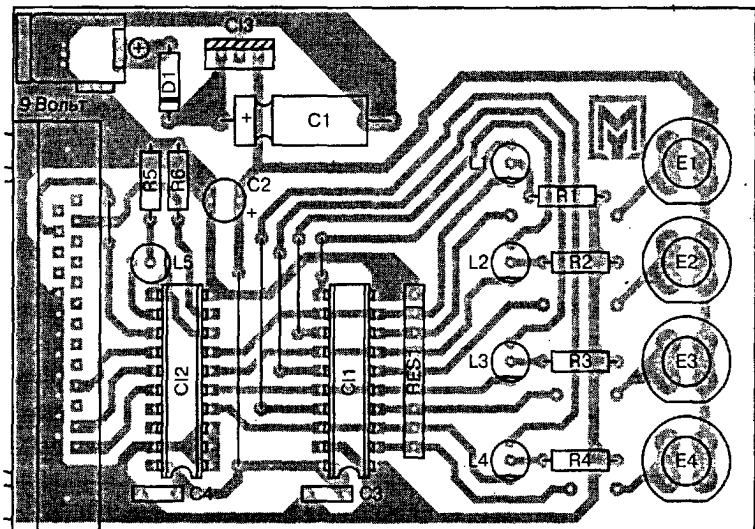


Рис. 5.4. Размещение компонентов имитатора на плате

степени прочности элементов. Придерживайтесь следующего порядка: резисторы, диод, держатели микросхем, резистивная сборка, керамические конденсаторы, кнопки управления, разъем подключения питания, разъем связи с ПК DB25, электролитические конденсаторы, стабилизатор C13 для согласования по питанию с компьютером и светодиоды, выравненные по высоте.

Включение

После запаивания последнего элемента следует все проверить перед подключением напряжения питания. Короткое замыкание дорожек может разрушить элементы и испортить параллельный порт компьютера, поэтому следуйте рекомендациям в последовательности: сначала найдите с помощью лупы и исключите с помощью паяльника короткое замыкание, если оно имеется, иначе плата не работоспособна. Проверьте положение элементов (соответствие ключей-выемок микросхем C11, C12 на их одной короткой стороне по чертежу и установленных на печатной плате – ключи должны быть одинаково

ориентированы, полярность диода D1, светодиодов L1–L5, полярность и номиналы электролитических конденсаторов C1, C2 – их не нужно путать между собой) и номиналы каждого радиоэлемента. Установленные радиоэлементы должны неукоснительно соответствовать электрической схеме. Только после этого можете подсоединить напряжение питания 9 или 12 В (от батареи или преобразователя) ко входу схемы стабилизатора напряжения 5 В без установки микросхем в держатели и не подсоединяя плату к компьютеру. При помощи цифрового мультиметра в режиме вольтметра проверьте напряжение питания на выводах питания держателей микросхем (красный провод «+» подсоедините к выводу 20 микросхем; черный провод «–» подсоедините к выводу 10 микросхем), оно должно быть 5 В. При этом все светодиоды должны гореть красным цветом.

После этого отключите напряжение питания, вставьте микросхемы и подсоедините плату к компьютеру. Снова подключив питание, вы можете проверить работу входов. При нажатии на кнопки E1–E4 соответствующие светодиоды начинают светиться зеленым светом. Светодиод, определяющий состояние выхода, загорается в зависимости от программы, которую вы предварительно ввели в компьютер.

Список компонентов:

Резисторы с 5-процентным допуском:

- R1–R5: 1 кОм (коричневый, черный, красный);
- R6: 10 кОм (коричневый, черный, оранжевый);
- RES1: резисторная сборка 8 Ч 10 кОм.

Конденсаторы:

- C1: 220–470 пФ/ 16 В (полярный электролитический с аксиальными выводами);
- C2: 10–47 мкФ/ 16 В (полярный электролитический с аксиальными выводами);
- C3, C4: 0,1 мкФ неполярные керамические.

Полупроводники:

- L1-L5: двухцветные светодиоды с тремя выводами;
- D1: диод 1N4007;
- C11, C12: ИМС 74НС540;
- C13: Стабилизатор напряжения 7805.

Прочие элементы:

- 2 держателя 20-выводных микросхем;
- 1 штырьковый разъем DB25 для подключения к параллельному порту компьютера;
- 4 кнопки на замыкание;
- 1 разъем подключения питания диаметром 2,1 мм;
- болты и стойки диаметром 3 мм.

Программирование

Каким бы ни было программное обеспечение, заставляющее функционировать ваш имитатор, оно должно отвечать следующим требованиям:

- выбор логической функции;
- считывание четырех входов по каналам параллельного порта ERROR, SELECT, PAPER END, ACK;
- обработка информации;
- вывод информации по каналу D0 параллельного порта на светодиод L5;
- включение программы.

В качестве примера вам предлагается листинг 5.1 программы `simulog.bas`, написанной на QBASIC.

Листинг 5.1. Программа работы логического имитатора на ПК в QBASIC

```
REM === ПРОГРАММА РАБОТЫ ЛОГИЧЕСКОГО ИМИТАТОРА НА ПК В QBASIC ===
REM === Для параллельного порта LPT1. © Yves XERGY 2001 ===

' === Объявление подпрограммы или процедуры. ===
DECLARE SUB LECTURE ( )

' === Отображение главного экрана для выполнения выбора. ===
' === Стирание экрана в начале.
```

CLS

```

' Для каждой из следующих строк мы имеем две команды:
' первая команда устанавливает курсор в желаемое место,
' а вторая отображает текст.
LOCATE 3, 27: PRINT "ЛОГИЧЕСКИЙ ИМИТАТОР ДЛЯ ПК"
LOCATE 5, 5: PRINT "Выберите функцию нажатием на клавишу (0 ...
8) на клавиатуре"
LOCATE 7, 10: PRINT "- 1 ==> функция ДА (однократным нажатием
клавиши E1 имитатора)"
LOCATE 8, 10: PRINT "- 2 ==> функция НЕТ (двойным нажатием
клавиши E1 имитатора)"
LOCATE 9, 10: PRINT "- 3 ==> функция ИЛИ (нажатием четырех
клавиш имитатора)"
LOCATE 10, 10: PRINT "- 4 ==> функция НЕ-ИЛИ (нажатием четырех
клавиш имитатора)"
LOCATE 11, 10: PRINT "- 5 ==> функция И (нажатием четырех
клавиш имитатора)"
LOCATE 12, 10: PRINT "- 6 ==> функция НЕ-И (нажатием четырех
клавиш имитатора)"
LOCATE 13, 10: PRINT "- 7 ==> функция ИСКЛЮЧАЮЩЕЕ ИЛИ (нажатием
четырех клавиш имитатора)"
LOCATE 14, 10: PRINT "- 8 ==> функция ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ
(нажатием четырех клавиш имитатора)"
LOCATE 16, 10: PRINT "- 0 ==> СПЕЦИАЛЬНАЯ функция (нажатием
четырех клавиш имитатора):"
LOCATE 17, 22: PRINT "комбинированная функция И (нажатием E1 ...
E3)"
LOCATE 18, 22: PRINT "с функцией ИЛИ (на клавише E4)"
LOCATE 20, 18: PRINT "НАЖМИТЕ НА 'Q' ДЛЯ ВЫХОДА ИЗ ПРОГРАММЫ"

НАЧАЛО:           ' Метка возврата перед первым выбором.
C$ = INKEY$      ' Испытательное нажатие на любую клавишу ПК.
ВЫБОР:           ' Метка возврата для последующих выборов.
LOCATE 23, 30: PRINT "Ваш выбор: "; C$
' Подключение в зависимости от выбранной функции или выход.
IF C$ = "1" THEN GOTO ДА
IF C$ = "2" THEN GOTO НЕТ
IF C$ = "3" THEN GOTO ИЛИ
IF C$ = "4" THEN GOTO ИЛИ-НЕ
IF C$ = "6" THEN GOTO И-НЕ
IF C$ = "7" THEN GOTO ИСКЛЮЧАЮЩЕЕ ИЛИ
IF C$ = "8" THEN GOTO ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ

IF C$ = "0" THEN GOTO SPEC
IF C$ = "Q" THEN GOTO END
' Возврат в начало в случае отсутствия выбора
GOTO НАЧАЛО

```

' Для каждой функции:
 ' вызов процедуры чтения входных данных,
 ' обработка выхода (D0) по адресу 888 (порт LPT1),
 ' OUT (888),1 ==> подключение линии D0 к положительному
 ' напряжению +V_{cc},
 ' OUT (888),0 ==> подключение линии D0 к «земле»,
 ' тестирует выполнение другой выбранной функции,
 ' в противном случае функция закичивается на своей
 ' первой команде.

ДА:

ЧТЕНИЕ

IF E1 = 1 THEN OUT (888), 1 ELSE OUT (888), 0

C\$ = INKEY\$

IF C\$ <> "" THEN GOTO ВЫБОР ELSE GOTO ДА

НЕТ:

ЧТЕНИЕ

IF E1 = 0 THEN OUT (888), 1 ELSE OUT (888), 0

C\$ = INKEY\$

IF C\$ <> "" THEN GOTO ВЫБОР ELSE GOTO НЕТ

ИЛИ:

ЧТЕНИЕ

IF (E1 = 1 OR E2 = 1 OR E3 = 1 OR E4 = 1) THEN OUT (888), 1

ELSE

OUT (888), 0

C\$ = INKEY\$

IF C\$ <> "" THEN GOTO ВЫБОР ELSE GOTO ИЛИ

НЕ-ИЛИ:

ЧТЕНИЕ

IF (E1 = 1 OR E2 = 1 OR E3 = 1 OR E4 = 1) THEN OUT (888), 0

ELSE

OUT (888), 1

C\$ = INKEY\$

IF C\$ <> "" THEN GOTO ВЫБОР ELSE GOTO НЕ-ИЛИ

И:

ЧТЕНИЕ

IF (E1 = 1 AND E2 = 1 AND E3 = 1 AND E4 = 1) THEN OUT (888), 1 ELSE

OUT (888), 0

C\$ = INKEY\$

IF C\$ <> "" THEN GOTO ВЫБОР ELSE GOTO И

НЕ-И:

ЧТЕНИЕ

IF (E1 = 1 AND E2 = 1 AND E3 = 1 AND E4 = 1) THEN OUT (888), 0 ELSE

OUT (888), 1

C\$ = INKEY\$

IF C\$ <> "" THEN GOTO ВЫБОР ELSE GOTO НЕ-И

ИСКЛЮЧАЮЩЕЕ ИЛИ:

ЧТЕНИЕ

' Расчет входных значений после возврата
' из процедуры чтения, значение должно быть равным 1
' для подтверждения активности только одного входа.

TOTAL = E1 + E2 - E3 + E4

IF TOTAL = 1 THEN OUT (888), 1 ELSE OUT (888), 0

CS = INKEYS

IF CS <> "" THEN GOTO ВЫБОР ELSE GOTO ИСКЛЮЧАЮЩЕЕ ИЛИ

ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ:

ЧТЕНИЕ

TOTAL = E1 + E2 + E3 + E4

IF TOTAL = 1 THEN OUT (888), 0 ELSE OUT (888), 1

CS = INKEYS

IF CS <> "" THEN GOTO ВЫБОР ELSE GOTO ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ

SPEC

ЧТЕНИЕ

IF (E1 = 1 AND E2 = 1 AND E3 = 1) OR (E4 = 1) THEN OUT
(888), 1

ELSE OUT (888), 0

CS = INKEYS

IF CS <> "" THEN GOTO ВЫБОР ELSE GOTO SPEC

ЗАМЕЩЕНИЕ ЧТЕНИЯ

' Процедура чтения клавиш имитатора:
' переменные объявляются для всей программы,
' чтение входов по адресу 889 (порт LPT1),
' и присвоение значения в зависимости от их состояния.

SHARED E1, E2, E3, E4

A = INP(889)

IF (A AND 8) = 8 THEN E1 = 1 ELSE E1 = 0

IF (A AND 16) = 16 THEN E4 = 1 ELSE E4 = 0

IF (A AND 32) = 32 THEN E3 = 1 ELSE E3 = 0

IF (A AND 64) = 64 THEN E2 = 1 ELSE E2 = 0

END SUB

Часть II

Цифровая электроника

В первой части этой книги были описаны непрограммируемые элементы, другими словами, логические схемы на «жесткой логике». Когда мы начали изготовление программируемого устройства, для работы с ним потребовался компьютер. Наши схемы программировались с помощью компьютера и были неразрывно с ним связаны. Таким образом, мы перешли к *схемам с программируемой логикой*. Вторая часть книги посвящена цифровой электронике. Мы будем использовать в работе компьютер, но не только для программирования компонентов, но и для разработки различных устройств. После завершения программирования устройства станут самоуправляющимися.

Глава 6

Программируемые элементы

6.1. Запоминающие устройства

Как указывает название, *запоминающие устройства (ЗУ)*, используются для хранения информации. Они состоят из некоторого числа ячеек памяти. Например, D-триггер является простейшей элементарной ячейкой памяти в статических запоминающих элементах. Такая единичная ячейка памяти называется *битом* и может представляться логической единицей или логическим нулем. Бит может принимать только два значения: 1 или 0. Чтобы работать с большими значениями ячеек памяти, принята более крупная единица памяти в 8 бит, которая носит обозначение *байт*. 1 байт может сохранить числовое значение 2^8 – от 0 до 255 (см. основы двоичного счисления в начале этой книги).

Объем памяти измеряется в разных величинах: битах, байтах, килобайтах (Кб), мегабайтах (Мб) (последние величины являются кратными от байта). Запомните, что один килобайт равен не 1000 битам, а 1024 бита (ввиду того, что основанием степени двоичной системы является 2); таким же образом, один мегабайт равен 1 048 576 байтам.

Когда нужно записать или сосчитать значение одного байта памяти, и если число ячеек измеряется тысячами, возникает ситуация, когда надо сделать это точно и быстро. Применяемая методика называется *адресацией памяти*.

Как у жителей большого города, у каждой ячейки памяти есть свой собственный один единственный уникальный адрес. Речь идет о ячейке, в которой находится число, принимающее одно из двух значений – или 0 или 1. В первом случае ячейка свободна; во втором случае можно считать или запрограммировать это состояние ячейки. В ячейках памяти хранятся данные, причем каждую ячейку можно найти по ее уникальному адресу. Адреса задаются двоичными числами от самого младшего адреса A0 до самого старшего, например как в следующем примере, – A12:

- вход A0 с высоким логическим уровнем позиционирует ячейку памяти с адресом 0 (первый адрес);
- вход A1 с высоким логическим уровнем позиционирует ячейку памяти с адресом 1 (второй адрес);
- входы A0 и A1 с высоким логическим уровнем позиционируют ячейку памяти с адресом 2;
- вход A2 с высоким логическим уровнем позиционирует ячейку памяти с адресом 3;
- входы A0 и A2 с высоким логическим уровнем позиционируют ячейку памяти с адресом 4;
- далее процесс повторяется по такому же принципу в соответствии с двоичным выражением до последней ячейки памяти.

Содержимое ячейки памяти представлено на выходах D0–D7 микросхемы ОЗУ в корпусе 6264 также в двоичном выражении: самый младший разряд D0 равен 1, если этот вход находится на высоком логическом уровне, то есть равен 2 в степени 0 (2^0); самый старший разряд D7 равен 128, когда на нем присутствует логическая единица, то есть 2 в степени 7 (2^7).

На рис. 6.1 показана интегральная микросхема запоминающего устройства типа 6264 емкостью 2^{13} – это 8196 байт или 65 536 бит и организацией памяти 8К × 8. Микросхема имеет двунаправленную 8-разрядную шину данных D0–D7, 13-разрядную шину адресов A0–A12. Кроме выводов данных и адреса, имеются также вывод 26 выбора кристалла (CS) – включения/выключения микросхемы, вывод 20 выбора элемента памяти (CE), вывод 22 разрешения выхода данных (OE).

вывод 27 выбора режима записи/чтения. Всего микросхема типа 6264 имеет 28 выводов. Вывод 28 – напряжение питания V_{cc} , вывод 14 – «земля».

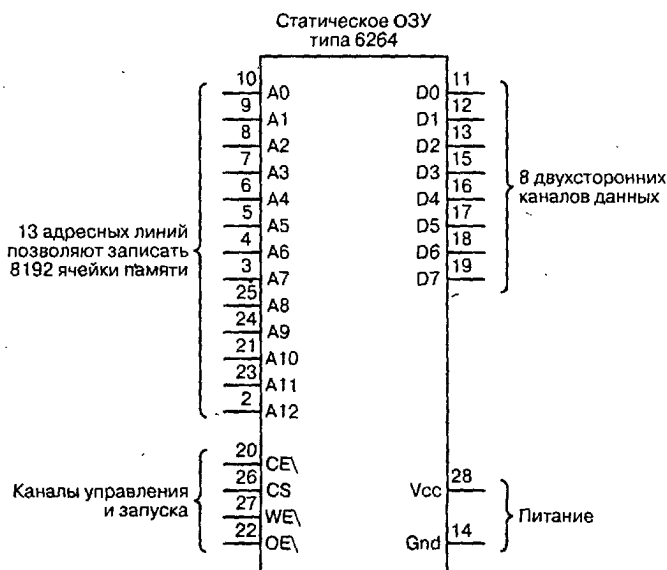


Рис. 6.1. Интегральная микросхема запоминающего устройства в корпусе 6264

Принцип работы запоминающего устройства похож на тот, который мы только что изучили. Большинство различий между разными видами запоминающих устройств заключается в способе считывания данных, записи информации и доступа к ячейкам памяти.

Статические оперативные запоминающие устройства

Статические оперативные запоминающие устройства (СОЗУ) так и называются *статическими RAM* (память с произвольным доступом – Random Access Memory), запоминающим элементом которых является триггер, потребляющий очень малую мощность в режиме хранения, если он выполнен по КМОП-технологии, поэтому статические КМОП ОЗУ резервируют энергией питания. Доступ возможен к любому адресу в

любой момент. Они легко программируются (записываются) или считываются, давая доступ к информации без дополнительных управляющих сигналов. Достаточно вначале выбрать режим работы (запись или чтение) с помощью управляющего сигнала. В случае отключения питания вся информация стирается, то есть ОЗУ являются энергозависимыми. Чтобы ОЗУ не зависели от источника питания, их подключают к дополнительному резервному питанию от батарей, которое включается при пропадании основного питания. Этот тип запоминающих устройств используется совместно с различными микроконтроллерами.

Динамические оперативные запоминающие устройства

Динамические оперативные запоминающие устройства (ДОЗУ) также программируются и считываются, предоставляя доступ к информации.

Элементарной ячейкой в динамическом ОЗУ будет заряд на конденсаторе, который нужно постоянно поддерживать (регенерировать), но при этом для корректной работы требуются так называемые сигналы регенерации или перезаписи определенной частоты. Нужно также выбирать режим работы. В случае выключения питания вся информация также стирается. Этот вид ОЗУ применяется в компьютерах под названием динамическое ОЗУ (ДОЗУ), или (DRAM - Dynamic Random Access Memory). В настоящее время этот вид ЗУ имеет самое малое время выборки, по этой причине и самую высокую производительность.

Постоянные запоминающие устройства с ультрафиолетовым стиранием

Программирование *постоянных запоминающих устройств с ультрафиолетовым стиранием (УФППЗУ)* осуществляется при большем напряжении (12,5, 21 или 25 В), которое подается на специальный вывод микросхемы. Считывание данных осуществляется привычным способом. Для того чтобы стереть данные в постоянном запоминающем устройстве с ультрафиолетовым стиранием памяти (УФППЗУ, или UVPROM -

UltraViolet Programmable Read-Only Memory), требуется облучить устройство ультрафиолетовыми лучами в течение нескольких минут. Поэтому они не являются, как статические и динамические ОЗУ, быстродействующими устройствами. Верхняя часть корпуса микросхемы содержит специальное прозрачное окошко, предназначенное для многократного стирания и повторного программирования содержимого постоянного запоминающего устройства (ПЗУ). Эти элементы очень широко применяются, несмотря на техническую простоту.

Постоянные запоминающие устройства с электрическим стиранием

Здесь речь идет об устройствах памяти, способных сохранять данные при отключении питания. Для *постоянных запоминающих устройств с электрическим стиранием (ЭСПЗУ)*, или EEPROM – Electrically Erasable Programmable Read-Only Memory) данного типа памяти, в противоположность UVPROM, напряжение программирования не отличается от напряжения питания. Более того, стирание данных происходит быстрее и практичнее, так как осуществляется электрически. Элементы этого вида памяти, как и другие, имеют шины адресов и данных с общим количеством выводов от 28 до 44. Количество выводов можно значительно уменьшить (до восьми выводов) при использовании двухпроводной шины для соединения ИС между собой, используя шину I²C (ИИС или I²C – Inter IC).

Двухпроводная шина I²C состоит из адресной шины и шины данных. Адрес и данные поступают от одной микросхемы к другим, соединенных по шине I²C, в зависимости от направления передачи данных, к микросхеме или от нее. Такое использование последовательных шин адреса и данных резко уменьшает количество выводов постоянного ЗУ. Поэтому микросхемы можно изготавливать с малым количеством выводов (8 выводов независимо от емкости памяти), так как адресная шина и шина данных объединены в двухпроводную шину связи I²C. Все данные проходят по шине данных (SDA), синхронизируясь тактовым сигналом (SCI). Этот тип памяти наиболее распространен для работы с микроконтроллерами. В отличие от КЛМ и EPROM, которые являются ЗУ с произвольным

доступом и можно сразу выбрать любую ячейку, чтобы обратиться к конкретной ячейке памяти в ЗУ этого типа (с управлением по шине I²C), следует пройти все ячейки, начиная с первой.

6.2. Другие программируемые схемы

Программируемые логические матрицы (ПЛМ), различной степени сложности, имеют различные названия: PAL – Programmable Array Logic (программируемая матричная логика), GAL – Generic Array Logic (типовая матричная логика), EPLD – Electrically Programmable Logic Device (логическое устройство с электрическим программированием), pLSI – Power Large-Scale Integration (большая интегральная схема (БИС) с высоким уровнем интеграции), ispLSI – Indexed Sequential Processing LSI (БИС с индексно-последовательной обработкой информации). Чтобы упростить описание, следует запомнить, что ПЛМ состоят из логических элементов, сгруппированных в матрицы. Программирование заключается в пережигании «перемычек», устанавливающих связи между элементами. Отсоединение одного из элементов соответствует установлению логического нуля. Можно сказать, что ПЛМ – это микросхемы «жесткой логики» последнего поколения. На них можно произвести модернизацию принципиальных схем на «жесткой логике». При этом останется тот же самый проверенный алгоритм работы, но на несколько порядков сократятся вес и габариты устройств.

6.3. Микроконтроллер

Микроконтроллер (МК) является очень сложным устройством, которое самостоятельно выполняет целый комплекс операций. В одном корпусе (точнее кристалле – название по технологии изготовления) находятся микропроцессор, один или несколько видов запоминающих устройств, цифровые (иногда аналоговые) устройства ввода/вывода, одно или несколько синхронизирующих устройств, и другие элементы, выполняющие многочисленные функции. Полное описание этих элементов выходит за рамки данной книги. Имейте в

виду, что существует большое число микроконтроллеров, как и их производителей, и каждый микроконтроллер выполняет свои специфические функции. Каждая ведущая фирма, известная по торговой марке, производит также устройства для программирования и программное обеспечение, позволяющее лучше использовать все возможности производимых микроконтроллеров и установленных в них программ.

Наиболее известны микроконтроллеры PIC фирмы Microchip®, Atmel®, Motorola® и широко известное семейство BASIC STAMP фирмы Parallax®, из которого наиболее популярен у зарубежных радиолюбителей микроконтроллер в виде платы – BASIC STAMP 2. Другие фирмы также предлагают свои микроконтроллеры разной производительности, но мы не можем их все приводить здесь, хотя они могут посостязаться в производительности и других показателях с вышеперечисленными микроконтроллерами... .

Глава **7**

Выбор микроконтроллера

Перед тем как приступить к изучению цифровой электроники, следует выбрать микроконтроллер среди большого ряда предложений на современном рынке (рис. 7.1). Он будет служить основным устройством в изложении материала по микроконтроллерам и соответственно в обзоре типовых схем на микроконтроллерах на всем протяжении книги.

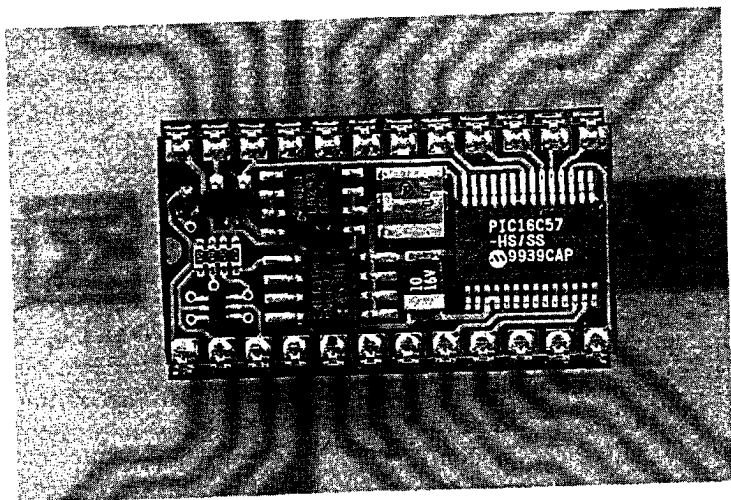


Рис. 7.1. Микроконтроллер BASIC STAMP 2 CMS фирмы Parallax®

При первом рассмотрении, руководствуясь прагматизмом большинства наших читателей, мы исключили слишком сложные и слишком дорогие устройства. В качестве языка программирования мы выбрали наиболее простой и понятный даже школьнику язык BASIC. Последним критерием, по которому выбирался микроконтроллер, явился тип корпуса: например, для квадратной микросхемы с 68 выводами (PLCC) – по 17 с каждой из четырех сторон – тяжело провести разводку печатной платы, а также распаять планарные выводы с шагом не 2,5 мм или 2,54 мм, если применяются дюймовые величины, а в 2 раза меньше.

В результате изысканий выбор пал на микроконтроллер BASIC STAMP фирмы Parallax® (сокращенно его называют BS2). Он выпускается в продажу в виде небольшой печатной платы размером с микросхему с 24 выводами в корпусе DIP. Элементы этого микроконтроллера выполнены в виде CMS, и практически все легко распознаются. Он собран на основе микроконтроллера PIC16C57, перепрограммируемой постоянной памяти с электрическим стиранием EEPROM типа 24LC16 с доступом по шине I²C и несколькими элементами, которые обеспечивают его инициализацию и передачу данных при программировании. Двухсторонние DIP-выводы позволяют вставлять его в стандартный держатель микросхем, так называемую кровать, на 24 вывода. Так как его конструкция видна невооруженным взглядом, этот микроконтроллер очень полезен с точки зрения эффективного обучения. В дальнейшем мы изучим схемотехническое решение этого микроконтроллера на компонентах DIP – с двухрядным расположением выводов.

Программы для этого микроконтроллера составляются на учебном языке PBASIC2 (это одна из разновидностей BASIC, разработанная специально для BS2).

В этом языке основные функции программирования применяются совместно с функциями, предназначенными для работы с внешними устройствами ввода и вывода. Этот язык применяется для использования устаревших типов памяти PIC предыдущих модификаций, что объясняет его высокую цену. Кроме того, в микроконтроллерах BS2 не используется специализированный компилятор, который также дорого стоит.

В отличие от других языков, в которых функции описываются очень громоздко, язык PBASIC2 позволяет реализовать различные функции одной командой (составить номер телефона, вывести одновременно два сигнала с разной частотой на один выход, сгенерировать или сосчитать калиброванный импульс, предоставить двухсторонний доступ к памяти, обмениваться данными по последовательному порту...).

Глава 8

Характеристики микроконтроллера BASIC STAMP 2

В этой главе мы рассмотрим основные характеристики BASIC STAMP 2. Этот микроконтроллер, кроме своеобразного внешнего вида в виде платы, обладает следующими характеристиками:

- 16 независимых двухсторонних входных и выходных каналов, которые могут обеспечивать максимальный ток 20 мА (для 8 каналов максимальный ток равен 50 мА);
- 2 канала, которые предназначены для связи с РС для выполнения программирования;
- 2 канала доступа к памяти EEPROM;
- 1 канал начальной установки – инициализации (Сброс);
- высокая скорость обмена: 20 МГц;
- напряжение питания 5 или 9–12 В в зависимости от вида микроконтроллера;
- усовершенствованный язык программирования PBASIC2;
- память EEPROM 2 Кб (2048 байт) с шиной I²C с возможностью одновременного хранения программы, переменных и данных;
- возможность считывания и записи в память, используя программу;
- бесплатное программное обеспечение и инструкция по программированию.

Глава 9

Программирование микроконтроллера BASIC STAMP 2

9.1. Программное обеспечение

Память BS2, как и всего микроконтроллера, должна содержать программу, написанную для управления устройством. Фактически нужно установить связь между компьютером PC и BS2, чтобы ввести исходный код, а также исключить возможные ошибки. Компания Parallax Inc предлагает проверенную программу STAMPW.EXE, которая предназначена для работы с микроконтроллерами. Вы найдете эту программу и можете бесплатно скачать ее с сайтов <http://www.parallaxinc.com> и <http://www.dunod.com>. Тем читателям, у которых нет доступа в Интернет, эта программа предлагается на диске CD-ROM, который поставляется в номере 243 журнала «Практическая электроника» (Magazine «Practical Electronics»). Найти его можно в поисковых системах Интернет, например Google и других.

Программа STAMPW включает в себя мощный блок редактирования программ. Также эта программа служит для программирования BS2, индикации использования виртуальной памяти, синтаксической проверки языка PBASIC2 и запуска основной программы. Во время выполнения программы можно заставить STAMPW выводить значения переменных на экран PC с помощью команды **DEBUG**. Все команды можно запускать с помощью иконок или через меню. На рис. 9.1 показано окно программы на экране компьютера.

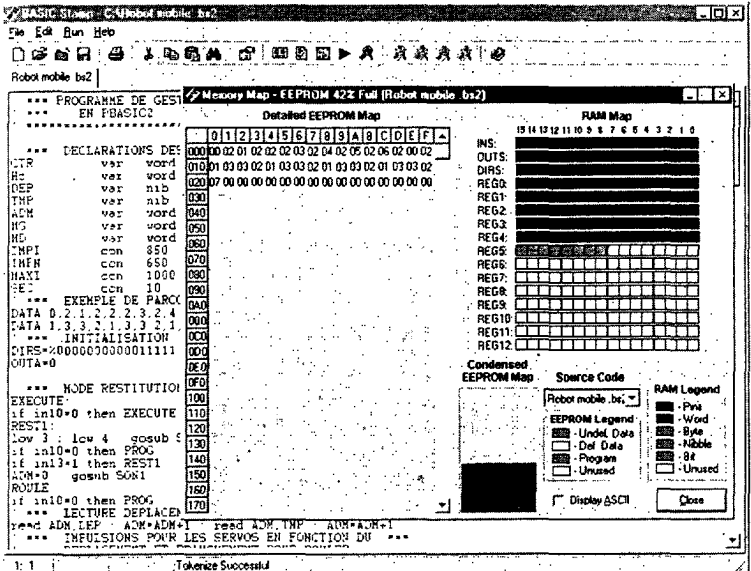


Рис. 9.1. Окно программы STAMPW.EXE в начальной стадии инсталляции

При первом использовании программы STAMPW необходимо настроить программу для работы с используемым микроконтроллером BS2 и последовательным портом компьютера. Выбор последовательного порта может осуществляться автоматически. Эти настройки можно найти на вкладке **Редактор операций** (Editor operation) подменю **Задание параметров** (Preferences) меню **Правка** (Edit). Эта вкладка показана на рис. 9.2.

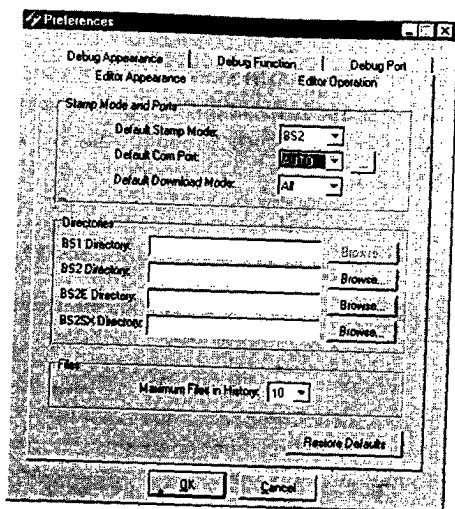


Рис. 9.2. Окно программы STAMPW.EXE в конфигурации BS2 и работы с последовательным портом

9.2. Язык программирования PBASIC2

Язык программирования PBASIC2 очень разнообразен. Благодаря большому количеству встроенных функций этот язык легко сочетается с микроконтроллером BS2*. К сожалению, спецификация языка программирования PBASIC2 занимает по объему книгу, такую же как эта. Для досконального изучения языка потребуется очень много времени, поэтому мы опишем в данной книге кратко основные команды языка. Читатели, желающие более подробно изучить этот язык программирования, могут изучить файл ST2_MAN.PDF, доступный на сайтах Интернета, ссылки на которые были даны выше. В этом файле на английском языке описывается полная документация BASIC STAMP2.

* PBASIC2 был изначально разработан именно для микроконтроллера BS2. – *Прим. науч. ред.*

Вводные замечания

Перед тем как начать описание всех команд, следует уточнить несколько обязательных правил этого языка программирования. Для данного языка программирования прописные и строчные буквы не разделяются между собой. Кроме того, в одну строчку можно записывать несколько операторов, разделенных знаком «:».

Константы

Константы должны вводиться в самом начале программы. Они могут представляться в четырех формах. Перед константами должна присутствовать команда `con`.

```
Number1 con 157           'Число десятичное,
                          'если перед ним нет
                          'никаких символов
Number2 con %10011101    'Число двоичное, если
                          'перед ним символ %
Number3 con $9D          'Число 16-ричное, если
                          'перед ним символ $
Number4 con "Значение:"  'Последовательность
                          'символов, заключенная
                          'в кавычки
```

Переменные

Введение переменных осуществляется по тем же правилам, что и констант. Переменные хранятся в ОЗУ, после них пишется их размер. Перед введением переменной следует написать команду `var`.

```
Number1 var bit          'переменная типа бит
                          '(1 бит)
Number2 var nib          'переменная типа полу-
                          'байт (4 бита)
Number3 var byte         'переменная типа байт
                          '(8 бит)
Number4 var word         'переменная типа сло-
                          'во (16 бит)
```

Таблицы определяются по следующей форме.

```
Counter var byte(4)     'таблица размером
                          '4 байта
```

Кстати, также можно определить переменные `alias`, определяющие часть переменной.

Метки

Метки помогают определить операции перехода в программе. Они подчиняются строгим правилам:

- состоят из букв и цифр;
- не должны иметь пробелы;
- меткам запрещается присваивать имена команд;
- метки должны начинаться с новой строки;
- метки должны заканчиваться двоеточием.

Во время последующих запросов двоеточие следует убрать.

Комментарии

Комментарии можно вводить на всю строку или писать в конце строки. Перед комментариями пишется апостроф (').

При написании программы или в готовой программе комментарии очень полезны: они позволяют в любой момент разобраться в переменных и умозаключениях, как самому программисту, написавшему программу, так и любому другому, пожелавшему разобраться с текстом программы. И самый главный аргумент в их пользу: они присутствуют только в программе редактирования и не занимают объема памяти.

Логические и арифметические операторы

В PBASIC2 математические операторы выполняются по своему написанию, то есть слева направо. В других языках программирования операции придерживаются первоочередности вычислений. Тем не менее возможно изменить порядок счета с помощью скобок.



Результаты вычислений выводятся без десятичных долей, только целые числа.

Пример:

$12 + 3 \times 2 / 4$ это значит:

$12 + 3 = 15 \Rightarrow 15 \times 2 = 30 \Rightarrow 30 / 4 = 7$

А не 7,5, как должно было быть по правилам счета!

Ниже представлен список операторов:

Логические и условные

Эти операторы определяют порядок приоритетов (первоочередности).

NOT	И-НЕ (приоритет 1)
AND	И (приоритет 2)
OR	ИЛИ (приоритет 2)
XOR	ИСКЛЮЧАЮЩЕЕ ИЛИ (приоритет 2)
=	Равно (приоритет 3)
<>	Не равно (приоритет 3)
<	Меньше (приоритет 3)
>	Больше (приоритет 3)
<=	Меньше или равно (приоритет 3)
>=	Больше или равно (приоритет 3)

Операторы аргумента

ABS	Модуль аргумента
SQR	Корень квадратный аргумента
DCD	Преобразует десятичное число в двоичное
NCD	Преобразует двоичное число в десятичное
SIN	Дополнение до 2 к 8 битам синуса угла, выраженного в бинарных радианах (brad)
COS	Дополнение до 2 к 8 битам косинуса угла, выраженного в бинарных радианах
-	Дополнение до 2
-	Дополнение до 1

Двойные аргументы

+	Сложение
-	Вычитание
/	Деление целых чисел (возвращает частное)
//	Деление целых чисел (возвращает остаток)
*	Умножение до 16 бит
**	Умножение, возвращает избыток свыше 16 бит (сильное кодовое слово)
*/	Умножение, показывает слабое кодовое слово
MIN	Ограничение величины снизу
MAX	Ограничение величины сверху

DIG	Вывод цифры по запрашиваемой позиции
<<	Сдвиг влево
>>	Сдвиг вправо
REV	Инверсия определенных битов числа, начиная справа
&	Логическое И двух чисел
	Логическое ИЛИ двух чисел
^	ИСКЛЮЧАЮЩЕЕ-ИЛИ двух чисел

Команды PBASIC2

Классификация по функциям позволяет разъяснить использование каждой команды. Конечно, мы не смогли в полной мере прокомментировать каждое описание, но чаще всего в описание включен пример программирования, который помогает понять использование команды. Несколько более сложных команд связи с более насыщенным синтаксисом только упоминаются и сопровождаются всего лишь одной строкой пояснений. Эти команды не используются в данной книге. Читатели, желающие углубить свои познания, могут обратиться к инструкции по использованию микроконтроллера BS2 или к книгам, указанным в библиографии.

Переходы

BRANCH

Эта команда очень похожа на команду CASE в других языках программирования. Она оценивает выражение переменной *index* и продолжает выполнение программы с соответствующего адреса списка.

BRANCH *index*, [адрес 0, адрес 1,...адрес N]

index : оцениваемая переменная.

адрес0, 1, 2,... : метка, указывающая адрес перехода программы.

Пример: BRANCH xxx, [adr0, adr1, adr2]

Означает: Если xxx равно 0, то выполнение программы продолжается с метки с адресом adr0.

Если xxx равно 1, то выполнение программы продолжается с метки с адресом adr1.

Если xxx равно 2, то выполнение программы продолжается с метки с адресом adr2.

GOSUB

Команда переводит к выполнению подпрограммы, расположенной в строке с меткой etiq1ab. Программа может содержать 255 подпрограмм и 4 уровня вложений. Возврат к основной программе осуществляется при помощи команды RETURN.

GOSUB etiq1ab

- etiq1ab : Метка, где находится подпрограмма.

GOTO

Команда переводит к выполнению подпрограммы, расположенной в строке с меткой etiq1ab. Команды, осуществляющей возврат, не предусмотрено.

GOTO etiq1ab

- etiq1ab : Метка, где должно продолжиться выполнение программы.

IF...THEN

Оценивает условие с помощью оператора сравнения. При выполнении условия программа переходит к выполнению строки с адресом etiq1ab. Операторы сравнения могут быть следующими: =; <>; <; >; <=; >=.

IF condition THEN etiq1ab

condition : Выражение рода, например, выражение X=7 может оцениваться как истинное или как ложное.

etiq1ab : Адрес, на который переходит программа, если условие является истинным.

RETURN

Команда возврата из подпрограммы.

RETURN

Циклы

FOR ... {STEP}... NEXT

Создает цикл, в котором строки, находящиеся между командами FOR и NEXT, выполняются повторно. Вычисления происходят столько раз, сколько значений принимает переменная цикла, от начального значения *debut* до конечного *end*. Шаг переменной цикла может быть как положительным, так и отрицательным.

FOR переменная = начало TO конец (STEP шаг: ... NEXT

переменная : Переменная, используемая в качестве счетчика.

начало : Начальное значение переменной.

конец : Конечное значение переменной.

шаг : произвольный шаг приращения (положительный или отрицательный).

```
Пример:   FOR I=0 TO 10
           DEBUG "Значение I равно ",I
           NEXT
```

Цифровые команды

LOOKDOWN

Эта команда сравнивает значение переменной с величинами из списка. При выполнении условия, заданного одним из операторов сравнения, переменной присваивается номер первой величины из списка, удовлетворяющей условию. Если ни одно число из списка не удовлетворяет условию, переменная не изменяется. Операторы сравнения те же самые, что и для команды IF...THEN. Если оператор не указан, то по умолчанию применяется оператор «=». Нумерация индексов начинается с 0.

LOOKDOWN значение, {Условие,}[val0, val1, ...valN],
результат

значение : Значение, сравниваемое в списке.

Условие : Оператор сравнения " = , <> , > , < , >= , <= "

val0, val1 ... valN : Составляют список значений.

результат : Переменная, в которую загружается номер индекса в случае истинности сравнения.

```

Пример: значение var byte
        Результат var nib
        значение = 17
        результат = 15
LOOKDOWN значение , [26,177,13,1,0,17,99], ре-
        зультат
DEBUG "Номер индекса :", десятичный результат,
        " списка"

```

LOOKUP

Данная команда по номеру находит величину из списка и присваивает ее значение переменной resultat. Если номер больше, чем номер последней величины в списке, значение переменной не изменяется. Список может содержать 255 чисел. Нумерация индексов начинается с 0.

```

LOOKUP индекс, [val0, val1, ..., valN], результат
- индекс : Ранг загружаемого значения списка.
- val0, val1, ... valN : Составляют список значений.
- результат : Переменная, в которую загружается значение списка, соответствующее индексу.

```

```

Пример: индекс var byte
        Результат var nib
        индекс = 3
        результат = 225
LOOKUP индекс , [26,177,13,1,0,17,99], результат
DEBUG "Индекс :", десятичный индекс, " , десятичный
        результат.

```

RANDOM

Генерирует случайное число.

```
RANDOM результат
```

- результат : Переменная типа "byte" или "word", в которой находится результат.

Управление цифровыми входами/выходами

BUTTON

Данная команда выполняет функцию защиты от нежелательного дребезга контактов, возникающего при переключении контактов кнопки. Она выполняет также функцию автоповтора и перехода к строке программы после нажатия кнопки.

BUTTON вывод, состояние, задержка, скорость, переменная, мишень, адрес.

вывод : от 0 до 15. Входной вывод, к которому подключается кнопка.

состояние : 0 или 1. Логическое состояние при нажатии кнопки.

задержка : от 0 до 255. Задержка перед автоповтором.

Если = 0, то функция автоповтора отсутствует.

Если = 255, то присутствует функция антидребезга, но отсутствует функция автоповтора.

скорость : от 0 до 255. Количество циклов " BUTTON " между двумя повторами.

переменная : Рабочее пространство. Должно инициализироваться значением 0 в начале

мишень : 0 или 1. Логическое состояние кнопки (0= ненажатое состояние ; 1= нажатое состояние) для перехода к адресу.

Адрес : Метка, на которую переключается программа при нажатии.

```
Пример:   espt var   byte
          espt = 0
          цикл:
            BUTTON 7,0,255,250,espt,0,so-so
            DEBUG " * "
            so-so:
            GOTO цикл
```

COUNT

Считает число фронтов импульсов (нарастающий – от 0 к 1 или спадающий – от 1 к 0), поступающих на вход за определенный интервал времени.

COUNT вывод, период, результат

вывод : от 0 до 15. Входной вывод, на котором будет выполняться счет.

период : от 1 до 65535. Длительность счета в мс.

результат : Переменная, в которой будет сохранен результат.

```
Пример:   COUNT 4, 1000, счет
```

HIGH

Переводит определенный выход в состояние логической 1. Если команду применить к выводу, который работал как вход,

он автоматически становится выходом. Устанавливает в 1 соответствующие биты регистров DIRS и OUTS.

HIGH вывод

вывод : от 0 до 15. Выходу принудительно присваивается значение 1

```
Пример:   DEBUG "Перед : ", cr: DEBUG bin16? dirs, bin16?
           outs, cr, cr
           PAUSE 1000
           HIGH 4
           DEBUG "После : ", cr: DEBUG bin16? dirs, bin16?
           outs
```

INPUT

Конфигурирует определенный вывод на вход. Устанавливает в 0 соответствующие биты регистра DIRS.

INPUT вывод

вывод : от 0 до 15. Вывод, конфигурируемый на вход.

```
Пример:   INPUT 5
```

LOW

Переводит определенный выход в низкое логическое состояние (0). Если команду применить к выводу, который работал как вход, он автоматически становится выходом путем установки в 1 соответствующего бита регистра DIRS. Устанавливает в 0 соответствующие биты регистра OUTS.

LOW вывод

вывод : от 0 до 15. вывод, устанавливаемый на низкий уровень.

```
Пример:   DIRS = %10000
           DEBUG "Перед: ", cr
           DEBUG bin16 ? DIRS, bin16 ? OUTS, cr, cr
           PAUSE 1000
           LOW 4
           DEBUG "После: ", cr
           DEBUG bin16 ? DIRS, bin16 ? OUTS
```

OUTPUT

Конфигурирует определенный вывод на выход. Устанавливает в высокое логическое состояние (1) соответствующие биты регистра DIRS.

OUTPUT вывод

вывод : от 0 до 15. Вывод, конфигурируемый на выход.

Пример: OUTPUT 5

PULSIN

Измеряет длительность импульса на входе с точностью до 2 мкс.

PULSIN вывод, состояние, результат

вывод : от 0 до 15. Вывод, сконфигурированный на вход, на котором выполняется измерение длительности импульса.

состояние : 0 или 1. Указывает, начинается ли измеряемый импульс с 0 до 1 (состояние = 1), или с 1 до 0 (состояние = 0).

результат : Переменная, в которую будет загружен результат измерения в единицах, соответствующих 2 мкс. Максимальная длительность задержки ожидания составляет 0,131 с. После этой

задержки : результат = 0. Следует использовать переменную соответствующего типа, в противном случае результат будет усеченным.

```
Пример:  tps  var word
          else:
          PULSIN 7, 1, tps      ' Измерение 1 - положи-
                                ' тельный импульс
          If tps = 0 then еще   ' Если 0: новое
                                ' измерение
          DEBUG cis,dec ? tps   ' В противном
                                ' случае: отображение
                                ' результата
          GOTO else             ' новое измерение
```

PULSOUT

Генерирует импульс длительностью от 2 до 131 мкс, определяемый с точностью до 2 мкс.

PULSOUT вывод, длительность

вывод : от 0 до 15. Вывод подачи импульса.

длительность : от 0 до 65535. Выражает длительность

XOUT

Команда, позволяющая передавать данные по интерфейсу X-10 (дистанционное управление девиацией (изменением) несущей частоты). Устройства, поддерживающие данный интерфейс, по нашим сведениям, пока не продаются во Франции, но широко применяются в мире.

Последовательная передача данных**SERIN**

Последовательный асинхронный прием данных. Эта команда, как и три последующих, требует описания большого числа параметров, объяснение назначения которых выходит за рамки книги. Читатели, желающие узнать больше о данных командах, могут прочитать инструкцию BS2 или одну из трех книг, приведенных в библиографии.

SEROUT

Последовательная асинхронная передача данных.

SHIFTIN

Последовательный синхронный прием данных:

SHIFTOUT

Последовательная синхронная передача данных.

Управление аналоговыми входами/выходами**RCTIME**

Определяет время, в течение которого входной вывод поддерживает свое состояние. Эта команда, главным образом, используется для определения времени заряда-разряда конденсатора RC-цепи.

RCTIME вывод, состояние, результат

вывод : от 0 до 15. Вывод, на котором выполняется измерение.

состояние : 0 или 1. Состояние вывода во время измерения. (1 = С, подключенному к напряжению +5 В; 0 = С, подключенному к «земле»)

результат : от 0 до 65535. Результат измерения, выраженные в единицах, кратных 2 мкс.

Если при подключении конденсатора была нарушена его полярность, то «результат = 1», так как понадобится один цикл для нахождения запрошенного состояния.

Если время заряда превышает максимальную длительность (131 мс), то «результат = 0», так как запрошенное состояние не будет получено. Формула приблизительного расчета является следующей:

Результат = 602 * R (в КОМ) * С (в МКФ)

```
Пример:  result      var word ' Переменная для ре-
          HIGH 7          ' зультата.
          PAUSE 1         ' Разряд конденсатора ...
          RCTIME 7,1, result ' ... в течение 1 мс.
          DEBUG ? result  ' Измерение времени за-
                          ' рядки.
                          ' Отображение резуль-
                          ' тата.
```

PWM

Переводит цифровые данные в аналоговое напряжение посредством широтно-импульсной модуляции (ШИМ) цепи заряда-разряда RC-цепи. Следует регулярно обновлять выполнение команды, чтобы поддержать заряд конденсатора. После выполнения команды вывод работает в качестве входа.

PWM или ШИМ вывод, отношение, циклы

вывод : от 0 до 15. Выводы, на которые будет подаваться модулируемое напряжение.

отношение : от 0 до 255. Переменная, указывающая уровень выходного напряжения (между 0 и 5 В).

Формула определения напряжения: $U = (\text{значение} / 255) * 5 \text{ В}$

циклы : от 0 до 65535. Указывает приблизительную длительность напряжения шагами длительностью 5 мс.

```
Пример:  else:
          PWM 8,100,40    ' 40 циклов (100/255)*5 в В
          PAUSE 1000     ' Задержка длительностью в 1 с
          GOTO else
```


Звуки

DTMFOUT

Генерирует 2-тональные телефонные послышки частоты стандарта DTMF*.

DTMFOUT вывод, {тр громкий, тр тихий,}{клавиша, клавиша, ...}

- вывод от 0 до 15. Выходной вывод, на который подаются послышки частоты.

тр высокий : от 0 до 65535. Длительность тональности в мс. 200 мс по умолчанию.

тр тихий : от 0 до 65535. Длительность промежутка тишины между двумя тональностями в мс. 50 мс по умолчанию.

клавиша : от 0 до 15. Соответствие телефонным клавишам. от 0 до 9 = Цифры.

10 = Звездочка * ; 11 - Диез # ; от 12 до 15 = колонки от A до D.

Пример: DTMFOUT 3,[6,2,4,8,3,3,3,]
DTMFOUT 7,500,100,[6,2,4,8,3,3,3]

FREQOUT

Подает гармонический сигнал или два гармонических сигнала одновременно определенной длительности на один вывод. Для этой команды требуется RC-фильтр, чтобы отфильтровать синусоиду.

FREQOUT вывод, длительность, частот1 (,частот2)

вывод : от 0 до 15. Вывод, на который подаются частоты.

длительность : от 1 до 65535. Длительность частоты в мс.

частот 1 : от 0 до 32767. Первая частота в герцах.

частот 2 : от 0 до 32767. вторая факультативная частота в герцах.

Пример: FREQOUT 2,1000,2500
FREQOUT 9,1000,1000,3000

* DTMF - сокращение от Dual-Tone Multi-Frequency - двухтональный многочастотный набор телефонных посылок частот звукового диапазона (от 300 до 3000 Гц). - Примеч. науч. ред.

Доступ к памяти EEPROM

DATA

Помещает данные в нижнюю область памяти EEPROM во время запуска программы. Так как данные, отвечающие за запуск программы, находятся по адресам в верхней области памяти, то следует опасаться совмещения данных. В случае возникновения проблемы программа STAMPW сигнализирует об ошибке.

Предусматривается несколько различных синтаксисов.

```
Mark DATA meaning1, meaning2, ...
```

Описывает константы meaning, meaning2, ... , начиная с помеченного адреса.

```
Mark DATA adr, meaning1, meaning2, ...
```

Описывает константы meaning1, meaning2, ... , начиная с адреса adr.

```
Mark DATA nb
```

Сохраняет число байтов, равное nb, начиная с запланированного адреса. При этом содержимое ячеек памяти не учитывается.

```
Mark DATA meaning, nb
```

Сохраняет число байтов, равное nb, после отмеченного адреса. Содержимое каждой ячейки памяти принимает значение, равное valeur.

```
Mark DATA word meaning
```

Разбивает 16-битное слово на 2 байта. Помещает старший байт в указанную ячейку, а младший байт – в следующую за ней ячейку.

READ

Считывает содержимое ячейки памяти EEPROM и помещает его в переменную.

```
READ адрес, переменная
```

адрес : от 0 до 2047. Указывает адрес, содержание которого необходимо прочесть.

переменная : Переменная, в которой будет храниться содержание адреса.

```
Пример: Message data "ES2 хранит данные в EEPROM ! ",0
        Address var word
        car var byte
```

```
Address = Message ' Первый адрес
                ' сообщения
```

```

readsymbol:
READ Address, car           ' Считывает символ из
                             ' памяти EEPROM.
If car <> 0 THEN next       ' Конец или
                             ' продолжение.
STOP                         ' Остановка программы.
next:
DEBUG car                   ' Отображает символ на
                             ' экране.
Address = Address + 1       ' Указывает следующий
                             ' адрес.
GOTO readsymbol             ' Чтение следующего
                             ' символа.

```

WRITE

Записывает байт в память EEPROM во время выполнения программы, в противоположность команде DATA. Следует отметить, что постоянная память EEPROM имеет ограниченное количество циклов записи, примерно 10 млн. При интенсивности записи один раз в секунду, получается в день 86 400 циклов, то есть примерно 116 дней работы до превышения предела.

```
WRITE address,data
```

адрес : от 0 до 2047. Указывает адрес EEPROM, по которому будет записано данные.

данные : Данные типа «байт», записываемые в память EEPROM

Пример: WRITE 100,127

Временные команды

PAUSE

Останавливает выполнение команды на заданное время, выраженное в мс.

```
PAUSE time
```

time : Длительность задержки, выражаемая в мс.

```

Пример: flash:
        LOW 7
        PAUSE 100
        HIGH 7
        PAUSE 100
        GOTO flash

```

Управление энергопотреблением

END

Останавливает выполнение программы и переводит BS2 в спящий режим пониженного энергопотребления.

NAP

Программа переводит микроконтроллер в спящий режим пониженного энергопотребления на короткий период времени. Энергопотребление уменьшается примерно до 100 мкА. Никакие команды не выполняются.

NAP time

- длительность : от 0 до 7. Период неактивности в мс, соответствующий формуле $18 \times 2^{\text{длительность}}$

Если вызвана какая-то команда, то микроконтроллер выходит из спящего режима, потребление тока увеличивается и все выходы в течение 18 мс работают как входы. Данное состояние следует устанавливать в определенных ситуациях.

SLEEP

Переводит BS2 в дежурный режим на определенное время.

SLEEP длительность

длительность : от 0 до 65535. Теоретически каждая единица времени имеет значение 1 с; но число округляется до ближайших 2,304 с. Внимание на биение напряжения длительностью 18 мс каждые 2,304 с. SLEEP 1 продлится 2,304 с, тогда как SLEEP 10 продлится 11,52 с.

Пример:

```

low 3      ' На светодиод подается напряжение
pause 1000 ' Задержка длительностью 1 с
else:
high 3     ' Выключение светодиода
pause 1000 ' Задержка длительностью 1 секунда
low 3     ' На светодиод подается напряжение
SLEEP 10  ' Состояние дежурного режима в течение
           ' 11,52 с.
goto else

```

Контроль и управление программой

DEBUG

Выводит данные на экран компьютера. Эта команда может принимать разный вид в зависимости от выводимых данных и от выбранного формата. Мы будем использовать эту команду в некоторых примерах, но для получения большей информации о ней, воспользуйтесь инструкцией к STAMPW.EXE или книгами, указанными в библиографии.

STOP

Останавливает выполнение программы, не переводя ее в дежурный режим (с низким энергопотреблением).

Глава 10

Программатор BASIC STAMP 2 на основе элементов в корпусе DIP

10.1. Представление

BASIC STAMP 2 (BS2) – микроконтроллер, с помощью которого легко изготавливать схемы по причинам, изложенным выше (рис. 10.1). Одним из его немногочисленных недостатков является немного завышенная цена. В оправдание отметим, что язык высокого уровня программатора является встроенным и программное обеспечение распространяется абсолютно бесплатно! Эти доводы показывают нам, что любой может при наличии желания и средств собрать программатор BS2 на основе элементов в корпусе DIP. Эта схема будет иметь несколько большие размеры (она размещается на плате размером 72×51 мм). Она будет иметь те же характеристики, что и схема CMS, за исключением того, что плата с радиоэлементами не упакована в корпус, и поэтому имеется возможность при неправильной эксплуатации закоротить источник питания. Данное бескорпусное конструктивное решение позволяет сэкономить примерно 50–60% от стоимости модуля BS2 в конструктиве CMS. Кроме того, организуется доступ ко всем элементам, включая память 26LC16, которая становится сменной, так как устанавливается в микроразъем «кроватку». Другим большим плюсом этой реализации является обучающий эффект, когда все на виду, что позволяет изучить внутреннюю структуру схемы BASIC STAMP 2.

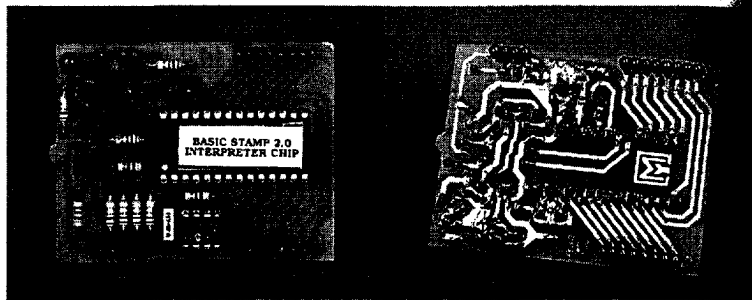


Рис. 10.1. Программатор BASIC STAMP 2 на основе элементов в корпусе DIP: вид с двух сторон

Читатели, не желающие воспользоваться представленным здесь описанием BASIC STAMP 2, могут сразу перейти к следующей главе и использовать во всех предлагаемых в данной книге устройствах модуль BS2 на основе элементов CMS.

Проинформируем читателя, что эксклюзивным дистрибутором продуктов Parallax Inc® во Франции является фирма Selectronic. В ней вы найдете любые комплектующие BS2*.

10.2. Принципиальная схема

Если сравнить принципиальные схемы BS2, одну – показанную в инструкции Parallax Inc®, и другую схему – на рис. 10.2, можно заметить, что в них соблюдено все вплоть до расположения выводов для более удобного согласования их между собой.

Центральным местом в схеме является, безусловно, незаменимый интерпретатор фирмы Parallax Inc®. Он работает на частоте 20 МГц, стабилизированной по частоте с большой точностью пьезорезонатором X1.

Этот микроконтроллер семейства PIC (PIC 16C57) продается с установленной программой на языке PBASIC2, записанной в постоянную память ROM (мы не будем рассматривать этот вид памяти). Другим основным элементом данной схемы является память EEPROM на микросхеме CI2 – 24LC16

* Любые микросхемы и другие радиоэлементы можно найти через Интернет или через известные фирмы по продаже радиоэлементов – «Чип и Дип», «Золотой шар» и др. – Прим. науч. ред.

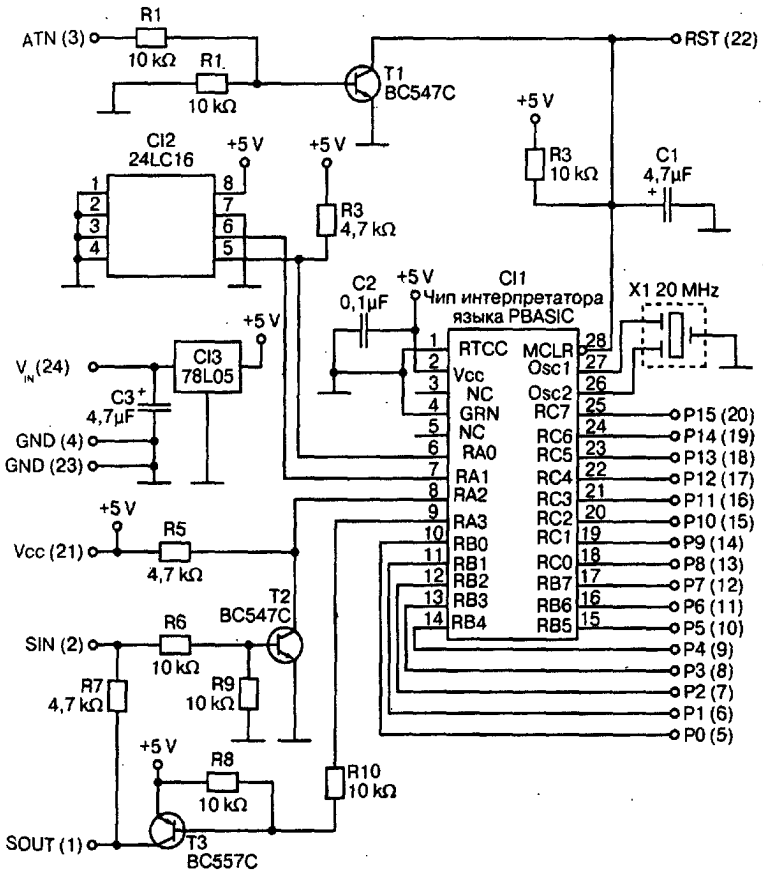


Рис. 10.2. Принципиальная схема BS2 на основе элементов в корпусе DIP

с 8 выводами с последовательным доступом по интерфейсу I²C. В функцию памяти EEPROM, в отличие от оперативной памяти RAM, которая расположена в интерпретаторе, входит сохранение программы и энергонезависимых данных. Связь между двумя этими элементами осуществляется по двум каналам с использованием протокола I²C: канал данных (RA0) и канал синхронизации (RA1). В состоянии покоя сигнал данных находится в состоянии высокого логического уровня благодаря подтягивающему резистору R4, подключенному к источнику напряжения 5 В. Связь с компьютером осуществляется

по другим двум каналам: RA2 – каналу приема данных, RA3 – каналу передачи данных. Остальные элементы схемы предназначены для вспомогательных целей. На базу транзистора T2 через резистор R6 подаются последовательные данные SIN компьютера, далее с коллектора транзистора T2 выходной сигнал поступает на вывод RA2 микросхемы C11. С вывода RA3 C11 через резистор R10 выходные данные от C11 поступают на базу транзистора T3 и далее с коллектора выходные данные SOUT (1) приходят на компьютер. Резисторы R5, R7, R8 и R9 определяют рабочую точку усилителей по постоянному току и совместно с транзисторами T2 и T3 выполняют соответственно преобразование двуполярных уровней RS232 – в однополярные – TTL (T2 – от ПК к плате) и обратное преобразование однополярных (TTL) в двуполярные (T3 – от платы к ПК). В начале программирования происходит инициализация C11 при подаче сигнала ATN через резистор R1 на базу транзистора T1. В исходном состоянии резистор R2 подключает базу к потенциалу «земли», и транзистор T1 закрыт. При включении питания RC-цепь из конденсатора C1 и резистора R3 формирует обнуляющий импульс на инверсный вход MCLR микросхемы C11.

16 двухсторонних каналов RB0–RB7 и RC0–RC7 предназначены для подключения модуля по линиям P0–P15.

Питание BS2 может обеспечиваться двумя способами. Вывод V_{cc} предназначен для подачи стабилизированного напряжения +5 В. На вывод V_{IN} можно подавать большее напряжение (12 В), при этом стабилизация до 5 В осуществляется с помощью линейного стабилизатора C13. Выход стабилизатора напряжения +5 В фильтруется по низкой частоте полярным конденсатором C1, а по высокой – неполярным керамическим конденсатором C2.

10.3. Реализация

Цель данной работы состоит в том, чтобы на основе обычных элементов в корпусе DIP построить модуль BASIC STAMP 2. В результате получается модуль на печатной плате с размерами 72×51 мм с относительно высокой плотностью монтажа.

Следует запаковать в плату микроразъем «кроватьку», позволяющий вставлять микросхему C11 интерпретатора PBASIC2.

Рисунок печатной платы, показанный на рис. 10.3, не представляет особых сложностей. Расположение элементов на печатной плате показано на рис. 10.4. Вы можете изготовить плату любым из известных способов. Но мы вам советуем использовать метод фотолитографии, который дает самые лучшие результаты. Мы можете найти материалы по технике фотолитографии в приложении к данной книге. В верхней части платы сохраните небольшой выступ, который служит для ориентирования платы.

Список компонентов:

Резисторы 5-процентные:

- R1, R2, R3, R6, R8, R9, R10: 10 кОм (коричневый, черный, оранжевый);
- R4, R5, R7: 4,7 кОм (желтый, фиолетовый, красный).

Конденсаторы:

- C1: 4,7 мкФ / 16 В (полярный танталовый или электролитический с радиальными выводами);
- C2: 0,1 мкФ неполярный керамический;
- C3: 10–47 мкФ / 16 В (полярный танталовый или электролитический с радиальными выводами).

Полупроводники:

- C11: PBASIC2, микроконтроллер BASIC STAMP 2 (Parallax Inc®);
- C12: 24LC16;
- C13: 78L05;
- T1, T2: BS547C, T3: BC557C.

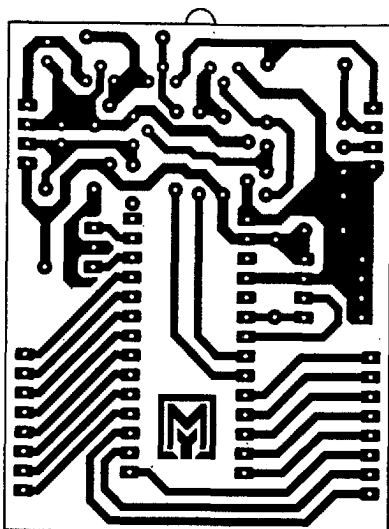


Рис. 10.3. Рисунок печатной платы

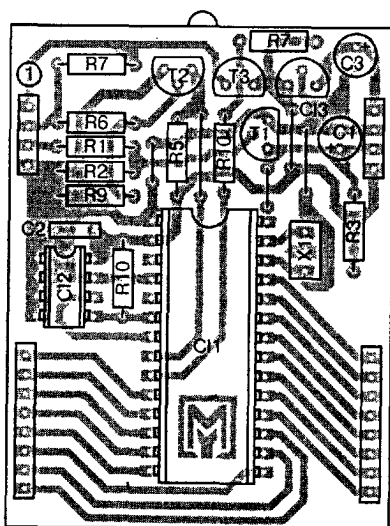


Рис. 10.4. Вид печатной платы с размещенными радиоэлементами

Прочие элементы:

- X1: пьезорезонатор на 20 МГц;
- 1 держатель-микроразъем 28-выводной микросхемы;
- 1 держатель-микроразъем 8-выводной микросхемы;
- 24-выводной разъем для внешних подключений.

Работы по монтажу элементов, которые приведены на рис. 10.4, следует начинать с запаивания пяти перемычек из монтажного провода. Надо сказать, что без них нам пришлось бы изготавливать двухстороннюю схему, которую изготовить в домашних условиях гораздо труднее. Детали следует запаивать на плату в очередности, определяемой размерами и степенью прочности элементов. Придерживайтесь следующего порядка: резисторы, держатели микросхем, резонатор, керамический конденсатор, транзисторы, стабилизатор напряжения и, наконец, полярные танталовые или электролитические конденсаторы, которые устанавливаются с соблюдением полярности. Разъемы для подключения модуля следует запаивать с обратной стороны платы. Их следует предварительно выравнять (рис. 10.5).

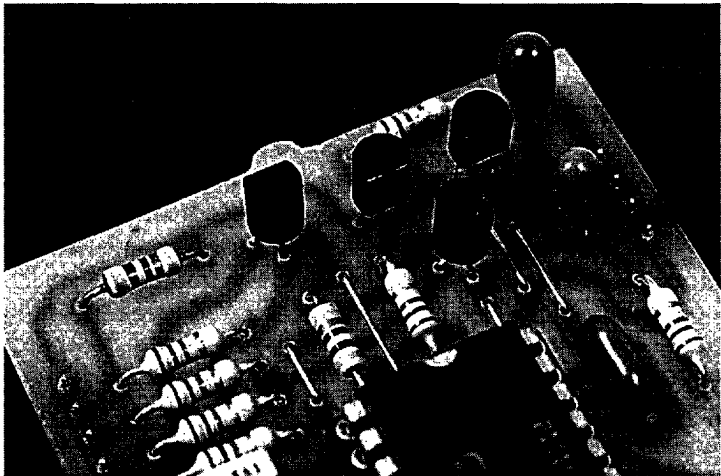


Рис. 10.5. Вид основных элементов платы

Перед тем как вставить микросхемы CI1 и CI2 в отверстия платы, внимательно проверьте плату BS2. Брак может быть разным: большая капля припоя может вызвать короткое замыкание; медная дорожка может содержать обрыв; элемент, который ориентирован неправильно (касается транзисторов T1–T3, конденсаторов C1 и C3) или имеет неправильный номинал (касается всех резисторов и конденсаторов). Любой из них приведет к серьезным повреждениям.

После точного следования всем рекомендациям данной главы у вас есть микроконтроллер BS2 собственного изготовления, такой же надежный, как и коммерческий модуль BS2 в корпусе CMS. В следующей главе мы изучим устройство программатора BASIC STAMP 2, снабженного небольшой тестовой платой.

Глава **11**

Программатор BS2 и плата тестирования

11.1. Краткое представление содержания главы 11

Что бы вы ни выбрали: коммерческий модуль BASIC STAMP 2 на основе элементов в корпусе CMS или собранный самостоятельно в предыдущей главе в корпусе DIP, вам обязательно потребуется программатор. Он служит не только для того, чтобы записать данные в память микроконтроллера, но и для установления связи между компьютером и вашим устройством. Программатор, который мы будем изучать, поддерживает обе версии BS2. У этого программатора имеется 4-выводной разъем, позволяющий напрямую работать со схемами, не прибегая к многочисленным операциям с BS2 (рис. 11.1).

Изучение нового микроконтроллера будем начинать с проведения всевозможных тестов и экспериментов. Для этого в состав программатора мы включили плату тестирования, на которой расположены: 5 кнопок, 8 индикаторных светодиодов, 8-канальный инвертирующий усилитель. Эксперименты проводятся без пайки, подключение проводится при помощи двух гибких плоских кабелей, подключаемых к плате и компьютеру – один кабель разъемами (с двух сторон) DB9, другой – 4-выводными вилками.

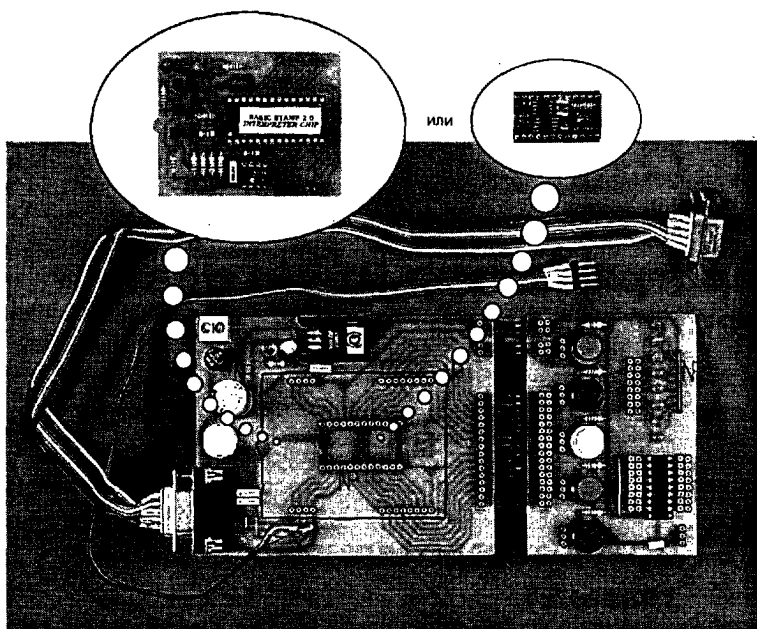


Рис. 11.1. Программатор BS2, плата тестирования и соединительные провода (варианты микроконтроллеров в корпусах DIP и CMS)

11.2. Принципиальная схема программатора BS2 и платы тестирования

Принципиальная схема на рис. 11.2 включает в себя программатор, плату тестирования и расположение выводов микроконтроллера BASIC STAMP 2 в корпусе CMS.

Программатор

Схема программатора (рис. 11.3) выглядит очень компактной, простота схемы обусловлена высокой степенью интеграции BASIC STAMP 2. Основную часть схемы занимает схема питания. Напряжение питания 9 В может вырабатываться от сети переменного тока или поступать от батарейного источника питания. В первом случае на входе диодного моста поступает переменное напряжение 9 В с понижающего трансфор-

матора 220/9 В, которое выпрямляется диодным мостом PR1, а C3 – фильтрует напряжение по высокой частоте. Наличие напряжения питания +5 В определяется по зажиганию светодиода L1. Ток через светодиод ограничивается резистором R1.

Связь с компьютером осуществляется через последовательный интерфейс RS-232, разъем DB9 служит для подсоединения к компьютеру. Замыкание ножек 6 и 7 разъема DB9 связи с компьютером очень важно, так как оно позволяет обнаружить присутствие BASIC STAMP 2 для компьютера.

Сигнал Rx интерфейса RS-232 служит для приема данных от компьютера к BS2, а сигнал Tx – для передачи данных от BS2 к компьютеру.

Перед тем как начать программирование, следует обязательно провести инициализацию BS2.

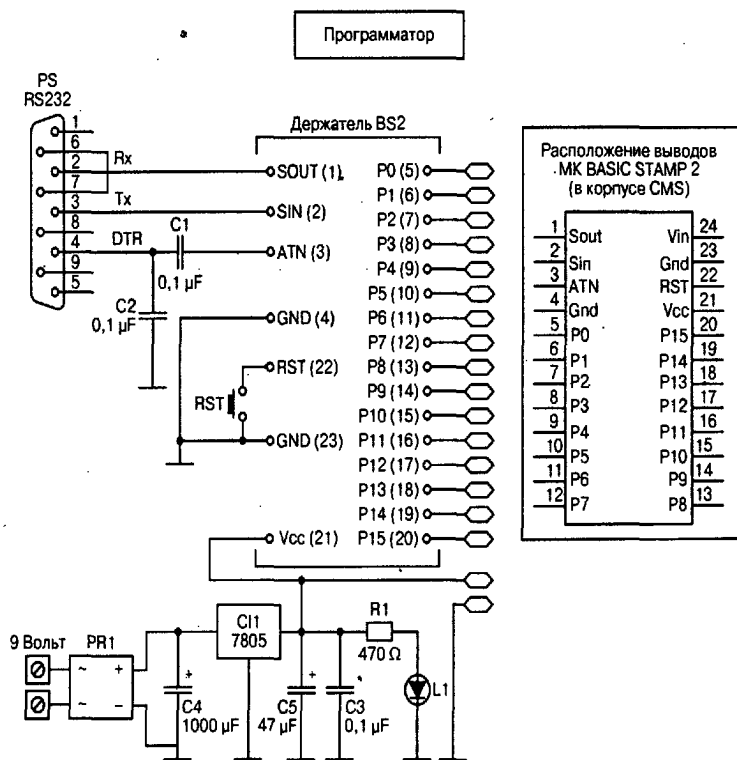


Рис. 11.2. Принципиальная схема программатора

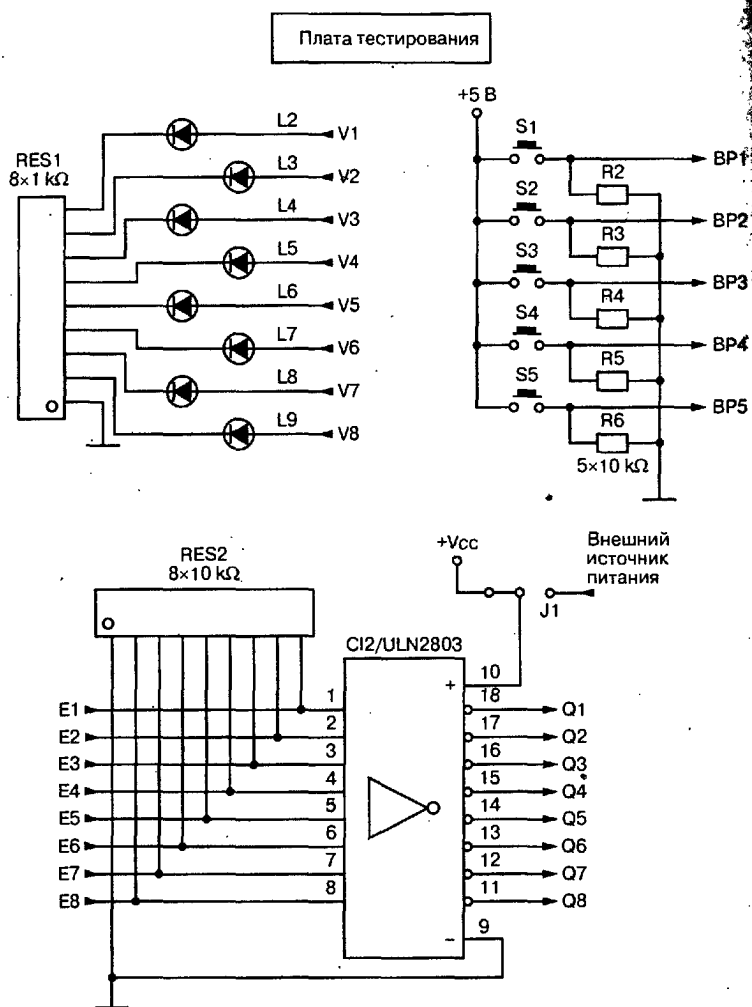


Рис. 11.2 (продолжение) Принципиальная схема программатора

Эта операция осуществляется автоматически по сигналу готовности контроллера к обмену данными DTR – Data Terminal Ready через фильтр на конденсаторах C1, C2. Кнопкой RST можно сбросить в первоначальное состояние микроконтроллер (МК) в любой момент при выполнении программы.

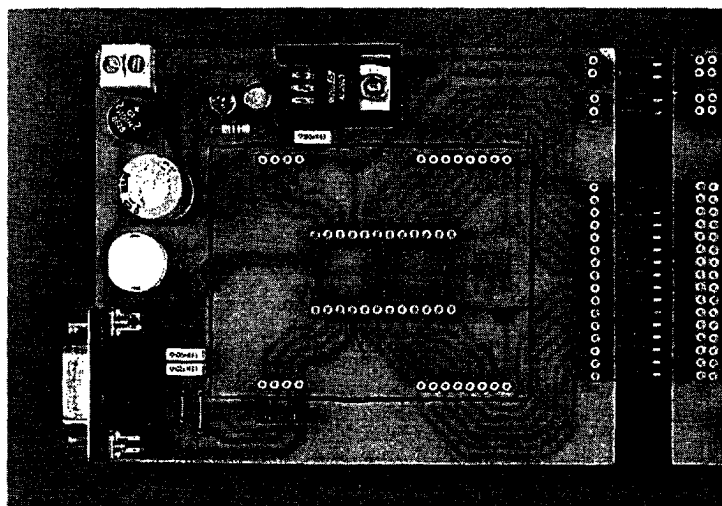


Рис. 11.3. Программатор BS2

16 двунаправленных каналов (когда сигналы поступают с входа на выход и, наоборот, с выхода на вход), так же как и напряжение питания, подведены к разъемам связи на плате.

Плата тестирования

Схема платы представляет собой несколько независимых блоков, доступ к которым также осуществляется через разъемы на плате:

- 8 индикаторов на светодиодах, ток через которые ограничен резисторами, собранными в резисторную сборку RES1. При подаче положительного напряжения на аноды светодиоды загораются;
- 5 кнопок: в состоянии покоя, резисторы R2–R6 устанавливают выводы разъемов в низкое состояние. При нажатии на кнопку соответствующий нажатой кнопки вывод переходит на состояние высокого уровня;
- 8-канальный инвертирующий усилитель.

Каждый из каналов микросхемы CI2 по выходу рассчитан на ток 0,5 А при напряжении 50 В. Таким образом она является идеальным схемным решением для управления реле,

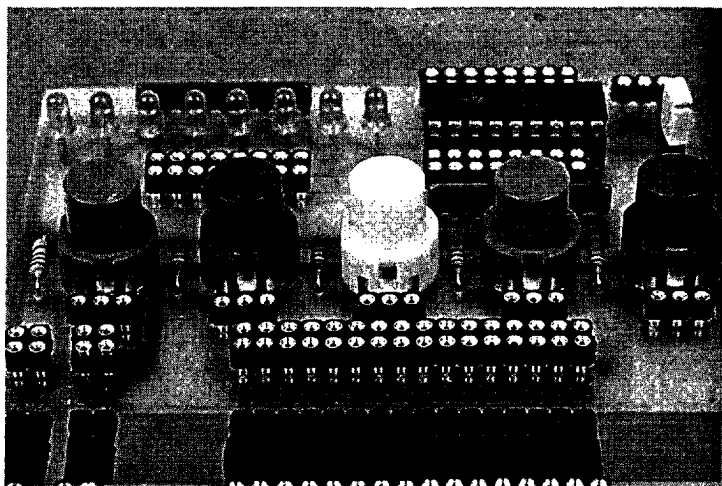


Рис. 11.4. Плата тестирования

двигателем и в основном для согласования сильноточных исполнительных устройств с BS2. Все входы усилителя устанавливаются в 0 с помощью резисторной сборки RES2. При этом на выходах усилителя будет присутствовать логическая единица. Если на какой-то вход усилителя с выхода BS2 подается положительное напряжение, то соответствующий выход обнуляется. Перемычка J1 определяет напряжение питания, подводимое к микросхеме CI2: оно может равняться +5 В и подаваться с платы или с внешнего источника напряжения, не превосходящего 50 В, для управления по выходам подключаемых исполнительных устройств.

11.3. Сборка

Схема печатной платы, показанная на рис. 11.5, включает в себя программатор и плату тестирования. Монтаж элементов проводится согласно рис. 11.6

Можно разделить эту схему на две части, чтобы различить две выполняемые функции. Тогда впоследствии нужно будет объединить две платы при помощи соединительных разъемов.

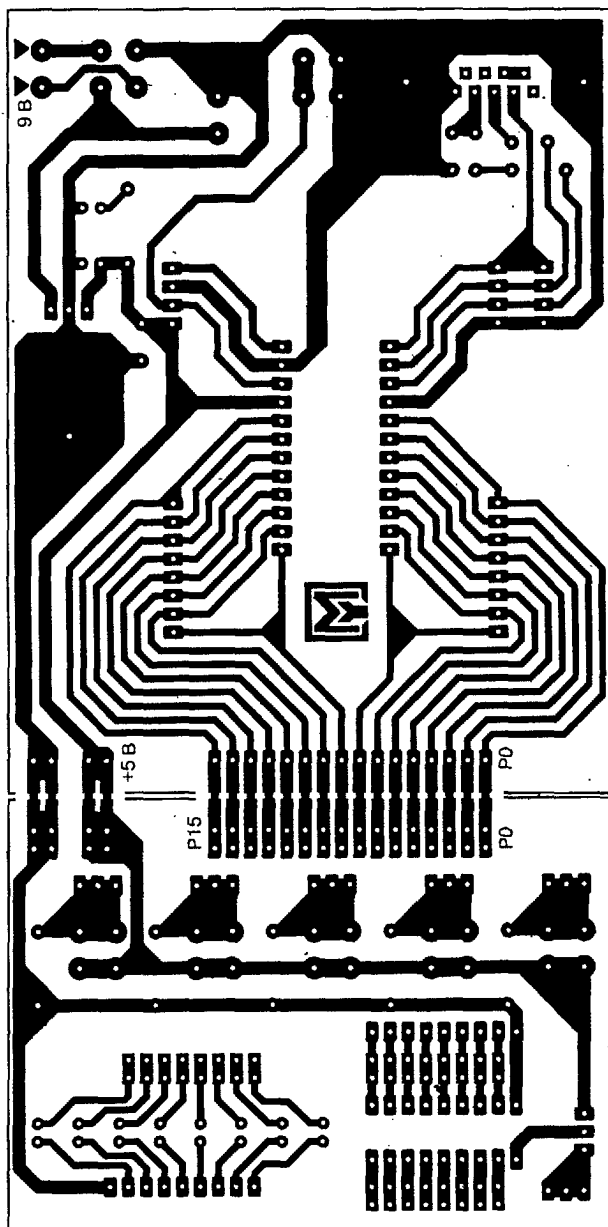
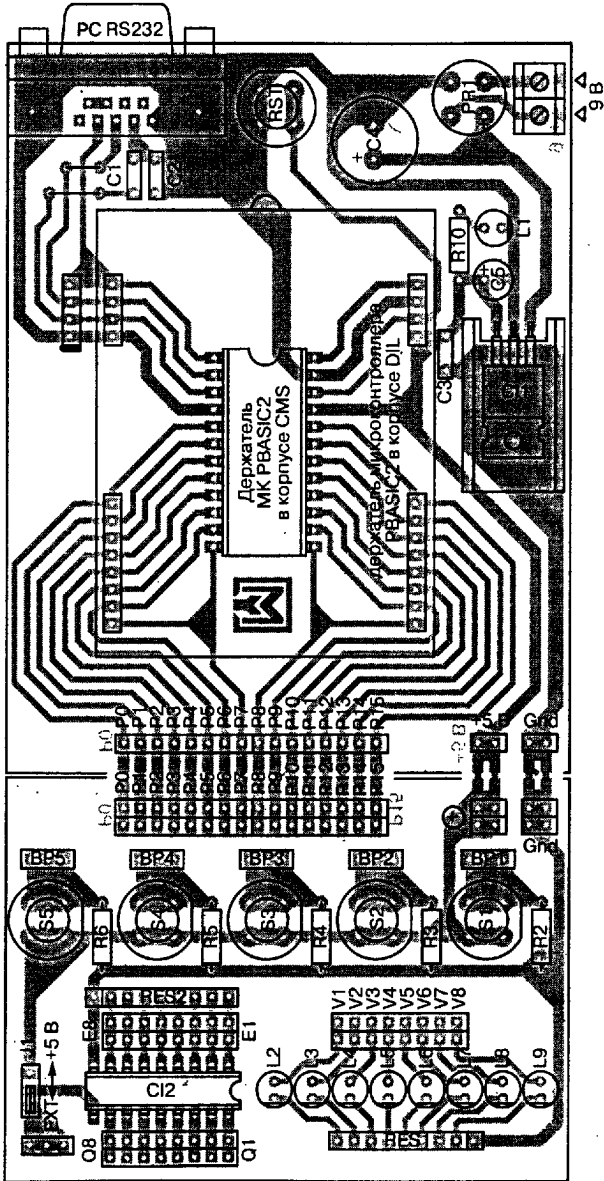


Рис. 11.5. Разводка печатной платы



Внешнее питание

Рис. 11.6. Расположение компонентов на печатной плате

Изготовление печатной платы не представляет особой сложности и может осуществляться по любой выбранной технологии. Информацию по различным технологиям литографии вы можете найти в библиографии в конце книги.

Сборку устройства начните с запаивания 2 перемычек, расположенных на плате программатора. При монтаже элементов их следует запаивать на плату в очередности, определяемой размерами и степенью прочности элементов.

Придерживайтесь следующего порядка: резисторы, держатели микросхем, все разъемы, в частности, разъемы «тюльпан» для подсоединения платы BS2 в корпусе DIP к программатору; резистивные сборки, керамические конденсаторы, диодный мостик, кнопки управления, светодиоды, разъем подключения к компьютеру DB9, гнездовые разъемы, запаиваемые в плату в один ряд, разъем перемычки J1, полярные танталовые или электролитические конденсаторы, линейный стабилизатор С13, установленный на радиатор.

Перед первым включением схемы, внимательно проверьте качество пайки и состояние дорожек. Проверьте также номиналы всех элементов и их ориентацию.

При первом подключении напряжения питания не подключайте BASIC STAMP 2 и компьютер к схеме. Питание схемы осуществляйте с помощью батарейки 9 В или от выпрямителя переменного напряжения 9 В.

Проверьте все питающие напряжения в различных местах платы, подсоедините плату тестирования при помощи гибкого плоского кабеля. Если все работает нормально, вы можете приступить к программированию этого замечательного с нашей точки зрения микроконтроллера. Перед тем, как начать работу с этим микроконтроллером, запомните *золотое правило, заключающееся в том, что ни в коем случае нельзя подсоединять или отсоединять BS2 при включенном питании*. Также при выполнении любых работ с микроконтроллером используйте антистатические браслеты для защиты микросхемы от статического электричества.

11.4. Программирование

Теперь вы можете начать программировать несложный образец, чтобы ближе познакомиться с данным устройством.

Вначале запустите программу STAMPW.EXE. Для этого в меню **Правка** (Edit) откройте вкладку **Задание параметров** (Preferences) и настройте конфигурацию COM-порта, который будет использоваться для связи с устройством. Выбор последовательного порта может осуществляться автоматически.

Подсоедините к компьютеру программатор с BS2. Подключите к программатору питание от батарейки 9 В или от блока выпрямления переменного напряжения 9 В. Открыв меню **Пуск** (Run), затем меню **Идентификация** (распознавание) (Identify), вы увидите сообщение на экране компьютера о определенном им устройстве BS2.

С этого момента вы можете ввести листинг 11.1 в программу TESTPROG.BS2. Эта программа является введением в язык программирования PBASIC2. В программе есть блоки программирования регистра направления каналов P0–P15, программирования и чтения данных в память EEPROM, тестирования входов, конфигурирования выходов и блок цикла.

С помощью этой программы направляются сигналы на 8 светодиодов, с двумя различными периодами следования, которые выбираются в зависимости от нажатой кнопки.

Список компонентов:

Резисторы с 5-процентным допуском:

- R1: 470 Ом (желтый, фиолетовый, коричневый);
- R2–R6: 10 кОм (коричневый, черный, оранжевый).

Конденсаторы:

- C1–C3: 0,1 мкФ неполярные керамические;
- C4: 1000 мкФ / 25 В (полярный танталовый или электролитический с радиальными выводами);
- C5: 10–47 мкФ / 25 В (полярный танталовый или электролитический с радиальными выводами).

Полупроводники:

- CI1: 7805;
- CI2: ULN2803;
- L1: светодиод диаметром 5 мм;
- L2-L9: светодиод диаметром 3 мм;
- PR1: диодный мостик 50 В/1 А (W04).

Прочие элементы:

- 1 держатель микросхемы с 24 выводами;
- 6 кнопок;
- 1 радиатор для TO220;
- 1 разъем DB9 розетка;
- однорядный гнездовой разъем;
- переключатель;
- разъемы «тюльпан»;
- кабельные разъемы для подсоединения микроконтроллера к ПК.

Ниже приведен листинг для подключения проводов на плате тестирования для данной программы:

Листинг 11.1. Пример тестирования программатора

```
' === ПРОГРАММА ТЕСТИРОВАНИЯ ПРОГРАММАТОРА ===
' === НА PBASIC2 © Yves Mergy 2001 ===

DATA 129,66,36,24,36,66,129,0 ' Внесение 14 значений ...
DATA 129,195,219,255,219,195. ' ... в память EEPROM.

ADDRESS      var   nib   'Переменная АДРЕС типа полубайт
MEANING      var   byte  'Переменная ЗНАЧЕНИЕ типа байт
TIME        var   nib   'Переменная ДЛИТЕЛЬНОСТЬ типа
                    'полубайтMAXIcon 13
                    'Константа МАКСИ со значением 13

DIRS=%11111111100000000    'Входы P0-P7, выходы P8-P15
START:
OUTH=0                    'P8-P15 на нижнем логическом
уровне
If IN0=1 then TEMP01      'Если вход P0=1, то переход к
                    'TEMP01
```


If IN1=1 then TEMPO2	'Если вход P1=1, то переход к TEMPO2
If IN2=1 then TEMPO3	'Если вход P2=1, то переход к TEMPO3
goto START	'Зацикливание в случае отсутствия активированных клавиш
TEMPO1:	
TIME=1	
goto READ	
TEMPO2:	
TIME=5	
goto READ	
TEMPO3:	
TIME=10	
READ:	
for ADDRESS=0 to MAXI	'Цикл чтения из 14 значений
if IN3=0 then CONTINUATION	'Если вход P3=0, чтение ...
goto START	'... в противном случае возврат в начало.
CONTINUATION:	
Read ADDRESS, MEANING	'Чтение текущего адреса
OUTH=MEANING	'Подтверждение выходов P8-P15
pause TIME*100	'Ожидание в мс. 100_ длительность
next	'Следующий адрес
goto	'Зацикливание в начало.

Произведите следующие действия:

- соедините P0 с выводом BP5 кнопки S5;
- соедините P1 с выводом BP4 кнопки S4;
- соедините P2 с выводом BP3 кнопки S3;
- соедините P3 с выводом BP1 кнопки S1;
- соедините P8 с выводом V1 индикаторов;
- соедините P9 с выводом V2 индикаторов;
- соедините P10 с выводом V3 индикаторов;
- соедините P11 с выводом V4 индикаторов;
- соедините P12 с выводом V5 индикаторов;
- соедините P13 с выводом V6 индикаторов;
- соедините P14 с выводом V7 индикаторов;
- соедините P15 с выводом V8 индикаторов.

После того, как вы закончили ввод программы, запустите ее, одновременно нажав кнопки Ctrl+R. Запуск программы можно также осуществить через вкладку Запуск (Run) меню Запуск (Run) или нажатием на кнопку в виде стрелки под

панелью меню. В случае ошибки программа показывает на ошибочное место. В случае успеха начинается загрузка, и программа обращается к BASIC STAMP 2.

После этого микроконтроллер BS2 работает в автономном режиме.



Компьютер после программирования больше не используется, его можно использовать по прямому назначению.

При нажатии на одну из трех кнопок S1-S3 светодиоды начинают светиться с различными периодами следования. Нажатие на кнопку S5 останавливает индикацию включенного светодиода до следующего нажатия на одну из кнопок S1-S3. Этот небольшой пример показывает простоту использования этого замечательного (не побоюсь этого слова) микроконтроллера. В следующих главах мы приступим к изучению более сложных приложений, использующих цифровую логику, основанную на BASIC STAMP 2.

Часть

От теории к практике

Последняя часть этой книги базируется на полученных теоретических знаниях для выполнения практических устройств. С помощью трех больших проектов мы увидим, как можно создать одни и те же устройства по разным технологиям. Вначале мы перейдем к программированию на РС, а затем к использованию микроконтроллера.

Мы предлагаем вам изучить схемы, использующие каждую из предложенных технологий. От логических схем на «жесткой логике» мы постепенно перейдем к цифровой электронике.

Глава 12

Охранная система

12.1. Общее описание

Первым из наших практических проектов будет простая охранная система обрывного действия, основанная на определении обрыва одной или нескольких зон охраны (всего охранных зон – 4). Такой выбор обусловлен двумя факторами. Во-первых, охранные системы широко распространены. Кроме того, проектирование охранной системы может быть осуществлено тремя разными способами: на основе цифровой логики, на основе программируемой логики и с помощью микроконтроллера. Мы закрепим пройденный материал этой книги разработкой схем – теорию мы соединим с практикой.

Независимо от способа изготовления наши охранные системы обладают следующими общими свойствами:

- 4 датчика непосредственного контроля без задержки;
- 4 датчика контроля с задержкой по времени;
- регулируемое время входа для сотрудника охраны – «снятие с охраны»;
- регулируемое время выхода для сотрудника охраны – «постановка под охрану»;
- возможность регулирования времени срабатывания сирены;
- индикация различных состояний;
- управление сиреной без использования реле.

12.2. Схема на основе цифровой логики

Этот практический проект нельзя отнести к самым простым. Он дает возможность продемонстрировать на практике работу различных логических элементов. В схеме используются только логические элементы, изученные в первой части книги. Например, будет показано, что D-триггер может осуществлять задержку информации (рис. 12.1).

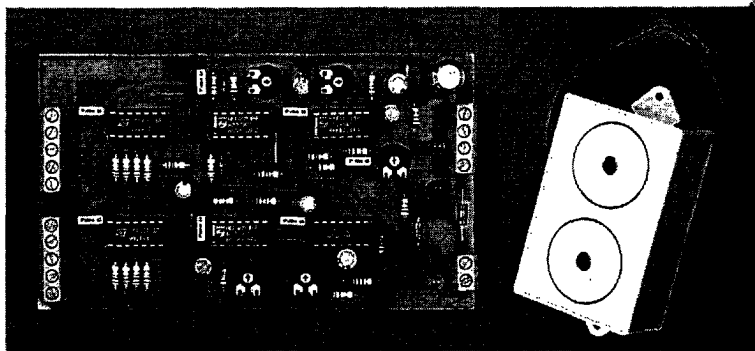


Рис. 12.1. Модуль охранной системы на основе цифровой логики и звуковое устройство оповещения – сирена

Принципиальная схема

Для лучшего изучения схемы на рис. 12.2, выполняющей логические функции охранной системы, разобьем ее на несколько составных частей. Данная охранная система состоит из следующих составных узлов:

- схема охраны объектов с задержкой, включающая схему обнаружения помехи и схему задержки включения сирены;
- схема непосредственной охраны объекта, включающая в себя схему обнаружения помехи и схему задержки включения сирены;
- схема управления сиреной;
- индикация режимов работы;
- схема питания;
- узел выходной задержки охраняемого объекта.

Выходная задержка осуществляется медленным зарядом конденсатора С1 через резистор R9 и переменное сопротивление АJ1, подключенных ко входу 2-входовой схемы 2И-НЕ в инверторном включении с триггером Шмитта N8 во время установки питания. При выключении напряжения питания конденсатор С1 быстро разряжается через резистор R9 и блокировочный диод D9. Далее сигнал с выхода схемы N8 разветвляется на две части: на вход N16, выполняющего логическую функцию ИЛИ-НЕ, после задержанного обнаружения и на вход логического элемента И-НЕ, микросхемы N5 после инверсии элементом N7 для включения сирены.

Схема замедленного обнаружения вмешательства защищает 4 зоны. В спокойном состоянии (нет тревоги) 4 входа элемента ИЛИ-НЕ должны быть связаны с «землей» через блокировочные диоды D1–D4. Тогда на выводе 1 схемы 4ИЛИ-НЕ N1 будет логическая «1». Если один из контуров охраны разомкнут (нет контакта, порванный провод и т.д.), то на входах 2, 3, 4, 5 микросхемы N1 появляется потенциал, обусловленный напряжением +12 В (уровень логической единицы). Вследствие чего выход 1 N1 переходит в состояние «0» и затем инвертируется в состояние «1» элементом N2 (вторая половина схемы 4И-НЕ, включенная как инвертор, когда все 4 входа которого соединены между собой). Единичный сигнал с выхода 13 инвертора N2 через диод D13 действует на схему индикации. Также сигнал с этого вывода управляет включением генератора, который требуется для образования пачки импульсов в случае нарушения безопасности. Его принцип был изучен в первой части книги. Генератор выполнен на схеме 2И-НЕ с триггером Шмитта N11, с времязадающими элементами – резистором R11 и конденсатором С2. Выход генератора подсоединен ко второму входу 13 N16. От него в случае нарушения пачка импульсов подается на вход 3 CLK триггера N17.

Триггер N17 с «электронной обвязкой» составляют входной датчик, что, согласитесь, не совсем обычная схема для триггера, но достаточно эффективная. Аналогичная интегрирующая цепь С3-R12 подключена ко входу 4 R и сбрасывает триггер N17 в 0 при включении напряжения питания. По каждому импульсу, поступающему на вывод 3 CLK, выходной вывод 1 Q переключается в состояние 1 (так как вывод 5 D

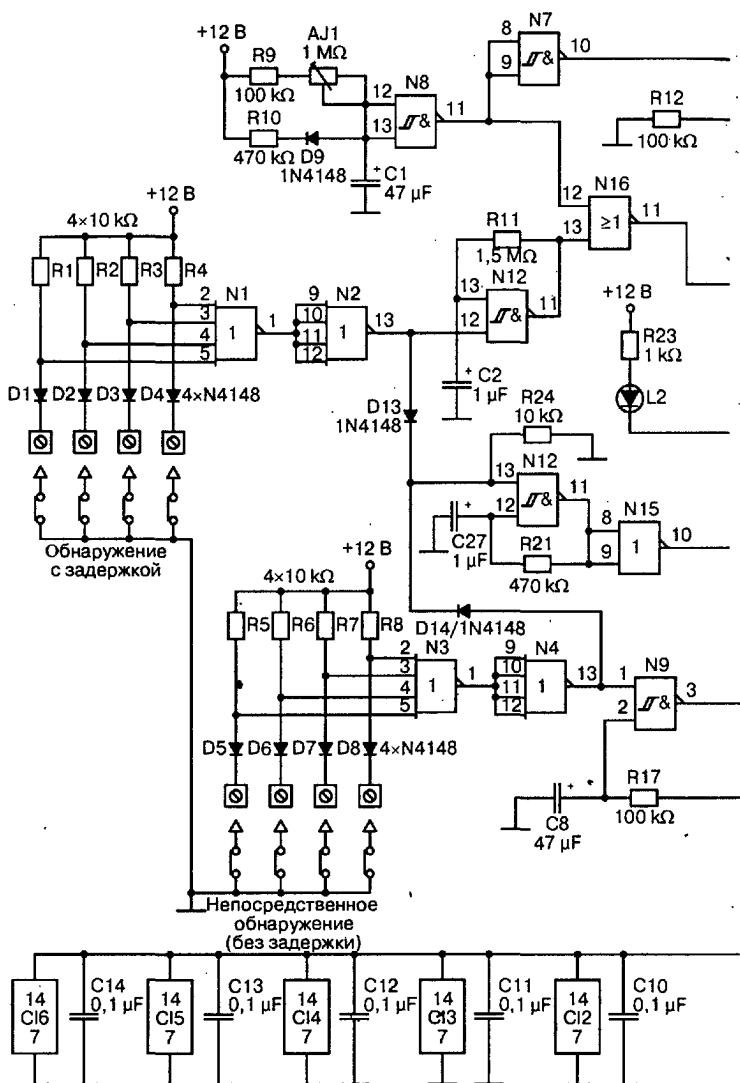


Рис. 12.2. Схема устройства на основе цифровой логики

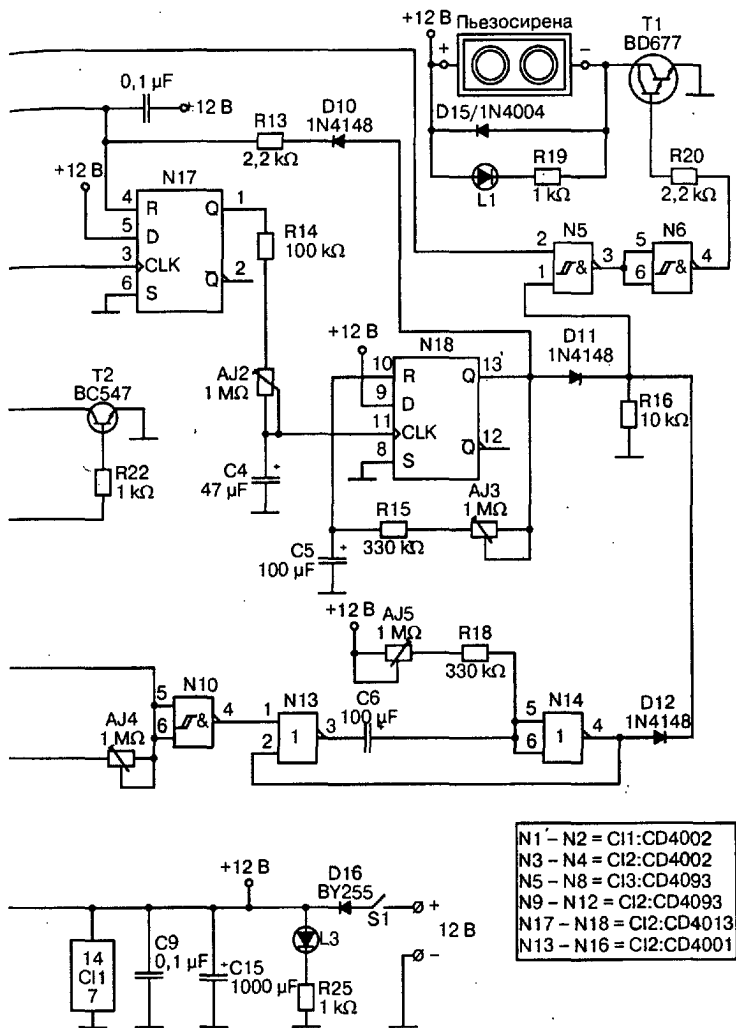


Рис. 12.2. (продолжение) Схемa устройства на основе цифровой логики

подсоединен к +12 В), если он не был раньше в этом состоянии. Это вызывает медленный заряд конденсатора С4, через резистор R12 и переменное сопротивление AJ2.

Когда напряжение на положительной обкладке конденсатора С4 достигает уровня, достаточного чтобы вызвать импульс на входе 11 CLK триггера N18, его выход 13 Q переходит в состояние логической единицы и через диод D10 и резистор R13 по входу R устанавливает триггер N17 в состояние 0 и на выходе Q. N18 определяет время срабатывания сирены для обнаружения несанкционированного вторжения на охраняемую территорию. Выход Q в состоянии «1» также активирует вывод 1 N5 через блокировочный диод D11 и заряжает конденсатор С5 через резистор R15 и переменное сопротивление AJ3. Когда напряжение на конденсаторе достигает достаточно уровня, на соединенном с ним выводе R установится положительный потенциал, которым триггер N18 установится в 0, вывод 13 Q также переходит в состояние «0», и цикл может начинаться вновь, когда новый импульс придет на тактовый вход CLK. Если входной триггер долгое время остается открытым, конденсатор С4 не может разрядиться полностью, приводя на короткое время отключению сирены. Сирена будет включаться прерывисто, привлекая внимание окружающих.

Цепь непосредственного обнаружения вмешательства на элементах N3, N4 также защищает 4 зоны, ее работа идентична описанной выше, до выхода 13 элемента N4. При нарушении на одном из входов (2–5) вместо 0 установится 1 и после двойного инвертирования положительный сигнал через диод D14 откроет вход 13 N13, тем самым активируя медленный генератор, построенный на схеме 2И-НЕ с триггером Шмитта. Его времязадающая цепочка на конденсаторе С8, резисторе R17 и переменном сопротивлении AJ4 должна вырабатывать импульсы для включения сирены во время незамедлительного обнаружения. Положительный сигнал с вывода 13 N4 поступает также на вход 1 N9 и после двойного инвертирования (элементами N9, N10) поступает на вход 1 N13 и управляет работой схемы с одним устойчивым положением, определяющей время паузы между включением сирены во время незамедлительного обнаружения. Сирена работает в двухтональном режиме – работает с кратковременными пау-

зами, затем длинная пауза и затем повторяется все сначала, пока не устранится обрыв одного из проводов, протянутых вокруг охраняемой территории. Эта схема одновибратора (с одним устойчивым состоянием) состоит из элементов N13 и N14, конденсатора C6, резистора R18, переменного сопротивления AJ5. Сигнал с выхода 4 N4 одновибратора действует так же на второй вход N5 через блокировочный диод D12.

Управление сиреной происходит в любом случае через логический элемент И-НЕ N5. Управляющий сигнал должен включать сирену только когда есть нарушение и когда происходит выходная задержка. При отсутствии положительного сигнала на выводе 1 N5, через резистор R16 этот вход подключен к «земле», и сирена не включена. Элементы N5 и N6 управляют составным транзистором Дарлингтона T1 через сопротивление базы R20. По поступлению разрешающего сигнала транзистор без теплоотвода включает сирену-пьезоизлучатель, состоящую из нескольких пьезопластин под напряжением 12 В*. Диод D15 защищает T1 от обратных токов пробоя в случае использования вместо сирены реле. Светодиод L1, ток которого ограничивается резистором R19, показывает проводящее состояние T1.

Оптическая индикация факта нарушения периметра охраны осуществляется генератором, заставляющим мигать светодиод в его нагрузке с частотой колебаний генератора. Он построен на элементах N12, N15 с времязадающей цепью из конденсатора C7 и резистора R21. В состоянии покоя, генератор блокируется низким уровнем через резистор R24**. Сигнал выхода N12 инвертируется N15, чтобы воздействовать на базу транзистора T2 через сопротивление R22. Светодиод L2, ограниченный по току сопротивлением R23, мигает в ритм генератору при нарушениях территории охраны.

Непрерывная нагрузка – это выход напряжения 12 В, предусмотренного чтобы запитать необходимым током сирену.

* В отличие от других типов сирен, сирена на пьезоизлучателе вырабатывает большое звуковое давление (до 100–120 дБ, за которым наступает болевой порог для ушей человека), имеет достаточно большое внутреннее сопротивление (единицы кОм, а обычные динамики – единицы Ом), и за счет этого ток через них составляет десятки мА и мощный транзистор для управления сиреной не требуется. Поэтому ставить его на теплоотвод не обязательно. – *Прим. науч. ред.*

** Мы знаем, какой бывает покой, с шашлыками и другими радостями жизни, пока его не нарушит какой-нибудь бомж или солдат, сбежавший из армии. – *Прим. науч. ред.*

Выключатель S1 позволяет выключить систему. Он должен быть надежно спрятан от лишних глаз в жилище или машине. Диод D6 защищает схему от переплюсовки во время подключения питания. Конденсатор C5 фильтрует основное напряжение, C9–C14 фильтруют конкретно каждый ИМС C11–C6.

Сборка

На рис. 12.3 показан чертеж печатной платы в масштабе 1:1. Плотность разводки платы довольно высокая. Как и все схемы в данной книге, она может быть изготовлена любым из методов литографии на ваш выбор. Материалы по технологии литографии вы найдете в библиографии в конце книги.

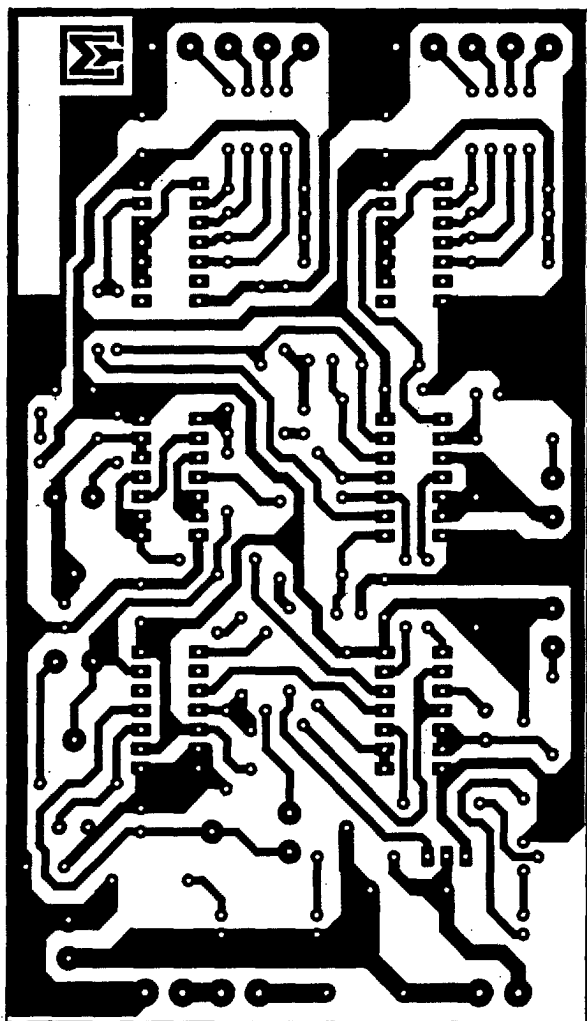
Тщательно изучите монтажную схему сигнализации с расположением компонентов, представленную на рис. 12.4, чтобы избежать ошибок. Чтобы не забыть, в первую очередь рекомендуется запаять 10 перемычек.

Вы можете их изготовить из монтажного провода. Детали следует запаивать на плату в очередности, определяемой размерами и степенью прочности элементов. Придерживайтесь следующего порядка: диоды (при пайке их не следует перегревать), резисторы, держатели микросхем, переменные сопротивления, неполярные керамические конденсаторы, транзисторы, штырьевые разъемы, светодиоды, и, наконец, полярные танталовые или электролитические конденсаторы.

Список компонентов:

Резисторы с 5-процентным допуском:

- R1–R8; R16; R24: 10 кОм (коричневый, черный, оранжевый);
- R9; R12; R14; R17: 100 кОм (коричневый, черный, желтый);
- R10: 470 Ом (желтый, фиолетовый, коричневый);
- R11: 1,5 МОм (коричневый, зеленый, зеленый);
- R13, R20: 2,2 кОм (красный, красный, красный);
- R15, R18: 330 Ом (оранжевый, оранжевый, желтый);
- R19, R22, R23, R25: 1 кОм (коричневый, черный, красный);
- R21: 470 кОм (желтый, фиолетовый, желтый).



с. 12.3. Рисунок печатной платы охранной сигнализации на основе элементов цифровой логики (вид со стороны печатной платы)

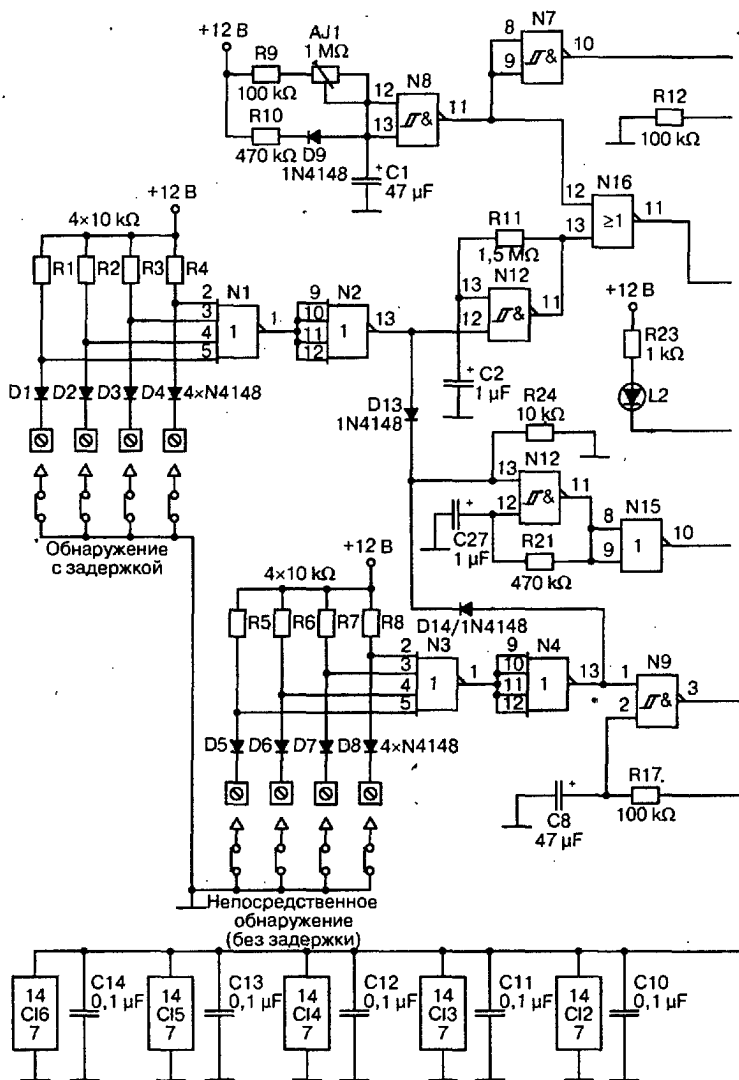


Рис. 12.4. Расположение компонентов на печатной плате

Конденсаторы:

- C1, C4, C8: 47 мкФ / 25 В (полярный танталовый или электролитический с радиальными выводами);
- C2, C7: 1 мкФ / 25 В (полярный танталовый или электролитический с радиальными выводами);
- C3, C9–C14: 0,1 мкФ, неполярный керамический;
- C5, C6: 100 мкФ / 25 В (полярный танталовый или электролитический с радиальными выводами);
- C15: 1000–2200 мкФ / 25 В (полярный танталовый или электролитический с радиальными выводами).

Полупроводники:

- CI1, CI2: CD4002;
- CI3, CI4: CD4093;
- CI5: CD4013;
- CI6: CD4001;
- T1: BD667, T2: BC547;
- D1–D4: 1N4148;
- D15, D16: 1N4007 или BY255 (в зависимости от мощности устройств сигнализации);
- L1–L3: светодиод 5 мм (любой цвет на выбор).

Прочие элементы:

- AJ1–AJ5: переменное сопротивление 1 Мом горизонтальной установки;
- 6 держателей микросхемы с 14 выводами;
- 2 блока винтовых разъемов с 5 выводами и 1 блок винтовых разъемов с 4 выводами;
- 1 блок винтовых разъемов с 2 выводами;
- 1 сирена на пьезоизлучателе с 2 или 4 пьезопластинами;
- 1 выключатель (простой или с ключом).

После выполнения последней пайки тщательно проверьте вашу работу: как качество печатной платы (пайку и состояние дорожек), так и элементы (номиналы и расположение, правильность установки полярности электролитических конденсаторов и полупроводниковых диодов и транзисторов).

Во время первого включения платы не вставляйте микросхемы в держатели. Проверьте присутствие напряжения питания на выводе 14 («плюс» напряжения питания) относительно вывода 7 («минус») каждого держателя микросхем. После выполнения всех проверок вставьте при отключенном напряжении питания микросхемы в держатели, обращая внимание на их цоколевку, ключ и название!

Процесс регулировки сигнализации не должен вызвать больших проблем, так как если все номиналы элементов подобраны правильно, то достаточно выставить 5 переменных резисторов в среднее положение до первого включения устройства. Затем следует отрегулировать различные времена срабатывания, действуя из соображений логики и в зависимости от различных ситуаций (включение под напряжением, задержка выхода, постоянное или быстрое вмешательство, задержка входа, аварийный сигнал, включение без напряжения). Чтобы избежать ранней глухоты и возмущений других людей – соседей по даче или квартире – при проведении испытаний разумно использовать вместо сирены реле и использовать индикацию светодиода L1.

Ниже представлена табл. 12.1, в которой даны названия элементов с номиналами и точными временами, с помощью которых настраиваются управляющие времена работы.

Таблица 12.1. Регулировочные элементы и управляющие времена работы

Обнаружение	Задержка	Конденсатор, мкФ	Переменное сопротивление, МОм	Резистор, кОм	Задержка времени, с
Любое	Выход	C1 = 47	AJ1 = 1	R9 = 100	3–36
С задержкой	Вход	C4 = 47	AJ2 = 1	R9 = 100	3–36
	Сирена	C5 = 100	AJ3 = 1	R9 = 330	23–93
Без задержки	Сирена	C6 = 100	AJ5 = 1	R9 = 330	23–93
	С интервалами	C8 = 47	AJ4 = 1	R9 = 100	4–41

12.3. Схема охранной сигнализации на основе программируемой логики

Использование компьютера в разработке схемы охранной сигнализации позволяет сильно упростить электронную часть. Однако одно должно заместить другое: следует дополнительно разработать программу. Кроме того, устройство теряет автономность. В качестве языка программирования предлагается использовать популярный QBASIC, который полностью описывает работу схемы, как показывает этот проект (рис. 12.5).

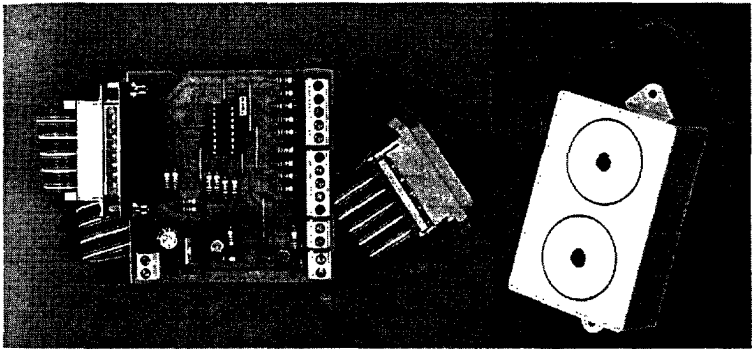


Рис. 12.5. Охранная сигнализация на основе программируемой логики со звуковой сигнализацией-сиреной

Принципиальная схема

Наша охранная сигнализация включает 8 обрывных датчиков контроля (4 датчика непосредственного контроля и 4 датчика с задержкой по времени). К сожалению, параллельный порт РС содержит только 5 входов, тогда как для нашей схемы требуется 8 входов! Решить эту проблему можно с помощью микросхемы С11, позволяющей «раздвоить» 4 входа на 8.

Рассматривая схему на рис. 12.6, вы можете увидеть, что электроника всех восьми датчиков идентична. Диоды D1–D8 являются блокировочными, они предотвращают короткое замыкание напряжения +5 В на «землю». В случае размыкания одного из датчиков контроля на соответствующий вход микросхемы С11 поступает логическая единица через резисторы R1–R8.

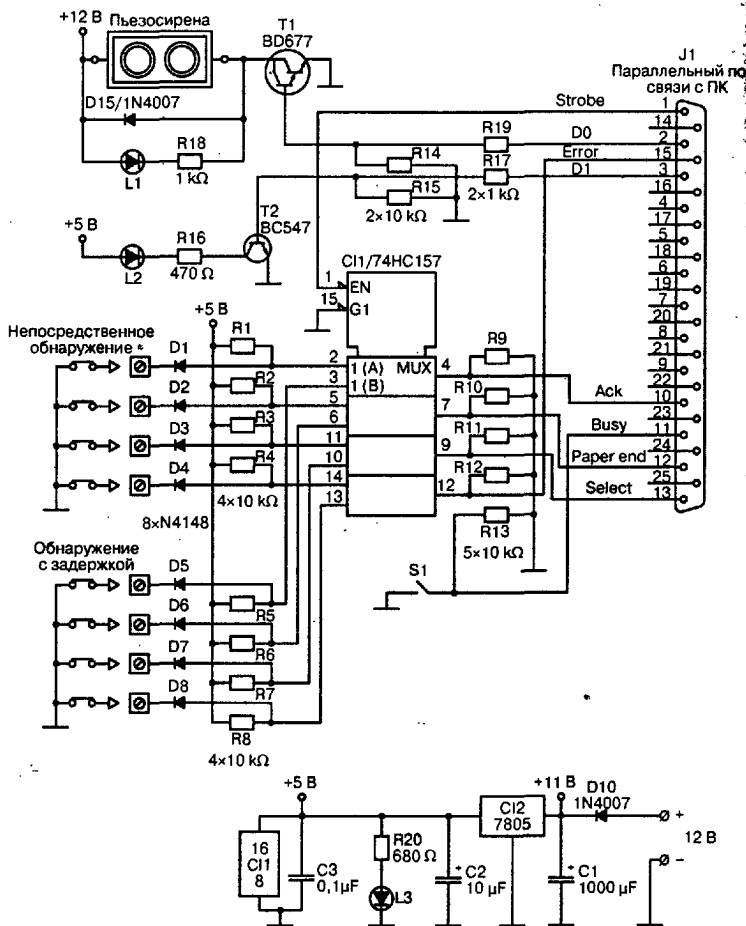


Рис. 12.6. Принципиальная схема охранной сигнализации на основе программируемой логики

Микросхема С11 (74НС157) является четырехканальным мультиплексором 2 к 1. Другими словами, он включает 4 группы входов по 2 в каждом (А и В), то есть нужных нам 8 входов, при этом число выходов равно только 4. Когда низкий логический уровень подается на вывод 1 EN С11, то с выходом соединяются входы А (входы 2,5,11,14). Если же на вывод 1 подается высокий логический уровень, то с выходом соединяются входы В (входы 3,6,10,13). Можно легко заметить, что с входами А С11 связаны датчики непосредственного контроля, а с входами В – датчики контроля с задержкой по времени. Четыре выхода соединяются с каналами АСК, PAPER END, SELECT и ERROR параллельного порта РС. Резисторы R9–R12 подключают выходы С11 к определенному потенциалу в состоянии покоя – к «земле».

Канал STROBE управляет непосредственно переключением входов: вывод 1 микросхемы С11.

Выключатель S1 служит для включения или выключения охранной сигнализации. Канал BUSY служит для считывания состояния выключателя S1. Резистор R13 подключает к «земле» выходную линию, если выключатель S1 открыт.

Сирена управляется по каналу D0. Так как выходы параллельного порта связи с ПК не предназначены для передачи большой мощности, то ток предварительно усиливается с помощью составного транзистора Дарлингтона Т1. Сигнал с выхода D0 поступает на вход транзистора Т1 через резистор базы R19, а в коллекторе Т1 подключена нагрузка – пьезосирена. В состоянии покоя резистор R14 запирает транзистор Т1, заземляя его базу. Дiode D9 защищает транзистор от тока размыкания при использовании реле для проверки работоспособности схемы. Светодиод L1, ток которого ограничивается резистором R18, показывает индикацию включенного состояния сирены.

Светодиод L2, ток которого ограничивается резистором R16, управляется транзистором Т2 и начинает светиться сразу после срабатывания одного из восьми датчиков. Он также управляется по каналу D1, но через транзистор Т2. Рабочая точка транзистора задается с помощью резисторов R17 и R15.

Охранная сигнализация питается от напряжения 5 В, чтобы обеспечить совместимость с компьютером. В то же время, большинство сирен работает от постоянного напряжения 12 В. Поэтому питание на схему управления пьезосиреной подается

от источника 12 В. Диод D10 защищает схему от переплюсовки напряжения во время подключения питания. Далее напряжение +12 В фильтруется с помощью конденсатора С1 и подается на сирену. Напряжение 5 В вырабатывается с помощью линейного стабилизатора С12. Конденсатор С2 отфильтровывает переменные пульсации напряжения 5 В. Светодиод L3, ограниченный резистором R20, показывает наличие в схеме питания. Конденсатор С3 дополнительно стабилизирует напряжение питания ИМС С11.

Сборка

На рис. 12.7 показан рисунок печатной платы. Материалы по технике изготовления плат вы найдете в библиографии в конце книги. При изготовлении платы постарайтесь избежать короткого замыкания контактов разъема DB25 для соединения с

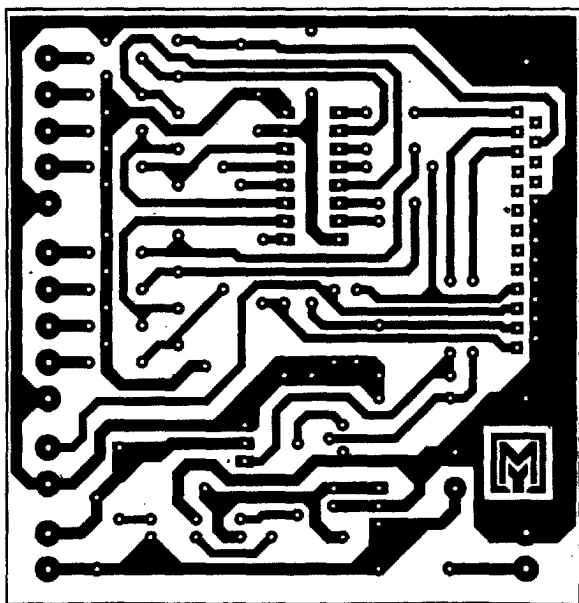


Рис. 12.7. Рисунок печатной схемы охранной сигнализации на основе программируемой логики

РС, так как они расположены довольно близко.

Рисунок монтажной платы представлен на рис. 12.8. Чтобы

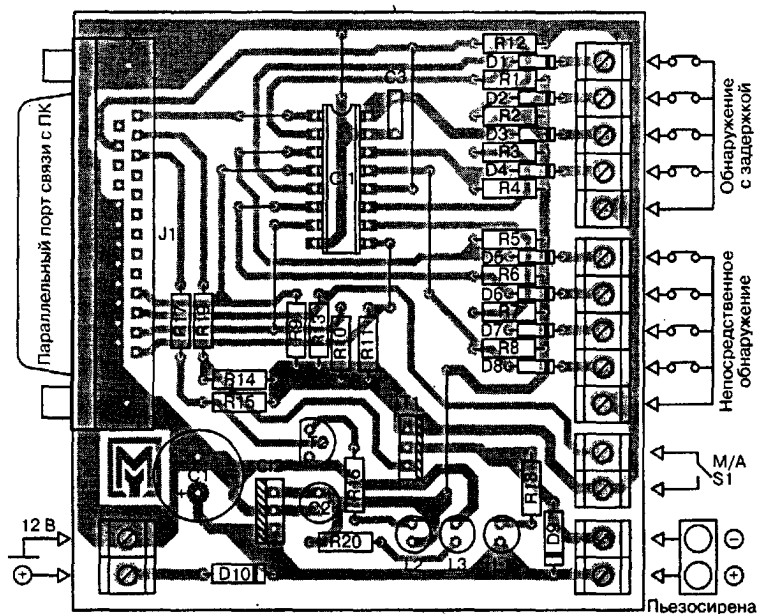


Рис. 12.8. Монтаж компонентов на плате охранной сигнализации на основе программируемой логики

не забыть, в первую очередь рекомендуется запаять 10 перемычек. Вы можете их изготовить из монтажного провода.

Детали следует запаивать на плату в очередности, определяемой размерами и степенью прочности элементов. Начните с пайки резисторов, затем запаяйте диоды, держатель микросхемы С11, керамический конденсатор, транзистор Т2, разъем DB25, штырьевые разъемы, светодиоды, транзистор Т1, линейный стабилизатор напряжения и, наконец, полярные танталовые или электролитические конденсаторы. При запаивании взаимозаменяемых транзисторов соблюдайте цоколевку!

После выполнения всех работ тщательно проверьте сделанное, чтобы избежать короткого замыкания между дорожками в

результате небрежной пайки элементов. Перед тем, как вставить микросхему С11 в держатель и подсоединить плату к компьютеру с помощью кабеля DB25, проверьте наличие напряжения питания на всех участках схемы. При отсутствии проблем вставьте микросхему С11 в держатель, подсоедините плату к компьютеру и подключите питание к схеме.

Список компонентов:

Резисторы с 5-процентным допуском:

- R1–R15: 10 кОм (коричневый, черный, оранжевый);
- R16: 470 Ом (желтый, фиолетовый, коричневый);
- R17–R19: 1 кОм (коричневый, черный, красный);
- R20: 680 Ом (синий, серый, коричневый).

Конденсаторы:

- C1: 1000 мкФ / 25 В (полярный танталовый или электролитический с радиальными выводами);
- C2: 10 мкФ / 25 В (полярный танталовый или электролитический с радиальными выводами);
- C3: 0,1 мкФ, неполярный керамический.

Полупроводники:

- D1–D4: 1N4148;
- D9, D10: 1N4007 или BY255 (в зависимости от мощности сигнализации);
- L1–L3: светодиод 5 мм (любой цвет на выбор);
- T1: BD667;
- T2: BC547;
- C11: 74НС157;
- C12: 7805.

Прочие элементы:

- 1 держатель микросхемы с 16 выводами;
- штырьевые разъемы;
- 1 пьезосирена с 2 или 4 пьезопластинами;

- 1 выключатель (простой или с ключом); .
- 1 угловой разъем DB25 для печатной платы.

Программирование

Листинг 12.1 программы (ALARME.BAS), написанной на QBASIC, можно легко переделать на другие языки программирования. Каждый блок программы включает подробные пояснения для лучшего понимания. С помощью кнопки F1 можно открыть меню помощи для объяснения непонятных мест в программе.

Листинг 12.1. Программа станции охранной сигнализации, управляемой с ПК

```
REM === ПРОГРАММА на QBASIC СТАНЦИИ ОХРАННОЙ СИГНАЛИЗАЦИИ С
УПРАВЛЕНИЕМ ОТ ПК ===
REM === Для параллельного порта LPT1. © Yves Mergy 2001 ===

' === Объявление переменных и констант. ===
DIM EN(16) AS INTEGER
T0$ = "ЗАМКНУТО": T1$ = "РАЗОМКНУТО"
TEMPOSIL = 60: TEMPOSIR = 30: TEMPOENT = 10: TEMPOSOR = 10

' === Инициализация. ===
CUT (888), 0
' === Отображение на главном экране. ===
SCREEN 0, 0, 0
LOCATE 3, 27: PRINT "СТАНЦИЯ ОХРАННОЙ СИГНАЛИЗАЦИИ НА ПК"
LOCATE 6, 15: PRINT "Зона 1 непосредственного срабатывания :
ЗАМКНУТА"
LOCATE 7, 15: PRINT "Зона 2 непосредственного срабатывания :
ЗАМКНУТА"
LOCATE 8, 15: PRINT "Зона 3 непосредственного срабатывания :
ЗАМКНУТА"
LOCATE 9, 15: PRINT "Зона 4 непосредственного срабатывания :
ЗАМКНУТА"
LOCATE 11, 15: PRINT "Зона 1 срабатывания с задержкой : ЗАМ-
КНУТА"
LOCATE 12, 15: PRINT "Зона 3 срабатывания с задержкой : ЗАМ-
КНУТА"
LOCATE 13, 15: PRINT "Зона 3 срабатывания с задержкой : ЗАМ-
КНУТА"
LOCATE 14, 15: PRINT "Зона 4 срабатывания с_задержкой : ЗАМ-
КНУТА"
LOCATE 16, 15: PRINT "Переключатель разомкнут ==> Сигнали-
зация не включена"
```

```
LOCATE 17, 15: PRINT "Переключатель замкнут ==> Сигнализа-
ция в режиме наблюдения"
```

```
' === Установка в режим наблюдения. ===
```

```
НАЧАЛО:
```

```
GOSUB ЧТЕНИЕ ПЕРЕКЛЮЧАТЕЛЯ
```

```
IF ПЕРЕКЛЮЧАТЕЛЬ = 0 THEN GOTO НАЧАЛО
```

```
' === Вызов задержки выхода. ===
```

```
DP = 0
```

```
TXS = " выхода ==> "
```

```
TPS = TEMPOSOR
```

```
GOSUB TEMPO
```

```
GOSUB ЧТЕНИЕ ВЫКЛЮЧАТЕЛЯ
```

```
IF INTER = 0 THEN GOTO НАЧАЛО
```

```
' === Наблюдение зон. ===
```

```
SUPETVISION: ' Метка возврата.
```

```
GOSUB READ SWITCH
```

```
IF SWITCH = 0 THEN GOTO START
```

```
GOSUB READ ZONES
```

```
IF EN(0) = 1 THEN OUT (888), 3: GOTO SIREN
```

```
IF EN(1) = 1 THEN OUT (888), 3: GOTO SIREN
```

```
IF EN(2) = 1 THEN OUT (888), 3: GOTO SIREN
```

```
IF EN(3) = 1 THEN OUT (888), 3: GOTO SIREN
```

```
IF EN(4) = 1 THEN OUT (888), 2: GOTO TEMPENT
```

```
IF EN(5) = 1 THEN OUT (888), 2: GOTO TEMPENT
```

```
IF EN(6) = 1 THEN OUT (888), 2: GOTO TEMPENT
```

```
IF EN(7) = 1 THEN OUT (888), 2: GOTO TEMPENT
```

```
GOTO SUPETVISION
```

```
' === Непосредственное срабатывание и ===
```

```
' === вызов задержки сирены. ===
```

```
SIREN:
```

```
DP = 1
```

```
TXS = "сирены ==> "
```

```
TPS = DELAY OF SIREN
```

```
IF DS = 0 THEN OUT (888), 1 else out (888), 3
```

```
GOSUB TEMPO
```

```
' === Вызов задержки тишины. ===
```

```
IF DS = 0 THEN OUT (888), 0 ELSE OUT (888), 2
```

```
DP = 0
```

```
TXS = "тишины ==> "
```

```
TPS = ЗАДЕРЖКА ТИШИНЫ
```

```
GOSUB TEMPO
```

```
' === Выключение или наблюдение в зависимости от положения
переключателя. ===
```

```
GOSUB READ SWITCH
```

```
IF INTER = 1 THEN GOTO SUPETVISION
```

```

GOTO START
' === Срабатывание с задержкой и ===
' === Вызов задержки входа. ===
ENTER WITH DELAY:
DP = 0
TX$ = " входа ==> "
TPS = ENTER WITH DELAY
GOSUB TEMPO
' === Выключение или сирена в зависимости от положения пере-
реключателя. ===
GOSUB READ SWITCH
IF INTER = 1 THEN GOTO SIREN
CUT (888), 0
GOTO НАЧАЛО

' #####=== ПОДПРОГРАММЫ #####
' === Обновление главного экрана и ===
' === активация выходов. ===
MAIN SCREEN:
FOR I = 0 TO 7
  IF EN(I) <> EN(I + 8) THEN
    IF DS = 0 AND DP = 0 THEN OUT (888), 0
    IF DS = 0 AND DP = 1 THEN OUT (888), 1
    IF DS = 1 AND DP = 0 THEN OUT (888), 2
    IF DS = 1 AND DP = 1 THEN OUT (888), 3
    IF I < 4 THEN CURS = 6 ELSE CURS = 7
    LOCATE CURS + I, 50
    IF EN(I) = 1 THEN PRINT T1$ ELSE PRINT T0$
    EN(I + 8) = EN(I)
  END IF
NEXT I
RETURN

' === Чтение положения переключателя. ===
READ SWITCH
  OUT (890), 0
  A = INP(889)
  IF (A AND 128) = 128 THEN INTER = 1 ELSE INTER = 0
RETURN

' === Чтение состояния контуров зон. ===
READ OF ZONES:
  OUT (890), 0
  A = INP(889)
  IF (A AND 8) = 8 THEN EN(7) = 1 ELSE EN(7) = 0
  IF (A AND 16) = 16 THEN EN(6) = 1 ELSE EN(7) = 0
  IF (A AND 32) = 32 THEN EN(5) = 1 ELSE EN(7) = 0
  IF (A AND 64) = 64 THEN EN(4) = 1 ELSE EN(7) = 0
  OUT (890), 1

```



```

A = INP(889)
IF (A AND 8) = 8 THEN EN(3) = 1 ELSE EN(3) = 0
IF (A AND 16) = 16 THEN EN(2) = 1 ELSE EN(2) = 0
IF (A AND 32) = 32 THEN EN(1) = 1 ELSE EN(1) = 0
IF (A AND 64) = 64 THEN EN(4) = 1 ELSE EN(0) = 0
DS = 0
FOR I = 0 TO 7
  IF EN(I) = 1 THEN DS = 1
NEXT I
RETURN
' === Задержка. ===
TEMPO
TIMER ON
LOCATE 19, 15: PRINT "Задержка"; TX$
START = TIMER
DO
  TIME = TIMER - START
  IF TIME > TPS THEN EXIT DO
  LOCATE 19, 45: PRINT USING "##.##"; TIME
  GOSUB READ SWITCH
  IF INTER = 0 THEN EXIT DO
  GOSUB READ OF ZONES
  GOSUB BASIC SCREEN
LOOP
TIMER OFF
LOCATE 19, 15: PRINT SPC(44);
RETURN

```

В первую очередь программа выводит на экран состояние всех датчиков и ожидает выключения выключателя М/А для приведения сигнализации в действие. После того, как этот этап выполнен, включается задержка на 10 секунд, которая позволяет при настройке покинуть места расположения датчиков, после чего система начинает постоянный опрос датчиков контроля. В состоянии покоя все датчики находятся под потенциалом «земли». Если срабатывает один из датчиков непосредственного контроля, то сирена срабатывает сразу и работает в течение 30 с. После этого сирена замолкает на 1 минуту. Если через это время датчик контроля остается в открытом состоянии (в состоянии обрыва), то сирена снова начинает работать. Если срабатывает датчик контроля с задержкой по времени, то задержка времени входа 10 с позволяет отключить сирену до начала работы.

На экран компьютера выводится в реальном масштабе времени состояние каждого датчика и состояние сирены с названием задержки и временем срабатывания.

12.4. Схема охранной сигнализации на основе микроконтроллера

Схема охранной сигнализации на основе микроконтроллера является самой интересной схемой для рассмотрения (рис. 12.9). Она соединяет в себе простоту и автономность. Эти преимущества позволяют установить ее на наземном транспортном средстве или на судне*.

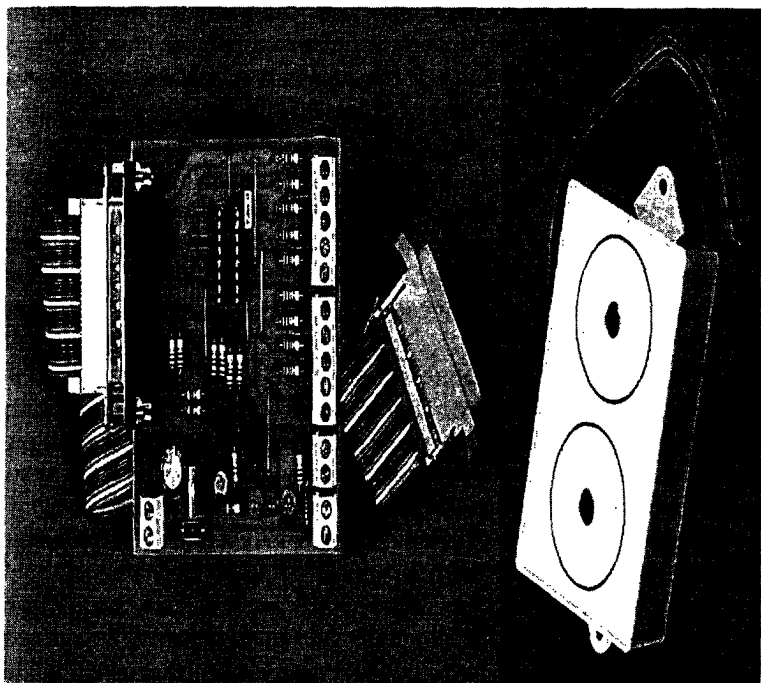


Рис. 12.9. Охранная сигнализация на основе микроконтроллера BS2 (в корпусах DIP или CMS) со звуковой сигнализацией-сиреной

Любое действие можно запрограммировать простым изменением программы, не преобразуя схему. Необходимо знать язык программирования используемого микроконтроллера. Мы будем использовать уже знакомый нам язык BASIC STAMP2. Для

* Можно даже установить эту охранную сигнализацию на восьмисотый «мерседес» нашего рядового олигарха или яхту Абрамовича. – *Прим. науч. ред.*

программирования микроконтроллера необходимо иметь компьютер, который после программирования нам не понадобится.

Принципиальная схема

На рис. 12.10 представлена принципиальная схема охранной сигнализации. Она сильно схожа с предыдущей схемой на рис. 12.6, но в ней параллельный порт РС заменен микроконтроллером BASIC STAMP2.

Электроника датчиков непосредственного контроля и датчиков с задержкой по времени осталась той же самой. Упрощение заключается в том, что каждый из входов непосредственно соединяется с контактами P0 – P8 микроконтроллера BS2. Выключатель S1 управляется по каналу P8. Резистор R10 поддерживает высокое логическое состояние при разомкнутом состоянии выключателя S1. Схема управления сиреной абсолютно идентична и состоит из тех же элементов. Управление сиреной происходит по каналу P15. Канал P14 управляет светодиодом, который светится при обрыве одного из датчиков контроля.

Выводы микроконтроллера могут работать с довольно большими токами (до 25 мА). Поэтому чтобы заставить светиться светодиод, не требуется подключать светодиод через транзистор. Дополнительная кнопка RST позволяет вручную обнулить микроконтроллер. Разъем J1 является неотъемлемой частью устройства и служит для программирования микроконтроллера, не вынимая его из держателя.

Питание схемы осуществляется так же, как и на схеме с программируемой логикой. Диод D10 защищает схему от переполюсовки напряжения во время подсоединения питания +12 В. Далее напряжение +12 В фильтруется с помощью конденсатора C1 и подается на сирену. Проследите за тем, чтобы напряжение питания сирены было достаточным для уверенной работы. Напряжение 5 В для микроконтроллера образуется с помощью линейного стабилизатора CI2. Конденсатор C2 отфильтровывает переменные пульсации напряжения 5 В. Конденсатор C3 дополнительно стабилизирует напряжение питания микроконтроллера. Светодиод L3, ограниченный резистором R14, показывает наличие питания схемы.

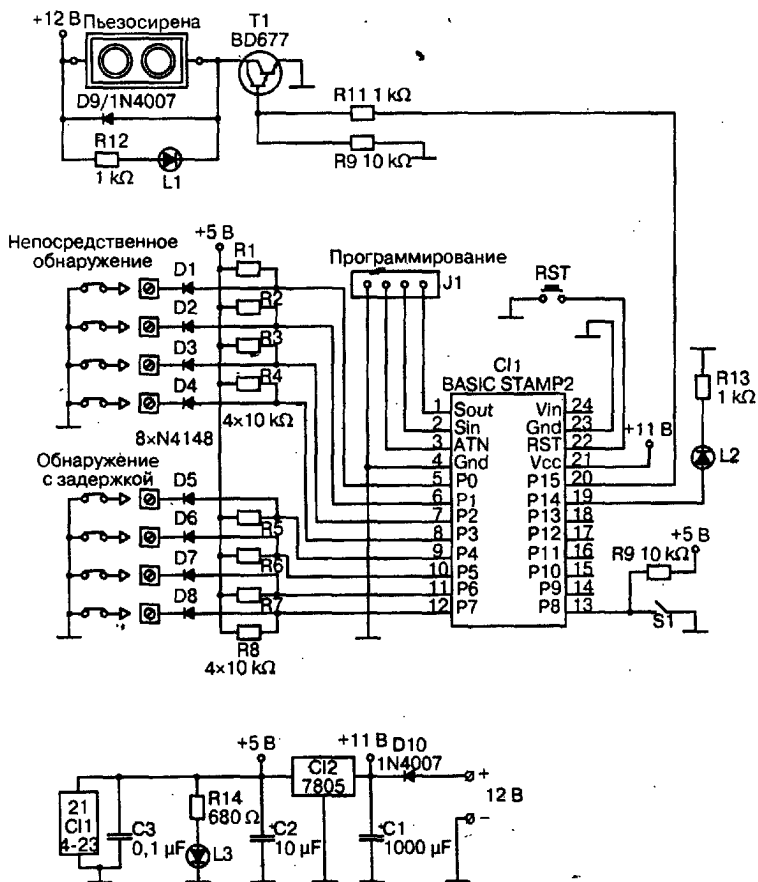


Рис. 12.10. Принципиальная схема охранной сигнализации на основе микроконтроллера BASIC STAMP2

Сборка

Рисунок печатной платы, представленный на рис. 12.11, простой и легко реализуемый. Материалы по технике литографии вы можете найти в библиографии в конце книги.

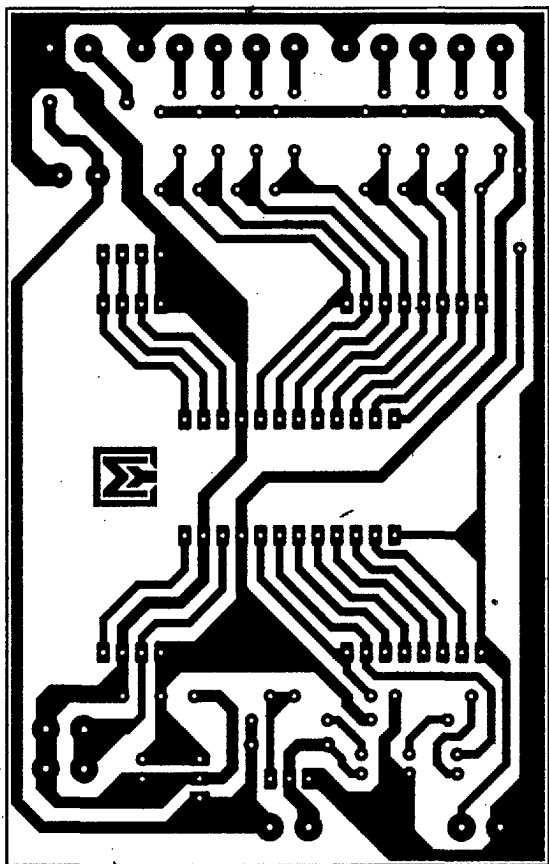


Рис. 12.11. Рисунок печатной платы охранной сигнализации на основе микроконтроллера BASIC STAMP2

На рис. 12.12 представлена монтажная плата устройства. Обратите внимание, что в этой схеме нет ни одной переключки. Детали следует запаивать на плату в очередности, определяемой размерами и прочностью элементов.

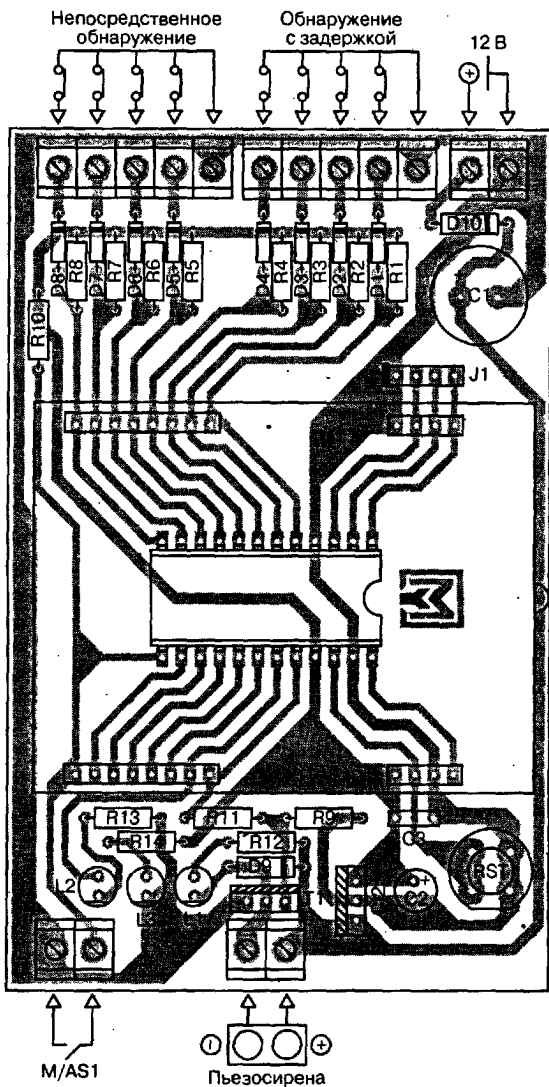


Рис. 12.12. Монтаж компонентов на плате охранной сигнализации на основе микроконтроллера BASIC STAMP2

- штырьевые разъемы;
- 1 пьезосирена с 2 или 4 пьезопластинами;
- 1 выключатель (простой или с ключом).

Подключите питание к охранной сигнализации и запрограммируйте микроконтроллер с помощью программы STAMP.EXE. Если все подсоединения произведены правильно, то через несколько минут программа будет записана на BS2. После этого вы можете выключить программатор!

Листинг 12.2. Программа станции охранной сигнализации

```
' === ПРОГРАММА УПРАВЛЕНИЯ СТАНЦИЕЙ ОХРАННОЙ СИГНАЛИЗАЦИИ ===
' === НА PBASIC2                               © Yves Mergy 2001 ===

' === ОБЪЯВЛЕНИЯ И ИНИЦИАЛИЗАЦИЯ ===
CT var      word
TSIR con    30
TENT con    10
TSOR con    10
TSIL con    60
DIRS=%1100000000000000 'P0-P13 входы, P14 и P15 выходы

' === ОЖИДАНИЕ НАРУШЕНИЯ (Замыкания S1) ===
START:
CT=0
low 14      'P14 на нижнем уровне
low 15      'P15 на нижнем уровне
if in8 = 1 then START

' === ЗАДЕРЖКА ВЫХОДА ===
OUTPUT:
if in8 = 1 then START
gosub TEST
pause 100
CT=CT+1
if CT<TSOR*10 then ВЫХОД
CT=0

' === АНАЛИЗ И ТЕСТИРОВАНИЕ ЗОН ===
SUPERVISION:
if in8 = 1 then STARTif in0 = 1 then SIREN
if in1 = 1 then SIRENif in2 = 1 then SIREN
if in3 = 1 then SIREN
if in4 = 1 then DELAY
if in5 = 1 then DELAY
if in6 = 1 then DELAY
if in7 = 1 then DELAY
goto SUPERVISION
```

```

' === ЗАДЕРЖКА ВКЛЮЧЕНИЯ СИРЕНА ===
SIREN:
if in8 = 1 then START
gosub TEST
high 15
pause 100
CT=CT+1
if CT<TSIR*10 then SIREN
CT=0
low 15

' === ЗАДЕРЖКА ТИШИНЫ ===
SILENCE:
if in8 = 1 then START
gosub TEST
high 15
pause 100
CT=CT+1
if CT<TSIL*10 then SILENCE
CT=0
goto SUPERVISION

' === ЗАДЕРЖКА ВХОДА ===
DELAY:
if in8 = 1 then START
gosub TEST
high 15
pause 100
CT=CT+1
if CT<TENT*10 then DELAY
CT=0
goto SIREN

' === ПОДПРОГРАММА ===
' === ТЕСТИРОВАНИЕ ПРИ РАЗМЫКАНИИ ХОТЯ БЫ ОДНОЙ ЗОНЫ ===
TEST:
if in1 > 0 then CONTINUATION1
low 14
goto CONTINUATION2
CONTINUATION1:
high 14
CONTINUATION2:
Return

```

Работа программы абсолютно идентична программе охранной сигнализации на основе программируемой логики. Данная схема работает автономно и не требует компьютера. Для того чтобы узнать о состоянии системы, используются светодиоды, правда, с меньшей информативностью, чем экран монитора компьютера.

Глава 13

«Умный» имитатор присутствия

13.1. Общее описание

В предыдущей главе мы изучили устройство охранной сигнализации в трех разных вариантах исполнения, один из которых осуществляется на основе элементов цифровой логики. В этой главе мы изучим более амбициозный проект «умного» имитатора присутствия. Читатель должен знать, что такое помпезное название свидетельствует лишь о том, что данное устройство будет иметь свойство запоминания. На самом деле, это устройство способно запоминать с псевдослучайной частотой от одной до семи минут ваши действия на 4 прерывателя и затем воспроизводить их с той же частотой. Таким образом, можно периодически напоминать о своем присутствии в течение дня или недели*.

Из материала этой главы можно легко понять, что этот проект нельзя больше рассматривать на основе только логических элементов. Мы предложим вашему вниманию схему на основе программируемой логики и схему на основе нашего знаменитого микроконтроллера BASIC STAMP2.

* Подобного рода имитатор присутствия используется для защиты жилища на период отпуска хозяев дома или отсутствия по другим причинам. Воры вычисляют квартиры, где постоянно выключен свет, и грабят их. В квартирах, оборудованных такими устройствами охранной сигнализации, вор обманывается, считая, что дома кто-то есть, в такую квартиру не заходит. Таким образом, охранная сигнализация по-своему выполняет ту же задачу, что и описанные ранее в этой книге системы охранной сигнализации. — Прим. науч. ред.

13.2. Схема на основе программируемой логики

Применение данной схемы оправдывается значительным объемом свободной памяти компьютера и легкостью связи с внешними устройствами. Кроме того, параллельный порт компьютера обладает необходимым количеством каналов для нашего устройства. Из-за наличия компьютера охранная система не является автономной, так как не содержит устройства памяти. Память находится в ПК.

Принципиальная схема

Простота принципиальной схемы на рис. 13.1 обусловлена тем, что основная работа выполняется компьютером. Каждый датчик соединен непосредственно с каналом параллельного порта.

Состояние всех выключателей SZ1–SZ4 периодически проверяется, прерыватели соединены соответственно с входами ERROR, SELECT, PAPER END и ACK параллельного порта. «Умный» имитатор присутствия на основе программируемой логики приведен на рис. 13.2. Чертеж печатной платы имитатора присутствия приведен на рис. 13.3, а монтажная схема имитатора – на рис. 13.4. Когда контакты выключателей в верхнем положении по схеме, они не подключены к «земле».

Резистивная сборка RES1 переводит все входы на высокий уровень (+5 В). При переключении в нижнее положение любого одного из ключей соответствующий вход переходит на низкий уровень, соединяясь с «землей». Выключатель SPL, предназначенный для выбора режима работы: программирование или функционирование охранной системы, включен по такому же принципу, как и выключатели SZ1–SZ4. Он также использует один из резисторов сборки RES1.

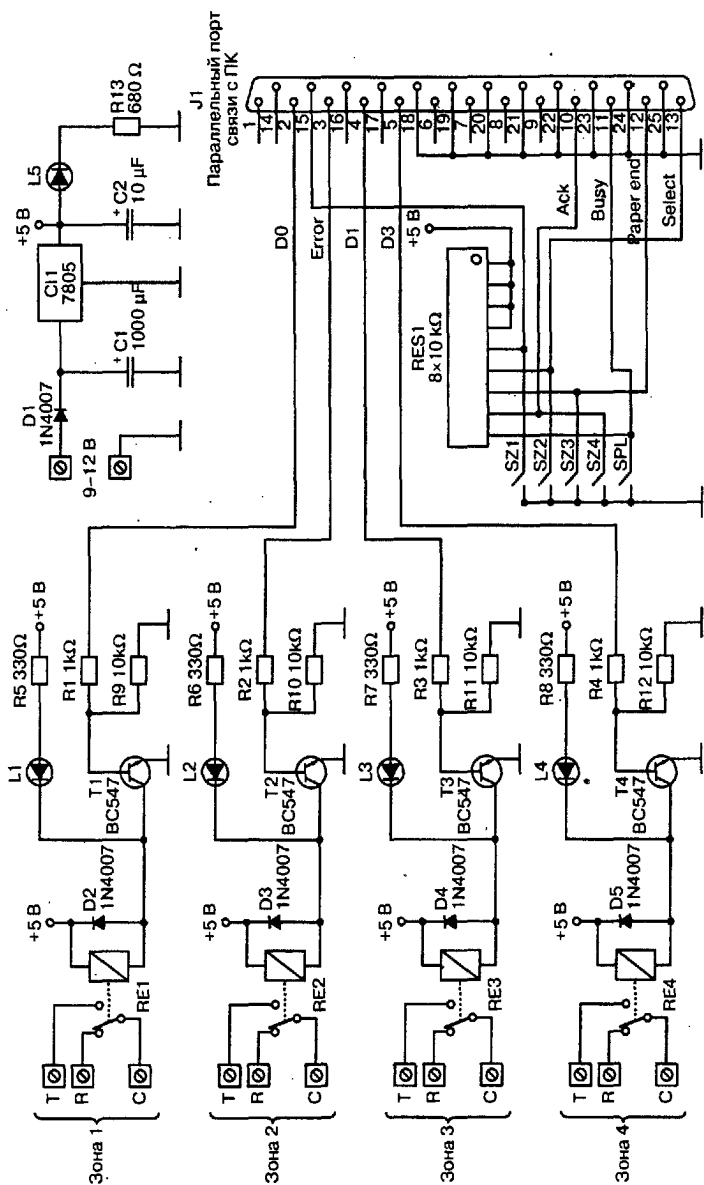


Рис. 13.1. Принципиальная схема имитатора присутствия на основе программируемой логики

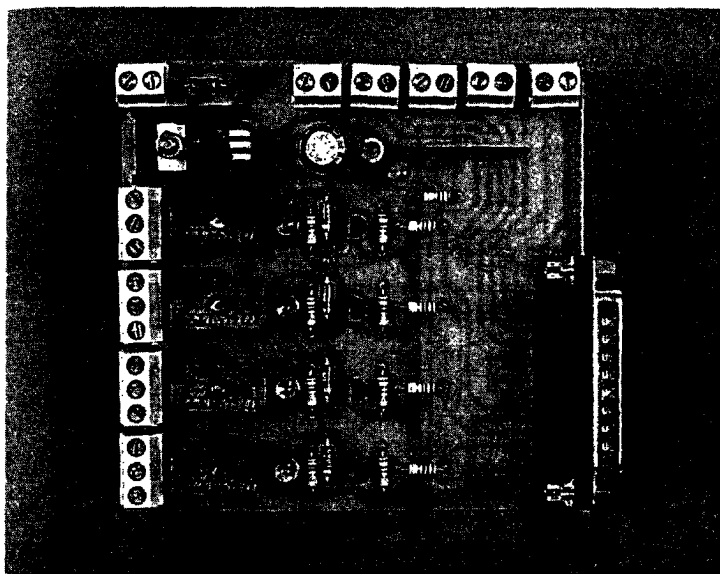


Рис. 13.2. «Умный» имитатор присутствия на основе программируемой логики

Список компонентов:

Резисторы с 5-процентным допуском:

- R1–R4: 1 кОм (коричневый, черный, красный);
- R5–R8: 330 Ом (оранжевый, оранжевый, коричневый);
- R9–R12: 10 кОм (коричневый, черный, оранжевый);
- R13: 680 Ом (синий, серый, коричневый);
- RES1: резистивная сборка 8r10 кОм.

Конденсаторы:

- C1: 1000 мкФ / 25 В (полярный танталовый или электролитический с радиальными выводами);
- C2: 10 мкФ / 25 В (полярный танталовый или электролитический с радиальными выводами).

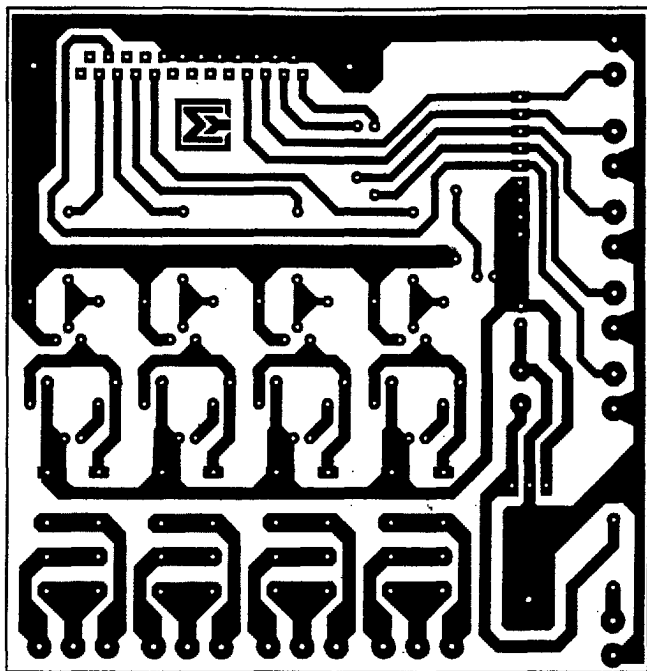


Рис. 13.3. Чертеж печатной платы имитатора присутствия на основе программируемой логики

Полупроводники:

- D1-D5: 1N4007;
- L1-L5: светодиод 5 мм (любой цвет на выбор);
- T1-T4: BC547;
- C11:7805.

Прочие элементы:

- штырьевые разъемы с шагом 5,08;
- 1 угловой разъем DB25 для печатной платы;
- 1 плоский радиатор для TO220;
- RE1-RE4: реле на 5 В в корпусе DIP.

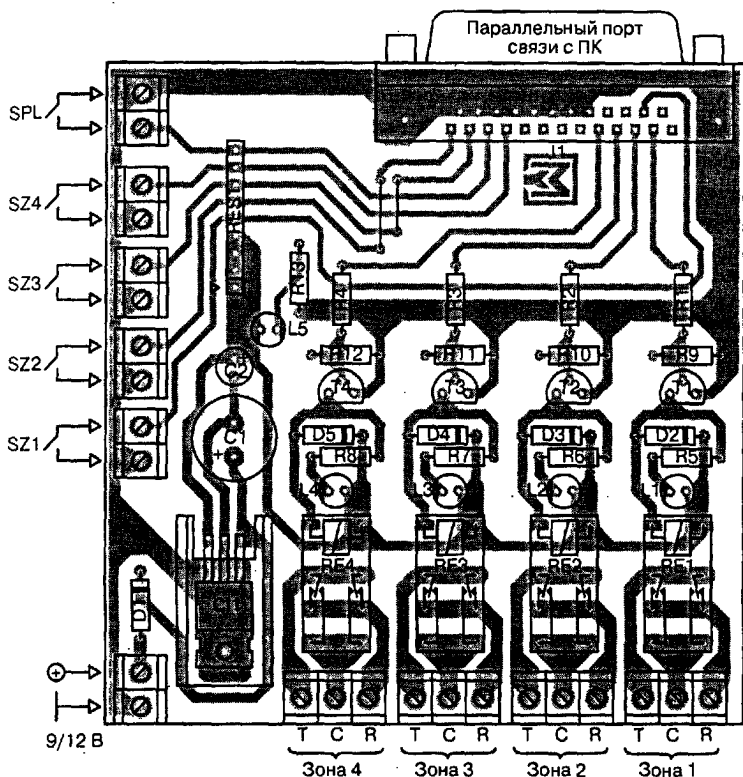


Рис. 13.4. Монтажная схема имитатора присутствия на основе программируемой логики

Четыре схемы управления реле идентичны и управляются по каналам D0 – D3 параллельного порта. Изучим работу одного из них. Сигнал от ПК по каналу D0 подается на базу транзистора T1 через резистор в цепи базы R1. Если уровень сигнала низкий, транзистор остается в закрытом состоянии. Резистор R9 запирает транзистор при отключении схемы от компьютера. Если уровень сигнала будет высоким, транзистор T1 открывается, и питание подается к реле. Диод D2 защищает транзистор от токов самоиндукции, образующихся в реле при переключении. Светодиод L1, ток которого ограничивается резистором R5, светится при подаче напряжения питания на реле.

Схема питания устройства традиционная и не обладает оригинальностью. Диод D1 защищает схему от неправильной полярности (переполюсовки) напряжения питания. Конденсатор C1 осуществляет первичную фильтрацию питания. Линейный стабилизатор С11 стабилизирует напряжение 5 В, это напряжение необходимо для работы параллельного порта. Конденсатор C2 осуществляет окончательную фильтрацию. Светодиод L5, ток которого ограничен резистором R13, показывает наличие напряжения питания в схеме.

Сборка

Рисунок печатной платы представлен ранее на рис. 13.2. Единственной трудностью при разводке печатной платы было близкое расположение выводов разъема DB25 между собой. Материалы по технике литографии вы можете найти в библиографии в конце книги.

На рис. 13.3 представлена монтажная плата устройства. В этой схеме имеются две короткие перемычки, их следует запаивать в первую очередь. Элементы следует запаивать на плату в очередности, определяемой размерами и степенью их прочности. Начните с пайки резисторов, затем запаяйте диоды, резистивную сборку, светодиоды, транзисторы, штырьевые разъемы, разъем DB25, реле, полярные танталовые или электролитические конденсаторы, и, наконец, линейный стабилизатор напряжения, который следует обязательно установить на радиатор.

Перед тем как подключить питание к схеме, с помощью лупы визуально проверьте качество сборки платы и пайку элементов. Некоторые полярные элементы (электролитические конденсаторы) и диоды нельзя запаивать в неправильной полярности и схема, как минимум, не будет работать, а как максимум – конденсаторы в таком нежелательном включении при подаче питания могут взрываться и летать, возможно, причинив неприятности хозяину схемы. А нужно только внимательно проверить номиналы элементов, полярность диодов и конденсаторов и не перепутать между собой выводы транзисторов.

Данная схема предназначена для подсоединения к компьютеру. Перед тем как подсоединить схему, проверьте присутствие

питающих напряжений на всех участках печатной платы. Подсоединяйте плату к компьютеру, только если вы убедились в правильности номинала и полярности всех тестовых напряжений.

Программирование

Как и любое устройство, работающее с компьютером, имитатор присутствия, который вы только что изготовили, не может работать без соответствующей программы. Для написания программы можно использовать простой и доступный язык программирования QBASIC. Листинг программы 13.1 (SIMUPRES.BAS) наглядно свидетельствует об этом.

Листинг 13.1. Программа имитатора присутствия на базе ПК.

```

REM === ПРОГРАММА ИМИТАТОРА ПРИСУТСТВИЯ НА БАЗЕ ПК НА ЯЗЫКЕ
QBASIC ===
REM === Для параллельного порта LPT1. © Yves MERGY 2001 ===

' === ОБЪЯВЛЕНИЕ ПЕРЕМЕННЫХ И КОНТАНТ ===
DIM TP(2000) AS INTEGER
DIM EN(10) AS INTEGER
DIM S0(4) AS INTEGER
T0S = "ЗАМКНУТЫЙ": T1S = "РАЗОМКНУТЫЙ"
T2S = "ЗАПИТАН": T3S = "ВЫКЛЮЧЕН"
ТЕМПО = 60'60 => 1 минута 420 => 7 минут
МАХ1 = 1400'Количество минут / день (или 7 минут / неделя)

' === ИНИЦИАЛИЗАЦИЯ ===
СТ = 0: ОУТ (888), 0
FOR I = 0 TO МАХ1: TP(I) = 0: NEXT I
FOR I = 0 TO 3: S0(I) = 0: NEXT I

' === ОТОБРАЖЕНИЕ НА ГЛАВНОМ ЭКРАНЕ ===
SCREEN 0, 0, 0
LOCATE 4, 22: PRINT "ИМИТАТОР ПРИСУТСТВИЯ НА ПК"
LOCATE 6, 18: PRINT "Зона 1 : РАЗОМКНУТА ==> Реле 1 :
ВЫКЛЮЧЕНО"
LOCATE 7, 18: PRINT "Зона 2 : РАЗОМКНУТА ==> Реле 2 :
ВЫКЛЮЧЕНО"
LOCATE 8, 18: PRINT "Зона 3 : РАЗОМКНУТА ==> Реле 3 :
ВЫКЛЮЧЕНО"
LOCATE 9, 18: PRINT "Зона 4 : РАЗОМКНУТА ==> Реле 4 :
ВЫКЛЮЧЕНО"
    
```



```

LOCATE 11, 26: PRINT "ПЕРЕКЛЮЧАТЕЛЬ ==> 1"
LOCATE 12, 26: PRINT "СЧЕТЧИК ==> 0"
LOCATE 13, 26: PRINT "ВРЕМЯ ==> 0.0"
LOCATE 14, 26: PRINT "ЗНАЧЕНИЕ ==> 0"

' === ВЫБОР РЕЖИМА (ВОЗВРАТ ИЛИ ПРОГРАММИРОВАНИЕ) ===
START:
GOSUB READ OF ZONES
LOCATE 11, 39: PRINT EN(4)
IF EN(4) = 0 THEN GOSUB READ1 ELSE GOSUB PROGRAMMING1

' === РЕЖИМ ПРОГРАММИРОВАНИЯ ===
PROGRAMMING1:
CT = 0: OUT (888), 0
LOCATE 16, 10
PRINT "Имитатор в режиме программирования, действия "
LOCATE 17, 10
PRINT "на 4 зонах будут записываться каждую минуту. "
LOCATE 19, 10
PRINT "Нажмите на клавишу пробела для подтверждения или
измените"
LOCATE 20, 10
PRINT " положение SPL и нажмите на любую другую клавишу.
"
GOSUB KEY
IF AS <> CHR$(32) THEN GOTO START
PROGRAMMING2:
IF CT = MAXI + 1 THEN GOTO WRITE
GOSUB CANCELLATION
LOCATE 17, 23: "РЕЖИМ ПРОГРАММИРОВАНИЯ"
GOSUB READ OF ZONES
LOCATE 11, 39: PRINT EN(4)
IF EN(4) = 0 THEN TP(CT) = 255: GOTO WRITE
LOCATE 12, 39: PRINT CT
TIMER ON
START = TIMER
DO
  TEP = TIMER - START
  IF TEP > TEMPO THEN EXIT DO
  LOCATE 13, 39: PRINT USING "##.#"; TEP
  GOSUB READ OF ZONES
  FOR I = 0 TO 3
    IF EN(I) <> EN(1 + 5) THEN
      LOCATE 6 + I, 27
      IF EN(I) = 0 THEN PRINT T0$ ELSE PRINT T1$
      LOCATE 6 + 1, 50
      IF EN(I) = 0 THEN PRINT T2$ ELSE PRINT T3$
      EN(I + 5) = EN(I)

```

```

    ЗНАЧЕНИЕ = EN(0) - EN(1) + EN(2) EN(3)
    OUT (888), MEANING
END IF
NEXT I
LOOP
TIMER OFF
TP(CT) = MEANING
LOCATE 14, 39: PRINT MEANING
CT = CT + 1
SOUND 1000, 3
GOTO PROGRAMMING2
' === СОЗДАНИЕ И ЗАПИСЬ В ФАЙЛ (ДАННЫЕ.TXT) ===
WRITE:
OUT (888), 0
FOR I = 100 TO 2000 STEP 100: SOUND I, 1: NEXT I
LOCATE 16, 10
PRINT "Имитатор создаст файл ДАННЫЕ.TXT в "
LOCATE 17, 10
PRINT "директории C:\. Он будет содержать действия,
выполненные за день."
LOCATE 19, 10
PRINT "Нажмите на клавишу пробела для подтверждения или на
любую "
LOCATE 20, 10
PRINT "другую клавишу для возврата в начало режима
программирования."
GOSUB KEY
IF AS <> CHR$(32) THEN GOTO PROGRAMMING1
OPEN "o", #1, "c:\данные.txt"
FOR I = 0 TO MAXI: PRINT #1, TP(1): NEXT I
CLOSE #1
GOTO START

' === РЕЖИМ ВОЗВРАТА ===
READ1:
CT = 0: OUT (888), 0
LOCATE 16, 10
PRINT "Имитатор находится в режиме возврата, 4 реле "
LOCATE 17, 10
PRINT "вернуть записанное состояние на входы в режиме
реального времени "
LOCATE 19, 10
PRINT "Нажмите на клавишу пробела для подтверждения или
измените"
LOCATE 20, 10
PRINT " положение SPL и нажмите на любую другую клавишу.
"
GOSUB KEY

```

```

IF A$ <> CHR$(32) THEN GOTO START
GOSUB READ
ЧТЕН2:
GOSUB CANCELLATION
LOCATE 17, 23: PRINT "РЕЖИМА ВОЗВРАТА"
GOSUB READ OF ZONES
LOCATE 11, 39: PRINT EN(4)
IF EN(4) = 1 THEN GOTO PROGRAMMING1
MEANING = TP(CT)
OUT (888), MEANING
LOCATE 14, 39: PRINT MEANING
FOR I = 3 TO 0 STEP -1
  IF MEANING - (2 ^ I) < 0 THEN
    S0(I) = 0
    LOCATE 6 + I, 27: PRINT T1$
    LOCATE 6 + I, 50: PRINT T3$
  ELSE
    S0(I) = 2 ^ I
    LOCATE 6 + I, 27: PRINT T0$
    LOCATE 6 + I, 50: PRINT T2$
  END IF
  MEANING = MEANING - S0(I)
NEXT I
LOCATE 12, 39: PRINT CT
TIMER ON
START = TIMER
DO
  TEL = TIMER - START
  IF TEL > TEMPO THEN EXIT DO
  LOCATE 13, 39: PRINT USING "##.##"; TEL
LOOP
TIMER OFF
CT = CT + 1
IF TP(CT) = 255 THEN CT = 0; SOUND 1000, 8
SOUND 1000, 3
GOTO READ2

'=====>> ПОДПРОГРАММЫ <<=====
'=== ТЕСТИРОВАНИЕ СОСТОЯНИЯ ПЕРЕКЛЮЧАТЕЛЕЙ ЗОНЫ ===
READ OF ZONES:
A = INP (889)
IF (A AND 8) = 8 THEN EN(0) = 0 ELSE EN(0) = 1
'(15) ошибка
IF (A AND 16) = 16 THEN EN(1) = 0 ELSE EN(0) = 2
'(13) выбор
IF (A AND 32) = 32 THEN EN(2) = 0 ELSE EN(0) = 4

```

```

'(12) конец бумаги
IF (A AND 64) = 64 THEN EN(3) = 0 ELSE EN(0) = 8
'(10) подтверждение
IF (A AND 128) = 128 THEN EN(4) = 0 ELSE EN(0) = 1
'(11) занято
RETURN

' === ЧТЕНИЕ ДАННЫХ ФАЙЛА (ДАННЫЕ.TXT) ===
READ:
FOR I = 2000 TO 100 STEP -100: SOUND I, 1: NEXT I
OPEN "i", #1, "c:\данные.txt"
FOR I = 0 TO MAXI: INPUT #1, TP(I): NEXT I
CLOSE #1
RETURN

' === ОЖИДАНИЕ НАЖАТИЯ КЛАВИШИ КЛАВИАТУРЫ ===
KEY:
AS = INKEY$
IF AS = ** THEN GOTO KEY
RETURN

' === ЧАСТИЧНОЕ СТИРАНИЕ ЭКРАНА ===
CANCELLATION:
LOCATE 16, 10: PRINT SPC(58);
LOCATE 17, 10: PRINT SPC(58);
LOCATE 19, 10: PRINT SPC(58);
LOCATE 20, 10: PRINT SPC(58);
RETURN

```

При запуске программа выводит на экран основную таблицу визуализации и анализирует состояние выключателя SPL, чтобы перейти в режим программирования или функционирования (восстановления).

В режиме программирования идет анализ состояния четырех выключателей путем сканирования их с определенной частотой. В конце работы создается файл, который содержит всю информацию. Затем программа снова анализирует состояние выключателя SPL.

В режиме восстановления программа считывает данные, записанные ранее, и выводит их на четыре выключателя с той же частотой.

На экран компьютера выводятся в реальном масштабе времени сообщения о состояниях всех реле.

13.3. Схема на основе микроконтроллера BS2

На рис. 13.5 приведена схема микроконтроллера в корпусе CMS, также возможен вариант микроконтроллера в корпусе DIP. Применение микроконтроллера в схеме с таким сложным алгоритмом работы – включением-выключением по псевдослучайному закону полностью оправдывает себя. Схема с микроконтроллером полностью автономна в отличие от схемы с применением программирования на PC. Схема проста в исполнении, так как микроконтроллер обладает высокой степенью интеграции элементов. Память используемого микроконтроллера BS2 имеет высокую производительность. В память помещается не только программа, но и энергонезависимые данные.

Микроконтроллер BS2
показан на рис. 13.5
в корпусе CMS,
но возможна его
установка в корпусе DIP

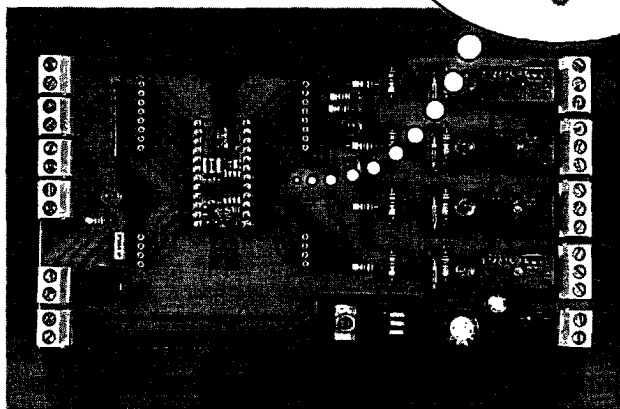
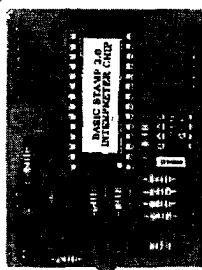


Рис 13.5. Схема имитатора присутствия на основе микроконтроллера

Принципиальная схема

Схема, приведенная на рис. 13.6, очень похожа на принципиальную схему из предыдущего раздела на рис. 13.1.

Производительный микроконтроллер BS2 полностью заменяет компьютер, выполняя все математические операции. Остается только осуществить связь между ним и остальными элементами. Каждый функциональный элемент схемы соединяется с определенным выводом микроконтроллера, который работает на вход или на выход в зависимости от требований. Состояние четырех выключателей SZ1-SZ4 периодически считывается по каналам P8-P11. Выключатель SPL, предназначенный для выбора режима работы, связан с каналом P15. Так как данное устройство не работает с ПК и соответственно не имеет клавиатуру ПК, требуется добавить в схему кнопку VALID, обеспечивающую запуск устройства. Эта кнопка связана с микроконтроллером по каналу P14. Резистивная сборка RES1 поддерживает высокий уровень сигнала на перечисленных выводах микроконтроллера (каналы P8-P11, P14, P15), если кнопки не нажаты.

Так как в данном варианте без ПК отсутствует экран монитора, его заменяют два светодиода L5 и L6, подсоединенные к P5 и P6. Резисторы R14 и R15 ограничивают ток этих светодиодов. Для удобства в схеме применяется маломощная пьезосирена, управляемая по каналу P7 непосредственно с выхода микроконтроллера.

Четыре схемы управления реле RE1-RE4 идентичны и управляются по каналам P0-P3. Мы не будем снова описывать работу этих схем и схему питания, так как они абсолютно идентичны предыдущим схемам (рис. 13.1). Заметим только, что в схеме питания присутствует дополнительно конденсатор C3, осуществляющий дополнительно развязку по питанию микроконтроллера BASIC STAMP 2.

Разъем J1 позволяет производить программирование микроконтроллера прямо на плате, не вынимая из держателя, как и в устройстве охранной сигнализации. Дополнительная кнопка RST производит ручной сброс устройства.

Сборка

Рисунок печатной платы представлен на рис. 13.7. Его монтажная схема дана на рис. 13.8. Изготовление платы не представляет никаких трудностей. Материалы по технике литографии вы можете найти в библиографии в конце книги.

Список компонентов:

Резисторы с 5-процентным допуском:

- R1–R4: 1 кОм (коричневый, черный, красный);
- R5–R8: 330 Ом (оранжевый, оранжевый, коричневый)
- R9–R12: 10 кОм (коричневый, черный, оранжевый);
- R13: 680 Ом (синий, серый, коричневый);
- RES1: резистивная сборка 8г10 кОм.

Конденсаторы:

- C1: 1000 мкФ/ 25 В (полярный танталовый или электролитический с радиальными выводами);
- C2: 10 мкФ/ 25 В (полярный танталовый или электролитический с радиальными выводами);
- C3: 0,1 мкФ, неполярный керамический.

Полупроводники:

- D1–D5: 1N4007;
- L1–L7: светодиод 5 мм (любой цвет на выбор);
- T1–T4: BC547;

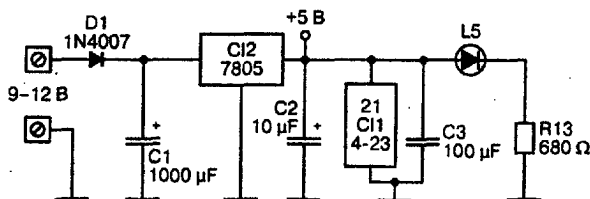


Рис. 13.6. Принципиальная схема имитатора присутствия на основе микроконтроллера BS2

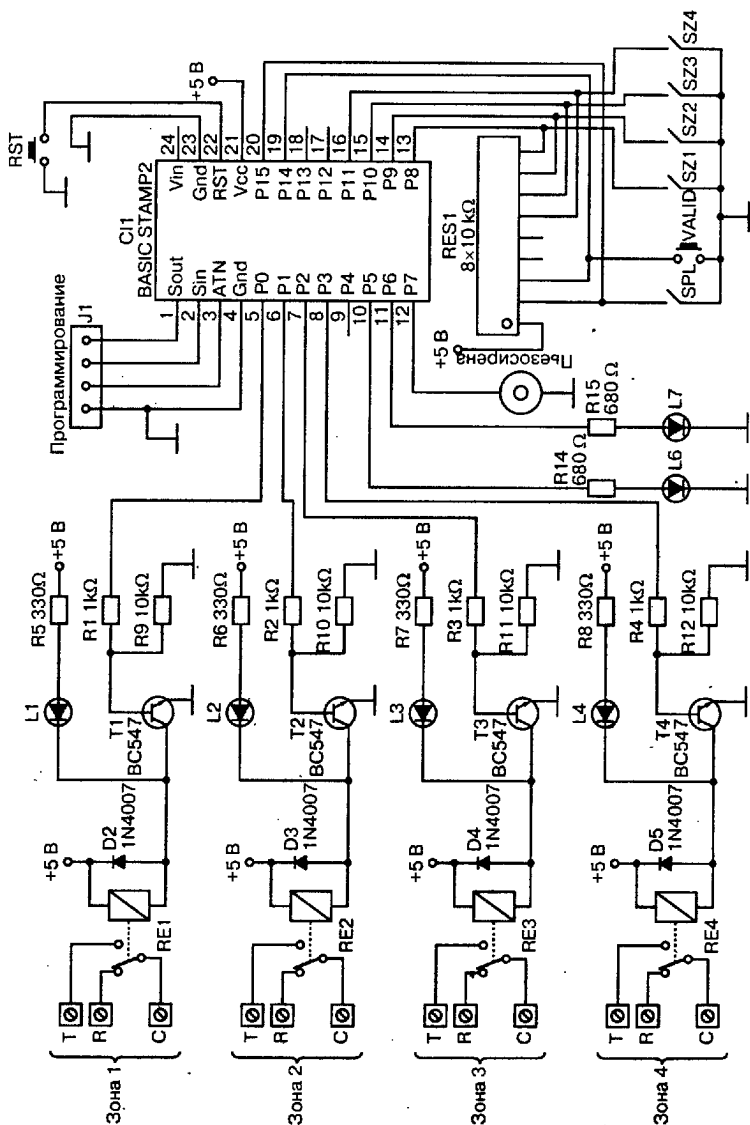


Рис. 13.7. (Продолжение) Принципиальная схема имитатора присутствия на основе микроконтроллера BS2

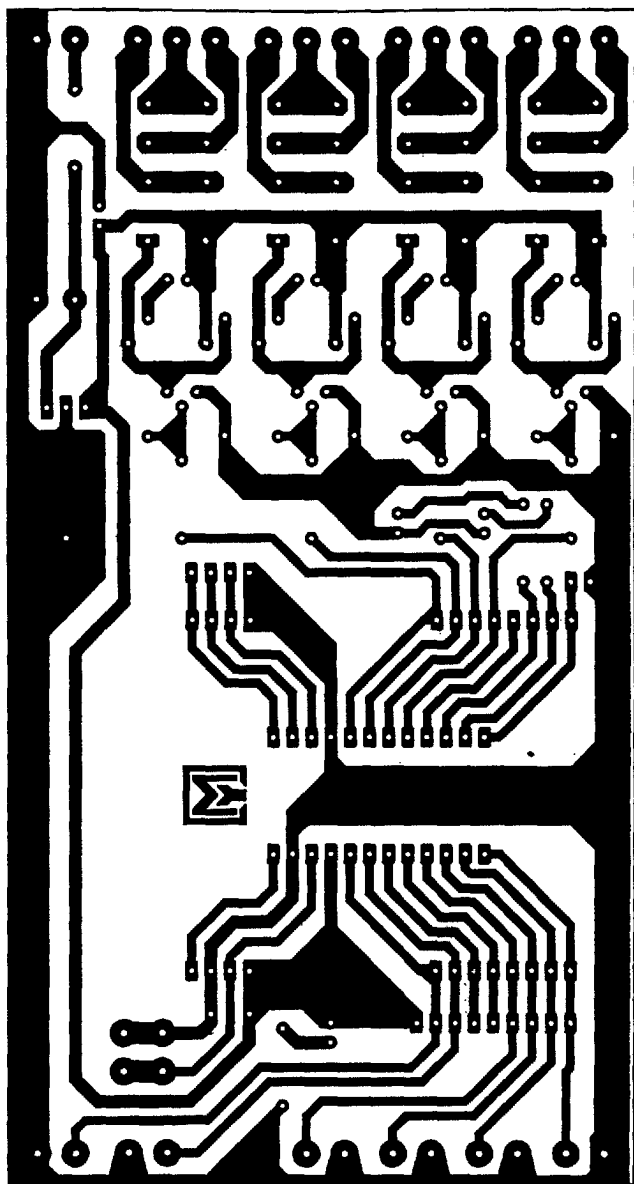


Рис. 13.7. Чертеж печатной платы имитатора присутствия на основе микроконтроллера

- CI1: BASIC STAMP 2 (модуль в корпусе CMS или исполнение в корпусе DIP);
- CI2: 7805.

Прочие элементы:

- штырьевые разъемы с шагом 5,08;
- 1 плоский горизонтальный радиатор для TO220;
- RE1-RE4: реле на 5 В в корпусе DIP;
- 1 держатель микросхемы на 24 вывода;
- 24 выводных разъемов для подключения платы микроконтроллера в корпусе DIP;
- 4 выводных разъемов для подключения кабеля программирования;
- 1 кнопка;
- 1 пьезосирена.

На рис. 13.8 представлена монтажная плата устройства. Используйте эту схему для сборки устройства. В этой схеме нет ни одной перемычки. Детали следует запаивать на плату в очередности, определяемой размерами и степенью прочности элементов. Начните с пайки резисторов, затем запаяйте диоды, резисторную сборку, держатель микроконтроллера на 24 вывода, 24 выводных разъема для подключения платы микроконтроллера в корпусе DIP, керамический конденсатор, кнопку RST, светодиоды, транзисторы, штырьевые разъемы, реле, полярные танталовые или электролитические конденсаторы, и, наконец, линейный стабилизатор напряжения, который следует обязательно установить на радиатор.

Перед тем как подключить питание к схеме, визуально проверьте качество сборки платы и пайку элементов. Как уже неоднократно замечалось, что некоторые элементы имеют полярность, и их нельзя запаивать в неправильной полярности. Внимательно проверьте номиналы элементов.

Не вставляя микроконтроллер в держатель «кроватьку», подключите питание к схеме и проверьте наличие напряжений на держателе в соответствии со схемой. После этого вставьте микроконтроллер BASIC STAMP 2 в держатель при выключенном питании. Теперь вы можете перейти к программированию устройства для получения полноценных характеристик этого устройства защиты от воров.

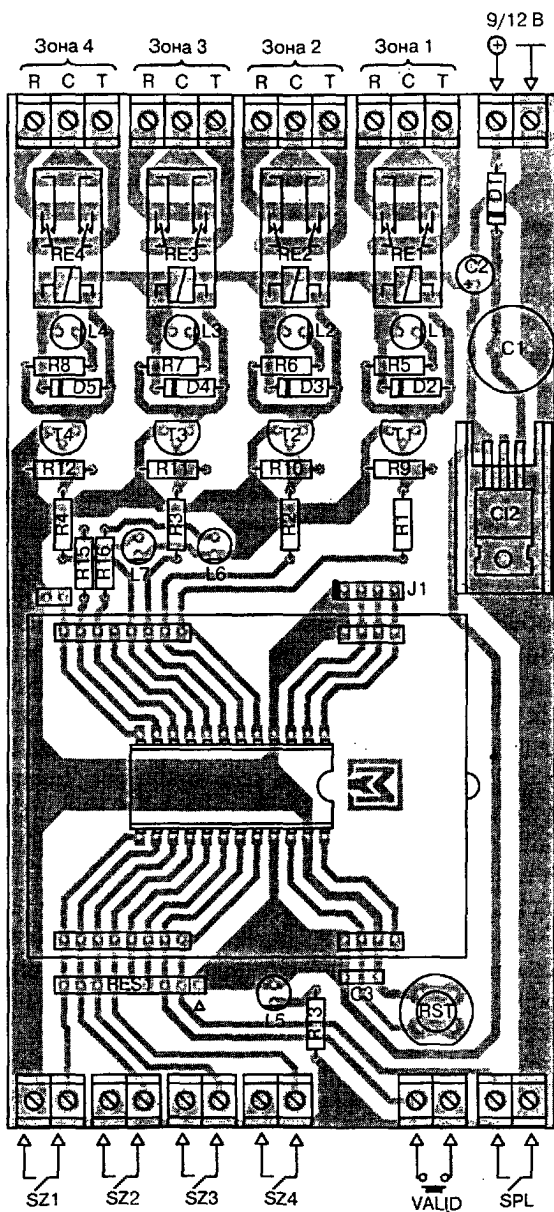


Рис. 13.8. Монтажная схема имитатора присутствия на основе микроконтроллера

Программирование

Методика программирования микроконтроллеров BASIC STAMP 2 была описана во второй части этой книги, в разделе «Программирование» параграфа 12.3. Аналогично описанному в этом параграфе, напишите или скачайте с нашего сайта программу имитатора присутствия на языке программирования PBASIC2. Листинг программы 13.2 (SIMUPRES.BS2) представлен ниже.

Подключите питание к устройству, вставив микроконтроллер BS2 в держатель микросхемы. Проведите программирование устройства с помощью программы STAMPW.EXE. Если все в порядке, то программирование произойдет в течение нескольких минут. Теперь вы можете выключить программатор и компьютер!

Листинг 13.2. Программа имитатора присутствия на основе микроконтроллера BS2

```
' === ПРОГРАММА УПРАВЛЕНИЯ ИМИТАТОРА ПРИСУТСТВИЯ ===
' === НА ЯЗЫКЕ PBASIC2 © Yves MERGY 2001 ===

' === ОБЪЯВЛЕНИЯ Т ИНИЦИАЛИЗАЦИЯ ===
CT var word
SEQ var word
ADM var word
REL var byte
Hz var word
DP15 var bit
TEMPO con 60 '60 для 1 минуты; 420 для
'7 минут
MAXAD con 1440 '1440 мин / день или
'1440x7 мин / неделя
DIRS=%0000000011111111 'P0-P7 - выходы, P8 и p15
'- входы
Data 1,2,4,8,0,1,3,7,15,14,12,8,0,15,255 ' Пример
'последовательности

START:
CT=0 : out1=0 : DP15=in15
gosub SOUND1

' === РЕЖИМ ПРОГРАММИРОВАНИЯ ===
PROGRAMMING1:
ADM=0 : CT=0 : outa=0 : low 6 : toggle
if in15=DP15 then PROGRAMMING2
DP15=in15
gosub SOUND1
PROG2:
```

```

if in15=1 then REST1
if in14=0 then PROGRAMMING3
pause 3
CT=CT+1
if CT<100 then PROGRAMMING2
goto PROGRAMMING1
PROGRAMMING3:
high 5
if in15=1 or ADM=MAXAD then PROGRAMMING4
outa=inc
pause 100
SEQ=SEQ-1
if SEQ<TEMPO*10 then PROGRAMMING3
SEQ=0 : write ADM, inc : ADM=ADM-1
freqout 7,200,1000
goto PROGRAMMING3
PROGRAMMING4:
write ADM,255 : ADM=0 : SEQ=0
gosub SOUND1

```

```

' === РЕЖИМ ВОЗВРАТА ===
REST1:
CT=0 : low 5 : toggle 6
if in15=DP15 then REST2
DP15=in15
Gosub SOUND1
REST2:
if in15=0 then PROGRAMMING1
if in14=0 then PROGRAMMING2
pause 3
CT=CT-1
if CT<100 then REST2
goto REST1
REST3:
high 6
if in15=0 then PROGRAMMING1
read ADM,REL
if ADM=MAXAD or REL=255 then REST4
outa=REL
pause 100
SEQ=SEQ-1
if SEQ<TEMPO*10 then REST3
SEQ=0 : ADM=ADM+1
freqout 7,200,1000
goto REST3
REST4:
ADM=0 : SEQ=0
gosub SOUND1
goto REST3

```

```
' === ЗВУКОВАЯ ПОДПРОГРАММА ===
SOUND1:
for Hz=1 to 2000 step 100
freqout 7,50,Hz
next
return
```

Сначала опишем назначение двух светодиодов L6 и L7. Первый служит для индикации режима программирования, а второй – для воспроизведения данных. Одновременно оба светодиода светиться не могут из-за невозможности одновременной работы обоих режимов. Независимо от режима работы сначала светодиод мигает, а уже потом светится непрерывно. Программа постоянно анализирует состояние выключателя SPL, чтобы перейти в режим программирования или восстановления данных по требованию пользователя.

Сразу после включения устройство находится в режиме восстановления данных и для начала работы программа ожидает нажатие на кнопку согласия VALID.

В режиме программирования идет анализ состояния четырех выключателей с определенной частотой (частота определяется значением константы ТЕМПО в программе). В конце работы создается файл, который содержит всю информацию и записывается в устройство памяти EERPOM.

В режиме восстановления данных программа считывает данные из памяти, записанные ранее, и воспроизводит их на четырех выключателях с той же частотой.

Пьезосирена сигнализирует об окончании каждого интервала времени негромким сигналом «бип», а изменение режима работы короткой мелодией.

Глава 14

Мобильный автономный робот

14.1. Общее описание

Мы закончим эту книгу изучением самого сложного проекта — проекта робота (рис. 14.1). Полная имитация движения робота как живого существа продолжает удивлять, несмотря на то что их работа в автономном режиме прекрасно налажена. Чтобы быть автономным, наш робот не должен зависеть от компьютера. Устройство робота нельзя осуществить на основе элементов с проводной связью. Этот вариант проявляет в работе плохую гибкость, особенно если нужно сохранить данные. Поэтому возникает интерес и необходимость установки и использования микроконтроллера в «теле» робота. Он великолепно справляется с поставленной задачей по управлению роботом и является «мозгом» робота. Электрическая схема не перегружена элементами и позволяет легко проводить программирование. Также схема предусматривает дальнейшее усовершенствование, и время от времени возникает желание со стороны пытливых радиолюбителей улучшить данное устройство.

Механическая часть устройства упрощена оригинальным способом. В устройстве отсутствуют металлические части, чтобы робот казался более привлекательным и во избежание скучного использования. Шасси оригинально и изготовлено из печатных плат, которые изготавливаются вручную. Такое решение соединяет в себе надежность, гибкость и легкость изготовления. Передвижение робота осуществляется с помощью

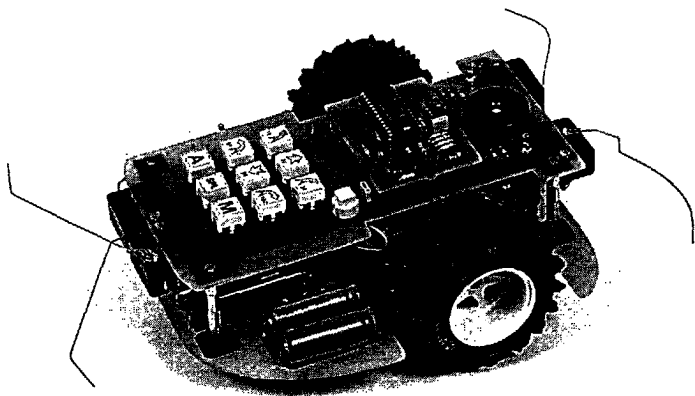


Рис. 14.1. Фотография мобильного автономного робота, который вам понравится своими техническими возможностями

двух сервомеханизмов, которые изготавливаются вручную по описанию на цветной вкладке (далее рис. 14.9–14.16). Сервомеханизмы включают в себя двигатели, редуктор с понижающими шестернями и электронную схему управления, включающую блок широтно-импульсной модуляции (ШИМ). Перейдем к более подробному описанию устройства.

14.2. Сборка робота

Принципиальная схема

Как было сообщено выше, принципиальная схема на рис. 14.2 не загромождена сложными элементами. Конструктивная реализация осуществляется на двух отдельных платах (см. рис. 14.1 и 14.3), однако принципиальная схема не содержит разъемов J2 и J3.

Микроконтроллер BS2 можно изготовить как в корпусе DIP, так и в корпусе CMS, как показано на рис. 14.3.

Нижняя плата малых размеров (см. рис. 14.4 а) содержит схему питания и сервомеханизмы; на верхней большой плате расположен весь блок управления устройством (рис. 14.4 б).

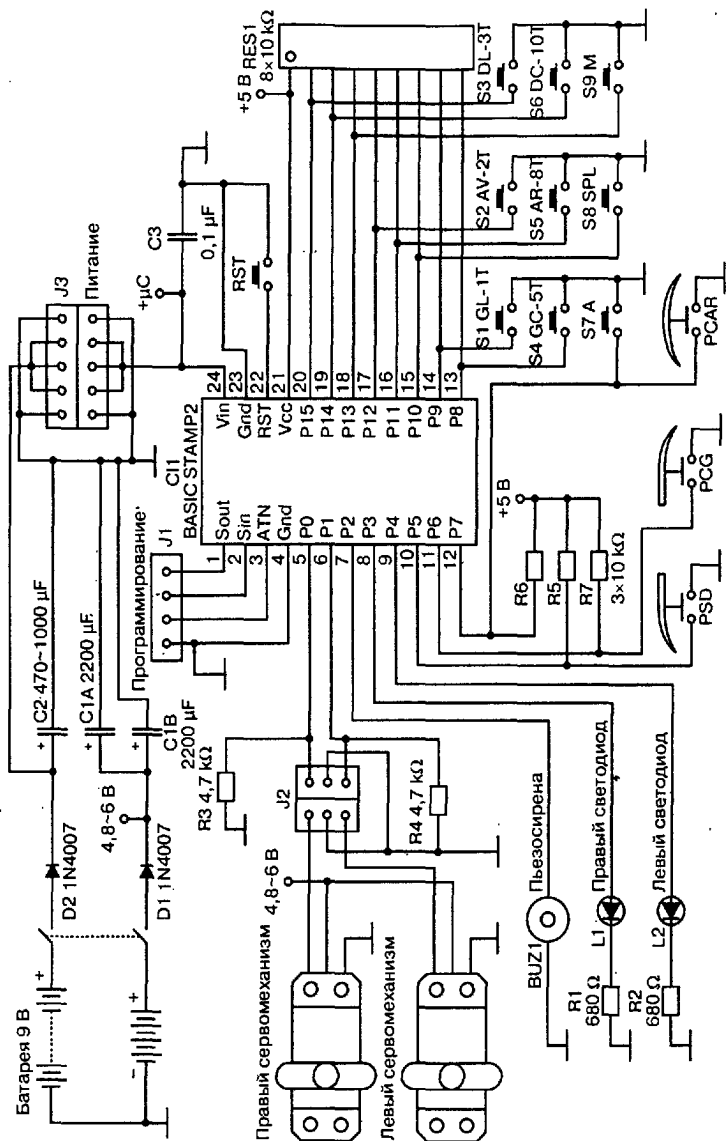
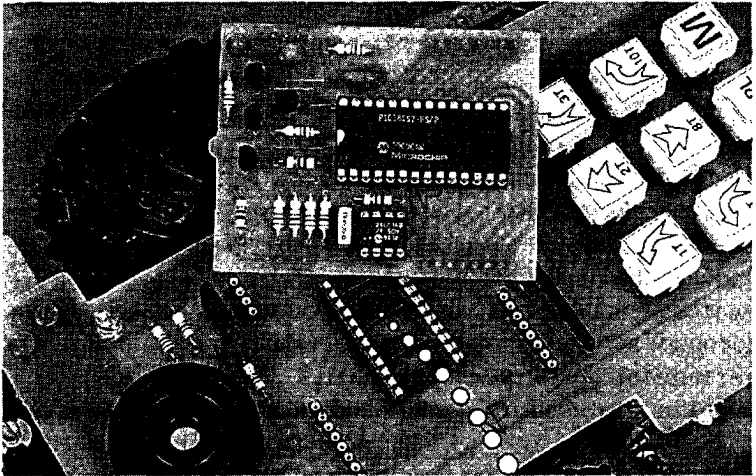


Рис. 14.2. Принципиальная схема мобильного робота



Микроконтроллер показан в корпусе DIP, но возможна его установка в корпусе CMS

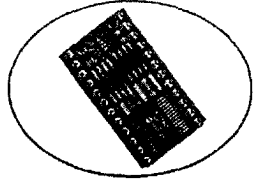
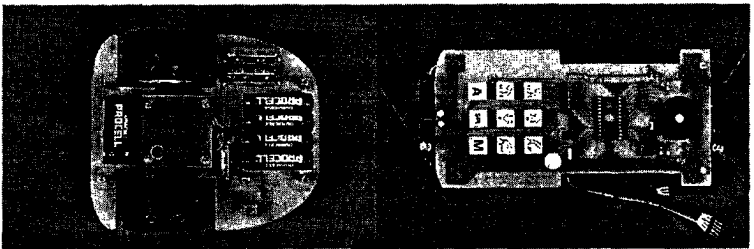


Рис. 14.3. Варианты выбора микроконтроллера BASIC STAMP 2

Питание мощных элементов полностью независимо от питания микроконтроллера. Это сделано для того, чтобы устранить помехи, вносимые двигателем. Батарея 9 В служит для питания блока управления через блокировочный диод D2. Затем напряжение фильтруется конденсатором C2. Два сервомеханизма питаются от четырех батарей 1,5 В (тип R6) общим



а б
Рис. 14.4. Плата шасси и плата управления робота

напряжением питания 6 В или четырех аккумуляторов 1,2 В в таком же конструктивном исполнении, но с меньшим общим напряжением питания +4,8 В. Диод D1 защищает схему от неправильного подключения батарей, а два конденсатора большой емкости (C1A и C1B) фильтруют это напряжение. Выключатель сделан сдвоенным, чтобы избежать утечки тока. Разъем J2 осуществляет связь с платой управления. Внимательно проверьте правильность расположения выводов, чтобы избежать проблем при подключении.

В схеме используется микроконтроллер BASIC STAMP 2, с которым вы уже хорошо знакомы. Обратите внимание, что питание микроконтроллера подается от батареи в 9 В на контакт Vin. Встроенный в микроконтроллер стабилизатор напряжения приводит его к 5 В для питания схемы управления. Этот стабилизатор обладает невысокой мощностью (ток нагрузки менее 100 мА). Конденсатор C3 осуществляет развязку входного напряжения на выводе 24 микросхемы C11. Дополнительная кнопка RST выполняет ручной сброс устройства. Разъем J1 позволяет выполнять программирование, не отсоединяя микроконтроллер.

Питание сервомеханизмов осуществляется сразу после включения питания. При отсутствии сигнала с микроконтроллера BS2, каналы управления P0 и P1 через резисторы R3 и R4 подключены к «земле». С помощью соединителя J2 устанавливается связь между двумя платами.

Индикаторные светодиоды L1 (правый) и L2 (левый), ток которых ограничивается резисторами R1 и R2, управляются по каналам P3 и P4. Робот также может осуществлять звуковую сигнализацию различных состояний с помощью пьезосирены малой мощности иначе называемой пьезозуммером или просто зуммером. Управление зуммером осуществляется по каналу P2. Эта небольшой радиокомпонент – зуммер – оживляет использование устройства. В схеме мы найдем также периферийные контакты, выполненные в форме правого, левого и задних амортизаторов. Эти проволочные контакты связаны с контактами микроконтроллера P5, P6 и P7 соответственно. Резисторы R5–R7 переводят данные входы на высокий уровень в состояние покоя.

Кнопки S1–S9, управляемые по каналам P7–P15, позволяют передавать команды роботу во время работы. Отметим, что канал P7 выполняет двойную функцию. Резистивная сборка RES1 поддерживает каналы P7–P15 на высоком уровне в состоянии покоя. В схеме шесть кнопок выполняют двойную роль (перемещение в выбранном направлении и длительность действия), три другие кнопки выполняют только одну комбинированную функцию: остановку, запуск и программирование/воспроизведение. Поведение робота, а следовательно, и действие каждого механизма полностью описывается программой, загруженной в микроконтроллер.

Электронная часть устройства

Устройство содержит две печатных платы, чертежи которых показаны на рис. 14.5 и 14.6. Материалы по технологии литогра-

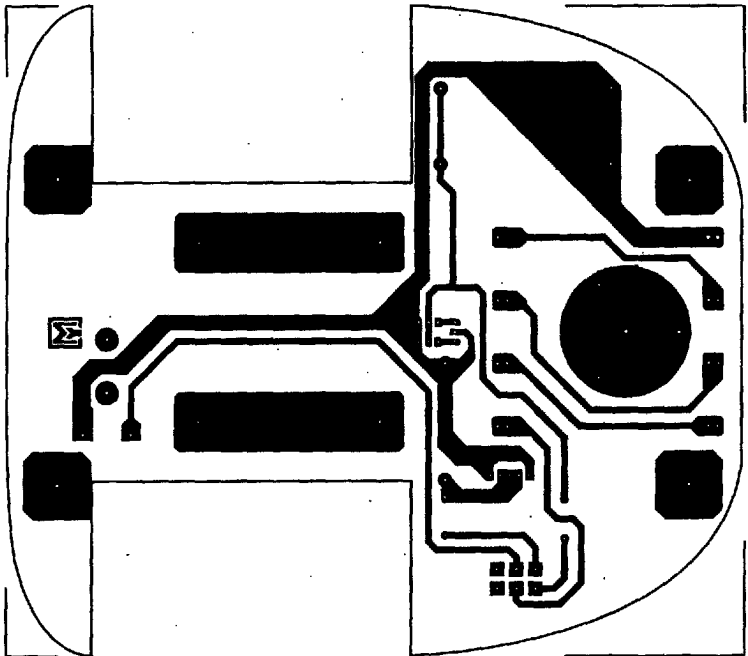


Рис. 14.5. Чертеж печатной платы шасси мобильного робота (в масштабе 1:2)

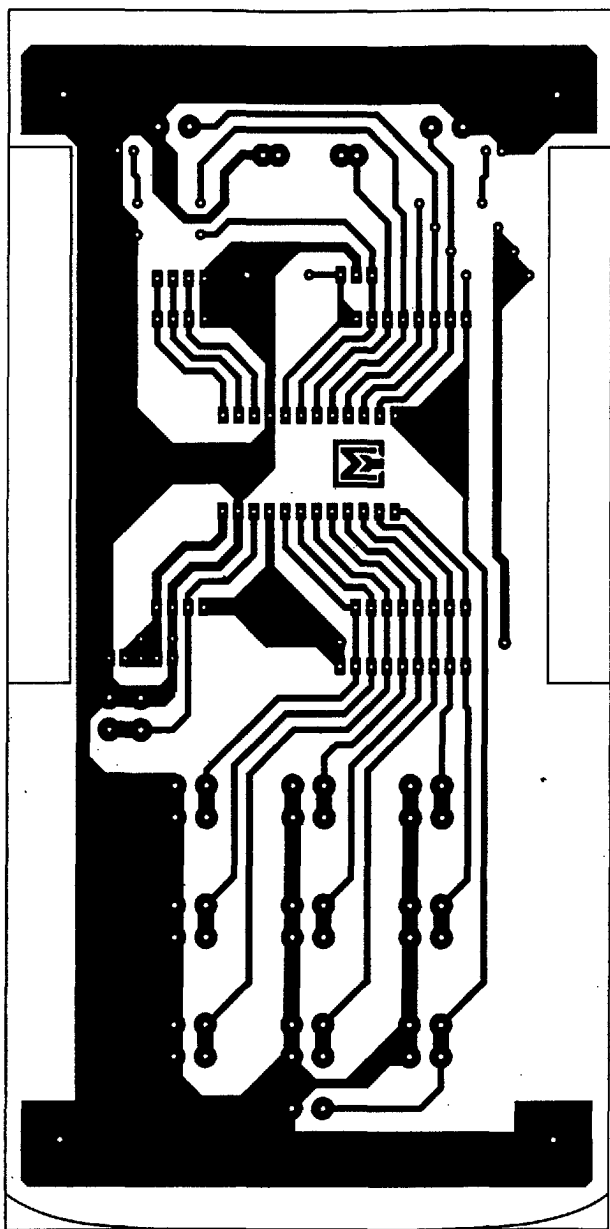


Рис. 14.6. Чертеж печатной платы управления мобильного робота (в масштабе 1:1)

фии вы можете найти в конце книги в библиографии. Одна из плат образует шасси нашего робота. Эта плата в оригинале имеет слишком большой размер, поэтому мы поместили ее на рисунке уменьшенной в два раза.

Тем не менее, чертеж этой платы, как и всех печатных плат, описанных в данной книге, доступен для скачивания с сайта <http://www.dunod.com> в масштабе 1:1 и занимает лист формата А4. После получения платы следует выполнить несколько обрезаов пилой по металлу согласно чертежу, а затем округлить края платы напильником. Не сверлите отверстия для элементов в этой плате. Чертеж второй печатной платы управления мобильного робота дан в масштабе 1:1 на рис. 14.6. Плату следует делать по обычной технологии литографии. После получения платы в ней необходимо сделать два выреза для колес.

Плата шасси содержит элементы питания и блок передвижения. Все элементы на этой плате запаиваются на стороне с медным покрытием согласно рисунку 14.7, который также

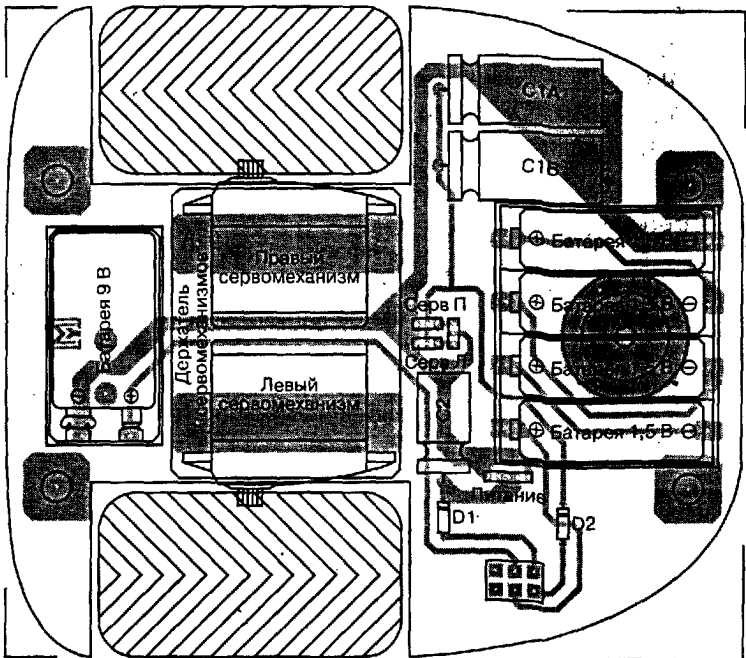


Рис. 14.7. Чертеж монтажной платы шасси мобильного робота (в масштабе 1:2)

уменьшен в два раза.

Начните с пайки диодов, затем запаяйте соединители J2 и J3, составленные из отдельных разъемов, соединители сервомеханизмов, образованные тремя отдельными разъемами, конденсаторы, выключатель, и, наконец, держатель батареи 9 В. Не устанавливайте сразу сервомеханизмы, так как они требуют серьезную доработку данной модификации, которую мы опишем ниже. После этого вы должны тщательно проверить вашу работу по монтажу и пайке выводов компонентов без подключения питания, установив также батарейку в держатель.

Чертеж монтажной схемы платы управления показана на рис. 14.8 в масштабе 1:1.

Список компонентов:

Резисторы с 5-процентным допуском:

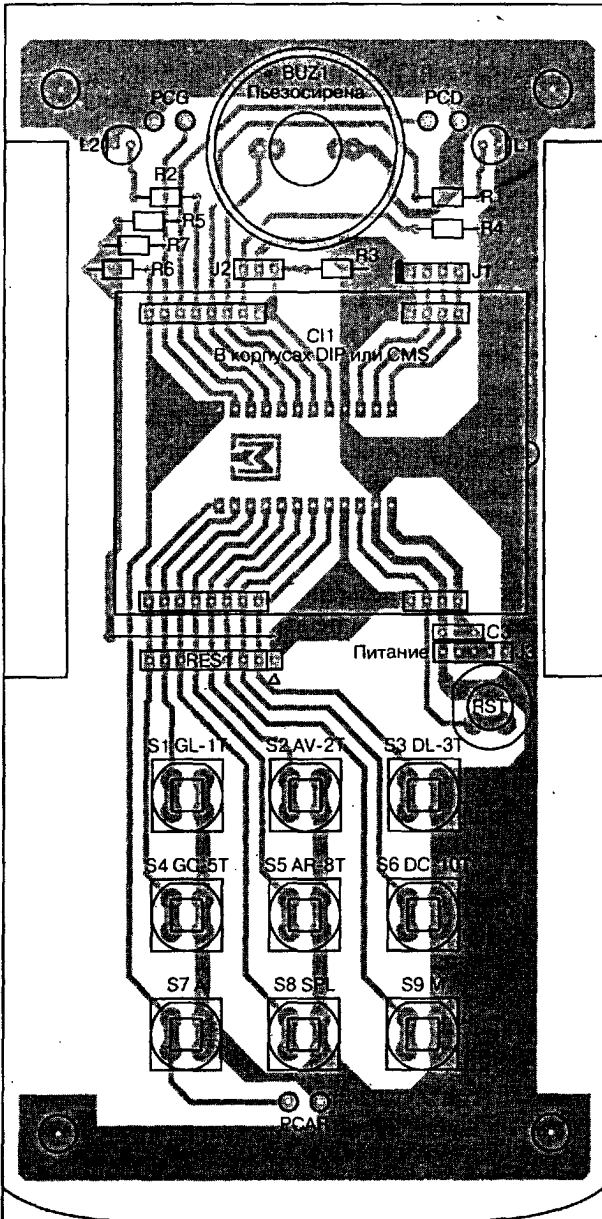
- R1, R2: 680 Ом (синий, серый, коричневый);
- R3, R4: 4,7 кОм (желтый, фиолетовый, красный);
- R5–R7: 10 кОм (коричневый, черный, оранжевый);
- RES1: резистивная сборка 8 × 10 кОм.

Конденсаторы:

- C1A и C1B: 2200 мкФ / 25 В (полярный танталовый или электролитический с аксиальными выводами);
- C2: 1000 мкФ / 25 В (полярный танталовый или электролитический с аксиальными выводами);
- C3: 0,1 мкФ, неполярный керамический.

Полупроводники:

- D1, D2: 1N4007;
- L1–L2: светодиод 5 мм (любой цвет на выбор);
- C11: BASIC STAMP 2 (модуль CMS или DIL).



с. 14.8. Чертеж монтажной платы управления мобильного робота (в масштабе 1:1)

Прочие элементы:

- 1 держатель микросхемы на 24 вывода;
- 24 отдельных разъема для подсоединения микроконтроллера в корпусе DIP;
- 12 отдельных разъемов;
- 14 отдельных разъемов;
- 10 кнопок;
- 4 держателя батарей типа R6;
- 1 держатель батареи 9 В;
- 1 пьезозуммер;
- 4 проволочных вывода;
- 1 сдвоенный переключатель;
- 2 модифицированных сервомеханизма;
- 1 роликовый шарикоподшипник для передвижения;
- 2 колеса диаметром около 80 мм;
- набор винтов и гаек диаметром 3 мм.

На плате управления элементы запаиваются как обычно: на обратной стороне, не покрытой медными дорожками. Сначала запаяйте перемычку, чтобы не забыть о ней впоследствии. Далее продолжите пайку, начав с самых маленьких элементов на плате – резисторов, затем запаяйте держатель микроконтроллера BASIC STAMP 2 на 24 вывода 24 отдельных разъема для подсоединения микроконтроллера в корпусе DIP, резистивную сборку, керамический конденсатор, кнопки, светодиоды, разъем J1, составленный из 4 отдельных разъемов типа «гнездо», и, наконец, пьезозуммер. Разъемы J2 и J3 типа «гнездо», составленные из 3 и 5 соединителей соответственно, нельзя запаивать на саму плату, их следует запаять на провода, идущие от платы шасси. Периферийные выводы PCD, PCG и PCAR приклеиваются к плате и при необходимости прикручиваются винтами рядом с местом пайки. Два первых вывода изготовлены из одного проволочного контакта, а последний – из двух выводов (по одному в каждое тыльное направление: с левого и правого боков). При этом они запаиваются параллельно к одной точке PCAR на плате. Периферийные выводы должны быть изогнуты таким образом, чтобы их срабатывание осуществлялось на фиксированном расстоянии до препятствия. Этими выводами робот определяет препятствия на своем пути.

После этого приступайте к этапу полной проверки выполненной работы. Подключите питание к плате, не вставляя микроконтроллер в разъем, предварительно подключив соединительные разъемы между двумя платами. Замерьте напряжение между выводом 24 микроконтроллера и «землей». Оно должно быть равно 5 В. Если после проверки все работает корректно, приступим к изготовлению механических деталей устройства и программированию.

Изготовление механических деталей

Механическая часть устройства включает сборку робота и модификацию внутренней конструкции сервомеханизмов. Мы начнем именно с этой тонкой работы, пошаговое описание которой в восемь этапов приводится в цветной вкладке книги (рис. 14.9–14.16).

Модификация сервомеханизмов

Эти небольшие устройства развивают крутящий момент в несколько килограмм на сантиметр, и это при напряжении в 5 или 6 В. Первоначальное предназначение этих устройств – передвижение узлов моделей уменьшенных размеров (рулевое управление, заслонки и др.). Сервомеханизмы обеспечивают круговое движение на 180° при длительности входных импульсов от 1,2 до 1,8 мс, что соответствует крайним значениям длительности (среднее значение равно 1,5 мс). Период повторения импульсов должен быть равен 20 мс. В нашем случае сервомеханизмы должны совершать круговое движение в 360° , количество оборотов при этом не ограничивается. Таким образом, необходимо обеспечить вращение механизмов, сохранив электронный принцип управления длительностью (шириной) подаваемых импульсов. Предельным значениям длительности импульсов соответствуют быстрые передвижения вперед и назад, а при средней длительности импульсов робот останавливается.

Наша доработка распространяется только на данное устройство. Возможно, сервомеханизмы, которые вы используете, немного отличаются от приведенных в нашей книге, однако принцип остается тем же самым. Иллюстрации на цветной вкладке должны помочь осуществить все доработки без особых трудностей.

1-й этап: откручивание винтов корпуса

Корпус сервомеханизма можно снять, открутив четыре винта из основания. Придерживайте корпус руками, чтобы во время откручивания составляющие сервомеханизм не рассыпались (рис. 14.9). Поместите корпус на промежуточную поверхность, гарантирующую отсутствие выпадения винтов с нее, чтобы не искать винты на полу.

2-й этап: снятие верхней крышки сервомеханизма

Снимите верхнюю крышку сервомеханизма, удерживая его строго вертикально. Старайтесь, чтобы оси шестеренок остались в своих позициях. Перед снятием крышки установите механизм в устойчивое положение, это особенно касается применения другого сервомеханизма, отличающегося от описываемого (рис. 14.10).

3-й этап: снятие нижней крышки сервомеханизма

Нижняя крышка сервомеханизма защищает двигатель, потенциометр привода, небольшую электрическую схему от внешних ударов. Крышка снимается без особого труда (рис. 14.11).

4-й этап: виртуальный обзор внутреннего устройства сервомеханизма

Этот виртуальный вид показывает все элементы устройства, находящиеся на своих местах, без корпуса. Все компоненты висят в воздухе как по волшебству (рис. 14.12). Это так называемая кинематическая схема. В реальности этого, конечно, добиться нельзя. Этот рисунок позволяет понять, каким образом передается момент от двигателя к системе шестеренок, оценить размер каждой детали и понять, как действует последняя шестерня на потенциометр привода.

В этой конструкции следует сделать две модификации, которые заключаются в том, чтобы убрать ограничители вращения выходной шестеренки.

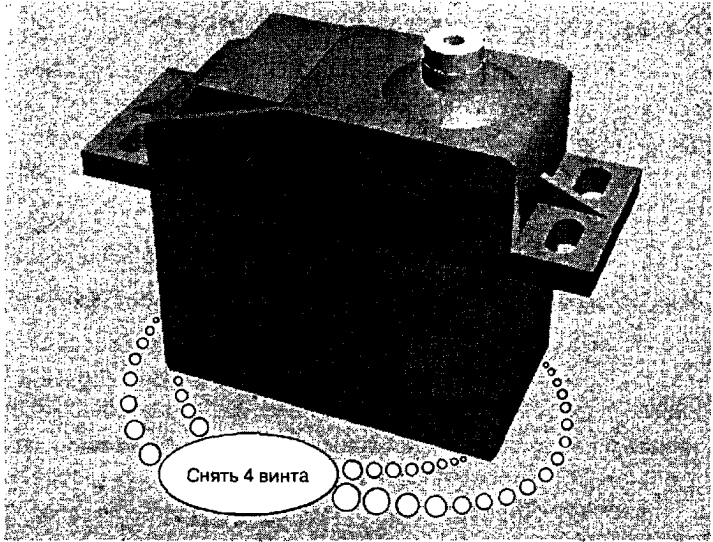


Рис. 14.9. Этап 1: модификации внутренней конструкции сервомеханизмов: откручивание винтов корпуса

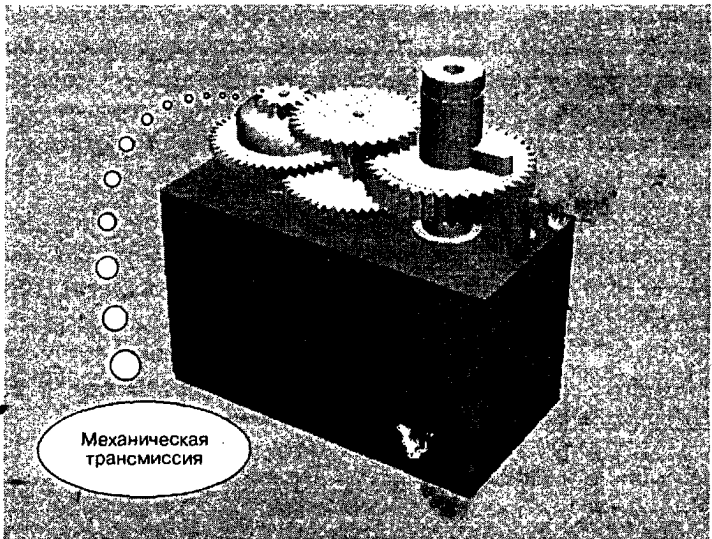


Рис. 14.10. Этап 2: модификации внутренней конструкции сервомеханизмов: снятие верхней крышки сервомеханизма

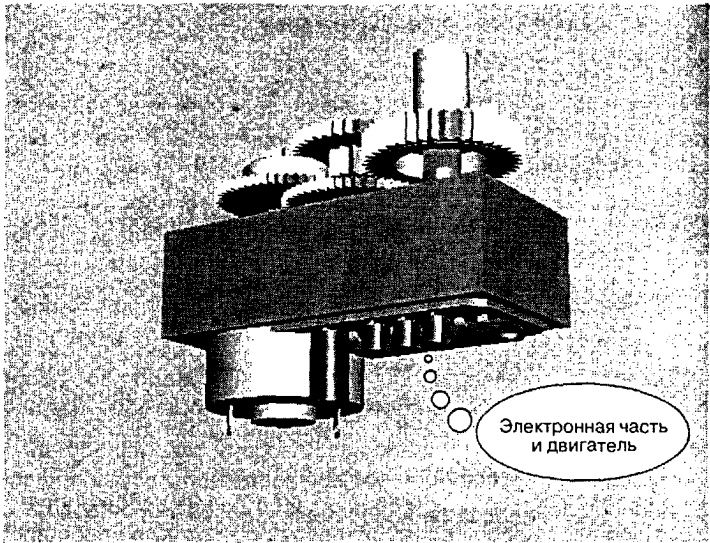


Рис. 14.11. Этап 3: модификации внутренней конструкции сервомеханизмов: снятие нижней крышки сервомеханизма

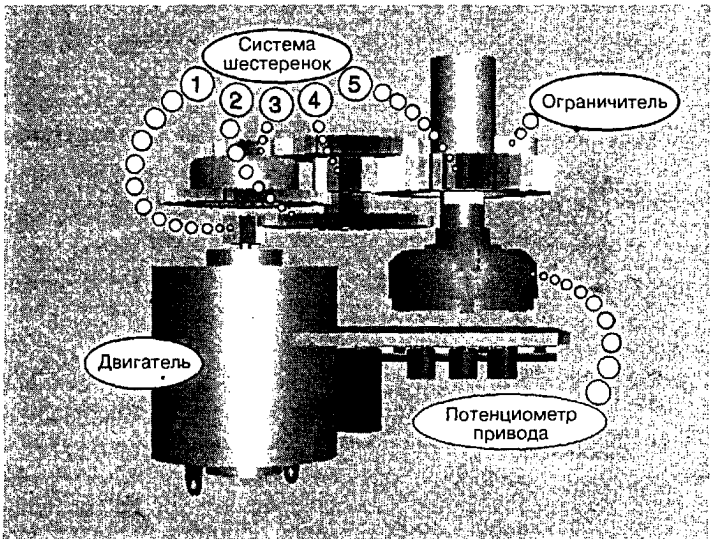


Рис. 14.12. Этап 4: модификации внутренней конструкции сервомеханизмов: виртуальный обзор внутреннего устройства сервомеханизма

5-й этап: изменение конструкции выходной шестеренки

Начнем с самого легкого: удаление ограничителя выходной шестеренки. Перед тем как выполнить эту операцию, надо снять шестеренку.

info Некоторые современные сервомеханизмы имеют небольшой шарикоподшипник на этой позиции.

Для выполнения операции воспользуйтесь малой ножовкой по металлу с тонким полотном. После того, как вы отпилили ограничитель, зачистите место резцом или напильником. Будьте аккуратны и не повредите зубья шестеренки во время выполнения работы (рис. 14.13).

6-й этап: модификация потенциометра привода

В оригинальной конструкции потенциометр привода служит для электрической индикации положения выходной шестеренки. Сигнал подается в схему управления. В нашем устройстве необходимость в этой информации отпадает, так как колесо должно совершить большое количество оборотов. Однако легко и просто удалить его не получится, так как этот потенциометр составляет выходную ось. Кроме того, часто этот потенциометр приклеивается к корпусу с помощью синтетического клея. Осторожно снимите печатную плату, не отрывая соединительный провод. Затем аккуратно отпаяйте три провода, запомнив или лучше зарисовав их расположение на отдельном листке. Эти провода затем следует запаять обратно на новый небольшой многооборотный потенциометр, который в отличие от удаляемого может совершать бесконечное количество оборотов. После этого следует отогнуть три защелки на корпусе и снять крышку потенциометра, потом срезать ограничитель в корпусе и удалить направляющие контактные пластины. После этих доработок вставьте верхнюю крышку, закрепив три защелки, и установите потенциометр на свое место (рис. 14.14).

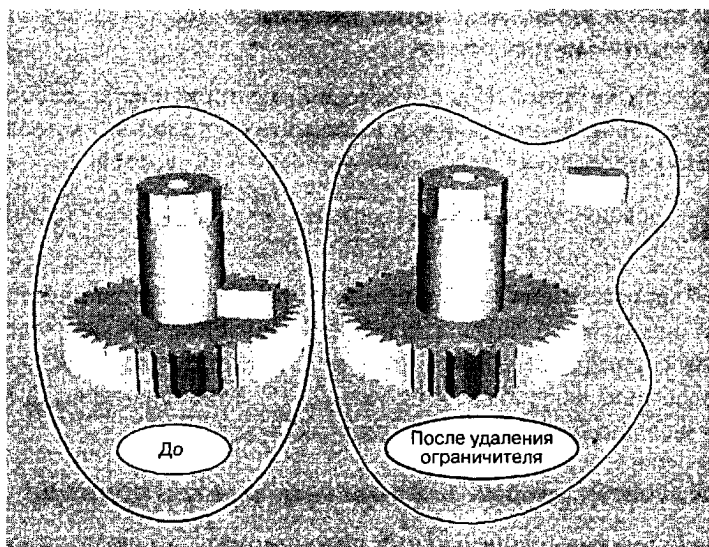


Рис. 14.13. Этап 5: модификации внутренней конструкции сервомеханизмов: изменение конструкции выходной шестеренки (отпиливание ограничителя шестеренки)

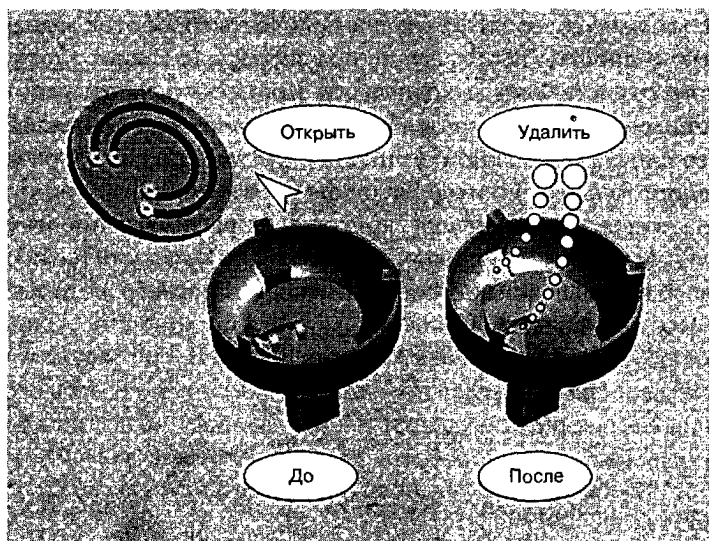


Рис. 14.14. Этап 6: модификации внутренней конструкции сервомеханизмов: модификация потенциометра привода

7-й этап: установка дополнительного потенциометра

Установите доработанный потенциометр на свое место (предварительно убедитесь, что он легко проворачивается на 360° без помех). Запаяйте 3 провода на новый, дополнительный потенциометр, соблюдая правильное расположение проводов и хорошо заизолировав места пайки. После этого прикрепите его с помощью клея у основания старого потенциометра. Установите плату таким образом, чтобы нижняя крышка сервомеханизма смогла закрыться. Далее сделайте отверстие в корпусе сервомеханизма так, чтобы в него попал регулировочный винт дополнительного потенциометра (рис. 14.15). Если новой конструкции мешает электролитический конденсатор, то его нужно аккуратно переместить на другую сторону платы.

В худшем случае придется поменять потенциометр внутри сервомеханизма, однако автору книги ни в одном случае не приходилось прибегать к этому варианту. Закрепите нижнюю крышку корпуса, обращая внимание на то, чтобы обеспечивался свободный доступ к регулировочному винту.

8-й этап: установка шестеренок и закрытие верхней крышки

Установите все шестеренки в порядке, обратном порядку при вскрытии. Обратите внимание на то, чтобы два маленьких вала встали на свои места. Не забудьте нанести на зубья шестеренок смазочное масло, которое содержится на дне корпуса. После этого установите верхнюю крышку и закрутите четыре нижних винта (рис. 14.16). Теперь приступайте к выполнению регулировки сервомеханизма с помощью специального тестера или системы радиоконанд.

Выходная шестеренка должна вращаться без остановок при подаче управляющих импульсов предельной длительности: максимальной или минимальной. В среднем положении регулировкой дополнительного потенциометра можно добиться остановки сервомеханизма. После выполнения регулировки определите ускорение двигателя, изменения ширину (длительность) импульса от средней до максимальной или минимальной при постоянном периоде повторения.

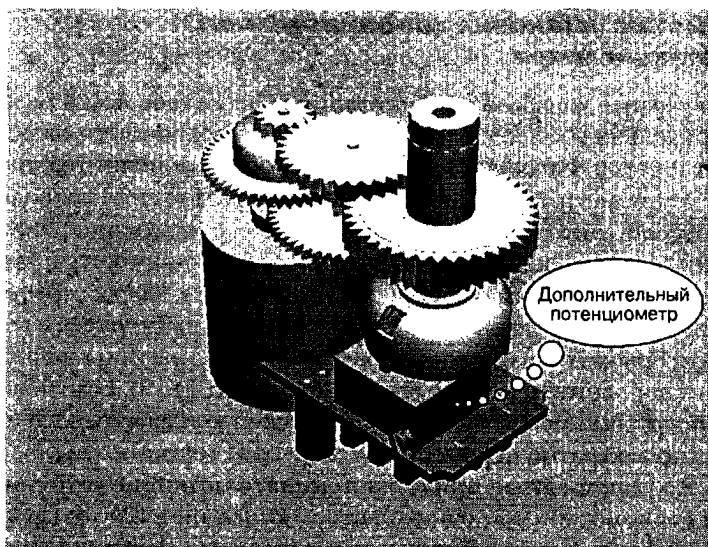


Рис. 14.15. Этап 7: модификации внутренней конструкции сервомеханизмов: установка дополнительного потенциометра

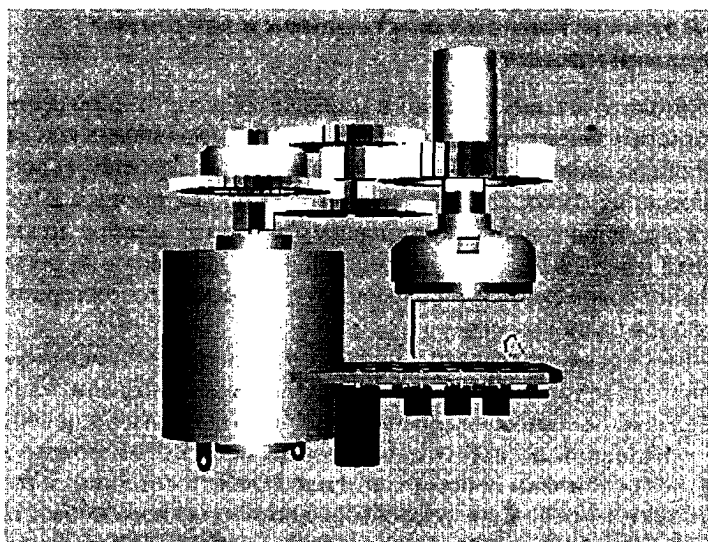


Рис. 14.16. Этап 8: модификации внутренней конструкции сервомеханизмов: установка шестеренок и закрытие верхней крышки

Монтаж устройства

На рис. 14.17 показаны датчики в передней части робота и проводочные контакты. Аналогично изготавливаются и датчики, установленные сзади робота.

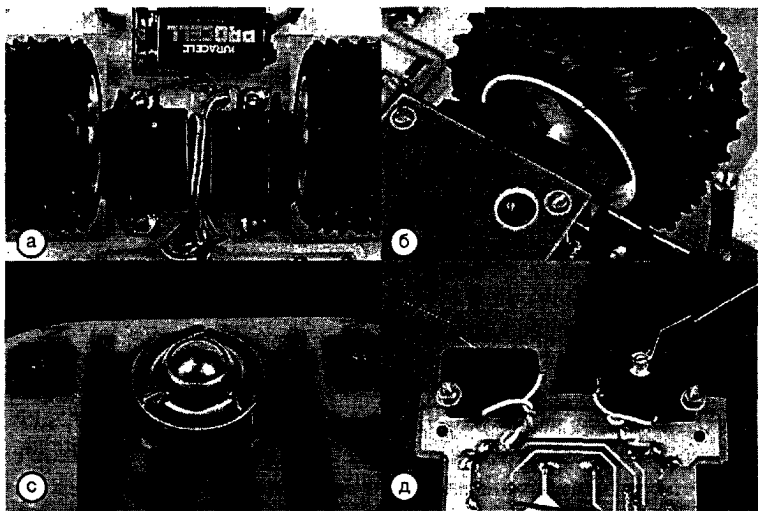


Рис. 14.17. Различные особенности механической сборки робота

- а – крепление сервомеханизмов на плату шасси с помощью клея;
- б – крепление колеса на сервомеханизме;
- в – крепление шарикового колеса сзади на плате шасси;
- г – принцип крепления контактных датчиков движения.

В первую очередь вы должны найти два колеса диаметром около 80 мм. Возможно приобрести их на развале бывших в употреблении товаров для моделизма или в магазинах соответствующего назначения. Самым простым решением, которое использовали мы, является повторное использование колес от детских игрушек. Более умелые могут изготовить их самостоятельно, используя в качестве материала картон, прессованный полистирол, пластинки печатных плат, самоклеющуюся пену для банджа колес... Какой бы метод вы не выбрали, следует обязательно состыковать колесо с сервомеханизмом с помощью специальной зубчатой передачи. Также следует найти передвижное шариковое колесо; эту деталь можно легко найти в многочисленных самодельных устройствах или, если

вы не боитесь затрат, в каталогах дистрибьюторов элементов, осуществляющих доставку по почте. После этого подберите небольшой кусок изолированной платы для крепления сервомеханизмов в отходах материалов для печатных плат, например из плексигласа. Размер этой пластины приводится в масштабе 1:2 на рисунке монтажной платы шасси.

Теперь у вас есть все детали для завершения монтажа робота. Установите сервомеханизмы, как показано на фотографии с помощью специального клея. Закрепите провода с разъемами в правильном положении. Прикрутите винтами плату крепления, чтобы держать сервомеханизмы. Если гайки сделаны из латуни, их можно припаять непосредственно к плате шасси. Рекомендуется сделать два отверстия в корпусах сервомеханизмов для того, чтобы производить при необходимости настройку режима полной остановки робота.

Закрепите шариковое колесо под держателем батареек 1,5 В, регулируя при этом его высоту с помощью столбиков-бонок с нарезанной резьбой или металлических банок. Высота столбиков подбирается таким образом, чтобы получить небольшой наклон робота назад. Затем запаяйте 4 держателя батарей, предварительно приклеив их с помощью двухстороннего скотча.

Последним этапом является состыковка двух плат при помощи столбиков с нарезанной резьбой для получения необходимой высоты. Если платы находятся на слишком близком расстоянии, вы рискуете замкнуть накоротко верхнюю плату. Достаточное расстояние между платами позволит менять использованные батарейки, не разбирая устройство.

Программирование и включение

Методика программирования микроконтроллера BASIC STAMP 2 была описана во второй части этой книги. Таким же способом напишите программу мобильного робота или скачайте ее с нашего сайта в Интернете, написанную на языке программирования PBASIC 2. Программа представлена в листинге 14.1 (MOBILE.BS2). В случае каких-либо проблем, возникающих при программировании, мы советуем вам руководствоваться правилами программирования этого языка, написанными во второй части книги. В библиографии в конце книги вы найдете различные журналы и книги, посвященные этой теме.

Листинг 14.1. Программа мобильного робота на базе микроконтроллера

```

' === ПРОГРАММА УПРАВЛЕНИЯ МОБИЛЬНЫМ РОБОТОМ ===
' === НА ЯЗЫКЕ PVBASIC2 © Yves MERGY 2001 ==
' =====

' === ОБЪЯВЛЕНИЯ ПЕРЕМЕННЫХ И КОНСТАНТ ===
CTR var word ' Счетчик импульсов
Hz var word ' Звуковая частота
DEP var nib ' Тип перемещения
TMP var nib ' Длительность перемещения
ADM var word ' Адрес в памяти EEPROM
MG var word ' Импульс левого двигателя
MD var word ' Импульс правого двигателя
IMP1 con 850 ' Нормальный импульс = 1,7 мс
IMPN con 650 ' Импульс обратной полярности = 1,3 мс
MAX1 con 1000 ' Старший адрес в памяти EEPROM
SEC con 10 ' Множитель времени
' === ПРИМЕР ПРОХОЖДЕНИЯ В ПАМЯТИ EEPROM ===
DATA 0,2,1,2,2,2,3,2,4,2,5,2,6,2,0,2
DATA 1,3,3,2,1,3,3,2,1,3,3,2,1,3,3,2,7
' === ИНИЦИАЛИЗАЦИЯ ===
DIRS=%0000000000001111'P0-P4 - выходы, P5-P15 - входы
OUUTA=0

' === РЕЖИМ ВОЗВРАТА ===
EXECUTION:
if in10=0 then EXECUTE
REST1:
low 3 : low 4 : gosub SOUND2
if in10=1 then PROGRAMMING
if in13=1 then REST1
ADM=0 : gosub SOUND1
RELOCATION:
if in10=0 then PROGRAMMING
' === ЧТЕНИЕ ПЕРЕМЕЩЕНИЯ, ДЛИТЕЛЬНОСТИ И АДРЕСА + 1 ===
read ADM,DEP : ADM=ADM+1 : read ADM,TMP : ADM=ADM+1
' === ИМПУЛЬСЫ ДЛЯ СЕРВОМЕХАНИЗМОВ В ЗАВИСИМОСТИ ===
' === ОТ ПЕРЕМЕЩЕНИЯ И ПОДКЛЮЧЕНИЯ ===
branch DEP,[D0, D1, D2, D3, D4, D5, D6, ADFIN]
D0:
MD=750 : MG=750 : goto ROUL1
D1:
MD=IMP1 : MG=IMPN : goto ROUL1
D2:
MD=IMPN : MG=IMP1 : goto ROUL1
D3:
MD=IMP1 : MG=750 : goto ROUL1
D4:
MD=750 : MG=IMPN : goto ROUL1

```

```

D5:
MD=IMPI : MG=IMPI : goto ROUL1
D6:
MD=IMPN : MG=IMPN : goto ROUL1
ADFIN`
ADM=0 : goto ROULE
ROUL1:
CTR=0
' === ТЕСТИРОВАНИЕ СРАБАТЫВАНИЯ КОНТАКТОВ СОУДАРЕНИЯ, ==
' === ОТПРАВЛЕНИЕ ИМПУЛЬСОВ НА СЕРВОМЕХАНИЗМЫ И ===
' === ЗАЦИКЛИВАНИЕ ПРОМЕЖУТКА МЕЖДУ 2 ИМПУЛЬСАМИ ===
ROUL2:
if in10=0 then PROGRAMMING
if in7=0 then C_BACK
if in5=0 then C_RIGHT
if in6=0 then C_LEFT
pulsout 0,MD : pulsout 1,MG
pause 30 : CTR=CTR+1
if CTR=TMP*SEC then ROULE
' === ЦИКЛ ВЫПОЛНЕНИЯ ПЕРЕМЕЩЕНИЯ ===
goto ROUL2
C_BACK:
high 3 : high 4 : gosub SOUND3
if in13=0 then EXECUTION
goto C_BACK
C_RIGHT:
high 3 : high 4 : gosub SOUND3
if in13=0 then EXECUTION
goto C_RIGHT
C_LEFT:
high 3 : high 4 : gosub SOUND3
if in13=0 then EXECUTION
goto C_LEFT

' === РЕЖИМ ПРОГРАММИРОВАНИЯ ===
PROGRAMMING:
if in10=0 then PROGRAMMING
PROGRAMMING1:
toggle 3 : toggle 4 : gosub SOUND1
if in10=0 then EXECUTION
if in13=1 then PROGRAMMING1
ADM=0
RELOCATION:      'Сканирование перемещений
low 3 : high 4
if ADM=MAXI then END
if in10=0 then END
if in7=0 then STOPPING
if in8=0 then SM_LEFT 'SM - короткое перемещение
if in9=0 then LM_LEFT 'LM - длинное перемещение

```

```

if in11=0 then BACK
if in12=0 then FORWARD
if in14=0 then SM_RIGHT
if in15=0 then SL_RIGHT
goto RELOCATION
TIME:      'Сканирование промежутков времени
high 3 : low 4
if in8=0 then T5
if in9=0 then T1
if in11=0 then T8
if in12=0 then T2
if in14=0 then T10
if in15=0 then T3
goto TIME
' === Перемещения ===
STOPPING:  'Остановка
if in7=0 then STOPPING
DEP=0 : gosub STORAGE
goto TIME
ST_LEFT:   'Короткий поворот влево
if in8=0 then ST_LEFT
DEP=5 : gosub STORAGE
goto TIME
LT_LEFT:   'Длинный поворот влево
if in9=0 then LT_LEFT
DEP=3 : gosub STORAGE
goto TIME
BACK:      'Задний ход
if in11=0 then BACK
DEP=2 : gosub STORAGE goto TIME
FORWARD:   'Движение вперед
if in12=0 then FORWARD
DEP=1 : gosub STORAGE
goto TIME
ST_RIGHT:  'Короткий поворот вправо
if in14=0 then ST_RIGHT
DEP=6 : gosub STORAGE
goto TIME
LT_RIGHT:  'Длинный поворот вправо
if in15=0 then LT_RIGHT
DEP=4 : gosub STORAGE
goto TIME
END: 'Конец программирования
if in10=0 then END
write ADM,7 : goto EXECUTION
STORAGE:   'Запоминание перемещений
freqout 2,200,1000
write ADM,DEP : ADM=ADM+1
return

```

```

' === Длительности ===
T1:
if in9=0 then T1
freqout 2,200,2000
TMP=1 : write ADM,TMP : ADM=ADM-1
goto RELOCATION
T2:
if in12=0 then T2
freqout 2,200,2000
TMP=2 : write ADM,TMP : ADM=ADM+1
goto RELOCATION
T3:
if in15=0 then T3
freqout 2,200,2000
TMP=3 : write ADM,TMP : ADM=ADM+1
goto RELOCATION
T5:
if in8=0 then T5
freqout 2,200,2000
TMP=5 : write ADM,TMP : ADM=ADM+1
goto RELOCATION
T8:
if in11=0 then T8
freqout 2,200,2000
TMP=8 : write ADM,TMP : ADM=ADM+1
goto RELOCATION
T10:
if in14=0 then T10
freqout 2,200,2000
TMP=10 : write ADM,TMP : ADM=ADM+1
goto

' === МЕЛОДИЯ -1- ===
SOUND1:
FOR Hz = 0 to 4000 step 500
freqout 2, 10, Hz
next
return
' === МЕЛОДИЯ -2- ===
SOUND2:
FOR Hz = 0 to 2000 step 100
freqout 2, 10, 2000-Hz
next
return
' === МЕЛОДИЯ -3- ===
SOUND3:
FOR Hz = 0 to 2000 step 100
freqout 2, 20, Hz, 2000-Hz
next
return

```

Вставьте микроконтроллер и батарейки в держатели и включите питание робота, сдвоенным выключателем. После этого с помощью программы STAMPW.EXE произведите программирование робота. Если все в порядке, то программирование произойдет в течение нескольких минут. Теперь вы можете выключить программатор и компьютер! Ваш робот может сделать первые передвижения автономно.

Управление устройством очень простое и осуществляется от кнопок (рис. 14.18). Вначале программа переходит в режим воспроизведения данных и ожидает нажатия на кнопку М для начала передвижения, заложенного в память.

Нажатие на кнопку А или воздействие на один из периферийных контактов останавливает робота. Светодиод указывает причину остановки. Кнопка SPL позволяет переходить из одного режима в другой (программирование или воспроизведение). Она работает после нажатия на кнопку М. В режиме программирования загорается левый светодиод, который информирует о том, что программа ожидает команды перемещения в заданном направлении. Команды подаются с помощью кнопок S1-S7. Кроме того, требуется дополнительно задать длительность перемещения с помощью кнопок S1-S6. Индикацию этих режимов обеспечивает правый светодиод. Описание работы каждой кнопки вы можете увидеть в табл. 14.1.

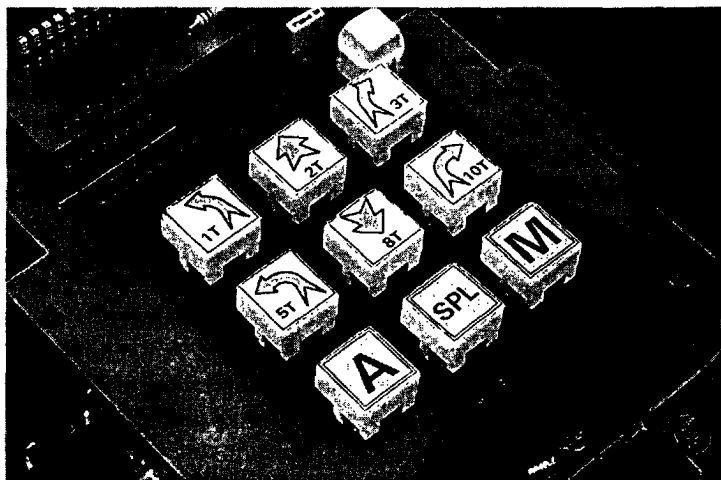


Таблица 14.1. Функции кнопок управления

Кнопка	Перемещение	Длительность, отн. ед.*	Примечание
S1	Длинный поворот налево	1	1Т
S2	Вперед	2	2Т
S3	Длинный поворот направо	3	3Т
S4	Короткий поворот налево	5	5Т
S5	Назад	8	8Т
S6	Короткий поворот нвправо	10	10Т
S7	Остановка		Остановка цикла
S8			Выбор режима
S9			Запуск

* отн. ед. = относительных единиц. – Примеч. науч.ред.

После этого снова загорается левый светодиод, который показывает, что можно приступить к записи следующего движения и т. д. Чтобы остановить цикл программирования, достаточно перейти в режим воспроизведения, когда горит левый светодиод. После того, как при программировании робот проделывает последнее передвижение, доступное для записи в память, он автоматически переходит в режим воспроизведения, чтобы не стереть уже записанные движения. Памяти устройства достаточно, чтобы записать путь в 500 отрезков перемещения и 500 разных длительностей перемещений!

Короткие звуки «бип» помогут вам правильно запрограммировать робота, а звуковые эффекты оповещают о смене режима работы. При наезде на препятствие и срабатывании одного из периферийных контактов загорается один из светодиодов в случае движения назад и два светодиода в случае движения вперед. Также это действие сопровождается звуком тревожной сигнализации.

В случае, когда сервомеханизмы передвигаются с разными скоростями, необходимо немного изменить константы IMP1 или IMPN, проведя различные эксперименты при движении вперед и назад. Наилучшие результаты должны получиться, когда эти константы принимают средние значения из своих диапазонов, так как в случае, когда эти константы принимают предельные значения, скорость максимальна.

14.3. Рекомендации по работе устройства

Наконец настал момент, когда ваш мобильный робот полностью готов и работает согласно описанной в книге инструкции. Возможно, вы будете этим удовлетворены, однако робот – это такое устройство, которое должно действовать по заложенным роботостроителем в него алгоритмам работы, который ограничен только своим воображением. Описанные ниже строки позволят вам более широко использовать это устройство.

Прежде всего важно определить «сценарий» работы. Основываясь на описанном выше тексте, возможно построить простую модель робота, который бы просто совершал серию перемещений, описанных программой в BS2 и при этом не наткнулся ни на какие препятствия. Эта модель является идеальной для начального изучения и имеет ограниченный интерес. Более интересной является модель, в которой, не изменяя электронную часть устройства, можно изменить программу, которая бы наделила нашего робота искусственным разумом. Цепочка алгоритмов может быть описана следующим образом:

1. Робот начинает движение из определенной точки комнаты в режиме программирования.
2. Он двигается до тех пор, пока не столкнется с препятствием. Длительность этого перемещения заносится в переменную.
3. В собственной памяти EEPROM регистрируется это перемещение и длительность, которая меньше определенного интервала времени (обозначается, например UT), который позволяет избежать столкновения.
4. После момента времени UT в память микроконтроллера ничего не записывается.
5. Траектория движения корректируется следующим образом: определяется какой из периферийных контактов столкнулся с препятствием и затем выбирается противоположное движение небольшой длительности. Это движение записывается в микроконтроллер. Эта корректирующая программа позволит роботу не столкнуться с препятствием.
6. Далее процесс повторяется по пункту 2 до тех пор, пока не заполнится память или вы не остановите процесс программирования.

7. Начав движение со своего первоначального положения, робот больше не встретит препятствие.

Для выполнения такой задачи требуется хорошо знать язык программирования, но эти знания вы легко приобретете на практике.

Наш робот может также претерпеть незначительные доработки в электронной схеме, поэтому в него лучше установить датчики. Для этого нужно удалить две-три кнопки с блока управления, соответствующие каналы будут также находиться на высоком логическом уровне за счет резисторной сборки RES1. На место кнопок достаточно запаковать небольшие схемы датчиков различного типа: на основе фототранзисторов, термисторов, микрофона, инфракрасных датчиков, радиоуправляемых датчиков, например *Miprot*, *Aurel*, *Telecontrolli* и др. Этот список неисчерпаем и ограничен только вашим собственным воображением.

Эти преобразования снабжают робота базовыми органами чувств, такими, как слух, зрение, осязание. Ему теперь не хватает только голоса! Современная электроника не знает границ. В настоящее время существуют любительские схемы, записывающие и воспроизводящие речь, а также другие устройства воспроизведения звука. Со временем, производя целый ряд усовершенствований, можно превратить робота в компаньона, который чем-то может напоминать меньших братьев, но только говорящих. Это кому-то скрасит одиночество или заставит прекратить хандрить и поможет заняться коллективным техническим творчеством.

14.4. Заключительное слово

Мы надеемся, что эта книга удовлетворила ваши запросы и вызвала у вас желание продвигаться дальше в этом увлекательном разделе техники, как цифровая электроника. Базовые знания, которые вы приобрели, помогут вам применить их в разработке своих собственных приложений. Не падайте духом, очень редко бывает, что программа начинает работать с первого раза. Настойчивость и практика – ключи успеха в изучении естественных наук. Через занятия электроникой люди приходят в новую увлекательную профессию, складываются сообщества по интересам и заводятся новые верные друзья.

Приложение **1**

Логический зонд

Универсальный источник питания

Дополнительные монтажные схемы, описанные в этой книге, являются эффективными практичными устройствами, которые повсеместно продаются. Но, несомненно, будет гораздо дешевле, если вы их сделаете сами, к тому же вы получите от этого удовольствие. Изготовление этих схем не является обязательным, в рамках этой книги не ставилась цель углубленного изучения этих устройств. Тем не менее их краткое описание дополняется всеми необходимыми схемами и чертежами печатных плат.

Логический зонд

Принципиальная схема

Логический зонд, принципиальная схема которого представлена на рис. П.1.1, способен различить три логических уровня: высокий уровень, низкий уровень и неопределенное состояние. Он поможет при настройке схем данной книги.

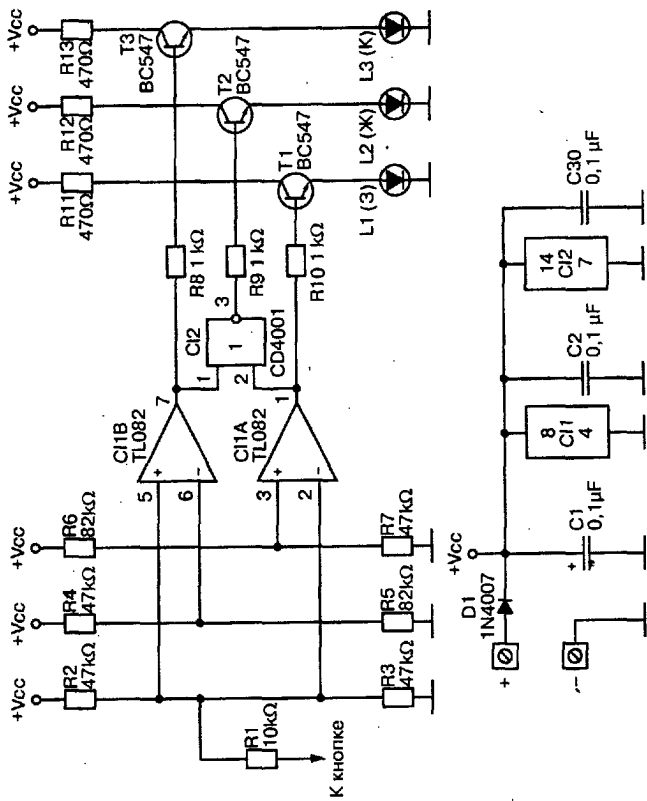


Рис. П. 1. 1. Принципиальная схема логического зонда

Список компонентов:**Резисторы с 5-процентным допуском:**

- R1, R2: 680 Ом (синий, серый, коричневый);
- R3, R4: 4,7 кОм (желтый, фиолетовый, красный);
- R5, R6: 82 кОм (серый, красный, оранжевый);
- R8–R10: 1 кОм (коричневый, черный, красный);
- R11–R13: 470 Ом (желтый, фиолетовый, коричневый).

Конденсаторы:

- C1: 47–100 мкФ / 25 В (полярный танталовый или электролитический с радиальными выводами);
- C2, C3: 0,1 мкФ, полярные керамические.

Полупроводники:

- CI1: TL082;
- CI2: CD4001;
- T1–T3: BC547;
- L1: светодиод красного цвета, диаметр 5 мм;
- L2: светодиод желтого цвета, диаметр 5 мм;
- L3: светодиод зеленого цвета, диаметр 5 мм.

Прочие элементы:

- 1 держатель микросхемы на 14 выводов;
- 1 держатель микросхемы на 8 выводов;
- 1 красный монтажный провод;
- 1 черный монтажный провод;
- 1 красный монтажный провод;
- 1 вывод (можно использовать заостренный конец неисправного диода).

Принцип работы зонда довольно прост. Два операционных усилителя (ОУ) – С1А и С1В – собраны по схеме компаратора с «окном». Три моста делителя (R2–R3, R4–R5, R6–R7) определяют пороги напряжения. Выход 1 первого ОУ переходит на высокий уровень при подаче логической единицы на вход и открывает транзистор Т1. При этом загорается зеленый светодиод L1. Выход 7 второго ОУ переходит на высокий уровень при подаче логического нуля на вход и открывает транзистор Т3. При этом загорается красный светодиод L3. Если на обоих выходах присутствует низкий логический уровень, то сигнал с выхода 3 логического элемента С12 ИЛИ-НЕ открывает транзистор Т2. При этом загорается желтый светодиод L2, указывающий на неопределенный уровень сигнала на входе.

Питание зонда, ограничиваемое диодом, берется с дополнительной платы для тестирования с помощью двух контактных проводов. Фото логического зонда приведено (рис. П.1.2).

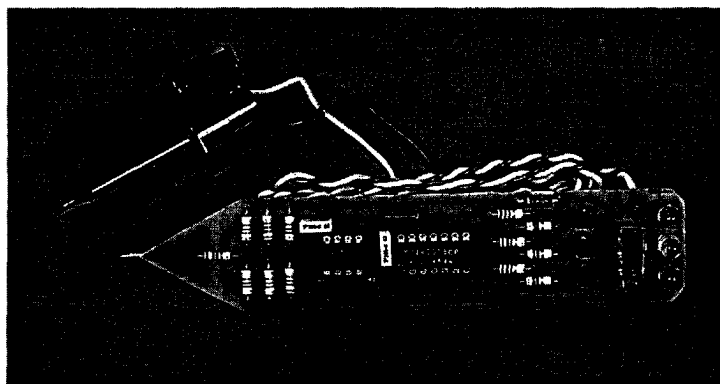


Рис. П.1.2. Логический зонд в виде контактного щупа с проводами для подключения питания

Реализация

На рис. П.1.3. представлена схема печатной платы в масштабе 1:1. Работа по сборке платы осуществляется в соответствии со схемой монтажной платы устройства, показанной на рис. П.1.4.

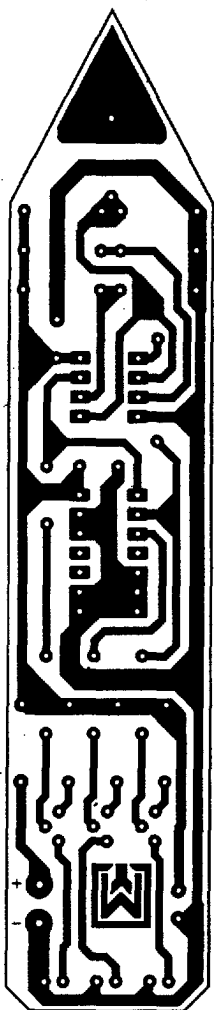


Рис. П. 1.3. Схема печатной платы логического зонда

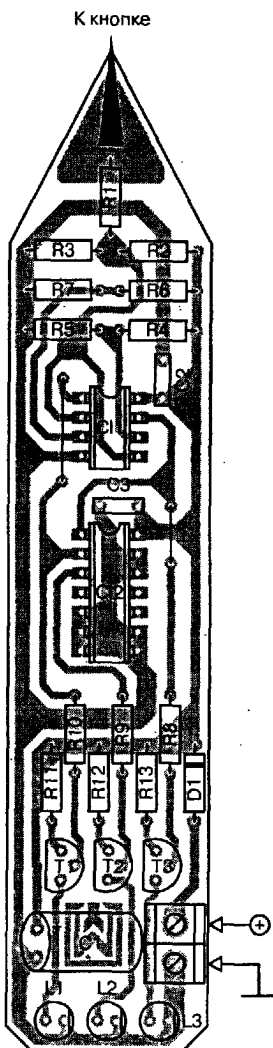


Рис. П. 1.4. Монтажная схема элементов логического зонда

Универсальный источник питания

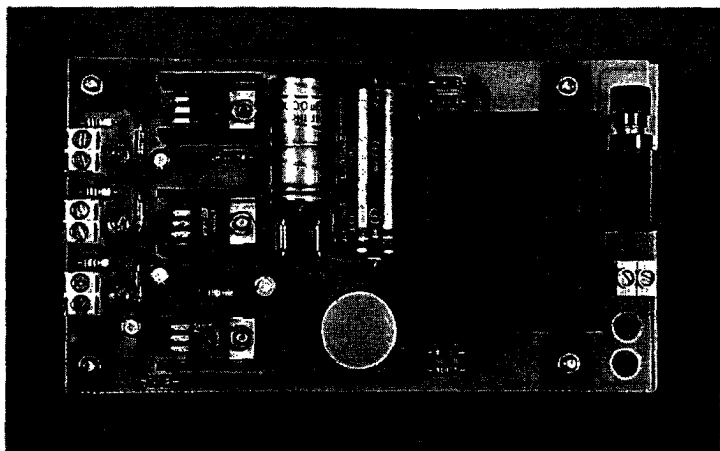


Рис. П. 1.5. Универсальный источник питания, который позволяет одновременно получить три положительных напряжения питания

Принципиальная схема источника питания

Источник питания, принципиальная схема которого представлена на рис. П.1.6, выдает одновременно три положительных питающих напряжения.

В настоящее время в продаже есть отдельные устройства, но они, чаще всего, низкого качества, не дают никакой защиты, например от короткого замыкания по выходу, и выдают только одно значение напряжения. Чтобы иметь источник питания, эквивалентный нашему, необходимо заплатить от 50 до 100 евро. Эта сумма значительно больше его реальной стоимости. Поэтому мы рекомендуем изготовить его самостоятельно.

На выходе данного устройства присутствуют фиксированные значения напряжений питания +12 и +5 В и регулируемое с помощью потенциометра напряжения питания от 1,25 до 15 В. Каждый из трех источников питания рассчитан на максимальное значение выходного тока от 0,6 до 0,8 А.

Напряжение с понижающего трансформатора 220/12 В, подается на двухполупериодный выпрямитель и далее на фильтр. С1. После фильтрации напряжения двухполупериодного

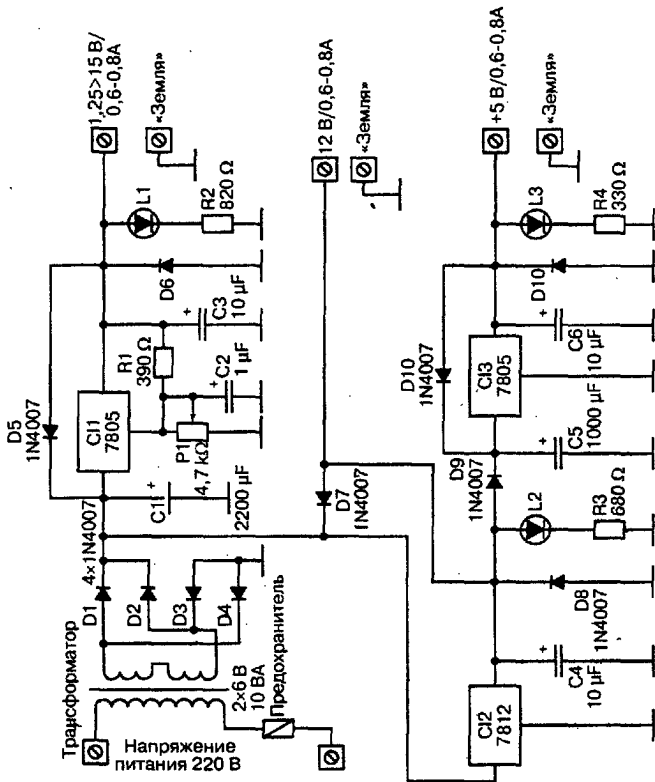


Рис. П. 1.6. Принципиальная схема универсального источника питания

выпрямителя, оно поступает на входы двух стабилизаторов напряжения: регулируемого стабилизатора напряжения на C11 и вход постоянного стабилизатора напряжения на 12 В (C12). Первый выдает напряжение в диапазоне от 1,25 до 15 В, значение которого регулируется потенциометром P1. Второй выдает фиксированное значение напряжения 12 В. Это напряжение поступает на последний (третий) стабилизатор C13, который выдает напряжение 5 В. Диоды D5–D11 предназначены для защиты стабилизаторов от переплюсовки напряжений. На каждом выходе стабилизаторов стоит по фильтрующему конденсатору, также каждый стабилизатор имеет защиту по выходу от короткого замыкания. Три светодиода являются индикаторами напряжения на выходах стабилизаторов.

Реализация

На рис. П.1.7 представлена схема печатной платы в масштабе 1:1. Работа по сборке осуществляется в соответствии со схемой монтажной платы, показанной на рис. П.1.8. Все стабилизаторы прикручиваются винтами на радиаторы, защищающие их от перегрева.

Их нагрев является нормальным, если при повышенном токе выходное напряжение занижено. Не пренебрегайте использованием радиаторов.

Список компонентов:

Резисторы с 5-процентным допуском:

- R1: 390 Ом (оранжевый, белый, коричневый);
- R2: 820 Ом (серый, красный, коричневый);
- R3: 680 Ом (синий, серый, коричневый);
- R4: 330 Ом (оранжевый, оранжевый, коричневый).

Конденсаторы:

- C1: 2200 мкФ/25 В (полярный танталовый или электролитический с аксиальными выводами);
- C2: 1 мкФ/25 В (полярный танталовый или электролитический с радиальными выводами);
- C3; C4; C6: 10 мкФ/25 В (полярный танталовый или электролитический с радиальными выводами);
- C5: 1000 мкФ/25 В (полярный танталовый или электролитический с аксиальными выводами).

Полупроводники:

- CI1: LM317;
- CI2: 7812;
- CI3: 7805;
- D1-D11: 1N4007;
- L1-L3: светодиод, диаметр 5 мм (цвет по выбору, обычно зеленый или красный).

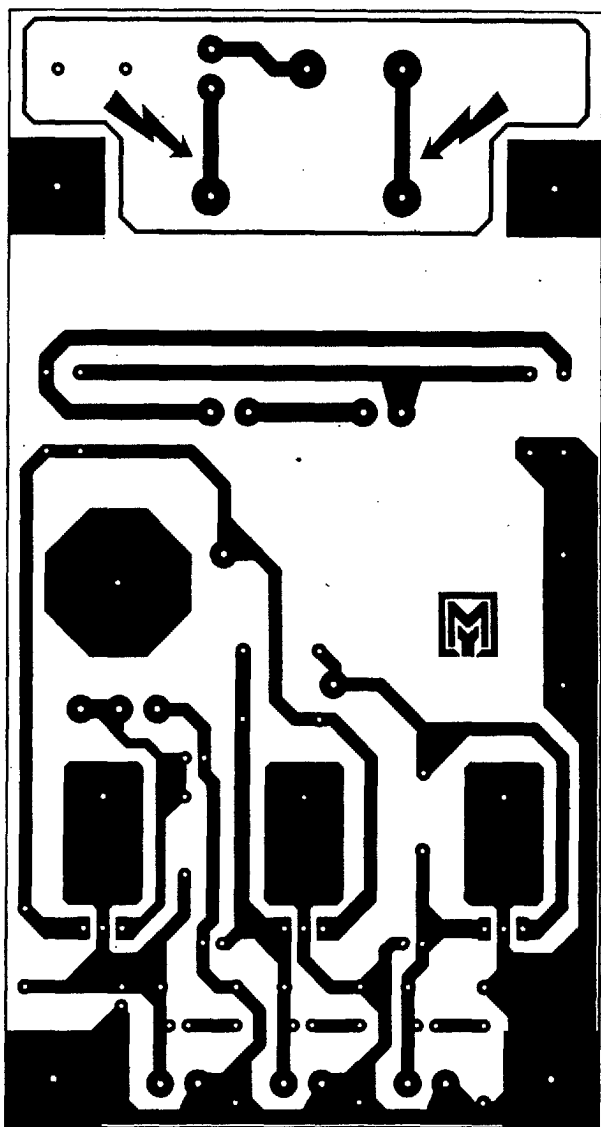


Рис. П. 1.7. Печатная схема универсального источника питания

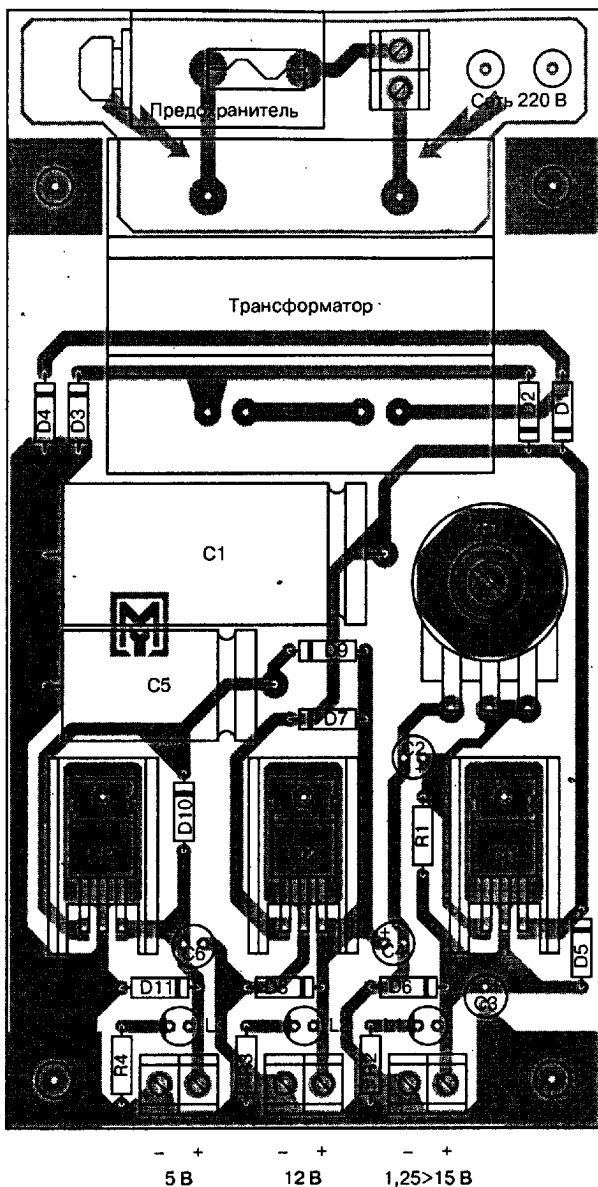


Рис. П. 1.8. Монтажная схема универсального источника питания

Прочие элементы:

- 1 трансформатор 2Ч6 В/10ВА;
- 1 предохранитель 0,5–1 А;
- держатель предохранителя;
- штыревые разъемы с шагом 5,08;
- 3 радиатора для ТО220;
- набор винтов диаметром 3 мм.



Часть печатной схемы подвержена действию напряжения питания электросети из-за установки на ней силового трансформатора. Этот участок платы представляет большую опасность.

Поэтому настоятельно советуем и даже считаем обязательным в месте установки трансформатора на плату прикрутить изолирующую пластинку под печатную плату из специального материала (плексиглас, пластик и др.)*.

* Лучше не выводить печатные проводники с напряжением 220 В на печатную плату, а провести их навесной монтаж силовыми проводами большого сечения. – *Прим. науч. ред.*

Приложение **2**

Технология изготовления печатных плат методом литографии

В этой книге содержится большое количество печатных плат, которые вы должны изготовить своими руками. Многие из вас имеют свой собственный метод изготовления, каждый из которых, возможно, имеет свои достоинства. Несмотря на это, мы опишем здесь наш метод, дополненный различными ухищрениями, которые помогут сделать этот процесс более приятным.

1. Сделайте фотошаблон печатной платы в зеркальном изображении, чтобы получить рисунок платы со стороны элементов. Схемы, сделанные в данной книге, нарисованы со стороны дорожек. Также возможно загрузить файлы со схемами с сайта издателя: <http://www.dunod.com>.
2. Затем проэкспонируйте чувствительную медную пластинку, предварительно удалив пластиковый защитный слой. Если у вас нет крепежной рамки для экспонирования, то можете легко изготовить ее сами из деревянной коробочки подходящего размера. Две электрических лампы, мощность каждой из которых 18 Вт, должны располагаться на расстоянии 4 см от стекла толщиной не менее 4 мм. Отражатель ламп может быть изготовлен из алюминиевой бумаги, однако идеальным решением будет все-таки применение кусков зеркала или купить готовый в магазине электротоваров.

3. Лист фотошаблона расположите на стекле, медная пластинка помещается под ним таким образом, чтобы медная сторона соприкасалась с печатным рисунком фотошаблона. Вся эта конструкция накрывается сверху специальной крышкой с вставкой из губки. Время экспонирования составляет около 13 минут.
4. После этого технология изготовления идентична, какой бы ни была используемая лампа. Этот метод имеет преимущество в том, что при длительном экспонировании не сильно ухудшается качество литографии, так как рисунок менее прозрачен. Автор книги гарантирует отличное качество изготовленных плат, если рисунок имеет хорошую контрастность и четкость и нанесен на бумагу с плотностью 80–90 г.
5. После экспонирования плата помещается в ванну с проявителем при температуре 20 °С (гранулированный порошок, разбавленный в 1 л. воды). Время проявления составляет 1–2 минуты. В течение этого времени рисунок на плате будет постепенно вырисовываться. Прежде чем ополаскивать плату убедитесь, чтобы фоторезист полностью ушел с проявленных мест, освободив медь. При необходимости периодически протирайте плату руками в специальных защитных перчатках.
6. После обильного ополаскивания платы водой можно переходить к операции травления. Наиболее распространенным травителем для меди является трехвалентный хлорид железа. Он продается в шариках и растворяется в одном литре воды. В продаже также имеются вертикальные устройства для травления, которые имеют доступную цену. Эти устройства являются наилучшим и надежным решением. Однако если у вас нет возможности приобрести такое устройство, вы можете использовать пластиковую ванночку, в которую аккуратно поместите плату.
7. После выполнения операции травления следует тщательно смыть водой остатки травления. После этого остается только просверлить отверстия для элементов с помощью сверла диаметром 0,8 мм. Некоторые отверстия должны

быть расточены под больший диаметр в зависимости от используемого элемента. При необходимости доработайте форму платы, как, например, в случае платы робота. После этого шкуркой снимите заусенцы с краев платы.

8. Операция запаивания выводов элементов в плату может происходить без снятия оставшейся пленки фоторезиста. Эта пленка дополнительно защищает печатную плату от окисления.

Во время проведения всех работ постарайтесь никогда не смешивать вещества, нейтрализующие друг друга (щелочи и кислоты). Иначе получится неудачный опыт как на уроке химии. Также соблюдайте элементарные правила безопасности (перчатки, очки, халат) для защиты глаз и других частей тела. Приобретя некоторый опыт, вы получите гарантированный результат, используя этот метод.

Приложение 3

Расположение контактов элементов

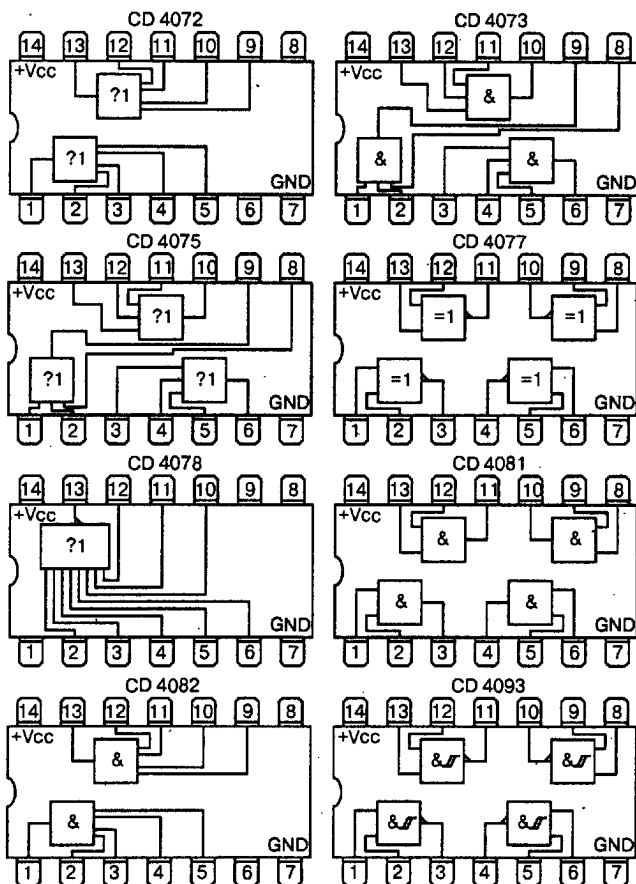


Рис. П.3.1. Расположение контактов элементов, применяющихся в книге (1)

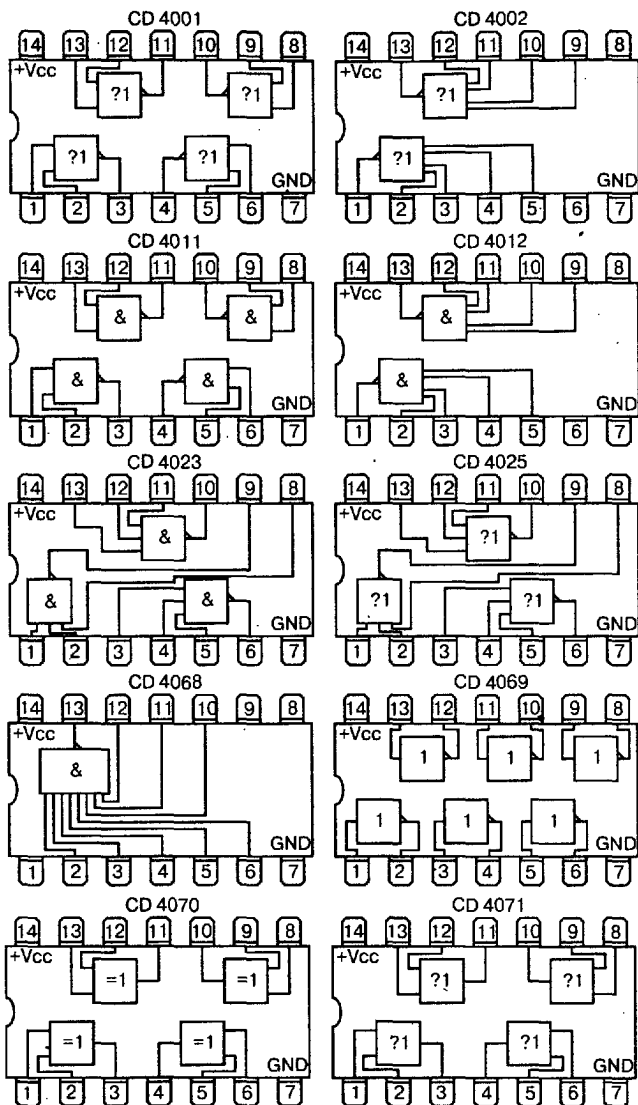


Рис. П.3.1. (Продолжение) Расположение контактов элементов, применяющихся в книге (1)

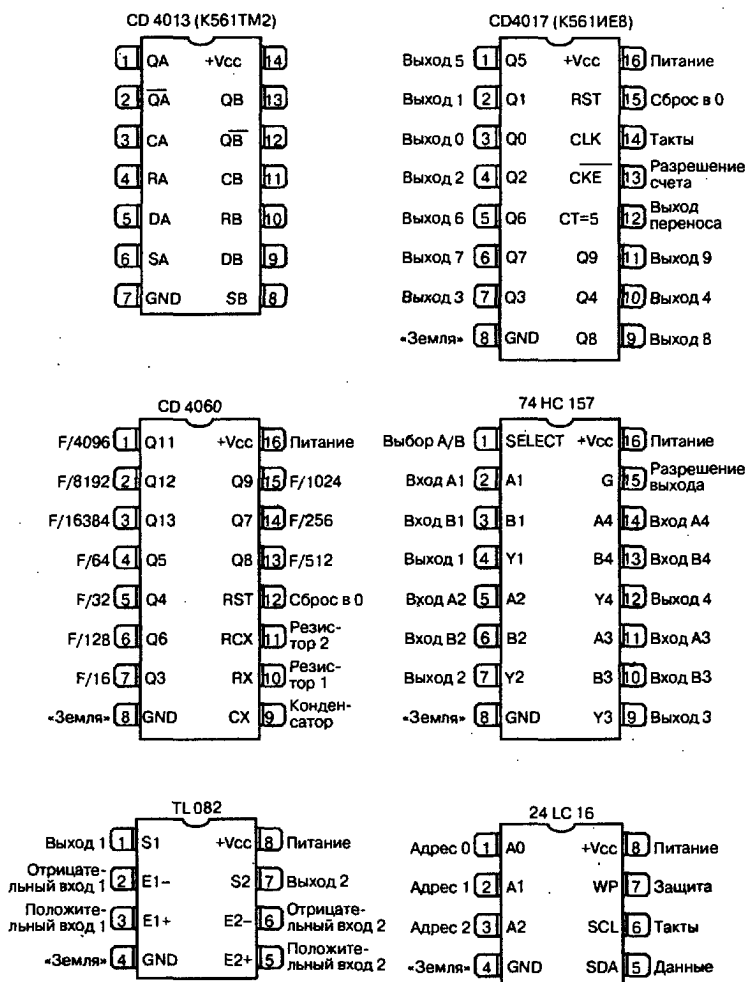


Рис. П.3.2. Расположение контактов элементов, применяющихся в книге (2)

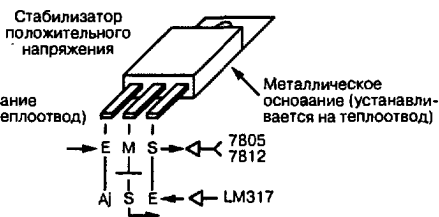
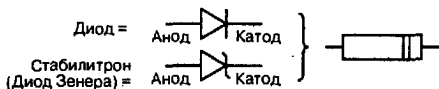
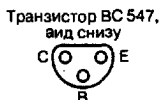
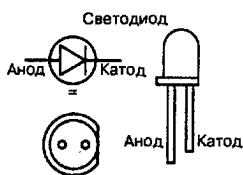
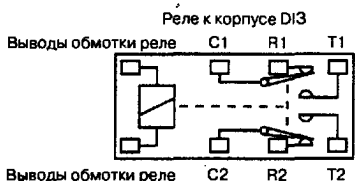


Рис. П.3.2. (Продолжение) Расположение контактов элементов, применяющихся в книге (2)

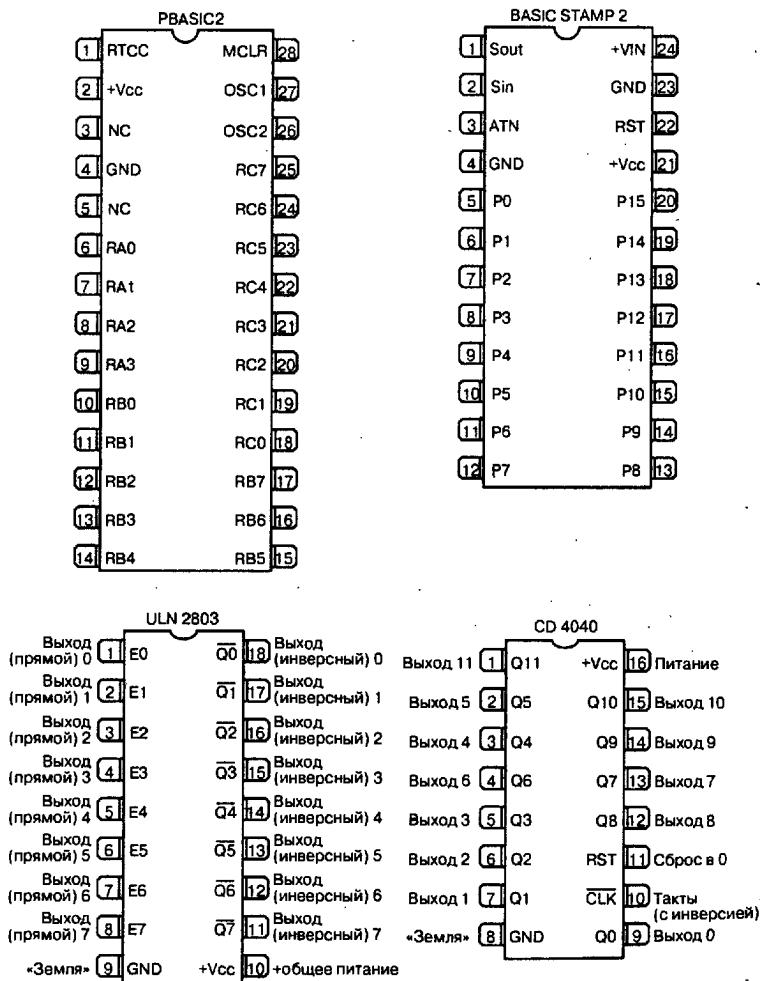


Рис. П.3.3. Расположение контактов элементов, применяющихся в книге (3)

Справочные материалы

Библиография

Все ссылки, приведенные ниже, имеют тесную связь с этой книгой. В них имеется дополнительная информация, позволяющая углубить свои знания по этой теме.

1. DITTRICH S. – Le grand livre de QBASIC. PARIS, Micro Application, 1991.

Эта внушительная по размерам книга, которую сегодня уже трудно найти в продаже, является настоящей библией самого известного языка программирования микроконтроллеров QBASIC. После глубокого изучения эта книга поможет вам написать сложные программы на самом простейшем компьютере.

2. WALLERICH P. – Application CMOS. PARIS, coll. ETSF, Dunod, 1998.

Эта книга также является настоящей библией, но уже в области логических элементов, и особенно элементов проводной логики. В книге вы найдете полезные приложения.

3. GUEULLE P. – Circuits imprimés – Conception et réalisation, 2 édition. PARIS, coll. ETSF, Dunod, 1987.

В книге подробно описана технология изготовления печатных плат. Автор книги дает читателям ряд ценных советов по их изготовлению.

4. TAVERNIER C. – Les Basic Stamp. PARIS, coll. EEA, Dunod, 2000.

5. EDWARDS S. – Programming and customizing the Basic Stamp computer. États-Unis, McGraw-Hill, 1998.

Эти две книги помогут вам овладеть искусством программирования на BASIC STAMP 2. Каждая книга содержит диск с программой. Вторая книга, к сожалению, издается на английском языке*. Однако она содержит привлекательные сведения.

* А может, к счастью. В наших школах изучался больше английский, чем французский язык. – *Прим. науч. ред.*

6. GIAMARCHI F. – Petits robots mobiles – Etude et construction. PARIS, coll. ETSF, Dunod, 2000.
7. JONES J.-L. et FLYNN A.-M. – Les robots mobiles. PARIS, Diderot multimédia, 1997.

Две книги, сведения о роботах в которых хорошо раскрываются авторами. В книгах также рассматриваются всевозможные виды датчиков, что придает книге дополнительную ценность.

Адреса интернет-сайтов

Список интернет-сайтов, который мы вам предлагаем, далеко не полный. Здесь приводятся только те сайты, которые понравились автору больше всего. Вы можете самостоятельно углубиться во Всемирную паутину и найти большое количество полезных ссылок по данной тематике. Используйте поисковые системы, чтобы найти информацию по микросхемам, BS2, робототехнике и многим другим интересным разделам:

- сайт компании Parallax Inc®, изготавливающей микроконтроллеры BASIC STAMP 2: <http://www.parallaxinc.com>;
- сайт, описывающий всевозможные приложения BS2: <http://www.stampsinclass.com>;
- сайт французского издательства этой книги: <http://www.dunod.com>;
- сайт журналов «Практическая электроника», «Интерфейсы PC», «Микророботы»: <http://www.eprat.com>;
- сайт компании Microchip®, производителя микроконтроллеров PIC и его российского представительства: <http://www.microchip.com>, <http://www.microchip.ru>;
- сайт дистрибьютора электронных компонентов: <http://www.selectronic.fr>;
- сайт дистрибьютора электронных компонентов: <http://www.lexctronic.fr>;
- сайт компании Micro Engineering Labs Inc®, продающей компиляторы PIC BASIC, совместимые с BASIC STAMP: <http://www.melbas.com>.

Полезные адреса

Мы хотим подчеркнуть, что список приведенных ниже адресов, безусловно, не является полным. Каждый из вас привык покупать комплектующие у конкретной фирмы. Хотя в данной книге используются только часто применяемые элементы, может оказаться так, что у некоторых фирм случаются сбои в поставках или они не продают некоторые конкретные элементы. Поэтому ниже мы приводим список адресов магазинов*, в которых можно купить всевозможные комплектующие:

- Saint – Quentin Radio
75010, Париж, ул. Saint – Quentin, д.6
Тел.: 0140377074
- ACER Composants
75010, Париж, ул. Chabrol, д.42
Тел.: 0147702831
- Sйlectronic (официальный дистрибьютор BASIC STAMP во Франции)
75011, Париж, пл. Наций, д.11
Тел.: 0155258800
или
59022, Лилль, ул. Chambrai, д.86 BP513
Тел.: 0328550328
- Lextronic
94510, La Queu en Brie, ул. Генерала де Голля, д.36/40
Тел.: 0145768388
- Central Train
75002, Париж, ул. Reaumur, д.81
Тел.: 0142367037
- Motor Model
93100, Montreuil, ул. Робеспьера, д.95
Тел.: 0148515115

* Радиокomпоненты в нашей стране можно купить на радиорынках, магазинах «Чип и Дип», «Золотой шар» и в других местах. Можно также их найти по русскоязычным поисковым системам Rambler и Yandex. Во Францию за радиокomпонентами ездить не нужно, разве что на Париж посмотреть и на парижан с парижанками. – Прим. науч. ред.

Предметный указатель

Д

- Дополнение 13
 - до 1 13
 - до 2 14

И

- Инвертор 24

К

- Код
 - 8421 12
 - двоичный 12
 - дополнительный 14
 - обратный 13
 - прямой 12

П

- Повторитель
 - буферный 26

С

- Семейства
 - схем
 - логических 19
- Система
 - счисления
 - восьмеричная 17
 - шестнадцатиричная 16
- Система счисления
 - двоичная 11

десятичная 10

- Схемы
 - на жесткой логике 19

Т

- Тетрада 16
- Технология КМОП-микросхем 20
- Технология ТТЛ-микросхем 20
- Триггер
 - Шмитта 38

У

- Уровень
 - логический 21

Э

- Элемент
 - логический
 - ДА 26
 - И 31
 - И-НЕ 33
 - ИЛИ 27
 - ИЛИ-НЕ 29
 - ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ 36, 37
 - НЕ 24

Иностранные термины

- D-триггер 4-6

Издательство «НТ Пресс» представляет

Книга начинающего радиолюбителя

Серия «В помощь радиолюбителю»



Автор: Никитин В. А.
Формат: 84×108 1/32
Объем: 384 с.
ISBN: 5-477-00078-3

Популярное описание основ элементарной электротехники, радиотехники и электроники, с самых азов и до современного состояния техники.

Для начинающих радиолюбителей, желающих понять, что происходит в схеме аппарата, который они хотят собрать или пытаются отремонтировать. Большую помощь книга может оказать учащимся школ, техникумов и училищ в процессе изучения соответствующих разделов физики.

Самоучитель по радиоэлектронике

Серия «В помощь радиолюбителю»



Автор: Николаенко М. Н.
Формат: 84×108 ¹/₃₂
Объем: 224 с.
ISBN: 5-477-00054-6

Всем известно, что проектирование и создание электронных устройств требует хотя бы минимальных знаний основ электроники. Но когда дело доходит до практической реализации, оказывается, что каких-то тонкостей вы не знаете. Ведь в процессе изучения теории мало кто обращает внимание на все нюансы использования радиоэлементов и схем.

Как правильно выбрать нужный компонент? Как подобрать оптимальное схемотехническое решение? Как самому разработать и изготовить печатную плату? Как грамотно использовать измерительные приборы при тестировании схемы? Как быстро устранить неисправность? Ответы на эти и другие вопросы вы найдете на страницах книги, представляющей собой самоучитель с практическими рекомендациями и советами по проектированию, изготовлению и наладке аналоговых и цифровых электронных устройств различного назначения.

Книга рассчитана на читателя с техническим складом ума и адресована всем тем, кто хочет научиться паять, читать схемы, конструировать и ремонтировать различные электронные устройства.

Современные усилители

Серия «В помощь радиолюбителю»

Автор: Баширов С. Р.
Формат: 84×108 ¹/₃₂
Объем: 112 с.
ISBN: 5-477-00045-7



Данный справочник содержит основные электрические параметры, а также стандартные и модифицированные схемы включения интегральных схем тракта современного усилителя: блоки коммутации и индикации, активные фильтры и аудиопроцессоры, регуляторы громкости, эквалайзеры и усилители мощности НЧ. Для всех схем приведены чертежи печатных плат и внешний вид собранных устройств.

Книга будет полезна как радиолюбителям, так и специалистам по ремонту бытовой радиоаппаратуры.

Зарядные устройства

Серия «Автоэлектроника». Выпуск 1



Автор: Ходасевич А. Г.,
Ходасевич Т. И.
Формат: 60×88 1/16
Объем: 192 с.
ISBN: 5-477-00101-1

В первом выпуске справочника по зарядным устройствам представлено огромное количество схем ЗУ, собранных как частным образом, так и в заводских условиях. В книге приведен материал, позволяющий не только доработать простые зарядные устройства, уже находящиеся в эксплуатации, но и изготовить их самостоятельно.

**Вы можете приобрести эти книги
в розницу
в книжных магазинах вашего города**

**По вопросам оптовой покупки книг
издательства «ИТ Пресс»
обращаться по адресу: Москва,
Звездный бульвар, дом 21, 7-й этаж
Тел. 615-43-38, 615-01-01,
615-55-13**

Научно-популярное издание

Ив Мержи

**ПРАКТИЧЕСКОЕ РУКОВОДСТВО
ПО ЛОГИЧЕСКИМ МИКРОСХЕМАМ
И ЦИФРОВОЙ СХЕМОТЕХНИКЕ**

Главный редактор *Захаров И. М.*
zim@ntpress.ru

Научный редактор *Симонов В. Я.*

Ответственный редактор *Тульсанова Е. А.*

Технический редактор *Матвеева В. И.*

Верстка *Белова Д. А.*

Графика *Волкова Е. В.*

Дизайн обложки *Краснопирка В. А.*

Издательство «НТ Пресс», 129085, Москва,
Звездный б-р, д. 21, стр. 1.

Издание осуществлено при техническом участии
ООО «Издательство АСТ»

ОАО «Владимирская книжная типография»
600000, г. Владимир, Октябрьский проспект, д. 7.

Качество печати соответствует
качеству предоставленных диапозитивов