Применение 7–сегментных ЖКИ–модулей

Практически любое микроконтроллерное устройство имеет те или иные средства индикации. В простейшем случае это всего несколько светодиодов, а порой — цветной графический дисплей. Появление модулей ЖКИ со встроенными контроллерами значительно упростило схемы сопряжения. Наиболее универсальными из таких модулей являются матричные алфавитно-цифровые, которые позволяют отображать цифры, буквы латинского и русского алфавита и даже псевдографику. Однако такие ЖКИ-модули довольно дороги, они не отличаются малым энергопотреблением, да и в ряде устройств просто избыточны. Там, где не требуются устройства индикации, обладающие широкими возможностями, более подходящими могут оказаться 7-сегментные ЖКИ-модули.

Среди 7-сегментных ЖКИ-модулей большое распространение получили модули на основе контроллера HT1611 (или HT1613). Они имеют 10 знакомест и управляются по последовательной шине (рис. 1). Кроме отображения информации, передаваемой в модуль по шине, он может работать автономно в режиме часов реального времени. Для этого модуль имеет кварцевый резонатор (рис. 2) и выводы для подключения кнопок установки времени. Кроме того, возможна работа модуля в режиме таймера. Низкое напряжение питания (1,5 В) и малый ток потребления (не более 10 мкА) легко позволяют организовать резервное питание. Однако то, что встроенные часы никак не связаны с последовательным интерфейсом, очень ограничивает их использование. Что поделаешь, индикатор в основном предназначен для телефонных аппаратов и оптимизирован именно для такого применения. Кратко остановимся на описании модуля, подробное описание его работы можно получить из документации на микросхему контроллера **(http://www.holtek.com.tw/pdf/comm/1611c.PDF).**

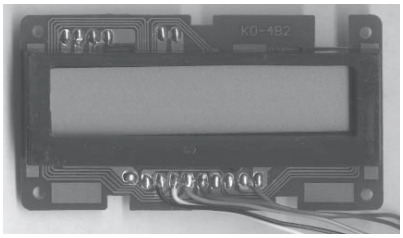

Рис. 1. Внешний вид ЖКИ-модуля на основе контроллера HT1611

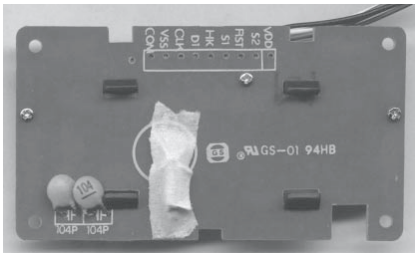

Рис. 2. На плате модуля под липкой бумажной лентой установлен кварцевый резонатор

Назначение выводов модуля показано в табл. 1.

Режимом работы модуля управляет сигнал НК, который в телефонном аппарате соответствует сигналу поднятия трубки. Когда НК=1, модуль находится в режиме отображения реального времени. Если установить НК=0, то модуль переходит в режим таймера, начиная отсчет с нуля. Максимальный интервал, отсчитываемый таймером — 59 мин. 59 с. Если в режиме таймера по шине приходят символы, индикатор очищается, и они начинают отображаться с крайней правой позиции. Новый поступивший символ вызывает сдвиг имеющихся символов влево. Если более 10 секунд новые символы не поступают, модуль снова переходит в режим таймера, начиная отсчет с нуля. Если во время отображения поступивших по шине символов установить НК=1, то модуль вернется в режим отображения реального времени. Если же установить НК=1 во время индикации значения таймера, то возврат в режим отображения реального времени произойдет с задержкой в 5 секунд (рис. 3).

Таблица 1

Номер вывода	Название вывода	Функция
1	12/24	переключение формата времени
2	Vss	общий
3	SK	тактовая линия шины
4	DI	линия данных шины
5	HK	переключение часы/индикатор
6	S1	установка времени
7	S2	выбор режима установки времени
8	TMR	сброс таймера
9	Vdd	напряжение питания

Примечание: выводы 1, 6, 7, 8 имеют внутреннее подключение к общему проводу через резисторы примерно 5 МОм. Выводы 3, 4, 5 имеют внутреннее подключение к выводу питания через резисторы примерно 1 МОм.

К выводам модуля S1 и S2 могут быть подключены кнопки для установки текущего времени. Когда модуль находится в режиме отображения реального времени, нажатие кнопки S2 позволяет перейти в режим установки часов и минут (остальные цифры гаснут). При этом кнопкой S1 можно на-

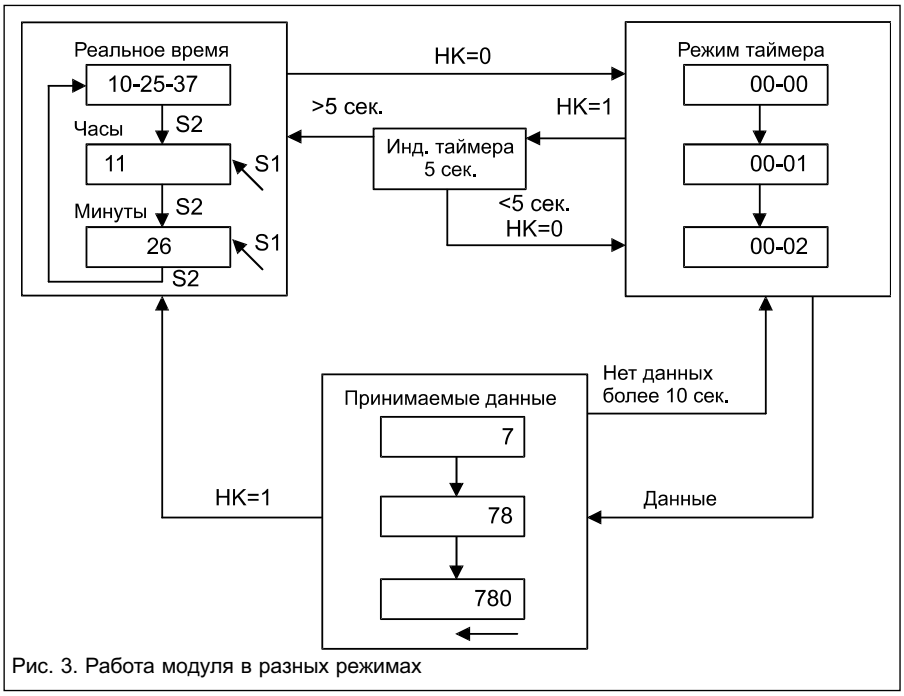

Рис. 3. Работа модуля в разных режимах

брать требуемое значение. Затем с помощью кнопки S2 можно снова вернуться в режим отображения реального времени.

К выводу TMR может быть подключена кнопка сброса таймера. Когда модуль находится в режиме индикации значения таймера, первое нажатие этой кнопки сбрасывает таймер, и отсчет начинается с нуля. Повторное нажатие останавливает таймер (режим удержания показаний). Следующее нажатие снова запустит таймер с нулевого значения (рис. 4).

Когда модуль находится в режиме отображения реального времени, нажатие кнопки TMR переводит его в режим секундомера (рис. 5). Как и в режиме таймера, кнопкой TMR можно сбросить его значение, запустив счет с нуля, или включить режим удержания показаний. Если модуль в течении пяти секунд находится в режиме удержания показаний,

Рис. 4. Действия кнопки TMR в режиме таймера

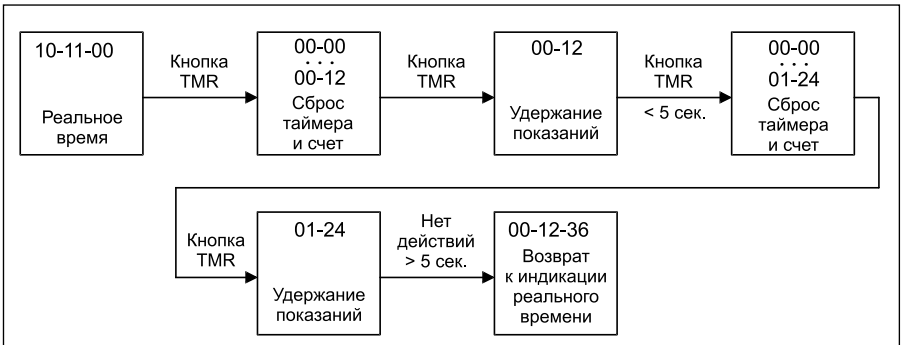

Рис. 5. Действия кнопки TMR в режиме часов реального времени

то происходит автоматический переход в режим отображения реального времени. Все кнопки подключаются между соответствующими выводами модуля и выводом питания.

При использовании модуля в микроконтроллерной системе только для отображения загружаемых по последовательной шине символов, требуется соединить вывод НК с общим проводом, а выводы 12/24, S1, S2 и TMR оставить свободными. Временная диаграмма передачи данных по последовательной шине приведена на рис. 6, где t_a — время установки данных (более 1 мкс), t_b — время удержания данных (более 2 мкс), t_c — интервал между символами (более 5 мкс). Данные подаются на линию DI и защелкиваются по спаду тактовых импульсов на линии SK. Символы отображаются в крайней правой позиции, уже имеющиеся на

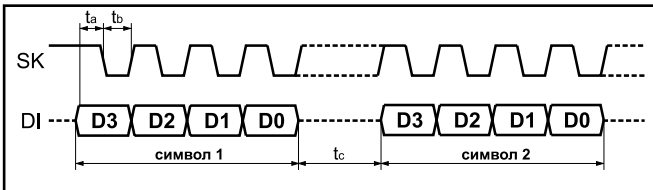

Рис. 6. Временная диаграмма передачи данных по последовательной шине

индикаторе символы сдвигаются влево. После того, как все необходимые данные переданы, линию SK следует оставить в состоянии низкого логического уровня, чтобы предотвратить автоматический переход модуля в режим отображения значения таймера.

Каждый символ кодируется четырьмя битами, поэтому всего имеется 16 символов. Таблица знакогенератора приведена на рис. 7.

Нужно отметить, что напряжение питания индикатора сильно влияет на контрастность. При низком напряжении контрастность недостаточна, а при большом засвечиваются погашенные сегменты. Оптимальная контрастность может быть достигнута при напряжении питания 1,5...1,65 В. Распространенная схема питания, где в качестве источника образцового напряжения используются диоды в прямом включении (рис. 8, а), не позволяет получить оптимальную контрастность,

D3	D2	D1	D0	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7

D3	D2	D1	D0	
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	E
1	1	1	1	F

Рис. 7. Таблица знакогенератора модуля

так как двух диодов оказывается мало, а трех — много. Более того, желательно иметь возможность регулировки этого напряжения. Простая схема на одном транзисторе позволяет получить нужное напряжение питания и регулировать его (рис. 8, b). Учитывая очень низкий ток потребления индикатора можно обойтись и простым резисторным делителем, если входное напряжение питания постоянно. Описанные схемы питания не являются экономичными и подходят, например, для устройств с сетевым пи-

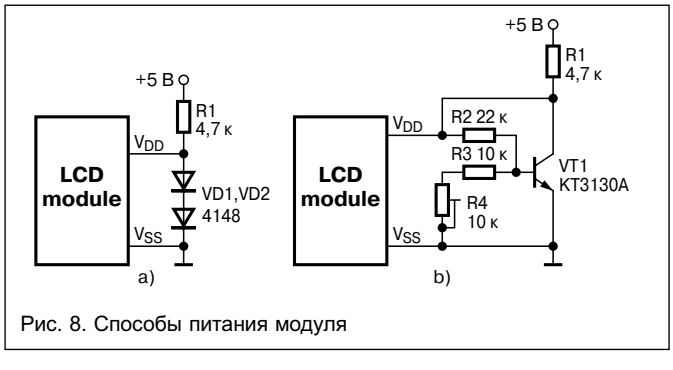

Рис. 8. Способы питания модуля

танием. Система питания автономного устройства может быть очень сложной, и конкретные решения зависят от специфики задачи. Одним из вариантов может быть питание устройства от элемента напряжением 1,5 В, от которого индикатор питается непосредственно. Микроконтроллерная часть устройства питается от того же элемента через повышающий DC/DC преобразователь.

Для согласования логических уровней можно использовать разные схемы. Учитывая тот факт, что входы DI и SK имеют внутренние подтягивающие резисторы, можно обойтись просто диодами (рис. 9, а). Преимущество такого способа заклю-

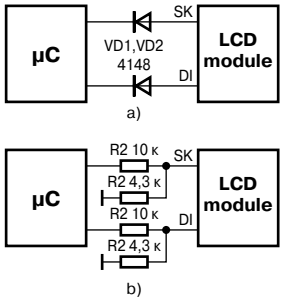

Рис. 9. Согласование логических уровней

чается в том, что согласование не будет зависеть от напряжения питания микроконтроллера. Однако такой способ имеет и недостаток. Ввиду больших номиналов подтягивающих резисторов уровни на входах будут довольно медленно достигать состояния лог. 1, что потребует значительного снижения скорости обмена. Поэтому предпочтительнее для согласования использовать резисторные делители (рис. 9, b).

С программной точки зрения работа с индикатором очень проста. Ниже приведен набор подпрограмм для микроконтроллеров семейства AVR:

```
;Определение регистров:

.def  temp   =r16          ;временный регистр temp
.def  Cnt    =r17          ;временный регистр Cnt
.def  Del    =r18          ;временный регистр Del

;Определение портов:

.equ DIRD   = 0b00000011  ;направление для порта D
.equ PUPD   = 0b01111100  ;pull-ups для порта D

.equ SD     = PD0         ;линия данных DI
.equ SC     = PD1         ;тактовая линия SK

;Инициализация:

INIT:ldi    temp,RAMEND;инициализация стека
     out    SPL,temp

     ldi    temp,PUPD
     out    PORTD,temp    ;инициализация порта D
     ldi    temp,DIRD
     out    DDRD,temp     ;задание направления порта D

;Основная программа:

     ldi    temp,3
     rcall  LCD           ;вывод на индикатор числа 3
     ..... ;продолжение программы

;Подпрограмма вывода символа на индикатор:
;код символа должен находиться в регистре temp

LCD:      ldi   Cnt,4          ;загрузка счетчика
     swap  temp           ;подготовка мл. тетрады к сдвигу

LDL: rol   temp           ;сдвиг
     brcs  LD1
 LD0: cbi   PORTD,SD           ;сброс линии данных, если
бит=0
     rjmp  STR
 LD1: sbi   PORTD,SD           ;установка линии данных,
если бит=1

 STR: sbi   PORTD,SC           ;установка линии тактиро-
вания
     ldi   Del,5
 HNG1:     dec   Del          ;задержка на установку
данных
     brne  HNG1

     cbi   PORTD,SC           ;сброс линии тактирования
     ldi   Del,10
 HNG2:     dec   Del          ;задержка на удержание
данных
     brne  HNG2

     dec   Cnt
     brne  LDL           ;цикл, если не последний бит

     ldi   Del,25
HNG3:     dec   Del          ;межсимвольная задержка
     brne  HNG3
     ret
```

Необходимо отметить, что в течение примерно 2 секунд после включения питания модуль не воспринимает данные, передаваемые ему по последовательной шине. Поэтому всегда должна быть задержка между включением питания и началом обмена.

Описанный модуль на основе контроллера HT1611 обладает рядом существенных недостатков:

- отсутствие децимальных точек, что затрудняет отображение произвольных цифровых величин;
- невозможность управления отдельными сегментами; контроллер имеет фиксированный знакогенератор, содержащий всего 16 символов, в то же время возможности 7-сегментного индикатора значительно шире, он может отображать многие буквы латинского алфавита и специальные символы;
- заметные мерцания изображения при загрузке, вызванные отсутствием буферизации выводимой информации, из-за чего при необходимости обновления изображения с большой частотой создается весьма неприятный зрительный эффект;
- высокая инерционность, ограничивающая применение таких эффектов, как, например, мигающие символы;
- плохой угол обзора, т. к. индикатор оптимизирован для применения в телефонных аппаратах, где он устанавливается на слегка наклонной панели, и при установке индикатора, например, на вертикальную переднюю панель прибора контрастность изображения резко падает;
- отсутствие возможности регулировки контрастности, ведь такую регулировку можно осуществить только изменением напряжения питания, что не всегда удобно;
- отсутствие встроенной подсветки, т. к. место для ее установки не предусмотрено;
- требуются дополнительные элементы для получения напряжения питания 1,5 В и для согласования логических уровней, т. к. напряжение питания микропроцессорной системы обычно выше.

Еще одним распространенным типом 7-сементного ЖКИ-модуля со встроенным контроллером является модуль MT10T7-7 (рис. 10) производства компании «Мэлт». Этот модуль лишен практически всех недостатков, перечисленных выше. Наличие децимальных точек и управление отдельными сегментами значительно расширяет возможности модуля. При перезагрузке индикатора новыми данными отсутствуют какие-либо видимые мерцания изображения, инерционность индикатора также заметно ниже. Имеется специальный вывод регулировки контрастности. При желании можно установить элементы подсветки, для чего на плате имеется соответствующая разводка, а торцы стеклянных пластин индикатора открыты. Напряжение питания модуля составляет 5 В, что соответствует напряжению питания большинства микроконтроллерных устройств. Это исключает необходимость специального согласования логических уровней. Потребляемый индикатором ток очень мал и составляет примерно 30 мкА.

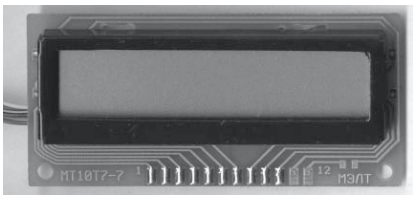

Рис. 10. Внешний вид ЖКИ-модуля MT10T7-7 компании «Мэлт».

В то же время некоторые недостатки модуль все же имеет. Один из

них — плохой угол обзора. Этот модуль также оптимизирован для установки на слегка наклонную (почти горизонтальную) панель, что хорошо для телефонного аппарата, но не подходит, например, для измерительного прибора с вертикальной передней панелью. Вообще, при производстве индикатора оптимальный угол зрения может быть сделан любым. В номенклатуре ЖКИ зарубежных фирм почти для любого типа индикатора имеются разные версии, оптимизированные для разных углов зрения. У MT10T7 таких версий нет, и это можно объяснить тем, что основным потребителем таких индикаторов являются изготовители телефонных аппаратов с АОН, под чьи нужды и оптимизирован индикатор.

Второй недостаток модуля MT10T7 заключается в том, что он имеет параллельную шину управления. Шина содержит 4 разряда данных, 1 разряд адреса и 2 сигнала стробирования. При подключении к микроконтроллеру индикатор требует как минимум 6 линий вывода (модуль на основе HT1611 требует всего 2 линии). Учитывая тот факт, что высокого быстродействия при обмене с индикатором обычно не требуется, параллельную шину нельзя назвать оптимальной.

Простая доработка позволяет оснастить указанный модуль последовательной шиной управления. Описание самого модуля здесь приводить нет смысла, так как на сайте производителя имеется довольно хороший файл с документацией (**http://www.melt.aha.ru/pdf/mt-10t7-7.pdf**), а в журнале «Схемотехника», №2/2000, уже печатался материал на эту тему. Поэтому здесь будет рассмотрена только доработка.

Самым универсальным способом преобразования последовательного формата в параллельный является использование сдвигового регистра с буфером хранения, например, 74HC595. Схема подключения такого сдвигового регистра к модулю показана на рис. 11. При этом модуль приобретает шину, совместимую с SPI. Для того чтобы загрузить индикатор, нужно сформировать строб записи. Поэтому потребуются две последовательные загрузки регистра одними и теми же битами данных и с противоположными значениями сигнала записи. Это несколько замедляет загрузку, но, учитывая, что сдвиговый регистр имеет высокое быстродействие, загружать его можно с высокой скоростью, не используя программных задержек. Ниже приведен набор подпрограмм загрузки индикатора для микроконтроллеров семейства MCS-51:

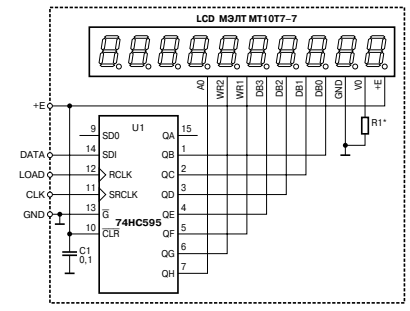

Рис. 11. Подключение регистра 74HC595 к модулю MT10T7-7

```
;Определение портов:

CLK .EQU P1.0       ;линия clock
LOAD        .EQU P1.1       ;линия load
DATA        .EQU P1.2       ;линия data

;Основная программа:

MAIN:       LCALL CLEAR;очистка индикатора

        MOV A,#00H
        LCALL WR_A  ;загрузка адреса первого знакоместа
        MOV A,#3
        LCALL WR_S  ;отображение числа 3
        .....       ;продолжение программы

;Подпрограммы:

;Очистка индикатора:
CLEAR:      MOV A,#0FH  ;адрес регистра BLK
        LCALL WR_A  ;запись адреса
        MOV A,#0FH  ;код разрешения шины
        LCALL WR_N  ;запись тетрады

        MOV A,#00H  ;адрес знакоместа SG1
        LCALL WR_A  ;запись адреса
        MOV R0,#10  ;загрузка счетчика
CL1: CLR A
        LCALL WR_D  ;очистка знакомест SG1 – SG10
        DJNZ R0,CL1
        RET

;Вывод символа на индикатор:
;A – код символа (ACC.7 – точка)

WR_S:       MOV DPTR,#FONT   ;загрузка адреса начала
знакогенератора
        MOV C,ACC.7 ;сохранение признака точки
        CLR ACC.7
        MOVC A,@A+DPTR  ;перекодировка
        MOV H,C             ;восстановление признака точки
        LCALL WR_D  ;запись данных
        RET

;Запись байта данных:
;A – байт данных

WR_D:       PUSH ACC            ;сохранение байта данных
        LCALL WR_N  ;запись младшей тетрады
        POP  ACC            ;восстановление байта данных
        SWAP A              ;обмен тетрадами
        LCALL WR_N  ;запись старшей тетрады
        RET

;Запись тетрады:
;A – тетрада (00H..0FH)

WR_N:       RL A
        ANL A,#1EH
        ORL A,#80H  ;A=1, WR=0
        PUSH ACC
        ORL A,#20H  ;A=1, WR=1
        LCALL WR595 ;загрузка регистра 74HC595
        POP  ACC
        LCALL WR595 ;загрузка регистра 74HC595
        RET

;Запись адреса:
;A – адрес (00H..0FH)

WR_A:       RL A
        ANL A,#1EH  ;A=0, WR=0
        PUSH ACC
        ORL A,#20H  ;A=0, WR=1
        LCALL WR595 ;загрузка регистра 74HC595
        POP  ACC
        LCALL WR595 ;загрузка регистра 74HC595
        RET

;Загрузка регистра 74HC595:
;A – данные

WR595:      PUSH B
        MOV B,#8            ;загрузка счетчика
        CLR LOAD            ;LOAD=0
WR1:        CLR CLK             ;CLK=0
        RLC A           ;сдвиг бита данных в C
        MOV DATA,C
        SETB CLK            ;CLK=1
        DJNZ B,WR1
        SETB LOAD   ;LOAD=1
        POP B
        RET

;Таблица знакогенератора:
```

```
;       FCBHADEG
FONT    .DB 11101110B ;code 00H, character 0
        .DB 01100000B ;code 01H, character 1
        .DB 00101111B ;code 02H, character 2
        .DB 01101101B ;code 03H, character 3
        .DB 11100001B ;code 04H, character 4
        .DB 11001101B ;code 05H, character 5
        .DB 11001111B ;code 06H, character 6
        .DB 01101000B ;code 07H, character 7
        .DB 11101111B ;code 08H, character 8
        .DB 11101101B ;code 09H, character 9
        .DB 11101011B ;code 0AH, character A
        .DB 11000111B ;code 0BH, character b
        .DB 10001110B ;code 0CH, character C
        .DB 01100111B ;code 0DH, character d
        .DB 10001111B ;code 0EH, character E
        .DB 10001011B ;code 0FH, character F
        .DB 00000000B ;code 10H, character blank
        .DB 00000100B ;code 11H, character _
        .DB 00000001B ;code 12H, character −
        .DB 00001000B ;code 13H, character ~
        .DB 10101001B ;code 14H, character degree
        .DB 00000111B ;code 15H, character c
        .DB 11001110B ;code 16H, character G
        .DB 11100011B ;code 17H, character H
        .DB 01100000B ;code 18H, character I
        .DB 10000110B ;code 19H, character L
        .DB 00000110B ;code 1AH, character l
        .DB 01000011B ;code 1BH, character n
        .DB 01000111B ;code 1CH, character o
        .DB 10101011B ;code 1DH, character P
        .DB 10001010B ;code 1EH, character R
        .DB 00000011B ;code 1FH, character r
        .DB 10000111B ;code 20H, character t
        .DB 11100110B ;code 21H, character U
        .DB 01000110B ;code 22H, character u
        .DB 11100101B ;code 23H, character Y

H       .EQU ACC.4    ;точка
```

Еще удобнее использовать обычный сдвиговый регистр, не имеющий регистра-защелки, например, 74HC164 (или 74HC4094 с учетом разницы в разводке). При этом стробирование осуществляется сигналом WR индикатора. Нужно учесть тот факт, что входы WR индикатора являются потенциальными, это означает, что во время загрузки сдвигового регистра на них нужно удерживать пассивный уровень. И только после того, как загрузка завершена, на этих входах нужно сформировать сигнал стробирования. Поскольку в этом варианте схемы у сдвигового регистра используется только 5 выходов, нет необходимости вдвигать все 8 бит. Все это увеличивает скорость загрузки по сравнению с предыдущим вариантом в 3 раза. Схема второго варианта приведена на рис. 12. С программной точки зрения меняется только несколько подпрограмм. Их текст приведен ниже:

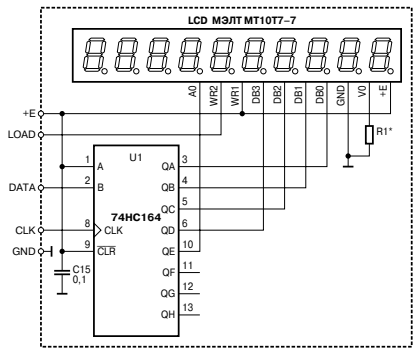

Рис. 12. Подключение регистра 74HC164 к модулю MT10T7-7

```
;Запись тетрады:
;A − тетрада (00H..0FH)

WR_N:     ANL A,#0FH
     ORL A,#10H   ;подготовка 5 битов с A=1
     SWAP A
     RR A
     LCALL WR_5  ;загрузка регистра 74HC164
     RET

;Запись адреса:
;A − адрес (00H..0FH)
WR_A:     ANL A,#0FH
     SWAP A            ;подготовка 5 битов с A=0
     RR A
     LCALL WR_5  ;загрузка регистра 74HC164
     RET

;Загрузка 5 бит в регистр 74HC164 и
;защелкивание данных в модуле:
;A − данные

WR_5:     PUSH B
     MOV B,#5          ;загрузка счетчика
WR_B:     CLR CLK           ;CLK=0
     RLC A        ;сдвиг бита данных в C
     MOV DATA,C
     SETB CLK          ;CLK=1
     DJNZ B,WR_B
     CLR LOAD          ;LOAD=0
     SETB DATA    ;DATA=1
     SETB LOAD    ;LOAD=1
     POP B
     RET
```

Ниже приведен аналогичный набор подпрограмм работы с модулем для микроконтроллеров семейства AVR:

```
;Определение регистров:

.def  temp   =r16          ;временный регистр temp
.def  Cnt    =r17          ;временный регистр Cnt
.def  Del    =r18          ;временный регистр Del

;Определение портов:

.equ DIRB   = 0b00000111  ;направление для порта B
.equ PUPB   = 0b11111111  ;pull−ups для порта B

.equ DATA   =PB0          ;линия data
.equ LOAD   =PB1          ;линия load
.equ CLK    =PB2          ;линия clock

;Инициализация:

INIT:ldi     temp,RAMEND;инициализация стека
     out     SPL,temp

     ldi     temp,PUPB
     out     PORTB,temp   ;инициализация порта B
     ldi     temp,DIRB
     out     DDRB,temp    ;задание направления порта B

;Основная программа:

     rcall   LCD_CL            ;очистка индикатора

     ldi     temp,0x00
     rcall   LCD_WA            ; загрузка адреса первого
знакоместа

     ldi     temp,3
     rcall   LCD_WS            ;отображение числа 3
     .....           ;продолжение программы

;Подпрограммы:

;Очистка индикатора:

LCD_CL:   ldi     temp,0x0F     ;загрузка адреса регистра
BLK
     rcall   LCD_WA            ;запись адреса
     ldi     temp,0x0F    ;код разрешения шины 0x0F
     rcall   LCD_WN            ;запись тетрады
```

```
        ldi     temp,0x00       ;адрес первого знакоместа SG1
        rcall   LCD_WA          ;запись адреса
        ldi     Del,10          ;загрузка счетчика
  cl1:  clr     temp
        rcall   LCD_WD          ;очистка индикатора
        dec     Del
        brne    cl1
        ret

  ;Вывод символа на индикатор:
  ;temp – код символа (temp.7 – точка)

  LCD_WS:   bst     temp,7          ;сохранение признака точки в T
        andi    temp,0x7F       ;temp.7 <– 0
        ldi     ZL,low (FONT*2)         ;начало таблицы знакогенератора
        ldi     ZH,high(FONT*2)
        add     ZL,temp                 ;добавить смещение (Z + temp)
        brcc    wc1
        inc     ZH
  wc1: lpm                      ;чтение таблицы в temp0
        mov     temp,temp0
        bld     temp,H          ;восстановление признака точки
        rcall   LCD_WD          ;запись данных
        ret

  ;Запись данных:

  LCD_WD:   push    temp            ;сохранение данных
        rcall   LCD_WN          ;запись младшей тетрады
        pop     temp            ;восстановление данных
        swap    temp            ;обмен тетрад
        rcall   LCD_WN          ;запись старшей тетрады
        ret

  ;Запись тетрады:

  LCD_WN:   andi    temp,0x0F       ;обнуление неиспользуемых битов
        ori     temp,0x10       ;A=1
        rjmp    wa1

  ;Запись адреса:

  LCD_WA:   andi    temp,0x0F       ; обнуление неиспользуемых битов
  wa1: lsl      temp
        lsl     temp
        lsl     temp
        ldi     Cnt,5           ;загрузка счетчика
  w5:   cbi     PORTB,CLK       ;CLK=0
        rol     temp
        brcs    w51
  w50: cbi      PORTB,DATA      ;DATA=0 или
        rjmp    w52
  w51: sbi      PORTB,DATA      ;DATA=1
  w52: dec      Cnt
        sbi     PORTB,CLK       ;CLK=1
        brne    w5
        cbi     PORTB,LOAD      ;LOAD=0
        sbi     PORTB,DATA
        sbi     PORTB,LOAD      ;LOAD=1
        ret

  ;Таблица знакогенератора:

  FONT:             ;FCBHADEG   FCBHADEG
        .DB 0b11101110, 0b01100000      ;0, 1
        .DB 0b00101111, 0b01101101      ;2, 3
        .DB 0b11100001, 0b11001101      ;4, 5
        .DB 0b11001111, 0b01101000      ;6, 7
        .DB 0b11101111, 0b11101101      ;8, 9
        .DB 0b11101011, 0b11000111      ;A, b
        .DB 0b10001110, 0b01100111      ;C, d
        .DB 0b10001111, 0b10001011      ;E, F
        .DB 0b00000000, 0b00000100      ;blank, _
        .DB 0b00000001, 0b00001000      ;–, ~
        .DB 0b10101001, 0b00000111      ;degree, c
        .DB 0b11001110, 0b11100011      ;G, H
        .DB 0b01100000, 0b10000110      ;I, L
        .DB 0b00000110, 0b01000011      ;l, n
        .DB 0b01000111, 0b10101011      ;o, P
        .DB 0b10001010, 0b00000011      ;R, r
        .DB 0b10000111, 0b11100110      ;t, U
        .DB 0b01000110, 0b11100101      ;u, Y

  .equ H        =4                      ;точка
```

Конструктивно сдвиговый регистр располагается на небольшой односторонней печатной плате, которая закрепляется с обратной стороны индикатора (рис. 13). Поскольку там все равно имеются ушки крепления металлической рамки, установка дополнительной платы не увеличивает габаритов модуля. Выводы платы, которыми она должна соединяться с модулем, выполнены в виде площадок с отверстиями, которые расположены точно напротив соответствующих отверстий в плате модуля. Соединения выполняются перемычками, проходящими сквозь две платы. Если перемычки расклепать, то они будут осуществлять и крепление платы.

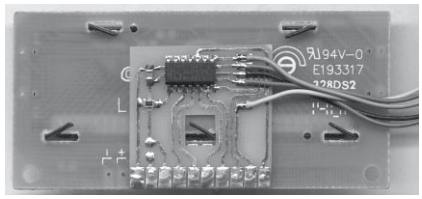

Рис. 13. Установка дополнительной платы на модуль MT10T7-7

Чертежи дополнительной платы для двух вариантов доработки приведены на рис.14.

Очень часто в системе требуется не только индикатор, но и клавиатура управления. Чтобы сэкономить порты микроконтроллера, которых всегда не хватает, для сканирования клавиатуры можно использовать сдвиговый регистр, который применяется для индикатора. В простейшем случае, добавив всего одну входную линию порта микроконтроллера, можно подключить до 8 кнопок. Если их требуется больше, можно организовать матрицу 8xn, где n — число

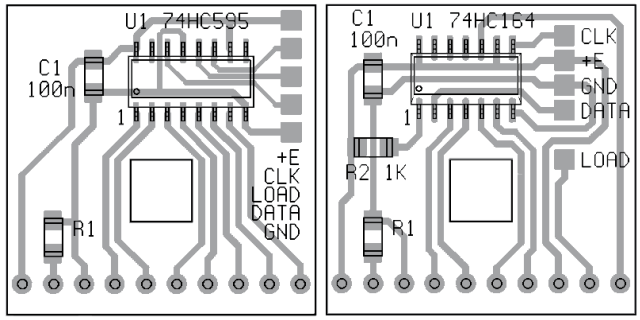

Рис. 14. Чертежи двух вариантов дополнительной платы

линий возврата клавиатуры. Пример подключения 4-х кнопок показан на рис. 15. Кнопки обычно расположены конструктивно вблизи индикатора, поэтому все соединения получаются короткими. Необходимо отметить, что линии возврата клавиатуры требуют внешних подтягивающих резисторов сопротивлением 2,2-4,7 кОм. Иначе уровень лог.1 достигается слишком медленно, что потребует введения дополнительных задержек в подпрограмму сканирования. Программно обрабатываются как одиночные нажатия, так и совместные нажатия кнопок. Совместно можно нажимать любое количество кнопок, все эти комбинации будут иметь индивидуальные скан-коды. Подпрограмма сканирования клавиатуры для микроконтроллеров семейства AVR приведена ниже:

```
;Описание линии возврата клавиатуры:

.equ RETL   =PINB4              ;keyboard return line

;Скан-коды кнопок:

.equ K_NO   =0x00       ;нет нажатия
.equ K_SL   =0x01       ;скан-код кнопки SELECT
.equ K_DN   =0x02       ;скан-код кнопки DOWN
.equ K_UP   =0x04       ;скан-код кнопки UP
.equ K_EN   =0x08       ;скан-код кнопки ENTER

;Сканирование клавиатуры:
;temp - выходной скан-код

SCAN:   ldi     temp,0xF7       ;инициализация
переменной temp
        ldi     Cnt,8           ;счетчик циклов сканирования
sc1:    cbi     PORTB,CLK       ;CLK=0
        rol     temp            ;C <- temp.7..temp.0 <- C
        brcs    sc11
sc10:   cbi     PORTB,DATA      ;DATA=0 или
        rjmp    sc2
sc11:   sbi     PORTB,DATA      ;DATA=1
sc2:    dec     Cnt
        sbi     PORTB,CLK       ;CLK=1
        sec
        sbis    PINB,RETL       ;C <- RETL
        clc
        brne    sc1             ;цикл, если сканирование не
завершено
        rol     temp            ;temp.0 < - C, последний бит
скан-кода
        com     temp            ;инвертирование temp
        andi    temp,0x0F       ;temp = скан-код
        ret
```

Несмотря на то, что на выходе получается 4-разрядный скан-код, циклов сканирования всего 8. Первых 4 цикла подготавливают сдвиговый регистр, записывая в него единицы. Затем осуществляется сканирование «бегущим нулем». На выходе скан-код инвертируется, чтобы нажатие кнопки давало единицу в соответствующем разряде.

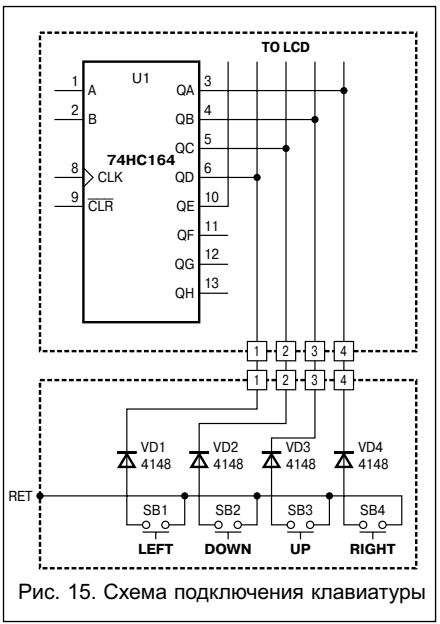

Рис. 15. Схема подключения клавиатуры

Необходимо отметить, что подпрограмма сканирования не производит операцию подавления дребезга. Для подавления дребезга нужно убедится, что в течение 20-30 мс состояние кнопки не изменилось. Только после этого скан-код можно считать действительным. Такое время на обработку нажатий клавиатуры, когда никакие другие процессы выполняться не могут, в некоторых системах может оказаться недопустимо большим. В таких случаях процесс подавления дребезга нужно оформить как одну из задач многозадачного ядра.

Леонид Ридико,
wubblick@yahoo.com