



(/)

Разделы

Новости (/index.php/novosti.html)
Встраиваемые системы (/index.php/embedded-programming.html)
Программирование AVR (/index.php/programming-avr.html)
Программирование ARM (/index.php/programmirovanie-arm.html)
Инструменты/технологии (/index.php/instruments-technologies.html)
Как подключить (/index.php/how-connection.html)
Компоненты (/index.php/electronic-components.html)
RTOS (/index.php/rtos.html)
Софт (/index.php/iar-embedded-workbench.html)
Проекты (/index.php/projects-avr.html)
Ссылки (/index.php/links.html)

1wire (/index.php/programming-avr/itemlist/tag/1wire.html) arm (/index.php/programming-avr/itemlist/tag/arm.html)

avr (/index.php/programming-avr/itemlist/tag/avr.html)

avr программатор (/index.php/programming-avr/itemlist/tag/avr-программатор.html) ds18b20 (/index.php/programming-avr/itemlist/tag/ds18b20.html)

eeprom (/index.php/programming-avr/itemlist/tag/eeprom.html) i2c (/index.php/programming-avr/itemlist/tag/i2c.html)

IAR (/index.php/programming-avr/itemlist/tag/IAR.html) lcd (/index.php/programming-avr/itemlist/tag/lcd.html)

tsop (/index.php/programming-avr/itemlist/tag/tsop.html) twi (/index.php/programming-avr/itemlist/tag/twi.html)

алгоритмы (/index.php/programming-avr/itemlist/tag/алгоритмы.html)

библиотеки (/index.php/programming-avr/itemlist/tag/библиотеки.html) датчик (/index.php/programming-avr/itemlist/tag/датчик.html)

драйвер (/index.php/programming-avr/itemlist/tag/драйвер.html) интерфейс (/index.php/programming-avr/itemlist/tag/интерфейс.html)

компоненты (/index.php/programming-avr/itemlist/tag/компоненты.html) макросы (/index.php/programming-avr/itemlist/tag/макросы.html)

oy (/index.php/programming-avr/itemlist/tag/oy.html)

программирование микроконтроллеров (/index.php/programming-avr/itemlist/tag/программирование-микроконтроллеров.html)

расчет (/index.php/programming-avr/itemlist/tag/расчет.html)

семисегментный индикатор (/index.php/programming-avr/itemlist/tag/семисегментный-индикатор.html) си (/index.php/programming-avr/itemlist/tag/си.html)

событийная система (/index.php/programming-avr/itemlist/tag/событийная-система.html)

схемотехника (/index.php/programming-avr/itemlist/tag/схемотехника.html) таймер (/index.php/programming-avr/itemlist/tag/таймер.html)

управление (/index.php/programming-avr/itemlist/tag/управление.html) устройства (/index.php/programming-avr/itemlist/tag/устройства.html)

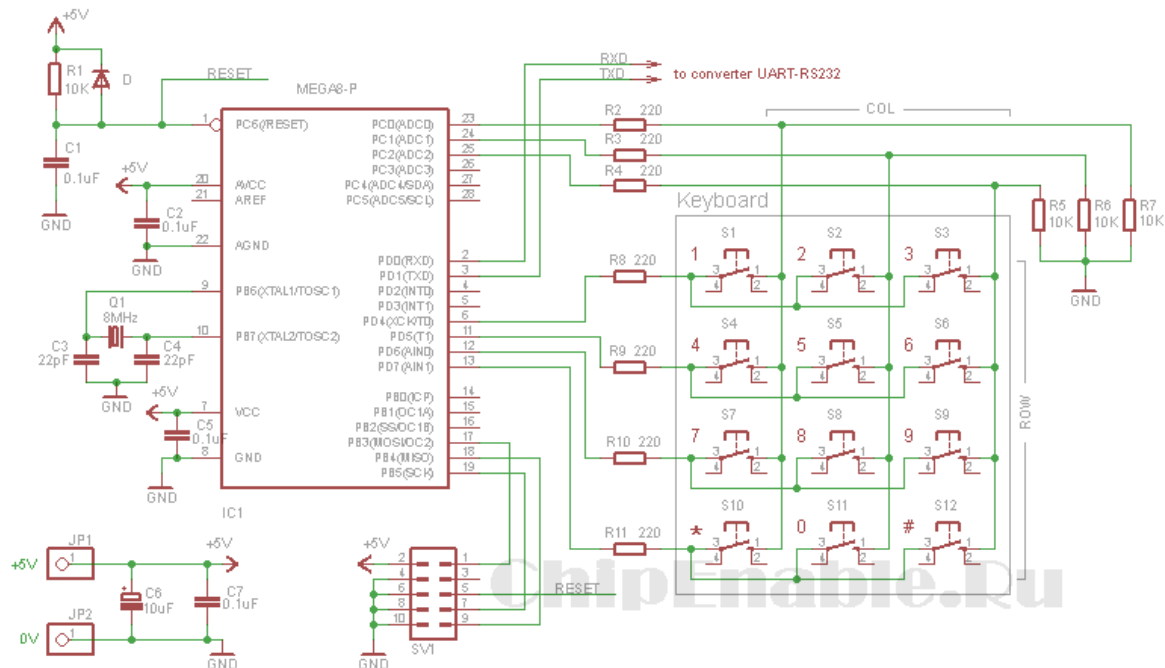
учебный курс avr (/index.php/programming-avr/itemlist/tag/учебный-курс-avr.html)

шим (/index.php/programming-avr/itemlist/tag/шим.html)

Учебный курс. Опрос матричной клавиатуры. Пример использования автомата (State Machine) ☆

14/03/2010 - 14:23 | Pavel Bobkov (/index.php/programming-avr/itemlist/user/800-pavelbobkov.html)

Схема



(/images/stories/articles/c_course/keyboard/keyboard-2.png)

Резисторы R2 – R4, R8 – R11 предназначены для ограничения входного/выходного тока в случае неправильной настройки портов или одновременного нажатия нескольких кнопок. Выводы PD0(RXD), PD1(TXD) подключены к преобразователю UART-RS232, который на схеме не отображен. Обмен по USART`у используется для отладки программы.

Алгоритм опроса матричной клавиатуры

Строки клавиатуры подключены к выводам PD4, PD5, PD6, PD7. Они настроены на выход и в начальном состоянии на этих выводах напряжение логического нуля. Столбцы подключены к выводам PC0, PC1, PC2. Они настроены на вход, внутренние подтягивающие резисторы отключены и эти линии "придавлены к нулю питания" с помощью внешних резисторов номиналом в 10 КОм.

Процедура сканирования клавиатуры выглядит следующим образом. Выставляем 1 на выводе PD4 и проверяем состояние выводов PC0, PC1, PC2 (то есть считываем содержимое регистра PINC). Если на каком-то из выводов установлена 1, значит, на клавиатуре в данный момент нажата кнопка, подключенная к первой строке. Сохраняем биты PD4, PD5, PD6, PD7 и PC0, PC1, PC2 в одной переменной – по этому коды мы будем определять номер нажатой кнопки. Если ни одна из кнопок не нажата, продолжаем процедуру сканирования.

Сбрасываем 1 на выводе PD4 и устанавливаем 1 на выводе PD5. Снова проверяем состояние выводов PC0, PC1, PC2, и в случае нажатия кнопки сохраняем биты PD4, PD5, PD6, PD7 и PC0, PC1, PC2 в переменной.

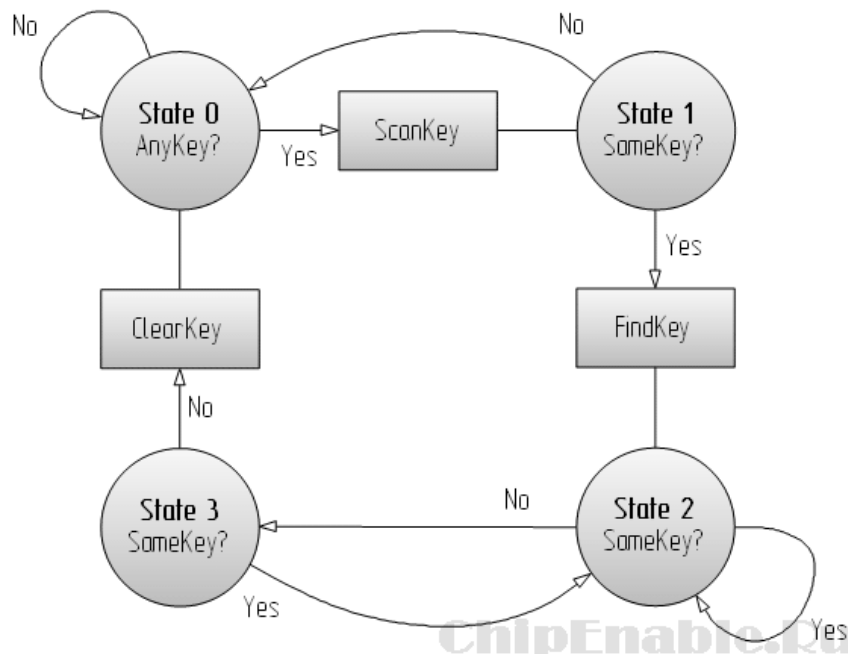
Повторяем описанную последовательность для двух оставшихся строк.

Переключения кнопок клавиатуры сопровождаются дребезгом контактов, который микроконтроллер может "воспринимать" как многократные нажатия. В приложениях, использующих клавиатуры, это нежелательное явление, поэтому в программе опроса клавиатуры обязательно должна быть какая-нибудь защита. Обычно делают так - факт нажатия кнопки регистрируется, если она удерживается в течение нескольких циклов опроса.

Код, получаемый в процессе сканирования клавиатуры, часто требуется преобразовать в символьное значение номера/буквы кнопки (например, для передачи по USART`у). Для этого можно создать таблицу перекодировки - двумерный массив. В первом столбце таблицы будут храниться коды кнопок, а во втором соответствующие символьные значения. Методом последовательного перебора в таблице можно находить нужное значение.

Алгоритм опроса матричной клавиатуры можно реализовать в виде конечного автомата (State Machine) – функции, которая в зависимости от своего состояния (значения определенной переменной) и входного воздействия, выполняет разную работу. На рисунке ниже представлена диаграмма подобного автомата.

Автомат (State Machine) для опроса клавиатуры



Начальное состояние автомата - 0. В этом состоянии автомат находится, пока не будет нажата какая-нибудь кнопка. Когда зафиксировано нажатие на кнопку, запускается функция сканирования клавиатуры ScanKey(), запоминается код нажатой кнопки и автомат переходит в состояние 1.

В состоянии 1 автомат проверяет, нажата ли в данный момент та же кнопка, что и в состоянии 0 или нет. Если коды кнопок не совпадают, автомат возвращается в состояние 0, если совпадают, запускается функция FindKey(), которая находит символическое значение номера кнопки и устанавливает флаги, сигнализирующие системе о нажатой кнопке. По завершению функции автомат переходит в состояние 2.

Пока удерживается одна и та же кнопка, автомат находится в состоянии 2. Если произошли какие-то изменения, он переходит в состояние 3.

Если изменения были случайными, автомат возвращается в состояние 2, а если нет, переходит в начальное состояние, чтобы снова запустить функцию сканирования клавиатуры.

Программная реализация автомата

Представленную диаграмму конечного автомата легко преобразовать в программный код.

//хранит текущее состояние автомата

unsigned char keyState;

//прототипы функций используемых автоматом

unsigned char AnyKey(void);

unsigned char SameKey(void);

void ScanKey(void);

unsigned char FindKey(void);

void ClearKey(void);

void ScanKeyboard(void)

```

{
    switch (keyState){
        case 0:
            if (AnyKey()) {
                ScanKey();
                keyState = 1;
            }
            break;

        case 1:
            if (SameKey()) {
                FindKey();
                keyState = 2;
            }
            else keyState = 0;
            break;

        case 2:
            if (SameKey()){}
            else keyState = 3;
            break;

        case 3:
            if (SameKey()) {
                keyState = 2;
            }
            else {

```

```

        ClearKey();
        keyState = 0;
    }
    break;

default:
    break;
}
}

```

Довольно наглядная запись. Теперь все, что нам остается, это написать/разобрать реализации недостающих функций. Чтобы код программы был максимально понятным, давайте обойдемся без макроопределений. Итак, по порядку.

unsigned char AnyKey(void)

```

{
    PORTD |= 0xf0;
    return (PINC & 0x07);
}

```

Устанавливаем на выводах PD7 – PD4 единицы, возвращаем состояния выводов PC2 – PC0. Если какая-нибудь из кнопок клавиатуры в этот момент нажата, функция вернет значение отличное от нуля, то есть true.

//хранит код нажатой кнопки

unsigned char keyCode;

void ScanKey(void)

```

{
    unsigned char activeRow = (1<<4);
    while (activeRow) {
        PORTD = (PORTD & 0x0f)|activeRow;
        if (PINC & 0x07) {
            keyCode = (PINC & 0x07);
            keyCode |= (PORTD & 0xf0);
        }
        activeRow <<= 1;
    }
}

```

Устанавливаем в 1 четвертый бит переменной activeRow (активизируем первую строку). Обнулив биты PD7 – PD4, записываем переменную в PORTD. Если какой-нибудь из трех младших разрядов регистра PINC установлен в единицу, значит в данный момент нажата кнопка, относящаяся к первой строке. Сохраняем биты PD7 – PD4 и PC2 - PC0 в переменную keyCode. Сдвигаем влево значение переменной activeRow на один разряд и повторяем цикл еще три раза.

unsigned char SameKey(void)

```

{
    PORTD = (PORTD & 0x0f) | (keyCode & 0xf0);
    return ((PINC & keyCode) & 0x07);
}

```

Функция проверяет, совпадает ли код нажатой в данный момент кнопки с кодом, полученным в предыдущем цикле опроса. Для этого на нужной строке устанавливается 1 – в PORTD записываются старшие 4 бита keyCode. Затем считывается регистр PINC, на него накладывается маска в виде переменной keyCode и выделяются 3 младших разряда. Полученное значение возвращается. Если коды кнопок совпадут, значение будет отлично от нуля, то есть true.

//хранит символьное значение нажатой кнопки

unsigned char keyValue;

//флаговая переменная - устанавливается, если кнопка удерживается

unsigned char keyDown;

//флаговая переменная - устанавливается, когда нажата новая кнопка

unsigned char keyNew;

//таблица перекодировки

```

__flash unsigned char keyTable[][2] = {
    { 0x11, '1'},
    { 0x12, '2'},
    { 0x14, '3'},
    { 0x21, '4'},
    { 0x22, '5'},
    { 0x24, '6'},
    { 0x41, '7'},
    { 0x42, '8'},
    { 0x44, '9'},
    { 0x81, "**"},
    { 0x82, '0'},
    { 0x84, '#'}
};

```

unsigned char FindKey(void)

```

{
    unsigned char index;

```

```
for (index = 0; index < 12; index++) {  
    if (keyTable [index][0] == keyCode) {  
        keyValue = keyTable [index][1];  
        keyDown = 1;  
        keyNew = 1;  
        return 1;  
    }  
}  
return 0;  
}
```

Эта функция ищет по таблице перекодировки номер кнопки, соответствующий коду, полученному на этапе сканирования клавиатуры. Если номер найден, он записывается в переменную keyValue и устанавливаются флаги keyDown и keyNew. Первый флаг сигнализирует о том, что кнопка удерживается нажатой, второй, что автомат зафиксировал нажатие новой кнопки.

```
void ClearKey(void)  
{  
    keyDown = 0;  
}
```

Эта функция сбрасывает флаговую переменную keyDown.

Для полного счастья нам не хватает функции инициализации и функции возвращающей содержимое переменной keyValue. Здесь мы их уже не будем разбирать, потому что ничего интересного там нет.
Полный текст программного модуля

keyboard.h (/images/stories/articles/c_course/keyboard/keyboard-h.html)
keyboard.c (/images/stories/articles/c_course/keyboard/keyboard-c.html)

Файлы

Тестовые проекты – микроконтроллер atmega8 опрашивает клавиатуру и посылает номер нажатой кнопки по USART`у в терминал.

IAR. Опрос матричной клавиатуры (/files/course/keyboard/Keyboard-iar.rar)

WINAVR. Опрос клавиатуры (/files/course/keyboard/Keyboard-winavr.rar)

CodeVision. Опрос клавиатуры (/files/course/keyboard/Keyboard-CV.rar)

Proteus – проект для отладки

(/files/course/keyboard/Keyboard-Proteus.rar) Драйвер для матричной клавиатуры (/files/course/keyboard/Keyboard-driver.rar)

Твитнуть



Tagged under #матричная клавиатура (/index.php/programming-avr/itemlist/tag/матричная%20клавиатура.html) #опрос клавиатуры (/index.php/programming-avr/itemlist/tag/опрос%20клавиатуры.html) #state machine (/index.php/programming-avr/itemlist/tag/state%20machine.html) #автомат (/index.php/programming-avr/itemlist/tag/автомат.html) #автоматное программирование (/index.php/programming-avr/itemlist/tag/автоматное%20программирование.html)

Related items

- Реализация конечного автомата (state machine) (/index.php/programming-avr/item/90-realizatsiya-konechnogo-avtomata-state-machine.html)
- Организация программ. Событийная система (Event driven system) (/index.php/programming-avr/item/47-organizatsiya-programm-sobytiynaya-sistema-event-driven-system.html)

Комментарии (/index.php/component/jcomments/feed/com_k2/53.html)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-klaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-246)
Guest 14.03.2010 16:01
Спасибо, за статью :-)
Насущный вопрос, а то моих девайсов вечно управление какое-то не человеческое :oops:

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-klaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-247)
foxit 15.03.2010 04:30
супер
давно ждал!

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-klaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-250)
Рыку_He_оТTyda 16.03.2010 12:21
Спасибо!
Очень интересно!
Не планируете сделать описание модной сейчас QTouch?

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-klaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-251)
Pashgan 16.03.2010 21:49
В ближайшее время вряд ли. Но вообще тема довольно интересная.

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-klaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-256)
Guest 19.03.2010 13:31
не лучше ли поставить диоды вместо R8...R11, для устранения замыкания портов при нажатии нескольких кнопок.

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-klaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-257)
Pashgan 19.03.2010 15:30

лучше, но немного дороже

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-klaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-2438) **Chupa** 24.01.2013 17:19

Цитирую Pashgan:

лучше, но немного дороже

В чем недостатки использования резисторов на 330 Ом вместо диодов?

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-klaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-2439) **JoJo** 24.01.2013 18:47

Резисторы ограничат ток в случае замыканий, а диоды замыкание просто не допустят.

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-klaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-259) **Guest** 20.03.2010 11:31

Предлагаю С сделать на выход, а D на вход и нормальное состояние не 0 а 1. Имеем: R5-7 не нужны, R8-11 не нужны (включаем pullup), ну и по желанию R2-4 заменить на диоды.

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-klaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-260) **Guest** 20.03.2010 11:49

Нет предела совершенству. Подана идея, а что куда подключить - это уже вариации.

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-klaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-300) **Guest** 12.04.2010 19:01

а как быть, если в одной линии нажмется > 1 кнопки? S1 и S2 например, scankey даст 1 + 2 = 3, в таблице такого значения нет - значит проигнорируется

а если хочется комбинации использовать...

но статья полезная, 10х

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-klaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-301) **Pashgan** 12.04.2010 20:08

Если хочется использовать комбинации, нужно усложнять код. А в каком приложении это может потребоваться?

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-klaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-303) **Guest** 13.04.2010 17:29

ну даже не знаю пока, чисто теоретически :)

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-klaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-312) **Guest** 15.04.2010 09:18

а как заставить протеус что-то показывать? туло распаковать, залить прошивку из этой же папки - ничего не происходит

может протеус у меня кривой какой-то.. не могу никак с кнопками научиться в нем что-то делать, думал - тут же готовый работающий проект :(

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-klaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-313) **Pashgan** 15.04.2010 10:32

Распаковать, взять прошивку из этой же папки. Если не идет, попробуй с этой страницы скачать проекты и взять из них прошивки. У меня на Proteus'e 7.53 SP3 все работает. А другие проекты запускал?

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-klaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-314) **Guest** 15.04.2010 13:45

та запускал - трабла именно если кнопки надо опрашивать, что проект к этому уроку, что моя схемка, с которой все началось (там тоже 3 кнопки)

шахматы из тестовых программ работают, мои тесты работают с выводом, с uart-ом..

грешу на кривость протеуса (мож с взломом както связанную).. отладка в avr studio показывает, что все нормально - но хотелось же упростить жизнь себе :)

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-klaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-315) **Pashgan** 15.04.2010 19:15

У меня в Proteus'e глючила матричная клавиатура - есть там такой компонент. Весь мозг себе сломал, пока не догадался собрать ее из отдельных кнопок. Тогда программа заработала как часы.

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-klaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-316) **Guest** 15.04.2010 19:16

начал по шагам пробовать - одну кнопку, диод на другой порт, led = key - зажигает

то же самое в свою схему - не зажигает

попробовал на другой порт - зажигает

может где-то в инициализации порт используется как-то по другому..

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-klaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-331) **DVF** 17.04.2010 20:34

Как мы привыкли таскать за собой программную проверку надребезг контактов... В примере к AVR вспомните, что на входе используется элемент Шмитта к которому можно подцепить С и, если его не достаточно, то поставив на вход внешний такой элемент с RC (что, конечно же,

сильно удорожит схему ;-)) можно отказаться от повторов.

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-332)

Pashgan 17.04.2010 20:45

По мне так главное чтоб устройство правильно работало, а какую защиту использовать - аппаратную или программную - это дело вкуса/предпочтений. Триггер Шмитта на входе AVR вообще не спасает. Он помогает от затянутых фронтов.

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-334)

DVF 18.04.2010 07:53

Вот искусственно затянутый аппаратно фронт и спасет от дребезга ;-)
Я только дополняю изложенное Вами.

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-337)

Guest 21.04.2010 15:58

Подскажите пожалуйста как доработать код, чтобы убрать резисторы 5, 6, 7?

У меня 4x4 клавиатура - все работает, но надо убрать эти резисторы.

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-338)

Pashgan 21.04.2010 16:46

Ммм...это займет некоторое время.

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-339)

Guest 22.04.2010 00:18

Заранее благодарю. Если придумаю раньше - опишусь здесь... кстати у меня все линии на одном порту.

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-344)

Pashgan 23.04.2010 18:10

SVK, выложил на форум в тему "Arnote AVR242 на Си"

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-345)

Guest 24.04.2010 04:04

Спасибо!

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-377)

foxit 07.05.2010 11:35

Pashgan

Расскажи про теорию конечных автоматов(StateMachine).

Что это такое?

Как использовать в целом и применительно
к контроллерам и написанию программ для МК?
Как строятся диаграммы?

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-1028)

foxit 20.01.2011 14:21

Pashgan, Если не трудно сделай модуль для работы с клавиатурой 4x4. Спасибо.

Возможен ли вариант без применения подтягивающих резисторов?

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-1036)

Pashgan 22.01.2011 23:16

Уже есть. Я его для программатора цифровых резисторов делал. Просто руки никак не дойдут до переделки этой статьи и написания небольшого продолжения.

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-1050)

гость 28.01.2011 09:54

почему я не могу разместить KeyTable в keyboard.h? IAR выдает ошибку.

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-1058)

Pashgan 29.01.2011 22:04

Потому что переменная оказывается объявленной сразу в двух местах. Добавь перед ней ключевое слово static. Это слово ограничивает область видимости переменной в пределах файла.

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-1071)

Гость 31.01.2011 16:28

Pashgan, расшифруйте это выражение:

Код:

```
( *(&PORTX_ROW-2) )
```

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrichnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-1296)

Pashgan 05.03.2011 06:47

Это макрос, вычисляющий указатель на регистр PINx.

В хедер файле задается название порта, например PORTB, а адреса остальных регистров PINB и DDRB вычисляются. Сделано для удобства.

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrchnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-1276)

Читатель 28.02.2011 07:50

Цитирую Гость:

не лучше ли поставить диоды вместо R8...R11, для устранения замыкания портов при нажатии нескольких кнопок.

Если ставить диоды, то анодом к МК, а катодом к клавиатуре?

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrchnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-1299)

foxit 05.03.2011 07:28

Pashgan, а можно модуль под клавиатуру 4x4?

Как добавить срабатывание буззера по нажатию клавиши?

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrchnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-1313) **Pashgan** 06.03.2011 21:42

Так я его на форуме выложил.

Цитата:

Как добавить срабатывание буззера по нажатию клавиши?

Реализую в материале по клавиатуре 4 на 4. Думаю в этом месяце сгенерию.

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrchnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-1471)

Dmi3 29.04.2011 11:29

Это безумие какое-то! Зачем всё так сложно делать? Обычное решение для устранения дребезга и повышения производительности это

РЕДКИЙ опрос кнопок. Если опрашивать реже чем время типичного дребезга то в чём проблема?

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrchnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-1477) **Pashgan** 01.05.2011 18:22

А что сложного то? Код простой как три копейки.

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrchnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-1854)

svenher 16.11.2011 12:34

у меня в курсовой такое есть задание.

может кто-нибудь такое в ассемблере изобразить??

задание: сделать домофон. вместо * и # сброс и вызов. должна быть светодиодная индикация нажатых цифр.

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrchnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-2020)

wukrlvy 17.02.2012 17:17

На этой страничке достаточно комментариев, что говорит о заинтересованности в данной теме. Нет предельного совершенствованию программы, да и задача стояла чисто учебная, но я хотел бы все таки добавить пару "ложек дегтя", чтобы программисты внимательней приглядывались к тому, что написано:

1)таймер TA0 работает в режиме NORMAL и считает в диапазоне от 0x83 до 0x00 (125 тактов). Если его переключить в режим CTC, то тогда в прерывании не надо будет перезаписывать счетчик TCNT0. 125 тактов по 256/16 = 16 мкс равно 2 мс, в комментарии в программе написано 4 мс. Я бы поставил минимум 20 мс.

2)Алгоритм цифрового автомата рассчитан на "убирание дребезга" по одной кнопке. Именно по той, которую Вы затем будете удерживать. Реально есть такие вещи, как "слежка промазал". В этом случае появляется дребезг не только на той кнопке, которую Вы жмете, но и на той, которую Вы зацепили. Действие ScanKey стоит в цикле устранения дребезга. И оно первый раз может сработать по кнопке "которую промазали". Если через 2 мс "лишний" сигнал не снялся, нажатие воспринимается как правильное.

3)в комментарии к строке

Код:

```
return ((PINC & KeyCode) & 0x07)
```

я бы написал: "Если нажата еще та кнопка, по которой сформирован код". Потому, что тут никакого сравнения не производится. Это место в программе "перл", и я ставлю ему много плюсов!

4)Ну уж очень много глобальных переменных хранят внутреннее состояние программы - keyState, KeyCode, keyDown, keyNew, KeyValue.

keyDown не используется, об ней не будем.

keyNew равна единице, когда keyValue не равна нулю. Т.е. keyValue прекрасно заменяет первую переменную, если в функции GetKey ее обнулять.

Я бы три оставшиеся переменные объявил как статические локальные для функции ScanKeyboard. Тогда они, как и прежде, будут доступны для "всех заинтересованных", но при этом будут защищены от внешних воздействий.

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrchnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-3988)

Vasilich 21.04.2014 12:51

А вот спли-мод тут хрен применишь

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrchnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-3989) **Pashgan** 21.04.2014 12:56

Мир не совершенен. Увы..

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrchnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-4575)

skullhead 28.08.2015 13:08

Добрый день! Подскажите пожалуйста, как адаптировать схему под использование нормально-замкнутых контактов (герконов)?

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)

(/index.php/programming-avr/item/53-uchebnyy-kurs-opros-matrchnoy-kaviatury-primer-ispolzovaniya-avtomata-state-machine.html#comment-4576) **skullhead** 28.08.2015 13:08

возможно не верно назвал тип контакта. в общем, в исходном состоянии все контакты замкнуты, и срабатыванием считает размыкание контактов.

[Ответить](#) | [Ответить с цитатой](#) | [Цитировать](#)