LATEST ARTICLES Comparing Low Cost AVR Dev Boards Ultrasonic Rangefinder HC-SR04 Interfacing with ATmega8 PIC16F877A Serial Communication Demo USART Library for PIC –

4x3 Matrix Keypad Interface - AVR Tutorial

Oct 13, 2010 Avinash AVR Tutorials 44

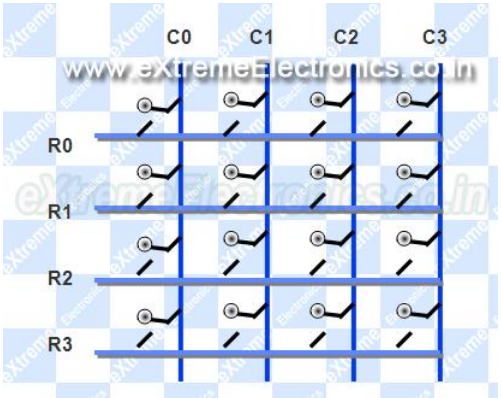
Many application requires large number of keys connected to a computing system. Example includes a PC keyboard, Cell Phone keypad and Calculators. If we connect a single key to MCU, we just connect it directly to i/o line. But we cannot connect, say 10 or 100 keys directly MCUs i/o. Because :-

It will eat up precious i/o line.

MCU to Keypad interface will contain lots of wires.

Buy Matrix Keypad

We want to avoid all these troubles so we use some clever technique. The technique is called multiplexed matrix keypad. In this technique keys are connected in a matrix (row/column) style as shown below.



Matrix Keypad Basic Connection

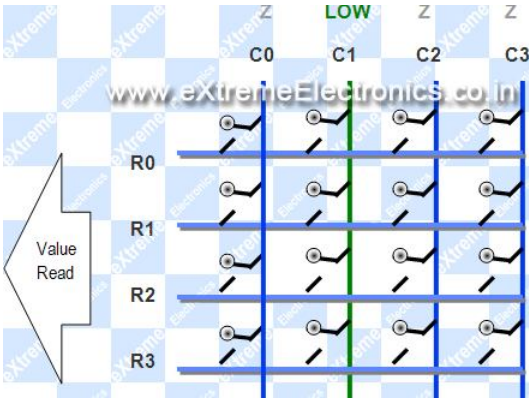
The rows R0 to R3 are connected to Input lines of Microcontroller. The i/o pins where they are connected are made Input. This is done by setting the proper DDR Register in AVR and TRIS Register in PIC. The column C0 to C3 are also connected to MCUs i/o line. These are kept at High Impedance State (AKA input), in high z state (z= impedance) state these pins are neither HIGH or LOW they are in TRISTATE. And in their PORT value we set them all as low, so as soon as we change their DDR bit to 1 they become output with value LOW.

One by One we make each Column LOW (from high Z state) and read state of R0 to R3.

Column 0 Selected

As you can see in the image above C0 is made LOW while all other Columns are in HIGH Z State. We can read the Value of R0 to R3 to get their pressed status. If they are high the button is NOT pressed. As we have enabled internal pullups on them, these pullups keep their value high when they are floating (that means NOT connected to anything). But when a key is pressed it is connected to LOW line from the column thus making it LOW.

After that we make the C0 High Z again and make C1 LOW. And read R0 to R3 again. This gives us status of the second column of keys. Similarly we scan all columns.



Column 1 Selected

How to Do it All with AVRs

Each i/o port in AVR has three related registers PORTx, DDRx and PINx. For example port A has

PORTA Port Driver - when any bit is set to 1 it appears as HIGH i.e. 5v . But this is the case only if that bit is OUTPUT. If it is input, setting any bit to 1 enables the internal pullup on that bit.

DDRA DATA DIRECTION REGISTER - Make any pin on than port as IN or OUT. When bit is 1 it represents Output. When bit is 0 it represents Input. Input state is also called tristate or high Z state.

PINA - Read it to get the level (HIGH or LOW) at the actual i/o pin. It is read when the pin is made input.

So now you know

How to make any i/o line Input(high Z) or Output.

How to enable internal pullup register on input lines.

How to read value that is present on input lines.

Please see the following tutorial for more clarification.

http://extremeelectronics.co.in/avr-tutorials/part-v-digital-io-in-avrs/

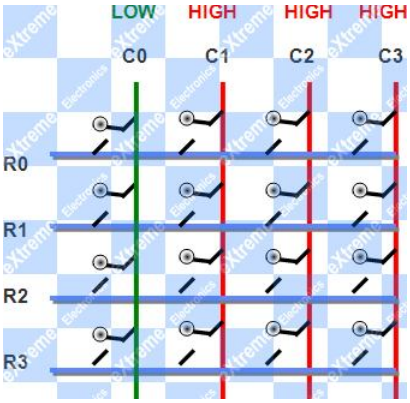
Recent Article
No Image
Comparing Low Cost AVR Dev Boards
Nov 22, 2013 5
Over the years I have seen people just open up the mouth...

Recent Comments
Hello Sir, Please am not good at writing programs...
1 day ago
i want to know more about spi communication in pi...
3 days ago
may i know how to use keypad by using assembly la...
1 week ago
Hey my previous problem of lcd initialisation was...
1 week ago
sir, i have few question. 1> in my pc wher...
1 week ago

Popular
GUI Softwar based USB Programmer
May 10, 2
Using LCD M AVRs
Jul 28, 20
Programmin for Embedde Developmen
Jul 17, 20
RF Communi Between Microcontro
Jan 17, 2

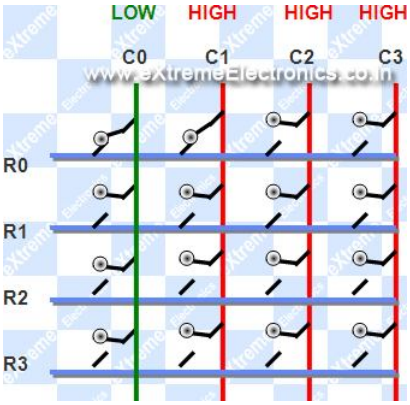
Why we make other Columns High Impedance while one column is made LOW?

Lets say we selected column number C0, so we make it LOW(i.e. GND or logic 0), in the same time we make all other columns high impedance (i.e. input). If we don't make other lines high impedance (tristate or Input) they are in output mode. And in output mode they must be either LOW(GND or logic 0) or HIGH (5v or logic 1). We can't make other lines LOW as we can select only one line at a time and C0 is already low as per assumption. So the only other possible state is all other columns are HIGH. This is shown in figure below. Red colour on column indicate high state while green is for low state.



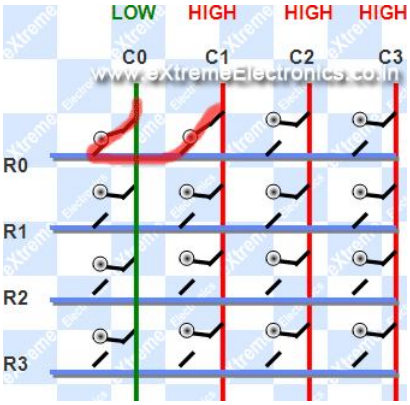
Wrong Way!

Suppose at that time the user presses KEY0 and KEY1 simultaneously as shown below.



Short Circuit !

As you can see clearly that it create a short between C0 (GND) and C1 (5v), this will burn out the buffer of the MCU immediately!



Short!

That's why all other columns are kept at tristate(neither LOW nor HIGH) but very high input impedance that prevent either source or sink of current from them. So if we kept C1 at high impedance state it wont allow current to flow to GND on C0.

avr-gcc C code for 4x3 matrix keypad

```
/*****************************************************************************

   Program to learn the use of Multiplexed 4x3 keypad with AVR Microcontroller.

   Specific Skills Required
      >> AVR GPIO details.(http://bit.ly/aq3ouw)
      >> LCD Library.(http://bit.ly/agVUVc)
      >> Operations on bits using C.(http://bit.ly/aFqg5n)


   General Skills Required
      >> AVR Studio Setup and use. (http://bit.ly/aZ43SZ)
      >> avr-gcc setup and use.


   Hardware
   --------
   ATmega32 @ 16MHz external crystal.
   Fuse Byte setting HIGH = C9 and LOW = FF (MOST IMP.)


   LCD   <->   AVR Connection

      VSS ->GND
      VDD ->+5V
      VEE -> CENTER PIN OF 10K POT (OTHER TWO PIN OF POT TO +5V AND GND)
            ADJ. THE POT UNTIL YOU HAVE A CLEAR TEXT DISPLAY.

      RS -> PD3
      RW -> PD6
      E  -> PB4

      DB0 -> N/C
      DB1 -> N/C
      DB2 -> N/C
      DB3 -> N/C

      DB4 -> PB0
      DB5 -> PB1
      DB6 -> PB2
      DB7 -> PB3

      LED+ ->+5V (VIA 100 OHM RES)
      LED- ->GND

   KEYPAD

      COL1 ->  PA6
      COL2 ->  PA5
      COL3 ->  PA4

      ROW1 ->  PA3
      ROW2 -> PA2
      ROW3 ->  PA1
      ROW4 -> PA0

                              NOTICE
                              --------
   NO PART OF THIS WORK CAN BE COPIED, DISTRIBUTED OR PUBLISHED WITHOUT A
   WRITTEN PERMISSION FROM EXTREME ELECTRONICS INDIA. THE LIBRARY, NOR ANY PART
   OF IT CAN BE USED IN COMMERCIAL APPLICATIONS. IT IS INTENDED TO BE USED FOR
   HOBBY, LEARNING AND EDUCATIONAL PURPOSE ONLY. IF YOU WANT TO USE THEM IN
   COMMERCIAL APPLICATION PLEASE WRITE TO THE AUTHOR.


   WRITTEN BY:
   AVINASH GUPTA
   me@avinashgupta.com



   ******************************************************************************/

   #include <avr/io.h>
   #include <util/delay.h>
```

```
#include "lcd.h"
#include "myutils.h"

#define KEYPAD A  //KEYPAD IS ATTACHED ON PORTA

//Don't Touch the lines below
//*******************************
#define KEYPAD_PORT PORT(KEYPAD)
#define KEYPAD_DDR   DDR(KEYPAD)
#define KEYPAD_PIN   PIN(KEYPAD)
//*******************************


/*****************************************

Function return the keycode of keypressed
on the Keypad. Keys are numbered as follows

[00] [01] [02]
[03] [04] [05]
[06] [07] [08]
[09] [10] [11]

Arguments:
   None

Return:
   Any number between 0-11 depending on
   keypressed.

   255 (hex 0xFF) if NO keypressed.

Precondition:
   None. Can be called without any setup.

*****************************************/
uint8_t GetKeyPressed()
{
   uint8_t r,c;

   KEYPAD_PORT|= 0X0F;

   for(c=0;c<3;c++)
   {
      KEYPAD_DDR&=~(0X7F);

      KEYPAD_DDR|=(0X40>>c);
      for(r=0;r<4;r++)
      {
         if(!(KEYPAD_PIN & (0X08>>r)))
         {
            return (r*3+c);
         }
      }
   }

   return 0XFF;//Indicate No key pressed
}


void main()
{
   //Wait for LCD To Start
   _delay_loop_2(0);

   //Now initialize the module
   LCDInit(LS_NONE);

   uint8_t key;

   while(1)
   {
      key=GetKeyPressed(); //Get the keycode of pressed key

      LCDWriteIntXY(0,0,key,3);  //Print it at location 0,0 on LCD.
   }

```

```
154 }
```

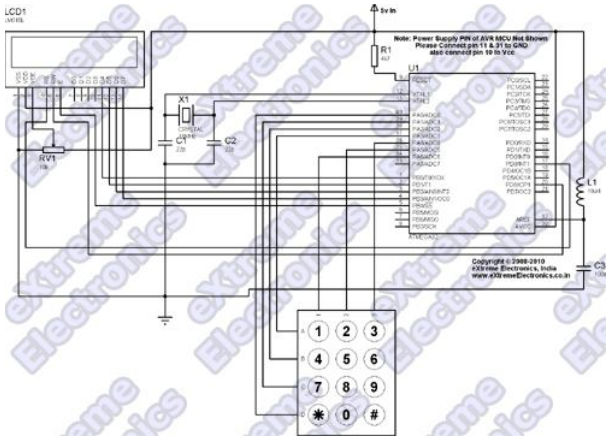

The above code make use of the LCD Library. You can get more information on LCD Library here :-

http://extremeelectronics.co.in/avr-tutorials/using-lcd-module-with-avrs/

Hardware for 4x3 Matrix Keypad and AVR interface.

The test circuit will be built around **ATmega32 microcontroller**. The output device will be a **16x2 lcd module**. So we set up a basic ATmega32 circuit. The circuit will have the following :-

1. ATmega32 MCU
2. 16MHz Crystal
3. Reset Circuit.
4. 5v Power Supply Circuit.
5. ISP (For programming)
6. LCD Module.
7. LCD Module Contrast adjust pot.



ATmega32 + LCD + Keypad Interface.

We have built the above circuit on a **Low Cost AVR Development Board**, but it does not has inbuilt LCD Module connector so you need to solder it yourself at the free area (and also do the wiring).

Compile the above program using AVR Studio (compiler is avr-gcc). And finally burn the program using any **ISP Programmer** to the ATmega32. **The fuse bits must be set as following to enable external crystal as clock source.**

High Fuse = C9 (hex value)

Low fuse =FF (hex value)

After burning the HEX file to MCU, finally you are ready to power up the setup. **When powered on, the LCD Screen Should show you the keycode of the key pressed on the keypad.** This complete our test.

Troubleshooting

NO Display on LCD

Make sure AVR Studio Project is set up for clock frequency of **16MHz** (16000000Hz)

Adjust the Contrast Adj Pot.

Press reset few times.

Power On/Off few times.

Connect the LCD only as shown on schematic above.

No response to key press.

Check that keypad is connected on PORTA only.

If you want to attach keypad on different port, change the line 80 on source code (keypad.c)

```
#define KEYPAD A //KEYPAD IS ATTACHED ON PORTA
```

Compiler Errors

1. Many people these days has jumped to embedded programming without a solid concept of computer science and programming. They don't know the basics of compiler and lack experience. To learn basic of compilers and their working PC/MAC/Linux(I mean a desktop or laptop) are great platform. But embedded system is **not good** for learning about compilers and programming basics. **It is for those who already have these skills and just want to apply it.**
2. Make sure all files belonging to the LCD Library are "added" to the "Project".
3. avr-gcc is installed. (The Windows Binary Distribution is called **WinAVR**)
4. The AVR Studio project Type is AVR GCC.

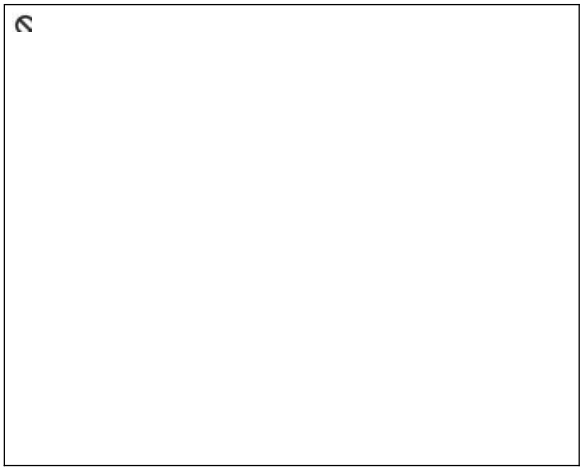
5. Basics of Installing and using AVR Studio with avr-gcc is described in this **tutorial**
6. How to add files to project is described in this **tutorial.**

General Tips for newbies

Use ready made **development boards** and **programmers**.

Try to follow the AVR Tutorial Series from the very beginning. (Remember the list spans four pages, page 1 is most recent addition thus most advance)

Video For 4x3 Keypad Interfacing.



User Videos

By Brendin

I really appreciate Brendin's approach on getting his problem solved and successfully porting the demo to ATmega48. What I recommend the users is to get your basics strong. You need full understanding of C language concept and the full details of the device you are programming, this will save you lots of time. So please go and read the good book on C and the datasheet of AVRs before you dive in! - ***Avinash***

Downloads

AVR Studio Project For 4x3 Keypad Interface.

VMLab simulation project : VMLAB is a very good free simulator for AVR's, you can use it to simulate the above circuit without making any hardware. Simply load "my_idea.prj" (available in the above package) in VMLab.

Proteus VSM Simulation Project.

HEX Code for 4x3 Keypad Interface.

Help Us!

We try to publish beginner friendly tutorials for latest subjects in embedded system as fast as we can. If you like these tutorials and they have helped you solve problems, please help us in return. **You can donate any amount as you like securely using a Credit or Debit Card or Paypal**.

We would be very thankful for your kind help.

By
Avinash Gupta
Facebook, Follow on Twitter.

4x3, 4x4, atmega32, input, keyboard, keypad, Keypads, lcd, low cost avr board

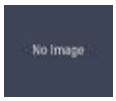
« Happy Diwali ! VS-2 Servo, Ultrasonic, RFID, SolarcelIs, Keypads, MicroSD - All New Arrivals! »

Avinash

Related articles

Comparing Low Cost AVR Dev Boards
Nov 22, 2013 5

Making A Thermometer with PIC16F877A
Jul 13, 2013 0



LCD Library for PIC - Setup on...
Jun 24, 2013 5

LCD Module Interface with PIC16F877A
Jun 12, 2013 10

More in this category

Ultrasonic Rangefinder HC-SR04...
Aug 12, 2013 21

Getting Started with AVR...
May 24, 2013 10

Introduction to AVR Microcontrollers
Oct 31, 2012 35

Interfacing Analog Joystick with AVR...
Oct 20, 2012 10

44 thoughts on "4×3 Matrix Keypad Interface - AVR Tutorial"

1. Mayukh Banerjee
October 16, 2010 at 12:29 pm

Hi Avinash,
great tutorial again, but you diddnt add the .c file as a part of the download. It would have helped to find out how your program actually works!!
Regards,
Mayukh.

Reply ↓

1. Avinash Post author
October 16, 2010 at 12:55 pm

All the files are provided at the end! I think you should be more careful before complaining. This confuses the other readers

Reply ↓

2. Mayukh Banerjee
October 16, 2010 at 6:00 pm

Hi Avinash,
the objective was to remove my confusion!! When i checked ur site, i didd'nt see the first link under downloads (which does not necessarily mean it was not there). I wanted to compare my code (for a 4×4 kpad) with yours (keeping yours as referance) as my approach for scanning the keys was somewhat different, making use of one loop and without the shifting.
Regards,
Mayukh.

Reply ↓

3. Brendin
October 20, 2010 at 4:35 am

Great tutorial. Could this be adapted for my atmega48 mcu? I was able to get the LCD tutorial to compile for the 48 but this one won't build.

Build started 19.10.2010 at 17:05:01
avr-gcc -mmcu=atmega48 -Wall -gdwarf-2 -std=gnu99 -DF_CPU=16000000UL -O2 -funsigned-char -funsigned-bitfields -fpack-struct -fshort-enums -MD -MP -MT Keypad.o -MF dep/Keypad.o.d -c ../Keypad.c
../Keypad.c: In function 'GetKeyPressed':
../Keypad.c:117: error: 'PORTA' undeclared (first use in this function)
../Keypad.c:117: error: (Each undeclared identifier is reported only once
../Keypad.c:117: error: for each function it appears in.)
../Keypad.c:121: error: 'DDRA' undeclared (first use in this function)
../Keypad.c:126: error: 'PINA' undeclared (first use in this function)
../Keypad.c: At top level:
../Keypad.c:137: warning: return type of 'main' is not 'int'
make: *** [Keypad.o] Error 1
Build failed with 5 errors and 1 warnings…

Reply ↓

4. Brendin
October 20, 2010 at 4:43 am

Sorry…but I got it to compile…I just needed to change to Keypad.c to use PORT B.

#define KEYPAD B //KEYPAD IS ATTACHED ON PORTA

I will try and breadboard it and see if it works.

Reply ↓

5. Brendin
October 20, 2010 at 6:47 am

I got it working on the atmega48. I needed to also set the reset pin disable to set PC6 as I/O (this unfortunately means I can no longer program with the ISP programmer). The code works for every button except 2 and 3 (using a 4×3 keypad 2 and 3 are equivalent to 2nd and 3rd button on the top row in this tutorial). When I hold down the 2 button the LCD flashes between 0 and 1 very fast. When I hold down 3 button the same thing happens but flashes between 0 and 2. All the other buttons work properly

I am not using an external clock. Fuse settings are Low=E2 High=5F and Extended=FF

What would cause this?

thanks,
Brendin

Reply ↓

6. Brendin
October 20, 2010 at 8:27 am

I changed some things around and got it working. I put the lcd data on port C and the keypad on port B. Now I don't have to worry about setting port c6 as an input and disabling reset. All of the keys output as expected. Here is a youtube link.

http://www.youtube.com/watch?v=7buBfN0kn04

Changes I made

lcd.h
/***
LCD CONNECTIONS
***/

#define LCD_DATA C //Port PB0 TO PB3 are connected to D4-D7

#define LCD_E C //Enable/strobe signal
#define LCD_E_POS PC4 // PB4 Position of enable in above port

Keypad.c
#define KEYPAD B //KEYPAD IS ATTACHED ON PORTA

Reply ↓

Avinash Post author

October 20, 2010 at 9:02 am

@Brendin
Congratulation on your success in porting the code to ATmega48!

Reply ↓

8. 

phani chakravarthi

October 22, 2010 at 11:15 pm

hi avinash..!!nice tutorial again..very helpful to me..
thank you..!

Reply ↓

9. 

maxmiaggi

June 18, 2011 at 12:22 am

Hi Avinash

I have interfaced a 4×4 keypad using atmega32 and I face similar problems as Brendin. I mean, whenever I press the 2nd, 3rd and 4th keys, the display flashes between 0 and 1, 0 and 2, 0 and 3 respectively. Unlike his case, the problem did not solve by changing the ports…

Reply ↓

10. 

Avinash Post author

June 18, 2011 at 8:56 am

@maxmiaggi

Bring out your oscilloscope, logic analyzer and in circuit emulator and fire your debugger !!!!!

Coz I have got no ESP like spiderman.

Reply ↓

11. 

Maruf

August 13, 2011 at 12:40 pm

Hi Avinash

Great Work….keep going. please give complete tutorial how to interface LCD with ATmega32 microcontroller with code…thank u for your time

Reply ↓

12. 

Sinet Rags

September 13, 2011 at 9:51 pm

@ Maruf ,

U have to climb b4 u reach the hilltop, so b4 commentin plz check the complete archive of this website…. Its all given….

LCD module has been explained quite well.

@ Avinash ….

Gr8 work dude . Can u post an article on how to hack/mod an RF toy car and put it to use instead of buying an new RF module .

Lol , only for those who already have a toy car at hom. !

Reply ↓

1. 

Avinash Post author

September 14, 2011 at 8:13 am

@Sinet

Its better to use ready made RF Module because.

1)Better quality.
2)Optimized for data transfer.
3)Cheap

Reply ↓

2.  Arkyro

August 16, 2013 at 8:02 pm

Hi, Google "hack toy car to make wireless robot" the first link that come should be what you are looking for

Reply ↓

13. Sinet Rags

September 15, 2011 at 10:14 am

@Avinash

I understand ur point of view …. but if a person has already got a toy car , all opened up then why hesitate in experimenting. If u have time can u atleast tell how can i connect an AtmegaXX to the RF toy car circuit … Correct me if i am wrong anywhere ….

Another question is how can i increase aerial data transfer range if i bought a RF module, for, currently i noe its limited to a modest span of some feet. Cuz elec.mag waves can propagate long distances.

Is it achieved with the aid of repeaters (Power Boosting). And if i dont have repeaters(non commercial experimentation) then how can i increase range of RF module by increasing its Power i/p .

Reply ↓

14.  Avinash Post author

September 15, 2011 at 10:20 am

@Sinet Rags,

Their are various RF Toy car in the market. Each one has a module made specially for that car.

Even the cheapest RF module works great at-least giving 50 feet range in full urban environment.

If your is giving only few feets then their may be series design fault in your designs.

Reply ↓

15. Sinet Rags

September 15, 2011 at 10:24 am

@ Avinash …

any coding tricks in order to reduce the size of this code ???

In the "p" section a user can delete a written character .
Feedback required / THANKS !

Reply ↓

16. Sinet Rags

September 15, 2011 at 10:30 am

@ Avinash . dude remove my code . it isnt full . if u dont mind i can post the complete code ! its ready . tested . woking

Reply ↓

17. Rahul Ch

Hi Avinash,

As Brendin said that to use PC6 as input we have to disable RESET. Then,
1. How can we program AVR in such cases?
2. Whenever, I use PC6 as output, on pressing the
RESET switch, how does RESET gets activated,or,
what happens internally??? Does it generate some
kind of interrpt???
3. Somedays ago in a wireless robo-boat championship I
saw that none of the boats were working.
All the boats had RF modules attached all of the
same frequency, but DIFFERENT ADDRESS BITS set in
the Rx and Tx. Can you explain? Is it because all
of them were trying to work at the same frequency?
If yes, Why are address bits provided in Rx and Tx
modules using a DIP switch……..

Thank You in advance

Reply ↓

18. Sagar
February 24, 2012 at 7:37 pm

@Avinash

Plz guide me for how to interface 4*4 touch keypad to atmega 32.

Reply ↓

1. Sagar
February 25, 2012 at 4:30 pm

and if u have any readymade module for the same purpose then let me know.

Reply ↓

19. ydhakal
March 7, 2012 at 12:17 pm

where is the pdf version for this ?

Reply ↓

20. Girish
March 26, 2012 at 10:30 pm

@avinash can u tell me how to use this code for 4X4 keypad ?
i am a beginner kindly help me !!

Reply ↓

21. Tamilvanan.A
April 6, 2012 at 10:11 am

The program works no doubt in that.

But, you might encounter some key malfunctioning problems if you don't add a 1ms or 2ms delay before reading the inputs especially when you are running at 16MHz.

It's always advisable to add a 1ms or 2ms delay after you change the state of the port pins before reading.

If anybody encounters a problem with interfacing with the above program, kindly add '_delay_ms(1);' in line 122 and between lines 123 and 124. If problem persists increase the delay to 2ms.

Reply ↓

1. Max

October 25, 2013 at 4:32 am

Thank you so much for this comment. I amended his code for my own 4×4 keypad and spent hours wondering why my first row wasn't quite working until I read your comment and added a few asm nop's in there, thank you so much. Works a charm.

Reply ↓

1.  Avinash Post author

November 12, 2013 at 8:02 pm

@Max

Thanks for feedback !

22.  vick

April 21, 2012 at 11:44 am

Can I get the pdf version of this article ?

Reply ↓

23. X Man Emran

April 25, 2012 at 8:39 am

Please give me matrix keypad interface code for PIC 16f84 microconttrolar in C language. plzzzzzzzzzzzz help me.

Reply ↓

24. Sunny Ghuman

April 25, 2012 at 8:28 pm

@Emran this is advance of C code.You can change for different micro controller with vary few changes.No one can write code for every micro controller and compiler.If you know about the 16f84 then change it according to your requirement.other wise jaisa bana hai bana do jayada tension kyun lena.

Reply ↓

25.  vick

April 26, 2012 at 7:12 am

Can I have the pdf fille of this ?

Thank you

Reply ↓

26. nikhilr57

November 5, 2012 at 5:37 pm

What changes should i make if i want to read multiple keys at same time. more than one keys will be pressed at a same time, and i want to read them. Thanks

Reply ↓

27. nyoman

May 3, 2013 at 5:26 pm

use diode in reverse to the output pin and pull up all port…that's it .. simple

Reply ↓

28. Sägär

May 31, 2013 at 11:05 am

Mr. Avinash,

Thank you very much for such an elaborate tutorial.
I was trying to interface a 4 x 4 key pad with MSP430x, after few hick-ups in the beginning I was able to debug the problems and resolve them.
Once again thank you very much for your article.

regards,
s-ä-g-ä-r

Reply ↓

1.  Avinash Post author
May 31, 2013 at 11:58 am

@Sagar,

Welcome.

Reply ↓

29. Mohammad
June 26, 2013 at 8:54 pm

thank for your code.
but in line 28 have a problem and must be 3*r-c to get right number

Reply ↓

30. Krishna
August 30, 2013 at 12:25 pm

Dear Avinash How to convert this program like mobile key pad say 1 key have 3 charters to display.kindly help me out please

Reply ↓

31. jay dubey
October 24, 2013 at 5:06 pm

hi Avinash sir, i tried to interface 4*4 keypad with atmega16 but it displays numbers more then one time. e.g if i press 1 it displays 111111.

Reply ↓

32. muhammad
December 25, 2013 at 2:18 pm

may i know how to use keypad by using assembly language……(i mean code in assembly laguage for keypad)

Reply ↓

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

You may use these HTML tags and attributes: `<a href="" title=""> <abbr title=""> <acronym title=""> <b> <blockquote cite=""> <cite> <code> <del datetime=""> <em> <i> <q cite=""> <strike> <strong>`

Post Comment

STORE FORUM PRIVACY POLICY

+FOL

Notify me of followup comments via e-mail ☐

Recent Comments

Hello Sir, Please am not good at writing programs...
1 day ago

i want to know more about spi communication in pi...
3 days ago

may i know how to use keypad by using assembly la...
1 week ago

Tags

ADC atmega atmega8 atmega16 atmega32 AVR avr studio clock development board display glcd graphic lcd gsm hi-tech c i2c ir lcd library lm35 mcu microchip microcontroller mplab pic pic16f pic16f877 pic16f877a pic18 pic18f pic18f2550 pic18f4520 pic18f4550 ProGFX project pwm Robotics rs232 sensor Serial Seven Segment sim300 Timer UART USART winavr

Categories

- **32bit ARM Projects** (1)
- **AVR Development Board** (2)
- **AVR Projects** (15)
- **AVR Tutorials** (62)
- **Chitchat** (8)
- **Code Libraries** (13)
- **Code Snippets** (10)
- **Electronics** (5)
- **GSM Projects** (2)
- **Hardwares** (8)
- **Microchip PIC Tutorials** (17)
- **News** (17)
- **PIC Development Board** (6)
- **PIC16F877A Tutorials** (6)
- **Programming in 'C'** (1)
- **RF** (3)
- **Robotics** (7)
- **Software** (7)
- **Tools** (10)
- **Uncategorized** (1)