A 60 keys keyboard project using only 8 bits

The project is flexible and keys can be reduced to any number of columns

Instructions

Schematics

Source code

By a.ratti

This project is a 60 keys matrix with conventional rows while for the columns a decimal decoding has been implemented.

The binary to decimal convertion,in this project, is achieved using 32 small diodes (1N4148).
Using the decimal decoding technique it is possible to have 4 Row x 15 Columns (total 60 keys) using only 8 IO pins.

Every key has his own ascii character, the code is stored into the eeprom from address 1 to address 60.(decimal)

The relation between eeprom address and key address is as follows:

FIRST ROW		SECOND ROW	
eeprom address	Key address	eeprom address	Key address
1	1,1	16	2,1
2	1,2	17	2,2
3	1,3	18	2,3
4	1,4	19	2,4
5	1,5	20	2,5
6	1,6	21	2,6
7	1,7	22	2,7
8	1,8	23	2,8
9	1,9	24	2,9
10	1,10	25	2,10
11	1,11	26	2,11
12	1,12	27	2,12
13	1,13	28	2,13
14	1,14	29	2,14
15	1,15	30	2,15

THIRD ROW		FOURTH ROW	
eeprom address	Key address	eeprom address	Key address
31	3,1	46	4,1
32	3,2	47	4,2
33	3,3	48	4,3
34	3,4	49	4,4
35	3,5	50	4,5
36	3,6	51	4,6
37	3,7	52	4,7
38	3,8	53	4,8
39	3,9	54	4,9
40	3,10	55	4,10
41	3,11	56	4,11
42	3,12	57	4,12
43	3,13	58	4,13
44	3,14	59	4,14
45	3,15	60	4,15

See document #1 and document #3 to better understand key position and address.

When a key is pressed the action is decoded by the row and column buses.
Rows go high one at the time commencing with row 1 through row 4, columns are decoded by the decimal content of the lowest 4 bits of portB.

When one key is pressed the corresponding ascii character is sent serially (T9600) via PortA.2 A beeper will generate a short tone to acknowledge the transmission event.

A RTS flag goes low (PortA.3)as soon as the key pressed is decoded, hence there is a delay of 20 ms before transmission begins, flag returns high at the end of transmission.
This flag can be used to generate an interrupt to synchronize the master receiver.

Keeping the button pressed the character will repeat at a rate of 250 ms/char.

At startup characters are in UPPERCASE, to switch to LOWER CASE you need to press the key associated to ascii code 26. When in LOWER CASE re-pressing the same key will make system return to UPPER CASE.

The Key pad can be built with a different number of columns, let assume you want to build a key pad with 5 columns (4 x 5 = 20 keys). Take the schematic (document#1) and hide with a sheet 10 swiches comencing for the right side. The circuit visible is what you have to build for a 4 x 5 key pad. You will notice that the number of diodes drop to 7. With this 20 keys key pad you will be able to use the software without modifications exept for the ascii code that you want associate to your pad.

If you do not modify the ascii code associated you will receive the following:

```
First Row:        1 2 3 4 5
Second Row:       A B C D E
Third Row:        K L M N O
Fourth Row:       U V W X Z
```

To change the second row for the missing numbers, you have to look at the previous table and note that first column of the second row (2,1) has the corresponding ascii code stored at address 16 (decimal).
So go to the source code and modify the ascii code at address 16 through 20 with 54 (6); 55 (7); 58 (8); 59 (9) and 48 (0).
Ricompile and load program into your chip. Try again you will receive:

```
First Row:        1 2 3 4 5
Second Row:       6 7 8 9 0
Third Row:        K L M N O
Fourth Row:       U V W X Z
```

Redo the process for row 3 and 4 and your keypad will work with the coding you have choosen.

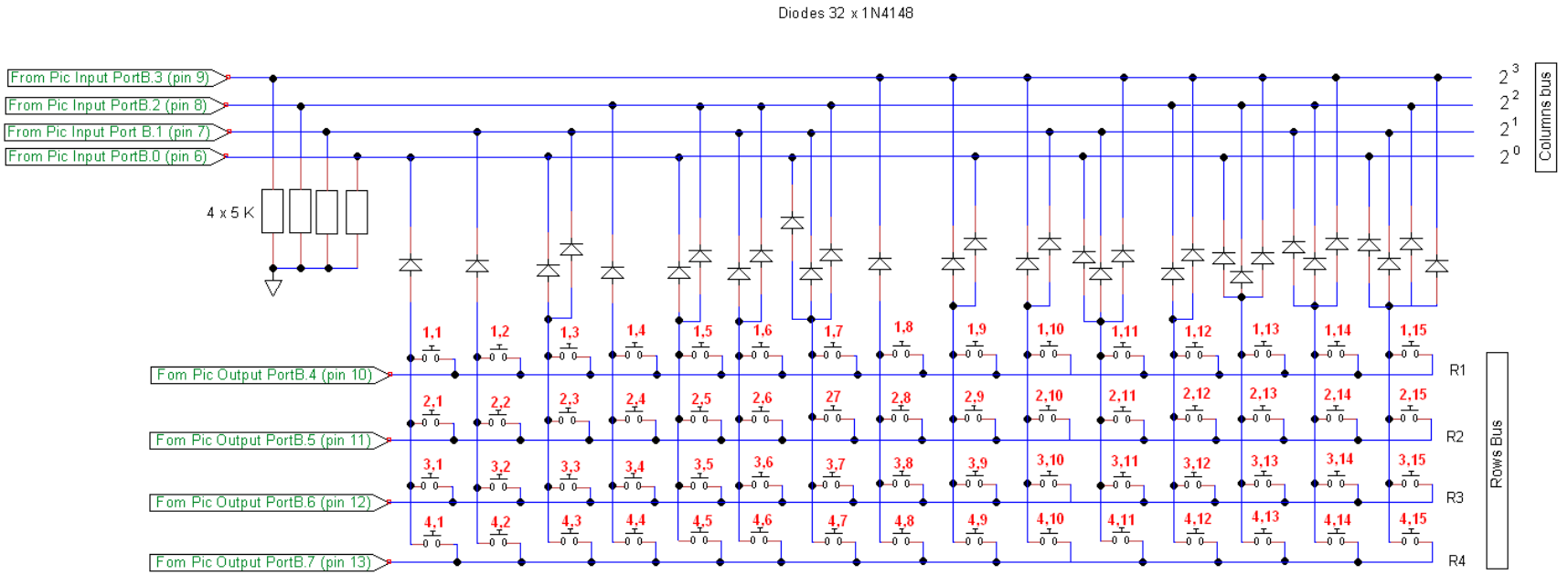
By default the system Tx the ascii character, if you need to TX the code then replace in the source code the (character SEROUT command) with the disabled (code SEROUT command).

The beeper is not mandatory! It is useful having a beep as keypress feedback but the circuit and program will work fine also without it.

To conclude the program is quite universal since it can work with one single column up to 15 columns without modifications, and ascii code can be personalized. Remember when you choose your code! ASCII 26 is a system command switch from UPPER CASE to LOWER CASE and viceversa.
The hardware it is easily tailored to suit also the needs for smaller keypad of any size from 4 x 1 to 4 x 15, so I think that embedded keypad should not be a problem anymore. (at least for hobbysts)

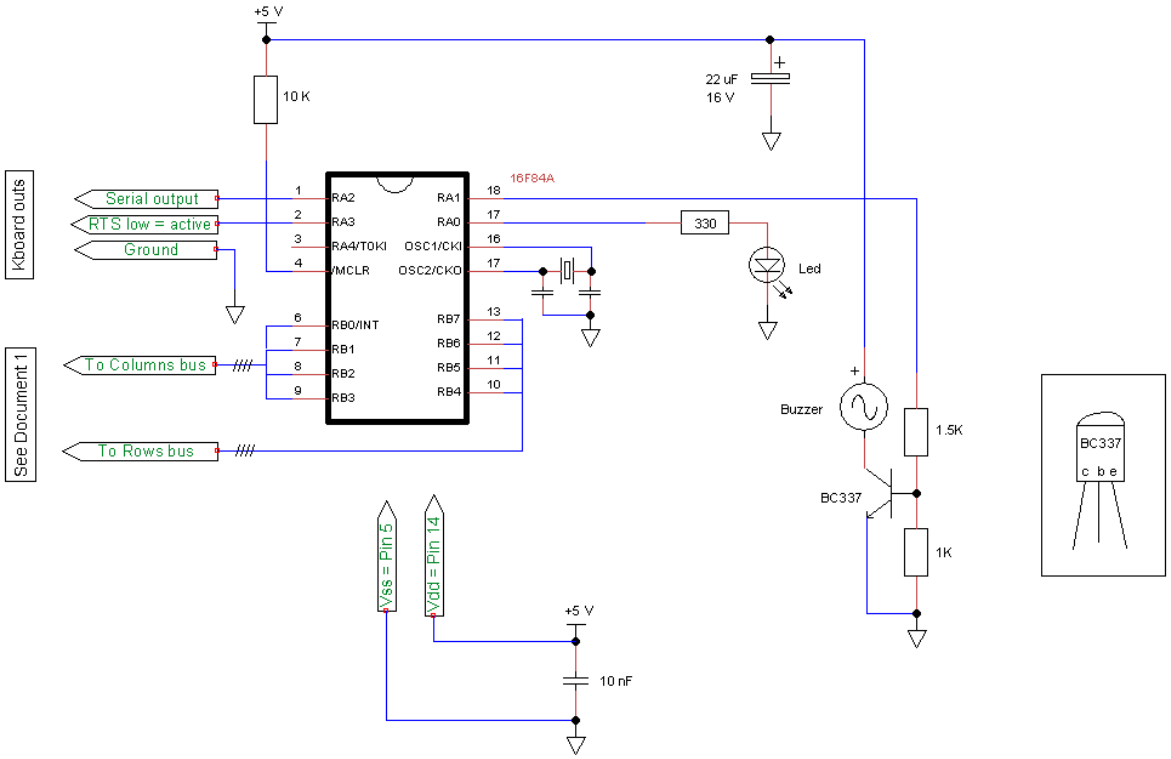
The microprocessor used is a PIC16F84A with crystal @ 4MHz.

A.ratti 16-11-2008
All progess begun with an idea

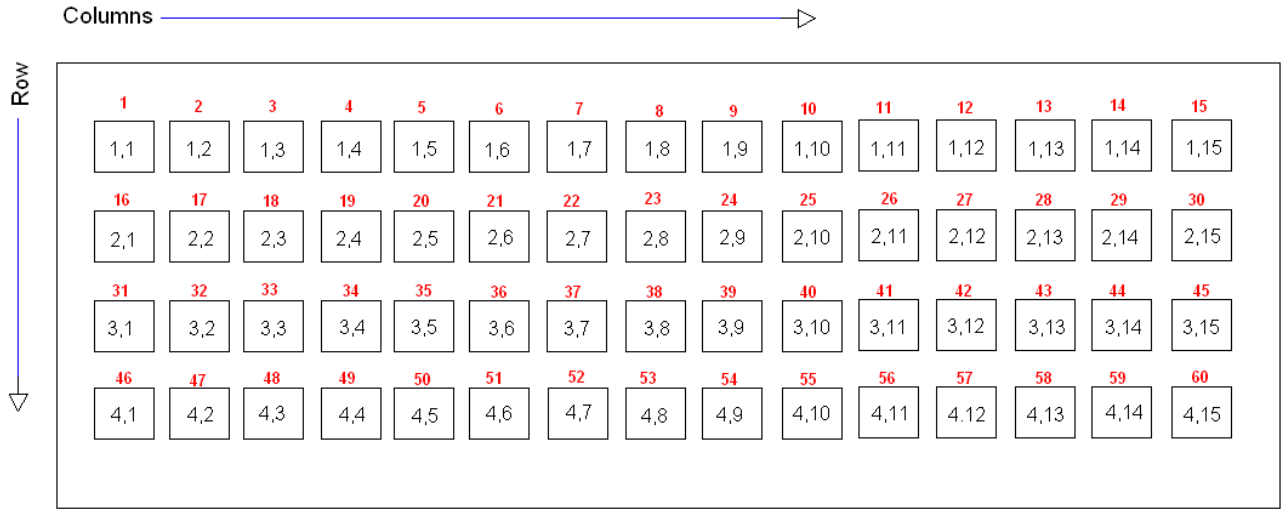


(Note: The matrix can be reduced to any number of columns)

Title 60 Keys Matrix Key pad using Pic16F84a		
Author a.ratti		
File cumenti\TinyCAD Examples\60_Keys_matrix.dsn		Document 1
Revision 1.0	Date 14-11-2008	Sheets 1 of 1



Title 40 keys matrix using Pic16F84A		
Author a.ratti		
File :\Alberto\Desktop\40_Keys_matrix_pic_Conn.dsn		Document 2
Revision 1.0	Date 14-11-2008	Sheets 1 of 1

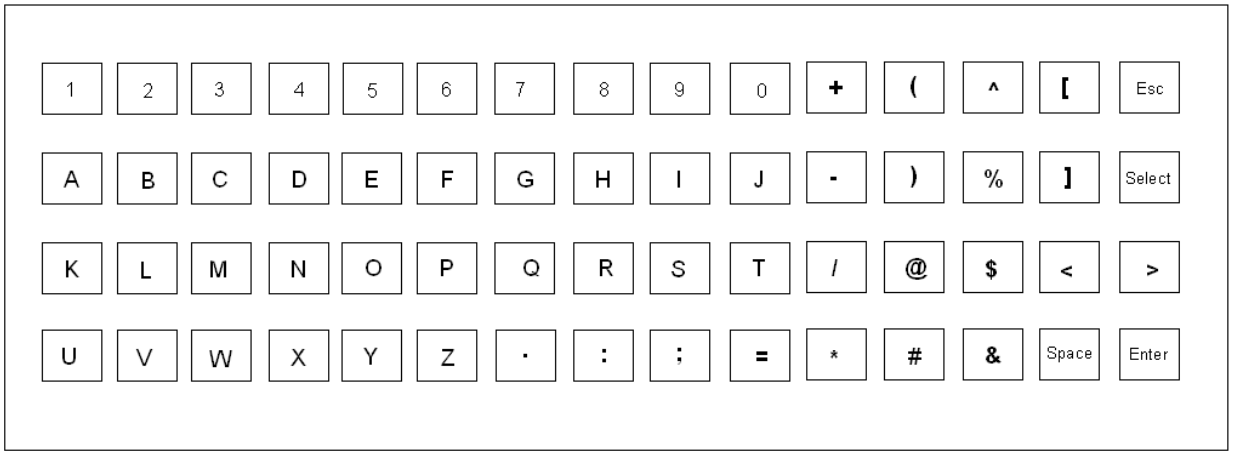


Key addresses layout

In Red : EEPROM decimal address

In Black : Key address (Row,Column)

Title Key addresses layout		
Author a.ratti		
File :ttings\Alberto\Desktop\Key_address_Default.dsn		Document 3
Revision 1.0	Date 16-11-2008	Sheets 1 of 1



Default characters layout

,

Title Default characters layout		
Author a.ratti		
File ettings\Alberto\Desktop\Default_Char_Layout.dsn		Document 4
Revision 1.0	Date 16-11-2008	Sheets 1 of 1

```
'****************************************************************
'*  Name    : 60 Keys_Kpad.BAS                                  *
'*  Author  : [a.ratti]                                         *
'*  Notice  : Copyright (c) 2008                                *
'*          : All Rights Reserved                               *
'*  Date    : 16/11/2008                                        *
'*  Version : 2.0                                               *
'*  Notes   :           Micro used: Pic 16F84a                  *
'*          : The software will decode any matrix keypad 4 x n  *
'*          :    where n is any columns number from 1 to 15     *
'****************************************************************

TrisA = %00000000
TRISB = %00001111

Define OSC 4

'--- EEprom ASCII code assignment to define keys response ---------

'------------------- First Row --------------------
Eeprom 1,[49,50,51,52,53,54,55,56,57,48,43,40,94,91,27]' (1,1)-(1,15)

'--------------------Second Row --------------------
Eeprom 16,[65,66,67,68,69,70,71,72,73,74,45,41,37,93,26]' (2,1)-(2,15)

'---------------------- Third Row ------------------
Eeprom 31,[75,76,77,78,79,80,81,82,83,84,47,64,36,60,62]' (3,1)-(3,15)

'---------------------- Fourth Row ------------------
Eeprom 46,[85,86,87,88,89,90,46,58,59,61,42,35,38,32,13]' (4,1)-(4,15)
'----------------------------------------------------------------



'--------------------------- Set variables and pins --------------------

LED         VAR PortA.0     ' Heart beating led
Buz         var PortA.1     ' got character acknowledgement
Tx          var PortA.2     ' Serial out
RTS         var portA.3     ' Ready to sent when low
T9600       con 2           ' baud rate = 9600

UCase       var byte
KeyFlag     var byte
KbOut       var Byte    [2] ' Buffer Array

high Tx                     ' purge serial line
High RTS                    ' disable RTS
low Buz                     ' Bepper off

KeyFlag = 0                 ' reset variable
Ucase = 0                   ' set uppercase


'--------------------------- Main program ------------------------------

 Loop:
 if KeyFlag>0 then           'If <> 0 then Tx ASCII character and Beep
 Low RTS                     'RTS flag enabled
 pause 20
 if KbOut[1]<32 then Loop    'Skip if ascii code lower than space

 If KbOut[1]=>65 then        'Skip ascii code <> character
 KbOut[0] = KbOut[1] + UCase 'If UCase = 0 then UpperCase
```

```
else
KbOut[0] = KbOut[1]
endif

'--------------------------------------------------------------------
Serout Tx,T9600,[KbOut[0]]  ' Tx ascii character
'----- Choose one of the two serout comand DON'T SELECT BOTH !!! -----
'Serout Tx,T9600,[#KbOut[0]]  ' Tx ascii code
'--------------------------------------------------------------------

High RTS                    ' RTS flag disabled
high buz                    ' Beep on
pause 30                    ' time duration of the beep
low Buz                     ' Bepper off
pause 150
endif
pause 50                    ' Heart beating frequency

toggle Led                  ' Heart beating


'----------------------------- KeyScan routine ------------------------

KeyFlag=0                   ' reset flag variable

'------------------------ First Row - keys from 1 to 15  --------------
PortB =16
If PortB>16 then            ' one key of the first row has been pressed
KeyFlag = (PortB-16)        ' decode to which column the key belongs
Gosub GetChar
goto Loop
endif

'------------------------ Second Row - keys from 16 to 30 -------------
PortB =32
If PortB>32 then            ' one key of the second row has been pressed
KeyFlag = (PortB-32) + 15   ' decode to which column the key belongs
gosub GetChar
goto Loop
endif

'------------------------ Third Row - keys from 31 to 45 --------------
PortB =64
If PortB>64 then            ' one key of the third row has been pressed
KeyFlag = (PortB-64) + 30   ' decode to which column the key belongs
gosub GetChar
goto Loop
endif

'------------------------ Fourth Row - keys from 46 to 60 -------------
PortB =128
If PortB>128 then           ' one key of the fourth row has been pressed
KeyFlag = (PortB-128) + 45 ' decode to which column the key belongs
gosub GetChar
endif

goto Loop

'------------------------ End of keyscan routine ----------------------


'--------------------------------------------------------------------
'---------------- Select Uppercase or LowCase letters ----------------
'------ (Ucase = 32 then lowerCase)  (UCase = 0 then UpperCase) -------
```

```
'-------------------------------------------------------------------------
GetChar: ' Read ASCII code from eeprom & set UCase or LCase
Read KeyFlag, KbOut[1]
If  KbOut[1]=26 then
If UCase=0 then
Ucase=32
else
UCase=0
endif
endif
return


end
```