A 40 keys keyboard project using only 8 bits

The project is flexible and keys can be reduced or increased up to 60

Instructions

Schematics

Source code

By a.ratti

The keyboard project is a 40 keys matrix with conventional rows while for the columns a decimal decoding is implemented.

The binary to decimal convertion,in this project, is achieved using 17 small diodes (1N4148)
Using the decimal decoding technique it is possible to have up to 15 columns x 4 rows (total 60 keys) using the same concept and with minor modifications.

Every key has his own ascii character, and this is loaded into an array at startup of the program. (see document #3 for keyboard default layout)

This coding can be customized changing the relative ascii reference in the array. (see source code)

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 receiver.

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

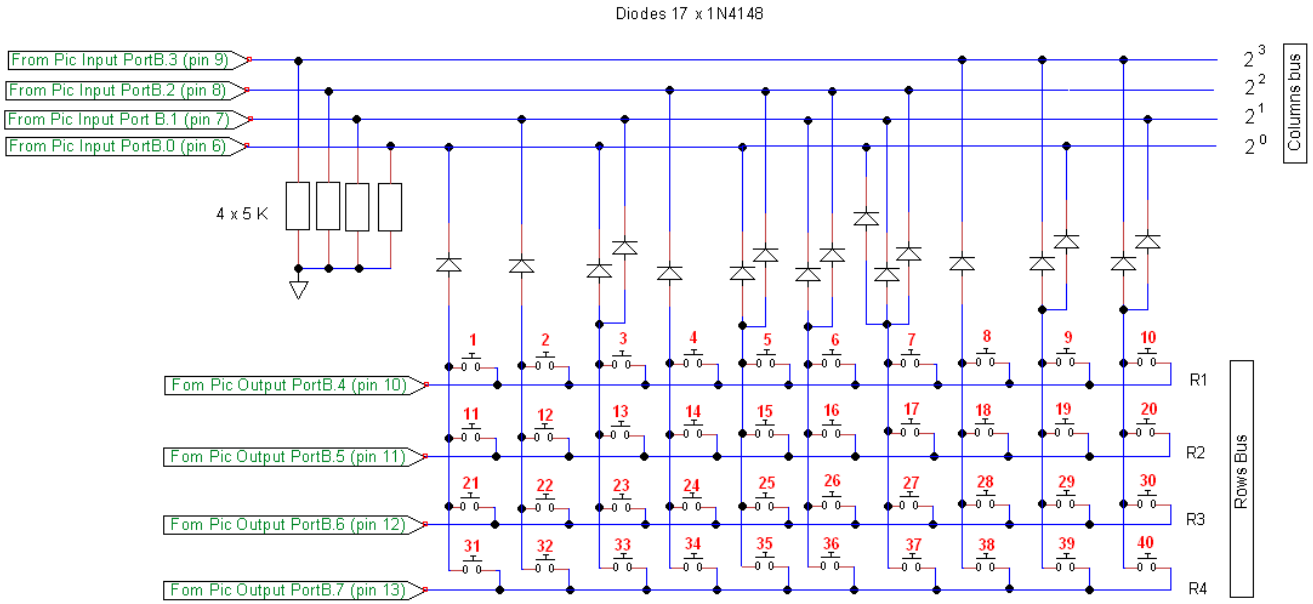
At startup characters are in UPPERCASE, to switch to lower case press key SELECT (ascii code 26). If Select key will be re-pressed system will return to UPPER CASE

The source code is well documented and you can modify it if needed. For instance by default the system Tx the ascii character if you need the code then replace the SEROUT comand.

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

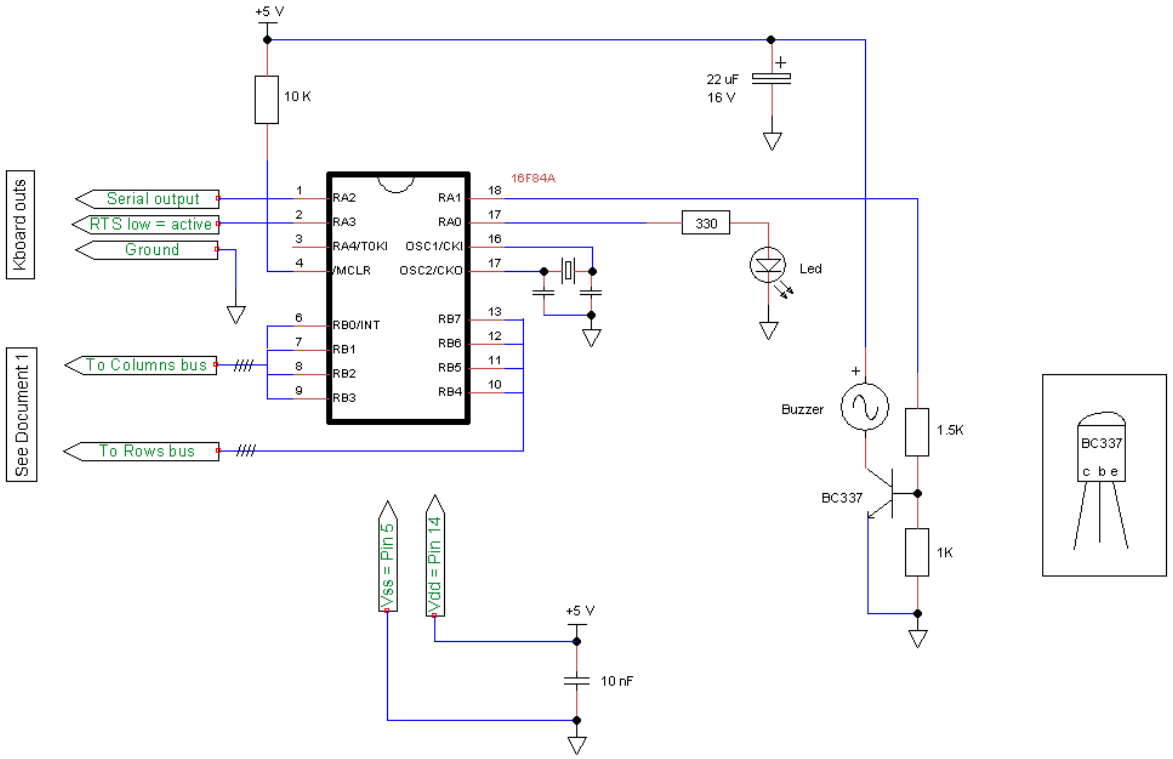
The microprocessor used is a PIC16F84A with crystal @ 4MHz.

A.ratti 14-11-2008

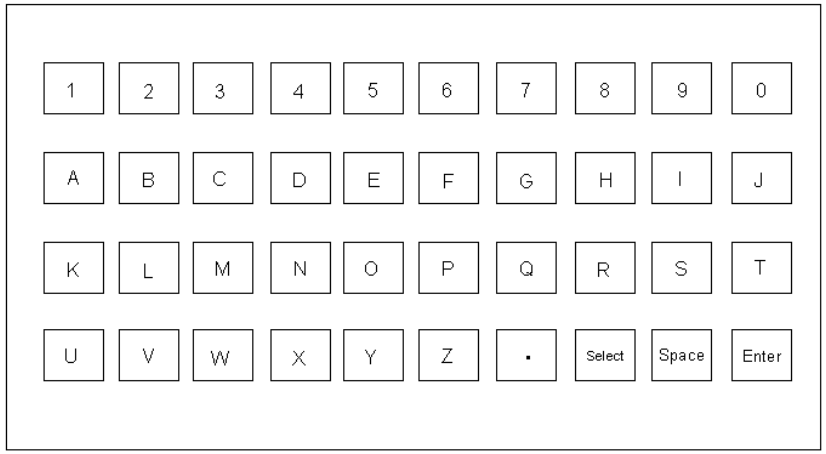


(Note: If needed the matrix can be expanded up to 60 keys)

Title 40 Keys Matrix KeyBoard using Pic16F84a		
Author a.ratti		
File cumenti\TinyCAD Examples\40_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



Keyboard default layout

Title Default layout		
Author a.ratti		
File igs\Alberto\Desktop\40_Keys_matrix_Default.dsn		Document 3
Revision 1.0	Date	Sheets 1 of 1

```
'*****************************************************************
'*  Name    : 40Ks_Kboard.BAS                                    *
'*  Author  : [a.ratti]                                          *
'*  Notice  : Copyright (c) 2008                                 *
'*          : All Rights Reserved                                *
'*  Date    : 14/11/2008                                         *
'*  Version : 1.0                                                *
'*  Notes   : Pic 16F84a                                         *
'*          :                                                    *
'*****************************************************************

TrisA = %00000000
TRISB = %00001111

Define OSC 4

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

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

UCase       var byte
Keyb        var byte
KbOut       var Byte    [40] ' Array to be loaded with ASCII code

high Tx   ' purge serial line
High RTS  ' disable RTS
Keyb = 0    ' reset variable
Ucase = 0   ' set uppercase
'-------------------------------------------------------------------------
'----- Loading ascii code into the array to define keys response -------
'-------------------------------------------------------------------------
KbOut[0]=0      'Null
KbOut[1]=49     '1
KbOut[2]=50     '2
KbOut[3]=51     '3
KbOut[4]=52     '4
KbOut[5]=53     '5
KbOut[6]=54     '6
KbOut[7]=55     '7
KbOut[8]=56     '8
KbOut[9]=57     '9
KbOut[10]=48    '0

KbOut[11]=65    'A
KbOut[12]=66    'B
KbOut[13]=67    'C
KbOut[14]=68    'D
KbOut[15]=69    'E
KbOut[16]=70    'F
KbOut[17]=71    'G
KbOut[18]=72    'H
KbOut[19]=73    'I
KbOut[20]=74    'J

KbOut[21]=75    'K
KbOut[22]=76    'L
KbOut[23]=77    'M
KbOut[24]=78    'N
KbOut[25]=79    'O
```

```
KbOut[26]=80    'P
KbOut[27]=81    'Q
KbOut[28]=82    'R
KbOut[29]=83    'S
KbOut[30]=84    'T

KbOut[31]=85    'U
KbOut[32]=86    'V
KbOut[33]=87    'W
KbOut[34]=88    'X
KbOut[35]=89    'Y
KbOut[36]=90    'Z
KbOut[37]=46    '.
KbOut[38]=26    'Sostitute
KbOut[39]=32    'Space
KbOut[40]=13    'Enter

'-------------------------------------------------------------------------
'-- You can change the above ASCII/Key association to suit your need ---
'-------------------------------------------------------------------------



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

 Loop:
 if KeyB>0 then '  If key pressed then Tx out ASCII character and Beep
 Low RTS    ' RTS flag enabled
 pause 20
 if KbOut[Keyb]<32 then Loop 'Ignore ascii code lower than space

 If KbOut[Keyb]=>65 then 'Select letter code skip if numbers or others
 KbOut[0] = KbOut[Keyb] + UCase 'If UCase = 0 then UpperCase
 else
 KbOut[0] = KbOut[Keyb]
 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
 pause 30
 low Buz
 pause 150
 endif
 pause 50

 toggle Led ' chip working

 '----------------------------- KeyScan ---------------------------------

 Keyb=0   ' reset variable

 '----------------------- First Row - keys from 1 to 10  --------------
 PortB =16
 If PortB>16 then       ' one button of the first row has been pressed
 Keyb = (PortB-16)       ' decode to which column it belongs
 Gosub Upper_Lower
 goto Loop
 endif
```

```
'------------------------ Second Row - keys from 11 to 20 -------------
PortB =32
If PortB>32 then       ' one button of the second row has been pressed
Keyb = (PortB-32) + 10  ' decode to which column it belongs
gosub Upper_Lower
goto Loop
endif

'------------------------ Third Row - keys from 21 to 30 ---------------
PortB =64
If PortB>64 then       ' one button of the third row has been pressed
Keyb = (PortB-64) + 20  ' decode to which column it belongs
gosub Upper_Lower
goto Loop
endif

'------------------------ Fourth Row - keys from 31 to 40 -------------
PortB =128
If PortB>128 then      ' one button of the fourth row has been pressed
Keyb = (PortB-128) + 30 ' decode to which column it belongs
gosub Upper_Lower
endif

goto Loop

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


'----------------------------------------------------------------------
'---------------- Select Uppercase or LowCase letters -----------------
'-------- Ucase = 32 then lowerCase  UCase = 0 then UpperCase ---------
'----------------------------------------------------------------------
Upper_Lower: ' Set and reset UCase variable
If  KbOut[Keyb]=26 then
If UCase=0 then
Ucase=32
else
UCase=0
endif
endif
return



end
```