

```
'{$$STAMP BS2p, GetName.bsp, GetNum2.bsp, memory.bsp, edit.bsp, delete.bsp, sort.bsp}
'{$PBASIC 2.5}
```

```
addr      VAR      Word
T_addr    VAR      Word
db        VAR      Bit          ' Debounce bit for use by keyScan.
press     VAR      Bit
keyB      VAR      Byte
key       VAR      keyB.LOWNIB  ' Key number 0-15.
keyB2     VAR      Byte
key2      VAR      keyB2.LOWNIB ' Key number 0-15.
Row       VAR      Nib          ' Counter used in scanning keys.
cols      VAR      INB          ' Input states of pins P4-P7.
ReadKey   VAR      Byte
loops     VAR      Word          ' Variable TO STORE counted cycles.
counter   VAR      Word
location  VAR      Byte
eeAddr    VAR      Word
Record    VAR      Word
cursor    VAR      Word
inVal     VAR      Byte
ReadIn    VAR      Byte
dial_n    VAR      Byte
```

```
***** INITIALIZING LCD DISPLAY *****
```

```
AUXIO
LCDCMD 1,%00110000 'Send WAKEUP sequence to LCD
PAUSE 10           'These pauses are necessary to meet the LCD specs
LCDCMD 1,%00110000
PAUSE 1
LCDCMD 1,%00110000
PAUSE 1
LCDCMD 1,%00100000 'Set buss to 4-bit mode
LCDCMD 1,%00101100 'Set to 2-line mode with 5x10 font
LCDCMD 1,%00001000 'Turn display off
LCDCMD 1,%00000110 'Set to auto-increment loop (no display shift)
LCDCMD 1,%00000001 'Clear the display
```

```
DEBUG "reseted ", DEC ReadKey,CR
LCDOUT 1, %10000000+0, ["INTELLIGENT MINI-DIR"]
LCDOUT 1, %10000000+20, [" [esc] to MAIN MENU "]
LCDOUT 1, %10000000+64, ["-----"]
LCDCMD 1, %00001111
```

```
STORE 0
READ 5, loops
IF loops = $aa THEN start
' [ DIAL FIRST ] ===== made use of pulse dialing ===
' -----
```

```
again:
PAUSE 50
AUXIO
HIGH 0
MAINIO
FOR row = 0 TO 3          ' Scan rows one at a time.
  LOW row                ' Output a 0 on current row.
  key2= ~cols            ' Get the inverted state of column bits.
  key2= NCD key2         ' Convert to bit # + 1 with NCD.
  IF key2<> 0 THEN push3 ' No high on cols? No key pressed.
  INPUT row              ' Disconnect output on row.
NEXT row                 ' Try the next row.
db = 0                   ' Reset the debounce bit.
AUXIO
LOW 0
IF press = 0 THEN again
press = 0
IF (keyB2=10) THEN start
IF (keyB2=11) THEN up
IF (keyB2=12) THEN down
IF (keyB2=13) THEN
  AUXIO
  HIGH 10
  PAUSE 1000
  readIn = 9             ' for Innovex, with extension 9
  GOSUB dial             ' for Innovex, with extension 9
  FOR counter = 0 TO (ReadKey-1)
    READ 10+counter, ReadIn
    GOSUB dial
  NEXT counter
  PAUSE 7500
  AUXIO
  LOW 10
  ReadKey = 0
  GOTO again
ELSEIF (keyB2 <> 10) AND (keyB2 <> 14) AND (keyB2 <> 15) THEN
  WRITE 10+ReadKey,keyB2
  LCDOUT 1, %10000000+84+ReadKey, [DEC keyB2]
  PAUSE 5
  ReadKey = ReadKey + 1
ENDIF
GOTO again
```

```

=====
push3:
  IF db = 1 THEN done3      ' Already responded to this press, so done.  ''
  db = 1: press = 1        ' Set debounce and keypress flags.          ''
  key2 = (key2-1)+(row*4)   ' Add column (0-3) to row x 4 (0,4,8,12).  ''
  LOOKUP key2,[1,2,3,10,4,5,6,11,7,8,9,12,14,0,15,13],keyB2          ''

done3:
  ''
  INPUT row                 ' Disconnect output on row.          ''
RETURN                     ' Return to again from keyScan.        ''
=====

up:
down:
dial:
  IF (ReadIn = 0) THEN
    ReadIn = 10
  ENDIF
  FOR dial_n = 1 TO ReadIn
    LOW 10
    GOSUB calling
    HIGH 10
    GOSUB calling
  NEXT
  FOR dial_n = 1 TO 3
    GOSUB calling
  NEXT
RETURN

  calling:
    LCDOUT 1, %10000000+94, ["CALLING..."]
    PAUSE 35
    LCDOUT 1, %10000000+94, ["      "]
    PAUSE 35
  RETURN
' *****
' *****
' the rest is fine....

```