

```

*****
'*
'*      PIC 16F648A @ 4MHz
'*      program to read a PC AT keyboard
'*      and output equivalent ASCII codes to Interak computer
'*
'*      Alan Paton                      revised January 2009
'*
'*      connected as follows:
'*      KeyBd Clock...PortA.0          KeyBd Data...PortA.1
'*      PortB.0-6 ASCII characters    PortB.7 Strobe/Latch
'*
*****

'this code is the minimum required for PC keyboard to ASCII
'conversion. It does not operate the Caps Lock LED or use any
'commands which require writing data to the keyboard.

'this code added 20.01.2009 for 16F648A (replacement for 16F84)
    CMCON = 7          'make Port A digital I/O
    @ device INTRC_OSC_NOCLKOUT 'use internal oscillator

'
'  define variables
PBIT    VAR    BYTE    'Parity counter
D       VAR    BIT(12) 'serial data bit array
SC      VAR    BYTE    'Scan Code
CNT     VAR    BYTE    'control key flag
J       VAR    BYTE    'counter
CLK     VAR    PortA.0 'Clock
DAT     VAR    PortA.1 'Data
CL      VAR    BYTE    'caps lock flag
SH      VAR    BYTE    'shift flag
AC      VAR    BYTE    'ASCII code
LN      VAR    BYTE    'Lookdown List No.
LB      VAR    BYTE    'last byte flag

'initialise
    PAUSE 1000          'wait for kybd start-up
    TrisB = 0          'set PortB to output
    TrisA = 0          'set PortA to output
    SH=0              'shift flag
    CNT=0             'control flag
    CL=0             'caps-lock flag
    GOTO SCAN

***** START *****
START:  IF SC=$12 OR SC=$59 THEN 'shift key up
        SH=0                  'reset shift
    ENDIF
    IF SC=$14 THEN           'control key up
        CNT=0                 'reset control flag
    ENDIF
    LB=0                     'clear last-byte flag

'setup ports to read keyboard
***** SCAN shifts serial Scan Code into SC *****
SCAN:   SC=0                  'clear scan code
        TrisA.1=1            'set DAT to input
        TrisA.0=1            'set CLK to input

'wait for clock transition

```

```

LP1:    IF CLK=0 THEN           'make sure clock is high
        GOTO LP1
        ENDIF
LP2:    IF CLK=1 THEN           '(clock idles high)
        GOTO LP2               'wait for CLK to go low
        ENDIF
'CLK now low - Start Bit of scan code
        FOR J=0 TO 9           'get scan code
LP3:    IF CLK=0 THEN           'make sure clock is high
        GOTO LP3
        ENDIF
        D(J)=DAT               'read data bit
        NEXT J
'shift bits into variable SC - start bit is not in array D
        FOR J=7 TO 0 STEP -1   'D(8) & D(9) are parity & stop bits
        SC=SC << 1            'shift 1 place left
        SC=SC+D(J)            'set bit 0 of SC lsb=0 msb=7
        NEXT J
        TrisA.0=0             'set CLK to output
        CLK=0                  'and hold it
'scan code now in SC
        IF LB=2 THEN           'last byte of up-code - don't display
        GOTO START
        ENDIF
        IF SC=$F0 THEN         'check for key-up code
        LB=2                   'set last byte flag
        GOTO SCAN
        ENDIF
        IF SC=$12 OR SC=$59 THEN 'left & right shift keys
        SH=$80                 'set shift flag
        GOTO SCAN
        ENDIF
        IF SC=$58 THEN         'caps lock
        GOSUB CPLOCK
        GOTO SCAN
        ENDIF
        IF SC=$E0 THEN         'extended keys
        GOTO SCAN              'get next byte
        ENDIF
        IF SC=$14 THEN         'left Control key (& right..$E0 $14)
        CNT=2                   'set flag
        GOTO SCAN
        ENDIF

'and into Scan Code to ASCII conversion
'***** Convert to ASCII code *****
'scan code in SC
ASCII:  IF SC=$5A THEN         'is it CR ?
        AC=$0D
        GOTO ASOUT            'yes- send it
        ENDIF
        SC=SC+SH              'SH=80H if shift key down (0 if not)
        LN=$FF                'set List No. to high value
'convert to ASCII (codes from 20H to 7FH)
        LOOKDOWN SC, [$29,$96,$9E,$5D,$A5,$AE,$BD,$52,$C6,$C5,$BE,$D5,
        $41,$4E,$49,$4A,$45,$16,$1E,$26,$25,$2E,$36,$3D,$3E,$46,$CC,$4C,$C1,$55,
        $C9,$CA,$D2,$9C,$B2,$A1,$A3,$A4,$AB,$B4,$B3,$C3,$BB,$C2,$CB,$BA,$B1,$C4,
        $CD,$95,$AD,$9B,$AC,$BC,$AA,$9D,$A2,$B5,$9A,$54,$61,$5B,$B6,$CE,$A6,$1C,
        $32,$21,$23,$24,$2B,$34,$33,$43,$3B,$42,$4B,$3A,$31,$44,$4D,$15,$2D,$1B,
        $2C,$3C,$2A,$1D,$22,$35,$1A,$D4,$E1,$DB,$DD,$66],LN

```

```
IF LN=$FF THEN           'Scan code not found
  GOTO NUMPAD           'now check number pad
ENDIF
'matching scan code has been found
IF CNT=2 THEN           'is control key pressed ?
  AC=LN-$40           'get ASCII code
  GOTO ASOUT           'send it
ENDIF
IF CL=0 THEN           'Caps Lock OFF
  GOTO AS2           '& matching code found
ENDIF
'CAPS LOCK ON - check for the following conditions
IF LN>$40 AND LN<$5B THEN 'caps lock on & lower case selected
  AC=LN
  GOTO ASOUT
ENDIF
IF LN=$1B THEN           'caps lock on & (semi)colon selected
  AC=$3A           'ASCII code for colon
  GOTO ASOUT
ENDIF
GOTO AS2           'send out to computer keyboard port
'check number pad
NUMPAD: LOOKDOWN SC, [$70,$69,$72,$7A,$6B,$73,$74,$6C,$75,$7D,$7C,$7B,
$79,$71,$0D,$76],LN
IF LN=$FF THEN
  GOTO SCAN           'matching code not found
ENDIF
LOOKUP LN, [$30,$31,$32,$33,$34,$35,$36,$37,$38,$39,$2A,$2D,$2B,
$2E,$09,$1B],AC
GOTO ASOUT
AS2: AC=LN+$20           'convert list No. to ASCII
ASOUT: PortB.7=1           'strobe - normally high - pulse low
PAUSE 2
PortB=AC           'send ASCII code to Port B
PAUSE 2           'bit 7 (strobe) is zero
PortB.7=1
GOTO SCAN

END
'***** Caps Lock *****
CPLOCK: IF CL=0 THEN           'toggle caps lock
  CL=4           '4 is bit2 0000 0100
ELSE
  CL=0
ENDIF
RETURN
```