Interfacing a Switch to a Computer

Port AD Initialization Software

```c
void PortAD_Init(void) {
  ATDDIEN |= 0x03; // PAD1-0 digital I/O
  DDRAD &= ~0x03; // PAD1-0 inputs
  PPSAD |= 0x02; // pull-down on PAD1
  PPSAD &= ~0x01; // pull-up on PAD0
  PERAD |= 0x03; // enable pull-up and pull-down
}
```

Switch Bounce

Hardware Debouncing Using a Capacitor

Hardware Debouncing Using a Capacitor (cont)
Software Debouncing

```c
void Key_WaitPress(void)
{
    while(PTT&0x08); // PT3=0 when pressed
    Timer_Wait10ms(1); // debouncing
}
```

```c
void Key_WaitRelease(void)
{
    while((PTT&0x08)==0); // PT3=1 -> released
    Timer_Wait10ms(1); // debouncing
}
```

```c
void Key_Init(void)
{
    Timer_Init();
    DDRT &=~0x08; // PT3 is input
}
```

Another Approach to Software Debouncing

```c
void Key_Init(void)
{
    TIOS |= 0x20; // enable OC5 (see Chapter 6)
    TSCR1 = 0x80; // enable
    TSCR2 = 0x01; // 500 ns clock
    DDRT &=~0x08; // PT3 is input

    unsigned char Key_Read(void)
    {
        unsigned char old;
        old = PTT&0x08; // Current value
        TC5 = TCNT+20000; // 10ms delay
        TFLG1 = 0x20; // Clear CF
        while((TFLG1&0x20)==0) // 10ms?
        {
            if(old<>(PTT&0x08)) // changed?
            {
                old = PTT&0x08; // New value
                TC5 = TCNT+20000; // restart delay
            }
        }
        return(old);
    }

    // 4 by 4 Scanned Keypad
```
There are two steps to scan a particular row:
- Select that row by driving low while other rows are not driven.
- Read the columns to see if any keys are pressed in that row (0 means key pressed, 1 means not pressed).

The scanned keyboard operates properly if:
- No key is pressed.
- Exactly one key is pressed.
- Exactly two keys are pressed.

---

Software for a Matrix Scanned Keypad

```c
/* Returns ASCII code for key pressed,
   Num is the number of keys pressed
   both equal zero if no key pressed */
unsigned char Key_Scan(short *Num){
    RowType *pt; unsigned char column,key;
    short j;
    (*Num)=0; key=0; // default values
    pt=ScanTab[0];
    while(pt->direction){
        DDRT = pt->direction; // one output
        column = PTT; // read columns
        for(j=3; j>=0; j--){
            if((column&0x01)==0){
                key = pt->keycode[j];
                (*Num)++;
            }
            column>>=1; // shift into position
            pt++; }
    return key;}
```