Basic Approaches to Interfacing Multiple Keys

<table>
<thead>
<tr>
<th>Row</th>
<th>Out3</th>
<th>Out2</th>
<th>Out1</th>
<th>Out0</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>HiZ</td>
<td>HiZ</td>
<td>HiZ</td>
</tr>
<tr>
<td>2</td>
<td>HiZ</td>
<td>0</td>
<td>HiZ</td>
<td>HiZ</td>
</tr>
<tr>
<td>1</td>
<td>HiZ</td>
<td>HiZ</td>
<td>0</td>
<td>HiZ</td>
</tr>
<tr>
<td>0</td>
<td>HiZ</td>
<td>HiZ</td>
<td>HiZ</td>
<td>0</td>
</tr>
</tbody>
</table>

Sixteen-Key Electronic Piano

Hardware for Generating Interrupts

Multiplexed/Demultiplexed Scanned Keyboard

Software for Multiplexed Keyboard

```c
unsigned char Key; // current pattern
unsigned char PreviousKey; // 10 ms ago
#define period 20000 // 10 ms
unsigned char KeyScan(void){
    unsigned char key,row;
    key=0; // means no key pressed
    for(row=0;row<16;row++)
        PORTJ=row<<4; // Select row
    if((PORTJ&0x0F)!=0x0F){
        key=PORTJ^0x0F;
    }
    return(key);
}
```
Software for Multiplexed Keyboard (cont)

```c
void Ritual(void){
    asm(" sei"); // make atomic
    DDRJ=0xF0;
    PreviousKey=Key=KeyScan(); // read
    TMSK1|=0x20; // Arm OC5
    TIOS|=OC5; // enable OC5
    TSCR|=0x80; // enable
    TMSK2=0x32; // 500 ns clock
    TC5=TCNT+wait;
    TFLG1=0x20; // clear OC5F
    asm(" cli");
}
```

#pragma interrupt_handler TOC5handler()

```c
void TOC5handler(void){
    unsigned char NewKey;
    NewKey=KeyScan(); // Current pattern
    if(NewKey==PreviousKey) Key=NewKey;
    PreviousKey=NewKey;
    TOC5=TOC5+period;
    TFLG1=0x20;} // ack OC5F
```

Interfacing Multiple LEDs

![Diagram of direct vs. multiplexed LED interfacing]

Typical Voltage/Current Response of a LED

<table>
<thead>
<tr>
<th>Parameter</th>
<th>red</th>
<th>green</th>
<th>yellow</th>
<th>orange</th>
<th>units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max power</td>
<td>55</td>
<td>75</td>
<td>60</td>
<td>75</td>
<td>mW</td>
</tr>
<tr>
<td>Peak current</td>
<td>160</td>
<td>100</td>
<td>80</td>
<td>100</td>
<td>mA</td>
</tr>
<tr>
<td>Max current</td>
<td>25</td>
<td>25</td>
<td>20</td>
<td>25</td>
<td>mA</td>
</tr>
</tbody>
</table>

Calculating the Resistor Value

![Diagram of resistor calculation for LED]
Seven-Segment LED Interfaces (Common-Cathode)

Seven-Segment LED Interfaces (Common-Anode)

Scanned Seven-Segment LED Interface

Circuit Used to Scan a LED Interface

Software for Scanned LED Display

Software for Scanned LED Display

// PT7-PT0 output, 7 bit pattern
// PM2-PM0 output, selects LED digit
unsigned char code[3]; // binary codes
static unsigned char select[3] = {4, 2, 1};
unsigned short index; // 0, 1, 2

void LED_Init(void) {
  asm sei // make atomic
  index = 0;
  DDRT = 0xFF; // outputs 7 segment code
  DDRM |= 0x03; // outputs select LED
  TIE |= 0x20; // Arm OC5
  TIOS |= 0x20; // enable OC5
  TSCR1 = 0x80; // enable
  TSCR2 = 0x01; // 500 ns clock
  T5 = TCNT+10000;
  asm cli }

void interrupt 13 TC5handler(void)
{
  TFLG1 = 0x20; // Acknowledge
  TC5 = TC5+10000; // every 5 ms
  PTM = select[index]; // which LED?
  PTT = code[index]; // enable
  if (++index == 3) index = 0;
  asm("cli");
}
Scanned LED Interface Using Decoder

For MC9S12C32, replace PB with PT.

Software for Multiplexed LED Display

```c
unsigned short Global; // 12-bit packed BCD
const struct LED
{ unsigned char enable; // select
    unsigned char shift; // bits to shift
    const struct LED *Next; // Link
} LEDTab[3]={
    ( 0x04, 8, &LEDTab[1] ), // Most sig
    ( 0x02, 4, &LEDTab[2] ),
    ( 0x01, 0, &LEDTab[0] )}; // least sig
LEDPtr Pt; // Points to current digit

void LED_Init(void) {
    asm sei // make atomic
    DDRT = 0xFF; // outputs to LED’s
    Global = 0;
    Pt=&LEDTab[0];
    TIE |= 0x20; // Arm OC5
    TIOS |= 0x20; // enable OC5
    TSCR1 = 0x80; // enable
    TSCR2 = 0x01; // 500 ns clock
    TC5 = TCNT+10000;
    asm cli }
```

void interrupt 13 TC5handler(void){
    TFLG1 = 0x20; // Acknowledge
    TC5 = TC5+10000; // every 5 ms
    PTT = (Pt->enable) + (Global>>(pt->shift))<<4);
    Pt = Pt->Next; }

Extensions to Multiple Digits

Two issues to consider as number of digits is increased:
- Scan frequency - for display to "look" continuous, each digit must be updated faster than 60 Hz.
  - interrupt rate = 60 Hz × #digits
- Duty cycle - this decreases as digits added, so must increase instantaneous current.
  - instantaneous current = desired current × #digits
- Ratio of maximum instantaneous current to desired LED current determines maximum number of digits.

Integrated IC Interface for LED Digits
Data Timing of Integrated LED Controller

Software for Integrated LED Display

// PM4/MOSI = MC14489 DATA IN
// PM5/SCLK = MC14489 CLOCK IN
// PM3 (simple output) = MC14489 ENABLE
void LED_Init(void) {
    DDRM |= 0x38; // outputs to MC14489
    SPICR1 = 0x50;
    SPICR2 = 0x00; // regular drive
    SPIBR = 0x01; // 1MHz SCLK
    PTM |= 0x08; // ENABLE=1
    PTM &=~0x08; // ENABLE=0
    SPIDR= 0x01; // hex format
    while((SPISR&0x80)==0){};
    PTM |=0x08;} // ENABLE=1

void LED_out(unsigned char data[3]){
unsigned char dummy;
PTM &=~0x08; // ENABLE=0
while((SPISR&SPTEF)==0); // wait for transmit empty
SPIDR = data[2]; // send MSbyte
dummy = SPIDR; // clear SPIF
while((SPISR&SPTEF)==0); // wait for transmit empty
SPIDR = data[1]; // send middle byte
dummy = SPIDR; // clear SPIF
while((SPISR&SPTEF)==0); // wait for transmit empty
SPIDR = data[0]; // send LSbyte
dummy = SPIDR; // clear SPIF
Timer_Wait(10); // wait for SPI output completion
PTM |=0x08; // ENABLE=1

LCD Fundamentals

- LCD display consume less power than LED displays.
- LCDs are more flexible in sizes and shapes, allowing for combination of numbers, letters, words, and graphics.
- Uses liquid-crystal material that behaves as a capacitor.
- While LED converts electric power to emitted optical power, LCD uses AC voltage to change light reflectivity.
- Light energy is supplied by room or separate back light.
- Display controlled by altering reflectivity of each segment.
- Disadvantage is that they have slow response time, but typically fast enough for human perception.

Basic Idea of a Liquid Crystal Interface

Direct Interface of a LCD
Latched Interface of a LCD

Artwork for 8-Segment LCD Digits

Interface of a 48-Segment LCD Display