

1. 独立键盘

sbit DU = P2^6;

sbit WE = P2^7;

sbit key-S2 = P3^0; 独立按键S2

uchar code table[] = { 0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, }
uchar num;

{
1° 按下之后不变化 松开变化
2° 按下之后变化

void main()
{

WE = 1;

P0 = 0xfe;

WE = 0;

while(1)

{ if (key-S2 == 0)

{

num++;

if (num == 10)

num = 0;

}

DU = 1;

P0 = table[num];

DU = 0;

}



if (key-S2 == 0)

{ delay(20); 按键消抖

if (key-S2 == 0)

{ num++;

if (num == 10)

num = 0;

本行检测 while (!key-S2);

}

2. 矩阵键盘

列扫描和行扫描

1° 列扫描时先把接在列上面的所有IO口拉高，接在行上的所有IO口置低

2° 行扫描 ····· 行 ····· 列 ·····

3° 位置 = 行 + 列

```
sbit DU = P2^6;
```

```
sbit WE = P2^7;
```

```
uchar code table[] = {····};
```

```
uchar keyValue = 20; 按键值, 显示 '-'
```

```
void keyScan( )
```

```
{ P3 = 0xf0; 列扫描
```

```
if ( P3 != 0xf0)
```

```
{ delay(10);
```

```
if ( P3 != 0xf0)
```

```
{ switch(P3) 判断哪一列被按下
```

```
case 0xe0: keyValue = 0; break;
```

```
case 0xd0: keyValue = 1; break;
```

```
case 0xb0: keyValue = 2; break;
```

```
case 0x70: keyValue = 3; break;
```

```
}
```

```
P3 = 0x0f; 行扫描
```

```
switch(P3) 判断哪一行被按下
```

```
{
```

```
case 0x0e: keyValue = keyValue; break;
case 0x0d: keyValue = keyValue + 4; break;
case 0x0b: keyValue = keyValue + 8; break;
case 0x07: keyValue = keyValue + 12; break;
}
```

```
while ( P3 != 0x0f) 松手检测
```

```
}
```

```
}
```

```
void main() {
```

```
WE = 1;
PO = 0xfe;
```

```
WE = 0;
```

```
while(1)
```

```
{ keyScan(); 4*4矩阵键盘扫描
```

```
DU = 1;
```

```
PO = table[keyValue];
```

```
DU = 0;
```

```
}
```

```
}
```