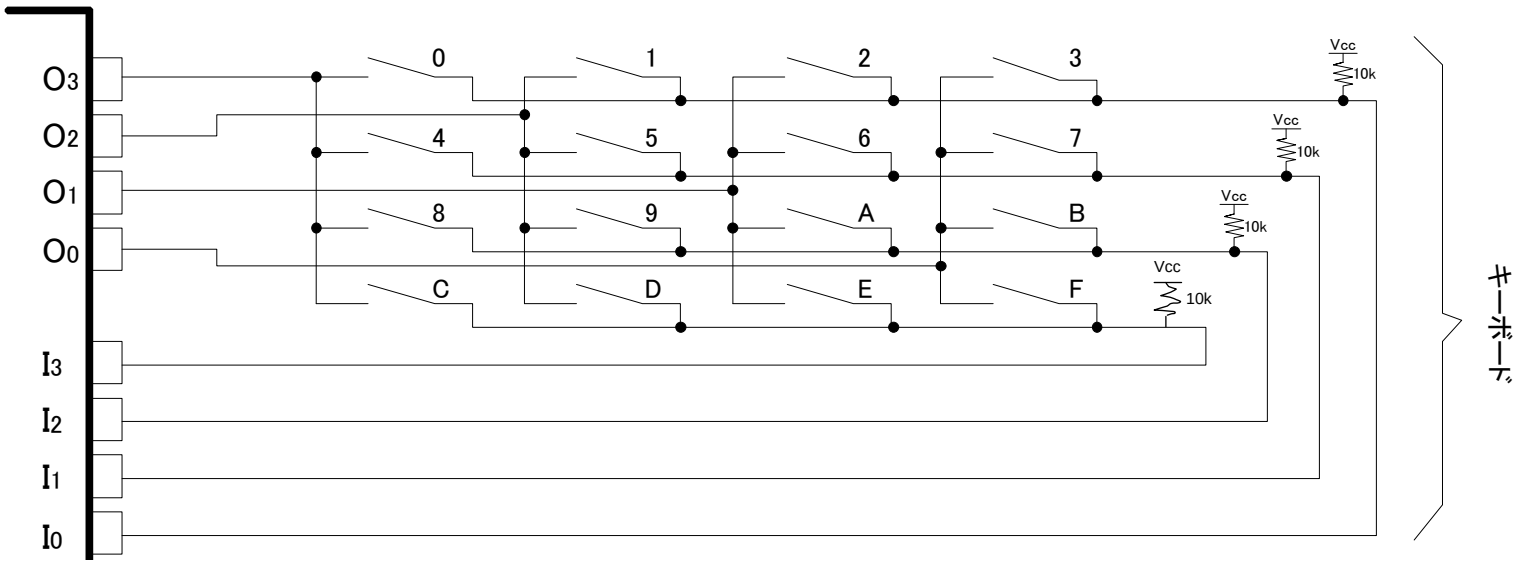

上記のようにキーマトリックスを設計したとします. O は出力ピンで論理 0 または論理 1 のデジタル信号を電圧として出力します. (論理 1 の時は電源電圧 Vcc, 論理 0 の時は GND) I は入力ピンで電圧をデジタル信号として受け入れます．しかし，上記の図でスイッチをどれも押さないと入力ピンには Vcc でも GND でもない, 電位差が計測できない状態が入力されます．これを防ぐために右上にあるようにプールアップ抵抗をつけてスイッチが押されていない時には Vcc 即ち論理 1 が入力されるようにします．従って，スイッチを押したときには論理 0 が入力されるようにソフトウェアを設計しなければなりません．(スイッチを押している時とそうじゃない時が区別されるように)

キーマトリックスを使用する理由は使用する入力または出力ピンを最低限に抑えたいからです．上の図のように 4 つの出力と 4 つの入力ピンを使えば 16 個のキー（スイッチ）を使うことが可能ですが，もし入力ピンだけの単純な設計だと 16 個のキーに対して 16 個の入力ピンを割り当てなければなりません．この差はキーが増えればもっと広がります．100 個のキーを区別するためには 10 個の入力ピンと 10 個の出力ピンを使えば済みますが，入力ピンだけだと 100 個の入力ピンが必要となります.

上記のようなキーマトリックスの場合は出力ピンに 1110b, 1101b, 1011b, 0111b を順番に循環しながら出力します（b は 2 進数を意味する）．1110b を出力ピンに出力すると O0 からは論理 0 に該当する GND の電圧，即ち 0V が出力されます．O3~2 からは論理 1 に該当する Vcc 電圧が出力されます．もし 1101b を出力している時に‘６’キーが押されていたとします．そうすると入力ピン I3~I0 には 1101b が入力されます．もし‘A’キーが押されていたとすると，入力ピン I3~I0 には 1011b が入力されます．もし‘３’キーが押されていれば入力ピン I3~I0 には 1111b が入力されます（キーが押されていないのと一緒）. しかし, 出力ピンからはプログラムによって高速で 1110b, 1101b, 1011b, 0111b が循環されながら出力されるので出力と入力の組み合わせを確認することでどのキーが押されているかが確認できます.

例えばあるキーが押された時にそのキーに該当する文字を出力するプログラムを作成するとします．このプログラムでは inp が入力ピンに割り当てられていて入力ピンに入った入力は inp に格納されることとします．また，outp は出力ピンに割り当てられ，outp に格

納した値は出力ピンから出力されることとします．そうするとプログラムは以下のようになります．このプログラムの動作が理解できない人は値を 2 進数で書きながら確認すると理解できると思います．

```
前略
int inp, outp;

void main ( ) {
    outp = 1;
    while(1) {
        out( 0x0f – outp );
        inp = 0x0ff & in( );
        if (( inp & 0x0f ) != 0x0f )
            key( outp, inp );
        outp = outp * 2;
        if ( outp == 16 )
            outp = 1;
    }
    中略
}
```

```
void key ( int outp, int inp ) {
    if (( outp == 0x01 ) && ((inp & 0x01) == 0 ))
        printf(“3”);
    if (( outp == 0x01 ) && ((inp & 0x02) == 0 ))
        printf(“7”);
    中略
    if (( outp == 0x04 ) && ((inp & 0x08) == 0 ))
        printf(“D”);
    中略
}
```