

KeyPad(4×4)

概要

データを入力するとき、プッシュスイッチでは、1~2桁くらいなら特に不便を感じないのですが、3桁以上になると使い勝手が悪くなります。

そこで登場するのがKeyPad(キーパッド)ですKeyPadは、特定の用途に特化した小型のキーボードのことで、今回は、4×4のキーマトリクスからなるものを使用し、そのキースキャンについて動作確認してみました。

動作原理

複数のキースwitchの状態を調べるためには、本来ならスイッチの数だけ入力ポートが必要となります。しかし、キーボードのように大量のキーでは入力ポートが不足してしまいます。そこで、このような時に使われる方法が、**キーマトリックススキャン**です。

例えば、今回のように16キー必要な場合には、16の入力ポートが必要となりますが、マトリクスにすることで半分の8ポート(4入力ポート、4出力ポート)で済みます。

キースキャンには、プルアップ方式とプルダウン方式があります。

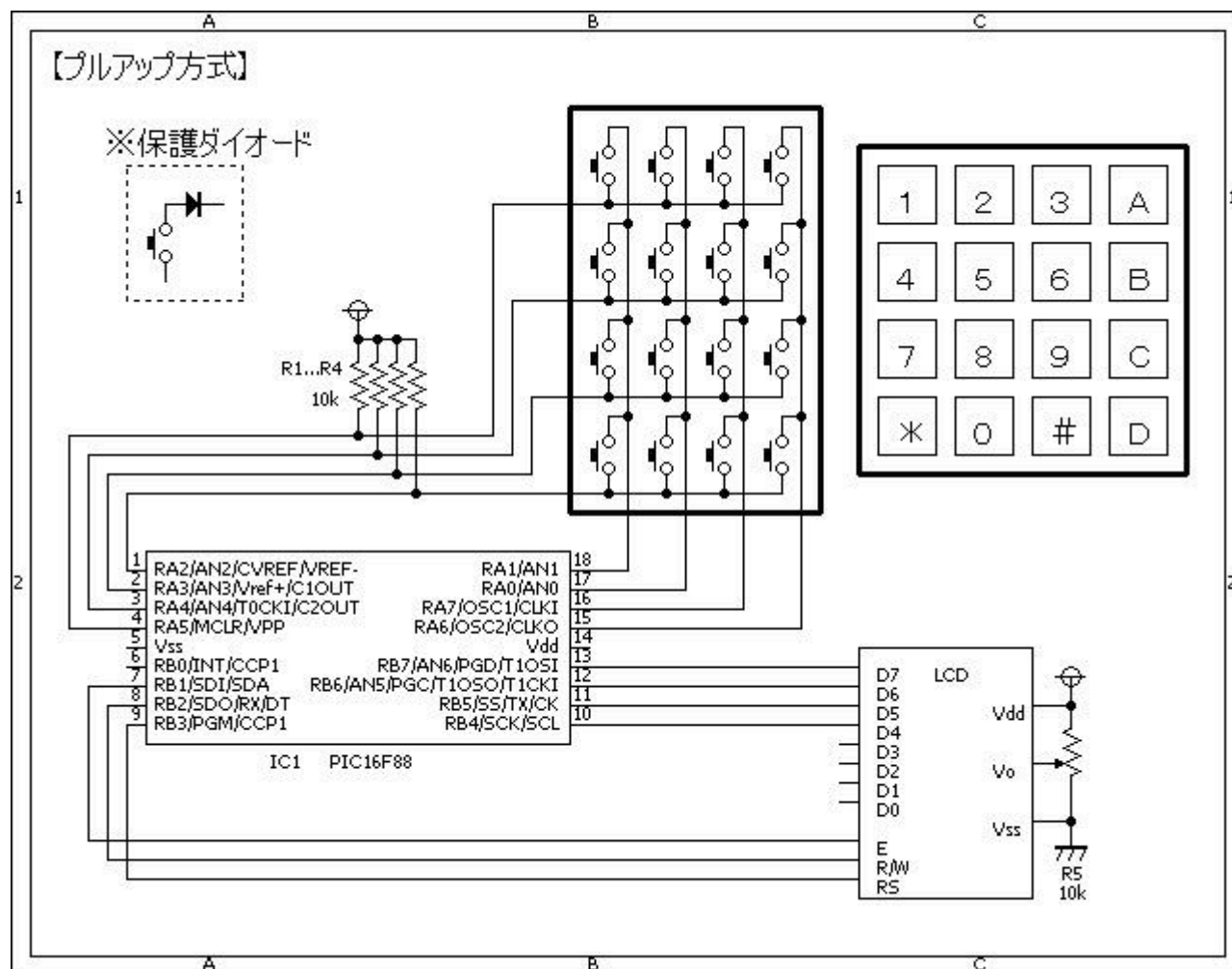
<プルアップ方式> 4つの入力ポートをプルアップ“1”し、4つの出力ポートを順次“0”にしながら、キー押下“0”がされているかをサーチします。

<プルダウン方式> 4つの入力ポートをプルダウン“0”し、4つの出力ポートを順次“1”にしながら、キー押下“1”がされているかをサーチします。

尚、キーのチャタリング除去は、ソフトウェアで考慮しました。同一キーの押下が、100μSEC間隔で10回連続した場合のみ押下されたものと見なします。

回路図

複数キー押下時を考慮すると保護ダイオードが必要となりますが、単一キーの押下のみとするのであれば不要となります。



ソースコード

プルアップ方式のみ掲載します。少しの修正でプルダウン方式に変更することができます。

[keyPad.c](#)

```
//*****
*
/*
□□□□□□□□□□ プルアップ方式 ) >
*/
//*****
*

unsigned short keyPad_keySearch()
{
    unsigned short kp;
    //
    kp = 0x00;
    //
    PORTA.F6 = 0;
    switch (PORTA & 0b00111100) {
    case 0b00111000:
```

```
        kp = 'D';
        break;
    case 0b00110100:
        kp = 'C';
        break;
    case 0b00101100:
        kp = 'B';
        break;
    case 0b00011100:
        kp = 'A';
        break;
    }
    PORTA.F6 = 1;
    //
    PORTA.F7 = 0;
    switch (PORTA & 0b00111100) {
    case 0b00111000:
        kp = '#';
        break;
    case 0b00110100:
        kp = '9';
        break;
    case 0b00101100:
        kp = '6';
        break;
    case 0b00011100:
        kp = '3';
        break;
    }
    PORTA.F7 = 1;
    //
    PORTA.F0 = 0;
    switch (PORTA & 0b00111100) {
    case 0b00111000:
        kp = '0';
        break;
    case 0b00110100:
        kp = '8';
        break;
    case 0b00101100:
        kp = '5';
        break;
    case 0b00011100:
        kp = '2';
        break;
    }
    PORTA.F0 = 1;
    //
    PORTA.F1 = 0;
    switch (PORTA & 0b00111100) {
    case 0b00111000:
```

```

        kp = '*';
        break;
    case 0b00110100:
        kp = '7';
        break;
    case 0b00101100:
        kp = '4';
        break;
    case 0b00011100:
        kp = '1';
        break;
    }
    PORTA.F1 = 1;
    //
    return (kp);
}

//*****
*

unsigned short keypad_GetKey()
{
    unsigned short newkp, oldkp, cnt;
    //
    oldkp = 0x00;
    //
    while (1) {
        for (cnt = 0; cnt < 10; cnt++) {
            newkp = keypad_keySearch();
            if (newkp != oldkp)
                cnt = 0;
            oldkp = newkp;
            Delay_us(100);
        }
        return (newkp);
    }
}

//*****
*

void main()
{
    unsigned short kp, l_cnt, c_cnt;
    //
    OSCCON = 0b01110000; // クロックを8Mhzに設定する。
    ANSEL = 0b00000000; // □□□変換は使用しない。
    TRISA = 0b00111100;
    //
    Lcd_Custom_Config(&PORTB, 7, 6, 5, 4, &PORTB, 3, 2, 1);
    Lcd_Custom_Cmd(LCD_CURSOR_OFF);

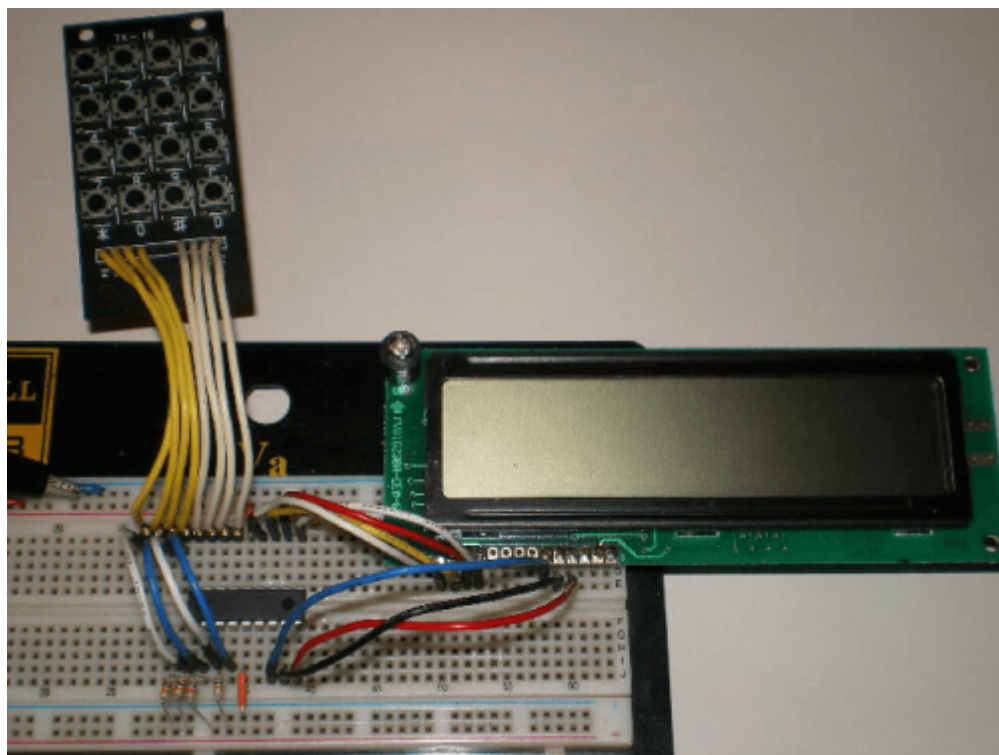
```

```
Lcd_Custom_Cmd(LCD_CLEAR);
//
PORTA.F6 = 1;
PORTA.F7 = 1;
PORTA.F0 = 1;
PORTA.F1 = 1;
//
l_cnt = 1;
c_cnt = 1;
//
while (1) {
    kp = keyPad_GetKey();
    if (kp == 0x00)
        continue;
    Lcd_Custom_Chr(l_cnt, c_cnt, kp);
    c_cnt++;
    if (c_cnt == 17) {
        c_cnt = 1;
        l_cnt++;
        if (l_cnt == 3)
            l_cnt = 1;
    }
    Delay_ms(500);
}
}

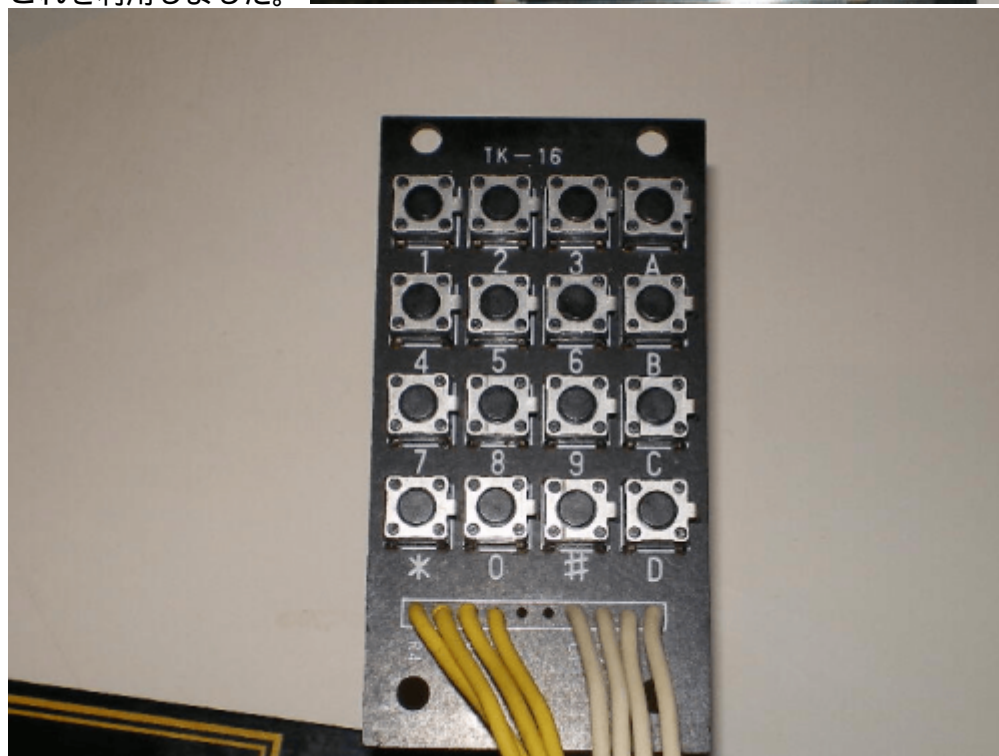
//*****
*
```

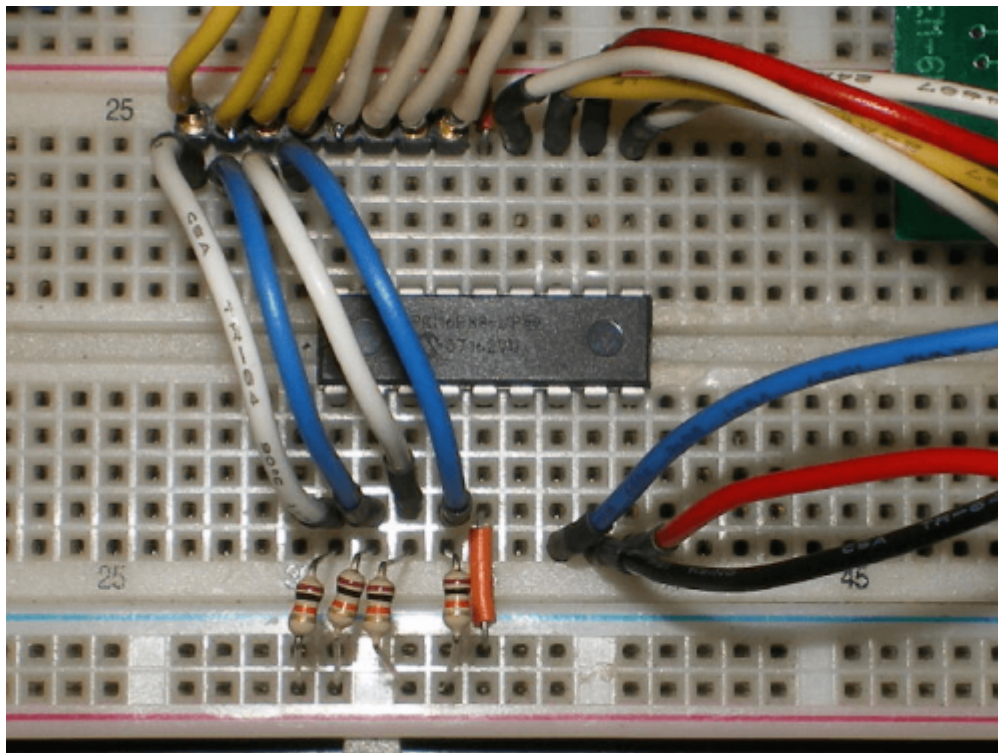
動作確認

以前に、DTMF(Dual-Tone Multi-Frequency)で使った□KeyPad(キーパッド)が、手元にありましたので、



これを利用しました。





KeyPadで押下したキーに対応した文字をLCDへ順次表示させて見ました。



如何ですか？これでデータ設定が、し易くなりますね 😊

著作権表示 copyright notice

このページは稲崎様の閉鎖したHPのコピーで、著作権は稲崎様にあります。[詳細](#) This page is a copy of Mr. Inasaki's closed website, and the copyright is held by him.[Details](#)

From:

<http://deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://deepsky.jp/wiki/doku.php?id=elechobby:picdic:pic16f88:97>

Last update: **2020/05/01 17:44**

