

簡易電卓(PS/2テンキー)

概要

PS/2テンキーは、数値を入力するインターフェイスとして優れています。例えば、

- 発振器を作った場合に、`1000000`+`Enter`と入力すれば`1MHz`を発振する。
- 可変電源を作った場合に、`12.5`+`Enter`と入力すれば`12.5V`を出力する。



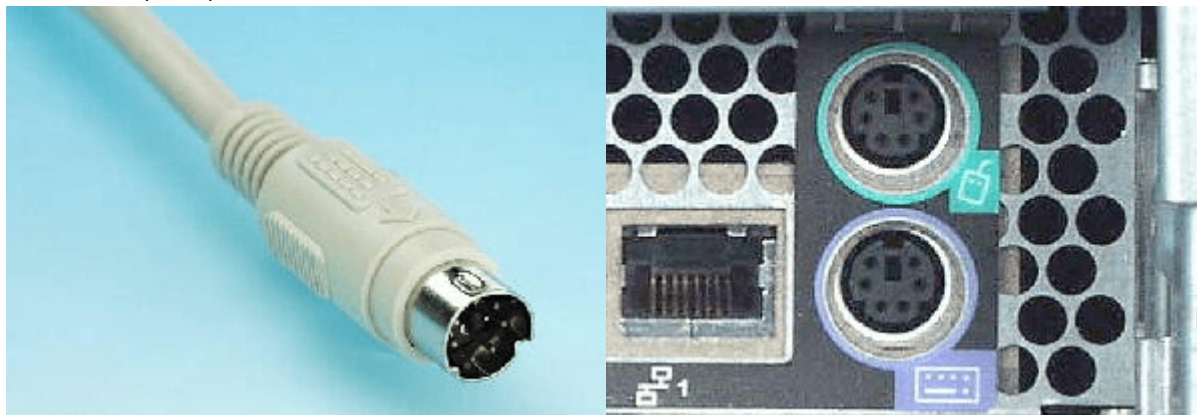
等の用途には打って付けだと思います。

動作原理

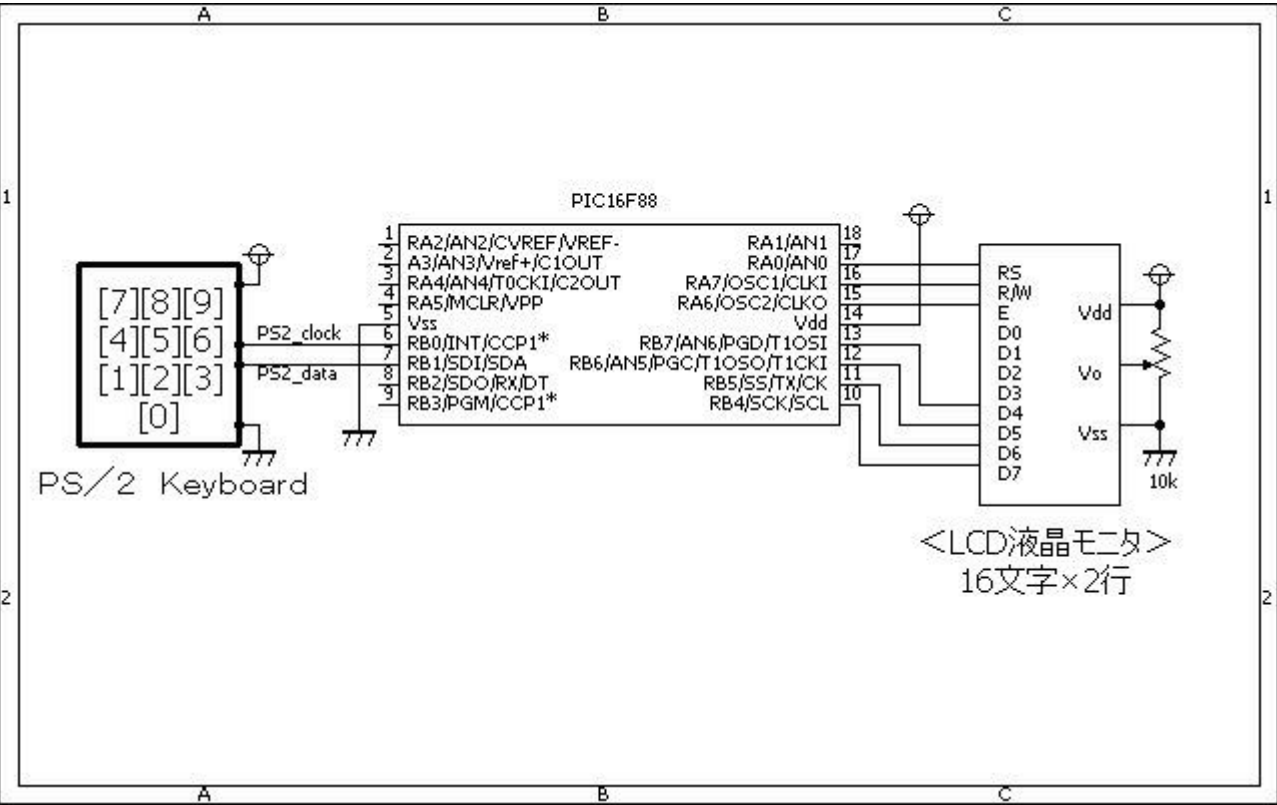
ハードウェアとしてはPICとPS/2テンキーとはCLOCK信号とDATA信号の2本を接続するだけです。ソフトウェアとしては、今回は一桁の数値の四則演算に限定しましたので、とても簡単です。

- PS/2テンキーからのデータを受信する。
- 受信したデータをLCDに表示させる。
- Delキーが押下されると、最後に押されたデータを削除する。
- Enterキーが押下されると、計算をして結果をLCDに表示させる。

<Mini-DIN(6Pin)ピン配置> 1:DATA 2: 3:GND 4:+5V 5:CLOCK 6:



回路図



ソースコード

Calculator.c

```
//*****
*
/*
『簡易電卓（四則演算）』
*/
//*****
*

short ShortToStrEx(short s, char* buf)
{
    static    char    d1, d2;
    if (s >= 0) {
        d2 = s / 10;
        d1 = s - (d2 * 10);
        buf[0] = d2 + '0';
        buf[1] = d1 + '0';
        buf[2] = 0x00;
        return(0);
    } else {
        s = 0 - s;
    }
}
```

```
        d2 = s / 10;
        d1 = s - (d2 * 10);
        buf[0] = d2 + '0';
        buf[1] = d1 + '0';
        buf[2] = 0x00;
        return(-1);
    }
}

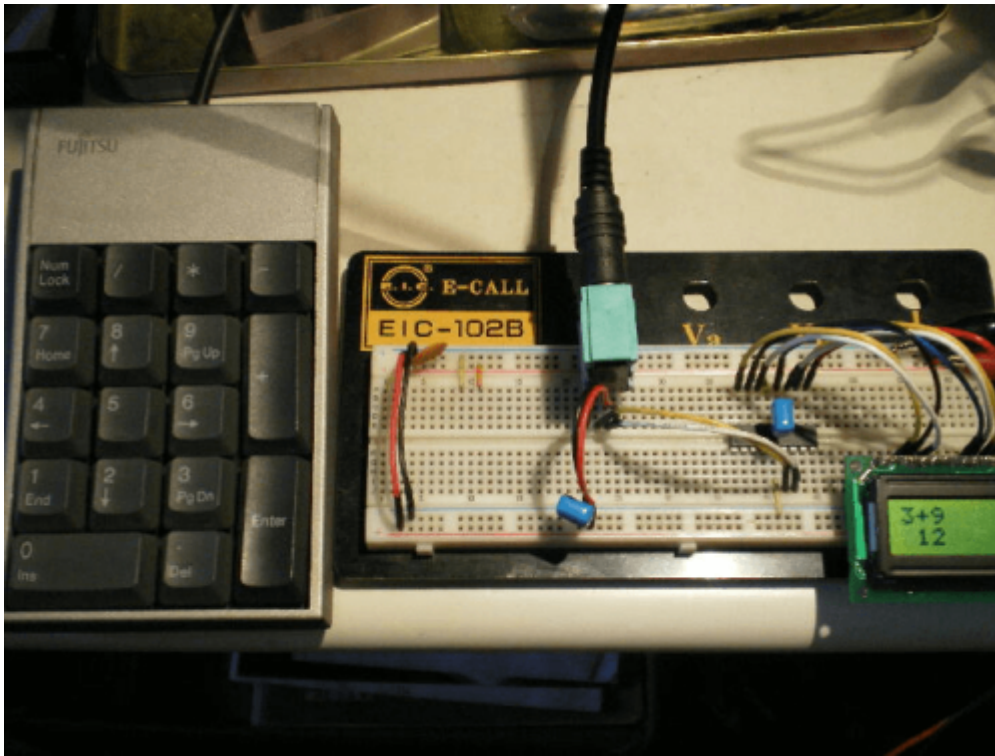
//*****
*

void main()
{
    unsigned    short    keydata, special, down;
    unsigned    char     buf[8], len;
    signed      short    dat1, dat2, kotae;
    //
    OSCCON = 0b01110000;          // クロックは8Mhz
    CMCON   = 0b00000111;          // コンパレータは使用しない。
    ANSEL   = 0b00000000;
    TRISA   = 0b00111110;
    TRISB   = 0b00001111;
    OPTION_REG.F7 = 0;
    // 初期化。
    Lcd_Custom_Config(&PORTB, 4, 5, 6, 7, &PORTA, 0, 7, 6);
    Lcd_Custom_Cmd(LCD_CURSOR_OFF);
    Lcd_Custom_Cmd(LCD_CLEAR);
    Lcd_Custom_Out(1, 1, "Calculator V1");
    Delay_ms(500);
    Lcd_Custom_Cmd(LCD_CLEAR);
    //
    Ps2_Config(&PORTB, 0, 1);
    Delay_ms(100);
    //
    len = 0;
    buf[0] = 0x00;
    //
    while (1) {
        if (Ps2_Key_Read(&keydata, &special, &down) == 1) {
            if (down && !special && keydata) {
                switch (keydata) {
                    case '#':
                        break;
                    default:
                        if (keydata == '.') {
                            len = (len > 0) ? (len - 1) : len;
                            buf[len] = 0x00;
                        } else {
                            buf[len] = keydata;
                            len++;
                        }
                    }
                }
            }
        }
    }
}
```

```
        buf[len] = 0x00;
    }
    Lcd_Custom_Cmd(LCD_CLEAR);
    Lcd_Custom_Out(1, 1, buf);
    break;
}
}
if (down && special && (keydata == 13)) {    // enter-key
    dat1 = buf[0] - '0';
    dat2 = buf[2] - '0';
    switch (buf[1]) {
    case '+':
        kotae = dat1 + dat2;
        break;
    case '-':
        kotae = dat1 - dat2;
        break;
    case '*':
        kotae = dat1 * dat2;
        break;
    case '/':
        kotae = dat1 / dat2;
        break;
    }
    if (ShortToStrEx(kotae, buf) == -1)
        Lcd_Custom_Out(2, 1, "-");
    else
        Lcd_Custom_Out(2, 1, " ");
    if (buf[0] == '0')
        Lcd_Custom_Out(2, 2, &buf[1]);
    else
        Lcd_Custom_Out(2, 2, buf);
    //
    len = 0;
}
}
}
}

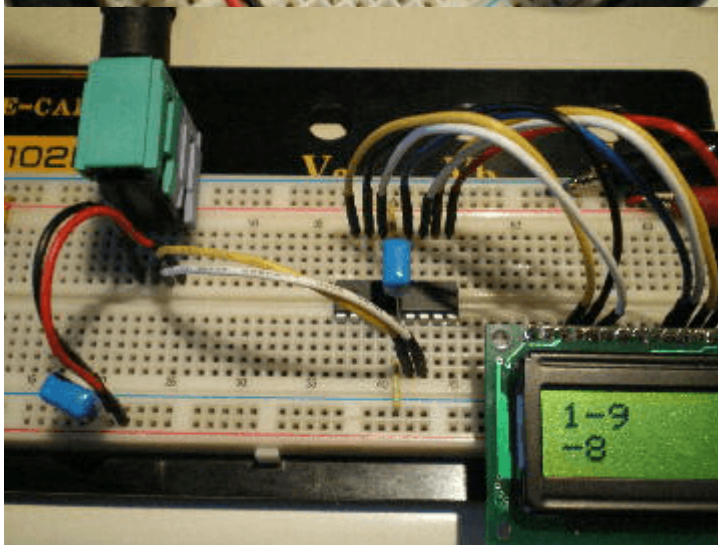
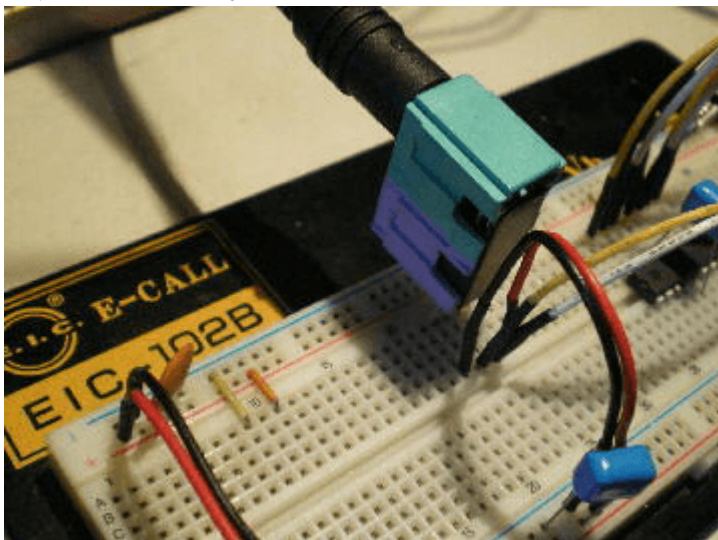
//*****
*
```

動作確認

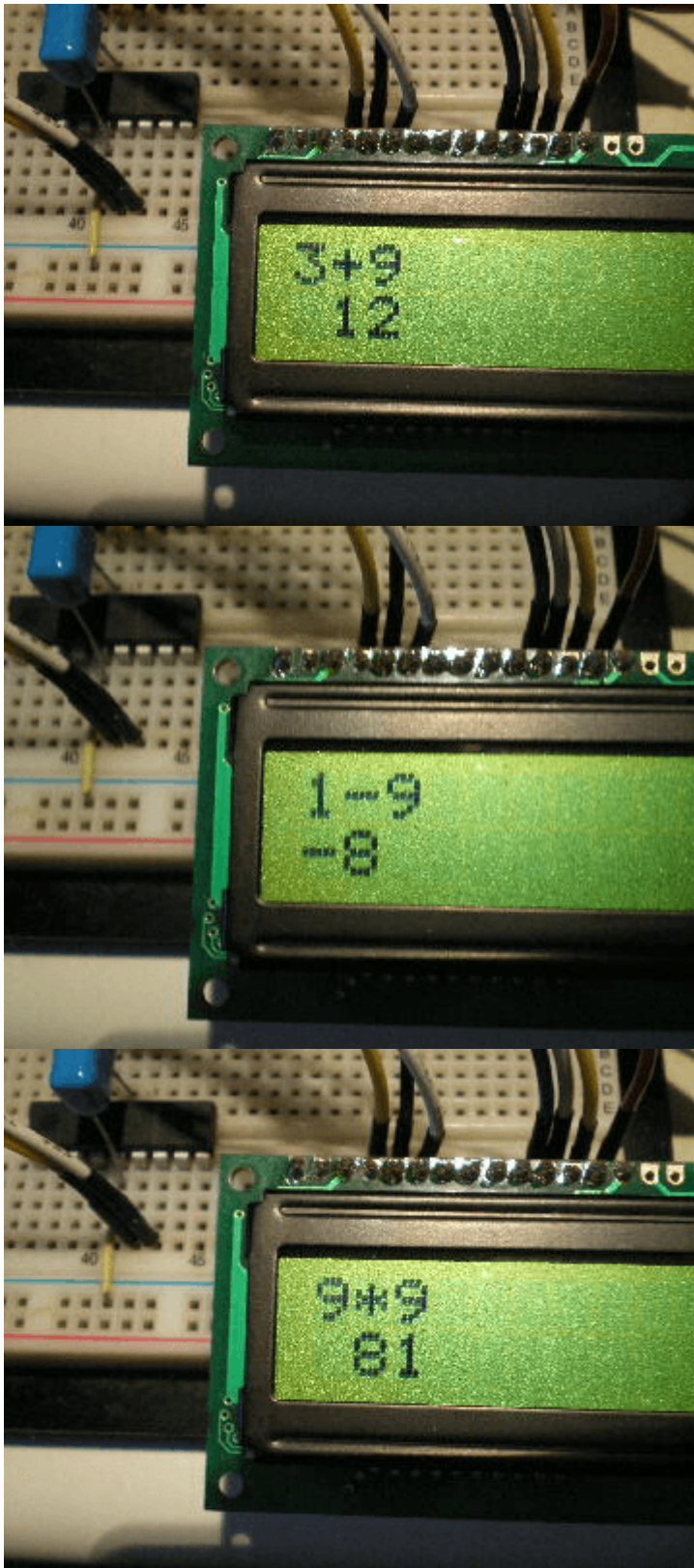


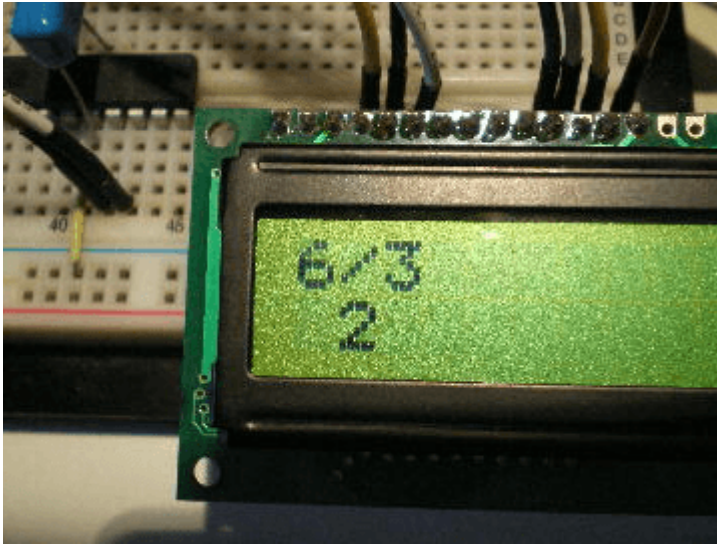
PS/2テンキーへの配線

は、4本だけです。 黒色:GND 赤色:+5V 黄色:CLOCK信号 白色:DATA信号



実際に四則演算をしました。





From:

<http://deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://deepsky.jp/wiki/doku.php?id=elechobby:picdic:pic16f88:78&rev=1588156371>

Last update: **2025/10/17 14:28**

