

8桁

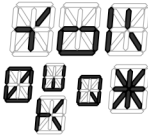


バックライトつき
セグメントLCD

☆ 数字、アルファベット、カタカナなどを表示できる8桁14セグメントのLCDです。セグメント表示以外に、ベルやアンテナマークなど特殊記号も表示できます。
☆ シンプルなシリアルインターフェイスで簡単に表示できます。(シリアルデータ、シフトクロック、ラッチイネーブルの3本で制御できます)

主な仕様

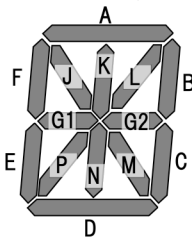
- ◎ 表示：14セグメント×8桁 + 特殊記号 (ベル、電池マーク、アンテナマークなど)
- ◎ インターフェイス：142ビット シリアル転送
- ◎ バックライトLED：4.3V 20mA
- ◎ (実測値、外部に電流制限用の抵抗が必要)
- ◎ 電源電圧：5V 単一電源 (3V動作不可)



オーディオ・マルチメディア電子パーツ 年中無休・営業時間：AM10:00~PM8:00
〒556-0055 大阪市東淀川区西中島4-7
TEL:06-6544-4555 / FAX:06-6544-1744
E:shop@vov.com / www.vov.com
Shop: http://vov.com / Twitter: @vov4555

電子工作向けの学習、実験、趣味向けであり、
必ずしも産業用途です。目視検査のみの品質や誤りがある場合があります。
商品の性能等を保証するものではありません。
各種設定や、使用についてはお各自責任でお願いいたします。
いかなる事故、損失においても製造者、流通者、販売者は一切の責任を負いかねます。返品、交換、保証等の対応はいたしません。

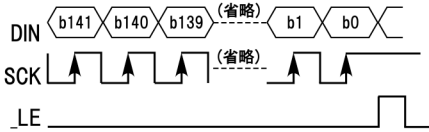
◎ ビット位置と表示セグメントの関係



下の表はデータのビット位置と表示されるセグメントの関係を示したものです。ビットを「1」にしたところが表示されます。

	7	6	5	4	3	2	1	0
0	2J	1K	1L	1M	1N	1P	1Q2	1J
1	3G2	3J	2K	2L	2M	2N	2P	2G2
2	4P	4G2	4J	3K	3L	3M	3N	3P
3	5N	5P	5G2	5J	4K	4L	4M	4N
4	6M	6N	6P	6G2	6J	5K	5L	5M
5	7L	7M	7N	7P	7G2	7J	6K	6L
6	8K	8L	8M	8N	8P	8G2	8J	7K
7	なし	なし	電池	アンテナ	■	ダイヤ	※	なし
8	なし	なし	なし	なし	なし	なし	なし	なし
9	なし	なし	なし	なし	なし	なし	なし	なし
10	2F	1A	1B	1C	1D	1E	1G1	1F
11	3G1	3F	2A	2B	2C	2D	2E	2G1
12	4E	4G1	4F	3A	3B	3C	3D	3E
13	5D	5E	5G1	5F	4A	4B	4C	4D
14	6C	6D	6E	6G1	6F	5A	5B	5C
15	7B	7C	7D	7E	7F	7F	6A	6B
16	8A	8B	8C	8D	8E	8G1	8F	7A
17	なし	なし	SPK	♪	なし	サイクル	ベル	>

◎ コントロールのしかた



- 表示データ(142ビット)は上位ビットから順に送ります(SCKの立上りエッジで取り込み)
- データを送ったあとLEに「H」パルスを与えるとデータが表示に反映されます。

◎ ピン配置

信号名	機能概要
1	VCC 電源入力 (5V DC)
2	_LE ラッチイネーブル (「L」でデータラッチ)
3	DIN シリアルデータ入力
4	SCK 転送クロック入力 (立上りエッジで取り込み)
5	GND グラウンド
6	NC ノンコネクション
7	LED-A1 バックライトLED アノード (L)
8	LED-A2 バックライトLED アノード (R)
9	LED-K バックライトLED 共通カソード
10	NC ノンコネクション

◎ (参考)表示用サブルーチン例

下記のサブルーチンは表示用バッファに置いたセグメントデータを送って表示させるサブルーチンです。(WinAVRで作成、ATtiny2313A用)

```
// 表示用バッファ (142ビット分)
unsigned char disp_buf[142];

void LCD_disp(void)
{
    volatile int i, j, k;
    delay_us(1);
    for(i=142; i>=0; i--)
    {
        PORTB = PORTB & (unsigned char)~(1<<SCK); // SCK = L
        j = i / 8;
        k = i % 8;
        if((disp_buf[j] & (1<<k)) != 0)
        {
            PORTA = PORTA | (1<<DIN);
        }
        else
        {
            PORTA = PORTA & (unsigned char)~(1<<DIN);
        }
        delay_us(1);
        PORTB = PORTB | (1<<SCK); // SCK = H
        delay_us(1);
    }
    delay_us(1);
    PORTA = PORTA | (1<<_LE); // _LE = H
    delay_us(1);
    PORTA = PORTA & (unsigned char)~(1<<_LE); // _LE = L
}
}
```