

簡易メロディボックス(明るさ検知)

12曲も収められたメロディオルゴールIC[UM3482]が手に入りましたので、早速簡単なメロディボックスを製作してみました。

単純に演奏させるだけなら[UM3482]に少しの部品(抵抗、コンデンサ、スピーカなど)を付ければ済みます。しかし、それだけでは面白くありませんので[PICとCDS(Cadmium Sulfide)を使って、周りの明るさの変化によって曲を鳴らしてみました。

<仕様>

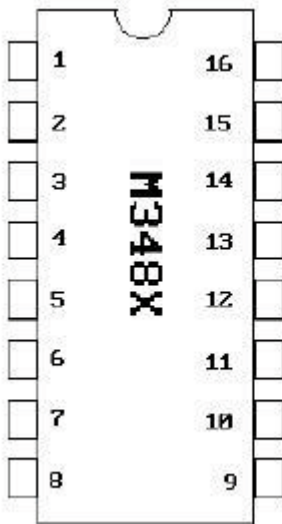
- 起動直後の周りの明るさを閾値とします。
- 閾値より明るくなるか暗くなるかによってメロディを1曲鳴らせます。
- メロディ(曲)は、12種類を順次切り替えていきます。
- 乾電池2本で動作可能とします。

M3482 (12 songs)

American Patrol
Rabbits
Oh My Darling, Clementine
Butterfly
London Bridge Is Falling Down
Row, Row, Row Your Boat
Are You Sleeping
Happy Birthday
Joy Symphony
Home Sweet Home
Weigenlied
Melody On Purple Bamboo

<UM3482の曲目>

<UM3482のピン配置>



Pin no.	Symbol	Description
1	TSP	Output flag signal of melody auto stop : In normal operation, this should be open.
2	CE	Chip is enabled if connected to Vdd. Chip is disabled if connected to Vss.
3	LP	Only one song plays if connected to Vdd. All songs play if connected to Vss.
4	SL	A positive going edge will change to play the next song.
5	AS	The melody will repeat if connected to Vdd and will stop automatically if to Vss.
6	NC	No connection
7	ENV	Envelope circuit terminal
8	Vss	Negative power supply
9	MTO	Modulated tone signal output.
10	OP1	Pre-amplifier output 1
11	OP2	Pre-amplifier output 2
12	MT1	Modulated tone signal input to the pre-amplifier.
13	NC	No connection
14	OSC2	External oscillator terminal 1
15	OSC1	External oscillator terminal 2
16	Vdd	Positive power supply

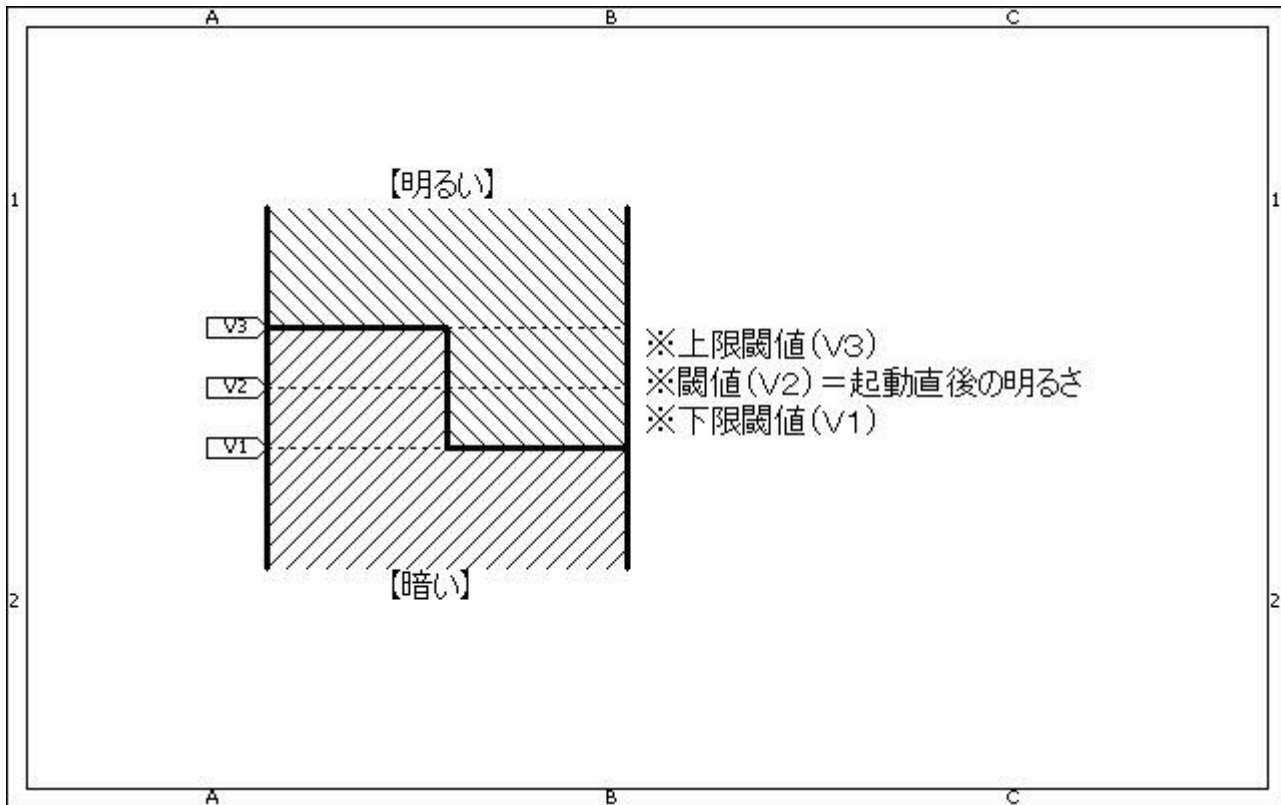
動作原理

<明るさの電圧変換>

- CDSの抵抗値は、暗いと高くなり、明るいと低くなります。
- この特性を利用して、明るさ(照度)を電圧に変換します。
- 回路図のようにCDSと抵抗を接続すると、出力電圧は“明るいと高く”なり、“暗いと低く”なります。

<明るさの判断>

- 起動直後の明るさを測定し、閾値(V2)とします。
- 閾値(V2)にはヒステリシス(hysteresis)特性を持たせます。
- 上限閾値(V3)以上になると“明るい”と判断します。
- 一旦“明るい”と判断したら、下限閾値(V1)以上であれば“明るい”と判断します。
- 下限閾値(V1)以下になると“暗い”と判断します。
- 一旦“暗い”と判断したら、上限閾値(V3)以下であれば“暗い”と判断します。



<UM3482の制御>

- 動作モード(7)を使用します。
- SL信号を、“ \square ”(lowからhigh)にすると、次の曲が1度だけ演奏され停止します。
- 明るさが、変化(“明るい 暗い”、“暗い 明るい”)した時のみ \square SL信号を、“ ” にします。

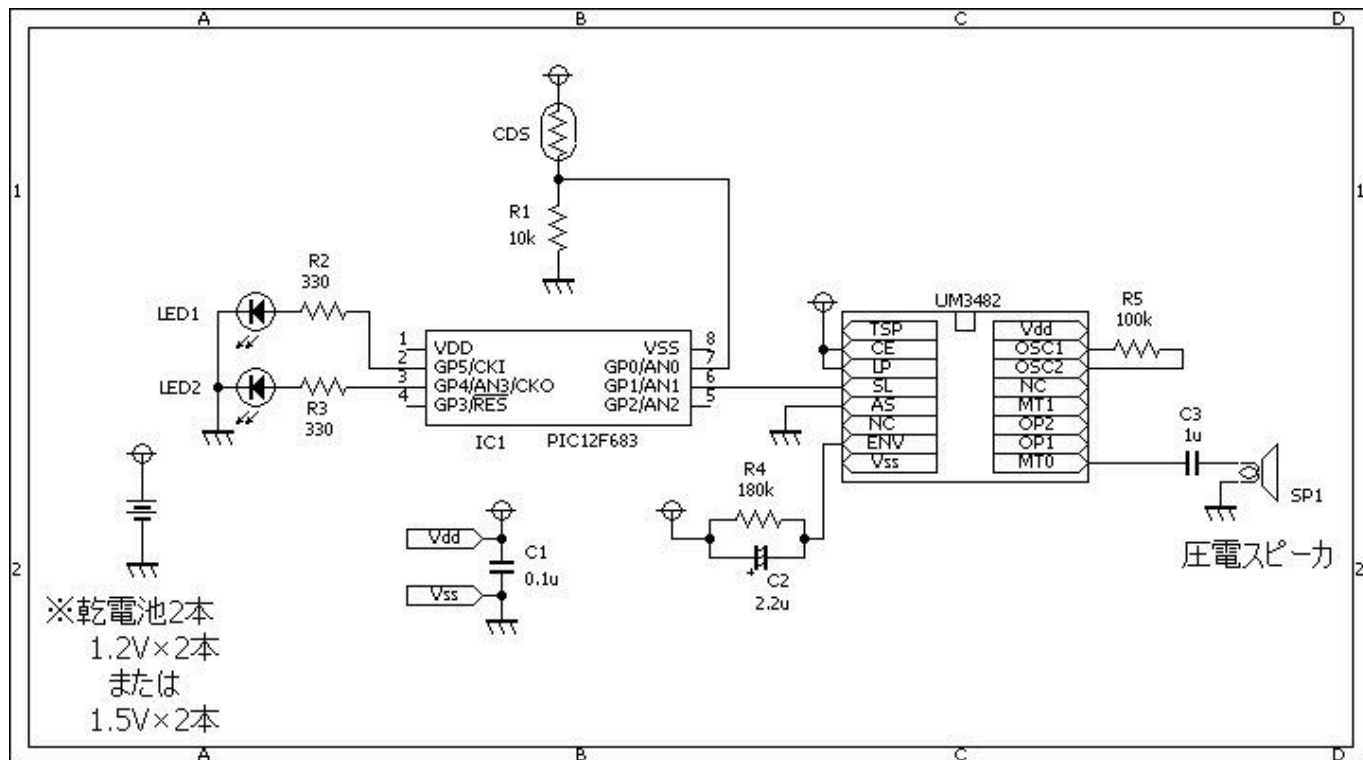
UM3482の動作モード

Playing modes

Mode	CE	SL	LP	AS	Program
0	0	X	X	X	Standby
1	1	0	0	0	First melody→...→Last melody→Stop
2	\uparrow	0	0	1	First melody→...→Last melody→Repeat from first melody
3	\uparrow	0	1	0	Start from the present melody→Stop
4	1	0	1	1	Repeat the present melody
5	1	\uparrow	0	0	Change to the next melody→...→Last melody→Stop
6	1	\uparrow	0	1	Next melody→...→Last melody→Repeat from first melody
7	1	\uparrow	1	0	Change to the next melody→Stop
8	1	\uparrow	1	1	Change to the next melody→Repeat the same melody

(\uparrow means a low to high voltage level transaction)

回路図



ソースコード

[melody_box_v1.c](#)

```
//*****
*
*/
< 簡易メロディボックス (12曲) >
*/
//*****
*

#define UM3482      GPIO.F1
#define LED1        GPIO.F4
#define LED2        GPIO.F5

#define ON          1
#define OFF         0

//*****
*

int measurement()
{
    int      ad;
    short    cnt;
    //
    ad = OFF;
}
```

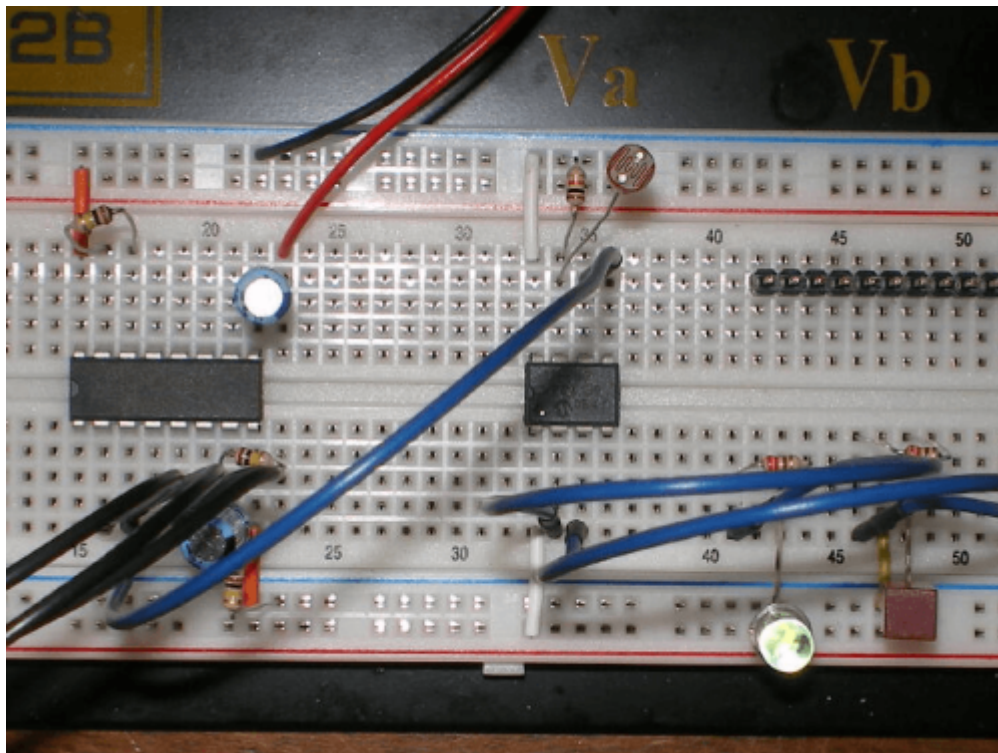
```
    for (cnt = OFF; cnt < 10; cnt++) {
        ad += Adc_Read(0);
    }
    return (ad / 10);
}

//*****
*

void    main()
{
    int    threshold;
    short  flag, cnt;
    //
    OSCCON = 0b00000000;    //32kHz
    CMCON0 = 0b00000111;
    ANSEL  = 0b00000001;
    TRISIO = 0b00001001;
    //
    flag = OFF;
    UM3482 = OFF;
    LED1 = OFF;
    LED2 = OFF;
    //起動直後の明るさを基準値として取り込む。
    Delay_ms(1000);
    threshold = measurement();
    for (cnt = OFF; cnt < 10; cnt++) {
        LED1 = ON;
        LED2 = ON;
        Delay_ms(100);
        LED1 = OFF;
        LED2 = OFF;
        Delay_ms(100);
    }
    //
    while (flag == 0) {
        //基準値に比べて明るくなったかをチェックする。
        if (measurement() > (threshold + 50)) {
            UM3482 = ON;
            Delay_ms(100);
            UM3482 = OFF;
            flag = 1;
        }
        //基準値に比べて暗くなったかをチェックする。
        if (measurement() < (threshold - 50)) {
            UM3482 = ON;
            Delay_ms(100);
            UM3482 = OFF;
            flag = 2;
        }
    }
    //
}
```

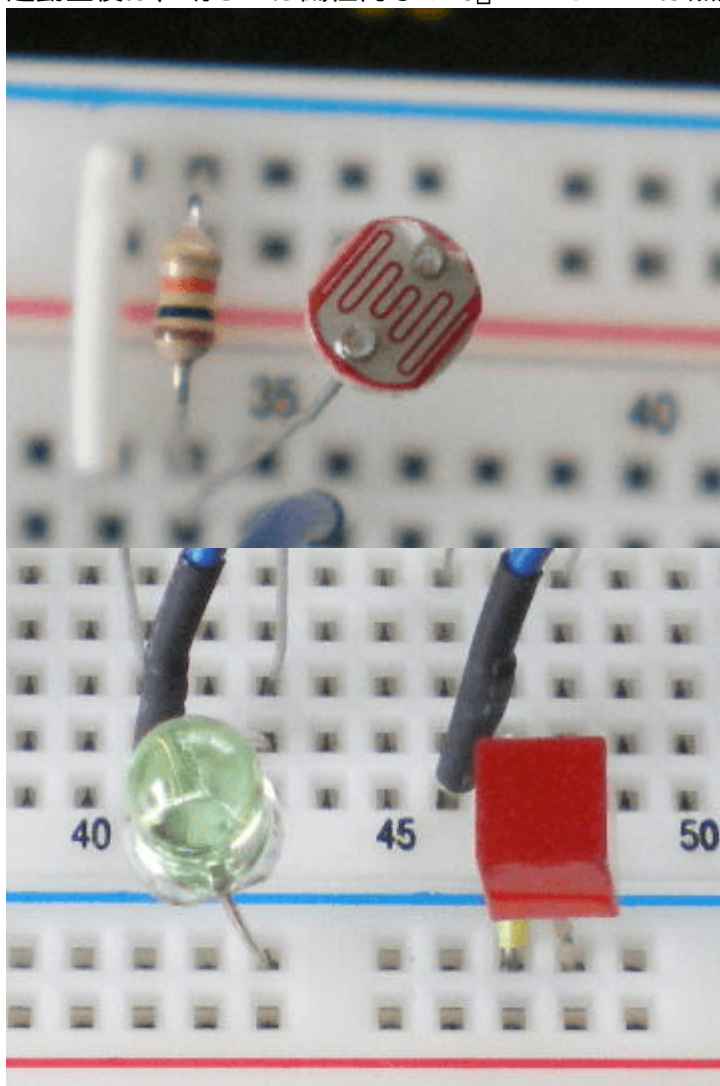
```
        LED1 = ON;
        LED2 = ON;
        Delay_ms(100);
        LED1 = OFF;
        LED2 = OFF;
        Delay_ms(100);
    }
    //
    while (1) {
        //基準値に比べて明るくなったかをチェックする。
        if ((measurement() > (threshold + 50)) && (flag == 2))
        {
            UM3482 = ON;
            Delay_ms(100);
            UM3482 = OFF;
            flag = 1;
        }
        //基準値に比べて暗くなったかをチェックする。
        if ((measurement() < (threshold - 50)) && (flag == 1))
        {
            UM3482 = ON;
            Delay_ms(100);
            UM3482 = OFF;
            flag = 2;
        }
    }
    //
    if (flag == 1) {
        LED1 = ON;
        Delay_ms(100);
        LED1 = OFF;
        Delay_ms(100);
    } else {
        LED2 = ON;
        Delay_ms(100);
        LED2 = OFF;
        Delay_ms(100);
    }
}
//*****
*
```

動作確認

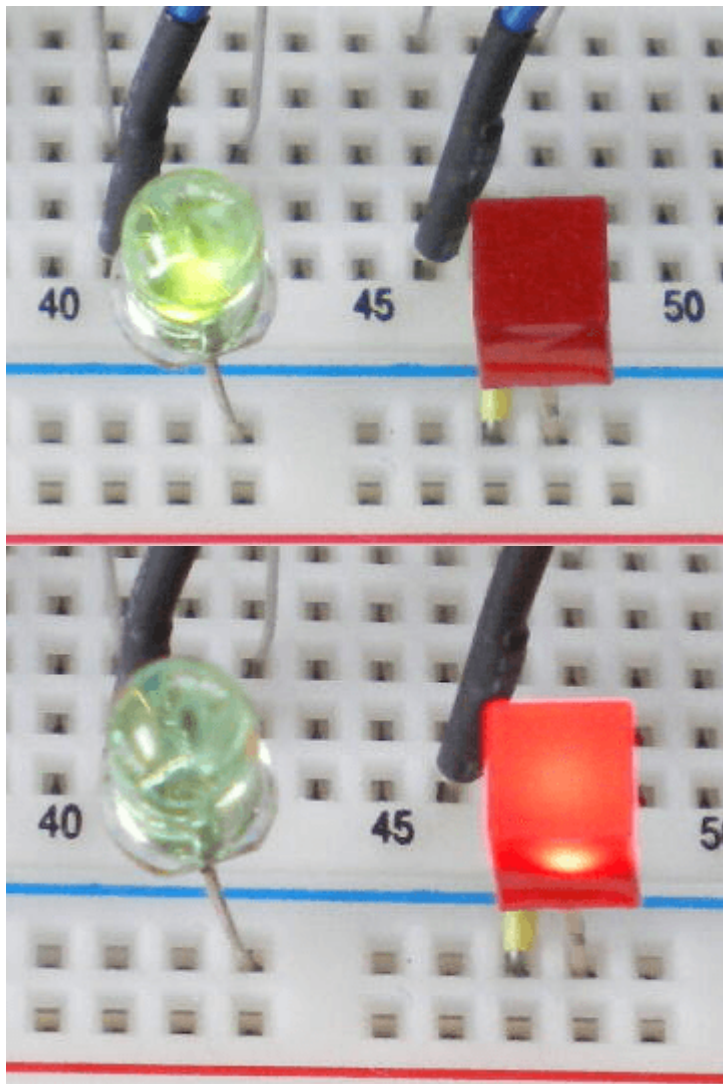


左側:CDS部分です。 右側:

起動直後は、明るさは閾値内なのでLED1とLED2は点滅します。



左側:明るくなるとLED1が点滅します。 右側:暗くなるとLED2が点滅します。



如何ですか? 用途としては、部屋の照度が明るくなり過ぎたり、暗くなり過ぎた時に、オルゴールがなるので、次のような効果が期待できます。

- 明るくなり過ぎが分かるので、電気代を節約することができます。
- 暗くなり過ぎが分かるので、視力低下を防ぐことができます。

From:
<http://www.deepsky.jp/wiki/> - うごくといいな

Permanent link:
<http://www.deepsky.jp/wiki/doku.php?id=elechobby:picdic:pic12f683:32&rev=1588131465>

Last update: 2025/10/17 14:27

