

簡易汎用タイマー

概要

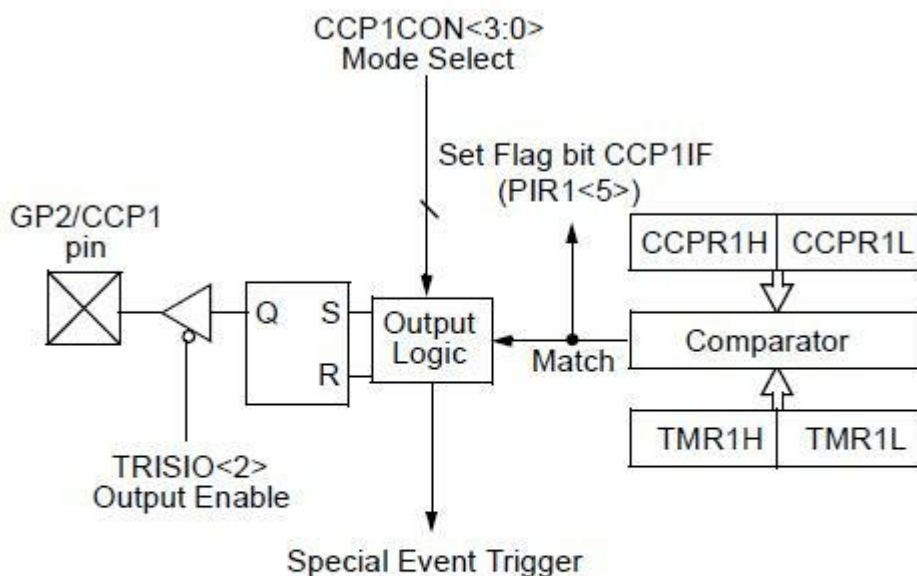
電灯を一定時間だけ点灯させたり、半田ごてを一定時間が経過するとOFFさせたりする、簡易で汎用的なタイマーを製作してみました。

<仕様>

- タイマーのカウント時間は、4種類(10秒、10分、100分、10時間)から選択可能とする。
- タイマーの残り時間は、10個のLEDで表示する□(LEDを順次消灯していく)
- タイムアウトしたことを、ブザーで知らせる□(1kHz(50msec)×5回)
- タイマーのカウント中は、LEDを点灯させる。
この信号でSSR(ソリッドステートリレー)等を駆動することにより□AC100VをON/OFF可能です。
- タイマーのカウントを途中で停止可能とする。

動作原理

<基準時間(0.1秒)> PIC16F88が内蔵している□CCPモジュールをコンペアモードで使用し、0.1秒の割り



Special Event Trigger will:

- Clear TMR1H and TMR1L registers
- NOT set interrupt flag bit TMR1F (PIR1<0>)
- Set the GO/DONE bit (ADCON0<1>)

込みを発生させます。

<4種類のタイマー>

- 各タイマーは、0.1秒の割り込みの回数によって実現します。

|<200px>|

提供するタイマー	LED1個あたりの時間量	割り込みの回数
10秒タイマー	1秒	100回
10分タイマー	1分	6000回
100分タイマー	10分	60000回
10時間タイマー	1時間	360000回

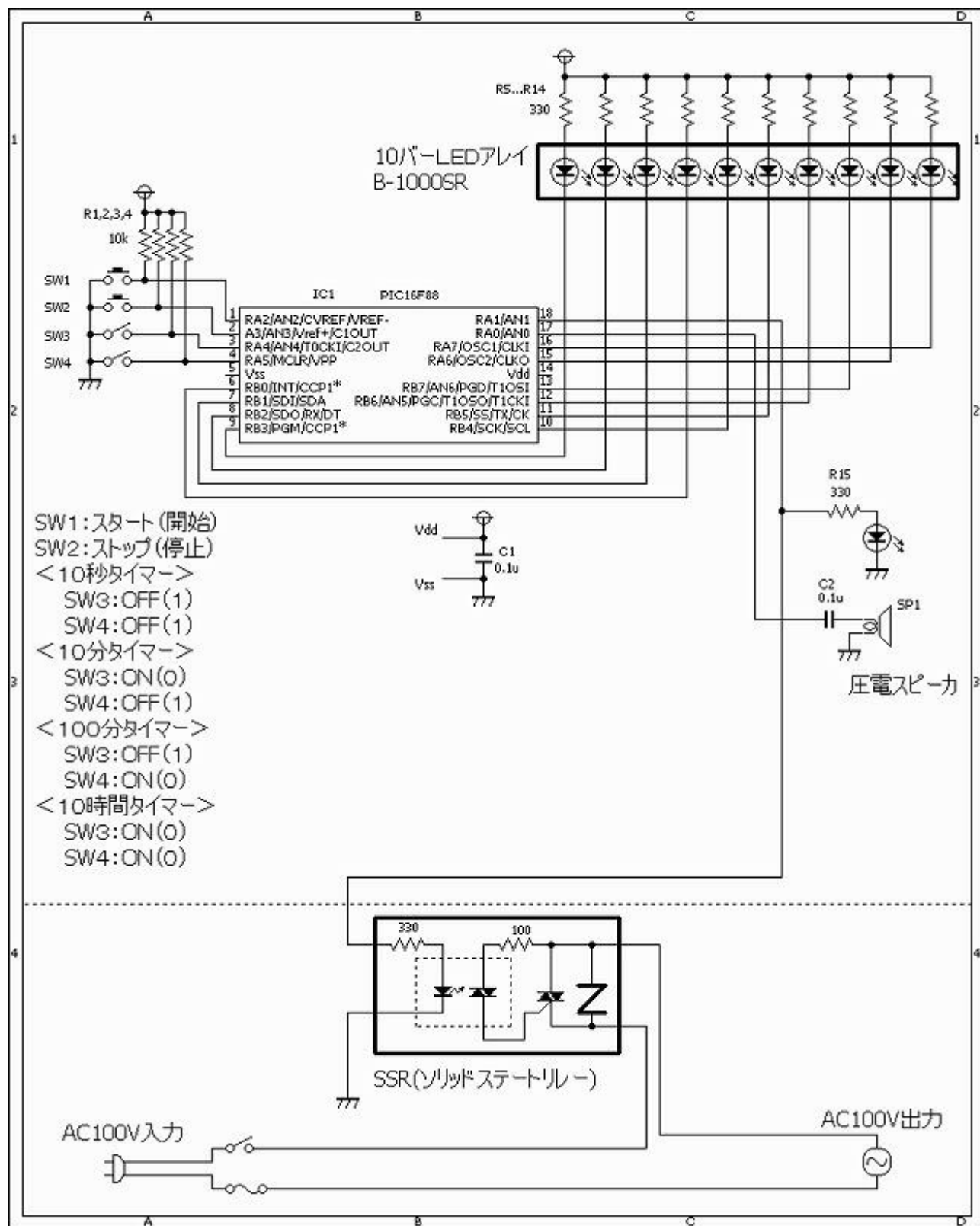
<LEDの点灯と消灯>

- 点灯 タイマーが開始されると10個のLED全てを点灯させます。
- 消灯 時間が経過していくと、順次LEDを消灯していきます。

<ブザー音> mikroCが提供する、サウンドライブラリを使用します。

- 初期化□Sound_Init関数 使用するポートとピン番号を指定します。
- ブザー音 Sound_Play関数 発振周波数と長さを指定します。

回路図



ソースコード

timer_v1.c

```
//*****
*
/*
```

```
<簡易タイマー>

*/
//*****
*

#define      LED0      PORTB.F0
#define      LED1      PORTB.F1
#define      LED2      PORTB.F2
#define      LED3      PORTB.F3
#define      LED4      PORTB.F4
#define      LED5      PORTB.F5
#define      LED6      PORTB.F6
#define      LED7      PORTB.F7
#define      LED8      PORTA.F6
#define      LED9      PORTA.F7

#define      LED        PORTA.F1

#define      ON          0
#define      OFF         1

#define      START_SW    PORTA.F2
#define      STOP_SW     PORTA.F3

#define      MODE1_SW    PORTA.F4
#define      MODE2_SW    PORTA.F5

//*****
*

static long    cnt_time;

void    interrupt()
{
    if (PIR1.CCP1IF == 1) {
        PIR1.CCP1IF = 0;
        //
        if (cnt_time > 0)
            cnt_time--;
    }
}

//*****
*

void    Pwm_Change_DutyEx(unsigned int duty_ratio)
{
    CCPR1L = duty_ratio >> 2;
    CCP1CON.CCP1Y = (duty_ratio & 0b0000000000000001) == 0 ? 0 : 1;
    CCP1CON.CCP1X = (duty_ratio & 0b0000000000000010) == 0 ? 0 : 1;
}
```

```
//*****
*

void led_all_on()
{
    LED0 = ON;
    LED1 = ON;
    LED2 = ON;
    LED3 = ON;
    LED4 = ON;
    LED5 = ON;
    LED6 = ON;
    LED7 = ON;
    LED8 = ON;
    LED9 = ON;
}

void led_all_off()
{
    LED0 = OFF;
    LED1 = OFF;
    LED2 = OFF;
    LED3 = OFF;
    LED4 = OFF;
    LED5 = OFF;
    LED6 = OFF;
    LED7 = OFF;
    LED8 = OFF;
    LED9 = OFF;
}

void main()
{
    static short cnt, cnt_flg;
    static int cnt_offset;
    //
    TRISA = 0b00111100;
    TRISB = 0b00000000;
    OSCCON = 0b01110000; // クロックを4Mhzに設定する。
    ANSEL = 0b00000000; // □□□変換を使用しない。
    // TIMER1の設定
    PIE1.TMR1IE = 0;
    PIR1.TMR1IF = 0;
    T1CON.T1CKPS0 = 1;
    T1CON.T1CKPS1 = 1;
    T1CON.TMR1ON = 0;
    TMR1L = 0;
    TMR1H = 0;
    // CCPの設定
    PIE1.CCP1IE = 1;
```

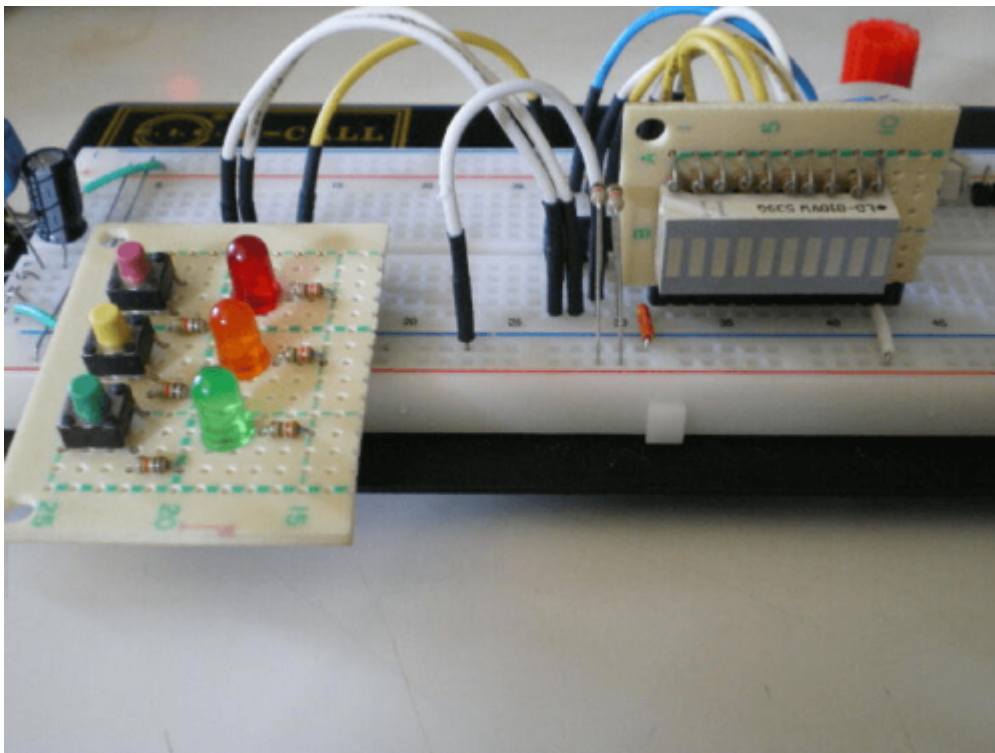
```
PIR1.CCP1IF = 0;
CCP1CON.CCP1M3 = 1;
CCP1CON.CCP1M2 = 0;
CCP1CON.CCP1M1 = 1;
CCP1CON.CCP1M0 = 1;
CCPR1L = 0xA8;    // 0.1sec...(1÷8000000)*4*8*25000
CCPR1H = 0x61;    //
//
Sound_Init(&PORTA, 0);
//
LED = 0;
led_all_off();
// 割り込みを許可する。
INTCON.PEIE = 1;
INTCON.GIE = 1;
//
T1CON.TMR1ON = 1;
//
LED0 = ON;
Delay_ms(100);
LED0 = OFF;
LED1 = ON;
Delay_ms(100);
LED1 = OFF;
LED2 = ON;
Delay_ms(100);
LED2 = OFF;
LED3 = ON;
Delay_ms(100);
LED3 = OFF;
LED4 = ON;
Delay_ms(100);
LED4 = OFF;
LED5 = ON;
Delay_ms(100);
LED5 = OFF;
LED6 = ON;
Delay_ms(100);
LED6 = OFF;
LED7 = ON;
Delay_ms(100);
LED7 = OFF;
LED8 = ON;
Delay_ms(100);
LED8 = OFF;
LED9 = ON;
Delay_ms(100);
LED9 = OFF;
//
for (cnt = 0; cnt < 5; cnt++) {
    led_all_on();
```

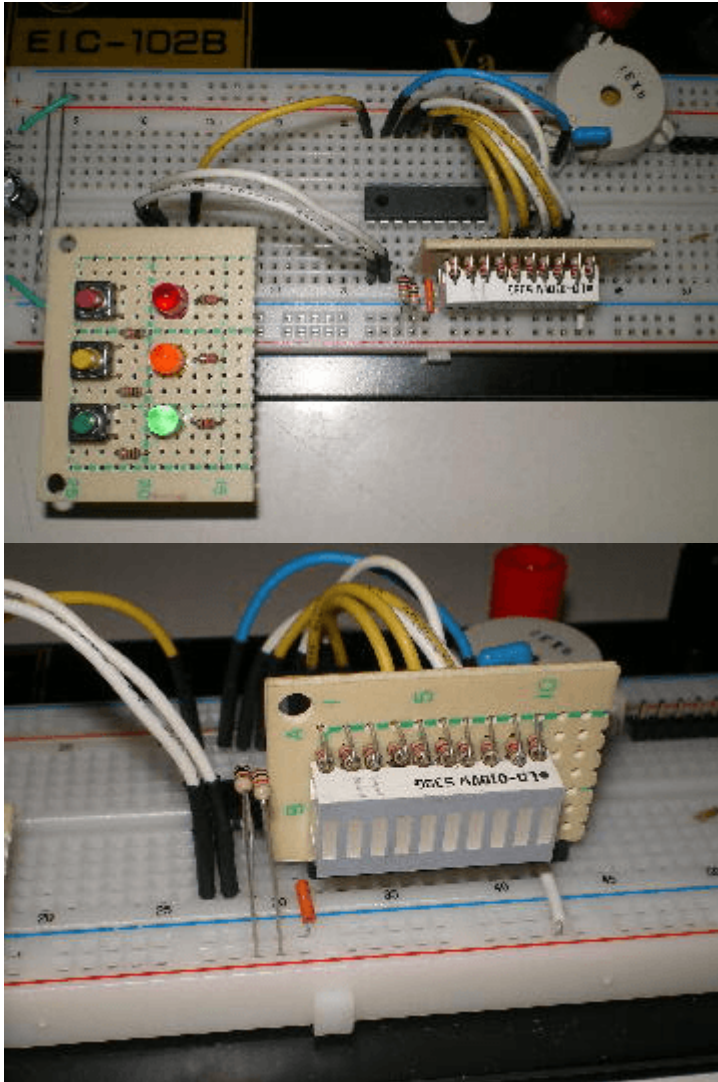
```
    Sound_Play(1000, 50);
    led_all_off();
    Delay_ms(50);
}
//
while (1) {
    //スタート処理
    if (START_SW == 0) {
        while (Button(&PORTA, 2, 1, 1) == 0)
            Delay_ms(10);
        //
        if ((MODE1_SW == 1) && (MODE2_SW == 1)) {
            cnt_time = 100;           //10秒タイマー
            cnt_offset = 10;
        }
        if ((MODE1_SW == 0) && (MODE2_SW == 1)) {
            cnt_time = 6000;         //10分タイマー
            cnt_offset = 600;
        }
        if ((MODE1_SW == 1) && (MODE2_SW == 0)) {
            cnt_time = 60000;        //100分タイマー
            cnt_offset = 6000;
        }
        if ((MODE1_SW == 0) && (MODE2_SW == 0)) {
            cnt_time = 360000;       //10時間タイマー
            cnt_offset = 36000;
        }
        cnt_flg = 9;
        led_all_on();
        LED = 1;
    }
    //ストップ処理
    if (STOP_SW == 0) {
        while (Button(&PORTA, 3, 1, 1) == 0)
            Delay_ms(10);
        //
        cnt_time = 0;
        led_all_off();
        LED = 0;
    }
    //カウントダウン処理
    switch (cnt_flg) {
    case 9:
        if (cnt_time == (cnt_flg * cnt_offset)) {
            LED9 = OFF;
            cnt_flg--;
        }
        break;
    case 8:
        if (cnt_time == (cnt_flg * cnt_offset)) {
            LED8 = OFF;
```

```
        cnt_flg--;  
    }  
    break;  
case 7:  
    if (cnt_time == (cnt_flg * cnt_offset)) {  
        LED7 = OFF;  
        cnt_flg--;  
    }  
    break;  
case 6:  
    if (cnt_time == (cnt_flg * cnt_offset)) {  
        LED6 = OFF;  
        cnt_flg--;  
    }  
    break;  
case 5:  
    if (cnt_time == (cnt_flg * cnt_offset)) {  
        LED5 = OFF;  
        cnt_flg--;  
    }  
    break;  
case 4:  
    if (cnt_time == (cnt_flg * cnt_offset)) {  
        LED4 = OFF;  
        cnt_flg--;  
    }  
    break;  
case 3:  
    if (cnt_time == (cnt_flg * cnt_offset)) {  
        LED3 = OFF;  
        cnt_flg--;  
    }  
    break;  
case 2:  
    if (cnt_time == (cnt_flg * cnt_offset)) {  
        LED2 = OFF;  
        cnt_flg--;  
    }  
    break;  
case 1:  
    if (cnt_time == (cnt_flg * cnt_offset)) {  
        LED1 = OFF;  
        cnt_flg--;  
    }  
    break;  
case 0:  
    if (cnt_time == (cnt_flg * cnt_offset)) {  
        LED0 = OFF;  
        cnt_flg--;  
        //タイムアウト処理  
        LED = 0;  
    }
```

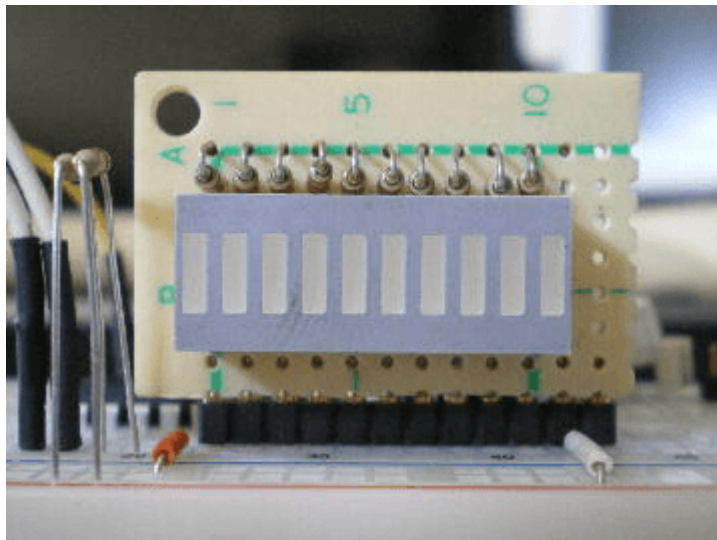
```
        for (cnt = 0; cnt < 10; cnt++) {  
            led_all_on();  
            Sound_Play(1000, 50);  
            led_all_off();  
            Delay_ms(50);  
        }  
    }  
    break;  
}  
}  
}  
}  
}  
  
//*****  
*
```

動作確認

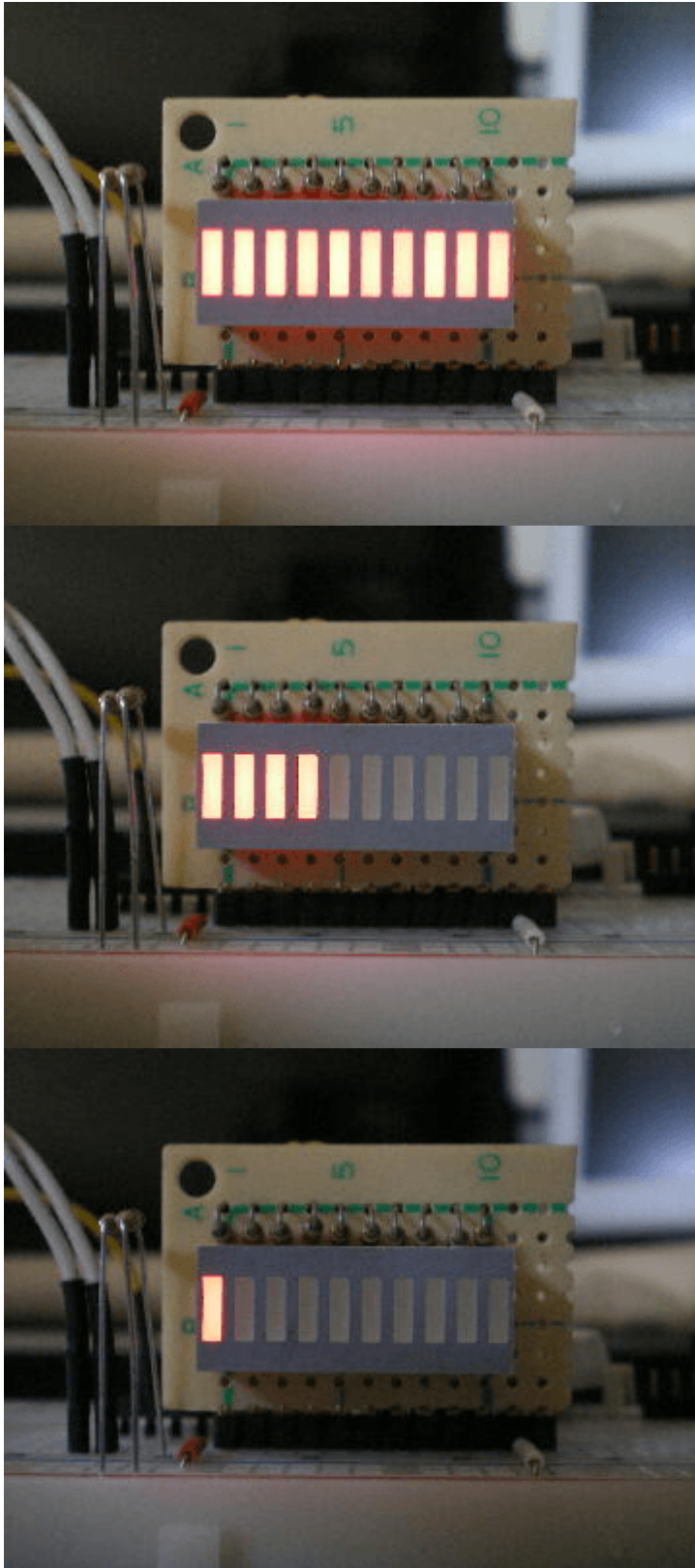




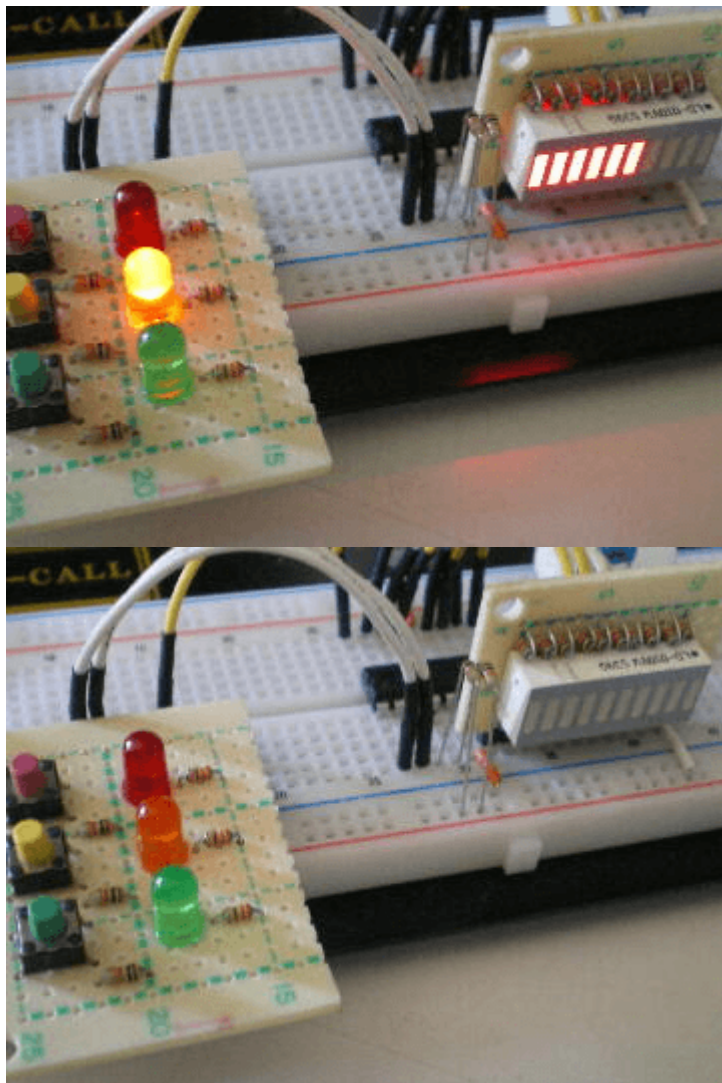
スタート前からスタート後のLEDの点灯の流れです。スタート直後は、全LEDが点灯し、1個ずつLEDが



消灯していきます。



左側:タイマーのカウント中です 右側:タイムアウトしたところです。



すよ 😊

如何ですか? 1台手元にあると、何かと重宝しま

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:pic16f88:126>

Last update: **2020/05/01 18:48**

