

# 方向探知機能付き“飛猫ブザー”機体発見器

2016.8.8



## 仕様:

|        |                        |       |              |
|--------|------------------------|-------|--------------|
| 発報周波数  | 2.9                    | [kHz] |              |
| 電源     | 2.3 ~ 5.5              | [V]   | (1cell Lipo) |
| 重さ     | 1.3                    | [g]   | (本体のみ)       |
|        | 1.6                    | [g]   | (リード付)       |
| 寸法     | 9Φ × 14.7              | [mm]  | (直径 × 長さ)    |
| ブザー    | “飛猫ブザー”(9Φ × 5 [mm])   |       |              |
| コントローラ | PIC12F1822             |       |              |
|        | 方向探知用電波 : 1MHz , DSM生成 |       |              |

用途 : 50gクラスの超小型ラジコン機の機体発見等

発報時期とパターン:

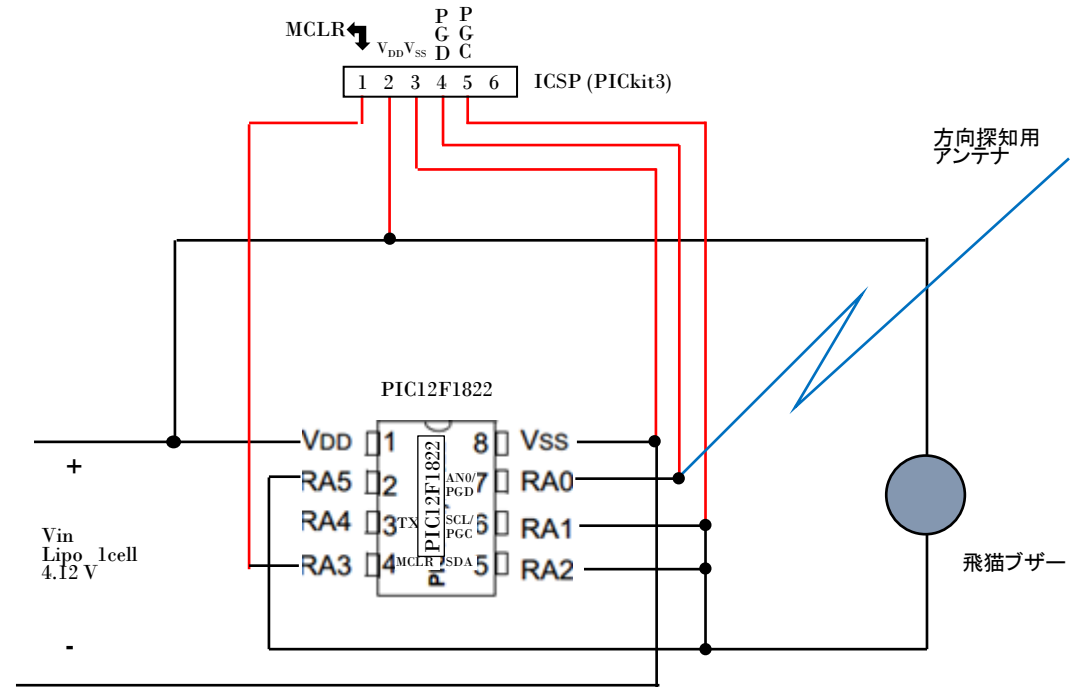
時期[秒]    パターン    発報時間[秒]

| 時期[秒] | パターン   | 発報時間[秒]       |
|-------|--------|---------------|
| 0     | P1     | 3             |
| 35    | P1, P2 | 9             |
| 69    | P1, P2 | 9             |
| 98    | P1, P2 | 9             |
| 122   | P1, P2 | 9             |
| 141   | P1, P2 | 9             |
| 157   | P1, P2 | 9             |
| 180   | P3     | ~ バッテリーが落ちるまで |

|    |                                                |
|----|------------------------------------------------|
| P1 | ピッピピッピッ_ピッピッ                                   |
| P2 | “タスケテ”のモールス符号                                  |
| P3 | “SOS”, “SOS”, “タスケテ”のモールス符号,<br>クレッシエンド, を繰り返す |

=====  
発報パターンは1年以上のテストで得た最適な形です.

# 回路



ブザーON/OFFは RA1, A2, A5 電流シンク駆動

# プログラム(HITECH xc8 コンパイラ)

```
/******  
* Test DSM(Data Signal Modulation)      2016.8.7 mtsw  
* Alarm & Radio Direction finding by PIC12F1822 and Sekiguchi Speaker  
***** */  
/* ** IO PIN config **  
2:RA5/RX :      :Buzzer (sink)  
3:RA4/TX/CK :NC  
4:MCLR/RA3 :NC  
5:RA2/(TX)SS :  :Buzzer (sink)  
6:RA1/(RX)/SCL/SCK :Buzzer (sink)  
7:RA0/SDA      :MODOUT ASK transmitter  
***** */  
#include <xc.h>  
//#define _XTAL_FREQ 31250  
#pragma config PLLEN = OFF , FOSC = INTOSC  
#define _XTAL_FREQ 2000000  
  
__CONFIG(CLKOUTEN_OFF & FOSC_INTOSC & FCMEN_OFF & IESO_OFF & BOREN_ON &  
        PWRTEN_ON & WDTE_OFF & MCLRE_OFF & CP_OFF & CPD_OFF);  
  
void Pi(){  
    LATA = 0x00;  
    __delay_ms(40);  
    LATA = 0xff;  
    __delay_ms(40);  
}  
void ToToTo(){  
    Pi();Pi();Pi();  
}  
void Tu(){  
    LATA = 0x00;  
    __delay_ms(100);  
    LATA = 0xff;  
    __delay_ms(50);  
}  
void delay(int times){  
    times /= 100;  
    while(times-- > 0){  
        CLKRCON = 0b11110101 ; // 50% 500 kHz  
        __delay_ms(100);  
        CLKRCON = 0b11110100 ; // 50% 1 MHz  
    }  
}  
void TanTaTa(){  
    Pi();  
    __delay_ms(100);  
    Pi();  
    Pi();  
    Pi();  
    __delay_ms(100);  
    Pi();  
    CLKRCON = 0b11110101 ; // 50% 500 kHz  
    __delay_ms(400);  
    Pi();  
    __delay_ms(100);  
    Pi();  
    __delay_ms(100);  
    __delay_ms(100);  
    CLKRCON = 0b11110100 ; // 50% 1 MHz  
}
```

```
void crescendo() {  
    int cnt;  
    int delaytime;  
    delaytime = 1000;  
    for(cnt=0 ; cnt<10 ; cnt++){  
        Pi();  
        delay(delaytime);  
        delaytime -= 80;  
    }  
}  
  
void SOS(){  
    CLKRCON = 0b11110100 ; // 50% 1 MHz  
    ToToTo();  
    CLKRCON = 0b11110101 ; // 50% 500 kHz  
    Tu();  
    CLKRCON = 0b11110100 ; // 50% 1 MHz  
    Tu();  
    CLKRCON = 0b11110101 ; // 50% 500 kHz  
    Tu();  
    CLKRCON = 0b11110101 ; // 50% 500 kHz  
    ToToTo();  
    CLKRCON = 0b11110100 ; // 50% 1 MHz  
    ToToTo();  
    CLKRCON = 0b11110101 ; // 50% 500 kHz  
    __delay_ms(300);  
}  
  
void second30(){  
    int cnt;  
    cnt = 30;  
    while(cnt-- > 0){  
        __delay_ms(1000);  
    }  
}  
  
main()  
{  
    //OSCCON = 0b01110010 ; //  
    //ANSELA = 0b00000000 ; //  
    //ANSELB = 0b00000000 ; //  
    //PORTA = 0b00000000; // PORTAの中身をきれいにする  
    //TRISA = 0b00000000 ; // PORTAの全ての端子を0:出力に設定  
    //TRISB = 0b00000000 ; //  
    //PORTA = 0b00000000 ; //  
    //PORTB = 0b00000000 ; //  
    //OSCCON = 0b00010010 ; // 31.25kHz  
    OSCCON = 0b01111111 ; //  
    OPTION_REG = 0b00000000 ;  
    ANSELA = 0b00000000 ;  
    WPUA = 0b00000000 ;  
    PORTA = 0b00000000 ;  
    TRISA = 0b00000000 ; // PORTAの全ての端子を0:出力に設定  
    MDCON = 0b01100000; // MDEN=1;MODE=1output_enable;  
    MDSRC = 0b10000000; // [bit3-0] 0000 = MDBIT  
    MDCARL = 0b10000000; // disable(OOK); MDCL=0000(Vss)  
    MDCARH = 0b10100011; // MDCH<3:0>0100(CCP1),0011(ReferClock)
```

```
LATA = 0xff;  
__delay_ms(1000);  
//SOS();  
TanTaTa();  
second30();  
  
SOS();  
SOS();  
TanTaTa();  
__delay_ms(500);  
crescendo();  
__delay_ms(500);  
crescendo();  
__delay_ms(500);  
SOS();  
  
second30();  
SOS();  
SOS();  
TanTaTa();  
__delay_ms(500);  
crescendo();  
__delay_ms(500);  
crescendo();  
__delay_ms(500);  
SOS();  
  
second30();  
  
while(1) {  
    SOS();  
    SOS();  
    TanTaTa();  
    __delay_ms(500);  
    crescendo();  
    __delay_ms(500);  
    crescendo();  
    __delay_ms(500);  
}  
}
```