簡易モールス練習機

概要

CW(モールス電信)は、昨今の無線通信技術の発展に伴い、ともすると忘れられてしまいがちな通信モードですが、アマチュア無線の世界ではまだ健在です。そこでこれから電信を初める方や、3 級~1級を目指す方までの 電気通信術の受信送信練習に最適な、簡易モールス練習機を製作してみました。

<仕様>

- 受信と送信を切り替えることが出来ます。(モード切替)
- 受信電文は、英数字をランダムに発生させます。(5文字発生、1秒休止を繰り返します)
- 受信速度は、10段階に切り替えることが出来ます。(速度切り替え)
- 送信は、電鍵(スイッチ)を接続するだけです。
- 信号(モールス音)は、800Hzの音の発生と、LEDの光の点灯で行います。
- 使用する電源は、乾電池2本とします(2V~5V)

動作原理

<受信電文の発生の仕組み> 英字(A~Z)と数字(0~9)の36文字は、乱数を用いて発生させます。 mikroCが標準で提供しているrand関数は、0~32767のランダムな数値を返します。 この数値を936で割ることにより0~35の数値に変換します。 この0~35の数値を、英字(A~Z)と数字(0~9)の36文字に変換します。 5文字分発生させて1語とします。その後1秒間休止します。

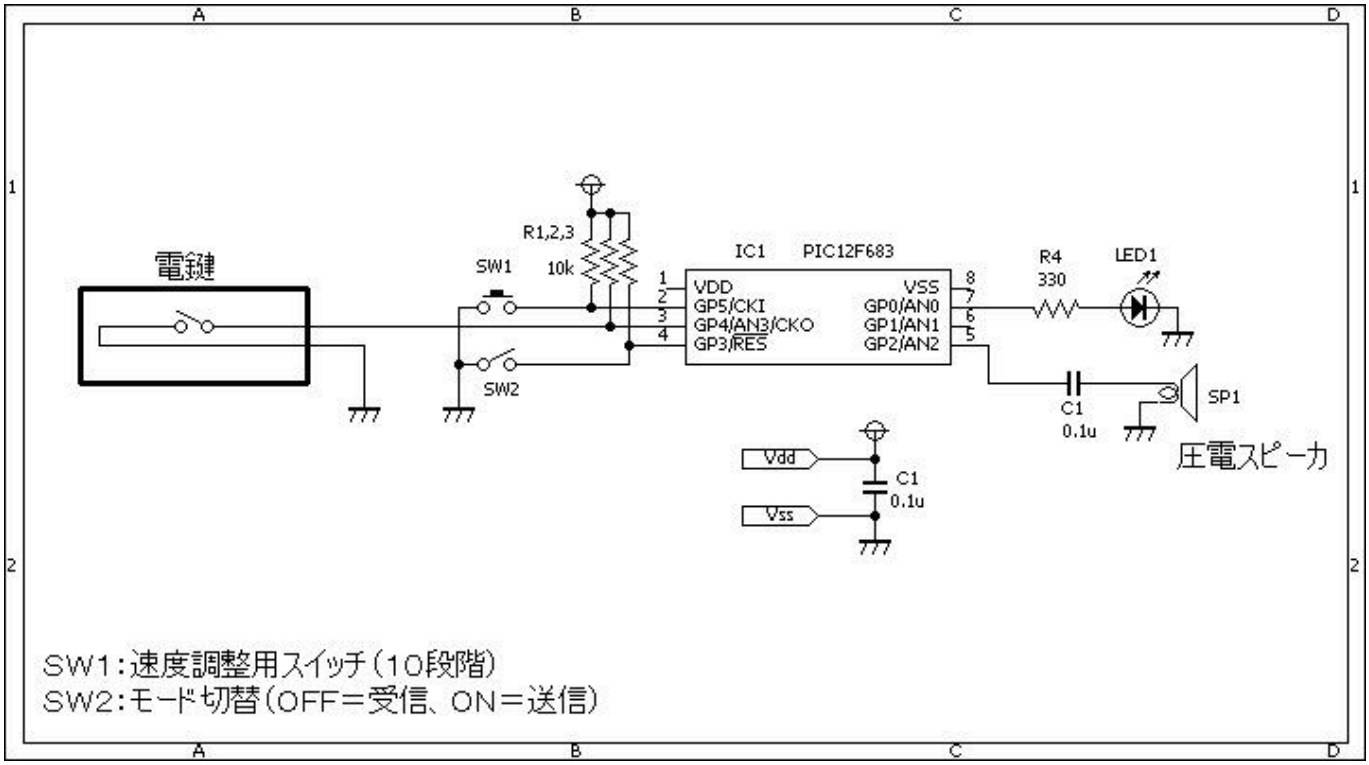
<同一系列の乱数の発生を防ぐための工夫> 起動時にTIMER1(16ビット)をスタートさせます。 操作者が電鍵を叩くことにより、TIMER1をストップします。 その時の、タイマー値をシード(種)値として、srand関数を使って設定します。

<受信速度の切り替え> 速度切り替えスイッチが押下される毎に、受信速度を切り替えていきます。(10段階) モールス音の短点の長さに例えると、次のようになります。 高速(1)=40msec 中速(5)=80msec 低速(10)=130msec

<送信の仕組み> 電鍵が叩かれるとモールス音をONします。 電鍵が離されるとモールス音をOFFします。

<モールス音の発生> モールス音は、一般的に800Hzの信号とされています PICが内蔵しているCCPモジュールをPWMモードで動作させて、約800Hzの矩形波の信号を得ます。 800Hzの信号を、圧電スピーカをとおして音に変換します。

回路図



ソースコード

morse_v2.c

```
//*****************************************************************************
*
/*
    ＜モールス練習機□□□
*/
//*****************************************************************************
*

#define       LED             GPIO.F0
#define       NON             GPIO.F1
#define       BEEP        GPIO.F2

#define       SW_MODE         GPIO.F3
#define       SW_KEY        GPIO.F4
#define       SW_SPEED    GPIO.F5

#define       DELAY_TIME    (30 + (speed_flg * 10))

//*****************************************************************************
*

const  char     table[][8] = {
    "A.-",
    "B-...",
```

```
    "C-.-.",
    "D-..",
    "E.",
    "F..-.",
    "G--.",
    "H....",
    "I..",
    "J.---",
    "K-.-",
    "L.-..",
    "M--",
    "N-.",
    "O---",
    "P.--.",
    "Q--.-",
    "R.-.",
    "S...",
    "T-",
    "U..-",
    "V...-",
    "W.--",
    "X-..-",
    "Y-.--",
    "Z--..",
    "1.----",
    "2..---",
    "3...--",
    "4....-",
    "5.....",
    "6-....",
    "7--...",
    "8---..",
    "9----.",
    "0-----",
/*
    "..-.-.-",
    ",--..--",
    ":---...",
    "?..--..",
    "'.----.",
    "--....-",
    "(-.--.",
    ")-.--.-",
    "/-..-.",
    "=-...-",
    "+.-.-.",
    "\".-..-.",
    "*-..-",
    "@.--.-."
*/
};
```

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

static short    speed_flg;

void   Delay_ms_ex(int ms)
{
    static    int    cnt;
    //
    for (cnt = 0; cnt < ms; cnt++) {
        Delay_us(1000);
        //
        if (SW_SPEED == 0) {
            while (SW_SPEED == 0) {
                Delay_ms(10);
            }
            if (speed_flg < 10) {
                speed_flg++;
            } else {
                speed_flg = 1;
            }
        }
    }
}

void   short_beep()
{
    Pwm_Start();
    LED = 1;
    Delay_ms_ex(DELAY_TIME);
    LED = 0;
    Pwm_Stop();
    Delay_ms_ex(DELAY_TIME);
}

void   long_beep()
{
    Pwm_Start();
    LED = 1;
    Delay_ms_ex(DELAY_TIME * 3);
    LED = 0;
    Pwm_Stop();
    Delay_ms_ex(DELAY_TIME);
}

void   morse_chr(char chr)
{
    static    short    cnt, i;
    //
```

```
    if (chr == ' ') {
        Delay_ms_ex(DELAY_TIME * 3);
        return;
    }
    //
    for (cnt = 0; cnt < 36; cnt++) {
        if (table[cnt][0] == chr) {
            for (i = 1; table[cnt][i] != 0x00; i++) {
                if (table[cnt][i] == '.')
                    short_beep();
                else
                    long_beep();
            }
            Delay_ms_ex(DELAY_TIME * 2);
            return;
        }
    }
}

void   morse_str(char* str)
{
    while (*str != 0x00) {
        morse_chr(*str);
        str++;
    }
}

//*****************************************************************************
*

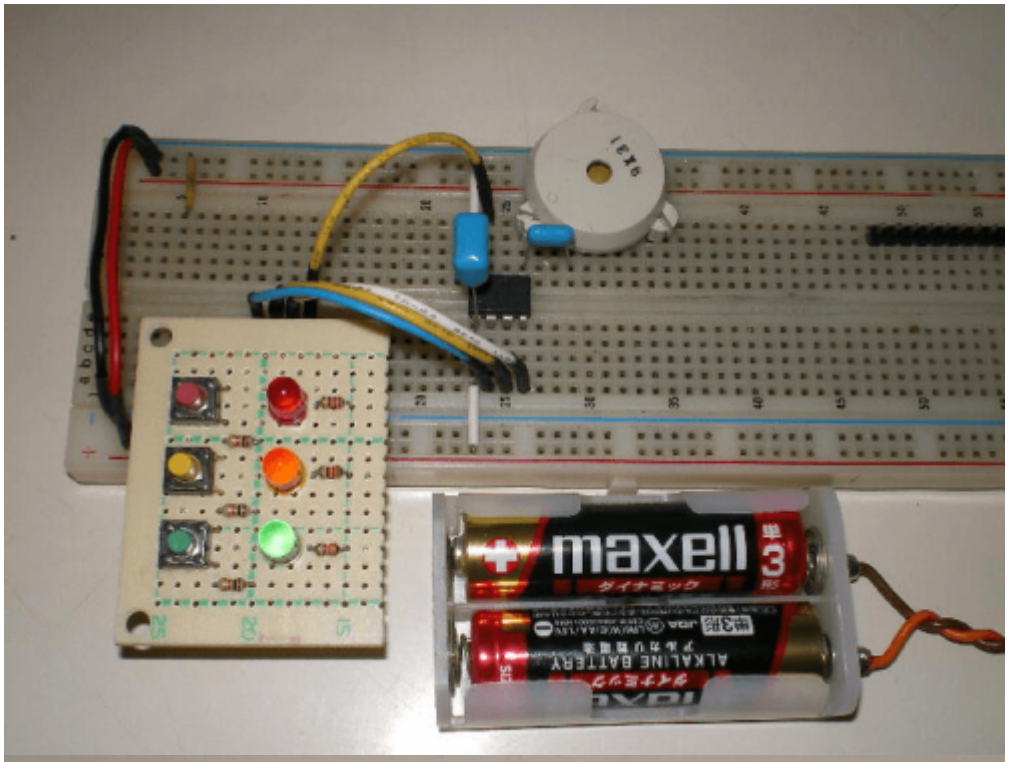
void main()
{
    static    short        cnt, i;
    static    char        buf[10];
    static    unsigned    seed;
    //
    OSCCON = 0b01010000;    //クロックは2Mhz
    CMCON0 = 0b00000111;    //コンパレータを使用しない。
    ANSEL  = 0b00000000;    // □□□変換は使用しない。
    TRISIO = 0b00111000;
    //□□□の初期化□□□□□□□
    Pwm_Init(800);
    Pwm_Change_Duty(PR2 / 2);
    Pwm_Stop();
    //
    LED = 0;
    speed_flg = 5;
    //
    morse_str("CQ CQ DE JF3SFB PSE K");
    Delay_ms(500);
    //乱数の種の設定
```

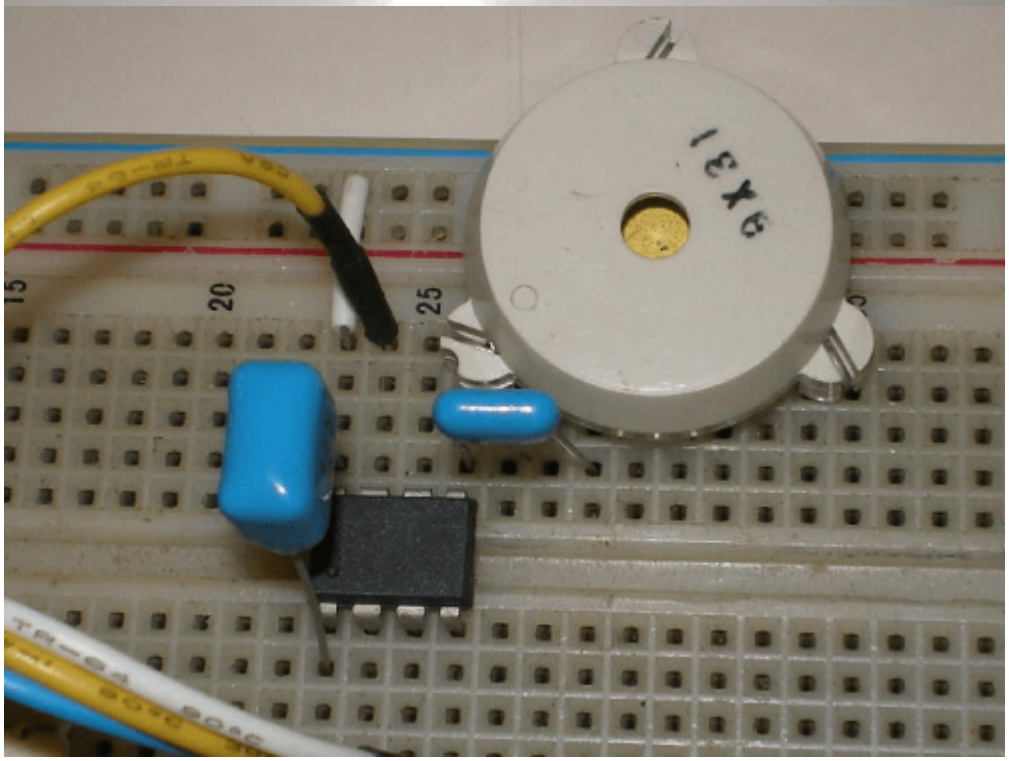
```
    T1CON.TMR1ON = 1;
    while (SW_KEY == 1) {
        LED = 1;
        Delay_ms(50);
        LED = 0;
        Delay_ms(50);
    }
    T1CON.TMR1ON = 0;
    seed = TMR1H;
    seed = (seed < 8) | TMR1L;
    srand(seed);
    //
    while (1) {
        if (SW_MODE == 1) {
            for (cnt = 0; cnt < 5; cnt++) {
                i = rand() / 936;    //i=0...35
                buf[cnt] = table[i][0];
            }
            buf[cnt] = 0x00;
            morse_str(buf);
            Delay_ms(1000);
        } else {
            if (SW_KEY == 0)
                Pwm_Start();
            else
                Pwm_Stop();
        }
    }
}

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

動作確認

ブレッドボードで動作確認しましたが、部品点数が少ないので、蛇の目基板に実装して、ボタン電池を使用すれ ばかなりコンパクトに仕上がると思います。





如何ですか? 今回は、電文の中には、記号(終点、小読点、問符、略符、左括弧、右括弧。。。)を含めていませんが、少しの 工夫で含めることが出来ます。是非、機能拡張に挑戦してみてください。😄

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://www.deepsky.jp/wiki/ - **うごくといいな**

Permanent link:
http://www.deepsky.jp/wiki/doku.php?id=elechobby:picdic:pic12f683:27&rev=1588321538

Last update: **2025/10/17 14:27**