

『電飾しましょっ！ 3 Tips 編』ヘッダーコード公開

2020年11月に刊行した『電飾しましょっ！ 3 PIC 編』で使用したPICのプログラム用のテンプレートを公開いたします。

正誤、公開ソースコードに関するお問い合わせは編集部までお願いいたします。

iwao@modelkasten.com

//PIC12F1822 テンプレート どろぼうひげ

```
/*      使用するキット名
      日付
      PIC12F1822
      プログラマー
*/

// PIC12F1822 Configuration Bit Settings

// CONFIG1
#pragma config FOSC = INTOSC      // Oscillator Selection (INTOSC oscillator: I/O function
on CLKIN pin)
#pragma config WDTE = OFF          // Watchdog Timer Enable (WDT disabled)
#pragma config PWRTE = ON          // Power-up Timer Enable (PWRT enabled)
#pragma config MCLRE = OFF         // MCLR Pin Function Select (MCLR/VPP pin function is
digital input)
#pragma config CP = OFF            // Flash Program Memory Code Protection (Program memory
code protection is disabled)
#pragma config CPD = OFF           // Data Memory Code Protection (Data memory code
protection is disabled)
#pragma config BOREN = OFF         // Brown-out Reset Enable (Brown-out Reset disabled)
#pragma config CLKOUTEN = OFF      // Clock Out Enable (CLKOUT function is disabled. I/O or
oscillator function on the CLKOUT pin)
#pragma config IESO = OFF         // Internal/External Switchover (Internal/External
Switchover mode is disabled)
#pragma config FCMEN = OFF        // Fail-Safe Clock Monitor Enable (Fail-Safe Clock Monitor
is disabled)

// CONFIG2
#pragma config WRT = OFF           // Flash Memory Self-Write Protection (Write protection
off)
#pragma config PLLEN = OFF        // PLL Enable (4x PLL enabled)
#pragma config STVREN = ON        // Stack Overflow/Underflow Reset Enable (Stack Overflow or
Underflow will cause a Reset)
#pragma config BORV = LO          // Brown-out Reset Voltage Selection (Brown-out Reset
Voltage (Vbor), low trip point selected.)
```

```
#pragma config LVP = OFF          // Low-Voltage Programming Enable (Low-voltage programming
enabled)

#include <xc.h>
#include "stdlib.h"                //rand 関数を使う時は有効にする
#define _XTAL_FREQ 8000000        //delay 関数を使う時は有効にする

void main(void) {

    OSCCON = 0b01110010; // 内部クロック 8MHz 4MHz は 0b01101010
    ANSELA = 0b00000000; // アナログは使用しない (すべてデジタル I/O に割当てて)
    TRISA = 0b00001000;      // RA3 だけ入力その他のピンは出力に割当てて (RA3 は入力専用)

    PORTA = 0;                // 出力ピンの初期化 (全て LOW にする)

    while(1){

        //ここにプログラムを書きます

    } //End of while

} //End of main
```

//PIC16F1827 テンプレート どろぼうひげ

```
/*    使用するキット名
      日付
      PIC16F1827
      プログラマー
*/
```

```
// PIC16F1827 Configuration Bit Settings
```

```
// CONFIG1
```

```
#pragma config FOSC = INTOSC    // Oscillator Selection (INTOSC oscillator: I/O function
on CLKIN pin)
#pragma config WDTE = OFF        // Watchdog Timer Enable (WDT disabled)
#pragma config PWRTE = ON        // Power-up Timer Enable (PWRT enabled)
#pragma config MCLRE = OFF       // MCLR Pin Function Select (MCLR/VPP pin function is
digital input)
#pragma config CP = OFF          // Flash Program Memory Code Protection (Program memory
code protection is disabled)
#pragma config CPD = OFF         // Data Memory Code Protection (Data memory code
protection is disabled)
#pragma config BOREN = OFF       // Brown-out Reset Enable (Brown-out Reset disabled)
#pragma config CLKOUTEN = OFF    // Clock Out Enable (CLKOUT function is disabled. I/O or
oscillator function on the CLKOUT pin)
#pragma config IESO = OFF       // Internal/External Switchover (Internal/External
Switchover mode is disabled)
#pragma config FCMEN = OFF      // Fail-Safe Clock Monitor Enable (Fail-Safe Clock Monitor
is disabled)
```

```
// CONFIG2
```

```
#pragma config WRT = OFF        // Flash Memory Self-Write Protection (Write protection
off)
#pragma config PLLEN = OFF       // PLL Enable (4x PLL enabled)
#pragma config STVREN = ON       // Stack Overflow/Underflow Reset Enable (Stack Overflow or
Underflow will cause a Reset)
#pragma config BORV = LO         // Brown-out Reset Voltage Selection (Brown-out Reset
Voltage (Vbor), low trip point selected.)
#pragma config LVP = OFF         // Low-Voltage Programming Enable (Low-voltage programming
enabled)
```

```
#include <xc.h>
```

```
##include "stdlib.h"           //rand 関数を使う時は有効にする
```

```
##define _XTAL_FREQ 8000000    //delay 関数を使う時は有効にする
```

```
void main(void) {

    OSCCON = 0b01110010; // 内部クロック 8MHz 4MHz は 0b01101010
    ANSELA = 0b00000000; // アナログは使用しない (すべてデジタル I/O に割当てる)
    ANSELB = 0b00000000;

    TRISA = 0b00100000; // RA5 だけ入力その他のピンは出力に割当てる (RA5 は入力専用)
    TRISB = 0b00000000; //PORTB も全て出力に割り当てる

    PORTA = 0; // 出力ピンの初期化 (全て LOW にする)
    PORTB = 0; //PORTB も初期化

    while(1){

        // ここにプログラムを書きます

    } //End of while

} //End of main
```