無線リモートコントローラ受信部 (品番RF-0299-RX)説明書

概要

本キットは、マイクロチップ社ナノワット超低電力(XLP)テクノロジー、8ビットマイクロコントローラ、ミッドレンジ、エンハンスシリーズPIC 16F1826を使用した無線リモートコントローラ(ラジコン)受信部(品番RF-0299-RX)(完成品)です。

無線リモートコントローラ(ラジコン)送信部(品番RF-0299L-TX)と組み合わせて、315メガ帯での無線リモコン制御ができます。

本機は送信側キースイッチ対応したキーコードを16進で出力します。

また、本機の出力に10進変換アダプター(品番HtoD-0297)を繋げば出力信号が保持され、種々の機器の制御が簡単になります。

特長

1. 消費電力が小さい。約2. 5mA@5V
2. 小型でローコスト
3. 16個のキースイッチを使って最大26種類のキーコード(5ビット)を受信。
4. 16個のキースイッチ中、二度押しスイッチが4個、四度押しスイッチが2個、単発スイッチが10個ある。
5. デバイスコードは、4ビットDIPスイッチを使って最大14個のデバイスを設定できる。
6. 通信可能距離は直線で約50m(注1)
7. 動作電源電圧は4.0V-5.0V

準備するもの

1. 無線リモートコントローラ(ラジコン)受信部本体(品番RF-0299-RX)
   完成品、ICはプログラム済み、動作確認済み
2. 無線リモートコントローラ(ラジコン)送信部(品番RF-0299L-TX)
3. 5V電源(単一電池3本直列接続可能)
4. 3ピンのピンソケット(メス)1個
5. キーパッド(品番ACC-0295-KEY4x4)1個
6. 10cm程の線材(アンテナとして使用する)。
7. 10進変換アダプター(品番HtoD-0297)(オプション)

組み立て

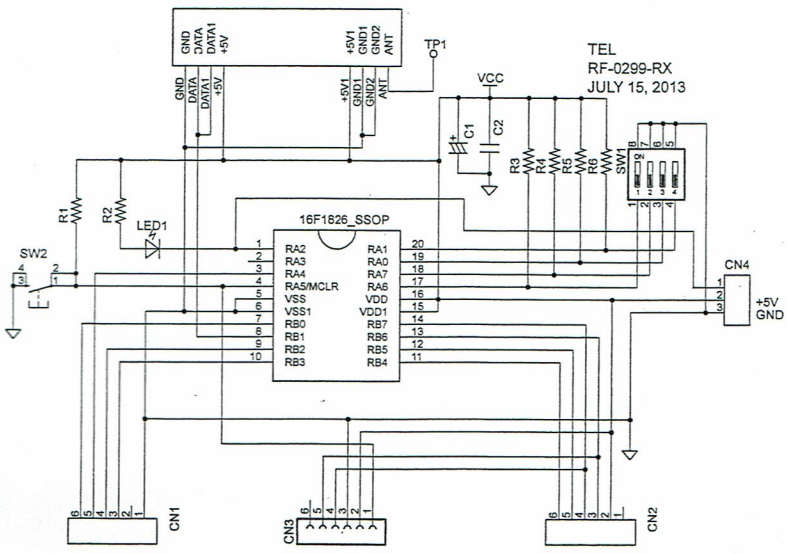
1. キーパッドのピンヘッダCN1,CN2と本体のピンソケットCN1,CN2のコネクターを差し込む。
2. 2P電池ケースのプラス側を本体(CN4)5V、マイナス側をグランドGNDにピンソケットを使ってCN4に繋ぐ。
   極性に十分注意すること。
3. 本機に電源が入れば直ちにラジコン受信機として動作する。

キーパッド各スイッチと出力(5ビットキーコード)の関係

送信側のキーパッド各スイッチと受信側の出力の関係を表1に示す。

また十進変換アダプター(品番HtoD-0297)を接続すれば送信側キーパッドの各スイッチに対応した発光ダイオード(LED)が点灯する。デバイスコードのLED表示は下三桁のみ。例えば、デバイスコードが"1110"の場合、"110"と表示される。

無線リモートコントローラ受信部 (品番RF-0299-RX) 回路図



単発パルス出力について

本機は受信の度にLED1が点灯し、コネクターCN4の1番ピンに約65mSの負論理パルス信号が出力されている。つまり、1番ピンは通常Hi であるが、送信側のキーパットの何れかのスイッチが押されると発光ダイオードLED1と連動してLoの短いパルス信号が出る。特定のスイッチに対する単発パルス出力が必要なときは、1番ピンと10進変換アダプター(品番HtoD-0297)の特定のスイッチのコネクター出力ピンとNORゲートをとること。(後述する使用例参照)

キースイッチ機能

本機にキーパットは必要なし。キーパッドを接続しないこと。

送信側キーパッドのスイッチ機能を下記する。

1、二度押しスイッチはトグルスイッチとして使用できる。
   例えば、パワースイッチなどON/OFFスイッチ。
2、四度押しスイッチは4個の設定が必要な制御に使用できる。
   例えば、音量調整(切る、小、中、大)、風速調整(切る、弱、中、強)。
3、単発キーは十連のロータリースイッチとして使える。
   また、スライドスイッチ(2個以上のスイッチを使う)としても使用できる。
   初期値(デフォルト値)は10進変換アダプターで設定しSW7とした。
4、.リセット(RESET SW1)、電源投入時の二度押し四度押しキーの
   デフォルト値は、SW1-0,SW2-0,SW3-0,SW4-0,SW5-0,SW6-0である。

デバイスコードの変更

デバイスコードは4PDIPスイッチSW1を使って設定する。1番がLSB (LEAST-SIGNIFICANT-BIT)、ONにするとTRUE(負論理で'0')となる。

4ビット使用で"0001"から"1111"まで14種類の設定が可能。

"0001" "0010" "0011" "0100" "0101" "0110" "0111" "1001" "1010" "1011" "1100" "1101" "1110"(デフォルト) "1111"

但し、"0000","1000"(DIPスイッチ1,2,3番ON)は使用禁止。

例えば"1110"の場合DIPスイッチの1をON、2,3,4をOFFとする。

デバイスコードは送信側と必ず一致させる。

通信形式について

本機はEUSERT(Enhanced Universal Synchronous Receiver Transmitter)周辺回路モジュールを使用し、無線でシリアル受信を行う。

一個のデータは8ビットで、LSBからMSBの順にボーレートジェネレターのクロック信号に同期して自動的にサンプリングされRCREGに格納される。

先ず、8個のデータを一回分として、最初に文字データ'T' ,'E', 'L' を受信し、続いて、デバイスコード、データコード、反転デバイスコード、反転データコード、最後に文字データ'E'を受信し一回分を終了する。

次に同じ8個のデータを三回、1mSの間隔で連続受信し、一回の通信を完了する。

プログラムについて

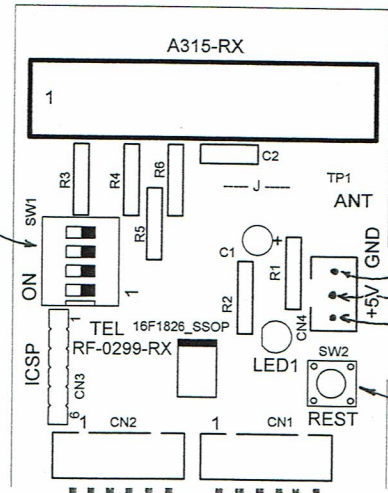
MPLAB XC8 C コンパイラー ライトモードV1.20を使用。
ROM(プログラムメモリ) 2048ワード中1172使用 約57%
RAM(データーメモリ) 884バイト中40バイト使用 約5%
参考資料としてCソースファイルのリスト(Listing)を添付する。
(注意)CONFIGワードのコードプロテクションビット はOFFになっており、ICプログラムメモリの上書き読込みが自由にできますが、ICSPを使用しプログラムを変更した場合には、保証できなくなりますから注意してください。また、プログラムに関するご質問はお受けできませんのでご了承ください。

注1:通信可能距離は直線は障害物、アンテナの長さ、形状などによって大きく変わってくる。受信側のアンテナについては制限がないが、送信側のアンテナを余り大きくすると電波法に抵触する可能性があるので注意すること。

T.E.L. キット製造販売
(有)谷岡電子
〒164 東京都中野区東中野1-51-13
-0003 大島ビル第一別館402
☎03-3366-4552

無線リモートコントローラ受信部
(品番RF-0299-RX)と
10進変換アダプター(品番HtoD-0297)
を組み合わせて使用した例

無線リモートコントローラ受信部
(品番RF-0299-RX) 本体



LED1
受信ごとに
約65mS点灯

ANT
10cm程の線材を
アンテナとして
ここに接続する。

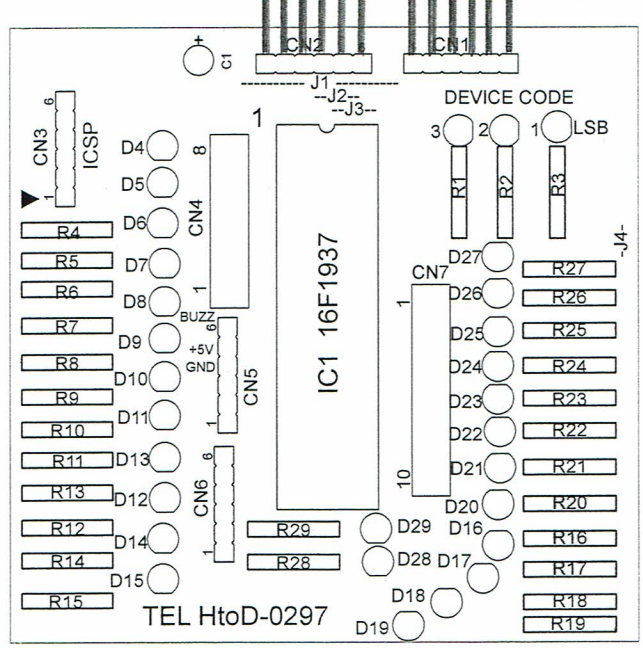
(注意)
デバイスコード"0000","1000"
(DIPスイッチ1,2,3番ON)は
使用禁止。動作しません。

DIPスイッチ
4ビットデバイス
コードを設定。
例えば"1110"は
1番をON, 2,3,4番
をOFFとする。
送信側も"1110"と
設定すること。

電源グランドGND
5V電源プラス側
単発パルス出力
(約65mS)

ICSP(In-Circuit
Serial Programming)
用コネクター。
使用しないこと。

RESET(SW2)
本機をリセットし
デフォルト値に
戻す。
10進変換アダプター
(品番HtoD-0297)
にリセットをかけ
デフォルト(初期値)
に戻す。



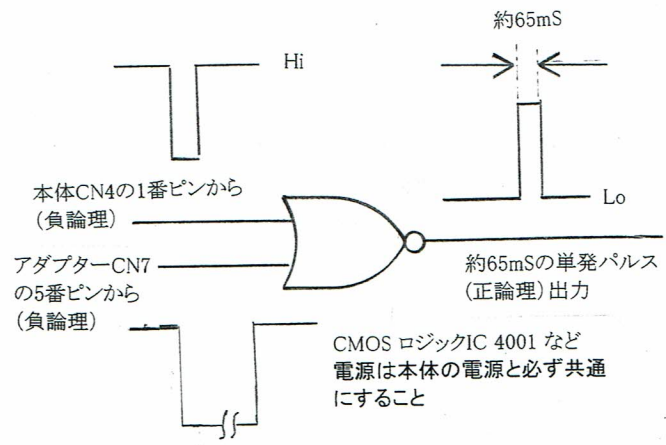
10進変換アダプター
(品番HtoD-0297)

10進変換アダプターの詳細
については10進変換アダプター
の説明書参照

表1 キーパッド各スイッチと出力(5ビットキーコード)の関係

コネクター番号	CN2-3	CN2-4	CN2-5	CN2-6	CN1-3	CN1-4	CN1-5	CN1-6
ポート	RB6	RB7	RB5	RB4	RB3	RB2	RA4	RB0
	3ビットデバイスコード(負論理) デイフォルト値='110'				5ビットキーコード(正論理)			
SW1-0	1	1	0	0	0	0	0	0
SW1-1	1	1	0	0	0	0	0	1
SW2-0	1	1	0	0	0	0	1	0
SW2-1	1	1	0	0	0	0	1	1
SW3-0	1	1	0	0	0	1	0	0
SW3-1	1	1	0	0	0	1	0	1
SW4-0	1	1	0	0	0	1	1	0
SW4-1	1	1	0	0	0	1	1	1
SW5-0	1	1	0	0	1	0	0	0
SW5-1	1	1	0	0	1	0	0	1
SW5-2	1	1	0	0	1	0	1	0
SW5-3	1	1	0	0	1	0	1	1
SW6-0	1	1	0	0	1	1	0	0
SW6-1	1	1	0	0	1	1	0	1
SW6-2	1	1	0	0	1	1	1	0
SW6-3	1	1	0	0	1	1	1	1
SW7	1	1	0	1	0	0	0	0
SW8	1	1	0	1	0	0	0	1
SW9	1	1	0	1	0	0	1	0
SW10	1	1	0	1	0	0	1	1
SW11	1	1	0	1	0	1	0	0
SW12	1	1	0	1	0	1	0	1
SW13	1	1	0	1	0	1	1	0
SW14	1	1	0	1	0	1	1	1
SW15	1	1	0	1	1	0	0	0
SW16	1	1	0	1	1	0	0	1

単発パルスの作り方の一例
単発キーSW7と単発パルス出力とのNOR ゲート
から単発パルスを得る方法。



無線リモートコントローラ受信部
(品番RF-0299-RX)
Cソースファイルリスト(Listing)

```
/*****************************************************************
******* RF Wireless Remote Controller Receiver *******
* Project name: RF0299RX
* File name: RF0299RX.c
* Features:
*  *nano watt extreme low power(XLP), 1.8μwatt typical,
*  *Micro controller PIC enhanced mid-range 16F1826.
*  *3 bits device code
*  *Wide operating voltage, 4.0V-5.0V.
* Pin assignments:
*  *Device code input <RA6:RA7>,<RA1:RA0>
*  *Data Input  <RB1>
*  *Device Code Output <RB7:RB4>
*  *Key Code Output(Data Output) <RB3:RA4> <RB2:RB0>
*  *Led Indicator Output <RA2>
* Compiler: xc8 v1.12
* AUG 10, 2013. Coded by yuji Tanioka(TEL company)
*****************************************************************/
#include <xc.h>

#pragma config MCLRE = ON, CP = OFF, BOREN = OFF, WDTE = OFF,
#pragma config PWRTE = ON, CPD = OFF, CLKOUTEN = OFF, FCMEN = OFF
#pragma config WRT = OFF, STVREN = OFF, BORV = LO, LVP = OFF,
#pragma config FOSC = INTOSC, PLLEN = OFF

#define     _XTAL_FREQ      8000000                 //8MHZ
#define     STR_LEN         8
#define     ERROR           0xFF
#define     TRUE            1
#define     FALSE           0

/*funcfion prototype declaration*/
void reg_init(void);                                //F13
void rf_receive(void);                              //F7
unsigned char getch(void);                          //F10
void output_table(unsigned char pin, unsigned char key);   //F11
void led_on(void);                                  //F18
void interrupt isr(void);                           //F17

typedef     bit     bool;

/*GLOBAL VARIABLES*/
unsigned char   str_buf[STR_LEN + 1];
unsigned char   str_pos = 0;
bool    str_flag = FALSE;

/*Main function*/
void main(void)
{
    reg_init();                                     //F13

    for(;;)
        rf_receive();

}

/*function definition F13, register initialization*/
void reg_init(void)
{
    OSCCON = 0b01110000;            //8Mhz,HF:IRCF=01110
    OSCTUNE = 0x00;
    TRISA = 0b11000011;             //input <RA6:RA7>,<RA1:RA0>
    TRISB = 0b00000010;             //RB1 INPUT
    PORTB = 0;
    RA4 = 0;
    ANSELA = 0x00; ANSELB = 0x00;   //DIGITAL I/O
    WPUA5 = 0;                      //WEAK PULL-UP RA5 DISABLED
    WPUB = 0x00;                    //WEAK PULL-UP DISABLED
```



```
    nWPUEN = 1;
    LATA2 = 1;
    BAUDCON = 0b00000000;           //0x08
              //---0----            //CLOCK POLARITY SELECT SCKP=0
              //----0---            //BRG16=0

    //SPBRGL = 0X40;                //BAUD RATE=2400,BRGH=1,BRG16=1
    //SPBRGH = 0X03;                //832(0X0340), ERROR 0.04%
    //SPBRGL = 0X0A;                //BAUD RATE=300,BRGH=1,BRG16=1
    //SPBRGH = 0X1A;                //6666(0X1A0A), ERROR 0%
    SPBRGL = 103;                   //BAUD RATE=1200,BRGH=0,BRG16=0
    SPBRGH = 0;                     //103 ERROR 0.16%

    TXIE = 0;                       //TX1 INTERRUPT DISABLED

    TXSTA = 0b00000000;             //0x00
            //-0------              TX9=0
            //--0-----              TXEN=0
            //---0----              SYNC=0
            //-----0--              BRGH=0

    RCIE = 1;                       //RC INTERRUPT ENABLED

    RCSTA = 0b10010000;             //0x90
            //1-------              //Serial Port enable(SPEN)
            //-0------              //use ninth bit(RX9)
            //---1----              //CREN=1
            //----0---              //ADDEN=0


    ADDEN = 0;
    SYNC = 0;

    T2CON = 0b00111011;             //0x0B
            //-0111---              1:8 POSTSCALER T2OUTPS<3:0>
            //-----0--              TIMER2 OFF
            //------11             PRESCALER IS 64 T2CKPS<1:0>]
    TMR2 = 0x00;                    //TIMER 2 CLEARE
    PR2 = 0xFF;                     //TOTAL LED ON TIME:
                                    //0.5μSx64x256x8=65.536mS
    TMR2IE = 1;                     //TIMER 2 ENABLED

    PEIE = 1;
    ei();
}

void rf_receive(void)               //F7
{
    unsigned char pin_code, key_code, n_key_code, pin_data, n_pin_data;

    pin_code = PORTA;
    pin_code &= 0xC3;               //CLEAR <RA5:RA2>
    pin_code = pin_code << 2 | pin_code >> 6;

    CREN = 1;

    if(getch() == 'T')
        str_pos++;
    else
    {
        str_pos = 0;
        return;
    }

    if(getch() == 'E')
        str_pos++;
    else
    {
```



```
        str_pos = 0;
        return;
    }

    if(getch() == 'L')
        str_pos++;
    else
    {
        str_pos = 0;
        return;
    }

    pin_data = getch();             //read device code
    pin_data &= 0x0F;
    if(pin_code == pin_data)        //device code test
        str_pos++;
    else
    {
        str_pos = 0;
        return;
    }

    key_code = getch();             //read key code
    str_pos++;
    n_pin_data = ~getch();          //read compliment device code
    n_pin_data &= 0x0F;
    if(pin_code == n_pin_data)      //pin code compliment test
    str_pos++;
    else
    {
        str_pos = 0;
        return;
    }

    n_key_code = ~getch();          //read compliment key code
    if(key_code == n_key_code)      //key code test
        str_pos++;
    else
    {
        str_pos = 0;
        return;
    }
    if(getch() == 'E')
        str_pos++;
    else
    {
        str_pos = 0;
        return;
    }

    if(!OERR && !FERR)
    {
        output_table(pin_code, key_code);
        led_on();
    }
    if(OERR)
        CREN = 0;
    str_pos = 0;
}

void interrupt isr(void)
{
    if(RCIF)
    {
        str_flag = TRUE;
        str_buf[str_pos] = RCREG;
    }
```



```
    if(TMR2IE && TMR2IF)
    {
TMR2IF = 0;
TMR2ON = 0;
        LATA2 = 1;              //LED OFF
    }
}

/*function definition F18, led on */
void led_on(void)                                   //F18
{
    PR2 = 0xFF;
    TMR2IE = 1;
    TMR2ON = 1;
    LATA2 = 0;                                      //LED ON
}

unsigned char getch(void)                           //F10
{
    while(str_flag == FALSE);
    str_flag = FALSE;
    return str_buf[str_pos];
}

/*function definition F11, OUTPUT TABLE*/
void output_table(unsigned char pin, unsigned char key)   //F11
{
    unsigned char sPORTA, sLATB;

    sPORTA = pin;
    sLATB = LATB;
    sPORTA &= 0x0F;
    sLATB &= 0x1F;
    sPORTA <<= 5;
    sLATB |= sPORTA;
    LATB = sLATB;               //PIN CODE OUTPUT

    switch(key)                 //KEY CODE OUTPUT
    {
        case 0:
            sLATB = LATB;
            sLATB &= 0xE2;
            sLATB |= 0x00;
            LATB = sLATB;
            LATA4 = 0;
            break;
        case 1:
            sLATB = LATB;
            sLATB &= 0xE2;
            sLATB |= 0x01;
            LATB = sLATB;
            LATA4 = 0;
            break;
        case 2:
            sLATB = LATB;
            sLATB &= 0xE2;
            sLATB |= 0x00;
            LATB = sLATB;
            LATA4 = 1;
            break;
        case 3:
            sLATB = LATB;
            sLATB &= 0xE2;
            sLATB |= 0x01;
            LATB = sLATB;
            LATA4 = 1;
            break;
        case 4:
```

```
273             sLATB = LATB;
274             sLATB &= 0xE2;
275             sLATB |= 0x04;
276             LATB = sLATB;
277             LATA4 = 0;
278             break;
279         case 5:
280             sLATB = LATB;
281             sLATB &= 0xE2;
282             sLATB |= 0x05;
283             LATB = sLATB;
284             LATA4 = 0;
285             break;
286         case 6:
287             sLATB = LATB;
288             sLATB &= 0xE2;
289             sLATB |= 0x04;
290             LATB = sLATB;
291             LATA4 = 1;
292             break;
293         case 7:
294             sLATB = LATB;
295             sLATB &= 0xE2;
296             sLATB |= 0x05;
297             LATB = sLATB;
298             LATA4 = 1;
299             break;
300         case 8:
301             sLATB = LATB;
302             sLATB &= 0xE2;
303             sLATB |= 0x08;
304             LATB = sLATB;
305             LATA4 = 0;
306             break;
307         case 9:
308             sLATB = LATB;
309             sLATB &= 0xE2;
310             sLATB |= 0x09;
311             LATB = sLATB;
312             LATA4 = 0;
313             break;
314         case 10:
315             sLATB = LATB;
316             sLATB &= 0xE2;
317             sLATB |= 0x08;
318             LATB = sLATB;
319             LATA4 = 1;
320             break;
321         case 11:
322             sLATB = LATB;
323             sLATB &= 0xE2;
324             sLATB |= 0x09;
325             LATB = sLATB;
326             LATA4 = 1;
327             break;
328         case 12:
329             sLATB = LATB;
330             sLATB &= 0xE2;
331             sLATB |= 0x0C;
332             LATB = sLATB;
333             LATA4 = 0;
334             break;
335         case 13:
336             sLATB = LATB;
337             sLATB &= 0xE2;
338             sLATB |= 0x0D;
339             LATB = sLATB;
340             LATA4 = 0;
```



```
341             break;
342         case 14:
343             sLATB = LATB;
344             sLATB &= 0xE2;
345             sLATB |= 0x0C;
346             LATB = sLATB;
347             LATA4 = 1;
348             break;
349         case 15:
350             sLATB = LATB;
351             sLATB &= 0xE2;
352             sLATB |= 0x0D;
353             LATB = sLATB;
354             LATA4 = 1;
355             break;
356         case 16:
357             sLATB = LATB;
358             sLATB &= 0xE2;
359             sLATB |= 0x10;
360             LATB = sLATB;
361             LATA4 = 0;
362             break;
363         case 17:
364             sLATB = LATB;
365             sLATB &= 0xE2;
366             sLATB |= 0x11;
367             LATB = sLATB;
368             LATA4 = 0;
369             break;
370         case 18:
371             sLATB = LATB;
372             sLATB &= 0xE2;
373             sLATB |= 0x10;
374             LATB = sLATB;
375             LATA4 = 1;
376             break;
377         case 19:
378             sLATB = LATB;
379             sLATB &= 0xE2;
380             sLATB |= 0x11;
381             LATB = sLATB;
382             LATA4 = 1;
383             break;
384         case 20:
385             sLATB = LATB;
386             sLATB &= 0xE2;
387             sLATB |= 0x14;
388             LATB = sLATB;
389             LATA4 = 0;
390             break;
391         case 21:
392             sLATB = LATB;
393             sLATB &= 0xE2;
394             sLATB |= 0x15;
395             LATB = sLATB;
396             LATA4 = 0;
397             break;
398         case 22:
399             sLATB = LATB;
400             sLATB &= 0xE2;
401             sLATB |= 0x14;
402             LATB = sLATB;
403             LATA4 = 1;
404             break;
405         case 23:
406             sLATB = LATB;
407             sLATB &= 0xE2;
408             sLATB |= 0x15;
```



```
409             LATB = sLATB;
410             LATA4 = 1;
411             break;
412         case 24:
413             sLATB = LATB;
414             sLATB &= 0xE2;
415             sLATB |= 0x18;
416             LATB = sLATB;
417             LATA4 = 0;
418             break;
419         case 25:
420             sLATB = LATB;
421             sLATB &= 0xE2;
422             sLATB |= 0x19;
423             LATB = sLATB;
424             LATA4 = 0;
425             break;
426         default: break;
427
428     }
429
430 }
```