送受リモコンモジュール(赤外線データ通信)

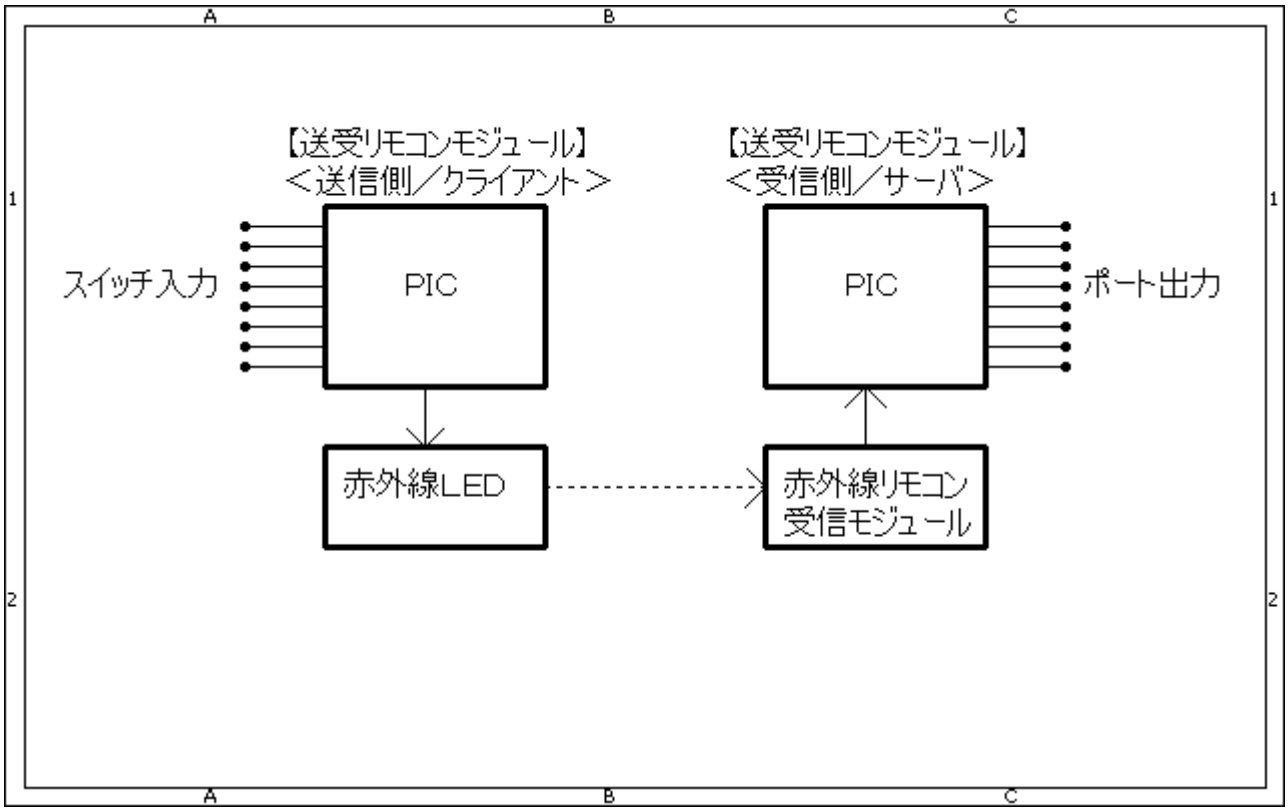
概要

前回製作した、「ワイヤレスLCD(赤外線データ通信)」は、データ(文字)を赤外線で送信して□LCDに表示させるためのものでした。

今回製作した、「送受リモコンモジュール(赤外線データ通信)」は、データ(チャネル番号)を赤外線で送信して、ポートをON/OFFさせるためのものです。

<仕様>

- 制御する出力ポート数は、8チャネルとします。
- その内、4チャネルは、スイッチを押下する毎に、ON/OFFが切り替わります。
- 残りの、4チャネルは、スイッチを押下している間だけONします。
- データ通信には、赤外線(副搬送波(38kHz))を用います。
- ひとつのPICに送信機能および受信機能を搭載します。(起動時のスイッチで指定します)



動作原理

送受リモコンモジュールは、

- リモコンの受信側【サーバ(Server)□
- リモコン送信側【クライアント(Client)□

より構成されます。(クライアント・サーバ方式)

動作原理(ハードウェア)

ワイヤレスLCD(赤外線データ通信)を参照してください。

動作原理(ソフトウェア)

◎メイン処理 スイッチ(SW1)の設定により、サーバ側、クライアント側を切り替え、各処理を呼び出します SW1=GND サーバ側(Remocon_Server関数) SW1=Vdd クライアント側(Remocon_Client関数)

◎クライアント処理 副搬送波(38kHz)を発生させるために CCPモジュールをPWMモードで初期化します。 データは、1200bpsの速度で送信(ソフトウエア処理)します。

- スタートビット(1ビット)
- データビット(8ビット)
- ストップビット(1ビット)
  ※1ビットあたりの送信時間は、約833usec(1秒÷1200bps)になります。

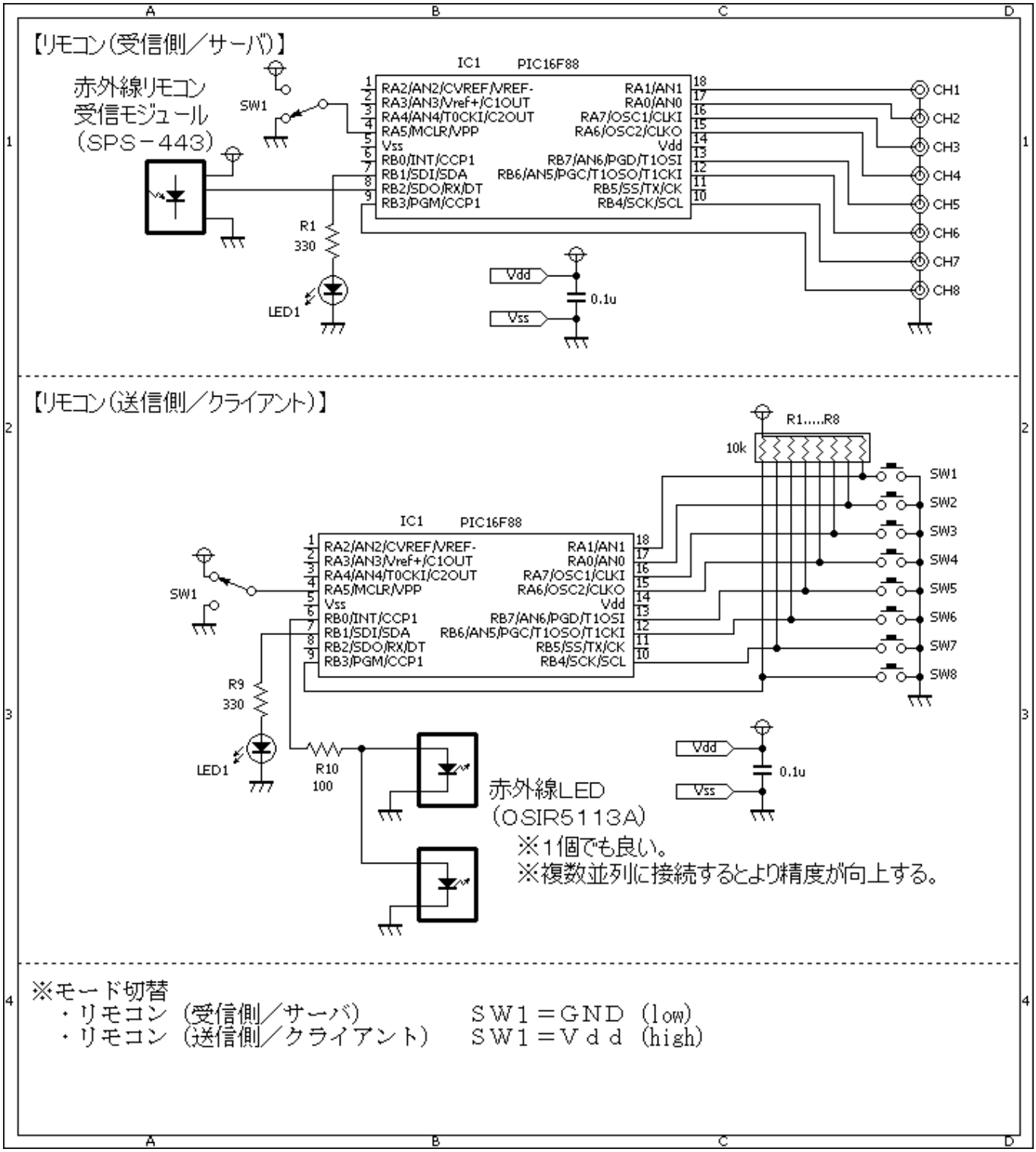
SW1~SW8をチェックし、押下されていれば、押下されたスイッチの番号をサーバ側に送信します。

- SW1~SW4は、スイッチの押下状態から離された状態になることを確認してからデータを送信します。
- SW5~SW8は、スイッチの押下状態の間、連続してデータを送信し続けます。

◎サーバ処理 USART(1200bps)を初期化します。 クライアントからのデータを受信します。 受信データを解析し、ポートを制御します。

- CH1~CH4は、ON状態であればOFFに、OFF状態であればONにします。
- CH5~CH8は、100msec間ON状態にします。この間に連続してデータを受信すると ON状態が持続することになります。

回路図



ソースコード

remote_controller.c

```
//*****************************************************************************
*
/*
    <送受リモコンモジュール（赤外線データ通信）>
*/
```

```
//*********************************************************************
*
//■マクロ定義
#define BYTE        unsigned short
#define WORD        unsigned int
#define DWORD       unsigned long
//
sbit CH1    at RA1_bit;
sbit CH2    at RA0_bit;
sbit CH3    at RA7_bit;
sbit CH4    at RA6_bit;
sbit CH5    at RB7_bit;
sbit CH6    at RB6_bit;
sbit CH7    at RB4_bit;
sbit CH8    at RB3_bit;
//
sbit SW1    at RA1_bit;
sbit SW2    at RA0_bit;
sbit SW3    at RA7_bit;
sbit SW4    at RA6_bit;
sbit SW5    at RB7_bit;
sbit SW6    at RB6_bit;
sbit SW7    at RB4_bit;
sbit SW8    at RB3_bit;
//
sbit SW     at RA5_bit;
//
sbit LED    at RB1_bit;
#define ON          1
#define OFF         0
//*********************************************************************
*
//■関数宣言
extern  void    main();
extern  void    Remocon_Server();
extern  void    Remocon_Client();
extern  void    Remocon_Send_Chr(unsigned short dt);
extern  void    interrupt();
extern  void    timer_start();
extern  void    led_on_off();
//*********************************************************************
*
//■メイン関数
void    main()
{
        OSCCON = 0b01110000;
        ANSEL  = 0b00000000;
        //
        if (SW == 1) {
                Remocon_Client();
        } else {
```

```
                Remocon_Server();
        }
}
//*****************************************************************************
*
//■リモコンモジュールサーバ関数（受信用）
void    Remocon_Server()
{
        short   cnt;
        //
        TRISA  = 0b00111100;
        TRISB  = 0b00000101;
        //
        CH1 = OFF;
        CH2 = OFF;
        CH3 = OFF;
        CH4 = OFF;
        CH5 = OFF;
        CH6 = OFF;
        CH7 = OFF;
        CH8 = OFF;
        //
        Uart1_Init(1200);
        //
        for (cnt = 0; cnt < 10; cnt++) {
                LED = ON;
                Delay_ms(50);
                LED = OFF;
                Delay_ms(50);
        }
        //
        while(1) {
                if (Uart1_Data_Ready() != 1) {
                        continue;
                }
                //
                switch (Uart1_Read()) {
                case '1':
                        CH1 = ~CH1;
                        break;
                case '2':
                        CH2 = ~CH2;
                        break;
                case '3':
                        CH3 = ~CH3;
                        break;
                case '4':
                        CH4 = ~CH4;
                        break;
                case '5':
                        timer_start();
```

```
                        CH5 = ON;
                        break;
                case '6':
                        timer_start();
                        CH6 = ON;
                        break;
                case '7':
                        timer_start();
                        CH7 = ON;
                        break;
                case '8':
                        timer_start();
                        CH8 = ON;
                        break;
                }
                //
                LED = ON;
                Delay_ms(10);
                LED = OFF;
        }
}
//*****************************************************************************
*
//■リモコンモジュールクライアント関数（送信用）
void    Remocon_Client()
{
        short   cnt;
        //
        TRISA  = 0b11111111;
        TRISB  = 0b11111100;
        //
        Pwm1_Init(38000);       // 38kHz duty=50%
        Pwm1_Set_Duty(PR2 / 2);
        Pwm1_Stop();
        //
        for (cnt = 0; cnt < 10; cnt++) {
                LED = ON;
                Delay_ms(50);
                LED = OFF;
                Delay_ms(50);
        }
        //
        while (1) {
                if (SW1 == 0) {
                        while (SW1 == 0) {
                                Delay_ms(100);
                        }
                        Remocon_Send_Chr('1');
                        led_on_off();
                }
                if (SW2 == 0) {
```

```
                        while (SW2 == 0) {
                                Delay_ms(100);
                        }
                        Remocon_Send_Chr('2');
                        led_on_off();
                }
                if (SW3 == 0) {
                        while (SW3 == 0) {
                                Delay_ms(100);
                        }
                        Remocon_Send_Chr('3');
                        led_on_off();
                }
                if (SW4 == 0) {
                        while (SW4 == 0) {
                                Delay_ms(100);
                        }
                        Remocon_Send_Chr('4');
                        led_on_off();
                }
                if (SW5 == 0) {
                        Remocon_Send_Chr('5');
                        led_on_off();
                }
                if (SW6 == 0) {
                        Remocon_Send_Chr('6');
                        led_on_off();
                }
                if (SW7 == 0) {
                        Remocon_Send_Chr('7');
                        led_on_off();
                }
                if (SW8 == 0) {
                        Remocon_Send_Chr('8');
                        led_on_off();
                }
        }
}
//*********************************************************************
*
//■リモコンモジュールデータ送信関数（クライアント側送信用）
void    Remocon_Send_Chr(unsigned short dt)
{
        short   cnt;
        //start-bit
        Pwm1_Start();
        Delay_us(833);
        Pwm1_Stop();
        //data-bit(LSB.....MSB)
        for (cnt = 0; cnt < 8; cnt++) {
                if ((dt & 0b00000001) != 0) {
```

```
                        Delay_us(833);
                } else {
                        Pwm1_Start();
                        Delay_us(833);
                        Pwm1_Stop();
                }
                //
                dt >>= 1;
        }
        //stop-bit
        Delay_us(833);
        //
        Delay_ms(10);
}
//*********************************************************************
*
//■割り込み関数
void    interrupt()
{
        if (PIR1.TMR1IF == 1) {
                PIR1.TMR1IF = 0;
                T1CON.TMR1ON = 0;
                //
                CH5 = OFF;
                CH6 = OFF;
                CH7 = OFF;
                CH8 = OFF;
        }
}
//*********************************************************************
*
//■タイマー開始関数(100msec)
void    timer_start()
{
        PIE1.TMR1IE = 1;
        PIR1.TMR1IF = 0;
        T1CON.T1CKPS0 = 1;
        T1CON.T1CKPS1 = 1;
        TMR1L = 0x58;
        TMR1H = 0x9E;
        INTCON.PEIE = 1;
        INTCON.GIE = 1;
        T1CON.TMR1ON = 1;
}
//*********************************************************************
*
//■□□□点滅関数
void    led_on_off()
{
        LED = ON;
        Delay_ms(10);
```

```
        LED = OFF;
}
//*****************************************************************************
*
```

動作確認

左側:リモコン(送信側)のスイッチが、押下されていないのでリモコン(受信側)のLEDは消灯しています。
右側:リモコン(送信側)のスイッチが、押下されると、リモコン(受信側)のLEDは点灯します。