

ワイヤレスLCD(赤外線データ通信)

概要

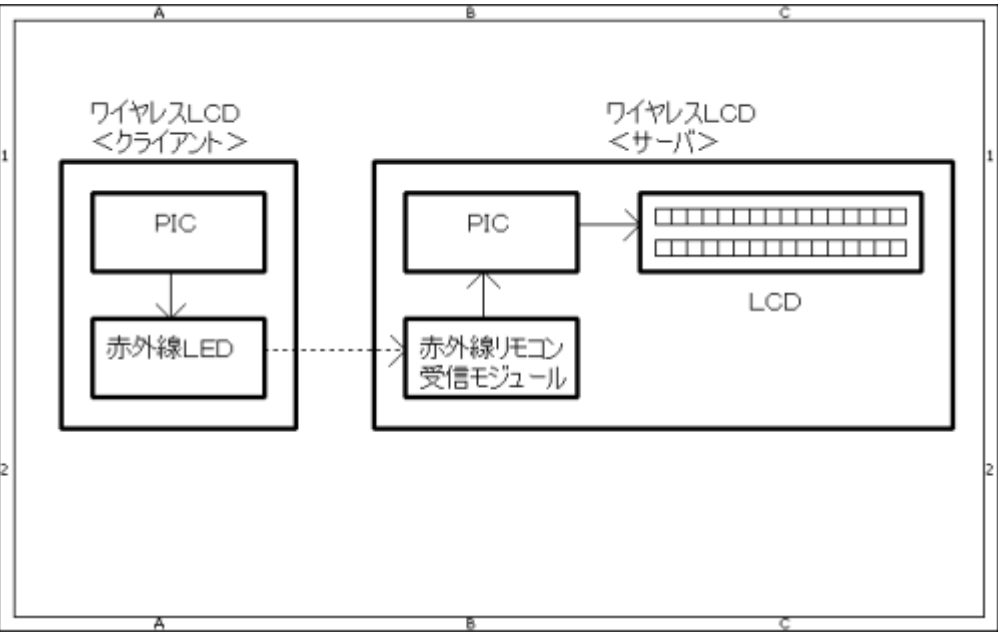
LCDは製作する装置に内蔵して利用するのが一般的です。しかし、離れた場所に設置するような装置では、常時LCDに表示する必要はありません。つまり、確認する必要が生じた時のみLCDを持って行き接続し、データを表示をさせれば良い事になります。そうすることにより、装置の小型化や低消費電力化を図ることが出来ます。

通常は、装置とLCDの接続は、コネクタ(ケーブル)で行いますが、今回は、赤外線を利用(ワイヤレス)して、非接触でデータ通信を行い、LCDにデータを表示させました。

非接触のメリットは、

- コネクタの抜き差しの手間が省ける。
- 雨天であっても、装置に近づけるだけでデータが表示できる。(つまり装置の蓋を開ける必要がない)

等が考えられます。



クライアント側に実装する、ユーザアプリケーションのために、「ワイヤレスLCD制御ライブラリ」を提供します。ライブラリの仕様は、mikroCコンパチブルとしました。従って、関数名さえ変更すれば、どちらのライブラリでも同じように動作させることが出来ます。

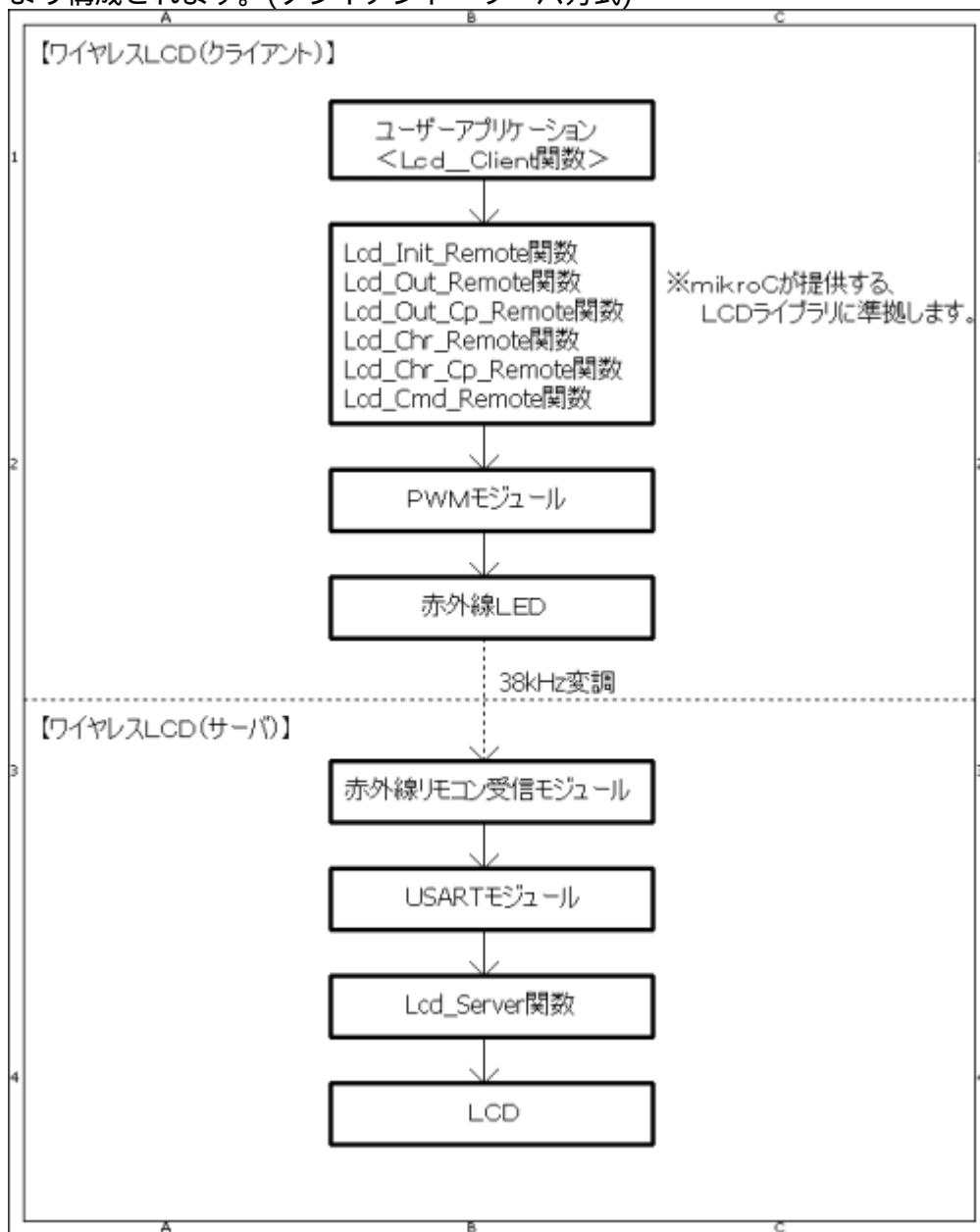
mikroC版LCD制御ライブラリ	ワイヤレスLCD制御ライブラリ <ユーザアプリケーションより呼び出します>
Lcd_Init	Lcd_Init_Remote
Lcd_Out	Lcd_Out_Remote
Lcd_Out_Cp	Lcd_Out_Cp_Remote
Lcd_Chr	Lcd_Chr_Remote
Lcd_Chr_Cp	Lcd_Chr_Cp_Remote
Lcd_Cmd	Lcd_Cmd_Remote

動作原理

ワイヤレスLCDは、

- LCDのサービス(機能)を提供する側【サーバ(Server)】
- LCDのサービスを利用する側【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)



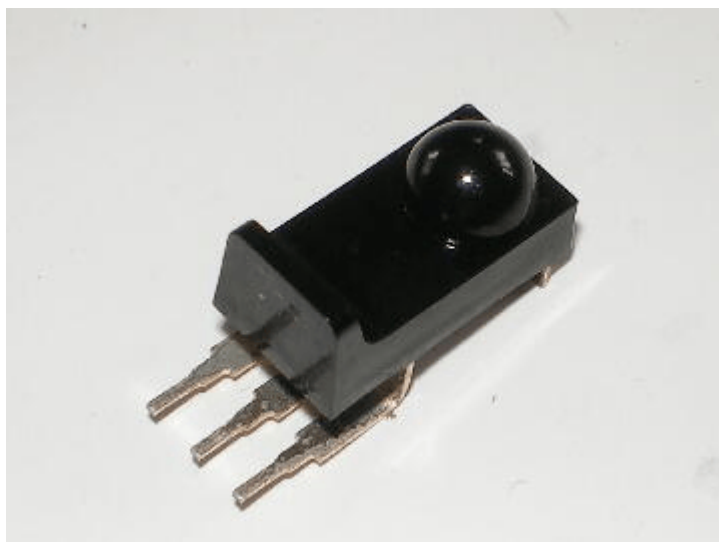
動作原理（ハードウェア）

赤外線を送信 高輝度赤外線LED(OSIR5113A)を使用します。



<赤外線LED(OSIR5113A)>

赤外線を受信テレビ・ビデオ等のリモコンに使用されている受光モジュール(SPS-443)を使用します。ピンフォトダイオード、アンプ、フィルター回路が内蔵されています。38kHzの副搬送波で振幅変調された赤外線のみを検出する事(ノイズ対策)が出来ます。



<赤外線リモコン受信モジュール(SPS-443)>

副搬送波(38kHz)の発生 PIC内蔵のCCPモジュールをPWMモードで使用し、38kHzを発生させます。

動作原理(ソフトウェア)

提供関数(クライアント側)

- Lcd_Init_Remote();
LCDを初期化します。
- Lcd_Out_Remote(char row, char column, char *text);
行(row)と列(column)を指定してLCDに文字列(text)を表示します。
- Lcd_Out_Cp_Remote(char *text);
カレントカーソルに、LCDに文字列(text)を表示します。
- Lcd_Chr_Remote(char row, char column, char out_char);
行(row)と列(column)を指定してLCDに文字(out_char)を表示します。
- Lcd_Chr_Cp_Remote(char out_char);
カレントカーソルに、LCDに文字(out_char)を表示します。

- Lcd_Cmd_Remote(char out_char);
LCDにコマンド(out_char)を送信します。

※mikroCコンパチブルにしましたので、詳細については、其方(HELP)を参照してください。 動作確認のため、クライアント側にはPIC16F88を使用しましたが、他のPICでも同様に動作します。

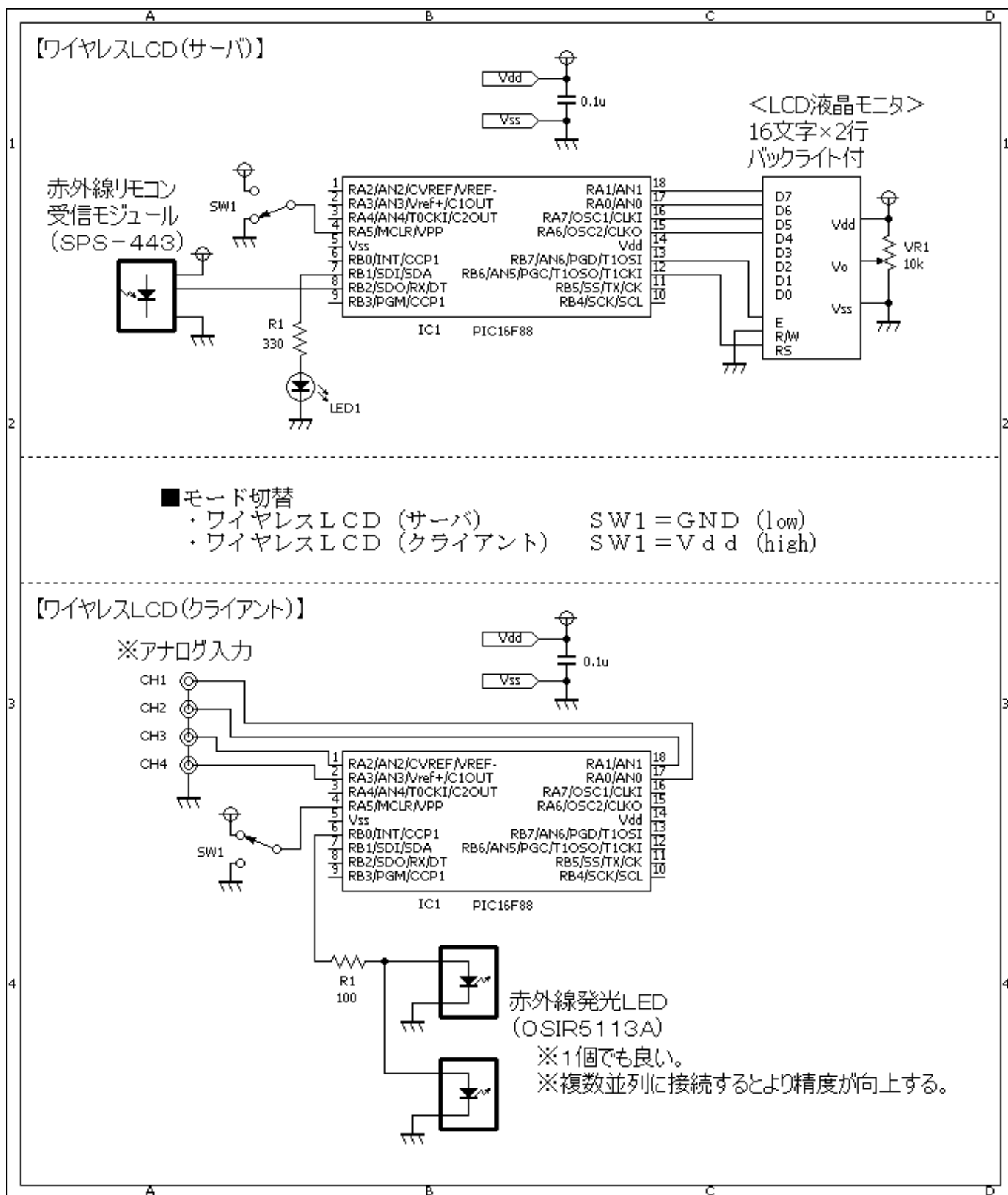
メイン処理 スイッチ(SW1)の設定により、サーバ側、クライアント側を切り替え、各処理を呼び出します
SW1=GNDサーバ側(Lcd_Server関数) SW1=Vddクライアント側(Lcd_Client関数)

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
A/D変換を行い、サーバ側に表示データを送信します。

ユーザアプリケーションは、このクライアント処理を修正して、実装します。

サーバ処理 LCDおよびUSART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、LCDを制御します。

回路図



ソースコード

[remote_lcd.c](#)

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

```

<ワイヤレス赤外線データ通信)>
*/
//*****
*
// ■ マクロ定義
#define BYTE      unsigned short
#define WORD      unsigned int
#define DWORD     unsigned long
//
sbit SW          at RA5_bit;
#define EOF      0x00
//
sbit LED         at RB1_bit;
#define ON        1
#define OFF       0
//LCD
sbit LCD_RS at RB6_bit;
sbit LCD_EN at RB7_bit;
sbit LCD_D7 at RA1_bit;
sbit LCD_D6 at RA0_bit;
sbit LCD_D5 at RA7_bit;
sbit LCD_D4 at RA6_bit;
sbit LCD_RS_Direction at TRISB6_bit;
sbit LCD_EN_Direction at TRISB7_bit;
sbit LCD_D7_Direction at TRISA1_bit;
sbit LCD_D6_Direction at TRISA0_bit;
sbit LCD_D5_Direction at TRISA7_bit;
sbit LCD_D4_Direction at TRISA6_bit;
//*****
*
// ■ 関数宣言
extern void      main();
extern void      Lcd_Server();
extern void      Lcd_Client();
extern void      Lcd_Init_Remote();
extern void      Lcd_Out_Remote(char row, char column, char *text);
extern void      Lcd_Out_Cp_Remote(char *text);
extern void      Lcd_Chr_Remote(char row, char column, char out_char);
extern void      Lcd_Chr_Cp_Remote(char out_char);
extern void      Lcd_Cmd_Remote(char out_char);
extern void      Lcd_Send_Chr(unsigned short dt);
extern void      Lcd_Send_Str(unsigned short* pData);
//*****
*
// ■ メイン関数
void      main()
{
    OSCCON = 0b01110000;
    TRISA  = 0b11111111;
    TRISB  = 0b00001100;
    //

```

```
    LED = OFF;
    //
    if (SW == 1) {
        Lcd_Client();
    } else {
        Lcd_Server();
    }
}
//*****
*
// ■ ワイヤレス  サーバ関数
void    Lcd_Server()
{
    char    dt, cnt, buf[32];
    //
    ANSEL   = 0b00000000;
    //
    Lcd_Init();
    Lcd_Cmd(_LCD_CURSOR_OFF);
    Lcd_Cmd(_LCD_CLEAR);
    for (cnt = 0; cnt < 16; cnt++) {
        Lcd_Chrc(1, cnt + 1, 0xFF);
        Delay_ms(50);
    }
    for (cnt = 0; cnt < 16; cnt++) {
        Lcd_Chrc(2, cnt + 1, 0xFF);
        Delay_ms(50);
    }
    Delay_ms(1000);
    Lcd_Cmd(_LCD_CLEAR);
    //
    Uart1_Init(1200);
    //
    while(1) {
        cnt = 0;
        while (1) {
            if (Uart1_Data_Ready() != 1) {
                continue;
            }
            //
            dt = Uart1_Read();
            if (dt != EOF) {
                buf[cnt] = dt;
                cnt++;
            } else {
                buf[cnt] = 0x00;
                LED = ~LED;
                break;
            }
            //
            if (cnt == 32) {
```

```
                cnt = 0;
                continue;
            }
        }
    }
    //
    switch (buf[0]) {
    case 'A':
        Lcd_Out(buf[1], buf[2], &buf[3]);
        break;
    case 'B':
        Lcd_Out_Cp(&buf[1]);
        break;
    case 'C':
        Lcd_Chrc(buf[1], buf[2], buf[3]);
        break;
    case 'D':
        Lcd_Chrcp(buf[1]);
        break;
    case 'E':
        Lcd_Cmd(buf[1]);
        break;
    }
}

//*****
*
// ■ ワイヤレスクライアント関数
void Lcd_Client()
{
    char    buf[16];
    WORD    ad;
    //
    ANSEL   = 0b00001111;
    //
    Lcd_Init_Remote();
    //
    while (1) {
        ad = Adc_Read(0);
        ad = ad * 4.8828125;
        WordToStr(ad, buf);
        Lcd_Out_Remote(1, 1, buf);
        Lcd_Out_Cp_Remote("mV");
        //
        ad = Adc_Read(1);
        ad = ad * 4.8828125;
        WordToStr(ad, buf);
        Lcd_Out_Remote(1, 9, buf);
        Lcd_Out_Cp_Remote("mV");
        //
        ad = Adc_Read(2);
        ad = ad * 4.8828125;
```

```

        WordToStr(ad, buf);
        Lcd_Out_Remote(2, 1, buf);
        Lcd_Out_Cp_Remote("mV");
        //
        ad = Adc_Read(3);
        ad = ad * 4.8828125;
        WordToStr(ad, buf);
        Lcd_Out_Remote(2, 9, buf);
        Lcd_Out_Cp_Remote("mV");
        //
        Delay_ms(100);
    }
}

//*****
*
// ■ ワイヤレス初期化関数
void    Lcd_Init_Remote()
{
    Pwm1_Init(38000);          // 38kHz duty=50%
    Pwm1_Set_Duty(PR2 / 2);
    Pwm1_Stop();
}

//*****
*
// ■ ワイヤレス文字列出力(行列指定)関数
void    Lcd_Out_Remote(char row, char column, char *text)
{
    Lcd_Send_Chr('A');
    Lcd_Send_Chr(row);
    Lcd_Send_Chr(column);
    Lcd_Send_Str(text);
    Lcd_Send_Chr(E0F);
}

//*****
*
// ■ ワイヤレス文字列出力(カレント位置)関数
void    Lcd_Out_Cp_Remote(char *text)
{
    Lcd_Send_Chr('B');
    Lcd_Send_Str(text);
    Lcd_Send_Chr(E0F);
}

//*****
*
// ■ ワイヤレス文字出力(行列指定)関数
void    Lcd_Chr_Remote(char row, char column, char out_char)
{
    Lcd_Send_Chr('C');
    Lcd_Send_Chr(row);
    Lcd_Send_Chr(column);
    Lcd_Send_Chr(out_char);
}

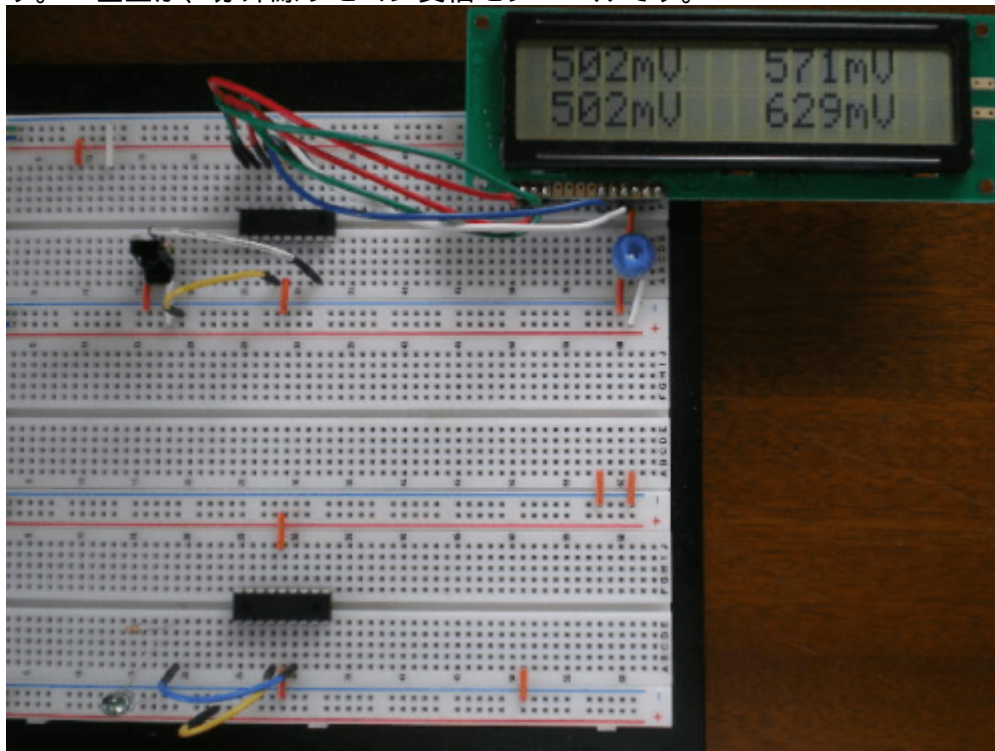
```

```
        Lcd_Send_Chr(E0F);
    }
    //*****
    *
    // ■ ワイヤレス文字出力 (カレント位置) 関数
    void    Lcd_Chr_Cp_Remote(char out_char)
    {
        Lcd_Send_Chr('D');
        Lcd_Send_Chr(out_char);
        Lcd_Send_Chr(E0F);
    }
    //*****
    *
    // ■ ワイヤレスコマンド出力関数
    void    Lcd_Cmd_Remote(char out_char)
    {
        Lcd_Send_Chr('E');
        Lcd_Send_Chr(out_char);
        Lcd_Send_Chr(E0F);
    }
    //*****
    *
    // ■ ワイヤレスデータ送信 (1文字) 関数
    void    Lcd_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);
    }
    //*****
    *
    // ■ ワイヤレスデータ送信 (文字列) 関数
    void    Lcd_Send_Str(unsigned short* pData)
```

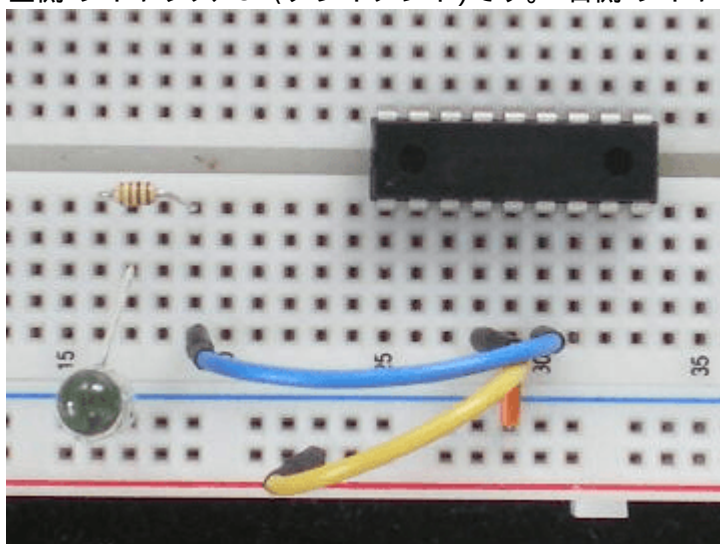
```
{  
    while (*pData != 0x00) {  
        Lcd_Send_Chr(*pData);  
        pData++;  
    }  
}  
//*****  
*
```

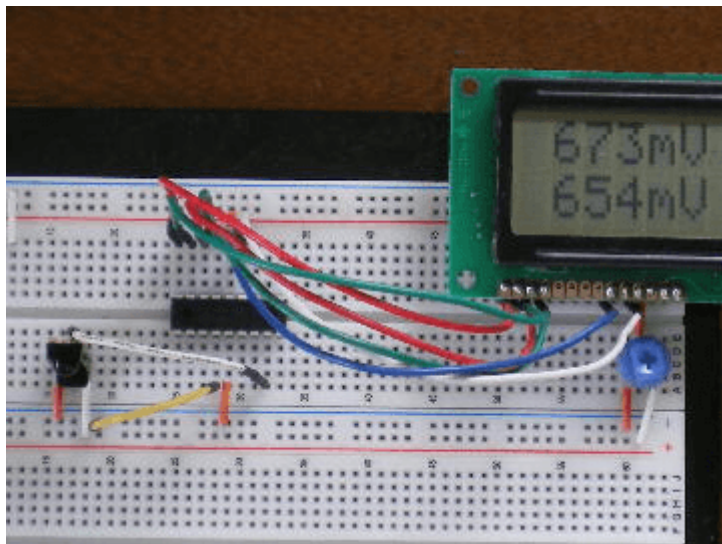
動作確認

下側:ワイヤレスLCD(クライアント)です。 左下が、赤外線LEDです。 上側:ワイヤレスLCD(サーバ)です。 左上が、赤外線リモコン受信モジュールです。

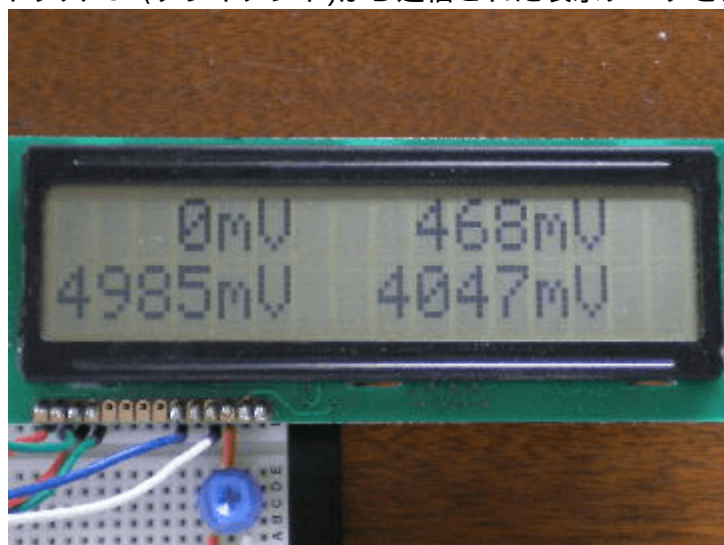


左側:ワイヤレスLCD(クライアント)です。 右側:ワイヤレスLCD(サーバ)です。

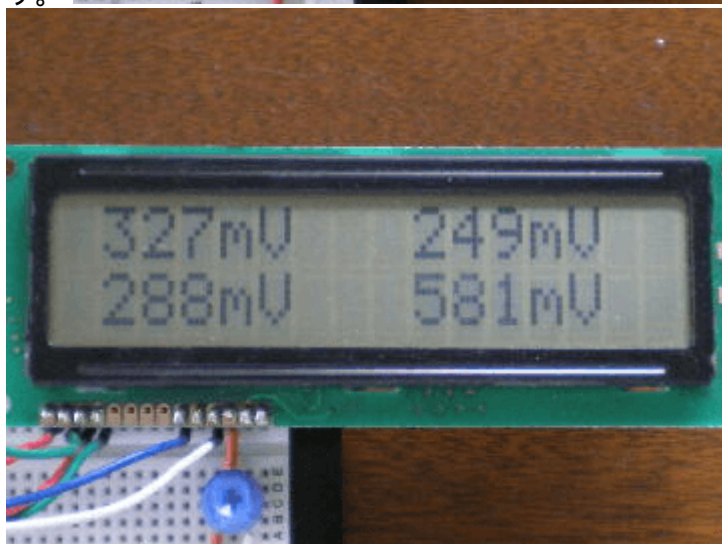




ワイヤレスLCD(クライアント)から送信された表示データを、ワイヤレスLCD(サーバ)でLCDに表示しま



す。



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://deepsky.jp/wiki/> - うごくといいな

Permanent link:

<http://deepsky.jp/wiki/doku.php?id=elechobby:picdic:pic16f88:144>

Last update: **2020/05/01 18:11**

