

簡易遠隔操作ユニット(携帯電話版)

概要

必要に迫られて、簡単な仕組みで、且つ、費用が余りかからない「簡易遠隔操作ユニット」を製作しました。

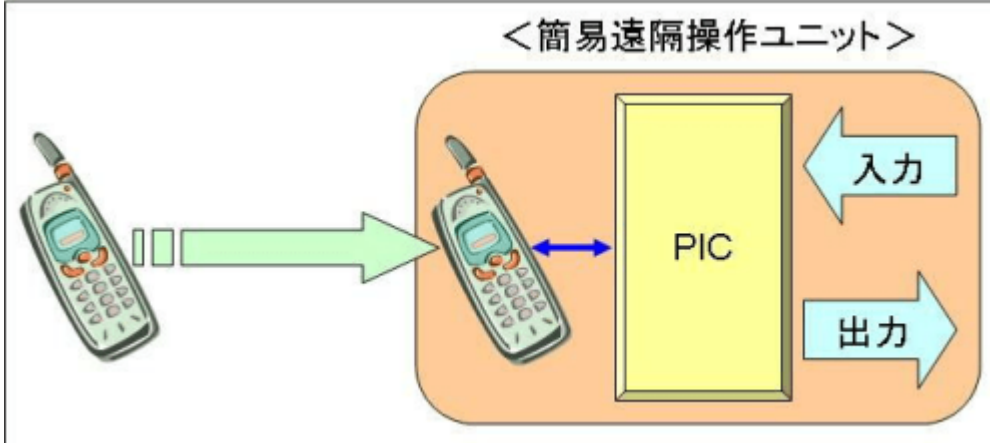
<本ユニットへの要望事項>

- 操作が簡単であること。
- 入出力を制御できること。
- 通信費用が安いこと。
- 近くに商用電源(AC100V)やネットワーク(LAN)が無くても使用できること。
- 仕組みが簡単で、構築に時間がかからないこと。
- 入手が容易な部品を使用すること。
- 持ち運びが容易であること。

これらの要望をある程度満足させられる方法として、以下の理由で携帯電話を採用しました。

- 携帯電話用接続コード(平型プラグ)を使用することにより、容易に発着信を制御できる。
- ファミリー割引を適用することにより、通話料が無料となる。
- 利用可能範囲が広い。いつでも、どこでも通話 通信ができる。

<簡易遠隔操作ユニットのイメージ図> 遠隔操作する携帯電話(左側)は、固定電話でも利用できますが、通信費用が高くなってしまいます。



<仕様>

- 入力は、3チャンネルのデジタル(PICのポート)とします。
- 出力は、3チャンネルのデジタル(PICのポート)とします。
- 遠隔からの操作は、携帯電話のピポパ音(DTMF: Dual-Tone Multi-Frequency)を使用します。

- 「1」 入力チャンネル(1)の状態を音で知らせます。(ハイ状態=2000Hz □ロー状態=500Hz)
- 「2」 入力チャンネル(2)の状態を音で知らせます。(ハイ状態=2000Hz □ロー状態=500Hz)
- 「3」 入力チャンネル(3)の状態を音で知らせます。(ハイ状態=2000Hz □ロー状態=500Hz)
- 「4」 出力チャンネル(1)をオンにします。(完了状態=1000Hz)
- 「5」 出力チャンネル(1)をオフにします。(完了状態=1000Hz)
- 「6」 出力チャンネル(2)をオンにします。(完了状態=1000Hz)
- 「7」 出力チャンネル(2)をオフにします。(完了状態=1000Hz)

「8」 出力チャンネル(3)をオンにします。(完了状態=1000Hz)
「9」 出力チャンネル(3)をオフにします。(完了状態=1000Hz)
「0」 通話を終了します。

- 自動応答モード
着信をトリガとして、遠隔操作処理を行います。
- 自動通知モード
入力チャンネル(1)が、ロー(ON)状態になると、自動発信を行い、遠隔操作処理を行います。

動作原理(ハードウェア)

携帯電話との接続 携帯電話には、ヘッドセット(イヤホンマイク)を接続するための平型端子が用意されています。ヘッドセットを接続することにより、ハンズフリーで運転中も安心して通話することが出



来ます。

今回使用する、ヘッドセットは、発着信がワンタッチ操作で行える□ON/OFFスイッチ付きのものを使用

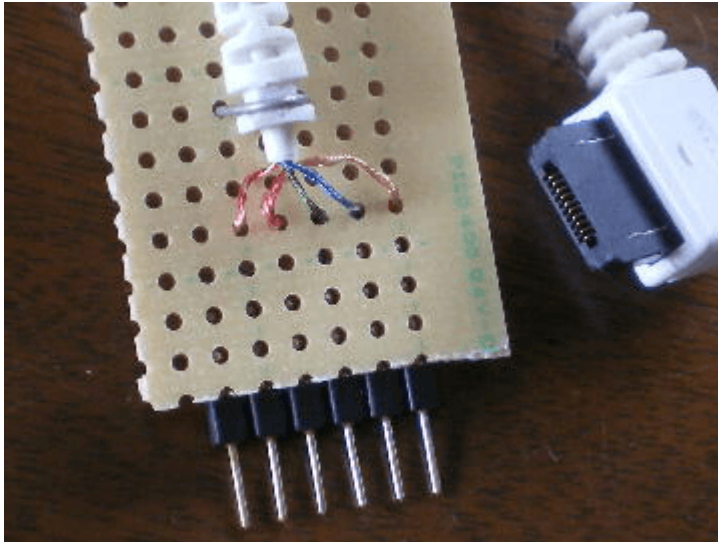


します。

500円前後で販売されています。

ヘッドセットに付いている、マイク、スイッチ、イヤホンを取り外します。そして、ブレッドボードに接続し易いようにユニバーサル基板で治具を作ります。





平型端子のピンアサイン(ヘッドセット接続時)は、次のようになっています。

- 【01】 アナログ(GND)
- 【02】 送信信号(マイク入力)
- 03HS検出(ヘッドセット有無)有り=Low無し=オープン
- 【04】 受信信号(ヘッドホン出力:R)モノラル時に出力されない。
- 【05】 受信信号(ヘッドホン出力:L)モノラル時に出力される。
- 【06】 ステレオ/モノラル切り替えステレオ=Lowモノラル=オープン
- 【07】 予約
- 【08】 通話スイッチ(発信、着信、終了)
- 【09】 予約
- 【10】 デジタルGND

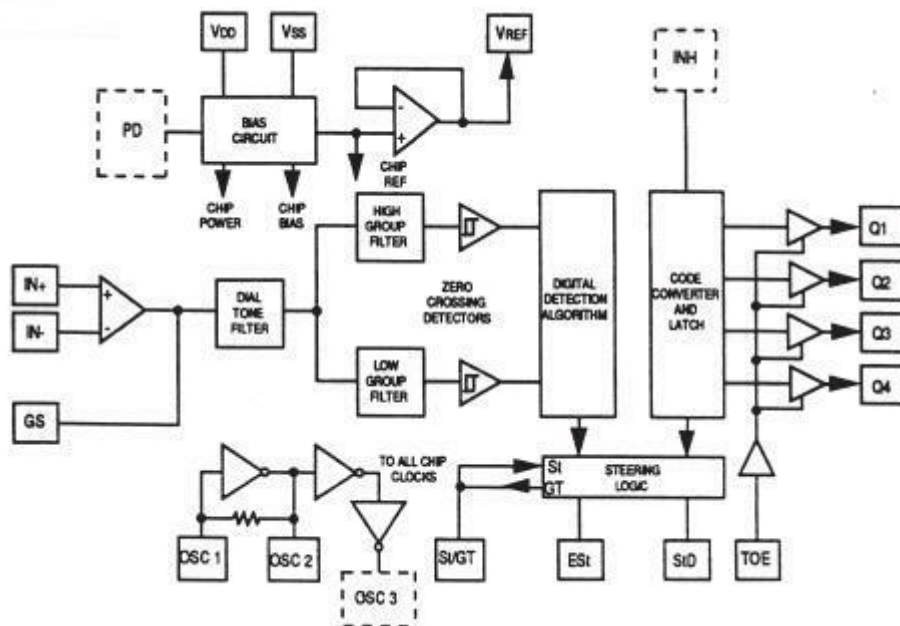
マイクの入力インピーダンスは、1k Ω 以上 ヘッドホンの負荷インピーダンスは、16 Ω 以上

通話スイッチの操作が重要となるので、少し詳しく説明します。

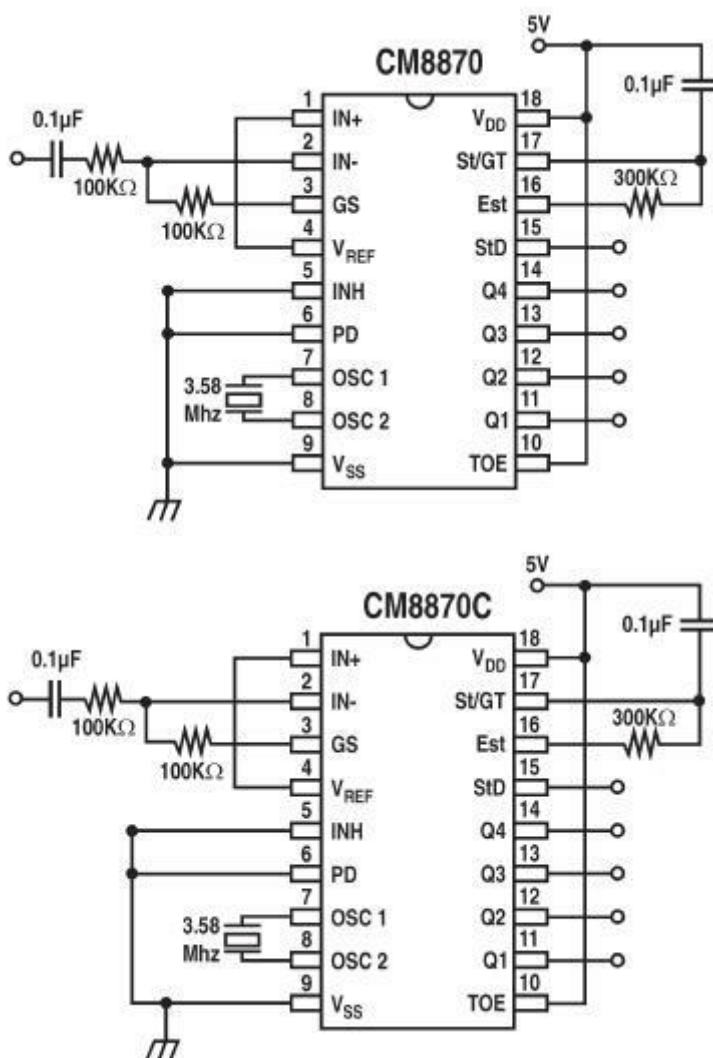
- 電話を受ける。
着信中に、通話スイッチを押下(約1秒)すると、通話が始まります。
- 電話をかける。
待受中に、通話スイッチを押下(約2秒)すると、メモリNo.999に登録された番号に電話がかかります。
- 電話を切る。
通話中に、通話スイッチを押下(約2秒)すると、電話が切れます。

これらの仕様は、機種によって若干異なります。詳細は各機種の取扱説明書をご覧ください。

©DTMFレシーバ 携帯電話からのDTMF信号を受けて、4ビットのデジタル信号に変換するDTMFレシーバにはCM8870を使用しました (California Micro Devices Corp)



<ブロックダイアグラム>



<片線接地(シングルエンド)での使用例>

着信の検出 着信音をLM386を使用して20倍に増幅し、ダイオードで倍電圧整流し、着信電圧に変換します。

通話スイッチの制御 トランジスタのコレクタを使用して、通話スイッチをオープン状態(オフ)またはGNDに接地状態(オン)とします。

動作原理(ソフトウェア)

自動応答モード着信電圧を検出して $\geq 300\text{mV}$ 以上であれば、着信とみなします。着信とみなした場合、通話スイッチを1秒間オンにし、通話を開始します。

自動通知モード待受中に、入力チャンネル(1)が、オン状態(ロー)になると、通話スイッチを2秒間オンにし、通話を開始します。

遠隔操作処理

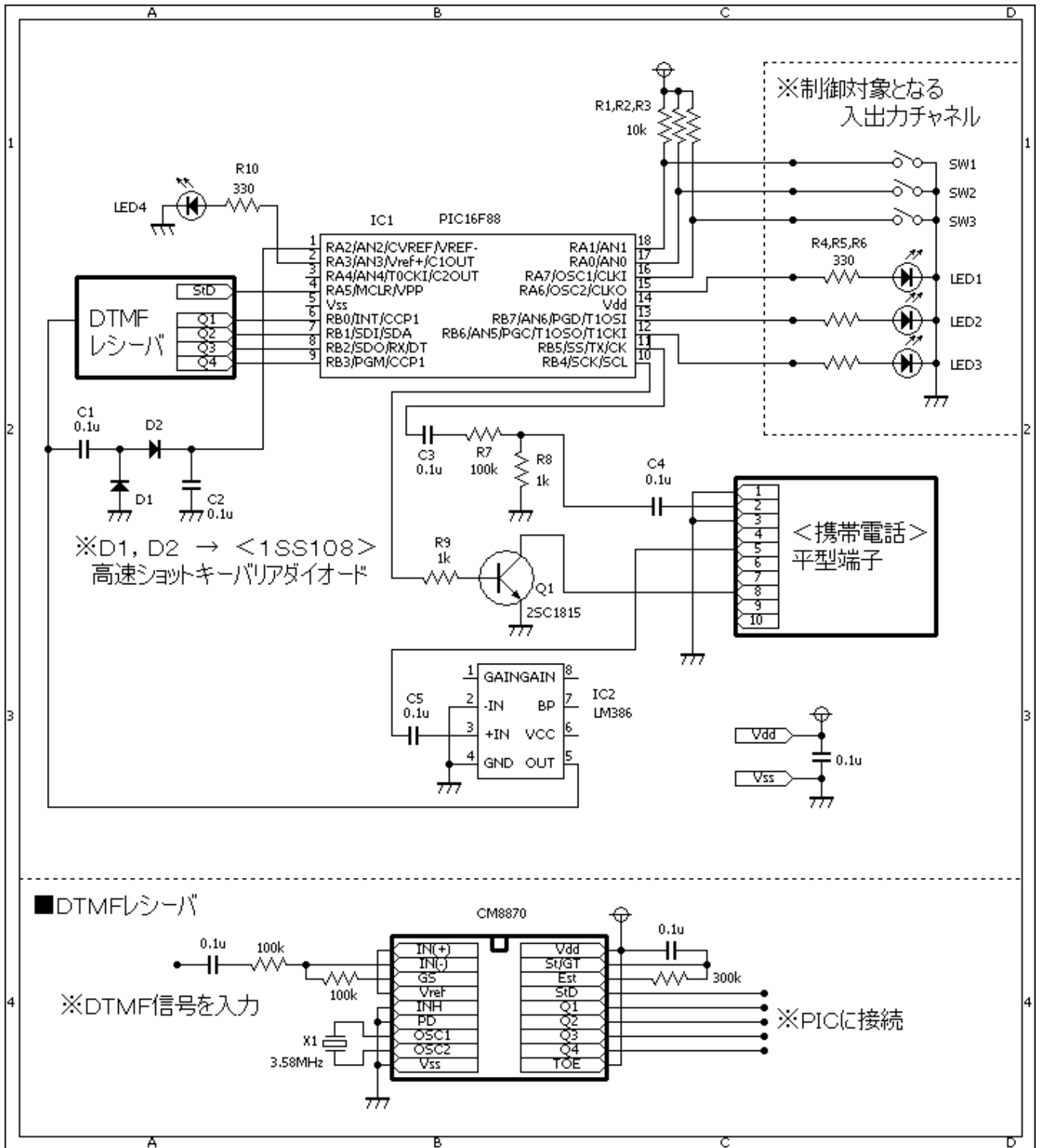
1. 開始を知らせる音「ドミソド」を鳴らせます。
2. 携帯電話のキー(DTMF信号)が、押されるのを待ちます。
3. 押されたキーに対応した処理を行います。

- 「1」 入力チャンネル(1)の状態を音で知らせます。(ハイ状態=2000Hz□ロー状態=500Hz)
- 「2」 入力チャンネル(2)の状態を音で知らせます。(ハイ状態=2000Hz□ロー状態=500Hz)
- 「3」 入力チャンネル(3)の状態を音で知らせます。(ハイ状態=2000Hz□ロー状態=500Hz)
- 「4」 出力チャンネル(1)をオンにします。(応答音=1000Hz)
- 「5」 出力チャンネル(1)をオフにします。(応答音=1000Hz)
- 「6」 出力チャンネル(2)をオンにします。(応答音=1000Hz)
- 「7」 出力チャンネル(2)をオフにします。(応答音=1000Hz)
- 「8」 出力チャンネル(3)をオンにします。(応答音=1000Hz)
- 「9」 出力チャンネル(3)をオフにします。(応答音=1000Hz)
- 「0」 通話を終了します。終了を知らせる音「ドソミド」を鳴らせます。

※DTMF信号は、携帯電話の通信状態によっては、認識率が低下します。従って、応答音がない場合には、再度キーを押してください。

サウンド処理 maikorC提供のサウンドライブラリ(ソフト処理)を使用します。

回路図



ソースコード

[cellphone_remote_control_v2.c](#)

```
//*****
*
/*
『遠隔操作(携帯電話使用)』
*/
```

```
//*****
*

#define      PHONE          PORTB.B4
#define      PHONE_ON      1
#define      PHONE_OFF     0

#define      BYTE          unsigned char
#define      WORD          unsigned int

#define      DELAY         100

#define      SW1           PORTA.B1
#define      SW2           PORTA.B0
#define      SW3           PORTA.B7
#define      SW_ON         0
#define      SW_OFF        1

#define      LED4          PORTA.B3
#define      LED1          PORTA.B6
#define      LED2          PORTB.B7
#define      LED3          PORTB.B6
#define      LED_ON        1
#define      LED_OFF       0

#define      DTMF_STD      PORTA.B5

//*****
*

void do_mi_so_do()
{
    Sound_Play(523, 250);      //ド (ピン)
    Sound_Play(659, 250);      //ミ (ボン)
    Sound_Play(784, 250);      //ソ (パン)
    Sound_Play(1047, 500);     //ド (ポーン)
}

//*****
*

void do_so_mi_do()
{
    Sound_Play(1047, 250);     //ド (ピン)
    Sound_Play(784, 250);      //ソ (ボン)
    Sound_Play(659, 250);      //ミ (パン)
    Sound_Play(523, 500);      //ド (ポーン)
}

//*****
```

```
*  
  
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-...",  
    ".-..."  
};  
  
void short_beep()  
{  
    Sound_Play(800, DELAY);  
    Delay_ms(DELAY);  
}  
  
void long_beep()  
{  
    Sound_Play(800, DELAY * 3);  
}
```

```
        Delay_ms(DELAY);
    }

    void morse_chr(char chr)
    {
        static      short      cnt, i;
        //
        if (chr == ' ') {
            Delay_ms(DELAY * 3);
            return;
        }
        //
        for (cnt = 0; cnt < 37; 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(DELAY * 3);
                return;
            }
        }
    }

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

    //*****
    *

    int ring_tone()
    {
        WORD      ad;
        short     cnt;
        //
        ad = 0;
        for (cnt = 0; cnt < 50; cnt++) {
            ad += ADC_Get_Sample(2);
        }
        return (ad / 50);
    }

    //*****
```

```
*  
  
void proc()  
{  
    short key;  
    //通知開始“ドミソド”を知らせます。  
    Delay_ms(500);  
    do_mi_so_do();  
    Delay_ms(500);  
    //  
    while (1) {  
        //キ[ ]信号)が押されるのを待ちます。  
        while (DTMF_STD == 0) {  
        }  
        while (DTMF_STD == 1) {  
        }  
        key = PORTB & 0x0F;  
        //  
        switch (key) {  
        case 1:  
            if (SW1 == 1) {  
                Sound_Play(2000, 500);  
            } else {  
                Sound_Play(500, 500);  
            }  
            break;  
        case 2:  
            if (SW2 == 1) {  
                Sound_Play(2000, 500);  
            } else {  
                Sound_Play(500, 500);  
            }  
            break;  
        case 3:  
            if (SW3 == 1) {  
                Sound_Play(2000, 500);  
            } else {  
                Sound_Play(500, 500);  
            }  
            break;  
        case 4:  
            LED1 = LED_ON;  
            Sound_Play(1000, 500);  
            break;  
        case 5:  
            LED1 = LED_OFF;  
            Sound_Play(1000, 500);  
            break;  
        case 6:  
            LED2 = LED_ON;  
            Sound_Play(1000, 500);  
        }  
    }  
}
```

```
        break;
    case 7:
        LED2 = LED_OFF;
        Sound_Play(1000, 500);
        break;
    case 8:
        LED3 = LED_ON;
        Sound_Play(1000, 500);
        break;
    case 9:
        LED3 = LED_OFF;
        Sound_Play(1000, 500);
        break;
    case 10:
        //通知終了“ドソミド”を知らせます。
        Delay_ms(500);
        do_so_mi_do();
        Delay_ms(500);
        return;
        break;
    }
}

//*****
*

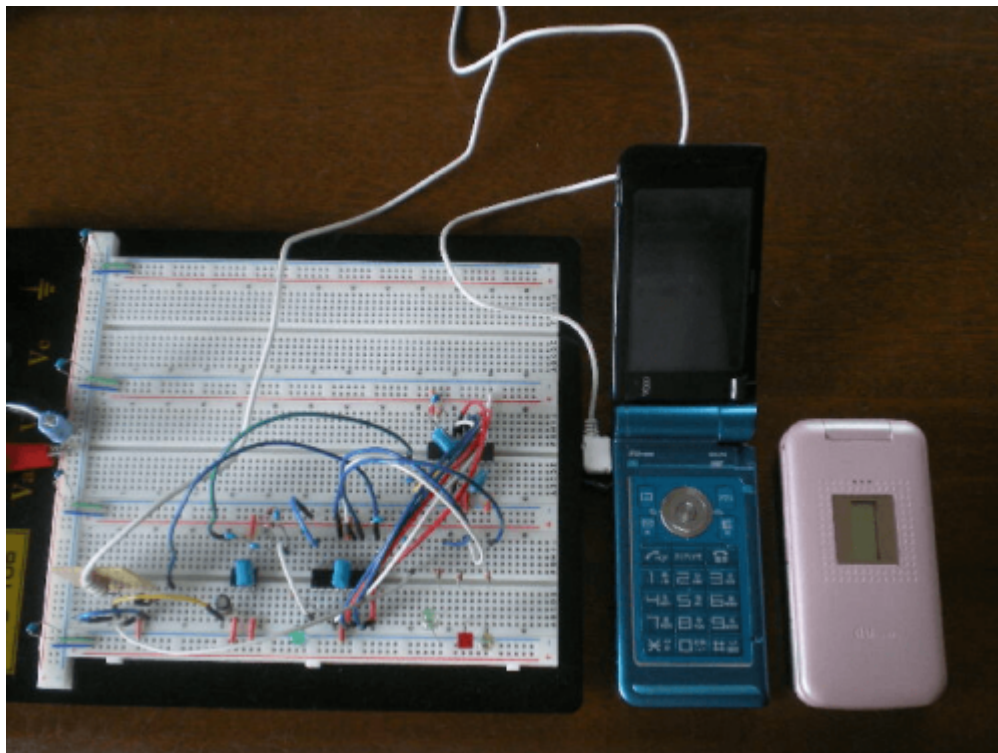
void main()
{
    double ad;
    short cnt;
    //
    OSCCON = 0b01110000; // クロックは8Mhz
    CMCON = 0b00000111; // コンパレータは使用しない。
    ANSEL = 0b00000100; // □□□変換を使用する。
    TRISA = 0b10110111;
    TRISB = 0b00001111;
    PHONE = PHONE_OFF;
    LED4 = LED_OFF;
    //
    for (cnt = 0; cnt < 10; cnt++) {
        LED1 = LED_ON;
        LED2 = LED_ON;
        LED3 = LED_ON;
        Delay_ms(100);
        LED1 = LED_OFF;
        LED2 = LED_OFF;
        LED3 = LED_OFF;
        Delay_ms(100);
    }
}
```

```
//
ADC_Init();
Sound_Init(&PORTB, 5);
Delay_ms(1000);
//
while (1) {
    while (1) {
        //自動応答：着信をチェックします。
        ad = ring_tone();
        ad *= 4.8828125;
        if (ad > 300.0) { //電圧が300mV以上で自動着信します。
            //着信を承諾(受付)します。
            PHONE = PHONE_ON;
            Delay_ms(1000);
            PHONE = PHONE_OFF;
            break;
        }
        //自動通知□□□□の押下で自動通知します。
        if (SW1 == 0) {
            //自動発信します。
            PHONE = PHONE_ON;
            Delay_ms(2000);
            PHONE = PHONE_OFF;
            break;
        }
    }
    //自動応答または自動通知の処理を行います。
    LED4 = LED_ON;
    proc();
    LED4 = LED_OFF;
    //通話を終了します。
    PHONE = PHONE_ON;
    Delay_ms(2000);
    PHONE = PHONE_OFF;
    //暫らく待機します。
    Delay_ms(10000);
}
} //~!

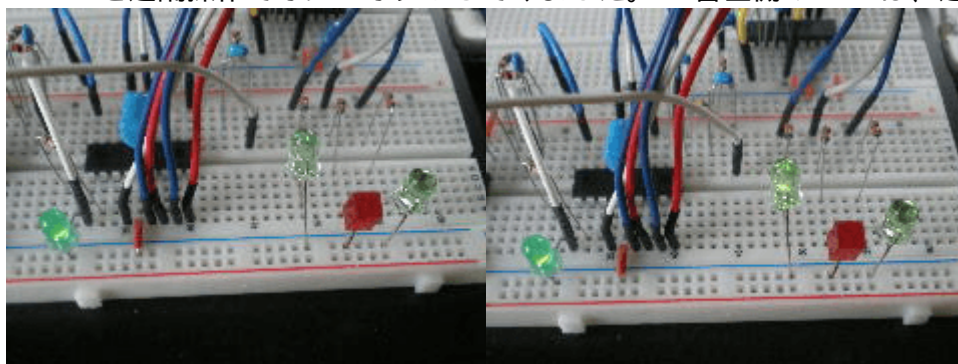
//*****
*
```

動作確認

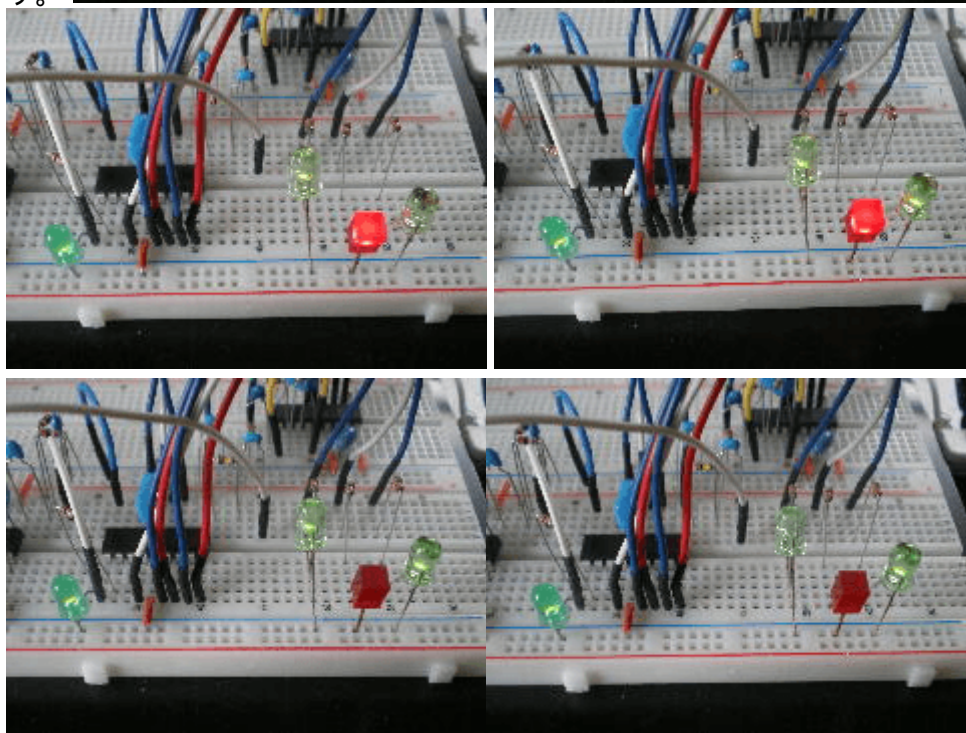
右側の携帯で遠隔操作します。



LED1~LED3を遠隔操作でオン・オフさせてみました。一番左側のLED4は、通話中はオンになっています。



す。



入力スイッチ(SW1□SW2□SW3)の操作の画像はありませんが、携帯電話の「1」「2」「3」を押すことにより、受話器から、スイッチの状態に応じた音(ハイ状態(オフ)=2000Hz□ロー状態(オン)=500Hz)が聞こえてきます。

如何ですか? これを応用することによって、遠隔から様々なコントロールが出来ますね!{😊}!

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:pic16f88:142&rev=1588225266>

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

