

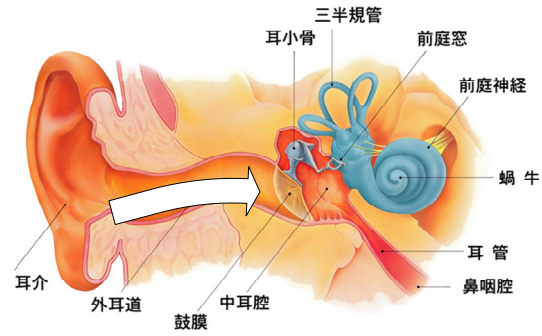
深部体温計の製作

ワイマチック 山田
www.ymatic.co.jp

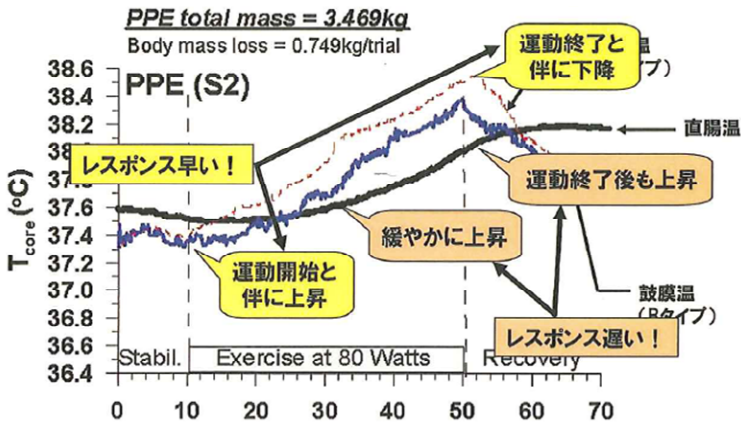
■深部体温とは体の中枢の温度の事で、ヒトは脳の温度を維持するため、高度な温度制御システムで維持されている。体内深部の温度の変化は重要な指標で、特にスポーツでの熱中症の予防、不眠症の治療での一日の体温のリズムや、低体温症での体内の免疫力の状態等、応用範囲は広い。

しかしながら、深部体温は、従来は肛門に熱電対を入れるなど、精神的苦痛があり、簡単には測定できなかつた。そこで放射温度センサーを使った、耳で深部体温を測定できる温度計を製作した。

鼓膜温は視床下部の体温中枢の近辺の温度を測定するため、レスポンスが早く、より深部体温に近い。



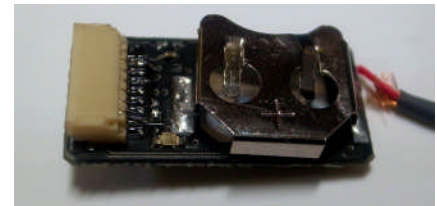
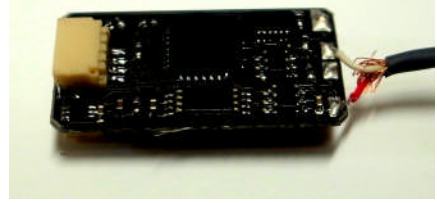
従来の直腸温と鼓膜温の比較



日本原子力研究開発機構 (JAEA) 学会発表資料より



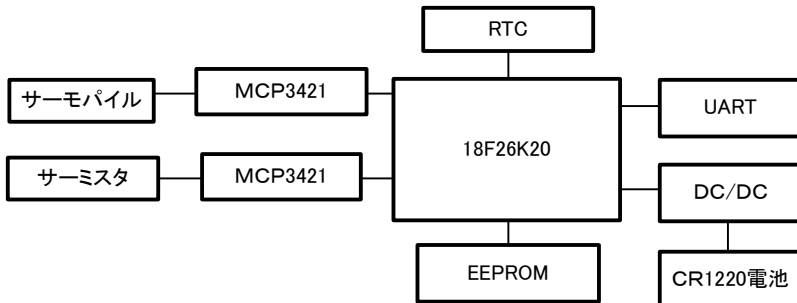
深部体温計基板



深部体温計の構成

サーモパイルと補償温度用のサーミスタが入った、石塚電子製のサーモパイル素子をMCP3421のシグマデルタ型のAD変換機でダイレクトにデジタル化し、I2CでCPUに入力し、データをEEPROMに記録する。MCP3421は18ビットでそれだけで15μVの分解能があり、8倍のアンプで2μVまで分解能が上げられる。スピードの遅い、熱電対や歪ゲージ等はOPアンプなしでダイレクトに接続できる便利なADである。

RTCの割り込みで駆動し、それ以外の時間はSleepコントロールしているのと電池をDC/DCで3.3Vに昇圧しているので12mm径のCR1220でも2,3日は電池が持つ。



深部体温計につけたI2C液晶画面



液晶をつけると、自分で深部体温をモニターできる。

自分の体温をダレクトにリアルタイムで見ることが出来る体温計は不思議と世の中にはない。

上は日時、下は左が記録回数、右が深部体温である。

WINDOWSアプリケーションの画面

USBで接続し、温度のモニターやデータのダウンロード等ができる。

スタートレコード

記録を開始します。このボタンを押すと過去のデータはリセットされます。

表示幅

グラフの表示幅を指定します。

クロックセット

内部時計を合わせます。

ストップレコード

記録を終了します。

ダウンロードを開始します。

自動的に日付からファイル名がつけられたCSVファイルがプログラムと同じホルダーにできます。

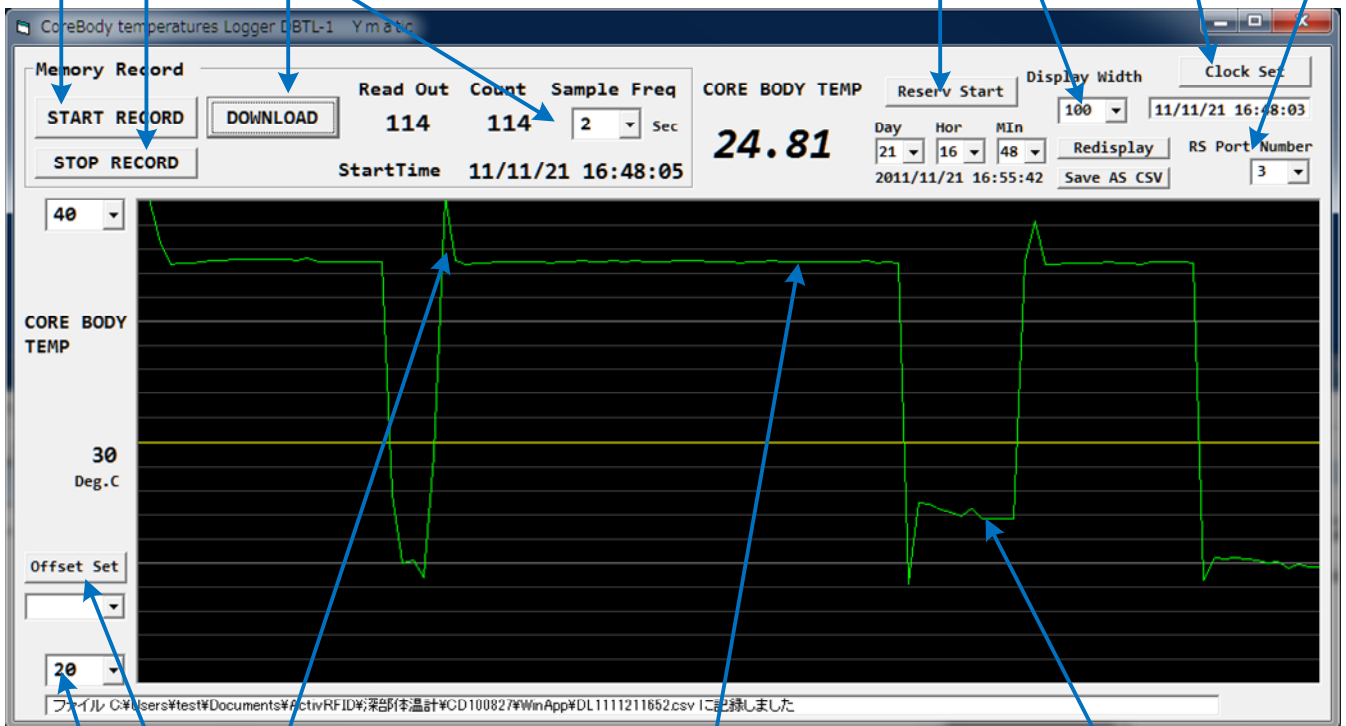
USBで接続している現在のデータをダウンロードファイルとは別のCSVファイルに記録します。

記録間隔を設定します。

予約スタート

日,時間,分を指定し、時間開始予約を設定します。予約中はLEDが点滅します。過去の時間の予約はできません。

USBのポート番号の設定。



スケール変更

耳に挿入

定常状態

耳から外す

オフセット設定

オフセットは内部で計算する数値で、キャリブレーション時に調整します。深部体温の絶対的な値は鼓膜に温度センサーを直接つけるしかなく、これは外科的な手術が必要です。人により、外耳道の形や体質で同じ深部体温でも違いがあります。一般的に脇下で測定する皮膚温のプラス1℃が深部体温と言われてます。まず、皮膚温を測り、深部体温がその値をプラス1した値になるようにオフセット値を調整してください。皮膚温は周囲環境温度により大きく変化しますので、室温環境で測定してください。

グラフは当初は室温を表示しており、外耳道に挿入すると、当初零接点温度測定用サーミスタの温度が低いので急激に温度が上がります。そのうち、サーミスタが外耳道の温度になるにつれて、深部体温が定常状態に近くなります。

グラフ上をポイントすると、ポイント点の時間と深部体温が表示されます。

ストロベリーリナックスで販売している16文字液晶ライブラリー

この液晶はわずか4本で、3Vで駆動できるので便利な液晶であるが、サンプルソースがAVRしかなく、マニュアルもわかりづらい。そこで、PICで動作するライブラリーを作った。CCSのコンパイラで作成した後関さんの液晶表示器制御ライブラリーコンパチである。

コントラストを調整する場合はI2Cでコントラストレジスターに書き込む。

CCSによる液晶表示ライブラリーソース

```
// I2CLCD TEST

#include <18F25K20.h>
#include INTRCNOWDT,NOPROTECT,NOBROWNOUT,MCLR
#include delay(CLOCK = 1000000) //クロック
#include rs232(BAUD =9600, XMIT = PIN_C6, RCV = PIN_C7)
#include i2c(MASTER, SDA=PIN_A4,SCL=PIN_A5,STREAM=RTG)
#include i2c(MASTER, SDA=PIN_C0,SCL=PIN_C1,STREAM=LCD)

////////////////////////////////////
// 液晶表示器制御ライブラリ
// 内蔵関数は以下
// lcd_init() —— 初期化
// lcd_cmd(cmd) —— コマンド出力
// lcd_data(chr) —— 1文字表示出力
// lcd_clear() —— 全消去
////////////////////////////////////

//////// データ出力サブ関数
void lcd_out(int code, int flag)
{
    if (flag == 0){
        i2c_start(LCD);
        i2c_write(LCD,0x7C);
        i2c_write(LCD,0x40);
        i2c_write(LCD,code);
        i2c_stop(LCD);
    }
    else{
        i2c_start(LCD);
        i2c_write(LCD,0x7C);
        i2c_write(LCD,0x00);
        i2c_write(LCD,code);
        i2c_stop(LCD);
    }
}

//////// 1文字表示関数
void lcd_data(int ascii)
{
    lcd_out(ascii, 0);
    delay_us(50); //50μsec待ち
}

//////// コマンド出力関数
void lcd_cmd(int cmd)
{
    lcd_out(cmd, 1);
    delay_ms(2); //2msec待ち
}

//////// 全消去関数
void lcd_clear()
{
    lcd_cmd(0x01); //初期化コマンド出力
    delay_ms(15); //15msec待ち
}

//////// 初期化関数
void lcd_init()
{
    lcd_cmd(0x38); //Function set
    delay_ms(2);
    lcd_cmd(0x39); //Function set
    delay_ms(2);
    lcd_cmd(0x14); //InterVal
    delay_ms(2);
    lcd_cmd(0x74); //Cntrast Low
    delay_ms(2);
    lcd_cmd(0x5E); //Cntrast High
    delay_ms(2);
    lcd_cmd(0x6C); //Follwer
    delay_ms(2);
    lcd_cmd(0x38); //Function set
    delay_ms(2);
    lcd_cmd(0x0C); //DisPlay on
    delay_ms(2);
    lcd_cmd(0x01); //Clear Disp
    delay_ms(2);
}

//*****
//
//-----
//////// メイン関数
//-----
void main()
{
    long Count=0;

    setup_oscillator(OSC_1MHZ);
    lcd_init();
    lcd_cmd(0x02);
    printf(lcd_data,"LCD Test ");
    while(True){
        lcd_cmd(0xc0);
        printf(lcd_data,"%5lu ",Count++);
        delay_ms(1000);
    }
}
```