-273~300℃！Pi カメラ・サーモグラフィの LCD モニタ表示

●小型モニタに温度分布を表示する

小型 LCD 表示にチャレンジします．ベースとなるソースは raspberrypi_capture です．ディレクトリごと適当な場所に展開すると，

$make

で実行ファイルが生成されます．そのままなら単に画像が PGM 形式でセーブされるだけですので，このソースを改造して LCD 対応にします．LCD ライブラリはさまざまなものが存在しますが，ソース・ファイルを単純にしたかったので今回は，

http://blog.tkjelectronics.dk/2010/03/arduino-mega-and-ili9320-display/

の ILI9320 用ライブラリの一部と，ILI9325 用初期化ルーチンは Arduino 用の UTFT ライブラリのものを使用しました．

●wiringPi を使えるにする

wiringPi は標準でインストールされていますが，使うために include 文で宣言します．また I2C も wiringPi ですので以下のように記述します．

```
#include <wiringPi.h>
#include <wiringPiI2C.h>
```

次に Makefile の LIBS を以下の様に変更します．

```
LIBS = -lm -lwiringPi
```

これで wiringPi が使えるになりました．

●ラズパイ用 IO

元となるライブラリは Arduino 用なので IO が異なります．ラズパイの GPIO 用にバス出力用の関数を作成しました．速度はそんなに速くなく，もっと効率の良いものがあるかもしれませんが，GPIO の好きなピンにデータ・バスを割り付けることができます．ソースの一部を下記に示します．1 バイトのデータを 1 ビットずつに分けて，wireingPi の digitalWrite 関数で 1 ビットずつ出力します．

```
static void LCD_busout(unsigned char data)
{
    unsigned char d0,d1,d2,d3,d4,d5,d6,d7；
   d0 = data & 0x01;
   d1 = (data >> 1) & 0x01;
   d2 = (data >> 2) & 0x01;
```

```
    d3 = (data >> 3) & 0x01;
    d4 = (data >> 4) & 0x01;
    d5 = (data >> 5) & 0x01;
    d6 = (data >> 6) & 0x01;
    d7 = (data >> 7) & 0x01;
    digitalWrite(PIN_D0, d0 );
    digitalWrite(PIN_D1, d1 );
    digitalWrite(PIN_D2, d2 );
    digitalWrite(PIN_D3, d3 );
    digitalWrite(PIN_D4, d4 );
    digitalWrite(PIN_D5, d5 );
    digitalWrite(PIN_D6, d6 );
    digitalWrite(PIN_D7, d7 );
}
```

●表示

Lepton より読み出したデータは 14 ビットの情報です．これを見やすくするために，画面内の最低温度と最高温度を計算し，その間を 1280 段階で正規化して，カラー画像で表示します．元となるソースはリファレンス・コードの中にある windows 用表示アプリの ThermalView です．

この中の generate_pallette でカラーパレットを生成して， scale_image で，最低，最高温度を計算してスケーリングします．画像は 80×60 ですので，LCD の水平解像度 240 に合わせるために 1 ピクセル当たり 3×3 画素の四角形を描画しています．

●I²C 経由による温度取得

raspberrypi_capture より leptonSDKEmb32PUB を利用するのは，そのままでは煩雑なので，リファレンス・コードにある arduino_i2c のソースを流用して必要最小限部分の実装を行いました．

原稿執筆の 2017 年 10 月時点では，正確な値が読み出せておらず原因究明中です．Arduino を使用した動作は確認できているので，移植の際の問題かと思われます．wiringPi の I²C 関数は I²C のスタート・コンディションやストップ￥コンディションは関数内で隠蔽されており，単純な read/write 関数なら，1 バイト送受信するたびにストップ・コンディションが発生してしまうみたいです．引き続き解析を行い，後日ソース・コードを更新する予定です．現状では，温度は 30℃固定として表示用温度の計算を行っています．

●温度表示

画面内の最低，最高温度を表示するようにしています．

画面内の青色が最低温度，赤が最高温度です．写真では人間の顔を表示していますが，最高温度表示

が 32.27℃で若干低く出ています. これは前述の通りチップ温度を 30℃と仮定しているためで, 今後 I2C 経由で実際の温度が読み出されれば，誤差は無くなると思われます．また応用例としては，画面内の任意の場所（例えば中心部分）の温度を表示するようにすると更に利便性が高まると思います.