

# PICマイコンによるタッチパネル付き TFT 液晶の制御

小山工業高等専門学校 電子制御工学科 5年 本澤 上  
指導教官 金野 茂男 (<http://www.oyama-ct.ac.jp/D/kinnoken/>)

## 1、はじめに

近年発売されている携帯電話には、タッチパネルが搭載された機種が数多くある。画面に直接触れて操作するため、直観的に扱いやすいという点や、他にボタンやスイッチを用意しなくてよいという大きな利点がある。現在ではインターネット通販で、タッチパネル付きの液晶や、タッチパネルそのものを簡単に入手することができるため、この制御を行うことができれば、機器の操作性の向上など、多くの恩恵を受けることが出来ると考えられる。

今回は、PIC を使用して液晶とタッチパネルを制御し、簡単なお絵かきが出来るようにすることを目的として、研究・製作を行った。動作の様子を図1に示す。

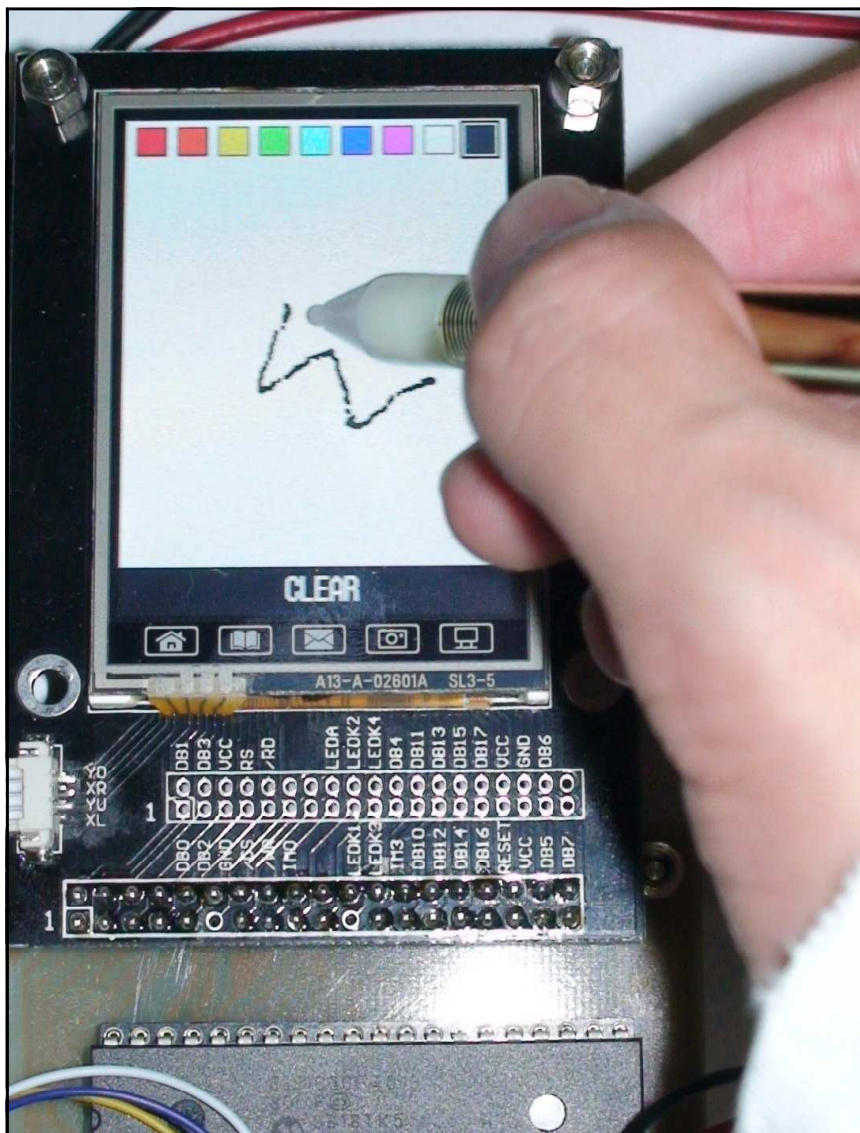


図1、動作している様子

## 2、概要

以下に、使用したデバイス・素子を列記する。

- ・dsPIC30F4013 (Microchip)
- ・2.4inch タッチパネル付き TFT 液晶 (aitendo@shopping - YHY024006A-PCB)
- ・3 端子レギュレータ 3.3V
- ・セラミック発振子 7.37MHz
- ・抵抗、コンデンサ等 数本

この度は、高速な動作が出来ること、A/D 変換が出来ること、入出力ピンが豊富であること、手元にあることなどの条件から、dsPIC30F4013 を使用することにした。クロックは、7.37MHz を 16 倍して 117.92MHz としている。dsPIC のデータシートでは、最大クロックは 120MHz とされているため、ほぼ最高速度で動作させたと言える。

液晶・タッチパネルについては、これらが一体となっていて、更にキャリーボードに予め接続されているものがネット通販で手に入る。著者は、株式会社秋葉原が管理・運営を行う電子工作関連の通販サイト aitendo@shopping にある、2.4inch のタッチパネル付 TFT 液晶を購入した。商品コードは YHY024006A-PCB であった。この液晶の動作電圧は 3.0V~3.3V と使いやすいため、電源用のレギュレータは 3.3V を 1 つだけ使用した。この TFT 液晶と PIC との接続は、16bit パラレルと 8bit パラレルのどちらかを選ぶことが出来るが、8bit 接続で使用する場合、液晶とキャリーボードを繋ぐフレキシブル基板上のチップ抵抗を付け替えなくてはならない。これは非常に難しい作業であるため、16bit 接続で使用することにした。

参考までに、著者の開発環境を記しておく。

- ・ OS : Windows XP Pro SP3
- ・ PIC ライタ : Microchip 社 ICD2 (ICD2 Version - 08.43.02.02)
- ・ 開発ソフト : MPLAB IDE 8.46 & ASM30 コンパイラ 3.23

## 3、タッチパネルの原理

今回使用する 4 線抵抗膜式のタッチパネルが、どのようにして接触点を検出しているのか説明する。解説のための図を図 2 に示す。まず、4 線抵抗膜の 4 線とは X+, X-, Y+, Y- という名称の 4 本の線のことを指す。YHY024006A-PCB では、これらの線はそれぞれ XL、XR、YD、YU と呼ばれているため、以降はそのように呼称する。

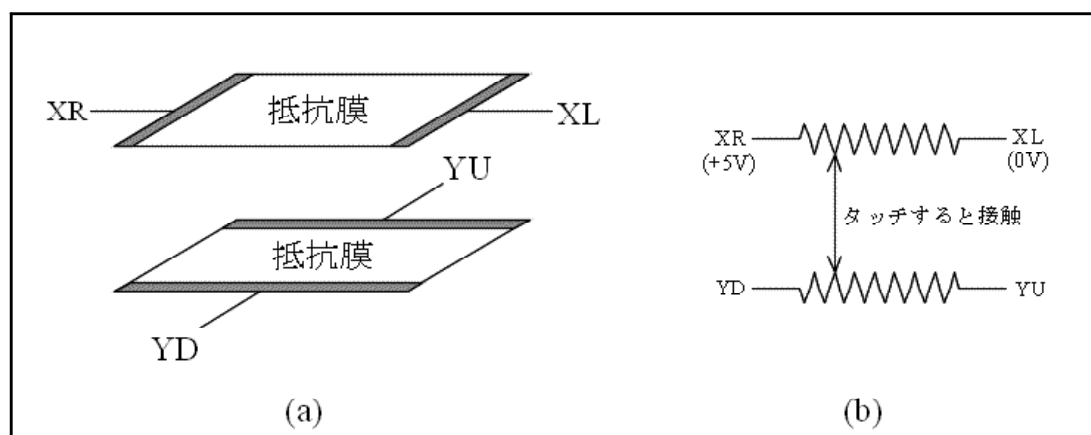


図 2、4 線抵抗膜式タッチパネルの原理

4線抵抗膜式タッチパネルの基本構成は、それぞれ2本の端子と電極を持った2枚の抵抗膜が重ねられているという、意外に単純なものである(図2-a)。これらの中にはドット状のスペーサが等間隔に並べられており、通常の状態では接触しないようになっている。

使い方は、例えば図2-bに示すように、XRに+5Vを印加し、XLをグランドに接続したとする。抵抗膜上のある点をタッチすれば、2枚の膜が接触し、分圧された電圧がY側の抵抗膜に現れることになる。YU、YDに電流が流れ込まない(PICの端子がアナログモード=ハイインピーダンス)ならば、YU、YDに現れる電圧がタッチ位置を示していることになる。つまり、タッチした位置のX座標を読み取ることができるのである。同じ様にY座標を読み取れば、触れた点の座標(X,Y)が確定するのである。

しかし、実際にはノイズなどの影響で、現れる電圧値はある程度ばらついてしまう。そのため、何度かサンプリングをしてから平均値を採用するなどの工夫が必要である。

次に、この4線抵抗膜方式の特徴を述べる。まず、第1の特徴は単純な構造である。つくりが複雑でないため、他方式に比べて非常に安く入手できる。また、座標がアナログ値で出力されるため、高精細化しやすいという利点もある。一方、構造上の理由から、傷などに弱く、寿命が短いという欠点も持っている。最悪の場合、長い傷が1本できただけで使用できなくなってしまうこともある。

#### 4、ハードウェア

図3と図4に、回路図と使用したパターン図を示す。このパターン図を使用して回路を製作する場合、解像度の問題で綺麗に印刷できないことがあると思われる。印刷する際の反転・非反転などの点も含め、注意して頂きたい。どうしても上手くいかない場合は、回路パターン作成ソフト等で、各自製作してほしい。

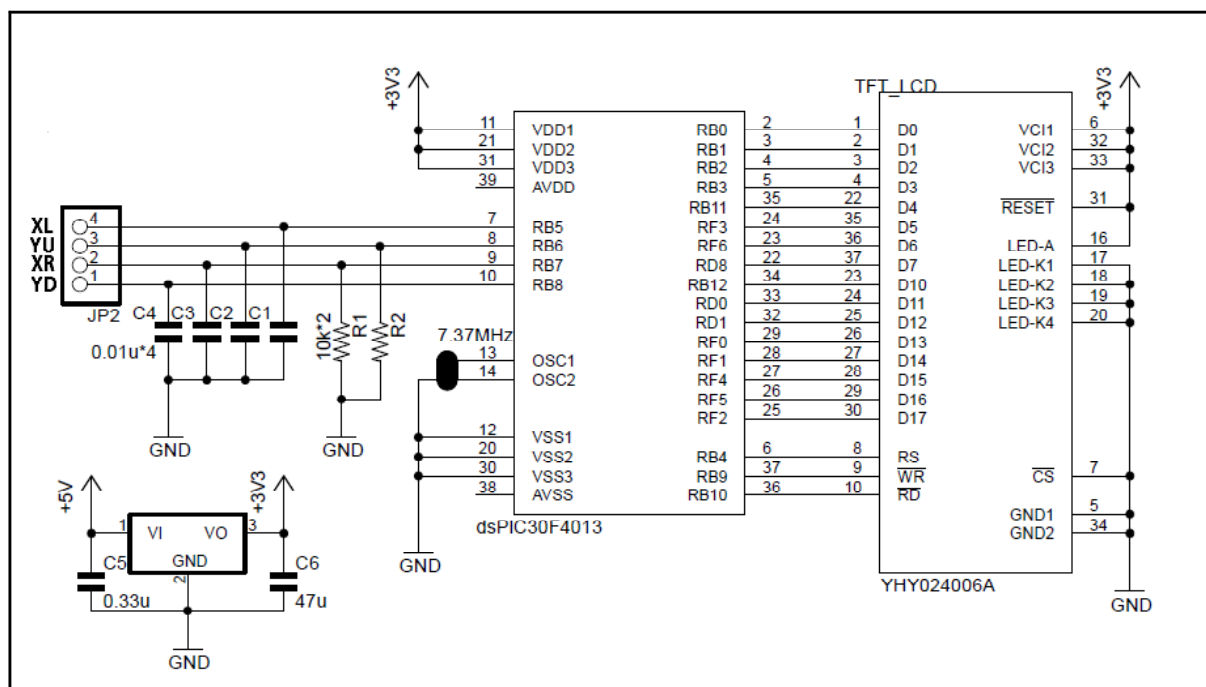


図3、回路図

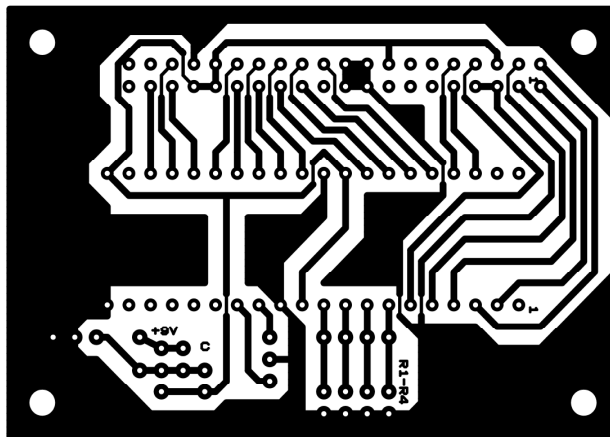


図 4、パターン図

部品の実装について説明する。基本的には図 5、図 6 の 2 枚の写真を見ていただければ、おおまかな配置は確認できると思われるが、分かりにくい点などに関して記述する。まず、最も分かり辛いのは、タッチパネルとの接続の回路における、コンデンサの位置である。ノイズの低減のために後から付け足したものであるため、非常に不格好な配置になってしまった。図 6 の下部に見えるように、ハンダ面に実装するか、最悪付けなくてもそれほど問題ではない。なお、回路修正を何度か行ったためプルダウン抵抗周辺に余分な穴がいくつかあるが、無視していただいて構わない。

ちなみに、回路図では電源は+5V となっているが、著者は 006P 電池用のコネクタを介して電源装置から供給している。この辺りは、各自工夫をしてほしい。

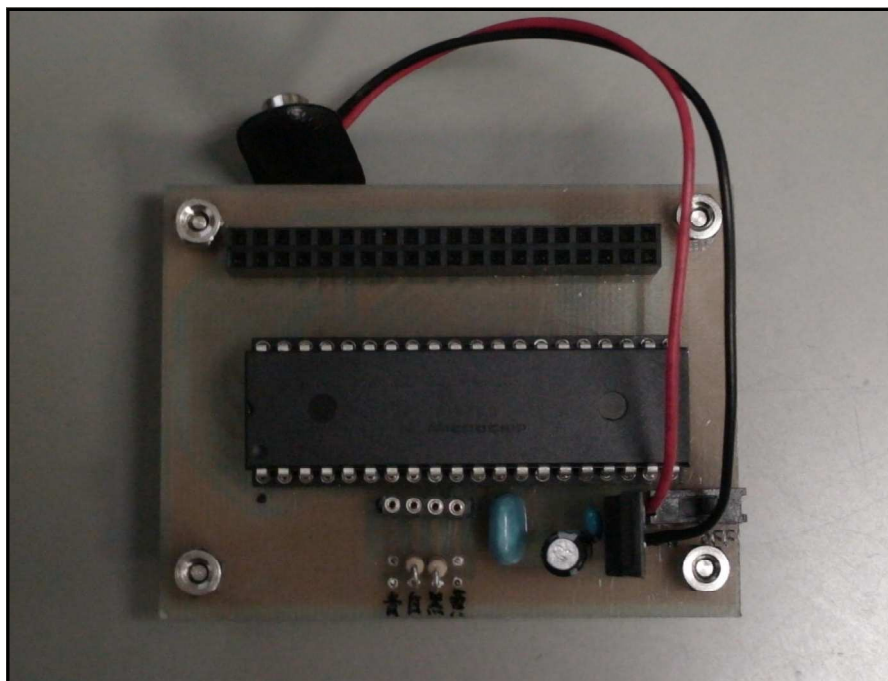


図 5、回路写真①

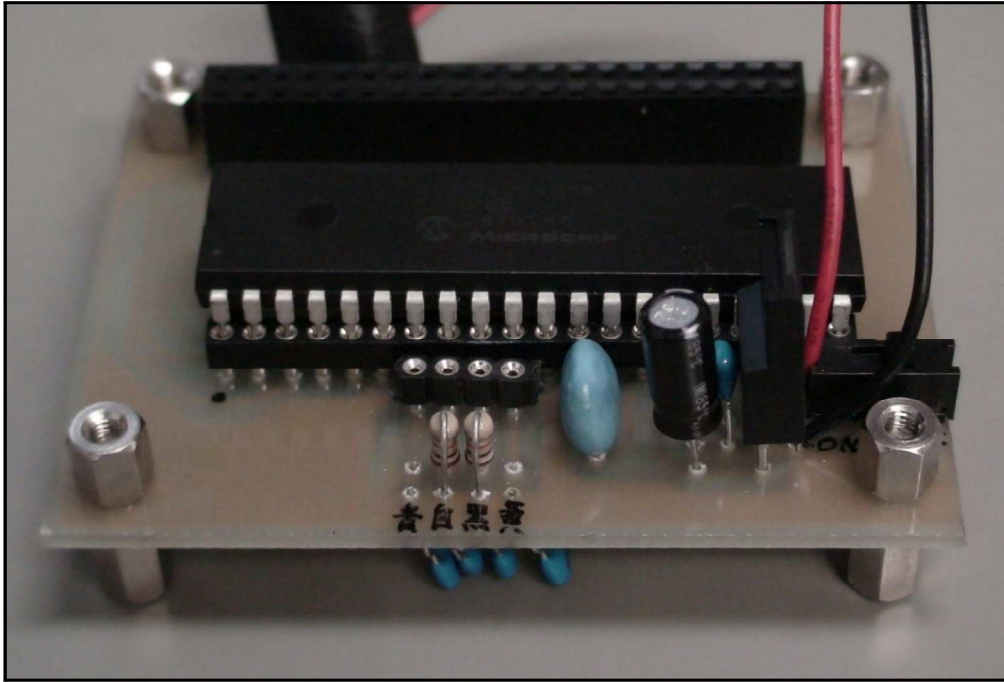


図 6、回路写真②

次に、TFT 液晶との接続に関して注意がある。今回使用した TFT 液晶は、2 列× 19 本のピンを持っている。しかし、世に出回っているピンヘッダ・ソケットは 2 列× 20 本が一般的である。そのため、TFT 液晶に接続するオスのピンヘッダは、図 7 の様に端のピンを 2 本引き抜いたものを使用している。基板に取り付けるソケットの穴を減らすことはできないので、20 ピンタイプをそのまま使用している

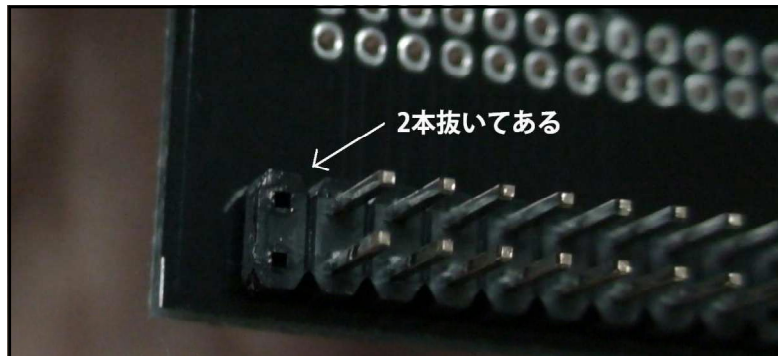


図 7、キャリアボード端子の写真

図 8 は、タッチパネルと PIC を繋ぐケーブルの、先端部分の写真である。抜き差ししやすいようにオスピンをハンダ付けして使用している。しかし、どう見ても耐久性に問題があるため、熱圧縮チューブで保護する等の対策が必要ではないかと考えている。なお、このケーブルは YHY024006A-PCB 付属のものである。

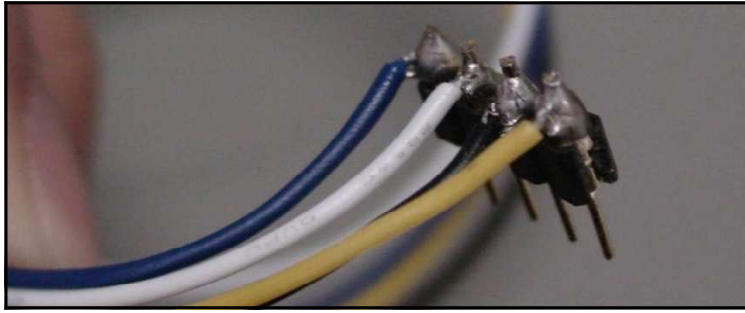


図 8、タッチパネルケーブルの先端部分の写真

図 9 に、完成したハードウェアの全体像を示す。液晶の向きを 180 度回転させたほうが、コンパクトになり見た目も良かったのだが、プログラムの修正などをやり易くするためにこのような配置となった。

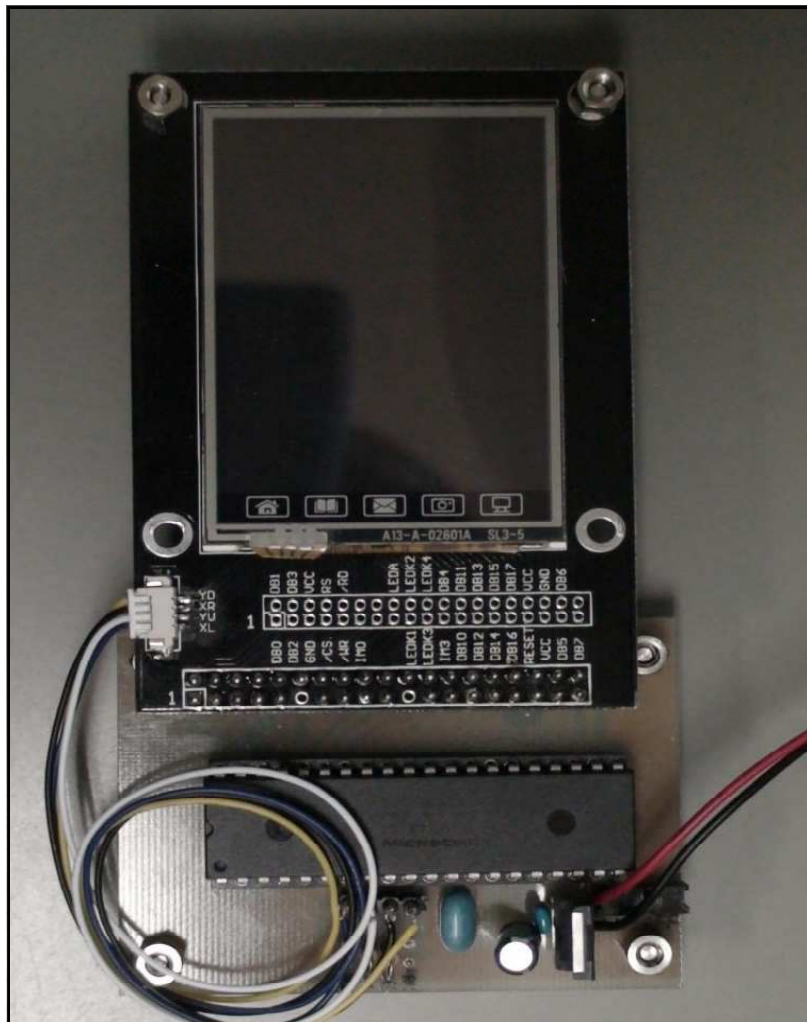


図 9、システムの全体写真

## 5、ソフトウェア

### 5-1 お絵かきソフトの概要

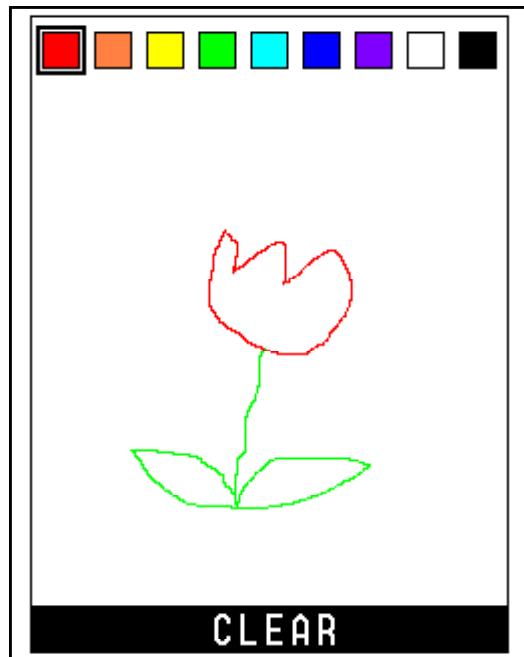


図 9、画面イメージ

製作したハードウェアに実装するソフトウェアの概要を説明する。おおまかなイメージは図 9 に示したような画面である。画面上部にはペンの色を選択できるパレットを用意し、画面の下部には描いた絵を消去するクリアボタンを設ける。現在選択している色分かるように、選んでいる色のパレットの周囲を黒い枠で囲むことにする。そして、画面中央の空白部分に自由にお絵かきが出来るという構成である。

### 5-2 プログラム

ここでは、PIC に書き込んだアセンブラプログラムのおおまかな説明をする。詳しい内容は、添付するソース（著者が所属する研究室のホームページよりダウンロード可能）を見てほしい。アセンブラでのプログラミングにはあまり慣れていないため、少々見苦しいソースコードになってしまったが、ご容赦いただきたい。

まず、液晶に関する部分であるが、TFT-LCD の基本的な動作（初期化、データ・コマンドの送信）は、aitendo による C 言語サンプルをアセンブラに書き換えたようなものになっている。その他のドット描画や塗りつぶしのルーチンは、著者が作成したものである。細かい通信の仕様については、ソースコードを読んでいただくか、YHY024006A のコントローラチップである ILI9325 のデータシートを読んでいただきたい。また、前述した aitendo による C 言語サンプルも非常に参考になるであろう。

次にタッチパネル制御部分を解説する。原理は前に述べた通りで、X、Y それぞれ 64 回のサンプリングを行い、それらの平均をとっている。dsPIC30F4013 の A/D 変換結果は 12bit で出力されるため、単純に 64 回分のサンプリング結果を足し合わせるとオーバーフローすることは目に見えている。そのため、2 回サンプリングするごとにそれらの和を 4 で割るこ

とで、オーバーフローを防いでいる。なお、画面がタッチされていない場合の Y 座標は 0 を示すので、「Y 座標がある程度の値を示したらタッチ検出」とし、タッチが検出できていない間は待機とした。タッチが検出できたら、次は座標を取得しなければならないが、

- ・ Y 座標を 2 回取得し、2 回のサンプリング値の差が 30 未満であること
- ・ X 座標を 2 回取得し、2 回のサンプリング値の差が 30 未満であること

という条件を満たしたときのみ、値を採用することにした。このようにしなければ、タッチする際の力加減などで、タッチしていないはずの点を検出してしまうためである。

## 6、動作結果

図 10 に、実際に動作している様子を示す。カラーで絵が描けていることがわかる。また、CLEAR ボタンを長押しすることで画面の初期化が行えることも確認できた。

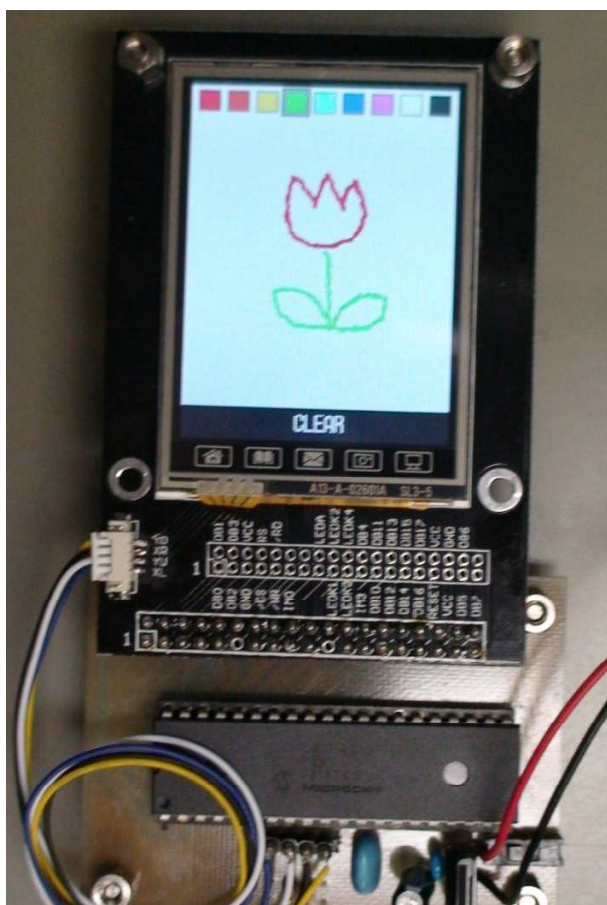


図 10、動作の様子

## 6、今後の課題

今回の研究では、イメージした通りのお絵かきアプリケーションを製作することが出来た。タッチパネルと液晶画面を制御するという目標は達成できたと言えるが、様々な問題が残っている。

まずハードウェアの問題は、PIC の 16bit バス (PORTB) を生かしていないということである。片面基板での製作を目指したためにソフトウェアに皺寄せが生じてしまい、結果として実行速度が落ちてしまった。また、思っていた以上に 3 端子レギュレータが高熱になることも問題点として挙げられる。消費電力を下げることはそう簡単には出来ないので、ヒート



シンクを設けるなどの対策が必要である。

一方ソフト側では、タッチパネル上の座標の歪みを修正する `adjust_pos` ルーチンと、A/D 変換値を画面上のピクセルに変換する `get_pixel` ルーチンの計算式に問題があると思われる。`adjust_pos` ルーチンでは、画面の 4 隅をタッチした際に、A/D 変換によって得られる電圧から算出した式

$$\begin{cases} x = X_{AD} + \frac{Y_{AD} - 0x120}{0x28} \\ y = Y_{AD} \end{cases} \quad (1)$$

を用いたが、この式は著者が使用したタッチパネルにしか適用できないものであり、汎用性は全く無いと言える。実用的なものを目指すのなら、タッチ位置の補正を行えるような機能を作成することが必要不可欠であろう。つまり、いくつかの点を指定してユーザーにタッチしてもらい、その情報から 4 隅をタッチした際の電圧値を計算出来るような機能が求められるのである。具体的には、以下の様な計算になる。

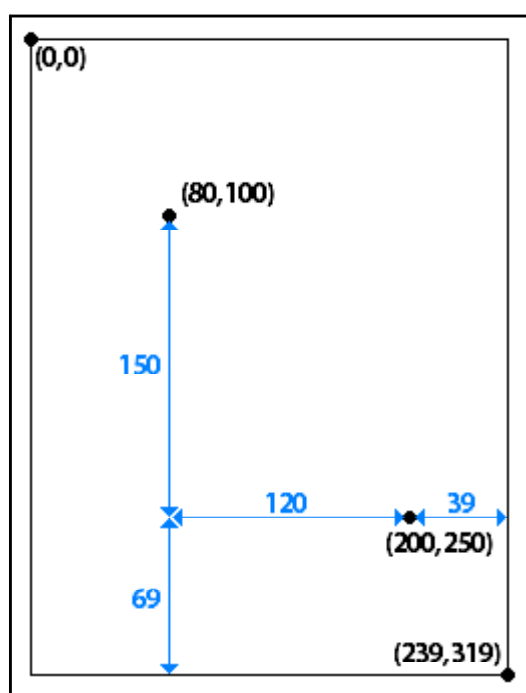


図 11、座標補正の例

仮に、ユーザーに触れてもらう点を (80,100) と (200,250) の 2 点とする。そして、これらの点をタッチした際に得られる A/D 変換値を、それぞれ  $X_{80}$ 、 $Y_{100}$ 、 $X_{200}$ 、 $Y_{250}$  とおく。さらに、画面端の (0,0) と (239,319) をタッチした際に得られる A/D 変換値を、それぞれ  $X_{\min}$ 、 $Y_{\min}$ 、 $X_{\max}$ 、 $Y_{\max}$  とおけば、

$$\left\{ \begin{array}{l} X_{\min} = X_{80} - 80 \times \frac{X_{200} - X_{80}}{120} \\ Y_{\min} = Y_{100} - 100 \times \frac{Y_{250} - Y_{100}}{150} \\ X_{\max} = X_{200} + 39 \times \frac{X_{200} - X_{80}}{120} \\ Y_{\max} = Y_{80} + 69 \times \frac{Y_{250} - Y_{100}}{150} \end{array} \right. \quad (2)$$

という計算により画面端の座標が求められる。これらの値を用いれば、

$$\left\{ \begin{array}{l} X_{dot} = 239 \times \frac{X_{AD} - X_{\min}}{X_{\max} - X_{\min}} \\ Y_{dot} = 319 \times \frac{Y_{AD} - Y_{\min}}{Y_{\max} - Y_{\min}} \end{array} \right. \quad (3)$$

として、A/D 変換値からピクセル（ドットの座標）を計算することが出来る。

## 7、今後の拡張性

今回の成果を活用すれば、液晶とタッチパネルを組み合わせた機器を製作できると思われる。具体的には、市販されている音楽プレーヤーなどのような機器である。タッチパネルを使用すれば、より直観的で、より分かりやすいインターフェースが作成できると考えられる。

さらに、タッチパネルを用いることで受けられる最大の恩恵は、スイッチ類を省くことができる点であるから、従来の機器を小型化することもできるであろう。例えば、オシロスコープやファンクションジェネレータ等のある程度大きな装置も、スイッチ類を無くせば小さくできると考えられる。

また、タッチパネルを使用した機器にある代表的な機能として、マルチタッチ（2点以上のタッチを検出可能）がある。しかし、これについては特許の問題があり、現在も各社間で争いが続いている。そのため、しばらくはマルチタッチ機能の実装は不可能ではないかと思われる。

## 8、参考文献

- [1] Active matrix 2.4" colour Digital TFT module Technical Product Specification  
ShenZhen ShuangXiang LCD Tech. Co., Ltd. 2009年5月20日
  - [2] ILI9325 - a-Si TFT LCD Single Chip Driver 240RGBx320 Resolution and 262K color Datasheet  
ILI Technology Corp. 2007年7月5日
  - [3] トランジスタ技術 2009年8月号 CQ 出版社 2009年8月1日
- [1],[2]は、aitendo@shopping の YHY024006A のページからダウンロード可能

2010年10月13日