

■ PIC18F4620 で TFT カラー液晶の制御

更新情報

【12/1/3】

- ・加速度センサの型番修正 (MMA7361LC→KXM52-1050)
 - ・下記プログラムを使用したゲームの公開
-

概要

TFT カラー液晶(YHY024006A)が手に入ったので、上記 PIC で動かしてみました。

現在、このページで公開してる関数群で、直線、四角形、円形、文字列の表示、タッチ座標の取得、16bit 画像の表示、グラデーションの計算等が可能です。

(今後機能を増やす予定)



(タッチした座標に適当なグラデーションで円を描くプログラム)

動作環境

- PIC18F4620[40MHz(10x4)動作]
- [YHY024006A-PCB](#)
- ブレッドボード [EIC-801](#)
- MPLAB IDE v8.80 + C18 コンパイラ
- PICkit 2

YHY024006A-PCB の裏にブレッドボードを貼り付け、ジャンパー線で配線しました。

(YHY024006A-PCB の端子には 2x20 のピンソ

ケットを使用)



表からみた図↓

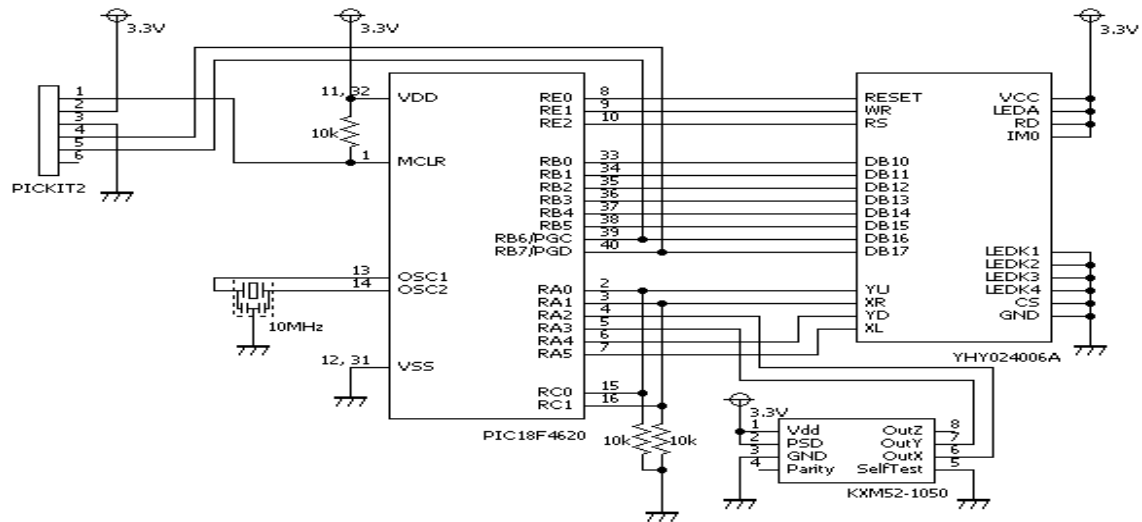


このやり方だと 40pinDIP タイプのマイコンならなんでも繋がりますし、非常にコンパクトになるのでオススメです。

現在、YHY024006A-PCB が手に入りにくくなっていますが、[LCM240YP04-07](#) といった YHY024006A の互換機と、液晶 I/F 変換基板 [IFB-YHY024006A](#) などで代用できます。

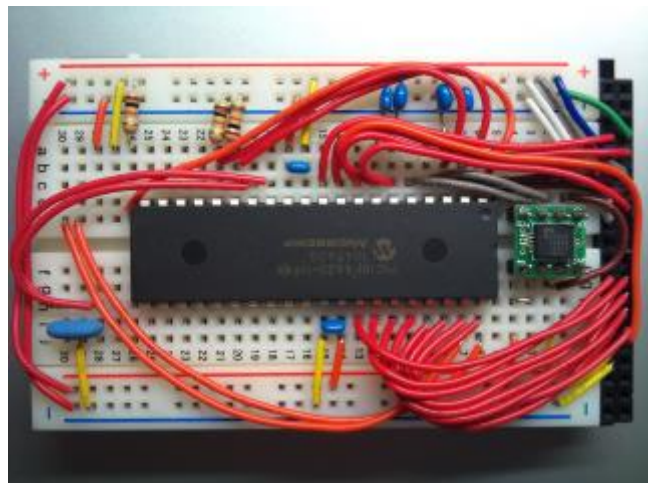
ただし、今回 8bit モードで通信しているので、抵抗を付け替えないと 8bit モードにならないタイプの液晶は、抵抗を付け替える必要があります。

回路図



加速度センサが付いていますが、今回は使用していないので無くても問題ありません。
(使用した加速度センサ:[KXM52-1050](#))

ブレッドボード上の配線↓



ソースコード

YHY024006A 用の関数群と、サンプルプログラム、MPLAB 用のプロジェクトを zip でまとめました。

クリックでダウンロード→[circle_test2.zip](#)

書き込むときの注意として、必ず pickit2 の出力電圧を 3.2v 程度にして 行なってください。

5v ではおそらく液晶が壊れます

circle_test2.zip の中に入っている

“YHY024006A.h”の使い方としては、下記プログラムのよう

初期化(Lcd_Init)、font の設定、画面クリア、Lcd_Draw_〇〇といった流れで〇〇部分を 液晶に表示することができます。

“YHY024006A.h”を使ったプログラムのサンプル
↓

```
#include <p18f4620.h>
```

```
#include <delays.h>
```

```
#include <stdlib.h>
```

```
#include "YHY024006A.h"
```

```
#include "font_patt.h"
```

```
#pragma config OSC=HSPLL, FCMEN=OFF, IESO=OFF, PWRT=ON
```

```
#pragma config BOREN=ON, BORV=2, WDT=OFF, WDTPS=1024
```

```
#pragma config MCLRE=ON, PBADEN=OFF, CCP2MX=PORTC
```

```
#pragma config STVREN=OFF, LVP=OFF, DEBUG=OFF
```

```
#pragma config CP0=OFF, CP1=OFF, CP2=OFF, CP3=OFF, CPB=OFF
```

```
#pragma config CPD=OFF, WRT0=OFF, WRT1=OFF, WRT2=OFF, WRT3=OFF
```

```
#pragma config WRTB=OFF, WRTC=OFF, WRTD=OFF, EBTR0=OFF
```

```
#pragma config EBTR1=OFF, EBTR2=OFF, EBTR3=OFF, EBTRB=OFF
```

```
//=====main program=====//
```

```
void main(void) {
```

```
    int i;
```

```
int size;
```

```
COLOR16 color, color2;
```

```
int x, y;
```

```
//---Port I/O Setting---//
```

```
ADCON1 = 0x0F;
```

```
TRISA = 0x0F;
```

```
TRISB = 0x00;
```

```
PORTA = 0x00;
```

```
PORTB = 0x00;
```

```
Delay1KTCYx(100);
```

```
Lcd_Init();
```

```
Lcd_Set_Font(font_patt_816);
```

```
Lcd_Set_Font_Sub_Color(WHITE);
```

```
Lcd_Clear(WHITE);
```

```
while(1) {
```

```
    size = rand()%30+10;
```

```
    color = (COLOR16)rand()*2;
```

```
    color2 = (COLOR16)rand()*2;
```

```
    Lcd_Set_Font_Sub_Color(color2);
```

```
    Lcd_printf(0, 0, color, "Let's Touch Display!!!");
```

```
    while(Lcd_Get_Touch(&x, &y) == 0);
```

```
    Lcd_printf(-1, -1, color, " %3d %3d", x, y);
```

```
    for(i=0; i<size; i++) {
```

```
        Lcd_Draw_Circle(x, y, i, Gradation(color, color2, size, i));
```

```
}  
}  
}
```

なお、タッチパネル関係はかなり適当に作ってあるので、各々調節してください。

使用例

“YHY024006A.h”を使うことで、簡単なゲームも作れたりもします。

今回は、4つ同じ色を揃えて消すというあのゲームを作ってみました。



クリックでダウンロード→[dango_drops.zip](#)

このプログラムでは加速度センサを使用しますので、上記回路図通り KXM52-1050 を付けてください。

操作方法は、画面を左右に傾けることで移動し、タッチで回転するようになっています。

現在、他に作れそうなゲームを募集しています。
良い案があれば掲示板に書いていただくと助かります。