# LCR使用恩智浦COG显示输出

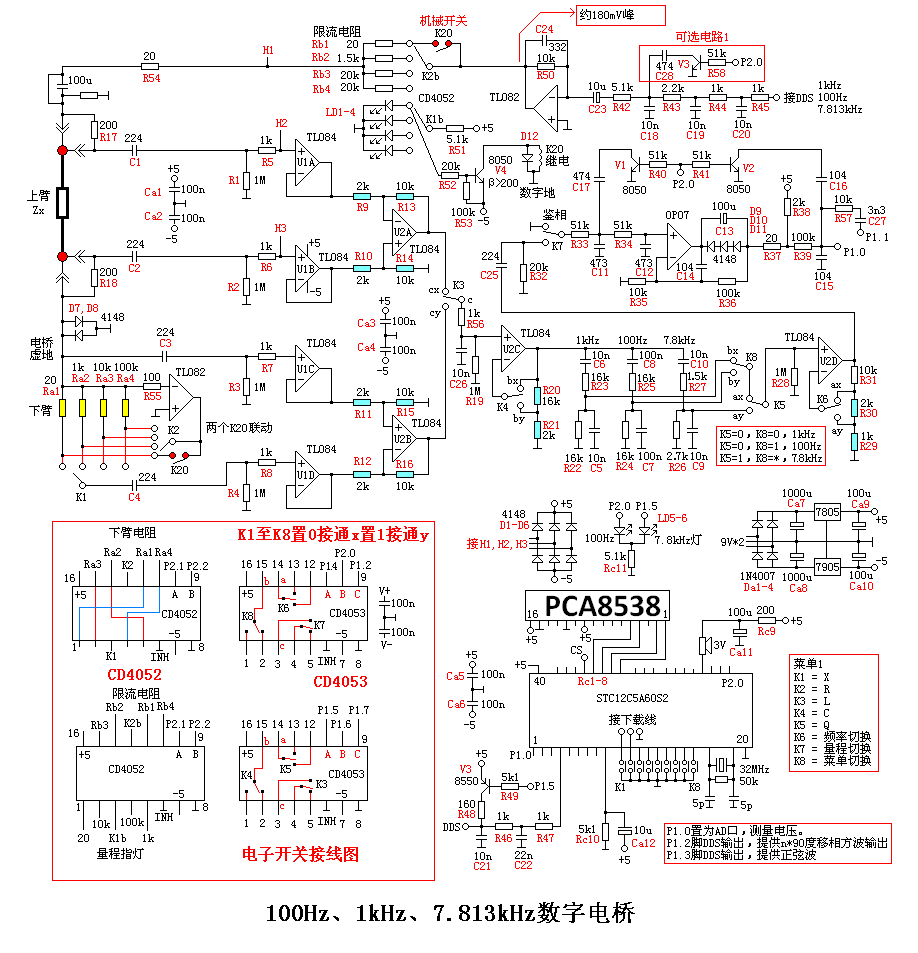
## 概述

LCR是电子工程师手上常用的一种测量仪器，但是成品LCR价格昂贵，作为电子工程师的我们，正好可以发挥我们的特长，自己DIY，好在上网找一找，很多开源的LCR，本文所用的LCR就是矿坛许老师开源的，其原机设计MCU用的是STC12C06S2，显示用的是LCD1602，显示效果不太满意，一直想试试COG点阵屏的显示效果，正好NXP搞活动，让我有机会试试。

## 原理

关于LCR，原作者有详细的原理分析及制作说明，喜欢的同学，放狗搜下就有了，也有DIY爱好者提供套件，没有必要自己从头来过，直接购套件改制，我采购的硬件模拟部份和数字部份所有器件都在一块板，通过插线引出输出到1602显示，我们要做的就是将输出更改到PCA8538显示，其余沿用原来的设计， 当然配套的软件也必须相应更改。

在原机原理图上，只需要做很小的改动就可以实现，改制后的原理图如下:



## PCA8538

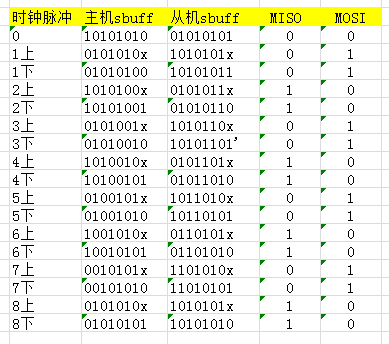
再说说PCA8538， 多通道，多模式，多背板，段式驱动的COG，手册上的这些概念让人看的眼晕，要想很好的驱动和使用PCA8538，细读手册是必需的，总之这是块可定制的IC，宽电压范围，功能强大，实验板基本上把它的所有功能都罗列出来了，7段，14段，点阵显示都用一片IC实现，你可以通过编程看显示效果，熟悉它的控制方式和命令，由于LCR显示信息主要是数字及单位符号，在这块实验板上适用的显示区域只能是点阵区了，所以以下改制主要针对点阵区的驱动实现。

## 软件

软件是重头戏，改动思路也是尽可能沿用原程序，毕境这个系统经过长时间的沉淀，代码中的思想经过验证有其合理部份，我们主要改动显示部份，其余部份尽可能不改动，以下对改动过程中的要点说明如下

### SPI

STC12C06S2自带硬件SPI，但很不幸，由于LCR中已使用了这些IO，当务之急是必须先写个用通用IO模似实现SPI的程序，所以简单说下SPI的工作原理，SPI有多种模式，我们使用的模式是，SCK时钟上升沿来时 发送，下降沿来时接收， 高位先发送， 实际上上升沿来时，主机和从机已经把信号已同时送到MISO和MOSI， 等下降沿时，主机和从机将数据保存到变量中，传送一个字节需要8个脉冲， 以下是说明每个脉冲时，发送字节及接收字节的BIT位变化情况。



可以看到，发和收是同时进行的，双工，在发送的同时也收到从机送回的数据，这里我们只是关心传送出去，所以忽略送回的数据。

以下是程序实现，是不是很简单啊。

uchar SPI\_Sends(uchar out\_data)

{

uchar i,n;

for(i=0;i<8;i++)

{

n<<=1;

SCK=0;

MOSI=(out\_data&0x80);

if(MISO) n++;

SCK=1;

out\_data<<=1;

}

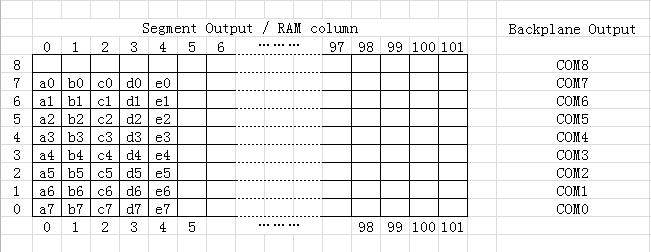
return n;

}

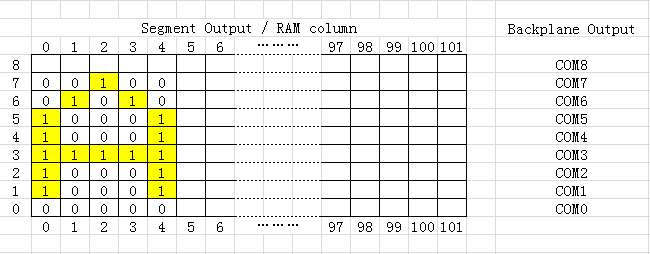
### 点阵字库

用点阵方式显示，一定会用到字库，有各种各样的字库，对于字符显示来说，有5X8 ，8X8 ，24X24……等等， 一个字符的点越多显示的字就越精细，对应占用的空间及处理时间也越多，这对于存储空间小的MCU是个挑战，对于STC12C06S2，空间不是问题，为了节省处理时间，同时更重要的是为了尽可能显示多的字符，我选用的是5X8点阵字符 。

PCA8538的显示RAM, 就是个矩阵，它有多种模式可选择设置使用，我选择1：9模式，其Y轴纵向从0－8共9个Bit, X轴横向从0－101共102个bit，纵向只用到8Bit，传送到显示RAM的5个字节，顺次摆放在RAM中，其存储映象如下图所示：



PCA8538根据显示RAM的值决定显示的点，其中1对应开（白色），0对应关（黑色），根据RAM中存放的值，屏幕点亮对应的点，就是我们看到的字符，假设我们传送到显示RAM的数据，是需要显示的字符A，哪么这些数据的值及显示出来的效果如下图所示：



把这些数据 按纵向取模，上高位 换算成16进制数据记为

0x3E, 0x48, 0x88, 0x48, 0x3E, // A

这是个5X8的点阵字模，一个字符占用5个字节的空间，按上面所说的方法把要显示的字符制作出来，就是我们最终需要的显示点阵字库，当把这些数据传送到显示幕时，相邻的两个字符间还需要加一列空列分隔，所以PCA8538可以最多显示一行， 95/6 =16个字符。

在程序中字库对应就是个只读的数组，对于可显示字符，最常用的是标准的ASCII，所以其对应的存放字模的顺序，按ASCII值安排是最自然不过的了， 这样就可以简单的根据ASCII值在字库数组中，定位要显示的字符 AscLib[ (ASCII - 32) \* 5 ]。

如此在屏上指定位置，显示一个字符自然是水到渠成：

void PCA8538\_WriteData(unsigned char XPointer, unsigned char YPointer, unsigned char Data)

{

unsigned char i;

cog\_cs = 0;

SPI\_Sends(0X20); //SUBADRESS

SPI\_Sends(0X80); //控制字节

SPI\_Sends((0x80 | ((XPointer >> 4) & 0x07))); // Set Data pointer x-MSB = 0

SPI\_Sends(0X80); //控制字节

SPI\_Sends((0x90 | (XPointer & 0x0F))); // Set Data pointer x-LSB = 0

SPI\_Sends(0X80); //控制字节--最后一个命令

SPI\_Sends((0xA0 | (YPointer & 0x01))); // Set Data pointer y = 0

SPI\_Sends(0X20); //SUBADRESS

for (i=0; i<5; i++) {SPI\_Sends(AscLib[ (Data - 32) \* 5 +i]);}

SPI\_Sends(0x00); //字符间加空列

cog\_cs = 1;

}

### 显示座标

7段，14段，点阵 三种显示区，使用点阵方式显示信息，是最灵活的方式，每个显示的点都可以通过坐标(x,y)标示，在PCA8538已有内置坐标指针指示当前点的位置（x,y），但对于用户而言，不可能会去数点，直观的方式是直接看到屏上的字符，如果我们使用5X8的点阵+再加一列空栏，每个字符在屏上，横向占用6个点，纵向8个点， 从屏的左边到右边，从0开时画分， 则可以把屏幕上显示每个字符的起始点位置固定下来，例如：我们要在屏上最左边，第0个位置显示一个字符，则应该在显示RAM的坐标(0,0) 开始写入字符的点阵数据， 如果要在第一个位置显示字符，则应该在显示RAM的坐标(0，48)开始写入字符的点阵数据，其余的依此类推，这就是显示座标，它和内置坐标指钍间的关系是：

//第1行x位

void lcd\_goto1(unsigned char x)

{

LCD\_SetXY(0,x\*6);

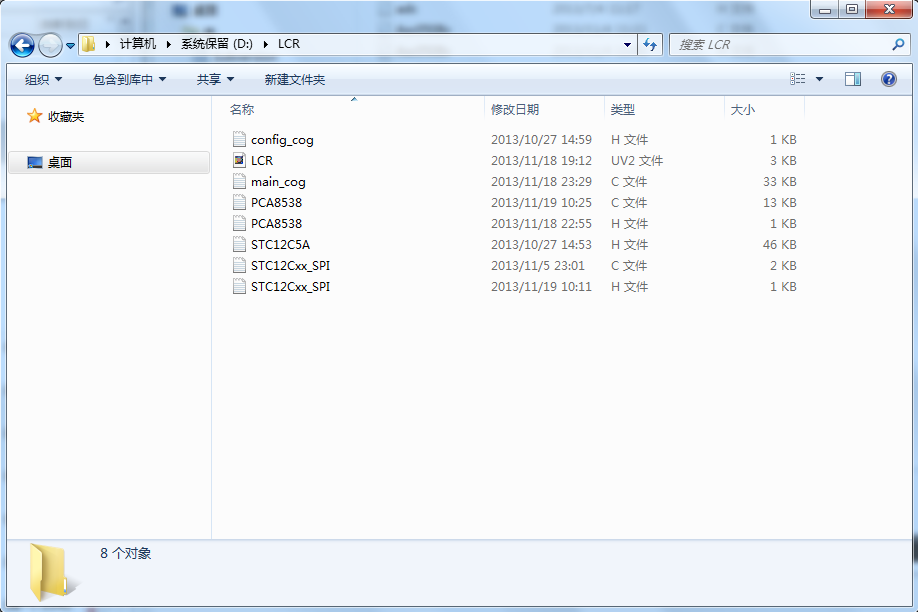
curr\_x=x\*6;

curr\_y=0;

}

### 整合

原程序就一个文件， 将原程序中显示部份剔除，改名为main\_cog.c , 加入PCA8538及SPI相关程序，编释后大功告成。

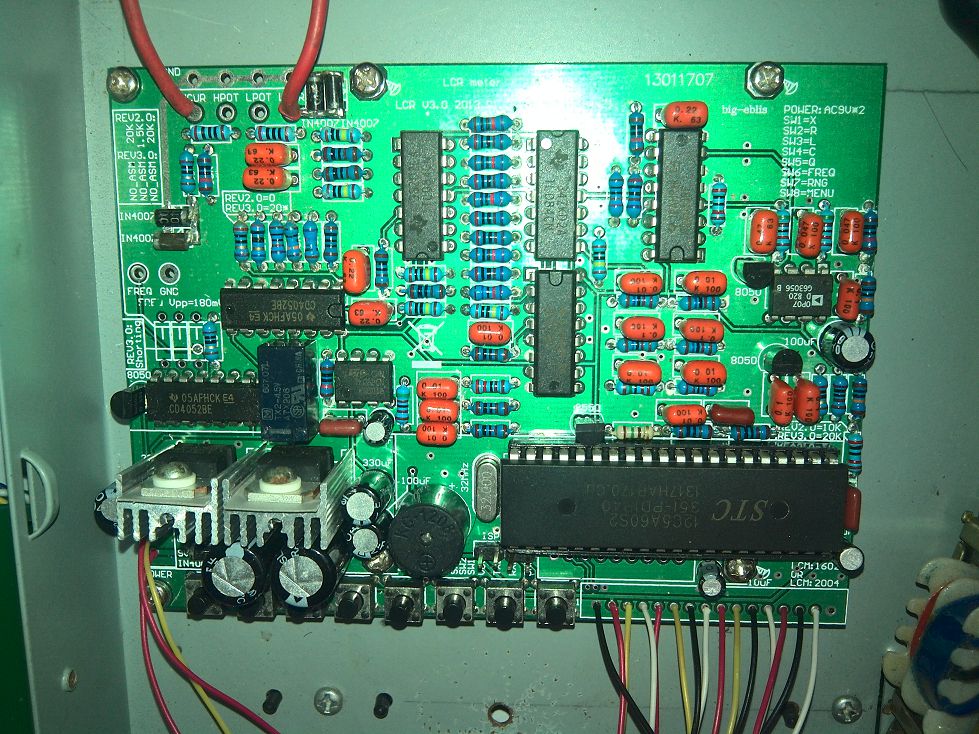


## 图片说明

### LCR主机，PCA8535 ，USB程序下载 全家福



### LCR主机



### PCA8535 接线

序号 COG 单片机

1 RST -------------------RST P0^4

2 SCL -------------------SCK P0^0

4 SDAOUT-------------MISO P0^1

5 SDAIN----------------MOSI P0^2

CS --------------------CS P0^3

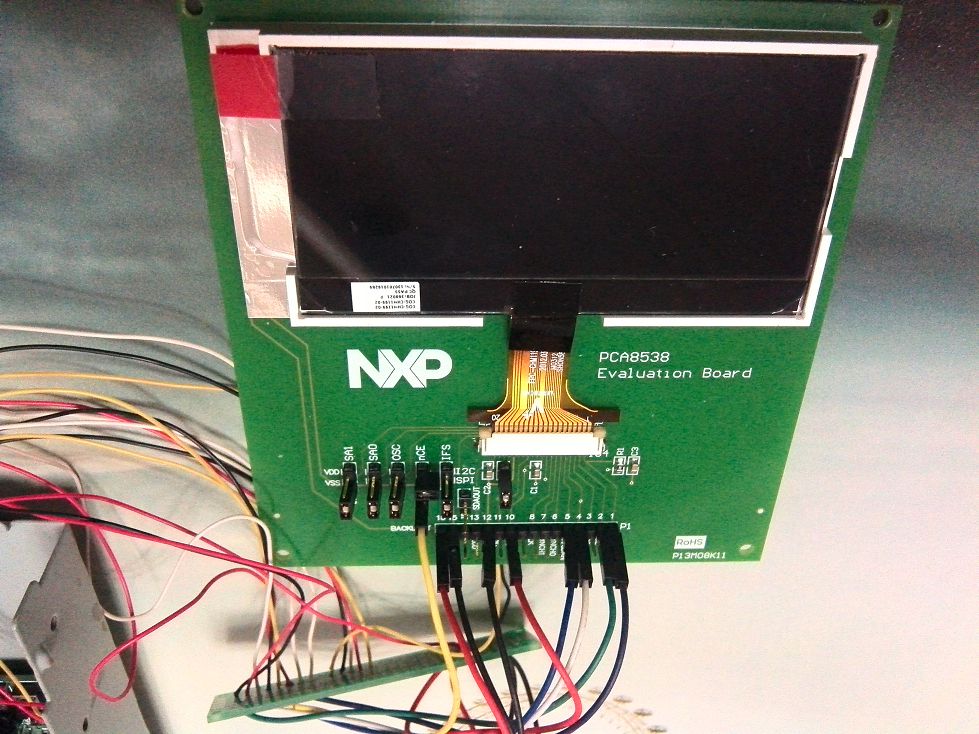
10 VDD------------------VCC

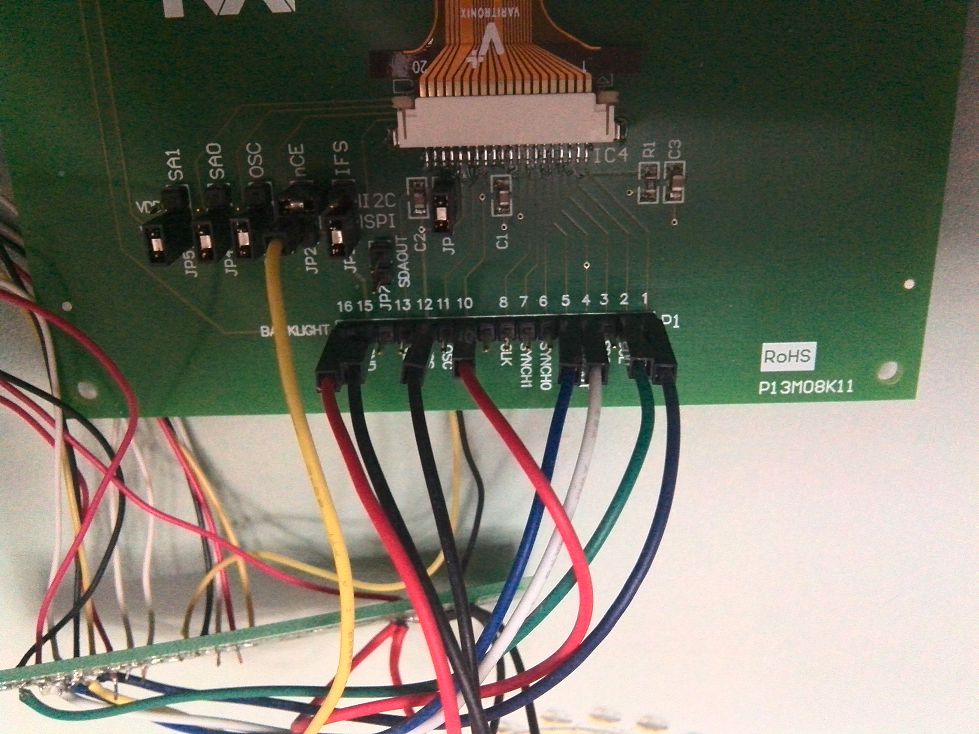
12 VSS-------------------GND

15 GND------------------GND

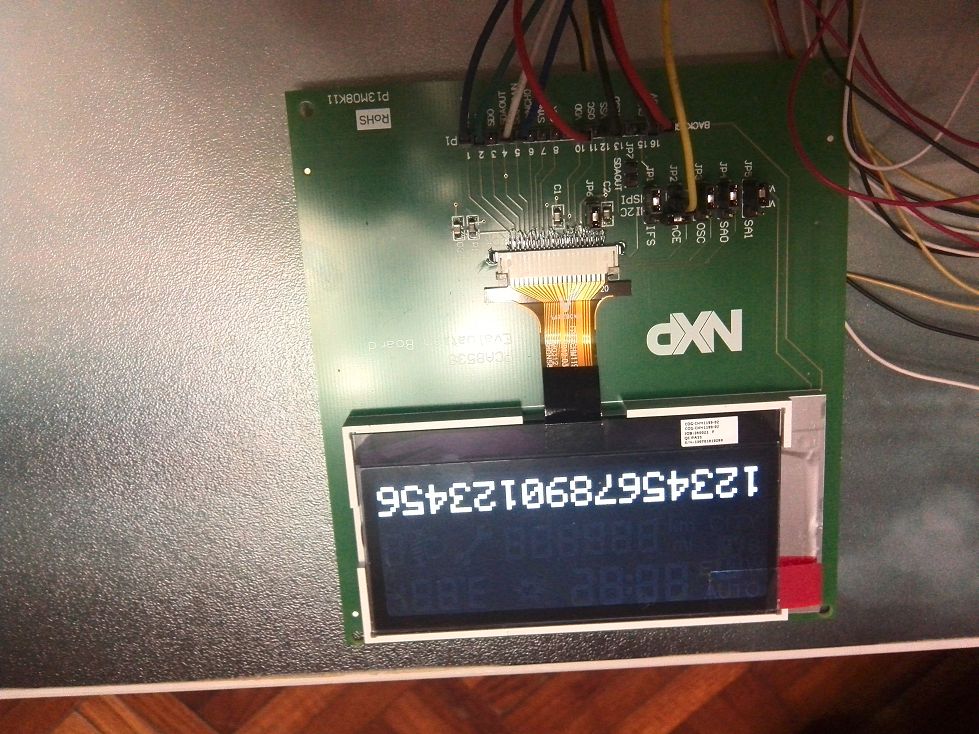
16 5V---------------------VCC

跳冒JP1——JP5接VSS，JP6短接，JP7不接





### 最大显示16个字符



### 显示测试电容

