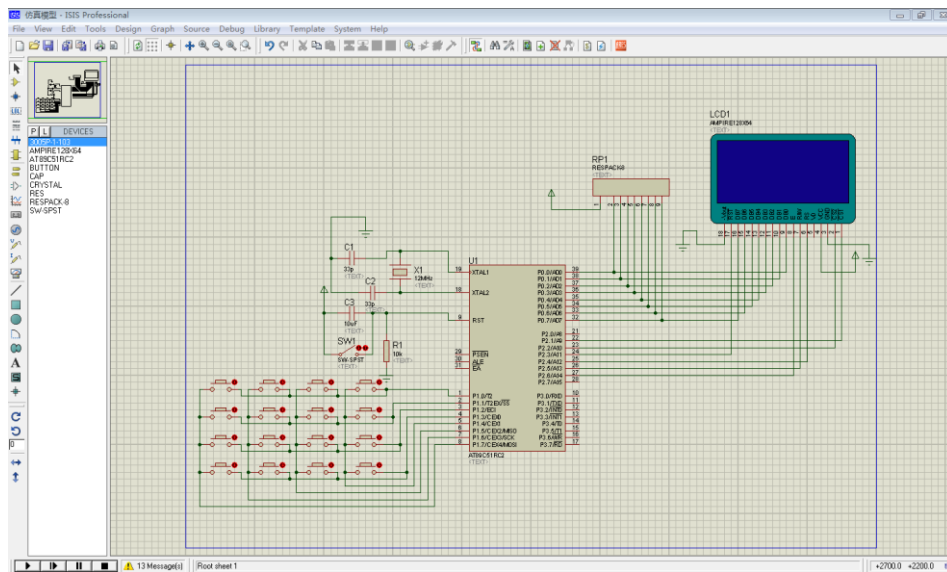


# 51 Matrix Calculator

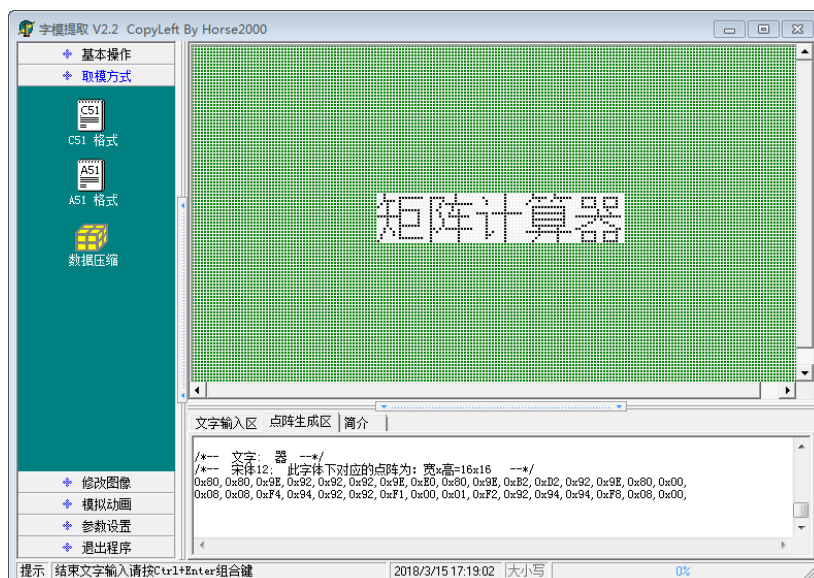
当初本科的毕业设计不是很完美，一直想把它完善一下，这学期刚开始有点时间，就以一个小项目的方式，综合自己这一年多来的代码阅读收获，重写了这个工程。

## 一，平台的重新搭建

- 在 win7 上重新安装了 proteus7.8 用于仿真阶段的测试，并从以前的资料中找到了仿真模型；

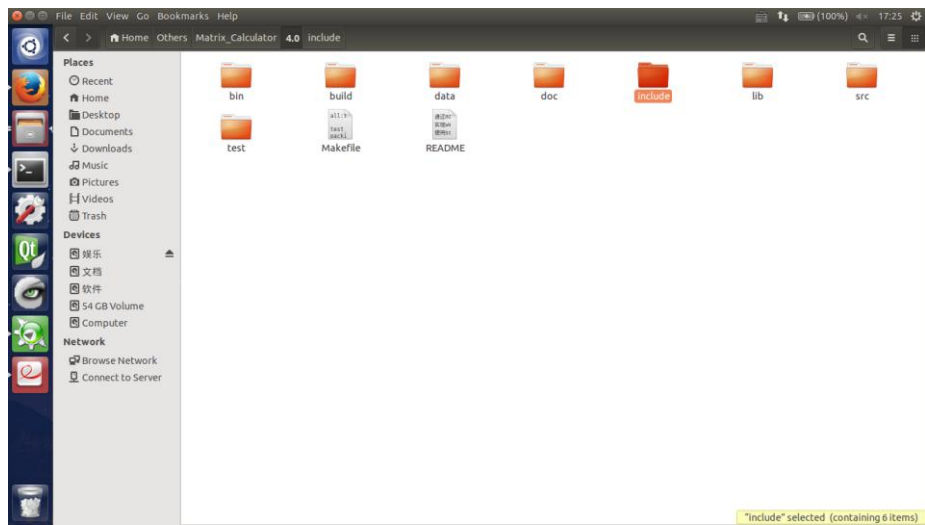


- 为了生成单片机可执行文件，选择使用了 Ubuntu 系统下的 sdcc 编译器，安装使用都很简单（想多在 Ubuntu 下进行编程），参考 [http://blog.csdn.net/adam\\_embedded/article/details/45310325](http://blog.csdn.net/adam_embedded/article/details/45310325)；
- 字模提取软件，用于各种字符码的提取；

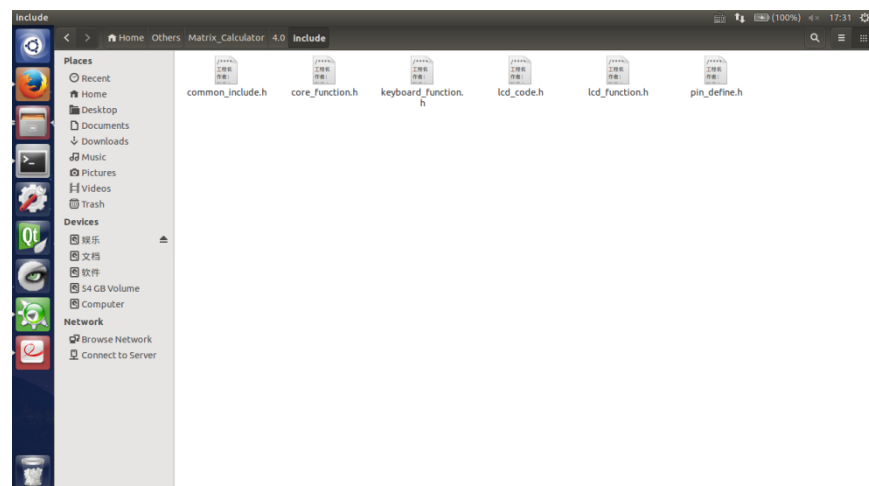
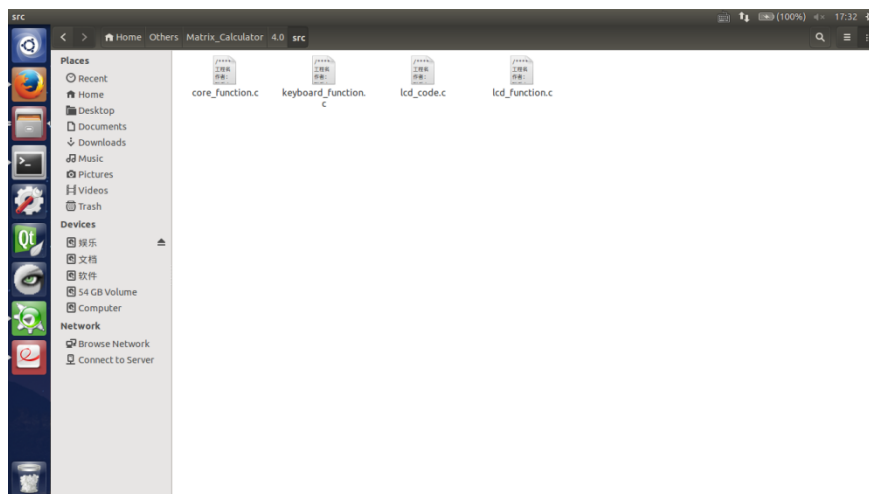


## 二，仿真代码重写

- 仿照之前见过的比较系统的工程目录，创建了自己的工程目录：



- 模仿 C++ 类的方式，按照不同的功能，将之前冗杂的单个工程文件分成了几个分开的功能文件：



- 每个.c实现不同类的功能：

keyboard\_function：

```
//初始化函数
extern void KeyBoard_Init();
//延迟函数
extern void KeyBoard_Delay(unsigned int t);
//键盘扫描函数，返回键值
extern unsigned char KeyBoard_Scan();
```

lcd\_function：

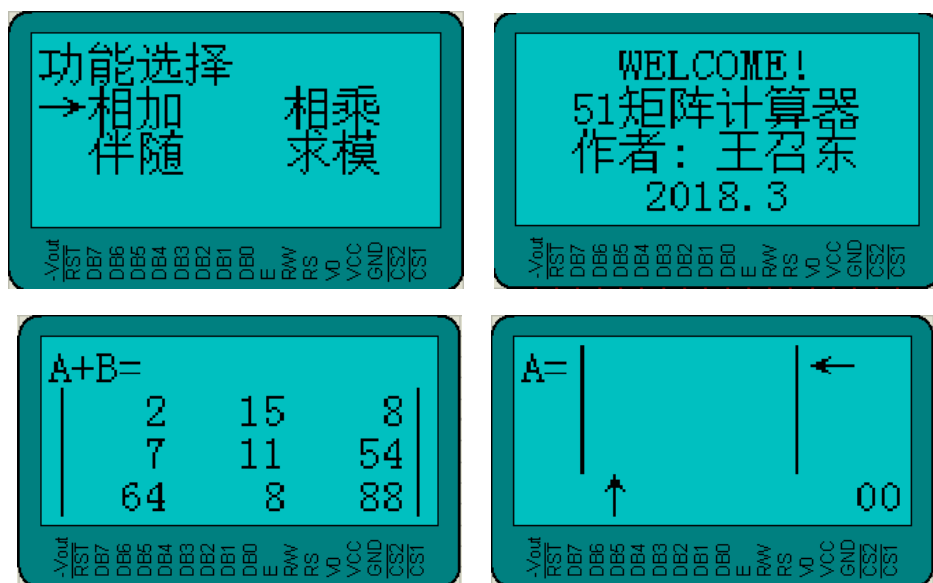
```
//延迟程序
extern void LCD_Delay(unsigned int t);
//状态检测程序
extern void LCD_CheckBusy();
//写指令函数
extern void LCD_WriteCode(unsigned char cmdcode);
//写数据函数
extern void LCD_WriteData(unsigned char cmddata);
//在page页，column列显示指定字符code，左后一个参数决定是全角还是半角
extern void LCD_Show(unsigned char page,unsigned char cols, unsigned char *code_table,int chinese);
//清屏
extern void LCD_Clear();
//初始化界面
extern void LCD_Welcome();
//初始化函数
extern void LCD_Init();
```

core\_function：

```
//三个函数处理所需的矩阵
extern int MatrixA[3][3];
extern int MatrixB[3][3];
extern int MatrixC[3][3];
//系统初始化函数
extern void SystemInit();
//功能显示界面函数
extern void FunctionMenu();
//功能选择函数
extern unsigned char FunctionSelect();
//右下角当前数字显示函数
extern void CurrentNumber( int numb );
//清箭头函数
extern void LocateArrow( int rows, int cols );
//将全局的矩阵变量清零
extern void ClearMatrix();
//矩阵输入函数
extern void InputMatrix( int (*matrix)[3][3], int number );
//矩阵显示函数
extern void ShowMatrix( int (*matrix)[3][3],int function);
//矩阵相加
extern void AddMatrix();
//矩阵相乘
extern void MultiplyMatrix();
//矩阵求逆
extern void InverseMatrix();
//矩阵求模
extern void ModuleMatrix();
//矩阵求伴随
extern void AdjointMatrix();
```

### 三，实验效果

完成了相加，相乘，求模，求转置的功能，求逆运算因为会牵扯到小数运算和显示，并未将其放入该工程，以下是主要的显示效果：



#### 四，项目缺陷

- 由于屏幕尺寸问题，局限在 3\*3 的矩阵计算上；
- 没有删除错误输入的清除功能（待实现）；
- 较小的 RAM 使得很容易内存不足，程序编写受到限制（待配合硬件加入外部存储器）；
- 单循环机制，没有有效的使用中断来避免各种意外情况（待将键盘扫面、功能选择等函数写入中断函数）；

2018.3.15