第13章 STC单片机ADC原理及实现

何宾

2018.03

交流电压测量和12864 LCD显示
--设计目标

本设计将从外部输入信号源，经过ADC转换器转换后，得到数字量的值，经过计算后，通过12864图形点阵LCD屏，一方面以字符显示得到的交流信号的最大值MAX、最小值MIN、峰峰值PTP（Peak to Peak）；另一方面以图形的方式显示采集交流信号的波形。

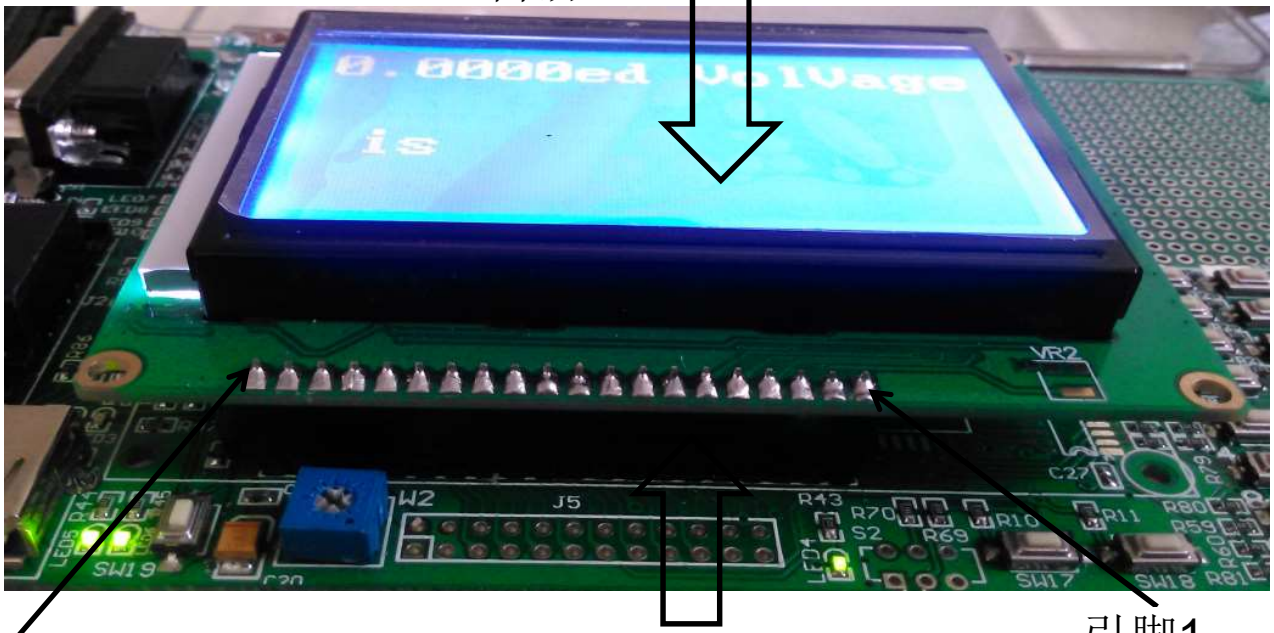
交流电压测量和12864 LCD显示
--硬件电路设计

在该设计中，12864图形/字符点阵屏通过自己焊接的插座与STC学习板上的标记为J12的单排插座连接。

- 在图中标出了STC学习板上插针引脚1的位置和12864图形/字符点阵屏引脚1的位置。

交流电压测量和12864 LCD显示
--硬件电路设计

12864图形/字符LCD点阵屏



引脚20

STC学习板上提供的标记为J12的单排插针和12864的单排插座插在一起

引脚1

交流电压测量和12864 LCD显示
--硬件电路设计

STC学习板上J12提供20个插针，可以直接与12864图形/字符LCD进行连接。

交流电压测量和12864 LCD显示
--硬件电路设计

STC学习板 J12插座引脚号	信号名字	与单片机引脚的连接关系	12864 图形点阵 LCD引脚号	信号名字	功能
1	GND	地	1	VSS	地
2	VCC	+5V电源	2	VCC	+5V电源
3	V0	--	3	V0	LCD驱动电压输入
4	RS	P2.5	4	RS	寄存器选择。RS=1，数据；RS=0，指令
5	R/W	P2.6	5	R/W	读写信号。R/W=1，读操作；R/W=0，写操作
6	E	P2.7	6	E	芯片使能信号
7	DB0	P0.0	7	DB0	8位数据总线信号
8	DB1	P0.1	8	DB1	
9	DB2	P0.2	9	DB2	
10	DB3	P0.3	10	DB3	
11	DB4	P0.4	11	DB4	
12	DB5	P0.5	12	DB5	
13	DB6	P0.6	13	DB6	
14	DB7	P0.7	14	DB7	
15	PSB	P2.4	15	PSB	并/串模式选择。PSB=1，并行；PSB=0，串行
16	N.C	P2.2	16	N.C	不连接
17	/RST	P2.3	17	/RST	复位，低电平有效
18	VOUT	--	18	VOUT	倍压输出脚
19	A	+5V电源	19	BLA	背光源正极，接+5.0V
20	K	地	20	BLK	背光源负极，接地

交流电压测量和12864 LCD显示
--12864图形点阵LCD指标

12864是指LCD的屏幕分辨率为128×64。

- **12864中文汉字图形点阵液晶显示模块，可显示汉字及图形，内置8192 个中文汉字（16×16 点阵）、128个字符（8×16 点阵）及64X256 点阵显示RAM（GDRAM）。JGD12864图形点阵LCD的特性指标。**

12864图形点阵**LCD**主要技术参数

显示容量	128×64个像素。每屏可显示4行8列共32个16×16点阵的汉字，或者4行16列共64个ASCII字符
工作电压范围	4.5V~+5V。对于STC单片机来说，推荐5.0V给12864供电
显示颜色	黄绿/蓝屏/灰屏
LCD类型	STN
与MCU接口	8/4位并行，或者3位串行
屏幕尺寸	93×70×12.5mm（长×宽×高）
多重模式	显示光标、画面移动、自定义字符、睡眠模式等

交流电压测量和12864 LCD显示
--12864图形点阵LCD内部存储空间

在介绍下面内容前，对12864涉及到的存储空间进行简单的说明：

- 数据显示RAM
  - 即：Data Display Ram ，DDRAM。往里面写什么数据，LCD就会显示写入的数据。
- 字符发生ROM
  - 即：Character Generation ROM ，CGROM。里面存储了中文汉字的字模，也称作中文字库，编码方式有GB2312（中文简体）和BIG5（中文繁体）。

交流电压测量和12864 LCD显示
--12864图形点阵LCD内部存储空间

- 字符发生RAM
  - 即：Character Generation RAM， CGRAM。 12864内部提供了64×2字节的CGRAM，可用于用户自定义4个16×16字符，每个字符占用32个字节。
- 图形显示RAM
  - 即：Graphic Display RAM，GDRAM。这一块区域用于绘图，往里面写什么数据，12864屏幕就会显示相应的数据，它与DDRAM的区别在于，往DDRAM中写的数据是字符的编码，字符的显示先是在CGROM中找到字模，然后映射到屏幕上，而往GDRAM中写的数据时图形的点阵信息，每个点用1比特来保存其显示与否。

交流电压测量和12864 LCD显示
--12864图形点阵LCD内部存储空间

- 半宽字符发生器
  - 即：Half height Character Generation ROM，HCGROM。就是字母与数字，也就是ASCII码。
  - 12864内部有4行×32字节的DDRAM空间。但是某一时刻，屏幕只能显示2行×32字节的空间，剩余的这些空间呢可以用于缓存，在实现卷屏显示时就可以利用这些空间。DDRAM结构如下所示：

交流电压测量和12864 LCD显示
--12864图形点阵LCD内部存储空间

DDRAM结构如下所示：

80H、81H、82H、83H、84H、85H、86H、87H、88H、89H、8AH、8BH、8CH、8DH、8EH、8FH

90H、91H、92H、93H、94H、95H、96H、97H、98H、99H、9AH、9BH、9CH、9DH、9EH、9FH

A0H、A1H、A2H、A3H、A4H、A5H、A6H、A7H、A8H、A9H、AAH、ABH、ACH、ADH、AEH、AFH

B0H、B1H、B2H、B3H、B4H、B5H、B6H、B7H、B8H、B9H、BAH、BBH、BCH、BDH、BEH、BFH

地址与屏幕显示对应关系如下：

12864第一行：80H、81H、82H、83H、84H、85H、86H、87H

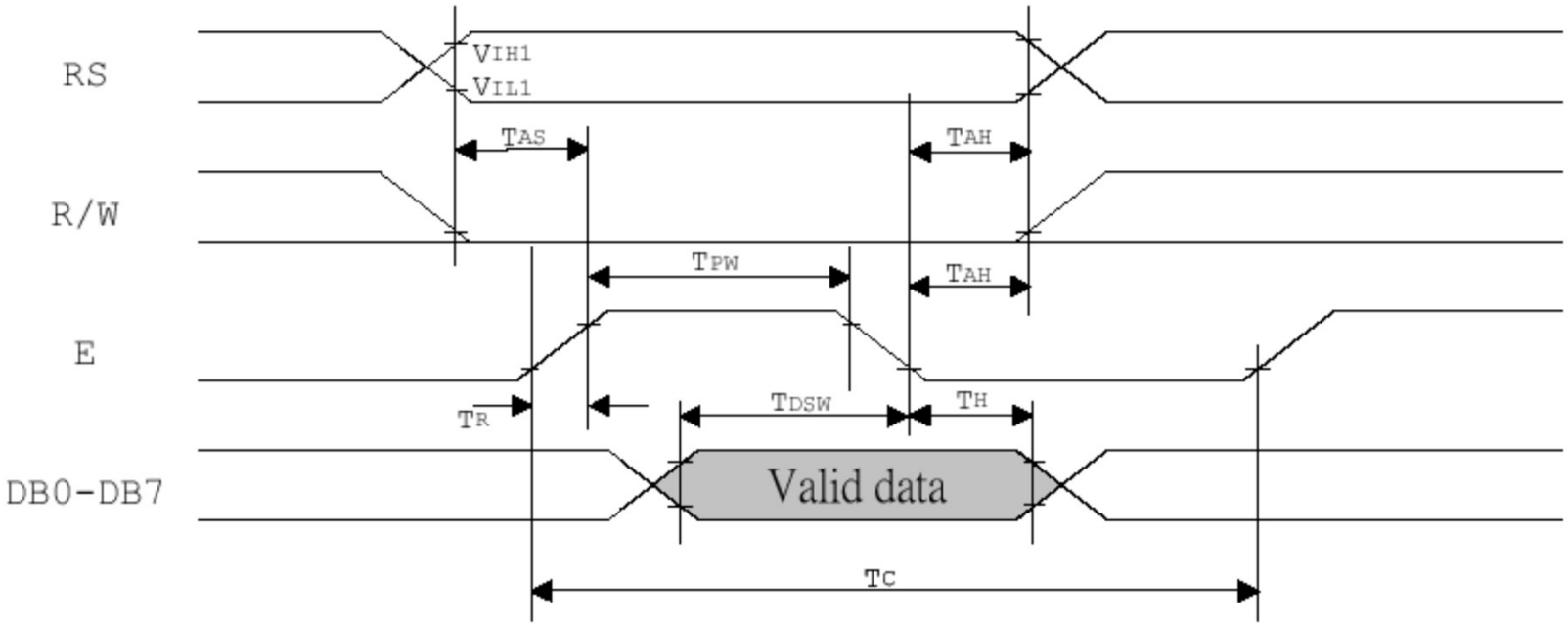
12864第二行：90H、91H、92H、93H、94H、95H、96H、97H

12864第三行：88H、89H、8AH、8BH、8CH、8DH、8EH、8FH

12864第四行：98H、99H、9AH、9BH、9CH、9DH、9EH、9FH

交流电压测量和12864 LCD显示
--12864图形点阵LCD读写时序

写操作时序

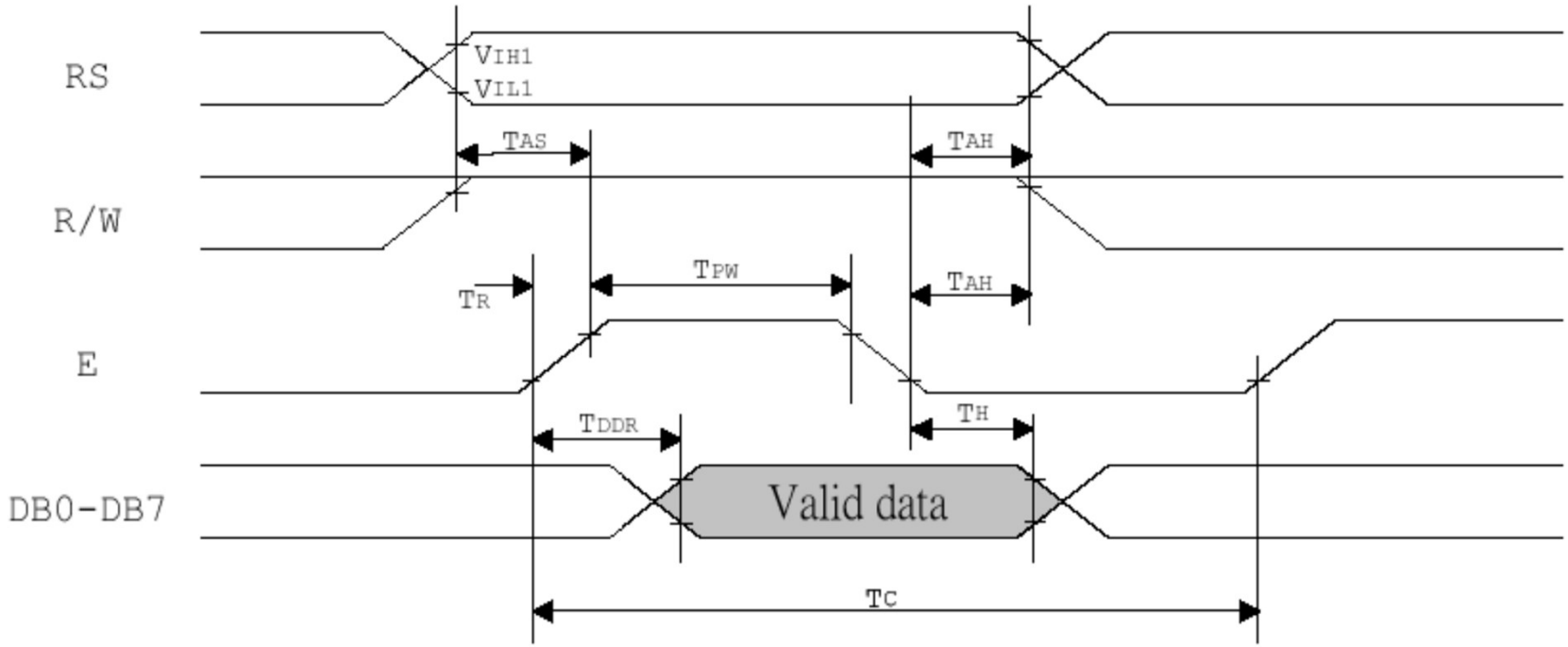


交流电压测量和12864 LCD显示
--12864图形点阵LCD读写时序

- 首先，将R/W信号拉低。同时，给出RS信号，该信号为1或者0，用于区分数据和命令。
- 然后，将E信号拉高。当E信号拉高后，STC单片机将写入12864图形点阵 LCD的数据放在DB7~DB0数据线上。当数据有效一段时间T_{DSW}后，首先将E信号拉低。然后，数据再维持一段时间T_H。这样，数据就写到12864图形点阵LCD中。
- 最后，撤除/保持R/W信号。

交流电压测量和12864 LCD显示
--12864图形点阵LCD读写时序

读操作时序



交流电压测量和12864 LCD显示
--12864图形点阵LCD读写时序

- 首先，将R/W信号拉高。同时，给出RS信号，该信号为1或者0，用于区分数据和状态。
- 然后，将E信号拉高。当E信号拉高，并且延迟一段时间t_{DDR}后，12864图形点阵LCD将数据放在DB7~DB0数据线上。当维持一段时间t_{TH}后，将E信号拉低。
- 最后，撤除/保持R/W信号。

交流电压测量和12864 LCD显示
---12864图形点阵LCD命令和数据

- 在STC单片机对12864图形点阵LCD操作的过程中，会用到下面的命令

注：12864图形点阵LCD提供了基本命令和扩展命令。

12864图形点阵LCD基本命令（RE=0）

指令	指令操作码										功能
	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
清屏	0	0	0	0	0	0	0	0	0	1	将“20H”写到DDRAM，将DDRAM地址从AC（地址计数器）设置到“00”
光标归位	0	0	0	0	0	0	0	0	1	-	将DDRAM的地址设置为“00”，光标如果移动，则将光标返回到初始的位置。DDRRAM的内容保持不变
进入点设定	0	0	0	0	0	0	0	1	I/D	S	指定在读数据和写数据时，设定光标移动的方向以及指定显示的移位。 I=0，递减模式。I=1，递增模；S=0，关闭整个移动。S=1，打开整个移动；S=0，关闭整个移动；

交流电压测量和12864 LCD显示
---12864图形点阵LCD命令和数据

显示打开/关闭控制	0	0	0	0	0	0	1	D	C	B	设置显示（D），光标（C）和光标闪烁（B）打开/关闭控制。 D=0，显示关闭； D=1，打开显示； C=0，关闭光标； C=1，打开光标； B=0，关闭闪烁； B=1，打开闪烁；
光标或者显示移动	0	0	0	0	0	1	S/C	R/L	-	-	设置光标移动和显示移动的控制位，以及方向，不改变DDRAM数据 S/C=0，R/L=0，光标左移； S/C=0，R/L=1，光标右移； S/C=1，R/L=0，显示左移，光标跟随显示移动 S/C=1，R/L=1，显示右移，光标跟随显示移动
功能设置	0	0	0	0	1	DL	--	RE=0	-	-	DL=1（必须设置为1） RE=1，扩展指令集；RE=0，基本指令集

交流电压测量和12864 LCD显示
---12864图形点阵LCD命令和数据

设置CGRAM地址	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	在地址计数器中，设置CGRAM地址
设置DDRAM地址	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	在计数器中，设置DDRAM地址
读忙标志和地址计数器	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	读BF标志，知道LCD屏内部是否正在操作。也可以读取地址计数器的内容
将数据写到RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	写数据到内部RAM（DDRAM/CGRAM/IRAM/GDRAM）
从RAM读数据	1	1	D7	D6	D5	D4	D3	D2	D1	D0	从内部RAM（DDRAM/CGRAM/IRRAM/GDRAM）读取数据

交流电压测量和12864 LCD显示
---12864图形点阵LCD命令和数据

12864图形点阵LCD扩展命令（RE=1）

指令	指令操作码										功能
	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
待命模式	0	0	0	0	0	0	0	0	0	1	将“20H”写到DDRAM，将DDRAM地址从AC（地址计数器）设置到“00”
卷动地址或IRAM地址选择	0	0	0	0	0	0	0	0	1	SR	SR=1，允许输入垂直卷动地址；SR=0，允许输入IRAM地址
反白选择	0	0	0	0	0	0	0	1	R1	R0	选择4行中的任意一行做反白显示，并可决定是否 反白
休眠模式	0	0	0	0	0	0	1	SL	-	-	SL=1，脱离休眠模式；SL=0，进入休眠模式

交流电压测量和12864 LCD显示
---12864图形点阵LCD命令和数据

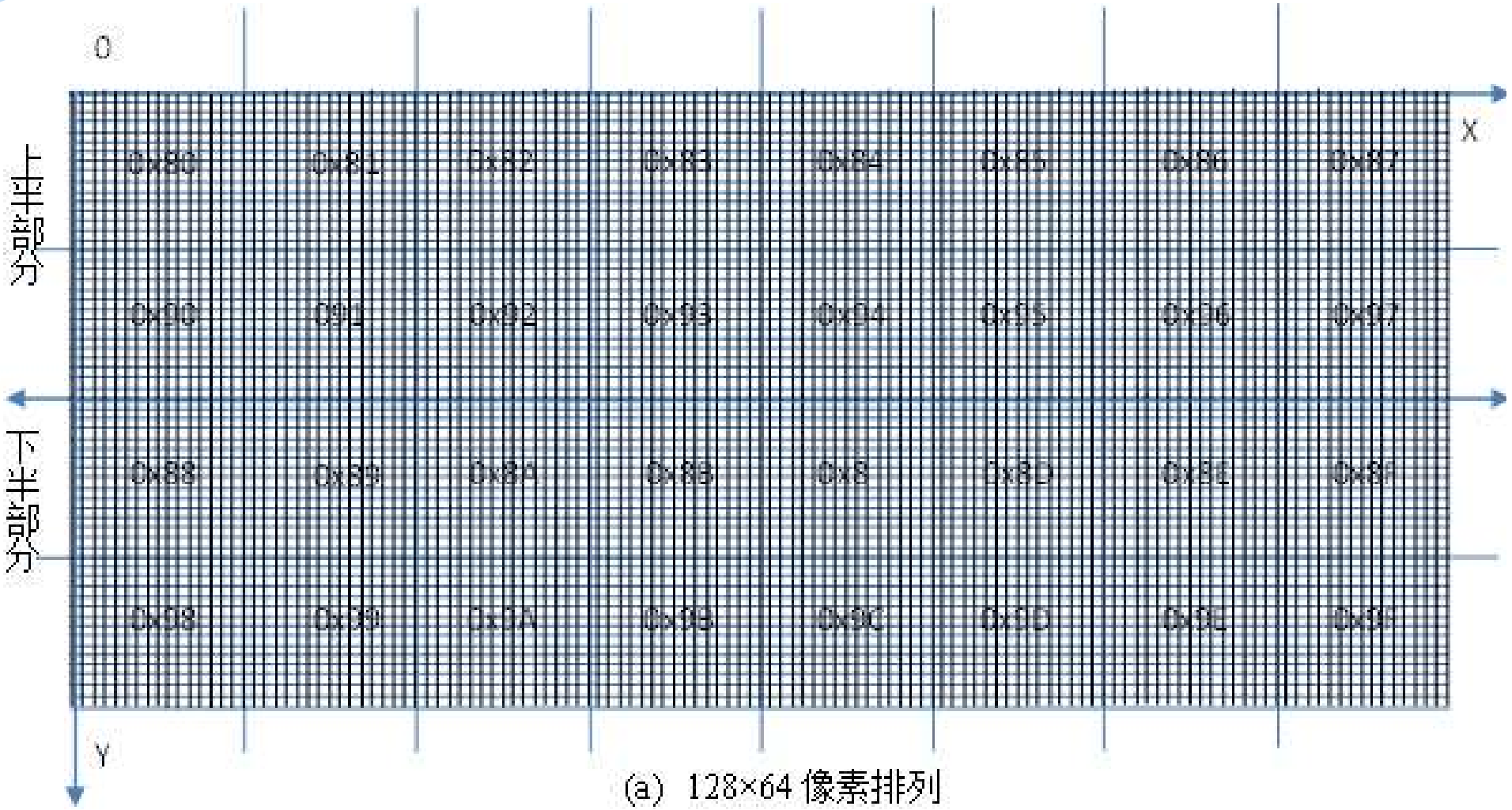
扩充功能设定	0	0	0	0	1	1	--	RE=1	G	0	RE=1，扩充指令集；RE=0，基本指令集 G=1，打开绘图模式；G=0，关闭绘图模式
设定IRAM地址或卷动地址	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	SR=1，AC5~AC0为垂直卷动地址 SR=0，AC3~AC0为ICON IRAM地址
设置绘图RAM地址	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	在地址计数器中，设置CGRAM地址

交流电压测量和12864 LCD显示
--12864图形点阵字符/图象表示方法

12864图形点阵128×64像素的构成方式。

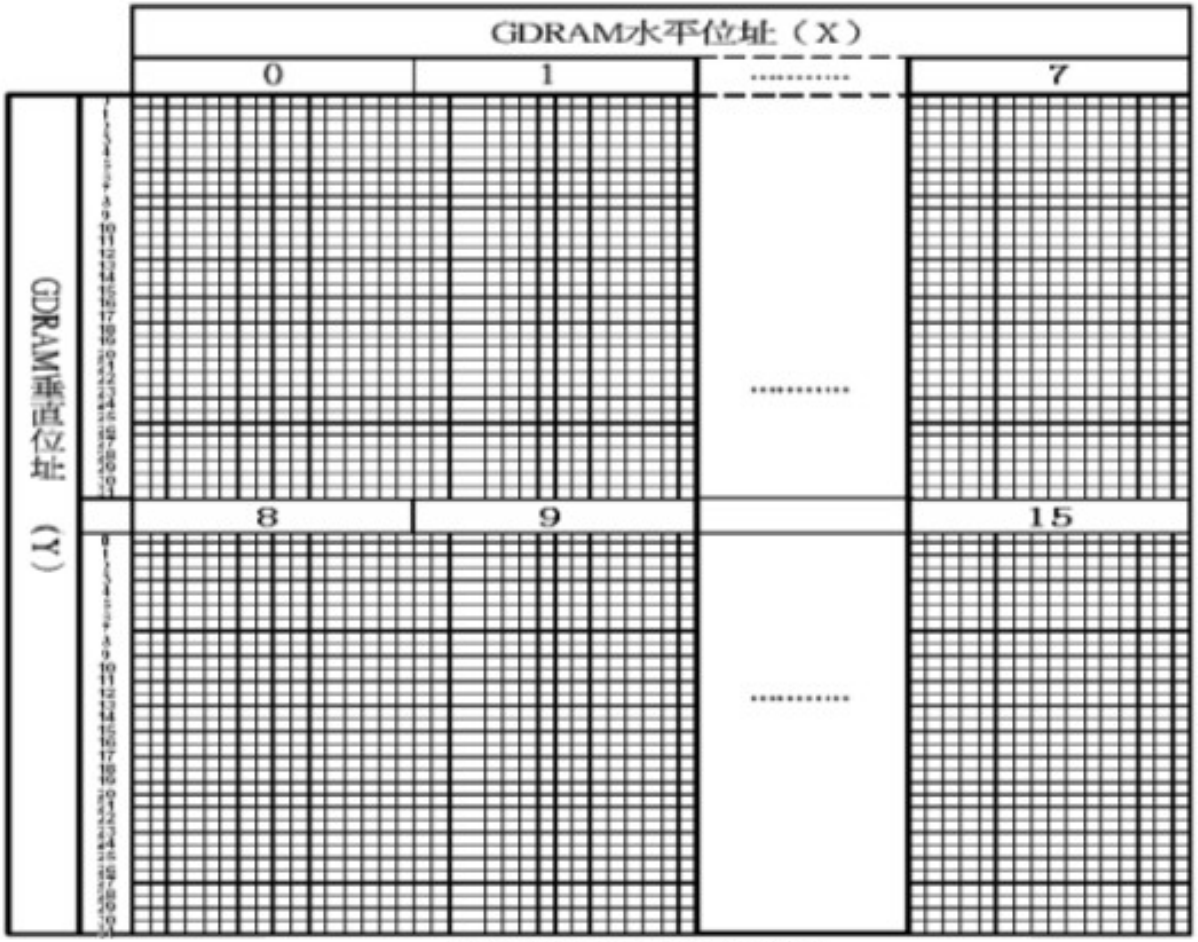
- 从后图(a)和(b)中可以看出，将屏幕分成上下部分，每部分包含32行和128列的像素。
- X方向以字节为单位，而Y方向以位为单位。X方向确定列，Y方向确定行。
- 该绘图显示RAM 提供128×8 个字节的存储空间。

交流电压测量和12864 LCD显示
--12864图形点阵字符/图象表示方法



(a) 128×64 像素排列

交流电压测量和12864 LCD显示
--12864图形点阵字符/图象表示方法



(b) CDRAM 地址分配情况

交流电压测量和12864 LCD显示
--12864图形点阵字符/图象表示方法

字符/汉字表示方法

- 在字符模式下，将12864图形点阵 分为4行8列一共32个区域。
  - 每个区域包含16×16个像素值，可以显示两个ASCII码字符/数字，或者一个汉字。
  - 每行起始地址分别为：0x80、0x90、0x88、0x98。
  - 每行可显示区域分配了8个地址，一个地址包含两个字节。
  - 从图中可以看出，只要从32个地址中选择一个地址，就可以确定所要显示的字母/数字/汉字的位置。

交流电压测量和12864 LCD显示
--12864图形点阵字符/图象表示方法

图像的表示方法

- 现在的问题就是给定了一个(x,y)坐标，如何向该坐标写数据。当向该坐标写0时，该点不亮；而向该坐标写1时，该点变亮。在更改绘图RAM的内容时，步骤包括：
  - 先连续写入水平与垂直的坐标值；
  - 再写入两个字节的数据到绘图RAM，而地址计数器（AC）会自动加一。

交流电压测量和12864 LCD显示
--12864图形点阵字符/图象表示方法

写入所有绘图RAM的步骤包括：

- 关闭绘图显示功能。
- 先将水平的位元组坐标X，即：所在上半部分/下半的行，写入绘图RAM地址。
- 再将垂直坐标Y，即：所在屏幕的上半部分还是下半部分，写入绘图RAM地址；
- 将D15~D8写入到RAM中；
- 将D7~D0写入到RAM中；
- 打开绘图显示功能。

交流电压测量和12864 LCD显示
--12864图形点阵字符/图象表示方法

在该设计中，在12864点阵中显示波形。策略是：

- 声明一个16×64大小的unsigned char类型数组，即：

  unsigned char pix[16][64];

  - 该声明的目的是，将128×64像素点分成16×64的区域。每个区域8列1行像素。8列正好是8位，一个无符号字节。因此，该数组可以表示12864图形点阵LCD中的所有128×64像素的当前状态。

交流电压测量和12864 LCD显示
--12864图形点阵字符/图象表示方法

将该pix数组所有元素赋初值为0，即128×64像素的当前状态都是0，不亮。很明显，在12864图形点阵LCD上画波形实际上就是让需要亮的像素点赋值为1即可。显示波形，包括正弦和三角波：

- 为了在LCD上显示一屏数据，因此每次得到128个采样，每个采样对应于x坐标；
- 每个采样所对应的离散的值是10位，范围在[0，1023]之间，经过量化处理，即：将该离散的值/16，范围限制在[0，63]之间。也就是说，每个采样的幅度在[0，63]。在该设计中，每个采样的幅度表示为y[i]，i=0~128。

交流电压测量和12864 LCD显示
--12864图形点阵字符/图象表示方法

- 下面得到每个采样和pix数组之间的对应关系。pix数组的每个元素的索引和y[i]存在下面的对应关系，即：

  pix[i/8][y[i]], i=0~128

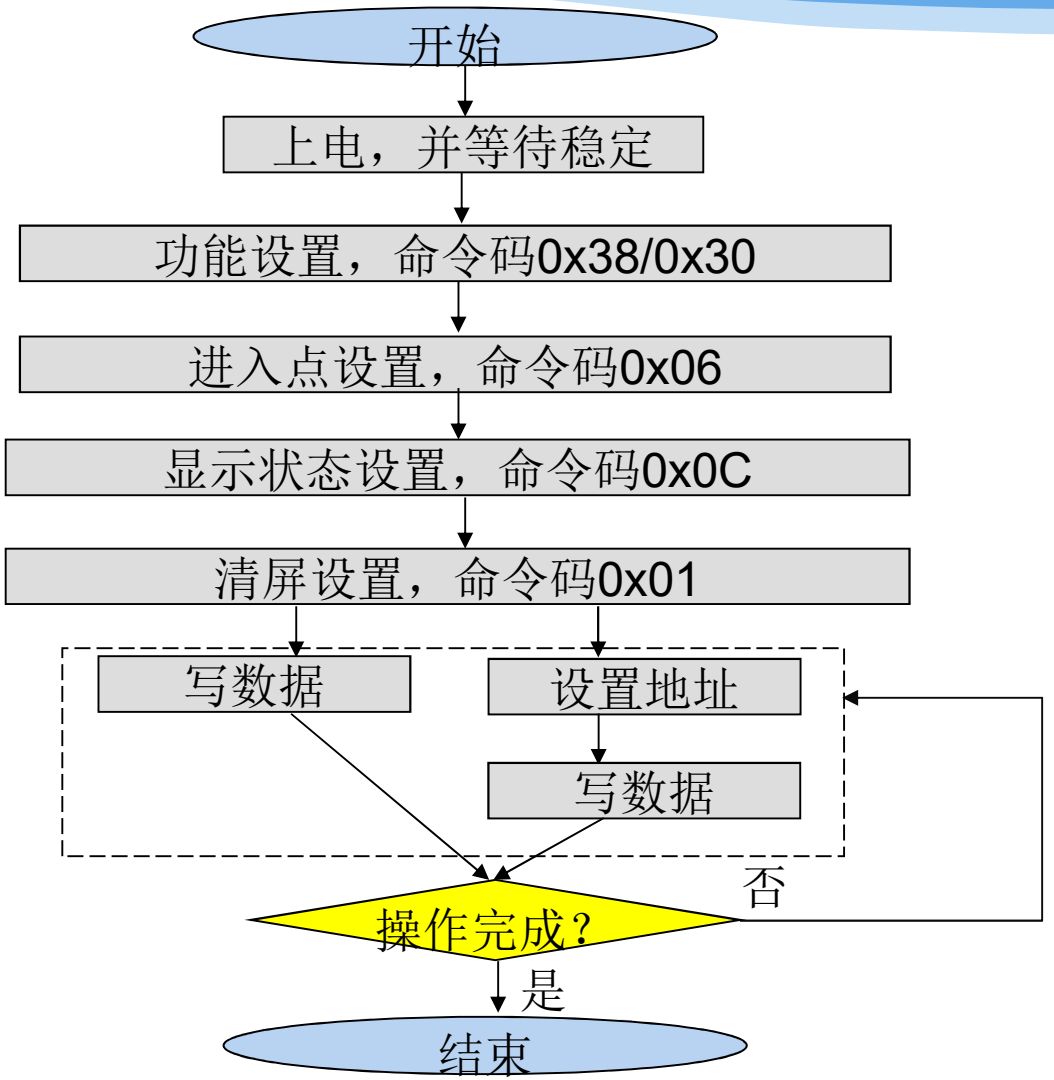
  - i/8将x坐标对应到16个区域中
  - y[i]是每个采样点经过量化处理后的幅度，很明显是pix数组中每个像素的行坐标
- 下面具体是pix数组的每个元素是unsigned char类型，也就是8个像素。表示为：

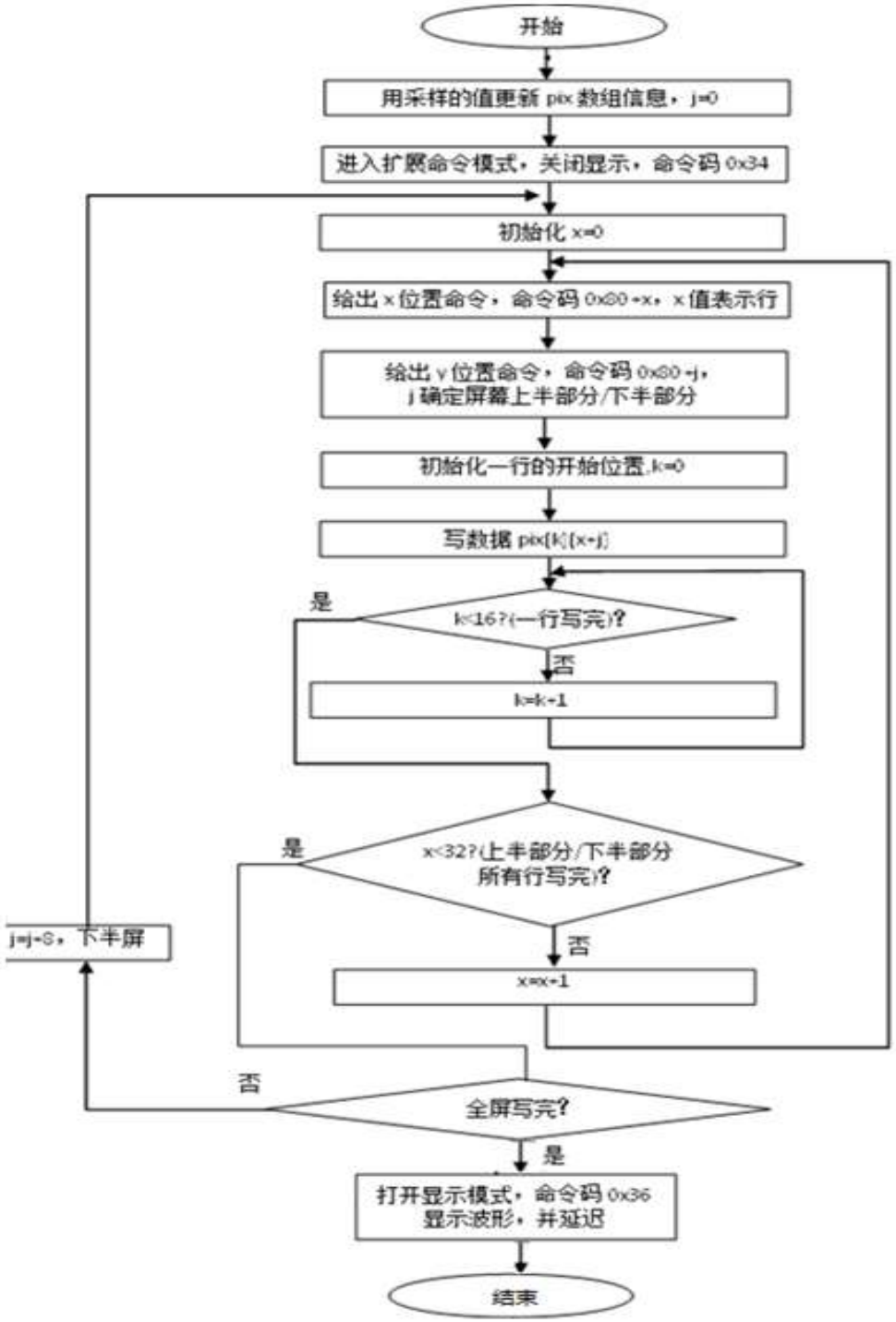
  pix[i/8][y[i]]=(0x80>>(i%8));

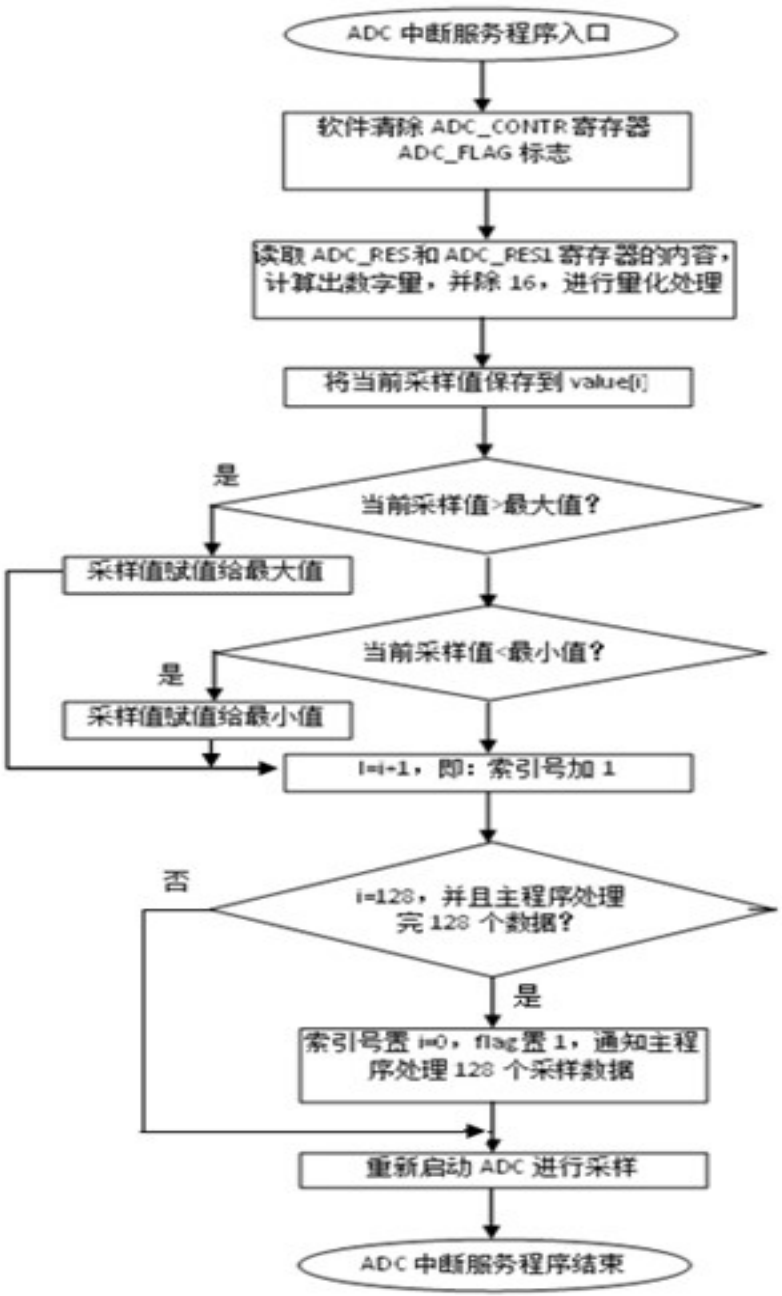
交流电压测量和12864 LCD显示
--12864图形点阵字符/图象表示方法

- (0x80>>(i%8))具体含义如下：
  - i%8得到每个x坐标i在pix数组每个区域i/8内的偏移位置。
  - 0x80表示一个8位的像素中，有一个像素是亮的，即：亮点。
  - (0x80>>(i%8))意思即表示：在x坐标i在pix数组每个区域i/8内的偏移位置上赋值为1。
- 重复步骤4，128次，将128个采样点的幅度具体表示在pix数组中。

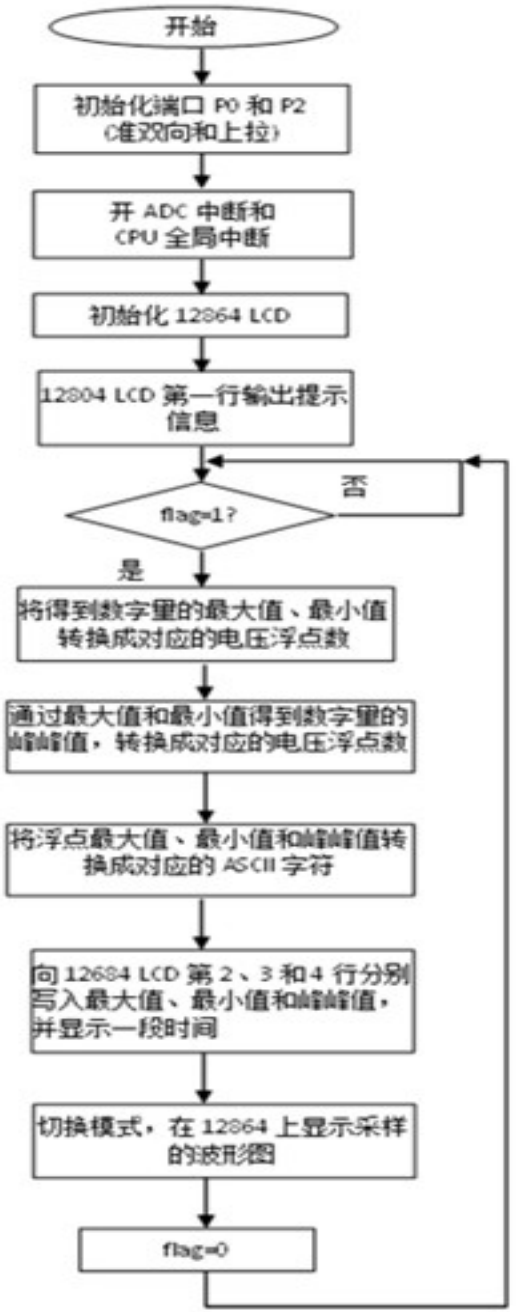
交流电压测量和12864 LCD显示
--12864 LCD字符模式初始化和显示字符操作过程




开始
用采样的值更新 pix 数组信息，j=0
进入扩展命令模式，关闭显示，命令码 0x34
初始化 x=0
给出 x 位置命令，命令码 0x80+x，x 值表示行
给出 y 位置命令，命令码 0x80+j，
j 确定屏幕上半部分/下半部分
初始化一行的开始位置 k=0
写数据 pix[k][x+j]
是
k<16?(一行写完)?
否
k=k+1
是
x<32?(上半部分/下半部分
所有行写完)?
否
x=x+1
否
全屏写完?
是
j=j+8，下半屏
打开显示模式，命令码 0x36
显示波形，并延迟
结束



（a）ADC 中断处理程序流程



（b）主程序处理流程

交流电压测量和12864 LCD显示
--ADC外部输入信号要求

由于STC15系列单片机是单电源供电，所以，其内部集成的ADC模块也是单电源供电。

- **典型地，供电电压是+5V。**
  - **因此，直接输入到ADC模块的信号范围在0~Vcc（单片机供电电压）范围内。**

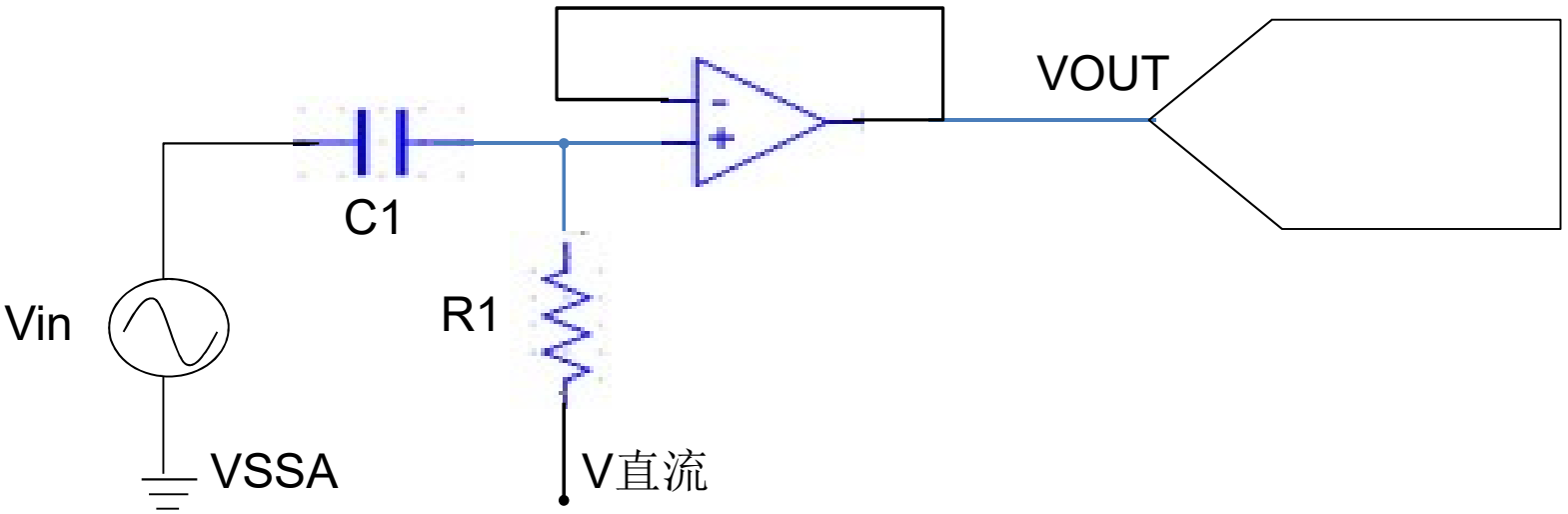
交流电压测量和12864 LCD显示
--ADC外部输入信号要求

输入信号采用的方式包括：

- 在信号源将信号直接接入到STC单片机ADC输入引脚时，在信号里给出直流偏置。
  - 交流信号在这个直流偏置信号上摆动，摆动的范围在0~5V单片机供电电压内，即：给出的信号是单极性的信号。

交流电压测量和12864 LCD显示
--ADC外部输入信号要求

- 信号源直流偏置为0，直接给出交流信号，这个交流信号经过下面的单电源放大器构成的电压跟随器的处理，引入直流偏置，然后将跟随器的输出信号Vout连接到STC单片机ADC的输入引脚上。



交流电压测量和12864 LCD显示
--ADC外部输入信号要求

- 当没有交流信号输入时，即信号的直流通路输出满足：

$$V_{out}' = V_{直流}$$

- 当有交流小信号输入时，信号的交流通路（直流偏置接地）输出满足：

$$V_{out}'' = V_{IN}$$

其中：

- 输入信号Vin的电压通过电阻R1降低到GND（地）。使用叠加定理，则总输出为：

$$V_{out} = V_{out}' + V_{out}'' = V_{IN} + V_{直流}$$

交流电压测量和12864 LCD显示
--具体实现过程

【例】采集外部交流信号，并在18624图形点阵LCD上显示的C语言描述的例子。

12864.h文件

```
#ifndef _12864_                  //条件编译命令，如果没有定义12864
#define _12864_                  //则定义12864
#include "reg51.h"               //包含reg51.h头文件
#include "intrins.h"             //包含intrins.h头文件
sbit LCD12864_RS=P2^5;           //声明sbit类型变量LCD12864_RS为P2.5引脚
sbit LCD12864_RW=P2^6;           //声明sbit类型变量LCD12864_RW为P2.6引脚
sbit LCD12864_E =P2^7;           //声明sbit类型变量LCD12864_E为P2.7引脚
sbit LCD12864_PSB=P2^4;          //声明sbit类型变量LCD12864_PSB为P2.4引脚
sfr  LCD12864_DB=0x80;          //声明LCD12864_DB寄存器地址0x80（P0端口）
sfr  P0M1=0x93;                  //定义P0端口P0M1寄存器地址0x93
sfr  P0M0=0x94;                  //定义P0端口P0M0寄存器地址0x94
```

交流电压测量和12864 LCD显示
--具体实现过程

```
sfr  P2M1=0x95;                                  //定义P2端口P2M1寄存器地址0x95
sfr  P2M0=0x96;                                  //定义P2端口P2M0寄存器地址0x96
void lcdwait();                                  //定义子函数lcdwait类型
void lcdwritecmd(unsigned char cmd);             //定义子函数lcdwritecmd类型
void lcdwritedata(unsigned char dat);            //定义子函数lcdwritedata类型
void lcdinit();                                  //定义子函数lcdinit类型
void lcdsetcursor(unsigned char x, unsigned char y);
                                                 //定义子函数lcdsetcursor类型
void lcdshowstr(unsigned char x, unsigned char y,unsigned char *str);
                                                 //定义子函数lcdshowstr类型
void drawpoint(unsigned char y[]);               //声明子函数drawpoint类型
#endif                                           //条件预编译命令结束
```

ADC应用实现3
--交流电压测量和12864 LCD显示

```
#include "12864.h"
void lcdwait()                              //定义lcdwait函数，用于检测12864的忙标志
{       LCD12864_DB=0xFF;                   //读取P0端口前，先给P0端口置位0xFF
        _nop_();
        _nop_();
        _nop_();
        _nop_();
        LCD12864_RS=0;                      //将LCD12864_RS指向的P2.5引脚拉低
        LCD12864_RW=1;                      //将LCD12864_RW指向的P2.6引脚拉高
        LCD12864_E=1;                       //将LCD12864_E指向的P2.7引脚拉高
        while(LCD12864_DB & 0x80);          //如果12864LCD内部忙，则等待
        LCD12864_E=0;                       //将LCD12864_E指向的P2.7引脚拉低
        _nop_();
        _nop_();
        _nop_();
        _nop_();
}
```

交流电压测量和12864 LCD显示
--具体实现过程

```
void lcdwritecmd(unsigned char cmd)
{           lcdwait();
            _nop_();
            _nop_();
            _nop_();
            _nop_();
            LCD12864_RS=0;          //将LCD12864_RS指向的P2.5引脚拉低
            LCD12864_RW=0;          //将LCD12864_RW指向的P2.6引脚拉低
            LCD12864_DB=cmd;//将命令码cmd放到LCD12864_DB 指向的P0端口
            LCD12864_E=1;           //将LCD12864_E指向的P2.7引脚拉高
```

交流电压测量和12864 LCD显示
--具体实现过程

```
    _nop_();
    _nop_();
    _nop_();
    _nop_();
     LCD12864_E=0;          //将LCD12864_E指向的P2.7引脚拉低
}
```

交流电压测量和12864 LCD显示
--具体实现过程

```
void lcdwritedata(unsigned char dat)
{
    lcdwait();
    _nop_();
    _nop_();
    _nop_();
    _nop_();
    LCD12864_RS=1;              //将LCD12864_RS指向的P2.5引脚拉高
   LCD12864_RW=0;               //将LCD12864_RW指向的P2.6引脚拉低
   LCD12864_DB=dat;             //将数据dat放到LCD12864_DB 指向的P0端口
```

交流电压测量和12864 LCD显示
--具体实现过程

```
    LCD12864_E=1;  //将LCD12864_E指向的P2.7引脚拉高
    _nop_();
    _nop_();
    _nop_();
    _nop_();
   LCD12864_E=0;          //将LCD12864_E指向的P2.7引脚拉低
}
```

交流电压测量和12864 LCD显示
--具体实现过程

```
void lcdinit()                    //定义子函数lcdinit，用于初始化12864
{
        lcdwritecmd(0x38);//调用函数lcdwritecmd，给12864发命令0x38
         lcdwritecmd(0x06);//调用函数lcdwritecmd，给12864发命令0x06
         lcdwritecmd(0x01);//调用函数lcdwritecmd，给12864发命令0x01
         lcdwritecmd(0x0c);//调用函数lcdwritecmd，给12864发命令0x0c
}
```

交流电压测量和12864 LCD显示
--具体实现过程

```
//声明lcdsetcursor函数，设置显示RAM的地址，x和y表示在12864的列和行参数
void lcdsetcursor(unsigned char x, unsigned char y)
{       unsigned char address;          //声明无符号char类型变量address
        if(y==0)                        //如果是第一行
                address=0x80+x;         //从存储器地址0x80的位置开始
        else if(y==1)                   //如果是第二行
                address=0x90+x;         //从存储器地址0x90的位置开始
        else if(y==2)                   //如果是第三行
                address=0x88+x;         //从存储器地址0x88的位置开始
```

交流电压测量和12864 LCD显示
--具体实现过程

```
        else                                    //如果是第四行
            address=0x98+x;                     //从存储器地址0x98的位置开始
            lcdwritecmd(address|0x80);          //写12864存储器地址命令
}
```

交流电压测量和12864 LCD显示
--具体实现过程

```
void lcdshowstr(unsigned char x, unsigned char y,unsigned char *str)
                                        //在12864上指定的x和y位置，显示字符
{
        lcdsetcursor(x,y);              //设置写RAM的地址
        while((*str)!='\0')             //如果字符没有结束
        {
                lcdwritedata(*str);//将当前字符写到RAM，即在12864上显示
                str++;                  //指向下一个字符
        }
}
```

交流电压测量和12864 LCD显示
--具体实现过程

```
void drawpoint(unsigned char y[])   //声明子函数drawpoint，在12864上显示波形
{
        unsigned char i,j,k;                    //定义无符号char类型变量i，j和k
        unsigned long int l;                    //定义无符号长整型变量l
        unsigned char x;                        //定义无符号char类型变量x
        xdata unsigned char pix[16][64];    //在xdata定义二维数组pix[16][64]
        for(i=0;i<16;i++)                       //二重循环初始化pix数组为0
          for(j=0;j<64;j++)
                pix[i][j]=0;
                    for(i=0;i<128;i++)//用采样的数据数组y[128]修改pix数组的值
                    pix[i/8][y[i]]=(0x80>>(i%8));//在给定的x和y像素位置上置1
```

交流电压测量和12864 LCD显示
--具体实现过程

```
for(i=0,j=0;i<9;i+=8,j+=32)                    //该循环确定屏幕上半部分和下半部分
{       for(x=0;x<32;x++)                      //该循环定位所在屏幕所在的行
        {       lcdwritecmd(0x34);             //使用扩展命令，关闭图像显示模式
                lcdwritecmd(0x80+x);           //写x坐标信息
                lcdwritecmd(0x80+i);           //写y坐标信息
                lcdwritecmd(0x30);             //使用基本命令
                for(k=0;k<16;k++);             //该循环连续写指定一行的128个像素
                lcdwritedata(pix[k][x+j]); //16列，每列8个像素
          }
}
lcdwritecmd(0x36);                             //打开显示模式
for(l=0;l<500000;l++);                          //显示图像并持续一段时间
}
```

交流电压测量和12864 LCD显示
--具体实现过程

main.c文件

```
#include "reg51.h"
#include "stdio.h"
#include "12864.h"
#define ADC_POWER    0x80          //定义ADC_POWER的值0x80
#define ADC_FLAG     0x10          //定义ADC_FLGA的值0x10
#define ADC_START    0x08          //定义ADC_START的值0x08
#define ADC_SPEEDLL  0x00          //定义ADC_SPEEDLL的值0x00
#define ADC_SPEEDL   0x20          //定义ADC_SPEEDL的值0x20
#define ADC_SPEEDH   0x40          //定义ADC_SPEEDH的值0x40
#define ADC_SPEEDHH  0x60          //定义ADC_SPEEDHH的值0x60
```

交流电压测量和12864 LCD显示
--具体实现过程

```
sfr AUXR =0x8E;                    //声明AUXR寄存器的地址0x8E
sfr ADC_CONTR =0xBC;               //声明ADC_CONTR寄存器的地址0xBC
sfr ADC_RES =0xBD;                 //声明ADC_RES寄存器的地址0xBD
sfr ADC_RESL =0xBE;                //声明ADC_RESL寄存器的地址0xBE
sfr P1ASF =0x9D;                   //声明P1ASF寄存器的地址0x9D
unsigned char ch=4;                //声明char类型变量ch
bit flag=1;                        //声明bit类型变量flag
unsigned char max_tstr[10],min_tstr[10],avg_tstr[10];
                                   //声明全局无符号char类型数组
unsigned int tmp=0;                //声明全局无符号int类型变量tmp
xdata unsigned char value[128];    //xdata区域声明无符号char类型数组value
unsigned int max_value=0 ,min_value=1024,avg_value=10;
                                   //声明全局无符号int类型变量
unsigned char inc=0;               //声明全局无符号char类型变量inc
```

交流电压测量和12864 LCD显示
--具体实现过程

```
void adc_int() interrupt 5                    //声明ADC中断服务程序
{
    unsigned char i=0;                        //定义无符号char类型变量i
    ADC_CONTR &=!ADC_FLAG;                    //清除ADC_FLAG标志
    tmp=(ADC_RES*4+ADC_RESL);                 //得到输入模拟量对应的10位转换数字量
    if(inc!=128 && flag==0)                   //如果没有采够128个数据，并且flag=0
    {
        value[inc]=tmp/16;                    //数字量除16，量化到0~64用于将来显示
        if(tmp>max_value)                     //如果当前采样的数字量大于最大值的数字量
            max_value=tmp;                    //将当前采样的数字量赋值给最大值
        if(tmp<min_value)                     //如果当前采样的数字量小于最小的数字量
            min_value=tmp;                    //将当前采样的数字量赋值给最小值
        inc++;                                //索引号递增1
    }
```

交流电压测量和12864 LCD显示
--具体实现过程

```
    else                                //采满128个采样数据
        {
            inc=0;                      //索引号归零
            flag=1;                     //将flag位置1
        }
ADC_CONTR=ADC_POWER |ADC_SPEEDLL | ADC_START | ch;
}
```

交流电压测量和12864 LCD显示 --具体实现过程

```
void main()
{
    long unsigned int i;
    P0M0=0;                    //设置P0M0和P0M1寄存器，将P0端口
    P0M1=0;                    //设置为准双向/弱上拉
    P2M0=0;                    //设置P2M0和P2M1寄存器，将P2端口
    P2M1=0;                    //设置为准双向/弱上拉
    P1ASF=0xFF;                //将P1端口设置为模拟输入
    ADC_RES=0;                 //将ADC_RES寄存器清0
```

交流电压测量和12864 LCD显示
--具体实现过程

```
//配置ADC控制寄存器ADC_CONTR
    ADC_CONTR=ADC_POWER|ADC_SPEEDLL | ADC_START | ch;
    for(i=0;i<10000;i++);                    //延迟一段时间
    IE=0xA0;                                 //CPU允许响应中断，允许ADC中断
    lcdinit();                               //初始化12864
    lcdshowstr(0,0,"测量交流信号 ");  //在12864第一行打印信息
    for(i=0;i<600000;i++);                   //延迟一段时间
    lcdwritecmd(0x01);                       //给12684发清屏命令
```

交流电压测量和12864 LCD显示
--具体实现过程

```
while(1)                                    //无限循环
{
   if(flag==1)                              //如果flag标志为1
     {
          lcdinit();                        //初始化12864
          sprintf(max_tstr,"%+1.4f",(max_value*5.0)/1024);
                                            //将浮点最大值转换成字符串
          sprintf(min_tstr,"%+1.4f",(min_value*5.0)/1024);
                                            //将浮点最小值转换成字符串
           sprintf(avg_tstr,"%+1.4f",((max_value-min_value)*5.0)/1024);
                                            //将计算得到的浮点峰峰值转换成字符串
           max_value=0;                     //将max_value重新赋值为0
           min_value=1024;                  //将min_value重新赋值为1024
```

交流电压测量和12864 LCD显示
--具体实现过程

```
lcdshowstr(0,1,"MAX: ");        //在12684第二行的开头打印“MAX:”信息
lcdshowstr(2,1,max_tstr);       //在12684第二行继续打印采集信号的最大值
lcdshowstr(5,1," V");           //在12684第二行继续打印“V:”信息
lcdshowstr(0,2,"MIN: ");        //在12684第三行的开头打印“MIN:”信息
lcdshowstr(2,2,min_tstr);       //在12684第三行继续打印采集信号的最小值
lcdshowstr(5,2," V");           //在12684第三行继续打印“V:”信息
lcdshowstr(0,3,"PTP: ");        //在12684第四行的开头打印“PTP:”信息
lcdshowstr(2,3,avg_tstr);       //在12684第四行继续打印采集信号的峰峰值
lcdshowstr(5,3," V");           //在12684第四行继续打印“V:”信息
for(i=0;i<300000;i++);          //延迟一段时间
```

交流电压测量和12864 LCD显示
--具体实现过程

```
        lcdwritecmd(0x01);              //清屏
        drawpoint(value);               //绘制采集信号的128个值的波形图
        flag=0;                         //将flag标志置0
        }
    }
}
```

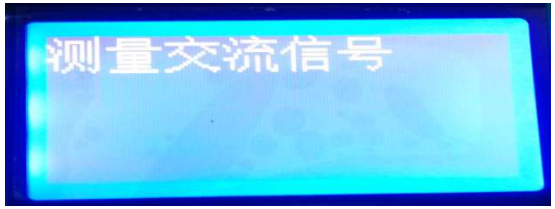
交流电压测量和12864 LCD显示
--测量和验证

下载和分析设计的步骤主要包括：

- **打开STC-ISP软件，在该界面内，选择硬件选项。将“输入用户程序运行时的IRC频率”设置为22.000MHz。**
- **单击下载/编程按钮，按前面的方法下载设计到STC单片机。**
- **打开信号源，将信号源的输出分别连接到STC开发板的P1.4和GND。**

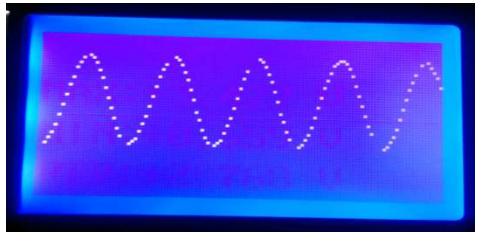
交流电压测量和12864 LCD显示
--测量和验证

- 观察12864图形点阵LCD上的输出，第一屏、第二屏和第三屏显示，如图所示。



12684输出的第一屏信息

12684输出的第二屏信息



12684输出的第三屏信息