

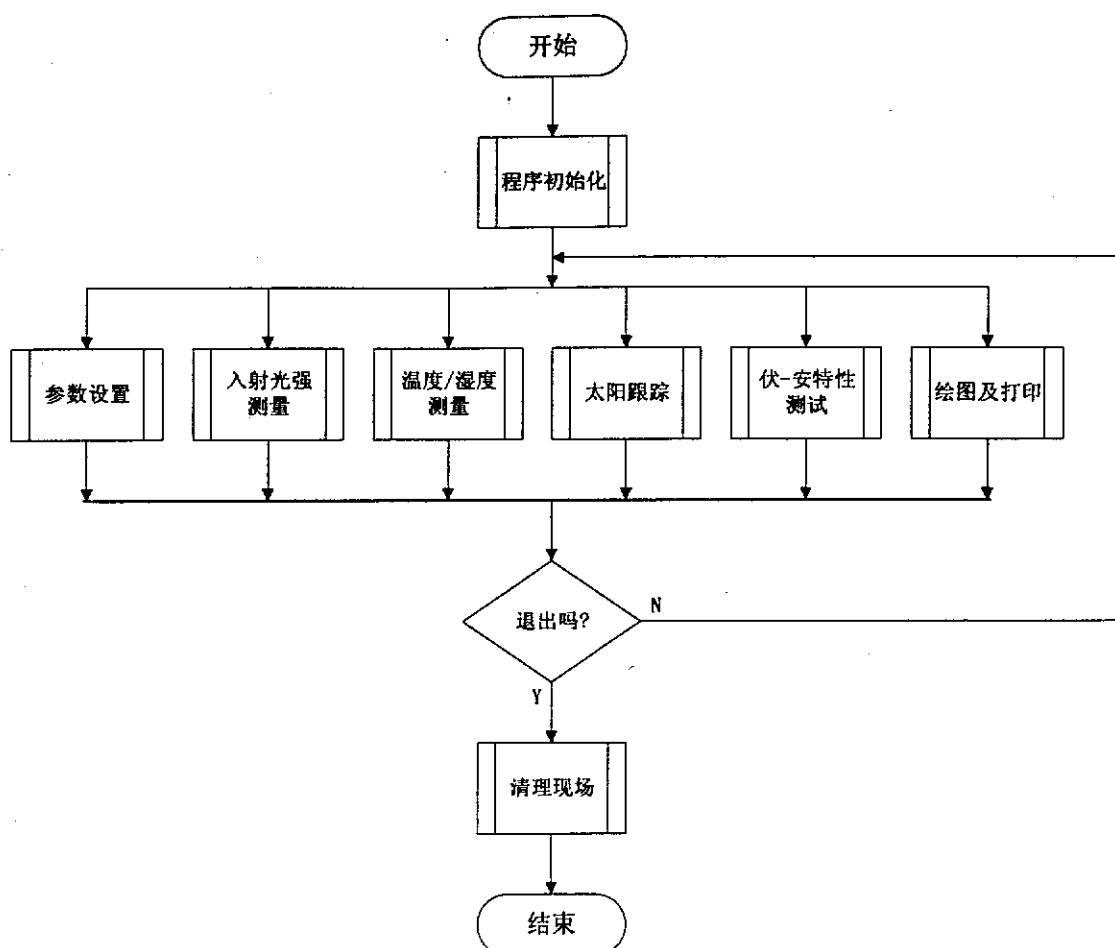


图 5-4 PC 机程序总流程图

该模块用来测量干、湿两个温度传感器通道的电压，得到干、湿温度，再根据温度的差值计算出湿度，将温、湿度值写入到参数文件中。

5. 太阳跟踪^{[12][18][24]}

太阳跟踪模块的流程图如图 5-5 所示。该模块内包含了通信协议的实现（详见 5.1 节）。跟踪参数和确认信号的发送、接收均通过该通信协议来进行。通过直接读写异步通讯芯片 8250（或其兼容芯片）的片内寄存器来实现对串行口的直接操作，控制字节发送、接收。

6. 伏-安特性测试

伏-安特性测试模块的流程图如图 5-6 所示。

为了提高测试速度，通过直接读写 PCL-812PG 多功能数据采集卡的内部寄存器来实现设置 A/D 增益、选择 A/D 通道、启动 A/D 采样、读取 A/D 结果、D/A 输出、数字量输入/输出等基本操作。

7. 绘图及打印

本模块用于将原始数据和测试结果以直观的形式表达出来。每次屏

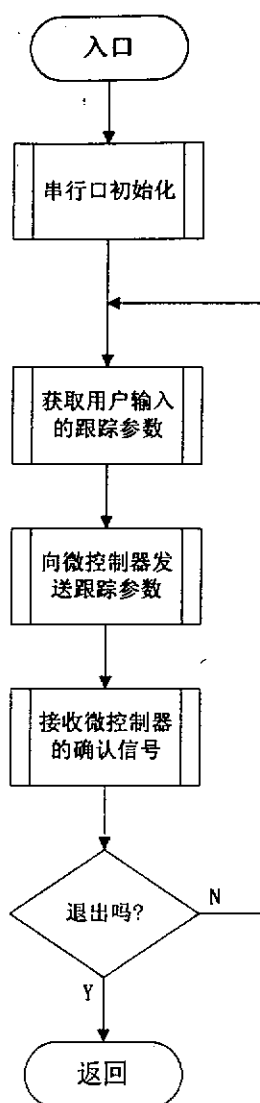


图 5-5 跟踪模块流程图

幕刷新时，本模块即被调用。

利用了 MFC 对打印功能的支持，使得屏幕输出可以方便地打印出来。绘图及打印模块的流程图如图 5-7 所示。

5.3 微控制器软件系统

一、Franklin C51 语言简介^{[24][25]}

Franklin C51 语言是美国 Franklin 公司推出的 8051 微控制器 C 语言编译器。它在代码生成方面领先，可产生最少的代码。

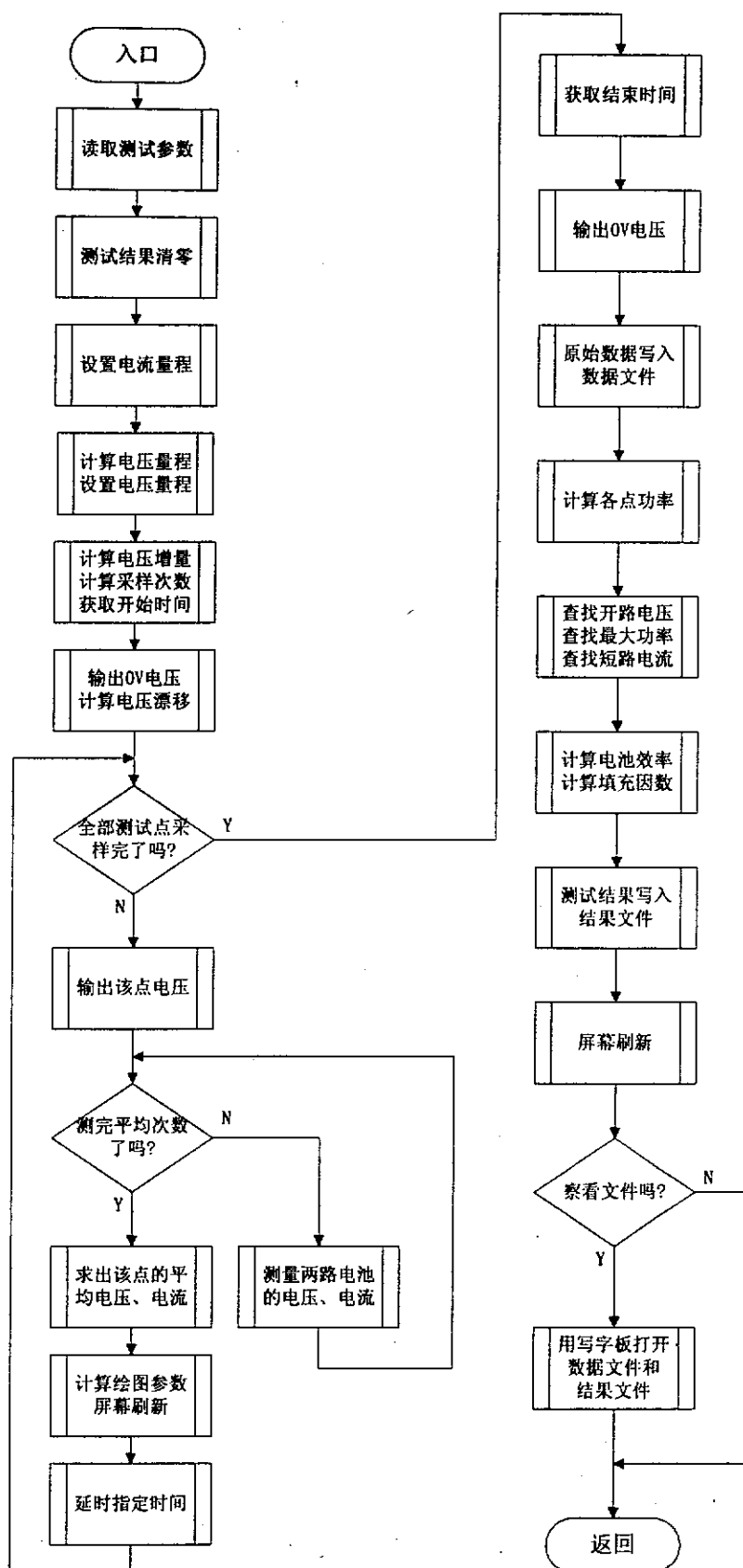


图 5-6 伏-安特性测试模块流程图

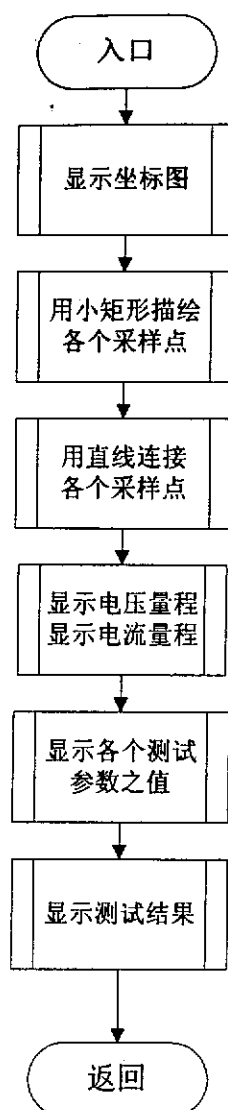


图 5-7 打印和绘图模块流程图

它支持浮点数和长整数、重入和递归。

它支持小、紧凑、大三种存储模式，可以访问所有的存储器单元。通过对标准 C 语言的扩充，可以用它来开发中断服务程序、定义位

变量、直接访问 8051 微控制器内的特殊功能位、特殊功能寄存器、片内外任意地址的存储器单元。可以用 interrupt 关键字开发中断服务程序。

Franklin C51 可以将即使是有经验的程序员编制的代码进行优化，其具有一般优化：常数折迭优化、跳转优化、死码消除优化、寄存器变量优化、通过寄存器传递参数优化、全局公共子式优化；基于 8051 的优化：窥孔优化、访问优化、数据覆盖优化、CASE/SWITCH 语句优化等优化功能。

它包含有完整的 ANSI 标准库函数。

Franklin C51 在很大程度上可以取代 8051 汇编语言。

二、微控制器软件设计

微控制器程序总流程图如图 5-8 所示。它主要由以下模块组成:

1. CPU 初始化

完成对中断的设置工作。

2. 定时器初始化

将定时器/计数器 0(T/C0)设置为 16 位计数器,可用于对外部事件计数;定时器/计数器 1(T/C1)设置为常数自动装入的 8 位定时器,作为串行口的波特率发生器。

3. 串行口初始化

设置串行通信为波特率为 1200,工作方式为方式 1(10 位异步收发)。

4. 步进电机控制

它用来发出步进电机驱动器所需的方向电平和步进脉冲。其流程图如图 5-9 所示。

程序首先检查参数的合法性,然后计算步进脉冲的周期,设置定时器的初值,然后发出相应的方向电平,并按照步进脉冲周期发出给定个数的步进脉冲。步进脉冲的高电平时间很短是为了满足步进电机驱动器的要求。

5. 通信协议^[24]

详见 5.1 节介绍。

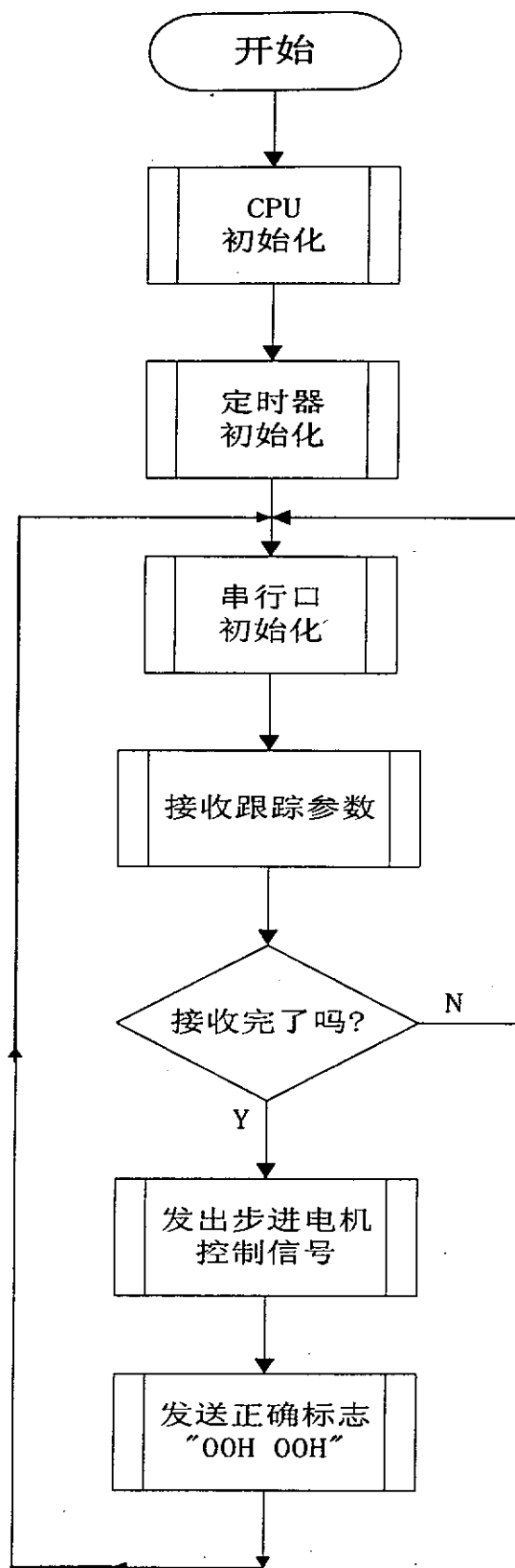


图 5-8 微控制器程序总流程图

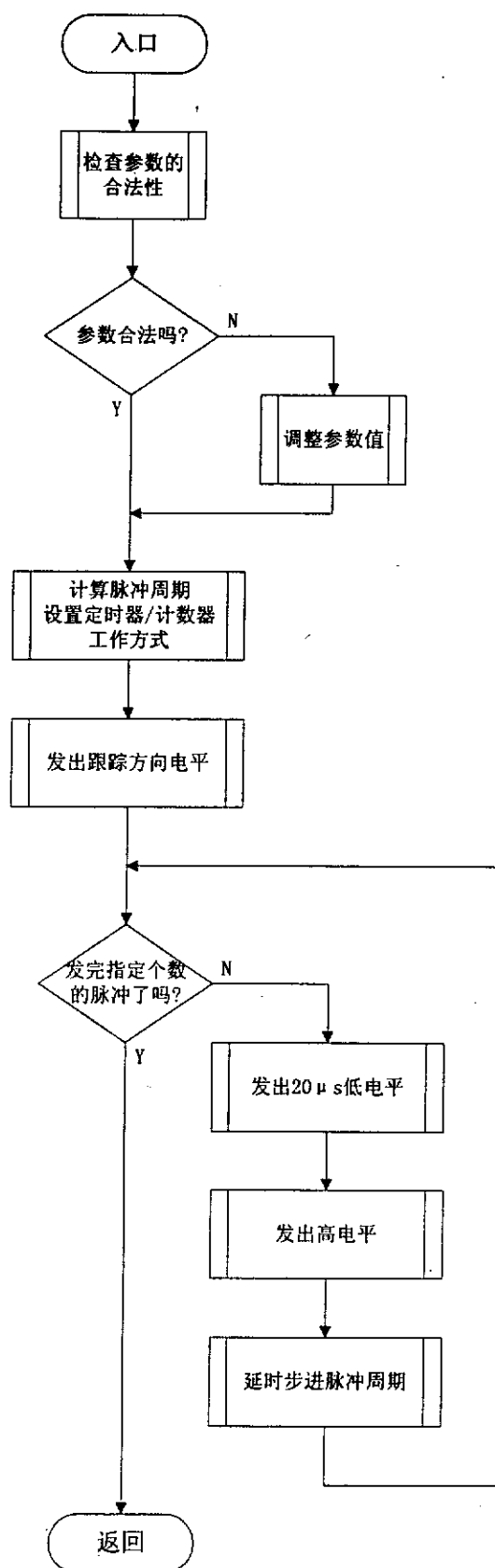


图 5-9 步进电机控制流程图

第六章 全文总结

6.1 测试结果分析

为了验证太阳电池测试系统的功能和获得测试误差, 分别采用纯电阻器和图 6-1 所示的电池模拟电路来测试伏安特性, 分别在 5mA、100mA、2A 三个有代表性的量程上进行。测试结果和误差特性曲线分别示于图 6-2 至图 6-13 中。

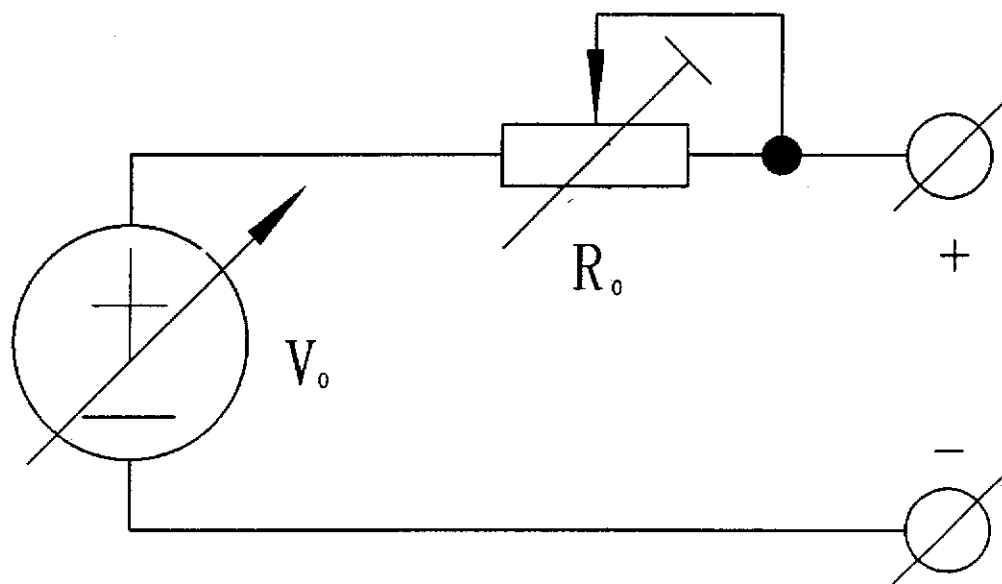


图 6-1 测量用的电池模拟电路

由实验结果可知, 无论是对于纯电阻器, 还是电池模拟电路, 所获得的伏安特性曲线均较平滑, 线性良好。

对于纯电阻器, 定义误差为:

$$\text{error} = (V(n) / I(n) - R_0) / R_0 * 100\% \quad (6-1)$$

其中 $V(n)$, $I(n)$ 分别为第 n 个采样点的电压值和电流值, R_0 为纯电阻器的电阻值。

误差曲线显示, 各个量程的误差均在 $\pm 1.5\%$ 以内 (其中 100mA 档在 $\pm 1.0\%$ 以内)。

测试电压较低时, 流经纯电阻器的电流值亦较低, 由于干扰的影响, 使得此时的相对误差较大。

对于电池模拟电路，定义误差为：

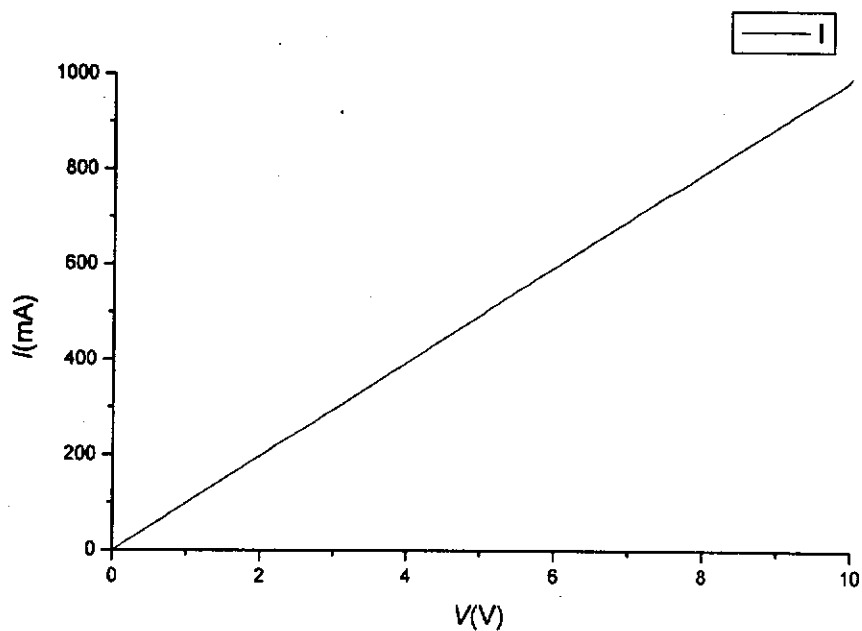
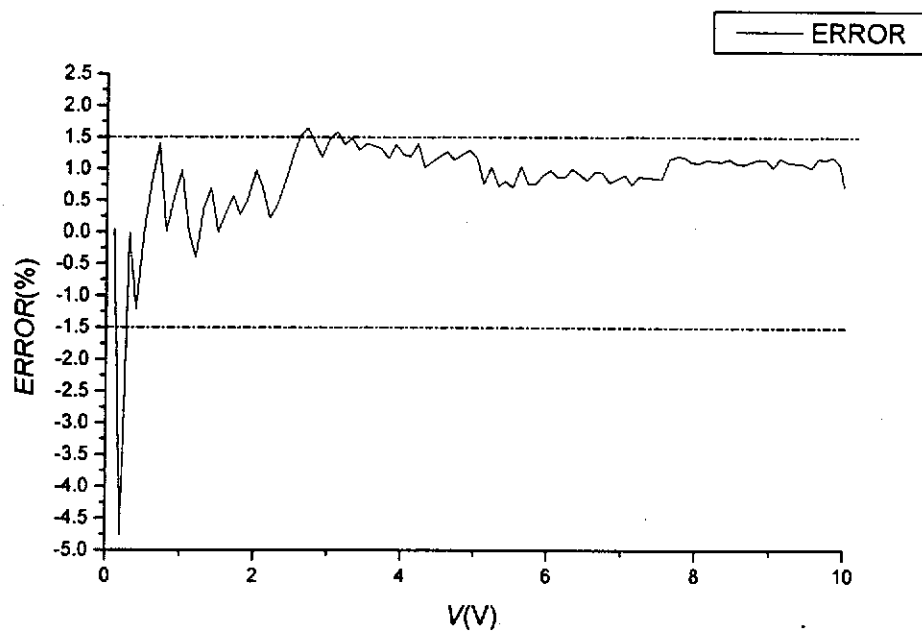
$$\text{error} = (I(n) - (V_0 - V(n)) / R_0) / I_{FS} * 100\% \quad (6-2)$$

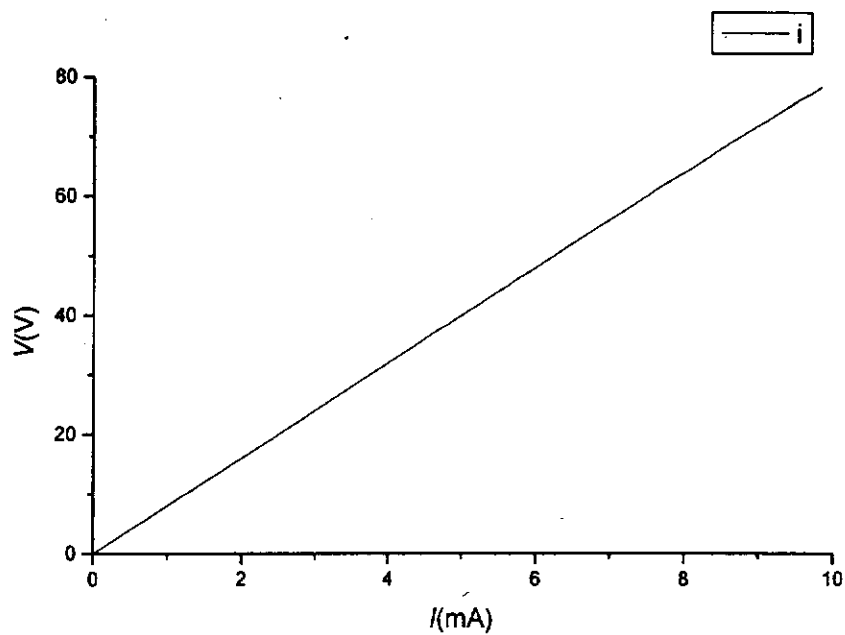
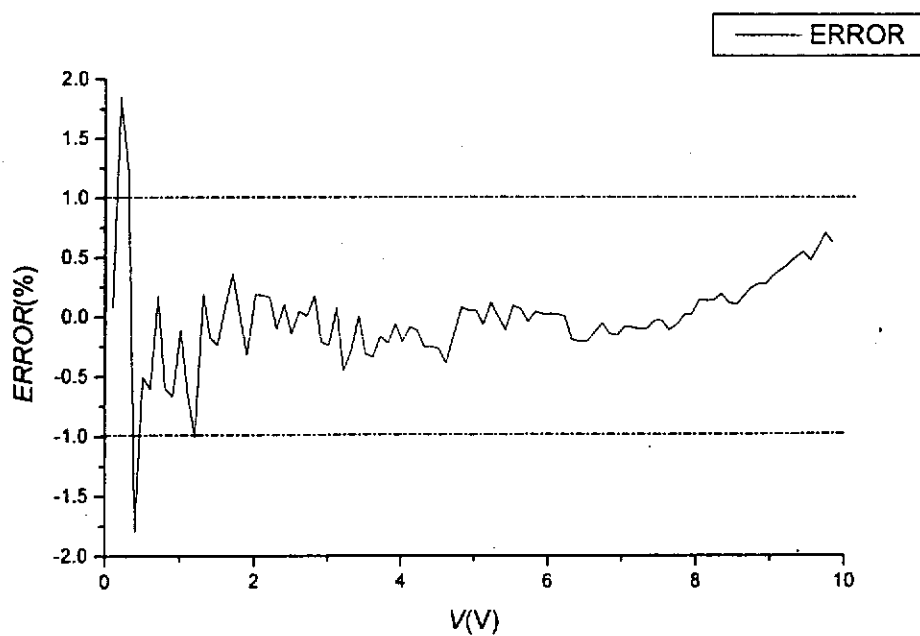
其中 $V(n)$ ， $I(n)$ 分别为第 n 个采样点的电压值和电流值， V_0 为电压源的电压值， R_0 为串联电阻器的电阻值， I_{FS} 为该电流量程的满度值。

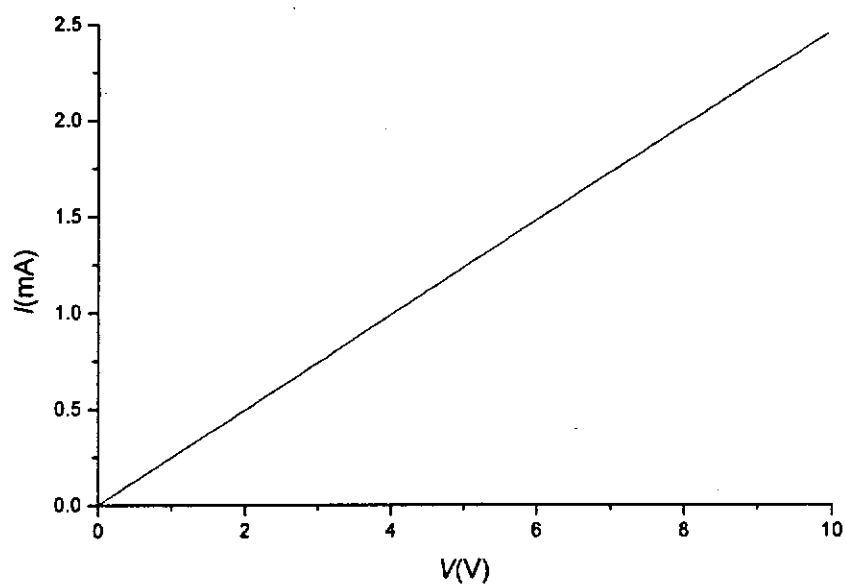
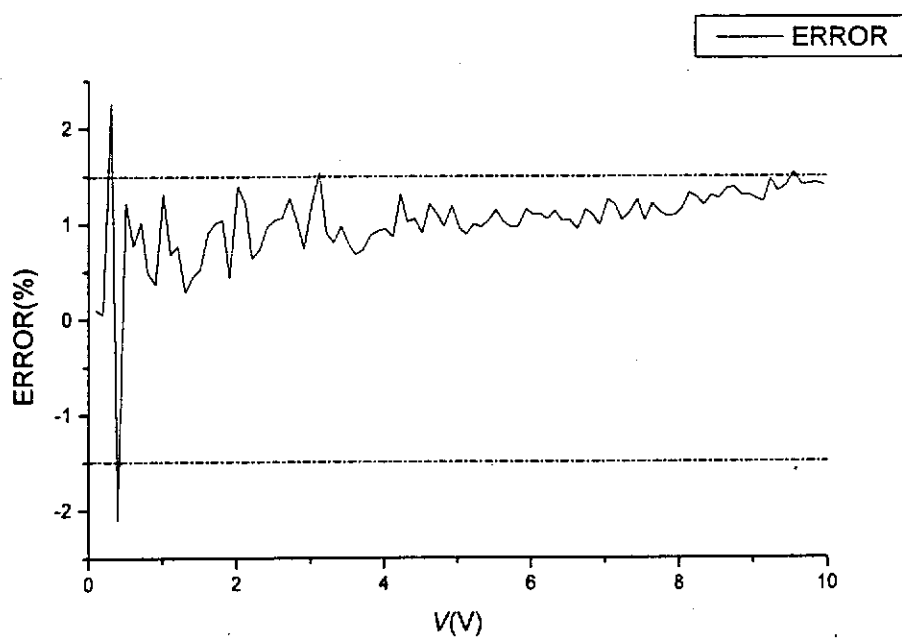
误差曲线显示，各个量程的误差均在 $\pm 1.5\%$ 以内（其中 100mA 档在 $\pm 0.25\%$ 以内）。

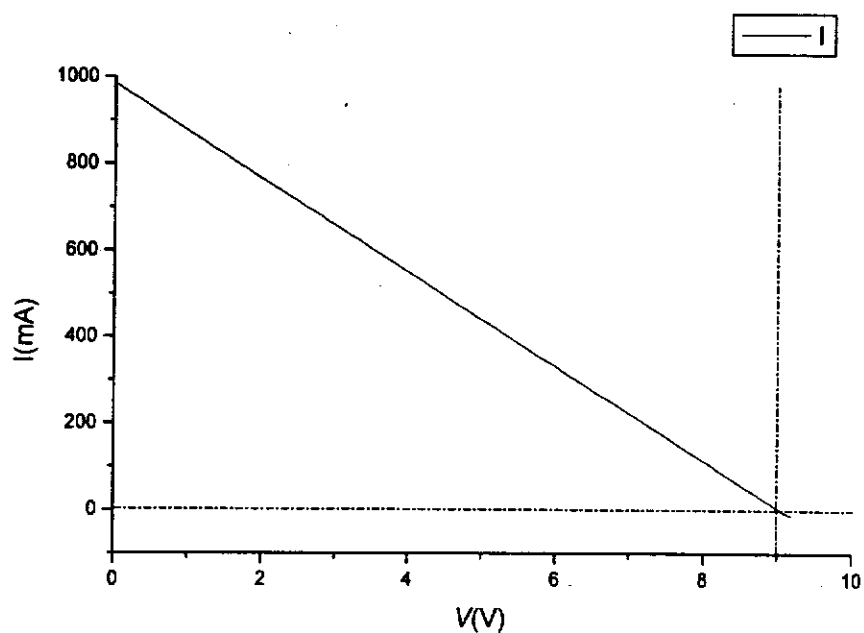
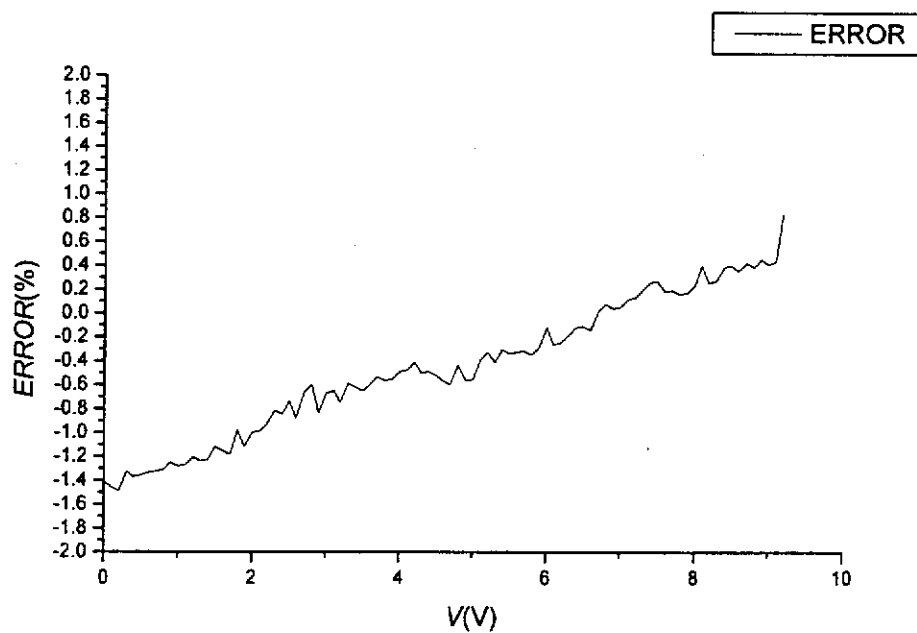
测试电压较低时，流经串联电阻器的电流值较大；测试电压较高时，流经串联电阻器的电流值较小，在这两种情况下，由于干扰的影响，使得测试的相对误差较大。

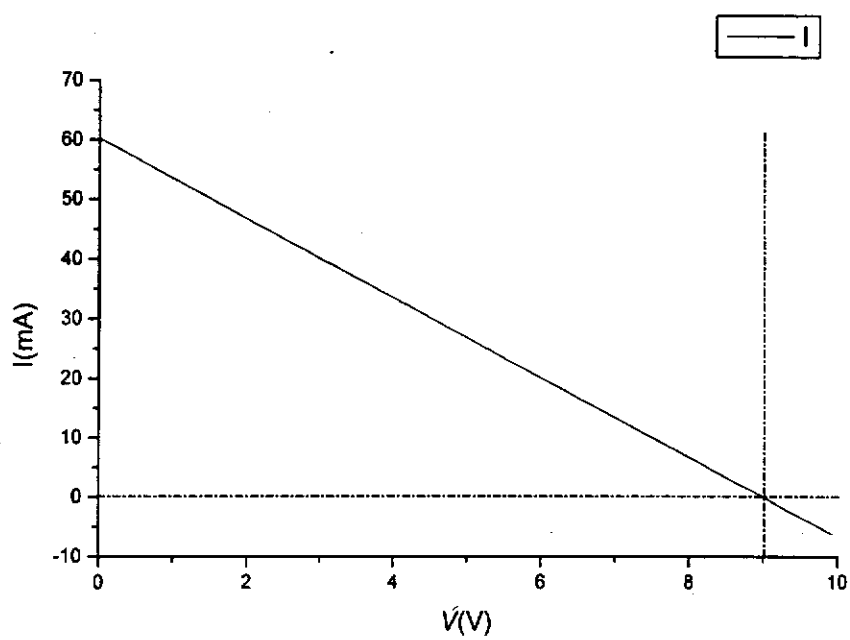
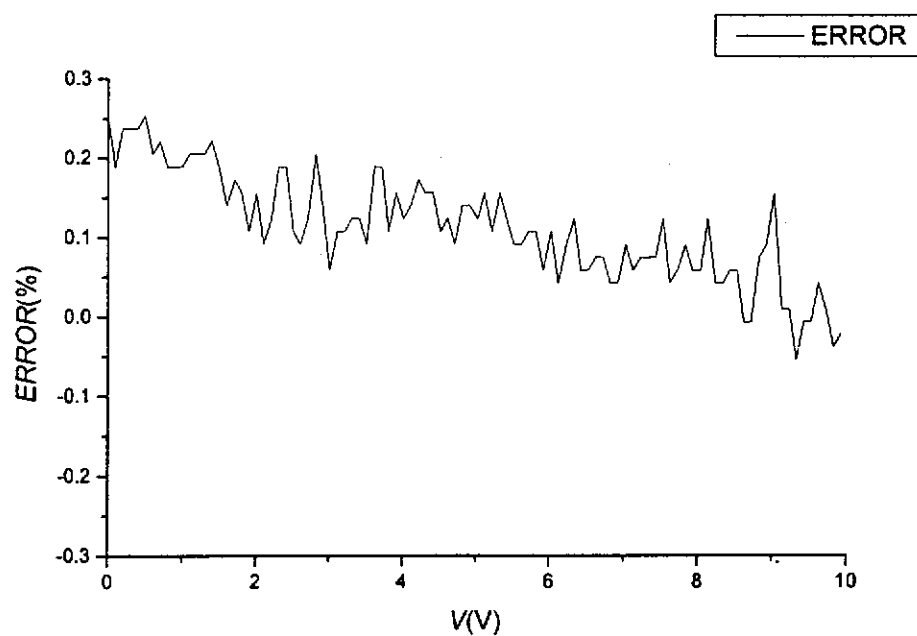
综合上述测试结果，可知本测试系统各量程的测试误差均在 $\pm 1.5\%$ 以内，满足使用需要。

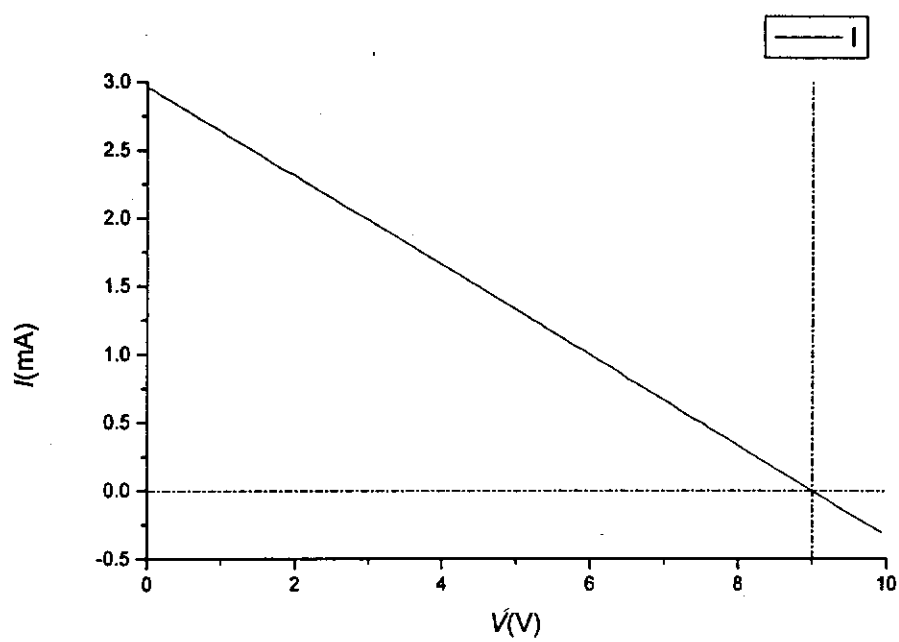
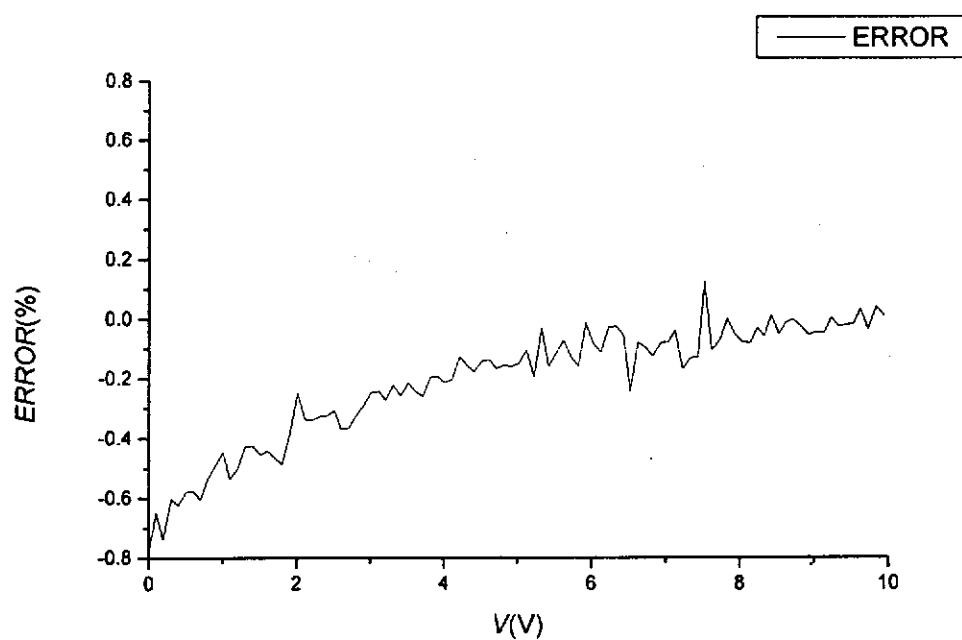
图 6-2 测量 10Ω 电阻器的伏安特性图 6-3 测量 10Ω 电阻器的误差曲线

图 6-4 测量 $125\ \Omega$ 电阻器的伏安特性图 6-5 测量 $125\ \Omega$ 电阻器的误差曲线

图 6-6 测量 $4000\ \Omega$ 电阻器的伏安特性图 6-7 测量 $4000\ \Omega$ 电阻器的误差曲线

图 6-8 测量 9V 电源串联 9Ω 电阻器的伏安特性图 6-9 测量 9V 电源串联 9Ω 电阻器的误差曲线

图 6-10 测量 9V 电源串联 $150\ \Omega$ 电阻器的伏安特性图 6-11 测量 9V 电源串联 $150\ \Omega$ 电阻器的误差曲线

图 6-12 测量 9V 电源串联 3000 Ω 电阻器的伏安特性图 6-13 测量 9V 电源串联 3000 Ω 电阻器的误差曲线

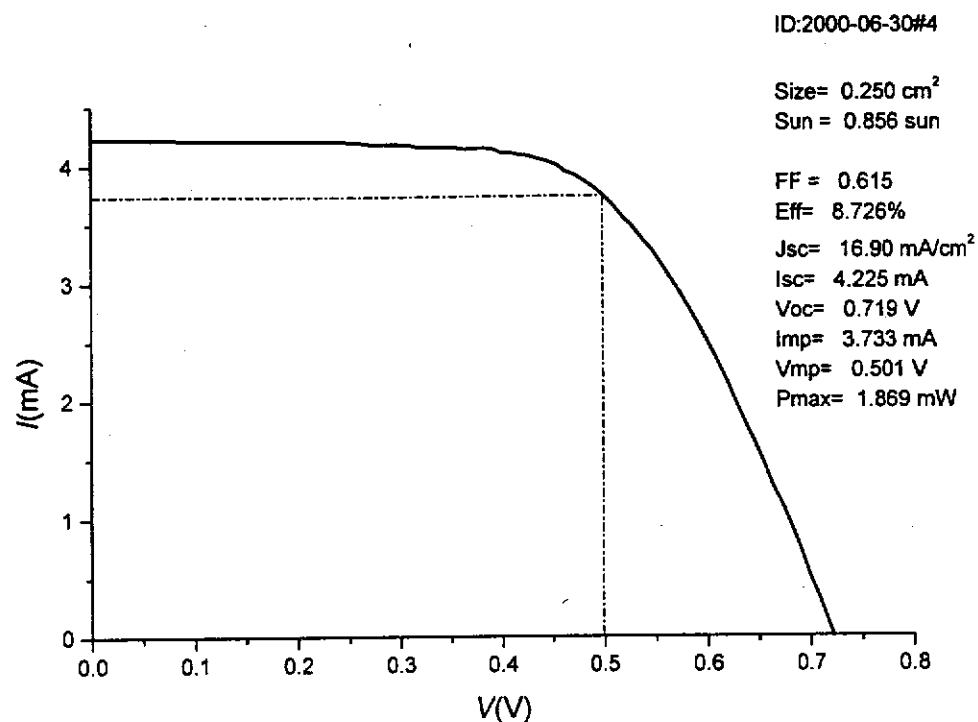
图 6-14 0.25cm²小电池伏安特性

图 6-14 为 0.25cm² 小电池伏安特性曲线，其测试结果为：

效率：8.726%
填充因数：0.615
开路电压：0.719V
短路电流：4.225mA
短路电流密度：16.90mA/cm²
最大功率：1.869mW

测试结果表明：小电池有较高的效率、较大的填充因数和较大的短路电流密度。伏安特性曲线较接近矩形，与硅太阳能电池的典型伏安特性曲线较接近。

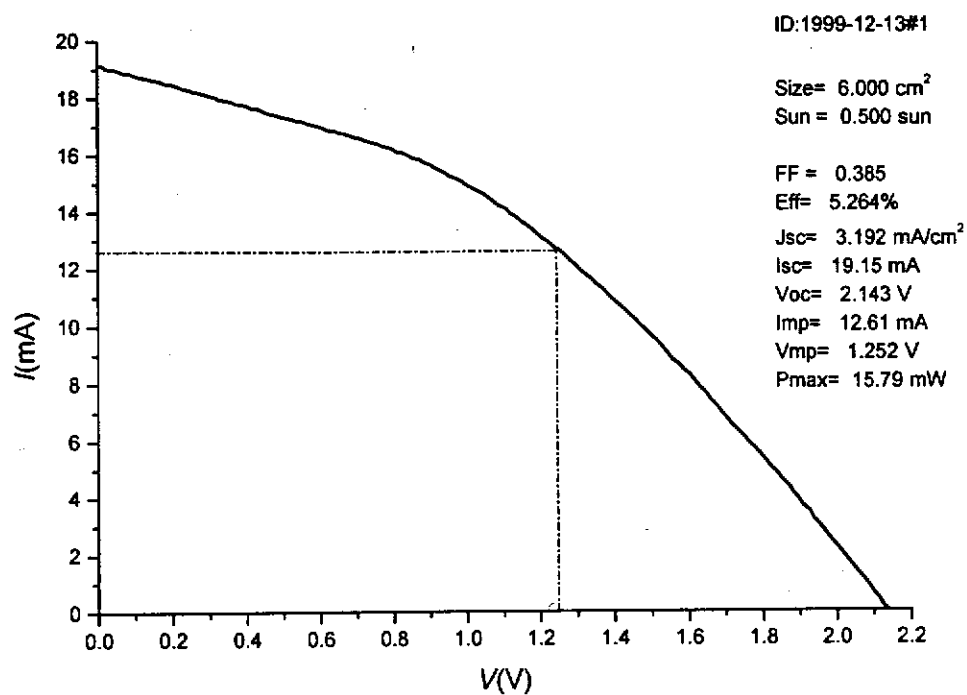
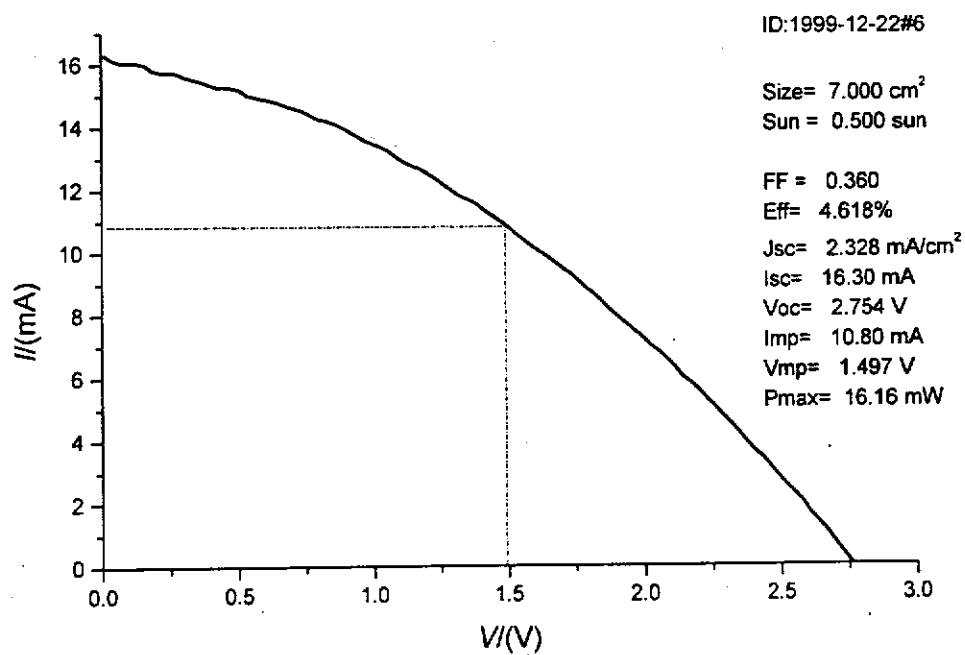
图 6-15 6cm² 串联三节电池伏安特性图 6-16 7cm² 串联四节电池伏安特性

图 6-15 为 6cm^2 串联三节电池伏安特性曲线, 其测试结果为:

效率: 5.264%
填充因数: 0.385
开路电压: 2.143V
短路电流: 19.15mA
短路电流密度: $3.192\text{mA}/\text{cm}^2$
最大功率: 15.79mW

图 6-16 为 7cm^2 串联四节电池伏安特性曲线, 其测试结果为:

效率: 4.618%
填充因数: 0.360
开路电压: 2.754V
短路电流: 16.30mA
短路电流密度: $2.328\text{mA}/\text{cm}^2$
最大功率: 16.16mW

这两只电池均具备较大的功率输出 ($>15\text{mW}$), 在日光照射下即可驱动小型电机旋转。已经具有一定的实用价值。

与 0.25cm^2 小电池相比较, 这两种电池的特点如下:

1. 单体电池的开路电压较高 ($>0.65\text{V}$), 比小电池稍低一些, 说明串联后电压基本未受到影响;
2. 效率只比小电池的 50%稍大一些;
3. 填充因数比小电池的 50%稍大一些;

由伏安特性曲线中可以明显看出, 从开路电压点附近到最佳工作点附近串联电池的等效内阻很大, 所以尽管短路电流较大, 但是供给负载的功率却并不大, 这是影响电池的效率 and 填充因数的关键因素。

根据第一章图 1-1, 太阳电池产生的电流必须经过透明导电膜才能供给负载使用。目前串联电池所使用的透明导电膜的方块电阻为 $5\sim 20\Omega/\square$, 这就使得串联电池的等效内阻很大。因此研究新型的、具有较低的方块电阻的透明导电膜将是一个重要的课题。

6.2 改进意见

本自动测试系统经过一年多的实际使用, 证明其较好地满足了科研工作的需求, 取得了较好的测试效果。所采用的计数方案较为合理, 元器件均为通用型产品, 适宜于推广应用。整个测试系统尽可能地选用高集成度的器件和组件, 结构简单明了, 各个子系统相对独立, 可以根据需要加以灵活组合和配置。

所获得的几十份测试结果对科研工作起到了相应的指导作用, 为电池工艺的改进指出了方向。

由于经费的限制和经验的不足, 下面几个方面尚需改进:

1. 关于太阳模拟器的光谱特性

现在使用的卤素灯的光谱特性与 AM1.5 标准相差很大, 致使同一只电池在模拟器光源下和太阳光下测得的效率相差很大。因此必须对卤素灯的光谱特性进行校正, 以消除这个误差; 同时, 卤素灯的功率也应该相应加大, 以弥补光谱校正所减少的辐照强度。

2. 关于硬件系统

现在所能测量的范围为 10V, 2A, 完全能够满足目前的测试需求。今后若需要更高的电压、电流范围时, 只需更改相应的功率驱动器件即可, 其余部分可不必改动。

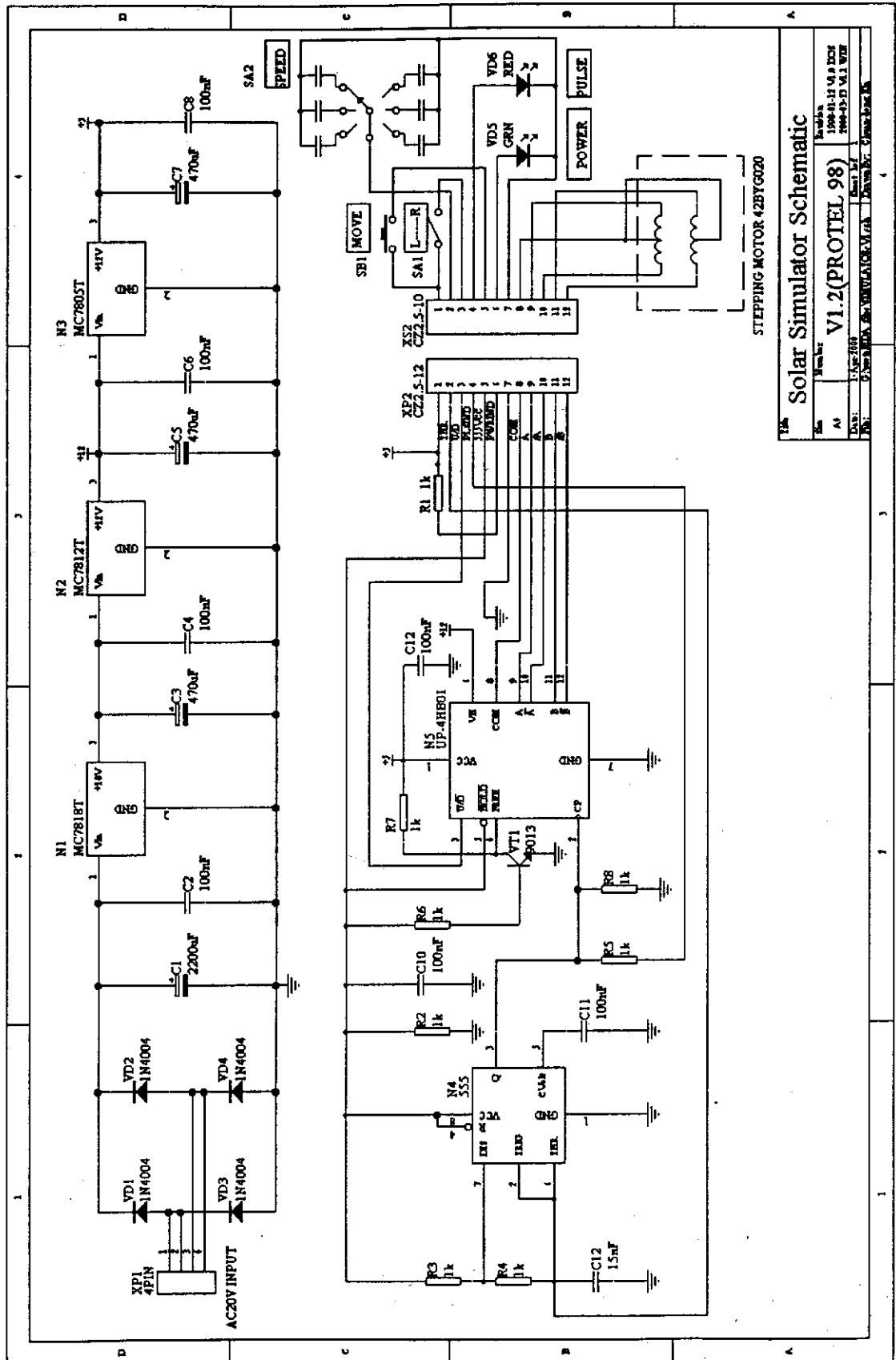
3. 关于软件系统

目前软件系统的数据处理能力(例如: 数据的拟合、平滑、去噪声)不强, 应予以加强。

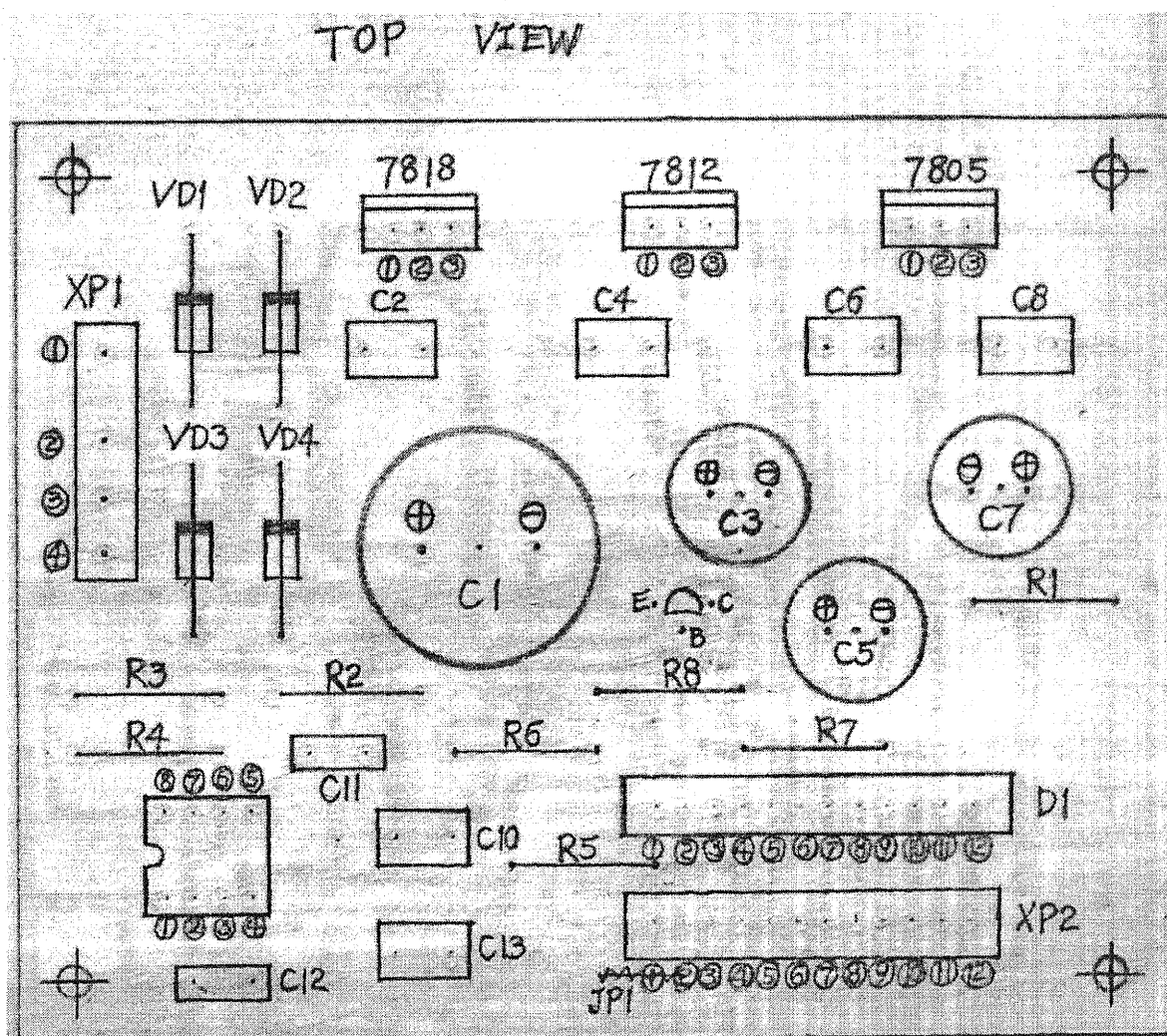
测试文件与数据库系统结合不紧密, 今后应当做到与数据库系统直接接口, 以便于进行测试结果的分类、整理、检索、统计等工作。

视图类可考虑增加缩放功能, 打印方式也应灵活一些, 以方便用户。

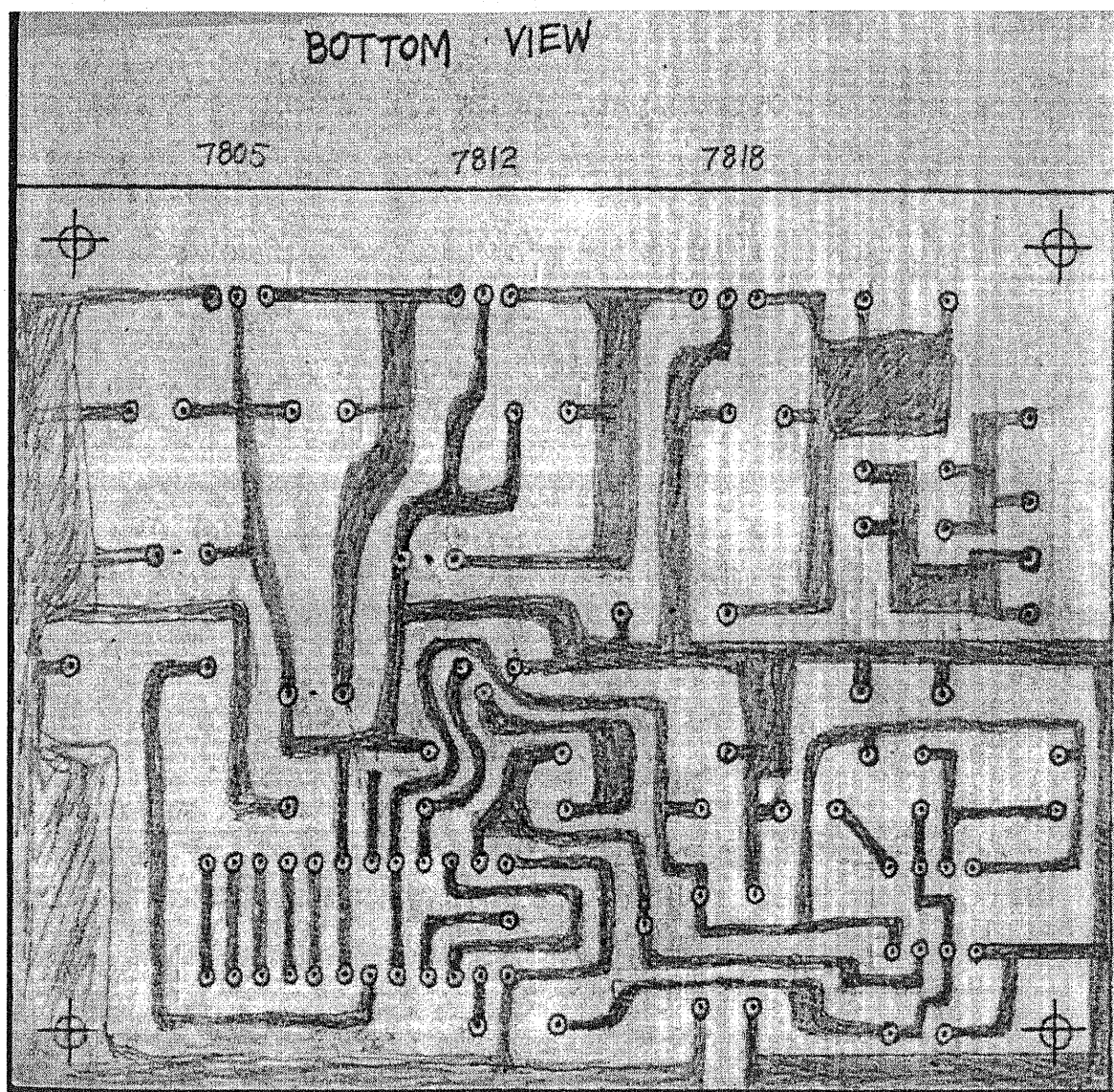
附录 A 太阳模拟器电原理图



附录 B 太阳模拟器印刷电路板图

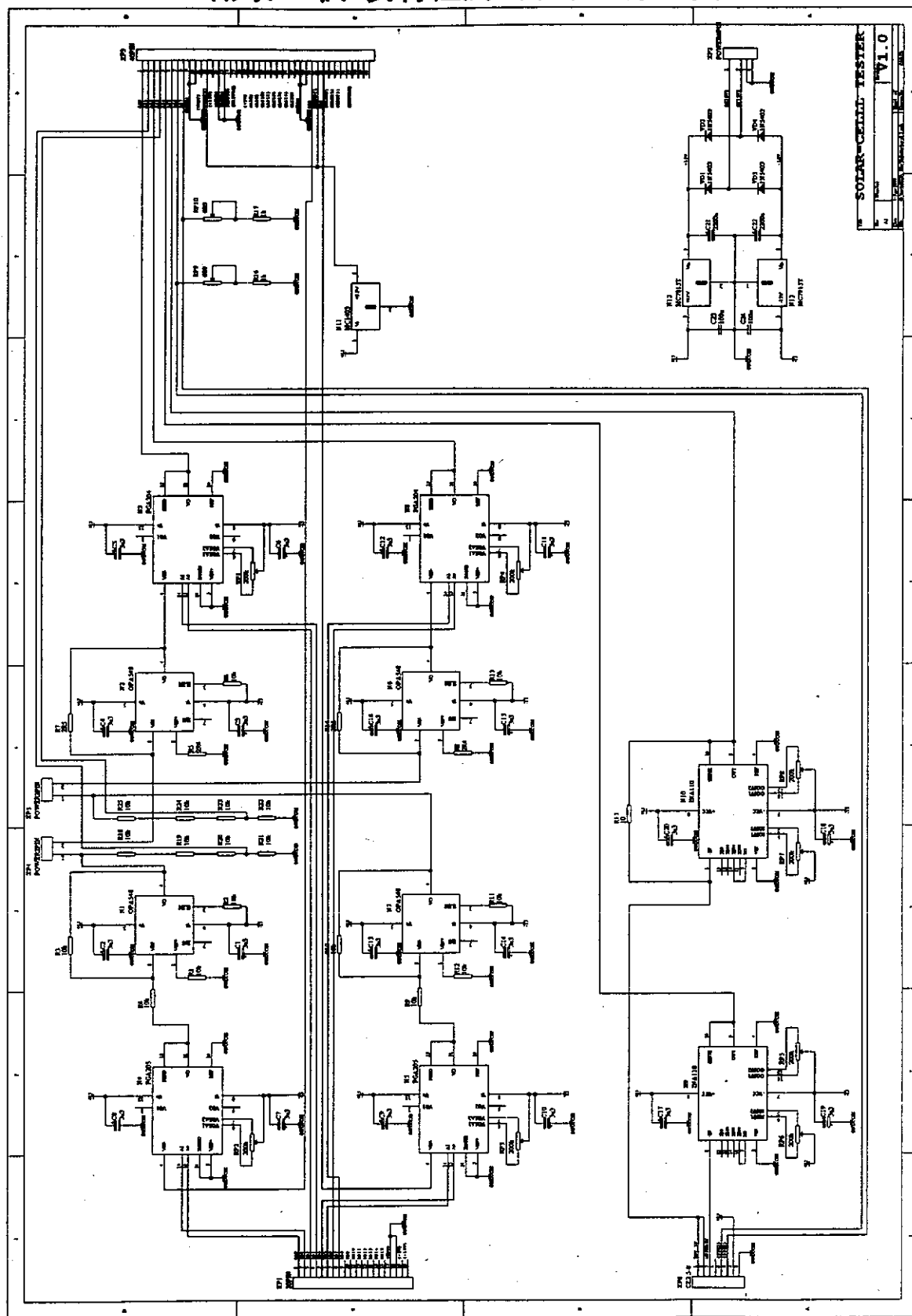


1) 元件面

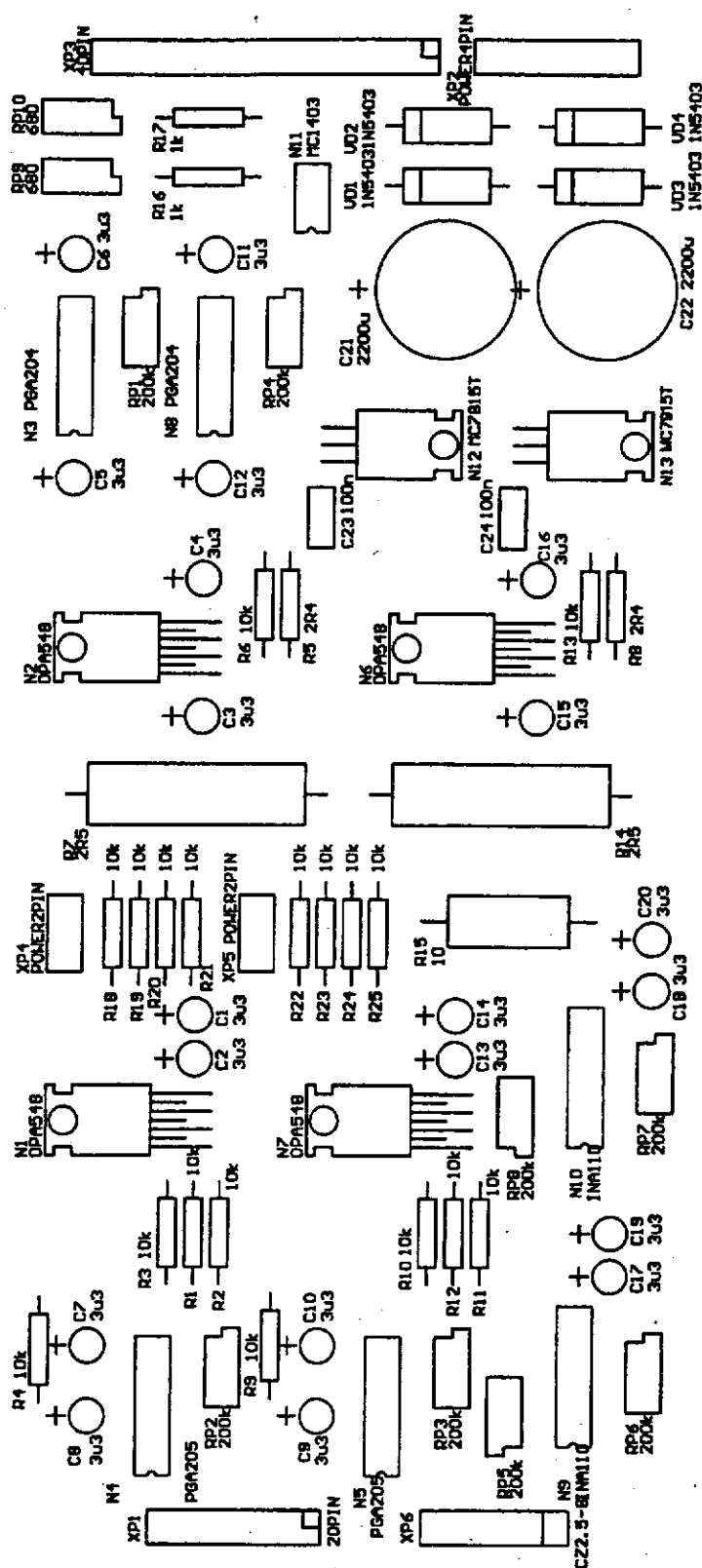


2) 焊接面

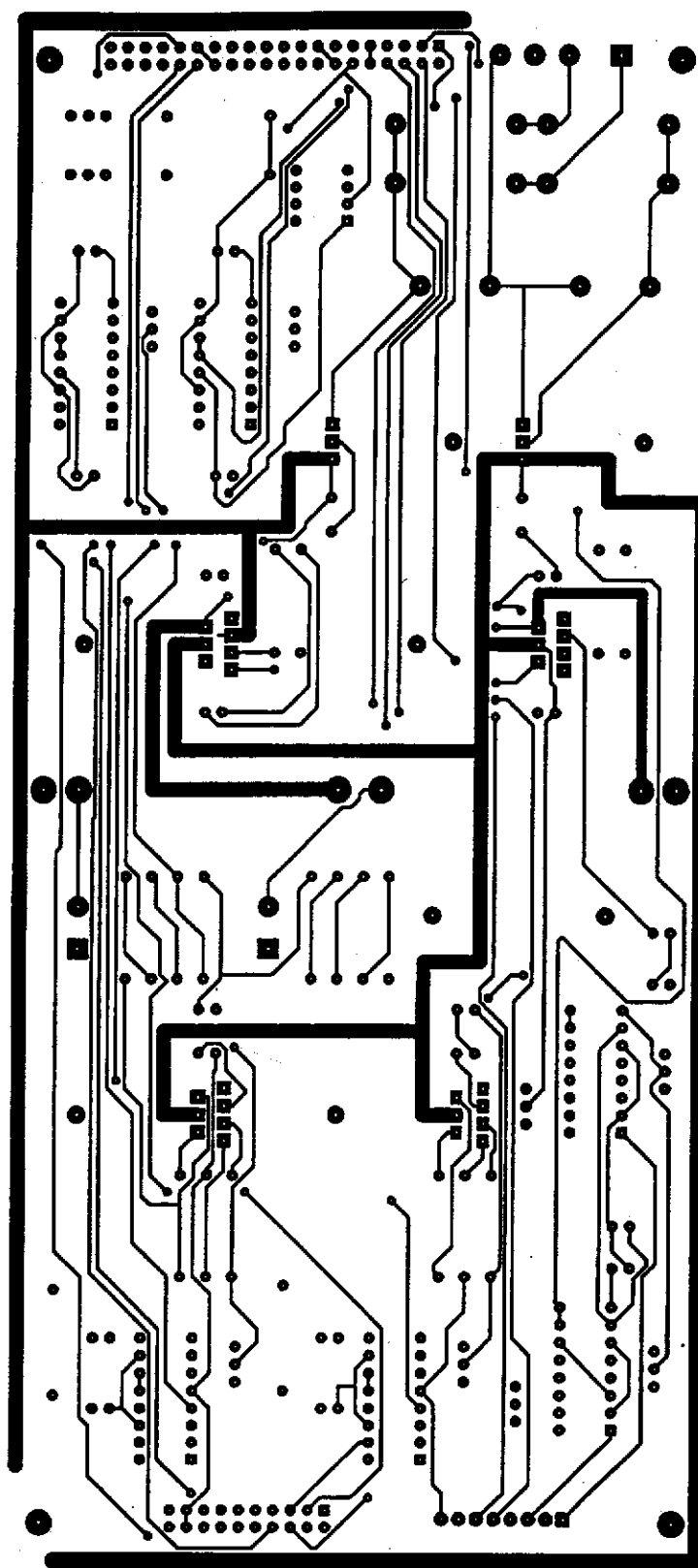
附录 C 伏-安特性测试系统电原理图



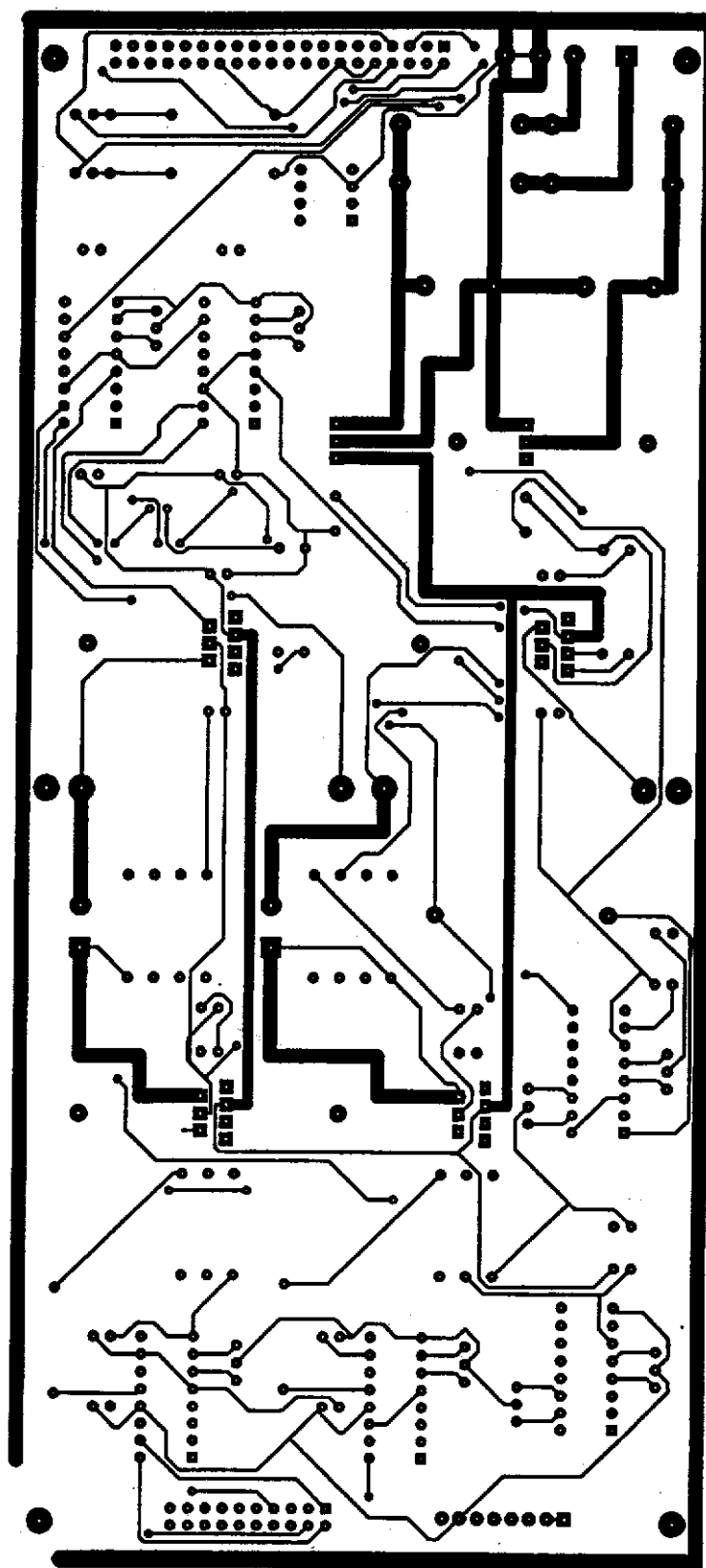
附录 D 伏-安特性测试系统印刷电路板图



1) 字符层

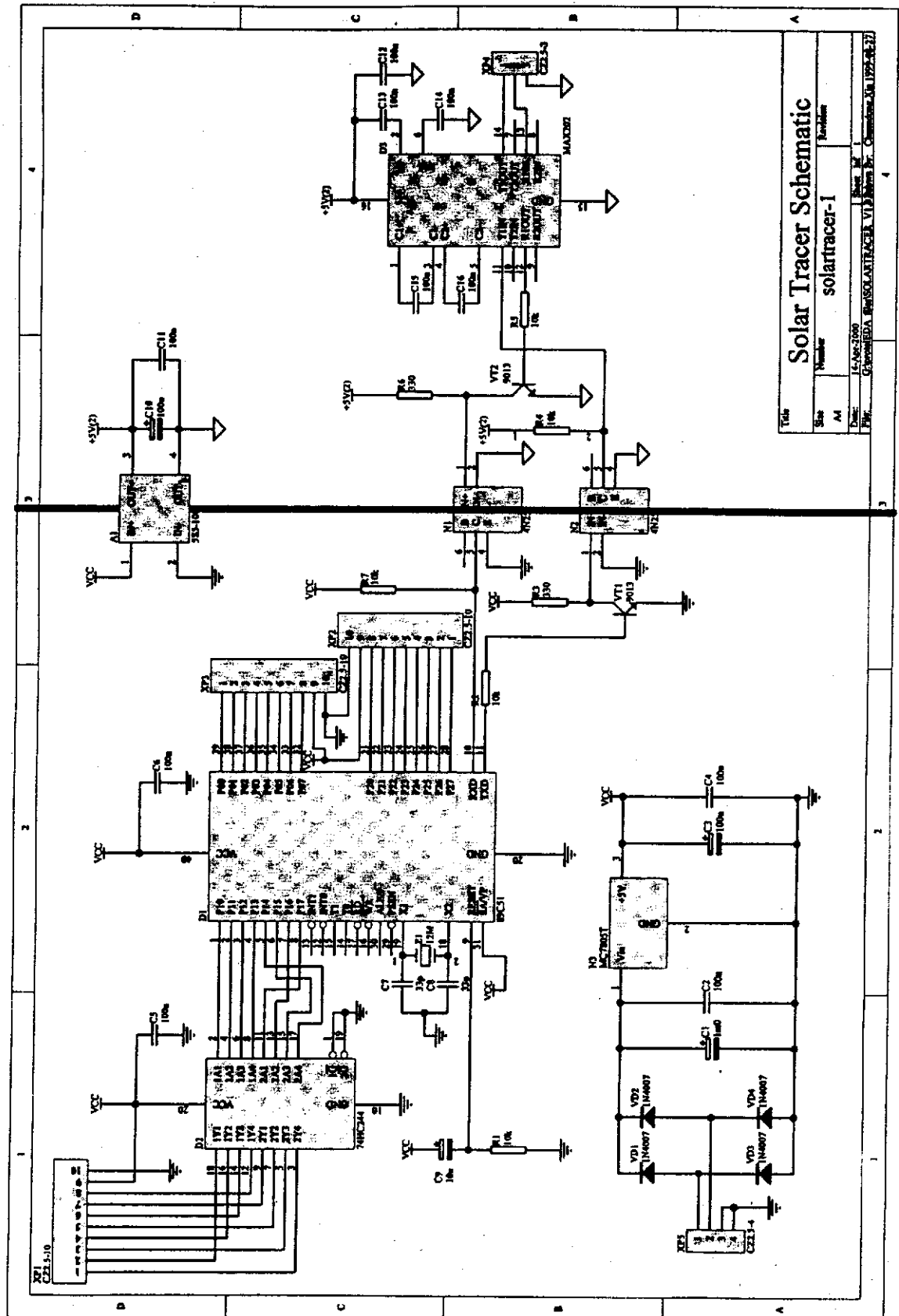


2) 顶层

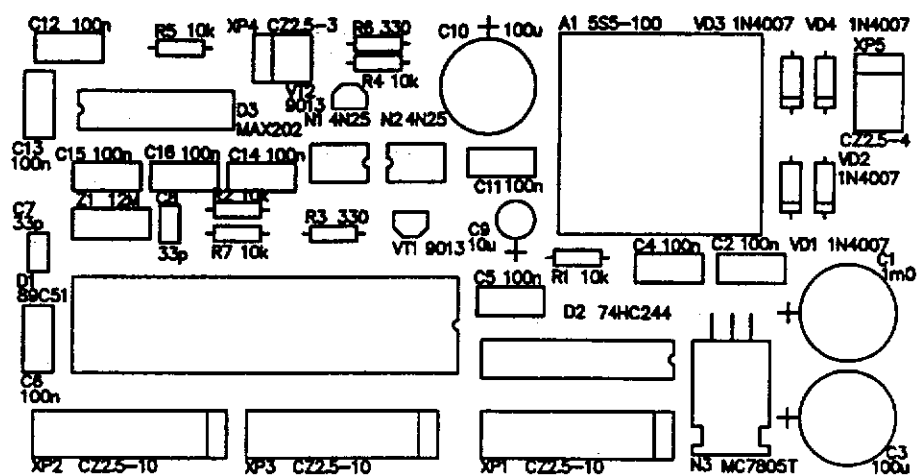


3) 底层

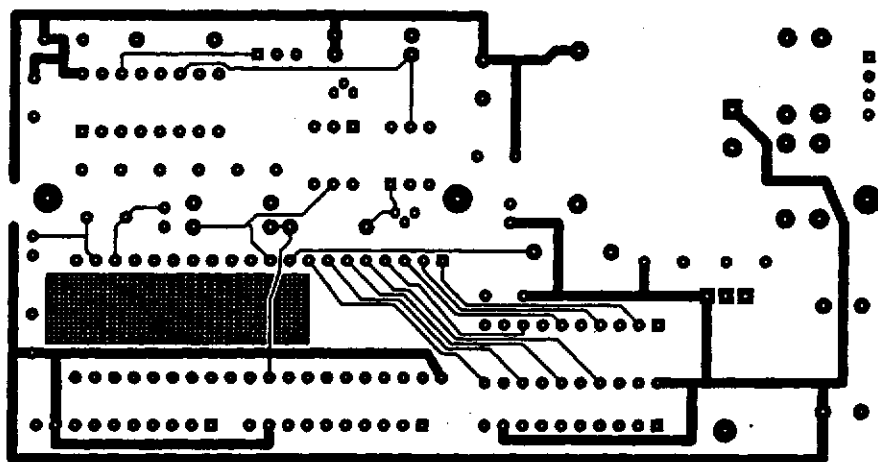
附录 E 太阳跟踪系统电原理图



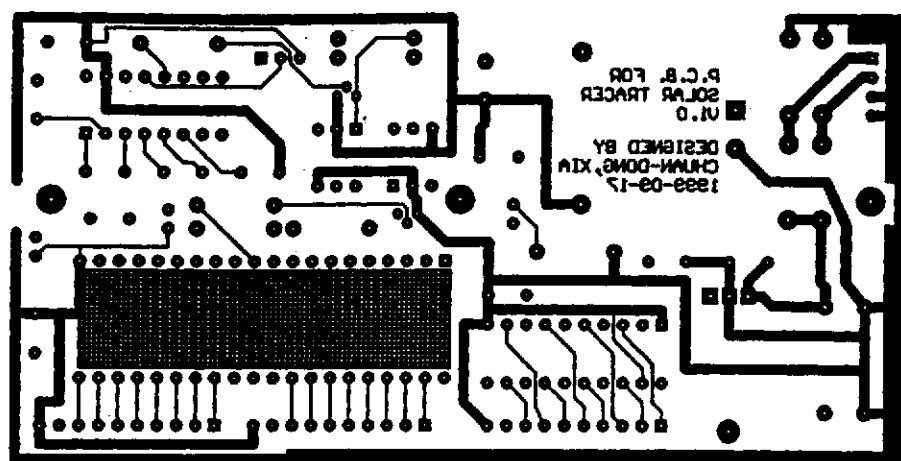
附录 F 太阳跟踪系统印刷电路板图



1) 字符层



2) 顶层



3) 底层

附录 G PC 机源程序清单

1、文件 MainFrm.h

```
1 // MainFrm.h : interface of the CMainFrame class
2 ///////////////////////////////////////////////////////////////////
3
4 #if !defined(AFX_MAINFRM_H_49A03BCD_3FD8_11D3_B89D_0080C875FA7E__INCLUDED_)
5 #define AFX_MAINFRM_H_49A03BCD_3FD8_11D3_B89D_0080C875FA7E__INCLUDED_
6
7 #if _MSC_VER > 1000
8 #pragma once
9 #endif // _MSC_VER > 1000
10
11 class CMainFrame : public CFrameWnd
12 {
13
14 protected: // create from serialization only
15     CMainFrame();
16     DECLARE_DYNCREATE(CMainFrame)
17
18 // Attributes
19 public:
20
21 // Operations
22 public:
23
24 // Overrides
25     // ClassWizard generated virtual function overrides
26     //{AFX_VIRTUAL(CMainFrame)
27     virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
28     //}AFX_VIRTUAL
29
30 // Implementation
31 public:
32     virtual ~CMainFrame();
33 #ifdef _DEBUG
34     virtual void AssertValid() const;
35     virtual void Dump(CDumpContext& dc) const;
36 #endif
37
38 protected: // control bar embedded members
39     CStatusBar m_wndStatusBar;
40     CToolBar m_wndToolBar;
41
42 // Generated message map functions
43 protected:
44     //{AFX_MSG(CMainFrame)
45     afx_msg int OnCreate(LPCREATESTRUCT lpCreateStruct);
46     afx_msg void OnViewToolBar();
47     //}AFX_MSG
48     DECLARE_MESSAGE_MAP()
49 };
```

```

50
51 //////////////////////////////////////////////////
52
53 //({AFX_INSERT_LOCATION})
54 // Microsoft Visual C++ will insert additional declarations immediately before the previous line.
55
56 #endif // !defined(AFX_MAINFRM_H__49A03BCD_3FD8_11D3_B89D_0080C875FA7E__INCLUDED_)

```

2. 文件 MainFrm.cpp

```

1  /* This program is for the PCL812PG data acquisition card, Imax=20mA. */
2  /* Written by Chuan-dong Xia, Sun., Oct. 10th, 1999. */
3  /* Version 1.00 */
4
5
6
7  // MainFrm.cpp : implementation of the CMainFrame class
8
9  #include "stdafx.h"
10 #include "solartester.h"
11
12 #include "MainFrm.h"
13
14 #ifdef _DEBUG
15 #define new DEBUG_NEW
16 #undef THIS_FILE
17 static char THIS_FILE[] = __FILE__;
18 #endif
19
20 //////////////////////////////////////////////////
21 // CMainFrame
22
23 IMPLEMENT_DYNCREATE(CMainFrame, CFrameWnd)
24
25 BEGIN_MESSAGE_MAP(CMainFrame, CFrameWnd)
26     //({AFX_MSG_MAP(CMainFrame)
27     ON_WM_CREATE()
28     ON_COMMAND(ID_VIEW_TOOLBAR, OnViewToolBar)
29     //})AFX_MSG_MAP
30 END_MESSAGE_MAP()
31
32 static UINT indicators[] =
33 {
34     ID_SEPARATOR, // status line indicator
35     ID_INDICATOR_CAPS,
36     ID_INDICATOR_NUM,
37     ID_INDICATOR_SCRL,
38 };
39
40 //////////////////////////////////////////////////
41 // CMainFrame construction/destruction
42
43 CMainFrame::CMainFrame()
44 {

```

```

45     // TODO: add member initialization code here
46
47 }
48
49 CMainFrame::~CMainFrame()
50 {
51 }
52
53 int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
54 {
55     if (CFrameWnd::OnCreate(lpCreateStruct) == -1)
56         return -1;
57
58     if (!m_wndToolBar.CreateEx(this, TBSTYLE_FLAT, WS_CHILD | WS_VISIBLE | CBRS_TOP
59         | CBRS_GRIPPER | CBRS_TOOLTIPS | CBRS_FLYBY | CBRS_SIZE_DYNAMIC) ||
60         !m_wndToolBar.LoadToolBar(IDR_MAINFRAME))
61     {
62         TRACE0("Failed to create toolbar\n");
63         return -1;    // fail to create
64     }
65
66     if (!m_wndStatusBar.Create(this) ||
67         !m_wndStatusBar.SetIndicators(indicators,
68             sizeof(indicators)/sizeof(UINT)))
69     {
70         TRACE0("Failed to create status bar\n");
71         return -1;    // fail to create
72     }
73
74     // TODO: Delete these three lines if you don't want the toolbar to
75     // be dockable
76     m_wndToolBar.EnableDocking(CBRS_ALIGN_ANY);
77     EnableDocking(CBRS_ALIGN_ANY);
78     DockControlBar(&m_wndToolBar);
79
80     return 0;
81 }
82
83 BOOL CMainFrame::PreCreateWindow(CREATESTRUCT& cs)
84 {
85     if( !CFrameWnd::PreCreateWindow(cs) )
86         return FALSE;
87     // TODO: Modify the Window class or styles here by modifying
88     // the CREATESTRUCT cs
89
90     // cs.style &= ~WS_MAXIMIZEBOX;
91     cs.cx=1023;
92     cs.cy=767;
93
94     return TRUE;
95 }
96
97 //////////////////////////////////////
98 // CMainFrame diagnostics
99

```

```

100  #ifdef _DEBUG
101  void CMainFrame::AssertValid() const
102  {
103      CFrameWnd::AssertValid();
104  }
105
106  void CMainFrame::Dump(CDumpContext& dc) const
107  {
108      CFrameWnd::Dump(dc);
109  }
110
111  #endif //_DEBUG
112
113  //////////////////////////////////////
114  // CMainFrame message handlers
115
116
117  void CMainFrame::OnViewToolbar()
118  {
119      // TODO: Add your command handler code here
120  }
121

```

3、文件 Para.h

```

1  #if !defined(AFX_PARA_H_CC5DE021_541F_11D3_B89F_0080C875FA7E__INCLUDED_)
2  #define AFX_PARA_H_CC5DE021_541F_11D3_B89F_0080C875FA7E__INCLUDED_
3
4  #if _MSC_VER > 1000
5  #pragma once
6  #endif // _MSC_VER > 1000
7  // Para.h : header file
8  //
9
10  //////////////////////////////////////
11  // CPara dialog
12
13  class CPara : public CDialog
14  {
15  // Construction
16  public:
17      CPara(CWnd* pParent = NULL);    // standard constructor
18
19  // Dialog Data
20      //{AFX_DATA(CPara)
21      enum { IDD = IDD_DIALOG1 };
22      UINT m_nAverage;
23      double m_fVmax;
24      UINT m_nSample;
25      UINT m_nDelay;
26      double m_fSun;
27      double m_fSize;
28      double m_fHum;
29      double m_fT_dry;

```