

控温模块说明书

一、产品特点

DUT 系列温度控制模块可选多种控温模式，连接七路热电偶温度传感器或八路热电阻温度传感器，实现多路控温。开关量输出直接驱动固态继电器实现 PWM 调功。开关量输出可以组态成报警逻辑信号输出，也可以当为离散输出点由上位机控制其逻辑状态。模块通过 RS-485 总线接口与通用人机界面设备通讯，如：触摸屏、PC 机、组态软件等。通讯协议以 MODBUS-RTU、MODBUS-ASC 为主，开放模块全部功能，可实现全部参数的设置和读取。模块也可设成兼容台湾研华 ADAM4000 系列或松下 PLC 的通讯协议，但只能读取温度和修改部分参数。本产品主要用于各种需要温度控制的现场，控温机电设备配套，以及工业现场总线系统的构建等。产品特点如下：

- 1、以工业级单片机为核心，集成电路全部采用 CMOS 器件，产品全部经过工业级温度考核；
- 2、隔离 DC-DC 变换器和隔离 RS485 输出，以及隔离开关量输出接口，全浮空电路设计；
- 3、采用三线制热电阻测量电路，自动补偿引线电阻；
- 4、模块内置测温元件，软件完成热电偶参比端温度补偿；
- 5、无电位器设计，软件校准零点和满度；
- 6、采用数字校零技术消除运放的失调和漂移，保证各个通道的精度和一致性；
- 7、软件选择同类型传感器中的不同型，通用性强；
- 8、输入和电源加有完善的 TVS 保护电路，误加几十伏高压不会损坏；可耐受几千伏 ESD 火花放电。
- 9、电源监视电路和看门狗电路，保证恶劣环境下可靠运行。

二、技术指标

- 1、传感器：PT100/Cu50/Cu100，各种热电偶，电流、电压
- 2、控温路数：8 路（热电偶：7 路，1 路用于环境温度补偿）
- 3、范围：温度传感器全范围，0~50mV，0~5V，4~20mA
- 4、分辨率：0.1℃，300 码/mV，2000 码/V，500 码/mA
- 5、电路精度：±0.2℃，±2 字
- 6、转换时间：有滤波：2.16 秒/8 通道，无滤波：0.72 秒/8 通道；
- 7、开关量输出：9 路隔离 OC 输出，最大驱动能力：500mA；
- 8、开关量输入：4 路光耦隔离输入，0~30V 输入；

- 9、 通讯接口：隔离 RS485，波特率可选，出厂默认：9600, 8, N, 1;
- 10、 电源电压：9~30V
- 11、 功 耗：<2W
- 12、 外形尺寸：146 × 116 × 45
- 13、 环境温度：-30~70℃
- 14、 相对湿度：≤85%

三、电路原理框图

温度控制模块原理框图如图 1 所示，温度传感器经模拟开关由单片机控制顺序选通到放大电路和 A/D 转换器 ICL7135 (±20000 码)。单片机采集数据后，经校零、滤波、参比端补偿 (热电偶) 和线性化处理后将转换为对应的温度数字量，存入内部 RAM。

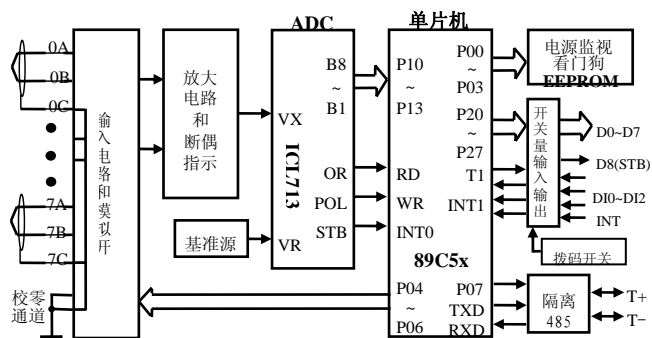


图 1 温度控制模块电路原理框图

单片机每隔 2.16 秒 (有滤波) 或 0.72 秒 (无滤波) 完成一次以上操作，自动刷新内部 RAM。上位机可以通过 485 口取得这些温度数据。开关量输出口用于控温和报警输出，单片机根据每个开关量输出的控制参数进行控温计算，或报警逻辑运算，驱动输出。

当开关量输出不用于控温和报警时，可作为异步并行口输出温度数据，详见后面说明。

四、底部外型尺寸和端子定义

图 2 为模块底部外型装配图，外型尺寸为 146 × 98.4 × 45 (单位 mm)，模块装配在工业标准导轨上。此外，模块两侧各有一固定孔，孔径为 5.5mm，也可以用 φ4 螺丝通过这两个固定孔将模块固定。

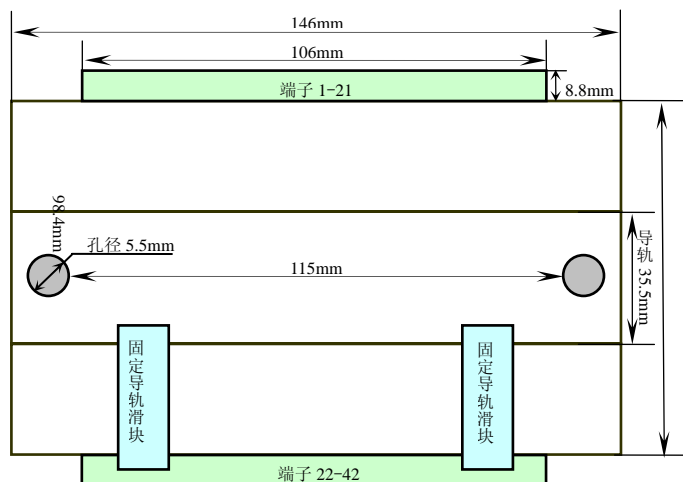


图 2 模块底部外型装配图

图 3 为端子定义图，模块电源使用直流 24V 接于 V+ 和 V- 之间; T+ 和 T- 为 RS485 接口的 A 和 B (研华的 ADAM4520 为 DATA+ 和 DATA-);

STB 为第九个开关量输出，D0~D7 为八个开关量输出，IN1~IN4 为开关量输入，iA、iB、iC 为模拟量输入端。

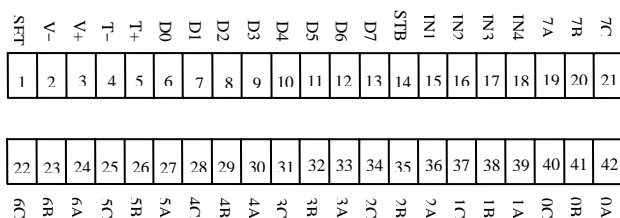


图3 端子定义图

图 4.1 与图 4.2 为采集模块的传感器输入端子接线图。其中，传感器输入端每个通道为 3 个接线端子，8 路共 24 位端子。热电阻温度传感器采用三线制连接，接线如图 4.1 所示，8 路接法相同，热电阻温度传感器连接时要注意三根导线要采用同一规格，且 iB 和 iC 两根导线在现场的传感器端连接到一起。热电偶和电流、电压输入接法基本相同，如图 4.2 所示，即每组的 iA 为传感器的正端和 iB 为传感器的负端，iC 为屏蔽端。

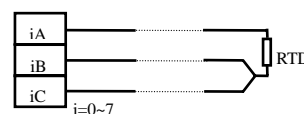


图 4.1 单路热电阻端子接线图



图 4.2 单路热电偶端子接线图

在模块内部，所有的 C 端都接模拟信号地，即 0C~7C 是连通的，该地线和电源及通讯都是隔离的。在实际应用中，同一个模块的八个通道之间，共模电压差异不要太大，各个传感器之间应该基本保持等电位。在一些强干扰环境，还应该考虑将 C 端与被测对象的外壳连接在一起。

热电偶控温模块，最后一路温度用于冷端补偿，模块内自带补偿元件，端子上不要接任何东西，模块给出的最后一路温度数据为环境温度。模块内的测温元件也可以通过端子 7A、7B 和 7C 连接到现场，这样可以节省补偿导线。

五、传感器类型

模块内部有一传感器类型码用以选择传感器类型和模块的工作方式，该字节定义如下：

| | | | | | | | |
|--------|--------|----|--------|-------|----|----|----|
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 滤 波 | 补 偿 | C8 | 独 立 | 选择传感器 | | | |

D7: = 0 滤波工作方式，2.16S/8 通道；

= 1 不滤波工作方式，0.72S/8 通道。

D6: = 0 热偶冷端不补偿；

= 1 热偶冷端补偿。

D5: = 0 通道 7 为外输入传感器；

= 1 通道 7 为模块内环境温度测试（冷端补偿）。

D4: = 0 由 D3~D0 选择传感器类型，如表 1 所示；

= 1 单独设置传感器类型码，这时模块内另有 8 个字节各自独立定义每个通道的输入传感器类型。传感器类型码出厂时根据用户要求已设置好。各种型号的热电偶和 0~50mV 电压由于硬件电路相同可以互换，传感器类型码由设置程序设置。铂热电阻 PT100、铜热电阻 Cu50 和 Cu100 硬件电路相同，可以由设置程序选择。4~20mA、0~5V 等型号硬件有差异，需在订货时选择。传感器类型设置在 DUTSETMD.EXE 程序中可以很方便实现。

表 1 传感器类型码表

| D3 ~ D0 | 传感器 | 范围 | 灵敏度 | 路数 |
|---------|-----------|--------------------|----------|----|
| 0H | 双极性 A/D | -19999~+19999 9 | 1 字/码 | 8 |
| 1H | 电压 | 0~50mV | 300 码/mV | 8 |
| 2H | 电流 | 4~20mA | 500 码/mA | 8 |
| 3H | 铂电阻 PT100 | -70~270 | 0.01℃/码 | 8 |
| 4H | J 型热电偶 | -210~1200 | 0.1℃/码 | 7 |
| 5H | E 型热电偶 | -230~1000 | 0.1℃/码 | 7 |
| 6H | N 型热电偶 | -230~1300 | 0.1℃/码 | 7 |
| 7H | T 型热电偶 | -230~400 | 0.1℃/码 | 7 |
| 8H | W 型热电偶 | 0~2310 | 0.1℃/码 | 7 |
| 9H | R 型热电偶 | -50~1760 | 0.1℃/码 | 7 |
| AH | S 型热电偶 | -50~1760 | 0.1℃/码 | 7 |
| BH | B 型热电偶 | 50~1820 | 0.1℃/码 | 7 |
| CH | K 型热电偶 | -230~1370 | 0.1℃/码 | 7 |
| DH | 铂电阻 PT100 | -200~850 | 0.1℃/码 | 8 |
| EH | 铜电阻 CU50 | -50~150 | 0.1℃/码 | 8 |
| FH | 铜电阻 CU100 | -50~150 | 0.1℃/码 | 8 |

六、串行通讯协议

(一) 串行通讯设置

1、通讯波特率：

模块内有一波特率控制字，用以选择波特率、通讯校验方式和异步并行输出接口的工作方式，其格式如下：

| b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|----|----|----|-----|----|-------|----|----|
| 校 | 请 | 字 | 通讯协 | | 选择波特率 | | |
| 验 | 求 | 节 | 议 | | | | |

b7=0 串行通讯无校验方式；

b7=1 串行通讯奇校验方式。

b6=0 异步并行口无条件输出数据，每 2.16 秒（不滤波为 0.72S）输出 8 个通道数据；

注意：此功能会影响到开关量输出口的控温操作，

b6=1 异步并行口请求输出，IN4 为 0N（接地）时，请求输出一次数据。

注意：若将第九路模入（虚设的温度通道）的上限报警设为 0，可取消此功能。详见后面说明。

b5: 当开关量输出当作异步并行口时，选择字节/半字节输出，

b5=0 D0~D3 并行半字节输出，每次输出 4 位二进制数；

注意：这时 D4~D7 仍然可以设成控温输出。

b5=1 D0~D7 并行字节输出，每次输出 8 位二进制数。

b4、b3: 通讯协议选择

00: MODBUSRTU

01: MODBUSASC

10: 研华协议

11: 松下协议

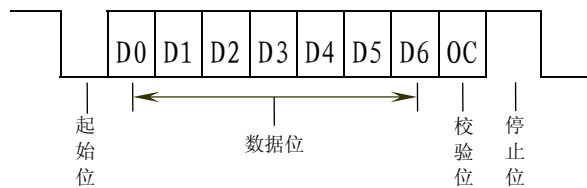
b2~b0 选择串行通讯波特率（bps），如表 2 所示。

表 2 通讯波特率码表

| 代 码 | 波特 率 | 代 码 | 波特 率 |
|--------|---------|--------|---------|
| 0 | 1200 | 3 | 9600 |
| 1 | 2400 | 4 | 19200 |
| 2 | 4800 | 5 | 38400 |

默认值为 03H，即波特率为 9600，无校验。波特率控制字可由设置程序 DUTSETMD.EXE 改变。

2、字符格式



字符格式采用标准异步串行通讯格式，一个起始位、8 个数据位，或 7 个数据位加上一个奇校验位（由波特率控制字最高位选择），一个停止位。

3、通讯方式

MODBUS SLAVE 从机工作方式，响应主机发出的指令。

(二) 模块站地址设置

模块电路板上有一个 5 位 DIP 开关，用于设置模块的站地址，设定范围为 0^{31} ，即 5 位 DIP 开关对应五位二进制数，开关拨到 ON 表示 1，OFF 表示 0。

模块单片机系统内部也有一个寄存器(1CH)用于设定模块站地址，地址设定范围为： 0^{255} 。

模块的实际站地址为拨码开关设定地址和模块内部设定地址之和。该地址的有效范围是 1^{255} 。需要注意的是：0 是无效地址，即内部和外部地址之和不能为 0；如果内外地址之和大于 255，则高位溢出，地址折回 0^{255} 。

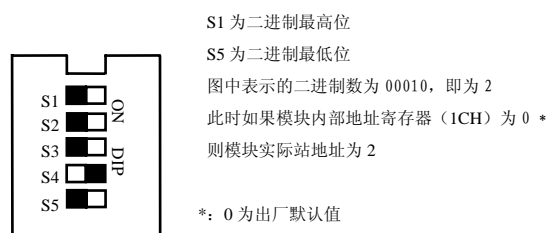


图5 DIP开关设模块站地址图

模块出厂默认设定内部地址寄存器（1CH）为 0，此时五位拨码开关表示的数就是模块的实际站地址。如下图所示：

（三）MODBUS 协议（默认协议）和 PLC 兼容地址空间

MODBUS 最初是 MODICON 公司用于 PLC 相互信息交互而设计的，随后为众多 PLC 厂家所接受，并成为一种通用的通讯标准。MODBUS 是 MODICON 公司的注册商标。

MODBUS 协议是与应用层协议，位于 OSI 模型的第七层。在不同的总线和网络设备之间提供客户机/服务器通讯服务，即主/从通讯，客户机是主，服务器是从。工作的基本过程是一系列周而复始的呼叫—应答，呼叫为主方设备（PC 机、触摸屏和 PLC 等）发出，指向挂在总线某处具有确定地址的从方设备（例如 DUT 系列模块），从方则以应答方式确认收到呼叫并返回主方所要的数据。

1、MODBUS RTU 传输格式、CRC 校验

RTU 协议以总线上至少 3.5 个字节传输周期的空闲为开始和结束标志，中间的格式按顺序依次为站地址、功能码、数据和 CRC 校验码。主机发送和从机响应都是如此，如下图所示：

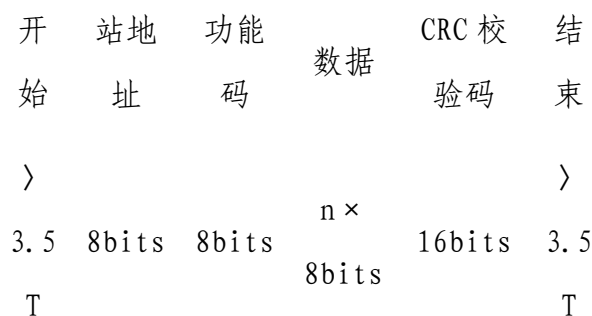


图 6. MODBUS RTU 传输格式

T 为一个字节的传输时间

CRC校验，即循环冗余校验，为CRC-16 校验，校验多项式为 $(X^{16}+X^{15}+X^2+1)$ 。由于低位先发送，所以校验二进制为 1010000000000001 (A001H)。

2、MODBUS ASC II 传输格式、LRC 校验

ASC II 协议以冒号 “:” 为开始标志；以回车<CR>换行<LF>符为结束标志，对应的 ASC II 码为 0D、0A。中间的格式按顺序依次为站地址、功能码、数据和 LCR 校验码。主机发送和从机响应都是如此，如下图所示：

| 开始 | 站地 址 | 功能 码 | 数据 | LRC 校 验码 | 结束 |
|-----|---------|---------|-------|-------------|----------|
| “:” | 2 | 2 | n | 2 | <CR><LF> |
| | CHARS | CHARS | CHARS | CHARS | |

图 7. MODBUS ASC II 传输格式

其中站地址、功能码和数据的每一个字符与 RTU 格式的每个字节的上下两个半字节数的 16 进制表示符 “0~9”，“A~F” 相对应，高半字节在前，低半字节在后。例如：

RTU 发送：

<02> <01> <00> <01> <00> <01> <crc1> <crc2>

对应 ASC 发送：

: 020100010001<LCR1><LCR2><CR><LF>

LCR 校验码是累加和校验，是将站地址、功能码和数据的所有 ASC 码加在一起，进位丢弃，得到的八位二进制数取其负数，再转换成两个 ASC 码，附在数据串后面。

3、MODBUS 功能码

DUT 控温模块所用到的功能码只是 MODBUS 功能码的一部分，只限于数据的存取操作，下面以 RTU 格式进行举例说明，但只要转换一下很容易得知 ASC 格式。有如下这些：

01H: 位操作，读内部位存储和离散输出；

可操作的模块资源有：开入、开出和上下限报警点。

命令格式：（以模块地址 2，读 D1 状态为例）

主机呼叫:

<02> <01> <00> <01> <00> <01> <crc1> <crc2>

站地址 功能码 起始地址 位数量 CRC 校验

从机响应:

<02> <01> <01> <00 或 01> <crc1> <crc2>

站地址 功能码 后面的字节数 位状态 CRC 校验

02H: 位操作, 读离散输入的状态;

可操作的模块资源有: 开入和上下限报警点。

命令格式: (以模块地址 2, 读通道 1 上限报警状态为例)

主机呼叫:

<02> <02> <00> <01> <00> <01> <crc1> <crc2>

站地址 功能码 起始地址 位数量 CRC 校验

从机响应:

<02> <02> <01> <00 或 01> <crc1> <crc2>

站地址 功能码 位状态字节数 位状态 CRC 校验

注: 一个字节可以表示 8 个位状态, 如果位数量超过 8 个, 响应会自动增加字节数, 空余高位补 0。

03H、04H: 字操作, 读内部寄存器;

可操作的模块资源有: 全部内部寄存器。

命令格式: (以模块地址 2, 读通道 1 温度为例)

主机呼叫:

<02> <03> <00> <01> <00> <01> <crc1> <crc2>

站地址 功能码 起始地址 通道数量 CRC 校验

从机响应:

<02> <03> <02> <0F> <F6> <crc1> <crc2>

站地址 功能码 后面的字节数 温度值 4086 (0FF6H) CRC 校验

05H: 位操作, 写单个内部位存储和离散输出;

可操作的模块资源有: 开关量输出和控温总控。

命令格式: (以模块地址 2, 写 D1 输出为例)

主机呼叫:

<02> <05> <00> <01> <FFor00> <00> <crc1> <crc2>

站地址 功能码 输出地址 输出值 CRC 校验

注: 输出值 0000 表示输出 0, 输出值 FF00 表示输出 1

从机响应:

<02> <05> <00> <01> <FFor00> <00> <crc1> <crc2>

站地址 功能码 输出地址 输出值 CRC 校验

06H: 字操作, 写单个寄存器;

可操作的模块资源有: 全部控温参数、报警参数和部分系统参数, 详见后面说明。

命令格式: (以模块地址 2, 写通道 1 设定值为例)

主机呼叫:

<02> <06> <01> <18> <03> <E8> <crc1> <crc2>

站地址 功能码 写入地址 设定值 100 度 CRC 校验

从机响应:

<02> <06> <01> <18> <03> <E8> <crc1> <crc2>

站地址 功能码 写入地址 设定值 100 度 CRC 校验

10H: 字操作, 写多个寄存器;

可操作的模块资源有: 全部控温参数、报警参数和部分系统参数, 详见后面说明。

命令格式: (以模块地址 2, 写通道 1 控制参数和设定值为例)

主机呼叫:

<02> <06> <01> <17> <00> <02> <04>

站地址 功能码 起始地址 寄存器数量 后面的字节数

<00> <17> <03> <E8> <crc1> <crc2>

控制参数 设定值 CRC 校验

从机响应:

<02> <06> <01> <17> <00> <02> <crc1> <crc2>

站地址 功能码 起始地址 寄存器数量 CRC 校验

11H: 厂家信息读取

命令格式: (以模块地址 2 为例)

主机呼叫:

<02> <11>

站地址 功能码

从机响应:

<02> <06> <10> <'CCIDUT4000CONM'> <02> <00>

<CRC1> <CRC2>

3、模块内部资源和 PLC 兼容地址空间

模块内部的资源分模块系统参数区、位地址区、控制参数区和报警参数区。

整个地址空间可以容纳在 0^7FFH 的范围内, 并且, 设计成以 $800H$ 为跨度在 0^FFFFH 空间里产生镜像地址。即读写寄存器 $2H$, 和读写寄存器 $802H$, $1002H$, $1802H$, $2002H$,, 都是一样的。

这样做的目的, 是为了兼容多种 PLC, 和他们的地址空间相对应。举例说明: 模块的寄存器地址是从 0 开始, 台达 PLC 的寄存器地址则是从 $1000H$ 开始, LG PLC $K80H$ 的寄存器地址从 $800H$ 开始。这样, 对于通用触摸屏、PC 机等人机界面设备, 不管设成与台达 PLC 通讯, 还是设成与 LG PLC 通讯都能存取模块内的寄存器内容。

模块内部地址空间分配如下表所示,

表 3、模块内部地址空间分配 (位空间)

| 地址 | 功能码 | 功能码 | 操作 |
|------------------|----------|---------|----|
| 模块资源 | 01H, 05H | 02H | |
| 控温总控 | 30H | - | 读写 |
| D0~D7, STB 开关量输出 | 0~7, 8 | - | 读写 |
| IN1~IN4 开关量输入 | 20H~23H | 10H~13H | 只读 |
| 通道 0~7 下限报警 | 18H~1FH | 8~0FH | 只读 |
| 通道 0~7 上限报警 | 10H~17H | 0~7 | 只读 |

表 4、模块内部地址空间分配(寄存器空间)

功能码: 03H, 04H, 06H, 10H

| 参数类别 | 内存地址 | | 内容 | 操作 |
|------|------|-----|----------------|----|
| | 十六进制 | 十进制 | | |
| 测量值 | 00H | 0 | AI0 的测量值 | 只读 |
| | 01H | 1 | AI1 的测量值 | 只读 |
| | 02H | 2 | AI2 的测量值 | 只读 |
| | 03H | 3 | AI3 的测量值 | 只读 |
| | 04H | 4 | AI4 的测量值 | 只读 |
| | 05H | 5 | AI5 的测量值 | 只读 |
| | 06H | 6 | AI6 的测量值 | 只读 |
| | 07H | 7 | AI7 的测量值 | 只读 |
| 系统参数 | 14H | 20 | 波特率字节设置, 低字节有效 | 读写 |
| | 15H | 21 | 传感器选择字节, 低字节有效 | 读写 |
| | 1CH | 28 | 模块地址字节, 低字节有效 | 读写 |

| | | | | |
|-----------|------|-----|-------------------|----|
| | 1DH | 29 | 热电偶参比端校正字节，低字节有效 | 读写 |
| 传感器单独选择字节 | 60H | 96 | AI0 传感器选择字节，低字节有效 | 读写 |
| | 61H | 97 | AI1 传感器选择字节，低字节有效 | 读写 |
| | 62H | 98 | AI2 传感器选择字节，低字节有效 | 读写 |
| | 63H | 99 | AI3 传感器选择字节，低字节有效 | 读写 |
| | 64H | 100 | AI4 传感器选择字节，低字节有效 | 读写 |
| | 65H | 101 | AI5 传感器选择字节，低字节有效 | 读写 |
| | 66H | 102 | AI6 传感器选择字节，低字节有效 | 读写 |
| | 67H | 103 | AI7 传感器选择字节，低字节有效 | 读写 |
| D00 输 | 102H | 258 | 输出参数选择 | 读写 |

| | | | | |
|-----------------------------------|------|-----|------------------|--------|
| 出 控 制 参 数 | 104H | 260 | 采样时间 | 读 写 |
| | 106H | 262 | 控制温度设置值 0.1 度 | 读 写 |
| | 108H | 264 | 控制系数 P | 读 写 |
| | 10AH | 266 | 控制系数 Ti | 读 写 |
| | 10CH | 268 | 控制系数 Td | 读 写 |
| | 10EH | 270 | 起控范围，以度 为单位 | 读 写 |
| D01 输 出 控 制 参 数 | 114H | 276 | 输出参数选择 | 读 写 |
| | 116H | 278 | 采样时间 | 读 写 |
| | 118H | 280 | 控制温度设置值 0.1 度 | 读 写 |
| | 11AH | 282 | 控制系数 P | 读 写 |
| | 11CH | 284 | 控制系数 Ti | 读 写 |
| | 11EH | 286 | 控制系数 Td | 读 写 |
| D02 输 | 120H | 288 | 起控范围，以度 为单位 | 读 写 |
| | 126H | 294 | 输出参数选择 | 读 写 |

| | | | | |
|-----------------------------------|------|-----|------------------|--------|
| 出 控 制 参 数 | 128H | 296 | 采样时间 | 读 写 |
| | 12AH | 298 | 控制温度设置值 0.1 度 | 读 写 |
| | 12CH | 300 | 控制系数 P | 读 写 |
| | 12EH | 302 | 控制系数 Ti | 读 写 |
| | 130H | 304 | 控制系数 Td | 读 写 |
| | 132H | 306 | 起控范围，以度 为单位 | 读 写 |
| D03 输 出 控 制 参 数 | 138H | 312 | 输出参数选择 | 读 写 |
| | 13AH | 314 | 采样时间 | 读 写 |
| | 13CH | 316 | 控制温度设置值 0.1 度 | 读 写 |
| | 13EH | 318 | 控制系数 P | 读 写 |
| | 140H | 320 | 控制系数 Ti | 读 写 |
| | 142H | 322 | 控制系数 Td | 读 写 |
| D04 输 | 144H | 324 | 起控范围，以度 为单位 | 读 写 |
| | 14AH | 330 | 输出参数选择 | 读 写 |

| | | | | |
|-----------------------------------|------|-----|------------------|--------|
| 出 控 制 参 数 | 14CH | 332 | 采样时间 | 读 写 |
| | 14EH | 334 | 控制温度设置值 0.1 度 | 读 写 |
| | 150H | 336 | 控制系数 P | 读 写 |
| | 152H | 338 | 控制系数 Ti | 读 写 |
| | 154H | 340 | 控制系数 Td | 读 写 |
| | 156H | 342 | 起控范围，以度 为单位 | 读 写 |
| D05 输 出 控 制 参 数 | 15CH | 348 | 输出参数选择 | 读 写 |
| | 15EH | 350 | 采样时间 | 读 写 |
| | 160H | 352 | 控制温度设置值 0.1 度 | 读 写 |
| | 162H | 354 | 控制系数 P | 读 写 |
| | 164H | 356 | 控制系数 Ti | 读 写 |
| | 166H | 358 | 控制系数 Td | 读 写 |
| D06 输 | 168H | 360 | 起控范围，以度 为单位 | 读 写 |
| | 16EH | 366 | 输出参数选择 | 读 写 |

| | | | | |
|-----------------------------------|------|-----|------------------|--------|
| 出 控 制 参 数 | 170H | 368 | 采样时间 | 读 写 |
| | 172H | 370 | 控制温度设置值 0.1 度 | 读 写 |
| | 174H | 372 | 控制系数 P | 读 写 |
| | 176H | 374 | 控制系数 Ti | 读 写 |
| | 178H | 376 | 控制系数 Td | 读 写 |
| | 17AH | 378 | 起控范围，以度 为单位 | 读 写 |
| D07 输 出 控 制 参 数 | 180H | 384 | 输出参数选择 | 读 写 |
| | 182H | 386 | 采样时间 | 读 写 |
| | 184H | 388 | 控制温度设置值 0.1 度 | 读 写 |
| | 186H | 390 | 控制系数 P | 读 写 |
| | 188H | 392 | 控制系数 Ti | 读 写 |
| | 18AH | 394 | 控制系数 Td | 读 写 |
| | 18CH | 396 | 起控范围，以度 为单位 | 读 写 |
| D08 输 | 192H | 402 | 输出参数选择 | 读 写 |

| | | | | |
|-----------------------|------|-----|------------------|--------|
| 出 控 制 参 数 | 194H | 404 | 采样时间 | 读 写 |
| | 196H | 406 | 控制温度设置值 0.1 度 | 读 写 |
| | 198H | 408 | 控制系数 P | 读 写 |
| | 19AH | 410 | 控制系数 Ti | 读 写 |
| | 19CH | 412 | 控制系数 Td | 读 写 |
| | 19EH | 414 | 起控范围，以度 为单位 | 读 写 |
| AI0 报 警 | 1A8H | 424 | 绝对报警上限值 | 读 写 |
| | 1AAH | 426 | 绝对报警下限值 | 读 写 |
| | 1ACH | 428 | 相对报警，为 0 绝对报警 | 读 写 |
| AI1 报 警 | 1B2H | 434 | 绝对报警上限值 | 读 写 |
| | 1B4H | 436 | 绝对报警下限值 | 读 写 |
| | 1B6H | 438 | 相对报警，为 0 绝对报警 | 读 写 |
| AI2 报 警 | 1BCH | 444 | 绝对报警上限值 | 读 写 |
| | 1BEH | 446 | 绝对报警下限值 | 读 写 |

| | | | | |
|---------------|------|-----|-------------------|--------|
| | 1C0H | 448 | 相对报警, 为 0 绝对报警 | 读 写 |
| AI3 报 警 | 1C2H | 454 | 绝对报警上限值 | 读 写 |
| | 1C4H | 456 | 绝对报警下限值 | 读 写 |
| | 1C6H | 458 | 相对报警, 为 0 绝对报警 | 读 写 |
| AI4 报 警 | 1CCH | 464 | 绝对报警上限值 | 读 写 |
| | 1CEH | 466 | 绝对报警下限值 | 读 写 |
| | 1D0H | 468 | 相对报警, 为 0 绝对报警 | 读 写 |
| AI5 报 警 | 1D6H | 474 | 绝对报警上限值 | 读 写 |
| | 1D8H | 476 | 绝对报警下限值 | 读 写 |
| | 1DAH | 478 | 相对报警, 为 0 绝对报警 | 读 写 |
| AI6 报 警 | 1E0H | 484 | 绝对报警上限值 | 读 写 |
| | 1E2H | 486 | 绝对报警下限值 | 读 写 |
| | 1E4H | 488 | 相对报警, 为 0 绝对报警 | 读 写 |
| AI7 报 | 1EAH | 494 | 绝对报警上限值 | 读 写 |

| | | | | |
|--------------|------|-----|-------------------|----|
| 警 | 1ECH | 496 | 绝对报警下限值 | 读写 |
| | 1EEH | 498 | 相对报警, 为 0 绝对报警 | 读写 |
| DI 报 警 | 1F4H | 504 | 非 0, 请求输出 有效 | 读写 |
| | 1F6H | 506 | 非 0, 外启动控 温 | 读写 |
| | 1F8H | 508 | 未用 | 读写 |

(四) 松下兼容通讯协议

1、通讯的块格式:

| | | | | | | |
|---|--------|--------|--------|--------|-----|----|
| % | 机 号 | 标 识 | 命 令 | 数 据 | BCC | CR |
|---|--------|--------|--------|--------|-----|----|

由“%”字符开始, 由回车符“CR”结束, 中间由一个字符的机号、标识符、命令、数据和 BCC 校验组成。BCC 校验为从“%”开始到数据结束所有字符异或后形成一个字节数据, 将这个数据转换成两个 16 进制字符作为 BCC 校验值。参见松下 PLC 手册。

2、通讯命令:

(1) 主机发送:

| | | | | | | | |
|---|---|---|---|---|-----|----|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| % | C | # | R | D | BCC | CR | |

地址号默认为字符“C”(ASCII 码为 43H), 可以用设置程序改变。

(2) 从机响应:

正确响应:

| | | | | | | | | | |
|---|---|----|---|---|---|----|-----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 45 | 46 | 47 | 48 |
| % | C | \$ | R | D | 数 | 据 | BCC | CR | |

数据为 8 个通道的当前数据。每个通道数据为 5 位 ASCII 码，值为-9999~99999，表示温度值为-999.9℃~9999.9℃，共 8 个通道占 40 字节。命令串的长度为 48 字节。若输入开路，输出值为-9999。

错误响应：

| | | | | | | | | | |
|---|---|----|---|---|---|---|-----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| % | C | \$ | R | D | 0 | 1 | BCC | CR | |

（五）研华 ADAM 兼容通讯协议

1、命令集

表 3 研华兼容协议命令集

| 命令语法 | 命令响应 | 功能 | 说明 |
|------------|---------------|------|------------|
| %AANN (CR) | ! AA (CR) | 修改地址 | 改模块地址为 NN |
| #AAN (CR) | >(data) (CR) | 读数 | 读通道 N 的值 |
| #AA (CR) | >(data) (CR) | 读数 | 读取 8 个通道数据 |
| \$AA2 (CR) | !AATCCFF (CR) | 读配置 | 取波特率 |
| \$AA3 (CR) | !AASS (CR) | 读传感器 | 模块传感器 |
| \$AA6 (CR) | !AAFF (CR) | 通道状态 | 恒为 FF |
| \$AAM (CR) | !AA4017 (CR) | 模块名 | 为 4017 |
| \$AAF (CR) | !AAD1.0 (CR) | 软件版本 | 为 D1.0 |

说明：研华协议兼容研华 ADAM4017 命令集，但与 ADAM4017 的差别是：无奇偶校验，自动

识别累加和。

命令详述:

(1) %AANN

名称: 修改模块地址

语法: %AANN (CR)

%为定界符

AA (范围 00⁻FFH) 表示要修改模块的两位十六进制地址

NN 将模块修改为新的十六进制地址, 地址范围 00⁻FFH

(CR) 为结束符, 即回车 (0DH)

回答: 如果命令有效: ! AA (CR)

! 定界符, 表示收到有效命令

AA (范围 00⁻FF) 新的输入模块的两位十六进制地址

(CR) 为结束符, 即回车 (0DH)

注: 在通讯命令中, 如果模块检测到语法错、通讯错或指定的地址不存在, 则没有回答。

(2) #AAN

名称: 读模块中通道 N 的数值

说明: 本命令从模块 (地址为 AAH) 8 个通道中的一个读回数据

语法: #AAN (CR)

#为定界符

AA (范围 00⁻FF) 表示模块的两位十六进制地址

N 为将要读出的通道号, N 值为 0⁻7

(CR) 为结束符, 即回车 (0DH)

回答: >(data) (CR)

>为定界符

(data) 是通道号为 N 的测量数据值, 该数据以 “+” 或 “-” 开头, 后面跟 4 位十进制整数和一位小数 (温度 0.1℃分辨率输出), 或以 “+” 或 “-” 开头后面跟 6 位十进制整数 (电压、电流和 0.01℃分辨率温度输出)。传感器开路输出 -0999.9 或 -009999

(CR) 为结束符, 即回车 (0DH)

例、命令: #430 (CR)

回答: >+0408.6 (CR)

本命令请求从地址为 43H 的模块读回通道 0 的温度值。模块应答通道 0 的温度值为+408.6℃。

(3) #AA

名称: 读模块中 8 个通道的数值

说明: 本命令从模块 (地址为 AAH) 读回 8 个通道的数据

语法: #AA (CR)

#为定界符

AA (范围 00~FF) 表示模块的两位十六进制地址

(CR) 为结束符, 即回车 (0DH)

回答: >(data) (CR)

>为定界符

(data) 是 8 个通道当前的测量数据值。每个通道的数据格式与 #AAN 命令中对 (data) 的说明相同

(CR) 为结束符, 即回车 (0DH)

例、命令: #43 (CR)

回答: >+0408.6+0408.6+0408.6+0408.6+0408.6+0408.6
+0408.6+0408.6 (CR)

本命令请求从地址为 43H 的模块读回 8 个通道的温度值。模块应答 8 个通道的温度值均为+408.6℃。

(4) \$AA2

名称: 读模块的配置状态

说明: 本命令从地址 AA 的模块读回配置数据

语法: \$AA2 (CR)

\$为定界符

AA (范围 00~FF) 表示要查询模块的两位十六进制地址

2 为读配置状态命令

(CR) 为结束符，即回车 (0DH)

回答：如果命令有效：! AATTCCFF (CR)

! 为定界符

AA (范围 00~FF) 表示输入模块的两位十六进制地址

TT 类型码，本模块返回值为 0BH

CC 表示波特率代码，如表 4 所示

表 4 研华协议通讯波特率码表

| 代码 | 波特率 | 代码 | 波特率 |
|----|------|----|-------|
| 03 | 1200 | 06 | 9600 |
| 04 | 2400 | 07 | 19200 |
| 05 | 4800 | 08 | 38400 |

FF 数据格式，本模块返回 80H

例、命令：\$432 (CR)

回答：!430B0680 (CR)

地址为 43H (字符 C)，波特率为 9600bps。

(5) \$AA3

名称：读 DUT 模块传感器类型

说明：本命令从地址 AA 的模块读回传感器类型码

语法：\$AA3 (CR)

\$为定界符

AA (范围 00~FF) 表示要查询模块的两位十六进制地址

3 为读传感器类型码命令

(CR) 为结束符，即回车 (0DH)

回答: 如果命令有效: ! AASS (CR)

! 为定界符

AA (范围 00~FF) 表示模块的两位十六进制地址

SS 用十六进制码表示的传感器类型码, 参见表 1

例、命令: \$433 (CR)

回答: !430D (CR)

滤波方式, PT100 传感器温度采集模块。

(6) \$AA6

名称: 读通道状态

说明: 本命令从地址 AA 的模块读回 8 个通道状态

语法: \$AA6 (CR)

\$为定界符

AA (范围 00~FF) 表示要查询模块的两位十六进制地址

6 为读通道状态命令

(CR) 为结束符, 即回车 (0DH)

回答: 如果命令有效: ! AAVV (CR)

! 为定界符

AA (范围 00~FF) 表示输入模块的两位十六进制地址

VV 为 8 位二进制值, 分别表示 8 个通道状态, “0” 表示通道禁止, “1” 表示通道允许,

本模块返回 FFH

(7) \$AAF

名称: 读防火墙软件版本号

说明: 本命令从地址 AA 的模块读防火墙版本号

语法: \$AAF (CR)

\$为定界符

AA (范围 00~FF) 表示要查询模块的两位十六进制地址

F 为读防火墙版本号命令

(CR) 为结束符，即回车 (0DH)

回答：如果命令有效：! AAD1.0 (CR)

防火墙软件版本被设为 1.0

(8) \$AAM

名称：读模块名称

说明：本命令从地址 AA 的模块读模块名

语法：\$AAF (CR)

\$ 为定界符

AA (范围 00~FF) 表示要查询模块的两位十六进制地址

M 为读模块名命令

(CR) 为结束符，即回车 (0DH)

回答：如果命令有效：! AA4017 (CR)

本模块名定为 4017，命令完全兼容 ADAM4017

2、测试、修改地址程序 DUTADAM

DUT-4000 系列温度、电流、电压采集模块支持研华协议时，命令完全兼容研华 ADAM4017，软件 DUTADAM.EXE 用以测试模块和修改地址。也可由 ADAM.EXE 或 ADAM304.EXE 测试和修改模块地址。出厂时模块各个通道的零点和满度已经校准好，足以保证测量精度要求，用户无需再校准。

3、GENIE 软件使用

研华的组态软件 GENIE 可直接读取模块数据，使用方法和 ADAM4017 完全相同。温度数据含一个小数位，单位为℃；电流和电压数据为码数，标度变换可以由驱动程序完成。传感器悬空输出值为-999.9℃。DUT 模块自动识别检验和，在 GENIE 软件中可根据需要设置。另外，组态王等一些组态软件也支持这种通讯协议，用户可参阅相应组态软件手册进行配置。

七、模块控温组态、逻辑输出组态及其应用

1、控温的总控

模块控温功能受到控温总控位[地址 48 (30H)]的控制，该位设为 0 时，停止所有的控温运算，保留运算值，所有设为控温的开关量输出点逻辑值为 0，即 OC 输出开路高电平，停止加热器加热。该位设为 1 时，继续正常控温。

模块上电时，如果 DI 报警下限寄存器[地址 506 (1F6H)]为 0，则控温总控位上电为 1，如果该寄存器非 0，模块上电时控温总控位上电为 0 不进入控温状态，需要人机界面将控温总控位设为 1，启动控温。

2、设置控温前的准备

模块设置为控温前，必须将异步并行输出设为请求方式，禁止并行输出口输出脉冲。将 DI 报警上限寄存器[地址 505 (1F5H)]设为 0，禁止 IN4 的异步并行输出请求功能。

但是，如果只用到 D4⁻D7 输出口控温，D0⁻D3, STB 仍然可以用作异步并行口输出温度数据。

3、控温和逻辑输入输出组态

对 9 个输出量 D0 选择，除用调功输出完成控制功能外，还可以选择 AI 上下限和 4 位开关量输入之间的逻辑操作后输出。每个开关量输出由以下结构体定义相应的控制参数。

```
struct
{
    int CtrlSel    //控制选择变量，低字节有效，高字节为 0
    int SampleT    //采样时间 Ts，以 80mS 为单位，低字节有效，高字节为 0
    int Set_Val    //设置控制温度值，整型数，乘以 10 的温度值
    int SeriesD0   //PID 比例系数 P，整型数
    int SeriesD1   //PID 积分时间 Ti，以 80mS 为单位
    int SeriesD2   //PID 微分时间 Td，以 80mS 为单位
    int CtrlArea   //起控范围，以度为单位
} Out [9]
```

控制变量选择 CtrlSel 定义如下：

CtrlSel

| D7 | D6 | D5 | D4 | D3 - D0 |
|----|----|----|----|----------|
| 0 | 保留 | 控制 | 算法 | 输入的 AI 号 |

控制算法选择:

| D5 | D4 | 控制算法 | 表达式 |
|----|----|--------|--|
| 0 | 0 | 不控制 | |
| 0 | 1 | PID 控制 | $U = P * E_i + (T_s / T_i) * \text{SUM}(E_i) + (T_d / T_s) * (E_i - E_{(i-1)})$ (起控范围内) |
| 1 | 0 | 模糊控制 | 控制表查找索引 $\text{Index} = P * E_i + T_i * (E_i - E_{(i-1)})$ 范围: $-7 \sim +7$ |
| 1 | 1 | 增量 PID | $U_i = U_{(i-1)} + P * E_i + T_i * E_{(i-1)} + T_d * E_{(i-2)}$ |

D3~D0 用来选择 8 路模拟输入 (AI) 通道。

如果选择开关量输入及 AI 上下限作为输入对某一开关量输出进行组态, 则定义如下:

CtrlSel

| | | | | |
|----|----|----|----|---------|
| D7 | D6 | D5 | D4 | D3 - D0 |
| 1 | 整体 | 上限 | 下限 | 输入的字节数 |

D7=1 开关量输出选择

D6=1 报警输入和输出全部参与输出

SampleT 的低字节为屏蔽字节，bit0 对应 AI0，bit7 对应 AI7，如果某个 bit 为 1 则对应 AI 通道被屏蔽，其上下限报警都不参与运算。

D5=1 AI 上限报警参与运算

D4=1 AI 下限报警参与运算

D3=1 输出取反

D3=0 输出不变

D2=0 上限、下限之间“或”运算

D2=1 上限、下限之间“与”运算

D1=0 各 AI 下限报警之间“或”运算

D1=1 各 AI 下限报警之间“与”运算

D0=0 各 AI 上限报警之间“或”运算

D0=1 各 AI 上限报警之间“与”运算

D7=1 控制参数选择

D6=0 由后面字节定义输入和运算

D3~D0 为后面参与运算的字节数（最多 6 个）

D6=1 时，SampleT 的低字节为屏蔽字节。

在 CtrlSel 后每个字的低字节可以选择一个逻辑输入，低字节定义如下（高字节未用）：

| | | | | | | | |
|----|-------|----|------|----|----|----|----|
| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 有效 | 与 / 或 | 取反 | 选择输入 | | | | |

D4⁻D0 选择逻辑输入:

0⁻7: AI 上限

8⁻15: AI 下限

16⁻19: IN1⁻IN4

D5 取反

0: 输入值直接参与运算

1: 输入值取反参与运算

D6 逻辑

0: 与前面进行或逻辑运算

1: 与前面进行与逻辑运算

D7 有效

0: 该字节无效

1: 该字节有效

4、上下限报警设置

DUT6000 系列控制模块有 8 个 AI 和 4 路 DI, 每个 AI 可以单独设置上限和下限报警值, DI 的上限报警低字节非 0 时, 请求输出才有效。上下限定义由以下结构体定义:

```
Struct
{
    int AIUpLmt      //上限报警值
    int AIDownLmt //下限报警值
    int AIOut        //保留字
} AlarmValue[9]
```

其中: 结构体数组 AIUpLmt 和 AIDownLmt 分别定义 AI0⁻7 上下限报警值, AIOut 定义相对报警值, 如果 AIOut 为 0, 则该通道报警由绝对报警值 AIUpLmt 和 AIDownLmt 设定, 如果 AIOut 不为 0, 则报警值为相对报警值, 上限报警值为 Set_Val+AIOut, 下限报警值为 Set_Val-AIOut。8 定义 DI 的请求输出是否有效, AlarmValue[8].AIUpLmt 的低字节为 0 则并口请求输出无效, 非 0 时请求输出有效。AlarmValue[8].AIDownLmt 的低字节为 0 则上电后就开始控温, 非 0 时写入启动控温命令后, 控温开始。

5、 控温举例

以三温区回流焊控温系统应用，举例说明模块控温参数的设置，该系统框图如下：

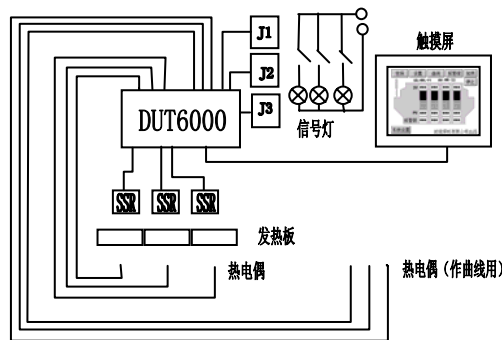


图 8 . 系统框图

DUT 模块的 AI0~AI2 用于三个温区的控温，AI3~AI5 用于拖偶，测工艺曲线，D0~D2 连接三温区的固态继电器，D5~D7 作为逻辑输出控制三个信号灯。D7 输出超温信号，即三温区上限报警的或逻辑，D6 输出升温信号，即三温区下限报警的或逻辑，D5 输出恒温信号，即三温区上下限报警或运算之后取反输出。各通道参数设置如下：

开出通道 D0: 温区一控温

```
int CtrlSel      = 16, PID 控温, 选择 AI0 温度通道;
int SampleT     = 250, 以 80mS 为单位, 采样时间为 20 秒;
int Set_Val     = 2800, 设定温度 280 度;
int SeriesD0    = 11 比例值
int SeriesD1    =1300 积分时间 Ti, 以 80mS 为单位
int SeriesD2    =10000 微分时间 Td, 以 80mS 为单位
int CtrlArea    =100 起控范围 100 度
```

开出通道 D1: 温区二控温

```
int CtrlSel      = 17, PID 控温, 选择 AI1 温度通道;
int SampleT     = 250, 以 80mS 为单位, 采样时间为 20 秒;
int Set_Val     = 1700, 设定温度 170 度;
int SeriesD0    = 7 比例值
int SeriesD1    =2500 积分时间 Ti, 以 80mS 为单位
int SeriesD2    =10000 微分时间 Td, 以 80mS 为单位
int CtrlArea    =100 起控范围 100 度
```

开出通道 D2: 温区三控温

```
int CtrlSel      = 18, PID 控温, 选择 AI2 温度通道;  
int SampleT     = 250, 以 80mS 为单位, 采样时间为 20 秒;  
int Set_Val     = 2000, 设定温度 200 度;  
int SeriesD0    = 8 比例值  
int SeriesD1    = 3000 积分时间 Ti, 以 80mS 为单位  
int SeriesD2    = 8000 微分时间 Td, 以 80mS 为单位  
int CtrlArea    = 100 起控范围 100 度
```

开出通道 D3~D4: 不用

```
int CtrlSel      = 0, 不控温, 也不参与逻辑运算;
```

开出通道 D5: 输出恒温信号

```
int CtrlSel      = 248 (F8H), 全部参与运算, 上限报警参与运算, 下限报警参与运算, 上限之间或运算, 下限之间或运算, 上下限之间或运算, 输出取反。  
int SampleT     = 248 (F8H), 屏蔽字节, 将高 5 位屏蔽;
```

开出通道 D6: 输出升温信号

```
int CtrlSel      = 208 (D0H), 全部参与运算, 下限报警参与运算, 下限之间或运算, 上下限之间或运算, 直接输出。  
int SampleT     = 248 (F8H), 屏蔽字节, 将高 5 位屏蔽;
```

开出通道 D7: 输出超温信号

```
int CtrlSel      = 224 (E0H), 全部参与运算, 上限报警参与运算, 上限之间或运算, 上下限之间或运算, 直接输出。  
int SampleT     = 248 (F8H), 屏蔽字节, 将高 5 位屏蔽;
```

八、控温模块与温度数据采集

DUT6000 控温模块完全兼容原 DUT3000 和 DUT4000 的全部功能, 可用于采集温度电流电压数据。

1、通过串行口采集数据

模块与计算机通过串行口连接如图 6 所示。计算机的串行口 1 或串行口 2 通过 RS232 到

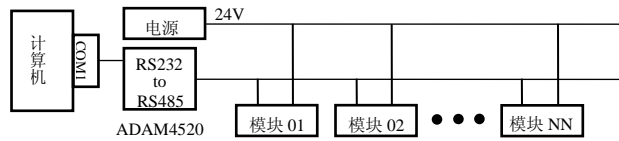


图 6 采集模块连接图

RS485 转换器（可以选用 ADAM4520）转换为 RS485 总线，各个采集模块以 RS485 总线形式和计算机相连。ADAM4520 的 DATA+和 DUT 模块的 T+相连，DATA-与 T-相连。+24V 电源也对应连接。一般一个系统可直接连接 32 个模块，超过 32 个需要加中继器，总线长度不超过 1200 米，总线两端最好加终端器。

在工作状态下，主机仅从 DUT 模块中读取数据。即主机发送读数据命令串，模块返回当前数据。模块响应时间一般小于 150ms（9600 波特时）。若超过 150ms 没有响应，可以重发。连续三次没有响应，进行错误告警。随产品提供各种语言数据采集源程序，这些程序也可以访问我们的网页 <http://www.dlia.com.cn> 得到。

2、通过异步并行接口采集温度数据

(1)、隔离异步并行接口输出时序及应用

模块内有一波特率控制字除用以选择串行通讯波特率和奇偶校验外，还控制 DUT-4000 的并行接口的输出时序。

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|----|----|----|----|----|----|----|-------|
| 校 | 请 | 字 | 通 | 通 | | | |
| 验 | 求 | 节 | 讯 | 协 | | | |
| | | | 议 | | | | 选择波特率 |

其中，D7=0 串行通讯无校验方式；D7=1 串行通讯奇校验方式。D6=0 并行接口无条件输出，每 2.16 秒（不滤波 0.72S）输出 8 个通道数据；D6=1 并行接口请求输出，IN+和 IN-为 ON 请求输出一次数据。D5=0 并行接口半字节输出，每次输出 4 位二进制数；D5=1 并行接口字节输出，每次输出 8 位二进制数。D4⁻D3 选择通讯协议。D2⁻D0 选择串行通讯波特率。

(2)、无条件半字节输出时序

当模块内波特率控制字的 D6=0、D5=0 选择并行无条件半字节输出，接口时序如图 7 和图 8 所示，选通脉冲 STB 可以是上升沿选通或下降沿选通，由板上的 DIP 开关 S4 选择。S4=OFF，上升沿选通（默认状态）；S4=ON，下降沿选通。数据由 D3⁻D0 输出，每个半字节（4 位二进制）

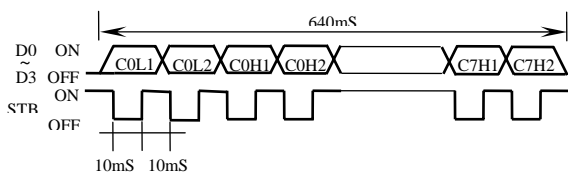


图 7 无条件半字节输出上升沿选通时序

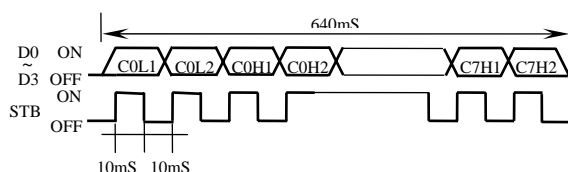


图 8 无条件半字节输出下降沿选通时序

输出时间为 20mS（默认），选通脉冲 STB 高电平和低电平时间各为 10mS。每个通道数据分 4 次输出，依次由低到高。数据为两个字节二进制补码，表示温度乘 10 的数据。每次连续输出 8 个通道共 16 个字节，输出时间为 640mS。模块在滤波工作方式下每 2.16S 转换完 8 个通道数据，然后按上述时序输出。不滤波方式下 0.72S 输出一次数据。并行接口的输出时间可以由设置程序设置，参见 DUTSET 说明。

(3)、无条件字节输出时序

当模块内波特率控制字的 D6=0、D5=1 选择并行无条件字节输出，接口时序如图 9 和图 10 所示。选通脉冲 STB 可以是上升沿选通或下降沿选通，由板上的 DIP 开关 S4 选择。S4=OFF，上升沿选通（默认状态）；S4=ON，下降沿选通。数据由 D7-D0 输出，每个字节输出时间为 20mS（默认），选通脉冲 STB 高电平和低电平时间各为 10mS。每个通道数据为两个字节二进制补码表示的有符号数，表示温度乘 10 的数据，先输出低位，然后输出高位。每次连续输出 8 个通道共 16 个字节，输出时间为 320mS。滤波工作方式下模块每 2.16S 转换完 8 个通道数据，然后按上述时序输出。不滤波方式下 0.72S 输出一次数据。并行接口的输出时间可以由设置程序设置，参见 DUTSET 说明。

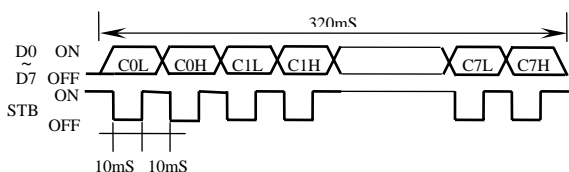


图 9 无条件字节输出上升沿选通时序

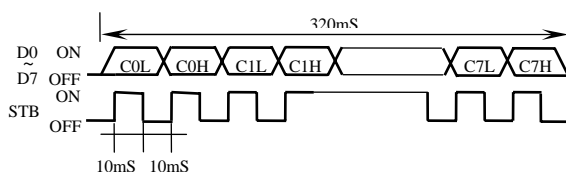


图 10 无条件字节输出下降沿选通时序

(4)、请求半字节输出时序

当模块内波特率控制字的 D6=1、D5=0 选择并行请求半字节输出，接口时序如图 11 所示，请求信号 IN（IN+与 IN-）由 OFF 到 ON 引起请求输出。选通脉冲 STB 可以是上升沿选通或下降沿选通，由板上的 DIP 开关 S4 选择。S4=OFF，上升沿选通（默认状态）；S4=ON，下降沿选通。

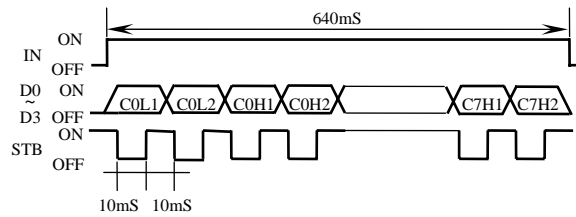


图 11 请求半字节输出上升沿选通时序

数据由 $D3\bar{D}0$ 输出，每半个字节输出时间为 20mS（默认），选通脉冲 STB 高电平和低电平时间各为 10mS。每个通道数据分 4 次输出，依次由低到高。数据为两个字节二进制补码表示的有符号数，表示温度乘 10 的数据。每次连续输出 8 个通道共 16 个字节，输出时间为 640mS。并行接口的输出时间可以由设置程序设置，参见 DUTSET 说明。

(5)、请求字节输出时序

当模块内波特率控制字的 $D6=1$ 、 $D5=1$ 选择并行请求字节输出，接口时序如图 12 所示，请求信号 IN（IN+与 IN-）由 OFF 到 ON 引起请求输出，选通脉冲 STB 可以是上升沿选通或下降沿选通，由板上的 DIP 开关 S4 选择。S4=OFF，上升沿选通（默认状态）；S4=ON，下降沿选通。数据由 $D7\bar{D}0$ 输出，每个字节输出时间为 20mS，选通脉冲 STB 高电平和低电平时间各为 10mS。每个通道数据为两个字节二进制补码表示的有符号数，表示温度乘 10 的数据，先输出低位，然后输出高位。每次连续输出 8 个通道共 16 个字节，输出时间为 320mS。并行接口的输出时间可以由设置程序设置，参见 DUTSET 说明。

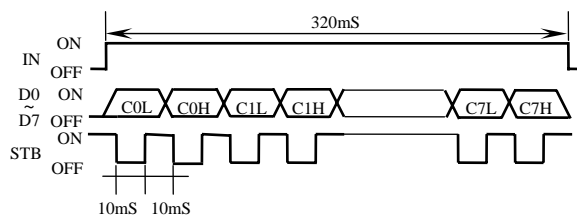


图 12 请求字节输出上升沿选通时序

(6)、模块与松下 PLC（FP1-C40）连接采集数据

DUT-4000 系列采集模块以半字节无条件数据传送方式与 PLC（松下 FP1-C40）硬件连接如图 13 所示， $D0\bar{D}3$ 连接到 PLC 的 $X8\bar{X}11$ ，STB 接 PLC 的 X7，X7 应能引起中断，电源直接取 PLC 的 24V 接到模块的 V+和 V-。

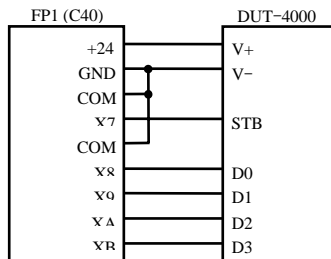


图 13 DUT-4000 模块与松下 PLC 连接图

编程使 PLC 的 X7 每次的上升沿（或下降沿）引起中断，在中断程序中读入 X11~X8 的数据，根据读入的次数和通道计数，将数据存入对应通道寄存器的低半字节到高字节，直到读入 8 通道 32 次数据，图 14 为 FP1 的数据采集程序梯形图，程序运行前将系统寄存器 NO.403 的 X7 设为中断输入。PLC 内数据为二进制有符号数补码，不用转换，该数即为温度乘 10 的值。为保证同步的可靠性，使用一个定时器 T0 定时 700mS

（在 640mS~2.16S 之间）监视选通脉冲

STB，若定时器溢出，则这次输出已经结束，

下一个选通脉冲为第一个数据。

由于松下 PLC 扫描后才刷新输入，然后才引起中断，程序长则扫描时间长，刷新间隔超过 20 毫秒时数据无法同步。解决的方法是在主程序中每隔一段时间加入强制刷新命令 IORF (F143)，如图 15 所示。每隔 300~400 步，加入 IORF 指令，读取的数据就比较可靠。

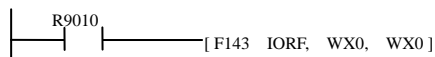
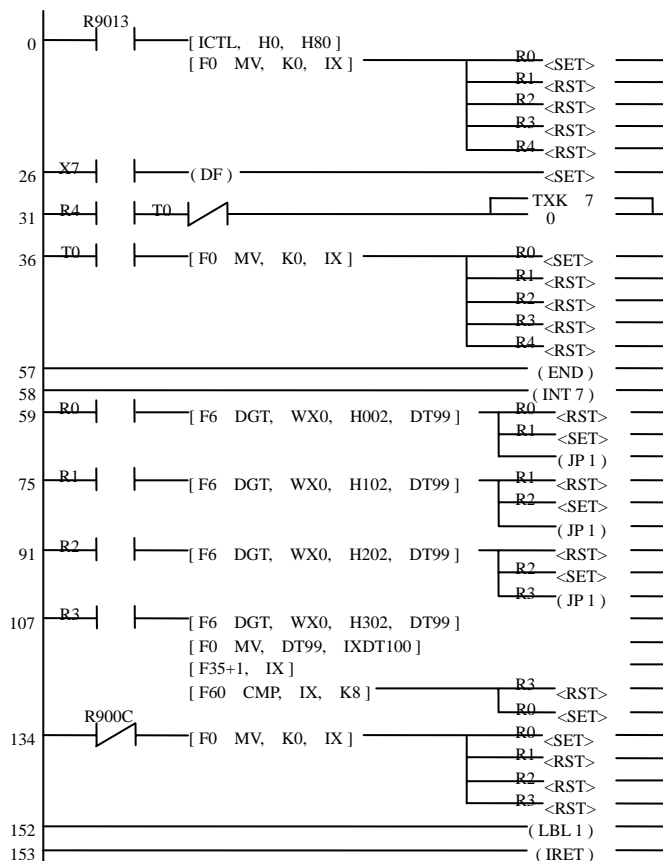


图 15 刷新指令



(7)、模块与西门子 S7-226 连接采集数据

硬件连接如图 16 所示，将 DUT-4000 模块的 STB 连接到 S7 的 IO.0，D0~D3 连接到 S7 的

I1.0~I1.3, S7 的+24V 电源和 GND 连接到 DUT 模块的 V+和 V-, S7 的 M1 连接到 GND。

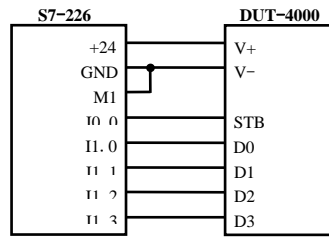


图 16 DUT-4000 模块与西门子 PLC 连接图

编程 I0.0 使其在上升沿产生中断。在中断程序中，根据顺序采集数据。用定时器 T33 定时 30mS 检测 I0.0 的脉冲，如果 30mS 不来脉冲，则本次数据采集结束。在程序中用字节 VB320 的低四位，指示每个通道数据的半字节顺序，字 VW324 采集半字节，字 VW322 为每个通道数据暂存变量，由 VW324 的半个字节组成通道数据。VD316 为间址存储器，通过间接寻址将采集的各个通道数据存储存储在 VW300~VW314 中。

结果存储在 VW300~VW314 中，从 0 通道~7 通道顺序存储。温度测量模块采集的为温度乘 10 的值，电流或电压模块采集的值是码数。Siemens S7-226 采集 DUT-4000 模块程序清单如下：

```

//主程序
//PROGRAM COMMENTS
//Press F1 for help and example program
NETWORK1//Acquisition Module Program
//
//NETWORK COMMENTS
//
LD SM0.1
ATCH INT_0,0
MOVD &VB300,VD316
MOVB 1,VB320
ENI
//中断 0 程序
NETWORK1 //Interrupt 0 for Collect Data
//
//NETWORK COMMENTS
//
LD SM0.0
BIR IB1,VB325
ANDW 16#000F,VW324
NETWORK2
LD V320.0
MOVW VW324,VW322
SLB VB320,1
CRETI
NETWORK3
LD V320.1
SLW VW324,4
ORW VW324,VW322
SLB VB320,1
CRETI
NETWORK4
LD V320.2
SLW VW324,8
ORW VW324,VW322
SLB VB320,1
CRETI
NETWORK5
LD V320.3
LPS
SLW VW324,12
ORW VW324,VW322
MOVW VW322,*VD316
INCD VD316
AENO
INCD VD316
LRD
MOVB 1,VB320
LPP
CRETI
NETWORK2
LDN T33
EU
MOVD &VB300,VD316
MOVB 1,VB320
NETWORK3
LD I0.0
TOF T33,+3
NETWORK4
LD V320.2
SLW VW324,8
ORW VW324,VW322
SLB VB320,1
CRETI
NETWORK5
LD V320.3
LPS
SLW VW324,12
ORW VW324,VW322
MOVW VW322,*VD316
INCD VD316
AENO
INCD VD316
LRD
MOVB 1,VB320
LPP
CRETI

```

(8)、用计算机的打印机接口采集温度数据

DUT-4000 模块可以通过并行接口与计算机连接采集数据。模块工作在请求半字节输出工作方式，硬件连接如图 17 所示。

注意：只有支持计算机并口的产品允许这样连接。

通过并行接口的计算机采集数据子程序如下，port 为打印机接口地址（一般 LPT1 为 378H），pointer 为数据缓冲区存放模块 8 个通道数据，由主程序申请 8 个元素的整型数组。这个程序在 VC5.0 下调试通过。DUT-4000 模块采集子程序如下：

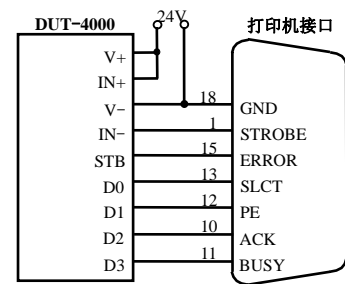


图 17 通过打印机接口与计算机连接

```
#define OVERFLOW 32767
```

```
#include <dos.h>
```

```
void rdboard(short int *pointer, short int port)
```

```
{ unsigned long times=6000000, j=0; // 查询次数
```

```
int i;
```

```
unsigned char *p1, c-byte;
```

```
p1=(unsigned char *)pointer;
```

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

```
pointer[i]=OVERFLOW; // 置不可读数据
```

```
_asm cli;
```

```
_outp(port+2, _inp(port+2) | 0x01); //发请求信号
```

```
j=0;
```

```
times=1000000;
```

```
while(j++<times)
```

```
{ if ((_inp(port+1) & 0x08) != 0) //上升沿选通
```

```
break; }
```

```
for (i=0; i<32; i++)
```

```
{ times=30000;
```

```
j=0;
```

```
while (j++<times)
```

```
{ if ((_inp(port+1) & 0x08) != 0) //高电平读数
```

```
{ if (i%2==0)
    c_byte=(-inp(port+1)^0x80); //最高位电路反相
else
    p1[i/2]=((-inp(port+1)^0x80)&0xf0 | ((c_byte>>4)&0xf); //高半字节
break; }
}
if (j>=times) break;
times=30000;
j=0;
while (j++<times)
{ if ((-inp(port+1)&0x08)==0) //低电平等待
    break; }
if (j>=times) break;
}
-outp(port+2, -inp(port+2)&0xfe); //清请求信号
-asm sti;
}
```

九、模块设置程序 DUTSETW 使用说明

DUTSET 程序使用之前要将 DUT 系列模块设置成设置状态，要将 SET 端与地端 V-连接在一起。只有在这种状态下，才能够用 DUTSETW 程序设置。

- 1、按图 6 所示连接好连线后，将需要设置的模块置成设置方式，**注意，只能有一个模块处于设置方式。**
- 2、运行 DUTSET 程序，屏幕上显示系统参数分别为：系列号 (Series No.)、模块地址 (Slave Number)、电路零偏 (Circuit zero value)、模块识别号 (Module identify)、通讯波特率和校验字节 (baud and check)、模块软件版本号 (Module version)、传感器类型码 (Sensor, Cold and filter)、测量方式 (Measurement mode)、制造日期 (Manufacture date)、制造时间 (Manufacture time) 和并行接口输出一位的一半时间 (Pout half time) 等。然后显示每个通道零点校准值 (Chnl Zero)、满度校准值 (Chnl Full)、当前测出的 8 个通道的数据 (Chnl value) 以及各个通道的传感器类型。

3、系统参数设置

按 F7 键，进入系统参数的设置。依次显示：

(1) 输入模块地址

显示：Enter New Address: (Old=67)，后面以兰底黄字显示当前的模块地址，如果不修改，直接按回车。修改时要以十进制形式输入的地址，输入后按回车。接着输入波特率码。

(2) 输入波特率码

显示：Enter Baudrate Code (3)，括弧中是原来的波特率码，后面以兰底黄字显示当前的波特率码，前面显示各种速率的波特率码。波特率码除选择通讯速率 (bps) 外，还控制串行通讯的校验和并行接口输出方式 (参见七)。根据需要选择各位 “0” 或 “1”，然后转换成十进制输入，按回车。接着输入传感器类型码。

(3) 传感器类型码

显示：Enter Sensor Code (13)，后面以兰底黄字显示当前的传感器类型码，在上面显示各种传感器和类型码对应关系。传感器类型码除选择传感器外还控制热电偶的冷端补偿 (参见五)。根据需要选择各位 “0” 或 “1”，然后转换成十进制输入，按回车。接着输入冷端补偿误差。

(4) 输入冷端补偿 AD590 零点误差

显示：Enter AD590 Zero error (0)，后面显示当前冷端补偿 AD590 的零点误差，如果有误差，可以调节该数。当前显示值是测量值加上该值。显示值是温度乘 10 的数据。接着输入并行接口输出时间。

(5) 输入并行接口输出一位时间。

显示：Input Parallel output half digit time (uS): 10001，该值以微秒为单位，并行接口输出时间的一半时间 (默认 10mS)。在用查询方式采集模块数据时，有时需要修改该值来改变并行接口输出数据的速度。

输入完成后，程序提示是否接受这些参数，按 “y”，接受参数，如果按其他键，可以重新输入这些参数。

4、每个通道传感器单独设置

如果系统参数中传感器类型码是 16，可以用 F10 对每个通道传感器进行单独设置。传感器类型码如表 1 所示。当模块是热电阻传感器时，PT100、Cu50 和 Cu100 三种传感器可以

由软件选择；当模块是热电偶传感器时，各种型号热电偶传感器能够由软件选择，还可以将热电偶模块设置成电压输入，此时测试的是 mV 信号；标准电流和电压输入的模块不能用于其它传感器输入。

电 话：0411-84732220 / 84732221

传 真：0411-84732225

邮 编：116023

网 址：www.dlia.com.cn

电子邮箱：service@dlia.com.cn

地 址：大连市高新园区七贤岭学子街 2-3 号楼 4 单元 6-4 号

附表 1 热电偶简易分度表

| 温度 (℃) | 微伏 | | | | | |
|-----------|-------|------|-------|-------|-------|-------|
| | S | B | W | T | E | K |
| -200 | | | | -5603 | -8824 | -5891 |
| -150 | | | | -4648 | -7279 | -4912 |
| -100 | | | | -3378 | -5237 | -3553 |
| -50 | -236 | | | -1819 | -2787 | -1889 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 50 | 299 | 2 | 699 | 2035 | 3047 | 2022 |
| 100 | 645 | 33 | 1451 | 4277 | 6317 | 4095 |
| 150 | 1029 | 92 | 2250 | 6702 | 9787 | 6137 |
| 200 | 1440 | 178 | 3089 | 9286 | 13419 | 8137 |
| 250 | 1873 | 291 | 3962 | 12011 | 17178 | 10151 |
| 300 | 2323 | 431 | 4864 | 14860 | 21033 | 12207 |
| 350 | 2786 | 596 | 5788 | 17816 | 24961 | 14292 |
| 400 | 3260 | 786 | 6731 | 20869 | 28943 | 16395 |
| 450 | 3743 | 1002 | 7688 | | 32960 | 18513 |
| 500 | 4234 | 1241 | 8655 | | 36999 | 20640 |
| 550 | 4732 | 1505 | 9629 | | 41045 | 22772 |
| 600 | 5237 | 1791 | 10608 | | 45085 | 24902 |
| 650 | 5751 | 2100 | 11584 | | 49109 | 27022 |
| 700 | 6274 | 2430 | 12559 | | 53110 | 29128 |
| 750 | 6805 | 2782 | 13530 | | 57083 | 31214 |
| 800 | 7345 | 3154 | 14494 | | 61022 | 33277 |
| 850 | 7892 | 3546 | 15451 | | 64924 | 35314 |
| 900 | 8448 | 3957 | 16397 | | 68783 | 37325 |
| 950 | 9012 | 4386 | 17333 | | 72593 | 39310 |
| 1000 | 9585 | 4833 | 18257 | | 76358 | 41269 |
| 1050 | 10165 | 5297 | 19169 | | | 43202 |

| | | | | | | |
|------|-------|-------|-------|--|--|-------|
| 1100 | 10754 | 5777 | 20066 | | | 45108 |
| 1150 | 11348 | 6273 | 20950 | | | 46985 |
| 1200 | 11947 | 6783 | 21820 | | | 48828 |
| 1250 | 12550 | 7308 | 22674 | | | 50633 |
| 1300 | 13155 | 7845 | 23514 | | | 52398 |
| 1350 | 13761 | 8393 | 24339 | | | 54125 |
| 1400 | 14368 | 8952 | 25149 | | | |
| 1450 | 14973 | 9519 | 25943 | | | |
| 1500 | 15576 | 10094 | 26723 | | | |
| 1550 | 16176 | 10674 | 27487 | | | |
| 1600 | 16771 | 11257 | 28236 | | | |
| 1650 | 17360 | 11842 | 28970 | | | |
| 1700 | 17942 | 12426 | 29688 | | | |
| 1750 | 18504 | 13008 | 30391 | | | |
| 1800 | | 13585 | 31079 | | | |

附表 2 热电阻简易分度表

| 温度 (°C) | 铂热电阻 Rt (Ω) | | | | 铜热电阻 Rt (Ω) | |
|------------|-------------|--------|--------|--------|-------------|--------|
| | 新分度号 | | 老分度号 | | 新分度号 | |
| | PT10 | PT100 | BA1 | BA2 | Cu50 | Cu100 |
| -200 | 1.849 | 18.49 | 7.95 | 17.28 | | |
| -150 | 3.971 | 39.71 | 17.85 | 38.80 | | |
| -100 | 6.025 | 60.25 | 27.44 | 59.65 | | |
| -50 | 8.031 | 80.31 | 36.80 | 80.00 | 39.24 | 78.49 |
| -40 | 8.427 | 84.27 | 38.65 | 84.03 | 41.40 | 82.80 |
| -30 | 8.822 | 88.22 | 40.50 | 88.04 | 43.55 | 87.10 |
| -20 | 9.216 | 92.16 | 42.34 | 92.04 | 45.70 | 91.40 |
| -10 | 9.609 | 96.09 | 44.17 | 96.03 | 47.85 | 95.70 |
| 0 | 10.000 | 100.00 | 46.00 | 100.00 | 50.00 | 100.00 |
| 10 | 10.390 | 103.90 | 47.82 | 103.96 | 52.14 | 104.28 |
| 20 | 10.779 | 107.79 | 49.64 | 107.91 | 54.28 | 108.56 |
| 30 | 11.167 | 111.67 | 51.45 | 111.85 | 56.42 | 112.84 |
| 40 | 11.554 | 115.54 | 53.26 | 115.78 | 58.56 | 117.12 |
| 50 | 11.940 | 119.40 | 55.06 | 119.70 | 60.70 | 121.40 |
| 100 | 13.850 | 138.50 | 63.99 | 139.10 | 71.40 | 142.80 |
| 150 | 15.731 | 157.31 | 72.78 | 158.21 | 82.13 | 164.27 |
| 200 | 17.584 | 175.84 | 81.43 | 177.03 | | |
| 250 | 19.407 | 194.07 | 89.96 | 195.56 | | |
| 300 | 21.202 | 212.02 | 98.34 | 213.79 | | |
| 350 | 22.997 | 229.97 | 106.60 | 231.73 | | |
| 400 | 24.704 | 247.04 | 114.72 | 249.38 | | |
| 450 | 26.411 | 264.11 | 122.70 | 266.74 | | |
| 500 | 28.090 | 280.90 | 130.55 | 283.80 | | |
| 550 | 29.739 | 297.39 | 138.21 | 300.58 | | |
| 600 | 31.359 | 313.59 | 145.85 | 317.06 | | |

| | | | | | | |
|-----|--------|--------|--------|--------|--|--|
| 650 | 32.951 | 329.51 | 153.30 | 333.25 | | |
| 700 | 34.513 | 345.13 | | | | |
| 750 | 36.047 | 360.47 | | | | |
| 800 | 37.551 | 375.51 | | | | |
| 850 | 39.026 | 390.26 | | | | |

附录 3 组态软件 GENIE 采集模块数据的使用方法

1、启动 GENIE Builder。

2、选择菜单 File/New，新建一个文件。

3、设置设备

(1) 选择 Setup/Devices 进入设备设置窗口；

(2) 按 Add>>按钮，在窗口的下部显示设备列表 (List of Devieces)；

(3) 选择 Advantech COM Devices，然后按 Install 进入串行口设置窗口；

(4) 按 port 按钮，选择工作端口 Comm. port ，设备连接在串行口 1 上；

(5) 按 Add>>按钮，显示设备列表，选择 Advatech ADAM-4000 Modules，然后按 Install 按钮；

(6) 模块类型 (Moduel Type) 选择 4017，模块地址选择 67 (出厂时模块默认地址是 67 [43H])；

(7) 关闭串行口设置窗口和设备设置窗口。

4、任务设计

(1) 进入任务设计窗口 (TASK1)；

(2) 从工具栏中选择 AI 模块添加到任务设计窗口中；

(3) 用鼠标右键点击 AI 模块，进入设置窗口；

(4) 设备 (Devices) 选择 COM1，模块 (Module) 选择 4017，最后通道 (To Channel) 选择 7 (输入通道 07)。

5、显示设计

(1) 进入显示设计窗口 (DISP1)；

(2) 从工具栏中选择 8 个数字显示框到显示窗口中；

(3) 用鼠标右键点击第一个数字显示框，进入显示输入选择窗口；

(4) 用鼠标点击选择按钮 (SELECT)，弹出连接窗口；

(5) 任务/显示 (Task/Display/Virtual) 选择 TASK1，标签名 (TagName) 选择 AI1: AI1，通

道 (Channel) 选择 0 (Output 0);

(6) 重复 (3)~(5) 步, 将其余 7 个显示框分别设置成显示通道 1~7 的数据。

6、程序运行

(1) 将设计好的界面存盘;

(2) 连接好 RS232C/RS485 转换模块 (ADAM4520) 和 DUT-4000 数据采集模块及 +24V 电源, 系统上电;

(3) 选择运行菜单下的开始 (RUN/Start), 程序开始运行, 在 8 个显示框中分别显示 8 个通道数据;

(4) 选择 Run/Stop 停止运行。

附录 4 DUTADAM 程序使用说明

1、功能简介

DUTADAM 是 DUT-4000 系列模块在研华协议下的测试和修改地址程序, 该程序能够测试多个模块, 并在显示窗口的最后一行显示当前模块的 8 个通道的数据。该程序是在 DOS 界面下由 Borlandc3.1 开发的。

2、测试模块

(1) 运行 DUTADAM 程序后, 程序自动按默认参数搜索系统中连接的模块。每找到一个模块后, 显示模块的参数;

(2) 测试程序的默认参数是: 模块连接在串行口 1 (COM1) 上; 通讯波特率是 9600bps; 扫描地址范围是 0~255;

(3) 每找到一个模块后, 从模块中读取相关参数, 显示到模块窗口中。其中有: 模块地址 (Address) 显示测到模块的地址; 研华模块型号 (ADAM) 显示 4017; 模块名 (Name) 显示 DUT 系列模块的型号; 传感器 (Sensor) 类型显示模块对应传感器的类型 (有 PT100、Cu50、Cu100、K、S、B、T、E、J、R、W、A/D、V、I 等); 以及模块是否工作在滤波模式下。

3、修改测试参数

(1) 用 F3 按键修改模块连接的串行口, 每按一次 F3 串口号加 1, 到达串口 4 后又返回到串行口 1, 4 个串行口的参数如下表所示;

| | | | | |
|-----|------|------|------|------|
| 串口号 | COM1 | COM2 | COM3 | COM4 |
| 基地址 | 3F8H | 2F8H | 3E8H | 2E8H |
| 中断号 | 4 | 3 | 4 | 3 |

- (2) 用 F4 修改通讯波特率，默认 9600bps。每按一次 F4 波特率按 1200、2400、4800、9600、19200 的顺序递增；
- (3) 按 F5 键输入扫描的最后地址，默认扫描到 255；
- (4) 按 F6 重新扫描模块，修改参数后要按 F6 键按新的参数扫描模块；
- (5) 扫描完成后，红颜色显示当前模块，在窗口的最下边一行显示从当前模块读取的实时数据。可以用 UP 和 DOWN 键改变当前模块。

4、修改地址

- (1) 用 UP 和 DOWN 键改变要修改地址的模块为当前模块；
- (2) 按回车键 (Enter) 后，显示输入新的地址。输入地址后按回车，屏幕上当前模块的地址号被改变；
- (3) 可以再按 F6 键重新扫描模块，已确认地址修改的有效性。

5、按 ESC 键后程序退出。