

CC2125A 中文版 V 2.10 操作说明

1993年1月

(日本岛电公司FP21/SR25通讯接口中文学习软件)

本资料 and 开发的学习软件, 作为用户学习带RS232C或 RS422通讯接口仪表接口通讯命令的参考, 不足之处请给与指正。

日本岛电公司中国技术服务中心 北京市希曼顿自动化研究所

目 录

- § 1. 软件清单
- § 2. 调节器通讯参数的设置
- § 3. 通讯命令的组成及格式
- § 4. CC2125 V2.10 工作环境
- § 5. CC2125 V2.10 的安装
- § 6. CC2125 V2.10 的使用方法
- § 7. FP21通讯命令的设计特点

CC2125A的通讯学习软件操作说明书目录

- § 1. 软件清单
- § 2. 调节器通讯参数的设置
- § 3. 通讯命令的组成及格式
- § 4. CC2125 V2.10 工作环境
- § 5. CC2125 V2.10 的安装
- § 6. CC2125 V2.10 的使用方法
- § 7. FP21通讯命令的设计特点
- § 8. SR25通讯命令的设计特点
- § 9. #3 盘上的BASICA程序说明
- § 10. 在PC计算机上, 采用BASICA语言, 实现对FP21, SR25数据采集的编程例
- § 11. 附录: A. 通讯串口接线方法
 - B. RS232通讯口的技术数据
 - C. RS422通讯口的技术数据

§1 软件清单

- #1 盘: CC2125A.EXE - 汉化FP21和SR25通讯学习软件(版本2.10)
在CC2125A软盘内, 还提供了下述的应用资料及其它调用文件.
 - C21254.bat - 批文件(调用鼠标驱动及执行文件)
 - C2125.EXE - SR25和FP21英文通讯学习软件(执行文件)
 - BASICA.EXE - COMPAQ.BASIC
 - B1.BAS - BASIC参考的通讯学习软件
 - B21.BAS - BASIC参考的FP21的"D1"命令数据采集软件
 - B25.BAS - BASIC参考的SR25的"DS"命令数据采集软件
 - 232T.BAS - BASIC的PC机232口及先锋RS422口测试软件
 - CC2125.WPS - CC2125A的通讯学习软件操作说明书
 - CMFP21.WPS - CMFP21汉化工业控制软件操作说明书
- #2 ~ #5盘: UCDS2.01 - 汉化磁盘操作系统 (选件)
用户可用WPS或中文WORD STAR 的"N"命令或UCDOS、CCDOS中的TYPE命令,

检查或打印CC2125.txt和CMFP21.txt 文本内容。

提示: 在WPS中, 用"X"命令退出后, 设置SPDOS/600, 可增加屏幕显示宽度。

§2 调节器通讯参数的设置

参照先锋的FP21和SR25中文操作流程图, 手动选择如下设置:

FP21 [66] 窗口设置地址"00", 通讯波特率1200(2400, 4800等)

FP21 [67] 窗口设置"0"方式, 即7位数据位, 偶校验的通讯字符格式。

FP21 [OPM] 窗口设置"COM"通讯工作方式, 此时, FP21的面板COM灯亮。

SR25 [L2-H3]窗口设置地址"02", 通讯波特率1200(2400, 4800, 9600) SR25[L2-H4]窗口设置7位数据位, 偶校验的通讯字符格式。

SR25仅能由上位机控制, 转成通讯方式(COM)。

§3 发送/接收通讯文件格式规约

通讯仪表采用了六个专用ASCII码控制符:

字符名称	英文名称	16进制表示	ASCII码	屏幕显示
读写命令的引导符	STX	02H	CHR\$(2)	
读写命令的结束符	ETX	03H	CHR\$(3)	
建立连接命令的引导符	EOT	04H	CHR\$(4)	
建立连接命令的结束符	ENQ	05H	CHR\$(5)	
正常应答符号	ACK	06H	CHR\$(6)	
不正常应答符号	NAK	15H	CHR\$(&H15)	

BASICA程序例

131 REM 设置六个通讯控制字符

140 STX\$=CHR\$(2): ETX\$=CHR\$(3): EOT\$=CHR\$(4)

150 ENQ\$=CHR\$(5): ACK\$=CHR\$(6): NAK\$=CHR\$(&H15)

建立通讯的连接命令

RS422通讯的连接是下位调节器的接收线高(RDA)和低(RSD)都挂在上位机RS422A口的发送数据总线上, 下位调节器的发送线高(SDA)和低(SDB)都挂在上位机RS422A口的接收数据总线上。平时, 下位调节器的发送线处于高阻关闭态。上位机和下位调节器是按主、从方式进行数据通讯。通讯时, 上位机必需根据调节器窗口设定的区分地址, 通讯数据格式, 设置的波特率, 按图示的通讯连接格式呼叫调节器。调节器在地址符合, 接收字符校验正确后, 将调节器的发送线高(SDA)和低(SDB)转成低阻态, 并回送地址和正常ACK字符串, 表示该调节器已向数据总线开放, 等待进行数字通讯。RS232接口无三态功能, 不能总线并联。



(上位机和调节器连接示意图)

ASICA程序例: 仪表口地址为"00"

200 REM 使用PC COM1口, 设置1200波特, 偶效验, 7位数据, 1停止位, 禁止联络信号.

210 OPEN "COM1: 1200, E, 7, 1, CD, RS, CS, DS" AS #1 :REM 初始化串行口 COM(1)

230 PRINT#1, EOT\$:REM 关闭总线上的外设 (GO TO SLEEP)

510 ADR\$="00":REM 访问口地址"00"号

530 C\$=EOT\$+ADR\$+ENQ\$

540 PRINT #1, C\$

550 FOR T=0 to 500:NEXT: REM 延时

560 A\$=INPUT #1

570 IF A\$=ADR\$+ACK\$

580 PRINT "连接成功!"

关闭通讯连接:

当通讯完成后,上位机发送关闭命令.通讯数据总线上的已建立通讯的调节器响应后,自动将调节器的发送线高(SDA)和低(SDB)由低阻态转成高阻态,完成解除连接.其后上位机才能访问通讯数据总线上其它口地址的调节仪表,否则不能进行正常通讯.
RS232接口,只能单台点对点间进行通讯。

上位机 (主)	调节器 (从)
EOT	无响应

BASICA程序例

```
PRINT #1, CHR$(4)
```

上位机读命令的组成

读命令是对调节器的控制参数,设置工作内容的读入.
读命令的全文件是由读文件,引导符,结束符,BCC二进制块校验符的字符串组成.
正常返回的是由引导符,数据文件,结束符和BCC二进制块校验的字符串.
不正常的答复包括对接收字符串格式错误,校验错误回送的错误号码.

上位机	调节器
STX+'文件'+ETX+BCC (数据请求)	STX+'文件'+ETX+BCC(正常答复)
NAK(请求重发)	(最多三次)
ACK (正常)	不响应
	'错误号'+NAK (不正常答复)

通讯的读命令示意图

BASICA程序例

```
PRINT #1, CHR$(2)+"读文件"+CHR$(3)+块校验符
```

上位机写命令的组成

写命令是对调节器的控制参数,工作参数内容的写入.
写命令的全文件是由写文件,引导符,结束符,BCC二进制块校验符的字符串组成.
正常返回的是响应码(ACK)的字符.表示写命令成功.
不正常的响应中包括对发送命令格式,校验错误号的回送.

上位机	调节器
STX+'写文件'+ETX+BCC	
	ACK (正常)
	'错误号'+NAK (不正常)

通讯的写命令的示意图

BASICA程序例

```
PRINT #1, CHR$(2)+"写文件"+CHR$(3)+块校验符
```

BCC 块校验

除字节的奇偶校验外, 传送的字符串还采用二进制BCC块校验方式。校验码的产生是将引导符STX后到BCC效验符前的全部数据(含EXT)二进制数求和后得到的ASC码。

例如: STX + 'M1' + ETX + BCC
 (02H) (4DH)(31H) (03H) (81H)
 二进制求和 4DH+31H+03H=81H (8bits)
 转成ASC 码 81H&7FH=01H (7bits)

注: "M1"是FP21的读命令

BASICA BCC块效验程序例, 其中CMD\$为读/写文件

```
560 CMD$=CMD$+ETX$: LEC=LEN(CMD$): BCC=0
570 FOR I=1 TO LEC: S$=MID$(CMD$, I, 1)
580 BCC=BCC+ASC(S$)
590 NEXT
600 BCC=BCC MOD 128: REM 屏蔽字节第8位
610 BCC$=CHR$(BCC): REM BCC的ASC码
620 TXD$=STX$+CMD$+BCC$: REM 生成发送读写字符串
630 RETURN
```

通讯中的错误, 将自动返回错误识别码

FP21的错误识别码

- ER0: 错误的操作方式
- ER1: 错误的文本格式
- ER2: 错误的命令
- ER3: 错误的的数据(如: 量程溢出)
- ER4: 奇偶校验错
- ER5: 非法的写命令
- ER6: 不合法的工作方式改变

SR25的错误识别码

- ER1: 错误的文本格式
- ER2: 错误的命令
- ER3: 错误的的数据(如: 量程溢出)
- ER4: 奇偶校验错

通讯的定时

上位机发送命令后, 3秒内无回答, 可视为通讯超时错误。

联接命令成功后, 5分钟内不读写, 仪表将自动关闭通讯连接, 需重新建立联接。

§4 CC2125A V2.10 的工作环境

- IBM PC/XT, AT, 286, 386或兼容机
- EGA(640×350)或EGA/VGA彩色显示器
- 640KB 以上的存储器
- 1.2MB 5.25" 软盘驱动器
- 至少5MB硬盘空间
- 一个RS232串口
- MS-DOS 3.0以上版本的磁盘操作系统
- UCDOS V2.0 汉字操作系统

§ 5. CC2125 V2.10的安装

A. UCSDOS 2.01 的安装 (已配置UCSDOS的用户跳过此项)

将UCSDOS的#2盘装入A驱动器,键入INSTALL后回车。按屏幕上的提示依次将#3、#4盘装入。

B. UCSDOS 2.01系统的配置

键入命令: C:/S 后,进入UCSDOS系统配置的设置。

建议如下: 字库设置: 64K 用于放入基本内存
显示设置: EGA 方式
打印模式: 西文方式
系统减: 键盘 ON; 联想 ON; 打印 ON

注: 更详细的说明见 UCSDOS 操作说明书。

C. CC2125A的装入

键入 MD回车后,建立CC2125的子目录。

键入CD后回车,进入子目录。

将#1盘插入A驱动器后,键入 COPY A:*. * 后回车。至此系统安装完成。

§6. CC2125A V2.10软件的使用方法

(一) 在UCSDOS操作系统下,键入CD后,在C盘提示符下键入:

C>CC2125A 即可启动CC2125学习软件,进入主菜单。

(二) CC2125的操作简单,均可按屏幕提示进行。

在主菜单时的操作:

按左、右方向键(),可水平移动主菜单,当光标指向某一功能项时,按回车键即可进入主菜单内的子菜单。

在子菜单时的操作:

按上、下方向键(),可上、下移动菜单,当光标指向某一子菜单命令项时,按回车键即可选择此命令项功能。

选择子菜单内命令项时的操作:

CC2125A软件已被设计成自动向用户提示命令参数的选择。按左、右方向键(),可水平移动子菜单命令参数窗口,当光标指向某一参数时,按翻页上键、下键(PgUp, PgDn)或删除键(Del)修改参数。按回车键即可执行该命令。按Esc键后,将返回子菜单。

其它的画面显示

CC2125A软件已被设计成自动向用户显示发接的全文件通讯格式、接收文件的分项参数定义、通讯的出错信息等较为完善的学习功能。

(三) 主菜单和子菜单

进入主菜单后,将显示七个主要功能:

1. 串口接线

PC机与带RS-232C通讯口调接器的连线

PC机与先锋RS-232C/RS-422A通讯变换器连线

先锋RS-232C/RS-422A通讯变换器与带RS-422A通讯口调节器的连线

PC机25针、九针串口接线对照表

2. 通讯协议

建立数据链接

取消数据链接

读数据

写数据

BCC块校验码

控制码

出错信息

时间的约定

3. 参数设置

CC2125A对调节器通讯地址和PC机串口设置和选择,PC机的通讯参数设置。

4. 仪表的通讯测试

检测仪表的通讯的链接状态。

5. FP21通讯学习窗口(内容见后 § 7 FP21通讯命令的设计特点)
6. SR25通讯学习窗口(内容见后 § 8 SR25通讯命令的设计特点)
7. 工具
 - PC机串口检查。
 - 先锋RS-232C/RS-422A通讯转换器检查。
 - 发送连续脉冲串,对通讯数据波形测试。

四 进入通讯命令学习前的准备工作

1. 初次连接系统的准备工作 (仪表未连接)

参照串口接线窗口和附录 A . 通讯串口接线方法,对系统进行正确地接线。

检查通讯系统: 参照屏幕的工具窗口,以短路线方法,通过测试软件检查。

PC机通讯口正常(包括地线、握手信号)。

232接口连线正确。

先锋RS232/RS422接口正确。

先锋RS232/RS422接口到仪表的连线正确。

如果远距离通讯(1200米),利用示波测量发送波形的前沿,确定通讯线路的传输品质,选择合适的通讯波特率或先锋RS232C/RS422A通讯口的分配。

2. 通常的操作

连接仪表并上电,确信仪表已进行了 § 2节的通讯参数的设定。

根据前述FP21和SR25通讯参数的设置,在学习软件中相应的画面应设置与仪表相一致的PC机通讯地址和字符参数,否则将不能正常通讯。

执行仪表的通讯测试窗口功能,画面将自动指示仪表的通讯状态。检测通过后,程序才能进入下一步的通讯学习窗口。否则用户需先排除软硬件故障。

§ 7. FP21通讯命令的设计特点

(一) 读/写命令的分类:

参照先锋编写的FP21中文操作说明书和操作流程。FP21的通讯命令可按流程图分为后述的9组画面群。共计36组读写命令,基本包括了全部的操作流程图窗口。

在流程图参数窗口的右下脚,标出了有关的通讯读写命令。命令中标有"R"脚注的,该命令仅能作为读命令。通讯命令是由9个大写ASC 字符和0~9数字组成,文件可分为单独命令,或者带有参数的命令。加上引导符、结束符、校验符后,组成了发送接收的全文件。

数据格式说明:

- (a) 字符... 大写ASC 字符
- (b) "÷", " - ", ". "号, " - "号也用于读命令和数据间的分隔。
- (b) " " 空格,用于命令和数据间的分隔(读命令发送除外)
- (c) ", "用于数据间的分隔。
- (d) "; " 用于写命令的数据省略。

数字表示: 括号内为数字例

N: (1)、NN: (02)、NNN: (012)、NNNN(1234):、NN. N: (12. 3)、NNN. N: (700. 0)

带符号的数字表示:

SNNN(+123)、SNN. N(-12. 3)、SNNN. N(+005. 0)

SXXXX表示不确定的数值,将视用户量程及传感器选择的不同而定。

PTN No.: 曲线号,如PTN 1~10、分别用01~10,表示

STP No.: 曲线的步号,分别用01~10(9组曲线时),01~81步(一组曲线81步方式时)表示

PID No.: PID,如PID 1~10、分别用1~10,表示

ALARM No.: 报警号,如ALARM 1~10、分别用1~10,表示

命令由两位字符组成,第一位是大写的ASC 字符,第二为是数字表示,如 "D1", "M1"等

在的说明中，符号【 】仅作为引用命令表示。符号内的 等数字号表示分项参数和在命令中的位置。有分项参数内容的，参数间必须使用“,”逗号作为分隔符。命令必需严格按书写格式要求，不得随意增减符号，空格，正负号，改变参数长度和小数点位置。

为方便用户对照参考，在说明中，引用【 】内的数字，来标明命令在 F P 2 1 中文流程图的窗口位置。

读命令(R):

例如最常用的读命令"D1"键入后，画面将显示返回接收的"STX D1, , , , +ETX+BCC"全文件字符串和显示字符串中文件的参数内容：测量值PV= , 设定值= , 曲线号= , 步号= 。读命令后如果带的参数，读命令和参数之间，需加“-”作为分隔符。例如【S1-1,01】第一组曲线，第一步和运行时间参数的读文件书写格式。读命令无省略格式，参数间无空格。

写命令(W):

写命令字符后需加一个空格作为分隔符，以区别读命令，后续说明中用字符" "表示键入一个空格。例如键入【E1 RUN】写命令后，程序将执行。由于不同的量程，时标等初始化参数设置不同等原因，准确记忆和书写命令的文件中参数格式是困难的。 规律的是，FP21 的读命令的返回参数，正是写命令的标准格式。这里，CC2125已经被设计成，对于又读又写命令，先发送该命令的读命令。用户仅需参照画面返回的参照格式，修改文件中的参数回填即可。例如，一号PID参数的读命令为【C1-1】，参照接收显示的"C1 1, , , "格式，修改" , , , "的 P I D 参数项。

部分参数项内容的修改写命令可采用命令的省略格式。符号";"分号和",,"双逗号可作为分项参数的省略符，如写文件【C1 1, ;】或【C1 1, , ;】或【C1 1, , , 】省略格式都是正确的，并仅对局部PID参数进行修改。在CC2125的编辑窗口中，通过

, 数字, 字符和"Del"删除键, 试运行下述的写曲线步的控制参数修改命令格式。

FP21写曲线步的控制参数	A. 全参数	B. 仅改报警号2	C. 仅改PID号2
【S2 1, 01, , 1 1】		【S2 1,01,,2】	【S2 ,1,01,2;】
曲线号_PTN_No.,	【1】	【1】	【1】
步号_STP_No.,	【01】	【01】	【01】
PID号_PID_NO.,	【1】	【 】	【2】
报警号_ALM_NO.	【1】	【2】	【 】

写命令成功后，可通过再次的读命令或FP21相对应的窗口，检查修改的内容。

说明:

如用户发送错误的命令或错误的格式或通讯错误等，经FP21接受并将回送六种出错 讯息，用户可根据画面的出错号，对错误进行分析。

上述文件书写过程中，软件将自动生成文件完整的发送字符串，自动指示出全文件的 格式。

FP21处于机内和外部开关工作方式时，仅"D1~D4"读命令有效。只有处于在机内(LOC)工作方时，在[OPM]操作窗口，键选"COM"后，FP21才能进入通讯工作方式，此时，面板上的"COM"通讯工作方式灯亮。处于COM通讯，可执行全部的通讯命令。通讯方式仅能由手动键解除。处于通讯方式时，只有在上位机发送【01 E】命令后，才能进入在COM-EXT(通讯-外部)复合工作方式。此时，面板的"COM"和"EXT"工作方式灯亮。处于复合工作方式时，可执行全部的读命令和唯一的【01 C】解除复合返回通讯工作方式的写命令。运行程序中，上位机发送【E1 HLD】的程序保持命令后，曲线，PID等参数可被上位机写命令修改。

规律的是，FP21的读命令的返回参数，正是写命令的标准格式

(二) 读/写命令群的细则:

F P 2 1 读 写 命 令 群

O1: 工作方式窗口	【01】	R/W	命令
DN: 面板LED显示	【D1~D4】	R	命令
MN: 监示窗口群	【M1~M3】	R	命令
EN: 执行参数窗口	【E1~E5】	R/W	命令

P1: 曲线的编程参数	[[P1]]	R/W	命令
SN: 曲线的步参数	[[S1~S6]]	R/W	命令
CN: 控制参数窗口	[[C1~C3]]	R/W	命令
KN: 键锁定窗口	[[K1~K3]]	R/W	命令
IN: 读初始化窗口群	[[I1~I9]]	R	命令

【01】 : 工作方式窗口 R/W 命令
 R : 【01】 读工作方式 【OPM】窗口
 通讯工作方式COM/通讯-外部开关复合工作方式EXT"
 W : 【01 COM/EXT】 写工作方式 (01 EXT) (01 COM)

【DN】 : 面板LED显示 R 命令
 R : 【D1】 读当前测量值/设定值
 测量值 PV, , 设定值 SV , 曲线号 PTN No. , 步号 STEP No.
 R : 【D2】 读状态指示灯 (ON/OFF)
 复位 RST, 确保平台 GUA, 步进 ADV, 保持 HLD, 运行 RUN, 定值 FIX, 手动 MAN,
 自整定 AT, 编程检查 CFM
 R : 【D3】 读时间开关状态 (ON/OFF)
 时间开关TS1(ON/OFF), 时间开关TS2(ON/OFF), 时间开关TS3(ON/OFF), 时间开关
 TS4(ON/OFF)
 R : 【D4】 读报警状态 (ON/OFF)
 报警AL1(ON/OFF), 报警AL2(ON/OFF), 超量程S0(ON/OFF)

MN: 监示窗口群 [[M1~M3]] R 命令 (. M1特例的写命令)

R : 【M1】 读调节输出 【11】【12】【13】窗口
 调节输出值_OUT, 偏差值_DEV, 当前程序步运行的剩余时间_TIME
 W : 【M1 】 写手动调节输出值 (. M1特例的写命令)
 E1 MAN写命令后, FP21手动方式的手操调节输出值的设置。

R : 【M2】 读曲线链接 【14】【15】窗口
 链接格式 LINK FORMAT, 链接指针(当前的运行曲线), 链接执行重复次数(实际值), 链
 接设定的总重复次数, 当前曲线重复次数(实际值), 曲线设定的重复次数

R : 【M3】 读当前运行参数 【16】【17】窗口
 当前运行的PID号, 当前运行的报警号, 当前目标设定值, 当前步的设定时间

E N: 执行参数窗口 [[E1~E5]] R/W 命令

R : 【E1】 读键功能作用执行状态 【21】窗口 (ON/OFF)
 复位 RST, 确保平台 GUA, 步进 ADV, 保持 HLD, 运行 RUN, 定值 FIX,
 手动 MAN, 自整定 AT, 编程检查 CFM

W : 【E1 RST/ GUA/ ADV/ HLD/ RUN/ FIX/ MAN/ AT/ CFM】 写键功能执行
 命令例: E1 RUN E1 HLD E1 AT

R : 【E2】 读程序执行曲线入口和入口步 【22】窗口
 起始曲线号_START_PTN_No., 起始步号_START_STEP_No.

W : 【E2 , 】 写程序执行曲线入口和入口步

R : 【E3】 读曲线链接形式 【23】【24】窗口

 曲线链接格式_LINK_FORMAT, 曲线链接设定的重复执行次数_LINK_EXEC, 伺服启动
 _PV_START(YES/NO)

W : [[E3 , , Y/N] 写曲线链接形式

R : [[E4] 读跳步方式参数 【2 5】窗口
跳步方式_ADV_MODE(STP/TIME), 跳步时间_ADV_TIME

W : [[E4 ,] 写跳步方式参数

R : [[E5] 读定值控制参数 【2 6】窗口
定值方式设定值_FIX_SV, 定值方式PID号_FIX_PID_No, 定值方式报警号_FIX_ALM_No.

W : [[E5 , ,] 写定值控制参数

P1: 曲线的编程参数 [[P1] R/W 命令

R : [[P1- 曲线号_PTN_No.] 读曲线的初始参数 【3 2】【3 3】【3 4】窗口
曲线号_PTN_No., 曲线的起始设定值_START_SV, 确保的允许偏差_GUA_ZONE,
确保的允许等待时间_GUA_TIME, 曲线结束步_PTN_END, 曲线重复次数_PTN_RPT

W : [[P1 , , , ,] 写曲线的初始参数

SN: 曲线的步参数 [[S1~S6] R/W 命令

R : [[S1- 曲线号_PTN_No., 步号_STP_No.] 读曲线步的目标参数 【3 6】窗口
: 曲线号_PTN_No., 步号_STP_No., 步的目标设定值_SV, 步的目标执行时间_TIME

W : [[S1 , , ,] 写曲线步的目标参数

R : [[S2- 曲线号_PTN_No., 步号_STP_No.] 读曲线步的控制参数 【3 7】窗口
: 曲线号_PTN_No., 步号_STP_No., PID号_PID_NO., 报警号_ALM_NO.

W : [[S2 , , ,] 写曲线步的控制参数

R : [[S3- 曲线号_PTN_No., 步号_STP_No.] 读曲线步的时间开关TS1 【3 8】窗口
: 曲线号_PTN_No., 步号_STP_No., 时间开关1允许_TS1(YES/NO),
开关1开时间_TS1_ON_TIME, 开关1关时间_TS1_OFF_TIME

W : [[S3 , , , ,] 写曲线步的时间开关TS1

R : [[S4- 曲线号_PTN_No., 步号_STP_No.] 读曲线步的时间开关TS2 【3 9】窗口
曲线号_PTN_No., 步号_STP_No., 时间开关2允许_TS2(YES/NO),
开关2开时间_TS2_ON_TIME, 开关2关时间_TS2_OFF_TIME

W : [[S4 , , , ,] 写曲线步的时间开关TS2

R : [[S5- 曲线号_PTN_No., 步号_STP_No.] 读曲线步的时间开关TS3 【3 1 0】窗口
: 曲线号_PTN_No., 步号_STP_No., 时间开关3允许_TS3(YES/NO),
开关3开时间_TS3_ON_TIME, 开关3关时间_TS3_OFF_TIME

W : [[S5 , , , ,] 写曲线步的时间开关TS3

R : [[S6- 曲线号_PTN_No., 步号_STP_No.] 读曲线步的时间开关TS4 【3 1 1】窗口
: 曲线号_PTN_No., 步号_STP_No., 时间开关4允许_TS4(YES/NO),
开关4开时间_TS4_ON_TIME, 开关4关时间_TS4_OFF_TIME

W : [[S6 , , , ,] 写曲线步的时间开关TS4

CN: FP21控制参数窗口 [[C1~C3] R/W 命令

R : [[C1- PID控制号] 读PID参数 【4 2】窗口
: PID控制号, 比例带_P, 积分时间_I, 微分时间_D

W : 【C1 , , , 】写PID参数

R : 【C2- PID控制号】读调节输出限幅参数 【4 3】窗口

: PID控制号, 调节输出上限%_OH, 调节输出下限%_OL

W : 【C2 PID控制号, 调节输出上限%_OH, 调节输出下限%_OL】写调节输出限幅参数

R : 【C3- 报警号】读报警参数 【4 4】【4 5】窗口

: 报警号, 报警继电器1报警设定值_AL1, 报警继电器2报警设定值_AL2

W : 【C3 , , , 】写报警参数

KN: 键锁定窗口 【K1~K3】 R 命令

R : 【K1】读设定值设定限定范围 【5 4】窗口

: 设定值上限_SVHL, 设定值下限_SVLL

W : 【K1】写设定值限定范围

R : 【K2】读9组曲线或一组81步曲线编程方式 【5 5】窗口

: 曲线9组(NO)或单81步(YES)选择, 程序的81步循环允许(NO/YES)

W : 【K2】写9组曲线或一组81步曲线编程方式

R : 【K3】读曲线检查(CFM_MODE)方式 【5 6】窗口

: 仅入口编程曲线(PTN)或/全部曲线(LINK)检查, 编程检查时间倍率_CFM_TIME

W : 【K3】写曲线检查方式

IN: FP21 初始化窗口群(只读) 【I1~I9】 R 命令

R : 【I1】读正反作用, 数字滤波参数 【6 1】【6 2】窗口

: 测量值滤波常数 PV_FILTER, 测量值偏移 PV_BIAS,
调节反(R)/正(D)作用, SSR和继电器接点的比例周期 CYC_TIME

R : 【I2】读模拟通讯参数 【6 3】【6 4】【6 5】选件窗口

: 模拟通讯1发送类型(SV/PV) TMT1_MODE, 模拟通讯2发送类型(PV/SV) TMT2_MODE,
模拟通讯1发送上限 TMT1_HL, 模拟通讯1发送下限 TMT1_LL,
模拟通讯2发送上限 TMT2_HL, 模拟通讯2发送下限 TMT2_LL

R : 【I3】读报警动作方式 【6 B 1】【6 B 2】【6 B 3】窗口

: 报警1 动作方式 AL1_MODE, 报警2 动作方式 AL2_MODE,
报警1 灵敏度 AL1_SENS, 报警1 报警待机STBY (NO/YES),
报警2 灵敏度 AL2_SENS, 报警2 报警待机STBY (NO/YES)

报警动作方式:

A. 绝对值报警	LL1	HL1	LL2	HL2
	下限绝对值上回差	上限绝对值下回差	下限绝对值下回差	上限绝对值上回差
B. 绝对偏差报警	AD1	AD2		上下限偏差
值外	上下限偏差值内	C. 偏差值报警	LD1	HD1
LD2	HD2			
	下限偏差值外	上限偏差值外	下限偏差值内	上限偏差值内报警

待机: 调节输出0%(反作用) 100%(正作用)

R : 【I4】读外部I/O接口定义 【6 B 4】【6 B 5】【6 B 6】窗口

: 外部I/O DI_1 定义, 外部I/O DI_15 定义
外部 I/O OC门D021, 外部 I/O OC门D022, 外部 I/O OC门D023,
外部 I/O 继电器D031, 外部 I/O 继电器D032, 外部 I/O 继电器D033

定义内容: 时间开关信号TS1, TS2, TS3, TS4和S0, RUN, END, EXT状态

- R : 【15】读系统输出配置说明 【6 C 1】窗口
: 调节输出类型(MA、V、CNT(继电器)、SSR),
模拟通讯1配置型式(NON:无、MA、V、MV), 模拟通讯2配置型式(NON:无、MA、V、MV),
数字通讯接口类型(232C、422A)
- R : 【16】读计量单位 【6 C 2】窗口
: 计量(C= , F=华氏, N=无, %)单位, 铂电阻标准(PT=Pt100(1385) JPT=JPt100(BA2 1391))
- R : 【17】读输入类型 【6 C 3】窗口
: 输入类型(MV, V, MA, 热电偶TC, 铂电阻PT), 传感器类型, 0% 量程, 100% 量程
- R : 【18】读直流数字量程 【6 C 4】【6 C 5】窗口
: 直流数字量程下限 SCALE L, 直流数字量程上限 SCALE H,
小数点位置(0: xxx.x 1: xxx.x 2: xx.xx 3: x.xxx)
热电偶, 铂电阻输入时 内容为"--"
- R : 【19】读上电方式、PID算法、时间单位 【6 C 6】【6 C 7】【6 C 8】窗口
: 超量程对策 SO MODE(RST/HLD), 超量程保持输出值 SO OUT,
五种上电方式 NML: 普通 RST: 全部复位 AUTO: 上电自动运行 GUA: 保护方式 AUG: 自动运行和保护
保护的复合方式
程序运行时间单位(MIN: 分 SEC: 秒), 三种 PID 算法选择(SER: 串行 PARA: 并行 DPA: 微分超前)

§ 8. SR25通讯命令的设计特点

(一) 读/写命令的分类:

参照先锋编写的SR25中文操作说明书和中文操作流程图, SR25的通讯命令可按流程图分为三层、26组读写命令, 包括了全部的流程图窗口。通讯命令是由9个大写ASC 字符和0~9数字组成, 命令可分为单独命令, 或带参数的命令。加上引导符、结束符、校验符后, 组成了发送接收的全文件。

命令的书写格式说明:

- 字符... 大写ASC 字符
- "÷", " - ", " ."号。
- " " 空格, 用于写命令和数据间的分隔(读命令发送除外)
- ","用于数据间的分隔。
- ";" 用于写命令的数据项省略。

数字表示: 括号内为数字例

N: (1)、NN: (02)、NNN: (012)、NNNN(1234):、NN.N: (12.3)、NNN.N: (700.0)

带符号的数字表示:

SNNN(+123)、SNN.N(-12.3)、SNNN.N(+005.0)

SXXXX表示不确定的数值, 将视用户量程及传感器选择的不同而定。

SV No.: 设定值号, 如SV1~10、RSV, 分别用01~10, 00表示

命令字由两位大写的ASC 字符组成, 如 "DS", "MD"等

在下述的说明中, 符号【 】仅作为引用命令的表示。符号内的 等数字号表示分项参数及其在命令中的位置。有分项参数内容的, 参数间必须使用", "逗号作为分隔符。命令必需严格按书写格式要求, 不得随意增减符号, 空格, 正负号, 改变参数长度和小数点位置。

为方便用户对照参考, 在下述的说明中, 引用【 】来标明命令在SR25流程图的窗口位置。

读命令(R):

例如常用的"DS"读命令, 返回的参数是"STX DS, , , , , +ETX+BCC". 画面将指示, 测量值PV= , 设定SV的选择号= , 设定值SV= , 自动/手动= , 调节输出1输出百分比= 。如果读命令后带有参数项, 在读命令字符和参数间不加分隔符。例如

[[S V 01]] 的第一组设定值SV, [[C P 01]] 的第一组PID参数读命令。读命令无省略格式, 参数间无空格。
写命令(W):

写命令字符后需加一个空格作为分隔符, 以区别读命令, 后续说明中用字符" "表示键入一个空格。
例如自整定的[[A T E]]。又例如PV选择号写命令[[S N 02, 0]] 为不带斜率, 立即调用第二组设定值的运行命令。由于不同的传感器, 量程范围, SR25的四种工作方式的选择等原因, 造成书写不同格式文件的困难。CC2125已被设计成, 对于大多数的写命令, 先发送该命令的读命令。用户仅需参照返回的文件格式, 修改或删除其它项参数回填即可。如[[S V 01]] 读命令返回的文件是"SV 01, ", 可参照 内的参数格式, 修改第一号设定值后, 重发该参数的写命令。

部分参数项内容的修改写命令可采用省略格式。符号";"分号和",,"双逗号可作为分项参数的省略符, 如写文件[[O L 02, ;]] 或[[O L 02,, ;]] 或[[O L 02,,,,]] 省略格式都是正确的, 仅对局部功率限幅参数进行了修改。在CC2125的编辑窗口中, 通过 , 数字, 字符, "Del"键, 试运行下述的工作方式修改命令格式:

SR25工作方式写的全文件	A. 全参数	B. 仅改工作方式2	C. 仅改正作用和禁止自动返回 [[M D
0, R, U, I, Y, 60]]	[[M D 2;]]	[[M D , D, , , N;]]	
四种工作方式0、1、2、3	[[0]]	[[2]]	[[]]
正反作用(D/R)	[[R]]	[[]]	[[D]]
串级转移有效(T/U)	[[U]]	[[]]	[[]]
热电偶冷端补偿内/外(I/E)	[[I]]	[[]]	[[]]
显示自动返回[00]窗口(Y/N)	[[Y]]	[[]]	[[N]]
自动返回时的显示停留时间	[[60]]	[[]]	[[]]

写命令成功后, 可发送读命令或在SR25中相对应的参数窗口, 检查被修改的参数内容。
说明:

如用户发送错误的命令或错误的格式或通讯错误等, 经SR25接受并将回送四种出错讯息, 用户可根据画面显示的出错类型号, 对错误进行分析, 以便重新修改。

上述文件书写完成确认后, 软件将自动生成该文件完整的发送字符串, 自动指示出该文件的格式。

SR25 处于机内工作方式, 仅能执行全部的读命令。处于通讯工作方式时, 可执行全部的读和写命令。只有在上位机发送"CM C"命令后, SR25才能进入通讯工作方式, 此时, SR25面板的"COM"通讯工作方式灯亮。通讯方式既可由"CM L"通讯命令解除, 也可由手动键解除。

(二) 读/写命令群的细则:

SR25 读写命令群

第一层命令 LEVEL 1

: D S 监视窗口 R
: A M 自动/手动 W
: S N SV号的选择 W
: S V SV值的设定 R/W
: C P PID 参数 R/W
: E D 事件设定 R/W
: R P 斜率参数 R/W
: O L 输出限幅 R/W

第二层命令 LEVEL 2

: C D 控制状态 R
: A T 自整定AT W
: S S 内/外SV选择 W
: C M 通讯方式 W
: R M 斜率控制 W
: S B 脱机(ST-BY) W
: R O 调节参数 R/W
: I N 输入参数 R/W
: D I DI 开关设定 R/W
: S C 输入定标 R/W
: R D 斜率时间 R/W
: M D 工作方式 R/W
: T X 模拟发送 R/W
: C C 通讯参数 R

第三层命令 LEVEL 3

: K L 键锁定 R
: R G 传感器类型和量程 R
: S Y 系统配置 R
: E O 事件输出状态 R

第一层命令LEVEL 1

:监视窗口 D S R 命令 【L 1 - A】窗口群

R : 【D S】

当前测量值PV 当前SV的选择号 设定值SV 自动/手动 调节输出1输出百分比
调节输出2输出百分比 (双输出时)

注: 内的参数有4种小数点位置: 0 = SONNNN 1 = SNNN.N 2 = SNN.NN 3 = SN.NNN

超量程时参数为: PV > 110% : H.H.-- PV < -10% : L.L.--

铂电阻输入断线时: b.---, H.H.--, c.---

热电偶输入断线时: H.H.--

:自动/手动 AM W 命令 【L 1 - A 1】窗口

W: 【AM A】

无扰动地转自

动

:

【AM M;】

无扰动地转手动

【AM M, 输出 1 (OUT1) SNNN.N】 (AM M, +010.0)

手动输出 1 设置

【AM M, 输出 1 (OUT1) SNNN.N, 输出 2 (OUT2)SNNN.N】

双输出的手动输出 1, 2 设置

【AM M, 输出 1 (OUT1) SNNN.N;】 (AM M, +050.0;)

双输出的手动输出 1 设置

【AM M,, 输出 2 (OUT2) SNNN.N】 (AM M,, +050.0)

双输出的手动输出 2 设置

:设定值SV号的选择 S N W 命令 【L 1 - A 1】窗口显示时,面板“S V”键操作

W: 【S N NN Q】

(SN 02, Q)

无斜率的SV号的立即运行

【S N NN】

(SN 02)

如果斜率被设定的话,带斜率的SV目标号的运行

: S V SV值的设定

R/W 命令 【L 1 - B】窗口群

R: 【S V】

SV No. 当前显示的运行SV值 SV No. 的SV值 读当前运行SV值

(当斜率运行时 的值不相等, 如方式工作方式0或无斜率运行时 的值相等)

R: 【S V SV No.】

SV No. SV号的设定值 (SV02)

读SV号的设定值

W: 【S V SV No, SXXXX】

(SV 03, SXXXX)

写SV号的设定值

W: 【S V SXXXX】

(SV SXXXX)

写当前运行SV号的设定值

: C P PID 参数

R/W 命令 【L 1 - C】窗口群

R: 【C P】

读当前运行SV号的PID值

【C P NN】 (CP00)(CP02) 读SV号的PID值 NN=00为串级PID号

设定值号_SVNo. 比例带_P 积分时间_I/R 微分时间_D/H1 (单输出时)

比例系数_K2 回差_H2 死带_DB (双输出时)

W: 【C P SV No., , , ,】 (CP 02, NNNN;)

【C P SV No, , , , , , ,】 写SV号的PID参数

省略法例: 【C P , ;】 【C P , , , , , ,】

: E D 事件设定

R/W 命令 【L 1 - D】窗口群

R: 【E DN】 (ED1) (ED5) 读事件参数

事件号 类型方式 报警方式 报警值 报警的回差 报警的待机 动作延迟时间

机背的输出端子号 1: EV1 2: EV2 3: EV3 4: D01 5: D02

动作的类型方式 0: DEV 1: PV 2: SV 3: AT 4: RUN 5: ERR 6: RSV 7: MAN

报警方式 0: H1 1: H2 2: L1 3: L2 4: DH 5: DL 6: ADH 7: ADL

报警值

报警的回差

(DI 0,0,0,0,0) (DI 1;) (DI ,3;) (DI ,,,,6)

: S C 输入定标 R/W 命令 【L 2 - E】窗口群

R : 【S C】 读输入定标

直流的小数点位置 设定值/测量值下限 设定值/测量值上限 串级设定值下限 串级设定值上限

0: XXXX 1: XXX.X 2: XX.XX 3: X.XXX

W : 【S C , , , ,] 输入定标

: R D 斜率时间参数 R/W 命令 【L 2 - F】窗口群

R : 【R D】 读斜率时间参数

斜率时间单位秒S/分M 斜率系数0:***.*(×0.1) 1:****(×1)

W : 【R D ,] 写斜率时间参数

省略法例: 【R D S;】 【R D M;】 【R D ,1】

: M D 工作方式 R/W 命令 【L 2 - G】窗口群

R : 【M D】 读工作方式

四种工作方式0、1、2、3 正反作用(D/R) 串级转移有效(T/U)

热电偶冷端补偿内/外(I/E) 显示自动返回[00]窗口(Y/N) 其它窗口中自动返回时的显示停留时间

W : 【M D , , , , ,] :写工作方式

省略法例: 【M D 2;】 【M D ,R,,,] 仅适用热电偶输入类型

: T X 模拟发送参数 R/W 命令 【L 2 - H】 【L 2 - H 1】 【L 2 - H 2】窗口

R : 【T X】 读模拟发送参数

模拟通讯1发送类型 模拟通讯2发送类型 , 模拟通讯1下限 模拟通讯1上限 模拟通讯2下限 模拟通讯2上限

W : 【T X , , , , , ,] 写模拟发送参数

: C C 通讯参数读 R : 【C C】 【L 2 - H 3】 【L 2 - H 4】窗口(该参数仅能由机内设定)

通讯地址(00~31) 通讯波特率(0:1200 1:2400 2:4800 3:9600 通讯数据格式(0:7个数据位,偶效验(E) 1:8个数据位,无效验)

第三层命令 LEVEL 3

: K L 读键锁定ASC 码 R: 【K L】 【L 3 - A】窗口群

键锁定1代码NN 键锁定2代码NN

* 16个键功能锁定(ON锁定, OFF解锁) *

BCD码	键锁定1内容	BCD码	键锁定2内容
1	MON [L1-A]手动禁止	1	INP 输入关系组
2	SVn SV设定组	2	DIA 四个外部DI开关
4	PID PID设置组	4	SCL 上/下限范围限制
8	E/D 事件设置组	8	RDT 斜率参数
1	RMP 斜率设置组	1	MOD 工作方式选择
2	OLT 调节输出限幅组	2	OPT 模拟发送通讯参数
4	CTL 控制作用组	4	INI 初始化
8	OUT 调节输出组	8	RNG 输入类型选择组

NN=16进制的ASC 码表示

例如: 代码=45(01000101) 表示CTL、PID、MON键被锁定

代码=C0(11000000) 表示RNG、INI 键被锁定

: R G 传感器类型和量程 R: [[R G] 【 L 2 - C 】窗口群
 计量单位 铂电阻标准 量程号
 : 单位 0: 摄氏 1: 华氏 2: 百分比 3: 无
 : 铂电阻类型 1: Pt100 0: JPt100(BA2) (TC和直流输入时无此项)
 : 测量范围NN码的定义

热电偶输入(, 华氏略)	线性输入	铂电阻输入(, 华氏略)
00: B 0~1800	22: -10 ~ 10 mV / -1~1 V	31: -199.9~600.0
01: R 0~1700	23: 0 ~ 10 mV / 0 ~ 1 V	32: -100.0~100.0
02: S 0~1700	24: 0 ~ 20 mV / 0 ~ 2 V	33: -100.0~300.0
03: K -100.0~400.0	25: 0 ~ 50 mV / 0 ~ 5 V	34: - 40.0~ 60.0
04: K 0.0~800.0	或 0~20 mA	35: 0.00~50.00
05: K 0~1200	26: 10 ~ 50 mV / 1 ~ 5 V	36: 0.0~100.0
06: E 0.0~700.0	或 4~20 mA	37: 0.0~200.0
07: J 0.0~600.0	27: 0 ~ 100 mV / 0 ~ 10 V	38: 0.0~500.0
08: T -199.9~200.0		
09: N 0~1300		
10: PL 0~1300		
11: PR40-20 0~1800		
12: WRe5-26 0~2300		
13: U -199.9~-200.0		
14: L 0.0~600.0		

: S Y 系统配置读 R: [[S Y]

调节输出1类型 调节输出2类型 模拟通讯1类型 模拟通讯2类型 数字通讯类型 串级隔离形式
 串级输入类型

, 0: 无 1: 继电器 2: SSR 3: 4~20 mA 4: 其它 mA 范围 5: 0~10V 6: 其它 V 范围
 , 0: 无 1: 0~10 mV 2: 其它 mV 范围 3: 4~20 mA 4: 其它 mA 范围 5: 0~10 V 6: 其它 V 范围
 0: 无 1: RS232C 2: RS422A
 N: 非隔离 I: 隔离
 0: 0~10 V 1: 1~5 V 2: 其它 V 范围 3: 4~20 mA 4: 其它 mA 范围

: E O 事件输出状态读 R: [[E O] 状态= ON/OFF (开/关)前面板状态灯
 事件1 事件2 事件3 外部D01 外部D02

§ 9. BASICA的程序通讯软件说明:

在用户的DOS3.0以上的操作系统上,插入#1号软盘起动BASICA后,可列表打印或
 执行带有.BAS后缀的源程序.

B21和B25分别为FP21的"D1",SR25"DS"读命令的数据采集程序.除命令和地址的区别外,程序基本上是相同的.用户可参考测量值和设定值的数据采集,PC机通讯口初始化,发送接收缓存区的访问,BCC校验,接收文件的自动分类和错误分类,通讯定时关系等子程序,扩展应用程序.

B1参考软件类似C2125功能.其中,可提供参考的子程序是SR25,FP21初始化的块文件的写入,FP21,SR25接收文件拆除字符串的应用程序.其余内容同B21和B25.

232T为通讯口测试程序,对COM1通讯口发接和屏幕显示发接字符,完成:

短路PC机COM1口的2,3端,检测PC机通讯口的好坏.

将先锋422转换器口的SDA接RDA,SDB接RDB后,分别检测四个通讯口.

参照 的短接方式,在1200米外,利用示波测量发送波形的前沿,确定通讯线路的品质以及选择合适的通讯波特率。

§10. 在PC计算机上,采用BASICA语言,实现对FP21, SR25数据采集的编程例

附录一: B21. BAS程序清单

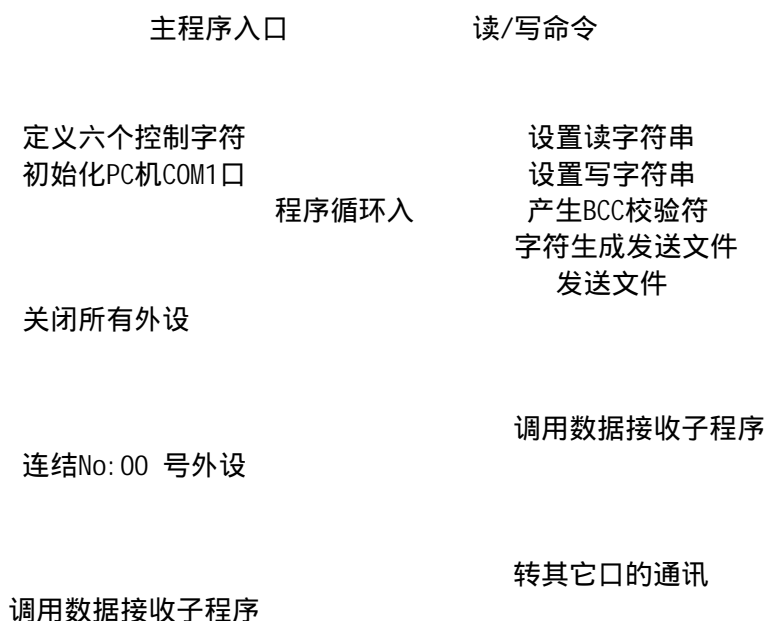
```
130 REM FP21的 "D1"命令数据采集软件PC机应用示例
131 REM 设置六个通讯控制字符
140 STX$=CHR$(2):ETX$=CHR$(3):EOT$=CHR$(4)
150 ENQ$=CHR$(5):ACK$=CHR$(6):NAK$=CHR$(&H15)
160 BPS$="1200":REM 设置1200通讯波特率
210 OPEN "COM1:"+BPS$+",E,7,1,CD,RS,CS,DS" AS #1:REM 初始化串行口,握手无效
230 PRINT#1,EOT$:REM 关闭总线上的外设(GO TO SLEEP)
500 CMD$="D1":REM 这里可换成SR25的"DS"命令
510 ADR$="00":REM 访问口地址"00"号
520 RC=0:REM 开始通讯,设置重发次数"0"次
530 C$=EOT$+ADR$+ENQ$:GOSUB 1150:REM 调用连联结
540 GOSUB 560:REM 调用发送文件的BCC校验
545 GOSUB 640:REM 调用数据接收
550 GOTO 500:REM 循环或继续
560 CMD$=CMD$+ETX$:LEC=LEN(CMD$):BCC=0:REM BCC块效验
570 FOR I=1 TO LEC:S$=MID$(CMD$,I,1)
580 BCC=BCC+ASC(S$)
590 NEXT
600 BCC=BCC MOD 128
610 BCC$=CHR$(BCC)
620 TXD$=STX$+CMD$+BCC$
630 RETURN
640 PRINT #1, TXD$:REM 发送文件
641 PRINT "SENDING DATA = ";TXD$
650 T3=VAL(MID$(TIME$,7,2)):REM 2秒定时
660 IF EOF(1)=0 THEN 700:REM 接收缓存器有效否?
670 T4=VAL(MID$(TIME$,7,2))
680 IF ABS(T4-T3)<2 THEN 660:REM 超过等待时间否?
690 PRINT"超过2秒,通讯出错!":RETURN
700 D$="" :REM 接收数据
710 A$=INPUT$(1,#1):REM 接一个字符
720 D$=D$+A$:REM 拼字符串
730 IF A$=ACK$ THEN GOTO 1040:REM 写命令跳转
740 IF A$=ETX$ THEN 780:REM 读命令跳转
750 IF A$=NAK$ THEN 1060:REM 错误命令跳转
760 GOTO 710
770 REM 接收数据的BCC校验
780 A$=INPUT$(1,#1)
790 H$=D$:D$=D$+A$
800 LEC=LEN(H$):BCC=0
810 FOR I=2 TO LEC:S$=MID$(H$,I,1)
820 BCC=BCC+ASC(S$)
830 NEXT
840 BCC=BCC MOD 128
850 BC$=CHR$(BCC):TXD$=NAK$
860 IF A$<>BC$ THEN 1030:REM 接收数据BCC错误,跳转重发
```

```

870 LEC=LEN(D$):F$="":K=1:REM 除掉数据中的控制符
880 X$=MID$(D$,2,2) :REM STX$,ETX$,BCC$
890 FOR P=5 TO LEC :REM 以", "号为分割,确定接收参数
900 N$=MID$(D$,P,1) :REM 个数N,用于的字符串分解.(详见B1)
910 IF N$="," THEN U$(K)=F$:K=K+1:F$="" :GOTO 940
920 IF N$=ETX$ THEN U$(K)=F$:N=K :GOTO 990
930 F$=F$+N$
940 NEXT
990 PRINT"读数据BCC校验成功 ! " TAB(25) "RD = ";D$
1010 PRINT
1020 RETURN:REM 返回
1030 RC=RC+1
1035 IF RC< 4 THEN GOTO 640 ELSE PRINT"DATA BCC ERROR!!":GOTO 1010
1040 IF D$=ADR$+ACK$ THEN RETURN 1200:REM 连结或写命令判断
1050 PRINT"写命令成功! RD = ";:PRINT D$;TAB(32);"TD=";TXD$:GOTO 1020
1060 B$=LEFT$(D$,3):REM 查出文件的错误类型
1140 PRINT "错误号 No. = ";B$:GOTO 1020
1150 TS=0
1161 PRINT
1160 PRINT #1,C$:TS=TS+1
1161 PRINT "联结命令 = ";C$
1170 GOSUB 650
1180 IF TS<3 THEN 1160 ELSE PRINT"无法联结(查错:地址,通讯速度,联线等)"
1190 RETURN
1200 PRINT D$; " 地址连接成功 !!"
1210 RETURN 540

```

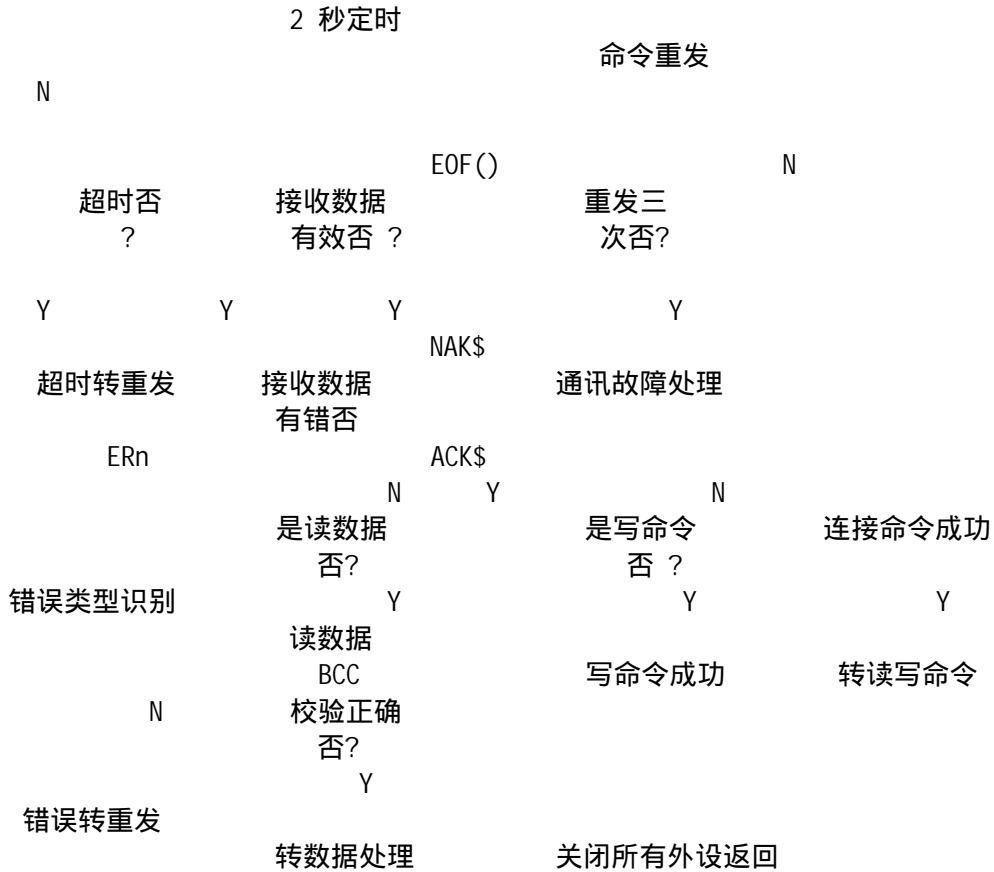
附录二: B21. BAS软件通讯程序流程图



联接命令成功转读写

FP 2 1、SR 2 5、SR 5 0通用接收发送子程序

程序入口



主程序入口

读/写命令

§11. 附录:

A. 通讯串口接线方法

RS-232C通讯口接线示意图

数据发送	SD	2	3	RD
数据接收	RD	3	2	SD
请求发送	RTS	4		FP-21
清除发送	CTS	5		RS-232C
数据设备准备	DSR	6		
载波检测	CD	8		
数据终端准备	DTR	20		

信号地 7 7

P C 机 25 针 RS-232C 仪表9针 RS232

RS-422A通讯口接线示意图

SD 2	3	□一 SDA(4)	RDA (4)接受低(-)
RD 3	2	□二 SDB(6)	RDB (6)接受高(+)
PC机 RS 4		□三 RDA(3)	SDA (3)发送低(-)
CS 5		□四 RDB(9)	SDB (9)发送高(+)
25 针 DS 6		SG(5)	SG (5)信号地
RS-232C CD 8		先锋232/422	屏蔽线 (1)
DR20		隔离转换器	
信号地 7	7		仪表的422接口

说明: 1. 短距离时, 可不使用信号地和屏蔽线.

2. 转换器提供了4个RS422驱动口, 在总线上都是并联的. 使用数量, 具体视现场的通讯距离来分配.

RS-422A通讯口接线示意图

3	3	RS-232C	4	4
2	2	RS-422A	6	6
PC机 7		通讯	3	3 FP-21
8		转换器	9	9
9 针 1			5	5 RS-422A
RS-232C 4				
6				
5	7			

PC机 RS-232C 串口25针与9针接线对照表:

9PIN	1	2	3	4	5	6	7	8	9
25PIN	8	3	2	20	7	6	4	5	22
13				1		5			1
25				14		9			6

25针连接器接线图

九针准连接器接线图

B. RS232C通讯口的技术数据

信号电平: EIA RS-232C 电平(±12V)
 通讯方式: RS232C 3线半双工
 同步系统: 起始位-停止位
 通讯距离: RS232C 15 米

通讯速度: 1200, 2400, 4800, 9600(SR25) 波特率
数据格式: 1. 数据7位, 一个偶校验位, 一个停止位
2. 数据8位, 无校验位, 一个停止位
数据块校验: 数据和(单字节)
通讯码: ASCII
控制码: 未使用
连接台数: RS-232C 1 台

C RS422通讯接口的技术数据

信号电平: EIA RS422A 电平 5V差动
通讯方式: RS422A 4线半双工(多路)
同步系统: 起始-停止位同位
通讯距离: RS422A 1200 米
通讯速度: 1200, 2400, 4800, 9600(SR25) 波特率
数据格式: 1. 数据7位, 一个偶校验位, 一个停止位
2. 数据8位, 无校验位, 一个停止位
数据块校验: 数据和(单字节)
通讯码: ASCII
控制码: 未使用
连接台数: RS-422A 10 台 1.2公里
RS-422A 32 台(配先锋RS232/422接口转换器)

C语言程序例(仅供参考!)

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<bios.h>
#include<string.h>
#define CH1_PA 0x3f8
void transmitter();
int receiver();
void main()

    int length, i, h, j, m, number, flag, k;
    char receive[100], comm[3], ccc[4];
    char ch;
    char send[]="D1"; /*发送命令*/
    printf("please input number:");
    scanf("%d", &k);
    outportb(CH1_PA+4, 0); /*disable interrupt, RTS, DTR*/
    outportb(CH1_PA+1, 0); /*disable interrupt enable register*/
    bioscom(0, 0xfa, 1); /*9600, 7, e, 1*/
    clrscr();
    gotoxy(26, 1);
    printf("SHIMADEN CONTROLLER");
    gotoxy(10, 2); printf("No");
    gotoxy(16, 2); printf("SV");
    gotoxy(22, 2); printf("PV");
    gotoxy(28, 2); printf("OP");
    gotoxy(35, 2); printf("ALM1");
    gotoxy(43, 2); printf("ALM2");
```

```

gotoxy(50, 2); printf("LBA");
gotoxy(58, 2); printf("ST");
gotoxy(64, 2); printf("A/M");
while(!kbhit())

for(number=0; number<k; number++)
h=number+1;
gotoxy(10, 3+number); printf("%d", h);
itoa(h, comm, 10);
if(h<10) send[1]='0'; send[2]=comm[0];
else send[1]=comm[0]; send[2]=comm[1];
length=strlen(send);
for(m=0; m<3; m++) /* Retry */

transmitter(send, CH1_PA, length);
i=receiver(receive, CH1_PA);
if(i!=0) break;

if(i==0)
printf("Timeout");
else gotoxy(15, 3+number);
for(j=5; j<9; j++)
printf("%c", receive[j]);
gotoxy(21, 3+number);
for(j=10; j<14; j++)
printf("%c", receive[j]);
gotoxy(27, 3+number);
for(j=15; j<19; j++)
printf("%c", receive[j]);
for(j=21; j<24; j++) ccc[j-21]=receive[j];
flag=atoi(ccc);
gotoxy(35, 3+number);
if(flag&0x01) printf("SAFE");
else printf("ACTI");
gotoxy(43, 3+number);
if(flag&0x02) printf("SAFE");
else printf("ACTI");
gotoxy(50, 3+number);
if(flag&0x10) printf("SAFE");
else printf("ACTI");
gotoxy(57, 3+number);
if(flag&0x04) printf("ENAB");
else printf("DISA");
gotoxy(64, 3+number);
if(flag&0x20) printf("MANU");
else printf("AUTO");

```

```

/*Transmitter Function*/
void transmitter(char *send,int BASE_ADD,int len)

int j;
outportb(BASE_ADD+4,0x02);/*enable RTS, enable send*/
for(j=0;j<len;j++)
outportb(BASE_ADD,send[j]); /*send a byte*/
/*check if transmitting shift register is empty*/
while((inportb(BASE_ADD+5)&0x40)==0);

return;

/* Receiver Function,success return the number of received byte,
failure return 0 */
int receiver(char *rec,int BASE_ADD)
int i=0;
long j=0;
outportb(BASE_ADD+4,0);/*disable RTS, enable receive*/
while((inportb(BASE_ADD+5)&0x01)!=1)/* receive data available? */
j++;if(j>20000) return(0);
rec[0]=inportb(BASE_ADD);/*receive data*/
do
while((inportb(BASE_ADD+5)&1)!=1)/*receive data available?*/
j++;if(j>20000) return(0);
rec[i]=inportb(BASE_ADD);/*receive data*/
i++;
while(rec[i-1]!=0x2a);
return(i);

_e data available?*/
j++;if(j>20000) return(0);
rec[i]=inportb(BASE_ADD);/*receive data*/
i++;

```