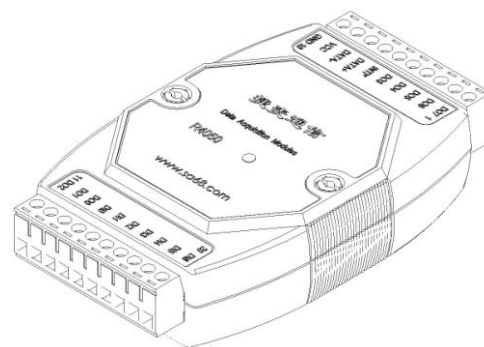


DATA ACQUISITION MODULES



R4050 开关量模块用户手册

2013年10月

<http://www.sa68.com>
info@sa68.com

北京捷麦通信器材有限公司

目录

1 概述	3
1.1 特点	3
1.2 外观和安装	3
1.3 技术指标	4
1.4 指示灯状态	4
2 电路连接	5
2.1 输入电路连接	5
2.2 输出电路连接	6
2.3 通信电路的连接	7
2.4 模块的输出值	7
3 通用指令和模块的基本设置	7
3.1 指令的基本格式	7
3.2 设置模块配置—%AANNTTCCFF	9
3.3 读配置信息—\$AA2	10
3.4 读版本信息—\$AAF	10
3.5 读复位状态—\$AA5	11
3.6 读模块名—\$AAM	11
3.7 写模块名—~AAO(数据)	12
4 采集和输出指令	13
4.1 同步采样—#**	13
4.2 读同步数据—\$AA4	14
4.3 数字输出—#AABBDD	14
4.4 设置数字输出—@AA(数据)	15
4.5 读数字输出/输入状态—\$AA6	16
4.6 读数字输入/输出状态—@AA	16
4.7 读锁存状态—\$AALS	17
4.8 清除锁存状态—\$AAC	18
4.9 读入计数值—#AAN	19
4.10 清计数器—\$AACN	19
5 遥控和报警指令及应用	20
5.1 模式	20
5.2 读模块模式—#AAM	22
5.3 设置模块模式—#AAMAB	23
5.4 读遥控目的地址—#AAR	24
5.5 写遥控目的地址—#AAR(数据)	24
5.6 读重发间隔时间—#AAT	25
5.7 设置重发间隔时间—#AATDD	25
6 看门狗的使用和指令	26
6.1 ~**	26
6.2 ~AA0	27
6.3 ~AA1	27
6.4 ~AA2	28
6.5 ~AA3EVV	28
6.6 ~AA4V	29
6.7 ~AA5V	30

附件 1: ASCII 码对照表	31
------------------------	----

1 概述

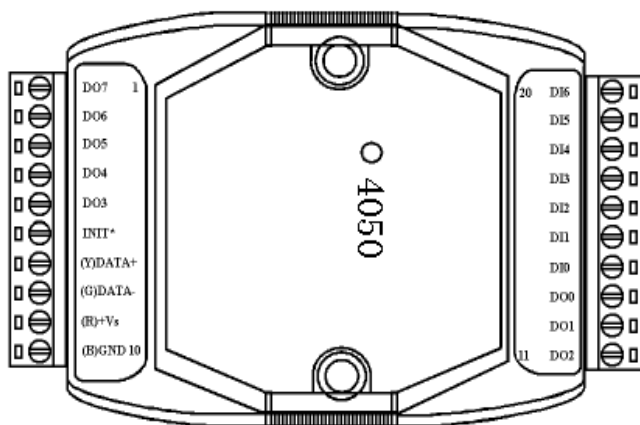
1.1 特点

R4050模块是R4000系列采集模块中的开关量输入输出模块，具有7路电压开关量输入，8路晶体管集电极开路输出。特点如下：

- ✧ 输入通道：7路，电压开关量输入。
- ✧ 输出通道：8路，晶体管集电极开路输出
- ✧ 8-30V较宽的供电电压
- ✧ 串口带有电源隔离
- ✧ 具有与其他开关量模块联网的遥控功能
- ✧ 具有主动上传的报警功能
- ✧ 具有主机看门狗功能，看门狗参数可设置
- ✧ 具有突发信号的锁存功能
- ✧ 具有重新上电的监测功能
- ✧ 具有网络上相同模块的同步采样功能
- ✧ 输入通道具有计数功能

1.2 外观和安装

端子分配和外观见下图：



下表是模块的端子说明表

端子号	标识	说明	说明	标识	端子号
1	D07	输出7	输入6	DI6	20
2	D06	输出6	输入5	DI5	19
3	D05	输出5	输入4	DI4	18
4	D04	输出4	输入3	DI3	17
5	D03	输出3	输入2	DI2	16
6	INIT*	参数还原端子	输入1	DI1	15
7	DATA+	485数据线正	输入0	DI0	14
8	DATA_	485数据线负	输出2	D02	13
9	+VS	电源8-30V	输出1	D01	12
10	GND	电源地	输出0	D00	11

1.3 技术指标

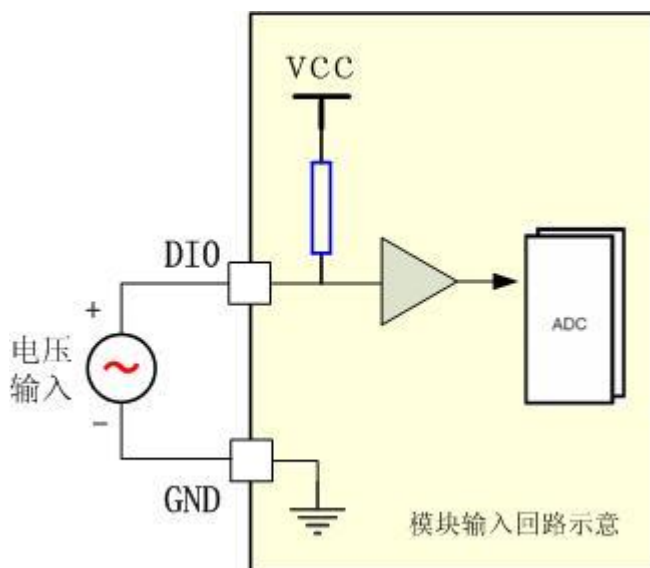
参数名称	指标
电源电压	DC8—30V
功耗	0.4W
输入逻辑 0 电压	< 1V
输入逻辑 1 电压	> 4—25V
输入阻抗	> 3000Ω
输出负载电压	< 25V
输出负载电流	< 150mA
串口隔离电压	> 1000V
体积	121.5*72*33mm
重量	

1.4 指示灯状态

等待：绿灯亮，
 接收：绿灯闪，
 发送：红灯闪，
 看门狗溢出：红灯亮。

2 电路连接

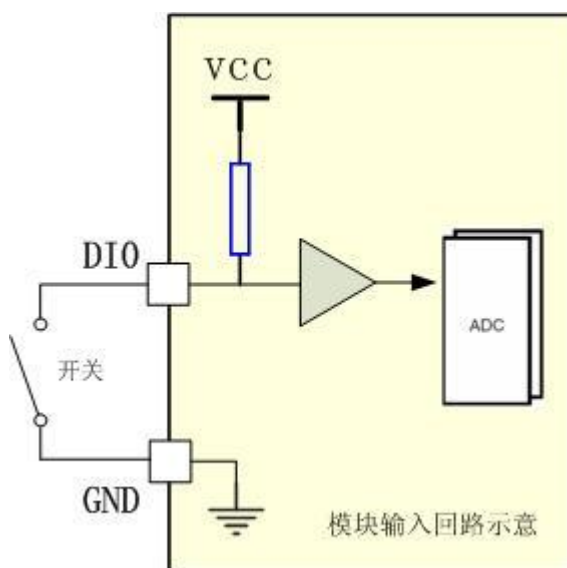
2.1 输入电路连接



当电压输入大于 2.5V 时输入的状态为逻辑 1，当电压小于 0.5V 是输入的状态为逻辑 0。

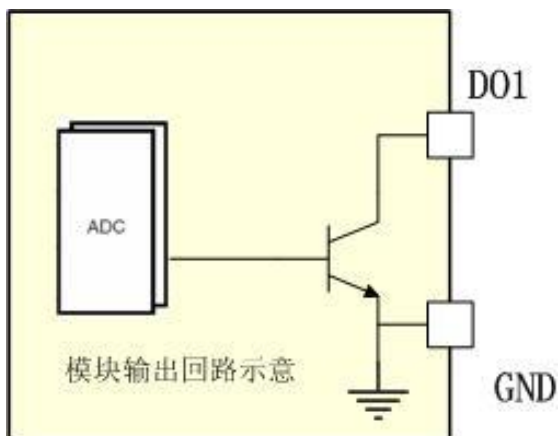
当端子空载不接电压时，因为模块内部有上拉电阻 R，输入的状态是逻辑 1。

如果输入只接开关，电路如下：



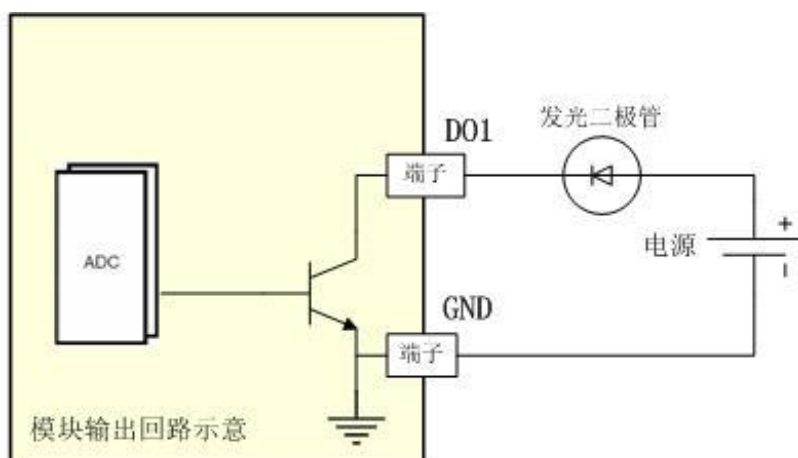
当开关闭合时输入状态为逻辑 0，当开关打开时输入状态为逻辑 1

2.2 输出电路连接

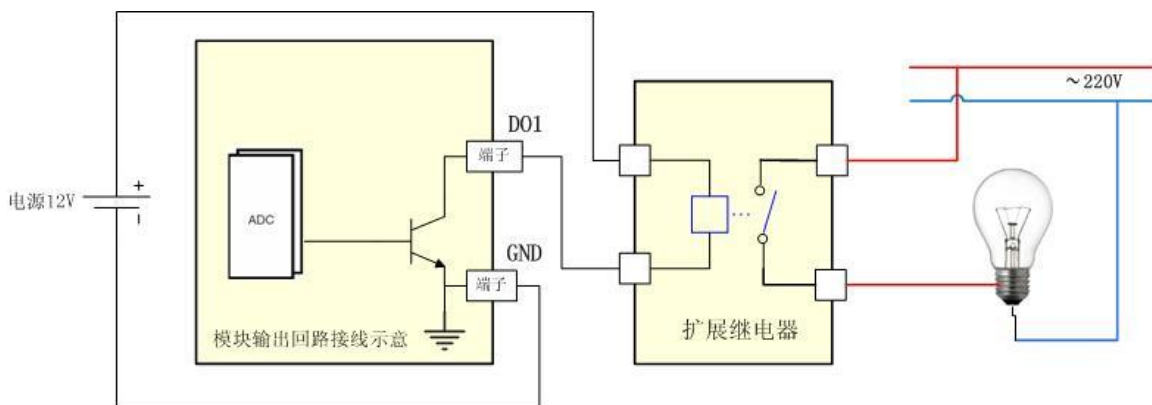


当模块输出逻辑 1 时晶体管导通，当模块输出逻辑 0 时晶体管截止。

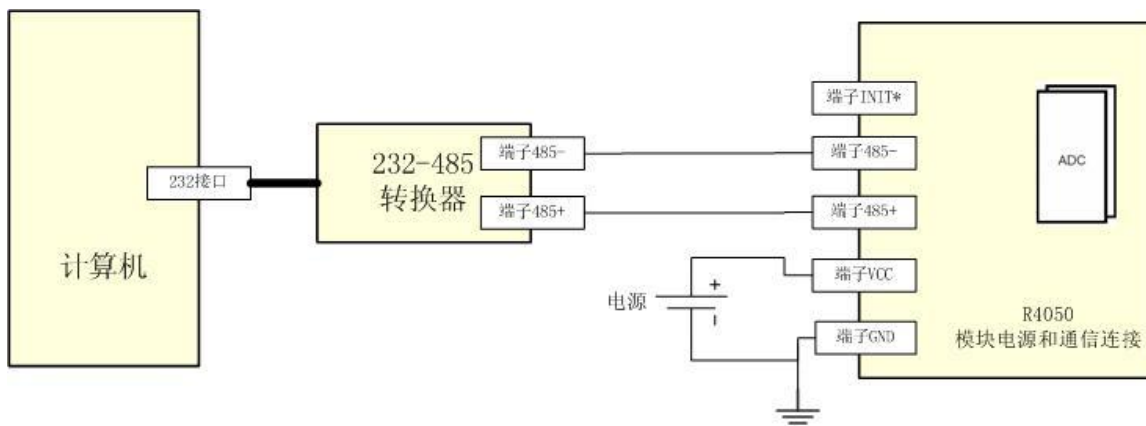
如果使用模块控制一个电流小于 150 毫安的负载时（例如一个发光二极管）电路如下图：



如果控制一个大电流或交流电负载，就得用一个扩展电流的继电器，连接电路如下：



2.3 通信电路的连接



2.4 模块的输出值

模块输出有三种不同情况：

- <1> 安全值. 如果主看门狗溢出状态被设置, 输出将会变为安全值。当模块接收到输出指令, 如@AA(数据) 或#AABBDD, 模块将会忽略指令, 并且不会转换输出到输出指令值。当主看门狗溢出超时, 主看门狗溢出状态被设置并存储到**EEPROM**, 且只有指令~AA1可清除。如果用户想转换输出, 他首先要清除看门狗溢出状态, 并且发送输出指令来转换输出值为目标值。
- <2> 上电值. 只有当模块复位, 且主看门狗溢出状态被清除时, 模块的输出值才被设置为预先确定的上电值。
- <3> 输出指令值. 如果主看门狗溢出状态被清除, 且用户发送一个数字输出指令, 如@AA(数据) 或#AABBDD, 发送到模块以转换输出值。模块将成功响应(以>响应)。

3 通用指令和模块的基本设置

3.1 指令的基本格式

本文中ASCII字符直接使用字符表达。如指令\$01M。如果指令中有数据, 每个字节的数据用2个ASCII码表示。如数据81H, 用ASCII码表达成81。

每条指令都是由ASCII码串组成, 如: \$01M、!01400600(cr)。在通信时, 每个ASCII码对应二进制的的一个字节, ASCII码与二进制的对应关系详见附件1。如果需使用二进制表达字节内容时, 使用字节的十六进制表达方式, 并在字节的后面加大写的H。如ASCII码 A, 表达成二进制为41H。

指令ASCII码串\$01M的二进制表达为：24H30H31H4DH0DH

指令ASCII码串!01400600(cr)的二进制表达为：21H30H31H34H30H30H36H30H30H0D

指令格式：[分隔符][地址][指令][CHK][cr]

[分隔符]：1位ASCII码。有：！有效响应分隔符

？无效响应分隔符

@、#、¥、%等！

[地址]：2位ASCII码。范围：0-255。指令中常用ASCII码AA表示。

[指令]：不同的指令的长度不同

[CHK]：校验和，用2个ASCII码表达二进制的1个字节。校验和的算法如下：

1. 除了cr字符，计算所有指令（或响应）字符串的ASCII总和。

2. 将字符串总和标记为0ffh.

例如：

指令字符串：\$012(cr)

字符串总和=‘\$’+‘0’+‘1’+‘2’=24H+30H+31H+32H=B7H（校验和的二进制为1个字节）

校验和是B7H, [CHK] = “B7”，这里将B7H变成了ASCII码的“B7”

则带校验的指令字符串为：\$012B7(cr)

响应字符串：!01400600(cr)

字符串总和：‘!’+‘0’+‘1’+‘4’+‘0’+‘0’+‘6’+‘0’+‘0’

=21H+30H+31H+34H+30H+30H+36H+30H+30H=1ACH（进位要忽略）

校验和是ACH, [CHK] = “AC”

带校验的响应字符串：!01400600AC(cr)

[cr]：指令结束符，是一位非显示的ASCII码，二进制码为（0DH）

响应格式：[分隔符][地址][指令][CHK][cr]

不同的指令会有不同的响应结构，语法或者通信错误可能导致没有响应。详细参照指令的解释部分

通用指令包括下表的指令

同通用指令设置		
指令	响应	描述
%AANNTTCCFF	!AA	设置模块配置
\$AA2	!AATTCCFF	读取配置
\$AAF	!AA(DATA)	读软硬件版本号
\$AA5	!AAS	读复位状态

\$AAM	!AA(DATA)	读模块名
~AAO(DATA)	!AA	设置模块名

3.2 设置模块配置—%AANNTTCCFF

描述：设置模块配置

语法： %AANNTTCCFF[CHK](cr)

% 分隔符

AA 模块设置地址(00到FF)

NN 模块的新地址(00到FF)

TT 本模块采用类型40，是一个固定数字。

CC 模块新波特率设置。当转换波特率时需要将INIT* 接地
波特率设置（CC）：

代码	03	04	05	06	07
波特率	1200	2400	4800	9600	19200

FF 新的数据格式设置。当改变校验设置时需要将INIT*接地。

数据格式设置（FF）表如下，下面的表中第一行是数据格式的二进制位的位置，使用第6位表达计数器的更新方式，第7位表达是否使用校验位。

7	6	5	4	3	2	1	0
*1	*2	0	0	0	0	0	0

1：计数器更新方式：0=下降沿；1=上升沿。

2：校验位：0=禁止；1=允许

如是下降沿计数且禁止校验位，FF的二进制数值为：40H，ASCII为：40

响应：有效指令：!AA[CHK](cr)

无效指令：?AA[CHK](cr)

例如：

指令： %0102400600 接收： !02

设置模块地址01为02，返回成功。

INIT*端子的作用：每个R4000模块都内置了一块EEPROM 来存储配置信息，如地址，方式，波特率和其他信息。有时，用户会忘记模块的这些配置。为此，R4000 有一个特殊的模式“INIT 模式”，它可以用来帮助用户解决这些问题。在“INIT 模式”下模块被强行置地址为**00**，波特率=**9600bps**，无校验和。

要启动INIT模式,请参照以下步骤:

步骤1. 关掉模块电源

步骤2. 将INIT* 端子接地。

步骤3. 打开电源

步骤4. 以9600bps发送指令\$002(cr)来读取存储在EEPROM中的配置。

默认设置为:

地址: 01

波特率: 9600bps

效验: 禁止

工作模式: 1

3.3 读配置信息—\$AA2

描述: 读配置信息

指令: \$AA2[CHK](cr)

\$ 字符分隔符

AA 读模块地址(00 到 FF)

2 读配置指令

响应: 有效指令: !AATTCCFF[CHK](cr)

无效指令: ?AA[CHK](cr)

例如:

指令: \$012 接收: !01400600

读地址 01 的状态, 返回 DIO 模式, 波特率 9600, 无校验。

3.4 读版本信息—\$AAF

描述: 读版本信息

指令: \$AAF[CHK](cr)

\$ 字符分隔符

AA 读模块地址(00 到 FF)

F 读版本信息指令

响应: 有效指令: !AA(数据)[CHK](cr)

无效指令: ?AA[CHK](cr)

(数据) 模块的版本信息

例如:

指令: \$01F 接收: !01EEAC2
读地址为 01 的版本信息, 返回版本 EEAC2.
指令: \$02F 接收: !02EEAC1
读地址为 02 的版本信息, 返回版本 EEAC1.

3.5 读复位状态—\$AA5

描述: 当模块重新上电后, 复位状态为 1。使用这条指令读复位状态后如果读前复位状态为 1, 发送这条指令后返回这个状态, 并将复位状态置为 0。通常使用这条指令检查模块在上一次使用这条指令后有没有发生重新上电现象。

指令: \$AA5[CHK](cr)

\$ 字符分隔符
AA 读模块地址 (00 到 FF)
5 读复位状态指令

响应: 有效指令: !AAS[CHK](cr)

无效指令: ?AA[CHK](cr)

S 复位状态, 1 = 模块已被复位, 0 = 模块还没被复位。

例如:

指令: \$015 接收: !011
读地址为 01 的复位状态, 返回第一次读数。模块已被复位
指令: \$015 接收: !010
读地址为 01 的复位状态, 返回无复位发生状态。

3.6 读模块名—\$AAM

描述: 读模块名称

指令: \$AAM[CHK](cr)

\$ 字符分隔符

AA 读模块地址 (00 到 FF)

M 读模块名指令

响应: 有效指令: !AA(数据)[CHK](cr)

无效指令: ?AA[CHK](cr)

(数据) 模块名称

例如:

指令: \$01M 接收: !014050

读地址为 01 的模块名, 返回名 4050.

指令: \$03M 接收: !034050

读地址为 03 的模块名, 返回名 4050.

3.7 写模块名—~AAO(数据)

描述: 设置模块名称

指令: ~AAO(数据)[CHK](cr)

~ 字符分隔符

AA 设置模块地址 (00 到 FF)

O 设置模块名指令

(数据) 模块新名称, 最大 9 字符, 字符只能为大写字母或者数字

响应: 有效指令: !AA[CHK](cr)

无效指令: ?AA[CHK](cr)

例如:

指令: ~0104050 接收: !01

设置地址为 01 的模块名 4050, 返回成功.

指令: \$01M 接收: !014050

读地址为 01 的模块名, 返回名 4050.

4 采集和输出指令

采集和输出指令见下表：

采集和输出指令		
***	无响应	同步取样
\$AA4	!S(DATA)	读同步数据
#AABBDD	>	数字输出
\$AA6	!(DATA)	读数字输入/输出状态
@AA(DATA)	>	设置数字输出
@AA	>(DATA)	读数字输入
\$AALS	!(DATA)	读锁存数字输入
\$AAC	!AA	清除锁存数字输入
#AAN	!AA(DATA)	读数字输入计数器
\$AACN	!AA	清除数字输入计数

读数据输入/输出格式：

\$AA6,\$AA4,\$AALS 指令的返回数据共有3个数据字节，每个数据字节使用2个ASCII码表示。其中第一第二字节的含义如下表，第三个字节为00

@AA数据： (第一字节)(第二字节)

	第一字节		第二字节	
R4050	DO(0-6)	00-7F	DI(0-7)	00-FF

4.1 同步采样—***

描述：同步采样。当总线上的R4050模块收到这条指令后会将开关量的输入状态同时存入同步采样寄存器，以备\$AA4指令读取。这是一条广播指令，无响应。

语法： *****[CHK](cr)**

字符分隔符

** 同步采样指令

响应： 无响应

例如：

指令： *** 无响应

给所有模块发送同步采样指令

指令: \$014 接收: !10F0000

从地址01读同步数据, 返回S=1, 第一次读数及数据。

4.2 读同步数据—\$AA4

描述: 读同步数据

指令: \$AA4[CHK](cr)

\$ 字符分隔符

AA 读模块地址(00到FF)

4 读同步地址的指令

响应: 有效指令: !S(数据)[CHK](cr)

 无效指令: ?AA[CHK](cr)

语法错误或者通信错误可能导致无响应。

! 有效指令分隔符

? 无效指令分隔符

AA 模块响应地址(00到FF)

S 同步地址状态, 1 = 第一次读数, 0 = 已读数
(数据)同步DIO值

例如:

指令: \$014 接收: ?01

读地址为01的同步数据, 返回无效数据。.

指令: #** 接收: 无响应

发送给所有模块同步数据。

指令: \$014 接收: !10F0000

读地址为01的同步数据, 返回S=1, 第一次读数且同步数据为0F00.

4.3 数字输出—#AABBDD

描述: 数字输出

指令: #AABBDD[CHK](cr)

字符分隔符

AA 读模块地址(00 to FF)

BB 输出指令和参数

DD 输出值。

对于多通道输出, BB = 00; DD=00到FF,对应DO (0-7)

对于单通道输出, BB = 1C, C用来选择通道; DD=00输出清零, 而01来设置输出。

响应: 有效指令: >[CHK] (cr)

无效指令: ?[CHK] (cr)

可忽略指令: ![CHK] (cr)

语法错误或通信错误可能导致无响应。

> 有效指令的分隔符

? 无效指令的分隔符

! 可忽略指令的分隔符. 模块主看门狗溢出状态被设置, 并且输出被设置为安全值。

例如:

指令: #0100FF 接收: >

设置地址为01的输出值为FF, 返回成功。

指令: #021001 接收: >

设置地址为02的通道0开, 返回成功。

指令: #021801 接收: ?

返回通道错误, 因为R4050只有8个通道 (0-7)。

指令: #0300FF 接收: !

设置地址为03的输出值为FF, 返回可忽略, 模块主看门狗溢出状态被设置, 输出被设置为安全值。

4.4 设置数字输出—@AA(数据)

描述: 设置数字输出。本条指令与#AABBDD指令的不同点是: 本条指令不能单路输出, 且指令长度较短。

指令: @AA(数据)[CHK](cr)

@ 字符分隔符

AA 设置模块地址(00到FF)

(数据) 输出值, 两个字符, 从00到FF

响应: 无响应

例如：

指令： @0100 接收： 无响应

模块 01 的输出值为 00, 返回成功.

4.5 读数字输出/输入状态—\$AA6

描述： 读数字输出/输入状态

指令： **\$AA6[CHK](cr)**

\$ 字符分隔符

AA 读模块地址(00到FF)

6 读数字输入/输出状态的指令

响应： 有效指令： **!(数据)[CHK](cr)**

无效指令： **?AA[CHK](cr)**

语法错误或通信错误可能导致无响应。

! 有效指令分隔符

? 无效指令分隔符

AA 响应模块地址(00到FF)

(数据) 数字输出/输入值

例如：

指令： \$016 接收： !0F0000

数字输出0-3通道为“ON”。

4.6 读数字输入/输出状态—@AA

描述： 读数字输入/输出状态本指令与\$AA6的不同点是：本指令的响应为2字节数据，\$AA6

指令3个字节，最后一个字节为00.

指令： **@AA[CHK](cr)**

@ 字符分隔符

AA 读模块地址(00到FF)

响应： 有效指令： **>(数据)[CHK](cr)**

无效指令： **?AA[CHK](cr)**

语法错误或通信错误可能导致无响应。

> 有效指令分隔符

? 无效指令分隔符

AA 响应模块地址(00到FF)

(数据) 读DIO状态

例如：

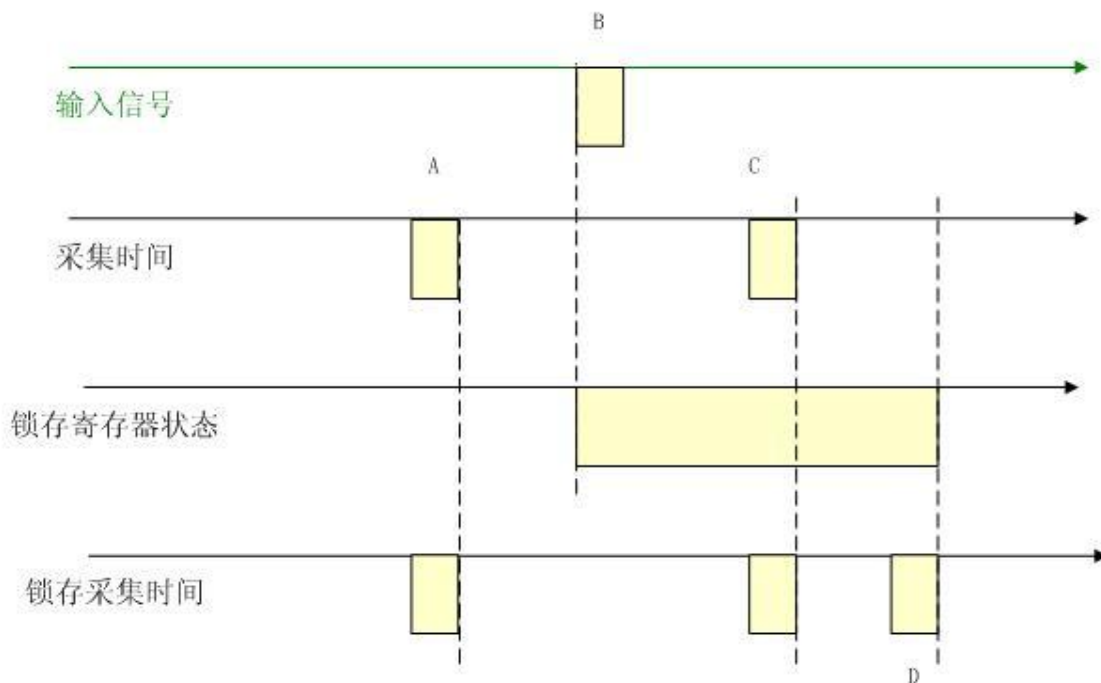
指令： @01 接收： >0F00

读地址为01的DIO状态，返回0F00.

4.7 读锁存状态—\$AALS

描述： 每个输入端都有一个高位的读锁存数字输入寄存器和一个低位锁存输入寄存器。

有时信号的变化发生在两次采集的间隔中，靠采集指令不能发现这种变化。R4050通过访问数据输入锁存寄存器就能发现这种数据的变化。示意图如下：



在这个例子中，输入信号在时间点B发生了变化，如果使用一般采集指令，在时间点A和时间点C采集输入状态，采集命令均不会采集到B点的变化，如果设置了锁存功能，当B点发生变化后锁存器也发生了变化，当主机在C点采集锁存器值时就能采集到B点发生的变化，在C点采集完成后主机在D点发生锁存器复位命令，使锁存器复位，以便再次捕捉输入信号的变化。

指令： **\$AALS[CHK](cr)**

\$ 字符分隔符

AA 读模块地址(00到FF)

L 读锁存数字输入指令

S 1 =选择高位锁存状态, 0 =选择低位锁存状态

响应: 有效指令: **!(数据)[CHK](cr)**

无效指令: **?AA[CHK](cr)**

语法错误或通信错误可能导致无响应。

! 有效指令分隔符

? 无效指令分隔符

AA 响应模块地址(00到FF)

(数据) 读状态.1=输入通道锁存, 0=输入通道不锁存.

例如:

指令: \$01L0 接收: !012300

读地址为01的低位锁存地址, 返回0123.

指令: \$01C 接收: !01

清零地址为01的锁存数据, 返回成功.

指令: \$01L0 接收: !000000

读地址为01的低位锁存地址, 返回0.

4.8 清除锁存状态—\$AAC

描述: 清除锁存数字输入

指令: **\$AAC[CHK](cr)**

\$ 字符分隔符

AA 设置模块地址(00到FF)

C 清除锁存数字输入指令

响应: 有效指令: **!AA[CHK](cr)**

无效指令: **?AA[CHK](cr)**

语法错误或通信错误可能导致无响应,

! 有效指令分隔符

? 无效指令分隔符

AA 响应模块地址(00到FF)

例如:

指令: \$01L0 接收: !01FFFF00

读地址为01的低位锁存数据, 返回FFFF.

指令: \$01C 接收: !01

清除地址为01的锁存数据, 返回成功

指令: \$01L0 接收: !01000000

读地址为01的低位锁存数据, 返回0000.

4.9 读入计数值—#AAN

描述: 从通道N读数字输入计时器(计数器最大频率为100Hz)。

指令: #AAN[CHK](cr)

字符分隔符

AA 读模块地址(00到FF)

N 读通道

响应: 有效指令: !AA(数据)[CHK](cr)

无效指令: ?AA[CHK](cr)

语法错误或通信错误会导致无响应。

! 有效指令分隔符

? 无效指令分隔符

AA 模块响应地址(00到FF)

(数据) 数字输入计数器值按照十进制算, 从00000到65535。

例如:

指令: #032 接收: !0300103

读地址为03的通道2的数字输入计数器值, 返回值00103。

指令: #025 接收: ?02

读地址为02的通道5的数字输入计数器值, 返回通道为无用值。

4.10 清计数器—\$AACN

描述: 清零数字输入计数器

指令: \$AACN[CHK](cr)

\$ 字符分隔符

AA 设置模块地址(00到FF)

C 清零数字输入计数器指令

N 清零通道N的数字计数器

响应： 有效指令： **!AA[CHK](cr)**

无效指令： **?AA[CHK](cr)**

语法错误或通信错误可能导致无响应。

! 有效指令分隔符

? 无效指令分隔符

AA 响应模块地址(00到FF)

例如：

指令： **#010** 接收： **!0100123**

读地址为01输入通道0的计数器值，返回123.

指令： **\$01C0** 接收： **!01**

清零地址为01输入通道0的计数器值，返回成功.

指令： **#010** 接收： **!0100000**

读地址为01输入通道0的计数器值，返回0.

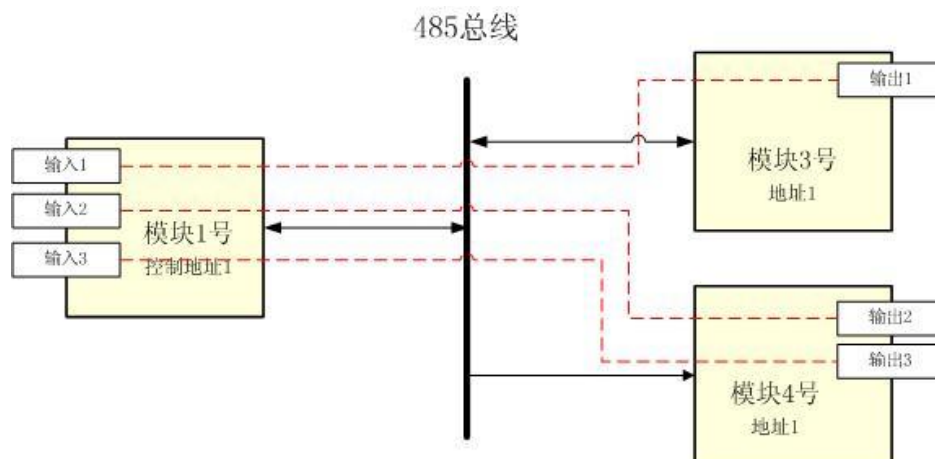
5 遥控和报警指令及应用

5.1 模式

模式一（普通模式）：在此模式下，模块接受上位机指令，随即作出响应，模块不会主动发出指令。

模式二（多通道遥控）：与模式一的不同点是当该模块的输入端口的输入电平发生变化后，模块向通信端口发送多通道写数据命令**#AA00DD**，写数据命令中的模块地址（即遥控的目的地址）可通过**#AAR（数据）**来进行设置。挂在同一总线上的R4050 模块收到这个写数据命令后会使对应的端口输出相同电平，用于单一模块控制一个或多个模块。

下图是使用模式2的应用例子：



控制关系如下：

模块1的输入1→模块3的输出1

模块1的输入2→模块4的输出2

模块2的输入3→模块4的输出3。

上述的控制关系在上图中使用虚线表示。

在这个图中，模块1的控制地址是1，被控模块的地址都是1。

模式三（单通道遥控）：与模式二的不同点是当该模块的输入端口的输入电平发生变化后，模块向通信端口发送单通道写数据命令#AA1CDD，挂在同一总线上的R4050 模块收到这个写数据命令后会使对应的端口输出相同电平。

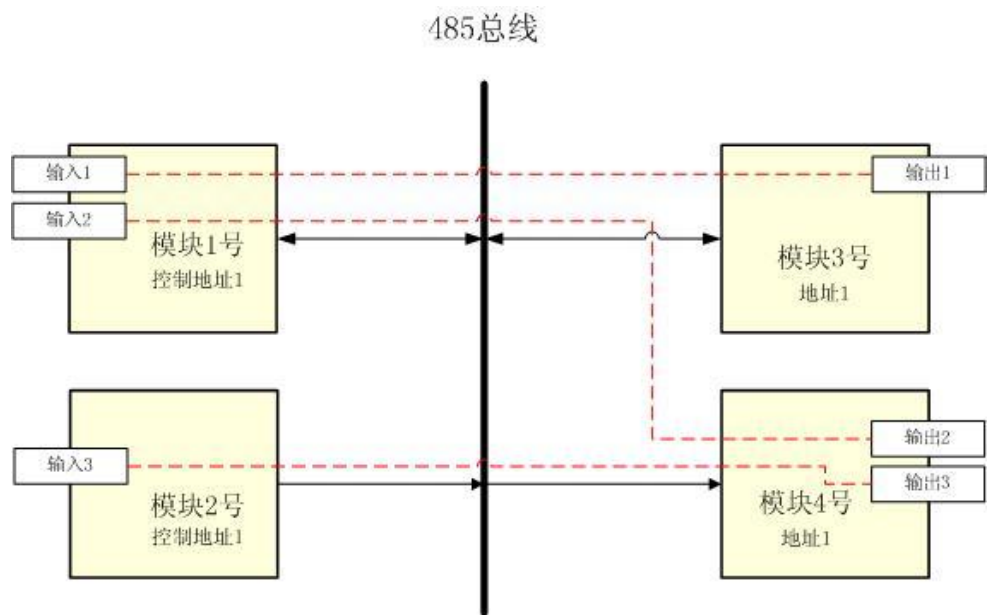
如果希望被控模块与控制模块的对应端子的输出完全一致应使用模式模式二，如果被控端有的模块还受其它总线上的控制器控制，应使用模式三。下图中4个模块的地点不同，模块1和模块2是控制模块。控制关系如下：

模块1的输入1→模块3的输出1

模块1的输入2→模块4的输出2

模块2的输入3→模块4的输出3。

上述的控制关系在下图中使用虚线表示。



在这个图中如果使用模式2，当发送模块2的输入状态变化数据后，模块3的输出1也会随着模块2的输入1变化，而不是保持模块1的输入1的状态。

模式四（报警）：与模式一的不同点是当该模块的输入端口的输入电平发生变化后，模块向通信端口发送(!AA(输出)(输入)00)。挂在同一总线上的控制中心收到这个命令后就收到了输入端口的变化信息。

遥控报警指令设置		
指令	响应	描述
#AAM	!AAMAB	读模块模式
#AAMAB	!AA	设置模块模式
#AAR	!AAOO	读遥控目的地址
#AAROO	!AA	设置遥控目的地址
#AAT	!AATDD	读间隔时间
#AATDD	!AA	设置间隔时间

5.2 读模块模式—#AAM

描述：读模块模式

指令： #AAM[CHK](cr)

字符分隔符

AA 设置模块地址(00到FF)

M 读模式指令

响应： 有效指令： **!AAMAB[CHK](cr)**

无效指令： **?AA[CHK](cr)**

语法错误或通信错误可能导致无响应。

! 有效指令分隔符

? 无效指令分隔符

AA 响应模块地址(00到FF)

A 模块模式1-4

B 重发次数（B）：1-9。为了保证数据的准确接收，增加了重发次数的设置。对于模式一无效。会主动发出数据的后3种模式有效。

例如：

指令： **#01M** 接收： **!01M11**

读地址为01的模式为1，重发次数为1。

指令： **#02M** 接收： **!02M25**

读地址为02的模式为2，重发次数为5。

5.3 设置模块模式—#AAMAB

描述：设置模块模式

指令： **#AAMAB[CHK](cr)**

字符分隔符

AA 设置模块地址(00到FF)

M 读模式指令

A 模块模式1-4

B 重发次数1-9

响应： 有效指令： **!AA [CHK](cr)**

无效指令： **?AA[CHK](cr)**

语法错误或通信错误可能导致无响应。

! 有效指令分隔符

? 无效指令分隔符

AA 响应模块地址(00到FF)

例如：

指令: #01M11 接收: !01M

设置地址为01的模式为1, 重发次数为1。

指令: #02M25 接收: !02M

设置地址为02的模式为2, 重发次数为5。

指令: #01M53 接收: !01M

设置地址为01的模式为5, 重发次数为3。

5.4 读遥控目的地址—#AAR

描述: 读遥控目的地址

指令: #AAR[CHK](cr)

字符分隔符

AA 设置模块地址(00到FF)

R 读/写遥控目的地址指令

响应: 有效指令: !AAR (数据) [CHK](cr)

无效指令: ?AA[CHK](cr)

语法错误或通信错误可能导致无响应。

! 有效指令分隔符

? 无效指令分隔符

AA 响应模块地址(00到FF)

(数据) 遥控目的地址

例如:

指令: #01R 接收: !01R02

读地址为01的遥控目的地址为02。

5.5 写遥控目的地址— #AAR (数据)

描述: 写遥控目的地址

指令: #AAR (数据) [CHK](cr)

字符分隔符

AA 设置模块地址(00到FF)

R 读/写遥控目的地址指令

(数据) 遥控目的地址

响应: 有效指令: **!AA [CHK](cr)**

无效指令: **?AA[CHK](cr)**

语法错误或通信错误可能导致无响应。

! 有效指令分隔符

? 无效指令分隔符

AA 响应模块地址(00到FF)

例如:

指令: **#01R03** 接收: **!03**

设置地址为01的遥控目的地址为03。

5.6 读重发间隔时间—#AAT

描述: 读重发间隔时间

指令: **#AAT[CHK](cr)**

字符分隔符

AA 设置模块地址(00到FF)

T 读重发间隔时间

响应: 有效指令: **!AATDD [CHK](cr)**

无效指令: **?AA[CHK](cr)**

语法错误或通信错误可能导致无响应。

! 有效指令分隔符

? 无效指令分隔符

AA 响应模块地址(00到FF)

DD 重发间隔时间, 01-FF; 5ms为1单位。

例如:

指令: **#01T** 接收: **!01T01**

读地址为01的重发间隔时间为5ms。

5.7 设置重发间隔时间—#AATDD

描述: 设置重发间隔时间

指令: **#AATDD[CHK](cr)**

字符分隔符

AA 设置模块地址(00到FF)
T 读重发间隔时间
DD 重发间隔时间, 01-FF; 5ms为1单位。

响应: 有效指令: **!AA [CHK](cr)**

无效指令: **?AA[CHK](cr)**

语法错误或通信错误可能导致无响应。

! 有效指令分隔符

? 无效指令分隔符

AA 响应模块地址(00到FF)

例如:

指令: #01T04 接收: !01

设置地址为01的重发间隔时间为20ms。

主机看门狗指令设置		
指令	响应	描述
~**	无响应	主机 OK
~AA0	! AASS	读模块状态
~AA1	! AA	复位模块状态
~AA2	! AA EVV	读主机看门狗空闲状态值
~AA3 EVV	! AA	设置主机看门狗空闲状态值
~AA4V	! AA(DATA)	读上电值/安全值
~AA5V	! AA	设置上电值/安全值

6 看门狗的使用和指令

6.1 ~**

描述: 主机正常.

主机发送“Host OK”信息给所有模块

指令: **~**[CHK](cr)**

~ 字符分隔符

****** 给所有模块的指令

响应: 无

例如:

指令: ~** 无响应

6.2 ~AA0

描述: 读模块状态

指令: ~AA0[CHK](cr)

~ 字符分隔符

AA 读模块地址(00到FF)

0 读模块状态指令

响应: 有效指令: !AASS[CHK](cr)

无效指令: ?AA[CHK](cr)

语法错误或通信错误会导致无响应。

! 有效指令分隔符

? 无效指令分隔符

AA 响应模块地址(00到FF)

SS 模块状态, 00=清除主看门狗溢出状态, 04=设置主看门狗溢出状态。此状态将会
储存到EEPROM并且只有指令~AA1能复位。

例如: 见~AA3EVV[CHK](cr)

6.3 ~AA1

描述: 复位模块状态

指令: ~AA1[CHK](cr)

~ 字符分隔符

AA 设置模块地址(00到FF)

1 复位模块状态指令

响应: 有效指令: !AA[CHK](cr)

无效指令: ?AA[CHK](cr)

语法错误或通信错误会导致无响应

! 有效指令分隔符

? 无效指令分隔符

AA 响应模块地址(00到FF)

例如: 见~AA3EVV[CHK](cr)

6.4~AA2

描述：读主看门狗溢出时间值

指令：~AA2[CHK](cr)

~ 字符分隔符

AA 读模块地址(00到FF)

2 读主看门狗溢出时间值指令

响应：有效指令：!AAEVV[CHK](cr)

无效指令：?AA[CHK](cr)

语法错误或通信错误会导致无响应

! 有效指令分隔符

? 无效指令分隔符

AA 响应模块地址(00到FF)

E 主看门狗工作状态, 1=允许, 0=禁止

VV 溢出时间值用16进制格式表示, 每个计数是0.1秒。01=0.1秒, FF=25.5 秒

例如：见~AA3EVV[CHK](cr)

6.5 ~AA3EVV

描述：设置主看门狗溢出时间值

指令：~AA3EVV[CHK](cr)

~ 字符分隔符

AA 设置模块地址(00到FF)

3 设置主看门狗溢出时间值指令

E 主机看门狗状态, 1=允许, 0=禁止

VV 溢出时间值, 从 01 到 FF, 每个计数是 0.1 秒

响应：有效指令：!AA[CHK](cr)

无效指令：?AA[CHK](cr)

语法错误或通信错误会导致无响应

! 有效指令分隔符

? 无效指令分隔符

AA 响应模块地址(00 到 FF)

例如：

指令：~010 接收：!0100

读地址为 01 的模块状态,返回主看门狗溢出状态被清除

指令：~013164 接收：!01

设置地址为 01 的主看门狗溢出时间值为 10.0 秒并且启动看门狗，返回成功。

指令： ~012 接收： !01164

读地址为 01 的主看门狗溢出时间值,返回主看门狗工作中，且超时间隔为 10.0 秒。

指令： ~** 无响应

复位看门狗计数器。

等待 10 秒并且不发送指令~**，模块的发光二极管将会闪烁。闪烁的发光二极管表示主看门狗溢出状态被设置。

指令： ~010 接收： !0104

读地址为 01 的模块状态，返回主看门狗溢出状态被设置。

指令： ~012 接收： !01064

读地址为 01 的主看门狗溢出时间值，返回机看门狗没有工作，并且超时间隔为 10.0 秒。

指令： ~011 接收： !01

复位地址为 01 的主看门狗溢出状态，返回成功且模块的发光二极管停止闪烁。

指令： ~010 接收： !0100

读地址为 01 的模块状态，返回主看门狗溢出状态被清除。

6.6 ~AA4V

描述：读上电值/安全值。

指令： ~AA4V[CHK](cr)

~ 字符分隔符

AA 读模块地址(00到FF)

4 读上电值/安全值指令

V P =读上电值, S =读安全值

响应： 有效指令： !AA(数据)[CHK](cr)

无效指令： ?AA[CHK](cr)

语法错误或通信错误会导致无响应。

! 有效指令分隔符

? 无效指令分隔符

AA 响应模块地址(00到FF)

(数据)上电值或安全值

对R4042/42D/43/43D (数据)来说为VVVV, VVVV是上电值(或安全值)。

对其他模块讲, (数据)就是VV00, VV是上电值(或安全值)。

例如:

指令: @0100 接收: 无响应
输出地址为01的值00, 返回成功.

指令: ~015S 接收: !01
设置地址为01的安全值, 返回成功.

指令: @01FF 接收: 无响应
输出地址为01的值FF, 返回成功.

指令: ~015P 接收: !01
设置地址为01的上电值, 返回成功.

指令: ~014S 接收: !010000
读地址为01的安全值, 返回0000.

指令: ~014P 接收: !01FFFF
读地址为01的上电值, 返回FFFF.

6.7 ~AA5V

描述: 设置上电值/安全值

指令: ~AA5V[CHK](cr)

~ 字符分隔符

AA 模块设置地址(00到FF)

5 设置上电值/安全值指令

V P = 将当前输出值设置为上电值,
 S = 将当前输出值设置为安全值。

响应: 有效指令: !AA[CHK](cr)

 无效指令: ?AA[CHK](cr)

语法错误或通信错误会导致无响应。

! 有效指令分隔符

? 无效指令分隔符

AA 响应模块地址(00到FF)

例如:

指令: @01AA 接收: 无响应

输出地址为01的值AA, 返回成功.

指令: ~015P 接收: !01

设置地址为01的上电值, 返回成功.

指令: @0155 接收: 无响应

输出地址为01的值55, 返回成功.

指令: ~015S 接收: !01

设置地址为01的安全值, 返回成功.

指令: ~014P 接收: !01AA00

读地址为01的上电值, 返回上电值AA.

指令: ~014S 接收: !015500

读地址为01的安全值, 返回安全值55.

附件 1: ASCII 码对照表

八进制	十六进制	十进制	字符	八进制	十六进制	十进制	字符
00	00	0	nul	100	40	64	@
01	01	1	soh	101	41	65	A
02	02	2	stx	102	42	66	B
03	03	3	etx	103	43	67	C
04	04	4	eot	104	44	68	D
05	05	5	enq	105	45	69	E
06	06	6	ack	106	46	70	F
07	07	7	bel	107	47	71	G
10	08	8	bs	110	48	72	H
11	09	9	ht	111	49	73	I
12	0a	10	lf	112	4a	74	J
13	0b	11	vt	113	4b	75	K
14	0c	12	ff	114	4c	76	L

八进制	十六进制	十进制	字符	八进制	十六进制	十进制	字符
15	0d	13	cr	115	4d	77	M
16	0e	14	so	116	4e	78	N
17	0f	15	si	117	4f	79	O
20	10	16	dle	120	50	80	P
21	11	17	dc1	121	51	81	Q
22	12	18	dc2	122	52	82	R
23	13	19	dc3	123	53	83	S
24	14	20	dc4	124	54	84	T
25	15	21	nak	125	55	85	U
26	16	22	syn	126	56	86	V
27	17	23	etb	127	57	87	W
30	18	24	can	130	58	88	X
31	19	25	em	131	59	89	Y
32	1a	26	sub	132	5a	90	Z
33	1b	27	esc	133	5b	91	[
34	1c	28	fs	134	5c	92	\
35	1d	29	gs	135	5d	93	]
36	1e	30	rs	136	5e	94	^
37	1f	31	us	137	5f	95	_
40	20	32	sp	140	60	96	'
41	21	33	!	141	61	97	a
42	22	34	"	142	62	98	b
43	23	35	#	143	63	99	c
44	24	36	\$	144	64	100	d
45	25	37	%	145	65	101	e

八进制	十六进制	十进制	字符	八进制	十六进制	十进制	字符
46	26	38	&	146	66	102	f
47	27	39	`	147	67	103	g
50	28	40	(	150	68	104	h
51	29	41	)	151	69	105	i
52	2a	42	*	152	6a	106	j
53	2b	43	+	153	6b	107	k
54	2c	44	,	154	6c	108	l
55	2d	45	-	155	6d	109	m
56	2e	46	.	156	6e	110	n
57	2f	47	/	157	6f	111	o
60	30	48	0	160	70	112	p
61	31	49	1	161	71	113	q
62	32	50	2	162	72	114	r
63	33	51	3	163	73	115	s
64	34	52	4	164	74	116	t
65	35	53	5	165	75	117	u
66	36	54	6	166	76	118	v
67	37	55	7	167	77	119	w
70	38	56	8	170	78	120	x
71	39	57	9	171	79	121	y
72	3a	58	:	172	7a	122	z
73	3b	59	;	173	7b	123	{
74	3c	60	<	174	7c	124	
75	3d	61	=	175	7d	125	}
76	3e	62	>	176	7e	126	~

八进制	十六进制	十进制	字符	八进制	十六进制	十进制	字符
77	3f	63	?	177	7f	127	del