

IR100S 车辆数据格式

1001+主机地址+从机地址+数据返回时间（月日时分秒）+1002+B0+数据长度+数据类型+产生车辆数据的时间（月日时分秒）+存储内容类型+1002+线圈 1 数据+1002+线圈 2 数据+.....+1002+线圈 n 数据+CRC+1003

①车辆数据：没有车辆经过时

```
<<<1001AABBCC0C10000D24011002B074010C10000D24005F100201000000000000100202000000
00000010020300000000000000100204000000000000001002050000000000000010020600000000000001002
070000000000000000100208000000000000001002090000000000000010020A0000000000000010020B00000000
000010020C000000000000006E141003
```

- 1001 报头开始，AA 是主机地址 170，BBCC 是 IR100 地址 187.204。
- 0C10000D2401 是主机发送查询命令的时间，即 12 月 16 日 13 时 36 分 01 秒。
- 1002 正文开始。
- B0 是数据的 MI 代码。
- 74 为数据长度，116 个字节（从下一字节到 CRC 之前，不含 CRC）。
- 01 是数据类型，表示车辆数据（02 为校时时刻时间数据，00 表示没有数据）。
- 0C10000D2400 是数据产生的时间，即 12 月 16 日 13 时 36 分 00 秒。
- 5F 为存储内容类型，表示 6 种数据都有。
- 01000000000000 是线圈数据共 7 字节，它的格式为：线圈号码（1 字节）+车辆计数（2 字节）+速度（1 字节）+车长（1 字节）+车头时距（1 字节）+道路占有率（1 字节），其余车道数据类似。
- 6E14 是数据包的 CRC。
- 1003 正文结束。

②车辆数据：六个车道都有车辆经过时

```
<<<1001AABBCC0C10000D26011002B074010C10000D26005F10020100011E1DFF06100202000000
0000001002030001281FFF041002040000000000000010020500012B1FFF0310020600000000000000100
2070001281EFF041002080000000000000010020900011A20FF0610020A0000000000000010020B00011A
1FFF0610020C00000000000000D06A1003
```

- 0100011E1DFF06 是 1 号线圈数据，车辆计数是 1，平均速度 1E 是 30km/h，车长 1D 是 2.9m。

- 030001281FFF04 是 3 号线圈数据, 车辆计数是 1, 平均速度 28 是 40km/h, 车长 1F 是 3.1m。
- 0500012B1FFF03 是 5 号线圈数据, 车辆计数是 1, 平均速度 2B 是 43km/h, 车长 1F 是 3.1m。
- 070001281EFF04 是 7 号线圈数据, 车辆计数是 1, 平均速度 28 是 40km/h, 车长 1E 是 3.0m。
- 0900011A20FF06 是 9 号线圈数据, 车辆计数是 1, 平均速度 1A 是 26km/h, 车长 20 是 3.2m。
- 0B00011A1FFF06 是 11 号线圈数据, 车辆计数是 1, 平均速度 1A 是 26km/h, 车长 1F 是 3.1m。

③逆行车辆报警信息:

<<< 1001AABBCC0C10000B3B2610022703451D8AE6791003

27 为逆行事件的 MI 代码, 03451D 即 2 车道, 速度: 69km/h, 车长: 2.9m。

4.其它数据

① 校时数据

>>> 1001AABBCC001002AC040C10000B353516471003

<<< 1001AABBCC0C10000B3536100201AC9FB61003

AC 为校时命令的 MI 代码。

在返回的信息中 1002 后的 01 为 ACK 代码, 表示操作成功。

②线圈信息

>>> 1001AABBCC00100203B7931003

<<< 1001AABBCC0C10000D2739100204FFF00005A11003

03 是查询线圈状态的 MI 代码, 04 为对应的线圈状态代码。

FFFFF0 是线圈信息。0 表示有线圈, F 表示没有线圈。FFF000 就说明三块 TD634ES 所连接的 12 个线圈是好的。

循环冗余码校验 (CRC) 仅供参考

1. 所有的数据包都有一个 16 位校验和字段。在 NP601 协议中使用的校验和的方法是 CRC-CCITT 多项式。这种检测错误的方法可以捕捉所有单和双的错误, 所有的错误跟着几位奇数, 这种错误检测的方法可以捕捉所有单、双错误, 所有的错误都跟着一个奇数位, 16 或更少位的区间误差, 17 位 99.997% 的区间误差, 18 位 99.998% 的区间误差。2. 数据发送期间, 除了 ETX 控制计数, 发送方计算在数据包内所有数据字节的 CRC, 在数据包的末端附加结果。接收方计算在数据包内所有数据字节的 CRC, 在数据包的末端附加结果。如果接收的数据是正确的, 在接收末端的 CRC 计算结果应该是 00h, 必需注意的是, CRC 也影响 DLE 编码。DLE 在做 CRC 计算前必须从两个 CRC 位上去掉。3. 程序执行 CRC-CCITT 计算如下: 4. 数据包一次供给 CRC 发生器一个字节。在一个新数据包开始时, 累加器被复位, 计算使用一个 16 位数据寄存器和一个 16 位累加器。5. 当一个新的数据字节被收到时, 在数据寄存器内的所有位都复

位，数据位被装载进寄存器的高目的位。（1）数据寄存器和累加器之间执行一个异或操作。（2）如果结果的最高标志位是“逻辑 0”，累加器的内容将进行逻辑左移，形成一个新的累加器。（3）如果结果的最高标志位是“逻辑 1”，累加器的内容将进行逻辑左移，与多项式 $X^{16}+X^{12}+X^5+X^0$ 进行一个异或操作，形成一个新的累加器。（4）数据寄存器的内容逻辑左移。6. 此过程字符串的每个字节循环 8 次（接收的字节每位一次），最后一个字节被完全编程后，累加器内的结果就是 CRC。7. 如果字节如下的例子：8. (0AH), (03H), (3EH), (44H), (B3H), (BEH), (DCH) 和 (DDH)，CRC 为 440H，发送的最高标志位是 44H。以上部分仅供参考

IR100 车辆检测器协议中的 10——1000 转义有什么含义？

【答】在 IR100 协议中规定：如果值是 10H 的字符被传送时，为了避免与报头开始、报头结束中的 10H 混淆，必须在后面加上 00H，接收时再去掉才是实际的数据。

(比如说需要传输的数据为 10 03，那么有可能被误认为是正文结束的字符 1003，所以要在 10 后加 00，实际传输的数据就为 10 00 03。)

例如：

如果下面的数据要被传送：

10 01 00 01 10 02 02 03 04 05 06 07 08 09 10 11 12 13 14 15 10 03

实际发送的数据如下：

10 01 00 01 10 02 02 03 04 05 06 07 08 09 10 00 11 12 13 14 15 10 03

接收后经过转义处理过的数据：

10 01 00 01 10 02 02 03 04 05 06 07 08 09 10 11 12 13 14 15 10 03