

协议

多路传输



协议

协议

历史:

➤ **VAN历史:**

- **1987** : 产生 VAN 协议
- **1990** : VAN 成为 AFNOR R-13708标准
- **1992** : 第一个上路的协议 (CITROËN XM)
- **1993** : 1000 辆车出现在生产线上 (CITROËN XM)
- **1994** : VAN 成为 ISO 11519-3 标准
- **1995** : Renault 放弃 VAN
- **1999** : 在 XSARA 和XSARA Picasso用第一个带 BSI-VAN 标准

协议

历史：

➤ CAN 历史：

- **1980** : 产生 CAN 协议
- **1991** : CAN 低速成为 ISO 标准
- **1992** : MERCEDES 在 S 级上用 CAN
- **1993** : CAN 高速成为 ISO 11898 标准
- **1994** : CAN 低速成为 ISO 11519-2 标准
- **1995** : PSA 用 CAN HS 网 (更换 VAN)
- **2000** : PSA 决定从在车身和舒适方面从 VAN 转到 CAN 低速

多路传输



概念和综合信息

协议

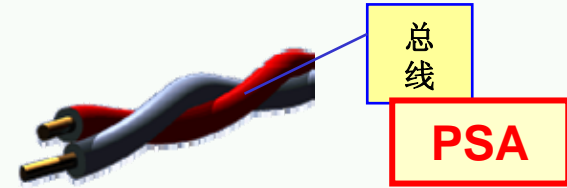
概念：

➤ 总线：

支持各个ECU之间进行的数据传递

CAN和VAN通讯协议不需要任何特殊总线(光导纤维、Hertz波、电线)

电线连接在汽车应用最好。



VAN 网和CAN网的总线包括两根导线

DATA 和 DATA/

CAN H 和CAN L

协议

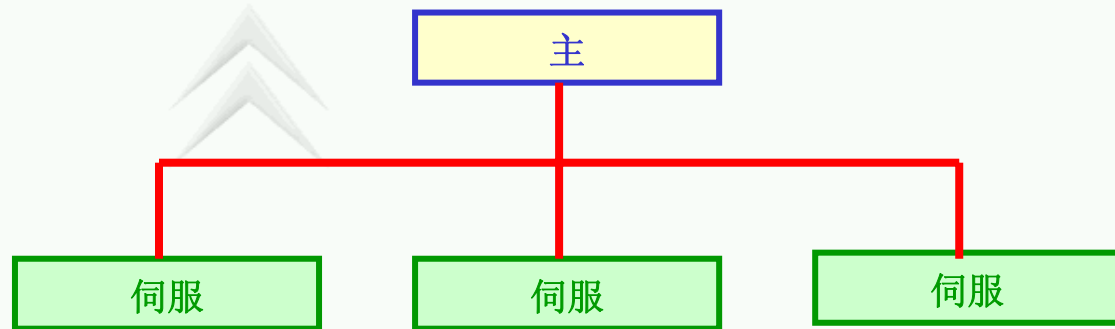
概念：

➤ 主和伺服概念

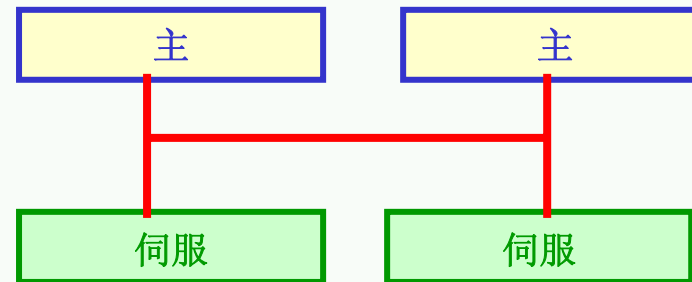
网络由制造商根据费用和功能来定义



主/伺网



多主/多伺网



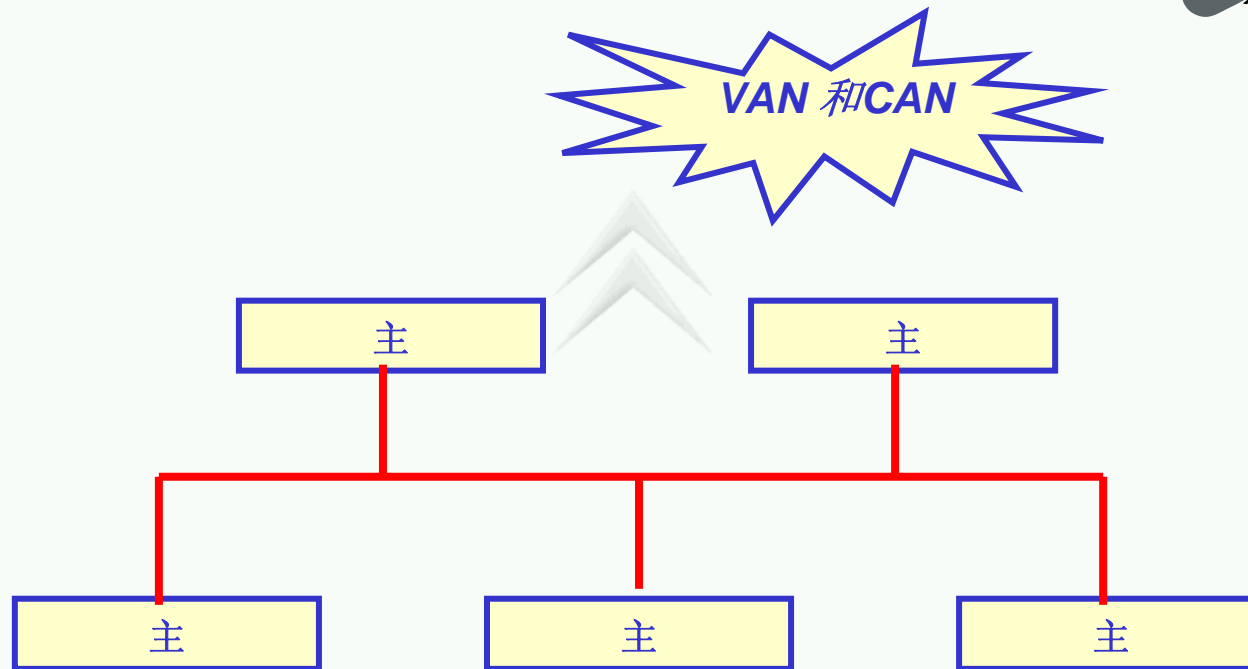
协议

概念：

➤ 主和伺服概念



多主网

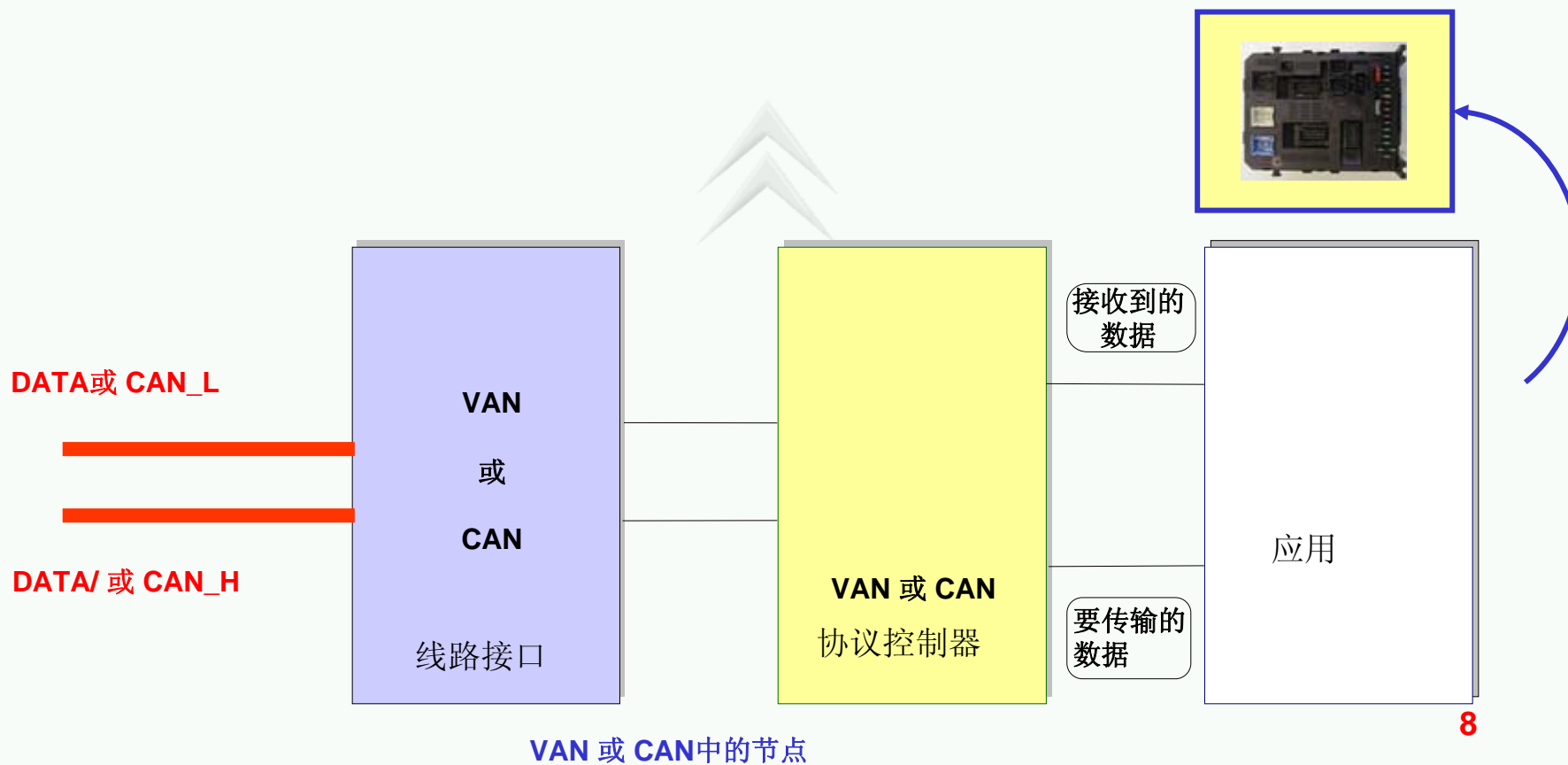


协议

概念：

▶ 节点、站、组件

系统由标准的电子介面组成，能实现在VAN或CAN上通讯。



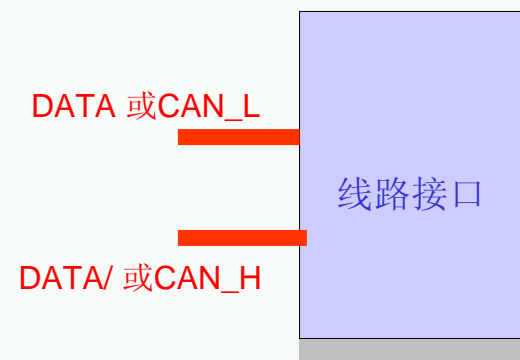
协议

概念:

➤ 各部件的作用

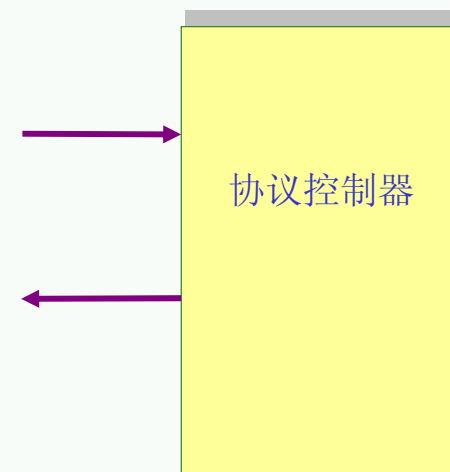
• 线路接口

它实现了协议控制器与总线之间的连接，它管理网络唤醒与休眠功能，过滤干扰、过高电压...



• 协议控制器

它根据网络协议接收并且传输应用数据



协议

概念：

➤ 线路接口 特性

- 接收阶段

一个CAN高速网的比较器和三个VAN或CAN低速网的比较器

- (TX)发送阶段

1 阶段；DATA 和 DATA/ 对于 VAN，
CAN L 和 CAN H 对于 CAN

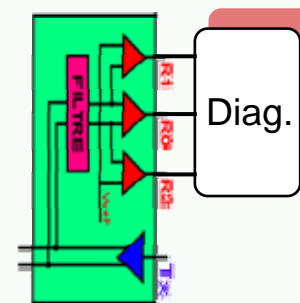
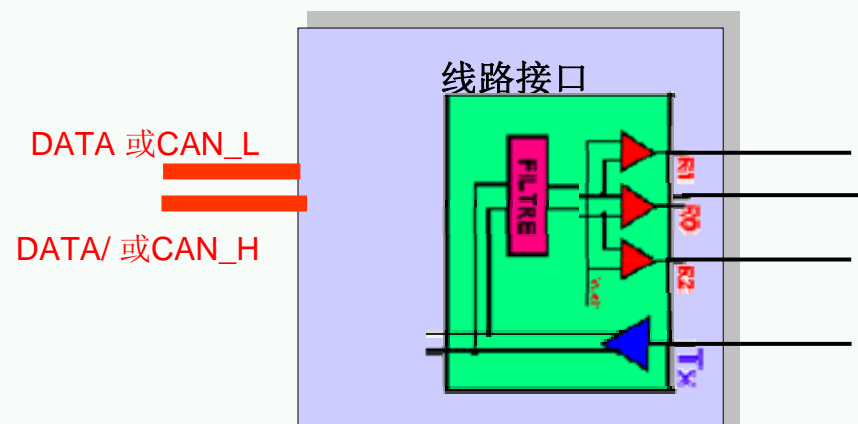
- 线路诊断

1 个用于CAN LS 低速网上的一个处理单元

在VAN协议控制器内集成此功能

- 休眠、唤醒功能

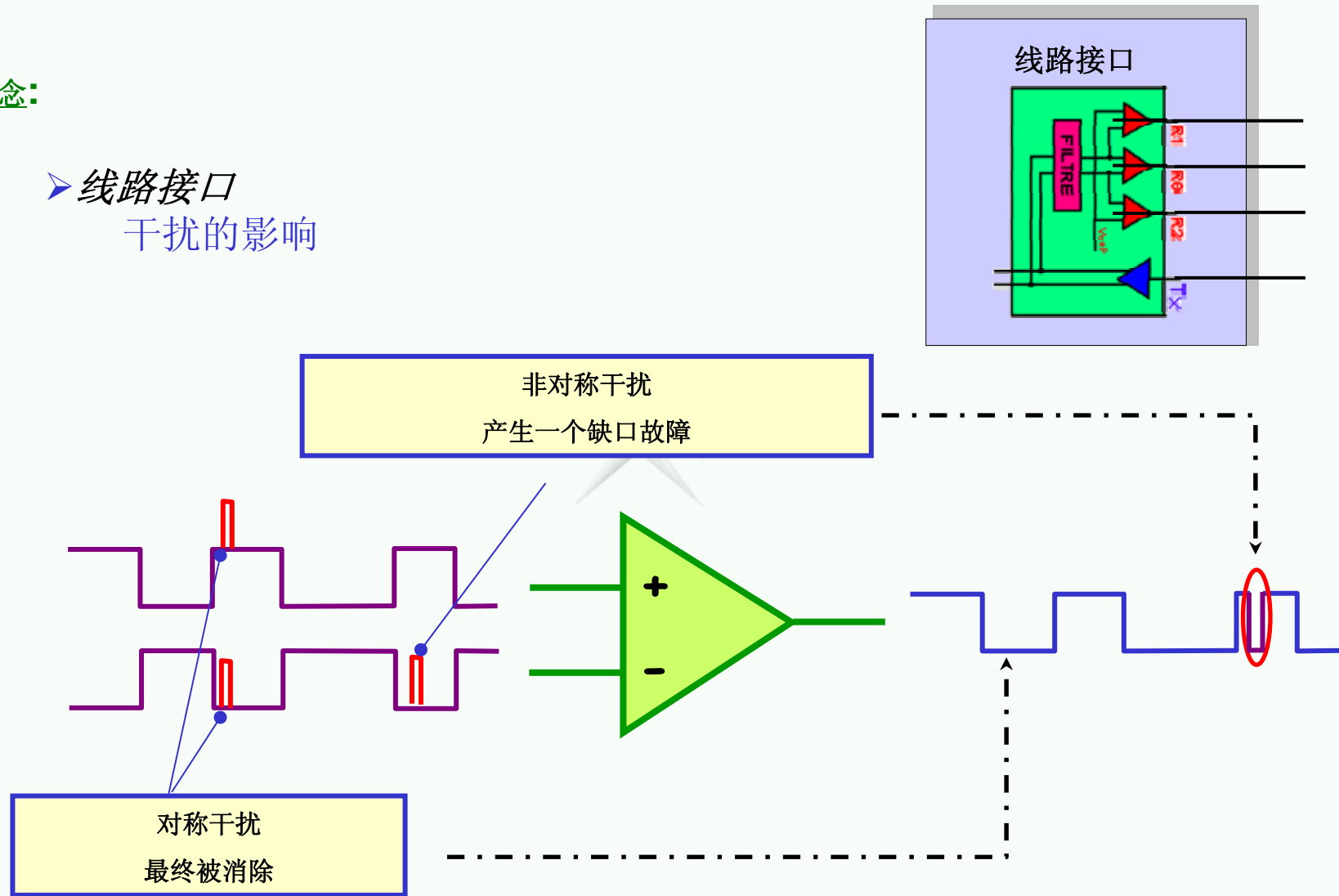
根据CAN低速网的网络活动，VAN网当前DATA/的消耗



协议

概念:

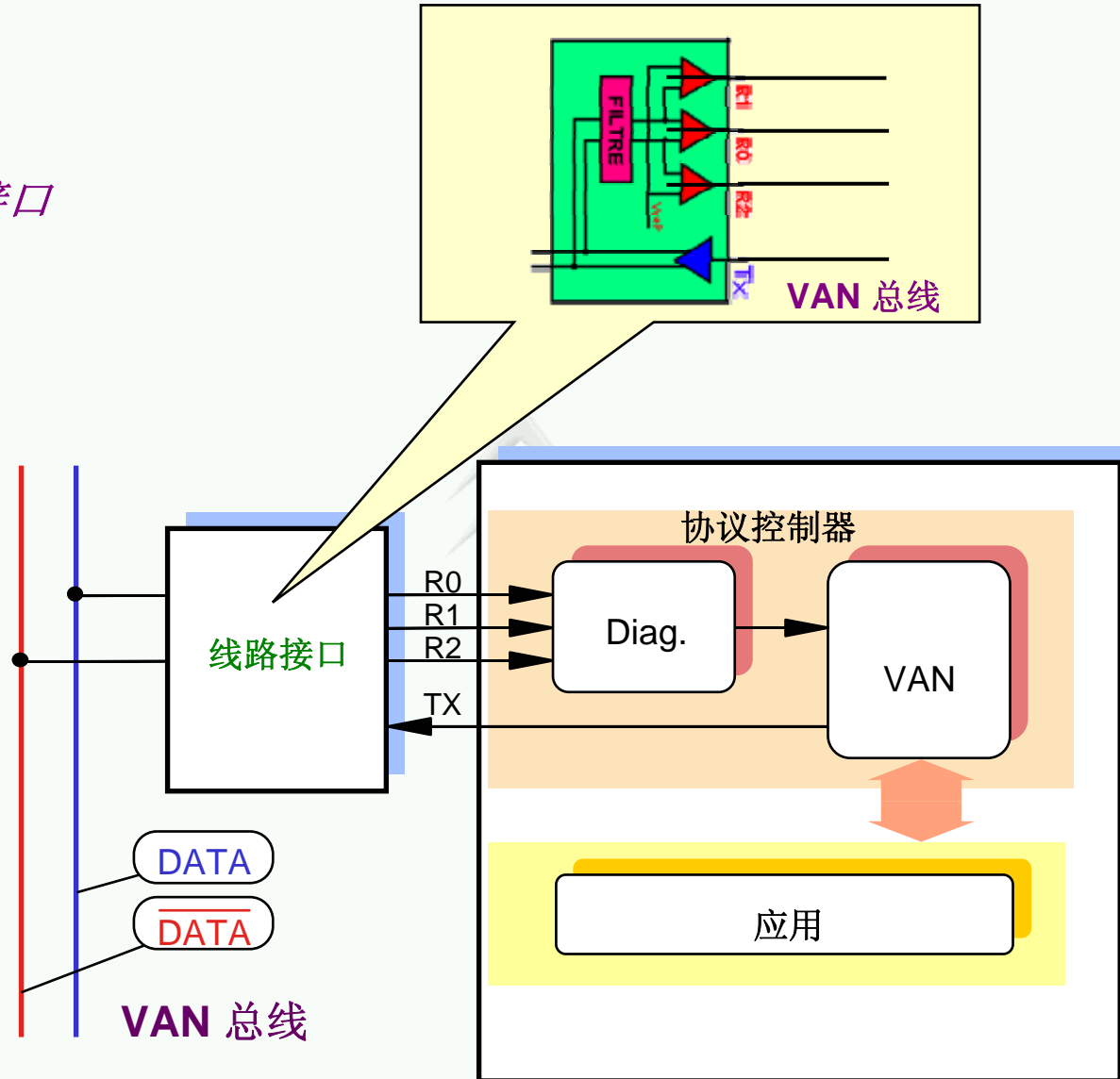
➤ 线路接口
干扰的影响



协议

概念:

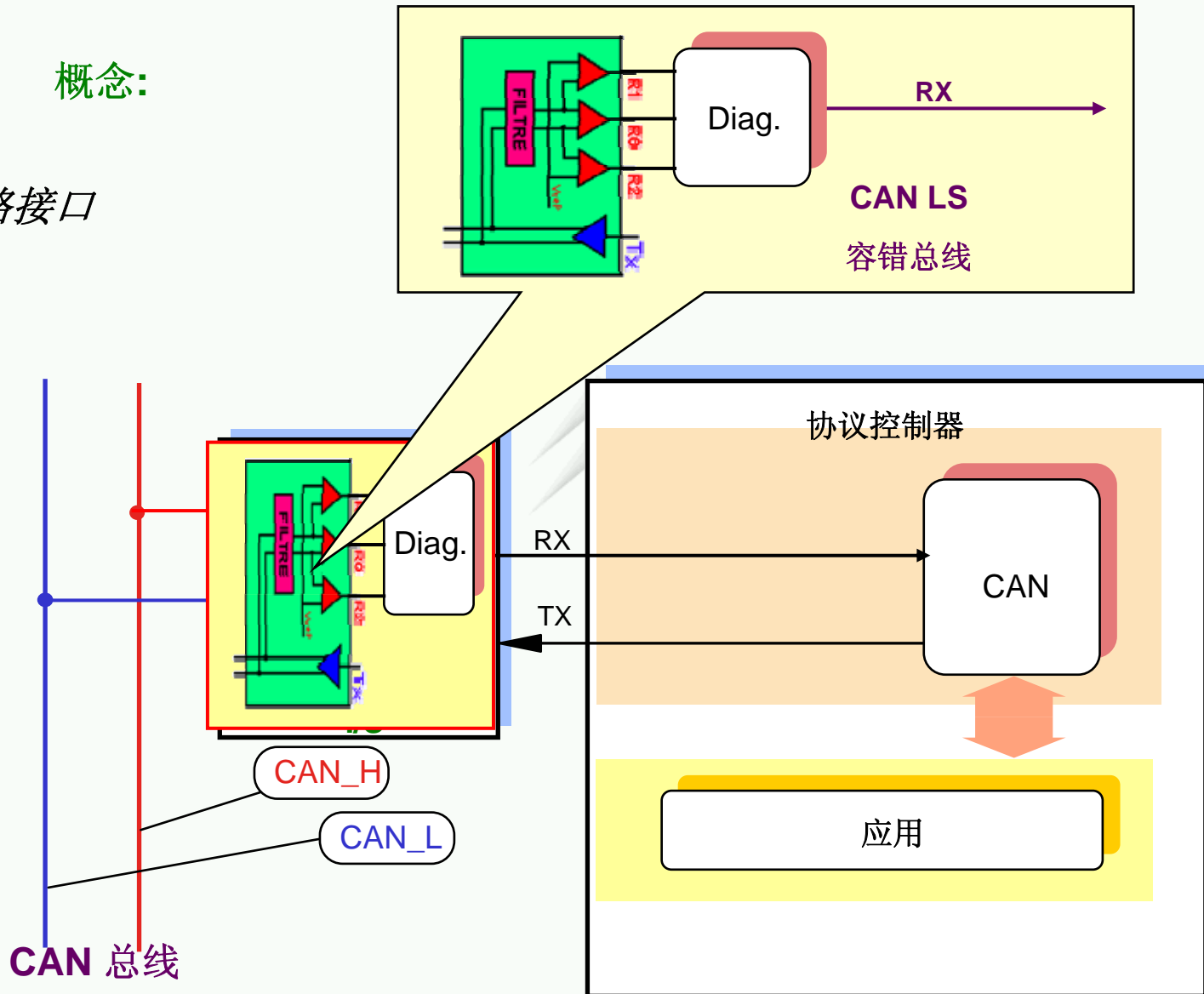
VAN 线路接口



协议

概念:

CAN 线路接口



协议

多路传输

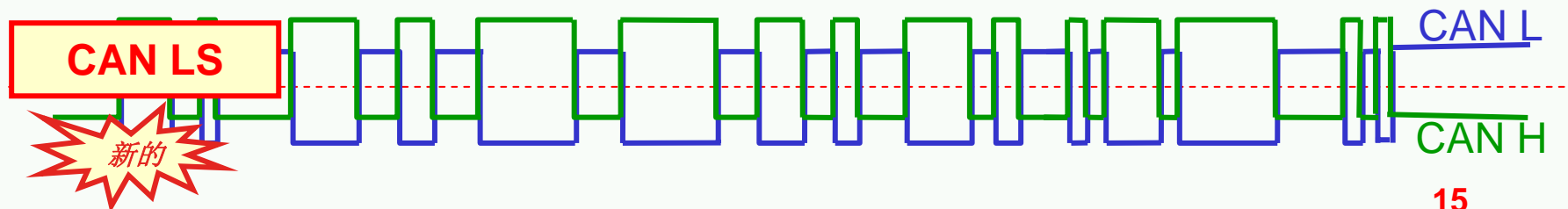
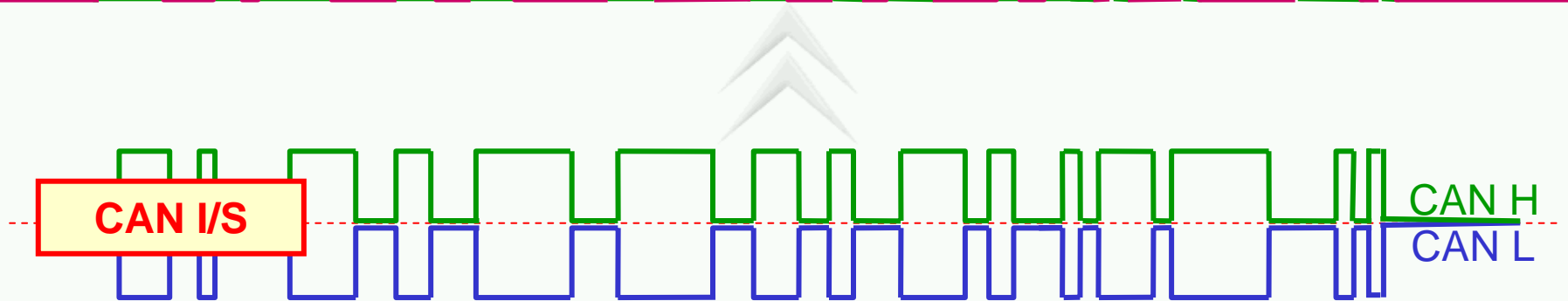
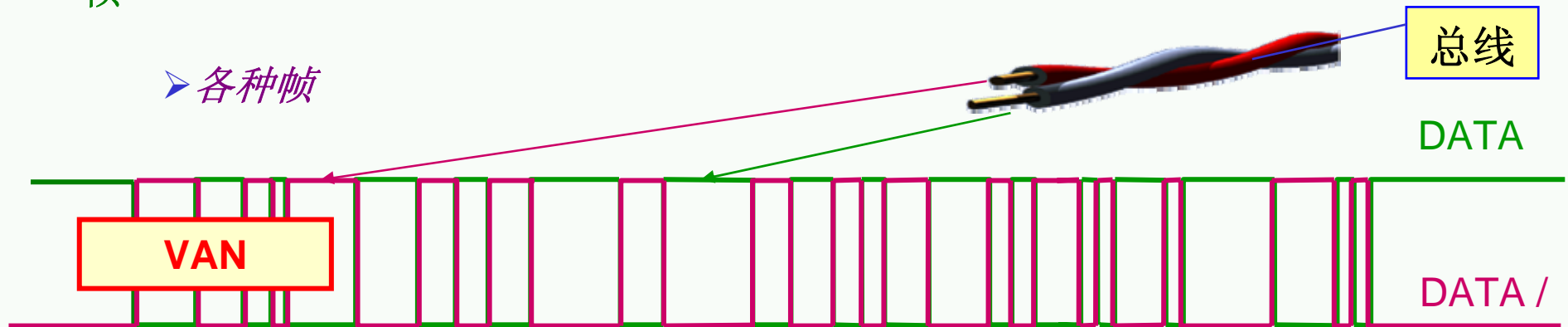


VAN 和 CAN
的通讯帧

协议

帧

➤ 各种帧



协议

帧

➤ 帧的构成

一个VAN 帧 9个片段



1 帧开始

2 识别

3 控制

4 数据

5 信息有效检查

6 数据结束

7 确认

8 帧结束

帧间隔

协议

帧

► 帧的构成:
一个CAN 帧 9个片段



1 帧开始

2 识别

3 控制

4 数据

5 信息有效检查

6 分隔符

7 确认

8 帧结束

帧间隔

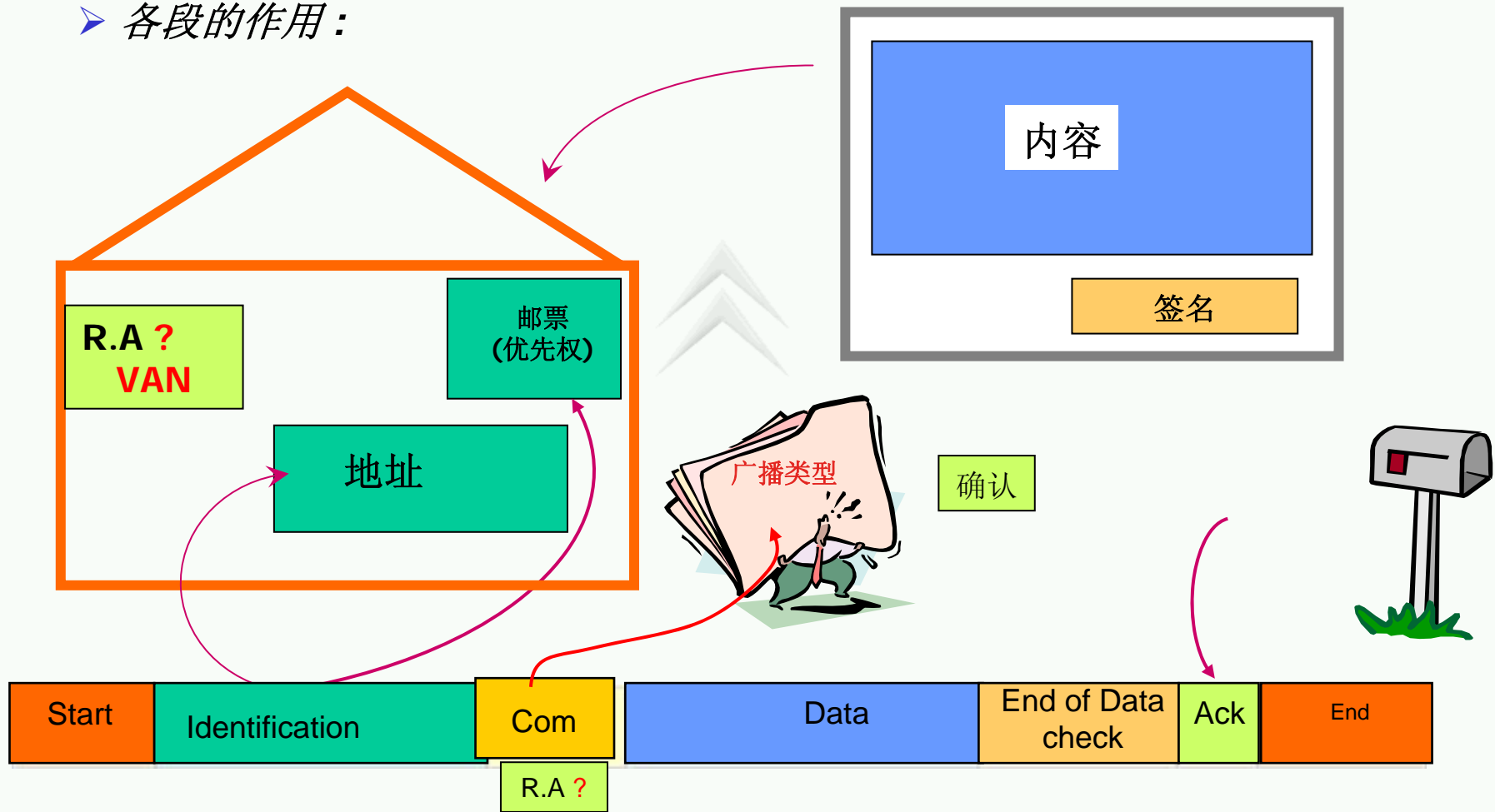
协议

帧：

用一封信来比喻



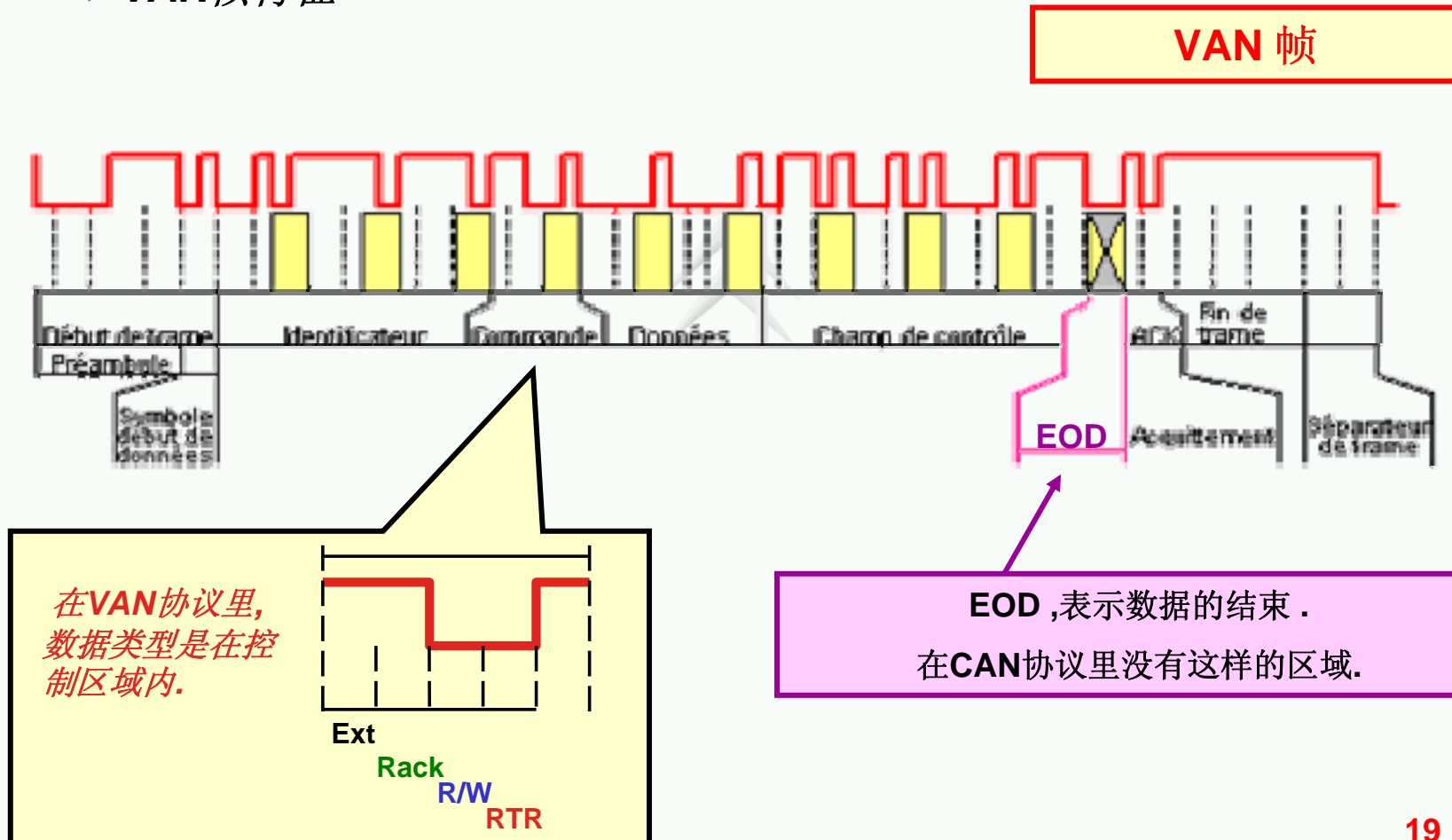
➤ 各段的作用：



协议

帧：

➤ VAN帧特征

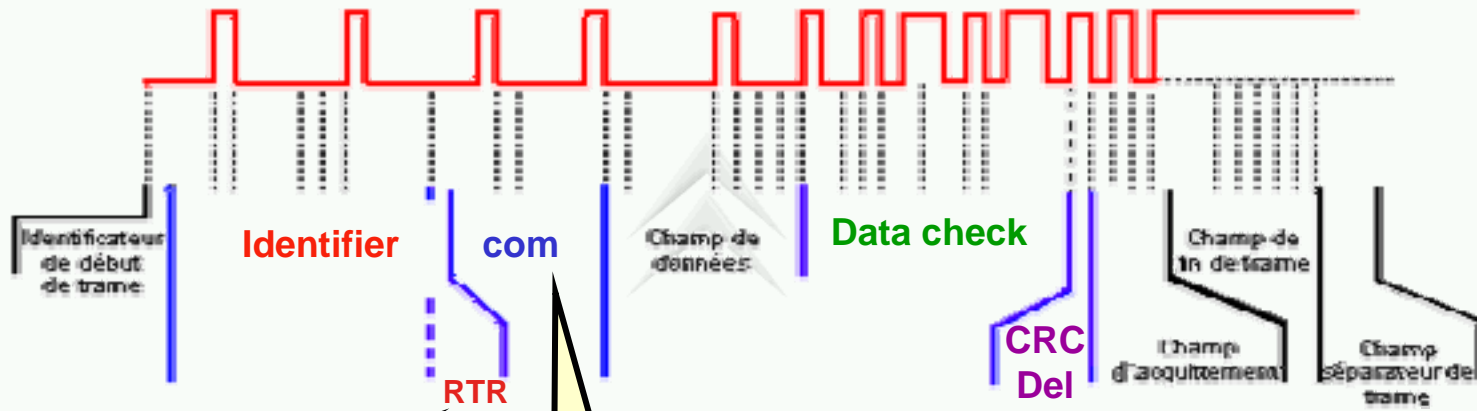


协议

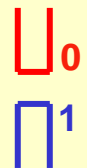
帧：

➤ CAN 帧特性：

CAN 帧



住处类型在数据识别里
request



控制决定数据长度

CRC Del,表示数据检测结束,用位1表示.

协议

帧：

➤ 通讯类型：

VAN

CAN

- 点对点
(带确认)

- 数据广播

- 点对点
- 协议里没有规定，但软件可以实现
- 数据广播

协议

帧：

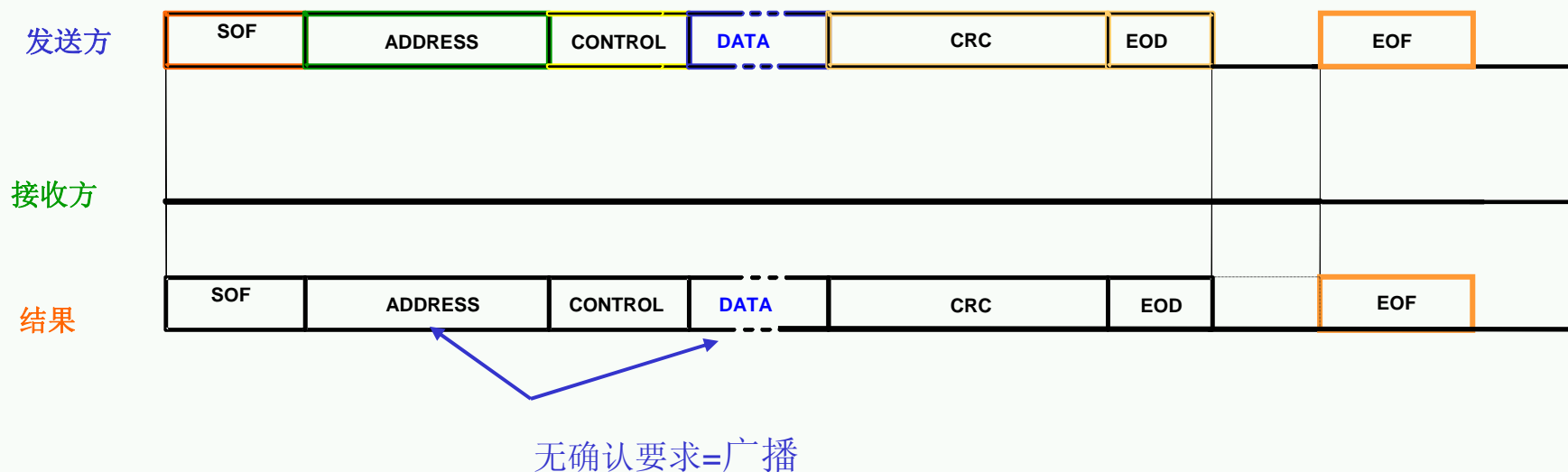
➤ 通讯方式

VAN

• 广播方式

没有帧确认要求

地址指明识别数据的特性（没有标明目的地）



协议

帧：

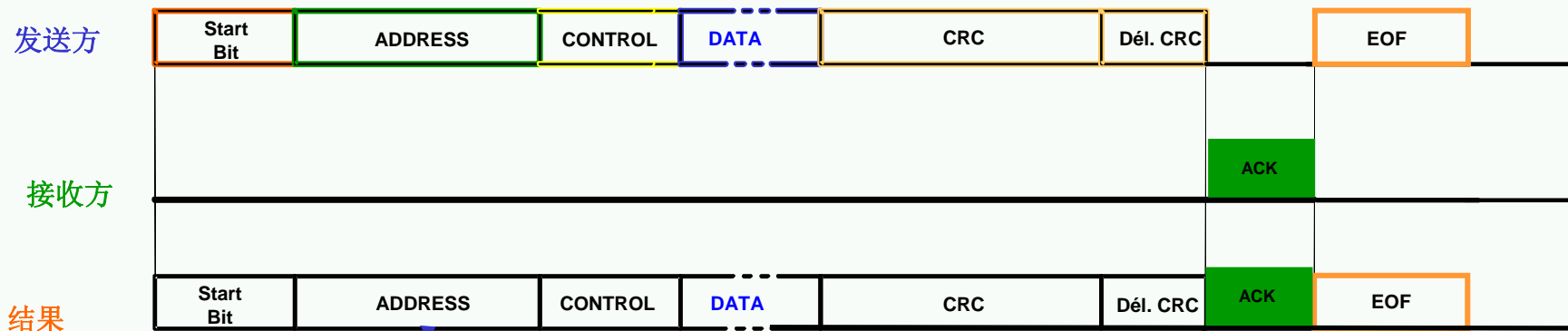
➤ 通讯类型：

CAN

- 广播

帧内确认

地址指明数据特性



用CAN协议不可能禁止确认

协议

帧：

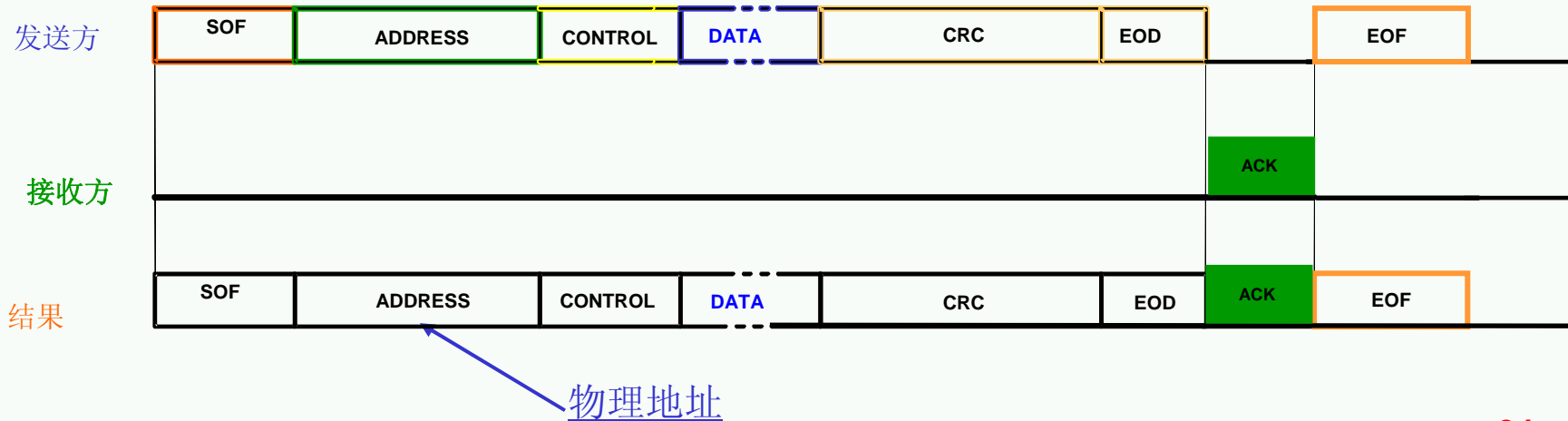
➤ 通讯类型：

VAN

• 点对点

帧内确认

物理地址指明ECU确认数据



协议

帧：

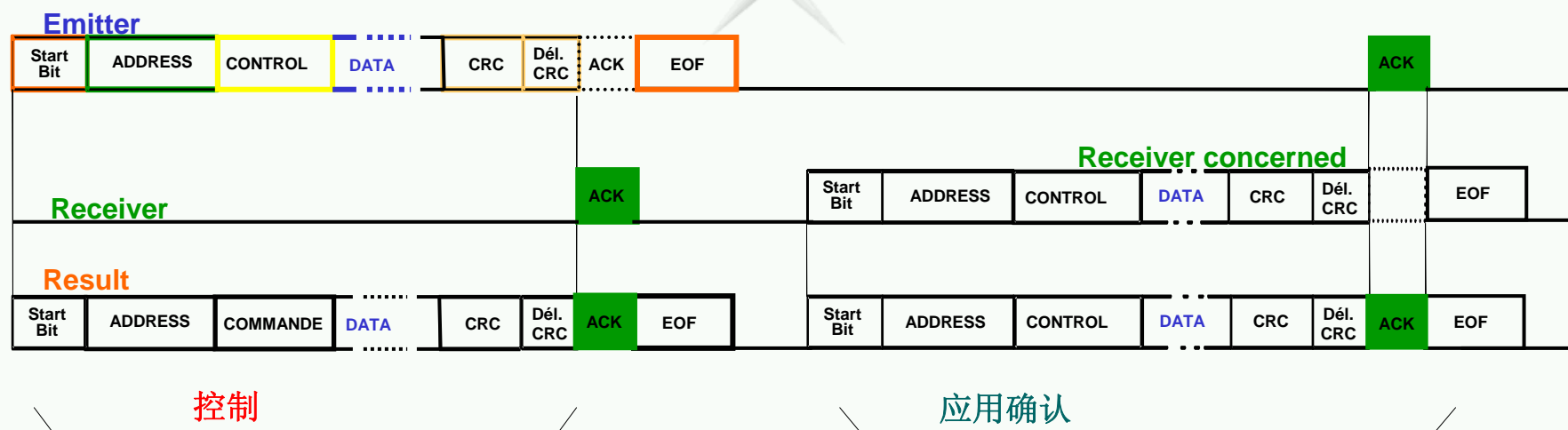
CAN

➤ 通讯类型：

- 写：点对点模式

帧内确认

通过发送一个确认信息来应用确认



协议

帧：

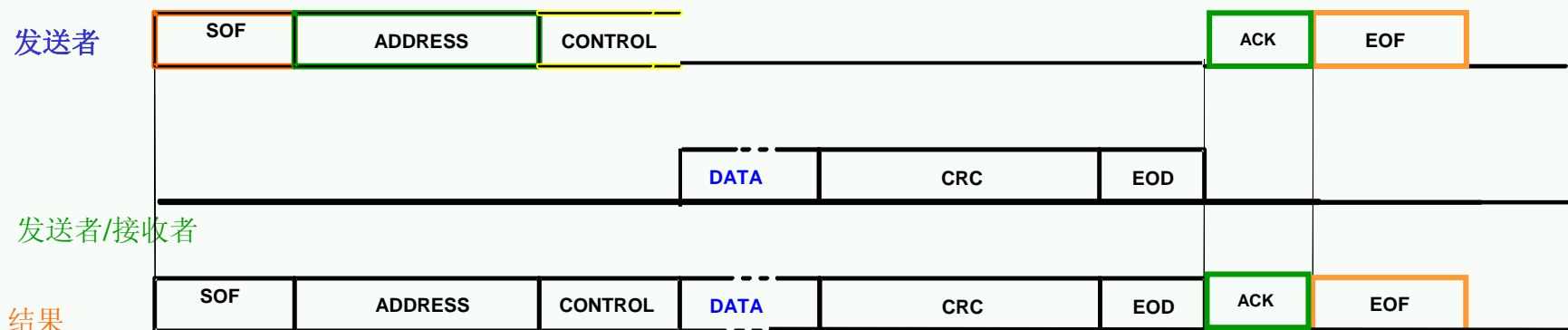
VAN

➤ 通讯类型：

•写 点对点且全案在帧内

要求直接答复

(IFR：帧内答复)



多路传输



数字格式

1001100111110001000000011010011010110001100010110001101101011110101101100110101000101

1001100111110001000000011010011010110001100010110001101101011110101101100110101000101

1001100111110001000000011010011010110001100010110001101101011110101101100110101000101

1001100111110001000000011010011010110001100010110001101101011110101101100110101000101

协议

数据格式

➤ 原理

➤ 用数据语言进行通讯

➤ 数据语言的作用是针对ECU和电子元件将一个物理值转换为一个数据

➤ 系统使用：

- ✓ Decimal numbering 十进制数
- ✓ Binary numbering 二进制数
- ✓ Hexadecimal numbering 十六进制数

协议

数字格式：

➤ *Decimal numbering* 十进制数:

➤ 在日常生活中使用

数字是 ; 0, 1, 2, ..., 9

➤ 可以通过乘10来进行计算

$$\begin{aligned} \text{例 : } 2624 &= (2 \times 10^3) + (6 \times 10^2) + (2 \times 10^1) + (4 \times 10^0) \\ &= 2000 + 600 + 20 + 4 \end{aligned}$$

协议

Digital format :

- **Binary numbering** :二进制数

用于电子元件是计算机

- 数基是 2 (2种可能的状态 « 0 » 或« 1 »)
- an elementary binary data is called a « Bit » Binary Digit一个基本的二进制数称为一个位的二进制数

总线上的所有数据是二进制数

1001100111110001000000011010011010110001100010110001101101011110101101100110101000101

协议

数字格式：

➤ 十六进制

它可以压缩二进制表达

它的数基是 16

包括：0,1,..., 9, A, B, C, D, E et F

十六进制只用于简化数据处理

协议

数字格式：

➤ 转换：

十六进制计数的优点

十进制 **100**

二进制 **0110 0100**

十六进制

 6 4

Note : 8 位等于 1 个字节

{01000101}

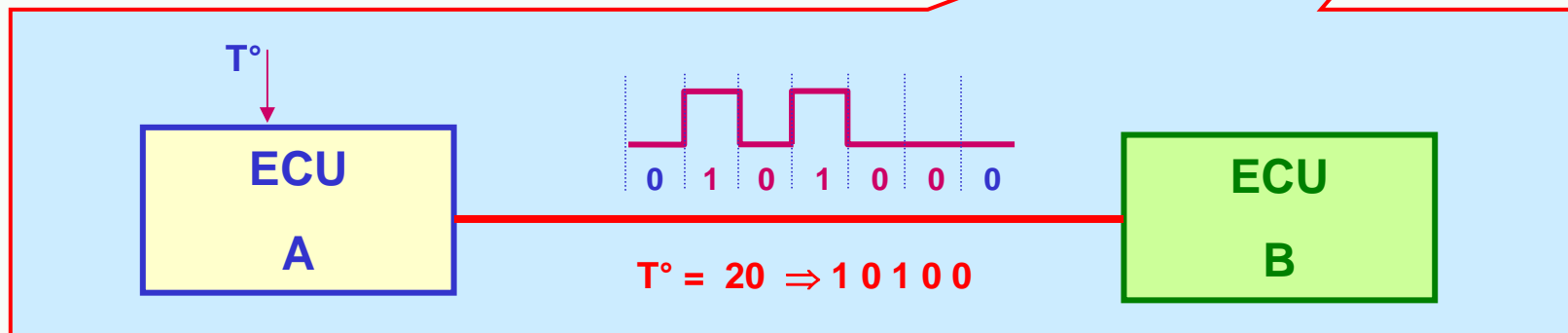
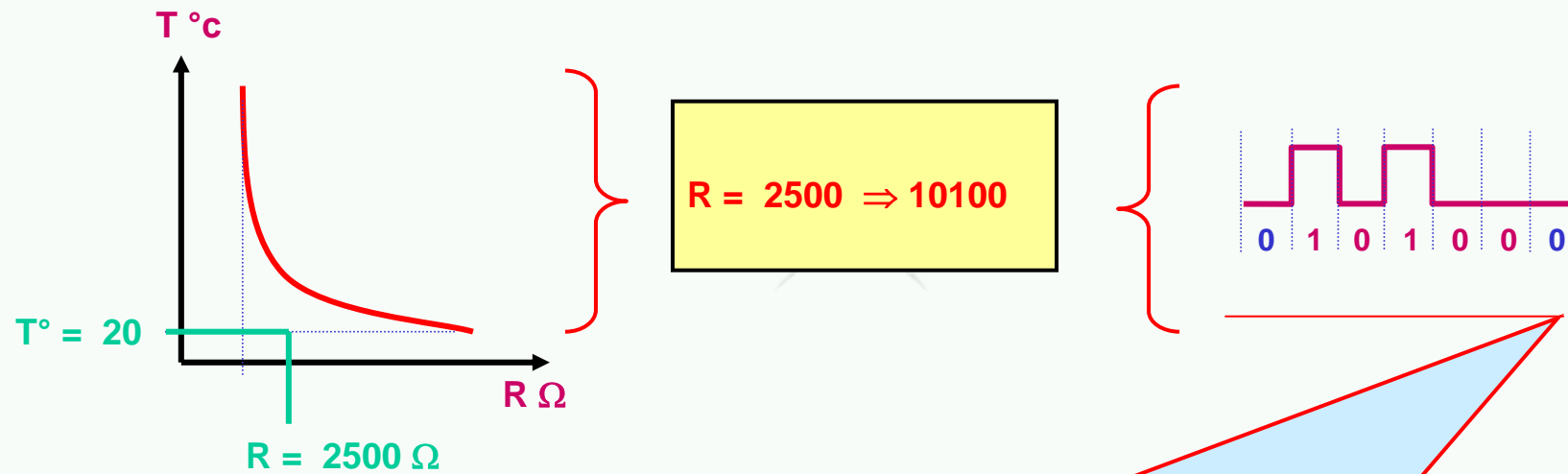
| 十进制 | 二进制 | 十六进制 |
|----------|-------------|----------|
| 0 | 0000 | 0 |
| 1 | 0001 | 1 |
| 2 | 0010 | 2 |
| 3 | 0011 | 3 |
| 4 | 0100 | 4 |
| 5 | 0101 | 5 |
| 6 | 0110 | 6 |
| 7 | 0111 | 7 |
| 8 | 1000 | 8 |
| 9 | 1001 | 9 |
| 10 | 1010 | A |
| 11 | 1011 | B |
| 12 | 1100 | C |
| 13 | 1101 | D |
| 14 | 1110 | E |
| 15 | 1111 | F |

Ex : conversion

协议

数字格式：

➤ 一个负温度系数NTC传感器的例子：

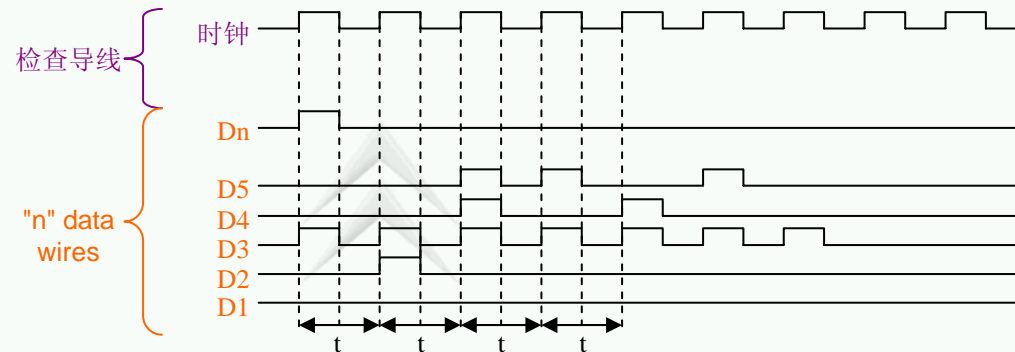


协议

数字格式：

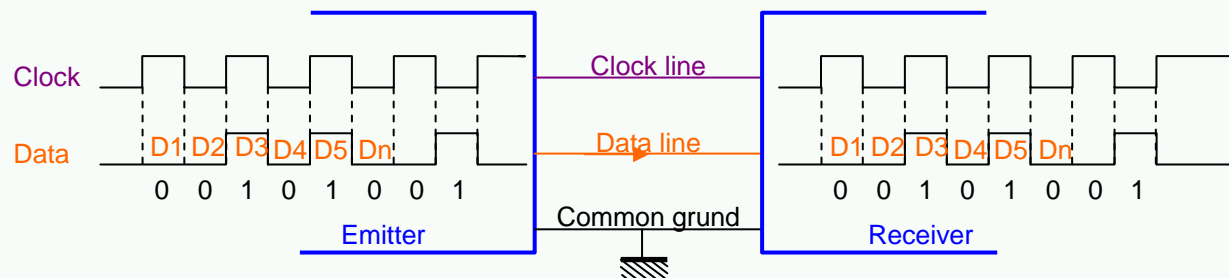
➤ 并行：

每根线传送二进制数据



➤ 串行：

一个特殊的设置可以用来在一根线上传输数据

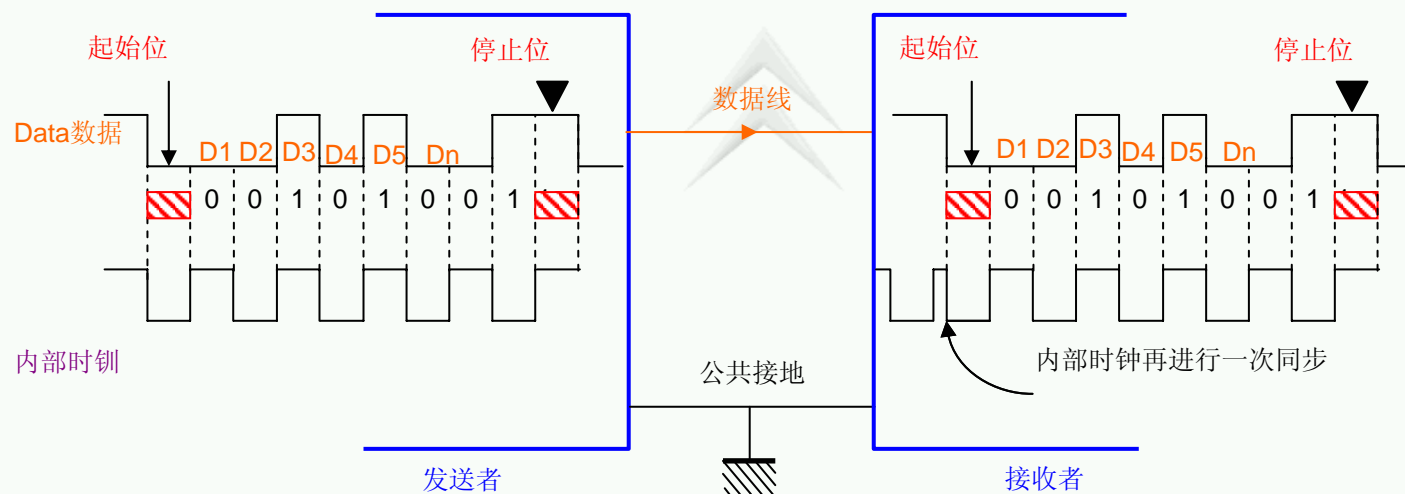


协议

数字格式：

用于 PSA

➤ 串行,带内置时钟 接由器时钟与带一个开始位的信息同步.



Note : 为了抗干扰而且使系统安全,数据线由两根线组成

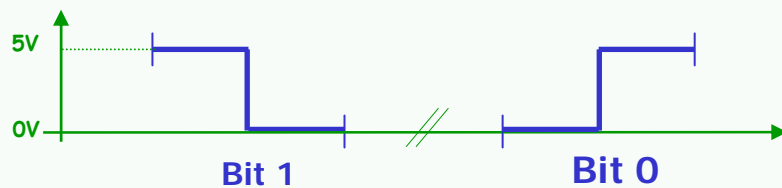
协议

数字格式：

➤ 解码方式：

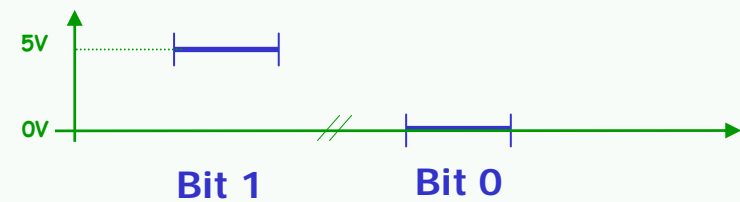
MANCHESTER 码

由 2 个互补的时间元素组成一位 (time-slot)
2 Time-Slots = 1 Bit



NRZ 码

一个时间元素代表一位 (Time-Slot)
1 Time-Slot = 1 Bit



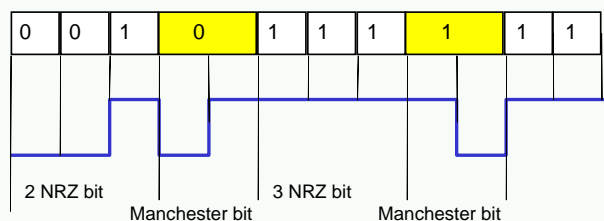
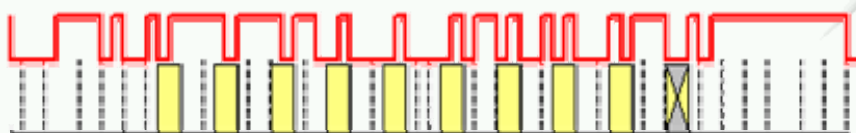
协议

数据格式：

➤ 编码方式：

E-MANCHESTER码

通过一种编码方式实现同步化
(3 NRZ 后跟一个 MAN)

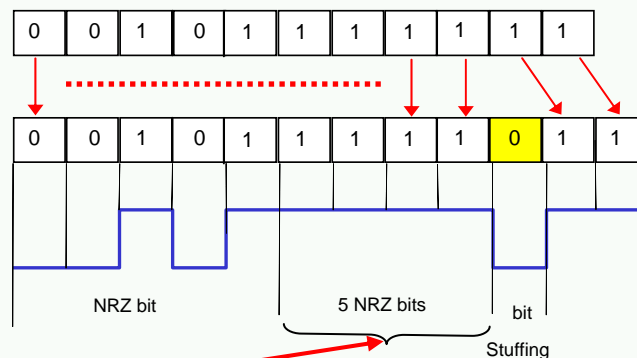


系统再同步

VAN

NRZ 由位填满

同步化是由一个非系统编码方式进行的 « 位填满 », activated following detection of 5 consecutive bits of the same level.



非系统再同步

CAN