

One-Wire

[1. Introduction](#)

[2. 硬件结构](#)

[3. 命令序列](#)

[初始化](#)

[ROM 命令](#)

[功能命令](#)

[3. 信号方式](#)

[初始化序列：复位和应答脉冲](#)

[读/写时序](#)

[Example: ROM 搜索实例](#)

FROM: <https://zhuanlan.zhihu.com/p/223031548>

1. Introduction

One-wire是Maxim全资子公司Dallas的一项专有技术

- 采用单根信号线，既传输时钟，又传输数据，数据传输是双向的
- 适用于单个主机系统，能够控制一个或多个从机设备
- 节省I/O口线资源、结构简单、成本低廉、便于总线扩展和维护

2. 硬件结构

单总线的闲置状态为高电平

3. 命令序列

- 第一步：初始化
- 第二步：ROM命令（跟随需要交换的数据）
- 第三步：功能命令（跟随需要交换的数据）

必须严格遵守这个命令序列

初始化

基于单总线上的所有传输过程都是以**初始化**开始的，初始化过程由**主机发出的复位脉冲**和**从机响应的应答脉冲**组成。应答脉冲使主机知道，**总线上有从机设备**，且准备就绪

ROM 命令

在主机检测到应答脉冲后，就可以发出 **ROM 命令**。这些命令与各个从机设备的唯一64位 ROM代码相关

```

graph TD
    subgraph InitSeq [初始化序列]
        S1[主机发送复位脉冲] --> S2[从机发送应答脉冲]
    end
    S2 --> S3[主机发送ROM命令]
    S3 --> D1{33h 读ROM命令}
    S3 --> D2{66h 写ROM命令}
    S3 --> D3{F0h 搜索ROM命令}
    S3 --> D4{ECh 报警搜索命令}
    S3 --> D5{CCh 跳越ROM命令}

    D1 -- Y --> S4[从机发送产品代码一个字节]
    S4 --> S5[从机发送系列码6个字节]
    S5 --> S6[从机发送CRC字节]
    S6 --> End[主机发送功能命令图3]

    D2 -- Y --> S7[主机发送第0位]
    S7 --> D6{第0位匹配?}
    D6 -- Y --> S8[主机发送第1位]
    S8 --> D7{第1位匹配?}
    D7 -- Y --> S9[主机发送第63位]
    S9 --> D8{第63位匹配?}
    D8 -- Y --> End

    D3 -- Y --> S10[从机发送第0位]
    S10 --> S11[从机发送第0位反码]
    S11 --> S12[主机发送第0位]
    S12 --> D9{第0位匹配?}
    D9 -- Y --> S13[从机发送第1位]
    S13 --> S14[从机发送第1位反码]
    S14 --> S15[主机发送第1位]
    S15 --> D10{第1位匹配?}
    D10 -- Y --> S16[从机发送第63位]
    S16 --> S17[从机发送第63位反码]
    S17 --> S18[主机发送第63位]
    S18 --> D11{第63位匹配?}
    D11 -- Y --> End

    D4 -- Y --> S19[从机发送第0位]
    S19 --> S20[从机发送第0位反码]
    S20 --> S21[主机发送第0位]
    S21 --> D12{从机设置了报警标志?}
    D12 -- Y --> End
    D12 -- N --> End

    D5 -- Y --> End
    
```

- 主机通过重复执行搜索 ROM 循环（搜索ROM命令跟随着位数据交换），以找出总线上所有的从机设备。
- 在每次执行完搜索 ROM 循环后，主机必须返回至命令序列的第一步（初始化）。

该命令仅适用于总线上只有一个从机设备。它允许主机直接读出从机的64位 ROM 代码，无须执行搜索 ROM 过程

匹配ROM [55h]

匹配 ROM 命令跟随64位 ROM 代码，从而允许主机访问多节点系统中某个指定的从机设备。仅当从机完全匹配64位 ROM 代码时，才会响应主机随后发出的功能命令

跳跃ROM [CCh] (仅适用于单节点)

主机能够采用该命令同时访问总线上的所有从机设备，而无须发出任何 ROM 代码信息

报警搜索 [ECh]

除那些设置了报警标志的从机响应外，该命令的工作方式完全等同于搜索 ROM 命令。该命令允许主机设备判断那些从机设备发生了报警

功能命令

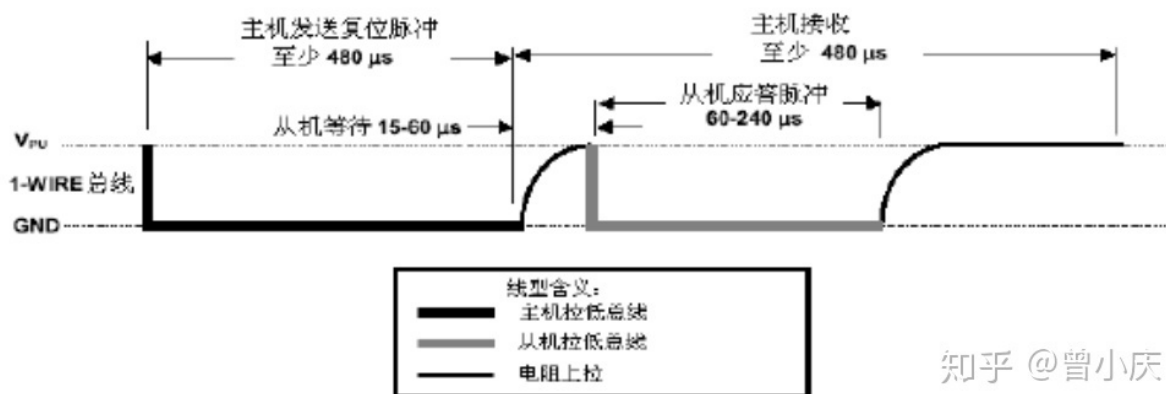
通过发出复位脉冲，主机能够在任何时候中断数据传输

3. 信号方式

所有的单总线器件要求采用严格的通信协议，以保证数据的完整性。该协议定义了几种信号类型：**复位脉冲**、**应答脉冲**、**写0**、**写1**、**读0**和**读1**。所有这些信号，除了应答脉冲以外，都由主机发出同步信号。并且发送所有的命令和数据都是字节的**低位在前**

初始化序列：复位和应答脉冲

单总线上的所有通信都是以初始化序列开始，包括：主机发出的复位脉冲及从机的应答脉冲



知乎 @曾小庆

1. 主机通过拉低单总线至少480 μ s，以产生 (Tx) 复位脉冲
2. 接着，主机释放总线，并进入接收模式 (Rx)。当总线被释放后，5k上拉电阻将单总线拉高
3. 在单总线器件检测到上升沿后，延时15-60 μ s，接着通过拉低总线60-240 μ s，以产生应答脉冲。

读/写时隙

在写时隙期间，主机向单总线器件写入数据；而在读时隙期间，主机读入来自从机的数据。在**每一个时隙**，总线只能传输一位数据

写时隙

写时隙：“写1”和“写0”

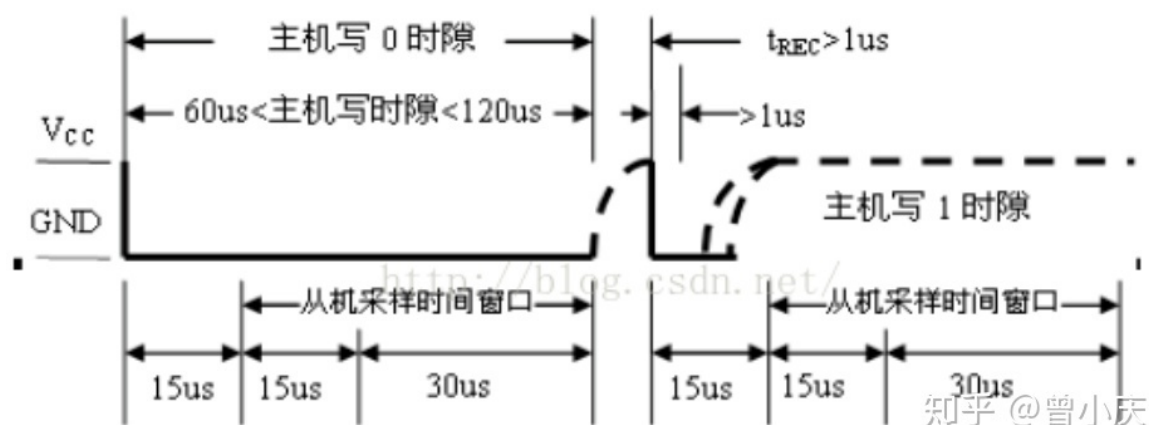
- 两次独立的写时隙之间至少需要1us的恢复时间
- 在写时隙起始后15-60us期间，单总线器件采样总线电平状态。如果在此期间采样为高电平，则逻辑1被写入该器件；如果为0，则写入逻辑0。

产生写1时隙

产生写1时隙的方式：主机在拉低总线后，接着必须在15us之内释放总线，由5k上拉电阻将总线拉至高电平

产生写0时隙

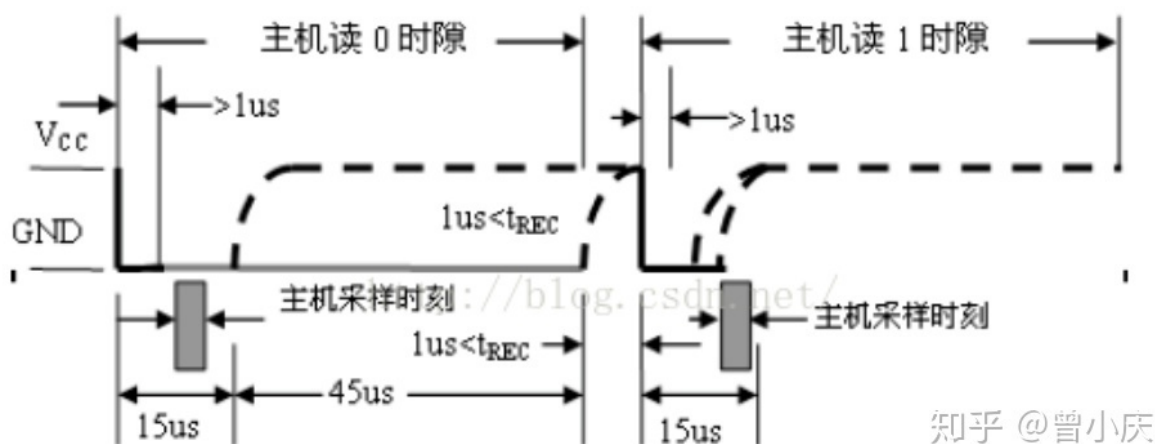
产生写0时隙的方式：在主机拉低总线后，只需在整个时隙期间保持低电平即可（至少60us）。



读时隙

仅在主机发出读时隙时，才向主机传输数据。所以，在主机发出读数据命令后，必须马上产生读时隙，以便从机能够传输数据

- 所有读时隙至少需要60us
- 每个读时隙都由主机发起，至少拉低总线1us
- 拉低总线1us（图6所示）。在主机发起读时隙之后，单总线器件才开始在总线上发送0或1。



Example: ROM 搜索实例

| <https://zhuanlan.zhihu.com/p/223031548>

Too long, just read the link