

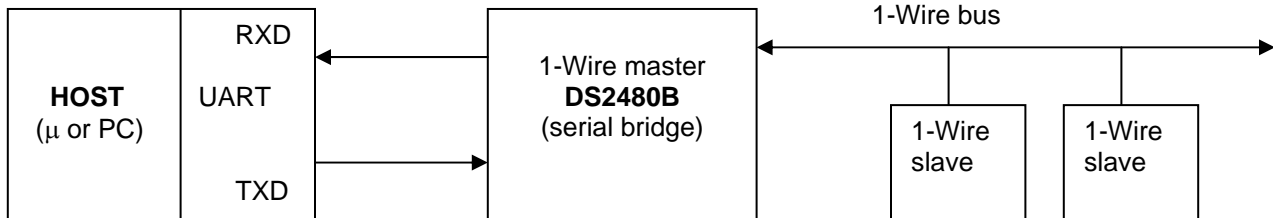
## 引言

1-Wire®通信协议可以通过微处理器上的一个 IO 引脚产生，不过，为了创建可靠的 1-Wire 网络，必须提供正确的时序和适当的输出电压摆率。1-Wire 主机发送的时序不正确会导致与 1-Wire 从器件之间的通信中断或完全失败。输出电压摆率若不加以控制，可能严重限制网络的长度，并产生时通时断的现象。如果可以提供串行通信 UART，那么通过一个串口至 1-Wire 桥接器(DS2480B)就可以解决上述问题。

DS2480B 是从串行接口到 1-Wire 网络协议转换的桥接器。只要主机具有普通的串行通信 UART，就可以通过该桥接器产生严格定时和电压摆率控制的 1-Wire 波形。DS2480B 接收所要发送的指令与数据，执行 1-Wire 操作，并将结果返回至主机。图 1 给出了 DS2480B 配置的简化框图。实现该协议并浏览所有的 DS2480B 指令可能需要花大量的时间并容易混淆。为此，本指南将介绍通用的 1-Wire 操作，并解释 DS2480B 的输入串行数据包结构，以及输出串行数据包的含义。

本文作为 DS2480B 数据资料的补充，并不能代替数据资料。该数据资料可以在网页 [http://www.maxim-ic.com.cn/quick\\_view2.cfm/qv\\_pk/2923](http://www.maxim-ic.com.cn/quick_view2.cfm/qv_pk/2923)中找到。

## DS2480B 的使用(简化) 图 1



与该桥接器一起工作的主机 UART 至少应能够支持 8 位、无奇偶校验、9600 波特(位/秒)的通信。该桥接器可以实现更高的数据速率，最高可以达到 115200 波特，但是上电时以 9600 波特启动。DS2480B 数据资料中介绍了电气方面的注意事项，如 RS232。

## 1-Wire 接口

只有在所有指令与模式均被转换到 1-Wire 通信接口，使得各种应用能够基于该接口使用，DS2480B 才是有用的。为了执行 1-Wire 操作，应用中必须具备的一些基本 1-Wire 函数。第一项操作是使总线上所有 1-Wire 从器件复位，为接收来自 1-Wire 主机的指令做好准备。第二项操作是从 1-Wire 主机向从器件执行写一位操作，第三项操作是从 1-Wire 从器件中读一位操作。由于必须由 1-Wire 主机来启动所有的 1-Wire 位通信，所以“读取”操作实际上是在“写入”一位后采样得到的结果。几乎所有其他 1-Wire 操作都可以由这三个操作构成。例如，向 1-Wire 总线写 1 个字节相当于 8 个一位写操作。1-Wire 搜索算法(参见应用笔记 187，位于 [http://www.maxim-ic.com.cn/appnotes.cfm/an\\_pk/187](http://www.maxim-ic.com.cn/appnotes.cfm/an_pk/187))也可以由上述基本操作构成。不过，这并不见得是效率最高的

实现方法。DS2480B 包含搜索加速器模式，可以大大降低执行搜索操作所需的串行通信量。效率更高的方法是将位通信捆绑为字节组，甚至将字节组捆绑为字节块。为了获得最高的效率，应用中应当尽可能使用最大的指令组(最大的数据包)。

表 1 是高效率 1-Wire 函数至少应具备的接口操作。操作名称以特定操作的标识提供，将在下面文中使用。

**基本的 1-Wire 操作 表 1**

操作	说明
OWReset	发送 1-Wire 复位激励，并检查 1-Wire 从器件的在线脉冲。
OWWriteBit / OWReadBit	发送或接收 1-Wire 总线上的 1 位数据。
OWWriteByte / OWReadByte	发送或接收 1-Wire 总线上的 1 字节数据。
OWBlock	向 1-Wire 总线发送或从总线接收多个字节的数据。
OWSearch	执行 1-Wire 搜索算法(参见应用笔记 187)。

在基本操作之外，还有扩展的 1-Wire 函数。一些 1-Wire 从器件可以工作在两种不同的通信速度下：标准和加速模式。所有器件都至少支持标准速度。加速模式大约是标准速度的 10 倍。DS2480B 同时支持这两种 1-Wire 速度。

1-Wire 器件通常从 1-Wire 总线上获得部分或全部的操作能量。不过有些器件在协议的特定位置需要额外的电源传输。例如，某器件可能需要进行温度转换或计算 SHA-1 散列。这时电源通过 1-Wire 总线上允许更强的上拉来提供。在这种电源传输方式中无法进行正常的通信。DS2480B 具有一些先进的特性，用来提供电源传输。

EPROM (一次性可编程) 1-Wire 存储器在进行写操作时需要专门的 12V 脉冲。如果 DS2480B 具有 12V，那么它将被要求向 1-Wire 总线传输一个脉冲，用于 EPROM 编程。

表 2 中列出了扩展的 1-Wire 操作，用于 1-Wire 速度设定、电源传输，以及编程脉冲。

**扩展的 1-Wire 操作 表 2**

操作	说明
OWSpeed	设定 1-Wire 通信速度。选项为标准速度或加速模式。需要注意的是，该操作仅改变 1-Wire 主机的通信速度，而对 1-Wire 从器件，必须通过指令，使其完成从标准速度到加速模式的切换。1-Wire 从器件在遇到标准速度的 1-Wire 复位时，总是还原到标准速度。
OWLevel	设定 1-Wire 电源电平(正常或电源传输)。
OWProgramPulse	为 EPROM 1-Wire 器件的写操作发送定时的 12V 编程脉冲。
OWReadBitPower	从 1-Wire 总线上读取 1 位数据，可以选择在该位读取结束后，立即进行电源传输。
OWWriteBytePower	向 1-Wire 总线发送 1 字节数据，在该字节发送结束后，立即进行电源传输。



本篇应用笔记中的‘C’代码工具，使用的是 Microsoft Windows 32 位操作系统 RS232 串口，以 DS9097U 适配器作为主机，该代码可以从以下链接中下载：

[http://files.maxim-ic.com/sia\\_bu/public/an192.zip](http://files.maxim-ic.com/sia_bu/public/an192.zip)

该 C 代码工具是 1-Wire Public Domain 套件提供的简化版本。1-Wire Public Domain 套件中还包含器件特殊的模块与实例，可以在以下链接中找到：

<http://www.maxim-ic.com.cn/products/ibutton/software/tmex/>。

## DS2480B 配置

在尝试 1-Wire 操作之前，主机必须初始化，并与 DS2480B 串行 1-Wire 线驱动器进行同步。检测到主机与桥接器之间的通信故障时，也需要执行该初始化与同步过程。在初始化过程中，DS2480B 要求 9600 波特率。初始化结束后，可以实现高达 115200 的波特率。不过，改变波特率时应当注意，因为 DS2480B 只有 1 字节的输入缓冲区。输入的 1-Wire 指令必须能够在下一条指令移入之前完成。关于各指令分别工作在多少波特率下，请参照 DS2480B 数据资料中的表 7。

## DS2480B\_Detect

由于 DS2480B 中没有晶振，它必须通过采样主机发送的串行通信来调节自身的时基。初始化的时序从复位 DS2480B 开始，然后发送一个预先定义好的定时字节。该器件的复位将使所有 1-Wire 配置参数复位到缺省状态。为了在小型到中等长度的 1-Wire 网络中获得良好的性能，在进行标准速度通信时，推荐在“伸缩”模式下使用 DS2480B。1-Wire 设置参数用来为伸缩模式下的 1-Wire 信号整形。因此，不管什么时候 DS2480B 被复位，配置参数就需要重装。理想的伸缩模式设置为  $PDSRC = 1.37V/\mu s$ ,  $WILD = 10\mu s$ ,  $DSO/WORT = 8\mu s$ 。该复位与配置时序被组合为一项操作，称为 *DS2480B\_Detect*。

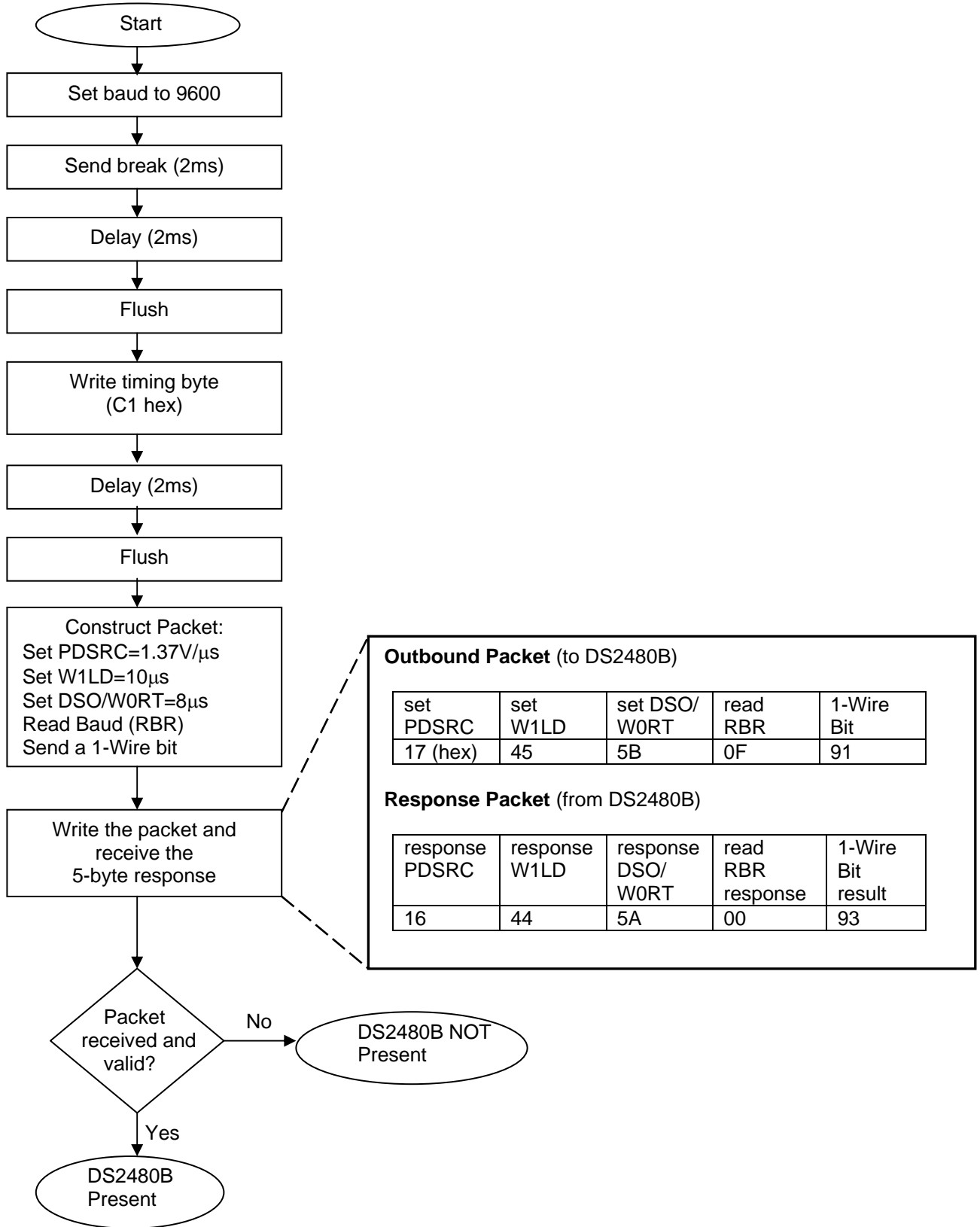
如果检测到停止位为空号，DS2480B 就被复位。产生该复位信号最简单的方法是，发送比 9600 波特 8 位的字长更长的串行暂停。如果主机 UART 上不提供暂停，那么切换到较低的波特率下，并发送一个 0 字节，可以模拟一次暂停。切换到空号奇偶校验位或改为最高有效位为 0 的 9 位字长也可以模拟一次暂停。

在配置时序中的一些延时值(参见图 2)可以任意大，以适应大多数 UARTS。这些值可以被减小。

在初始化时序结束之后执行 read baud rate register 和 write 1-Wire bit 操作，以检验 DS2480B 初始化功能的正确性。如果这些操作中任意一个的返回值为无效应答，则认定初始化过程失败。

需要注意的是，该执行过程并不检查 DS2480B 主动在线脉冲通报字节。当调用 *DS2480B\_Detect* 函数时，这可能引起某些 1-Wire 操作得到格式化不正确的应答字节。由于在一些接触不良的环境中也会产生一些异常的在线脉冲，每个 1-Wire 应用都应该具有重试功能，因此，这也解决了上述问题。

DS2480B\_DETECT 流程 图 2



## DS2480B\_ChangeBaud

为改变主机与 DS2480B 之间的通信速度，必须对 RBR (RS232 波特率)寄存器进行写操作。DS2480B 立即以新的波特率响应输出设定的波特率值，但是这很可能被主机错过。因此，图 3 推荐的流程是忽略响应字节。

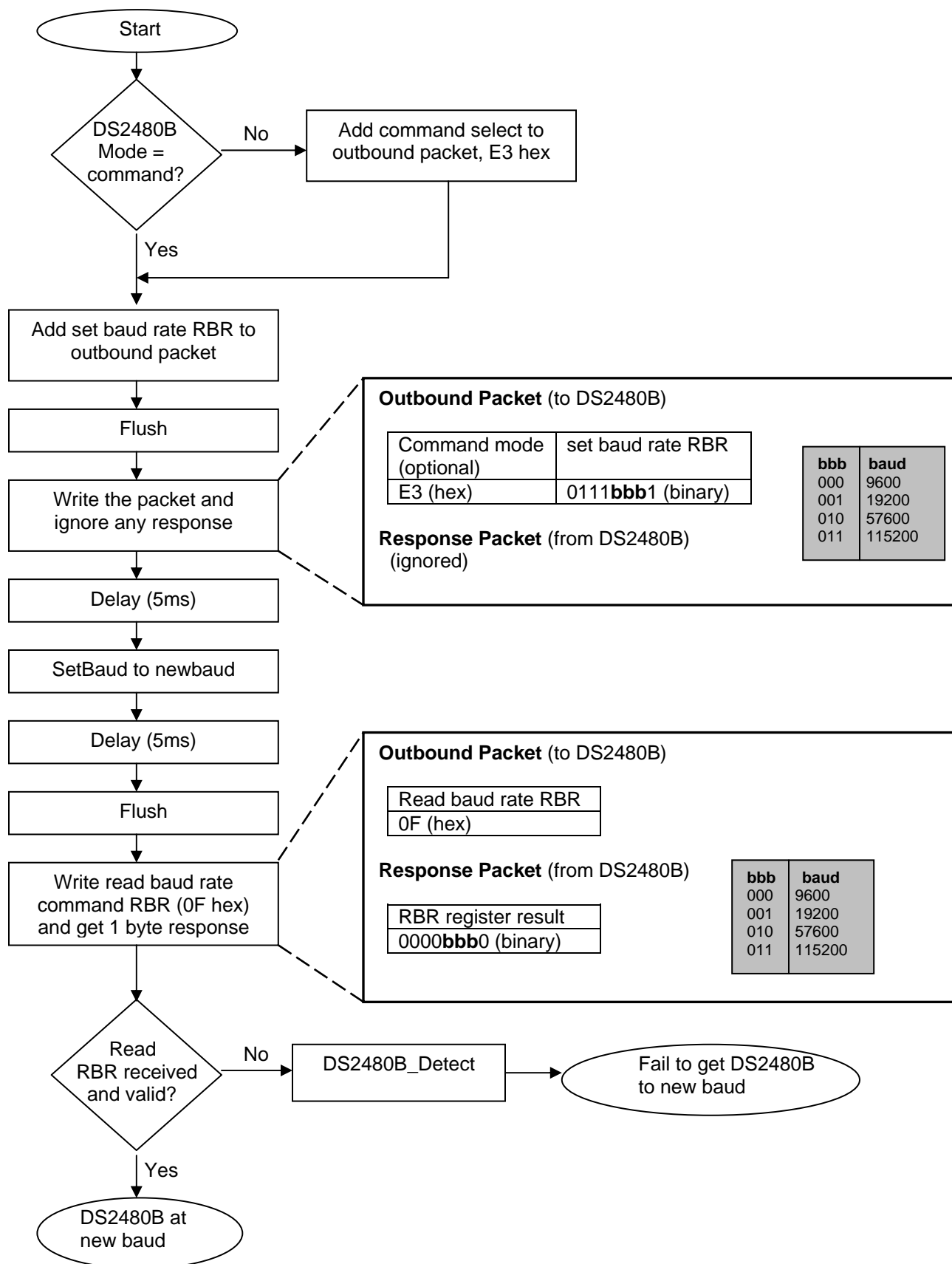
在主机与 DS2480B 都切换了波特率以后，波特率寄存器被读回，以确认操作完成。如果 DS2480B 没有工作在正确的波特率下，读回的操作将失败，并且将调用初始化时序 *DS2480B\_Detect*。

DS2480B 可以工作在四种不同的波特率下：9600、19200、57600 和 115200。由于 DS2480B 只有 1 字节的缓冲区，需要发送的指令必须在下一条指令到达前发送完毕。数据资料中的图 7 说明了每条指令可以在多少波特率下执行，而不会有覆盖指令的危险。建议总是在进入下一条指令之前接收 1-Wire 复位结果，但是，不要将复位操作包含在一个连续的字节流中。这样的执行过程将 1-Wire 复位分离成自身的 *OWReset* 操作，因而不再存在这个问题。类似地，单个位和单字节操作也可以分解成自身的操作：*OWReadBit*, *OWWriteBit*, *OWReadByte*, 以及 *OWWriteByte*。由于这些操作不会与其他指令连续，因此可以使用最高波特率。另外需要注意，在进行标准速度的 1-Wire 通信时，这样的执行过程只能使用带扩展位定时的伸缩模式，这将影响容许的波特率。推荐的波特率参见表 4。为了简化，执行过程仅使用两种波特率：标准速度(伸缩)操作中使用 9600 波特；非搜索的加速模式操作中使用 115200 波特。

最大数据流波特率 表 4

函数	标准 (伸缩模式)	加速模式
搜索(OWSearch)	9600 (波特)	57600
指令(所有非 1-Wire 操作)	115200	115200
数据(OWBlock)	9600	115200

## DS2480B\_CHANGEBAUD 流程 图 3



## 1-Wire 操作

基本的与扩展的 1-Wire 操作创建了通用的 1-Wire 接口，便于在任何 1-Wire 器件上实现任何操作。

这些 1-Wire 操作中每一个的执行过程都具有一些共同的特点。只要有可能，指令与数据就被组合到一起，以降低与 DS2480B 交换的数据包数量。DS2480B 的当前模式被作为状态保存，所以数据包可以从模式转换的指令开始。如果应答数据包的长度或者格式不正确，则将调用 *DS2480B\_Detect* 时序。

当改变 1-Wire 通信速度时，唯一改变主机与 DS2480B 之间串行通信速率的操作是在 *OWSpeed* 中完成的。执行完毕后，*OWSearch* 就不能在加速模式中运行了。当在更快的通信速率下执行搜索操作时，需要进行简单的波特率检查，以降低波特率至 57600。

每个 1-Wire 操作都必须首先确定当前的电平上拉是正常的。所以每个流程都从调用隐含的 *OWLevel(normal)* 开始。

### OWReset

*OWReset* 操作命令 DS2480B 向 1-Wire 总线发送一个复位脉冲并采样，以检测来自总线上 1-Wire 从器件的在线应答脉冲。尽管该指令的主要目的是实现复位操作，它也返回了其他有用信息。它还提供了 3 位字段用来说明芯片的版本。版本字段在所有的 DS2480B 中都是固定的，不过它可以用来检测该桥接器的前一个型号—DS2480。该执行过程与 DS2480 是兼容的。同样，将该字段屏蔽可以使主机软件或硬件与未来的桥接器版本至少是部分兼容的。

1-Wire 复位指令操作以通信速度执行。需要注意的是，在采用标准 1-Wire 速度通信时，该执行过程使用的是 DS2480B 伸缩模式。

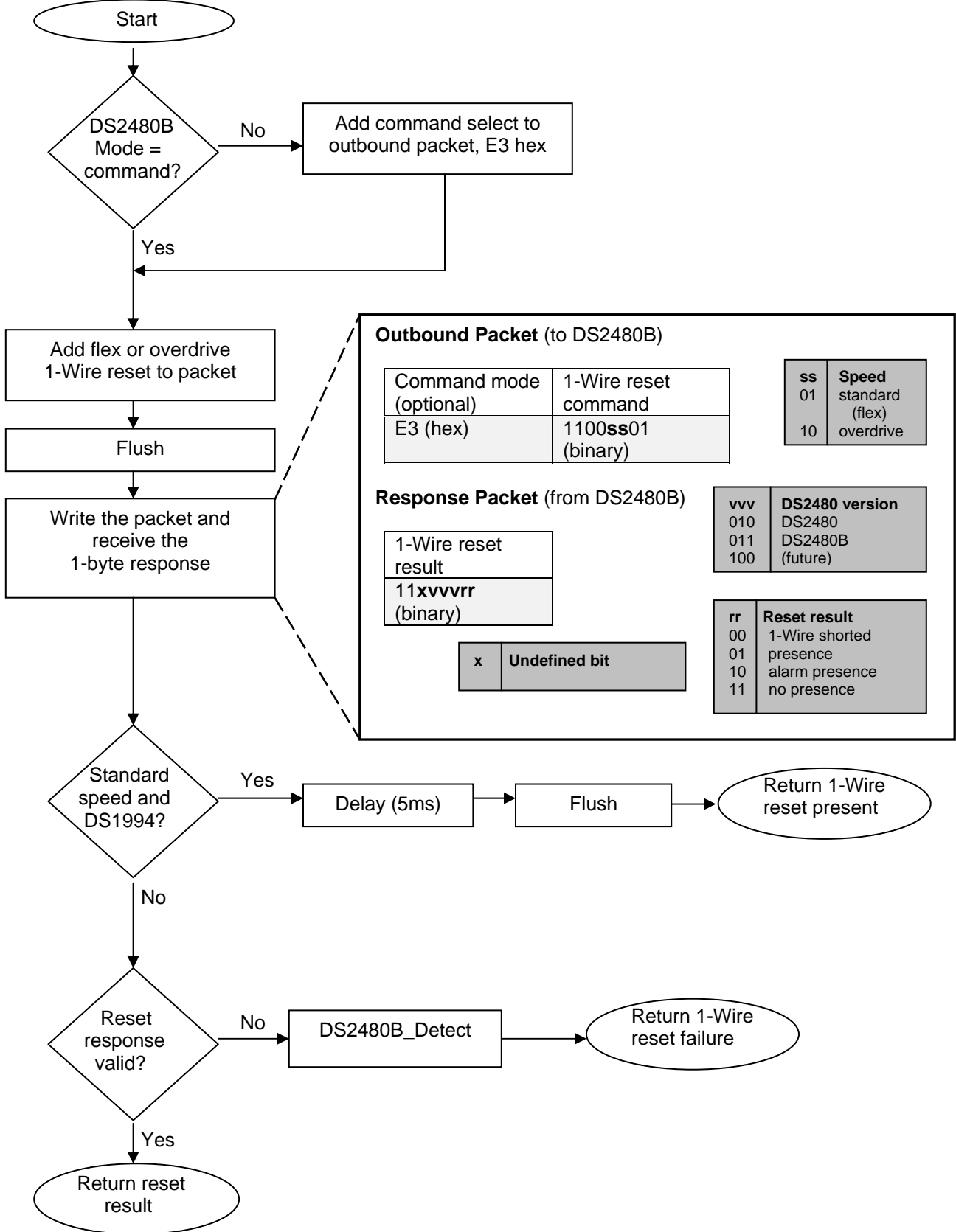
完成该操作所需的时间取决于是否有告警出现。这就是该操作没有与其他 1-Wire 操作一起组合到数据包中的主要原因。如果 DS1994\*或 DS2404\*可能在 1-Wire 总线上，则需要注意额外的 5ms 延时与刷新。DS2480B 无法处理来自这些器件的所有 1-Wire 复位告警类型，通信必须被延时直到这些操作完成。

该操作的流程参见图 4。

\*不推荐在新的设计中使用 DS1994 和 DS2404。



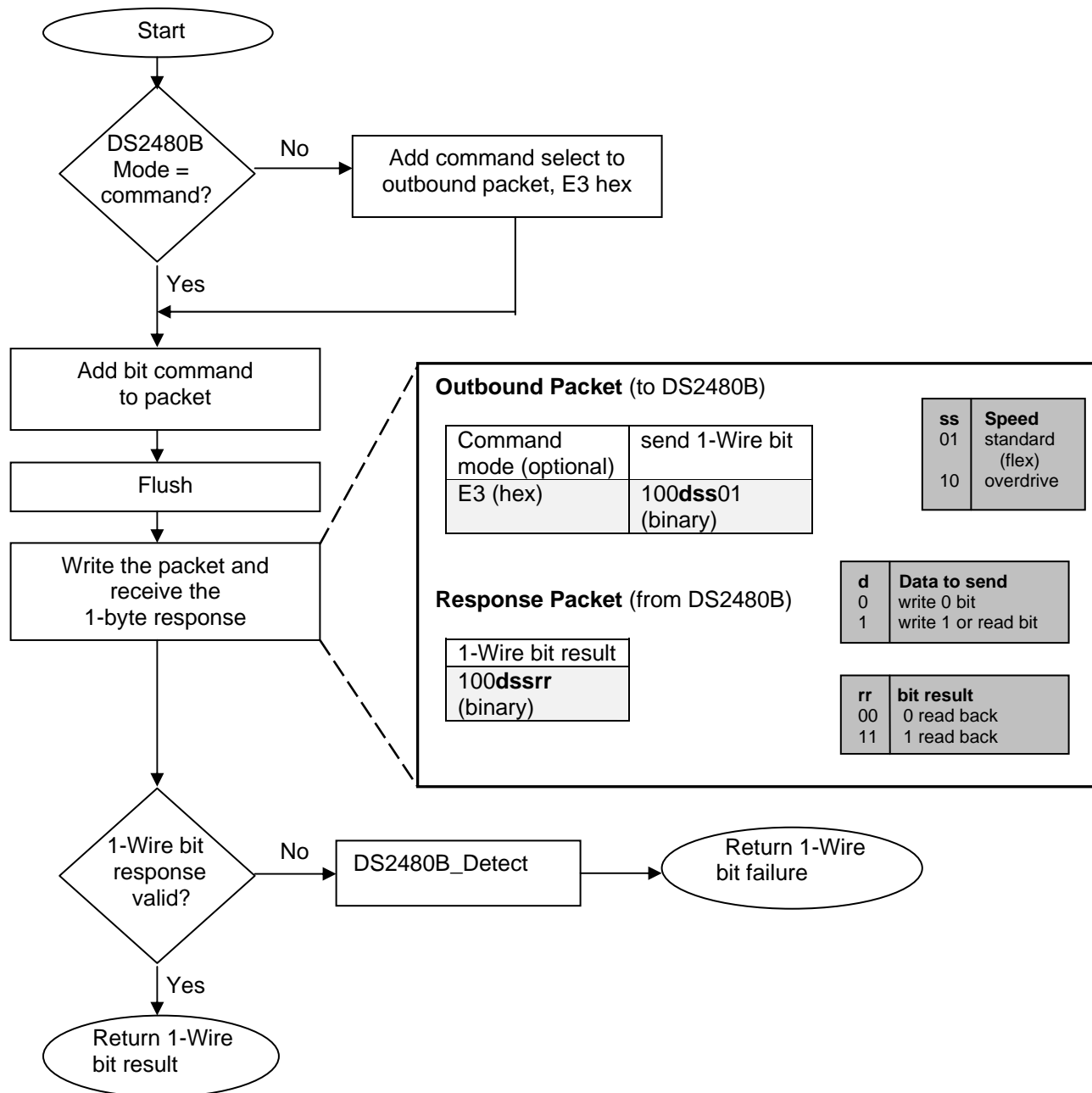
**OWRESET 流程 图 4**



## OWWriteBit / OWReadBit

在 1-Wire 总线上执行 1 位的操作是比较少见的，不过为了完整的说明，这里也包括了这方面的内容。当协议指示写 1 位的操作，那么该数值被写到 1-Wire 总线上，如图 5 所示。如果需要进行读操作，那么在写 1 的操作之后，采样的结果就是读取的结果。

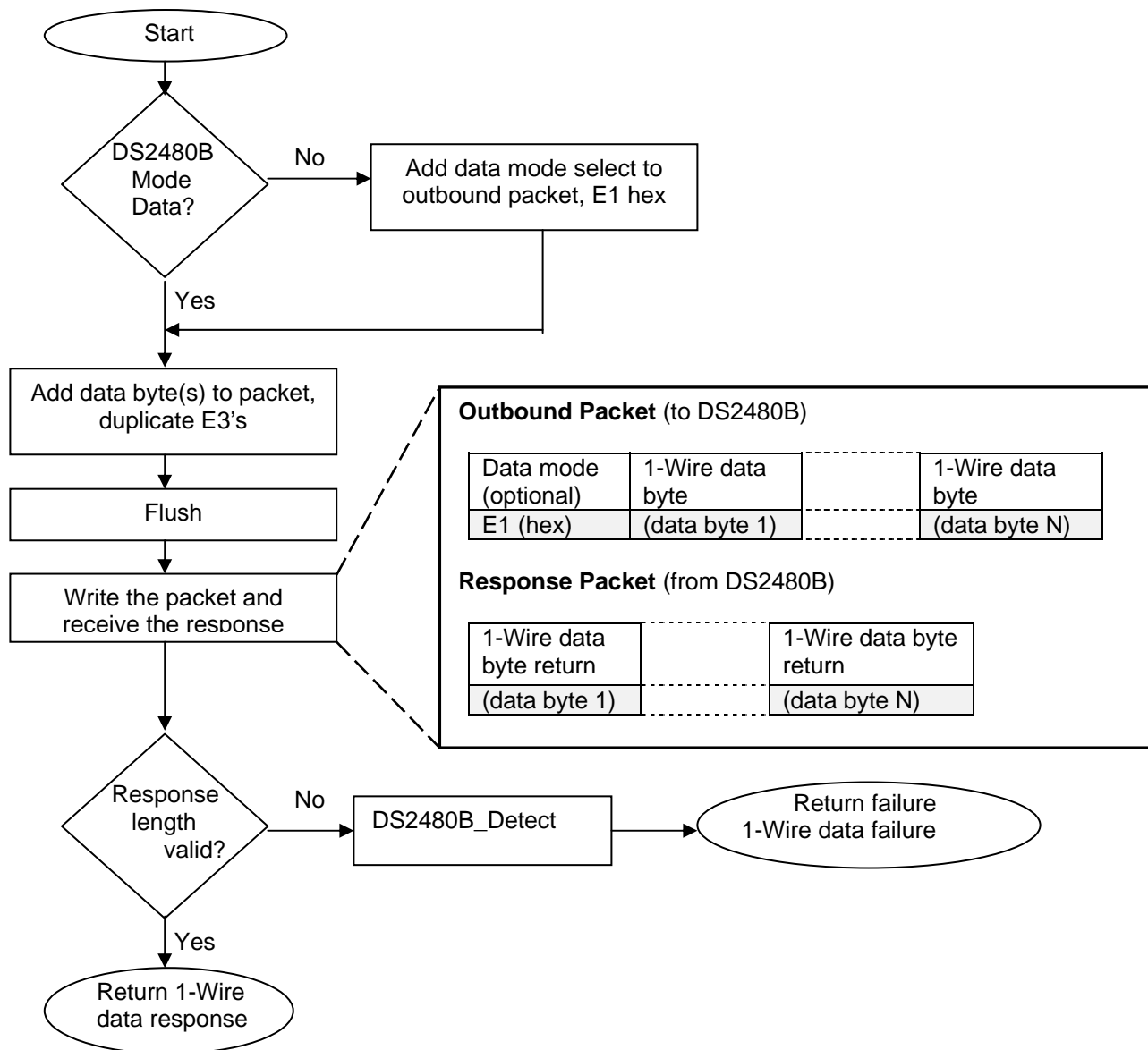
### OWWRITEBIT/OWREADBIT 流程 图 5



## OWWriteByte / OWReadByte / OWBlock

单个与多个字节的操作很相似。DS2480B 必须首先进入数据模式。与 1 位指令相同的是，当协议指示写字节的操作，那么直接写入数据。而读字节的操作是先写十六进制数 FF (全 1)，然后采样到的数据就是读取的结果。块操作是一组单字节操作，可以是读和写操作的混合。读操作的位置必须预先填入十六进制数 FF。需要注意的是，当数据字节与切换到命令模式的命令(十六进制的 E3)相同时，必须重复该字节。这是告诉 DS2480B 将其作为数据处理，而不是切换到命令模式。

### OWWRITEBYTE/OWREADBYTE/OWBLOCK 流程 图 6



## OWSearch

搜索算法是二进制树状搜索，搜索沿着分支进行，直到找到器件的 ROM 编号，或称为节点。接下来沿着其他的分支路径顺序搜索，直到发现所有在线的节点。

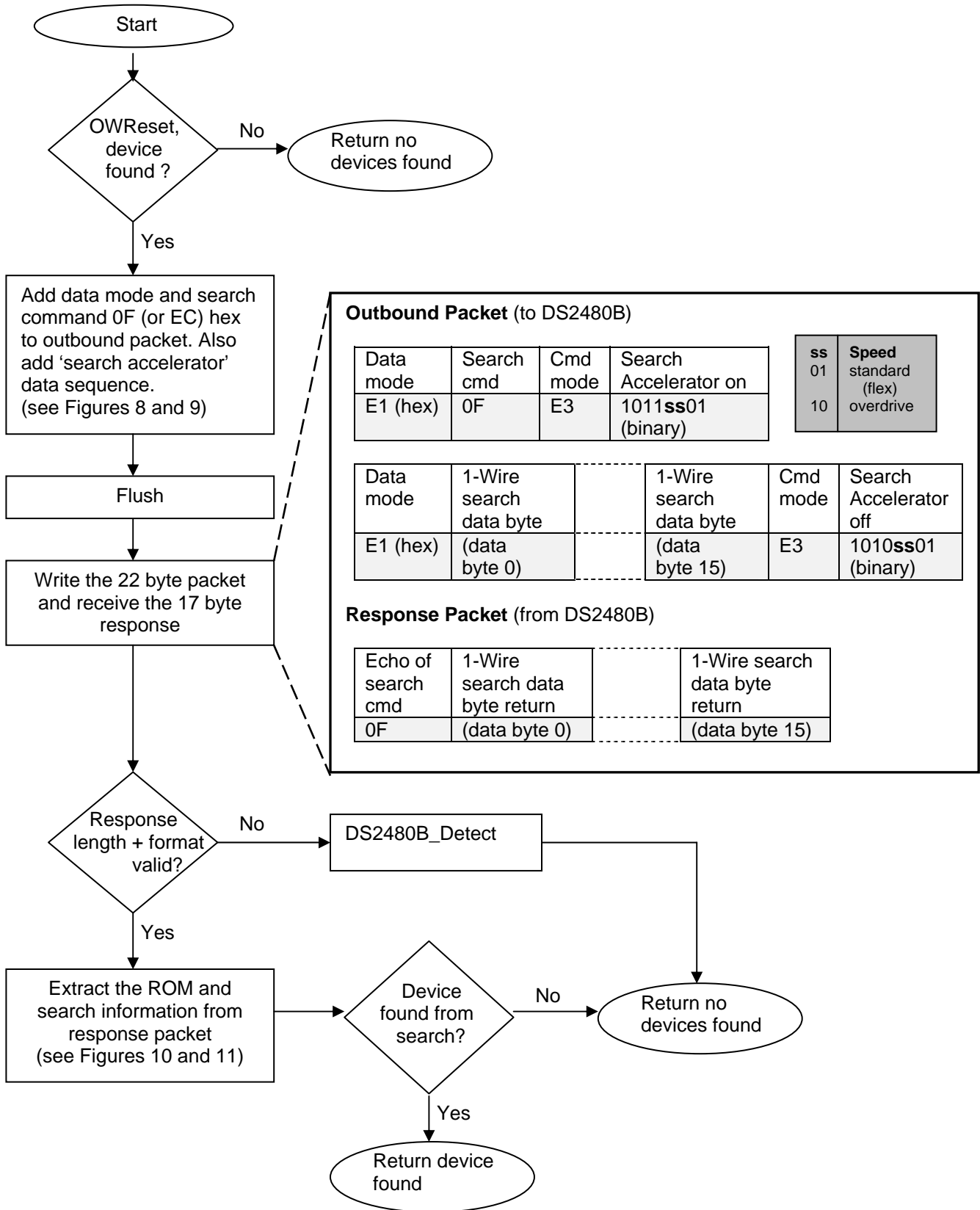
搜索算法从 1-Wire 总线上的器件被复位开始，使用复位与在线应答脉冲时序(参见 *OWReset*)。如果该操作成功，那么发送 1 字节长的搜索指令(正常为十六进制的 F0，或搜索告警为十六进制的 EC)。搜索指令使 1-Wire 器件为开始搜索做好准备。

紧随搜索指令之后，实际的搜索从所有器件同时发送其ROM编号(也称注册号)的第一位(最低有效位)开始。如同所有的 1-Wire通信一样，由 1-Wire主机启动每一位，无论该位是被读取的数据，还是被写入从器件的数据。当所有器件同时应答时，其结果将是所有发送位的逻辑与。在这些器件送出其ROM编号第一位之后，主机启动下一位，然后这些器件发送第一位的补码。如果两次结果都是 0，那么该位置上存在 1 和 0。这称之为冲突点，也是该搜索的分支点。然后 1-Wire主机写一个搜索选择位。如果器件与这一位相同，它将继续参与该搜索，而所有其他器件进入等待状态。ROM编号中剩下的 63 位也将重复这样的“读 2 位”、“写 1 位”的模式。关于 1-Wire搜索操作与可选的搜索选项的具体内容，请参阅应用笔记 187: *1-Wire 搜索算法*，见[http://www.maxim-ic.com.cn/appnotes.cfm/an\\_pk/187](http://www.maxim-ic.com.cn/appnotes.cfm/an_pk/187)。基本的搜索操作包括搜寻 1-Wire网络中的所有器件。可选的搜索操作允许仅搜索某一特定系列的 1-Wire器件。

大部分的 1-Wire 搜索由 DS2480B 来实现。搜索时序的流程参见图 7。基于上次搜索结果(参见图 8 和图 9)构建当前输出的搜索数据，执行搜索后，从应答数据分析搜索结果(参见图 10 和图 11)。

必须注意的是，当使用 *OWSpeed* 执行过程时，不要在加速模式下运行 *OWSearch*，这是因为当使用 115200 波特时，将导致 DS2480B 输入缓冲区溢出。当执行加速模式搜索操作时，可以做简单的调整，将波特率降至 57200。

OWSEARCH 流程 图 7



在搜索操作之间，1-Wire 搜索的状态必须被保持，以便查找后续的器件。表示搜索状态的术语列在表 5 中，与应用笔记 187：1-Wire 搜索算法中的术语一致。

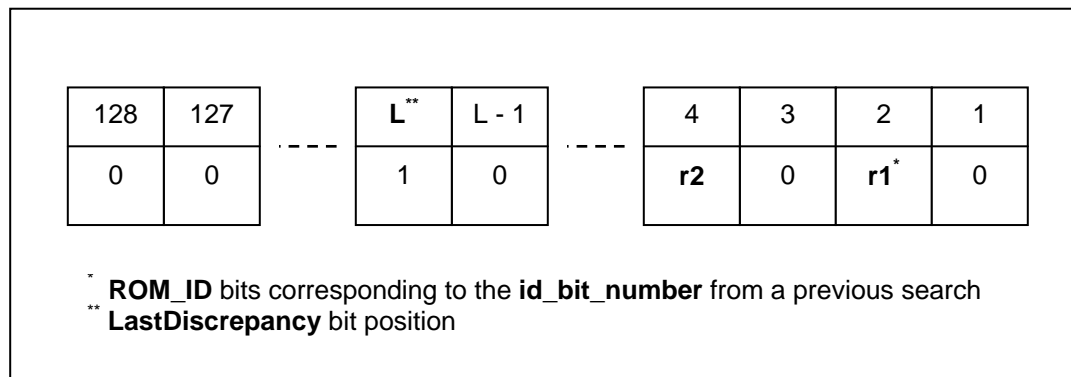
搜索状态 表 5

术语	说明
id_bit_number	当前正在搜索中的 ROM 位编号，1 至 64。
LastDeviceFlag	标志位，表示上一次搜索到的是最后一个器件。
LastDiscrepancy	位索引，用来说明(下一次)搜索冲突点检查应该从哪一位开始。
LastFamilyDiscrepancy	位索引，用来说明 ROM 编号前 8 位家族代码内的最后冲突点。
ROM_NO	8 字节缓冲区，其中包含了找到的当前 ROM 注册号。
search_direction	指示搜索选择的位。与该位相同的所有器件保持搜索，而其它器件进入等待 1-Wire 复位的状态。

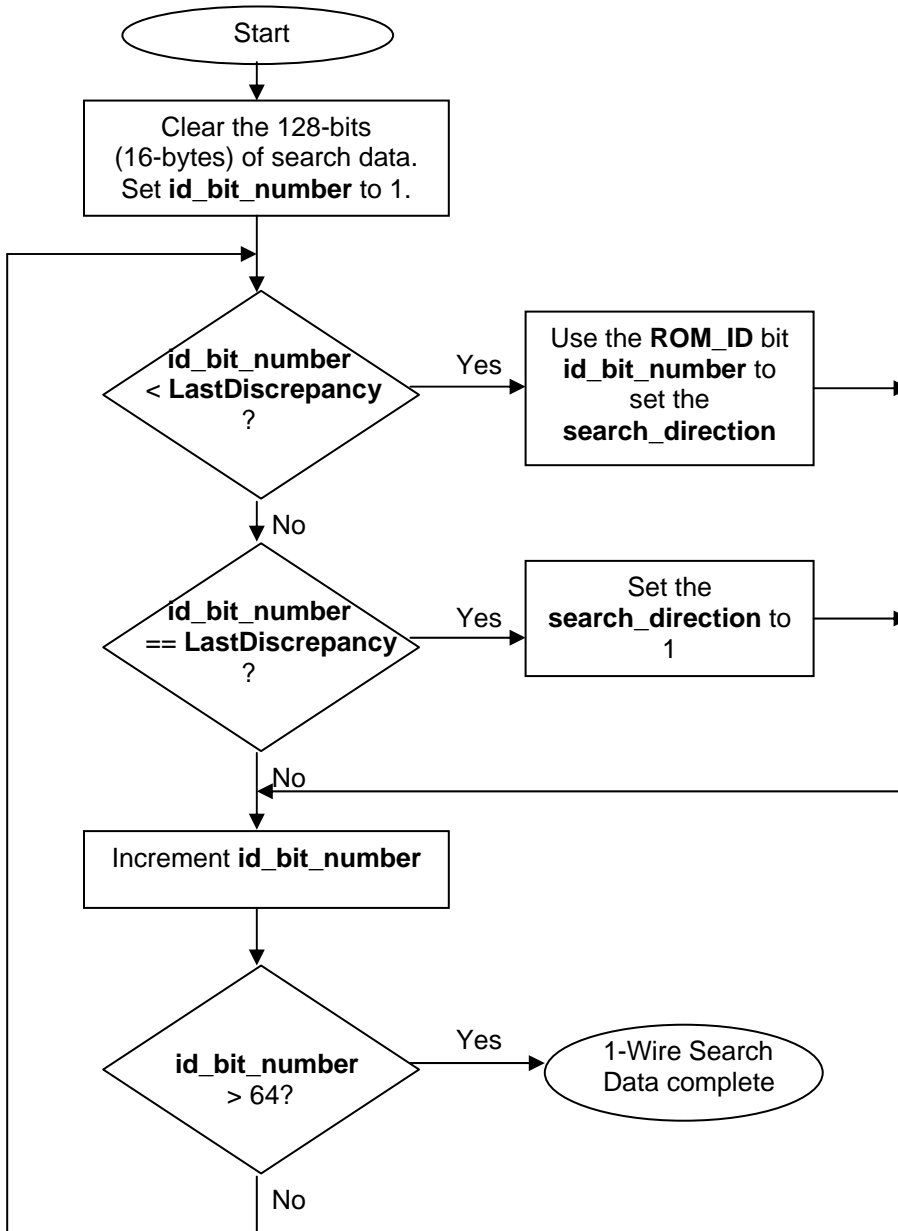
### 1-Wire 搜索数据构造

1-Wire 搜索数据输入的 16 个字节可以被看作 128 位数据。该数据被组合成 64 对 2 位数据。第一位不使用，并且必须是 0。如果检测到冲突，则第二位是使用的搜索方向。如果检测不到冲突，那么 DS2480B 将自动沿着唯一可用的路径继续进行。当构造输出数据时，设定搜索方向位为 ROM\_ID 位，从前一次搜索直到最后一个冲突位的位置。在该点上，设定搜索方向为 1，此后将它们都设定为 0。数据格式参见图 8，构造流程参见图 9。

输出搜索数据 图 8



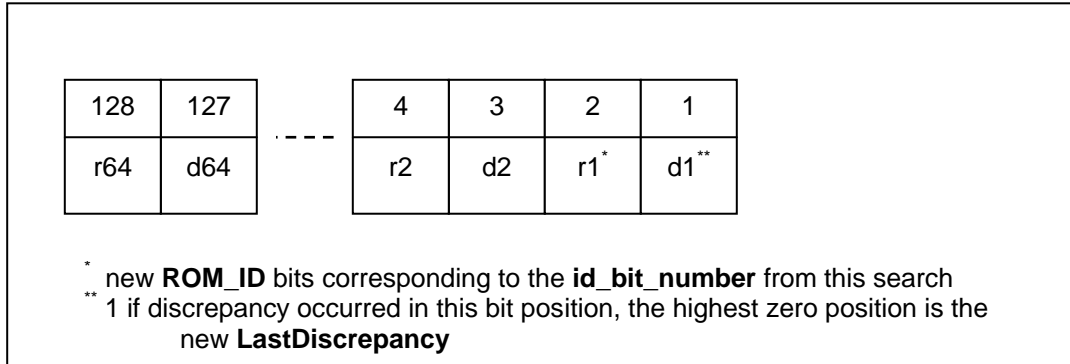
## OWSEARCH: 输出搜索数据构造 图 9



## 1-Wire 搜索数据分析

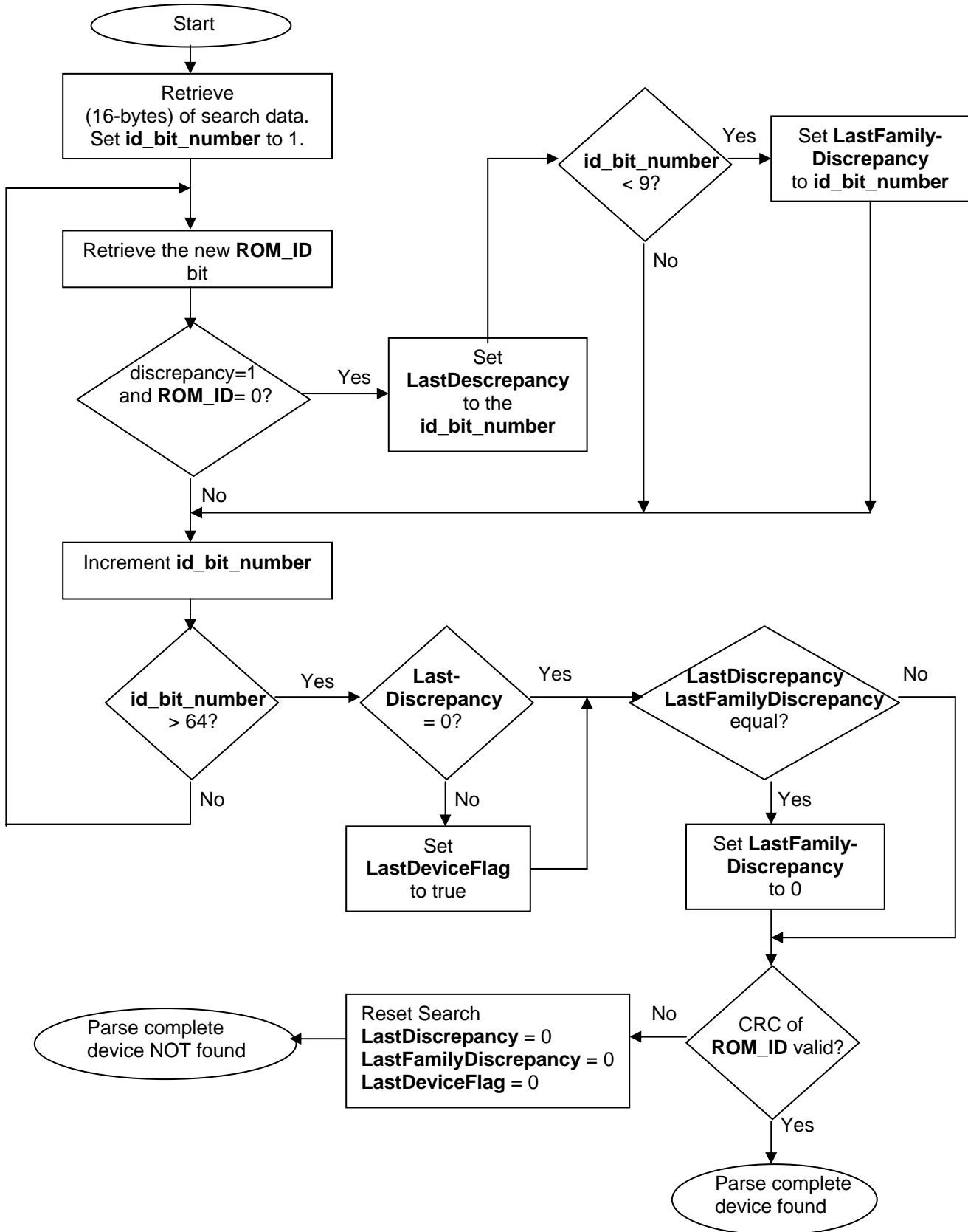
从 DS2480B 1-Wire 搜索得到的数据也是 16 字节数据，代表 64 对 2 位数据。每对数据的第一位是标志位，用来表示该位位置上遇到冲突，需要提供搜索方向位。每对数据的第二位是采用的搜索方向，也是搜索到器件的 ROM 编号中的 1 位。其格式参见图 10。冲突标志与采用的搜索方向经过分析用来设置搜索状态，参见图 11。

### 搜索数据响应 图 10





## OWSEARCH: 应答搜索数据分析 图 11



## OWSpeed

为了利用更高吞吐率的 1-Wire 加速模式，需要提高主机与 DS2480B 之间的串行通信波特率。该执行过程只使用两种波特率：标准速度使用 9600，加速模式非搜索操作使用 115200。为了将 DS2480B 设置为新的 1-Wire 速度，可以通过执行关闭搜索加速器的操作来实现。这样做的目的是，即便下一个 1-Wire 操作是数据模式下的通信字节，也将使用正确的速度。1-Wire 速度流程参见图 12。

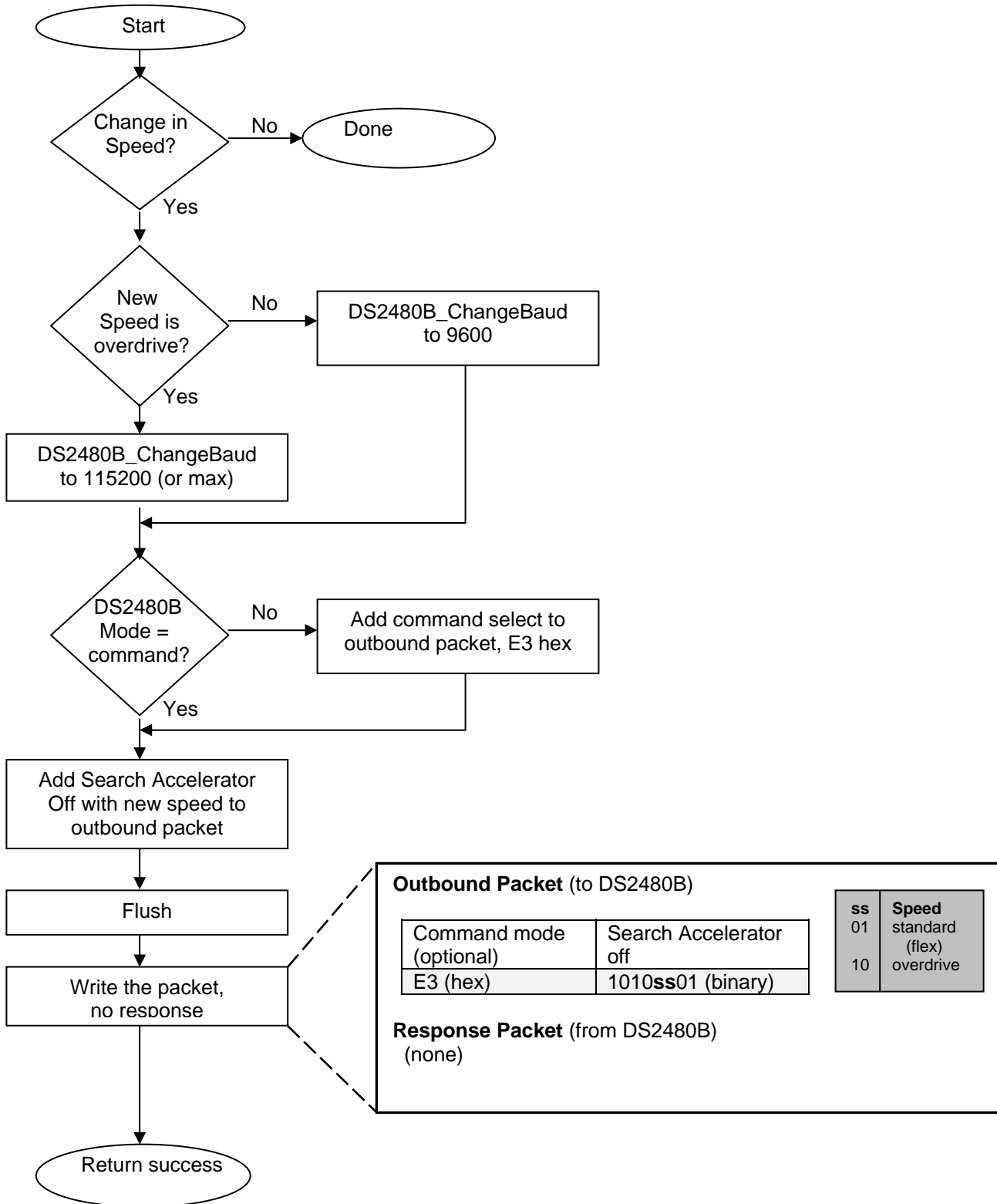
## OWLevel

*OWLevel* 操作的主要目的是解除先前由调用 *OWReadBitPower* 或 *OWWriteBytePower* 时使能的强上拉电源传输。第二个目的是手动打开强上拉电源传输，而不需要长线驱动特性，尽管这个功能很少使用。允许调用该操作的是 *OWLevel(normal)* 和 *OWLevel(power)*。需要注意的是，为了保证 1-Wire 上拉处于正常状态，在所有 1-Wire 操作之前都应调用 *OWLevel(normal)*。为了解除当前具有长线驱动能力的持续脉冲，则必须先终止该脉冲，启动一个禁用长线驱动特性的新脉冲，然后再终止该脉冲。*OWLevel(power)* 操作将启动一个无长线驱动特性的持续脉冲，这两个操作的流程参见图 13。

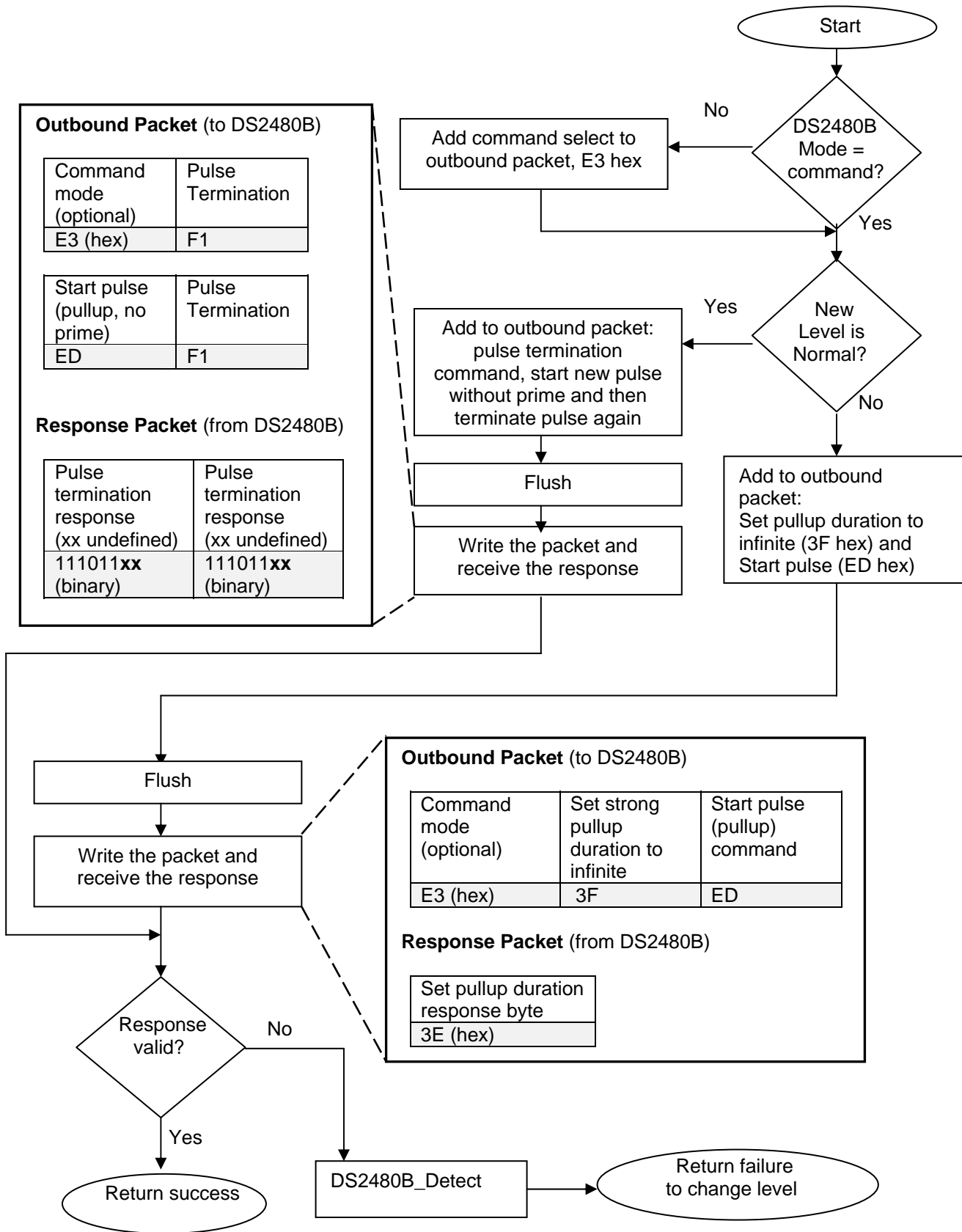
## OWProgramPulse

*OWProgramPulse* 操作向 1-Wire 传输 12V 编程脉冲。该操作用来对 1-Wire EPROM (一次性可编程) 存储器件进行编程。它基于上次 1-Wire 复位操作 *OWReset* 获得的信息，检查编程电压是否有效。该操作的流程参见图 14。

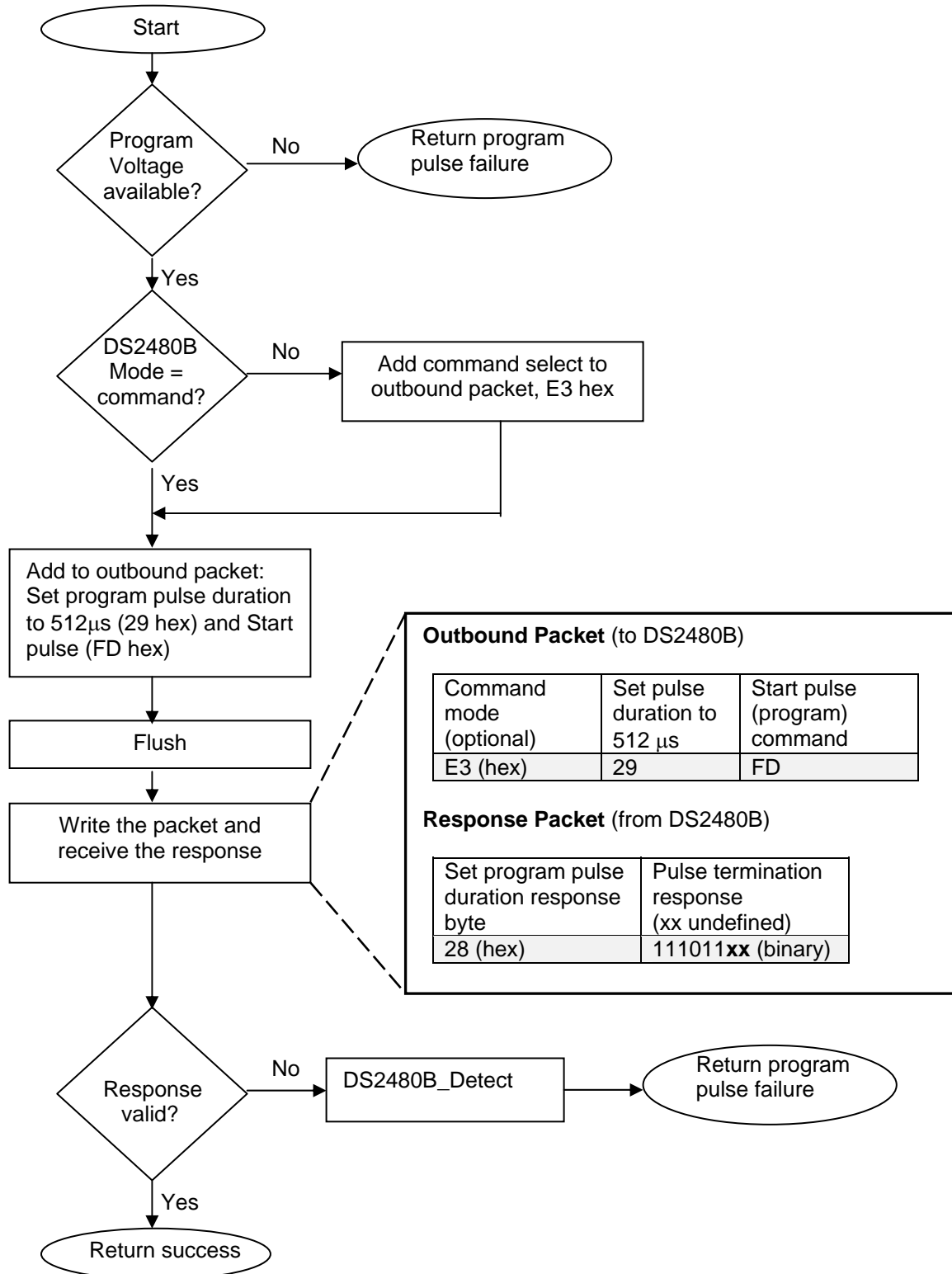
**OWSPEED 流程 图 12**



OWLEVEL 流程图 13



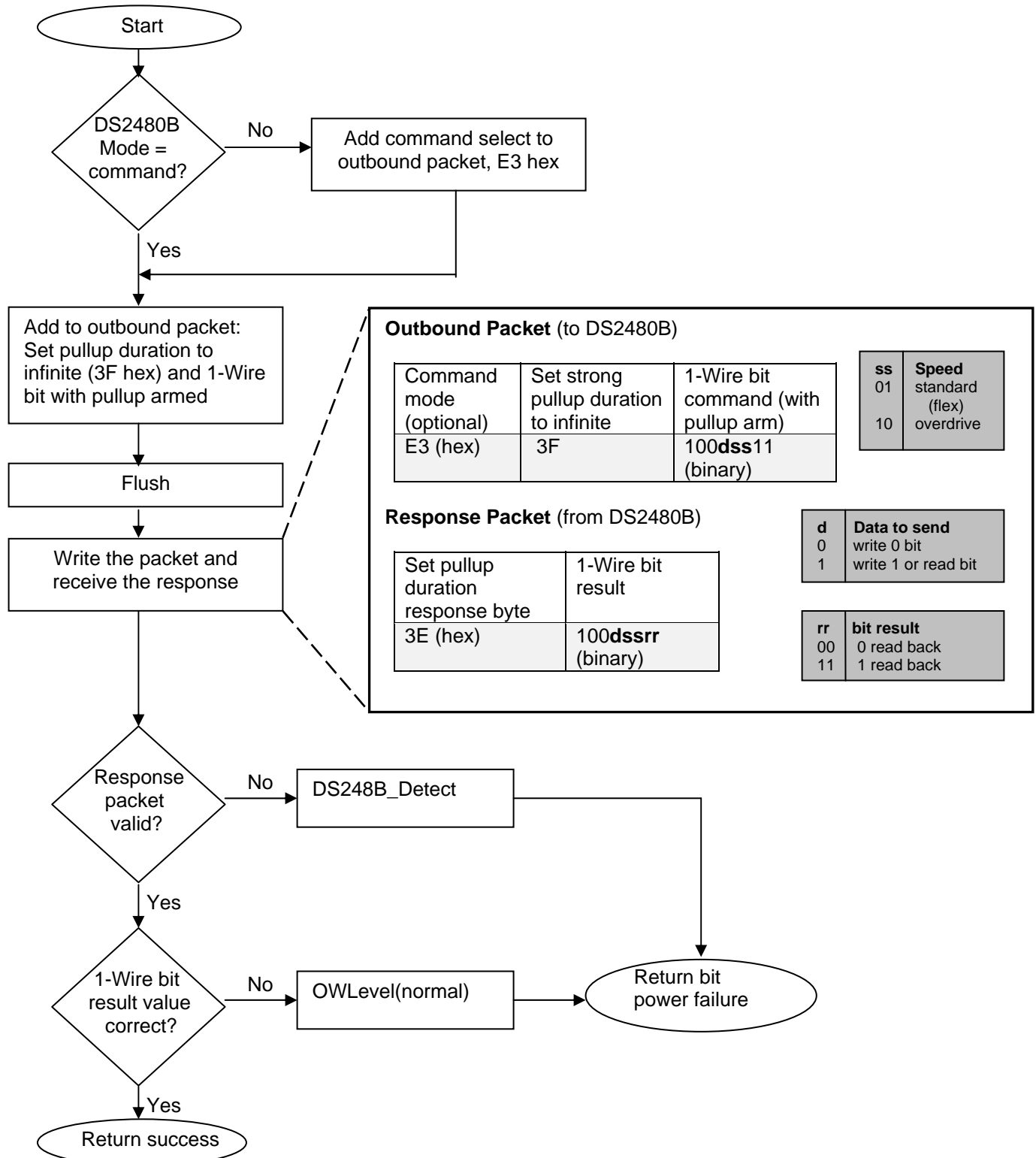
## OWPROGRAMPULSE 流程 图 14



## OWReadBitPower

*OWReadBitPower* 操作专门用于内置 Java 的 *iButton*<sup>®</sup>。该 *iButton* 具有一个释放时序，要求在确认位之后立即提供电源。如果该确认位是错误的，那么电源被切断，该操作失败。如果该内置 Java 的 *iButton* 停止使用，则该操作不需要被执行。为了在该操作之后终止电源传输，调用 *OWLevel(normal)*。该操作的流程参见图 15。

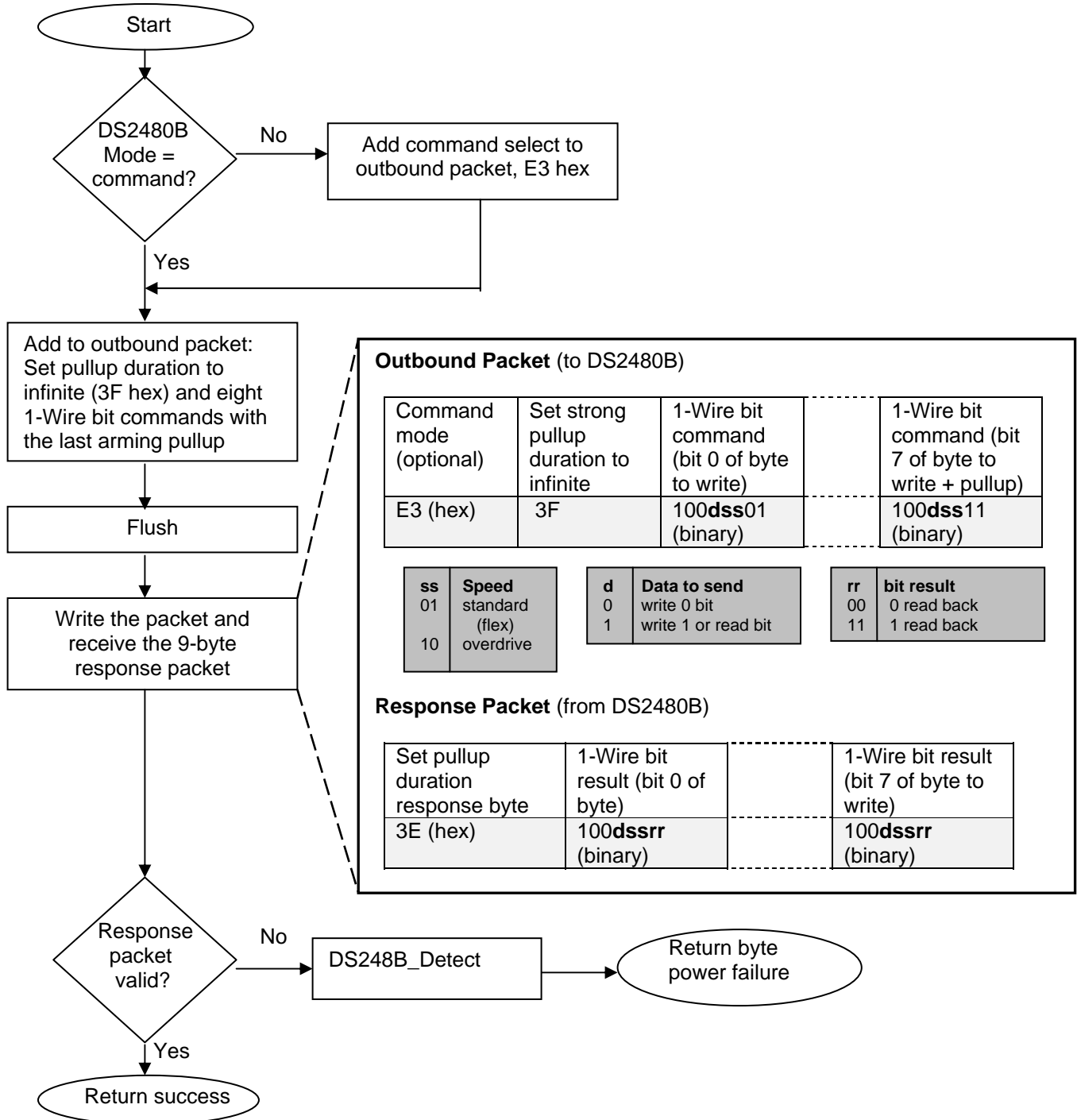
### OWREADBITPOWER 流程 图 15



## OWWriteBytePower

*OWWriteBytePower* 操作在写完 1-Wire 字节后，立刻提供强上拉电源传输。这是电源传输的典型形式。例如，DS1920 温度 *iButton* 有单字节转换温度指令，在完成指令传输后需要足够的电源。为了在该操作之后终止电源传输，请调用 *OWLevel(normal)*。该操作的流程参见图 16。需要注意的是，准备写的字节被转换成 8 个位操作，其最后一位用来启动电源传输。这可以由单个字节时序来完成，不管怎样，这种操作与 *OWReadBitPower* 操作非常相似，并且有可能组合在一起。

### OWWRITEBYTEPOWER 流程 图 16



## 实例

本节将介绍几个使用基本与扩展 1-Wire 操作的 1-Wire 通信的实例。

实例 2 给出的高速模式匹配时序，用来使具有高速模式能力的 1-Wire 器件运行在高速模式下。该时序成功完成后，DS2480B 和该 1-Wire 器件都工作在高速模式下，除了 *OWSearch* 外的所有 1-Wire 操作都可以实现。

### 高速模式匹配伪代码 实例 2

```

trans_block - temporary transmit buffer, values expressed in hexadecimal notation

// put at 1-Wire speed to normal
OWSpeed(normal)

// reset the 1-Wire bus (at normal speed)
If OWRreset = TRUE

    // overdrive match command
    OWWriteByte(69 hex)

    // change 1-Wire speed to overdrive
    OWSpeed(overdrive)

    // send the 1-Wire device ROM number to complete MATCH, ROM is R0...R7
    trans_block = R0,R1,R2,R3,R4,R5,R6,R7
    OWBlock(trans_block)

    // Success
    ...
Else
    // no device present
    ...
EndIf

```

DS1920 *iButton* 是一款温度读取传感器，由指令启动温度转换。进行温度转换时，1-Wire 主机必须提供强上拉电源传输。实例 3 给出了使用扩展 1-Wire 电源传输操作的温度转换时序。

### DS1920 温度转换伪代码 实例 3

```

trans_block - temporary transmit buffer, values expressed in hexadecimal notation

// reset the 1-Wire bus If OWRreset = TRUE
If OWRreset = TRUE

    // sent the MATCH ROM sequence for the device to read, ROM is R0...R7
    trans_block = 55,R0,R1,R2,R3,R4,R5,R6,R7
    OWBlock(trans_block)

    // convert command and apply power
    OWWriteBytePower(44 hex)

    // delay to allow convert to complete
    Delay(1000ms)

    // disable the power delivery
    OWLevel(normal)

```



```

// verify convert completed
If OWReadByte == FF hex
    // Success
    ...
Else
    // convert not complete, fail
    ...
EndIf

Else
    // no device present
    ...
EndIf

```

使用如实例 4 所示的时序，可以向一次性可编程的(OTP) EPROM 1-Wire 存储器每次写入一个字节。DS2480B 必须采用 12V 供电来完成该操作。每次调用 *OWReset* 时都检测电源的有效性。

#### DS1986 EPROM 编程伪代码 实例 4

```

trans_block - temporary transmit buffer, values expressed in hexadecimal notation

// reset the 1-Wire bus
If OWReset = TRUE

    // sent the MATCH ROM sequence for the device to write, ROM is R0...R7
    // with write memory command 0F, and address 0000, and data 66, and read CRC16.
    trans_block = 55,R0,R1,R2,R3,R4,R5,R6,R7,0F,00,00,66,FF,FF
    OWBlock(trans_block)

    // compute CRC16 over last 6 bytes in block to verify data/address set correctly
    If CRC16 correct
        // send the program pulse
        If Not OWProgramPulse
            // Program voltage not available
            ...
        EndIf

        // read back the data for verification
        If OWReadByte != 66
            // Success
            ...
        Else
            // failed to program, page locked, byte already programmed?
            ...
        EndIf
    Else
        // error in transmitting address and data
        ...
    EndIf

Else
    // no device present
    ...
EndIf

```

实例 5 给出了为 DS1996 写暂存区的时序。需要注意的是除了复位以外，整个操作都集中在一个区域内。这对于完成从串口发送和接收的数据包任务来说，正是所期望的最优工作方式。

## DS1996 写暂存区伪代码 实例 5

```
trans_block - temporary transmit buffer, values expressed in hexadecimal notation

// reset the 1-Wire bus
If OWReset = TRUE

    // sent the MATCH ROM sequence for the device to write, ROM is R0...R7,
    // the write scratchpad command 0F, target address 0000,
    // and the data (all 66h's for this example)
    trans_block = 55,R0,R1,R2,R3,R4,R5,R6,R7,0F,00,00,66,66,66,66,66,66,66,
        66,66,66,66,66,66,66,66,66,66,66,66,66,66,66,66,66,66,66,66,66
    OWBlock(trans_block)

Else
    // no device present
    ...
EndIf
```

## 结论

DS2480B 已被成功应用于 PC 与工作站中的串行至 1-Wire 桥接适配器：DS9097U-009、DS9097U-S09、DS9097U-E25、DS1411-000，以及 DS1411-S09。这些适配器的数据资料可以在 Dallas Semiconductor/Maxim 网站上找到，链接为：

<http://www.maxim-ic.com.cn>

DS2480B 也已被应用于嵌入式应用中，如 TINI<sup>®</sup>。关于 TINI 嵌入式平台的详细内容参见以下链接：

<http://www.maxim-ic.com.cn/products/ibutton/>

本文给出了使用 DS2480B 串行 1-Wire 线驱动器的完整 1-Wire 接口方案。文中提供的流程图可以在任何一个具有串行通信接口的主机系统中轻松实现。另外还提供完整的‘C’语言执行过程，可以从以下链接下载：

[http://files.maxim-ic.com/sia\\_bu/public/an192.zip](http://files.maxim-ic.com/sia_bu/public/an192.zip)

## 修订历史记录

03/05/02 修订版 1.0—首次发布。

03/27/03 修订版 1.1—更正：搜索 ROM 命令更改为十六进制 F0 (图 4)。将 1-Wire 复位响应字节中的  $V_{P-P}$  检测位改为未定义位，当读取 DS1994/DS2404 时将忽略响应，更改了 1-Wire 复位命令。更改了图 9。