



AnyWatt™ 可编程电源技术标准

RVE1.0

Revision History

Revision	Description	Issue Date
REV 1.0	初始版本	15 October, 2016

REVISION HISTORY	2
1 介绍	4
1.1、概述	4
1.2、应用范围	4
2、系统架构	4
3、实现流程	4
4、电源要求	6
4.1、基本电气参数	6
4.2、SRC 端的 USB PD 通信功能	6
4.3、线性反馈功能	7
4.4、SRC 的过压保护	7
4.5、SRC 的过流保护	7
4.6、SRC 的过温保护	7
5、负载要求	8
5.1、基本电气参数	8
5.2、SNK USB PD 通信功能	8
5.3、SNK 的过压保护	8
5.4、SNK 的过流保护	8
5.5、SNK 的过温保护	8
附录 A 使用 ANYWATT™功能实现手机快充（恒压模式）	9
附录 B 使用 ANYWATT™功能实现手机直充（恒流模式）	13

1 介绍

1.1、概述

AnyWatt™，是基于 USB PD 通信协议制定的智能供电技术。其特征在于，负载端可以通过标准的 USB PD 协议对供电端的电源行为进行控制，实现技术标准范围内，任意电压的恒压输出，或者任意电流的恒流输出，或者任意功率的恒定功率输出。由于通过这项技术，负载可以获得任意电压、电流以及功率，从而满足自己所需要的电能，因此称为 AnyWatt™ 技术。与其他基于 USB PD 的智能供电协议相比，具有完全兼容 USB PD 2.0 和 USB PD 3.0，并且无需采用 VDM，易以实现的特点。

1.2、应用范围

AnyWatt™ 技术适用于采用 USB Type-C 物理接口，供电电压在 20V 以内，供电功率不大于 100W 的设备，可广泛应用于移动终端，家用电器，办公设备，工业医疗等领域。同时，也为内置锂离子电池的设备充电过程管理提供了快充和直充的实现途径。

2、系统架构

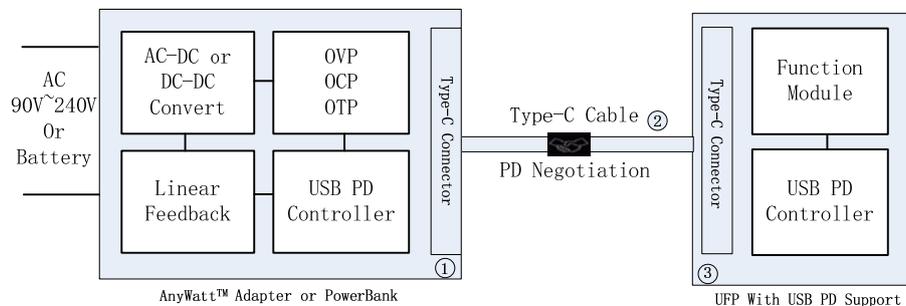


图 2-1 AnyWatt™ 技术系统架构

如图所示，AnyWatt™ 技术包括了①USB TYPE-C SRC(或者 USB TYPE-C DRP)，②USB TYPE-C Cable 和③USB TYPE-C SNK (或者 USB TYPE-C DRP)。其中，SRC 端（即①）和 SNK 端都包含了 USB PD Controller 用于进行供电协商，一个线性可调反馈模块用于进行恒压、恒流和恒功率输出控制。当双方或者其中一方为 DRP 设备时，可以根据实际供电需要，进行 Power Role Swap。

3、实现流程

AnyWatt™ 技术的关键是，SNK 可以通过发送带 Mismatch 标志位的 USB PD Request 数据包，来诱导 SRC 发送 Get_Sink_Capability 数据包，并通过 Sink_Capability 数据包发送自己当前所需要的供电配置选项。SRC 收到返回的 Sink_Capability 数据包后，根据自身的供电能力，重新构建与 Sink_Capability 相同的 Source_Capability 数据包，或者在自己的供电能力范围内，最接近于 Sink_Capability 的 Source_Capability 数据包。SNK 在收到为自己所定制的 Source_Capability 数据包后，评估出最适合自己当前状态的供电方式，并发送 Request，完成电源配置。当负载内部电能消耗形势发生改变时，SNK 可以重复以上过程，让 SRC 的输出随 SNK 的需求变化而变化。

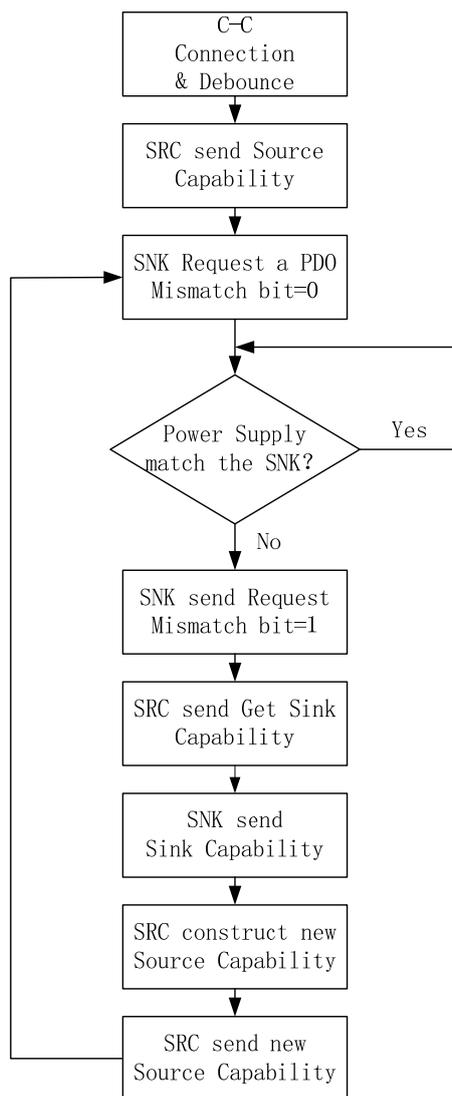


图 3-1 AnyWatt™技术的工作流程

在图 3-1 的工作流程中，SNK 所发送的 Sink_Capability 数据包，可以是 Fixed Supply, Variable Supply, Battery Supply 中的任意一种或多种的组合，分别对应着恒压供电，恒流供电和恒功率供电三种供电方式。SRC 收到 SNK 的 Sink_Capability 数据包后，根据该数据包重新构建自己的 Source_Capability 数据包，并发送给 SNK。SNK 还可以通过发送 SOFT_RESET 数据包，复位 SRC 的 Source_Capability 数据包，使其回到最初的默认值。

4、电源要求

4.1、基本电气参数

AnyWatt™技术要求电源端具有恒压，恒流，恒功率三种供电方式。在总功率不超过自身电源能力的情况下，满足 SNK 端的供电需求。以下表格中 U_{MAX} 、 I_{MAX} 、 P_{MAX} 分别为 SRC 的电源模块，最大能够输出的电压，电流和功率。

表 4-1 Maximum character with E-mark Cable support 5A

	Max	Min
I	5000mA	100mA
U	20V	3V
P	100W	300mW

表 4-2 Maximum character without E-mark Cable support 3A

	Max	Min
I	3000mA	100mA
U	20V	3V
P	60W	300mW

表 4-3 受电源本身最大功率 P_{MAX} 及最大电压 U_{MAX} 和最大电流 I_{MAX} 限制的情况

	Max	Min
I	Max ($I_{MAX}, P_{MAX}/U_{OUT}$)	100mA
U	Max ($U_{MAX}, P_{MAX}/I_{OUT}$)	3V
P	P_{MAX}	300mW

*数值如果超过表 4-1 和表 4-2，则以表 4-1 和表 4-2 结果为准

4.2、SRC 端的 USB PD 通信功能

AnyWatt™技术要求 SRC 具有 USB PD 通信功能，能够判别 SNK 所发送的 Request 数据包中是否包含有 Mismatch 标志位，从而决定是否需要发送 Get_Sink_Capability 重新去获取 SNK 的电能需求。在供电过程中，SRC 也可以主动发出 Get_Sink_Capability 命令，用于检测 SNK 的供电需求是否已经发生改变，或者作为 Ping 事件，确认 SNK 的持续存在，保障供电安全。

SRC 收到 Sink_Capability 数据包之后，构建 Source_Capability 的方法应遵循以下原则：

第一，当 SRC 收到了 Fixed 类型 Sink_Capability 数据包时，如果 PDO 数据包的供电要求不超过 SRC 自身的供电能力(U_{MAX} , I_{MAX} , P_{MAX})，则回复的 Source_Cap 数据包可以直接使用 Sink_Capability 中的 PDO 所要求的电气参数。当 SRC 自身的供电能力达不到 Sink_Capability 的要求时，应遵从 U, I, P 逐级适应原则。即先检查电压参数，如果电压参数超出了，优先把电压降低到 U_{MAX} 。然后检测电流参数，如果电流参数超过了 I_{MAX} ，则把电流降低的 I_{MAX} 。然后检测降低后的 U 和 I，相乘是否超过 P_{MAX} ，如果超出，则在维持电压不变的情况下，降低电流。

第二，当 SRC 收到了 Variable 类型的 PDO 数据包时。应该在遵循第一条供电原则的前提下，把 Variable 的 PDO 分拆成两个 PDO，包含到 Source_Capability 数据包中。第一个是 Fixed 类型 PDO，其输出电压为收到的 Variable PDO 的最高电压，输出电流为 Variable 类型 PDO 中的工作电流。另外一个 Variable 类型的 PDO，其内容可以与收到的 Variable 类型 PDO 一致。拆分成两个 PDO 的原因是，某些 SNK 用 Variable 类型 PDO 并不是表达真实的用电需求，而是表达能够接受的电源范围。最终每一个 PDO 的 U, I, P 都应该遵循第一条所述的原则。

第三，当 SRC 收到了 Battery 类型的 PDO 数据包时。应该在遵循第一条供电原则的前提下，把 Battery 类型的 PDO 分拆成两个 PDO，包含到 Source_Capability 数据包中。第一个是 Fixed 类型的 PDO，其输出电压为收到的 Battery PDO 的最高电压，输出电流为 Battery PDO 中的工作功率除以最高电压所对应的电流。第二个是 Battery 类型的 PDO，其内容可以与收到的 Battery 类型的 PDO 一致。拆分成两个 PDO 的原因是，某些 SNK 用 battery 类型的 PDO，并不是表达真实的用电需求，而是表达能够接受的电源范围。最终每一个 PDO 的 U, I, P 都应该遵循第一条所述原则。

第四，在以上三条规则符合的情况下，所有的 PDO 排列顺序应该遵循 USB PD 标准文档，即 fixed 类型 PDO 放在前面，然后是 Battery 类型，最后是 Variable 类型。如果最终的 Source_Capability 数据包中不包含 vsafe5v 选项，则应该增加一个 vsafe5v 的 PDO，放在第一个 PDO 的位置。

第五，在以上四条规则符合的情况下，最终的 Source_Capability 数据包中的 PDO 数量如果超过了 7 个，则遵循 Fixed 类型优先，然后 Battery，然后 Variable 的原则，优先把排在前面的 7 个 PDO 作为 Source_Capability 的内容。

第六，当 SRC 收到 SNK 的 SOFT_RESET 或 HARD_RESET 数据包，或者主动发出这两种数据包时，SRC 应该复位 Source_Capability 数据包，使其恢复到最初的默认值。当 CC 连接重新建立时，SRC 也应该复位 Source_Capability 数据包，使其恢复到最初的默认值。

4.3、线性反馈功能

传统的固定电阻反馈功能，只能够输出几档固定的电压。而线性反馈功能，可以通过 ADC 检测及 DAC 输出，形成闭环，灵活的实现恒压，恒流，恒功率三种模式的输出。实现该线性反馈功能的途径，包括但不限于 DAC，PWM 等功能模块。

4.4、SRC 的过压保护

SRC 端需要对来自 AC-DC 或者 DC-DC 的输出电压进行监控，如果完成反馈设置后，输出电压仍然高于 PD 协商电压值的 20%，则必须关闭 VBUS_EN。并持续监控，在 VBUS 电压不超出 PD 协商值 10%后，才可以打开 VBUS，往 SNK 端供电。如果保护期间，CC 连接中断，或者 PD 通信被 HardReset，则应该重新初始化供电过程。

4.5、SRC 的过流保护

SRC 端需要持续检测输出电流值，当输出电流值超出 PD 协商值 10%时，应该开始线性降低电压。当输出电流超出协商值的 20%应该启动过流保护机制，关闭 VBUS 输出，在 3 秒后，重新初始化供电过程。

4.6、SRC 的过温保护

SRC 需要持续监控自身的温度，当 PCBA 温度高于 120℃时，应启动过温保护机制。关闭 VBUS 输出，当温度降低至 80 度以下时，重新初始化供电过程。

5、负载要求

5.1、基本电气参数

AnyWatt™技术要求 SNK 端能够在 5V 供电条件下，保证 PD 模块的正常工作和支持 Deadbattery 功能。

5.2、SNK USB PD 通信功能

SNK 端必须具备 USB PD 2.0 以上版本的通信功能。SNK 端需要时刻监控系统的工作状态，如果当前从 SRC 所 Request 的电能供应已经不适合下一步的正常工作需求了，则需要通过发送一个带 Mismatch=1 标志位的 Request 数据包给 SRC，从而通知 SRC，需要重新进行供电协商。SRC 收到 Mismatch=1 的 Request 数据包后，会发送 Get_Sink_Capability 数据包，并在分析此数据包后，结合自身电能模块的能力，生成新的 Source_Capability 数据包。SNK 端可以从中选择最适合自己下一步工作的 PDO，完成一次电源的动态协商。

5.3、SNK 的过压保护

SNK 端需要对来自 SRC 的供电电压进行持续监控，如果 SRC 端发送 PS_RDY 后，输出电压仍然高于 PD 协商电压值的 20%，则必须关闭 VBUS_EN，并持续监控，在 VBUS 电压不超出 PD 协商值 5%后，才可以打开 VBUS 输入。如果保护期间，CC 连接中断，或者 PD 通信被 HardReset，则应该重新初始化供电过程。如果电压持续过高时间超过 3 秒，则发出 HardReset。

5.4、SNK 的过流保护

SNK 端需要持续检测输入电流值，当输入电流值超出 PD 协商值 10%时，则应该关闭 VBUS_EN，并发送 HardReset。

5.5、SNK 的过温保护

SNK 端的过温保护，根据器件自身的功能和性能而设定，不在本文讨论范围。

附录 A 使用 AnyWatt™功能实现手机快充（恒压模式）

AnyWatt™技术允许 SRC 设备在 SNK 设备的请求下，输出 3V~20V 之间的电压，最小分辨率为 50mV，可以为手机提供高压快充解决方案。随着电池对充电电压的变化，该技术及时调整 SRC 的输出电压。

图 A-1 和图 A-2 展示使用 AnyWatt™功能实现手机快充的通信过程。

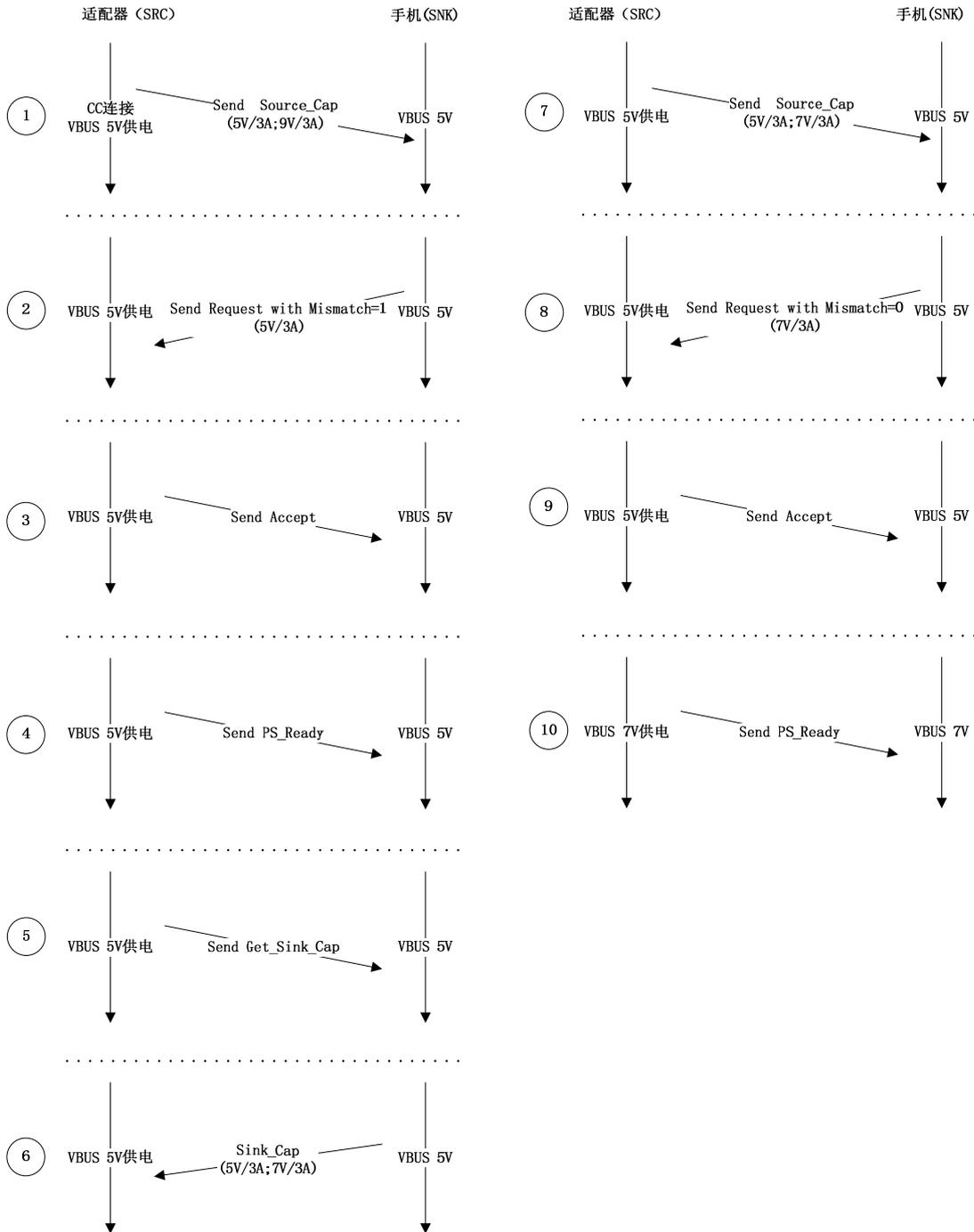


图 A-1 Fixed Supply 类型 Power Negotiation

图 A-1 对应的步骤如表 A-1 所示：

表 A-1 Power Negotiation 步骤

步骤	适配器	手机
1	适配器检测到有手机接入，打开 VBUS，提供 5V 电压，并发送自身支持的 Source_Cap	
2		手机收到适配器的 Source_Cap，评估以后，没有完全满足自己的 Source_Cap，发送包含 Mismatch=1 的 Request，请求其中的一档
3	适配器收到手机的请求，经评估在提供 Source_Cap 范围内，发送 Accept	
4	同意手机的请求以后，设置反馈，输出手机请求电压，并发送 PS Ready	
5	由于检测到手机的请求中 Mismatch=1，发送 Get_Sink_Cap，询问手机所需 Power	
6		收到适配器的 Get_Sink_Cap，回复 Sink_Cap 告知适配器需要提供的 Power
7	适配器收到手机的 Sink_Cap，构造出在其供电范围内最合适的 Source_Cap 发送给手机	
8		手机收到新 Source_Cap，发送包含 Mismatch=0 的 Request
9	适配器收到手机的请求，经评估在提供 Source_Cap 范围内，发送 Accept	
10	同意手机的请求以后，设置反馈，输出手机请求电压，并发送 PS Ready	

充电过程中，手机根据自己需求的变化，随时可以主动发起通信，使适配器的供电跟随手机的需求变化。过程如图 A-2 所示（以加电压为例）。

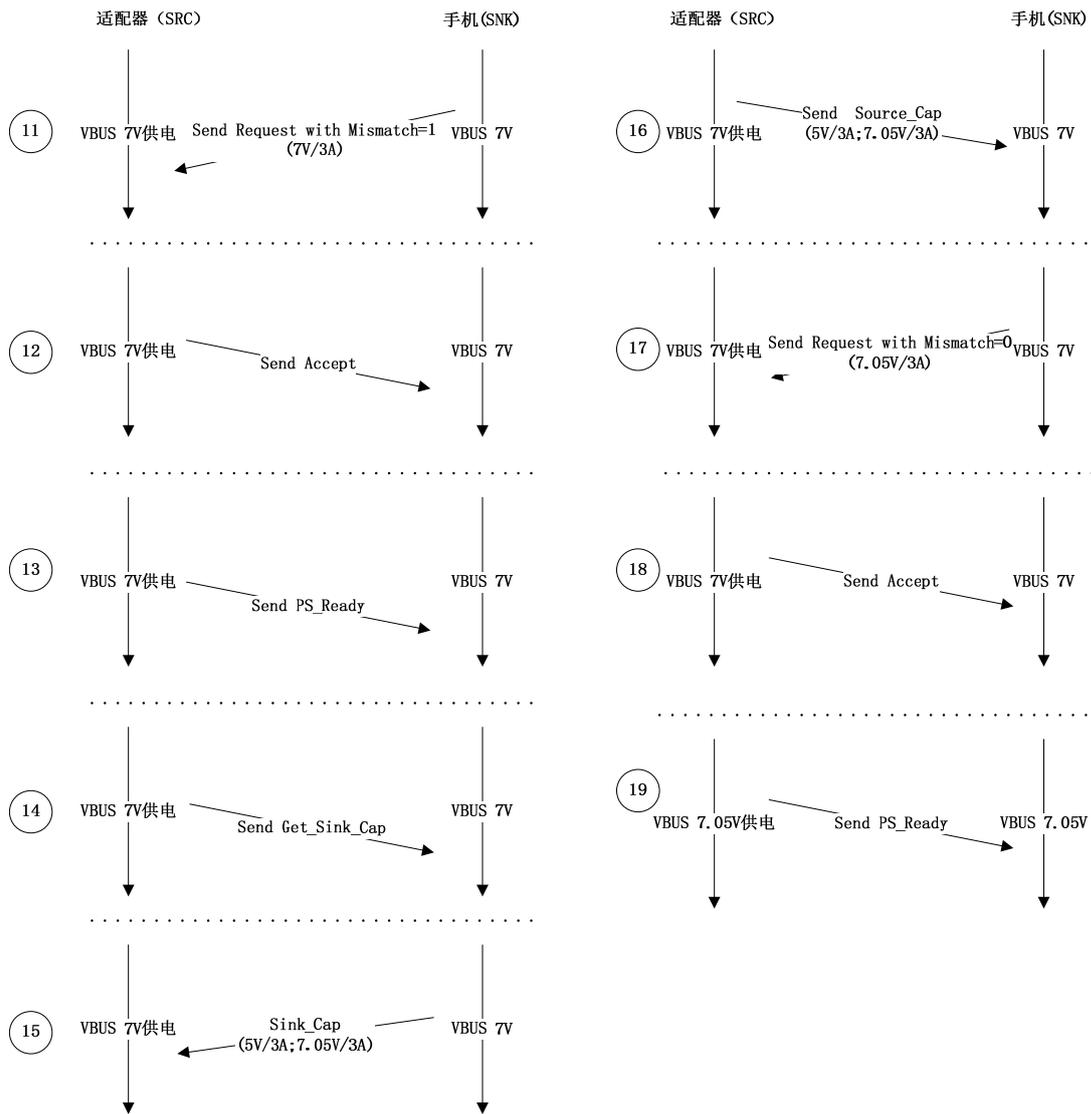


图 A-2 手机发起 Power Negotiation

图 A-2 对应的步骤如表 A-2 所示：

表 A-2 手机发起 Power Negotiation 步骤

步骤	适配器	手机
11		手机的需求发生变化，主动发送包含 Mismatch=1 的 Request，请求的 Power 与上一次 Request 相同，只是 Mismatch 不同
12	适配器收到手机的请求，由于与上一次请求的 Power 一样，立即发送 Accept	
13	由于与上一次请求的 Power 一样，不用设置反馈，立即发送 PS Ready	
14	由于检测到手机的请求中 Mismatch=1，发送 Get_Sink_Cap，询问手机所需 Power	
15		收到适配器的 Get_Sink_Cap，回复

		Sink_Cap 告知适配器需要提供的 Power, 电压值比上一次增加了 0.05V
16	适配器收到手机的 Sink_Cap, 构造出在其供电范围内最合适的 Source_Cap 发送给手机	
17		手机收到构造的 Source_Cap, 发送包含 Mismatch=0 的 Request
18	适配器收到手机的请求, 经评估在提供 Source_Cap 范围内, 发送 Accept	
19	同意手机的请求以后, 设置反馈, 输出手机请求电压, 并发送 PS_Ready	

附录 B 使用 AnyWatt™功能实现手机直充（恒流模式）

AnyWatt™技术的恒流供电模式，可以让手机电池在恒流充电阶段实现低电压大电流直充(手机电池的预充电和恒压充电阶段，仍然采用传统的充电管理模式)。实现方法为，手机的充电管理模块在检测到手机电池进入恒流充电模式(即电池电压高于预充电电压，例如 3.0V)后，通过 PD 标准协议，要求 AnyWatt™适配器进入恒流供电模式，输出恒定大电流（例如 5A）直接对手机电池进行充电。这时适配器会在手机通过 PD 协议设定的输出电压范围内（例如 3V~5V），尽可能的满足 5A 恒流输出。手机这边的充电管理模块则在实现恒流直充后，持续监控电池电压，在电池电压达到充满电压值，例如 4.35V 后，则重新进行一次 PD 协商，把 5A 恒流输出调整为 4.9A 恒流输出。完成协商后，电池电压会下降到 4.35V 以下。手机充电管理模块继续监控电池电压，当电池电压再次达到 4.35V 时，重新进行 PD 协商，把恒流输出值调整到 4.8A，如此重复，直至恒流输出值小于 100mA，则判定为电池充满，停止充电。具体过程请参照以下图例。

图 B-1 和图 B-2 展示使用 AnyWatt™功能实现手机直充的通信过程。

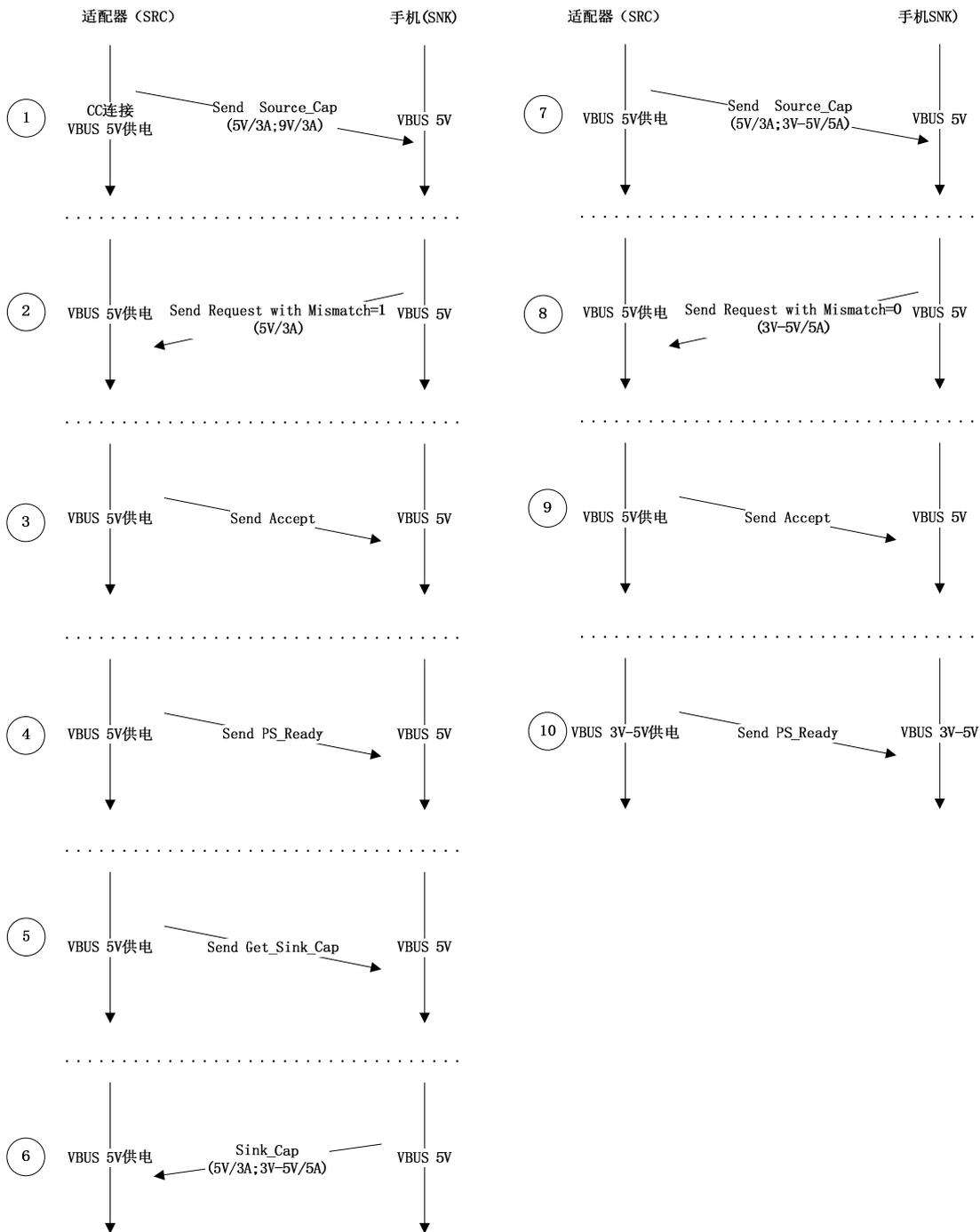


图 B-1 Variable Supply 类型 Power Negotiation

图 B-1 对应的步骤如表 B-1 所示：

表 B-1 Power Negotiation 步骤

步骤	适配器	手机
1	适配器检测到有手机接入，打开 VBUS，提供 5V 电压，并发送自身支持的 Source_Cap	
2		手机收到适配器的 Source_Cap，评估以后，没有完全满足自己的

		Source_Cap, 发送包含 Mismatch=1 的 Request, 请求其中的一档
3	适配器收到手机的请求, 经评估在提供 Source_Cap 范围内, 发送 Accept	
4	同意手机的请求以后, 设置反馈, 输出手机请求电压, 并发送 PS Ready	
5	由于检测到手机的请求中 Mismatch=1, 发送 Get_Sink_Cap, 询问手机所需 Power	
6		收到适配器的 Get_Sink_Cap, 回复 Sink_Cap 告知适配器需要提供的 Power
7	适配器收到手机的 Sink_Cap, 构造出在其供电范围内最合适的 Source_Cap 发送给手机	
8		手机收到新 Source_Cap, 发送包含 Mismatch=0 的 Request
9	适配器收到手机的请求, 经评估在提供 Source_Cap 范围内, 发送 Accept	
10	同意手机的请求以后, 设置反馈, 输出手机请求的电压, 并发送 PS Ready	

可以看出图 B-1 的通信流程展示了适配器的供电类型随手机的需求切换, 即由 Fixed Supply 类型变为 Variable Supply 类型。实际上, 三种供电类型之间可以随 SNK 的需求而任意切换。

图 B-2 展示了手机请求减电流的通信过程。

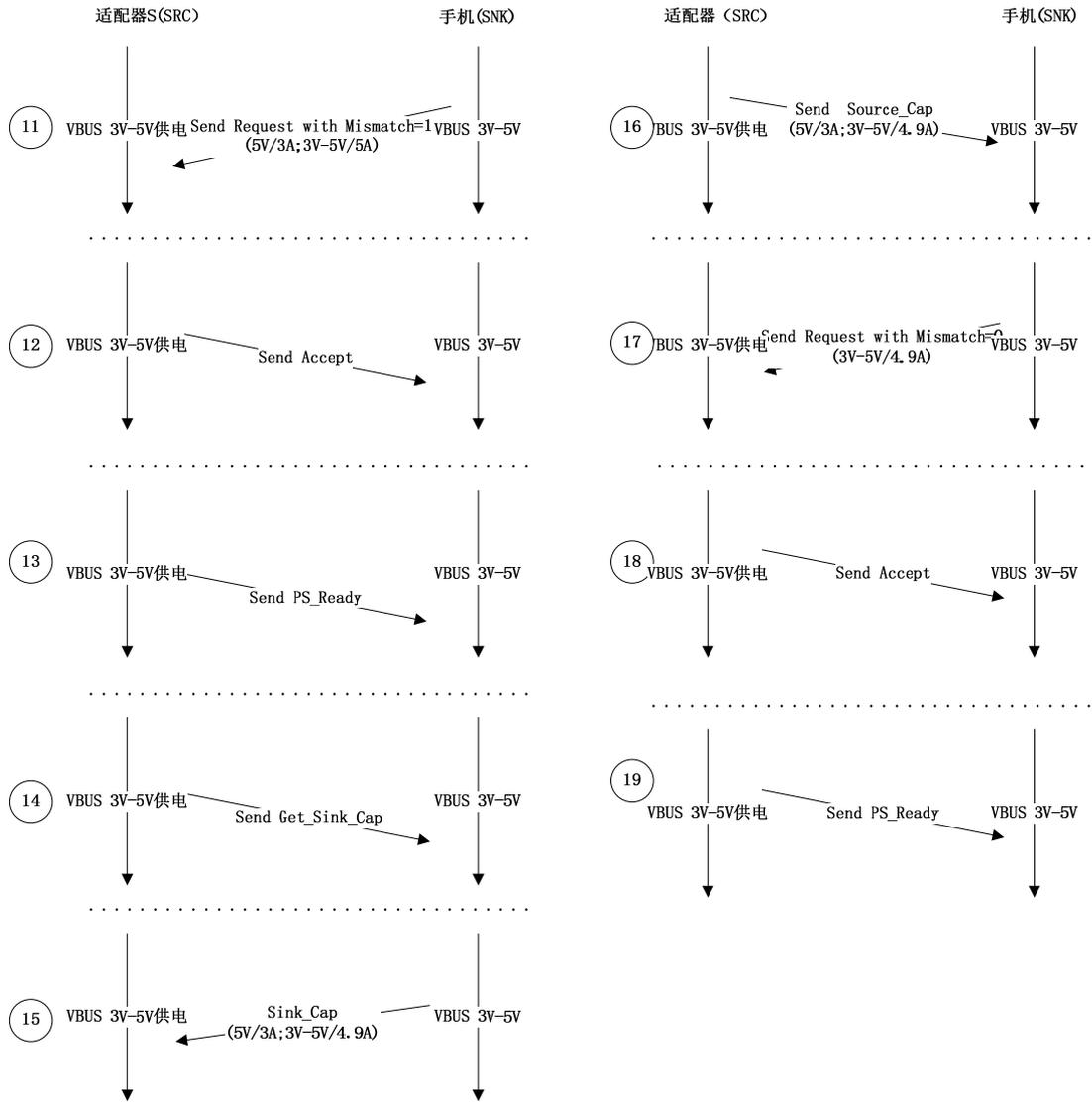


图 B-2 手机发起 Power Negotiation

图 B-2 对应的步骤如表 B-2 所示：

表 B-2 手机发起 Power Negotiation 步骤

步骤	适配器	手机
11		手机的需求发生变化，主动发送包含 Mismatch=1 的 Request，请求的 Power 与上一次 Request 相同，只是 Mismatch 不同
12	适配器收到手机的请求，由于与上一次请求的 Power 一样，立即发送 Accept	
13	由于与上一次请求的 Power 一样，不用设置反馈，立即发送 PS Ready	
14	由于检测到手机的请求中 Mismatch=1，发送 Get_Sink_Cap，询问手机所需 Power	
15		收到适配器的 Get_Sink_Cap，回复

		Sink_Cap 告知适配器需要提供的 Power, 电流值比上一次减少了 100mA
16	适配器收到手机的 Sink_Cap, 构造出在其供电范围内最合适的 Source_Cap 发送给手机	
17		手机收到构造的 Source_Cap, 发送包含 Mismatch=0 的 Request
18	适配器收到手机的请求, 经评估在提供 Source_Cap 范围内, 发送 Accept	
19	同意手机的请求以后, 设置反馈, 输出手机请求的电压, 并发送 PS_Ready	