MEETING 29 AGENDA:

Call to Order UPAMD Power Subgroup meeting ï Paul Panepinto 5pm Pacific 1 Dec 2011

I. Introductions/Attendance
Bob Davis, Edgar Brown, Xia Zhang, Paul Panepinto

II. Approval of 12/01 /2011 Power Subgroup Agenda
Bob Moved to Approve. Edgar Seconded.

III. Presentation of 11/17/2011 Power Subgroup Meeting 28 Minutes
Edgar Moved to Approve. Bob Seconded.

IV. IEEE Call for Patents. See
https://development.standards.ieee.org/myproject/Public/mytools/mob/slideset.pdf

V. Working through use-cases

a) Sink requests info of another connected sink
b) Sink requests info of all other sinks in the network
c) Sink requests info on its Source
d) Sink request info on all other sources in its network
e) Source requests info on individual Sink
f) Source request info on all connected Sinks.
All the above are accomplished by requesting the State Table, which contains the relevant information.

g) Sink changes to source.
This is equivalent to sending a message to the Source that it no longer needs power. That would trigger a renegotiation. At such time of renegotiation, the former source can choose to accept power from the former sink or neither device can want power, which would terminate the negotiation until either side changes its mind via sending a Power Available of Power Request message to renegotiate.

h) Source changes to sink.
Power Request is sent from the Source to the Sink. If the Sink is willing to convert to a Source to supply it power, it will respond with a Power Available message. The former sink, now acting as a Source will send a message to the former Source, now acting as a sink, to set its power to 0, which will initiate a Power Negotiation sequence.

i) Sink requests a change of power.
Accomplished by sending a Power Request Message to its Source.

j) Available power changes from source
Issues the Power Available message to all its connected sinks.

Cases:

a. Still sufficient for all connected sinks
Keep operating unchanged.

b. Insufficient for all connected sinks
Initiate the power negotiation stage that occurs whenever there is a change of available power, using priority and stored available power as clues to which devices remain with power and which may get power terminated. These conditions have already been analyzed in previous meetings reviewing the Power Request and Power Available negotiation sequences, which lead to the development of Application Priority, Request Priority, Stored Available Power registers and related concepts of a smart power supply.

k) Find out all available stored power in the network.
l) Find out which source has the highest available power.
m)Find out which device is consuming the most energy.
n) Get all status from any device in the network.
All these functions can be accomplished by issuing one or repeated State Table requests.

The ability to request the State Table depends upon the Data Transfer mechanism that does 2 things: (a) Enables addressing across devices potentially connected more than one device way, requiring multiple hops to reach it (see Connectivity Diagram), and (b) Enables transferring more than 8 bytes of data, a limitation of CAN-bus, associated with a single request.

There are 3 open issues remaining to solve prior to completing the definition of the Data Transfer mechanism:

- Header definition
- Broadcast messages when the topology may have loops
- How exactly to define messages that use Data Transfer to accomplish the request for State Tables and any other request that might involve multiple hops and/or larger data transfers than 8 bytes, such as vendor names, unique identifiers and such.

VI. New business?

1. Bob proposed modifying the Initialization Message for storing nominal voltage
We agreed to replace the max power byte with a nominal voltage in the Initialization Message as that could cut down on messages in a power

negotiation. The max power information was redundant as it is stored in the Power Request message too.

Bob also proposed we use Status Bit 7 to indicate that the device uses Secure Communications and we agreed to make that change.

We further agreed that the first byte in the first row of the State Table of a device was 0, for port 0, a special port indicating the device itself. That byte, instead of being 0, could be used as the version of the State Table. Having a State Table version is useful, because a device that gets a Request for its State Table, must return its State Table in the format that the requesting device will understand. If a version 1.1 device requests a State Table from a version 3.0 device, it may not understand the format of a version 3.0 State Table. Therefore, every device must make available the knowledge of what version its State Table is.

VII. Adjourn
Motion Approved at 6:41pm Pacific.