UPAMD Operational flow chart 201103221100

This is the more detailed description of the operation detailed bubble diagram.

Starting with the flow chart from the adapter/source port point of view, the first block will cover State1 through State5. From power applied to the adapter to communications power applied to the sink device through the active port.

Start is when power is applied to the adapter/source, either a single port adapter or a multi-port complex adapter power console. As the source power stabilizes, the control processor, or processors, start and complete their boot sequences and power on testing, preparing for operation.

STATE1 – First operation on a port is to determine if there is a voltage already present on the port. This could happen if a cable is attached and the cable is connected to another source capable port in sleep, or standby, mode where the low energy power is maintained. STATE1 is where all terminal, and terminated, operations return to for all sourcing ports.

STATE2 – This state decides on the result of the test for voltage on the power pins. If the voltage is greater than 3V there probably a low energy power source attached. If this >3V is detected the control is passed to STATE300 to communicate with the attached device to determine priority and expected operation where class 3x, 4x, or 5x class devices may be attached and negotiate operations. If power pin voltage is <3V it is assumed not to be present and proceed to STATE3.

STATE3 – First determine if being probed by the port on the other end of the cable. Before enabling the driver of probe power, measure VCANH voltage, if VCANH is active (still determining probing method) goto STATE300 to communicate with the device that is probing. If VCANH is inactive, less than 0.25V, proceed to STATE4.

STATE4 – This state determines if there is another port attached to the output of this port, through an attached cable or a detachable cable. Depending on the type of port attached to the cable, the startup is determined. (temporarily assuming method 2)

```
IF ((CANCurrent > 40ma) && (CANCurrent < 60ma)) {
PortClass = 28h;  //  Set port to be dumb port
Max20watt();  // Goto the 20W max power delivery
}
Else if  ((CANCurrent > 5ma) && (CANCurrent < 40ma)) {
PortClass = 20h;  //  Set port to be Smart port
Continue;  // go on to STATE5
}
Else if  (CANCurrent < 5ma) {
PortClass = 00h;  //  Set port to be no port
GOTO STATE1;  // go on to STATE1
}
```

STATE5 – Start Communications Power.

```
Output_Voltage = LowEnergyVoltage;   //    12V +/- 10%
Output_Current_Limit = LowEnergyCurrentLimit;  //    25ma
EnablePowerOut();   // Start output of 12V @ 25ma max
Delay(500);  // Delay 500 ms for output surge to complete
If (OutputCurrent  > 1.5*OutputCurrentLimit)) {
        DisablePowerOut();  // overcurrent = short circuit
        Goto STATE1; // return to beginning
}
Else If (OutputCurrent  < ( 0.1*OutputCurrentLimit)) {
        DisablePowerOut();  // undercurrent = no valid device attached
        Goto STATE1; // return to beginning
}
Continue;
```

STATE6 – Send Available Power Message (1) with requested message number 2 (Requested Power Message).

Looping will occur in this state while the load device power controller powers up and starts to respond to messages on the CAN bus.

If no ACK bit set; Return to STATE6.   Loop on sending Power Available Message until acknowledged.

If ACK bit set; Message acknowledged, Continue to STATE7.

STATE7 – Wait for Power Request Message.

Delay until the Requested Power message is received. If message number other than Requested Power Message received, goto STATE6 for 5 iterations of request. If Requested Power Message still not received, goto STATE1.

When Requested Power Message received: Goto STATE8

STATE8 – Having received the Power Requested from the load port, this state determines if the requested power and voltage are within the capabilities of this port. By comparing requested power and power available, it is determined if the load port can be operated at requested power. If full requested power is not available but the minimum required power is available, the port resend the power available message showing the maximum available power. The load should respond with a requested power that is equal to the power available. If the load port minimum power is greater than the current available power the source will send an NAK to show it is incapable of meeting the minimum power requirement. The source port will return to STATE1.
If requested power is less than available power, the requested voltage is tested against the available voltage. If the source port is not capable of requested voltage, the source port message restates the available voltage and waits for response from load port. If load port changes requested voltage to match available voltage, the source port sets the output voltage, output current limit and enables the output. If

the RequestedVoltage is within the available voltage range, the AvailableVoltage should be set to RequestedVoltage to signal the load that the requested voltage will be supplied.

```
Enter from STATE7 and STATE13; Exit STATE7 or STATE9
if (RequestedPower <= MaximumAvailablePower && MaxAvailVoltage >= VoltageRequest) {
            CurrentAvailablePower = RequestedPower;
            VoltageOut = VoltageRequest;
            AvailableVoltage == VoltageRequest; // set AvailableVoltage for load to see change.
            EnablePowerOut;
            Goto STATE9;
      }
      Else if (MinimumPower  > MaximumAvailablePower) {
            send NAK;
            send AvailablePowerMessage;
            Goto STATE6;
      }
      Else if (MaxAvailVoltage < VoltageRequest) {
                  send NAK;
                  send AvailablePowerMessage;
                  Goto STATE6;
      }
```

STATE9 – Information exchange. This state collects the information available from the load port and send the source device information to the load port. The use of this information is determined by the receiving device. It provides information needed for diagnostics of fault conditions. This data can be read out by any UPAMD diagnostic port.

```
      Send Message5; /* Vendor Identification */
      Request Message5;
      Send Message6; /* Model Number */
      Request Message6;
      Send Message7; /* Serial Number */
      Request Message7;
      Send Message8; /* Unique Identifier */
      Request Message8;
      Send Message9; /* Manufactured Date */
      Request Message9;
      Send Message10; /* Software Version */
      Request Message10;
      Send Message11; /* Firmware Version */
      Request Message11;
      Request Message13; /* Timestamp NTP*/
      Send Message14; /* UPAMD Standard Version */
      Request Message14;
      Request Message16; /* Load Device Power State */
      Goto STATE 10;
```

STATE10 – Nominal power delivery STATE. Responds to disconnect and power change requests.

System stays in this state and delivers power until requested otherwise or the connection is lost. This state continuously loops through a series of tests for changes of conditions that need attention. The loop is broken by a disconnect of the load communications connection with the source port. This disconnect causes a reduction in voltage to zero and the reduction of current to zero to get the power below the low energy state before the power pins separate and have the ability to create a dangerous spark. This returns the state machine to STATE1 to restart the connection if available.
While looping in the state the tests are made for overcurrent condition, monitoring at 110% of max current, which will then cause a complete shutdown and control returned to STATE1.

Continuously monitored during this looping state is the CurrentPowerRequest from the load for any changes in the requested load. If a change in Requested Power is made, the source determines if it can accommodate the change and respond with a equivalent change in the PowerAvailable message or a NAK to disapprove the power change. During this process the source may need to reallocate power among the attached ports.

During this looping state the source may need to change the available power message and ask the sink port to accept the new available power. If the sink port cannot accept the requested change it can respond with a NAK. Priority will be used to resolve conflicts in power allocation.

While in the looping state the source port also monitors for request of sleep or standby power and for the request to remove all power. These are signaled by PowerRequested values below the 10W minimum with a value of 00 being request for disconnect and 01 being a request to go to low energy state of 12V and 25ma maximum current. For low energy requests, the software will stay in the current state and loop while supplying the low energy power until a new power request is receive to either disconnect or return to the higher working power level. For the 00 request for zero power, the port will go to 0 Volts and 0 current expecting a disconnect and will return to STATE1.

```
    While (!Disconnect) {  // Disconnect detected by no current in CANH line(method 2)
        If (OutputCurrent  >  (1.1*CurrentLimit)) {  //  Over current condition possible short
            BREAK;  //  Kill output and go back to STATE1
        }
        Else If (CurrentPowerRequest  !=  CurrentAvailablePower) {
            NegotiateSourcePower();  // request changes in device power per source need
            Continue;  // Continue in While loop.
        }
        Else If (CurrentPowerRequest  !=  LastPowerRequest) {
            NegotiateSinkPower();  // determine new power demand from load device
            LastPowerRequest = CurrentPowerRequest; // update last power request
            Continue;  // Continue in While loop.
        }
        Else if (VoltageRequest != CurrentVoltage) {  // voltage request from device
            NegotiateVotage();  // If capable,  meet device need – printer/workstation etc.
            Continue;  // Continue in While loop.
        }
/* if we define the power requests for less than 10 watts, the low end of UPAMD in several ways it can
be used to request low or no power.   Setting PowerRequest  = 0 requests power be removed.  Setting
PowerRequest = 1 requests return to standby, low energy mode.  */
        Else if (CurrentPowerRequest = 0) {  // Device requesting turnoff
```

```
                Break;  // end loop and kill output  then to STATE1
            }
            Else if (CurrentPowerRequest = 1) {  // Device requesting sleep state
                OutputCurrent = LowEnergyCurrent;  // Set output current limit to 25ma
                OutputVoltage = LowEnergyVoltage;  // Set output voltage to 12V
                Continue;  // loop in current while loop state.
            }
        }
        OutputCurrent = 0;  // Set output current to zero
        OutputVoltage = 0;  // set output voltage to zero
        DisableOutput();  // Kill the output
        Goto STATE1;  // goto starting state.
```

STATE100 – 20W mode power = 20V at 1A max.

This state is entered by the detection of a non-communicating port on the load device. The source port maintains the status of the connection of the communications lines at all times. The OutputVoltage is set to 20V and the current limit is set to 1A. If current exceeds current limit by 10% all power is removed and control is returned to STATE1. If disconnect occurs, all power is removed and control is returned to STATE1.

```
        OutputVoltage = LowPowerVoltage;  // Set voltage to 20V
        CurrentLimit = LowPowerCurrent;  // Set current limit to 1A
        EnableOutput();  // Turn on power
        While (!Disconnect ) {  // Disconnect detected by no current in CANH line(method 2)
            If (OutputCurrent  >  (1.1*CurrentLimit)) {  //  Over current condition possible short
                BREAK;  //  Kill output and go back to STATE1
            }
        }
        OutputCurrent = 0;  // Set output current to zero
        OutputVoltage = 0;  // set output voltage to zero
        DisableOutput();  // Kill the output
        Goto STATE1;  // goto starting state.
```

Load Device Ports

STATE200 – This starts the operation of the power sink. It starts with receiving enough power from the low energy power systems to provide operational power for the microcontroller. The assumption is the 12V input to the low voltage power supply will generate the 2.5 to 3.3V for controller operation. This state may be entered from a power direction change or from a standby/idle low power operation.

If a differential CAN bus is being used, the line that is pulled high is designated as the CANH line and the differential receiver is switched if needed to correct polarization along with the driving transistors. If a single wire CAN bus is used, this is not necessary.

```
        Device Boot: // sink controller comes out of reset or boot state.
```

```
    If (DifferentialBus) { // variable for this discussion only
    PickHighLine; // determine which is the highest voltage
    AssignCANPolarity; // orient the differential receiver and driver
    }
    SendRequestedPowerMessage; // send requested power to source
    If (NAK) {  // if NAK – negative acknowledge – no communications
    Goto STATE200;  // return to start
    }
    Else if (ACK) { // Working CAN bus – continue to next state
    Goto STATE201; // or Continue
    }
```

STATE201: Read the AvailablePower message from source port and decide if there is sufficient power to start up. Minimum power may be keep alive power and allow for very restrictive operations.

```
    ReadAvailablePowerMessage;
    If (CurrentAvailablePower >= MinimumOperationalPower && AvailableVoltage >=
MinimumVoltage) { // got enough power and voltage to operate
    Goto STATE203; // sufficient power for startup
    }
    Else if (CurrentAvailablePower < MinimumOperationalPower || AvailableVoltage <
MinimumVoltage) { // power below minimum power or voltage.
    Send NAK; // send negative acknowledge to source
    Send RequestedPowerMessage;  // Resend the Requested Power message to source
    Goto STATE200;  // Go back to start
    }
```

STATE203: This state validates the available power. If device requests a higher voltage and power, it asks for the power and voltage. If source can provide the additional voltage or power, it acknowledges the request and supplies the additional voltage and/or power. If the source cannot accommodate the request is send a NAK and device decides if it will use the available power or NAK the source power offer and, fault and return to STATE200. If power agreement is made, this state enables the power converters to either the default voltage state or the agreed upon power supply voltage.

```
    If (AvailableVoltage >= MinimumOperational) { // Test for good voltage
        EnableLoadPower;  // turn on the power supply
    }
```

STATE204 – Retrieve information on source of power. If not already started by the adapter, request available information from the source.

```
    Request Vendor ID;
    Request Model Number;
    Request Serial Number;
    Request Software Version;
    Request Firmware Version;
    Request EUI;
    Request Manufacturer Date Code;
    Validate device as possible;
```

STATE205 – This state starts the main loop of the power delivery for the sink device.   During this loop we look for changes and continuously verify conditions. Conditions are monitored by the source and the sink. Changes in power request are determined with mutual agreement required. Reduction of power usage to 0watts or standby/idle power of low energy only modes are determined and monitored. This state is broken by request for power removal. Standby power remains in this loop as a request for full power may be issued. If source priority is raised to be dominant, power flow will be determined by the source.

```
    STATE205:
    If (MyPriority <= SourcePriority) { // Check for priority change
        GoTo STATE207;  // Source became higher priorty
    }
    Else If (MyPriority > SourcePriority) {  // Sink still higher priority
        Continue;
    }
    STATE206:
    If (InternalZeroPowerRequest) { // If device request power turnoff
        RequestedPower == 0;   // request power off
        Send RequestedPowerMessage;  // Tell the source
        Wait(Acknowledge);  //  Wait for acknowledgement through available power
        If (AvailablePower = 0) {  // message received and no power avialable
        Goto STATE200;     // Back to the head of the sink operation
        }
        Else {Continue;} // lower power usage and repeat message.
    }
    If (InternalStandByPowerRequest) { // If device requests Standby/Sleep power
        RequestedPower = 1;  // Request reduction to standby/sleep power
        Send RequestedPowerMessage;  // Tell source
        Wait(Acknowledge);  // Wait for next AvailablePowerMessage
        If (AvailablePower = 1) {  // acknowledgement received
        Continue;  // Stay in loop for later restart of power.
        }

        Else { Goto STATE205} // back to top of monitor loop
    }
    STATE208:
    If ((DeviceInternalPowerRequest > CurrentAvailablePower) && (DeviceInternalPowerRequest <=
MaximumAvailablePower)) {  // Check for need for more power
        RequestedPower = DeviceInternalPowerRequest;  // increase source power requested
        Send RequestedPowerMessage;  // tell the source
        Wait (Acknowledge);  // Wait for the answer
        If (AvailablePower == RequestedPower) {   // Check answer
            RaiseInternalPower;  // increase power maximum power usage
        }
        Else If (AvailablePower < RequestedPower) { // check for request not granted
            DecideOnPriorityChange;  // decision on if current level usable
            If (PriorityIncreaseRequest) {  // increase priority if needed to increase power.
                DevicePriority++;  // increase the priority by one.
            }
            Continue;
        }
    If (CurrentAvailablePower < RequestedPower) { // Check for source request for lower power use
        LowerPowerRequest;  // determine if lower power is acceptable.
        If (LowerPowerRequestAccept) {  // test result of above request
        RequestedPower = CurrentAvailablePower;  // if acceptable – lower requested power.
        Send RequestedPowerMessage;  // send the message
        }
        Else {  // if not acceptable
        Send NAK:  // say no and retain requested power.
        }
    }
    Goto STATE205;
```

STATE207 – Change in priority. This state determines the change in priority to source priority driven. During this state, the determination of the role of the sink is determined. It may just be a reduction in available power due to other priorities in the source or the source requesting a role change. Role change is indicated by change in device type from the source.

```
    STATE207:  TBD
```

STATE300 – Bidirectional power messaging and exchange. Input is from STATE2/3 when a bidirectional sink/source is detected by the power class. First devices sends a RequestPowerMessage and AvailablePowerMessage to state it conditions and request both the AvailablePowerMessage and RequestedPowerMessage from other end of the link.
It then compares priorities and proceeds to STATE301 if it has a less than or equal priority to the connected device.
If the comparison of priority determines this device has the highest priority, it proceeds to STATE302 an act as a sink to fill storage or standby.

```
    STATE300:  To be detailed.
```