libusbctrl

Supports USB detection and control functionality to enable USB detection and control from application. This ensures that the USB port works as expected before using peripherals and capabilities that use the USB port.

- Functionality
- Sample Control Flow

Functionality

- Support for USB hot plug detection & Control from service manager module
- Notifies application via Service Manager when device is connected to USB port, including device type (and make / model if available).
- Provides software control via Service Manager to enable/disable USB port.
- Allows multiple USB devices to connect via USB hub.
- Upon initialization, libusbctrl scan the existing devices and start monitor hotplug events.

Sample Control Flow

- usbctl is loaded. It initializes data structures to store callbacks.
- App calls registerCallback()
- App calls init()
  - usbctrl initializes internal data structures.
  - usbctrl launches hotplug monitoring task.
- usbctl enumerates all usb_interface device types and assigns a unique number to each interface.
- Each interface is entered into a list of structs. Each struct has the unique identifier and the interface-specifier
- usbctrl launches callback for each interface with "add" event.
- Callback is allowed to internally call other usbctrl APIs to get properties.
- When a getProperty() call is received, usbctrl compares unique identifier against its internal data structures and validates.
  - if unique identifier is valid,
  - if property is defined by interface, return that.
  - if property is not defined by interface, iterate up the USB hierarchy to find one that does and return the value.
- App calls term()
- usbctrl stops monitoring of hotplug events.
- usbctrl clears data structures storing callbacks and usb_interfaces.