rf IDEAS

PcProxAPI-Web SDK Document

Trust begins here.™

Supporting OS with browser compatibility.

ChromeOS	Linux	MacOS
Google Chrome	Google Chrome	Google Chrome
	Microsoft Edge	Microsoft Edge
	Opera	

Note: Chromium must be updated to version 115 and above.

Limitation: There is an issue for communication of reader on Windows OS. Hence, PcProxAPI-WebSDK is currently not supported for Windows OS.

connect

async connect ()

DESCRIPTION

A JavaScript interface to connect (). It is an async function. It will open a popup to select the USB device and on click of connect it will return true.

PARAMETERS

None

RETURNS

True= Success / False = Failure

disconnect

async disconnect ()

DESCRIPTION

A JavaScript interface to disconnect (). It is an async function. It will disconnect the reader.

PARAMETERS

None

RETURNS

True = Success / False = Failure

getpartNumber

async getPartNumber(

DESCRIPTION

A JavaScript interface to getPartNumber(). It is an async function. Read the device part number string. This is a 24 (MAXPRODUCTNAMESZ) character string such as "MS3-00M1AKU" and null-terminated.

PARAMETERS

None

RETURNS

Part number= Success / False = Failure

writeCFG

async writeCFG()

DESCRIPTION

A JavaScript interface to writeCFG(). It is an async function. A call to this function writes all configuration information in the library memory space to the device for non-volatile storage. Any changed parameters take effect immediately after writeCFG().

PARAMETERS

None

RETURNS

True= Success / False = Failure

readCfg

async readCFG()

DESCRIPTION

A JavaScript interface to readCFG(). It is an async function. A call to this function pulls the device configuration information into the library memory space to be manipulated by the Get*() and Set*() functions. After altering the data the user must call writeCFG()to write the changes back to the device so they can take effect.

PARAMETERS

None

RETURNS

True= Success / False = Failure

getActConfig

async getActConfig()

DESCRIPTION

A JavaScript interface to getActConfig(). It is an async function. Get the active configuration (0..N) of the pcProx Plus. For devices that only have one configuration, this will return 0. All devices have one configuration, so zero is always valid.

PARAMETERS

None

RETURNS

ActConfig = Success / -1= Failure

setActConfig

async setActConfig(ConfigurationIndex)

DESCRIPTION

A JavaScript interface to setActConfig(). It is an async function. Set the active configuration (0..N) of the pcProx Plus device.

PARAMETERS

ConfigurationIndex

RETURNS

True= Success / False = Failure

getActiveID32

async getActiveID32()

DESCRIPTION

A JavaScript interface to getActiveID32(). It is an async function. It will return the reader Id. This is the 32-byte version of getActiveID() for FIP-201 and CHUID readers to get the card ID with up to 255 bits of data. Cards with larger ID's than 64 bits will be truncated with getActiveID(), so this function is recommended instead.

PARAMETERS

None

RETURNS

Number of bits read 0..255

getActiveID

async getActiveID()

DESCRIPTION

A JavaScript interface to getActiveID(). It is an async function. Reads the active data from the card" presently on the reader". The Least significant bit is Buf[0] bit 0. It is recommended not toc all this faster than 250msec, or about twice the data hold time of the active card. Call sooner than 250msec will not be sent to the reader but will return cached data.

PARAMETERS

None

RETURNS

Number of bits read 0..255

getCfgFlags_bHaltKBSnd

async getCfgFlags_bHaltKBSnd(configurationIndex)

DESCRIPTION

A JavaScript interface to getCfgFlags_bHaltKBSnd(). It is an async function. Halt Keyboard Sending of data. When set, keystroking data will not be key out, all pcProx information must be read using the API, it will get bHaltKBSnd flag.

PARAMETERS

configurationIndex

RETURNS

A Boolean for bHaltKBSnd= Success / Sets the error code in lastLibErr = Failure

setCfgFlags_bHaltKBSnd

async setCfgFlags_bHaltKBSnd(value: number, configuration: number)

DESCRIPTION

A JavaScript interface to setCfgFlags_bHaltKBSnd(). It is an async function. It will set bHaltKBSnd flag. WriteCFg needs to be called in case of we need to write on reader from library memory.

PARAMETERS

bHaltKBSndvalue, configurationIndex

RETURNS

True= Success / Sets the error code in lastLibErr = Failure

getIDBitCnts_iTrailParityBitCnt

async getIDBitCnts_iTrailParityBitCnt(configurationIndex)

DESCRIPTION

A JavaScript interface to getIDBitCnts_iTrailParityBitCnt(). It is an async function. You must call ReadCfg() first. This calls GetIDBitCnts() internally. It will get TrailParityBitCnt

configurationIndex

RETURNS

A Boolean for iTrailParityBitCnt= Success / Sets the error code in lastLibErr = Failure

setIDBitCnts_iTrailParityBitCnt

async setIDBitCnts_iTrailParityBitCnt(value: number, configuration: number)

DESCRIPTION

A JavaScript interface to setIDBitCnts_iTrailParityBitCnt(). It is an async function. It will set iTrailParityBitCnt flag.

PARAMETERS

iTrailParityBitCntvalue, configurationIndex

RETURNS

True= Success / Sets the error code in lastLibErr = Failure

getIDBitCnts_iLeadParityBitCnt

async getIDBitCnts_iLeadParityBitCnt(configurationIndex)

DESCRIPTION

A JavaScript interface to getIDBitCnts_iLeadParityBitCnt(). It is an async function. You must call ReadCfg() first. This calls GetIDBitCnts() internally. It will get LeadParityBitCnt.

PARAMETERS

configurationIndex

RETURNS

A Boolean for iLeadParityBitCnt= Success / Sets the error code in lastLibErr = Failure

setIDBitCnts_iLeadParityBitCnt

async setIDBitCnts_iLeadParityBitCnt(value: number, configuration: number)

DESCRIPTION

A JavaScript interface to setIDBitCnts_iLeadParityBitCnt(). It is an async function. It will set iLeadParityBitCnt flag.

PARAMETERS

iLeadParityBitCntvalue, configurationIndex

RETURNS

True= Success / Sets the error code in lastLibErr = Failure

GetLastLibErr

async getLastLibErr()

DESCRIPTION

A JavaScript interface to getLastLibErr(). It is an async function. Returns the last library error code (see Library Error Codes) for the active device. The last error code is valid until another library call is made. This does not reset the last library error code. When a function returns FALSE and has failed, it is good practice to call this function to check the error code.

PARAMETERS

None

RETURNS

Error code= Success

Error code table:

Error Description	Error code in Hex	Error code in number
USBConnect	0x00010000	65536
FailedReadCfgLoop	0x00011000	69632
ReadCfg	0x00020000	131072
WriteCfg	0x00040000	262144
ReadCfgNotCalled	0x00100002	1048578
CouldNotOpenHID	0x00000003	3
DeviceNotOpened	0x00000200	512
GetNULLmodule	0x00000102	258
SetNULLmodule	0x00000202	514
ModuleCallFail	0x00000103	259