# Jubiter-SDK-CXX Read Me

Contents

[Jubiter-SDK-CXX Read Me 1](#_Toc14220)

[1 Interface for enumerating HID device 1](#_Toc23459)

[2 Interface for connecting to devices 2](#_Toc19451)

[3 Interface for disconnecting connected devices 2](#_Toc29835)

[4 Interface for getting device properties 2](#_Toc8130)

[5 Interface for getting device certificate 3](#_Toc14939)

[6 Interface for sending an APDU directly 3](#_Toc13574)

[7 Interface for checking device initialization 3](#_Toc18452)

[8 Interface for checking BootLoader 4](#_Toc7790)

[9 Interface for enumerating applet 4](#_Toc25282)

[10 Interface for getting applet version 4](#_Toc7633)

[11 Interface for creating a context to operate BTC series coins 4](#_Toc31220)

[12 Interface for destroying a BTC series context 5](#_Toc3401)

[13 Interface for getting HD Node public key 5](#_Toc11516)

[14 Interface for getting the public key of the current context 6](#_Toc3582)

[15 Interface for getting one address 6](#_Toc791)

[16 Interface for setting address 6](#_Toc4643)

[17 Interface for setting the unit of BTC shown on the device 7](#_Toc13804)

[18 Interface for signing one BTC transaction 7](#_Toc17399)

[19 Interface for generating the output of an USDT transaction 8](#_Toc10509)

## Interface for enumerating HID device

JUB\_RV JUB\_ListDeviceHid(OUT JUB\_UINT16 deviceIDs[MAX\_DEVICE]);

|  |  |
| --- | --- |
| Description | List all plugged JuBiter Blade deivces |
| IN | Null |
| OUT | deviceIDs: Virtual deivce IDs are one to one correspondence with JuBiter devices. This function should be recalled each time the JuBiter device changes. |
| Additional Information | Available in Windows/Linux/Mac |

## Interface for connecting to devices

JUB\_RV JUB\_ConnetDeviceHid(IN JUB\_UINT16 deviceID);

|  |  |
| --- | --- |
| Description | Connect to JuBiter Blade devices |
| IN | deviceID: the device ID returned from JUB\_ListDeviceHid |
| OUT | Null |
| Additional Information | Available in Windows/Linux/Mac |

## Interface for disconnecting connected devices

JUB\_RV JUB\_DisconnetDeviceHid(IN JUB\_UINT16 deviceID);

|  |  |
| --- | --- |
| Description | Disconnect connected JuBiter Blade devices |
| IN | deviceID : the device ID returned from JUB\_ListDeviceHid |
| OUT | Null |
| Additional Information | Available in Windows/Linux/Mac |

## Interface for getting device properties

JUB\_RV JUB\_GetDeviceInfo(IN JUB\_UINT16 deviceID, OUT JUB\_DEVICE\_INFO& info);

|  |  |
| --- | --- |
| Description | Get the hardware properties of one JuBiter Blade device |
| IN | deviceID : the device ID returned from JUB\_ListDeviceHid |
| OUT | info: property structure of JuBiter device, including:  Label: Device label, which is set when JuBiter device is produced.  Sn: The serial number of JuBiter device, which is set when JuBiter device is produced.  pin\_retry: Current password retries  pin\_max\_retry: Password maximum retries  ble\_version: Bluetooth firmware version  firmware\_version: Kernel firmware version |
| Additional Information | Read Only |

## Interface for getting device certificate

JUB\_RV JUB\_GetDeviceCert(IN JUB\_UINT16 deviceID, OUT JUB\_CHAR\_PTR\_PTR cert);

|  |  |
| --- | --- |
| Description | Get the device certificate of one JuBiter Blade |
| IN | deviceID: the device ID returned from JUB\_ListDeviceHid |
| OUT | cert: This is the origin device certificate signed by Jubiter root certificate, which can be verified by the Jubiter server for authenticity of the device. |
| Additional Information | Device certificate format conforms to GP specification. |

## Interface for sending an APDU directly

JUB\_RV JUB\_SendOneApdu(IN JUB\_UINT16 deviceID, IN JUB\_CHAR\_PTR apdu , OUT JUB\_CHAR\_PTR\_PTR response);

|  |  |
| --- | --- |
| Description | Send an APDU command directly |
| IN | deviceID: the device ID returned from JUB\_ListDeviceHid  apdu: APDU command in Hex String |
| OUT | Response: response in Hex String |
| Additional Information |  |

## Interface for checking device initialization

JUB\_ENUM\_BOOL JUB\_IsInitialize(IN JUB\_UINT16 deviceID);

|  |  |
| --- | --- |
| Description | Check if the JuBiter device has generated mnemonics |
| IN | deviceID: the device ID returned from JUB\_ListDeviceHid |
| OUT | return: true or false |
| Additional Information | For security reasons, the process of generating mnemonics is done entirely on the JuBiter Blade device, and the software is not involved in all the processes of generating mnemonics. This interface is used to determine if the hardware has gone through this process. The generation of mnemonics follows the BIP39 specification. |

## Interface for checking BootLoader

JUB\_ENUM\_BOOL JUB\_IsBootLoader(IN JUB\_UINT16 deviceID);

|  |  |
| --- | --- |
| Description | Check if the hardware is in BootLoader mode. |
| IN | deviceID: the device ID returned from JUB\_ListDeviceHid |
| OUT | return: true or false |
| Additional Information |  |

## Interface for enumerating applet

JUB\_RV JUB\_EnumApplets(IN JUB\_UINT16 deviceID,OUT JUB\_CHAR\_PTR\_PTR appList);

|  |  |
| --- | --- |
| Description | Enumerate the applets already installed on the current JuBiter device |
| IN | deviceID: the device ID returned from JUB\_ListDeviceHid |
| OUT | appList: list all applet IDs separated by spaces |
| Additional Information | The Jubiter Blade device uses the Java card architecture, which each applet corresponds to one series of cryptocurrencies. |

## Interface for getting applet version

JUB\_RV JUB\_GetAppletVersion(IN JUB\_UINT16 deviceID,IN JUB\_CHAR\_PTR appID,OUT JUB\_CHAR\_PTR\_PTR version);

|  |  |
| --- | --- |
| Description | Get applet version |
| IN | deviceID: the device ID returned from JUB\_ListDeviceHid  appID: the applet ID returned from JUB\_EnumApplets |
| OUT | version: version of applet |
| Additional Information |  |

## Interface for creating a context to operate BTC series coins

JUB\_RV JUB\_CreateContextBTC(IN CONTEXT\_CONFIG\_BTC cfg , IN JUB\_UINT16 deviceID, OUT JUB\_UINT16\* contextID);

|  |  |
| --- | --- |
| Description | Create a context for operating the BTC series coins for subsequent BTC related operations |
| IN | deviceID: the device ID returned from JUB\_ListDeviceHid  cfg : config for creating context  cointype: The Enum type of coins, detailed information in Jub\_SDK.h  main\_path: BIP44 main Path. It is recommended to detail main path at least to the account level.  transtype: The Enum type of transactions, detailed information in Jub\_SDK.h |
| OUT | contextID: Generate a context ID |
| Additional Information | Recently, this interface is available for BTC, LTC, BCH, and USDT. |

## Interface for destroying a BTC series context

JUB\_RV JUB\_ClearContext(IN JUB\_UINT16 contextID);

|  |  |
| --- | --- |
| Description | Destroy the operation context of a BTC series coin |
| IN | contextID: the context ID returned from CreateContextBTC |
| OUT | Null |
| Additional Information |  |

## Interface for getting HDNode public key

JUB\_RV JUB\_GetHDNodeBTC(IN JUB\_UINT16 contextID, IN BIP32\_Path path,OUT JUB\_CHAR\_PTR\_PTR xpub);

|  |  |
| --- | --- |
| Description | Get a HDNode public key |
| IN | contextID: the context ID returned from CreateContextBTC  path: the last two levels of bip44. The change level could only be 0 or 1. |
| OUT | Xpub: public key in XPUB format, includes chaincode, fingerprint and so on. |
| Additional Information |  |

## Interface for getting the public key of the current context

JUB\_RV JUB\_GetMainHDNodeBTC(IN JUB\_UINT16 contextID, OUT JUB\_CHAR\_PTR\_PTR xpub);

|  |  |
| --- | --- |
| Description | Get the public key of the current context |
| IN | contextID: the context ID returned from CreateContextBTC |
| OUT | Xpub: public key in XPUB format, includes chaincode, fingerprint and so on. |
| Additional Information | Get the xpub public key of main\_path specified by cfg of JUB\_CreateContextBTC. If main\_path is specified to account level, the ordinary subkeys can be derived from this xpub, following the harden subkeys generated by JuBiter device, which can effectively reduce hardware and software communication. |

## Interface for getting one address

JUB\_RV JUB\_GetAddressBTC(IN JUB\_UINT16 contextID, IN BIP32\_Path path, IN JUB\_ENUM\_BOOL bshow, OUT JUB\_CHAR\_PTR\_PTR address);

|  |  |
| --- | --- |
| Description | Get one address of a BTC series coin |
| IN | contextID: the context ID returned from CreateContextBTC  path: the last two levels of bip44. The change level could only be 0 or 1.  bshow: Displayed on the device screen or not |
| OUT | address: address in Base58 |
| Additional Information | This interface can be used by the user to confirm the payment address on Jubiter, in case that the screen is displayed incorrectly after the software is hacked. When the device displays the address, this interface is blocked. When the user presses the confirmation button on the device, the interface returns. |

## Interface for setting address

JUB\_RV JUB\_SetMyAddressBTC(IN JUB\_UINT16 contextID, IN BIP32\_Path path, OUT JUB\_CHAR\_PTR\_PTR address);

|  |  |
| --- | --- |
| Description | Set an address |
| IN | contextID: the context ID returned from CreateContextBTC  path: the last two levels of bip44. The change level could only be 0 or 1. |
| OUT | address: the set address with Base58 format |
| Additional Information | The Jubiter device allows users to set up a transaction address. It is safe and convenient to display the text and QR code of the address on the Jubiter Blade device without connecting to any other equipment.  This interface requires a verification PIN code. |

## Interface for setting the unit of BTC shown on the device

JUB\_RV JUB\_SetUnitBTC(IN JUB\_UINT16 contextID, IN JUB\_BTC\_UNIT\_TYPE unit);

|  |  |
| --- | --- |
| Description | Set the unit of BTC shown on the device during transactions |
| IN | contextID: the context ID returned from CreateContextBTC  unit: unit of Enum, detailed information in Jub\_SDK.h |
| OUT | Null |
| Additional Information | mBTC is the default |

## Interface for signing one BTC transaction

JUB\_RV JUB\_SignTransactionBTC(IN JUB\_UINT16 contextID ,IN INPUT\_BTC inputs[], IN JUB\_UINT16 iCount, IN OUTPUT\_BTC outputs[], IN JUB\_UINT16 oCount, IN JUB\_UINT32 locktime, OUT JUB\_CHAR\_PTR\_PTR raw);

|  |  |
| --- | --- |
| Description | Sign a BTC transaction |
| IN | contextID: the context ID returned from CreateContextBTC  inputs: a list of inputs, with detailed information in Jub\_SDK.h  iCount: the number of inputs  outputs: a list of outputs, with detailed information in Jub\_SDK.h  oCount: the number of outputs  locktime: the locktime of the signed transaction |
| OUT | raw: This is the signed transactions that can be used directly for broadcasting. If the user cancels the transaction, it returns null\_ptr. |
| Additional Information | This interface is blocked when the JuBiter device displays transaction information for user confirmation.  When the Jubiter device signs the transaction, it would verify whether the specified change address in the outputs is the address set in the device. If not, the device would report an error. If correct, the amount of this output will not be displayed in the transfer amount of the transaction information. It is safe, clear and correct to show the user's real transaction amount.  For security reasons, the Jubiter device would only use the hash\_all method for signing a transaction. |

## Interface for generating the output of an USDT transaction

JUB\_RV JUB\_BuildUSDTOutputs(IN JUB\_UINT16 contextID, IN JUB\_CHAR\_PTR USDT\_to, IN JUB\_UINT64 amount, OUT OUTPUT\_BTC outputs[2]);

|  |  |
| --- | --- |
| Description | Generate outputs that conform to the onmi specification |
| IN | contextID: the context ID returned from CreateContextBTC  USDT\_to: the receiving USDT address of the transaction  Amount: the amount of the transaction |
| OUT | Outputs: outputs for JUB\_SignTransactionBTC |
| Additional Information | Auxiliary interface, this interface does not need to call the JuBiter device. |