

Windows SDK

This is an SDK for Windows to control DotPad devices (such as braille displays). The SDK is provided as a DLL and must be used with the header files.

Table of Contents

1. SDK Configuration File
2. Loading Order
3. Enum
4. Function

- SDK Configuration File

File	Description
DotPadSDK.dll	<ul style="list-style-type: none">• SDK library• Place in the same directory as the executable file• Used via dynamic loading
DotSDKAPI.h	<ul style="list-style-type: none">• API declarations, enums, function pointer type definitions

- Loading Order

1. Load the DLL with ``LoadLibrary("DotPadSDK.dll")``
2. Obtain each API's address using ``GetProcAddress(hDll, "function_name")`` and store it in a function pointer
3. Before use, verify the function pointer is not ``NULL``, then call
4. Unload the DLL with ``FreeLibrary`` when the app terminates (recommended to first disconnect the connected device using ``DOT_PAD_DISCONNECT``)

- Enum

export	Description	value
DOT_DATA_CODE	Indicates the type of event/response occurring during communication with the device.	<ul style="list-style-type: none">• DOT_DATA_CODE_CONNECTED : Connection successful• DOT_DATA_CODE_DISCONNECTED : Connection disconnected• DOT_DATA_CODE_DEVICE_NAME : Device name lookup result

		<ul style="list-style-type: none"> • DOT_DATA_CODE_DEVICE_FW_VERSION : Firmware version lookup result • DOT_DATA_CODE_DEVICE_HW_VERSION : Hardware version lookup result • Others : Refer to the 'DOT_DATA_CODE' enum in 'DotSDKAPI.h'
DOT_KEY_CODE	Indicates the type of key event sent by the device.	<ul style="list-style-type: none"> • DOT_KEY_CODE_FUNCTION1 : Function 1 key input • DOT_KEY_CODE_FUNCTION2 : Function 2 key input • DOT_KEY_CODE_FUNCTION3 : Function 3 key input • DOT_KEY_CODE_FUNCTION4 : Function 4 key input • DOT_KEY_CODE_FUNCTION12: Function 1 key and Function 2 key input • DOT_KEY_CODE_FUNCTION13: Function 1 key and Function 3 key input • DOT_KEY_CODE_FUNCTION14: Function 1 key and Function 4 key input • DOT_KEY_CODE_FUNCTION23 : Function 2 and 3 keys pressed • DOT_KEY_CODE_FUNCTION24 : Function 2 and 4 keys pressed • DOT_KEY_CODE_FUNCTION34 : Function 3 and 4 keys pressed • DOT_KEY_CODE_ELSE : Input of key combinations not defined in KeyCodes • DOT_KEY_CODE_PANNING_ALL : Input of Panning Left and Right keys • DOT_KEY_CODE_PANNING_LEFT : Input for Panning Left key • DOT_KEY_CODE_PANNING_RIGHT : Input for Panning Right key • DOT_KEY_CODE_LPF1 : Input for Panning Left key and Function 1 key • DOT_KEY_CODE_RPF4 : Input for Panning Right key and Function 4 key

- Function

4.1 Device Connection and Disconnection

Function Name	Description
---------------	-------------

DOT_PAD_BLE_SCAN(void(CALLBACK* cb)(const wchar_t*))	<ul style="list-style-type: none"> • Start BLE scan • Call `callback(const wchar_t* deviceName)` when discovered
DOT_PAD_BLE_SCAN_STOP()	<ul style="list-style-type: none"> • Stops BLE scanning
DOT_PAD_USB_SCAN(void(CALLBACK* cb)(const wchar_t*))	<ul style="list-style-type: none"> • Start USB scan • Call `callback(const wchar_t* portName)` when discovered
DOT_PAD_CONNECT_BLE(const wchar_t* deviceName)	<ul style="list-style-type: none"> • Connects to BLE • Returns the DotDevice attempting to connect; returns null if connection attempt fails • Connection is complete only when the DOT_DATA_CODE_CONNECTED response arrives in the CALLBACK
DOT_PAD_CONNECT_SERIAL(const wchar_t* portName)	<ul style="list-style-type: none"> • USB connection • Returns the DotDevice attempting to connect; returns null if connection attempt fails • Connection is complete only when the DOT_DATA_CODE_CONNECTED response arrives in the CALLBACK
DOT_PAD_DISCONNECT(void* deviceHandle)	<ul style="list-style-type: none"> • Disconnect • If dotDevice == nullptr, disconnect all devices and initialize the internal list • If a specific dotDevice is specified, disconnect only that device and remove it from the list <p>Disconnect all: DOT_PAD_DISCONNECT(nullptr)</p> <p>Disconnect only a specific device: DOT_PAD_DISCONNECT(targetDevice)</p>
DOT_PAD_GET_CONNECTED_DEVICE_COUNT()	<ul style="list-style-type: none"> • Query the number of connected devices
DOT_PAD_GET_CONNECTED_DEVICE_HANDLE(int index, void** deviceHandle)	<ul style="list-style-type: none"> • Retrieve handle of connected device (index: starts at 0) • Returns via parameter deviceHandle, returns true if successful

4.2 Device Output

Function Name	Description
DOT_PAD_DISPLAY_FILE(displayFile, deviceHandle)	<ul style="list-style-type: none"> • Display graphic data from a file path

DOT_PAD_DISPLAY_DATA(data, len, deviceHandle)	• Display graphic data as a byte array
DOT_PAD_RESET_DISPLAY(deviceHandle)	• Reset graphic display
DOT_PAD_BRAILLE_DISPLAY(strInput, language, grade, englishGradeIfKorean, deviceHandle, callback)	• Display text converted to Braille. Receive conversion result via callback
DOT_PAD_BRAILLE_DISPLAY_DATA(brailleData, dataSize, deviceHandle)	• Directly displays braille data
DOT_PAD_BRAILLE_ASCII_DISPLAY(brailleASCII, deviceHandle)	• Display as Braille ASCII string
DOT_PAD_RESET_BRAILLE_DISPLAY(deviceHandle)	• Reset braille display

4.3 Settings

Function Name	Description
DOT_PAD_SET_LANGUAGE(language, grade)	<ul style="list-style-type: none"> Set braille language and grade 1 = ARABIC 2 = CHINESE_TRADITIONAL 3 = CHINESE_SIMPLIFIED 4 = DUTCH 5 = ENGLISH 6 = FRENCH 7 = GERMAN 8 = ITALIAN 9 = JAPANESE 10 = KOREAN 11 = RUSSIAN 12 = SPANISH 13 = VIETNAMESE 14 = BULGARIAN 15 = PORTUGUESE 16 = CZECH 17 = POLISH 18 = NORWEGIAN
DOT_PAD_SET_ENGLISH_GRADE_IF_KOREAN(grade)	<ul style="list-style-type: none"> English Braille Grade when using Korean

4.4 Information Lookup

Function Name	Description
DOT_PAD_GET_DEVICE_NAME(deviceHandle)	<ul style="list-style-type: none"> Query device name (result received via message callback)

DOT_PAD_GET_FW_VERSION(deviceHandle)	<ul style="list-style-type: none"> • Get Firmware Version (Result received via message callback)
DOT_PAD_GET_HW_VERSION(deviceHandle)	<ul style="list-style-type: none"> • Hardware Version
DOT_PAD_GET_DISPLAY_INFO(deviceHandle, &width, &height, &braille)	<ul style="list-style-type: none"> • Display size and braille availability

4.5 Listener

Function Name	Description
DOT_PAD_REGISTER_KEY_CALLBACK(cb)	<ul style="list-style-type: none"> • Key input callback. • (deviceHandle, DOT_KEY_CODE, message)
DOT_PAD_REGISTER_MESSAGE_CALLBACK(cb)	<ul style="list-style-type: none"> • Message callback • (deviceHandle, DOT_DATA_CODE, message)
DOT_PAD_REGISTER_DISPLAY_CALLBACK(cb)	<ul style="list-style-type: none"> • Display completion callback • (deviceHandle)