

# Protech API

## User Guide

**Version notice**

<b>Date</b>	<b>Version</b>	<b>Remark</b>
2023-04-06	A01-J510-000-01-230406	New Release version
2023-04-11	A01-J510-000-02-230411	Add PA-J500 support
2023-04-28	A01-J510-000-03-230428	Add PA-J520 support Add PA-J501 support Add PA-J511 support Add PA-J521 support
2023-05-03	A01-J500-000-03-230503	Change default to J500
2023-05-24	A01-D226-000-01-230524	Add BM-D226 (K600) support
2023-07-28	A01-N108-000-01-230728	Add SE-N108 IO board support for U236
2023-08-14	A01-N108-000-02-230814	Support both Internal / External DIO for N108

Note: API functions support list should be different according to the product.

# API Package Content

Operation System	Windows 10 64bits	
Directory	Contents / File Name	Description
.\	API User Guide A01-xxx-xxx-xx-xxxxxx.pdf	User Guide
.\Demo\	Demo applications base on sample code	Include necessary library and xml files, see <b>Function DLL</b> for detail.
.\Demo Project\	Sample code projects	See <b>Sample Program</b> for detail.
.\ProxAPI standard\	Integrated test application for all functions.	Single application for test all functions quickly.
Function DLL		
Function	File Name	Description
<i>Cash Drawer</i>	CashDrawer.dll	Library for Cash Drawer API.
<i>Device Power Control</i>	DevPowerControl.dll	Library for Device Power Control API.
<i>Digital IO</i>	Digital.dll	Library for Digital IO API
<i>Hardware Monitor</i>	Hardware Monitor.dll	Library for Hardware Monitor API.
<i>Watch Dog</i>	WatchDog.dll	Library for Watch Dog API.
<i>Windows Memory Access</i>	Winlo64.dll	Common library for Memory Access.
<i>Windows Memory Access</i>	Winlo64.sys	Common driver for Memory Access.
<i>Windows IO Port Access</i>	inpoutx64.dll	Common library for IO Port Access.
<i>XML configuration file Access</i>	multilangXML.dll	Common library for XML access.
Sample Program		
Directory	Contents / File Name	Description
<b>Demo Project\ CashDrawer</b>	Sample code for Cash Drawer	Visual Studio Project
<b>Demo Project\ DevicePowerControl</b>	Sample code for Device Power Control	Visual Studio Project
<b>Demo Project\ Digital</b>	Sample code for Digital IO	Visual Studio Project
<b>Demo Project\ HardwareMonitor</b>	Sample code for Hardware Monitor	Visual Studio Project
<b>Demo Project\ WatchDog</b>	Sample code for Watch Dog	Visual Studio Project

## Cash Drawer API usage

---

### **int Initial\_CashDrawer(void)**

Parameters:

Void

Return Value:

True (1) success, False (0) failed

### **bool CashDrawerOpen(short num\_drawer)**

Parameters:

Num\_drawer:

Index to the drawer .

0 – Open first Cash Drawer.

1 – Open second Cash Drawer.

Return Value:

True (1) success, False (0) failed

### **bool GetCashDrawerStatus(short num\_drawer)**

Parameters:

Num\_drawer:

Index to the drawer .

0 – Open first Cash Drawer.

1 – Open second Cash Drawer.

Return Value:

True (1) Drawer status is Open, (0) Drawer status is Close

## Device Power Control API usage

---

### **bool DevPowerControl\_API DevPowerControl\_Initialize()**

Parameters:

Void

Return Value:

True (1) success, False (0) failed

### **bool DevPowerControl\_API DevPowerControl\_Set(int Index, bool data)**

Parameters:

Index:

Index to the device and start from 0, use DevPowerControl\_GetNum() to know the max device number.

Logic\_status:

True – Enable device power.

False – Disable device power.

Return Value:

True (1) success, False (0) failed

### **bool DevPowerControl\_API DevPowerControl\_Get(int Index)**

Parameters:

Index:

Index to the device and start from 0, use DevPowerControl\_GetNum() to know the max device number.

Return Value:

True – Device power is ON.

False – Device power is OFF.

### **bool DevPowerControl\_API DevPowerControl\_Switch(int Index)**

Parameters:

Index:

Index to the device and start from 0, use DevPowerControl\_GetNum() to know the max device number.

Return Value:

True – New status of device power is ON.

False – New status of device power is OFF.

**bool DevPowerControl\_API DevPowerControl\_GetName(wchar\_t wzBuf[], int Index)**

Parameters:

wzBuf:

Pointer of uni-code string of the Indexed device name.

Index:

Index to the device and start from 0, use DevPowerControl\_GetNum() to know the max device number.

Return Value:

True (1) success, wzBuf will contain the device name.

False (0) failed

## Digital-IO API usage

---

### **bool Digital\_SetMode(bool mode)**

Parameters:

mode:

Boolean to represent external IO mode, must set it before Digital\_Initialize().

True – Set to external DIO mode.

False – Set to internal DIO mode.

Return Value:

True (1) success, (0) failed

### **bool Digital\_Initialize(void)**

Parameters:

Void

Return Value:

True (1) success, (0) failed

### **bool Digital\_Set(short logic\_status)**

Parameters:

Logic\_status:

Bitwise to represent a single port by bit address.

True – Set Digital I/O to logic “1” to relative bit.

False – Set Digital I/O to logic “0” to relative bit.

Return Value:

True (1) success, False (0) failed

### **short Digital\_Get(void)**

Parameters:

Void

Return Value:

Bitwise to represent a single port by bit address.

True – Digital I/O logic status at relative bit is High

False – Digital I/O logic status at relative bit is Low

## Hardware Monitor API usage

---

### **float HWMVoltage\_Get(short VoltType)**

Parameters:

**VoltType:**

- 1 – VCORE
- 2 – N/A
- 3 – VCC3V
- 4 – VSB3V
- 5 – VCC5V
- 6 – VSB5V
- 7 – VCC12V
- 8 – VBAT
- 9 – N/A

Return Value:

Voltage value in float.

### **float HWMFanSpeed\_Get(short fanType)**

Parameters:

**fanType:**

- 1 – CPU
- 2 – SYS
- 3 – N/A

Return Value:

Fan RPM value in float.

### **float HWMTemperature\_Get(short tempType)**

Parameters:

**tempType:**

- 1 – CPU
- 2 – N/A

Return Value:

DegreeC value in float.

## Watch Dog API usage

---

### **bool Watchdog\_Set(int value)**

Parameters:

**value:**

Set watchdog timer to the seconds/minutes number of value from 0 ~ 255.

Return Value:

True (1) success, False (0) failed

### **bool SetMinSec(int kind)**

Parameters:

**kind:**

1 – Second

others – Minute

Return Value:

True (1) success, False (0) failed

### **bool Watchdog\_Stop(void)**

Parameters:

Void

Return Value:

True (1) success, False (0) failed