

# Tablet API Specifications

---

***Revision: 3.5***

***June 5 2018***

## **Shortcuts:**

Algiz 10X V2 = A10XV2

Algiz 10X V3 = A10XV3

Algiz 8X = A8X

# Contents

---

<b>1. General Description .....</b>	<b>5</b>
<b>1.1 Introduction .....</b>	<b>5</b>
<b>2. Driver Installation .....</b>	<b>6</b>
<b>2.1 WMMIO Driver Installation .....</b>	<b>6</b>
<b>3. Programming Environment .....</b>	<b>7</b>
<b>3.1 Project Setting .....</b>	<b>7</b>
<b>3.2 WMIO2.h File Reference: .....</b>	<b>7</b>
<b>4. Function Key Definition .....</b>	<b>9</b>
<b>4.1 Consumer Mode Key Define .....</b>	<b>9</b>
<b>4.2 Function Procedure: .....</b>	<b>10</b>
4.2.1 Function Block.....	10
<b>4.3 Function Name: .....</b>	<b>10</b>
4.3.1 ModeOpen(UINT uiMode);.....	10
4.3.2 ModeClose();.....	11
4.3.3 SetDevice(BYTE uiValue); .....	11
4.3.4 GetDevice(PBYTE puiValue); .....	12
4.3.5 GetSMBIOSInfo(PCHAR pstName, PCHAR pstValue); .....	12
4.3.6 SetDevice2(BYTE uiValue); .....	13
4.3.7 GetDevice2(PBYTE puiValue); .....	15
4.3.8 GetDockingStatus(PBYTE puiValue); .....	15
4.3.9 GetBattery1SpecificInfo(UINT uiCommand, PUINT puiValue); .....	16
4.3.10 GetBattery2SpecificInfo(UINT uiCommand, PUINT puiValue); .....	18
4.3.11 GetCPUTemperature(PBYTE puiValue); .....	19
4.3.12 SetCPUPL1(BYTE uiValue); .....	20
4.3.13 GetCPUPL1(PBYTE uiValue); .....	21
<b>5. Sample Code .....</b>	<b>24</b>
<b>5.1 ECcomm 24</b>	
<b>5.2 BatteryCheckInfo.....</b>	<b>24</b>
<b>5.3 CameraFlashDLLTestAP .....</b>	<b>24</b>
<b>5.4 PackagePower .....</b>	<b>25</b>
<b>6. Appendix-A Device Table .....</b>	<b>26</b>
<b>6.1 A10XV2: 26</b>	

<b>6.2</b>	<b>A8X:</b>	<b>26</b>
<b>6.3</b>	<b>A10XV3:</b>	<b>26</b>
<b>7.</b>	<b>Appendix-B Keycode Table .....</b>	<b>1</b>
<b>7.1</b>	<b>Key Event and Platform Support Comparison Table .....</b>	<b>1</b>
7.1.1	Table 3 for A10XV2/A10XV3/A8X .....	1
<b>7.2</b>	<b>Combine Key and scanCode Comparison Table.....</b>	<b>2</b>
<b>8.</b>	<b>Appendix-C Function Support Table .....</b>	<b>3</b>
<b>9.</b>	<b>Appendix-E Brightness Control Method .....</b>	<b>1</b>
<b>9.1</b>	<b>WmiSetBrightness method of the WmiMonitorBrightnessMethods class .....</b>	<b>1</b>
<b>10.</b>	<b>Appendix-F Three Touch Modes Control .....</b>	<b>2</b>
<b>10.1</b>	<b>Introduction.....</b>	<b>2</b>
<b>10.2</b>	<b>Project Setting .....</b>	<b>2</b>
<b>10.3</b>	<b>WMTOUCHMODE.h File Reference:.....</b>	<b>2</b>
<b>10.4</b>	<b>Function Name: .....</b>	<b>3</b>
10.4.1	SetTouchMode(BYTE uiValue); .....	3
10.4.2	GetTouchMode(PBYTE puiTouch); .....	3
10.4.3	GetTocuhModelVersion(PCHAR pstModel,PCHAR pstVersion); .....	4
<b>10.5</b>	<b>Sample Code:.....</b>	<b>4</b>
10.5.1	TouchModeSetting .....	4

## Revision History

Revision	Date	Description
2.0	May 26, 2014	1. Initial draft.
2.1	Nov 28, 2014	1. Add Device Table 2. Add Keycode Table
2.2	May 29, 2015	1. Add Algiz Device Table 2. Add Algiz keycode Table
2.3	Oct 30, 2015	1. Update WMIO2.dll to v1.0.1.4 (1) Fix multi-thread call api conflict issue.
2.4	Nov 4, 2015	1. Add battery info function. (1) GetBattery1SpecificInfo()

2.5	Nov 23, 2015	<ol style="list-style-type: none"> <li>1. Update Appendix-D Camera Flash Control</li> <li>2. Update WMIO2.dll to v1.0.1.6 (1) Optimization all command read/write speed.</li> <li>3. Update sample code (1) CameraFlashDLLTestAP(5M) V1.0.0.7 (2) BatteryCheckInfo V1.1.0.3 (3) ECcommV1.3.0.0</li> <li>4. Add section 5.Sample Code description</li> </ol>
2.6	Mar 29, 2016	<ol style="list-style-type: none"> <li>1. Support x64 WMIO2 DLL</li> <li>2. Add WMTOUCHMODE DLL(x86/x64) for Three Touch Modes control.(refer Appendix-F)</li> </ol>
2.7	Jun 01, 2016	<ol style="list-style-type: none"> <li>1. Add Algiz 8X Device Table</li> </ol>
2.8	Sep 30, 2016	<ol style="list-style-type: none"> <li>1. Add A10XV3 Device Table</li> <li>2. Add A10XV3 Key Event Table</li> <li>3. Add A10XV3 Function Support Table</li> <li>4. Modify A8X Key Event Table</li> <li>5. Update WMTOUCHMODE DLL to v1.0.0.1 , support A10XV3/A8X</li> <li>6. Update WMMIO driver to v3.1.0.1</li> </ol>
2.9	May 11, 2017	<ol style="list-style-type: none"> <li>1. Update WMIO2.dll to v1.0.2.5 add package power command for A8X (1) Add command GetCPUtemperature() (2)Add command SetCPUPL1() (3)Add command GetCPUPL1()</li> </ol>
3.0	August 8, 2017	<ol style="list-style-type: none"> <li>1. Add Device Table</li> <li>2. Add keycode Table</li> </ol>
3.1	November 28, 2017	<ol style="list-style-type: none"> <li>1. Update Device Table</li> </ol>

3.2	December 4, 2017	<ol style="list-style-type: none"> <li>1. Update WMIO2.dll to v1.0.2.7 (1)Add command GetBackupBatteryStatus()</li> <li>2. Update sample code BatteryCheckInfo to v1.1.0.4</li> </ol>
3.3	December 25, 2017	<ol style="list-style-type: none"> <li>1. Add Device Table</li> <li>2. Add Keycode Table</li> <li>3. Add Function Support Table</li> </ol>
3.4	February 23, 2018	<ol style="list-style-type: none"> <li>1. Add Device Table</li> <li>2. Add Keycode Table</li> <li>3. Add Function Support Table</li> <li>4. Update command SetCPUPL1() and GetCPUPL1() support all platform. (1)Update sample code PackagePower to v1.0.0.1</li> <li>5. Remove command GetBackupBatteryStatus() , use GetBattery2SpecificInfo() instead. (1)Update sample code BatteryCheckInfo to v1.2.0.0</li> </ol>
3.5	June 5, 2018	<ol style="list-style-type: none"> <li>1. Add MIT-AA26DFBS Rear camera flash control. (1)Update sample code CameraFlashDLLTestAP to v1.0.0.9</li> <li>2. Fix smbios information some data can't display in A8X. (1)Update WMIO.dll to v1.0.3.0 (2)Update sample code ECcomm to v1.3.0.1</li> <li>3. Update sample code PackagePower to v1.0.0.2</li> </ol>

# 1. General Description

---

## 1.1 Introduction

Tablet SDK provide features to control the power state, camera flash, three touch modes and obtain SMBIOS information of the devices.

## 2. Driver Installation

---

### 2.1 WMMIO Driver Installation

The WMMIO driver provided is supported in Windows 7/8/8.1/10

Refer: Hottab\_WMMIO\_Driver Installation Guide v103.pdf

Notice: Only one **WMMIO** devices should be appear in Device Manager.

## 3. Programming Environment

---

### 3.1 Project Setting

For application developers, Dynamic-Link Library(.dll), Library(.lib) and Header(.h) files should be included while develop the application using for power state control, led flash control or obtaining SMBIOS information.

WMIO2.lib	Library
WMIO2.dll	Dynamic-Link Library
WMIO2.h	Header File

1. Include “WMIO2.h” in the project.
2. Add “WMIO2.lib” into project Link.
3. Put “WMIO2.dll” in the same path with application or into “windows directory.”
4. Check “WMIO2.dll” version is 1.0.3.0  
**※Support x86/x64 DLL.**

### 3.2 WMIO2.h File Reference:

#### Defines

```
#define WMIO2_API __declspec(dllexport)
```

#### Functions

```
extern "C" WMIO2_API bool ModeOpen (UINT uiMode);  
extern "C" WMIO2_API bool ModeClose();  
extern "C" WMIO2_API bool SetDevice(BYTE uiValue);  
extern "C" WMIO2_API bool GetDevice(PBYTE piValue);  
extern "C" WMIO2_API bool SetDevice2(BYTE uiValue);  
extern "C" WMIO2_API bool GetDevice2(PBYTE piValue);  
extern "C" WMIO2_API bool GetSMBIOSInfo(PCHAR pstName, PCHAR  
pstValue);  
extern "C" WMIO2_API bool GetDockingStatus(PBYTE puiValue);  
extern "C" WMIO2_API bool GetBattery1SpecificInfo(UINT uiCommand,  
PUINT puiValue);  
extern "C" WMIO2_API bool GetBattery2SpecificInfo(UINT uiCommand,
```



```
PUINT puiValue);  
extern "C" WMIO2_API bool GetCPUTemperature(PBYTE puiValue);  
extern "C" WMIO2_API bool SetCPUPL1(BYTE uiValue);  
extern "C" WMIO2_API bool GetCPUPL1(PBYTE uiValue);
```

## 4. Function Key Definition

---

### 4.1 Consumer Mode Key Define

Different key code would be send by the function keys depend on the consumer mode open.

Example:

Menu Key Press : Alt + Ctrl + Shift + 0

Volume Up(Up) Key Press : Alt + Ctrl + Shift + 1

Volume Down(Down) Key Press : Alt + Ctrl + Shift + 2

F1 Key Short Press : Alt + Ctrl + Shift + 3

F1 Key Long Press : Alt + Ctrl + Shift + 4

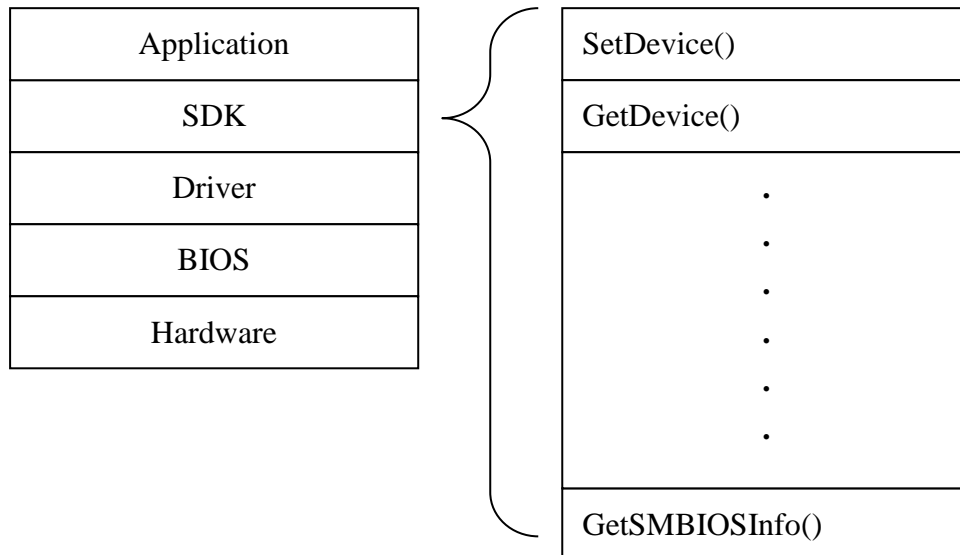
F2 Key Short Press : Alt + Ctrl + Shift + 5

F2 Key Long Press : Alt + Ctrl + Shift + 6

※Refer to 7.Appendix-B Keycode Table of different platforms

## 4.2 Function Procedure:

### 4.2.1 Function Block



## 4.3 Function Name:

### 4.3.1 ModeOpen(UINT uiMode);

The ModeOpen() function enable the key code event according to the setting mode.

```
WMIO2_API bool ModeOpen (UINT uiMode);
```

#### Parameters

2: Consumer mode

#### Return Value

The function returns TRUE if it successfully open the application control mode, and FALSE otherwise.

#### Requirements

Header: Declared in WMIO2.h

### 4.3.2 ModeClose();

The ModeClose() function disable the key code of function keys.

```
WMIO2_API bool ModeClose();
```

#### Parameters

None

#### Return Value

The function returns TRUE if it successfully close the application control mode, and FALSE otherwise.

#### Requirements

Header: Declared in WMIO2.h

### 4.3.3 SetDevice(BYTE uiValue);

The SetDevice() Function set the device power on/off status.

```
WMIO2_API bool SetDevice(BYTE uiValue);
```

#### Parameters

0: The device power off

1: The device power on

Example: 0xA1 (refer Appendix-A Device Table)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
AIILED	-	WebCam Rear	Bluetooth	-	GPS	Gobi3G	Wifi

Device On: AIILED, WebCamRear, Wifi

Device Off: Bluetooth, Gobi3G, GPS

#### Return Value

The function returns TRUE if it successfully set the device, and FALSE otherwise.

#### Requirements

Header: Declared in WMIO2.h

#### 4.3.4 GetDevice(PBYTE puiValue);

The GetDevice() Function get the device power on/off status.

```
WMIO2_API bool GetDevice(PBYTE puiValue);
```

##### Parameters

OUT BYTE \* puiValue:

##### **puiValue :**

The function returns the data of device power status.

0: The device power off

1: The device power on

Example: 0x21 (refer Appendix-A Device Table)

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
AIILED	-	WebCam Rear	Bluetooth	-	GPS	Gobi3G	Wifi

Device On: AIILED, WebCamRear, Wifi

Device Off: Bluetooth, Gobi3G, GPS

##### Return Value

The function returns TRUE if it successfully get the device status, and FALSE otherwise.

##### Requirements

Header: Declared in WMIO2.h

#### 4.3.5 GetSMBIOSInfo(PCHAR pstName, PCHAR pstValue);

The GetSMBIOSInfo() Function get the smbios information level.

```
WMIO2_API bool GetSMBIOSInfo(PCHAR pstName, PCHAR  
pstValue);
```

##### Parameters

**pstName:** Input string index to get smbios information from pstValue

Index:

"/IV" – BIOS version

"/ID" – BIOS release date

"/SM" – System manufacture  
"/SP" – System product  
"/SV" – System version  
"/SS" – System Serial number  
"/SU" – System UUID  
"/SK" – System SKU number  
"/SF" – System family

"/BM" – Baseboard manufacture  
"/BP" – Baseboard product  
"/BV" – Baseboard version  
"/BS" – Baseboard Serial number  
"/BT" – Baseboard Asset Tag

"/CM" – Chassis manufacture  
"/CT" – Chassis type  
"/CV" – Chassis version  
"/CS" – Chassis Serial number  
"/CA" – Chassis Tag number  
"/CO" – Chassis OEM-defined value

OUT CHAR \*pstValue:

**pstValue :**

The function returns the data of SMBIOS information string.

Example:

```
GetSMBIOSInformation("/IV", pstValue);  
pstValue = "V103"
```

**Return Value**

The function returns TRUE if it is successful get the smbios information string, and FALSE otherwise.

**Requirements**

Header: Declared in WMIO2.h

#### 4.3.6 SetDevice2(BYTE uiValue);

The SetDevice2() Function set the device power on/off status.

```
WMIO2_API bool SetDevice2(BYTE uiValue);
```

### Parameters

0: The device power off

1: The device power on

Example: 0x03 (refer Appendix-A Device Table)

Device2

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
-	-	Expand COM	Expand USB	GPS Antenna	RFID	WebCam Front	Barcode

Device On: Barcode, WebCamFront

Device Off: ExpandCOM, ExpandUSB, GPSAntenna, RFID

### Return Value

The function returns TRUE if it successfully set the device, and FALSE otherwise.

### Requirements

Header: Declared in WMIO2.h

#### 4.3.7 GetDevice2(PBYTE puiValue);

The GetDevice2() Function get the device power on/off status.

```
WMIO2_API bool GetDevice2(PBYTE puiValue);
```

##### Parameters

OUT BYTE \* puiValue:

##### **puiValue :**

The function returns the data of device power status.

0: The device power off

1: The device power on

Example: 0x03 (refer Appendix-A Device Table)

Device2

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
-	-	Expand COM	Expand USB	GPS Antenna	RFID	WebCam Front	Barcode

Device On: Barcode, WebCamFront

Device Off: ExpandCOM, ExpandUSB, GPSAntenna, RFID

##### Return Value

The function returns TRUE if it successfully get the device status, and FALSE otherwise.

##### Requirements

Header: Declared in WMIO2.h

#### 4.3.8 GetDockingStatus(PBYTE puiValue);

The GetDockingStatus() Function get the docking status.

```
WMIO2_API bool GetDockingStatus(PBYTE puiValue);
```

##### Parameters

OUT BYTE \* puiValue:

##### **puiValue :**

The function returns the data of docking status.



- 0: Docking Out
- 1: Ground Docking In
- 2: Air Docking In

### Return Value

The function returns TRUE if it successfully get the docking status, and FALSE otherwise.

### Requirements

Header: Declared in WMIO2.h

## 4.3.9 GetBattery1SpecificInfo(UINT uiCommand, PUINT puiValue);

The GetBattery1SpecificInfo() Function get the battery specific information.

```
WMIO2_API bool GetBattery1SpecificInfo(UINT uiCommand, PUINT puiValue);
```

### Parameters

**uiCommand:** Input command index to get battery specific information from puiValue

Index:

Name	Index	Data
reserved	0x00	
reserved	0x01	
reserved	0x02	
BatteryMode	0x03	bit flags
reserved	0x04	
reserved	0x05	
reserved	0x06	
reserved	0x07	
Temperature	0x08	0.1°K
Voltage	0x09	mV
Current	0x0A	mA
reserved	0x0B	

reserved	0x0C	
RelativeStateOfCharge	0x0D	percent
AbsoluteStateOfCharge	0x0E	percent
RemainingCapacity	0x0F	mAh
FullChargeCapacity	0x10	mAh
RunTimeToEmpty	0x11	minutes
AverageTimeToEmpty	0x12	minutes
AverageTimeToFull	0x13	minutes
ChargingCurrent	0x14	mA
ChargingVoltage	0x15	mV
BatteryStatus	0x16	bit flags
CycleCount	0x17	count
DesignCapacity	0x18	mAh
DesignVoltage	0x19	mV
reserved	0x1A	
ManufactureDate	0x1B	unsigned int
reserved	0x1C	

OUT UINT \* puiValue:

**puiValue :**

The function returns the data of battery specific information.

ManufactureDate:

Field	Bits Used	Format	Allowable Values
Day	0...4	5 bit binary value	1 - 31 (corresponds to date)
Month	5...8	4 bit binary value	1 - 12 (corresponds to month number)
Year	9...15	7 bit binary value	0 - 127 (corresponds to year biased by 1980)

### Return Value

The function returns TRUE if it successfully get battery specific information, and FALSE otherwise.

### Requirements

Header: Declared in WMIO2.h

#### 4.3.10 GetBattery2SpecificInfo(UINT uiCommand, PUINT puiValue);

The GetBattery2SpecificInfo() Function get the battery specific information.

```
WMIO2_API bool GetBattery2SpecificInfo(UINT uiCommand,  
PUINT puiValue);
```

##### Parameters

**uiCommand:** Input command index to get battery specific information from puiValue

Index:

Name	Index	Data
reserved	0x00	
reserved	0x01	
reserved	0x02	
BatteryMode	0x03	bit flags
reserved	0x04	
reserved	0x05	
reserved	0x06	
reserved	0x07	
Temperature	0x08	0.1°K
Voltage	0x09	mV
Current	0x0A	mA
reserved	0x0B	
reserved	0x0C	
RelativeStateOfCharge	0x0D	percent
AbsoluteStateOfCharge	0x0E	percent
RemainingCapacity	0x0F	mAh
FullChargeCapacity	0x10	mAh
RunTimeToEmpty	0x11	minutes
AverageTimeToEmpty	0x12	minutes
AverageTimeToFull	0x13	minutes
ChargingCurrent	0x14	mA
ChargingVoltage	0x15	mV

BatteryStatus	0x16	bit flags
CycleCount	0x17	count
DesignCapacity	0x18	mAh
DesignVoltage	0x19	mV
reserved	0x1A	
ManufactureDate	0x1B	unsigned int
reserved	0x1C	

OUT UINT \* puiValue:

**puiValue :**

The function returns the data of battery specific information.

ManufactureDate:

Field	Bits Used	Format	Allowable Values
Day	0...4	5 bit binary value	1 - 31 (corresponds to date)
Month	5...8	4 bit binary value	1 - 12 (corresponds to month number)
Year	9...15	7 bit binary value	0 - 127 (corresponds to year biased by 1980)

### Return Value

The function returns TRUE if it successfully get battery specific information, and FALSE otherwise.

### Requirements

Header: Declared in WMIO2.h

#### 4.3.11 GetCPUTemperature(PBYTE puiValue);

The GetCPUTemperature() Function get the CPU temperature.

```
WMIO2_API bool GetCPUTemperature(PBYTE puiValue);
```

### Parameters

OUT BYTE \* puiValue:

**puiValue :**

The function returns the CPU temperature(unit degree C).

### Return Value

The function returns TRUE if it successfully get the CPU temperature, and FALSE otherwise.

## Requirements

Header: Declared in WMIO2.h

### 4.3.12 SetCPUPL1(BYTE uiValue);

The SetCPUPL1() Function set the CPU Package Power Limit-1.

```
WMIO2_API bool SetCPUPL1(BYTE uiValue);
```

#### Parameters

**uiValue:** range 0-9

(for A8X)

0: level0 , 1.024watts

1: level0 , 1.536watts

2: level0 , 2.048watts

3: level0 , 2.560watts

4: level0 , 3.072watts

5: level0 , 3.584watts

6: level0 , 4.096watts

7: level0 , 4.608watts

8: level0 , 5.120watts

9: level0 , 5.632watts (A8X)

**uiValue:** range 0-28

(for A10XV3)

0: level0 , 1.000watts

1: level0 , 1.500watts

2: level0 , 2.000watts

3: level0 , 2.500watts

4: level0 , 3.000watts

5: level0 , 3.500watts

6: level0 , 4.000watts

7: level0 , 4.500watts

8: level0 , 5.000watts

9: level0 , 5.500watts

10: level0 , 6.000watts

11: level0 , 6.500watts

12: level0 , 7.000watts

13: level0 , 7.500watts (A10XV3 maximum)

14: level0 , 8.000watts  
 15: level0 , 8.500watts  
 16: level0 , 9.000watts  
 17: level0 , 9.500watts  
 18: level0 , 10.000watts  
 19: level0 , 10.500watts  
 20: level0 , 11.000watts  
 21: level0 , 11.500watts  
 22: level0 , 12.000watts  
 23: level0 , 12.500watts  
 24: level0 , 13.000watts  
 25: level0 , 13.500watts  
 26: level0 , 14.000watts  
 27: level0 , 14.500watts  
 28: level0 , 15.000watts

### Return Value

The function returns TRUE if it successfully set the CPU PL1 value, and FALSE otherwise.

### Requirements

Header: Declared in WMIO2.h

### 4.3.13 GetCPUPL1(PBYTE uiValue);

The GetCPUPL1() Function get the CPU Package Power Limit-1 value.

```
WMIO2_API bool GetCPUPL1(PBYTE uiValue);
```

### Parameters

OUT BYTE \* uiValue:

The function returns the CPU PL1 value.

**uiValue:** range 0-9

(for A8X)

0: level0 , 1.024watts  
 1: level0 , 1.536watts  
 2: level0 , 2.048watts  
 3: level0 , 2.560watts  
 4: level0 , 3.072watts  
 5: level0 , 3.584watts

6: level0 , 4.096watts  
7: level0 , 4.608watts  
8: level0 , 5.120watts  
9: level0 , 5.632watts (A8X maximum)

**uiValue:** range 0-28

(for A10XV3)

0: level0 , 1.000watts  
1: level0 , 1.500watts  
2: level0 , 2.000watts  
3: level0 , 2.500watts  
4: level0 , 3.000watts  
5: level0 , 3.500watts  
6: level0 , 4.000watts  
7: level0 , 4.500watts  
8: level0 , 5.000watts  
9: level0 , 5.500watts  
10: level0 , 6.000watts  
11: level0 , 6.500watts  
12: level0 , 7.000watts  
13: level0 , 7.500watts (A10XV3 maximum)  
14: level0 , 8.000watts  
15: level0 , 8.500watts  
16: level0 , 9.000watts  
17: level0 , 9.500watts  
18: level0 , 10.000watts  
19: level0 , 10.500watts  
20: level0 , 11.000watts  
21: level0 , 11.500watts  
22: level0 , 12.000watts  
23: level0 , 12.500watts  
24: level0 , 13.000watts  
25: level0 , 13.500watts  
26: level0 , 14.000watts  
27: level0 , 14.500watts  
28: level0 , 15.000watts

## **Return Value**

The function returns TRUE if it successfully get the device status, and FALSE otherwise.

**Requirements**

Header: Declared in WMIO2.h



## 5. Sample Code

---

### 5.1 ECcomm

ECcomm sample code , demonstrate how to control the power switch, smbios information obtained, function key hook demo.

**Development environment:**

Microsoft Visual Studio 2010 (or later) must be installed as well as .NET Framework 4.0

**Latest version is 1.3.0.1**

### 5.2 BatteryCheckInfo

BatteryCheckInfo sample code , demonstrate how to obtain specific battery information. Support **A8X**.

**Development environment:**

Microsoft Visual Studio 2010 (or later) must be installed as well as .NET Framework 4.0

**Latest version is 1.2.0.0**

### 5.3 CameraFlashDLLTestAP

CameraFlashDLLTestAP sample code , demonstrate how to control the camera flash method. Support **A10XV3 / A8X**

**Development environment:**

Microsoft Visual Studio 2010 (or later) must be installed as well as .NET Framework 4.0

**Latest version is 1.0.0.9**

## 5.4 PackagePower

PackagePower sample code , demonstrate how to obtain CPU temperature and CPU PL1 value. Support **A10XV3 / A8X**.

### **Development environment:**

Microsoft Visual Studio 2010 (or later) must be installed as well as .NET Framework 4.0

**Latest version is 1.0.0.2**

## 6. Appendix-A Device Table

---

### 6.1 A10XV2:

#### Device1

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
-	-	WebCam Rear	Bluetooth	-	GPS	Gobi3G	Wifi

#### Device2

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
-	-	-	-	GPS Antenna	-3G Antenna		

### 6.2 A8X:

#### Device1

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
-	Config LED1	WebCam Rear	Bluetooth	-	GPS	Gobi3G	Wifi

#### Device2

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
-	Config LED2	-	-	GPS Antenna	-3G Antenna	WebC am Front	BarCod e

### 6.3 A10XV3:

#### Device1

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
-	-	WebCam Rear	Bluetooth	-	GPS	Gobi3G	Wifi

**Device2**

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
-	-	-	-	GPS Antenna	-3G Antenna		

## 7. Appendix-B Keycode Table

### 7.1 Key Event and Platform Support Comparison Table

7.1.1 Table 3 for A10XV2/A10XV3/A8X

Event	Combine Key	Platform			
		A10XV2	A10XV3	A8X	
Menu Key Press	Alt+Ctrl+Shift+0	<b>V</b>	<b>V</b>	<b>V</b>	
Volume UP Key Press	Alt+Ctrl+Shift+1	<b>UP</b>			
Volume Down Key Press	Alt+Ctrl+Shift+2	<b>DOWN</b>			
F1 Key Short Press	Alt+Ctrl+Shift+3	<b>V</b>	<b>V</b>	<b>V</b>	
F1 Key Long Press	Alt+Ctrl+Shift+4	<b>V</b>	<b>V</b>	<b>V</b>	
F2 Key Short Press	Alt+Ctrl+Shift+5	<b>V</b>	<b>V</b>	<b>V</b>	
F2 Key Long Press	Alt+Ctrl+Shift+6	<b>V</b>	<b>V</b>	<b>V</b>	
F3 Key Short Press	Alt+Ctrl+Shift+7	<b>V</b>	<b>Lock</b>	<b>Lock</b>	
F3 Key Long Press	Alt+Ctrl+Shift+8	<b>V</b>	<b>Lock</b>	<b>Lock</b>	
F4 Key Short Press	Alt+Ctrl+Shift+N				
F4 Key Long Press	Alt+Ctrl+Shift+F				
F5 Key Short Press	Alt+Ctrl+Shift+G				
F5 Key Long Press	Alt+Ctrl+Shift+H				
F6 Key Short Press	Alt+Ctrl+Shift+I				
F6 Key Long Press	Alt+Ctrl+Shift+J				
F7 Key Short Press	Alt+Ctrl+Shift+K				
F7 Key Long Press	Alt+Ctrl+Shift+L				
Home Key Short Press	Alt+Ctrl+Shift+D		<b>V</b>	<b>V</b>	
Home Key Long Press	Alt+Ctrl+Shift+M		<b>V</b>	<b>V</b>	
Left Key Press	Alt+Ctrl+Shift+9	<b>V</b>			
Right Key Press	Alt+Ctrl+Shift+A	<b>V</b>			
OK Key Press	Alt+Ctrl+Shift+E	<b>V</b>			
Docking AC Out(AIR / GROUND)	Alt+Ctrl+Shift+Q				
AIR DOCKING IN	Alt+Ctrl+Shift+N				
GROUND DOCKING IN	Alt+Ctrl+Shift+P				

## 7.2 Combine Key and scanCode Comparison Table

Event	combine key				scanCode(Hex)				
Menu Key Press	Alt	Ctrl	Shift	0	38	1D	36	0B	
Volume Up(Up) Key Press	Alt	Ctrl	Shift	1	38	1D	36	02	
Volume Down(Down) Key Press	Alt	Ctrl	Shift	2	38	1D	36	03	
F1 Key Short Press	Alt	Ctrl	Shift	3	38	1D	36	04	
F1 Key Long Press	Alt	Ctrl	Shift	4	38	1D	36	05	
F2 Key Short Press	Alt	Ctrl	Shift	5	38	1D	36	06	
F2 Key Long Press	Alt	Ctrl	Shift	6	38	1D	36	07	
F3 Key Short Press	Alt	Ctrl	Shift	7	38	1D	36	08	
F3 Key Long Press	Alt	Ctrl	Shift	8	38	1D	36	09	
F4 Key Short Press	Alt	Ctrl	Shift	N	38	1D	36	31	
F4 Key Long Press	Alt	Ctrl	Shift	F	38	1D	36	21	
F5 Key Short Press	Alt	Ctrl	Shift	G	38	1D	36	22	
F5 Key Long Press	Alt	Ctrl	Shift	H	38	1D	36	23	
F6 Key Short Press	Alt	Ctrl	Shift	I	38	1D	36	17	
F6 Key Long Press	Alt	Ctrl	Shift	J	38	1D	36	24	
F7 Key Short Press	Alt	Ctrl	Shift	K	38	1D	36	25	
F7 Key Long Press	Alt	Ctrl	Shift	L	38	1D	36	26	
Home Key Short Press	Alt	Ctrl	Shift	D	38	1D	36	20	
Home Key Long Press	Alt	Ctrl	Shift	M	38	1D	36	32	
Left Key Press	Alt	Ctrl	Shift	9	38	1D	36	0A	
Right Key Press	Alt	Ctrl	Shift	A	38	1D	36	1E	
OK Key Press	Alt	Ctrl	Shift	E	38	1D	36	12	
Docking AC Out(AIR / GROUND)	Alt	Ctrl	Shift	Q	38	1D	36	10	
AIR DOCKING IN	Alt	Ctrl	Shift	N	38	1D	36	31	
GROUND DOCKING IN	Alt	Ctrl	Shift	P	38	1D	36	19	

## 8. Appendix-C Function Support Table

---

Command	A8X	A10XV2	A10XV3
ModeOpen	V	V	V
ModeClose	V	V	V
SetDevice	V	V	V
GetDevice	V	V	V
SetDevice2	V	V	V
GetDevice2	V	V	V
GetSMBIOSInfo	V	V	V
GetDockingStatus			
GetBattery1SpecificInfo	V		
GetBattery2SpecificInfo	V		
GetCPUTemperature	V	V	V
SetCPUPL1	V	V	V
GetCPUPL1	V	V	V

## 9. Appendix-E Brightness Control Method

---

### 9.1 WmiSetBrightness method of the WmiMonitorBrightnessMethods class

Refer follow Microsoft MSDN Link

<https://msdn.microsoft.com/en-us/library/aa394549.aspx>



## 10. Appendix-F Three Touch Modes Control

---

### 10.1 Introduction

The dll provide features to control the three touch modes.

### 10.2 Project Setting

For application developers, Dynamic-Link Library(.dll), Library(.lib) and Header(.h) files should be included while develop the application using for three touch modes setting.

WMTOUCHMODE.lib	Library
WMTOUCHMODE.dll	Dynamic-Link Library
WMTOUCHMODE.h	Header File

5. Include “WMTOUCHMODE.h” in the project.
6. Add “WMTOUCHMODE.lib” into project Link.
7. Put “WMTOUCHMODE.dll” and " HIDdAPI.dll" in the same path with application or into “windows directory.”
8. Check “WMTOUCHMODE.dll” version is 1.0.0.4

※Support x86/x64 DLL.

◆ Support **A10XV3/A8X**.

### 10.3 WMTOUCHMODE.h File Reference:

Defines

```
#define WMTOUCHMODE_API __declspec(dllexport)
```

Functions

```
extern "C" WMTOUCHMODE_API bool GetTouchMode(PBYTE puiTouch);  
extern "C" WMTOUCHMODE_API bool SetTouchMode(BYTE uiValue);  
extern "C" WMTOUCHMODE_API bool GetTocuhModelVersion(PCHAR  
pstModel,PCHAR pstVersion);
```

◆ Support **A10XV3/A8X**

## 10.4 Function Name:

### 10.4.1 SetTouchMode(BYTE uiValue);

The SetTouchMode() function set three touch mode setting.

```
WMTOUCHMODE_API bool SetTouchMode (BYTE uiValue);
```

#### Parameters

- 0: Hand/Rain Mode
- 1: Stylus Mode
- 2: Glove Mode

#### Return Value

The function returns TRUE if it successfully set touch mode, and FALSE otherwise.

#### Requirements

Header: Declared in WMTOUCHMODE.h

### 10.4.2 GetTouchMode(PBYTE puiTouch);

The GetDevice2() Function get the device power on/off status.

```
WMTOUCHMODE_API bool GetTouchMode(PBYTE puiTouch);
```

#### Parameters

OUT BYTE \* puiTouch:

**puiTouch :**

The function returns the touch mode setting.

- 0: Hand/Rain Mode
- 1: Stylus Mode
- 2: Glove Mode

#### Return Value

The function returns TRUE if it successfully get touch mode setting, and FALSE otherwise.

## Requirements

Header: Declared in WMTOUCHMODE.h

### 10.4.3 GetTocuhModelVersion(PCHAR pstModel,PCHAR pstVersion);

The GetTocuhModelVersion() Function get the touch mode model and version string.

```
WMTOUCHMODE_API bool GetTocuhModelVersion(PCHAR  
pstModel,PCHAR pstVersion);
```

#### Parameters

OUT CHAR \*pstModel:

**pstModel :**

The function returns the touch mode model string.

OUT CHAR \*pstVersion:

**pstVersion :**

The function returns the touch mode version string.

Example:

```
GetTocuhModelVersion(pstValue1, pstValue2);
```

```
pstValue1 = "Sirius_1424"
```

```
pstValue1 = "00_T4"
```

#### Return Value

The function returns TRUE if it is successful support three touch mode get the touch mode model and version string, and FALSE otherwise.

## Requirements

Header: Declared in WMTOUCHMODE.h

## 10.5 Sample Code:

### 10.5.1 TouchModeSetting

TouchModeSetting sample code , demonstrate how to set three touch mode.

Support **A10XV3/A8X**

**Development environment:**

Microsoft Visual Studio 2010 (or later) must be installed as well as .NET Framework 4.0

**Latest version is 1.0.0.3**