



®

AXIOMTEK

Programming Guide of EApi of AXView 2.0 1.0.0.1

Published by Software R&D Center

Release Date: 2106/09/13





Programming Guide of EApi of AXView2.0



Software R&D Center

Programming Guide of EApi of AXView2.0

Release Note

Version	Revised Date	Author	Description
1.0.0.1	2106/09/13	Mark	- 1st release

Index

Release Note	2
Index.....	2
Type Definitions	3
Constants.....	4
Functions	7

Programming Guide of EApi of AXView2.0

Type Definitions

● Typedef

- **__IN**
 - ◆ Immediate value: Input value that must be specified and is essential to function operation.
 - ◆ Pointer: Valid pointer to initialized buffer/variable.

- **__OUT**
 - ◆ Pointer: Valid pointer to initialized buffer/variable.

- **__INOUT**
 - ◆ Pointer: Valid pointer to initialized buffer/variable. Contents of buffer/variable updated before return.

- **int8_t** definition type is int8

- **uint8_t** definition type is unsigned int8

- **int32_t** definition type is int32

- **uint32_t** definition type is unsigned int32

- **EApiStatus_t** definition type is uint32_t

- **EApild_t** definition type is uint32_t

Programming Guide of EApi of AXView2.0

Constants

● Constant

■ Status Codes – (EApiStatus_t)

◆ EAPI_STATUS_NOT_INITIALIZED

- Value: 0xFFFFFFFF
- Description:
The EAPI library is not yet or unsuccessfully initialized.
EApiLibInitialize needs to be called prior to the first access of any other EAPI function.
- Function Actions:
Call EApiLibInitialize.

◆ EAPI_STATUS_INITIALIZED

- Value: 0xFFFFFFFFE
- Description:
Library is initialized.
- Function Actions:
None.

◆ EAPI_STATUS_ALLOC_ERROR

- Value: 0xFFFFFFFFD
- Description:
Memory Allocation Error.
- Function Actions:
Free memory and try again.

Programming Guide of EApi of AXView2.0

- ◆ EAPI_STATUS_DRIVER_TIMEOUT
 - Value: 0xFFFFFFFFC
 - Description:
Time out in driver. This is Normally caused by hardware/software semaphore timeout.
 - Function Actions:
Retry.

- ◆ EAPI_AX_STATUS_NOT_RELEASED
 - Value: 0xFFFFFFFFB
 - Description:
Library of AXView is loaded failed.
 - Function Actions:
None.

- ◆ EAPI_AX_STATUS_DLLA_LOAD_FAILED
 - Value: 0xFFFFFFFFA
 - Description:
Library of AXView is loaded failed.
 - Function Actions:
None.

- ◆ EAPI_AX_STATUS_DLLB_LOAD_FAILED
 - Value: 0xFFFFFFFF9
 - Description:
Library of AXWIO is loaded failed.
 - Function Actions:
None.

Programming Guide of EApi of AXView2.0

- ◆ EAPI_STATUS_UNSUPPORTED
 - Value: 0xFFFFCFF
 - Description:
Function dose not support.
 - Function Actions:
None.

- ◆ EAPI_STATUS_SUCCESS
 - Value: 0x0
 - Description:
The operation was successful.
 - Function Actions:
None.

- ID Codes – (EApild_t)
 - ◆ String Information
 - EAPI_ID_BOARD_MANUFACTURER_STR 0
 - EAPI_ID_BOARD_NAME_STR 1
 - EAPI_ID_BOARD_BIOS_REVISION_STR 4

 - ◆ Value Information
 - EAPI_ID_BOARD_SENSOR_COUNTER_TEMP 12
 - EAPI_ID_BOARD_SENSOR_TEMP 13
 - EAPI_ID_BOARD_SENSOR_COUNTER_VOLTAGE 14
 - EAPI_ID_BOARD_SENSOR_VOLTAGE 15
 - EAPI_ID_BOARD_SENSOR_COUNTER_FAN 16

Programming Guide of EApi of AXView2.0

● EAPI_ID_BOARD_SENSOR_FAN	17
● EAPI_ID_BOARD_SENSOR_COUNTER_DIO_INTERNAL	31
● EAPI_ID_BOARD_SENSOR_GET_DIO_INTERNAL	32
● EAPI_ID_BOARD_SENSOR_SET_DIO_INTERNAL	32
● EAPI_ID_BOARD_SENSOR_COUNTER_WDT_INTERNAL	39
● EAPI_ID_BOARD_SENSOR_GET_WDT_INTERNAL	40
● EAPI_ID_BOARD_SENSOR_SET_WDT_INTERNAL	40
● EAPI_ID_BOARD_SENSOR_RELOAD_WDT_INTERNAL	41

Functions

● Functions

■ AXLibInitialize

◆ Should be called before calling any other API function is called.

◆ Syntax

EApi EApiStatus_t EAPI_CALLTYPE AXLibInitialize (void);

● Parameters

None

● Return Value

Status Codes (EApiStatus_t)

● Example (C#)

```
/* Initial Library */
if (AXLibInitialize() != EAPI_STATUS_SUCCESS)
{
    DialogResult DR = new System.Windows.Forms.DialogResult();
    DR = MessageBox.Show("Initial library is failed. (EApi)", "AXView - Agent", MessageBoxButtons.OK);
    if (DR == System.Windows.Forms.DialogResult.OK)
        this.Close();
}
```




Programming Guide of EApi of AXView2.0

■ AXBoardGetStringA

- ◆ To read board information strings
- ◆ Syntax

```
EApi EApiStatus_t EAPI_CALLTYPE EApiBoardGetStringA (  
    __IN      EApiId_t    Id,  
    __OUT     char        *pBuffer,  
    __INOUT   uint32_t    *pBufLen  
);
```

- Parameters

Id: ID of String Information

*pBuffer: Destination Buffer of String

*pBufLen: Length of Buffer

- Return Value

Status Codes (EApiStatus_t)

- Example (C#)

```
//Get Board String  
private static UInt32 ProductResult;  
  
private void GetProductNo()  
{  
    //M/B Model  
    byte[] BOARD_MODEL = new byte[64];  
    UInt32 pBufLen = (UInt32)BOARD_MODEL.Length;  
    ProductResult = AXBoardGetStringA(EAPI_ID_BOARD_NAME_STR, ref BOARD_MODEL[0], ref pBufLen);  
    if (ProductResult == EAPI_STATUS_SUCCESS)  
    {  
        InfoMB.Text = System.Text.Encoding.Default.GetString(BOARD_MODEL).Replace("\0", ' ').Trim();  
    }  
    else  
    {  
        InfoMB.Text = "ERC: 0x" + ProductResult.ToString("X");  
    }  
}
```

Programming Guide of EApi of AXView2.0

■ AXBoardGetValue

- ◆ To read board information values
- ◆ Syntax

```
EApi EApiStatus_t EAPI_CALLTYPE AXBoardGetValue (  
    __IN      EApiId_t    Id,  
    __OUT     char        *pBuffer,  
    __INOUT   uint32_t    *pBufLen,  
    __INOUT   uint32_t    *pIndex,  
    __INOUT   uint32_t    *pValue  
);
```

- Parameters

Id: ID of Value Information

*pBuffer: Destination Buffer of String (Name of each items)

*pBufLen: Length of Buffer

*pIndex: Index of Information

*pValue: Value of Informatin

(Conversions: Temperature/10, Voltage/100)

- Return Value
Status Codes (EApiStatus_t)
- Example (C#)

Programming Guide of EApi of AXView2.0

```
// Monitoring
// Get Numbers of Temperature
uint TempResult;
byte[] BOARD_TEMP = new byte[1];
UInt32 pBufLen = (UInt32)BOARD_TEMP.Length;
UInt32 pIndex = 0;
TempResult = AXBoardGetValue(EAPI_ID_BOARD_SENSOR_COUNTER_TEMP, ref BOARD_TEMP[0], ref pBufLen, ref pIndex, ref TempCounter);
if (TempResult != EAPI_STATUS_SUCCESS)
{
    TempCounter = 0;
    AXDataSet.Tables[0].Rows.Add("ErrorCode: 0x" + TempResult.ToString("X"));
}

// Get Value of each Temperature
for (int i = 0; i < TempCounter; i++)
{
    byte[] BOARD_TEMP_NAME = new byte[64];
    pBufLen = (UInt32)BOARD_TEMP_NAME.Length;
    UInt32 index = (UInt32)i;
    UInt32 pValue = 0;
    TempResult = AXBoardGetValue(EAPI_ID_BOARD_SENSOR_TEMP, ref BOARD_TEMP_NAME[0], ref pBufLen, ref index, ref pValue);
    if (TempResult == EAPI_STATUS_SUCCESS)
    {
        if (System.Text.Encoding.Default.GetString(BOARD_TEMP_NAME).Replace("\0", ' ').Trim() != "N/A")
        {
            AXDataSet.Tables[0].Rows.Add(System.Text.Encoding.Default.GetString(BOARD_TEMP_NAME).Replace("\0", ' ').Trim()); // Name of Item
            AXDataSet.Tables[0].Rows[(AXDataSet.Tables[0].Rows.Count - 1)][1] = ((float)pValue / 10).ToString("f1"); // Value of Item
        }
    }
}
}
```

■ AXBoardSetValue

- ◆ To write board information values
- ◆ Syntax

```
EApi EApiStatus_t EAPI_CALLTYPE AXBoardSetValue (
    __IN    EApild_t    Id,
    __IN    char        *pBuffer,
    __IN    uint32_t    *pBufLen,
    __IN    uint32_t    *pIndex,
    __IN    uint32_t    *pValue
);
```

Programming Guide of EApi of AXView2.0

- Parameters
 - Id: ID of Value Information
 - *pBuffer: Destination Buffer
 - *pBufLen: Length of Buffer
 - *pIndex: Index of Information
 - *pValue: Value of Information
- Return Value
 - Status Codes (EApiStatus_t)
- Example (C#)

```
// Set Some DO Pin High/Low
private bool SetDO(UInt32 position, UInt32 HL) // position: 0 ~ 143, HL: 0 means LOW, 1 means HIGH
{
    UInt32[] BOARD_DIO = new UInt32[64];
    UInt32 pBufLen = (UInt32)BOARD_DIO.Length;
    UInt32 index = UInt32.Parse(AXDataSet.Tables[12].Rows[(int)position][2].ToString());
    UInt32 SetHigh = HL;

    if (AXBoardSetValue(EAPI_ID_BOARD_SENSOR_SET_DIO_INTERNAL, ref BOARD_DIO[0], ref pBufLen, ref index, ref SetHigh) != EAPI_STATUS_SUCCESS)
    {
        AXDataSet.Tables[12].Rows[(int)position][0] = "ErrorCode: 0x" + DIOResult.ToString("X");
        return false;
    }
    else
    {
        return true;
    }
}
```

Programming Guide of EApi of AXView2.0

```
// Set On Board WDT
private bool SetWDT(UInt32 id) // id of On Board WDT is 0
{
    UInt32[] BOARD_WDT = new UInt32[64];
    UInt32 pBufLen = (UInt32)BOARD_WDT.Length;
    UInt32 index = UInt32.Parse(AXDataSet.Tables[6].Rows[(int)id][2].ToString());
    UInt32 value = 0; // 0 means STOP WDT, 5 ~ 15300 means START WDT

    WDTResult = AXBoardSetValue(EAPI_ID_BOARD_SENSOR_SET_WDT_INTERNAL, ref BOARD_WDT[0], ref pBufLen, ref index, ref value);

    if (WDTResult == EAPI_STATUS_SUCCESS && value != 0)
    {
        // After START WDT, remember to create a timer that calls a procedure every (value -2) seconds to RELOAD WDT.
        WDTResult = AXBoardSetValue(EAPI_ID_BOARD_SENSOR_RELOAD_WDT_INTERNAL, ref BOARD_WDT[0], ref pBufLen, ref index, ref value);
    }
    else if (WDTResult != EAPI_STATUS_SUCCESS)
    {
        MessageBox.Show("Watchdog can't be started. ErrorCode: 0x" + WDTResult.ToString("X"), "AXView - Agent");
        return false;
    }
    return true;
}
```