

UIROBOT

Programmer's Manual

uirSDK3.0
Control System Development Kit

uirSDK3.0 Control System Development Kit

[Intellectual Property Protection Statement]

- UIROBOT products meet the specification contained in their particular UIROBOT user manual;
- UIROBOT is willing to work with the customer who is concerned about intellectual property protection;
- Any attempt to destroy the code protection function of UIROBOT products may be regarded as a violation of intellectual property protection laws and regulations. If this behavior leads to the acquisition of software or other results protected by intellectual property rights without UIROBOT's authorization, UIROBOT has the right to file a lawsuit to stop this behavior in accordance with the Act.

[Disclaimer]

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. UIROBOT MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. UIROBOT disclaims all liability arising from this information and its use. Use of UIROBOT Products in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless UIROBOT from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any UIROBOT intellectual property rights.

[Trademark and Design Statement]

- UIROBOT's name and logo is a registered trademark of UIROBOT in China and other countries or regions.
- The appearance designs of UIROBOT's products have applied for patent protection.

[Contact information]

United Intelligence Robotics (Zhejiang)Co.,Ltd.

Addr: Bldg B3, Innovation Park, Pinghu, Jiaxing City, Zhejiang Province, China 314200

Tel: (086)0573-85235951

Email: support@uirobot-zj.com

Web: www.uirobot.com

Catalog

1.0	Programming Guides.....	5
1.1	Function List.....	6
1.2	Key Data Structures	9
2.0	CAN NETWORK.....	11
2.1	Open CAN Network.....	11
2.2	Close CAN Network	13
2.3	Get CAN Gateway Information.....	14
2.4	Get CAN Member Information List	16
2.5	Set ETH Gateway IP Address	18
3.0	COMMUNICATION PARAMETERS.....	19
3.1	Set Communication Parameters	19
3.2	Get Communication Parameters	21
4.0	SYSTEM INFORMATION	23
4.1	Getserial number.....	23
4.2	Get Device Model.....	24
4.3	Get Error Report.....	25
4.4	Clear Error Report.....	27
4.5	Get the Latest Error.....	28
4.6	Get Motion Status	29
4.7	Reset Motion Status	31
4.8	Get System Power-On Time	33
4.9	Reset System Power-On Time.....	34
4.10	System Operation	35
5.0	INITIAL CONFIG.....	36
5.1	Set Initial Config	36
5.2	Get Initial Config.....	38
6.0	INFORMATION ENABLE	40
6.1	Set Information Enable.....	40
6.2	Get Information Enable	42
7.0	MOTION CONTROL STATIC PARAMETERS	44
7.1	Set Motor Config	44
7.2	Get Motor Config.....	46
7.3	Set Acceleration	48
7.4	Get Acceleration.....	49
7.5	Set Deceleration.....	50
7.6	Get Deceleration	51
7.7	Set Cut-In Speed.....	52
7.8	Get Cut-In speed.....	53
7.9	Set Emergency Stop Deceleration	54
7.10	Get Emergency Stop Deceleration	55
7.11	Set Encoder Parameters	56
7.12	Get Encoder Parameters	57
7.13	Get Auxiliary Location Information	59
7.14	Set Backlash Compensation Value	60
7.15	Get Backlash Compensation Value.....	61
7.16	Set Motion Limits.....	62
7.17	Get Motion Limits	64
7.18	Reset Motion Limits.....	66
7.19	Set Limit Check Status.....	67
7.20	Get Limit Check Status	68

uirSDK3.0 Control System Development Kit

7.21	Get Desired Value.....	69
8.0	MOTION CONTROL DYNAMIC PARAMETERS	70
8.1	Set Motor Enable Status	70
8.2	Get Motor Enable Status	72
8.3	Set Motor Being Motion.....	73
8.4	Set Motor Being Motion With Time.....	74
8.5	Set Motor Emergency Stop	75
8.6	Set Origin Position	77
8.7	Set Jog Motion Speed	78
8.8	Get Jog Motion Speed	79
8.9	Set PTP Motion Absolute Position	80
8.10	Get PTP Motion Absolute Position	82
8.11	Set PTP Motion Relative Position	83
8.12	Get PTP Motion Relative Position	85
9.0	I/O OPERATIONS.....	86
9.1	Set Port Trigger Mode.....	86
9.2	Get Port Trigger Mode	88
9.3	Set Port Digital Output	89
9.4	Get IO Port Digital Status.....	96
9.5	Set Input Logic Acceleration.....	98
9.6	Get Input Logic Acceleration	100
9.7	Set Input Logic Deceleration	102
9.8	Get Input Logic Deceleration.....	104
9.9	Set Input Logic Speed	106
9.10	Get Input Logic Speed	108
9.11	Set Input Logic Relative Position.....	110
9.12	Get Input Logic Relative Position	112
9.13	Set Input Logic Absolute Position.....	114
9.14	Get Input Logic Absolute Position	116
9.15	Set Input Logic	118
9.16	Get Input Logic.....	121
10.0	IO MODULE.....	123
10.1	Set Auxiliary Encoder Period.....	123
10.2	Reset Auxiliary Encoder Count	125
10.3	Get Auxiliary Encoder Speed	126
10.4	Get Auxiliary Encoder Count.....	127
10.5	Get Auxiliary Encoder Increment.....	128
11.0	OTHER INSTRUCTIONS.....	129
11.1	Real-Time Notification	129
12.0	PROGRAMMING EXAMPLES	131
12.1	C ++ routine description	131
12.2	C# routine description	131
12.3	Program writing process description	131

1.0 PROGRAMMING GUIDES

Welcome to uirSDK 3.0 Programming Guide.

With the help of this document, you can learn how to program on Windows systems through the dynamic library files and header files of uirSDK 3.0.

This programming manual introduces the calling functions and data structures of UIM342, UIM620, and UIM720 series products. Users must also refer to the UIM342 User Manual, UIM620 User Manual, and UIM720 User Manual to deepen their understanding of the product.

You can download the uirSDK package at <http://www.uirobot.com>.

Files included in the uirSDK development package:

/ uirSDKfiles

CAN30SDK.h (static library function API)

CAN30SDK_DLL.h (dynamic library function API)

UIError.h (error type definition)

uirSDKgen3.h (contains all structures and type definitions)

/ UirSdkLib

libUirSdk.so (static library: For Linux)

libUirSdkWin32.dll (dynamic library: window system 32-bit version)

libUirSdkWin64.dll (dynamic library: window system 64-bit version)

libUirSdkWin32.lib (static library: window system 32-bit version)

libUirSdkWin64.lib (static library: window system 32-bit version)

libUirSdkWin32.dll is suitable for the 32-bit version of the window system, and libUirSdkWin64.dll is suitable for the 64-bit version of the window system. Please choose one according to your computer system version.

uirSDK3.0 Control System Development Kit

1.1 Function List

Category	Functions	Note
CAN NETWORK	SdkStartCanNet	Open CAN Network
	SdkCloseCanNet	Close Can Network
	SdkGetGtwyInfo	Get Can Gateway Information
	SdkGetMemberList	Get Can Member Information List
	SdkSetIPAddr	Set Eth Gateway Ip Address
COMMUNICATION PARAMETERS	SdkSetProtocolPara	Set Communication Parameters
	SdkGetProtocolPara	Get Communication Parameters
SYSTEM INFORMATION	SdkGetSerial	Getserial Number
	SdkGetModel	Get Device Model
	SdkGetErrReport	Get Error Report
	SdkClrErrReport	Clear Error Report
	SdkGetLastErr	Get Last Error Report
	SdkGetMotionStatus	Get Motion Status
	SdkRstMotionStatus	Reset Motion Status
	SdkGetSystemTime	Get System Power-On Time
	SdkRstSystemTime	Reset System Power-On Time
	SdkSetSystem	System Operation
INITIAL CONFIG	SdkSetInitialConfig	Setinitial Config
	SdkGetInitialConfig	Getinitial Config
INFORMATION ENABLE	SdkSetInformEnable	Setinformation Enable
	SdkGetInformEnable	Getinformation Enable
MOTION CONTROL STATIC PARAMETERS	SdkSetMotorConfig	Set Motor Config
	SdkGetMotorConfig	Get Motor Config
	SdkSetAcceleration	Set Acceleration
	SdkGetAcceleration	Get Acceleration
	SdkSetDeceleration	Set Deceleration

uirSDK3.0 Control System Development Kit

	SdkGetDeceleration	Get Deceleration
	SdkSetCutInSpeed	Set Cut-In Speed
	SdkGetCutInSpeed	Get Cut-In speed
	SdkSetStopDeceleration	Set Emergency Stop Deceleration
	SdkGetStopDeceleration	Get Emergency Stop Deceleration
	SdkSetEncoderConfig	Set Encoder Parameters
	SdkGetEncoderConfig	Get Encoder Parameters
	SdkGetAuxPosition	Get Auxiliary Location Information
	SdkSetBacklashComp	Set Backlash Compensation Value
	SdkGetBacklashComp	Get Backlash Compensation Value
	SdkSetMotionLimits	Set Motion Limits
	SdkGetMotionLimits	Get Motion Limits
	SdkRstMotionLimits	Reset Motion Limits
	SdkSetLimitCheck	Set Limit Check Status
	SdkGetLimitCheck	Get Limit Check Status
	SdkGetDesiredValue	Get Desired Value
MOTION CONTROL DYNAMIC PARAMETERS	SdkSetMotorOn	Setmotor Enable Status
	SdkGetMotorOn	Getmotor Enable Status
	SdkSetBeginMxn	Set Motor Being Motion
	SdkSetStopMxn	Set Motor Emergency Stop
	SdkSetOrigin	Set Origin Position
	SdkSetJogMxn	Set Jog Motion Speed
	SdkGetJogMxn	Get Jog Motion Speed
	SdkSetPtpMxnA	Set Ptp Motion Absolute Position
	SdkGetPtpMxnA	Get Ptp Motion Absolute Position
	SdkSetPtpMxnR	Set Ptp Motion Relative Position
	SdkGetPtpMxnR	Get Ptp Motion Relative Position
I/O OPERATIONS	SdkSetTriggerMode	Set Port Trigger Mode

uirSDK3.0 Control System Development Kit

	SdkGetTriggerMode	Get Port Trigger Mode
	SdkSetDigitalOutput	Set Port Digital Output
	SdkGetDIOport	Get IO Port Digital Status
	SdkSetInputLogicAC	Set Input Logic Acceleration
	SdkGetInputLogicAC	Get Input Logic Acceleration
	SdkSetInputLogicDC	Set Input Logic Deceleration
	SdkGetInputLogicDC	Get Input Logic Deceleration
	SdkSetInputLogicSP	Set Input Logic Speed
	SdkGetInputLogicSP	Get Input Logic Speed
	SdkSetInputLogicPR	Set Input Logic Relative Position
	SdkGetInputLogicPR	Get Input Logic Relative Position
	SdkSetInputLogicPA	Set Input Logic Absolute Position
	SdkGetInputLogicPA	Get Input Logic Absolute Position
	SdkSetInputLogic	Set Input Logic
	SdkGetInputLogic	Get Input Logic
IO MODULE	SdkSetEncoderPeriod	Set Auxiliary Encoder Period
	SdkRstEncoderCount	Reset Auxiliary Encoder Count
	SdkGetEncoderSpeed	Get Auxiliary Encoder Speed
	SdkGetEncoderCount	Get Auxiliary Encoder Count
	SdkGetEncoderIncrement	Get Auxiliary Encoder Increment
OTHER INSTRUCTIONS	SdkGetRTCN	Real-Time Notification

uirSDK3.0 Control System Development Kit

1.2 Key Data Structures

Gateway Info Structure:

```
1. ///  
2. typedef union  _GATEWAY_INFO_OBJ  
3. {  
4.     struct  
5.     {  
6.         uint GtwyHandle;    // Gateway handle (including CAN network  
           type/network segment number/CAN-ID)  
7.         uint COMidx;        // serial port number  
8.         uint COMbaud;       // Baud rate (for serial devices)  
9.         uint BTRidx;        // Gateway CAN bit rate  
10.        uint SerialNo;      // Gateway SN number  
11.        uint FirmVer;       // Gateway firmware version  
12.        char  ModelStr[64]; // Gateway model information  
13.        uint  MemberQty;     // Number of Members  
14.        uchar MemberCid[100]; // Members CAN-ID List  
15.    };  
16.  
17. struct  
18. {  
19.     uint GtwyHandle;    // Gateway handle (including CAN network  
           type/network segment number/CAN-ID)  
20.     uint IPaddr;        // IP ADDRESS, e.g. 192.168.1.254  
21.     uint IPport;        // IP PORT, e.g. 8888  
22.     uint BTRidx;        // Gateway CAN bit rate  
23.     uint SerialNo;      // Gateway SN number  
24.     uint FirmVer;       // Gateway firmware version  
25.     char  ModelStr[64];  
26.     uint  MemberQty;     // Number of Members  
27.     uchar MemberCid[100]; // Members CAN-ID List  
28. }tcp;  
29.  
30. struct  
31. {  
32.     uchar  CANNid : 8; // Gateway CAN-ID  
33.     uchar  zone : 8;  // CAN SEGMENT NUMBER  
34.     ushort type : 16; // Gateway Type ((i.e.CAN20/CAN30)  
35.     uchar  IP[4]; // IP  
           ADDRESS: IP[3]=192, IP[2]=168, IP[1]=1, IP[0]=100 -> ip = 192.168.1.100  
36.     uint IPport;        // IP PORT, e.g. 8888  
37.     uint BTRidx;        // Gateway CAN bit rate  
38.     uint SerialNo;      // Gateway SN number  
39.     uint FirmVer;       // Gateway firmware version  
40.     char  ModelStr[64];  
41.     uint  MemberQty;     // Number of Members  
42.     uchar MemberCid[100]; // Members CAN-ID List  
43. }detail;
```

uirSDK3.0 Control System Development Kit

```
44. }GATEWAY_INFO_OBJ;
```

Gateway Member Data Structure:

```
1. //[ Member ]
2. typedef struct _MEMBER_INFO_OBJ
3. {
4.     uint    GtwyHandle;    // Gateway handle (including CAN network
        type/network segment number/CAN-ID).
5.     uchar   CANNid;       // Self CAN-ID
6.     uchar   CANGid;       // Group CAN-ID
7.     ushort  COMbaud;       // Baud rate (for serial devices)
8.     uint    SerialNo;      // Device serial number
9.     MODEL_INFO_OBJ Model;  // Member model (see Member Model Data
        Structure).
10.} MEMBER_INFO_OBJ, DRV_INFO_OBJ;
```

Member Model Data Structure:

```
1. //[ Member ] Model Info.
2. typedef struct _MODEL_INFO_OBJ
3. {
4.     uchar   Main;         // The main part number of the device
5.     uchar   SubModel;     // The sub-part of the device
6.     uchar   FuncSpec;     // Functional description of the device
7.     uchar   MaxAmp;       // Maximum drive current
8.     uint    FirmVer;      // Firmware version
9.     uint    PCBVer;       // PCB version
10.
11.     char    ModelStr[32]; // Model string
12.} MODEL_INFO_OBJ;
```

Universal Interface Message Data Structure:

```
1. //[ UIR MSG ] UIROBOT's Universal Interface Message
2. typedef struct _UI_MSG_OBJ
3. {
4.     uchar   id;           //CANid; Controller (including gateway) CAN-ID;
5.     uchar   cw;           //Control Word;
6.     uchar   dl;           //Data Length;
7.     uchar   data[9];      //Data Array;
8.
9.     uint    ix;           //Sub-Index, Resreved;
} UI_MSG_OBJ;
```

2.0 CAN NETWORK

This chapter will introduce how to open and close the CAN network and obtain member information through function calls related to the CAN network. Open CAN Network is the first step for you to control the motor.

2.1 Open CAN Network

Function Name:

ERR0

```
SdkStartCanNet(GATEWAY_INFO_OBJ* pGtwy, MEMBER_INFO_OBJ* pMember, uint UseConstLink = 0);
```

Parameter List:

Parameter	Type	Required	Note
pGtwy	GATEWAY_INFO_OBJ	yes	Gateway information (Refer to uirSDKcommon.h)
pMember	MEMBER_INFO_OBJ	yes	Member information (Refer to uirSDKcommon.h)
UseConstLink	uint	no	User Connection Mode

Legal Value:

Parameter	Legal Value	Note
pGtwy		Gateway information (refer to chapter 1.2)
pMember		Member information (refer to chapter 1.2)
UseConstLink	0(default)	Normal Mode
	1	Debug Mode

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. bool m_bIs232Gtwy = true;
2. bool m_bIsEthGtwy = false;
3. GATEWAY_INFO_OBJ Gtwy;
4. if (m_bIs232Gtwy)
5. {
6.     Gtwy.detail.type = UIGW3_GW2513;
7.     Gtwy.COMidx = 7;
8.     Gtwy.COMbaud = 57600;
9. }
10. else if (m_bIsEthGtwy)
11. {
12.     Gtwy.detail.type = UIGW2_ETHCAN;
13.     Gtwy.detail.IP[0] = 254;
14.     Gtwy.detail.IP[1] = 1;
15.     Gtwy.detail.IP[2] = 168;
16.     Gtwy.detail.IP[3] = 192;
17.     if (!Gtwy.tcp.IPaddr)
18.     {
19.         std::cout << "Wrong gateway parameter!\n";
20.         return -1;
21.     }
22.     Gtwy.tcp.IPport = 8888;
23. }
24.
25. MEMBER_INFO_OBJ Member[100];    // Members (Max. 100 members)
26. ERRO err = 0;
27. err = SdkStartCanNet(&Gtwy, Member, 0);
28. if (err)
29. {
30.     std::cout << "No gateway found, please check the gateway
        communication parameters!\n";
31.     return -1;
32. }
33. // Number of members
34. int MemberQuantity = 0;          // Number of Members
35. MemberQuantity = Gtwy.MemberQty;
36.
37. for (int i = MemberQuantity; i > 0; i--)
38. {
39.     std::cout << "CAN id:" << (int)Member[i - 1].CANnid
40.         << "Model:" << Member[i - 1].Model.ModelStr
41.         << "Firmware:" << Member[i - 1].Model.FirmVer
42.         << std::endl;
43. }
44. g_GtwyHandle = Gtwy.GtwyHandle;    // Get the Gateway Handle
```

uirSDK3.0 Control System Development Kit

2.2 Close CAN Network

Function Name: `ERRO SdkCloseCanNet(uint GtwyHandle);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle(Refer to chapter 2.1)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. uint g_GtwyHandle;           //Define Gateway Handle
2. ERRO err                     //Define error type
3. GATEWAY_INFO_OBJ m_Gtwy = { 0 }; //Gateway information Initialization
4. MEMBER_INFO_OBJ m_Member[100] = { 0 }; //Member information Initialization
5.
6. //Open CAN Network
7. err = SdkStartCanNet(&m_Gtwy, m_Member, 0);
8. if (err)
9. {
10.     std::cout << "Open CAN Network Failed!\n";
11.     return -1;
12. }
13.
14. g_GtwyHandle = m_Gtwy.GtwyHandle; //Get the Gateway Handle
15.
16. //Close CAN Network
17. if (m_Gtwy.GtwyHandle)
18. {
19.     err = SdkCloseCanNet(g_GtwyHandle);
20.     if (err)
21.     {
22.         std::cout << "Close CAN Network Failed!\n";
23.         return -1;
24.     }
25. }
```

uirSDK3.0 Control System Development Kit

2.3 Get CAN Gateway Information

Function Name: **ERR0** `SdkGetGtwyInfo(GATEWAY_INFO_OBJ* pGtwy, MEMBER_INFO_OBJ* pMember);`

Parameter List:

Parameter	Type	Required	Note
pGtwy	GATEWAY_INFO_OBJ	yes	Gateway information (Refer to uirSDKcommon.h)
pMember	MEMBER_INFO_OBJ	yes	Member information (Refer to uirSDKcommon.h)

Legal Value:

Parameter	Note
pGtwy	Gateway information (refer to chapter 2.1)
pMember	Member information (refer to chapter 2.1)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. //first step: open can network, refer to 2.1 Open CAN Network
2. GATEWAY_INFO_OBJ pGtwy; //Define Gateway Handle
3. MEMBER_INFO_OBJ pMember; // Define Member information
4.
5. //Define parameters
6. int m_MemberQuantity;
7. string m_ModelStr;
8. uint g_GtwyHandle, g_GtwyCOMidx, g_GtwyCOMbaud,
9.   g_GtwyBTRidx, g_GtwySerialNo, g_GtwyFirmVer;
10.
11. //GetCANGateway information
12. ERRO err = SdkGetGtwyInfo(&pGtwy, &pMember);
13. if (err)
14. {
15.     std::cout << "Get CAN Gateway information Failed!\n";
16.     return;
17. }
18.
19. g_GtwyHandle = pGtwy.GtwyHandle; //Gateway Handle
20. g_GtwyCOMidx = pGtwy.COMidx; //Serial Port Number
21. g_GtwyCOMbaud = pGtwy.COMbaud; //Serial port baud rate
22. g_GtwyBTRidx = pGtwy.BTRidx; //CAN BIT RATE
23. g_GtwySerialNo = pGtwy.SerialNo; //Gateway S/N Number
24. g_GtwyFirmVer = pGtwy.FirmVer; //Gateway Firmware Version
25. m_ModelStr = pGtwy.ModelStr; //Gateway Model Information
26. m_MemberQuantity = pGtwy.MemberQty; //Number of Members
27. //.....Get other struct information as above
```

uirSDK3.0 Control System Development Kit

2.4 Get CAN Member Information List

Function Name: `ERRRO SdkGetMemberList(uint GtwyHandle, MEMBER_INFO_OBJ* pMember);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
pMember	MEMBER_INFO_OBJ	yes	Member information (Refer to uirSDKcommon.h)

Legal Value:

Parameter	Note
pMember	Member information(refer to chapter 2.1)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. GATEWAY_INFO_OBJ pGtwy; //Define Gateway Info
2. MEMBER_INFO_OBJ pMember[100]; //Define Member info
3.
4. int m_MemberQuantity; //Define parameters
5. //GetCANGateway information
6. ERRO err = SdkGetGtwyInfo(&pGtwy, pMember);
7. if (err)
8. {
9.     std::cout << "Get CAN Gateway information Failed!\n";
10.    return;
11. }
12.
13. m_MemberQuantity = pGtwy.MemberQty; //Number of Members
14.
15. //Get CAN Member information
16. err = SdkGetMemberList(g_GtwyHandle, pMember);
17. if (err)
18. {
19.     std::cout << "Get CAN Member information Failed!\n";
20.     return;
21. }
22.
23. //print Member information
24. for (int i = m_MemberQuantity; i > 0; i--)
25. {
26.     std::cout
27.     << "CAN id:" << (int)pMember[i - 1].CANnid
28.     << "Model:" << pMember[i - 1].Model.ModelStr
29.     << "FirmVer:" << pMember[i - 1].Model.FirmVer
30.     << std::endl;
31. }
```

uirSDK3.0 Control System Development Kit

2.5 Set ETH Gateway IP Address

Function Name: `ERRO SdkSetIPAddr(uint GtwyHandle, uchar* IPAddr);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
IPAddr	uchar	yes	IP address (string)

Legal Value:

Parameter	Note
IPAddr	IP ADDRESS (STRING), example: "192.168.1.25" After modification, you need to use the new address to open the ETH gateway

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. uchar IPAddr[4] = {25, 1, 168, 192};
2.
3. //g_GtwyHandle Get the gateway handle, see 2.1 Opening the CAN network
4. ERRO err = SdkSetIPAddr(g_GtwyHandle, IPAddr);
5. if (err)
6. {
7.     std::cout << "Set IPAddr Failed!\n";
8.     return;
9. }
```

3.0 COMMUNICATION PARAMETERS

This chapter will introduce how to modify the communication parameters of the CAN network through functions, which will help you configure according to the actual situation of the device.

3.1 Set Communication Parameters

Function Name:

```
ERRNO SdkSetProtocolPara(uint GtwyHandle, uint CANid, uint index, uint data, uint* pRxdata = NULL);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
index	uint	yes	Parameter index number
data	uint	yes	Parameter value
pRxdata(Return Value)	uint	yes	Result feedback

Legal Value:

Parameter	Legal Value	data	Note
	0 (PPS_SysUpdate)	0/1	System Update
	1 (PPS_BDR)		Serial port baud rate
index	5 (PPS_BTR)	0: 1000K 1: 800K 2: 500K 3: 250K 4: 125K	CAN bit rate
	7 (PPS_NodeID)	5...126	Device's ID (Can't be repeated with Group ID)
	8 (PPS_GroupID)	5...126	Group ID (Can't be repeated with Device's ID)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint pRxdata; //Return value definition
3. uint NodeID = 5; //Device's ID
4.
5. //modify Device's ID, 5->10, To modify the Device's ID, you need to power
   off and restart to take effect.
6. err = SdkSetProtocolPara(g_GtwyHandle, NodeID,PPS_NodeID,10, &pRxdata);
7. if (err)
8. {
9.     std::cout << "Modify Device's ID Failed!\n";
10.    return;
11. }
```

uirSDK3.0 Control System Development Kit

3.2 Get Communication Parameters

Function Name:

```
ERR0 SdkGetProtocolPara(uint GtwyHandle, uint CANid, uint index, uint* pRxdata);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
index	uint	yes	Parameter index number
pRxdata(Return Value)	uint	yes	Result feedback

Legal Value:

Parameter	Legal Value	pRxdata	Note
	0 (PPS_SysUpdate)	0/1	System Update
	1 (PPS_BDR)		Serial port baud rate
index	5 (PPS_BTR)	0: 1000K 1: 800K 2: 500K 3: 250K 4: 125K	CAN BIT RATE
		7 (PPS_NodeID)	Device's ID(Can't be repeated with Group ID)
		8 (PPS_GroupID)	Group ID(Can' t be repeated with Device' s ID)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1.  ERRO err = 0;           //Error definition
2.  uint pRxdata;           //Return value definition
3.  uint NodeID = 5;         //Device's ID
4.  uint Can_BTR;           //CAN bitrate value
5.
6.  //g_GtwyHandle Get the gateway handle, see 2.1 Opening the CAN network
7.
8.  err = SdkGetProtocolPara(g_GtwyHandle, NodeID, PPS_BTR, &pRxdata);
9.  if (err)
10. {
11.     std::cout << "Get Bitrate failed!\n";
12.     return;
13. }
14.
15. Can_BTR = pRxdata; //Get CAN bitrate value
16.
17. switch (Can_BTR)
18. {
19. case 0:
20.     std::cout << "Bitrate = 1000K\n";
21.     break;
22. case 1:
23.     std::cout << "Bitrate = 800K\n";
24.     break;
25. case 2:
26.     std::cout << "Bitrate = 500K\n";
27.     break;
28. case 3:
29.     std::cout << "Bitrate = 250K\n";
30.     break;
31. case 4:
32.     std::cout << "Bitrate = 125K\n";
33.     break;
34. default:
35.     break;
36. }
```

4.0 SYSTEM INFORMATION

This chapter will introduce how to Get or Set device parameters through functions.

4.1 Getserial number

Function Name: `ERROR SdkGetSerial(uint GtwyHandle, uint CANid, uint * pSerialNumber);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pSerialNumberOut(Return Value)	uint	yes	Serial Number feedback

Legal Value:

Parameter	Legal Value	Note
pSerialNumberOut	Supports up to 10 digits (decimal)	Serial Number feedback

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. uint pRxdata; //Return value definition
2. uint NodeID = 5; //Device's ID
3. uint SerialNumber; //serial number
4. uint SNOout; //Serial number return value
5.
6. ERROR err = SdkGetSerial(g_GtwyHandle, NodeID, &SNOout);
7. if (err)
8. {
9.     std::cout << "Get serial number Failed!\n";
10.    return;
11. }
12.
13. SerialNumber = SNOout; // serial number
```

uirSDK3.0 Control System Development Kit

4.2 Get Device Model

Function Name: `ERRRO SdkGetModel(uint GtwyHandle, uint CANid, MODEL_INFO_OBJ* pModelInfo);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pModelInfo(Return Value)	MODEL_INFO_OBJ	yes	Device Model feedback

Legal Value:

Parameter	Legal Value	Note
pModelInfo	(Refer to chapter 1.2)	Device Model feedback

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. uint NodeID = 5;    // Device's ID
2.
3. // g_GtwyHandle Get the gateway handle, see 2.1 Opening the CAN network
4.
5. MODEL_INFO_OBJ modelinfo;
6. ERRRO err = SdkGetModel(g_GtwyHandle, NodeID, &modelinfo);
7. if (err)
8. {
9.     std::cout << "Get Device ModelFailed!\n";
10.    return;
11. }
```


uirSDK3.0 Control System Development Kit

4.3 Get Error Report

Function Name:

```
ERR0 SdkGetErrReport(uint GtwyHandle, uint CANid, uint ErrIdx, ERR_REPORT_OBJ* pRxData);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
ErrIdx	uint	yes	historical error record index
pRxData(Return Value)	ERR_REPORT_OBJ	yes	error report (Refer to uirSDKcommon.h)

Legal Value:

Parameter	Attributes	Legal Value	Note
ErrIdx		10....17	10....17 (newest...oldest)
pRxData	index		error index(0...7 newest...oldest)
	ecode		error code
	errcw		Error related "command code/control code"
	errix		Error related " command auxiliary code / control auxiliary code "

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. uint NodeID = 5; //Device's ID
2.
3. //Get Error Report
4. uchar Errcode[32]; //Error message
5. uint ErrIdx = 10; //Get the latest error
6. ERR_REPORT_OBJ pErrData;
7.
8. ERRO err = SdkGetErrReport(g_GtwyHandle, NodeID, ErrIdx, &pErrData);
9. if (err)
10. {
11.     std::cout << "Get Error Report Failed!\n";
12.     return;
13. }
14.
15. Errcode[0] = pErrData.index; //error index
16. Errcode[1] = pErrData.ecode; //error code
17. Errcode[2] = pErrData.errcw; //Error related "command code/control code"
18. Errcode[3] = pErrData.errix; //Error related "command auxiliary code /
    control auxiliary code "
```

uirSDK3.0 Control System Development Kit

4.4 Clear Error Report

Function Name: `ERRO SdkClrErrReport(uint GtwyHandle, uint CANid, ERR_REPORT_OBJ* pRxData);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pRxData (Return Value)	ERR_REPORT_OBJ	yes	error report (Refer to uirSDKcommon.h)

Legal Value:

Parameter	Legal Value	Note
pRxData	0/!0	Return 0, indicating that the error has been cleared Return !0, there is an error (See 4.4 Get Error Reports legal value description)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint pRxdata; //Return value definition
3. uint NodeID = 5; //Device's ID
4.
5. ERR_REPORT_OBJ pErrData;
6. //Clear Error Report
7. err = SdkClrErrReport(g_GtwyHandle, NodeID, &pErrData);
8. if (err)
9. {
10.     std::cout << "Clear Error Report Failed!\n";
11.     return;
12. }
```

uirSDK3.0 Control System Development Kit

4.5 Get the Latest Error

Function Name: `ERR0 SdkGetLastError(uint GtwyHandle, uint CANid, ERR_REPORT_OBJ* pRxData);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pRxData(Return Value)	ERR_REPORT_OBJ	yes	error report (Refer to uirSDKcommon.h)

Legal Value:

Parameter	Legal Value	Note
pRxData	0/!0	Return 0, indicating that the error has been cleared Return !0, there is an error (See 4.4 Get Error Reports legal value description)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. uint pRxdata; //Return value definition
2. uint NodeID = 5; //Device's ID
3. uchar Errcode[32]; //Error message
4. ERR_REPORT_OBJ pErrData;
5. //Get the Latest Error
6. err = SdkGetLastError(g_GtwyHandle, NodeID, &pErrData);
7. if (err)
8. {
9.     std::cout << "Get the Latest Error Failed!\n";
10.    return;
11. }
12.
13. Errcode[0] = pErrData.index; //error index
14. Errcode[1] = pErrData.ecode; //error code
15. Errcode[2] = pErrData.errcw; //Error related "command code/control code"
16. Errcode[3] = pErrData.errix; //Error related "command auxiliary code /
    control auxiliary code "
```

uirSDK3.0 Control System Development Kit

4.6 Get Motion Status

Function Name:

```
ERRO SdkGetMotionStatus(uint GtwyHandle, uint CANid, MOTION_STATUS_OBJ* pStatus, int* pVe, int* pPr, int* pPa);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pStatus (Return Value)	MOTION_STATUS_OBJ	yes	Motion Status (Refer to uirSDKcommon.h)
pVe (Return Value)	int	yes	Motor speed
pPr (Return Value)	int	yes	Motor displacement (relative position)
pPa (Return Value)	int	yes	Motor position (absolute position)

Legal Value:

Parameter	Attributes	Legal Value	Note
pStatus	bits.MODE	0/1/2	Motion mode, 0=JOG, 1=PTP, 2=PVT
	bits.SVON	0/1	Motor enable status, 1=enabled, 0=offline
	bits.IN0LV	0/1	IN0 level
	bits.IN1LV	0/1	IN1 level
	bits.IN2LV	0/1	IN2 level
	bits.QP0LV	0/1	OP0 level
	bits.QP1LV	0/1	OP1 level
	bits.STOP	0/1	Motor speed is zero
	bits.TCIF	0/1	Torque exceeds limit
	bits.HEAT	0/1	Temperature exceeds limit
	bits.LOCK	0/1	System locked
	bits.GALM	0/1	1=Crew alert sent (Note: Passive receiver=0)
	bits.SERR	0/1	1=A system error occurred

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1.  ERRO err = 0; //Error definition
2.  uint NodeID = 5; //Device's ID
3.
4.  MOTION_STATUS_OBJ MotorStatus; //Motion Status information feedback
5.  int Ve, Pr, Pa; //Motor speed and position parameter feedback
6.
7.  unsigned MStatus[32]; //Storage Motion Status
8.  int Motor_Speed, RelativePositon, AbsPostion; //Storage motor speed and
   position parameters
9.
10. //Get Motion Status
11. err = SdkGetMotionStatus(g_GtwyHandle, NodeID, &MotorStatus, &Ve, &Pr, &Pa);
12. if (err)
13. {
14.     std::cout << "Get Motion Status Failed!\n";
15.     return;
16. }
17.
18. MStatus[0] = MotorStatus.bits.MODE; //Motion mode, 0=JOG, 1=PTP, 2=PVT
19. MStatus[1] = MotorStatus.bits.SVON; //Motor enable status, 1=enabled,
   0=offline
20. MStatus[2] = MotorStatus.bits.IN0LV; //IN0 Level
21. MStatus[3] = MotorStatus.bits.IN1LV; //IN1 Level
22. MStatus[4] = MotorStatus.bits.IN2LV; //IN2 Level
23. MStatus[5] = MotorStatus.bits.QP0LV; //OP0 Level
24. MStatus[6] = MotorStatus.bits.QP1LV; //OP1 Level
25. MStatus[7] = MotorStatus.bits.STOP; //Motor speed is zero
26. MStatus[8] = MotorStatus.bits.TCIF; //Torque exceeds limit
27. MStatus[9] = MotorStatus.bits.HEAT; //Temperature exceeds limit
28. MStatus[10] = MotorStatus.bits.LOCK; //System Locked
29. MStatus[11] = MotorStatus.bits.GALM; //1=Crew alert sent
30. MStatus[12] = MotorStatus.bits.SERR; //1=A system error occurred
31.
32. Motor_Speed = Ve; //Current motor running speed
33. RelativePositon = Pr; // Current motor displacement
34. AbsPostion = Pa; //Current motor position
```

uirSDK3.0 Control System Development Kit

4.7 Reset Motion Status

Function Name: **ERR0** `SdkRstMotionStatus`(uint GtwyHandle, uint CANid, MOTION_STATUS_OBJ* pStatus);

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pStatus (Return Value)	MOTION_STATUS_OBJ	yes	Motion Status (Refer to uirSDKcommon.h)

Legal Value:

Parameter	Note
pStatus	Motion Status(Refer to Chapter 4.6)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3.
4. //g_GtwyHandle Get the gateway handle, see 2.1 Opening the CAN network
5.
6. MOTION_STATUS_OBJ MotorStatus; //Motion Status information feedback
7.
8. int Ve, Pr, Pa; //Motor speed and position parameter feedback
9.
10. unsigned MStatus[32]; //Storage Motion Status
11.
12. //Reset Motion Status
13. err = SdkRstMotionStatus(g_GtwyHandle, NodeID, &MotorStatus);
14. if (err)
15. {
16.     std::cout << "Reset Motion Status Failed!\n";
17.     return;
18. }
19.
20. MStatus[0] = MotorStatus.bits.MODE; //Motion mode, 0=JOG, 1=PTP, 2=PVT
21. MStatus[1] = MotorStatus.bits.SVON; //Motor enable status, 1=enabled,
    0=offline
22. MStatus[2] = MotorStatus.bits.IN0LV; //IN0 Level
23. MStatus[3] = MotorStatus.bits.IN1LV; //IN1 Level
24. MStatus[4] = MotorStatus.bits.IN2LV; //IN2 Level
25. MStatus[5] = MotorStatus.bits.QP0LV; //OP0 Level
26. MStatus[6] = MotorStatus.bits.QP1LV; //OP1 Level
27. MStatus[7] = MotorStatus.bits.STOP; //Motor speed is zero
28. MStatus[8] = MotorStatus.bits.TCIF; //Torque exceeds limit
29. MStatus[9] = MotorStatus.bits.HEAT; //Temperature exceeds limit
30. MStatus[10] = MotorStatus.bits.LOCK; //System Locked
31. MStatus[11] = MotorStatus.bits.GALM; //1=机组警报(注: 被动接收者=0)
32. MStatus[12] = MotorStatus.bits.SERR; //1=系统错误
```


uirSDK3.0 Control System Development Kit

4.8 Get System Power-On Time

Function Name: `ERRO SdkGetSystemTime(uint GtwyHandle, uint CANid, uint * pPoweOnTime);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pPoweOnTime (Return Value)	uint	yes	power-on duration feedback (unit: us)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. uint NodeID = 5; //Device's ID
2.
3. uint PowTime,pPoweOnTime;
4. //Getpower on time
5. ERRO err = SdkGetSystemTime(g_GtwyHandle, NodeID,&pPoweOnTime);
6. if (err)
7. {
8.     std::cout << "Get System Power-On Time Failed!\n";
9.     return;
10. }
11.
12. PowTime = pPoweOnTime; // Get the power-on time, unit (us)
```

uirSDK3.0 Control System Development Kit

4.9 Reset System Power-On Time

Function Name: `ERRO SdkRstSystemTime(uint GtwyHandle, uint CANid, uint * piPoweOnTime);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pPoweOnTime (Return Value)	uint	yes	power-on duration feedback (unit: us)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. uint NodeID = 5; //Device's ID
2.
3. uint pPoweOnTime;
4.
5. //Reset System Power-On Time
6. ERRO err = SdkRstSystemTime(g_GtwyHandle, NodeID, &pPoweOnTime);
7. if (err)
8. {
9.     std::cout << "Reset System Power-On Time Failed!\n";
10.    return;
11. }
```

uirSDK3.0 Control System Development Kit

4.10 System Operation

Function Name:

```
ERRO SdkSetSystem(uint GtwyHandle, uint CANid, uint index, uint CfgData, uchar* pRxData = NULL);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
index	uint	yes	System operating parameter index
CfgData	uint	yes	System operating parameter content
pRxData(Return Value)	uchar	no	content feedback

Legal Value:

Parameter	Legal Value	CfgData	Note
index	0 (_SY_ULK_SYS)	0/1	Unlock System (620/720/630/730 only)
	1 (_SY_RST_SYS)	0/1	Reset System
	2 (_SY_RST_DFL)	0/1	Factory settings, Restore Default
	3 (_SY_SYN_ENA)	0/1	Enable sync signal 0/1=disable/enable (2503/MMC900/MMC901 only)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. uint CfgData=1; //Content
4. uchar SysData[32] = {0};
5. //System Operation
6. err = SdkSetSystem(g_GtwyHandle, NodeID, _SY_RST_SYS, CfgData, SysData);
7. if (err)
8. {
9.     std::cout << "System Operation Failed!\n";
10.    return;
11. }
```

5.0 INITIAL CONFIG

This chapter will introduce how to Get or Set device power-on parameters through functions.

5.1 Set Initial Config

Function Name:

```
ERR0 SdkSetInitialConfig(uint GtwyHandle, uint CANid, uint index, uint TxData, uint* pRxData = NULL);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
index	uint	yes	Parameter index number
TxData	uint	yes	Parameter value
pRxdata(Return Value)	uint	no	Result feedback

Legal Value:

Parameter	Legal Value	TxData	Note
index	0	0/1	Power-on enable (0 = disabled, 1 = enabled)
	1	0/1	Forward steering setting (0 = cw, 1 = ccw)
	2	0/1	UPG enabled (0 = disabled, 1 = enabled)
	3	0/1	Input trigger latch (0 = disabled, 1 = enabled)
	4	0/1	Acceleration and deceleration mode (0 = value, 1 = time)
	5	0/1	Encoder type (0 = incremental, 1 = absolute)
	6	0/1	Open and closed loop enable (0 = open loop, 1 = closed loop)
	7	0/1	Software limit enable (0 = disabled, 1 = enabled)
	8	0/1	Brake port (0 = normal, 1 = brake control enabled)
	9	0/1	Brake closed loop (0 = normal, 1 = brake monitoring enabled)
	10	0/1	Forced sleep (0 = normal, 1 = power on and enter sleep)

Return Value:

0	No Error (Success)
---	--------------------

uirSDK3.0 Control System Development Kit

!0	Error Code (refer to UIError.h)
----	---------------------------------

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3.
4.
5. uint ICindex, ICdata, ICdataRx; //Define parameters
6. ICindex = 0; //Configure motor power-on enable
7. ICdata = 1; //Set the power-on enable status to enabled
8.
9. //SetInitial Config
10. err = SdkSetInitialConfig(g_GtwyHandle, NodeID, ICindex, ICdata, &ICdataRx);
11. if (err)
12. {
13.     std::cout << "Set Initial Config Failed!\n";
14.     return;
15. }
```

uirSDK3.0 Control System Development Kit

5.2 Get Initial Config

Function Name:

```
ERR0 SdkGetInitialConfig(uint GtwyHandle, uint CANid, uint index, uint* pRxData);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
index	uint	yes	Parameter index number
pRxdata(Return Value)	uint	yes	Result feedback

Legal Value:

Parameter	Legal Value	pRxdata	Note
index	0	0/1	Power-on enable (0 = disabled, 1 = enabled)
	1	0/1	Forward steering setting (0 = cw, 1 = ccw)
	2	0/1	UPG enabled (0 = disabled, 1 = enabled)
	3	0/1	Input trigger latch (0 = disabled, 1 = enabled)
	4	0/1	Acceleration and deceleration mode (0 = value, 1 = time)
	5	0/1	Encoder type (0 = incremental, 1 = absolute)
	6	0/1	Open and closed loop enable (0 = open loop, 1 = closed loop)
	7	0/1	Software limit enable (0 = disabled, 1 = enabled)
	8	0/1	Brake port (0 = normal, 1 = brake control enabled)
	9	0/1	Brake closed loop (0 = normal, 1 = brake monitoring enabled)
	10	0/1	Forced sleep (0 = normal, 1 = power on and enter sleep)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3.
4. uint ICindex, ICdataRx, MotorDir; //Define parameters
5. ICindex = 1; //Get motor rotation
6.
7. //Get Initial Config
8. err = SdkGetInitialConfig(g_GtwyHandle, NodeID, ICindex, &ICdataRx);
9. if (err)
10. {
11.     std::cout << "GetInitial Config Failed!\n";
12.     return;
13. }
14.
15. MotorDir = ICdataRx; //Motor rotation value (0 = cw clockwise, 1 = ccw
    counterclockwise)
```

6.0 INFORMATION ENABLE

This chapter will introduce how to Enabled/Disabled the Information notification status of the device through functions.

6.1 Set Information Enable

Function Name:

```
ERR0 SdkSetInformEnable(uint GtwyHandle, uint CANid, uint index, uint TxData, uint* pRxData = NULL);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
index	uint	yes	Parameter index number
TxData	uint	yes	Parameter value
pRxdata(Return Value)	uint	no	Result feedback

Legal Value:

Parameter	Legal Value	TxData	Note
index	0	0/1	P1 port change notification (0 = disabled, 1 = enabled)
	1	0/1	P 2 port change notification (0 = disabled, 1 = enabled)
	2	0/1	P 3 port change notification (0 = disabled, 1 = enabled)
	3	0/1	P 4 port change notification (0 = disabled, 1 = enabled)
	4	0/1	P 5 port change notification (0 = disabled, 1 = enabled)
	5	0/1	P 6 port change notification (0 = disabled, 1 = enabled)
	6	0/1	P 7 port change notification (0 = disabled, 1 = enabled)
	7	0/1	P 8 port change notification (0 = disabled, 1 = enabled)
	8	0/1	PTP movement in place notification (0 = disabled, 1 = enabled)
	9	0/1	Motor stall notification (0 = disabled, 1 = enabled)
	10	0/1	PVT data buffer water level is completely dry (0 = disabled, 1 = enabled)
	11	0/1	PVT data buffer water level is low (0 = disabled, 1 = enabled)

Return Value:

uirSDK3.0 Control System Development Kit

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3.
4.
5.
6. uint IEindex, IEdata, IEindexRx;//Define parameters
7. IEindex = 8; //PTP MOTION IN PLACE NOTIFICATION
8. IEdata = 1; //Turn on notification status (notification status enabled)
9.
10. //Set Information Enable
11. err = SdkSetInformEnable(g_GtwyHandle,NodeID,IEindex,IEdata,&IEindexRx);
12. if (err)
13. {
14.     std::cout << "Set Information Enable Failed!\n";
15.     return;
16. }
```

uirSDK3.0 Control System Development Kit

6.2 Get Information Enable

Function Name:

```
ERROR SdkGetInformEnable(uint GtwyHandle, uint CANid, uint index, uint* pRxData);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
index	uint	yes	Parameter index number
pRxdata(Return Value)	uint	yes	Result feedback

Legal Value:

Parameter	Legal Value	TxDATA	Note
index	0	0/1	P1 port change notification (0 = disabled, 1 = enabled)
	1	0/1	P 2 port change notification (0 = disabled, 1 = enabled)
	2	0/1	P 3 port change notification (0 = disabled, 1 = enabled)
	3	0/1	P 4 port change notification (0 = disabled, 1 = enabled)
	4	0/1	P 5 port change notification (0 = disabled, 1 = enabled)
	5	0/1	P 6 port change notification (0 = disabled, 1 = enabled)
	6	0/1	P 7 port change notification (0 = disabled, 1 = enabled)
	7	0/1	P 8 port change notification (0 = disabled, 1 = enabled)
	8	0/1	PTP movement in place notification (0 = disabled, 1 = enabled)
	9	0/1	Motor stall notification (0 = disabled, 1 = enabled)
	10	0/1	PVT data buffer water level is completely dry (0 = disabled, 1 = enabled)
	11	0/1	PVT data buffer water level is low (0 = disabled, 1 = enabled)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3.
4. uint IEindex,IEindexRx,IEStatus;//Define parameters
5. IEindex = 1; //Get P2 port notification status
6.
7. //Get Information Enable
8. err = SdkGetInformEnable(g_GtwyHandle, NodeID, IEindex, &IEindexRx);
9. if (err)
10.{
11.     std::cout << "Get Information Enable Failed!\n";
12.     return;
13.}
14.
15.IEStatus = IEindexRx;//(0 = disable - notification is turned off, 1 =
    enable - notification is on)
```

7.0 MOTION CONTROL STATIC PARAMETERS

This chapter will introduce how to Get or Set the motion control static parameters of the device through functions.

7.1 Set Motor Config

Function Name:

```
ERR0 SdkSetMotorConfig(uint GtwyHandle, uint CANid, uint index, uint data, uint* pRxData = NULL);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
index	uint	yes	Parameter index number
data	uint	yes	Parameter value
pRxdata(Return Value)	uint	no	Result feedback

Legal Value:

Parameter	Legal Value	data	Note
index	0 (MTS_MCS_IDX)	1/2/4/8/16	Subdivision [1.2.4.8.16]
	1 (MTS_CUR_IDX)	10...80	Current[10...80]
	2 (MTS_PSV_IDX)	0...100	Standby throttling [0...100(%)]
	3 (MTS_ENA_IDX)	0...60000	Power-on automatic enable delay [0...60000(ms)]
	5 (MTS_BRK_IDX)	0/1	brake control

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3.
4. uint MCindex, MCdata, MCdataRx;//Define parameters
5. MCindex = 1;//Set the rated operating current of the motor
6. MCdata = 60;//Set the current value to 6A (unit 0.1A, the maximum value
   is determined by the controller parameters)
7.
8. //Set Motor Config
9. err = SdkSetMotorConfig(g_GtwyHandle,NodeID,MCindex,MCdata,&MCdataRx);
10. if (err)
11. {
12.     std::cout << "Set Motor Config - rated current Failed!\n";
13.     return;
14. }
```

uirSDK3.0 Control System Development Kit

7.2 Get Motor Config

Function Name:

```
ERR0 SdkGetMotorConfig(uint GtwyHandle, uint CANid, uint index, uint* pRxData);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
index	uint	yes	Parameter index number
pRxdata(Return Value)	uint	yes	Result feedback

Legal Value:

Parameter	Legal Value	data	Note
index	0 (MTS_MCS_IDX)	1/2/4/8/16	Subdivision [1.2.4.8.16]
	1 (MTS_CUR_IDX)	10...80	Current[10...80]
	2 (MTS_PSV_IDX)	0...100	Standby throttling [0...100(%)]
	3 (MTS_ENA_IDX)	0...60000	Power-on automatic enable delay [0...60000(ms)]
	5 (MTS_BRK_IDX)	0/1	brake control

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3.
4. uint MCindex, MCdataRx, MotorBreakStatus; //Define parameters
5. MCindex = 5; //Get brake status
6.
7. //Get Motor Config
8. err = SdkGetMotorConfig(g_GtwyHandle, NodeID, MCindex, &MCdataRx);
9. if (err)
10. {
11.     std::cout << "Get Motor Config - brake status Failed!\n";
12.     return;
13. }
14.
15. MotorBreakStatus = MCdataRx; //Motor braking status (0-free, 1-brake)
```

uirSDK3.0 Control System Development Kit

7.3 Set Acceleration

Function Name:

```
ERRO SdkSetAcceleration(uint GtwyHandle, uint CANid, uint data, uint* pRxData = NULL);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
data	uint	yes	Acceleration value
pRxdata(Return Value)	uint	no	Result feedback

Legal Value:

Parameter	Legal Value	Note
data	1...65,000,000	Define acceleration and deceleration numerically, unit (pulse/sec ²)
	1...60,000	Define acceleration and deceleration in time, unit (ms)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. uint NodeID = 5; //Device's ID
2. uint ICdataRx;
3. err = SdkSetInitialConfig(g_GtwyHandle, NodeID, 4, 1, &ICdataRx);
4. if (err)
5. {
6.     std::cout << "Set Initial Config Failed!\n";
7.     return;
8. }
9. uint MotorAC, MotorACRx; //Define parameters
10. MotorAC = 1000; //Acceleration value 1000 ms
11. //Set Acceleration
12. err = SdkSetAcceleration(g_GtwyHandle, NodeID, MotorAC, &MotorACRx);
13. if (err)
14. {
15.     std::cout << "Set Acceleration Failed!\n";
16.     return;
17. }
```


uirSDK3.0 Control System Development Kit

7.4 Get Acceleration

Function Name: `ERRO SdkGetAcceleration(uint GtwyHandle, uint CANid, uint* pRxData);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pRxdata(Return Value)	uint	yes	Result feedback

Legal Value:

Parameter	Legal Value	Note
data	1...65,000,000	Define acceleration and deceleration numerically, unit (pulse/sec ²)
	1...60,000	Define acceleration and deceleration in time, unit (ms)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3.
4. uint MotorAC, MotorACRx;
5. //Get Acceleration
6. err = SdkGetAcceleration(g_GtwyHandle, NodeID, &MotorACRx);
7. if (err)
8. {
9.     std::cout << "Get Acceleration Failed!\n";
10.    return;
11. }
12.
13. MotorAC = MotorACRx; //Acceleration value
```

uirSDK3.0 Control System Development Kit

7.5 Set Deceleration

Function Name: `ERRO SdkSetDeceleration(uint GtwyHandle, uint CANid, uint data, uint* pRxData = NULL);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
data	uint	yes	Deceleration value
pRxdata(Return Value)	uint	no	Result feedback

Legal Value:

Parameter	Legal Value	Note
data	1...65,000,000	Define acceleration and deceleration numerically, unit (pulse/sec ²)
	1...60,000	Define acceleration and deceleration in time, unit (ms)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. //Acceleration and deceleration unit settings, refer to 5.1 Initial
   Config Set, here takes the time mode as an example
4.
5. uint MotorDC, MotorDCRx; //Define parameters
6. MotorDC = 1000; //Deceleration value 1000ms
7. //Set deceleration
8. err = SdkSetDeceleration(g_GtwyHandle, NodeID, MotorDC, &MotorDCRx);
9. if (err)
10. {
11.     std::cout << "Set Deceleration Failed!\n";
12.     return;
13. }
```

uirSDK3.0 Control System Development Kit

7.6 Get Deceleration

Function Name: `ERRO SdkGetDeceleration(uint GtwyHandle, uint CANid, uint* piDeceleration);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
piDeceleration(Return Value)	uint	yes	Result feedback

Legal Value:

Parameter	Legal Value	Note
data	1...65,000,000	Define acceleration and deceleration numerically, unit (pulse/sec ²)
	1...60,000	Define acceleration and deceleration in time, unit (ms)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. uint MotorDC, MotorDCRx; //Define parameters
4. //Get motor deceleration
5. ERRO err = SdkGetDeceleration(g_GtwyHandle, NodeID, &MotorDCRx);
6. if (err)
7. {
8.     std::cout << "Get Deceleration Failed!\n";
9.     return;
10. }
11. MotorDC = MotorDCRx; //Deceleration value
```

uirSDK3.0 Control System Development Kit

7.7 Set Cut-In Speed

Function Name:

```
ERRO SdkSetCutInSpeed(uint GtwyHandle, uint CANid, uint data, uint* pRxData = NULL);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
data	uint	yes	Cut-in speed value
pRxdata(Return Value)	uint	no	Result feedback

Legal Value:

Parameter	Legal Value	Note
data	0...2 ³²	Cut-in speed value, unit (pulse/sec)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3.
4. uint CutInSpd, CutInSpdRx; //Define parameters
5. CutInSpd = 5000; // The cut-in speed is set to 5000pulse/sec
6.
7. //Set Cut-In Speed
8. err = SdkSetCutInSpeed(g_GtwyHandle, NodeID, CutInSpd, &CutInSpdRx);
9. if (err)
10. {
11.     std::cout << "Set Cut-In Speed Failed!\n";
12.     return;
13. }
```

uirSDK3.0 Control System Development Kit

7.8 Get Cut-In speed

Function Name: `ERRO SdkGetCutInSpeed(uint GtwyHandle, uint CANid, uint* pRxData);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pRxdata(Return Value)	uint	yes	Result feedback

Legal Value:

Parameter	Legal Value	Note
data	0...2 ³²	Cut-in speed value, unit (pulse/sec)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3.
4. uint CutInSpd, CutInSpdRx; //Define parameters
5.
6. //Set Cut-In Speed
7. err = SdkGetCutInSpeed(g_GtwyHandle, NodeID, &CutInSpdRx);
8. if (err)
9. {
10.     std::cout << "Get Cut-In speed Failed!\n";
11.     return;
12. }
13.
14. CutInSpd = CutInSpdRx; //Cut-in speed value
```

uirSDK3.0 Control System Development Kit

7.9 Set Emergency Stop Deceleration

Function Name:

ERRO `SdkSetStopDeceleration`(uint GtwyHandle, uint CANid, uint data, uint* pRxData = NULL);

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
data	uint	yes	Emergency stop deceleration value
pRxdata(Return Value)	uint	no	Result feedback

Legal Value:

Parameter	Legal Value	Note
data	0...2 ³²	Emergency stop deceleration value , unit (pulse/sec)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3.
4. uint MotorSD, MotorSDRx; //Define parameters
5. MotorSD = 1000000; //Set emergency stop deceleration 1000000(pulse/sec)
6.
7. //Set emergency stop deceleration
8. err = SdkSetStopDeceleration(g_GtwyHandle, NodeID, MotorSD, &MotorSDRx);
9. if (err)
10. {
11.     std::cout << "Set emergency stop deceleration Failed!\n";
12.     return;
13. }
```

uirSDK3.0 Control System Development Kit

7.10 Get Emergency Stop Deceleration

Function Name: `ERRO SdkGetStopDeceleration(uint GtwyHandle, uint CANid, uint* pRxData);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pRxdata(Return Value)	uint	yes	Result feedback

Legal Value:

Parameter	Legal Value	Note
data	0...2 ³²	Emergency stop deceleration value , unit (pulse/sec)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. uint NodeID = 5; //Device's ID
2.
3. uint MotorSD, MotorSDRx; //Define parameters
4. //Get Emergency Stop Deceleration
5. ERRO err = SdkGetStopDeceleration(g_GtwyHandle, NodeID, &MotorSDRx);
6. if (err)
7. {
8.     std::cout << "Get Emergency Stop Deceleration Failed!\n";
9.     return;
10. }
11.
12. MotorSD = MotorSDRx; //Emergency stop deceleration value
```

uirSDK3.0 Control System Development Kit

7.11 Set Encoder Parameters

Function Name:

```
ERRO SdkSetEncoderConfig(uint GtwyHandle, uint CANid, uint index, uint data, uint* pRxData  
= NULL);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
index	uint	yes	Parameter index number
data	uint	yes	Parameter value
pRxdata(Return Value)	uint	no	Result feedback

Legal Value:

Parameter	Legal Value	data	Note
index	0 (QES_QER_IDX)	10...16000	Number of encoder lines (after x4 = number of pulses per revolution)
	1 (QES_PQE_IDX)	10 ... 65535	stall tolerance
	2 (QES_AER_IDX)	17/20	Number of single revolutions for the encoder
	4 (QES_QEX_IDX)	1 ... 65535	INC encoder line number (actual value, such as magnetic encoder 90 lines)
	5 (QES_SYN_IDX)		Clear residual position error, usually set after stop

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. uint ECdataRx;
4. //Set encoder parameters -> Encoder Line number = 800
5. err = SdkSetEncoderConfig(g_GtwyHandle,NodeID,QES_QER_IDX,800,&ECdataRx);
6. if (err)
7. {
8.     std::cout << "Set encoder parameters Failed!\n"; return;
9. }
```


uirSDK3.0 Control System Development Kit

7.12 Get Encoder Parameters

Function Name:

```
ERRO SdkGetEncoderConfig(uint GtwyHandle, uint CANid, uint index, uint* pRxData);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
index	uint	yes	Parameter index number
pRxdata(Return Value)	uint	yes	Result feedback

Legal Value:

Parameter	Legal Value	data	Note
index	0 (QES_QER_IDX)	10...16000	Number of encoder lines (after x4 = number of pulses per revolution)
	1 (QES_PQE_IDX)	10 ... 65535	stall tolerance
	2 (QES_AER_IDX)	17/20	Number of single revolutions for the encoder
	3 (QES_ABW_IDX)	0/1	Absolute encoder battery status (0: low battery, 1: OK)
	4 (QES_QEX_IDX)	1 ... 65535	INC encoder line number (actual value, such as magnetic encoder 90 lines)
	5 (QES_SYN_IDX)		Clear residual position error, usually set after stop

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. uint BatteryStatus, ECdataRx; //Define parameters
4.
5. //Get Encoder Parameters -> battery power
6. err = SdkGetEncoderConfig(g_GtwyHandle, NodeID, QES_ABW_IDX, &ECdataRx);
7. if (err)
8. {
9.     std::cout << "Get Encoder Parameters Failed!\n"; return;
10. }
11. BatteryStatus = ECdataRx; // Battery status (0: Low battery, 1: OK)
```


uirSDK3.0 Control System Development Kit

7.13 Get Auxiliary Location Information

Function Name: `ERRO SdkGetAuxPosition(uint GtwyHandle, uint CANid, int* pRxData);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pRxdata(Return Value)	int	yes	Result feedback (Refer to uirSDKcommon.h)

Legal Value:

Return Value	Legal Value	Note
pRxdata		Closed-loop control mode, returns open-loop pulse value; Open loop control mode, returns encoder value

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3.
4. int AuxPos, AuxPosRx; //Define parameters
5.
6. //Get auxiliary location information
7. err = SdkGetAuxPosition(g_GtwyHandle, NodeID, &AuxPosRx);
8. if (err)
9. {
10.     std::cout << "Get auxiliary location information Failed!\n";
11.     return;
12. }
13.
14. AuxPos = AuxPosRx; //Position information value
```

uirSDK3.0 Control System Development Kit

7.14 Set Backlash Compensation Value

Function Name:

```
ERRO SdkSetBacklashComp(uint GtwyHandle, uint CANid, uint data, uint* pRxData = NULL);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
data	uint	yes	Parameter value
pRxdata(Return Value)	uint	no	Result feedback

Legal Value:

Parameter	Legal Value	Note
data	0...2 ³²	Backlash compensation value, unit (pulse)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3.
4. uint BLCdata, BLCdataRx; //Define parameters
5. BLCdata = 100; //Set the compensation value to 100 (pulse)
6.
7. //Set backlash compensation value
8. err = SdkSetBacklashComp(g_GtwyHandle, NodeID, BLCdata, &BLCdataRx);
9. if (err)
10. {
11.     std::cout << "Set backlash compensation value Failed!\n";
12.     return;
13. }
```

uirSDK3.0 Control System Development Kit

7.15 Get Backlash Compensation Value

Function Name: `ERRO SdkGetBacklashComp(uint GtwyHandle, uint CANid, uint* pRxData);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pRxdata(Return Value)	uint	yes	Result feedback

Legal Value:

Parameter	Legal Value	Note
data	0...2 ³²	Backlash compensation value, unit (pulse)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. uint BLCdata, BLCdataRx; //Define parameters
4. //Get Backlash Compensation Value
5. err = SdkGetBacklashComp(g_GtwyHandle, NodeID, &BLCdataRx);
6. if (err)
7. {
8.     std::cout << "Set backlash compensation value Failed!\n";
9.     return;
10. }
11.
12. BLCdata = BLCdataRx; //Backlash Compensation Value
```

uirSDK3.0 Control System Development Kit

7.16 Set Motion Limits

Function Name:

```
ERRRO SdkSetMotionLimits(uint GtwyHandle, uint CANid, MOTION_LIMIT_OBJ* pData,  
MOTION_LIMIT_OBJ* pRxData = NULL);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pData	MOTION_LIMIT_OBJ	yes	Parameter value (Refer to uirSDKcommon.h)
pRxdata(Return Value)	MOTION_LIMIT_OBJ	no	Result feedback (Refer to uirSDKcommon.h)

Legal Value:

Parameter	Attributes	Legal Value	Note
pData	MaxVelocity		Maximum working speed pps
	LowWorkRange		Minimum working limit pls (If exceeded, emergency stop and deceleration will occur)
	UppWorkRange		Maximum working limit pls (if exceeded, emergency stop and deceleration will occur)
	LowBumpLimit		Minimum collision limit pls (If exceeded, it will stop instantly)
	UppBumpLimit		Maximum collision limit pls (will stop instantly if exceeded)
	MaxVeloError		Maximum speed error pps (exceeding it means backdrugging has occurred)
	MaxPosiError		Maximum displacement error pls (exceeding it indicates stalling)
	MaxAccValue		Maximum acceleration (numeric mode pps/ms, 107 = acceleration per ms 107pps, in PVT mode, limit the calculated acceleration to be too large)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3.
4.
5. MOTION_LIMIT_OBJ MotionLi, MotionLiRx; //Define parameters
6. MotionLi.MaxVelocity = 600000; //Maximum working speed pps
7. MotionLi.LowWorkRange = -2000000; //Lowest working limit pls (exceeding
   it will result in emergency stop and deceleration)
8. MotionLi.UppWorkRange = 2000000; //Maximum working limit pls (exceeding
   it will cause emergency stop and deceleration)
9. MotionLi.LowBumpLimit = -2500000; //Lowest collision limit pls (will stop
   instantly if exceeded)
10. MotionLi.UppBumpLimit = 2500000; //Highest collision limit pls (will stop
    instantly if exceeded)
11. MotionLi.MaxVeloError = 100; //Maximum speed error pps (exceeding means
    backdragging has occurred )
12. MotionLi.MaxPosiError = 100; //Maximum displacement error pls (exceeding
    indicates stalled rotor)
13. MotionLi.MaxAccValue = 50000; //Maximum acceleration (numeric mode
    pps/ms)
14. //Set motion limits
15. err = SdkSetMotionLimits(g_GtwyHandle, NodeID, &MotionLi, &MotionLiRx);
16. if (err)
17. {
18.     std::cout << "Set motion limits Failed!\n";
19.     return;
20. }
```

uirSDK3.0 Control System Development Kit

7.17 Get Motion Limits

Function Name:

```
ERR0 SdkGetMotionLimits(uint GtwyHandle, uint CANid, MOTION_LIMIT_OBJ* pRxData);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pRxdata(Return Value)	MOTION_LIMIT_OBJ	yes	Result feedback (Refer to uirSDKcommon.h)

Legal Value:

Parameter	Attributes	Legal Value	Note
pData	MaxVelocity		Maximum working speed pps
	LowWorkRange		Minimum working limit pls (If exceeded, emergency stop and deceleration will occur)
	UppWorkRange		Maximum working limit pls (if exceeded, emergency stop and deceleration will occur)
	LowBumpLimit		Minimum collision limit pls (If exceeded, it will stop instantly)
	UppBumpLimit		Maximum collision limit pls (will stop instantly if exceeded)
	MaxVeloError		Maximum speed error pps (exceeding it means backdragging has occurred)
	MaxPosiError		Maximum displacement error pls (exceeding it indicates stalling)
	MaxAccValue		Maximum acceleration (numeric mode pps/ms, 107 = acceleration per ms 107pps, in PVT mode, limit the calculated acceleration to be too large)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3.
4.
5. MOTION_LIMIT_OBJ MotionLi, MotionLiRx; //Define parameters
6. //Get Motion Limits
7. err = SdkGetMotionLimits(g_GtwyHandle, NodeID, &MotionLiRx);
8. if (err)
9. {
10.     std::cout << "Set motion limits Failed!\n";
11.     return;
12. }
13.
14. MotionLi.MaxVelocity = MotionLiRx.MaxVelocity; //Maximum working speed
    pls
15. MotionLi.LowWorkRange = MotionLiRx.LowWorkRange; //Lowest working
    limit pls
16. MotionLi.UppWorkRange = MotionLiRx.UppWorkRange; //Highest working limit
    pls
17. MotionLi.LowBumpLimit = MotionLiRx.LowBumpLimit; //Minimum collision
    limit pls
18. MotionLi.UppBumpLimit = MotionLiRx.UppBumpLimit; //Highest collision
    limit pls
19. MotionLi.MaxVeloError = MotionLiRx.MaxVeloError; //Maximum speed error
    pls
20. MotionLi.MaxPosiError = MotionLiRx.MaxPosiError; //Maximum displacement
    error pls
21. MotionLi.MaxAccValue = MotionLiRx.MaxAccValue; //Maximum acceleration
```

uirSDK3.0 Control System Development Kit

7.18 Reset Motion Limits

Function Name: `ERRO SdkRstMotionLimits(uint GtwyHandle, uint CANid);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3.
4. //Reset Motion Limits
5. err = SdkRstMotionLimits(g_GtwyHandle, NodeID);
6. if (err)
7. {
8.     std::cout << "Reset Motion limits Failed!\n";
9.     return;
10. }
```

uirSDK3.0 Control System Development Kit

7.19 Set Limit Check Status

Function Name:

```
ERRO SdkSetLimitCheck(uint GtwyHandle, uint CANid, uint data, uint* pRxData = NULL);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
data	uint	yes	Limit check status
pRxData(Return Value)	uint	no	Limit check status feedback

Legal Value:

Parameter	Legal Value	Note
data	0/1	Enable/Disable limit check, (0: disable/1: enable)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. uint LimitStatus, LimitStatusRx; //Define parameters
4. LimitStatus = 1; //Limit check Enable
5. //Set Limit Check Status
6. err = SdkSetLimitCheck(g_GtwyHandle, NodeID, LimitStatus, &LimitStatusRx);
7. if (err)
8. {
9.     std::cout << "Set limit check status Failed!\n";
10.    return;
11. }
```

uirSDK3.0 Control System Development Kit

7.20 Get Limit Check Status

Function Name: `ERRO SdkGetLimitCheck(uint GtwyHandle, uint CANid, uint* pRxData);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pRxData(Return Value)	uint	yes	Limit check status feedback

Legal Value:

Parameter	Legal Value	Note
data	0/1	Enable/Disable limit check, (0: disable/1: enable)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. uint LimitStatus, LimitStatusRx; //Define parameters
4.
5. //Set Limit Check Status
6. err = SdkGetLimitCheck(g_GtwyHandle, NodeID, &LimitStatusRx);
7. if (err)
8. {
9.     std::cout << "Set limit check status Failed!\n";
10.    return;
11. }
12. LimitStatus = LimitStatusRx; //Limit check status (0: Disabled / 1: Enabled)
```

uirSDK3.0 Control System Development Kit

7.21 Get Desired Value

Function Name:

```
ERRO SdkGetDesiredValue(uint GtwyHandle, uint CANid, uint index, void* pRxData);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
index	uint	yes	Parameter index number
pRxData(Return Value)	void	yes	Desired Value feedback

Legal Value:

Parameter index number	Legal Value	Note
index	0 (DVR_MOD_IDX)	Target control mode: JV / PTP / PVT
	1 (DVR_CUR_IDX)	Target operating current
	2 (DVR_SPD_IDX)	target speed
	3 (DVR_PRM_IDX)	target displacement
	4 (DVR_PAM_IDX)	target location
	5 (DVR_TIS_IDX)	target alarm temperature

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. int MotorSPD,pRxData;
4. //Get Desired Value -> target speed
5. err = SdkGetDesiredValue(g_GtwyHandle, NodeID, DVR_SPD_IDX,&pRxData);
6. if (err)
7. {
8.     std::cout << "Get Desired Value Failed!\n";
9.     return;
10.}
11. MotorSPD = pRxData;//Set the speed value
```

8.0 MOTION CONTROL DYNAMIC PARAMETERS

This chapter will introduce how to Get or Set the dynamic parameters of motion control through functions.

8.1 Set Motor Enable Status

Function Name:

```
ERRO SdkSetMotorOn(uint GtwyHandle, uint CANid, uint bMotorOn, uint *pbEnableOut = NULL);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
bMotorOn	uint	yes	Motor enable status
pbEnableOut(Return Value)	uint	no	Result feedback

Legal Value:

Parameter	Legal Value	Note
bMotorOn	0/1	Motor enable status (0: offline/1: enabled)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. uint MotorONRx; //Define parameters
4.
5. //SetMotor enable status -> enabled
6. err = SdkSetMotorOn(g_GtwyHandle, NodeID, 1, &MotorONRx);
7. if (err)
8. {
9.     std::cout << "SetMotor enable status Failed!\n";
10.    return;
11. }
```


uirSDK3.0 Control System Development Kit

8.2 Get Motor Enable Status

Function Name: `ERRRO SdkGetMotorOn(uint GtwyHandle, uint CANid, uint *pbEnable);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pbEnable(Return Value)	uint	yes	Result feedback

Legal Value:

Parameter	Legal Value	Note
bMotorOn	0/1	Motor enable status (0: offline/1: enabled)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. uint MotorON, MotorONRx; //Define parameters
4.
5. //GetMotor enable status
6. err = SdkGetMotorOn(g_GtwyHandle, NodeID, &MotorONRx);
7. if (err)
8. {
9.     std::cout << "GetMotor enable status Failed!\n";
10.    return;
11. }
12. MotorON = MotorONRx; //Motor enable status (0: offline/1: enabled)
```


uirSDK3.0 Control System Development Kit

8.3 Set Motor Being Motion

Function Name: `ERRO SdkSetBeginMxn(uint GtwyHandle, uint CANId, uint* pRxData = NULL);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANId	uint	yes	Device's ID
pRxData(Return Value)	uint	no	Motion moment feedback

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. int pRxVelo;
4. //SetJogJog speed -> 10000 (pulse/sec)
5. err = SdkSetJogMxn(g_GtwyHandle, NodeID,10000,&pRxVelo);
6. if (err)
7. {
8.     std::cout << "Set Jog Speed Failed!\n";
9.     return;
10.}
11.
12.uint pRxData;
13.//motor start motion
14.err = SdkSetBeginMxn(g_GtwyHandle, NodeID);
15.if (err)
16.{
17.    std::cout << "Start Motion Failed!\n";
18.    return;
19.}
```

uirSDK3.0 Control System Development Kit

8.4 Set Motor Being Motion With Time

Function Name: `ERRO SdkSetBeginMxnTime(uint GtwyHandle, uint CANid, uint time ,uint* pRxData = NULL);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
time	uint	no	Motion moment (unit:us)
pRxData(Return Value)	no	no	Motion moment feedback

Legal Value:

Parameter	Legal Value	Note
time	1...2 ³²	Motion moment (unit:us)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. int pRxVelo;
4. //SetJogJog speed -> 10000 (pulse/sec)
5. err = SdkSetJogMxn(g_GtwyHandle, NodeID,10000,&pRxVelo);
6. if (err)
7. {
8.     std::cout << "Set Jog Speed Failed!\n";
9.     return;
10. }
11. //motor start motion in 1000us
12. err = SdkSetBeginMxnTime(g_GtwyHandle, NodeID,1000);
13. if (err)
14. {
15.     std::cout << "Start Motion Failed!\n";
16.     return;
17. }
```

uirSDK3.0 Control System Development Kit

8.5 Set Motor Emergency Stop

Function Name: `ERR0 SdkSetStopMxn(uint GtwyHandle, uint CANid, uint* pRxData = NULL);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle(Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pRxData(Return Value)	uint	no	Emergency stop state feedback

Legal Value:

Return Value	Legal Value	Note
pRxData	0/1	emergency stop state

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. int pRxVelo;
4. //SetJogJog speed -> 10000 (pulse/sec)
5. err = SdkSetJogMxn(g_GtwyHandle, NodeID,10000,&pRxVelo);
6. if (err)
7. {
8.     std::cout << "Set Jog Speed Failed!\n";
9.     return;
10.}
11.
12.uint pRxData;
13.// motor start motion
14.err = SdkSetBeginMxn(g_GtwyHandle, NodeID,&pRxData);//开始运动
15.if (err)
16.{
17.    std::cout << "start motion Failed!\n";
18.    return;
19.}
20.
21.PauseMS(2000);//delay time 2000ms
22.
23.//Motor Emergency Stop
24.err = SdkSetStopMxn(g_GtwyHandle, NodeID, &pRxData);
25.if (err)
26.{
27.    std::cout << "Emergency Stop Failed!\n";
28.    return;
29.}
```

uirSDK3.0 Control System Development Kit

8.6 Set Origin Position

Function Name: `ERRO SdkSetOrigin(uint GtwyHandle, uint CANid);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. //Set Origin Position
4. err = SdkSetOrigin(g_GtwyHandle, NodeID);
5. if (err)
6. {
7.     std::cout << "8.5 Set Origin Position Failed!\n";
8.     return;
9. }
```

uirSDK3.0 Control System Development Kit

8.7 Set Jog Motion Speed

Function Name: `ERRO SdkSetJogMxn(uint GtwyHandle, uint CANid, int Velo, int* pRxVelo = NULL);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
Velo	int	yes	Speed target value
pRxVelo(Return Value)	int	no	Speed target value feedback

Legal Value:

Parameter	Legal Value	Note
Velo	$-2^{31} \dots +2^{31}$	Speed value, unit (pulse/sec)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3.
4. int pRxVelo;
5. //SetJogJog speed -> 10000 (pulse/sec)
6. err = SdkSetJogMxn(g_GtwyHandle, NodeID, 10000, &pRxVelo);
7. if (err)
8. {
9.     std::cout << "Set Jog Speed Failed!\n";
10.    return;
11. }
```

uirSDK3.0 Control System Development Kit

8.8 Get Jog Motion Speed

Function Name: `ERRO SdkGetJogMxn(uint GtwyHandle, uint CANid, int* pRxVelo);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pRxVelo(Return Value)	int	yes	Current speed feedback

Legal Value:

Return Value	Legal Value	Note
pRxVelo	$-2^{31} \dots +2^{31}$	Speed value, unit (pulse/sec)

Return Value:

Return Value	Note
0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. int Velo, VeloRx; //Define parameters
4. //Get Jog Motion Speed
5. err = SdkGetJogMxn(g_GtwyHandle, NodeID, &VeloRx);
6. if (err)
7. {
8.     std::cout << "Get Jog Motion Speed Failed!\n";
9.     return;
10.}
11.Velo = VeloRx; //Current speed
```

uirSDK3.0 Control System Development Kit

8.9 Set PTP Motion Absolute Position

Function Name:

```
ERRO SdkSetPtpMxnA(uint GtwyHandle, uint CANid, int Velo, int Pa, int* pRxVelo, int* pRxPa  
= NULL);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
Velo	int	yes	Speed value
Pa	int	yes	Encoder position (absolute position pulses)
pRxVelo(Return Value)	int	no	Current Speed feedback
pRxPa(Return Value)	int	no	Current Position feedback

Legal Value:

Parameter	Legal Value	Note
Velo	$-2^{31} \dots +2^{31}$	Speed value, unit (pulse/sec)
Pa	$-2^{31} \dots +2^{31}$	Encoder position value, unit (pulse)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. int Velo, pRxVelo; // Define speed
4. int Pa, pRxPa; // Define position
5.
6. Velo = 20000; //Speed 20000(pulse/sec)
7. Pa = 8000; //Absolute position 8000(pulse)
8. //Set PTP Motion Absolute Position
9. err = SdkSetPtpMxnA(g_GtwyHandle, NodeID, Velo, Pa,&pRxVelo,&pRxPa);
10. if (err)
11. {
12.     std::cout << "Set PTP Motion Absolute Position Failed!\n";
13.     return;
14. }
```


uirSDK3.0 Control System Development Kit

8.10 Get PTP Motion Absolute Position

Function Name: `ERRO SdkGetPtpMxnA(uint GtwyHandle, uint CANid, int* pRxVelo, int* pRxPa);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pRxVelo(Return Value)	int	yes	Current Speed feedback
pRxPa(Return Value)	int	yes	Current Position feedback

Legal Value:

Parameter	Legal Value	Note
Velo	$-2^{31} \dots +2^{31}$	Speed value, unit (pulse/sec)
Pa	$-2^{31} \dots +2^{31}$	Encoder position value, unit (pulse)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3.
4.
5. int Velo, pRxVelo; //Define speed
6. int Pa, pRxPa; //Define position
7.
8. //Get PTP Motion Absolute Position
9. err = SdkGetPtpMxnA(g_GtwyHandle, NodeID, &pRxVelo, &pRxPa);
10. if (err)
11. {
12.     std::cout << "Get PTP Motion Absolute Position Failed!\n";
13.     return;
14. }
15.
16. Velo = pRxVelo; //Current Speed
17. Pa = pRxPa; //Current Position
```

uirSDK3.0 Control System Development Kit

8.11 Set PTP Motion Relative Position

Function Name:

```
ERRO SdkSetPtpMxnR(uint GtwyHandle, uint CANid, int Velo, int Pr, int* pRxVelo = NULL, int* pRxPr = NULL);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
Velo	int	yes	speed value
Pr	int	yes	Displacement pulses
pRxVelo(Return Value)	int	no	Current Speed feedback
pRxPr(Return Value)	int	no	Current Position feedback

Legal Value:

Parameter	Legal Value	Note
Velo	$-2^{31} \dots +2^{31}$	Speed value, unit (pulse/sec)
Pr	$-2^{31} \dots +2^{31}$	Number of displacement pulses , unit (pulse)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. uint NodeID = 5; //Device's ID
2. int Velo, pRxVelo; //Define speed
3. int Pr, pRxPr; //Define position
4.
5. Velo = 10000; //Speed 10000(pulse/sec)
6. Pr = 5000; //Displacement 5000(pulse)
7.
8. //Set PTP Motion Relative Position
9. ERRO err = SdkSetPtpMxnR(g_GtwyHandle, NodeID, Velo, Pr, &pRxVelo, &pRxPr);
10. if (err)
11. {
12.     std::cout << "Set PTP Motion Relative Position Failed!\n";
13.     return;
14. }
```


uirSDK3.0 Control System Development Kit

8.12 Get PTP Motion Relative Position

Function Name: `ERRO SdkGetPtpMxnR(uint GtwyHandle, uint CANid, int* pRxVelo, int* pRxPr);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pRxVelo(Return Value)	int	yes	Current Speed feedback
pRxPr(Return Value)	int	yes	Current Position feedback

Legal Value:

Parameter	Legal Value	Note
Velo	$-2^{31} \dots +2^{31}$	Speed value, unit (pulse/sec)
Pr	$-2^{31} \dots +2^{31}$	Number of displacement pulses , unit (pulse)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3.
4. int Velo, pRxVelo; //Define speed
5. int Pr, pRxPr; //Define position
6.
7. //Get PTP Motion Relative Position
8. err = SdkGetPtpMxnR(g_GtwyHandle, NodeID, &pRxVelo, &pRxPr);
9. if (err)
10. {
11.     std::cout << "Get PTP Motion Relative Position Failed!\n";
12.     return;
13. }
14.
15. Velo = pRxVelo; //Current Speed
16. Pr = pRxPr; //Current Position
```

9.0 I/O OPERATIONS

This chapter will introduce how to Get or Set IO input and output ports through functions.

9.1 Set Port Trigger Mode

Function Name:

```
ERRRO SdkSetTriggerMode(uint GtwyHandle, uint CANId, uint PortIx, uint TrigTime, uint* pRxData = NULL);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANId	uint	yes	Device's ID
PortIx	uint	yes	IO port number
TrigTime	uint	yes	Trig Mode
pRxData(Return Value)	uint	no	Result feedback

Legal Value:

Parameter	Legal Value	Note
PortIx	0...15	IO port number (1-16)
	0	Continuous trigger
TrigTime	1...60000	Filter time 1...60000 (ms)
	60001	Single trigger, needs to be reset after triggering to work again.

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. uint PortIx, TrigTime,pRxData;
4. PortIx = 0; //P1 port
5. TrigTime = 0; //Continuous triggering
6.
7. //Set Port Trigger Mode -> P1 port, continuous trigger
8. err = SdkSetTriggerMode(g_GtwyHandle, NodeID, PortIx, TrigTime,&pRxData);
9. if (err)
10.{
11.     std::cout << "Set Port Trigger Mode Failed!\n";
12.     return;
13.}
```

uirSDK3.0 Control System Development Kit

9.2 Get Port Trigger Mode

Function Name:

```
ERRO SdkGetTriggerMode(uint GtwyHandle, uint CANid, uint PortIx, uint* pRxData);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
PortIx	uint	yes	IO port number
pRxData(Return Value)	uint	yes	Result feedback

Legal Value:

Parameter	Legal Value	Note
PortIx	0...15	IO port number (1-16)
	0	Continuous trigger
pRxData	1...60000	Filter time 1...60000 (ms)
	60001	Single trigger, needs to be reset after triggering to work again.

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. uint NodeID = 5; //Device's ID
2. uint PortIx, TrigTime, pRxData;
3. PortIx = 1; //P2 port
4.
5. //Get Port Trigger Mode -> P2 port
6. ERRO err = SdkGetTriggerMode(g_GtwyHandle, NodeID, PortIx, &pRxData);
7. if (err)
8. {
9.     std::cout << "Get Port Trigger Mode Failed!\n";
10.    return;
11. }
12. TrigTime = pRxData; // delay time
```


uirSDK3.0 Control System Development Kit

9.3 Set Port Digital Output

Function Name:

```
ERR0 SdkSetDigitalOutput(uint GtwyHandle, uint CANid, DIGITAL_OUTPUT_OBJ* pDOValue,  
UIM_DIO_OBJ* pRxData = NULL);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pDOValue	DIGITAL_OUTPUT_OBJ	yes	IO port output value (Refer to uirSDKcommon.h)
pRxData(Return Value)	UIM_DIO_OBJ	no	IO port input and output (Refer to uirSDKcommon.h)

Legal Value:

Parameter	Attributes	Legal Value	Note
pDOValue	bits.bQP00	0/1	P1 OutPut level
	bits.bQP01	0/1	P2 OutPut level
	bits.bQP02	0/1	P3 OutPut level
	bits.bQP03	0/1	P4 OutPut level
	bits.bQP04	0/1	P5 OutPut level
	bits.bQP05	0/1	P6 OutPut level
	bits.bQP06	0/1	P7 OutPut level
	bits.bQP07	0/1	P8 OutPut level
	bits.bMSK00	0/1	P1 OutPut Mask
	bits.bMSK01	0/1	P2 OutPut Mask
	bits.bMSK02	0/1	P3 OutPut Mask
	bits.bMSK03	0/1	P4 OutPut Mask
	bits.bMSK04	0/1	P5 OutPut Mask
	bits.bMSK05	0/1	P6 OutPut Mask
	bits.bMSK06	0/1	P7 OutPut Mask
	bits.bMSK07	0/1	P9 OutPut Mask
	bits.bQP08	0/1	P9 OutPut level

uirSDK3.0 Control System Development Kit

pRxData	bits.bQP09	0/1	P10 OutPut level
	bits.bQP10	0/1	P12 OutPut level
	bits.bQP11	0/1	P13 OutPut level
	bits.bQP12	0/1	P14 OutPut level
	bits.bQP13	0/1	P15 OutPut level
	bits.bQP14	0/1	P15 OutPut level
	bits.bQP15	0/1	P16 OutPut level
	bits.bMSK08	0/1	P9 OutPut Mask
	bits.bMSK09	0/1	P10 OutPut Mask
	bits.bMSK10	0/1	P11 OutPut Mask
	bits.bMSK11	0/1	P12 OutPut Mask
	bits.bMSK12	0/1	P13 OutPut Mask
	bits.bMSK13	0/1	P14 OutPut Mask
	bits.bMSK14	0/1	P15 OutPut Mask
	bits.bMSK15	0/1	P16 OutPut Mask
	byte.LocalOutPut		
	byte.LocalOutPutMask		
	byte.RemoteOutPut		
	byte.RemoteOutPutMask		
	bits.bIN00	0/1	P1 InPut level
	bits.bIN01	0/1	P2 InPut level
	bits.bIN02	0/1	P3 InPut level
	bits.bIN03	0/1	P4 InPut level
	bits.bIN04	0/1	P5 InPut level
	bits.bIN05	0/1	P6 InPut level
	bits.bIN06	0/1	P7 InPut level
	bits.bIN07	0/1	P8 InPut level
	bits.bQP00	0/1	P1 OutPut level
	bits.bQP01	0/1	P2 OutPut level

uirSDK3.0 Control System Development Kit

bits.bQP02	0/1	P3 OutPut level
bits.bQP03	0/1	P4 OutPut level
bits.bQP04	0/1	P5 OutPut level
bits.bQP05	0/1	P6 OutPut level
bits.bQP06	0/1	P7 OutPut level
bits.bQP07	0/1	P8 OutPut level
bits.bIN08	0/1	P9 InPut level
bits.bIN09	0/1	P10 InPut level
bits.bIN10	0/1	P11 InPut level
bits.bIN11	0/1	P12 InPut level
bits.bIN12	0/1	P13 InPut level
bits.bIN13	0/1	P14 InPut level
bits.bIN14	0/1	P15 InPut level
bits.bIN15	0/1	P16 InPut level
bits.bQP08	0/1	P9 OutPut level
bits.bQP09	0/1	P10 OutPut level
bits.bQP10	0/1	P11 OutPut level
bits.bQP11	0/1	P12 OutPut level
bits.bQP12	0/1	P13 OutPut level
bits.bQP13	0/1	P14 OutPut level
bits.bQP14	0/1	P15 OutPut level
bits.bQP15	0/1	P16 OutPut level
byte.LocalInPut		
byte.LocalOutPut		
byte.RemoteInPut		
byte.RemoteOutPut		

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. //Macro definition
2.
3. #define UIM_BIT_MASK(bit)      (1 << bit)
4. #define UIM_BIT_SET(var, bit)  (var |= (UIM_BIT_MASK(bit)))
5. #define UIM_BIT_CLR(var, bit)  (var &= (~UIM_BIT_MASK(bit)))
6.
7. //-----Dividing Line -----
8.
9. ERRO err = 0; //Error definition
10. uint NodeID = 5; //Device's ID
11.
12.
13. DIGITAL_OUTPUT_OBJ obj = {}; //Define output parameters
14. UIM_DIO_OBJ AckValue = { 0 }; //Define return parameters
15.
16. obj.bits.bQP00 = 1; //Set P1port OutPut 1
17. obj.bits.bMSK00 = 1; //Set P1 port Mask 1, the output is valid
18. //Set IO port digital output P1 -> high Level
19. err = SdkSetDigitalOutput(g_GtwyHandle, NodeID, &obj, &AckValue);
20. if (err)
21. {
22.     std::cout << "Set Port Digital Output Failed!\n";
23.     return;
24. }
25.
26. PauseMS(1000); //delay time 1000ms
27.
28. obj.bits.bQP00 = 0; //Set the P1 port to output 0
29. obj.bits.bMSK00 = 1; //Set P1 port mask 1, the output is valid
30.
31. //Set IO port digital output P1 -> Low Level
32. err = SdkSetDigitalOutput(g_GtwyHandle, NodeID, &obj, &AckValue);
33. if (err)
34. {
35.     std::cout << "SetIOport 数字 OutPut Failed!\n";
36.     return;
37. }
38.
39. //IOport 循环 OutPut
40. bool bLoopIOTest = 1;
41. bool bHighLowRevert = 0;
42. long cnt = 0;
43.
44. while (bLoopIOTest)
45. {
46.
47.     // Switching between high and low Levels
48.     if (bHighLowRevert)
```

uirSDK3.0 Control System Development Kit

```
49. {
50.     //P1.. Set P8 to Low
51.     for (int i = 0; i < 8; i++)
52.     {
53.         UIM_BIT_SET(obj.byte.LocalOutPutMask, i);
54.         UIM_BIT_CLR(obj.byte.LocalOutPut, i);
55.     }
56.     //P9.. Set the P16 Level to Low
57.     for (int i = 8; i < 16; i++)
58.     {
59.         UIM_BIT_SET(obj.byte.RemoteOutPutMask, (i - 8));
60.         UIM_BIT_CLR(obj.byte.RemoteOutPut, (i - 8));
61.     }
62.     bHighLowRevert = 0;
63.     Sleep(5000);
64. }
65. else
66. {
67.     //P1.. P8 is set to high
68.     for (int i = 0; i < 8; i++)
69.     {
70.         UIM_BIT_SET(obj.byte.LocalOutPutMask, i);
71.         UIM_BIT_SET(obj.byte.LocalOutPut, i);
72.     }
73.
74.     //P9.. P16 is set to high
75.     for (int i = 8; i < 16; i++)
76.     {
77.         UIM_BIT_SET(obj.byte.RemoteOutPutMask, (i - 8));
78.         UIM_BIT_SET(obj.byte.RemoteOutPut, (i - 8));
79.     }
80.     bHighLowRevert = 1;
81.     Sleep(1000);
82. }
83.
84. cnt++;
85.
86. err = SdkSetDigitalOutput(g_GtwyHandle, NodeID, &obj, &AckValue);
87. if (err) { std::cout << "SdkSetDigitalOutput Fail!\n"; return; }
88.
89. std::cout << "Couter:
    " << cnt << ", Output:" << bHighLowRevert << std::endl;
90.
91. PauseMS(2000); //delay time2000ms
92.
93. if (cnt > 10) break;
94. }
```


uirSDK3.0 Control System Development Kit

9.4 Get IO Port Digital Status

Function Name: **ERRRO** **SdkGetDIOport**(uint GtwyHandle, uint CANid, UIM_DIO_OBJ* pRxData);

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pRxData(Return Value)	UIM_DIO_OBJ	yes	IO port input and output feedback (Refer to uirSDKcommon.h)

Legal Value:

Parameter	Legal Value	Note
pRxData	refer to 9.3 "Set Port Digital Output" Legal Value description	IO port input and output feedback (Refer to uirSDKcommon.h)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. uint NodeID = 5; //Device's ID
2. UIM_DIO_OBJ AckValue = { 0 }; //Define return parameters
3.
4. //Get InPut OutPut port status
5. ERRO err = SdkGetDIOport(g_GtwyHandle, NodeID, &AckValue);
6. if (err)
7. {
8.     std::cout << "Get IO Port Digital Status Failed!\n";
9.     return;
10. }
11. std::cout
12. << "IN00:" << AckValue.bits.bIN00 << "IN01:" << AckValue.bits.bIN01
13. << "IN02:" << AckValue.bits.bIN02 << "IN03:" << AckValue.bits.bIN03
14. << "IN04:" << AckValue.bits.bIN04 << "IN05:" << AckValue.bits.bIN05
15. << "IN06:" << AckValue.bits.bIN06 << "IN07:" << AckValue.bits.bIN07
16. << "IN08:" << AckValue.bits.bIN08 << "IN09:" << AckValue.bits.bIN09
17. << "IN10:" << AckValue.bits.bIN10 << "IN11:" << AckValue.bits.bIN11
18. << "IN12:" << AckValue.bits.bIN12 << "IN13:" << AckValue.bits.bIN13
19. << "IN14:" << AckValue.bits.bIN14 << "IN15:" << AckValue.bits.bIN15
20. << std::endl;
21.
22. std::cout
23. << "QP00:" << AckValue.bits.bQP00 << "QP01:" << AckValue.bits.bQP01
24. << "QP02:" << AckValue.bits.bQP02 << "QP03:" << AckValue.bits.bQP03
25. << "QP04:" << AckValue.bits.bQP04 << "QP05:" << AckValue.bits.bQP05
26. << "QP06:" << AckValue.bits.bQP06 << "QP07:" << AckValue.bits.bQP07
27. << "QP08:" << AckValue.bits.bQP08 << "QP09:" << AckValue.bits.bQP09
28. << "QP10:" << AckValue.bits.bQP10 << "QP11:" << AckValue.bits.bQP11
29. << "QP12:" << AckValue.bits.bQP12 << "QP13:" << AckValue.bits.bQP13
30. << "QP14:" << AckValue.bits.bQP14 << "QP15:" << AckValue.bits.bQP15
31. << std::endl;
```

uirSDK3.0 Control System Development Kit

9.5 Set Input Logic Acceleration

Function Name:

```
ERR0 SdkSetInputLogicAC(uint GtwyHandle, uint CANid, uint Catagory, uint data, uint* pRxData = NULL);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
Catagory	uint	yes	Trigger category
data	uint	yes	Acceleration value
pRxData(Return Value)	uint	no	Result feedback

Legal Value:

Parameter	Legal Value	Note
Catagory	0 (IOL_CAT_GENERAL)	Parameters applied to the received command
	1 (IOL_CAT_STALLED)	Motor stall trigger
	2 (IOL_CAT_IN1RISE)	IN1 Rising Edge Trigger
	3 (IOL_CAT_IN1FALL)	IN1 Falling Edge Trigger
	4 (IOL_CAT_IN2RISE)	IN2 Rising Edge Trigger
	5 (IOL_CAT_IN2FALL)	IN2 Falling Edge Trigger
	6 (IOL_CAT_IN3RISE)	IN3 Rising Edge Trigger
	7 (IOL_CAT_IN3FALL)	IN3 Falling Edge Trigger
	8 (IOL_CAT_IN4RISE)	IN4 Rising Edge Trigger
	9 (IOL_CAT_IN4FALL)	IN4 Falling Edge Trigger
data	1...65,000,000	Define acceleration and deceleration numerically, unit (pulse/sec ²)
	1...60,000	Define acceleration and deceleration in time, unit (ms)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. uint Catagory, data, pRxData;
4. Catagory = 2; //IN1rising edge trigger
5. data = 1000; //触发Acceleration value 1000 pulse/sec
6.
7. //Set input logic acceleration -> InPut port1 rising edge
   trigger,Acceleration value1000
8. err = SdkSetInputLogicAC(g_GtwyHandle, NodeID, Catagory, data, &pRxData);
9. if (err)
10. {
11.     std::cout << "Set input logic acceleration Failed!\n";
12.     return;
13. }
```

uirSDK3.0 Control System Development Kit

9.6 Get Input Logic Acceleration

Function Name:

```
ERR0 SdkGetInputLogicAC(uint GtwyHandle, uint CANid, uint Catagory, uint* pRxData);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
Catagory	uint	yes	Trigger category
pRxData(Return Value)	uint	yes	Result feedback

Legal Value:

Parameter	Legal Value	Note
Catagory	0 (IOL_CAT_GENERAL)	Parameters applied to the received command
	1 (IOL_CAT_STALLED)	Motor stall trigger
	2 (IOL_CAT_IN1RISE)	IN1 Rising Edge Trigger
	3 (IOL_CAT_IN1FALL)	IN1 Falling Edge Trigger
	4 (IOL_CAT_IN2RISE)	IN2 Rising Edge Trigger
	5 (IOL_CAT_IN2FALL)	IN2 Falling Edge Trigger
	6 (IOL_CAT_IN3RISE)	IN3 Rising Edge Trigger
	7 (IOL_CAT_IN3FALL)	IN3 Falling Edge Trigger
	8 (IOL_CAT_IN4RISE)	IN4 Rising Edge Trigger
	9 (IOL_CAT_IN4FALL)	IN4 Falling Edge Trigger
pRxData	1...65,000,000	Define acceleration and deceleration numerically, unit (pulse/sec ²)
	1...60,000	Define acceleration and deceleration in time, unit (ms)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. uint NodeID = 5; //Device's ID
2. uint Catagory, MotorAC, pRxData; //Define parameters
3. Catagory = 5; //IN2 falling edge trigger
4.
5. //Get Input Logic Acceleration -> InPut port2 falling edge trigger
6. ERRO err = SdkSetInputLogicAC(g_GtwyHandle, NodeID, Catagory, &pRxData);
7. if (err)
8. {
9.     std::cout << "Get Input Logic Acceleration Failed!\n";
10.    return;
11. }
12. MotorAC = pRxData; // InPut port2 falling edge trigger Acceleration value
```

uirSDK3.0 Control System Development Kit

9.7 Set Input Logic Deceleration

Function Name:

```
ERROR SdkSetInputLogicDC(uint GtwyHandle, uint CANid, uint Catagory, uint data, uint* pRxData = NULL);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
Catagory	uint	yes	Trigger category
data	uint	yes	Deceleration value
pRxData(Return Value)	uint	no	Result feedback

Legal Value:

Parameter	Legal Value	Note
Catagory	0 (IOL_CAT_GENERAL)	Parameters applied to the received command
	1 (IOL_CAT_STALLED)	Motor stall trigger
	2 (IOL_CAT_IN1RISE)	IN1 Rising Edge Trigger
	3 (IOL_CAT_IN1FALL)	IN1 Falling Edge Trigger
	4 (IOL_CAT_IN2RISE)	IN2 Rising Edge Trigger
	5 (IOL_CAT_IN2FALL)	IN2 Falling Edge Trigger
	6 (IOL_CAT_IN3RISE)	IN3 Rising Edge Trigger
	7 (IOL_CAT_IN3FALL)	IN3 Falling Edge Trigger
	8 (IOL_CAT_IN4RISE)	IN4 Rising Edge Trigger
	9 (IOL_CAT_IN4FALL)	IN4 Falling Edge Trigger
data	1...65,000,000	Define acceleration and deceleration numerically, unit (pulse/sec ²)
	1...60,000	Define acceleration and deceleration in time, unit (ms)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3.
4.
5. uint Catagory, data, pRxData;
6. Catagory = 3; //IN1 falling edge trigger
7. data = 2000; //Deceleration value 2000 pulse/sec
8.
9. //Set input logic deceleration -> InPut port1 falling edge
   trigger,Deceleration value2000
10. err = SdkSetInputLogicDC(g_GtwyHandle, NodeID, Catagory, data, &pRxData);
11. if (err)
12. {
13.     std::cout << "Set input logic deceleration Failed!\n";
14.     return;
15. }
```

uirSDK3.0 Control System Development Kit

9.8 Get Input Logic Deceleration

Function Name:

```
ERR0 SdkGetInputLogicDC(uint GtwyHandle, uint CANid, uint Catagory, uint* piDeceleration);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
Catagory	uint	yes	Trigger category
piDeceleration(Return Value)	uint	yes	Result feedback

Legal Value:

Parameter	Legal Value	Note
Catagory	0 (IOL_CAT_GENERAL)	Parameters applied to the received command
	1 (IOL_CAT_STALLED)	Motor stall trigger
	2 (IOL_CAT_IN1RISE)	IN1 Rising Edge Trigger
	3 (IOL_CAT_IN1FALL)	IN1 Falling Edge Trigger
	4 (IOL_CAT_IN2RISE)	IN2 Rising Edge Trigger
	5 (IOL_CAT_IN2FALL)	IN2 Falling Edge Trigger
	6 (IOL_CAT_IN3RISE)	IN3 Rising Edge Trigger
	7 (IOL_CAT_IN3FALL)	IN3 Falling Edge Trigger
	8 (IOL_CAT_IN4RISE)	IN4 Rising Edge Trigger
	9 (IOL_CAT_IN4FALL)	IN4 Falling Edge Trigger
piDeceleration	1...65,000,000	Define acceleration and deceleration numerically, unit (pulse/sec ²)
	1...60,000	Define acceleration and deceleration in time, unit (ms)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. uint Catagory, MotorDC, pRxData;//Define parameters
4. Catagory = 4; //IN2 rising edge trigger
5.
6. //Get Input Logic Deceleration -> InPut port2 rising edge trigger
7. err = SdkGetInputLogicDC(g_GtwyHandle, NodeID, Catagory, &pRxData);
8. if (err)
9. {
10.     std::cout << "Get Input Logic Deceleration Failed!\n";
11.     return;
12. }
13. MotorDC = pRxData; // Deceleration value
```

uirSDK3.0 Control System Development Kit

9.9 Set Input Logic Speed

Function Name:

```
ERR0 SdkSetInputLogicSP(uint GtwyHandle, uint CANid, uint Catagory, int PTPSpeed, int* pRxPTPSpd = NULL);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
Catagory	uint	yes	Trigger category
PTPSpeed	int	yes	Speed value
pRxPTPSpd(Return Value)	int	no	Result feedback

Legal Value:

Parameter	Legal Value	Note
Catagory	0 (IOL_CAT_GENERAL)	Parameters applied to the received command
	1 (IOL_CAT_STALLED)	Motor stall trigger
	2 (IOL_CAT_IN1RISE)	IN1 Rising Edge Trigger
	3 (IOL_CAT_IN1FALL)	IN1 Falling Edge Trigger
	4 (IOL_CAT_IN2RISE)	IN2 Rising Edge Trigger
	5 (IOL_CAT_IN2FALL)	IN2 Falling Edge Trigger
	6 (IOL_CAT_IN3RISE)	IN3 Rising Edge Trigger
	7 (IOL_CAT_IN3FALL)	IN3 Falling Edge Trigger
	8 (IOL_CAT_IN4RISE)	IN4 Rising Edge Trigger
	9 (IOL_CAT_IN4FALL)	IN4 Falling Edge Trigger
PTPSpeed	$-2^{31} \dots +2^{31}$	Speed value, unit (pulse/sec)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. uint Catagory, PTPSpeed, pRxPTPSpd;
4. Catagory = 3; //IN1 falling edge trigger
5. PTPSpeed = 2000; //speed value 2000 pulse/sec
6.
7. //Set Input logic speed -> port1 falling edge trigger, speed value 2000
8. err=SdkSetInputLogicSP(g_GtwyHandle,NodeID,Catagory,PTPSpeed,&pRxPTPSpd);
9. if (err)
10. {
11.     std::cout << "Set Input logic speed Failed!\n";
12.     return;
13. }
```

uirSDK3.0 Control System Development Kit

9.10 Get Input Logic Speed

Function Name:

```
ERR0 SdkGetInputLogicSP(uint GtwyHandle, uint CANid, uint Catagory, int* pRxPTPSpd);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
Catagory	uint	yes	Trigger category
pRxPTPSpd(Return Value)	int	yes	Result feedback

Legal Value:

Parameter	Legal Value	Note
Catagory	0 (IOL_CAT_GENERAL)	Parameters applied to the received command
	1 (IOL_CAT_STALLED)	Motor stall trigger
	2 (IOL_CAT_IN1RISE)	IN1 Rising Edge Trigger
	3 (IOL_CAT_IN1FALL)	IN1 Falling Edge Trigger
	4 (IOL_CAT_IN2RISE)	IN2 Rising Edge Trigger
	5 (IOL_CAT_IN2FALL)	IN2 Falling Edge Trigger
	6 (IOL_CAT_IN3RISE)	IN3 Rising Edge Trigger
	7 (IOL_CAT_IN3FALL)	IN3 Falling Edge Trigger
	8 (IOL_CAT_IN4RISE)	IN4 Rising Edge Trigger
	9 (IOL_CAT_IN4FALL)	IN4 Falling Edge Trigger
pRxPTPSpd	$-2^{31} \dots +2^{31}$	Speed value, unit (pulse/sec)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. uint NodeID = 5; //Device's ID
2.
3.
4. uint Catagory, MotorSpd, pRxData; //Define parameters
5. Catagory = 4; //IN2 rising edge trigger
6.
7. //Get Input Logic Speed -> InPut port2 rising edge trigger
8. ERRO err = SdkGetInputLogicDC(g_GtwyHandle, NodeID, Catagory, &pRxData);
9. if (err)
10. {
11.     std::cout << "Get Input Logic Speed Failed!\n";
12.     return;
13. }
14. MotorSpd = pRxData; // speed value
```

uirSDK3.0 Control System Development Kit

9.11 Set Input Logic Relative Position

Function Name:

```
ERRRO SdkSetInputLogicPR(uint GtwyHandle, uint CANid, uint Catagory, int data, int* pRxData = NULL);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
Catagory	uint	yes	Trigger category
data	int	yes	Relative position
pRxData(Return Value)	int	no	Result feedback

Legal Value:

Parameter	Legal Value	Note
Catagory	0 (IOL_CAT_GENERAL)	Parameters applied to the received command
	1 (IOL_CAT_STALLED)	Motor stall trigger
	2 (IOL_CAT_IN1RISE)	IN1 Rising Edge Trigger
	3 (IOL_CAT_IN1FALL)	IN1 Falling Edge Trigger
	4 (IOL_CAT_IN2RISE)	IN2 Rising Edge Trigger
	5 (IOL_CAT_IN2FALL)	IN2 Falling Edge Trigger
	6 (IOL_CAT_IN3RISE)	IN3 Rising Edge Trigger
	7 (IOL_CAT_IN3FALL)	IN3 Falling Edge Trigger
	8 (IOL_CAT_IN4RISE)	IN4 Rising Edge Trigger
	9 (IOL_CAT_IN4FALL)	IN4 Falling Edge Trigger
data	$-2^{31} \dots +2^{31}$	Relative position, unit (pulse)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. uint Catagory;
4. int data, pRxData; //Define parameters
5. Catagory = 2; //IN1 rising edge trigger
6. data = 10000; //Relative position 10000(pulse)
7.
8. //Set Input Logic Relative Position -> IN1 rising edge trigger, Relative
   position 10000(pulse)
9. err = SdkSetInputLogicPR(g_GtwyHandle, NodeID, Catagory, data, &pRxData);
10. if (err)
11. {
12.     std::cout << "Set Input Logic Relative Position Failed!\n";
13.     return;
14. }
```

uirSDK3.0 Control System Development Kit

9.12 Get Input Logic Relative Position

Function Name:

```
ERR0 SdkGetInputLogicPR(uint GtwyHandle, uint CANid, uint Catagory, int* pRxData);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
Catagory	uint	yes	Trigger category
pRxData(Return Value)	int	yes	Result feedback

Legal Value:

Parameter	Legal Value	Note
Catagory	0 (IOL_CAT_GENERAL)	Parameters applied to the received command
	1 (IOL_CAT_STALLED)	Motor stall trigger
	2 (IOL_CAT_IN1RISE)	IN1 Rising Edge Trigger
	3 (IOL_CAT_IN1FALL)	IN1 Falling Edge Trigger
	4 (IOL_CAT_IN2RISE)	IN2 Rising Edge Trigger
	5 (IOL_CAT_IN2FALL)	IN2 Falling Edge Trigger
	6 (IOL_CAT_IN3RISE)	IN3 Rising Edge Trigger
	7 (IOL_CAT_IN3FALL)	IN3 Falling Edge Trigger
	8 (IOL_CAT_IN4RISE)	IN4 Rising Edge Trigger
	9 (IOL_CAT_IN4FALL)	IN4 Falling Edge Trigger
pRxData	-2 ³¹ ...+2 ³¹	Relative position, unit (pulse)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. uint NodeID = 5; //Device's ID
2. uint Catagory;
3. int Pr, pRxData; //Define parameters
4. Catagory = 2; //IN1 上升沿触发
5.
6. //Get Input Logic Relative Position -> IN1 上升沿触发
7. ERRO err = SdkGetInputLogicPR(g_GtwyHandle, NodeID, Catagory, &pRxData);
8. if (err)
9. {
10.     std::cout << "Get Input Logic Relative Position Failed!\n";
11.     return;
12. }
13. Pr = pRxData; //IN1 上升沿触发, 位移值
```

uirSDK3.0 Control System Development Kit

9.13 Set Input Logic Absolute Position

Function Name:

```
ERR0 SdkSetInputLogicPA(uint GtwyHandle, uint CANid, uint Catagory, int data, int* pRxData = NULL);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
Catagory	uint	yes	Trigger category
data	int	yes	Absolute position
pRxData(Return Value)	int	no	Result feedback

Legal Value:

Parameter	Legal Value	Note
Catagory	0 (IOL_CAT_GENERAL)	Parameters applied to the received command
	1 (IOL_CAT_STALLED)	Motor stall trigger
	2 (IOL_CAT_IN1RISE)	IN1 Rising Edge Trigger
	3 (IOL_CAT_IN1FALL)	IN1 Falling Edge Trigger
	4 (IOL_CAT_IN2RISE)	IN2 Rising Edge Trigger
	5 (IOL_CAT_IN2FALL)	IN2 Falling Edge Trigger
	6 (IOL_CAT_IN3RISE)	IN3 Rising Edge Trigger
	7 (IOL_CAT_IN3FALL)	IN3 Falling Edge Trigger
	8 (IOL_CAT_IN4RISE)	IN4 Rising Edge Trigger
	9 (IOL_CAT_IN4FALL)	IN4 Falling Edge Trigger
data	$-2^{31} \dots +2^{31}$	Position value, unit (pulse)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. uint Catagory;
4. int data, pRxData; //Define parameters
5. Catagory = 3; //IN1 falling edge trigger
6. data = 10000; //Position 10000(pulse)
7.
8. //Set Input Logic Absolute Position -> IN1 falling edge trigger, position
   10000 (pulse)
9. err = SdkSetInputLogicPA(g_GtwyHandle, NodeID, Catagory, data, &pRxData);
10. if (err)
11. {
12.     std::cout << "Set Input Logic Absolute Position Failed!\n";
13.     return;
14. }
```

uirSDK3.0 Control System Development Kit

9.14 Get Input Logic Absolute Position

Function Name:

```
ERR0 SdkGetInputLogicPA(uint GtwyHandle, uint CANid, uint Catagory, int* pRxData);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
Catagory	uint	yes	Trigger category
pRxData(Return Value)	int	yes	Position value feedback

Legal Value:

Parameter	Legal Value	Note
Catagory	0 (IOL_CAT_GENERAL)	Parameters applied to the received command
	1 (IOL_CAT_STALLED)	Motor stall trigger
	2 (IOL_CAT_IN1RISE)	IN1 Rising Edge Trigger
	3 (IOL_CAT_IN1FALL)	IN1 Falling Edge Trigger
	4 (IOL_CAT_IN2RISE)	IN2 Rising Edge Trigger
	5 (IOL_CAT_IN2FALL)	IN2 Falling Edge Trigger
	6 (IOL_CAT_IN3RISE)	IN3 Rising Edge Trigger
	7 (IOL_CAT_IN3FALL)	IN3 Falling Edge Trigger
	8 (IOL_CAT_IN4RISE)	IN4 Rising Edge Trigger
	9 (IOL_CAT_IN4FALL)	IN4 Falling Edge Trigger
pRxData	$-2^{31} \dots +2^{31}$	Position value, unit (pulse)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

uirSDK3.0 Control System Development Kit

Example code (C++):

```
1. ERRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. uint Catagory;
4. int PA, pRxData; //Define parameters
5. Catagory = 2; //IN1 rising edge trigger
6.
7. //Get Input Logic Absolute Position -> IN1 rising edge trigger
8. err = SdkGetInputLogicPA(g_GtwyHandle, NodeID, Catagory, &pRxData);
9. if (err)
10. {
11.     std::cout << "Get Input Logic Absolute Position Failed!\n";
12.     return;
13. }
14. PA = pRxData; //position value
```

uirSDK3.0 Control System Development Kit

9.15 Set Input Logic

Function Name:

```
ERROR SdkSetInputLogic(uint GtwyHandle, uint CANid, uint PortIx, INPUT_ACT_OBJ MxnCode, INPUT_ACT_OBJ* pRxData = NULL);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
PortIx	uint	yes	IO port number
MxnCode	INPUT_ACT_OBJ	yes	Action code (Refer to chapter 1.2)
pRxData(Return Value)	INPUT_ACT_OBJ	no	Result feedback (Refer to chapter 1.2)

Legal Value:

Parameter	Attributes	Legal Value	Note
PortIx		0...15	IO port number
MxnCode	Action0/ Action1 (refer to chapter 1.2)	0 (ILC_NOP_IDX)	No action
		1 (ILC_OFF_IDX)	Offline
		2 (ILC_EST_IDX)	Stop quickly
		3 (ILC_DST_IDX)	Deceleration to stop
		4 (ILC_OPR_IDX)	Origin + commutation + movement according to displacement abs (PR)
		5 (ILC_OES_IDX)	Origin+quick stop
		6 (ILC_ODS_IDX)	Origin + deceleration stop
		7 (ILC_RJV_IDX)	Change direction + press the preset abs (JV) movement
		8 (ILC_SJV_IDX)	Press Default +/-JV Movement
		9 (ILC_RPR_IDX)	Change direction + move according to preset displacement abs (PR)
		10 (ILC_SPR_IDX)	Move according to preset displacement (+/-PR)
		11 (ILC_SPA_IDX)	Exercise according to preset position (+/-PA)
	bNoTrig0/ bNoTrig1 (refer to chapter 1.2)	15 (ILC_PVT_IDX)	Execute PVT motion according to preset parameters
		0/1	Power on trigger status (0 - trigger; 1 - no action)

Return Value:

uirSDK3.0 Control System Development Kit

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. //Input trigger port details
2. typedef union _INPUT_ACT_OBJ
3. {
4.     struct
5.     {
6.         unsigned Action0 : 7; // Falling edge action code
7.         unsigned bNoTrig0 : 1; //Power-on trigger status, 0-power-on
           trigger, 1-power-on not trigger
8.
9.         unsigned Action1 : 7; // rising edge action code
10.        unsigned bNoTrig1 : 1; //Power-on trigger status, 0-power-on
           trigger, 1-power-on not trigger
11.        uchar res[2];
12.    };
13.
14.    uint w;
15.
16. }INPUT_ACT_OBJ;
17.
18. //-----Dividing line -----
19.
20. ERRO err = 0; //Error definition
21. uint NodeID = 5; //Device's ID
22.
23. uint RxData; //Define parameters
24.
25. INPUT_ACT_OBJ InputAct = { 0 };
26. // Set falling edge MxnCode = 5 (origin + rapid stop)
27. InputAct.Action0 = ILG_SORG_JVSD_IDX;
28. InputAct.bNoTrig0 = 1; //Does not trigger when powered on
29.
30. // Set rising edge MxnCode = 0 (no action)
31. InputAct.Action1 = ILG_NULL_0000_IDX;
32. InputAct.bNoTrig1 = 1; //Does not trigger when powered on
33. //Set Input Logic, InPut port1
34. err = SdkSetInputLogic(g_GtwyHandle, NodeID, 0, InputAct.w, &RxData);
35. if (err)
36. {
37.     std::cout << "Set Input Logic Failed!\n";
38.     return;
39. }
```


uirSDK3.0 Control System Development Kit

9.16 Get Input Logic

Function Name:

ERROR `SdkGetInputLogic(uint GtwyHandle, uint CANid, uint PortIx, INPUT_ACT_OBJ* pRxData);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
PortIx	uint	yes	IO port number
pRxData(Return Value)	INPUT_ACT_OBJ	yes	Result feedback

Legal Value:

Parameter	Attributes	Legal Value	Note
PortIx		0...15	IO port number
pRxData	Action0/ Action1 (refer to chapter 1.2)	0 (ILC_NOP_IDX)	No action
		1 (ILC_OFF_IDX)	Offline
		2 (ILC_EST_IDX)	Stop quickly
		3 (ILC_DST_IDX)	Deceleration to stop
		4 (ILC_OPR_IDX)	Origin + commutation + movement according to displacement abs (PR)
		5 (ILC_OES_IDX)	Origin+quick stop
		6 (ILC_ODS_IDX)	Origin + deceleration stop
		7 (ILC_RJV_IDX)	Change direction + press the preset abs (JV) movement
		8 (ILC_SJV_IDX)	Press Default +/-JV Movement
		9 (ILC_RPR_IDX)	Change direction + move according to preset displacement abs (PR)
		10 (ILC_SPR_IDX)	Move according to preset displacement (+/-PR)
		11 (ILC_SPA_IDX)	Exercise according to preset position (+/-PA)
pRxData	bNoTrig0/ bNoTrig1 (refer to chapter 1.2)	15 (ILC_PVT_IDX)	Execute PVT motion according to preset parameters
		0/1	Power on trigger status (0 - trigger; 1 - no action)

Return Value:

0	No Error (Success)
---	--------------------

uirSDK3.0 Control System Development Kit

!0 Error Code (refer to UIError.h)

Example code (C++):

```
1. // Input trigger port details
2. typedef union _INPUT_ACT_OBJ
3. {
4.     struct
5.     {
6.         unsigned Action0 : 7;    // Falling edge action code
7.         unsigned bNoTrig0 : 1;   // Power-on trigger status, 0-power-on trigger,
            1-power-on not trigger
8.
9.         unsigned Action1 : 7;    // rising edge action code
10.        unsigned bNoTrig1 : 1;   // Power-on trigger status, 0-power-on trigger,
            1-power-on not trigger
11.        uchar res[2];
12.    };
13.
14.    uint w;
15.
16. }INPUT_ACT_OBJ;
17.
18. //----- Dividing Line -----
19.
20. #define ERR0 err = 0; //Error definition
21. #define NodeID 5; //Device's ID
22. //g_GtwyHandle Get the gateway handle, see 2.1 Open CAN Network
23.
24. INPUT_ACT_OBJ InputAct = { 0 };
25. err = SdkGetInputLogic(g_GtwyHandle, NodeID,0,&InputAct.w);
26. if (err)
27. {
28.     std::cout << "9.16 Get Input Logic Failed!\n";
29.     return;
30. }
31.
32. uint* ActionCode = 0; //define action code
33. uint* PowerOnTrg = 0; //define poweron trigger status
34.
35. ActionCode = InputAct.Action1; // rising edge action code
36. ActionCode = InputAct.Action0; // Falling edge action code
37. PowerOnTrg = InputAct.bNoTrig1; // rising edge trigger status
38. PowerOnTrg = InputAct.bNoTrig0; // Falling edge trigger status
```

10.0 IO MODULE

This chapter will introduce how to Get or Set the encoder function of the IO module through functions.

10.1 Set Auxiliary Encoder Period

Function Name:

```
ERRO SdkSetEncoderPeriod(uint GtwyHandle, uint CANid, uint Data, uint* pRxData = NULL);
```

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
Data	uint	yes	Encoder Count Period Value
pRxData	uint	no	Result feedback

Legal Value:

Return Value	Legal Value	Note
Data	0	Disable Encoder Period
	1...65535	Encoder Count Period Value (Unit: ms)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. uint NodeID = 5; //Device's ID
2.
3. int Period, pRxCount; //Define parameters
4. Period = 200; //period time 200ms
5.
6. //Set Auxiliary Encoder Period
7. ERRO err = SdkSetEncoderCount(g_GtwyHandle, NodeID, Period, &pRxCount);
8. if (err)
9. {
10.     std::cout << "Set Auxiliary Encoder PeriodFailed!\n";
11.     return;
12. }
```


uirSDK3.0 Control System Development Kit

10.2 Reset Auxiliary Encoder Count

Function Name: `ERRO SdkRstEncoderCount(uint GtwyHandle, uint CANid);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. uint NodeID = 5; //Device's ID
2.
3. //Reset Auxiliary Encoder Count
4. ERRO err = SdkRstEncoderCount(g_GtwyHandle, NodeID);
5. if (err)
6. {
7.     std::cout << "10.1 Reset Auxiliary Encoder Count Failed!\n";
8.     return;
9. }
```

uirSDK3.0 Control System Development Kit

10.3 Get Auxiliary Encoder Speed

Function Name: `ERRRO SdkGetEncoderSpeed(uint GtwyHandle, uint CANid, int* pRxData);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pRxData(Return Value)	int	yes	Encoder Value (per second) feedback

Legal Value:

Return Value	Legal Value	Note
pRxData		Encoder Value (per second)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. ERRRO err = 0; //Error definition
2. uint NodeID = 5; //Device's ID
3. int Count, pRxCount; //Define parameters
4. //Get Auxiliary Encoder Speed - Encoder Value (per second)
5. err = SdkGetEncoderSpeed(g_GtwyHandle, NodeID, &pRxCount);
6. if (err)
7. {
8.     std::cout << "Get Auxiliary Encoder Speed Failed!\n";
9.     return;
10. }
11. Count = pRxCount; //Encoder Value pps
```

uirSDK3.0 Control System Development Kit

10.4 Get Auxiliary Encoder Count

Function Name: **ERR0** `SdkGetEncoderCount`(uint GtwyHandle, uint CANid, int* pRxCount);

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pRxCount(Return Value)	int	yes	Encoder Value (abs position) feedback

Legal Value:

Return Value	Legal Value	Note
pRxCount	0...65535	Encoder Value (abs position) feedback

Return Value:

Return Value	Note
0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. uint NodeID = 5; //Device's ID
2. int Count, pRxCount; //Define parameters
3. //Get Auxiliary Encoder Count
4. ERR0 err = SdkGetEncoderCount(g_GtwyHandle, NodeID, &pRxCount);
5. if (err)
6. {
7.     std::cout << "Get Auxiliary Encoder Count Failed!\n";
8.     return;
9. }
10. Count = pRxCount; //Encoder Value
```

uirSDK3.0 Control System Development Kit

10.5 Get Auxiliary Encoder Increment

Function Name: `ERR0 SdkGetEncoderIncrement(uint GtwyHandle, uint CANid, int* pRxData);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
CANid	uint	yes	Device's ID
pRxData(Return Value)	int	yes	Encoder Counts value within period time

Legal Value:

Return Value	Legal Value	Note
pRxData		Encoder Counts value within period time (This function requires the encoder counting period to be set)

Return Value:

0	No Error (Success)
!0	Error Code (refer to UIError.h)

Example code (C++):

```
1. uint NodeID = 5; //Device's ID
2.
3. int Count, pRxCount; //Define parameters
4. //Get Auxiliary Encoder Increment - Encoder Value(within the period time)
5. ERR0 err = SdkGetEncoderIncrement(g_GtwyHandle, NodeID, &pRxCount);
6. if (err)
7. {
8.     std::cout << "Get Auxiliary Encoder Increment Failed!\n";
9.     return;
10. }
11. Count = pRxCount; //Encoder Value Encoder Value(within the period time)
```


11.0 OTHER INSTRUCTIONS

This chapter will introduce how to use Real-Time Notification function.

11.1 Real-Time Notification

Function Name: `SdkGetRTCN(uint GtwyHandle, UI_MSG_OBJ* pUiMsg);`

Parameter List:

Parameter	Type	Required	Note
GtwyHandle	uint	yes	Gateway Handle (Refer to chapter 2.1)
pUiMsg	UI_MSG_OBJ	yes	Notification Message (Refer to uirSDKcommon.h)

Legal Value:

Parameter	Legal Value	Note
pUiMsg	UiMsg details refer to Motion Controller User Manual	Notification Message (Refer to chapter 1.2)

Return Value:

1	No Error (Success)
0	Error Code (refer to UIError.h)

Example code (C++):

```
1. //-----Macro definition -----
2. #define RTCN_MXN_STP  41
3. #define RTCN_MXN_ORG  42
4. #define RTCN_MXN_PVW  44
5. #define RTCN_MXN_PVS  45
6. #define RTCN_MXN_TLC  50
7. #define RTCN_MXN_ST0  51
8. #define RTCN_MXN_ST1  52
9. #define RTCN_UPG_PRT  60
10.
11.//----- Dividing Line -----
12. UI_MSG_OBJ uimsg = { 0 };
13.
14. ERRO retv = SdkGetRTCN(g_GtwyHandle, &uimsg);
15.
16. uchar rtcn_code = uimsg.data[0];
17.
18. if (retv == 1)
19. {
20. switch (rtcn_code)
```

```
21. {
22.     case RTCN_MXN_STL: break; // Blocked rotation detected
23.
24.     case RTCN_MXN_TLC: break; // The torque limit is detected, and the m
        otor position is exceeded;
25.     case RTCN_MXN_ST0:
26.     case RTCN_MXN_ST1:
27.     {
28.         if (rtcn_code == RTCN_MXN_ST0)
29.         {
30.             // Get system status
31.
32.             MOTION_STATUS_OBJ MotionStatus = { 0 }; // Motor state
33.             memcpy(&MotionStatus, &uimsg.data[1], 2);
34.             // Obtain the relative displacement of the motor
35.             int CurPosR = 0; // displacement
36.             memcpy(&CurPosR, &uimsg.data[4], 4);
37.         }
38.         else if (rtcn_code == RTCN_MXN_ST1)
39.         {
40.             // Gain motor speed
41.             int CurSpeed = 0;
42.             memcpy(&CurSpeed, &uimsg.data[1], 3);
43.             if (uimsg.data[3] & 0x80) gCurSpeed |= 0xFF000000;
44.             // Gain motor speed
45.             int CurPosA = 0; // position
46.             memcpy(&CurPosA, &uimsg.data[4], 4);
47.         }
48.     }
49.     break; // Get the RTCN push
50.     default: break;
51. }
```

12.0 PROGRAMMING EXAMPLES

12.1 C ++ routine description

Open the uirSDK_VC_Demo.sln file, see the Example code.

12.2 C# routine description

Open the uirSDK_CSharp_Demo source code. The routine has converted the relevant header files and data structures into C# into the required style. The code is in csUIM.cs, and there is no need to use the header files under the uirSDKFiles folder. After No Error (Success)ful compilation, when running, the corresponding uirWin32SDK.dll needs to be placed in the bin directory.

Detailss are slightly different in C# and C++. For an introduction to structures, please refer to the introduction to C++ structures.

12.3 Program writing process description

- Open the gateway using SdkStartCanNet.
- Sets up the create thread function on No Error (Success) to process the real-time message (call SdkGetRTCN).
- To send commands to the UIM342, the motor needs to be enabled (SdkSetMotorOn) first, and then can send speed (SdkSetJogMxn), relative displacement (SdkSetPtpMxnR), and position control (SdkSetPtpMxnA).
- If you send the position control command (SdkSetPtpMxnA), you need to wait for UIM342 to send the position message (set the position notification function through SdkSetInformEnable). You can obtain the position message from the callback function SdkGetRTCN.
- To program the process of writing back to the sensor, connect the S1 sensor port on the UIM342 to the sensor and run the SdkSetInputLogic function to set S1 falling edge to SCR + Clear. Configure (through SdkSetInformEnable function to set S1 status change notification function turned on), send speed instruction SdkSetJogMxn (generally negative) to the origin direction. When the motor senses the origin, UIM342 will cause the motor to stop, and send S1 notification, S1 position notification can be obtained through the SdkGetRTCN function.