## Communication protocol implement by VB language

APIs of communication with resources

* **OpenBox**

This function is used to initiate communication with different resources; it implements the communication board in document, to switch channels of resources.

*Function OpenBox(ByVal PortID As Short, ByVal DoorNo As Short, Optional ByRef SettingStr As String = "19200,n,8,2") As String*

**Return Values**

If the function succeeds, it returns characters >>.

If the function fails, it returns error information

**Parameters**

PortID: Standard RS232 comm port identifier. Comm1 = 1, Comm2 = 2,

etc.

SettingStr: “19200,n,8,2” (const)

DoorNo: 1 (error calculator device unit), 2 (power source), 7 (standard meter), 13 (element 1), 14 (element 2).

APIs of Power Source

* **ConnectModeCtrl**

This function is to change the meter type of the test bench, it implements the command 0x11 in the Communication Protocol document, to set output phase sequence, meter type, load type

**VB declare as below**

*Function ConnectModeCtrl(ByVal PortID As Short, ByVal SettingStr As String, ByVal DeviceID As Short, ByVal PM As String, ByVal CM As String, ByVal fLoadL As Boolean) As String*

**Return Values**

If the function succeeds, it returns characters >>.

If the function fails, it returns error information

**Parameters**

PortID: Standard RS232 comm port identifier. Comm1 = 1, Comm2 = 2, etc.

SettingStr: “19200,n,8,2” (const)

DeviceID: 220 phase A, 221 phase B, 222 phase C

PM: Phase sequence (reserved)

CM:

D - Single phase

4w – P3-3 4-wire active power

4R – P3-2 4Wire react

3W – Q3-3 3Wire active

3R – Q3-2 3W reactive

fLoadL: false(mechanical meter), true (electronic meter)

* **CosOut**

This function is to change the phase angle, it implements the command 0xOC in the Communication Protocol document, to set output phase angle

**VB declare as below**

*Function CosOut(ByVal PortID As Short, ByVal SettingStr As String, ByVal DeviceID As Short, ByVal CosData As Double) As String*

**Return Values**

If the function succeeds, it returns characters >>.

If the function fails, it returns error information

**Parameters**

PortID: Standard RS232 comm port identifier. Comm1 = 1, Comm2 = 2,

etc.

SettingStr: “19200,n,8,2” (const)

DeviceID: 220 phase A, 221 phase B, 222 phase C

CosData: numerical value of phase angle

* **CurrentOut**

This function is to set the current output, it implements the command 0x0A in the Communication Protocol document, to set output of current only

**VB declare as below**

*Function CurrentOut(ByVal PortID As Short, ByVal SettingStr As String, ByVal DeviceID As Short, ByVal CurrentData As Double, Optional ByVal FloatOutM As Boolean = Nothing) As String*

**Return Values**

If the function succeeds, it returns characters >>.

If the function fails, it returns error information

**Parameters**

PortID: Standard RS232 comm port identifier. Comm1 = 1, Comm2 = 2,

etc.

SettingStr: “19200,n,8,2” (const)

DeviceID: 220 phase A, 221 phase B, 222 phase C

CurrentData: numerical value of current output

FloatOutM: set to true for high precision output

* **SignalSourceCtrl**

This function is to set outputs including voltage, current, phase angle, frequency at a time on one phase, it implements the command 0x12 of communication protocol of power souce.

**VB declare as below**

*Function SignalSourceCtrl(ByVal PortID As Short, ByVal SettingStr As String, ByVal DeviceID As Short, ByVal CurrentData As Double, ByVal VoltageData As Double, ByVal CosData As Double, ByVal FrequencyData As Double, Optional ByVal FloatOutM As Boolean = Nothing) As String*

**Return Values**

If the function succeeds, it returns characters >>.

If the function fails, it returns error information

**Parameters**

PortID: Standard RS232 comm port identifier. Comm1 = 1, Comm2 = 2,

etc.

SettingStr: “19200,n,8,2” (const)

DeviceID: 220 phase A, 221 phase B, 222 phase C

CurrentData: numerical value of current output

VoltageData: numerical value of voltage output

CosData: numerical value of phase angle output

FrequencyData: numerical value of frequency output

FloatOutM: set to true for high precision output

APIs of Error Calculator

* **ErrCounterTest**

This function is used for setting of accuracy test.

**VB declare as below**

*Function ErrCounterTest(ByVal PortID As Short, ByVal SettingStr As String, ByVal DeviceID As Short, CircleNu ByVal As Short, ByVal stdImpulse As Double, ByVal AccessNu As Short, ByVal ImpulseMode As Short, ByVal UpLimit As Double, ByVal DownLimit As Double) As String*

**Return Values**

If the function succeeds, it returns characters >>.

If the function fails, it returns error information

**Parameters**

PortID: Standard RS232 comm port identifier. Comm1 = 1, Comm2 = 2,

etc.

SettingStr: “19200,n,8,2” (const)

DeviceID: meter position or 199, 199(broadcast mode), for example, no.1 position goes to accuracy test, set value with 1, and so on.

CircleNu: circle number of accuracy test. (Impulse number input)

stdImpulse: that’s value need to be calculated by this formula  
 standard impulse number = circles \* co / cx

where: circles = circleNu (number of samples), co= Standard const (the multiplier.. ie:10e9, cx= meter constant (imp/kWH)

AccessNu: *access number channel of meter impulse input. Default is 1 or 0(scanning head input)*

ImpulseMode: default 0 for scanning head using

UpLimit: 100 (default)

DownLimit: -100 (default)

* **ErrCounterReadDataAll**

This function is to read the value of error from 6 meter position at a same time, augments Cdata1...Cdata6 holds the value of error.

**VB declare as below**

*Function ErrCounterReadDataAll(ByVal PortID As Short, ByVal SettingStr As String, ByVal DeviceID As Short, ByRef CData1 As String, ByRef CData2 As String, ByRef CData3 As String, ByRef CData4 As String, ByRef CData5 As String, ByRef CData6 As String, ByVal DigitalNo As Short, Optional ByVal LEDType As Short = Nothing) As String*

**Return Values**

If the function succeeds, it returns characters >>, and augments Cdata1..Cdata6 holds the value of error with following format

*[#impulses,error %,position]*

*For example:*

*"3,+0.0048,13" or ",14" as string*

If the function fails, it returns error information

**Parameters**

PortID: Standard RS232 comm port identifier. Comm1 = 1, Comm2 = 2,

etc.

SettingStr: “19200,n,8,2” (const)

DigitalNo: 2, 3, 4 – decimal places

LEDType: always 1, the 0 is only used on older types of display

CData1..Cdata6: it holds the value of error with following format

*[#impulses,error %,position]*

*For example:*

*"3,+0.0048,13" or ",14" as string*

APIs of Reference Standards

If Radian Standards are using, please refer to RRkit library of Radian software

* **K6D\_ReadWrite**

This function is for read instant measurement parameters from K6/K8 reference standard meter, see the function return

**VB declare as below**

*Function K6D\_ReadWrite(ByVal PortID As Short, ByVal StdMod As Short, ByVal Order As String, ByVal ABC As String, ByVal Op As String, Optional ByRef V As Object = Nothing, Optional ByRef C As Object = Nothing, Optional ByRef SettingStr As String = "19200,n,8,2") As String*

**Return Values**

If the function succeeds, it returns characters that contains measurement parameters; if the functions fails, it return empty string

**Sample return:**

U(V) DCU(V) I(A) DCI(A) P(W) Q(var) COS PHASE UPHASE A:239.980 0.02164 20.0144 -0.0024 2403.62 4158.39 0.50043 59.9713 0.00000B:240.050 0.02236 19.9967 -0.0021 2398.65 4157.97 0.49969 60.0201 119.995C:240.041 0.02231 20.0087 -0.0040 2401.80 4159.26 0.50007 59.9953 239.944FEQ=49.999 PSUM=7204.08 QSUM=12475.6 SSUM=14406.2 COS"

**Parameters**

PortID: serial port number in us

StdMod: always = 0

Order: always = “D”

ABC: always = “”

Op: always = “”

V: harmonic V

C: harmonic of C

SettingStr: “19200,n,8,1”