# IP Power Center

## SDK User Manual

IP Power Center SDK
Version: 1.0
Jan. 05, 2009

# IP Power Center

**IMPORTANT NOTICE**

1. This SDK helps you to add IP Power Device on the list

# 1.0 Introduction

The IP Power Center SDK was created to help the user in developing software or programs that would be able to control the IP Power devices. The devices that are compatible with the IP Power Center are:
**9211 IP Sensor**
**9222 IP Sensor**
**9212 Delux**
**9258 -** Note: Only two units can be controlled for this version of the IP Power Center
**9258 with Ping**
**9280 IP Boot Manager**

# 2.0 Installation

Before using the IP Power Center SDK, you will need to have install
**Microsoft NET. Framework Version 2.0** and above.

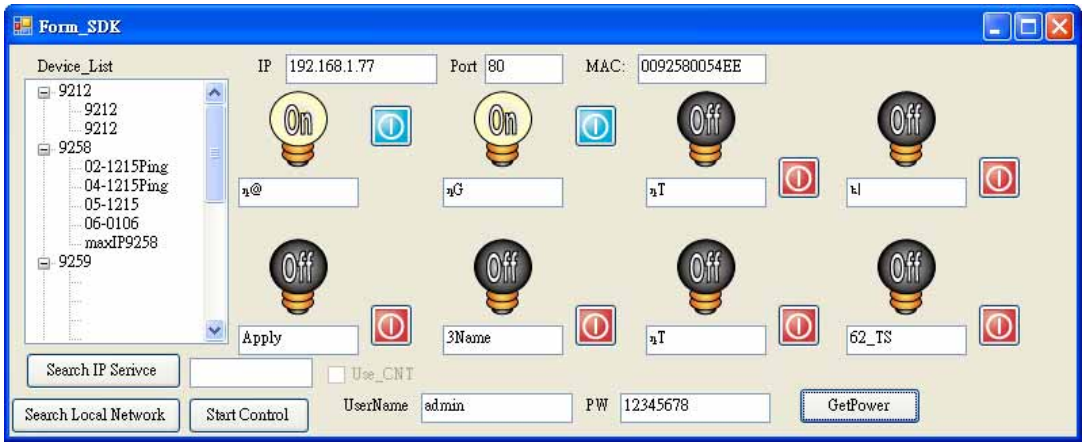
Please download the program from Microsoft's Website

# 3.0 Using the IP Power Center SDK

## *3.1 Interface Description*

The IP Power Center consists of a library called the "IP Power Center Library"

## *3.2 Form_SDK*




The picture below of the Form_SDK is a basic demo version for you to control the devices. There are 3 buttons on the button of the form that can be used.

1. **Search IP Service**: This allows you to search for your device on IP Edit
2. **Search Local Network**: Allows you to search for a device within the local area network
3. **Start Control**: Hit start control to control the device that you have found

### *3.3 Control your device on the local network*

**Start to get a Device Class**

Black Ground: It`s Code

Press “Search Local Network”

```
m_UDP.SearchLocalNetDevice()
```

You would find your device on the list

```
    Sub HandleEventAck_udp(ByVal deviceInfo As
C_UDPClient.LocalNetSearchResult) Handles m_UDP.RriseGetDevice
        InvokeTreeView(deviceInfo)
    End Sub
```

<-You have to add sub to get deviceinfo(which is a Class to record device info like IP,MAC..) form C_UDPClient (which is used to conect device form use udp socket)

Then you can add each “deviceinfo” to your list.

1. then select your device

```
    Private Sub TreeView_AfterSelect(ByVal sender As System.Object,
ByVal e As System.Windows.Forms.TreeViewEventArgs) Handles
TreeView.AfterSelect
        Me.TextIp.Text = CType(TreeView.SelectedNode,
SearchDeviceNode).m_deviceinfo.m_Ip
        Me.Textport.Text = CType(TreeView.SelectedNode,
SearchDeviceNode).m_deviceinfo.m_Ip_Port
        Me.TextMAC.Text = CType(TreeView.SelectedNode,
SearchDeviceNode).m_deviceinfo.m_Mac

        Me.Use_CNT.Enabled = CType(TreeView.SelectedNode,
SearchDeviceNode).m_deviceinfo.m_SupportCNT
        Me.Use_CNT.Checked = CType(TreeView.SelectedNode,
SearchDeviceNode).m_deviceinfo.m_CNT
            End Sub
```

2. and press “Start Control”

```
   Sub StartControl()

        Try
            CType(TreeView.SelectedNode,
```

```
SearchDeviceNode).m_deviceinfo.m_Username = Me.TextUser.Text
```
<-if you want to control your device well , you have to give Device UserName and Password correctly.

```
            CType(TreeView.SelectedNode,
SearchDeviceNode).m_deviceinfo.m_PW = Me.TextPw.Text
```

```
            Dim device As DeviceBase
```
<-DeviceBase is a “Class” for device ,you can get it form correctly deviceinfo

```
            device = DeviceBase.GetDevice(CType(TreeView.SelectedNode,
SearchDeviceNode).m_deviceinfo)
            device.m_CNT = Me.Use_CNT.Checked
            m_control = ControlCommand.GetControl(device)
```
<-m_Control is a “Class” for Control IP Power device , when you get a devicebase ,you should give it to control ‘class’ t get a control class.

```
            m_control.GetPowerStateWasPressed()
```
<-finally you can make a command to Device

```
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try

    End Sub
```

3. after a moment

```
    Sub HandleEventAck(ByVal device As DeviceInfo) Handles
g_control.RriseGetDataAndDoSomeThing
        InvokeInfo()
    End Sub
```
<- if Control ‘Class’ find something change, it would send a deviceinfo back , then you could reset state again.

## Make a On/Off Command

4. Press “On/Off”

```
            g_control.offButtonWasPressed(targetSendingPowerNum)
```
should make a On/Off Command form control ‘class’

```
            g_control.GetPowerStateWasPressed()
```
should make a GetPowerStateWasPressed Command again to get correct state.

### *3.4 How to Use IP Service And CNT*

**Start to get a Device (Support CNT)**

1. Input 'Search Word' in the text
2. Press "Search IP Service

```
    Private Sub ButtonIP_Service_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ButtonIP_Service.Click
        Me.TreeView.Nodes.Clear()
        If Me.TextBoxSearchIP_Service.Text.Length < 4 Then
            MsgBox("must more than 3 word")
            Exit Sub
        End If

        g_TCP.SearchDevice(Me.TextBoxSearchIP_Service.Text)

    End Sub<-You have to add sub to get deviceinfo(which is a Class to
record device info like IP,MAC..) form TCPClient(which is used to conect
device form use tcp socket)
```

Then you can add each "deviceinfo" to your list.

```
  Sub HandleEventAck_tcp(ByVal deviceInfo As DeviceInfo) Handles
g_TCP.RriseGetDevice
        InvokeTreeView(deviceInfo)
    End Sub

    Sub HandleEventAckCNT(ByVal deviceInfo As DeviceInfo) Handles
g_TCP.RriseCheckDeviceOnCNTServer

        InvokeSetSupportTreeView(deviceInfo)
        'Set support cnt
  End Sub <-HandleEventAckCnt it would reset deviceinfo of list node
```

<-

3. If you want to use CNT then you should check the CNT checkbox
4. And press "Start Control"
5. then Control Class 'would use CNT way to control your device