

---

# HUNTRON WORKSTATION REMOTE CONTROL

---

## Contents

Overview.....2

Huntron Remote Sample Application.....4

    Setup Tab.....4

    Scan Tab.....6

    Prober Tab .....8

    Align Tab ..... 10

Huntron Remote Pin Down Sample Application ..... 12

Huntron Remote One PC LabVIEW Sample Application..... 14

Huntron Remote Two PC LabVIEW Sample Application..... 18

Huntron Remote Test Stand Sample Application..... 19

Huntron Client DLL Function List ..... 20

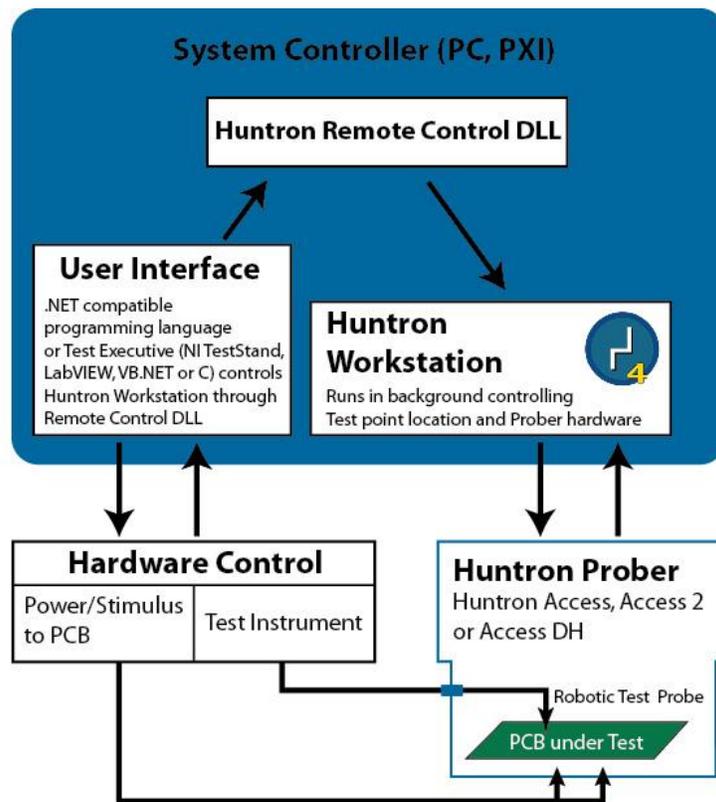
# Overview

The Huntron Workstation optional Remote Control feature allows control of the software from other programs. Its main purpose is to allow other programs to control Huntron hardware products through Huntron Workstation and control other software or instruments. Tests are created and verified in Huntron Workstation and then “controlled” by other programs.

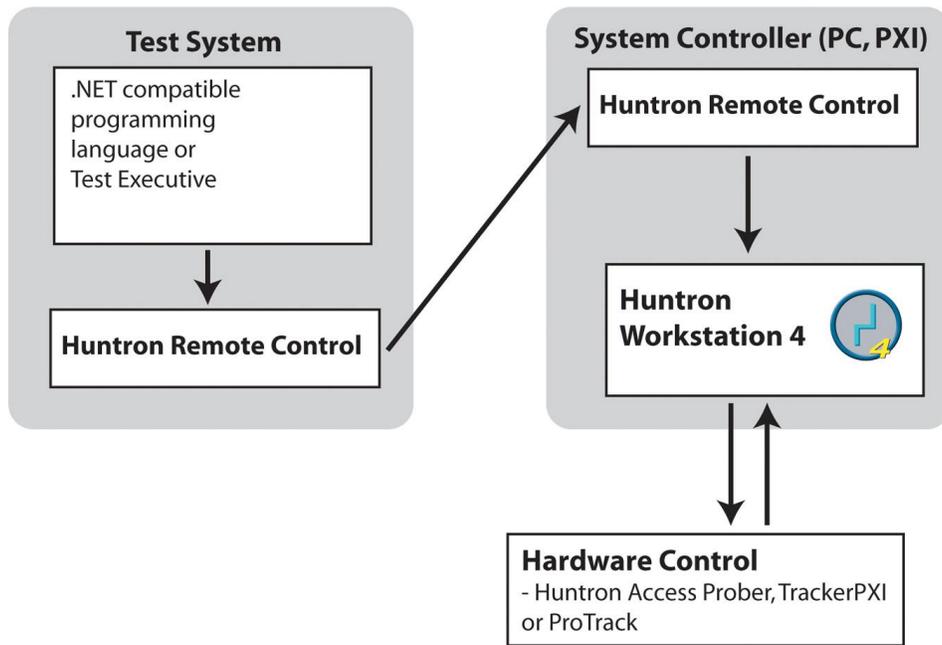
Other features include controlling a Prober to put the pin down on a pin to take a measurement with another tester, capturing a camera image over a selected pin and retrieving scan and reference signature data for the selected pin.

The process of adding Huntron Tracker and Prober capabilities to other programs and testers using drivers requires a lot of programming by the customer. Huntron Workstation Remote Control makes this a lot easier by providing most of the functionality needed with minimal programming.

Remote Control is used by programs talking to the Huntron Client VB.NET DLL (HuntronClient.dll). The DLL exposes functions that are called by the programs to control Huntron Workstation. The DLL communicates to Workstation through Microsoft Remoting Technologies.



Remote Control on a single Controller

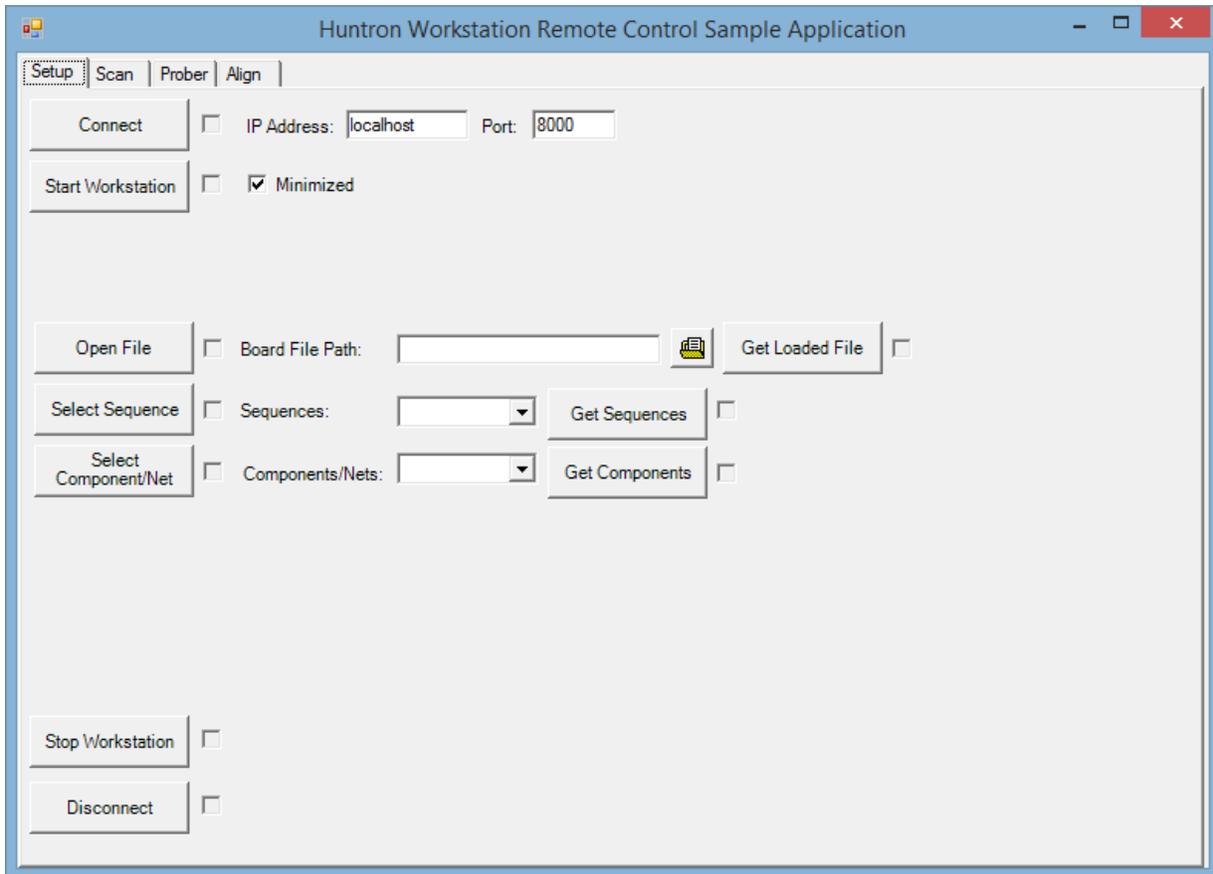


**Remote Control through a separate controller**

# Huntron Remote Sample Application

Below are screens captures of the Huntron Remote sample program and a detailed listing of the button functions used. Refer to the end of this document for a detailed description of the functions (Huntron Client DLL Function List) utilized by the application.

## Setup Tab



**Huntron Remote Control Sample VB.Net Program Setup Tab (HuntronRemoteApp.exe)**

### Connect Button

This button creates a connection to Microsoft Remoting technologies. It uses the IP Address “localhost” to connect to the same PC or an “IP address” of a different PC running Huntron Workstation. It also uses Port as the Microsoft Remoting technologies communication port. The box next to the button is checked automatically if the operation is completed successfully.

### **Start Workstation Button**

This button starts Huntron Workstation in Remote mode using a shell call. The Minimized checkbox hides Huntron Workstation after it starts. This is the preferred way to start Huntron Workstation. Starting with Minimized unchecked allows Huntron Workstation to be seen, but any interaction with Workstation can make the Remote Session invalid and cause issues. The box next to the button is checked automatically if the operation is completed successfully.

### **Open File Button**

This button causes Workstation to open the file in the Board File Path text box. The button next to the text control allows browsing for the file path. The box next to the button is checked automatically if the operation is completed successfully.

### **Get Loaded File Button**

This button causes Workstation to return the file path of the file currently loaded in Workstation. The file path is put in the text box to the left. The box next to the button is checked automatically if the operation is completed successfully.

### **Select Sequence Button**

This button causes Workstation to select the Sequence that is selected in the combo box to the right of the button. The box next to the button is checked automatically if the operation is completed successfully.

### **Get Sequences Button**

This button causes Workstation to return a list of the Sequence Names of the board loaded into Workstation. The Sequence names are loaded into the combo box next to the button. The box next to the button is checked automatically if the operation is completed successfully.

### **Select Component/Net Button**

This button causes Workstation to select the Component or Net that is selected in the combo box to the right of the button. The box next to the button is checked automatically if the operation is completed successfully.

### **Get Components Button**

This button causes Workstation to return a list of the Component Names of the selected Sequence in Workstation. The Component or Net Names are loaded into the combo box next to the button. The box next to the button is checked automatically if the operation is completed successfully.

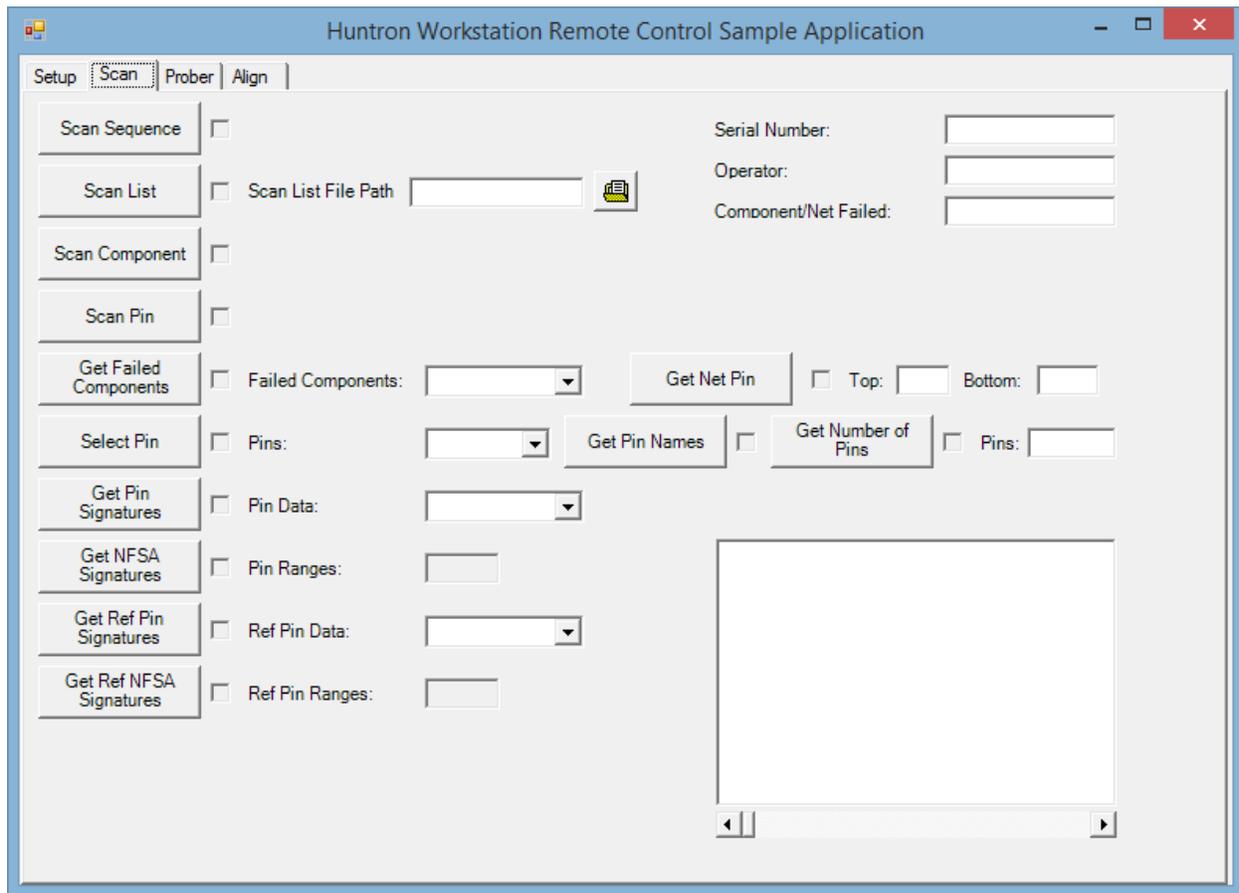
### **Stop Workstation Button**

This button causes Workstation to close. The box next to the button is checked automatically if the operation is completed successfully.

### **Disconnect Button**

This button causes the connection to Microsoft Remoting Technologies to be closed. The box next to the button is checked automatically if the operation is completed successfully.

# Scan Tab



**Huntron Remote Control Sample VB.Net Program Scan Tab (HuntronRemoteApp.exe)**

## Scan Sequence Button

This button causes Workstation to perform a scan of the selected sequence. It uses the values in the Serial Number and Operator text boxes. It fills the Component/Net Failed text box with the number of components that failed the scan. The box next to the button is checked automatically if the operation is completed successfully.

## Scan List Button

This button causes Workstation to perform a scan of the Scan List specified in the Scan List File Path text box. The button next to the text box allows browsing for the file path. It uses the values in the Serial Number and Operator text boxes. It fills the Component/Net Failed text box with the number of components that failed the scan. The box next to the button is checked automatically if the operation is completed successfully.

### **Scan Component Button**

This button causes Workstation to perform a scan of the selected component or net. It uses the values in the Serial Number and Operator text boxes. It fills the Component/Net Failed text box with the number of components that failed the scan. The box next to the button is checked automatically if the operation is completed successfully.

### **Scan Pin Button**

This button causes Workstation to capture signatures for all the ranges of the pin. The box next to the button is checked if the operation is completed successfully.

### **Get Failed Components Button**

This button causes Workstation to return a list of the component/net names that failed the last scan. The combo box next to the button is filled with the names. The box next to the button is checked automatically if the operation is completed successfully.

### **Get Net Pin Button**

This button causes Workstation to return the top side and bottom side pins of the net. They are put in the Top and Bottom text boxes. The box next to the button is checked automatically if the operation is completed successfully.

### **Select Pin Button**

This button causes Workstation to select the Pin specified in the combo box next to the button. The box next to the button is checked automatically if the operation is completed successfully.

### **Get Pin Signatures Button**

This button causes Workstation to return the range signatures for the selected pin from the last scan. The data of the signatures is loaded into the combo box next to the button. The picture box displays the signature from the first range. Other range signatures can be displayed by scrolling the scroll bar under the picture box. The box next to the button is checked automatically if the operation is completed successfully.

### **Get NFSA Signatures Button**

Not supported at this time.

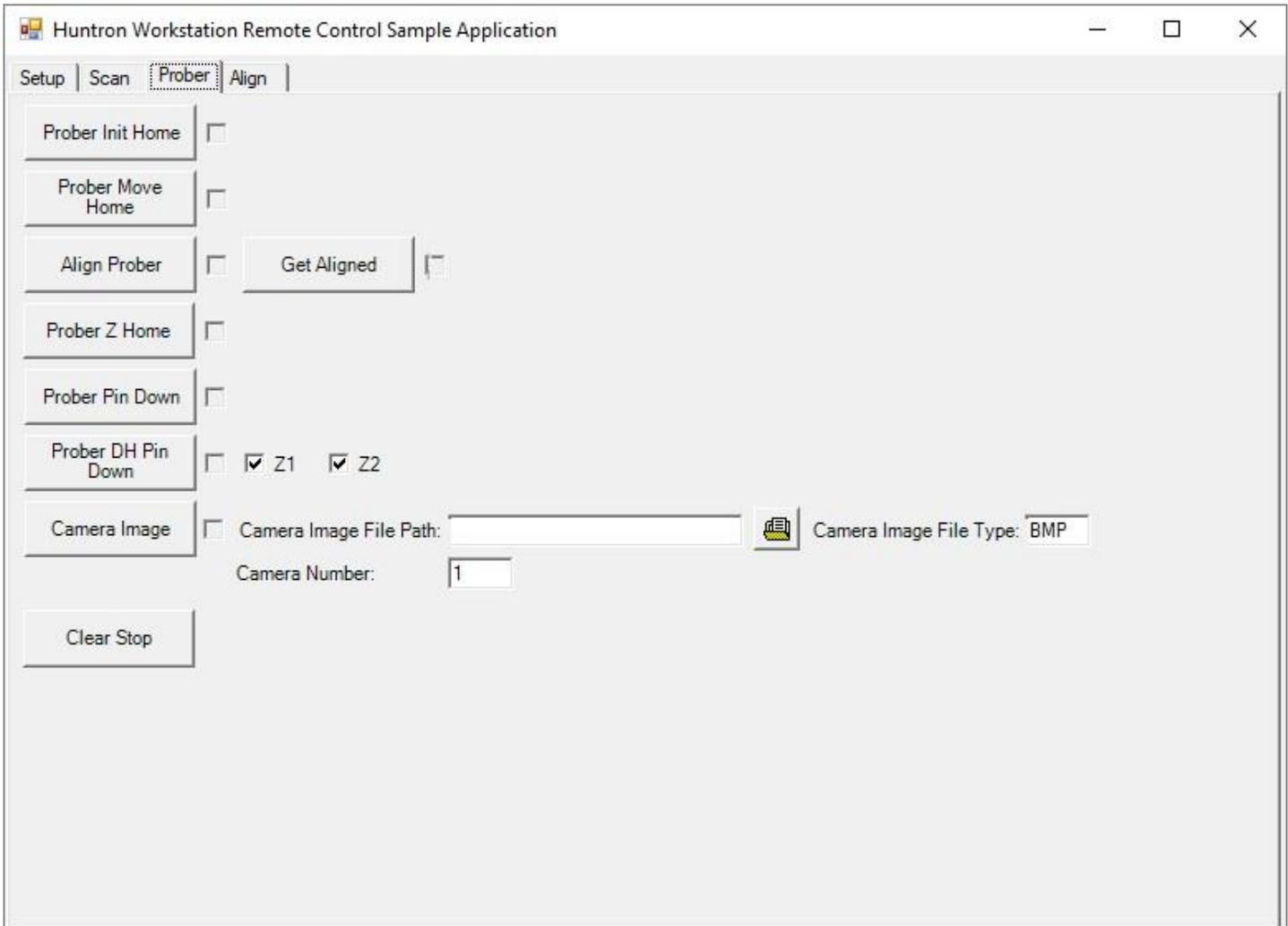
### **Get Ref Pin Signatures Button**

This button causes Workstation to return the range reference signatures for the selected pin from the last scan. The data of the signatures is loaded in to the combo box next to the button. The picture box displays the reference signature from the first range. Other range reference signatures can be displayed by scrolling the scroll bar under the picture box. The box next to the button is checked automatically if the operation is completed successfully.

### **Get Ref NFSA Signatures Button**

Not supported at this time.

# Prober Tab



**Huntron Remote Control Sample VB.Net Program Prober Tab (HuntronRemoteApp.exe)**

## **Prober Init Home Button**

This button causes Workstation to initialize the probe(s) to the home position. The box next to the button is checked automatically if the operation is completed successfully.

## **Prober Move Home Button**

This button causes Workstation to move the probe(s) to the home position. The box next to the button is checked automatically if the operation is completed successfully.

## **Align Prober Button**

This button causes Workstation to perform an Auto Align. The box next to the button is checked automatically if the operation is completed successfully.

### **Get Aligned Button**

This button causes Workstation to return whether or not the board is aligned. The box next to the button is checked automatically if the operation is completed successfully.

### **Prober Z Home Button**

This button causes Workstation to move the probe(s) to the up home position. The box next to the button is checked automatically if the operation is completed successfully.

### **Prober Pin Down Button**

This button causes Workstation to put the probe tip down on the selected pin. This is for single head probers. The box next to the button is checked automatically if the operation is completed successfully.

### **Prober DH Pin Down Button**

This button causes Workstation to put the probe(s) tip down on the selected pin. The Z1 and Z2 check boxes select which probe will be moved. This is for the Access DH. The box next to the button is checked automatically if the operation is completed successfully.

### **Prober DH Setup Button**

This button causes Workstation to control the internal switching relays of the Access DH. The Internal Tracker check box selects the Internal Tracker of the External Tracker BNCs. The Chanel A check box selects the Tracker Channel A or B to be used. The Z1 Signal check box connects the Tracker Channel to Z1 and Common to Z2 or the other way around. The box next to the button is checked automatically if the operation is completed successfully.

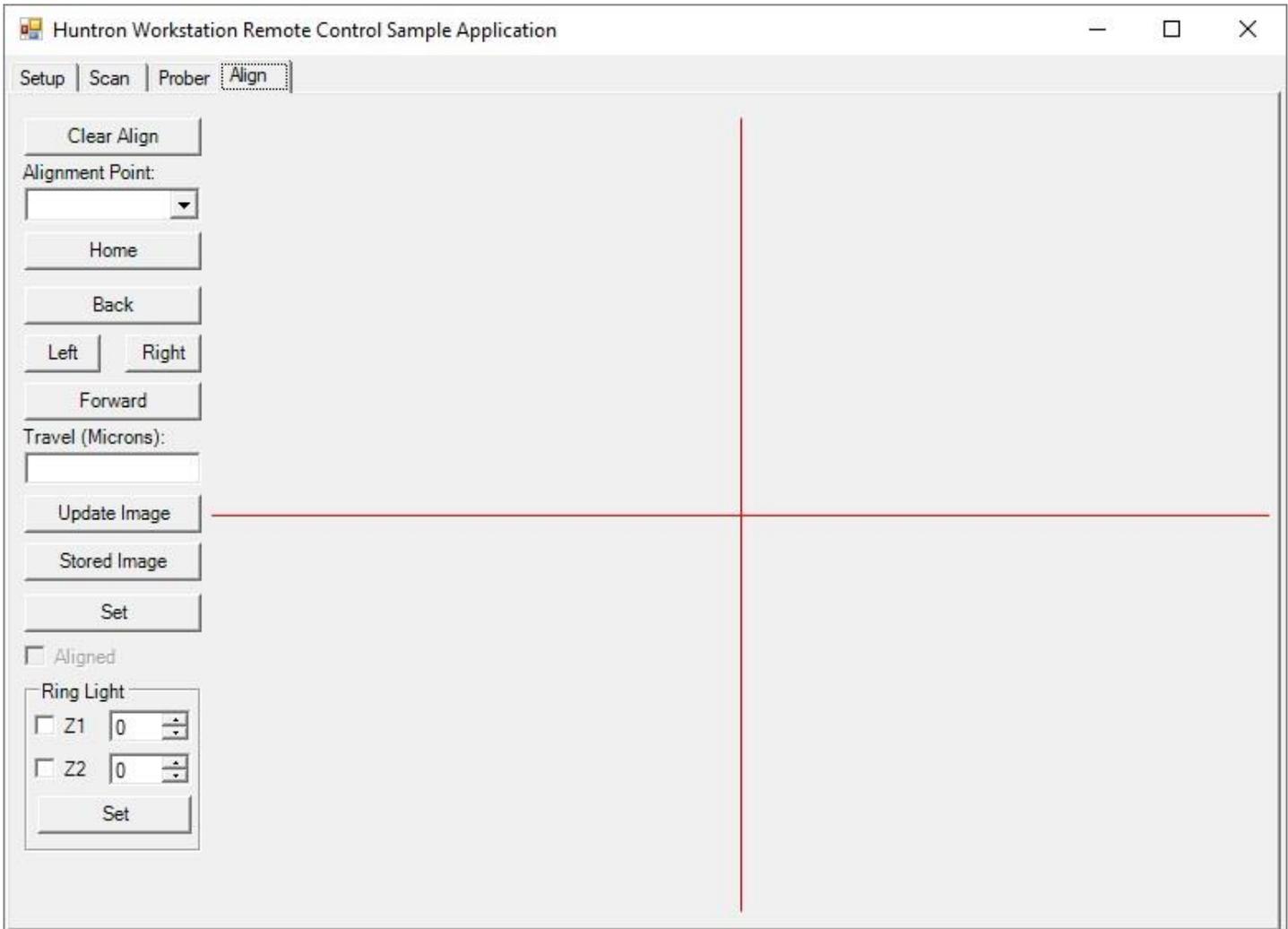
### **Camera Image Button**

This button causes Workstation to move the camera specified in the Camera number text box over the selected pin and return an image of the type specified in the Camera Image File Type text box. Valid entries are BMP, JPG or PNG. The image will be written to the file path specified in the Camera Image File Path text box. The button next to the text box allows browsing for the file path. The box next to the button is checked automatically if the operation is completed successfully.

### **Clear Stop Button**

This button causes Workstation to clear the stop button being pressed. The box next to the button is checked automatically if the operation is completed successfully.

# Align Tab



## Huntron Remote Control Sample VB.Net Program Align Tab (HuntronRemoteApp.exe)

### Clear Align Button

This button causes Workstation to clear the current alignment of the board.

### Home Button

This button causes Workstation to move the camera home based on the selection in the Alignment Point combo box.

### Back Button

This button causes Workstation to move the camera towards the back of the prober the number of microns specified in the Travel (microns) text box based on the selection in the Alignment Point combo box.

### Left Button

This button causes Workstation to move the camera towards the left side of the prober the number of microns specified in the Travel (microns) text box based on the selection in the Alignment Point combo box.

### **Right Button**

This button causes Workstation to move the camera towards the right side of the prober the number of microns specified in the Travel (microns) text box based on the selection in the Alignment Point combo box.

### **Forward Button**

This button causes Workstation to move the camera towards the front of the prober the number of microns specified in the Travel (microns) text box based on the selection in the Alignment Point combo box.

### **Update Image Button**

This button causes the Workstation to return a camera image of the current camera location and display it in the picture box.

### **Stored Image Button**

This button displays the stored image of the selected alignment in the picture box. The image was returned from Workstation when the Alignment Point was selected.

### **Set Button**

This button causes Workstation to set the selected Alignment Point to the current position of the camera. If Alignment Point 1 and Alignment Point 2 to have been set the Aligned check box will be checked.

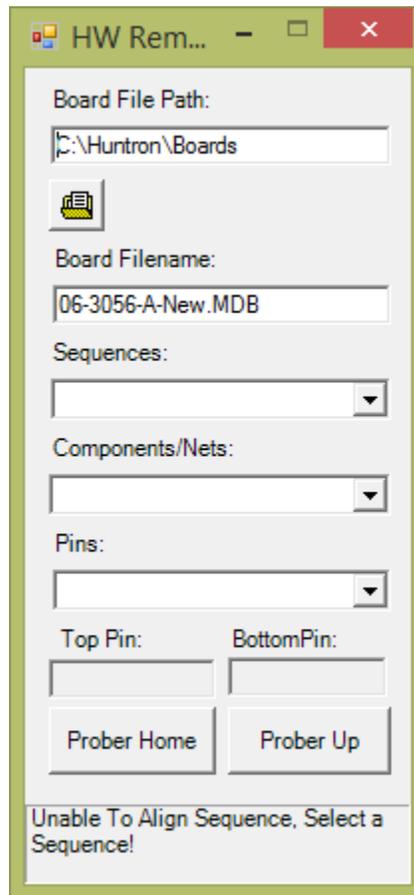
### **Ring Light**

This button causes Workstation to control the Prober Ring Lights. Checking the box turns on the light to the intensity set in the Up/Down scroll box.

### **More Information**

For more detail on what each button does, see the form1.vb in the "Program Files\Huntron Workstation\Huntron Remote App" folder of a Workstation installation.

# Huntron Remote Pin Down Sample Application



**Huntron Remote Control Pin Program (HuntronRemotePin.exe)**

## **File Folder Button**

This button allows browsing to and Workstation MDB file and causes Workstation to load the file. The Board File Path and Board Filename text boxes are updated. This also fills the Sequences combo box.

## **Sequences Combo Box**

This button allows selecting a Sequence of the board and causes Workstation to select the Sequence. This also fills the Component/Nets combo box.

## **Components/Nets Combo Box**

This button allows selecting of a Component/Net of the Sequence and causes Workstation to select the Component/Net. This also fills the Pins combo box and updates the Top Pin and Bottom Pin text boxes.

**Pins Combo Box**

This button allows selecting a Pin of the Component/Net and causes Workstation to select the Pin and move the probe tip(s) down onto the pin of the board.

**Prober Home Button**

This button causes Workstation to move the Probe(s) to their home positions.

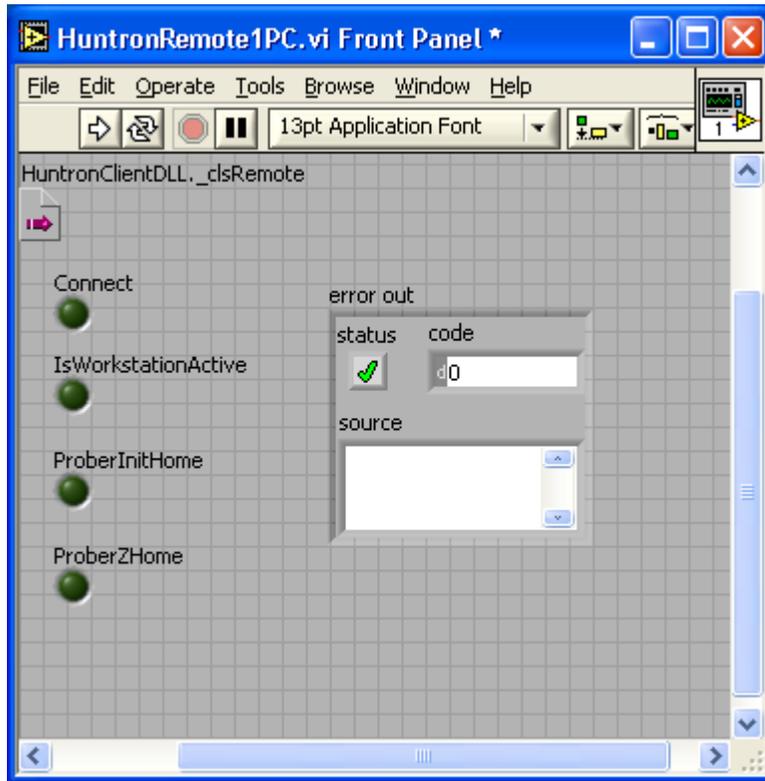
**Prober Up Button**

This button causes Workstation to move the probe(s) to their Pin up positions.

**More Information**

For more detail on what each button does, see the form1.vb in the "Program Files\Huntron Workstation\Huntron Remote Pin App" folder of a Workstation installation.

# Huntron Remote One PC LabVIEW Sample Application

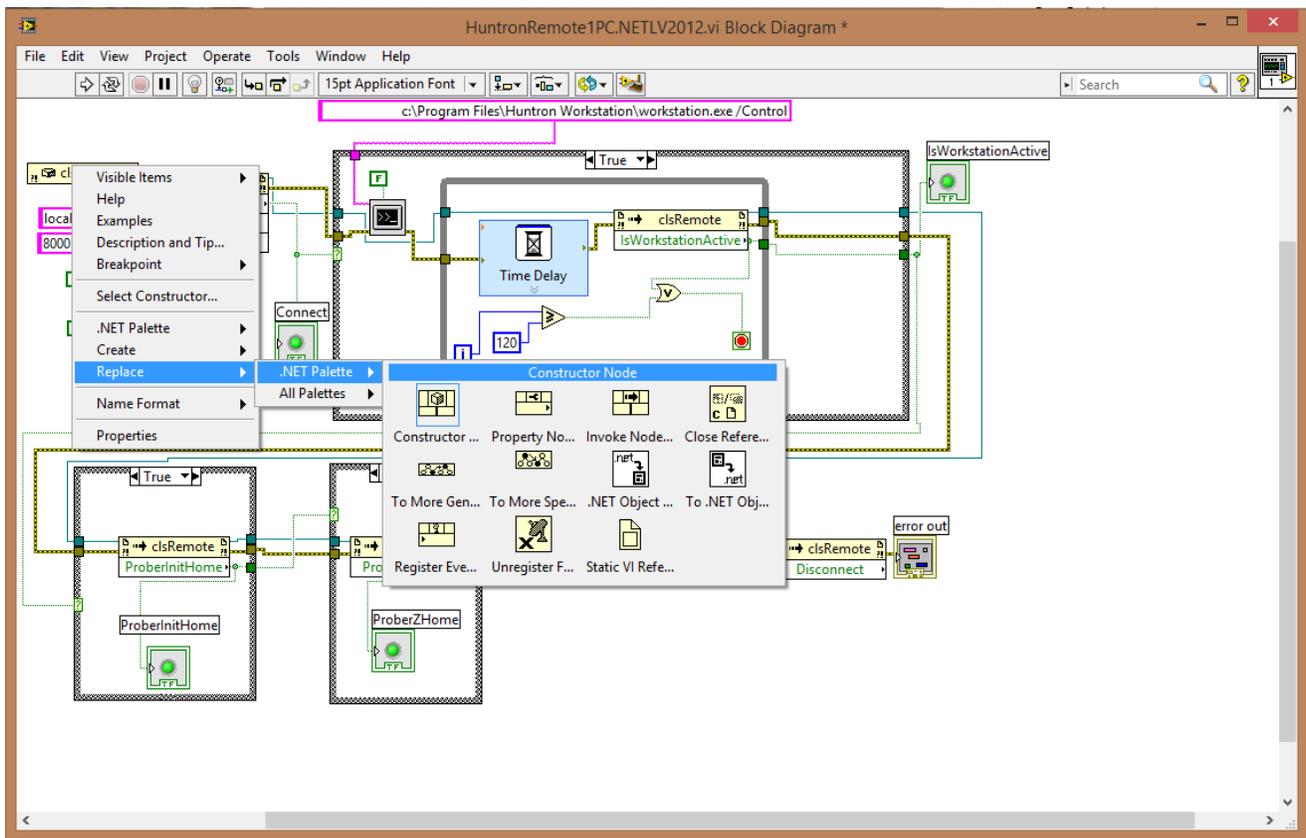


**Huntron Remote One PC Sample LabVIEW Program (HuntronRemote1PC.vi)**

## More Information

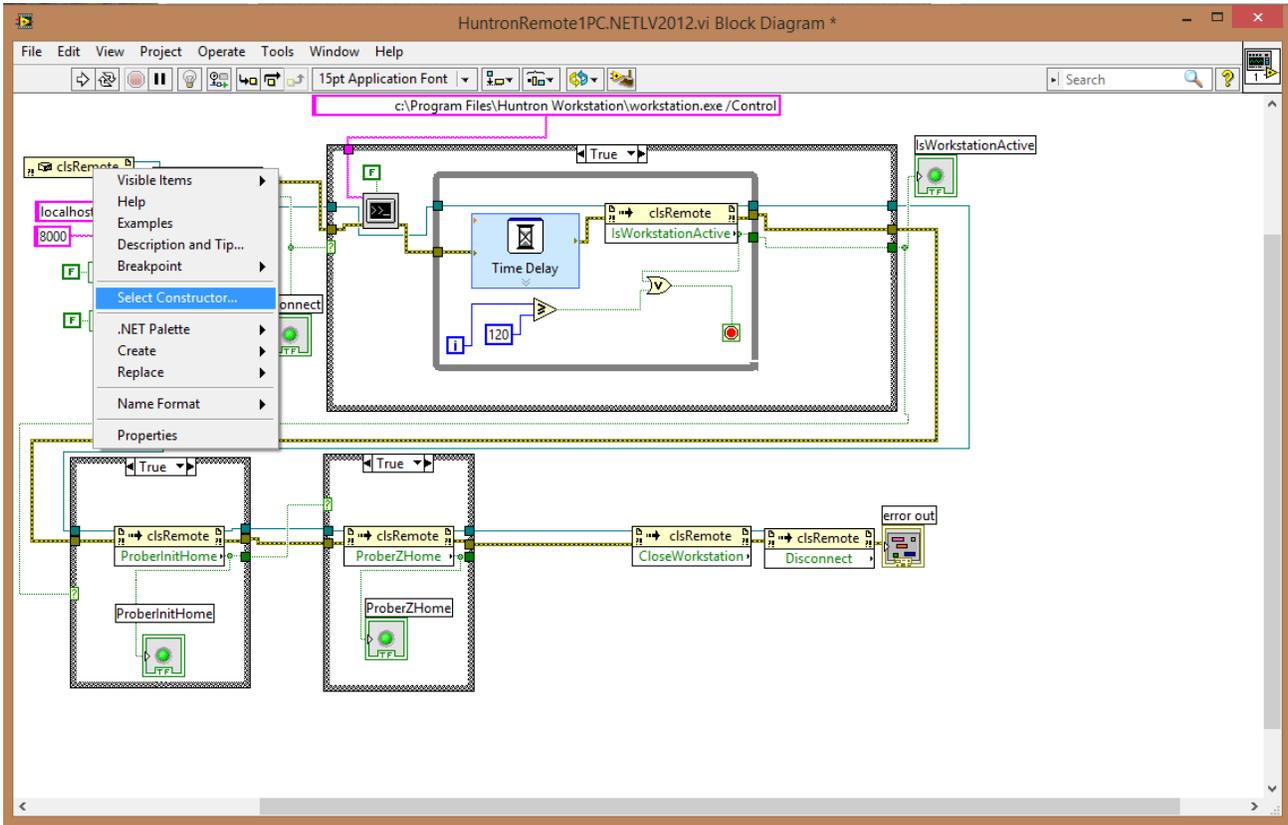
For more detail see the HuntronRemote1PC.vi file in the "Program Files\Huntron Workstation" folder of a Workstation installation.

You may need to update the constructor in the VI sample files. To accomplish this right click on the constructor in the block diagram.



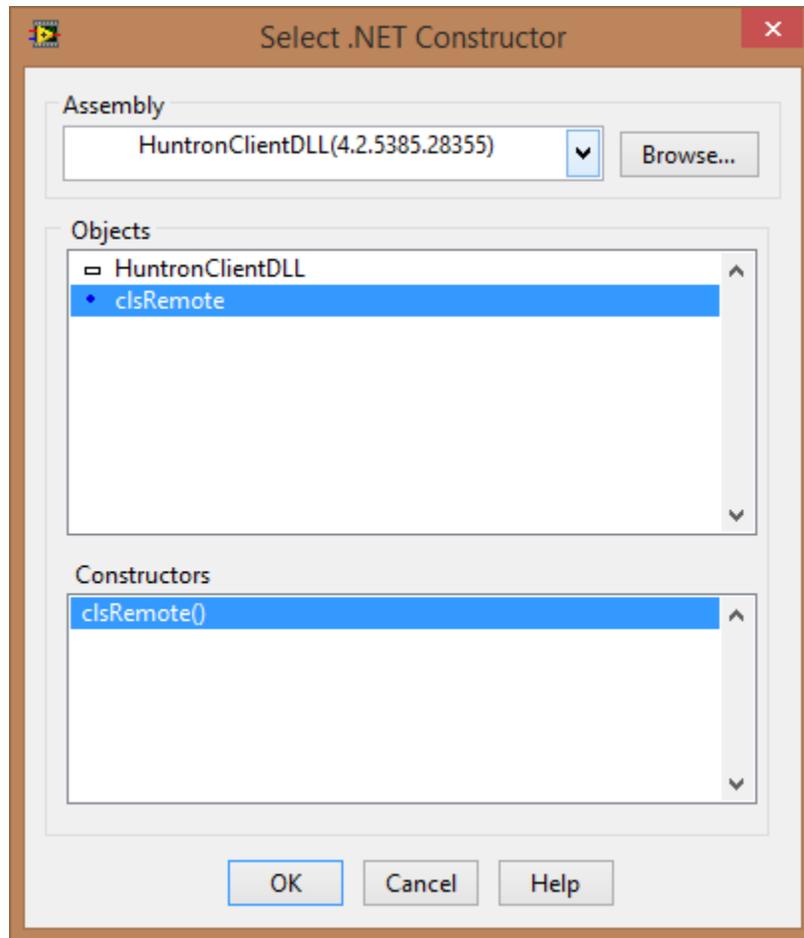
**LabVIEW Program Block Diagram (HuntronRemote1PC.vi)**

Select Replace then .NET Palatte and then Constructor.



**LabVIEW Program Select .NET Constructor (HuntronRemote1PC.vi)**

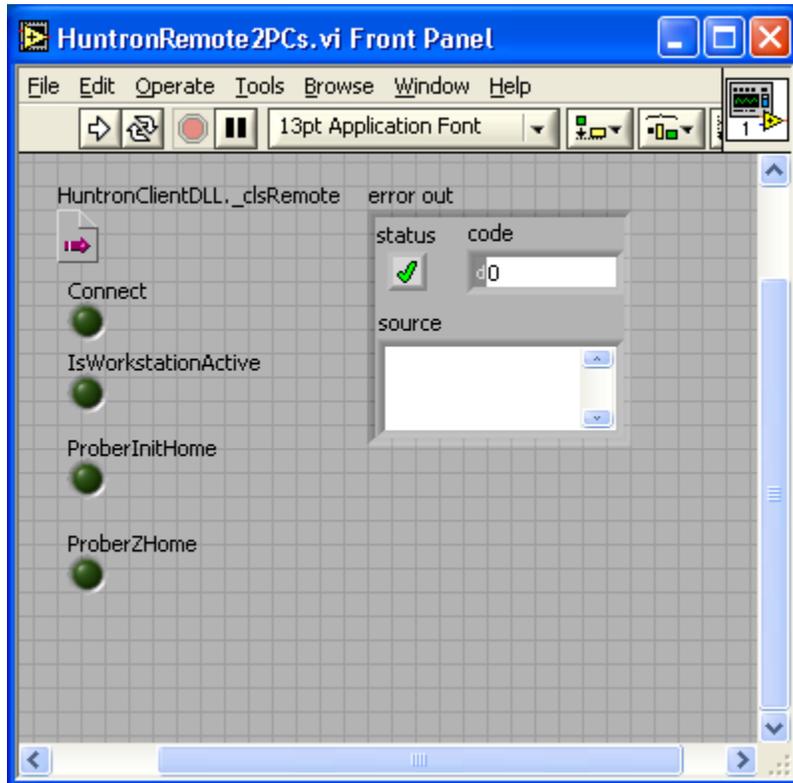
Then right click on the new constructor and click Select Constructor.



### LabVIEW Program Block Diagram (HuntronRemote1PC.vi)

The Browse to the HuntronClientDLL.dll file in the Program Files (x86)\Huntron Workstation folder. Then select the clsRemote() constructor under HuntronClientDLL.

# Huntron Remote Two PC LabVIEW Sample Application

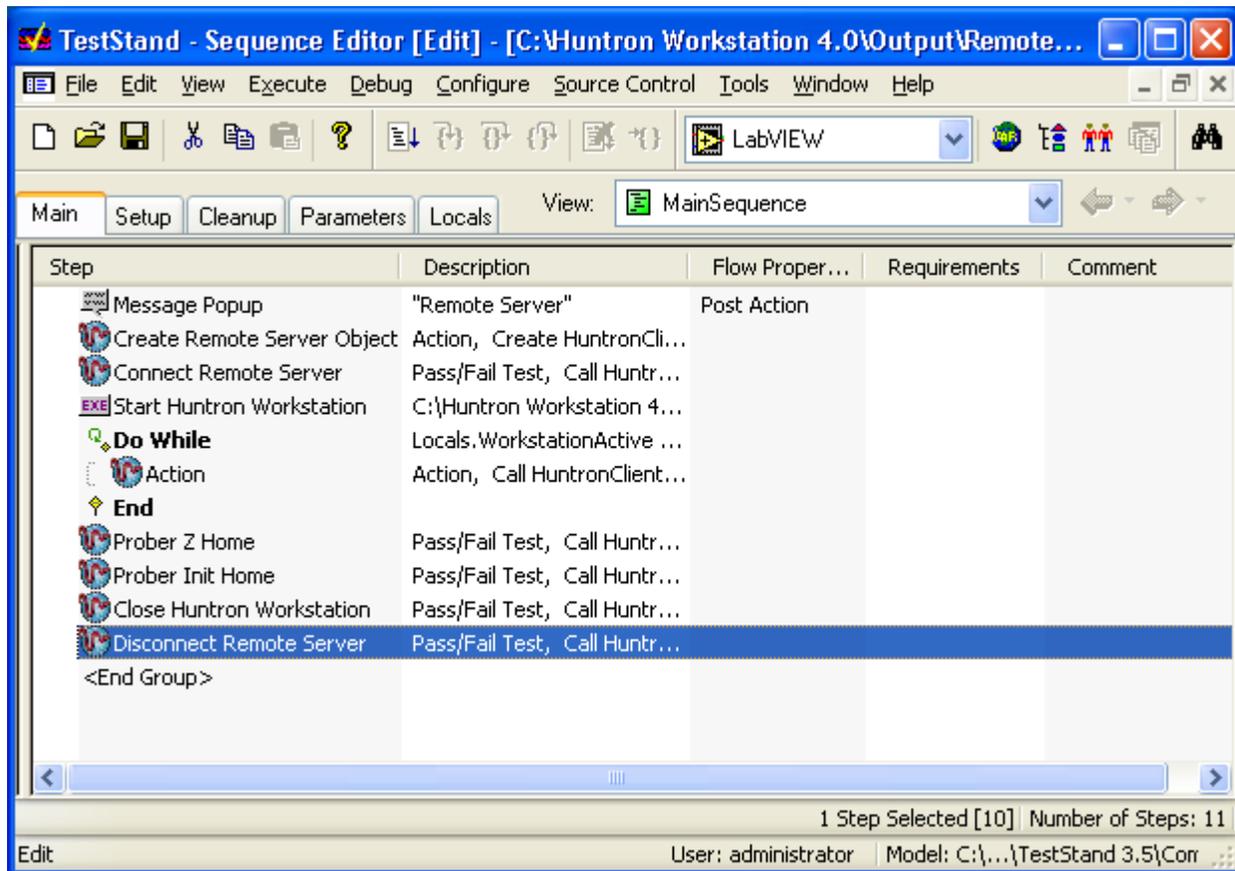


**Huntron Remote Two PC Sample LabVIEW Program (HuntronRemote2PC.vi)**

## More Information

For more detail see the HuntronRemote2PC.vi file in the "Program Files\Huntron Workstation" folder of a Workstation installation.

# Huntron Remote Test Stand Sample Application



**Huntron Remote Sample TestStand Program (RemoteControl.seq)**

## More Information

For more detail see the RemoteControl.seq1PC.vi file in the "Program Files\Huntron Workstation" folder of a Workstation installation.

# Huntron Client DLL Function List

## **Function CameraPicture(ByVal FilePath As String, ByVal FileType As String, ByVal CameraNumber as Integer) As Boolean**

Captures an image from the Prober camera at the location of the probe tip. FilePath should be the full file path of the location for the image file. FileType should be set to "BMP", "JPEG" or "PNG". Anything else defaults to "BMP". The FilePath filename extension should match the FileType (.bmp for "BMP", .jpg for "JPEG" and .png for "PNG"). The Camera number is always set to 1 unless accessing the Z2 camera on the Access DH, then it should be set to 2. Returns true if the function succeeds.

## **Function CloseWorkstation() As Boolean**

Closes Huntron Workstation. This should be done before disconnecting. Returns true if the function succeeds.

## **Function Connect(ByVal sIPAddress As String, ByVal sPort as String) As Boolean**

Connects to the Huntron Remote Server. To connect to the server on the same PC, sIPAddress should be set to "localhost". To connect to the server on another PC, sIPAddress should be set to the IP address of that PC. The sPort is a hex string of the server port number to be used (normally set to 8000). The Huntron Remote server runs on the PC where Huntron Workstation is running. Returns true if the function succeeds.

## **Function Disconnect() As Boolean**

Disconnects from the Huntron Remote. Server Returns true if the function succeeds.

## **Function GetAligned() As Boolean**

Gets the aligned status of the Prober connected to Huntron Workstation. Returns true if aligned.

## **Function GetComponentPins(ByRef iNumberOfPins as Integer) As Boolean**

Gets the number of pins for the currently selected component/net in Huntron Workstation as iNumberOfPins. Returns true if the function succeeds.

## **Function GetComponents(ByRef sComponents As String) As Boolean**

Gets the components/nets of the currently selected sequence in Huntron Workstation. sComponents is a tab delimited string of component/net names in OrderNumber order. Returns true if the function succeeds.

**Function GetExternalPinSignatures(ByRef daWaveforms() As Double, ByRef iaWaveformXSizes() As Integer, ByRef iaWaveformYSizes() As Integer, ByRef daWaveformXMins() As Double, ByRef daWaveformXMaxs() As Double, ByRef daWaveformYMins() As Double, ByRef daWaveformYMaxs() As Double, ByRef baWaveformOvertimes() As Boolean, ByRef daReadings() As Double, ByRef daReadings2() As Double, ByRef daReadings3() As Double, ByRef msalmages() As System.IO.MemoryStream, ByRef iRanges As Integer) As Boolean**

Gets the External Hardware reading, waveform and image data for all the ranges of the currently selected pin from the last scan. daWaveforms returns an array of doubles. Its length is equal to the value of iaWaveformXSizes + iaWaveformYSizes for each range. daWaveformXMins, daWaveformXMaxs, daWaveformYMins, daWaveformYMaxs are the min and max data values for each range. daReadings, daReadings2() and daReadings3 are the reading values for each range. msalmages are the image data for each range. iRanges is set to the number of ranges. This function is used for NFS remote control.

**Function GetExternalReferencePinSignatures(ByRef daWaveforms() As Double, ByRef iaWaveformXSizes() As Integer, ByRef iaWaveformYSizes() As Integer, ByRef daWaveformXMins() As Double, ByRef daWaveformXMaxs() As Double, ByRef daWaveformYMins() As Double, ByRef daWaveformYMaxs() As Double, ByRef baWaveformOvertimes() As Boolean, ByRef daReadings() As Double, ByRef daReadings2() As Double, ByRef daReadings3() As Double, ByRef msalmages() As System.IO.MemoryStream, ByRef iRanges As Integer) As Boolean**

Gets the External Hardware reading, waveform and image data for all the ranges of the currently selected pin from the last scan. daWaveforms returns an array of doubles. Its length is equal to the value of iaWaveformXSizes + iaWaveformYSizes for each range. daWaveformXMins, daWaveformXMaxs, daWaveformYMins, daWaveformYMaxs are the min and max data values for each range. daReadings, daReadings2() and daReadings3 are the reading values for each range. msalmages are the image data for each range. iRanges is set to the number of ranges. This function is used for NFS remote control.

**Function GetFailedComponents(ByRef sFailedComponents As String, ByRef daDeviations() As Double, ByRef daAreas() As Double) As Boolean**

Gets the failed components for the last scan in Huntron Workstation. sFailedComponents is a tab delimited string of component/net names in CompNetId order. daDeviations() returns an array of doubles containing the deviation for the component/net in the same order as the failed component/net names. daAreas() returns an array of doubles containing the area for the component/net in the same order as the failed component/net names. Returns true if the function succeeds.

**Function GetFileLoaded(ByRef sFilePath As String) As Boolean**

Gets the file path for the Huntron Workstation created MDB board database file loaded into Huntron Workstation as sFilePath. Returns true if the function succeeds.

**Function GetPinNames(ByRef sPinNames As String) As Boolean**

Gets the pin names for the current selected component in Huntron Workstation. sPinNames is a tab delimited string of pin names in PinNumber order. Returns true if the function succeeds.

### **Function GetNetPin(ByRef iTopPin As Integer, ByRef iBottomPin As Integer) As Boolean**

Gets the pin numbers for the current selected net in Huntron Workstation. iTopPin is the top pin setting and iBottomPin is the bottom pin setting. Returns true if the function succeeds.

### **Function GetPinSignatures(ByRef baSigData() As Byte, ByRef iRanges As Integer) As Boolean**

Gets the signature data for all the ranges of the currently selected pin from the last scan. baSigData returns an array of bytes. Its length is equal to the value of iRanges \* 200. iRanges is set to the number of ranges that there are signatures for in baSigData. The first 100 bytes of the signature data is the horizontal waveform and the second hundred bytes are the vertical waveform. To display a signature, plot the corresponding horizontal and vertical bytes in XY. Returns true if the function succeeds.

### **Function GetReferencePinSignatures(ByRef baSigData() As Byte, ByRef iRanges As Integer) As Boolean**

Gets the signature data for the reference signatures for all the ranges of the currently selected pin from the last scan. baSigData returns an array of bytes. Its length is equal to the value of iRanges \* 200. iRanges is set to the number of ranges that there are signatures for in baSigData. The first 100 bytes of the signature data is the horizontal waveform and the second hundred bytes are the vertical waveform. To display a signature, plot the corresponding horizontal and vertical bytes in XY. Returns true if the function succeeds.

### **Function GetSequences (ByRef sSequences As String) As Boolean**

Gets the sequences of the loaded board in Huntron Workstation. sSequences is a tab delimited string of sequence names in OrderNumber order. Returns true if the function succeeds.

### **Function IsWorkstationActive() As Boolean**

Checks to see if Huntron Workstation is running and available. Returns true if the function succeeds.

### **Function OpenFile(ByVal MDBPath As String) As Boolean**

Loads a Huntron Workstation created MDB board database file into Huntron Workstation. Set MDBPath to the full path name of the MDB file (i.e. "C:\Documents and Settings\username\My Documents\Huntron\Boards\Test.mdb"). Returns true if the function succeeds.

### **Function ProberAlign() As Boolean**

Uses the Huntron Workstation Auto Align feature to align the board in the Prober for scanning. Make sure that Auto Align performs successfully in Huntron Workstation before using it through Remote Control. Returns true if function succeeds.

### **Function ProberInitHome() As Boolean**

Moves the Prober probe tip all the way up and to the back right corner. Returns true if the function succeeds.

### **Function ProberPinDown() As Boolean**

Moves the Prober probe tip down on to the selected pin. Returns true if the function succeeds.

### **Function DHPinDown(ByVal bZ1 as Boolean, ByVal bZ2 as Boolean) As Boolean**

Moves the Prober probe tip down on to the selected pin. Z1 should be set to True to move the Z1 probe down. Z2 should be set to True to move the Z2 probe down. Returns true if the function succeeds.

### **Function DHSetup(ByVal bInternalTracker as Boolean, ByVal bChannelA as Boolean, ByVal bZ1Signal as Boolean) As Boolean**

Moves the Prober probe tip down on to the selected pin. bInternalTracker should be set to True to Use the Internal Tracker and False to use the External Tracker BNCs. bChannelA should be set to True to use Tracker Channel A and False to use Tracker Channel B. bZiSignal should be set to True to connect the Tracker Channel to Z1 and tracker Common to Z2; Set To False to connect Tracker Channel to Z2 and Tracker Common to Z1. Returns true if the function succeeds.

### **Function ProberZHome() As Boolean**

Moves the Prober probe tip all the way up. Returns true if the function succeeds.

### **Function ScanComponent(ByVal SerialNumber As String, ByVal Operator As String, ByRef iFailed As Integer) As Boolean**

Scans the selected Component/Net in Huntron Workstation. SerialNumber should be set to the serial number of the board being scanned. Operator should be set the name of the operator running the test. iFailed will be set to the number of components that failed during the scan. Returns true if the function succeeds.

### **Function ScanList(ByVal SerialNumber As String, ByVal Operator As String, ByVal ListPath As String, ByRef iFailed As Integer) As Boolean**

Scans the selected sequence using the provided scan list in Huntron Workstation. SerialNumber should be set to the serial number of the board being scanned. Operator should be set the name of the operator running the test. ListPath should be set to the full path of the scan list file to be used (i.e. "C:\Documents and Settings\username\My Documents\Huntron\Lists\Test.lst"). iFailed will be set to the number of components that failed during the scan. Returns true if the function succeeds.

**Function ScanPin(ByVal SerialNumber As String, ByRef iFailed As Integer, ByRef baSigData() as Byte, ByRef iRanges as Integer, ByRef baRefSigData() as Byte) As Boolean**

Scans the selected Pin in Huntron Workstation. SerialNumber should be set to the serial number of the board being scanned. iFailed will be set to the number of components that failed during the scan. iRanges will be set to the number of ranges that there are signatures for in baSigData. A scan is not created in Huntron Workstation. Returns true if the function succeeds.

**Function ScanSequence(ByVal SerialNumber As String, ByVal Operator As String, ByRef iFailed As Integer) As Boolean**

Scans the selected sequence in Huntron Workstation. SerialNumber should be set to the serial number of the board being scanned. Operator should be set the name of the operator running the test. iFailed will be set to the number of components that failed during the scan. Returns true if the function succeeds.

**Function SelectComponentNet(ByVal ComponentNetName As String) As Boolean**

Selects a Component or Net of the selected sequence. ComponentNetName should match one of the Component or Net names on the Component/Net grid in Huntron Workstation. Returns true if the function succeeds.

**Function SelectPin(ByVal PinNumber As String) As Boolean**

Selects a Pin of the selected component or net. PinNumber should match one of the Pin numbers on the Pin grid in Huntron Workstation. Returns true if the function succeeds.

**Function SelectSequence(ByVal SequenceName As String) As Boolean**

Selects a sequence of the loaded board. SequenceName should match one of the sequence names on the Sequences grid in Huntron Workstation. Returns true if the function succeeds.

**Function CurrentCameraPicture(ByVal FilePath As String, ByVal FileType As String, ByVal CameraNumber As Integer) As Boolean**

Writes a camera image of the current location to the file of the type and path specified. The FilePath filename extension should match the FileType (.bmp for "BMP", .jpg for "JPEG" and .png for "PNG"). Returns true is the functions succeeds.

**Function GetStoredAlignPoint(ByVal iPoint As Integer, ByRef dX As Double, ByRef dY As Double, ByVal ImageFilePath As String) As Boolean**

Returns the X and Y coordinates of the the selected Alignment Point. iPoint = 1 is Z2 Camera, iPoint = 2 is Z1 Camera, iPoint = 3 is Alignment Point 1, iPoint = 4 is alignment Point 2. Writes a stored image BMP file of the alignment point to the path specified. Returns true is the functions succeeds.

**Function MoveXYPoint(ByVal dX As Double, ByVal dY As Double, ByVal bZ1 As Boolean) As Boolean**

Moves the specified camera to the XY position specified in microns. bZ1 true selects the single head prober camera or Z1 Camera of the Access DH. bZ1 false selects the Access DH Z2 Camera. Returns true if the function succeeds.

**Function MoveXYDistance(ByVal dX As Double, ByVal dY As Double, ByVal bZ1 As Boolean) As Boolean**

Moves the selected camera the XY distances specified. Back is negative Y, Forward is positive Y, Left is positive X and Right is negative X. bZ1 true selects the single head prober camera or Z1 Camera of the Access DH. bZ1 false selects the Access DH Z2 Camera. Returns true if the function succeeds.

**Function SetAlign(ByVal iPoint As Integer) As Boolean**

Sets the X and Y coordinates of the selected Alignment Point. iPoint = 1 is Z1 Camera, iPoint = 2 is Z2 Camera, iPoint = 3 is Alignment Point 1, iPoint = 4 is alignment Point 2. If Alignment Point 1 and Alignment Point 2 are set to valid positions. GetAligned() will return true. Returns true if the function succeeds.

**Function ClearAlign() As Boolean**

Clears the current alignment. After calling GetAligned() will return false. Returns true if the function succeeds.

**Function ClearStop() As Boolean**

Clears the stop condition. Returns true if the function succeeds.

**Function RingLightControl(ByVal iZ1Ring as integer, ByVal iZ2Ring as integer) As Boolean**

Controls the ring light intensity. Returns true if the function succeeds.

**Function ProberMoveHome() As Boolean**

Moves the Probes to their home positions. Returns true if the function succeeds.