SpiderControl™

AutomationBrowser
Android Touch Panel Edition
Version 4

iniNet . Erfolgreiche Kunden
E-mail: info@ininet.ch. www.ininet.ch. Tel: +41 61 - 716 96 26

Table of contents

License Agreement

READ THE TERMS OF THIS AGREEMENT AND ANY PROVIDED SUPPLEMENTAL LICENSE TERMS (COLLECTIVELY "AGREEMENT") CAREFULLY BEFORE OPENING THE SOFTWARE MEDIA PACKAGE. BY OPENING THE SOFTWARE MEDIA PACKAGE, YOU AGREE TO THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL THESE TERMS, PROMPTLY RETURN THE UNUSED SOFTWARE TO YOUR PLACE OF PURCHASE.

1. LICENSE TO USE

ININET GRANTS YOU A NON-EXCLUSIVE AND NON-TRANSFERABLE LICENSE LIMITED TO THE INTERNAL USE OF THE ACCOMPANYING SOFTWARE AND DOCUMENTATION AND ANY ERROR CORRECTIONS PROVIDED BY ININET SOLUTIONS GMBH
(COLLECTIVELY "SOFTWARE").

2. GOVERNING LAW

THIS AGREEMENT SHALL BE GOVERNED BY THE LAWS OF KANTON BASELLAND, SWITZERLAND.

3. DISCLAIMER OF WARRANTY

THIS SOFTWARE AND THE ACCOMPANYING FILES ARE SOLD "AS IS" AND WITHOUT WARRANTIES AS TO PERFORMANCE OR MERCHANTABILITY OR ANY OTHER WARRANTIES WHETHER EXPRESSED OR IMPLIED. BECAUSE OF THE VARIOUS HARDWARE AND SOFTWARE ENVIRONMENTS INTO WHICH ININET SOLUTIONS MAY BE PUT, NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE IS OFFERED. GOOD DATA PROCESSING PROCEDURE DICTATES THAT ANY PROGRAM BE THOROUGHLY TESTED WITH NON-CRITICAL DATA BEFORE RELYING ON IT. THE USER MUST ASSUME THE ENTIRE RISK OF USING THE PROGRAM. ANY LIABILITY OF THE SELLER WILL BE LIMITED EXCLUSIVELY TO PRODUCT REPLACEMENT OR REFUND OF PURCHASE PRICE.

4. RESTRICTIONS

SOFTWARE IS CONFIDENTIAL AND COPYRIGHT. THE TITLE OF ANY SOFTWARE AND ALL ASSOCIATED INTELLECTUAL PROPERTY RIGHTS ARE RETAINED BY ININET SOLUTIONS GMBH AND/OR ITS LICENSORS. YOU MAY NOT MAKE COPIES OF SOFTWARE OTHER THAN A SINGLE COPY OF SOFTWARE FOR ARCHIVAL PURPOSES. YOU MAY NOT MODIFY, DECOMPILE, AND REVERSE ENGINEER SOFTWARE. SOFTWARE IS NOT DESIGNED OR LICENSED FOR USE IN ON-LINE CONTROL OF AIRCRAFT, AIR TRAFFIC, AIRCRAFT NAVIGATION OR AIRCRAFT COMMUNICATIONS; OR IN THE DESIGN, CONSTRUCTION, OPERATION OR MAINTENANCE OF ANY NUCLEAR FACILITY. YOU WARRANT THAT YOU WILL NOT USE SOFTWARE FOR THESE PURPOSES.

5. LIMITATION OF LIABILITY

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ININET SOLUTIONS GMBH OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF OR RELATED TO THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF ININET SOLUTIONS GMBH HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
IN NO EVENT WILL ININET SOLUTIONS GMBH'S LIABILITY TO YOU, WHETHER IN CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, EXCEED THE AMOUNT PAID BY YOU FOR SOFTWARE UNDER THIS AGREEMENT. THE FOREGOING LIMITATIONS WILL APPLY EVEN IF THE ABOVE STATED WARRANTY FAILS ITS ESSENTIAL PURPOSE.

6. TERMINATION

THIS AGREEMENT IS EFFECTIVE UNTIL TERMINATED. YOU MAY TERMINATE THIS AGREEMENT AT ANY TIME BY DESTROYING ALL COPIES OF SOFTWARE. THIS AGREEMENT WILL TERMINATE IMMEDIATELY WITHOUT NOTICE FROM ININET SOLUTIONS GMBH IF YOU FAIL TO COMPLY WITH ANY PROVISION OF THIS AGREEMENT.
UPON TERMINATION, YOU MUST DESTROY ALL COPIES OF SOFTWARE.

Abbreviations

HMI or MMI

Human-Machine or Man-Machine Interface, e.g. a SpiderControl™ views displayed at a touch panel or browser.

View

A view is what the user of an HMI sees at one moment inside a window or a browser. In SpiderControl a TEQ file implements a view.

TEQ File (*.teq)

File extension for a SpiderControl view file generated by the SpiderControl EDITOR.

URL

A URL (Uniform Resource Locator) is a web address that provides a unique, specific location for a particular resource on the internet (or on the local network).

VNC

VNC stands for Virtual Network Computing. It is a cross-platform remote desktop system that uses the Remote Frame Buffer (RFB) protocol to allow users to view and control another computer's graphical desktop from a remote location over a network.

RTSP

RTSP stands for Real-Time Streaming Protocol. It is a network protocol used to control media servers to stream audio and video content over the internet.

I4.0 and IoT

I4.0 or Industry 4.0, stands for the Fourth Industrial Revolution, can be defined as the integration of intelligent digital technologies such as IoT, into manufacturing and industrial processes
IoT stands for the Internet of Things. It refers to the network of physical objects.

RFID

RFID stands for “Radio Frequency Identification.” It's a form of wireless communication that uses radio waves created through electromagnetic coupling to communicate between devices

Trademarks

Installation

SpiderControl™ AutomationBrowser for Android™

The SpiderControl AutomationBrowser app for Android is available for download on this page:
https://www.ininet.ch/public/MicroBrowser/Android/automb.html

This Browser is intended to be used on Industrial Touch Panel with Android version 5.0 (Lollipop) and higher, for ARM or x86 processor. The SpiderControl AutomationBrowser app supports web visualization designed with SpiderControl Editor or any of the OEM editors, CODESYS V.2, CODESYS V.3, but it can also be used to open standard HTML5 web pages.

Attention

Please, pay attention that the SpiderControl AutomationBrowser is intended to be used on industrial web panels and not to be installed by customers on their smartphones or tablets from public market. Since this app requires an Activation code, depending on the Hardware. An update of your Android may cause the loss of the license key. Also, when you change your phone, you will have to buy a new license. Therefore, we suggest to install the SpiderControl MicroBrowser app from Google Play Store with your Google Play account, for this purpose.
https://play.google.com/store/apps/details?id=net.spidercontrol.app

Installation on your Android Touch Panel

Open this page on your Android device with a standard web browser, and click on the hyperlink of the APK file.

Download SpiderControl AutomationBrowser (Open link above)

Once the download is finished, open the APK file stored in your Download folder and follow instructions to complete the installation.
Note: Some authorizations can be asked to allow the installation of an APK file out of the official store.

For Industrial Touch Panel

If you wish the Automation Browser to be the main app of your device, then Automation Browser Home Screen edition is the right choice.
It will automatically start on reboot and every time you press the Home button it will show the app again.

Download SpiderControl AutomationBrowser (Home screen)

Once the download is finished, open the APK file stored in your Download folder and follow instructions to complete the installation.
Note: Some authorizations can be asked to allow the installation of an APK file out of the official store.

Home Screen (Launcher app)

After the installation of the SpiderControl AutomationBrowser Home Screen edition, you will be able to change the Launcher app. Press the Android Home button, you will be asked to choose for the default launcher(*). Press the Home button again to see the ALWAYS option.
Once you have chosen the Automation Browser (ALWAYS), it will automatically be launched on the next start and you will no longer see the default Android launcher app. To give you the choice again, you will have to clear the "Open defaults" setting in the Automation Browser app

Open the Android Settings (app menu in the top right corner) and depending on the Android version you will have to go under:

- Device > Home and select the app you want
- Apps > Automation Browser > Open by default > CLEAR DEFAULTS
- Storage & USB > Apps > Automation Browser > (i) icon > Open by default > CLEAR DEFAULTS

Afterwards, press on the home button again to have the choice and select the app you want to define as Home Screen launcher

(*) If you don't see the pop up to select the default launcher, it is probably because another launcher is already selected as default. In that case you will have to clear the "Open defaults" settings in the current launcher app. The name of the app is often something like "Launcher" or "Google Now Launcher", ...

Why use SpiderControl AutomationBrowser?

Brief Summary

- Support Panel Operation, Kiosk Mode
- One Browser always works, also with legacy HMI which is no longer supported in standard web browser
- Easy application installation, setup and maintenance
- The user is restricted to the desired URL
- The user has a list of all available stations (PLCs, servers)
- No exit to OS level, but allows some OS configurations (e.g. IP address)
- Interact with additional input devices: On-Screen-Keyboard, RFID, Scanner
- Have a better performance with CODESYS V3.x clients even on slower HW
- Allow remote control functions: PLC can turn on light on the panel
- Involve and control other programs on panel without leaving the user interface context
- Offer a platform for I4.0/IoT integration which can be used by factory level staff (no IT pro’s needed)

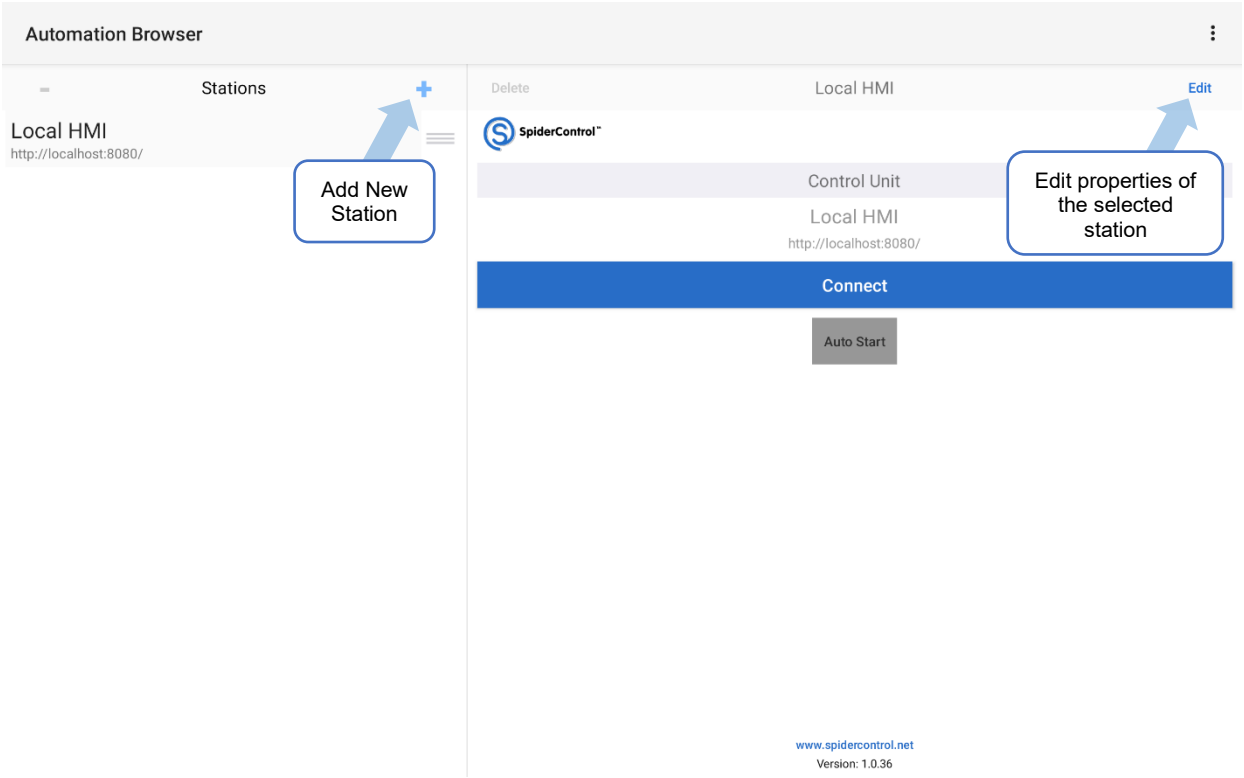
Detailed description

Web-based HMIs are today standard in automation for operation and monitoring. Through the available browsers on a wide variety of devices. Web technologies offer a high degree of simplification and modularity in the development of graphical user interfaces. The same technology also allows operation in the operator panel, on the PC, smartphone or tablet. But: so far, so good – where are still unresolved problems today?
Web-based HMIs can be easily stored directly on controllers, because integrated embedded Web servers are available almost everywhere today.
One of the main obstacles today is the legacy web-based HMIs, which are used on many installed controllers based on Java applets and are no longer supported by common browsers. These include, for example CODESYS WebVisu V2.x or even older SpiderControl OEM versions on PLC by Phoenix Contact, SAIA-Burgess, Panasonic and many others.
Another problem will arise when you have several automation systems in one system and the operator station has to jump back and forth between the various Web servers, so that the operator can even see all relevant information there. From technical point of view, this is not a problem. To switch from one server to another, the stored URL link will do. In practice, however, this can be complex and often problematic. That means you have to deposit the possible URL jumps in the HMIs and on all Web servers in advance. It is also a considerable effort and in some cases it´s not possible at all, if the Web HMI was developed by a third company. Another problem will be the log-ins. Usually there are several user levels in an operation for which one has to identify first. But if you jump from one Web server to another, this information is lost and the login procedure will start again. Another important issue is pop-up messages that should be displayed to the user immediately if there is a problem with the system. However, if this happens while the panel displays HMI from another Web server, the user does not get this error.
The SpiderControl AutomationBrowser by iniNet Solutions is developed to solve these problems and contains also other useful functionalities for automation. For example, a so-called station list can be created directly in the AutomationBrowser, in which all connected controllers are stored with their URL. This list can be displayed at any time, even if the browser is currently located on an HTML page of a controller. It is not necessary to make any changes to existing Web HMIs of controllers. The difference to a „favorites“ list – as well known from a conventional browser: as the browser on the panel is expected to operate in „kiosk mode“ (full screen), it cannot access all normal functions of the browser menu. Only the station selection and, if necessary, a “back” button will be displayed. Also, the station list can be displayed in a touch-screen-friendly, large format in order to increase user friendliness. This start page can even be customized by a programmer.
To solve the problem of compatibility with older web HMIs, several browsers have been integrated into AutomationBrowser. Invisible to the user, the AutomationBrowser simply selects the right viewer by itself, which makes it easy to integrate older Web visualizations like CODESYS V2.

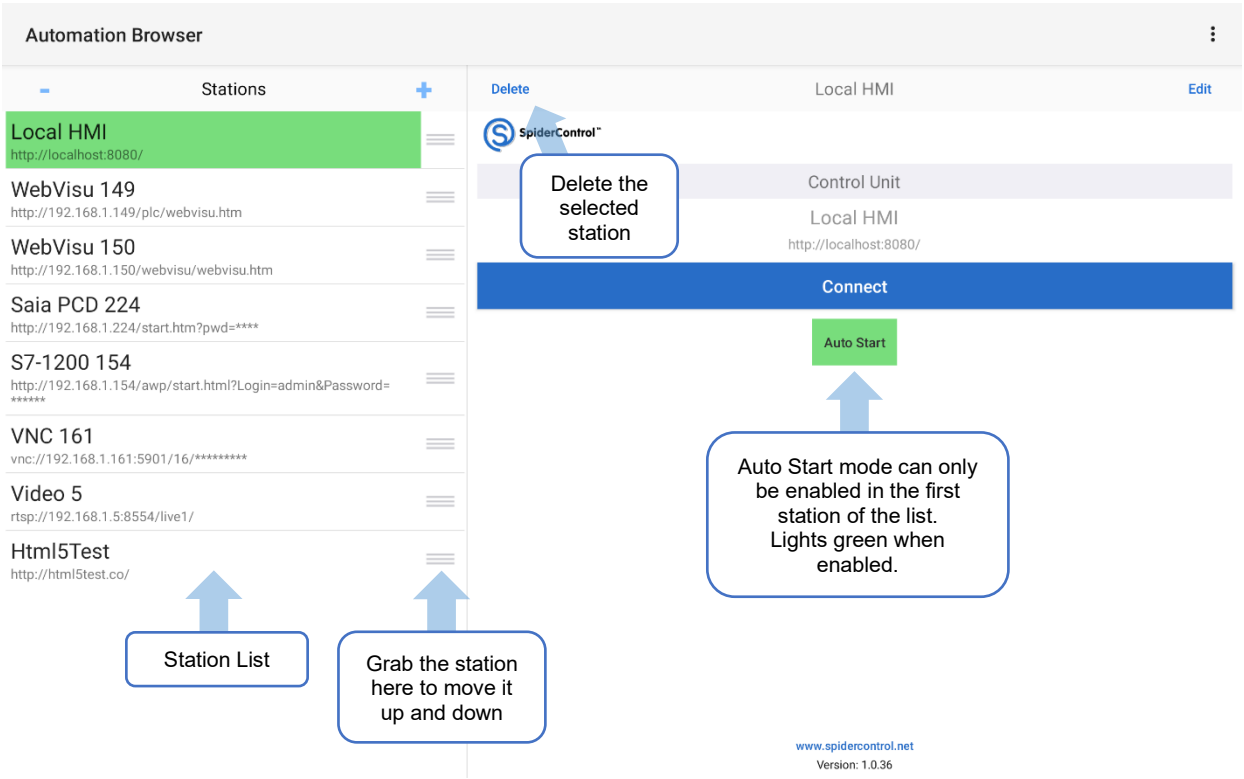
First Steps

Main window

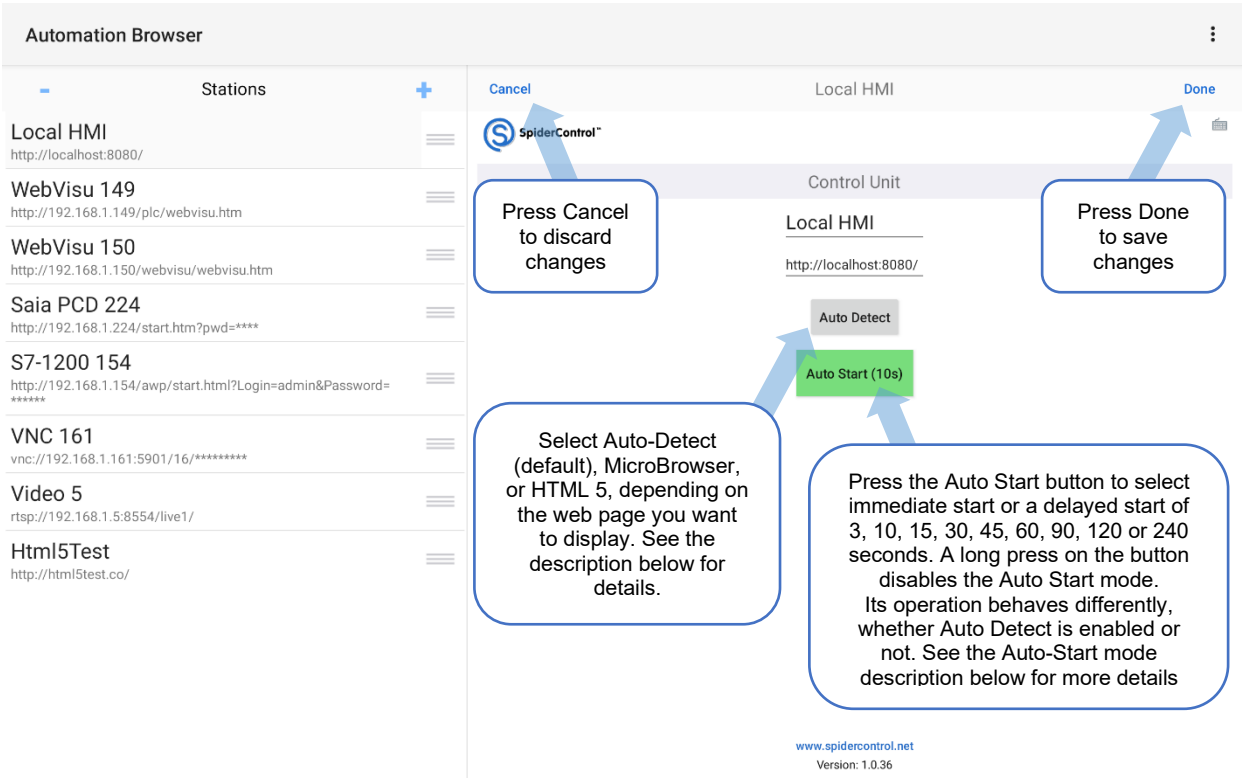
After successful installation, the following window appears:



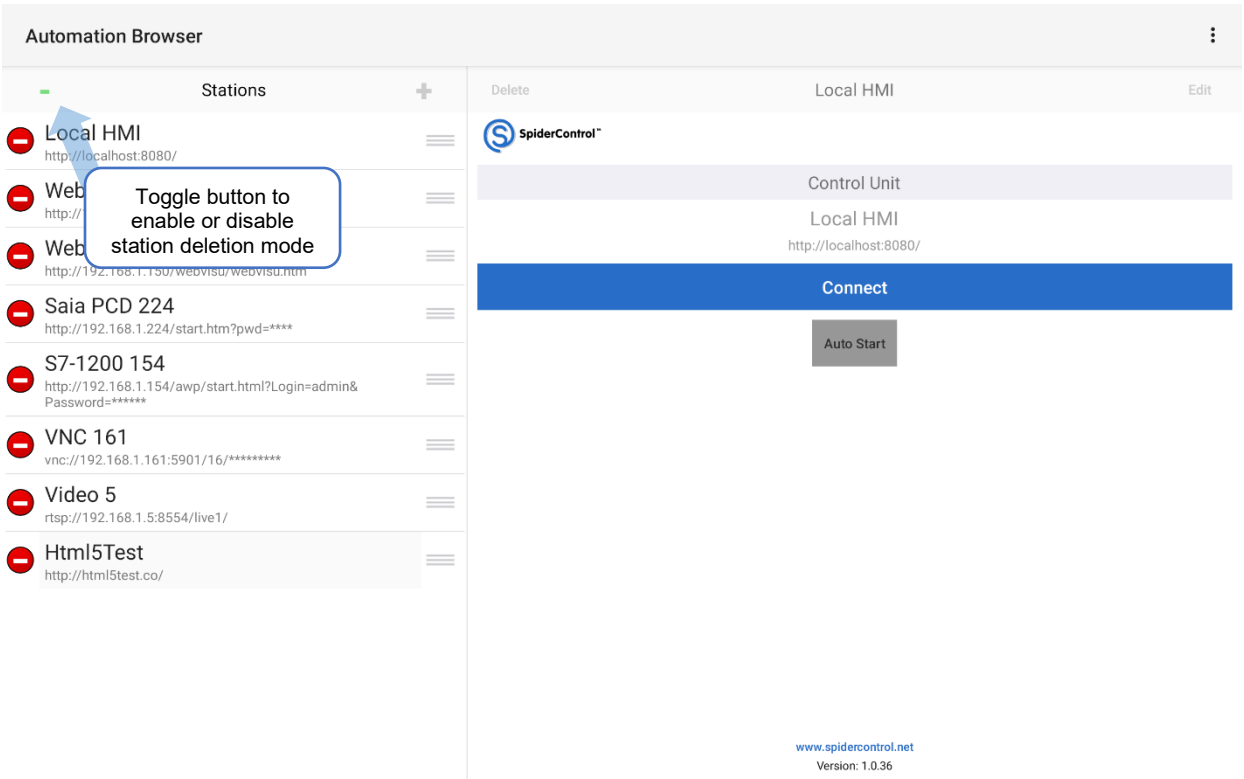
With more stations and Auto Start mode enabled on the first station in the list



Once you press the Edit button, this window will appear



Delete stations



MicroBrowser vs HTML 5 (Chromium)

The SpiderControl AutomationBrowser has two ways to open a web page, either with the MicroBrowser or with the Chromium web browser.
The MicroBrowser is able to display

- All Web-HMI designed with a SpiderControl HMI Editor or an OEM version of it
- CODESYS WebVisu V 2.3
- CODESYS WebVisu V 3.x

For all other content, the Automation Browser will use the built-in HTML5 Chromium web client.

When opening a URL with the default option (Auto-Detect), the Automation Browser will first analyze the HTML page and then automatically open the MicroBrowser or Chromium. Older SpiderControl HMI projects as well as the CODESYS WebVisu V 2.3 using Java Applets, which are no longer supported in any Browser. The MicroBrowser is able to display these HMI without a Java VM by using a native implementation. CODESYS WebVisu V 3.x can be visualized both with the MicroBrowser as well as with Chromium HTML5. The MicroBrowser offers better performance as well as other possibilities with some limitations, so for this type of HMI the Automation Browser will open the Chromium web browser when Auto Detect (or HMTL5) mode is enabled.
When you display a newer HTML5 based SpiderControl HMI, the Auto Detect mode will open the MicroBrowser, but you can also force it to use the Chromium web browser by selecting the HTML 5 mode.

The Android System WebView (Chromium)

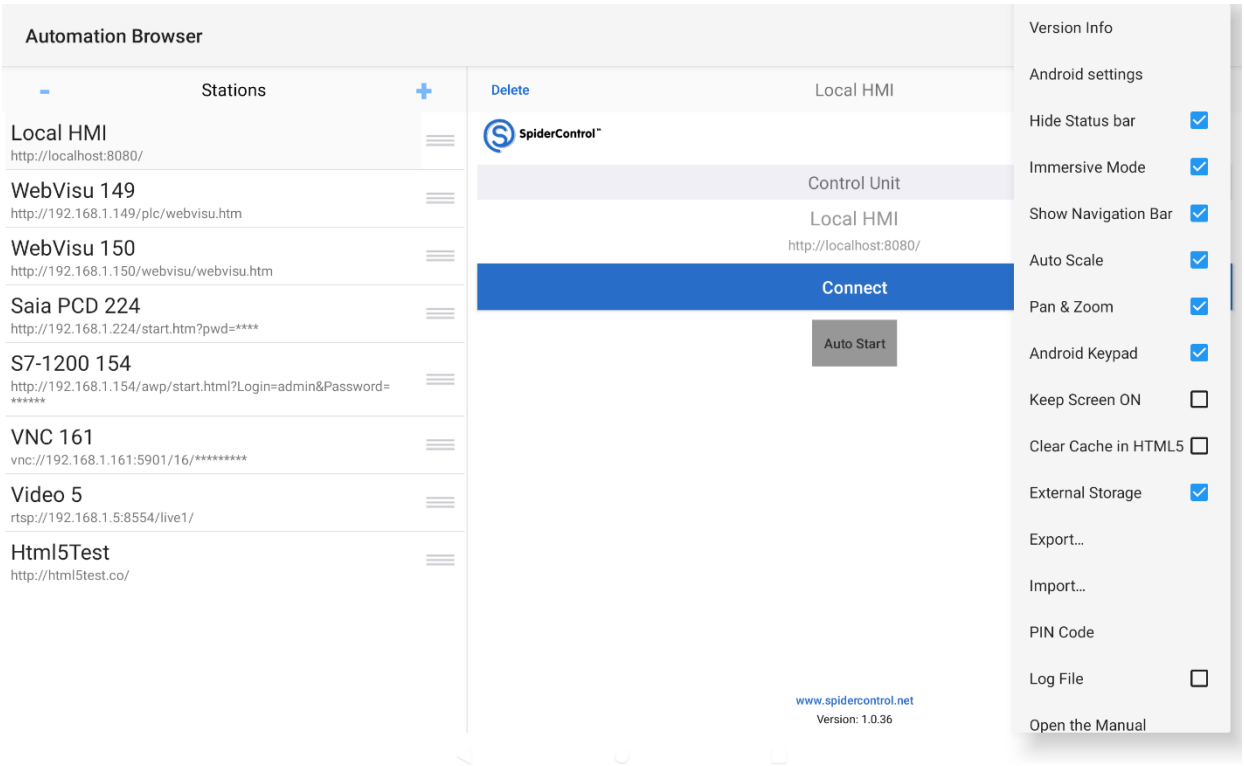
To support the latest HTML5 features, it's highly recommended to install the latest version of the Android System WebView by Google LLC, on your device. This software component can be found here:
https://www.apkmirror.com/apk/google-inc/android-system-webview/

If needed, contact your vendor to find out for which platform and version you need to download and install the APK file.

To find out which version is currently used on your device and which HTML5 features are supported, open the following URL in the SpiderControl AutomationBrowser app:
http://html5test.co (.co and not .com)

Main menu

The main menu can be opened in the upper right corner (3 dots)



Version Info

Show the app version, Android version and Android WebView component version. Also, we can see the version of the SpiderControl SCADA Server, if it is installed. To update the app, it's preferable to ask the vendor of the Android panel or send us the request with a screenshot of the Version Info box.

Android settings

This option is only available in the AutomationBrowser (Home Screen), to open the Android Settings in order to modify network, display, date and time, and many other parameters of the system. This is useful in Home Screen mode, since the user cannot switch to another App.

Hide Status bar

Option to hide and show the status bar in the top of the screen.

Immersive Mode

Immersive Mode is used to remove both the status bar and the navigation bar at the bottom of the screen to enjoy a full screen display. We can display the navigation bar again, for example, to return to the stations list, by swiping up from the bottom of the screen.

Show Navigation Bar

This option is only used in HTML 5 mode, to show a control bar at the top of the screen, to navigate with previous and next buttons, to refresh the view or to return to the home view.

Auto-scale

Enable or disable auto scale of the web page
In MicroBrowser: The view will automatically resize to fit the screen (isotropic, maintaining the aspect ratio)
In HTML 5: This is not really supported, as it depends on the HTML page code. But it will activate an option of the WebView control to fit, as much as possible the screen, if the html page (web project) supports it.

Pan & Zoom

In MicroBrowser: Enable/disable panning and zooming of the view.
In HMTL 5: Unused, since it is handled in the HTML code

Anti-Alias

This option is only available in apps built with MicroBrowser in 16-bpp
It will improve the rendering with anti-alias if the Android device does not already support this feature in the graphic controller (HW). In most recent devices we don't need to enable this option.

Android Keypad

In MicroBrowser it will show the Android Keypad to edit a value or use the keypad/alphapad, built-in TEQ files
In HTML 5 this option is not used, it will show the Android Keypad, or use the proprietary input method supported in the web page.

Keep Screen ON

Keep Screen ON, once connected to the target and displaying the MicroBrowser, HTML 5, VNC, or Video view

Clear Cache in HTML5

Clears HTML5 cache of the WebView on each connection to the target

External Storage

If this option is activated, it will import/export and save all files on the first external storage found (SD card, USB stick…)

Export

Export station list, configuration and log file if any into Download/AutomationBrowser/

Import

Import station list and configuration file from Download/AutomationBrowser/

PIN Code

PIN code is used to prevent any modification from operator user. The password is needed to change anything. It allows locking up the operator panel to avoid a user to modify settings or to exit the App (Kiosk Mode)

Log File

Option to generate a log file
/Download/AutomationBrowser/automb_log.txt

Open the Manual

Open this manual in the PDF viewer

License Report

Shows information regarding open-source licenses (PDF)

Close

Close the menu

Extending the AutomationBrowser

The version that you have installed so far covers the basic functionality. If you need a customized behaviour, the AutomationBrowser can be extended with the SpiderPLC components. In the following, we present two examples of such extensions. If you have an application which could be solved using this approach, please contact us.

AutomationBrowser with seamless navigation from PLC to camera

The SpiderControl AutomationBrowser allows for a seamless navigation from a HTML5 WebVisu on a PLC to a web-cam which is linked directly with its H264 / rtsp:// URL (which is supported by almost any webcam).
A fully customized user interface in the control panel offers the option of providing a self-contained, consistent operation.
https://www.youtube.com/watch?v=ohQA5tI2A8E

AutomationBrowser with programmable logic to jump, block touch and switch backlight

The SpiderControl AutomationBrowser can be programmed with an integrated functional block logic to force a jump to a specific URL, to block the touch screen or to switch on/off the back light. The integrated SpiderPLC can be programmed with any standard Browser and connects to external PLC using standard protocols like OPC UA, Modbus, ISO-on-TCP and more. Like this, a Web-Panel can be used to display several Web-HMIs, but still be controlled by a connected PLC.
https://www.youtube.com/watch?v=2klVhjvNuk8

Features

HTTP / HTTPS (Authentication)

The AutomationBrowser supports both http and https URL, with optional credentials.

```
http://[user:password@]hostname/...
https://[user:password@]hostname/...
```

If credentials are not specified in the URL and the server prompts for HTTP authentication, it will pop up an authentication dialog, to enter the username and password.
If you choose "Save Password" in this dialog, it will still show the authentication dialog the next time, to allow the user to use the stored information or enter another username/password.
If credentials are specified in the URL, the HTTP authentication dialog will not be displayed, unless the specified credentials are wrong. In that case, you will have to update the credentials from the URL.
If the server uses as untrusted SSL certificate or self-made certificate, the user will be prompted to accept it and continue loading the page, or not.

- Press YES (ALWAYS) to permanently save your choice.
- Press CLEAR DATA in the Android Settings of the app, to restore the default settings and clear all stored credentials.

VNC

The AutomationBrowser supports VNC, with an URL like this

```
vnc://hostname[:port]/[bpp[.depth]]/[password]
```

Optional parameters:

- port, the default port number is 5900 and can be omitted.
- depth is an optional parameter and depends on the bpp parameter. Supported values are 16.15 (555), 8.6 (64 colors) or 8.3 (8 colors)
- password if need, must be defined after the pixel format. Use 0 for the default pixel format

If supported on the VNC server side, the optimal choice is often 16 bpp (565), for example:

```
vnc://192.168.1.123/16/
```

Notes:

- A Floating Button is used to open and close the Android Keypad,
- The Floating Button is visible only if Android Keypad option is enabled in the menu.
- A long press on the Floating Button helps to drag the button.
- The Auto-Start in MicroBrowser mode is recommended to be used instead of Auto detection
- If the connection is lost and the Auto Start is defined with a delay of 3s or more, it will return to the countdown page and retry the connection every 10s

Video streaming

The SpiderControl AutomationBrowser is able to display a video stream from a camera, using an RTSP URL like this

```
rtsp://[user:password@]hostname/[live0][?caching=MILLISECONDS]
```

Example: `rtsp://192.168.1.123/live0`
Depending on the camera, you will have to specify (or not) something like /live0, /live1, … after the hostname.
The optional network caching parameter in the rtsp URL, allows you to reduce the live video latency, but may cause artifacts or refresh issues if the setting is too low.
The default value is 200 ms, for 50 ms:
`rtsp://192.168.1.123/live0?caching=50`

Auto-Start mode

⇨ Immediate start

1. Auto start with auto detection will shortly show the setup page and open either the MicroBrowser or HTML 5 view
2. Auto start with MicroBrowser will directly jump to the MicroBrowser if the server is reachable (*)
3. Auto start with HTML 5 will directly jump to the WebView, if the server is reachable (*)

⇨ Delayed start of 3, 10, 15, 30, 45, 60, 90, 120 or 240 seconds

1. Auto start with auto detection will show the setup page and either open the MicroBrowser or HTML 5 view after a delay of the specified seconds
2. Auto start with MicroBrowser will jump to the MicroBrowser view if the server is reachable after a delay of the specified seconds (**)
3. Auto start with HTML 5 will jump to the WebView, if the server is reachable after a delay of the specified seconds (**)

Notes:
- (*) If the server is not reachable or if the connection is lost, it will jump to the setup page.
- (**) If the server is not reachable or if the connection is lost, it will do endless retries every 10 seconds. To abort the endless retries and return to the setup page, press 5 times in the top-left corner. Or press the back button if available.
- Instead of showing the loading message "Loading...", set a boot image to show on startup, loaded from `/sdcard/Download/bootscreen.png`
- Install the SpiderControl AutomationBrowser Home Screen edition to enable the application to start automatically when the device boots.

Single station mode

Single station mode is used to directly start the specified URL in the XML app setting file, skipping the setup page. In this mode, the end user will not see the Android-like setup page.

File: automb.xml

```
<settings>
  <name>My Visu</name>
  <start_url>http://localhost/Visu.html</start_url>
  <start_mode>MicroBrowser</start_mode>
  <auto_start_delay_sec>3</auto_start_delay_sec>
  <use_android_keypad>false</use_android_keypad>
</settings>
```

⇨ Procedure with loading file from External Storage

1) Export files from Automation Browser into external storage (USB, SD)
2) Copy automb.xml into:
   for Android < 10:
   `<ExternalStorage>/Download/AutomationBrowser/automb.xml`
   for Android >= 10:
   `<ExternalStorage>/Android/data/net.spidercontrol.automb/files/automb.xml`
   (Note: The folder must already exist on the external storage)
3) Import files in Automation Browser (menu)

Once the XML file has been loaded with `<start_url>`, it will automatically start with the specified URL and settings.
Pressing back button will close the app.
In this mode, you don't see the standard station list, menu, config, etc...
To restore the standard mode, remove the XML file from external storage (or remove the external storage)

⇨ Procedure with loading file from Internal Storage

1) Export files from Automation Browser into internal storage
2) Copy automb.xml into:
   for Android < 10:
   `<InternalStorage>/Download/AutomationBrowser/automb.xml`
   for Android >= 10:
   `<InternalStorage>/Android/data/net.spidercontrol.automb/files/automb.xml`
   (Note: The folder should already exist on the internal storage)
3) Import files in Automation Browser (menu)

To restore the standard mode, remove the XML file (automb.xml) from internal storage

All settings that can be set in the XML file are:

```
<settings>
  <name>My Visu</name>
  <start_url>http://192.168.1.123/Visu.html</start_url>
  <start_mode>MicroBrowser</start_mode>
  <auto_start_delay_sec>3</auto_start_delay_sec>
  <show_navig_bar>false</show_navig_bar>
  <full_screen>true</full_screen>
  <immersive>false</immersive>
  <auto_scale>true</auto_scale>
  <pan_zoom>true</pan_zoom>
  <use_android_keypad>false</use_android_keypad>
  <use_external_storage>false</use_external_storage>
  <pin_code></pin_code>
</settings>
```

`<start_mode>` is either an integer or a string
0: Auto (Auto detect is not supported for auto start)
1: HTML5
2: MicroBrowser

Frameset in HTML file

The Automation Browser supports HTML FRAMESET, but only few configurations are supported to see either 2, 3, 4 or 6 frames
An URL can be defined for each frame, either a http URL for HTML5 WebView or an RTSP URL for video streaming.

For example, to split the display into two views (left and right), I will create an HTML file like frameset2.html and load it from the server.

File: frameset2.html

```
<FRAMESET cols="50%,50%">
  <FRAME src="http://192.168.1.20/frame1.html">
  <FRAME src="http://192.168.1.20/frame2.html">
</FRAMESET>
```

File: frameset4.html

```
<FRAMESET rows="50%,50%" cols="50%,50%">
  <FRAME src="http://192.168.1.20/frame1.html">
  <FRAME src="http://192.168.1.20/frame2.html">
  <FRAME src="rtsp://192.168.1.10/live0">
  <FRAME src="rtsp://192.168.1.11/live1">
</FRAMESET>
```

File: frameset3.html

```
<FRAMESET cols="50%,50%">
  <FRAME src="http://192.168.1.20/Visu.html">
  <FRAMESET rows="50%,50%">
    <FRAME src="rtsp://192.168.1.5:8554/live1">
    <FRAME src="rtsp://192.168.1.6:8554/live1">
  </FRAMESET>
</FRAMESET>
```

File: frameset6.html

```
<FRAMESET rows="50%,50%" cols="33%,33%,33%">
  <FRAME src="rtsp://192.168.1.101/live1">
  <FRAME src="rtsp://192.168.1.102/live1">
  <FRAME src="rtsp://192.168.1.103/live1">
  <FRAME src="rtsp://192.168.1.104/live1">
  <FRAME src="rtsp://192.168.1.105/live1">
  <FRAME src="rtsp://192.168.1.106/live1">
</FRAMESET>
```

Notes:
- Frame size in pixel (px) or percent (%) are not supported.
- Relative URL like `src="frame1.html"` is not supported.

CODESYS MicroBrowser FAQ

Following questions are being asked often concerning the MicroBrowser CODESYS.

I can't connect! It says “File Not Found!”

Some CODESYS PLC's have a case sensitive file system (Linux). Older versions of MicroBrowser CODESYS get the entry page with the name “PLC_VISU.xml” written in big letters. But the file on the web server is written “plc_visu.xml”. Possible workaround: Modify the “webvisu.htm” File on your PLC and change this line:

```
<param name="STARTVISU" value="PLC_VISU">
```

to

```
<param name="STARTVISU" value="plc_visu">
```

Why are arrays not displayed correctly?

Normally arrays are not working with the applet from CODESYS itself. But with our MicroBrowser CODESYS it is possible, BUT the indexed variants of the needed array elements must also exist in the view, otherwise the variable address is unknown.

Example:
`".g_afb_GF[.g_index].i_bo_configured"`
is an indexed variant which can be read and written with MicroBrowser CODESYS, but you must already know during build-time of your project, which indexes are used, and you also have to include the resolved variants of the variable in the view:
`".g_afb_GF[6].i_bo_Configured"`
(for instance, in a hidden text-field).