

常问问题 • 01 月/14 年

# S120 自定义网页功能

S120、Web

---

## 目录

<b>1 概述</b> .....	<b>3</b>
1.1 标题 .....	3
<b>2 网页基础</b> .....	<b>4</b>
2.1 超文本标记语言 (HTML) .....	4
2.2 Mini WebServer 语言 (MWSL) .....	6
2.3 JavaScript 语言 .....	7
2.4 CSS 语言 .....	8
<b>3 网页的准备和上传</b> .....	<b>9</b>

# 1 概述

## 1.1 标题

从固件版本 V4.6 开始，S120 支持 Web 功能，通过 Web 功能可以实现 S120 设备的调试和诊断功能。SINAMICS S120（CU310-2、CU320-2 V4.6 固件版本）的 Web 可以通过调试接口（X127）进行访问，并且对具有 PROFINET 接口的控制单元 CU，也可以控制单元 CU 集成的 PROFINET 接口进行访问。访问 Web 服务器可以使用标准的 Web 浏览器（如 Internet Explorer 或 Chrome 以及 Mozilla Firefox）。用户可以自己创建网页（自定义的 Web 页面）随后上传到 SINAMICS S120 的 Web 服务器，使用这些 Web 页面可以读取驱动参数实现需要的可视化效果。

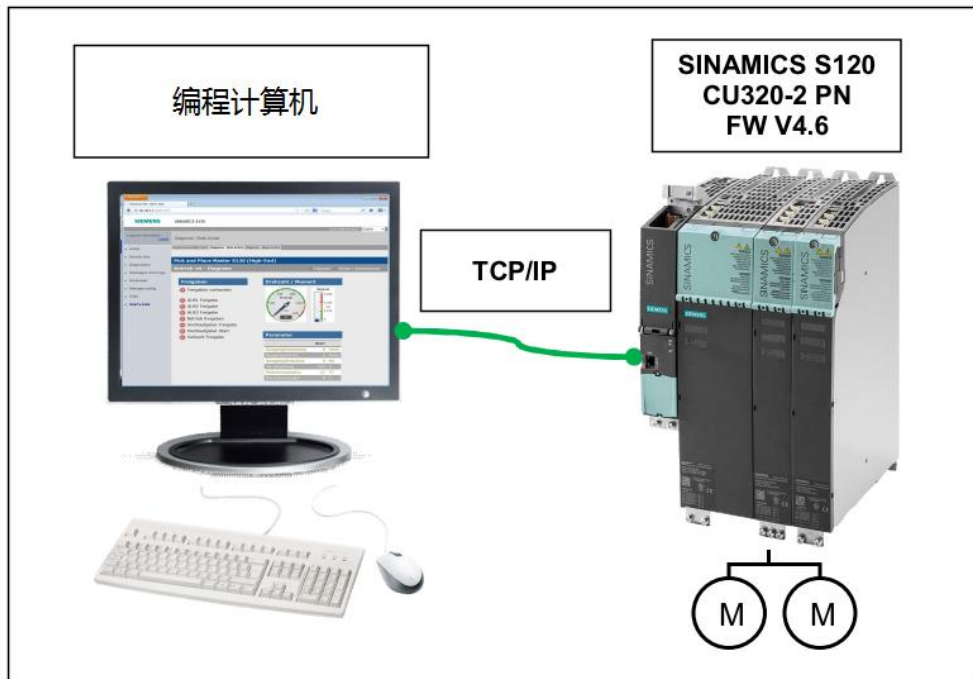


图 1-1 S120 Web 功能图

S120 的标准网页功能可以参考如下链接的手册，第 6 章 28 节 网络服务器部分：

<http://support.automation.siemens.com/CN/view/zh/68042590/0/zh>

以下内容介绍通过 S120 的自定义网页功能实现驱动数据的显示，通过 HTML 语言以及 java script 即可实现自定义的驱动数据刷新和显示。

在 S120 中使用自定义网页，需要准备一些必要的文件：

- Html 文件，显示网页的内容，此文件需要专用的工具转换为后缀为\*.mbs 后传送到 s120 中。
- JS 文件，用于动态刷新数据和实现一些基本的动画显示。建议使用西门子提供的脚本库，可以缩短编程时间。
- 使用 MWSL 专用语言编写的驱动变量文件，此文件用于提供网页中显示的驱动变量数据，例如当前转矩，当前电流等信息。
- CSS 文件（可选），网页的内容进行格式化描述，字体，颜色等等。

即在网页 html 中显示的驱动数据需要使用脚本语言 Mini Web Server (MWSL) 预先定义，并使用 JavaScript 让数据在以特定的时间间隔刷新。例如层叠样式表 (CSS) 设计网页的外观和形式并以\*.css 格式集中保存。

## 2 网页基础

### 2.1 超文本标记语言 (HTML)

超文本标记语言 (HTML) 是一个基于文本的标记结构的语言，可以操作如文本、图形和超链接的文件。HTML 文件是 Web 的基础，利用 Web 浏览器可以显示 HTML。本文介绍在使用 HTML 语言的基本方法。更多内容请参考下述链接：<http://www.w3school.com.cn/html>

一个基本的 HTML 语言结构如下所示，可以把例子 1 内容复制到 txt 文件中，把文件后缀修改为 html，即可在浏览器中看到网页内容，如下为例子 1:

```
<html >
  <head>
    <title>
      //title of the Website
    </title>
  </head>
  <body>
    //content of the Website , only for first time test!
  </body>
</html >
```

通过<html> 和 </html> 之间的文本符号表示 HTML 代码范围，在这个代码范围内通过<head>、<body>进行进一步的划分。<title>表示网页的标题，<body>部分包含实际的网页内容。

在 body 内部可以使用一些基本的元素进行网页的定义:

- (1) <div>和<span> 可以进行划分区域，即在网页中划分一个独立的显示区域，通过 CSS (2.4 节介绍) 的进一步定义，可以指定每一行的颜色、字体和位置关系等等。

- (2) <table>可以划分一个表格，在表格内部可以通过<tr>进行每一行的划分，每一行的元素可以通过<td>进行描述。例如：

```
<table>
  <tr>
    <td>
      first row, first data cell
    </td>
    <td>
      first row, second data cell
    </td>
  </tr>
</table>
```

- (3) <select>可以实现下拉菜单的效果，通过<option>定义下拉菜单的具体内容，size属性决定同时显示几个条目。例如：

```
<select id=selectList size=1 onchange=" switch()" >
  <option value=0> Off </option>
  <option value=1> On </option>
</select>
```

在上面的例子中实现的是一个开关的下拉菜单选择，当进行修改的时候会触发 java 脚本中的 switch 函数。Java 脚本通过 id 以及属性 value 进行数据的处理。

- (4) <ul>列表功能，在列表中的成员由<li>进行定义。例如：

```
<ul >
  <li>...</li>
  <li>...</li>
</ul >
```

- (5) iframes (inline 框架=嵌入式框架) 是一个布局资源，允许开发者在 HTML 页面嵌入第三方来源。不同于正常的框架，在这种情况下，Web 页不拆分。在页面中定义一个地区，即保留显示源（如图形或其他 Web 页等。iframes 使用<iframe>和</iframe>进行划分。例如：

```
<iframe src=" picture1.jpg" id=" iframe" >
  Your browser does not support IFrames.
</iframe>
```

在例子中表示如果浏览器支持 iframe 则显示图片，如果不支持则显示文字“ Your browser does not support IFRAMES” 。如果需要通过 css 修改相关的属性则需要分配 ID。

- (6) <canvas> 标签用于绘制图像（通过脚本，通常是 JavaScript）。可以通过 canvas 实现曲线、颜色生成、动画等等。需要注意，canvas 需要浏览器的支持：Internet

Explorer from Version 9.0 、 Mozilla Firefox from Version 3.6 、 Opera from Version 11.0。例如：

```
<canvas id=" canvas" width=" 160" height=" 160" >
  Canvas is not supported by your browser.
</canvas>
```

通过一个基本的 HTML Web 实现 S120 参数显示的编程举例如下，在头部声明了使用两个脚本（随后解释），在内容中显示 infeed 模块的直流母线电压和驱动轴的速度，如下为例子 2:

```
<html >
<head>
  <title>Pick and Place Master - Diagnostics</title>
  <script src="scripts/libByMichael.js" type="text/javascript"></script>
  <script src="scripts/readData.js" type="text/javascript"></script>
</head>
<body onload="setupPage()">
<table>
  <tbody>
    <tr>
      <td>DC-link voltage</td>
      <td id="valDcLinkVoltage">
        <span class="value">---</span>
      </td>
      <td>V</td>
    </tr>
    <tr>
      <td>Speed</td>
      <td id="valActualSpeed">
        <span class="value">---</span>
      </td>
      <td>S</td>
    </tr>
  </tbody>
</table>
</div>
</body>
</html >
```

## 2.2 MiniWebServer 语言 (MWSL)

通过 MWSL 可以访问驱动中的参数，为此需要单独定义一个网页文件用于声明需要访问的驱动器变量，这些变量可以通过 ID 标志在其他的 HTML 网页中显示和刷新(使用 Javascript 进行)。MWSL 通过<MWSL>和</MWSL>进行标识。

可以在 MWSL 中定义变量和编写一些逻辑函数。为了使上文中的速度和电压能够正常显示, 需要进行变量的定义, 用户可以根据如下例子进行必要的修改来实现需要的驱动数据显示。注意: 需要定义变量的名称必须与项目中的驱动对象名称一致, 例如 A\_INF\_02、SERVO\_02 等等。

变量文件 readV 举例如下, 例子 3:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<root>
  <versions>
    <document>1</document>
    <minInterpreter>1</minInterpreter>
  </versions>
  <variables>
    <numericVariables>
      <val DcLinkVoltage>
        <MWSL><!--
          WriteVar("A_INF_02.Params." + "70", "PROCESS", "%3.0f");
        --></MWSL>
      </val DcLinkVoltage>
      <val Actual Speed>
        <MWSL><!--
          WriteVar("SERVO_02.Params." + "22", "PROCESS", "%3.0f");
        --></MWSL>
      </val DcLinkVoltage>
    </numericVariables>
  </variables>
</root>
```

### 2.3 JavaScript 语言

JavaScript 是一种主要用于动态 Web 的脚本语言, 可对用户行为的变化、重新加载或生成页面时进行相关的函数处理, 因此扩展 HTML 功能。作为一个结果, 所提供的 JavaScript 动态网页的功能, 可以生成在静态 HTML 页面里。所有的 JavaScript 被限制仅对网页中定义或者包含的数据进行访问。西门子提供了一个脚本库: libByMichael.js, 可以通过该脚本库进行驱动数据的刷新和访问并且支持两种动画的显示方式:

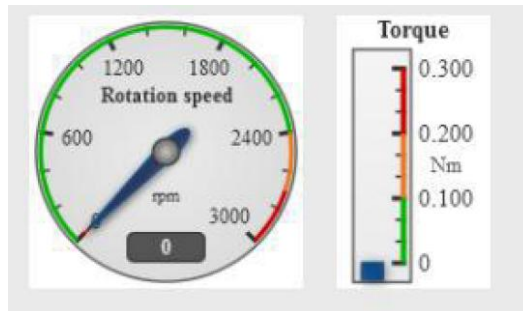


图 2 java 脚本实现的动画

---

对于 Java 脚本常用的几种启动事件为:

- (1) onload: 装载文件时触发函数
- (2) onclick: 点击鼠标时触发函数 (例如一个按钮)
- (3) onchange: 当修改时触发函数 (例如下拉菜单的选择)

在本例中使用 onload 事件, 此事件填写在 2.1 节例子 2 的 html 文本中:

```
<body onload="setupPage()">。
```

SetupPage 函数的定义在 myWeb.js 文件中, 以 3 秒的频率刷新 2.2 节中定义的例子 3 驱动数据文件 readV.mbs, 注意不要设置刷新时间小于 1 秒:

```
/* Init the canvas controls of the Web page and change to cyclic updating mode */  
function setupPage() {  
/* Change to cyclic updating mode with a period of 3000ms */  
    setInterval(function () { updateDocument.updateValuesEx('readV.mbs', null,  
null) }, 3000);  
}
```

## 2.4 CSS 语言

通过 CSS 可以对网页的内容进行格式化描述, 常见的描述通过 class 和 id 进行描述, 举例如下:

Class:

```
h1 {  
    font-family : Arial;  
    font-size : 2em;  
    font-weight : normal;  
}  
h1.back {  
    background-color : #FFFF00  
}  
*.back {  
    background-color : #00FFFF  
}
```

Id:

```
#redArea {  
    position : absolute;  
    top : 130px;  
    left : 30px;  
    width : 320px;  
    padding : 10px;  
    margin : 0px;  
    border : 4px solid #EE0000;  
}
```

Class 和 id 的区别为, 一个独立特定的对象为 id, 可以反复使用的对象为 class 类。



### 3 网页的准备和上传

表 1. 网页文件 myWeb.html 和参数文件 readV.html 的介绍

序号	网页 myWeb.html	参数文件 readV.html
1.	<p>显示一个 BOOL 型的变量：并且通过一个红色或者绿色的图片进行状态的显示，该状态的处理通过 libByMichael.js 来实现</p> <pre> &lt;div&gt;   &lt;div&gt;     &lt;img id="indicatorOff1Enable"           src="images/indicatorOff.png"           alt="off" /&gt;     &lt;p&gt;OFF1 Enable&lt;/p&gt;   &lt;/div&gt; &lt;/div&gt; </pre>	<p>通过如下参数定义，把绿色的省略号替换为添加需要显示的变量，例如：</p> <pre> write((GetVar("SERVO_02.Params." + "898") &amp; 0x0001));  &lt;indicatorVariables&gt;   &lt;indicatorOff1Enable&gt;     &lt;MWSL&gt;&lt;!--       ...     --&gt;&lt;/MWSL&gt;   &lt;/indicatorOff1Enable&gt; &lt;/indicatorVariables&gt; </pre>
2.	<p>显示变量的仪表盘动画：可以如下定义，该 id gaugespeed 的处理由 libByMichael.js 来实现</p> <pre> &lt;div&gt;   &lt;div&gt;     &lt;canvas id="gaugeSpeed"             width="160"             height="160"&gt;     &lt;/canvas&gt;   &lt;/div&gt; &lt;/div&gt; </pre>	<p>把绿色省略号替换为需要显示的变量，例如</p> <pre> WriteVar("SERVO_02.Params." + "22", "PROCESS", "%3.0f");  &lt;gaugeVariables&gt;   &lt;gaugeSpeed&gt;     &lt;value&gt;       &lt;MWSL&gt;&lt;!--         ...       --&gt;&lt;/MWSL&gt;     &lt;/value&gt;     &lt;setValue&gt;       &lt;MWSL&gt;&lt;!--         ...       --&gt;&lt;/MWSL&gt;     &lt;/setValue&gt;   &lt;/gaugeSpeed&gt; &lt;/gaugeVariables&gt; </pre>
3.	<p>显示变量的数值，该数值的刷新也是由 libByMichael.js 来实现：</p> <pre> &lt;div&gt;   &lt;table&gt;     &lt;tr&gt;       &lt;td&gt;Output Voltage&lt;/td&gt;       &lt;td id="valOutputVoltage"&gt;         &lt;span class="value"&gt;---&lt;/span&gt;         &lt;span class="warningLimit"&gt;300&lt;/span&gt;         &lt;span class="criticalLimit"&gt;320&lt;/span&gt;       &lt;/td&gt;       &lt;td&gt;Vrms&lt;/td&gt;     &lt;/tr&gt;   &lt;/table&gt; &lt;/div&gt; </pre>	<p>把绿色省略号替换为需要显示的变量，例如</p> <pre> WriteVar("SERVO_02.Params." + "70", "PROCESS", "%3.0f");  &lt;numericVariables&gt;   &lt;valOutputVoltage&gt;     &lt;MWSL&gt;&lt;!--       ...     --&gt;&lt;/MWSL&gt;   &lt;/valOutputVoltage&gt; &lt;/numericVariables&gt; </pre>

序号	网页 myWeb.html	参数文件 readV.html
4.	显示柱状变量动画：  <pre> &lt;div&gt;   &lt;div&gt;     &lt;canvas id="barTorque"             width="160"             height="160"&gt;     &lt;/canvas&gt;   &lt;/div&gt; &lt;/div&gt; </pre>	把绿色省略号替换为需要显示的变量，例如 <pre> WriteVar("SERVO_02.Params."+ "31", "PROCESS", "%3.0f"):   &lt;barVariables&gt;     &lt;barTorque&gt;       &lt;value&gt;         &lt;MWSL&gt;&lt;!--           ...         --&gt;&lt;/MWSL&gt;       &lt;/value&gt;     &lt;/barTorque&gt;   &lt;/barVariables&gt; </pre>
5.	根据实际的需要进行其他数据的定义和网页的编写。	

同时需要准备 java 脚本的文件，在该文件中如果使用了仪表盘或者柱状条动画则需要  
在该脚本中进行必要的初始化 myWeb.js 文件为：

```

function ini tGauges() {
  var gaugeSpeed = new canvasControls.Gauge(' gaugeSpeed', 0, 3000, ' Rotati on
speed', ' rpm');
  gaugeSpeed.addDefaul tCol oredSecti ons(0, 2400, 2700, 3000);
  gaugeSpeed.refresh(null, null);



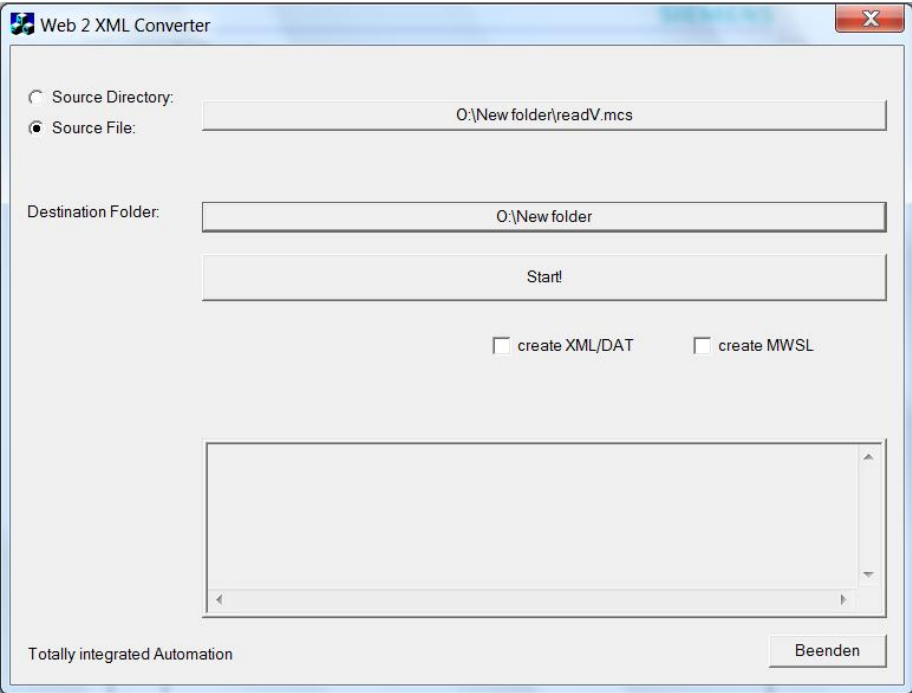
  return [gaugeSpeed];
}
function ini tBars() {
  var barTorque = new canvasControls.Bar(' barTorque', 0, 0.3, ' Torque', ' Nm');
  barTorque.addDefaul tCol oredSecti ons(0, 0.1, 0.2, 0.3);
  barTorque.refresh(null);

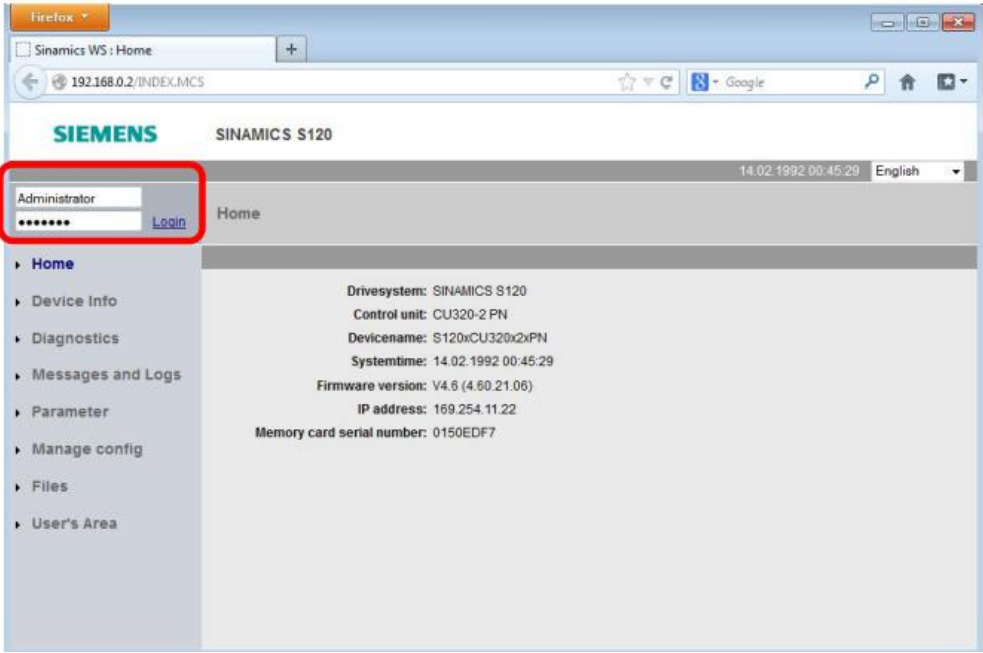
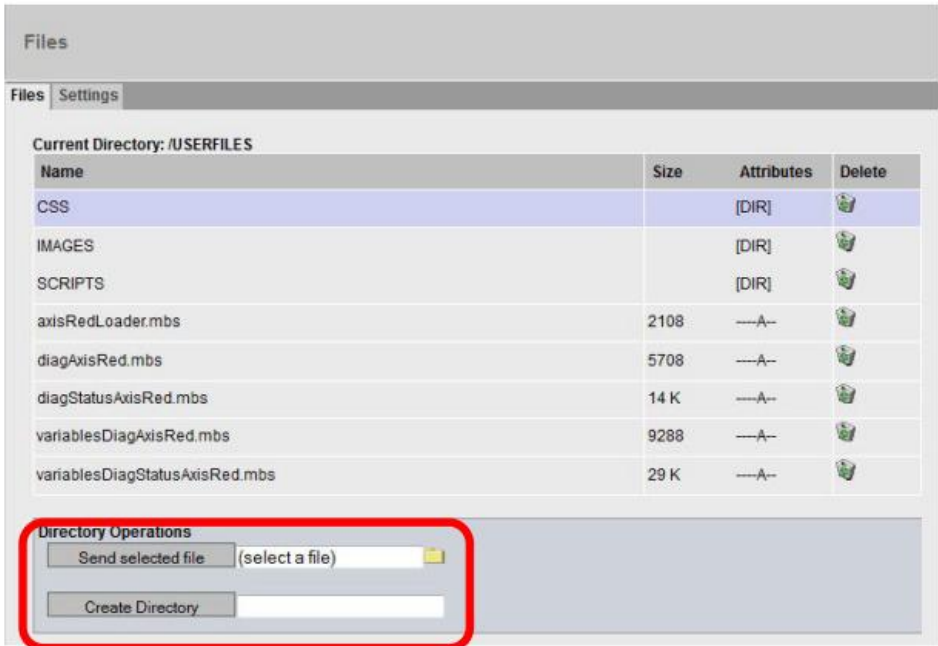
  return [barTorque];
}
function setupPage() {
  var gauges,
    bars,
    canvasControls = null;

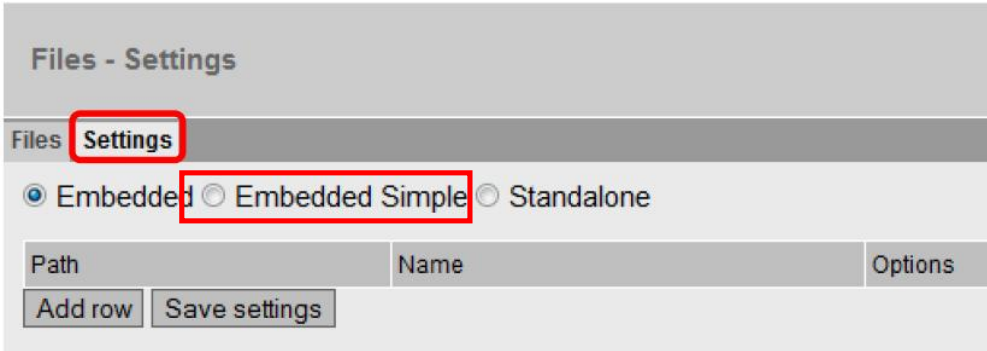
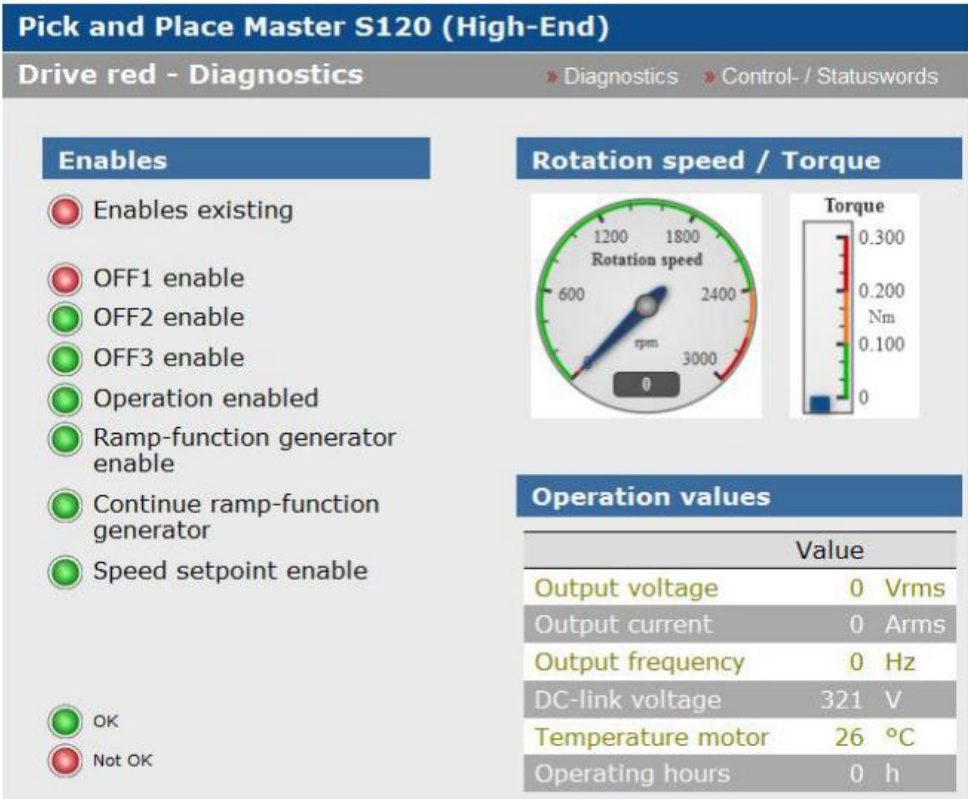
  /* Change to cyclic updating mode with a period of 3000ms */
  setInterval (function () { updateDocument.updateValuesEx(' readV.mbs',
canvasControls, null) }, 3000);
}

```

完成上述步骤后，需要在 S120 的 Web 中上传相关的网页文件。具体的步骤如下：

序号	步骤
6.	<p>首先把文件的后缀从 HTML 改为 mcs。</p>  <p>改为:</p>  <p>其他的 HTML 文件也需要修改为 MCS。</p>
7.	<p>转换 MCS 为 MBS 文件:</p> <p>使用 U7NW2XMX.exe, 如下图所示, 点击 START 按钮进行转换:</p>  <p>Totally integrated Automation</p>

序号	步骤
8.	<p>将使用的文件传送到 S120 的 CF 卡中： 在 IE 浏览器中输入 S120 的 IP 地址，例如 X127 接口为 169.254.11.22，并且输入用户名 Administrator 以及其密码：</p> 
9.	<p>在文件 files 界面进行文件的上传：</p> 

序号	步骤														
10	<p>指定自定义网页的首页，选择 Embedded simple 后保存即可：</p> 														
11	<p>随后即可查看自定义网页：</p>  <p><b>Enables</b></p> <ul style="list-style-type: none"> <li><input type="radio"/> Enables existing</li> <li><input type="radio"/> OFF1 enable</li> <li><input checked="" type="radio"/> OFF2 enable</li> <li><input checked="" type="radio"/> OFF3 enable</li> <li><input checked="" type="radio"/> Operation enabled</li> <li><input checked="" type="radio"/> Ramp-function generator enable</li> <li><input checked="" type="radio"/> Continue ramp-function generator</li> <li><input checked="" type="radio"/> Speed setpoint enable</li> </ul> <p><input checked="" type="radio"/> OK <input type="radio"/> Not OK</p> <p><b>Rotation speed / Torque</b></p> <p>Rotation speed: 0 rpm Torque: 0 Nm</p> <p><b>Operation values</b></p> <table border="1"> <thead> <tr> <th></th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Output voltage</td> <td>0 Vrms</td> </tr> <tr> <td>Output current</td> <td>0 Arms</td> </tr> <tr> <td>Output frequency</td> <td>0 Hz</td> </tr> <tr> <td>DC-link voltage</td> <td>321 V</td> </tr> <tr> <td>Temperature motor</td> <td>26 °C</td> </tr> <tr> <td>Operating hours</td> <td>0 h</td> </tr> </tbody> </table>		Value	Output voltage	0 Vrms	Output current	0 Arms	Output frequency	0 Hz	DC-link voltage	321 V	Temperature motor	26 °C	Operating hours	0 h
	Value														
Output voltage	0 Vrms														
Output current	0 Arms														
Output frequency	0 Hz														
DC-link voltage	321 V														
Temperature motor	26 °C														
Operating hours	0 h														
12	<p>可以参考附件的简单例子：HTML_READ.zi p</p>														

