OpenBAS MiPages for AI, ethwebJS v0.0.1 — Integration Guide (LT-6796)

Overview

Mircom's OpenBAS-NWK-ETH3 user-customizable web pages—**MiPages** for short—reinvent the earlier ETH3 custom web page features, now with a focus on leveraging AI.
A simple and intuitive HTML data-attribute notation allows custom web pages to read and write points from any controller in the system, in real-time.

With the provided example below, any displayed point value becomes clickable, so users can quickly override or set new values using pop-up forms. Dedicated form controls or buttons are also supported.

No JavaScript writing is required by you—the user—just copy and paste the examples as shown.

You can use this guide (the `.md` version) as a prompt attachment for your favourite AI tool.

VERY IMPORTANT: Escaping the Percent (%) Character

❗**ETH3 CGI Legacy Issue — You MUST escape all percent signs:**❗

Due to a legacy CGI feature in the OpenBAS-NWK-ETH3, **every `%` character in your MiPage file MUST be escaped by doubling it to** `%%` (two percent signs).
If you do not do this, your MiPage may not display or function correctly after uploading.

For example:

```
.overlay-img {
  max-width: 100%%;       /* Correct: double percent */
}
.left: 50%%;              /* Correct: double percent */
```

```
<!-- Wrong: <div style="left: 20%;"> ... -->
<div style="left: 20%%;"> ... </div>   <!-- Correct! -->
```

> **How to do this?**
> Before uploading, use your editor's **Find/Replace** feature to replace every `%` with `%%` in your HTML file.
>
> - In Notepad++:
>   - Press `Ctrl+H`, enter `%`, replace with `%%`, select "Replace All."

Installation

Add this script tag to every MiPage using these features:

```
<script src="../ewj0.0.1.js"></script>
```

Data Attribute Reference

Place these attributes on each element you want linked to an OpenBAS point:

Attribute	Required?	Description & Options
`data-openbas-opto-type`	✓ Yes	Point type: `AI`, `AO`, `BI`, `BO`, `ADF`, `ADI`, `ADB`, `RB`, `RF`, `RMT`, `TMR`
`data-openbas-opto-index`	✓ Yes	Index or channel of the point on the controller/module (`1`, `2`, `3`, etc.)
`data-openbas-target-controller`	✓ Yes	Target controller: `ETH3_LOCAL`, `ETH3_COM1_SLAVE`, `ETH3_COM2_SLAVE`, `SPI-NX_LOCAL`, `SPI-NX_COM2_SLAVE`
`data-openbas-device-address`	See note	**Required unless** `data-openbas-target-controller` is `ETH3_LOCAL` or `SPI-NX_LOCAL`. For those two controller types, the library automatically uses 254 or 255 and any user-entered value is ignored.
`data-openbas-target-ip`	Only if point is on another ETH3	Only needed if accessing a point on a *different* ETH3 than where this MiPage is hosted. Omit for local ETH3.
`data-openbas-display-precision`	No	Decimal places (`0`, `1`, `2`, etc.)
`data-openbas-display-unit`	No	Displayed units (`°C`, `%%`, `psi`, etc.)
`data-openbas-display-color`	No	CSS color for value text (e.g. `"red"`, `"#007bff"`)
`data-openbas-write-type`	✓ Yes, for write forms only	**Required only for write forms**: Set as `"true"`. **Do NOT include** this attribute for clickable points or read-only points.
`data-openbas-write-value`	Only for preset write buttons	Value to write for on/off or preset controls (e.g. `1`, `0`)

MiPages Capacity and Mapping

- **You may create up to 51 MiPages per ETH3**.

  - Filenames must be 8 characters or less (excluding `.htm`), e.g., `main.htm`, `zone1.htm`, or `wf-12.htm`.
  - Pages must be mapped in the `web_idx.ini` file to `wf-0.htm` through `wf-50.htm`.
    - `wf-0.htm` is the default "main" MiPage after login. Others become accessible via direct link or navigation.
    - After logging in via a web browser, users will be redirected to the file mapped to `wf-0.htm`.
  - The sample `web_idx.ini` file and tools can be found after installing System Design Studio at: `C:\OpenBAS\Documents\ETH3_WebPagesUser\_OpenBAS\`
- **You must update the `web_idx.ini` mapping file** every time you add a new MiPage.

Generating MiPages with AI

MiPages for AI is designed to work seamlessly with generative AI tools.
You can attach this markdown documentation, your building or equipment background image, and a points list—exported from the System Design Studio *Names* table or from a script file—to your AI prompt.
Let the AI read/generate a custom HTML dashboard, place the points, and structure the UI to your needs.

Example AI prompts and best practices:

- Example prompt:
  Following the attached MiPages for AI documentation, create a webpage with the attached AHU image as the main focus. Analyze the image, and for each element you find, overlay the appropriate point label and value. Use ETH3 com1 slave 6 for all points. Note that regarding airflow, the top right is the return air, top left exhaust, bottom left is fresh air, and bottom right is supply air.

  Do your best to place the elements, but make sure they do not overlap. I will fine-tune the locations.

  Make the page look modern and add common sections (specs/title/nav bar etc) you would expect from a BAS interface.

- We recommend asking AI to do its best to locate the point values over the image, but to make sure they're spread out enough to allow you to fine tune the locations afterwards.

- The template script in the documentation has a feature where you can click anywhere on the background image, and the coordinates (in %) will be logged to the browser's console for easy fine-tuning by the user. In our testing, this feature was always included in the generated web page, but if you have issues, you can add or fix it manually or ask the AI for help.

- Either list the points you want or attach a points list. You can export a full or partial points list from the System Design Studio Names table or copy from a script file.

- Specify which controller(s) you want points to be read from by including an IP address (if not from the ETH3 hosting the MiPages), the connection path (ETH3 com1/com2 etc.), and the address for slave controllers.

For graphic creation with AI:

- Integrators can use their existing plant graphics and tools, or simply photograph equipment. For rapid results, you can use AI to generate graphics from a hand drawing. (Results will vary — Gemini Flash 2.5

gave the best outcomes in our testing.)

Sample graphic prompt:
Based on the attached hand drawing, create a high quality, professional looking 3D graphical representation of an Air Handling Unit (AHU). The graphic should be a side view of a cross section of the AHU. As detailed in the hand drawing, there should be the following elements:

- *rectangular ducts forming a sideways "H" (light grey with 3D shadows)*
- *3x dampers (use the same icon for all 3)*
- *1x heating coil (show in red)*
- *1x cooling coil (show in blue)*
- *1x fan*
- *1x return air temp sensor probe*
- *1x supply air temp sensor probe*
- *1x static air pressure sensor probe (supply)*
- *All text and arrows in the drawing are for reference only. Do not include any text in the graphic. Use realistic icons for each element. Do not include any extra elements. All elements should be in the locations shown in the hand drawing. The background can be pure white.*

Example Mipage: (CSS, HTML, and JS)

> - Use the following as a base for your MiPages.
> - For every point you want to display and make clickable/editable, **copy and paste everything inside the `<div class="point-label">...</div>` block below and customize** for your point label and attributes.
> - See `C:\OpenBAS\Documents\ETH3_WebPagesUser\_OpenBAS\MiPageTp.htm` for easier copy and paste version of below example.

1. CSS (Place inside a `<style>` tag)

```
<style>
.graphic-container {
  position: relative;
  display: inline-block;
  /* max-width: 100%%; */
}
.overlay-img {
  display: block;
  max-width: 100%%;      /* Double percent REQUIRED */
  height: auto;
}
.point-label {
  position: absolute;
  color: rgba(0,90,200,0.8);
  background: #fff;
  padding: 2px 6px;
  border-radius: 4px;
  font-size: 13px;
  transform: translate(-50%%, -70%%);  /* Double percent REQUIRED */
  pointer-events: auto;
  white-space: nowrap;
  box-shadow: 0 2px 8px rgba(0,0,0,.2);
  cursor: pointer;
}
.popup-form {
  display: none;
  position: absolute;
  background: #fff;
  color: #222;
  padding: 12px 16px;
  border: 2px solid #007bff;
  border-radius: 8px;
  z-index: 100;
  box-shadow: 0 6px 32px rgba(0,0,0,0.2);
  min-width: 200px;
  font-weight: bold;
}
</style>
```

2. HTML (Inside `<body>`)

```
<!--
  Plant overview image.
  Images MUST be named -image01.jpg through -image50.jpg (with a leading dash).
  To use the imagecvt.exe utility, place your images with no leading dash (e.g. image01.jpg)—
  the utility will compress and rename them with the dash automatically for ETH3 compatibility.
-->
<div class="graphic-container">
  <img src="-image03.jpg" class="overlay-img" />
  <!--
    For all clickable/display points, use the class `point-label` on the value container.
    This class provides positioning, styling, and the clickable popup behavior.
    Copy and paste this entire div for each point you want to show on your page.
    Edit the visible label text, change left/top for position, and customize the data attributes for each OpenBAS point.
    Remember to use double percent (%%) in any inline styles!
    **Do NOT include data-openbas-write-type for clickable or read-only display points.**
  -->
  <div class="point-label" style="top:10%%; left:75%%;">
    BO_1 from ETH3 COM1 Slave Address 1
    <span class="point-value"
          data-openbas-opto-type="BO"
          data-openbas-opto-index="1"
          data-openbas-target-controller="ETH3_COM1_SLAVE"
          data-openbas-device-address="1"
          data-openbas-display-precision="0"
          >--</span>
  </div>
</div>
<!-- Only one my-popup is needed per page for all points/popups -->
<my-popup id="myPopup" class="popup-form" width="300px" height="80px"
          pointelemattr=""
          style="display: none;" onclose="hidePopup()"></my-popup>
```

3. JavaScript (At the end, within a `<script>` tag)

```
<script src="../ewj0.0.1.js"></script>
<script>
// You typically just need to copy/paste this code as is.
// AI should always use this code exactly as is.
// it enables popup editing for all your displayed points automatically.
document.addEventListener("DOMContentLoaded", function() {
  // Makes every .point-label clickable: opens popup for setpoint/override on any value
  document.querySelectorAll('.point-label').forEach(lbl => {
    lbl.addEventListener('click', function(e){
      const pv = lbl.querySelector('.point-value');
      if(pv) showOpenBasPopup(lbl, pv);
      e.stopPropagation();
    });
    const pv = lbl.querySelector('.point-value');
    if(pv) {
      pv.addEventListener('click', function(e){
        showOpenBasPopup(lbl, pv);
        e.stopPropagation();
      });
    }
  });
  document.addEventListener('click', function(e){
    var popup = document.getElementById('myPopup');
    if (popup.style.display !== 'none' && !popup.contains(e.target)) hidePopup();
  });
  // For overlay planning: log %% coordinates for easy positioning
  document.querySelectorAll('.overlay-img').forEach(img => {
    img.addEventListener('click', function(e){
      var rect = e.target.getBoundingClientRect();
      var xPercent = ((e.clientX - rect.left) / rect.width) * 100;
      var yPercent = ((e.clientY - rect.top) / rect.height) * 100;
      console.log("Top: " + yPercent.toFixed(2) + "%%%%, Left: " + xPercent.toFixed(2) +"%%%%");
    });
  });
});
function showOpenBasPopup(pointLabel, pointValueEl)
{
  var popup = document.getElementById('myPopup');
  popup.setAttribute("pointelemattr", JSON.stringify(pointValueEl.dataset));
  // Place popup near clicked element
  const rect = pointLabel.getBoundingClientRect();
  const scrollX = window.pageXOffset || document.documentElement.scrollLeft;
  const scrollY = window.pageYOffset || document.documentElement.scrollTop;
  popup.style.left = (rect.left + scrollX) + 'px';
  popup.style.top = (rect.bottom + scrollY + 8) + 'px';
  popup.style.display = 'block';
}
window.hidePopup = function() {
  document.getElementById('myPopup').style.display = "none";
};
</script>
```

Forms: User Setpoints and Overrides

Use this form example anywhere you want users to type or submit a setpoint/override.
You can include as many forms as needed—just copy and paste the example, and change the visible label and data attributes.
Be sure the inner `name` properties are exactly as shown.

Form requirements:

- The `<form>` must include all OpenBAS data attributes for the point, **and must include `data-openbas-write-type="true"`**.
- **Do NOT include `data-openbas-write-type` for points that are only intended for display/clickable popup.**
- Inner element `name` attributes (case-sensitive, required for automatic operation):
  - `<input name="openBasSetValue">` — value to write
  - `<button name="openBasSubmit">` — send/override
  - `<button name="openBasAuto">` — release override (optional)
  - `<div name="openBasResultIcon"></div>` — shows result/fail icon (optional)

Example:

```
<form
  data-openbas-opto-type="AO"
  data-openbas-opto-index="1"
  data-openbas-target-controller="SPI-NX_LOCAL"
  data-openbas-display-unit="%%"
  data-openbas-display-precision="0"
  data-openbas-write-type="true">
  Valve Setpoint
  <input type="number" name="openBasSetValue" />
  <button name="openBasSubmit">Override</button>
  <button name="openBasAuto">Auto</button>
  <div name="openBasResultIcon"></div>
</form>
```

Summary of required names inside the form:

Element Type	`name` value	Purpose	Required
`<input>`	`openBasSetValue`	User data entry (number, text, etc.)	Yes
`<button>`	`openBasSubmit`	Trigger to write/send value	Yes
`<button>`	`openBasAuto`	Return to auto mode/release override	No
`<div>`	`openBasResultIcon`	Shows result (success/fail icon/text)	No

Add a label or explanation in the form as shown, so users know what they are controlling.
The library automatically handles writing the new value and updating your page after submission.

Read-Only (Non-Clickable) Points

For points that should display a live value but **should not be clickable and should not allow pop-up editing or setpoint entry**, simply omit the `.point-label` class and use only the `.point-value` span with the appropriate data attributes.

Example:

```
<span class="point-value"
      data-openbas-opto-type="AI"
      data-openbas-opto-index="2"
      data-openbas-target-controller="SPI-NX_LOCAL"
      data-openbas-device-address="255"
      data-openbas-display-precision="1"
      data-openbas-display-unit="°C"
      >--</span>
```

- The value will update automatically with live data.
- The text (such as "Temperature:") and visual placement can be handled as you wish.
- **Do NOT use the `.point-label` class** or `data-openbas-write-type` attribute for these.

Example with a label:

```
<div>
  Temperature (Room 1):
  <span class="point-value"
        data-openbas-opto-type="AI"
        data-openbas-opto-index="2"
        data-openbas-target-controller="SPI-NX_LOCAL"
        data-openbas-device-address="255"
        data-openbas-display-precision="1"
        data-openbas-display-unit="°C"
        >--</span>
</div>
```

Image File Preparation and Naming

- All schematic/background images **must be named in the format image01.jpg (without a leading dash):** `image01.jpg` through `image50.jpg`.
- Place your properly named images in the same folder as the supplied `imagecvt.exe` utility, run it, and the utility will compress and rename them with the dash (e.g. `-image01.jpg`) for ETH3 compatibility. The resulting compressed images retain the `.jpg` extension.
- The `imagecvt.exe` utility and sample files are located at: `C:\OpenBAS\Documents\ETH3_WebPagesUser\_OpenBAS\` (after System Design Studio installation).
- Place your images in the same directory as the MiPage HTML (the `/_OpenBAS/` folder of the USB stick).

Loading MiPages to the ETH3

1. **Filename rules:**
   - HTML files must have at most 8 characters in their filename (plus `.htm`).
   - Examples: `main.htm`, `index.htm`, `zone1.htm`, `wf-12.htm`, etc.
2. **web_idx.ini mapping: You MUST update your `web_idx.ini` file** on the ETH3's USB memory stick to map your MiPage filenames.
   - The sample file is located (after System Design Studio install) at: `C:\OpenBAS\Documents\ETH3_WebPagesUser\_OpenBAS\web_idx.ini`
   - This controls which pages appear and which is the main user page.
   - Pages are mapped to `wf-0.htm` through `wf-50.htm` in `web_idx.ini`. The page mapped to `wf-0.htm` is the default after login.
3. **Uploading files:**
   You can use either option below:
   - **A. Upload through the ETH3 web form:**
     Go to `http://<ETH3_IP>/protect/usbupl.htm` and upload your files; they will go directly into the `/_OpenBAS/` folder on the attached flash drive.
   - **B. USB direct copy:**
     Remove the USB stick from ETH3, plug it into your PC, open the `/_OpenBAS/` folder, and copy files over directly there.

Accessing and Navigating MiPages

- Browse or link directly to your HTML file—such as `/protect/index.htm`, `/protect/wf-xx.htm`, etc.
- When visiting the ETH3 IP directly (e.g. `http://<ETH3_IP>/`), the only link shown is “Go to main page,” which takes you to `wf-0.htm` (the file mapped in `web_idx.ini`). All other files (`wf-xx.htm`) must be accessed by direct navigation or links you create.

Polling (Live Updates)

- All points automatically refresh with live controller data at regular intervals (default: every 10 seconds).

- All `.point-label` points will become automatically clickable for Sets and Overrides.

- Adjust polling speed (optional):

  ```
  window.instOpenBAS.init({ pollInterval: 5000 }); // ms
  ```

- For large pages, set polling to 10–15 seconds for best performance.

FAQ

Why do I need to use double percent signs (%%)?
ETH3's legacy CGI system treats single `%` characters as special codes. Unescaped percent signs can break your HTML/CSS/JS. Escape every `%` as `%%` before uploading.

How do I label my points?
Just type your label text in each `.point-label` div before the `<span class="point-value">`.
Example:

```
<div class="point-label" style="...">Boiler Supply <span ...>--</span></div>
```

How do I position my points?
Click your image in the browser; the JS will log top/left %% coordinates in the console. Use these for the style in each `.point-label`.

Why must I use `.point-label`?
It enables correct popup events, layout, and styling for all your point displays.