Setting Up Sphinx for 3DPrintIoT Documentation

3DPrintIoT Team

May 2025

Contents

1 Introduction

This guide outlines the process of setting up Sphinx, a powerful documentation generator, to create offline HTML documentation for the 3DPrintIoT repository. The repository hosts manufacturing guides, schematics, diagrams, 3D print files (e.g., STL/OBJ), and software build instructions for IoT technology. Sphinx enables structured, professional documentation that can be accessed offline, ensuring accessibility for developers, engineers, and enthusiasts.

2 Prerequisites

Before setting up Sphinx, ensure the following are installed:

- **Python 3.8 or higher**: Required to run Sphinx.
- **pip**: Python package manager for installing Sphinx and dependencies.
- **Git**: To clone and manage the 3DPrintIoT repository.
- **A text editor**: Such as VS Code or PyCharm for editing documentation files.
- **(Optional) LaTeX distribution**: For generating PDF outputs alongside HTML.

Verify Python installation:

```
python3 --version
```

Install pip if not present:

```
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
python3 get-pip.py
```

3 Installing Sphinx

Install Sphinx and the reStructuredText (reST) parser:

```
pip install sphinx sphinx-rtd-theme
```

The `sphinx-rtd-theme` provides a clean, professional look for the HTML output, suitable for technical documentation.

4 Setting Up the Sphinx Project

1. **Create a documentation directory** in the 3DPrintIoT repository:

```
mkdir -p 3DPrintIoT/docs/source
cd 3DPrintIoT/docs/source
```

2. **Initialize Sphinx**:

```
sphinx-quickstart
```

Answer the prompts:

- Root path: Use the default (`.`).
- Project name: `3DPrintIoT Documentation`.
- Author: `3DPrintIoT Team`.
- Project release: `1.0`.
- Language: `en`.

This generates a `conf.py` file and an `index.rst` file.

5 Configuring Sphinx

Edit `docs/source/conf.py` to customize the project:

- **Set the theme**: Uncomment and set:

```
html_theme = 'sphinx_rtd_theme'
```

- **Enable extensions**: Add to the extensions list for enhanced functionality:

```
extensions = [
    'sphinx.ext.autodoc',
    'sphinx.ext.napoleon',
    'sphinx.ext.viewcode',
    'sphinx.ext.imgmath'
]
```

- **Add static paths**: For images and 3D print files:

```
html_static_path = ['_static']
```

Create a `_static` directory for assets:

```
mkdir docs/source/_static
```

6 Structuring Documentation

Organize content in `docs/source`:

- **Manufacturing Guides**: Create `guides.rst` for 3D printing processes and material selection.

- **Schematics**: Add `schematics.rst` with links to circuit diagrams (e.g., PNG files in `_static`).
- **Diagrams**: Include `diagrams.rst` for system architectures and embed 3D model previews using `:image:`.
- **3D Print Files**: Store STL/OBJ files in `_static/3dmodels` and reference them in `3dprints.rst`.
- **Software Guides**: Create `software.rst` for IoT firmware and build instructions.

Example `index.rst`:

```
Welcome to 3DPrintIoT Documentation
===================================
.. toctree::
   :maxdepth: 2

   guides
   schematics
   diagrams
   3dprints
   software
```

7 Adding Content

Write documentation in reStructuredText. Example for `guides.rst`:

```
Manufacturing Guides
====================
This section covers 3D printing processes for IoT devices.

.. image:: _static/printer_setup.png
   :width: 400px
   :alt: 3D Printer Setup
```

For 3D print files:

```
3D Print Files
==============
Download STL files for IoT device enclosures:

* `Sensor Case <_static/3dmodels/sensor_case.stl>`_
```

Place images and STL files in `docs/source/_static`.

8 Building Offline HTML Documentation

Generate HTML files:

```
cd docs
make html
```

Output is in `docs/build/html`. Open `index.html` in a browser to view the documentation offline.

9 Verifying and Distributing

- **Test locally**: Ensure all links and images render correctly.
- **Include in repository**: Commit `docs/source` to 3DPrintIoT.
- **Distribute**: Share the `docs/build/html` folder or host it on a static server.

10 Troubleshooting

- **Missing dependencies**: Re-run `pip install sphinx sphinx-rtd-theme`.
- **Broken links**: Verify file paths in `_static`.
- **Rendering issues**: Check reST syntax using `sphinx-build -b html source build/html`.