
pax-devices Documentation
Documentation
Release 1.0 2020.02.22

Calin Radoni

Mar 20, 2020

Contents

1	pax-Gateway	3
2	pax-DLED	5
3	pax-DLEDGrid	7
4	Development for ESP32-based pax-devices	9
4.1	Compiling with esp-idf	9
5	Development of this documentation	13
5.1	Project configuration	13
5.2	Read the Docs Sphinx Theme	13
5.3	Building from terminal	14
5.4	Building from Visual Studio Code	14
5.5	Preview from Visual Studio Code	14
6	ESP32DLEDController	17
7	ESP32RMT	19
8	ESP32SimpleOTA	21
9	ESP32Timers	23
10	Indices and tables	25

This is the documentation for pax-devices.

<i>pax-Gateway</i>	<i>pax-DLED</i>	<i>pax-DLEDGrid</i>	
<i>ESP32DLEDController</i>	<i>ESP32RMT</i>	<i>ESP32SimpleOTA</i>	<i>ESP32Timers</i>

CHAPTER 1

pax-Gateway

CHAPTER 2

pax-DLED

CHAPTER 3

pax-DLEDGrid

Development for ESP32-based pax-devices

This document specifies the development configuration ESP32 based pax-devices

4.1 Compiling with esp-idf

I am compiling in Linux, Debian 10 to be precise.

4.1.1 Locations for code and application

- ESP-IDF is installed in the default path, `~/esp/esp-idf`.
- The `pax-devices` project is in the `CFG_Projects_Path` directory
- The Visual Studio Code workspace is saved with the name `CFG_Projects_Path/pax-devices/pax-devices.code-workspace`
- Additional ESP-IDF components are in `CFG_CodeLib_Path/ESP32` and `CFG_CodeLib_Path/Generic` directories

Create the `~/vscode-pax-devices` with the following content:

```
#!/bin/bash

ProjectPath="CFG_Projects_Path/pax-devices"
ProjectName="pax-devices"

export CodeLib_PATH=CFG_CodeLib_Path

cd $ProjectPath
. ~/esp/esp-idf/export.sh
code $ProjectPath/$ProjectName.code-workspace
```

Source that file with `. vscode-pax-devices` and it will launch Visual Studio Code and a terminal. From **that** terminal, to compile, flash and test (for example `pax-Gateway`), use:

```
cd CFG_Projects_Path/pax-Gateway/SW
idf.py menuconfig
idf.py build
idf.py flash
idf.py monitor
```

4.1.2 Example directory structure

- ESP-IDF is installed in the default path, `~/esp/esp-idf`.
- The `pax-devices` project is in the `/data/Projects/pax-devices` directory
- The Visual Studio Code workspace is saved with the name `CFG_Projects_Path/pax-devices/pax-devices.code-workspace`
- **Additional ESP-IDF components are in `/data/Projects/CodeLibrary` directories:**
 - Debouncer is located in `/data/Projects/CodeLibrary/Generic`
 - ESP32DLEDController, ESP32RMT, ... `esp-idf` components are located in `/data/Projects/CodeLibrary/ESP32`

The `/data/Projects/pax-devices/pax-devices.code-workspace` file is:

```
{
  "folders": [
    {
      "path": "."
    },
    {
      "path": "/data/Projects/pax-Gateway"
    },
    {
      "path": "/data/Projects/pax-DLED"
    },
    {
      "path": "/data/Projects/pax-DLEDGrid"
    },
    {
      "path": "/data/Projects/CodeLibrary"
    }
  ],
  "extensions": {
    "recommendations": [
      "ms-vscode.cpptools",
      "editorconfig.editorconfig",
      "lxtudio.restructuredtext",
      "ms-python.python",
      "gruntfuggly.todo-tree"
    ]
  }
}
```

The `vscode-pax-devices` file is:

```
#!/bin/bash
ProjectPath="/data/Projects/pax-devices"
```

(continues on next page)

(continued from previous page)

```
ProjectName="pax-devices"
export CodeLib_PATH=/data/Projects/CodeLibrary
cd $ProjectPath
. ~/esp/esp-idf/export.sh
code $ProjectPath/$ProjectName.code-workspace
```

Development of this documentation

This documentation is created with `Sphinx` .

In Debian install `Sphinx` with `sudo apt install python3-sphinx` .

5.1 Project configuration

In the project's directory create the `docs` directory and run the `sphinx-quickstart` command inside that directory:

```
mkdir docs
cd docs
sphinx-quickstart
```

Except for project's **name**, **author** and **version** I did not change implicit values.

5.2 Read the Docs Sphinx Theme

To add the `Read the Docs Sphinx Theme` go to the `docs` directory and run these commands:

```
mkdir /tmp/sphinx-rtd
git clone https://github.com/readthedocs/sphinx_rtd_theme.git /tmp/sphinx-rtd
cp -r /tmp/sphinx-rtd/sphinx_rtd_theme _themes/sphinx_rtd_theme
rm -rf /tmp/sphinx-rtd
```

to download it and copy the theme inside this project's `_themes` directory.

To use the theme edit the `conf.py` to replace default theme:

```
html_theme = 'alabaster'
```

with the new one:

```
html_theme = 'sphinx_rtd_theme'  
html_theme_path = ['_themes', ]
```

See `sphinx_rtd_theme` for it's license.

5.3 Building from terminal

To build it run `make html` in project's docs directory.

5.4 Building from Visual Studio Code

In project's root directory create the `.vscode/tasks.json` file with the following content:

```
{  
  "version": "2.0.0",  
  "linux": {  
    "type": "shell",  
    "options": {  
      "cwd": "${workspaceFolder}"  
    },  
    "presentation": {  
      "echo": true,  
      "reveal": "always",  
      "focus": true,  
      "panel": "shared",  
      "showReuseMessage": false,  
      "clear": false  
    },  
  },  
  "problemMatcher": [],  
  "tasks": [  
    {  
      "label": "Build HTML",  
      "command": "make",  
      "args": [ "html" ],  
      "group": "build",  
      "options": {  
        "cwd": "${workspaceFolder}/docs"  
      }  
    }  
  ]  
}
```

To create HTML files use `Ctrl+Shift+B` and select *Build HTML pax-devices* option.

5.5 Preview from Visual Studio Code

In Visual Studio Code install `lextrud.io.restructuredtext` extension ([GitHub repository](#) and [Documentation](#)).

In Debian install `rstcheck` linter with `sudo python3 -m pip install rstcheck`.

Create `.vscode/settings.json` with the following content:

```
{
  "restructuredtext.builtDocumentationPath": "${workspaceRoot}/docs/_build/html",
  "restructuredtext.confPath" :           "${workspaceFolder}/docs",
  "restructuredtext.updateOnTextChanged" : "false",
  "restructuredtext.updateDelay":         1000,
  "restructuredtext.sphinxBuildPath":     "/usr/bin/sphinx-build",
  "restructuredtext.linter.executablePath": "/usr/local/bin/rstcheck"
}
```

Use Ctrl+Shift+R for preview.

CHAPTER 6

ESP32DLEDController

CHAPTER 7

ESP32RMT

CHAPTER 8

ESP32SimpleOTA

CHAPTER 9

ESP32Timers

CHAPTER 10

Indices and tables

- `genindex`
- `search`