ESP8266 Libraries

Nefastor Online libraries documentation 03/08/2016

DOCUMENT STATUS : DRAFT IN PROGRESS – UNFIT FOR USE

This document and the code it pertains to are stored in the GitHub repository Nefastor/ESP8266, where later versions may be found.

This document assumes the libraries are used as part of the template project stored in the same repository.

For more information, head over to www.nefastor.com.

1. GPIO

1.1. Introduction

This library provides an API for General Purpose Input Output (GPIO) pins.

1.1.1. Origin of the library

The GPIO library originates with the official Espressif SDK. Several source files were merged to provide a single library for controlling the ESP8266 I/O pins. This includes:

- Espressif's "gpio.h" and "gpio.c" files
- Espressif's "gpio16.h" and "gpio16.c" files

GPIO 16 is very different form GPIO 0 through 15 in that it is actually part of a different internal peripheral and is controlled through different registers. It also has reduced functionality, and notably lacks interrupt support. I consider

Relatively few modifications to the actual code were necessary. The most important modification is FreeRTOS support.

1.2. Using the library

1.2.1. Building

By default, the library's source code is part of the ESP8266 template project and will compile and link with your code. All that may be required for building is to #include "gpio.h" in your own source files.

1.2.2. Using GPIO

Typical GPIO programming requires an initialization step.

1.3. API

1.3.1. gpio_config

1.3.2. gpio_output_conf

1.3.3. gpio_input_get

Returns the current state of pins GPIO 0 to 15 as an integer.

You will need to use bit masks to get the value of individual pins or sets of pins.

1.3.4. gpio_intr_handler_register

Status: validated.

1.3.5. gpio_pin_wakeup_enable

Status: untested.

1.3.6. gpio_pin_wakeup_disable

1.3.7. gpio_pin_intr_state_set

1.3.8. gpio16_output_conf

1.3.9. gpio16_output_set

1.3.10. gpio16_input_conf

1.3.11. gpio16_input_get