

SmartHomeLED

Generated by Doxygen 1.8.17



<b>1 SmartLed</b>	<b>1</b>
<b>2 Class Index</b>	<b>3</b>
2.1 Class List	3
<b>3 File Index</b>	<b>5</b>
3.1 File List	5
<b>4 Class Documentation</b>	<b>7</b>
4.1 channel Struct Reference	7
4.1.1 Member Data Documentation	7
4.1.1.1 b	7
4.1.1.2 brightness	7
4.1.1.3 g	7
4.1.1.4 name	8
4.1.1.5 r	8
4.2 gatts_profile_inst Struct Reference	8
4.2.1 Member Data Documentation	8
4.2.1.1 app_id	8
4.2.1.2 char_handle	8
4.2.1.3 char_uuid	9
4.2.1.4 conn_id	9
4.2.1.5 descr_handle	9
4.2.1.6 descr_uuid	9
4.2.1.7 gatts_cb	9
4.2.1.8 gatts_if	9
4.2.1.9 perm	9
4.2.1.10 property	9
4.2.1.11 service_handle	10
4.2.1.12 service_id	10
4.3 Node Struct Reference	10
4.3.1 Detailed Description	10
4.3.2 Member Data Documentation	10
4.3.2.1 next	10
4.3.2.2 schedule	11
4.4 prepare_type_env_t Struct Reference	11
4.4.1 Member Data Documentation	11
4.4.1.1 prepare_buf	11
4.4.1.2 prepare_len	11
4.5 Schedule_Object Struct Reference	11
4.5.1 Detailed Description	12
4.5.2 Member Data Documentation	12
4.5.2.1 b	12

4.5.2.2 brightness	12
4.5.2.3 dawn	12
4.5.2.4 duration	12
4.5.2.5 dusk	13
4.5.2.6 enabled	13
4.5.2.7 g	13
4.5.2.8 ID	13
4.5.2.9 isRGB	13
4.5.2.10 name	13
4.5.2.11 r	13
4.5.2.12 repeat_mask	14
4.5.2.13 repeat_time	14
4.5.2.14 start	14
<b>5 File Documentation</b>	<b>15</b>
5.1 include/ArduinoJson-v6.14.1.h File Reference	15
5.2 include/bleSL.h File Reference	15
5.2.1 Detailed Description	15
5.2.2 Function Documentation	15
5.2.2.1 Init_Bluetooth()	15
5.3 include/dawndusk.h File Reference	16
5.3.1 Detailed Description	16
5.3.2 Macro Definition Documentation	16
5.3.2.1 distSun	17
5.3.2.2 pi	17
5.3.2.3 planeDist	17
5.3.2.4 radius	17
5.3.3 Function Documentation	17
5.3.3.1 dawnCalc()	17
5.3.3.2 degToRad()	18
5.3.3.3 duskCalc()	18
5.3.3.4 sign()	19
5.4 include/decode_bluetooth.h File Reference	19
5.4.1 Detailed Description	20
5.4.2 Macro Definition Documentation	20
5.4.2.1 DECODE_BLUETOOTH_H	20
5.4.3 Function Documentation	20
5.4.3.1 decode_ble_delete()	20
5.4.3.2 decode_ble_direct()	20
5.4.3.3 decode_ble_schedule()	21
5.4.3.4 decode_ble_schedule_name()	21
5.4.3.5 decode_ble_time()	22

5.4.3.6 get_Int32()	22
5.4.3.7 set_schedule_read()	22
5.4.3.8 start_schedule_read()	23
5.5 include/espntp.h File Reference	23
5.5.1 Detailed Description	23
5.5.2 Function Documentation	24
5.5.2.1 obtain_time()	24
5.5.2.2 set_time()	24
5.6 include/gatts_table_creat_demo.h File Reference	24
5.6.1 Enumeration Type Documentation	24
5.6.1.1 anonymous enum	24
5.7 include/http.h File Reference	25
5.7.1 Detailed Description	25
5.7.2 Function Documentation	26
5.7.2.1 direct_control_post_handler()	26
5.7.2.2 favicon_ico_get_handler()	26
5.7.2.3 homepage_handler()	26
5.7.2.4 init_http()	27
5.7.2.5 sch_data_post_handler()	27
5.7.2.6 schedule_post_handler()	28
5.7.2.7 schedules_handler()	28
5.7.2.8 schTokenProcess()	28
5.7.2.9 scripts_handler()	29
5.7.2.10 styles_handler()	29
5.7.2.11 time_post_handler()	29
5.8 include/led.h File Reference	30
5.8.1 Detailed Description	30
5.8.2 Macro Definition Documentation	31
5.8.2.1 GPIO_CHANNEL_0	31
5.8.2.2 GPIO_CHANNEL_1	31
5.8.2.3 GPIO_CHANNEL_2	31
5.8.2.4 GPIO_CHANNEL_3	31
5.8.2.5 GPIO_CHANNEL_4	31
5.8.2.6 GPIO_CHANNEL_5	31
5.8.2.7 NUM_CHANNELS	31
5.8.3 Function Documentation	31
5.8.3.1 channel_off()	31
5.8.3.2 channel_on()	32
5.8.3.3 clear_shutdown()	32
5.8.3.4 init_channels()	32
5.8.3.5 set_color()	32
5.8.3.6 shutdown_outputs()	33

5.9 include/main.h File Reference	33
5.9.1 Detailed Description	33
5.10 include/measurement.h File Reference	33
5.10.1 Detailed Description	34
5.10.2 Function Documentation	34
5.10.2.1 clearFaults()	34
5.10.2.2 init_oc()	35
5.10.2.3 isCurrentFault()	35
5.10.2.4 isVoltageFault()	35
5.10.2.5 set_current_level()	35
5.10.2.6 set_voltage_level()	36
5.11 include/memory.h File Reference	36
5.11.1 Detailed Description	37
5.11.2 Macro Definition Documentation	37
5.11.2.1 SETTINGS_BUFFER_SIZE	37
5.11.3 Function Documentation	38
5.11.3.1 clear_schedule_data()	38
5.11.3.2 clear_setting_data()	38
5.11.3.3 get_setting_byte()	38
5.11.3.4 get_setting_double()	39
5.11.3.5 get_setting_int()	39
5.11.3.6 get_setting_string()	39
5.11.3.7 init_memory()	40
5.11.3.8 recall_schedules()	40
5.11.3.9 store_schedules()	40
5.11.3.10 store_setting_byte()	40
5.11.3.11 store_setting_double()	41
5.11.3.12 store_setting_int()	41
5.11.3.13 store_setting_string()	42
5.12 include/pin_defs.h File Reference	42
5.12.1 Detailed Description	42
5.12.2 Macro Definition Documentation	43
5.12.2.1 CH3_HIGH	43
5.12.2.2 CH3_LOW	43
5.12.2.3 DAC1	43
5.12.2.4 DAC2	43
5.12.2.5 OC_ALERT	43
5.12.2.6 OC_ENABLE	43
5.12.2.7 OC_LATCH	43
5.12.2.8 OC_LIMIT	44
5.13 include/rtcdefine.h File Reference	44
5.13.1 Detailed Description	45

5.13.2 Macro Definition Documentation	45
5.13.2.1 CLRRAM	46
5.13.2.2 CONTROL	46
5.13.2.3 EEREAD	46
5.13.2.4 EEWRDI	46
5.13.2.5 EEWREN	46
5.13.2.6 EEWRITE	46
5.13.2.7 H12	46
5.13.2.8 HSPI_CLK	46
5.13.2.9 HSPI_CS	47
5.13.2.10 HSPI_MISO	47
5.13.2.11 HSPI_MOSI	47
5.13.2.12 IDREAD	47
5.13.2.13 IDWRITE	47
5.13.2.14 LPYR	47
5.13.2.15 OSCRUN	47
5.13.2.16 OSCTRIM	47
5.13.2.17 OUT	48
5.13.2.18 PM	48
5.13.2.19 PWRDNDATE	48
5.13.2.20 PWRDNHOUR	48
5.13.2.21 PWRDNMIN	48
5.13.2.22 PWRDNMONTH	48
5.13.2.23 PWRFAIL	48
5.13.2.24 RTC_READ	48
5.13.2.25 RTC_TAG	49
5.13.2.26 RTC_UNLOCK	49
5.13.2.27 RTC_WRITE	49
5.13.2.28 RTCDATE	49
5.13.2.29 RTCHOUR	49
5.13.2.30 RTCHSEC	49
5.13.2.31 RTCMIN	49
5.13.2.32 RTCMTH	49
5.13.2.33 RTCSEC	50
5.13.2.34 RTCWKDAY	50
5.13.2.35 RTCYEAR	50
5.13.2.36 SQWEN	50
5.13.2.37 SRREAD	50
5.13.2.38 SSWRITE	50
5.13.2.39 ST	50
5.13.2.40 TRIMSIGN	50
5.13.2.41 VBATEN	51

5.13.3 Function Documentation	51
5.13.3.1 getTime()	51
5.13.3.2 RTCHandler()	51
5.13.3.3 setTime()	51
5.13.3.4 ST_StartRTCHandler()	52
5.13.4 Variable Documentation	52
5.13.4.1 currTime	52
5.13.4.2 RTC_Handle	52
5.14 include/schedule_object.h File Reference	52
5.14.1 Detailed Description	53
5.14.2 Typedef Documentation	53
5.14.2.1 List	53
5.14.2.2 schedule_object	53
5.15 include/scheduler.h File Reference	54
5.15.1 Detailed Description	55
5.15.2 Function Documentation	55
5.15.2.1 create_schedule()	55
5.15.2.2 delete_all_schedules()	55
5.15.2.3 delete_schedule_by_id()	56
5.15.2.4 delete_schedule_by_name()	56
5.15.2.5 disable_all_schedules()	56
5.15.2.6 disable_schedule_by_id()	57
5.15.2.7 disable_schedule_by_name()	57
5.15.2.8 enable_all_schedules()	57
5.15.2.9 enable_schedule_by_id()	58
5.15.2.10 enable_schedule_by_name()	58
5.15.2.11 get_schedule()	58
5.15.2.12 get_schedule_names()	59
5.15.2.13 init_schedule()	59
5.15.3 Variable Documentation	59
5.15.3.1 schedules	59
5.16 include/wifi.h File Reference	60
5.16.1 Detailed Description	60
5.16.2 Macro Definition Documentation	60
5.16.2.1 MIN	60
5.16.3 Function Documentation	60
5.16.3.1 wifi_init_softap()	61
5.16.3.2 wifi_init_sta()	61
5.17 README.md File Reference	61
5.18 src/bleSL.cpp File Reference	61
5.18.1 Detailed Description	62
5.18.2 Macro Definition Documentation	62



5.18.2.1 ADV_CONFIG_FLAG . . . . .	62
5.18.2.2 CHAR_DECLARATION_SIZE . . . . .	62
5.18.2.3 CONFIG_SET_RAW_ADV_DATA . . . . .	63
5.18.2.4 ESP_APP_ID . . . . .	63
5.18.2.5 GATTS_DEMO_CHAR_VAL_LEN_MAX . . . . .	63
5.18.2.6 GATTS_TABLE_TAG . . . . .	63
5.18.2.7 PREPARE_BUF_MAX_SIZE . . . . .	63
5.18.2.8 PROFILE_APP_IDX . . . . .	63
5.18.2.9 PROFILE_NUM . . . . .	63
5.18.2.10 SAMPLE_DEVICE_NAME . . . . .	63
5.18.2.11 SCAN_RSP_CONFIG_FLAG . . . . .	64
5.18.2.12 SVC_INST_ID . . . . .	64
5.18.3 Function Documentation . . . . .	64
5.18.3.1 example_exec_write_event_env() . . . . .	64
5.18.3.2 example_prepare_write_event_env() . . . . .	64
5.18.3.3 Init_Bluetooth() . . . . .	64
5.18.4 Variable Documentation . . . . .	64
5.18.4.1 heart_rate_handle_table . . . . .	64
5.19 src/dawndusk.cpp File Reference . . . . .	65
5.19.1 Detailed Description . . . . .	65
5.19.2 Function Documentation . . . . .	65
5.19.2.1 dawnCalc() . . . . .	65
5.19.2.2 degToRad() . . . . .	66
5.19.2.3 duskCalc() . . . . .	66
5.19.2.4 sign() . . . . .	67
5.20 src/decode_bluetooth.cpp File Reference . . . . .	67
5.20.1 Detailed Description . . . . .	68
5.20.2 Function Documentation . . . . .	68
5.20.2.1 decode_ble_delete() . . . . .	68
5.20.2.2 decode_ble_direct() . . . . .	69
5.20.2.3 decode_ble_schedule() . . . . .	69
5.20.2.4 decode_ble_schedule_name() . . . . .	69
5.20.2.5 decode_ble_time() . . . . .	70
5.20.2.6 get_Int32() . . . . .	70
5.20.2.7 set_schedule_read() . . . . .	71
5.20.2.8 start_schedule_read() . . . . .	71
5.20.3 Variable Documentation . . . . .	71
5.20.3.1 done_string . . . . .	71
5.20.3.2 it . . . . .	72
5.20.3.3 saved_channel_num . . . . .	72
5.20.3.4 saved_len . . . . .	72
5.20.3.5 saved_name . . . . .	72

5.20.3.6 schedule_name . . . . .	72
5.20.3.7 schedule_value . . . . .	72
5.20.3.8 schedules . . . . .	72
5.20.3.9 state . . . . .	73
5.21 src/espsntp.cpp File Reference . . . . .	73
5.21.1 Detailed Description . . . . .	73
5.21.2 Function Documentation . . . . .	73
5.21.2.1 obtain_time() . . . . .	73
5.21.2.2 set_time() . . . . .	74
5.21.3 Variable Documentation . . . . .	74
5.21.3.1 CONNECTED_BIT . . . . .	74
5.22 src/http.cpp File Reference . . . . .	74
5.22.1 Detailed Description . . . . .	75
5.22.2 Macro Definition Documentation . . . . .	75
5.22.2.1 DELIMITER . . . . .	76
5.22.3 Function Documentation . . . . .	76
5.22.3.1 direct_control_post_handler() . . . . .	76
5.22.3.2 favicon_ico_get_handler() . . . . .	76
5.22.3.3 homepage_handler() . . . . .	76
5.22.3.4 init_http() . . . . .	78
5.22.3.5 sch_data_post_handler() . . . . .	78
5.22.3.6 schedule_post_handler() . . . . .	79
5.22.3.7 schedules_handler() . . . . .	79
5.22.3.8 schTokenProcess() . . . . .	79
5.22.3.9 scripts_handler() . . . . .	80
5.22.3.10 styles_handler() . . . . .	80
5.22.3.11 time_post_handler() . . . . .	80
5.23 src/led.cpp File Reference . . . . .	81
5.23.1 Detailed Description . . . . .	81
5.23.2 Function Documentation . . . . .	82
5.23.2.1 channel_off() . . . . .	82
5.23.2.2 channel_on() . . . . .	83
5.23.2.3 clear_shutdown() . . . . .	83
5.23.2.4 init_channels() . . . . .	83
5.23.2.5 set_color() . . . . .	83
5.23.2.6 shutdown_outputs() . . . . .	84
5.23.3 Variable Documentation . . . . .	84
5.23.3.1 shutdown_status . . . . .	84
5.24 src/main.cpp File Reference . . . . .	84
5.24.1 Detailed Description . . . . .	85
5.24.2 Function Documentation . . . . .	85
5.24.2.1 app_main() . . . . .	85

5.25 src/measurement.cpp File Reference	85
5.25.1 Detailed Description	86
5.25.2 Macro Definition Documentation	86
5.25.2.1 TAG	86
5.25.3 Function Documentation	86
5.25.3.1 clearFaults()	87
5.25.3.2 init_oc()	87
5.25.3.3 isCurrentFault()	87
5.25.3.4 isVoltageFault()	87
5.25.3.5 set_current_level()	87
5.25.3.6 set_voltage_level()	88
5.25.4 Variable Documentation	88
5.25.4.1 currentFault	88
5.25.4.2 voltageFault	88
5.26 src/memory.cpp File Reference	88
5.26.1 Detailed Description	89
5.26.2 Function Documentation	90
5.26.2.1 clear_schedule_data()	90
5.26.2.2 clear_setting_data()	90
5.26.2.3 get_setting_byte()	90
5.26.2.4 get_setting_double()	91
5.26.2.5 get_setting_int()	91
5.26.2.6 get_setting_string()	91
5.26.2.7 init_memory()	92
5.26.2.8 init_spiffs()	92
5.26.2.9 read_settings_to_buffer()	92
5.26.2.10 recall_schedules()	92
5.26.2.11 store_schedules()	93
5.26.2.12 store_setting_byte()	93
5.26.2.13 store_setting_double()	93
5.26.2.14 store_setting_int()	94
5.26.2.15 store_setting_string()	94
5.26.3 Variable Documentation	94
5.26.3.1 bSPIFFS	95
5.26.3.2 readNeeded	95
5.26.3.3 settingsString	95
5.27 src rtc.cpp File Reference	95
5.27.1 Detailed Description	95
5.27.2 Function Documentation	96
5.27.2.1 getTime()	96
5.27.2.2 readData()	96
5.27.2.3 rtc_config()	96

5.27.2.4 RTCHandler()	96
5.27.2.5 setTime()	97
5.27.2.6 ST_StartRTCHandler()	97
5.27.2.7 writeData()	97
5.27.3 Variable Documentation	97
5.27.3.1 rtc	97
5.28 src/scheduler.cpp File Reference	98
5.28.1 Detailed Description	99
5.28.2 Macro Definition Documentation	99
5.28.2.1 MAX	99
5.28.2.2 MIN	99
5.28.3 Function Documentation	99
5.28.3.1 create_schedule()	99
5.28.3.2 delete_all_schedules()	100
5.28.3.3 delete_schedule_by_id()	100
5.28.3.4 delete_schedule_by_name()	100
5.28.3.5 disable_all_schedules()	101
5.28.3.6 disable_schedule_by_id()	101
5.28.3.7 disable_schedule_by_name()	101
5.28.3.8 enable_all_schedules()	102
5.28.3.9 enable_schedule_by_id()	102
5.28.3.10 enable_schedule_by_name()	103
5.28.3.11 get_schedule()	103
5.28.3.12 get_schedule_names()	103
5.28.3.13 init_schedule()	104
5.28.3.14 update_start_time()	104
5.28.4 Variable Documentation	104
5.28.4.1 schedules	104
5.29 src/sdkconfig.h File Reference	105
5.29.1 Macro Definition Documentation	111
5.29.1.1 CONFIG_A2D_INITIAL_TRACE_LEVEL	111
5.29.1.2 CONFIG_A2D_TRACE_LEVEL_WARNING	112
5.29.1.3 CONFIG_ADC2_DISABLE_DAC	112
5.29.1.4 CONFIG_ADC_CAL_EFUSE_TP_ENABLE	112
5.29.1.5 CONFIG_ADC_CAL_EFUSE_VREF_ENABLE	112
5.29.1.6 CONFIG_ADC_CAL_LUT_ENABLE	112
5.29.1.7 CONFIG_APP_COMPILE_TIME_DATE	112
5.29.1.8 CONFIG_APPL_INITIAL_TRACE_LEVEL	112
5.29.1.9 CONFIG_APPL_TRACE_LEVEL_WARNING	112
5.29.1.10 CONFIG_ARDUINO_EVENT_RUN_CORE1	113
5.29.1.11 CONFIG_ARDUINO_EVENT_RUNNING_CORE	113
5.29.1.12 CONFIG_ARDUINO_RUNNING_CORE	113

5.29.1.13 CONFIG_ARDUINO_UDP_RUN_CORE1 . . . . .	113
5.29.1.14 CONFIG_ARDUINO_UDP_RUNNING_CORE . . . . .	113
5.29.1.15 CONFIG_AUTOSTART_ARDUINO . . . . .	113
5.29.1.16 CONFIG_AVCT_INITIAL_TRACE_LEVEL . . . . .	113
5.29.1.17 CONFIG_AVCT_TRACE_LEVEL_WARNING . . . . .	113
5.29.1.18 CONFIG_AVDT_INITIAL_TRACE_LEVEL . . . . .	114
5.29.1.19 CONFIG_AVDT_TRACE_LEVEL_WARNING . . . . .	114
5.29.1.20 CONFIG_AVRC_INITIAL_TRACE_LEVEL . . . . .	114
5.29.1.21 CONFIG_AVRC_TRACE_LEVEL_WARNING . . . . .	114
5.29.1.22 CONFIG_AWS_IOT_MQTT_HOST . . . . .	114
5.29.1.23 CONFIG_AWS_IOT_MQTT_MAX_RECONNECT_WAIT_INTERVAL . . . . .	114
5.29.1.24 CONFIG_AWS_IOT_MQTT_MIN_RECONNECT_WAIT_INTERVAL . . . . .	114
5.29.1.25 CONFIG_AWS_IOT_MQTT_NUM_SUBSCRIBE_HANDLERS . . . . .	114
5.29.1.26 CONFIG_AWS_IOT_MQTT_PORT . . . . .	115
5.29.1.27 CONFIG_AWS_IOT_MQTT_RX_BUF_LEN . . . . .	115
5.29.1.28 CONFIG_AWS_IOT_MQTT_TX_BUF_LEN . . . . .	115
5.29.1.29 CONFIG_AWS_IOT_SDK . . . . .	115
5.29.1.30 CONFIG_AWS_IOT_SHADOW_MAX_JSON_TOKEN_EXPECTED . . . . .	115
5.29.1.31 CONFIG_AWS_IOT_SHADOW_MAX_SHADOW_TOPIC_LENGTH_WITHOUT_THINGNAME . . . . .	115
5.29.1.32 CONFIG_AWS_IOT_SHADOW_MAX_SIMULTANEOUS_ACKS . . . . .	115
5.29.1.33 CONFIG_AWS_IOT_SHADOW_MAX_SIMULTANEOUS_THINGNAMES . . . . .	115
5.29.1.34 CONFIG_AWS_IOT_SHADOW_MAX_SIZE_OF_THING_NAME . . . . .	116
5.29.1.35 CONFIG_AWS_IOT_SHADOW_MAX_SIZE_OF_UNIQUE_CLIENT_ID_BYTES . . . . .	116
5.29.1.36 CONFIG_BLE_ADV_REPORT_DISCARD_THRSHOLD . . . . .	116
5.29.1.37 CONFIG_BLE_ADV_REPORT_FLOW_CONTROL_NUM . . . . .	116
5.29.1.38 CONFIG_BLE_ADV_REPORT_FLOW_CONTROL_SUPPORTED . . . . .	116
5.29.1.39 CONFIG_BLE_ESTABLISH_LINK_CONNECTION_TIMEOUT . . . . .	116
5.29.1.40 CONFIG_BLE_SCAN_DUPLICATE . . . . .	116
5.29.1.41 CONFIG_BLE_SMP_ENABLE . . . . .	116
5.29.1.42 CONFIG_BLUEDROID_ENABLED . . . . .	117
5.29.1.43 CONFIG_BLUEDROID_PINNED_TO_CORE . . . . .	117
5.29.1.44 CONFIG_BLUEDROID_PINNED_TO_CORE_0 . . . . .	117
5.29.1.45 CONFIG_BLUFI_INITIAL_TRACE_LEVEL . . . . .	117
5.29.1.46 CONFIG_BLUFI_TRACE_LEVEL_WARNING . . . . .	117
5.29.1.47 CONFIG_BNEP_INITIAL_TRACE_LEVEL . . . . .	117
5.29.1.48 CONFIG_BNEP_TRACE_LEVEL_WARNING . . . . .	117
5.29.1.49 CONFIG_BOOTLOADER_VDDSDIO_BOOST_1_9V . . . . .	117
5.29.1.50 CONFIG_BOOTLOADER_WDT_ENABLE . . . . .	118
5.29.1.51 CONFIG_BOOTLOADER_WDT_TIME_MS . . . . .	118
5.29.1.52 CONFIG_BROWNOUT_DET . . . . .	118
5.29.1.53 CONFIG_BROWNOUT_DET_LVL . . . . .	118
5.29.1.54 CONFIG_BROWNOUT_DET_LVL_SEL_0 . . . . .	118

5.29.1.55 CONFIG_BT_ACL_CONNECTIONS	118
5.29.1.56 CONFIG_BT_ENABLED	118
5.29.1.57 CONFIG_BT_RESERVE_DRAM	118
5.29.1.58 CONFIG_BT_INITIAL_TRACE_LEVEL	119
5.29.1.59 CONFIG_BT_TASK_STACK_SIZE	119
5.29.1.60 CONFIG_BT_TRACE_LEVEL_WARNING	119
5.29.1.61 CONFIG_BTDM_CONTROLLER_BLE_MAX_CONN	119
5.29.1.62 CONFIG_BTDM_CONTROLLER_BLE_MAX_CONN_EFF	119
5.29.1.63 CONFIG_BTDM_CONTROLLER_BR_EDR_MAX_ACL_CONN_EFF	119
5.29.1.64 CONFIG_BTDM_CONTROLLER_BR_EDR_MAX_SYNC_CONN_EFF	119
5.29.1.65 CONFIG_BTDM_CONTROLLER_HCI_MODE_VHCI	119
5.29.1.66 CONFIG_BTDM_CONTROLLER_MODE_BLE_ONLY	120
5.29.1.67 CONFIG_BTDM_CONTROLLER_PINNED_TO_CORE	120
5.29.1.68 CONFIG_BTDM_CONTROLLER_PINNED_TO_CORE_0	120
5.29.1.69 CONFIG_BTIF_INITIAL_TRACE_LEVEL	120
5.29.1.70 CONFIG_BTIF_TRACE_LEVEL_WARNING	120
5.29.1.71 CONFIG_BTM_INITIAL_TRACE_LEVEL	120
5.29.1.72 CONFIG_BTM_TRACE_LEVEL_WARNING	120
5.29.1.73 CONFIG_BTU_TASK_STACK_SIZE	120
5.29.1.74 CONFIG_CONSOLE_UART_BAUDRATE	121
5.29.1.75 CONFIG_CONSOLE_UART_DEFAULT	121
5.29.1.76 CONFIG_CONSOLE_UART_NUM	121
5.29.1.77 CONFIG_DMA_RX_BUF_NUM	121
5.29.1.78 CONFIG_DMA_TX_BUF_NUM	121
5.29.1.79 CONFIG_DUPLICATE_SCAN_CACHE_SIZE	121
5.29.1.80 CONFIG_EFUSE_CODE_SCHEME_COMPAT_3_4	121
5.29.1.81 CONFIG_EFUSE_MAX_BLK_LEN	121
5.29.1.82 CONFIG_EMAC_CHECK_LINK_PERIOD_MS	122
5.29.1.83 CONFIG_EMAC_TASK_PRIORITY	122
5.29.1.84 CONFIG_EMAC_TASK_STACK_SIZE	122
5.29.1.85 CONFIG_ENABLE_ARDUINO_DEPENDS	122
5.29.1.86 CONFIG_ESP32_APPTRACE_DEST_NONE	122
5.29.1.87 CONFIG_ESP32_APPTRACE_LOCK_ENABLE	122
5.29.1.88 CONFIG_ESP32_DEBUG_OCDAWARE	122
5.29.1.89 CONFIG_ESP32_DEBUG_STUBS_ENABLE	122
5.29.1.90 CONFIG_ESP32_DEEP_SLEEP_WAKEUP_DELAY	123
5.29.1.91 CONFIG_ESP32_DEFAULT_CPU_FREQ_160	123
5.29.1.92 CONFIG_ESP32_DEFAULT_CPU_FREQ_MHZ	123
5.29.1.93 CONFIG_ESP32_DEFAULT_PTHREAD_CORE_NO_AFFINITY	123
5.29.1.94 CONFIG_ESP32_DPORT_WORKAROUND	123
5.29.1.95 CONFIG_ESP32_ENABLE_COREDUMP_TO_NONE	123
5.29.1.96 CONFIG_ESP32_PANIC_PRINT_REBOOT	123

5.29.1.97 CONFIG_ESP32_PHY_CALIBRATION_AND_DATA_STORAGE . . . . .	123
5.29.1.98 CONFIG_ESP32_PHY_MAX_TX_POWER . . . . .	124
5.29.1.99 CONFIG_ESP32_PHY_MAX_WIFI_TX_POWER . . . . .	124
5.29.1.100 CONFIG_ESP32_PTHREAD_TASK_CORE_DEFAULT . . . . .	124
5.29.1.101 CONFIG_ESP32_PTHREAD_TASK_NAME_DEFAULT . . . . .	124
5.29.1.102 CONFIG_ESP32_PTHREAD_TASK_PRIO_DEFAULT . . . . .	124
5.29.1.103 CONFIG_ESP32_PTHREAD_TASK_STACK_SIZE_DEFAULT . . . . .	124
5.29.1.104 CONFIG_ESP32_REV_MIN . . . . .	124
5.29.1.105 CONFIG_ESP32_REV_MIN_0 . . . . .	124
5.29.1.106 CONFIG_ESP32_RTC_CLK_CAL_CYCLES . . . . .	125
5.29.1.107 CONFIG_ESP32_RTC_CLOCK_SOURCE_INTERNAL_RC . . . . .	125
5.29.1.108 CONFIG_ESP32_TIME_SYSCALL_USE_RTC_FRC1 . . . . .	125
5.29.1.109 CONFIG_ESP32_WIFI_AMPDU_RX_ENABLED . . . . .	125
5.29.1.110 CONFIG_ESP32_WIFI_AMPDU_TX_ENABLED . . . . .	125
5.29.1.111 CONFIG_ESP32_WIFI_DYNAMIC_RX_BUFFER_NUM . . . . .	125
5.29.1.112 CONFIG_ESP32_WIFI_DYNAMIC_TX_BUFFER . . . . .	125
5.29.1.113 CONFIG_ESP32_WIFI_DYNAMIC_TX_BUFFER_NUM . . . . .	125
5.29.1.114 CONFIG_ESP32_WIFI_IRAM_OPT . . . . .	126
5.29.1.115 CONFIG_ESP32_WIFI_MGMT_SBUF_NUM . . . . .	126
5.29.1.116 CONFIG_ESP32_WIFI_NVS_ENABLED . . . . .	126
5.29.1.117 CONFIG_ESP32_WIFI_RX_BA_WIN . . . . .	126
5.29.1.118 CONFIG_ESP32_WIFI_SOFTAP_BEACON_MAX_LEN . . . . .	126
5.29.1.119 CONFIG_ESP32_WIFI_STATIC_RX_BUFFER_NUM . . . . .	126
5.29.1.120 CONFIG_ESP32_WIFI_TASK_PINNED_TO_CORE_0 . . . . .	126
5.29.1.121 CONFIG_ESP32_WIFI_TX_BA_WIN . . . . .	126
5.29.1.122 CONFIG_ESP32_WIFI_TX_BUFFER_TYPE . . . . .	127
5.29.1.123 CONFIG_ESP32_XTAL_FREQ . . . . .	127
5.29.1.124 CONFIG_ESP32_XTAL_FREQ_40 . . . . .	127
5.29.1.125 CONFIG_ESP_ERR_TO_NAME_LOOKUP . . . . .	127
5.29.1.126 CONFIG_ESP_GRATUITOUS_ARP . . . . .	127
5.29.1.127 CONFIG_ESP_HTTP_CLIENT_ENABLE_HTTPS . . . . .	127
5.29.1.128 CONFIG_ESPTOOLPY_AFTER . . . . .	127
5.29.1.129 CONFIG_ESPTOOLPY_AFTER_RESET . . . . .	127
5.29.1.130 CONFIG_ESPTOOLPY_BAUD . . . . .	128
5.29.1.131 CONFIG_ESPTOOLPY_BAUD_115200B . . . . .	128
5.29.1.132 CONFIG_ESPTOOLPY_BAUD_OTHER_VAL . . . . .	128
5.29.1.133 CONFIG_ESPTOOLPY_BEFORE . . . . .	128
5.29.1.134 CONFIG_ESPTOOLPY_BEFORE_RESET . . . . .	128
5.29.1.135 CONFIG_ESPTOOLPY_COMPRESSED . . . . .	128
5.29.1.136 CONFIG_ESPTOOLPY_FLASHFREQ . . . . .	128
5.29.1.137 CONFIG_ESPTOOLPY_FLASHFREQ_40M . . . . .	128
5.29.1.138 CONFIG_ESPTOOLPY_FLASHMODE . . . . .	129

5.29.1.139 CONFIG_ESPTOOLPY_FLASHSIZE	129
5.29.1.140 CONFIG_ESPTOOLPY_FLASHSIZE_2MB	129
5.29.1.141 CONFIG_ESPTOOLPY_FLASHSIZE_DETECT	129
5.29.1.142 CONFIG_ESPTOOLPY_PORT	129
5.29.1.143 CONFIG_FATFS_CODEPAGE	129
5.29.1.144 CONFIG_FATFS_CODEPAGE_437	129
5.29.1.145 CONFIG_FATFS_FS_LOCK	129
5.29.1.146 CONFIG_FATFS_LFN_NONE	130
5.29.1.147 CONFIG_FATFS_PER_FILE_CACHE	130
5.29.1.148 CONFIG_FATFS_TIMEOUT_MS	130
5.29.1.149 CONFIG_FLASHMODE_DIO	130
5.29.1.150 CONFIG_FOUR_UNIVERSAL_MAC_ADDRESS	130
5.29.1.151 CONFIG_FREERTOS_ASSERT_FAIL_ABORT	130
5.29.1.152 CONFIG_FREERTOS_ASSERT_ON_UNTESTED_FUNCTION	130
5.29.1.153 CONFIG_FREERTOS_CHECK_MUTEX_GIVEN_BY_OWNER	130
5.29.1.154 CONFIG_FREERTOS_CHECK_STACKOVERFLOW_CANARY	131
5.29.1.155 CONFIG_FREERTOS_CORETIMER_0	131
5.29.1.156 CONFIG_FREERTOS_HZ	131
5.29.1.157 CONFIG_FREERTOS_IDLE_TASK_STACKSIZE	131
5.29.1.158 CONFIG_FREERTOS_INTERRUPT_BACKTRACE	131
5.29.1.159 CONFIG_FREERTOS_ISR_STACKSIZE	131
5.29.1.160 CONFIG_FREERTOS_MAX_TASK_NAME_LEN	131
5.29.1.161 CONFIG_FREERTOS_NO_AFFINITY	131
5.29.1.162 CONFIG_FREERTOS_QUEUE_REGISTRY_SIZE	132
5.29.1.163 CONFIG_FREERTOS_TASK_FUNCTION_WRAPPER	132
5.29.1.164 CONFIG_FREERTOS_THREAD_LOCAL_STORAGE_POINTERS	132
5.29.1.165 CONFIG_GAP_INITIAL_TRACE_LEVEL	132
5.29.1.166 CONFIG_GAP_TRACE_LEVEL_WARNING	132
5.29.1.167 CONFIG_GARP_TMR_INTERVAL	132
5.29.1.168 CONFIG_GATT_INITIAL_TRACE_LEVEL	132
5.29.1.169 CONFIG_GATT_TRACE_LEVEL_WARNING	132
5.29.1.170 CONFIG_GATTC_ENABLE	133
5.29.1.171 CONFIG_GATTS_ENABLE	133
5.29.1.172 CONFIG_GATTS_SEND_SERVICE_CHANGE_AUTO	133
5.29.1.173 CONFIG_GATTS_SEND_SERVICE_CHANGE_MODE	133
5.29.1.174 CONFIG_HCI_INITIAL_TRACE_LEVEL	133
5.29.1.175 CONFIG_HCI_TRACE_LEVEL_WARNING	133
5.29.1.176 CONFIG_HEAP_POISONING_DISABLED	133
5.29.1.177 CONFIG_HID_INITIAL_TRACE_LEVEL	133
5.29.1.178 CONFIG_HID_TRACE_LEVEL_WARNING	134
5.29.1.179 CONFIG_HTTPD_ERR_RESP_NO_DELAY	134
5.29.1.180 CONFIG_HTTPD_MAX_REQ_HDR_LEN	134



5.29.1.181 CONFIG_HTTPD_MAX_URI_LEN . . . . .	134
5.29.1.182 CONFIG_HTTPD_PURGE_BUF_LEN . . . . .	134
5.29.1.183 CONFIG_IDF_TARGET . . . . .	134
5.29.1.184 CONFIG_IDF_TARGET_ESP32 . . . . .	134
5.29.1.185 CONFIG_INT_WDT . . . . .	134
5.29.1.186 CONFIG_INT_WDT_CHECK_CPU1 . . . . .	135
5.29.1.187 CONFIG_INT_WDT_TIMEOUT_MS . . . . .	135
5.29.1.188 CONFIG_IP_LOST_TIMER_INTERVAL . . . . .	135
5.29.1.189 CONFIG_IPC_TASK_STACK_SIZE . . . . .	135
5.29.1.190 CONFIG_L2CAP_INITIAL_TRACE_LEVEL . . . . .	135
5.29.1.191 CONFIG_L2CAP_TRACE_LEVEL_WARNING . . . . .	135
5.29.1.192 CONFIG_LIBSODIUM_USE_MBEDTLS_SHA . . . . .	135
5.29.1.193 CONFIG_LOG_BOOTLOADER_LEVEL . . . . .	135
5.29.1.194 CONFIG_LOG_BOOTLOADER_LEVEL_INFO . . . . .	136
5.29.1.195 CONFIG_LOG_COLORS . . . . .	136
5.29.1.196 CONFIG_LOG_DEFAULT_LEVEL . . . . .	136
5.29.1.197 CONFIG_LOG_DEFAULT_LEVEL_INFO . . . . .	136
5.29.1.198 CONFIG_LWIP_DHCP_DOES_ARP_CHECK . . . . .	136
5.29.1.199 CONFIG_LWIP_DHCP_MAX_NTP_SERVERS . . . . .	136
5.29.1.200 CONFIG_LWIP_DHCPS_LEASE_UNIT . . . . .	136
5.29.1.201 CONFIG_LWIP_DHCPS_MAX_STATION_NUM . . . . .	136
5.29.1.202 CONFIG_LWIP_LOOPBACK_MAX_PBUFS . . . . .	137
5.29.1.203 CONFIG_LWIP_MAX_ACTIVE_TCP . . . . .	137
5.29.1.204 CONFIG_LWIP_MAX_LISTENING_TCP . . . . .	137
5.29.1.205 CONFIG_LWIP_MAX_RAW_PCBS . . . . .	137
5.29.1.206 CONFIG_LWIP_MAX_SOCKETS . . . . .	137
5.29.1.207 CONFIG_LWIP_MAX_UDP_PCBS . . . . .	137
5.29.1.208 CONFIG_LWIP_NETIF_LOOPBACK . . . . .	137
5.29.1.209 CONFIG_LWIP_SO_REUSE . . . . .	137
5.29.1.210 CONFIG_LWIP_SO_REUSE_RXTOALL . . . . .	138
5.29.1.211 CONFIG_MAIN_TASK_STACK_SIZE . . . . .	138
5.29.1.212 CONFIG_MAKE_WARN_UNDEFINED_VARIABLES . . . . .	138
5.29.1.213 CONFIG_MB_CONTROLLER_NOTIFY_QUEUE_SIZE . . . . .	138
5.29.1.214 CONFIG_MB_CONTROLLER_NOTIFY_TIMEOUT . . . . .	138
5.29.1.215 CONFIG_MB_CONTROLLER_STACK_SIZE . . . . .	138
5.29.1.216 CONFIG_MB_EVENT_QUEUE_TIMEOUT . . . . .	138
5.29.1.217 CONFIG_MB_QUEUE_LENGTH . . . . .	138
5.29.1.218 CONFIG_MB_SERIAL_BUF_SIZE . . . . .	139
5.29.1.219 CONFIG_MB_SERIAL_TASK_PRIO . . . . .	139
5.29.1.220 CONFIG_MB_SERIAL_TASK_STACK_SIZE . . . . .	139
5.29.1.221 CONFIG_MB_TIMER_GROUP . . . . .	139
5.29.1.222 CONFIG_MB_TIMER_INDEX . . . . .	139

5.29.1.223 CONFIG_MB_TIMER_PORT_ENABLED . . . . .	139
5.29.1.224 CONFIG_MBEDTLS_AES_C . . . . .	139
5.29.1.225 CONFIG_MBEDTLS_CCM_C . . . . .	139
5.29.1.226 CONFIG_MBEDTLS_ECDH_C . . . . .	140
5.29.1.227 CONFIG_MBEDTLS_ECDSA_C . . . . .	140
5.29.1.228 CONFIG_MBEDTLS_ECP_C . . . . .	140
5.29.1.229 CONFIG_MBEDTLS_ECP_DP_BP256R1_ENABLED . . . . .	140
5.29.1.230 CONFIG_MBEDTLS_ECP_DP_BP384R1_ENABLED . . . . .	140
5.29.1.231 CONFIG_MBEDTLS_ECP_DP_BP512R1_ENABLED . . . . .	140
5.29.1.232 CONFIG_MBEDTLS_ECP_DP_CURVE25519_ENABLED . . . . .	140
5.29.1.233 CONFIG_MBEDTLS_ECP_DP_SECP192K1_ENABLED . . . . .	140
5.29.1.234 CONFIG_MBEDTLS_ECP_DP_SECP192R1_ENABLED . . . . .	141
5.29.1.235 CONFIG_MBEDTLS_ECP_DP_SECP224K1_ENABLED . . . . .	141
5.29.1.236 CONFIG_MBEDTLS_ECP_DP_SECP224R1_ENABLED . . . . .	141
5.29.1.237 CONFIG_MBEDTLS_ECP_DP_SECP256K1_ENABLED . . . . .	141
5.29.1.238 CONFIG_MBEDTLS_ECP_DP_SECP256R1_ENABLED . . . . .	141
5.29.1.239 CONFIG_MBEDTLS_ECP_DP_SECP384R1_ENABLED . . . . .	141
5.29.1.240 CONFIG_MBEDTLS_ECP_DP_SECP521R1_ENABLED . . . . .	141
5.29.1.241 CONFIG_MBEDTLS_ECP_NIST_OPTIM . . . . .	141
5.29.1.242 CONFIG_MBEDTLS_GCM_C . . . . .	142
5.29.1.243 CONFIG_MBEDTLS_HARDWARE_AES . . . . .	142
5.29.1.244 CONFIG_MBEDTLS_HAVE_TIME . . . . .	142
5.29.1.245 CONFIG_MBEDTLS_INTERNAL_MEM_ALLOC . . . . .	142
5.29.1.246 CONFIG_MBEDTLS_KEY_EXCHANGE_DHE_RSA . . . . .	142
5.29.1.247 CONFIG_MBEDTLS_KEY_EXCHANGE_ECDH_ECDSA . . . . .	142
5.29.1.248 CONFIG_MBEDTLS_KEY_EXCHANGE_ECDH_RSA . . . . .	142
5.29.1.249 CONFIG_MBEDTLS_KEY_EXCHANGE_ECDHE_ECDSA . . . . .	142
5.29.1.250 CONFIG_MBEDTLS_KEY_EXCHANGE_ECDHE_RSA . . . . .	143
5.29.1.251 CONFIG_MBEDTLS_KEY_EXCHANGE_ELLIPTIC_CURVE . . . . .	143
5.29.1.252 CONFIG_MBEDTLS_KEY_EXCHANGE_RSA . . . . .	143
5.29.1.253 CONFIG_MBEDTLS_PEM_PARSE_C . . . . .	143
5.29.1.254 CONFIG_MBEDTLS_PEM_WRITE_C . . . . .	143
5.29.1.255 CONFIG_MBEDTLS_RC4_DISABLED . . . . .	143
5.29.1.256 CONFIG_MBEDTLS_SSL_ALPN . . . . .	143
5.29.1.257 CONFIG_MBEDTLS_SSL_MAX_CONTENT_LEN . . . . .	143
5.29.1.258 CONFIG_MBEDTLS_SSL_PROTO_TLS1 . . . . .	144
5.29.1.259 CONFIG_MBEDTLS_SSL_PROTO_TLS1_1 . . . . .	144
5.29.1.260 CONFIG_MBEDTLS_SSL_PROTO_TLS1_2 . . . . .	144
5.29.1.261 CONFIG_MBEDTLS_SSL_RENEGOTIATION . . . . .	144
5.29.1.262 CONFIG_MBEDTLS_SSL_SESSION_TICKETS . . . . .	144
5.29.1.263 CONFIG_MBEDTLS_TLS_CLIENT . . . . .	144
5.29.1.264 CONFIG_MBEDTLS_TLS_ENABLED . . . . .	144

5.29.1.265 CONFIG_MBEDTLS_TLS_SERVER . . . . .	144
5.29.1.266 CONFIG_MBEDTLS_TLS_SERVER_AND_CLIENT . . . . .	145
5.29.1.267 CONFIG_MBEDTLS_X509_CRL_PARSE_C . . . . .	145
5.29.1.268 CONFIG_MBEDTLS_X509_CSR_PARSE_C . . . . .	145
5.29.1.269 CONFIG_MCA_INITIAL_TRACE_LEVEL . . . . .	145
5.29.1.270 CONFIG_MCA_TRACE_LEVEL_WARNING . . . . .	145
5.29.1.271 CONFIG_MDNS_MAX_SERVICES . . . . .	145
5.29.1.272 CONFIG_MONITOR_BAUD . . . . .	145
5.29.1.273 CONFIG_MONITOR_BAUD_115200B . . . . .	145
5.29.1.274 CONFIG_MONITOR_BAUD_OTHER_VAL . . . . .	146
5.29.1.275 CONFIG_MQTT_PROTOCOL_311 . . . . .	146
5.29.1.276 CONFIG_MQTT_TRANSPORT_SSL . . . . .	146
5.29.1.277 CONFIG_MQTT_TRANSPORT_WEBSOCKET . . . . .	146
5.29.1.278 CONFIG_MQTT_TRANSPORT_WEBSOCKET_SECURE . . . . .	146
5.29.1.279 CONFIG_NEWLIB_STDIN_LINE_ENDING_CR . . . . .	146
5.29.1.280 CONFIG_NEWLIB_STDOUT_LINE_ENDING_CRLF . . . . .	146
5.29.1.281 CONFIG_NUMBER_OF_UNIVERSAL_MAC_ADDRESS . . . . .	146
5.29.1.282 CONFIG_OPENSSL_ASSERT_DO_NOTHING . . . . .	147
5.29.1.283 CONFIG_OPTIMIZATION_ASSERTIONS_ENABLED . . . . .	147
5.29.1.284 CONFIG_OPTIMIZATION_LEVEL_DEBUG . . . . .	147
5.29.1.285 CONFIG_OSI_INITIAL_TRACE_LEVEL . . . . .	147
5.29.1.286 CONFIG_OSI_TRACE_LEVEL_WARNING . . . . .	147
5.29.1.287 CONFIG_PAN_INITIAL_TRACE_LEVEL . . . . .	147
5.29.1.288 CONFIG_PAN_TRACE_LEVEL_WARNING . . . . .	147
5.29.1.289 CONFIG_PARTITION_TABLE_CUSTOM_FILENAME . . . . .	147
5.29.1.290 CONFIG_PARTITION_TABLE_FILENAME . . . . .	148
5.29.1.291 CONFIG_PARTITION_TABLE_MD5 . . . . .	148
5.29.1.292 CONFIG_PARTITION_TABLE_OFFSET . . . . .	148
5.29.1.293 CONFIG_PARTITION_TABLE_SINGLE_APP . . . . .	148
5.29.1.294 CONFIG_PTHREAD_STACK_MIN . . . . .	148
5.29.1.295 CONFIG_PYTHON . . . . .	148
5.29.1.296 CONFIG_REDUCE_PHY_TX_POWER . . . . .	148
5.29.1.297 CONFIG_RFCOMM_INITIAL_TRACE_LEVEL . . . . .	148
5.29.1.298 CONFIG_RFCOMM_TRACE_LEVEL_WARNING . . . . .	149
5.29.1.299 CONFIG_SCAN_DUPLICATE_BY_DEVICE_ADDR . . . . .	149
5.29.1.300 CONFIG_SCAN_DUPLICATE_TYPE . . . . .	149
5.29.1.301 CONFIG_SDP_INITIAL_TRACE_LEVEL . . . . .	149
5.29.1.302 CONFIG_SDP_TRACE_LEVEL_WARNING . . . . .	149
5.29.1.303 CONFIG_SMP_ENABLE . . . . .	149
5.29.1.304 CONFIG_SMP_INITIAL_TRACE_LEVEL . . . . .	149
5.29.1.305 CONFIG_SMP_TRACE_LEVEL_WARNING . . . . .	149
5.29.1.306 CONFIG_SPI_FLASH_ERASE_YIELD_DURATION_MS . . . . .	150

5.29.1.307 CONFIG_SPI_FLASH_ERASE_YIELD_TICKS . . . . .	150
5.29.1.308 CONFIG_SPI_FLASH_ROM_DRIVER_PATCH . . . . .	150
5.29.1.309 CONFIG_SPI_FLASH_WRITING_DANGEROUS_REGIONS_ABORTS . . . . .	150
5.29.1.310 CONFIG_SPI_FLASH_YIELD_DURING_ERASE . . . . .	150
5.29.1.311 CONFIG_SPI_MASTER_ISR_IN_IRAM . . . . .	150
5.29.1.312 CONFIG_SPI_SLAVE_ISR_IN_IRAM . . . . .	150
5.29.1.313 CONFIG_SPIFFS_CACHE . . . . .	150
5.29.1.314 CONFIG_SPIFFS_CACHE_WR . . . . .	151
5.29.1.315 CONFIG_SPIFFS_GC_MAX_RUNS . . . . .	151
5.29.1.316 CONFIG_SPIFFS_MAX_PARTITIONS . . . . .	151
5.29.1.317 CONFIG_SPIFFS_META_LENGTH . . . . .	151
5.29.1.318 CONFIG_SPIFFS_OBJ_NAME_LEN . . . . .	151
5.29.1.319 CONFIG_SPIFFS_PAGE_CHECK . . . . .	151
5.29.1.320 CONFIG_SPIFFS_PAGE_SIZE . . . . .	151
5.29.1.321 CONFIG_SPIFFS_USE_MAGIC . . . . .	151
5.29.1.322 CONFIG_SPIFFS_USE_MAGIC_LENGTH . . . . .	152
5.29.1.323 CONFIG_SPIFFS_USE_MTIME . . . . .	152
5.29.1.324 CONFIG_STACK_CHECK_NONE . . . . .	152
5.29.1.325 CONFIG_SUPPORT_TERMIOS . . . . .	152
5.29.1.326 CONFIG_SUPPRESS_SELECT_DEBUG_OUTPUT . . . . .	152
5.29.1.327 CONFIG_SW_COEXIST_ENABLE . . . . .	152
5.29.1.328 CONFIG_SW_COEXIST_PREFERENCE_BALANCE . . . . .	152
5.29.1.329 CONFIG_SW_COEXIST_PREFERENCE_VALUE . . . . .	152
5.29.1.330 CONFIG_SYSTEM_EVENT_QUEUE_SIZE . . . . .	153
5.29.1.331 CONFIG_SYSTEM_EVENT_TASK_STACK_SIZE . . . . .	153
5.29.1.332 CONFIG_TASK_WDT . . . . .	153
5.29.1.333 CONFIG_TASK_WDT_CHECK_IDLE_TASK_CPU0 . . . . .	153
5.29.1.334 CONFIG_TASK_WDT_CHECK_IDLE_TASK_CPU1 . . . . .	153
5.29.1.335 CONFIG_TASK_WDT_TIMEOUT_S . . . . .	153
5.29.1.336 CONFIG_TCP_MAXRTX . . . . .	153
5.29.1.337 CONFIG_TCP_MSL . . . . .	153
5.29.1.338 CONFIG_TCP_MSS . . . . .	154
5.29.1.339 CONFIG_TCP_OVERSIZE_MSS . . . . .	154
5.29.1.340 CONFIG_TCP_QUEUE_OOSEQ . . . . .	154
5.29.1.341 CONFIG_TCP_RECVMBOX_SIZE . . . . .	154
5.29.1.342 CONFIG_TCP_SND_BUF_DEFAULT . . . . .	154
5.29.1.343 CONFIG_TCP_SYNMAXRTX . . . . .	154
5.29.1.344 CONFIG_TCP_WND_DEFAULT . . . . .	154
5.29.1.345 CONFIG_TCPIP_LWIP . . . . .	154
5.29.1.346 CONFIG_TCPIP_RECVMBOX_SIZE . . . . .	155
5.29.1.347 CONFIG_TCPIP_TASK_AFFINITY . . . . .	155
5.29.1.348 CONFIG_TCPIP_TASK_AFFINITY_NO_AFFINITY . . . . .	155

5.29.1.349 CONFIG_TCPIP_TASK_STACK_SIZE . . . . .	155
5.29.1.350 CONFIG_TIMER_QUEUE_LENGTH . . . . .	155
5.29.1.351 CONFIG_TIMER_TASK_PRIORITY . . . . .	155
5.29.1.352 CONFIG_TIMER_TASK_STACK_DEPTH . . . . .	155
5.29.1.353 CONFIG_TIMER_TASK_STACK_SIZE . . . . .	155
5.29.1.354 CONFIG_TOOLPREFIX . . . . .	156
5.29.1.355 CONFIG_TRACEMEM_RESERVE_DRAM . . . . .	156
5.29.1.356 CONFIG_UDP_RECVMBOX_SIZE . . . . .	156
5.29.1.357 CONFIG_ULP_COPROC_RESERVE_MEM . . . . .	156
5.29.1.358 CONFIG_UNITY_ENABLE_DOUBLE . . . . .	156
5.29.1.359 CONFIG_UNITY_ENABLE_FLOAT . . . . .	156
5.29.1.360 CONFIG_UNITY_ENABLE_IDF_TEST_RUNNER . . . . .	156
5.29.1.361 CONFIG_WIFI_PROV_SCAN_MAX_ENTRIES . . . . .	156
5.29.1.362 CONFIG_WL_SECTOR_SIZE . . . . .	157
5.29.1.363 CONFIG_WL_SECTOR_SIZE_4096 . . . . .	157
5.30 src/wifi.cpp File Reference . . . . .	157
5.30.1 Detailed Description . . . . .	158
5.30.2 Macro Definition Documentation . . . . .	158
5.30.2.1 ESP_WIFI_PASS . . . . .	158
5.30.2.2 ESP_WIFI_SSID . . . . .	158
5.30.2.3 MAX_STA_CONN . . . . .	158
5.30.3 Function Documentation . . . . .	158
5.30.3.1 wifi_init_softap() . . . . .	159
5.30.3.2 wifi_init_sta() . . . . .	159
5.30.4 Variable Documentation . . . . .	159
5.30.4.1 WIFI_CONNECTED_BIT . . . . .	159
<b>Index</b>	<b>161</b>



## **Chapter 1**

# **SmartLed**

Repository for Smart LED Control System Senior Design Project





## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">channel</a> . . . . .	<a href="#">7</a>
<a href="#">gatts_profile_inst</a> . . . . .	<a href="#">8</a>
<a href="#">Node</a>	
Nodes of a singly linked list data structure . . . . .	<a href="#">10</a>
<a href="#">prepare_type_env_t</a> . . . . .	<a href="#">11</a>
<a href="#">Schedule_Object</a>	
Data object that specified parameters a schedule can contain . . . . .	<a href="#">11</a>



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

include/ArduinoJson-v6.14.1.h	15
include/bleSL.h	15
include/dawndusk.h	16
include/decode_bluetooth.h	19
include/espsntp.h	23
include/gatts_table_creat_demo.h	24
include/http.h	25
include/led.h	30
include/main.h	33
include/measurement.h	33
include/memory.h	36
include/pin_defs.h	42
include/rctdefine.h	44
include/schedule_object.h	52
include/scheduler.h	54
include/wifi.h	60
src/bleSL.cpp	
GATT server demo code used as base from esp-idf. Most modifications made in Init_Bluetooth function and gatts_profile_event_handler callback	61
src/dawndusk.cpp	65
src/decode_bluetooth.cpp	
This file contains functions used to decode/set byte packets from GATT server	67
src/espsntp.cpp	73
src/http.cpp	
File for hosting http server specific to LED controller/scheduler	74
src/led.cpp	81
src/main.cpp	84
src/measurement.cpp	85
src/memory.cpp	88
src/rct.cpp	95
src/scheduler.cpp	98
src/sdkconfig.h	105
src/wifi.cpp	
File to connect ESP32 to WiFi	157



## Chapter 4

# Class Documentation

### 4.1 channel Struct Reference

#### Public Attributes

- char [name](#) [250]
- uint16\_t [r](#)
- uint16\_t [g](#)
- uint16\_t [b](#)
- uint8\_t [brightness](#)

#### 4.1.1 Member Data Documentation

##### 4.1.1.1 b

```
uint16_t channel::b
```

##### 4.1.1.2 brightness

```
uint8_t channel::brightness
```

##### 4.1.1.3 g

```
uint16_t channel::g
```

#### 4.1.1.4 name

```
char channel::name[250]
```

#### 4.1.1.5 r

```
uint16_t channel::r
```

The documentation for this struct was generated from the following file:

- [src/led.cpp](#)

## 4.2 gatts\_profile\_inst Struct Reference

### Public Attributes

- [esp\\_gatts\\_cb\\_t](#) [gatts\\_cb](#)
- [uint16\\_t](#) [gatts\\_if](#)
- [uint16\\_t](#) [app\\_id](#)
- [uint16\\_t](#) [conn\\_id](#)
- [uint16\\_t](#) [service\\_handle](#)
- [esp\\_gatt\\_srvc\\_id\\_t](#) [service\\_id](#)
- [uint16\\_t](#) [char\\_handle](#)
- [esp\\_bt\\_uuid\\_t](#) [char\\_uuid](#)
- [esp\\_gatt\\_perm\\_t](#) [perm](#)
- [esp\\_gatt\\_char\\_prop\\_t](#) [property](#)
- [uint16\\_t](#) [descr\\_handle](#)
- [esp\\_bt\\_uuid\\_t](#) [descr\\_uuid](#)

### 4.2.1 Member Data Documentation

#### 4.2.1.1 app\_id

```
uint16_t gatts_profile_inst::app_id
```

#### 4.2.1.2 char\_handle

```
uint16_t gatts_profile_inst::char_handle
```

#### 4.2.1.3 char\_uuid

esp\_bt\_uuid\_t gatts\_profile\_inst::char\_uuid

#### 4.2.1.4 conn\_id

uint16\_t gatts\_profile\_inst::conn\_id

#### 4.2.1.5 descr\_handle

uint16\_t gatts\_profile\_inst::descr\_handle

#### 4.2.1.6 descr\_uuid

esp\_bt\_uuid\_t gatts\_profile\_inst::descr\_uuid

#### 4.2.1.7 gatts\_cb

esp\_gatts\_cb\_t gatts\_profile\_inst::gatts\_cb

#### 4.2.1.8 gatts\_if

uint16\_t gatts\_profile\_inst::gatts\_if

#### 4.2.1.9 perm

esp\_gatt\_perm\_t gatts\_profile\_inst::perm

#### 4.2.1.10 property

esp\_gatt\_char\_prop\_t gatts\_profile\_inst::property

#### 4.2.1.11 service\_handle

```
uint16_t gatts_profile_inst::service_handle
```

#### 4.2.1.12 service\_id

```
esp_gatt_srvc_id_t gatts_profile_inst::service_id
```

The documentation for this struct was generated from the following file:

- [src/bleSL.cpp](#)

## 4.3 Node Struct Reference

Nodes of a singly linked list data structure.

```
#include <schedule_object.h>
```

### Public Attributes

- [schedule\\_object](#) [schedule](#)
- struct [Node](#) \* [next](#)

#### 4.3.1 Detailed Description

Nodes of a singly linked list data structure.

#### 4.3.2 Member Data Documentation

##### 4.3.2.1 next

```
struct Node* Node::next
```

Pointer to the next node in the linked list data structure



#### 4.3.2.2 schedule

`schedule_object` Node::schedule

the schedule associated with this node

The documentation for this struct was generated from the following file:

- include/schedule\_object.h

## 4.4 prepare\_type\_env\_t Struct Reference

### Public Attributes

- uint8\_t \* `prepare_buf`
- int `prepare_len`

#### 4.4.1 Member Data Documentation

##### 4.4.1.1 prepare\_buf

uint8\_t\* prepare\_type\_env\_t::prepare\_buf

##### 4.4.1.2 prepare\_len

int prepare\_type\_env\_t::prepare\_len

The documentation for this struct was generated from the following file:

- src/bleSL.cpp

## 4.5 Schedule\_Object Struct Reference

data object that specified parameters a schedule can contain

```
#include <schedule_object.h>
```

## Public Attributes

- uint8\_t [ID](#)
- char [name](#) [250]
- uint8\_t [enabled](#)
- uint32\_t [start](#)
- uint32\_t [duration](#)
- uint8\_t [repeat\\_mask](#)
- uint32\_t [repeat\\_time](#)
- uint8\_t [dawn](#)
- uint8\_t [dusk](#)
- uint8\_t [isRGB](#)
- uint8\_t [brightness](#)
- uint8\_t [r](#)
- uint8\_t [g](#)
- uint8\_t [b](#)

### 4.5.1 Detailed Description

data object that specified parameters a schedule can contain

### 4.5.2 Member Data Documentation

#### 4.5.2.1 **b**

```
uint8_t Schedule_Object::b
```

The blue portion of the color. Values from 0-255.

#### 4.5.2.2 **brightness**

```
uint8_t Schedule_Object::brightness
```

The brightness of the LED. 0-255 with 0 being off, and 255 being maximum brightness.

#### 4.5.2.3 **dawn**

```
uint8_t Schedule_Object::dawn
```

Whether the schedule runs at dawn. Not compatible with `repeat_time`

#### 4.5.2.4 **duration**

```
uint32_t Schedule_Object::duration
```

The duration, in seconds, of the schedule. If set to the max value the schedule will run indefinitely.

#### 4.5.2.5 dusk

```
uint8_t Schedule_Object::dusk
```

Whether the schedule runs at dusk. Not compatible with repeat\_time.

#### 4.5.2.6 enabled

```
uint8_t Schedule_Object::enabled
```

The enabled status of the schedule. If 0, the schedule does not run. If not 0 the schedule actively runs according to other parameters.

#### 4.5.2.7 g

```
uint8_t Schedule_Object::g
```

The green portion of the color. Values from 0-255.

#### 4.5.2.8 ID

```
uint8_t Schedule_Object::ID
```

The ID for the schedule. Each schedule in a list should have its own unique ID.

#### 4.5.2.9 isRGB

```
uint8_t Schedule_Object::isRGB
```

The type of LED associated. If this field is 0, the LED is not an RGB LED and only the brightness field is used in controlling the LED. If isRGB is not 0, the brightness, R, G, and B fields control the LED

#### 4.5.2.10 name

```
char Schedule_Object::name[250]
```

The name of the schedule. Each schedule in a list should have its own unique name.

#### 4.5.2.11 r

```
uint8_t Schedule_Object::r
```

The red portion of the color. Values from 0-255.

#### 4.5.2.12 repeat\_mask

```
uint8_t Schedule_Object::repeat_mask
```

A bit mask for which days the schedule runs on. If repeat\_time is set this field is not used. The MSB bit is not used and following bits represent the days. The 8 bits represent [Unused, Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday].

#### 4.5.2.13 repeat\_time

```
uint32_t Schedule_Object::repeat_time
```

An alternative way to have schedules repeat instead of by day. This field represents the time, in seconds, that the schedule repeats. The next start time is calculated by adding the current start time plus the repeat\_time. Settings this field to 5 seconds means that the schedule's start is triggered every 5 seconds. If this field is enabled then repeat\_mask and dawn/dusk will not work.

#### 4.5.2.14 start

```
uint32_t Schedule_Object::start
```

The unix timestamp of the start time. If a schedule repeats, this field will update to the next start time as the schedule runs.

The documentation for this struct was generated from the following file:

- [include/schedule\\_object.h](#)

## Chapter 5

# File Documentation

### 5.1 include/ArduinoJson-v6.14.1.h File Reference

### 5.2 include/bleSL.h File Reference

#### Functions

- void [Init\\_Bluetooth](#) (void)  
*Initializes GATT Server.*

#### 5.2.1 Detailed Description

Description: Function declaration to initialize BLE GATT Server.

#### Author

: Hunaid Puri

#### Date

last modified: 4/26/2020

#### 5.2.2 Function Documentation

##### 5.2.2.1 Init\_Bluetooth()

```
void Init_Bluetooth (
    void )
```

Initializes GATT Server.

## 5.3 include/dawndusk.h File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
```

### Macros

- #define [radius](#) 6378  
*Size of radius of Earth, used in dawn/dusk calculation.*
- #define [pi](#) 3.1415926  
*Value of pi, used in dawn/dusk calculations.*
- #define [planeDist](#) 23.45  
*Distance between x-y plane and ecliptic plane, used in dawn/dusk calculations.*
- #define [distSun](#) 149598000  
*Distance from Earth to sun, used in dawn/dusk calculations.*

### Functions

- double [degToRad](#) (double deg)  
*Convert degrees to radians.*
- int [sign](#) (double x)  
*Checks sign of given value. If longitude value is negative, an offset is taken into account for dawn/dusk calculations.*
- double [dawnCalc](#) (int day, int month, int year, double latitude, double longitude)  
*Calculation of approximate dawn time based on given day and location.*
- double [duskCalc](#) (int day, int month, int year, double latitude, double longitude)  
*Calculation of approximate dusk time based on given day and location.*

#### 5.3.1 Detailed Description

Description: Contains function declarations and constants used in approximate dawn and dusk times.

Author

: Shipra Vaidya

Date

last modified: 4/26/2020

#### 5.3.2 Macro Definition Documentation

### 5.3.2.1 distSun

```
#define distSun 149598000
```

Distance from Earth to sun, used in dawn/dusk calculations.

### 5.3.2.2 pi

```
#define pi 3.1415926
```

Value of pi, used in dawn/dusk calculations.

### 5.3.2.3 planeDist

```
#define planeDist 23.45
```

Distance between x-y plane and ecliptic plane, used in dawn/dusk calculations.

### 5.3.2.4 radius

```
#define radius 6378
```

Size of radius of Earth, used in dawn/dusk calculation.

## 5.3.3 Function Documentation

### 5.3.3.1 dawnCalc()

```
double dawnCalc (
    int day,
    int month,
    int year,
    double latitude,
    double longitude )
```

Calculation of approximate dawn time based on given day and location.

#### Parameters

<i>day</i>	Numerical value, day of month.
<i>month</i>	Numerical value, month in year.
<i>year</i>	Numerical value, year.
<i>latitude</i>	Degree value of latitude from user's location.
<i>longitude</i>	Degree value of longitude from user's location.

**Returns**

double Returns approximate time of dawn on given date at given location.

**5.3.3.2 degToRad()**

```
double degToRad (  
    double deg )
```

Convert degrees to radians.

**Parameters**

<i>deg</i>	Latitude or longitude value in degrees.
------------	---

**Returns**

double Returns converted value in radians.

**5.3.3.3 duskCalc()**

```
double duskCalc (  
    int day,  
    int month,  
    int year,  
    double latitude,  
    double longitude )
```

Calculation of approximate dusk time based on given day and location.

**Parameters**

<i>day</i>	Numerical value, day of month.
<i>month</i>	Numerical value, month in year.
<i>year</i>	Numerical value, year.
<i>latitude</i>	Degree value of latitude from user's location.
<i>longitude</i>	Degree value of longitude from user's location.

**Returns**

double Returns approximate time of dusk on given date at given location.



### 5.3.3.4 sign()

```
int sign (
    double x )
```

Checks sign of given value. If longitude value is negative, an offset is taken into account for dawn/duck calculations.

#### Parameters

x	Value of longitude.
---	---------------------

#### Returns

int Returns 1 if positive, -1 if negative, to be multiplied in offset calculation.

## 5.4 include/decode\_bluetooth.h File Reference

```
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/event_groups.h"
#include "esp_system.h"
```

### Macros

- #define [DECODE\\_BLUETOOTH\\_H](#)

### Functions

- esp\_err\_t [start\\_schedule\\_read](#) (uint8\_t \*packet, uint16\_t handle)
 

*Looks for "Begin Read" from the packet recieved. Saves channel number found in packet in saved\_channel\_num. Calls set\_schedule\_read, passing handle parameter.*
- esp\_err\_t [set\\_schedule\\_read](#) (uint16\_t handle)
 

*Sets GATT characteristic with the respective handle to the value of a schedule. First sets the characteristic to the name of the first schedule. On the next read it will set the characteristic to the schedule information. This will repeat until there are no more schedules in the requested channel (saved\_channel\_num).*
- esp\_err\_t [decode\\_ble\\_schedule](#) (uint8\_t \*packet)
 

*Converts byte packet in order to create schedule object. Recieves name from saved\_name.*
- esp\_err\_t [decode\\_ble\\_schedule\\_name](#) (uint8\_t \*packet, uint16\_t length)
 

*Converts byte packet into string and saves value in saved\_name.*
- esp\_err\_t [decode\\_ble\\_time](#) (uint8\_t \*packet)
 

*Converts byte packet into UNIX time value and calls set\_time function.*
- esp\_err\_t [decode\\_ble\\_direct](#) (uint8\_t \*packet)
 

*Converts byte packet into channel, brightness, and color information. If RGB channel is written, calls set\_color function. If normal LED channel is written, calls channel\_on function.*
- esp\_err\_t [decode\\_ble\\_delete](#) (uint8\_t \*packet)
 

*Converts byte packet into schedule name and channel. The schedule on the corresponding channel is then deleted using delete\_schedule\_by\_name function.*
- uint32\_t [get\\_Int32](#) (uint8\_t \*begin)
 

*Takes byte array and converts into a uint32 using the first 4 indices. Assumes Big-endian.*

### 5.4.1 Detailed Description

Description:

Author

: Hunaid Puri

Date

last modified: 4/26/2020

### 5.4.2 Macro Definition Documentation

#### 5.4.2.1 DECODE\_BLUETOOTH\_H

```
#define DECODE_BLUETOOTH_H
```

### 5.4.3 Function Documentation

#### 5.4.3.1 decode\_ble\_delete()

```
esp_err_t decode_ble_delete (
    uint8_t * packet )
```

Converts byte packet into schedule name and channel. The schedule on the corresponding channel is then deleted using delete\_schedule\_by\_name function.

Parameters

<i>packet</i>	recieved byte packet with schedule name to delete
---------------	---

Returns

esp\_err\_t

#### 5.4.3.2 decode\_ble\_direct()

```
esp_err_t decode_ble_direct (
    uint8_t * packet )
```

Converts byte packet into channel, brightness, and color information. If RGB channel is written, calls set\_color function. If normal LED channel is written, calls channel\_on function.

**Parameters**

<i>packet</i>	recieved byte packet with led control information
---------------	---

**Returns**

esp\_err\_t

**5.4.3.3 decode\_ble\_schedule()**

```
esp_err_t decode_ble_schedule (  
    uint8_t * packet )
```

Converts byte packet in order to create schedule object. Recieves name from saved\_name.

**Parameters**

<i>packet</i>	recieved byte packet with schedule information
---------------	--

**Returns**

esp\_err\_t

**5.4.3.4 decode\_ble\_schedule\_name()**

```
esp_err_t decode_ble_schedule_name (  
    uint8_t * packet,  
    uint16_t length )
```

Converts byte packet into string and saves value in saved\_name.

**Parameters**

<i>packet</i>	recieved byte packet with scheudle name
<i>length</i>	length of recieved packet

**Returns**

esp\_err\_t

#### 5.4.3.5 decode\_ble\_time()

```
esp_err_t decode_ble_time (
    uint8_t * packet )
```

Converts byte packet into UNIX time value and calls set\_time function.

##### Parameters

<i>packet</i>	recieved byte packet with time information
---------------	--

##### Returns

esp\_err\_t

#### 5.4.3.6 get\_Int32()

```
uint32_t get_Int32 (
    uint8_t * begin )
```

Takes byte array and converts into a uint32 using the first 4 indices. Assumes Big-endian.

##### Parameters

<i>begin</i>	byte array
--------------	------------

##### Returns

uint32\_t

#### 5.4.3.7 set\_schedule\_read()

```
esp_err_t set_schedule_read (
    uint16_t handle )
```

Sets GATT characteristic with the respective handle to the value of a schedule. First sets the characteristic to the name of the first schedule. On the next read it will set the characteristic to the schedule information. This will repeat until there are no more schedules in the requested channel (saved\_channel\_num).

##### Parameters

<i>handle</i>	handle for the characteristic that will be read
---------------	---

## Returns

esp\_err\_t

## 5.4.3.8 start\_schedule\_read()

```
esp_err_t start_schedule_read (
    uint8_t * packet,
    uint16_t handle )
```

Looks for "Begin Read" from the packet recieved. Saves channel number found in packet in saved\_channel\_num. Calls set\_schedule read, passing handle parameter.

## Parameters

<i>packet</i>	recieved byte packet
<i>handle</i>	handle for the characteristic that will be read

## Returns

esp\_err\_t

## 5.5 include/espntp.h File Reference

```
#include <time.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/event_groups.h"
#include "esp_log.h"
#include "lwip/apps/sntp.h"
```

## Functions

- void [obtain\\_time](#) (void)
- void [set\\_time](#) (uint32\_t time)

## 5.5.1 Detailed Description

Description: Contains function declarations to start the process of obtaining time from an SNTP server. NOTE: not tested with latest version of project. Was initially developed very early in project and abandoned for other priorities.

## Author

: Jesse Cannon (based on [https://github.com/espressif/esp-idf/blob/5aa21584cfb4cfe4f2c7f623/\\_example\\_main.c](https://github.com/espressif/esp-idf/blob/5aa21584cfb4cfe4f2c7f623/_example_main.c))

## Date

last modified: 4/26/2020

## 5.5.2 Function Documentation

### 5.5.2.1 obtain\_time()

```
void obtain_time (
    void )
```

### 5.5.2.2 set\_time()

```
void set_time (
    uint32_t time )
```

## 5.6 include/gatts\_table\_creat\_demo.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

## Enumerations

- enum {  
    [IDX\\_SVC](#), [IDX\\_CHAR\\_A](#), [IDX\\_CHAR\\_VAL\\_A](#), [IDX\\_CHAR\\_CFG\\_A](#),  
    [IDX\\_CHAR\\_B](#), [IDX\\_CHAR\\_VAL\\_B](#), [IDX\\_CHAR\\_C](#), [IDX\\_CHAR\\_VAL\\_C](#),  
    [HRS\\_IDX\\_NB](#) }

## 5.6.1 Enumeration Type Documentation

### 5.6.1.1 anonymous enum

anonymous enum

#### Enumerator

IDX_SVC	
IDX_CHAR_A	
IDX_CHAR_VAL_A	
IDX_CHAR_CFG_A↔ _A	
IDX_CHAR_B	
IDX_CHAR_VAL_B	
IDX_CHAR_C	

## 5.7 include/http.h File Reference

```
#include <string.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/event_groups.h"
#include "esp_system.h"
#include "esp_wifi.h"
#include "esp_event_loop.h"
#include "esp_log.h"
#include "esp_http_server.h"
```

### Functions

- `esp_err_t init_http` (`httpd_handle_t` server)  
*Initializes a http server with other handlers defined the file.*
- `esp_err_t homepage_handler` (`httpd_req_t *req`)  
*Handler to retrieve home/direct control page. Accessed when server receives "/" or "/index.html".*
- `esp_err_t styles_handler` (`httpd_req_t *req`)  
*Handler to retrieve styles CSS page. Accessed when server receives "/styles.css".*
- `esp_err_t schedules_handler` (`httpd_req_t *req`)  
*Handler to retrieve schedules page. Accessed when server receives "/schedules.html".*
- `esp_err_t scripts_handler` (`httpd_req_t *req`)  
*Handler to retrieve scripts page. Accessed when server receives "/schedules.js".*
- `esp_err_t schedule_post_handler` (`httpd_req_t *req`)  
*Handler to create new schedule. Accessed when server receives or "/post\_sch".*
- `esp_err_t favicon_ico_get_handler` (`httpd_req_t *req`)  
*Handler to retrieve favicon. Accessed when server receives "/favicon.ico" - DOES NOT WORK.*
- `esp_err_t time_post_handler` (`httpd_req_t *req`)  
*Handler to post time to esp32. Accessed when server receives "/time".*
- `esp_err_t direct_control_post_handler` (`httpd_req_t *req`)  
*Handler for direct control. Access when server receives "/direct\_control".*
- `esp_err_t sch_data_post_handler` (`httpd_req_t *req`)  
*Handler to retrieve schedule data Access when server receives "/sch\_data".*
- `void schTokenProcess` (`char *str`)  
*Function to process ";;"-delimited string. Parses out scheduling data and for schedule post handler.*

### 5.7.1 Detailed Description

Description:

Author

: Andy Yeung

Date

: 4/26/2020

## 5.7.2 Function Documentation

### 5.7.2.1 direct\_control\_post\_handler()

```
esp_err_t direct_control_post_handler (  
    httpd_req_t * req )
```

Handler for direct control. Access when server receives "/direct\_control".

#### Parameters

<i>req</i>	An incoming HTTP request.
------------	---------------------------

#### Returns

`esp_err_t` Returns `ESP_OK` on successful deletion. `ESP_ERR_INVALID_ARG` if channel number is invalid. `ESP_ERR_NOT_FOUND` if schedule not found

### 5.7.2.2 favicon\_ico\_get\_handler()

```
esp_err_t favicon_ico_get_handler (  
    httpd_req_t * req )
```

Handler to retrieve favicon. Accessed when server receives "/favicon.ico" - DOES NOT WORK.

#### Parameters

<i>req</i>	An incoming HTTP request.
------------	---------------------------

#### Returns

`esp_err_t` Returns `ESP_OK` on successful deletion. `ESP_ERR_INVALID_ARG` if channel number is invalid. `ESP_ERR_NOT_FOUND` if schedule not found

### 5.7.2.3 homepage\_handler()

```
esp_err_t homepage_handler (  
    httpd_req_t * req )
```

Handler to retrieve home/direct control page. Accessed when server receives "/" or "/index.html".



## Parameters

<i>req</i>	An incoming HTTP request.
------------	---------------------------

## Returns

`esp_err_t` Returns `ESP_OK` on successful deletion. `ESP_ERR_INVALID_ARG` if channel number is invalid.  
`ESP_ERR_NOT_FOUND` if schedule not found

### 5.7.2.4 `init_http()`

```
esp_err_t init_http (  
    httpd_handle_t server )
```

Initializes a http server with other handlers defined the file.

## Parameters

<i>server</i>	A handle for the server.
---------------	--------------------------

## Returns

`esp_err_t` Returns `ESP_OK` on successful deletion. `ESP_ERR_INVALID_ARG` if channel number is invalid.  
`ESP_ERR_NOT_FOUND` if schedule not found

### 5.7.2.5 `sch_data_post_handler()`

```
esp_err_t sch_data_post_handler (  
    httpd_req_t * req )
```

Handler to retrieve schedule data Access when server receives `"/sch_data"`.

## Parameters

<i>req</i>	An incoming HTTP request.
------------	---------------------------

## Returns

`esp_err_t` Returns `ESP_OK` on successful deletion. `ESP_ERR_INVALID_ARG` if channel number is invalid.  
`ESP_ERR_NOT_FOUND` if schedule not found

### 5.7.2.6 `schedule_post_handler()`

```
esp_err_t schedule_post_handler (  
    httpd_req_t * req )
```

Handler to create new schedule. Accessed when server receives or "/post\_sch".

#### Parameters

<i>req</i>	An incoming HTTP request.
------------	---------------------------

#### Returns

`esp_err_t` Returns `ESP_OK` on successful deletion. `ESP_ERR_INVALID_ARG` if channel number is invalid. `ESP_ERR_NOT_FOUND` if schedule not found

### 5.7.2.7 `schedules_handler()`

```
esp_err_t schedules_handler (  
    httpd_req_t * req )
```

Handler to retrieve schedules page. Accessed when server receives "/schedules.html".

#### Parameters

<i>req</i>	An incoming HTTP request.
------------	---------------------------

#### Returns

`esp_err_t` Returns `ESP_OK` on successful deletion. `ESP_ERR_INVALID_ARG` if channel number is invalid. `ESP_ERR_NOT_FOUND` if schedule not found

### 5.7.2.8 `schTokenProcess()`

```
void schTokenProcess (  
    char * str )
```

Function to process ";;"-delimited string. Parses out scheduling data and for schedule post handler.

#### Parameters

<i>str</i>	Received string
------------	-----------------

### 5.7.2.9 scripts\_handler()

```
esp_err_t scripts_handler (  
    httpd_req_t * req )
```

Handler to retrieve scripts page. Accessed when server receives "/schedules.js".

#### Parameters

<i>req</i>	An incoming HTTP request.
------------	---------------------------

#### Returns

`esp_err_t` Returns `ESP_OK` on successful deletion. `ESP_ERR_INVALID_ARG` if channel number is invalid. `ESP_ERR_NOT_FOUND` if schedule not found

### 5.7.2.10 styles\_handler()

```
esp_err_t styles_handler (  
    httpd_req_t * req )
```

Handler to retrieve styles CSS page. Accessed when server receives "/styles.css".

#### Parameters

<i>req</i>	An incoming HTTP request.
------------	---------------------------

#### Returns

`esp_err_t` Returns `ESP_OK` on successful deletion. `ESP_ERR_INVALID_ARG` if channel number is invalid. `ESP_ERR_NOT_FOUND` if schedule not found

### 5.7.2.11 time\_post\_handler()

```
esp_err_t time_post_handler (  
    httpd_req_t * req )
```

Handler to post time to esp32. Accessed when server receives "/time".

#### Parameters

<i>req</i>	An incoming HTTP request.
------------	---------------------------

**Returns**

esp\_err\_t Returns ESP\_OK on successful deletion. ESP\_ERR\_INVALID\_ARG if channel number is invalid. ESP\_ERR\_NOT\_FOUND if schedule not found

**5.8 include/led.h File Reference**

```
#include <stdio.h>
#include "esp_system.h"
#include "esp_log.h"
#include "driver/ledc.h"
```

**Macros**

- #define [GPIO\\_CHANNEL\\_0](#) 32  
*Pin definitions for tech demo, needs to be changed for PCB.*
- #define [GPIO\\_CHANNEL\\_1](#) 33
- #define [GPIO\\_CHANNEL\\_2](#) 25
- #define [GPIO\\_CHANNEL\\_3](#) 19
- #define [GPIO\\_CHANNEL\\_4](#) 18
- #define [GPIO\\_CHANNEL\\_5](#) 5
- #define [NUM\\_CHANNELS](#) 6  
*Six pwm channels supported as either six individual channels, two RGB channels, or combination of two.*

**Functions**

- esp\_err\_t [shutdown\\_outputs](#) (void)  
*Shuts down all channels and prevents them from turning on again. Enter a permanent shutdown state until clear\_↔ shutdown is called or device is reset.*
- esp\_err\_t [clear\\_shutdown](#) (void)  
*Clears a shutdown state, allowing outputs to be controlled again.*
- void [init\\_channels](#) (void)  
*Initialize channels to set all timer and channel configurations.*
- void [channel\\_on](#) (uint8\_t index, uint8\_t brightness)  
*Sets brightness of individual channel to turn that output on.*
- void [channel\\_off](#) (uint8\_t index)  
*Sets brightness of individual channel to 0 to turn off channel output.*
- void [set\\_color](#) (uint8\_t index, uint16\_t r, uint16\_t g, uint16\_t b, uint8\_t brightness)  
*Controls RGB channel by changing R, G, and B based off chosen brightness.*

**5.8.1 Detailed Description**

Description: Contains function declarations and settings to configure GPIO outputs to be used as PWM control for LED strips.

**Author**

: Primary: Shipra Vaidya, Secondary: Jesse Cannon (shutdown features)

**Date**

last modified: 4/26/2020

## 5.8.2 Macro Definition Documentation

### 5.8.2.1 GPIO\_CHANNEL\_0

```
#define GPIO_CHANNEL_0 32
```

Pin definitions for tech demo, needs to be changed for PCB.

### 5.8.2.2 GPIO\_CHANNEL\_1

```
#define GPIO_CHANNEL_1 33
```

### 5.8.2.3 GPIO\_CHANNEL\_2

```
#define GPIO_CHANNEL_2 25
```

### 5.8.2.4 GPIO\_CHANNEL\_3

```
#define GPIO_CHANNEL_3 19
```

### 5.8.2.5 GPIO\_CHANNEL\_4

```
#define GPIO_CHANNEL_4 18
```

### 5.8.2.6 GPIO\_CHANNEL\_5

```
#define GPIO_CHANNEL_5 5
```

### 5.8.2.7 NUM\_CHANNELS

```
#define NUM_CHANNELS 6
```

Six pwm channels supported as either six individual channels, two RGB channels, or combination of two.

## 5.8.3 Function Documentation

### 5.8.3.1 channel\_off()

```
void channel_off (
    uint8_t index )
```

Sets brightness of individual channel to 0 to turn off channel output.

**Parameters**

<i>index</i>	Value between 0 and 5 corresponding to channel being controlled.
--------------	--

**5.8.3.2 channel\_on()**

```
void channel_on (
    uint8_t index,
    uint8_t brightness )
```

Sets brightness of individual channel to turn that output on.

**Parameters**

<i>index</i>	Value between 0 and 5 corresponding to channel being controlled.
<i>brightness</i>	Value of brightness chosen per channel.

**5.8.3.3 clear\_shutdown()**

```
esp_err_t clear_shutdown (
    void )
```

Clears a shutdown state, allowing outputs to be controlled again.

**Returns**

esp\_err\_t ESP\_OK on success, ESP\_FAIL if not in shutdown state.

**5.8.3.4 init\_channels()**

```
void init_channels (
    void )
```

Initialize channels to set all timer and channel configurations.

**5.8.3.5 set\_color()**

```
void set_color (
    uint8_t index,
    uint16_t r,
    uint16_t g,
    uint16_t b,
    uint8_t brightness )
```

Controls RGB channel by changing R, G, and B based off chosen brightness.

## Parameters

<i>index</i>	Value between 0 and 1 corresponding to two supported RGB channel outputs repectively.
<i>r</i>	R value of chosen color.
<i>g</i>	G value of chosen color.
<i>b</i>	B value of chosen color.
<i>brightness</i>	Value of brightness chosen for RGB channel.

**5.8.3.6 shutdown\_outputs()**

```
esp_err_t shutdown_outputs (
    void )
```

Shuts down all channels and prevents them from turning on again. Enter a permanent shutdown state until clear↔\_shutdown is called or device is reset.

## Returns

esp\_err\_t ESP\_OK on success, ESP\_FAIL if already in shutdown state.

**5.9 include/main.h File Reference****5.9.1 Detailed Description**

Description: Contains code version of latest build. Used by memory subsystem to keep old versions separated from new versions so that no conflicts occur with updates.

## Author

: Jesse Cannon

## Date

last modified: 4/26/2020

**5.10 include/measurement.h File Reference**

```
#include "driver/gpio.h"
#include "driver/adc.h"
#include "driver/dac.h"
#include "esp_system.h"
#include "esp_log.h"
#include "pin_defs.h"
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "led.h"
#include "memory.h"
```

## Functions

- `esp_err_t init_oc` (void)  
*Init the overcurrent and overvoltage detection systems. The previously used values will be set as the level, with defaults of 24 Volts and 2 Amps triggering faults.*
- `esp_err_t set_current_level` (double ampLimit)  
*Set the current level that will trigger a fault.*
- `esp_err_t set_voltage_level` (double voltLimit)  
*Set the voltage level that will trigger a fault.*
- `uint8_t isCurrentFault` ()  
*Determines if the device is in shutdown from an overcurrent condition.*
- `uint8_t isVoltageFault` ()  
*Determines if the device is in shutdown from an overvoltage condition.*
- `esp_err_t clearFaults` ()  
*Clears all faults from the system and allows outputs to be controlled again.*

### 5.10.1 Detailed Description

Description: Contains function declarations for OC/OV subsystem. Intended to set levels and take actions for overcurrent and overvoltage conditions. NOTE: this system is basically just an outline. Development was not completed b/c of a lack of hardware and time constraints.

#### Author

: Jesse Cannon

#### Date

last modified: 4/26/2020

### 5.10.2 Function Documentation

#### 5.10.2.1 clearFaults()

```
esp_err_t clearFaults ( )
```

Clears all faults from the system and allows outputs to be controlled again.

#### Returns

`esp_err_t ESP_OK` if successfully re-enables outputs. `ESP_FAIL` if error occurs.



### 5.10.2.2 init\_oc()

```
esp_err_t init_oc (  
    void )
```

Initializes the overcurrent and overvoltage detection systems. The previously used values will be set as the level, with defaults of 24 Volts and 2 Amps triggering faults.

#### Returns

esp\_err\_t

### 5.10.2.3 isCurrentFault()

```
uint8_t isCurrentFault ( )
```

Determines if the device is in shutdown from an overcurrent condition.

#### Returns

uint8\_t A value of 1 means that there is currently a fault. A value of 0 means there is no fault

### 5.10.2.4 isVoltageFault()

```
uint8_t isVoltageFault ( )
```

Determines if the device is in shutdown from an overvoltage condition.

#### Returns

uint8\_t A value of 1 means that there is currently a fault. A value of 0 means there is no fault

### 5.10.2.5 set\_current\_level()

```
esp_err_t set_current_level (  
    double ampLimit )
```

Set the current level that will trigger a fault.

#### Parameters

<i>ampLimit</i>	The current level to be used to determine faults, such as 1.1 Amps.
-----------------	---

**Returns**

esp\_err\_t ESP\_OK if the new level was successfully set, else if error occurred.

**5.10.2.6 set\_voltage\_level()**

```
esp_err_t set_voltage_level (
    double voltLimit )
```

Set the voltage level that will trigger a fault.

**Parameters**

<i>voltLimit</i>	The voltage level to be used to determine faults, such as 12.5 Volts.
------------------	---

**Returns**

esp\_err\_t ESP\_OK if the new level was successfully set, else if error occurred.

**5.11 include/memory.h File Reference**

```
#include "ArduinoJson-v6.14.1.h"
#include <stdio.h>
#include <string.h>
#include <sys/unistd.h>
#include <sys/stat.h>
#include <dirent.h>
#include "main.h"
#include "esp_err.h"
#include "esp_log.h"
#include "esp_spiffs.h"
#include "scheduler.h"
```

**Macros**

- #define [SETTINGS\\_BUFFER\\_SIZE](#) 512  
*The size of the buffer used to store settings. Increase this if storing a lot of settings.*

**Functions**

- esp\_err\_t [init\\_memory](#) (void)  
*Starts the memory system. Call this before calling any other memory related functions.*
- esp\_err\_t [store\\_schedules](#) (void)  
*Stores all currently running schedules in persistent memory.*
- esp\_err\_t [recall\\_schedules](#) (void)

*Recall schedules from persistent memory. All schedules recalled will now be located in the respective channel's schedule list.*

- `esp_err_t clear_schedule_data` (void)

*Clears all schedules from persistent memory. Does not delete the schedules from the list.*

- `esp_err_t store_setting_string` (const char \*name, char \*setting)

*Persistently store a string setting.*

- `esp_err_t store_setting_int` (const char \*name, int setting)

*Persistently store an integer setting.*

- `esp_err_t store_setting_byte` (const char \*name, uint8\_t setting)

*Persistently store a byte setting.*

- `esp_err_t store_setting_double` (const char \*name, double setting)

*Persistently store a double setting.*

- `esp_err_t get_setting_string` (const char \*name, char \*setting)

*Recall a persistent string setting.*

- `esp_err_t get_setting_int` (const char \*name, int \*setting)

*Recall a persistent integer setting.*

- `esp_err_t get_setting_byte` (const char \*name, uint8\_t \*setting)

*Recall a byte string setting. Uint8\_t is equivalent to unsigned char.*

- `esp_err_t get_setting_double` (const char \*name, double \*setting)

*Recall a persistent double setting. Exact storage depends on ArduinoJson configuration, but typically a maximum of 9 decimal places with rounding. No trailing zeros are included.*

- `esp_err_t clear_setting_data` (void)

*Clear settings from persistent memory and RAM.*

### 5.11.1 Detailed Description

Description: Contains function declarations and settings used by the memory subsystem. Functions are primarily used by other files to get and retrieve values from persistent memory.

Author

: Jesse Cannon

Date

last modified: 4/26/2020

### 5.11.2 Macro Definition Documentation

#### 5.11.2.1 SETTINGS\_BUFFER\_SIZE

```
#define SETTINGS_BUFFER_SIZE 512
```

The size of the buffer used to store settings. Increase this if storing a lot of settings.

### 5.11.3 Function Documentation

#### 5.11.3.1 `clear_schedule_data()`

```
esp_err_t clear_schedule_data (  
    void )
```

Clears all schedules from persistent memory. Does not delete the schedules from the list.

##### Returns

`esp_err_t` Returns `ESP_OK` on success. `ESP_Fail` or related if an error occurred.

#### 5.11.3.2 `clear_setting_data()`

```
esp_err_t clear_setting_data (  
    void )
```

Clear settings from persistent memory and RAM.

##### Returns

`esp_err_t` `ESP_OK` on success. `ESP_FAIL` if error occurs.

#### 5.11.3.3 `get_setting_byte()`

```
esp_err_t get_setting_byte (  
    const char * name,  
    uint8_t * setting )
```

Recall a byte string setting. `uint8_t` is equivalent to unsigned char.

##### Parameters

<i>name</i>	The name of the setting to recall.
<i>setting</i>	The output of that setting. Set to NULL if not found, the value if found.

##### Returns

`esp_err_t` Returns `ESP_OK` on success. `ESP_Fail` if not found or an error occurred.

#### 5.11.3.4 `get_setting_double()`

```
esp_err_t get_setting_double (
    const char * name,
    double * setting )
```

Recall a persistent double setting. Exact storage depends on ArduinoJson configuration, but typically a maximum of 9 decimal places with rounding. No trailing zeros are included.

##### Parameters

<i>name</i>	The name of the setting to recall.
<i>setting</i>	The output of that setting. Set to NULL if not found, the value if found.

##### Returns

`esp_err_t` Returns `ESP_OK` on success. `ESP_Fail` if not found or an error occurred.

#### 5.11.3.5 `get_setting_int()`

```
esp_err_t get_setting_int (
    const char * name,
    int * setting )
```

Recall a persistent integer setting.

##### Parameters

<i>name</i>	The name of the setting to recall.
<i>setting</i>	The output of that setting. Set to NULL if not found, the value if found.

##### Returns

`esp_err_t` Returns `ESP_OK` on success. `ESP_Fail` if not found or an error occurred.

#### 5.11.3.6 `get_setting_string()`

```
esp_err_t get_setting_string (
    const char * name,
    char * setting )
```

Recall a persistent string setting.

**Parameters**

<i>name</i>	The name of the setting to recall.
<i>setting</i>	The output of that setting. Set to NULL if not found, the value if found.

**Returns**

`esp_err_t` Returns `ESP_OK` on success. `ESP_Fail` if not found or an error occurred.

**5.11.3.7 init\_memory()**

```
esp_err_t init_memory (  
    void )
```

Starts the memory system. Call this before calling any other memory related functions.

**Returns**

`esp_err_t` Returns `ESP_OK` on success. `ESP_Fail` or related if an error occurred.

**5.11.3.8 recall\_schedules()**

```
esp_err_t recall_schedules (  
    void )
```

Recall schedules from persistent memory. All schedules recalled will now be located in the tespective channel's schedule list.

**Returns**

`esp_err_t` Returns `ESP_OK` on success. `ESP_Fail` or related if an error occurred.

**5.11.3.9 store\_schedules()**

```
esp_err_t store_schedules (  
    void )
```

Stores all currently running schedules in persistent memory.

**Returns**

`esp_err_t` Returns `ESP_OK` on success. `ESP_Fail` or related if an error occurred.

**5.11.3.10 store\_setting\_byte()**

```
esp_err_t store_setting_byte (  
    const char * name,  
    uint8_t setting )
```

Persistently store a byte setting.

## Parameters

<i>name</i>	The name of the setting to store. If the setting with that name already exists it is overwritten with the new value.
<i>setting</i>	The setting to associate with the given name. <code>uint8_t</code> is equivalent to an unsigned char.

## Returns

`esp_err_t` Returns `ESP_OK` on success. `ESP_Fail` or related if an error occurred.

**5.11.3.11 store\_setting\_double()**

```
esp_err_t store_setting_double (
    const char * name,
    double setting )
```

Persistently store a double setting.

## Parameters

<i>name</i>	The name of the setting to store. If the setting with that name already exists it is overwritten with the new value.
<i>setting</i>	The setting to associate with the given name. Exact storage depends on ArduinoJson configuration, but typically a maximum of 9 decimal places with rounding. No trailing zeros are included

## Returns

`esp_err_t` Returns `ESP_OK` on success. `ESP_Fail` or related if an error occurred.

**5.11.3.12 store\_setting\_int()**

```
esp_err_t store_setting_int (
    const char * name,
    int setting )
```

Persistently store an integer setting.

## Parameters

<i>name</i>	The name of the setting to store. If the setting with that name already exists it is overwritten with the new value.
<i>setting</i>	The setting to associate with the given name.

#### Returns

esp\_err\_t Returns ESP\_OK on success. ESP\_Fail or related if an error occurred.

#### 5.11.3.13 store\_setting\_string()

```
esp_err_t store_setting_string (
    const char * name,
    char * setting )
```

Persistently store a string setting.

#### Parameters

<i>name</i>	The name of the setting to store. If the setting with that name already exists it is overwritten with the new value.
<i>setting</i>	The setting to associate with the given name.

#### Returns

esp\_err\_t Returns ESP\_OK on success. ESP\_Fail or related if an error occurred.

## 5.12 include/pin\_defs.h File Reference

### Macros

- #define CH3\_HIGH 39
- #define CH3\_LOW 36
- #define DAC1 1
- #define DAC2 26
- #define OC\_ENABLE GPIO\_NUM\_22
- #define OC\_ALERT GPIO\_NUM\_26
- #define OC\_LATCH
- #define OC\_LIMIT DAC1

#### 5.12.1 Detailed Description

Description: Pin definitions for external hardware. Used by measurement subsystem for INA300 for overcurrent detection

#### Author

: Jesse Cannon

#### Date

last modified: 4/26/2020



## 5.12.2 Macro Definition Documentation

### 5.12.2.1 CH3\_HIGH

```
#define CH3_HIGH 39
```

### 5.12.2.2 CH3\_LOW

```
#define CH3_LOW 36
```

### 5.12.2.3 DAC1

```
#define DAC1 1
```

### 5.12.2.4 DAC2

```
#define DAC2 26
```

### 5.12.2.5 OC\_ALERT

```
#define OC_ALERT GPIO_NUM_26
```

### 5.12.2.6 OC\_ENABLE

```
#define OC_ENABLE GPIO_NUM_22
```

### 5.12.2.7 OC\_LATCH

```
#define OC_LATCH
```

### 5.12.2.8 OC\_LIMIT

```
#define OC_LIMIT DAC1
```

## 5.13 include/rtcdefine.h File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "esp_system.h"
#include "esp_timer.h"
#include "esp_log.h"
#include <time.h>
#include <sys/time.h>
#include "driver/spi_master.h"
#include "esp_spi_flash.h"
#include "sdkconfig.h"
```

### Macros

- #define [HSPI\\_MISO](#) 12
- #define [HSPI\\_MOSI](#) 13
- #define [HSPI\\_CLK](#) 14
- #define [HSPI\\_CS](#) 15
- #define [EEREAD](#) 0x03
- #define [EEWRITE](#) 0x02
- #define [EEWRDI](#) 0x04
- #define [EEWREN](#) 0x06
- #define [SRREAD](#) 0x05
- #define [SSWRITE](#) 0x01
- #define [RTC\\_READ](#) 0x13
- #define [RTC\\_WRITE](#) 0x12
- #define [RTC\\_UNLOCK](#) 0x14
- #define [IDWRITE](#) 0x32
- #define [IDREAD](#) 0x33
- #define [CLRRAM](#) 0x54
- #define [RTCHSEC](#) 0x00
- #define [RTCSEC](#) 0x01
- #define [RTCMIN](#) 0x02
- #define [RTCHOUR](#) 0x03
- #define [RTCWKDAY](#) 0x04
- #define [RTCDATE](#) 0x05
- #define [RTCMTH](#) 0x06
- #define [RTCYEAR](#) 0x07
- #define [CONTROL](#) 0x08
- #define [OSCTRIM](#) 0x09
- #define [PWRDNMIN](#) 0x18
- #define [PWRDNHOUR](#) 0x19
- #define [PWRDNDATE](#) 0x1A

- #define `PWRDNMONTH` 0x1B
- #define `ST` 0x80
- #define `LPYR` 0x20
- #define `PM` 0x20
- #define `H12` 0x40
- #define `TRIMSIGN` 0x80
- #define `OUT` 0x80
- #define `SQWEN` 0x40
- #define `OSCRUN` 0x20
- #define `VBATEN` 0x08
- #define `PWRFAIL` 0x10
- #define `RTC_TAG` "RTC"

## Functions

- `esp_err_t setTime` (const struct tm \*time)  
*Set the time of the external RTC to the specified value.*
- `esp_err_t getTime` (struct tm \*outTime)  
*Get the current time on the external RTC.*
- void `RTCHandler` (void \*pvParms)  
*The RTOS task that handles RTC related actions. This task synchronizes the esp32's and RTC's time.*
- `esp_err_t ST_StartRTCHandler` (void)  
*Starts the RTOS task that handles RTC and esp32 time synchronization.*

## Variables

- TaskHandle\_t `RTC_Handle`  
*The task handle for the RTC task. If this field is != NULL then task is currently running. Can be used to stop task with RTOS functions.*
- struct tm \* `currTime`  
*A copy of the most recent time obtained from the RTC. Is not guaranteed to be accurate to the actual time stored on the RTC. This value is updated as often as the RTC task runs.*

### 5.13.1 Detailed Description

Description: Contains function and pin definitions for the time manager subsystem. Controls external RTC MCP79510. see: <http://ww1.microchip.com/downloads/en/DeviceDoc/MCP7951X-MCP7952X-Battery-Backed-SPI-RTCC-20002300C.pdf>

#### Author

: Lead: Shipra Vaidya, Seoncdary: Jesse Cannon

#### Date

last modified: 4/26/2020

### 5.13.2 Macro Definition Documentation

#### 5.13.2.1 CLRRAM

```
#define CLRRAM 0x54
```

#### 5.13.2.2 CONTROL

```
#define CONTROL 0x08
```

#### 5.13.2.3 EEREAD

```
#define EEREAD 0x03
```

#### 5.13.2.4 EEWRDI

```
#define EEWRDI 0x04
```

#### 5.13.2.5 EEWREN

```
#define EEWREN 0x06
```

#### 5.13.2.6 EEWRITE

```
#define EEWRITE 0x02
```

#### 5.13.2.7 H12

```
#define H12 0x40
```

#### 5.13.2.8 HSPI\_CLK

```
#define HSPI_CLK 14
```

#### 5.13.2.9 HSPI\_CS

```
#define HSPI_CS 15
```

#### 5.13.2.10 HSPI\_MISO

```
#define HSPI_MISO 12
```

#### 5.13.2.11 HSPI\_MOSI

```
#define HSPI_MOSI 13
```

#### 5.13.2.12 IDREAD

```
#define IDREAD 0x33
```

#### 5.13.2.13 IDWRITE

```
#define IDWRITE 0x32
```

#### 5.13.2.14 LPYR

```
#define LPYR 0x20
```

#### 5.13.2.15 OSCRUN

```
#define OSCRUN 0x20
```

#### 5.13.2.16 OSCTRIM

```
#define OSCTRIM 0x09
```

**5.13.2.17 OUT**

```
#define OUT 0x80
```

**5.13.2.18 PM**

```
#define PM 0x20
```

**5.13.2.19 PWRDNDATE**

```
#define PWRDNDATE 0x1A
```

**5.13.2.20 PWRDNHOUR**

```
#define PWRDNHOUR 0x19
```

**5.13.2.21 PWRDNMIN**

```
#define PWRDNMIN 0x18
```

**5.13.2.22 PWRDNMONTH**

```
#define PWRDNMONTH 0x1B
```

**5.13.2.23 PWRFAIL**

```
#define PWRFAIL 0x10
```

**5.13.2.24 RTC\_READ**

```
#define RTC_READ 0x13
```

#### 5.13.2.25 RTC\_TAG

```
#define RTC_TAG "RTC"
```

#### 5.13.2.26 RTC\_UNLOCK

```
#define RTC_UNLOCK 0x14
```

#### 5.13.2.27 RTC\_WRITE

```
#define RTC_WRITE 0x12
```

#### 5.13.2.28 RTCDATE

```
#define RTCDATE 0x05
```

#### 5.13.2.29 RTCHOUR

```
#define RTCHOUR 0x03
```

#### 5.13.2.30 RTCHSEC

```
#define RTCHSEC 0x00
```

#### 5.13.2.31 RTCMIN

```
#define RTCMIN 0x02
```

#### 5.13.2.32 RTCMTH

```
#define RTCMTH 0x06
```

**5.13.2.33 RTCSEC**

```
#define RTCSEC 0x01
```

**5.13.2.34 RTCWKDAY**

```
#define RTCWKDAY 0x04
```

**5.13.2.35 RTCYEAR**

```
#define RTCYEAR 0x07
```

**5.13.2.36 SQWEN**

```
#define SQWEN 0x40
```

**5.13.2.37 SRREAD**

```
#define SRREAD 0x05
```

**5.13.2.38 SSWRITE**

```
#define SSWRITE 0x01
```

**5.13.2.39 ST**

```
#define ST 0x80
```

**5.13.2.40 TRIMSIGN**

```
#define TRIMSIGN 0x80
```



#### 5.13.2.41 VBATEN

```
#define VBATEN 0x08
```

### 5.13.3 Function Documentation

#### 5.13.3.1 getTime()

```
esp_err_t getTime (  
    struct tm * outTime )
```

Get the current time on the external RTC.

##### Parameters

<i>outTime</i>	The current time on the RTC will be stored in this value. Check the return type against ESP_OK to ensure this value was set correctly.
----------------	--

##### Returns

esp\_err\_t ESP\_OK on success, else on failure.

#### 5.13.3.2 RTCHandler()

```
void RTCHandler (  
    void * pvParams )
```

The RTOS task that handles RTC related actions. This task synchronizes the esp32's and RTC's time.

##### Parameters

<i>pvParams</i>	Required parameter of RTOS tasks. Not used in this task.
-----------------	--

#### 5.13.3.3 setTime()

```
esp_err_t setTime (  
    const struct tm * time )
```

Set the time of the external RTC to the specified value.

**Parameters**

<i>time</i>	The time to be set on the device.
-------------	-----------------------------------

**Returns**

esp\_err\_t ESP\_OK on success, else on failure.

**5.13.3.4 ST\_StartRTCHandler()**

```
esp_err_t ST_StartRTCHandler (
    void )
```

Starts the RTOS task that handles RTC and esp32 time synchronization.

**Returns**

esp\_err\_t ESP\_OK if task started successfully, else on failure.

**5.13.4 Variable Documentation****5.13.4.1 currTime**

```
struct tm* currTime
```

A copy of the most recent time obtained from the RTC. Is not guaranteed to be accurate to the actual time stored on the RTC. This value is updated as often as the RTC task runs.

**5.13.4.2 RTC\_Handle**

```
TaskHandle_t RTC_Handle
```

The task handle for the RTC task. If this field is != NULL then task is currently running. Can be used to stop task with RTOS functions.

**5.14 include/schedule\_object.h File Reference**

```
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "esp_system.h"
```

## Classes

- struct [Schedule\\_Object](#)  
*data object that specified parameters a schedule can contain*
- struct [Node](#)  
*Nodes of a singly linked list data structure.*

## Typedefs

- typedef struct [Schedule\\_Object](#) [schedule\\_object](#)  
*data object that specified parameters a schedule can contain*
- typedef struct [Node](#) [List](#)  
*Nodes of a singly linked list data structure.*

### 5.14.1 Detailed Description

Description: Contains the definition of the data object used for schedules in the scheduler subsystem. Defines a simple linked list data structure to contain schedule data object.

#### Author

: Jesse Cannon

#### Date

last modified: 4/26/2020

### 5.14.2 Typedef Documentation

#### 5.14.2.1 List

```
typedef struct Node List
```

Nodes of a singly linked list data structure.

#### 5.14.2.2 schedule\_object

```
typedef struct Schedule\_Object schedule\_object
```

data object that specified parameters a schedule can contain

## 5.15 include/scheduler.h File Reference

```
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "esp_system.h"
#include "esp_log.h"
#include <time.h>
#include <string.h>
#include "schedule_object.h"
#include "led.h"
#include "dawndusk.h"
#include "memory.h"
```

### Functions

- void [init\\_schedule](#) (void)  
*Init the scheduler code. Starts the background task if not already started. If this is not called the schedules will not run.*
- esp\_err\_t [create\\_schedule](#) (uint8\_t [channel](#), [schedule\\_object](#) s)  
*Create a schedule to associate with the specified channel. If the schedule already exists then it is replaced with the new schedule.*
- esp\_err\_t [delete\\_schedule\\_by\\_id](#) (uint8\_t [channel](#), uint8\_t ID)  
*Delete a schedule by ID.*
- esp\_err\_t [delete\\_schedule\\_by\\_name](#) (uint8\_t [channel](#), char \*name)  
*Delete a schedule by name.*
- esp\_err\_t [disable\\_schedule\\_by\\_id](#) (uint8\_t [channel](#), uint8\_t ID)  
*Disable a schedule by ID. The schedule remains in the linked list but does not run or perform updates.*
- esp\_err\_t [disable\\_schedule\\_by\\_name](#) (uint8\_t [channel](#), char \*name)  
*Disable a schedule by name. The schedule remains in the linked list but does not run or perform updates.*
- esp\_err\_t [enable\\_schedule\\_by\\_id](#) (uint8\_t [channel](#), uint8\_t ID)  
*Enable a schedule by ID. If the schedule was disabled, the schedule now runs and performs updates.*
- esp\_err\_t [enable\\_schedule\\_by\\_name](#) (uint8\_t [channel](#), char \*name)  
*Enable a schedule by name. If the schedule was disabled, the schedule now runs and performs updates.*
- esp\_err\_t [disable\\_all\\_schedules](#) (void)  
*Disables all schedules on all channels. The schedules remain located in memory but no longer run or perform updates.*
- esp\_err\_t [enable\\_all\\_schedules](#) (void)  
*Enable all schedules on all channels. If the schedule was disabled, the schedule now runs and performs updates.*
- esp\_err\_t [delete\\_all\\_schedules](#) (void)  
*Delete all schedules on all channels.*
- esp\_err\_t [get\\_schedule\\_names](#) (uint8\_t [channel](#), char \*&out)  
*Get a json string of all schedules and their enabled status for a given channel.*
- esp\_err\_t [get\\_schedule](#) (uint8\_t [channel](#), char \*name, [schedule\\_object](#) \*out)  
*Get the schedule object specified.*

### Variables

- List \* [schedules](#) [NUM\_CHANNELS]  
*The linked list associated with each channel. Can be used directly instead of scheduler functions if so desired.*

### 5.15.1 Detailed Description

Description: Contains functions available to start and manipulate the scheduler subsystem.

Author

: Jesse Cannon

Date

: 4/26/2020

### 5.15.2 Function Documentation

#### 5.15.2.1 create\_schedule()

```
esp_err_t create_schedule (
    uint8_t channel,
    schedule_object s )
```

Create a schedule to associate with the specified channel. If the schedule already exists then it is replaced with the new schedule.

Parameters

<i>channel</i>	The channel that the schedule will be placed on.
<i>s</i>	The schedule object that will be placed.

Returns

esp\_err\_t Returns ESP\_OK on success. ESP\_ERR\_INVALID\_ARG if channel number is invalid. ESP\_ERR\_NO\_MEM if unable to allocate schedule.

#### 5.15.2.2 delete\_all\_schedules()

```
esp_err_t delete_all_schedules (
    void )
```

Delete all schedules on all channels.

Returns

esp\_err\_t ESP\_OK on successful deletion.

### 5.15.2.3 delete\_schedule\_by\_id()

```
esp_err_t delete_schedule_by_id (
    uint8_t channel,
    uint8_t ID )
```

Delete a schedule by ID.

#### Parameters

<i>channel</i>	The channel the schedule is located on.
<i>ID</i>	The ID of the schedule to delete.

#### Returns

esp\_err\_t Returns ESP\_OK on successful deletion. ESP\_ERR\_INVALID\_ARG if channel number is invalid. ESP\_ERR\_NOT\_FOUND if schedule not found

### 5.15.2.4 delete\_schedule\_by\_name()

```
esp_err_t delete_schedule_by_name (
    uint8_t channel,
    char * name )
```

Delete a schedule by name.

#### Parameters

<i>channel</i>	The channel the schedule is located on.
<i>name</i>	The name of the schedule to delete.

#### Returns

esp\_err\_t Returns ESP\_OK on successful deletion. ESP\_ERR\_INVALID\_ARG if channel number is invalid. ESP\_ERR\_NOT\_FOUND if schedule not found

### 5.15.2.5 disable\_all\_schedules()

```
esp_err_t disable_all_schedules (
    void )
```

Disables all schedules on all channels. The schedules remain located in memory but no longer run or perform updates.

#### Returns

esp\_err\_t ESP\_OK on successful disable.

### 5.15.2.6 disable\_schedule\_by\_id()

```
esp_err_t disable_schedule_by_id (
    uint8_t channel,
    uint8_t ID )
```

Disable a schedule by ID. The schedule remains in the linked list but does not run or perform updates.

#### Parameters

<i>channel</i>	The channel the schedule is located on.
<i>ID</i>	The name of the schedule to disable.

#### Returns

esp\_err\_t ESP\_OK on successful disable. ESP\_ERR\_INVALID\_ARG if channel number is invalid. ESP\_ERR\_NOT\_FOUND if schedule not found

### 5.15.2.7 disable\_schedule\_by\_name()

```
esp_err_t disable_schedule_by_name (
    uint8_t channel,
    char * name )
```

Disable a schedule by name. The schedule remains in the linked list but does not run or perform updates.

#### Parameters

<i>channel</i>	The channel the schedule is located on.
<i>name</i>	The name of the schedule to disable.

#### Returns

esp\_err\_t ESP\_OK on successful disable. ESP\_ERR\_INVALID\_ARG if channel number is invalid. ESP\_ERR\_NOT\_FOUND if schedule not found

### 5.15.2.8 enable\_all\_schedules()

```
esp_err_t enable_all_schedules (
    void )
```

Enable all schedules on all channels. If the schedule was disabled, the schedule now runs and performs updates.

#### Returns

esp\_err\_t ESP\_OK on successful disable.

### 5.15.2.9 enable\_schedule\_by\_id()

```
esp_err_t enable_schedule_by_id (
    uint8_t channel,
    uint8_t ID )
```

Enable a schedule by ID. If the schedule was disabled, the schedule now runs and performs updates.

#### Parameters

<i>channel</i>	The channel the schedule is located on.
<i>ID</i>	The ID of the schedule to enable.

#### Returns

esp\_err\_t ESP\_OK on successful enable. ESP\_ERR\_INVALID\_ARG if channel number is invalid. ESP\_ERR\_NOT\_FOUND if schedule not found

### 5.15.2.10 enable\_schedule\_by\_name()

```
esp_err_t enable_schedule_by_name (
    uint8_t channel,
    char * name )
```

Enable a schedule by name. If the schedule was disabled, the schedule now runs and performs updates.

#### Parameters

<i>channel</i>	The channel the schedule is located on.
<i>name</i>	The name of the schedule to enable.

#### Returns

esp\_err\_t ESP\_OK on successful enable. ESP\_ERR\_INVALID\_ARG if channel number is invalid. ESP\_ERR\_NOT\_FOUND if schedule not found

### 5.15.2.11 get\_schedule()

```
esp_err_t get_schedule (
    uint8_t channel,
    char * name,
    schedule_object * out )
```

Get the schedule object specified.



**Parameters**

<i>channel</i>	The channel the schedule is located on.
<i>name</i>	The name of the schedule to find.
<i>out</i>	The schedule returned. NULL if failure occurred.

**Returns**

esp\_err\_t ESP\_OK if successful. ESP\_ERR\_INVALID\_ARG if channel is invalid. ESP\_ERR\_NOT\_FOUND if the schedule could not be found.

**5.15.2.12 get\_schedule\_names()**

```
esp_err_t get_schedule_names (
    uint8_t channel,
    char *& out )
```

Get a json string of all schedules and their enabled status for a given channel.

**Parameters**

<i>channel</i>	The channel to get the schedule names and status from.
<i>out</i>	The json string that is returned. Format is '{ "Name1":0, "Name2":1, "Name3":0 }'

**Returns**

esp\_err\_t ESP\_OK if successful. ESP\_ERR\_INVALID\_ARG if channel is invalid.

**5.15.2.13 init\_schedule()**

```
void init_schedule (
    void )
```

Init the scheduler code. Starts the background task if not already started. If this is not called the schedules will not run.

**5.15.3 Variable Documentation****5.15.3.1 schedules**

```
List* schedules[NUM\_CHANNELS]
```

The linked list associated with each channel. Can be used directly instead of scheduler functions if so desired.

## 5.16 include/wifi.h File Reference

```
#include <string.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/event_groups.h"
#include "esp_system.h"
#include "esp_wifi.h"
#include "esp_event_loop.h"
#include "esp_log.h"
```

### Macros

- #define [MIN](#)(x, y) ((x)<(y)?(x):(y))

### Functions

- void [wifi\\_init\\_softap](#) ()
- void [wifi\\_init\\_sta](#) ()

#### 5.16.1 Detailed Description

Description: Contains function declarations to connect to a WiFi networking using SSID defined in [wifi.cpp](#)

#### Author

: Lead: Andy Yeung, Secondary: Jesse Cannon

#### Date

last modified: 4/26/2020

#### 5.16.2 Macro Definition Documentation

##### 5.16.2.1 MIN

```
#define MIN(  
    x,  
    y )  ((x)<(y)?(x):(y))
```

#### 5.16.3 Function Documentation

### 5.16.3.1 wifi\_init\_softap()

```
void wifi_init_softap ( )
```

### 5.16.3.2 wifi\_init\_sta()

```
void wifi_init_sta ( )
```

## 5.17 README.md File Reference

## 5.18 src/bleSL.cpp File Reference

GATT server demo code used as base from esp-idf. Most modifications made in Init\_Bluetooth function and gatts\_profile\_event\_handler callback.

```
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/event_groups.h"
#include "esp_system.h"
#include "esp_log.h"
#include "nvs_flash.h"
#include "esp_bt.h"
#include "esp_gap_ble_api.h"
#include "esp_gatts_api.h"
#include "esp_bt_main.h"
#include "gatts_table_creat_demo.h"
#include "esp_gatt_common_api.h"
#include "bleSL.h"
#include "decode_bluetooth.h"
```

### Classes

- struct [prepare\\_type\\_env\\_t](#)
- struct [gatts\\_profile\\_inst](#)

### Macros

- #define [GATTS\\_TABLE\\_TAG](#) "GATTS\_TABLE\_DEMO"
- #define [PROFILE\\_NUM](#) 1
- #define [PROFILE\\_APP\\_IDX](#) 0
- #define [ESP\\_APP\\_ID](#) 0x55
- #define [SAMPLE\\_DEVICE\\_NAME](#) "ESP\_GATTS\_DEMO"
- #define [SVC\\_INST\\_ID](#) 0
- #define [GATTS\\_DEMO\\_CHAR\\_VAL\\_LEN\\_MAX](#) 500
- #define [PREPARE\\_BUF\\_MAX\\_SIZE](#) 1024
- #define [CHAR\\_DECLARATION\\_SIZE](#) (sizeof(uint8\_t))
- #define [ADV\\_CONFIG\\_FLAG](#) (1 << 0)
- #define [SCAN\\_RSP\\_CONFIG\\_FLAG](#) (1 << 1)
- #define [CONFIG\\_SET\\_RAW\\_ADV\\_DATA](#)

## Functions

- void [example\\_prepare\\_write\\_event\\_env](#) (esp\_gatt\_if\_t gatts\_if, [prepare\\_type\\_env\\_t](#) \*prepare\_write\_env, esp\_ble\_gatts\_cb\_param\_t \*param)
- void [example\\_exec\\_write\\_event\\_env](#) ([prepare\\_type\\_env\\_t](#) \*prepare\_write\_env, esp\_ble\_gatts\_cb\_param\_t \*param)
- void [Init\\_Bluetooth](#) (void)

*Initializes GATT Server.*

## Variables

- uint16\_t [heart\\_rate\\_handle\\_table](#) [[HRS\\_IDX\\_NB](#)]

### 5.18.1 Detailed Description

GATT server demo code used as base from esp-idf. Most modifications made in [Init\\_Bluetooth](#) function and [gatts\\_profile\\_event\\_handler](#) callback.

#### Author

Hunaid Puri ( [hunaid14@gmail.com](mailto:hunaid14@gmail.com) )

#### Version

0.1

#### Date

2020-04-26

#### Copyright

Copyright (c) 2020

### 5.18.2 Macro Definition Documentation

#### 5.18.2.1 ADV\_CONFIG\_FLAG

```
#define ADV_CONFIG_FLAG (1 << 0)
```

#### 5.18.2.2 CHAR\_DECLARATION\_SIZE

```
#define CHAR_DECLARATION_SIZE (sizeof(uint8_t))
```

### 5.18.2.3 CONFIG\_SET\_RAW\_ADV\_DATA

```
#define CONFIG_SET_RAW_ADV_DATA
```

### 5.18.2.4 ESP\_APP\_ID

```
#define ESP_APP_ID 0x55
```

### 5.18.2.5 GATTS\_DEMO\_CHAR\_VAL\_LEN\_MAX

```
#define GATTS_DEMO_CHAR_VAL_LEN_MAX 500
```

### 5.18.2.6 GATTS\_TABLE\_TAG

```
#define GATTS_TABLE_TAG "GATTS_TABLE_DEMO"
```

### 5.18.2.7 PREPARE\_BUF\_MAX\_SIZE

```
#define PREPARE_BUF_MAX_SIZE 1024
```

### 5.18.2.8 PROFILE\_APP\_IDX

```
#define PROFILE_APP_IDX 0
```

### 5.18.2.9 PROFILE\_NUM

```
#define PROFILE_NUM 1
```

### 5.18.2.10 SAMPLE\_DEVICE\_NAME

```
#define SAMPLE_DEVICE_NAME "ESP_GATTS_DEMO"
```

### 5.18.2.11 SCAN\_RSP\_CONFIG\_FLAG

```
#define SCAN_RSP_CONFIG_FLAG (1 << 1)
```

### 5.18.2.12 SVC\_INST\_ID

```
#define SVC_INST_ID 0
```

## 5.18.3 Function Documentation

### 5.18.3.1 example\_exec\_write\_event\_env()

```
void example_exec_write_event_env (
    prepare_type_env_t * prepare_write_env,
    esp_ble_gatts_cb_param_t * param )
```

### 5.18.3.2 example\_prepare\_write\_event\_env()

```
void example_prepare_write_event_env (
    esp_gatt_if_t gatts_if,
    prepare_type_env_t * prepare_write_env,
    esp_ble_gatts_cb_param_t * param )
```

### 5.18.3.3 Init\_Bluetooth()

```
void Init_Bluetooth (
    void )
```

Initializes GATT Server.

## 5.18.4 Variable Documentation

### 5.18.4.1 heart\_rate\_handle\_table

```
uint16_t heart_rate_handle_table[HRS_IDX_NB]
```

## 5.19 src/dawndusk.cpp File Reference

```
#include "dawndusk.h"
```

### Functions

- double [degToRad](#) (double deg)  
*Convert degrees to radians.*
- int [sign](#) (double x)  
*Checks sign of given value. If longitude value is negative, an offset is taken into account for dawn/dusk calculations.*
- double [dawnCalc](#) (int day, int month, int year, double latitude, double longitude)  
*Calculation of approximate dawn time based on given day and location.*
- double [duskCalc](#) (int day, int month, int year, double latitude, double longitude)  
*Calculation of approximate dusk time based on given day and location.*

### 5.19.1 Detailed Description

Description: Contains function definitions to approximate dawn and dusk times based on latitude and longitude. This article was first published on Quantitative Ecology, and adapted from R-bloggers.

#### Author

: Shipra Vaidya

#### Date

last modified: 4/26/2020

### 5.19.2 Function Documentation

#### 5.19.2.1 dawnCalc()

```
double dawnCalc (
    int day,
    int month,
    int year,
    double latitude,
    double longitude )
```

Calculation of approximate dawn time based on given day and location.

#### Parameters

<i>day</i>	Numerical value, day of month.
<i>month</i>	Numerical value, month in year.
<i>year</i>	Numerical value, year.
<i>latitude</i>	Degree value of latitude from user's location.
<i>longitude</i>	Degree value of longitude from user's location.

**Returns**

double Returns approximate time of dawn on given date at given location.

**5.19.2.2 degToRad()**

```
double degToRad (  
    double deg )
```

Convert degrees to radians.

**Parameters**

<i>deg</i>	Latitude or longitude value in degrees.
------------	---

**Returns**

double Returns converted value in radians.

**5.19.2.3 duskCalc()**

```
double duskCalc (  
    int day,  
    int month,  
    int year,  
    double latitude,  
    double longitude )
```

Calculation of approximate dusk time based on given day and location.

**Parameters**

<i>day</i>	Numerical value, day of month.
<i>month</i>	Numerical value, month in year.
<i>year</i>	Numerical value, year.
<i>latitude</i>	Degree value of latitude from user's location.
<i>longitude</i>	Degree value of longitude from user's location.

**Returns**

double Returns approximate time of dusk on given date at given location.



### 5.19.2.4 sign()

```
int sign (
    double x )
```

Checks sign of given value. If longitude value is negative, an offset is taken into account for dawn/duck calculations.

#### Parameters

x	Value of longitude.
---	---------------------

#### Returns

int Returns 1 if positive, -1 if negative, to be multiplied in offset calculation.

## 5.20 src/decode\_bluetooth.cpp File Reference

This file contains functions used to decode/set byte packets from GATT server.

```
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/event_groups.h"
#include "esp_system.h"
#include "decode_bluetooth.h"
#include "scheduler.h"
#include "schedule_object.h"
#include "espsntp.h"
#include "esp_gatts_api.h"
```

### Functions

- `esp_err_t start_schedule_read` (uint8\_t \*packet, uint16\_t handle)  
*Looks for "Begin Read" from the packet recieved. Saves channel number found in packet in saved\_channel\_num. Calls set\_schedule read, passing handle parameter.*
- `esp_err_t set_schedule_read` (uint16\_t handle)  
*Sets GATT characteristic with the respective handle to the value of a schedule. First sets the characteristic to the name of the first schedule. On the next read it will set the characteristic to the schedule information. This will repeat until there are no more schedules in the requested channel (saved\_channel\_num).*
- `esp_err_t decode_ble_time` (uint8\_t \*packet)  
*Converts byte packet into UNIX time value and calls set\_time function.*
- `esp_err_t decode_ble_direct` (uint8\_t \*packet)  
*Converts byte packet into channel, brightness, and color information. If RGB channel is written, calls set\_color function. If normal LED channel is written, calls channel\_on function.*
- `esp_err_t decode_ble_delete` (uint8\_t \*packet)  
*Converts byte packet into schedule name and channel. The schedule on the corresponding channel is then deleted using delete\_schedule\_by\_name function.*
- `esp_err_t decode_ble_schedule_name` (uint8\_t \*packet, uint16\_t length)  
*Converts byte packet into string and saves value in saved\_name.*
- `esp_err_t decode_ble_schedule` (uint8\_t \*packet)  
*Converts byte packet in order to create schedule object. Recieves name from saved\_name.*
- `uint32_t get_Int32` (uint8\_t \*begin)  
*Takes byte array and converts into a uint32 using the first 4 indices. Assumes Big-endian.*

## Variables

- uint8\_t `schedule_value` [20]
- uint8\_t `saved_channel_num`
- char `schedule_name` [20]
- List \* `schedules` [NUM\_CHANNELS]

*The linked list associated with each channel. Can be used directly instead of scheduler functions if so desired.*

- List \* `it`
- char `saved_name` [21]
- uint16\_t `saved_len`
- uint8\_t `state` =4
- uint8\_t `done_string` [10] ="Done Read"

### 5.20.1 Detailed Description

This file contains functions used to decode/set byte packets from GATT server.

#### Author

Hunaid Puri ( [hunaid14@gmail.com](mailto:hunaid14@gmail.com) )

#### Version

0.1

#### Date

2020-04-26

#### Copyright

Copyright (c) 2020

### 5.20.2 Function Documentation

#### 5.20.2.1 `decode_ble_delete()`

```
esp_err_t decode_ble_delete (
    uint8_t * packet )
```

Converts byte packet into schedule name and channel. The schedule on the corresponding channel is then deleted using `delete_schedule_by_name` function.

#### Parameters

<i>packet</i>	recieved byte packet with schedule name to delete
---------------	---

**Returns**

esp\_err\_t

**5.20.2.2 decode\_ble\_direct()**

```
esp_err_t decode_ble_direct (
    uint8_t * packet )
```

Converts byte packet into channel, brightness, and color information. If RGB channel is written, calls set\_color function. If normal LED channel is written, calls channel\_on function.

**Parameters**

<i>packet</i>	recieved byte packet with led control information
---------------	---

**Returns**

esp\_err\_t

**5.20.2.3 decode\_ble\_schedule()**

```
esp_err_t decode_ble_schedule (
    uint8_t * packet )
```

Converts byte packet in order to create schedule object. Recieves name from saved\_name.

**Parameters**

<i>packet</i>	recieved byte packet with schedule information
---------------	--

**Returns**

esp\_err\_t

**5.20.2.4 decode\_ble\_schedule\_name()**

```
esp_err_t decode_ble_schedule_name (
    uint8_t * packet,
    uint16_t length )
```

Converts byte packet into string and saves value in saved\_name.

**Parameters**

<i>packet</i>	recieved byte packet with scheudle name
<i>length</i>	length of recieved packet

**Returns**

esp\_err\_t

**5.20.2.5 decode\_ble\_time()**

```
esp_err_t decode_ble_time (  
    uint8_t * packet )
```

Converts byte packet into UNIX time value and calls set\_time function.

**Parameters**

<i>packet</i>	recieved byte packet with time information
---------------	--

**Returns**

esp\_err\_t

**5.20.2.6 get\_Int32()**

```
uint32_t get_Int32 (  
    uint8_t * begin )
```

Takes byte array and converts into a uint32 using the first 4 indices. Assumes Big-endian.

**Parameters**

<i>begin</i>	byte array
--------------	------------

**Returns**

uint32\_t

### 5.20.2.7 set\_schedule\_read()

```
esp_err_t set_schedule_read (
    uint16_t handle )
```

Sets GATT characteristic with the respective handle to the value of a schedule. First sets the characteristic to the name of the first schedule. On the next read it will set the characteristic to the schedule information. This will repeat until there are no more schedules in the requested channel (saved\_channel\_num).

#### Parameters

<i>handle</i>	handle for the characteristic that will be read
---------------	---

#### Returns

esp\_err\_t

### 5.20.2.8 start\_schedule\_read()

```
esp_err_t start_schedule_read (
    uint8_t * packet,
    uint16_t handle )
```

Looks for "Begin Read" from the packet recieved. Saves channel number found in packet in saved\_channel\_num. Calls set\_schedule read, passing handle parameter.

#### Parameters

<i>packet</i>	recieved byte packet
<i>handle</i>	handle for the characteristic that will be read

#### Returns

esp\_err\_t

## 5.20.3 Variable Documentation

### 5.20.3.1 done\_string

```
uint8_t done_string[10] ="Done Read"
```

### 5.20.3.2 it

```
List* it
```

### 5.20.3.3 saved\_channel\_num

```
uint8_t saved_channel_num
```

### 5.20.3.4 saved\_len

```
uint16_t saved_len
```

### 5.20.3.5 saved\_name

```
char saved_name[21]
```

### 5.20.3.6 schedule\_name

```
char schedule_name[20]
```

### 5.20.3.7 schedule\_value

```
uint8_t schedule_value[20]
```

### 5.20.3.8 schedules

```
List* schedules[NUM_CHANNELS]
```

The linked list associated with each channel. Can be used directly instead of scheduler functions if so desired.

### 5.20.3.9 state

```
uint8_t state =4
```

## 5.21 src/espntp.cpp File Reference

```
#include "espntp.h"
```

### Functions

- void [set\\_time](#) (uint32\_t time)
- void [obtain\\_time](#) (void)

### Variables

- const int [CONNECTED\\_BIT](#) = BIT0

### 5.21.1 Detailed Description

Description: Contains function definitions to start the process of obtaining time from an SNTP server. NOTE: not tested with latest version of project. Was initially developed very early in project and abandoned for other priorities.

#### Author

: Jesse Cannon (based on [https://github.com/espressif/esp-idf/blob/5aa21584cfb4cfe4f2c7f623\\_example\\_main.c](https://github.com/espressif/esp-idf/blob/5aa21584cfb4cfe4f2c7f623_example_main.c))

#### Date

last modified: 4/26/2020

### 5.21.2 Function Documentation

#### 5.21.2.1 obtain\_time()

```
void obtain_time (  
    void )
```

### 5.21.2.2 set\_time()

```
void set_time (
    uint32_t time )
```

## 5.21.3 Variable Documentation

### 5.21.3.1 CONNECTED\_BIT

```
const int CONNECTED_BIT = BIT0
```

## 5.22 src/http.cpp File Reference

File for hosting http server specific to LED controller/scheduler.

```
#include <string.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/event_groups.h"
#include "esp_system.h"
#include "esp_wifi.h"
#include "esp_event_loop.h"
#include "esp_log.h"
#include "esp_http_server.h"
#include "http.h"
#include "scheduler.h"
#include "espsntp.h"
```

## Macros

- #define [DELIMITER](#) (";:")

## Functions

- [esp\\_err\\_t init\\_http](#) (httpd\_handle\_t server)  
*Initializes a http server with other handlers defined the file.*
- [esp\\_err\\_t homepage\\_handler](#) (httpd\_req\_t \*req)  
*Handler to retrieve home/direct control page. Accessed when server receives "/" or "/index.html".*
- [esp\\_err\\_t styles\\_handler](#) (httpd\_req\_t \*req)  
*Handler to retrieve styles CSS page. Accessed when server receives "/styles.css".*
- [esp\\_err\\_t schedules\\_handler](#) (httpd\_req\_t \*req)  
*Handler to retrieve schedules page. Accessed when server receives "/schedules.html".*
- [esp\\_err\\_t scripts\\_handler](#) (httpd\_req\_t \*req)  
*Handler to retrieve scripts page. Accessed when server receives "/schedules.js".*



- `esp_err_t schedule_post_handler (httpd_req_t *req)`  
*Handler to create new schedule. Accessed when server receives or "/post\_sch".*
- `void schTokenProcess (char *str)`  
*Function to process ";;"-delimited string. Parses out scheduling data and for schedule post handler.*
- `esp_err_t favicon_ico_get_handler (httpd_req_t *req)`  
*Handler to retrieve favicon. Accessed when server receives "/favicon.ico" - DOES NOT WORK.*
- `esp_err_t time_post_handler (httpd_req_t *req)`  
*Handler to post time to esp32. Accessed when server receives "/time".*
- `esp_err_t direct_control_post_handler (httpd_req_t *req)`  
*Handler for direct control. Access when server receives "/direct\_control".*
- `esp_err_t sch_data_post_handler (httpd_req_t *req)`  
*Handler to retrieve schedule data Access when server receives "/sch\_data".*

### 5.22.1 Detailed Description

File for hosting http server specific to LED controller/scheduler.

#### Author

Andy Yang ( [andyyeung123@gmail.com](mailto:andyyeung123@gmail.com))

#### Version

0.1

#### Date

2020-04-27

#### Copyright

Copyright (c) 2020

#### Description:

#### Author

: Andy Yeung

#### Date

last modified: 4/26/2020

### 5.22.2 Macro Definition Documentation

### 5.22.2.1 DELIMITER

```
#define DELIMITER ("';")
```

## 5.22.3 Function Documentation

### 5.22.3.1 direct\_control\_post\_handler()

```
esp_err_t direct_control_post_handler (  
    httpd_req_t * req )
```

Handler for direct control. Access when server receives "/direct\_control".

#### Parameters

<i>req</i>	An incoming HTTP request.
------------	---------------------------

#### Returns

`esp_err_t` Returns `ESP_OK` on successful deletion. `ESP_ERR_INVALID_ARG` if channel number is invalid. `ESP_ERR_NOT_FOUND` if schedule not found

### 5.22.3.2 favicon\_ico\_get\_handler()

```
esp_err_t favicon_ico_get_handler (  
    httpd_req_t * req )
```

Handler to retrieve favicon. Accessed when server receives "/favicon.ico" - DOES NOT WORK.

#### Parameters

<i>req</i>	An incoming HTTP request.
------------	---------------------------

#### Returns

`esp_err_t` Returns `ESP_OK` on successful deletion. `ESP_ERR_INVALID_ARG` if channel number is invalid. `ESP_ERR_NOT_FOUND` if schedule not found

### 5.22.3.3 homepage\_handler()

```
esp_err_t homepage_handler (  
    httpd_req_t * req )
```

Handler to retrieve home/direct control page. Accessed when server receives "/" or "/index.html".

**Parameters**

<i>req</i>	An incoming HTTP request.
------------	---------------------------

**Returns**

`esp_err_t` Returns `ESP_OK` on successful deletion. `ESP_ERR_INVALID_ARG` if channel number is invalid.  
`ESP_ERR_NOT_FOUND` if schedule not found

**5.22.3.4 `init_http()`**

```
esp_err_t init_http (  
    httpd_handle_t server )
```

Initializes a http server with other handlers defined the file.

**Parameters**

<i>server</i>	A handle for the server.
---------------	--------------------------

**Returns**

`esp_err_t` Returns `ESP_OK` on successful deletion. `ESP_ERR_INVALID_ARG` if channel number is invalid.  
`ESP_ERR_NOT_FOUND` if schedule not found

**5.22.3.5 `sch_data_post_handler()`**

```
esp_err_t sch_data_post_handler (  
    httpd_req_t * req )
```

Handler to retrieve schedule data Access when server receives `"/sch_data"`.

**Parameters**

<i>req</i>	An incoming HTTP request.
------------	---------------------------

**Returns**

`esp_err_t` Returns `ESP_OK` on successful deletion. `ESP_ERR_INVALID_ARG` if channel number is invalid.  
`ESP_ERR_NOT_FOUND` if schedule not found

### 5.22.3.6 schedule\_post\_handler()

```
esp_err_t schedule_post_handler (
    httpd_req_t * req )
```

Handler to create new schedule. Accessed when server receives or "/post\_sch".

#### Parameters

<i>req</i>	An incoming HTTP request.
------------	---------------------------

#### Returns

`esp_err_t` Returns `ESP_OK` on successful deletion. `ESP_ERR_INVALID_ARG` if channel number is invalid. `ESP_ERR_NOT_FOUND` if schedule not found

### 5.22.3.7 schedules\_handler()

```
esp_err_t schedules_handler (
    httpd_req_t * req )
```

Handler to retrieve schedules page. Accessed when server receives "/schedules.html".

#### Parameters

<i>req</i>	An incoming HTTP request.
------------	---------------------------

#### Returns

`esp_err_t` Returns `ESP_OK` on successful deletion. `ESP_ERR_INVALID_ARG` if channel number is invalid. `ESP_ERR_NOT_FOUND` if schedule not found

### 5.22.3.8 schTokenProcess()

```
void schTokenProcess (
    char * str )
```

Function to process ";;"-delimited string. Parses out scheduling data and for schedule post handler.

#### Parameters

<i>str</i>	Received string
------------	-----------------

### 5.22.3.9 scripts\_handler()

```
esp_err_t scripts_handler (  
    httpd_req_t * req )
```

Handler to retrieve scripts page. Accessed when server receives "/schedules.js".

#### Parameters

<i>req</i>	An incoming HTTP request.
------------	---------------------------

#### Returns

`esp_err_t` Returns `ESP_OK` on successful deletion. `ESP_ERR_INVALID_ARG` if channel number is invalid. `ESP_ERR_NOT_FOUND` if schedule not found

### 5.22.3.10 styles\_handler()

```
esp_err_t styles_handler (  
    httpd_req_t * req )
```

Handler to retrieve styles CSS page. Accessed when server receives "/styles.css".

#### Parameters

<i>req</i>	An incoming HTTP request.
------------	---------------------------

#### Returns

`esp_err_t` Returns `ESP_OK` on successful deletion. `ESP_ERR_INVALID_ARG` if channel number is invalid. `ESP_ERR_NOT_FOUND` if schedule not found

### 5.22.3.11 time\_post\_handler()

```
esp_err_t time_post_handler (  
    httpd_req_t * req )
```

Handler to post time to esp32. Accessed when server receives "/time".

#### Parameters

<i>req</i>	An incoming HTTP request.
------------	---------------------------

### Returns

esp\_err\_t Returns ESP\_OK on successful deletion. ESP\_ERR\_INVALID\_ARG if channel number is invalid. ESP\_ERR\_NOT\_FOUND if schedule not found

## 5.23 src/led.cpp File Reference

```
#include "led.h"
```

### Classes

- struct [channel](#)

### Functions

- esp\_err\_t [shutdown\\_outputs](#) (void)  
*Shuts down all channels and prevents them from turning on again. Enter a permanent shutdown state until clear\_↔ shutdown is called or device is reset.*
- esp\_err\_t [clear\\_shutdown](#) (void)  
*Clears a shutdown state, allowing outputs to be controlled again.*
- void [init\\_channels](#) (void)  
*Initialize channels to set all timer and channel configurations.*
- void [channel\\_on](#) (uint8\_t index, uint8\_t brightness)  
*Sets brightness of individual channel to turn that output on.*
- void [channel\\_off](#) (uint8\_t index)  
*Sets brightness of individual channel to 0 to turn off channel output.*
- void [set\\_color](#) (uint8\_t index, uint16\_t r, uint16\_t g, uint16\_t b, uint8\_t brightness)  
*Controls RGB channel by changing R, G, and B based off chosen brightness.*

### Variables

- uint8\_t [shutdown\\_status](#) = 0

#### 5.23.1 Detailed Description

Description: Contains function definitions to setup and manipulate GPIO outputs as PWM outputs for controlling LED strips

#### Author

: Primary: Shipra Vaidya, Secondary: Jesse Cannon (bugfixes and shutdown feature)

#### Date

last modified: 4/26/2020

## 5.23.2 Function Documentation

### 5.23.2.1 `channel_off()`

```
void channel_off (
    uint8_t index )
```

Sets brightness of individual channel to 0 to turn off channel output.



**Parameters**

<i>index</i>	Value between 0 and 5 corresponding to channel being controlled.
--------------	--

**5.23.2.2 channel\_on()**

```
void channel_on (
    uint8_t index,
    uint8_t brightness )
```

Sets brightness of individual channel to turn that output on.

**Parameters**

<i>index</i>	Value between 0 and 5 corresponding to channel being controlled.
<i>brightness</i>	Value of brightness chosen per channel.

**5.23.2.3 clear\_shutdown()**

```
esp_err_t clear_shutdown (
    void )
```

Clears a shutdown state, allowing outputs to be controlled again.

**Returns**

esp\_err\_t ESP\_OK on success, ESP\_FAIL if not in shutdown state.

**5.23.2.4 init\_channels()**

```
void init_channels (
    void )
```

Initialize channels to set all timer and channel configurations.

**5.23.2.5 set\_color()**

```
void set_color (
    uint8_t index,
    uint16_t r,
    uint16_t g,
    uint16_t b,
    uint8_t brightness )
```

Controls RGB channel by changing R, G, and B based off chosen brightness.

**Parameters**

<i>index</i>	Value between 0 and 1 corresponding to two supported RGB channel outputs respectively.
<i>r</i>	R value of chosen color.
<i>g</i>	G value of chosen color.
<i>b</i>	B value of chosen color.
<i>brightness</i>	Value of brightness chosen for RGB channel.

**5.23.2.6 shutdown\_outputs()**

```
esp_err_t shutdown_outputs (
    void )
```

Shuts down all channels and prevents them from turning on again. Enter a permanent shutdown state until `clear_shutdown` is called or device is reset.

**Returns**

`esp_err_t` ESP\_OK on success, ESP\_FAIL if already in shutdown state.

**5.23.3 Variable Documentation****5.23.3.1 shutdown\_status**

```
uint8_t shutdown_status = 0
```

**5.24 src/main.cpp File Reference**

```
#include <stdio.h>
#include <string.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/event_groups.h"
#include "esp_event_loop.h"
#include "esp_http_server.h"
#include "esp_log.h"
#include "esp_spiffs.h"
#include "esp_system.h"
#include "esp_wifi.h"
#include "nvs.h"
#include "nvs_flash.h"
#include "lwip/err.h"
#include "lwip/sys.h"
#include "espntp.h"
```

```
#include "http.h"
#include "led.h"
#include "scheduler.h"
#include "measurement.h"
#include "memory.h"
#include "wifi.h"
#include "bleSL.h"
#include "decode_bluetooth.h"
```

## Functions

- void [app\\_main](#) ()

### 5.24.1 Detailed Description

Description: Contains code to initialize and start 'IoT LED Controller Device' subsystems

#### Author

: Jesse Cannon, Hunaid Puri, Shipra Vaidya, Andy Yeung

#### Date

last modified: 4/26/2020

### 5.24.2 Function Documentation

#### 5.24.2.1 app\_main()

```
void app_main ( )
```

## 5.25 src/measurement.cpp File Reference

```
#include "driver/gpio.h"
#include "measurement.h"
```

## Macros

- #define [TAG](#) "OC"

## Functions

- `uint8_t isCurrentFault ()`  
*Determines if the device is in shutdown from an overcurrent condition.*
- `uint8_t isVoltageFault ()`  
*Determines if the device is in shutdown from an overvoltage condition.*
- `esp_err_t clearFaults ()`  
*Clears all faults from the system and allows outputs to be controlled again.*
- `esp_err_t set_current_level (double ampLimit)`  
*Set the current level that will trigger a fault.*
- `esp_err_t set_voltage_level (double voltLimit)`  
*Set the voltage level that will trigger a fault.*
- `esp_err_t init_oc (void)`  
*Init the overcurrent and overvoltage detection systems. The previously used values will be set as the level, with defaults of 24 Volts and 2 Amps triggering faults.*

## Variables

- `uint8_t currentFault = 0`
- `uint8_t voltageFault = 0`

### 5.25.1 Detailed Description

Description: Contains function definitions to set up and detect overvoltage and overcurrent conditions. NOTE: This is basically just out outline. Needs significant testing with actual hardware before development can finish.

#### Author

: Jesse Cannon

#### Date

last modified: 4/26/2020

### 5.25.2 Macro Definition Documentation

#### 5.25.2.1 TAG

```
#define TAG "OC"
```

### 5.25.3 Function Documentation

### 5.25.3.1 clearFaults()

```
esp_err_t clearFaults ( )
```

Clears all faults from the system and allows outputs to be controlled again.

#### Returns

esp\_err\_t ESP\_OK if successfully re-enables outputs. ESP\_FAIL if error occurs.

### 5.25.3.2 init\_oc()

```
esp_err_t init_oc (
    void )
```

Initiates the overcurrent and overvoltage detection systems. The previously used values will be set as the level, with defaults of 24 Volts and 2 Amps triggering faults.

#### Returns

esp\_err\_t

### 5.25.3.3 isCurrentFault()

```
uint8_t isCurrentFault ( )
```

Determines if the device is in shutdown from an overcurrent condition.

#### Returns

uint8\_t A value of 1 means that there is currently a fault. A value of 0 means there is no fault

### 5.25.3.4 isVoltageFault()

```
uint8_t isVoltageFault ( )
```

Determines if the device is in shutdown from an overvoltage condition.

#### Returns

uint8\_t A value of 1 means that there is currently a fault. A value of 0 means there is no fault

### 5.25.3.5 set\_current\_level()

```
esp_err_t set_current_level (
    double ampLimit )
```

Set the current level that will trigger a fault.

**Parameters**

<i>ampLimit</i>	The current level to be used to determine faults, such as 1.1 Amps.
-----------------	---

**Returns**

esp\_err\_t ESP\_OK if the new level was successfully set, else if error occurred.

**5.25.3.6 set\_voltage\_level()**

```
esp_err_t set_voltage_level (
    double voltLimit )
```

Set the voltage level that will trigger a fault.

**Parameters**

<i>voltLimit</i>	The voltage level to be used to determine faults, such as 12.5 Volts.
------------------	---

**Returns**

esp\_err\_t ESP\_OK if the new level was successfully set, else if error occurred.

**5.25.4 Variable Documentation****5.25.4.1 currentFault**

```
uint8_t currentFault = 0
```

**5.25.4.2 voltageFault**

```
uint8_t voltageFault = 0
```

**5.26 src/memory.cpp File Reference**

```
#include "memory.h"
```

## Functions

- `esp_err_t read_settings_to_buffer` (void)
- `esp_err_t init_spiffs` (void)
- `esp_err_t init_memory` (void)  
*Starts the memory system. Call this before calling any other memory related functions.*
- `esp_err_t recall_schedules` (void)  
*Recall schedules from persistent memory. All schedules recalled will now be located in the tespective channel's schedule list.*
- `esp_err_t store_schedules` (void)  
*Stores all currently running schedules in persistent memory.*
- `esp_err_t clear_schedule_data` (void)  
*Clears all schedules from persistent memory. Does not delete the schedules from the list.*
- `esp_err_t store_setting_string` (const char \*name, char \*setting)  
*Persistently store a string setting.*
- `esp_err_t store_setting_int` (const char \*name, int setting)  
*Persistently store an integer setting.*
- `esp_err_t store_setting_byte` (const char \*name, uint8\_t setting)  
*Persistently store a byte setting.*
- `esp_err_t store_setting_double` (const char \*name, double setting)  
*Persistently store a double setting.*
- `esp_err_t get_setting_string` (const char \*name, char \*setting)  
*Recall a persistent string setting.*
- `esp_err_t get_setting_int` (const char \*name, int \*setting)  
*Recall a persistent integer setting.*
- `esp_err_t get_setting_byte` (const char \*name, uint8\_t \*setting)  
*Recall a byte string setting. Uint8\_t is equivalent to unsigned char.*
- `esp_err_t get_setting_double` (const char \*name, double \*setting)  
*Recall a persistent double setting. Exact storage depends on ArduinoJson configuration, but typically a maximum of 9 decimal places with rounding. No trailing zeros are included.*
- `esp_err_t clear_setting_data` (void)  
*Clear settings from persistent memory and RAM.*

## Variables

- bool `bSPIFFS` = false
- bool `readNeeded` = true
- char \* `settingsString`

### 5.26.1 Detailed Description

Description: Contains function definitions and settings used by memory subsystem. Provides interface to store values in persistent memory in the JSON format using ArduinoJson-v6.14.1

Author

: Jesse Cannon

Date

last modified: 4/26/2020

## 5.26.2 Function Documentation

### 5.26.2.1 clear\_schedule\_data()

```
esp_err_t clear_schedule_data (  
    void )
```

Clears all schedules from persistent memory. Does not delete the schedules from the list.

#### Returns

esp\_err\_t Returns ESP\_OK on success. ESP\_Fail or related if an error occurred.

### 5.26.2.2 clear\_setting\_data()

```
esp_err_t clear_setting_data (  
    void )
```

Clear settings from persistent memory and RAM.

#### Returns

esp\_err\_t ESP\_OK on success. ESP\_FAIL if error occurs.

### 5.26.2.3 get\_setting\_byte()

```
esp_err_t get_setting_byte (  
    const char * name,  
    uint8_t * setting )
```

Recall a byte string setting. Uint8\_t is equivalent to unsigned char.

#### Parameters

<i>name</i>	The name of the setting to recall.
<i>setting</i>	The output of that setting. Set to NULL if not found, the value if found.

#### Returns

esp\_err\_t Returns ESP\_OK on success. ESP\_Fail if not found or an error occurred.



#### 5.26.2.4 `get_setting_double()`

```
esp_err_t get_setting_double (
    const char * name,
    double * setting )
```

Recall a persistent double setting. Exact storage depends on ArduinoJson configuration, but typically a maximum of 9 decimal places with rounding. No trailing zeros are included.

##### Parameters

<i>name</i>	The name of the setting to recall.
<i>setting</i>	The output of that setting. Set to NULL if not found, the value if found.

##### Returns

`esp_err_t` Returns `ESP_OK` on success. `ESP_Fail` if not found or an error occurred.

#### 5.26.2.5 `get_setting_int()`

```
esp_err_t get_setting_int (
    const char * name,
    int * setting )
```

Recall a persistent integer setting.

##### Parameters

<i>name</i>	The name of the setting to recall.
<i>setting</i>	The output of that setting. Set to NULL if not found, the value if found.

##### Returns

`esp_err_t` Returns `ESP_OK` on success. `ESP_Fail` if not found or an error occurred.

#### 5.26.2.6 `get_setting_string()`

```
esp_err_t get_setting_string (
    const char * name,
    char * setting )
```

Recall a persistent string setting.

**Parameters**

<i>name</i>	The name of the setting to recall.
<i>setting</i>	The output of that setting. Set to NULL if not found, the value if found.

**Returns**

`esp_err_t` Returns ESP\_OK on success. ESP\_Fail if not found or an error occurred.

**5.26.2.7 init\_memory()**

```
esp_err_t init_memory (  
    void )
```

Starts the memory system. Call this before calling any other memory related functions.

**Returns**

`esp_err_t` Returns ESP\_OK on success. ESP\_Fail or related if an error occurred.

**5.26.2.8 init\_spiffs()**

```
esp_err_t init_spiffs (  
    void )
```

**5.26.2.9 read\_settings\_to\_buffer()**

```
esp_err_t read_settings_to_buffer (  
    void )
```

**5.26.2.10 recall\_schedules()**

```
esp_err_t recall_schedules (  
    void )
```

Recall schedules from persistent memory. All schedules recalled will now be located in the tespective channel's schedule list.

**Returns**

`esp_err_t` Returns ESP\_OK on success. ESP\_Fail or related if an error occurred.

### 5.26.2.11 store\_schedules()

```
esp_err_t store_schedules (
    void )
```

Stores all currently running schedules in persistent memory.

#### Returns

esp\_err\_t Returns ESP\_OK on success. ESP\_Fail or related if an error occurred.

### 5.26.2.12 store\_setting\_byte()

```
esp_err_t store_setting_byte (
    const char * name,
    uint8_t setting )
```

Persistently store a byte setting.

#### Parameters

<i>name</i>	The name of the setting to store. If the setting with that name already exists it is overwritten with the new value.
<i>setting</i>	The setting to associate with the given name. Uint8_t is equivalent to an unsigned char.

#### Returns

esp\_err\_t Returns ESP\_OK on success. ESP\_Fail or related if an error occurred.

### 5.26.2.13 store\_setting\_double()

```
esp_err_t store_setting_double (
    const char * name,
    double setting )
```

Persistently store a double setting.

#### Parameters

<i>name</i>	The name of the setting to store. If the setting with that name already exists it is overwritten with the new value.
<i>setting</i>	The setting to associate with the given name. Exact storage depends on ArduinoJson configuration, but typically a maximum of 9 decimal places with rounding. No trailing zeros are included

**Returns**

`esp_err_t` Returns `ESP_OK` on success. `ESP_Fail` or related if an error occurred.

**5.26.2.14 store\_setting\_int()**

```
esp_err_t store_setting_int (
    const char * name,
    int setting )
```

Persistently store an integer setting.

**Parameters**

<i>name</i>	The name of the setting to store. If the setting with that name already exists it is overwritten with the new value.
<i>setting</i>	The setting to associate with the given name.

**Returns**

`esp_err_t` Returns `ESP_OK` on success. `ESP_Fail` or related if an error occurred.

**5.26.2.15 store\_setting\_string()**

```
esp_err_t store_setting_string (
    const char * name,
    char * setting )
```

Persistently store a string setting.

**Parameters**

<i>name</i>	The name of the setting to store. If the setting with that name already exists it is overwritten with the new value.
<i>setting</i>	The setting to associate with the given name.

**Returns**

`esp_err_t` Returns `ESP_OK` on success. `ESP_Fail` or related if an error occurred.

**5.26.3 Variable Documentation**

### 5.26.3.1 bSPIFFS

```
bool bSPIFFS = false
```

### 5.26.3.2 readNeeded

```
bool readNeeded = true
```

### 5.26.3.3 settingsString

```
char* settingsString
```

## 5.27 src/rtc.cpp File Reference

```
#include "rtcdefine.h"
```

### Functions

- void [rtc\\_config](#) (void)
- esp\_err\_t [readData](#) (uint32\_t addr, uint8\_t \*out)
- esp\_err\_t [writeData](#) (uint32\_t addr, uint8\_t data)
- esp\_err\_t [setTime](#) (const struct tm \*time)  
*Set the time of the external RTC to the specified value.*
- esp\_err\_t [getTime](#) (struct tm \*outTime)  
*Get the current time on the external RTC.*
- void [RTCHandler](#) (void \*pvParms)  
*The RTOS task that handles RTC related actions. This task synchronizes the esp32's and RTC's time.*
- esp\_err\_t [ST\\_StartRTCHandler](#) (void)  
*Starts the RTOS task that handles RTC and esp32 time synchronization.*

### Variables

- spi\_device\_handle\_t [rtc](#)

### 5.27.1 Detailed Description

Description: Provides function definitions to start an RTOS task and interface with an external RTC (MCP79510)  
Part of the timing subsystem

#### Author

: Primary: Shipra Vaidya, Secondary: Jesse Cannon

#### Date

last modified: 4/26/2020

## 5.27.2 Function Documentation

### 5.27.2.1 getTime()

```
esp_err_t getTime (
    struct tm * outTime )
```

Get the current time on the external RTC.

#### Parameters

<i>outTime</i>	The current time on the RTC will be stored in this value. Check the return type against ESP_OK to ensure this value was set correctly.
----------------	--

#### Returns

esp\_err\_t ESP\_OK on success, else on failure.

### 5.27.2.2 readData()

```
esp_err_t readData (
    uint32_t addr,
    uint8_t * out )
```

### 5.27.2.3 rtc\_config()

```
void rtc_config (
    void )
```

### 5.27.2.4 RTCHandler()

```
void RTCHandler (
    void * pvParams )
```

The RTOS task that handles RTC related actions. This task synchronizes the esp32's and RTC's time.

#### Parameters

<i>pvParams</i>	Required parameter of RTOS tasks. Not used in this task.
-----------------	--

### 5.27.2.5 setTime()

```
esp_err_t setTime (
    const struct tm * time )
```

Set the time of the external RTC to the specified value.

#### Parameters

<i>time</i>	The time to be set on the device.
-------------	-----------------------------------

#### Returns

esp\_err\_t ESP\_OK on success, else on failure.

### 5.27.2.6 ST\_StartRTCHandler()

```
esp_err_t ST_StartRTCHandler (
    void )
```

Starts the RTOS task that handles RTC and esp32 time synchronization.

#### Returns

esp\_err\_t ESP\_OK if task started successfully, else on failure.

### 5.27.2.7 writeData()

```
esp_err_t writeData (
    uint32_t addr,
    uint8_t data )
```

## 5.27.3 Variable Documentation

### 5.27.3.1 rtc

```
spi_device_handle_t rtc
```

## 5.28 src/scheduler.cpp File Reference

```
#include "scheduler.h"
```

### Macros

- #define [MIN](#)(a, b) a < b ? a : b
- #define [MAX](#)(a, b) a > b ? a : b

### Functions

- [esp\\_err\\_t create\\_schedule](#) (uint8\_t [channel](#), [schedule\\_object](#) s)  
*Create a schedule to associate with the specified channel. If the schedule already exists then it is replaced with the new schedule.*
- [esp\\_err\\_t delete\\_schedule\\_by\\_id](#) (uint8\_t [channel](#), uint8\_t ID)  
*Delete a schedule by ID.*
- [esp\\_err\\_t delete\\_schedule\\_by\\_name](#) (uint8\_t [channel](#), char \*name)  
*Delete a schedule by name.*
- [esp\\_err\\_t disable\\_schedule\\_by\\_id](#) (uint8\_t [channel](#), uint8\_t ID)  
*Disable a schedule by ID. The schedule remains in the linked list but does not run or perform updates.*
- [esp\\_err\\_t disable\\_schedule\\_by\\_name](#) (uint8\_t [channel](#), char \*name)  
*Disable a schedule by name. The schedule remains in the linked list but does not run or perform updates.*
- [esp\\_err\\_t enable\\_schedule\\_by\\_id](#) (uint8\_t [channel](#), uint8\_t ID)  
*Enable a schedule by ID. If the schedule was disabled, the schedule now runs and performs updates.*
- [esp\\_err\\_t enable\\_schedule\\_by\\_name](#) (uint8\_t [channel](#), char \*name)  
*Enable a schedule by name. If the schedule was disabled, the schedule now runs and performs updates.*
- [esp\\_err\\_t disable\\_all\\_schedules](#) (void)  
*Disables all schedules on all channels. The schedules remain located in memory but no longer run or perform updates.*
- [esp\\_err\\_t enable\\_all\\_schedules](#) (void)  
*Enable all schedules on all channels. If the schedule was disabled, the schedule now runs and performs updates.*
- [esp\\_err\\_t delete\\_all\\_schedules](#) (void)  
*Delete all schedules on all channels.*
- [esp\\_err\\_t get\\_schedule\\_names](#) (uint8\_t [channel](#), char \*&out)  
*Get a json string of all schedules and their enabled status for a given channel.*
- [esp\\_err\\_t get\\_schedule](#) (uint8\_t [channel](#), char \*name, [schedule\\_object](#) \*out)  
*Get the schedule object specified.*
- void [update\\_start\\_time](#) ([schedule\\_object](#) \*s, time\_t curr)
- void [init\\_schedule](#) (void)  
*Init the scheduler code. Starts the background task if not already started. If this is not called the schedules will not run.*

### Variables

- [List \\* schedules](#) [[NUM\\_CHANNELS](#)]  
*The linked list associated with each channel. Can be used directly instead of scheduler functions if so desired.*



## 5.28.1 Detailed Description

Description: Provides function definitions for running and maintaining repeating schedules as part of the scheduler subsystem.

Author

: Jesse Cannon

Date

last modified: 4/26/2020

## 5.28.2 Macro Definition Documentation

### 5.28.2.1 MAX

```
#define MAX(  
    a,  
    b ) a > b ? a : b
```

### 5.28.2.2 MIN

```
#define MIN(  
    a,  
    b ) a < b ? a : b
```

## 5.28.3 Function Documentation

### 5.28.3.1 create\_schedule()

```
esp_err_t create_schedule (  
    uint8_t channel,  
    schedule_object s )
```

Create a schedule to associate with the specified channel. If the schedule already exists then it is replaced with the new schedule.

Parameters

<i>channel</i>	The channel that the schedule will be placed on.
<i>s</i>	The schedule object that will be placed.

Generated by Doxygen

**Returns**

`esp_err_t` Returns `ESP_OK` on success. `ESP_ERR_INVALID_ARG` if channel number is invalid. `ESP_ERR_NO_MEM` if unable to allocate schedule.

**5.28.3.2 delete\_all\_schedules()**

```
esp_err_t delete_all_schedules (
    void )
```

Delete all schedules on all channels.

**Returns**

`esp_err_t` `ESP_OK` on successful deletion.

**5.28.3.3 delete\_schedule\_by\_id()**

```
esp_err_t delete_schedule_by_id (
    uint8_t channel,
    uint8_t ID )
```

Delete a schedule by ID.

**Parameters**

<i>channel</i>	The channel the schedule is located on.
<i>ID</i>	The ID of the schedule to delete.

**Returns**

`esp_err_t` Returns `ESP_OK` on successful deletion. `ESP_ERR_INVALID_ARG` if channel number is invalid. `ESP_ERR_NOT_FOUND` if schedule not found

**5.28.3.4 delete\_schedule\_by\_name()**

```
esp_err_t delete_schedule_by_name (
    uint8_t channel,
    char * name )
```

Delete a schedule by name.

## Parameters

<i>channel</i>	The channel the schedule is located on.
<i>name</i>	The name of the schedule to delete.

## Returns

esp\_err\_t Returns ESP\_OK on successful deletion. ESP\_ERR\_INVALID\_ARG if channel number is invalid. ESP\_ERR\_NOT\_FOUND if schedule not found

**5.28.3.5 disable\_all\_schedules()**

```
esp_err_t disable_all_schedules (
    void )
```

Disables all schedules on all channels. The schedules remain located in memory but no longer run or perform updates.

## Returns

esp\_err\_t ESP\_OK on successful disable.

**5.28.3.6 disable\_schedule\_by\_id()**

```
esp_err_t disable_schedule_by_id (
    uint8_t channel,
    uint8_t ID )
```

Disable a schedule by ID. The schedule remains in the linked list but does not run or perform updates.

## Parameters

<i>channel</i>	The channel the schedule is located on.
<i>ID</i>	The name of the schedule to disable.

## Returns

esp\_err\_t ESP\_OK on successful disable. ESP\_ERR\_INVALID\_ARG if channel number is invalid. ESP\_ERR\_NOT\_FOUND if schedule not found

**5.28.3.7 disable\_schedule\_by\_name()**

```
esp_err_t disable_schedule_by_name (
```

```
uint8_t channel,
char * name )
```

Disable a schedule by name. The schedule remains in the linked list but does not run or perform updates.

#### Parameters

<i>channel</i>	The channel the schedule is located on.
<i>name</i>	The name of the schedule to disable.

#### Returns

esp\_err\_t ESP\_OK on successful disable. ESP\_ERR\_INVALID\_ARG if channel number is invalid. ESP\_ERR\_NOT\_FOUND if schedule not found

### 5.28.3.8 enable\_all\_schedules()

```
esp_err_t enable_all_schedules (
    void )
```

Enable all schedules on all channels. If the schedule was disabled, the schedule now runs and performs updates.

#### Returns

esp\_err\_t ESP\_OK on successful disable.

### 5.28.3.9 enable\_schedule\_by\_id()

```
esp_err_t enable_schedule_by_id (
    uint8_t channel,
    uint8_t ID )
```

Enable a schedule by ID. If the schedule was disabled, the schedule now runs and performs updates.

#### Parameters

<i>channel</i>	The channel the schedule is located on.
<i>ID</i>	The ID of the schedule to enable.

#### Returns

esp\_err\_t ESP\_OK on successful enable. ESP\_ERR\_INVALID\_ARG if channel number is invalid. ESP\_ERR\_NOT\_FOUND if schedule not found

### 5.28.3.10 enable\_schedule\_by\_name()

```
esp_err_t enable_schedule_by_name (
    uint8_t channel,
    char * name )
```

Enable a schedule by name. If the schedule was disabled, the schedule now runs and performs updates.

#### Parameters

<i>channel</i>	The channel the schedule is located on.
<i>name</i>	The name of the schedule to enable.

#### Returns

esp\_err\_t ESP\_OK on successful enable. ESP\_ERR\_INVALID\_ARG if channel number is invalid. ESP\_ERR\_NOT\_FOUND if schedule not found

### 5.28.3.11 get\_schedule()

```
esp_err_t get_schedule (
    uint8_t channel,
    char * name,
    schedule_object * out )
```

Get the schedule object specified.

#### Parameters

<i>channel</i>	The channel the schedule is located on.
<i>name</i>	The name of the schedule to find.
<i>out</i>	The schedule returned. NULL if failure occurred.

#### Returns

esp\_err\_t ESP\_OK if successful. ESP\_ERR\_INVALID\_ARG if channel is invalid. ESP\_ERR\_NOT\_FOUND if the schedule could not be found.

### 5.28.3.12 get\_schedule\_names()

```
esp_err_t get_schedule_names (
    uint8_t channel,
    char *& out )
```

Get a json string of all schedules and their enabled status for a given channel.

## Parameters

<i>channel</i>	The channel to get the schedule names and status from.
<i>out</i>	The json string that is returned. Format is '{ "Name1":0, "Name2":1, "Name3":0 }'

## Returns

esp\_err\_t ESP\_OK if successful. ESP\_ERR\_INVALID\_ARG if channel is invalid.

## 5.28.3.13 init\_schedule()

```
void init_schedule (
    void )
```

Init the scheduler code. Starts the background task if not already started. If this is not called the schedules will not run.

## 5.28.3.14 update\_start\_time()

```
void update_start_time (
    schedule_object * s,
    time_t curr )
```

## 5.28.4 Variable Documentation

## 5.28.4.1 schedules

```
List* schedules[NUM_CHANNELS]
```

The linked list associated with each channel. Can be used directly instead of scheduler functions if so desired.

## 5.29 src/sdkconfig.h File Reference

### Macros

- #define [CONFIG\\_ENABLE\\_ARDUINO\\_DEPENDS](#) 1
- #define [CONFIG\\_AUTOSTART\\_ARDUINO](#) 1
- #define [CONFIG\\_ARDUINO\\_RUNNING\\_CORE](#) 1
- #define [CONFIG\\_ARDUINO\\_UDP\\_RUN\\_CORE](#) 1
- #define [CONFIG\\_ARDUINO\\_EVENT\\_RUN\\_CORE](#) 1
- #define [CONFIG\\_ARDUINO\\_EVENT\\_RUNNING\\_CORE](#) 1
- #define [CONFIG\\_ARDUINO\\_UDP\\_RUNNING\\_CORE](#) 1
- #define [CONFIG\\_GATTC\\_ENABLE](#) 1
- #define [CONFIG\\_ESP32\\_PHY\\_MAX\\_TX\\_POWER](#) 20
- #define [CONFIG\\_TRACEMEM\\_RESERVE\\_DRAM](#) 0x0
- #define [CONFIG\\_FREERTOS\\_MAX\\_TASK\\_NAME\\_LEN](#) 16
- #define [CONFIG\\_MQTT\\_TRANSPORT\\_SSL](#) 1
- #define [CONFIG\\_BLE\\_SMP\\_ENABLE](#) 1
- #define [CONFIG\\_FATFS\\_LFN\\_NONE](#) 1
- #define [CONFIG\\_SDP\\_INITIAL\\_TRACE\\_LEVEL](#) 2
- #define [CONFIG\\_MB\\_SERIAL\\_TASK\\_PRIO](#) 10
- #define [CONFIG\\_MQTT\\_PROTOCOL\\_311](#) 1
- #define [CONFIG\\_TCP\\_RECVMBOX\\_SIZE](#) 6
- #define [CONFIG\\_FATFS\\_CODEPAGE\\_437](#) 1
- #define [CONFIG\\_BLE\\_SCAN\\_DUPLICATE](#) 1
- #define [CONFIG\\_AVDT\\_TRACE\\_LEVEL\\_WARNING](#) 1
- #define [CONFIG\\_AWS\\_IOT\\_SHADOW\\_MAX\\_SIMULTANEOUS\\_ACKS](#) 10
- #define [CONFIG\\_TCP\\_WND\\_DEFAULT](#) 5744
- #define [CONFIG\\_PARTITION\\_TABLE\\_OFFSET](#) 0x8000
- #define [CONFIG\\_SW\\_COEXIST\\_ENABLE](#) 1
- #define [CONFIG\\_SPIFFS\\_USE\\_MAGIC\\_LENGTH](#) 1
- #define [CONFIG\\_AVCT\\_INITIAL\\_TRACE\\_LEVEL](#) 2
- #define [CONFIG\\_IPC\\_TASK\\_STACK\\_SIZE](#) 1024
- #define [CONFIG\\_WIFI\\_PROV\\_SCAN\\_MAX\\_ENTRIES](#) 16
- #define [CONFIG\\_FATFS\\_PER\\_FILE\\_CACHE](#) 1
- #define [CONFIG\\_ESPTOOLPY\\_FLASHFREQ](#) "40m"
- #define [CONFIG\\_AWS\\_IOT\\_SHADOW\\_MAX\\_SIZE\\_OF\\_THING\\_NAME](#) 20
- #define [CONFIG\\_MBEDTLS\\_KEY\\_EXCHANGE\\_RSA](#) 1
- #define [CONFIG\\_UDP\\_RECVMBOX\\_SIZE](#) 6
- #define [CONFIG\\_SPI\\_FLASH\\_YIELD\\_DURING\\_ERASE](#) 1
- #define [CONFIG\\_FREERTOS\\_QUEUE\\_REGISTRY\\_SIZE](#) 0
- #define [CONFIG\\_MBEDTLS\\_AES\\_C](#) 1
- #define [CONFIG\\_MBEDTLS\\_ECP\\_DP\\_SECP521R1\\_ENABLED](#) 1
- #define [CONFIG\\_ESP32\\_WIFI\\_SOFTAP\\_BEACON\\_MAX\\_LEN](#) 752
- #define [CONFIG\\_MBEDTLS\\_GCM\\_C](#) 1
- #define [CONFIG\\_ESPTOOLPY\\_FLASHSIZE](#) "2MB"
- #define [CONFIG\\_HEAP\\_POISONING\\_DISABLED](#) 1
- #define [CONFIG\\_SPIFFS\\_CACHE\\_WR](#) 1
- #define [CONFIG\\_BROWNOUT\\_DET\\_LVL\\_SEL\\_0](#) 1
- #define [CONFIG\\_ESP32\\_WIFI\\_DYNAMIC\\_TX\\_BUFFER](#) 1
- #define [CONFIG\\_SPIFFS\\_CACHE](#) 1
- #define [CONFIG\\_INT\\_WDT](#) 1
- #define [CONFIG\\_BTDM\\_CONTROLLER\\_BLE\\_MAX\\_CONN](#) 3
- #define [CONFIG\\_MBEDTLS\\_SSL\\_PROTO\\_TLS1](#) 1
- #define [CONFIG\\_ESP\\_GRATUITOUS\\_ARP](#) 1

- #define CONFIG\_AWS\_IOT\_SHADOW\_MAX\_SIZE\_OF\_UNIQUE\_CLIENT\_ID\_BYTES 80
- #define CONFIG\_MBEDTLS\_ECDSA\_C 1
- #define CONFIG\_ESPTOOLPY\_FLASHFREQ\_40M 1
- #define CONFIG\_LOG\_BOOTLOADER\_LEVEL\_INFO 1
- #define CONFIG\_ESPTOOLPY\_FLASHSIZE\_2MB 1
- #define CONFIG\_HTTPD\_MAX\_REQ\_HDR\_LEN 512
- #define CONFIG\_BTDM\_CONTROLLER\_PINNED\_TO\_CORE 0
- #define CONFIG\_AWS\_IOT\_MQTT\_PORT 8883
- #define CONFIG\_FREERTOS\_THREAD\_LOCAL\_STORAGE\_POINTERS 1
- #define CONFIG\_MBEDTLS\_ECDH\_C 1
- #define CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_ELLIPTIC\_CURVE 1
- #define CONFIG\_ESP32\_WIFI\_STATIC\_RX\_BUFFER\_NUM 10
- #define CONFIG\_AWS\_IOT\_MQTT\_MAX\_RECONNECT\_WAIT\_INTERVAL 128000
- #define CONFIG\_MBEDTLS\_SSL\_ALPN 1
- #define CONFIG\_BTDM\_TRACE\_LEVEL\_WARNING 1
- #define CONFIG\_MBEDTLS\_PEM\_WRITE\_C 1
- #define CONFIG\_RFCOMM\_TRACE\_LEVEL\_WARNING 1
- #define CONFIG\_LOG\_DEFAULT\_LEVEL\_INFO 1
- #define CONFIG\_BT\_RESERVE\_DRAM 0xdb5c
- #define CONFIG\_APP\_COMPILE\_TIME\_DATE 1
- #define CONFIG\_FATFS\_FS\_LOCK 0
- #define CONFIG\_IP\_LOST\_TIMER\_INTERVAL 120
- #define CONFIG\_SPIFFS\_META\_LENGTH 4
- #define CONFIG\_ESP32\_PANIC\_PRINT\_REBOOT 1
- #define CONFIG\_MB\_CONTROLLER\_NOTIFY\_QUEUE\_SIZE 20
- #define CONFIG\_MBEDTLS\_ECP\_DP\_BP384R1\_ENABLED 1
- #define CONFIG\_MBEDTLS\_ECP\_DP\_SECP256K1\_ENABLED 1
- #define CONFIG\_AWS\_IOT\_MQTT\_RX\_BUF\_LEN 512
- #define CONFIG\_MB\_SERIAL\_BUF\_SIZE 256
- #define CONFIG\_CONSOLE\_UART\_BAUDRATE 115200
- #define CONFIG\_LWIP\_MAX\_SOCKETS 10
- #define CONFIG\_LWIP\_NETIF\_LOOPBACK 1
- #define CONFIG\_MCA\_TRACE\_LEVEL\_WARNING 1
- #define CONFIG\_ESP32\_PTHREAD\_TASK\_NAME\_DEFAULT "pthread"
- #define CONFIG\_EMAC\_TASK\_PRIORITY 20
- #define CONFIG\_TIMER\_TASK\_STACK\_DEPTH 2048
- #define CONFIG\_TCP\_MSS 1436
- #define CONFIG\_MBEDTLS\_ECP\_DP\_CURVE25519\_ENABLED 1
- #define CONFIG\_BTIF\_INITIAL\_TRACE\_LEVEL 2
- #define CONFIG\_BTDM\_CONTROLLER\_BLE\_MAX\_CONN\_EFF 3
- #define CONFIG\_EFUSE\_CODE\_SCHEME\_COMPAT\_3\_4 1
- #define CONFIG\_FATFS\_CODEPAGE 437
- #define CONFIG\_APPL\_TRACE\_LEVEL\_WARNING 1
- #define CONFIG\_BTC\_INITIAL\_TRACE\_LEVEL 2
- #define CONFIG\_ESP32\_DEFAULT\_CPU\_FREQ\_160 1
- #define CONFIG\_ULP\_COPROC\_RESERVE\_MEM 0
- #define CONFIG\_LWIP\_MAX\_UDP\_PCBS 16
- #define CONFIG\_ESPTOOLPY\_BAUD 115200
- #define CONFIG\_INT\_WDT\_CHECK\_CPU1 1
- #define CONFIG\_AVRC\_INITIAL\_TRACE\_LEVEL 2
- #define CONFIG\_ADC\_CAL\_LUT\_ENABLE 1
- #define CONFIG\_AWS\_IOT\_MQTT\_TX\_BUF\_LEN 512
- #define CONFIG\_FLASHMODE\_DIO 1
- #define CONFIG\_ESPTOOLPY\_AFTER\_RESET 1
- #define CONFIG\_OPTIMIZATION\_ASSERTIONS\_ENABLED 1



- #define CONFIG\_LWIP\_DHCP\_MAX\_STATION\_NUM 8
- #define CONFIG\_TOOLPREFIX "xtensa-esp32-elf-"
- #define CONFIG\_MBEDTLS\_ECP\_C 1
- #define CONFIG\_FREERTOS\_IDLE\_TASK\_STACKSIZE 1536
- #define CONFIG\_MBEDTLS\_RC4\_DISABLED 1
- #define CONFIG\_GAP\_TRACE\_LEVEL\_WARNING 1
- #define CONFIG\_CONSOLE\_UART\_NUM 0
- #define CONFIG\_AWS\_IOT\_SHADOW\_MAX\_JSON\_TOKEN\_EXPECTED 120
- #define CONFIG\_ESP32\_APPTRACE\_LOCK\_ENABLE 1
- #define CONFIG\_PTHREAD\_STACK\_MIN 768
- #define CONFIG\_ESP32\_RTC\_CLOCK\_SOURCE\_INTERNAL\_RC 1
- #define CONFIG\_ESPTOOLPY\_BAUD\_115200B 1
- #define CONFIG\_TCP\_OVERSIZE\_MSS 1
- #define CONFIG\_FOUR\_UNIVERSAL\_MAC\_ADDRESS 1
- #define CONFIG\_CONSOLE\_UART\_DEFAULT 1
- #define CONFIG\_MBEDTLS\_SSL\_MAX\_CONTENT\_LEN 16384
- #define CONFIG\_NUMBER\_OF\_UNIVERSAL\_MAC\_ADDRESS 4
- #define CONFIG\_GATT\_TRACE\_LEVEL\_WARNING 1
- #define CONFIG\_ESPTOOLPY\_FLASHSIZE\_DETECT 1
- #define CONFIG\_TIMER\_TASK\_STACK\_SIZE 3584
- #define CONFIG\_BTIF\_TRACE\_LEVEL\_WARNING 1
- #define CONFIG\_ESP32\_ENABLE\_COREDUMP\_TO\_NONE 1
- #define CONFIG\_HCI\_INITIAL\_TRACE\_LEVEL 2
- #define CONFIG\_AVDT\_INITIAL\_TRACE\_LEVEL 2
- #define CONFIG\_MBEDTLS\_X509\_CRL\_PARSE\_C 1
- #define CONFIG\_FREERTOS\_CHECK\_MUTEX\_GIVEN\_BY\_OWNER 1
- #define CONFIG\_HTTPD\_PURGE\_BUF\_LEN 32
- #define CONFIG\_SCAN\_DUPLICATE\_BY\_DEVICE\_ADDR 1
- #define CONFIG\_AWS\_IOT\_SHADOW\_MAX\_SHADOW\_TOPIC\_LENGTH\_WITHOUT\_THINGNAME 60
- #define CONFIG\_UNITY\_ENABLE\_IDF\_TEST\_RUNNER 1
- #define CONFIG\_MB\_SERIAL\_TASK\_STACK\_SIZE 2048
- #define CONFIG\_GATTS\_SEND\_SERVICE\_CHANGE\_AUTO 1
- #define CONFIG\_LWIP\_DHCP\_LEASE\_UNIT 60
- #define CONFIG\_EFUSE\_MAX\_BLK\_LEN 192
- #define CONFIG\_SPIFFS\_USE\_MAGIC 1
- #define CONFIG\_TCPIP\_TASK\_STACK\_SIZE 2048
- #define CONFIG\_BLUFI\_TRACE\_LEVEL\_WARNING 1
- #define CONFIG\_BLUEDROID\_PINNED\_TO\_CORE\_0 1
- #define CONFIG\_TASK\_WDT 1
- #define CONFIG\_RFCOMM\_INITIAL\_TRACE\_LEVEL 2
- #define CONFIG\_MAIN\_TASK\_STACK\_SIZE 3584
- #define CONFIG\_SPIFFS\_PAGE\_CHECK 1
- #define CONFIG\_ESP32\_WIFI\_TASK\_PINNED\_TO\_CORE\_0 1
- #define CONFIG\_LWIP\_MAX\_ACTIVE\_TCP 16
- #define CONFIG\_TASK\_WDT\_TIMEOUT\_S 5
- #define CONFIG\_INT\_WDT\_TIMEOUT\_MS 300
- #define CONFIG\_ESPTOOLPY\_FLASHMODE "dio"
- #define CONFIG\_BTC\_TASK\_STACK\_SIZE 3072
- #define CONFIG\_BLUEDROID\_ENABLED 1
- #define CONFIG\_NEWLIB\_STDIN\_LINE\_ENDING\_CR 1
- #define CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_ECDH\_RSA 1
- #define CONFIG\_ESPTOOLPY\_BEFORE "default\_reset"
- #define CONFIG\_ADC2\_DISABLE\_DAC 1
- #define CONFIG\_BLE\_ADV\_REPORT\_FLOW\_CONTROL\_NUM 100
- #define CONFIG\_ESP32\_REV\_MIN\_0 1

- #define CONFIG\_LOG\_DEFAULT\_LEVEL 3
- #define CONFIG\_FREERTOS\_ASSERT\_ON\_UNTESTED\_FUNCTION 1
- #define CONFIG\_TIMER\_QUEUE\_LENGTH 10
- #define CONFIG\_ESP32\_REV\_MIN 0
- #define CONFIG\_SUPPRESS\_SELECT\_DEBUG\_OUTPUT 1
- #define CONFIG\_GATTS\_SEND\_SERVICE\_CHANGE\_MODE 0
- #define CONFIG\_TCPIP\_TASK\_AFFINITY\_NO\_AFFINITY 1
- #define CONFIG\_MAKE\_WARN\_UNDEFINED\_VARIABLES 1
- #define CONFIG\_FATFS\_TIMEOUT\_MS 10000
- #define CONFIG\_ESP32\_WIFI\_DYNAMIC\_RX\_BUFFER\_NUM 32
- #define CONFIG\_ESP\_HTTP\_CLIENT\_ENABLE\_HTTPS 1
- #define CONFIG\_PAN\_INITIAL\_TRACE\_LEVEL 2
- #define CONFIG\_MBEDTLS\_CCM\_C 1
- #define CONFIG\_SPI\_MASTER\_ISR\_IN\_IRAM 1
- #define CONFIG\_MCA\_INITIAL\_TRACE\_LEVEL 2
- #define CONFIG\_ESP32\_PHY\_MAX\_WIFI\_TX\_POWER 20
- #define CONFIG\_A2D\_INITIAL\_TRACE\_LEVEL 2
- #define CONFIG\_ESP32\_RTC\_CLK\_CAL\_CYCLES 1024
- #define CONFIG\_ESP32\_WIFI\_TX\_BA\_WIN 6
- #define CONFIG\_ESP32\_WIFI\_NVS\_ENABLED 1
- #define CONFIG\_MDNS\_MAX\_SERVICES 10
- #define CONFIG\_IDF\_TARGET\_ESP32 1
- #define CONFIG\_ETH\_CHECK\_LINK\_PERIOD\_MS 2000
- #define CONFIG\_MBEDTLS\_ECP\_DP\_SECP224R1\_ENABLED 1
- #define CONFIG\_SPI\_FLASH\_ERASE\_YIELD\_DURATION\_MS 20
- #define CONFIG\_LIBSODIUM\_USE\_MBEDTLS\_SHA 1
- #define CONFIG\_AWS\_IOT\_SDK 1
- #define CONFIG\_DMA\_RX\_BUF\_NUM 10
- #define CONFIG\_MBEDTLS\_ECP\_DP\_SECP384R1\_ENABLED 1
- #define CONFIG\_TCP\_SYNMAXRTX 6
- #define CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_ECDHE\_ECDSA 1
- #define CONFIG\_BTDM\_CONTROLLER\_BR\_EDR\_MAX\_SYNC\_CONN\_EFF 0
- #define CONFIG\_PYTHON "python"
- #define CONFIG\_MBEDTLS\_ECP\_NIST\_OPTIM 1
- #define CONFIG\_ESP32\_TIME\_SYSCALL\_USE\_RTC\_FRC1 1
- #define CONFIG\_ESPTOOLPY\_COMPRESSED 1
- #define CONFIG\_PARTITION\_TABLE\_FILENAME "partitions\_singleapp.csv"
- #define CONFIG\_MB\_CONTROLLER\_STACK\_SIZE 4096
- #define CONFIG\_TCP\_SND\_BUF\_DEFAULT 5744
- #define CONFIG\_GARP\_TMR\_INTERVAL 60
- #define CONFIG\_LWIP\_DHCP\_MAX\_NTP\_SERVERS 1
- #define CONFIG\_BNEP\_INITIAL\_TRACE\_LEVEL 2
- #define CONFIG\_HCI\_TRACE\_LEVEL\_WARNING 1
- #define CONFIG\_TCP\_MSL 60000
- #define CONFIG\_MBEDTLS\_SSL\_PROTO\_TLS1\_1 1
- #define CONFIG\_LWIP\_SO\_REUSE\_RXTOALL 1
- #define CONFIG\_MB\_CONTROLLER\_NOTIFY\_TIMEOUT 20
- #define CONFIG\_ESP32\_WIFI\_MGMT\_SBUF\_NUM 32
- #define CONFIG\_PARTITION\_TABLE\_SINGLE\_APP 1
- #define CONFIG\_UNITY\_ENABLE\_FLOAT 1
- #define CONFIG\_ESP32\_WIFI\_RX\_BA\_WIN 6
- #define CONFIG\_MBEDTLS\_X509\_CSR\_PARSE\_C 1
- #define CONFIG\_SPIFFS\_USE\_MTIME 1
- #define CONFIG\_BT\_TRACE\_LEVEL\_WARNING 1
- #define CONFIG\_ETH\_TASK\_STACK\_SIZE 3072

- `#define CONFIG_SMP_TRACE_LEVEL_WARNING 1`
- `#define CONFIG_MB_QUEUE_LENGTH 20`
- `#define CONFIG_SW_COEXIST_PREFERENCE_VALUE 2`
- `#define CONFIG_MBEDTLS_KEY_EXCHANGE_ECDHE_RSA 1`
- `#define CONFIG_LWIP_DHCP_DOES_ARP_CHECK 1`
- `#define CONFIG_FREERTOS_TASK_FUNCTION_WRAPPER 1`
- `#define CONFIG_SYSTEM_EVENT_TASK_STACK_SIZE 2304`
- `#define CONFIG_BOOTLOADER_VDDSDIO_BOOST_1_9V 1`
- `#define CONFIG_A2D_TRACE_LEVEL_WARNING 1`
- `#define CONFIG_ESP32_DEEP_SLEEP_WAKEUP_DELAY 2000`
- `#define CONFIG_AWS_IOT_MQTT_NUM_SUBSCRIBE_HANDLERS 5`
- `#define CONFIG_BROWNOUT_DET_LVL 0`
- `#define CONFIG_MBEDTLS_PEM_PARSE_C 1`
- `#define CONFIG_SPIFFS_GC_MAX_RUNS 10`
- `#define CONFIG_ESP32_APPTRACE_DEST_NONE 1`
- `#define CONFIG_MBEDTLS_INTERNAL_MEM_ALLOC 1`
- `#define CONFIG_MBEDTLS_SSL_PROTO_TLS1_2 1`
- `#define CONFIG_MBEDTLS_KEY_EXCHANGE_DHE_RSA 1`
- `#define CONFIG_ESP32_WIFI_DYNAMIC_TX_BUFFER_NUM 32`
- `#define CONFIG_HTTPD_MAX_URI_LEN 512`
- `#define CONFIG_MBEDTLS_ECP_DP_BP256R1_ENABLED 1`
- `#define CONFIG_AVCT_TRACE_LEVEL_WARNING 1`
- `#define CONFIG_MBEDTLS_ECP_DP_SECP224K1_ENABLED 1`
- `#define CONFIG_TASK_WDT_CHECK_IDLE_TASK_CPU1 1`
- `#define CONFIG_ESP32_DEFAULT_CPU_FREQ_MHZ 160`
- `#define CONFIG_MBEDTLS_HARDWARE_AES 1`
- `#define CONFIG_FREERTOS_HZ 100`
- `#define CONFIG_LOG_COLORS 1`
- `#define CONFIG_OSI_TRACE_LEVEL_WARNING 1`
- `#define CONFIG_ESP32_PHY_CALIBRATION_AND_DATA_STORAGE 1`
- `#define CONFIG_STACK_CHECK_NONE 1`
- `#define CONFIG_ADC_CAL_EFUSE_TP_ENABLE 1`
- `#define CONFIG_BNEP_TRACE_LEVEL_WARNING 1`
- `#define CONFIG_FREERTOS_ASSERT_FAIL_ABORT 1`
- `#define CONFIG_BROWNOUT_DET 1`
- `#define CONFIG_AWS_IOT_SHADOW_MAX_SIMULTANEOUS_THINGNAMES 10`
- `#define CONFIG_ESP32_XTAL_FREQ 40`
- `#define CONFIG_OSI_INITIAL_TRACE_LEVEL 2`
- `#define CONFIG_MONITOR_BAUD_115200B 1`
- `#define CONFIG_LOG_BOOTLOADER_LEVEL 3`
- `#define CONFIG_MBEDTLS_TLS_ENABLED 1`
- `#define CONFIG_LWIP_MAX_RAW_PCBS 16`
- `#define CONFIG_BTU_TASK_STACK_SIZE 4096`
- `#define CONFIG_SMP_ENABLE 1`
- `#define CONFIG_HID_TRACE_LEVEL_WARNING 1`
- `#define CONFIG_AVRC_TRACE_LEVEL_WARNING 1`
- `#define CONFIG_MBEDTLS_SSL_SESSION_TICKETS 1`
- `#define CONFIG_SPIFFS_MAX_PARTITIONS 3`
- `#define CONFIG_ESP_ERR_TO_NAME_LOOKUP 1`
- `#define CONFIG_BTDM_CONTROLLER_PINNED_TO_CORE_0 1`
- `#define CONFIG_MBEDTLS_SSL_RENEGOTIATION 1`
- `#define CONFIG_HID_INITIAL_TRACE_LEVEL 2`
- `#define CONFIG_ESPTOOLPY_BEFORE_RESET 1`
- `#define CONFIG_MB_EVENT_QUEUE_TIMEOUT 20`
- `#define CONFIG_ESPTOOLPY_BAUD_OTHER_VAL 115200`

- #define CONFIG\_SPIFFS\_OBJ\_NAME\_LEN 32
- #define CONFIG\_ESP32\_PTHREAD\_TASK\_PRIO\_DEFAULT 5
- #define CONFIG\_BTDM\_CONTROLLER\_BR\_EDR\_MAX\_ACL\_CONN\_EFF 0
- #define CONFIG\_PARTITION\_TABLE\_MD5 1
- #define CONFIG\_TCPIP\_RECVMBOX\_SIZE 32
- #define CONFIG\_TCP\_MAXRTX 12
- #define CONFIG\_BTM\_INITIAL\_TRACE\_LEVEL 2
- #define CONFIG\_ESPTOOLPY\_AFTER "hard\_reset"
- #define CONFIG\_TCPIP\_TASK\_AFFINITY 0x7FFFFFFF
- #define CONFIG\_LWIP\_SO\_REUSE 1
- #define CONFIG\_ESP32\_XTAL\_FREQ\_40 1
- #define CONFIG\_BTDM\_CONTROLLER\_MODE\_BLE\_ONLY 1
- #define CONFIG\_DMA\_TX\_BUF\_NUM 10
- #define CONFIG\_LWIP\_MAX\_LISTENING\_TCP 16
- #define CONFIG\_FREERTOS\_INTERRUPT\_BACKTRACE 1
- #define CONFIG\_WL\_SECTOR\_SIZE 4096
- #define CONFIG\_ESP32\_DEBUG\_OCDAWARE 1
- #define CONFIG\_MQTT\_TRANSPORT\_WEBSOCKET 1
- #define CONFIG\_TIMER\_TASK\_PRIORITY 1
- #define CONFIG\_MBEDTLS\_TLS\_CLIENT 1
- #define CONFIG\_AWS\_IOT\_MQTT\_MIN\_RECONNECT\_WAIT\_INTERVAL 1000
- #define CONFIG\_BTDM\_CONTROLLER\_HCI\_MODE\_VHCI 1
- #define CONFIG\_BT\_ENABLED 1
- #define CONFIG\_ESP32\_DEFAULT\_PTHREAD\_CORE\_NO\_AFFINITY 1
- #define CONFIG\_SDP\_TRACE\_LEVEL\_WARNING 1
- #define CONFIG\_SW\_COEXIST\_PREFERENCE\_BALANCE 1
- #define CONFIG\_MBEDTLS\_ECP\_DP\_SECP256R1\_ENABLED 1
- #define CONFIG\_MONITOR\_BAUD 115200
- #define CONFIG\_ESP32\_PTHREAD\_TASK\_CORE\_DEFAULT -1
- #define CONFIG\_ESP32\_DEBUG\_STUBS\_ENABLE 1
- #define CONFIG\_BLE\_ESTABLISH\_LINK\_CONNECTION\_TIMEOUT 30
- #define CONFIG\_TCPIP\_LWIP 1
- #define CONFIG\_REDUCE\_PHY\_TX\_POWER 1
- #define CONFIG\_BOOTLOADER\_WDT\_TIME\_MS 9000
- #define CONFIG\_PAN\_TRACE\_LEVEL\_WARNING 1
- #define CONFIG\_FREERTOS\_CORETIMER\_0 1
- #define CONFIG\_PARTITION\_TABLE\_CUSTOM\_FILENAME "partitions.csv"
- #define CONFIG\_MBEDTLS\_HAVE\_TIME 1
- #define CONFIG\_FREERTOS\_CHECK\_STACKOVERFLOW\_CANARY 1
- #define CONFIG\_TCP\_QUEUE\_OOSEQ 1
- #define CONFIG\_GATTS\_ENABLE 1
- #define CONFIG\_ADC\_CAL\_EFUSE\_VREF\_ENABLE 1
- #define CONFIG\_MBEDTLS\_TLS\_SERVER 1
- #define CONFIG\_MBEDTLS\_TLS\_SERVER\_AND\_CLIENT 1
- #define CONFIG\_BLE\_ADV\_REPORT\_FLOW\_CONTROL\_SUPPORTED 1
- #define CONFIG\_FREERTOS\_ISR\_STACKSIZE 1536
- #define CONFIG\_SUPPORT\_TERMIOS 1
- #define CONFIG\_OPENSSL\_ASSERT\_DO\_NOTHING 1
- #define CONFIG\_IDF\_TARGET "esp32"
- #define CONFIG\_WL\_SECTOR\_SIZE\_4096 1
- #define CONFIG\_OPTIMIZATION\_LEVEL\_DEBUG 1
- #define CONFIG\_GATT\_INITIAL\_TRACE\_LEVEL 2
- #define CONFIG\_FREERTOS\_NO\_AFFINITY 0x7FFFFFFF
- #define CONFIG\_AWS\_IOT\_MQTT\_HOST ""
- #define CONFIG\_L2CAP\_TRACE\_LEVEL\_WARNING 1

- #define CONFIG\_ESP32\_WIFI\_AMPDU\_TX\_ENABLED 1
- #define CONFIG\_HTTPD\_ERR\_RESP\_NO\_DELAY 1
- #define CONFIG\_MB\_TIMER\_INDEX 0
- #define CONFIG\_SCAN\_DUPLICATE\_TYPE 0
- #define CONFIG\_MBEDTLS\_ECP\_DP\_SECP192R1\_ENABLED 1
- #define CONFIG\_APPL\_INITIAL\_TRACE\_LEVEL 2
- #define CONFIG\_MBEDTLS\_ECP\_DP\_BP512R1\_ENABLED 1
- #define CONFIG\_SPI\_FLASH\_ERASE\_YIELD\_TICKS 1
- #define CONFIG\_SMP\_INITIAL\_TRACE\_LEVEL 2
- #define CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_ECDH\_ECDSA 1
- #define CONFIG\_SPI\_SLAVE\_ISR\_IN\_IRAM 1
- #define CONFIG\_L2CAP\_INITIAL\_TRACE\_LEVEL 2
- #define CONFIG\_SYSTEM\_EVENT\_QUEUE\_SIZE 32
- #define CONFIG\_BT\_ACL\_CONNECTIONS 4
- #define CONFIG\_ESP32\_WIFI\_TX\_BUFFER\_TYPE 1
- #define CONFIG\_BOOTLOADER\_WDT\_ENABLE 1
- #define CONFIG\_GAP\_INITIAL\_TRACE\_LEVEL 2
- #define CONFIG\_ESP32\_WIFI\_AMPDU\_RX\_ENABLED 1
- #define CONFIG\_LWIP\_LOOPBACK\_MAX\_PBUFS 8
- #define CONFIG\_MB\_TIMER\_GROUP 0
- #define CONFIG\_SPI\_FLASH\_ROM\_DRIVER\_PATCH 1
- #define CONFIG\_MQTT\_TRANSPORT\_WEBSOCKET\_SECURE 1
- #define CONFIG\_SPIFFS\_PAGE\_SIZE 256
- #define CONFIG\_MBEDTLS\_ECP\_DP\_SECP192K1\_ENABLED 1
- #define CONFIG\_ESP32\_DPORT\_WORKAROUND 1
- #define CONFIG\_TASK\_WDT\_CHECK\_IDLE\_TASK\_CPU0 1
- #define CONFIG\_ESP32\_PTHREAD\_TASK\_STACK\_SIZE\_DEFAULT 3072
- #define CONFIG\_MB\_TIMER\_PORT\_ENABLED 1
- #define CONFIG\_DUPLICATE\_SCAN\_CACHE\_SIZE 50
- #define CONFIG\_MONITOR\_BAUD\_OTHER\_VAL 115200
- #define CONFIG\_NEWLIB\_STDOUT\_LINE\_ENDING\_CRLF 1
- #define CONFIG\_ESPTOOLPY\_PORT "COM19"
- #define CONFIG\_SPI\_FLASH\_WRITING\_DANGEROUS\_REGIONS\_ABORTS 1
- #define CONFIG\_UNITY\_ENABLE\_DOUBLE 1
- #define CONFIG\_BLE\_ADV\_REPORT\_DISCARD\_THRESHOLD 20
- #define CONFIG\_BLUEDROID\_PINNED\_TO\_CORE 0
- #define CONFIG\_ESP32\_WIFI\_IRAM\_OPT 1
- #define CONFIG\_BLUFI\_INITIAL\_TRACE\_LEVEL 2

## 5.29.1 Macro Definition Documentation

### 5.29.1.1 CONFIG\_A2D\_INITIAL\_TRACE\_LEVEL

```
#define CONFIG_A2D_INITIAL_TRACE_LEVEL 2
```

#### 5.29.1.2 CONFIG\_A2D\_TRACE\_LEVEL\_WARNING

```
#define CONFIG_A2D_TRACE_LEVEL_WARNING 1
```

#### 5.29.1.3 CONFIG\_ADC2\_DISABLE\_DAC

```
#define CONFIG_ADC2_DISABLE_DAC 1
```

#### 5.29.1.4 CONFIG\_ADC\_CAL\_EFUSE\_TP\_ENABLE

```
#define CONFIG_ADC_CAL_EFUSE_TP_ENABLE 1
```

#### 5.29.1.5 CONFIG\_ADC\_CAL\_EFUSE\_VREF\_ENABLE

```
#define CONFIG_ADC_CAL_EFUSE_VREF_ENABLE 1
```

#### 5.29.1.6 CONFIG\_ADC\_CAL\_LUT\_ENABLE

```
#define CONFIG_ADC_CAL_LUT_ENABLE 1
```

#### 5.29.1.7 CONFIG\_APP\_COMPILE\_TIME\_DATE

```
#define CONFIG_APP_COMPILE_TIME_DATE 1
```

#### 5.29.1.8 CONFIG\_APPL\_INITIAL\_TRACE\_LEVEL

```
#define CONFIG_APPL_INITIAL_TRACE_LEVEL 2
```

#### 5.29.1.9 CONFIG\_APPL\_TRACE\_LEVEL\_WARNING

```
#define CONFIG_APPL_TRACE_LEVEL_WARNING 1
```

**5.29.1.10 CONFIG\_ARDUINO\_EVENT\_RUN\_CORE1**

```
#define CONFIG_ARDUINO_EVENT_RUN_CORE1 1
```

**5.29.1.11 CONFIG\_ARDUINO\_EVENT\_RUNNING\_CORE**

```
#define CONFIG_ARDUINO_EVENT_RUNNING_CORE 1
```

**5.29.1.12 CONFIG\_ARDUINO\_RUNNING\_CORE**

```
#define CONFIG_ARDUINO_RUNNING_CORE 1
```

**5.29.1.13 CONFIG\_ARDUINO\_UDP\_RUN\_CORE1**

```
#define CONFIG_ARDUINO_UDP_RUN_CORE1 1
```

**5.29.1.14 CONFIG\_ARDUINO\_UDP\_RUNNING\_CORE**

```
#define CONFIG_ARDUINO_UDP_RUNNING_CORE 1
```

**5.29.1.15 CONFIG\_AUTOSTART\_ARDUINO**

```
#define CONFIG_AUTOSTART_ARDUINO 1
```

**5.29.1.16 CONFIG\_AVCT\_INITIAL\_TRACE\_LEVEL**

```
#define CONFIG_AVCT_INITIAL_TRACE_LEVEL 2
```

**5.29.1.17 CONFIG\_AVCT\_TRACE\_LEVEL\_WARNING**

```
#define CONFIG_AVCT_TRACE_LEVEL_WARNING 1
```

**5.29.1.18 CONFIG\_AVDT\_INITIAL\_TRACE\_LEVEL**

```
#define CONFIG_AVDT_INITIAL_TRACE_LEVEL 2
```

**5.29.1.19 CONFIG\_AVDT\_TRACE\_LEVEL\_WARNING**

```
#define CONFIG_AVDT_TRACE_LEVEL_WARNING 1
```

**5.29.1.20 CONFIG\_AVRC\_INITIAL\_TRACE\_LEVEL**

```
#define CONFIG_AVRC_INITIAL_TRACE_LEVEL 2
```

**5.29.1.21 CONFIG\_AVRC\_TRACE\_LEVEL\_WARNING**

```
#define CONFIG_AVRC_TRACE_LEVEL_WARNING 1
```

**5.29.1.22 CONFIG\_AWS\_IOT\_MQTT\_HOST**

```
#define CONFIG_AWS_IOT_MQTT_HOST ""
```

**5.29.1.23 CONFIG\_AWS\_IOT\_MQTT\_MAX\_RECONNECT\_WAIT\_INTERVAL**

```
#define CONFIG_AWS_IOT_MQTT_MAX_RECONNECT_WAIT_INTERVAL 128000
```

**5.29.1.24 CONFIG\_AWS\_IOT\_MQTT\_MIN\_RECONNECT\_WAIT\_INTERVAL**

```
#define CONFIG_AWS_IOT_MQTT_MIN_RECONNECT_WAIT_INTERVAL 1000
```

**5.29.1.25 CONFIG\_AWS\_IOT\_MQTT\_NUM\_SUBSCRIBE\_HANDLERS**

```
#define CONFIG_AWS_IOT_MQTT_NUM_SUBSCRIBE_HANDLERS 5
```



**5.29.1.26 CONFIG\_AWS\_IOT\_MQTT\_PORT**

```
#define CONFIG_AWS_IOT_MQTT_PORT 8883
```

**5.29.1.27 CONFIG\_AWS\_IOT\_MQTT\_RX\_BUF\_LEN**

```
#define CONFIG_AWS_IOT_MQTT_RX_BUF_LEN 512
```

**5.29.1.28 CONFIG\_AWS\_IOT\_MQTT\_TX\_BUF\_LEN**

```
#define CONFIG_AWS_IOT_MQTT_TX_BUF_LEN 512
```

**5.29.1.29 CONFIG\_AWS\_IOT\_SDK**

```
#define CONFIG_AWS_IOT_SDK 1
```

**5.29.1.30 CONFIG\_AWS\_IOT\_SHADOW\_MAX\_JSON\_TOKEN\_EXPECTED**

```
#define CONFIG_AWS_IOT_SHADOW_MAX_JSON_TOKEN_EXPECTED 120
```

**5.29.1.31 CONFIG\_AWS\_IOT\_SHADOW\_MAX\_SHADOW\_TOPIC\_LENGTH\_WITHOUT\_THINGNAME**

```
#define CONFIG_AWS_IOT_SHADOW_MAX_SHADOW_TOPIC_LENGTH_WITHOUT_THINGNAME 60
```

**5.29.1.32 CONFIG\_AWS\_IOT\_SHADOW\_MAX\_SIMULTANEOUS\_ACKS**

```
#define CONFIG_AWS_IOT_SHADOW_MAX_SIMULTANEOUS_ACKS 10
```

**5.29.1.33 CONFIG\_AWS\_IOT\_SHADOW\_MAX\_SIMULTANEOUS\_THINGNAMES**

```
#define CONFIG_AWS_IOT_SHADOW_MAX_SIMULTANEOUS_THINGNAMES 10
```

**5.29.1.34 CONFIG\_AWS\_IOT\_SHADOW\_MAX\_SIZE\_OF\_THING\_NAME**

```
#define CONFIG_AWS_IOT_SHADOW_MAX_SIZE_OF_THING_NAME 20
```

**5.29.1.35 CONFIG\_AWS\_IOT\_SHADOW\_MAX\_SIZE\_OF\_UNIQUE\_CLIENT\_ID\_BYTES**

```
#define CONFIG_AWS_IOT_SHADOW_MAX_SIZE_OF_UNIQUE_CLIENT_ID_BYTES 80
```

**5.29.1.36 CONFIG\_BLE\_ADV\_REPORT\_DISCARD\_THRSHOLD**

```
#define CONFIG_BLE_ADV_REPORT_DISCARD_THRSHOLD 20
```

**5.29.1.37 CONFIG\_BLE\_ADV\_REPORT\_FLOW\_CONTROL\_NUM**

```
#define CONFIG_BLE_ADV_REPORT_FLOW_CONTROL_NUM 100
```

**5.29.1.38 CONFIG\_BLE\_ADV\_REPORT\_FLOW\_CONTROL\_SUPPORTED**

```
#define CONFIG_BLE_ADV_REPORT_FLOW_CONTROL_SUPPORTED 1
```

**5.29.1.39 CONFIG\_BLE\_ESTABLISH\_LINK\_CONNECTION\_TIMEOUT**

```
#define CONFIG_BLE_ESTABLISH_LINK_CONNECTION_TIMEOUT 30
```

**5.29.1.40 CONFIG\_BLE\_SCAN\_DUPLICATE**

```
#define CONFIG_BLE_SCAN_DUPLICATE 1
```

**5.29.1.41 CONFIG\_BLE\_SMP\_ENABLE**

```
#define CONFIG_BLE_SMP_ENABLE 1
```

**5.29.1.42 CONFIG\_BLUEDROID\_ENABLED**

```
#define CONFIG_BLUEDROID_ENABLED 1
```

**5.29.1.43 CONFIG\_BLUEDROID\_PINNED\_TO\_CORE**

```
#define CONFIG_BLUEDROID_PINNED_TO_CORE 0
```

**5.29.1.44 CONFIG\_BLUEDROID\_PINNED\_TO\_CORE\_0**

```
#define CONFIG_BLUEDROID_PINNED_TO_CORE_0 1
```

**5.29.1.45 CONFIG\_BLUFI\_INITIAL\_TRACE\_LEVEL**

```
#define CONFIG_BLUFI_INITIAL_TRACE_LEVEL 2
```

**5.29.1.46 CONFIG\_BLUFI\_TRACE\_LEVEL\_WARNING**

```
#define CONFIG_BLUFI_TRACE_LEVEL_WARNING 1
```

**5.29.1.47 CONFIG\_BNEP\_INITIAL\_TRACE\_LEVEL**

```
#define CONFIG_BNEP_INITIAL_TRACE_LEVEL 2
```

**5.29.1.48 CONFIG\_BNEP\_TRACE\_LEVEL\_WARNING**

```
#define CONFIG_BNEP_TRACE_LEVEL_WARNING 1
```

**5.29.1.49 CONFIG\_BOOTLOADER\_VDDSDIO\_BOOST\_1\_9V**

```
#define CONFIG_BOOTLOADER_VDDSDIO_BOOST_1_9V 1
```

**5.29.1.50 CONFIG\_BOOTLOADER\_WDT\_ENABLE**

```
#define CONFIG_BOOTLOADER_WDT_ENABLE 1
```

**5.29.1.51 CONFIG\_BOOTLOADER\_WDT\_TIME\_MS**

```
#define CONFIG_BOOTLOADER_WDT_TIME_MS 9000
```

**5.29.1.52 CONFIG\_BROWNOUT\_DET**

```
#define CONFIG_BROWNOUT_DET 1
```

**5.29.1.53 CONFIG\_BROWNOUT\_DET\_LVL**

```
#define CONFIG_BROWNOUT_DET_LVL 0
```

**5.29.1.54 CONFIG\_BROWNOUT\_DET\_LVL\_SEL\_0**

```
#define CONFIG_BROWNOUT_DET_LVL_SEL_0 1
```

**5.29.1.55 CONFIG\_BT\_ACL\_CONNECTIONS**

```
#define CONFIG_BT_ACL_CONNECTIONS 4
```

**5.29.1.56 CONFIG\_BT\_ENABLED**

```
#define CONFIG_BT_ENABLED 1
```

**5.29.1.57 CONFIG\_BT\_RESERVE\_DRAM**

```
#define CONFIG_BT_RESERVE_DRAM 0xdb5c
```

**5.29.1.58 CONFIG\_BTC\_INITIAL\_TRACE\_LEVEL**

```
#define CONFIG_BTC_INITIAL_TRACE_LEVEL 2
```

**5.29.1.59 CONFIG\_BTC\_TASK\_STACK\_SIZE**

```
#define CONFIG_BTC_TASK_STACK_SIZE 3072
```

**5.29.1.60 CONFIG\_BTC\_TRACE\_LEVEL\_WARNING**

```
#define CONFIG_BTC_TRACE_LEVEL_WARNING 1
```

**5.29.1.61 CONFIG\_BTDM\_CONTROLLER\_BLE\_MAX\_CONN**

```
#define CONFIG_BTDM_CONTROLLER_BLE_MAX_CONN 3
```

**5.29.1.62 CONFIG\_BTDM\_CONTROLLER\_BLE\_MAX\_CONN\_EFF**

```
#define CONFIG_BTDM_CONTROLLER_BLE_MAX_CONN_EFF 3
```

**5.29.1.63 CONFIG\_BTDM\_CONTROLLER\_BR\_EDR\_MAX\_ACL\_CONN\_EFF**

```
#define CONFIG_BTDM_CONTROLLER_BR_EDR_MAX_ACL_CONN_EFF 0
```

**5.29.1.64 CONFIG\_BTDM\_CONTROLLER\_BR\_EDR\_MAX\_SYNC\_CONN\_EFF**

```
#define CONFIG_BTDM_CONTROLLER_BR_EDR_MAX_SYNC_CONN_EFF 0
```

**5.29.1.65 CONFIG\_BTDM\_CONTROLLER\_HCI\_MODE\_VHCI**

```
#define CONFIG_BTDM_CONTROLLER_HCI_MODE_VHCI 1
```

**5.29.1.66 CONFIG\_BTDM\_CONTROLLER\_MODE\_BLE\_ONLY**

```
#define CONFIG_BTDM_CONTROLLER_MODE_BLE_ONLY 1
```

**5.29.1.67 CONFIG\_BTDM\_CONTROLLER\_PINNED\_TO\_CORE**

```
#define CONFIG_BTDM_CONTROLLER_PINNED_TO_CORE 0
```

**5.29.1.68 CONFIG\_BTDM\_CONTROLLER\_PINNED\_TO\_CORE\_0**

```
#define CONFIG_BTDM_CONTROLLER_PINNED_TO_CORE_0 1
```

**5.29.1.69 CONFIG\_BTIF\_INITIAL\_TRACE\_LEVEL**

```
#define CONFIG_BTIF_INITIAL_TRACE_LEVEL 2
```

**5.29.1.70 CONFIG\_BTIF\_TRACE\_LEVEL\_WARNING**

```
#define CONFIG_BTIF_TRACE_LEVEL_WARNING 1
```

**5.29.1.71 CONFIG\_BTM\_INITIAL\_TRACE\_LEVEL**

```
#define CONFIG_BTM_INITIAL_TRACE_LEVEL 2
```

**5.29.1.72 CONFIG\_BTM\_TRACE\_LEVEL\_WARNING**

```
#define CONFIG_BTM_TRACE_LEVEL_WARNING 1
```

**5.29.1.73 CONFIG\_BTU\_TASK\_STACK\_SIZE**

```
#define CONFIG_BTU_TASK_STACK_SIZE 4096
```

**5.29.1.74 CONFIG\_CONSOLE\_UART\_BAUDRATE**

```
#define CONFIG_CONSOLE_UART_BAUDRATE 115200
```

**5.29.1.75 CONFIG\_CONSOLE\_UART\_DEFAULT**

```
#define CONFIG_CONSOLE_UART_DEFAULT 1
```

**5.29.1.76 CONFIG\_CONSOLE\_UART\_NUM**

```
#define CONFIG_CONSOLE_UART_NUM 0
```

**5.29.1.77 CONFIG\_DMA\_RX\_BUF\_NUM**

```
#define CONFIG_DMA_RX_BUF_NUM 10
```

**5.29.1.78 CONFIG\_DMA\_TX\_BUF\_NUM**

```
#define CONFIG_DMA_TX_BUF_NUM 10
```

**5.29.1.79 CONFIG\_DUPLICATE\_SCAN\_CACHE\_SIZE**

```
#define CONFIG_DUPLICATE_SCAN_CACHE_SIZE 50
```

**5.29.1.80 CONFIG\_EFUSE\_CODE\_SCHEME\_COMPAT\_3\_4**

```
#define CONFIG_EFUSE_CODE_SCHEME_COMPAT_3_4 1
```

**5.29.1.81 CONFIG\_EFUSE\_MAX\_BLK\_LEN**

```
#define CONFIG_EFUSE_MAX_BLK_LEN 192
```

**5.29.1.82 CONFIG\_EMAC\_CHECK\_LINK\_PERIOD\_MS**

```
#define CONFIG_EMAC_CHECK_LINK_PERIOD_MS 2000
```

**5.29.1.83 CONFIG\_EMAC\_TASK\_PRIORITY**

```
#define CONFIG_EMAC_TASK_PRIORITY 20
```

**5.29.1.84 CONFIG\_EMAC\_TASK\_STACK\_SIZE**

```
#define CONFIG_EMAC_TASK_STACK_SIZE 3072
```

**5.29.1.85 CONFIG\_ENABLE\_ARDUINO\_DEPENDS**

```
#define CONFIG_ENABLE_ARDUINO_DEPENDS 1
```

**5.29.1.86 CONFIG\_ESP32\_APPTRACE\_DEST\_NONE**

```
#define CONFIG_ESP32_APPTRACE_DEST_NONE 1
```

**5.29.1.87 CONFIG\_ESP32\_APPTRACE\_LOCK\_ENABLE**

```
#define CONFIG_ESP32_APPTRACE_LOCK_ENABLE 1
```

**5.29.1.88 CONFIG\_ESP32\_DEBUG\_OCDAWARE**

```
#define CONFIG_ESP32_DEBUG_OCDAWARE 1
```

**5.29.1.89 CONFIG\_ESP32\_DEBUG\_STUBS\_ENABLE**

```
#define CONFIG_ESP32_DEBUG_STUBS_ENABLE 1
```



**5.29.1.90 CONFIG\_ESP32\_DEEP\_SLEEP\_WAKEUP\_DELAY**

```
#define CONFIG_ESP32_DEEP_SLEEP_WAKEUP_DELAY 2000
```

**5.29.1.91 CONFIG\_ESP32\_DEFAULT\_CPU\_FREQ\_160**

```
#define CONFIG_ESP32_DEFAULT_CPU_FREQ_160 1
```

**5.29.1.92 CONFIG\_ESP32\_DEFAULT\_CPU\_FREQ\_MHZ**

```
#define CONFIG_ESP32_DEFAULT_CPU_FREQ_MHZ 160
```

**5.29.1.93 CONFIG\_ESP32\_DEFAULT\_PTHREAD\_CORE\_NO\_AFFINITY**

```
#define CONFIG_ESP32_DEFAULT_PTHREAD_CORE_NO_AFFINITY 1
```

**5.29.1.94 CONFIG\_ESP32\_DPORT\_WORKAROUND**

```
#define CONFIG_ESP32_DPORT_WORKAROUND 1
```

**5.29.1.95 CONFIG\_ESP32\_ENABLE\_COREDUMP\_TO\_NONE**

```
#define CONFIG_ESP32_ENABLE_COREDUMP_TO_NONE 1
```

**5.29.1.96 CONFIG\_ESP32\_PANIC\_PRINT\_REBOOT**

```
#define CONFIG_ESP32_PANIC_PRINT_REBOOT 1
```

**5.29.1.97 CONFIG\_ESP32\_PHY\_CALIBRATION\_AND\_DATA\_STORAGE**

```
#define CONFIG_ESP32_PHY_CALIBRATION_AND_DATA_STORAGE 1
```

**5.29.1.98 CONFIG\_ESP32\_PHY\_MAX\_TX\_POWER**

```
#define CONFIG_ESP32_PHY_MAX_TX_POWER 20
```

**5.29.1.99 CONFIG\_ESP32\_PHY\_MAX\_WIFI\_TX\_POWER**

```
#define CONFIG_ESP32_PHY_MAX_WIFI_TX_POWER 20
```

**5.29.1.100 CONFIG\_ESP32\_PTHREAD\_TASK\_CORE\_DEFAULT**

```
#define CONFIG_ESP32_PTHREAD_TASK_CORE_DEFAULT -1
```

**5.29.1.101 CONFIG\_ESP32\_PTHREAD\_TASK\_NAME\_DEFAULT**

```
#define CONFIG_ESP32_PTHREAD_TASK_NAME_DEFAULT "pthread"
```

**5.29.1.102 CONFIG\_ESP32\_PTHREAD\_TASK\_PRIO\_DEFAULT**

```
#define CONFIG_ESP32_PTHREAD_TASK_PRIO_DEFAULT 5
```

**5.29.1.103 CONFIG\_ESP32\_PTHREAD\_TASK\_STACK\_SIZE\_DEFAULT**

```
#define CONFIG_ESP32_PTHREAD_TASK_STACK_SIZE_DEFAULT 3072
```

**5.29.1.104 CONFIG\_ESP32\_REV\_MIN**

```
#define CONFIG_ESP32_REV_MIN 0
```

**5.29.1.105 CONFIG\_ESP32\_REV\_MIN\_0**

```
#define CONFIG_ESP32_REV_MIN_0 1
```

**5.29.1.106 CONFIG\_ESP32\_RTC\_CLK\_CAL\_CYCLES**

```
#define CONFIG_ESP32_RTC_CLK_CAL_CYCLES 1024
```

**5.29.1.107 CONFIG\_ESP32\_RTC\_CLOCK\_SOURCE\_INTERNAL\_RC**

```
#define CONFIG_ESP32_RTC_CLOCK_SOURCE_INTERNAL_RC 1
```

**5.29.1.108 CONFIG\_ESP32\_TIME\_SYSCALL\_USE\_RTC\_FRC1**

```
#define CONFIG_ESP32_TIME_SYSCALL_USE_RTC_FRC1 1
```

**5.29.1.109 CONFIG\_ESP32\_WIFI\_AMPDU\_RX\_ENABLED**

```
#define CONFIG_ESP32_WIFI_AMPDU_RX_ENABLED 1
```

**5.29.1.110 CONFIG\_ESP32\_WIFI\_AMPDU\_TX\_ENABLED**

```
#define CONFIG_ESP32_WIFI_AMPDU_TX_ENABLED 1
```

**5.29.1.111 CONFIG\_ESP32\_WIFI\_DYNAMIC\_RX\_BUFFER\_NUM**

```
#define CONFIG_ESP32_WIFI_DYNAMIC_RX_BUFFER_NUM 32
```

**5.29.1.112 CONFIG\_ESP32\_WIFI\_DYNAMIC\_TX\_BUFFER**

```
#define CONFIG_ESP32_WIFI_DYNAMIC_TX_BUFFER 1
```

**5.29.1.113 CONFIG\_ESP32\_WIFI\_DYNAMIC\_TX\_BUFFER\_NUM**

```
#define CONFIG_ESP32_WIFI_DYNAMIC_TX_BUFFER_NUM 32
```

**5.29.1.114 CONFIG\_ESP32\_WIFI\_IRAM\_OPT**

```
#define CONFIG_ESP32_WIFI_IRAM_OPT 1
```

**5.29.1.115 CONFIG\_ESP32\_WIFI\_MGMT\_SBUF\_NUM**

```
#define CONFIG_ESP32_WIFI_MGMT_SBUF_NUM 32
```

**5.29.1.116 CONFIG\_ESP32\_WIFI\_NVS\_ENABLED**

```
#define CONFIG_ESP32_WIFI_NVS_ENABLED 1
```

**5.29.1.117 CONFIG\_ESP32\_WIFI\_RX\_BA\_WIN**

```
#define CONFIG_ESP32_WIFI_RX_BA_WIN 6
```

**5.29.1.118 CONFIG\_ESP32\_WIFI\_SOFTAP\_BEACON\_MAX\_LEN**

```
#define CONFIG_ESP32_WIFI_SOFTAP_BEACON_MAX_LEN 752
```

**5.29.1.119 CONFIG\_ESP32\_WIFI\_STATIC\_RX\_BUFFER\_NUM**

```
#define CONFIG_ESP32_WIFI_STATIC_RX_BUFFER_NUM 10
```

**5.29.1.120 CONFIG\_ESP32\_WIFI\_TASK\_PINNED\_TO\_CORE\_0**

```
#define CONFIG_ESP32_WIFI_TASK_PINNED_TO_CORE_0 1
```

**5.29.1.121 CONFIG\_ESP32\_WIFI\_TX\_BA\_WIN**

```
#define CONFIG_ESP32_WIFI_TX_BA_WIN 6
```

**5.29.1.122 CONFIG\_ESP32\_WIFI\_TX\_BUFFER\_TYPE**

```
#define CONFIG_ESP32_WIFI_TX_BUFFER_TYPE 1
```

**5.29.1.123 CONFIG\_ESP32\_XTAL\_FREQ**

```
#define CONFIG_ESP32_XTAL_FREQ 40
```

**5.29.1.124 CONFIG\_ESP32\_XTAL\_FREQ\_40**

```
#define CONFIG_ESP32_XTAL_FREQ_40 1
```

**5.29.1.125 CONFIG\_ESP\_ERR\_TO\_NAME\_LOOKUP**

```
#define CONFIG_ESP_ERR_TO_NAME_LOOKUP 1
```

**5.29.1.126 CONFIG\_ESP\_GRATUITOUS\_ARP**

```
#define CONFIG_ESP_GRATUITOUS_ARP 1
```

**5.29.1.127 CONFIG\_ESP\_HTTP\_CLIENT\_ENABLE\_HTTPS**

```
#define CONFIG_ESP_HTTP_CLIENT_ENABLE_HTTPS 1
```

**5.29.1.128 CONFIG\_ESPTOOLPY\_AFTER**

```
#define CONFIG_ESPTOOLPY_AFTER "hard_reset"
```

**5.29.1.129 CONFIG\_ESPTOOLPY\_AFTER\_RESET**

```
#define CONFIG_ESPTOOLPY_AFTER_RESET 1
```

**5.29.1.130 CONFIG\_ESPTOOLPY\_BAUD**

```
#define CONFIG_ESPTOOLPY_BAUD 115200
```

**5.29.1.131 CONFIG\_ESPTOOLPY\_BAUD\_115200B**

```
#define CONFIG_ESPTOOLPY_BAUD_115200B 1
```

**5.29.1.132 CONFIG\_ESPTOOLPY\_BAUD\_OTHER\_VAL**

```
#define CONFIG_ESPTOOLPY_BAUD_OTHER_VAL 115200
```

**5.29.1.133 CONFIG\_ESPTOOLPY\_BEFORE**

```
#define CONFIG_ESPTOOLPY_BEFORE "default_reset"
```

**5.29.1.134 CONFIG\_ESPTOOLPY\_BEFORE\_RESET**

```
#define CONFIG_ESPTOOLPY_BEFORE_RESET 1
```

**5.29.1.135 CONFIG\_ESPTOOLPY\_COMPRESSED**

```
#define CONFIG_ESPTOOLPY_COMPRESSED 1
```

**5.29.1.136 CONFIG\_ESPTOOLPY\_FLASHFREQ**

```
#define CONFIG_ESPTOOLPY_FLASHFREQ "40m"
```

**5.29.1.137 CONFIG\_ESPTOOLPY\_FLASHFREQ\_40M**

```
#define CONFIG_ESPTOOLPY_FLASHFREQ_40M 1
```

**5.29.1.138 CONFIG\_ESPTOOLPY\_FLASHMODE**

```
#define CONFIG_ESPTOOLPY_FLASHMODE "dio"
```

**5.29.1.139 CONFIG\_ESPTOOLPY\_FLASHSIZE**

```
#define CONFIG_ESPTOOLPY_FLASHSIZE "2MB"
```

**5.29.1.140 CONFIG\_ESPTOOLPY\_FLASHSIZE\_2MB**

```
#define CONFIG_ESPTOOLPY_FLASHSIZE_2MB 1
```

**5.29.1.141 CONFIG\_ESPTOOLPY\_FLASHSIZE\_DETECT**

```
#define CONFIG_ESPTOOLPY_FLASHSIZE_DETECT 1
```

**5.29.1.142 CONFIG\_ESPTOOLPY\_PORT**

```
#define CONFIG_ESPTOOLPY_PORT "COM19"
```

**5.29.1.143 CONFIG\_FATFS\_CODEPAGE**

```
#define CONFIG_FATFS_CODEPAGE 437
```

**5.29.1.144 CONFIG\_FATFS\_CODEPAGE\_437**

```
#define CONFIG_FATFS_CODEPAGE_437 1
```

**5.29.1.145 CONFIG\_FATFS\_FS\_LOCK**

```
#define CONFIG_FATFS_FS_LOCK 0
```

**5.29.1.146 CONFIG\_FATFS\_LFN\_NONE**

```
#define CONFIG_FATFS_LFN_NONE 1
```

**5.29.1.147 CONFIG\_FATFS\_PER\_FILE\_CACHE**

```
#define CONFIG_FATFS_PER_FILE_CACHE 1
```

**5.29.1.148 CONFIG\_FATFS\_TIMEOUT\_MS**

```
#define CONFIG_FATFS_TIMEOUT_MS 10000
```

**5.29.1.149 CONFIG\_FLASHMODE\_DIO**

```
#define CONFIG_FLASHMODE_DIO 1
```

**5.29.1.150 CONFIG\_FOUR\_UNIVERSAL\_MAC\_ADDRESS**

```
#define CONFIG_FOUR_UNIVERSAL_MAC_ADDRESS 1
```

**5.29.1.151 CONFIG\_FREERTOS\_ASSERT\_FAIL\_ABORT**

```
#define CONFIG_FREERTOS_ASSERT_FAIL_ABORT 1
```

**5.29.1.152 CONFIG\_FREERTOS\_ASSERT\_ON\_UNTESTED\_FUNCTION**

```
#define CONFIG_FREERTOS_ASSERT_ON_UNTESTED_FUNCTION 1
```

**5.29.1.153 CONFIG\_FREERTOS\_CHECK\_MUTEX\_GIVEN\_BY\_OWNER**

```
#define CONFIG_FREERTOS_CHECK_MUTEX_GIVEN_BY_OWNER 1
```



**5.29.1.154 CONFIG\_FREERTOS\_CHECK\_STACKOVERFLOW\_CANARY**

```
#define CONFIG_FREERTOS_CHECK_STACKOVERFLOW_CANARY 1
```

**5.29.1.155 CONFIG\_FREERTOS\_CORETIMER\_0**

```
#define CONFIG_FREERTOS_CORETIMER_0 1
```

**5.29.1.156 CONFIG\_FREERTOS\_HZ**

```
#define CONFIG_FREERTOS_HZ 100
```

**5.29.1.157 CONFIG\_FREERTOS\_IDLE\_TASK\_STACKSIZE**

```
#define CONFIG_FREERTOS_IDLE_TASK_STACKSIZE 1536
```

**5.29.1.158 CONFIG\_FREERTOS\_INTERRUPT\_BACKTRACE**

```
#define CONFIG_FREERTOS_INTERRUPT_BACKTRACE 1
```

**5.29.1.159 CONFIG\_FREERTOS\_ISR\_STACKSIZE**

```
#define CONFIG_FREERTOS_ISR_STACKSIZE 1536
```

**5.29.1.160 CONFIG\_FREERTOS\_MAX\_TASK\_NAME\_LEN**

```
#define CONFIG_FREERTOS_MAX_TASK_NAME_LEN 16
```

**5.29.1.161 CONFIG\_FREERTOS\_NO\_AFFINITY**

```
#define CONFIG_FREERTOS_NO_AFFINITY 0x7FFFFFFF
```

**5.29.1.162 CONFIG\_FREERTOS\_QUEUE\_REGISTRY\_SIZE**

```
#define CONFIG_FREERTOS_QUEUE_REGISTRY_SIZE 0
```

**5.29.1.163 CONFIG\_FREERTOS\_TASK\_FUNCTION\_WRAPPER**

```
#define CONFIG_FREERTOS_TASK_FUNCTION_WRAPPER 1
```

**5.29.1.164 CONFIG\_FREERTOS\_THREAD\_LOCAL\_STORAGE\_POINTERS**

```
#define CONFIG_FREERTOS_THREAD_LOCAL_STORAGE_POINTERS 1
```

**5.29.1.165 CONFIG\_GAP\_INITIAL\_TRACE\_LEVEL**

```
#define CONFIG_GAP_INITIAL_TRACE_LEVEL 2
```

**5.29.1.166 CONFIG\_GAP\_TRACE\_LEVEL\_WARNING**

```
#define CONFIG_GAP_TRACE_LEVEL_WARNING 1
```

**5.29.1.167 CONFIG\_GARP\_TMR\_INTERVAL**

```
#define CONFIG_GARP_TMR_INTERVAL 60
```

**5.29.1.168 CONFIG\_GATT\_INITIAL\_TRACE\_LEVEL**

```
#define CONFIG_GATT_INITIAL_TRACE_LEVEL 2
```

**5.29.1.169 CONFIG\_GATT\_TRACE\_LEVEL\_WARNING**

```
#define CONFIG_GATT_TRACE_LEVEL_WARNING 1
```

**5.29.1.170 CONFIG\_GATTC\_ENABLE**

```
#define CONFIG_GATTC_ENABLE 1
```

**5.29.1.171 CONFIG\_GATTS\_ENABLE**

```
#define CONFIG_GATTS_ENABLE 1
```

**5.29.1.172 CONFIG\_GATTS\_SEND\_SERVICE\_CHANGE\_AUTO**

```
#define CONFIG_GATTS_SEND_SERVICE_CHANGE_AUTO 1
```

**5.29.1.173 CONFIG\_GATTS\_SEND\_SERVICE\_CHANGE\_MODE**

```
#define CONFIG_GATTS_SEND_SERVICE_CHANGE_MODE 0
```

**5.29.1.174 CONFIG\_HCI\_INITIAL\_TRACE\_LEVEL**

```
#define CONFIG_HCI_INITIAL_TRACE_LEVEL 2
```

**5.29.1.175 CONFIG\_HCI\_TRACE\_LEVEL\_WARNING**

```
#define CONFIG_HCI_TRACE_LEVEL_WARNING 1
```

**5.29.1.176 CONFIG\_HEAP\_POISONING\_DISABLED**

```
#define CONFIG_HEAP_POISONING_DISABLED 1
```

**5.29.1.177 CONFIG\_HID\_INITIAL\_TRACE\_LEVEL**

```
#define CONFIG_HID_INITIAL_TRACE_LEVEL 2
```

**5.29.1.178 CONFIG\_HID\_TRACE\_LEVEL\_WARNING**

```
#define CONFIG_HID_TRACE_LEVEL_WARNING 1
```

**5.29.1.179 CONFIG\_HTTPD\_ERR\_RESP\_NO\_DELAY**

```
#define CONFIG_HTTPD_ERR_RESP_NO_DELAY 1
```

**5.29.1.180 CONFIG\_HTTPD\_MAX\_REQ\_HDR\_LEN**

```
#define CONFIG_HTTPD_MAX_REQ_HDR_LEN 512
```

**5.29.1.181 CONFIG\_HTTPD\_MAX\_URI\_LEN**

```
#define CONFIG_HTTPD_MAX_URI_LEN 512
```

**5.29.1.182 CONFIG\_HTTPD\_PURGE\_BUF\_LEN**

```
#define CONFIG_HTTPD_PURGE_BUF_LEN 32
```

**5.29.1.183 CONFIG\_IDF\_TARGET**

```
#define CONFIG_IDF_TARGET "esp32"
```

**5.29.1.184 CONFIG\_IDF\_TARGET\_ESP32**

```
#define CONFIG_IDF_TARGET_ESP32 1
```

**5.29.1.185 CONFIG\_INT\_WDT**

```
#define CONFIG_INT_WDT 1
```

**5.29.1.186 CONFIG\_INT\_WDT\_CHECK\_CPU1**

```
#define CONFIG_INT_WDT_CHECK_CPU1 1
```

**5.29.1.187 CONFIG\_INT\_WDT\_TIMEOUT\_MS**

```
#define CONFIG_INT_WDT_TIMEOUT_MS 300
```

**5.29.1.188 CONFIG\_IP\_LOST\_TIMER\_INTERVAL**

```
#define CONFIG_IP_LOST_TIMER_INTERVAL 120
```

**5.29.1.189 CONFIG\_IPC\_TASK\_STACK\_SIZE**

```
#define CONFIG_IPC_TASK_STACK_SIZE 1024
```

**5.29.1.190 CONFIG\_L2CAP\_INITIAL\_TRACE\_LEVEL**

```
#define CONFIG_L2CAP_INITIAL_TRACE_LEVEL 2
```

**5.29.1.191 CONFIG\_L2CAP\_TRACE\_LEVEL\_WARNING**

```
#define CONFIG_L2CAP_TRACE_LEVEL_WARNING 1
```

**5.29.1.192 CONFIG\_LIBSODIUM\_USE\_MBEDTLS\_SHA**

```
#define CONFIG_LIBSODIUM_USE_MBEDTLS_SHA 1
```

**5.29.1.193 CONFIG\_LOG\_BOOTLOADER\_LEVEL**

```
#define CONFIG_LOG_BOOTLOADER_LEVEL 3
```

**5.29.1.194 CONFIG\_LOG\_BOOTLOADER\_LEVEL\_INFO**

```
#define CONFIG_LOG_BOOTLOADER_LEVEL_INFO 1
```

**5.29.1.195 CONFIG\_LOG\_COLORS**

```
#define CONFIG_LOG_COLORS 1
```

**5.29.1.196 CONFIG\_LOG\_DEFAULT\_LEVEL**

```
#define CONFIG_LOG_DEFAULT_LEVEL 3
```

**5.29.1.197 CONFIG\_LOG\_DEFAULT\_LEVEL\_INFO**

```
#define CONFIG_LOG_DEFAULT_LEVEL_INFO 1
```

**5.29.1.198 CONFIG\_LWIP\_DHCP\_DOES\_ARP\_CHECK**

```
#define CONFIG_LWIP_DHCP_DOES_ARP_CHECK 1
```

**5.29.1.199 CONFIG\_LWIP\_DHCP\_MAX\_NTP\_SERVERS**

```
#define CONFIG_LWIP_DHCP_MAX_NTP_SERVERS 1
```

**5.29.1.200 CONFIG\_LWIP\_DHCPS\_LEASE\_UNIT**

```
#define CONFIG_LWIP_DHCPS_LEASE_UNIT 60
```

**5.29.1.201 CONFIG\_LWIP\_DHCPS\_MAX\_STATION\_NUM**

```
#define CONFIG_LWIP_DHCPS_MAX_STATION_NUM 8
```

**5.29.1.202 CONFIG\_LWIP\_LOOPBACK\_MAX\_PBUFS**

```
#define CONFIG_LWIP_LOOPBACK_MAX_PBUFS 8
```

**5.29.1.203 CONFIG\_LWIP\_MAX\_ACTIVE\_TCP**

```
#define CONFIG_LWIP_MAX_ACTIVE_TCP 16
```

**5.29.1.204 CONFIG\_LWIP\_MAX\_LISTENING\_TCP**

```
#define CONFIG_LWIP_MAX_LISTENING_TCP 16
```

**5.29.1.205 CONFIG\_LWIP\_MAX\_RAW\_PCBS**

```
#define CONFIG_LWIP_MAX_RAW_PCBS 16
```

**5.29.1.206 CONFIG\_LWIP\_MAX\_SOCKETS**

```
#define CONFIG_LWIP_MAX_SOCKETS 10
```

**5.29.1.207 CONFIG\_LWIP\_MAX\_UDP\_PCBS**

```
#define CONFIG_LWIP_MAX_UDP_PCBS 16
```

**5.29.1.208 CONFIG\_LWIP\_NETIF\_LOOPBACK**

```
#define CONFIG_LWIP_NETIF_LOOPBACK 1
```

**5.29.1.209 CONFIG\_LWIP\_SO\_REUSE**

```
#define CONFIG_LWIP_SO_REUSE 1
```

**5.29.1.210 CONFIG\_LWIP\_SO\_REUSE\_RXTOALL**

```
#define CONFIG_LWIP_SO_REUSE_RXTOALL 1
```

**5.29.1.211 CONFIG\_MAIN\_TASK\_STACK\_SIZE**

```
#define CONFIG_MAIN_TASK_STACK_SIZE 3584
```

**5.29.1.212 CONFIG\_MAKE\_WARN\_UNDEFINED\_VARIABLES**

```
#define CONFIG_MAKE_WARN_UNDEFINED_VARIABLES 1
```

**5.29.1.213 CONFIG\_MB\_CONTROLLER\_NOTIFY\_QUEUE\_SIZE**

```
#define CONFIG_MB_CONTROLLER_NOTIFY_QUEUE_SIZE 20
```

**5.29.1.214 CONFIG\_MB\_CONTROLLER\_NOTIFY\_TIMEOUT**

```
#define CONFIG_MB_CONTROLLER_NOTIFY_TIMEOUT 20
```

**5.29.1.215 CONFIG\_MB\_CONTROLLER\_STACK\_SIZE**

```
#define CONFIG_MB_CONTROLLER_STACK_SIZE 4096
```

**5.29.1.216 CONFIG\_MB\_EVENT\_QUEUE\_TIMEOUT**

```
#define CONFIG_MB_EVENT_QUEUE_TIMEOUT 20
```

**5.29.1.217 CONFIG\_MB\_QUEUE\_LENGTH**

```
#define CONFIG_MB_QUEUE_LENGTH 20
```



**5.29.1.218 CONFIG\_MB\_SERIAL\_BUF\_SIZE**

```
#define CONFIG_MB_SERIAL_BUF_SIZE 256
```

**5.29.1.219 CONFIG\_MB\_SERIAL\_TASK\_PRIO**

```
#define CONFIG_MB_SERIAL_TASK_PRIO 10
```

**5.29.1.220 CONFIG\_MB\_SERIAL\_TASK\_STACK\_SIZE**

```
#define CONFIG_MB_SERIAL_TASK_STACK_SIZE 2048
```

**5.29.1.221 CONFIG\_MB\_TIMER\_GROUP**

```
#define CONFIG_MB_TIMER_GROUP 0
```

**5.29.1.222 CONFIG\_MB\_TIMER\_INDEX**

```
#define CONFIG_MB_TIMER_INDEX 0
```

**5.29.1.223 CONFIG\_MB\_TIMER\_PORT\_ENABLED**

```
#define CONFIG_MB_TIMER_PORT_ENABLED 1
```

**5.29.1.224 CONFIG\_MBEDTLS\_AES\_C**

```
#define CONFIG_MBEDTLS_AES_C 1
```

**5.29.1.225 CONFIG\_MBEDTLS\_CCM\_C**

```
#define CONFIG_MBEDTLS_CCM_C 1
```

**5.29.1.226 CONFIG\_MBEDTLS\_ECDH\_C**

```
#define CONFIG_MBEDTLS_ECDH_C 1
```

**5.29.1.227 CONFIG\_MBEDTLS\_ECDSA\_C**

```
#define CONFIG_MBEDTLS_ECDSA_C 1
```

**5.29.1.228 CONFIG\_MBEDTLS\_ECP\_C**

```
#define CONFIG_MBEDTLS_ECP_C 1
```

**5.29.1.229 CONFIG\_MBEDTLS\_ECP\_DP\_BP256R1\_ENABLED**

```
#define CONFIG_MBEDTLS_ECP_DP_BP256R1_ENABLED 1
```

**5.29.1.230 CONFIG\_MBEDTLS\_ECP\_DP\_BP384R1\_ENABLED**

```
#define CONFIG_MBEDTLS_ECP_DP_BP384R1_ENABLED 1
```

**5.29.1.231 CONFIG\_MBEDTLS\_ECP\_DP\_BP512R1\_ENABLED**

```
#define CONFIG_MBEDTLS_ECP_DP_BP512R1_ENABLED 1
```

**5.29.1.232 CONFIG\_MBEDTLS\_ECP\_DP\_CURVE25519\_ENABLED**

```
#define CONFIG_MBEDTLS_ECP_DP_CURVE25519_ENABLED 1
```

**5.29.1.233 CONFIG\_MBEDTLS\_ECP\_DP\_SECP192K1\_ENABLED**

```
#define CONFIG_MBEDTLS_ECP_DP_SECP192K1_ENABLED 1
```

**5.29.1.234 CONFIG\_MBEDTLS\_ECP\_DP\_SECP192R1\_ENABLED**

```
#define CONFIG_MBEDTLS_ECP_DP_SECP192R1_ENABLED 1
```

**5.29.1.235 CONFIG\_MBEDTLS\_ECP\_DP\_SECP224K1\_ENABLED**

```
#define CONFIG_MBEDTLS_ECP_DP_SECP224K1_ENABLED 1
```

**5.29.1.236 CONFIG\_MBEDTLS\_ECP\_DP\_SECP224R1\_ENABLED**

```
#define CONFIG_MBEDTLS_ECP_DP_SECP224R1_ENABLED 1
```

**5.29.1.237 CONFIG\_MBEDTLS\_ECP\_DP\_SECP256K1\_ENABLED**

```
#define CONFIG_MBEDTLS_ECP_DP_SECP256K1_ENABLED 1
```

**5.29.1.238 CONFIG\_MBEDTLS\_ECP\_DP\_SECP256R1\_ENABLED**

```
#define CONFIG_MBEDTLS_ECP_DP_SECP256R1_ENABLED 1
```

**5.29.1.239 CONFIG\_MBEDTLS\_ECP\_DP\_SECP384R1\_ENABLED**

```
#define CONFIG_MBEDTLS_ECP_DP_SECP384R1_ENABLED 1
```

**5.29.1.240 CONFIG\_MBEDTLS\_ECP\_DP\_SECP521R1\_ENABLED**

```
#define CONFIG_MBEDTLS_ECP_DP_SECP521R1_ENABLED 1
```

**5.29.1.241 CONFIG\_MBEDTLS\_ECP\_NIST\_OPTIM**

```
#define CONFIG_MBEDTLS_ECP_NIST_OPTIM 1
```

**5.29.1.242 CONFIG\_MBEDTLS\_GCM\_C**

```
#define CONFIG_MBEDTLS_GCM_C 1
```

**5.29.1.243 CONFIG\_MBEDTLS\_HARDWARE\_AES**

```
#define CONFIG_MBEDTLS_HARDWARE_AES 1
```

**5.29.1.244 CONFIG\_MBEDTLS\_HAVE\_TIME**

```
#define CONFIG_MBEDTLS_HAVE_TIME 1
```

**5.29.1.245 CONFIG\_MBEDTLS\_INTERNAL\_MEM\_ALLOC**

```
#define CONFIG_MBEDTLS_INTERNAL_MEM_ALLOC 1
```

**5.29.1.246 CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_DHE\_RSA**

```
#define CONFIG_MBEDTLS_KEY_EXCHANGE_DHE_RSA 1
```

**5.29.1.247 CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_ECDH\_ECDSA**

```
#define CONFIG_MBEDTLS_KEY_EXCHANGE_ECDH_ECDSA 1
```

**5.29.1.248 CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_ECDH\_RSA**

```
#define CONFIG_MBEDTLS_KEY_EXCHANGE_ECDH_RSA 1
```

**5.29.1.249 CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_ECDHE\_ECDSA**

```
#define CONFIG_MBEDTLS_KEY_EXCHANGE_ECDHE_ECDSA 1
```

**5.29.1.250 CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_ECDHE\_RSA**

```
#define CONFIG_MBEDTLS_KEY_EXCHANGE_ECDHE_RSA 1
```

**5.29.1.251 CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_ELLIPTIC\_CURVE**

```
#define CONFIG_MBEDTLS_KEY_EXCHANGE_ELLIPTIC_CURVE 1
```

**5.29.1.252 CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_RSA**

```
#define CONFIG_MBEDTLS_KEY_EXCHANGE_RSA 1
```

**5.29.1.253 CONFIG\_MBEDTLS\_PEM\_PARSE\_C**

```
#define CONFIG_MBEDTLS_PEM_PARSE_C 1
```

**5.29.1.254 CONFIG\_MBEDTLS\_PEM\_WRITE\_C**

```
#define CONFIG_MBEDTLS_PEM_WRITE_C 1
```

**5.29.1.255 CONFIG\_MBEDTLS\_RC4\_DISABLED**

```
#define CONFIG_MBEDTLS_RC4_DISABLED 1
```

**5.29.1.256 CONFIG\_MBEDTLS\_SSL\_ALPN**

```
#define CONFIG_MBEDTLS_SSL_ALPN 1
```

**5.29.1.257 CONFIG\_MBEDTLS\_SSL\_MAX\_CONTENT\_LEN**

```
#define CONFIG_MBEDTLS_SSL_MAX_CONTENT_LEN 16384
```

**5.29.1.258 CONFIG\_MBEDTLS\_SSL\_PROTO\_TLS1**

```
#define CONFIG_MBEDTLS_SSL_PROTO_TLS1 1
```

**5.29.1.259 CONFIG\_MBEDTLS\_SSL\_PROTO\_TLS1\_1**

```
#define CONFIG_MBEDTLS_SSL_PROTO_TLS1_1 1
```

**5.29.1.260 CONFIG\_MBEDTLS\_SSL\_PROTO\_TLS1\_2**

```
#define CONFIG_MBEDTLS_SSL_PROTO_TLS1_2 1
```

**5.29.1.261 CONFIG\_MBEDTLS\_SSL\_RENEGOTIATION**

```
#define CONFIG_MBEDTLS_SSL_RENEGOTIATION 1
```

**5.29.1.262 CONFIG\_MBEDTLS\_SSL\_SESSION\_TICKETS**

```
#define CONFIG_MBEDTLS_SSL_SESSION_TICKETS 1
```

**5.29.1.263 CONFIG\_MBEDTLS\_TLS\_CLIENT**

```
#define CONFIG_MBEDTLS_TLS_CLIENT 1
```

**5.29.1.264 CONFIG\_MBEDTLS\_TLS\_ENABLED**

```
#define CONFIG_MBEDTLS_TLS_ENABLED 1
```

**5.29.1.265 CONFIG\_MBEDTLS\_TLS\_SERVER**

```
#define CONFIG_MBEDTLS_TLS_SERVER 1
```

**5.29.1.266 CONFIG\_MBEDTLS\_TLS\_SERVER\_AND\_CLIENT**

```
#define CONFIG_MBEDTLS_TLS_SERVER_AND_CLIENT 1
```

**5.29.1.267 CONFIG\_MBEDTLS\_X509\_CRL\_PARSE\_C**

```
#define CONFIG_MBEDTLS_X509_CRL_PARSE_C 1
```

**5.29.1.268 CONFIG\_MBEDTLS\_X509\_CSR\_PARSE\_C**

```
#define CONFIG_MBEDTLS_X509_CSR_PARSE_C 1
```

**5.29.1.269 CONFIG\_MCA\_INITIAL\_TRACE\_LEVEL**

```
#define CONFIG_MCA_INITIAL_TRACE_LEVEL 2
```

**5.29.1.270 CONFIG\_MCA\_TRACE\_LEVEL\_WARNING**

```
#define CONFIG_MCA_TRACE_LEVEL_WARNING 1
```

**5.29.1.271 CONFIG\_MDNS\_MAX\_SERVICES**

```
#define CONFIG_MDNS_MAX_SERVICES 10
```

**5.29.1.272 CONFIG\_MONITOR\_BAUD**

```
#define CONFIG_MONITOR_BAUD 115200
```

**5.29.1.273 CONFIG\_MONITOR\_BAUD\_115200B**

```
#define CONFIG_MONITOR_BAUD_115200B 1
```

**5.29.1.274 CONFIG\_MONITOR\_BAUD\_OTHER\_VAL**

```
#define CONFIG_MONITOR_BAUD_OTHER_VAL 115200
```

**5.29.1.275 CONFIG\_MQTT\_PROTOCOL\_311**

```
#define CONFIG_MQTT_PROTOCOL_311 1
```

**5.29.1.276 CONFIG\_MQTT\_TRANSPORT\_SSL**

```
#define CONFIG_MQTT_TRANSPORT_SSL 1
```

**5.29.1.277 CONFIG\_MQTT\_TRANSPORT\_WEBSOCKET**

```
#define CONFIG_MQTT_TRANSPORT_WEBSOCKET 1
```

**5.29.1.278 CONFIG\_MQTT\_TRANSPORT\_WEBSOCKET\_SECURE**

```
#define CONFIG_MQTT_TRANSPORT_WEBSOCKET_SECURE 1
```

**5.29.1.279 CONFIG\_NEWLIB\_STDIN\_LINE\_ENDING\_CR**

```
#define CONFIG_NEWLIB_STDIN_LINE_ENDING_CR 1
```

**5.29.1.280 CONFIG\_NEWLIB\_STDOUT\_LINE\_ENDING\_CRLF**

```
#define CONFIG_NEWLIB_STDOUT_LINE_ENDING_CRLF 1
```

**5.29.1.281 CONFIG\_NUMBER\_OF\_UNIVERSAL\_MAC\_ADDRESS**

```
#define CONFIG_NUMBER_OF_UNIVERSAL_MAC_ADDRESS 4
```



**5.29.1.282 CONFIG\_OPENSSL\_ASSERT\_DO\_NOTHING**

```
#define CONFIG_OPENSSL_ASSERT_DO_NOTHING 1
```

**5.29.1.283 CONFIG\_OPTIMIZATION\_ASSERTIONS\_ENABLED**

```
#define CONFIG_OPTIMIZATION_ASSERTIONS_ENABLED 1
```

**5.29.1.284 CONFIG\_OPTIMIZATION\_LEVEL\_DEBUG**

```
#define CONFIG_OPTIMIZATION_LEVEL_DEBUG 1
```

**5.29.1.285 CONFIG\_OSI\_INITIAL\_TRACE\_LEVEL**

```
#define CONFIG_OSI_INITIAL_TRACE_LEVEL 2
```

**5.29.1.286 CONFIG\_OSI\_TRACE\_LEVEL\_WARNING**

```
#define CONFIG_OSI_TRACE_LEVEL_WARNING 1
```

**5.29.1.287 CONFIG\_PAN\_INITIAL\_TRACE\_LEVEL**

```
#define CONFIG_PAN_INITIAL_TRACE_LEVEL 2
```

**5.29.1.288 CONFIG\_PAN\_TRACE\_LEVEL\_WARNING**

```
#define CONFIG_PAN_TRACE_LEVEL_WARNING 1
```

**5.29.1.289 CONFIG\_PARTITION\_TABLE\_CUSTOM\_FILENAME**

```
#define CONFIG_PARTITION_TABLE_CUSTOM_FILENAME "partitions.csv"
```

**5.29.1.290 CONFIG\_PARTITION\_TABLE\_FILENAME**

```
#define CONFIG_PARTITION_TABLE_FILENAME "partitions_singleapp.csv"
```

**5.29.1.291 CONFIG\_PARTITION\_TABLE\_MD5**

```
#define CONFIG_PARTITION_TABLE_MD5 1
```

**5.29.1.292 CONFIG\_PARTITION\_TABLE\_OFFSET**

```
#define CONFIG_PARTITION_TABLE_OFFSET 0x8000
```

**5.29.1.293 CONFIG\_PARTITION\_TABLE\_SINGLE\_APP**

```
#define CONFIG_PARTITION_TABLE_SINGLE_APP 1
```

**5.29.1.294 CONFIG\_PTHREAD\_STACK\_MIN**

```
#define CONFIG_PTHREAD_STACK_MIN 768
```

**5.29.1.295 CONFIG\_PYTHON**

```
#define CONFIG_PYTHON "python"
```

**5.29.1.296 CONFIG\_REDUCE\_PHY\_TX\_POWER**

```
#define CONFIG_REDUCE_PHY_TX_POWER 1
```

**5.29.1.297 CONFIG\_RFCOMM\_INITIAL\_TRACE\_LEVEL**

```
#define CONFIG_RFCOMM_INITIAL_TRACE_LEVEL 2
```

**5.29.1.298 CONFIG\_RFCOMM\_TRACE\_LEVEL\_WARNING**

```
#define CONFIG_RFCOMM_TRACE_LEVEL_WARNING 1
```

**5.29.1.299 CONFIG\_SCAN\_DUPLICATE\_BY\_DEVICE\_ADDR**

```
#define CONFIG_SCAN_DUPLICATE_BY_DEVICE_ADDR 1
```

**5.29.1.300 CONFIG\_SCAN\_DUPLICATE\_TYPE**

```
#define CONFIG_SCAN_DUPLICATE_TYPE 0
```

**5.29.1.301 CONFIG\_SDP\_INITIAL\_TRACE\_LEVEL**

```
#define CONFIG_SDP_INITIAL_TRACE_LEVEL 2
```

**5.29.1.302 CONFIG\_SDP\_TRACE\_LEVEL\_WARNING**

```
#define CONFIG_SDP_TRACE_LEVEL_WARNING 1
```

**5.29.1.303 CONFIG\_SMP\_ENABLE**

```
#define CONFIG_SMP_ENABLE 1
```

**5.29.1.304 CONFIG\_SMP\_INITIAL\_TRACE\_LEVEL**

```
#define CONFIG_SMP_INITIAL_TRACE_LEVEL 2
```

**5.29.1.305 CONFIG\_SMP\_TRACE\_LEVEL\_WARNING**

```
#define CONFIG_SMP_TRACE_LEVEL_WARNING 1
```

**5.29.1.306 CONFIG\_SPI\_FLASH\_ERASE\_YIELD\_DURATION\_MS**

```
#define CONFIG_SPI_FLASH_ERASE_YIELD_DURATION_MS 20
```

**5.29.1.307 CONFIG\_SPI\_FLASH\_ERASE\_YIELD\_TICKS**

```
#define CONFIG_SPI_FLASH_ERASE_YIELD_TICKS 1
```

**5.29.1.308 CONFIG\_SPI\_FLASH\_ROM\_DRIVER\_PATCH**

```
#define CONFIG_SPI_FLASH_ROM_DRIVER_PATCH 1
```

**5.29.1.309 CONFIG\_SPI\_FLASH\_WRITING\_DANGEROUS\_REGIONS\_ABORTS**

```
#define CONFIG_SPI_FLASH_WRITING_DANGEROUS_REGIONS_ABORTS 1
```

**5.29.1.310 CONFIG\_SPI\_FLASH\_YIELD\_DURING\_ERASE**

```
#define CONFIG_SPI_FLASH_YIELD_DURING_ERASE 1
```

**5.29.1.311 CONFIG\_SPI\_MASTER\_ISR\_IN\_IRAM**

```
#define CONFIG_SPI_MASTER_ISR_IN_IRAM 1
```

**5.29.1.312 CONFIG\_SPI\_SLAVE\_ISR\_IN\_IRAM**

```
#define CONFIG_SPI_SLAVE_ISR_IN_IRAM 1
```

**5.29.1.313 CONFIG\_SPIFFS\_CACHE**

```
#define CONFIG_SPIFFS_CACHE 1
```

**5.29.1.314 CONFIG\_SPIFFS\_CACHE\_WR**

```
#define CONFIG_SPIFFS_CACHE_WR 1
```

**5.29.1.315 CONFIG\_SPIFFS\_GC\_MAX\_RUNS**

```
#define CONFIG_SPIFFS_GC_MAX_RUNS 10
```

**5.29.1.316 CONFIG\_SPIFFS\_MAX\_PARTITIONS**

```
#define CONFIG_SPIFFS_MAX_PARTITIONS 3
```

**5.29.1.317 CONFIG\_SPIFFS\_META\_LENGTH**

```
#define CONFIG_SPIFFS_META_LENGTH 4
```

**5.29.1.318 CONFIG\_SPIFFS\_OBJ\_NAME\_LEN**

```
#define CONFIG_SPIFFS_OBJ_NAME_LEN 32
```

**5.29.1.319 CONFIG\_SPIFFS\_PAGE\_CHECK**

```
#define CONFIG_SPIFFS_PAGE_CHECK 1
```

**5.29.1.320 CONFIG\_SPIFFS\_PAGE\_SIZE**

```
#define CONFIG_SPIFFS_PAGE_SIZE 256
```

**5.29.1.321 CONFIG\_SPIFFS\_USE\_MAGIC**

```
#define CONFIG_SPIFFS_USE_MAGIC 1
```

**5.29.1.322 CONFIG\_SPIFFS\_USE\_MAGIC\_LENGTH**

```
#define CONFIG_SPIFFS_USE_MAGIC_LENGTH 1
```

**5.29.1.323 CONFIG\_SPIFFS\_USE\_MTIME**

```
#define CONFIG_SPIFFS_USE_MTIME 1
```

**5.29.1.324 CONFIG\_STACK\_CHECK\_NONE**

```
#define CONFIG_STACK_CHECK_NONE 1
```

**5.29.1.325 CONFIG\_SUPPORT\_TERMIOS**

```
#define CONFIG_SUPPORT_TERMIOS 1
```

**5.29.1.326 CONFIG\_SUPPRESS\_SELECT\_DEBUG\_OUTPUT**

```
#define CONFIG_SUPPRESS_SELECT_DEBUG_OUTPUT 1
```

**5.29.1.327 CONFIG\_SW\_COEXIST\_ENABLE**

```
#define CONFIG_SW_COEXIST_ENABLE 1
```

**5.29.1.328 CONFIG\_SW\_COEXIST\_PREFERENCE\_BALANCE**

```
#define CONFIG_SW_COEXIST_PREFERENCE_BALANCE 1
```

**5.29.1.329 CONFIG\_SW\_COEXIST\_PREFERENCE\_VALUE**

```
#define CONFIG_SW_COEXIST_PREFERENCE_VALUE 2
```

**5.29.1.330 CONFIG\_SYSTEM\_EVENT\_QUEUE\_SIZE**

```
#define CONFIG_SYSTEM_EVENT_QUEUE_SIZE 32
```

**5.29.1.331 CONFIG\_SYSTEM\_EVENT\_TASK\_STACK\_SIZE**

```
#define CONFIG_SYSTEM_EVENT_TASK_STACK_SIZE 2304
```

**5.29.1.332 CONFIG\_TASK\_WDT**

```
#define CONFIG_TASK_WDT 1
```

**5.29.1.333 CONFIG\_TASK\_WDT\_CHECK\_IDLE\_TASK\_CPU0**

```
#define CONFIG_TASK_WDT_CHECK_IDLE_TASK_CPU0 1
```

**5.29.1.334 CONFIG\_TASK\_WDT\_CHECK\_IDLE\_TASK\_CPU1**

```
#define CONFIG_TASK_WDT_CHECK_IDLE_TASK_CPU1 1
```

**5.29.1.335 CONFIG\_TASK\_WDT\_TIMEOUT\_S**

```
#define CONFIG_TASK_WDT_TIMEOUT_S 5
```

**5.29.1.336 CONFIG\_TCP\_MAXRTX**

```
#define CONFIG_TCP_MAXRTX 12
```

**5.29.1.337 CONFIG\_TCP\_MSL**

```
#define CONFIG_TCP_MSL 60000
```

**5.29.1.338 CONFIG\_TCP\_MSS**

```
#define CONFIG_TCP_MSS 1436
```

**5.29.1.339 CONFIG\_TCP\_OVERSIZE\_MSS**

```
#define CONFIG_TCP_OVERSIZE_MSS 1
```

**5.29.1.340 CONFIG\_TCP\_QUEUE\_OOSEQ**

```
#define CONFIG_TCP_QUEUE_OOSEQ 1
```

**5.29.1.341 CONFIG\_TCP\_RECVMBOX\_SIZE**

```
#define CONFIG_TCP_RECVMBOX_SIZE 6
```

**5.29.1.342 CONFIG\_TCP\_SND\_BUF\_DEFAULT**

```
#define CONFIG_TCP_SND_BUF_DEFAULT 5744
```

**5.29.1.343 CONFIG\_TCP\_SYNMAXRTX**

```
#define CONFIG_TCP_SYNMAXRTX 6
```

**5.29.1.344 CONFIG\_TCP\_WND\_DEFAULT**

```
#define CONFIG_TCP_WND_DEFAULT 5744
```

**5.29.1.345 CONFIG\_TCPIP\_LWIP**

```
#define CONFIG_TCPIP_LWIP 1
```



**5.29.1.346 CONFIG\_TCPIP\_RECVMBOX\_SIZE**

```
#define CONFIG_TCPIP_RECVMBOX_SIZE 32
```

**5.29.1.347 CONFIG\_TCPIP\_TASK\_AFFINITY**

```
#define CONFIG_TCPIP_TASK_AFFINITY 0x7FFFFFFF
```

**5.29.1.348 CONFIG\_TCPIP\_TASK\_AFFINITY\_NO\_AFFINITY**

```
#define CONFIG_TCPIP_TASK_AFFINITY_NO_AFFINITY 1
```

**5.29.1.349 CONFIG\_TCPIP\_TASK\_STACK\_SIZE**

```
#define CONFIG_TCPIP_TASK_STACK_SIZE 2048
```

**5.29.1.350 CONFIG\_TIMER\_QUEUE\_LENGTH**

```
#define CONFIG_TIMER_QUEUE_LENGTH 10
```

**5.29.1.351 CONFIG\_TIMER\_TASK\_PRIORITY**

```
#define CONFIG_TIMER_TASK_PRIORITY 1
```

**5.29.1.352 CONFIG\_TIMER\_TASK\_STACK\_DEPTH**

```
#define CONFIG_TIMER_TASK_STACK_DEPTH 2048
```

**5.29.1.353 CONFIG\_TIMER\_TASK\_STACK\_SIZE**

```
#define CONFIG_TIMER_TASK_STACK_SIZE 3584
```

**5.29.1.354 CONFIG\_TOOLPREFIX**

```
#define CONFIG_TOOLPREFIX "xtensa-esp32-elf-"
```

**5.29.1.355 CONFIG\_TRACEMEM\_RESERVE\_DRAM**

```
#define CONFIG_TRACEMEM_RESERVE_DRAM 0x0
```

**5.29.1.356 CONFIG\_UDP\_RECVMBOX\_SIZE**

```
#define CONFIG_UDP_RECVMBOX_SIZE 6
```

**5.29.1.357 CONFIG\_ULP\_COPROC\_RESERVE\_MEM**

```
#define CONFIG_ULP_COPROC_RESERVE_MEM 0
```

**5.29.1.358 CONFIG\_UNITY\_ENABLE\_DOUBLE**

```
#define CONFIG_UNITY_ENABLE_DOUBLE 1
```

**5.29.1.359 CONFIG\_UNITY\_ENABLE\_FLOAT**

```
#define CONFIG_UNITY_ENABLE_FLOAT 1
```

**5.29.1.360 CONFIG\_UNITY\_ENABLE\_IDF\_TEST\_RUNNER**

```
#define CONFIG_UNITY_ENABLE_IDF_TEST_RUNNER 1
```

**5.29.1.361 CONFIG\_WIFI\_PROV\_SCAN\_MAX\_ENTRIES**

```
#define CONFIG_WIFI_PROV_SCAN_MAX_ENTRIES 16
```

### 5.29.1.362 CONFIG\_WL\_SECTOR\_SIZE

```
#define CONFIG_WL_SECTOR_SIZE 4096
```

### 5.29.1.363 CONFIG\_WL\_SECTOR\_SIZE\_4096

```
#define CONFIG_WL_SECTOR_SIZE_4096 1
```

## 5.30 src/wifi.cpp File Reference

File to connect ESP32 to WiFi.

```
#include <string.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/event_groups.h"
#include "esp_system.h"
#include "esp_wifi.h"
#include "esp_event_loop.h"
#include "esp_log.h"
#include "wifi.h"
```

### Macros

- `#define ESP_WIFI_SSID "Esp32"`
- `#define ESP_WIFI_PASS "hello123"`
- `#define MAX_STA_CONN 4`

### Functions

- `void wifi_init_softap ()`
- `void wifi_init_sta ()`

### Variables

- `const int WIFI_CONNECTED_BIT = BIT0`

### 5.30.1 Detailed Description

File to connect ESP32 to WiFi.

#### Author

Andy Yang ( [andyyeung123@gmail.com](mailto:andyyeung123@gmail.com))

#### Version

0.1

#### Date

2020-04-27

#### Copyright

Copyright (c) 2020

### 5.30.2 Macro Definition Documentation

#### 5.30.2.1 ESP\_WIFI\_PASS

```
#define ESP_WIFI_PASS "hello123"
```

#### 5.30.2.2 ESP\_WIFI\_SSID

```
#define ESP_WIFI_SSID "Esp32"
```

#### 5.30.2.3 MAX\_STA\_CONN

```
#define MAX_STA_CONN 4
```

### 5.30.3 Function Documentation

#### 5.30.3.1 wifi\_init\_softap()

```
void wifi_init_softap ( )
```

#### 5.30.3.2 wifi\_init\_sta()

```
void wifi_init_sta ( )
```

### 5.30.4 Variable Documentation

#### 5.30.4.1 WIFI\_CONNECTED\_BIT

```
const int WIFI_CONNECTED_BIT = BIT0
```



# Index

ADV\_CONFIG\_FLAG  
    bleSL.cpp, 62  
app\_id  
    gatts\_profile\_inst, 8  
app\_main  
    main.cpp, 85  
  
b  
    channel, 7  
    Schedule\_Object, 12  
bleSL.cpp  
    ADV\_CONFIG\_FLAG, 62  
    CHAR\_DECLARATION\_SIZE, 62  
    CONFIG\_SET\_RAW\_ADV\_DATA, 62  
    ESP\_APP\_ID, 63  
    example\_exec\_write\_event\_env, 64  
    example\_prepare\_write\_event\_env, 64  
    GATTS\_DEMO\_CHAR\_VAL\_LEN\_MAX, 63  
    GATTS\_TABLE\_TAG, 63  
    heart\_rate\_handle\_table, 64  
    Init\_Bluetooth, 64  
    PREPARE\_BUF\_MAX\_SIZE, 63  
    PROFILE\_APP\_IDX, 63  
    PROFILE\_NUM, 63  
    SAMPLE\_DEVICE\_NAME, 63  
    SCAN\_RSP\_CONFIG\_FLAG, 63  
    SVC\_INST\_ID, 64  
bleSL.h  
    Init\_Bluetooth, 15  
brightness  
    channel, 7  
    Schedule\_Object, 12  
bSPIFFS  
    memory.cpp, 94  
  
CH3\_HIGH  
    pin\_defs.h, 43  
CH3\_LOW  
    pin\_defs.h, 43  
channel, 7  
    b, 7  
    brightness, 7  
    g, 7  
    name, 7  
    r, 8  
channel\_off  
    led.cpp, 82  
    led.h, 31  
channel\_on  
    led.cpp, 83  
    led.h, 32  
CHAR\_DECLARATION\_SIZE  
    bleSL.cpp, 62  
char\_handle  
    gatts\_profile\_inst, 8  
char\_uuid  
    gatts\_profile\_inst, 8  
clear\_schedule\_data  
    memory.cpp, 90  
    memory.h, 38  
clear\_setting\_data  
    memory.cpp, 90  
    memory.h, 38  
clear\_shutdown  
    led.cpp, 83  
    led.h, 32  
clearFaults  
    measurement.cpp, 86  
    measurement.h, 34  
CLRRAM  
    rtcdefine.h, 45  
CONFIG\_A2D\_INITIAL\_TRACE\_LEVEL  
    sdkconfig.h, 111  
CONFIG\_A2D\_TRACE\_LEVEL\_WARNING  
    sdkconfig.h, 111  
CONFIG\_ADC2\_DISABLE\_DAC  
    sdkconfig.h, 112  
CONFIG\_ADC\_CAL\_EFUSE\_TP\_ENABLE  
    sdkconfig.h, 112  
CONFIG\_ADC\_CAL\_EFUSE\_VREF\_ENABLE  
    sdkconfig.h, 112  
CONFIG\_ADC\_CAL\_LUT\_ENABLE  
    sdkconfig.h, 112  
CONFIG\_APP\_COMPILE\_TIME\_DATE  
    sdkconfig.h, 112  
CONFIG\_APPL\_INITIAL\_TRACE\_LEVEL  
    sdkconfig.h, 112  
CONFIG\_APPL\_TRACE\_LEVEL\_WARNING  
    sdkconfig.h, 112  
CONFIG\_ARDUINO\_EVENT\_RUN\_CORE1  
    sdkconfig.h, 112  
CONFIG\_ARDUINO\_EVENT\_RUNNING\_CORE  
    sdkconfig.h, 113  
CONFIG\_ARDUINO\_RUNNING\_CORE  
    sdkconfig.h, 113  
CONFIG\_ARDUINO\_UDP\_RUN\_CORE1  
    sdkconfig.h, 113  
CONFIG\_ARDUINO\_UDP\_RUNNING\_CORE  
    sdkconfig.h, 113

- CONFIG\_AUTOSTART\_ARDUINO  
sdkconfig.h, [113](#)
- CONFIG\_AVCT\_INITIAL\_TRACE\_LEVEL  
sdkconfig.h, [113](#)
- CONFIG\_AVCT\_TRACE\_LEVEL\_WARNING  
sdkconfig.h, [113](#)
- CONFIG\_AVDI\_INITIAL\_TRACE\_LEVEL  
sdkconfig.h, [113](#)
- CONFIG\_AVDI\_TRACE\_LEVEL\_WARNING  
sdkconfig.h, [114](#)
- CONFIG\_AVRC\_INITIAL\_TRACE\_LEVEL  
sdkconfig.h, [114](#)
- CONFIG\_AVRC\_TRACE\_LEVEL\_WARNING  
sdkconfig.h, [114](#)
- CONFIG\_AWS\_IOT\_MQTT\_HOST  
sdkconfig.h, [114](#)
- CONFIG\_AWS\_IOT\_MQTT\_MAX\_RECONNECT\_WAIT\_INTERVAL  
sdkconfig.h, [114](#)
- CONFIG\_AWS\_IOT\_MQTT\_MIN\_RECONNECT\_WAIT\_INTERVAL  
sdkconfig.h, [114](#)
- CONFIG\_AWS\_IOT\_MQTT\_NUM\_SUBSCRIBE\_HANDLES  
sdkconfig.h, [114](#)
- CONFIG\_AWS\_IOT\_MQTT\_PORT  
sdkconfig.h, [114](#)
- CONFIG\_AWS\_IOT\_MQTT\_RX\_BUF\_LEN  
sdkconfig.h, [115](#)
- CONFIG\_AWS\_IOT\_MQTT\_TX\_BUF\_LEN  
sdkconfig.h, [115](#)
- CONFIG\_AWS\_IOT\_SDK  
sdkconfig.h, [115](#)
- CONFIG\_AWS\_IOT\_SHADOW\_MAX\_JSON\_TOKEN\_EXPONENT  
sdkconfig.h, [115](#)
- CONFIG\_AWS\_IOT\_SHADOW\_MAX\_SHADOW\_TOPIC\_LENGTH  
sdkconfig.h, [115](#)
- CONFIG\_AWS\_IOT\_SHADOW\_MAX\_SIMULTANEOUS\_ACS  
sdkconfig.h, [115](#)
- CONFIG\_AWS\_IOT\_SHADOW\_MAX\_SIMULTANEOUS\_THING\_NAMES  
sdkconfig.h, [115](#)
- CONFIG\_AWS\_IOT\_SHADOW\_MAX\_SIZE\_OF\_THING\_NAME  
sdkconfig.h, [115](#)
- CONFIG\_AWS\_IOT\_SHADOW\_MAX\_SIZE\_OF\_UNIQUE\_IDENTIFIER  
sdkconfig.h, [116](#)
- CONFIG\_BLE\_ADV\_REPORT\_DISCARD\_THRSHOLD  
sdkconfig.h, [116](#)
- CONFIG\_BLE\_ADV\_REPORT\_FLOW\_CONTROL\_NUM  
sdkconfig.h, [116](#)
- CONFIG\_BLE\_ADV\_REPORT\_FLOW\_CONTROL\_SUPPORTED  
sdkconfig.h, [116](#)
- CONFIG\_BLE\_ESTABLISH\_LINK\_CONNECTION\_TIMEOUT  
sdkconfig.h, [116](#)
- CONFIG\_BLE\_SCAN\_DUPLICATE  
sdkconfig.h, [116](#)
- CONFIG\_BLE\_SMP\_ENABLE  
sdkconfig.h, [116](#)
- CONFIG\_BLUEDROID\_ENABLED  
sdkconfig.h, [116](#)
- CONFIG\_BLUEDROID\_PINNED\_TO\_CORE  
sdkconfig.h, [117](#)
- CONFIG\_BLUEDROID\_PINNED\_TO\_CORE\_0  
sdkconfig.h, [117](#)
- CONFIG\_BLUFI\_INITIAL\_TRACE\_LEVEL  
sdkconfig.h, [117](#)
- CONFIG\_BLUFI\_TRACE\_LEVEL\_WARNING  
sdkconfig.h, [117](#)
- CONFIG\_BNEP\_INITIAL\_TRACE\_LEVEL  
sdkconfig.h, [117](#)
- CONFIG\_BNEP\_TRACE\_LEVEL\_WARNING  
sdkconfig.h, [117](#)
- CONFIG\_BOOTLOADER\_VDDSDIO\_BOOST\_1\_9V  
sdkconfig.h, [117](#)
- CONFIG\_BOOTLOADER\_WDT\_ENABLE  
sdkconfig.h, [117](#)
- CONFIG\_BOOTLOADER\_WDT\_TIME\_MS  
sdkconfig.h, [118](#)
- CONFIG\_BROWNOUT\_DET  
sdkconfig.h, [118](#)
- CONFIG\_BROWNOUT\_DET\_LVL  
sdkconfig.h, [118](#)
- CONFIG\_BROWNOUT\_DET\_LVL\_SEL\_0  
sdkconfig.h, [118](#)
- CONFIG\_BT\_ACL\_CONNECTIONS  
sdkconfig.h, [118](#)
- CONFIG\_BT\_ENABLED  
sdkconfig.h, [118](#)
- CONFIG\_BT\_RESERVE\_DRAM  
sdkconfig.h, [118](#)
- CONFIG\_BT\_INITIAL\_TRACE\_LEVEL  
sdkconfig.h, [118](#)
- CONFIG\_BT\_TASK\_STACK\_SIZE  
sdkconfig.h, [119](#)
- CONFIG\_BT\_SHORT\_CHANGE\_WARNING  
sdkconfig.h, [119](#)
- CONFIG\_BTDM\_CONTROLLER\_BLE\_MAX\_CONN  
sdkconfig.h, [119](#)
- CONFIG\_BTDM\_CONTROLLER\_BLE\_MAX\_CONN Eff  
sdkconfig.h, [119](#)
- CONFIG\_BTDM\_CONTROLLER\_BR\_EDR\_MAX\_ACL\_CONN Eff  
sdkconfig.h, [119](#)
- CONFIG\_BTDM\_CONTROLLER\_BR\_EDR\_MAX\_SYNC\_CONN Eff  
sdkconfig.h, [119](#)
- CONFIG\_BTDM\_CONTROLLER\_HCI\_MODE\_VHCI  
sdkconfig.h, [119](#)
- CONFIG\_BTDM\_CONTROLLER\_MODE\_BLE\_ONLY  
sdkconfig.h, [119](#)
- CONFIG\_BTDM\_CONTROLLER\_PINNED\_TO\_CORE  
sdkconfig.h, [120](#)
- CONFIG\_BTDM\_CONTROLLER\_PINNED\_TO\_CORE\_0  
sdkconfig.h, [120](#)
- CONFIG\_BTIF\_INITIAL\_TRACE\_LEVEL  
sdkconfig.h, [120](#)
- CONFIG\_BTIF\_TRACE\_LEVEL\_WARNING  
sdkconfig.h, [120](#)
- CONFIG\_BTM\_INITIAL\_TRACE\_LEVEL  
sdkconfig.h, [120](#)
- CONFIG\_BTM\_TRACE\_LEVEL\_WARNING  
sdkconfig.h, [120](#)



- CONFIG\_BTU\_TASK\_STACK\_SIZE  
sdkconfig.h, [120](#)
- CONFIG\_CONSOLE\_UART\_BAUDRATE  
sdkconfig.h, [120](#)
- CONFIG\_CONSOLE\_UART\_DEFAULT  
sdkconfig.h, [121](#)
- CONFIG\_CONSOLE\_UART\_NUM  
sdkconfig.h, [121](#)
- CONFIG\_DMA\_RX\_BUF\_NUM  
sdkconfig.h, [121](#)
- CONFIG\_DMA\_TX\_BUF\_NUM  
sdkconfig.h, [121](#)
- CONFIG\_DUPLICATE\_SCAN\_CACHE\_SIZE  
sdkconfig.h, [121](#)
- CONFIG\_EFUSE\_CODE\_SCHEME\_COMPAT\_3\_4  
sdkconfig.h, [121](#)
- CONFIG\_EFUSE\_MAX\_BLK\_LEN  
sdkconfig.h, [121](#)
- CONFIG\_EMAC\_CHECK\_LINK\_PERIOD\_MS  
sdkconfig.h, [121](#)
- CONFIG\_EMAC\_TASK\_PRIORITY  
sdkconfig.h, [122](#)
- CONFIG\_EMAC\_TASK\_STACK\_SIZE  
sdkconfig.h, [122](#)
- CONFIG\_ENABLE\_ARDUINO\_DEPENDS  
sdkconfig.h, [122](#)
- CONFIG\_ESP32\_APPTRACE\_DEST\_NONE  
sdkconfig.h, [122](#)
- CONFIG\_ESP32\_APPTRACE\_LOCK\_ENABLE  
sdkconfig.h, [122](#)
- CONFIG\_ESP32\_DEBUG\_OCDAWARE  
sdkconfig.h, [122](#)
- CONFIG\_ESP32\_DEBUG\_STUBS\_ENABLE  
sdkconfig.h, [122](#)
- CONFIG\_ESP32\_DEEP\_SLEEP\_WAKEUP\_DELAY  
sdkconfig.h, [122](#)
- CONFIG\_ESP32\_DEFAULT\_CPU\_FREQ\_160  
sdkconfig.h, [123](#)
- CONFIG\_ESP32\_DEFAULT\_CPU\_FREQ\_MHZ  
sdkconfig.h, [123](#)
- CONFIG\_ESP32\_DEFAULT\_PTHREAD\_CORE\_NO\_AFFINITY  
sdkconfig.h, [123](#)
- CONFIG\_ESP32\_DPORT\_WORKAROUND  
sdkconfig.h, [123](#)
- CONFIG\_ESP32\_ENABLE\_COREDUMP\_TO\_NONE  
sdkconfig.h, [123](#)
- CONFIG\_ESP32\_PANIC\_PRINT\_REBOOT  
sdkconfig.h, [123](#)
- CONFIG\_ESP32\_PHY\_CALIBRATION\_AND\_DATA\_STORAGE  
sdkconfig.h, [123](#)
- CONFIG\_ESP32\_PHY\_MAX\_TX\_POWER  
sdkconfig.h, [123](#)
- CONFIG\_ESP32\_PHY\_MAX\_WIFI\_TX\_POWER  
sdkconfig.h, [124](#)
- CONFIG\_ESP32\_PTHREAD\_TASK\_CORE\_DEFAULT  
sdkconfig.h, [124](#)
- CONFIG\_ESP32\_PTHREAD\_TASK\_NAME\_DEFAULT  
sdkconfig.h, [124](#)
- CONFIG\_ESP32\_PTHREAD\_TASK\_PRIO\_DEFAULT  
sdkconfig.h, [124](#)
- CONFIG\_ESP32\_PTHREAD\_TASK\_STACK\_SIZE\_DEFAULT  
sdkconfig.h, [124](#)
- CONFIG\_ESP32\_REV\_MIN  
sdkconfig.h, [124](#)
- CONFIG\_ESP32\_REV\_MIN\_0  
sdkconfig.h, [124](#)
- CONFIG\_ESP32\_RTC\_CLK\_CAL\_CYCLES  
sdkconfig.h, [124](#)
- CONFIG\_ESP32\_RTC\_CLOCK\_SOURCE\_INTERNAL\_RC  
sdkconfig.h, [125](#)
- CONFIG\_ESP32\_TIME\_SYSCALL\_USE\_RTC\_FRC1  
sdkconfig.h, [125](#)
- CONFIG\_ESP32\_WIFI\_AMPDU\_RX\_ENABLED  
sdkconfig.h, [125](#)
- CONFIG\_ESP32\_WIFI\_AMPDU\_TX\_ENABLED  
sdkconfig.h, [125](#)
- CONFIG\_ESP32\_WIFI\_DYNAMIC\_RX\_BUFFER\_NUM  
sdkconfig.h, [125](#)
- CONFIG\_ESP32\_WIFI\_DYNAMIC\_TX\_BUFFER  
sdkconfig.h, [125](#)
- CONFIG\_ESP32\_WIFI\_DYNAMIC\_TX\_BUFFER\_NUM  
sdkconfig.h, [125](#)
- CONFIG\_ESP32\_WIFI\_IRAM\_OPT  
sdkconfig.h, [125](#)
- CONFIG\_ESP32\_WIFI\_MGMT\_SBUF\_NUM  
sdkconfig.h, [126](#)
- CONFIG\_ESP32\_WIFI\_NVS\_ENABLED  
sdkconfig.h, [126](#)
- CONFIG\_ESP32\_WIFI\_RX\_BA\_WIN  
sdkconfig.h, [126](#)
- CONFIG\_ESP32\_WIFI\_SOFTAP\_BEACON\_MAX\_LEN  
sdkconfig.h, [126](#)
- CONFIG\_ESP32\_WIFI\_STATIC\_RX\_BUFFER\_NUM  
sdkconfig.h, [126](#)
- CONFIG\_ESP32\_WIFI\_TASK\_PINNED\_TO\_CORE\_0  
sdkconfig.h, [126](#)
- CONFIG\_ESP32\_WIFI\_TX\_BA\_WIN  
sdkconfig.h, [126](#)
- CONFIG\_ESP32\_WIFI\_TX\_BUFFER\_TYPE  
sdkconfig.h, [126](#)
- CONFIG\_ESP32\_XTAL\_FREQ  
sdkconfig.h, [127](#)
- CONFIG\_ESP32\_XTAL\_FREQ\_40  
sdkconfig.h, [127](#)
- CONFIG\_ESP\_ERR\_TO\_NAME\_LOOKUP  
sdkconfig.h, [127](#)
- CONFIG\_ESP\_GRATUITOUS\_ARP  
sdkconfig.h, [127](#)
- CONFIG\_ESP\_HTTP\_CLIENT\_ENABLE\_HTTPS  
sdkconfig.h, [127](#)
- CONFIG\_ESPTOOLPY\_AFTER  
sdkconfig.h, [127](#)
- CONFIG\_ESPTOOLPY\_AFTER\_RESET  
sdkconfig.h, [127](#)
- CONFIG\_ESPTOOLPY\_BAUD  
sdkconfig.h, [127](#)

- CONFIG\_ESPTOOLPY\_BAUD\_115200B  
sdkconfig.h, [128](#)
- CONFIG\_ESPTOOLPY\_BAUD\_OTHER\_VAL  
sdkconfig.h, [128](#)
- CONFIG\_ESPTOOLPY\_BEFORE  
sdkconfig.h, [128](#)
- CONFIG\_ESPTOOLPY\_BEFORE\_RESET  
sdkconfig.h, [128](#)
- CONFIG\_ESPTOOLPY\_COMPRESSED  
sdkconfig.h, [128](#)
- CONFIG\_ESPTOOLPY\_FLASHFREQ  
sdkconfig.h, [128](#)
- CONFIG\_ESPTOOLPY\_FLASHFREQ\_40M  
sdkconfig.h, [128](#)
- CONFIG\_ESPTOOLPY\_FLASHMODE  
sdkconfig.h, [128](#)
- CONFIG\_ESPTOOLPY\_FLASHSIZE  
sdkconfig.h, [129](#)
- CONFIG\_ESPTOOLPY\_FLASHSIZE\_2MB  
sdkconfig.h, [129](#)
- CONFIG\_ESPTOOLPY\_FLASHSIZE\_DETECT  
sdkconfig.h, [129](#)
- CONFIG\_ESPTOOLPY\_PORT  
sdkconfig.h, [129](#)
- CONFIG\_FATFS\_CODEPAGE  
sdkconfig.h, [129](#)
- CONFIG\_FATFS\_CODEPAGE\_437  
sdkconfig.h, [129](#)
- CONFIG\_FATFS\_FS\_LOCK  
sdkconfig.h, [129](#)
- CONFIG\_FATFS\_LFN\_NONE  
sdkconfig.h, [129](#)
- CONFIG\_FATFS\_PER\_FILE\_CACHE  
sdkconfig.h, [130](#)
- CONFIG\_FATFS\_TIMEOUT\_MS  
sdkconfig.h, [130](#)
- CONFIG\_FLASHMODE\_DIO  
sdkconfig.h, [130](#)
- CONFIG\_FOUR\_UNIVERSAL\_MAC\_ADDRESS  
sdkconfig.h, [130](#)
- CONFIG\_FREERTOS\_ASSERT\_FAIL\_ABORT  
sdkconfig.h, [130](#)
- CONFIG\_FREERTOS\_ASSERT\_ON\_UNTESTED\_FUNCTION  
sdkconfig.h, [130](#)
- CONFIG\_FREERTOS\_CHECK\_MUTEX\_GIVEN\_BY\_OWNER  
sdkconfig.h, [130](#)
- CONFIG\_FREERTOS\_CHECK\_STACKOVERFLOW\_CANARY  
sdkconfig.h, [130](#)
- CONFIG\_FREERTOS\_CORETIMER\_0  
sdkconfig.h, [131](#)
- CONFIG\_FREERTOS\_HZ  
sdkconfig.h, [131](#)
- CONFIG\_FREERTOS\_IDLE\_TASK\_STACKSIZE  
sdkconfig.h, [131](#)
- CONFIG\_FREERTOS\_INTERRUPT\_BACKTRACE  
sdkconfig.h, [131](#)
- CONFIG\_FREERTOS\_ISR\_STACKSIZE  
sdkconfig.h, [131](#)
- CONFIG\_FREERTOS\_MAX\_TASK\_NAME\_LEN  
sdkconfig.h, [131](#)
- CONFIG\_FREERTOS\_NO\_AFFINITY  
sdkconfig.h, [131](#)
- CONFIG\_FREERTOS\_QUEUE\_REGISTRY\_SIZE  
sdkconfig.h, [131](#)
- CONFIG\_FREERTOS\_TASK\_FUNCTION\_WRAPPER  
sdkconfig.h, [132](#)
- CONFIG\_FREERTOS\_THREAD\_LOCAL\_STORAGE\_POINTERS  
sdkconfig.h, [132](#)
- CONFIG\_GAP\_INITIAL\_TRACE\_LEVEL  
sdkconfig.h, [132](#)
- CONFIG\_GAP\_TRACE\_LEVEL\_WARNING  
sdkconfig.h, [132](#)
- CONFIG\_GARP\_TMR\_INTERVAL  
sdkconfig.h, [132](#)
- CONFIG\_GATT\_INITIAL\_TRACE\_LEVEL  
sdkconfig.h, [132](#)
- CONFIG\_GATT\_TRACE\_LEVEL\_WARNING  
sdkconfig.h, [132](#)
- CONFIG\_GATTC\_ENABLE  
sdkconfig.h, [132](#)
- CONFIG\_GATTS\_ENABLE  
sdkconfig.h, [133](#)
- CONFIG\_GATTS\_SEND\_SERVICE\_CHANGE\_AUTO  
sdkconfig.h, [133](#)
- CONFIG\_GATTS\_SEND\_SERVICE\_CHANGE\_MODE  
sdkconfig.h, [133](#)
- CONFIG\_HCI\_INITIAL\_TRACE\_LEVEL  
sdkconfig.h, [133](#)
- CONFIG\_HCI\_TRACE\_LEVEL\_WARNING  
sdkconfig.h, [133](#)
- CONFIG\_HEAP\_POISONING\_DISABLED  
sdkconfig.h, [133](#)
- CONFIG\_HID\_INITIAL\_TRACE\_LEVEL  
sdkconfig.h, [133](#)
- CONFIG\_HID\_TRACE\_LEVEL\_WARNING  
sdkconfig.h, [133](#)
- CONFIG\_HTTPD\_ERR\_RESP\_NO\_DELAY  
sdkconfig.h, [134](#)
- CONFIG\_HTTPD\_MAX\_REQ\_HDR\_LEN  
sdkconfig.h, [134](#)
- CONFIG\_HTTPD\_MAX\_URI\_LEN  
sdkconfig.h, [134](#)
- CONFIG\_HTTPD\_PURGE\_BUF\_LEN  
sdkconfig.h, [134](#)
- CONFIG\_IDF\_TARGET  
sdkconfig.h, [134](#)
- CONFIG\_IDF\_TARGET\_ESP32  
sdkconfig.h, [134](#)
- CONFIG\_INT\_WDT  
sdkconfig.h, [134](#)
- CONFIG\_INT\_WDT\_CHECK\_CPU1  
sdkconfig.h, [134](#)
- CONFIG\_INT\_WDT\_TIMEOUT\_MS  
sdkconfig.h, [135](#)
- CONFIG\_IP\_LOST\_TIMER\_INTERVAL  
sdkconfig.h, [135](#)

- CONFIG\_IPC\_TASK\_STACK\_SIZE  
sdkconfig.h, [135](#)
- CONFIG\_L2CAP\_INITIAL\_TRACE\_LEVEL  
sdkconfig.h, [135](#)
- CONFIG\_L2CAP\_TRACE\_LEVEL\_WARNING  
sdkconfig.h, [135](#)
- CONFIG\_LIBSODIUM\_USE\_MBEDTLS\_SHA  
sdkconfig.h, [135](#)
- CONFIG\_LOG\_BOOTLOADER\_LEVEL  
sdkconfig.h, [135](#)
- CONFIG\_LOG\_BOOTLOADER\_LEVEL\_INFO  
sdkconfig.h, [135](#)
- CONFIG\_LOG\_COLORS  
sdkconfig.h, [136](#)
- CONFIG\_LOG\_DEFAULT\_LEVEL  
sdkconfig.h, [136](#)
- CONFIG\_LOG\_DEFAULT\_LEVEL\_INFO  
sdkconfig.h, [136](#)
- CONFIG\_LWIP\_DHCP\_DOES\_ARP\_CHECK  
sdkconfig.h, [136](#)
- CONFIG\_LWIP\_DHCP\_MAX\_NTP\_SERVERS  
sdkconfig.h, [136](#)
- CONFIG\_LWIP\_DHCP\_LEASE\_UNIT  
sdkconfig.h, [136](#)
- CONFIG\_LWIP\_DHCP\_MAX\_STATION\_NUM  
sdkconfig.h, [136](#)
- CONFIG\_LWIP\_LOOPBACK\_MAX\_PBUFS  
sdkconfig.h, [136](#)
- CONFIG\_LWIP\_MAX\_ACTIVE\_TCP  
sdkconfig.h, [137](#)
- CONFIG\_LWIP\_MAX\_LISTENING\_TCP  
sdkconfig.h, [137](#)
- CONFIG\_LWIP\_MAX\_RAW\_PCBS  
sdkconfig.h, [137](#)
- CONFIG\_LWIP\_MAX\_SOCKETS  
sdkconfig.h, [137](#)
- CONFIG\_LWIP\_MAX\_UDP\_PCBS  
sdkconfig.h, [137](#)
- CONFIG\_LWIP\_NETIF\_LOOPBACK  
sdkconfig.h, [137](#)
- CONFIG\_LWIP\_SO\_REUSE  
sdkconfig.h, [137](#)
- CONFIG\_LWIP\_SO\_REUSE\_RXTOALL  
sdkconfig.h, [137](#)
- CONFIG\_MAIN\_TASK\_STACK\_SIZE  
sdkconfig.h, [138](#)
- CONFIG\_MAKE\_WARN\_UNDEFINED\_VARIABLES  
sdkconfig.h, [138](#)
- CONFIG\_MB\_CONTROLLER\_NOTIFY\_QUEUE\_SIZE  
sdkconfig.h, [138](#)
- CONFIG\_MB\_CONTROLLER\_NOTIFY\_TIMEOUT  
sdkconfig.h, [138](#)
- CONFIG\_MB\_CONTROLLER\_STACK\_SIZE  
sdkconfig.h, [138](#)
- CONFIG\_MB\_EVENT\_QUEUE\_TIMEOUT  
sdkconfig.h, [138](#)
- CONFIG\_MB\_QUEUE\_LENGTH  
sdkconfig.h, [138](#)
- CONFIG\_MB\_SERIAL\_BUF\_SIZE  
sdkconfig.h, [138](#)
- CONFIG\_MB\_SERIAL\_TASK\_PRIO  
sdkconfig.h, [139](#)
- CONFIG\_MB\_SERIAL\_TASK\_STACK\_SIZE  
sdkconfig.h, [139](#)
- CONFIG\_MB\_TIMER\_GROUP  
sdkconfig.h, [139](#)
- CONFIG\_MB\_TIMER\_INDEX  
sdkconfig.h, [139](#)
- CONFIG\_MB\_TIMER\_PORT\_ENABLED  
sdkconfig.h, [139](#)
- CONFIG\_MBEDTLS\_AES\_C  
sdkconfig.h, [139](#)
- CONFIG\_MBEDTLS\_CCM\_C  
sdkconfig.h, [139](#)
- CONFIG\_MBEDTLS\_ECDH\_C  
sdkconfig.h, [139](#)
- CONFIG\_MBEDTLS\_ECDSA\_C  
sdkconfig.h, [140](#)
- CONFIG\_MBEDTLS\_ECP\_C  
sdkconfig.h, [140](#)
- CONFIG\_MBEDTLS\_ECP\_DP\_BP256R1\_ENABLED  
sdkconfig.h, [140](#)
- CONFIG\_MBEDTLS\_ECP\_DP\_BP384R1\_ENABLED  
sdkconfig.h, [140](#)
- CONFIG\_MBEDTLS\_ECP\_DP\_BP512R1\_ENABLED  
sdkconfig.h, [140](#)
- CONFIG\_MBEDTLS\_ECP\_DP\_CURVE25519\_ENABLED  
sdkconfig.h, [140](#)
- CONFIG\_MBEDTLS\_ECP\_DP\_SECP192K1\_ENABLED  
sdkconfig.h, [140](#)
- CONFIG\_MBEDTLS\_ECP\_DP\_SECP192R1\_ENABLED  
sdkconfig.h, [140](#)
- CONFIG\_MBEDTLS\_ECP\_DP\_SECP224K1\_ENABLED  
sdkconfig.h, [141](#)
- CONFIG\_MBEDTLS\_ECP\_DP\_SECP224R1\_ENABLED  
sdkconfig.h, [141](#)
- CONFIG\_MBEDTLS\_ECP\_DP\_SECP256K1\_ENABLED  
sdkconfig.h, [141](#)
- CONFIG\_MBEDTLS\_ECP\_DP\_SECP256R1\_ENABLED  
sdkconfig.h, [141](#)
- CONFIG\_MBEDTLS\_ECP\_DP\_SECP384R1\_ENABLED  
sdkconfig.h, [141](#)
- CONFIG\_MBEDTLS\_ECP\_DP\_SECP521R1\_ENABLED  
sdkconfig.h, [141](#)
- CONFIG\_MBEDTLS\_ECP\_NIST\_OPTIM  
sdkconfig.h, [141](#)
- CONFIG\_MBEDTLS\_GCM\_C  
sdkconfig.h, [141](#)
- CONFIG\_MBEDTLS\_HARDWARE\_AES  
sdkconfig.h, [142](#)
- CONFIG\_MBEDTLS\_HAVE\_TIME  
sdkconfig.h, [142](#)
- CONFIG\_MBEDTLS\_INTERNAL\_MEM\_ALLOC  
sdkconfig.h, [142](#)
- CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_DHE\_RSA  
sdkconfig.h, [142](#)

CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_ECDH\_ECDSA sdkconfig.h, [142](#)  
 CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_ECDH\_RSA sdkconfig.h, [142](#)  
 CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_ECDHE\_ECDSA sdkconfig.h, [142](#)  
 CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_ECDHE\_RSA sdkconfig.h, [142](#)  
 CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_ELLIPTIC\_CURVE sdkconfig.h, [143](#)  
 CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_RSA sdkconfig.h, [143](#)  
 CONFIG\_MBEDTLS\_PEM\_PARSE\_C sdkconfig.h, [143](#)  
 CONFIG\_MBEDTLS\_PEM\_WRITE\_C sdkconfig.h, [143](#)  
 CONFIG\_MBEDTLS\_RC4\_DISABLED sdkconfig.h, [143](#)  
 CONFIG\_MBEDTLS\_SSL\_ALPN sdkconfig.h, [143](#)  
 CONFIG\_MBEDTLS\_SSL\_MAX\_CONTENT\_LEN sdkconfig.h, [143](#)  
 CONFIG\_MBEDTLS\_SSL\_PROTO\_TLS1 sdkconfig.h, [143](#)  
 CONFIG\_MBEDTLS\_SSL\_PROTO\_TLS1\_1 sdkconfig.h, [144](#)  
 CONFIG\_MBEDTLS\_SSL\_PROTO\_TLS1\_2 sdkconfig.h, [144](#)  
 CONFIG\_MBEDTLS\_SSL\_RENEGOTIATION sdkconfig.h, [144](#)  
 CONFIG\_MBEDTLS\_SSL\_SESSION\_TICKETS sdkconfig.h, [144](#)  
 CONFIG\_MBEDTLS\_TLS\_CLIENT sdkconfig.h, [144](#)  
 CONFIG\_MBEDTLS\_TLS\_ENABLED sdkconfig.h, [144](#)  
 CONFIG\_MBEDTLS\_TLS\_SERVER sdkconfig.h, [144](#)  
 CONFIG\_MBEDTLS\_TLS\_SERVER\_AND\_CLIENT sdkconfig.h, [144](#)  
 CONFIG\_MBEDTLS\_X509\_CRL\_PARSE\_C sdkconfig.h, [145](#)  
 CONFIG\_MBEDTLS\_X509\_CSR\_PARSE\_C sdkconfig.h, [145](#)  
 CONFIG\_MCA\_INITIAL\_TRACE\_LEVEL sdkconfig.h, [145](#)  
 CONFIG\_MCA\_TRACE\_LEVEL\_WARNING sdkconfig.h, [145](#)  
 CONFIG\_MDNS\_MAX\_SERVICES sdkconfig.h, [145](#)  
 CONFIG\_MONITOR\_BAUD sdkconfig.h, [145](#)  
 CONFIG\_MONITOR\_BAUD\_115200B sdkconfig.h, [145](#)  
 CONFIG\_MONITOR\_BAUD\_OTHER\_VAL sdkconfig.h, [145](#)  
 CONFIG\_MQTT\_PROTOCOL\_311 sdkconfig.h, [146](#)  
 CONFIG\_MQTT\_TRANSPORT\_SSL sdkconfig.h, [146](#)  
 CONFIG\_MQTT\_TRANSPORT\_WEBSOCKET sdkconfig.h, [146](#)  
 CONFIG\_MQTT\_TRANSPORT\_WEBSOCKET\_SECURE sdkconfig.h, [146](#)  
 CONFIG\_NEWLIB\_STDIN\_LINE\_ENDING\_CR sdkconfig.h, [146](#)  
 CONFIG\_NEWLIB\_STDOUT\_LINE\_ENDING\_CRLF sdkconfig.h, [146](#)  
 CONFIG\_NUMBER\_OF\_UNIVERSAL\_MAC\_ADDRESS sdkconfig.h, [146](#)  
 CONFIG\_OPENSSL\_ASSERT\_DO\_NOTHING sdkconfig.h, [146](#)  
 CONFIG\_OPTIMIZATION\_ASSERTIONS\_ENABLED sdkconfig.h, [147](#)  
 CONFIG\_OPTIMIZATION\_LEVEL\_DEBUG sdkconfig.h, [147](#)  
 CONFIG\_OSI\_INITIAL\_TRACE\_LEVEL sdkconfig.h, [147](#)  
 CONFIG\_OSI\_TRACE\_LEVEL\_WARNING sdkconfig.h, [147](#)  
 CONFIG\_PAN\_INITIAL\_TRACE\_LEVEL sdkconfig.h, [147](#)  
 CONFIG\_PAN\_TRACE\_LEVEL\_WARNING sdkconfig.h, [147](#)  
 CONFIG\_PARTITION\_TABLE\_CUSTOM\_FILENAME sdkconfig.h, [147](#)  
 CONFIG\_PARTITION\_TABLE\_FILENAME sdkconfig.h, [147](#)  
 CONFIG\_PARTITION\_TABLE\_MD5 sdkconfig.h, [148](#)  
 CONFIG\_PARTITION\_TABLE\_OFFSET sdkconfig.h, [148](#)  
 CONFIG\_PARTITION\_TABLE\_SINGLE\_APP sdkconfig.h, [148](#)  
 CONFIG\_PTHREAD\_STACK\_MIN sdkconfig.h, [148](#)  
 CONFIG\_PYTHON sdkconfig.h, [148](#)  
 CONFIG\_REDUCE\_PHY\_TX\_POWER sdkconfig.h, [148](#)  
 CONFIG\_RFCOMM\_INITIAL\_TRACE\_LEVEL sdkconfig.h, [148](#)  
 CONFIG\_RFCOMM\_TRACE\_LEVEL\_WARNING sdkconfig.h, [148](#)  
 CONFIG\_SCAN\_DUPLICATE\_BY\_DEVICE\_ADDR sdkconfig.h, [149](#)  
 CONFIG\_SCAN\_DUPLICATE\_TYPE sdkconfig.h, [149](#)  
 CONFIG\_SDP\_INITIAL\_TRACE\_LEVEL sdkconfig.h, [149](#)  
 CONFIG\_SDP\_TRACE\_LEVEL\_WARNING sdkconfig.h, [149](#)  
 CONFIG\_SET\_RAW\_ADV\_DATA bleSL.cpp, [62](#)  
 CONFIG\_SMP\_ENABLE sdkconfig.h, [149](#)

- CONFIG\_SMP\_INITIAL\_TRACE\_LEVEL  
sdkconfig.h, [149](#)
- CONFIG\_SMP\_TRACE\_LEVEL\_WARNING  
sdkconfig.h, [149](#)
- CONFIG\_SPI\_FLASH\_ERASE\_YIELD\_DURATION\_MS  
sdkconfig.h, [149](#)
- CONFIG\_SPI\_FLASH\_ERASE\_YIELD\_TICKS  
sdkconfig.h, [150](#)
- CONFIG\_SPI\_FLASH\_ROM\_DRIVER\_PATCH  
sdkconfig.h, [150](#)
- CONFIG\_SPI\_FLASH\_WRITING\_DANGEROUS\_REGIONS  
sdkconfig.h, [150](#)
- CONFIG\_SPI\_FLASH\_YIELD\_DURING\_ERASE  
sdkconfig.h, [150](#)
- CONFIG\_SPI\_MASTER\_ISR\_IN\_IRAM  
sdkconfig.h, [150](#)
- CONFIG\_SPI\_SLAVE\_ISR\_IN\_IRAM  
sdkconfig.h, [150](#)
- CONFIG\_SPIFFS\_CACHE  
sdkconfig.h, [150](#)
- CONFIG\_SPIFFS\_CACHE\_WR  
sdkconfig.h, [150](#)
- CONFIG\_SPIFFS\_GC\_MAX\_RUNS  
sdkconfig.h, [151](#)
- CONFIG\_SPIFFS\_MAX\_PARTITIONS  
sdkconfig.h, [151](#)
- CONFIG\_SPIFFS\_META\_LENGTH  
sdkconfig.h, [151](#)
- CONFIG\_SPIFFS\_OBJ\_NAME\_LEN  
sdkconfig.h, [151](#)
- CONFIG\_SPIFFS\_PAGE\_CHECK  
sdkconfig.h, [151](#)
- CONFIG\_SPIFFS\_PAGE\_SIZE  
sdkconfig.h, [151](#)
- CONFIG\_SPIFFS\_USE\_MAGIC  
sdkconfig.h, [151](#)
- CONFIG\_SPIFFS\_USE\_MAGIC\_LENGTH  
sdkconfig.h, [151](#)
- CONFIG\_SPIFFS\_USE\_MTIME  
sdkconfig.h, [152](#)
- CONFIG\_STACK\_CHECK\_NONE  
sdkconfig.h, [152](#)
- CONFIG\_SUPPORT\_TERMIOS  
sdkconfig.h, [152](#)
- CONFIG\_SUPPRESS\_SELECT\_DEBUG\_OUTPUT  
sdkconfig.h, [152](#)
- CONFIG\_SW\_COEXIST\_ENABLE  
sdkconfig.h, [152](#)
- CONFIG\_SW\_COEXIST\_PREFERENCE\_BALANCE  
sdkconfig.h, [152](#)
- CONFIG\_SW\_COEXIST\_PREFERENCE\_VALUE  
sdkconfig.h, [152](#)
- CONFIG\_SYSTEM\_EVENT\_QUEUE\_SIZE  
sdkconfig.h, [152](#)
- CONFIG\_SYSTEM\_EVENT\_TASK\_STACK\_SIZE  
sdkconfig.h, [153](#)
- CONFIG\_TASK\_WDT  
sdkconfig.h, [153](#)
- CONFIG\_TASK\_WDT\_CHECK\_IDLE\_TASK\_CPU0  
sdkconfig.h, [153](#)
- CONFIG\_TASK\_WDT\_CHECK\_IDLE\_TASK\_CPU1  
sdkconfig.h, [153](#)
- CONFIG\_TASK\_WDT\_TIMEOUT\_S  
sdkconfig.h, [153](#)
- CONFIG\_TCP\_MAXRTX  
sdkconfig.h, [153](#)
- CONFIG\_TCP\_MSL  
sdkconfig.h, [153](#)
- CONFIG\_TCP\_MSS  
sdkconfig.h, [153](#)
- CONFIG\_TCP\_OVERSIZE\_MSS  
sdkconfig.h, [154](#)
- CONFIG\_TCP\_QUEUE\_OOSEQ  
sdkconfig.h, [154](#)
- CONFIG\_TCP\_RECVMBOX\_SIZE  
sdkconfig.h, [154](#)
- CONFIG\_TCP\_SND\_BUF\_DEFAULT  
sdkconfig.h, [154](#)
- CONFIG\_TCP\_SYNMAXRTX  
sdkconfig.h, [154](#)
- CONFIG\_TCP\_WND\_DEFAULT  
sdkconfig.h, [154](#)
- CONFIG\_TCPIP\_LWIP  
sdkconfig.h, [154](#)
- CONFIG\_TCPIP\_RECVMBOX\_SIZE  
sdkconfig.h, [154](#)
- CONFIG\_TCPIP\_TASK\_AFFINITY  
sdkconfig.h, [155](#)
- CONFIG\_TCPIP\_TASK\_AFFINITY\_NO\_AFFINITY  
sdkconfig.h, [155](#)
- CONFIG\_TCPIP\_TASK\_STACK\_SIZE  
sdkconfig.h, [155](#)
- CONFIG\_TIMER\_QUEUE\_LENGTH  
sdkconfig.h, [155](#)
- CONFIG\_TIMER\_TASK\_PRIORITY  
sdkconfig.h, [155](#)
- CONFIG\_TIMER\_TASK\_STACK\_DEPTH  
sdkconfig.h, [155](#)
- CONFIG\_TIMER\_TASK\_STACK\_SIZE  
sdkconfig.h, [155](#)
- CONFIG\_TOOLPREFIX  
sdkconfig.h, [155](#)
- CONFIG\_TRACEMEM\_RESERVE\_DRAM  
sdkconfig.h, [156](#)
- CONFIG\_UDP\_RECVMBOX\_SIZE  
sdkconfig.h, [156](#)
- CONFIG\_ULP\_COPROC\_RESERVE\_MEM  
sdkconfig.h, [156](#)
- CONFIG\_UNITY\_ENABLE\_DOUBLE  
sdkconfig.h, [156](#)
- CONFIG\_UNITY\_ENABLE\_FLOAT  
sdkconfig.h, [156](#)
- CONFIG\_UNITY\_ENABLE\_IDF\_TEST\_RUNNER  
sdkconfig.h, [156](#)
- CONFIG\_WIFI\_PROV\_SCAN\_MAX\_ENTRIES  
sdkconfig.h, [156](#)

CONFIG\_WL\_SECTOR\_SIZE  
     sdkconfig.h, 156  
 CONFIG\_WL\_SECTOR\_SIZE\_4096  
     sdkconfig.h, 157  
 conn\_id  
     gatts\_profile\_inst, 9  
 CONNECTED\_BIT  
     espsntp.cpp, 74  
 CONTROL  
     rtcdefine.h, 46  
 create\_schedule  
     scheduler.cpp, 99  
     scheduler.h, 55  
 currentFault  
     measurement.cpp, 88  
 currTime  
     rtcdefine.h, 52  
  
 DAC1  
     pin\_defs.h, 43  
 DAC2  
     pin\_defs.h, 43  
 dawn  
     Schedule\_Object, 12  
 dawnCalc  
     dawndusk.cpp, 65  
     dawndusk.h, 17  
 dawndusk.cpp  
     dawnCalc, 65  
     degToRad, 66  
     duskCalc, 66  
     sign, 66  
 dawndusk.h  
     dawnCalc, 17  
     degToRad, 18  
     distSun, 16  
     duskCalc, 18  
     pi, 17  
     planeDist, 17  
     radius, 17  
     sign, 18  
 decode\_ble\_delete  
     decode\_bluetooth.cpp, 68  
     decode\_bluetooth.h, 20  
 decode\_ble\_direct  
     decode\_bluetooth.cpp, 69  
     decode\_bluetooth.h, 20  
 decode\_ble\_schedule  
     decode\_bluetooth.cpp, 69  
     decode\_bluetooth.h, 21  
 decode\_ble\_schedule\_name  
     decode\_bluetooth.cpp, 69  
     decode\_bluetooth.h, 21  
 decode\_ble\_time  
     decode\_bluetooth.cpp, 70  
     decode\_bluetooth.h, 21  
 decode\_bluetooth.cpp  
     decode\_ble\_delete, 68  
     decode\_ble\_direct, 69  
     decode\_ble\_schedule, 69  
     decode\_ble\_schedule\_name, 69  
     decode\_ble\_time, 70  
     done\_string, 71  
     get\_Int32, 70  
     it, 71  
     saved\_channel\_num, 72  
     saved\_len, 72  
     saved\_name, 72  
     schedule\_name, 72  
     schedule\_value, 72  
     schedules, 72  
     set\_schedule\_read, 70  
     start\_schedule\_read, 71  
     state, 72  
 decode\_bluetooth.h  
     decode\_ble\_delete, 20  
     decode\_ble\_direct, 20  
     decode\_ble\_schedule, 21  
     decode\_ble\_schedule\_name, 21  
     decode\_ble\_time, 21  
     DECODE\_BLUETOOTH\_H, 20  
     get\_Int32, 22  
     set\_schedule\_read, 22  
     start\_schedule\_read, 23  
 DECODE\_BLUETOOTH\_H  
     decode\_bluetooth.h, 20  
 degToRad  
     dawndusk.cpp, 66  
     dawndusk.h, 18  
 delete\_all\_schedules  
     scheduler.cpp, 100  
     scheduler.h, 55  
 delete\_schedule\_by\_id  
     scheduler.cpp, 100  
     scheduler.h, 55  
 delete\_schedule\_by\_name  
     scheduler.cpp, 100  
     scheduler.h, 56  
 DELIMITER  
     http.cpp, 75  
 descr\_handle  
     gatts\_profile\_inst, 9  
 descr\_uuid  
     gatts\_profile\_inst, 9  
 direct\_control\_post\_handler  
     http.cpp, 76  
     http.h, 26  
 disable\_all\_schedules  
     scheduler.cpp, 101  
     scheduler.h, 56  
 disable\_schedule\_by\_id  
     scheduler.cpp, 101  
     scheduler.h, 56  
 disable\_schedule\_by\_name  
     scheduler.cpp, 101  
     scheduler.h, 57  
 distSun

- dawndusk.h, 16
- done\_string
  - decode\_bluetooth.cpp, 71
- duration
  - Schedule\_Object, 12
- dusk
  - Schedule\_Object, 12
- duskCalc
  - dawndusk.cpp, 66
  - dawndusk.h, 18
- EEREAD
  - rtcdefine.h, 46
- EEWRDI
  - rtcdefine.h, 46
- EEWREN
  - rtcdefine.h, 46
- EEWRITE
  - rtcdefine.h, 46
- enable\_all\_schedules
  - scheduler.cpp, 102
  - scheduler.h, 57
- enable\_schedule\_by\_id
  - scheduler.cpp, 102
  - scheduler.h, 57
- enable\_schedule\_by\_name
  - scheduler.cpp, 102
  - scheduler.h, 58
- enabled
  - Schedule\_Object, 13
- ESP\_APP\_ID
  - bleSL.cpp, 63
- ESP\_WIFI\_PASS
  - wifi.cpp, 158
- ESP\_WIFI\_SSID
  - wifi.cpp, 158
- espsntp.cpp
  - CONNECTED\_BIT, 74
  - obtain\_time, 73
  - set\_time, 73
- espsntp.h
  - obtain\_time, 24
  - set\_time, 24
- example\_exec\_write\_event\_env
  - bleSL.cpp, 64
- example\_prepare\_write\_event\_env
  - bleSL.cpp, 64
- favicon\_ico\_get\_handler
  - http.cpp, 76
  - http.h, 26
- g
  - channel, 7
  - Schedule\_Object, 13
- gatts\_cb
  - gatts\_profile\_inst, 9
- GATTS\_DEMO\_CHAR\_VAL\_LEN\_MAX
  - bleSL.cpp, 63
- gatts\_if
  - gatts\_profile\_inst, 9
- gatts\_profile\_inst, 8
  - app\_id, 8
  - char\_handle, 8
  - char\_uuid, 8
  - conn\_id, 9
  - descr\_handle, 9
  - descr\_uuid, 9
  - gatts\_cb, 9
  - gatts\_if, 9
  - perm, 9
  - property, 9
  - service\_handle, 9
  - service\_id, 10
- gatts\_table\_creat\_demo.h
  - HRS\_IDX\_NB, 24
  - IDX\_CHAR\_A, 24
  - IDX\_CHAR\_B, 24
  - IDX\_CHAR\_C, 24
  - IDX\_CHAR\_CFG\_A, 24
  - IDX\_CHAR\_VAL\_A, 24
  - IDX\_CHAR\_VAL\_B, 24
  - IDX\_CHAR\_VAL\_C, 24
  - IDX\_SVC, 24
- GATTS\_TABLE\_TAG
  - bleSL.cpp, 63
- get\_Int32
  - decode\_bluetooth.cpp, 70
  - decode\_bluetooth.h, 22
- get\_schedule
  - scheduler.cpp, 103
  - scheduler.h, 58
- get\_schedule\_names
  - scheduler.cpp, 103
  - scheduler.h, 59
- get\_setting\_byte
  - memory.cpp, 90
  - memory.h, 38
- get\_setting\_double
  - memory.cpp, 90
  - memory.h, 38
- get\_setting\_int
  - memory.cpp, 91
  - memory.h, 39
- get\_setting\_string
  - memory.cpp, 91
  - memory.h, 39
- getTime
  - rtc.cpp, 96
  - rtcdefine.h, 51
- GPIO\_CHANNEL\_0
  - led.h, 31
- GPIO\_CHANNEL\_1
  - led.h, 31
- GPIO\_CHANNEL\_2
  - led.h, 31
- GPIO\_CHANNEL\_3



- led.h, 31
- GPIO\_CHANNEL\_4
  - led.h, 31
- GPIO\_CHANNEL\_5
  - led.h, 31
- H12
  - rtcdefine.h, 46
- heart\_rate\_handle\_table
  - bleSL.cpp, 64
- homepage\_handler
  - http.cpp, 76
  - http.h, 26
- HRS\_IDX\_NB
  - gatts\_table\_creat\_demo.h, 24
- HSPI\_CLK
  - rtcdefine.h, 46
- HSPI\_CS
  - rtcdefine.h, 46
- HSPI\_MISO
  - rtcdefine.h, 47
- HSPI\_MOSI
  - rtcdefine.h, 47
- http.cpp
  - DELIMITER, 75
  - direct\_control\_post\_handler, 76
  - favicon\_ico\_get\_handler, 76
  - homepage\_handler, 76
  - init\_http, 78
  - sch\_data\_post\_handler, 78
  - schedule\_post\_handler, 78
  - schedules\_handler, 79
  - schTokenProcess, 79
  - scripts\_handler, 79
  - styles\_handler, 80
  - time\_post\_handler, 80
- http.h
  - direct\_control\_post\_handler, 26
  - favicon\_ico\_get\_handler, 26
  - homepage\_handler, 26
  - init\_http, 27
  - sch\_data\_post\_handler, 27
  - schedule\_post\_handler, 27
  - schedules\_handler, 28
  - schTokenProcess, 28
  - scripts\_handler, 28
  - styles\_handler, 29
  - time\_post\_handler, 29
- ID
  - Schedule\_Object, 13
- IDREAD
  - rtcdefine.h, 47
- IDWRITE
  - rtcdefine.h, 47
- IDX\_CHAR\_A
  - gatts\_table\_creat\_demo.h, 24
- IDX\_CHAR\_B
  - gatts\_table\_creat\_demo.h, 24
- IDX\_CHAR\_C
  - gatts\_table\_creat\_demo.h, 24
- IDX\_CHAR\_CFG\_A
  - gatts\_table\_creat\_demo.h, 24
- IDX\_CHAR\_VAL\_A
  - gatts\_table\_creat\_demo.h, 24
- IDX\_CHAR\_VAL\_B
  - gatts\_table\_creat\_demo.h, 24
- IDX\_CHAR\_VAL\_C
  - gatts\_table\_creat\_demo.h, 24
- IDX\_SVC
  - gatts\_table\_creat\_demo.h, 24
- include/ArduinoJson-v6.14.1.h, 15
- include/bleSL.h, 15
- include/dawndusk.h, 16
- include/decode\_bluetooth.h, 19
- include/esp8266.h, 23
- include/gatts\_table\_creat\_demo.h, 24
- include/http.h, 25
- include/led.h, 30
- include/main.h, 33
- include/measurement.h, 33
- include/memory.h, 36
- include/pin\_defs.h, 42
- include/rtcdefine.h, 44
- include/schedule\_object.h, 52
- include/scheduler.h, 54
- include/wifi.h, 60
- Init\_Bluetooth
  - bleSL.cpp, 64
  - bleSL.h, 15
- init\_channels
  - led.cpp, 83
  - led.h, 32
- init\_http
  - http.cpp, 78
  - http.h, 27
- init\_memory
  - memory.cpp, 92
  - memory.h, 40
- init\_oc
  - measurement.cpp, 87
  - measurement.h, 34
- init\_schedule
  - scheduler.cpp, 104
  - scheduler.h, 59
- init\_spiffs
  - memory.cpp, 92
- isCurrentFault
  - measurement.cpp, 87
  - measurement.h, 35
- isRGB
  - Schedule\_Object, 13
- isVoltageFault
  - measurement.cpp, 87
  - measurement.h, 35
- it
  - decode\_bluetooth.cpp, 71



- led.cpp
  - channel\_off, [82](#)
  - channel\_on, [83](#)
  - clear\_shutdown, [83](#)
  - init\_channels, [83](#)
  - set\_color, [83](#)
  - shutdown\_outputs, [84](#)
  - shutdown\_status, [84](#)
- led.h
  - channel\_off, [31](#)
  - channel\_on, [32](#)
  - clear\_shutdown, [32](#)
  - GPIO\_CHANNEL\_0, [31](#)
  - GPIO\_CHANNEL\_1, [31](#)
  - GPIO\_CHANNEL\_2, [31](#)
  - GPIO\_CHANNEL\_3, [31](#)
  - GPIO\_CHANNEL\_4, [31](#)
  - GPIO\_CHANNEL\_5, [31](#)
  - init\_channels, [32](#)
  - NUM\_CHANNELS, [31](#)
  - set\_color, [32](#)
  - shutdown\_outputs, [33](#)
- List
  - schedule\_object.h, [53](#)
- LPYR
  - rtcdefine.h, [47](#)
- main.cpp
  - app\_main, [85](#)
- MAX
  - scheduler.cpp, [99](#)
- MAX\_STA\_CONN
  - wifi.cpp, [158](#)
- measurement.cpp
  - clearFaults, [86](#)
  - currentFault, [88](#)
  - init\_oc, [87](#)
  - isCurrentFault, [87](#)
  - isVoltageFault, [87](#)
  - set\_current\_level, [87](#)
  - set\_voltage\_level, [88](#)
  - TAG, [86](#)
  - voltageFault, [88](#)
- measurement.h
  - clearFaults, [34](#)
  - init\_oc, [34](#)
  - isCurrentFault, [35](#)
  - isVoltageFault, [35](#)
  - set\_current\_level, [35](#)
  - set\_voltage\_level, [36](#)
- memory.cpp
  - bSPIFFS, [94](#)
  - clear\_schedule\_data, [90](#)
  - clear\_setting\_data, [90](#)
  - get\_setting\_byte, [90](#)
  - get\_setting\_double, [90](#)
  - get\_setting\_int, [91](#)
  - get\_setting\_string, [91](#)
  - init\_memory, [92](#)
  - init\_spiffs, [92](#)
  - read\_settings\_to\_buffer, [92](#)
  - readNeeded, [95](#)
  - recall\_schedules, [92](#)
  - settingsString, [95](#)
  - store\_schedules, [92](#)
  - store\_setting\_byte, [93](#)
  - store\_setting\_double, [93](#)
  - store\_setting\_int, [94](#)
  - store\_setting\_string, [94](#)
- memory.h
  - clear\_schedule\_data, [38](#)
  - clear\_setting\_data, [38](#)
  - get\_setting\_byte, [38](#)
  - get\_setting\_double, [38](#)
  - get\_setting\_int, [39](#)
  - get\_setting\_string, [39](#)
  - init\_memory, [40](#)
  - recall\_schedules, [40](#)
  - SETTINGS\_BUFFER\_SIZE, [37](#)
  - store\_schedules, [40](#)
  - store\_setting\_byte, [40](#)
  - store\_setting\_double, [41](#)
  - store\_setting\_int, [41](#)
  - store\_setting\_string, [42](#)
- MIN
  - scheduler.cpp, [99](#)
  - wifi.h, [60](#)
- name
  - channel, [7](#)
  - Schedule\_Object, [13](#)
- next
  - Node, [10](#)
- Node, [10](#)
  - next, [10](#)
  - schedule, [10](#)
- NUM\_CHANNELS
  - led.h, [31](#)
- obtain\_time
  - espsntp.cpp, [73](#)
  - espsntp.h, [24](#)
- OC\_ALERT
  - pin\_defs.h, [43](#)
- OC\_ENABLE
  - pin\_defs.h, [43](#)
- OC\_LATCH
  - pin\_defs.h, [43](#)
- OC\_LIMIT
  - pin\_defs.h, [43](#)
- OSCRUN
  - rtcdefine.h, [47](#)
- OSCTRIM
  - rtcdefine.h, [47](#)
- OUT
  - rtcdefine.h, [47](#)
- perm

- gatts\_profile\_inst, 9
- pi
  - dawndusk.h, 17
- pin\_defs.h
  - CH3\_HIGH, 43
  - CH3\_LOW, 43
  - DAC1, 43
  - DAC2, 43
  - OC\_ALERT, 43
  - OC\_ENABLE, 43
  - OC\_LATCH, 43
  - OC\_LIMIT, 43
- planeDist
  - dawndusk.h, 17
- PM
  - rtcdefine.h, 48
- prepare\_buf
  - prepare\_type\_env\_t, 11
- PREPARE\_BUF\_MAX\_SIZE
  - bleSL.cpp, 63
- prepare\_len
  - prepare\_type\_env\_t, 11
- prepare\_type\_env\_t, 11
  - prepare\_buf, 11
  - prepare\_len, 11
- PROFILE\_APP\_IDX
  - bleSL.cpp, 63
- PROFILE\_NUM
  - bleSL.cpp, 63
- property
  - gatts\_profile\_inst, 9
- PWRDNDATE
  - rtcdefine.h, 48
- PWRDNHOUR
  - rtcdefine.h, 48
- PWRDNMIN
  - rtcdefine.h, 48
- PWRDNMONTH
  - rtcdefine.h, 48
- PWRFAIL
  - rtcdefine.h, 48
- r
  - channel, 8
  - Schedule\_Object, 13
- radius
  - dawndusk.h, 17
- read\_settings\_to\_buffer
  - memory.cpp, 92
- readData
  - rtc.cpp, 96
- README.md, 61
- readNeeded
  - memory.cpp, 95
- recall\_schedules
  - memory.cpp, 92
  - memory.h, 40
- repeat\_mask
  - Schedule\_Object, 13
- repeat\_time
  - Schedule\_Object, 14
- rtc
  - rtc.cpp, 97
- rtc.cpp
  - getTime, 96
  - readData, 96
  - rtc, 97
  - rtc\_config, 96
  - RTCHandler, 96
  - setTime, 97
  - ST\_StartRTCHandler, 97
  - writeData, 97
- rtc\_config
  - rtc.cpp, 96
- RTC\_Handle
  - rtcdefine.h, 52
- RTC\_READ
  - rtcdefine.h, 48
- RTC\_TAG
  - rtcdefine.h, 48
- RTC\_UNLOCK
  - rtcdefine.h, 49
- RTC\_WRITE
  - rtcdefine.h, 49
- RTCDATE
  - rtcdefine.h, 49
- rtcdefine.h
  - CLRRAM, 45
  - CONTROL, 46
  - currTime, 52
  - EEREAD, 46
  - EEWRDI, 46
  - EEWREN, 46
  - EEWRITE, 46
  - getTime, 51
  - H12, 46
  - HSPI\_CLK, 46
  - HSPI\_CS, 46
  - HSPI\_MISO, 47
  - HSPI\_MOSI, 47
  - IDREAD, 47
  - IDWRITE, 47
  - LPYR, 47
  - OSCRUN, 47
  - OSCTRIM, 47
  - OUT, 47
  - PM, 48
  - PWRDNDATE, 48
  - PWRDNHOUR, 48
  - PWRDNMIN, 48
  - PWRDNMONTH, 48
  - PWRFAIL, 48
  - RTC\_Handle, 52
  - RTC\_READ, 48
  - RTC\_TAG, 48
  - RTC\_UNLOCK, 49
  - RTC\_WRITE, 49

- RTCDATE, 49
- RTCHandler, 51
- RTCHOUR, 49
- RTCHSEC, 49
- RTCMIN, 49
- RTCMTH, 49
- RTCSEC, 49
- RTCWKDAY, 50
- RTCYEAR, 50
- setTime, 51
- SQWEN, 50
- SRREAD, 50
- SSWRITE, 50
- ST, 50
- ST\_StartRTCHandler, 52
- TRIMSIGN, 50
- VBATEN, 50
- RTCHandler
  - rtc.cpp, 96
  - rtcdefine.h, 51
- RTCHOUR
  - rtcdefine.h, 49
- RTCHSEC
  - rtcdefine.h, 49
- RTCMIN
  - rtcdefine.h, 49
- RTCMTH
  - rtcdefine.h, 49
- RTCSEC
  - rtcdefine.h, 49
- RTCWKDAY
  - rtcdefine.h, 50
- RTCYEAR
  - rtcdefine.h, 50
- SAMPLE\_DEVICE\_NAME
  - bleSL.cpp, 63
- saved\_channel\_num
  - decode\_bluetooth.cpp, 72
- saved\_len
  - decode\_bluetooth.cpp, 72
- saved\_name
  - decode\_bluetooth.cpp, 72
- SCAN\_RSP\_CONFIG\_FLAG
  - bleSL.cpp, 63
- sch\_data\_post\_handler
  - http.cpp, 78
  - http.h, 27
- schedule
  - Node, 10
- schedule\_name
  - decode\_bluetooth.cpp, 72
- Schedule\_Object, 11
  - b, 12
  - brightness, 12
  - dawn, 12
  - duration, 12
  - dusk, 12
  - enabled, 13
  - g, 13
  - ID, 13
  - isRGB, 13
  - name, 13
  - r, 13
  - repeat\_mask, 13
  - repeat\_time, 14
  - start, 14
- schedule\_object
  - schedule\_object.h, 53
- schedule\_object.h
  - List, 53
  - schedule\_object, 53
- schedule\_post\_handler
  - http.cpp, 78
  - http.h, 27
- schedule\_value
  - decode\_bluetooth.cpp, 72
- scheduler.cpp
  - create\_schedule, 99
  - delete\_all\_schedules, 100
  - delete\_schedule\_by\_id, 100
  - delete\_schedule\_by\_name, 100
  - disable\_all\_schedules, 101
  - disable\_schedule\_by\_id, 101
  - disable\_schedule\_by\_name, 101
  - enable\_all\_schedules, 102
  - enable\_schedule\_by\_id, 102
  - enable\_schedule\_by\_name, 102
  - get\_schedule, 103
  - get\_schedule\_names, 103
  - init\_schedule, 104
  - MAX, 99
  - MIN, 99
  - schedules, 104
  - update\_start\_time, 104
- scheduler.h
  - create\_schedule, 55
  - delete\_all\_schedules, 55
  - delete\_schedule\_by\_id, 55
  - delete\_schedule\_by\_name, 56
  - disable\_all\_schedules, 56
  - disable\_schedule\_by\_id, 56
  - disable\_schedule\_by\_name, 57
  - enable\_all\_schedules, 57
  - enable\_schedule\_by\_id, 57
  - enable\_schedule\_by\_name, 58
  - get\_schedule, 58
  - get\_schedule\_names, 59
  - init\_schedule, 59
  - schedules, 59
- schedules
  - decode\_bluetooth.cpp, 72
  - scheduler.cpp, 104
  - scheduler.h, 59
- schedules\_handler
  - http.cpp, 79
  - http.h, 28

schTokenProcess

http.cpp, 79

http.h, 28

scripts\_handler

http.cpp, 79

http.h, 28

sdkconfig.h

CONFIG\_A2D\_INITIAL\_TRACE\_LEVEL, 111

CONFIG\_A2D\_TRACE\_LEVEL\_WARNING, 111

CONFIG\_ADC2\_DISABLE\_DAC, 112

CONFIG\_ADC\_CAL\_EFUSE\_TP\_ENABLE, 112

CONFIG\_ADC\_CAL\_EFUSE\_VREF\_ENABLE,  
112

CONFIG\_ADC\_CAL\_LUT\_ENABLE, 112

CONFIG\_APP\_COMPILE\_TIME\_DATE, 112

CONFIG\_APPL\_INITIAL\_TRACE\_LEVEL, 112

CONFIG\_APPL\_TRACE\_LEVEL\_WARNING, 112

CONFIG\_ARDUINO\_EVENT\_RUN\_CORE1, 112

CONFIG\_ARDUINO\_EVENT\_RUNNING\_CORE,  
113

CONFIG\_ARDUINO\_RUNNING\_CORE, 113

CONFIG\_ARDUINO\_UDP\_RUN\_CORE1, 113

CONFIG\_ARDUINO\_UDP\_RUNNING\_CORE, 113

CONFIG\_AUTOSTART\_ARDUINO, 113

CONFIG\_AVCT\_INITIAL\_TRACE\_LEVEL, 113

CONFIG\_AVCT\_TRACE\_LEVEL\_WARNING, 113

CONFIG\_AVDT\_INITIAL\_TRACE\_LEVEL, 113

CONFIG\_AVDT\_TRACE\_LEVEL\_WARNING, 114

CONFIG\_AVRC\_INITIAL\_TRACE\_LEVEL, 114

CONFIG\_AVRC\_TRACE\_LEVEL\_WARNING, 114

CONFIG\_AWS\_IOT\_MQTT\_HOST, 114

CONFIG\_AWS\_IOT\_MQTT\_MAX\_RECONNECT\_WAIT\_INTERVAL,  
114

CONFIG\_AWS\_IOT\_MQTT\_MIN\_RECONNECT\_WAIT\_INTERVAL,  
114

CONFIG\_AWS\_IOT\_MQTT\_NUM\_SUBSCRIBE\_HANDLERS, 119

CONFIG\_AWS\_IOT\_MQTT\_PORT, 114

CONFIG\_AWS\_IOT\_MQTT\_RX\_BUF\_LEN, 115

CONFIG\_AWS\_IOT\_MQTT\_TX\_BUF\_LEN, 115

CONFIG\_AWS\_IOT\_SDK, 115

CONFIG\_AWS\_IOT\_SHADOW\_MAX\_JSON\_TOKEN\_EXPECTED,  
115

CONFIG\_AWS\_IOT\_SHADOW\_MAX\_SHADOW\_TOPIC\_LENGTH\_WITHOUT\_ASYNCHRONOUS,  
115

CONFIG\_AWS\_IOT\_SHADOW\_MAX\_SIMULTANEOUS\_ACTIONS,  
115

CONFIG\_AWS\_IOT\_SHADOW\_MAX\_SIMULTANEOUS\_THING\_NAMES,  
115

CONFIG\_AWS\_IOT\_SHADOW\_MAX\_SIZE\_OF\_THING\_NAME,  
115

CONFIG\_AWS\_IOT\_SHADOW\_MAX\_SIZE\_OF\_UNIQUE\_IDENTIFIER,  
116

CONFIG\_BLE\_ADV\_REPORT\_DISCARD\_THRSHOLD, 116

CONFIG\_BLE\_ADV\_REPORT\_FLOW\_CONTROL\_NUM, 116

CONFIG\_BLE\_ADV\_REPORT\_FLOW\_CONTROL\_SUPPORT, 116

116

CONFIG\_BLE\_ESTABLISH\_LINK\_CONNECTION\_TIMEOUT,  
116

CONFIG\_BLE\_SCAN\_DUPLICATE, 116

CONFIG\_BLE\_SMP\_ENABLE, 116

CONFIG\_BLUEDROID\_ENABLED, 116

CONFIG\_BLUEDROID\_PINNED\_TO\_CORE, 117

CONFIG\_BLUEDROID\_PINNED\_TO\_CORE\_0,  
117

CONFIG\_BLUFI\_INITIAL\_TRACE\_LEVEL, 117

CONFIG\_BLUFI\_TRACE\_LEVEL\_WARNING, 117

CONFIG\_BNEP\_INITIAL\_TRACE\_LEVEL, 117

CONFIG\_BNEP\_TRACE\_LEVEL\_WARNING, 117

CONFIG\_BOOTLOADER\_VDDSDIO\_BOOST\_1\_9V,  
117

CONFIG\_BOOTLOADER\_WDT\_ENABLE, 117

CONFIG\_BOOTLOADER\_WDT\_TIME\_MS, 118

CONFIG\_BROWNOUT\_DET, 118

CONFIG\_BROWNOUT\_DET\_LVL, 118

CONFIG\_BROWNOUT\_DET\_LVL\_SEL\_0, 118

CONFIG\_BT\_ACL\_CONNECTIONS, 118

CONFIG\_BT\_ENABLED, 118

CONFIG\_BT\_RESERVE\_DRAM, 118

CONFIG\_BTC\_INITIAL\_TRACE\_LEVEL, 118

CONFIG\_BTC\_TASK\_STACK\_SIZE, 119

CONFIG\_BTC\_TRACE\_LEVEL\_WARNING, 119

CONFIG\_BTDM\_CONTROLLER\_BLE\_MAX\_CONN,

119

CONFIG\_BTDM\_CONTROLLER\_BLE\_MAX\_CONN\_EFF,

119

CONFIG\_BTDM\_CONTROLLER\_BR\_EDR\_MAX\_ACL\_CONN\_EFF,

CONFIG\_BTDM\_CONTROLLER\_BR\_EDR\_MAX\_SYNC\_CONN\_EFF,

CONFIG\_BTDM\_CONTROLLER\_HCI\_MODE\_VHCI,

CONFIG\_BTDM\_CONTROLLER\_MODE\_BLE\_ONLY,

119

CONFIG\_BTDM\_CONTROLLER\_PINNED\_TO\_CORE,

120

CONFIG\_BTDM\_CONTROLLER\_PINNED\_TO\_CORE\_0,

120

CONFIG\_BTIF\_INITIAL\_TRACE\_LEVEL, 120

CONFIG\_BTIF\_TRACE\_LEVEL\_WARNING, 120

CONFIG\_BTM\_INITIAL\_TRACE\_LEVEL, 120

CONFIG\_BTM\_TRACE\_LEVEL\_WARNING, 120

CONFIG\_BTU\_TASK\_STACK\_SIZE, 120

CONFIG\_CONSOLE\_UART\_BAUDRATE, 120

CONFIG\_CONSOLE\_UART\_DEFAULT, 121

CONFIG\_CONSOLE\_UART\_NUM, 121

CONFIG\_DMA\_RX\_BUF\_NUM, 121

CONFIG\_DMA\_TX\_BUF\_NUM, 121

CONFIG\_DUPLICATE\_SCAN\_CACHE\_SIZE, 121

CONFIG\_EFUSE\_CODE\_SCHEME\_COMPAT\_3\_4,

121

CONFIG\_EFUSE\_MAX\_BLK\_LEN, 121

CONFIG\_EMAC\_CHECK\_LINK\_PERIOD\_MS,

121

- CONFIG\_EMAC\_TASK\_PRIORITY, [122](#)
- CONFIG\_EMAC\_TASK\_STACK\_SIZE, [122](#)
- CONFIG\_ENABLE\_ARDUINO\_DEPENDS, [122](#)
- CONFIG\_ESP32\_APPTRACE\_DEST\_NONE, [122](#)
- CONFIG\_ESP32\_APPTRACE\_LOCK\_ENABLE, [122](#)
- CONFIG\_ESP32\_DEBUG\_OCDAWARE, [122](#)
- CONFIG\_ESP32\_DEBUG\_STUBS\_ENABLE, [122](#)
- CONFIG\_ESP32\_DEEP\_SLEEP\_WAKEUP\_DELAY, [122](#)
- CONFIG\_ESP32\_DEFAULT\_CPU\_FREQ\_160, [123](#)
- CONFIG\_ESP32\_DEFAULT\_CPU\_FREQ\_MHZ, [123](#)
- CONFIG\_ESP32\_DEFAULT\_PTHREAD\_CORE\_NO\_AFFINITY, [123](#)
- CONFIG\_ESP32\_DPORT\_WORKAROUND, [123](#)
- CONFIG\_ESP32\_ENABLE\_COREDUMP\_TO\_NONE, [123](#)
- CONFIG\_ESP32\_PANIC\_PRINT\_REBOOT, [123](#)
- CONFIG\_ESP32\_PHY\_CALIBRATION\_AND\_DATA\_STORAGE, [123](#)
- CONFIG\_ESP32\_PHY\_MAX\_TX\_POWER, [123](#)
- CONFIG\_ESP32\_PHY\_MAX\_WIFI\_TX\_POWER, [124](#)
- CONFIG\_ESP32\_PTHREAD\_TASK\_CORE\_DEFAULT, [124](#)
- CONFIG\_ESP32\_PTHREAD\_TASK\_NAME\_DEFAULT, [124](#)
- CONFIG\_ESP32\_PTHREAD\_TASK\_PRIO\_DEFAULT, [124](#)
- CONFIG\_ESP32\_PTHREAD\_TASK\_STACK\_SIZE\_DEFAULT, [124](#)
- CONFIG\_ESP32\_REV\_MIN, [124](#)
- CONFIG\_ESP32\_REV\_MIN\_0, [124](#)
- CONFIG\_ESP32\_RTC\_CLK\_CAL\_CYCLES, [124](#)
- CONFIG\_ESP32\_RTC\_CLOCK\_SOURCE\_INTERNAL\_RC, [125](#)
- CONFIG\_ESP32\_TIME\_SYSCALL\_USE\_RTC\_FRC1, [125](#)
- CONFIG\_ESP32\_WIFI\_AMPDU\_RX\_ENABLED, [125](#)
- CONFIG\_ESP32\_WIFI\_AMPDU\_TX\_ENABLED, [125](#)
- CONFIG\_ESP32\_WIFI\_DYNAMIC\_RX\_BUFFER\_NUM, [125](#)
- CONFIG\_ESP32\_WIFI\_DYNAMIC\_TX\_BUFFER, [125](#)
- CONFIG\_ESP32\_WIFI\_DYNAMIC\_TX\_BUFFER\_NUM, [125](#)
- CONFIG\_ESP32\_WIFI\_IRAM\_OPT, [125](#)
- CONFIG\_ESP32\_WIFI\_MGMT\_SBUF\_NUM, [126](#)
- CONFIG\_ESP32\_WIFI\_NVS\_ENABLED, [126](#)
- CONFIG\_ESP32\_WIFI\_RX\_BA\_WIN, [126](#)
- CONFIG\_ESP32\_WIFI\_SOFTAP\_BEACON\_MAX\_LEN, [126](#)
- CONFIG\_ESP32\_WIFI\_STATIC\_RX\_BUFFER\_NUM, [126](#)
- CONFIG\_ESP32\_WIFI\_TASK\_PINNED\_TO\_CORE\_0, [126](#)
- CONFIG\_ESP32\_WIFI\_TX\_BA\_WIN, [126](#)
- CONFIG\_ESP32\_WIFI\_TX\_BUFFER\_TYPE, [126](#)
- CONFIG\_ESP32\_XTAL\_FREQ, [127](#)
- CONFIG\_ESP32\_XTAL\_FREQ\_40, [127](#)
- CONFIG\_ESP\_ERR\_TO\_NAME\_LOOKUP, [127](#)
- CONFIG\_ESP\_GRATUITOUS\_ARP, [127](#)
- CONFIG\_ESP\_HTTP\_CLIENT\_ENABLE\_HTTPS, [127](#)
- CONFIG\_ESPTOOLPY\_AFTER, [127](#)
- CONFIG\_ESPTOOLPY\_AFTER\_RESET, [127](#)
- CONFIG\_ESPTOOLPY\_BAUD, [127](#)
- CONFIG\_ESPTOOLPY\_BAUD\_115200B, [128](#)
- CONFIG\_ESPTOOLPY\_BAUD\_OTHER\_VAL, [128](#)
- CONFIG\_ESPTOOLPY\_BEFORE, [128](#)
- CONFIG\_ESPTOOLPY\_BEFORE\_RESET, [128](#)
- CONFIG\_ESPTOOLPY\_COMPRESSED, [128](#)
- CONFIG\_ESPTOOLPY\_FLASHFREQ, [128](#)
- CONFIG\_ESPTOOLPY\_FLASHFREQ\_40M, [128](#)
- CONFIG\_ESPTOOLPY\_FLASHMODE, [128](#)
- CONFIG\_ESPTOOLPY\_FLASHSIZE, [129](#)
- CONFIG\_ESPTOOLPY\_FLASHSIZE\_2MB, [129](#)
- CONFIG\_ESPTOOLPY\_FLASHSIZE\_DETECT, [129](#)
- CONFIG\_ESPTOOLPY\_PORT, [129](#)
- CONFIG\_FATFS\_CODEPAGE, [129](#)
- CONFIG\_FATFS\_CODEPAGE\_437, [129](#)
- CONFIG\_FATFS\_FS\_LOCK, [129](#)
- CONFIG\_FATFS\_LFN\_NONE, [129](#)
- CONFIG\_FATFS\_PER\_FILE\_CACHE, [130](#)
- CONFIG\_FATFS\_TIMEOUT\_MS, [130](#)
- CONFIG\_FLASHMODE\_DIO, [130](#)
- CONFIG\_FOUR\_UNIVERSAL\_MAC\_ADDRESS, [130](#)
- CONFIG\_FREERTOS\_ASSERT\_FAIL\_ABORT, [130](#)
- CONFIG\_FREERTOS\_ASSERT\_ON\_UNTESTED\_FUNCTION, [130](#)
- CONFIG\_FREERTOS\_CHECK\_MUTEX\_GIVEN\_BY\_OWNER, [130](#)
- CONFIG\_FREERTOS\_CHECK\_STACKOVERFLOW\_CANARY, [130](#)
- CONFIG\_FREERTOS\_CORETIMER\_0, [131](#)
- CONFIG\_FREERTOS\_HZ, [131](#)
- CONFIG\_FREERTOS\_IDLE\_TASK\_STACKSIZE, [131](#)
- CONFIG\_FREERTOS\_INTERRUPT\_BACKTRACE, [131](#)
- CONFIG\_FREERTOS\_ISR\_STACKSIZE, [131](#)
- CONFIG\_FREERTOS\_MAX\_TASK\_NAME\_LEN, [131](#)
- CONFIG\_FREERTOS\_NO\_AFFINITY, [131](#)
- CONFIG\_FREERTOS\_QUEUE\_REGISTRY\_SIZE, [131](#)
- CONFIG\_FREERTOS\_TASK\_FUNCTION\_WRAPPER, [132](#)
- CONFIG\_FREERTOS\_THREAD\_LOCAL\_STORAGE\_POINTERS, [132](#)

- 132
- CONFIG\_GAP\_INITIAL\_TRACE\_LEVEL, 132
- CONFIG\_GAP\_TRACE\_LEVEL\_WARNING, 132
- CONFIG\_GARP\_TMR\_INTERVAL, 132
- CONFIG\_GATT\_INITIAL\_TRACE\_LEVEL, 132
- CONFIG\_GATT\_TRACE\_LEVEL\_WARNING, 132
- CONFIG\_GATTC\_ENABLE, 132
- CONFIG\_GATTS\_ENABLE, 133
- CONFIG\_GATTS\_SEND\_SERVICE\_CHANGE\_AUTO, 133
- CONFIG\_GATTS\_SEND\_SERVICE\_CHANGE\_MODE, 133
- CONFIG\_HCI\_INITIAL\_TRACE\_LEVEL, 133
- CONFIG\_HCI\_TRACE\_LEVEL\_WARNING, 133
- CONFIG\_HEAP\_POISONING\_DISABLED, 133
- CONFIG\_HID\_INITIAL\_TRACE\_LEVEL, 133
- CONFIG\_HID\_TRACE\_LEVEL\_WARNING, 133
- CONFIG\_HTTPD\_ERR\_RESP\_NO\_DELAY, 134
- CONFIG\_HTTPD\_MAX\_REQ\_HDR\_LEN, 134
- CONFIG\_HTTPD\_MAX\_URI\_LEN, 134
- CONFIG\_HTTPD\_PURGE\_BUF\_LEN, 134
- CONFIG\_IDF\_TARGET, 134
- CONFIG\_IDF\_TARGET\_ESP32, 134
- CONFIG\_INT\_WDT, 134
- CONFIG\_INT\_WDT\_CHECK\_CPU1, 134
- CONFIG\_INT\_WDT\_TIMEOUT\_MS, 135
- CONFIG\_IP\_LOST\_TIMER\_INTERVAL, 135
- CONFIG\_IPC\_TASK\_STACK\_SIZE, 135
- CONFIG\_L2CAP\_INITIAL\_TRACE\_LEVEL, 135
- CONFIG\_L2CAP\_TRACE\_LEVEL\_WARNING, 135
- CONFIG\_LIBSODIUM\_USE\_MBEDTLS\_SHA, 135
- CONFIG\_LOG\_BOOTLOADER\_LEVEL, 135
- CONFIG\_LOG\_BOOTLOADER\_LEVEL\_INFO, 135
- CONFIG\_LOG\_COLORS, 136
- CONFIG\_LOG\_DEFAULT\_LEVEL, 136
- CONFIG\_LOG\_DEFAULT\_LEVEL\_INFO, 136
- CONFIG\_LWIP\_DHCP\_DOES\_ARP\_CHECK, 136
- CONFIG\_LWIP\_DHCP\_MAX\_NTP\_SERVERS, 136
- CONFIG\_LWIP\_DHCP\_LEASE\_UNIT, 136
- CONFIG\_LWIP\_DHCP\_MAX\_STATION\_NUM, 136
- CONFIG\_LWIP\_LOOPBACK\_MAX\_PBUFS, 136
- CONFIG\_LWIP\_MAX\_ACTIVE\_TCP, 137
- CONFIG\_LWIP\_MAX\_LISTENING\_TCP, 137
- CONFIG\_LWIP\_MAX\_RAW\_PCBS, 137
- CONFIG\_LWIP\_MAX\_SOCKETS, 137
- CONFIG\_LWIP\_MAX\_UDP\_PCBS, 137
- CONFIG\_LWIP\_NETIF\_LOOPBACK, 137
- CONFIG\_LWIP\_SO\_REUSE, 137
- CONFIG\_LWIP\_SO\_REUSE\_RXTOALL, 137
- CONFIG\_MAIN\_TASK\_STACK\_SIZE, 138
- CONFIG\_MAKE\_WARN\_UNDEFINED\_VARIABLES, 138
- CONFIG\_MB\_CONTROLLER\_NOTIFY\_QUEUE\_SIZE, 138
- CONFIG\_MB\_CONTROLLER\_NOTIFY\_TIMEOUT, 138
- CONFIG\_MB\_CONTROLLER\_STACK\_SIZE, 138
- CONFIG\_MB\_EVENT\_QUEUE\_TIMEOUT, 138
- CONFIG\_MB\_QUEUE\_LENGTH, 138
- CONFIG\_MB\_SERIAL\_BUF\_SIZE, 138
- CONFIG\_MB\_SERIAL\_TASK\_PRIO, 139
- CONFIG\_MB\_SERIAL\_TASK\_STACK\_SIZE, 139
- CONFIG\_MB\_TIMER\_GROUP, 139
- CONFIG\_MB\_TIMER\_INDEX, 139
- CONFIG\_MB\_TIMER\_PORT\_ENABLED, 139
- CONFIG\_MBEDTLS\_AES\_C, 139
- CONFIG\_MBEDTLS\_CCM\_C, 139
- CONFIG\_MBEDTLS\_ECDH\_C, 139
- CONFIG\_MBEDTLS\_ECDSA\_C, 140
- CONFIG\_MBEDTLS\_ECP\_C, 140
- CONFIG\_MBEDTLS\_ECP\_DP\_BP256R1\_ENABLED, 140
- CONFIG\_MBEDTLS\_ECP\_DP\_BP384R1\_ENABLED, 140
- CONFIG\_MBEDTLS\_ECP\_DP\_BP512R1\_ENABLED, 140
- CONFIG\_MBEDTLS\_ECP\_DP\_CURVE25519\_ENABLED, 140
- CONFIG\_MBEDTLS\_ECP\_DP\_SECP192K1\_ENABLED, 140
- CONFIG\_MBEDTLS\_ECP\_DP\_SECP192R1\_ENABLED, 140
- CONFIG\_MBEDTLS\_ECP\_DP\_SECP224K1\_ENABLED, 141
- CONFIG\_MBEDTLS\_ECP\_DP\_SECP224R1\_ENABLED, 141
- CONFIG\_MBEDTLS\_ECP\_DP\_SECP256K1\_ENABLED, 141
- CONFIG\_MBEDTLS\_ECP\_DP\_SECP256R1\_ENABLED, 141
- CONFIG\_MBEDTLS\_ECP\_DP\_SECP384R1\_ENABLED, 141
- CONFIG\_MBEDTLS\_ECP\_DP\_SECP521R1\_ENABLED, 141
- CONFIG\_MBEDTLS\_ECP\_NIST\_OPTIM, 141
- CONFIG\_MBEDTLS\_GCM\_C, 141
- CONFIG\_MBEDTLS\_HARDWARE\_AES, 142
- CONFIG\_MBEDTLS\_HAVE\_TIME, 142
- CONFIG\_MBEDTLS\_INTERNAL\_MEM\_ALLOC, 142
- CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_DHE\_RSA, 142
- CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_ECDH\_ECDSA, 142
- CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_ECDH\_RSA, 142
- CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_ECDHE\_ECDSA, 142
- CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_ECDHE\_RSA, 142



- CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_ELLIPTIC\_CURVE, 143
- CONFIG\_MBEDTLS\_KEY\_EXCHANGE\_RSA, 143
- CONFIG\_MBEDTLS\_PEM\_PARSE\_C, 143
- CONFIG\_MBEDTLS\_PEM\_WRITE\_C, 143
- CONFIG\_MBEDTLS\_RC4\_DISABLED, 143
- CONFIG\_MBEDTLS\_SSL\_ALPN, 143
- CONFIG\_MBEDTLS\_SSL\_MAX\_CONTENT\_LEN, 143
- CONFIG\_MBEDTLS\_SSL\_PROTO\_TLS1, 143
- CONFIG\_MBEDTLS\_SSL\_PROTO\_TLS1\_1, 144
- CONFIG\_MBEDTLS\_SSL\_PROTO\_TLS1\_2, 144
- CONFIG\_MBEDTLS\_SSL\_RENEGOTIATION, 144
- CONFIG\_MBEDTLS\_SSL\_SESSION\_TICKETS, 144
- CONFIG\_MBEDTLS\_TLS\_CLIENT, 144
- CONFIG\_MBEDTLS\_TLS\_ENABLED, 144
- CONFIG\_MBEDTLS\_TLS\_SERVER, 144
- CONFIG\_MBEDTLS\_TLS\_SERVER\_AND\_CLIENT, 144
- CONFIG\_MBEDTLS\_X509\_CRL\_PARSE\_C, 145
- CONFIG\_MBEDTLS\_X509\_CSR\_PARSE\_C, 145
- CONFIG\_MCA\_INITIAL\_TRACE\_LEVEL, 145
- CONFIG\_MCA\_TRACE\_LEVEL\_WARNING, 145
- CONFIG\_MDNS\_MAX\_SERVICES, 145
- CONFIG\_MONITOR\_BAUD, 145
- CONFIG\_MONITOR\_BAUD\_115200B, 145
- CONFIG\_MONITOR\_BAUD\_OTHER\_VAL, 145
- CONFIG\_MQTT\_PROTOCOL\_311, 146
- CONFIG\_MQTT\_TRANSPORT\_SSL, 146
- CONFIG\_MQTT\_TRANSPORT\_WEBSOCKET, 146
- CONFIG\_MQTT\_TRANSPORT\_WEBSOCKET\_SECURE, 146
- CONFIG\_NEWLIB\_STDIN\_LINE\_ENDING\_CR, 146
- CONFIG\_NEWLIB\_STDOUT\_LINE\_ENDING\_CRLF, 146
- CONFIG\_NUMBER\_OF\_UNIVERSAL\_MAC\_ADDRESS, 146
- CONFIG\_OPENSSL\_ASSERT\_DO\_NOTHING, 146
- CONFIG\_OPTIMIZATION\_ASSERTIONS\_ENABLED, 147
- CONFIG\_OPTIMIZATION\_LEVEL\_DEBUG, 147
- CONFIG\_OSI\_INITIAL\_TRACE\_LEVEL, 147
- CONFIG\_OSI\_TRACE\_LEVEL\_WARNING, 147
- CONFIG\_PAN\_INITIAL\_TRACE\_LEVEL, 147
- CONFIG\_PAN\_TRACE\_LEVEL\_WARNING, 147
- CONFIG\_PARTITION\_TABLE\_CUSTOM\_FILENAME, 147
- CONFIG\_PARTITION\_TABLE\_FILENAME, 147
- CONFIG\_PARTITION\_TABLE\_MD5, 148
- CONFIG\_PARTITION\_TABLE\_OFFSET, 148
- CONFIG\_PARTITION\_TABLE\_SINGLE\_APP, 148
- CONFIG\_PTHREAD\_STACK\_MIN, 148
- CONFIG\_PYTHON, 148
- CONFIG\_REDUCE\_PHY\_TX\_POWER, 148
- CONFIG\_RFCOMM\_INITIAL\_TRACE\_LEVEL, 148
- CONFIG\_RFCOMM\_TRACE\_LEVEL\_WARNING, 148
- CONFIG\_SCAN\_DUPLICATE\_BY\_DEVICE\_ADDR, 149
- CONFIG\_SCAN\_DUPLICATE\_TYPE, 149
- CONFIG\_SDP\_INITIAL\_TRACE\_LEVEL, 149
- CONFIG\_SDP\_TRACE\_LEVEL\_WARNING, 149
- CONFIG\_SMP\_ENABLE, 149
- CONFIG\_SMP\_INITIAL\_TRACE\_LEVEL, 149
- CONFIG\_SMP\_TRACE\_LEVEL\_WARNING, 149
- CONFIG\_SPI\_FLASH\_ERASE\_YIELD\_DURATION\_MS, 149
- CONFIG\_SPI\_FLASH\_ERASE\_YIELD\_TICKS, 150
- CONFIG\_SPI\_FLASH\_ROM\_DRIVER\_PATCH, 150
- CONFIG\_SPI\_FLASH\_WRITING\_DANGEROUS\_REGIONS\_ABORT, 150
- CONFIG\_SPI\_FLASH\_YIELD\_DURING\_ERASE, 150
- CONFIG\_SPI\_MASTER\_ISR\_IN\_IRAM, 150
- CONFIG\_SPI\_SLAVE\_ISR\_IN\_IRAM, 150
- CONFIG\_SPIFFS\_CACHE, 150
- CONFIG\_SPIFFS\_CACHE\_WR, 150
- CONFIG\_SPIFFS\_GC\_MAX\_RUNS, 151
- CONFIG\_SPIFFS\_MAX\_PARTITIONS, 151
- CONFIG\_SPIFFS\_META\_LENGTH, 151
- CONFIG\_SPIFFS\_OBJ\_NAME\_LEN, 151
- CONFIG\_SPIFFS\_PAGE\_CHECK, 151
- CONFIG\_SPIFFS\_PAGE\_SIZE, 151
- CONFIG\_SPIFFS\_USE\_MAGIC, 151
- CONFIG\_SPIFFS\_USE\_MAGIC\_LENGTH, 151
- CONFIG\_SPIFFS\_USE\_MTIME, 152
- CONFIG\_STACK\_CHECK\_NONE, 152
- CONFIG\_SUPPORT\_TERMIOS, 152
- CONFIG\_SUPPRESS\_SELECT\_DEBUG\_OUTPUT, 152
- CONFIG\_SW\_COEXIST\_ENABLE, 152
- CONFIG\_SW\_COEXIST\_PREFERENCE\_BALANCE, 152
- CONFIG\_SW\_COEXIST\_PREFERENCE\_VALUE, 152
- CONFIG\_SYSTEM\_EVENT\_QUEUE\_SIZE, 152
- CONFIG\_SYSTEM\_EVENT\_TASK\_STACK\_SIZE, 153
- CONFIG\_TASK\_WDT, 153
- CONFIG\_TASK\_WDT\_CHECK\_IDLE\_TASK\_CPU0, 153
- CONFIG\_TASK\_WDT\_CHECK\_IDLE\_TASK\_CPU1, 153
- CONFIG\_TASK\_WDT\_TIMEOUT\_S, 153
- CONFIG\_TCP\_MAXRTX, 153
- CONFIG\_TCP\_MSL, 153
- CONFIG\_TCP\_MSS, 153
- CONFIG\_TCP\_OVERSIZE\_MSS, 154

- CONFIG\_TCP\_QUEUE\_OOSEQ, [154](#)
- CONFIG\_TCP\_RECVMBOX\_SIZE, [154](#)
- CONFIG\_TCP\_SND\_BUF\_DEFAULT, [154](#)
- CONFIG\_TCP\_SYNMAXRTX, [154](#)
- CONFIG\_TCP\_WND\_DEFAULT, [154](#)
- CONFIG\_TCPIP\_LWIP, [154](#)
- CONFIG\_TCPIP\_RECVMBOX\_SIZE, [154](#)
- CONFIG\_TCPIP\_TASK\_AFFINITY, [155](#)
- CONFIG\_TCPIP\_TASK\_AFFINITY\_NO\_AFFINITY, [155](#)
- CONFIG\_TCPIP\_TASK\_STACK\_SIZE, [155](#)
- CONFIG\_TIMER\_QUEUE\_LENGTH, [155](#)
- CONFIG\_TIMER\_TASK\_PRIORITY, [155](#)
- CONFIG\_TIMER\_TASK\_STACK\_DEPTH, [155](#)
- CONFIG\_TIMER\_TASK\_STACK\_SIZE, [155](#)
- CONFIG\_TOOLPREFIX, [155](#)
- CONFIG\_TRACEMEM\_RESERVE\_DRAM, [156](#)
- CONFIG\_UDP\_RECVMBOX\_SIZE, [156](#)
- CONFIG\_ULP\_COPROC\_RESERVE\_MEM, [156](#)
- CONFIG\_UNITY\_ENABLE\_DOUBLE, [156](#)
- CONFIG\_UNITY\_ENABLE\_FLOAT, [156](#)
- CONFIG\_UNITY\_ENABLE\_IDF\_TEST\_RUNNER, [156](#)
- CONFIG\_WIFI\_PROV\_SCAN\_MAX\_ENTRIES, [156](#)
- CONFIG\_WL\_SECTOR\_SIZE, [156](#)
- CONFIG\_WL\_SECTOR\_SIZE\_4096, [157](#)
- service\_handle
  - gatts\_profile\_inst, [9](#)
- service\_id
  - gatts\_profile\_inst, [10](#)
- set\_color
  - led.cpp, [83](#)
  - led.h, [32](#)
- set\_current\_level
  - measurement.cpp, [87](#)
  - measurement.h, [35](#)
- set\_schedule\_read
  - decode\_bluetooth.cpp, [70](#)
  - decode\_bluetooth.h, [22](#)
- set\_time
  - espsntp.cpp, [73](#)
  - espsntp.h, [24](#)
- set\_voltage\_level
  - measurement.cpp, [88](#)
  - measurement.h, [36](#)
- setTime
  - rtc.cpp, [97](#)
  - rtcdefine.h, [51](#)
- SETTINGS\_BUFFER\_SIZE
  - memory.h, [37](#)
- settingsString
  - memory.cpp, [95](#)
- shutdown\_outputs
  - led.cpp, [84](#)
  - led.h, [33](#)
- shutdown\_status
  - led.cpp, [84](#)
- sign
  - dawndusk.cpp, [66](#)
  - dawndusk.h, [18](#)
- SQWEN
  - rtcdefine.h, [50](#)
- src/bleSL.cpp, [61](#)
- src/dawndusk.cpp, [65](#)
- src/decode\_bluetooth.cpp, [67](#)
- src/espsntp.cpp, [73](#)
- src/http.cpp, [74](#)
- src/led.cpp, [81](#)
- src/main.cpp, [84](#)
- src/measurement.cpp, [85](#)
- src/memory.cpp, [88](#)
- src/rtc.cpp, [95](#)
- src/scheduler.cpp, [98](#)
- src/sdkconfig.h, [105](#)
- src/wifi.cpp, [157](#)
- SRREAD
  - rtcdefine.h, [50](#)
- SSWRITE
  - rtcdefine.h, [50](#)
- ST
  - rtcdefine.h, [50](#)
- ST\_StartRTCHandler
  - rtc.cpp, [97](#)
  - rtcdefine.h, [52](#)
- start
  - Schedule\_Object, [14](#)
- start\_schedule\_read
  - decode\_bluetooth.cpp, [71](#)
  - decode\_bluetooth.h, [23](#)
- state
  - decode\_bluetooth.cpp, [72](#)
- store\_schedules
  - memory.cpp, [92](#)
  - memory.h, [40](#)
- store\_setting\_byte
  - memory.cpp, [93](#)
  - memory.h, [40](#)
- store\_setting\_double
  - memory.cpp, [93](#)
  - memory.h, [41](#)
- store\_setting\_int
  - memory.cpp, [94](#)
  - memory.h, [41](#)
- store\_setting\_string
  - memory.cpp, [94](#)
  - memory.h, [42](#)
- styles\_handler
  - http.cpp, [80](#)
  - http.h, [29](#)
- SVC\_INST\_ID
  - bleSL.cpp, [64](#)
- TAG
  - measurement.cpp, [86](#)
- time\_post\_handler
  - http.cpp, [80](#)



- http.h, [29](#)
- TRIMSIGN
  - rtcdefine.h, [50](#)
- update\_start\_time
  - scheduler.cpp, [104](#)
- VBATEN
  - rtcdefine.h, [50](#)
- voltageFault
  - measurement.cpp, [88](#)
- wifi.cpp
  - ESP\_WIFI\_PASS, [158](#)
  - ESP\_WIFI\_SSID, [158](#)
  - MAX\_STA\_CONN, [158](#)
  - WIFI\_CONNECTED\_BIT, [159](#)
  - wifi\_init\_softap, [158](#)
  - wifi\_init\_sta, [159](#)
- wifi.h
  - MIN, [60](#)
  - wifi\_init\_softap, [60](#)
  - wifi\_init\_sta, [61](#)
- WIFI\_CONNECTED\_BIT
  - wifi.cpp, [159](#)
- wifi\_init\_softap
  - wifi.cpp, [158](#)
  - wifi.h, [60](#)
- wifi\_init\_sta
  - wifi.cpp, [159](#)
  - wifi.h, [61](#)
- writeData
  - rtc.cpp, [97](#)