

CIS350 Release 1 Documentation

Generated by Doxygen 1.9.7

1 File Index	1
1.1 File List	1
2 File Documentation	3
2.1 lock/main/lock_main.c File Reference	3
2.1.1 Macro Definition Documentation	5
2.1.1.1 CONFIG_BROKER_URL	5
2.1.1.2 LCD_DB4	5
2.1.1.3 LCD_DB5	5
2.1.1.4 LCD_DB6	5
2.1.1.5 LCD_DB7	5
2.1.1.6 LCD_Enable	5
2.1.1.7 LCD_RS	5
2.1.1.8 LED_PIN	5
2.1.1.9 LEDC_CHANNEL	6
2.1.1.10 LEDC_DUTY_LOCKED	6
2.1.1.11 LEDC_DUTY_RES	6
2.1.1.12 LEDC_DUTY_UNLOCKED	6
2.1.1.13 LEDC_FREQUENCY	6
2.1.1.14 LEDC_MODE	6
2.1.1.15 LEDC_OUTPUT_IO	6
2.1.1.16 LEDC_TIMER	6
2.1.1.17 LOCK_STATUS_TOPIC	6
2.1.1.18 MAX_STRING_SIZE	6
2.1.1.19 NUMBER_OF_STRING	7
2.1.1.20 PIN_OUTPUT_TOPIC	7
2.1.1.21 SERVO_PIN	7
2.1.2 Function Documentation	7
2.1.2.1 app_main()	7
2.1.2.2 changeScreenStateLCD()	7
2.1.2.3 checkPin()	7
2.1.2.4 commandWrite()	8
2.1.2.5 dataWrite()	8
2.1.2.6 initLCD()	8
2.1.2.7 initSequenceLCD()	8
2.1.2.8 ledBlink()	8
2.1.2.9 lockBolt()	8
2.1.2.10 lockInit()	9
2.1.2.11 mqtt_pin_to_int_array()	9
2.1.2.12 printDeviceInfo()	9
2.1.2.13 printToLCD()	9
2.1.2.14 pulseEnable()	9

2.1.2.15 push_byte()	9
2.1.2.16 push_nibble()	10
2.1.2.17 unlockBolt()	10
2.1.2.18 writeEnterPinScreen()	10
2.1.2.19 writeLockScreen()	10
2.1.2.20 writeUnlockScreen()	10
2.1.3 Variable Documentation	11
2.1.3.1 arr	11
2.1.3.2 client	11
2.1.3.3 mqtt_cfg	11
2.1.3.4 pin	11
2.1.3.5 pinSize	11
Index	13

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

lock/main/ lock_main.c	3
---	---

Chapter 2

File Documentation

2.1 lock/main/lock_main.c File Reference

```
#include <stdio.h>
#include <stdbool.h>
#include <stdint.h>
#include <stddef.h>
#include <string.h>
#include <driver/gpio.h>
#include "esp_wifi.h"
#include "esp_system.h"
#include "nvs_flash.h"
#include "esp_event.h"
#include "esp_netif.h"
#include "protocol_examples_common.h"
#include "esp_chip_info.h"
#include "esp_flash.h"
#include "driver/ledc.h"
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/semphr.h"
#include "freertos/queue.h"
#include "lwip/sockets.h"
#include "lwip/dns.h"
#include "lwip/netdb.h"
#include "esp_log.h"
#include "mqtt_client.h"
```

Macros

- `#define LOCK_STATUS_TOPIC "brendan/lockStatus/"`
- `#define PIN_OUTPUT_TOPIC "brendan/pinEntry/"`
- `#define LED_PIN 2`
- `#define SERVO_PIN 4`
- `#define LCD_Enable GPIO_NUM_22`
- `#define LCD_RS GPIO_NUM_23`
- `#define LCD_DB4 GPIO_NUM_32`
- `#define LCD_DB5 GPIO_NUM_33`

- `#define LCD_DB6` GPIO_NUM_25
- `#define LCD_DB7` GPIO_NUM_26
- `#define LEDC_TIMER` LEDC_TIMER_0
- `#define LEDC_MODE` LEDC_LOW_SPEED_MODE
- `#define LEDC_OUTPUT_IO` (4)
- `#define LEDC_CHANNEL` LEDC_CHANNEL_0
- `#define LEDC_DUTY_RES` LEDC_TIMER_13_BIT
- `#define LEDC_DUTY_LOCKED` (((1 << 13) - 1) * 0.14)
- `#define LEDC_DUTY_UNLOCKED` (((1 << 13) - 1) * 0.07)
- `#define LEDC_FREQUENCY` (50)
- `#define MAX_STRING_SIZE` 40
- `#define NUMBER_OF_STRING` 4
- `#define CONFIG_BROKER_URL` "mqtt://test.mosquitto.org"

Functions

- void **ledBlink** (void *pvParams)
- void **printDeviceInfo** (void)
Prints the device information upon startup.
- void **lockBolt** (void)
Updates the duty cycle of the servo to lock the deadbolt.
- void **unlockBolt** (void)
Updates the duty cycle of the servo to unlock the deadbolt.
- void **lockInit** (void)
Initializes a timer for PWM signal to the servo.
- void **initLCD** (void)
Initializes pins used by the LCD and runs initialization sequence.
- void **initSequenceLCD** (void)
Initialization command sequence for HD44780 LCD controller.
- void **pulseEnable** (void)
Pulses the enable pin on the LCD.
- void **push_nibble** (uint8_t var)
Pushes a nibble (4 bits) to the data pins on the LCD.
- void **push_byte** (uint8_t var)
Pushes a byte (8 bits) to the data pins on the LCD, 1 nibble at a time.
- void **commandWrite** (uint8_t var)
Writes a command to the LCD.
- void **dataWrite** (uint8_t var)
Writes data to the LCD.
- void **writeEnterPinScreen** (void)
- void **printToLCD** (void)
Prints strings to four lines of the LCD.
- void **app_main** (void)
Main function for application.
- bool **checkPin** (int *entry, int size)
Checks the inputted PIN code, returns if correct or not.
- void **changeScreenStateLCD** (void)
Future method to be implemented, will include switch statement to change what is displayed on the LCD.
- void **writeUnlockScreen** (bool isRemote)
Future method to be implemented, will include data to be written when in the unlocked state.
- void **writeLockScreen** (bool isRemote)
Future method to be implemented, will include data to be written when in the locked state.
- void **mqtt_pin_to_int_array** (uint32_t kLen, char *input)
Takes the mqtt message and converts it into an array of integers This number is then compared to the stored pin and the deadbolt is locked/unlocked accordingly.

Variables

- `esp_mqtt_client_config_t` **mqtt_cfg**
- `esp_mqtt_client_handle_t` **client**
- `char arr [NUMBER_OF_STRING][MAX_STRING_SIZE]`
- `int pinSize = 6`
- `int pin [6] = {1, 2, 3, 4, 5, 6}`

2.1.1 Macro Definition Documentation

2.1.1.1 CONFIG_BROKER_URL

```
#define CONFIG_BROKER_URL "mqtt://test.mosquitto.org/"
```

2.1.1.2 LCD_DB4

```
#define LCD_DB4 GPIO_NUM_32
```

2.1.1.3 LCD_DB5

```
#define LCD_DB5 GPIO_NUM_33
```

2.1.1.4 LCD_DB6

```
#define LCD_DB6 GPIO_NUM_25
```

2.1.1.5 LCD_DB7

```
#define LCD_DB7 GPIO_NUM_26
```

2.1.1.6 LCD_Enable

```
#define LCD_Enable GPIO_NUM_22
```

2.1.1.7 LCD_RS

```
#define LCD_RS GPIO_NUM_23
```

2.1.1.8 LED_PIN

```
#define LED_PIN 2
```

2.1.1.9 LEDC_CHANNEL

```
#define LEDC_CHANNEL LEDC_CHANNEL_0
```

2.1.1.10 LEDC_DUTY_LOCKED

```
#define LEDC_DUTY_LOCKED (((1 << 13) - 1) * 0.14)
```

2.1.1.11 LEDC_DUTY_RES

```
#define LEDC_DUTY_RES LEDC_TIMER_13_BIT
```

2.1.1.12 LEDC_DUTY_UNLOCKED

```
#define LEDC_DUTY_UNLOCKED (((1 << 13) - 1) * 0.07)
```

2.1.1.13 LEDC_FREQUENCY

```
#define LEDC_FREQUENCY (50)
```

2.1.1.14 LEDC_MODE

```
#define LEDC_MODE LEDC_LOW_SPEED_MODE
```

2.1.1.15 LEDC_OUTPUT_IO

```
#define LEDC_OUTPUT_IO (4)
```

2.1.1.16 LEDC_TIMER

```
#define LEDC_TIMER LEDC_TIMER_0
```

2.1.1.17 LOCK_STATUS_TOPIC

```
#define LOCK_STATUS_TOPIC "brendan/lockStatus/"
```

2.1.1.18 MAX_STRING_SIZE

```
#define MAX_STRING_SIZE 40
```

2.1.1.19 NUMBER_OF_STRING

```
#define NUMBER_OF_STRING 4
```

2.1.1.20 PIN_OUTPUT_TOPIC

```
#define PIN_OUTPUT_TOPIC "brendan/pinEntry/"
```

2.1.1.21 SERVO_PIN

```
#define SERVO_PIN 4
```

2.1.2 Function Documentation

2.1.2.1 app_main()

```
void app_main (  
    void )
```

Main function for application.

2.1.2.2 changeScreenStateLCD()

```
void changeScreenStateLCD (  
    void )
```

Future method to be implemented, will include switch statement to change what is displayed on the LCD.

2.1.2.3 checkPin()

```
bool checkPin (  
    int * entry,  
    int size )
```

Checks the inputted PIN code, returns if correct or not.

Parameters

<i>entry</i>	PIN code in form of integer array
<i>size</i>	number of characters within PIN code

Returns

boolean value whether or not PIN code is correct

2.1.2.4 `commandWrite()`

```
void commandWrite (
    uint8_t var )
```

Writes a commmand to the LCD.

Parameters

<i>var</i>	command to be written
------------	-----------------------

2.1.2.5 `dataWrite()`

```
void dataWrite (
    uint8_t var )
```

Writes data to the LCD.

Parameters

<i>var</i>	data to be written (characters)
------------	---------------------------------

2.1.2.6 `initLCD()`

```
void initLCD (
    void )
```

Initializes pins used by the LCD and runs iniatialization sequence.

2.1.2.7 `initSequenceLCD()`

```
void initSequenceLCD (
    void )
```

Initialization command sequence for HD44780 LCD controller.

2.1.2.8 `ledBlink()`

```
void ledBlink (
    void * pvParams )
```

2.1.2.9 `lockBolt()`

```
void lockBolt (
    void )
```

Updates the duty cycle of the servo to lock the deadbolt.

2.1.2.10 lockInit()

```
void lockInit (
    void )
```

Initializes a timer for PWM signal to the servo.

2.1.2.11 mqtt_pin_to_int_array()

```
void mqtt_pin_to_int_array (
    uint32_t kLen,
    char * input )
```

Takes the mqtt message and converts it into an array of integers This number is then compared to the stored pin and the deadbolt is locked/unlocked accordingly.

Parameters

<i>kLen</i>	The length of the string
<i>input</i>	A string array of the input pin written as characters and converted to integers

2.1.2.12 printDeviceInfo()

```
void printDeviceInfo (
    void )
```

Prints the device information upon startup.

2.1.2.13 printToLCD()

```
void printToLCD (
    void )
```

Prints strings to four lines of the LCD.

2.1.2.14 pulseEnable()

```
void pulseEnable (
    void )
```

Pulses the enable pin on the LCD.

2.1.2.15 push_byte()

```
void push_byte (
    uint8_t var )
```

Pushes a byte (8 bits) to the data pins on the LCD, 1 nibble at a time.

Parameters

<i>var</i>	8 bit number to be sent to the LCD
------------	------------------------------------

2.1.2.16 push_nibble()

```
void push_nibble (
    uint8_t var )
```

Pushes a nibble (4 bits) to the data pins on the LCD.

Parameters

<i>var</i>	4 bit number to be sent to the LCD
------------	------------------------------------

2.1.2.17 unlockBolt()

```
void unlockBolt (
    void )
```

Updates the duty cycle of the servo to unlock the deadbolt.

2.1.2.18 writeEnterPinScreen()

```
void writeEnterPinScreen (
    void )
```

2.1.2.19 writeLockScreen()

```
void writeLockScreen (
    bool isRemote )
```

Future method to be implemented, will include data to be written when in the locked state.

Parameters

<i>isRemote</i>	boolean value if system is locked through mobile app or not
-----------------	---

2.1.2.20 writeUnlockScreen()

```
void writeUnlockScreen (
    bool isRemote )
```

Future method to be implemented, will include data to be written when in the unlocked state.

Parameters

<i>isRemote</i>	boolean value if system is unlocked through mobile app or not
-----------------	---

2.1.3 Variable Documentation

2.1.3.1 arr

```
char arr[ NUMBER_OF_STRING][ MAX_STRING_SIZE]
```

Initial value:

```
= {  
    "CIS 350", "Midterm Release", "", "Group 1"  
}
```

2.1.3.2 client

```
esp_mqtt_client_handle_t client
```

2.1.3.3 mqtt_cfg

```
esp_mqtt_client_config_t mqtt_cfg
```

Initial value:

```
= {  
    .broker.address.uri = CONFIG_BROKER_URL,  
}
```

2.1.3.4 pin

```
int pin[6] = {1, 2, 3, 4, 5, 6}
```

2.1.3.5 pinSize

```
int pinSize = 6
```


Index

- app_main
 - lock_main.c, 7
- arr
 - lock_main.c, 11
- changeScreenStateLCD
 - lock_main.c, 7
- checkPin
 - lock_main.c, 7
- client
 - lock_main.c, 11
- commandWrite
 - lock_main.c, 7
- CONFIG_BROKER_URL
 - lock_main.c, 5
- dataWrite
 - lock_main.c, 8
- initLCD
 - lock_main.c, 8
- initSequenceLCD
 - lock_main.c, 8
- LCD_DB4
 - lock_main.c, 5
- LCD_DB5
 - lock_main.c, 5
- LCD_DB6
 - lock_main.c, 5
- LCD_DB7
 - lock_main.c, 5
- LCD_Enable
 - lock_main.c, 5
- LCD_RS
 - lock_main.c, 5
- LED_PIN
 - lock_main.c, 5
- ledBlink
 - lock_main.c, 8
- LEDC_CHANNEL
 - lock_main.c, 5
- LEDC_DUTY_LOCKED
 - lock_main.c, 6
- LEDC_DUTY_RES
 - lock_main.c, 6
- LEDC_DUTY_UNLOCKED
 - lock_main.c, 6
- LEDC_FREQUENCY
 - lock_main.c, 6
- LEDC_MODE
 - lock_main.c, 6
- LEDC_OUTPUT_IO
 - lock_main.c, 6
- LEDC_TIMER
 - lock_main.c, 6
- lock/main/lock_main.c, 3
- lock_main.c
 - app_main, 7
 - arr, 11
 - changeScreenStateLCD, 7
 - checkPin, 7
 - client, 11
 - commandWrite, 7
 - CONFIG_BROKER_URL, 5
 - dataWrite, 8
 - initLCD, 8
 - initSequenceLCD, 8
 - LCD_DB4, 5
 - LCD_DB5, 5
 - LCD_DB6, 5
 - LCD_DB7, 5
 - LCD_Enable, 5
 - LCD_RS, 5
 - LED_PIN, 5
 - ledBlink, 8
 - LEDC_CHANNEL, 5
 - LEDC_DUTY_LOCKED, 6
 - LEDC_DUTY_RES, 6
 - LEDC_DUTY_UNLOCKED, 6
 - LEDC_FREQUENCY, 6
 - LEDC_MODE, 6
 - LEDC_OUTPUT_IO, 6
 - LEDC_TIMER, 6
 - LOCK_STATUS_TOPIC, 6
 - lockBolt, 8
 - lockInit, 8
 - MAX_STRING_SIZE, 6
 - mqtt_cfg, 11
 - mqtt_pin_to_int_array, 9
 - NUMBER_OF_STRING, 6
 - pin, 11
 - PIN_OUTPUT_TOPIC, 7
 - pinSize, 11
 - printDeviceInfo, 9
 - printToLCD, 9
 - pulseEnable, 9
 - push_byte, 9
 - push_nibble, 10

- SERVO_PIN, 7
- unlockBolt, 10
- writeEnterPinScreen, 10
- writeLockScreen, 10
- writeUnlockScreen, 10
- LOCK_STATUS_TOPIC
 - lock_main.c, 6
- lockBolt
 - lock_main.c, 8
- lockInit
 - lock_main.c, 8
- MAX_STRING_SIZE
 - lock_main.c, 6
- mqtt_cfg
 - lock_main.c, 11
- mqtt_pin_to_int_array
 - lock_main.c, 9
- NUMBER_OF_STRING
 - lock_main.c, 6
- pin
 - lock_main.c, 11
- PIN_OUTPUT_TOPIC
 - lock_main.c, 7
- pinSize
 - lock_main.c, 11
- printDeviceInfo
 - lock_main.c, 9
- printToLCD
 - lock_main.c, 9
- pulseEnable
 - lock_main.c, 9
- push_byte
 - lock_main.c, 9
- push_nibble
 - lock_main.c, 10
- SERVO_PIN
 - lock_main.c, 7
- unlockBolt
 - lock_main.c, 10
- writeEnterPinScreen
 - lock_main.c, 10
- writeLockScreen
 - lock_main.c, 10
- writeUnlockScreen
 - lock_main.c, 10