ESP32-S3-VOICE-PROJECT

1.0

Generated by Doxygen 1.13.2

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

Chapter 2

File Documentation

2.1 /Users/ryanjing/Local/Projects-Local/ESP32-S3-Voice-↩ Project/application/build/CMakeFiles/4.0.0/CompilerIdC/CMake↩ CCompilerId.c File Reference

Macros

- #define __has_include(x)
- #define COMPILER_ID ""
- #define STRINGIFY_HELPER(X)
- #define STRINGIFY(X)
- #define PLATFORM_ID
- #define ARCHITECTURE_ID
- #define DEC(n)
- #define HEX(n)
- #define C_STD_99 199901L
- #define C_STD_11 201112L
- #define C_STD_17 201710L
- #define C_STD_23 202311L
- #define C_VERSION

Functions

- int main (int argc, char *argv[ ])

Variables

- char const * info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
- char const * info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
- char const * info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
- const char * info_language_standard_default
- const char * info_language_extensions_default

2.1.1 Macro Definition Documentation

2.1.1.1 __has_include

```
#define __has_include(
              x)
```

Value:

```
0
```

2.1.1.2 ARCHITECTURE_ID

```
#define ARCHITECTURE_ID
```

2.1.1.3 C_STD_11

```
#define C_STD_11 201112L
```

2.1.1.4 C_STD_17

```
#define C_STD_17 201710L
```

2.1.1.5 C_STD_23

```
#define C_STD_23 202311L
```

2.1.1.6 C_STD_99

```
#define C_STD_99 199901L
```

2.1.1.7 C_VERSION

```
#define C_VERSION
```

2.1.1.8 COMPILER_ID

```
#define COMPILER_ID ""
```

2.1.1.9 DEC

```
#define DEC(
              n)
```

Value:

```
  ('0' + (((n) / 10000000)%10)), \
  ('0' + (((n) / 1000000)%10)),  \
  ('0' + (((n) / 100000)%10)),   \
  ('0' + (((n) / 10000)%10)),    \
  ('0' + (((n) / 1000)%10)),     \
  ('0' + (((n) / 100)%10)),      \
  ('0' + (((n) / 10)%10)),       \
  ('0' +  ((n) % 10))
```

2.1.1.10 HEX

```
#define HEX(
              n)
```

Value:

```
  ('0' + ((n)»28 & 0xF)), \
  ('0' + ((n)»24 & 0xF)), \
  ('0' + ((n)»20 & 0xF)), \
  ('0' + ((n)»16 & 0xF)), \
  ('0' + ((n)»12 & 0xF)), \
  ('0' + ((n)»8  & 0xF)), \
  ('0' + ((n)»4  & 0xF)), \
  ('0' + ((n)     & 0xF))
```

2.1.1.11 PLATFORM_ID

```
#define PLATFORM_ID
```

2.1.1.12 STRINGIFY

```
#define STRINGIFY(
              X)
```

Value:

```
STRINGIFY_HELPER(X)
```

2.1.1.13 STRINGIFY_HELPER

```
#define STRINGIFY_HELPER(
              X)
```

Value:

```
#X
```

2.1.2 Function Documentation

2.1.2.1 main()

```
int main (
             int argc,
             char * argv[])
```

2.1.3 Variable Documentation

2.1.3.1 info_arch

```
char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
```

2.1.3.2 info_compiler

```
char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
```

2.1.3.3 info_language_extensions_default

```
const char* info_language_extensions_default
```

Initial value:

```
= "INFO" ":" "extensions_default["



  "OFF"

"]"
```

2.1.3.4 info_language_standard_default

```
const char* info_language_standard_default
```

Initial value:

```
=
  "INFO" ":" "standard_default[" C_VERSION "]"
```

2.1.3.5 info_platform

```
char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
```

2.2 /Users/ryanjing/Local/Projects-Local/ESP32-S3-Voice-↩ Project/application/build/CMakeFiles/4.0.0/CompilerIdCXX/CMake↩ CXXCompilerId.cpp File Reference

Macros

- #define __has_include(x)
- #define COMPILER_ID ""
- #define STRINGIFY_HELPER(X)
- #define STRINGIFY(X)
- #define PLATFORM_ID
- #define ARCHITECTURE_ID
- #define DEC(n)
- #define HEX(n)
- #define CXX_STD_98 199711L
- #define CXX_STD_11 201103L
- #define CXX_STD_14 201402L
- #define CXX_STD_17 201703L
- #define CXX_STD_20 202002L
- #define CXX_STD_23 202302L
- #define CXX_STD __cplusplus

Functions

- int main (int argc, char ∗argv[ ])

Variables

- char const ∗ info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
- char const ∗ info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
- char const ∗ info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
- const char ∗ info_language_standard_default
- const char ∗ info_language_extensions_default

2.2.1 Macro Definition Documentation

2.2.1.1 __has_include

```
#define __has_include(
              x)
```

Value:

```
0
```

2.2.1.2 ARCHITECTURE_ID

```
#define ARCHITECTURE_ID
```

2.2.1.3 COMPILER_ID

```
#define COMPILER_ID ""
```

2.2.1.4 CXX_STD

```
#define CXX_STD __cplusplus
```

2.2.1.5 CXX_STD_11

```
#define CXX_STD_11 201103L
```

2.2.1.6 CXX_STD_14

```
#define CXX_STD_14 201402L
```

2.2.1.7 CXX_STD_17

```
#define CXX_STD_17 201703L
```

2.2.1.8 CXX_STD_20

```
#define CXX_STD_20 202002L
```

2.2.1.9 CXX_STD_23

```
#define CXX_STD_23 202302L
```

2.2.1.10 CXX_STD_98

```
#define CXX_STD_98 199711L
```

2.2.1.11 DEC

```
#define DEC(
              n)
```

Value:

```
  ('0' + (((n) / 10000000)%10)), \
  ('0' + (((n) / 1000000)%10)),  \
  ('0' + (((n) / 100000)%10)),   \
  ('0' + (((n) / 10000)%10)),    \
  ('0' + (((n) / 1000)%10)),     \
  ('0' + (((n) / 100)%10)),      \
  ('0' + (((n) / 10)%10)),       \
  ('0' +  ((n) % 10))
```

2.2.1.12 HEX

```
#define HEX(
              n)
```

Value:

```
  ('0' + ((n)»28 & 0xF)), \
  ('0' + ((n)»24 & 0xF)), \
  ('0' + ((n)»20 & 0xF)), \
  ('0' + ((n)»16 & 0xF)), \
  ('0' + ((n)»12 & 0xF)), \
  ('0' + ((n)»8  & 0xF)), \
  ('0' + ((n)»4  & 0xF)), \
  ('0' + ((n)     & 0xF))
```

2.2.1.13 PLATFORM_ID

```
#define PLATFORM_ID
```

2.2.1.14 STRINGIFY

```
#define STRINGIFY(
              X)
```

Value:

```
STRINGIFY_HELPER(X)
```

2.2.1.15 STRINGIFY_HELPER

```
#define STRINGIFY_HELPER(
            X)
```

Value:

```
#X
```

2.2.2 Function Documentation

2.2.2.1 main()

```
int main (
            int argc,
            char * argv[])
```

2.2.3 Variable Documentation

2.2.3.1 info_arch

```
char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
```

2.2.3.2 info_compiler

```
char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
```

2.2.3.3 info_language_extensions_default

```
const char* info_language_extensions_default
```

Initial value:

```
= "INFO" ":" "extensions_default["



  "OFF"
"]"
```

2.2.3.4 info_language_standard_default

```
const char* info_language_standard_default
```

Initial value:

```
= "INFO" ":" "standard_default["









  "98"
"]"
```

2.2.3.5 info_platform

```
char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
```

2.3 /Users/ryanjing/Local/Projects-Local/ESP32-S3-Voice-Project/application/main.c File Reference

This project aims to test sound processing algorithms on the ESP32 platform.

```
#include <stdio.h>
#include <stdlib.h>
#include "driver/i2s_std.h"
#include "esp_log.h"
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "utils.h"
#include "driver/gpio.h"
```

Macros

- #define TAG "main.c"
- #define TASK_STACK_SIZE 2048
- #define TASK_PRIORITY 5
- #define TASK_DELAY 1000 / portTICK_PERIOD_MS
- #define LED_GPIO 2
- #define LED_DELAY 1000 / portTICK_PERIOD_MS

Functions

- void communication_task (void *pvParameters)
- void audio_processing_task (void *pvParameters)

  This function is the main task for audio processing.
- void led_task (void *pvParameters)

  This function is the main task for LED blinking.
- void main_initialize (void)
- void app_main ()

2.3.1 Detailed Description

This project aims to test sound processing algorithms on the ESP32 platform.

Author

Ryan Jing (r5jing@uwaterloo.ca)

```
The goal is to implement a simple sound processing algorithm that can be run on the ESP32,
and to limit test the performance of the ESP32 platform for sound processing tasks.

The project will use the ESP-IDF framework and the ESP32's I2S interface to read and write
audio data. The project will also use FreeRTOS to manage tasks and scheduling.
```

Version

0.1

Date

2025-04-09

2.3.2 Macro Definition Documentation

2.3.2.1 LED_DELAY

```
#define LED_DELAY 1000 / portTICK_PERIOD_MS
```

2.3.2.2 LED_GPIO

```
#define LED_GPIO 2
```

2.3.2.3 TAG

```
#define TAG "main.c"
```

2.3.2.4 TASK_DELAY

```
#define TASK_DELAY 1000 / portTICK_PERIOD_MS
```

2.3.2.5 TASK_PRIORITY

```
#define TASK_PRIORITY 5
```

2.3.2.6 TASK_STACK_SIZE

```
#define TASK_STACK_SIZE 2048
```

2.3.3 Function Documentation

2.3.3.1 app_main()

```
void app_main ()
```

2.3.3.2 audio_processing_task()

```
void audio_processing_task (
            void * pvParameters)
```

This function is the main task for audio processing.

Parameters

out	*pvParameters*	

2.3.3.3 communication_task()

```
void communication_task (
            void * pvParameters)
```

Parameters

out	*pvParameters*	

2.3.3.4 led_task()

```
void led_task (
            void * pvParameters)
```

This function is the main task for LED blinking.

Parameters

pvParameters	

2.3.3.5 main_initialize()

```
void main_initialize (
            void )
```

Parameters

in	*void*	

2.4 /Users/ryanjing/Local/Projects-Local/ESP32-S3-Voice-↩ Project/application/mic.c File Reference

```
#include "driver/i2s.h"
```

Functions

- int main (void)

2.4.1 Function Documentation

2.4.1.1 main()

```
int main (
            void )
```

2.5 /Users/ryanjing/Local/Projects-Local/ESP32-S3-Voice-Project/include/mic.h File Reference

2.5.1 Detailed Description

Author

Ryan Jing (r5jing@uwaterloo.ca)

Version

0.1

Date

2025-04-12

2.6 mic.h

Go to the documentation of this file.

```
00001 /*****************************************************************************************************/
00013 /*****************************************************************************************************/
00014
00015 #ifndef MIC_H
00016 #define MIC_H
00017
00018 /*-----------------------------------------------------------------------------------------------------*/
00019 // HEADERS                                                                                            */
00020 /*-----------------------------------------------------------------------------------------------------*/
00021
00022
00023
00024 /*-----------------------------------------------------------------------------------------------------*/
00025 // GLOBAL VARIABLES                                                                                   */
00026 /*-----------------------------------------------------------------------------------------------------*/
00027
00028
00029
00030 /*-----------------------------------------------------------------------------------------------------*/
00031 // CLASS DECLARATIONS                                                                                 */
00032 /*-----------------------------------------------------------------------------------------------------*/
00033
00034
00035
00036 /*-----------------------------------------------------------------------------------------------------*/
00037 // FUNCTION DECLARATIONS                                                                              */
00038 /*-----------------------------------------------------------------------------------------------------*/
00039
00040
00041
00042 #endif // MIC_H
```

2.7 /Users/ryanjing/Local/Projects-Local/ESP32-S3-Voice-Project/include/utils.h File Reference

Variables

- volatile int audio_buffer [1024]

2.7.1 Detailed Description

Author

Ryan Jing (r5jing@uwaterloo.ca)

Version

0.1

Date

2025-04-12

2.7.2 Variable Documentation

2.7.2.1 audio_buffer

```
volatile int audio_buffer[1024]
```

2.8 utils.h

Go to the documentation of this file.

```
00001 /*****************************************************************************************************/
00014 /*****************************************************************************************************/
00015
00016 #ifndef HEADER_NAME_H
00017 #define HEADER_NAME_H
00018
00019 /*-----------------------------------------------------------------------------------------------------*/
00020 // HEADERS                                                                                            */
00021 /*-----------------------------------------------------------------------------------------------------*/
00022
00023 /*-----------------------------------------------------------------------------------------------------*/
00024 // GLOBAL VARIABLES                                                                                   */
00025 /*-----------------------------------------------------------------------------------------------------*/
00026
00027 volatile int audio_buffer[1024];  // Testing
00028
00029 /*-----------------------------------------------------------------------------------------------------*/
00030 // CLASS DECLARATIONS                                                                                 */
00031 /*-----------------------------------------------------------------------------------------------------*/
00032
00033 /*-----------------------------------------------------------------------------------------------------*/
00034 // FUNCTION DECLARATIONS                                                                              */
00035 /*-----------------------------------------------------------------------------------------------------*/
00036
00037 #endif  // HEADER_NAME_H
```