# 2puck

Generated by Doxygen 1.9.4

# Chapter 1

# Module Index

## 1.1 Modules

Here is a list of all modules:

# Chapter 2

# Data Structure Index

## 2.1 Data Structures

Here are the data structures with brief descriptions:

# Chapter 3

# File Index

## 3.1 File List

Here is a list of all documented files with brief descriptions:

# Chapter 4

# Module Documentation

## 4.1 Config

### Macros

- #define **CHPRINTF_USE_FLOAT** true

### Kernel hooks

- void **panic_handler** (const char *reason)

  *System halt hook.*
- #define **CH_CFG_THREAD_EXTRA_FIELDS** /* Add threads custom fields here.*/

  *Threads descriptor structure extension.*
- #define **CH_CFG_THREAD_INIT_HOOK**(tp)

  *Threads initialization hook.*
- #define **CH_CFG_THREAD_EXIT_HOOK**(tp)

  *Threads finalization hook.*
- #define **CH_CFG_CONTEXT_SWITCH_HOOK**(ntp, otp)

  *Context switch hook.*
- #define **CH_CFG_IDLE_ENTER_HOOK**()

  *Idle thread enter hook.*
- #define **CH_CFG_IDLE_LEAVE_HOOK**()

  *Idle thread leave hook.*
- #define **CH_CFG_IDLE_LOOP_HOOK**()

  *Idle Loop hook.*
- #define **CH_CFG_SYSTEM_TICK_HOOK**()

  *System tick event hook.*
- #define **CH_CFG_SYSTEM_HALT_HOOK**(reason)

### System timers settings

- #define **CH_CFG_ST_RESOLUTION** 32

  *System time counter resolution.*
- #define **CH_CFG_ST_FREQUENCY** 1000

  *System tick frequency.*
- #define **CH_CFG_ST_TIMEDELTA** 0

  *Time delta constant for the tick-less mode.*

### Kernel parameters and options

- #define **CH_CFG_TIME_QUANTUM** 20

  *Round robin interval.*
- #define **CH_CFG_MEMCORE_SIZE** 0

  *Managed RAM size.*
- #define **CH_CFG_NO_IDLE_THREAD** FALSE

  *Idle thread automatic spawn suppression.*

### Performance options

- #define **CH_CFG_OPTIMIZE_SPEED** TRUE

  *OS optimization.*

### Subsystem options

- #define **CH_CFG_USE_TM** TRUE

  *Time Measurement APIs.*
- #define **CH_CFG_USE_REGISTRY** TRUE

  *Threads registry APIs.*
- #define **CH_CFG_USE_WAITEXIT** TRUE

  *Threads synchronization APIs.*
- #define **CH_CFG_USE_SEMAPHORES** TRUE

  *Semaphores APIs.*
- #define **CH_CFG_USE_SEMAPHORES_PRIORITY** FALSE

  *Semaphores queuing mode.*
- #define **CH_CFG_USE_MUTEXES** TRUE

  *Mutexes APIs.*
- #define **CH_CFG_USE_MUTEXES_RECURSIVE** TRUE

  *Enables recursive behavior on mutexes.*
- #define **CH_CFG_USE_CONDVARS** TRUE

  *Conditional Variables APIs.*
- #define **CH_CFG_USE_CONDVARS_TIMEOUT** TRUE

  *Conditional Variables APIs with timeout.*
- #define **CH_CFG_USE_EVENTS** TRUE

  *Events Flags APIs.*
- #define **CH_CFG_USE_EVENTS_TIMEOUT** TRUE

  *Events Flags APIs with timeout.*
- #define **CH_CFG_USE_MESSAGES** TRUE

  *Synchronous Messages APIs.*
- #define **CH_CFG_USE_MESSAGES_PRIORITY** FALSE

  *Synchronous Messages queuing mode.*
- #define **CH_CFG_USE_MAILBOXES** TRUE

  *Mailboxes APIs.*
- #define **CH_CFG_USE_QUEUES** TRUE

  *I/O Queues APIs.*
- #define **CH_CFG_USE_MEMCORE** TRUE

  *Core Memory Manager APIs.*
- #define **CH_CFG_USE_HEAP** TRUE

  *Heap Allocator APIs.*
- #define **CH_CFG_USE_MEMPOOLS** TRUE

  *Memory Pools Allocator APIs.*
- #define **CH_CFG_USE_DYNAMIC** TRUE

  *Dynamic Threads APIs.*

### Debug options

- #define **CH_DBG_STATISTICS** TRUE

  *Debug option, kernel statistics.*
- #define **CH_DBG_SYSTEM_STATE_CHECK** TRUE

  *Debug option, system state check.*
- #define **CH_DBG_ENABLE_CHECKS** TRUE

  *Debug option, parameters checks.*
- #define **CH_DBG_ENABLE_ASSERTS** TRUE

  *Debug option, consistency checks.*
- #define **CH_DBG_ENABLE_TRACE** TRUE

  *Debug option, trace buffer.*
- #define **CH_DBG_ENABLE_STACK_CHECK** TRUE

  *Debug option, stack checks.*
- #define **CH_DBG_FILL_THREADS** TRUE

  *Debug option, stacks initialization.*
- #define **CH_DBG_THREADS_PROFILING** TRUE

  *Debug option, threads profiling.*

## 4.1.1 Detailed Description

Kernel related settings and hooks.

## 4.1.2 Macro Definition Documentation

### 4.1.2.1 CH_CFG_CONTEXT_SWITCH_HOOK

```
#define CH_CFG_CONTEXT_SWITCH_HOOK(
              ntp,
              otp )
```

**Value:**

```
{                                   \
       /* Context switch code here.*/                                   \
}
```

Context switch hook.

This hook is invoked just before switching between threads.

### 4.1.2.2 CH_CFG_IDLE_ENTER_HOOK

```
#define CH_CFG_IDLE_ENTER_HOOK( )
```

**Value:**

```
{                                       \
}
```

Idle thread enter hook.

**Note**

This hook is invoked within a critical zone, no OS functions should be invoked from here.

This macro can be used to activate a power saving mode.

#### 4.1.2.3 CH_CFG_IDLE_LEAVE_HOOK

```
#define CH_CFG_IDLE_LEAVE_HOOK( )
```

**Value:**

```
{                                       \
}
```

Idle thread leave hook.

**Note**

This hook is invoked within a critical zone, no OS functions should be invoked from here.

This macro can be used to deactivate a power saving mode.

#### 4.1.2.4 CH_CFG_IDLE_LOOP_HOOK

```
#define CH_CFG_IDLE_LOOP_HOOK( )
```

**Value:**

```
        {                                       \
        /* Idle loop code here.*/                                       \
}
```

Idle Loop hook.

This hook is continuously invoked by the idle thread loop.

#### 4.1.2.5 CH_CFG_MEMCORE_SIZE

```
#define CH_CFG_MEMCORE_SIZE 0
```

Managed RAM size.

Size of the RAM area to be managed by the OS. If set to zero then the whole available RAM is used. The core memory is made available to the heap allocator and/or can be used directly through the simplified core memory allocator.

**Note**

In order to let the OS manage the whole RAM the linker script must provide the `__heap_base__` and `__heap_end__` symbols.

Requires `CH_CFG_USE_MEMCORE`.

#### 4.1.2.6 CH_CFG_NO_IDLE_THREAD

```
#define CH_CFG_NO_IDLE_THREAD FALSE
```

Idle thread automatic spawn suppression.

When this option is activated the function `chSysInit()` does not spawn the idle thread. The application `main()` function becomes the idle thread and must implement an infinite loop.

#### 4.1.2.7 CH_CFG_OPTIMIZE_SPEED

```
#define CH_CFG_OPTIMIZE_SPEED TRUE
```

OS optimization.

If enabled then time efficient rather than space efficient code is used when two possible implementations exist.

**Note**

This is not related to the compiler optimization options.

The default is `TRUE`.

#### 4.1.2.8 CH_CFG_ST_FREQUENCY

```
#define CH_CFG_ST_FREQUENCY 1000
```

System tick frequency.

Frequency of the system timer that drives the system ticks. This setting also defines the system tick time unit.

#### 4.1.2.9 CH_CFG_ST_RESOLUTION

```
#define CH_CFG_ST_RESOLUTION 32
```

System time counter resolution.

**Note**

Allowed values are 16 or 32 bits.

#### 4.1.2.10 CH_CFG_ST_TIMEDELTA

```
#define CH_CFG_ST_TIMEDELTA 0
```

Time delta constant for the tick-less mode.

**Note**

If this value is zero then the system uses the classic periodic tick. This value represents the minimum number of ticks that is safe to specify in a timeout directive. The value one is not valid, timeouts are rounded up to this value.

#### 4.1.2.11 CH_CFG_SYSTEM_HALT_HOOK

```
#define CH_CFG_SYSTEM_HALT_HOOK(
              reason )
```

**Value:**

```
{                                   \
/* System halt code here.*/                                     \
panic_handler(reason);                                          \
}
```

#### 4.1.2.12 CH_CFG_SYSTEM_TICK_HOOK

```
#define CH_CFG_SYSTEM_TICK_HOOK( )
```

**Value:**

```
{                                       \
/* System tick event code here.*/                               \
}
```

System tick event hook.

This hook is invoked in the system tick handler immediately after processing the virtual timers queue.

#### 4.1.2.13 CH_CFG_THREAD_EXIT_HOOK

```
#define CH_CFG_THREAD_EXIT_HOOK(
              tp )
```

**Value:**

```
{                                     \
/* Add threads finalization code here.*/                        \
}
```

Threads finalization hook.

User finalization code added to the `chThdExit()` API.

**Note**

It is inserted into lock zone.

It is also invoked when the threads simply return in order to terminate.

#### 4.1.2.14 CH_CFG_THREAD_EXTRA_FIELDS

```
#define CH_CFG_THREAD_EXTRA_FIELDS  /* Add threads custom fields here.*/
```

Threads descriptor structure extension.

User fields added to the end of the `thread_t` structure.

#### 4.1.2.15 CH_CFG_THREAD_INIT_HOOK

```
#define CH_CFG_THREAD_INIT_HOOK(
              tp )
```

**Value:**

```
        {                                       \
        /* Add threads initialization code here.*/                              \
}
```

Threads initialization hook.

User initialization code added to the `chThdInit()` API.

**Note**

It is invoked from within `chThdInit()` and implicitly from all the threads creation APIs.

#### 4.1.2.16 CH_CFG_TIME_QUANTUM

```
#define CH_CFG_TIME_QUANTUM 20
```

Round robin interval.

This constant is the number of system ticks allowed for the threads before preemption occurs. Setting this value to zero disables the preemption for threads with equal priority and the round robin becomes cooperative. Note that higher priority threads can still preempt, the kernel is always preemptive.

**Note**

Disabling the round robin preemption makes the kernel more compact and generally faster.

The round robin preemption is not supported in tickless mode and must be set to zero in that case.

#### 4.1.2.17 CH_CFG_USE_CONDVARS

```
#define CH_CFG_USE_CONDVARS TRUE
```

Conditional Variables APIs.

If enabled then the conditional variables APIs are included in the kernel.

**Note**

The default is `TRUE`.

Requires `CH_CFG_USE_MUTEXES`.

#### 4.1.2.18 CH_CFG_USE_CONDVARS_TIMEOUT

```
#define CH_CFG_USE_CONDVARS_TIMEOUT TRUE
```

Conditional Variables APIs with timeout.

If enabled then the conditional variables APIs with timeout specification are included in the kernel.

**Note**

The default is `TRUE`.
Requires `CH_CFG_USE_CONDVARS`.

#### 4.1.2.19 CH_CFG_USE_DYNAMIC

```
#define CH_CFG_USE_DYNAMIC TRUE
```

Dynamic Threads APIs.

If enabled then the dynamic threads creation APIs are included in the kernel.

**Note**

The default is `TRUE`.
Requires `CH_CFG_USE_WAITEXIT`.
Requires `CH_CFG_USE_HEAP` and/or `CH_CFG_USE_MEMPOOLS`.

#### 4.1.2.20 CH_CFG_USE_EVENTS

```
#define CH_CFG_USE_EVENTS TRUE
```

Events Flags APIs.

If enabled then the event flags APIs are included in the kernel.

**Note**

The default is `TRUE`.

#### 4.1.2.21 CH_CFG_USE_EVENTS_TIMEOUT

```
#define CH_CFG_USE_EVENTS_TIMEOUT TRUE
```

Events Flags APIs with timeout.

If enabled then the events APIs with timeout specification are included in the kernel.

**Note**

The default is `TRUE`.

Requires `CH_CFG_USE_EVENTS`.

#### 4.1.2.22 CH_CFG_USE_HEAP

```
#define CH_CFG_USE_HEAP TRUE
```

Heap Allocator APIs.

If enabled then the memory heap allocator APIs are included in the kernel.

**Note**

The default is `TRUE`.

Requires `CH_CFG_USE_MEMCORE` and either `CH_CFG_USE_MUTEXES` or `CH_CFG_USE_SEMAPHORES`.

Mutexes are recommended.

#### 4.1.2.23 CH_CFG_USE_MAILBOXES

```
#define CH_CFG_USE_MAILBOXES TRUE
```

Mailboxes APIs.

If enabled then the asynchronous messages (mailboxes) APIs are included in the kernel.

**Note**

The default is `TRUE`.

Requires `CH_CFG_USE_SEMAPHORES`.

#### 4.1.2.24 CH_CFG_USE_MEMCORE

```
#define CH_CFG_USE_MEMCORE TRUE
```

Core Memory Manager APIs.

If enabled then the core memory manager APIs are included in the kernel.

**Note**

The default is `TRUE`.

#### 4.1.2.25 CH_CFG_USE_MEMPOOLS

```
#define CH_CFG_USE_MEMPOOLS TRUE
```

Memory Pools Allocator APIs.

If enabled then the memory pools allocator APIs are included in the kernel.

**Note**

The default is `TRUE`.

#### 4.1.2.26 CH_CFG_USE_MESSAGES

```
#define CH_CFG_USE_MESSAGES TRUE
```

Synchronous Messages APIs.

If enabled then the synchronous messages APIs are included in the kernel.

**Note**

The default is `TRUE`.

#### 4.1.2.27 CH_CFG_USE_MESSAGES_PRIORITY

```
#define CH_CFG_USE_MESSAGES_PRIORITY FALSE
```

Synchronous Messages queuing mode.

If enabled then messages are served by priority rather than in FIFO order.

**Note**

The default is `FALSE`. Enable this if you have special requirements.
Requires `CH_CFG_USE_MESSAGES`.

#### 4.1.2.28 CH_CFG_USE_MUTEXES

```
#define CH_CFG_USE_MUTEXES TRUE
```

Mutexes APIs.

If enabled then the mutexes APIs are included in the kernel.

**Note**

The default is `TRUE`.

#### 4.1.2.29 CH_CFG_USE_MUTEXES_RECURSIVE

```
#define CH_CFG_USE_MUTEXES_RECURSIVE TRUE
```

Enables recursive behavior on mutexes.

**Note**

Recursive mutexes are heavier and have an increased memory footprint.

The default is `FALSE`.

Requires `CH_CFG_USE_MUTEXES`.

#### 4.1.2.30 CH_CFG_USE_QUEUES

```
#define CH_CFG_USE_QUEUES TRUE
```

I/O Queues APIs.

If enabled then the I/O queues APIs are included in the kernel.

**Note**

The default is `TRUE`.

#### 4.1.2.31 CH_CFG_USE_REGISTRY

```
#define CH_CFG_USE_REGISTRY TRUE
```

Threads registry APIs.

If enabled then the registry APIs are included in the kernel.

**Note**

The default is `TRUE`.

#### 4.1.2.32 CH_CFG_USE_SEMAPHORES

```
#define CH_CFG_USE_SEMAPHORES TRUE
```

Semaphores APIs.

If enabled then the Semaphores APIs are included in the kernel.

**Note**

The default is `TRUE`.

#### 4.1.2.33 CH_CFG_USE_SEMAPHORES_PRIORITY

```
#define CH_CFG_USE_SEMAPHORES_PRIORITY FALSE
```

Semaphores queuing mode.

If enabled then the threads are enqueued on semaphores by priority rather than in FIFO order.

**Note**

The default is `FALSE`. Enable this if you have special requirements.
Requires `CH_CFG_USE_SEMAPHORES`.

#### 4.1.2.34 CH_CFG_USE_TM

```
#define CH_CFG_USE_TM TRUE
```

Time Measurement APIs.

If enabled then the time measurement APIs are included in the kernel.

**Note**

The default is `TRUE`.

#### 4.1.2.35 CH_CFG_USE_WAITEXIT

```
#define CH_CFG_USE_WAITEXIT TRUE
```

Threads synchronization APIs.

If enabled then the `chThdWait()` function is included in the kernel.

**Note**

The default is `TRUE`.

#### 4.1.2.36 CH_DBG_ENABLE_ASSERTS

```
#define CH_DBG_ENABLE_ASSERTS TRUE
```

Debug option, consistency checks.

If enabled then all the assertions in the kernel code are activated. This includes consistency checks inside the kernel, runtime anomalies and port-defined checks.

**Note**

The default is `FALSE`.

#### 4.1.2.37 CH_DBG_ENABLE_CHECKS

```
#define CH_DBG_ENABLE_CHECKS TRUE
```

Debug option, parameters checks.

If enabled then the checks on the API functions input parameters are activated.

**Note**

The default is `FALSE`.

#### 4.1.2.38 CH_DBG_ENABLE_STACK_CHECK

```
#define CH_DBG_ENABLE_STACK_CHECK TRUE
```

Debug option, stack checks.

If enabled then a runtime stack check is performed.

**Note**

The default is `FALSE`.

The stack check is performed in a architecture/port dependent way. It may not be implemented or some ports.

The default failure mode is to halt the system with the global `panic_msg` variable set to `NULL`.

#### 4.1.2.39 CH_DBG_ENABLE_TRACE

```
#define CH_DBG_ENABLE_TRACE TRUE
```

Debug option, trace buffer.

If enabled then the context switch circular trace buffer is activated.

**Note**

The default is `FALSE`.

#### 4.1.2.40 CH_DBG_FILL_THREADS

```
#define CH_DBG_FILL_THREADS TRUE
```

Debug option, stacks initialization.

If enabled then the threads working area is filled with a byte value when a thread is created. This can be useful for the runtime measurement of the used stack.

**Note**

The default is `FALSE`.

#### 4.1.2.41 CH_DBG_STATISTICS

```
#define CH_DBG_STATISTICS TRUE
```

Debug option, kernel statistics.

**Note**

The default is `FALSE`.

#### 4.1.2.42 CH_DBG_SYSTEM_STATE_CHECK

```
#define CH_DBG_SYSTEM_STATE_CHECK TRUE
```

Debug option, system state check.

If enabled the correct call protocol for system APIs is checked at runtime.

**Note**

The default is `FALSE`.

#### 4.1.2.43 CH_DBG_THREADS_PROFILING

```
#define CH_DBG_THREADS_PROFILING TRUE
```

Debug option, threads profiling.

If enabled then a field is added to the `thread_t` structure that counts the system ticks occurred while executing the thread.

**Note**

The default is `FALSE`.

This debug option is not currently compatible with the tickless mode.

### 4.1.3 Function Documentation

#### 4.1.3.1 panic_handler()

```
void panic_handler (
             const char * reason )
```

System halt hook.

This hook is invoked in case to a system halting error before the system is halted.

## 4.2 HAL_CONF

### Macros

- #define **HAL_USE_PAL** TRUE
  *Enables the PAL subsystem.*
- #define **HAL_USE_ADC** TRUE
  *Enables the ADC subsystem.*
- #define **HAL_USE_CAN** TRUE
  *Enables the CAN subsystem.*
- #define **HAL_USE_DAC** TRUE
  *Enables the DAC subsystem.*
- #define **HAL_USE_DCMI** TRUE
  *Enables the DCMI subsystem.*
- #define **HAL_USE_EXT** TRUE
  *Enables the EXT subsystem.*
- #define **HAL_USE_GPT** TRUE
  *Enables the GPT subsystem.*
- #define **HAL_USE_I2C** TRUE
  *Enables the I2C subsystem.*
- #define **HAL_USE_I2S** TRUE
  *Enables the I2S subsystem.*

- #define **HAL_USE_ICU** FALSE
  *Enables the ICU subsystem.*
- #define **HAL_USE_MAC** FALSE
  *Enables the MAC subsystem.*
- #define **HAL_USE_MMC_SPI** FALSE
  *Enables the MMC_SPI subsystem.*
- #define **HAL_USE_PWM** TRUE
  *Enables the PWM subsystem.*
- #define **HAL_USE_RTC** FALSE
  *Enables the RTC subsystem.*
- #define **HAL_USE_SDC** TRUE
  *Enables the SDC subsystem.*
- #define **HAL_USE_SERIAL** TRUE
  *Enables the SERIAL subsystem.*
- #define **HAL_USE_SERIAL_USB** TRUE
  *Enables the SERIAL over USB subsystem.*
- #define **HAL_USE_SPI** TRUE
  *Enables the SPI subsystem.*
- #define **HAL_USE_UART** FALSE
  *Enables the UART subsystem.*
- #define **HAL_USE_USB** TRUE
  *Enables the USB subsystem.*
- #define **ADC_USE_WAIT** TRUE
  *Enables synchronous APIs.*
- #define **ADC_USE_MUTUAL_EXCLUSION** TRUE
  *Enables the `adcAcquireBus()` and `adcReleaseBus()` APIs.*
- #define **CAN_USE_SLEEP_MODE** TRUE
  *Sleep mode related APIs inclusion switch.*
- #define **I2C_USE_MUTUAL_EXCLUSION** TRUE
  *Enables the mutual exclusion APIs on the I2C bus.*
- #define **MAC_USE_ZERO_COPY** FALSE
  *Enables an event sources for incoming packets.*
- #define **MAC_USE_EVENTS** TRUE
  *Enables an event sources for incoming packets.*
- #define **MMC_NICE_WAITING** TRUE
  *Delays insertions.*
- #define **SDC_INIT_RETRY** 100
  *Number of initialization attempts before rejecting the card.*
- #define **SDC_MMC_SUPPORT** FALSE
  *Include support for MMC cards.*
- #define **SDC_NICE_WAITING** TRUE
  *Delays insertions.*
- #define **SERIAL_DEFAULT_BITRATE** 115200
  *Default bit rate.*
- #define **SERIAL_BUFFERS_SIZE** 16
  *Serial buffers size.*
- #define **SERIAL_USB_BUFFERS_SIZE** 256
  *Serial over USB buffers size.*
- #define **SPI_USE_WAIT** TRUE
  *Enables synchronous APIs.*
- #define **SPI_USE_MUTUAL_EXCLUSION** TRUE
  *Enables the `spiAcquireBus()` and `spiReleaseBus()` APIs.*

### 4.2.1 Detailed Description

### 4.2.2 Macro Definition Documentation

#### 4.2.2.1 ADC_USE_MUTUAL_EXCLUSION

```
#define ADC_USE_MUTUAL_EXCLUSION TRUE
```

Enables the `adcAcquireBus()` and `adcReleaseBus()` APIs.

**Note**

Disabling this option saves both code and data space.

#### 4.2.2.2 ADC_USE_WAIT

```
#define ADC_USE_WAIT TRUE
```

Enables synchronous APIs.

**Note**

Disabling this option saves both code and data space.

#### 4.2.2.3 HAL_USE_DAC

```
#define HAL_USE_DAC TRUE
```

Enables the DAC subsystem.

- *+

#### 4.2.2.4 MMC_NICE_WAITING

```
#define MMC_NICE_WAITING TRUE
```

Delays insertions.

If enabled this options inserts delays into the MMC waiting routines releasing some extra CPU time for the threads with lower priority, this may slow down the driver a bit however. This option is recommended also if the SPI driver does not use a DMA channel and heavily loads the CPU.

#### 4.2.2.5 SDC_INIT_RETRY

```
#define SDC_INIT_RETRY 100
```

Number of initialization attempts before rejecting the card.

**Note**

Attempts are performed at 10mS intervals.

#### 4.2.2.6 SDC_MMC_SUPPORT

```
#define SDC_MMC_SUPPORT FALSE
```

Include support for MMC cards.

**Note**

MMC support is not yet implemented so this option must be kept at `FALSE`.

#### 4.2.2.7 SDC_NICE_WAITING

```
#define SDC_NICE_WAITING TRUE
```

Delays insertions.

If enabled this options inserts delays into the MMC waiting routines releasing some extra CPU time for the threads with lower priority, this may slow down the driver a bit however.

#### 4.2.2.8 SERIAL_BUFFERS_SIZE

```
#define SERIAL_BUFFERS_SIZE 16
```

Serial buffers size.

Configuration parameter, you can change the depth of the queue buffers depending on the requirements of your application.

**Note**

The default is 64 bytes for both the transmission and receive buffers.

#### 4.2.2.9 SERIAL_DEFAULT_BITRATE

```
#define SERIAL_DEFAULT_BITRATE 115200
```

Default bit rate.

Configuration parameter, this is the baud rate selected for the default configuration.

#### 4.2.2.10 SERIAL_USB_BUFFERS_SIZE

```
#define SERIAL_USB_BUFFERS_SIZE 256
```

Serial over USB buffers size.

Configuration parameter, the buffer size must be a multiple of the USB data endpoint maximum packet size.

**Note**

The default is 64 bytes for both the transmission and receive buffers.

#### 4.2.2.11 SPI_USE_MUTUAL_EXCLUSION

```
#define SPI_USE_MUTUAL_EXCLUSION TRUE
```

Enables the `spiAcquireBus()` and `spiReleaseBus()` APIs.

**Note**

Disabling this option saves both code and data space.

#### 4.2.2.12 SPI_USE_WAIT

```
#define SPI_USE_WAIT TRUE
```

Enables synchronous APIs.

**Note**

Disabling this option saves both code and data space.

# Chapter 5

# Data Structure Documentation

## 5.1 note_struct_t Struct Reference

### Data Fields

- char * **name**
- uint16_t **freq**

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

- music.c

## 5.2 song Struct Reference

### Data Fields

- char * **name**
- const uint8_t * **melody_ptr**
- const uint16_t **melody_size**
- char * **file_name**

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

- music.c

# Chapter 6

# File Documentation

## 6.1 audio_processing.h

```
1 /*
2  * @file        audio_processing.c
3  * @brief       Microphone sample processing library.
4  * @author      Karl Khalil
5  * @author      Joaquim Silveira
6  * @version     1.0
7  * @date        12 Apr 2022
8  * @copyright   GNU Public License
9  *
10  */
11
12 #ifndef AUDIO_PROCESSING_H
13 #define AUDIO_PROCESSING_H
14
15
16 void wait_note_played(void);
17 void wait_finish_playing(void);
18 void processAudioDataCmplx(int16_t *data, uint16_t num_samples);
19
20
21 #endif /* AUDIO_PROCESSING_H */
22
23
```

## 6.2 chconf.h

```
1 /*
2     ChibiOS - Copyright (C) 2006..2015 Giovanni Di Sirio
3
4     Licensed under the Apache License, Version 2.0 (the "License");
5     you may not use this file except in compliance with the License.
6     You may obtain a copy of the License at
7
8         http://www.apache.org/licenses/LICENSE-2.0
9
10     Unless required by applicable law or agreed to in writing, software
11     distributed under the License is distributed on an "AS IS" BASIS,
12     WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13     See the License for the specific language governing permissions and
14     limitations under the License.
15  */
16
28 #ifndef _CHCONF_H_
29 #define _CHCONF_H_
30
31 /*===========================================================================*/
36 /*===========================================================================*/
37
42 #define CH_CFG_ST_RESOLUTION                32
43
49 #define CH_CFG_ST_FREQUENCY                 1000
50
59 #define CH_CFG_ST_TIMEDELTA                 0
60
```

```
63 /*===========================================================================*/
68 /*===========================================================================*/
69
82 #define CH_CFG_TIME_QUANTUM                 20
83
95 #define CH_CFG_MEMCORE_SIZE                 0
96
103 #define CH_CFG_NO_IDLE_THREAD               FALSE
104
107 /*===========================================================================*/
112 /*===========================================================================*/
113
122 #define CH_CFG_OPTIMIZE_SPEED               TRUE
123
126 /*===========================================================================*/
131 /*===========================================================================*/
132
140 #define CH_CFG_USE_TM                       TRUE
141
148 #define CH_CFG_USE_REGISTRY                 TRUE
149
157 #define CH_CFG_USE_WAITEXIT                 TRUE
158
165 #define CH_CFG_USE_SEMAPHORES               TRUE
166
176 #define CH_CFG_USE_SEMAPHORES_PRIORITY      FALSE
177
184 #define CH_CFG_USE_MUTEXES                  TRUE
185
194 #define CH_CFG_USE_MUTEXES_RECURSIVE        TRUE
195
204 #define CH_CFG_USE_CONDVARS                 TRUE
205
214 #define CH_CFG_USE_CONDVARS_TIMEOUT         TRUE
215
222 #define CH_CFG_USE_EVENTS                   TRUE
223
232 #define CH_CFG_USE_EVENTS_TIMEOUT           TRUE
233
241 #define CH_CFG_USE_MESSAGES                 TRUE
242
252 #define CH_CFG_USE_MESSAGES_PRIORITY        FALSE
253
262 #define CH_CFG_USE_MAILBOXES                TRUE
263
270 #define CH_CFG_USE_QUEUES                   TRUE
271
279 #define CH_CFG_USE_MEMCORE                  TRUE
280
291 #define CH_CFG_USE_HEAP                     TRUE
292
300 #define CH_CFG_USE_MEMPOOLS                 TRUE
301
311 #define CH_CFG_USE_DYNAMIC                  TRUE
312
315 /*===========================================================================*/
320 /*===========================================================================*/
321
327 #define CH_DBG_STATISTICS                   TRUE
328
336 #define CH_DBG_SYSTEM_STATE_CHECK           TRUE
337
345 #define CH_DBG_ENABLE_CHECKS                TRUE
346
355 #define CH_DBG_ENABLE_ASSERTS               TRUE
356
364 #define CH_DBG_ENABLE_TRACE                 TRUE
365
376 #define CH_DBG_ENABLE_STACK_CHECK           TRUE
377
386 #define CH_DBG_FILL_THREADS                 TRUE
387
397 #define CH_DBG_THREADS_PROFILING            TRUE
398
401 /*===========================================================================*/
406 /*===========================================================================*/
407
412 #define CH_CFG_THREAD_EXTRA_FIELDS                                          \
413     /* Add threads custom fields here.*/
414
422 #define CH_CFG_THREAD_INIT_HOOK(tp) {                                       \
423         /* Add threads initialization code here.*/                          \
424 }
425
434 #define CH_CFG_THREAD_EXIT_HOOK(tp) {                                       \
435         /* Add threads finalization code here.*/                            \
436 }
```

```
437
442 #define CH_CFG_CONTEXT_SWITCH_HOOK(ntp, otp) {                              \
443         /* Context switch code here.*/                                          \
444 }
445
452 #define CH_CFG_IDLE_ENTER_HOOK() {                                          \
453 }
454
461 #define CH_CFG_IDLE_LEAVE_HOOK() {                                          \
462 }
463
468 #define CH_CFG_IDLE_LOOP_HOOK() {                                           \
469         /* Idle loop code here.*/                                               \
470 }
471
477 #define CH_CFG_SYSTEM_TICK_HOOK() {                                         \
478         /* System tick event code here.*/                                       \
479 }
480
486 #if !defined(_FROM_ASM_)
487 #ifdef __cplusplus
488 extern "C" {
489 #endif
490 void panic_handler(const char *reason);
491 #ifdef __cplusplus
492 }
493 #endif
494 #endif /* _FROM_ASM_ */
495 #define CH_CFG_SYSTEM_HALT_HOOK(reason) {                                   \
496         /* System halt code here.*/                                             \
497         panic_handler(reason);                                                  \
498 }
499
500
503 /*===========================================================================*/
504 /* Port-specific settings (override port settings defaulted in chcore.h).    */
505 /*===========================================================================*/
506
507 // chprintf float enable
508 #define CHPRINTF_USE_FLOAT true
509
510 #endif  /* _CHCONF_H_ */
511
```

## 6.3 console.h

```
1 /*
2  * @file        console.h
3  * @brief       Communication with serial monitor
4  * @author      Karl Khalil
5  * @author      Joaquim Silveira
6  * @version     1.0
7  * @date        7 May 2022
8  * @copyright   GNU Public License
9  *
10  */
11
12 #ifndef CONSOLE_H_
13 #define CONSOLE_H_
14
15 void console_init(void);
16 msg_t console_stop(void);
17 msg_t console_send_string(const char* msg);
18 msg_t console_send_int(int num, char* msg);
19 char console_get_char(char* input_msg);
20 void SendUint8ToComputer(uint8_t* data, uint16_t size);
21
22 #endif /* CONSOLE_H_ */
```

## 6.4 fft.h

```
1 /*
2  * @file        fft.h
3  * @brief       FFT wrapping functions.
4  * @author      Karl Khalil
5  * @author      Joaquim Silveira
6  * @version     1.0
7  * @date        12 Apr 2022
8  * @copyright   GNU Public License
```

```
9  *
10  */
11 #ifndef FFT_H
12 #define FFT_H
13 #include "arm_math.h"
14
15 void init_rfft_handler(uint16_t fft_size);
16 void doCmplxFFT_optimized(uint16_t size, float* complex_buffer);
17
18 #endif /* FFT_H */
```

## 6.5 game.h

```
1 /*
2  * @file        game.h
3  * @brief       Main FSM controlling the game logic.
4  * @author      Karl Khalil
5  * @version     1.0
6  * @date        29 Apr 2022
7  * @copyright   GNU Public License
8  *
9  */
10 #ifndef GAME_H_
11 #define GAME_H_
12
13
14 void game_init(void);
15 msg_t game_send_score(float score);
16
17
18 #endif /* GAME_H_ */
```

## 6.6 halconf.h

```
1 /*
2     ChibiOS - Copyright (C) 2006..2015 Giovanni Di Sirio
3
4     Licensed under the Apache License, Version 2.0 (the "License");
5     you may not use this file except in compliance with the License.
6     You may obtain a copy of the License at
7
8         http://www.apache.org/licenses/LICENSE-2.0
9
10     Unless required by applicable law or agreed to in writing, software
11     distributed under the License is distributed on an "AS IS" BASIS,
12     WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13     See the License for the specific language governing permissions and
14     limitations under the License.
15  */
16
28 #ifndef _HALCONF_H_
29 #define _HALCONF_H_
30
31 #include "mcuconf.h"
32
36 #if !defined(HAL_USE_PAL) || defined(__DOXYGEN__)
37 #define HAL_USE_PAL                 TRUE
38 #endif
39
43 #if !defined(HAL_USE_ADC) || defined(__DOXYGEN__)
44 #define HAL_USE_ADC                 TRUE
45 #endif
46
50 #if !defined(HAL_USE_CAN) || defined(__DOXYGEN__)
51 #define HAL_USE_CAN                 TRUE
52 #endif
53
57 #if !defined(HAL_USE_DAC) || defined(__DOXYGEN__)
58 #define HAL_USE_DAC                 TRUE
59 #endif
60
64 #if !defined(HAL_USE_DCMI) || defined(__DOXYGEN__)
65 #define HAL_USE_DCMI                 TRUE
66 #endif
67
71 #if !defined(HAL_USE_EXT) || defined(__DOXYGEN__)
72 #define HAL_USE_EXT                 TRUE
73 #endif
74
```

```
78 #if !defined(HAL_USE_GPT) || defined(__DOXYGEN__)
79 #define HAL_USE_GPT                 TRUE
80 #endif
81
85 #if !defined(HAL_USE_I2C) || defined(__DOXYGEN__)
86 #define HAL_USE_I2C                 TRUE
87 #endif
88
92 #if !defined(HAL_USE_I2S) || defined(__DOXYGEN__)
93 #define HAL_USE_I2S                 TRUE
94 #endif
95
99 #if !defined(HAL_USE_ICU) || defined(__DOXYGEN__)
100 #define HAL_USE_ICU                 FALSE
101 #endif
102
106 #if !defined(HAL_USE_MAC) || defined(__DOXYGEN__)
107 #define HAL_USE_MAC                 FALSE
108 #endif
109
113 #if !defined(HAL_USE_MMC_SPI) || defined(__DOXYGEN__)
114 #define HAL_USE_MMC_SPI             FALSE
115 #endif
116
120 #if !defined(HAL_USE_PWM) || defined(__DOXYGEN__)
121 #define HAL_USE_PWM                 TRUE
122 #endif
123
127 #if !defined(HAL_USE_RTC) || defined(__DOXYGEN__)
128 #define HAL_USE_RTC                 FALSE
129 #endif
130
134 #if !defined(HAL_USE_SDC) || defined(__DOXYGEN__)
135 #define HAL_USE_SDC                 TRUE
136 #endif
137
141 #if !defined(HAL_USE_SERIAL) || defined(__DOXYGEN__)
142 #define HAL_USE_SERIAL              TRUE
143 #endif
144
148 #if !defined(HAL_USE_SERIAL_USB) || defined(__DOXYGEN__)
149 #define HAL_USE_SERIAL_USB          TRUE
150 #endif
151
155 #if !defined(HAL_USE_SPI) || defined(__DOXYGEN__)
156 #define HAL_USE_SPI                 TRUE
157 #endif
158
162 #if !defined(HAL_USE_UART) || defined(__DOXYGEN__)
163 #define HAL_USE_UART                FALSE
164 #endif
165
169 #if !defined(HAL_USE_USB) || defined(__DOXYGEN__)
170 #define HAL_USE_USB                 TRUE
171 #endif
172
173 /*===========================================================================*/
174 /* ADC driver related settings.                                              */
175 /*===========================================================================*/
176
181 #if !defined(ADC_USE_WAIT) || defined(__DOXYGEN__)
182 #define ADC_USE_WAIT                TRUE
183 #endif
184
189 #if !defined(ADC_USE_MUTUAL_EXCLUSION) || defined(__DOXYGEN__)
190 #define ADC_USE_MUTUAL_EXCLUSION    TRUE
191 #endif
192
193 /*===========================================================================*/
194 /* CAN driver related settings.                                              */
195 /*===========================================================================*/
196
200 #if !defined(CAN_USE_SLEEP_MODE) || defined(__DOXYGEN__)
201 #define CAN_USE_SLEEP_MODE          TRUE
202 #endif
203
204 /*===========================================================================*/
205 /* I2C driver related settings.                                              */
206 /*===========================================================================*/
207
211 #if !defined(I2C_USE_MUTUAL_EXCLUSION) || defined(__DOXYGEN__)
212 #define I2C_USE_MUTUAL_EXCLUSION    TRUE
213 #endif
214
215 /*===========================================================================*/
216 /* MAC driver related settings.                                              */
217 /*===========================================================================*/
```

```
218
222 #if !defined(MAC_USE_ZERO_COPY) || defined(__DOXYGEN__)
223 #define MAC_USE_ZERO_COPY           FALSE
224 #endif
225
229 #if !defined(MAC_USE_EVENTS) || defined(__DOXYGEN__)
230 #define MAC_USE_EVENTS              TRUE
231 #endif
232
233 /*===========================================================================*/
234 /* MMC_SPI driver related settings.                                          */
235 /*===========================================================================*/
236
245 #if !defined(MMC_NICE_WAITING) || defined(__DOXYGEN__)
246 #define MMC_NICE_WAITING            TRUE
247 #endif
248
249 /*===========================================================================*/
250 /* SDC driver related settings.                                              */
251 /*===========================================================================*/
252
257 #if !defined(SDC_INIT_RETRY) || defined(__DOXYGEN__)
258 #define SDC_INIT_RETRY              100
259 #endif
260
266 #if !defined(SDC_MMC_SUPPORT) || defined(__DOXYGEN__)
267 #define SDC_MMC_SUPPORT             FALSE
268 #endif
269
276 #if !defined(SDC_NICE_WAITING) || defined(__DOXYGEN__)
277 #define SDC_NICE_WAITING            TRUE
278 #endif
279
280 /*===========================================================================*/
281 /* SERIAL driver related settings.                                           */
282 /*===========================================================================*/
283
289 #if !defined(SERIAL_DEFAULT_BITRATE) || defined(__DOXYGEN__)
290 #define SERIAL_DEFAULT_BITRATE      115200
291 #endif
292
300 #if !defined(SERIAL_BUFFERS_SIZE) || defined(__DOXYGEN__)
301 #define SERIAL_BUFFERS_SIZE         16
302 #endif
303
304 /*===========================================================================*/
305 /* SERIAL_USB driver related setting.                                        */
306 /*===========================================================================*/
307
315 #if !defined(SERIAL_USB_BUFFERS_SIZE) || defined(__DOXYGEN__)
316 #define SERIAL_USB_BUFFERS_SIZE     256
317 #endif
318
319 /*===========================================================================*/
320 /* SPI driver related settings.                                              */
321 /*===========================================================================*/
322
327 #if !defined(SPI_USE_WAIT) || defined(__DOXYGEN__)
328 #define SPI_USE_WAIT                TRUE
329 #endif
330
335 #if !defined(SPI_USE_MUTUAL_EXCLUSION) || defined(__DOXYGEN__)
336 #define SPI_USE_MUTUAL_EXCLUSION    TRUE
337 #endif
338
339 #endif /* _HALCONF_H_ */
340
```

## 6.7 lightshow.h

```
1 /*
2  * @file        lightshow.h
3  * @brief       Led's pattern control library.
4  * @author      Karl Khalil
5  * @author      Joaquim Silveira
6  * @version     1.0
7  * @date        18 Apr 2022
8  * @copyright   GNU Public License
9  *
10  */
11 #ifndef LIGHTSHOW_H
12 #define LIGHTSHOW_H
13
```

```
14 void lightshow_init(void);
15 void lighshow_stop(void);
16
17 #endif
```

## 6.8 mcuconf.h

```
1 /*
2     ChibiOS - Copyright (C) 2006..2015 Giovanni Di Sirio
3
4     Licensed under the Apache License, Version 2.0 (the "License");
5     you may not use this file except in compliance with the License.
6     You may obtain a copy of the License at
7
8         http://www.apache.org/licenses/LICENSE-2.0
9
10     Unless required by applicable law or agreed to in writing, software
11     distributed under the License is distributed on an "AS IS" BASIS,
12     WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13     See the License for the specific language governing permissions and
14     limitations under the License.
15  */
16
17 #ifndef _MCUCONF_H_
18 #define _MCUCONF_H_
19
20 /*
21  * STM32F4xx drivers configuration.
22  * The following settings override the default settings present in
23  * the various device driver implementation headers.
24  * Note that the settings for each driver only have effect if the whole
25  * driver is enabled in halconf.h.
26  *
27  * IRQ priorities:
28  * 15...0       Lowest...Highest.
29  *
30  * DMA priorities:
31  * 0...3        Lowest...Highest.
32  */
33
34 #define STM32F4xx_MCUCONF
35
36 /*
37  * HAL driver system settings.
38  */
39 #define STM32_NO_INIT                       FALSE
40 #define STM32_HSI_ENABLED                   TRUE
41 #define STM32_LSI_ENABLED                   TRUE
42 #define STM32_HSE_ENABLED                   TRUE
43 #define STM32_LSE_ENABLED                   FALSE
44 #define STM32_CLOCK48_REQUIRED              TRUE
45 #define STM32_SW                            STM32_SW_PLL // SYSCLK source is PLL.
46 #define STM32_PLLSRC                        STM32_PLLSRC_HSE // 24 MHz.
47 #define STM32_PLLM_VALUE                    24 // VCO input freq = 8/PLLM = 1 MHz // (define
       STM32_SYSCLK)
48 #define STM32_PLLN_VALUE                    336 // VCO output freq = VCO input * PLLN = 336 MHz
49 #define STM32_PLLP_VALUE                    2 // Main PLL clock = VCO output / PLLP = 336/2 = 168 MHz
50 #define STM32_PLLQ_VALUE                    7 // USB, SDIO clock = VCO outpu / PLLQ = 48 MHz
51 #define STM32_HPRE                          STM32_HPRE_DIV1     // AHB prescaler => 168 MHz. (define
       STM32_HCLK)
52 #define STM32_PPRE1                         STM32_PPRE1_DIV4    // APB1 prescaler => 168/4 = 42 MHz.
       (define STM32_PCLK1)
53 #define STM32_PPRE2                         STM32_PPRE2_DIV2    // APB2 prescaler => 168/2 = 84 MHz.
       (define STM32_PCLK2)
54 #define STM32_RTCSEL                        STM32_RTCSEL_LSI
55 #define STM32_RTCPRE_VALUE                  8
56 #define STM32_MCO1SEL                       STM32_MCO1SEL_HSI
57 #define STM32_MCO1PRE                       STM32_MCO1PRE_DIV1
58 #define STM32_MCO2SEL                       STM32_MCO2SEL_SYSCLK
59 #define STM32_MCO2PRE                       STM32_MCO2PRE_DIV5
60 #define STM32_I2SSRC                        STM32_I2SSRC_PLLI2S
61 #define STM32_PLLI2SN_VALUE                 192
62 #define STM32_PLLI2SR_VALUE                 6 // PLLI2SCLK = VCO input * PLLI2SN / PLLI2SR = 1*192/6 = 32
       MHz
63 #define STM32_PVD_ENABLE                    FALSE
64 #define STM32_PLS                           STM32_PLS_LEV0
65 #define STM32_BKPRAM_ENABLE                 FALSE
66
67 /*
68  * ADC driver system settings.
69  */
70 #define STM32_ADC_ADCPRE                    ADC_CCR_ADCPRE_DIV8
71 #define STM32_ADC_USE_ADC1                  TRUE
```

```
72 #define STM32_ADC_USE_ADC2                  TRUE
73 #define STM32_ADC_USE_ADC3                  TRUE
74 #define STM32_ADC_ADC1_DMA_STREAM           STM32_DMA_STREAM_ID(2, 4)
75 #define STM32_ADC_ADC2_DMA_STREAM           STM32_DMA_STREAM_ID(2, 2)
76 #define STM32_ADC_ADC3_DMA_STREAM           STM32_DMA_STREAM_ID(2, 1)
77 #define STM32_ADC_ADC1_DMA_PRIORITY         2
78 #define STM32_ADC_ADC2_DMA_PRIORITY         2
79 #define STM32_ADC_ADC3_DMA_PRIORITY         2
80 #define STM32_ADC_IRQ_PRIORITY              6
81 #define STM32_ADC_ADC1_DMA_IRQ_PRIORITY     6
82 #define STM32_ADC_ADC2_DMA_IRQ_PRIORITY     6
83 #define STM32_ADC_ADC3_DMA_IRQ_PRIORITY     6
84
85 /*
86  * DAC driver system settings.
87  */
88 #define STM32_DAC_DUAL_MODE                 FALSE
89 #define STM32_DAC_USE_DAC1_CH1              FALSE
90 #define STM32_DAC_USE_DAC1_CH2              TRUE
91 #define STM32_DAC_DAC1_CH1_IRQ_PRIORITY     10
92 #define STM32_DAC_DAC1_CH2_IRQ_PRIORITY     10
93 #define STM32_DAC_DAC1_CH1_DMA_PRIORITY     2
94 #define STM32_DAC_DAC1_CH2_DMA_PRIORITY     2
95 #define STM32_DAC_DAC1_CH1_DMA_STREAM       STM32_DMA_STREAM_ID(1, 5)
96 #define STM32_DAC_DAC1_CH2_DMA_STREAM       STM32_DMA_STREAM_ID(1, 6)
97
98 /*
99  * CAN driver system settings.
100  */
101 #define STM32_CAN_USE_CAN1                  TRUE
102 #define STM32_CAN_USE_CAN2                  FALSE
103 #define STM32_CAN_CAN1_IRQ_PRIORITY         11
104 #define STM32_CAN_CAN2_IRQ_PRIORITY         11
105
106 /*
107  * DCMI driver system settings.
108  */
109 #define STM32_DCMI_USE_DCMI                 TRUE
110 #define STM32_DCMI_IRQ_PRIORITY             11
111 #define STM32_DCMI_DMA_PRIORITY             2
112 #define STM32_DCMI_DMA_IRQ_PRIORITY         12
113 #define STM32_DCMI_DMA_STREAM               STM32_DMA_STREAM_ID(2, 1)
114
115 /*
116  * EXT driver system settings.
117  */
118 #define STM32_EXT_EXTI0_IRQ_PRIORITY        6
119 #define STM32_EXT_EXTI1_IRQ_PRIORITY        6
120 #define STM32_EXT_EXTI2_IRQ_PRIORITY        6
121 #define STM32_EXT_EXTI3_IRQ_PRIORITY        6
122 #define STM32_EXT_EXTI4_IRQ_PRIORITY        6
123 #define STM32_EXT_EXTI5_9_IRQ_PRIORITY      6
124 #define STM32_EXT_EXTI10_15_IRQ_PRIORITY    6
125 #define STM32_EXT_EXTI16_IRQ_PRIORITY       6
126 #define STM32_EXT_EXTI17_IRQ_PRIORITY       15
127 #define STM32_EXT_EXTI18_IRQ_PRIORITY       6
128 #define STM32_EXT_EXTI19_IRQ_PRIORITY       6
129 #define STM32_EXT_EXTI20_IRQ_PRIORITY       6
130 #define STM32_EXT_EXTI21_IRQ_PRIORITY       15
131 #define STM32_EXT_EXTI22_IRQ_PRIORITY       15
132
133 /*
134  * GPT driver system settings.
135  */
136 #define STM32_GPT_USE_TIM1                  FALSE
137 #define STM32_GPT_USE_TIM2                  FALSE
138 #define STM32_GPT_USE_TIM3                  FALSE
139 #define STM32_GPT_USE_TIM4                  FALSE
140 #define STM32_GPT_USE_TIM5                  FALSE
141 #define STM32_GPT_USE_TIM6                  TRUE
142 #define STM32_GPT_USE_TIM7                  FALSE
143 #define STM32_GPT_USE_TIM8                  FALSE
144 #define STM32_GPT_USE_TIM9                  FALSE
145 #define STM32_GPT_USE_TIM11                 TRUE
146 #define STM32_GPT_USE_TIM12                 TRUE
147 #define STM32_GPT_USE_TIM14                 FALSE
148 #define STM32_GPT_TIM1_IRQ_PRIORITY         7
149 #define STM32_GPT_TIM2_IRQ_PRIORITY         7
150 #define STM32_GPT_TIM3_IRQ_PRIORITY         7
151 #define STM32_GPT_TIM4_IRQ_PRIORITY         7
152 #define STM32_GPT_TIM5_IRQ_PRIORITY         7
153 #define STM32_GPT_TIM6_IRQ_PRIORITY         7
154 #define STM32_GPT_TIM7_IRQ_PRIORITY         7
155 #define STM32_GPT_TIM8_IRQ_PRIORITY         7
156 #define STM32_GPT_TIM9_IRQ_PRIORITY         7
157 #define STM32_GPT_TIM11_IRQ_PRIORITY        7
158 #define STM32_GPT_TIM12_IRQ_PRIORITY        7
```

```
159 #define STM32_GPT_TIM14_IRQ_PRIORITY        7
160
161 /*
162  * I2C driver system settings.
163  */
164 #define STM32_I2C_USE_I2C1                  TRUE
165 #define STM32_I2C_USE_I2C2                  FALSE
166 #define STM32_I2C_USE_I2C3                  FALSE
167 #define STM32_I2C_BUSY_TIMEOUT              50
168 #define STM32_I2C_I2C1_RX_DMA_STREAM        STM32_DMA_STREAM_ID(1, 0)
169 #define STM32_I2C_I2C1_TX_DMA_STREAM        STM32_DMA_STREAM_ID(1, 7)
170 #define STM32_I2C_I2C2_RX_DMA_STREAM        STM32_DMA_STREAM_ID(1, 2)
171 #define STM32_I2C_I2C2_TX_DMA_STREAM        STM32_DMA_STREAM_ID(1, 7)
172 #define STM32_I2C_I2C3_RX_DMA_STREAM        STM32_DMA_STREAM_ID(1, 2)
173 #define STM32_I2C_I2C3_TX_DMA_STREAM        STM32_DMA_STREAM_ID(1, 4)
174 #define STM32_I2C_I2C1_IRQ_PRIORITY         5
175 #define STM32_I2C_I2C2_IRQ_PRIORITY         5
176 #define STM32_I2C_I2C3_IRQ_PRIORITY         5
177 #define STM32_I2C_I2C1_DMA_PRIORITY         3
178 #define STM32_I2C_I2C2_DMA_PRIORITY         3
179 #define STM32_I2C_I2C3_DMA_PRIORITY         3
180 #define STM32_I2C_DMA_ERROR_HOOK(i2cp)      osalSysHalt("DMA failure")
181
182 /*
183  * ICU driver system settings.
184  */
185 #define STM32_ICU_USE_TIM1                  FALSE
186 #define STM32_ICU_USE_TIM2                  FALSE
187 #define STM32_ICU_USE_TIM3                  FALSE
188 #define STM32_ICU_USE_TIM4                  FALSE
189 #define STM32_ICU_USE_TIM5                  FALSE
190 #define STM32_ICU_USE_TIM8                  FALSE
191 #define STM32_ICU_USE_TIM9                  FALSE
192 #define STM32_ICU_TIM1_IRQ_PRIORITY         7
193 #define STM32_ICU_TIM2_IRQ_PRIORITY         7
194 #define STM32_ICU_TIM3_IRQ_PRIORITY         7
195 #define STM32_ICU_TIM4_IRQ_PRIORITY         7
196 #define STM32_ICU_TIM5_IRQ_PRIORITY         7
197 #define STM32_ICU_TIM8_IRQ_PRIORITY         7
198 #define STM32_ICU_TIM9_IRQ_PRIORITY         7
199
200 /*
201  * MAC driver system settings.
202  */
203 #define STM32_MAC_TRANSMIT_BUFFERS          2
204 #define STM32_MAC_RECEIVE_BUFFERS           4
205 #define STM32_MAC_BUFFERS_SIZE              1522
206 #define STM32_MAC_PHY_TIMEOUT               100
207 #define STM32_MAC_ETH1_CHANGE_PHY_STATE     TRUE
208 #define STM32_MAC_ETH1_IRQ_PRIORITY         13
209 #define STM32_MAC_IP_CHECKSUM_OFFLOAD       0
210
211 /*
212  * PWM driver system settings.
213  */
214 #define STM32_PWM_USE_ADVANCED              FALSE
215 #define STM32_PWM_USE_TIM1                  FALSE
216 #define STM32_PWM_USE_TIM2                  TRUE
217 #define STM32_PWM_USE_TIM3                  TRUE
218 #define STM32_PWM_USE_TIM4                  TRUE
219 #define STM32_PWM_USE_TIM5                  TRUE
220 #define STM32_PWM_USE_TIM8                  FALSE
221 #define STM32_PWM_USE_TIM9                  FALSE
222 #define STM32_PWM_TIM1_IRQ_PRIORITY         7
223 #define STM32_PWM_TIM2_IRQ_PRIORITY         7
224 #define STM32_PWM_TIM3_IRQ_PRIORITY         7
225 #define STM32_PWM_TIM4_IRQ_PRIORITY         7
226 #define STM32_PWM_TIM5_IRQ_PRIORITY         7
227 #define STM32_PWM_TIM8_IRQ_PRIORITY         7
228 #define STM32_PWM_TIM9_IRQ_PRIORITY         7
229
230 /*
231  * SDC driver system settings.
232  */
233 #define STM32_SDC_SDIO_DMA_PRIORITY         3
234 #define STM32_SDC_SDIO_IRQ_PRIORITY         9
235 #define STM32_SDC_WRITE_TIMEOUT_MS          250
236 #define STM32_SDC_READ_TIMEOUT_MS           25
237 #define STM32_SDC_CLOCK_ACTIVATION_DELAY    10
238 #define STM32_SDC_SDIO_UNALIGNED_SUPPORT    TRUE
239 #define STM32_SDC_SDIO_DMA_STREAM           STM32_DMA_STREAM_ID(2, 3)
240
241 /*
242  * SERIAL driver system settings.
243  */
244 #define STM32_SERIAL_USE_USART1             FALSE
245 #define STM32_SERIAL_USE_USART2             FALSE
```

```
246 #define STM32_SERIAL_USE_USART3             TRUE
247 #define STM32_SERIAL_USE_UART4              FALSE
248 #define STM32_SERIAL_USE_UART5              FALSE
249 #define STM32_SERIAL_USE_USART6             FALSE
250 #define STM32_SERIAL_USART1_PRIORITY        12
251 #define STM32_SERIAL_USART2_PRIORITY        12
252 #define STM32_SERIAL_USART3_PRIORITY        12
253 #define STM32_SERIAL_UART4_PRIORITY         12
254 #define STM32_SERIAL_UART5_PRIORITY         12
255 #define STM32_SERIAL_USART6_PRIORITY        12
256
257 /*
258  * SPI driver system settings.
259  */
260 #define STM32_SPI_USE_SPI1                  TRUE
261 #define STM32_SPI_USE_SPI2                  FALSE
262 #define STM32_SPI_USE_SPI3                  FALSE
263 #define STM32_SPI_SPI1_RX_DMA_STREAM        STM32_DMA_STREAM_ID(2, 0)
264 #define STM32_SPI_SPI1_TX_DMA_STREAM        STM32_DMA_STREAM_ID(2, 5)
265 #define STM32_SPI_SPI2_RX_DMA_STREAM        STM32_DMA_STREAM_ID(1, 3)
266 #define STM32_SPI_SPI2_TX_DMA_STREAM        STM32_DMA_STREAM_ID(1, 4)
267 #define STM32_SPI_SPI3_RX_DMA_STREAM        STM32_DMA_STREAM_ID(1, 2)
268 #define STM32_SPI_SPI3_TX_DMA_STREAM        STM32_DMA_STREAM_ID(1, 5)
269 #define STM32_SPI_SPI1_DMA_PRIORITY         1
270 #define STM32_SPI_SPI2_DMA_PRIORITY         1
271 #define STM32_SPI_SPI3_DMA_PRIORITY         1
272 #define STM32_SPI_SPI1_IRQ_PRIORITY         10
273 #define STM32_SPI_SPI2_IRQ_PRIORITY         10
274 #define STM32_SPI_SPI3_IRQ_PRIORITY         10
275 #define STM32_SPI_DMA_ERROR_HOOK(spip)      osalSysHalt("DMA failure")
276
277 /*
278  * ST driver system settings.
279  */
280 #define STM32_ST_IRQ_PRIORITY               8
281 #define STM32_ST_USE_TIMER                  2
282
283 /*
284  * UART driver system settings.
285  */
286 #define STM32_UART_USE_USART1               FALSE
287 #define STM32_UART_USE_USART2               FALSE
288 #define STM32_UART_USE_USART3               FALSE
289 #define STM32_UART_USE_UART4                FALSE
290 #define STM32_UART_USE_UART5                FALSE
291 #define STM32_UART_USE_USART6               FALSE
292 #define STM32_UART_USART1_RX_DMA_STREAM     STM32_DMA_STREAM_ID(2, 5)
293 #define STM32_UART_USART1_TX_DMA_STREAM     STM32_DMA_STREAM_ID(2, 7)
294 #define STM32_UART_USART2_RX_DMA_STREAM     STM32_DMA_STREAM_ID(1, 5)
295 #define STM32_UART_USART2_TX_DMA_STREAM     STM32_DMA_STREAM_ID(1, 6)
296 #define STM32_UART_USART3_RX_DMA_STREAM     STM32_DMA_STREAM_ID(1, 1)
297 #define STM32_UART_USART3_TX_DMA_STREAM     STM32_DMA_STREAM_ID(1, 3)
298 #define STM32_UART_UART4_RX_DMA_STREAM      STM32_DMA_STREAM_ID(1, 2)
299 #define STM32_UART_UART4_TX_DMA_STREAM      STM32_DMA_STREAM_ID(1, 4)
300 #define STM32_UART_UART5_RX_DMA_STREAM      STM32_DMA_STREAM_ID(1, 0)
301 #define STM32_UART_UART5_TX_DMA_STREAM      STM32_DMA_STREAM_ID(1, 7)
302 #define STM32_UART_USART6_RX_DMA_STREAM     STM32_DMA_STREAM_ID(2, 2)
303 #define STM32_UART_USART6_TX_DMA_STREAM     STM32_DMA_STREAM_ID(2, 7)
304 #define STM32_UART_USART1_IRQ_PRIORITY      12
305 #define STM32_UART_USART2_IRQ_PRIORITY      12
306 #define STM32_UART_USART3_IRQ_PRIORITY      12
307 #define STM32_UART_UART4_IRQ_PRIORITY       12
308 #define STM32_UART_UART5_IRQ_PRIORITY       12
309 #define STM32_UART_USART6_IRQ_PRIORITY      12
310 #define STM32_UART_USART1_DMA_PRIORITY      0
311 #define STM32_UART_USART2_DMA_PRIORITY      0
312 #define STM32_UART_USART3_DMA_PRIORITY      0
313 #define STM32_UART_UART4_DMA_PRIORITY       0
314 #define STM32_UART_UART5_DMA_PRIORITY       0
315 #define STM32_UART_USART6_DMA_PRIORITY      0
316 #define STM32_UART_DMA_ERROR_HOOK(uartp)    osalSysHalt("DMA failure")
317
318 /*
319  * USB driver system settings.
320  */
321 #define STM32_USB_USE_OTG1                  TRUE
322 #define STM32_USB_USE_OTG2                  FALSE
323 #define STM32_USB_OTG1_IRQ_PRIORITY         14
324 #define STM32_USB_OTG2_IRQ_PRIORITY         14
325 #define STM32_USB_OTG1_RX_FIFO_SIZE         512
326 #define STM32_USB_OTG2_RX_FIFO_SIZE         1024
327 #define STM32_USB_OTG_THREAD_PRIO           NORMALPRIO+10
328 #define STM32_USB_OTG_THREAD_STACK_SIZE     128
329 #define STM32_USB_OTGFIFO_FILL_BASEPRI      0
330
331 /*
332  * I2S driver system settings.
```

```
333  */
334 #define STM32_I2S_USE_SPI2                  TRUE
335 #define STM32_I2S_USE_SPI3                  FALSE
336 #define STM32_I2S_SPI2_IRQ_PRIORITY         10
337 #define STM32_I2S_SPI3_IRQ_PRIORITY         10
338 #define STM32_I2S_SPI2_DMA_PRIORITY         1
339 #define STM32_I2S_SPI3_DMA_PRIORITY         1
340 #define STM32_I2S_SPI2_RX_DMA_STREAM        STM32_DMA_STREAM_ID(1, 3)
341 #define STM32_I2S_SPI2_TX_DMA_STREAM        STM32_DMA_STREAM_ID(1, 4)
342 #define STM32_I2S_SPI3_RX_DMA_STREAM        STM32_DMA_STREAM_ID(1, 2)
343 #define STM32_I2S_SPI3_TX_DMA_STREAM        STM32_DMA_STREAM_ID(1, 5)
344 #define STM32_I2S_DMA_ERROR_HOOK(i2sp)      osalSysHalt("DMA failure")
345 #define STM32_I2S_SPI2_MODE                 (STM32_I2S_MODE_MASTER | STM32_I2S_MODE_RX)
346 #define STM32_I2S_SPI3_MODE                 (STM32_I2S_MODE_SLAVE | STM32_I2S_MODE_RX)
347
348 /*
349  * SPI slave driver system settings.
350  */
351 #define STM32_SPI_USE_SPI3_SLAVE            TRUE
352 #define STM32_SPI_SPI3_SLAVE_RX_DMA_STREAM  STM32_DMA_STREAM_ID(1, 2)
353 #define STM32_SPI_SPI3_SLAVE_TX_DMA_STREAM  STM32_DMA_STREAM_ID(1, 5)
354 #define STM32_SPI_SPI3_SLAVE_DMA_PRIORITY   1
355 #define STM32_SPI_SPI3_SLAVE_IRQ_PRIORITY   10
356 #define STM32_SPI_SLAVE_DMA_ERROR_HOOK(spip) osalSysHalt("DMA failure")
357
358 #endif /* _MCUCONF_H_ */
```

## 6.9 music.h

```
1 /*
2  * @file        music.h
3  * @brief       Musical processing of the frequency data.
4  * @author      Karl Khalil
5  * @author      Joaquim Silveira
6  * @version     1.0
7  * @date        20 Apr 2022
8  * @copyright   GNU Public License
9  *
10  */
11
12 #ifndef MUSIC_H
13 #define MUSIC_H
14
15
16 void music_init(void);
17 void music_stop(void);
18 void play_song(void);
19 void stop_song(void);
20 void music_listen(void);
21 msg_t music_send_freq(float freq);
22 bool music_is_playing(void);
23
24
25 #endif /*MUSIC_H*/
```

## 6.10 pathing.h

```
1 /*
2  * @file        pathing.h
3  * @brief       Pathing library for positional control of the ePuck
4  * @author      Joaquim Silveira
5  * @version     1.0
6  * @date        18 Apr 2022
7  * @copyright   GNU Public License
8  *
9  */
10
11 #ifndef     PATHING_H_
12 #define     PATHING_H_
13
14 typedef enum{
15     PATH_TO_PLAYER1,
16     PATH_TO_PLAYER2,
17     PATHING,
18     PATHING_FINISHED
19 }pathing_option_t;
20
21 void pathing_init(pathing_option_t option);
22 void pathing_stop(void);
```

```
23 void pathing_set(pathing_option_t option);
24 void pathing_wait_finish(void);
25
26 #endif  /* PATHING_H_ */
27
```

## 6.11 photo.h

```
1 /*
2  * @file        pathing.h
3  * @brief       Photo capture library
4  * @author      Joaquim Silveira
5  * @version     1.0
6  * @date        23 Apr 2022
7  * @copyright   GNU Public License
8  *
9  */
10
11 #ifndef     PHOTO_H
12 #define     PHOTO_H
13
14 void photo_init(void);
15 void photo_wait_finish(void);
16 void photo_stop(void);
17
18 #endif /* PHOTO_H_ */
```

## 6.12 rng.h

```
1 /*
2  * @file        rng.h
3  * @brief       Random Number Generator Library
4  * @author      Joaquim Silveira
5  * @version     1.0
6  * @date        1 May 2022
7  * @copyright   GNU Public License
8  *
9  */
10
11 #ifndef     RNG_H_
12 #define     RNG_H_
13
14 void rng_stop(void);
15 void rng_init(void);
16 uint32_t rng_get(void);
17
18 #endif /* RNG_H_*/
```

# Index