Ein Bild, das Logo enthält.

Automatisch generierte Beschreibung

Gitcon Driver V1.0

Firmware Reference Manual - [EN]

Generated with Doxygen 1.9.5

Author Simon Grundner

[1 Data Structure Index 4](#_Toc130772803)

[1.1 Data Structures 4](#_Toc130772804)

[2 File Index 5](#_Toc130772805)

[2.1 File List 5](#_Toc130772806)

[3 Data Structure Documentation 6](#_Toc130772807)

[3.1 gitcon\_context\_t Struct Reference 6](#_Toc130772808)

[3.1.1 Detailed Description 6](#_Toc130772809)

[3.1.2 Field Documentation 6](#_Toc130772810)

[3.1.2.1 midi\_handle 6](#_Toc130772811)

[3.1.2.2 midi\_queue 6](#_Toc130772812)

[3.1.2.3 sampler 7](#_Toc130772813)

[3.2 i2s\_sampler\_t Struct Reference 7](#_Toc130772814)

[3.2.1 Detailed Description 7](#_Toc130772815)

[3.2.2 Field Documentation 7](#_Toc130772816)

[3.2.2.1 buffer 7](#_Toc130772817)

[3.2.2.2 buffer\_pos 8](#_Toc130772818)

[3.2.2.3 buffer\_size 8](#_Toc130772819)

[3.2.2.4 dma\_queue 8](#_Toc130772820)

[3.2.2.5 dsp\_queue 8](#_Toc130772821)

[3.3 midi\_config\_t Struct Reference 9](#_Toc130772822)

[3.3.1 Detailed Description 9](#_Toc130772823)

[3.3.2 Field Documentation 9](#_Toc130772824)

[3.3.2.1 baudrate 9](#_Toc130772825)

[3.3.2.2 rx\_io 9](#_Toc130772826)

[3.3.2.3 tx\_io 9](#_Toc130772827)

[3.3.2.4 uart\_num 9](#_Toc130772828)

[3.4 midi\_context\_t Struct Reference 10](#_Toc130772829)

[3.4.1 Detailed Description 10](#_Toc130772830)

[3.4.2 Field Documentation 10](#_Toc130772831)

[3.4.2.1 cfg 10](#_Toc130772832)

[3.5 midi\_message\_t Struct Reference 10](#_Toc130772833)

[3.5.1 Detailed Description 10](#_Toc130772834)

[3.5.2 Field Documentation 11](#_Toc130772835)

[3.5.2.1 channel 11](#_Toc130772836)

[3.5.2.2 param1 11](#_Toc130772837)

[3.5.2.3 param2 11](#_Toc130772838)

[3.5.2.4 status 11](#_Toc130772839)

[4 Structural Diagram 12](#_Toc130772840)

[5 File Documentation 13](#_Toc130772841)

[5.1 config.h File Reference 13](#_Toc130772842)

[5.1.1 Detailed Description 14](#_Toc130772843)

[5.1.2 Macro Definition Documentation 14](#_Toc130772844)

[5.1.2.1 ADC\_RES 14](#_Toc130772845)

[5.1.2.2 ADC\_RES\_BITS 14](#_Toc130772846)

[5.1.2.3 AUDIO\_BUFFER\_SIZE 14](#_Toc130772847)

[5.1.2.4 DMA\_CHAN 14](#_Toc130772848)

[5.1.2.5 F\_SAMPLE\_HZ 14](#_Toc130772849)

[5.1.2.6 FFT\_SIZE 14](#_Toc130772850)

[5.1.2.7 FFT\_WINDOW\_SIZE 15](#_Toc130772851)

[5.1.2.8 INTERNAL\_ADC\_CHANNEL 15](#_Toc130772852)

[5.1.2.9 INTERNAL\_ADC\_IO 15](#_Toc130772853)

[5.1.2.10 INTERNAL\_ADC\_UNIT 15](#_Toc130772854)

[5.1.2.11 MIDI\_BAUD 15](#_Toc130772855)

[5.1.2.12 MIDI\_RX 15](#_Toc130772856)

[5.1.2.13 MIDI\_TX 15](#_Toc130772857)

[5.1.2.14 MIDI\_UART 15](#_Toc130772858)

[5.1.2.15 SPI\_CS 15](#_Toc130772859)

[5.1.2.16 SPI\_DEV 16](#_Toc130772860)

[5.1.2.17 SPI\_MISO 16](#_Toc130772861)

[5.1.2.18 SPI\_MOSI 16](#_Toc130772862)

[5.1.2.19 SPI\_SCLK 16](#_Toc130772863)

[5.2 config.h 17](#_Toc130772864)

[5.3 i2s\_sampler.c File Reference 18](#_Toc130772865)

[5.3.1 Macro Definition Documentation 18](#_Toc130772866)

[5.3.1.1 READER\_TIMEOUT\_MS 18](#_Toc130772867)

[5.3.1.2 READER\_TIMEOUT\_TICKS 18](#_Toc130772868)

[5.3.1.3 RESAMPLE\_DENOMINATOR 18](#_Toc130772869)

[5.3.2 Function Documentation 18](#_Toc130772870)

[5.3.2.1 i2s\_sampler\_start() 18](#_Toc130772871)

[5.3.2.2 i2s\_sampler\_stop() 19](#_Toc130772872)

[5.4 i2s\_sampler.c 20](#_Toc130772873)

[5.5 i2s\_sampler.h File Reference 22](#_Toc130772874)

[5.5.1 Detailed Description 22](#_Toc130772875)

[5.5.2 Function Documentation 22](#_Toc130772876)

[5.5.2.1 i2s\_sampler\_start() 22](#_Toc130772877)

[5.6 i2s\_sampler.h 24](#_Toc130772878)

[5.7 midi.c File Reference 24](#_Toc130772879)

[5.8 midi.c 24](#_Toc130772880)

[5.9 midi.h File Reference 24](#_Toc130772881)

[5.10 midi.h 24](#_Toc130772882)

[5.11 midi\_utils.c File Reference 24](#_Toc130772883)

[5.12 midi\_utils.c 24](#_Toc130772884)

[5.13 gitcon.c File Reference 24](#_Toc130772885)

[5.14 gitcon.c 24](#_Toc130772886)

[5.15 gitcon.h File Reference 24](#_Toc130772887)

[5.16 gitcon.h 24](#_Toc130772888)

[5.17 main.c File Reference 24](#_Toc130772889)

[5.18 main.c 24](#_Toc130772890)

[5.19 test\_fft.c File Reference 24](#_Toc130772891)

[5.20 test\_fft.c 24](#_Toc130772892)

[5.21 test\_midi.c File Reference 24](#_Toc130772893)

[5.22 test\_midi.c 24](#_Toc130772894)

[6 Index 25](#_Toc130772895)

# Data Structure Index

## Data Structures

Here are the data structures with brief descriptions:

gitcon\_context\_t

Driver passing parameters 6

i2s\_sampler\_t

Sampler Data 7

midi\_config\_t

MIDI Bus configuration 9

midi\_context\_t

MIDI configuration wrapper 10

midi\_message\_t

MIDI message data 10

# File Index

## File List

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

MTAP-MIDI-Guitar-Converter/firmware/Gitcon\_Driver/include/config.h

Gitcon Driver Configuration and Pin-Out File and dependencies 13

MTAP-MIDI-Guitar-Converter/firmware/Gitcon\_Driver/lib/audio/i2s\_sampler.c

I2S Sampler Driver Source for ESP32 18

MTAP-MIDI-Guitar-Converter/firmware/Gitcon\_Driver/lib/audio/i2s\_sampler.h

I2S Sampler Driver for ESP32 22

MTAP-MIDI-Guitar-Converter/firmware/Gitcon\_Driver/lib/midi/midi.c

MIDI Driver Source for ESP32 22

MTAP-MIDI-Guitar-Converter/firmware/Gitcon\_Driver/lib/midi/midi.h

MIDI UART Driver for ESP32 22

MTAP-MIDI-Guitar-Converter/firmware/Gitcon\_Driver/lib/midi/midi\_utils.c

MIDI Message Utilities 22

MTAP-MIDI-Guitar-Converter/firmware/Gitcon\_Driver/src/gitcon.c

Gitcon Driver Source 22

MTAP-MIDI-Guitar-Converter/firmware/Gitcon\_Driver/src/gitcon.h

Gitcon Driver 22

MTAP-MIDI-Guitar-Converter/firmware/Gitcon\_Driver/src/main.c

Main File for Gitcon Project 22

MTAP-MIDI-Guitar-Converter/firmware/Gitcon\_Driver/test/test\_fft/test\_fft.c

Unit Test for FFT 22

MTAP-MIDI-Guitar-Converter/firmware/Gitcon\_Driver/test/test\_midi/test\_midi.c

Unit Test for MIDI Driver 22

# Data Structure Documentation

## gitcon\_context\_t Struct Reference

Driver passing parameters.

#include <gitcon.h>

Data Fields

* i2s\_sampler\_t \*sampler
* midi\_handle\_t midi\_handle
* QueueHandle\_t midi\_queue

### Detailed Description

This data structure is passed around to all tasks to reconfigure driver parameters as needed.

Parameters

|  |  |
| --- | --- |
| sampler | I2S Sampler Handler |
| midi\_handle | MIDI Driver Context (MIDI over UART) |
| midi\_queue | MIDI Queue Handler |

* Definition at line 25 of file gitcon.h.

### Field Documentation

#### midi\_handle

midi\_handle\_t midi\_handle

MIDI driver context containing UART bus- and pin configuration.

* Definition at line 32 of file gitcon.h.

#### midi\_queue

QueueHandle\_t midi\_queue

MIDI-Messages are queued here to be written onto the UART bus by the MIDI driver.

* Definition at line 33 of file gitcon.h.

#### sampler

i2s\_sampler\_t\* sampler

Audio sampling parameters

* Definition at line 30 of file gitcon.h.

## i2s\_sampler\_t Struct Reference

Sampler Data.

#include <i2s\_sampler.h>

Data Fields

* QueueHandle\_t dma\_queue
* QueueHandle\_t dma\_queue
* QueueHandle\_t dsp\_queue
* size\_t ∗ buffer
* size\_t buffer\_pos
* size\_t buffer\_size

### Detailed Description

This struct contains the relevant data to sample audio correctly.

Parameters

|  |  |
| --- | --- |
| dma\_queue | Samples are sent to this queue by the DMA |
| dsp\_queue | Sampling result is sent to this queue |
| buffer | Buffer to store samples in |
| buffer\_pos | Current position in buffer |
| buffer\_size | Size of the buffer in samples |

* Definition at line 20 of file i2s\_sampler.h.

### Field Documentation

#### buffer

size\_t∗ buffer

The buffer stores sampled audio data with a size of 16 bit per sample due to the ADC resolution of 12 bit.

* Definition at line 24 of file i2s\_sampler.h.

#### buffer\_pos

size\_t buffer\_pos

To append incoming data correctly, a buffer position variable stores the index of where the DMA Controller left off.

* Definition at line 25 of file i2s\_sampler.h.

#### buffer\_size

size\_t buffer\_size

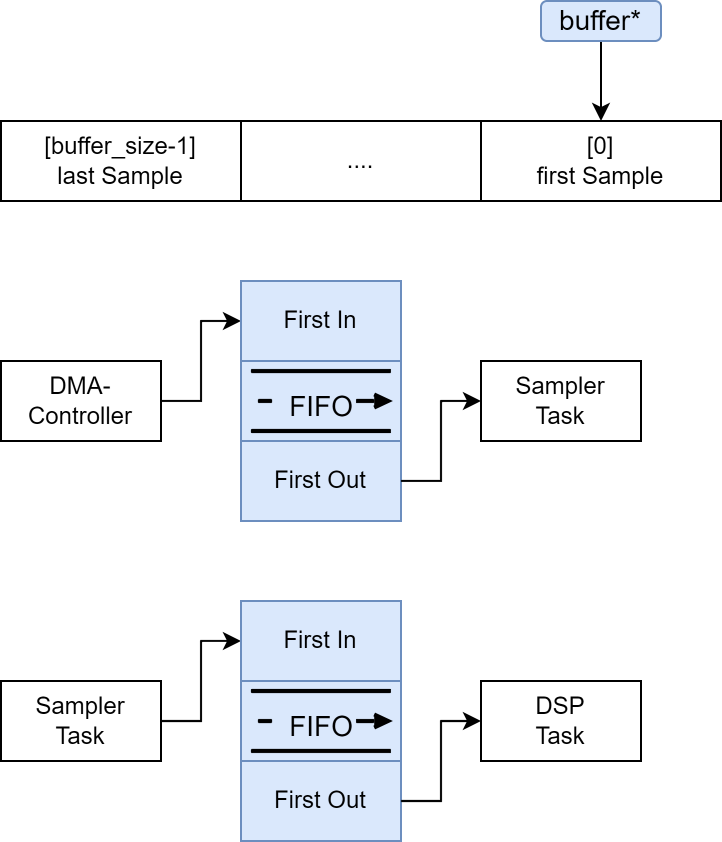
Stores the size of the buffer in Samples. This configuration is usually set during the initialization phase and not changed afterwards.

* Definition at line 26 of file i2s\_sampler.h.

#### dma\_queue

QueueHandle\_t dma\_queue

Raw data sent by the Direct-Memory-Access-Controller is read from a queue by the sampler.

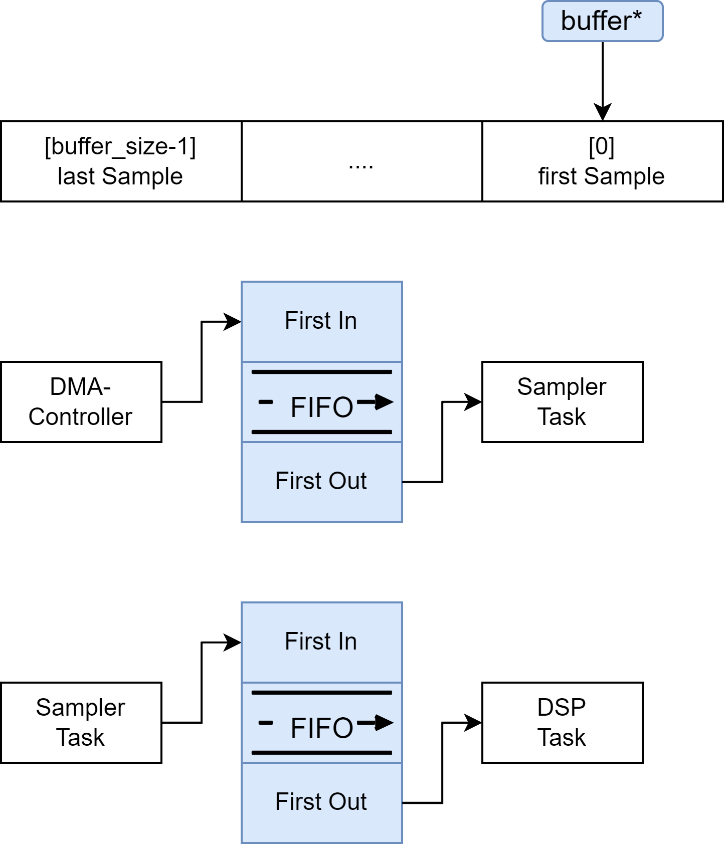


* Definition at line 22 of file i2s\_sampler.h.

#### dsp\_queue

QueueHandle\_t dsp\_queue

The buffer is sent into a queue for further processing.



* Definition at line 23 of file i2s\_sampler.h.

## midi\_config\_t Struct Reference

MIDI UART Configuration

#include <midi.h>

Data Fields

* uart\_port\_t uart\_num
* uint baudrate
* gpio\_num\_t rx\_io
* gpio\_num\_t tx\_io

### Detailed Description

UART Driver Configuration for MIDI transmission

Parameters

|  |  |
| --- | --- |
| uart\_num | UART Port |
| Baudrate | UART Baudrate |
| rx\_io | UART RX Pin |
| tx\_io | UART TX Pin |

* Definition at line 76 of file midi.h.

### Field Documentation

#### baudrate

uint baudrate

Baud rate configuration for UART

* Definition at line 79 of file midi.h.

#### rx\_io

gpio\_num\_t rx\_io

Receive GPIO pin number.

* Definition at line 80 of file midi.h.

#### tx\_io

gpio\_num\_t tx\_io

Transmit GPIO pin number.

* Definition at line 81 of file midi.h.

#### uart\_num

uart\_port\_t uart\_num

UART port id of the ESP32

* Definition at line 78 of file midi.h.

## midi\_context\_t Struct Reference

Wrapper for midi\_config\_t

Data Fields

* midi\_config\_t cfg

### Detailed Description

gitcon\_context\_t references the wrapper struct.

Parameters:

|  |  |
| --- | --- |
| cfg | MIDI Config |

* Definition at line 24 of file midi.c.

### Field Documentation

#### cfg

midi\_config\_t cfg

* Definition at line 26 of file midi.c.

## midi\_message\_t Struct Reference

MIDI Message data.

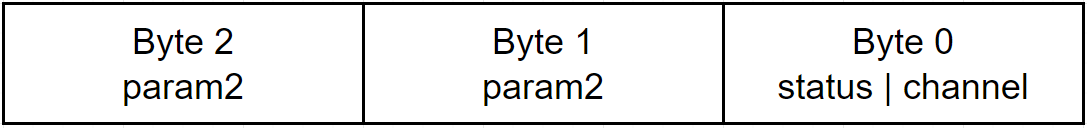
#include <midi.h>

Data Fields

* uint8\_t param1
* midi\_status\_t status
* uint8\_t channel
* uint8\_t param2

### Detailed Description

MIDI Message Data Struct



Parameters

|  |  |
| --- | --- |
| status | MIDI Status Byte |
| channel | MIDI Channel |
| param1 | MIDI Parameter 1 |
| param2 | MIDI Parameter 2 |

* Definition at line 60 of file midi.h.

### Field Documentation

#### channel

uint8\_t channel

Addresses a specific Channel on the MIDI-receiver side.

* Definition at line 64 of file midi.h.

#### param1

uint8\_t param1

Different functions depending on the status.

* Definition at line 62 of file midi.h.

#### param2

uint8\_t param2

Different functions depending on the status.

* Definition at line 65 of file midi.h.

#### status

midi\_status\_t status

* Definition at line 63 of file midi.h.

# Structural Diagram

Ein Bild, das Diagramm enthält.

Automatisch generierte Beschreibung

The Structural Flowchart Diagram above describes the **initialization setup** (in red), as well as the **repetitive** **process tasks** (in yellow). Once the firmware has been reset, an initialization is executed and all initial parameters are loaded into the driver parameter handler, which serves as a central data struct. When fully booted up and the real-time operating system has started, a continuous stream of data is sampled and processed, until a MIDI note is ready to be sent onto the UART bus.

# File Documentation

## config.h File Reference



Macros



### Detailed Description

Gitcon Driver Configuration and Pin-Out File and dependencies.

|  |  |  |  |
| --- | --- | --- | --- |
| Author | Version | Date | Copyright |
| @s-grundner | 0.1 | 2022-12-24 | Copyright (c) 2022 |

* Definition in file config.h.

### Macro Definition Documentation

#### ADC\_RES

#define ADC\_RES (1 << ADC\_RES\_BITS)

ADC Resolution ()

* Definition at line 50 of file config.h.

#### ADC\_RES\_BITS

#define ADC\_RES\_BITS 12

ADC Resolution in bit

* Definition at line 49 of file config.h.

#### AUDIO\_BUFFER\_SIZE

#define AUDIO\_BUFFER\_SIZE 512

Maximum number of samples stored in the audio buffer.

* Definition at line 55 of file config.h.

#### DMA\_CHAN

#define DMA\_CHAN 1

Direct memory access channel

* Definition at line 48 of file config.h.

#### F\_SAMPLE\_HZ

#define F\_SAMPLE\_HZ 15000

Sample rate at which the sampler reads audio data.

* Definition at line 56 of file config.h.

#### FFT\_SIZE

#define FFT\_SIZE (AUDIO\_BUFFER\_SIZE ∗ FFT\_WINDOW\_SIZE)

Number of samples the Fast Fourier Transform can process.

* Definition at line 58 of file config.h.

#### FFT\_WINDOW\_SIZE

#define FFT\_WINDOW\_SIZE 4

Factor by which the Fast Fourier Transform buffer is larger than the audio buffer.

* Definition at line 57 of file config.h.

#### INTERNAL\_ADC\_CHANNEL

#define INTERNAL\_ADC\_CHANNEL (ADC\_CHANNEL\_5)

Internal ADC channel. Corresponds to a hardware GPIO on the ESP32 chip.

* Definition at line 52 of file config.h.

#### INTERNAL\_ADC\_IO

#define INTERNAL\_ADC\_IO (GPIO\_NUM\_33)

Internal Analog Digital Converter GPIO Pin number.

* Definition at line 53 of file config.h.

#### INTERNAL\_ADC\_UNIT

#define INTERNAL\_ADC\_UNIT (ADC\_UNIT\_1)

Internal Analog Digital Converter unit.

* Definition at line 51 of file config.h.

#### MIDI\_BAUD

#define MIDI\_BAUD (115200)

Baud rate of the MIDI UART driver.

* Definition at line 44 of file config.h.

#### MIDI\_RX

#define MIDI\_RX (GPIO\_NUM\_27)

MIDI Driver receive GPIO.

* Definition at line 46 of file config.h.

#### MIDI\_TX

#define MIDI\_TX (GPIO\_NUM\_26)

MIDI Driver receive GPIO.

* Definition at line 45 of file config.h.

#### MIDI\_UART

#define MIDI\_UART (UART\_NUM\_1)

* Definition at line 43 of file config.h.

#### SPI\_CS

#define SPI\_CS (GPIO\_NUM\_5)

* Definition at line 40 of file config.h.

#### SPI\_DEV

#define SPI\_DEV (VSPI\_HOST)

* Definition at line 41 of file config.h.

#### SPI\_MISO

#define SPI\_MISO (GPIO\_NUM\_19)

* Definition at line 38 of file config.h.

#### SPI\_MOSI

#define SPI\_MOSI (GPIO\_NUM\_23)

* Definition at line 37 of file config.h.

#### SPI\_SCLK

#define SPI\_SCLK (GPIO\_NUM\_18)

* Definition at line 39 of file config.h.

## config.h



## i2s\_sampler.c File Reference

### Macro Definition Documentation

#### READER\_TIMEOUT\_MS

#define READER\_TIMEOUT\_MS 10

Time in milliseconds for the I2S reader to timeout.

* Definition at line 16 of file i2s\_sampler.c.

#### READER\_TIMEOUT\_TICKS

#define READER\_TIMEOUT\_TICKS (READER\_TIMEOUT\_MS / portTICK\_PERIOD\_MS)

Converts milliseconds into RTOS ticks.

* Definition at line 17 of file i2s\_sampler.c.

#### RESAMPLE\_DENOMINATOR

#define RESAMPLE\_DENOMINATOR 2

Buffer divider for double buffering.

* Definition at line 15 of file i2s\_sampler.c.

### Function Documentation

#### i2s\_sampler\_start()



Starts a sampler Task that samples from the given ADC1 Channel and sends the samples to the given Queue.

Parameters

|  |  |
| --- | --- |
| adc1\_channel | ADC1 Channel to use (Only ADC1 Channels are supported) |
| recv\_queue | Queue to send samples to |
| buffer\_size | Size of the buffer in samples |
| f\_sample | Sample rate at which the sampler samples the audio |

Returns

i2s\_sampler\_t∗ Sampler context or NULL if failed.

* Definition at line 53 of file i2s\_sampler.c.

#### i2s\_sampler\_stop()



Deletes sampler Task, frees memory, and uninstalls ADC as well as I2S driver.

Parameters

|  |  |
| --- | --- |
| sampler | Sampler to stop. |

Returns

ESP\_OK if successful

* Definition at line 92 of file i2s\_sampler.c.

## i2s\_sampler.c





## i2s\_sampler.h File Reference

I2S audio sampler for ESP32.



Data Structures

* struct i2s\_sampler\_t
  + *Sampler Data.*

Functions

* i2s\_sampler\_t ∗i2s\_sampler\_start (adc\_channel\_t adc1\_channel, QueueHandle\_t recv\_queue, size\_t buffer\_size, size\_t f\_sample)
  + *Starts a sampler Task that samples from the given ADC1 channel and sends the samples to the given Queue.*
* esp\_err\_t i2s\_sampler\_stop (i2s\_sampler\_t ∗sampler)
  + *Deletes sampler Task, frees memory and uninstalls ADC as well as i2s driver.*

### Detailed Description

I2S audio sampler for ESP32.

|  |  |  |  |
| --- | --- | --- | --- |
| Author | Version | Date | Copyright |
| @s-grundner | 0.1 | 2022-12-24 | Copyright (c) 2022 |

* Definition in file i2s\_sampler.h.

### Function Documentation

#### i2s\_sampler\_start()



Starts a sampler Task that samples from the given ADC1 Channel and sends the samples to the given Queue.

Parameters

|  |  |
| --- | --- |
| adc1\_channel | ADC1 Channel to use (Only ADC1 Channels are supported) |
| recv\_queue | Queue to send samples to |
| buffer\_size | Size of the buffer in samples |
| f\_sample | Sample rate |

Returns

i2s\_sampler\_t∗ Sampler context or NULL if failed

* Definition at line 53 of file i2s\_sampler.c.

#### i2s\_sampler\_stop()



Deletes sampler Task, frees memory, and uninstalls ADC as well as I2S driver.

Parameters

|  |  |
| --- | --- |
| sampler | Sampler to stop. |

Returns

ESP\_OK if successful

* Definition at line 92 of file i2s\_sampler.c.

## i2s\_sampler.h

## midi.c File Reference

## midi.c

## midi.h File Reference

## midi.h

## midi\_utils.c File Reference

## midi\_utils.c

## gitcon.c File Reference

## gitcon.c

## gitcon.h File Reference

## gitcon.h

## main.c File Reference

## main.c

## test\_fft.c File Reference

## test\_fft.c

## test\_midi.c File Reference

## test\_midi.c

# Index