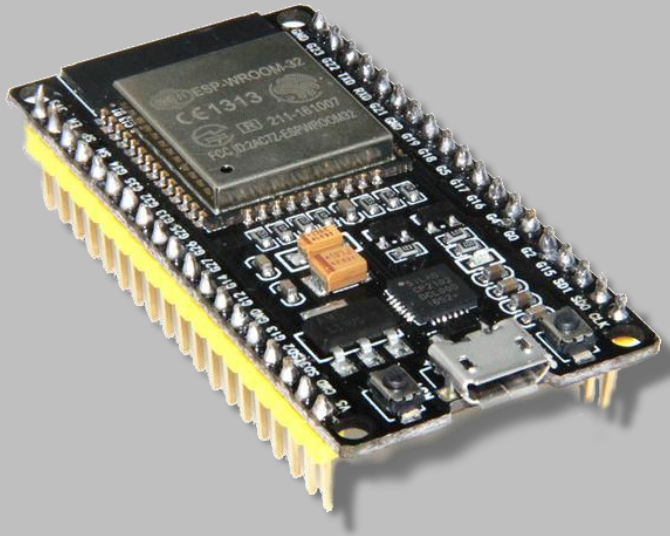


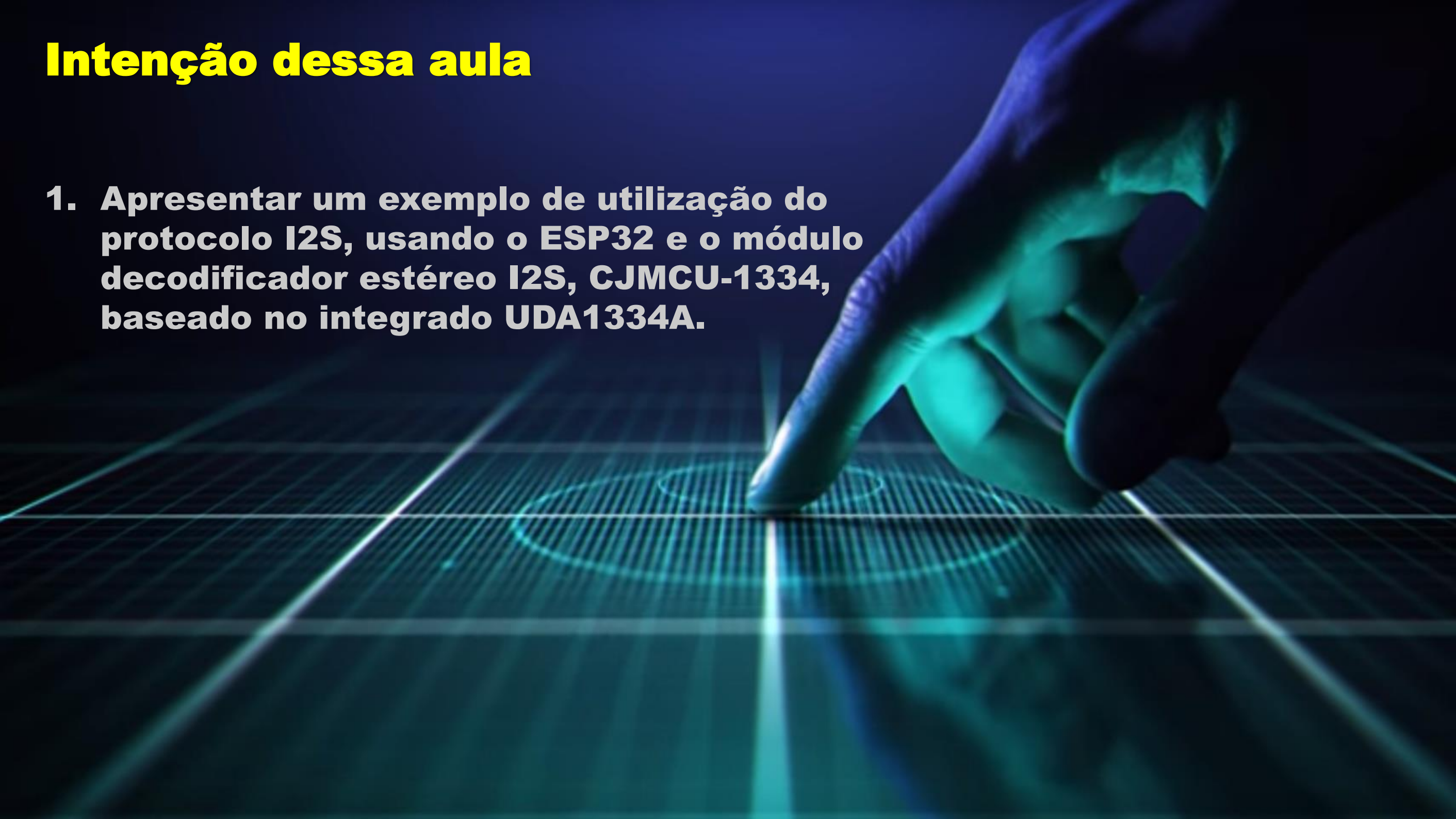
Inter-Integrated Circuit Sound e ESP32



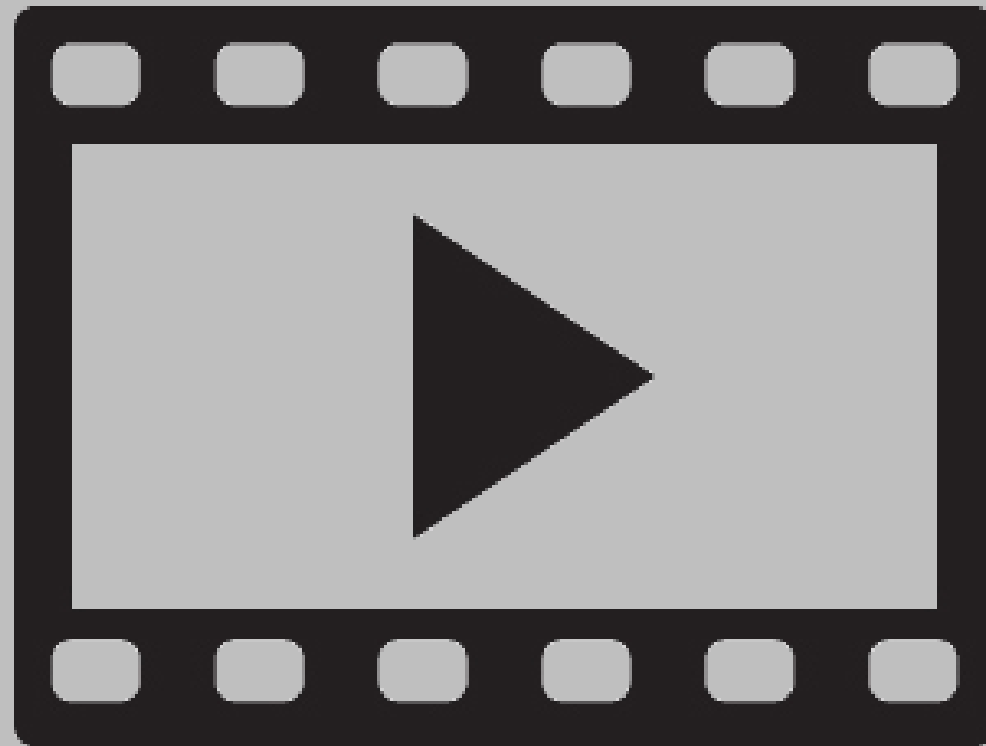
Por Fernando Koyanagi

Intenção dessa aula

- 1. Apresentar um exemplo de utilização do protocolo I2S, usando o ESP32 e o módulo decodificador estéreo I2S, CJMCU-1334, baseado no integrado UDA1334A.**

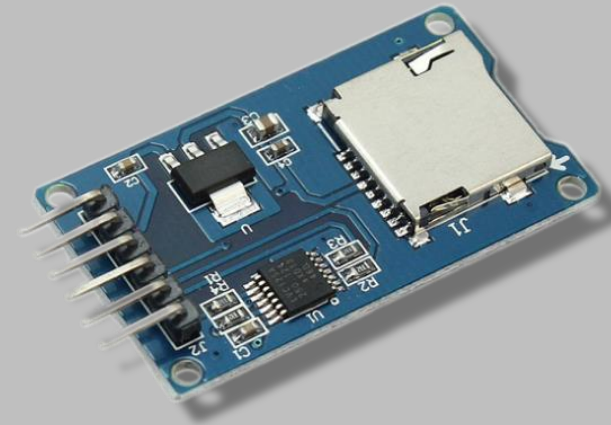
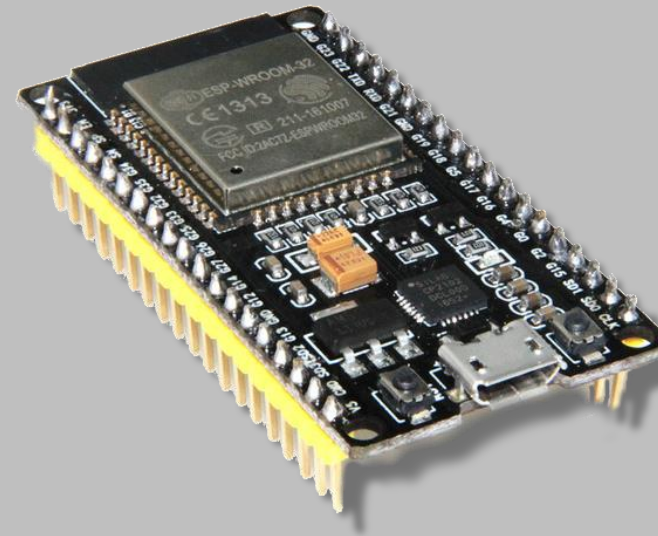
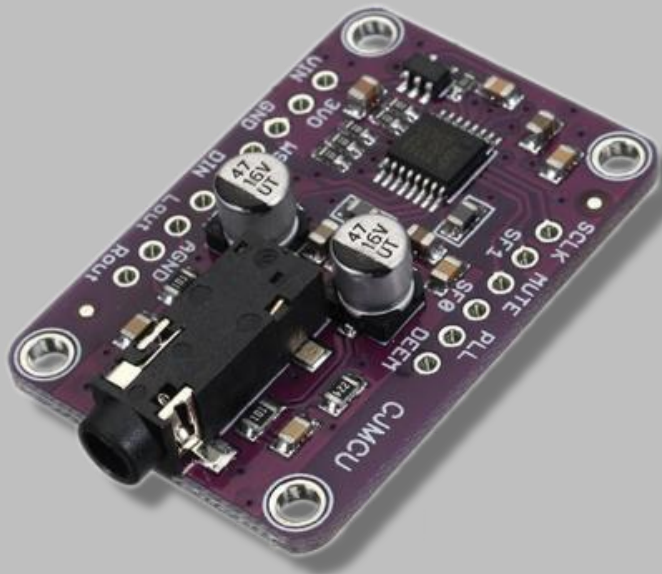


Demonstração



Recursos usados

- ESP32 Wroom 32.
- Módulo de cartão SD.
- Cartão SD (formatado FAT).
- Módulo decodificador CJMCU-1334.
- Fios.



FERNANDO K

Tutoriais
Tecnologia
Tendências

VÍDEOS EXCLUSIVOS PARA
O CLUBE DE MEMBROS
CLIQUE AQUI

FÓRUM

SOBRE

ARDUINO

ESP8266

ESP32

LORAWAN

STM32

MOTOR

DISPLAY


MATERIAL PARA DOWNLOAD

Receba o meu conteúdo
GRATUITAMENTE

Insira aqui seu melhor email...

QUERO RECEBER GRÁTIS

riden



SE EU SOUBESSE DISSO ANTES!


Se eu soubesse disso antes!

by **Fernando K** - 12 novembro

Voltamos a falar de instrumentação e, hoje, vamos falar de um módulo de obtenção de parâmetros elétricos, que é o RIDEN 100V/10A. ...

Leia mais

automacao



IOT MAIS BARATA DO
MUNDO COM ESP8266


IOT mais barata do mundo com ESP8266

by **Fernando K** - 05 novembro

Automação utilizando o ESP8266. Hoje quero te apresentar um exemplo de baixo custo utilizando este modelo de microcontrolador co...

Leia mais

ads5012h



OSCILOSCÓPIO 100MHZ PORTÁTIL:
INCRIVELMENTE BARATO!


Osciloscópio 100MHz portátil: Incrivelmente barato!

by **Fernando K** - 29 outubro

Conhece o osciloscópio ADS5012H - Daniu? Pois, eu ganhei um da Banggood e gostei demais. Primeiro: ele é barato. Segundo: a taxa de ...

Leia mais

iot



SPOILER
DO CURSO DE IOT

Spoiler do curso de IOT

by **Fernando K** - 22 outubro

Uma parte do curso de IOT que estou desenvolvendo. É o que mostro hoje para vocês: uma aula sobre Multitask, ou seja, sobre como ser pr...

Leia mais

Telegram

Supergrupo de colaboração entre meus seguidores.

Instagram

Conteúdo exclusivo, que não tem no Youtube!

INSTAGRAM @FERNANDOK_OFICIAL

FACEBOOK

Fernando K Tecnologia

27.909 curtidas


Curtir Página

Saiba mais

Seja o primeiro de seus amigos a curtir isso.

INSCREVA-SE NO YOUTUBE

Saiba tudo sobre Inter...



Fernando K Tecnologia

YouTube

999+

RECENTE

POPULAR

COMENTÁRIOS

Se eu soubesse disso antes!

Nov 12, 2019

SEJA MEMBRO

Links onde
comprei os
componentes

<https://bit.ly/2VM1ZdQ>

Em www.fernandok.com

5



Introdução ao #ESP32 - Parte 1

65.585 visualizações • 21 de nov. de 2017

3,1 MIL 38 COMPARTILHAR SALVAR



Fernando K Tecnologia
47,1 mil inscritos

SEJA MEMBRO

INSCRITO



Onde encontrar o ESP32 : <https://bit.ly/2sjBXRy>


Nunca compre só uma peça, porque se você queimar o seu componente, acabou a incadeira e
MOSTRAR MAIS

SEJA MEMBRO



SEJA MEMBRO

forum.fernandok.com



Fórum Fernando K Tecnologia
Fórum sobre dúvidas com relação ao conteúdo disponibilizado pelo Fernando Koyanagi

[www.fernandok.com](#) [/fernandokoyanagi](#) [/fernandokoyanagi](#) [/fernandok_oficial](#) [/fernandok_oficial](#)



Links rápidos















fernandokoyanagi

Bem-vindo: 05/Out/2018, 11:16

A sua última visita foi em 10/Set/2018, 15:47

Assinalar todos os fóruns como lidos

SUPPORT: FÓRUM FERNANDOK	TÓPICOS	MENSAGENS	ÚLTIMA MENSAGEM
 <div>Feedback Dúvidas, críticas ou sugestões sobre o Fórum FernandoK. Para demais questões utilize o fórum correto.</div>	6	11	Re: O russo voltou por Iperito  01/Out/2018, 08:25

FERNANDO K	TÓPICOS	MENSAGENS	ÚLTIMA MENSAGEM
 <div>Arduino Projetos de arduino</div>	31	79	skard y txgji por Soresorcem  05/Out/2018, 10:55
 <div>ESP32 Projetos de ESP32</div>	29	62	Duvidas sobre como instalar a... por Marcelo Jorge  04/Out/2018, 15:52
 <div>ESP8266 O ESP8266 é um microcontrolador do fabricante chinês Espressif que inclui capacidade de comunicação por Wi-Fi.</div>	24	51	Re: NodeMCU não conecta em qu... por reanailva  04/Out/2018, 14:39
 <div>LoRa Projetos com LoRa</div>	11	31	Projeto de irrigação de jardim por marlonc  04/Out/2018, 21:30
 <div>STM32 Projetos com STM32</div>	3	8	Re: Imprecisão de tempo de de... por biaroto  12/Set/2018, 09:15
 <div>Motor Projetos com motor</div>	5	11	Re: impressora 3d com motor dc por Magetron  24/Set/2018, 19:06
 <div>Display Projetos com Display</div>	4	11	Re: Alguem conhece o VIRTUINO... por Joel Luz  21/Set/2018, 11:39

QUEM ESTÁ ONLINE

No total, há **4** usuários online :: 2 usuários registrados, 0 anônimo e 2 visitantes (baseado em usuários ativos nos últimos 5 minutos)
O recorde de usuários online foi de **19** em 11/Set/2018, 05:37

Usuários registrados: alberto, **fernandokoyanagi**
Legenda: Administradores, Moderadores globais

ANIVERSÁRIOS

Não há aniversários hoje

ESTATÍSTICAS

Total de mensagens **703** • Total de tópicos **114** • Total de membros **469** • Novo usuário: **Soresorcem**

...

Powered by phpBB® Forum Software © phpBB Limited
Tradução por: Suporte phpBB
Painel de Controle da Administração



Instagram

fernandok_oficial



Telegram

fernandok_oficial



I2S.

O que é o I2S?

O I2S ou IIS, é um protocolo de transferência de dados (originalmente dados de áudio), criado pela Philips Semicondutores, com o objetivo de atender e servir como padrão para o emergente mercado de áudio digital no início dos anos 80.

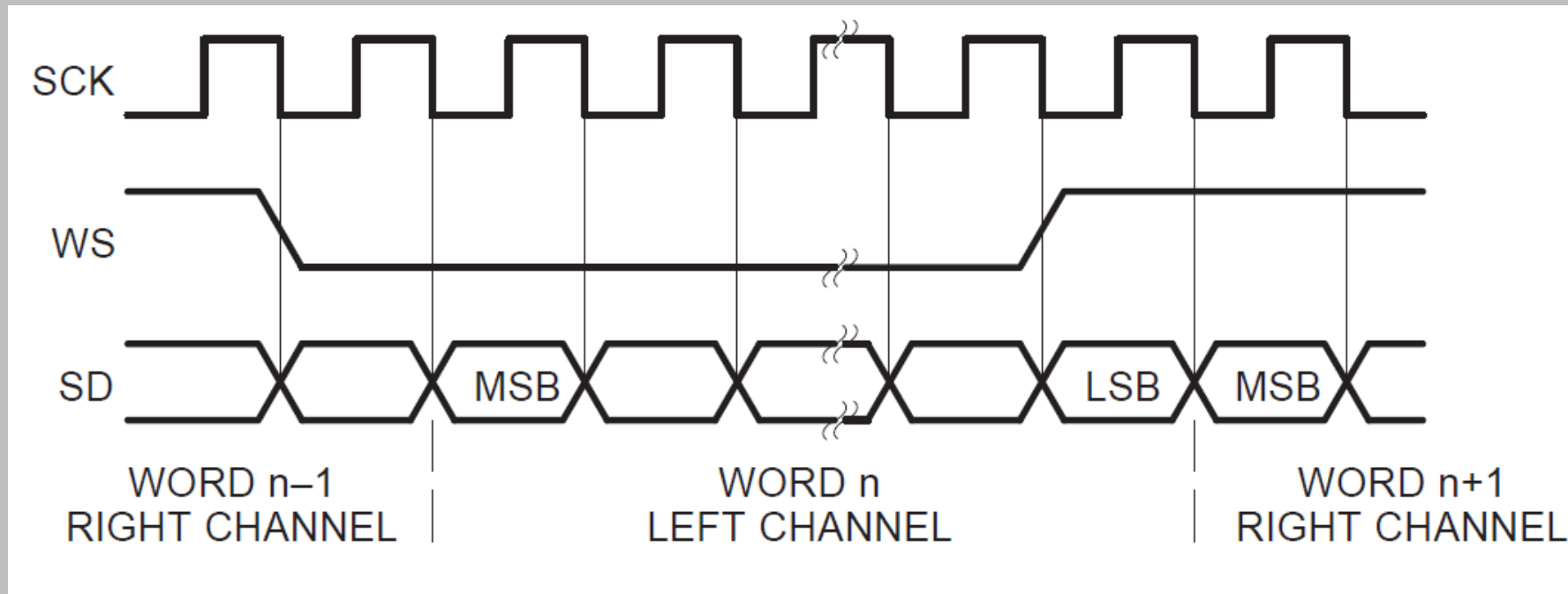
O Bus de dados I2S bus tem o objetivo de interligar circuitos integrados, daí o nome: **I**nter **I**ntegrated Circuit **S**ound Bus).

É formado por três linhas de sinais: um clock para sincronização, um sinal que indica se o canal é direito ou esquerdo e uma linha para os dados multiplexados.

Não confundir com o I2C, também da Philips Semicondutores, pois são distintos, embora muito próximos.

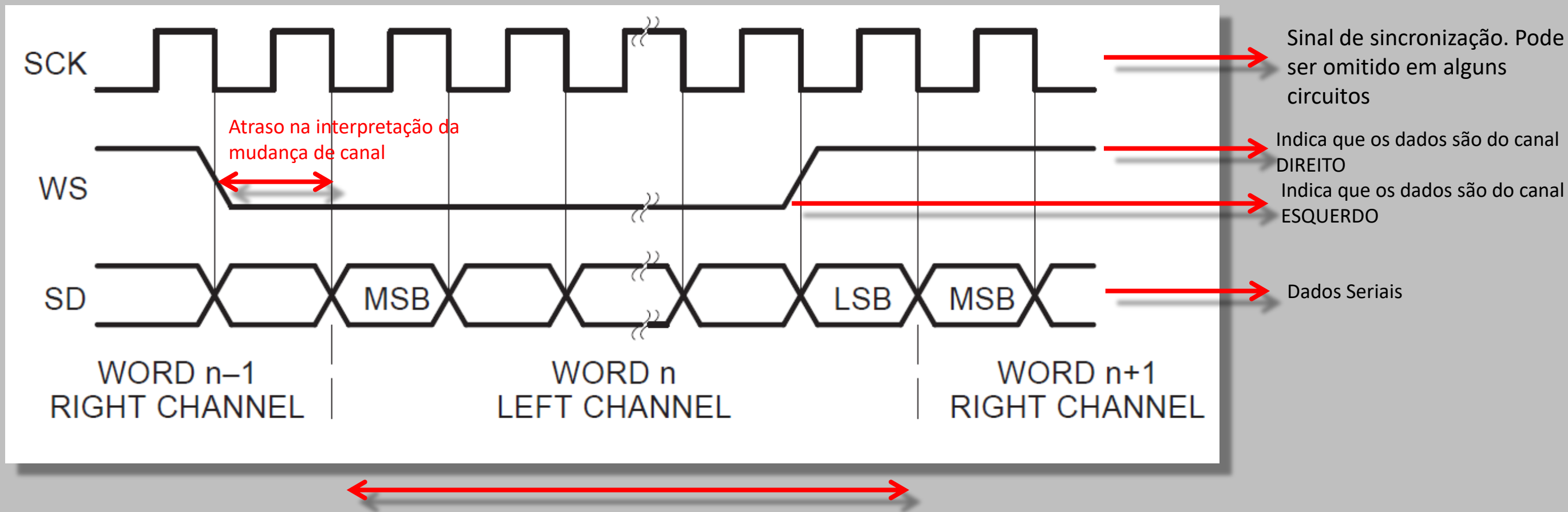
Bus I2S

- O sinal **SCK** é utilizado para a sincronização entre o transmissor e o receptor dos dados. Existe uma forma atual de I2S onde esse sinal é suprimido e a sincronização é realizada através da detecção do clock pelo receptor usando um circuito PLL.
- A linha **WS** indica se os dados referem-se ao canal direito (quando em nível alto) ou esquerdo (quando em nível baixo).
- A linha **SD** (Dados Seriais) contem os dados em si. Comumente uma série de amostras obtidas por um conversor analógico digital de certa profundidade. Estes dados são normalmente transmitidos usando uma Modulação por Código de Pulso (PCM), em algum dos seus vários formatos. (Não compactados como WAV e AIFF, ou compactados como MP3 e outros).



Bus I2S

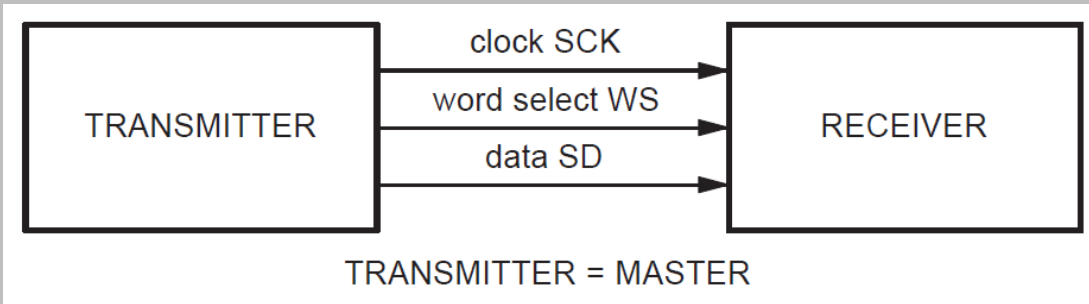
Mais alguns detalhes do barramento I2S, segundo a definição da Philips Semicondutores.



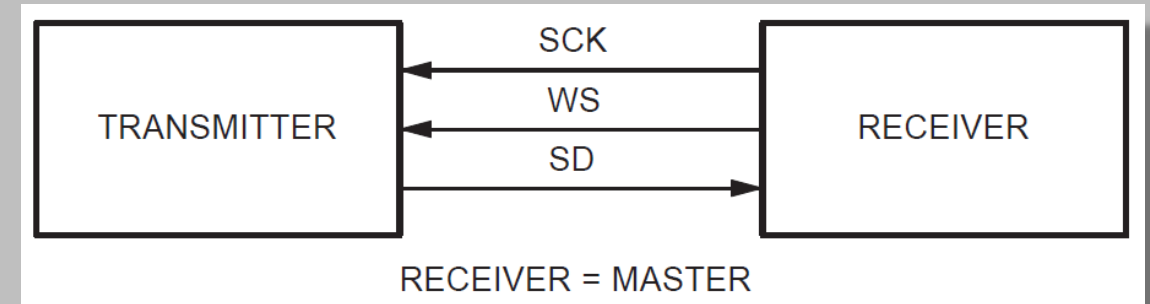
Uma palavra digital transmitida, sua profundidade depende da conversão (8, 16, ou 24 bits, por exemplo). O padrão original descreve que o bit mais significativo é transferido primeiro, mas atualmente existem variações. Note que há um atraso de um ciclo de clock entre a mudança do canal e sua interpretação na linha de dados. Isso para permitir a detecção e o chaveamento do circuito receptor.

Topologias sugeridas para o I2S

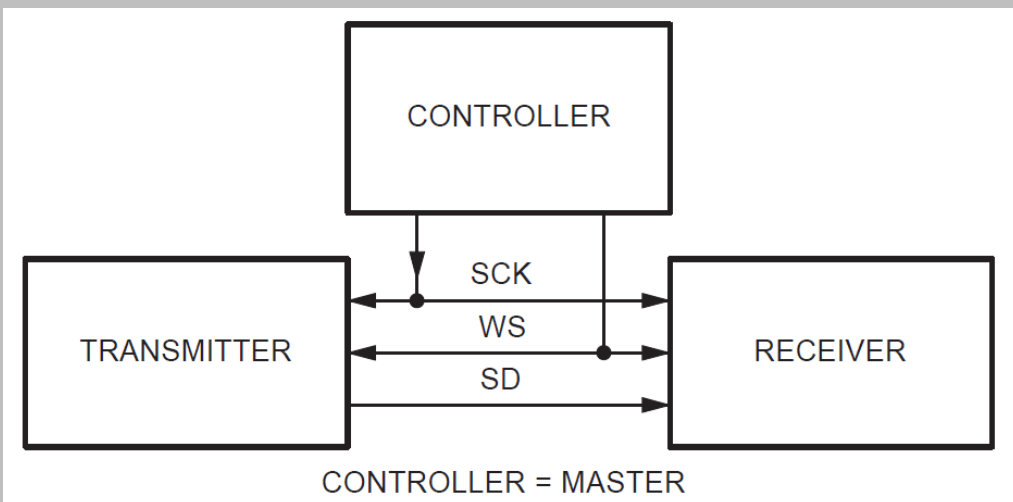
Ainda segundo a definição da Philips Semicondutores, as seguintes topologias poderiam ser utilizadas usando o I2S. O circuito que controla o sinal de clock e/ou sinal de canal, é considerado o mestre na topologia.



O **Transmissor** controla o clock e/ou canal, sendo o **mestre** nesta topologia



O **Receptor** controla o clock e/ou canal, sendo o **mestre** nesta topologia



O **Controlador Externo** controla o clock e/ou canal, sendo o **mestre** nesta topologia

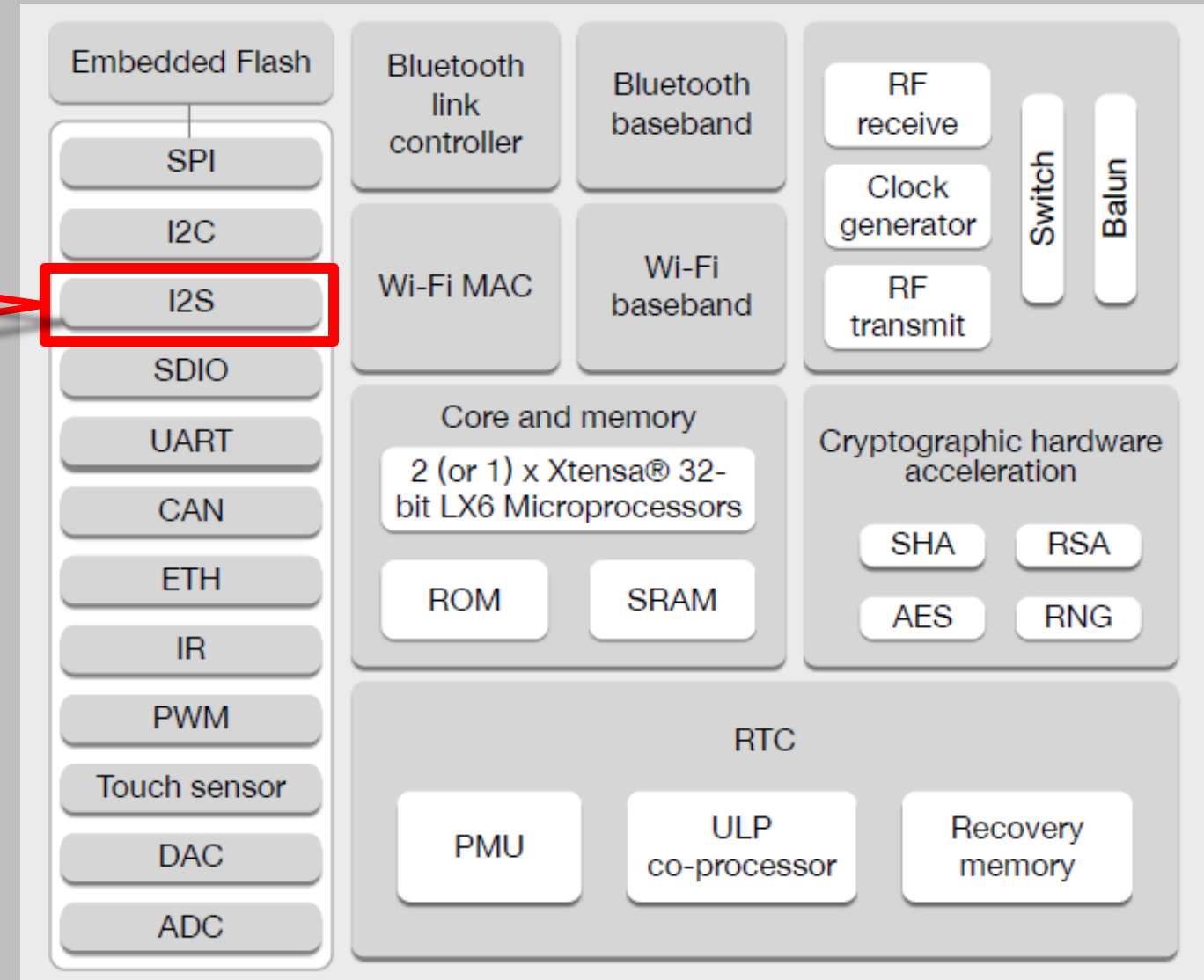
I2S no ESP32

I2S no ESP32

No diagrama de blocos do SoC Esp32 podemos notar a presença do periférico I2S... Neste caso, trata-se de dois periféricos I2S (I2S0 e I2S1), disponíveis.

I2S0	0x3FF4_F000	0x3FF4_FFFF	4 KB
I2S1	0x3FF6_D000	0x3FF6_DFFF	4 KB

- As duas interfaces I2S padrão disponíveis, pode operar em modo Mestre ou Escravo, em full-duplex ou Half-duplex.
- As resoluções podem ser de 8, 16, 32, 48 e 64 bits, como canais de entrada ou saída.
- O sinal de clock (BCK) vai de 10kHz a 40MHz.



UDA1334A

DAC de áudio de baixa potência com PLL

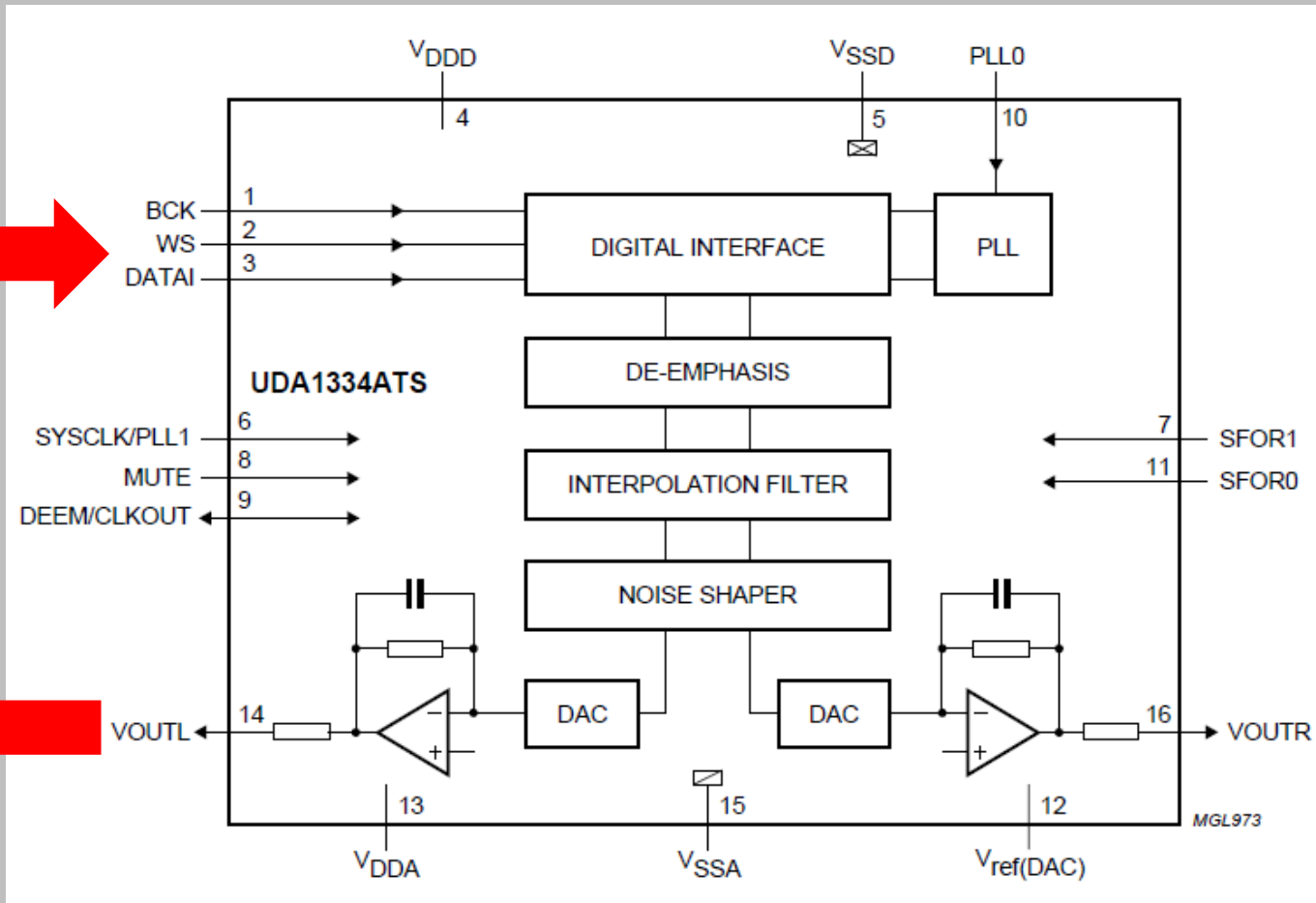
UDA1334 DAC de áudio de baixa potência com PLL

O módulo CJMCU-1334 é baseado no chip UDA1334, um DAC de áudio de dois canais. Por possuir um PLL, pode operar em sistemas onde o sinal de clock do I2S não está disponível.

Suporta I2S Bus com comprimento de palavra de até 24 bits MSB e LSB com comprimentos 16,20 e 24 bits.

A frequência de amostragem de 44,1kHz é suportada somente no modo MONO.

Entrada de sinais de **clock** (BCK), **seleção de canais** (WS) e **dados seriais** (DATAI)



Saída Esquerda



Saída Direita

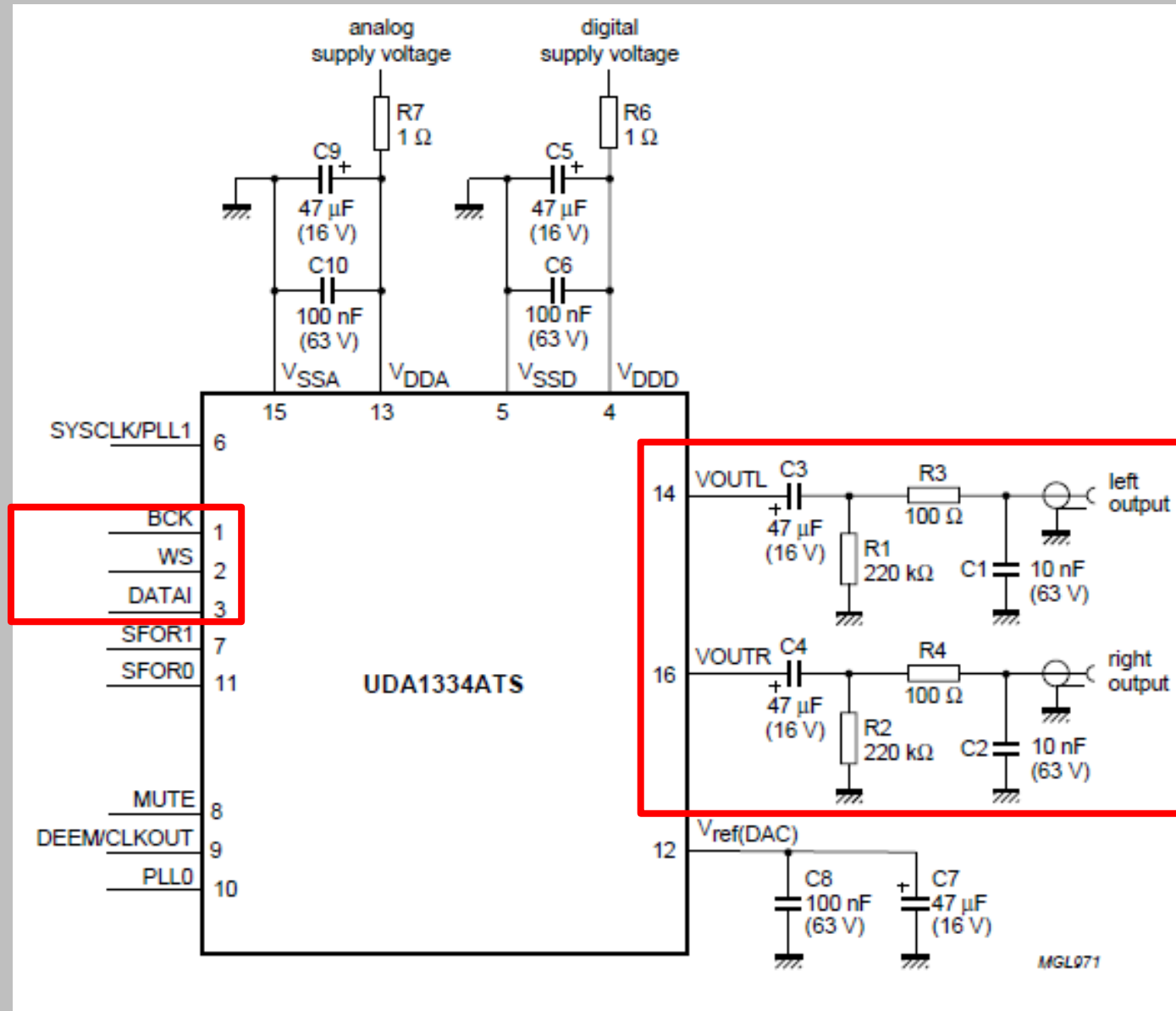


UDA1334 Sugestão de Aplicação

O datasheet do fabricante sugere o seguinte circuito para uma aplicação de áudio.

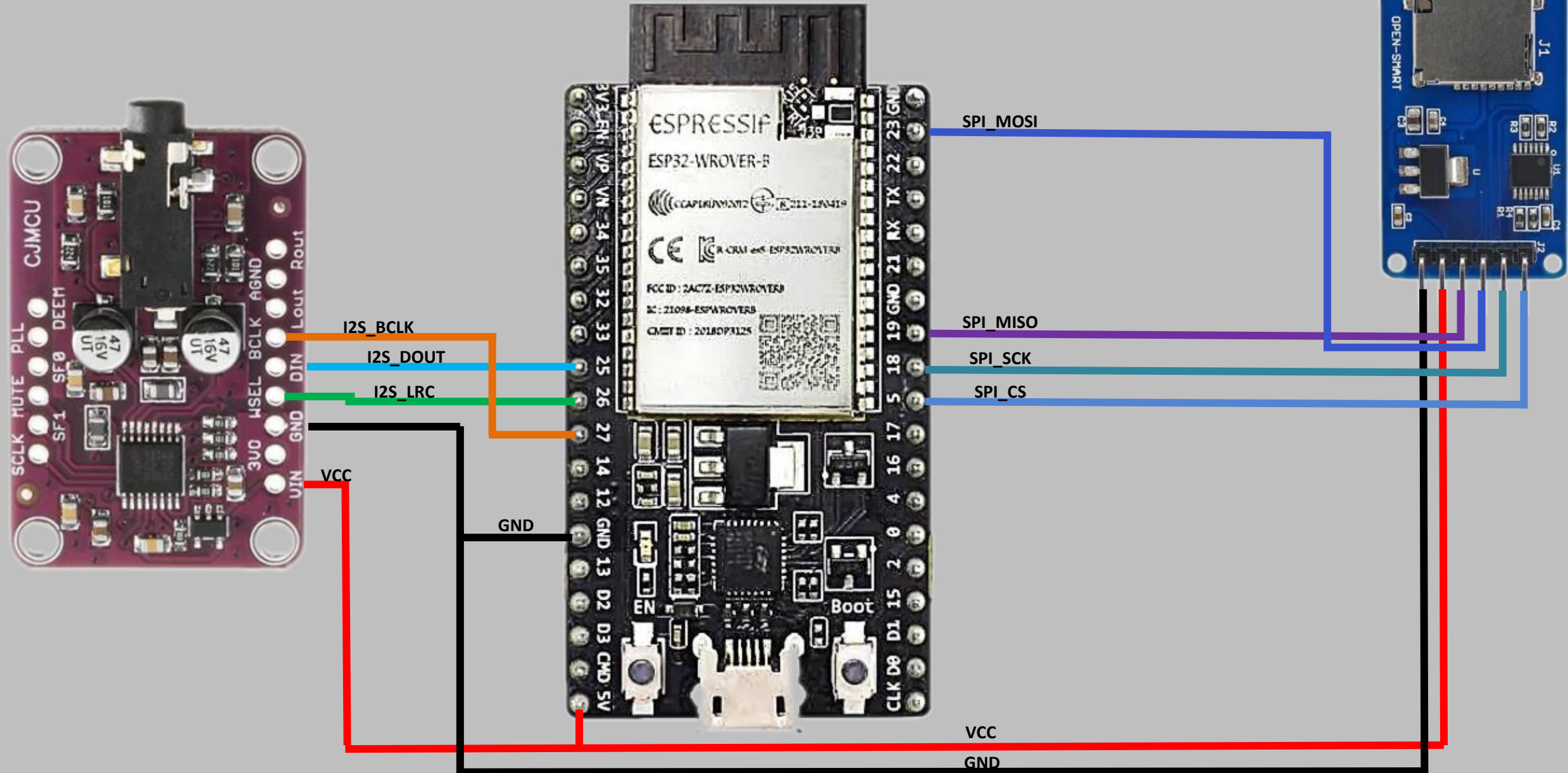
Os sinais I2S são introduzidos aqui.

Os outros sinais de controle são para configurações de modo do chip.



Os sinais de saída são obtidos após os circuitos de desacoplamento e filtragem

Circuito



Código-fonte

Código-fonte: Biblioteca

Para esse exemplo, vamos utilizar a biblioteca *ESP32-audiol2S* que pode ser encontrada no link abaixo.

O código será uma variação do exemplo apresentado na mesma biblioteca.

<https://github.com/schreibfaul1/ESP32-audiol2S>

Código-fonte: Declarações

```
//Bibliotecas utilizadas
#include "Arduino.h"
#include "WiFiMulti.h"
#include "Audio.h"
#include "SPI.h"
#include "SD.h"
#include "FS.h"

//Pinos de conexão do ESP32 e o módulo de cartão SD
#define SD_CS 5
#define SPI_MOSI 23
#define SPI_MISO 19
#define SPI_SCK 18

//Pinos de conexão do ESP32-I2S e o módulo I2S/DAC CJMCU 1334
#define I2S_DOUT 25
#define I2S_BCLK 27
#define I2S_LRC 26
```


Código-fonte: Declarações

```
//Cria o objeto que representará o áudio  
Audio audio;
```

```
//Cria o objeto que representará o Wi-Fi  
WiFiMulti wifiMulti;
```

```
// Definições da Rede Wi-Fi  
String ssid = "Seu_SSID";  
String password = "Sua_Senha";
```

Código-fonte: Setup()

```
void setup()
{
    //Inicia o Serial para debug
    Serial.begin(115200);

    //Configura e inicia o SPI para conexão com o cartão SD
    pinMode(SD_CS, OUTPUT);
    digitalWrite(SD_CS, HIGH);
    SPI.begin(SPI_SCK, SPI_MISO, SPI_MOSI);
    SPI.setFrequency(1000000);
    SD.begin(SD_CS);

    //Configura e inicia o Wi-Fi
    WiFi.mode(WIFI_STA);
    wifiMulti.addAP(ssid.c_str(), password.c_str());
    wifiMulti.run();
    if (WiFi.status() != WL_CONNECTED)
    {
        WiFi.disconnect(true);
        wifiMulti.run();
    }
}
```

Código-fonte: Setup()

```
//Ajusta os pinos de conexão I2S
audio.setPinout(I2S_BCLK, I2S_LRC, I2S_DOUT);

//Ajusta o volume de saída
audio.setVolume(20); // 0...21

//Para executar um arquivo MP3 no cartão SD, descomente esta linha
//audio.connecttoSD("nome_do_arquivo.mp3");

//Para executar uma síntese de voz, descomente esta linha
audio.connecttospeech("este é um exemplo de síntese de voz usando E.S.P. 32, I.2.S e um módulo D.A.C.", "pt");

//Para executar um streaming, descomente esta linha
//audio.connecttohost("http://mp3.ffh.de/radioffh/hqlivestream.mp3"); // 128k mp3
}
```

Código-fonte: Loop()

```
void loop()  
{  
    audio.loop(); //Executa o loop interno da biblioteca audio  
}
```

Código-fonte: Funções opcionais

```
// As seguintes funções são opcionais e retornam informações sobre a execução
```

```
void audio_info(const char *info)
```

```
{  
    Serial.print("info "); Serial.println(info);  
}
```

```
void audio_id3data(const char *info)
```

```
{ //id3 metadata  
    Serial.print("id3data "); Serial.println(info);  
}
```

```
void audio_eof_mp3(const char *info)
```

```
{ //end of file  
    Serial.print("eof_mp3 "); Serial.println(info);  
}
```

```
void audio_showstation(const char *info)
```

```
{  
    Serial.print("station "); Serial.println(info);  
}
```

```
void audio_showstreaminfo(const char *info)
```

```
{  
    Serial.print("streaminfo "); Serial.println(info);  
}
```

Código-fonte: Funções opcionais

```
void audio_showstreamtitle(const char *info)
{
    Serial.print("streamtitle "); Serial.println(info);
}

void audio_bitrate(const char *info)
{
    Serial.print("bitrate "); Serial.println(info);
}

void audio_commercial(const char *info)
{ //duração
    Serial.print("commercial "); Serial.println(info);
}

void audio_icyurl(const char *info)
{ //homepage
    Serial.print("icyurl "); Serial.println(info);
}

void audio_lasthost(const char *info)
{ //stream URL played
    Serial.print("lasthost "); Serial.println(info);
}

void audio_eof_speech(const char *info)
{
    Serial.print("eof_speech "); Serial.println(info);
}
```


Captura: Alguns dados fornecidos pelas funções opcionais

```
COM11

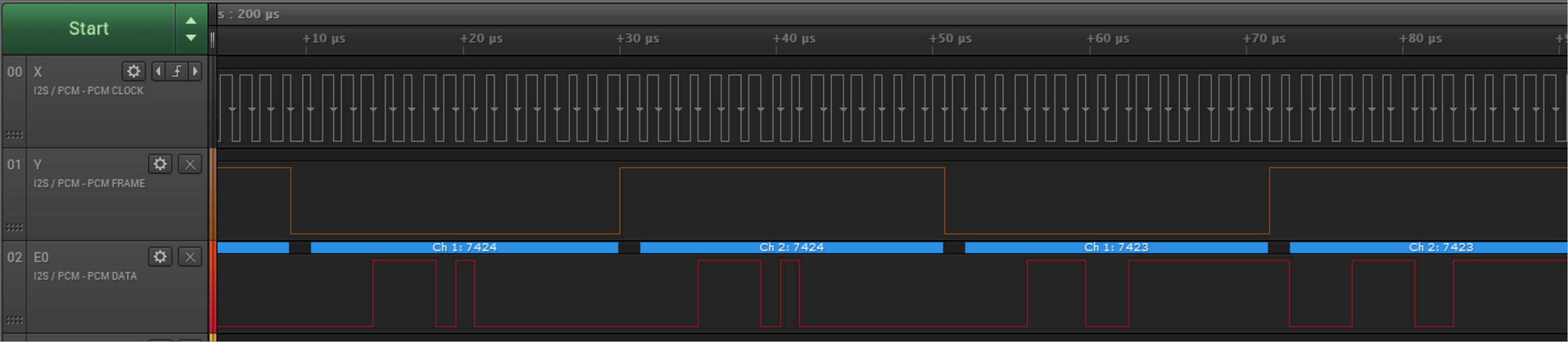
ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
flash read err, 1000
ets_main.c 371
ets Jun  8 2016 00:22:57

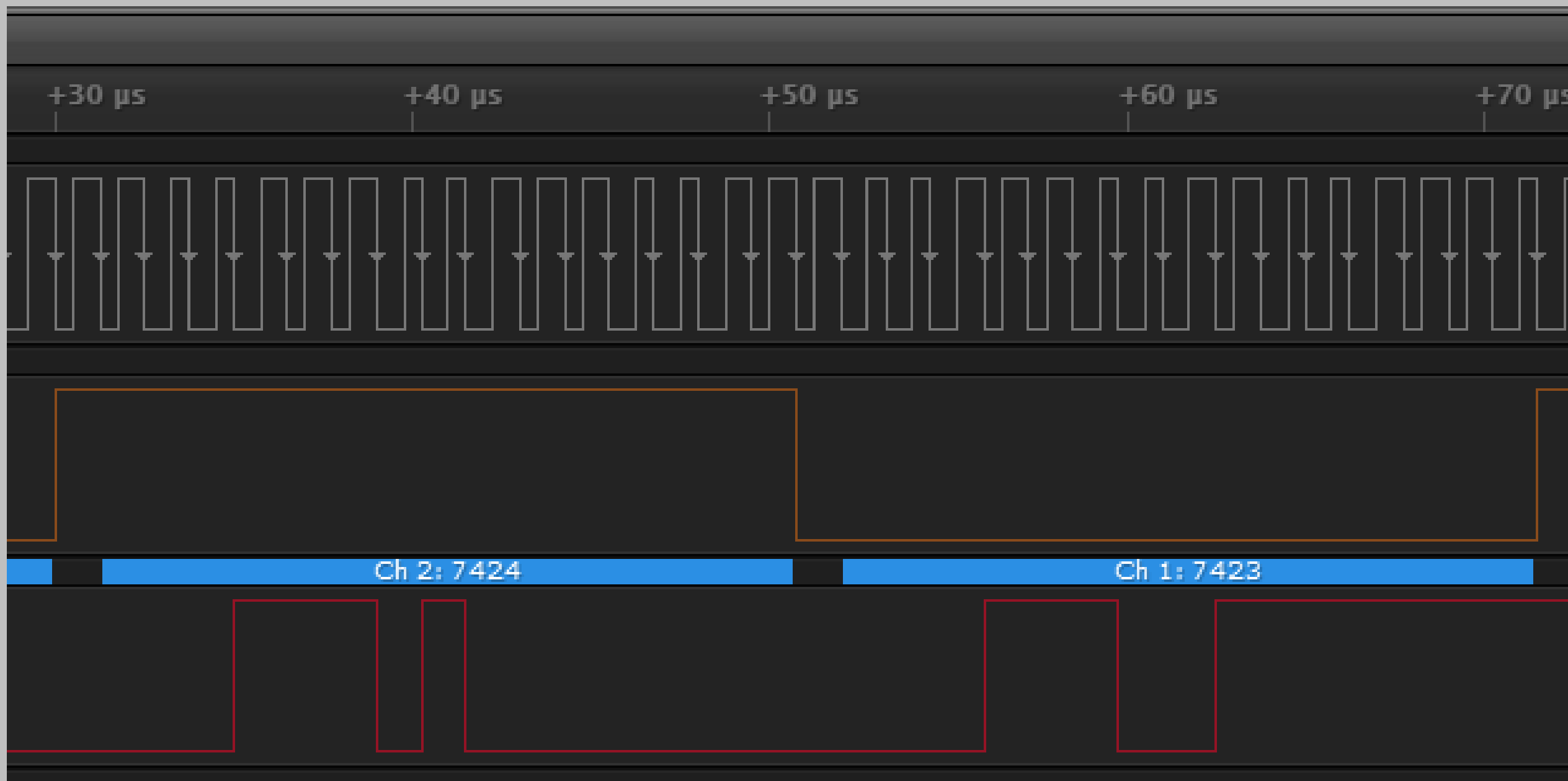
rst:0x10 (RTCWDT_RTC_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1216
ho 0 tail 12 room 4
load:0x40078000,len:9720
ho 0 tail 12 room 4
load:0x40080400,len:6352
entry 0x400806b8
info      Reading file: teste.mp3
info      ID3 version=4
info      ID3 framesSize=28
info      ID3 normal frames
info      stream ready
info      syncword found at pos 0
info      Channels=2
info      SampleRate=44100
info      BitsPerSample=16
info      Bitrate=320000
info      End of mp3file teste.mp3
eof_mp3   teste.mp3
```

```
info      Reading file: teste.mp3
info      ID3 version=4
info      ID3 framesSize=28
info      ID3 normal frames
info      stream ready
info      syncword found at pos 0
info      Channels=2
info      SampleRate=44100
info      BitsPerSample=16
info      Bitrate=320000
info      End of mp3file teste.mp3
eof_mp3   teste.mp3
```

Captura: Uma amostra das linhas I2S



Captura: Uma amostra das linhas I2S



Em www.fernandok.com

Download arquivo PDF e INO

