

Laboratoire 12 : ESP32 Audio

12.1 Bus I2S

I²S – I2S – IIS (Inter-IC Sound), prononcé (*eye-squared-ess*), est une norme d'interface de bus série électrique utilisée pour connecter ensemble des appareils audio numériques. Il est utilisé pour communiquer des données audio PCM entre des circuits intégrés dans un appareil électronique. Le bus I²S sépare les signaux de données **d'horloge** et de **série**, résultant en des récepteurs plus simples que ceux requis pour les systèmes de communication asynchrones qui doivent récupérer l'horloge du flux de données.

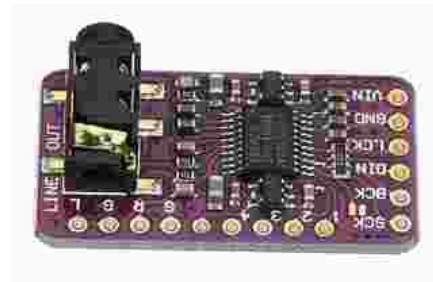
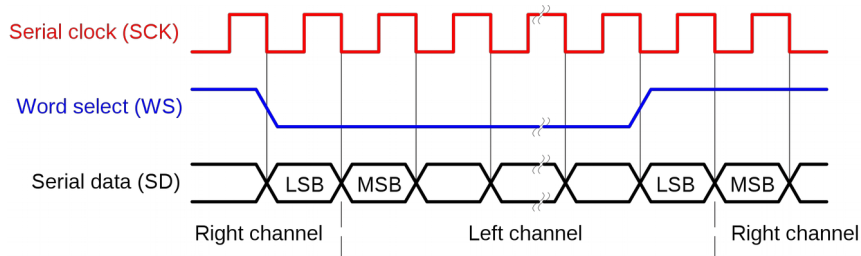


Figure Bus I2S et le module **PCM5102** (GY-PCM5102)

12.2 Un simple player mp3/aac

Le circuit PCM5102 permet de convertir le flux de valeurs PCM (16-bits) en signal analogique de l'audio. Le décodage du flux initial en **mp3** ou en **aac** est effectué par un processeur du SoC ESP32 grâce à la disponibilité de la bibliothèque **Audio.h**. A noter que le module GY-PCM5102 n'a pas d'amplificateur de puissance. Il est seulement suffisant pour les écouteurs.

Après la connexion par WiFi le flux demandé est analysé et décodé par les fonctions disponibles dans la bibliothèque **Audio.h**. Notez que l'utilisation de la bibliothèque **WiFiMulti.h** permet de rechercher un point d'accès parmi de multiples couples ssid-password.

```
#include "Arduino.h"
#include "WiFiMulti.h"
#include "Audio.h"
// Digit
#define I2S_DOUT      32
#define I2S_BCLK      14
#define I2S_LRC       2
#include <U8x8lib.h> // bibliothèque à charger a partir de
U8X8_SSD1306_128X64_NONAME_SW_I2C u8x8(15, 4, 16);
Audio audio;
WiFiMulti wifiMulti;
String ssid1 = "Livebox-08B0";
String password1 = "...";
String ssid2 = "PhoneAP"; // your network SSID (name)
String password2 = "smartcomputerlab"; // your network passw

void setup() {
  Serial.begin(9600);
  WiFi.mode(WIFI_STA);
```

```

wifiMulti.addAP(ssid1.c_str(), password1.c_str());
wifiMulti.addAP(ssid2.c_str(), password2.c_str());
wifiMulti.run();
if(WiFi.status() != WL_CONNECTED){
    WiFi.disconnect(true);
    wifiMulti.run();
}
Serial.println("WiFi connected");
u8x8.begin(); // initialize OLED
u8x8.setFont(u8x8_font_chroma48medium8_r);
u8x8.clear();
u8x8.drawString(0,0,"WiFi connected");
audio.setPinout(I2S_BCLK, I2S_LRC, I2S_DOUT);
audio.setVolume(18); // 0...21

//audio.connecttoSD("/320k_test.mp3");
audio.connecttohost("http://direct.franceinter.fr/live/franceinter-
midfi.mp3");
}

void loop()
{
    audio.loop();
}

// optional
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);
    u8x8.drawString(0,2,"Webradio");
    u8x8.drawString(0,3,info);
}
void audio_showstreaminfo(const char *info){
    Serial.print("streaminfo  "); Serial.println(info);
}
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){ //duration in sec
    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);
}

```

12.3 Simple ESP32 Internet Radio (BBC) with OLED Display

Le programme suivant complète notre première version du WebRadio par l'introduction d'une section d'initialisation permettant de choisir une parmi des stations prédéfinies dans le code. Cette section affiche les numéro de stations à sélectionner ; sans cette sélection on propose par défaut BBC2.

```
#include "Arduino.h"
#include "WiFiMulti.h"
#include "Audio.h"

// Digit
#define I2S_DOUT      32
#define I2S_BCLK      14
#define I2S_LRC       2

#include <U8x8lib.h> // bibliothèque à charger a partir de
// the OLED used
U8X8_SSD1306_128X64_NONAME_SW_I2C u8x8(15, 4, 16);

Audio audio;
WiFiMulti wifiMulti;

String ssid1 = "Livebox-08B0";
String password1 = "..";
String ssid2 = "PhoneAP"; // your network SSID (name)
String password2 = "smartcomputerlab"; // your network passw

int setparam()
{
  const int buttonPin = 0;
  int count=0, max_count=10;;
  int buttonState;
  char dbuf[32];
  u8x8.clear();
  u8x8.drawString(0, 0,"Set Station");
  while(1)
  {
    buttonState = digitalRead(buttonPin);
    if(buttonState)
    {
      if(!count)count=1;else count=0;
      Serial.println(count);
      if(count) u8x8.drawString(4, 2,"YES");
      else u8x8.drawString(4, 2,"NO ");
      max_count--; if(!max_count) return 0;
    }
    else
    {
      Serial.println(count);
      if(count)
      {
        u8x8.drawString(0, 2,"BBC1");
        u8x8.drawString(0, 3,"BBC2");
        u8x8.drawString(0, 4,"BBC3");
        u8x8.drawString(0, 5,"BBC4");
        u8x8.drawString(0, 6,"BBC5_live");
        u8x8.drawString(0, 7,"BBC6_music");
      }
      else
      {
        u8x8.drawString(0, 2,"Setting Default");
        u8x8.drawString(0, 3,"BBC2");
      }
    }
  }
}
```

```

        delay(3000);
        return count;
    }
    delay(3000);
}

int setstation()
{
    const int buttonPin = 0;
    int sf=1;
    int buttonState;
    u8x8.clear();
    char dbuf[32];
    u8x8.drawString(0, 0, "Setting Station");
    while(1)
    {
        buttonState = digitalRead(buttonPin);
        if(buttonState) { sf++; if(sf==7) sf=1;
                        Serial.println(sf);
                        sprintf(dbuf, "Station=%2.2d", sf);
                        u8x8.drawString(0, 2, dbuf);
                        }

        else
        {
            Serial.println(sf);
            u8x8.drawString(0, 4, "Station");
            sprintf(dbuf, "BBC%2.2d", sf);
            u8x8.drawString(0, 5, dbuf);
            return sf;
        }
        delay(3000);
    }
}

void setup() {
    int ret=0, sta=0;
    Serial.begin(9600);
    WiFi.mode(WIFI_STA);
    wifiMulti.addAP(ssid1.c_str(), password1.c_str());
    wifiMulti.addAP(ssid2.c_str(), password2.c_str());
    wifiMulti.run();
    if(WiFi.status() != WL_CONNECTED){
        WiFi.disconnect(true);
        wifiMulti.run();
    }
    Serial.println("WiFi connected");
    u8x8.begin(); // initialize OLED
    u8x8.setFont(u8x8_font_chroma48medium8_r);
    u8x8.clear();
    u8x8.drawString(0,0, "WiFi connected");
    audio.setPinout(I2S_BCLK, I2S_LRC, I2S_DOUT);
    audio.setVolume(18); // 0...21
    ret=setparam();
    if(ret)
    {
        sta=setstation();
        if(sta==1)
        audio.connecttohost("http://bbcmedia.ic.llnwd.net/stream/bbcmedia_radio1_mf_p");
        if(sta==2)
        audio.connecttohost("http://bbcmedia.ic.llnwd.net/stream/bbcmedia_radio2_mf_p");
        if(sta==3)
        audio.connecttohost("http://bbcmedia.ic.llnwd.net/stream/bbcmedia_radio3_mf_p");
    }
}

```

```

        if(sta==4)
audio.connecttohost("http://bbcmedia.ic.llnwd.net/stream/bbcmedia_radio4fm_mf_p"
);
        if(sta==5)
audio.connecttohost("http://bbcmedia.ic.llnwd.net/stream/bbcmedia_radio5live_mf_
p");
        if(sta==6)
audio.connecttohost("http://bbcmedia.ic.llnwd.net/stream/bbcmedia_6music_mf_p");
        }
        else
audio.connecttohost("http://bbcmedia.ic.llnwd.net/stream/bbcmedia_radio2_mf_p");

}

void loop()
{
    audio.loop();
}

// optional
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);
    u8x8.drawString(0,2,"Webradio");
    u8x8.drawString(0,3,info);
}
void audio_showstreaminfo(const char *info){
    Serial.print("streaminfo  "); Serial.println(info);
}
void audio_showstreamtitle(const char *info){
    Serial.print("streamtitle "); Serial.println(info);
}
void audio_bitrate(const char *info){
    Serial.print("bitrate      "); Serial.println(info);
    u8x8.drawString(0,4,"bitrate");
    u8x8.drawString(8,4,info);
}
void audio_commercial(const char *info){ //duration in sec
    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);
}
}

```

A faire

Développer une application WebRadio piloté par un lien Bluetooth à partir de votre smartphone. Cette application permettra de choisir les stations WEB en proposant les URLs associés au WebRadio à écouter.