1. 音頻設備:
   1. I2S總線:是專門處理音頻設備訊號的通訊協議，通常由3根線組成:

* SCK: Serial Clock,用於同步數據傳輸，所有I2S設備都需要根據這個時鐘信號來同步數據的發送和接收。
* WS: Word Select,左右聲道時鐘，指示當前傳輸的數據屬於左聲道還是右聲道。
* SD: Serial Data,實際傳輸音頻數據的信號線。
  1. SAI接口: 靈活性高、配置多樣，可支持多種音頻協議。該接口適用於許多立體聲或單聲道應用。簡單來說，I2S包含在SAI中。
  2. ALSA（Advanced Linux Sound Architecture）是Linux作業系統的一個音頻框架，負責管理和處理聲音設備的驅動程式及音頻數據。它提供了一套標準化的API，供應用程序和音頻硬件之間進行交互。以下是一些關於ALSA的特性：
* **音頻驅動**：ALSA包含了多種聲卡驅動，使不同品牌和型號的聲卡能夠在Linux系統上正常工作。
* **API接口**：提供了一套用於音頻應用程序開發的API，使開發者可以方便地控制和管理音頻設備，進行錄音、播放、音量控制等操作。
* **音頻處理**：支持多種音頻處理功能，包括音量控制、混音、音頻格式轉換等。
* **用戶空間工具**：ALSA還提供了一些用戶空間工具，如aplay（播放音頻文件）、arecord（錄製音頻）、alsamixer（音量控制界面）等，用於測試和控制音頻設備。

1. 音頻設備驅動實作:

首先，我們只需要修改Device Tree，接著在buildroot中啟用SAI和對應芯片即可，芯片驅動STM已經做好了是透過ALSA驅動框架完成的，此外alsa-lib已經在buildroot中包含了，只須過圖形化介面enable當中alsa-utils工具即可使用。(步驟如下表一)

(表一:音頻驅動取用流程)

|  |  |
| --- | --- |
| **步驟** | **說明** |
| 修改 Device Tree | 首先，我們只需要修改 Device Tree。 |
| 啟用 SAI 和對應芯片 | 在 Buildroot 中啟用 SAI 和對應芯片。 |
| 使用 STM 芯片驅動 | 芯片驅動 STM 已經完成，透過 ALSA 驅動框架實現。 |
| 包含 alsa-lib | alsa-lib 已經在 Buildroot 中包含。 |
| 啟用 alsa-utils 工具 | 只需要在圖形化界面中啟用 alsa-utils 工具即可使用。 |