

# Digitální záznamové zařízení

Cílem projektu bylo sestavit záznamník za použití SoC ESP32. Zařízení dále používá mikrofon SPH0645 a zesilovač MAX98357A, které komunikují po sběrnici I2S. Dále je připojena SD karta, na které jsou uchovány záznamy ve formátu WAV. Uživatel se zařízením interaguje pomocí tří tlačítek a OLED displaye SSD1306.

Demo: <https://youtu.be/XIY9lgNC9gY>

WAV nahrávky z videa jsou přiloženy ve složce recordings.

## Zapojení

Vzhledem k tomu, že ESP32 umožňuje připojení téměř jakékoli periferie k jakémukoli pinu, je toto zapojení poměrně arbitrární. V tomto projektu byla použit vývojový kit Wemos D1 R32. Všechny periferie byly napájeny z 3.3V vycházející z této desky, deska samotná byla napájena přes USB.

### SPH0645

Mikrofon komunikuje po sběrnici I2S. Jeho piny jsou zapojeny následovně:

SPH0645	ESP32
BCLK	17
DOUT	16
LRCL	25
SEL	3.3V

### MAX98357A

Zesilovač komunikuje po sběrnici I2S. Teoreticky by zřejmě mohla být použita stejná sběrnice, na které komunikuje mikrofon. Pro tento případ je ale použita druhá, protože ESP32 může obsluhovat až dvě najednou. Také lze individuálně zapínat a vypínat sběrnice pro tyto periferie. Jeho piny jsou zapojeny následovně:

MAX98357A	ESP32
BCLK	2
DIN	4
LRC	27

### SD karta

SD karta komunikuje přes rozhraní SPI. K mikrokontroléru je připojena adaptérem bez LDO regulátoru a bez level shifteru.

SD adaptér	ESP32
MISO	19
MOSI	23
CLK	18
CS	5

### SSD1306

OLED display komunikuje přes rozhraní I2C. Nebylo potřeba použití pullup rezistorů, zřejmě byly zapnuté softwarově v ESP32.

SSD1306	ESP32
SDA	21
SCL	22

## Tlačítka a LED

K mikrokontroléru jsou připojena 3 tlačítka (UP, DOWN, OK) na piny 13, 12 a 14. Všechna tlačítka jsou typu NO (normally open) a z druhé strany připojena k 3.3V.

LED není pro projekt nutná, její jediný účel je signalizace nahrávání. Anoda LED je připojena k pinu 26 přes rezistor.

## Způsob vývoje

Prvním krokem byla volba a instalace vývojového prostředí. Pro tento projekt bylo zvoleno ESP-IDF, které je vyvíjené a podporované výrobcem. ESP-IDF je open source pod licencí Apache 2.0.

Dále bylo potřeba oživit všechny periferie. Toto bylo nejjednodušší nahráním příkladu z ESP-IDF.

Poté co byla ověřena funkčnost a kompatibilita všech periférií zvlášť, bylo možné kód integrovat do jednoho projektu. Nejprve bylo zprovozněno nahrávání a převod na požadovaný formát. Pak následovalo přehrávání z tohoto formátu. Dále byla integrována SD karta a implementováno ukládání WAV souborů. Následně byly tyto funkce integrovány do subsystému, který operuje zcela nezávisle od uživatelského rozhraní. Nakonec bylo implementováno uživatelské rozhraní, které obsluhuje tlačítka a vypisuje na display.

## Implementační detaily

Implementace je rozdělena na dva hlavní subsystémy, které operují prakticky nezávisle. První subsystém je uživatelské rozhraní, které zpracovává vstupy z tlačítek a vypisuje na display. Ten druhý zase nahrává a přehrává soubory z SD karty.

SD karta je jediný zdroj, který je mezi těmito systémy sdílený. Nezávislost subsystémů je zajištěna pomocí úloh (tasks) ve FreeRTOS. V rámci jednotlivých subsystémů jsou použity synchronizační mechanismy, které také poskytuje FreeRTOS (semafore, fronty).

### Subsystém pro nahrávání

Tento celý subsystém je implementován v modulu `recplaymgr.c`. Jeho rozhraní je velmi jednoduché, poskytuje tyto funkce:

```
void recPlayMgrInit();
void startRec(char *filename);
void stopRec();
void startPlay(char *filename);
void stopPlay();
```

Je důležité, že žádná z těchto funkcí není blokující. Díky tomu může uživatelské rozhraní např. spustit nahrávání a potom provádět svůj kód. V rámci subsystému běží dvě úlohy - jedna pro nahrávání, jedna pro přehrávání. Při zpětném pohledu by možná stačila jedna, ale implementace je plně funkční a není důvod ji předělávat.

Všechny veřejné funkce modulu dělají jedinou věc - nastaví příslušné proměnné a semafore, které jsou později přečteny samotnou úlohou záznamníku, resp. přehrávače.

Úlohy záznamníku a přehrávače přímo přistupují k SD kartě, kde zapisují resp. čtou WAV soubory. Výběr tohoto formátu byl také důležité implementační rozhodnutí. Nejjednodušší představitelný přístup by byl ukládat vzorky přímo do souboru v binární podobě. Pro účely tohoto projektu by to bylo dostatečné, nevýhodou však je následné přehrávání na počítači. Formát WAV řeší tyto potíže, protože je standardizovaný a lze přehrát na jakémkoli počítači. Navíc v implementaci nepřidává skoro žádnou komplexitu. Stačí zapsat hlavičku o velikosti 44 bajtů a potom následují data ve stejné podobě, jako v hypotetické nejjednodušší implementaci. Také by se nabízel formát s kompresí jako MP3, to je ale daleko mimo rozsah tohoto projektu.

Z tohoto záznamníku vychází soubory WAV, které mají tyto vlastnosti:

- Vzorkovací frekvence 44100 Hz
- Rozlišení 16 bitů
- Jeden kanál

Tyto vlastnosti byly určeny čistě podle mikrofonu, který má přesnost 18 bitů a vzorkovací frekvenci mezi 32 a 64 kHz (datasheet). Validita výstupních WAV souborů byla ověřena přehráním na počítači aplikací VLC.

Nahrávání není zcela triviální, protože hodnoty z tohoto mikrofonu nejsou zarovnány na nulu - mají nějaký offset. Tento offset se také v řádu sekund po spuštění mikrofonu rychle mění. Implementace řeší druhý problém tak, že mikrofon běží permanentně. Když je offset konstantní, stačí před každým záznamem vzít pár vzorků a zprůměrovat je. Tento offset je dále odečten od všech vzorků nahrávky. Zde by také byla možnost na vylepšení a to použitím low pass filtru (v softwaru), který by odstranil tento offset i v delších nahrávkách.

Na druhou stranu přehrávání je zcela triviální. Současná implementace přeskočí header (44 bajtů) a potom už jen čte jednotlivé vzorky po 2 bajtech (16 bitů). Zde by šlo snadno implementovat přečtení správných hodnot z headeru a přehrání jakéhokoli WAV souboru. Pro účely tohoto projektu to ale není vůbec potřeba.

### Uživatelské rozhraní

Uživatelské rozhraní běží jako velmi jednoduchá while smyčka. Většinu času čeká na stisknutí tlačítka, což řeší funkce `waitEvent`. Po stisknutí je vykonána příslušná akce, případně zavolána funkce záznamového subsystému. Po každé akci je také obnoven obraz na displayi, a to pomocí funkce `showFiles`. Ta vypíše aktuální obsah adresáře.

Obsluha tlačítek by mohla být navržena komplexněji pomocí přerušení, bohužel to se ani po dlouhé době nepodařilo rozchodit. Přerušení na tlačítka způsobila, že přestala fungovat knihovna `i2s` na přehrávání zvuku. Nyní je funkce `waitEvent` implementována jako while smyčka, která kontroluje všechna tlačítka a potom čeká v řádu milisekund.

### Obsluha

Zařízení je velmi jednoduché a přátelské k uživateli. Po zapojení všech komponent je třeba vložit SD kartu naformátovanou jako FAT. Poté je zařízení připraveno k provozu a může být zapnuto napájením. Pokud je SD karta prázdná, při spuštění se na ní vytvoří složka `rec`. Do této složky se budou ukládat všechny nahrávky.

Záznamník se obsluhuje pomocí tří tlačítek - UP, DOWN a OK. Po spuštění se ukáže obrazovka se záznamy. Na začátku je vždy RECORD, následovaný již nahranými WAV soubory. Mezi nimi se lze pohybovat pomocí tlačítek UP a DOWN.

Stisknutím tlačítka OK na RECORD je zahájeno nahrávání. Rozsvítí se LED, která indikuje nahrávání. Nahrávání trvá, dokud není zastaveno stisknutím jakéhokoli tlačítka.

Stisknutím tlačítka OK na záznamu se daný záznam přehraje. Dlouhým podržením tlačítka OK se záznam smaže.

Při nahrávání není vhodné odebírat SD kartu, velice pravděpodobně to poškodí formátování.

### Použité zdroje

- ESP-IDF Examples
- SPH0645 datasheet
- MAX98357A datasheet
- WAV format