《嵌入式系统》大作业实验报告：Part 1

陈俊哲　2020010964
梁　烨　2020080093
田正祺　2020080095

1　实验内容

实验背景：本次实验的硬件系统采用 MPU 和 MCU 双平台设计，其中 MPU 用于 Linux 操作系统开发，MCU 用于 ARM 体系结构与裸机编程。本次实验会在 Linux 系统下进行。

实验前置知识：本次实验需要使用 C 语言来编写文件读取函数，并利用虚拟机交叉编译代码后，在开发平台上运行。因此需要提前掌握一定的 C 语言基础。

实验目标：1、使用系统 I/O 函数读取 wav 音频文件，并将 wav 音频文件的参数输出到命令行。2、将读取音频文件的参数写入开发平台上的 txt 文件中。注：文件操作恤经过交叉编译并在开发平台上运行

2　实验部署

本次实验的实验设备是嵌入式开发板以及 Windows10 电脑。开发板上要把 STM32MP157 芯片对应的启动方式拨码成 EMMC 启动模式，并插好电源开机。在 Windows10 电脑中，实验使用 Visual Studio Code 进行代码编写，使用的语言为 C。实验没有使用说明中的 VMware 虚拟机，而是使用 WSL 来进行交叉编译。电脑主机和开发板在实验过程中通过设置以太网 ip 地址和子网进行联通。

3　实验过程

具体实验中的操作步骤和重要的代码（注意，不要把所有代码全贴上来，只对你觉得最关键的代码环节，或者自己有创新型的优化进行说明，也可以放伪代码）；

操作步骤中可能出现的问题以及自己的解决方式（说的越详细越好，能够体现出你对实验，理论和系统底层的深入理解）。

4　实验结果

对测试集的描述，及用图片或者图表对测试结果进行描述。

有没有例外（不符合输出要求，或者达不到输出要求的样例），并分析为什么会出现这样的结果。

分析实验在哪些方面还有改进的空间，如何提升效果，优化代码（比如在嵌入式编程环境中，我们需要尽可能的优化代码执行文件的大小和执行的速度，以及代码是否包含安全漏洞和可能存在的内存泄露等等）。

```
root@myir:~# ./audioplayer-arm
Usage: ./audioplayer <filename>
root@myir:~# ./audioplayer-arm does-not-exist
Cannot open file
root@myir:~# ./audioplayer-arm audioplayer-arm
Invalid wave file
root@myir:~# ./audioplayer-arm test.wav
RIFFChunk
```

```
    ID            RIFF
    Size          589860
    Format        WAVE
FormatChunk
    ID            fmt
    Size          16
    AudioFormat   1
    NumChannels   2
    SampleRate    44100
    ByteRate      176400
    BlockAlign    4
    BitsPerSample 16
DataChunk
    ID            data
    Size          589824
Saved header to test.wav.txt
root@myir:~# cat test.wav.txt
RIFFChunk
    ID            RIFF
    Size          589860
    Format        WAVE
FormatChunk
    ID            fmt
    Size          16
    AudioFormat   1
    NumChannels   2
    SampleRate    44100
    ByteRate      176400
    BlockAlign    4
    BitsPerSample 16
DataChunk
    ID            data
    Size          589824
```

5 实验心得

对本次实验任务的评价，比如你在本次实验中学到了什么，实验对你的编程能力有没有提升，实验的难度是否过大或者过于简单，可以向助教和老师提出相关的意见等等。

- 开发板的 DHCP 没有自动配置 IP 地址。

  手动在开发板以及主机上配置 IP。

- 在开发板上运行时出现错误：

  ```
  ./audioplayer-xc: /lib/libc.so.6: version `GLIBC_2.34' not found (required by ./audioplayer-xc)
  ```

  由于主机的 glibc 的版本大于开发板上的 glibc 版本，所以在主机上交叉编译的可执行文件在开发板上找不到需要的库。使用 gcc 的 `-static` 编译参数进行静态链接可以解决此问题，但是在预先调研如何链接 ALSA 的 `libasound` 库时遇到了静态链接产生的问题。因此决定使用动态链接并降低主机的 glibc 版本，即从 Ubuntu 22.10 切换到 Ubuntu 20.04。

- 开发板启动后，操作系统会默认开启 `mxapp2` 程序。在 Linux 系统上，通常输入 Ctrl+Alt+Fn 会切换到命令行，但是 `mxapp2` 似乎禁用了此功能。

6 编译与运行

为开发板的 ARM 处理器进行交叉编译需要 Ubuntu 的 `gcc-arm-linux-gnueabihf` 包：