

CENG4480 Embedded System Development and Applications

Computer Science and Engineering Department

The Chinese University of Hong Kong

Laboratory 6: Sound Recorder (II)

October, 2018

Introduction

In this lab session you will play with sound recorder board.

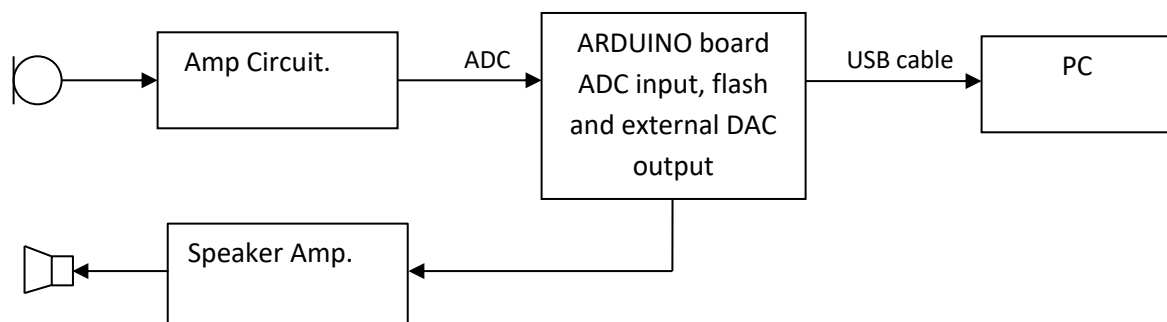
- Capture the audio signal by a microphone.
- Amplify the signal by an operation amplifier (Op-am).
- Feed the amplified signal to the Analog-to-digital input (ADC) of the ARDUINO microcontroller.
- The ARDUINO performs Analog to Digital conversion.
- The ARDUINO writes the data to flash.
- The ARDUINO reads the stored data from flash and performs Digital to Analog conversion and feed the signal to a speaker amplifier.

Objectives

- To learn how to interface an analogue signal to digital systems.
- To learn how to process audio signals in an embedded system.
- To develop an audio digitizer using an embedded system.

Procedure 1. Hardware Design

The following diagram shows the hardware system.



Procedure 2. Construction of sound recorder.

Connect the mic and DAC circuit with speaker to your PCB board (as shown in Fig. 1).
(Refer to sample board)

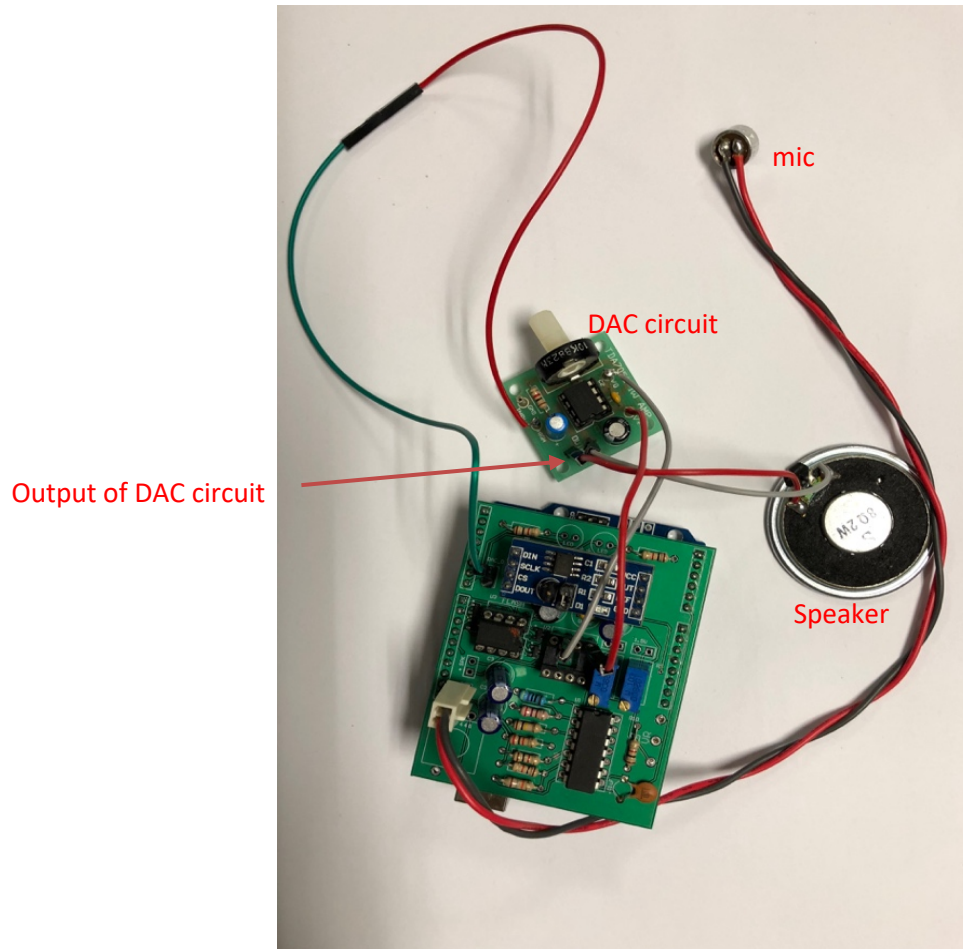


Fig. 1 Sample board

Procedure 3. Software Design

(i). The offered flash library functions:

- `void FlashStartWrite(uint32_t page);` (gives the starting address to write data in flash)
- `void FlashStoreWord(uint16_t input);` (writes 10-bit data into flash)
- `void FlashEndAction();` (tells flash that the current action “read” or “write” is finished)
- `boolean CheckForFinish();` (ensures the flash is ready for another action)
- `void FlashStartRead(uint32_t page);` (gives the starting address to read data from flash)
- `uint16_t FlashReadWord();` (reads 10-bit data)
- `void DeleteAll();` (erases the whole flash)

(ii). The operations on offered buffer structure:

- a. void addBuffer(uint16_t temp);(store the data in buffer during flash preparing next page)
- b. bool popBuffer(uint16_t* output);(pops stored data from a buffer and check whether the buffer is “empty”)

(iii). The offered driver for TI TLC5615 DAC:

- a. class TLC5615 (includes a method: analogWrite(uint16_t value));
please refer to TLC5615.h and TLC5615.cpp files.

Procedure 4. Complete the missing code in loop function so that the speaker can repeat what you say to mic. There are 6 places in loop function for you to complete. You need to demonstrate your results to TAs. Connect the input channel of oscilloscope to the output of DAC circuit, record the playback of audio signal displayed on the screen in your report.

(Hint: use the offered flash library functions and exploit buffer to avoid missing sampling data when the flash is waiting for the next page.)

END