CENG4480 Embedded System Development and Applications

Computer Science and Engineering Department

The Chinese University of Hong Kong

Laboratory 5: Audio Recorder

October, 2021

Introduction

In this lab session you will construct an audio recorder which make up of (1) Microphone amplifier, (2) Arduino, (3) SD card module and (4) Speaker amplifier.

The recording steps of the signal are as follows:

- 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 (A0) of the Arduino microcontroller.
- The Arduino performs Analog to Digital conversion and then store the data in the SD card.

The playing steps of the signal are as follows:

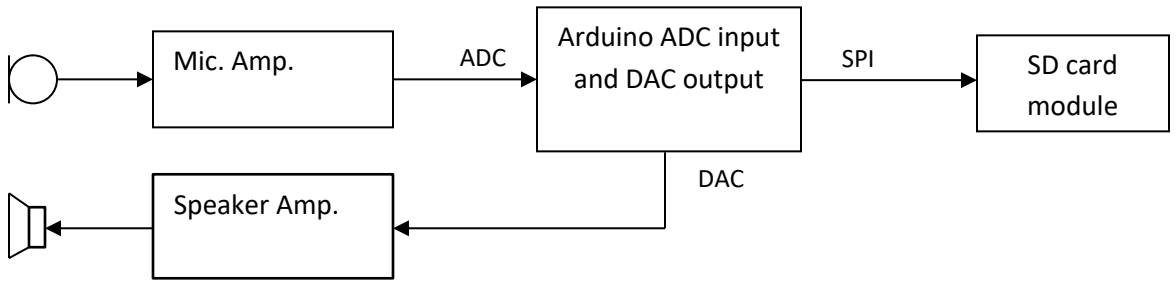
- The Arduino read data from SD card and performs Digital to Analog conversion and feed the signal to speaker amplifier.
- The speaker amplifier filter out the high frequency noise and amplify the audio signal to drive the speaker.

Objectives

- To learn how to interface an analogue signal to digital systems
- To learn how to use SD card to store audio signals in an embedded system.

Introduction to hardware part

The following diagram shows the hardware system.



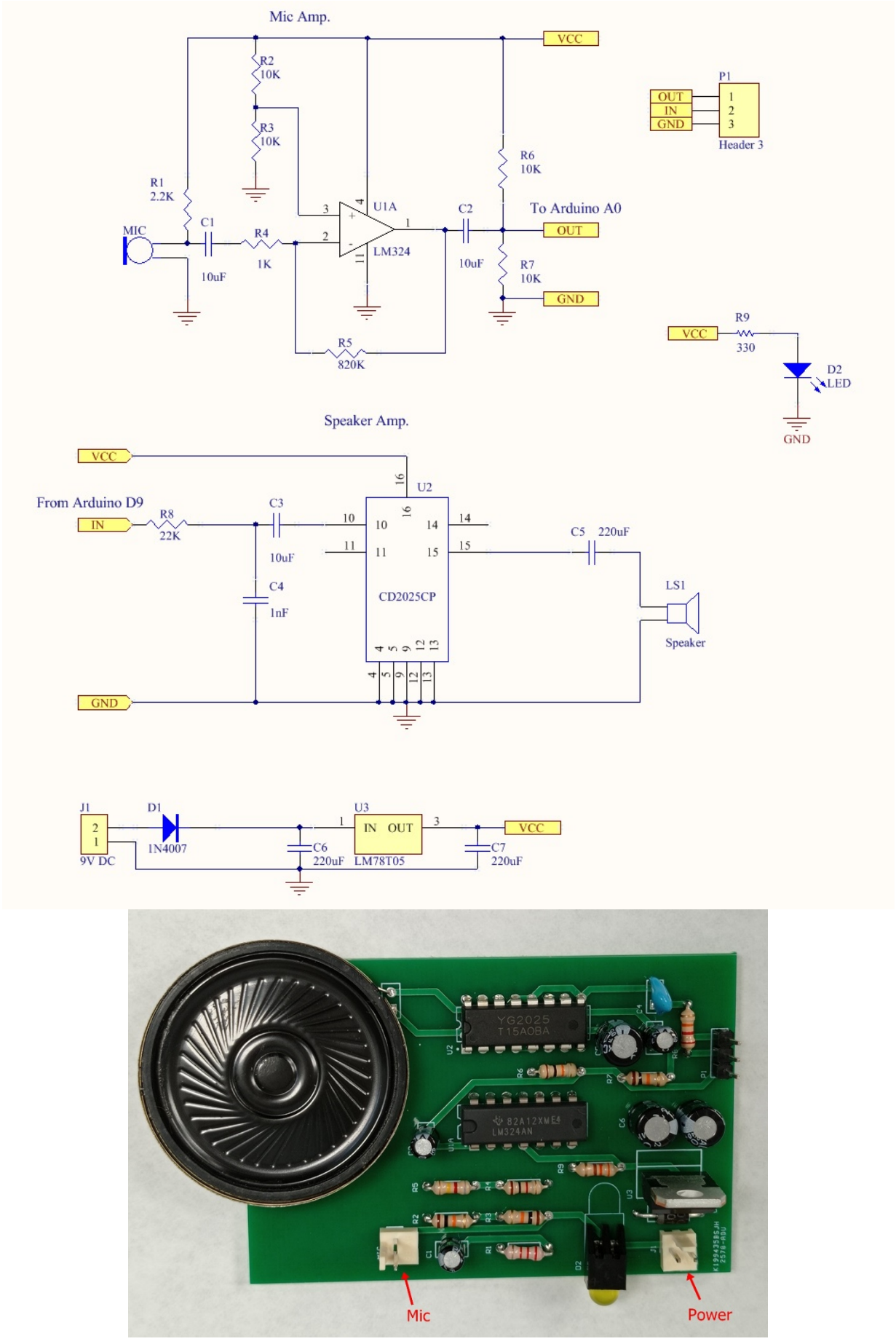


Figure 1. Mic amplifier and speaker amplifier (will provide to you as a single PCB board)

The audio signal input from the microphone is amplified by a Mic. Amp. Then the amplified signal (0 – 5V) is fed to the ADC input of the Arduino board (A0). The audio signal is then converted into digital data and saved in the SD card. In playing mode, the Arduino read data from SD card and convert to PWM analog signal and output to speaker amp. In the speaker amp a low pass filter remove the high frequency noise and amplify the audio signal to drive the speaker. The power of the board is provided by 4 x 3.2V battery pack and regulated to 5V by 7805 regulator.

Introduction to software part

The given example program Lab5.ino
The example program is using the Arduino TMRpcm library to record and play the audio signal. For detail operation you can refer to the TMRpcm library wiki :

https://github.com/TMRh20/TMRpcm/wiki

NOTICE:

(1) To use the TMRpcm library you have to install the library (TMRpcm-master.zip) on the Arduino IDE.

Download the TMRpcm-master.zip from here:

https://github.com/TMRh20/TMRpcm/archive/master.zip

(2) You have to uncomment 3 lines inside the **pcmConfig.h** file.

- #define buffSize 128,
- #define ENABLE_RECORDING and
- #define BLOCK_COUNT 10000UL

The **pcmConfig.h** is located in

C:\Users\XXX\Documents\Arduino\libraries\TMRpcm-master

(*XXX – username of your PC*)

Exercise 1. Connect the SD card module to Arduino board

Connect the SD card module to Arduino board by following the steps below:

- Connect CS (blue) to Arduino pins 4
- Connect SCK (yellow) to Arduino pins 13
- Connect MOSI (violet) to Arduino pins 11
- Connect MISO (green) to Arduino pins 12
- Connect VCC (red) to Arduino 5V
- Connect GND (black) to Arduino GND

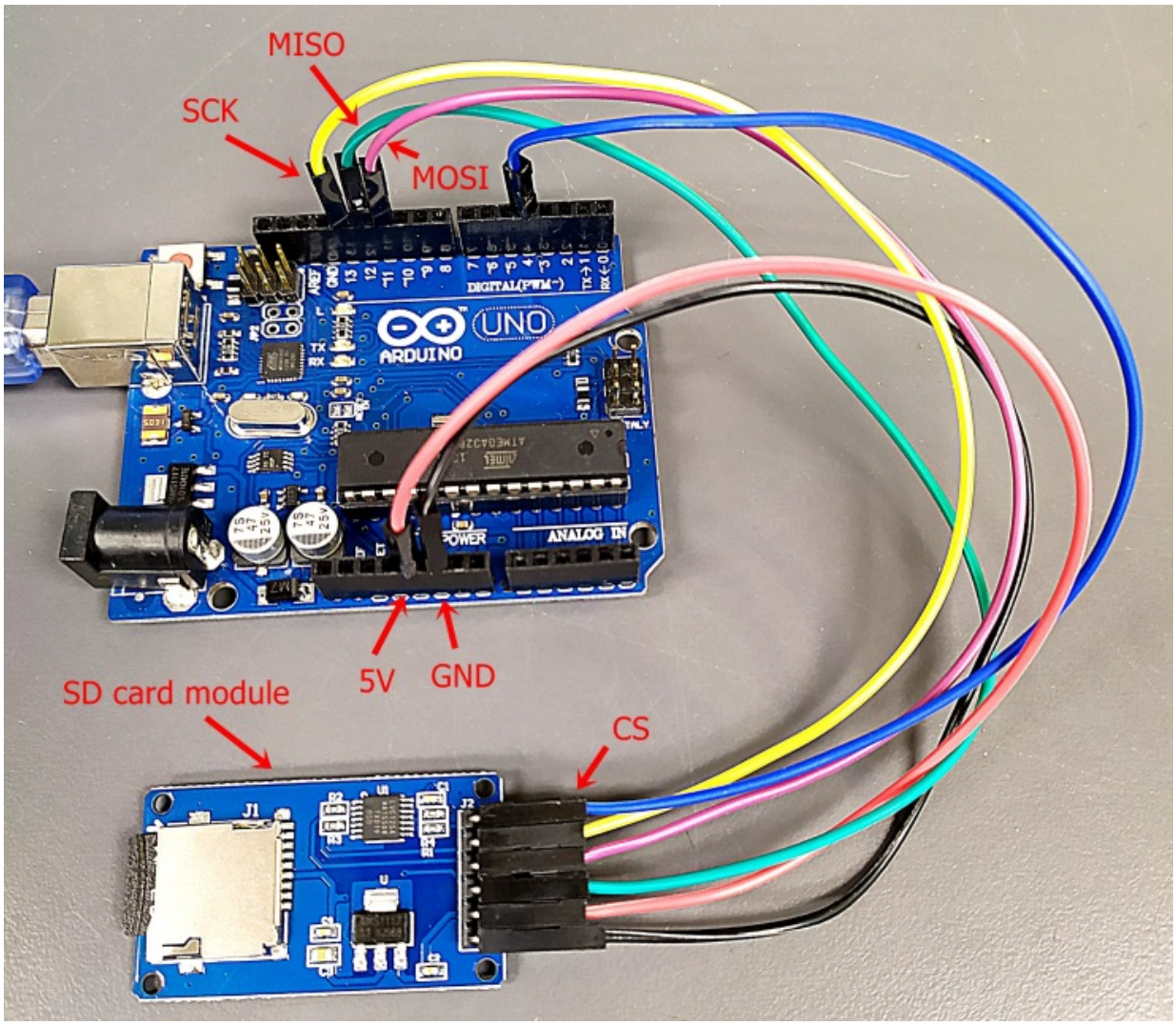


Figure 2. Connection between SD card module to Arduino

Exercise 2. Complete the audio recorder

Connect all the components to complete the audio recorder by following the steps below:

- Connect OUT P1-1 (yellow) on the Mic amp and Speaker amp PCB to Arduino A0
- Connect IN P1-2 (orange) on the Mic amp and Speaker amp PCB to Arduino pin 9
- Connect GND P1-3 (black) on the Mic amp and Speaker amp PCB to Arduino GND

- Connect condenser mic to the Mic amp and Speaker amp PCB
- Connect condenser mic to the Mic amp and Speaker amp PCB
- Connect battery pack to the Mic amp and Speaker amp PCB

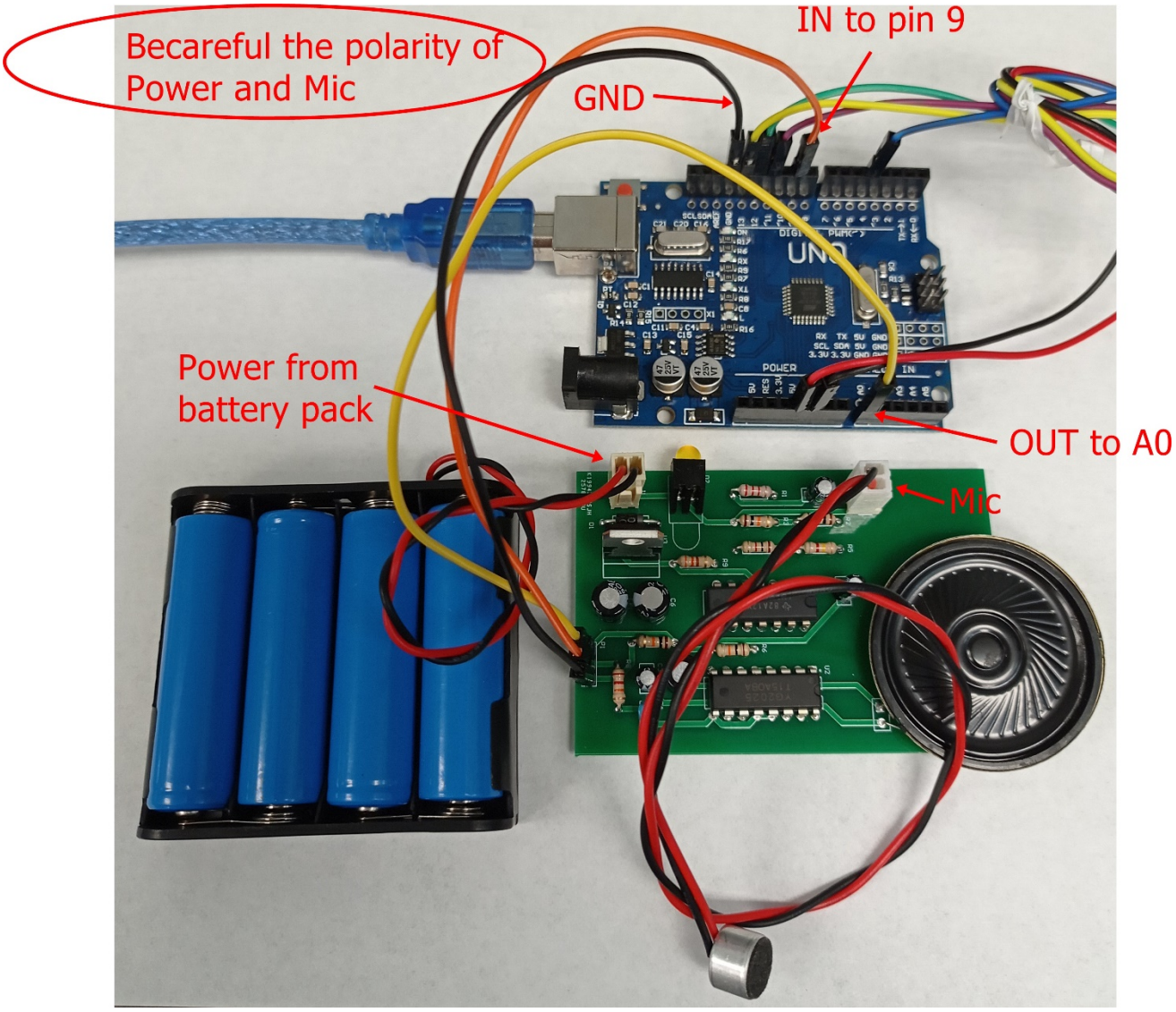


Figure 3. Connection of the rest components

Exercise 3. Programming and testing the audio recorder

1. Connect the USB cable of Arduino to PC. Download the given Lab5.ino program to Arduino.

2. Open RealTerm terminal program and configure the Baudrate(115200) and Port number. Reset the Arduino, you should see the SD OK message (as shown in Figure 5.).

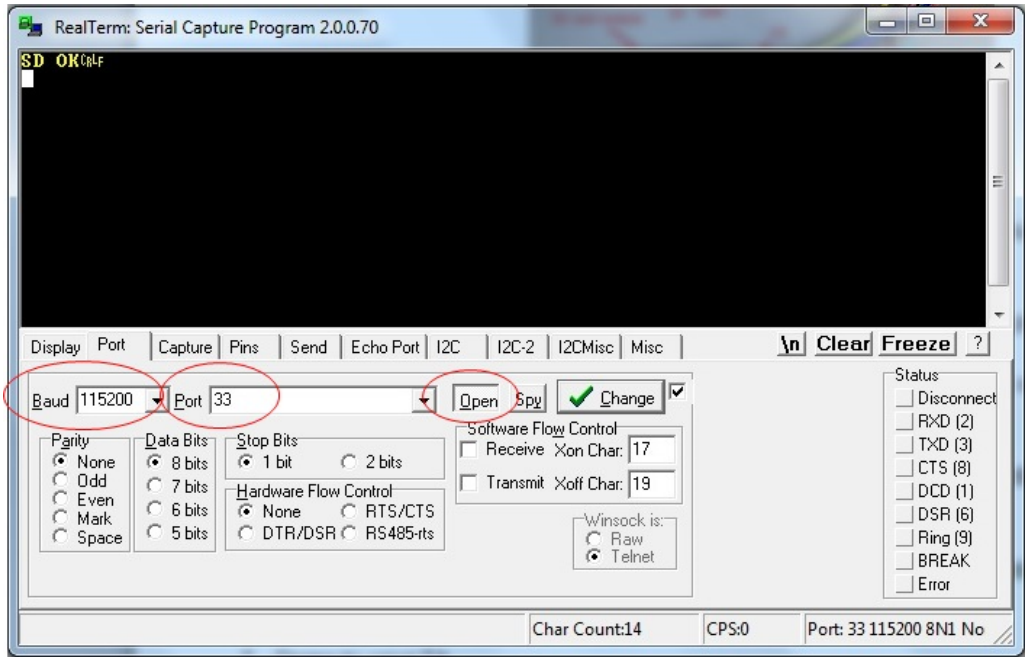


Figure 5. Configure Baud rate and Port number

3. Start to record some voice by sending “r” on RealTerm and stop recording by sending “s” on RealTerm.

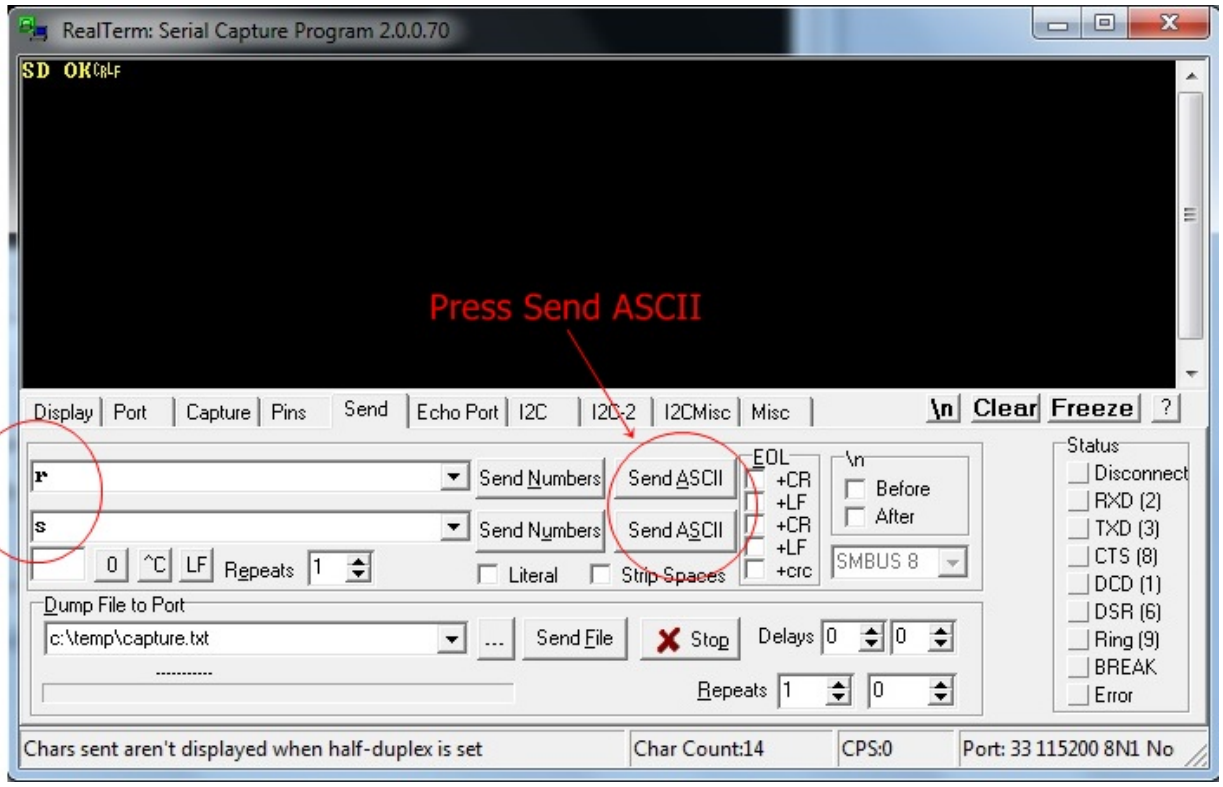
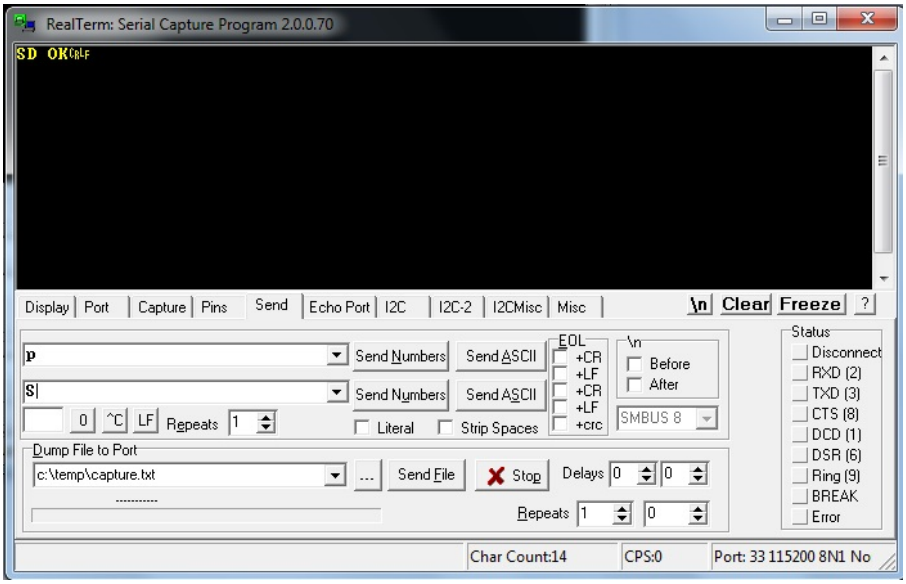


Figure 6. Send “r” to record and “”s” to stop

4. Play the recorded voice by sending “p” on RealTerm. You should heard the voice on the speaker. To stop playing send “S”.



5. Send your demo video and your report with the answer of the questions to TA before deadline.

END

Lab 5 Questions

Q1. What type is the op. amp. circuit used in the mic. amplifier ?

Q2. What is the gain of the mic. amplifier ?

Appendix

Lab5.ino

```
#include <SD.h>
#include <SPI.h>
#include <TMRpcm.h>

//#define SD_ChipSelectPin 53  //example uses hardware SS pin 53 on Mega2560
#define SD_ChipSelectPin 4  //using digital pin 4 on arduino nano 328, can use other pins

TMRpcm audio;   // create an object for use in this sketch

void setup() {
```

```
//  audio.speakerPin = 11; //5,6,11 or 46 on Mega, 9 on Uno, Nano, etc
//  pinMode(12,OUTPUT);  //Pin pairs: 9,10 Mega: 5-2,6-7,11-12,46-45

  audio.speakerPin = 9; //5,6,11 or 46 on Mega, 9 on Uno, Nano, etc
  pinMode(10,OUTPUT);  //Pin pairs: 9,10 Mega: 5-2,6-7,11-12,46-45

  Serial.begin(115200);

  if (!SD.begin(SD_ChipSelectPin)) {
    return;
  }else{
    Serial.println("SD OK");
  }
  // The audio library needs to know which CS pin to use for recording
  audio.CSPin = SD_ChipSelectPin;
}


void loop() {

    if(Serial.available()){                          //Send commands over serial to play
      switch(Serial.read()){
        case 'r': audio.startRecording("test.wav",16000,A0); break;    //Record at 16khz
sample rate on pin A0
        case 'R': audio.startRecording("test.wav",16000,A0,1); break;  //Record, but with
passthrough to speaker.
        case 't': audio.startRecording("test.wav",16000,A0,2); break;  //Do not record.
Output direct to speaker
                                                   //Note: If samples are dropped
before writing, it
                                                   //      will not be heard in
passthrough mode
        case 's': audio.stopRecording("test.wav"); break;              //Stop recording
        case 'p': audio.play("test.wav"); break;                       //Play the
recording
        case '=': audio.volume(1); break;                              //Increase volume
by 1. Does not affect recording
        case '-': audio.volume(0); break;                              //Decrease volume
by 1. Does not affect recording
        case 'S': audio.stopPlayback(); break;                         //Stop all
playback

      }
    }
}
```