EEC 172 Lab 5: DTMF Texting Over an Asynchronous Serial (UART) Link I Section: A03

Manprit Heer
912839204
mkheer@ucdavis.edu

Cathy Hsieh
912753571
cchsieh@ucdavis.edu

OBJECTIVE

This lab introduces the implementation of audio input into embedded systems. We create a simple audio interface to our system, specifically a touch-tone phone input. We design a system to decode dual-tone multi-frequency (DTMF) audio signals coming from our mobile device.

INTRODUCTION

In order the complete this lab, we first familiarize ourselves with the components and hardware interface. The LM1086CT-3.3 Low Dropout Voltage Regulator has the purpose of supplying a clean 3.3V power supply to the OLED. We will use the +5v output signal from the Launchpad as input voltage. Additionally, we will use 10 uF bypass capacitors at the input and output pins, and the Microphone and ADC will have their ground pins tied to the Launchpad ground pin. The only element that these have in common are the GROUND, otherwise the power supply varies.

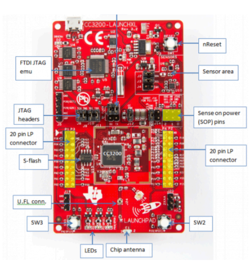
The electret microphone with the amplifier has numerous interface pins, will will be connected to the launchpad accordingly. It includes: AR (Attack/Release Ratio Select), OUT, GAIN, VDD, and GROUND. We continue to study the data sheet provided to us to figure out how the data is being read in, as well as what the gain should be in aiding the reduction of noise.

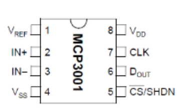
The MCP3001 ADC unit is a converter that will be interfaced to the CC3200 LaunchPad using SPI.

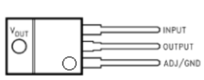
MATERIALS AND METHODS

The equipment needed for this lab are listed as follows:

1. CC3200 LaunchPad (CC3200-LAUNCHXL)
2. USB Micro-B plug to USB-A plug Cable
3. Mobile Device & App with dual-tone multi-frequency
4. LM1086CT-3.3 Low Dropout Voltage Regulator
5. MAX9814 Adafruit Electret Microphone Amplifier
6. MCP3001 A/D Converter (Microchip)







CC3200 LAUNCHPAD[1]

PROCEDURE

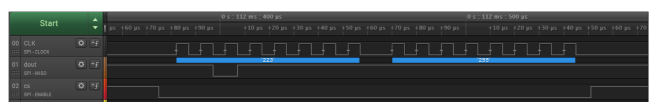
Part I. Connecting the Components

Hardware

We first connect all the proper components together, as noted in the introduction. In order to connect the ADC, we follow the bottom chart.

ADC connections	
IN +	MIC (out)
IN -	GND
Vref	5V
Vdd	5V

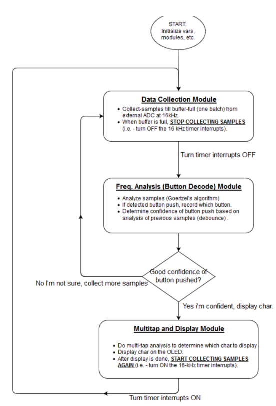
To test out the ADC connection before implementing other connections, we decide to bring out our Saleae Logic Analyzer. To test the functionality, we decide to match the voltage of 5V, excepting to get a full ON value of 1023. We receive 1023, but knew we had to use one more value to make sure it really works. We then input 3.3V which produced about 764, which was expected since our 5V value was 1023.



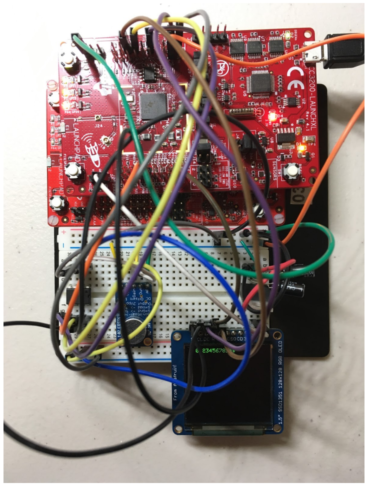
LAB V | SALEAE ANALYZER - TWO PACKETS

Software

In order to detect and decode DTMF tone output, we must use the Goertzel algorithm provided to us. First, we must parse the two packets accordingly and make sure we get the right data from this 16-bit data. To do this, we shifted the bits accordingly to get the right 10-bit information, 5 being in one packet and the other 5 being in the second one. The Goertzel algorithm is then used to transfer this number into more familiar encoding and helps us figure out what frequencies are high in the mapping, hence giving us the right number pressed.

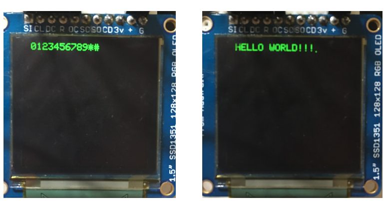


1 https://www.researchgate.net/figure/TI-CC3200-Launchpad-board_fig1_312559421



LAB V | OLED + MIC + ADC + REGULATOR SET-UP

After we decoded the numbers, we simply connect the outcome with the previous lab's OLED screen print-out component that we had previously put together, resulting in a texting interface with the OLED through the recognition of the DTMF tones. After much debugging, we were able to use the texting interface that we had implemented with the IR remote through these frequency-based tones.



LAB V | DTMF TEXTING

Issues and Troubleshooting

Our main issues had appeared throughout the software. We did not know how to properly shift the bits to get the right data, and in turn we would not get the appropriate value of 1023 when we had first tested with the 5V. To fix this, we began to experiment with the way we were storing and shifting the packets of data. After finding a right fit, we began to get a constant 1023. The next day of re-plugging in and setting up, we started getting inconsistent values again. We wanted to blame this on the board naturally, but realized that it may be noise or some wiring problem. We eventually debugged and troubleshooted our way to the way we were decoding our packets. It was indeed a shifting issue. After fixing it, our results become more constant and reliable.

SUMMARY

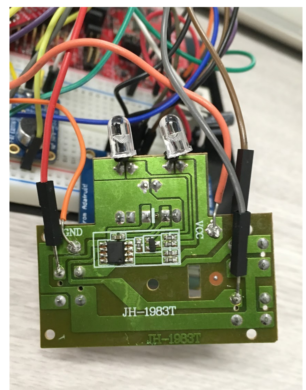
Integrating audio in embedded systems is a powerful, yet fairly easy process. This lab demonstrated by building upon the previous labs, that embedding multiple devices and systems together to create a working system is easier than it sounds. We were able to use DTMF tones, instead of the IR remote, to implement a texting interface on the OLED. Having started with a basic Blinky program, this lab truly demonstrated the greatness of the multifunctional microcontroller unit.

EXTRA CREDIT

While working on the lab, we realized we wanted to experiment with battery-powered objects and MCUs not provided by the lab to truly test our knowledge and capabilities. To do this, we bought a toy lizard from Walmart and dismantled its remote as well as the lizard itself to see how the system was set up on the inside.

LAB V | DTMF TEXTING

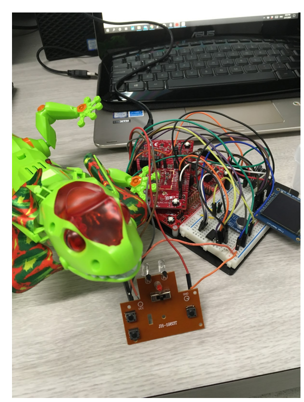
After carefully analyzing the wires and motors, we decided to connecting the DTMF tones to the lizards' various buttons.



LAB V | ORIGINAL REMOTE'S MCU

Essentially, the number 1's tone was able to move the lizard forward and 2 was able to make it move in reverse. Additionally 5 would allow it to stop and 8 would make the lizard ROAR.

We made this happen by soldering one side of the wires to the original remote control MCU's switches that controlled the robot's actions. This would provide power to the appropriate function when prompted by our code. The other side of the wires were connected to the CC3200 LaunchPad. This would allow us to access the relevant GPIO pins that we assigned and manipulate the Lows and Highs based off of the button that was pressed. At this point, like any other lab, we had to integrate two different systems together by coding the correct order and syntax for the way we wanted to implement it. At the end, we had the DTMF remote controlling the lizard's moves.



LAB V | EXTRA CREDIT

DEMO VIDEO:

https://www.youtube.com/watch?v=z5Ef5kmhsLg&feature=youtu.be