

# JamBand: a Human-Robot Interactive Band

## EE183

Wilson Chang  
University of California, Los Angeles

Contact Information:  
HSEAS  
UCLA  
7400 Boelter Hall, Los Angeles, CA 90095

Email: ckwojai@ucla.edu  
Website: https://github.com/ckwojai/EE183\_JamBand

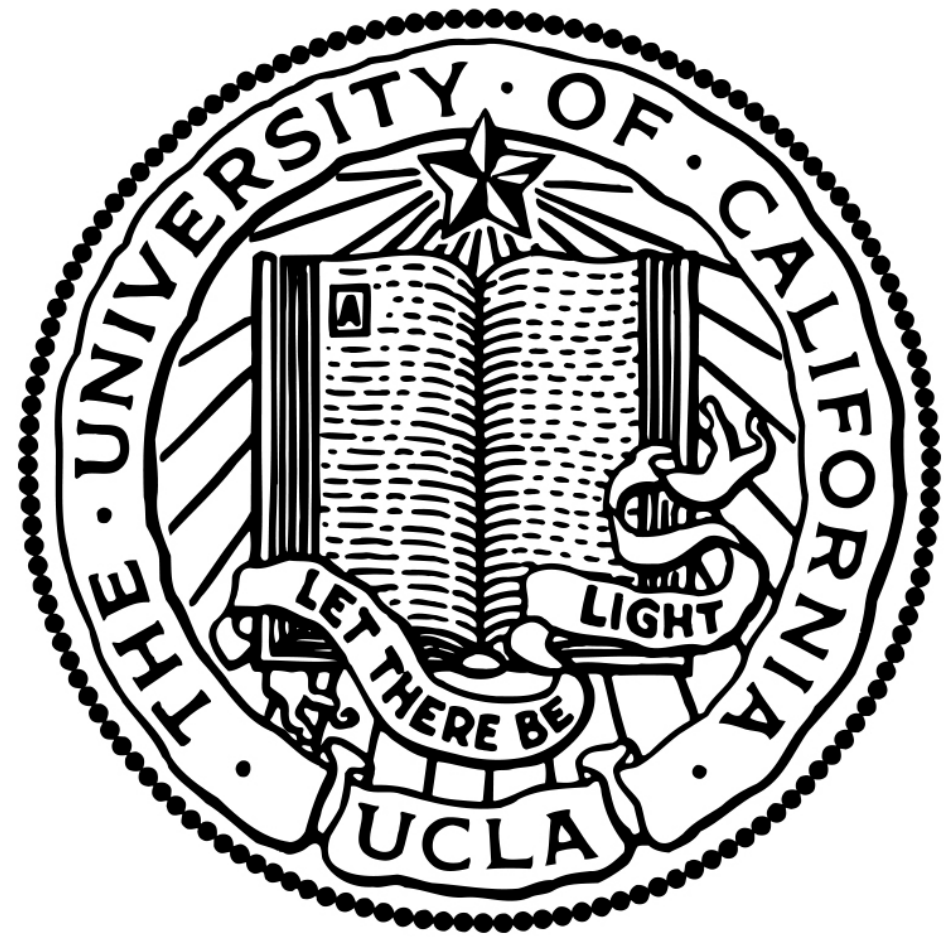


Figure 3: Our Awesome User Interface

**Abstract**

Our goal was to build a musical jam band driven by two ESP8266 micro-controllers, utilizing sensors and actuators. The main challenges in this project includes 1) composing music with Buzzer and Servo, 2) communication and synchronization between the two micro-controllers, and 3) Deploy the project online. We solved these problems using 1) note-frequency matching and delay time control, 2) Serial communication, and 3) Port-forwarding. In this poster, we will guide your through our solutions.

## Introduction

In this project, we build our JamBand with two micro-controller with Buzzer and Servo mimicking the trumpet and drumstick. We also incorporate ultrasonic sensor and touch sensor for human interaction with the band. A Web interface is also designed for use to access the autonomous mode of the band through Internet.

## Music Composition

In our project, two songs are composed, namely Twinkle Twinkle Little star and Row Row Row the boat. To code for the buzzer to sound like any music, we define a number of notes with its corresponding frequency. As shown below:

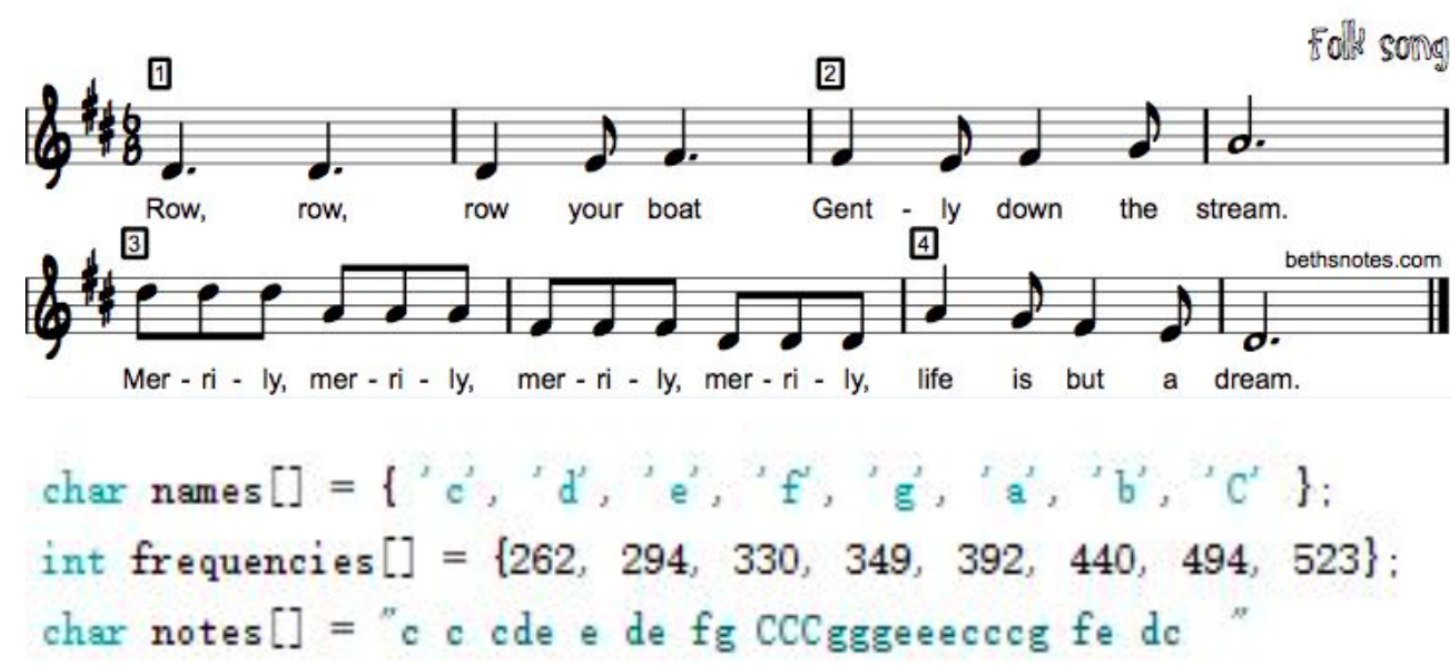


Figure 1: Figure caption

We then make sound with the buzzer using the corresponding frequency to match each note in the sheet. A duration is also added for the time length of each note. These two parameters are then pass to the Arduino function tone(frequency, duration) to let the buzzer make Sound.

As for the servo, it's better to code this together with the buzzer for synchronization purpose. We first need to determine the starting and ending angle for the servo. Fix the ending angle because that's where

the stick hit the drum every time. Then program the delay as shown below, assume the initial configuration of servo is at starting angle,

1. Set the servo to the ending angle
2. Apply some delay too allow time for the servo to go to the ending angle
3. After the above appropriate delay, this is when the stick hit the drum
4. Start playing one note of the buzzer music
5. Set the servo back to starting angle, and apply some delay for it to get there
6. Repeat the above steps

Note that there are two ways to set the duration for each stick hit: either by changing the starting angle or give it less delay time to go to starting angle (so that it's still on his way to starting angle when the next end angle command is already received). Both of these two methods are used in our program, it's also essential to parameterize these delays so that we can control the tempo of both instruments.

## Serial Communication

After composing our music, we get both instruments to play in the same speed and note. However, there have to be some communication between these two instruments for them to know when and what music and tempo to play. We accomplished this with Serial Communication between the micro-controllers.

To establish the communication channel, we wire up Tx↔Tx and Rx↔Tx for the two micro-controllers. Now, one has to act as a sender and the other receiver. In our project, we arbitrarily chose the servo to be the sender.

Sender's job is easy, through Serial, send (print) a Ready signal R, followed by a Song signal S, and then a tempo Signal T. On the receiver side, three layers of if statement are constantly checking if there are anything in the Serial buffer (if (Serial.available());0)) and read the message. Know the signals, it then set and play the song with the right tempo.

## Port Forwarding

After robots are functioning properly, we started to think of a way to get our Robots online. We followed the tutorials online, and was able to set up ESP8266 to be a WIFI access point. However, user has to connect to the designated WIFI and command through a ugly User Interface.

In order to change this, we need to somehow get our Robot really online so we can have a separate more customizable page to control our robot. After some research, Port Forwarding seems to be a feasible solution.

To set things up, we first need to set our ESP8266 now to be a Workstation rather than an Access Point. This Workstation is connected to our Home Wifi given SSID and Password. We should now be able to access the control page if we are connected to our home WIFI. Now to get page entirely online, we need to use our Router IP address (since our router is connected to the Internet).

First we go to <http://www.whatsmyip.org> to get our Router IP. We then go to our router setting and look for Port Forwarding and fill in the according IP address and port (we define this in our Arduino code). As shown below,

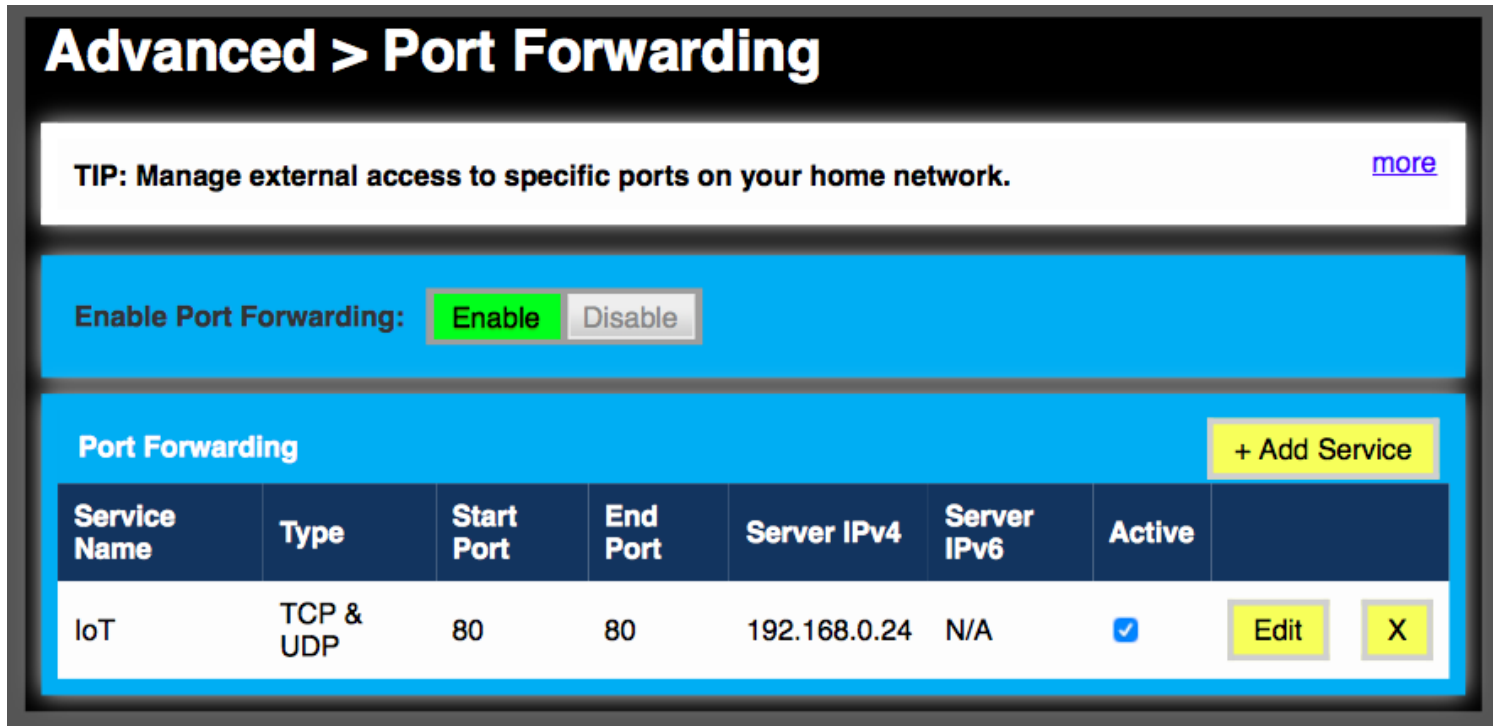
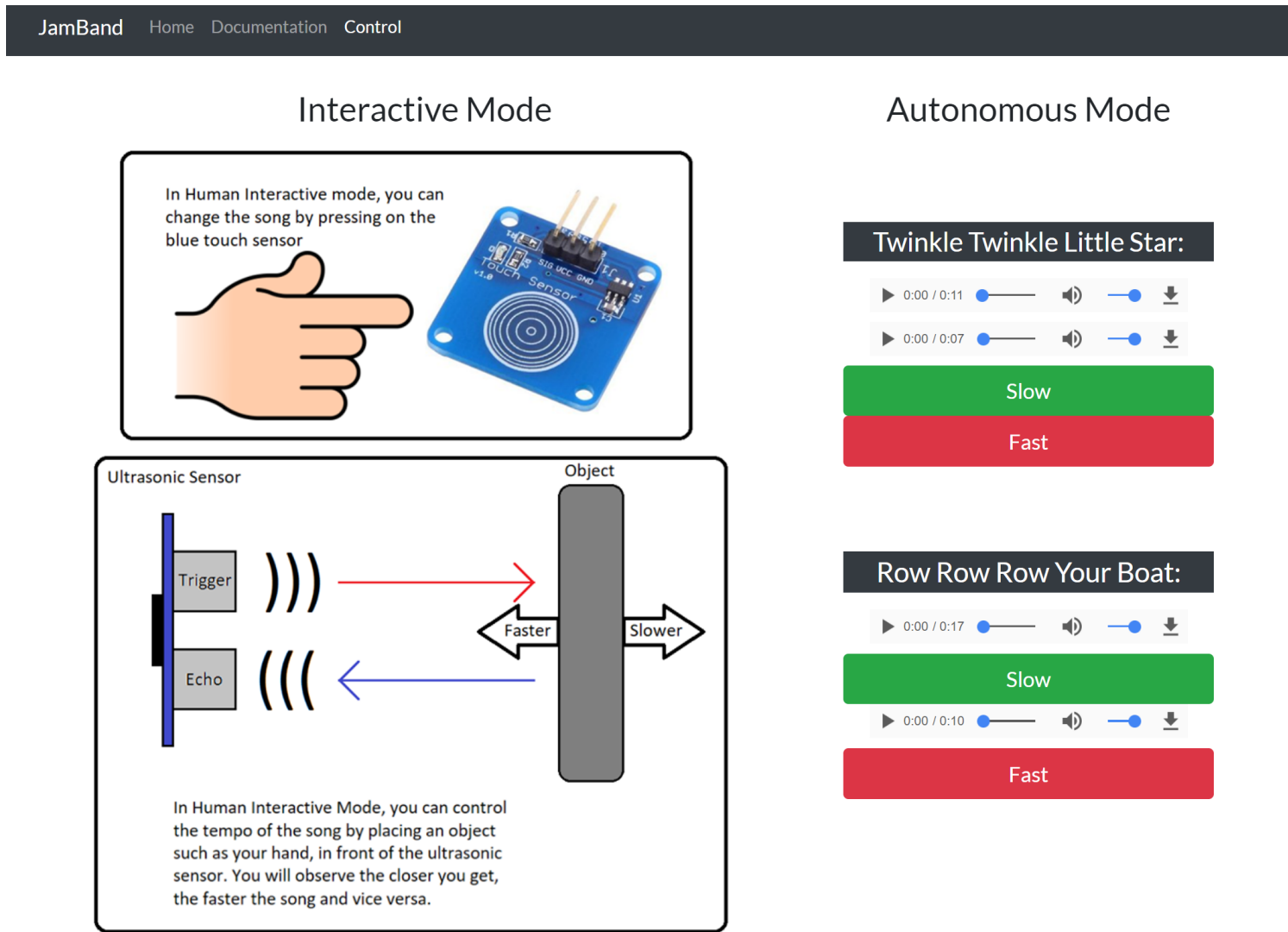


Figure 2: Port Forwarding in Router Setting

Now if we go to our router IP address, we should be able to get the control page from anywhere whether using WIFI or cellular data. All the above work makes this beautiful user interface possible:



## Potential Improvement

I would like to mentions at last about some potential improvements of this project. As listed below,

1. The servo end angle consistency of the two songs. Since the two songs are composed by two teammates individually, the end angle are not the same. As a result, the drum has to move to the right position every time the song switched for the stick to hit the drum properly.
2. User interaction is not immediate. Since we programmed in each loop that the music is played for a 7-14 second. During the playing of the music, the sensors' value won't be measured and also the client request won't be accepted. This caused the problem that users has to interact with the Robot at the end of a loop cycle.
3. To continue my first point, possible improvements can be made to divide the song playing function in a very small segments so that the loop still repeat in a reasonably short time to do all the instant measurement and web service. However, this maybe challenging as to parameterize the song with each loop variables.

## Conclusion

In this poster, we guided your through a detail problem solving process in our project, including Music Composition, Serial Communication, and Wireless setup. We also address some potential improvements of our project such as inconsistency of servo and immediate user interaction.

## References

- [1] CrtSuznik. *Control ESP8266 Over the Internet from Anywhere*. Instructables, 2016.

## Acknowledgments

I would like to thank my EE183 Professor Mehta for this awesome project.