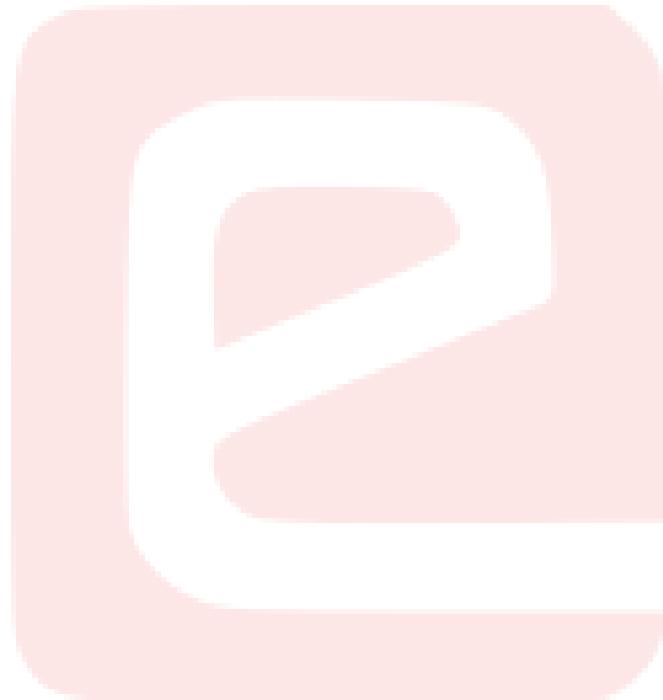


eYSIP2016

SIGN LANGUAGE INTERPRETER USING LEAP MOTION SENSOR



Intern: Sanket R Bhimani

Mentor1: Aditya Panwar

Mentor2: Rama Kumar

Duration of Internship: 21/05/2016 – 10/07/2016

2016, e-Yantra Publication

Sign Language Interpreter Using Leap Motion Sensor

Abstract

This project will recognize the gestures performed by hearing impaired or verbally challenged persons and convert it into natural audio. Here, a set of words is taken as input and is then converted into a full grammatically correct sentence using NLTK library.

Afterwards, these words in the sentence are mapped with pre-recorded audio file stored in the MP3 module and finally sentence is interpreted through the Galileo Board and is audible via mp3 module to the speaker/headphone.

The other part of the project is a robotic hand that tries to imitate the actions of a person's hand when performed over the Leap Motion Sensor. Here Leap sensor's data is mapped with servos' angle through Galileo Board which controls the robotic hand.

Completion status

Both parts of the project are almost completed. However, in Sign Language Interpreter, making of compact and portable version is remaining. In Robotic Hand making of power supply is left. Currently it is working on LAB DC power supply.

1.1 Hardware parts

- Galileo Board [Datasheet](#), [Vendor link](#)
- FN-M16P MP3 module [Datasheet](#), [Vendor link](#)
- Leap Motion Sensor [Documentation](#), [Vendor link](#)



1.2. SOFTWARE USED

- GS-5515MG Servo [Datasheet](#), [Vendor link](#)
- Any Speaker
- 3D Printed parts of Robotic Hand(STL files given in GitHub Repository)

1.2 Software used

- Leap SDK, [download link](#),
- BitVise SSH Client: version, [download link](#),
- Linux For Galileo [download link](#), For installation read Galileo Tutorial
- Python Programming Language 2.7 [download link](#)

1.3 Assembly of hardware

For Sign Language Interpreter,

Circuit Diagram

See Figure: 1.1

Step 1

Connect Leap Motion Sensor with PC with USB connection

Step 2

Connect Galileo Board with PC through WIFI or LAN

Step 3

Connect MP3 Module With Board.(Just give VCC and GND to module and board's Rx pin to Module's Tx pin and board's Tx pin to module's Rx pin.)(Refer data sheet of MP3 module for pin diagram)

1.3. ASSEMBLY OF HARDWARE

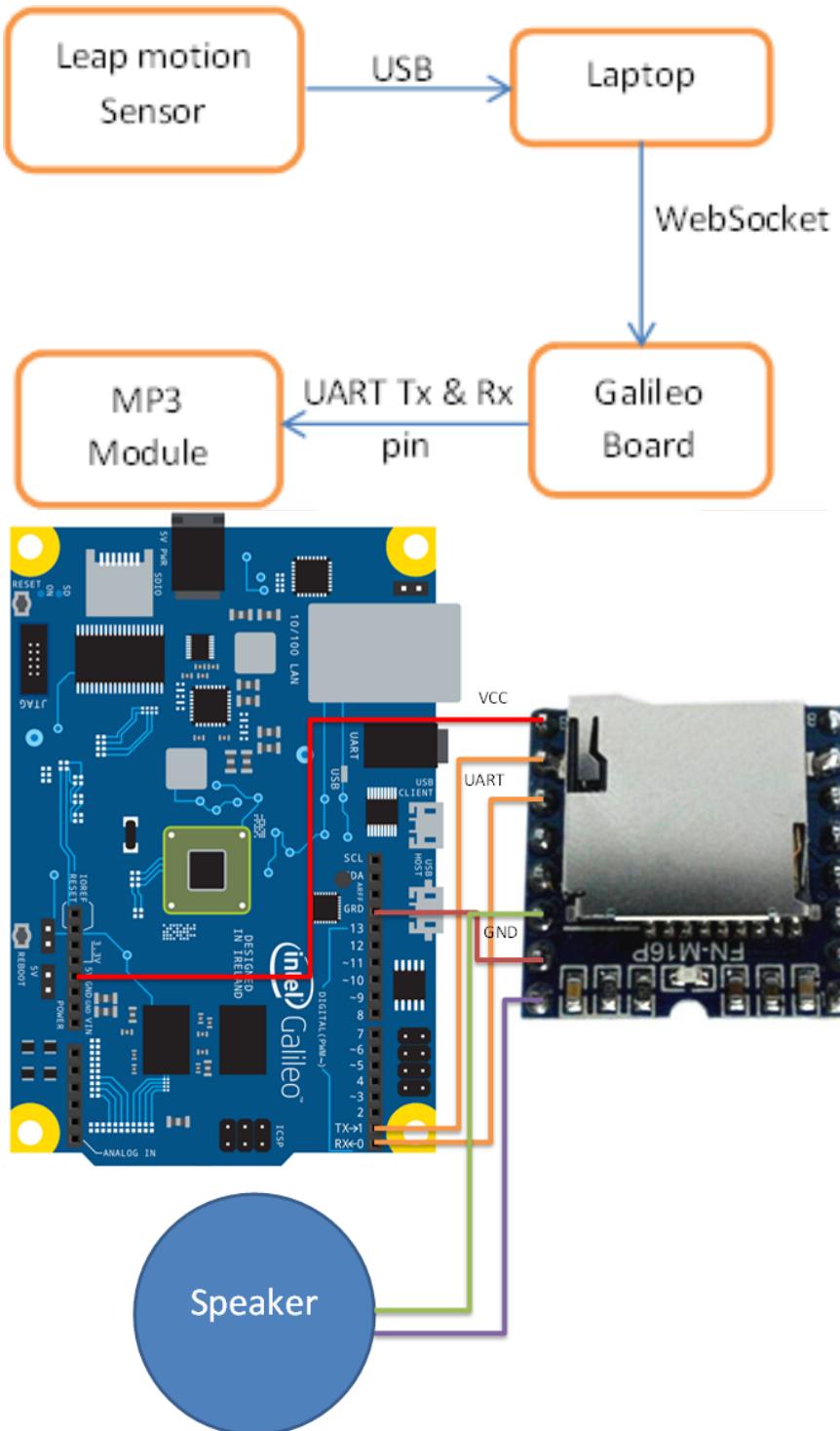


Figure 1.1: Block diagram for SIGN LANGUAGE INTERPRETER hardware system & Connection between Galileo board and MP3 module

1.3. ASSEMBLY OF HARDWARE

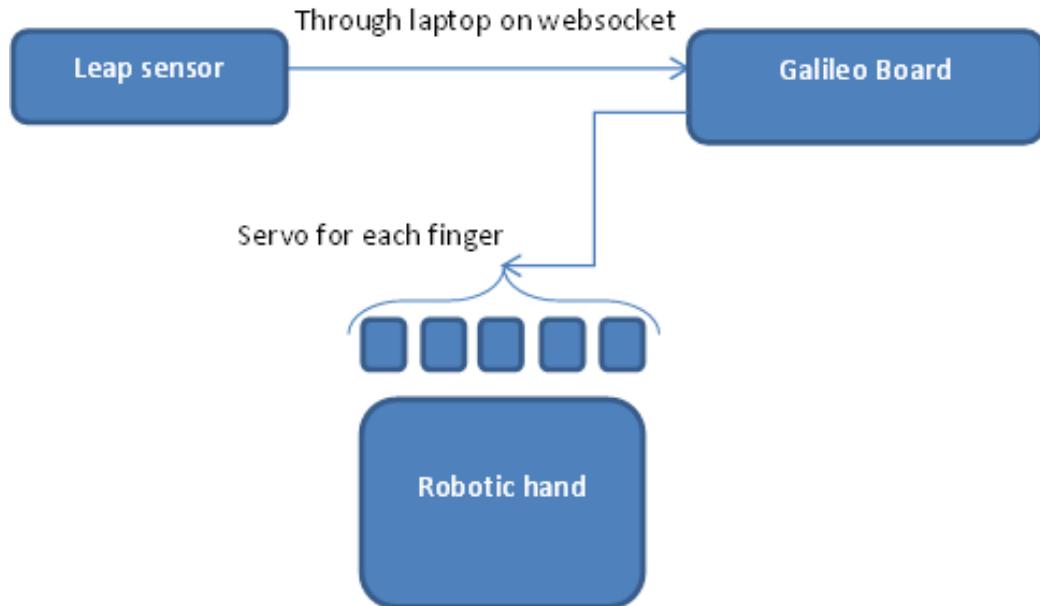


Figure 1.2: Block diagram for Robotic hand system

For Robotic Hand,

Circuit Diagram

See Figure: 1.2

Step 1

Assemble all 3D printed parts of Hand. Refer [this](#)

Step 2

Connect Leap Motion Sensor with PC with USB

Step 3

Connect PC with Galileo using LAN or WIFI



1.4. SOFTWARE AND CODE

Step 4

Connect PWM pin of servo to Board's PWM output pin(For Connection refer Galileo_board_tutorial)

Provide 5V DC power supply to each servo. Make sure, 700mA current is provided to each servo.

1.4 Software and Code

[Github link](#) for the repository of code

For Sign Language Interpreter:

There are two programs, one is for PC/Laptop and another is for Galileo Board. One script will be running on board to catch messages sent through PC. This script will also handle all the tasks related to the interfacing of MP3 module to play any audio file via UART communication. And One script will be running on PC to recognize Gestures and interpret them in words. The second script will be running on PC/Laptop to recognize gestures and interpret them as words. The tasks related to making natural sentences (grammatically correct) from words and making grammar are also carried out by this particular script. This thing will be done through NLTK library.

Task Achieved with each program:

on_laptop.py

- **Recognizing Gesture:**

For gesture recognition two parameters each figures position and direction of all three axis are counted. And also relate them with palm center to make independent of palm position from sensor. Now program takes 300 frame to record the gesture and then all data is stored in JSON file as a permanent copy. Now when program is restarted all gesture will be loaded from that JSON file. For recognizing gesture, again it will take 300 frame and compare them with saved frame. Then gesture with higher frames matched is become the result. For making this system faster, all data is converted into matrix and all processes are done through **numpy**.



1.4. SOFTWARE AND CODE

- **Making of sentence:**

After Recognizing single words, a set of words is generated. Then this set become sentence here. For example, if set is ['What', 'name', 'you'], then here some words are added to make proper sentence and it also corrects the places of word. So, here the output for above set will be ['what', 'is', 'you', 'name']. Then grammar will be corrected in next module.

- **Correcting Grammar**

Here sentence's grammar is corrected. For that, I have used online API of Ginger grammar. So this will make ['what', 'is', 'you', 'name'] to ['what', 'is', 'your', 'name'].

- **Sending sentence to board**

Then this sentence is sent to board through websocket here this code become client and board become server.

on_board.py

- Receives the sentence through websocket.
- Differentiate each word from sentence and map with file number.
- Generate hex code for MP3 module
MP3 module needs hex code to perform specific task. So here proper hex code is generated. hex code also includes check bit
- Sending hex code to MP3 module
then this generated hex code is sent to mp3 module through UART.

For Robotic hand:

Laptop side program

Here Leap data is mapped to servo angle. Here, two parameters are used, one is **tip position's y axis** and **palm center's y axis**. And **tip position's y axis** is relate to palm center. So, it become independent from hand position. Then Send data of all five angle of each finger to board through websocket

Board side program

It just receive angle and data and generate PWM signal according to that.



1.5. USE AND DEMO

1.5 Use and Demo

For Sign Language Interpreter

[Youtube Link](#)

See Figure: 1.3

step 1:

Connect board and mp3 module. As shown in Figure: 3

step 2:

Connect board with SSH to pc

step 3:

Enter IP of board in on_laptop.py

step 4:

run on_board.py on Galileo board.

step 5:

run on_laptop.py on laptop

For better recognition, sit on chair and put leap on table and before performing any new gesture show whole palm first.

For Robotic Hand

[Youtube Link](#)

See Figure: 1.4

step 1:

Make all connection with board, servos, leap and pc

step 2:

connect board with SSH to pc

step 3:

Enter IP of board in follow_hand_client.py

step 4:

run follow_hand_server.py on Galileo board.

step 5:

run follow_hand_client.py on laptop

Now perform pose on leap at least 15CM above the sensor.

1.5. USE AND DEMO

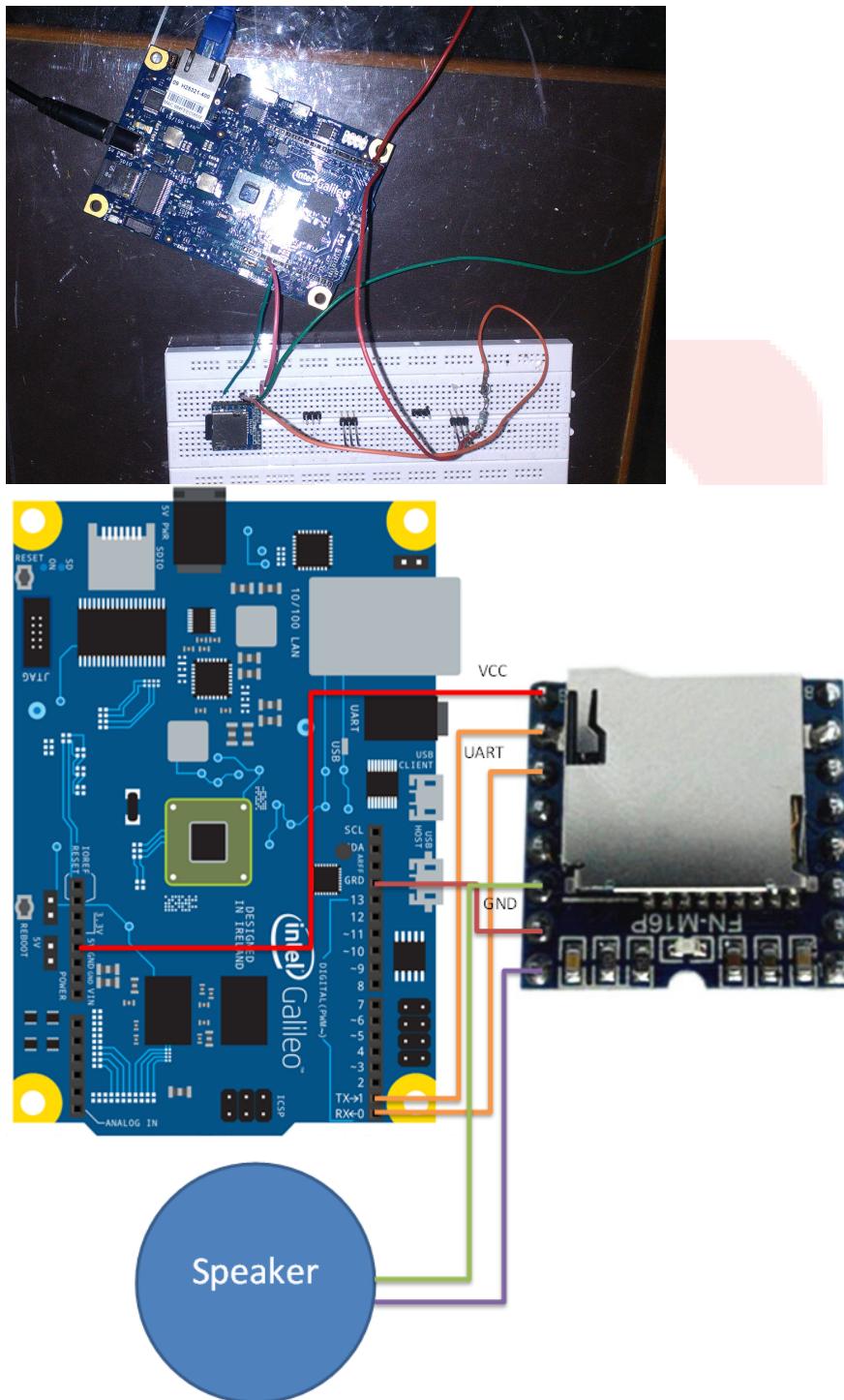


Figure 1.3: Connection between MP3 module and Galileo

1.6. FUTURE WORK

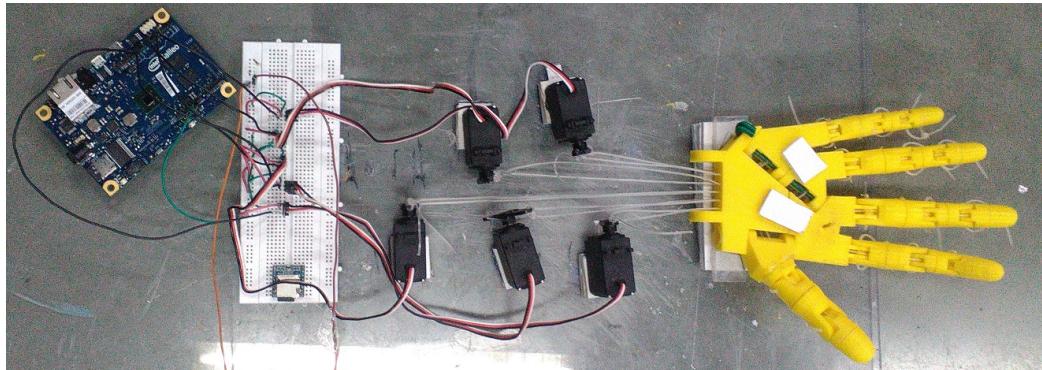


Figure 1.4: Connection for robotic hand

1.6 Future Work

For Sign language Recognizing system,

- Make whole system portable and compact like a tablet and make it more accurate.
- Word dictionary can be reach with all necessary words.

For robotic hand,

Make more joints accessible.



1.7. BUG REPORT AND CHALLENGES

1.7 Bug report and Challenges

Bug report:

The gesture of people other than the trainer who trained the system does not produce efficient output.

Challenges:

Initially, Reorganization was not accurate. Error rate was almost 50%. that was some LeapTrainer.js library. So it is decided to make own thing. So one algorithm is made, Now challenge is to make it faster. Because it was too slow. It take 18-21 seconds to recognize single word. So, all data is converted in form of matrix. For processing of that data numpy library is used. Numpy library complete tasks at most efficient and faster way. So reorganization become more faster, Now it takes only 0.3 second to recognize word.

In robotic hand there was a problem of power supply because it needs 5V with around 5A power supply. It was quit hard to build such supply. So, currently I am using LAB DC power supply.

Bibliography

- [1] Robotic hand STL files, <http://inmoov.fr/hand-and-forarm/>.
- [2] <https://github.com/intel-iot-devkit/mraa/>
- [3] www.developer.leapmotion.com/documentation/python/index.html
- [4] www.nltk.org
- [5] www.youtube.com/watch?v=FLZvOKSCkxY
- [6] www.gingersoftware.com/grammarcheck
- [7] www.github.com/zoncoen/python-ginger
- [8] www.blog.livedoor.jp/xaicron/archives/54466736.html
- [9] www.youtube.com/watch?v=gN3CGhFPF1s
- [10] www.downloadmirror.intel.com/26028/eng/iot-devkit-prof-dev-image-galileo-20160525.zip
- [11] www.software.intel.com/en-us/iot/hardware/galileo/downloads
- [12] www.downloadmirror.intel.com/25384/eng/w_galileo_2015.0.010.exe
- [13] www.youtube.com/watch?v=yrRMomesBKM
- [14] [www.github.com/intel-iot-devkit/mraa](https://github.com/intel-iot-devkit/mraa)