Sign Language Interpreter

Using Leap Motion Sensor

Sanket R Bhimani

July 8, 2016

Contents

1 Tutorial Name:

Sign language Interpreter
Using Leap Motion Sensor

1.1 Objective:

Objective behind this project is to help hearing-impaired and verbally challenged people to communicate with normal people. So this system will try to interpret the sign language into natural voice.

1.2 Abstract:

Here we are using leap motion sensor for detecting gesture and then we will convert in audio with mp3 module. Before conversion we are converting a set of words to a natural sentence using NLTK and ginger grammar API.

2   Prerequisites:

Before going ahead it is better to learn these basic things

- Basic knowledge of Python
- Basic knowledge of Linux
- Basic knowledge of NLTK
- Basic Electronics knowledge

3   Requirement

- Hardware Requirements:

  - Galileo Board
  - Leap Motion Sensor
  - microSD card of minimum size of 8 GB with card reader
  - speaker
  - MP3 Module
  - LAN cable or wifi module(for connecting board with PC)

- Software Requirements:

  - Python editor
  - Leap sdk
  - NLTK for Python
  - Tornado Web-socket
  - Bitwise SSH Client
  - Wireshark(optional)

4　Theory and Description

Essential topics to learn before starting the project:

- - Leap motion library for Python
  - LeapTrainer.js
  - Python programming
  - Websocket
  - getting started with Galileo board
  - interfacing with MP3 module

4.1 Leap motion Library:

In python library all class names are given as human body part so it is very easy to understand. Main and useful classes are,

- Frame

- Hand

- Finger

- Pointable

- Arm

For getting data from leap sensor first we need to register listener class with controller object and for that first we need to create listener class. And register that listener class object with controller object you can find the basic code for getting basic frame data from given code examples "connection_with_leap.py"

4.1.1 Frame:

A frame is an object which can be generated from controller object. Leap sensor sends data through this frame object in listener class. From frame object, we can find all other objects. Means all row data comes with the frame. As per leap FPS on_frame(self,controller) method of listener class is called with a new frame. Each frame has its unique id and time stamp.

In on_frame method from controller object, we can get the current frame. It also supports serialization.

4.1.2 Hand:

Hands are the main entity tracked by the Leap Motion controller. The Hand class represents a physical hand detected by the Leap. A Hand object provides access to lists of its pointables as well as attributes describing the hand position, orientation, and movement.

From current frame object, we can get the current set of detected hand as an array. We can get hand from hands object by these properties,

```
1  frame = controller.frame()
2  hands = frame.hands
3
4  leftmost = hands.leftmost
5  rightmost = hands.rightmost
6  frontmost = hands.frontmost
```
Listing 1: getting frame and hand object

We can get these characteristics from hand object,

- isRight, isLeft — Boolean variable which checks whether the hand detected by leap motion sensor is right or left.

- palm_position — The center of the palm measured in millimeters from the Leap Motion origin.

- palm_velocity — The speed and movement direction of the palm in millimeters per second.

- palm_normal — A vector perpendicular to the plane formed by the palm of the hand. The vector points downward out of the palm.

- direction — A vector pointing from the center of the palm toward the fingers.

- grab_strength, pinch_strength — Describe the posture of the hand.

- stabilized_palm_position — The stabilized palm position of this Hand.

- Motion factors — Provide relative scale, rotation, and translation factors for movement between two frames.

We can get a pitch, yaw, and roll of hand by these properties,

```
1 pitch = hand.direction.pitch
2 yaw = hand.direction.yaw
3 roll = hand.palm_normal.roll
```
Listing 2: getting pitch yaw and roll from hand object

4.1.3 Figures & Pointables:

Fingers are Pointable objects that the Leap Motion software has classified as a finger. Get valid Finger objects from a Frame or a Hand object. This can be generated from frame object or hand object. As this way,

```
1 frame_pointables = frame.pointables
2 hand_pointables = hand.pointables
3 known_pointable = hand.pointable(known_id)
4 finger = Finger(pointable)
```
Listing 3: getting figures and pointables

If pointable is detected as the particular figure then and then only it can be returned as finger object. But mostly this pointer is detected as a finger. We can get a bone object from finger object and get the position of

all joints.

```
1 bone = finger.bone(Bone.TYPE_PROXIMAL)
```

inputs:

- 0 = TYPE_METACARPAL – The metacarpal bone.

- 1 = TYPE_PROXIMAL – The proximal phalanx; the closest bone segment to the hand.

- 2 = TYPE_INTERMEDIATE – The intermediate, or middle, phalanx.

- 3 = TYPE_DISTAL – The distal phalanx; the bone segment furthest from the hand.

```
1
2  Joint_position = finger.joint_position(0)
3
```

inputs:

- 0 = JOINT_MCP – The metacarpophalangeal joint, or knuckle, of the finger.

- 1 = JOINT_PIP – The proximal interphalangeal joint of the finger. This joint is the middle joint of a finger.

- 2 = JOINT_DIP – The distal interphalangeal joint of the finger. This joint is closest to the tip.

- 3 = JOINT_TIP – The tip of the finger.

We can get this type of properties from pointable object,

- id - A unique ID assigned to this Pointable object.

- Frame - The Frame associated with this Pointable object.

- Hand - The Hand associated with this finger.

- tip_position - The tip position in millimeters from the Leap Motion origin.

- tip_velocity - The rate of change of the tip position in millimeters/second.

- direction - The direction in which this finger is pointing.

- width - The estimated width of the finger in millimeters.

- length - The estimated length of the finger in millimeters.

- is_finger - Whether or not the Pointable is classified as a finger.

- stabilized_tip_position - The stabilized tip position of this Pointable.

- time_visible - The duration of time this Pointable has been visible to the Leap Motion Controller.

4.1.4 Arm:

There is also an arm object which gives data related elbow_position, arm.wrist_position, width, direction, the basis of all three axises.
Arm object is generated from hand object.

```
1 hand = frame.hands.frontmost
2 arm = hand.arm
```

attributes:

- basis:
   Basis vectors specify the orientation of an arm.

   - x_basis. Perpendicular to the longitudinal axis of the arm; exits leterally through the sides of the wrist.
   - y_basis or up vector. Perpendicular to the longitudinal axis of the arm; exits the top and bottom of the arm. More positive in the upward direction.
   - z_basis. Aligned with the longitudinal axis of the arm. More positive toward the elbow.

```
1 basis = arm.basis
2 x_basis = basis.x_basis
3 y_basis = basis.y_basis
4 z_basis = basis.z_basis
```

```
5 center=arm.elbow_position+(arm.wrist_position−arm.
    elbow_position*.05
6 arm_transform = Leap.Matrix(x_basis, y_basis, z_basis,
    center)
```

- direction - The normalized direction of the arm from elbow to wrist.

- elbow_position - The position of the elbow.

- width - The average width of the arm.

- wrist_position - The position of the wrist.

Here we will use only some properties like,

- In hand,
    - direction
    - palm_position
    - stabilized_palm_position

- In pointable,
    - direction
    - tip_position
    - stabilized_tip_position

4.2 LeapTrainer.js:

This API is used gesture and pose learning and recognition framework for leap motion sensor. This is javascript based API, so using that framework we can detect gestures and poses in a web browser, and then using websocket it will go to python program. So here is the overview of this API, **Uses:**

```
1 <script src="http://js.leapmotion.com/0.2.0/leap.min.js"></
     script>
2 <script src="/path/to/leaptrainer.min.js"></script>
```

You need to include these two files for using this framework. The first file is for connection with leap sensor and second for this framework. Then we need to create leap sensor's object and trainer object

```
1 var leapController = new Leap.Controller();
2 var trainer = new LeapTrainer.Controller({controller:
     leapController});
3 leapController.connect();
```

4.2.1 Training the system:

We can train the system by adding new gesture or pose to system, and we can add new gesture or pose by calling this method,

```
1 trainer.create('Halt');
```

here in place of 'Halt' give any name related to your gesture or pose. And we can do something when this gesture is recognized like,

```
1 trainer.on('Halt', function() { console.log('Stop right there!')
     ; });
```

4.2.2 Importing and Exporting from LeapTrainer:

This feature is most useful in real application. Because normally wat happened whatever data we create like learning gestures or poses, are temporary means once you reload the script all objects will be reinitialized so all saved gestures or poses data will be lost. So by using this feature we can save that data.

This framework exports data in the form of JSON and also accepts imports in the form of JSON.

```
1  var savedGesture = trainer.toJSON('Halt');
2  //Exporting to JSON
3
4  trainer.fromJSON(savedData);
5  //Importing from JSON
```

This can be used for saving and fetching gesture or pose data into or from file

More details about LeapTrainer library is can be found on github. But we are not using this library because there is many error. like, it autometically detects gsutures, so when we are going to perform any gestures it will recognize another one before performance. so it become annoying. So, I have decided to make my own gesture rcognizing system we will see about it letter
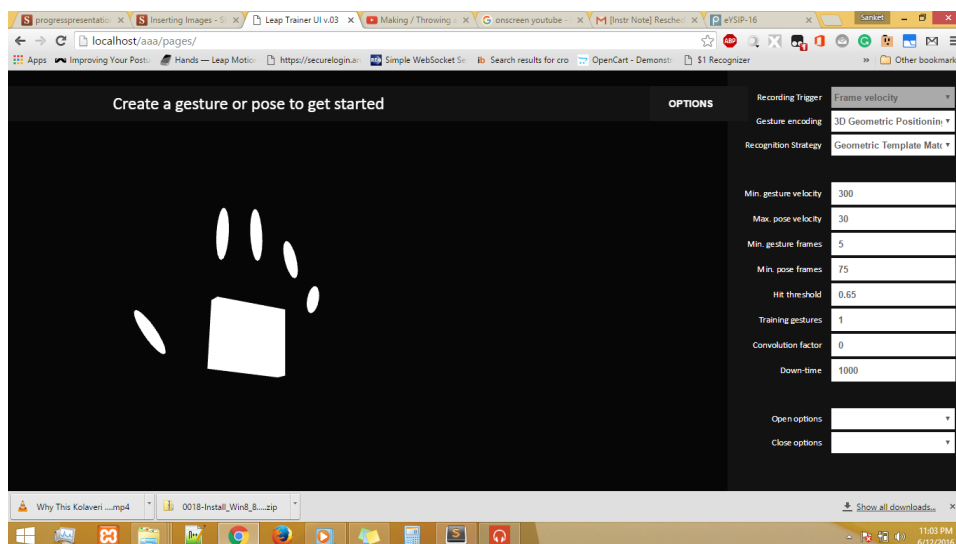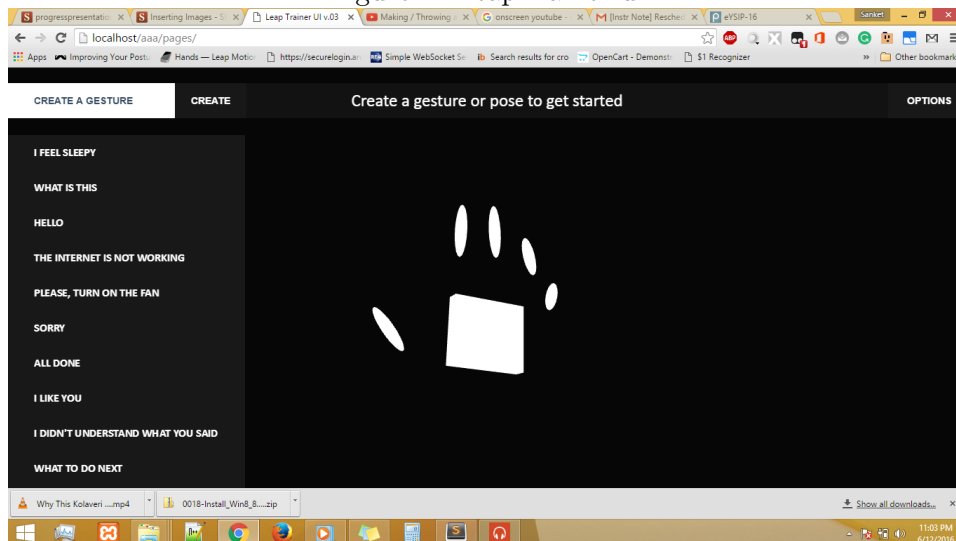
Figure 1: LeapTrainer-ui

Figure 2: LeapTrainer-ui

4.3 Installation of Linux image in Galileo board:

In Galileo board, we can install Linux, windows IOT or os x. So the interface with board become easy. Here we will use Linux image. Because currently windows IOT image is no more supported. And everyone is not used to OS X. For that, we need to write Linux image on the sd card. From sd card, Galileo board will boot.

4.3.1 How to write Linux image on Galileo

Step 1:
Download Linux image from this location.
https://downloadmirror.intel.com/26028/eng/iot-devkit-prof-dev-image-galileo-20160525.zip
You will get one zip file extract it. You will find one '.direct'. You need to write this into sd card

Step 2:
Download this software from this location,
http://download.softpedia.com/dl/d5a643aad00816ee735372c2d530a1a2/57584b56/100173006/softw
0.9.5-binary.zip
this will be helpful for writing an image in sd card.

Select Image File which you have downloaded earlier in 'Image File' tab('.direct' file)
Select sd card in 'Device tab'
Now click on 'Write' button, it will take some time. And wait until write process completes

Step 3:

Now insert sd card in galileo board and connect LAN cable with your PC and galileo board. Now connect the power supply with galileo board.
now the board will automatically boot. There will be blinking LED near sd card port. Wait until it stops lighting. Because it shows
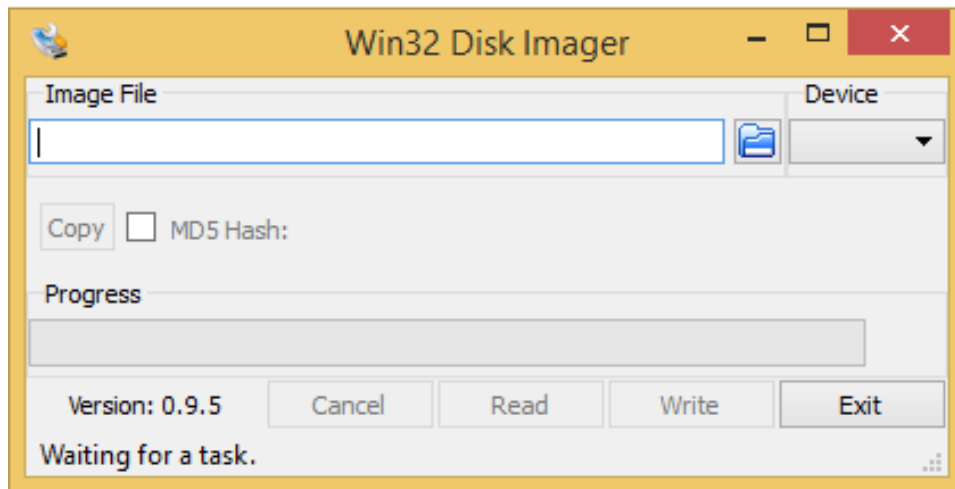
Figure 3: Win32 disk Imager window

IO related sd card. So, when it become stop, IO related sd card is also stopped so the booting process is completed. Now you need to find the IP address of your board. For that, you can use any network packet tracer software like Wireshark, ettercap or any other.

I am using Wireshark.

Now open Wireshark select ethernet as a network adapter. It will start tracing all packet data on ethernet. But here on an ethernet network, there is only two devices are there your PC and board. So there is only two IP address data. (ignore IPs with 255, 251, 224 and more than 200 value in last two part like 192.168.0.255 or 255.255.255.0) so find your PC IP address using this command

```
1 ipconfig (for windows)}
2 ifconfig (for Linux)}
```

Now another IP showing on Wireshark is your board IP.
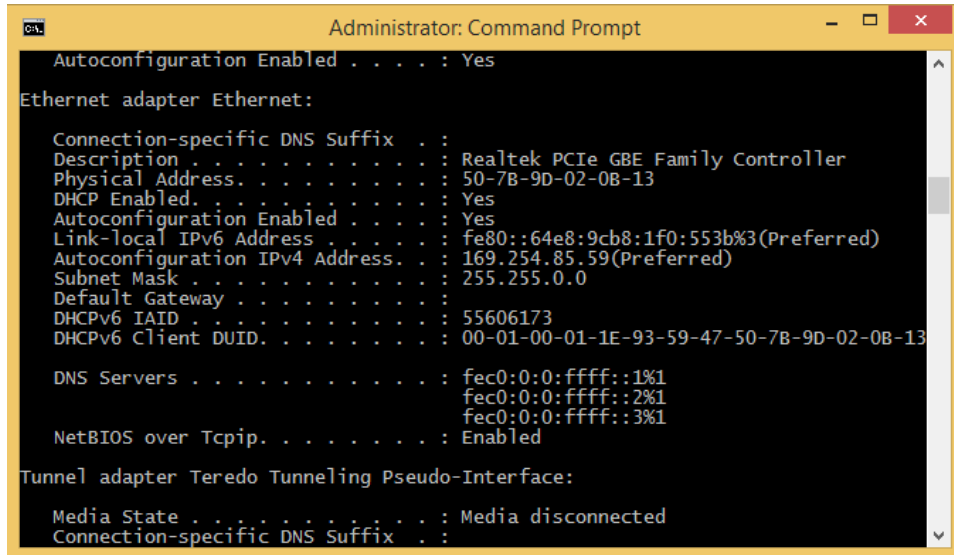
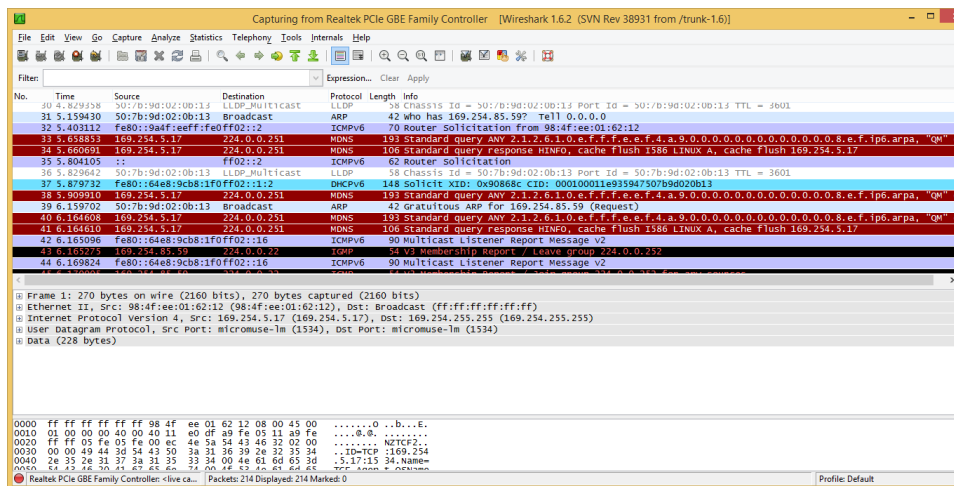Your Wireshark windows:

Figure 4: Your IP address
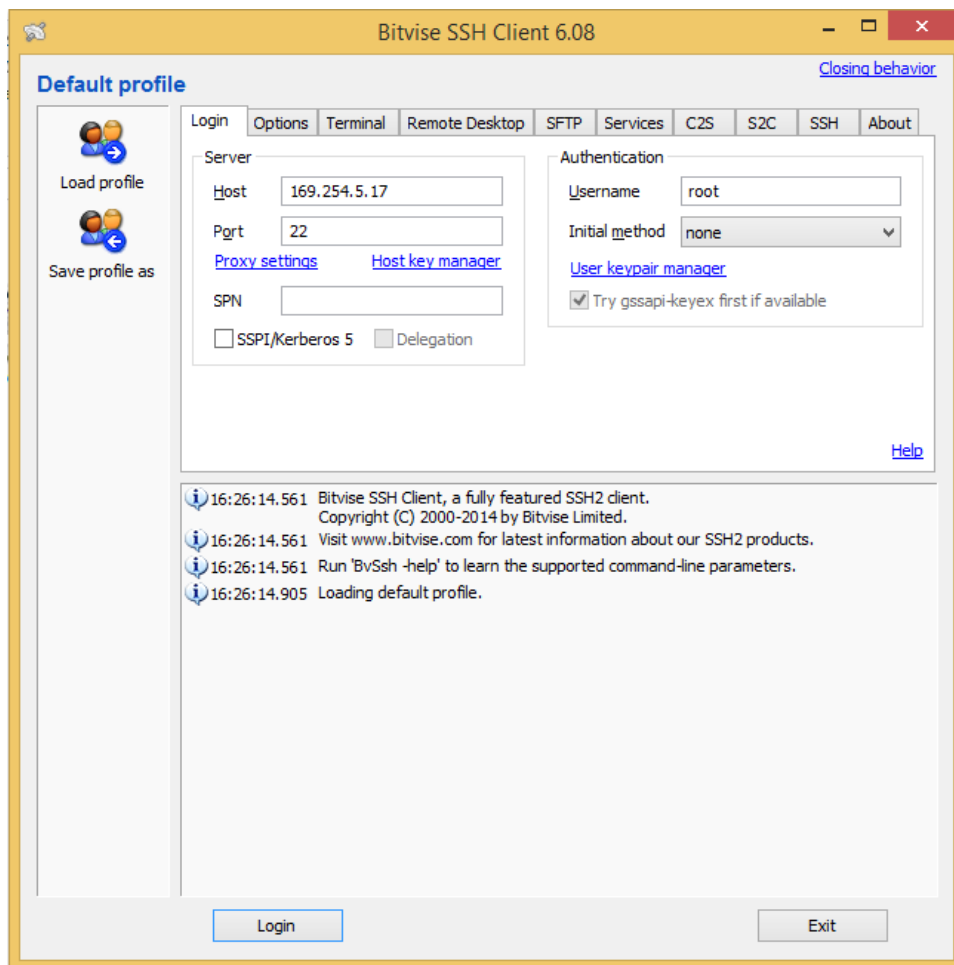

Figure 5: wireshark window

Figure 6: SSH client

Step 3:

Now, from any SSH client make connection with board and use. :?)
(I'm using Bitvise SSH client)
Enter IP address and username: root there is no password.
Click on login, and wait. Now two windows will open one for terminal and another for file transfer. Your board is ready! :?)
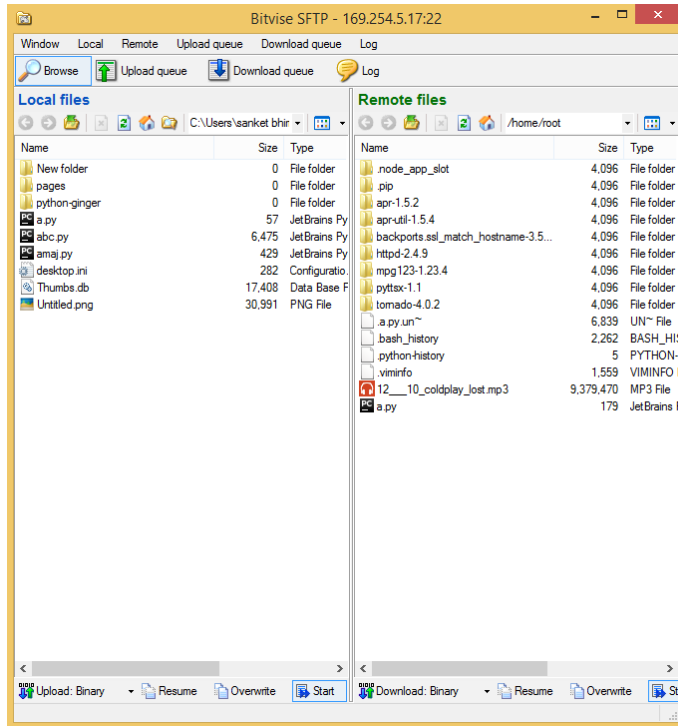
Figure 7: SFTP window

Figure 8: SSH terminal

4.4 Interfacing MP3 module:

MP3 module is used to play audio files stored in separate memory card attached with module. Control of this module is done the hex code sent from any controller via UART port. Here we will send code from Galileo board from Rx Tx pin to the board, there are some specified method to generating hex code which you can find on it's data sheet. We will see about that in next topic
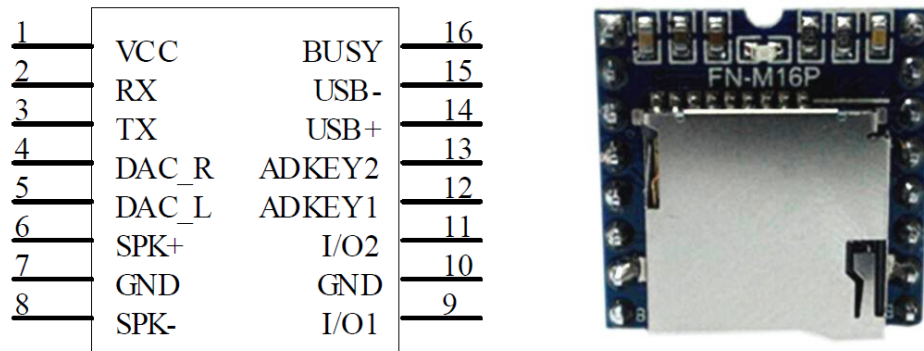
Figure 9: Pin diagram

No	Pin	Description	Note
1	VCC	DC3.2~5.0V	
2	RX	UART serial input	
3	TX	UART serial output	
4	DAC_R	Audio output right channel	Drive earphone and amplifier
5	DAC_L	Audio output left channel	Drive earphone and amplifier
6	SPK2	Speaker-	Drive speaker less than 3W
7	GND	Ground	Power GND
8	SPK1	Speaker+	Drive speaker less than 3W
9	IO1	Trigger port 1	Short press to play previous (long press to decrease volume)
10	GND	Ground	Power GND
11	IO2	Trigger port 2	Short press to play next (long press to increase volume)
12	ADKEY1	AD Port 1	Trigger to play first segment
13	ADKEY2	AD Port 2	Trigger to play fifth segment
14	USB+	USB+ DP	USB Port(connected with a USB flash or connected to PC)
15	USB-	USB- DM	USB Port(connected with a USB flash or connected to PC)
16	Busy	Indicator Pin	Low level when working, and high level when standby

Figure 10: Table for pin diagram

5   Implementation:

- Alphabet training and recognizing with leap motion

- Make natural sentence from words using NLTK

- Make simple player for MP3 module

- Other Experiment for gesture recognizing.

- First version of this system with LeapTrainer.js

- second and final version of this system

5.1 Alphabet training and recognizing with leap motion:

Here we want to make a programme which records or say learns pose hand and then it will recognize. Means there will be two mode learning and recognizing. In Learning mode we need to perform pose so this pose will be saved for future reference and need to provide one alphabet for that pose. and In recognizing mode we need to perform specific pose so programme will recognize that pose and give related alphabet

Here we have used pointables x direction, pointable z direction, pointable x stabilized tip position, pointable z stabilized tip position, stabilized palm position of x axis and z axis.

Here we want to implement a programme which is independent of hand's position means we can perform action at any position in range of sensor. so we have take data relative palm center so we have used stabilized palm position but here only x and z axis is useful because we don't need to care about height from sensor.

Here we will find radius distance from palm center to every tip of finger so we need to find mean position of tip's x and z co-ordinates related to palm center

We are also taking direction vectored so we can increase range of detection of radius distance

Then sum interface related programming. whole programme is in python language you can find at github `https://github.com/sanketbhimani/`

eYSIP-2016-Sign-Language-Interpreter-Leap-Motion-Sensor-/blob/
master/alphabet_using_leap.py

Problems faced:
    While copping object I was directly write like a = b. so programme
was not working properly and only large performed pose goes to every ob-
ject. I was not understanding what was happening. what was happening,
when I write a = b it just transfer a pointer no new memory is allocated so
when I change value for one object all values will change because they all
are pointing to same memory location. then i used a = copy.deepcopy(b)
then it will allocate new memory for that object. and it is working fine!!!

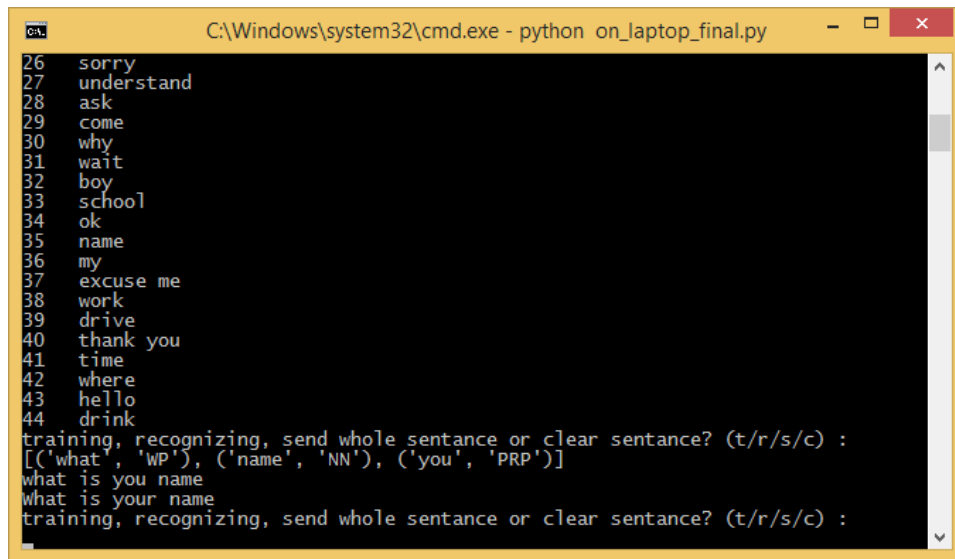## 5.2 Make natural sentence from words using NLTK:

In sign language, normally whole sentence is not actioned mean little grammar thing like "is" are not much important. because action just want to show the understanding about what he want to tell. they not try to build a complete sentence with correct grammar.

So, this programme will help to create a proper natural sentence with correct grammar. like if you do input like "WHAT NAME YOU", means you want to ask a name someone this should be like "WHAT IS YOU NAME?" so this type of conversation is done by this programme

For that I have use NLTK library, NLTK have lot of data to compare and detect what is this word means this is verb, pronoun, noun etc. Let take one example, "make a note" here note is noun but "note it down" here note is verb so for different cases same word can perform different role, so NLTK helps you to detect the role of words, it also helps you to organize sentence and word at smart way.

After detecting the role of this words, it's our job to make a proper sentence. This programme will put them in proper sequence and add necessary words like "is". then it will correct this sentence grammar. like if the input is like "WHAT NAME YOU" our programme generate new sentence like "WHAT IS YOU NAME" then after correcting grammar this will be like "WHAT IS YOUR NAME?" For correcting grammar I have use online API, it's ginger grammar. So you need to just send raw sentence and as a res ponce you will get proper correct sentence with some extra information as a JSON data you need to fetch correct sentence.

So you need to learn basic NLTK library and sending and receiving HTTP request and response using python.

Figure 11: Screen-Shot for NLTK

5.3 Make simple player for MP3 module:

This is a simple programme designed to test and get learn an interfacing with MP3 module. This programme is basically mp3 player which have option of next, previous, play/pause, volume up and volume down.

This programme is in command line interface. And it generates appropriate hex code for perform certain task and send it to mp3 module through UART (Rx and Tx pin available on board). This includes of generating oh check bits and all.

Sample Code for this program is can find on github repository

5.4 Other Experiment for Gesture recognizing:

Now, we need to do gesture recognizing. means in first alphabet recognizing programme we have just recognize just pose but now we need to recognize whole motion. For that i was searching for works done on this topic. From leap developer community I found two useful repository first one is $1 unistock gesture recognizing library. If know about $1 and $n algorithm of university of Sweden. This algorithm compares two pattern and gives percentages matches. So what i found that some one have made like using only one pointable we need to perform action and system will recognize that action. but here we can not use standard sign language because here only liner actions can be performed. And it with too much errors. Means it does mistakes so much while recognizing.

So, finally i decided to not to use this system. So I search another library named LeapTrainer.js. It was good and i start working on it.

5.5 First version of this system with LeapTrainer.js:

This is based on web pages meas whole system was web based. it runs as website. So gesture training and recognising happens on browser. Means, here we need to train the system first (Loading of gesture data), Then system will recognize that gestures. But problem was that after loading data when we reload the page or reopen the page saved data ware destroyed. meas the data are not being permanently stored so we need to make some system for permanent saving of data.

As explined in theory there is a listener function for each events. So, when gesture-created event generated i store data of gestures in form of JSON using toJSON method. and in starting (means when we open the page) I load all gestures from JSON file. But here we want to make our system fully portable so JSON file is on Galileo board so the transfer of data is through Websocket. On board there is one python programme running which will create websocket server and our webpage be a websocket client. And all website data is also on board. we are just opening site from board to our laptop's browser. Here we need to connect Leap Motion Sensor on laptop side.
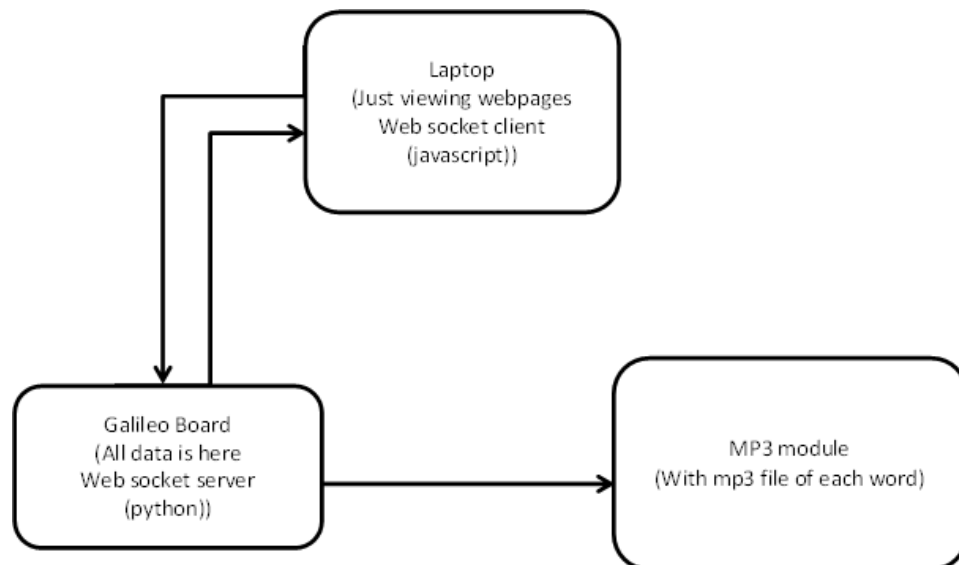
But here we also use that NLTK programme to convert sentence in natural language with correct grammar so we need to first pass the output of this web application to this programme. Now, this programme requires more processing power so it can not be run on Galileo board so we need to run it on laptop so we will pass the output to python script running on y laptop and then it sent to Galileo board. here also communication will be done through websocket.
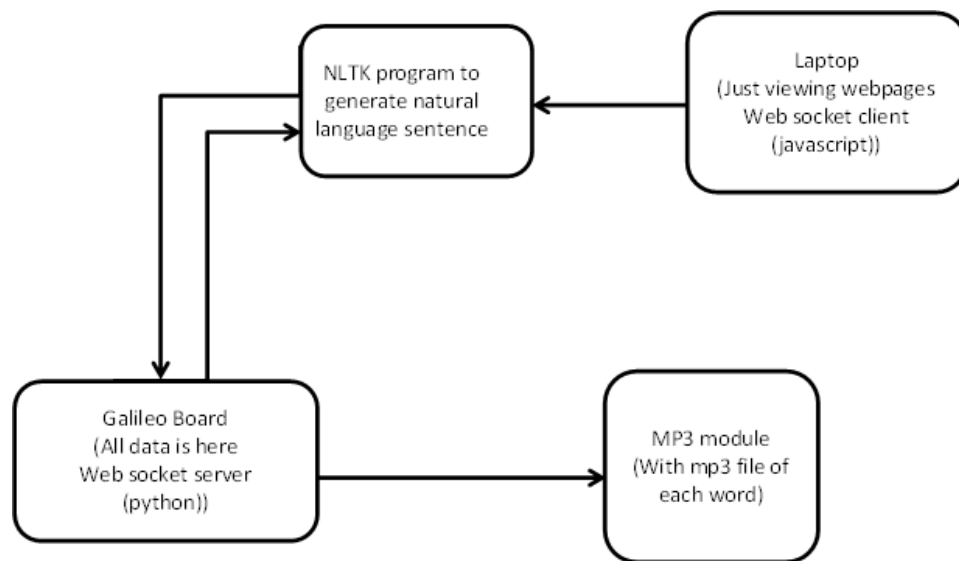
There is also some problem like they made mistake many time to recognising gesture and its interface is also quite annoying. So we need to think something new.

5.6 second version of this system:

Now, I have tried everything available on Internet related gesture recognising but nothing was efficient. So, I have decided to make my own programme to recognising gestures. And worked! So I've create my own algorithm to recognising gestures. It's very similar to the programme of alphabet recognition.If you remember in that programme we are taking average of 300 frames and we are using that average value for future comparison so what was happening that small movement while recording frames are gone by taking average but now in this case we want to capture the movements so we can not average. But we can directly compare that much frame to that much frame. So if sufficient frame will match, then we can say that, this is that gesture.

whole programme interface is command-line. There will be two mode of programme, Train and Recognize. So, in train mode we need to perform gesture so system will remember as a future reference. And in recognize mode we need to perform this gesture and system will recognize that gesture and give word matched with it.

Here, in each mode there is a timer of 3 second so you can set a proper position and check whether your each hand is detected properly or not. So, capturing of frame will start after 3 second. For changing of options and all thing see programme on github. There is well commented code so you can change every option like change the timer second or say change no of frame captures.

Here, NLTK programme also merged with main programme so output is directly sent to board and from board to MP3 module, Here all gesture's data will be in laptop. And the communication with board and laptop will be through websocket.

Here whole system is in python. there is two programme one for server and one for client.

- Task performed by client side programme:
  - Recognizing and Training of gesture.
  - Storing and Loading data to and from JSON file stored on same laptop/pc.
  - Add necessary words and Correcting the position of words.
  - Correcting the grammar.
  - Sending sentence to board through websocket.

- Task performed by server side programme:
  - Receiving sentence sent from client side programme.
  - Davide it into each word and find file name for that word in MP3 module
  - And play appropriate file stored in mp3 module through sending proper hex code.
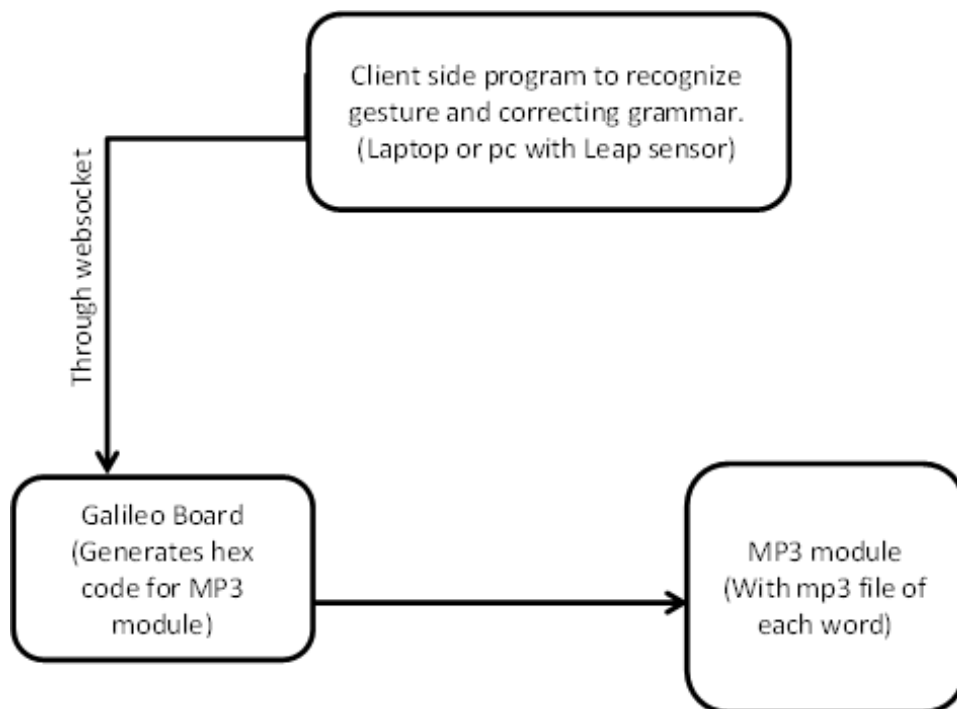
Now, there is only one problem need to be solved is speed. yes this system is very slow, because this algorithm is making 300*300*2*5*2*(number of words loaded) comparisons so, it takes time. one solution for this is to decrease frame captured but it affects on efficiency and accuracy of program so I need to find another solution.

5.7 Improved version of system

To make it faster I've tried many thing like with multi-threading, parallel processing, CUDA or many thing!!! but nothing works or say implemented! like in threading no improvement done because threading do same thing but with more threading so it not use more processing power. it just divide the process and do it in part.

And I tried to implement parallel processing but somehow I couldn't control on main thread so my main program is kept restarting and everything is happening so fast so i couldn't even found what is the error. So now I moved to CUDA. But it here also i failed to control on programme.

Now, i thought that let's do some improvement at normal way! So, I change the representation of data. first i was storing data in 4D list. So, it is becoming slow while changing the pointer from one location to another. So, I changed the representation of data storage to matrix form. So, it become just 2D list. And also I have converted it to numpy array. And also, all comparison I am doing is from numpy library. and it is the most efficient library to do process with matrix. So, finally it works! And now it is running faster. So there is no change in system but little change in calculation that makes my algorithm faster that is now I'm using.

6 Exercise

6.1 Finding Error rate for Alphabet recognising programme

Here is the result of experiments:

- "Yes" means recognize

- "No" means not recognize

Alphabet		test 1	test 2	test 3	test 4	test 5	test 6	test 7	test 8
A		yes	yes	yes	yes	yes	yes	yes	yes
B		yes	yes	yes	yes	yes	yes	yes	yes
C		yes	no	no	yes	yes	yes	yes	yes
D		no	yes	yes	yes	yes	yes	yes	yes
E		yes	yes	yes	yes	yes	yes	yes	yes
F		no	yes	yes	yes	no	yes	yes	yes
G		no	no	yes	yes	yes	yes	yes	yes
H		yes	yes	yes	yes	yes	yes	yes	yes
I		yes	yes	yes	yes	yes	yes	yes	yes
J		no	yes	yes	yes	yes	yes	yes	no
K		yes	yes	yes	yes	no	yes	yes	yes
L		yes	yes	yes	yes	yes	yes	yes	yes
M		yes	yes	yes	yes	yes	yes	yes	yes
N		no	yes	yes	no	no	yes	yes	yes
O		yes	yes	no	yes	yes	yes	yes	yes
P		yes	no	yes	yes	yes	yes	yes	yes
Q		yes	yes	yes	yes	yes	yes	no	yes
R		no	yes	no	yes	no	no	no	yes
S		yes	yes	yes	yes	yes	yes	yes	yes
T		yes	yes	yes	yes	yes	yes	yes	yes
U		yes	no	yes	yes	yes	yes	yes	yes
V		yes	yes	yes	yes	yes	yes	yes	yes
W		yes	yes	yes	yes	yes	yes	yes	yes
X		yes	yes	yes	yes	yes	yes	yes	yes
Z		no	yes	yes	yes	yes	yes	yes	yes

Result of alphabet recognition system

total: 26*8 = 208
Incorrect recognition: 23
So rate is: 23/208 = 0.11057 :)

7   References

References

[1] or learning python www.tutorialspoint.com/python/

[2] or Various doubt solution www.stackoverflow.com/questions/tagged/python/

[3] or understanding leap library in python www.developer.leapmotion.com/documentation/python/index.html

[4] or javascript gesture recognizing and learning system www.github.com/roboleary/LeapTrainer.js

[5] or $1 algorithm with Leap motion sensor www.github.com/liangzan/leap-demo

[6] ww.depts.washington.edu/aimgroup/proj/dollar

[7] ww.developer.leapmotion.com/libraries

[8] ww.stackoverflow.com/questions/tagged/javascript

[9] ww.stackoverflow.com/questions/tagged/jquery

[10] ww.t4t5.github.io/sweetalert

[11] ww.github.com/t4t5/sweetalert

[12] ww.pypi.python.org/pypi/nltk

[13] ww.nltk.org

[14] ww.youtube.com/watch?v=FLZvOKSCkxY

[15] ww.gingersoftware.com/grammarcheck

[16] ww.github.com/zoncoen/python-ginger

[17] ww.blog.livedoor.jp/xaicron/archives/54466736.html

[18] ww.youtube.com/watch?v=gN3CGhFPF1s

[19] ww.github.com/tornadoweb/tornado

[20] ww.pypi.python.org/pypi/tornado

[21] ww.gist.github.com/billroy/3761495

[22] ww.stackoverflow.com/questions/2835559/parsing-values-from-a-json-file-in-python/

[23] ww.stackoverflow.com/questions/29430333/simultaneously-reading-and-writing-to-json-file-in-python

[24] ww.downloadmirror.intel.com/26028/eng/iot-devkit-prof-dev-image-galileo-20160525.zip

[25] ww.software.intel.com/en-us/iot/hardware/galileo/downloads

[26] ww.downloadmirror.intel.com/25384/eng/w_galileo_2015.0.010.exe

[27] ww.youtube.com/watch?v=yrRMomesBKM

[28] ww.github.com/intel-iot-devkit/mraa

[29] ww.wiki.python.org/moin/ParallelProcessing

[30] ww.stackoverflow.com/questions/20548628/how-to-do-parallel-programming-in-python

[31] ttp://spartanideas.msu.edu/2014/06/20/an-introduction-to-parallel-programming-using-pythons-multiprocessing-module/

[32] ttps://philipwfowler.github.io/2015-01-13-oxford/intermediate/python/04-multiprocessing.html

[33] ttp://materials.jeremybejarano.com/MPIwithPython/

[34] ttp://scipy.github.io/old-wiki/pages/ParallelProgramming

[35] ttps://wiki.python.org/moin/ParallelProcessing

[36] ttp://stackoverflow.com/questions/20548628/how-to-do-parallel-programming-in-python

[37] ttps://developer.nvidia.com/how-to-cuda-python

[38] ttps://mathema.tician.de/software/pycuda/

[39] ttp://stackoverflow.com/questions/5957554/python-gpu-programming

[40] ttp://stackoverflow.com/questions/20548628/how-to-do-parallel-programming-in-python

[41] ttp://www.numpy.org/

[42] ttps://docs.scipy.org/doc/numpy-dev/user/quickstart.html

[43] ttps://pypi.python.org/pypi/numpy

[44] ttp://docs.scipy.org/doc/numpy/reference/