



Publisher homepage: www.universepg.com, ISSN: 2663-7804 (Online) & 2663-7790 (Print)

<https://doi.org/10.34104/ajeit.021.01020>

Australian Journal of Engineering and Innovative Technology

Journal homepage: www.universepg.com/journal/ajeit

Australian Journal of
Engineering and
Innovative Technology



Development and Design of a Humanoid Robot System MIRAA: Intelligent & Autonomous Mobile Assistant

Mahmuda Akter^{1*}, Mahtabur Rahman¹, Md. Rifat¹, Maruf Hasan¹, Sarmin Nahar¹, and Safik Hasan¹

¹Department of Computer Science and Engineering, Gono Bishwabidyalay, Savar, Dhaka-1344, Bangladesh.

*Correspondence: mahmudaakhi.gb@gmail.com

ABSTRACT

This study represents work in progress towards a complete system of a humanoid robot. With this aim, the system must be capable to interact with human, Give response to the question, do gestures, assist with finding an information, remember different types of thing, find internet data regarding to the user's query, find errors in its system and save a log file in order to further development and debug, autonomous servo control with the help of PID (proportional-integral-derivative) controlling. Open CV, Servo Control (Head) the system integrates with facial and object recognition to achieve its objectives based on PID quality, among other advanced capabilities derived from Open CV (camera). Details of the implementation of the currently developed system will be displayed. Towards the end, after success in the preliminary results obtained on our campus, we are encouraged to do so in order to obtain a complete phototype.

Keywords: Humanoid, Design, Servo, MyRobotLab, TTS, NLP, Autonomous, and AIML.

INTRODUCTION:

This is basically a Humanoid Robot Project and entitled as 'Development of a humanoid robot system'. This project can be categorized as 'Upper Torso Humanoid. The head goal of the project is to create a skilled and movable body using different kind of motors and make a software system that will operate the robot motors and other components of the robot and there are also development arrangements. The robot is controlled help Arduino micro-controllers those are connected to the computer using USB cable (Lentin Joseph, 2015).

Our project use MyRobotLab software for control. MyRobotLab is an open sources service based robotics framework. It is primarily written in Java, but has bindings for Python. It has web UI written in AngularJS which allows remote control (My-Robot-Lab video, 2019; MyRobotLab CookBook borg, 2019). The term humanoid comes from English word

"Human" and "oid" and the oid means something and resembles in the outer form but actually not the same. So in this sense, a humanoid is something that looks like a human but not a human. The first recorded use of the term in 1870 was to refer to the aborigines in the European colonies. In the twentieth century, the term describes fossils resembling human skeletons that are similar in morphology, but not identical (Humanoid Robot, 2019).

A robot is a machine, especially programmable by a computer that is capable of automatically performing a complex series of actions. A robot can be operated by an external control device or embedded in the control. Robots can be built on the basis of human appearance, but most robots are machines that are designed to perform any task related to their aesthetics. Robots can be autonomous or semi-autonomous and can range from humanoids to industrial robots, medical operating robots, patient support robots, dog therapy robots, co-programmed Swarm

robots, UAV drones and even microscopic Nano robots (Humanoid Robot, 2019).

A humanoid robot is a robot based on the general structure of a human, such as a robot that walks on two legs and has an upper torso, or a robot that has two arms, two legs and a head. A humanoid robot certainly doesn't look like a real person, The ASIMO humanoid robot, for example, has a helmet instead of a facial sub-section. An Android (male) or Gynoid (female) is a humanoid robot designed to look as real as possible, although these words often seem to be synonymous with humanoid. While there are many humanoid robots in fiction, some real humanoid robots have been built since the 1990s, and some real human-looking Android robots have been developed since 2002. Like robots, virtual avatars can be called humanoid when they resemble humans (Mamun *et al.*, 2020).

Review of Literature

An autonomous human-like robot that is able to adapt to changes in its environment and continues to reach its goals is considered a humanoid robot. These features set Android apart from other types of robots. There has been a lot of progress in the development of humanoid in recent years and there are still many opportunities in this field. Several research groups are trying to design and develop various platforms for humanoid based on mechanical and biological concepts. Many researchers focus on the designing of lower torso to make the Robot navigating as like as a normal human being do. Incorporating the chest, buttocks, knees, ankles and lower legs is a more complex and more challenging task. High torso design is complex but interesting work which includes arm and neck design. Walking guide analysis, optimal control of multiple motors or other actors, degree of freedom control (DOF), adaptability control, and intelligence are also challenging tasks to treat humanoids as human (Humanoid Robot, 2019).

Leonardo da Vinci drew a humanoid process in 1495, considered the first man. It was designed for sitting, for arms, for moving the head, and for closing the jaw. The 18th century can be considered as a fertile time for the development of many automations that were able to reproduce some human movements. In 1773, Pierre and Henry Louis invented the first automation that was able to write. The mechanical trumpet was made by Friedrich Kaufmann 1810. The trumpeter has a marked drum that was used to activate some valves that helped air flow in twelve languages.

The period of construction and development of the humanoid began in the 19th century when John Brainard discovered the steam man in when 1865. It was moved by the steam engine and used to pull the cart. The Electric Man was created by Frank Red Jr. in 1885, which was a more or less electronic version of the Steam Man. A prototype soldier named Boilerplate was built in 1893 by Dr. Achibald Campion. An evolutionary number of humanoid systems appeared in the 20th century. At the turn of the century, the Westinghouse Society developed a humanoid robot called LEEACTRO in 1938, which was able to move, talk and smoke.

From the 1960s to the 1990s, a wide variety of legged robot platforms began to appear in the United States, Russia, France, and especially Japan. A great job was done with the jumping robot at the Massachusetts Institute of Technology (MIT) in the 1980s. The bipod planar, Spring Flamingo, Spring Turkey, Uniru and the 3D bipad were built at MIT, which excels in walking in a dynamic and stable way (Akhtaruzzaman and Shafie, 2010).

METHODOLOGY:

1. Hardware

In this study project we have used various types of hardware components for different purposes. In the box below:

Component Name	Model/Min. Specification	Quantity
Computer	Minimum Core i5, 4 GB Ram, 20GB HDD.	1
Arduino	AT MEGA 2560	2
Servo	MG 996R (Tower Pro DIGI HI Torque TowerPro)	10
Servo	TowerPro MG90s – Micro Metal Servo	10
Screw Allen countersunk	M3x20MM, M4x20MM, M8x100MM	20,20,10
Camera	Min 1080p, 4MP	1
Speaker	Audio cable: 3.5mm Separation: ≥45dB SNR: ≥65dB	1 Pair

Microphone	360° coverage	1
Ball Bearings	Bore Size: 10- 80 mm, Shape: Round, Diameter:6 - 200 mm	3
LED Dot Matrix	8*8 (Single color/RGB)	1
Scuba Swivel Adaptor	Min 180 Degree.	3
Acrylic Sheet	Min 1'x1'	1
Casing/ Outer	Plastic/PVC Sheet Min 4mx4m, Diameter: 9mm	1
Power Supply	5v, 10-12A	1

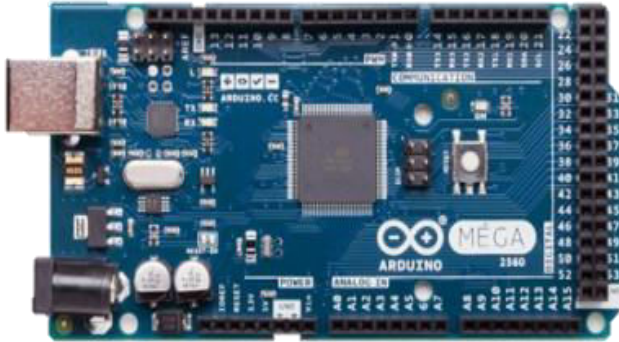


Fig 1: Arduino AT MEGA 2560

Arduino

Arduino is an open-source hardware and software company, project and user community that designs

and manufactures single-board microcontrollers and micro-control kits for building digital devices. Its products are licensed under the GNU Laser General Public License (LGPL) or the GNU General Public License (GPL), which allows the production and distribution of Arduino Board software by anyone.

I used as medium of communication from the computer in the robot to different servo. Arduino flashed first with the 'MRLcom' code that can be found in MRL's (My Robot Lab) resource folder. After Flashing with the code, now the arduino is ready to communicate with the computer using a USB port (MRLComm to the Arduino, 2019). Use Arduino AT MEGA 2560 is given below:

Table 1: Specification of the Arduino AT MEGA 2560

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37

Servo Motor

A servomotor is a rotating activator or linear activator that allows precise control of angular or linear position, speed and acceleration. It has suitable motors for sensor or position response. This requires a relatively sophisticated controller, a dedicated module designed specifically for use with most servomotors. Servomotor not a specific class of motors, basically the term servomotor is often used to refer to a motor for a closed loop control system. Servomotors are used in applications such as robo-

tics, CNC machinery or automotive manufacturing. Servo-motors are commonly used as a high-efficiency alternative to stepper motors. Steeper motors have some innate ability to control position, as they have internal output steps. This allows the encoder to be used as an open-loop position control without any feedback, as their drive signal determines the number of movement steps. However, this requires the controller to 'know' the location of the stepper motor.



Fig 2: A servo motor.

So, after the first power-up, the controller needs to activate the stepper motor and turn it to a familiar position, such as, e.g. until it activates the limit

Table 2: Features and Specification of the MG996R

Modulation:	Analog
Torque:	4.8V: 30.60 oz-in (2.20 kg-cm) ,6.0V: 34.70 oz-in (2.50 kg-cm)
Speed:	4.8V: 0.11 sec/60° ,6.0V: 0.10 sec/60°
Weight:	0.49 oz (14.0 g)
Dimensions:	Length:0.91 in (23.1 mm) ,Width:0.48 in (12.2 mm) ,Height:1.14 in (29.0 mm)
Gear Type:	Metal
Rotation/Support:	Dual Bearings
Pulse Cycle:	20 ms

Table 3: Features and Specification of the MG90s Micro Servo

Modulation:	Digital
Torque:	4.8V: 130.54 oz-in (9.40 kg-cm), 6.0V: 152.76 oz-in (11.00 kg-cm)
Speed:	4.8V: 0.19 sec/60° , 6.0V: 0.15 sec/60°
Weight:	1.94 oz (55.0 g)
Dimensions:	Length:1.60 in (40.7 mm) ,Width:0.78 in (19.7 mm) , Height:1.69 in (42.9 mm)
Gear Type:	Metal
Rotation/Support:	Dual Bearings
Pulse Cycle:	1 ms

LED Dot Matrix

A LED dot-matrix is an electronic digital display device that displays information from machines, clocks and watches, public transport departure indicators, and many other devices needed to display a simple alphabet (and / or graphic) of limited resolution. The display has an LED dot matrix equipped with a rectangular configuration (other sizes are possible, though not common) For example, text or graphics may be displayed by turning on or off the selected lights.

The dot matrix converts the instructions of the controller processor into signals, which turns on or off the index elements of the matrix to produce the required display. This project we will use LED 8x8 dot matrix which includes a total of 64 LED (Cheng, 2016).

switch. This can be noticed when switching to an inkjet printer; the controller ink will move the jet carrier to the extreme left and right to establish the end positions. Regardless of the initial position of a servomotor power-up, the angle at which the controller points it will be turned on immediately (Akhtaruzzaman and Shafie, 2010). We were used kind of 2 servo motors this project (MG-996R DIGI HI TORQUE & MG90s Micro Servo), MG-996R in Head X, Head Y, Omoplate, Shoulder, Arm, Wrist and MG90s in the Thumb, Index Finder, Middle Finger, Ring & Pinky Finger. Use MG996R and MG90s Micro Servo is given below:

Table 4: General Features of the MG90s Micro Servo

Operating Voltage:	DC 4.7V – 5.3V
Typical Voltage:	5V
Operating Current:	320mA
Max Operating Current:	2A

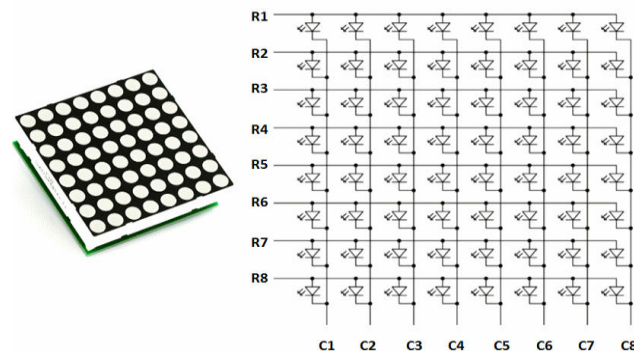


Fig 3: A 8x8 LED dot matrix display.

Other Hardware components

Among the hardware components I have discussed above there are some more hardware components we need as well.

- I have used a Power supply for providing the power and maintain voltage and current flow to the motors via negative side of the power supply attach to the Arduino on the Ground port and the positive side of the power supply attached to the 5V port on the Arduino board.
- I have used a 1 LED 8x8 Dot matrix's to imitate the eye of a human; I have micro-controller programs for different eye positions

and different animations to give the robot a more natural human character.

- I used a camera so that the robot could see, it is a 2 mega pixel wide angle camera that allows the Open CV program to perceive and see the environment.
- I used an external 360 degree microphone so like that robot could hear more accurately.
- I have used plastic, PVC, metal, acrylic for the external appearance of the body.

2. Software: Different software's are used in this study. The list of required software is below:

Table 5: List of required software

Component Name	Minimum Required Version
Operating System	Windows 10 64 bit
Framework	MyRobotLab V 1.0.2693 Manticore
Runtime Environment	Java Runtime Environment 8+ 64 bit
Programming Language	Python V2.7
Computer Vision	Open CV Latest Update (For Dev. purpose)
IDE	Sublime Text
Voice	VE American Noelle

Myrobotlab - Myrobotlab is an open source Java service based framework for robotics and creative machine control. Myrobotlab runs on the Java Runtime Environment 8 (or above) JVM, so any computer or device which supports this JVM conceptually could run Myrobotlab. Many of MyRobotLab Services will run on Android's Dalvik JVM. The services myrobotlab include machine vision from Javacv/Opencv, speech recognition from Sphinx 4, text to speech from Free TTS, Motor control, Servo control, GUI control and micro-controller communication. Third party services are wrapped in Myrobotlab Service wrappers and add to the framework (MyRobotLab-Video, 2019; My-RobotLab CookBook borg, 2019).

Starting MyRobotLab - It's pretty easy to start myrobotlab, just matter of following some specific things.

- Install Java 8 or above for operating system (64bit version if your system is 64).
- Download MyRobotLab in its latest build
- We can use the GUI to change message routing and some parameters of configuration or you can use the Zaihton service to create (Python-) scripts.
- The Jython environment has access to nearly all functionality of MyRobotLab.

Required services of Myrobotlab - This project I have used several services from myrobotlab to run the robot. The types of required services for this project are given below:

Arduino Service - The Arduino service is uses by communicate and control the very popular Arduino micro-controller. MyRobotLab runs on a computer and controls the Arduino hardware through a serial port.Arduino sketch called MRLComm which loaded by Arduino. The latest can be available in this last page. This sketch allows commands from MyRobotLab will process on Arduino hardware, it also allows the Arduino is capability to send data back to MyRobotLab (MRL) (MRLComm to the Arduino, 2019).

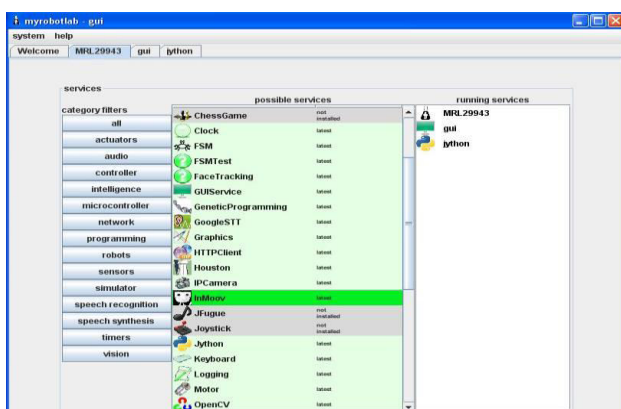


Fig 4: Desktop interface of myrobotlab.

Once MRL is connected by Arduino, it allows all other services to interact with Arduino.

- MRL can use Arduino is simple type of oscilloscope.
- MRL can use Python to effectively program the Arduino, turn on off or read pin values; interact with sensors, control servos, etc.
- MRL can control and communicate with many Arduinos, here not limit to it.
- MRL is used to communicate between Arduinos.

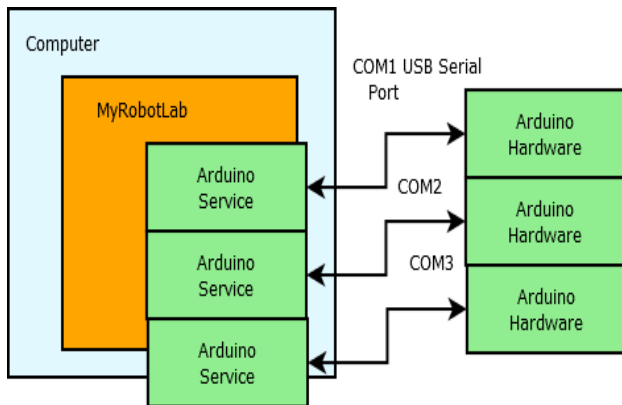


Fig 5: Arduino service diagram.

With MRL's WebGUI service you can very quickly and easily setup a web interface for an Arduino.

AudioFile Service - This service can play different types of audio files. I use this service to play feedback output audio files from robots.

CLI service - CLI; command line interface service allows you to interact directly with MyRobotLab (MRL) when it's started from an operating system shell (MRLComm to the Arduino, 2019).

For example; `/runtime/start/python/Python` will start a service named python of type Python.

ClockService - The watch can generate regular messages to engage other services.

HTML Filter Service - Htmlfilter is a service used by either strip out text or wrap the input text in html tags. For example, it uses the program to filter HTML code from AB before sending text to the speech synthesis service.

Local Speech Service - This text to speech synthesis service use local operating system text to speech engine. I have used this service for act as the voice of the robot and also installed Custom teen female voice from VE.

Log Service - Log is a helpful diagnostic service that knows how to connect itself to the user interface. It can display messages from other Services. It can also detect different types of errors and save the log in a text file so that developer can run further Bug fix and development process.

Open Weather Map Service - This service helps gather weathers current data as well as the prediction for MRL. This service I created an API token from the website for this project. API key will be found from this site: <https://home.openweathermap.org/>

PID Service - A PID service allows input-related tracking and output conversion. Tracking services are currently used as tracking strategies. Input is sent to PID, a "compute" method is sent and appropriate output is sent to a servo (PID Controller, 2019).

PIR Service - A passive infrared sensor (PIR sensor) is an electronic sensor that radiates infrared (IR) light from objects in its field of view. PIR-based speeds are often used in detectors. Arduino schema (We can also use another compatible controller).

Program AB Service - AIML is known as Artificial Intelligence Markup Language. Dr. Richard Wallace it's created by to get the ALIAS bot. AIML is a repetitive language that allows breaking down natural language text input to match the feedback that the chatbot can send.

Servo Service - The Servo service is use to control servos through a micro-controller such that an Arduino or a Raspberry pi (Servo Controller). The service will allow control of device attachment or disconnection, control his position and the speed at which it move to that position, and turn off/on the servo.

servo.attach (controller, pin, pos, velocity): Primary servo device attachment with microcontroller

servo.detach (controller): detach the servo from the microcontroller

servo.enable (): will turn the servo motor on

servo.disable (): will turn the servo motor off

servo.moveTo (pos): ask the servo to move to a position

servo.moveToBlocking (pos): Ask the servo to go to a location and wait for the location to arrive

servo.map (inputMin, inputMax, outputMin, outputMax): default to (0,180, 0, 180). Allow range

maps in any valid output range to be understood by servo. The values of outputmin and outputmax must be into 0 and 180. Modular Block Diagram showing the differences of sweeping a Servo from MyRobotLab versus sweeping from the micro-controller (Arduino) (MRLCommtothe Arduino, 2019).

WebkitSpeechRecognition Service

WebkitSpeechRecognition - uses the speech recognition that is built into the chrome web browser. This service requires the webgui to be running on chrome web browser. We used this service for speech recognition.

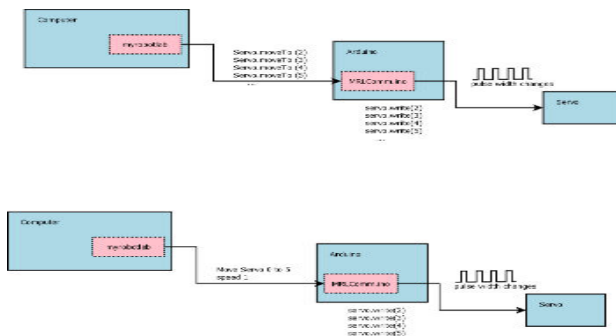


Fig 6: Modular diagram for different sweeping of a servo.

WikiDataFeatcher Service

This service grabs data from wikis website (for now wikidata). Wikidata store data by entities with an ID. For example, Adam Sandler has the ID Q132952. Each entity contains several elements: To use wikiDataFetcher, we need an ID for the feature, but not for the label

Natural Language Processing

Natural language processing, usually abbreviated as NLP, A branch of artificial intelligence that uses computers and natural language to interact with humans. The ultimate goal of the NLP is to read, understand, comprehend and comprehend human languages in a valuable way (Ashish Singh Bhatia, 2018). By processing natural language a common conversation between humans and machines can go as follow:

- a) A man speaks to the instrument.
- b) The device captures audio.
- c) Text conversion occurs from audio.
- d) Text data processing.
- e) Data takes up space to convert audio.
- f) Human respond when the machine plays the audio file.

Use of Natural Language Processing

Natural Language Processing is the driving the following common applications:

- Language translation application as Google Translate.
- Microsoft Word and word processors such as Grammar employ NLPs to test the grammatical accuracy of text.
- Interactive Voice Response (IVR) applications are used in call centers to respond to specific user requests.
- Personal assistant applications like as OK Google, Siri, Cortana, and Alexa.

Model of NLP project

The basic model of NLP we have use this project can be divided into 3 parts, here we have a figure for the illustration (Mike Barlow, 2016).

STT

We have used a Service from the framework called “WebkitSpeechRecognition” for Speech to Text conversion, this service requires the webgui to be running on chrome web browser. This service actually developed by Google and the service often called Web Speech (Mike Barlow, 2016).

AIML, or Artificial Intelligence Markup Language, is an XML dialect for creating natural language software agents. The XML dialect, called AIML, was developed between 1995 and 2002 by Richard Wallace and the global free software community. Initially, Eliza was referred to as "ALICEE." AIML formed the basis for naming ("Artificial Linguistic Internet Computer Entity"), which won the annual Labna Award for Artificial Intelligence three times and was also the Chatterbox Challenge Champion in 2004; because A.L.I.C.E.

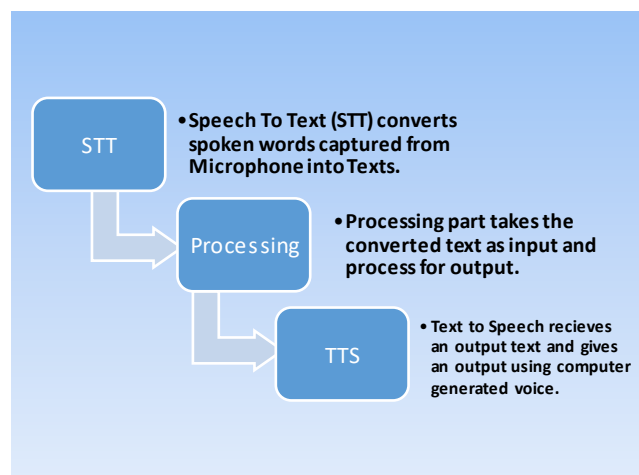


Fig 7: Block model of NLP.

Elements of a Wikidata statement

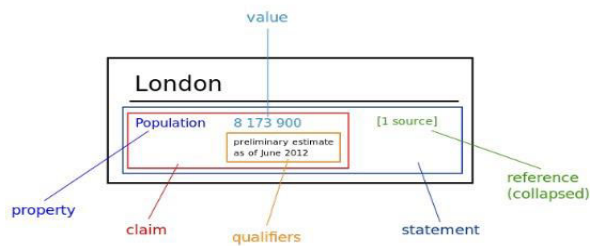


Fig 8: Elements of wikidata statement.

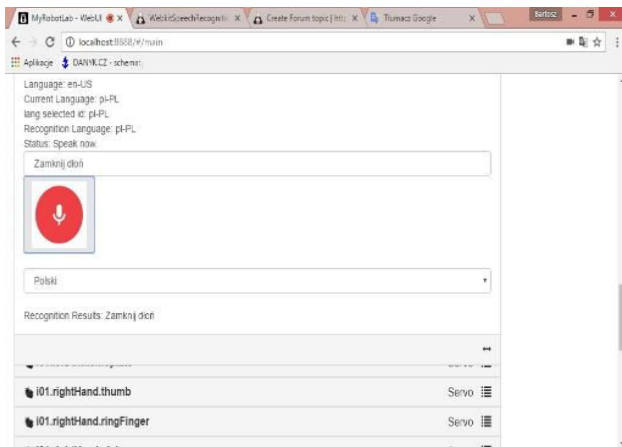


Fig 9: An image of webkit speech recognition web interface.

AIML

The ANML set was published under the GNU GPL, and most AIML interpreters are offered under a free or open source license, many "Alicebot clones" have been created based on the original implementation of the program and its AIML knowledge. Free AIML sets have been developed and made available by the user community in several languages. AIML Interpreter in Java, Ruby, Python, C ++, C #, Pascal and other languages (Lentin Joseph, 2015; Shahriare Satu, 2015).

Elements of AIML

AIML contains a variety of elements. The most important of these are described in more detail below:

Categories - In AIML the departments form the basic unit of knowledge. A section contains at least two more elements: pattern and template elements.

Patterns - A pattern is a string of characters that matches the inputs of one or more users.

Templates - A template specifies the response to a matched pattern. A template can be as simple as some literal text. Template elements include basic text formatting, conditional (if-then / otherwise) and random feedback. They can also redirect to other patterns using an element called SRAI (Symbolic Reduction in Artificial Intelligence). This can be used to implement synonymy.

Processing - In the processing section we used a service called "Program AB" in process data from STT. STT Sends the data to Program AB service, Program AB contains a variety of pre added AIML files, including various sections, template, pattern, sets, etc. The program then matches the data by prioritizing the AIML files compiled by AB. Finally it gives sends the output to the TTS to be produced as a Voice.

TTS - For Text to Speech (TTS) we have used Local Speech Service, This text to speech synthesis service use local operating system text to speech engine. Speech synthesis is an artificial production of human. A computer system used for this purpose is called a speech computer or speech synthesizer, and can be applied to software or hardware products. A text-to-speech (TTS) system converts ordinary language text into sentences; other systems represent symbolic linguistics, such as phonetic transcriptions in speech. In our project we have used "American Noelle VE" Voice of a teen aged girl (Mike Barlow, 2016). A diagram of typical Text to Speech synthesis system is given below:

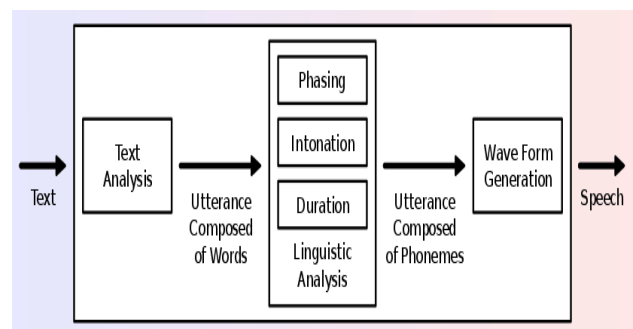


Fig 10: A typical TTS system.

Servo Control

A servomotor is a rotary actuator or linear actuator that allows precise control of angular or linear position, speed and acceleration. It consists of a suitable motor attached to a sensor for position response. It requires a relatively sophisticated controller, a dedicated module designed specifically for

use with almost all servomotors. To control the servo we add two service calls from the framework, "Arduino" and "Servo". The Arduino service has a servo motor connected to the Arduino. And the servo service is for sending signals to the servo using Arduino board. The figure below shows how the servo will be connected.

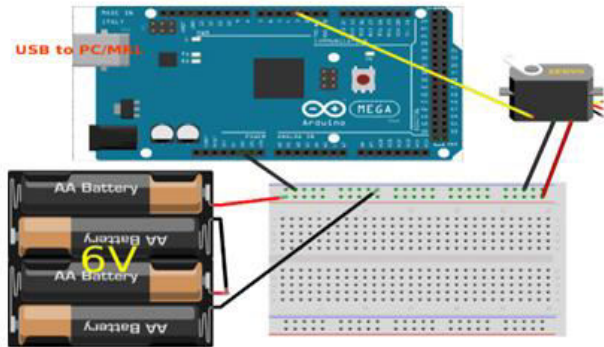


Fig 11: Servo motor connection.

Data: Some Coding used of this study.

<pre> 1. # ##### # ##### # ##### # ##### 2. # MRL SERVICE CALL 3. # ##### # ##### 4. 5. i01.ear=Runtime.createAndStart("i01.ear", EarEngine) 6. i01.startEar() 7. ear = i01.ear 8. ear.setAutoListen(setAutoListen) 9. if EarEngine=="WebkitSpeechRecognition":ear.setContinuous(setContinuous) 10. 11. if not isWebGuiActivated and EarEngine=="WebkitSpeechRecognition": 12. # start the browsers and show the WebkitSpeechRecognition service named i01.ear 13. webgui = Runtime.create("webgui","WebGui") 14. webgui.setAutoStartBrowser(False) 15. webgui.startService() isWebGuiActivated=True </pre>	<pre> <?xml version = "1.0" encoding = "UTF-8"?> 16. <aiml version = "1.0.1" encoding = "UTF-8"?> 17. 18. <category> 19. <pattern> WHO IS Isaac NEWTON </pattern> 20. <template>Isaac Newton was a English physicist and mathematician.</template> 21. </category> 22. 23. <category> 24. <pattern>DO YOU KNOW WHO * IS</pattern> 25. <template> 26. <srai>WHO IS <star/></srai> 27. </template> 28. </category> 29. </aiml> </pre>
--	---

Different Components used in this project:

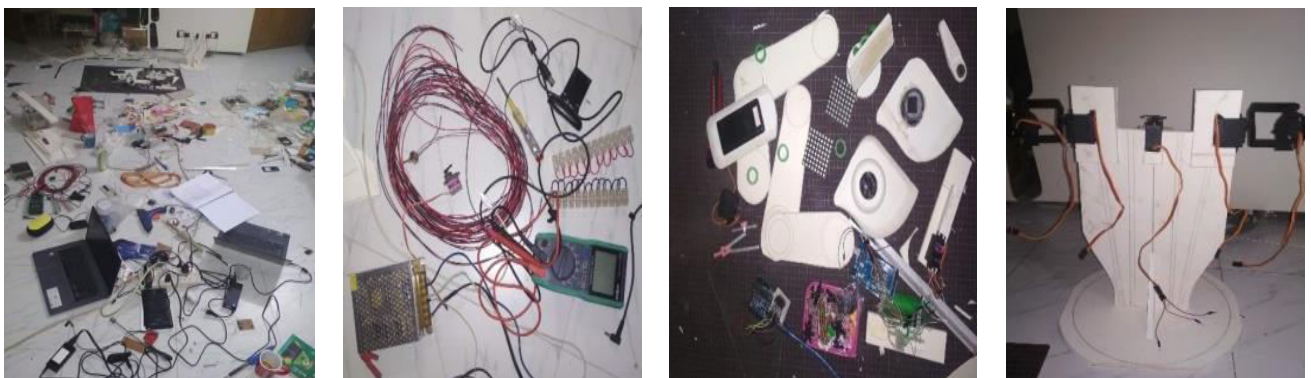


Fig 12: Instruments to design of the Humanoid Robot system.

RESULTS AND DISCUSSION:

Performance Testing

Two types of tests are performed,

1. Hardware Testing
2. Software Testing

Hardware testing process can be divided into two type, Individual hardware testing and combined hardware testing. Individual testing includes individual tests of other parts of the robot, such as head test, mouth test, ear test, hand test, fingers test etc. and Combines testing includes all the parts testing in a body as well as program code to run those components.



Fig 13: Structure of the Humanoid Robot (MIRAA).

This project was tested in all forms of hardware testing and we also have tested all the hardware in different Voltage, Ampere and Register value. And in software testing, this project was tested by giving different servo results the code to configure and calibrate the robot.

CONCLUSION AND FUTURE DIRECTION:

It's an humanoid project where we have applied a variety of abilities such as listening, speaking back, asking questions, remembering different things, learning from questions, taking internet data and talking back, moving different parts of body, making different gestures, randomly. Move the body, blink of an eye, etc. Since this is a humanoid project, so we have applied some basic capabilities to the robot.

There are many more things to like in this project and we can add these things to our future projects. They are:

- Our project is currently in the form of the upper torso (upper body), next we can create a full body humanoid system in the lower part.
- We can also make this robot walk after completing the lower part of the body.
- We can also add telepresence and teleportation capabilities. (Teleportation- to move the body

of the robot remotely, Telepresence- a human to see and interact with others without being presence in physically, robot will do the job).

- We can also add the capability to understand, Process and speak in Bengali.
- We can make an interactive UI for the robot so that user can control, configure and give input to the robot.
- Voice can be developed to express more emotions and add some additional capabilities like Pitch and Note Shifting, Delay flexibility to express emotions, Sing, Mimicry in different voice.
- Remote control using authentication.

ACKNOWLEDGEMENT:

Thanks to the supervisor and Head of the department supported with proper assistance and help for analysis and writing to conduct successful research study. Special thanks to Vice-Chancellor of Gono Bishwabidyalay, Professor Dr. Laila Parveen Banu to help us Robot MIRAA exhibition.

CONFLICTS OF INTEREST:

The authors declare that they have no conflicts of interest in this research to publish it.

REFERENCES:

- 1) Adam C. Harris, James M. Conrad, (2014). "Utilizing middleware to interface with the simulation environment for autonomous robots" in *IEEE SOUTHEASTCON 2014*, Lexington, KY, USA.
- 2) Akhtaruzzaman, Md., and Shafie, A. A., (2010). "Evolution of Humanoid Robot and Contribution of Various Countries in Advancing the Research and Development of the Platform" in *international conference on control, automation and systems*, 27-30, in KINTEX, Gyeonggi-do, Korea, Pp. 1021-1028. <https://doi.org/10.1109/ICCAS.2010.5669646>
- 3) Ashish Singh Bhatia, Richard M. Reese, (2018). "Natural Language Processing with Java - Second Edition" Explore various approaches to organize and extract useful text from unstructured data using Java. Publisher: Packt Publishing, ISBN: 9781788993494. <https://www.packtpub.com/product/natural-language-processing-with-java-second-edition/9781788993494>
- 4) Cheng, H., & Ji, G. (2016). "Design and implementation of a low cost 3D printed humanoid robotic platform." in *the 6th annual IEEE international conference on cyber technology in automation, control and intelligent systems*, June 19-22, 2016, Chengdu, China.
- 5) Humanoid Robot, available on the online - https://en.wikipedia.org/wiki/Humanoid_robot [Last Access on February 6, 2019]
- 6) InMoov Provides Everything from Prosthetics to Cocktails Around the World, <https://makezine.com/2015/06/23/inmoov-around-the-world/> [Last Access on March 6, 2019]
- 7) Lentin Joseph "Learning Robotics Using Python" Design, simulate, program, and prototype an interactive autonomous mobile robot from scratch with the help of Python, ROS, and Open-CV. Publisher: Packt Publishing, ISBN: 9781783287536.
- 8) Making InMoov Move, available on the - <https://www.instructables.com/id/Making-InMoov-Move/> [Last Access on May 1, 2019]
- 9) Mamun AMA, Islam T, Siam MMS, and Kabir ME. (2020). Development of smart librarian with the virtual assistant (PRIMO), *Aust. J. Eng. Innov. Technol.*, 2(4), 54-65. <https://doi.org/10.34104/ajeit.020.054065>
- 10) Mike Barlow, and Jon Bruner, "What Are Conversational Bots?" Publisher: O'Reilly Media Inc., ISBN: 9781492042808.
- 11) MyRobotLab - Video Tutorial #1 (Overview), <https://www.robotshop.com/community/forum/t/myrobotlab-video-tutorial-1-overview/13048> [Last Access on May 9, 2019]
- 12) MyRobotLab Cookbook borg, <https://github.com/MyRobotLab/myrobotlab/wiki> [Last Access on May 9, 2019]
- 13) Shahriare Satu Md., Md. Hasnat Parvez, Shamim-Al-Mamun, (2015). "Review of integrated applications with AIML based chatbot" in *international conference on computer and information engineering (ICCIE)*, Rajshahi, Bangladesh.
- 14) The Working Principle of a PID Controller for Beginners, <https://www.elprocus.com/the-working-of-a-pid-controller/> [Last Access on March 8, 2019]
- 15) Uploading MRLComm to the Arduino, <http://myrobotlab.org/content/uploading-mrlcomm-arduino-0> [Last Access on April 29, 2019]

Citation: Akter M, Rahman M, Rifat M, Hasan M, Nahar S, and Hasan S. (2021). Development and design of a humanoid Robot system MIRAA: intelligent & autonomous mobile assistant, *Aust. J. Eng. Innov. Technol.*, 3(1), 10-20. <https://doi.org/10.34104/ajeit.021.01020> 