

Implementing Physical Capabilities for an Existing Chatbot by Using a Repurposed Animatronic to Synchronize Motor Positioning with Speech

Alex Johnson, Tyler Roush, Mitchell Fulton, Anthony Reese

Kent State University

*Computer Science Department,
OH, United States*

Abstract— A chatbot is a type of computer program which contains the functionality for conducting conversations through auditory or textual means. These programs have recently shifted from the use of rule-based logic to more advanced use of artificial intelligence for carrying conversations [1]. Our project examined implementing physical capabilities for an existing chatbot by using a repurposed animatronic controlled by an Arduino micro controller. This project sought to synchronize motor positioning including eye and jaw movement with speech output from text provided through intelligent web APIs.

Keywords— Chatbot, Robotics, Animatronic, Arduino, Speech Synthesis, Cleverbot

I. INTRODUCTION

The Human Robot Interaction (HRI) is one of the areas in computer science which has many recent interests and applications, especially in social robotic [7], [8]. In this area recognition, emotion, especially from facial expression play the main role [12], [14]. Using some new methods, even make the accuracy and performance, better [9], [10] and [11].

There are many methods which can be used for finding better accuracy and for learning part [15], [16], [17] and [18]. Even we can use some traits, for example, facial features data to make better interaction with human [18], [19] and [20]. Using the physical components and chassis of a Wowwee Robotics' Alive Elvis animatronic, along with

several web APIs and .NET libraries, our group created a robot capable of holding a conversation with a human user through verbal and physical communication. The provided servo motors of the animatronic were attached to an Arduino microcontroller and reprogrammed with compiled code written in C++ running on the Arduino as well as an external program in C#. From an attached computer, an external program runs which handles speech recognition, taking audio input from a user and using the result to call multiple web APIs such as Cleverbot and Watson to determine an intelligent response. Commands for precise motor movement based on the response are then issued to the Arduino through serial ports which are synchronized with speech output. While the end result of this project is still a prototype, our robot handles the basic attributes of imitating human conversation.

In this article, in the next session we will explain the Algorithm and codes and circuits in details. In section 3 the flowchart is presented. Section 4 describes the Client code and the last section conclude the paper.

II. ARDUINO CODE

While the programming for processing, speech is handled externally, motor control is handled directly by the Arduino. The Arduino can interface with external devices through its integrated serial port which using a set baud rate, sends data over the channel a single bit at a time. Bytecodes are sent from the external C# program to the Arduino which are then interpreted into commands specific commands to the servomotors. Sending simple bytecodes that are resolved into commands allows for exponentially faster serial communication as opposed to sending full commands.

The Arduino is wired to several servo motors though with a larger board and shield more motors in the robot's head are accessible. A servo motor is a type of actuator that allows for precise control when positioning itself. It contains a coupled sensor that allows for position feedback which can be interpreted by its controller [6]. Understandably, servo motors are extremely useful for use in robotics. As the motors in this project are used to mimic facial features, the degree range in certain parts such as the jaw features a relatively low degree range. The amount of power that is applied to the motor depends on both the distance and speed at which it needs to travel.

Table1. The wire connectivity for each part of facial function

Function	Potentiometer Wires			Motor Wires		Elvis Connector Color	
	Red	Yellow	Green	Keyhole	Rectangle	Potentiometer	Motor
Jaw	Black	White	Red	Gray	Brown	White	Gray
Eyebrows	Black	Purple	Red	Blue	Brown	Purple	Blue
Neck L/R	Red	Brown	Black	White	Brown	Brown	White
Eyes Up/Down	Black	Orange	Red	Red	Brown	Orange	Red
Eyes L/R	Black	Yellow	Red	Brown	Orange	Red	Orange
Eyelid - Left	Black	Green	Red	Yellow	Brown	Green	Yellow
Eyelid - Right	Black	Blue	Red	Brown	Green	Blue	Green
Lip	Black	Gray	Red	Purple	Brown	Gray	Purple

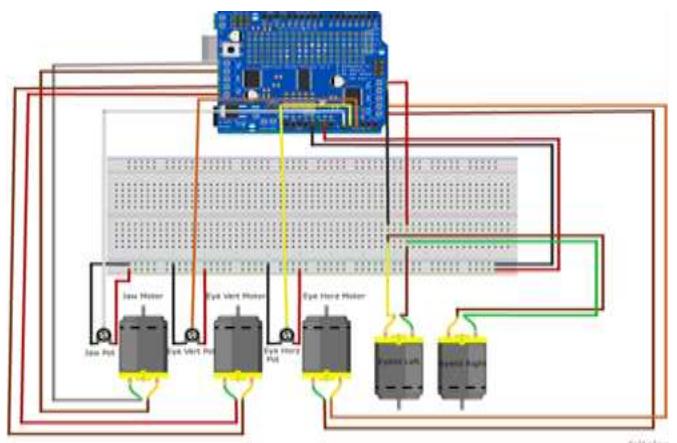


Fig.1. The motor and Arduino connectivity

III. Flowchart

In figure 2 you can find the flowchart of the algorithm.

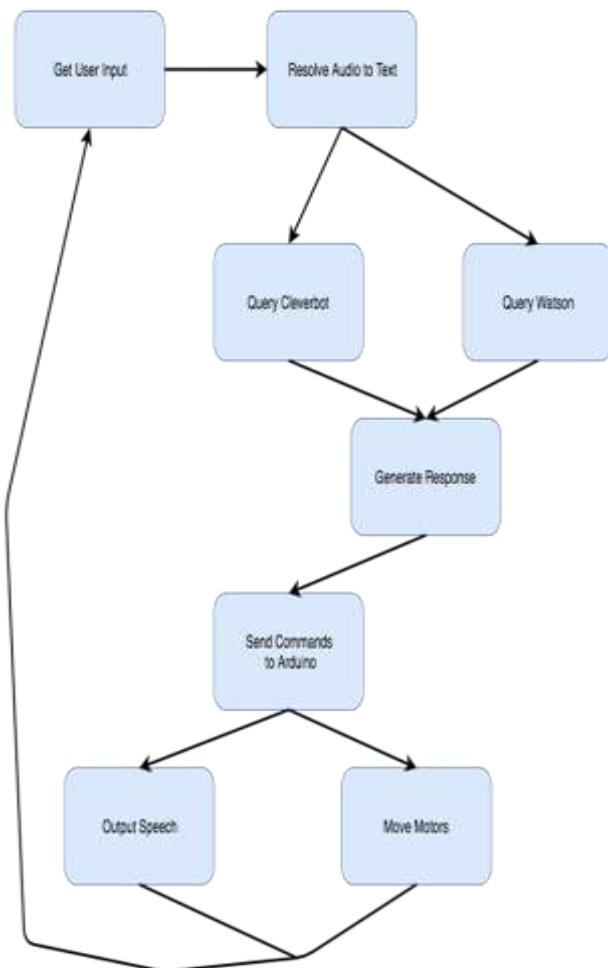


Fig. 2. The flowchart of algorithm

At the first, the system gets the data from the user then convert the audio to text and send the query to the Watson and clevbot. Then the Watson and clevbot generate the response the comment related to these responses will send to the Arduino. The Arduino generates the proper speech and move the motor for moving face part.

IV. CLIENT PROGRAM

On the external program which runs on either a connected laptop or single board computer, such as a Raspberry Pi, a loop run which controls the conversation between the human and robot. The loop begins by using functions from a speech

recognition library with a detected audio input device to translate words to text input which can be more easily processed. What the user said is then resolved from audio to text using some of C#'s libraries. This text is then used for several concurrent network calls one of which is the API for Cleverbot. The result that is then passed to the cloud based Cleverbot API occurs in a background worker thread which determines an appropriate response.

A. Cleverbot

Cleverbot is a web application which uses artificial intelligence to converse with humans based on past interactions stored in an enormous database. Unlike most chatbots, it learns from human input in past conversations and typically hosts over 80,000 conversations at any given time [2]. Our program sends a string to the api which then processes it and returns a response. From our program's perspective, using Cleverbot is quite simple. All that is needed is an API user account and key.

B. Watson

Concurrently in another thread, a call to the Watson API is made which will attempt to perform tone analysis on the text to determine the suggested mood, which will sequentially help determine the robotics nonverbal response. Watson is a computer system of IBM that is capable of answering questions with natural language. One specific functionality offered is the ability to determine the mood and tone in a conversation [3]. This function is referred to as the Watson Tone Analyzer Service and it is capable of detecting three types of tones from given text which are emotions, social tendencies and writing style. The emotions and social tendencies stem from the Big Five personality traits that are commonly categorized by many psychologists. Although using this API is not entirely free, a developer API key allows a

programmer to demo the functionality which for our project's case is enough.

Table2. The motor specifications

	Byte Codes Examples			
	Part	Speed	X Pos	Y Pos
// Open the jaw halfway at fastest speed	0	255	0	125
// Slowly close the jaw	0	55	0	0
// Move eyes to look at coord (100, 150)	1	255	100	150
// Blink eyes	2	-	-	-

Once the results have returned from the web, a suitable response is determined and a bytecode is send over serial to the Arduino. In the event of networking or server side issues with the APIs, generic values will be passed instead to continue the conversation. As shown on the graph, bytecodes for this project consist of 4 bytes: one for the part, one for the speed, and two of the desired coordinates. By sending simple bytecodes to the Arduino as opposed to full strings of commands, much time is saved when reading the input.

After the Serial.readBytes() function has run, which reads the characters from the serial port into a buffer, the analogRead() function takes only around 100 microseconds to complete a read from this buffer and thus is capable of reaching a maximum rate of nearly 10,000 times in a given second [4]. Though this project only makes use of several motors, the approach used in the code could be easily scaled to work with an exponential amount of motors.

C. .NET Libraries

Our external or client program which runs on a laptop or Raspberry Pi running Windows separate from the Arduino exists as a Windows GUI application or a Windows Form (Winforms) application as it is also known as. Windows Forms is an event-driven GUI class library that is included as part of Microsoft's .NET Framework. This project makes additional use of several of the built in libraries for C# on the client program including Speech Recognition and Speech Synthesis. This allows the program to communicate with the user in real time as the program interprets speech to text and later outputs text. The user interface that is provided allows the user to open and close serial communications between the client device and Arduino while additionally offering the ability to manually command motors and start the API routines.

D. Syllable Count

Another key element of the program is a function which calculates the amount of syllables in a given word. This is required to properly synchronize the speech output with the motors. Depending on the word length and the number of syllables, different commands are sent to the motors to adjust accordingly. To determine the syllables of a word, many times simply counting the vowels will provide an accurate answer. But cases such as silent ending and double vowels need to be accounted for. This program does this for each word in a given sentence while mapping the output to mouth movements. Shown below is some of the code for sending mouth movement bytecodes to the Arduino from the Windows Form application while synchronizing with the speech synthesizer speech output.

V. CONCLUSIONS

This project involved several different aspects that proved difficult during implementation. As our group's robot focused on human interaction,

efficiency in programming was necessary as quick responses were needed to maintain a conversation. Between the different API calls, concurrency of tasks was required as remote calls over the internet acted as a performance bottleneck. Another challenge we faced was with controlling the chassis' many servo motors. Servo motors are closed-loop servomechanisms that are controlled by an electric signal which will determine the position to. Servo motors typically can only turn a total 180 degrees, 90 in each direction. As the bytecode send to the Arduino can only contain a value from 0 to 255, position exact movement requires scaling for the passed variable. In one instance, due to an issue with determining the value of the bytecode, the jaw of the robot became stuck attempting to open to an angle that it could not reach. Regardless of the difficulties with implementation, our group was able to successfully repurpose the animatronic into a chatbot. Although the hardware used was relatively simple, the approach taken in handling event-driven motor coordination could easily be scaled to much more complex and precise hardware.

REFERENCES

- [1] J. C. Wong, "What is a chat bot, and should I be using one?" in The Guardian, The Guardian, 2016. [Online]. Available: <https://www.theguardian.com/technology/2016/apr/06/what-is-chat-bot-kik-bot-shop-messaging-platform>.
- [2] "Creating a bot instance · Cleverbot.io," Cleverbot.io. [Online]. Available: <https://docs.cleverbot.io/docs>.
- [3] "Tone analyzer service documentation," in IBM.com. [Online]. Available: <https://www.ibm.com/watson/developercloud/doc/tone-analyzer/>.
- [4] Arduino, "Analog Read," 2016. [Online]. Available: <https://www.arduino.cc/en/Reference/AnalogRead>.
- [5] Microsoft, "How to: Create a windows forms application," 2016. [Online]. Available: [https://msdn.microsoft.com/enus/library/ms235634\(v=vs.80\).aspx](https://msdn.microsoft.com/enus/library/ms235634(v=vs.80).aspx).
- [6] K. Ross, "What's a servo: A quick tutorial," [Online]. Available: <http://www.seattlerobotics.org/guide/servos.html>.
- [7] Mehdi Ghayoumi, Arvind. K. Bansal, "Real Emotion Recognition by Detecting Symmetry Patterns with Dihedral Group", MCSI, Greece, 2016.
- [8] Mehdi Ghayoumi, and et al, "Follower Robot with an Optimized Gesture Recognition System", RSS, USA, 2016.
- [9] Mehdi Ghayoumi, Arvind. K. Bansal, "Architecture of Emotion in Robots Using Convolutional Neural Networks", RSS, USA, 2016.
- [10] Mehdi Ghayoumi, Arvind. K. Bansal, "Emotion in Robots Using Convolutional Neural Networks", ICSR2016, USA, 2016.
- [11] Mehdi Ghayoumi, Arvind. K. Bansal, "Multimodal Convolutional Neural Networks Model for Emotion in Robots", FTC, USA, 2016.
- [12] M. Ghayoumi, A. Bansal, "Unifying Geometric Features and Facial Action Units for Improved Performance of Facial Expression Analysis", CSSCC 2015.
- [13] Mehdi Ghayoumi, "A Review of Multimodal Biometric Systems Fusion Methods and Its Applications", ICIS, USA, 2015.
- [14] M. Ghayoumi, A. Bansal, "An Integrated Approach for Efficient Analysis of Facial Expressions", SIGMAP 2014.
- [15] M. Ghayoumi, "A Quick Review of Deep Learning in Facial Expression", Journal of Communication and Computer, 2017.
- [16] M. Ghayoumi, A. K. Bansal, "Emotion Analysis Using Facial Key Points and Dihedral Group", International Journal of advanced studies in Computer Science and Engineering (IJASCSE), 2017.
- [17] Mehdi Ghayoumi, Maha A. Therefore, Arvind K. Bansal, "A Formal Approach for Multimodal Integration to Derive Emotions", Journal of Visual Languages and Sentient Systems, 2016.
- [18] Timothy Zee, Mehdi Ghayoumi, "Comparative Graph Model for Facial Recognition Systems", CSCI, Vegas, USA, 2016.
- [19] Mehdi Ghayoumi, Maha Tafar, Arvind. K. Bansal, "Towards Formal Multimodal Analysis of Emotions for Affective Computing", DMS, Italy, 2016.
- [20] H. Abrishami Moghaddam and M. Ghayoumi "Facial Image Feature Extraction Using Support Vector Machines", Proc. VISAPP 2006, Setubal, Portugal.