Quantum Fritz - A Prototype Intelligent Robot Head with ChatGPT and Controlled by Quantum Automata

# 1st Monica Bao

*Dept. of ECE Portland State University*

Portland, Oregon bao.monica@outlook.com

# 2nd Mehul Rajendra Shah

*Dept. of ECE Portland State University*

Portland, Oregon mehulshah@pdx.edu

# 3rd Marek Perkowski

*Dept. of ECE Portland State University*

Portland, Oregon h8mp@pdx.edu

***Abstract*—This paper presents Quantum Fritz, an integration of an inexpensive robot head with emotional facial gestures, ChatGPT for advanced conversation and a new type of quantum controller. The controller includes a quantum finite state machine to determine the internal state of the robot, a counter to track the time progression, and a motion generator combinational circuit. Combining these components is proposed for the first time as a complex quantum finite state machine system with standard probabilistic, quantum probabilistic and deterministic behaviors.**

1. Introduction

For years a research area in humanoid robotics was the development of robot heads with emotional gestures. Another area were chatbots that were able to discuss various topics with the audience. Finally, quantum-controlled mobile robots [3, 7] and humanoid robots [1, 2, 3, 10, 14] were proposed. In this paper we present how these three areas can be combined to a single prototype robot with more advanced capabilities.

Nowadays, humanoid robots like Sophia created by Hanson Robotics and Ameca created by Engineered Arts are famous, but these prototype robots are extremely expensive. In contrast, in the last twenty years we have developed several humanoid robots and robot heads, but our goal was to design a head that is friendly, funny and inexpensive, so that our project can be reproduced by hobbyists and robot enthusiasts from high schools. Fritz is an animatronic robot head (Fig.2) which can be programmed with simple classical software [16]. Here we introduce Quantum Fritz – a robot head interfaced with the famous GPT-3.5 software and controlled by simulated quantum automata. This is an innovative project that combines the power of artificial intelligence, quantum computing and emotional robotics to create a unique and engaging user experience. The advantage of using quantum control is that it can easily allow for deterministic, classical probabilistic, and quantum probabilistic (entangled) behaviors [8,1,2,3,4,6]. Fritz is an advanced but inexpensive animatronic head with 13 servo motors that enable it to move its jaw while speaking and singing, and to demonstrate several facial expressions with amazing theatrical effects. The project utilizes GPT-3.5,

one of the most advanced natural language processing models developed by OpenAI [12], the famous recent breakthrough in Artificial Intelligence. This enables the robot head to generate intelligent responses to user input, tell stories and improvise, all combined with facial motions and emotions. The system works by capturing the user’s voice input through a microphone, which is then converted into text format by the GPT-3.5 model. The GPT-3.5 model then processes the text input and generates an appropriate response based on the input received. The response is then passed back to the Quantum Fritz software which translates it into speech and movements of the jaw and facial expressions to provide a realistic and engaging experience for the user (Fig.1). The emotional poses and gestures (gestures are sequences of poses) are created by interlinked quantum automata and next translated to standard motion software that comes with Fritz. Using quantum au- tomata [6,13] in this project is an improvement to the previous use of quantum combinational circuits [1,2,3,4].

The Fritz Robot Head and GPT-3.5 interface have several potential applications, including education, entertainment, and communication. For example, the robot head could be used to teach language skills or as a conversational partner for individuals who may struggle with social interactions, for instance autistic children. In conclusion, the Quantum Fritz robot head interfaced with GPT-3.5 is a truly innovative project that combines cutting-edge technologies to provide a unique and engaging experience to its users. Now that the first prototype is completed, we are able to observe its be- haviors and do some step-by-step improvements. This project is already an excellent example of how various ideas from quantum computing, robotics, and artificial intelligence can be combined together to create many variants of future amazing products. The main goal of this paper is to present how similar projects can be developed for various undergraduate or even high-school competitions, because we use inexpensive and commonly available robot components. Moreover, the new AI software is widely available on the Internet. Full technical details of this project and a video are available in the complete Quantum Fritz documentation [14].



Fig. 1. Interface of Quantum Fritz to GPT-3.5

1. *Arduino UNO*
2. Used Components

Fig. 2. Smiling Fritz Robot Head

The Arduino Uno is a microcontroller board, based on the ATmega328P (for Arduino UNO R3) or ATmega4809 (for Arduino UNO WIFI R2) microcontroller by Atmel and was the first USB powered board of Arduino. The Atmega328 and ATmega4809 comes with built-in bootloader, which makes it very easy to flash the board with user’s code. Like all Arduino boards, one can program the software running on the board using a language derived from C and C++. The easiest development environment is the Arduino IDE.

1. *Servo Motors*

Servo motors are controlled through pulse width modulation (PWM), which involves sending an electrical pulse of variable width via the control wire. Each pulse has a minimum and maximum width, and a repetition rate. A servo motor can turn up to 90° in either direction, giving it a total of 180° movement. The neutral position of the motor is the point where the servo has the same potential rotation in both the clockwise and counterclockwise directions. The duration of the pulse sent via the control wire determines the position of the motor’s shaft. A pulse of 1.5 milliseconds (ms) will make the motor turn to the 90° position, while shorter or longer pulses will turn the servo in the counterclockwise or clockwise direction, respectively. Servos expect to receive a pulse every 20ms, and the length of the pulse will determine how far the motor turns.

1. *Robot Kit To Assembly*

It can be purchased from [17]. It contains all the parts for Fritz robot head, Arduino UNO, Servo Shield, and battery attachments. One will also need additional servo motors, constant 6V power batteries or USB powered device, glue gun, extra sets of screws and a few jumper wires.

1. Basic Head Design

To assemble the robot head, the user needs to fol- low instructions posted by kerkits. The instruction link is provided here: https://kerkits.com/pages/fritz-support-page- assembly-software-installation-etc After completing the as- sembly, the user should interface all the servos to the Arduino and run the test to check whether all servos are working

correctly. After ensuring all the servos are working, one needs to install software provided by kerkits, in order to check whether all programmed emotions are working correctly (Fig. 3). Fritz faces a significant challenge in terms of its power consumption. The 4.5V power kit provided by Kerkits is insufficient to power the servo motors, which draw more power and frequently require replacement. To address this issue, there are a couple of solutions available. One option is to use a separate power supply that provides a constant 6V. Another solution is to use rechargeable cells to ensure a continuous power supply.

Connecting the moving parts of Fritz can be challenging due to the compact spaces involved, particularly when it comes to the connections for the eyeballs and eyelids. Losing connection during motor rotation can be a difficult issue to address, requiring the opening of joint connections just to reconnect them. Moreover, many parts are joined using a hot glue gun, which makes rebuilding connections impossible. To avoid this, an alternative is to screw the parts instead of gluing them, providing flexibility to detach all items quickly. The Arduino to PC serial communication wire presents another hardware flaw, as it obstructs the vertical motion of the neck. To address this issue, a USB to TTL converter can be used to eliminate the serial wire, allowing for greater flexibility in neck movement.

1. *Interfacing with Arduino*

To establish a connection between the computer and the Arduino, a serial communication cable is used. The Arduino Uno is equipped with a motor shield that facilitates the direct connection of servo motors to the board’s analog and digital pins. This eliminates the need for a breadboard. Additionally, the motor shield provides a power source of 5V to power the motors and other external components. The current configu- ration employs a laboratory-provided variable power adapter instead of a battery connector.

1. *Emotion Detection*

Emotion detection is a critical task for various companies to comprehend how their consumers respond to their launched

products. It has numerous other applications, such as de- termining a person’s mood from a distance using a camera for detection. Face emotion detection can be also used for mimicking human facial emotions by the robot head [15].

1. ChatGPT – Revolution in Artificial Intelligence

ChatGPT is a large language model trained by OpenAI based on the GPT-3.5 architecture. It has been designed to provide conversational responses to user queries, making it easier for people to ask questions and receive quick, infor- mative answers. It also has several other applications that can be used in entertainment for our Quantum Fritz. One of the major benefits of ChatGPT is that it is available 24/7, making it easier for users to ask questions at any time, regardless of their location. Unlike human experts, ChatGPT can provide answers to a wide range of topics attempting to have no bias, making it an ideal platform for those looking for unbiased information. Additionally, ChatGPT can provide answers to even the most difficult and complex questions, making it an invaluable tool for researchers, students, and professionals alike. Of course, ChatGPT does errors sometimes, but in Quantum Fritz’s case these errors can be only entertaining, not disturbing. Another major benefit of ChatGPT is its ability to understand natural language, which makes it easier for users to ask questions in their own words. ChatGPT can understand and respond to questions in a variety of languages, making it accessible to people from different parts of the world. Additionally, the responses provided by ChatGPT are usually more accurate and detailed than those provided by other search engines or automated systems, making it an ideal choice for those looking for more in-depth information and entertaining values. ChatGPT can also be used to supplement traditional learning methods. Students can use ChatGPT to ask questions about topics that they are studying in school, and receive quick, informative answers that can help them better understand the subject matter. Another major benefit of ChatGPT is its ability to learn from its interactions with users. As more and more people use the platform, ChatGPT becomes more intelligent and better equipped to handle a wider range of questions. This means that over time, ChatGPT will become an even more valuable resource for those looking to ask questions and receive informative answers.

*A. ChatGPT for Quantum*

It is a useful tool for robot speech because it is a large language model that has been trained on a vast amount of text data. When integrated with a robot, ChatGPT can enable the robot to interact with humans in a more natural and enter- taining way. Using ChatGPT for robot speech can also help to improve the quality and consistency of the robot’s speech. Unlike traditional pre-programmed responses, ChatGPT can generate unique responses to specific questions or prompts, making the robot’s speech more engaging and informative. Another advantage of using ChatGPT for robot speech is that it can enable the robot to learn and adapt over time. As the

robot interacts with more people and receives feedback, it can refine its responses and improve its ability to understand and communicate with humans. Observing such behaviors on an emotional robot head is impressive.

1. Implementation of speech-to-text and ChatGPT

interface using Python

* 1. *Imports and Initial Setup*

Advantage of using Python language in such projects is that there are many available libraries for interfacing, vision, robotics, Machine Learning and Quantum Computing, so that the developer does not need to create all software from scratch. Below is a short description of some libraries that we use.

**import pyfirmata:** This imports the pyfirmata library, which allows us to communicate with Arduino boards using the Firmata protocol.import time: This imports the time li- brary, which provides various time-related functions. import speechrecognition as sr: This imports the speechrecognition library and aliases it as sr. This library provides access to various speech recognition engines, including Google Speech Recognition.

**import pyttsx3:** This imports the pyttsx3 library, which pro- vides a cross-platform interface for text-to-speech conversion. import serial: This imports the serial library, which provides access to serial ports.

**import openai:** This imports the openai library, which allows us to interface with OpenAI’s GPT-3.5 API.

**import threading:** This imports the threading library, which provides support for multithreaded programming.

The code sets up the API credentials for OpenAI, a company that specializes in Artificial Intelligence. It then initializes the speech recognition and synthesis engines, which allows the program to understand and speak. The speech rate is set to 80% of the original rate. The code also opens a serial com- munication with an Arduino board, which is a microcontroller board that allows the program to send and receive data to and from sensors and other devices. Finally, the code waits for two seconds to ensure that the connection is established before proceeding with the rest of the program.

* 1. *Main Logic*

This is a code block that creates an infinite loop which prompts the user to choose between two input methods: voice or typing. If the user chooses voice, the program uses the sr.Microphone() function from the SpeechRecognition library to record and process speech input. If the user chooses to type, the program waits for input from the console. The program then checks the input to see if it matches any of the preset emotion keywords (”fear”, ”neutral”, ”sad”, ”angry”, ”surprised”, or ”happy”). If there is a match, the program sends a signal to an Arduino board connected to the computer via a serial port. The signal corresponds to a specific facial expression and the Arduino board controls a set of LEDs that illuminate the expression. If the input is ”pause”, the speech engine is stopped, and if the input is ”interrupt”, the program stops the loop and restarts from the beginning. The

loop continues until the user inputs ”exit”, at which point the program exits.

* 1. *Talking Fritz*

This part of the code takes the user’s input (textinput), sends it to OpenAI’s GPT-3.5 model for processing and generates a response. The input prompt for GPT-3.5 is set as the user’s input text. The GPT-3.5 model is specified as ”text- davinci-002”, which is a language model capable of gener- ating high- quality text in response to various prompts.The maxtokens parameter specifies the maximum number of tokens (words and punctuations) that GPT-3.5 should output in its response. In this case, it is set to 50. The response from GPT-3.5 is then retrieved and stored in the variable textoutput. The estimated speech time for the response is calculated by dividing the number of words in the response by the average reading speed (2.5 words per second) and multiplying the result by 1.25 to account for pauses and other factors. Finally, the response is printed on the console with the prefix ”GPT-3.5:”.

* 1. *Python to Arduino Communication*

After generating a response using OpenAI’s GPT-3.5, the code sends the estimated speech time to the Arduino board by writing it to the serial port using the write() function. The estimated speech time is converted to a string with one decimal place using the format() method and then encoded using the encode() method before sending it to the Arduino. Next, the code waits for 2 seconds to allow the Arduino to receive the estimated speech time before converting the response text to a voice signal using the say() method of the engine object. The runAndWait() method is then called to play back the voice signal. This will cause the response to be spoken out loud.

1. Quantum Control based on Two Interlinked Quantum State Machines Combined With

Combinational Logic

The innovative research area of Quantum Robotics was first introduced by Benioff in 1998, who described a theoretical quantum robot in a fully quantum environment of molecules [4]. In contrast, the standard robot with quantum control was first introduced as a theoretical concept by Dong et al. [7]. The physical realization of quantum-controlled robot actors (toys) was first introduced by Raghuvanshi et al. in 2007 [2, 3], where a mobile robot and robot head were built based on quantum Braitenberg vehicle principles. Our team developed small robots which used simulated quantum circuits to control a classical microprocessor, and in turn control the classical sensors and effectors of the robot as well [2,3]. Future integration of quantum computation into robots will utilize the nature of quantum mechanics to such benefits as largely increased computational power and the ability to timely solve complex problems that current classical computers aren’t able to, as seen from quantum algorithms like Grover’s Algorithm

The other advantage of using quantum controllers is the ability to easily implement probabilistic and entangled be- haviors. This paper proposes introduction of a circuit-based quantum finite state machine (QFSM)[6,13]. This has never been implemented on a physical quantum-controlled robot before, and it will provide a further step forward into the field of quantum robotics, as quantum control allows for much higher variety of motions generated in real time and very rarely repeating. Due to the quantum advantage of information car- ried by the quantum particle, it retains its state until observed or disturbed by its outside environment, unlike the classically used electronic device. This preservation of state is used for the creation of sequential quantum circuits and the quantum finite state machine [4,6,13]. Therefore the same quantum circuit (quantum program) generates different but partially constrained motions. This way, behavior of the robot can be deterministic, classical probabilistic, or quantum probabilistic (entangled). Quantum Fritz is the first humanoid head that uses simulated quantum control based on a system of quantum automata and a physical robot head.

The goal of this project is to ultimately program complex quantum behaviors for a physical robot head. To illustrate this, we will present here a system with 4 servos used to manipulate the robot’s left eyebrow, right eyebrow, left lip corner, and right lip corner. The goal of the robot is to be able to make 4 consecutive predetermined facial gestures for each different internal state the robot experiences, those being the happy, sad, and neutral internal states which are determined using the quantum finite state machine.

The robot will make a predetermined sequence of four faces for every internal state that it is in. The robot can be in either a happy, sad, or neutral state, and in each state the robot will make a series of facial expressions to emulate how a person would feel when they are happy, sad or neutral. In this case, a pose would be a single face the robot makes, and a gesture is the string of faces that the robot shows. The servos in the robot show an animation-like quality to the robot’s facial expressions because of the continuous motion from pose to pose.

1. Internal State Transition Logic

The robot will always start off in a happy state, from there, there will be a 50% chance of the next state of the robot to be neutral, and a 50% chance of the next state of the robot to be sad. After the initial happy state, the robot will not be happy again. From there, the robot can be either neutral or sad, depending on the value of an external ultrasonic sensor. If the ultrasonic sensor determines that an object is too close to the robot, the robot’s next state will be sad. If the object is determined not to be too close to the robot, the robot’s next state will be neutral.

A Hadamard gate (H) is a quantum gate that puts a qubit into a superpositioned state. For instance, when H is applied to *√*a qubit *|*0 *>*, then this qubit will be in the superposition of

1*/* 2(*|*0 *>* +*|*1 *>*). Otherwise, when this qubit is in*√*the state

[9, 10] and other algorithms that are only able to be executed

on a quantum computer.

*|*1 *>*, and the H is applied, the superposition is 1*/*

*−|*1 *>*).

2(*|*0 *>*

For the transition between happy and sad or happy and neutral, this Hadamard gate is used to put the qubit to superposition, which means the value of one of the qubits that contain the robot’s internal state is randomized to either 1 or 0 when it is measured, creating a corresponding transition to one of two states with equal probability.

far

Happy

H

far

Neutral

H

Sad

close

start

close

Fig. 3. Quantum State Transition Diagram Of The Emotion Machine For The Fritz Robot

By encoding the different internal states as shown in Figure 3, a state transition table is used to create the state equations for the next state of the quantum-controlled robot. The behav- ior of the robot is deterministic in states Neutral and Sad, but it is classical probabilistic in state Happy. Similarly, entangled states (example of quantum probability) can be created [1]. The ultrasonic distance sensor measures if the user is close or far from the robot modeled by the input variable y. If the user is within 50 cm of the sensor, the sensor will output a 1, otherwise it will output a 0. This is to help simulate realistic reactions of the robot to its surroundings by using the sensor to impact state changes. The encoding of internal state of the Emotion Machine is shown in Figure 4. Figure 5 presents the Transition Table of the Emotion Machine of Quantum Fritz for the discussed example.

|  |  |
| --- | --- |
| **Emotion** | **Encoding (***Q*1**,** *Q*2**)** |
| Happy | 0, 0 |
| Neutral | 0, 1 |
| Sad | 1,1 |

Fig. 4. Encoding of internal states of the Emotion Machine

|  |  |
| --- | --- |
| *Q*1**,** *Q*2**, y** | *Q*1*next***,** *Q*2*next* |
| 0, 0, 0 | 50/50 chance of 0,1 or 1, 1 |
| 0, 0, 1 | 50/50 chance of 0,1 or 1, 1 |
| 0, 1, 0 | 0, 1 |
| 0, 1, 1 | 1, 1 |
| 1, 1, 0 | 0, 1 |
| 1, 1, 1 | 1, 1 |

Fig. 5. Transition Table of the Emotion Machine of Quantum Fritz

Since the only difference between the encoding of the sad and neutral state is *Q*1, a Hadamard gate is used only on *Q*1 to randomize the next state of the machine when the machine is at the happy state. A methodology is created to encode quantum state machines with deterministic, classical probabilistic and quantum probabilistic behaviors [13,14]. The encoded “Quantum Transition Table” for the Emotion Machine is shown in Figure 6.



Fig. 6. Quantum Transition Table for the Emotion Machine of Quantum Fritz

From this, the quantum state equation is determined as *Q*1*prime* = *y* and *Q*2*prime* = 1. This circuit will intake Q1 and Q2, modeling the Boolean algebraic expressions derived. The controlled Hadamard gate is only activated if the current internal state is detected to be Happy, thus allowing for the robot to transition to its next state. The introduction of the Hadamard gate makes this circuit probabilistic through the different probabilities of values that qubits can achieve, something that could not have been achieved with permuta- tive (reversible) Boolean gates such as Toffoli and Feynman gates. This machine will also output the correct order of the three internal states for the rest of the robot’s circuit to use. In general, while reversible gates provide deterministic behaviors, rotation gates and controlled rotation gates create classical probabilistic behaviors, and Hadamard gate together with Feynman gate (the so-called Einstein-Podolsky-Rosen –

EPR circuit) is the source of entangled quantum probabilistic behavior of the robot head (See the example in [1]).



Fig. 7. Inner Logic of QFSM For Fritz Emotion Machine, The Classical D Flip-FLops and Feedback Are Not Shown. See Theory and Details in [6,13].

The dashed vertical line before the two measurements of *Q*1*prime* and *Q*2*prime* is a barrier purely for organizational purposes, to make the separation between gates neater. It is important to note that this is only the transition logic that will produce the next state when *Q*1 and *Q*2 are initialized to any input. The classical D Flip-Flop for the memory and feedback loop of the QFSM are not shown. The logic of the

actual quantum finite state machine will be in section VIII, Full Robot Circuit.

1. *Timer as Quantum Circuit*

For each internal state of the Emotion Machine, there will be a fixed sequence of four facial expressions that the robot creates for it. To keep track of the faces, a quantum circuit is created to keep an internal count of which face should be displayed at a given time.

This timer circuit contains completely deterministic logic, as it does not have a Hadamard or other rotational gate that would put some qubits into superposition.

The order of ascension of the counter would be 00, 01, 11, and 10, with each count (a phase of the gesture) being another face in an internal state the robot would transition to. As a measure of time, a counter can be established from combinations and functions of variables Q1 and Q2. A map of the transitions between the present state and the next state can be made.

|  |  |
| --- | --- |
| *Q*3**,** *Q*4 | *Q*3*next***,** *Q*4*next* |
| 00 | 10 |
| 01 | 00 |
| 11 | 01 |
| 10 | 11 |

Fig. 8. Transition Table of Timer for Quantum Fritz



Fig. 9. Quantum Circuit as Timer. Classical D Flip-Flops For Memory And Feedback Are Not Shown.

1. *Motion Generator as a Quantum Circuit*

For this combinational circuit, different positions of the servos on the robot face are associated with different values of each qubit in the Motion Generator combinational circuit. For example Figure 10 shows the robot face in a position where the qubits representing their corresponding facial parts are equal to 0 and 1, respectively.

The Motion Generator is a machine that will interpret the values produced by the Emotion Generator and Timer. It will do this using multi-controlled Toffoli gates.

The Toffoli gates combined with the negative control gates check if the control bits of each multi-controlled Toffoli are equal to 1. If so, the Toffoli gate will invert the value of its target qubit. The inversion of the measured qubits will be displayed on the robot’s face through its servos. This internal measurement of the robot’s faces can show a different face for the robot for each timer and state combination.



Fig. 10. Fritz Robot Head Facial Part Configurations

For example, Figure 11 shows the four faces of the robot in the Sad internal state with its corresponding Motion Generator circuit. The multi-controlled Toffoli Gate has control qubits on *Q*1, *Q*2, *Q*3, and *Q*4. *Q*1 and *Q*2 represent the internal emotion of the machine, while *Q*3 and *Q*4 represent the state of the timer, and specifies what stage of the gesture the robot is on. The target qubits of the multi-controlled Toffoli change the value of the qubits holding the value of the facial parts, so they show the desired facial poses for each emotion at each timer value. These qubits are then measured at the end of the circuit, which would translate the qubit’s quantum value into the value of a classical bit to be used by the Fritz robot’s standard software. Similarly as in other Quantum Machines discussed earlier, rotation gates can be inserted locally to the Motion Generator circuit causing some small probabilistic differences of executed motion, for instance only in one or two servos.



Fig. 11. The Sad Section of the Motion Generator combinational circuit in the Lower Section Of The Figure and Motion Generator’s Corresponding Robot Faces In The Upper Section Of The Figure

The full Motion Generator combinaitonal circuit for all emotion states is shown in Figure 12. The full circuit contains all possible combinations of inputs from the timer and state

machine and determines the correct value of each facial part to select the desired face for a given pose.



Fig. 12. Full Motion Generator Combinational Logic (Has been split into multiple rows for editing purposes).

1. Full Robot Circuit

There are three steps to the execution of each quantum machine circuit[6].

* 1. Initialization: Qubits are first initialized to their desired value.
	2. Quantum Evolution: The values are then put through the quantum gates to manipulate their positions.
	3. Measurement: The qubits are then measured to convert their quantum position to a binary value.

The quantum components in the circuit only handle the quantum logical processing of their respective machines and combinational circuit. The classical D flip-flops take the mea- sured values of the qubits and together with the feedback make the circuit a sequential circuit. The feedback from the measurement back to the initialization is what allows the Emotion Generator and Timer to switch to their next states, as they receive the input from their past state.

Our model of quantum state machine is the “Automaton with classical memory” which uses external D-type flip- flops[6,13].

The binary vectors of the qubits measured are then trans- lated to signals sent to the classical software of the robot to move its servos, providing the correct facial expressions at a given internal state and timer count. Complete system of two quantum machines being used for the Motion Generator combinational circuit is shown in Figure 13.



Fig. 13. Full Quantum Fritz Circuit

*A. Frequency Divider for Clock*

The two D Flip-Flops labeled *D*5 and *D*6 is a frequency divider for the clock. This works by increasing the time for the clock fed into *D*1 and *D*2. By feeding the output of *D*5 to *D*6, D will lower the frequency of the active edge of the clock by 1*/*2. When this D Flip-Flop configuration is repeated one more time with *D*6 the output of *D*6 will switch to 1 1*/*2 of 1*/*2 of the time the clock is switching on. This allows the clock to send have its clock pulse changing from 0 to 1, setting off the positive edge triggered flip-flops *D*3 and *D*4 four times for every one time the positive edge is triggered for *D*1 and *D*2. This will allow the Motion Generator combinational circuit and Timer Circuit to run four times generating a gesture of faces for every one emotion generated.

1. Robot Actualization

The robot head’s servos are controlled by the Arduino Uno REV3 microcontroller board with the accompanying Arduino programming software which is used to control the robot head servos. Arduino does not have quantum computing software, so the quantum logic used in the programming of the robot is programmed in the Python programming language, due to the open source software Qiskit, which is able to simulate quantum circuits, as well as access real IBM quantum computers.[18] The Quantum Fritz uses the quantum simulator from Qiskit.

The robot will go through several steps to actualize its behavior:

1. Take the current state of the iteration of the robot stored in the D Flip-Flops (From the Emotion Generator and Timer), and uses it as input for next clock cycle.
2. The program will generate the new values of the Emo- tion Generator and Timer, and use it as the input to the Motion Combinational Circuit.
3. It will then send outputs of the Motion Combinational Circuit back to the Arduino Uno microcontroller via serial port, Arduino interprets the signals sent as servo values
4. Robot face is moved by servos, displaying a facial expression.
5. Conclusions

In this project, we have developed a system that combines emotional head gestures, natural language conversation and quantum control. Our system uses voice recognition, OpenAI’s GPT-3.5, and an Arduino board to control servo motors based on user input. The system can recognize speech input from the user, convert it into text, and process it using GPT-

3 to generate a response. The response is then converted into voice output, and an Arduino board controls the servo motors based on the response. The motions are generated by a combination of quantum and classical software. Quantum control is based on quantum state machines. The robot was able to move its facial gestures and demonstrate many different moods using multiplexers in the circuit to allow for groups of movement to happen based on internal state. This was created using quantum sequential circuits that transitions between its quantum brain and classical servos seamlessly.

The system has been successfully implemented and tested, and it can be used for various applications such as controlling robotic arms, home automation, and other IoT-based applica- tions. There is a lot of potential for future improvements and advancements in this project. Here are a few ideas for future work:

1. Expanding the system to control multiple servo motors simultaneously, which can be useful in applications such as robotic arms.
2. Integrating machine learning algorithms to learn from user interactions and generate more accurate responses over time [11].
3. Adding personalized natural language processing capa- bilities to the system to understand texts and generate texts related to specific topics of research such as quantum computing that are not available on ChatGPT.

References

1. [1] R. Deng, Y. Huang, and M. Perkowski, “Quantum motions and emotions for a humanoid robot actor,” 2021 IEEE 51st In- ternational Symposium on Multiple-Valued Logic (ISMVL), 2021. doi:10.1109/ismvl51352.2021.00043
2. A. Raghuvanshi and M. Perkowski, ”Fuzzy quantum circuits to model emotional behaviors of humanoid robots,” IEEE Congress on Evolutionary Computation, Barcelona, Spain, 2010, pp. 1-8, doi: 10.1109/CEC.2010.5586038.
3. A. Raghuvanshi, Y. Fan, M. Woyke and M. Perkowski, ”Quantum Robots for Teenagers,” 37th International Symposium on Multiple- Valued Logic (ISMVL’07), Oslo, Norway, 2007, pp. 18-18, doi: 10.1109/ISMVL.2007.46.
4. P. Benioff, “Quantum Robots Plus Environments,” Quantum Com- munication, Computing, and Measurement 2, pp. 3–9, Jul. 1998.

doi:10.1007/0-306-47097-7 1

1. E. Kagan, E. Salmona, and I. Ben-Gal, “Probabilistic mobile robot with quantum decision-making,” 2008 IEEE 25th Conven- tion of Electrical and Electronics Engineers in Israel, Dec. 2008. doi:10.1109/eeei.2008.4736561
2. [1] M. Lukac and M. Perkowski, “Quantum finite state machines as sequential quantum circuits,” 2009 39th International Symposium on Multiple-Valued Logic, 2009. doi:10.1109/ismvl.2009.46
3. D. Dong, Ch. Chen, Ch. Zhang and Z. Chen, Quantum Robot: structure, algorithms and applications, Robotica, Cambridge University Press, 21. Febr. 2006.
4. M. A. Nielsen, I. L. Chuang, Quantum Computation and Quantum Information (Cambridge University Press. Cambridge, England, 2000)
5. L. K. Grover, “A fast quantum mechanical algorithm for database search,” Proceedings of the twenty-eighth annual ACM sym- posium on Theory of computing - STOC ’96, May 1996. doi:10.1145/237814.237866
6. J. Viertel, R. Deng, Y. Huang, G. Opsahl and M. Perkowski, The First Quantum Robot with Grover’s Algorithm Brain Solving Various Shortest Path Problems, PSU Report, December 2023, to be soon submitted.
7. J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, “Quantum machine learning,” Nature, vol. 549, no. 7671, pp. 195–202, 2017.
8. ChatGPT, openai.com/blog/chatgpt
9. Y. Huang and M. Perkowski, One Hot Encoding Synthesis of Quantum Automata from Flowcharts, Journal of Quantum Information Science, Vol. 13, No. 3, 2023.
10. M. Bao, M. Shah, M. Perkowski, Quantum Fritz – an intelligent quantum emotional talking head, technical report PSU, December 2023. Will be available soon on a webpage.
11. R.J. Romano III, AV Facial Recording And Landmarking Dataset Tool, Final Report 2021 ECE, PSU.
12. hackintoshlover12 and Instructables, “Fritz - Animatronic Robotic head,” Instructables, [https://www.instructables.com/FRITZ-ANIMATRONIC-](http://www.instructables.com/FRITZ-ANIMATRONIC-) ROBOTIC-HEAD/ (accessed Dec. 14, 2023).
13. “Fritz Emoji Arduino controlled robot kit [Advanced edition],” Siqma Robotics, https://store.siqma.com/fritz-robot-kit.html (accessed Dec. 14, 2023).
14. R. Wille, R. Van Meter and Y. Naveh, ”IBM’s Qiskit Tool Chain: Work- ing with and Developing for Real Quantum Computers,” 2019 Design, Automation Test in Europe Conference Exhibition (DATE), Florence, Italy, 2019, pp. 1234-1240, doi: 10.23919/DATE.2019.8715261.