Arduino Based Gesture Control System
&
Chat Bot System Using Python

A NEC Project Submitted in partial fulfillment of the Requirement for the Degree of

Bachelor of Technology

By

Akash Singh Tiwari (0901EE221013)
Electrical Engineering Department
V Semester

Under the Supervision of

Dr. Karuna Markam
Department of Electronic Engineering

NEC in Robotics

CENTRE FOR ARTIFICIAL INTELLIGENCE
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR, INDIA
(DEEMED TO BE UNIVERSITY)
NAAC Accredited A++ grade
Nov 2024

MADHAV INSTITUTE OF TECHNOLOGY AND SCIENCE, GWALIOR

(Deemed to be University)

(A Govt. Aided UGC Autonomous & NAAC Accredited A++ Grade Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that Akash Singh Tiwari (0901EE221013) have submitted the mini-skill based project report titled "Department of Electronic Engineering" under the guidance of **Dr. Karuna Markam** in partial fulfillment of the requirement for the award of the degree of **B. Tech in Department of Electronic Engineering** from Madhav Institute of Technology and Science, Gwalior.

Dr. Karuna Markam

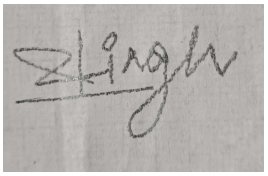
Assistant Professor

Department of Electronics Engineering

DECLARATION

I hereby declare that the Mini skill-based project for the course **Nec in robotics** is being submitted in the partial fulfillment of the requirement for the award of **Bachelor of Technology in Electrical Engineering Department**

All the information in this document has been obtained and presented in accordance with academic rule and ethical conduct.



Date:18 November, 2024 Place: Gwalior, MP

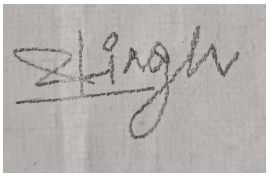
Akash Singh Tiwari 0901EE221013

ACKNOWLEDGEMENT

I would like to express my greatest appreciation to all the individuals who have helped and supported me throughout this Mini skill-based project. I am thankful to whole Centre of AI department and mechanical department for their ongoing support during the experiments, from initial advice and provision of contact in the first stages through ongoing advice and encouragement, which led to the final report of this Mini skill-based project.

I wish to thank my professor Dr. Karuna Markam as well for his undivided support and interests which inspired me and encouraged me to go my own way without whom I would be unable to complete my project.

At the end, we want to thank my friends who displayed appreciation for my work and motivated me to continue my work.



Akash Singh Tiwari 0901EE221013

Table of Content

Chapter-1 Project Overview

1.1. Introduction

In today's technology-driven world, Human-Computer Interaction (HCI) plays a crucial role in enhancing user experiences and making systems more intuitive. Gesture recognition, a significant aspect of HCI, enables devices to interpret human gestures as commands, allowing for a more natural and seamless interaction. This project focuses on developing a gesture detection system using Arduino, integrated with a chatbot system to provide a user-friendly interface for interaction.

The hardware component utilizes an Arduino microcontroller to detect specific hand gestures through sensors, while the software component features a chatbot capable of processing and responding to user inputs. The integration of these two systems opens up possibilities for creating smarter and more interactive applications, from controlling smart devices to enhancing accessibility for users with mobility constraints.

By combining hardware and software, this project aims to explore how gesture-based controls and conversational interfaces can work together to create an innovative and responsive user interaction system.

1.2. Project Features

1. **Gesture Detection Using Arduino:**
    - The system uses an Arduino microcontroller to recognize specific hand gestures via sensors (e.g., flex sensors, accelerometers, or ultrasonic sensors).
    - Real-time processing of gesture data for quick and accurate recognition.
    - Easy customization to detect a variety of gestures for different applications.
2. **Chatbot Integration:**
    - The chatbot is designed to interact with users, providing responses based on user commands or queries.
    - Natural language processing (NLP) capabilities allow the chatbot to understand and respond in a conversational manner.
    - Seamless interaction between gesture inputs and the chatbot for an enhanced user experience.

3. **User-Friendly Interface:**
    - The system provides an intuitive interface where users can control devices or interact with the chatbot using hand gestures or text-based inputs.
    - Suitable for users with limited mobility, as gestures provide an alternative to traditional input methods like typing or touch.
4. **Modular Design:**
    - Both hardware and software components are designed to be modular, allowing for easy upgrades or integration with additional features.
    - The system can be expanded to control smart home devices, robots, or even gaming applications.
5. **Real-Time Feedback:**
    - Immediate feedback from both the gesture detection system and the chatbot enhances interactivity.
    - Visual or auditory outputs can be used to confirm recognized gestures or chatbot responses.
6. **Cross-Platform Compatibility:**
    - The chatbot system is designed to be compatible with various platforms, such as web applications, desktop software, or mobile apps.
    - This ensures that the system can be used in different environments and for a wide range of applications.
7. **Energy-Efficient and Cost-Effective:**
    - The hardware components are chosen for their low power consumption and affordability, making the project suitable for low-cost, sustainable implementations.
    - The project emphasizes simplicity in design without compromising on performance.

1.3. Feasibility

1. **Technical Feasibility:**
    - **Hardware Availability:** The components required for gesture detection, such as an Arduino microcontroller, sensors (e.g., flex sensors, accelerometers), and necessary wiring, are widely available and cost-effective. Arduino is a well-documented and supported platform, making the implementation straightforward.
    - **Software Integration:** Developing a chatbot using widely used frameworks like Dialogflow, Rasa, or custom-built NLP models is feasible due to the availability of robust open-source libraries and APIs. Integration between the Arduino hardware and the chatbot system can be achieved using serial communication or IoT platforms.
    - **Skill Requirements:** Basic to intermediate knowledge in Arduino programming, sensor interfacing, and chatbot development is required. Tutorials, documentation, and online resources are readily accessible to assist in building both components, making it feasible for an engineering student with a background in electronics and programming.
2. **Economic Feasibility:**
    - **Low Cost:** The project requires affordable components such as the Arduino board, sensors, and basic connectivity tools. The software component can be developed using free or open-source chatbot frameworks, reducing the overall cost of the project.
    - **Maintenance:** The modular nature of the project allows for easy updates and maintenance, further minimizing long-term costs. The system can also be expanded at minimal expense by adding more sensors or enhancing the chatbot's capabilities.

3. **Operational Feasibility:**
    - **Ease of Use:** The project is designed to be user-friendly, with intuitive gesture controls and a conversational chatbot interface. This makes the system accessible even for non-technical users or people with limited mobility, improving the system's adaptability in real-world scenarios.
    - **Real-Time Functionality:** The system is capable of processing gestures and chatbot queries in real time, ensuring that users receive immediate feedback, enhancing practicality in daily operations or real-time control of devices.
4. **Market Feasibility:**
    - **Broad Applications:** The combination of gesture recognition and chatbot technology has a wide range of applications, including smart home systems, assistive technology for people with disabilities, interactive gaming, and robotics. This makes the project highly relevant to current market trends.
    - **Scalability:** The project can be easily scaled by adding more gestures or chatbot capabilities, making it suitable for larger, more complex applications without requiring a complete redesign.
5. **Sustainability Feasibility:**
    - **Energy Efficiency:** The hardware components, especially the Arduino board and sensors, are known for their low power consumption, ensuring that the system can run for extended periods without excessive energy use.
    - **Upgradability:** The modular design allows for continuous improvement of both hardware and software components, ensuring the longevity of the system and adaptability to future needs.

Chapter-2 Hardware Project

2.1. Tools Used

Arduino Uno or Arduino Nano:

- The central microcontroller for the project, used to process sensor inputs and control other hardware components.
- Chosen for its ease of use, affordability, and ample community support.

Gesture Recognition Sensors:

- **Flex Sensors:** Used to detect the bending motion of fingers or hands. These sensors change their resistance when bent, allowing for gesture-based input.
- **Accelerometer (e.g., MPU6050):** Detects movement and orientation of the hand in 3D space, useful for more complex gesture detection.
- **Ultrasonic Sensor:** Can be used to detect hand proximity or motion by measuring the time it takes for sound waves to return after bouncing off a nearby object.

Breadboard and Jumper Wires:

- For easy prototyping and connections between the Arduino, sensors, and other components without soldering.
- Allows flexibility in circuit design and quick adjustments.

Power Supply:

- A 9V battery or USB cable can be used to power the Arduino board during operation.
- Additional voltage regulators might be required to provide the correct voltage for sensors.

Resistors and Capacitors:

- Used in conjunction with sensors to stabilize signals and ensure the system operates smoothly.
- Essential for managing current and voltage in the circuit.

LEDs and Buzzer (Optional):

- LEDs can be used to provide visual feedback when a gesture is recognized.
- A buzzer can be used for auditory feedback, indicating successful detection of gestures.

2.1.1. Hardware Tools

Arduino Uno: Arduino Uno is a popular microcontroller board based on the ATmega328P chip. It features digital and analog input/output pins, making it versatile for various projects. Uno has a USB connection for programming and power, and it's compatible with a wide range of sensors, actuators, and shields, making it ideal for beginners and experienced makers alike.

Fig 1 Arduino Uno

Jumper Wires: Jumper wires are essential electrical components used in electronics and prototyping projects. They consist of flexible wires with connectors at each end, typically male pins or female sockets. They facilitate the temporary connection of electronic components on breadboards, allowing for easy experimentation, circuit testing, and rapid prototyping in electronics projects.

Fig 4 Jumper Wires

2.1.2 Software Tools

Arduino IDE: The Arduino Integrated Development Environment (IDE) is a software platform used for programming Arduino microcontrollers. It features a simple yet powerful interface for writing, compiling, and uploading code to Arduino boards. With built-in libraries and examples, the IDE streamlines the process of creating embedded projects for enthusiasts and professionals alike.

Fig 5 Arduino Ide

2.1.3Code

```
/*
 * Created by ArduinoGetStarted.com
 *
 * This example code is in the public domain
 *
 * Tutorial page: https://arduinogetstarted.com/tutorials/arduino-ultrasonic-sensor-piezo-buzzer
 */

// constants won't change
const int TRIG_PIN   = 6; // Arduino pin connected to Ultrasonic Sensor's TRIG pin
const int ECHO_PIN   = 7; // Arduino pin connected to Ultrasonic Sensor's ECHO pin
const int BUZZER_PIN = 3; // Arduino pin connected to Piezo Buzzer's pin
const int DISTANCE_THRESHOLD = 10; // centimeters

// variables will change:
float duration_us, distance_cm;

void setup() {
  Serial.begin (9600);        // initialize serial port
  pinMode(TRIG_PIN, OUTPUT);   // set arduino pin to output mode
  pinMode(ECHO_PIN, INPUT);    // set arduino pin to input mode
  pinMode(BUZZER_PIN, OUTPUT); // set arduino pin to output mode
}
void loop() {
  // generate 10-microsecond pulse to TRIG pin
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  // measure duration of pulse from ECHO pin
  duration_us = pulseIn(ECHO_PIN, HIGH);
  // calculate the distance
  distance_cm = 0.017 * duration_us;

  if(distance_cm < DISTANCE_THRESHOLD)
    digitalWrite(BUZZER_PIN, HIGH); // turn on Piezo Buzzer
  else
    digitalWrite(BUZZER_PIN, LOW);  // turn off Piezo Buzzer


  // print the value to Serial Monitor
  Serial.print("distance: ");
  Serial.print(distance_cm);
  Serial.println(" cm");

  delay(50);
}
```

Chapter-3 Software Project

3.1. Software Used: Botpress

Botpress is an open-source platform designed for building conversational AI chatbots. It provides developers with a highly customizable environment for creating intelligent chatbots that can interact with users in a natural, human-like manner.

3.2. Key Features of Botpress

Open-Source and Flexible:
Botpress is open-source, allowing developers to access its full codebase for customization. This gives you control over how your chatbot behaves, looks, and integrates with other systems.
The platform is built on Node.js, making it accessible for developers familiar with JavaScript.

Modular Architecture:
Botpress is modular, meaning it consists of different components or "modules" that handle specific functionalities, such as Natural Language Processing (NLP), user interaction flows, and content management.
You can add or remove modules based on your project's requirements, which makes it scalable and adaptable to different use cases.

Visual Flow Builder:
One of Botpress's most user-friendly features is its drag-and-drop flow builder. It allows you to visually design the conversation logic without having to code the entire chatbot manually.
You can create custom user journeys, defining how the bot interacts with users based on the flow and context of the conversation.

Natural Language Processing (NLP):
Botpress includes a built-in NLP engine that can handle intent detection, entity recognition, and context management.
This enables the chatbot to understand user inputs more effectively and provide relevant responses based on the intent.

Multi-Channel Support:
Botpress can be deployed across multiple channels, such as websites, mobile apps, social media platforms (e.g., Facebook Messenger), and even hardware systems.
This makes it ideal for projects where you need cross-platform interaction or integration with different systems.

Customizable and Extensible:
Developers can extend Botpress's functionality by creating custom modules or using APIs to integrate it with external services or data sources.
You can also customize the look and feel of the chatbot interface to fit your brand or project requirements.

Context Management:
Botpress excels in managing context, meaning the bot can remember previous conversations or data inputs to provide continuity in the dialogue.
This is particularly useful for multi-step interactions, where the bot needs to refer back to earlier inputs or store user preferences for future interactions.

Analytics and Reporting:
Botpress offers built-in analytics features, allowing you to track user interactions, engagement rates, and other important metrics.
You can use these insights to refine your chatbot's performance over time, improving the overall user experience.

Integration with External Services:
Botpress allows easy integration with various third-party services, such as CRM systems, databases, or external APIs, enabling more dynamic and responsive chatbots.
This is useful for fetching data in real-time, interacting with hardware, or extending the bot's functionalities.

Bot Training and Learning:
The platform supports training for intents and entities, improving the bot's understanding of user queries over time.
You can also manually fine-tune the NLP model as needed, or use pre-trained models for quicker setup.

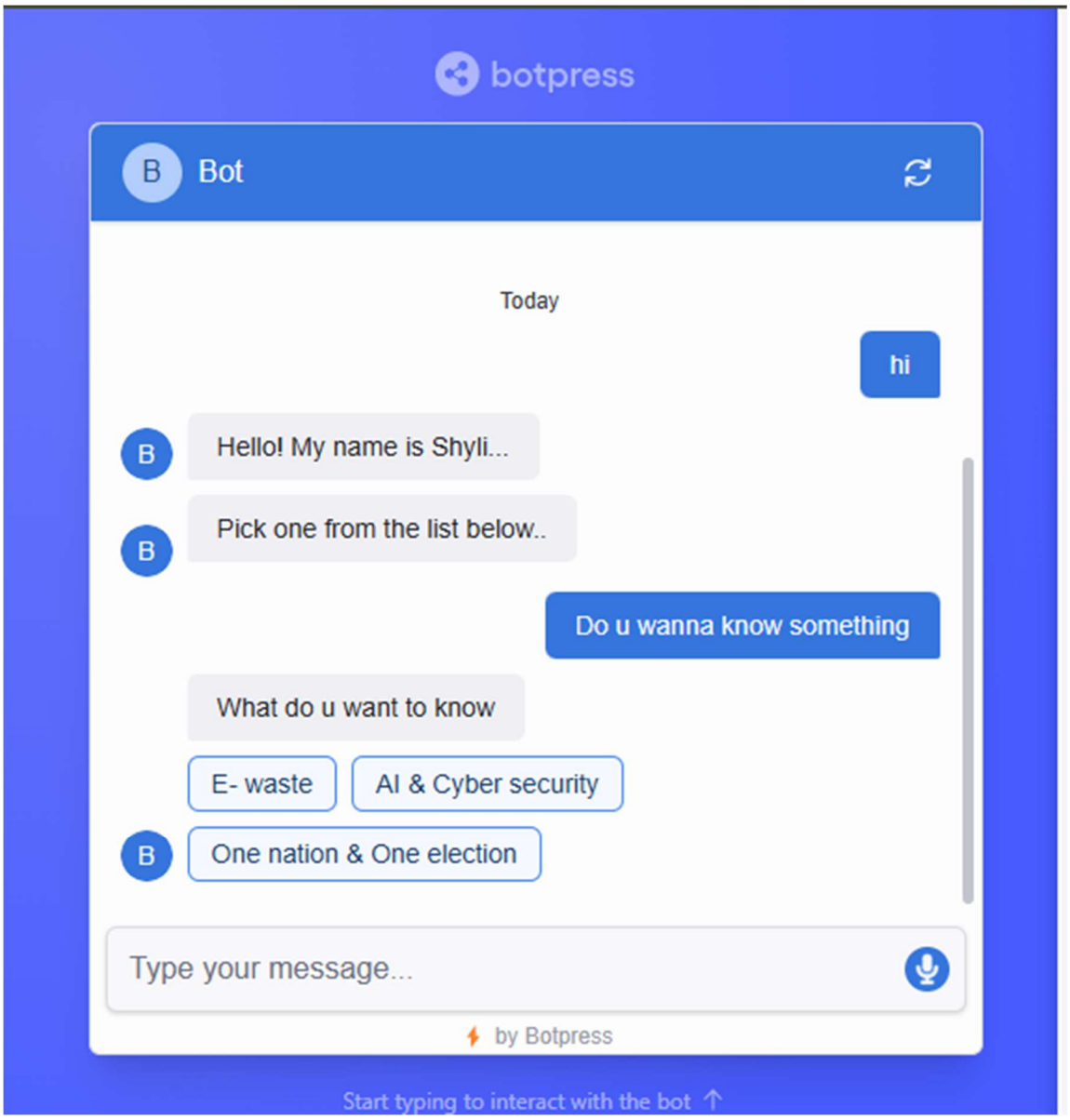
3.3. Why Use Botpress for Your Project?

Ease of Use:
With its visual flow builder and modular structure, Botpress makes it easy to design and deploy chatbots, even if you're not an expert in AI or machine learning.
Customization and Control:
You have full control over the chatbot's behavior, look, and integrations, which is useful for complex projects like yours that involve hardware interactions (gesture detection).

Low Cost and Open Source:
Since Botpress is open-source, it's free to use for most applications, with paid versions available for enterprise-level support and feature.

Link Of the bot

https://cdn.botpress.cloud/webchat/v2.2/shareable.html?configUrl=https://files.bpcontent.cloud/2024/10/20/12/20241020125726-K481YZZN.json



References:

https://github.com/Anikait22/Gesture-Detection
https://www.instructables.com/Gesture-Sensor-Control-Using-Arduino/
https://projecthub.arduino.cc/mellis/gesture-recognition-using-accelerometer-and-esp-f9e4f9
https://www.restack.io/botpress-self-hosted-setup-guide
https://github.com/botpress/botpress