BIPED ROBOTIC DOG

ADITHI JAYARAMAN
FALMOUTH UNIVERSITY

INTRODUCTION

This project aims to propose a bipedal robot for the "human vs. robot" competition. A biped is essentially a two-legged autonomous agent that can walk. Additionally, this biped can avoid obstacles while mimicking the overall general movement of any two-legged machine. The main objectives of this robot are mobility, walking front and back, obstacle avoidance, and head movement. This is achieved using C++, CAD design, and Arduino electronic construction.

CIRCUIT CONNECTION

The hardware configuration of the control system encompasses five servo motors, an ultrasonic sensor, and an Arduino Uno microcontroller. The servo motors provide precise actuation for various movements, interfacing with the Arduino Uno via PWM pins to receive control signals for angular adjustments. Meanwhile, the ultrasonic sensor, integrated with the Arduino Uno through digital I/O pins, serves as the primary sensory input, enabling distance estimation and environmental perception. The Arduino Uno acts as the central processing unit, orchestrating the system's operation by reading sensor inputs, processing data, and generating control signals for servo motors. This cohesive integration of hardware components forms a versatile control system capable of executing diverse tasks efficiently.

METHOD

The design process begins with 3D CAD software used to create brackets for connecting SG90 servo motors to each leg joint and the head movement servo, ensuring stability and precise alignment. Cardboard components are then assembled based on CAD designs, with servo motors securely attached using adhesive or mounting screws. The electronic circuit is built, connecting servo motors to specific Arduino pins, followed by programming to control the robot's movements based on sensor input. Customized code enables actions such as walking and head movement in response to feedback from the ultrasonic sensor. Servo motors are calibrated for stability and accuracy, with testing and refinement phases optimizing the robot's performance in various scenarios.

UML DIAGRAM & FUNCTIONALITY

UML Explanation (Figure 1)
Initial and Ready State: This is the starting point of the state machine, indicating that the system is in an idle or initialization phase. In ready state, the robot is ready to perform actions and awaits input or commands from the environment or user.

Obstacle Detected State: When an obstacle is detected within close range, the system transitions to this state. The robot's behavior in this state involves scanning the environment to determine the obstacle's location.

Head Movement States: These states represent the robot's head movement to scan for obstacles. This could include states like "Move Head to Right" and "Move Head to Left", indicating the direction in which the head moves to gather sensory information.

Obstacle on Right or Left Side Sub-State: These states indicate the presence of an obstacle on either the right or left side of the robot, triggering actions to avoid collision.

Turn Right, Turn Left State or move Forward state: In these states, the robot adjusts its direction to avoid obstacles, either by turning right or left. Or in the moving forward state the robot moving forward while ensuring there are no obstacles in its path.

Obstacle Cleared State: When the obstacle is no longer detected within the predefined range, the system transitions back to the "Ready" state, indicating that the obstacle has been successfully avoided.

Functionality
The obstacle-avoiding biped robot first senses its environment through the ultrasonic sensor, determining obstacles' presence and location. It then uses this information to decide on the safest route to its destination, adjusting its navigation accordingly. While continuously assessing its surroundings, the robot adapts its movements in real time to avoid collisions and maintain stability hence working like an obstacle avoiding robot

IMAGE

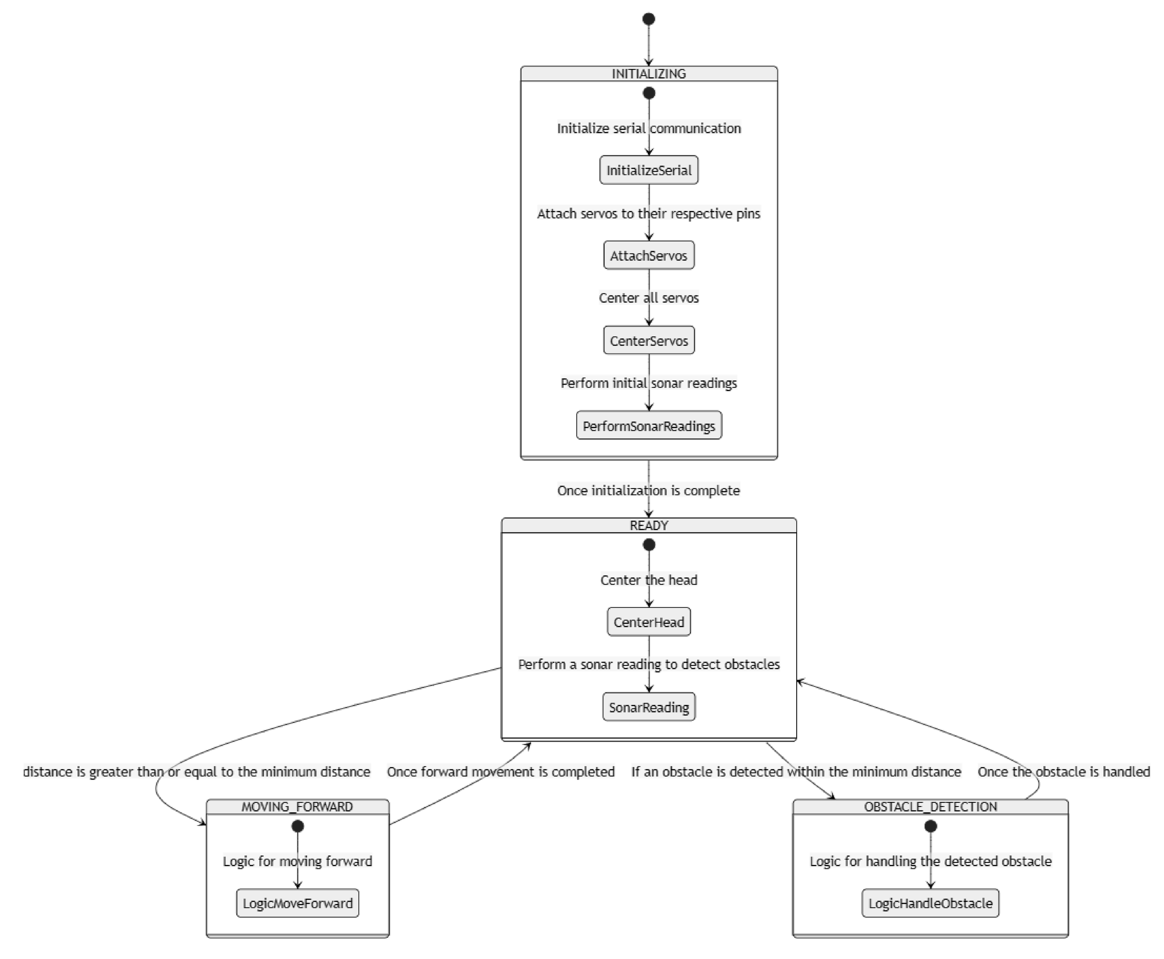

Figure 1: UML State Diagram .

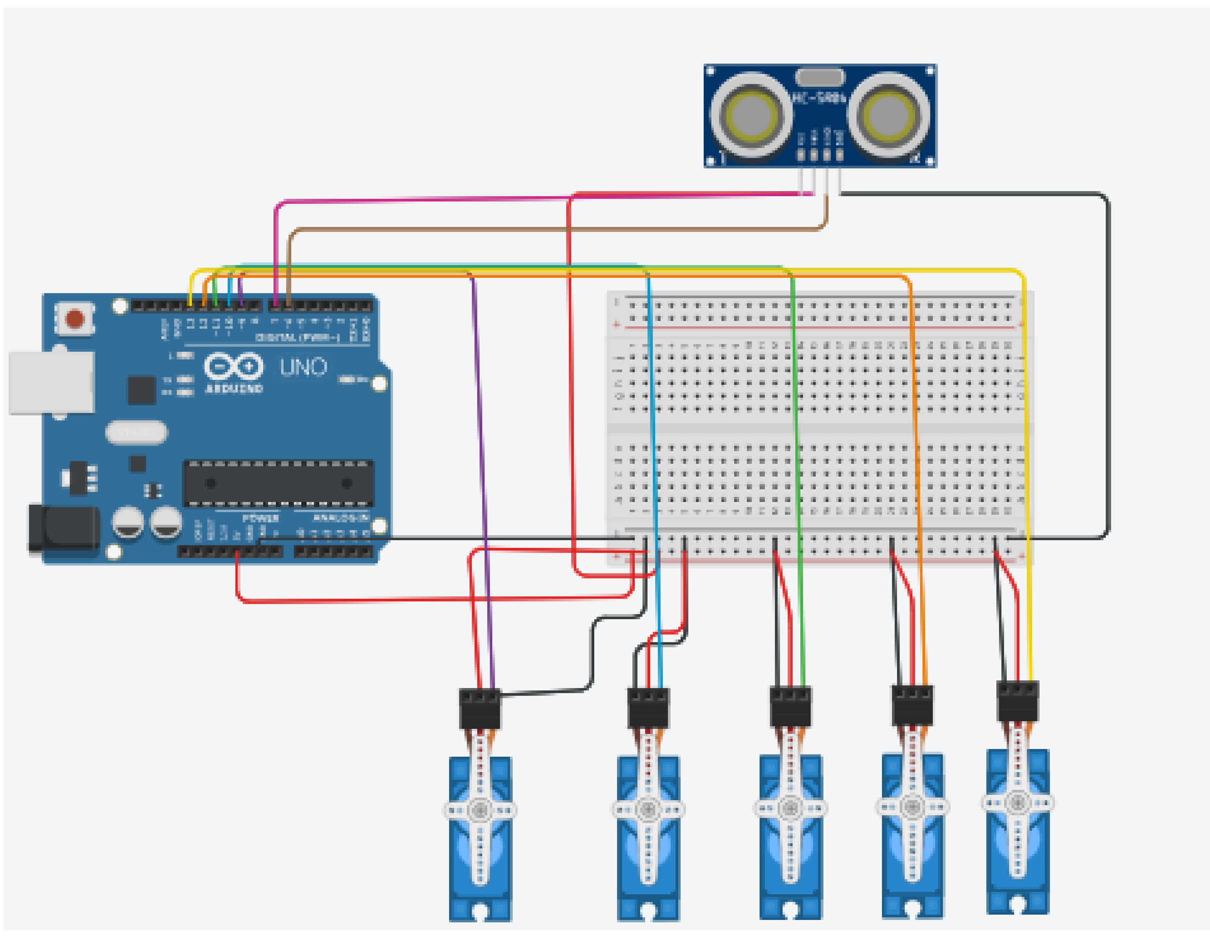

Figure 2: Biped Circuit connection

MATERIALS NEEDED

Electrical components

- x5/3 servos
- x1 ultrasonic sensor
- x1 Arduino Uno
- many male to male and female and male wires

3D Printed Parts

- x2 90 Degree Brackets
- x2 top and bottom plate
- x2 front and back plate

Other components

- Pieces of waste cardboard
- x1 Ultrasonic sensor holder
- x1 Hot glue

CODE

This code controls a bipedal robot, enabling autonomous navigation through its environment using a finite-state machine. It facilitates obstacle avoidance and basic locomotion, including forward movement, turning, and obstacle detection. The software utilizes the Servo.h library for servo control and distance measurement. Pin definitions are specified for sensors and servos, with a state machine managing robot behavior.

During setup, serial communication is initialized, servos are centered, and initial sonar readings are taken. The loop function executes state machine logic for transitions between initialization, obstacle detection, and movement states, with functions like GetSonar() and CenterServos() facilitating sensor readings and servo control. Additionally, error-handling mechanisms are implemented to ensure robust operation under various conditions.

CONCLUSION

This project provided valuable experience in implementing states within code and constructing state diagrams. While the initial circuitry appeared straightforward, the project's complexity exceeded expectations. The mechanical aspects, particularly finding optimal servo placement and circuit configuration for bipedal balance and gait, proved to be the most challenging element. In further development, I would focus on incorporating additional joints to achieve a more human-like walking pattern, surpassing the current four-joint, rudimentary walking motion.

REFERENCES

Github Link :- https://github.falmouth.ac.uk/Games-Academy-Student-Work-23-24/COMP102-1-2306514