AILite Robot Documentation

Who are you? Ans: I am AiLite an advanced AI Technology of Meritus AI designed by Vishnu J ,Machine Learning Engineer.

Introduction

AILite is a cutting-edge autonomous robot designed for a wide range of applications. With its sophisticated four-wheel drive system, integrated camera, ultrasonic sensors, IR sensor, color sensor, and touch sensor, AILite is capable of navigating complex environments with precision and intelligence. This versatile bot is perfect for tasks requiring advanced obstacle detection, machine learning applications, and real-time monitoring, making it an ideal solution for both educational and industrial purposes.Python is used for execution of the projects.

Are there any similar products like AiLite? Yes ,CvPro,DataScience pro, CvPro classic. For more information do visit https://www.robotixedu.com/

AiLite is equipped with a variety of sensors and modules, including an ultrasonic sensor with a range of 2 cm to 400 cm, an infrared sensor with a range of approximately 11-14 cm, and a color sensor with a range of about 5 mm. The motor operates at speeds ranging from 175 to 255 RPM, and the device is powered by a battery that lasts for 2 hours and 30 minutes, with a charging time of 3 hours using a C-type cable. The camera module is high-resolution, and the touch sensor detects contact by indicating whether a touch is sensed or not.

If you accidentally insert the cable into the wrong port, AiLite may not function correctly or may respond slowly, which could adversely affect its performance. It's important to connect the device properly to avoid such issues. To run AiLite, you'll need a laptop with Miniconda installed, at least 2 GB of usable storage in your user profile path (e.g., C:\Users\John), and a stable Ethernet connection.

Due to port constraints, only three sensors can be connected at a time. For more details on this limitation, please refer to the user manual. Looking ahead, AiLite's future development plans include adding a gripper for pick-and-place tasks, as well as integrating various LLM and GenAI-based projects. While Python is the primary language for running AiLite, it can also be accessed via a web URL or through MIT App Inventor for various projects. When connected to a web server, AiLite should first be connected to a desktop or laptop via Wi-Fi, after which you can use a unique URL to access the web server and test the respective peripherals.

To launch MIT App Inventor and start working on your projects, first, install the MIT AI2 Companion app on your mobile device from the Google Play Store or Apple App Store. Then, on your computer, visit appinventor.mit.edu and sign in. Open your existing project or create a new one. In the MIT App Inventor interface, click the "Connect" button at the top of the screen, and select "AI Companion" from the dropdown menu. Next, launch the MIT AI2 Companion app on your mobile device, and either scan the QR code displayed on your computer screen or enter the provided code manually. Your project will load on your mobile device, allowing you to test your app in real-time.For more information visit https://appinventor.mit.edu/

Among the projects and lessons available for AiLite using MIT App Inventor, two examples are the "Tilt Control AiLite-L2" and "Speech Control AiLite-L2." In the Tilt Control project, the user tilts their device forward for forward movement, backward for backward movement, and more, with AiLite implementing the corresponding actions. In the Speech Control project, the user gives voice commands like "move forward" or "move backward," which AiLite processes to perform the respective actions.

Overview

- **Multi-functional Robot:**
  - Equipped with a versatile set of sensors and modules.
  - Capable of performing various tasks such as face recognition, text recognition, object detection, natural language processing (NLP), hand gestures, and more.

Key Features:

- **Mobility:**
  - Moves forward, backward, left, and right with precision control.
- **Camera Module:**
  - Supports real-time video streaming.
- **Sensors:**
  - **Ultrasonic Sensor:** Measures distance to avoid obstacles.
  - **IR Sensor:** Detects infrared signals for various applications.
  - **Color Sensor:** Identifies colors for sorting and recognition tasks.
  - **Touch Sensor:** Reacts to physical touch for interactive control.
- **Advanced Capabilities:**
  - **Remote Control:** Operate the bot remotely via a user-friendly interface.
  - **Natural Language Processing (NLP):** Understands and responds to voice commands through actions.

Component List

Brain Block:

- Quantity: 1

Motor Blocks:

- Quantity: 4

Ultrasonic & IR Sensor Block:

- Quantity: 1

Wheels:

- Quantity: 4

Camera Module Block:

- Quantity: 1

Color Sensor Module:

- Quantity: 1

Touch Sensor Block:

- Quantity: 1

Half Brick - 2 x 4:

- Quantity: 2

Half Brick - 2 x 2:

- Quantity: 2

Plates – 2 x 6:

- Quantity: 4

Half Bricks – 2 x 6:

- Quantity: 5

Half Bricks - 1 x 6:

- Quantity: 2

Traffic Signs:

- Quantity: 4

Custom Blocks:

- Quantity: 2

Black Caster Cap:

- Quantity: 1

AILite Construction

Follow these steps to assemble AILite using the provided kit:

1. **Motor Setup:** Assemble the motors according to the instructions.
2. **Brain Block Assembly:** Connect the Brain Block to the motor assembly.
3. **Brain Block Placement:** Secure the Brain Block in its designated position.
4. **Fixing Wheels:** Attach the wheels to the motor assembly.
5. **Camera Module:** Install the camera module on the robot.
6. **Ultrasonic Sensor:** Attach the ultrasonic sensor to the front of the robot.
7. **Color Sensor:** Install the color sensor on the robot.
8. **Touch Sensor:** Attach the touch sensor for interactive control.

Video Tutorial:

- For Two Wheels Setup: https://vimeo.com/945741571/05dd714ddd?share=copy

AILite Connection & Testing

- **Connection Check:** Ensure all components are connected properly.
- **Color Significance:** Understand the significance of each color in connections.
- **Testing:** Test the robot's movements and sensor values to ensure proper operation.

Video Tutorial:

- AILite Connection & Testing: https://vimeo.com/963150265/c8cdb6c88a?share=copy

Do’s and Don'ts

- **Do's:**
  - Follow the guidelines for optimal performance.
  - Ensure safe and efficient operation.
  - Regularly check and maintain the bot for longevity and reliability.
- **Don'ts:**

- Avoid common mistakes that may cause damage.
- Do not expose the bot to extreme conditions.

PYTHON SETUP FOR AILITE
PREREQUISITES NEED TO BE INSTALLED :
Visual Studio Code 2) Python (Preferable Version : 3.11.10) 3) Install Python interpreter inside the VS code 4) Miniconda

CREATING MINICONDA ENVIRONMENT & REQUIREMENTS INSTALLATION: Step 1 : Open miniconda and create a separate miniconda environment with python 3.11 in the respective file path for this program: conda create -n python=3.11 -y Step 2 : Activate the environment: conda activate Step 3 : Now install the required Python packages: pip install -r requirements.txt Step 4 : Run the python program python

PYTHON PROJECTS Or LESSONS

#Object Detection

This project is designed to detect and recognize traffic signs using YOLO (You Only Look Once) object detection and then control a bot accordingly based on the detected sign.Refer python code or requirements.txt for further information

# Text Recognition

This project utilizes OCR (Optical Character Recognition) to detect text from a webcam feed and control a bot accordingly. It can recognize commands like "go" and "stop" and send corresponding signals to a specified IP address.Refer python code or requirements.txt for further information

#Hand Gesture Recognition

This project utilizes the Mediapipe library to perform hand gesture recognition using a webcam feed. It detects various hand gestures and sends corresponding commands to a specified host.Refer python code or requirements.txt for further information

#Face Motion Detection

This project utilizes MediaPipe's Face Mesh model to estimate the pose of a person's head using their nose tip landmark. Based on the head pose, it sends corresponding commands to a specified host, allowing for head-controlled actions.Refer python code or requirements.txt for further information

# Face Recognition Access Control System

This project performs face recognition using the DeepFace library and grants access based on the recognized faces. If a face in the captured frame matches any of the reference images, access is granted and commands can be triggered using keyboard inputs.Refer python code or requirements.txt for further information

# Remote-Controlled Bot

This project allows you to control a bot using keyboard inputs. You can use arrow keys to move the bot in different directions and the space key to stop it. Press 'ESC' to exit the program.Refer python code or requirements.txt for further information

# Speech Recognition Bot Control

This project allows you to control a bot using voice commands. You can give commands such as "turn right", "turn left", "go forward", or "go backward" using your microphone.Refer python code or requirements.txt for further information

# Emotion Detection

This project is an emotion detection system using a webcam and a trained deep learning model. The detected emotions trigger specific HTTP requests to control a robot.
Happy: Sends a command to move the robot forward.
Sad: Sends a command to move the robot backward.
Neutral: Sends a command to stop the robot.Refer python code or requirements.txt for further information

#Tic Tac Toe GUI Game

This project implements a simple Tic Tac Toe game with a graphical user interface (GUI) using the Tkinter library in Python. The game allows a human player to play against the computer.Refer python code or requirements.txt for further information

#Bot Control Script:

This project allows you to control a robot over a network using simple HTTP requests. You can send commands to move the robot forward, backward, left, or right.Refer python code or requirements.txt for further information

#Bot Control Quiz Script:

This project conducts a simple quiz and controls a robot over a network based on the user's answers. It sends HTTP requests to move the robot forward or backward depending on the user's responses.Refer python code or requirements.txt for further information

#Stone Paper Scissors Game with Bot Control:

This project implements a Stone Paper Scissors game using a graphical user interface (GUI) with Tkinter. The game includes functionality to control a robot over a network based on the computer's choice using HTTP requests.Refer python code or requirements.txt for further information

Speed Control :

This project allows you to control a robot over a network using keyboard inputs. It sends HTTP GET requests to move the robot forward, backward, left, or right, and allows you to adjust the speed. The script also includes functionality to stop the robot and exit the program.Refer python code or requirements.txt for further information

Bot Control Using Infrared Sensors:

This project allows you to control a robot over a network using data from infrared (IR) sensors. It sends HTTP GET requests to navigate the robot based on the sensor data received.Refer python code or requirements.txt for further information

Ultrasonic Sensor-Based Robot Control:

This project allows you to control a robot using data from an ultrasonic (US) sensor. It sends HTTP GET requests to the robot to navigate it based on the sensor data received.Refer python code or requirements.txt for further information

Face Recognition Access Control System

This project performs face recognition using the DeepFace library and grants access based on the recognized faces. If a face in the captured frame matches any of the reference images, access is granted and commands can be triggered using keyboard inputs.Refer python code or requirements.txt for further information

Traffic Light Detection Robot Control:

This project uses a neural network model to control a robot based on the detected color of a traffic light in a webcam feed. If the model detects a "GREEN" light, it sends a command to the robot to move forward. If the model detects a "RED" light, it sends a command to the robot to stop.Refer python code or requirements.txt for further information

Facial Recognition Control System

This project utilizes a deep learning model to recognize faces through a webcam feed and controls a device based on the recognized individual. It's designed to interact with a remote device by sending HTTP requests to trigger actions like moving forward or backward.Refer python code or requirements.txt for further information

Survilence Bot Control Interface

This project allows users to control a remote bot via keyboard commands while viewing a live stream from the bot's onboard camera. The application is designed for educational or testing purposes to demonstrate basic robot control and video streaming capabilities.Refer python code or requirements.txt for further information

Voice-Controlled Bot Interface

This project application allows users to control a bot by voice commands, processing responses through the Hugging Face API for decision-making. The script listens to user's commands, sends them for processing, and performs actions based on the interpreted response.Refer python code or requirements.txt for further information

Binary digits Robot Control:

This project allows you to control a robot using data from Touch sensor. It sends HTTP GET requests to the robot to navigate it based on the sensor data received.Refer python code or requirements.txt for further information

Morse code Robot Control:

This project allows you to control a robot using data from Touch sensor. It sends HTTP GET requests to the robot to navigate it based on the sensor data received.Refer python code or requirements.txt for further information

Pattern Drawing Robot:

This project allows bot to draw a pattern. It sends HTTP GET requests to the robot to navigate it based on the sensor data received.Refer python code or requirements.txt for further information

Color Sensor Robot Control:

This Python script allows you to control a robot using data from Color sensor. It sends HTTP GET requests to the robot to navigate it based on the sensor data received.Refer python code or requirements.txt for further information

How to launch MIT app inventor

1. **Install the MIT AI2 Companion App:**
   ○ Download and install the MIT AI2 Companion app from the Google Play Store or the Apple App Store on your mobile device.
2. **Open MIT App Inventor:**
   ○ Go to appinventor.mit.edu on your computer and sign in.
3. **Open Your Project:**
   ○ Select the project you want to work on or create a new one.
4. **Click "Connect":**
   ○ In the MIT App Inventor interface, click the "Connect" button at the top of the screen.
5. **Select "AI Companion":**
   ○ From the dropdown menu, select "AI Companion."
6. **Open the MIT AI2 Companion App:**
   ○ Launch the MIT AI2 Companion app on your mobile device.
7. **Scan QR Code or Enter Code:**
   ○ Use your mobile device to scan the QR code displayed on your computer screen, or enter the provided code manually in the Companion app.
8. **Connected:**
   ○ Your project will now load on your mobile device, and you can test your app in

real-time.

Mit app projects and Lessons

1.Tilt control AILite-L2

The user tilts the device forward for forward movement, backward for backward movement, and more, AILite implements the actions accordingly.

2.Speech control AILite - L2

The user gives voice commands like "move forward," "move backward," and more, which AiLite processes to perform the corresponding actions**.**