**Documentary Research on Makeblock mBot 2**

**Introduction**

Makeblock mBot2 is an educational robotics kit designed for beginners in programming and robotics. It is part of the Makeblock ecosystem, which focuses on offering tools and resources of STEM (Science, Technology, Engineering, and Mathematics) education. The mBot 2 is an advanced version of its predecessor, the original mBot, featuring improvements in design, functionality, and educational value.

**Overview of mBot 2**

**Key Features**

**1.Design and Build**

* **Chassis:** The mBot 2 features a robust and modular chassis that allows for easy assembly and disassembly.
* **Wheels and Motors:** The kit includes two motors for locomotion and is equipped with large wheels for better traction.
* **Colourful LEDs:** The mBot 2 incorporates LED lights that can be programmed to display various colours and patterns.

**2. Sensors**

The mBot 2 comes with multiple sensors, include

* **Ultrasonic Sensor:** For distance measurement and obstacle avoidance.
* **Line Tracking Sensor:** To follow lines on the ground, useful for maze navigation.
* **Light and Sound Sensors:** To detect ambient light levels and generate sound.

**3.Control and Programming**

* **Makeblock App:** Users can program mBot 2 using the Makeblock App, which supports block-based programming (like Scratch) and Python.
* **Compatibility with Arduino:** For advanced users, mBot 2 can be programmed using Arduino IDE, allowing for more sophisticated coding and custom projects.

**4.Bluetooth Connectivity**

* The mBot 2 uses Bluetooth for wireless connections to devices, making it easy to control and program from mobile devices or computers.

**5.Educational Resources**

* Makeblock provides a comprehensive curriculum with tutorials, coding challenges, and project ideas to enhance the learning experience.

**Example Projects**

**Block-Based Programming (Using Makeblock App)**

**Example 1: Basic Movement**

This simple program makes the mBot move forward for 2 seconds and then stop

**Steps:**

1.Open the Makeblock app.

2.Select the mBot 2 from the device list.

3.Create a new project with the code below

when green flag clicked

move forward at speed 100

wait 2 seconds

stop moving

**Python Code:**

from mblock import

# Initialise mBot

mbot = mBot()

# Move forward for 2 seconds

mbot.move\_foward()

Delay (2000) # Delay for 2 seconds

mbot.stop() # Stop the mBot

**Example 2: Obstacle Avoidance**

This program makes the mBot 2 move forward until it detects an obstacle, at which point it will stop and turn.

**Steps:**

1.Open the Makeblock app and create a new project.

2.Add the following blocks:

When green flag clicked

Forever

if ultrasonic distance < 20 cm then

stop moving

turn right at speed 100

wait 1 second

move forward at speed 100

end forever

**Python Code:**

from mblock import

Import time

# Initialise mBot

mbot = mBot()

while True:

Distance = mbot.read\_distance() # Read distance from ultrasonic

sensor

if distance < 20: # if an obstacle is detected within 20 cm

mbot.stop() # stop the mBot

mbot.turn\_right() # Turn right

time.sleep(1) # wait for 1 second

mbot.move\_forward() #Move forward

else:

mbot.move\_forward() # keep moving forward

time.sleep(0.1) # small delay for loop

**Example 3: Line Following**

This program uses the line tracking sensor to follow a black like on the ground.

Steps:

1. In the Makeblock App, create a new project.
2. Use the following blocks:

when green flag clicked

forever

if line tracking sensor is on the left, then

turn right at speed 100

else if line tracking sensor is on the right, then

Turn left at speed 100

else

move forward at speed 100

end if

end forever

**Python Code:**

from mblock import

Import time

# Initialise mBot

mbot = mBot()

while true:

left\_sensor = mbot.read\_line\_tracking\_left() #check left line

sensor

right\_sensor = mbot.read\_line\_tracking\_right() #check right line

sensor

if left\_sensor == 1: # if the left sensor is on the line

mbot.turn\_right() #Turn right

elif right\_sensor == 1: # if the right sensor is on the line

mbot.turn\_left() # turn left

else:

mbot.move\_forward() # move forward if no sensor is active

time.sleep(0.1) #small delay for loop

**Example 4: Sound-Activated Movement**

**Objective:** The mBot 2 will move forward when it detects a loud sound (like a clap).

Materials Needed:

* MBot 2
* Sound sensor

**Programming Steps:**

1.Setup: Connect the sound sensor to the mBot 2.

2.Code (Block Based)

when green flag clicked

Forever

if sound sensor value > 500 then

move forward at speed 100

else

stop moving

end if

end forever

**3.Code (Python)**

From mBlock import

Import time

# Initialise mBot

mbot = mBot ()

while True:

sound\_value = mbot.read\_sound() #Read sound level from sensor

if sound\_value > 500: # if sound level is above threshold

mbot.move\_forward()

else:

mbot.stop()

time.sleep(0.1) # small delay for loop

**Project 5: Remote-controlled mBot**

**Objective:** Control the mBot 2 using a smartphone or tablet via the Makeblock app.

Materials Needed:

* mBot 2
* Makeblock app installed on a smartphone/tablet

**Programming Steps:**

1. Setup: Pair the mBot 2 with smartphone using Bluetooth.
2. Code: No additional code is needed as the Makeblock app provides built-in controls. Simply use the app interface to control the mBot’s movements.

**Instructions:**

* Open the Makeblock app, connect to mBot 2, and use the joystick or buttons to move the robot forward, backward, left, or right.

**Conclusion**

These Programming examples provides a foundation for engaging with the mBot 2 through both block-based and Python programming environments. By experimenting with these codes, learners can develop their understanding of robotics, programming logic, and problem-solving skills while having fun with hands-on projects.

**DoBot Magician Lite**

**Introduction**

The Dobot Magician Lite is a lightweight, intelligent robotic arm designed primarily for educational purposes within the K12 artificial intelligence education ecosystem.it serves as a multifunctional tool that enhances students’ understanding of robotics, artificial intelligence, and mechanical systems through hands-on interaction.

**Key Features**

**1.Design and Build**

* Material: The Dobot Magician Lite is constructed from high-quality aluminium alloy,ensuring durability and stability.
* Dimensions: Compact size, allowing it to fit easily on a desktop or workstation
* Weight: The robotic arm weighs only 2.4 kg,making it portable and easy to handle in various educational settings.

**2.Technical Specifications**

* Repeatability: The arm boasts a high repeatability of 0.2 mm, ensuring precise movements.
* Payload Capacity: It can carry a payload of up to 0.25kg.
* Radius Range: The operational radius extends to 340mm, providing versatility in movement and task execution.

**3. End Effectors**

The Magician Lite is supplied with several interchangeable end tools, including:

* Suction Cup
* Soft Gripper
* Pen Holder

These tools allow for a variety of tasks, such as picking objects, drawing, and more, enhancing creative possibilities.

**4.Safety Features**

* Equipped with a collision detection function, the Magician Lite ensures a safe environment for users,particulary children.

**Control and Programming**

Magic Box Controller

* The Magician Lite is operated via the extend “Magic Box” which separates motion control algorithms from user tasks. This design simplifies programming and enhances user experience.

**Connectivity**

* Bluetooth Support: The robotic arm supports Bluetooth connectivity for wireless operation.
* Expansion Interfaces: With 12 expansion interfaces, the system is designed for maximum openness and compatibility, allowing for various hardware interactions.

**DobotBlock Software**

* The new DobotBlock platform enables students to program the robotic arm through a visual interface. By dragging and dropping coding blocks, users can easily create games, animations, and control the robot, making learning both interactive and enjoyable.

**Curriculum and Learning Approach**

**Structured Curriculum**

* The curriculum is tiered into three levels, catering to beginners through to advanced learners. This structured approach allows students to progressively build their knowledge and skills.

**Teaching Methodology**

* The program emphasizes experiential, project-based, and exploratory learning, providing a comprehensive educational experience.
* Students participate in robotics competitions, allowing them to apply their knowledge practically and develop critical skills in engineering and innovation.

**Project-Based Learning**

* The interactive and practical teaching demos are designed to foster logical thinking, hands-on skills, and creativity among students.
* The flexible nature of the Magician Lite enables it to facilitate various immersive application scenarios, helping students understand real-world applications of AI and robotics.

**Lab Test 1: Basic Movement Control**

Objective: To test the basic movement capabilities of the DOBOT Magician Lite using Python.

Setup:

* Install the necessary DOBOT Python SDK.
* Connect the DOBOT Magician Lite to your computer via USB.

Procedure:

* Import the necessary libraries and initialise the robot.
* Define a function to move the robot arm to specific coordinates.

Expected Outcome: The robotic arm should move smoothly between the specified coordinates, demonstrating its basic movement capabilities.

**Lab Test 2: Pick- and –Place Operation**

Objective: To test the pick- and –place functionality of the DOBOT Magician Lite using Python.

Setup:

* Ensure the Suction Cup or Soft Gripper is attached to the robotic arm.
* Prepare small objects (e.g., balls or blocks) to be picked and placed.

Procedure:

* Setup the initial positions for the pick-and-place operation.
* Write a function to perform the pick-and-place action.

Expected Outcome: The robotic arm should successfully pick up the object from the specified coordinates and place it at the target location.

**Lab Test 3: Drawing a Shape**

Objective: To test the drawing capabilities of the DOBOT Magician Lite using Python

Setup:

* Attach the Pen Holder and tool to the robotic arm.
* Position a piece of paper on the working surface.

Procedure:

* Initialise the pen holder and set the drawing coordinates.
* Write a function to draw a specified shape (e.g., a square).

Expected Outcome: The robotic arm should draw a complete square on the paper, demonstrating its precision and control during drawing tasks.

**Conclusion**

The DOBOT Magician Lite is an innovative educational tool that integrates robotics, artificial intelligence, and hands-on learning, its lightweight design, robust programming capabilities, and comprehensive curriculum make it an excellent choice for fostering creativity and problem-solving skills in students. By engaging with the Magician Lite,learners not only gain technical knowledge but also develop essential skills that prepare them for future challenges in technology and engineering.