**IOT BASED ROBOTIC ARM**

***A Report submitted***

***in partial fulfilment for the Degree of***

**Bachelor of Technology**

**in**

**Mechanical Engineering**

**by**

**Vedant Janardhan Bhute (2021BME027)**

**Sanveg Pandit Lokade (2021BME049)**

**Aditya Sunil Gulhane (2021BME051)**

**Under the guidance of**

**DR. ALOK MISHRA**

****

**2023-24**

**Shri Guru Gobind Singhji Institute of Engineering and Technology (SGGSIE&T), Vishnupuri, Nanded-431606, MH, India**

# **CERTIFICATE**

This is to certify that the project report entitled **IOT Based Robotic Arm** submitted by **Vedant Janardhan Bhute (2021BME027), Sanveg Pandit Lokade (2021BME049), Aditya Sunil Gulhane (2021BME051)** to Shri Guru Gobind Singhji Institute of Engineering and Technology (SGGSIE&T), Vishnupuri Nanded, in partial fulfillment for the award of the degree of **B. Tech in Mechanical Engineering** is a bona-fide record of project work carried out by him under our supervision. The contents of this report, in full or in parts, have not been submitted to any other Institution or University for the award of any degree.

Dr. Alok Mishra Dr. Praveen Somwanshi

Supervisor Co-Supervisor Department of Mechanical Engineering Department of Mechanical Engineering

Vishnupuri, Nanded HOD

08/03/2025 Dr. Saurabh Jogee

Department of Mechanical Engineering

# **EXAMINERS APPROVAL SHEET**

This is to certify that the project report entitled **IOT Based Robotic Arm** submitted by **Vedant Janardhan Bhute (2021BME027), Sanveg Pandit Lokade (2021BME049), Aditya Sunil Gulhane (2021BME051)** to Shri Guru Gobind Singhji Institute of Engineering and Technology (SGGSIE&T), Vishnupuri, Nanded, in partial fulfilment for the award of the degree of **B. Tech in Mechanical Engineering**; is here by approved for the award of degree.

Internal Examiner:

Supervisor:

Dr. Alok Mishra

# **DECLARATION**

We, declare that this project report titled **IOT Based Robotic Arm** submitted in partial fulfillment of the degree of **B. Tech in Mechanical Engineering** is a record of original work carried out by us under the supervision of **Dr. Alok Mishra,** and has not formed the basis for the award of any other degree, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

Vedant Janardhan Bhute Sanveg Pandit Lokade Aditya Sunil Gulhane

(2021BME027) (2021BME049) (2021BME051)

Vishnupuri, Nanded-431606

08/03/2025

# **ACKNOWLEDGEMENT**

We would like to express our heartfelt gratitude to the following individuals and institutions for their invaluable support and guidance throughout our project: Dr. Manesh B. Kokare, Director, SGGSIE&T, for providing us with the necessary resources and encouragement. Dr. Sourabh S. Jogee, Head of Mechanical Engineering Department, for his expert guidance and mentorship. Dr. Alok Mishra, for his expert guidance, valuable suggestions, and constant encouragement. Dr. Praveen Somwanshi, for his insightful feedback and support throughout the project. The faculty members of the Mechanical Engineering Department, for their valuable insights and suggestions. The entire SGGSIE&T family, for their unwavering support and cooperation. This project would not have been possible without the collective efforts of our team, and we are grateful for the opportunity to work together. We extend our sincere thanks to one and all of SGGSIE&T family for the completion of this document on the project report format guidelines.

Vedant Janardhan Bhute (2021BME027)

Sanveg Pandit Lokade (2021BME049)

Aditya Sunil Gulhane (2021BME051)

# **ABSTRACT**

This project presents the design and development of a robotic arm system, specifically tailored for industrial applications. The robotic arm is designed to perform tasks such as picking objects from a conveyor belt and placing them in a designated area, thereby streamlining production processes and enhancing efficiency. The system utilizes advanced sensors to detect the presence of objects on the conveyor belt, navigate the robotic arm to the correct position, and ensure precise placement. This sensor-based navigation system enables the robotic arm to adapt to changing production requirements and object shapes, sizes, and orientations. The integration of IoT technology into the robotic arm system allows for remote monitoring and control, enabling industries to optimize their production processes in real-time. This remote monitoring capability facilitates the tracking of production metrics, detection of potential issues, and implementation of corrective actions promptly, thereby minimizing downtime and maximizing productivity. The proposed robotic arm system offers several benefits, including increased efficiency, reduced labour costs, and improved accuracy. The system's scalability and compatibility with existing industrial automation systems make it an attractive solution for industries seeking to enhance their production processes. Overall, this project demonstrates the potential of robotic arm systems in transforming industrial processes and improving productivity.

# **TABLE OF CONTENTS**

## **DESCRIPTION PAGE NUMBER** CERTIFICATE I

EXAMINERS APPROVAL II

DECLARATION III

ACKNOWLEDGEMENTS IV

ABSTRACT V

LIST OF FIGURES VI

1. INTRODUCTION 1

1.1 Working principle 2

1.2 Assembly 3

* 1. Key technologies used 4

2. 3D PRINTED COMPONENTS OF ROBOTIC ARM 5

2.1 Base of Robotic Arm 5

2.2 Waist of Robotic Arm 6

* 1. Arm 1 of Robotic Arm 7

2.4 Arm 2 of Robotic Arm 8

2.5 Arm 3 of Robotic Arm 9

2.6 Gripper of Robotic Arm 10

2.6.1 Gripper Base 11

2.6.2 Gear 1 12

2.6.3 Gear 2 13

2.6.4 Links 14

2.6.5 Gripper Link 15

1. COMPONENTS OF ROBOTIC ARM 17

3.1 Arduino uno 17

3.2 Servo Motors MG996R 18

3.3 Servo Motors SG90 19

3.4 IR Sensor 20

3.5 Colour sensor 21

3.6 Jumper Wire 22

CONCLUSION 24

REFERENCES 25

APPENDIX 1 Arduino Code 25

# **LIST OF FIGURES**

## **FIGURE TITLE Page Number**

1.1 CAD Model of Robotic Arm 1

1.2 Work flow summary of procedures 3

2.1 Base of robotic Arm 6

2.2 Wrist of robotic Arm 7

2.3 Arm 1 of robotic arm 8

2.4 Arm 2 of robotic arm 9

2.5 Arm 3 of robotic arm 10

2.6 Gripper of robotic arm 11

2.7 Gripper base 12

2.8 Supporting gear of gripper 13

2.9 Power gear of gripper 14

2.10 Link of gripper 14

2.11 Gripper jaws 15

3.1 Arduino UNO 17

3.2 Servo Motor (MG996R) 18

3.3 Servo Motor (SG90) 19

3.4 IR Sensor 20

3.5 Colour Sensor 21

3.6 Jumper Wires 22

**CHAPTER 1 INTRODUCTION**

The manufacturing sector is undergoing a significant transformation with the advent of Industry 4.0. As industries strive to enhance their productivity, efficiency, and competitiveness, the need for innovative solutions that can streamline production processes has become increasingly important. Robotic arm systems have emerged as a key technology in this context, offering the potential to automate tasks, improve accuracy, and reduce labour costs. In industrial settings, robotic arm systems can perform tasks such as picking and placing objects, assembly, and inspection. However, the effectiveness of these systems depends on their ability to adapt to changing production requirements and object shapes, sizes, and orientations. The integration of advanced sensors and IoT technology can enhance the capabilities of robotic arm systems, enabling real-time monitoring and control, remote maintenance, and data analytics. This project focuses on the design and development of a robotic arm system specifically tailored for industrial applications. The proposed system utilizes advanced sensors and IoT technology to navigate the robotic arm, detect objects, and perform tasks with precision and accuracy. By leveraging the potential of robotic arm systems, this project aims to contribute to the transformation of industrial processes and improvement of productivity.

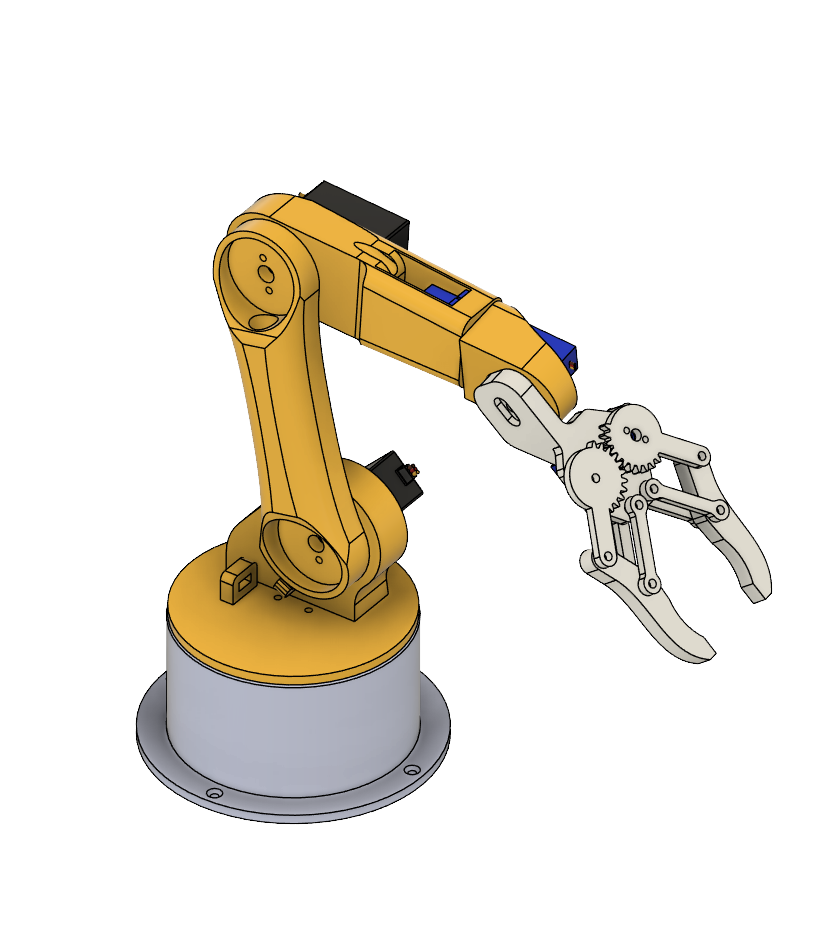


Figure 1.1 Cad Model of robotic arm

**1.1. WORKING PRINCIPLE**

The robotic arm operates in a fully automated process, picking up boxes from a conveyor belt and placing them into a cupboard based on their colour. This system integrates infrared (IR) and colour sensors to ensure accurate detection, classification, and sorting of the boxes. The process begins with the IR sensor, which detects the presence of a box on the conveyor belt. Once the box is detected, the system activates the colour sensor, which scans and identifies the colour of the box. The colour information is then processed by a microcontroller or PLC, categorizing the box into predefined groups such as red, blue, or green. Based on the detected colour, a sorting decision is made, determining the specific section of the cupboard where the box will be placed.

Once the box is identified, the robotic arm receives a pick-and-place command, instructing it to move towards the box. Using real-time sensor data and inverse kinematics, the arm calculates the precise coordinates required for accurate picking. The gripper mechanism, which can be a mechanical claw or a vacuum suction system, aligns itself with the box. After ensuring a secure grip, the arm smoothly lifts the box while compensating for movement and weight distribution. The robotic arm then follows an optimized path-planning algorithm to transport the box toward the cupboard.

As the arm reaches the cupboard, it carefully adjusts its position and moves to the predefined storage section based on the box's colour. The robotic arm gently releases the box into the appropriate shelf, ensuring proper stacking without disrupting previously placed items. After placing the box, the arm returns to its default home position near the conveyor belt, preparing for the next cycle. This process repeats continuously, maintaining efficiency and speed. The system is also equipped with real-time monitoring and error handling mechanisms. If a box is misaligned or improperly placed, feedback sensors trigger corrective actions. The robotic arm’s motion is synchronized with the conveyor belt to ensure smooth and seamless operation. Additionally, if an unrecognized or defective box colour is detected, the system can divert the box into a reject bin for further inspection.

The entire operation is powered by advanced technologies, including IR sensors for presence detection, colour sensors (RGB or vision-based) for classification, microcontrollers or PLCs for decision-making, and servo motors for precise movement control. The gripper mechanism, whether a mechanical claw, ensures secure handling of various types of boxes. This fully automated sorting system is widely used in industries such as warehouses, packaging plants, and automated storage facilities, significantly improving efficiency, accuracy, and productivity.

A close-up of a white background

AI-generated content may be incorrect.

Figure 1.2 Workflow summary of procedures

**1.2. ASSEMBLY:**

The robotic arm assembly designed for picking up boxes from a conveyor belt and placing them into a cupboard based on colour detection involves a combination of mechanical, electronic, and control components. The system consists of a robotic arm, a conveyor belt mechanism, and sensors for object detection and colour differentiation. The robotic arm is constructed with a sturdy base, a rotating joint, a vertical arm, a forearm, and an end effector (gripper), all controlled by servo motors for precise movement. The gripper mechanism, which can be either a two-finger gripper, is responsible for securely holding and placing the objects.

The conveyor belt is designed to transport boxes to the pickup position. It consists of a frame, rollers, and a rubber belt driven by a DC motor with adjustable speed control. To automate the process, sensors play a crucial role. An IR sensor is placed near the conveyor belt to detect the presence of an incoming box, while a colour sensor (TCS3200) identifies the colour of the box. This data is processed by a microcontroller (Arduino), which determines the appropriate storage section in the cupboard based on predefined colour sorting logic. The assembly process begins with mounting the robotic arm on a stable surface and securing the rotating joints and servos for smooth articulation. The conveyor system is then assembled by attaching rollers and placing the belt in position, ensuring seamless movement of the boxes. Sensors are carefully integrated to capture accurate readings, with the IR sensor positioned at the pickup zone and the colour sensor calibrated for precise detection. The electrical wiring is connected to provide power to motors and sensors, with motor drivers ensuring efficient movement control.

Once the system is powered, the operation cycle begins. When a box approaches, the IR sensor detects it, and the conveyor stops momentarily for the colour sensor to classify it. The microcontroller processes this information and instructs the robotic arm to move to the correct position. The gripper then picks up the box, lifts it, and moves to the designated cupboard section corresponding to its colour. Once the box is placed, the gripper releases it, and the arm returns to its home position, allowing the conveyor to move the next box forward. This cycle repeats continuously, ensuring an efficient, automated sorting and placement process.

After assembly, calibration is essential to fine-tune motor speeds, sensor accuracy, and gripping strength. The system is tested to ensure smooth operation, correct sorting, and precise placement of boxes. This robotic system enhances industrial automation by reducing manual labour, increasing sorting accuracy, and improving operational efficiency. With further integration of machine learning or AI, it can be enhanced for more complex sorting applications.

**1.3. Key Technologies Used:**

* IR Sensor – Detects presence of boxes.
* Colour Sensor (RGB, Vision Camera) – Identifies box colour for sorting.
* Microcontroller/PLC – Controls logic and movements.
* Servo Motors – Enables precise robotic arm movements.
* Mechanical Gripper– Handles box gripping and placement.

**CHAPTER 2 3D PRINTED COMPONENTS OF ROBOTIC ARM**

The robotic arm developed in this project utilizes 3D printed components to achieve a cost-effective and efficient design. The arm's structure, including the base, shoulder, elbow, and wrist, were designed using computer-aided design (CAD) software and printed using a fused deposition modelling (FDM) 3D printer. The 3D printed components provide a high degree of customization and flexibility, allowing for easy modifications and adjustments to be made during the design and testing phases. The use of 3D printed components also enables the robotic arm to be lightweight and compact, making it ideal for applications where space and weight are limited. Additionally, the 3D printed components are designed to be modular, allowing for easy replacement and maintenance of individual parts.

**Benefits of 3D Printed Components**:

1. Cost-effective: 3D printed components are significantly cheaper than traditional manufacturing methods.

2. Customizable: 3D printed components can be designed and printed with complex geometries and custom features.

3. Lightweight: 3D printed components can be designed to be lightweight, making them ideal for applications where weight is a concern.

4. Rapid prototyping: 3D printed components can be designed, printed, and tested quickly, allowing for rapid prototyping and development.

**2.1. Base of Robotic Arm**

The base of the robotic arm is a critical component that provides a stable foundation for the entire arm. Designed to be compact and lightweight, the base is 3D printed using a strong and durable material, such as ABS or PLA. The base features a circular platform with a diameter of 10 cm, providing a wide surface area for mounting the arm's shoulder joint. The base also includes a series of mounting holes and slots, allowing for easy attachment of the arm's control systems, sensors, and other components. Additionally, the base is designed with a low center of gravity, ensuring stability and preventing the arm from tipping over during operation. The base's compact design also allows for easy integration with other robotic systems, making it an ideal component for a wide range of robotic applications.

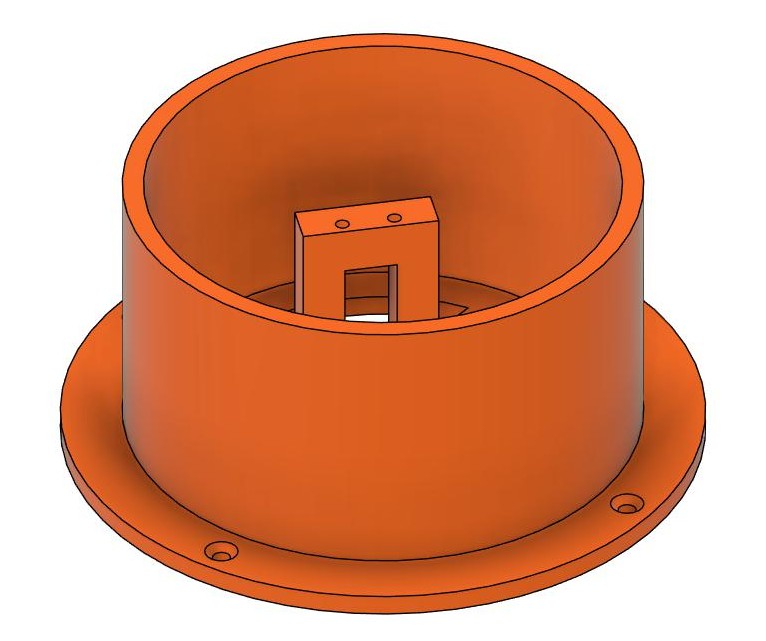
****

Figure 2.1 Base of robotic arm

**Key features of the base:**

1. Compact design: The base is designed to be compact and lightweight, making it easy to integrate with other robotic systems.
2. Mounting holes and slots: The base includes a series of mounting holes and slots, allowing for easy attachment of the arm's control systems, sensors, and other components.
3. Low centres of gravity: The base is designed with a low centre of gravity. ensuring stability and preventing the arm from tipping over during operation.

**Materials Used:**

1. PLA: Polylactic Acid (PLA) is a biodegradable and renewable material used for 3D printing the base.

**2.2. Wrist of Robotic Arm**

The wrist of the robotic arm is a critical component that provides the necessary flexibility and dexterity for precise manipulation and interaction with objects. Designed to be compact and lightweight, the wrist features a 3D printed housing that encases a high-precision servo motor, allowing for smooth and accurate rotation of the end-effector. The wrist also includes a universal joint that enables the end-effector to move freely in three-dimensional space, providing a wide range of motion and flexibility. Additionally, the wrist is equipped with a set of hall sensors and potentiometers that provide real-time feedback on the position, orientation, and velocity of the end-effector, allowing for precise control and manipulation of objects.

A close-up of a device

AI-generated content may be incorrect.

Figure 2.2 Wrist of robotic arm

**Key Features of the Wrist:**

1. Compact design: The wrist is designed to be compact and lightweight, making it easy to integrate with other robotic systems.

2. High-precision servo motor: The wrist features a high-precision servo motor that provides smooth and accurate rotation of the end-effector.

3. Universal joints: The wrist includes a universal joint that enables the end-effector to move freely in three-dimensional space.

**2.3. Arm 1 of the Robotic Arm**

Arm1 is the first segment of the robotic arm, responsible for providing the initial range of motion and positioning the subsequent arm segments. Designed to be strong and lightweight, Arm1 is constructed from high-quality, providing a robust structure that can withstand various loads and stresses. The arm is actuated by a high-torque servo motor, allowing for smooth and precise movement along the X-axis. The motor is controlled by the robotic arm's control system, which utilizes advanced algorithms and sensors to ensure accurate and repeatable movement. Arm1 also features a modular design, allowing for easy maintenance, upgrade, and customization to suit various applications and tasks.

****

Figure 2.3 ARM 1 of robotic arm

**Key Features of Arm1:**

1. Strong and lightweight: Arm1 is constructed from high-quality aluminum alloy, providing a robust structure that is also lightweight.

2. High-torque servo motor: Arm1 is actuated by a high-torque servo motor, allowing for smooth and precise movement along the X-axis.

**2.4. Arm 2 of the Robotic Arm**

Arm2 is the second segment of the robotic arm, responsible for providing additional range of motion and positioning the end-effector. Designed to be compact and precise, Arm2 is constructed from high-quality carbon fiber, providing a lightweight yet robust structure that can withstand various loads and stresses. The arm is actuated by a high-precision servo motor, allowing for smooth and accurate movement along the Y-axis. Arm2 also features a unique linkage mechanism, which enables the arm to move in a wide range of motions, including circular and elliptical paths. This feature makes Arm2 ideal for applications that require precise and complex movements, such as assembly, welding, and material handling.

A close-up of a device

AI-generated content may be incorrect.

Figure 2.4 Arm 2 of robotic arm

**Key Features of Arm2:**

1. Compact and precise: Arm2 is designed to be compact and precise, making it ideal for applications that require accurate and complex movements.
2. High-precision servo motor: Arm2 is actuated by a high-precision servo motor, allowing for smooth and accurate movement along the Y-axis.

**2.5. Arm 3 of the Robotic Arm**

Arm3 is the third segment of the robotic arm, responsible for providing the final range of motion and positioning the end-effector with high precision. Designed to be slender and agile, Arm3 is constructed from high-quality, providing a lightweight yet robust structure that can withstand various loads and stresses. The arm is actuated by a high-torque servo motor, allowing for smooth and accurate movement along the Z-axis. Arm3 also features a unique wrist mechanism, which enables the end-effector to rotate and pivot, providing a wide range of motion and flexibility. This feature makes Arm3 ideal for applications that require precise and delicate movements, such as assembly, inspection, and manipulation of small objects.

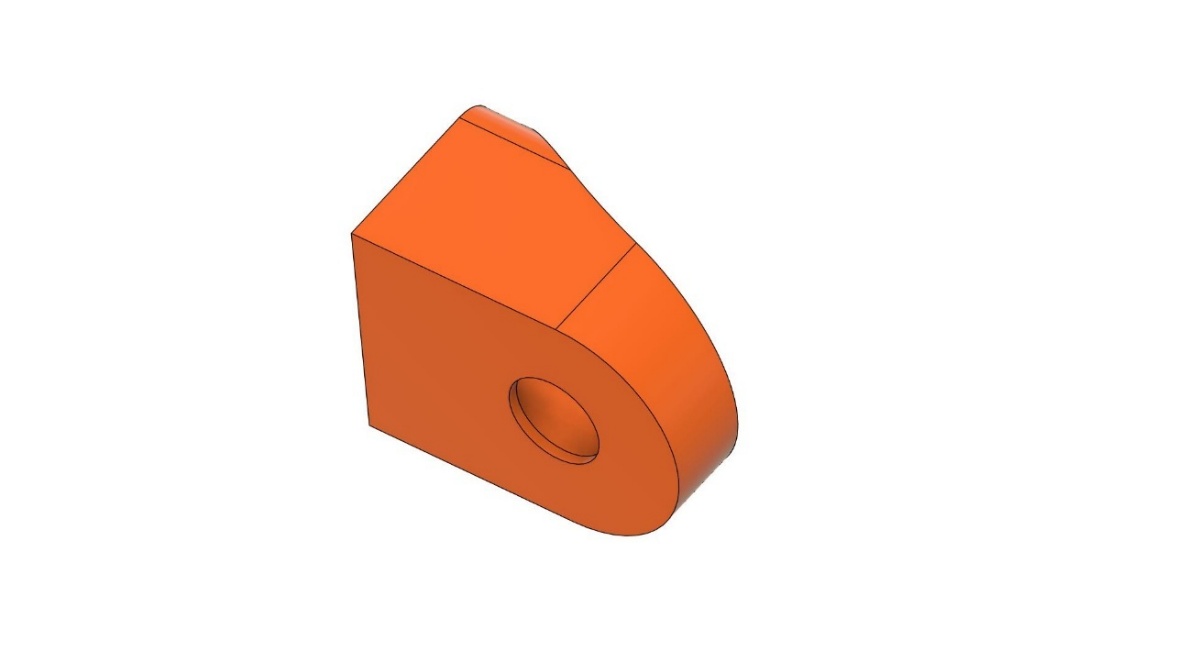


Figure 2.5 Arm 3 of robotic arm

**Key Features of Arm3:**

1. Slender and agile: Arm3 is designed to be slender and agile, making it ideal for applications that require precise and delicate movements.
2. High-torque servo motor: Arm3 is actuated by a high-torque servo motor, allowing for smooth and accurate movement along the Z-axis.
3. Unique wrist mechanism: Arm3 features a unique wrist mechanism, which enables the end-effector to rotate and pivot, providing a wide range of motion and flexibility.

**2.6. Gripper of Robotic Arm**

The gripper is the end-effector of the robotic arm, responsible for interacting with and manipulating objects. Designed to be versatile and adaptable, the gripper features a pair of parallel jaws that can be opened and closed to grasp and release objects of varying sizes and shapes. The gripper is actuated by a high-torque servo motor, allowing for precise control over the gripping force and movement. Additionally, the gripper is equipped with sensors and feedback mechanisms that enable it to detect and respond to the shape, size, and texture of objects, ensuring a secure and gentle grasp. This advanced gripper design enables the robotic arm to perform a wide range of tasks, from delicate assembly and inspection to heavy-duty material handling and manipulation. This gripper mechanism is inspired by a video [1].

**A close-up of a mechanical tool

AI-generated content may be incorrect.**

Figure 2.6 Gripper of robotic arm

**Key Features of the Gripper:**

1. Parallel jaws: The gripper features a pair of parallel jaws that can be opened and closed to grasp and release objects.

2. High-torque servo motor: The gripper is actuated by a high-torque servo motor, allowing for precise control over the gripping force and movement.

**2.6.1. Gripper Base**

The gripper base is the foundation of the robotic arm's end-effector, providing a stable and secure platform for the gripper's jaws and sensors. Designed to be compact and lightweight, the gripper base is constructed from high-quality aluminum alloy, providing a robust structure that can withstand various loads and stresses. The base features a series of mounting holes and slots, allowing for easy attachment of the gripper's jaws, sensors, and other components. Additionally, the gripper base includes a built-in bearing system, enabling smooth and precise rotation of the gripper's jaws and ensuring accurate and reliable grasping and manipulation of objects.

A black metal object with a hole

AI-generated content may be incorrect.

Figure 2.7 Gripper base

**Key Features of the Gripper Base:**

1. Compact and lightweight: The gripper base is designed to be compact and lightweight, making it ideal for applications where space and weight are limited.

3. Mounting holes and slots: The gripper base features a series of mounting holes and slots, allowing for easy attachment of the gripper's jaws, sensors, and other components.

4. Built-in bearing system: The gripper base includes a built-in bearing system, enabling smooth and precise rotation of the gripper's jaws.

**2.6.2. Gear 1**

The gripper gear is a critical component of the robotic arm's end-effector, responsible for transmitting the rotational motion of the motor to the gripper's jaws. Designed to provide high precision and reliability, the gripper gear is constructed from high-quality steel alloy, featuring a unique tooth profile and optimized gear ratio. The gear is carefully crafted to minimize backlash and ensure smooth, consistent motion, allowing the gripper to grasp and manipulate objects with precision and accuracy. Additionally, the gripper gear is designed to withstand high torque and stress loads, ensuring reliable operation in demanding applications.

**A black metal object with a hole

AI-generated content may be incorrect.**

Figure 2.8 Supporting gear of gripper

**Key Features of the Gripper Gear:**

1. Unique tooth profile: The gear features a unique tooth profile, optimized for high precision and reliability.

2. Optimized gear ratio: The gear ratio is carefully optimized to provide the ideal balance between speed and torque.

3.Minimal backlash: The gear is designed to minimize backlash, ensuring smooth, consistent motion.

**2.6.3 Gear 2**

The power transition gear is a critical component of the robotic arm's transmission system, responsible for transmitting power from the motor to the gripper gear while minimizing energy loss and maximizing efficiency. Designed to provide smooth and reliable power transmission, the power transition gear is constructed from high-quality steel alloy, featuring a unique gear profile and optimized tooth geometry. The gear is carefully crafted to ensure precise alignment and minimal backlash, allowing for seamless power transmission and efficient operation. Additionally, the power transition gear is designed to withstand high torque and stress loads, ensuring reliable operation in demanding applications.

****

Figure 2.9 Power gear of gripper

**Key Features of the Power Transition Gear:**

1. Unique gear profile: The gear features a unique profile, optimized for smooth and reliable power transmission.

2. Optimized tooth geometry: The tooth geometry is carefully optimized to minimize energy loss and maximize efficiency.

3. Precise alignment: The gear is designed to ensure precise alignment and minimal backlash.

**2.6.4. Links**

The links of the robotic arm are the structural components that connect the various joints and segments of the arm, providing a rigid and stable framework for movement and manipulation. Designed to be strong, lightweight, and compact, the links are constructed from high-quality aluminium alloy, featuring a unique tubular design that minimizes weight while maximizing strength and stiffness. The links are carefully crafted to ensure precise alignment and smooth movement, allowing the robotic arm to perform complex tasks with accuracy and precision. Additionally, the links are designed to be modular and interchangeable, enabling easy maintenance, upgrade, and customization of the robotic arm.

****

Figure 2.10 Link of gripper

**Key Features of the Links:**

1. Unique tubular design: The links feature a unique tubular design that minimizes weight while maximizing strength and stiffness.

2. Precise alignment: The links are carefully crafted to ensure precise alignment and smooth movement.

3. Modular and interchangeable: The links are designed to be modular and interchangeable, enabling easy maintenance, upgrade, and customization.

**2.6.5. Griper links**

The gripper links are the critical components that connect the gripper's jaws to the robotic arm's wrist, enabling precise and reliable grasping and manipulation of objects. Designed to be compact and lightweight, the gripper links are constructed from high-quality steel alloy, featuring a unique linkage mechanism that allows for smooth and precise movement. The links are carefully crafted to ensure precise alignment and minimal backlash, enabling the gripper to grasp and manipulate objects with accuracy and precision. Additionally, the gripper links are designed to be adjustable, allowing for easy customization of the gripper's jaw opening and closing motion.

**A pair of black plastic objects

AI-generated content may be incorrect.**

Figure 2.11 Gripper jaws

**Key Features of the Gripper Links**:

1. Compact and lightweight: The gripper links are designed to be compact and lightweight, making them ideal for applications where space and weight are limited.

2. Unique linkage mechanism: The gripper links feature a unique linkage mechanism that allows for smooth and precise movement.

3. Precise alignment: The links are carefully crafted to ensure precise alignment and minimal backlash.

4. Adjustable: The gripper links are designed to be adjustable, allowing for easy customization of the gripper's jaw opening and closing motion.

**CHAPTER 3 COMPONENTS OF ROBOTIC ARM**

**1. Arduino UNO**

The Arduino Uno is a popular microcontroller board used for building a wide range of interactive projects, from simple circuits to complex robots. Based on the ATmega328P microcontroller, the Arduino Uno features 14 digital input/output pins, 6 analog inputs, and a USB connection for programming and communication. The board is highly extensible, with a vast array of libraries and shields available for adding functionality such as Wi-Fi, Bluetooth, and motor control. With its ease of use, flexibility, and affordability, the Arduino Uno has become a go-to platform for makers, hobbyists, and professionals alike.



Figure 3.1 Arduino UNO (Ref.2)

**Key Features of the Arduino Uno:**

1. ATmega328P microcontroller: The Arduino Uno is based on the ATmega328P microcontroller, providing a powerful and flexible processing platform.

2. 14 digital input/output pins: The board features 14 digital input/output pins, allowing for a wide range of sensor and actuator connections.

3. 6 analog inputs: The Arduino Uno includes 6 analog inputs, enabling the connection of analog sensors and devices.

4. USB connection: The board features a USB connection for programming and communication.

**Technical Specifications:**

1. Microcontroller: ATmega328P

2. Clock speed: 16 MHz

3. Memory: 32 KB flash, 2 KB SRAM, 1 KB EEPROM

4. Input/output pins: 14 digital, 6 analog

**2. Servo Motor (MG996R)**

The MG996R is a high-torque, metal gear servo motor commonly used in robotics, RC vehicles, and automation projects. It operates within a voltage range of 4.8V to 7.2V, delivering a torque of up to 11 kg·cm at 6V, making it ideal for applications requiring strength and durability. With a rotational speed of 0.14 sec/60° at 6V and a range of approximately 180°, it provides precise and smooth motion control. Constructed with metal gears, the MG996R ensures longevity and resistance to wear, making it more reliable than plastic gear servos. It follows a PWM (Pulse Width Modulation) control signal, allowing seamless integration with microcontrollers like Arduino and Raspberry Pi. The servo has a standard three-wire connection: brown for ground (GND), red for power (VCC), and orange for the control signal. With a dead band width of 5 μs, it maintains stable positioning, making it highly effective for robotics, RC applications, and automation systems. Its versatility allows it to be used in robot arms, humanoid robots, RC cars, boats, planes, and DIY automation projects. Due to its high torque and metal construction, it is a preferred choice for projects that require both power and durability. Whether for hobbyists or professional engineers, the MG996R remains a reliable and cost-effective servo motor.

A small black electronic device with red wire

AI-generated content may be incorrect.

Figure 3.2 Servo Motor (MG996R) (Ref.3)

**3. Servo Motor (SG90)**

The SG90 servo motor is a small, lightweight, and widely used actuator designed for applications that require precise angular movement. It is commonly found in robotics, automation, remote-controlled vehicles, and hobbyist projects due to its affordability and ease of use. This servo motor operates on a voltage range of 4.8V to 6V, with a stall torque of 1.8 kg·cm at 4.8V and 2.2 kg·cm at 6V, making it suitable for light-duty applications. The SG90 features plastic gears, which contribute to its lightweight design but also limit its durability compared to metal-geared alternatives. Its rotation angle is typically 0° to 180°, controlled through Pulse Width Modulation (PWM) signals.

**A blue small device with a wire

AI-generated content may be incorrect.**

Figure 3.3 Servo Motor (SG90) (Ref.4)

The working principle of the SG90 servo motor is based on a closed-loop control system that utilizes a small DC motor, a gear reduction system, a control circuit, and a feedback potentiometer. The PWM signal, typically operating at 50 Hz (20 ms period), determines the angular position of the servo. A pulse width of approximately 0.5 ms corresponds to 0°, 1.5ms to 90°, and 2.4ms to 180°. The motor adjusts its position based on these signals, maintaining accuracy by continuously referencing the internal potentiometer.

Physically, the SG90 has three connection pins: the orange wire for the PWM signal, the red wire for the power supply (5V), and the brown wire for ground. This simple wiring setup allows easy interfacing with microcontrollers such as Arduino, Raspberry Pi, and ESP8266. A basic Arduino program can control the SG90 by using the Servo.h library, allowing it to rotate to specific angles with simple commands.

The SG90 offers several advantages, including its compact size, low power consumption, and ease of control. These features make it an excellent choice for beginners and professionals alike. However, there are some limitations, such as limited torque, non-continuous rotation, and the relatively low durability of its plastic gears. Despite these drawbacks, the SG90 is widely used in robotics (for arm movement and grippers), home automation (such as smart locks), animatronics, and remote-controlled aircraft.

In conclusion, the SG90 servo motor is an efficient and cost-effective solution for applications requiring controlled angular motion. Its simplicity, affordability, and compatibility with microcontrollers make it a popular choice in numerous DIY and professional projects. However, for applications requiring higher torque and durability, alternatives with metal gears or digital servo motors may be more suitable.

**4. IR Sensor**

Infrared (IR) sensors are widely used in robotic applications to detect and track objects, navigate through environments, and avoid obstacles. IR sensors emit infrared light and measure the amount of light reflected back by objects in their vicinity. By analyzing the reflected light, IR sensors can determine the distance, shape, and size of objects, as well as detect changes in temperature and motion. IR sensors are commonly used in robotic arms, grippers, and mobile robots to enable precise and reliable object detection and manipulation. They are also used in obstacle avoidance systems, line-following robots, and proximity sensing applications.



Figure 3.4 IR Sensor (Ref.5)

**Key Features of IR Sensors:**

1. Infrared light emission: IR sensors emit infrared light to detect objects.

2. Reflected light measurement: IR sensors measure the amount of light reflected back by objects.

3. Distance and shape detection: IR sensors can determine the distance, shape, and size of objects.

4. Motion and temperature detection: IR sensors can detect changes in temperature and motion.

**Technical Specifications**:

1. Wavelength: 850-1050 nm

2. Range: 0-5 meters

3. Resolution: 1-10 cm

4. Supply voltage: 5V

**5. Colour Sensor**

The color sensor is a versatile and accurate component used to detect and identify various colors in a robotic system. Based on advanced photodiode technology, the color sensor emits light in different wavelengths and measures the reflected light to determine the color of an object. With high sensitivity and accuracy, the color sensor can distinguish between subtle shades and hues, enabling the robotic system to perform complex tasks such as object recognition, sorting, and inspection. The color sensor is often used in applications such as robotic vision, quality control, and interactive installations, providing a reliable and efficient way to detect and respond to different colors.

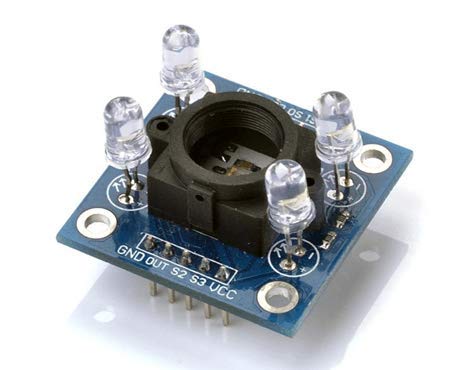


Figure 3.5 Colour Sensor (Ref.6)

**Key Features of the Color Sensor:**

1. Advanced photodiode technology: The color sensor uses advanced photodiode technology to detect and measure light in different wavelengths.

2. High sensitivity and accuracy: The color sensor has high sensitivity and accuracy, enabling it to distinguish between subtle shades and hues.

3. Compact and lightweight: The color sensor is compact and lightweight, making it easy to integrate into robotic systems.

4. Easy to use: The color sensor is easy to use, with simple communication protocols and a user-friendly interface.

**Technical Specifications:**

1. Detection range: 360-760 nm

2. Sensitivity: 10-100 lux

3. Accuracy: ±5%

4. Communication protocol: I2C, UART

**6. Jumper wire**

Jumper wires are versatile and essential components used to connect and disconnect electronic components, boards, and modules in a robotic system. These wires are designed to be flexible, durable, and easy to use, with insulated ends that prevent short circuits and ensure reliable connections. Jumper wires come in various lengths, colours, and types, including male-to-male, male-to-female, and female-to-female configurations, allowing for maximum flexibility and convenience. By using jumper wires, roboticists and makers can quickly prototype and test their circuits, adjust and modifications, and establish reliable connections between components.



Figure 3.6 Jumper Wires (Ref.7)

**Key Features of Jumper Wires**:

1. Flexible and durable: Jumper wires are designed to be flexible and durable, withstanding repeated use and handling.

2. Insulated ends: Jumper wires have insulated ends that prevent short circuits and ensure reliable connections.

3. Variety of lengths and colors: Jumper wires come in various lengths and colors, allowing for easy identification and organization.

4.Multiple configuration: Jumper wires are available in male-to-male, male-to-female, and female-to-female configurations.

**Technical Specifications:**

1. Length: 10-50 cm

2. Wire gauge: 20-24 AWG

3. Insulation: PVC or Teflon

4. Temperature range: -20°C to 80°C

**CONCLUSION**

The development and implementation of the robotic arm for automated box sorting based on colour detection successfully demonstrate the potential of robotic automation in industrial applications. The integration of IR and colour sensors allows real-time detection and classification of boxes, while the servo-controlled robotic arm efficiently picks and places them into designated sections of the cupboard. The system operates with high precision, reducing manual labour and increasing efficiency in sorting processes.

This project highlights the effectiveness of embedded systems and automation in streamlining material handling tasks. The microcontroller-based control system ensures seamless coordination between the conveyor, sensors, and robotic arm, resulting in an optimized workflow. The results show that the robotic arm can accurately sort boxes based on predefined colour categories, making it suitable for warehouses, logistics, and manufacturing industries.

**REFERENCES**

[1] Spark Plug, “How to make a Robotic Gripper in Fusion 360 | Fusion 360 Tutorial**,**” YouTube video, Mar. 24, 2022. [Online]. Available:

<https://youtu.be/ZmckF8zYbp0?si=Plfdzx33uabASA_G>

[2] Amazon, "Arduino UNO R3 image," Amazon.com. [Online]. Available: <https://amzn.in/d/7XZONeo>

[3] Amazon, "MG996R Servo Motor image," Amazon.com. [Online]. Available: <https://amzn.in/d/gYKaMKk>

[4] Amazon, "SG90 Servo Motor image," Amazon.com. [Online]. Available: <https://amzn.in/d/6SaB087>

[5] Amazon, "IR Sensor image," Amazon.com. [Online]. Available:

<https://amzn.in/d/6SaB087>

[6] Amazon, "Colour Sensor image," Amazon.com. [Online]. Available:

<https://amzn.in/d/0DDOvV4>

[7] Amazon, "Jumper wires image," Amazon.com. [Online]. Available:

<https://amzn.in/d/34JVLtN>

**Appendix 1 Arduino Code**

#include <Servo.h>

// Servo motors

Servo M0, M1, M2, M3, M4, M5, M6;

// Color sensor pins

#define S0 0

#define S1 1

#define S2 2

#define S3 3

#define sensorOut A0

// IR sensor

const int IR\_PIN = 4;

//color detection variables

int red = 0, green = 0, blue = 0;

int redCount = 0, greenCount = 0, blueCount = 0, yellowCount = 0;

void setup() {

    Serial.begin(9600);

    pinMode(S0, OUTPUT);

    pinMode(S1, OUTPUT);

    pinMode(S2, OUTPUT);

    pinMode(S3, OUTPUT);

    pinMode(sensorOut, INPUT);

    pinMode(IR\_PIN, INPUT);

    digitalWrite(S0, HIGH);

    digitalWrite(S1, LOW);

    // Attach conveyor servo

    M0.attach(5);  // Conveyor motor

    M0.write(65); // Start conveyor

}

void loop() {

    if (digitalRead(IR\_PIN) == LOW) {  // Object detected

        M0.write(0);

        delay(450);

        M0.write(90); // Stop conveyor

        delay(1000);

        String color = detectColor();

        Serial.print("Final detected color: ");

        Serial.println(color);

        M0.write(90);

        delay(5000);

        if (color == "Red") pickAndPlaceRed();

        else if (color == "Green") pickAndPlaceGreen();

        else if (color == "Blue") pickAndPlaceBlue();

        else if (color == "Yellow") pickAndPlaceYellow();

        M0.write(65); // Restart conveyor

    }

}

int getColor(int s2State, int s3State) {

    digitalWrite(S2, s2State);

    digitalWrite(S3, s3State);

    return pulseIn(sensorOut, LOW);

}

String detectColor() {

    redCount = 0;

    greenCount = 0;

    blueCount = 0;

    yellowCount = 0;

    red = getColor(LOW, LOW);

    green = getColor(HIGH, HIGH);

    blue = getColor(LOW, HIGH);

    Serial.print("Red: "); Serial.print(red);

    Serial.print(" Green: "); Serial.print(green);

    Serial.print(" Blue: "); Serial.println(blue);

    String detectedColor;

    if((red>green && red>blue) && green <= blue)

        {

            detectedColor = "Green";

            greenCount++;

        }

    else if((red<green && red<blue) && green > blue)

        {

            detectedColor = "Red";

            redCount++;

        }

    else if((blue>green && blue>red) && green >= red)

        {

            detectedColor = "Yellow";

            yellowCount++;

        }

    else if( blue < red && blue < green )

        {

            detectedColor = "Blue";

            blueCount++;

        }

    else

        {

            detectedColor = "Unknown Color";

        }

        return mostFrequentColor();

}

String mostFrequentColor() {

    if (redCount >= greenCount && redCount >= blueCount && redCount >= yellowCount) return "Red";

    else if (greenCount >= redCount && greenCount >= blueCount && greenCount >= yellowCount) return "Green";

    else if (blueCount >= redCount && blueCount >= greenCount && blueCount >= yellowCount) return "Blue";

    else if (yellowCount >= redCount && yellowCount >= greenCount && yellowCount >= blueCount) return "Yellow";

    else return "Unknown Color";

}

void pickAndPlaceRed() {

  Serial.begin(9600);

  Serial.println("");

  Serial.println("Initializing motors...");

  M2.attach(7);

  M2.write(120);

  Serial.println("M2 homed");

  delay(500);

  M2.detach();

  Serial.println("M2 detached");

  M1.attach(6);

  M1.write(0);

  Serial.println("M1 homed");

  delay(500);

  M1.detach();

  Serial.println("M1 detached");

  M3.attach(8);

  M3.write(75);

  Serial.println("M3 homed");

  delay(500);

  M3.detach();

  Serial.println("M3 detached");

  M5.attach(10);

  M5.write(110);

  Serial.println("Gripper homed");

  delay(2500);

  M5.detach();

  Serial.println("M5 detached");

  M2.attach(7);

  M2.write(90);

  Serial.println("Arm1 down");

  delay(2500);

  M2.detach();

  Serial.println("M2 detached");

  M6.attach(11);

  M6.write(90);

  Serial.println("Gripper open");

  delay(2500);

  M6.detach();

  Serial.println("M6 detached");

  M5.attach(10);

  M5.write(20);

  Serial.println("Gripper down");

  delay(2500);

  M5.detach();

  Serial.println("M5 detached");

  M6.attach(11);

  M6.write(3);

  Serial.println("Gripper close");

  delay(2500);

  M2.attach(7);

  M2.write(135);

  Serial.println("Gripper up");

  delay(2500);

  M2.detach();

  Serial.println("M2 detached");

  M1.attach(6);

  M1.write(180);

  Serial.println("Move to rack");

  delay(2500);

  M1.detach();

  Serial.println("M1 detached");

  M3.attach(8);

  M3.write(120);

  Serial.println("Arm2 down");

  delay(2500);

  M3.detach();

  Serial.println("M3 detached");

  M5.attach(10);

  M5.write(120);

  Serial.println("Gripper up");

  delay(2500);

  M5.detach();

  Serial.println("M5 detached");

  M2.attach(7);

  M2.write(112);

  Serial.println("Arm1 down");

  delay(2500);

  M2.detach();

  Serial.println("M2 detached");

  M3.attach(8);

  M3.write(100);

  Serial.println("Arm2 up");

  delay(2500);

  M3.detach();

  Serial.println("M3 detached");

  M2.attach(7);

  M2.write(100);

  Serial.println("Arm1 down");

  delay(2500);

  M2.detach();

  Serial.println("M2 detached");

  M6.write(30);

  Serial.println("Gripper open");

  delay(2500);

  M6.detach();

  Serial.println("M6 detached");

  M2.attach(7);

  M2.write(135);

  Serial.println("Arm1 down");

  delay(2500);

  M2.detach();

  Serial.println("M2 detached");

  M5.attach(10);

  M5.write(110);

  Serial.println("Gripper homed");

  delay(2500);

  M5.detach();

  Serial.println("M5 detached");

  M1.attach(6);

  M1.write(0);

  Serial.println("M1 homed");

  delay(2500);

  M1.detach();

  Serial.println("M1 detached");

  Serial.println("Red box placed!");

}

void pickAndPlaceGreen() {

  Serial.begin(9600);

  Serial.println("");

  Serial.println("Initializing motors...");

  M2.attach(7);

  M2.write(120);

  Serial.println("M2 homed");

  delay(500);

  M2.detach();

  Serial.println("M2 detached");

  M1.attach(6);

  M1.write(0);

  Serial.println("M1 homed");

  delay(500);

  M1.detach();

  Serial.println("M1 detached");

  M3.attach(8);

  M3.write(75);

  Serial.println("M3 homed");

  delay(500);

  M3.detach();

  Serial.println("M3 detached");

  M5.attach(10);

  M5.write(110);

  Serial.println("Gripper homed");

  delay(2500);

  M5.detach();

  Serial.println("M5 detached");

  M2.attach(7);

  M2.write(90);

  Serial.println("Arm1 down");

  delay(2500);

  M2.detach();

  Serial.println("M2 detached");

  M6.attach(11);

  M6.write(90);

  Serial.println("Gripper open");

  delay(2500);

  M6.detach();

  Serial.println("M6 detached");

  M5.attach(10);

  M5.write(20);

  Serial.println("Gripper down");

  delay(2500);

  M5.detach();

  Serial.println("M5 detached");

  M6.attach(11);

  M6.write(3);

  Serial.println("Gripper close");

  delay(2500);

  M2.attach(7);

  M2.write(135);

  Serial.println("Gripper up");

  delay(2500);

  M2.detach();

  Serial.println("M2 detached");

  M1.attach(6);

  M1.write(165);

  Serial.println("Move to rack");

  delay(2500);

  M1.detach();

  Serial.println("M1 detached");

  M3.attach(8);

  M3.write(120);

  Serial.println("Arm2 down");

  delay(2500);

  M3.detach();

  Serial.println("M3 detached");

  M5.attach(10);

  M5.write(120);

  Serial.println("Gripper up");

  delay(2500);

  M5.detach();

  Serial.println("M5 detached");

  M2.attach(7);

  M2.write(115);

  Serial.println("Arm1 down");

  delay(2500);

  M2.detach();

  Serial.println("M2 detached");

  M3.attach(8);

  M3.write(98);

  Serial.println("Arm2 up");

  delay(2500);

  M3.detach();

  Serial.println("M3 detached");

  M2.attach(7);

  M2.write(100);

  Serial.println("Arm1 down");

  delay(2500);

  M2.detach();

  Serial.println("M2 detached");

  M6.write(30);

  Serial.println("Gripper open");

  delay(2500);

  M6.detach();

  Serial.println("M6 detached");

  M2.attach(7);

  M2.write(135);

  Serial.println("Arm1 down");

  delay(2500);

  M2.detach();

  Serial.println("M2 detached");

  M1.attach(6);

  M1.write(0);

  Serial.println("M1 homed");

  delay(500);

  M1.detach();

  Serial.println("M1 detached");

  Serial.println("Green Box Placed!");

}

void pickAndPlaceBlue() {

  Serial.begin(9600);

  Serial.println("");

  Serial.println("Initializing motors...");

  M2.attach(7);

  M2.write(120);

  Serial.println("M2 homed");

  delay(500);

  M2.detach();

  Serial.println("M2 detached");

  M1.attach(6);

  M1.write(0);

  Serial.println("M1 homed");

  delay(500);

  M1.detach();

  Serial.println("M1 detached");

  M3.attach(8);

  M3.write(75);

  Serial.println("M3 homed");

  delay(500);

  M3.detach();

  Serial.println("M3 detached");

  M5.attach(10);

  M5.write(110);

  Serial.println("Gripper homed");

  delay(2500);

  M5.detach();

  Serial.println("M5 detached");

  M2.attach(7);

  M2.write(90);

  Serial.println("Arm1 down");

  delay(2500);

  M2.detach();

  Serial.println("M2 detached");

  M6.attach(11);

  M6.write(90);

  Serial.println("Gripper open");

  delay(2500);

  M6.detach();

  Serial.println("M6 detached");

  M5.attach(10);

  M5.write(20);

  Serial.println("Gripper down");

  delay(2500);

  M5.detach();

  Serial.println("M5 detached");

  M6.attach(11);

  M6.write(3);

  Serial.println("Gripper close");

  delay(2500);

  M2.attach(7);

  M2.write(135);

  Serial.println("Gripper up");

  delay(2500);

  M2.detach();

  Serial.println("M2 detached");

  M1.attach(6);

  M1.write(180);

  Serial.println("Move to rack");

  delay(2500);

  M1.detach();

  Serial.println("M1 detached");

  M5.attach(10);

  M5.write(170);

  Serial.println("Gripper up");

  delay(2500);

  M5.detach();

  Serial.println("M5 detached");

  M3.attach(8);

  M3.write(145);

  Serial.println("Arm2 down");

  delay(2500);

  M3.detach();

  Serial.println("M3 detached");

  M2.attach(7);

  M2.write(90);

  Serial.println("Arm1 down");

  delay(2500);

  M2.detach();

  Serial.println("M2 detached");

  M3.attach(8);

  M3.write(120);

  Serial.println("Arm2 up");

  delay(2500);

  M3.detach();

  Serial.println("M3 detached");

  M2.attach(7);

  M2.write(80);

  Serial.println("Arm1 down");

  delay(2500);

  M2.detach();

  Serial.println("M2 detached");

  M3.attach(8);

  M3.write(110);

  Serial.println("Arm2 up");

  delay(2500);

  M3.detach();

  Serial.println("M3 detached");

  M2.attach(7);

  M2.write(75);

  Serial.println("Arm1 down");

  delay(2500);

  M2.detach();

  Serial.println("M2 detached");

  M6.write(30);

  Serial.println("Gripper open");

  delay(2500);

  M6.detach();

  Serial.println("M6 detached");

  M2.attach(7);

  M2.write(105);

  Serial.println("Arm1 down");

  delay(2500);

  M2.detach();

  Serial.println("M2 detached");

  M3.attach(8);

  M3.write(140);

  Serial.println("Arm2 up");

  delay(2500);

  M3.detach();

  Serial.println("M3 detached");

  M2.attach(7);

  M2.write(135);

  Serial.println("Arm1 down");

  delay(2500);

  M2.detach();

  Serial.println("M2 detached");

  M1.attach(6);

  M1.write(0);

  Serial.println("M1 homed");

  delay(2500);

  M1.detach();

  Serial.println("M1 detached");

  Serial.println("Blue box placed!");

}

void pickAndPlaceYellow() {

  Serial.begin(9600);

  Serial.println("");

  Serial.println("Initializing motors...");

  M2.attach(7);

  M2.write(120);

  Serial.println("M2 homed");

  delay(500);

  M2.detach();

  Serial.println("M2 detached");

  M1.attach(6);

  M1.write(0);

  Serial.println("M1 homed");

  delay(500);

  M1.detach();

  Serial.println("M1 detached");

  M3.attach(8);

  M3.write(75);

  Serial.println("M3 homed");

  delay(500);

  M3.detach();

  Serial.println("M3 detached");

  M5.attach(10);

  M5.write(110);

  Serial.println("Gripper homed");

  delay(2500);

  M5.detach();

  Serial.println("M5 detached");

  M2.attach(7);

  M2.write(90);

  Serial.println("Arm1 down");

  delay(2500);

  M2.detach();

  Serial.println("M2 detached");

  M6.attach(11);

  M6.write(90);

  Serial.println("Gripper open");

  delay(2500);

  M6.detach();

  Serial.println("M6 detached");

  M5.attach(10);

  M5.write(20);

  Serial.println("Gripper down");

  delay(2500);

  M5.detach();

  Serial.println("M5 detached");

  M6.attach(11);

  M6.write(3);

  Serial.println("Gripper close");

  delay(2500);

  M2.attach(7);

  M2.write(135);

  Serial.println("Gripper up");

  delay(2500);

  M2.detach();

  Serial.println("M2 detached");

  M1.attach(6);

  M1.write(165);

  Serial.println("Move to rack");

  delay(2500);

  M1.detach();

  Serial.println("M1 detached");

  M5.attach(10);

  M5.write(170);

  Serial.println("Gripper up");

  delay(2500);

  M5.detach();

  Serial.println("M5 detached");

  M3.attach(8);

  M3.write(145);

  Serial.println("Arm2 down");

  delay(2500);

  M3.detach();

  Serial.println("M3 detached");

  M2.attach(7);

  M2.write(95);

  Serial.println("Arm1 down");

  delay(2500);

  M2.detach();

  Serial.println("M2 detached");

  M3.attach(8);

  M3.write(120);

  Serial.println("Arm2 up");

  delay(2500);

  M3.detach();

  Serial.println("M3 detached");

  M2.attach(7);

  M2.write(80);

  Serial.println("Arm1 down");

  delay(2500);

  M2.detach();

  Serial.println("M2 detached");

  M6.write(40);

  Serial.println("Gripper open");

  delay(2500);

  M6.detach();

  Serial.println("M6 detached");

  M2.attach(7);

  M2.write(105);

  Serial.println("Arm1 down");

  delay(2500);

  M2.detach();

  Serial.println("M2 detached");

  M3.attach(8);

  M3.write(150);

  Serial.println("Arm2 up");

  delay(2500);

  M3.detach();

  Serial.println("M3 detached");

  M2.attach(7);

  M2.write(135);

  Serial.println("Arm1 down");

  delay(2500);

  M2.detach();

  Serial.println("M2 detached");

  M1.attach(6);

  M1.write(0);

  Serial.println("M1 homed");

  delay(2500);

  M1.detach();

  Serial.println("M1 detached");

  Serial.println("Yellow box placed!");

}