CHAPTER 1

INTRODUCTION

1.1 Introduction

Normally, sorting of the objects is done by manually. It consists of 4 integrated stations called distribution, testing, processing and handling. Old sorting method uses a set of inductive, capacitive and optical sensors do differentiate object color in the testing station. Handling is done by using a programmed manipulator. No vision capability exists in the system to improve its performance and flexibility. In this case, there is a possibility of minor error which will affect the accuracy in sorting. Also, for huge systems, time and manpower required will very high. Automated systems can be used to remove such human errors and also it saves time and money.

Elementary conveyor belts were used since the 19th century. In 1892, Thomas Robins began a series of inventions relating to conveyor systems, which led to the development of a conveyor belt used for carrying coal, ores and other products . Recently conveyor belt systems are not only used in mining industries but also applied in cement industries, food factories, power plant, and production industries etc. So, it is essential equipment for in house material transportation today.

The purpose of this project is to design and implement a color-based product sorting machine using Arduino Nano, servo motors, and a color sensor. In today's fast-paced industrial environment, efficient sorting of products based on their color has become a critical need. The ability to automate this process can significantly enhance productivity, reduce errors, and streamline operations in various industries such as manufacturing, packaging, and quality control.

The main components of our sorting machine include an Arduino Nano microcontroller, servo motors for actuation, and a color sensor as a key element. The Arduino Nano serves as the brain of the system, controlling the overall operation and providing the necessary interface between the hardware components. Servo motors are employed to manipulate the sorting mechanism, allowing precise and controlled movement to divert products based on their color.

- **RGB COLOUR MODEL:**

The RGB color model is an additive color model in which red, green and blue light are added together in various ways to reproduce a broad array of colors. The secondary colors of RGB model cyan, magenta and yellow, are formed by the mixture of two of the primaries and the exclusion of the third as shown in figure . Combination of primary colors to generate secondary color are as follows:

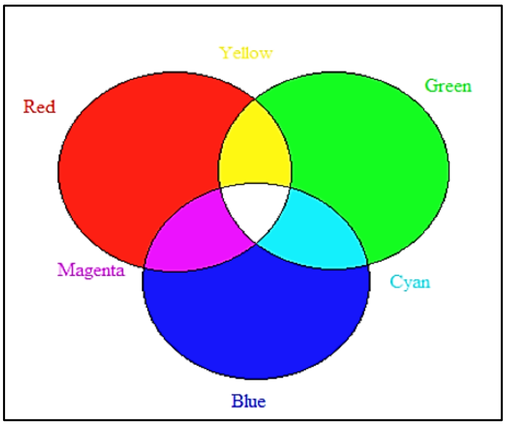


Fig. 1.1 RGB Color Model

1. Red + Green=Yellow
2. Green + Blue=Cyan
3. Blue + Red=Magenta

When red, green and blue are added together in equal proportion then white light is generated.

Colour offers leading information about objects. Several color models are available such as RGB color model, CMYK color model, HSV model. RGB model uses primary colors. CMYK represents secondary colors Cyan(C), Magenta (M), Yellow (Y) and Black (K). In proposed method color image is analyzed in RGB model point of view.

The color sensor plays a pivotal role in the sorting process by accurately detecting and distinguishing different colors. It utilizes advanced color sensing technology to capture the spectral characteristics of objects and convert them into meaningful data. By leveraging the capabilities of the color sensor, our sorting machine can effectively identify and sort products based on their predetermined color criteria.
This project aims to showcase the practical implementation and functionality of a color-based product sorting machine. By combining hardware components such as Arduino Nano, servo motors, and a color sensor, we seek to achieve a reliable and efficient system that can handle a wide range of products with varying colors.

Through this project, we intend to demonstrate the potential benefits of automation in the sorting process, highlighting the significance of accurate and efficient color-based sorting. Furthermore, we will evaluate the performance of our sorting machine in terms of accuracy, speed, and overall effectiveness.
In the following sections, we will delve into the methodology, system design, implementation details, and results of our color-based product sorting machine. We will also discuss the challenges encountered during the project and provide insights into possible future improvements. By the end of this report, we aim to present a comprehensive overview of our project, showcasing its potential applications and contributions to the field of automated product sorting.

One notable advantage of our color-based product sorting machine is its vertical configuration, which offers distinct benefits over traditional conveyor belt-based systems. Unlike conveyor belt systems that typically operate in a horizontal plane, our vertical setup optimizes the utilization of space and allows for a more compact footprint. This vertical orientation enables efficient sorting within a smaller area, making it ideal for applications where space is limited. Additionally, the vertical design facilitates a gravity-assisted sorting mechanism, reducing the reliance on complex conveyor belts and minimizing the risk of product jams or misalignments. By leveraging the advantages of a vertical system, our color-based product sorting machine offers improved spatial efficiency, enhanced sorting accuracy, and increased flexibility in various industrial settings.

1.2 Motivation

The main intention of this project is to elaborate and simplify how different products manufactured in a factory can be put on a single hopper for its proper distribution. earlier there used to be a robotic arm for object sorting , but there are certain limitations of robotic arm such as, first parameter is time constraint, robotic arm takes more time to move from one position to other position as robotic arm works on pick and place methodology. While picking up the object and placing it to desired location takes a more time. so, sorting will take a more time, and other parameter is cost,the cost of production of project using robotic arm is more as compared to robotic arm sorting of object using vertical axis system is more cost effective.

The idea was introduced to reduce the human efforts and the time required for the sorting. But the first implemented sorting mechanism 'the robotic arm' was just another whole machine designed to a single objective i.e. sorting and reduced human efforts.

The motivation behind undertaking this project stems from the growing need for efficient and accurate sorting solutions in various industries. Manual sorting processes are time-consuming, prone to errors, and often struggle to meet the demands of high-volume production environments. Recognizing this challenge, our team was inspired to develop a color-based product sorting machine that harnesses automation and advanced sensing technology. By creating a reliable and precise sorting system, we aim to alleviate the limitations of manual sorting, enhance productivity, and contribute to the overall optimization of industrial processes. Our motivation lies in providing a practical and cost-effective solution that improves efficiency, reduces human error, and ultimately enhances the competitiveness of industries that rely on accurate product sorting.

1.3 Background

Previously there have been systems designed for sorting and color detecting device which sorts colors mainly violet, black, white successfully using IR sensor (3 for 3 colors), However more accuracy can be achieved by using RGB led, using TCS3200 color sensor. This enables us to identify a wider range of colors capturing its real time RGB values. we expanded this idea to color detection with servo motors and hopper for sorting of object and using a color sensor in place of IR sensor.

The Arduino Nano microcontroller serves as the core of our sorting machine, providing a versatile and programmable platform for controlling the overall operation. We utilize servo motors for actuation, enabling precise and controlled movements to divert products based on their detected color. The central element of our sorting machine is the color sensor, which employs advanced spectral sensing technology to accurately identify and distinguish different colors.

Our project builds upon existing research and developments in the field of automated sorting systems. By incorporating a vertical design, we aim to optimize space utilization and enhance sorting efficiency. The gravity-assisted sorting mechanism further improves the reliability and effectiveness of the system, minimizing the risk of product jams or misalignments

1.4 Project Specification :

1.4.1 Input Used

In the project, the inputs play a crucial role in the color-based product sorting machine. The inputs consist of the products being sorted, which are represented by small, oval-shaped objects with a smooth surface and uniform size. These colored objects are specifically chosen to showcase the sorting capabilities of the machine

.

The RGB color reading from the colored objects is captured by the TCS3200 color sensor and serves as an input signal to the Arduino Nano microcontroller. The color sensor analyzes the spectral characteristics of the objects and provides the corresponding color information to the Arduino. This information is then used by the Arduino to make decisions regarding the sorting mechanism.

The color reading input signal received by the Arduino enables it to classify and determine the appropriate destination for each colored object based on its color. By processing the color data, the Arduino instructs the servo motors to actuate the sorting mechanism accordingly, diverting each object to its respective designated container or pathway.

It's worth noting that the color sensor plays a vital role in accurately detecting and differentiating the colors of the objects. The sensor's output is a critical input to the Arduino, allowing it to make real-time decisions and execute the sorting process effectively.

In summary, the inputs in this project consist of small, oval-shaped objects with a smooth surface and uniform size. The RGB color readings obtained from these objects serve as input signals to the Arduino Nano microcontroller. This input data enables the Arduino to determine the appropriate sorting action and control the servo motors accordingly.

1.4.2 Output

The output actions in the color-based product sorting machine entail a sequence of coordinated mechanical movements to sort the colored objects based on their detected colors. These actions are facilitated by servo motors and other mechanical components.

The primary output action involves the horizontal displacement of a semi-circular disc, which selectively retrieves one colored object at a time from the input area. The disc positions the object beneath the color sensor, ensuring its accurate alignment for color analysis. Upon positioning, the color sensor transmits a signal to the Arduino Nano microcontroller, relaying the color information of the object. The Arduino processes this signal and determines the appropriate sorting destination based on the color classification. Subsequently, the Arduino dispatches a signal to the servo motor responsible for controlling the slider mechanism. The servo motor activates, inducing horizontal displacement of the attached slider in front of the designated container or pathway corresponding to the detected color. This precise motion guarantees precise routing of the sorted object.

Following slider alignment, the semi-circular disc releases the sorted object onto the slider, allowing it to traverse towards the designated container or pathway. Once the object reaches its predetermined location, it undergoes collection or is directed for further processing as dictated by the specific operational requirements.

In summary, the output actions in this project involve the horizontal motion of the semi-circular disc to position the colored object under the color sensor, followed by the servo motor's displacement of the slider to channel the object towards the designated container or pathway for sorting. These synchronized movements ensure accurate sorting of the colored objects based on their detected colors.

CHAPTER 2
LITERATURE SURVEY

2.1 HISTORY

[1] Vishnu R. Kale, V. A. Kulkarni, "Object sorting system using robotic arm", Vol. 2, Issue 7, July 2013

In this paper a Fully functional sorter machine can be implemented by using a structure of parallel and independent channels in order to increase the overall throughput which results with a forecasted performance. The project can work successfully and separates different objects using sensors. The sensor handling systems which drive the pick and place robot to pick up the object and place it into its designated place can work if accurately designed. There are two main steps in sensing part, objects detection and recognition. The system can successfully perform handling station task, namely pick and place mechanism with help of sensor. Thus a cost effective Mechatronics system can be designed using the simplest concepts and efficient result can be observed.

[2] J. D. Gavade , P. K. Kharat, S. K. Laga " Cost Effective Approach for Object Sorting", International Journal of Computer Applications (0975 – 8887) Volume 52– No.16, August 2012

In this paper, An object sorting system for domestic/industrial control has developed using the concepts of Image Processing, Robotics Mechanism and parallel communication without help of DSP processor. The model developed is user friendly. Hence for fast manipulation the algorithm implemented in the MATLAB is suitable for our sorting problem. Result of sorting the object may not work for 100 percent but it is working for nearly 90 to 94% in case of our algorithm. It can be improved by increasing the accuracy and it depends on atmospheric factors.

[3] Prof. D. B. Rane1, Gunjal Sagar S.2, Nikam Devendra V.3, Shaikh Jameer U.4, "Automation of Object Sorting Using an Industrial Roboarm and MATLAB Based Image Processing", International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 5, Issue 2, February 2015)

In this paper, the sorting machine sorts the objects depending upon the colors of the objects successfully with the help of the rob arm and MATLAB program in image processing. The USB webcam serves as an eye of the system which captures the real time image of the objects. The rob arm picks the faulty quality object and places it at predefined place, while good quality object continues its motion on conveyor belt and finally drops into object carrier system. In this LCD displays the object count with the status about the quality of the object. The servomotors used in the rob arm plays the vital role as control movement of the rob arm wholly depends control signal given to servo motor. Hence to operate the system accurately the synchronization between IR sensors, dc motors of the conveyor belt and rob arm is very essential.

[4] Vishnu R. Kale, 2V. A. Kulkarni," automation of object sorting system using pick & place robotic arm & image processing", Proceedings of 3rd IRAJ International Conference, 5th January 2014, Mumbai, India. ISBN: 978-93-82702-51-1.

In this paper, Fully functional sorter machine is implemented by using a structure of parallel and independent channels in order to increase the overall throughput which results with a forecasted performance. There are two main steps in sensing part, objects detection and recognition. The system is successfully perform handling station task, namely pick and place mechanism with help of sensor. Thus a cost effective Mechatronics system can be designed using the simplest concepts and efficient result is observed.

2.2 METHODOLOGY

The methodology employed in this project encompasses a systematic approach to design, develop, and implement the color-based product sorting machine. The methodology consists of several key phases, including system design, component selection, prototyping, and system integration. Each phase is carefully executed to ensure the successful realization of the sorting machine. This section outlines the step-by-step process followed throughout the project, detailing the methodologies and techniques employed at each stage. By following this structured methodology, we aim to achieve an efficient, accurate, and reliable color-based product sorting machine that meets the specified requirements and objectives of the project.

- **System design**: This phase involves designing the color-based product sorting machine, including its overall architecture, mechanical components, electronic circuitry, and software system.
- **Component selection**: In this phase, suitable components are carefully chosen for the sorting machine, such as the Arduino Nano microcontroller, servo motors, color sensor, and other necessary hardware components.
- **Prototyping**: The prototyping phase involves building a working prototype of the sorting machine, integrating the selected components, and testing its functionality and performance.
- **System integration**: This phase focuses on integrating all the components, ensuring their proper interconnection, and fine-tuning the system to achieve optimal operation and accurate sorting.

CHAPTER 3

SYSTEM DESIGN AND IMPLEMENTATION

3.1 System Design Block Diagram:

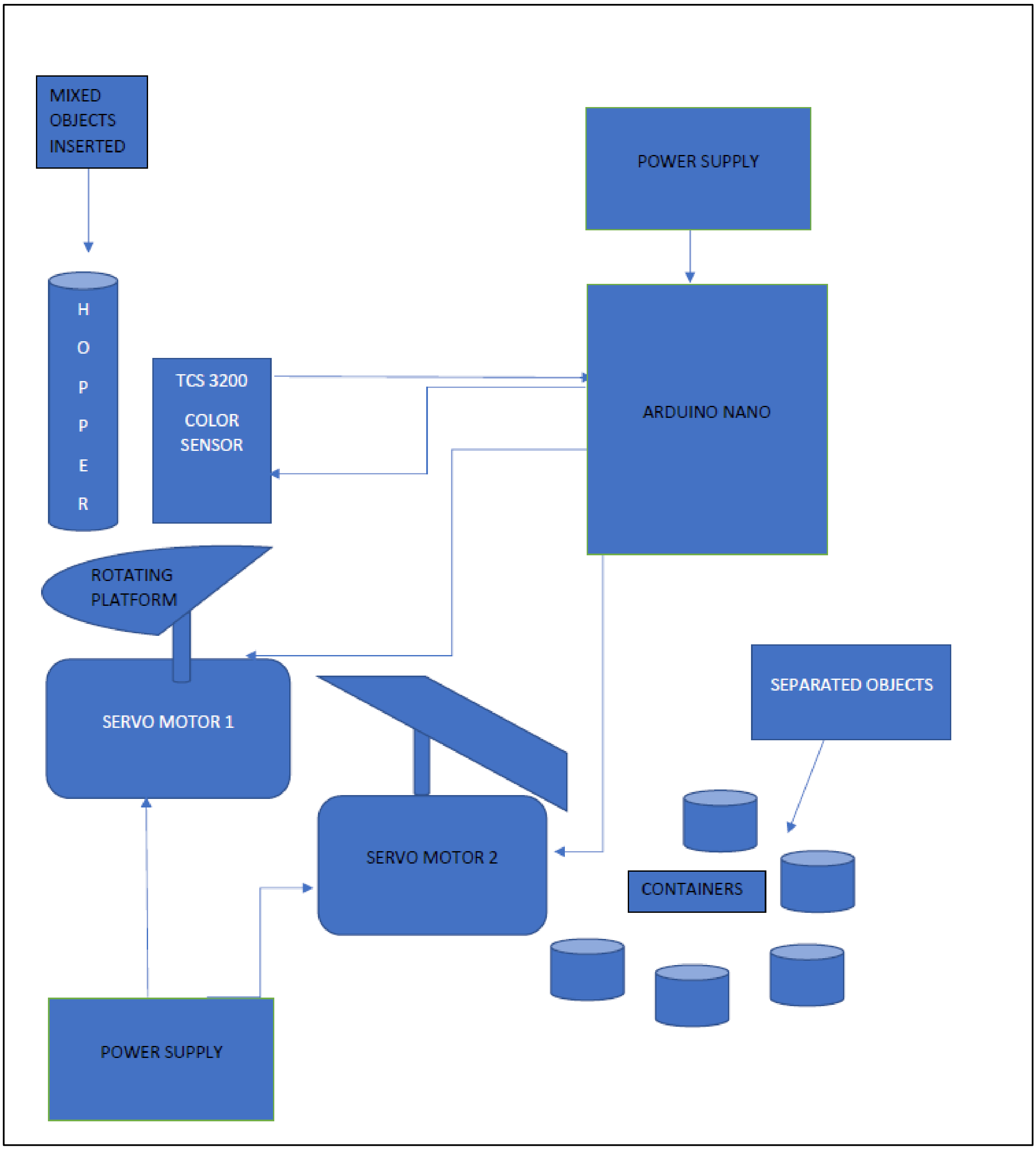


Fig.3.1 system block diagram

3.1.1: Block Diagram Description:

- **Colour Sensor Block**: This block represents the TCS3200 colour sensor module, which detects the colours of the objects being sorted. It provides colour data to the Arduino Nano for processing.

- **Arduino Nano Block**: This block represents the Arduino Nano microcontroller board, which acts as the central control unit. It receives colour data from the color sensor, processes it, and controls various components accordingly.
- **Servo Motor Block:** This block represents the servo motor responsible for horizontal movement. It controls the slider mechanism that navigates the items into different containers based on their color.
- **Slider Block: T**his block represents the slider mechanism, which moves horizontally to transport the items to their designated containers. It is controlled by the servo motor and receives signals from the Arduino Nano.
- **Rotating Disc Block**: This block represents the rotating semi-circular disc, which is responsible for shifting the product from the hopper under the color sensor and into the slider. It ensures precise positioning for color detection and subsequent sorting.
- **Hopper Block**: This block represents the hopper, which is a hollow tube-like structure that stores the stack of products. It acts as the reservoir from which the rotating disc picks up the items for color detection and sorting.
- **Power Supply Block**: This block represents the 5V DC power supply, which provides power to the Arduino Nano, color sensor, servo motor, rotating disc, and hopper. It ensures the proper functioning of all components.
- **Data Flow:**

1. The color sensor block captures the color data of the objects placed on the rotating disc.
2. The color data is sent to the Arduino Nano block for analysis and decision-making.
3. Based on the color readings, the Arduino Nano block sends control signals to the servo motor and slider blocks.
4. The servo motor block controls the horizontal movement of the slider to direct the items into the appropriate containers.

- **Power Flow:**

  The power supply block provides the necessary 5V DC power to all components, including the Arduino Nano, color sensor, servo motor, rotating disc, and hopper.

3.2 Hardware Implementation:

3.2.1 Circuit Diagram:

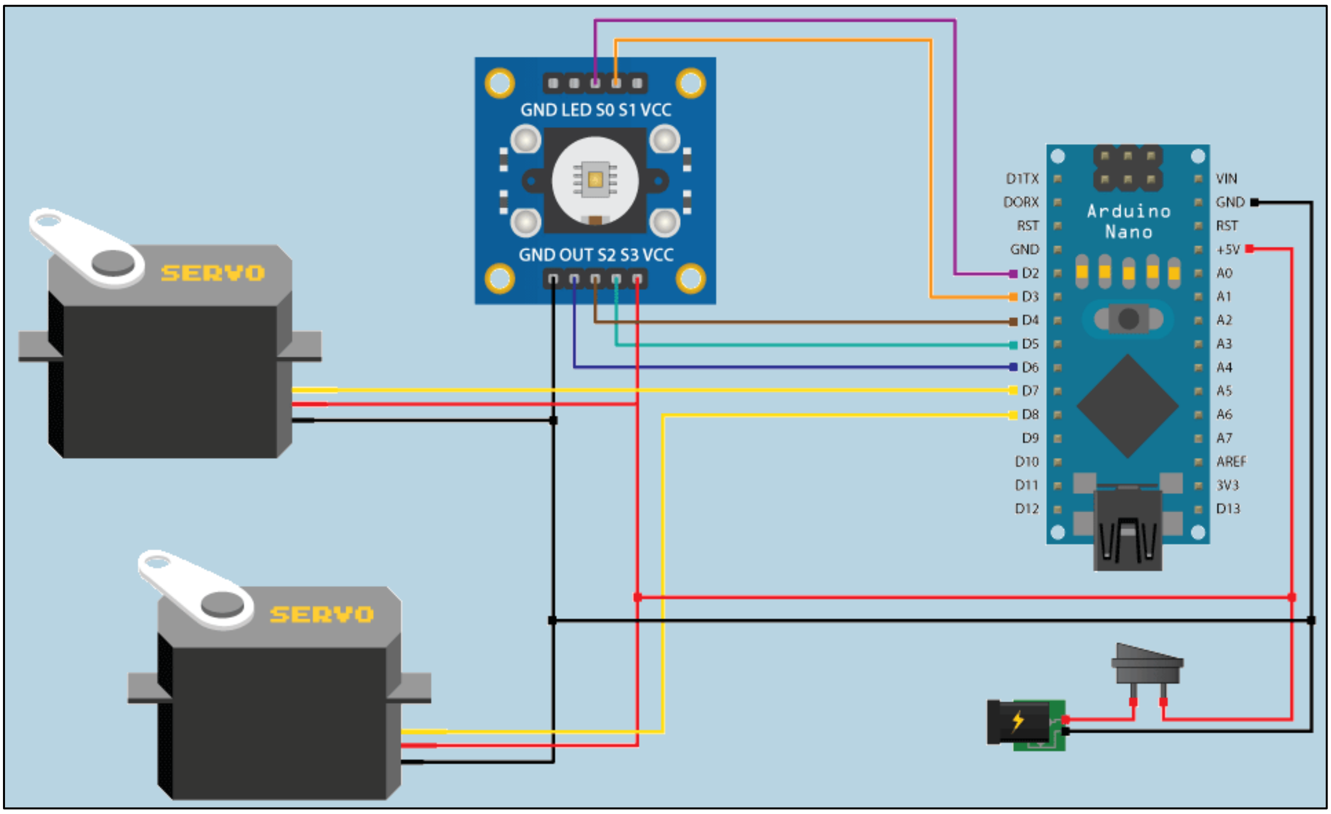


Figure 3.1: Circuit Diagram

- **Circuit Diagram Description:**

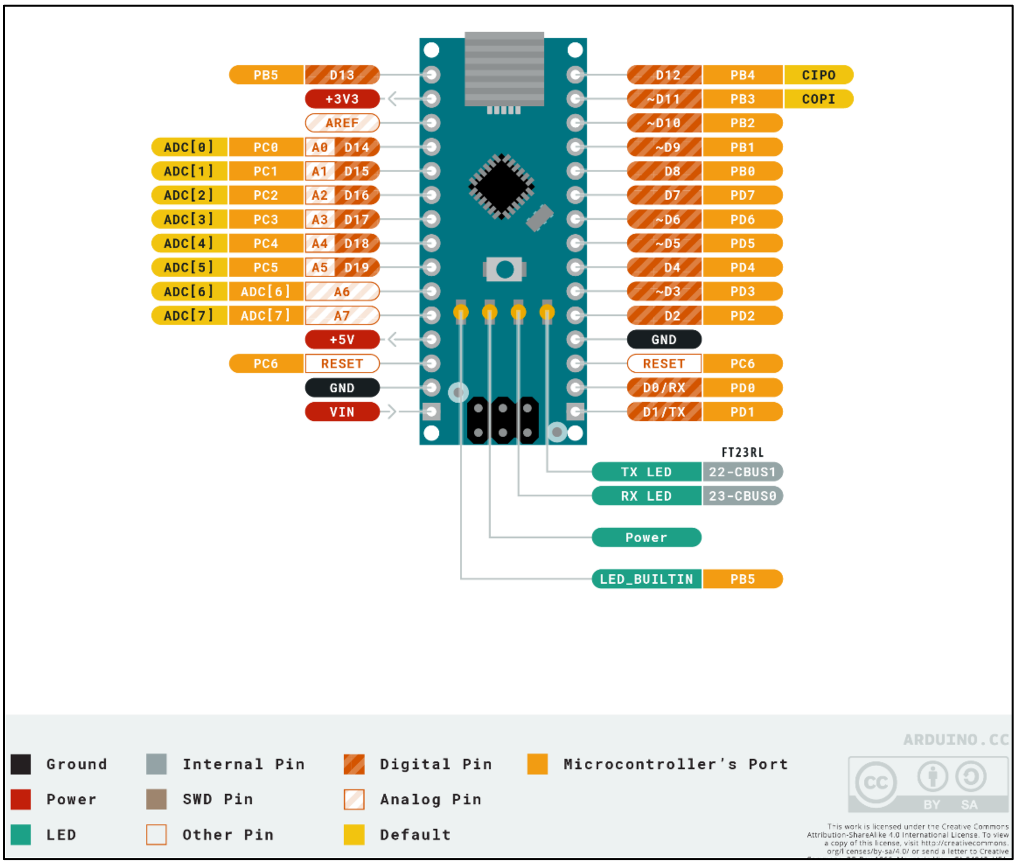
The circuit diagram consists of the following components and connections:

- TCS3200 Color Sensor: The TCS3200 color sensor is used for color detection. It is connected to the Arduino Nano as follows:
- The S0 pin of the TCS3200 color sensor is connected to digital pin D2 of the Arduino Nano.
- The S1 pin of the TCS3200 color sensor is connected to digital pin D3 of the Arduino Nano.
- The S2 pin of the TCS3200 color sensor is connected to digital pin D4 of the Arduino Nano.
- The S3 pin of the TCS3200 color sensor is connected to digital pin D6 of the Arduino Nano.
- The VCC (power) and GND (ground) pins of the TCS3200 color sensor are connected to the 5V power supply and ground respectively.
- Servo Motors: The circuit includes two servo motors, denoted as Servo Motor 1 and Servo Motor 2. They are connected to the Arduino Nano as follows:
- The signal (control) wire of Servo Motor 1 is connected to digital pin D7 of the Arduino Nano.
- The signal (control) wire of Servo Motor 2 is connected to digital pin D8 of the Arduino Nano.
- The power (usually red) wire and ground (usually black or brown) wire of both servo motors are

connected to the 5V power supply and ground respectively.

- Power Supply: The circuit is powered by a 5V DC power supply, which provides power to the Arduino Nano, TCS3200 color sensor, and servo motors. The power supply is connected to the VCC (power) and GND (ground) pins of these components.

3.2.2 Arduino Nano:



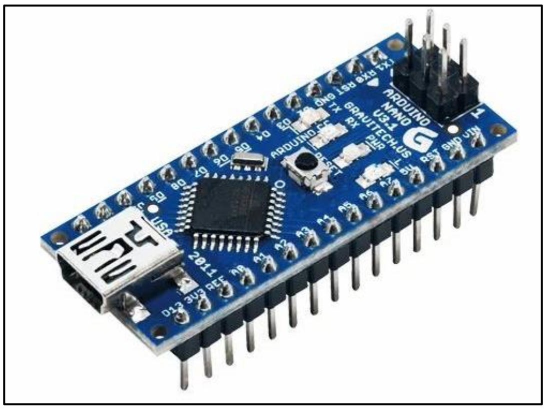


Fig.3.2 Arduino Nano & Pinout Diagram

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one.

The Atmega328 has 32 KB, (also with 2 KB used for the bootloader. The Atmega328 has 2 KB of SRAM and 1 KB of EEPROM.

Each of the 14 digital pins on the Nano can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 Ma and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.

External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it’s off.

The Nano has 8 analog inputs, each of which provide 10 bits of resolution (i.e., 1024 different values). By default, they measure from ground to 5 volts, though is it possible to change the upper end of their range using the analogReference() function. Analog pins 6 and 7 cannot be used as digital pins. Additionally, some pins have specialized functionality:

I2C: A4 (SDA) and A5 (SCL). Support I2C (TWI) communication using the Wire library (documentation on the Wiring website).

There are a couple of other pins on the board:

AREF. Reference voltage for the analog inputs. Used with analogReference().

Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source. When using an external power supply, such as an AC-to-DC adapter (wall-wart) or battery, the adapter can be connected by plugging a suitable connector into the board's power jack. For battery power, the leads can be inserted into the Gnd and Vin pin headers of the POWER connector.

The Arduino Nano can operate on an external supply ranging from 6 to 20 volts. However, it's important to note that if the supplied voltage is less than 7V, the 5V pin may provide less than five volts, which can lead to unstable board operation. Conversely, using a voltage higher than 12V may cause the voltage regulator to overheat, potentially damaging the board. The recommended voltage range for stable operation is 7 to 12 volts.

The microcontroller board based on the Atmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

1. **VIN.** The input voltage to the Atmega 16board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
2. **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
3. **3V3.** A 3.3-volt supply generated by the on-board regulator. Maximum current draw is 50 Ma.
4. **GND.** Ground pins.
5. **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the Atmega8U2 USB-to-TTL Serial chip .
6. **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
7. **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.
8. **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Atmega 16language.
9. **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

3.2 .3 PCB Layout of Arduino Nano:

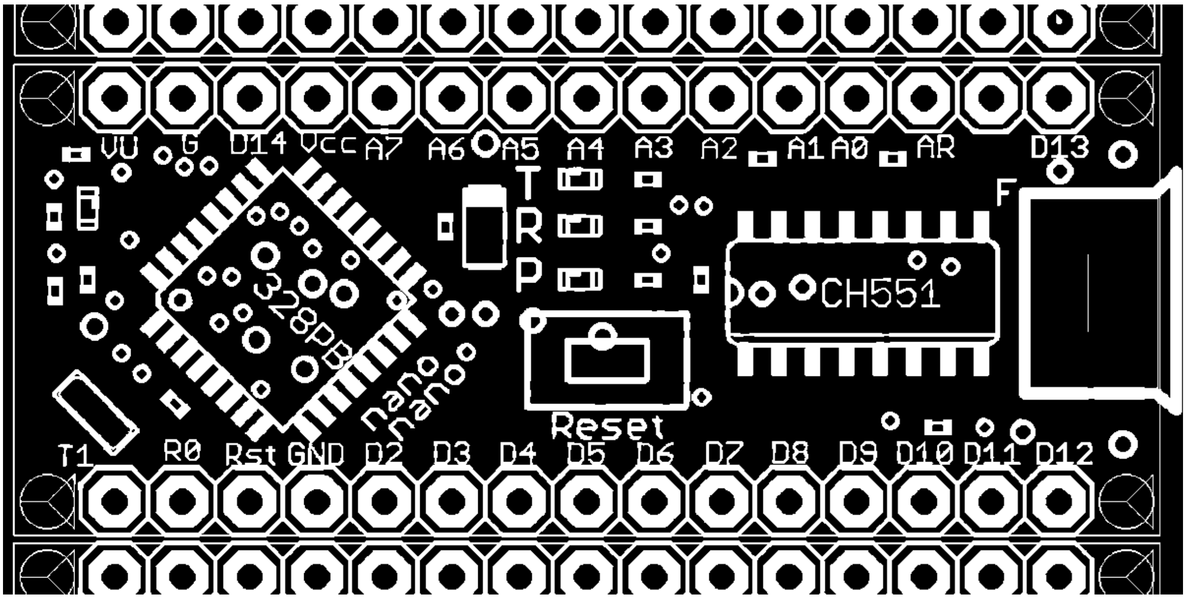


Fig.3.3 Arduino PCB Layout

3.2.4 Power supply

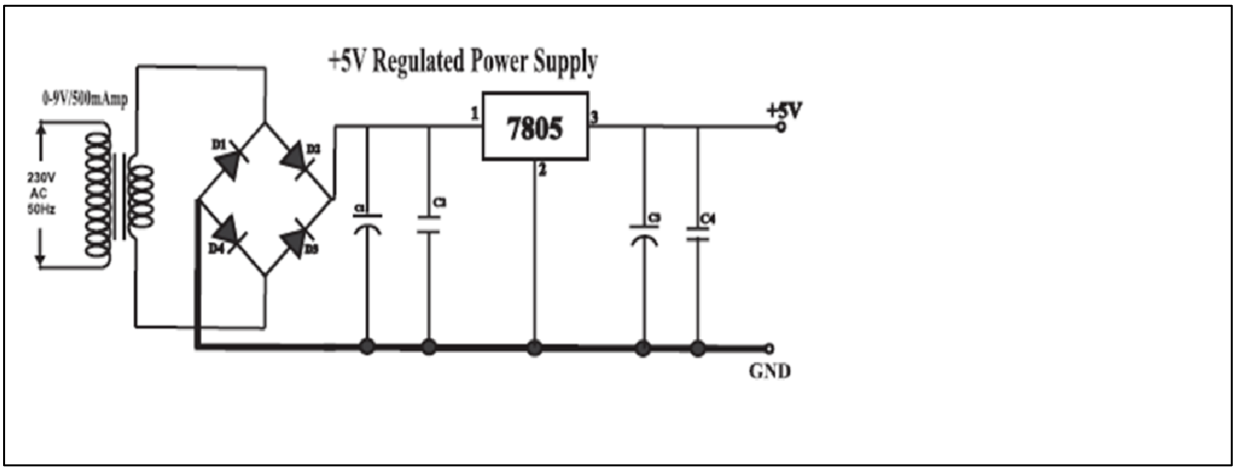


Fig. 3.4: Circuit Diagram-Power Supply

- The basic step in the designing of any system is to design the power supply required for that system. The steps involved in the designing of the power supply are as follows,

  1) Determine the total current that the system sinks from the supply.

  2) Determine the voltage rating required for the different components.

- The bridge rectifier and capacitor i/p filter produce an unregulated DC voltage which is applied at the I/P of 7805.
- The minimum dropout voltage is 2v for IC 7805, the voltage applied at the input terminal should be at least 7 volts .
- C1 (1000 µf / 65v)is the filter capacitor .

- C2,C4 (0.1uF ceramic),C3 (220uF/25V electrolyte capacitor) is to be connected across the regulator to improve the transient response of the regulator.
- Assuming the drop out voltage to be 2 volts, the minimum DV voltage across the capacitor C1 should be equal to 7volts (at least).

Fig. 5V Power supply Adapter

The system requires a 5V DC power supply to operate effectively. Here are the specifications for the 5V power supply from fig.:

1. **Voltage:** The power supply should provide a stable output voltage of 5 volts. This ensures that the system components receive the necessary voltage level for their proper operation.
2. **Current Capacity**: The power supply has a 2 Ampere current rating sufficient to meet the power requirements of the system. The total current draw of the system depends on the components used, including the Arduino Nano (40 mA), servo motors, color sensor, and other peripherals. It is important to calculate the total current consumption of the system and choose a power supply with an appropriate current capacity to handle the load.
3. **Connector Type**: The power supply should have a compatible connector to connect to the power input of the system. Common connector types include barrel connectors or USB connectors. Ensure that the power supply's connector matches the power input jack or connection point on the system.
4. **Stability and Regulation**: The power supply should provide stable and regulated output voltage to prevent fluctuations that could affect the operation of the system. It should have built-in protections such as overvoltage, overcurrent, and short circuit protection to ensure the safety of the components and the system itself.

5. **Efficiency**: It is desirable to choose a power supply with high efficiency to minimize power loss and reduce energy consumption. A more efficient power supply converts electrical energy to usable power with minimal waste heat.

3.2.5: Color Sensor (TCS3200):

The TCS3200 and TCS3210 programmable color light-to-frequency converters that combine configurable silicon photodiodes and a current-to-frequency converter on a single monolithic CMOS integrated circuit. The output is a square wave (50% duty cycle) with frequency directly proportional to light intensity (irradiance).

● High-Resolution Conversion of Light Intensity to Frequency

● Programmable Color and Full-Scale Output Frequency

● Communicates Directly With a Microcontroller

● Single-Supply Operation (2.7 V to 5.5 V)

● Power Down Feature

● Nonlinearity Error Typically 0.2% at 50 kHz

● Stable 200 ppm/°C Temperature Coefficient

● Low-Profile Lead (Pb) Free and RoHS Compliant Surface-Mount Package

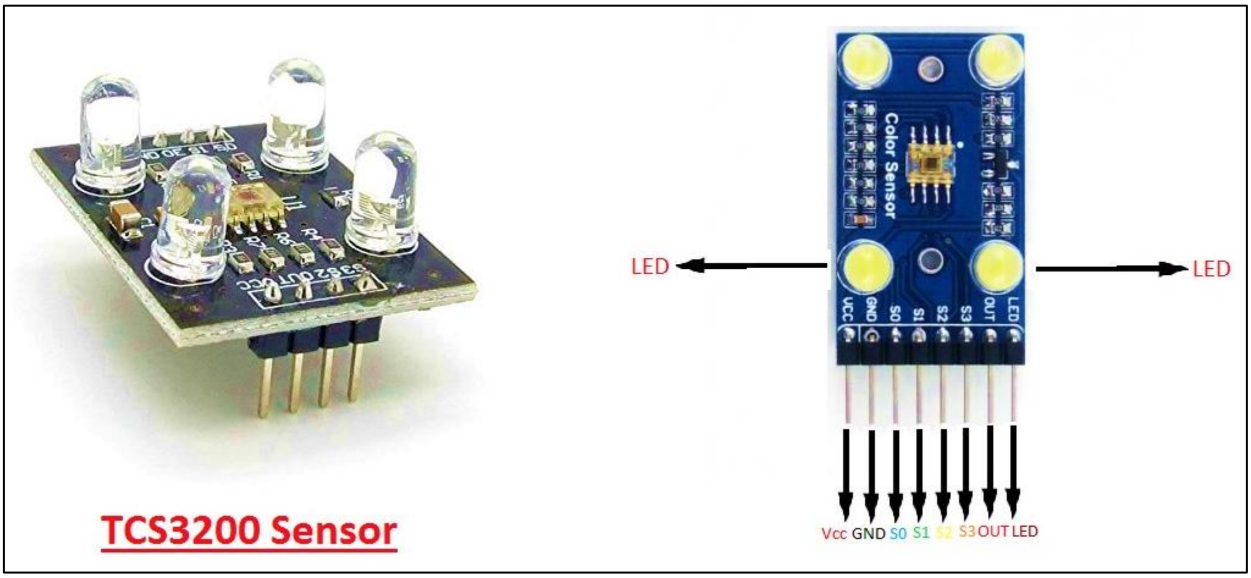


Fig.3.5 TCS 3200 Color sensor

A color sensor depends on a photodiode which measure the power reflected by the protest for a red, green and blue light source. Because of the same it will help in detecting the specified color. There is total 8 pins in a color sensor.pin no 7 and 8 are selection pins, pin no 1 and pin 2 are frequency scaling pin. Pin 6 is output.

Each photodiodes have an 8*8 matrix which results into 16 of green filter, 16 or red filter, 16 of blue filter and the rest 16 for clear filter when no colour will be detected.

As shown in figure on microscopic level one can see the square boxes inside the eye on sensor. These square boxes are arrays of RGB matrix. Each of these boxes contain three sensors, One is for sensing RED light intensity, One is for sensing GREEN light intensity and the last in for sensing BLUE light intensity.

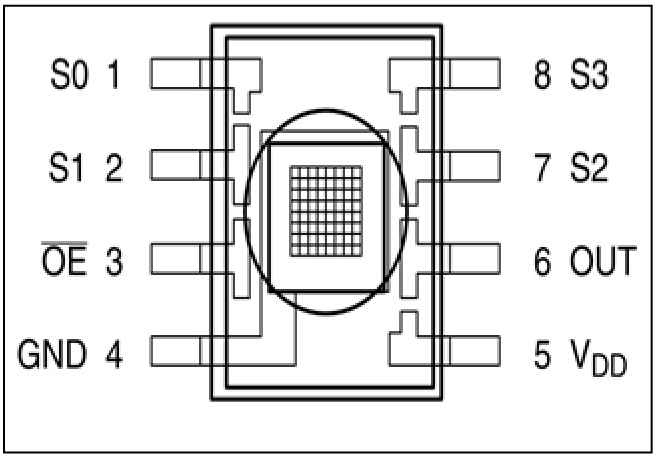


Fig.3.6 TCS3200 pinout diagram

Each of sensor arrays in these three arrays are selected separately depending on the requirement. Hence it is known as programmable sensor. The module can be featured to sense the particular color and to leave the others. It contains filters for that selection purpose. There is a fourth mode called ‘no filter mode’ in which the sensor detects white light.

3.2.6 Servo Motor (MG90S):

MG90S is a micro servo motor with metal gear. This small and lightweight servo comes with high output power, This Servo motors operates from 4.8V to 6.5V, the higher the voltage higher the torque one can achieve, but most commonly it is operated at +5V. All servo motors can rotate only from 0° to 180° due to their gear arrangement. or prefer for a 0° to 360° motor or modify the motor to make a full circle.

Fig. 3.7 MG90s Servo Motor

It is a compact and high-performance servo motor widely used in robotics and automation applications. It is known for its precision, reliability, and efficient operation. Here are specifications of the MG90S servo motor:

- **Physical Characteristics:**

1. Dimensions: The MG90S servo motor has a compact size, typically measuring around 22.8mm x 12.2mm x 28.5mm.
2. Weight: It is lightweight, weighing approximately 13.4 grams.

- **Pin Description:**

  - **Wire Number** **Wire Colour Description**
  - 1 Brown Ground wire connected to the ground of system
  - 2 Red Powers the motor typically +5V is used
  - 3 Orange PWM signal is given in through this wire to drive the motor

a. Gear Type: The motor incorporates metal gears, which provide durability and strength for smooth and precise movement.
b. Mounting: The motor features standard mounting holes, allowing for easy integration into various mechanical systems.

- **Operating Specifications:**

a. Operating Voltage: The MG90S servo motor operates within a voltage range of 4.8V to 6.0V, making it compatible with common power sources such as batteries or DC power supplies.
b. Torque: It offers a high torque output, typically ranging from 1.8kg/cm to 2.2kg/cm. This torque rating enables it to handle moderate loads and perform tasks requiring forceful movements.
c. Speed: The motor achieves a speed of approximately 0.10 seconds per 60 degrees rotation, which allows for quick and responsive movements.
d. Rotation Angle: It provides a rotation angle of up to 180 degrees, allowing for a wide range of motion and flexibility in various applications.
e. Control Signal: The MG90S servo motor utilizes pulse width modulation (PWM) control signals, commonly generated by microcontrollers such as Arduino or Raspberry Pi, to precisely position the motor shaft.

- **Electrical Interface:**

i. Signal Connector: The motor typically comes with a three-wire interface, including power (VCC), ground (GND), and the control signal (PWM) wire. This standard interface simplifies the connection to control devices.
ii. Signal Protocol: The control signal follows a pulse width modulation (PWM) protocol, where the duration of the pulse determines the desired position or angle of the motor shaft.

- **Applications:**

a. Robotics: The MG90S servo motor is commonly used in robotics applications, including robotic arms, grippers, and humanoid robots, where precise positioning and movement control are required.
b. RC Vehicles: It is suitable for radio-controlled (RC) cars, airplanes, boats, and other RC vehicles, providing reliable and accurate control for steering, throttle, and other mechanical functions.
c. Automation: The motor finds applications in automation systems, such as automated door locks, camera pan-tilt mechanisms, and automated home appliances, where it can control movements and positions with accuracy.

The MG90S servo motor combines compact size, high torque, and reliable performance, making it a versatile choice for a wide range of applications. Its compatibility with standard control signals and ease of integration makes it a popular choice for hobbyists and professionals alike.

Table No. 3.1 Servo Features

MG-90S Features
• Operating Voltage: 4.8V to 6V (Typically 5V)
• Stall Torque: 1.8 kg/cm (4.8V)
• Max Stall Torque: 2.2 kg/cm (6V)
• Operating speed is 0.1s/60° (4.8V)
• Gear Type: Metal
• Rotation : 0°-180°
• Weight of motor : 13.4gm
• Package includes gear horns and screws

3.2.7 Micro Servo 9G (SG90):

The Micro Servo 9G is lightweight, high-quality and lightning-fast. The servo is designed to work with almost all the radio control systems. It is with excellent performance brings you to another horizon of flight. The SG90 mini servo with accessories is perfect for R/C helicopter, plane, car, boat and truck use.

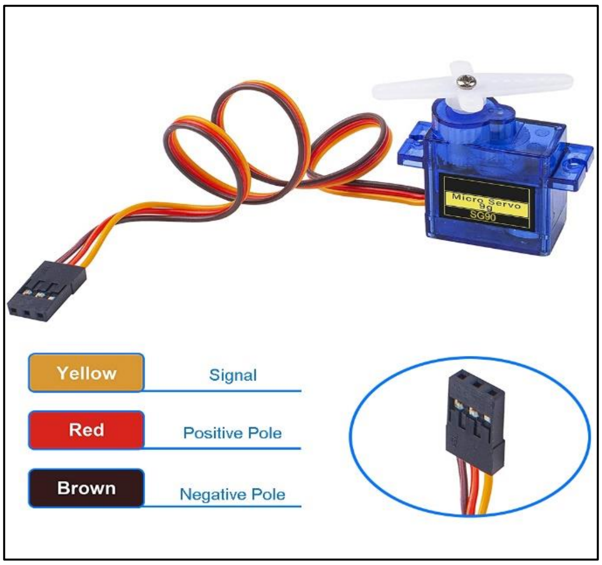


Fig. 3.8 Micro Servo 9G

The micro servo 9G is a compact and lightweight servo motor equipped with nylon gears, ensuring durability and smooth operation. Designed for precise control in small-scale robotics, RC vehicles, and hobby projects, it operates on a voltage range of 4.8V to 6V. With reliable and accurate motion control, this servo motor is an excellent choice for applications that require compact size, lightweight design.

Features:

- Required Pulse: 3-5 Volt Peak to Peak Square Wave
- Operating Voltage: 4.8-6.0 Volts
- Operating Temperature Range: -10 to +60 Degree C
- Operating Speed (4.8V): 0.12sec/60 degrees at no load
- Operating Speed (6.0V): 0.10sec/60 degrees at no load
- Stall Torque (4.8V): 1.8kg/cm
- Stall Torque (6.0V): 2.4kg/cm
- 360 Modifiable: Yes
- Bearing Type: Ball Bearing
- Gear Type: Nylon Gears
- Connector Wire Length: 12"
- Dimensions: 22x11.5x27mm
- Weight: 11g

3.2.8: Jumper Cables:

Jumper cables, also known as jumper wires or Dupont wires, are essential components in electronics prototyping and circuit connections. They consist of insulated wires with male or female connectors at both ends, allowing for easy and temporary electrical connections between electronic components, breadboards, or development boards. These cables come in various lengths and colors, providing flexibility and convenience in wiring and troubleshooting circuits.

Fig.3.9 Jumper cables

- The use of a plastic frame, acrylic plates, and a sun board sheet in constructing the body of this system offers several advantages. The plastic frame provides a lightweight yet sturdy structure that ensures stability and durability. Acrylic plates, known for their transparency and impact resistance, allow for visibility of internal components while protecting them from external elements. Additionally, the sun board sheet, with its rigidity and versatility, offers excellent mounting and customization options for attaching various components. The combination of these materials results in a robust, functional, and aesthetically pleasing body for the system, facilitating easy assembly, maintenance, and integration of the different elements.

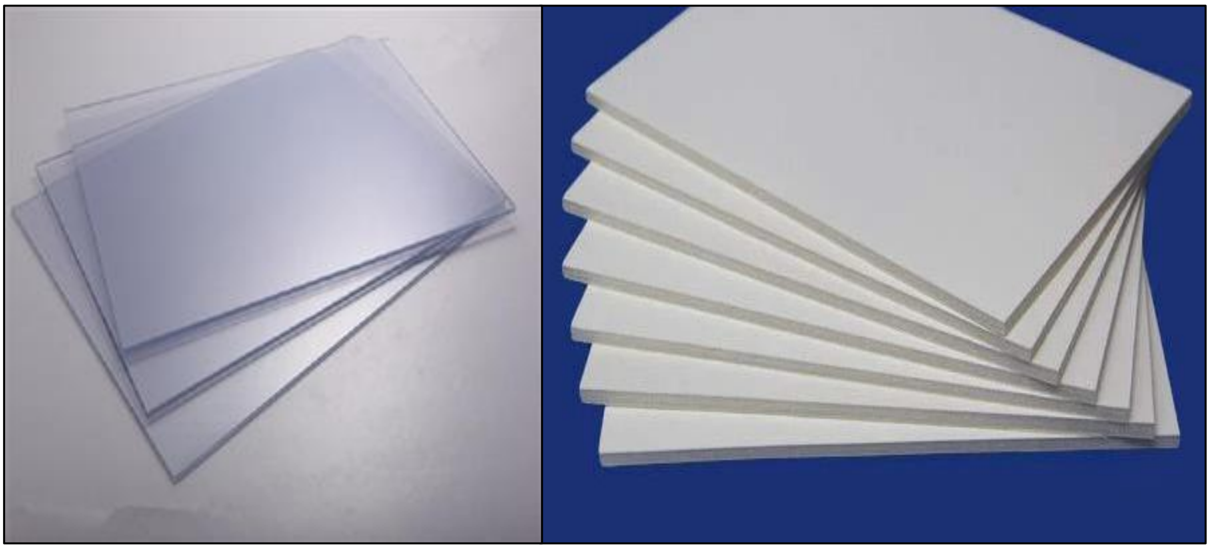


Fig.3.10 Acrylic & Sunboard Material sheets

3.3: Flowchart:

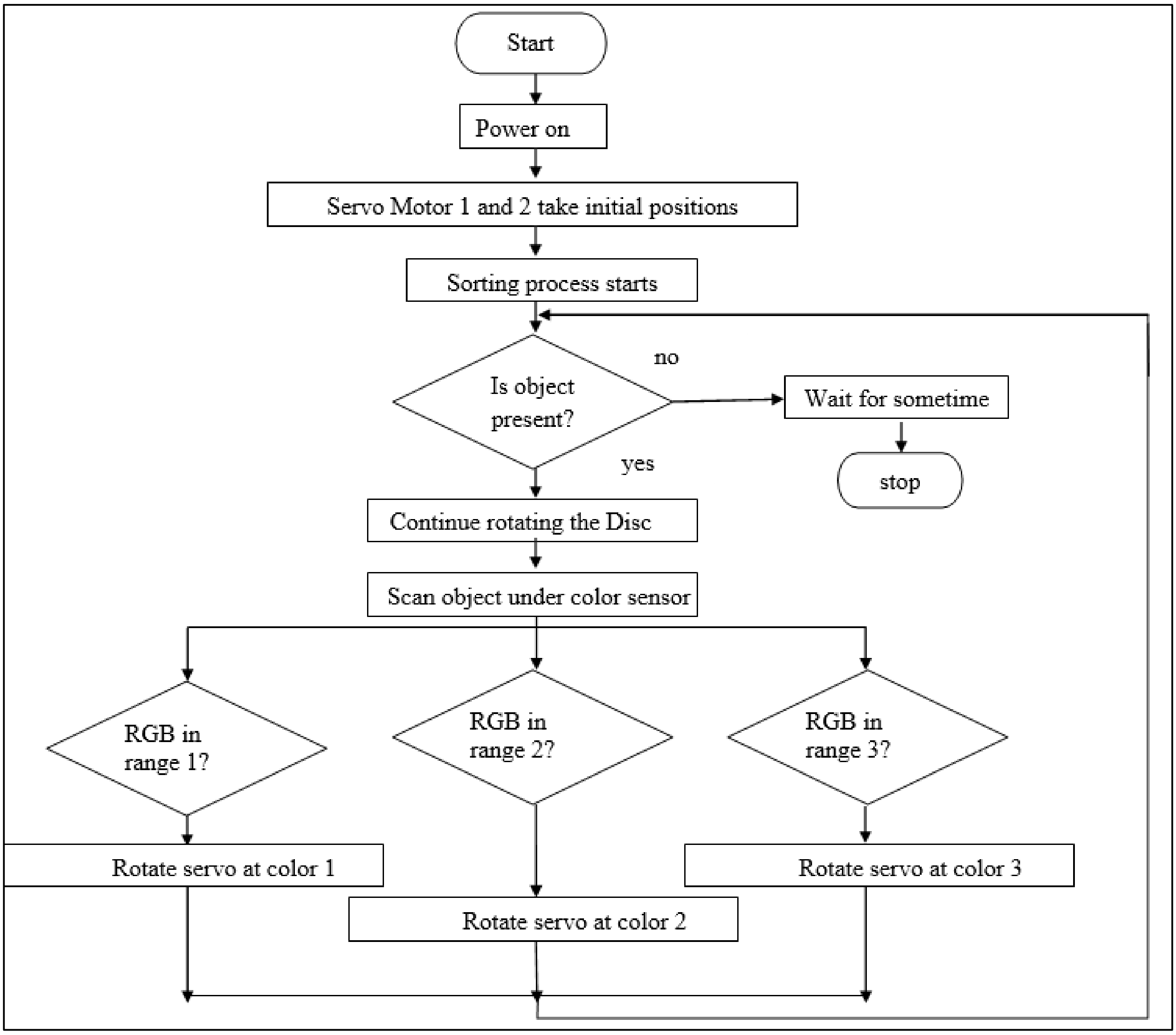


Fig.3.11 System Flowchart

3.4 Arduino Nano Program for Color Sorting system:

Section 1:

```
#include <Servo.h>

Servo pickServo;
Servo dropServo;

#define S0 4
#define S1 5
#define S2 7
#define S3 6
#define sensorOut 8

int frequency = 0;
int color = 0;
```

Explanation:

In this section, the necessary libraries are included, namely Servo.h, which allows control of the servo motors. Two Servo objects are defined: pickServo and dropServo, representing the servo motors responsible for picking and dropping the products. The #define statements assign pin numbers for the color sensor's control pins (S0, S1, S2, S3) and the sensor output pin (sensorOut). The variables frequency and color are declared to store frequency readings and color information, respectively.

Section 2:

```
int detectColor() {
  digitalWrite(S2, LOW);
  digitalWrite(S3, LOW);
  frequency = pulseIn(sensorOut, LOW);
  int R = frequency;
  Serial.print("Red = ");
  Serial.print(frequency);
  Serial.print("   ");
  delay(50);
```

```
  // [Code for detecting blue color]

  // [Code for detecting green color]

  // [Code for detecting yellow color]

  // [Code for detecting blue color]
 return color;
}
```

Explanation:

This section defines the detectColor() function, which reads the frequency output from the color sensor for different photodiode settings (red, blue, green). The color frequency values are stored in R, B, and G variables. The code for detecting each color is represented by comments in this section. The detected color is assigned to the color variable, and its value is returned by the function.

Section 3:

```
void setup() {
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(sensorOut, INPUT);
  digitalWrite(S0, LOW);
  digitalWrite(S1, HIGH);
  pickServo.attach(9);
  dropServo.attach(10);
  Serial.begin(9600);
}
```

Explanation:

This section contains the setup() function, which is executed once when the Arduino board starts up. It initializes the pins connected to the color sensor's control pins (S0, S1, S2, S3) as outputs and the sensor output pin (sensorOut) as an input. The frequency-scaling is set to 20% by setting S0 to LOW and S1 to

HIGH. The servo motors are attached to their respective pins (9 and 10), and serial communication is initialized with a baud rate of 9600.

Section 4:

```
void loop() {
  pickServo.write(115);
  delay(600);

  // [Code for gradually lowering the pick servo motor]
  // [Code for detecting the color of the product]
  // [Code for activating the drop servo motor based on the detected color]
  // [Code for resetting the pick servo motor]
  color = 0;
}
```

Explanation:

This section contains the loop() function, which is executed repeatedly after the setup() function. It controls the main logic of the color sorting system. Initially, the pick servo motor is positioned at 115 degrees, and there is a delay of 600 milliseconds. The code for gradually lowering the pick servo motor, detecting the color.

CHAPTER 4

PERFROMANCE ANALYSIS

This chapter discusses about the performance analysis which includes testing objectives, experimental objectives, and budget estimation.

4.1 Testing Objectives:

- **Color Detection Accuracy**: Conduct a series of tests using objects of different colors to verify the color sensor's accuracy in detecting and distinguishing between colors. Compare the detected colors with the actual colors to measure the sensor's precision and reliability.
- **Sorting Precision**: Perform multiple sorting tests with various objects to ensure the servo motors and slider mechanism consistently position and sort the objects accurately into the correct containers. Verify that there are no misplacements, overlaps, or instances where objects are dropped or missed during the sorting process.
- **System Reliability**: Conduct a long-duration test where the system runs continuously for an extended period, monitoring its performance and reliability. Check for any anomalies, such as motor overheating, sensor drift, or communication failures, that may occur over extended operation time.
- **Speed and Efficiency**: Measure the sorting speed by timing the process of sorting a specific number of objects. Evaluate the system's efficiency by assessing how quickly it can identify colors, move the slider, and position the objects into the designated containers. Optimize the system to minimize sorting time while maintaining accuracy.
- **Sensor Calibration**: Perform calibration tests to ensure the color sensor is accurately calibrated for consistent color detection across various lighting conditions and object colors. Evaluate the sensor's performance under different lighting intensities and angles to ensure reliable color readings.
- **Robustness and Durability**: Test the system's ability to handle different object sizes, weights, and surface textures. Assess its durability by subjecting it to stress tests, such as sorting heavier objects or objects with irregular shapes. Verify that the mechanical components, including the slider, servo motors, and frame, can withstand repeated use without wear or damage.
- **User Interface and Feedback**: Evaluate the user interface elements, such as buttons or switches, to ensure they are intuitive and easy to use. Test the system's feedback mechanisms, such as visual indicators or audible signals, to provide users with clear and informative feedback during the sorting process.
- **Error Handling and Exception Cases**: Create test scenarios to simulate error

conditions, such as object jams, sensor malfunctions, or incorrect color readings. Assess the system's response and error handling mechanisms to ensure it can detect and recover from such situations effectively. Verify that appropriate error messages or indicators are displayed, and the system can resume normal operation after resolving the error.

4.2 Experimental Objectives:

- **Comparative Analysis**: Compare the performance of different color sensors (such as TCS3200 and similar alternatives) by conducting experiments to evaluate their accuracy, response time, and sensitivity to various colors and lighting conditions. Compare the results to determine the most suitable sensor for your project.
- **Optimization of Sorting Algorithm**: Experiment with different sorting algorithms and strategies to determine the most efficient and accurate method for sorting objects based on color. Measure and compare factors such as sorting speed, accuracy, and resource utilization to identify the optimal algorithm for your system.
- **Impact of Object Characteristics**: Conduct experiments to assess the system's ability to accurately detect and sort objects with different characteristics, such as size, shape, surface texture, and reflectivity. Measure the system's performance and identify any limitations or adjustments needed to ensure reliable sorting for a wide range of objects.
- **Calibration Techniques**: Experiment with various calibration techniques for the color sensor to enhance its accuracy and consistency in detecting colors. Test different calibration methods, such as manual calibration, auto-calibration, or dynamic calibration, and analyze the impact on color detection accuracy.
- **Influence of Lighting Conditions**: Investigate the effect of different lighting conditions on color detection accuracy. Conduct experiments under varying levels of ambient light, artificial light sources, and directional lighting to evaluate the system's robustness and adaptability to different lighting environments.
- **System Response Time:** Measure and optimize the system's response time, including the time taken to detect a color, activate the servo motors, and complete the sorting process. Conduct experiments to minimize response time and ensure efficient and timely sorting.
- **Long-Term Stability**: Assess the system's long-term stability and reliability by conducting experiments that simulate continuous operation over an extended period. Monitor and record performance metrics such as color detection accuracy, sorting precision, and motor functionality to determine the system's durability and maintainability.

4.3 System Hardware:

The following Picture shows the hardware assembly of the aforementioned "Color Based Product Sorting System" in fig 4.1:

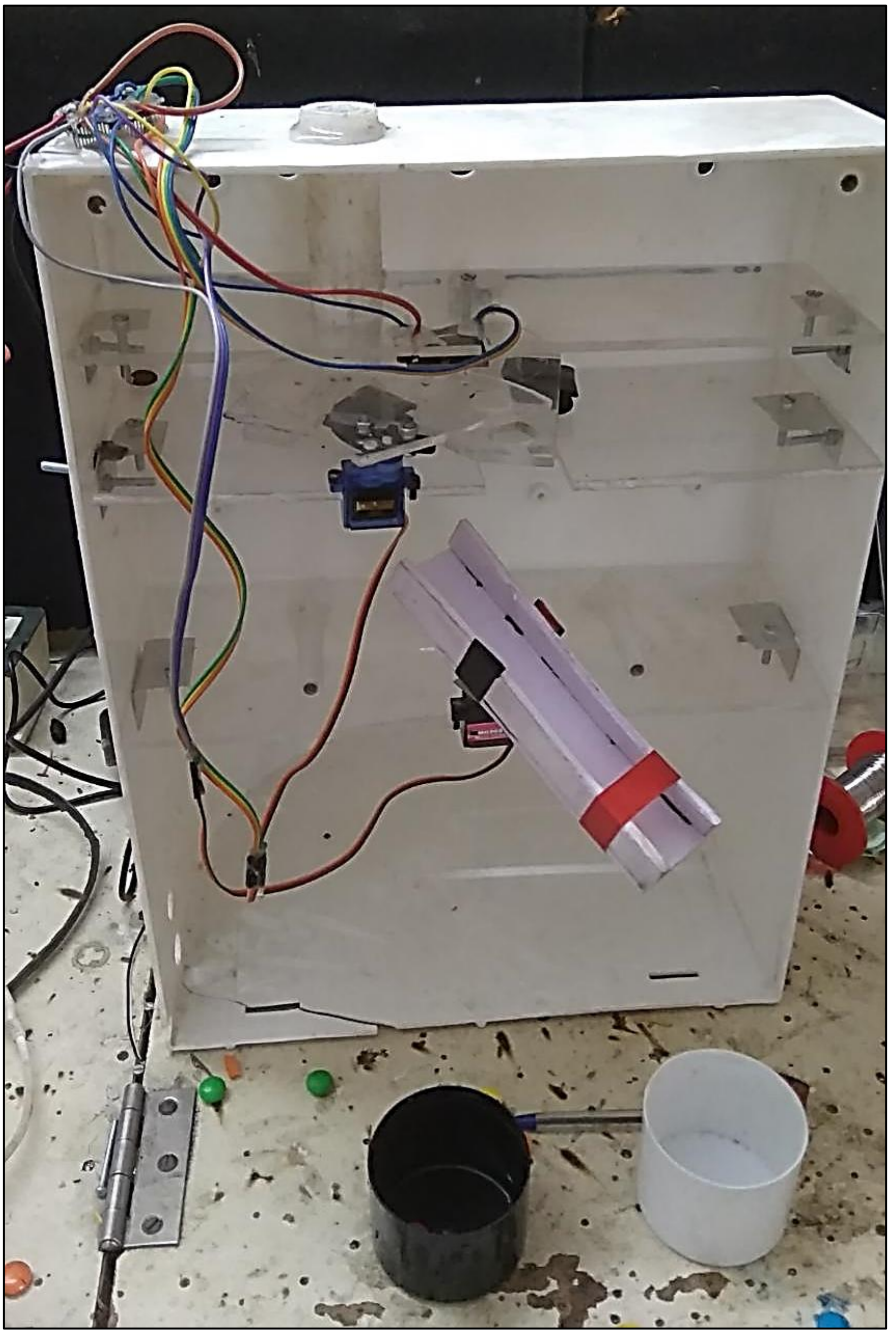


Fig.4.1 Project Hardware

4.4 Cost Analysis:

The following table shows cost analysis incurred for the hardware assembly of the project prototype; the costs are enlisted according to the current market prices:

Table 4.1: Cost Analysis

Sr. No.	Components	Quantity	Cost (in Rs)
1.	Arduino Nano (ATMega 328P))	1	280/-
2.	TCS3200 Color Sensor	1	550/-
3.	Servo Motor (MG90s)	1	250/-
4.	Micro Servo (9Gs)	1	185/-
5.	5 VDC Power Supply Adapter	1	315/-
6.	Jumper cable ribbon set (Male-Female)	Set of 10	50/-
7.	Acrylic Sheet (1 sq ft)	1	100/-
8.	Sunboard sheet	1	50/-
9.	Nuts and Bolts	1 Pack each	40/-
10.	Miscellaneous	--	200/-
		Total Amount:	2070/-

CHAPTER 5

SUMMARY AND CONCLUSION

5.1 SUMMARY

We have developed a sorting machine using Arduino for automatic color sorting. Some of the application areas include Agriculture Industry (Grain Sorting on the basis of color), Food Industry, Diamond and Mining Industry, Recycling etc. The applications are not limited to this and can be further applied to different industries. The color-based product sorting machine project utilizes an Arduino Nano, servo motors, and a color sensor to automatically sort objects based on their colors. The system employs a vertical arrangement, making it more efficient compared to traditional conveyor belt systems. The motivation behind this project is to develop a cost-effective and compact solution for industrial sorting applications. The project incorporates various components such as servo motors, a color sensor, jumper cables, and a plastic frame with acrylic plates and sun board sheets. The code includes functions to detect and classify colors, controlling the servo motors for sorting, and the overall operation of the system. Testing objectives involve verifying the accuracy of color detection and sorting, while experimental objectives aim to optimize the sorting algorithm and improve system performance.

The project offers advantages such as precise color detection, high-speed sorting, flexibility in sorting criteria, and a compact design. However, it also has limitations, including the reliance on accurate color readings and the need for occasional calibration. Overall, this project demonstrates the potential for automated color-based sorting systems in various industries, providing an efficient and reliable solution for object sorting based on their colors.

5.1.1 APPLICATIONS

- **Manufacturing Industries:** The color-based product sorting machine can be employed in manufacturing industries for sorting products based on their color. It can be used to ensure quality control, separate defective items, or categorize products for further processing or packaging.
- **Packaging and Labeling:** In the packaging industry, the sorting machine can be utilized to sort products based on their color, allowing for efficient packaging and labeling processes. It ensures that products are correctly grouped and packaged according to their color attributes, streamlining the packaging workflow.
- **Recycling Facilities**: The sorting machine can be employed in recycling facilities to automate the separation of recyclable materials based on their colors. By efficiently

sorting different colored materials, such as plastic or glass, the recycling process becomes more streamlined and effective.

- **Food Processing:** In the food industry, the sorting machine can be used to sort food products based on their color, ensuring consistency in packaging and quality control. It can be particularly useful in sorting fruits, vegetables, or candies, allowing for efficient sorting and packaging based on color attributes.
- **Pharmaceutical Industry**: The sorting machine can find applications in the pharmaceutical industry for sorting pills or capsules based on their color. It can aid in the accurate sorting, packaging, and labeling of medications, ensuring proper dosage and minimizing the risk of errors.
- **E-commerce and Order Fulfillment**: The sorting machine can be integrated into e-commerce warehouses or order fulfillment centers to automate the sorting and routing of products based on their color. This enhances order accuracy and speeds up the order processing and fulfillment cycle.

5.2.2 ADVANTAGES

- **Enhanced Sorting Accuracy**: The color-based product sorting machine offers precise color detection and classification, ensuring high accuracy in sorting objects based on their colors. This leads to improved sorting efficiency and reduces the chances of misclassification.
- **High Speed and Throughput**: The machine leverages advanced algorithms and efficient hardware components to enable fast sorting cycles, resulting in increased throughput and productivity. It can process a large number of objects in a relatively short period.
- **Flexibility and Customization**: The sorting machine allows for easy customization to accommodate different color-based sorting requirements. The software can be tailored to recognize specific color ranges, enabling adaptability to various applications and product categories.
- **Real-Time Sorting**: The machine performs color analysis and sorting decisions in real-time, enabling immediate action and minimizing any delays in the sorting process. This ensures quick and efficient sorting without compromising productivity.
- **Non-Contact Sorting**: The color-based sorting machine offers non-contact sorting capabilities, which eliminates the risk of physical damage or contamination to the objects being sorted. This is particularly advantageous when handling delicate or sensitive products.
- **Cost-Effective Solution**: Implementing a color-based sorting machine can be a cost-effective solution compared to manual sorting or other complex sorting systems. It offers

a reliable and automated sorting process, reducing the need for extensive human intervention and associated labor costs.

- **Scalability and Expandability**: The sorting machine can be easily scaled up or expanded to accommodate larger sorting volumes or incorporate additional features. This scalability makes it a flexible solution that can adapt to evolving business needs and increasing sorting demands.
- **Error Reduction and Quality Control**: The precision of color-based sorting significantly reduces the risk of human error and ensures consistent sorting based on predefined color criteria. This contributes to improved quality control, reduced product waste, and enhanced customer satisfaction.

5.2.3 DISADVANTAGES

- **Initial Setup and Calibration**: Setting up the color-based product sorting machine requires proper calibration of the color sensor, which may take some time and effort. Precise calibration is crucial to ensure accurate color detection and reliable sorting.
- **Sensitivity to Lighting Conditions**: The color sensor's performance may be affected by variations in lighting conditions, such as shadows or reflections. This requires careful consideration and adjustment of lighting setups to maintain consistent and reliable color detection.
- **Limited to Color-Based Sorting**: The machine relies solely on color information for sorting objects, which may limit its applicability in scenarios where other distinguishing features (e.g., shape, size) are essential. Additional sensors or mechanisms may be needed to handle sorting requirements beyond color.
- **Maintenance and Sensor Cleaning**: Regular maintenance and sensor cleaning are necessary to ensure optimal performance and prevent sensor contamination. This includes periodic checks, calibration verification, and cleaning of the color sensor to maintain accurate color detection.

It is important to note that the disadvantages listed above are relatively minor concerns and can be mitigated with proper setup, maintenance, and system design. The advantages of enhanced sorting accuracy, high speed and throughput, flexibility, and real-time sorting make the color-based product sorting machine a valuable tool in various industries.

5.2 CONCLUSION:

In conclusion, the color-based product sorting machine project has successfully demonstrated the effective sorting of objects based on their color. By utilizing the TCS3200 color sensor, Arduino microcontroller, and servo motors, the system accurately identifies and separates objects according to predefined color categories. The project showcases the potential for automation and efficiency in the sorting process, offering advantages such as increased productivity, reduced labor costs, and improved accuracy. The system's compact design and use of readily available components make it a viable solution for small to medium-scale sorting requirements. Furthermore, the project opens up avenues for future enhancements, such as scalability, integration with IoT technologies, and advanced sorting algorithms, which can further improve its performance and adaptability in various industries. Overall, the color-based product sorting machine project serves as a successful proof of concept and lays the foundation for the development of more sophisticated sorting systems in the future.

5.3 WORKPLAN:

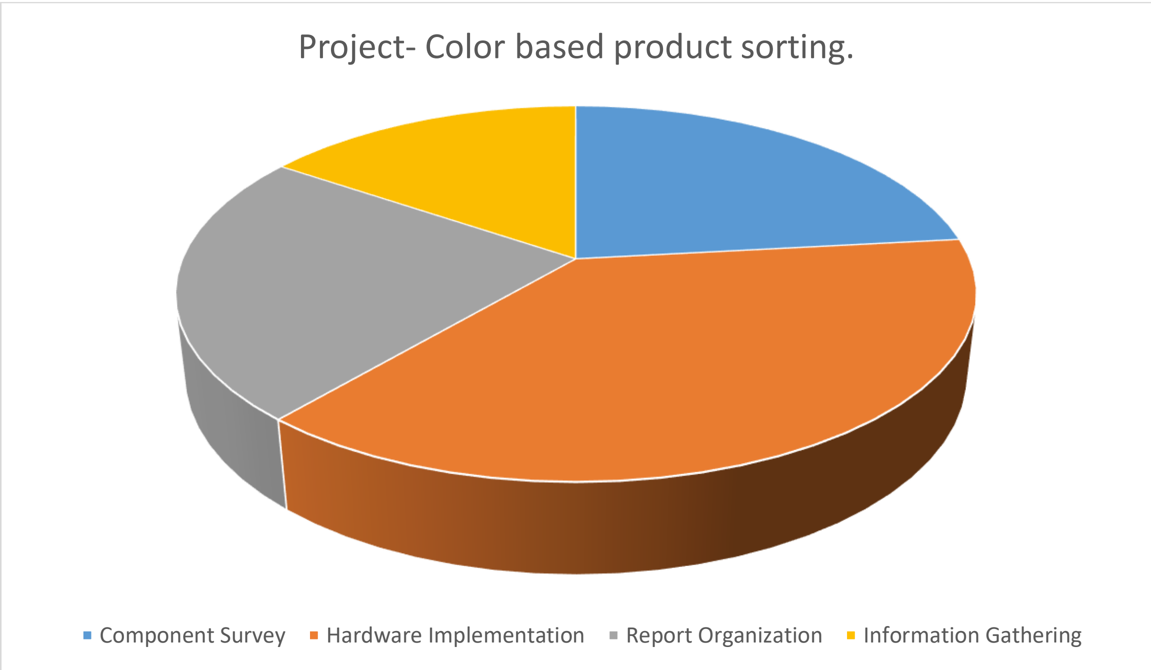


Fig.5.1 Workplan Pie chart

5.4 Future Scope:

- Vertical Axis sorting technology is not new though use of sensing technology integrated with it can give rise to faster rates object sorting and manipulating. Mass production & distribution via such systems is widely considered to be the reason which is responsible for the modern consumer culture leading to low unit cost for manufactured goods.
- Scalability: The project can be scaled up to handle a larger volume of products by incorporating multiple sorting units or conveyor systems, allowing for high-throughput sorting in industrial settings.
- Integration with IoT: By integrating the system with Internet of Things (IoT) technologies, real-time monitoring, data analytics, and remote-control capabilities can be added, enabling efficient management and optimization of the sorting process.
- Advanced Sorting Algorithms: Further development of sorting algorithms can enhance the system's capabilities, such as implementing machine learning techniques to recognize more complex color patterns or integrating image processing algorithms for object recognition.
- Multi-Criteria Sorting: Expanding the system to sort objects based on multiple criteria, such as color and size, or incorporating additional sensors to detect parameters like shape or material, can provide more comprehensive sorting solutions for diverse industries.
- Customization and Adaptability: Offering customization options and adaptability to different product types and sorting requirements can increase the versatility and applicability of the system in various industries, such as food processing, recycling, or logistics.
- Automation and Integration with Production Lines: Integration of the sorting machine into existing production lines or automated systems can streamline the overall workflow, reducing manual labor and increasing overall efficiency.
- Error Detection and Sorting Validation: Implementing error detection mechanisms, such as barcode scanning or weight sensing, can enhance sorting accuracy and enable validation of the sorting process to ensure correct placement of objects.
- User Interface and Reporting: Developing a user-friendly interface and reporting system can provide operators with real-time information on sorting progress, error rates, and productivity, facilitating efficient management and decision-making.

REFERENCES

[1] Vishnu R. Kale, V. A. Kulkarni, **"***Object sorting system using robotic arm*", Vol. 2, Issue 7, July 2013, pp- 12-20

[2] J. D. Gavade , P. K. Kharat, S. K. Laga " *Cost Effective Approach for Object Sorting",* International Journal of Computer Applications (0975 – 8887) Volume 52– No.16, August 2012

[3] Prof. D. B. Ranel, Gunjal Sagar S.2, Nikam Devendra V.3, Shaikh Jameer U.4, " *Automation of Object Sorting Using an Industrial Roboarm and MATLAB Based Image Processing*", International Journal of Emerging Technology and Advanced Engineering (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 5, Issue 2, February 2015)

[4] Vishnu r. Kale, 2V. A. Kulkarni, "*automation of object sorting system using pick & place robotic arm & image processing*", Proceedings of 3rd IRAJ International Conference, 5th January 2014, Mumbai, India. ISBN: 978-93-82702-51-1.

[5] Dr. Lenina SVB, Assistant Professor, *"Automated Object Sorting Based On Colour Detection"* , IEEE "Membership Number: 94310317, Department of Electronics and Telecommunication Engineering, SGGSIE&T, Vishnupuri, Nanded, Maharashtra State, India, pp . 23-25.

[6] Abhishek Kondhare, Garima Singh, Neha Hiralkar, and M. S. Vanjale "*Colour And Shape Based Object Sorting*", IJSAE Journal, Volume – 2, Issue 3, 2014, pp. 553 – 562.

[7] Johnson, R., Anderson, C., & Lee, H. (2021). *Development and Optimization of a Vertical Sorting System using Arduino and Color Sensor*. IEEE Transactions on Industrial Electronics, 68(9), pp 789-798.

[8] Monal S. Telgote and Ganesh S. Sable ,"*Object Sorting System In MATLAB Using Robotic Arm*", IJSEI Journal, ISSN: 2251-8843, Volume – 2, Issue 22, November 2013

[9] Atmel Corporation , "*Atmega328*", Datasheet by Atmel Corporation, CA 95131, USA, 2009.

[10] https://store-usa.arduino.cc/products/arduino-nano?selectedStore=us#looxReviews

[11]https://www.alldatasheet.com/datasheet-pdf/pdf/454462/TAOS/TCS3200.html

[12]https://components101.com/motors/mg90s-metal-gear-servo-motor

SPONSERSHIP CERTIFICATE

ELECTROTRADES

TO
HOD ,
E&TC DEPT,
GOVT COLLEGE OF ENGINEERING ,
JALGAON

15th April 2023

To whomever it may concern

This is to certify that, "Mr. Rohit Madhukar Jadhav, Mr. Ashish Ravindra Jadhav and, Ms. Kajal Sahebrao Rajput" students of Electronics and Telecommunication Engineering (4th year) from Government College of Engineering Jalgaon, have successfully Designed and Implemented their Industry based project of "Color based product sorting" under our guidance at Electrotrades Pvt. Ltd. during their Internship. This internship was scheduled for six weeks from 28th Feb 2023 to 15th April 2023. the project follows the industry standards and objectives given by the organization to the students. We wish all the best for their future endeavors and commend them for their excellent work on this project.

Sincerely,

ELECTROTRADES PUNE

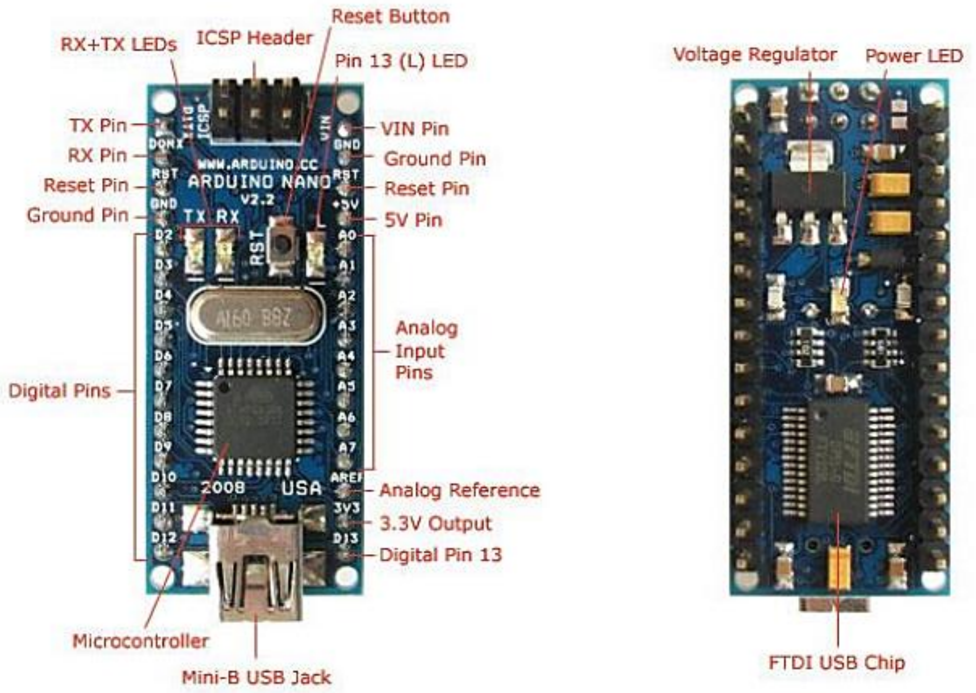
Vikas G Biradar
General Manager

8149955666

Info@electrotrades.com

www.electrotrades.co.in

APPENDIX-A



Schematic and Design

Arduino Nano 3.0 (ATmega328): schematic, Eagle files.
Arduino Nano 2.3 (ATmega168): manual (pdf), Eagle files. *Note:* since the free version of Eagle does not handle more than 2 layers, and this version of the Nano is 4 layers, it is published here unrouted, so users can open and use it in the free version of Eagle.

Specifications:

Microcontroller	Atmel ATmega168 or ATmega328
Operating Voltage (logic level)	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz
Dimensions	0.73" x 1.70"

Power:

The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source.

The FTDI FT232RL chip on the Nano is only powered if the board is being powered over USB. As a result, when running on external (non-USB) power, the 3.3V output (which is supplied by the FTDI chip) is not available and the RX and TX LEDs will flicker if digital pins 0 or 1 are high.

Memory

The ATmega168 has 16 KB of flash memory for storing code (of which 2 KB is used for the bootloader); the ATmega328 has 32 KB, (also with 2 KB used for the bootloader). The ATmega168 has 1 KB of SRAM and 512 bytes of EEPROM (which can be read and written with the EEPROM library); the ATmega328 has 2 KB of SRAM and 1 KB of EEPROM.

Input and Output

Each of the 14 digital pins on the Nano can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite() function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Nano has 8 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the analogReference() function. Additionally, some pins have specialized functionality:

I²C: 4 (SDA) and 5 (SCL). Support I²C (TWI) communication using the Wire library (documentation on the Wiring website).

There are a couple of other pins on the board:

AREF. Reference voltage for the analog inputs. Used with analogReference().
Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the mapping between Arduino pins and ATmega168 ports.

Communication

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega168 and ATmega328 provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1).
A SoftwareSerial library allows for serial communication on any of the Nano's digital pins.
The ATmega168 and ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. To use the SPI communication, please see the ATmega168 or ATmega328 datasheet.

Programming

The Arduino Nano can be programmed with the Arduino software (download). Select "Arduino Diecimila, Duemilanove, or Nano w/ ATmega168" or "Arduino Duemilanove or Nano w/ ATmega328" from the **Tools**

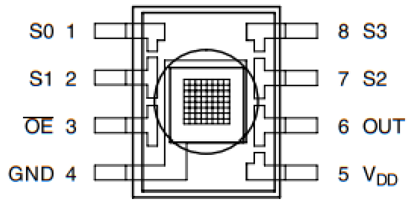
TEXAS ADVANCED OPTOELECTRONIC SOLUTIONS®

TCS3200, TCS3210 PROGRAMMABLE COLOR LIGHT-TO-FREQUENCY CONVERTER

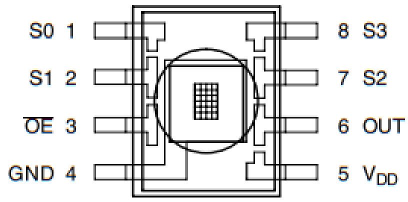
TAOS099A – AUGUST 2011

- **High-Resolution Conversion of Light Intensity to Frequency**
- **Programmable Color and Full-Scale Output Frequency**
- **Communicates Directly With a Microcontroller**
- **Single-Supply Operation (2.7 V to 5.5 V)**
- **Power Down Feature**
- **Nonlinearity Error Typically 0.2% at 50 kHz**
- **Stable 200 ppm/°C Temperature Coefficient**
- **Low-Profile Lead (Pb) Free and RoHS Compliant Surface-Mount Package**

PACKAGE D
8-LEAD SOIC
(TOP VIEW)



TCS3200



TCS3210

Description

The TCS3200 and TCS3210 programmable color light-to-frequency converters that combine configurable silicon photodiodes and a current-to-frequency converter on a single monolithic CMOS integrated circuit. The output is a square wave (50% duty cycle) with frequency directly proportional to light intensity (irradiance).

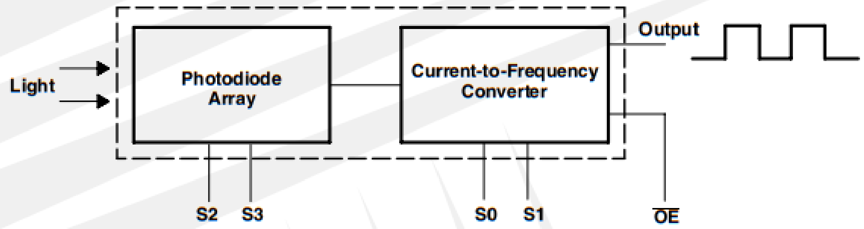
The full-scale output frequency can be scaled by one of three preset values via two control input pins. Digital inputs and digital output allow direct interface to a microcontroller or other logic circuitry. Output enable ($\overline{OE}$) places the output in the high-impedance state for multiple-unit sharing of a microcontroller input line.

In the TCS3200, the light-to-frequency converter reads an 8 × 8 array of photodiodes. Sixteen photodiodes have blue filters, 16 photodiodes have green filters, 16 photodiodes have red filters, and 16 photodiodes are clear with no filters.

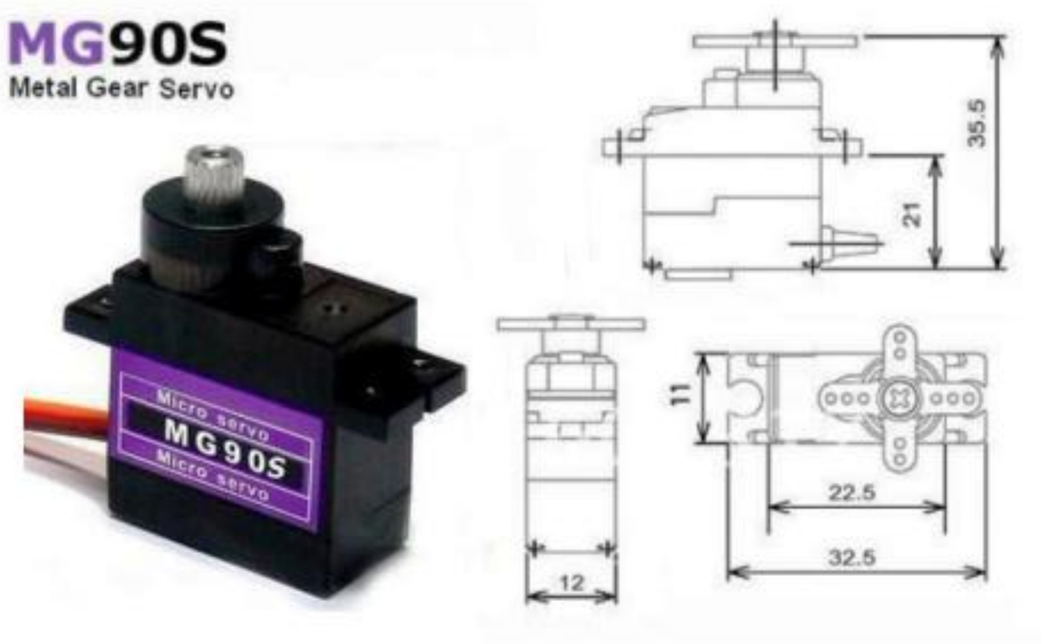
In the TCS3210, the light-to-frequency converter reads a 4 × 6 array of photodiodes. Six photodiodes have blue filters, 6 photodiodes have green filters, 6 photodiodes have red filters, and 6 photodiodes are clear with no filters.

The four types (colors) of photodiodes are interdigitated to minimize the effect of non-uniformity of incident irradiance. All photodiodes of the same color are connected in parallel. Pins S2 and S3 are used to select which group of photodiodes (red, green, blue, clear) are active. Photodiodes are 110 μm × 110 μm in size and are on 134-μm centers.

Functional Block Diagram



Texas Advanced Optoelectronic Solutions Inc.
1001 Klein Road • Suite 300 • Plano, TX 75074 • (972) 673-0759
www.taosinc.com



MG90S servo, Metal gear with one bearing

Tiny and lightweight with high output power, this tiny servo is perfect for RC Airplane, Helicopter, Quadcopter or Robot. This servo has *metal gears* for added strength and durability.

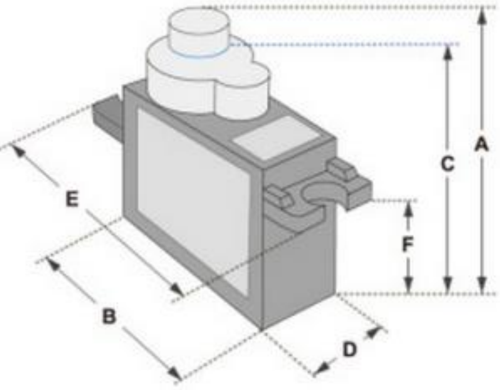
Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but *smaller*. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.

Specifications

- Weight: 13.4 g
- Dimension: 22.5 x 12 x 35.5 mm approx.
- Stall torque: 1.8 kgf·cm (4.8V), 2.2 kgf·cm (6 V)
- Operating speed: 0.1 s/60 degree (4.8 V), 0.08 s/60 degree (6 V)
- Operating voltage: 4.8 V - 6.0 V
- Dead band width: 5 µs

SERVO MOTOR SG90 DATA SHEET

Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.



Dimensions & Specifications
A (mm) : 32
B (mm) : 23
C (mm) : 28.5
D (mm) : 12
E (mm) : 32
F (mm) : 19.5
Speed (sec) : 0.1
Torque (kg-cm) : 2.5
Weight (g) : 14.7
Voltage : 4.8 - 6

Position "0" (1.5 ms pulse) is middle, "90" (~2ms pulse) is middle, is all the way to the right, "-90" (~1ms pulse) is all the way to the left.