AIN SHAMS UNIVERSITY

FACULTY OF ENGINEERING

Mechatronics Engineering

MCT 333s

Mechatronic System Design

(MCT 333s)

Production line project

NAME LIST:

Team (20)

NO	B. N	NAME	ID	sec
1		Peter Munir Azmy	1901934	2
2		Fares Ahmed Mohamed	19P6135	2
3		Hassan Ahmed Mohamden	2002008	2
4		Ahmed Hawary Abdelrehem ali	2000135	2
5		Mahmoud Mohamed Ibrahim Ali	2001481	3
6		Mohamed Hany Farouk	2001266	3

➢ Introduction:

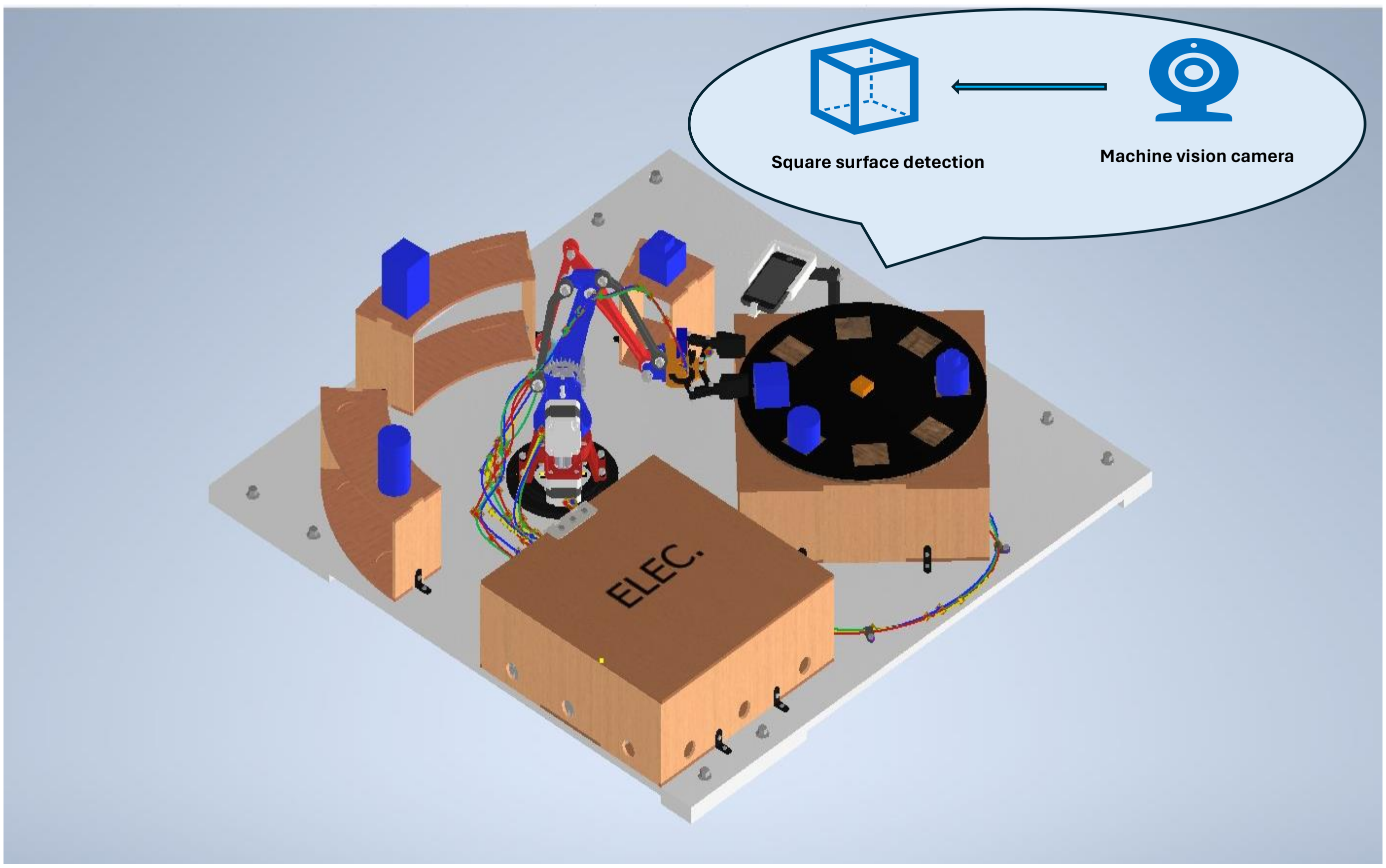
A production line, also known as an assembly line, is a manufacturing process in which parts are added to a product in a sequential manner to create a finished product more efficiently. This method is used in mass production and involves a series of workstations that perform different tasks simultaneously, but at a controlled pace. The key advantages of a production line include increased productivity, more efficient use of materials, improved product quality, and faster production times. This system was popularized by Henry Ford in the early 20th century, revolutionizing the automotive industry by making the assembly process quicker and more cost-effective. Today, production lines are integral to various industries, including electronics, food, and pharmaceuticals, adapting to new technologies such as automation and robotics to enhance efficiency and precision.

- **machine vision with production line _robotic arm:**

A robotic arm with machine vision on a production line represents the pinnacle of technological integration in industrial automation, blending precision mechanics with sophisticated image processing capabilities. These robotic arms are equipped with cameras and sensors that allow them to "see" their environment. This ability to visually analyze their surroundings enables them to perform complex tasks such as quality control inspections, precise assembly, and intricate component handling with higher accuracy and adaptability.

Machine vision systems enable robotic arms to identify, sort, and position parts with a level of precision that far surpasses human capabilities. These systems use real-time data to guide the robot's actions, allowing for adjustments on the fly and enabling the handling of a diverse range of products without manual reprogramming. This integration significantly enhances the flexibility and efficiency of production lines, particularly in industries where customization and high-quality standards are critical, such as in electronics manufacturing, pharmaceuticals, and automotive assembly.

The use of machine vision in robotic arms not only boosts production throughput and product quality but also reduces waste and operational costs by minimizing errors and improving safety. As part of smart manufacturing practices, these technologically advanced robotic systems play a crucial role in the ongoing evolution of Industry 4.0, pushing the boundaries of what automated production lines can achieve.



➢ V_model techniques for project:

In this project, we are constrained by 5 main stages, but we can make any design to achieve these stages, we design a simple design with main robot arm to make most of it.

- ➢ **We select our requirements based on 5 stages:**

Stage	Function	Requirement
1-Feeding	Feeding the products to certain position to pick up by the robot	We make a feeding mechanism by rotating disk rotated by DC motor controlled by PID controller
2-Sorting	Sorting the products based on shapes (circle _square) by edge detection	we need to use machine vision (via camera)
3-handling	we need to handle the products from place to place and assembly the similar products which other and storge it	We need to a robotic arm can make this function with 3 DOF or more with end effector
4-assembly		
5-storg		

➢ Mechanical design and modeling:

We divided the production line that makes five stages for a product to 2 stations:

1- Station includes (Feeding _ sorting)

2- Station includes (handling _assembly _storage)

- **Station includes (Feeding _ sorting):**

➢ Mechanical design:

The design of the feeding mechanism depends on the presence of a disk that rotates with the help of a DC motor with a PID controller to adjust the circular angles divided into every 45 degrees (in the form of square shape), from which the robot picks up the product after classifying it with the camera that knows if the type of product is Where if it is circular or square in shape.

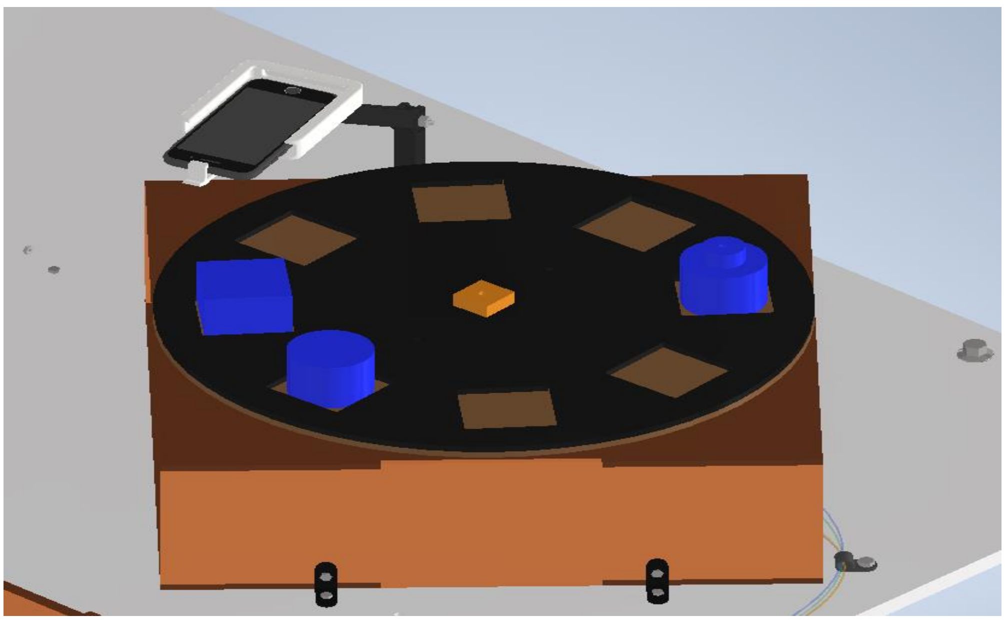
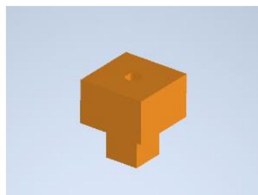


Image from cad file for feeding and sorting.

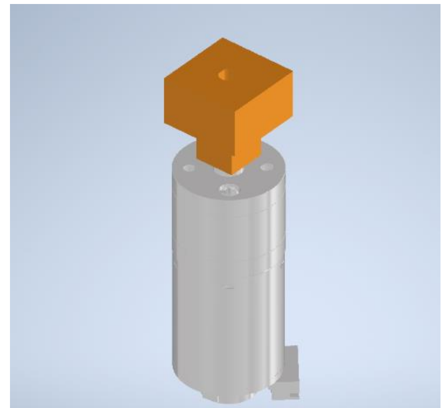
The used motor for feeding mechanism:

DC Motors 25GA-370 Metal Gear DC 12V Reduction Motor with Encoder and Gearbox for 280RPM With use with motor a 3d coupler to link between motor and rotating disk.

Dc motor



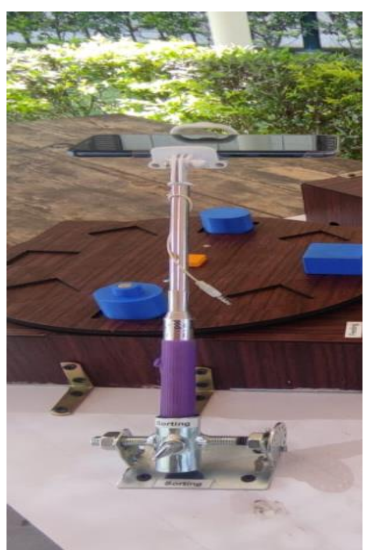
3D printed coupler



dc motor with coupler

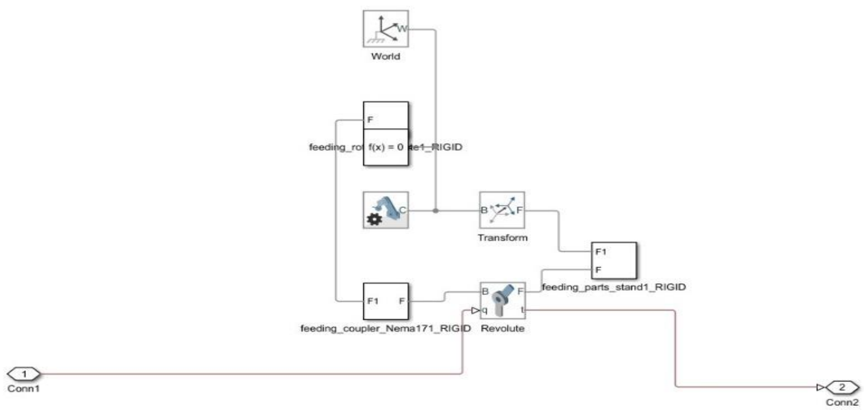
The used camera is:

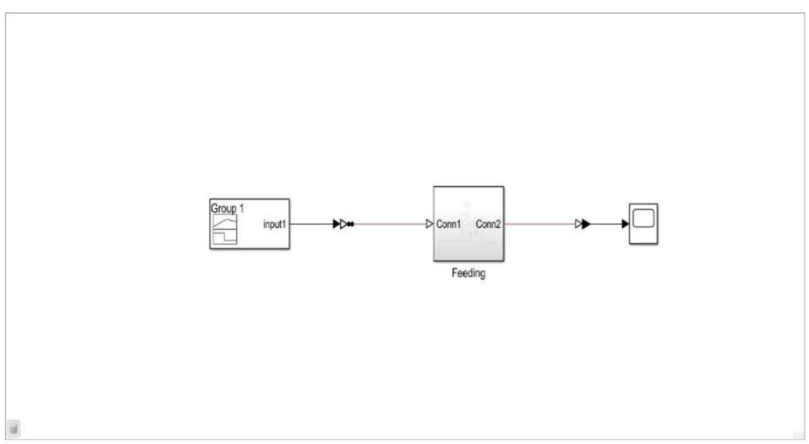
We use a mobile camera (mi note 10) connected with laptop by program (Iriun Webcam) with stand from our design to hold it.



We use a thrust bearing (51105) between the rotating disk and the fixed surface to reduce the friction force.

- **Modeling and simulation:**
  by using **MATLAB Simulink** to model the system





Feeding model in MATLAB

From the results we select the motor which achieves this torque and actuate the mechanism and the is **DC Motors 25GA-370 Metal Gear DC 12V Reduction Motor with Encoder and Gearbox for 280RPM with Stall Torque: 3Kg/cm.**

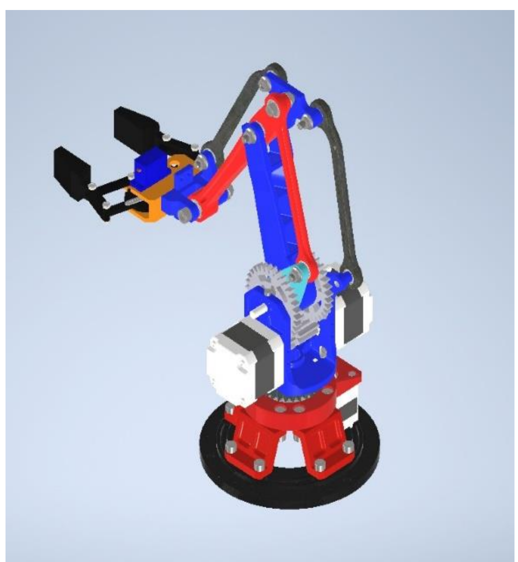
Link of cad files:

- **Station includes (handling _assembly _storage):**

➢ **Mechanical design:**

In these 3 stages, the robot picks up the product, then assembles it with a similar part, then storges it in a desired place for that.

This 3DOF_ robot is made by **Florin Tobler _2016** and we make an inverse engineering to it to be suitable at this case and can matching with Egyptian industry conditions and the presence of components in the markets.



Robotic arm from cad file

For the robotic arm we use inverse engineering techniques to check everything at the robot:

1- length of links and its ability to achieve the function.
2- considering the friction between joints by using a lot of types from bearing (deep grove ball bearing (686zz) – thrust bearing (51105)).



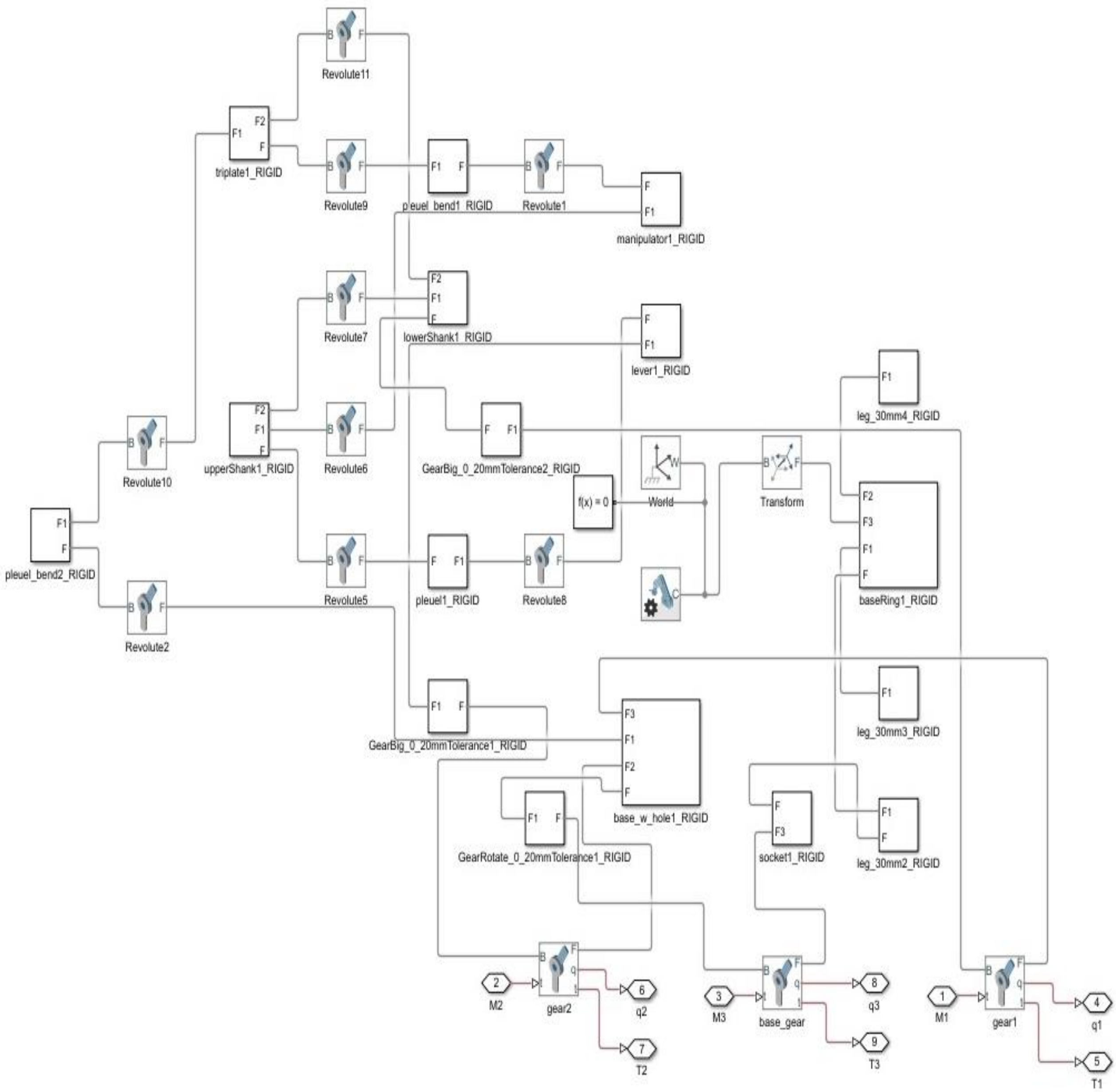
3- the loads in each part and the selection motor by modeling it **MATLAB.**
4- Considering The techniques of fabrication on The Egyptian environment and Availability of screws, nuts, and fixing tools.

the BOM to robot arm:

PARTS LIST			
ITEM	QTY	PART NUMBER	DESCRIPTION
1	1	base_w_hole	
2	1	lowerShank	
3	4	GearSmall	
4	1	lever	
5	2	GearBig_0.20mmTolerance	
6	1	upperShank	
7	1	pleuel	
8	1	triplate	
9	1	manipulator	
10	1	GearRotate_0.20mmTolerance	
11	2	stepper	
12	1	stepper_short	
13	1	socket	
14	3	leg_30mm	
15	1	baseRing	
16	2	pleuel_bend	
17	1	thrustBearing	
18	18	bearing_6x13x5	
19	1	rod_M6_80mm	
20	1	nut_M6_lock	
21	1	screw_M6_40mm	
22	2	ring_M6	
23	1	Assembly 2.0 upload	
24	8	ISO 4762 - M5 x 16	Hexagon Socket Head Cap Screw
25	12	IS 1364-3 - M 5	Hexagon Head Bolts
26	9	ISO 4762 - M5 x 20	Hexagon Socket Head Cap Screw
27	4	ISO 4762 - M3 x 8	Hexagon Socket Head Cap Screw
28	10	ISO 4762 - M3 x 20	Hexagon Socket Head Cap Screw
29	8	BS 3692 - M3	Precision hexagon nuts
30	8	ISO 4762 - M3 x 16	Hexagon Socket Head Cap Screw
31	1	washer_6mm_1mm	
32	2	washer_6mm_1.4mm	
33	6	DIN 6923 - M5	Hex Nut
34	1	ISO 4762 - M6 x 45	Hexagon Socket Head Cap Screw
35	1	DIN 6923 - M6	Hex Nut
36	1	ISO 4762 - M5 x 45	Hexagon Socket Head Cap Screw
37	1	ISO 4762 - M5 x 25	Hexagon Socket Head Cap Screw

5- Considering The size and the loads of the gripper and an ability to hold the parts.

- **Modeling and simulation:**
  by using **MATLAB Simulink** to model the system



Cad file robot in Simulink

-Actuator sizing:

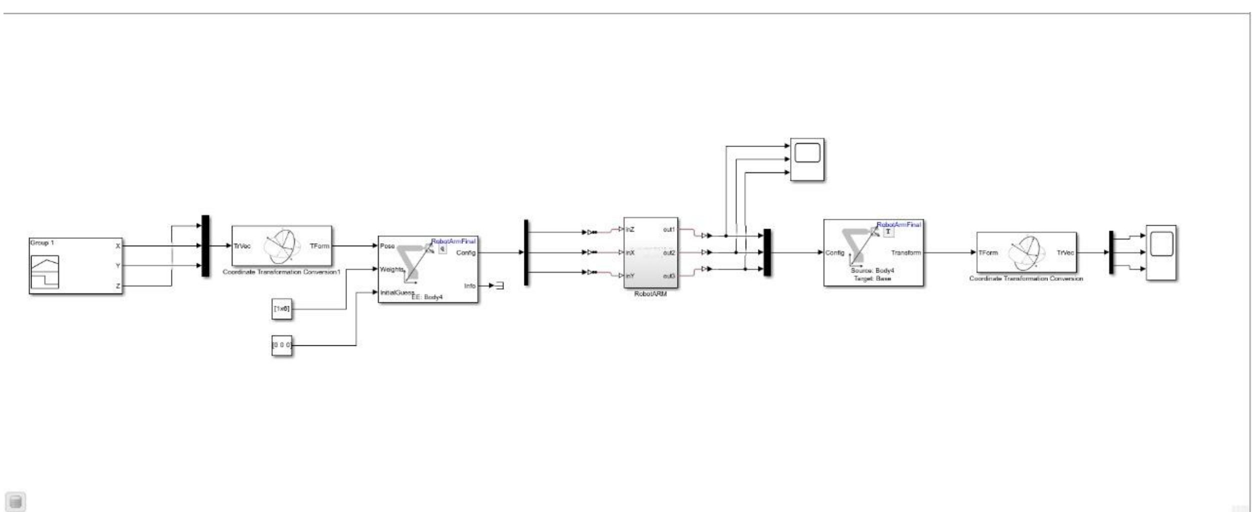
From the results we select the motor which achieves this torque and actuate the mechanism and the is This robot will be actuator by stepper motor _Nema 17_to achieving arrival for certain angle.

We use 3 motors to actuate 3 angles in a robot arm.

-Inverse kinematics:

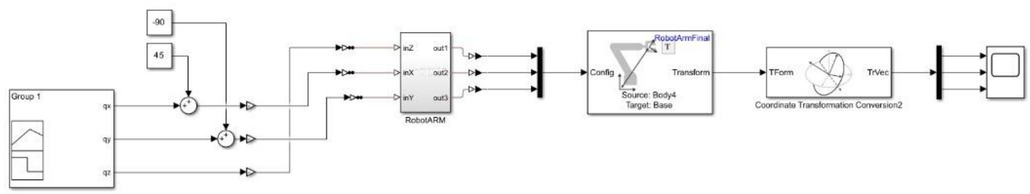
Inverse kinematics allows precise control over the motion of the end effector by calculating the required joint configurations to achieve a desired position and orientation in the workspace.

For robotic arms with multiple joints, determining the joint angles manually to reach a specific end effector position.



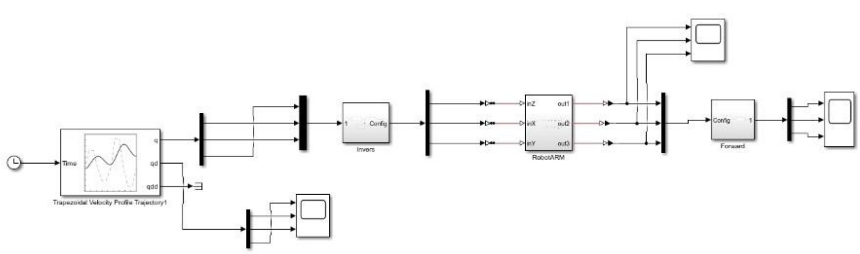
-Forward kinematics:

in robot kinematics, forward kinematics refers to the use of the kinematic equations of a robot to compute the position of the end-effector from specified values for the joint parameters.



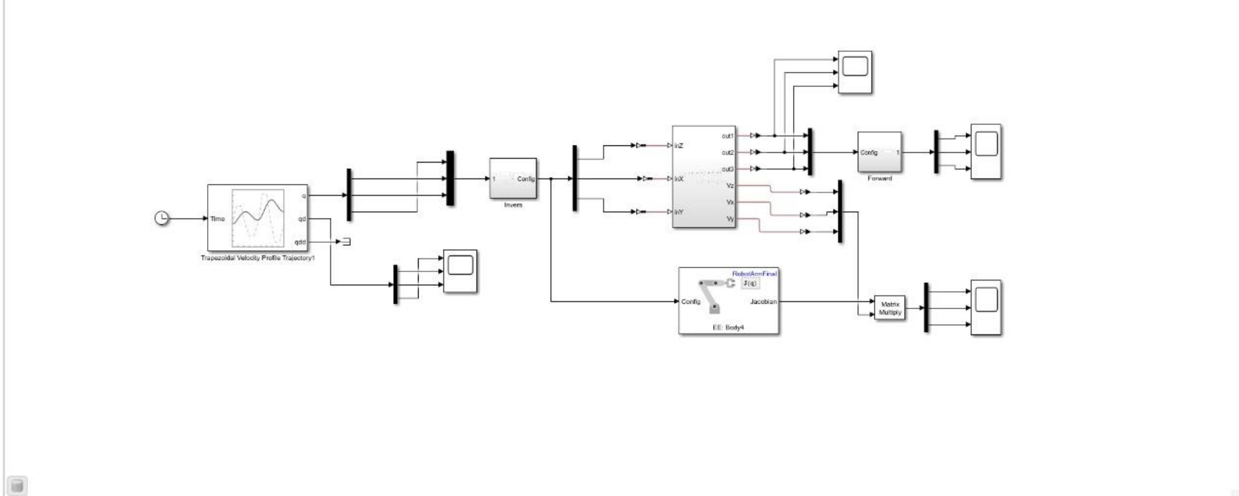
- trajectory planning:

robot manipulator trajectory planning takes the ideal trajectory kinematics parameters and the robot manipulator system as the input, and takes the displacement, velocity and acceleration of each joint and end effector as the output.



-Jacobian matrix:

Jacobian is the determinant of the Jacobian matrix. The matrix will contain all partial derivatives of a vector function. The main use of Jacobian is found in the transformation of coordinates. It deals with the concept of differentiation with coordinate transformation.



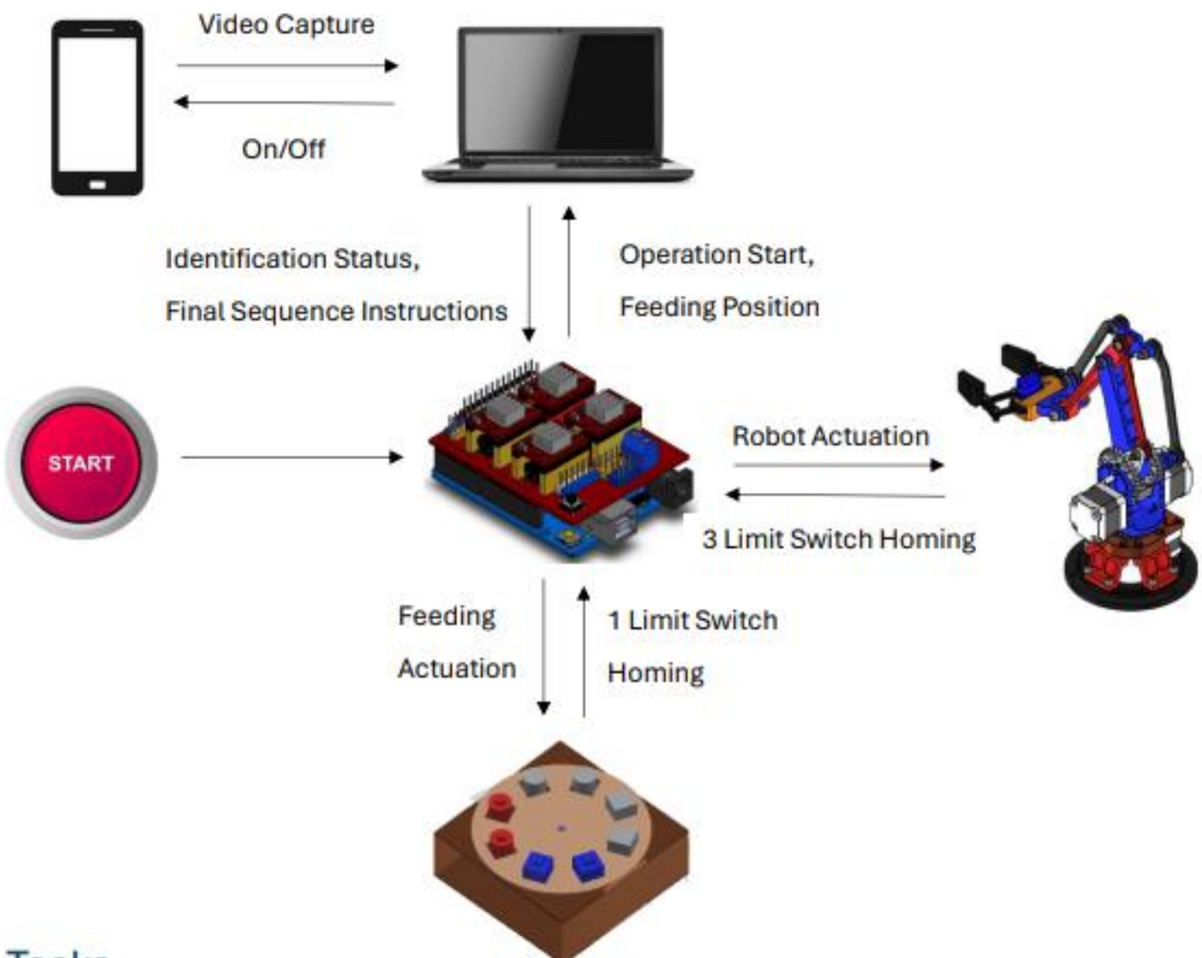
➢ The fabrication processes:

We use wooden laser cut to build the feeding mechanism and the stand which the robot will put the products in it.

And use the 3d printer to abdicate the robot arm.

➢ The electrical system and integration:

- **The sequence of the project and algorithms**



Tasks

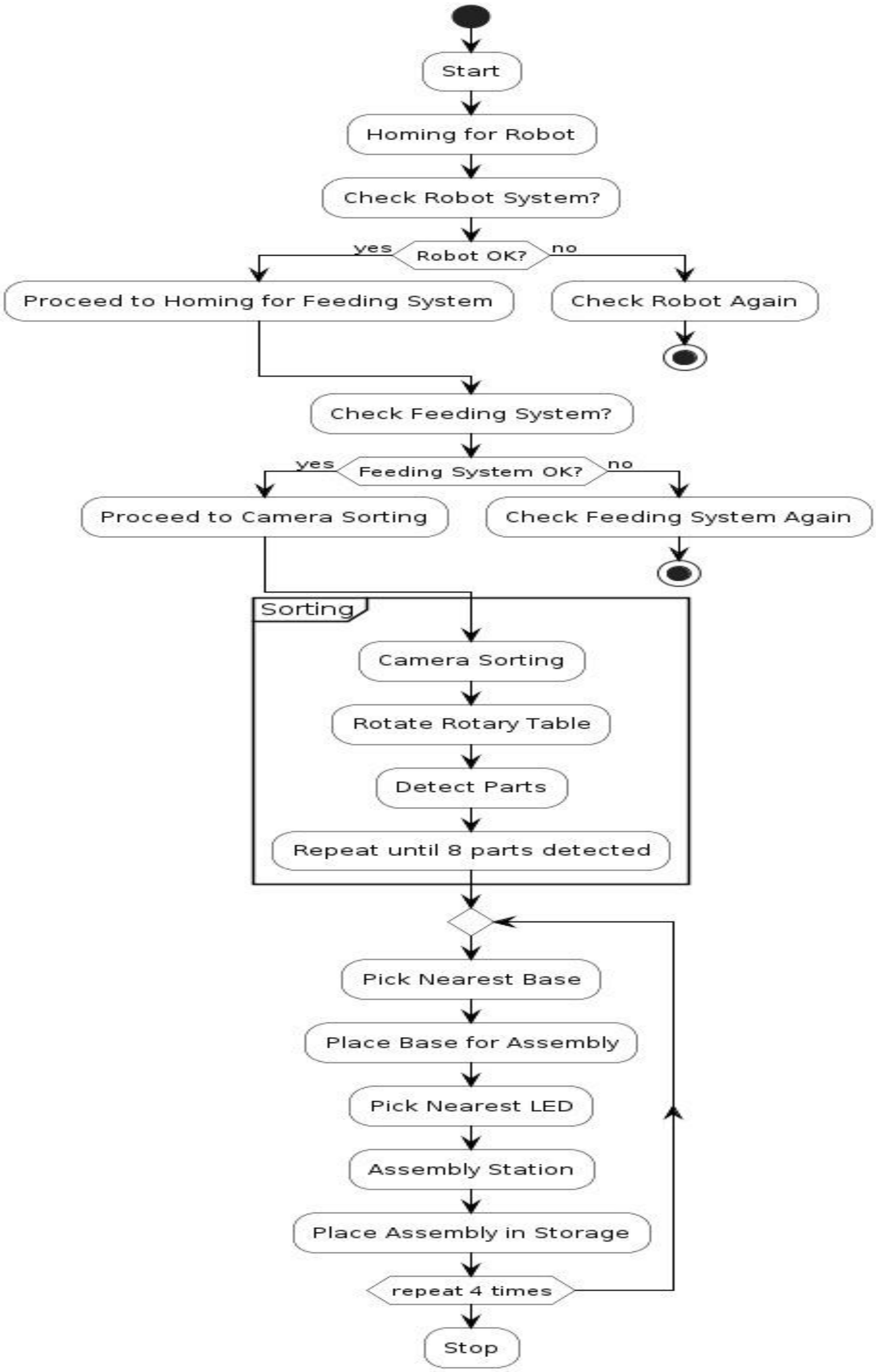
1. Homing: Robot arm and feeding positions are reset. → Arduino
2. Vision: Sequence for scanning each object on feeding disc. → Arduino and Laptop
3. Sequence Tuning: Identify efficient operation sequence. → Laptop
4. Handling, Sorting and Storage: Sequence execution. → Arduino

- **Summary:**

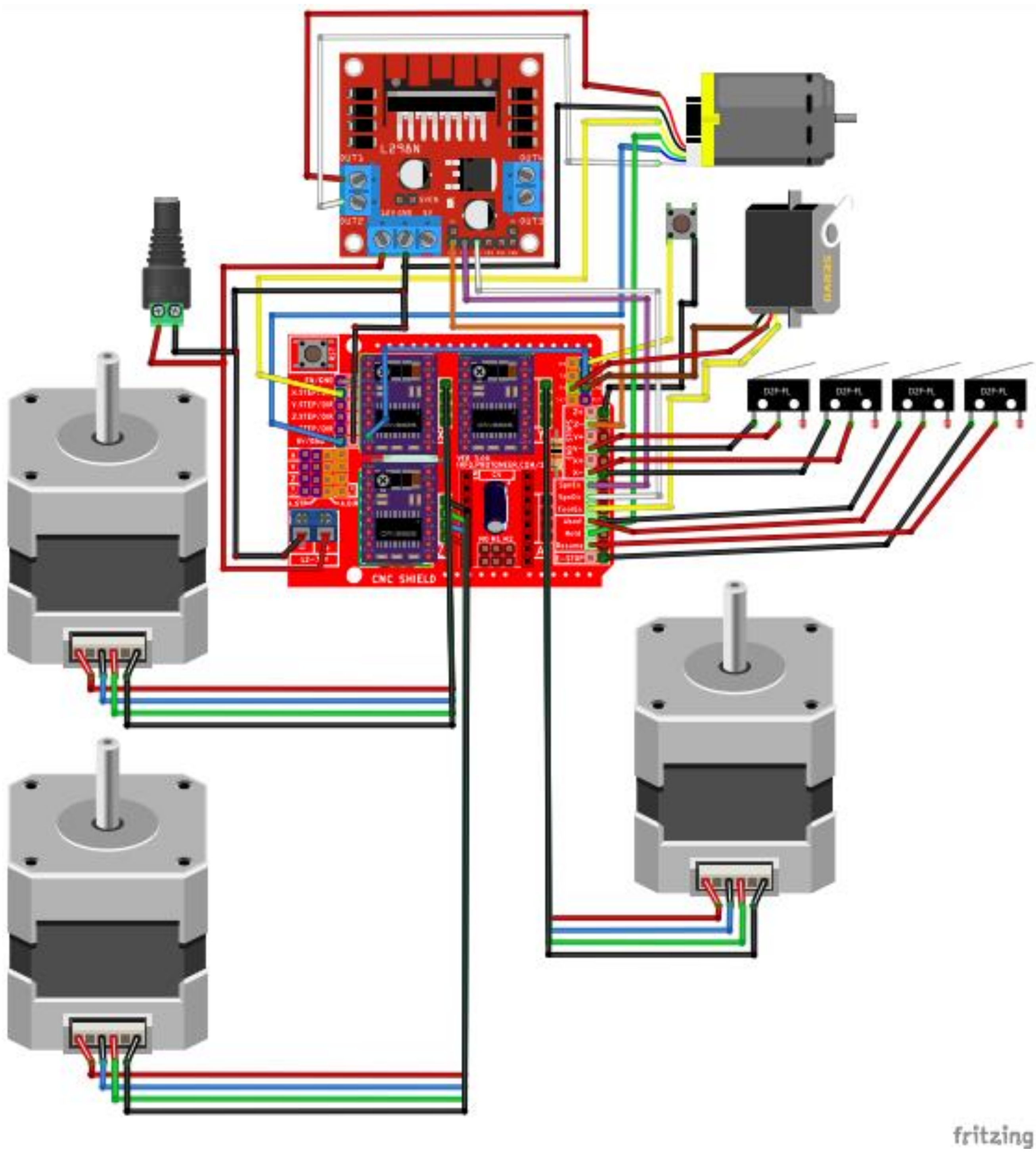
As soon as we connect the power, the robot performs a homing and returns to the initial position. After that, the feeding mechanism performs a homing and returns to the initial position. After that, the laptop takes pictures of each part separately to move to the next time once the products are captured correctly and the word “Done” is displayed, then after. The laptop then gives a sequence of the location of each product, so the robot collects similar products on this basis, assembles them, and then transports them to their storage location.

We use two microprocessors (Arduino uno - laptop) , the two microprocessors connecting with to each other with UART protocol via USB.

- **Flow chart:**


Start
Homing for Robot
Check Robot System?
yes
Robot OK?
no
Proceed to Homing for Feeding System
Check Robot Again
Check Feeding System?
yes
Feeding System OK?
no
Proceed to Camera Sorting
Check Feeding System Again
Sorting
Camera Sorting
Rotate Rotary Table
Detect Parts
Repeat until 8 parts detected
Pick Nearest Base
Place Base for Assembly
Pick Nearest LED
Assembly Station
Place Assembly in Storage
repeat 4 times
Stop

- **the schematic electrical circuit:**



- **The PID controller on DC motor in feeding:**

We Utilize sensors like encoders to measure the motor's current position, providing feedback for the control system.

1. **Proportional (P) Control:** Adjust motor speed based on the difference between desired and current position, increasing or decreasing power accordingly to move closer to the setpoint.

2. **Integral (I) Control**: Accumulate errors over time to eliminate steady-state error, correcting for any system bias preventing the motor from reaching the desired position.

4. **Derivative (D) Control**: Anticipate changes in error rate to dampen overshoot and oscillations, adjusting power based on the rate of change of error.

5. **Tuning Parameters**: Crucially adjust PID parameters to balance responsiveness, stability, and accuracy, typically starting with proportional gain and incrementally adding integral and derivative terms while observing system response.

6. **Continuous Adjustment**: Once tuned, the PID controller continuously computes a control signal based on the error, ensuring the motor reaches and maintains the desired position accurately and efficiently.

In essence, PID control offers a versatile and effective method for precise position control of DC motors, crucial for various applications like robotics and automation systems.

In our case We chose it due to previous experiences with PID values for same motor and modified it with our case.

	Kp	Ki	Kd
Value	5.2	0.001	1.0

Pode blot between desired and set point:

➢ The final product and problems faced:

Final product

- **problems faced:**

1- the small error on PID controller and This causes the product not to be caught by the robot.

We made a great effort to avoid this error by controlling and changing PID parameters, but we were not able to remove it due to the poor quality of the motor.

2- The machine vision camera is affected by external factors such as light, so it must be adjusted before conducting the experiment.

3- The problems of **MATLAB** program as error, we can't solve it.

4- The gripper when its puck up the products, we solve it by change motor with high torque to overcome the friction between links and bolts and nuts.

5- Modifying the CNC shield to can use one Arduino, That modification is to remove the Step-X motor pin 2 and connect it to the pin 0, so that we can use the pin 2 with interrupt with DC encoders.