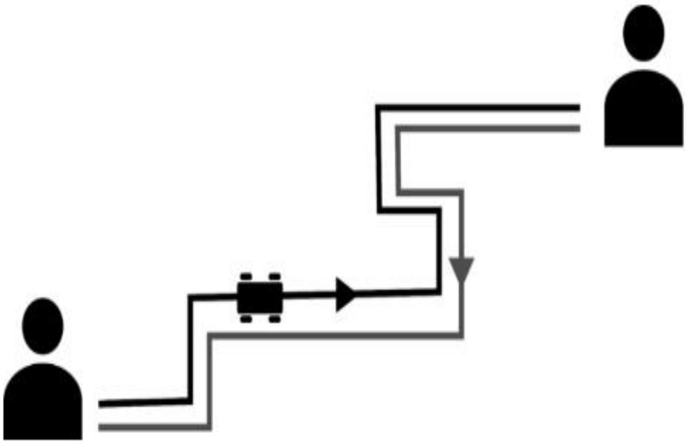
Table of Content



Introduction

A simple To and Fro motion robot is a path remembering car that repeats the action or path that is stored in its memory. This type of robot is often used in manufacturing and assembly lines to perform tasks such as picking up and placing objects. The robot's software can be programmed to define the direction, speed, and distance of the movement. For example, the robot can be programmed to move forward for a certain distance, then reverse its direction and move backward for the same distance. To-and-fro motion robots are often used in conjunction with other robots and automation systems to create a highly efficient production line. They are also used in a variety of other industries to perform repetitive tasks that require precision and accuracy.

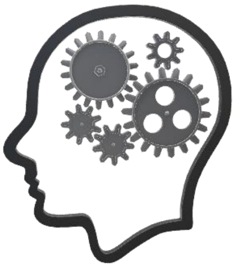
Applications

1. **Material handling:** Robots can be designed to perform a back-and-forth motion to move materials from one location to another in factories and warehouses.
2. **Assembly line automation:** They can be programmed to move in a repetitive "to and fro" motion to perform specific tasks, such as assembling parts or picking and placing items on a conveyor belt.
3. **Inspection and testing:** They can be designed to move in a back-and-forth motion to inspect and test parts and components for quality control purposes.
4. **Cleaning and maintenance:** They can be programmed to clean and maintain floors, walls, and other surfaces.
5. **Medical applications:** They can be designed to perform precise and controlled movements for surgical procedures, such as laparoscopy, which involves inserting a robotic tool into a patient's body and moving it back and forth to perform the operation.

Required Components

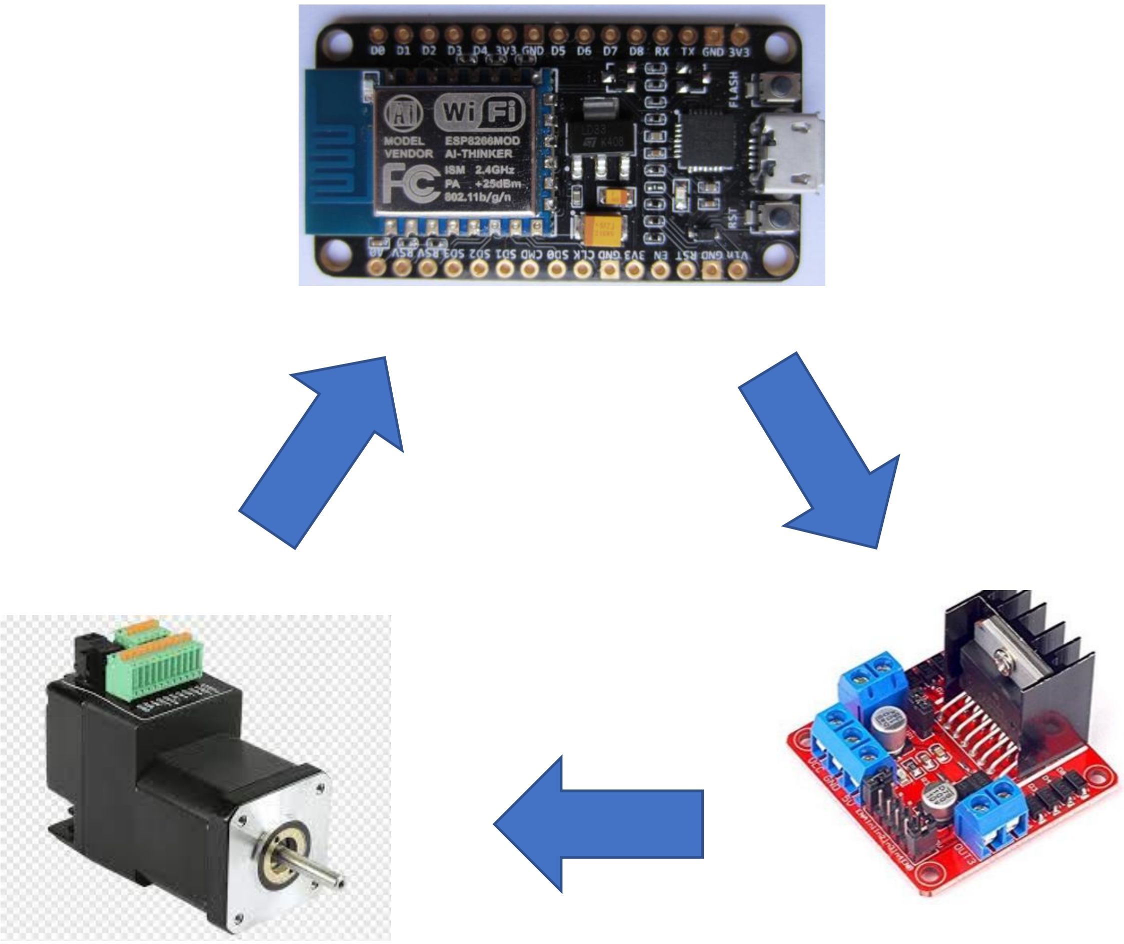
Sno.	Item	Quantity
1.	Acrylic Sheets 5MM (2x2 feet)	1
2.	6-12 V Geared- Encoder Motor 60 RPM	2
3.	Microcontroller board (Nord MCU)	1
4.	75mm rubber wheels	2
5.	Castor wheel	1
6.	Jumper wire M-M, F-F ,M-F	As per requirement
7.	Buck Converter	1
8.	XT 60 M-F Connector	4
9.	Copper Clad (1x1 feet)	1
10.	Battery 12V LIPO	1
11.	M2, M4 Nut & Bolt	As per requirement
12.	L Bracket (20x20 profile)	As per requirement
13.	Motor Driver L298/other	1
14.	Aluminium Profile (20x20 mm)	As per requirement
15.	Glossy paper	2
16.	Allen Bolt/Screws	As per requirement

Pseudo Code(logic)

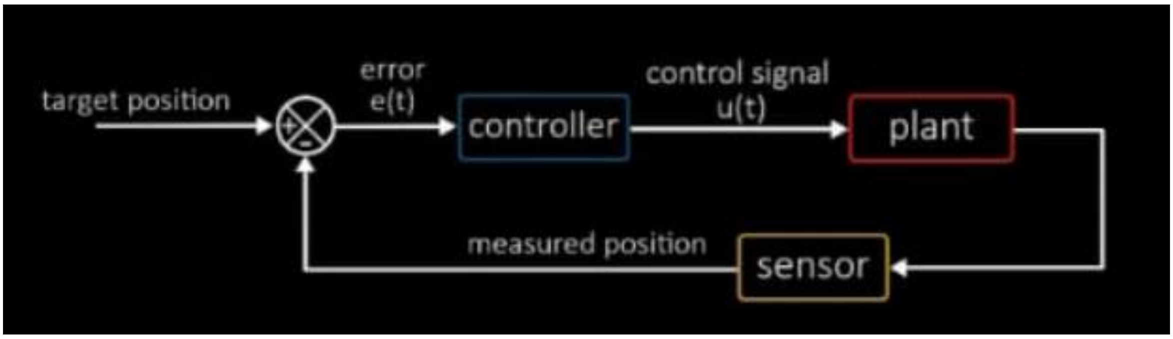


```
// Set up pins for motor control
// Set up pins for memory storage
// Set up variables for memory storage
void setup() {
  // Initialize motor pins
  // Initialize memory pins
  // Initialize memory array
  }
void loop() {
  // Read memory values and store in array
  // Move forward
  // Move backward
  // Update memory index
  // If memory index exceeds memory size, reset to 0
  // Retrace path stored in memory
  }
}
```

Working



Feedback control Loop



- **<u>PLANT:</u>** MOTOR AND MOTOR DRIVER
- **<u>SENSOR:</u>** THE ENCODER
- **<u>TARGET POSITION</u>** : FINAL LOCATION

  E(t)= target position – measured position
- **<u>CONTROLLER</u>**: IT WILL TRY TO REDUCE THE ERROR TERM BY ROTATING THE MOTOR i.e REACHING THE TARGET POSITION

  IT WILL SEND A CONTROL SIGNAL u(t) TO THE PLANT

  -PID (PROPORTIONAL INTEGRAL DERIVATIVE) CONTROL:

  $u(t)= k_p e(t) + k_i \int e(t)dt + k_d(de/dt)$

The proportional signal is used to reach the target position, the derivative and integral terms re typically used to smooth out the control system response. The three constants k_p, k_i, k_d determine how strongly each term is represented in the control loop, we can adjust these constant to tune the response.

Computer Aided Design (CAD)