ARDUINO CONTROLLED SELF-BALANCING ROBOT USING BLUETOOTH

A.B.Raghul[1], S.Kalaimani[1], A.B.Rakkesh[3], V.Arunachalam[4], A. Arun[5]

[2]Assistant Professor, Department of Mechatronics, Sri Manakula Vinayagar Engineering College, Madagadipet, Puducherry, India

[1345]UG student, Department of Mechatronics, Sri Manakula Vinayagar Engineering College, Madagadipet, Puducherry, India

---***---

Abstract - *The research seeks to explain the balancing capability of the robot even under external disturbances The study seeks to explain the advantages as well as disadvantages possessed through the application of stepper motor. The study seeks to explain elaborately about the uses of self balancing robot. The researcher has used secondary data to extract resources regarding the completion of the study. At final, the comparative results of and gyroscope are derived.*

Key Words: self-balancing robot, Stepper motor, Arduino UNO, Bluetooth

1.INTRODUCTION

Self-balancing robot is defined as a robot which has the capability to balance itself on two wheels and get controlled by the user even under external disturbances. Self-balancing robot consisting of both hardware and software components such as housing, motors, and Bluetooth module. Self-balancing robot is generally used to carry the hazardous substances like chemicals in chemical industry and used in vehicles to safeguard the physically impaired persons.

1.1 APPLICATION

Other applications of the Self-balancing robot such as military, medical applications and it is used for gardening which is controlled by the Internet of things(IOT) , airports for carrying luggage , automatic trolleys in the malls and supermarket. The main purpose of the Self-balancing robot is to oppose the external force from the environment and maintaining its upright position. Gyroscopic sensor(MPU6050) helps to retain its balanced position from the opposite forces with the help of Arduino controller which acts swiftly on the calibrated tilt angle.

1.2 ALGORITHM

The various methods of algorithms used are Kalman's filter algorithm, LIDAR based G-mapping algorithm and the artificial neural networks. The PID algorithm is an excellent method of building a control system PID stands for Proportional, Integral, Derivative. The Proportional, Integral, Derivative(PID) parameters such as Kp, Ki, Kd are used to improve the transient response, improve the steady state response and to improve the transient response settling time and rising time respectively. The Proportional, Integral, Derivative(PID) helps to provide the Self-balancing robot to make fast and stable response even in steepy surfaces.

2.COMPONENTS USED

The various components which are used to in the self-balancing robot are

- Rechargeable Battery
- Gyroscope sensor
- Bluetooth module
- Stepper motor
- A4988 stepper motor driver
- Arduino Nano
- Arduino IDE
- Bluetooth terminal

An array of parts is usually included in a self-balancing robot, such as: the robot's main frame, or chassis, is where all other parts are mounted. Space for mounting wheels, motors, and other gear is typically included. Typically, self – balancing robots use two stepper motors, one for each wheel. With the help of these motors, the robot can move its wheels. The robot's wheels allow it to move and get traction. Often, motor shafts or gears are used to connect them to the motors. These electrical parts regulate the motor's speed and orientation. They adapt the power provided to the motors based on the signals they get from the microcontroller (such an Arduino). The robot's microcontroller, which processes sensor data, is its brain.

The Bluetooth module is used to control the robot forward and backward wirelessly. And the Bluetooth

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 04 | Apr 2024 www.irjet.net p-ISSN: 2395-0072

terminal software is implemented by connecting the personal computer of laptop IP address to Bluetooth module. The range of the Bluetooth module is nearly from 0<100 metres.

2.1 RECHARGEABLE BATTERY

Self-balancing robots frequently use rechargeable batteries because of their portability, energy density, and capacity to supply the required power for the robot to function. Rechargeable batteries are commonly integrated into self-balancing robots. In order To run their motors, sensors, and control systems, self-balancing robots need a power supply. Because of their high energy density, rechargeable batteries—especially lithium-ion batteries—are preferred because they can supply enough power while maintaining an acceptable total weight for the robot .The self-balancing robot's unique requirements determine the battery voltage and capacity to use. More powerful motors might benefit from higher voltage batteries, while greater capacity batteries offer longer running intervals between charges.

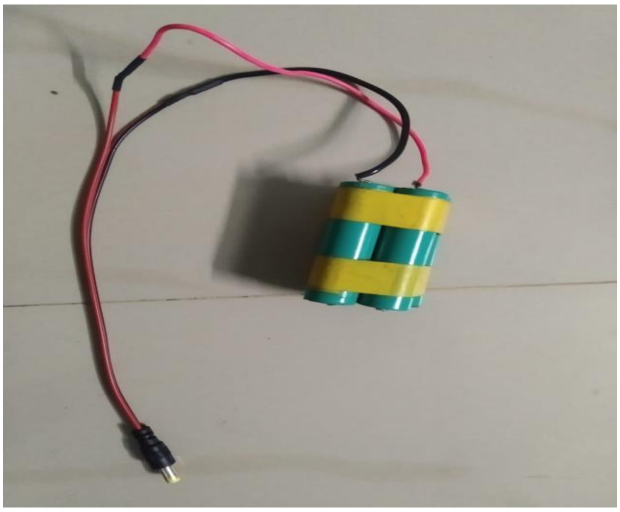

Fig-2.1: Rechargeable Battery

2.2 GYROSCOPE SENSOR

A gyroscope sensor is essential to a self-balancing robot's ability to sense its direction and keep its balance. The rate of rotation around the gyroscope sensor's axis is measured by the sensor. The sensor continuously measures these rotational changes to ascertain the orientation of the robot in relation to its initial position. The robot's control system receives input from the gyroscope sensor and utilizes it to modify the motor speeds as necessary. For instance, if the robot begins to tilt forward, the gyroscope recognizes this abnormality in orientation and notifies the control system to accelerate the motors propelling the wheels backward, reversing the tilt and preserving equilibrium.

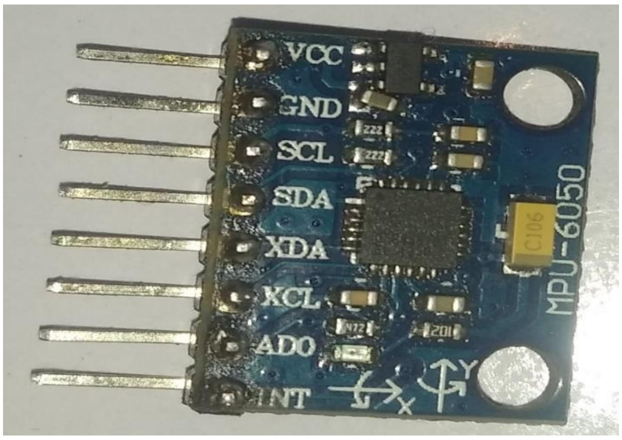

Fig-2.2: Gyroscope Sensor

2.3 BLUETOOTH MODULE

For wireless connection between the robot and a smartphone or other Bluetooth-enabled devices, self-balancing robots frequently use the HC-05 Bluetooth module. Using an app on a smartphone or a separate remote-control device, users can operate the self-balancing robot remotely with the HC-05 module. To control the robot's motions, users can transmit commands like stop, left, right, forward, and backward. Users can modify the PID controller gains, maximum speed, and turning sensitivity of the self-balancing robot via the Bluetooth connection. This allows the behavior of the robot to be adjusted to accommodate various operating situations or preferences.

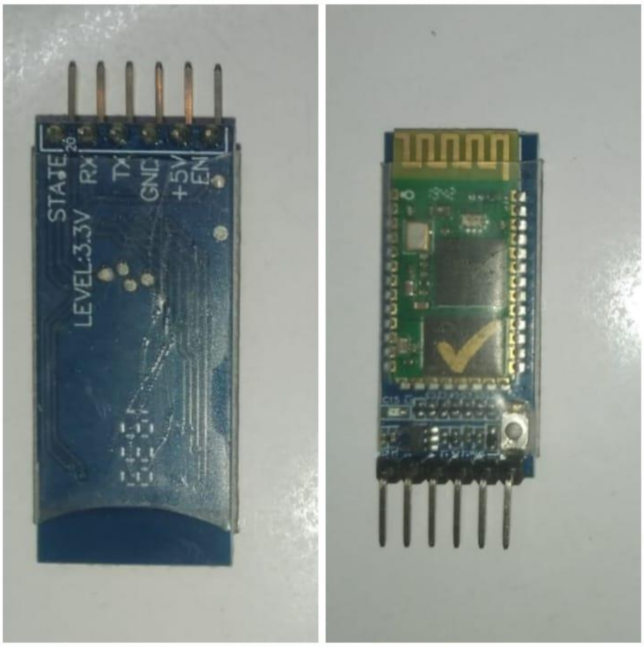

Fig-2.3: Bluetooth Module

2.4 STEPPER MOTOR

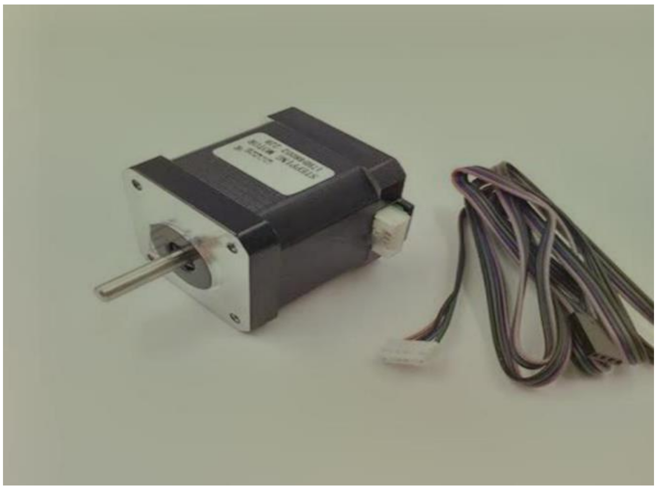

Fig-2.4: Stepper Motor

A self-balancing robot can use a stepper motor, albeit this is less popular than employing brushless DC motors or DC motors. Although stepper motors present a unique set of difficulties, they also have certain qualities that make them appropriate for particular uses. Using a stepper motor in a self-balancing robot can be approached in the following general ways. An inertial measuring unit (IMU) of some kind is usually used by self-balancing robots to determine their tilt angle. The robot's balance is then maintained by using this data to regulate the motors .Stepper Motors is used for the ability to precisely place and maintain a position without constant feedback is possessed by stepper motors

2.5 MOTOR DRIVER

The complex mechanics of a self-balancing robot heavily rely on the A4988 stepper motor driver. By employing a pulse-width modulation (PWM) control system, the A4988 effectively manages the power provided to the stepper motors, guaranteeing accurate and stable movement. With features like microstepping, which breaks up each step into smaller amounts, this driver allows for smoother mobility and more precise control over the robot's balance. The A4988's microstepping capabilities is invaluable in the context of a self-balancing robot, where exact adjustments are critical for preserving equilibrium. Moreover, the integrated safety features prevent overheating and overcurrent, so preventing possible harm to the motor and the entire system. The A4988 stepper motor driver stands out as a key element in due to its adaptability, dependability, and capacity to provide steady performance.

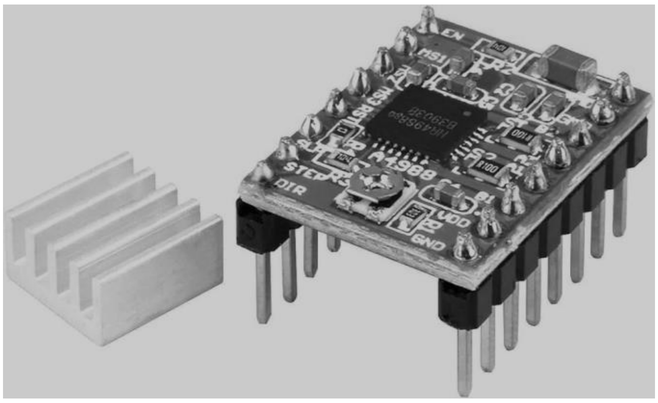

Fig-2.5: Motor Driver

2.6 ARDUINO NANO

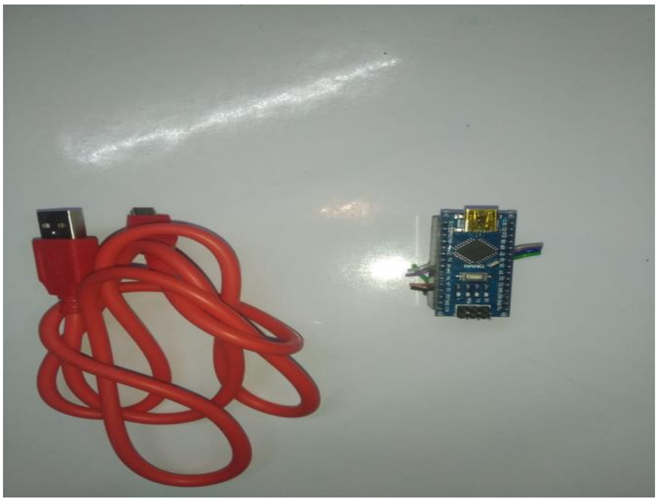

Fig-2.6: Arduino NANO

An Arduino Nano can be used in a self-balancing robot to provide a small and flexible microcontroller platform for managing several aspects of the robot's functioning .The orientation, acceleration, and wheel speed of the robot may be measured by the Arduino Nano by utilizing sensors like gyroscopes, accelerometers, and encoders. The Arduino Nano can produce PWM (Pulse Width Modulation) signals, which can be utilized to regulate the speed of DC geared motors with the use of a motor driver like the L293D. The Arduino Nano can control the PWM signals to maintain balance and speed of the motors by modifying them in response to sensor feedback.

2.7 ARDUINO IDE

To build the code that governs the behavior of the self-balancing robot, programmers utilize the Arduino IDE. Usually, this code contains algorithms for reading sensor data, putting control techniques (like PID control) into action, and managing motor speeds. The Arduino IDE is used to compile the code after it has been written. The Arduino IDE makes sure that the code is free of errors and transforms it into instructions to the Arduino microcontroller . The Arduino Nano, a microcontroller board frequently used in self-balancing robots, may be programmed, compiled, and uploaded using the Arduino Integrated Development Environment (IDE) software platform.

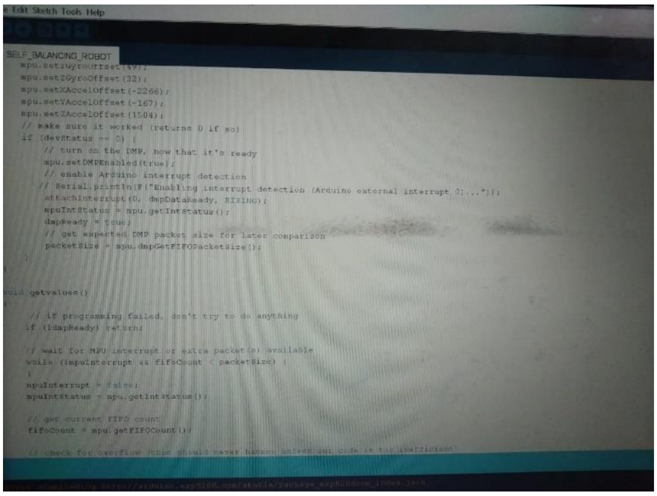

Fig-2.7: Arduino IDE(code)

2.8 ARDUINO BLUETOOTH ROBOT CAR

The intricate interaction between sensors, actuators, and control algorithms is orchestrated by the Arduino Bluetooth robot vehicle software, which acts as the digital brain behind the operation of a self-balancing robot. By using Bluetooth connectivity, this software enables users to easily monitor and manage the robot's motions remotely. It also enables real-time communication between the robot and external devices, like smartphones or tablets. In order to continuously determine the robot's orientation and make exact modifications to preserve balance, complex algorithms within the software evaluate data from onboard sensors, including gyroscopes and accelerometers. In order to adjust motor responses and maintain stable, smooth operation even in dynamic situations, it also integrates feedback loops and PID control algorithms. Self-balancing robots are made possible by the Arduino Bluetooth robot car software's flexible and robust programming.

Fig-2.8: ARDUINO Bluetooth Robot Car

2.9 SPECIFICATION OF THE COMPONENTS

S.No	Components/Equipments	Specifications
1	Rechargeable Battery	Lithium ion (3.3V each)
2	Gyroscope sensor	MPU-6050
3	Bluetooth Module	HC-05
4	Stepper motor	Nema 17
5	Motor driver	L293D
6	Arduino Nano	Atmega 328

Fig-2.9: Specification of the component

3. WORKING PRINCIPLE

Sensors, motors, a microprocessor, and a control algorithm are usually used by a self-balancing robot to keep its equilibrium. Sensor is used to assess its orientation and identify any tilt or inclination, the robot makes use of sensors like gyroscopes, accelerometers, or an Inertial Measurement Unit (IMU).The microcontroller, like an Arduino or Raspberry Pi, processes these sensor signals and determines the present position of the robot as well as the amount of adjustment required to keep it balanced. The third component is the Control Algorithm. This algorithm uses sensor data to calculate the necessary modifications to keep the robot upright. It is commonly a PID (Proportional-Integral-Derivative) controller. Accordingly, it modifies the motors' direction and speed. and the main aim of the self-balancing robot is to balance the Two wheeled Self-balancing robot using Arduino controller to process the data and Bluetooth technology(HC-05 module) to control the robot wirelessly via Bluetooth terminal software. The Bluetooth commands can be given through serial monitor using Arduino IDE software. The Stepper motors(NEMA 17) are used to make a precise movement and reduce vibration.

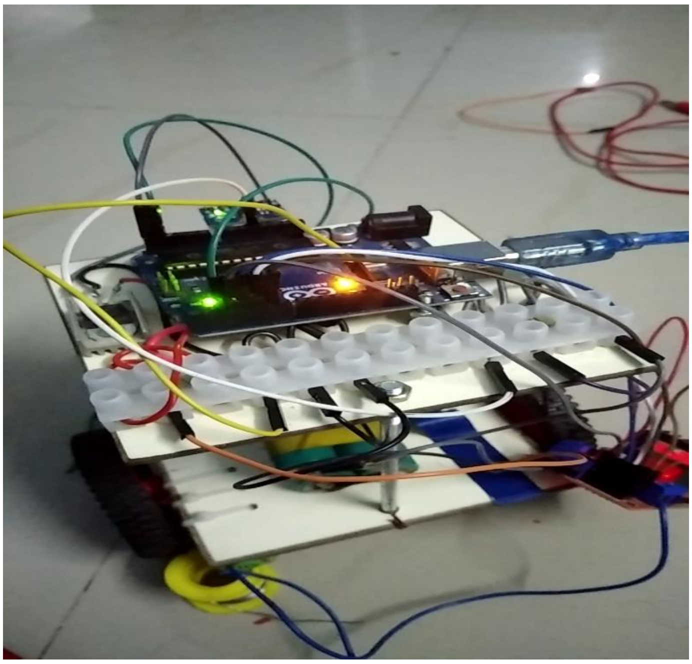

Fig-3: Working of self-balancing robot

3.1 IMPLEMENTATION OF PID ALGORITHM

The acronym PID represents Proportional, Integral, and Derivative. The input for this technique is the received error signal. And the erroneous signal is subjected to the subsequent equation.

U(t) = Kp*e(t) + Kd*d/dt(e(t)) + Ki*integral(e(t)) (1.1)

This involves the computation of the integral and derivative of the error signals. These are then multiplied by the corresponding constants and combined with the constant Kp multiplied by e(t). After that, the actuator receives the output, enabling the system to function. Let us examine each component of the function one by one now. The peak overshoot, settling time, rise time, fall time, and steady state error are all directly impacted by this function. This component lowers the steady state error and rising time. This implies that the system will get at its destination faster.

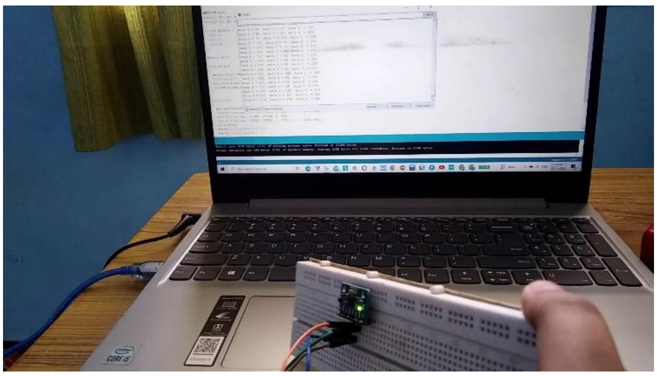

Fig-3.1: Calibration of self-balancing robot

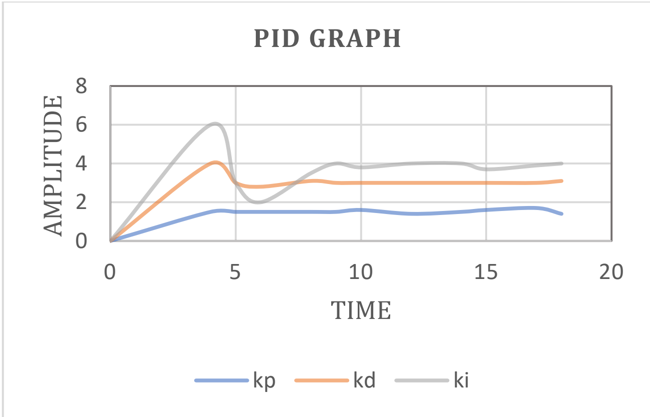

Fig-3.2: Graphical representation of PID

For a self-balancing robot to operate steadily, calibration is essential. This is an in-depth analysis. Mechanical calibration is to Verify that the robot's center of gravity is appropriately positioned above its wheels. This entails ensuring that parts like wheels, chassis, and engines are aligned. Small alignment errors can have an impact on balance. Most self-balancing robots use accelerometers and gyroscopes for measuring tilt and acceleration (see Sensor Calibration). To ensure reliable readings, they must be calibrated by setting their zero values. Software calibration algorithms or manual offset adjustments can be used to accomplish this. PID Tuning is used in the motor balance is maintained by the PID controller modifying them. Proportionate (P) is used to responds to the existing inaccuracy (difference between the intended and actual angle).

Controller Parameters	Tuned	Baseline
Kp	36.2542	355
Ki	138.3051	290
Kd	2.280	43

3.2 BLOCK DIAGRAM

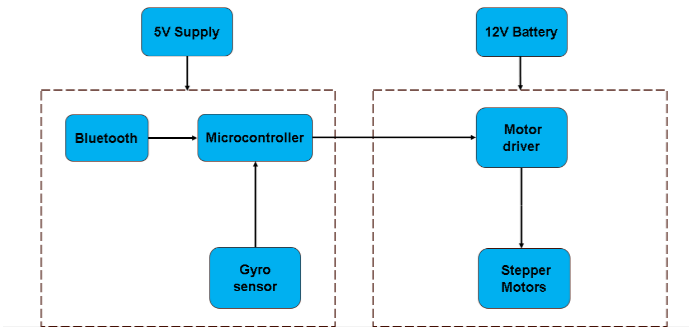

Fig-3.2: Functional block diagram of self-balancing robot

The system is powered by a 5V Supply. The Bluetooth module enables wireless connectivity to other devices, like computers and cellphones. Microcontroller gives the motor driver instructions, interprets sensor data, and manages the robot's general operation. The robot's tilt and orientation are measured via the Gyro Sensor. Motor Driver is used to apply the commands from the microcontroller to the motor(s) in order to operate them. Stepper Motor is usually employed for fine control, like regulating a self-balancing robot's movement or equilibrium. DC Motor is used to power actuators like wheels. The DC motor and stepper motor are powered by a 12V battery.

3.3 CIRCUIT DIAGRAM

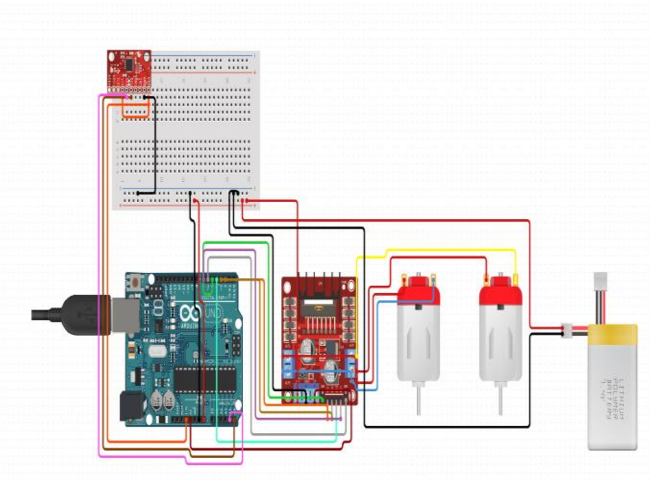

Fig-3.3: Circuit diagram of the self-balancing robot

Depending on the interface of the sensor, either digital or analog input pins are used to connect the gyroscope sensor to the Arduino Nano. Serial communication (usually utilizing UART pins) is used to link the Arduino Nano to the Bluetooth module. To control the rotation of the stepper motor, the motor driver is connected to the digital output pins of the Arduino Nano. The stepper motor's power and control signals are linked to the motor driver's output terminals. The Arduino Nano and the motor driver are both powered by the battery. This circuit diagram serves as the foundation for the hardware architecture of the self-balancing robot, which can be controlled and monitored wirelessly in addition to being able to sense direction, process information, and modify motion to maintain balance.

3.4 FLOW CHART

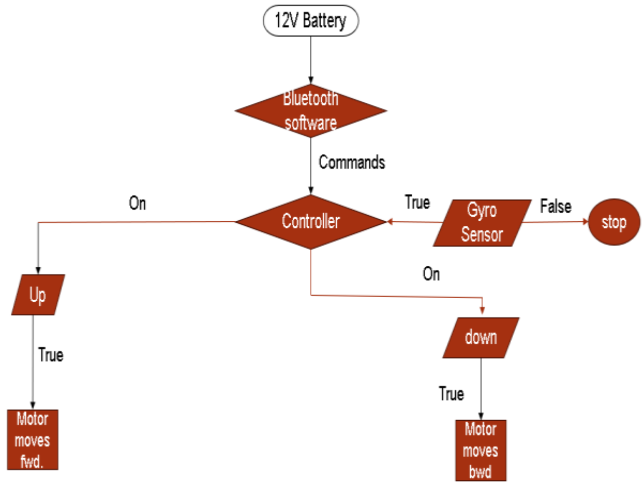

Fig-3.4: Control flow of self-balancing robot

The lithium-ion battery is used in this project to improve the power supply for long hours and it can be recharged faster. The robot angular movements are sensed by the gyroscopic sensor and sends the signal to the microcontroller. The microcontroller manages the motor driver to allow the motor either in clockwise or counterclockwise direction. The stepper motors will run based on the motor driver commands to balancing itself. The Bluetooth module helps to control the self-balancing robot wirelessly. The ultrasonic sensor is used in this project to identify the obstacles in the surroundings. The DC motor is used to make the arm to pick and place the objects itself.

4.CONCLUSION

In conclusion, the self-balancing robot project provides insightful knowledge on wireless communication technologies, robotics, control systems, and sensor integration. To sum up, the self-balancing robot project integrates several parts and technologies to produce a dynamic and communicative system. We can create a robot that can keep its equilibrium on its own by combining gyroscopic sensors, motor drivers, stepper motors, Bluetooth modules, and microcontrollers like Arduino Nano. We have studied the fundamentals of feedback control systems during the project, focusing on the PID (Proportional-Integral-Derivative) algorithm, which is essential for stabilizing the robot because it continuously modifies motor output in response to sensor input. Furthermore, the project gains adaptability and user engagement from the addition of a Bluetooth module, which allows remote control and behavior monitoring of the robot. This project will be successful in achieving its aim to balance a two-wheeled self-balancing autonomous robot based on the inverted endulum model. This self-balancing robot must change its position to maintain balance even an external force applied is not a problem only stability is considered. However, if the robot is in a narrow space, the movement caused by the disturbance can create a real problem.

5. REFERENCES

[1] Iulian Matesica, Mihai Nicolae, Liliana Barbulescu, Ana-Maria Margeruseanu, "Self-balancing robot implementing the inverted pendulum concept," in 15th RoEduNet Conference: Networking in Education and Research, Bucharest, Romania, 2016.

[2] Arduino. (2015). Arduino Software (IDE). (Arduino) Retrieved December 27, 2015, from https://www.arduino.cc/en/Guide/Environment.

[3] Ahasan, M. A., Hossain, S. A., Siddique, A. U., & Rahman M.M(2012). Obstacle invariant Navigation of An Autonomous Robot based on GPS. Khulna University.

[4] Sahib, F., Ullah, M. W., Hasan, M. K., & Mahmud, H. (2013). Obstacle Detection and Object Size Measurement for Autonomous Mobile Robot using Sensor. International Journal of Computer Applications.

[5] Qiu, J.; Wu, J.; Jin, S.; Jin, Y.; Shi, J. Research on Inspection robot of Coal mine Water pump House based on Optimized Fast-SLAM. *Coal Eng.* 2021, *53*, 139–145.

[6] Qiao, L. Research On Control Strategy of Two-Wheeled Self-Balancing Robot. Ph.D. Thesis, Harbin Engineering University, Harbin, China, 2019.

[7] Lin, X.; Zhang, S.; Chen, J.; Zhao, Z. Finite element simulation analysis integrated information model based on Design Reuse. *Comput. Integr. Manuf. Syst.* 2009, *15*, 2296–2302.

[8] Wang, X. Design of Two-Wheel Balancing Vehicle System Based on Genetic PD Control. Ph.D. Thesis, Harbin University of Science and Technology, Harbin, China, 2018.