Khulna University of Engineering & Technology

Department of Electronics and Communication Engineering
Report on ***ECE 3200***

Project: *Design and control Self Balancing Robot.*

Submitted By:

Md. Imran
Roll: 1809037
Batch: 2K18

Bishaw Kirti Chakma
Roll: 1809061
Batch: 2K18

Submitted To:

Md. Khorshed Alom
Lecturer
Department of Electronics and Communication Engineering, KUET.

Abstract:

Balancing robot is a mobile robot with wheels on the right and left side, able to maintain equilibrium. This robot is a development of an inverted pendulum model placed on a wheeled train. GY-521 MPU6050 3-axis gyroscope and accelerometer sensors are used to detect the slope as well as to detect angular velocity and to maintain the position of the robot in a position perpendicular to the earth's surface. The PID control method is used to control the DC motors.
Trial and error determine the value (tune) of the proportional and integral control parameters. The EZ-GUI Ground Station mobile app was used to control the robot's movement through Bluetooth. The robot mobility in this project still needs to be fixed.

Introduction:

A self-balancing robot is like an inverted pendulum, unlike a normal pendulum which oscillates back and forth when a force is applied. An inverted pendulum cannot balance itself. An example of this would be, trying to balance a broomstick on your fingers. You can't stay in one place, you have to move your finger back and forth. This is exactly what the vehicle is doing, it's moving back and forth so that the center of gravity will always remain under the wheels. To drive the motors we need some information on the state of the robot. We need to know the direction the robot is falling, how much the robot has tilted, and the speed with which it is falling.

To get this information we will be using a combination of a gyroscopic sensor and an accelerometer. The sensor I will be using is MPU6050. We combine all these inputs and generate a signal which drives the motors and keeps the robot balanced.

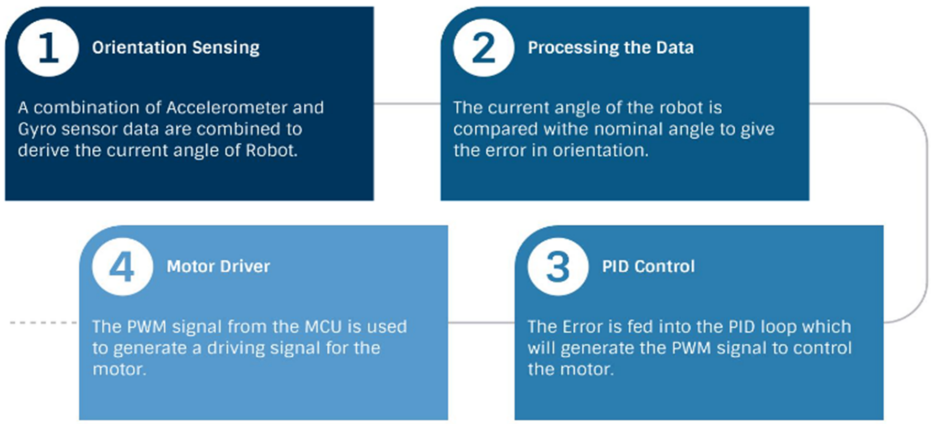


Figure 1. Flow Diagram of a Self-balancing robot

Methodology:

To balance the robot, we don't just need the angle the robot is at, we also need to know the rate at which it is falling. The MPU6050 has a 3-axis accelerometer that measures acceleration and a 3-axis gyroscope that measures angular rate about the three axes.

To measure the angle of inclination of the robot we need acceleration values along y and z-axes. The mpu.getAccelerationZ() function will retrieve the acceleration data from the corresponding registers. And the atan2(y,z) function gives the angle in radians between the positive z-axis of a plane and the point given by the coordinates (z,y) on that plane. With a positive sign for counter-clockwise angles, and a negative sign for clockwise angles.

We read the gyro value about the X-axis, convert them to degrees/sec, and then multiply it with the loop time to obtain the change in the angle. We have the angular velocity of the robot from the gyro sensor. We then multiply that value with the loop time to get the change in angle. Then we add that value to the previous angle to get the current angle of the robot.

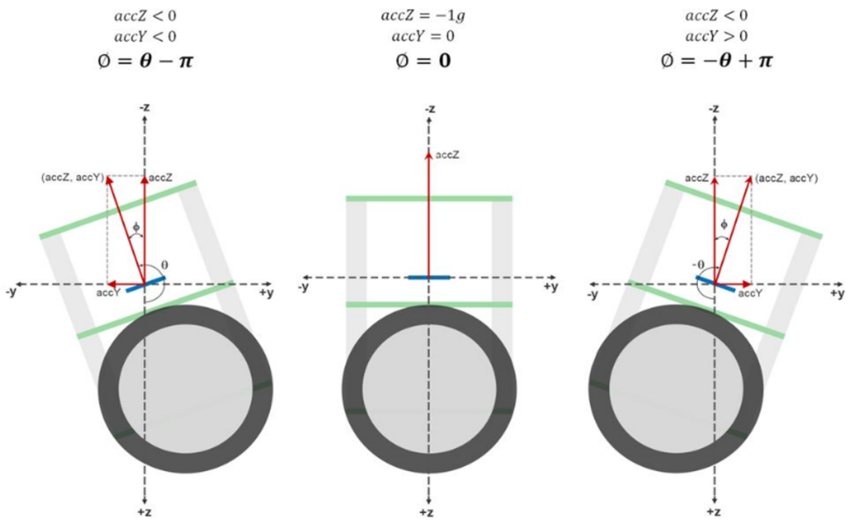


Figure 2. Measuring the angle of inclination

Besides, we employed a PID controller that uses tilt feedback to control the torque of the motors and keep the robot balanced. A PID controller continuously measures a process variable (the tilt of the robot) and calculates an error value (angle from the vertical), which is the deviation of the process variable from some desired, ideal value (0 degrees from the vertical). The controller attempts to minimize this error over time by continuously adjusting a control variable (motor

torque) according to the following equation, where u(t) is the control variable, e(t) is the current error in the process variable, and Kp, Ki, and Kd are coefficients that must be tuned to achieve the desired behavior of the controller:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de}{dt}$$

As seen from the equation the PID control variable depends on the current error (the proportional term), the rate of change of the error (the derivative term), and the long-term bias of the error (integral term), hence PID. This feedback loop, outlined in Figure 1 below, is the core of the robot's balancing behavior.

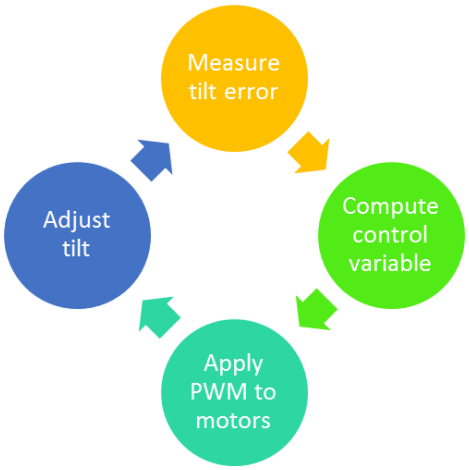


Figure 3. Steps involved in the feedback control loop of the balancing robot

In short Gyro sensor provides the x, y, and z data to a microcontroller. Input data is sent to the microcontroller triggers via Bluetooth from an android device. Arduino nano balances the robot by determining precise data from the PID control system.

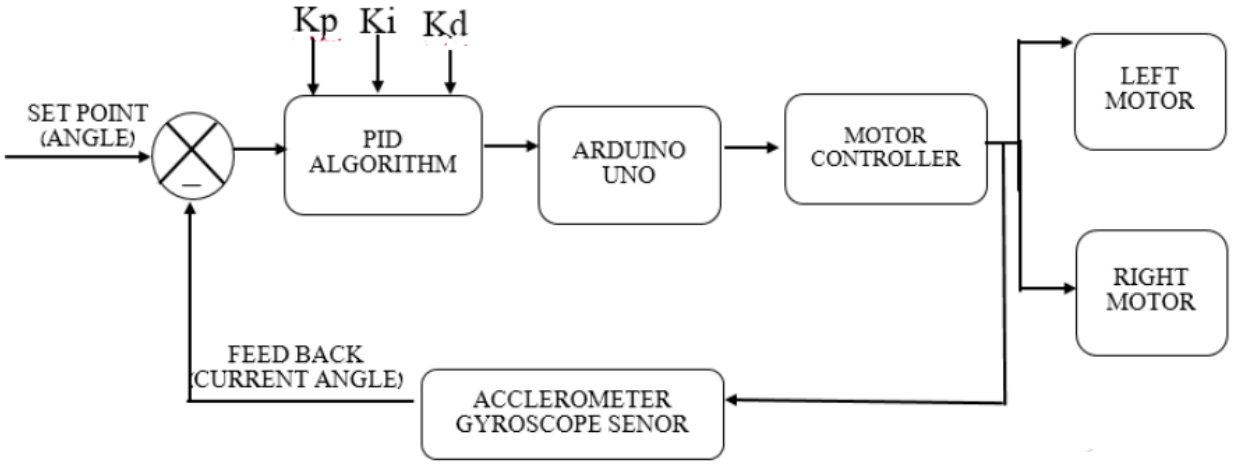
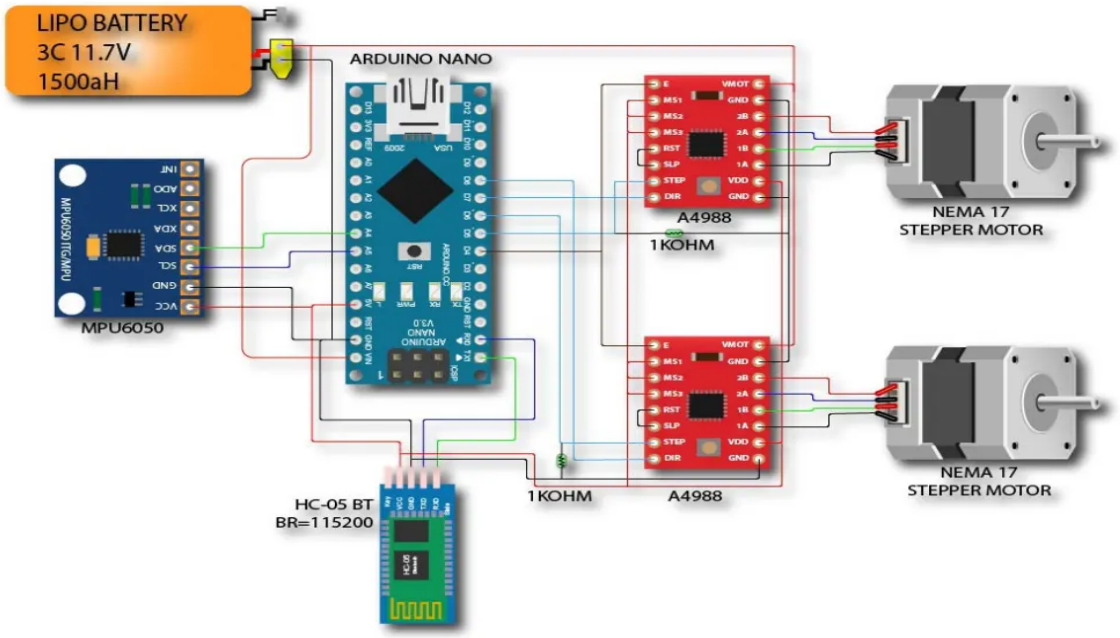


Figure 4. Block diagram of Balancing Robot

Circuit Diagram:



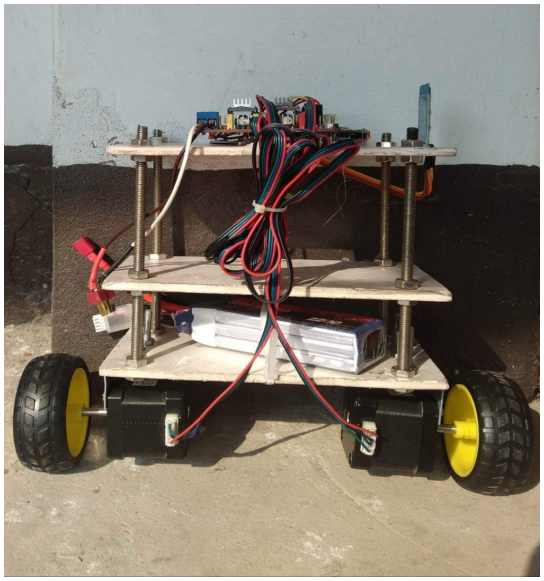


Figure 5. Robot in Balanced Position

Project Requirement:

1. Arduino Nano _____ 1x
2. MPU605 Gyro sensor _____ 1x
3. Nema 17 Stepper motors _____ 2x
4. 75mm Wheels _____ 2x
5. A4988 Stepper driver IC _____ 2x
6. HC-05 Bluetooth module _____1x
7. Arduino CNC Shield V3_____ 1x
8. Cardboard, LIPO Battery 2200 mAh _____1x
9. 145 mm Rods _____ 4x
10. Wire, nuts and bolts (as required)

Code:

Program for adjusting motor dc speed;

```
//PID calculation for motor drive
pid.Compute();
motorController.move(output, MIN_ABS_SPEED);
unsigned long currentMillis = millis();
if (currentMillis - time1Hz >= 1000)
{
loopAt1Hz;
time1Hz = currentMillis;
}
if (currentMillis - time5Hz >= 5000)
{
loopAt5Hz();
time5Hz = currentMillis;
}
}
```

Data:

Gyro sensor testing aims to look at the value of the output which is derived from the sensor.

Table 1. Data Trailer gyrosensor mpu-6050

X	Y	Z
3:29	71.91	-11.46
3:26	71.96	-11.34
3:19	72	-11.22
3:17	72.04	-11.1
3:16	72.07	-10.99
3:13	72.12	-10.86
3:29	71.91	-11.46
3:26	71.96	-11.34

Android Application Testing:

EZGUI android app has been used in test by downloading it from google play store and was connected to a Bluetooth device HC-05. After connecting, the input data command to turn to balance the robot on was given via the app, there was feedback in the form of a voice of the android application.

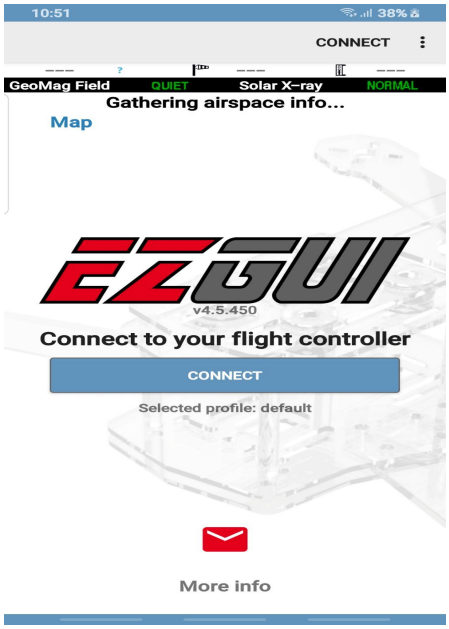


Figure 5. Application on Android

Useability & Application:

In terms of general useability, we tried to make operating this robot as simple as possible. The user simply has to connect the robot to power and send commands via Bluetooth from the android app. It is recommended that the robot lay down on a flat surface as the robot is powered on because that allows it to successfully complete the gyroscope calibration procedure before it begins operation.

Some of its applications are:

1. The vehicle can rotate in place to instantly change its direction of motion and precisely navigate tight spaces that a three or four-wheeled robot cannot.
2. It can be used in restaurants to serve food to customers just like waiters do today.
3. They can also be used in hospitals to carry necessary medical supplies to patients autonomously.
4. They can be used in malls for welcoming people entering the malls.

Future Work:

1. Improve the design by using lightweight material resulting in better balancing.
2. Fix the wobble by setting a minimum PWM duty cycle that is just below the threshold at which the motors begin to spin.
3. Ability to carry load on its top while balancing.
4. Implement automatic tilt calibration so that the robot will always remain balanced even if its center of gravity is shifted, requiring no action by the user.
5. Implement manual remote steering of the robot, commanding it to move forward, backward, and rotate clockwise or counter-clockwise.
6. Add a light sensing system to the bottom of the robot and make it autonomously follow a dark path drawn on the ground.

Conclusion:

Based on the measurement results we concluded that our robot with a gyro sensor is working properly and able to stand upright. Gyro sensors are capable of processing input to balance the communication between the robot and the android and the microcontroller using Bluetooth successfully using the data lines RX and TX to trigger the start of robots. But to control the direction of motion of the robot is still a difficulty for synchronous commands from android and gyro sensor to the dc motor.