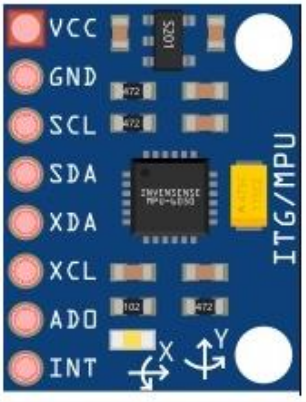
	ITER, SIKSHA 'O' ANUSANDHAN (Deemed to be University)	**Assignment**

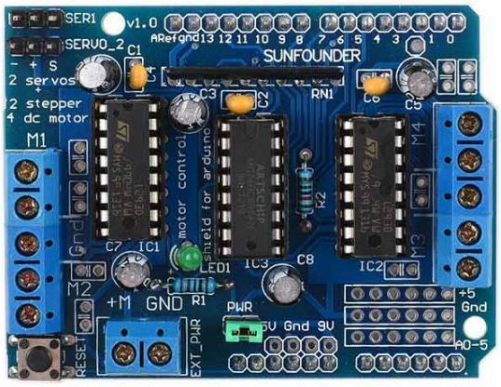
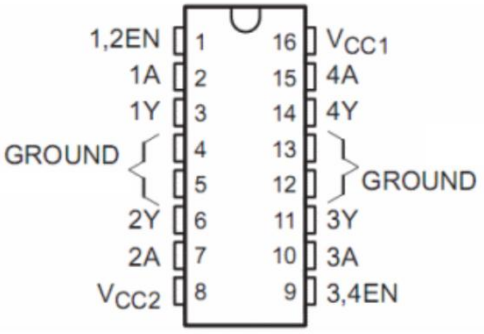
Branch	**Computer Science and Engineering,**	**Programme**	**B.Tech**
Course Name	**Practical Robotics Projects with Arduino**	**Semester**	**7th**
Course Code	**CSE-4571**	**Academic Year**	**2025/Odd**
Assignment-2	**Topic- Self Balancing Robot**		

Learning Level (LL)	**L1**: Remembering	**L3:** Applying	**L5**: Evaluating
	L2: Understanding	**L4**: Analyzing	**L6**: Creating

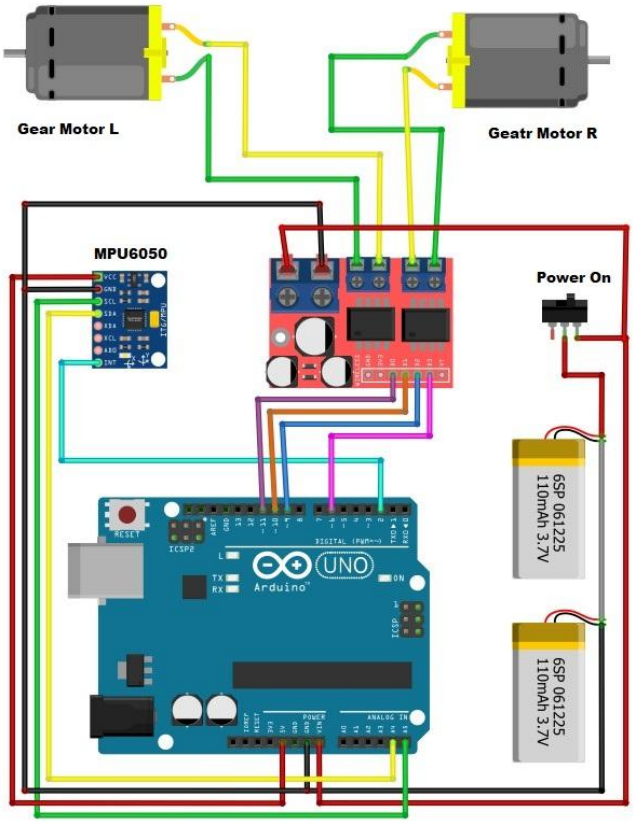
The Arduino Two-Wheel Self-Balancing Robot uses an Arduino UNO, MPU6050 sensor, and L298N motor driver to control two DC geared motors powered by a 7.4V Li-ion battery. The MPU6050 measures tilt angles, and the Arduino uses a PID control algorithm to keep the robot balanced on its wheels.

Q's	Questions	COs	LL
1.	What principle allows a two-wheel robot to balance itself? Explain. Ans:	**CO1**	**L1 &L2**
2.	What is the function of the MPU6050 sensor? What type of motors are used in the robot? Ans:	**CO1**	**L1 &L2**
3.	Draw or describe the wiring connections between Arduino UNO and MPU6050 (include SDA, SCL, VCC, GND, and INT pins). 	**CO1**	**L1 &L2**

	Ans:		
4.	A. Which Arduino libraries are commonly used to interface MPU6050 ? Ans: B. Write an Arduino code snippet to initialize the MPU6050 and verify its connection through the Serial Monitor. Ans: C. Explain the role of the Digital Motion Processor (DMP) in the MPU6050. Ans: D. What is the difference between raw data output and DMP-processed data output from the MPU6050? Ans:	**CO1**	**L1 &L2**

5.	A. Why can't the Arduino UNO drive DC motors directly without a driver like the L293D? Ans: B. What is the maximum voltage and current rating for each motor channel in L293D? Ans: C. What is the purpose of the Enable pins (EN1, EN2) in L293D, and how do they affect motor operation? Ans: D. Write an Arduino code snippet to rotate a motor forward and reverse using digitalWrite() and delay().   Ans:	**CO2**	**L2 & L3**

6.	Write an Arduino sketch to read raw accelerometer and gyroscope data from the MPU6050 and observe how values change when you tilt the sensor. What does this reveal about its axes? Ans:	**CO2**	**L2 & L3**
7.	Define the variables for PID (Kp, Ki, Kd, Input, Output, Setpoint). Write the Arduino code snippet that computes PID output. Explain how each term (P, I, D) influences balancing. Ans:	**CO2**	**L2 & L3**

8.	Construct the full circuit of Arduino UNO, MPU6050, and L298N motor driver. Label all connections clearly. How do SDA and SCL pins differ from digital I/O pins?  Ans:	**CO2**	**L2 & L3**

9.	Write a simple sketch to rotate both DC motors forward, backward, left, and right using PWM signals. Measure the actual motor speed at different PWM values using a tachometer or estimation method.
	Ans: *int ENA = 3; // Left motor speed (PWM)*
	int IN1 = 5;
	int IN2 = 6;
	
	int ENB = 11; // Right motor speed (PWM)
	int IN3 = 9;
	int IN4 = 10;
	
	int speedVal = 180; // PWM (0–255)
	
	void setup() {
	pinMode(ENA, OUTPUT);
		CO2	**L2 & L3**

```
 pinMode(IN1, OUTPUT);
 pinMode(IN2, OUTPUT);

 pinMode(ENB, OUTPUT);
 pinMode(IN3, OUTPUT);
 pinMode(IN4, OUTPUT);
}

void loop() {
 // Forward
 analogWrite(ENA, speedVal);
 analogWrite(ENB, speedVal);
 digitalWrite(IN1, HIGH); digitalWrite(IN2, LOW);
 digitalWrite(IN3, HIGH); digitalWrite(IN4, LOW);
 delay(2000);

 // Backward
 digitalWrite(IN1, LOW);  digitalWrite(IN2, HIGH);
 digitalWrite(IN3, LOW);  digitalWrite(IN4, HIGH);
 delay(2000);

 // Left Turn
 analogWrite(ENA, speedVal);
 analogWrite(ENB, speedVal);
 digitalWrite(IN1, LOW);  digitalWrite(IN2, HIGH);   // Left motor reverse
 digitalWrite(IN3, HIGH); digitalWrite(IN4, LOW);    // Right motor forward
 delay(2000);

 // Right Turn
 digitalWrite(IN1, HIGH); digitalWrite(IN2, LOW);    // Left forward
 digitalWrite(IN3, LOW);  digitalWrite(IN4, HIGH);   // Right reverse
 delay(2000);

 // Stop
 analogWrite(ENA, 0);
 analogWrite(ENB, 0);
 delay(1000);
}
```

10.	Write and explain the Arduino statement that maps the PID output (angle correction) to motor PWM values. Why is the map() function used, and how can incorrect mapping affect performance? Ans:	**CO2**	**L2 & L3**
11.	A. The MPU6050 DMP sends data using interrupts. Describe how interrupts improve sensor reading efficiency. Write the ISR (Interrupt Service Routine) function prototype used in the main sketch. Ans: B. Add Serial printing of pitch angle and PID output values. Observe and plot the values over time (using Excel or Serial Plotter). What pattern indicates a stable system? Ans:	**CO3**	**L3 & L4**

12.	A. Discuss how the physical design of your robot (wheel size, chassis weight, battery placement, sensor position) influenced its ability to balance. Explain what changes you made to improve stability and why they worked. Ans: B. Describe at least **five** major problems you encountered while building the robot (hardware or coding) and explain how you solved each one. Examples may include sensor drift, wrong motor wiring, unstable PID output, dead zones in PWM, incorrect interrupts, etc. Ans:	**CO4**	**L4 & L5**

13.	A. Explain how battery voltage affected the robot's performance. Ans: B. Discuss what happened when voltage dropped, how motor torque changed, and why balancing became unstable. Ans: C. What steps did you take to manage power issues? Ans:	**CO5**	**L4 & L5**
14.	A. Describe how the motor driver converts Arduino signals into motor movement, including direction control and PWM speed control. Ans: B. Explain how you tested motor direction, calibrated both motors to rotate equally, and what problems you faced if one motor moved faster or slower. Ans:	**CO6**	**L5 & L6**

Assignment 2	**Topic: Self - Balancing Robot**	**Date of Assignment2: 24.11.2025**	**Date of Submission: __.11.2025**

Note:

1. Assignment carries a weightage of **20 marks out of 100**
2. Course outcome CO1 to CO6 was covered.

Course Outcomes	CO1	Understand the fundamentals of Arduino hardware and software.
	CO2	Interface various sensors and actuators with Arduino.
	CO3	Apply programming logic to control robotic systems.
	CO4	Design and build basic to intermediate level robotics projects.
	CO5	CO5 Demonstrate problem-solving and debugging skills in robotics.
	CO6	Collaborate in teams to plan, execute, and present robotics projects