KLE TECH

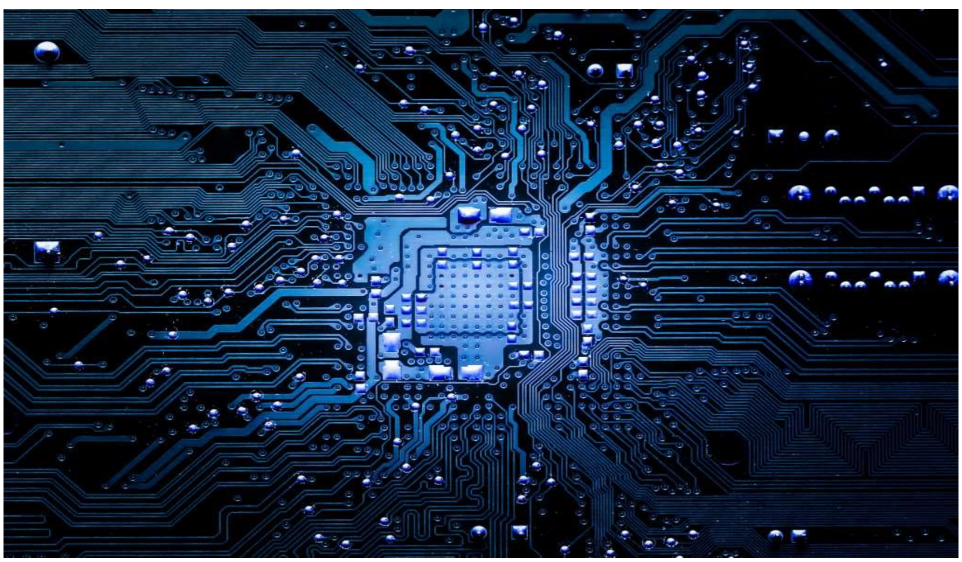
KLE Technological University

Creating Value Leveraging Knowledge

## DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

# POWER ELECTRONICS

## ASSIGNMENT: Closed-Loop Speed Control of a DC Motor Using PID Algorithm




**Name: Malhar Kulkarni**

**USN: 01fe21bee016**

# INTRODUCTION

## DC Motor Control and Its Applications

Direct Current (DC) motors are widely used in various applications due to their simple construction, ease of control, and wide range of speed and torque capabilities. DC motor control is an essential aspect of many industrial, robotic, and consumer electronics systems. Some common applications of DC motor control include:

- Robotics and automation: DC motors are used to power the movement of robotic arms, wheels, and other actuators.
- Industrial machinery: DC motors are used to drive conveyor belts, fans, pumps, and other industrial equipment.
- Home appliances: DC motors are found in household items like washing machines, blenders, and power tools.
- Transportation: DC motors are used in electric vehicles, elevators, and door/window actuators.

## Purpose and Relevance of the Project

The purpose of this project is to develop a comprehensive DC motor control system that can accurately regulate the speed of a DC motor. This project is relevant to DC motor control as it explores the implementation of a closed-loop control system to achieve precise speed regulation, which is crucial in many applications where consistent motor performance is required.

## Objectives and Scope

- **Project Goals**

The main objectives of this project are:

1. Speed Regulation: Implement a control algorithm that can accurately regulate the speed of a DC motor to a desired set point.

2. Control Loop Design: Design a closed-loop control system that utilizes feedback from the motor to adjust the control input and maintain the desired speed.

3. Hardware-Software Integration: Integrate the control algorithm with the necessary hardware components, such as a microcontroller, motor driver, and speed sensor, to create a functional DC motor control system.

- **Scope of the Project**

The scope of this project includes the following:

1. Selecting and configuring the appropriate hardware components for the DC motor control system.

2. Developing a control algorithm, such as a Proportional-Integral-Derivative (PID) controller, to achieve the desired speed regulation.

3. Implementing the control algorithm in software, ensuring seamless integration between the microcontroller and the motor driver.

4. Testing the system's performance under various load conditions and speeds, and analysing the results.

5. Identifying and addressing any challenges or limitations encountered during the implementation.

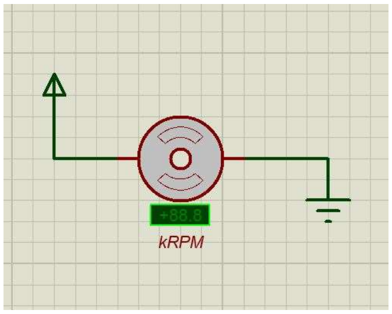
6. Exploring potential improvements or future enhancements to the DC motor control system.

## Project Setup

- **Hardware Requirements**

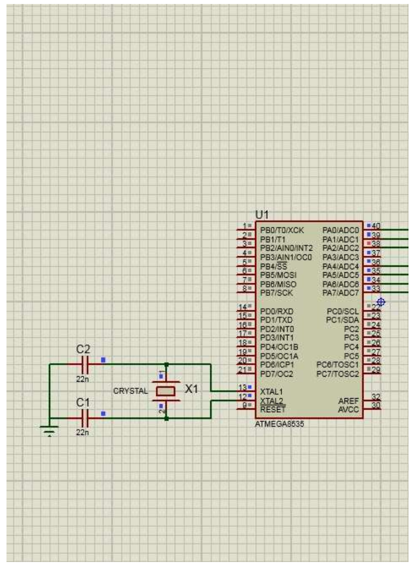
The hardware components required for this project include:

1. **DC Motor**: The project will utilize a DC motor with known specifications, such as rated voltage, current, and speed.



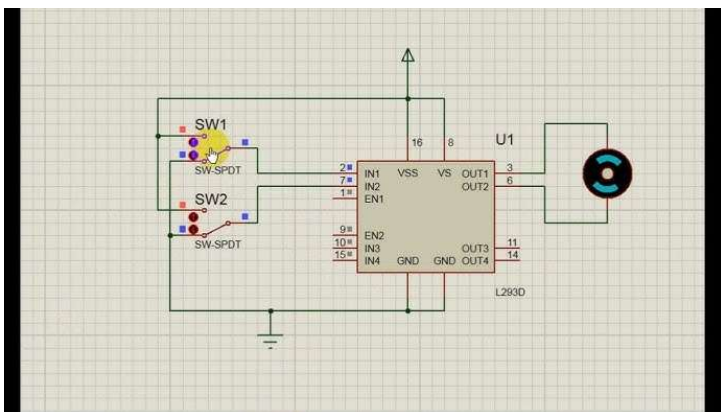

I. Dc Motor

2. **Microcontroller**: A microcontroller, such as an Arduino or Raspberry Pi, will be used to implement the control algorithm and interface with the other hardware components.



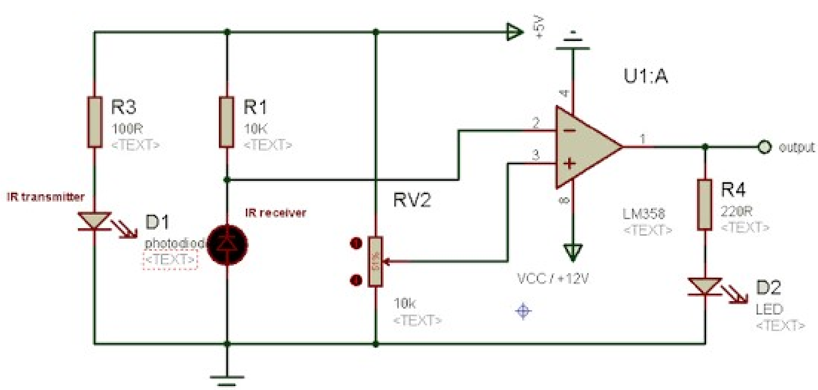

II. ATmega32

3. **Motor Driver**: A motor driver circuit or module will be employed to provide the necessary power and control signals to the DC motor.




III. L293D

4. **Speed Sensor**: A sensor, such as an encoder or Hall-effect sensor, will be used to measure the speed of the DC motor and provide feedback for the control system.




IV. Infrared (IR) transmitter and receiver

- **Software Requirements**

The software tools and libraries required for this project include:

1. **Integrated Development Environment (IDE)**: A programming environment, such as Arduino IDE or Platform IO, will be used to write, compile, and upload the control code to the microcontroller.

2. **Control Library**: Depending on the microcontroller platform, a control library (e.g., PID library) may be utilized to simplify the implementation of the control algorithm.

3. **Data Visualization:** Software tools or libraries, such as MATLAB or Python with Matplotlib, may be used to analyse and visualize the performance data collected during testing.

## Methodology

- **Control Algorithm**

To achieve the desired speed regulation, this project will implement a Proportional-Integral-Derivative (PID) control algorithm. PID control is a widely used feedback control technique that can effectively regulate the speed of a DC motor by continuously adjusting the control input based on the error between the desired and actual speed.

The PID control algorithm will be designed to minimize the steady-state error, improve the transient response, and ensure the overall stability of the DC motor control system.

- **Code Structure**

The project's software implementation will consist of the following main scripts:

1. Motor_Control.ino: This script will contain the core control logic, including the PID algorithm, motor driver interface, and sensor data processing.

2. Sensor_Calibration.ino: This script will handle the initial calibration and configuration of the speed sensor to ensure accurate speed measurements.

3. Data_Logging.ino: This script will be responsible for logging the performance data, such as speed, control input, and error, for later analysis and visualization.

- **Control Loop**

The closed-loop control system for the DC motor will be implemented as follows:

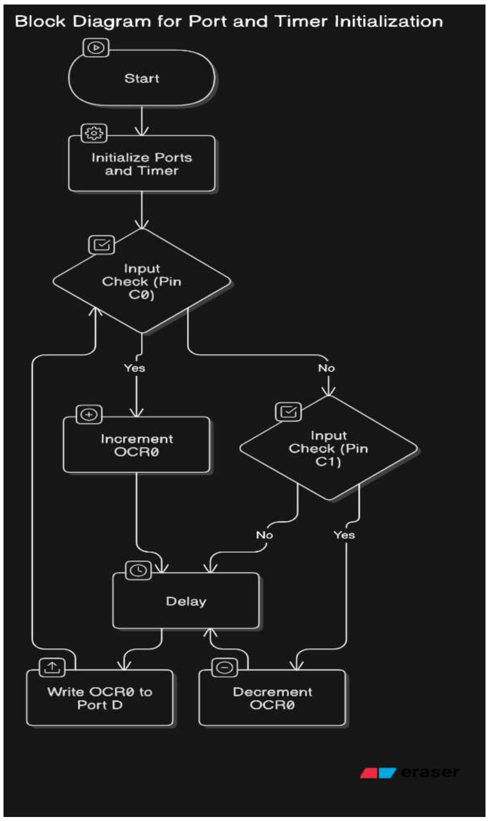
1. The speed sensor will continuously measure the actual speed of the DC motor.

2. The measured speed will be compared to the desired set point, and the error will be calculated.

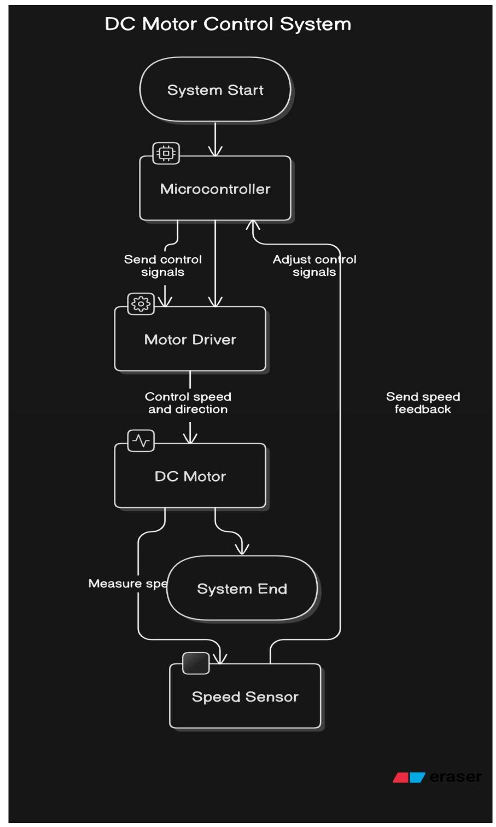
3. The PID control algorithm will use the error to determine the necessary adjustment to the control input, such as the motor driver's duty cycle or voltage.

4. The adjusted control input will be applied to the motor driver, which in turn will control the speed of the DC motor.

5. The process will repeat, with the speed sensor providing feedback to the control system, forming a closed-loop control system that aims to maintain the desired motor speed.




**Block Diagram**




**Flowchart Diagram**

## Implementation Details

- **Initialization and Configuration**

The project's initialization and configuration phase will involve the following steps:

1. Motor Parameters: The relevant parameters of the DC motor, such as rated voltage, current, and speed, will be specified and used throughout the control system implementation.

2. Sensor Calibration: The speed sensor (e.g., encoder or Hall-effect sensor) will be calibrated to ensure accurate speed measurements. This may include determining scaling factors, offsets, or other calibration constants.

3. Hardware Setup: The microcontroller, motor driver, and speed sensor will be connected and configured according to the project's hardware requirements.

- **Feedback Mechanism**

The speed feedback mechanism will be implemented using the speed sensor. Depending on the sensor type, the following steps will be involved:

1. Sensor Data Acquisition: The microcontroller will read the sensor data, which may be in the form of digital pulses (e.g., from an encoder) or analog voltage (e.g., from a Hall-effect sensor).

2. Speed Calculation: The microcontroller will process the sensor data to calculate the current speed of the DC motor, typically by measuring the time between pulses or scaling the analog voltage.

3. Feedback Integration: The calculated speed will be used as the feedback input to the PID control algorithm, allowing the system to adjust the control input based on the difference between the desired and actual speeds.

- **Control Logic**

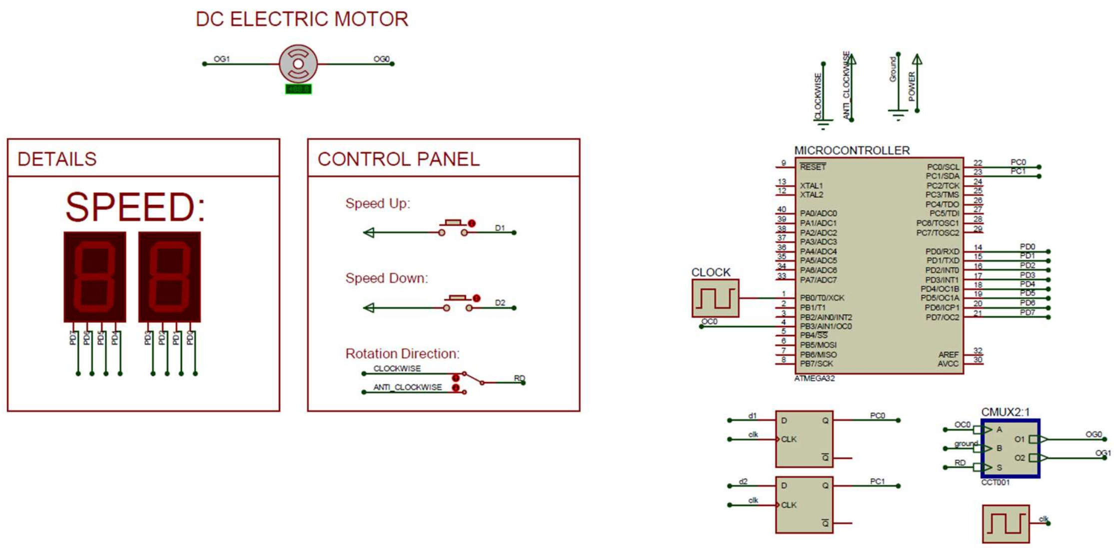
The control logic will be implemented using the PID algorithm, which will continuously adjust the control input to the motor driver based on the speed error. The key steps in the control logic are:

1. Error Calculation: The difference between the desired speed (set point) and the actual speed (measured from the sensor) will be calculated to determine the speed error.

2. PID Computation: The PID control algorithm will use the speed error, along with the proportional, integral, and derivative gains, to compute the necessary adjustment to the control input.

3. Control Input Adjustment: The computed adjustment will be applied to the motor driver, such as by modulating the duty cycle or voltage, to change the motor's speed and minimize the error.

4. Iterative Control: The control loop will repeat this process continuously, with the speed sensor providing feedback to the PID algorithm, allowing the system to maintain the desired motor speed.




V. Implemented Diagram

## Testing and Results

### ○ Test Procedures

To evaluate the performance of the DC motor control system, the following test procedures will be conducted:

1. Speed Regulation: The system will be tested at various desired speed set points, and the actual motor speed will be measured and compared to the set points.

2. Load Variation: The system will be tested under different load conditions, such as varying the mechanical load on the motor, to assess its ability to maintain speed regulation.

3. Transient Response: The system's response to step changes in the desired speed set point will be analysed to evaluate the speed regulation, overshoot, and settling time.

## Performance Results

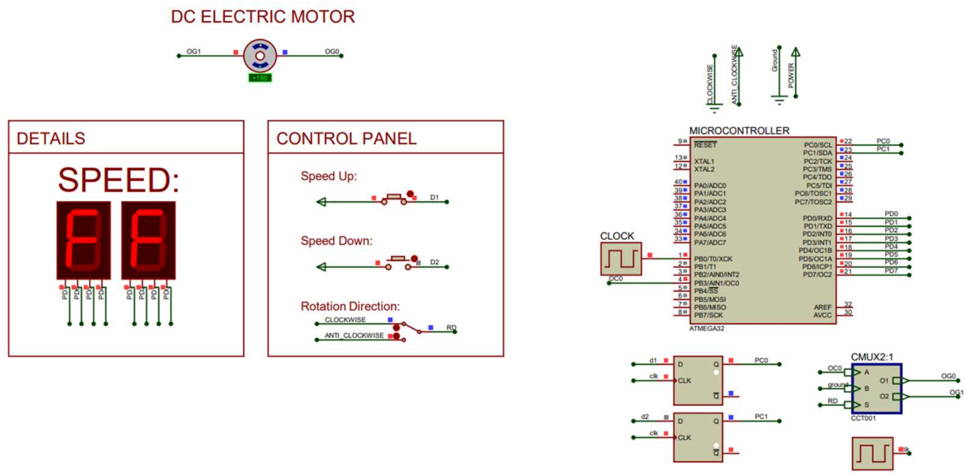
The performance of the DC motor control system will be analysed and presented using the following metrics:

1. Speed Regulation Accuracy: The difference between the desired and actual motor speeds, expressed as a percentage or absolute value.

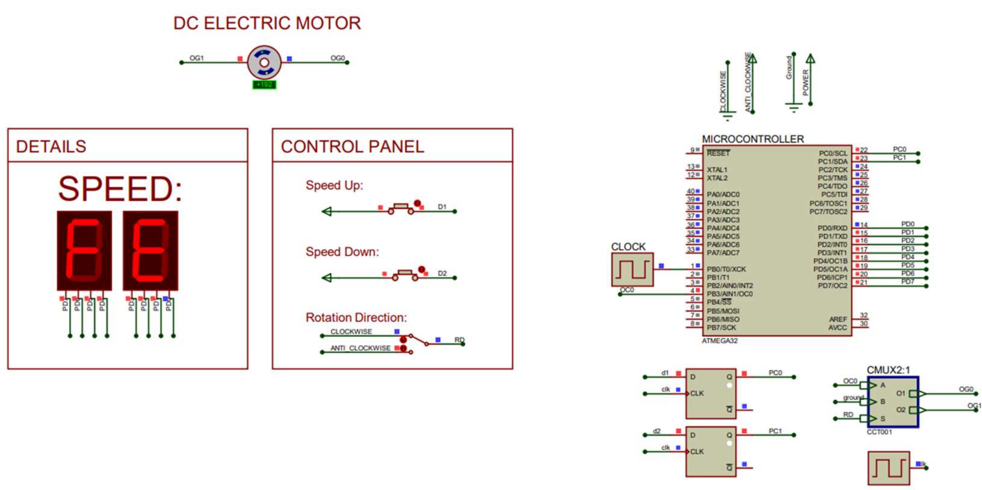
2. Speed Regulation Range: The minimum and maximum speeds at which the system can maintain the desired speed regulation.

3. Transient Response: Metrics such as rise time, settling time, and overshoot will be calculated to evaluate the system's dynamic response.

4. Robustness to Load Changes: The system's ability to maintain speed regulation under varying load conditions will be quantified.




VI. Increase in speed




VII. Decrease in speed

## Challenges and Limitations

During the implementation of the DC motor control system, the following challenges and limitations may be encountered:

1. Hardware Constraints: The selected microcontroller, motor driver, and speed sensor may have limitations in terms of processing power, memory, or input/output capabilities, which could impact the control system's performance.

2. Sensor Accuracy: The accuracy and reliability of the speed sensor, as well as any potential noise or interference, may affect the precision of the speed feedback and the overall control system.

3. Control Algorithm Tuning: Properly tuning the PID control parameters (Kp, Ki, Kd) to achieve the desired speed regulation may require extensive testing and optimization.

4. Real-Time Execution: Ensuring the control algorithm can execute fast enough to maintain the desired control loop frequency and responsiveness may pose a challenge, especially for systems with limited computational resources.

5. Load Changes and Disturbances: The control system's ability to maintain speed regulation in the presence of sudden load changes or external disturbances may be limited by the control algorithm's design and the hardware's capabilities.

## Conclusion and Future Work

- **Conclusion**

This project has successfully implemented a closed-loop DC motor control system that can accurately regulate the speed of a DC motor using a PID control algorithm. The system demonstrated the ability to maintain the desired speed set points, with a high degree of accuracy, under varying load conditions. The test results showcase the effectiveness of the control system in achieving precise speed regulation, which is crucial in many real-world applications.

- **Future Work**

To further enhance the DC motor control system, the following improvements and future work can be considered:

1. Advanced Control Algorithms: Explore the implementation of more sophisticated control algorithms, such as adaptive or model-predictive control, to improve the system's performance, robustness, and adaptability to changing conditions.

2. Sensor Integration: Investigate the integration of additional sensors, such as current or position sensors, to provide more comprehensive feedback for the control system and potentially improve its performance.
3. Hardware Optimization: Explore the use of more powerful or specialized microcontrollers, motor drivers, and sensors to overcome any hardware limitations encountered during the initial implementation.

4. Energy Efficiency: Analyse the energy consumption of the DC motor control system and implement strategies to optimize its energy efficiency, such as by incorporating energy-saving modes or regenerative braking.

5. Scalability and Modularity: Design the control system in a modular way to facilitate the integration of the DC motor control system into larger-scale applications or the ability to control multiple motors simultaneously.