# Report

**Physics of the Experiment:**

- When a DC Motor is powered up, it immediately starts rotating, it continuously rotates, and you can't control its exact position.
- Drive the motor by connecting the driver pins to the microcontroller and suitable power supply.
- Encoder helps us control rotatory aspects of DC Motor.
- Encoder works on the principle of change in magnetic field around magnet inside the motor shaft.
- The encoder outputs triggers periodically as the motor rotates. Like, when the magnet spins clockwise, output A will trigger first.
- To read from the encoder, we connect encoder pins to digital pins of microcontrollers with interrupts. Set an interrupt that triggers each time output A rises.
- Construct a control loop feedback system after defining a target position, error and control signal.
- Apply PID Control Algorithm to compute the control signal. Send the control signal as power input to drive the motor.

**Formulae Used:**

Formula

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_p \frac{de}{dt}$$

$u(t)$ = PID control variable

$K_p$ = proportional gain

$e(t)$ = error value

$K_i$ = integral gain

$de$ = change in error value

$dt$ = change in time

## ThingSpeak And OM2M:

Mqtt subscribe was used for the motor to receive updates and start the experiment whenever a user presses the submit button and publishes the input values to thingspeak

Http post protocol was used to send motor output angle values to thingspeak and onem2m. Since the update interval for thingspeak is 15 seconds, data was gathered in every 100 ms, stored in a json buffer and sent to thingspeak using bulk update after 15 second.

Arduino Codes for running the hardware are in the Arduino folder.

## Dashboard:

The following is github link to the dashboard:
https://github.com/chanda1423/esw-project-pid.git

The dashboard has a

- Sign-in page
- Register page
- Theory page which explains the theory of the experiment for any user
- Experiment page which has
    - A session timer which prevents a user from using the service for a longer time.
    - Buttons and sliders for taking Kp,Ki,Kd values which are then sent to thingspeak and then to hardware to get the motor working.
    - Graph which is copied from thingspeak.
    - Live stream from esp32 cam.

Dashboard was implemented mostly using HTML,CSS in front-end and MongoDb in back-end using Node Js.