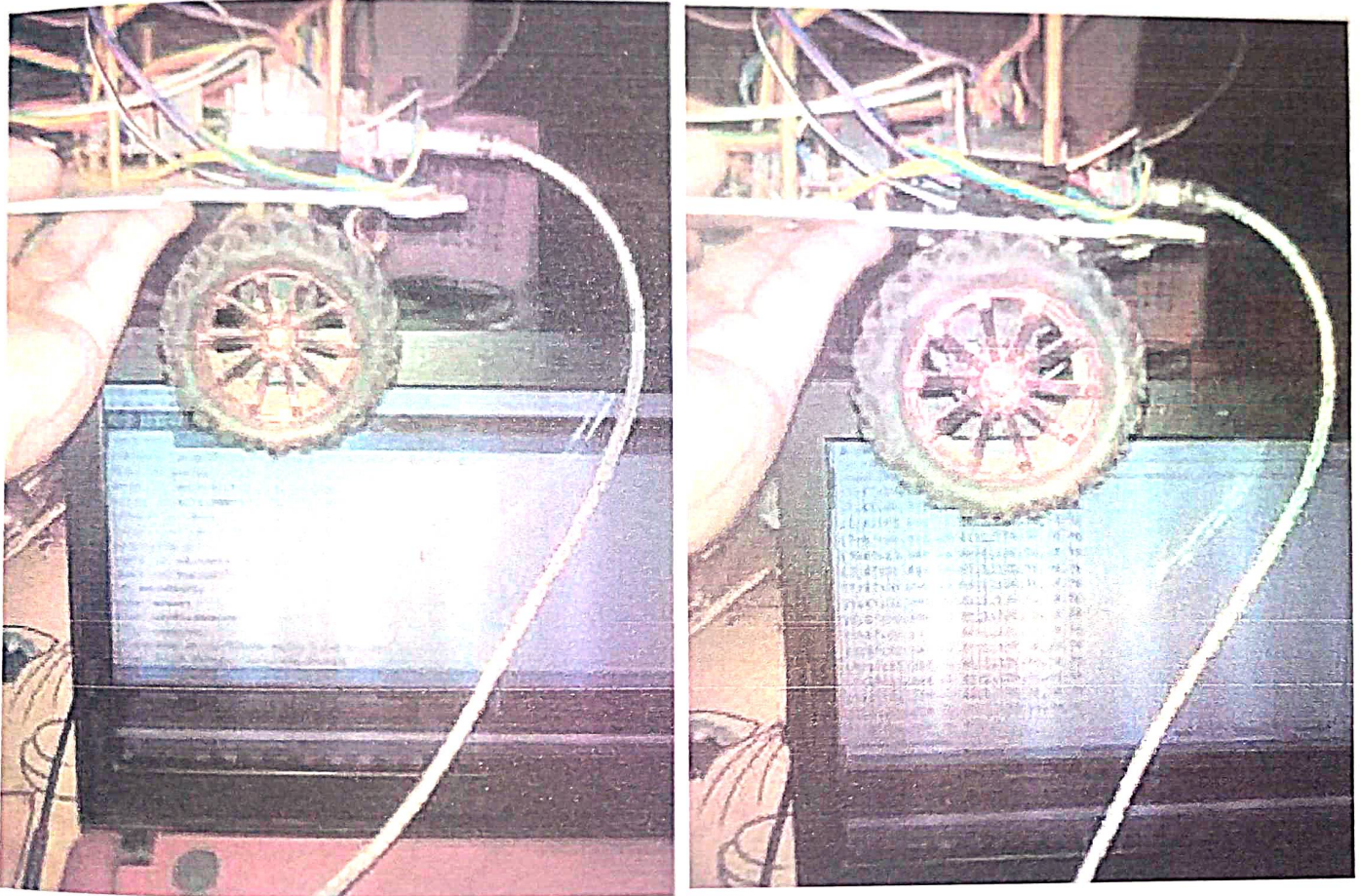




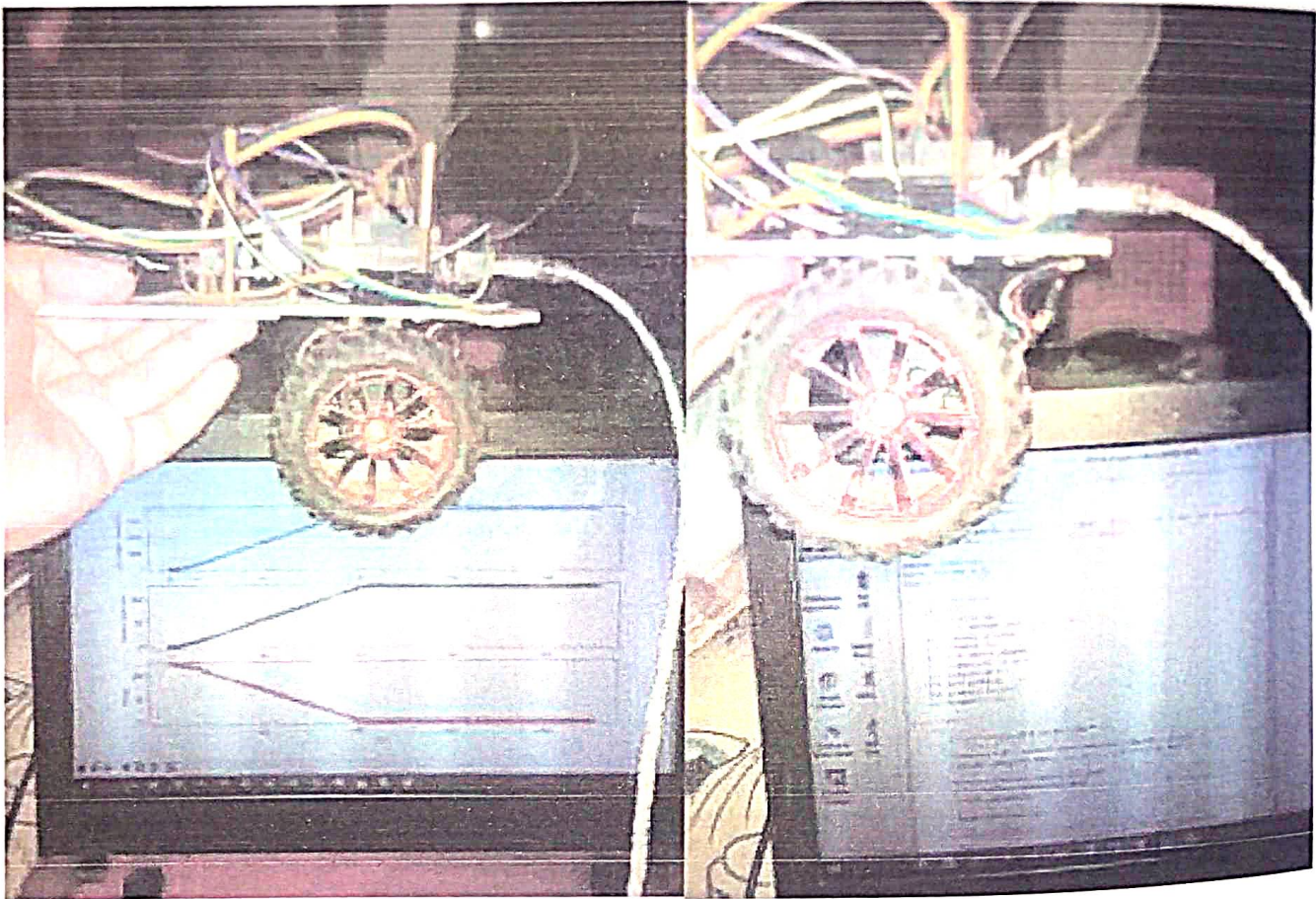
**LAMPIRAN**

## LAMPIRAN

Lampiran A. Pengujian dengan mengambil data di serial monitor Arduino IDE



Lampiran B. Pengujian dengan mengambil data grafik real-time di Phyton



## Lampiran C. Koding Arduino

*//Adjust Target Degree from here which degree do you want*

```
float targetDeg = 360;
```

```
const int motorDirPin = 4; // Input 1
```

```
const int motorPWMPin = 5; // Input 2
```

```
//const int EnablePin = 6; // Enable
```

```
//const int LED = 12;
```

```
// encoder pin
```

```
const int encoderPinA = 2;
```

```
const int encoderPinB = 3;
```

```
int encoderPos = 0;
```

```
float error;
```

```
float previouserror;
```

```
float cumerror;
```

```
// encoder value change motor turn angles
```

```
const float ratio = 360./188.611/48.;
```

```
// 360. -> 1 turn
```

```
// 188.611 -> Gear Ratio
```

```
// 48. -> Encoder: Countable Events Per Revolution (Motor Shaft)
```

```
// PID Tuning
```

```
float Kp = 30;
```

```
float Ki = 0.0001;
```

```
float kd = 40;
```

```
void doEncoderA()
```

```
{
```

```

    encoderPos += (digitalRead(encoderPinA) == digitalRead(encoderPinB)) ? 1 : -1;
}

void doEncoderB()
{
    encoderPos += (digitalRead(encoderPinA) == digitalRead(encoderPinB)) ? -1 : 1;
}

void doMotor(bool dir, int vel)
{
    digitalWrite(motorDirPin, dir);
    //digitalWrite(LED, dir);
    analogWrite(motorPWMPin, dir ? (255 - vel) : vel);
}

void setup()
{
    Serial.begin(9600);
    pinMode(encoderPinA, INPUT_PULLUP);
    attachInterrupt(0, doEncoderA, CHANGE);

    pinMode(encoderPinB, INPUT_PULLUP);
    attachInterrupt(1, doEncoderB, CHANGE);

    //pinMode(LED, OUTPUT);
    pinMode(motorDirPin, OUTPUT);
    //pinMode(EnablePin, OUTPUT);

```

```

cumerror=0;
previouserror=0;
}

void loop()
{
cumerror=cumerror+error;
previouserror=error;
float motorDeg = float(encoderPos)*ratio;
error = targetDeg - motorDeg;
float control = Kp*error+Ki*cumerror+kd*(error-previouserror);
//analogWrite(EnablePin, 255);
doMotor((control>=0)?HIGH:LOW, min(abs(control), 255));
Serial.print(encoderPos);
Serial.print(",");
Serial.print(motorDeg);
Serial.print(",");
Serial.println(error);

//if( abs(error) < 0.02 ){
// digitalWrite(LED,LOW);
// while(100);
// }
}

```

## Lampiran D. Koding Phyton

```
import serial

import matplotlib.pyplot as plt

from drawnow import drawnow

arduino_data = serial.Serial('COM6', 9600) # Adjust the 'COM6' to your Arduino's
serial port

plt.ion() # Enable interactive mode for matplotlib

encoder_positions = []
motor_degrees = []
errors = []

def plot_values():
    plt.clf()
    plt.subplot(311)
    plt.plot(encoder_positions, '-bx')
    plt.ylabel('Encoder Positions')
    plt.subplot(312)
    plt.plot(motor_degrees, '-gx')
    plt.ylabel('Motor Degrees')
    plt.subplot(313)
    plt.plot(errors, '-rx')
    plt.ylabel('Errors')
    plt.xlabel('Time')
```

```
while True:
    try:
        while arduino_data.in_waiting == 0:
            pass
        arduino_line = arduino_data.readline().decode().strip()
        enc_pos, mot_deg, err = map(float, arduino_line.split(','))

        encoder_positions.append(enc_pos)
        motor_degrees.append(mot_deg)
        errors.append(err)

        drawnow(plot_values)

    except KeyboardInterrupt:
        break
    except Exception as e:
        print(f"Error: {e}")
```