

```

#include <AutoPID.h>

/* Pin and variable definitions */
#define SET_IN A0
#define POS_IN A1
#define OUT_POS 3
#define OUT_NEG 9

/*maximum and minimum output for the PID object, time interval of operation*/
#define OUT_MAX 1023
#define OUT_MIN -1023
#define TIME_STEP 20

/* These are the proportional, integral and derivative constants for the PID loop */
#define Kp 4
#define Ki 0.7
#define Kd 0.3

/* global storage variables, passed by reference to PID object */
double setpoint, servopos, pwmout, pwmval;

/* PID object, makes riemann sum calculations of integral and derivative responses for
 * the discrete-time dataset provided to it */
AutoPID servoPID(&servopos, &setpoint, &pwmout, OUT_MIN, OUT_MAX, Kp, Ki, Kd);

/*converts a voltage ranging from 0-5 V into a PWM duty cycle proportional to the input
 * magnitude */
double getPWM(double PWMval) {
    double result = (PWMval/1023)*255;
    return result;
}

void setup() {
    pinMode(OUT_POS, OUTPUT);
    pinMode(OUT_NEG, OUTPUT);
    servoPID.setTimeStep(TIME_STEP); /*run PID calculations in TIME_STEP ms*/
    Serial.begin(9600);
}

void loop() {
    setpoint = analogRead(SET_IN); /*read setpoint potentiometer voltage*/
    servopos = analogRead(POS_IN); /*read motor position potentiometer voltage*/
    servoPID.run(); /*iterates the operation of the PID object. When the time defined by

```

```

        TIME_STEP is reached, performs calculations on collected data and
        changes the output variable accordingly*/
pwmval = abs(getPWM(pwmout));
Serial.println(servoPID.getIntegral());
/*Serial.println(servopos);*/
if (pwmout > 0){
    analogWrite(OUT_NEG, pwmval); /*writes a PWM signal to a digital pin*/
    digitalWrite(OUT_POS, LOW);
    /*if PID output is negative, turn the servo in the reverse direction according to
    * voltage across it*/
}
else if (pwmout < 0){
    analogWrite(OUT_POS, pwmval);
    digitalWrite(OUT_NEG, LOW);
    /*if PID output is positive, turn the servo in the forward direction according to
    * voltage across it*/
}
else {
    digitalWrite(OUT_POS, LOW);
    digitalWrite(OUT_NEG, LOW);
    /*do not turn the servo if PID output is 0*/
}
}

```