Міністерство освіти і науки України

**Прикарпатський національний університет**

**імені В.Стефаника**

*Факультет математики та інформатики*

*Кафедра інформаційних технологій*

*Програмування вбудованих систем*

Лабораторна робота № 3

Тема: «Тема: «Керування сервоприводом»»

Варіант : **21**

Виконав: **Рижкін О. К.**

Група ІПЗ-23

Дата: 10 жовтня 2023р.

Викладач: Лазарович І.М.

Івано-Франківськ – 2023

**Мета:** Вивчити принципи роботи і програмування сервопривода, цифрового енкодера та ШІМ-сигналів.

**Завдання на виконання**

Використовуючи наведені теоретичні відомості, розробити систему керування сервоприводом за допомогою енкодера, яка працює по наступному алгоритму. Початково сервопривід встановлюється в середнє положення. Читання енкодера відбувається з періодом 100см, при виявленні повертанні енкодера сервопривід теж здійснює поворот на 2 градуси у відповідну сторону. При досягненні крайніх положень (0 або 180 град), і подальшому обертанні енкодера в ту саму сторону з сервоприводом нічого не робити, але поступово (з кроком 5%) збільшувати яскравість вбудованого в Ардуіно світлодіода (під 13). При досягненні максимальної яскравості і обертанні енкодера в ту саму сторону – не робити нічого. При обертанні енкодера в іншу сторону, яскравість світлодіода зменшити, коли яскравість 0, тоді почати обертання сервопривода. Тобто яскравість світлодіода стає умовним «продовженням» ходу сервопривода. З метою попередньої підготовки до виконання лабораторної роботи студент повинен самостійно спроектувати схему з’єднань в Fritzing, а на початку лабораторного заняття продемонструвати її викладачеві. В звіті навести: тему, сформульоване завдання, програмний код, схему з’єднань з редактору Fritzing, посилання на youtoube з відео, де чітко видно роботу системи, висновки. Необхідні комплектуючі для виконання роботи: - Arduino UNO – 1 шт.; - Макетна плата – 1 шт.; - Цифровий інкрементний енкодер – 1 шт.; - Сервопривід SG – 90, або аналогічний – 1 шт.; - З’єднувальні провідники ~ 6 шт.

**Code**

#include <Servo.h>

#include <Encoder.h>

int pos = 20;

Servo myServo;

const int LED\_PIN = 13;

const float step = 12.75;

float ledBrightness = 0;

boolean directionLed;

long encoderPos = -100;

Encoder enc(2, 4);

unsigned long previousMillis = 0;

boolean isEncoderAtLimit = false;

const int pin = 5;

unsigned long tCurr, tLast = 0, t\_pwm = 30;

float kP = 0;

void setup() {

myServo.attach(3, 500, 2500);

pinMode(LED\_PIN, OUTPUT);

pinMode(pin, OUTPUT);

Serial.begin(9600);

myServo.write(pos);

}

void generatePWM(float kPWM) {

if (isEncoderAtLimit) {

if (tCurr - tLast >= t\_pwm) {

PORTB |= (1 << PORTB5); // Turn on the LED (assuming LED\_PIN is 13)

tLast = tCurr;

} else {

if (tCurr - tLast >= kPWM \* t\_pwm) {

PORTB &= ~(1 << PORTB5); // Turn off the LED

}

}

} else {

PORTB &= ~(1 << PORTB5); // Turn off the LED

}

}

void rotate(boolean isRight) {

directionLed = isRight;

if ((pos < 180 && pos > 0) || (ledBrightness == 0 && directionLed)) {

pos = isRight ? pos + 2 : pos - 2;

myServo.write(pos);

} else {

if (ledBrightness < 255 && !directionLed) {

kP += 0.05;

t\_pwm += 2.5;

ledBrightness += step;

} else if (ledBrightness > 0 && directionLed) {

kP -= 0.05;

t\_pwm -= 2.5;

ledBrightness -= step;

}

PORTB |= (1 << PORTB5); // Turn on the LED

}

}

void loop() {

tCurr = millis();

unsigned long currentMillis = millis();

if (currentMillis - previousMillis >= 100) {

previousMillis = currentMillis;

long newPos;

newPos = enc.read();

if (newPos != encoderPos) {

rotate(newPos < encoderPos);

encoderPos = newPos;

if (pos == 0 || pos == 180) {

isEncoderAtLimit = true;

} else {

isEncoderAtLimit = false;

}

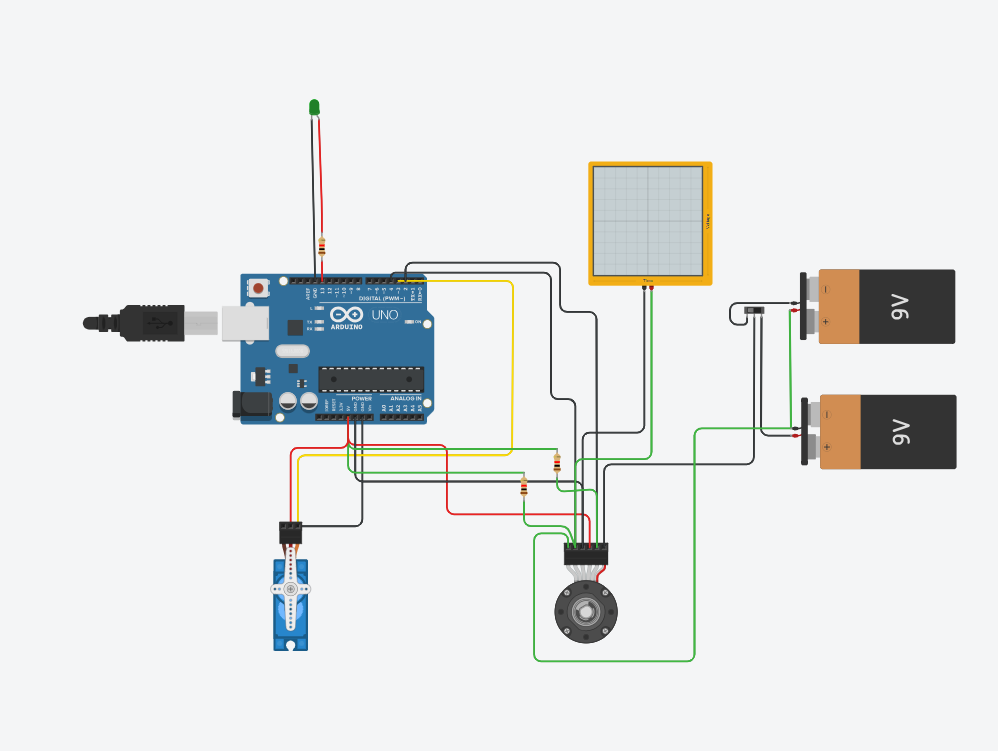
}

}

generatePWM(kP);

}

**Screenshot of the scheme**



https://www.tinkercad.com/things/laZu0GSGup5-lab3/editel