**Зміст**

1.Введення...................................................................................................................3

2. Ознайомлення з місцем практики.........................................................................6

3.Прохождение практики...........................................................................................7

4. Висновки................................................................................................................13

5. Література………………………………………………………………………..14

6. Додаток А……………………………………….……………………………..…15

7. Додаток В………………………………………………………………...………17

**ВВЕДЕННЯ**

Переддипломну практику була проведена на Кафедрі Автоматизації Електромеханічних систем та Електороприводу.

Кафедра готує бакалаврів, спеціалістів та магістрів денної, заочної та прискореної форм навчання за спеціальністю „ЕЛЕКТРОМЕХАНІЧНІ СИСТЕМИ АВТОМАТИЗАЦІЇ ТА ЕЛЕКТРОПРИВОД” - однією з найбільш конкурентноспроможних спеціальностей технічних вузів України за напрямом „ЕЛЕКТРОМЕХАНІКА”. Підготовка фахівців здійснюється на основі комп'ютерних технологій в галузі кібернетичних та звичайних електромеханічних систем автоматичного керування робочих машин, установок, процесів (загалом об'єктів) та автоматизованого електропривода. Кафедра має найвищий IV рівень акредитації та здійснює також підготовку аспірантів за двома спеціальностями.

Випускники спеціальності готуються для проектування, дослідження та експлуатації електромеханічних систем автоматизації об'єктів різних галузей промисловості, транспорту, сільського господарства та інших сфер діяльності в області електромеханічних систем автоматизації загальнопромислових механізмів та електромехатронних систем автоматизації (системи на основі електронно-механічних пристроїв з комп'ютерним керуванням).

Близько 2/3 виробленої в світі електричної енергії перетворюється за допомогою електромеханічних систем автоматизації та електроприводів в механічну енергію руху самих різних об'єктів в найширших сферах діяльності людини. Ускладнення завдань керування їх рухом потребує застосування новітніх теоретичних методів (в тому числі кібернетики) і комп'ютерно-інтелектуальних електромеханічних систем автоматизації.

Найширша і різноманітна сфера використання сучасних електромеханічних систем автоматизації та електроприводів визначає необхідність ґрунтовної базової підготовки фахівців широкого профілю в галузі теорії автоматичного керування та теорії електроприводу, автоматизації технологічних процесів та установок, обчислювальної техніки, електроніки, технічних засобів автоматизації, комп'ютерного моделювання та проектування.

Випускники спеціальності одержують також необхідні знання з технологічних особливостей загальнопромислових об'єктів, електроприводів роботів та гнучких виробництв, електропостачання, організації виробництва, економіки, маркетингу та менеджменту.

Випускники кафедри завдяки широкому профілю ґрунтовної підготовки мають великий попит як у галузі виробництва, так і в проектно-конструкторської та наукової сферах.

У складі кафедри 10 учбових лабораторій, оснащених технічними засобами, що базуються на сучасних комп’ютерних технологіях.  
Всі викладачі кафедри мають вчені ступені: 4 доктори технічних наук та 14 кандидатів технічних наук.

**Матеріально-технічна база**

Кафедра АЕМС-ЕП розміщується на площі 1063,4 кв.м. (корпус №20) Площа навчальних приміщень дорівнює 755,9 кв.м. Кількість навчальних лабораторій – 13, загальною площею 706,3 кв.м. У кожній лабораторії мають можливість розміститися до 25 студентів. Дві лабораторії використовуються як комп’ютерні класи, одна – як аудиторія кафедри. Крім того кафедра має кабінет курсового та дипломного проектування, загальною площею 49,6 кв.м. на 10 робочих місць.

Частина лабораторій, важке обладнання яких вимагало спеціальних фундаментів, розміщена в цокольному поверсі корпуса №20. До них відносяться лабораторії:

1. теорії електропривода (№006-20);
2. автоматизованого електропривода загально-промислових механізмів та електропостачання №016-20);
3. автоматизації технологічних процесів (№015-20);
4. автоматизованого електропривода для не електротехнічних спеціальностей (№017-20);
5. систем автоматичного керування електроприводами (№007-20).   
   Решта лабораторій розміщена на 4-му поверсі корпусу №20:
6. лабораторія оптимізації режимів електроприводів (№406-20);
7. теорії авторегулювання та керування (№407-20);
8. електроприводів робото-технічних та електромехатронних систем (408-20);
9. навчальний центр «Сучасні технології в автоматицації» (спільно з компанією ЕАТОN) (№412-20);
10. навчальний центр «Сучасні технології в електромеханіці» (cпільно з компанією АББ Україна) (№413-20);
11. автоматизованих електроприводів в металообробці та машинобудуванні (№413-20);
12. комп’ютерний клас (№414-20);
13. електричних апаратів (№415-20).

Крім того, кафедра має адміністративні та навчально-допоміжні приміщення загальною площею 307,5 кв.м. Таким чином, кафедра забезпечена навчальними приміщеннями для виконання навчальних програм зі студентами та приміщеннями для співробітників.

Балансова вартість лабораторного обладнання з розрахунку на одного студента денної форми навчання складає 886 тис. грн.. / 197 студ. = 4,5 тис. грн.

Кафедра добре забезпечена обчислювальною технікою. В розпорядженні студентів є 50 персональних комп’ютерів. Кількість робочих комп’ютерних місць на 100 студентів дорівнює 25,4.

Кафедра налагоджує контакти з провідними фірмами, які спеціалізуються на випуску необхідного електротехнічного обладнання. Крім того, на стадії підписання знаходиться договір з фірмами Сіменс та СВ Альтера на поставку електротехнічного обладнання для комплектації лабораторій.

Крім того, в Інституті електродинаміки Академії Наук України створено факультет фахової підготовки, на якому читаються лекції студентам та призначуються консультанти по магістерським роботам

**ОЗНАЙОМЛЕННЯ З МІСЦЕМ ПРАКТИКИ**

Перед початком практики був проведений інструктаж з техніки безпеки, проведена екскурсія по зонах робот, ознайомлення з обладнанням кафедри і приладами які використовуються на кафедрі. Було створене робоче місце для проходження практики, обладнане інструментами та іншим для проходження практики.

**ПРОХОДЖЕННЯ ПРАКТИКИ**

На переддипломній практиці була проведена робота по створенню стенду трьох-ланкового маніпулятора зі змінними насадками робочого органу. Цей маніпулятор припустимо буде використовувати для проведення лабораторних робіт з дисципліни «Теорія мехатронних систем».

Для створення маніпулятора була обрана схема трьох ланкового маніпулятора який складається з базової поворотної ланки до якої кріпиться послідовно дві ланки. На кінці останньої ланки розміщується робочий орган, який виконує головну роботу. Довжина ланок була обрана 150 мм., це дозволяє проводити роботу в великій зоні біля маніпулятора. Кінематична модель маніпулятора показана на Рис. 1.



Рис. 1. Кінематична модель маніпулятора

Базова ланка (1) повертається у площині x-y, ланка (2) обертатається навколо неї, ланка (3) кріпиться к ланці (2).

При керуванні роботами необхідно вирішити задачу, яким чином вивести робочий орган маніпулятора в задану точку простору, тобто необхідно знайти значення узагальнених координат, які забезпечать положення та орієнтацію робочого органу. Ця задача називається оберненою задачею кінематики.

Для приведення в рух маніпулятора була взяте рішення оберненої задачі кінематики з курсу «Теорії мехатронних систем-2» для випадку дволанкового маніпулятора, без використання базової ланки яка працює в площині (x-y).

При рішенні рівнянь оберненої задачі отримуються рівняння кутів ,  ланок маніпулятора:





Де:  це довжина ланок відповідно,

 задане положення робочого органу.

Алгоритм роботи було перевірено на комп’ютері і написана програма для перевірки двохланкового маніпулятора. Робота візуалізації представлена на Рис.1.

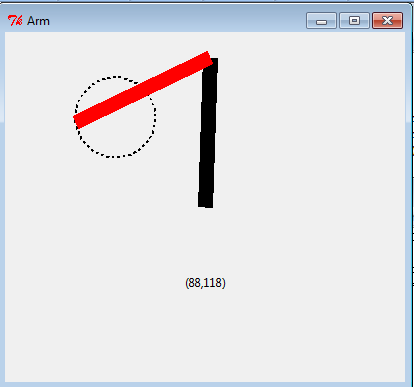


Рис.1. Програма симуляції двохланкового маніпулятора

В вікні візуалізатора будується траєкторія пунктирною лінією і дві ланки, цифрами пишуться кути які відправляються до стенду.

Для написання програми симуляції були використанні рівняння для знаходження кутів за допомогою обратної задачі кінематики, воні представленн у Додатку А функції def calk\_tet2(x0, y0), calk\_tet1(x0, y0, tet2) .

Щоб провести перевірку роботи цих алгоритмів у житті було зроблено стенд, структурна схема якого представлена на Рис.2.

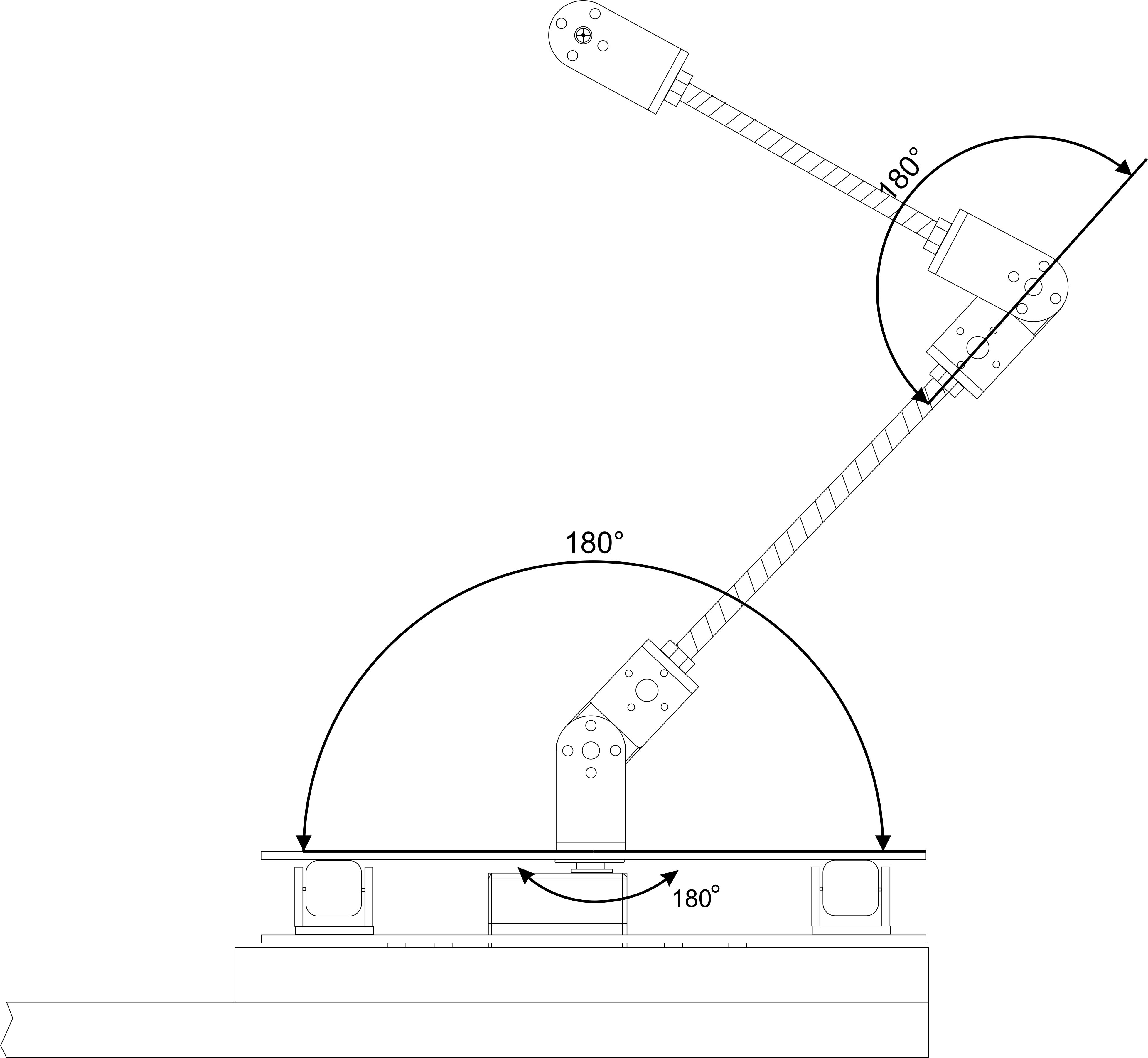


Рис.2. Структурна схема стенду

Стенд складається з бази яка розміщена внизу, в ній закріплен серводвигун якій обертає платформу яка утримується на 4йох роликах, базова анка може обертатися як і остальні ланки максимум на 180о. На поворотній платформі змонтовані решта ланок які з’єднані між собою за допомогою шпільки з різьбою.

Для приведення в рух маніпулятора потрібно мати двигуни і систему замкненого керування до них. В даній роботі використовуються серводвигуни, які вже мають замкнену систему керування по положенню. Це мотор-редуктор, здатний повертати вихідний вал строго в задане положення (на кут) і утримувати його там, всупереч опорам і збурень середовища. Вони використовуються у авіомоделюванні в основному та в аматорських роботах. У найпростіших аналогових серводвигунах кут задається тривалістю імпульсів з певною частотою, в більш кращих використовується протокол I2C (можна положення поточний дізнатися, можна і навантаження поточне дізнатися і швидкість руху). В використаних серводвигунах у лабораторному стенді використовуються прості серводвигуни керування яких виконується при зміні тривалості імпульсів. Але для того щоб дізнаватись поточне положення валу для алгоритмів керування виведен зворотній зв’язок з змінного резистора яких находиться в середині серводвигуна.

Для роботи було обрано серводвигун з такими технічними параметрами які приведені у Таблиці 1.

Таблиця 1. Параметри серводвигуна RDS3115

|  |  |
| --- | --- |
| Розміри(ВхШхГ) | 40см\*20см\*40.5см |
| Маса | 60 г. |
| Швидкість | 0.16 сек/60о |
| Момент | 15 кг/см |
| Редуктор | Метал |
| Максимальний кут повороту | 180о |

Для керування кутом повороту RDS3115 потрібно задавати і утримувати сигнал відповідної тривалості. Кут 0о відповідає тривалості імпульсу 240 мкс, кут 180о відповідає тривалість 2400о мкс. Система керування створює такі затримки і утримує їх доки не прийде нове завдання.

По структурній схеми було зроблено стенд який складається з маніпулятора, контролера керування, блока живлення на 5В с максимальним током 3А. Вигляд стенду можна побачити на Рис. 3.

Платою керування було обрано контролер Arduino. Ця плата дозволяє в короткі строки створювати на основі мікроконтролера ATmega328 схеми автоматики за допомогою фреймворку Arduino і пакету Arduino IDE. Плата використана в проекті показана на Рис.4

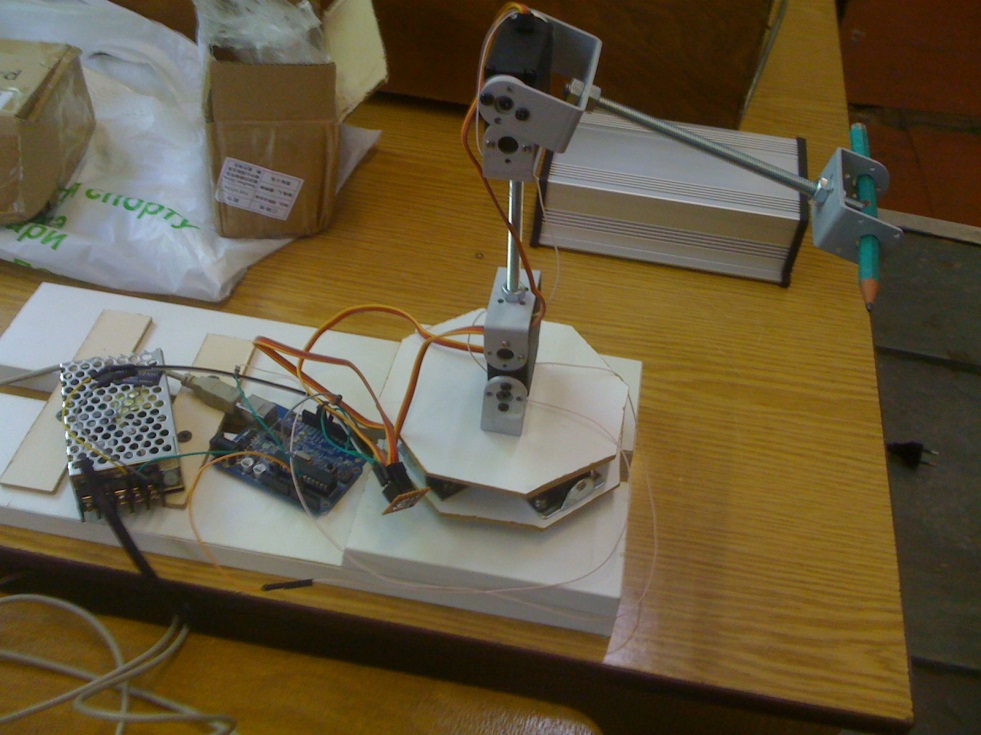


Рис.3. Маніпулятор на платформі з робочим органом

Була проведена перевірка роботи при різних заданих траєкторіях маніпулятора без використання зворотних зв’язків. Отримані результати відрізнялись від заданих. Ця різниця утворюється в наслідок того, що серводвигуни під навантаженням відпрацьовують кут з різною швидкодією. З використанням зворотних зв’язків програма керування знімала значення кута з внутрішнього резистора і чекала поки кут не дойде до заданого, після чого давалось наступне значення. Програма представлена в Додатку В.

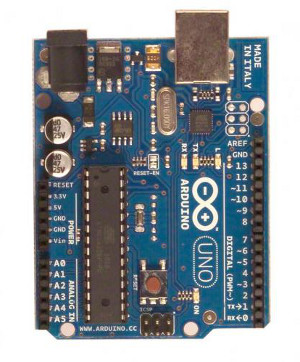


Рис. 4. Плата Arduino для протипування

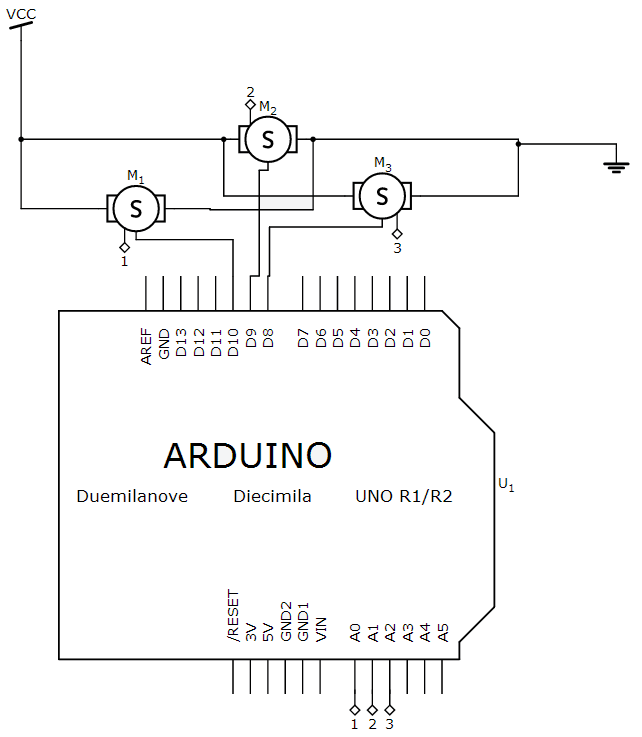


Рис. 5. Схема підключень

Як можна бачити з Рис. 5 можливо швидко підключати зовнішні пристрої і бистро писати і налагоджувати програмний код який наведено в Додатку В.

**ВИСНОВКИ**

Таким чином було проведено ознайомлення та отримано досвід в розробки робототехніки з використанням різноманітних сучасних програмних та апаратних комплексів, розробка свого маніпулятора і написання програм для керування ним з використанням теорії з курсу «Теорія мехатронних систем-2».

При тестуванні стенду с використанням зворотних зв’язкі результати стали кращі чим без них, але не ідеальні. Такі результати пов’язані з тим, що дискретність серводвигунів складає 1о, це створює при довжині ланок 15 см, неточність, якщо наприклад маніпулятор стоїть вертикально то дискретність положення робочого органу при зміні кута ±1о ставить ~0.5 см. Також треба враховувати люфти у редукторах серводвигунів. В використаних серводвигунах положення визначається за допомогою резистора змінного тому при пульсації напруги яку створює імпульсний блок живлення та зміні температури навколишнього середовища напруга знята з реостата може не відповідати номінальній для кута в якому встановлений серводвигун.

Виконана робота дає заставу до написання бакалаврської роботи та подальшої модернізації тестового стенду до лабораторного стенду для проходження практики студентів по курсу «Теорія мехатронних систем-2».

**ЛІТЕРАТУРА**

1. Методичні вказівки до курсу «Теорія мехатронних систем-2», Київ, 2013.
2. <http://www.aliexpress.com/snapshot/6021679416.html> “Original factory RDS3115 Metal gear digital servo Robot servo arduino servo for Robot diy 15kg/cm”
3. <http://robocraft.ru/blog/mechanics/240.html> “Сервы / Механика / RoboCraft”

**Додаток A. Програма симуляції дволанкового маніпулятора**

……..

l1 = 150.0

l2 = 150.0

def deg(a):

return math.degrees(a)

#розрахунок типових траєкторій кола та прямокутників

def circle(x0, y0, r):

x=[x0+r\*math.cos(i\*math.pi/180) for i in range(1, 361, 4)]

y=[y0+r\*math.sin(i\*math.pi/180) for i in range(1, 361, 4)]

return (x, y)

def box(x0, y0, x1, y1):

x=[i\*10 for i in range(int(math.fabs(x0)), int(math.fabs(x1)))]

y=[y0\*10 for i in range(int(math.fabs(x0)), int(math.fabs(x1)))]

return (x, y)

…….

#створення шляху

self.x0, self.y0 = circle(-90, 90, 40)

……

#розрахунок кутів

tet1p, tet2p = getAngles(self.x0[self.count1], self.y0[self.count1])

#розрахуок кутыв для відмалювання в симуляторі

tet1 = 180-(tet1p)

tet2 = 180-((tet2p)-tet1)+180

text = '('+str(tet1p) +',' + str(tet2p) + ')'

self.canvas.create\_text((width/2, hight-100),text=text)

line = arms[0].draw(tet1)

arms[1].draw(tet2, line, 'red')

…..

def getAngles(x0, y0):

tet2 = calk\_tet2(x0, y0)

tet1 = calk\_tet1(x0, y0, tet2)

return (int(deg(tet1)), int(deg(tet2)))

#розрахунок кутів за обратною задачею кінематики

def calk\_tet2(x0, y0):

angle = math.acos((x0\*x0+y0\*y0-l1\*l1-l2\*l2)/(2\*l1\*l2))

return angle

def calk\_tet1(x0, y0, tet2):

angle = math.atan((y0\*(l1+l2\*math.cos(tet2))-x0\*l2\*math.sin(tet2))

/(y0\*l2\*math.sin( tet2 ) + x0\*(l1+l2\*math.cos( tet2 ))))

return angle

**Додаток B. Програма керування дволанкового маніпулятора**

#include "Servo.h"; //import servo library

const int numServos = 3; //how many servos do you have?

const int sPin[numServos] = {9,10,11}; //what pins do they correlate to?

const int mPin[numServos] = {3,1,2}; //what pins do they correlate to?

const int annn[5] = {100, 45, 180, 80, 50};

char point = 0;

Servo servo[numServos]; //declare the servo array

float there[numServos]; //angle moving to

int reading[20];

int i = 0, h, t;

int feedBack; //used to hold servo feedback value

boolean doneMove[numServos];

float l1 = 149;

float l2 = 147;

double angle;

const int len = 13;

const float ys = 90;

int y0min = 100;

int y0max = 130;

int x0min = 100;

int x0max = 130;

int yy = y0min;

int xx = x0min;

int pp = 0;

int ppp = 2;

int sign = 0;

void setup() {

Serial.begin(19200); // initialize serial output

Serial.println("it's on!");

for(i=0; i<numServos; i++) {

servo[i].attach(sPin[i]);

servo[i].write(45);

there[i] = 45;

A[i] = 56;

B[i] = 590;

doneMove[i] = true;

}

delay(1000);

}

int angle\_fix(int a) {

if(a > 180)

a = 180;

if(a<0)

a = 0;

return a;

}

int pok;

void loop() {

if(doneMove[0] && doneMove[2]) {

there[0] = calk\_tet2(-xx, yy);

there[2] = angle\_fix(180-get\_deg(calk\_tet1(-xx, yy, there[0])));

there[0] = angle\_fix(get\_deg(there[0]));

pok = there[0];

Serial.print(pp);

Serial.print(" ");

Serial.print(x0min);

Serial.print(" ");

Serial.print(yy);

Serial.print(" Angle tet1:");

Serial.print(there[2]);

Serial.print(" tet2:");

Serial.println(there[0]);

if(sign)

yy++;

else

yy--;

if(yy>x0max) {

sign = 0;

}

if(yy<x0min)

sign = 1;

ppp=1;

}

for(i=0; i<numServos; i++) {

if (doneMove[i] && ppp) {

if(i==(numServos-1))

ppp = 0;

if(i == 0) {

there[i] = pok;

}

doneMove[i] = false;

servo[i].write(there[i]);

} else {

h= readMove(i);

if(i == 0) {

there[i] = pok;

}

h = abs(there[i]-h);

if(h<5) {

doneMove[i] = true;

}} }}// END VOID LOOP

//Get angle real

int readMove(int n){

int out;

out = getFeedback(mPin[n]);

out = map(out, A[n], B[n], 0, 180);

return out;

}

//Get ADC value of servo

int getFeedback(int a){

int mean;

int result;

int test;

boolean done;

for (int j=0; j<20; j++){

reading[j] = analogRead(a); //get raw data from servo potentiometer

delay(whlReading);

} // sort the readings low to high in array

done = false; // clear sorting flag

while(done != true){ // simple swap sort, sorts numbers from lowest to highest

done = true;

for (int j=0; j<20; j++){

if (reading[j] > reading[j + 1]){ // sorting numbers here

test = reading[j + 1];

reading [j+1] = reading[j] ;

reading[j] = test;

done = false;

}}}

mean = 0;

for (int k=6; k<14; k++){ //discard the 6 highest and 6 lowest readings

mean += reading[k];

}

result = mean/8; //average useful readings

return(result);

}

//convert radians to degrees

int get\_deg(double a) {

return int(a \* 4068) / 71;

}

double calk\_tet2(int x, int y) {

double out;

out = acos((x\*x+y\*y-l1\*l1-l2\*l2)/(2\*l1\*l2));

return out;

}

double calk\_tet1(int x, int y, double tet2) {

float out;

out = atan((y\*(l1+l2\*cos(tet2))-x\*l2\*sin(tet2))

/(y\*l2\*sin( tet2 ) + x\*(l1+l2\*cos( tet2 ))));

return out;

}