

## ДОДАТОК А

## ЛІСТИНГ КОДУ

```
void loop() {
    pos_1 = myservo_1.read();
    pos_2 = myservo_2.read();

    while ( pos_1 < 180 ) {
        pos_1 = smoothMove( myservo_1, pos_1 + 10, 10 );
        distance = getDistance();
        printDistance( distance );
    }
    pos_1 = smoothMove( myservo_1, 0, 10 );

    if ( pos_2 < 180 ) {
        pos_2 = smoothMove( myservo_2, pos_2 + 10, 10 );
    } else {
        pos_2 = smoothMove( myservo_2, 0, 5 );
    }
}

int smoothMove( Servo servo, int to, int smooth ) {
    int pos = servo.read();
    while ( pos != to ) {
        if ( pos > to ) {
            pos -= 1;
        } else {
            pos += 1;
        }
        servo.write( pos );
        pos = servo.read();
    }
}
```

```

        delay(smooth);
    }
    return servo.read();
}

```

```

float getDistance () {
    while ( i > 1 ) {
        if ( millis() - sensTimer > 50 ) {
            i++;
            dist_3[i] = (float)sonar.ping() / 57.5;
            sensTimer = millis();
        }
    }

    dist = middle_of_3( dist_3[0], dist_3[1], dist_3[2] );
    delta = abs( dist_filtered - dist );
    if ( delta > 1 ) k = 0.7;
    else k = 0.1;
    dist_filtered = dist * k + dist_filtered * ( 1 - k );

    if ( i > 1 ) i = 0;
    return dist_filtered;
}

```

```

float middle_of_3( float a, float b, float c ) {
    if ( ( a <= b ) && ( a <= c ) ) {
        middle = ( b <= c ) ? b : c;
    } else {

```

```

        if ( ( b <= a ) && ( b <= c ) ) {
            middle = ( a <= c ) ? a : c;
        } else {
            middle = ( a <= b ) ? a : b;
        }
    }
}

void printDistance ( float distance ) {
    pos_1 = myservo_1.read();
    pos_2 = myservo_2.read();
    pos_1 = pos_1 - 90;
    pos_1_rad = pos_1 * 3.14 / 180;
    pos_2_rad = pos_2 * 3.14 / 180;

    x = dist * sin( pos_2_rad ) * cos( pos_1_rad );
    y = dist * sin( pos_2_rad ) * sin( pos_1_rad );
    z = dist * cos( pos_2_rad );

    Serial.print(x);
    Serial.print(',');
    Serial.print(y);
    Serial.print(',');
    Serial.print(z);
    Serial.println(',');
}

void loop() {
    if ( TaskFlag1 ) {
        TaskFlag1 = false;

```

```

        measure();
        Serial.print(Distance1);
        Serial.print(",");
        Serial.println(Distance2);
    }
}
ISR( TIMER2_COMPA_vect ) {
    TaskTimer1++;
    if ( TaskTimer1 > 499 ) {
        TaskTimer1 = 0;
        TaskFlag1 = true;
    }
}

```

```

void measure() {
    unsigned long start_time;
    unsigned long finish_time1;
    unsigned long finish_time2;
    unsigned long time_taken;
    boolean echo_flag1;
    boolean echo_flag2;

    digitalWrite( Trig1, HIGH );
    digitalWrite( Trig2, HIGH );
    delayMicroseconds( 10 );
    digitalWrite( Trig1, LOW );
    digitalWrite( Trig2, LOW );

    while( !digitalRead( Echo1 ) );
    while( !digitalRead( Echo2 ) );

```

```

start_time = micros();

echo_flag1 = false;
echo_flag2 = false;

while( ( !echo_flag1 ) || ( !echo_flag2 ) ) {
    if ( ( !echo_flag1 ) && ( !digitalRead( Echo1 ) ) ) {
        echo_flag1 = true;
        finish_time1 = micros();
        time_taken = finish_time1 - start_time;
        Distance1 = ( (float)time_taken ) / 59;
    }

    if ( ( !echo_flag2 ) && ( !digitalRead( Echo2 ) ) ) {
        echo_flag2 = true;
        finish_time2 = micros();
        time_taken = finish_time2 - start_time;
        Distance2 = ( (float)time_taken ) / 29.5;
    }
}

```

```

void serialEvent( Serial myPort ) {
    Input_string = myPort.readStringUntil( '\n' );
    println( Input_string );

    if ( Input_string != null ) {
        Input_string = trim( Input_string );
    }
}

```

```
String[] values = split( Input_string, ',' );

float a = float( values[1] ) - float( values[0] );
float b = float( values[0] );
float c = Baseline;
float d = c*1.414;
float d = sqrt( 150*150 + 100*100 );
float s = ( a + b + c ) / 2;

boolean distances_valid = true;
if (
    ( a < 0 ) ||
    ( b > d ) ||
    ( a > d ) ||
    ( ( s - a ) < 0 ) ||
    ( ( s - b ) < 0 ) ||
    ( ( s - c ) < 0 )
) {
    distances_valid = false;
    X = 1000;
    Y = 1000;
}

if ( distances_valid ) {
    float area = sqrt( s * ( s - a ) * ( s - b ) * ( s - c ) );
    Y = area * 2 / c;
    X = sqrt( b * b - Y * Y );
    print( "    d1: " );
    println( b );
    print( "    d2: " );
```


```
println( a );
print( " base: " );
println( c );
print( "offset: " );
println( Offset );
print( " s: " );
println( s );
print( " area: " );
println( area );
print( " X: " );
println( X );
print( " Y: " );
println( Y );
println( "" );
}
myPort.clear();
}
}
```

# ДОДАТОК Б

## ДЕМОНСТРАЦІЙНИЙ МАТЕРІАЛ

### Слайд 1

## ТИТУЛЬНИЙ ЛИСТ РОБОТИ



Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Кафедра КІТАМ

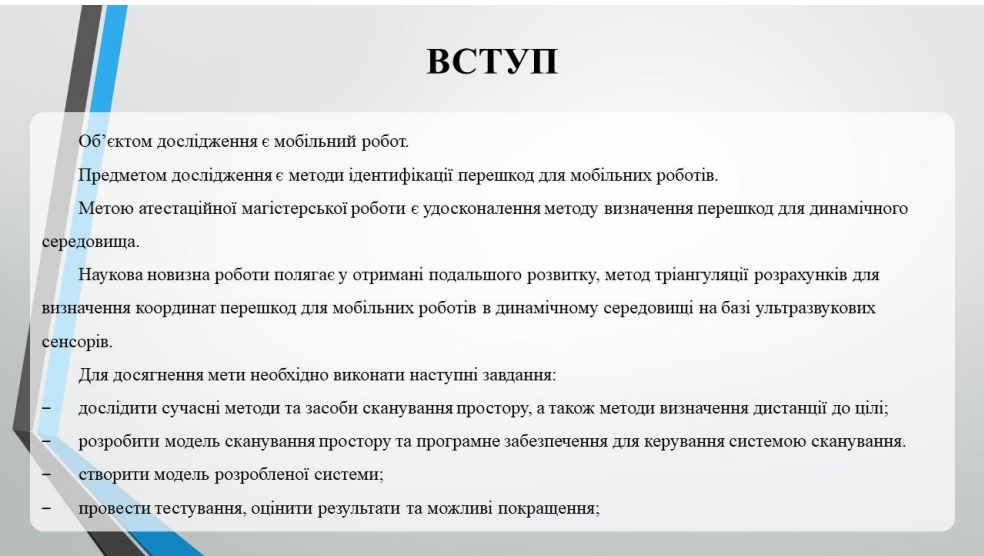
Магістерська атестаційна робота  
на тему: «Навігаційна система мобільного робота на основі  
скануючого далекоміра»

Студент:  
гр. КТРСМ-19-1  
Михайлов А.С.

Керівник:  
проф. Сіногін А.М.

### Слайд 2

## ВСТУП



## ВСТУП

Об'єктом дослідження є мобільний робот.

Предметом дослідження є методи ідентифікації перешкод для мобільних роботів.

Метою атестаційної магістерської роботи є удосконалення методу визначення перешкод для динамічного середовища.

Наукова новизна роботи полягає у отриманні подальшого розвитку, метод триангуляції розрахунків для визначення координат перешкод для мобільних роботів в динамічному середовищі на базі ультразвукових сенсорів.

Для досягнення мети необхідно виконати наступні завдання:

- дослідити сучасні методи та засоби сканування простору, а також методи визначення дистанції до цілі;
- розробити модель сканування простору та програмне забезпечення для керування системою сканування;
- створити модель розробленої системи;
- провести тестування, оцінити результати та можливі покращення.

## Слайд 3

## ПРИКЛАД СХЕМИ ТА РЕАЛІЗАЦІЇ СКАНЕРУ З ПОВОРОТНИМ УЛЬТРАЗВУКОВИМ МОДУЛЕМ



## Слайд 4

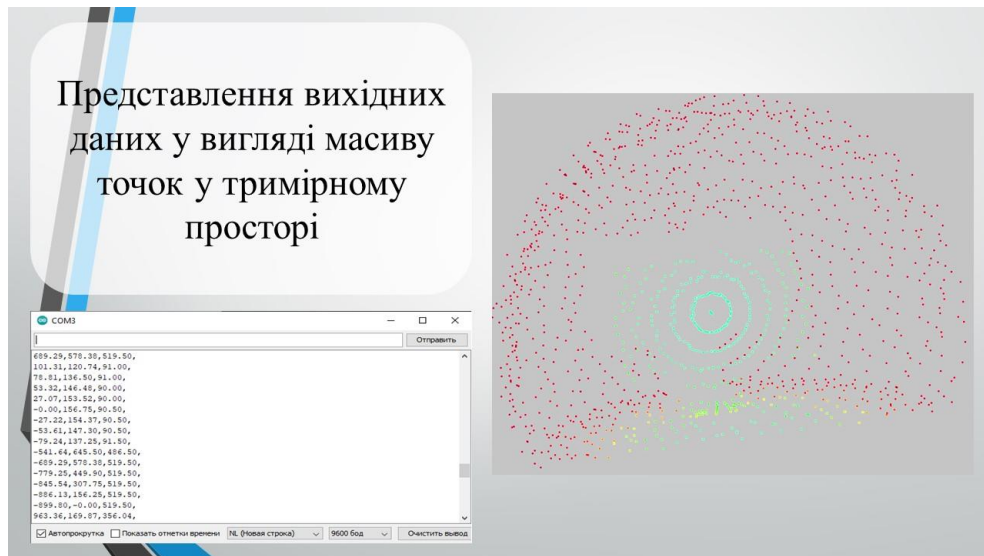
## ВІДНОШЕННЯ СФЕРИЧНОЇ СИСТЕМИ КООРДИНАТ ДО ДЕКАРТОВИХ

Перетворення сферичної системи координат у Декартову

В даній схемі далекомір відхилюється від початкової позиції та виміряє дистанцію до перешкоди.  
Дані кута відхилення потрібно перетворити з сферичної системи координат у Декартову завдяки приведеним формулам.

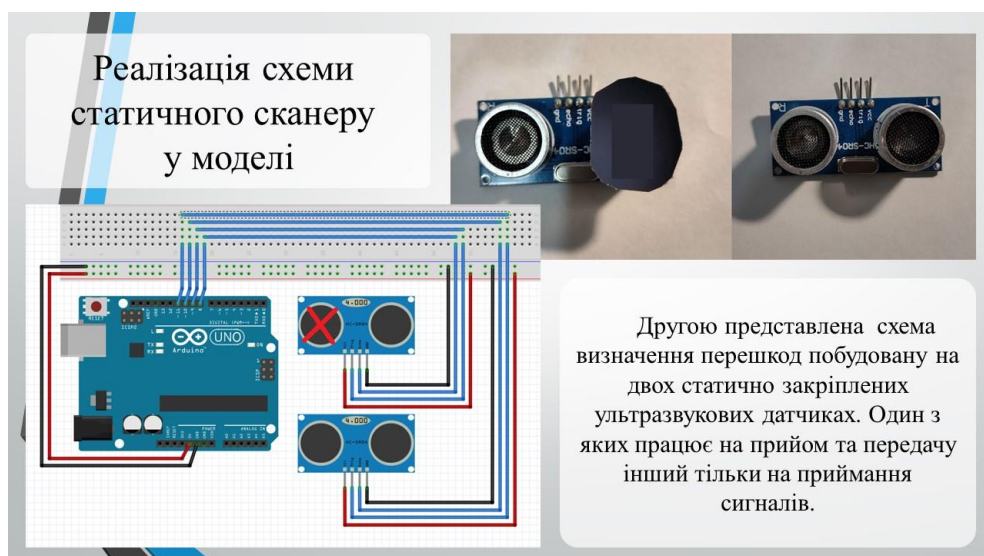
$$\begin{cases} x = r \sin \theta \cos \varphi, \\ y = r \sin \theta \sin \varphi, \\ z = r \cos \theta. \end{cases}$$

## Слайд 5

РЕЗУЛЬТАТ РОБОТИ СКАНУРУ З ПОВОРОТНИМ УЛЬТРАЗВУКОВИМ  
МОДУЛЕМ

## Слайд 6

## СХЕМА СИСТЕМИ З НЕРУХОДИМИ ДАТЧИКАМИ



## Слайд 7

## ПРИКЛАД РЕАЛІЗАЦІЇ СИСТЕМИ З НЕРУХОДИМИ ДАТЧИКАМИ

**Схема зон перекриття прослуховування**



За схемою була побудована модель, дана модель працює за рахунок прослуховування ділянки перекриття зон прослуховування двома приймачами. Обидва приймачі сканують одну зону.

## Слайд 8

## СХЕМА РУХУ СИНАЛІВ СИСТЕМИ З НЕРУХОДИМИ ДАТЧИКАМИ

**Принципи роботи сканеру з нерухомими датчиками**



Сигналу випущений з датчика А відбивається від перешкоди у всіх напрямках. Внаслідок чого ми можемо отримати час проходження сигналу від об'єкту до датчику А та В

## Слайд 9

## ПРИКЛАД ДАНИХ У ПРОЦЕСІ РОБОТИ СИСТЕМИ З НЕРУХОДИМИ ДАТЧИКАМИ

Цифрові результати отримані сканером з нерухомими датчиками

Отриманий час завдяки простим геометричним формулам з даними дистанції від сканеру до об'єкту та статичної відстані між датчиками можна отримати координати точки на площині.

```

1022.31,160.00

94.10,159.05

92.81,158.24

d1: 92.81
d2: 65.43001
base: 100.0
offset: 50.0
s: 129.12
area: 2948.771
X: 71.66307
Y: 58.975418

```

## Слайд 10

## РЕЗУЛЬТАТ РОБОТИ СИСТЕМИ З НЕРУХОДИМИ ДАТЧИКАМИ

Візуальні результати сканування простору системою з нерухомими ультразвуковими датчиками

```

d1: 92.81
d2: 65.43001
base: 100.0
offset: 50.0
s: 129.12
area: 2948.771
X: 71.66307
Y: 58.975418

```

## Слайд 11

### ВИСНОВКИ

#### ВИСНОВКИ

У ході виконання атестаційної роботи біло розроблено дві системи навігації, побудовані прототипи та написане програмне забезпечення для функціонування та тестування.

Безумовною перевагою є комбінування цих систем на мобільних платформах для отримання найкращих результатів при навігації у просторі.

Результати атестаційної роботи можуть бути використані для оснащення мобільних платформ промислового та побутового призначення та як основа для подальшого удосконалення ультразвукової орієнтації роботів у просторі.

