

## Сканирующий манипулятор

### *Общая постановка задачи.*

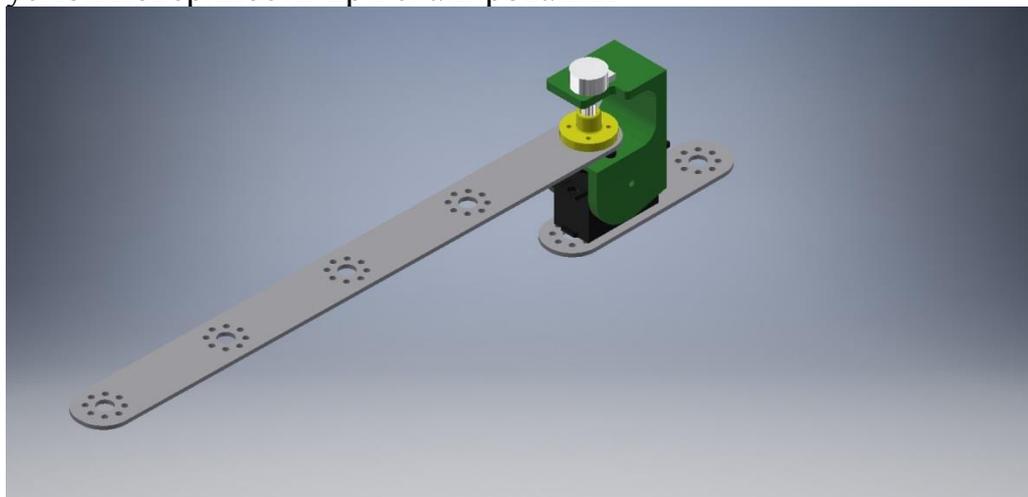
На участке контроля качества радиоэлектронного завода требуется внедрить роботизированную проверку дефектов поверхностного напыления материала на пластинах. В качестве дефекта выступает метка, нанесенная при помощи черного маркера, в качестве пластины – лист бумаги формата А4. В этом случае, контроль заключается в автоматическом определении координат  $[x, y]$  метки в пределах контура пластины. Контур пластины расположен в пределах границ сканируемой роботом области.

Группе разработчиков предлагается, из предложенного набора комплектующих собрать прототип двухзвенного плоского манипулятора, сканирующего поверхность пластины. Управляющая программа манипулятора должна предполагать возможность вычисления координат нанесенной контрольной метки.

### *Описание конструкции сканирующего манипулятора.*

Прототип манипулятора разрабатывается для горизонтальной плоскости. Он содержит 2 вращательных звена (алюминиевые балки в длину не более 20), плечо и предплечье. Обе оси вращения перпендикулярны горизонтальной плоскости сканирования и параллельны между собой. В обоих звеньях есть серводвигатель и потенциометр обратной связи по положению. Балка основания (размером не более 10 см), на которой закрепляется серводвигатель первого звена, жестко прижимается струбциной к столу. Пример расположения элементов для первого звена, показан на нескольких видах сборочного чертежа ниже.

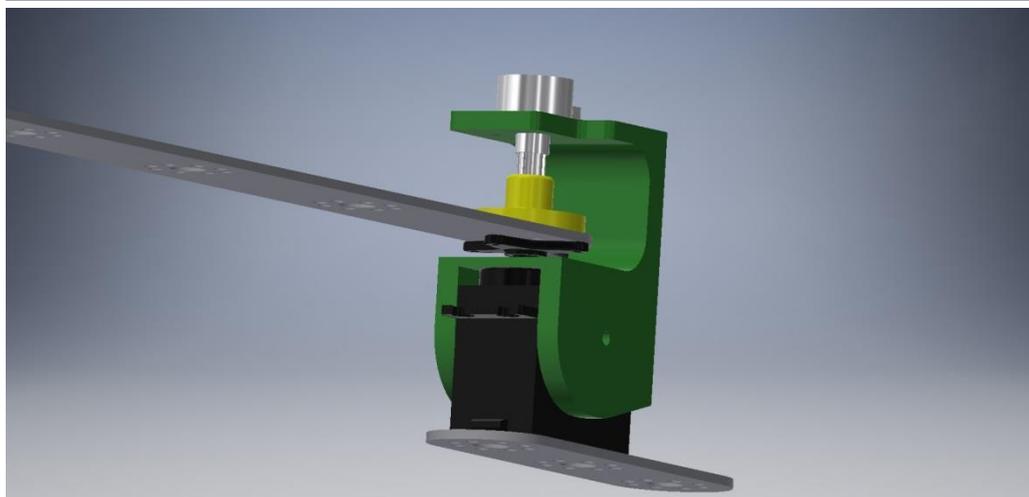
Необходимо конструктивно исключить возможность повреждения сканируемой поверхности при сканировании



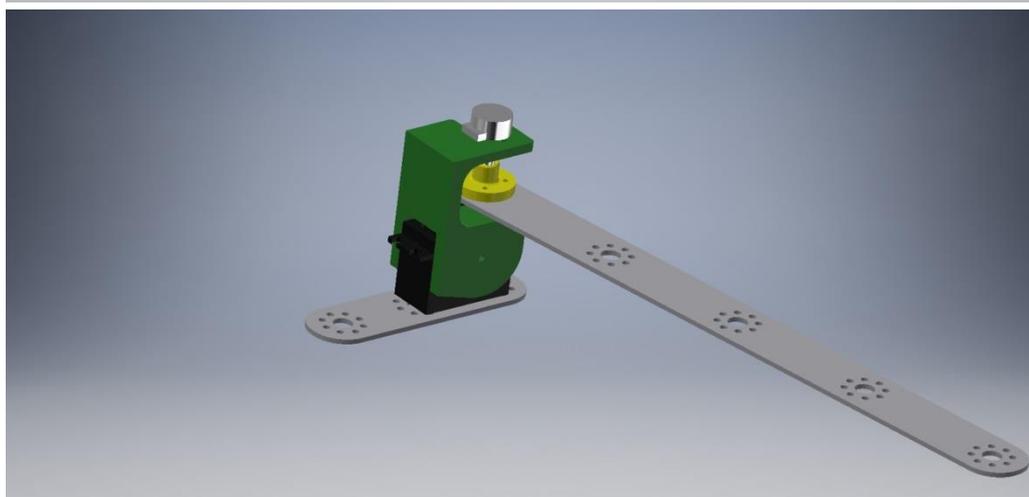
Вид 1.



Вид 2.



Вид 3.

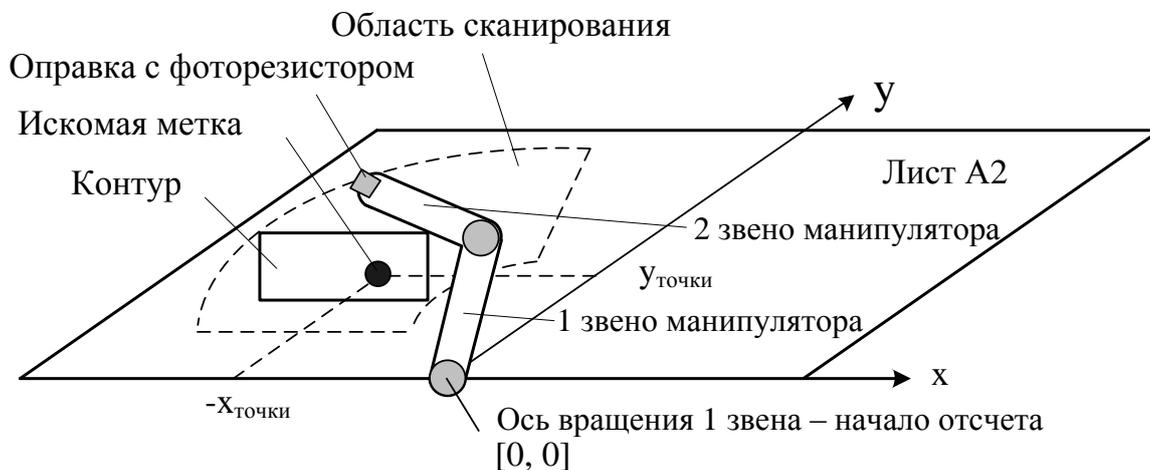


Вид 4.

Сканирующее устройство расположено на окончании второй балки и включает фоторезистор и светодиод, встроенные в корпус оправки (схема подключения светодиода и фоторезистора дана в приложении). Все радиоэлектронные и электромеханические устройства через макетную плату монтажными проводами подсоединяются к плате Arduino. Автономное питание обеспечивается 12 вольтовым элементом питания. Для понижения напряжения с 12 на 5 вольт используется DC-DC преобразователь. Подключение платы Arduino и загрузка управляющей программы происходит через USB-кабель.

Область сканирования представляет собой горизонтальную плоскость, в пределах которой разработчики самостоятельно намечают карандашом контур размером 210 на 297 мм для размещения в нем пластины с дефектом. В одном из 4 углов контура необходимо задать точку начала координат.

Область сканирования манипулятора с нанесенным контуром располагается на ватмане формата А2. Над заданной областью необходимо реализовать режим медленного плавного перемещения сканирующего устройства. Точка, обозначающая дефект на пластине, имеет диаметр 1 см, цвет - черный. Область сканирования, контур и искомая метка условно показаны на рисунке ниже.



Точность определения считается достаточной, если она не отклоняется более  $\pm 1,5$  см от замеров полученных экспертом-контролером вручную. Время одного полного сканирования не должно превышать 3,5 минут.

### ***Тестовые испытания.***

1. Осмотр конструкции. Манипулятор собран, произведен монтаж электрической схемы. Производится включение и ожидание запуска программы (манипулятор включен, но не двигается).

2. Разработана управляющая программа, при запуске которой происходит отработка манипулятором плавных движений над областью сканирования контура, найдя метку, манипулятор останавливается.

3. Разработанная управляющая программа предусматривает вывод координат  $[x, y]$  точки, над которой находится сканирующее устройство, при движении манипулятора, а также координат найденной метки при остановке в режиме монитора порта.

4. Разработанная управляющая программа выводит сканирующее устройство манипулятора в координаты тестовой метки, которую отметили сами разработчики в границах контура.

5. Разработанная управляющая программа определяет координаты случайно установленной метки, которую установил эксперт-контролер. Найдя эту метку, манипулятор останавливается.

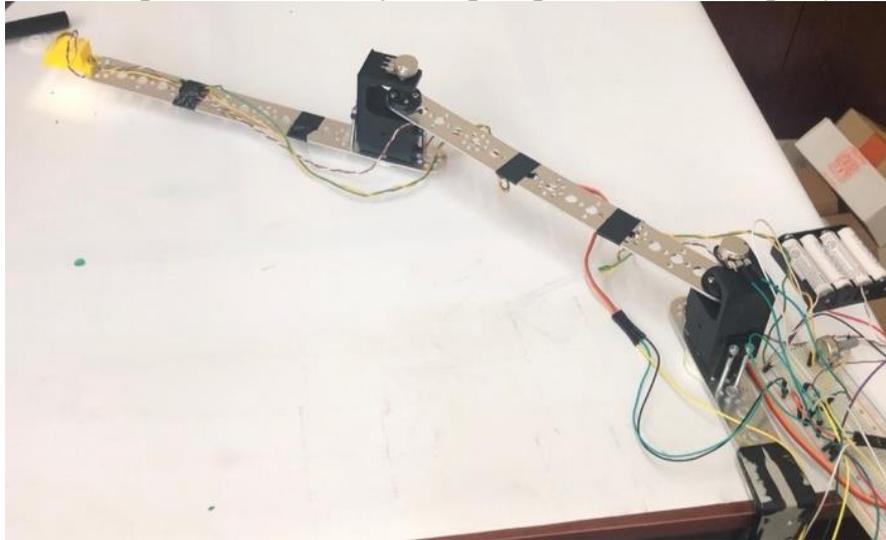
6. Выполненное сравнение координат вычисленных автоматически и экспертом-контролером вручную дают одинаковые результаты (допустимое отклонение  $\pm 1,5$  см). Количество попыток ограничено тремя (из них допускается выбрать лучший результат).

Комплект каждого участника включает:

№	Наименование	комплект
1	Серводвигатель 6 вольт	2
2	Arduino Uno	1
3	USB-провод к плате 1м.	1
4	Макетная доска	1
5	Провода папа-папа 20 см.	16
6	Шлейф (длинна180 мм)	1
7	Светодиод 350 mA 3.2 - 3.4 В (при необходимости)	1
8	Шлейф (длинна150 мм)	2
9	DC-DC понижающий преобразователь с 12 вольт на 6-7 Вольт.	1
10	Кабель с крокодилами 50см	2
11	Провода "папа-мама" 30см	12
12	Нейлоновая стяжка 10-15 см	4
13	Ватман А2	1
14	Оправка со встроенным светодиодом и фоторезистором	1
15	Струбцина зажимная	1
16	Полоса алюминиевая 30x2 мм, цвет серебро	1
19	Профиль алюминиевый П-образный 25x30x25x2x1000 мм	1
22	Изолента	1
24	Винт с потайной головкой, оцинкованный М4x40	8
25	Шаровая опора (металлическая)	1
26	Винт М4x20	5
27	Самоконтрящаяся гайка оцинкованная М4	14
28	Кронштейн на серводвигатель (3D печать)	2
31	Муфты (3D печать)	2
32	Потенциометр	2
35	Аккумулятор 12 Вольт (НЕ ВКЛЮЧАТЬ без DC-DC преобразователя)	1

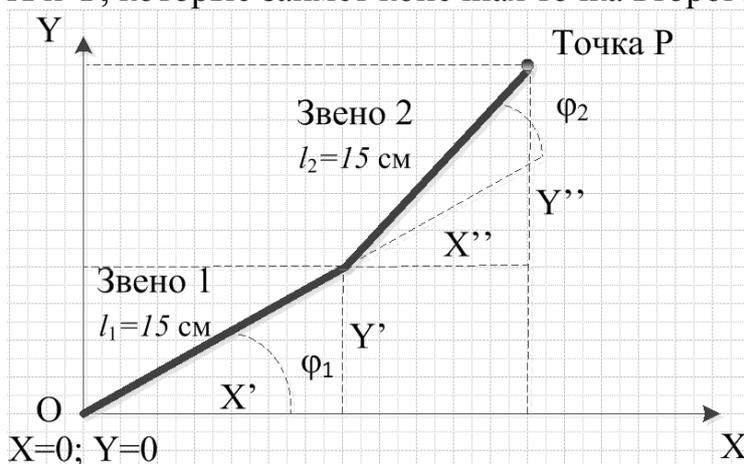
## Решение и ответы:

1. Внешний вид собранного манипулятора представлен на рисунке.



2. Вычисление координат конечной точки производится по формулам:

Манипулятор, содержащий два связанных друг с другом звена, оба звена которого вращательного типа, может выполнять перемещения в двумерной плоскости X-Y. Начало отсчета находится в начале координат ( $X=0$ ,  $Y=0$ ), там же находится точка крепления первого звена к основанию. Длины звеньев манипулятора, например,  $l_1=l_2=15$  см, второе звено присоединяется к первому через шарнир. Из вытянутого вправо, горизонтального положения, первое звено перемещается против часовой стрелки на угол  $\varphi_1=45^\circ$  относительно оси Y, второе звено перемещается против часовой стрелки на угол  $\varphi_2=45^\circ$  относительно первого звена. Требуется вычислить координаты X и Y, которые займет конечная точка второго звена.



При повороте звена робота координаты конечной точки вычисляются следующим образом:

$$x = l_2 \cdot \cos(\varphi_1 + \varphi_2) + l_1 \cdot \cos\varphi_1 = 15 \cdot \frac{\sqrt{2}}{2},$$
$$y = l_1 \cdot \sin\varphi_1 + l_2 \cdot \sin(\varphi_1 + \varphi_2) = 15 \cdot \frac{2 + \sqrt{2}}{2}.$$

$$x = 15 \cdot \frac{\sqrt{2}}{2}, \quad y = 15 \cdot \frac{2 + \sqrt{2}}{2}.$$

Решим задачу вычисления углов для двухзвенного манипулятора с вращательными парами аналитически.

Исходя из теоремы косинусов:

$$\varphi_2 = \arccos \left( \frac{x^2 + y^2 - l_1^2 - l_2^2}{2 \cdot l_1 \cdot l_2} \right),$$

Поделим уравнения системы друг на друга. Получим:

$$\operatorname{tg}(\varphi_1) = \frac{y - l_2 \cdot \sin(\varphi_1 + \varphi_2)}{x - l_2 \cdot \cos(\varphi_1 + \varphi_2)},$$

В первом приближении будем иметь:

$$\varphi_1 = \operatorname{atan} \left( \frac{y}{x} \right)$$

### 3. Реализация программного кода.

Первый возможный вариант базовой части программы, реализующей управляемое сканирующее движение.

```
#include <Servo.h>

Servo myservo1;
Servo myservo2;

int potpin1 = 0;
int potpin2 = 0;

void setup() {
  myservo.attach(9);
  myservo.attach(10);
  Serial.begin(9600);
}

void loop() {
  for (potpin1 = 0; potpin1 <= 180; potpin1 += 1) {
    myservo1.write(potpin1);
    delay(15);
    Serial.println(potpin1);
  }
  for (potpin1 = 180; potpin1 >= 0; potpin1 -= 1) {
    myservo.write(potpin1);
    delay(15);
    Serial.println(potpin1);
  }
  for (potpin2 = 0; potpin2 <= 180; potpin2 += 1) {
    myservo2.write(potpin2);
```

```

    delay(15);
    Serial.println(potpin2);
  }
  for (potpin2 = 180; potpin2 >= 0; potpin2 -= 1) {
    myservo.write(potpin2);
    delay(15);
    Serial.println(potpin2);
  }
}

```

Второй возможный вариант базовой части программы, реализующей управляемое сканирующее движение. Для второго варианта необходимо использование дополнительной библиотеки VarSpeedServo.

```
#include <VarSpeedServo.h>
```

```

// create servo object to control a servo
VarSpeedServo myservo1;
VarSpeedServo myservo2;

```

```

void setup() {
  pinMode(A0, INPUT);
  pinMode(A1, INPUT);
  pinMode(A2, INPUT);

```

```

// initialize serial:
Serial.begin(9600);

```

```

myservo1.attach(9);
myservo2.attach(8);
myservo1.write(120, 10);
myservo2.write(20, 10);
delay(3000);

```

```

}
int n = analogRead(A0);
int k = analogRead(A1);

```

```
void loop() {
```

```

  int i;
  k = analogRead(A1);
  Serial.println(k);

```

```

  while ((n < 300) ) {
    myservo1.write(120 - i, 50);
    delay(100);
    myservo2.write(20, 10);
    delay(1000);
    myservo2.write(100, 10);
    k = analogRead(A1);
    delay(3000);
    myservo2.write(20, 10);
    k = analogRead(A1);
    delay(3000);
    n = analogRead(A0);
  }
}

```

```

i = i + 5;
Serial.println(k);

}
myservo1.detach();
myservo2.detach();
}

```

Вариант 1. Для определения обобщенных координат манипулятора необходима управляющая программа, в которой задаются абсолютные координаты, лежащих в рабочей зоне манипулятора на основе аналитического решения обратной позиционной задачи.

```

function res=upr4(x,y);
%%peremennie
l1=0.15;
l2=0.15;
mnogo='manipulyator ne dostanet';
%%raschet if (x^2+y^2)>((l1+l2)^2)
disp(mnogo)
return
else
if or((x<0),(y<0))
disp('error')
return
else
q1=atan(y/x);
q2=acos((x^2+y^2-l1^2-l2^2)/(2*l1*l2));
q1*180/pi
q2*180/pi
end

```

Вариант 2. Программа для вычисления обобщенных координат и для заданных абсолютных координат, лежащих в рабочей зоне манипулятора на основе метода итераций.

```

function res=upr4_1(x0,y0); %%peremennie
l1=0.15;
l2=0.15;
flag=0;
mnogo='manipulyator ne dostanet'; malo='malo';
dq=pi/180; %%prirashenie uglov
eps=0.01; %%tochnjst' poiska %%raschet
if (x0^2+y0^2)>=((l1+l2)^2) disp(mnogo)
return
elseif y0<=0
disp(malo)
return
else
for q1=0:dq:pi
for q2=0:dq:pi x=l1*cos(q1)+l2*cos(q1+q2); y=l1*sin(q1)+l2*sin(q1+q2);
if and(abs(x-x0)<eps,abs(y-y0)<eps)
flag=1;
disp('reshenie naydeno')
q1*180/pi
q2*180/pi
return
break

```

```

end
if flag==1
break
end
end
end
end
end

```

### Числовые варианты решения.

Таблица соответствия обобщенных координат и абсолютных координат для различных точек рабочей зоны. Примем следующие значения длины звеньев манипулятора, например  $l_1=0.5$ ,  $l_2=0.3$ .

Для итерационного метода	Для аналитического решения
x	x
y	y
q1	q1
q2	q2
0	0
0.8	0.8
90	89,71
0	0
0.4	0.4
0.4	0.4
13	13,19
94	93,48
0.3	0.3
0.4	0.4
17	17,22
108	107,86
0.3	0.3
0.7	0.7
52	52,12
38	37,93