**TASK-4**

**2Dof**

**Algorithm to put robotic arm in position:**

Initial approaches: -

x= cos(Q1) + cos(Q2)

y=sinQ1 +sinQ2

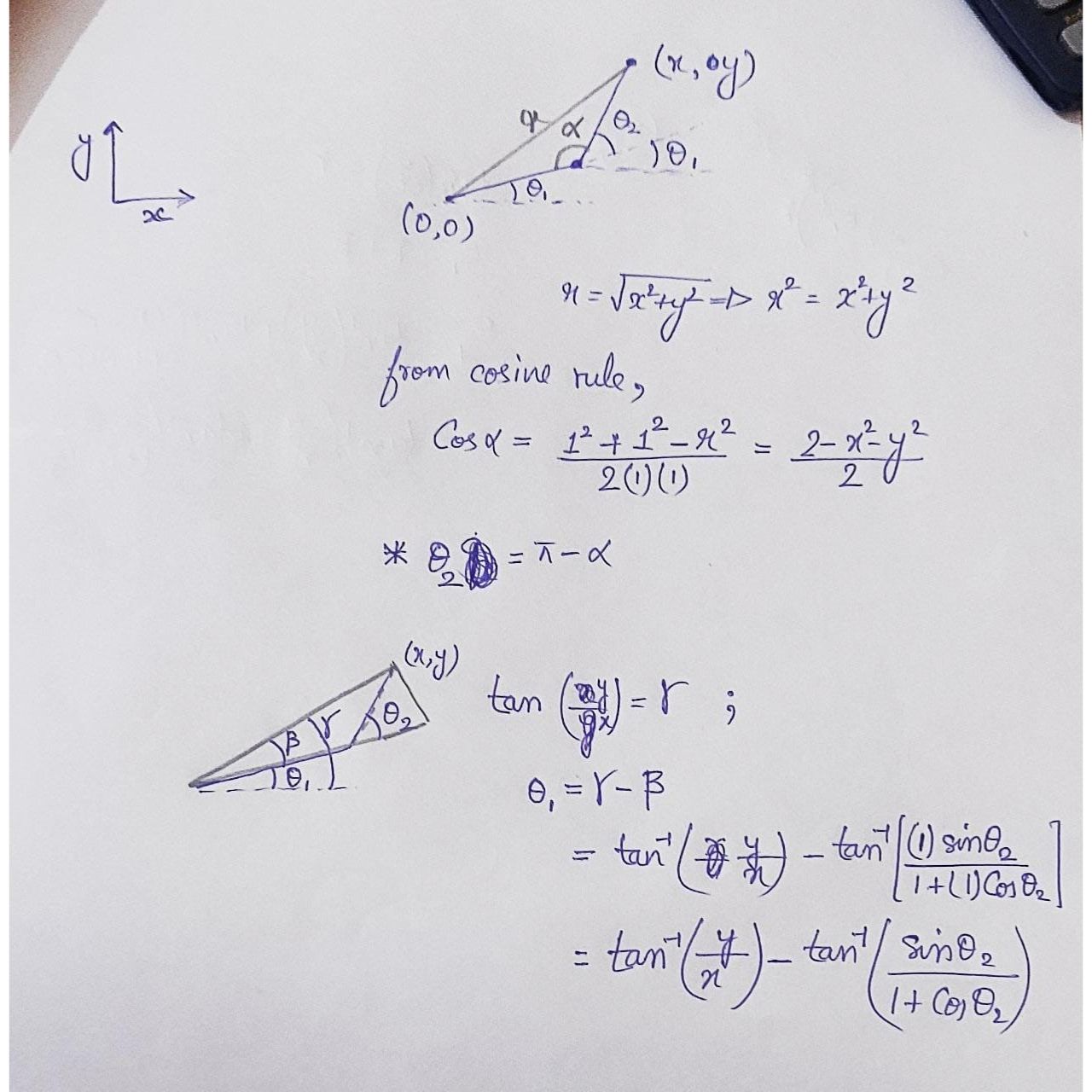
tan((Q1+Q2)/2) =x/y

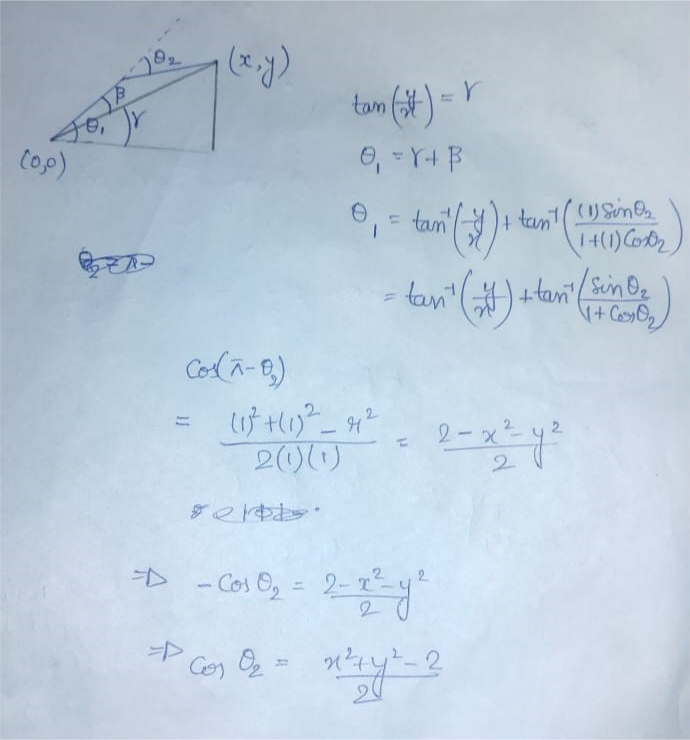
tan((Q1-Q2/2) =root((4-x^2-y^2)/(x^2+y^2))

We calculate the angle between (0,0) and (x, y) and initialize both theeta1 and theeta2 with the values, then we simultaneously increment theta1 and decrement theeta2 until we satisfy the condition-

1. Sin(theeta1) +sin(theeta2) ==y
2. Cos(theeta1) +cos(theeta2) ==x

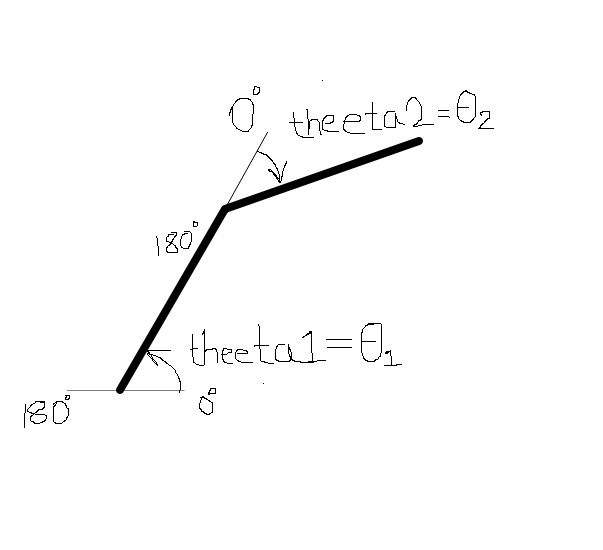
Used technique: -





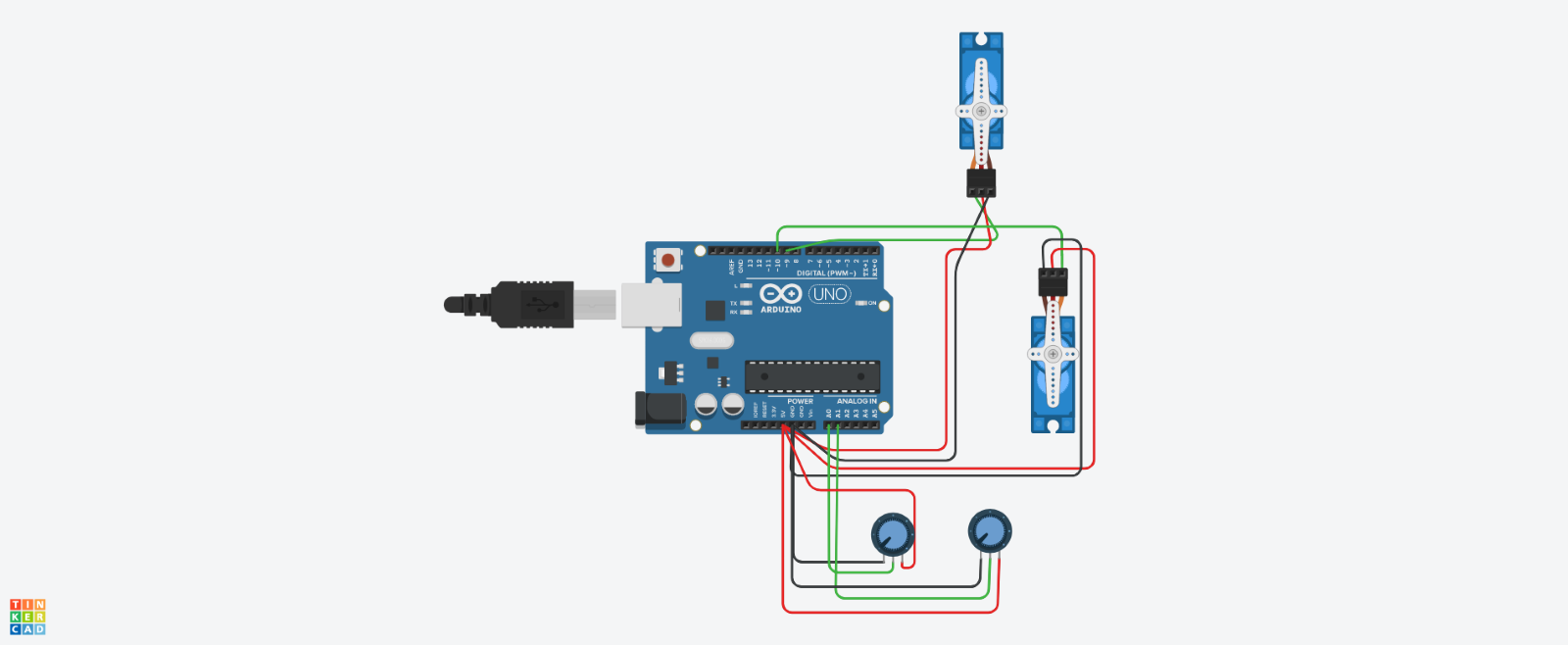
There are two solutions. But I have considered only the 2nd solution because the servo motor can bend up to 180 degrees only. The first type of solution (bent upward) can be easily obtained in the similar manner with slight change in the code and calculation. Also, this system is valid for first quadrant only.

Servo motors are attached to the arm as follows.



**Learning to code: -**

After the above logic was converted to code in C programming language, the challenge was to write the Arduino code. Components were arranged as per the need in the [Tinkercad](https://www.tinkercad.com/things/3BhMGAirkEj-agvdof2/editel?sharecode=E4SbYa65kzEhYqG5MgaGzqtAcDSzQRFj8_FJfHksWKE).



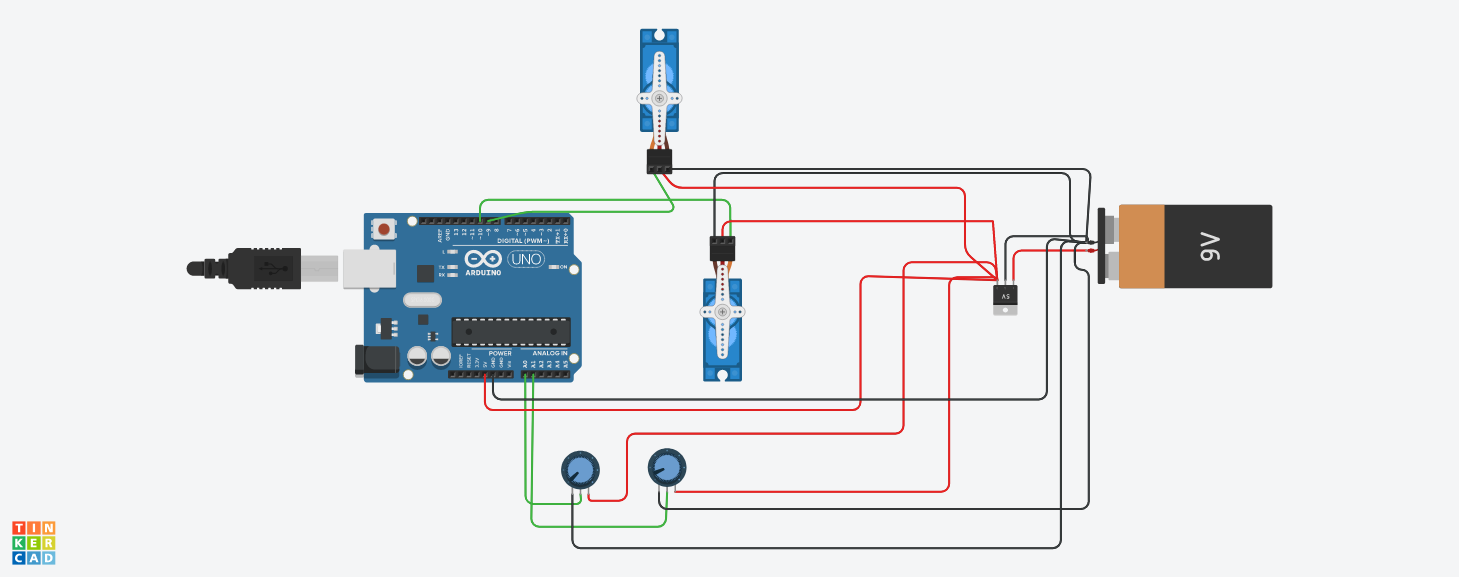
Potentiometer: - It provides a variable resistance, which we can read into the Arduino board as an analog value. Three wires to the Arduino board-5V, GND (powering) and one(wiper) in the respective analog pin (for input). With change in resistance, the input voltage changes. 0V corresponds to 0 analog input in the analog pin of Arduino. Similarly, 5V corresponds to 1023 analog input in the analog pin of the Arduino. This value can be read and returned by analogRead() function.

Servo motor: - It has three wires, two for powering (GND and 5V) and one for receiving digital signal from the specific pin. servo.attach(pin\_no., min\_pulsewidth, max\_pulsewidth ) function sets the servo signal pin to the particular pin number. Servo.write (angle\_in\_degree) sets the angle of servo motor wing.

Note-In the simulation, servo.write() was working fine so I kept it the same. But practically, this generally has a maximum of 180 steps between 0 to 180 degrees. So in that case servo.writeMicroseconds() can be used which allows upto 1000 steps.

We use map function to map the analog output from the potentiometer vs the coordinates. Here I used 20000 divisions for more accuracy. Rest of the code is shown in the [Tinkercad simulation](https://www.tinkercad.com/things/3BhMGAirkEj-agvdof2/editel?sharecode=E4SbYa65kzEhYqG5MgaGzqtAcDSzQRFj8_FJfHksWKE).

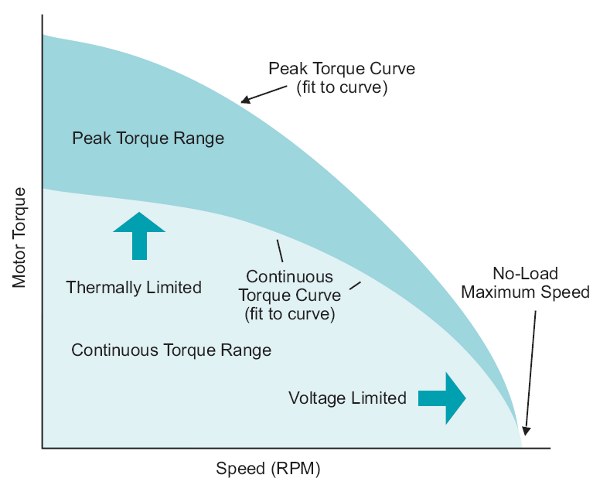
Rearranged circuit for keeping the current flowing in Arduino less thus preventing the damage.



**Speed control:**

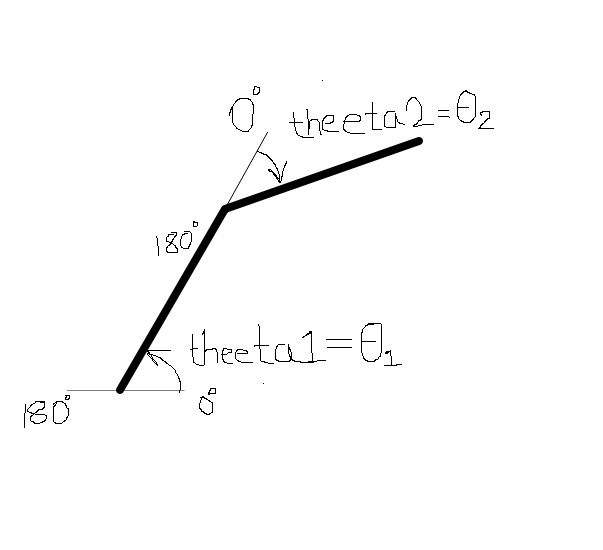
To control the speed, I have used a variable to take in speed in degree per second. Thus, time is calculated for every degree of movement. By using for loop and delay() function, such time is achieved which makes the speed so and so. It is to be noted that the speed is not 100% accurate as definitely some time must have been taken to run the statements of the code. But it surely tends to 100% accuracy.

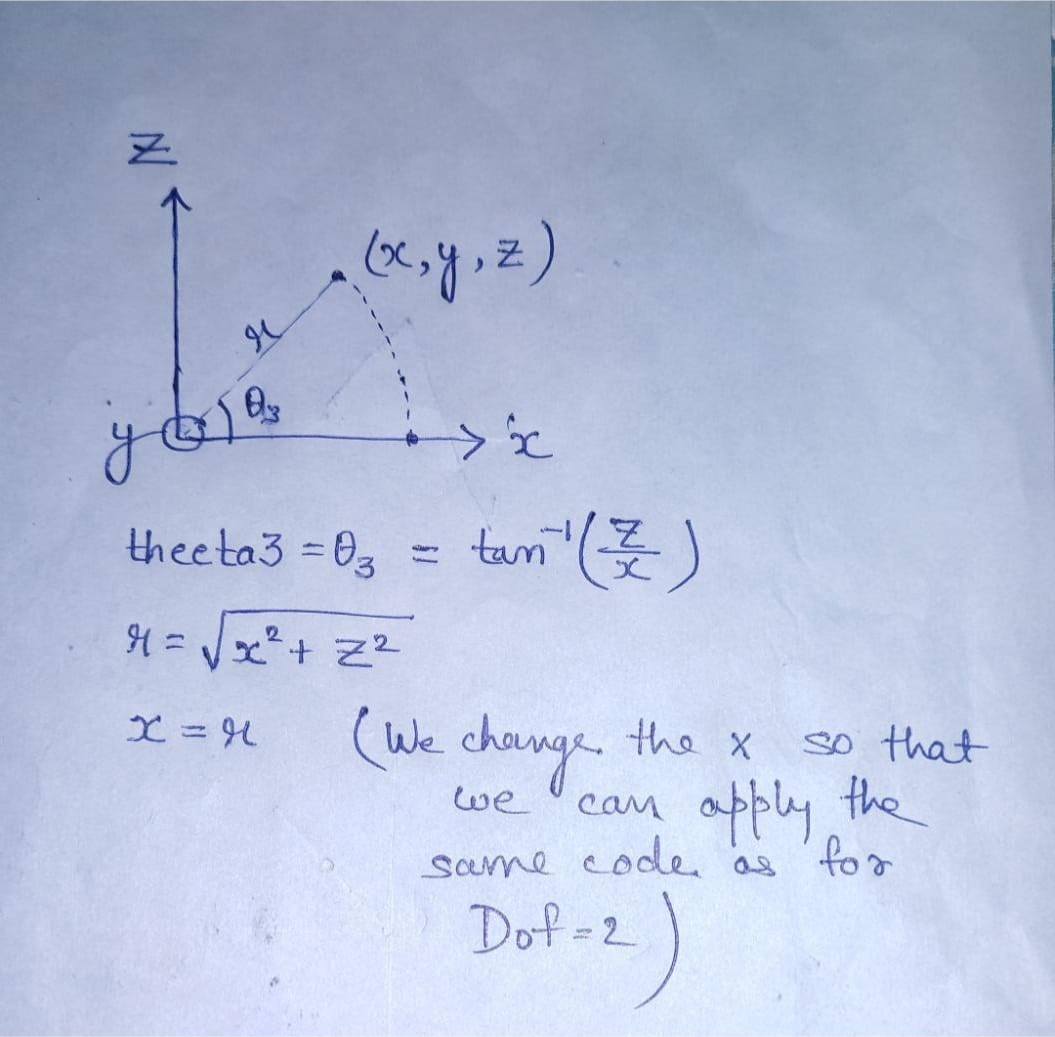
**Saving energy:**



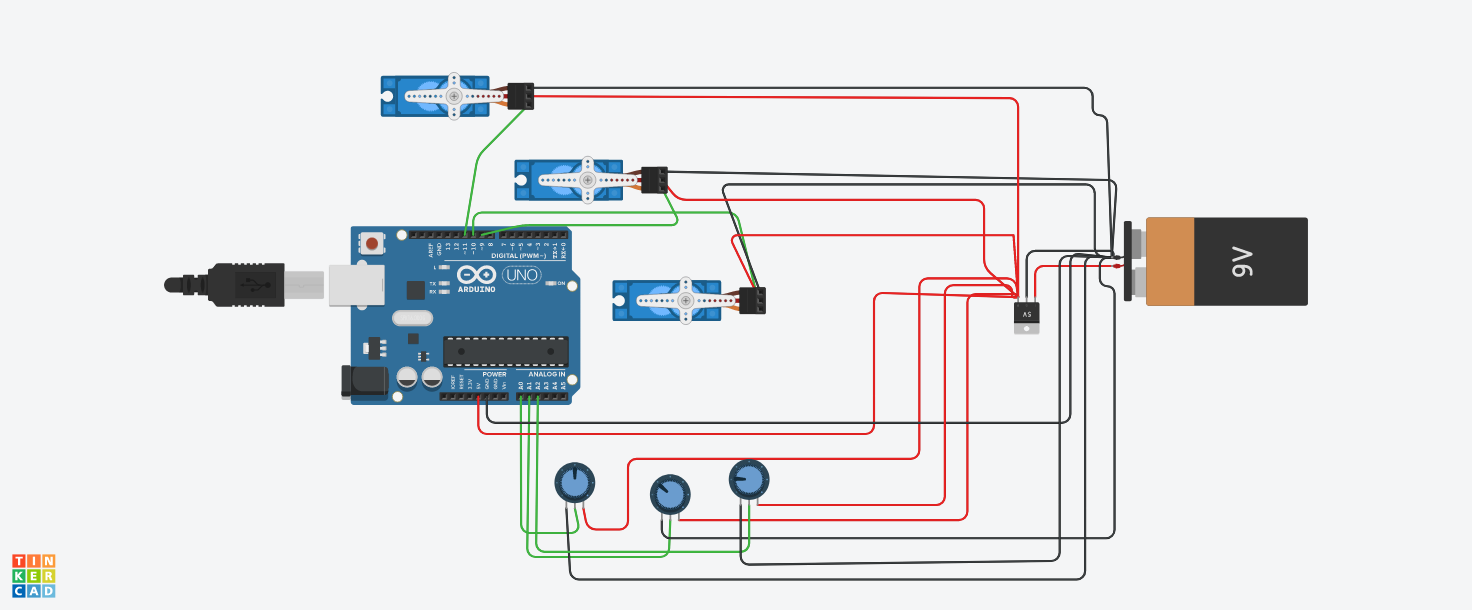
When the arm is bending, I drove the 2nd motor first so that moment of inertia of system first decreases so less overall energy is consumed. Similarly, when extending the arm, first motor is driven first. This way the power of the power source is saved which is beneficial for multiple usage. The code is written keeping this in mind.

**3DOF**





I wanted to reuse the code of the previous Dof2 arm. So, I used the x coordinate same as previous code. After some editing in the code, the Dof3 inverse kinematics is obtained.



In this case, I saved the **power** while using the third motor as well. When the projected radius has to increase, the third motor will run first and vice versa.

[Here](https://www.tinkercad.com/things/edWz0gvTrus-agvdof3/editel?sharecode=CNQng3UUrhlcT9wFFuJFGtbqVe_wCcOxUQkDbim7yvQ) is the link for the tinkercad simulation and [code](https://www.tinkercad.com/things/edWz0gvTrus-agvdof3/editel?sharecode=CNQng3UUrhlcT9wFFuJFGtbqVe_wCcOxUQkDbim7yvQ).