Oct. 6. 24

Today, I started picking what to research. I decided to improve my robot arm from a previous year.

Nov. 6. 24

I started working on background research; Arduino, servo motors, PCA9685, power supply, etc. Today, I also started my question and Hypothisis: If I program a homemade robotic arm using an Arduino to run at 6 volts instead of 5 volts, then it will be able to lift more weight, because the increased voltage will provide more power to the servo motors resulting in better lifting capacity and higher torque.

Nov. 13. 24

Today, I did more background research on robot arms, servo motors and Arduino. I discoverd there are different coding languages and two different types of software for the Arduino. My project uses the code language C++ and the two softwares are Arduino CLI (Command line interface) and Arduino IDE (Integrated development environment).

Nov. 30. 24

I started my variables today. My constant was the same type and size of servo motors. My Manipulated was the voltage; I used 5 and 6 volts. My responding variable was the amount of weight (in grams) my robot arm could lift with 5 and 6 volts

Dec. 7. 24

I worked on my materials today; servo motor brackets (u-shaped and multifunctional brackets), servo horns, servo motors, ball bearings, screws, nuts, DC power supply, PCA9685 PWM servo motor driver module, joysticks (to control the robotic arm), the Arduino R4 wifi, and solderless wires. and started learning to code.

Dec. 20. 24

Today I added more information to background research, materials, and started the procedure of building the joystick mount with the procedure for building the robotic arm. More learning about code and how to do it.

3

Dec. 22. 24

Today I started writing parts of code. I've learned that there are libraries you use at the start of your code to make it so that you don't have to start from nothing. Its basically "premade code". To add a library you have to type "#include<X>". X represents the library you want to use. For my project I only used one library, "Adafruit_PWMServoDriver.h". The library I used can control servo motors, LEDs, and even buzzers. The second part of my code: "Adafruit_PWMServoDriver pwm = Adafruit_PWM-ServoDriver();"

The second part of my code: "const int joystick1X =A0 const int joystick1Y=A1 Const int joystick2X =A2 const int joystick2Y = A3" establishes the connections of the two joysticks and the PCA9685 with the servo motors. The part of the code with A0, A1, A2, A3 all represent the connection of the PCA9685's pins, which is where the servo motors go. The reason I only use pins A0, A1, A2, A3, is because I only have four servo motors. This part of the code also tells the arduino that when the X axis of the joystick #1 is moved the servo motor in pin A0 on the PCA9685 has to be moved as well, same goes with the y axis on joystick 1 with A1 on the PCA9685, the x axis on joystick 2 with A2 on the PCA9685, and lastly the y axis on joystick 2 withe the PCA9685.

Dec. 28. 24

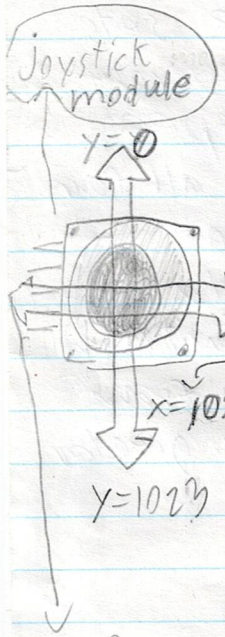
Today I continued my background research and wrote more of my code with the current knowledge I have. The third part of my code: "const int speed=5;" this part of my code sets a max speed that my servo motors can spin. For the fourth part of my code "int currentAngle[4]={90, 90, 90, 90};" initializes the current angle of all four servo motors when the circuit is powered on. The third part of my code is: "void setup() {pwm.begin(); pwm.setpwmfreq(50); for (int i=0; i<4; i++) {setServoAngle(i, 90);}}". The "pwm.begin" part initializes the PCA9685 and "gets it ready" to send PWM signals. "setpwmfreq(50);" this part of the code sets the frequency of the signal sent to the servo motors. "(int i=0; i<4; i++)" means it repeats this section of code four times (for all four servo motors) once the arduino sees that the code is repeated four times, it will try to repeat it a fifth but because of this part: "i<4" (with (i) being the variable) which will stop the code from repeating a fifth time because (i) is now greater than 4.
> "setServoAngle(i, 90);"

Jan. 12. 25

Today I learned more code. for the fifth part of my code: "void loop(){ int joystick1XValue = analogRead(joystick1X); int joystick1YValue = analogRead(joystick1Y); int joystick2XValue = analogRead(joystick2X); int joystick2YValue = analogRead(joystick2Y)". this part of my code initializes both of the joysticks X axis and y axis.

Jan. 17. 25

I wrote and learned more code today. I learned about the joysticks and how they have potentiometers built into them. These potentiometers divide voltage into small segments. They can also measure resistance in the joysticks X axis or y axis or both, then they move the component (your are controling with the joysticks) to the desired position. (In my case, this would be servo motors) Potentiometers have a scale (on both axis) that go from 0-1023. for my code: "int targetServoPos = map(joystick1XValue, 0, 1023, 0, 180); int targetServoPos = map(joystick1YValue, 0, 1023, 0, 180); int targetServoPos = map(joystick2XValue, 0, 1023, 0, 180); int targetServoPos = map(joystick2YValue, 0, 1023, 0, 180);" this is using the potentiometers to measure the resistance of both the x and y axis to move the servo motors.



Jan. 23. 25

Today I did more of my code. "SmoothMoveServo(0, targetServo1Pos); SmoothMoveServo(1, targetServo2Pos); SmoothMoveServo(2, targetServo3Pos); SmoothMoveServo(3, targetServo4Pos);" this part of my code allows the servo motors to be moved based on the position of the joysticks. "delay(20);" I added a delay of 20 miliseconds for the robotic arm to move slower and to allow time for updates. "Void smoothMoveServo(int servoNum, int targetAngle) int neutralThreshold = 50;" this part of my code is checking to see if the joystick is at its neutral position (which is around 512). "if (abs(targetAngle - currentAngle[servoNum]) < neutralThreshold)" this piece of code calculates the difference between the target angle and the current angle of the servo motor. Then it can tell how much the servo motor needs to move. "if (currentAngle[servoNum] != targetAngle) if (current angle [servoNum] < target Angle) else currentAngle [servoNum]--;" this part of my code checks if the current angle is different from the target angle. If they are equal, the servo motor will not move but if they arent equal, the servo motor will move to the designated angle so they will be equal.

7

Jan. 28. 25

Today I finished my code with the last part of it: "pwm.writeMicroseconds(servoNum, pulseLength);" this part of the code controls the servo motors by using Pulse width modulation (PWM). It sends a pulse to the servo motor and can tell the position of the servo motor.

Today I also started and finished all but 2 of the trials for my experiment. The rest were finished on the 29th of Jan.

Jan. 31. 25

I did the procedure and observations.

Feb. 1. 25

I did my conclusion, sources of error, and real world applications.

1

Jan. 28. 25

☒ = Was not able to lift
☑ = was able to lift

trials #1 | #2

Temp	6 volts	5 volts	temp
21.5	100g ✓	100g ✓	18.9
22.4	200g ✓	200g ✓	21.2
22.5	300g ✓	300g ✓	21.7
22.5	400g ✓	400g ✓	22.5
22.9	500g ✓	500g ✓	23.1
23.1	600g ✓	600g ✓	23.7
23.7	700g ✓	700g ✓	24.0
23.7	800g ✓	800g ✓	24.5
24.0	900g ✓	900g ✓	26.4
24.2	1000g ✓	1000g ✓	27.1
24.5	1100g ✓	1100g ✓	27.5
24.7	1200g ✓	1200g ✓	28.0
24.9	1300g ✓	1300g ✓	29.8
25.3	1400g ✓	1400g ✓	30.5
26.3	1500g ✓	1500g ✗	
	1600g ✗	1600g ✗	

Jan. 28. 25

2

☒ = was not able to lift

☑ = was able to lift

Trials:	#3		#4	
Temp	6 Volts		5 Volts	
21.4	100g	✓	100g	✓ 20.9
22.3	200g	✓	200g	✓ 21.4
22.6	300g	✓	300g	✓ 22.7
23.0	400g	✓	400g	✓ 23.1
23.7	500g	✓	500g	✓ 23.6
23.8	600g	✓	600g	✓ 24.4
24.2	700g	✓	700g	✓ 24.9
24.3	800g	✓	800g	✓ 25.3
24.8	900g	✓	900g	✓ 26.5
25.0	1000g	✓	1000g	✓ 27.2
25.5	1100g	✓	1100g	✓ 27.7
26.3	1200g	✓	1200g	✓ 28.1
26.9	1300g	✓	1300g	✓ 28.6
27.9	1400g	✓	1400g	✓ 29.4
30.9	1500g	✓	1500g	☒ 2
	1600g	☒		

Jan. 29. 25

trials:	#5	#6	Temp
	6 Volts	5 volts	☒ = was not able to lift
			☑ = was able to lift
20.8	100g	100g ✓	21.9
20.9	200g	200g	23.2
21.6	300g	300g ✓	24.9
22.7	400g	400g ✓	25.7
23.3	500g	500g ✓	26.3
23.8	600g	600g ✓	27.1
24.3	700g	700g ✓	27.6
24.9	800g	800g ✓	28.3
25.1	900g	900g ✓	29.5
26.8	1000g	1000g ✓	29.8
27.1	1100g	1100g ✓	30.6
27.6	1200g	1200g ✓	30.9
28.3	1300g	1300g ✓	31.3
28.7	1400g	1400g X	
28.7	1500g X	1500g X	

4

Jan. 29. 25

trials: #7 | #8

temp	6 Volts	5 volts	Temp
23.5	100	100	24.1
23.7	200	200	24.3
24.1	300	300	24.5
24.7	400	400	25.3
25.2	500	500	25.9
27.1	600	600	26.3
28.5	700	700	26.7
28.7	800	800	27.1
28.8	900	900	27.8
29.3	1000	1000	28.5
29.5	1100	1100	29.2
30.4	1200	1200	30.1
30.6	1300	1300	30.6
30.8	1400	1400	32.6
31.0	1500	1500 X	
31.6	1600		
32.1	1700		
	1800		

5

Jan. 28. 25

☒ = was not able to lift

☑ = was able to lift

trials:	#9	#10	
	6 Volts:	5 Volts	
21.8	100g ✓	100g ✓	23.9
22.1	200g ✓	200g ✓	24.1
22.5	300g ✓	300g ✓	24.2
23.3	400g ✓	400g ✓	25.4
23.9	500g ✓	500g ✓	25.6
24.7	600g ✓	600g ✓	26.3
25.3	700g ✓	700g ✓	28.8
25.8	800g ✓	800g ✓	27.2
26.3	900g ✓	900g ✓	27.3
26.9	1000g ✓	1000g ✓	28.2
27.5	1100g ✓	1100g ✓	28.7
28.3	1200g ✓	1200g ✓	29.5
28.7	1300g ✓	1300g ✓	30.1
29.1	1400g ✓	1400g ✓	30.8
	1500g ✓	1500g ✗	
29.7	1600g ✓		
	1700g ✗		