Build a Robotic Arm

https://www.sciencebuddies.org/science-fair-projects/project-ideas/Robotics_p050/robotics/arduino-robotic-arm
(https://www.sciencebuddies.org/science-fair-projects/project-ideas/Robotics_p050/robotics/arduino-robotic-arm)

Procedure PDF Date: 2023-10-26

Experimental Procedure

Note: This engineering project is best described by the **engineering design process,** as opposed to the **scientific method.** You might want to ask your teacher whether it's acceptable to follow the engineering design process for your project before you begin. You can learn more about the engineering design process in the Science Buddies Engineering Design Process Guide (http://www.sciencebuddies.org/science-fair-projects/engineering-design-process/engineering-design-process-steps).

1. Before you start building anything, you should come up with a design for your robotic arm. Try to answer the following questions (this is not an exhaustive list):
   a. What do you want to use your arm for?
   b. What objects will the arm need to pick up?
   c. How big will you want your arm to be? Will you need to order larger motors than the ones linked in the materials section?
   d. How can you design a gripper to pick up those objects?
   e. What will the arm do with the objects once it has picked them up?
   f. How many degrees of freedom does your arm need in order to accomplish this task? In other words, how many motors do you need for the required motions?
   g. How will you control the arm (buttons, potentiometers, etc)? Will you use the same type of control for each motor?
2. Draw a sketch of your arm. You do not necessarily have to stick with this design. You might need to change it later, and that is OK. The engineering design process is **iterative**, meaning you might go through multiple designs before you arrive at your final product. Figure 6 shows an example sketch of the arm in this project. You should also think about what materials you will use to build the arm. The example arm in this procedure uses popsicle sticks and double-sided foam tape. This works for a simple arm, but is not very sturdy and cannot lift heavy objects. You will need sturdier materials and connection methods if you want to build a more heavy-duty arm.

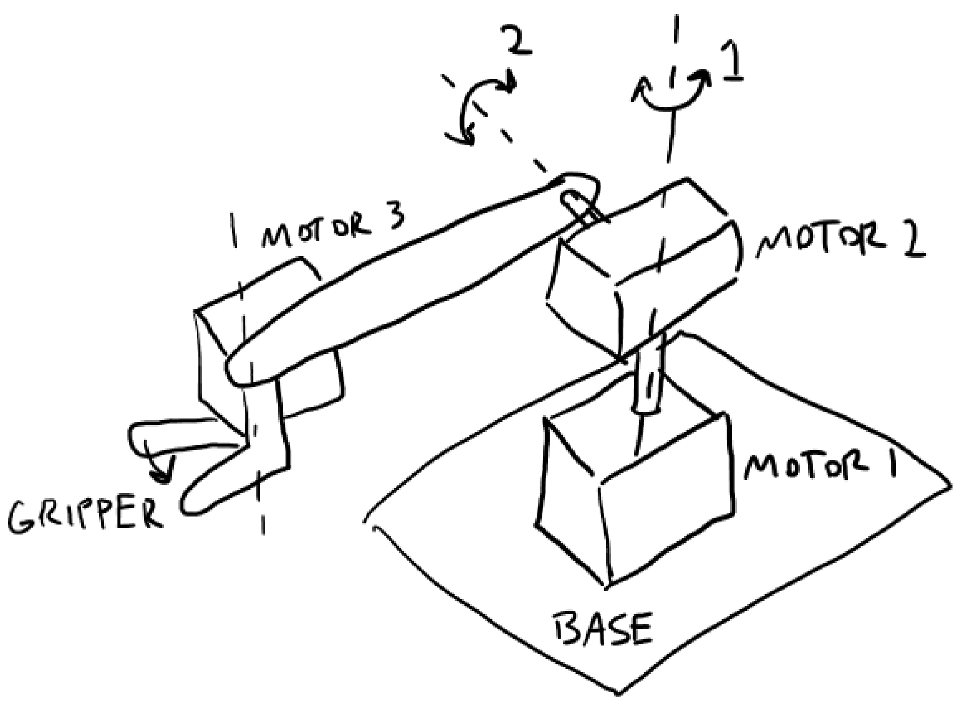

Figure 6. A hand-drawn sketch of the robotic arm design. Each motor's axis of rotation is shown with a dashed line.

3. Make a complete materials list for all the parts you will need to build your arm. It can take at least a few days for parts that you order online to arrive, so it will delay your project if you have to order more parts later. Carefully go over the materials list and make sure you are not missing any parts before you place your order.
4. Once you have your parts, you can start to build and test your circuit *before* you build the physical structure of your arm. Your exact circuit will depend on how many servo motors your arm has and what inputs you use to control them. Figure 7 shows a circuit which three motors, each of which is controlled by two buttons (one to rotate the motor in each direction, so six total). When wiring your breadboard, keep in mind that you will need physical access to the controls, so you do not want to obscure the buttons under other wires.

Here is a list of connections in this circuit. Note that servo motors from different manufacturers may have different conventions for their color-coding.

a. Servo 1
    i. Orange wire to Arduino pin 10
    ii. Red wire to 5V
    iii. Brown wire to ground
b. Servo 2
    i. Orange wire to Arduino pin 7
    ii. Red wire to 5V
    iii. Brown wire to ground
c. Servo 3
    i. Orange wire to Arduino pin 4
    ii. Red wire to 5V
    iii. Brown wire to ground
d. Button 1A
    i. One contact to Arduino pin 9
    ii. Other contact to ground
e. Button 1B
    i. One contact to Arduino pin 8
    ii. Other contact to ground
f. Button 2A
    i. One contact to Arduino pin 6
    ii. Other contact to ground
g. Button 2B
    i. One contact to Arduino pin 5
    ii. Other contact to ground
h. Button 3A
    i. One contact to Arduino pin 3
    ii. Other contact to ground
i. Button 3B
    i. One contact to Arduino pin 2
    ii. Other contact to ground

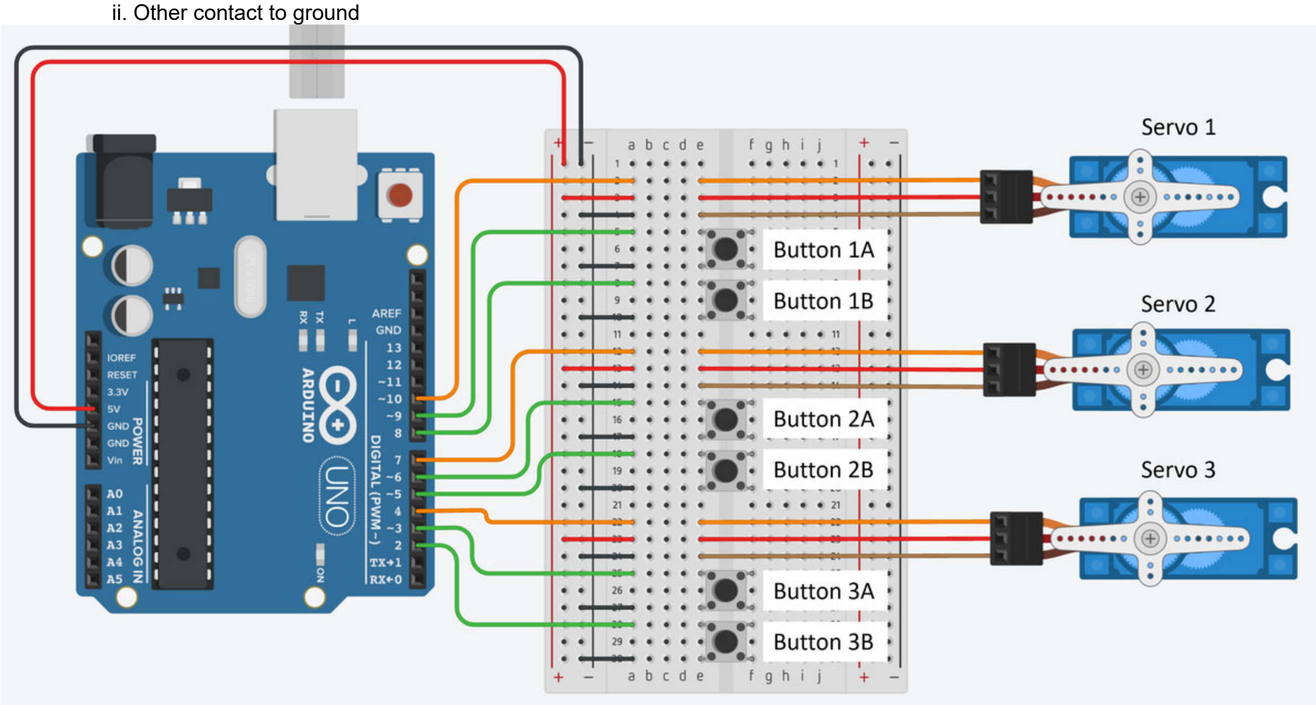

Figure 7. Breadboard diagram to control three servo motors with two buttons each. Click here for a larger version of the diagram (https://www.sciencebuddies.org/cdn/Files/19589/5/three-servo-breadboard-diagram-labeled.png). Image generated with Tinkercad.

5. This example code (http://www.sciencebuddies.org/cdn/Files/19582/5/servo_control_with_buttons.ino) will control the servos with the buttons if you build the circuit shown in Figure 7. You can also see a Tinkercad simulation of the circuit (https://www.tinkercad.com/things/0lObHqklYdi) here. You may wish to add more features to your code, like limiting the range of certain motors (which defaults to 0–180°). For example, the gripper in Figure 6 does not need to rotate a full 180°. If you use other inputs (like potentiometers or joysticks) or additional motors, you will need to modify the code accordingly. See our How to Use an Arduino (http://www.sciencebuddies.org/science-fair-projects/references/how-to-use-an-arduino) for tutorials about potentiometers and joysticks.
6. Test that your code, inputs, and motors are all working *before* you start building the physical structure for your arm. It will be easier to solve these problems now and re-wire things if necessary than after you have started building the arm.
7. Build the physical structure for your arm. There are several things to consider when building:
    a. How will you attach the motors to each other, the base, and other parts (like popsicle sticks)? Double-sided foam tape works for this purpose (Figure 8), but is not very sturdy. It might not be strong enough to support the weight of the entire arm, especially if you plan to add more motors. Note that the servo motors have holes on the sides that you can use to attach them to something (like a piece of wood or 3D-printed mounting bracket) using screws or bolts.

b. How will you manage all the wires? You do not want the arm to get tangled in the wires as it tries to move around, and you do not want the wires to be too tight and limit the arm's range of motion. Zip ties or twist ties are useful for organizing the wires (Figure 9).
c. Think about your gripper design. The gripper in this project is just made from two flat popsicle sticks (Figure 10). Can you design a gripper with more friction or a better shape for picking up your target objects?

Figure 8. Motor taped to a cardboard base with double-sided foam tape.

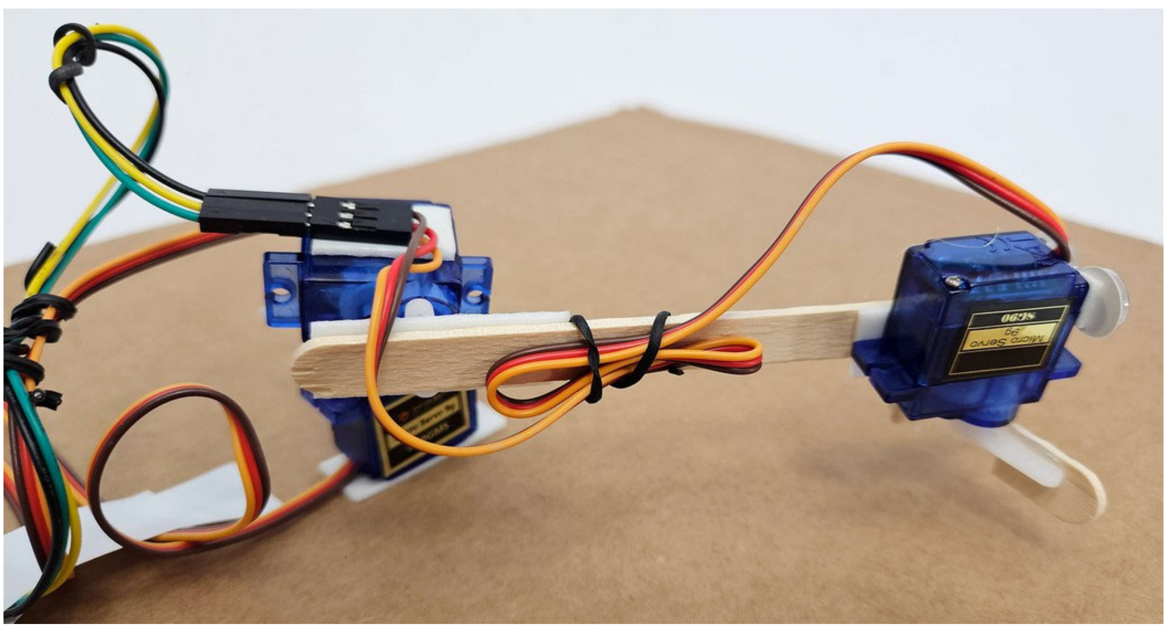

Figure 9. Servo motor wires held in place with a twist tie.

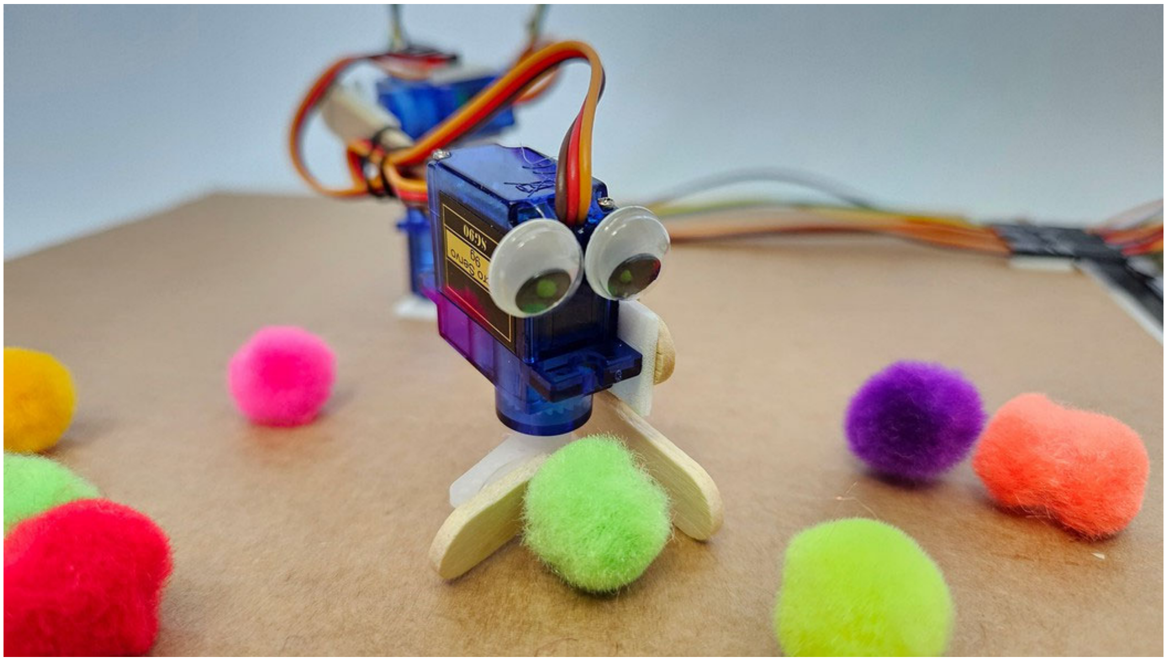

Figure 10. Close-up picture of the gripper made from two popsicle stick ends.

8. After you have finished building, test your arm's range of motion before you try lifting and manipulating objects. Can all the motors move through their full range without getting stuck or anything on the arm breaking? Do you need to repair anything or change something about your design?
9. After you have tested your arm's range of motion, try lifting and manipulating objects. What exactly you do here will depend on your original goal for your arm (for example, picking something up and placing it in a cup). Experiment with using the arm. Does the gripper do a good job grabbing the objects? Is the arm easy to control using the controls you have designed? Is it too fast? Too slow? What could you change about your code or your arm design to make it easier to use or improve its performance?
10. Continue iterating and improving your robotic arm until you are satisfied with its performance. See the Variations (#makeityourown) section for even more ideas about what you can do with your robotic arm.