



Arduino Servo Tester



by mbagrianski

This Instructable is meant to supplement the **Modified InMoov Robot** Instructable, currently being written. Here, we demonstrate how to control a servo from an Arduino without a computer. The use of this portable interface greatly speeds up the prototyping process, when determining limits for servo rotation. It's especially useful when you have a lot of servos, as we do.

If you are using this Instructable as part of the Modified InMoov Robot series, please keep in mind that this is an unfinished version of the control panel, mounted on the back of the robot. You can use it as we did, to help determine your servo limits. **This Instructable serves as part 1 of the Panel Build.**

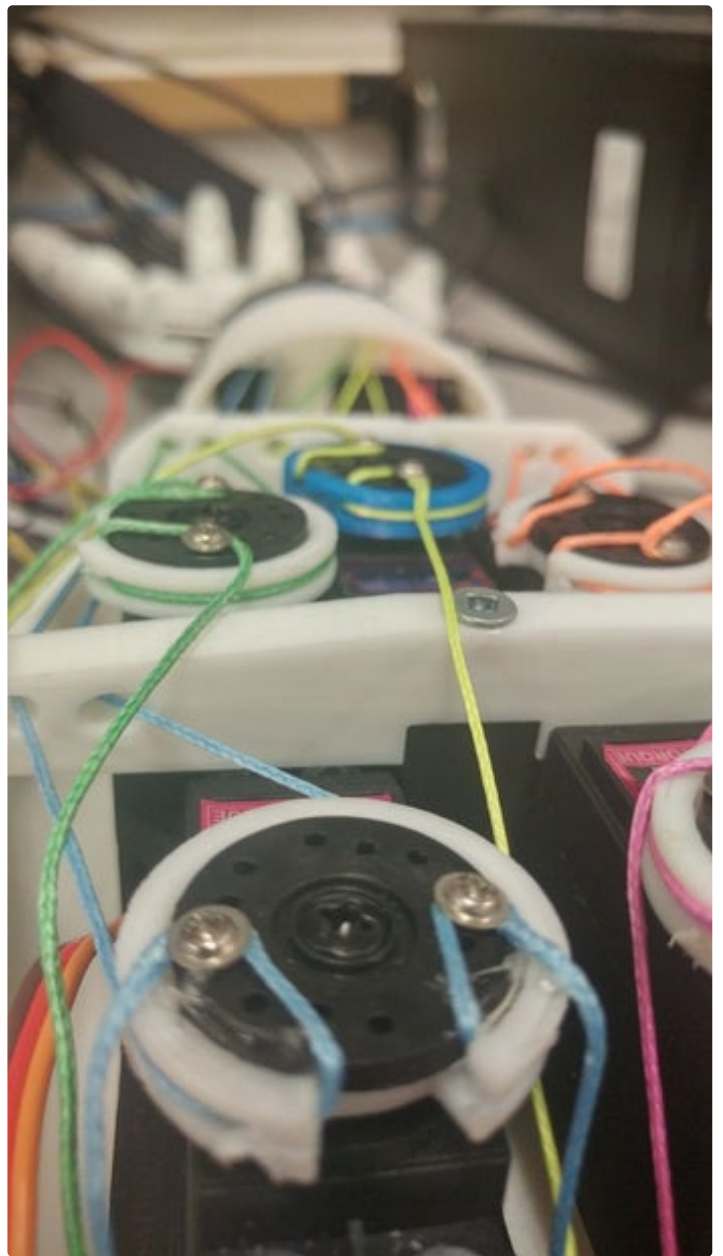
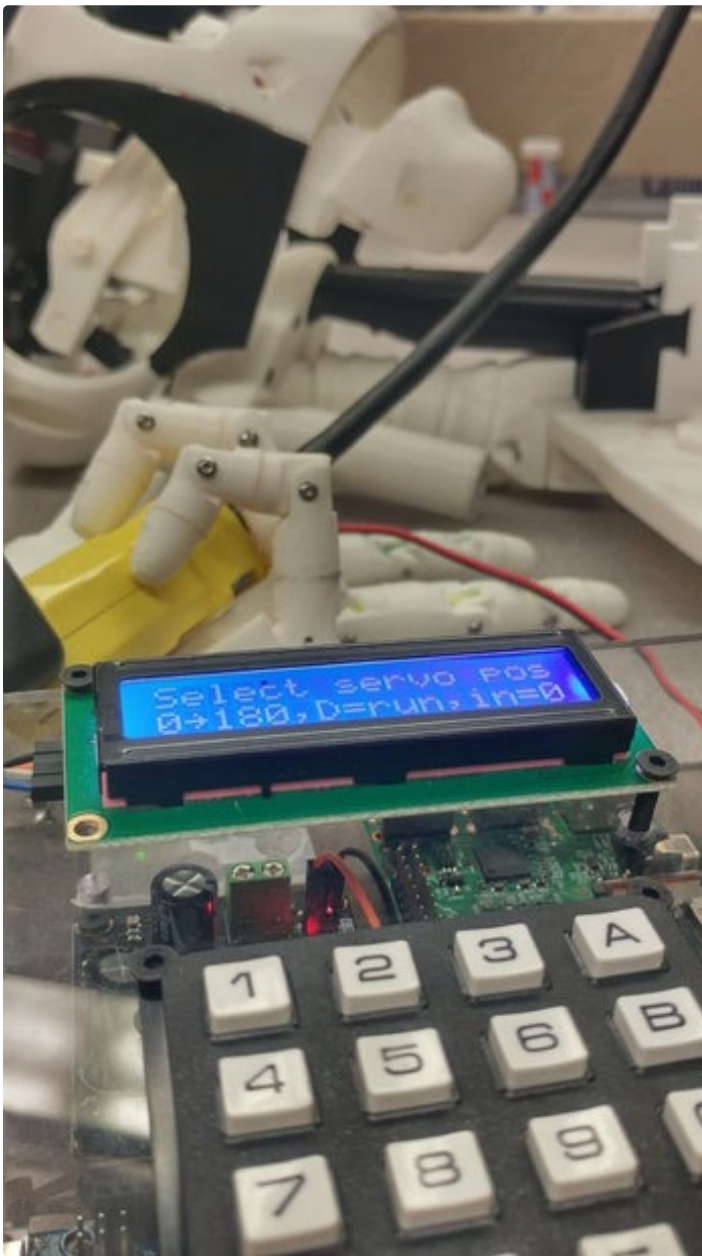
If you are just hoping to control a few servos with our method for another project, or just for fun, this Instructable is still for you, so don't leave the page! Please note that some images have notes on them, so look at each one individually.

Let's get started!

Supplies:

Not all supplies are listed here, please read through the entire Instructable.

- 1 x Arduino Uno
- 1 x power module, 5A DC-DC step down converter (We used [this](#) module but you could also use something like [this](#))
- 1 x I2C LCD ([this](#) is what we used, but any serial LCD or practically any standard LCD works)
- 1 x 4x4 matrix keypad ([for example](#))
- A breadboard and/ or a [perf board](#). If you are using the perf board you will need header pins such as are found [here](#), as well as all necessary equipment for soldering.
- Male to male, male to female, female to female jumper wires; assorted resistors, LEDs, and breadboard wires.
- Some sort of baseboard. We used 1/4" transparent acrylic. You can use any immalleable, easy-to-drill material.
- Hinges, screws, nuts/bolts, superglue



Step 1: Baseboard

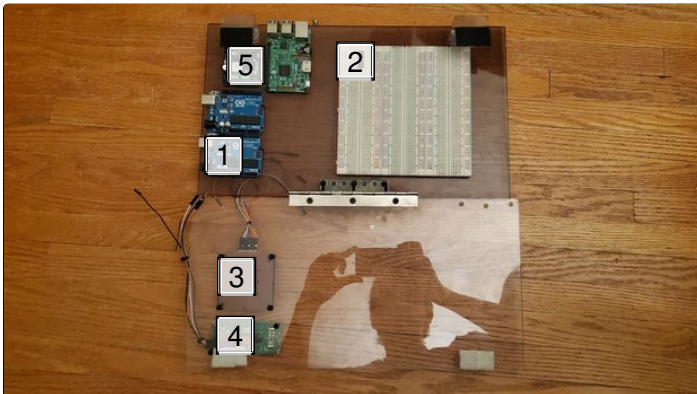
You will want to make a sturdy base for your electronics. If you are making this for the Modified InMoov Robot, it will need to accommodate another Arduino Uno (maybe Mega), a Raspberry Pi3, and several breadboards and/or perf boards for servo/sensor control. Our base panel was about 7x15". You can view the photos above for the general layout of the modules. We set up the keypad and LCD on the cover so that the panel would only have to be opened in case of failure, prototyping or maintenance. As you can see, the cover hinges from the bottom and latches on at the top by means of some velcro.

Try to organize your microcontrollers on one side of the board. The power module isn't powerful enough to drive all your servos on the completed robot. In fact, it can't even drive 3 high-torque servos of the type used in the forearm without the internal current switch

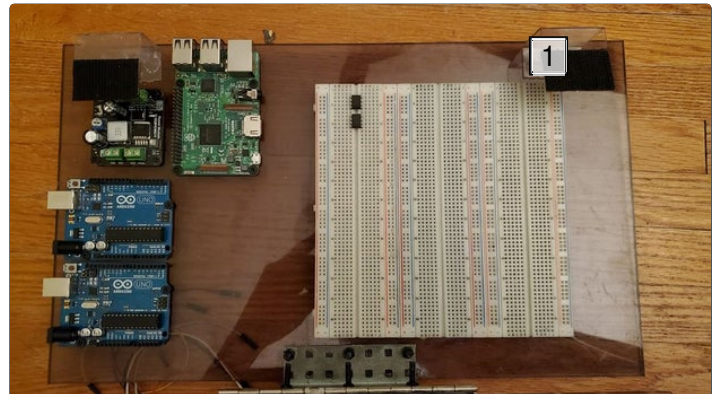
shutting off power. In any case, the code sample provided only works for one servo object, so you can test with this setup. Position this module close to the controllers as this will be their power regulator, as well as provide power for some micro servos that drain irrelevant stall current- but that's for later...

Set up the LCD and Keypad on the exterior of the cover that you're using, or off to the side, but close to the Arduino.

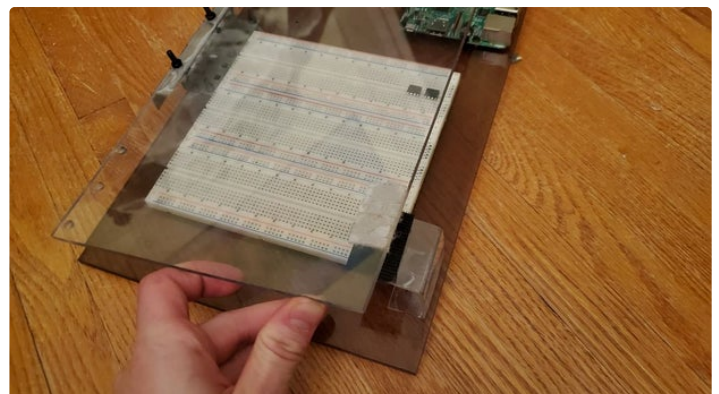
If you're doing this for yourself, still try to configure your board in a similar layout. You may not need the top cover, nor will you need 3 breadboards- but components need to be laid out in a logical way, such as the last photo of another project, which also uses a keypad and LCD. You can replace the circuit on the left with a breadboard, or omit it entirely.

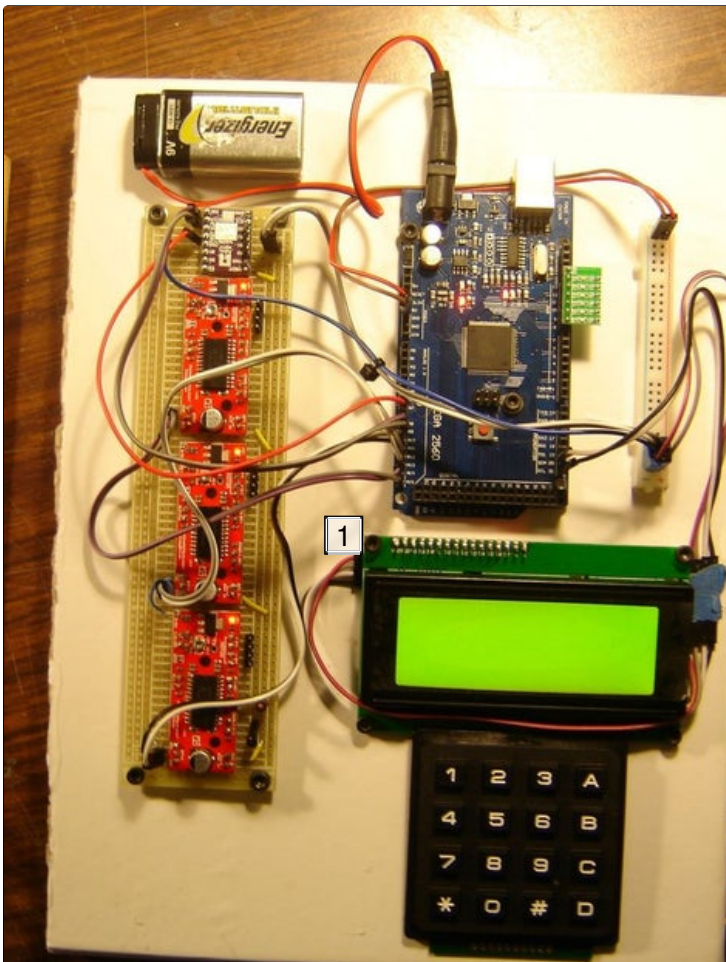


1. Arduino Uno that is used for this project
2. Breadboard that is used for this project
3. Keypad underside
4. Serial LCD underside
5. Power module



1. Detail of Velcro strip used to latch onto the lid.





1. This is the basic layout you want to have

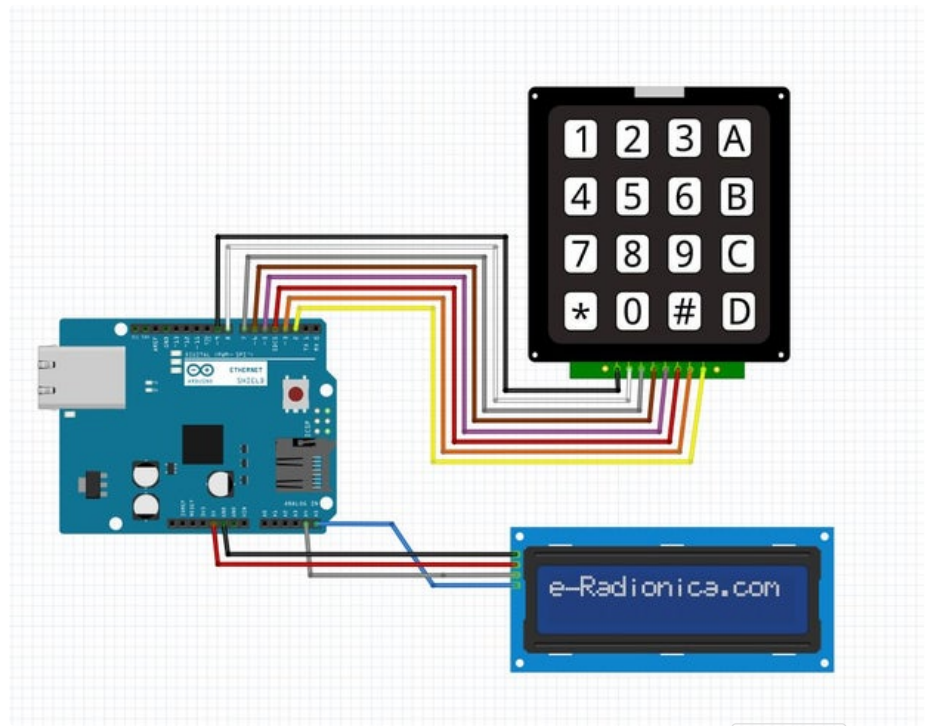
Step 2: Testing the I2C LCD and Keypad

We're now going to go ahead and test the keypad and LCD at the same time. Make the connections as seen in the picture above, and run the attached code. You will need to download the ZIP libraries for [liquidCrystal_I2C.h](#) if you haven't got it already; as well, you will need to import the Keypad.h and Wire.h libraries (this can be done in the IDE).

The Fritzing file is also attached. You don't need to power anything from the power module yet, although

you could if you wanted to. The Arduino will have enough power from the USB cable, which you upload code by.

When you upload your code, the LCD should prompt you to press a key on the keypad; once pressed, the LCD prints out the key pressed. You may have to change the layout of the keypad matrix; see the comments in the code for clarification.



<https://www.instructabl...>

Download



<https://www.instructabl...>

Download

Step 3: Controlling the Servo

Now we'll take this project one step further and add a servo. You need to connect it to Digital Pin 10, as is set in the code, or you can change that to suit your needs. Follow the above Fritzing diagram; it is the same as the previous one with the exception of the servo, RGB common cathode LED and buzzer. The latter two are used to supplement the user interface- warnings and status will be displayed on the LED and through the buzzer. We see above how we're using the completed servo tester to set limits for the InMoov Robotic Forearm and Hand.

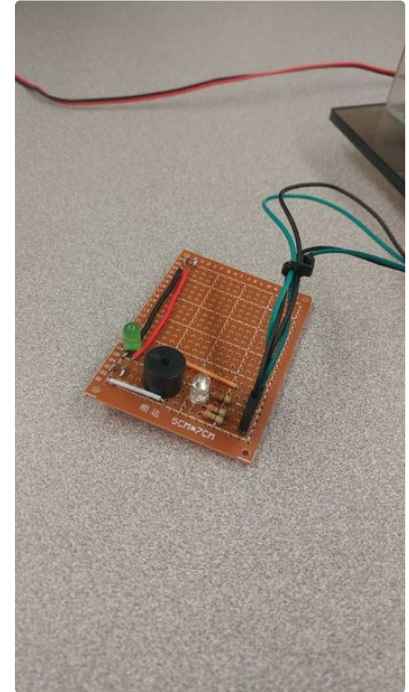
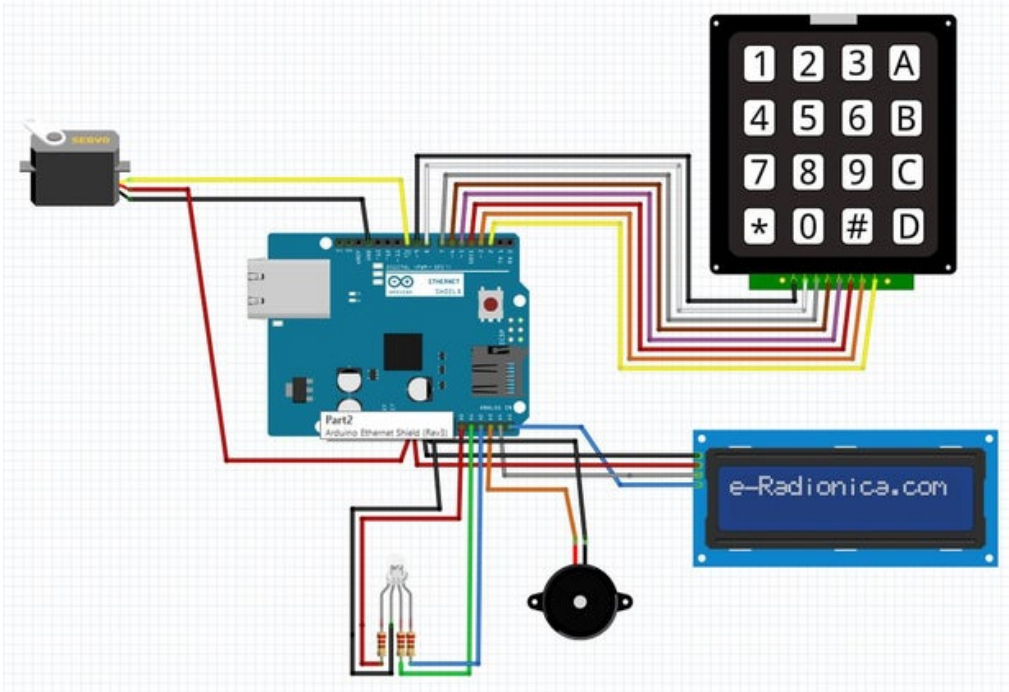
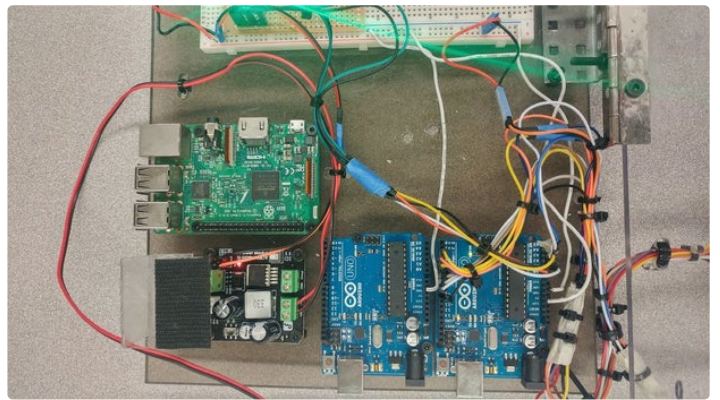
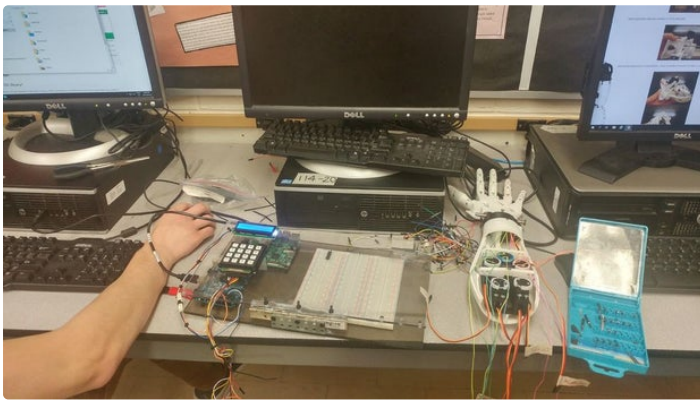
Download the attached code and run it. Pressing A brings up an info panel; everything else should be

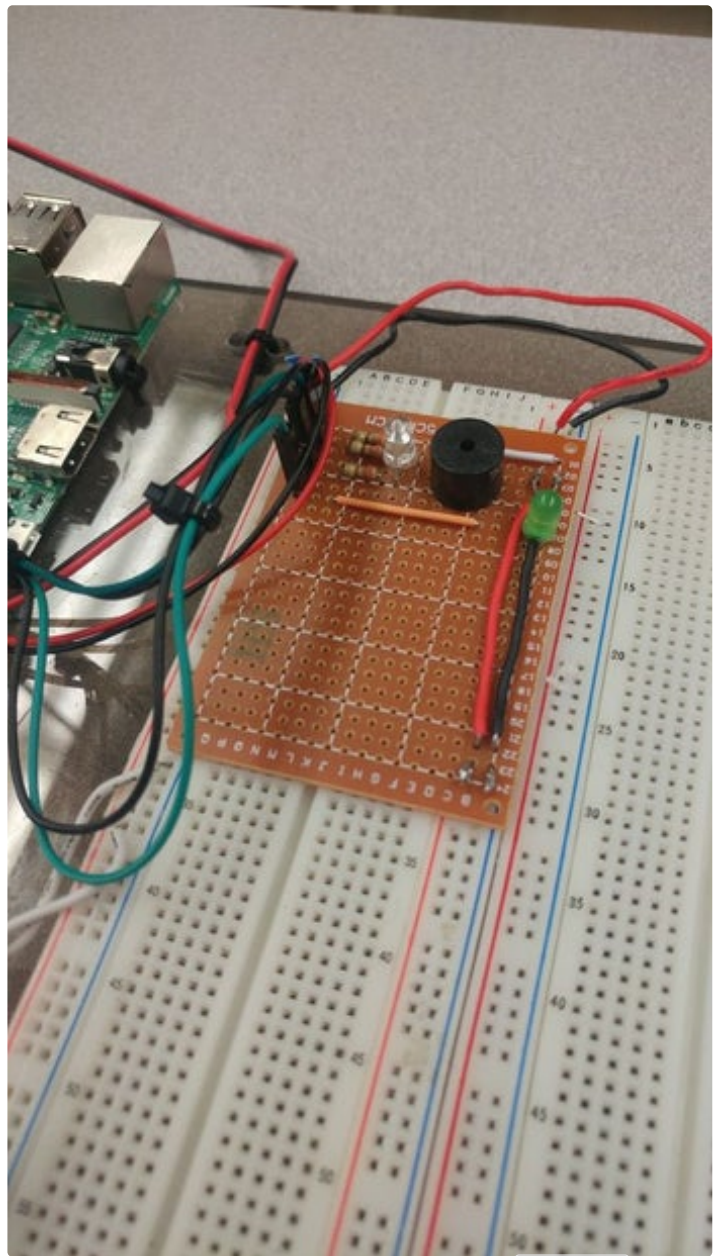
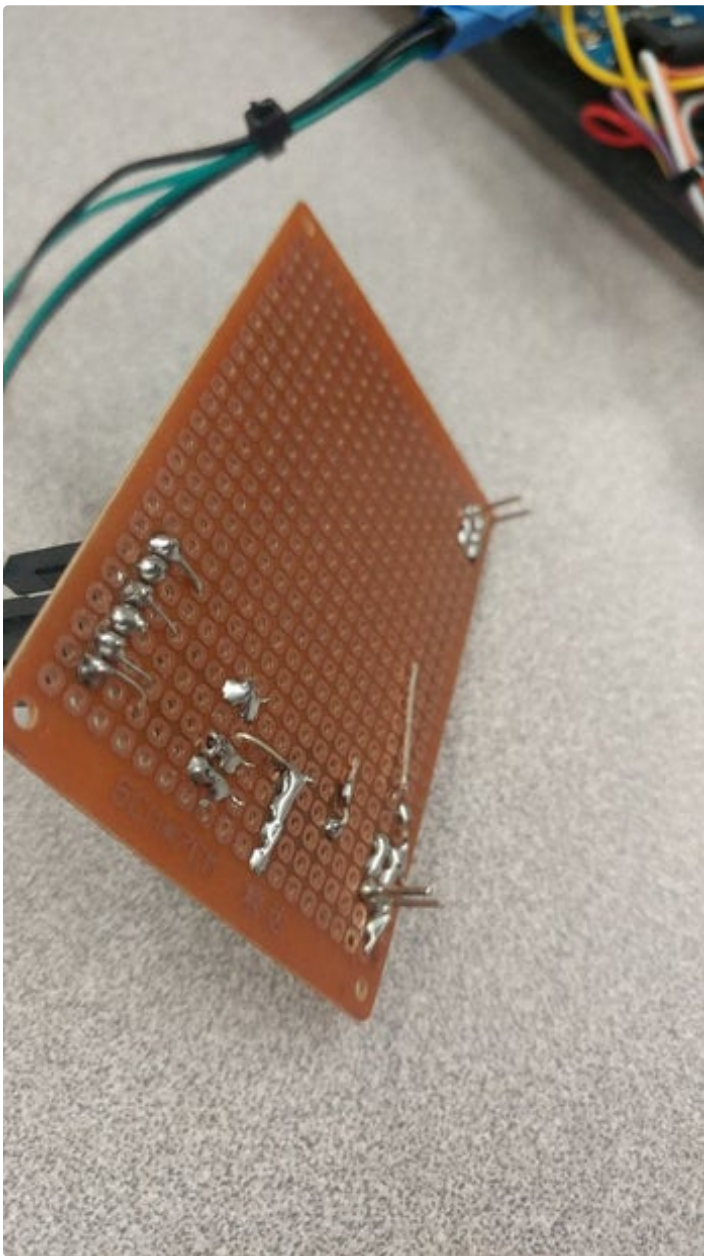
self-explanatory.

At this point you can use the power module to power the servo separately; especially if there is a high chance that the servo will reach stall current.

If you want, you can mount the buzzer and LED onto a perf board, as shown above. This is useful because it will later serve as the breakout board for the servos, and is neater overall.

Try to input a value greater than 180 degrees for the servo- see what happens!





<https://www.instructabl...>

Download



<https://www.instructabl...>

Download

Step 4: Completed Product

You can now control servos with this manual interface. Some tips are mentioned here.

Troubleshooting:

-LCD doesn't work: check all your connections, and that you have initialized the LCD with the correct number of rows and cols

-Keypad doesn't work: check connections

-Keypad works, but wrong numbers are printed: You have two alternatives here. You can either re-label the matrix in the code (i.e. if you are printing A instead of 1, relabel 'A' to '1'), or you can redo your wiring, taking special care to ensure that all the wires go to the right pins on the Arduino.

-LED doesn't work: Check to make sure you're using a common cathode (common ground) LED. If not, you can invert all the signals you send to it (i.e. change HIGH to LOW) and connect the common anode to +5V.

We will soon be posting a short Youtube video demonstrating the features of this device; as well, as a whole series on the construction of our Modified InMoov Robot. You can read more about the original InMoov [here](#). If this Instructable is entered in a contest, please vote for it! Anything we win will help us speed up the project and, consequently, additional Instructables on this topic.

This Instructable was last edited on Oct 5, 2019.

