Crickit Dancing Marionette Kit

Created by Dano Wall

https://learn.adafruit.com/universal-marionette-with-crickit

Last updated on 2023-08-29 04:01:17 PM EDT
Overview

This project demonstrates how to build a robotic marionette that is controlled with four arcade-style buttons.

The cardboard rig can be configured to control and animate any marionette, and draws inspiration from this Pinocchio marionette.

The Electronics

This project is based around the CRICKIT robotics platform. Four buttons and four small servo motors are connected through CRICKIT to create a rig capable of animating any marionette you choose to use.

Adafruit CRICKIT for Circuit Playground Express

Sometimes we wonder if robotics engineers ever watch movies. If they did, they'd know that making robots into servants always ends up in a robot rebellion. Why even go down that...

https://www.adafruit.com/product/3093
1 x USB cable - USB A to Micro-B
3 foot long

https://www.adafruit.com/product/592

4 x Sub-micro Servo
SG51R

https://www.adafruit.com/product/2201

1 x Arcade Button Quick-Connect Wire Pairs
0.11" (10 pack)

https://www.adafruit.com/product/1152

1 x Arcade Button - Red
30mm Translucent

https://www.adafruit.com/product/473

1 x Arcade Button - Yellow
30mm Translucent

https://www.adafruit.com/product/474

1 x Arcade Button - Green
30mm Translucent

https://www.adafruit.com/product/475

1 x Arcade Button - Blue
30mm Translucent

https://www.adafruit.com/product/476

1 x Premium Male-Male Jumper Wires
20x12" (300mm)


1 x 5V Power Supply
110/240V input, 5V output to 2A

https://www.adafruit.com/product/276

Other Materials

In addition to the electronics, for this project you will need:

- A large cardboard box
- Scrap corrugated cardboard
- Paper clips
- Scissors and/or box cutter
- Hot glue gun
- Hot glue sticks
- String (or thin fishing line)

CircuitPython Code

Getting to know CircuitPython

[CircuitPython](https://circuitpython.org) is a programming language based on Python, one of the fastest growing programming languages in the world. It is specifically designed to simplify experimenting and learning to code on low-cost microcontroller boards.

CircuitPython is easiest to use within the Mu Editor. If you haven't previously used Mu, [this guide will get you started](https://learn.adafruit.com/getting-started-with-circuitpython/mu-editor).

If you've never used Circuit Playground Express with CRICKIT before, make sure you've updated it with the latest special 'seesaw' version of the CPX firmware. [This guide will show you how](https://learn.adafruit.com/updating-circuit-playground-express-with-cricket).

Uploading

Make sure you've connected the Circuit Playground Express to your computer (mac/PC/Linux) via a known good USB A to micro-B cable. Your board should show up as a flash disk drive named CIRCUITPY (If you see a disk name CPLAYBOOT, try pressing the reset button again. If the only drive name you get is named CPLAYBOOT, CircuitPython may not be loaded on the board. You can load CircuitPython as [per this guide](https://learn.adafruit.com/uploading-circuitpython-to-circuit-playground-express)).
Once your board is connected, copy code.py from the window below and paste it into Mu. Press the Save button and your code should automatically be saved to the CIRCUITPY disk drive (which appears when the Circuit Playground Express is plugged into your computer) as code.py.

```python
import time
from adafruit_crickit import crickit

# For signal control, we'll chat directly with seesaw, use 'ss' to shorted typing!
ss = crickit.seesaw

# four buttons with pullups, connect to ground to activate
BUTTON_1 = crickit.SIGNAL1  # button #1 connected to signal port 1 & ground
BUTTON_2 = crickit.SIGNAL2  # button #2 connected to signal port 2 & ground
BUTTON_3 = crickit.SIGNAL3  # button #3 connected to signal port 3 & ground
BUTTON_4 = crickit.SIGNAL4  # button #4 connected to signal port 4 & ground

ss.pin_mode(BUTTON_1, ss.INPUT_PULLUP)  # Set as input with a pullup resistor
ss.pin_mode(BUTTON_2, ss.INPUT_PULLUP)
ss.pin_mode(BUTTON_3, ss.INPUT_PULLUP)
ss.pin_mode(BUTTON_4, ss.INPUT_PULLUP)

while True:
    if not ss.digital_read(BUTTON_1):
        print("Button 1 pressed")
        crickit.servo_1.angle = 40
        time.sleep(0.1)
    else:
        crickit.servo_1.angle = 140

    if not ss.digital_read(BUTTON_2):
        print("Button 2 pressed")
        crickit.servo_2.angle = 140
        time.sleep(0.1)
    else:
        crickit.servo_2.angle = 40

    if not ss.digital_read(BUTTON_3):
        print("Button 3 pressed")
        crickit.servo_3.angle = 40
        time.sleep(0.1)
    else:
        crickit.servo_3.angle = 140

    if not ss.digital_read(BUTTON_4):
        print("Button 4 pressed")
        crickit.servo_4.angle = 140
        time.sleep(0.1)
    else:
        crickit.servo_4.angle = 40
```

©Adafruit Industries
Troubleshooting

Problem: My Circuit Playground Express isn't recognized by Mu!

Solution: Make sure your board is set up with CircuitPython, which has the Circuit Playground Express show up as a flash drive named CIRCUITPY when you connect the CPX to your computer. If it is showing up as CPLAYBOOT on your computer, you can follow the steps in this guide to ensure CircuitPython is loaded and you see the CIRCUITPY drive.

Problem: My buttons don't work!

Solution: Make sure you've updated the CircuitPython firmware for CRICKIT from the CRICKIT guide.

Problem: Nothing is moving!

Solution: Check that your Circuit Playground Express is connected to a 5V power supply and the small slide switch on CRICKIT is set to "ON". Be sure the lightest servo wire faces away from the CRICKIT board, the darkest (black.brown) is closest to the center of CRICKIT.

Dry Run

Before going any further, it's best practice to get your buttons and motors connected and make sure they work as expected.
Quick-Connect

Connect the quick-connect wire pairs () to the button terminals, and plug the male/male jumper wires into the JST connector at the other end.
Connect Buttons
Use **male/male jumper wires** to connect buttons to the Signal ports 1, 2, 3, and 4.

Extend the Servo Arm
In the package with the servo are several plastic horns that can screw onto the servo. Select the one with two arms in a straight line.

Extend the reach of the servo arm by taping a bamboo skewer to it.

Plug each servo motor into the four servo ports on Crickit.
Optional 90° bend
For the arms that will connect to the legs, it may help to add a 90 degree bend, thereby extending the reach outward and forward at the same time.

Use hot glue to brace bends in wood skewers, then use tape and hot glue to affix the wooden armature to the servo motor hub.

Connect 5V Power

Now you should be able to plug in your Crickit and check that pressing each button results in movement from its corresponding servo motor.
The Build

Cardboard Marionette

This construction technique uses scrap pieces of cardboard, folded into open segments and linked together with paper clips.

You will need:

- Scrap cardboard
- Scissors
- Hot glue gun
- Paper clips

WARNING! BURN HAZARD! Hot glue is very sticky and it's easy to accidentally burn your fingertips. Be careful when using it.
Cardboard Limbs

Cut a rectangle of cardboard approximately 2”x3” and fold into thirds. The leg segments you may want to size slightly larger.

Glue small squares of cardboard to the ends of the folded rectangle.

Make 8 of these segments (2 for each arm and 2 for each leg).
Link Segments

Use something sharp to poke a hole in the end of each segment.

Insert a paperclip into this hole in the cardboard.
Push paper clips most of the way in and use dabs of hot glue to hold paper clips in place.

Allow a few minutes to cool before moving the linked segments around.
Torso

Cut a larger rectangle of cardboard and fold two flaps along slightly angled lines.

Glue cardboard strips along the top and bottom to hold the folded flaps in place.

Connect the arm linkages to the torso using the same paperclip technique as before.
Legs

As with the arms, build two leg linkages, connecting each with paper clips and gluing the paper clips in place with hot glue.
Hands and Feet

Connect the legs to the torso using paper clips.

Glue some cardboard hands and feet onto the model.
The Head

Make a head out of a folded strip of cardboard.

Insert a paper clip into the bottom of the head and top of the torso to link them together.

A small hole poked in the top of the head will allow for a string to be threaded through which will be used to support the marionette.
Add Strings

Use thin wire or string to connect limbs to servo extensions

Your marionette should now be ready to dance!
Connect the Electronics

Servo Platform

Make a small platform by introducing two 90° bends into each end of a strip of cardboard.

Glue this to the top of your box. This will allow you to stack two pairs of servos above the marionette.
Stacked Servos

Use dabs of hot glue to mount the servos on top of box.

IMPORTANT: Position the motors so that the arms are able to move freely without contacting each other.
Attach CRICKIT

Use some double sided tape (or hot glue if you prefer) to attach CRICKIT to the top edge of the box.
Buttons

Use a pencil to sketch the size and location of the buttons.

Cut holes for buttons using a hobby knife - it's better to make the holes on the small side at first and only widen them if you need to.
Once the buttons are in place, reconnect the quick-connect cables to each button.

Dance!

Your marionette is now ready to learn some dance moves!
Coding with MakeCode

If you'd like to use MakeCode for this project rather than CircuitPython, follow the instructions on this page.

Note that the circuit used here is slightly modified to work with MakeCode.

The current iteration of this code is stable in MakeCode/beta. Make sure your URL says makecode.adafruit.com/beta#editor

Getting Familiar

Microsoft MakeCode for Adafruit is a web-based code editor that provides a block editor, similar to Scratch or Code.org, and also a JavaScript editor for more advanced users.

If you'd like to learn more about MakeCode, [this guide is a good place to start](https://makecode.adafruit.com/beta#editor).
How to Upload Code

To upload code, connect your Circuit Playground Express to your computer using a micro USB cable and press the small reset button in the center of the board. All the LEDs will flash red briefly, then turn green. Your computer should now show removable drive called CPLAYBOOT.

Click the Download button to download the .uf2 file to your computer, and drag 'n drop that .UF2 file onto the CPLAYBOOT drive in your computer's file explorer or finder.

The drive will automatically eject itself (your computer may give you a "failed to eject drive correctly" error, you can ignore this). The code is now on your Circuit Playground Express and ready to run!

MakeCode

Click the button below to enter the portal to interact with the code for this project.
The Logic

The code for this project employs a tricky bit of logic to get the buttons to respond as they do.

The `not_` function returns true if the input is false, and false when the input is true. In this way, the servo motors are instructed to maintain the specified angle until a signal from the button is detected.

A single `forever` loop is used to link each of the 4 input buttons to the 4 servo motors that will animate the limbs of the marionette.

Depending on your setup, you may want to tweak the angles that each servo moves by changing the values in the `crickit set servo # angle to 40`. Keep in mind that the servos are placed in a mirror arrangement, so increasing the angle range for one will require decreasing that range for its mirror opposite. Some trial & error to sort this out is normal.
Once the code is uploaded to CPX you're ready to start putting the circuit together.

Signals & Servos

We will be utilizing 4 out of the 8 possible Signal I/O ports on CRICKIT to connect the buttons, and using up all 4 servo ports to plug in each servo.

For a deeper dive into using input and output signals with CRICKIT, reference [this guide](#).
Circuit Diagram

To use MakeCode for this project, we need to add some 1K pull up resistors to our circuit. These will allow the state of the buttons to be detected by the program.

If you'd prefer not to use a breadboard, resistors can be stuck directly into the signals port on the Crickit.

The resistors should connect the Signals line with the 3.3V line.