LO-LA59 Droid

Created by Ruiz Brothers

https://learn.adafruit.com/lola-droid

Last updated on 2023-11-18 02:53:32 PM EST
# Table of Contents

## Overview
- Animatronic Wings
- Mount with Magnets
- ItsyBitsy RP2040
- Parts

## Circuit Diagram
- Wired Connections

## CircuitPython
- CircuitPython Quickstart
- Safe Mode
- Flash Resetting UF2

## Code
- Upload the Code and Libraries to the ItsyBitsy

## 3D Printing
- Parts List
- Slice with settings for PLA material
- Supports
- Build Plate Adhesion
- Painting

## Assemble
- Solder Boards
- Connectors
- Mount Magnet
- Mount boards
- Mount Servo to Center Body
- Servo Horns
- Thread wires
- Align Bottom and Center Body
- Connects
- Attach Center to Bottom Body
- Wing Assemble
- Tail Assemble
- Connect NeoPixels
- Attach Top Body
- Attach diffuser domes
- Complete
Overview

Build Lola, the companion droid to young Princess Leia.

This cute shoulder bot emotes by flapping her wings and blinking both LED eyes. Small and light enough to carry around on your next mission!

Embedded magnets allow Lola to attach to your shoulder or metal surfaces!
Animatronic Wings

Two micro servos articulate each wing by pushing under the panel. This makes it simple to assemble and less likely to strip the servos.

Mount with Magnets

The bottom of the body feature spots for press fitting magnets to attach to legs, shoulders or metal objects!

ItsyBitsy RP2040

Powered by the ItsyBitsy RP2040. This dev board is packed with lots of great features and has the ability to use both Arduino and CircuitPython.
Parts

Adafruit ItsyBitsy RP2040
A new chip means a new ItsyBitsy, and the Raspberry Pi RP2040 is no exception. When we saw this chip we thought "this chip is going to be awesome when we give it the ItsyBitsy..."
https://www.adafruit.com/product/4888
Adafruit LiIon/LiPoly Backpack Add-On for Pro Trinket/ItsyBitsy
If you have an ItsyBitsy or Pro Trinket you probably know it's the perfect little size for a portable project. This LiPoly backpack makes it really easy to do! Instead of wiring 2...
https://www.adafruit.com/product/2124

Micro servo
Tiny little servo can rotate approximately 180 degrees (90 in each direction) and works just like the standard kinds you're used to but smaller. You can use any servo...
https://www.adafruit.com/product/169

NeoPixel Jewel - 7 x 5050 RGB LED with Integrated Drivers
Be the belle of the ball with the NeoPixel Jewel! We fit seven of our tiny 5050 (5mm x 5mm) smart RGB LEDs onto a beautiful, round PCB with mounting holes and a...
https://www.adafruit.com/product/2226

NeoPixel Mini Button PCB - Pack of 5
These are the smallest NeoPixel breakouts around! Tiny, bright RGB pixels to your project. These little PCBs are only 8mm x 10mm and have two sets of three pads on the back for...
https://www.adafruit.com/product/1612
**USB Cable**
USB C to Micro B - 1 ft / 0.3 meter  
https://www.adafruit.com/product/3879

**Pin Back**
Magnetic  
https://www.adafruit.com/product/1170

**Slide Switch**
Breadboard-friendly SPDT Switch  
https://www.adafruit.com/product/805

**Male Header**
36-pin 0.1" Short Break-away, Pack of 10  
https://www.adafruit.com/product/3009

**Female Header Set**
Header Kit for Feather - 12-pin and 16-pin Female Header Set  
https://www.adafruit.com/product/2886

**Spudger Tool**
Double Sided  
https://www.adafruit.com/product/3434

---

**Lithium Ion Battery - 3.7V 2000mAh**
Lithium-ion polymer (also known as 'lipo' or 'lipoly') batteries are thin, light, and powerful. The output ranges from 4.2V when completely charged to 3.7V. This...  
Circuit Diagram

The wiring diagram below provides a visual reference for connecting the components. It is not true to scale, it is just meant to be used as reference. This diagrams was created using the Fritzing software package. 

Take a moment to review the components in the circuit diagram. This illustration is meant for referencing wired connections - the length of wire, position and size of components are not exact.

Wires are measured and cut to have enough slack to reach each component.

Silicone ribbon wire is used to make them easier to coil and manage each wire inside the tight enclosure space.

Wired Connections

The two micro-servos share voltage, ground and signal connections.

NeoPixel to ItsyBisty

- DIN from NeoPixel to pin 5 on ItsyBitsy
- 5V from NeoPixel to VHI on ItsyBitsy
- GND from NeoPixel to GND on ItsyBitsy
Servo

- VCC wire to VHI on ItsyBitsy
- GND wire to GND on ItsyBitsy
- Signal wire to pin A2 on ItsyBitsy

Lipoly Backpack

- Bat on Backpack to Bat on ItsyBitsy
- G on Backpack to G on ItsyBitsy
- 5V on Backpack to USB on ItsyBitsy

Slide Switch

- Pin 1 to switch pin on Lipoly Backpack
- Pin 2 to switch pin on Lipoly Backpack

CircuitPython

CircuitPython is a derivative of MicroPython designed to simplify experimentation and education on low-cost microcontrollers. It makes it easier than ever to get prototyping by requiring no upfront desktop software downloads. Simply copy and edit files on the CIRCUITPY drive to iterate.

CircuitPython Quickstart

Follow this step-by-step to quickly get CircuitPython running on your board.

Download the latest version of CircuitPython for this board
Click the link above to download the latest CircuitPython UF2 file.

Save it wherever is convenient for you.

To enter the bootloader, hold down the BOOT/BOOTSEL button (highlighted in red above), and while continuing to hold it (don't let go!), press and release the reset button (highlighted in blue above). Continue to hold the BOOT/BOOTSEL button until the RPI-RP2 drive appears!

If the drive does not appear, release all the buttons, and then repeat the process above.

You can also start with your board unplugged from USB, press and hold the BOOTSEL button (highlighted in red above), continue to hold it while plugging it into USB, and wait for the drive to appear before releasing the button.

A lot of people end up using charge-only USB cables and it is very frustrating! Make sure you have a USB cable you know is good for data sync.
You will see a new disk drive appear called RPI-RP2.

Drag the adafruit_circuitpython_etc.uf2 file to RPI-RP2.

The RPI-RP2 drive will disappear and a new disk drive called CIRCUITPY will appear.

That's it, you're done! :)

Safe Mode

You want to edit your code.py or modify the files on your CIRCUITPY drive, but find that you can't. Perhaps your board has gotten into a state where CIRCUITPY is read-only. You may have turned off the CIRCUITPY drive altogether. Whatever the reason, safe mode can help.
Safe mode in CircuitPython does not run any user code on startup, and disables auto-reload. This means a few things. First, safe mode bypasses any code in boot.py (where you can set CIRCUITPY read-only or turn it off completely). Second, it does not run the code in code.py. And finally, it does not automatically soft-reload when data is written to the CIRCUITPY drive.

Therefore, whatever you may have done to put your board in a non-interactive state, safe mode gives you the opportunity to correct it without losing all of the data on the CIRCUITPY drive.

Entering Safe Mode

To enter safe mode when using CircuitPython, plug in your board or hit reset (highlighted in red above). Immediately after the board starts up or resets, it waits 1000ms. On some boards, the onboard status LED (highlighted in green above) will blink yellow during that time. If you press reset during that 1000ms, the board will start up in safe mode. It can be difficult to react to the yellow LED, so you may want to think of it simply as a slow double click of the reset button. (Remember, a fast double click of reset enters the bootloader.)

In Safe Mode

If you successfully enter safe mode on CircuitPython, the LED will intermittently blink yellow three times.

If you connect to the serial console, you'll find the following message.

```
Auto-reload is off.
Running in safe mode! Not running saved code.
CircuitPython is in safe mode because you pressed the reset button during boot. Press again to exit safe mode.
Press any key to enter the REPL. Use CTRL-D to reload.
```

You can now edit the contents of the CIRCUITPY drive. Remember, your code will not run until you press the reset button, or unplug and plug in your board, to get out of safe mode.

Flash Reseting UF2

If your board ever gets into a really weird state and doesn't even show up as a disk drive when installing CircuitPython, try loading this 'nuke' UF2 which will do a 'deep
clean' on your Flash Memory. You will lose all the files on the board, but at least you'll be able to revive it! After loading this UF2, follow the steps above to re-install CircuitPython.

Download flash erasing "nuke" UF2

---

# Code

Once you've finished setting up your ItsyBitsy with CircuitPython, you can access the code and necessary libraries by downloading the Project Bundle.

To do this, click on the Download Project Bundle button in the window below. It will download as a zipped folder.

```python
# SPDX-FileCopyrightText: 2021 Phil Burgess for Adafruit Industries
#
# SPDX-License-Identifier: MIT

"""
Lola Droid Shoulder Robot with servo and NeoPixel
"""

# pylint: disable=import-error
import time
import random
import board
import pwmio
import neopixel
from adafruit_motor import servo
from adafruit_led_animation.animation.comet import Comet
from adafruit_led_animation.animation.SparklePulse import SparklePulse
from adafruit_led_animation.sequence import AnimationSequence
from adafruit_led_animation.color import RED, BLUE

PIXEL_PIN = board.D5
SERVO_PIN = board.A2
NUM_PIXELS = 12
ORDER = neopixel.GRB
BRIGHTNESS = 0.1
```
# Initialize servo
PWM = pwmio.PWMOut(SERVO_PIN, frequency=50)
SERVO = servo.Servo(PWM)

# Initialize NeoPixels and animations
PIXELS = neopixel.NeoPixel(PIXEL_PIN, NUM_PIXELS, auto_write=False, 
    pixel_order=ORDER)
LARSON = Comet(PIXELS, bounce=True, speed=0.6/NUM_PIXELS, 
    tail_length=NUM_PIXELS//2, 
    color=(RED[0] * BRIGHTNESS,  # This is a little faster than 
    RED[1] * BRIGHTNESS,  # using the NeoPixel brightness 
    RED[2] * BRIGHTNESS))  # setting.
SPARKLE = SparklePulse(PIXELS, period=2, speed=0.15, 
    max_intensity=BRIGHTNESS, color=BLUE)
ANIMATIONS = AnimationSequence(LARSON, SPARKLE, advance_interval=7, 
    auto_clear=False)

SERVO.angle = POSITION = NEXT_POSITION = 7
MOVING = False  # Initial state = paused
START_TIME = time.monotonic()  # Initial time
DURATION = 1.0  # Hold initial position for 1 sec

while True:  # Loop forever...
    # Move turret -- randomly looks around and pauses
    NOW = time.monotonic()
    ELAPSED = NOW - START_TIME  # Seconds since start of motion or pause
    if ELAPSED >= DURATION:  # End motion/pause?
        MOVING = not MOVING  # Toggle between those two states
        START_TIME = NOW  # and record the new starting time
        ELAPSED = 0.0
    if MOVING:
        if ELAPSED < DURATION:
            POSITION = NEXT_POSITION
            while abs(POSITION - NEXT_POSITION) < 10:  # Min +/- 10 degrees
                NEXT_POSITION = random.uniform(0, 90)  # Try, try again
            DURATION = 0.2 + 0.6 * abs(POSITION - NEXT_POSITION) / 80
        else:
            SERVO.angle = NEXT_POSITION  # Move to end of sweep
            DURATION = random.uniform(0.5, 2.5)  # Pause time
            if MOVING:
                FRACTION = ELAPSED / DURATION  # Linear 0 to 1
                FRACTION = (3 * FRACTION ** 2) - (2 * FRACTION ** 3)  # Ease in/out
                SERVO.angle = POSITION + (NEXT_POSITION - POSITION) * FRACTION

    ANIMATIONS.animate()  # Cycle through NeoPixel animations

Upload the Code and Libraries to the ItsyBitsy

After downloading the Project Bundle, plug your board into the computer's USB port with a known good USB data+power cable. You should see a new flash drive appear in the computer's File Explorer or Finder (depending on your operating system) called CIRCUITPY. Unzip the folder and copy the following items to the PyPortal's CIRCUITPY drive.

- lib folder
- code.py
Your PyPortal CIRCUITPY drive should look like this after copying the lib folder and the code.py file.
3D Printing

Parts List
STL files for 3D printing are oriented to print "as-is" on FDM style machines. Parts are designed to 3D print without any support material. Original design source may be downloaded using the links below.

Edit Design
Download STLs
Slice with settings for PLA material

The parts were sliced using CURA using the slice settings below.

PLA filament 220c extruder
0.2 layer height
10% gyroid infill
60mm/s print speed
60c heated bed

Supports
Support Extrusion Width: .2
Support Density: 4%
Support Overhang Angle: 50
Support Z Height: .21
Interface: On
Support Roof: On
Support Pattern: Zig Zag

Build Plate Adhesion
Type: brim
Line Count: 8

Painting

Use a metallic paint to give the body a shine coat. It will take four to five coats to cover the droid.
Assemble

Solder Boards

Follow the circuit diagram to connect the two boards together.

Place the components over the mounts on the case to measure wire length.

Connectors

Use male and female headers to easy connect the two servos and two NeoPixel boards. You may use kapton tape to seal the connections.
Mount Magnet

Align the magnet pin to press fit through the cutouts on the case.

Mount boards

The ItsyBitsy, Lipo backpack and slide switch press fit into the bottom body. Align the ItsyBitsy USB port to the opening of the case.
Mount Servo to Center Body

Place the two servos with the mounting holes against the flat walls of the center body part.

Servo Horns

The left horn attaches to the servo with the corner cut out on the case. Use the included screws to secure both horns to each servo.

Thread wires

Push the two servo wires into the center hole on the case. Coil the wires to avoid cutting them short.
Align Bottom and Center Body

Rotate the Bottom Part so the USB port opening points towards the back of Lola. Orient the Center Body part so the two holes (LED eyes) face the opposite side of the USB port.

Pass the two LED wires from the underside of the Center Part to the top.

Place the battery on same side as the slide switch. The battery plugs into the JST port on the lipo backpack board.
Connects

Plug all the connectors together.

Attach Center to Bottom Body

Align the Bottom Body part clips to the center body grooves.

Use a Spudger tool () to disassemble if needed.
Wing Assemble

A metal paper clip is used to create the pin for the wings. Use Precision Flat Pliers to carefully bend the wire to fit the curvature of the wing tabs.

Use eye and protection and gloves and use caution in cutting paper clips as metal piece(s) may fly in uncertain directions.
Tail Assemble

Align the tail so the cut corner faces the eyes. This allows the tail to fully fold forward for transporting the droid.
Connect NeoPixels

Solder male headers to the NeoPixel Jewel and NeoPixel Button.

Pass the LED wires into the Top Body cut outs.

Attach Top Body

Align the clips on the top body part to the grooves on the Center Body part. Seat the first to clips into the groove then the press fit the last two clips into the grooves.
Attach diffuser domes

Connect the Neopixel boards to the wires.

Complete

Use skate lubricant to help the wings to flap smoothly.