Overview

Robotics has been tough to get into. Be it initial planning, software requirements, expensive hardware, and endless integration. We've been there, even poor Ladyada and her MIT robotics class back in the day (https://adafruit.com/BR7).

Today, robotics does not have to take extensive planning, arcane software, expensive hardware or anything else! Seriously! How is this so?

Today's modern innovations have all come together to bring robotics to the level that anyone can do sitting down in one session. Be that at your kitchen table, workbench, or in a classroom. In any of these settings, with some simple parts and any common computer, you can easily accomplish what is presented in this guide.

This may seem like a lot of pages for a "Hello World" introductory guide. We added pages for each step to ensure we cover things, so it's not a huge amount of information, it's just step by step to get you going on the right foot!

Parts List

Things you will want to have from the Adafruit shop:

1 x Adafruit CRICKIT for Circuit Playground Express
Our Creative Robotics and Interactive Construction Kit. It's an add-on to our popular Circuit Playground Express that lets you #MakeRobotFriend using CircuitPython and MakeCode

1 x Circuit Playground Express
The next step towards a perfect introduction to electronics and programming

1 x USB cable - A/MicroB - 3ft
Connecting your computer to Circuit Playground Express

1 x Micro Servo
A little servo can rotate approximately 180 degrees (90 in each direction)

Either:

1 x 3xAA Battery Pack with DC Jack
New for July, 2018, this will power projects like Crickit with 3 AA batteries.

1 x Alkaline AA batteries - 3 pack
This is a pack of 3 AA batteries.
1 x 5V 2A (2000mA) switching power supply
FCC/CE certified and UL listed power supply

Other Parts You'll Need

But feel free to improvise!

1 x drink straw (preferably), or popsicle stick, or plastic spoon

2 x bit of tape

1 x paper: plain or printed (a sample graphic is in this guide)

Tools

Medium Phillips Screwdriver (connecting Circuit Playground Express to the Crickit)

Small Phillips screwdriver (for servo horn connections)
Assembly

Only the Circuit Playground Express + Crickit combination needs assembly, the Feather and micro:bit Crickits have sockets which the microcontroller plugs into.

The Crickit HAT for Raspberry Pi does not need assembly either, it has female receptors for a male Raspberry Pi header.

The Circuit Playground Express version of Crickit comes with a package of six threaded, hexagonal brass standoffs. These will hold the Circuit Playground Express above and onto the Crickit.

Using a Philips screwdriver and the provided screws, attach the standoffs to the six large holes on the inside ring of Crickit. There are three holes near the Adafruit logo and three more near the Neopixel and speaker outputs. You do not want to put the standoffs on the holes on the outside edge of Crickit - there are 8 mounting holes there but these standoffs are needed for the Circuit Playground Express.

Tighten the screws firm but do not try to tighten excessively. A good mechanical and electrical connection is needed but excessive torque could crack a circuit board or at least make things hard to take apart later.

Once you have the six standoffs screwed into Crickit, place a Circuit Playground Express board (ID 3333, not the Circuit Playground Classic board ID 3000) onto the standoffs with the silver USB-B port of the Express pointing in the same direction as the Crickit black power jack. This will align the standoffs to the following pads:

4 o'clock: A1, "4:30": A0, 5 o'clock: VOUT
10 o'clock: SDA, "10:30": SCL, 11 o'clock: GND
Once you have the Circuit Playground Express lined up correctly, use the remaining screws to attach the boards together. Start with one screw into one standoff, say GND, leave it loose a bit, then put in the VOUT screw, loose, then the others loosely. Ensure things are lined up, then carefully tighten each screw. Again, a firm connection but not overly tight.

Watch out, the Circuit Playground Express (CPX) can go on 'backwards' and it won't work. Make sure the USB connector on the CPX is right below the DC jack, and the labels on the silkscreen of the Crickit match the ones on the CPX! See the images below!
Now the two boards should be attached to one another.

There are circular markings on the bottom of Crickit for four mounting pads (Adafruit ID 550 (https://adafruit.it/dLG)) if you would like to use the board on a surface and protect the surface and bottom of your Crickit.

If you happen to lose a standoff or screw(s), a new package is available from Adafruit:
Circuit Playground Bolt-On Kit

$3.95
IN STOCK
ADD TO CART

Little Rubber Bumper Feet - Pack of 4

$0.95
IN STOCK
ADD TO CART
Make Your Waving Arm

Making Your Servo Arm Attachment

It is much more satisfying having your first robotics project wave something rather than just seeing the servo move back and forth.

The Pole

Get one of the following items from around the house, the classroom, the store or a cafeteria:

- A drinking straw (preferred, nice & lightweight)
- A popsicle stick
- A small plastic spoon

That will be attached to the servo. Now we need something to attach to wave to us.

The Arm or Other Wavy Part

You can do one of the following things:

- Print out the Adabot head and arm in the green link below in PDF form.
- Print out any character you like on the printer. Or if you have a magazine or other material you are permitted to cut things out of, cut out your favorite character.
- On a piece of paper, use a pencil or crayon to trace your hand (fingers together) to make a hand on the paper.

Attachment to the Servo
Get some tape. Probably clear but any tape will do.

Cut out the waving part of the shape you want from the paper with scissors, if necessary.

Get your servo and one of the included horn pieces that has two arms 180 degrees apart. Using the short screw provided, connect the horn to the servo. The horn will press fit and then the screw will secure the horn. Never use one of the long screws or it could damage the servo.

Now tape your waving arm to the servo horn with some firm tape. As this will be moving, use some strong tape and go around things several times.

Or, I found using one of the long screws held my straw. The horn has some tiny holes to help, you might have to get the hole a bit bigger to get the screw in.

There you go, the servo assembly is complete! Now to hook all the rest of the parts together on the next page.
Hook It All Up!

Plug your servo wiring into the Crickit Servo block

Servo wires might vary in color a bit, but here’s the way to plug in the servo connector to Crickit: the lightest servo wire (white/yellow) goes closest to the number 1 on the outside of the board. The darkest wire (brown/black) goes in towards the center of Crickit.

Power Your Crickit

You can power Crickit two ways:

1) Use a 3x AA battery pack and 3 AA batteries

2) Use a wall power supply. 5 volts, 2 amps.
Plug the power supply female connector into the Crickit power connector.

There is a tiny off/on switch near the power connector. **Please move the switch to the ON position now and look at the tiny LEDs near the switch.**

You should see the green LED next to the smiley face lit.

If you see the yellow LED next to the warning triangle, your voltage is not good. Unplug the power to Crickit and see if it is too much voltage (more than 3 AA batteries) or a power supply that is more than 5 volts. You should not use a 6 volt, 9 volt, or higher supply, the Crickit cannot convert the voltage down.

You can use the on/off switch on the Crickit to turn the power off while working on the code. Just remember to turn the power back on when you're ready.

**Connect Circuit Playground to Your Computer**

Use a known good USB A to micro B connection cable from the Circuit Playground Express micro B connection (above the Crickit Power connector) to your computer.

Don't plug into the Crickit's USB port!
The micro-B USB connector is D-shaped and only plugs in one way. When plugging the cable in to the Circuit Playground Express, gently, with the round part down and the flat part up, wiggle the connector in.

If it doesn't seem to fit, pull apart, look, and try again. When it does fit together it may look like the cable isn't all the way in, it most likely is.

Connect USB to Your Computer

If you have a Macbook Pro or other fancy thin modern laptop, you may need a USB Type C to USB type A conversion dongle.

Most other laptop and desktop PCs have USB type A connections. USB 2 or USB 3 should be fine although USB 2 might be a good thing if you have both types of connections on your computer.

With the Circuit Playground Express end of the cable connected, connect the USB A side of the cable to the computer.

On Windows, you'll probably hear a sound as "Universal Plug & Play" detects your device and sets things up. This is all good.
Choosing Which Code to Use

There's two ways you can program your robot

You can program this project in either Microsoft MakeCode, a block (Scratch-like) language that only needs an Internet-connected web browser. You'll be downloading the code from the browser direct to the Circuit Playground Express flash memory.

or

CircuitPython, an implementation of Python for microcontrollers. You'll edit a small text file and drop it onto the Circuit Playground Express flash drive.

It is your choice and this guide explains both methods. You can jump to the appropriate page to run the software of your choice. You can always switch later, your decision is not binding!

If you are unsure, try MakeCode first and you can then consider CircuitPython afterwards.
With MakeCode, you can create robots simply and easily, using a drag-and-drop block interface. It's perfect for first-time robot-makers, people who don't have a lot of coding experience, or even programmers who just want to get something going fast.

MakeCode uses a web browser only, so no IDE is required to install. When you download a binary from MakeCode it is compiled for the Circuit Playground Express and you will overwrite any Arduino code or the CircuitPython runtime. You can always go back to programming other ways including Arduino (just use the Arduino IDE) or CircuitPython (by re-installing CircuitPython as shown here (https://adafru.it/Bfh))

Get Comfy With MakeCode

We recommend starting out by trying out the simple blinking NeoPixel example in our MakeCode guide, so you get a hang of how to install MakeCode apps on your Circuit Playground Express (https://adafru.it/wWd)

Once you feel comfortable with MakeCode, come back here and we'll add Crickit support!

Adding Crickit Extension

Now you're a MakeCode'r and you are ready to add Crickit support.

At this time, MakeCode support is being worked on and we're improving it every day, but it is Beta

For Circuit Playground Express and Feather Crickit (micro:bit is below)

Start by visiting
https://makecode.adafruit.com (https://adafru.it/Bly)

Click on New Project
In the list of blocks, select **ADVANCED** and then **EXTENSIONS**

In the **Search Bar** type in **Crickit** and click the magnifying glass.

**Click on the Crickit block** that shows up to install Crickit support!
You will now have a new CRICKIT bin of blocks you can use! Continue on to learn how to use these blocks.

For micro:bit Crickit

Start by visiting https://makecode.microbit.org/beta (https://adafru.it/Csj), be sure to use the beta version unless you see that Microsoft has made Crickit support standard in the Extensions category.

In the list of blocks, select Advanced and then Add Package.
In the Add Package... ? screen, place the following web address into the Search or enter project URL box:


Click on the Crickit block that shows up to install Crickit support!

You will now have a new CRICKIT bin of blocks you can use!
Continue on to learn how to use these blocks

Note - some of the example MakeCode block screen shots are from the Circuit Playground Express version of MakeCode. In all cases like this, there are micro:bit MakeCode equivalents. If things differ significantly, we'll show the micro:bit MakeCode separately.
Using MakeCode

Here is a simple program which will wave our servo arm:

![Code block]

This code will move the servo motor to 70 degrees, wait a second (which is 1000 milliseconds), move the servo motor to 135 degrees, wait another second, then repeat forever. This should result in a waving motion.

Be sure you use the green Servo blocks under the CRICKIT group and NOT the red Servo blocks under the PINS block group!

Adding Sound and Light Easily - The Final Code

MakeCode can play musical notes or preset short sounds. Below, a cute sound has been added when Adabot's arm waves.

The **on start** block sets the Circuit Playground lights to light blue as a background for Adabot's head. The **forever** loop waves Adabot's arm via the servo until power is removed.
And here is the MakeCode Robot in action:
CircuitPython Code

To use Crickit, we recommend CircuitPython. Python is an easy programming language to use, programming is fast, and its easy to read.

Install CPX Special Build

If you're using Circuit Playground Express (CPX), please install this special 'seesaw' version of the CPX firmware. Plug the USB cable into the CPX, double click the reset button until you see CPLAYBOOT drive, then drag the UF2 file onto the disk drive:

https://adafruit.it/Fj6

What's nice about this special version is that the adafruit_crickit, adafruit_seesaw and adafruit_motor library is built in, which saves you tons of space and makes it really fast to get started

As this version of CircuitPython already has the Crickit and Seesaw libraries "baked in", ensure that the /lib directory on your CircuitPython device (CIRCUITPY) does NOT contain the adafruit_crickit or adafruit_seesaw library as they may conflict and it could unnecessarily use additional memory.

Click the link above to download the latest UF2 file

Download and save it to your Desktop (or wherever is handy)
Plug your Circuit Playground Express into your computer using a known-good USB cable.

A lot of people end up using charge-only USB cables and it is very frustrating! So make sure you have a USB cable you know is good for data sync.

Double-click the small Reset button in the middle of the CPX, you will see all of the LEDs turn green. If they turn all red, check the USB cable, try another USB port, etc.

(If double-clicking doesn’t do it, try a single-click!)
You will see a new disk drive appear called CPLAYBOOT

Drag the .uf2 file onto it.

The CPLAYBOOT drive will disappear and a new disk drive will appear called CIRCUITPY

That's it! You're done :)
Using CircuitPython

If you would like to try out CircuitPython, you will find it rather easy to use compared to other programming environments. If you like, you can read about everything in the Adafruit Welcome to CircuitPython (https://adafruit.it/cpy-welcome) Guide.

First, since we’re going to use sound in the CircuitPython version of Hello World, hook a speaker up to the Crickit speaker terminals (to the right in the diagram below). If you have the AdaBox 008 mini-speaker, the white wires can go in either terminal, there is no red/black polarity. If you have a 4 or 8 ohm metal speaker, you’ll get better sound. The Adafruit shop has a nice selection of speakers.

Writing Code With Mu

Adafruit recommends the Mu editor to type your code into and interact with your project. You can find out how to install Mu on mac, PC, and linux in this guide page (https://adafruit.it/ANO) on the Adafruit Learning System.

The advantage to using Mu is it has an editor and added features like direct save to Circuit Playground Express and access to the Express’ REPL interactive command line and print output. Mu can even plot values for you.

The Program

The code below will use the adafruit_crickit library which provides powerful, high level access to the Crickit features.

Just like MakeCode, the goal is to move the servo to wave our arm.

Also, CircuitPython does have the ability to play real WAV audio file sounds. Adafruit has created a simple WAV file for this project, Adabot saying "Hello World"!

The goal will be to wave the flag and say Hello World.

The code for this is shown below:
Using Mu (or your text editor), copy this code and save it to your computer (wherever you save things for later, perhaps a Documents folder or a flash drive you brought to class), naming it `code.py`

Also download the Hello World WAV file named hello.wav by clicking the green button below, downloading and saving the file into the same place as your code. The `code.py` and `hello.wav` will go together in the root directory of your `CIRCUITPY` flash drive.

```python
import time
import audiioio
import board
import neopixel
from adafruit_crickit import crickit

# Set audio out on speaker
speaker = audiioio.AudioOut(board.A0)

# Start playing the file (in the background)
def play_file(wavfile):
    audio_file = open(wavfile, "rb")
    wav = audiioio.WaveFile(audio_file)
    speaker.play(wav)
    while speaker.playing:
        pass

# NeoPixels on the Circuit Playground Express Light Blue
pixels = neopixel.NeoPixel(board.NEOPIXEL, 10, brightness=0.3)
# Fill them with our favorite color "#0099FF light blue" -> 0x0099FF
# (see http://www.color-hex.com/ for more colors and find your fav!)
pixels.fill(0x0099FF)

while True:
    print("Hello world!")
    play_file("hello.wav")  # play Hello World WAV file
    crickit.servo_1.angle = 75  # Set servo angle to 75 degrees
    time.sleep(1.0)  # do nothing for a 1 second
    crickit.servo_1.angle = 135  # Set servo angle to 135 degrees
    time.sleep(1.0)  # do nothing for a 1 second
```

Installing Code On CircuitPlayground

At this point, make sure your Crickit and Circuit Playground Express are both plugged in per the Wiring It Up page (https://adafruit.it/BRa). The Crickit should have battery or wall power and a green light next to the smiley face. The round Circuit Playground Express should have a USB cable connected from its micro-B silver connector to your computer.

Now, on your desktop or laptop computer, navigate to where you saved your `code.py` and `hello.wav` files.

If you have correctly connected your Circuit Playground Express to your computer, you should have a drive named `CIRCUITPY` listed among the disk drives. That is the Circuit Playground Express flash memory.
Copy (usually by dragging the files and dropping them) to the CIRCUITY drive. First `hello.wav`, second `code.py`.

As soon as the `code.py` is detected on the CIRCUITY drive, CircuitPython will run your code. If it doesn't, you can press the Circuit Playground Express Reset button once to get it going or open the Mu REPL by pressing the REPL button and typing the Ctrl-D key.

When running, you should hear Adabot say "Hello World" once and the arm should start waving! You've made your first robotics project!!
Congratulations. With a few parts you have made your first robotics project.

The basics shown in this guide and others on the Adafruit Learning System will allow you to customize your own projects.

If you are working with CircuitPython and want to create your own WAV files, please follow this guide on ensuring the audio files are in the right format (https://adafruit.it/BvU) for Circuit Playground Express or other Adafruit microcontrollers. The audio files should not be too large (like a whole pop song) as space is limited on microcontrollers. Most sound effects and short sentences should work fine.

You can pick the colors for the NeoPixels for CircuitPython from the handy guide at https://www.w3schools.com/colors/colors__picker.asp (https://adafruit.it/BPY).