Make it Move with Crickit

Created by Anne Barela

https://learn.adafruit.com/make-it-move-with-crickit

Last updated on 2023-08-29 03:44:32 PM EDT
Table of Contents

Overview 5
• Introducing Adafruit Crickit

Motors at a Glance 8
• How do you select a motor?
• Motor Types
• Continuous DC Motor
• Why Use Them?
• Hobby Servo (a.k.a. Standard/Micro)
• Why use them?
• Continuous Rotation Servos
• Why use them?
• Stepper Motors
• Why use them?
• What about AC and Brushless DC motors?

Use a Continuous DC Motor Now 18
• Connecting a Continuous DC Motor to Crickit

MakeCode 20
• MakeCode For Continuous DC Motors
• Move a Motor Forward
• Stop and Start using Circuit Playground Buttons
• Using the Circuit Playground Slide Switch to Change Motor Direction

CircuitPython 23
• CircuitPython for Continuous DC Motors
• Move a Motor Forward
• Stop and Start using Circuit Playground Buttons
• Using the Circuit Playground Slide Switch to Change Motor Direction

Use a Standard Servo Now 25

MakeCode 27
• MakeCode for a Standard/Hobby Servo Motor
• Moving the Servo via Buttons
• Moving a Servo in a Slower, Controlled Fashion
• Example for Crickit for Circuit Playground Express and Crickit for Feather
• Example for Crickit for micro:bit

CircuitPython 30
• CircuitPython for Hobby Servo Motors
• Moving the Servo via Buttons
• Moving a Servo in a Slower, Controlled Fashion

Use a Continuous Servo Now 32
• Continuous Servos vs ‘Standard’ or ‘Hobby’ Servos
• The Servo Connections on Crickit

MakeCode 35
• MakeCode for a Continuous Servo Motor
- Running the Continuous Servo Like a Continuous DC Motor
- Running Two Continuous Servos on a Chassis

### CircuitPython

- CircuitPython for Continuous Servo Motors
- Running the Continuous Servo Like a Continuous DC Motor
- Running Two Continuous Servos on a Chassis
- Mismatched Wheel Speeds

### Use a Stepper Motor Now

### MakeCode

- MakeCode
- Move a Motor Port Stepper One Direction Forever
- Move the Motor Port Stepper Different Directions With Buttons
- Using a Stepper on the Drive Port in MakeCode
- Move a Drive Port Stepper Different Directions With Buttons

### CircuitPython

- CircuitPython for Stepper Motors
- Running the Stepper Motor Forward and Backwards
- Running a Stepper Motor Continuously
- Using the Drive Terminals for Unipolar Steppers

### Pick Your Movement Type

- Movement Types
- Spinning Movements
- Forward and backward, moving on the ground
- Pulling Up or Down
- Like water from a well, a crane or even side to side
- Back and Forth Motion
- Like windshield wipers, flapping wings, a deadbolt or other limited movement motion
- Linear Back & Forth, Push & Pull
- Precision movements for plotting or printing

### About Continuous DC Motors

- Continuous DC Motors in Detail!
- Geared Motors
- Picking a Continuous DC Motor
- Buying Guide
- Accessories

### About Standard Servo Motors

- Servo Motor (Hobby or Standard)
- Servo Sizes
- Inside of a Servo
- Horns & Accessories
- Picking a Standard Servo
- Buying Guide

### About Continuous Servo Motors

- Continuous Servo Motor
- Accessories
- Picking a Continuous Servo
- Buying Guide

©Adafruit Industries
About Stepper Motors

- Stepper Motor
- Use of a Stepper with a Belt System
- Steppers Available at Adafruit

Motors, the Next Step

- More Details on Crickit
- Details on Motor Specifications
- Motor Usage in the Adafruit Learning System
Overview

Robotics is all about movement. Just like you and I have muscles that push and pull, your robotic creation will likely have some sort of motor that will turn or spin. Part of your robot project will be to convert that turning or spinning into the movements you want. A twisting servo can become an elbow, a spinning motor can become a wheel.

You have a project in mind where something needs to move. This is the guide for you!

This Guide is in three parts:

PART 1: Getting Started Quickly with various motors and Crickit

PART 2: I want something to move like this, how do I do it?

PART 3: What are the various motors and how they work

Introducing Adafruit Crickit

The toughest thing about robotics is "Motor Control" - how to get your ideas into motion (literally!).

To make this as easy as possible, we'll be using CRICKIT. The Adafruit Crickit (a Creative Robotics & Interactive Construction Kit) allows you to connect different programmable boards to motors quickly and easily.

The Crickit is the muscle behind your project. It needs a brain like a Circuit Playground Express to plug in. Once that is done, you can hook the motor to Crickit, code your microcontroller, power it on, it moves!
Crickit comes in four types depending on the microcontroller/board you want to use:

- Adafruit Circuit Playground Express board
- Adafruit Feather board
- BBC micro:bit board
- Raspberry Pi (40 pin, all modern Pis)

Here are the four corresponding Crickit boards:

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
Adafruit CRICKIT FeatherWing for any Feather
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/3343

Adafruit CRICKIT for micro:bit
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/3928

Adafruit CRICKIT HAT for Raspberry Pi
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/3957

5V 2A (2000mA) switching power supply - UL Listed
This is an FCC/CE certified and UL listed power supply. Need a lot of 5V power? This switching supply gives a clean regulated 5V output at up to 2000mA. 110 or 240 input, so it works...
https://www.adafruit.com/product/276
USB cable - USB A to Micro-B
This here is your standard A to micro-B USB cable, for USB 1.1 or 2.0. Perfect for connecting a PC to your Metro, Feather, Raspberry Pi or other dev-board or...
https://www.adafruit.com/product/592

Motors at a Glance

How do you select a motor?

The next few pages will help you with what various types of motors can do and you can match capabilities to your project needs.

We will cover all these motors in greater detail in each section, with tips, tricks & gotcha's
For now let's take a quick tour of the different types of motors you can use!

Motor Types

There are the four types of motors that Crickit can drive:

- Continuous DC Motor (ungeared or geared)
- Standard/Hobby Servo
- Continuous Rotation Servo
- Stepper Motor

Here are the two types of motors that Crickit cannot drive:

- AC Motors
- BLDC/Brush-less DC Motors

Crickit can control different motors. Which type of motor you choose depends on the type of application you have in mind. Here is the list of motor types and an idea for when a certain motor may be appropriate for a certain type of movement.

We'll go into these in more detail but here's a rough overview:

Continuous DC Motor

Continuous DC motors can turn all the way around like a wheel. You can get them ungeared where they turn at a stunning 2000 to 6000 times a minute (RPM) or geared where they turn at only about 250 RPM.

With Crickit you can control the direction the motor turns (clockwise or counterclockwise) and the speed, from stopped to full speed.

All DC motors have two wires that are used to power and control them.
CD DVD Spindle Motor
What's this? A record player for ants?? Not at all! This is a DVD/CD Spindle Motor, that thing that's inside a CD or DVD player, that turns the disc...
https://www.adafruit.com/product/3882

Ungeared DC motors are good for fans and pinwheels and other light-weight spinning things, but they're too fast/weak for car wheels or moving something around.

DC Toy / Hobby Motor - 130 Size
These are standard '130 size' DC hobby motors. They come with a wider operating range than most toy motors: from 4.5 to 9VDC instead of 1.5-4.5V. This range makes them perfect...
https://www.adafruit.com/product/711

Geared/Gearbox DC motors have a 'gear box' a collection of gears that slow down the motor but increase its strength at the same time. They are good for car wheels, zoetropes, cam followers, or other items where you need strength but not a lot of speed

DC Gearbox Motor - "TT Motor" - 200RPM
- 3 to 6VDC
Perhaps you've been assembling a new robot friend, adding a computer for a brain and other fun personality touches. Now the time has come to let it leave the nest and fly on...
https://www.adafruit.com/product/3777
Why Use Them?

- They're really inexpensive compared to servos
- They are both strong and easy to use
- Perfect when you want something to rotate!

Hobby Servo (a.k.a. Standard/Micro)

"Hobby" Servos are also sometimes called "Standard" or "Micro" servos, are small boxy motors.

The size of the 'box' may vary but they always have three wires for power and control and the three wires are connected together into one 3-pin plug.

Servos are a lot different than DC motors because while DC motors turn all the way around, standard servos only move back and forth about 180 degrees. (Note there are 'continuous servos' but we'll discuss these later.)
With DC motors you can control speed & direction only! You can't make the DC motor move to a specific location or move just 15 degrees. It either spins or not.

With hobby servos you can control angle/location only. Since the servo doesn't turn all the way around, you can be specific about where you want it to move. But they don't spin.

Servos are great for small precision movements. For example here's two servos (one standard, one micro) that are used to control a robotic eyeball:

As you can see, you don't need the eyeball to rotate all the way around (that would be way too creepy). Instead, we can have it tilt up-down and left right by using the
two servos. By attaching them together we get a full range of 'realistic' eye movements.

Servos also tend to come with horns - these are snap on parts with little teeth that come in different shapes and sizes but are often round, X cross or I linear shaped. You can change the horns by simply removing and replacing, but using a screw with make the connection stronger.

Why use them?

- They’re the best and only way to have precision motion
- Dozens of brands are available, with different prices, strengths, and some range of sizes
- Very easy to use - set the angle you like and you’re done!
- Replaceable horns snap on and off

Continuous Rotation Servos

OK so you know how I just said that servos are good for precision back-and-forth motion only?

Well, I didn't tell you the whole story. There's also another kind of Servo called "Continuous Rotation" which does pretty much what you expect:
Continuous Rotation Micro Servo
Need to make a tiny robot? This little micro servo rotates 360 degrees fully forward or backward, instead of moving to a single position. You can use any servo code, hardware,...
https://www.adafruit.com/product/2442

Instead of moving back-and-forth with precision motion they rotate all the way around like a DC motor.

Now you might be wondering - why on earth are there two different kinds of servos?

Well, many many years ago, robotics started with just the 'standard' back and forth type. But then some makers realized that they rather liked the small square box of a servo and they modified it so it would rotate all the way around.

This makes it act a lot like a DC gearbox motor but continuous rotation servos are usually much smaller, and more expensive!

Why use them?

• They're smaller than DC gearbox motors, and lighter - for when DC gearbox motors won't fit or are too heavy
• For some small robot cars, they can act as wheel drives
• Some robotics boards don't have DC motor drivers because of the cost or complexity, they just have servo drivers.
• The enclosed box shape and horns may be easier for you to use than a DC motor!
Stepper Motors

Small Reduction Stepper Motor - 5VDC
32-Step 1/16 Gearing
This is a great first stepper motor, good for small projects and experimenting with steppers. This uni-polar motor has a built-in mounting plate with two mounting holes. There are only...
https://www.adafruit.com/product/858

Finally we come to stepper motors. Much like their name implies, they "step" along. These come with four, five or six wires.

Like DC motors, they rotate all the way around. But they do so very slowly, because of the little steps they have to take.

Like Servos, they have precision motion. But not the way servos do, where you can set a specific angle. Instead you can rotate forward and back by little steps.

For example, here's an animation of a stepper with a flag on it. As you can see it rotates around at a slow but steady rate and can move forward or backward.

What you can't see in the video is there's 200 "steps per rotation" and the flag is stepping along 200 times. From this distance it looks like a smooth rotation.
You'll see steppers a lot in precision electronic devices like CD/DVD players, 3D printers, scanners and inkjet printers.

Often times the stepper is connected to a gear or pulley that will turn the rotation of the stepper into linear motion. That linear motion is very precise.

But that precision means they're slow to rotate. So even though you could put a wheel on a stepper motor, it would be a very very slow robot car. (That said, it would be a very precise moving car, which is sometimes used with drawing robots!)

There's one catch - you can only step forward and backwards. You cannot tell the stepper to move to a particular angle like a servo. So for example you can rotate a servo 15 degrees clockwise, but you don't know where the stepper is exactly, just that it turned. Sometimes people set up limit switches or sensors to help detect the location of a stepper for this very reason.

For example, this 3D printed camera slider project uses a stepper motor.

If you tried to connect a DC gearbox motor to the pulley it would whip back and forth and the speeds would be hard to control!

The stepper moves the phone along the railing in slow and deliberate steps.

Likewise the Axidraw use two steppers to make an X-Y moving draw-bot

The movement is very very precise. So much so it can draw intricate patterns with a pen. But it's slow, and complex drawings can take an hour or more!
Why use them?

- They're very precise. Many steppers have 200 or even 500 steps per rotation and you can move one step at a time
- They can 'hold' the current location like servos, but also rotate all the way around like DC motors.

What about AC and Brushless DC motors?

There's two kinds of motors we won't cover at all here, AC motors and Brush-less DC (BLDC) Motors

**AC motors** require power from the wall, they're used for big devices like floor-standing fans, sewing machines and washing machines. These motors are powerful and need that 120 or 220 Volts from an outlet. Because they are so powerful an high voltage, you cannot control them from a Crickit like you can DC motors. Instead, look at using a relay to switch the motor on or off.

We recommend a enclosed outlet power relay like this one:

[Controllable Four Outlet Power Relay Module version 2](https://www.adafruit.com/product/2935)

Say goodbye to hazardous high voltage wiring and create the Internet of Things with safe, reliable power control....

**Brush-less DC (BLDC) motors** are often used in either high-power portable electronics like electric skateboards or drones. These motors require a more precise timing motor controller than we have on Crickit.

While there might be a future Crickit with BLDC drivers, there isn't one right now! To drive BLDC motors, check the place you purchased the motor and ask if they have a recommended driver.

Both types of motors (AC and BLDC) require a lot more care and effort to use so we recommend them for experts only!
Use a Continuous DC Motor Now

Connecting a Continuous DC Motor to Crickit

The Crickit drives up to two DC motors via the Motor connections.

Each motor can go forwards or backwards from 0 to 100%.

Let's start with a single motor. Your DC motor will have two wires.
Connect your two motor wires to the Motor 1 connectors. For now it does not matter which color wire goes in which Motor 1 slot.

The connections are the same for the Crickit for Circuit Playground Express, Crickit Wing for Feather, Crickit for micro:bit, and Crickit HAT.
Please see the following pages on how to program DC motor movement on Crickit in either Microsoft MakeCode or CircuitPython.

MakeCode

MakeCode For Continuous DC Motors

You can [learn the basics of Microsoft MakeCode here](https://www.microsoft.com/makecode/). You will need the Crickit block extensions installed. See this guide on doing so for both Circuit Playground Express and micro:bit versions of MakeCode.

The use of Motors in MakeCode for Circuit Playground Express and MakeCode for micro:bit are identical - the blocks look a bit different but the functions are 100% the same between versions.
With Crickit support in MakeCode, you should have a green CRICKIT block group available. Clicking on that group pops out various blocks for Crickit capabilities including those related to motor control as shown below.

![Crickit Block Group](image)

**Move a Motor Forward**

![Motor Forward](image)

Drag the `crickit run motor` block into the `Forever` loop

- Adjust the speed by changing the percentage.
- Motor will run forever.
- If the motor is going the wrong way, use a negative percentage.

**Stop and Start using Circuit Playground Buttons**

You can use the `on button click` event blocks to make quick interactive motor movements:
In this example above, one button makes the motor run. The other button will make the motor stop.

**Using the Circuit Playground Slide Switch to Change Motor Direction**

The **set motor inverted** block changes the motor direction without having to swap wires. Add the following blocks to utilize the slide switch to change the motor direction.
What if your motor goes backwards? If you’d like to switch the direction permanently without changing code, switch the wires around from the motor to Crickit.

CircuitPython

CircuitPython for Continuous DC Motors

You can learn the basics of CircuitPython with Circuit Playground Express here.

You will need install the special Crickit-optimized CircuitPython version, which will give you all the libraries already built in! See this guide on doing so.

With the special Crickit support in CircuitPython, it saves you tons of code and makes it really fast to get started.

For in-depth information on using DC motors with CircuitPython, see this page.

Only the Feather Crickit, Crickit HAT and the Crickit for Circuit Playground Express support CircuitPython, the micro:bit does not support CircuitPython at present. There is no Crickit support in programming the micro:bit in MicroPython which is different than CircuitPython.

Move a Motor Forward

```python
import time
from adafruit_crickit import crickit

# Create one motor on seesaw motor port #1
motor = crickit.dc_motor_1

# half speed forward
motor.throttle = 0.5

# Wait forever...
while True:
    time.sleep(1)
```

Use the code above to create a single motor on port #1

- Adjust the speed by changing the `motor.throttle` from `-1.0` (backwards full speed), to `1.0` (forward full speed)
- To stop, use `throttle = 0`
- Motor will run forever!
- If the motor is going the wrong way, use a negative number to reverse direction

Stop and Start using Circuit Playground Buttons

You can use the `adafruit_circuitplayground.express` library to make quick interactive motor projects, it's got ready-to-go button variables!

```python
from digitalio import DigitalInOut, Pull, Direction
from adafruit_crickit import crickit
import board

# Two onboard CPX buttons for input (low level saves memory)
button_a = DigitalInOut(board.BUTTON_A)
button_a.direction = Direction.INPUT
button_a.pull = Pull.DOWN

button_b = DigitalInOut(board.BUTTON_B)
button_b.direction = Direction.INPUT
button_b.pull = Pull.DOWN

# Create one motor on seesaw motor port #1
motor = crickit.dc_motor_1

while True:
    if button_a.value:
        print("Button A pressed, go!")
        motor.throttle = 1.0  # full speed!

    if button_b.value:
        print("Button B pressed, stop!")
        motor.throttle = 0    # stop!
```

In this example above, one button makes the motor run. The other button will make the motor stop.

Using the Circuit Playground Slide Switch to Change Motor Direction

Unlike MakeCode, we don't have the 'inverted' block, instead, we can keep track of the speed we want for our motor in a variable called `motor_speed`, then when we check the switch we can assign the `throttle` the positive version of that speed, or the negative of it!

```python
import time
from digitalio import DigitalInOut, Direction, Pull
from adafruit_crickit import crickit
import board

# Set up slide switch
switch = DigitalInOut(board.SLIDE_SWITCH)
switch.direction = Direction.INPUT
switch.pull = Pull.UP
```

©Adafruit Industries
# Create one motor on seesaw motor port #1
motor = crickit.dc_motor_1
motor_speed = 1.0  # full speed!

while True:
    if switch.value:
        motor.throttle = motor_speed  # positive, forwards!
    else:
        motor.throttle = - motor_speed  # negative means backwards!

    # small delay to keep from sending crickit tons of messages
    time.sleep(0.1)

What if your motor goes backwards? If you'd like to switch the direction permanently without changing code, switch the wires around from the motor to Crickit.

Use a Standard Servo Now

The Crickit can drive up to four standard (also called hobby) servo motors via the Servo connections block.
The four servo block is identical on the Crickit for Circuit Playground Express, Crickit Wing for Feather, and Crickit for micro:bit.

You can connect up to 4 servos.

Your servo will have a 3-pin connector with black or brown, red and white or yellow wires.
Connect the servo to the 3 pins next the number 1. The light colored wire should be pointing out and is closest to the number '1' marked on the Crickit, the darkest color wire pointing towards the center of Crickit.

Make sure it presses down and looks like all three wires of the servo are attached to Crickit.

Don't forget to also plug in the USB cable to your Circuit Playground Express for programming, and a battery pack or 5V wall plug to the DC jack on the Crickit.

Please see the following pages on how to program servos on Crickit in either Microsoft MakeCode or CircuitPython.

MakeCode

MakeCode for a Standard/Hobby Servo Motor

You can learn the basics of Microsoft MakeCode here.

You will need the Crickit block extensions installed. See this guide on doing so.

With Crickit support in MakeCode, you should have a green CRICKIT block group available. Clicking on that group pops out various blocks for Crickit capabilities including those related to motor control as shown below.
Moving the Servo via Buttons

We want to turn the servo horn on command from one side (0 degrees) to the other (180 degrees).

This program will move a servo to the zero degree point when the Circuit Playground Express button A is pressed. It will move the servo to 180 degrees when button B is pressed.

Depending on your servo, you may find the angle of motion isn't a full 180 degrees!

This trips up everyone the first time they use a servo - while the servos are often talked about in terms of 0 to 180 degree motion, there can be variation from servo to
servo. Some metal gear servos only move 90 degrees. Others may need longer ‘pulse lengths’ and we won't go into that there. There's nothing wrong with your servo just because it doesn't move a full 180 degrees! Visit https://learn.adafruit.com/adafruit-crickit-creative-robotic-interactive-construction-kit/makecode-servos#precise-pulses for more details on how to customize the pulse lengths!

Moving a Servo in a Slower, Controlled Fashion

The code below creates a stepped movement from 0 to 180 and back again. The timing is variable with the pause blocks. This is a common use in terms of things like wings flapping or other movements.

If you find the movement a bit jerky, you can increase the for index from 0 to 180 then take out the divide by 10 in the crickit set servo 1 angle to block. You might have to play with the pause time then if you feel the motion is too slow.

Example for Crickit for Circuit Playground Express and Crickit for Feather

![Code example for Crickit](image-url)
Example for Crickit for micro:bit

Open the micro:bit example in MakeCode

CircuitPython

CircuitPython for Hobby Servo Motors

You can learn the basics of CircuitPython with CircuitPlayground Express here.

You will need install the special Crickit-optimized CircuitPython version, which will give you all the libraries already built in! See this guide on doing so.

With the special Crickit support in CircuitPython, 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. No extra libraries required!

For in-depth information on using servo motors with CircuitPython, see this page.
Moving the Servo via Buttons

We want to turn the servo horn on command from one side (0 degrees) to the other (180 degrees).

```python
import time
from digitalio import DigitalInOut, Pull, Direction
from adafruit_crickit import crickit
import board

# Two onboard CPX buttons for input (low level saves memory)
button_a = DigitalInOut(board.BUTTON_A)
button_a.direction = Direction.INPUT
button_a.pull = Pull.DOWN

button_b = DigitalInOut(board.BUTTON_B)
button_b.direction = Direction.INPUT
button_b.pull = Pull.DOWN

# Create one servo on Crickit's servo port #1
servo = crickit.servo_1

while True:
    if button_a.value:
        print("Button A pressed!")
        servo.angle = 180  # about 180 degrees
        time.sleep(0.1)  # give crickit some time to move

    if button_b.value:
        print("Button B pressed!")
        servo.angle = 0   # about 0 degrees
        time.sleep(0.1)  # give crickit some time to move
```

This program will move a servo to the zero degree point when the Circuit Playground Express button A is pressed. It will move the servo to 180 degrees when button B is pressed.

Depending on your servo, you may find the angle of motion isn't a full 180 degrees!

This trips up everyone the first time they use a servo - while the servos are often talked about in terms of 0 to 180 degree motion, there can be variation from servo to servo. Some metal gear servos only move 90 degrees. Others may need longer 'pulse lengths' and we won't go into that there. There's nothing wrong with your servo just because it doesn't move a full 180 degrees! Visit https://learn.adafruit.com/adafruit-

Moving a Servo in a Slower, Controlled Fashion

The code below creates a stepped movement from 0 to 180 and back again. The timing is variable with short `time.sleep()` s. This is a common use in terms of things like wings flapping or other movements.

```python
import time
from adafruit_crickit import crickit

# Create one servo on seesaw servo port #1
servo = crickit.servo_1

while True:
    for angle in range(0, 180, 1):   # from 0 to 180, 1 degree at a time
        servo.angle = angle
        time.sleep(0.01)             # do nothing for a 1/100 second

    for angle in range(180, 0, -1):  # from 180 to 0, -1 deg at a time
        servo.angle = angle
        time.sleep(0.01)             # do nothing for a 1/100 second
```

You can make the code slower by increasing the number in `time.sleep()` to have more delay for every small movement. If you increase the range step from 1 to say 5 or 10 (and the same for the negative movement in the second `for` loop), you'll start to see jitter as the servo snaps to each movement creating some momentum that causes a small backwards motion when it stops.

Use a Continuous Servo Now

Continuous Servos vs 'Standard' or 'Hobby' Servos

Continuous rotation servo motors are geared motors like a regular servo but the horn can turn 360 degrees in a circle like a continuous DC motor.

Standard servos can move only to a specific angle between 0 and 180.

Continuous rotation servos can only change rotation speed, you cannot move the continuous rotation servo to a particular spot or location. They're really more like continuous DC motors than servos, but they're still called servos because the shape and wiring is so similar.
Continuous servos are usually marked continuous. But otherwise they look like their standard/hobby servo cousins. Check which one you've got as they look the same!

One easy way to check is place the horn on the servo and gently/slowly twist it by hand. If it stops, it's a standard servo. If it rotates all the way around, it's likely a continuous servo.

Control of a continuous servo in code is much like a continuous DC motor in that code tells the motor to run from 0 to 100% of full speed. Negative values indicate reversed motion.

If you use two continuous servos on a robot, often one is run the same speed as the other but one is reversed as the mounting is 180 degrees rotated.

The Servo Connections on Crickit

The Crickit can drive up to four continuous servo motors via the Servo connections block.
Connecting a continuous servo is the same for Crickit for Circuit Playground Express, Crickit Wing for Feather, and Crickit for micro:bit.

You can connect up to 4 servos at the same time.

Your continuous servo will have a 3-pin connector with black or brown, red and white or yellow wires.
Connect the servo to the 3 pins next the number 1. The light colored (white or yellow) wire should be pointing out and is closest to the number '1' marked on the Crickit, the darkest (brown or black) color wire pointing towards the center of Crickit.

Make sure it presses down and looks like all three wires of the servo are attached to Crickit.

Don't forget to also plug in the USB cable to your Circuit Playground Express for programming, and a battery pack or 5V wall plug to the DC jack on the Crickit.

Please see the following pages on how to program continuous servos on Crickit in either Microsoft MakeCode or CircuitPython.

MakeCode

MakeCode for a Continuous Servo Motor

You can learn the basics of Microsoft MakeCode here.

You will need the Crickit block extensions installed. See this guide on doing so.

With Crickit support in MakeCode, you should have a green CRICKIT block group available. Clicking on that group pops out various blocks for Crickit capabilities including those related to servo control as shown below.
For continuous servos, we will only use the middle (second) block called crickit continuous servo 1 run at 50%

Running the Continuous Servo Like a Continuous DC Motor

Here is a simple MakeCode program that will run one servo connected to Servo 1 (as shown at the top right of this page) at full speed (100%):

Running Two Continuous Servos on a Chassis

For two continuous servos back to back on a robot chassis, run them forward at full speed by running Servo 1 at 100% and Servo 2 at -100% for 2 seconds then stop:

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

CircuitPython for Continuous Servo Motors

You can [learn the basics of CircuitPython with Circuit Playground Express here](https://circuitpython.org/). With the special Crickit support in CircuitPython, it saves you tons of code and makes it really fast to get started.

For in-depth information on using servo motors with CircuitPython, see this page

Only the Feather Crickit, Crickit HAT and the Crickit for Circuit Playground Express support CircuitPython, the micro:bit does not support CircuitPython at present. There is no Crickit support in programming the micro:bit in MicroPython which is different than CircuitPython.

Running the Continuous Servo Like a Continuous DC Motor

Here is a simple CircuitPython program that will run one servo connected to Servo 1 (as shown at the top right of this page) at full speed (100%):

```python
import time
from adafruit_crickit import crickit

# Create one continuous servo on crickit servo port #1
servo = crickit.continuous_servo_1

servo.throttle = 1.0

while True:
    time.sleep(0.1)  # do nothing!
```

Running Two Continuous Servos on a Chassis

For two continuous servos back to back on a robot chassis, run them forward at full speed by running Servo 1 at 100% and Servo 2 at -100% for 2 seconds then stop:

```python
import time
from adafruit_crickit import crickit

# Create one continuous servo on crickit servo port #1
left_wheel = crickit.continuous_servo_1

# Create one continuous servo on crickit servo port #2
right_wheel = crickit.continuous_servo_2
```
while True:
    # Since the wheels are back-to-back
    # they have to have opposite rotation directions!
    left_wheel.throttle = 1.0
    right_wheel.throttle = -1.0
    time.sleep(2)  # move for 2 seconds...

    # Then stop both wheels
    left_wheel.throttle = 0
    right_wheel.throttle = 0
    time.sleep(2)  # stop for two seconds...
    # and repeat!

Mismatched Wheel Speeds

Even continuous servos from the same manufacturer and the same type may rotate at slightly different speeds when set to the exact same speed.

There is a small screw setting in the servo body, unlike standard servos (see picture below). Using a small screwdriver, carefully adjust a servo, say one set to throttle equals zero, to have the motor stopped at that speed.

Use a Stepper Motor Now

You may have one of two types of stepper motors, unipolar or bipolar. Here is how you wire them up:
A four-wire stepper indicates a bipolar stepper motor.

Wire the four wires to the Motor blocks on the Crickit in the order shown at the left.

For this type of stepper, none of the wires connect to the central Motor terminal marked GND (ground).

If you have one of the steppers with a blue plastic end and five rainbow colors, this is a unipolar stepper motor. They are readily available at reasonable prices. Be sure it is a 5 volt stepper. A common part number on the unit is 28BYJ-48 5VDC.

The data sheet shows the red wire to 5V but it can also go to GND, the pin between the two Motor groups. Be sure the colors are wired in the exact order or it will not step.

If you are connecting a unipolar stepper to the Drive port, use the wiring shown to the left. Follow the color connections carefully. On the Motor block, red was connected to GND. On the Drive block, red is connected to 5V. All is good with the wiring shown.
Wiring for the Crickit for Circuit Playground Express, Crickit Wing for Feather, Crickit HAT and Crickit for micro:bit are identical.
The Crickit drives up to two stepper motors via the Motor and/or the Drive connections.

For simplicity, if you have a choice, consider connecting your stepper motor to the Motor port connectors. Ensure the colors match the diagram above for the type of stepper you have and it's best to have a 5 volt stepper.

If you want to use CircuitPython, skip down the page to that section.

The MakeCode blocks to control a unipolar/bipolar stepper on the Motor port is DIFFERENT from the block used to control a unipolar stepper on the Drive port. Be sure you use the correct block depending on the block you are wiring the stepper motor to.

Please see the following pages on how to program stepper motors on Crickit in either Microsoft MakeCode or CircuitPython.

MakeCode

In the CRICKIT block group, scroll down until you see the Stepper heading and the block `crickit stepper move` block. Be sure not to use the `crickit drive stepper move block`, that is for using a unipolar stepper on the Drive port shown further down.

Move a Motor Port Stepper One Direction Forever

Here is a simple program that tells the stepper to move 20 steps, then waits 10 milliseconds, and repeats forever:
You'll see the motor shaft slowly turning in the "positive" direction. If you use a bit of non-clear tape on the stepper's shaft as a small flag, you can see the rotation better.

The more steps you take at once, the more jerk in the movement but you travel faster. 10 milliseconds is the lowest value to wait between movements.

The number of steps can be positive or negative, with negative going in the opposite direction from positive, re. -20 moves 20 steps in the opposite way.

Stepper motors may become warm or even hot when running continuously.

Move the Motor Port Stepper Different Directions With Buttons

An example: Moving a crane wire up and then down. This could also be any movement one way and then the opposite way. You might code this with blocks like these:
This code works as follows: When you push Button A on the Circuit Playground Express, the stepper starts to move in the positive direction. It takes 5 steps, waits 10 milliseconds, then repeats this 120 times for a total of 5 times 120 = 600 steps (3 to 6 revolutions, depending on the stepper) in a time of 120 times 50 milliseconds = 2 seconds.

If you push Button B, the same thing happens as in Button A but in reverse, because the steps taken are -5 and negative steps moves in the opposite direction.

Once you have the code running, you can change things to suit you. If 3 to 6 revolutions is too little, increase the value 120. If the speed is too slow, up the steps per movement (in positive and negative direction). If the movement is too fast, you can decrease the steps and increase the wait time.

Note that while the stepper is moving, any MakeCode music functions will stop until the stepper stops, then music can resume.
Using a Stepper on the Drive Port in MakeCode

In the CRICKIT block group, scroll down until you see the Stepper heading and the block `crickit drive stepper move` block.

Move a Drive Port Stepper One Direction Forever

Here is a simple program that tells the stepper to move 20 steps, then waits ten milliseconds, and repeats forever. 10 milliseconds should be the smallest pause between steps to ensure the motor settles before the next movement.

If you want to move the motor in the opposite direction, make the value negative, re. `-20`. The block takes positive and negative values.

Move a Drive Port Stepper Different Directions With Buttons

An example: Moving a crane wire up and then down. This could also be any movement one way and then the opposite way. You can code this with blocks like these:
CircuitPython

CircuitPython for Stepper Motors

You can learn the basics of CircuitPython with Circuit Playground Express here.

With the special Crickit support in CircuitPython it saves you tons of code and makes it really fast to get started.

For in-depth information on using stepper motors with CircuitPython, see this page.

Only the Feather Crickit, Crickit HAT and the Crickit for Circuit Playground Express support CircuitPython, the micro:bit does not support CircuitPython at present. There is no Crickit support in programming the micro:bit in MicroPython which is different than CircuitPython.
Running the Stepper Motor Forward and Backwards

Here is a simple CircuitPython program that will run one stepper connected to the Motor block (as wired at the top of this page). The stepper is moved in one direction 200 steps, wait a second, and then the opposite direction 200 steps.

10 milliseconds is the least amount of time you should use between steps. If you go much lower, the driver may miss steps. Of course the time can be longer for a slower speed.

```python
import time
from adafruit_crickit import crickit
from adafruit_motor import stepper

# Step motor forward for a second and then backward a second & repeat
while True:
    for i in range(400):
        crickit.stepper_motor.onestep(direction=stepper.FORWARD)
        time.sleep(0.010)  # minimum sleep between steps
        time.sleep(1.0)    # wait a second
    for i in range(400):
        crickit.stepper_motor.onestep(direction=stepper.BACKWARD)
        time.sleep(0.010)
        time.sleep(1.0)
```

Running a Stepper Motor Continuously

A stepper can run like a very slow continuous DC motor or continuous servo. The following code runs a stepper at maximum speed in one direction. Change `stepper.FORWARD` to `stepper.BACKWARD` to change the direction of the movement.

```python
import time
from adafruit_crickit import crickit
from adafruit_motor import stepper

# Step motor in one direction forever
while True:
    crickit.stepper_motor.onestep(direction=stepper.FORWARD)
    time.sleep(0.010)  # minimum sleep between steps
```

You will note the maximum speed, especially for the blue stepper (which is geared), is very slow!

Using the Drive Terminals for Unipolar Steppers

The Drive terminal block on Crickit can be used to control only unipolar steppers.
Here are the same two code samples as listed above. For Circuit Playground Express with Crickit, instead of using `crickit.stepper_motor`, the code uses `crickit.drive_stepper_motor`.

For a Feather with Crickit, instead of the original `crickit.stepper_motor`, use `crickit.feather_drive_stepper_motor`.

The following uses the changes for Circuit Playground Express - make the name change for Feather noted above if you have a Feather + Crickit.

```python
import time
from adafruit_crickit import crickit
from adafruit_motor import stepper

# Step motor forward for a second and then backward a second & repeat
while True:
    for i in range(400):
        crickit.drive_stepper_motor.onestep(direction=stepper.FORWARD)
        time.sleep(0.010)  # minimum sleep between steps
        time.sleep(1.0)        # wait a second
    for i in range(400):
        crickit.drive_stepper_motor.onestep(direction=stepper.BACKWARD)
        time.sleep(0.010)
        time.sleep(1.0)
```

And the continuous stepper example for the Drive port:

```python
import time
from adafruit_crickit import crickit
from adafruit_motor import stepper

# Step motor in one direction forever
while True:
    crickit.drive_stepper_motor.onestep(direction=stepper.FORWARD)
    time.sleep(0.010)  # minimum sleep between steps
```

Have fun integrating a stepper motor into your projects!

### Pick Your Movement Type

#### Movement Types

When designing your project, it would benefit you in the design process to think about how you want the project to work, then to pick the appropriate parts which meet your needs.

The following are typical types of movement you might want to make with a motor:
Spinning Movements

Forward and backward, moving on the ground

This is a very common movement. Do you want a vehicle to move around a room? An airplane propeller to turn? Let's see how to do this.

It's best to look at how we make other things move and model after that. The most common methods (in order) are:

1. Wheels
2. Tracks (wheels with connected treads like a tank and Johnny 5 in Short Circuit)
3. Friction (a crawling motion)

Most robots needing to roll use continuous DC motors for their speed at an affordable price. Continuous rotation servos can be used - we'll cover that after motors.

Go over the five points for continuous DC motors to decide what kind of motor you want to use.

For small rolling robots with wheels, spinning a platter, and other rotational movements that do not require high speed but do offer ease of use, Adafruit recommends gearbox TT motors. They are easy to mount with screws and brackets,
they offer good speed & torque, you can connect a variety of wheels, and they are low cost. The following TT motors are available from Adafruit:

- **Yellow TT Motor** - 200 rotations per minute, 3 to 6 volt DC, plastic gears
- **Blue Bi-metal Motor** - 120 rpm (at 6v), 3 to 6 volts, tougher steel gears but more noisy
- **Blue All-metal Motor** - 120 RPM (at 6v), 3 to 6 volts, toughest gears (also loud)

You pick based on your speed needs and durability. If you are spinning something that will not have anything heavy on it, yellow is fine, but spinning something heavy might be better for the blue versions. Likewise, for a flat terrain, low cost robot, yellow is fine vs. an all terrain Mars rover would benefit from the tougher gear versions.

The nice thing is these motors are similar size - you can start with one and if you find you need something more or less beefy, the others should fit if the length isn't too tight.

Adafruit carries a selection of wheels for TT motors and other shaft connectors including an adapter for lego gears which is very convenient for many projects.

To the left you can see TT motors attached to one of the Adafruit robotics chassis as an example of mounting.

Two TT Motors (yellow) attached to wheels and a chassis. A controller, such as Crickit, can be mounted on the opposite side.

---

**Pulling Up or Down**

Like water from a well, a crane or even side to side

Physics, cranes and puzzle rooms offer us ideas for getting motion in many different angles. Something moves and pulls something up or slides something. Here are some pictures of this type of movement:
For this type of motion, you will want to consider a pulley. A pulley can be driven by a motor directly or have a belt or string drive the pulley from somewhere else. It translates a rotational motion to a pulling motion.

Adafruit carries pulleys for:

- Continuous DC TT Motors: [36mm diameter](https://example.com)

Images above credit: Wikipedia CC-SA 3.0 [crane](https://example.com) and public domain [pulley](https://example.com)
• Stepper Motors: Timing Pulleys for GT2 Belts

You can also find different sized pulleys and other similar materials on the Internet. You can use dowels (round rods from craft and wood stores) to make your own pulleys. The time tested favorite is an empty thread spool affixed with something like a nail or screw.

For belts, you can use precision belts for specialized applications. But the easiest items for simple projects are usually around the house or craft store. Using rubber bands or thread are the two favorites.

You want enough slack so things will not break during use but not so loose that the mechanism slips when turning.

**Back and Forth Motion**

Like windshield wipers, flapping wings, a deadbolt or other limited movement motion

This type of motion is where Servo motors shine. Servos can move to any angle you want within a 180 degree arc.

For an excellent example of how to use a servo to create motion, look over this guide on animatronic eyes. You can see how two servos provide motion in two different directions. Or look at the animation below:

You can use a servo where you might consider a motor or pulley but the motion is not very big. Reviewing the different horns that can be attached to servos might help you visualize how the servo motion might map into the motion needed on your project.
Take, for example, a locking mechanism. You might have a servo move something that unlocks from the inside.

**Linear Back & Forth, Push & Pull**

Servos go back and forth but they rotate when they do that. If you absolutely need motion that goes back and forth but can move just a short distance (a few millimeters!) a solenoid can work.

A solenoid is an alternative to a servo for very short, straight motions. Solenoids use a coil of wire, similar to a motor but fixed, such that metal inside the coil will move in when energized.

Solenoids use a ton of power, are not very strong, and only move a few millimeters!
Adafruit sells several types of solenoids for such applications, but for use with Crickit, we only have a single solenoid that works at 5V. You can see in the animated GIF below, it doesn't move very far!

**Mini Push-Pull Solenoid - 5V**

Solenoids are basically electromagnets: they are made of a coil of copper wire with an armature (a slug of metal) in the middle. When the coil is energized, the slug is pulled into the...

https://www.adafruit.com/product/2776

---

**Precision movements for plotting or printing**

OK now're at the hard stuff. You want precision motion, like a drawing robot?

You really need to consider your project if you believe it needs a level of precision. You should ask:

- Will a servo be precise enough?
- Is a stepper motor the only way to accomplish this?

Stepper motors are generally a bit more expensive and controlling them takes different circuits than the other motors discussed. But they can travel a full 360 degrees like continuous motors and servos in very precise steps, both backward and forward.

If a stepper motor provides the motion you are looking for, you should decide how the connections will be made to the shaft. You can directly connect to the shaft, use a pulley system or use a toothed belt system.

For direct connections, you will want to use a piece known as a coupler. These are cylinders with a hole in the middle for the stepper shaft and the item you want to spin. There are tiny set screws on the side to hold things in place.

For belted connections, you will need to consider your use. Adafruit sells a nice selection of belts and pulleys adhering to the GT2 standard. GT2 has more rounded teeth and make for a very secure, reliable system.
About Continuous DC Motors

Continuous DC Motors in Detail!

This is the type of motor most people are familiar with in battery-powered toys. A motor body, usually round, drives a mechanical shaft at a certain rate of speed. If you connect something like a wheel or pulley to the motor shaft, the wheel or pulley turns.

Continuous DC motors turn when the electricity in coils makes a magnetic field which is opposed by permanent magnets in inside the case. Contacts (between the yellow and green wires in the diagram below) change the direction of the current every half cycle, ensuring the coils continue to oppose the magnets and maintains motion.
Continuous DC motors are cheap and plentiful - they're available in nearly any size and speed. Often they are customized to the exact needs of a product!

Motors have a body (the main bulk of the motor) and a shaft (the part in the center that turns). Some motors have a gear attached to power something.

The motors above most likely need a motor mount - a bracket of some sort to hold it in place. Common mounts are aluminum or plastic (3-D printing is good for mounts).
Geared Motors

Continuous motors are very fast - rotating at thousands of RPM. This is often too fast for a project, so to slow it down we use gears. The gears slow down the motor and at the same time increase its strength ("torque")

When the motor comes with several gears as a package, like the one below, it is called a Gearbox DC Motor.

Geared motors likely turn slower than the actual motor shaft. But due to the gearing action, the motor is able to provide more torque - a stronger turning force - which is great in effectively moving a load.

The yellow motor above, called a TT Motor, is a common size for hobby projects as it is powerful and affordable. TT Motors also are easy to mount due to the body shape and provided mounting holes.

As you can see on the right it uses a fully plastic gearbox. This makes it inexpensive and light.

You can also get TT motors with metal gearboxes. They look a lot like our yellow all-plastic-gearbox motors () but these have all of the motor gears machined from steel, so they won't strip as easily, and they're twice as slow (and twice as powerful) given their lower gearbox ratio. The metal gears also mean they're louder when running.
Picking a Continuous DC Motor

When looking at continuous DC motors (geared and ungeared), you will want to look at the following characteristics:

1. The body size, shape, and weight. What type of mounting points (screw holes, brackets) does it have?
2. What type of connection is on the shaft to help connect a wheel, pulley, etc. to the motor?
3. How fast does the motor shaft spin at full speed? How much torque (rotational force) does the motor have?
4. Does my application need the speed & torque that a geared DC motor provides?
5. What voltage does the motor run on and what electrical current does the motor draw when
   1. starting up
   2. running full speed
   3. stalled (if the shaft is grabbed so it cannot rotate)

Buying Guide

Here are some motors in the Adafruit Shop which have been hand-picked for various applications you may be looking for:
Perhaps you've been assembling a new robot friend, adding a computer for a brain and other fun personality touches. Now the time has come to let it leave the nest and fly on...

https://www.adafruit.com/product/3777

These durable (but affordable!) gearbox motors (also known as 'TT' motors) are an easy, low-cost way to get your projects moving. This is a TT DC Bi-Metal Gearbox...

https://www.adafruit.com/product/3801

These durable (but affordable!) gearbox motors (also known as 'TT' motors) are an easy, low-cost way to get your projects moving. This is a TT DC All-Metal Gearbox...

https://www.adafruit.com/product/3802
Continuous DC Motor (Ungeared)

CD DVD Spindle Motor
What's this? A record player for ants?? Not at all! This is a DVD/CD Spindle Motor, that thing that's inside a CD or DVD player, that turns the disc...
https://www.adafruit.com/product/3882

DC Toy / Hobby Motor - 130 Size
These are standard '130 size' DC hobby motors. They come with a wider operating range than most toy motors: from 4.5 to 9VDC instead of 1.5-4.5V. This range makes them perfect...
https://www.adafruit.com/product/711

DC Motor in Micro Servo Body
This tiny DC Motor in Micro Servo Body is an interesting motor - it's the same size and shape as our micro servo but it isn't a servo. It's...
https://www.adafruit.com/product/2941

Accessories

The main accessory you should consider is how to hook the part of your project that needs to spin to the shaft of the motor.

For larger shafts (5 mm to 10 mm), Adafruit sells [shaft couplers](https://www.adafruit.com/category/116) that fasten using set screws.
TT Motors have a specific shape to their shaft, making it easier to press fit various parts made specifically for a TT motor shaft. Parts including various types of wheels, pulleys, and shaft adapters make for flexible connections.

TT motors also easily support attachment of special disks that, when used with a sensor, can help encode the position of the shaft.

Orange and Clear TT Motor Wheel for TT DC Gearbox Motor
Usually when one needs an orange wheel it's a garnish for a cocktail, like a tasty Sidecar. And speaking of cars, this wheel is for driving, not...
https://www.adafruit.com/product/3766

Thin White Wheel for TT DC Gearbox Motors - 65mm Diameter
We're keepin' it wheel with this one! Plastic gear-box motors (also known as 'TT' motors) are an easy and low cost way to get your projects moving. But...
https://www.adafruit.com/product/3763

Skinny Wheel for TT DC Gearbox Motors
Plastic gear-box motors (also known as 'TT' motors) are an easy and low cost way to get your projects moving. But we've noticed that there are not a lot of nice TT motor...
https://www.adafruit.com/product/3757
TT Motor Pulley - 36mm Diameter
Mechanical transmission for the win, this simple plastic pulley can attach to your TT motor to transmit rotation from the motor axle to...somewhere else. This is a very...
https://www.adafruit.com/product/3789

DC Gearbox "TT" Motor to LEGO® compatible Cross Axle
Who doesn't love LEGO® compatible bricks? Nobody! Except when we accidentally step on them. There are soooo many mechanical things one can make with those ubiquitous...
https://www.adafruit.com/product/3810

TT Motor Encoder (Pack of 2)
Plastic gear-box motors (also known as 'TT' motors) are an easy and low cost way to get your projects moving. They're low cost, durable, and easy to use. The trade-off for...
https://www.adafruit.com/product/3782

Snap-on Hub for TT Motor
Sprockets, wheels, and gears, oh my! We really like our DC gear motors (a.k.a "TT Motor") but unless you have a 3D printer it can be a challenge to get something...
https://www.adafruit.com/product/3769
About Standard Servo Motors

Servo Motor (Hobby or Standard)

These specialty motors have electronics, gears, and a motor to turn what is connected to them. Unlike other motors, servos have a limited range of rotation, usually half way around a circle (180 degrees). A servo can be told at what angle to turn to so movements can be controlled by the user.

A servo with an electronic servo controller is great for moving things in a very controlled fashion. Most robot joints along with robot car steering systems use servos. They can also be used to move something in and out a short distance dependent on the type of horn and connection to the horn.

Servo Sizes

Servos come in different sizes. They are often a "standard size" although when you buy one you should check the dimensions to make sure it is the size you want. Typical names for sizes are standard and micro although other terms like nano, ultra-nano, and giant might be used.
Inside of a Servo

Inside a servo is a continuous DC motor, control electronics, a variable resistor (potentiometer) and gears for the range of motion.

On the left is a nice photo we took of the internal gear-box of a servo!

Some servos come with metal gears, which are more rugged, but they may emit a bit more noise.

The gearbox means a servo is strong, and somewhat slow. It can only move back and forth about 180 degrees. The movement might be more limited on certain servos, perhaps 170 degrees on some, only 90 degrees on another.

There are three wires on a standard servo, two for DC power and one for control. A servo requires some other electronics to tell it what angle to turn to.... like a Crickit!

Servos come with various screw on pieces, called horns, to help you attach the servo to something else, making connections easier than with Continuous DC motors.
Horns & Accessories

One of the benefits of the servo form factor in general are the number of accessories available to mount on the shaft. Nearly all servo shafts have grooves so attachments fit in without spinning. The shaft also has a hole at the end to secure attachments.

The attachment items for servos are almost always called horns. Horns can come in various shapes to allow users to make various types of connections depending on their project. Most servos you buy will come with two or three horns that fit that specific servo but often you can mix or match other horns and shops have alternatives to the ones in the package.

Horns most often have holes in them to help make attachments, often screws or wires.
There are specialty accessories for continuous rotation servos like wheels. For standard/hobby servos, a wheel would only turn 180 degrees, limiting the usefulness of a wheel.

**Continuous Rotation Servo Wheel**
Plastic wheel with a cutout specially designed to allow attachment to our larger continuous rotation servo. Makes it easy to get your...
https://www.adafruit.com/product/167

**Wheel for Micro Continuous Rotation FS90R Servo**
We're keepin' it wheel with this one! Need a great drive solution for your little robotic friends? This black plastic Micro Continuous...
https://www.adafruit.com/product/2744

As the cable of a servo has a connector on the end, there are extension cables available which extend the connection if your servo is not close to your controller board.

**Servo Extension Cable - 50cm / 19.5" long**
Stretch out your servo connections with this flexible servo extension cord. It has a 3 pin shrouded "male" connection to plug your servo into and then, 50cm later, a 3 pin...
https://www.adafruit.com/product/973
Servo Extension Cable - 30cm / 12" long -
Stretch out your servo connections with this flexible servo extension cord. It has a 3 pin shrouded "male" connection to plug your servo into and then, 30cm later, a 3 pin...
https://www.adafruit.com/product/972

Picking a Standard Servo

When looking for standard servo for your project, you will want to keep in mind these questions:

1. What range of motion are you looking for? Most standard servos rotate in about a 180 degree arc but some have a smaller range (older or specialty models).
2. What type of load are you trying to move? Cardboard and paper have different requirements than metal or other heavy loads. Look at the servo torque (rotational force) rating to see how much mechanical force the servo can handle safely. If your project needs more force, consider a larger servo or one with metal gears for more durability.
3. How will you control the servo? Most often a servo is hooked to a control circuit to tell the arm what position it should be at or travel to. This requires more circuitry compared to a continuous DC motor. Will you use a microcontroller or a special purpose robotics board like Crickit to use the servo?
4. What are the voltage and current ratings for the servo? Are they compatible with the controller?

Buying Guide

Here are some hobby servos in the Adafruit Shop which have been hand-picked for various applications you may be looking for:
Standard servo - TowerPro SG-5010
This high-torque standard servo can rotate approximately 180 degrees (90 in each direction). You can use any servo code, hardware, or library to control these servos. Good for...
https://www.adafruit.com/product/155

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

Sub-micro Servo - SG51R
This is just about the cutest, tiniest little micro servo we could find, even smaller than the 9-gram micro servos we love so much. It can rotate approximately 180 degrees (90 in...
https://www.adafruit.com/product/2201

For high torque applications:
Standard Size - High Torque - Metal Gear Servo
This high-torque standard servo now comes in a metal-gear flavor, for extra-high torque (10 kg-cm!) and reliability! It can rotate at least 120 degrees (60 in each direction) with a...
https://www.adafruit.com/product/1142

Micro Servo - High Powered, High Torque Metal Gear
Add even more power to your robot with this metal-geared servo. The tiny little servo can rotate approximately 180 degrees (~90 in each direction), and works just like the...
https://www.adafruit.com/product/2307

Micro Servo - MG90D High Torque Metal Gear
Add more power to your robot with this metal-geared MG90D servo. The tiny little servo can rotate approximately 90 degrees (45 in each direction) and works just like the standard...
https://www.adafruit.com/product/1143

Two servos in a pan & tilt configuration:
Mini Pan-Tilt Kit - Assembled with Micro Servos
This fully-assembled pan-tilt kit is the perfect way to give your project full range motion with two micro servos. The pan-tilt can rotate roughly 180° from side-to-side and...

About Continuous Servo Motors

Continuous Servo Motor

Think of a continuous servo as a hybrid of a continuous DC motor and a standard servo.

A continuous servo can turn around a full 360 degrees like a motor. A continuous servo also has three wires like a standard servo, 2 for power and one for control. The speed of a continuous servo is controllable. Due to the gears, continuous servos generally are slower than a continuous DC motor or a geared motor.
You can get servo horns in the shape of wheels to build things that roll on the ground.

Servos come in different sizes. They are often a "standard size" although when you buy one you should check the dimensions to make sure it is the size you want. Typical names for sizes are standard and micro.

Servos most often have mounting points so the body can be screwed down for stability. The rectangular shape also helps in mounting compared to round continuous DC motors.
Accessories

One of the benefits of the servo form factor in general are the number of accessories available to mount on the shaft. Nearly all servo shafts have grooves so attachments fit in without spinning. The shaft also has a hole at the end to secure attachments.

The attachment items for servos are almost always called horns. Horns can come in various shapes to allow users to make various types of connections depending on their project. Most servos you buy will come with two or three horns that fit that specific servo but often you can mix or match other horns and shops have alternatives to the ones in the package. Horns most often have holes in them to help make attachments, often screws or wires.

There are specialty accessories for continuous rotation servos like wheels. The wheels have servo style mounts and can be secured with screws.

As the cable of a servo has a connector on the end, there are extension cables available which extend the connection if your servo is not close to your controller board.

Picking a Continuous Servo

When looking for continuous rotation servos for your project, you will want to keep in mind these questions:

1. What type of load are you trying to move? Look at the servo torque (rotational force) rating to see how much mechanical force the servo can handle safely. If
your project needs more force, consider a larger servo or one with metal gears for more durability.

2. How will you control the servo? Most often a servo is hooked to a control circuit to tell the servo what position it should be at or travel to. This requires more circuitry compared to a continuous DC motor. Will you use a microcontroller or a special purpose robotics board like Crickit to use the servo?

3. What are the voltage and current ratings for the servo? Are they compatible with the controller?

Buying Guide

Here are some continuous rotation servos in the Adafruit Shop which may work for various applications you are planning:

**Continuous Rotation Servo**
This servo rotates fully forward or backward instead of moving to a position. You can use any servo code, hardware, or library to control these servos. Good for making simple moving...
https://www.adafruit.com/product/154

**Continuous Rotation Servo Wheel**
Plastic wheel with a cutout specially designed to allow attachment to our larger continuous rotation servo. Makes it easy to get your...
https://www.adafruit.com/product/167
Continuous Rotation Micro Servo
Need to make a tiny robot? This little micro servo rotates 360 degrees fully forward or backward, instead of moving to a single position. You can use any servo code, hardware,...
https://www.adafruit.com/product/2442

Wheel for Micro Continuous Rotation FS90R Servo
We're keepin' it wheel with this one! Need a great drive solution for your little robotic friends? This black plastic Micro Continuous...
https://www.adafruit.com/product/2744

About Stepper Motors

Stepper Motor

A stepper motor often looks like a continuous DC motor, coming in round or square shapes. You can tell they are different as a stepper motor has four to eight wires coming out if it. The wires inside form coils which allow the shaft of the motor to turn very precisely.

The wire connections must be made to an electrical circuit specially made for controlling a stepper motor as it will not work by putting DC power on the wires.

As shown below, pulses of electricity are sent to the coils in a timed manner to achieve motion.
Steppers can turn in a full circle like DC motors and continuous servos but they can be more expensive and require electronic controllers to achieve their precise movements. Often the shaft is connected to a belt or gear for moving something.

Examples of steppers being used are with scanners, plotters, 3-D printers, etc.
Use of a Stepper with a Belt System

A belted system provides excellent control over moving something. Most plotters, CNC machines, and 3-D Printers use some variation of belts for accuracy.

Adafruit carries the GT2 belt system. There are other systems but GT2 offers advantages noted by Ladyada in this video.

Steppers Available at Adafruit

Adafruit carries a number of steppers in various sizes. There are also couplers which help attach the stepper shaft to your project.

Note: It is recommended that you use 5 volt stepper motors. 12 volt stepper motors may not behave as expected.
Small Reduction Stepper Motor - 5VDC 32-Step 1/16 Gearing
This is a great first stepper motor, good for small projects and experimenting with steppers. This uni-polar motor has a built-in mounting plate with two mounting holes. There are only...
https://www.adafruit.com/product/858

Aluminum Flex Shaft Coupler - 5mm to 5mm
Connect this to that with a set-screw coupler. These couplers are made of machined aluminum and have a spiral cut that makes them slightly flexible so they can be fit to two shafts even...
https://www.adafruit.com/product/1175

Aluminum Flex Shaft Coupler - 5mm to 8mm
Connect this to that with a set-screw coupler. These couplers are made of machined aluminum and have a spiral cut that makes them slightly flexible so they can be fit to two shafts...
https://www.adafruit.com/product/1176

Motors, the Next Step

Now that you have seen a broad representation of motors and the basic code to use them on Crickit, what are the next steps on becoming more proficient? The resources presented on this page provide more in-depth information on various aspects of motors and Crickit.
More Details on Crickit

The Adafruit Learning System guide [Introducing Adafruit Crickit](https://learn.adafruit.com/introducing-adafruit-crickit) provides a very thorough review on what Crickit is and how it can be used in many ways. Feel free to jump to various sections, especially on motors, to find additional information beyond that presented in this guide.

That guide and other guides from Adafruit will be kept up to date on expansion of the Crickit product line and how to use Crickit and various microcontrollers.

Details on Motor Specifications

On the product page of each motor Adafruit sells, there is information specific to that motor including:

- Dimensions
- Weight
- Operating Current
- Revolutions Per Minute (varies by voltage for Continuous DC motors)
- Current (can vary from still to use to stall depending on type)
- Steps per Revolution (steppers)

Often the product page will recommend various Adafruit products to control that type of motor - this is usually for installations not using a Crickit.

Watch out! Crickit can only power 5V motors, servos and solenoids! Check that the motor you want to use can run at 5V. If its a 9V or 12V motor it *might* work but run very slowly or weakly. On the other hand, a 3V motor can burn out fast because it's not rated for 5V!

Motor Usage in the Adafruit Learning System

The [Adafruit Learning System](https://learn.adafruit.com) has project guides for using a wide array of Adafruit products. If you are considering using a type of motor, look for a guide that might use that particular motor or one like it. The code may or may not be in the language you are looking to utilize, Adafruit guides are often written for one of three languages: MakeCode, CircuitPython, or Arduino. But you can glean a lot on how a product was used in one project for building another.
An example: In the Using a Standard Servo Now page, there was demonstration code for a flapping effect. Going to https://learn.adafruit.com and doing a search for "flapping" (as "wing pulls up Feather products), the following tutorials have a wing flapping motion to look at:

- Trash Panda ()
- Crickit Powered Minerva Owl Robot ()
- Cardboard Box for Circuit Playground Express ()

All of those demonstrate a servo being used to create motion which can be adapted for other use.

It might take a couple of searches in the Adafruit Learning System to find what you are specifically looking for, there are approximately 1,500 guides at the time this was written!

Have fun with your motors!