What Will You Learn?

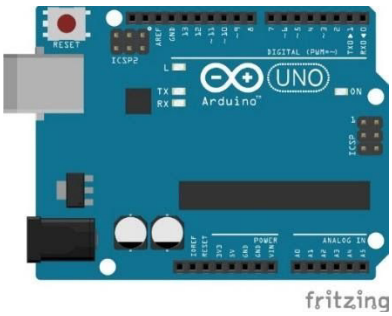
Learn to connect a Servo and upload example code that will make it move in a sweeping motion.

Why Should You Learn This?

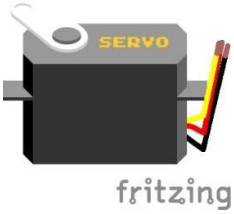
In this activity, you will use an example code created by Arduino to explore how a standard (position based) Servo can be programmed to move in a sweeping motion from side to side. You should have completed the "Servo" activity before moving on to this activity.

Here's What You'll Need:

Arduino Microcontroller



Standard (small) Servo



Red Wire

Black Wire

White Wire

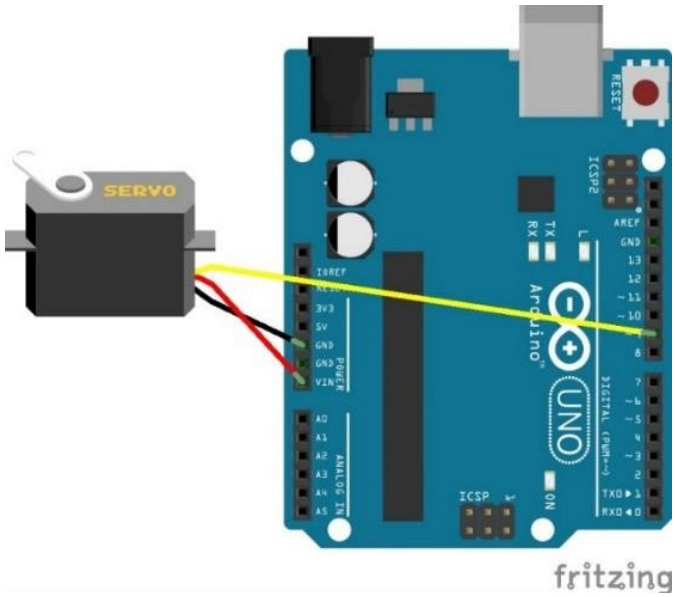
Safety First!

Never allow the red and black wires to touch together while they are connected to a powered microcontroller as that creates a short circuit. A short circuit can potentially cause the wires and/or the microcontroller to get hot enough to burn the skin. In addition, in the event of a short circuit the microcontroller can potentially catch on fire.

Let's Get Connected!

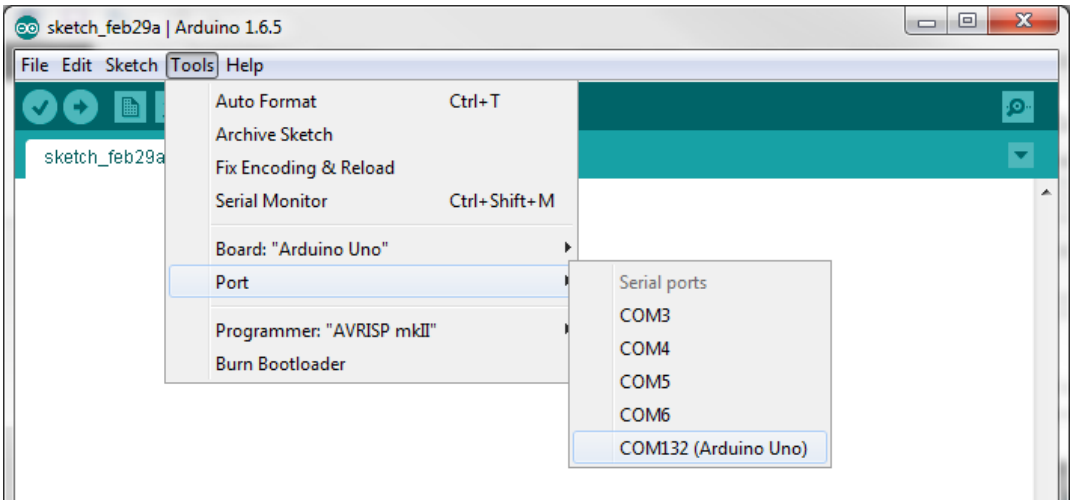
1. Connect the red and black wires to their respective places on the Servo. Connect the white wire to the yellow wire on the Servo. Note: Your wires attached to the Servo may be different colors than the ones pictured below. Remember, the color of the wire helps communicate its purpose. Red typically stands for positive (+), but sometimes manufacturers use orange instead. Black typically stands for negative (-), but sometimes brown is used. White typically stands for the control signal, but sometimes yellow is used.

Connect the red wire to Vin, the black wire to GND, and the white wire to pin 9.

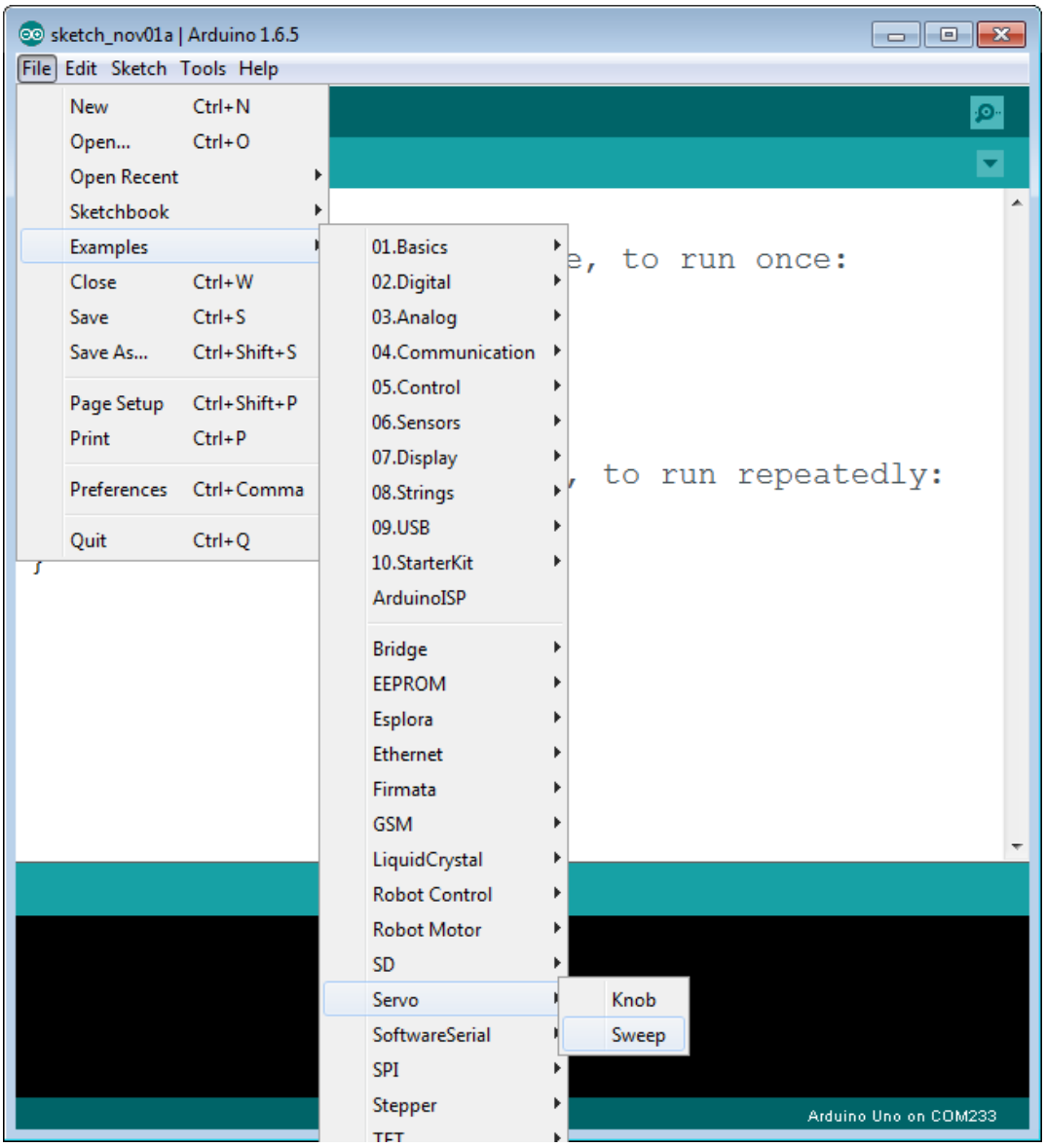


Time to Upload Some Code!

1. Start by connecting the Arduino microcontroller to the computer via the USB cable. Open the Arduino programming software and check that you're connected to the right port.



2. The motion you will use for this activity is detailed in the example “Sweep”. This is an example that is already on Arduino and you can open it and use it as your own. Go to File > Examples > Servo > Sweep.



The example code is below. Be sure to upload this code to the microcontroller to see your Servo sweep.

Sweep | Arduino 1.6.5

File Edit Sketch Tools Help

Sweep §

```
 http://www.arduino.cc/en/Tutorial/Sweep
*/

#include <Servo.h>

Servo myservo;  // create servo object to control a servo
                // twelve servo objects can be created on most boards
int pos = 0;    // variable to store the servo position

void setup()
{
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}

void loop()
{
  for(pos = 0; pos <= 180; pos += 1) // goes from 0 degrees to 180 degrees
  {                                  // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
  for(pos = 180; pos>=0; pos-=1)     // goes from 180 degrees to 0 degrees
  {
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
}
```

Focusing on the section boxed in red, this example uses a "for" statement. Once the program enters a "for" statement, it stays there until the condition within the "for" statement has been completed (which is why it is often called a "for loop").

The first statement: "`for(pos = 0; pos <= 180; pos += 1)`" uses "pos" as a variable for the position of the Servo motor. A variable is a place holder used to store information. In this example it stores the position of the Servo motor over time. The position value, stored in the variable "pos", is initially set to 0 (`int pos = 0`). The code will increase the "pos" variable by 1 (`pos += 1`) each time it runs through the `for` statement. The `for` statement is programmed to repeat until (`pos <= 180`) so the statement ends after the variable "pos" is equal to 180.

Inside the brackets of the "for" statement are the actions that will take place during each iteration, "`myservo.write(pos);`" and then "`delay(15);`". This means that the Servo will go to 1 degree, then 2 degrees, then 3 degrees, and so forth, all the way up to 180 degrees. It will pause for 15 milliseconds at each degree, which is just enough time for the Servo to move one degree, before it moves again, making it look like a continuous motion. The Servo moves 180 degrees at a rate of one degree every 15 milliseconds which means it should take it 2700 (15ms * 180 degrees) milliseconds (or 2.7 seconds) to sweep from 0 to 180 degrees.

Once the position value (the "pos" variable) reaches 180, the program moves out of this "for" statement to the next line of code, which happens to be another "for" statement. The next "for" statement, which reads that we will start at 180 and decrease the position value by 1 until we reach 0. This also results in the Servo moving to the angle defined by the variable "pos".

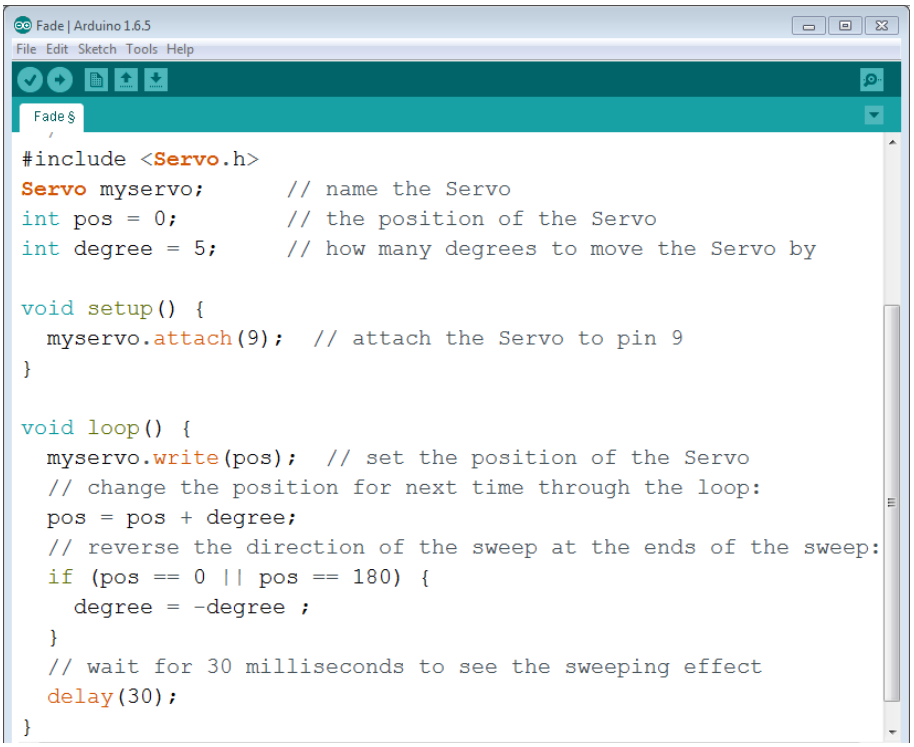
Tips for Success!

- If you're having errors uploading, check that you have written everything correctly and that you are connected to the right port.

- If your Servo isn't working, make sure all the wires are connected properly and check that the Servo isn't broken (try testing it with a different Servo).

Are You Ready for Some Challenges?

If you completed the LED Fade activity, you may have taken on the challenge of using the same type of code to control a Servo. If you haven't yet completed that challenge, go back and try it before moving on. That code may have looked something like this:



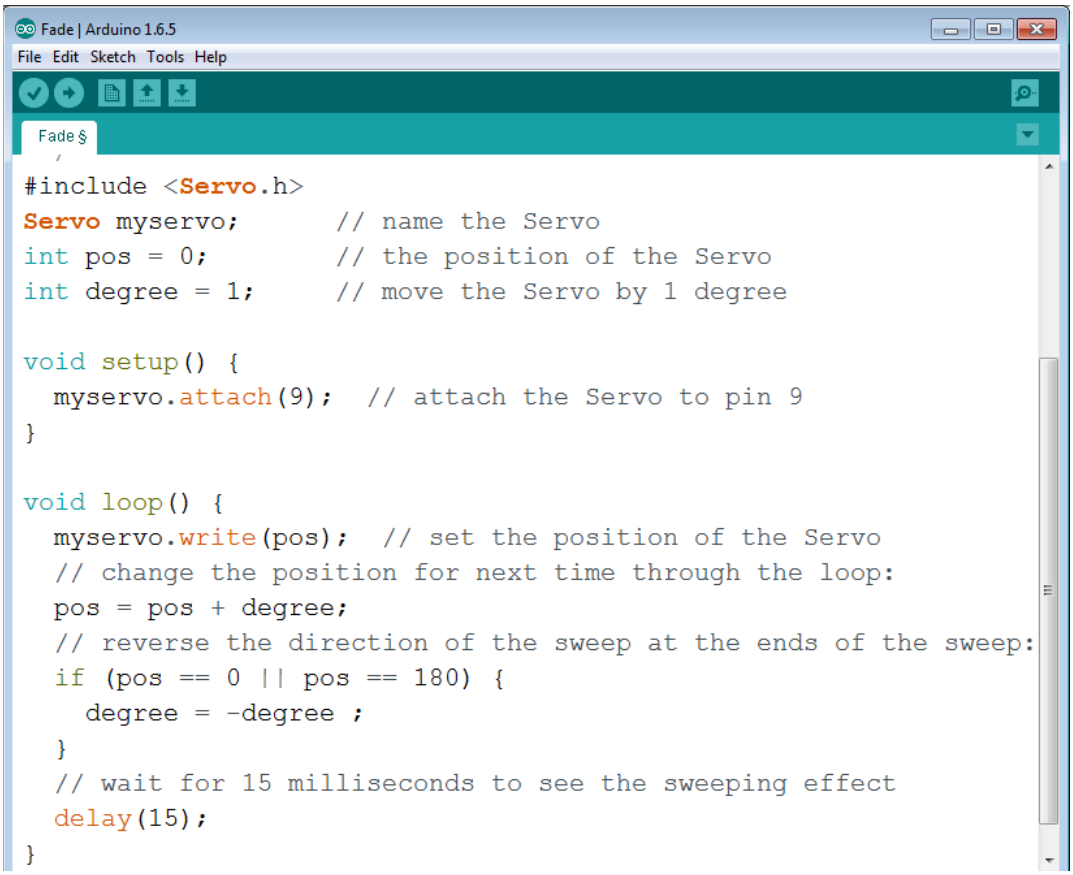
This program would make the standard Servo move from 0 degrees to 180 degrees, moving 5 degrees at a time, and pausing for 30 milliseconds between each move, allowing the Servo enough time to move the necessary amount before moving again.

1. Can you modify the altered Fade example above to do the exact same motion as the Sweep example code? You'll need to change the degree and the delay.

2. Can you use the Sweep example and alter it to make an LED fade on and off? Use the same logic of the Fade code, but use the "for" statement of the Sweep example instead of the "if" statement of the Fade example.

Challenge Solutions

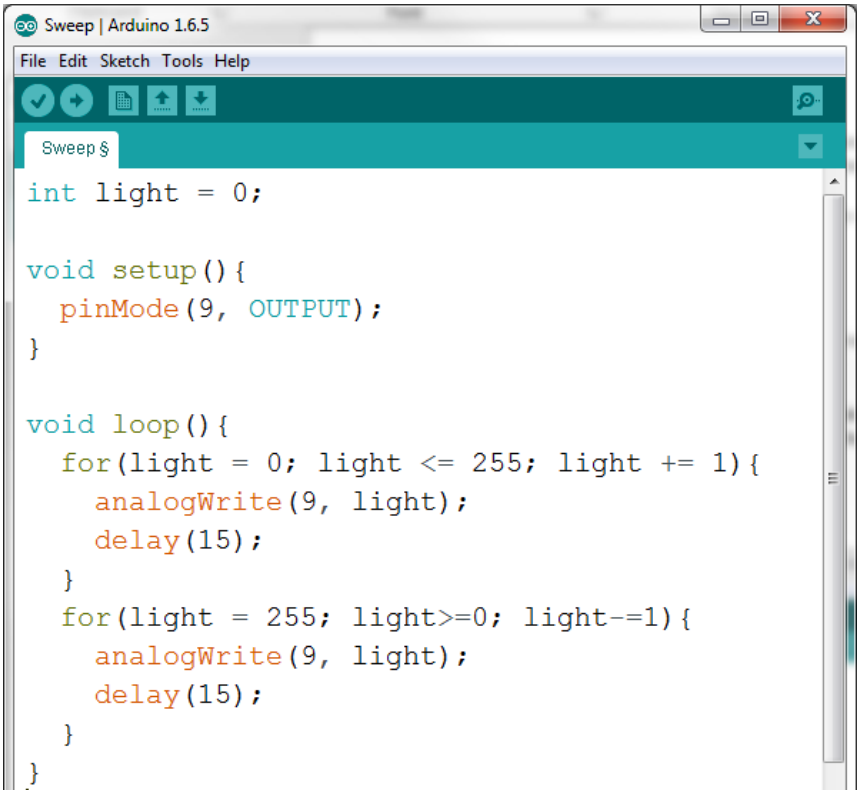
Challenge 1 Solution:



Challenge 1 Explanation:

Now, there is a 15 millisecond wait at each degree instead of a 30 millisecond wait every 5 degrees, so the Servo will make the exact same sweeping motion as it would with the sweep example.

Challenge 2 Solution:



Challenge 2 Explanation:

Notice that we use the variable "light" instead of "pos" and we use the language of an LED (**`pinMode`** and **`analogWrite`**) instead of a Servo (**`myservo.attach`** and **`myservo.write`**). When connecting your circuit to make this LED Fade, be sure to attach the LED to pin 9, as stated in the setup.