

Automated Fish Feeder Design



Main Concept: Automation

In computer science automation is defined as “controlling a process by automatic means, reducing human intervention to a minimum” using computer algorithms (Krauss and Protsman 2017).

Automation can have very negative consequences for local economies, such as when corporations automate factory work that results in job loss. Some people think this threat is so serious that they think we should all have Universal Basic Income - that is everyone should get “free” money each month for living.

But automation can also be helpful, reducing the need to perform tedious and dangerous work



Automated Fish Feeder

Today we are going to automate a job that no one is paid to do, except maybe folks who work in pet stores or at aquariums: feeding fish. In groups of 2-3 you are going to design an automated fish feeder to attach to an aquaponics system that we are going to use to grow ingredients for natural cosmetic products.



Three Design Theme

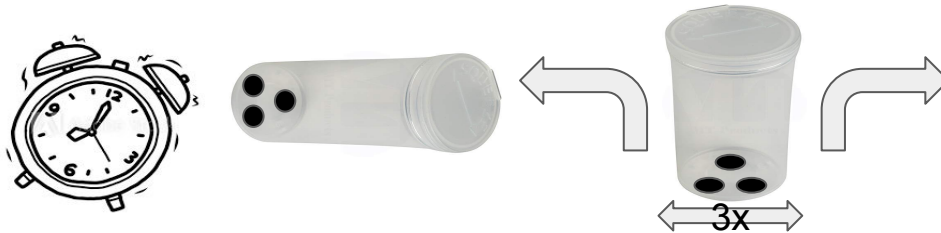
Dump -



Rotate -



Shake -



Working with your materials



Working with your materials



Dump

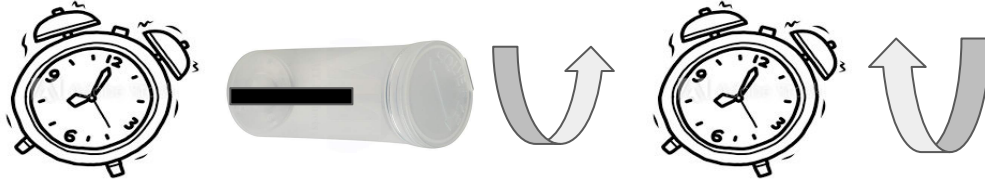


‘Dump’ is an example of an automatic fish feeder. It consists of a vial cut in half longways, looking like a boat. The servo is attached to the base of the vial.

The code (which we’ll get into later) works like this:

1. The servo waits a specified amount of time
2. The servo then rotates 180 degrees, completely dumping the fish food

Rotate

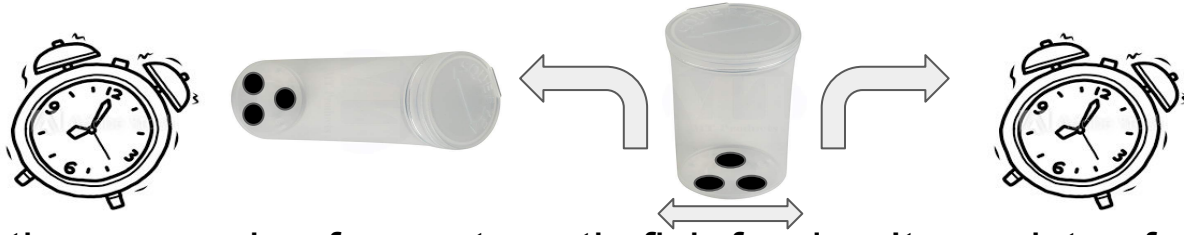


‘Rotate’ is another example of an automatic fish feeder. It consists of a vial with a small slit cut into it longways starting at the 9 o’clock position. The servo is attached to the base of the vial.

The code (which we’ll get into later) works like this:

1. The servo waits a specified amount of time
2. The servo then rotates 180 degrees counterclockwise, dumping only a feedings worth of food
3. The servo waits a specified amount of time
4. The servo then rotates 180 degrees clockwise, dumping only a feedings worth of food
5. The cycle starts again at step 1

Shake



‘Shake’ is another example of an automatic fish feeder. It consists of a vial with small holes cut into its base. The servo is attached to the side of the vial.

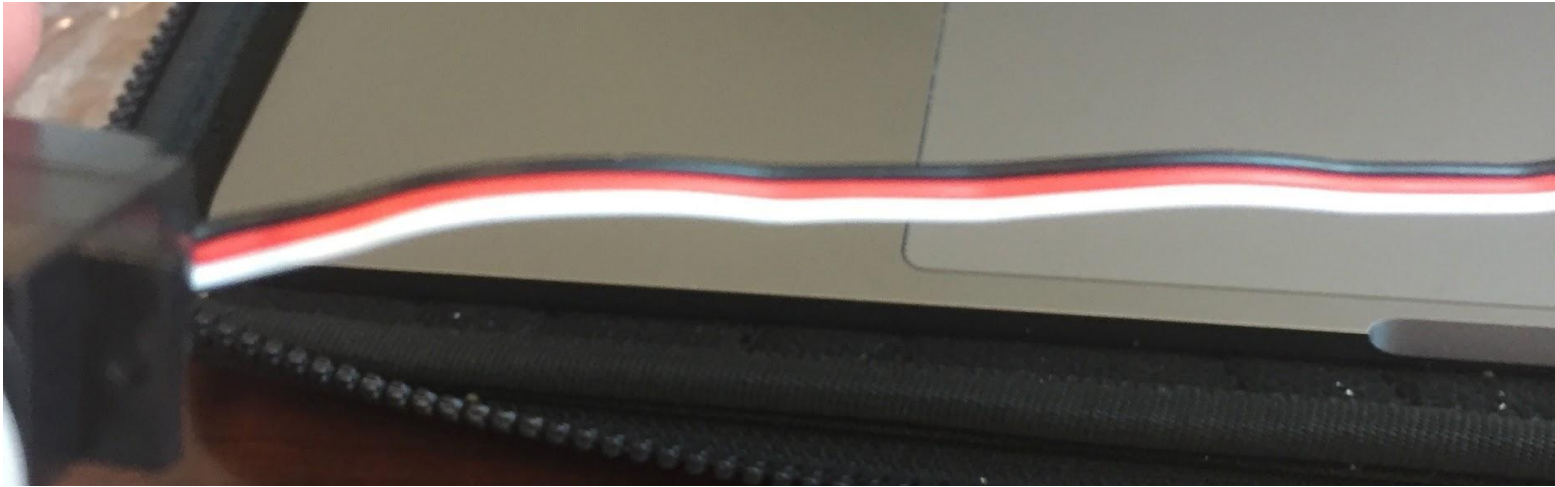
The code (which we’ll get into later) works like this:

1. The servo waits a specified amount of time
2. The servo then rotates 90 degrees counterclockwise so the vial is upright
3. The servo then shakes the vial 3 times
4. The servo then rotates 90 degrees clockwise so the vial is on its side
5. The cycle starts again at step 1

Take out your servo motor and choose and attachments and screw onto your servo motor



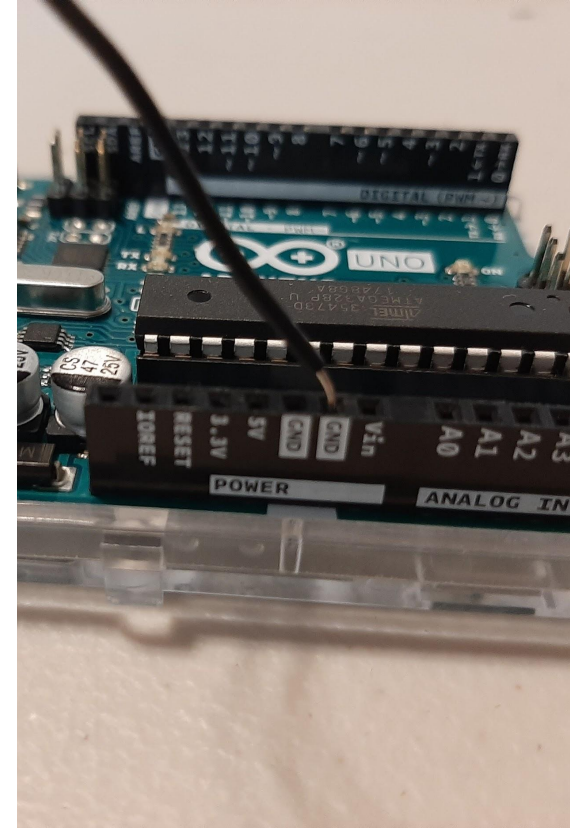
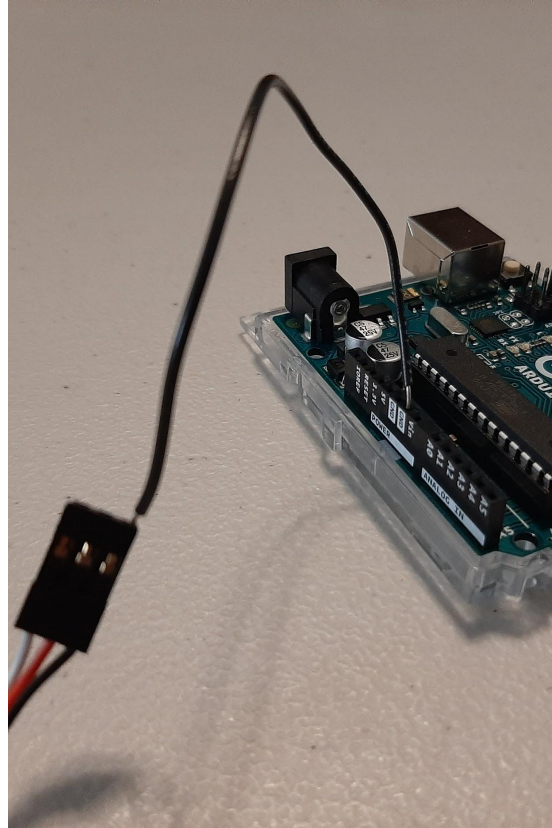
Note that there are three different color wires for your servo motor. We are going to be consistent with our wire colors when connecting the motor to the Arduino



Hooking up your servo to the Arduino

Cut some wire, strip about a quarter inch off of each end using the 22 AWG slot.

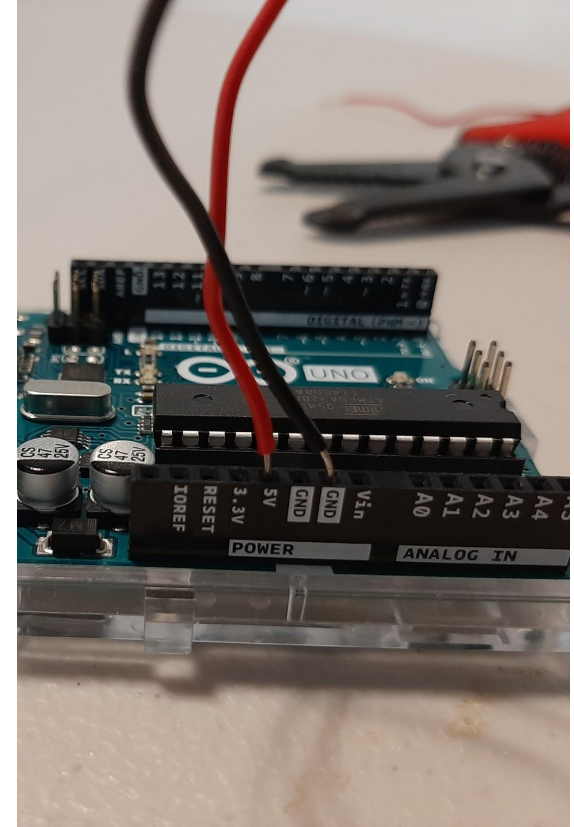
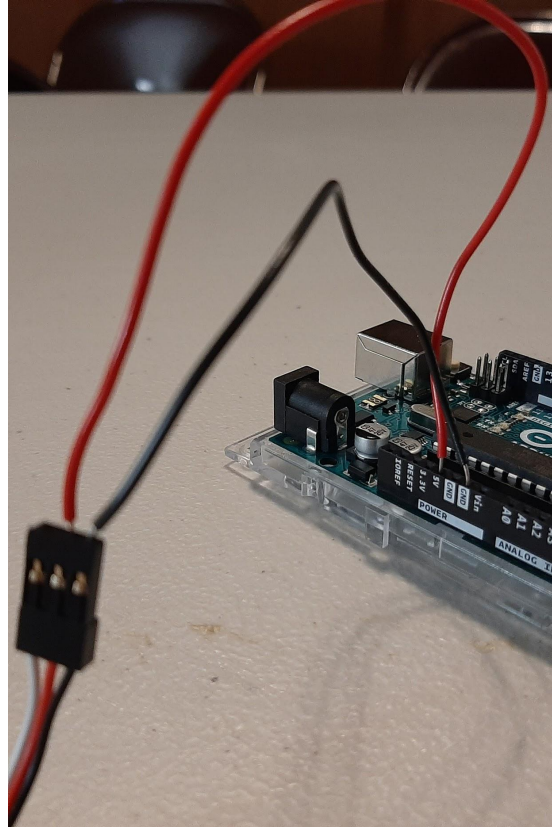
Using the wire, connect:
The black wire from the servo to the
GND pin on the Arduino



Hooking up your servo to the Arduino

Cut some wire, strip about a quarter inch off of each end using the 22 AWG slot.

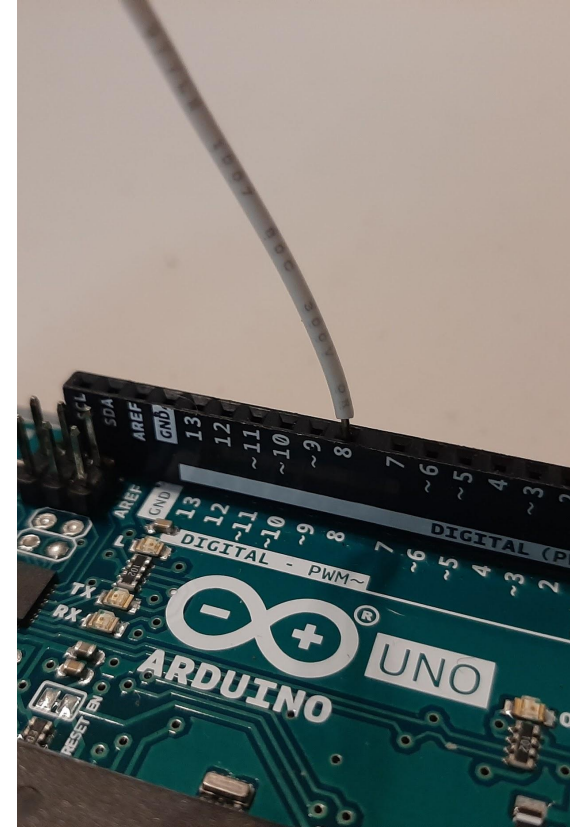
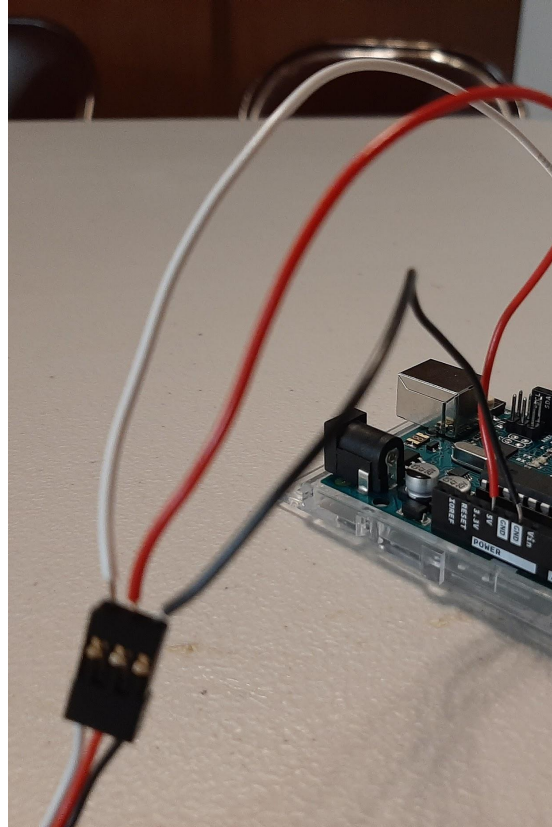
Using the wire, connect:
The red wire from the servo to the 5V pin on the Arduino



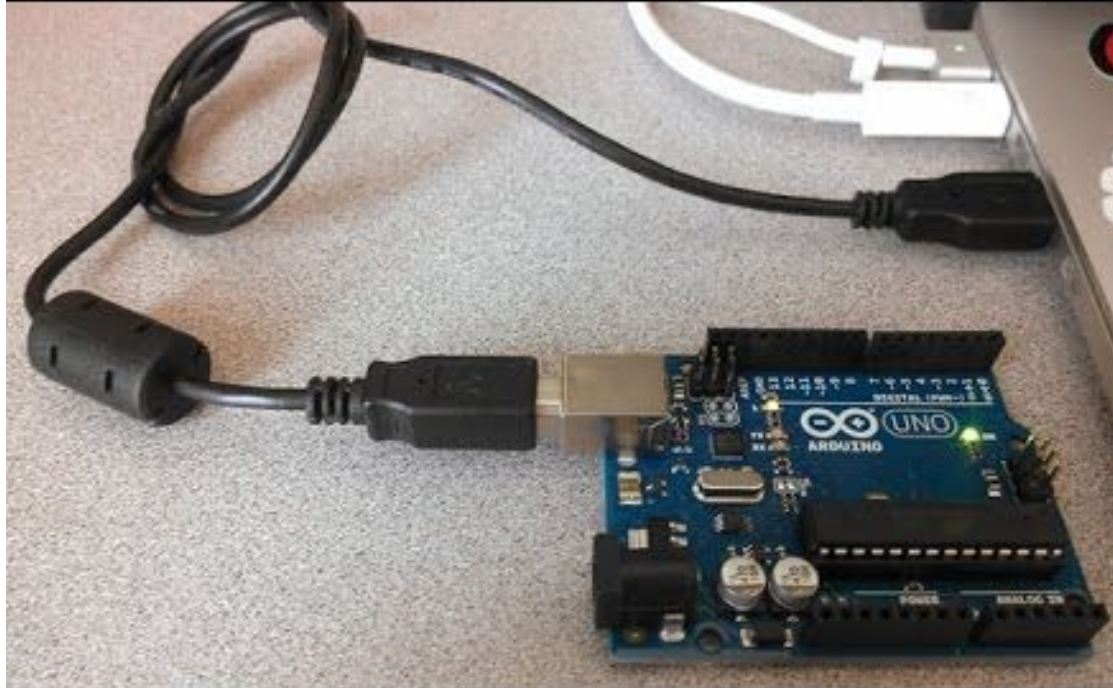
Hooking up your servo to the Arduino

Cut some wire, strip about a quarter inch off of each end using the 22 AWG slot.

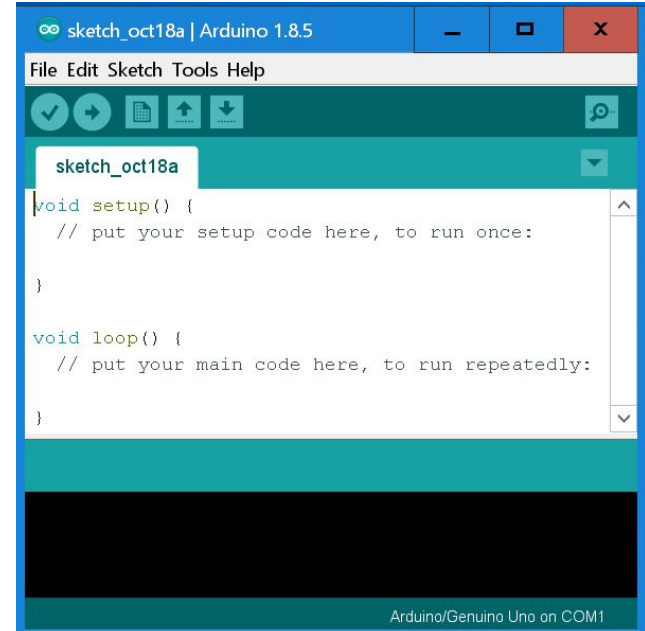
Using the wire, connect:
The white wire from the servo to the 8 pin on the Arduino



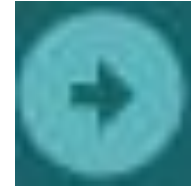
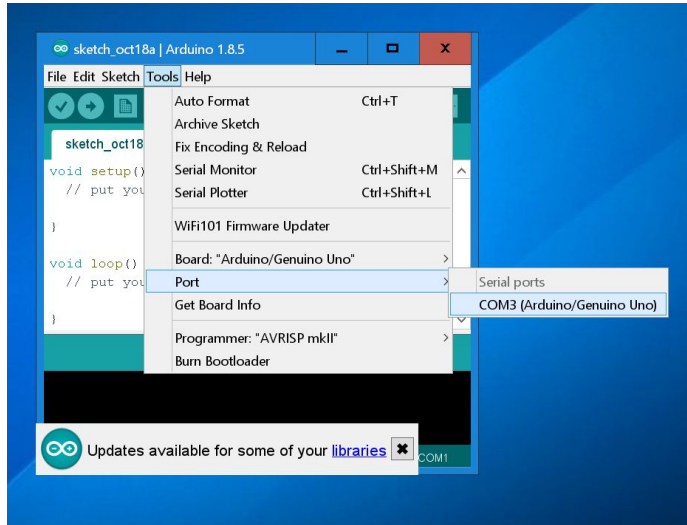
Plug one end of the USB cable into the USB slot in your computer and the other more boxy end into your Arduino.



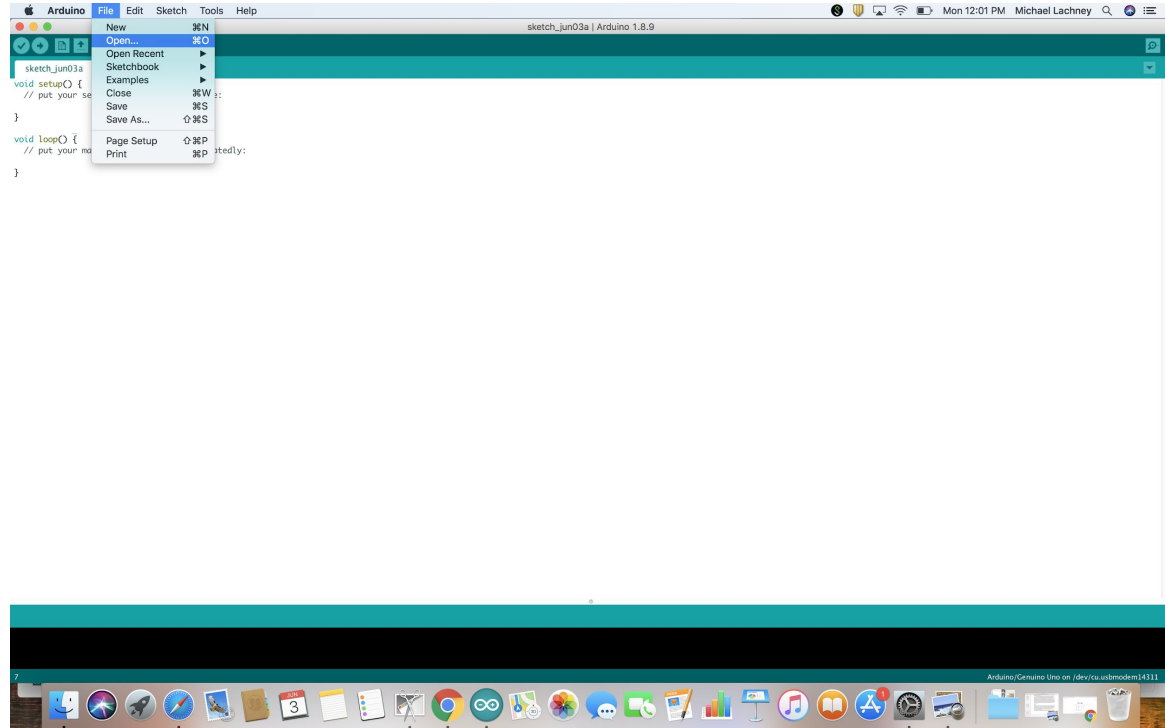
Now go to the Windows Desktop, double click on the Arduino IDE icon and the Arduino IDE (Integrated Development Environment) will start.



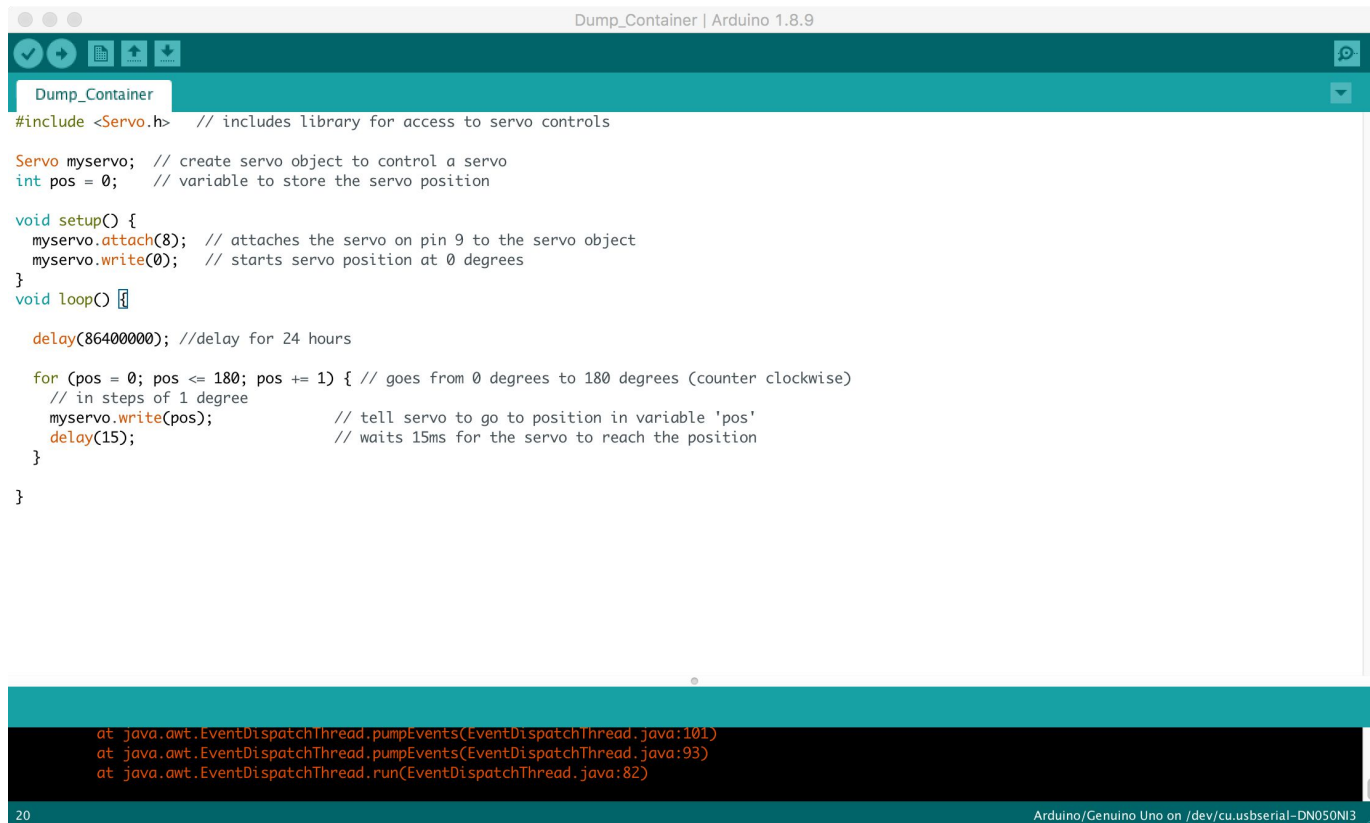
Once the file is open we need to select the "Port" on the computer that should be used to receive data from the Arduino. Click on the Tools menu at the top of the IDE, then click on "Port". The computer should show a list of ports with one of them indicating that the Arduino is attached. In the picture, this computer found the Arduino attached to the COM3 communications port. Click on the correct COM port for your computer and a checkmark will show that it is selected. Then press the arrow button to upload the code.



In the the Arduino IDE click “File” and then “Open”. Open “Code” folder on the Desktop. Open the “Dump” folder and click “Dump” file.



Dump: You should see something that looks like this



The screenshot shows an Arduino IDE window titled "Dump_Container | Arduino 1.8.9". The code in the editor is as follows:

```
#include <Servo.h> // includes library for access to servo controls

Servo myservo; // create servo object to control a servo
int pos = 0; // variable to store the servo position

void setup() {
  myservo.attach(8); // attaches the servo on pin 9 to the servo object
  myservo.write(0); // starts servo position at 0 degrees
}

void loop() {

  delay(86400000); //delay for 24 hours

  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees (counter clockwise)
    // 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
  }
}
```

The serial monitor at the bottom shows the following output:

```
at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:101)
at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:93)
at java.awt.EventDispatchThread.run(EventDispatchThread.java:82)
```

The status bar at the bottom indicates "Arduino/Genuino Uno on /dev/cu.usbserial-DN050N13".

Find the Delay Function and play the delay by changing the milliseconds

```
delay();
```

Pauses the program for the amount of time (in milliseconds) specified as parameter.

Example:

<pre>delay(1000);</pre>	Pauses the program for 1 second
<pre>delay(60000);</pre>	Pauses the program for 1 minute

Delay Challenge

Challenge: What number would you need to enter as the parameter (between the parentheses) to have the program wait 1 hour?

```
delay (?);
```

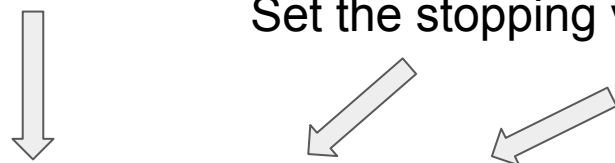
For Loop

A loop is used for section of the code repeatedly until a particular condition is satisfied. For example, if we want our servo to incrementally go from 0 degrees to 180 degrees. We would:

Set a starting value to 0

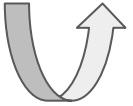
Set the stopping value to less than or equal to 180

Tell the value of 'pos' to increase by 1 every loop



```
for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees (counterclockwise)
  // 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
}
```

Dump Code



Dump §

```
#include <Servo.h> // includes library for access to servo controls

Servo myservo; // create servo object to control a servo
int pos = 0; // variable to store the servo position

void setup() {
  myservo.attach(8); // attaches the servo on pin 8 to the servo object
  myservo.write(0); // starts servo position at 0 degrees
}

void loop() {

  delay(10000); //delay for 10 seconds (10seconds*1000milliseconds)

  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees (counterclockwise)
    // 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
  }
}
```

Dump Code



You were given the code for Dump which shows:

1. Waits for 10 seconds

```
delay(10000); //delay for 10 seconds (10seconds*1000milliseconds)
```

2. Then rotates 180 degrees, dumping all the fish food

```
for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees (counterclockwise)
  // 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
}
```

Manipulate Dump Code to be Rotate

```
Rotate
#include <Servo.h> // includes library for access to servo controls

Servo myservo; // create servo object to control a servo
int pos = 0; // variable to store the servo position

void setup() {
  myservo.attach(8); // attaches the servo on pin 8 to the servo object
  myservo.write(0); // starts servo position at 0 degrees
}

void loop() {

  delay(10000); //delay for 10 seconds (10seconds*1000milliseconds)

  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees (counterclockwise)
    // 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
  }

  delay(10000); //delay for 10 seconds (10seconds*1000milliseconds)

  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees (clockwise)
    // 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
  }

}
```

(details on following pages)

NEW STUFF

Manipulate Dump Code to be Rotate

You were given the code for Dump:

1. Add an additional wait for 10 seconds under the last for loop

```
delay(10000); //delay for 10 seconds (10seconds*1000milliseconds)
```


Manipulate Dump Code to be Rotate


You were given the code for Dump which:

2. Then rotates 180 degrees clockwise, from 180 degrees to 0
 - a. Note the changes:

Set a starting value to 180

Set the stopping value greater than or equal to 0

Tell the value of 'pos' to decrease by 1 every loop



```
for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees (clockwise)
  // 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
}
```

Manipulate Dump Code to be Shake

(details on following pages)

NEW STUFF

```
Shake §
#include <Servo.h> // includes library for access to servo controls

Servo myservo; // create servo object to control a servo
int pos = 0; // variable to store the servo position

void setup() {
  myservo.attach(8); // attaches the servo on pin 8 to the servo object
  myservo.write(0); // starts servo position at 0 degrees
}

void loop() {
  delay(10000); //delay for 10 seconds (10seconds*1000milliseconds)

  for (pos = 0; pos <= 90; pos += 1) { // goes from 0 degrees to 90 degrees (counterclockwise)
    // 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 = 90; pos <= 120; pos += 1) { // goes from 0 degrees to 90 degrees (counterclockwise)
    // in steps of 1 degree
    myservo.write(pos); // tell servo to go to position in variable 'pos'
    delay(5); // waits 5ms for the servo to reach the position (quick shake)
  }

  for (pos = 120; pos >= 60; pos -= 1) { // goes from 120 degrees to 60 degrees (clockwise)
    // in steps of 1 degree
    myservo.write(pos); // tell servo to go to position in variable 'pos'
    delay(5); // waits 5ms for the servo to reach the position (quick shake)
  }

  for (pos = 60; pos <= 120; pos += 1) { // goes from 60 degrees to 120 degrees (counterclockwise)
    // in steps of 1 degree
    myservo.write(pos); // tell servo to go to position in variable 'pos'
    delay(5); // waits 5ms for the servo to reach the position (quick shake)
  }

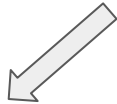
  for (pos = 120; pos >= 90; pos -= 1) { // goes from 120 degrees to 90 degrees (clockwise)
    // in steps of 1 degree
    myservo.write(pos); // tell servo to go to position in variable 'pos'
    delay(5); // waits 5ms for the servo to reach the position (quick shake)
  }

  for (pos = 90; pos >= 0; pos -= 1) { // goes from 90 degrees to 0 degrees (clockwise)
    // in steps of 1 degree
    myservo.write(pos); // tell servo to go to position in variable 'pos'
    delay(15); // waits 5ms for the servo to reach the position
  }
}
```

Manipulate Dump Code to be Rotate

1. Change for loop to match below

Set the stopping value less than or equal to 90



```
for (pos = 0; pos <= 90; pos += 1) { // goes from 0 degrees to 90 degrees (counterclockwise)
  // 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
}
```

Manipulate Dump Code to be Rotate

2. Add the following for loop

Set a starting value to 90

Set the stopping value less than or equal to 120

Tell the value of 'pos' to increase by 1 every loop

```
for (pos = 90; pos <= 120; pos += 1) { // goes from 0 degrees to 90 degrees (counterclockwise)
  // in steps of 1 degree
  myservo.write(pos);                // tell servo to go to position in variable 'pos'
  delay(5);                          // waits 5ms for the servo to reach the position (quick shak
}
```

Make sure to change the delay parameter to 5

Manipulate Dump Code to be Rotate

3. Add the following for loop

Set a starting value to 120

Set the stopping value greater than or equal to 60

Tell the value of 'pos' to decrease by 1 every loop

```
for (pos = 120; pos >= 60; pos -= 1) { // goes from 120 degrees to 60 degrees (clockwise)
  // in steps of 1 degree
  myservo.write(pos);                // tell servo to go to position in variable 'pos'
  delay(5);                          // waits 5ms for the servo to reach the position (quick shak
}
```

Make sure to change the delay parameter to 5

Manipulate Dump Code to be Rotate

4. Add the following for loop

Set a starting value to 60

Set the stopping value less than or equal to 120

Tell the value of 'pos' to increase by 1 every loop

```
for (pos = 60; pos <= 120; pos += 1) { // goes from 60 degrees to 120 degrees (counterclockwise)
  // in steps of 1 degree
  myservo.write(pos);                // tell servo to go to position in variable 'pos'
  delay(5);                          // waits 5ms for the servo to reach the position (quick shake)
}
```

Make sure to change the delay parameter to 5

Manipulate Dump Code to be Rotate

5. Add the following for loop

Set a starting value to 120

Set the stopping value greater than or equal to 90

Tell the value of 'pos' to decrease by 1 every loop

```
for (pos = 120; pos >= 90; pos -= 1) { // goes from 120 degrees to 90 degrees (clockwise)
  // in steps of 1 degree
  myservo.write(pos);           // tell servo to go to position in variable 'pos'
  delay(5);                     // waits 5ms for the servo to reach the position (quick shak
}
```

Make sure to change the delay parameter to 5

Manipulate Dump Code to be Rotate

6. Add the following for loop

Set a starting value to 90

Set the stopping value greater than or equal to 0

Tell the value of 'pos' to decrease by 1 every loop

```
for (pos = 90; pos >= 0; pos -= 1) { // goes from 90 degrees to 0 degrees (clockwise)
  // in steps of 1 degree
  myservo.write(pos);                // tell servo to go to position in variable 'pos'
  delay(15);                         // waits 5ms for the servo to reach the position
}
```

Make sure to change the delay parameter to 15

Design Challenge

For your mini-aquaponics system come up with a design for an automated fish feeder by following these directions:

- 1) In your group talk about what design you think might work best for your aquaponics system
- 2) Draw your design with labels and write an explanation
- 3) Build your design
- 4) Program your design
- 5) Test it out!