What is a Microcontroller?

Desktop Computer

Gemma

Both of these are computers, even though the Gemma doesn't have a screen, mouse, or a keyboard attached.

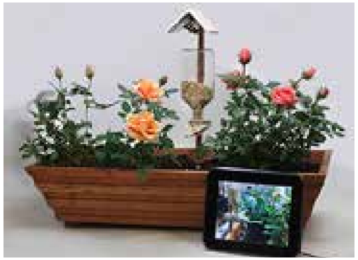
Computers are systems of inputs and outputs. You put information in by typing or clicking, and the computer will perform an operation. You will then get an output.

	Input	**Output**
Computer	Typing, clicking	Open document, send email
Gemma	Sensor, button, potentiometer	Light, sound, movement

What can I do with it?

You can make objects that respond to the world around you:

- light up
- measure temperature, moisture, or distance
- take photos
- move objects
- make sounds



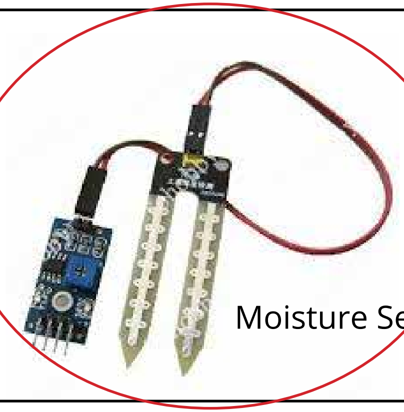
You can use the Arduino as the “brains” for many different gadgets -- for everyday life or just for fun. You could make a device that lights up a room when motion is detected, or monitor the air quality in your house. For fun, you could make a pair of electronic dice or program a tiny camera to take photos of the birds at your feeder.

What’s in this kit?

Inputs

Button

Potentiometer



Moisture Sensor

Outputs

LED

Buzzer

Servo

Connectors

Resistor

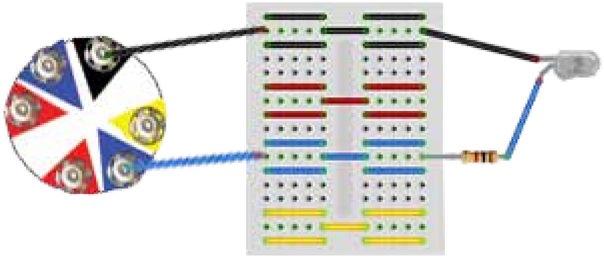
Wires

Breadboard

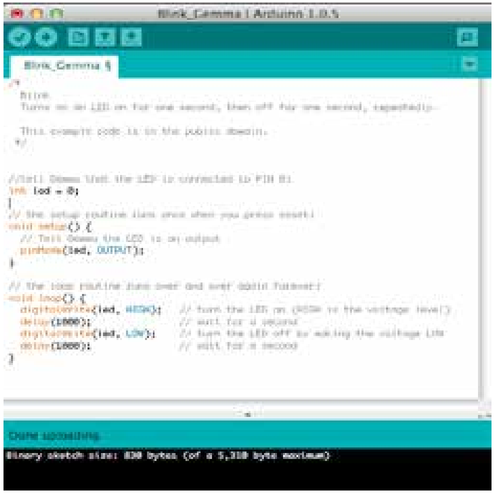
USB cord

How do I use it?

To use the Gemma, you need to do two things:



Build a circuit with wires and components



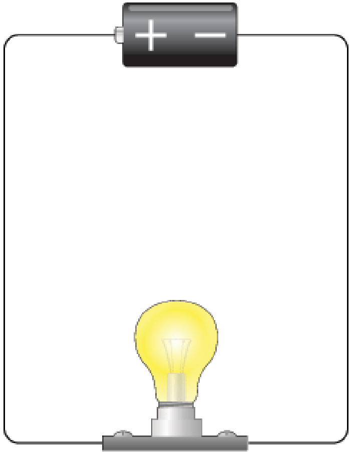
```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */


//Tell Gemma that the LED is connected to PIN 0!
int led = 0;
|
// the setup routine runs once when you press reset:
void setup() {
  // Tell Gemma the LED is an output
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

```
Done uploading.
Binary sketch size: 830 bytes (of a 5,310 byte maximum)
```

Write or choose a program and send it to the Gemma

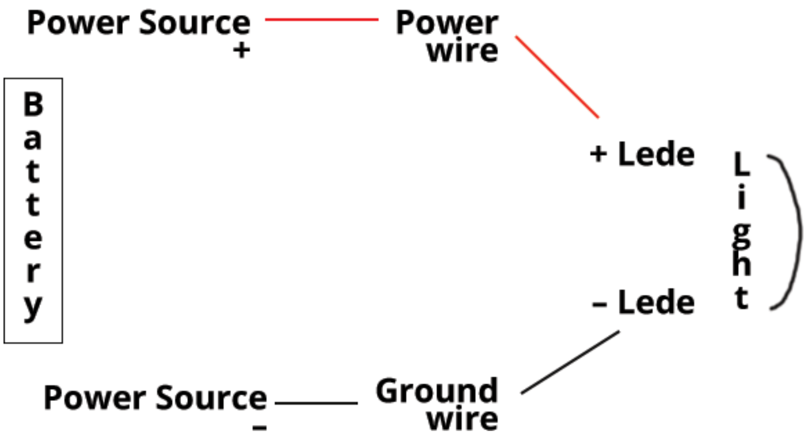
Building a circuit

The simplest circuit connects a light bulb to a battery. You just need a battery, a light, and two pieces of conductive material.



Using the cards in the kit, try to create a working circuit with the copper tape. If everything is connected properly, the LED will light.

You don’t need to use wires for every project. besides conductive copper tape, try conductive thread or alligator clips.



Power Source +

Power wire

Battery

+ Lede

– Lede

Light

Power Source –

Ground wire

The positive lede of an LED is the blue leg with the resistor attached.

The negative lede (black leg) always gets connected to GROUND.

These batteries have a holder with postive and negative terminals. The lithium battery also has positive and negative sides. Check the markings before inserting the battery. It should snap into the holder.

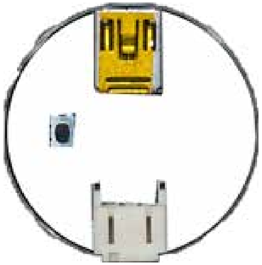
Building a circuit with Gemma

Gemma has several “pins” that allow us to connect to a power source, a ground, input, and output.

The Gemma is like a middleman. To make these connections, you will snap wires onto the corresponding pins.

BACK

USB port (to computer)



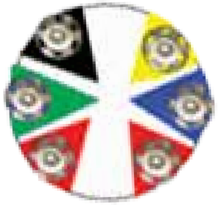
Reset button

Battery port

FRONT

Black - GROUND

Yellow - Pin A1 INPUT



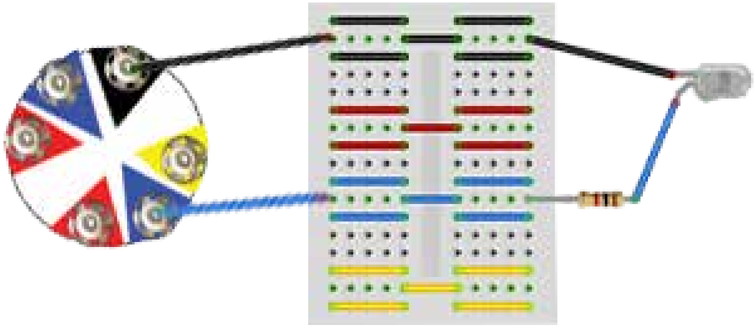
Green - Pin 1 OUTPUT

Blue - Pin 0 OUTPUT

Red - POWER

Our first gadget will be a blinking light. You already know how to set up a circuit that powers an LED. Now, you'll connect that light to the Gemma and use the computer to tell the light to blink on and off.

Snap a blue wire to the blue 0 pin and stick the other end of the wire into the blue row on the breadboard.



Snap a black wire to the black GROUND pin and stick the other end into a black row.

Stick an LED in the breadboard:

Put the end of the resistor (on the blue leg of the LED) into a hold in a blue row.

Stick the black leg of the LED into the black row.

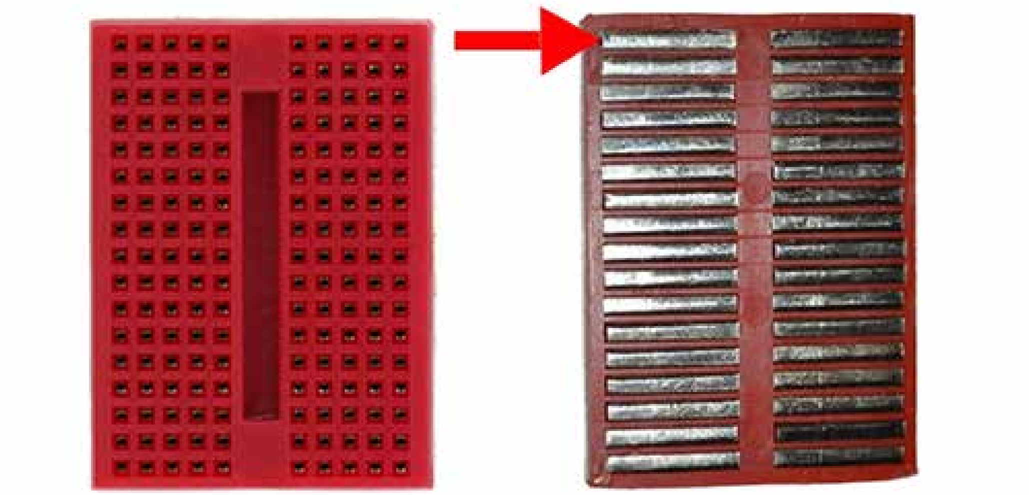
Try substituting a buzzer for the LED. Light the LED, the buzzer has a long "leg" for the positive lede.

Using a breadboard

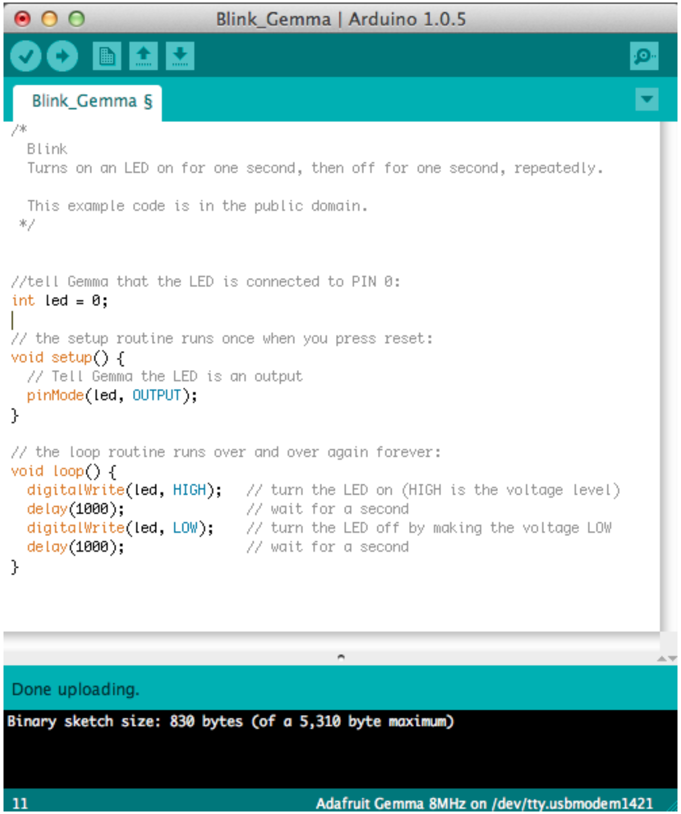
A breadboard makes it easy to connect two wires or connect a component to a wire. Breadboard keep projects neat and organized.

Each row of five holes is connected together.

Practice making a connection between two wires by placing the ends of the wires in holes in the same row. Trying to light your LED through the breadboard.



Programming Gemma


Blink_Gemma | Arduino 1.0.5
Blink_Gemma §
11
Adafruit Gemma 8MHz on /dev/tty.usbmodem1421

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */


//tell Gemma that the LED is connected to PIN 0:
int led = 0;

// the setup routine runs once when you press reset:
void setup() {
  // Tell Gemma the LED is an output
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

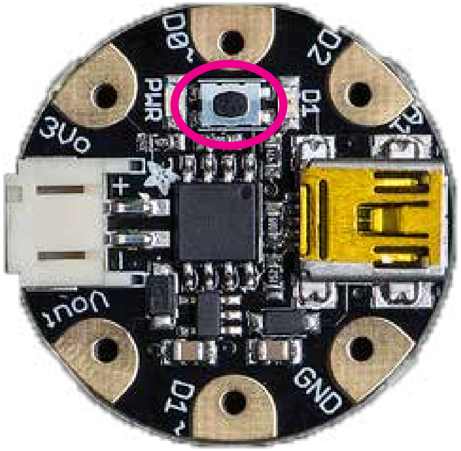
```
Done uploading.
Binary sketch size: 830 bytes (of a 5,310 byte maximum)
```

With the Arduino program open on the computer, select the Gemma_Blink sketch. It is located in the dropdown menu under FILE --> SKETCHBOOK.

Try to edit the sketch by changing the “delay” from (1000) to another number. Upload the sketch again (press the tiny button and click the “upload” button within ten seconds).

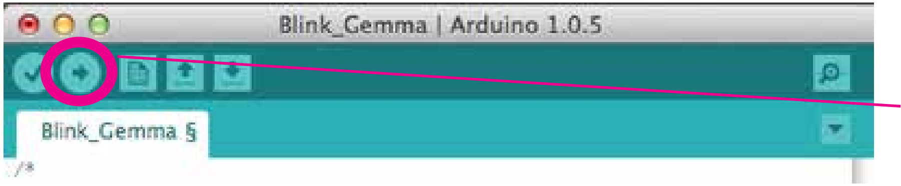
Connecting Gemma

To send a program to the Gemma, you need to connect it to your computer. After you upload the sketch, you can use a battery to power the Gemma. It will remember the last sketch.



Connect the mini USB (the smallest male plug) into the gold port on the Gemma.

Press the tiny black button (circled above) and then upload the Arduino sketch within 10 seconds by clicking the arrow button at the top of the sketch window (second from the left).



Blink_Gemma | Arduino 1.0.5

Blink_Gemma §

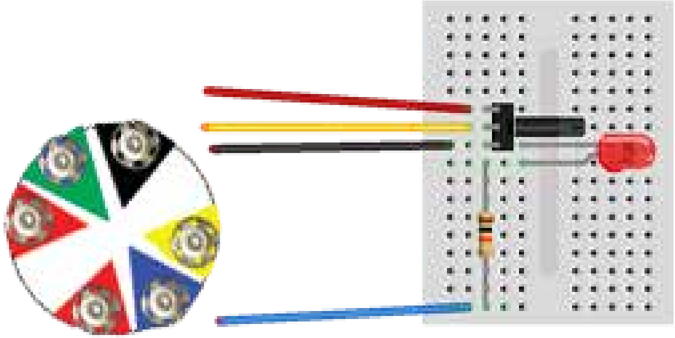
/*

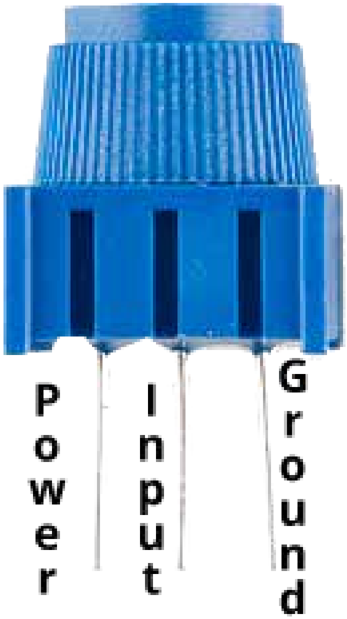
Add an input

Our blink sketch told the LED to blink at a steady rate. With an input, we can control the rate of blinking.

A POTENTIOMETER is a knob. It has three pins: power, analog in (sends a signal to the Gemma), and ground.

Open the gemma_potentiometer sketch and wire up the breadboard according to the diagram.

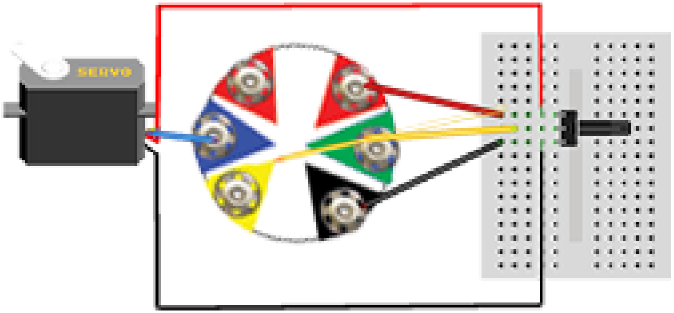




Change the output

The potentiometer can easily be used with another output: a servo. Open the gemma_servo sketch and wire the breadboard so that the red power wires are all on one row and the black grounds are all on another row.

What happens when you turn the knob?



Servos have three wires, just like the potentiometer. A red wire for power, black for ground, and a third wire that receives a signal from the Gemma.

Use a sensor

A moisture sensor has three wires, just like the servo and potentiometer.

This moisture sensor can tell you when it’s time to water a plant. Use it with an LED.

Wire the LED just as you did in the blink sketch.

The moisture sensor is an input with three wires -- just like the potentiometer.

Open the gemma_moisture sketch and upload it. You can test your circuit with a damp paper towel (don’t dunk any component in water!)

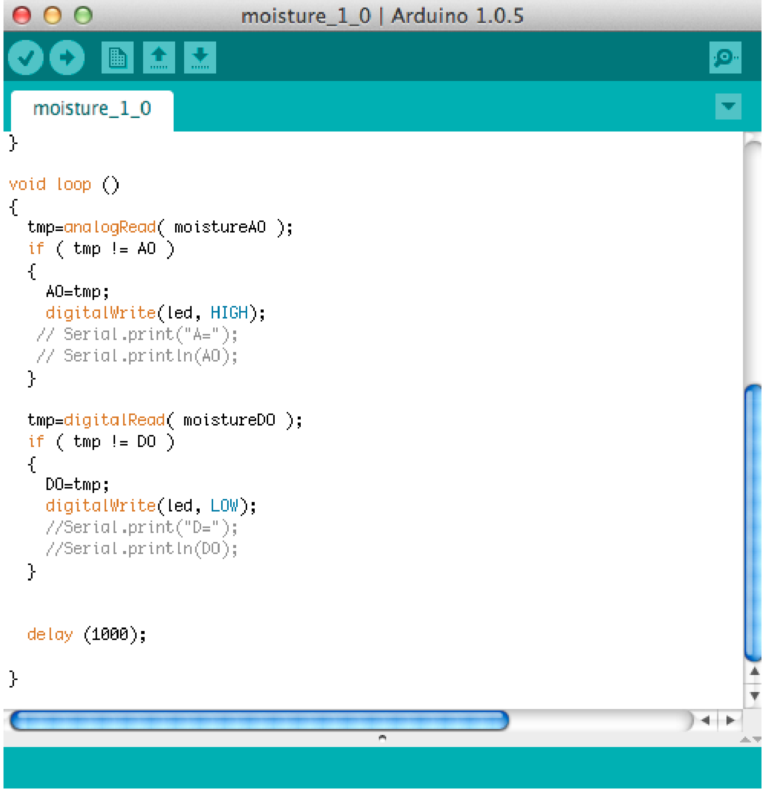
Edit the code

Making some small changes to the code can make a sketch serve your needs.

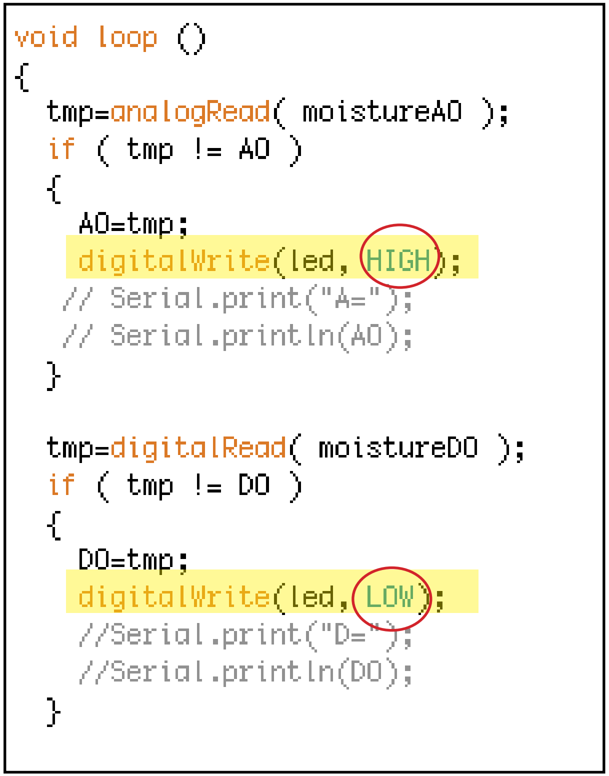
When you run the gemma_moisture sketch, you’ll notice that the LED turns on when the sensor touches something damp.

If you want to make a more effective gadget, you might want the LED to turn on when your plant *needs* more water.

We can do that by exchanging two bits of information in the code.



```
}

void loop ()
{
  tmp=analogRead( moistureA0 );
  if ( tmp != A0 )
  {
    A0=tmp;
    digitalWrite(led, HIGH);
   // Serial.print("A=");
   // Serial.println(A0);
  }

  tmp=digitalRead( moistureD0 );
  if ( tmp != D0 )
  {
    D0=tmp;
    digitalWrite(led, LOW);
    //Serial.print("D=");
    //Serial.println(D0);
  }


  delay (1000);

}
```

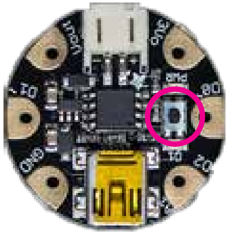


```
void loop ()
{
  tmp=analogRead( moistureA0 );
  if ( tmp != A0 )
  {
    A0=tmp;
    digitalWrite(led, HIGH);
   // Serial.print("A=");
   // Serial.println(A0);
  }

  tmp=digitalRead( moistureD0 );
  if ( tmp != D0 )
  {
    D0=tmp;
    digitalWrite(led, LOW);
    //Serial.print("D=");
    //Serial.println(D0);
  }
```

In order to make the light turn ON when the sensor is dry, we need to swap two statements:

```
digitalWrite(led, HIGH);

digitalWrite(led, LOW);
```

You can delete and type in the sketch as you would any word processor. Simply delete `HIGH` and type `LOW` in its place. Code is case-sensitive, so be sure to use all capital letters in this case. Don't forget to change the `LOW` to `HIGH` further down in the code.



After you make the changes, press the tiny button on the back of the Gemma and then click the upload button on the sketch.