

ENGINEERING A BLOOMING FLOWER!

ADDING SENSORS EXPANSION:

AUGUST 2025

PIR Motion Sensor: <https://thePIHUT.com/products/grove-adjustable-pir-motion-sensor>
Light Dependent Resistor: <https://thePIHUT.com/products/ldr-light-dependent-resistor>
Force Sensitive Resistor: <https://thePIHUT.com/products/round-high-force-sensitive-resistor-fsr-1-100-newton-force-alpha-mf01a-n-221-a05>

WHAT YOU'LL NEED:

(ON TOP OF PREVIOUS INSTRUCTIONS)

- CROCODILE LEADS/JUMPER WIRES
- PIR MOTION SENSOR
- 10KΩ RESISTOR
- LIGHT DEPENDENT RESISTOR
- FORCE SENSITIVE RESISTOR

Introduction to Adding Sensors

By adding sensors for motion, light, or touch, your flower can react automatically to the world around it. The micro:bit reads the sensor input and tells the servo motor when to open or close the bloom. 🌸🤖

Step 1: Connect the PIR Motion Sensor

Using your existing automated setup, connect the PIR motion sensor to the micro:bit:

Ensure the GND is shared between both the servo and the sensor. Connect the sensor's PWR pin to 3V on the micro:bit (the same supply used by the servo). Connect the sensor's data pin to Pin 1 (since Pin 0 is already in use for the servo).

Step 2: Code Your Micro:bit

Teach the micro:bit how to open and close the flower smoothly when the motion sensor spots someone! 🤖🌸

Make a variable: Call it **angle**. This will tell the motor what position to move to.

Check the sensor: Use a forever loop with an if block:

if digital read pin P1 ≠ 0 then → this means "if the motion sensor sees movement..."

Open the flower:

Set angle to 0.

Repeat 180 times:

Change **angle** by +1 (change this value to increase/decrease the speed).

Move the servo on Pin 0 to that angle.

Pause for 20 ms (to make it move smoothly).

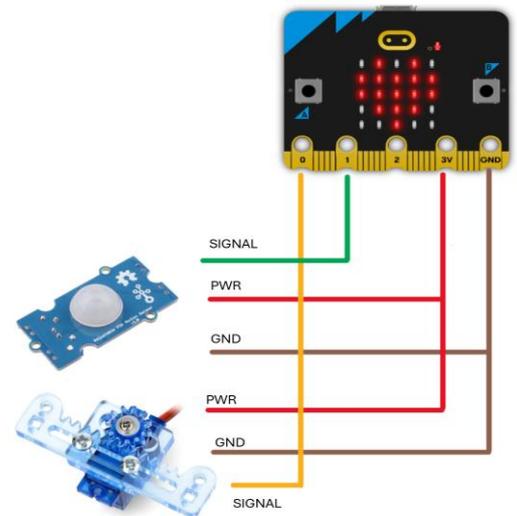
Close the flower:

Repeat 180 times:

Change **angle** by -1.

Move the servo on Pin 0 back to that angle.

Pause for 20 ms.



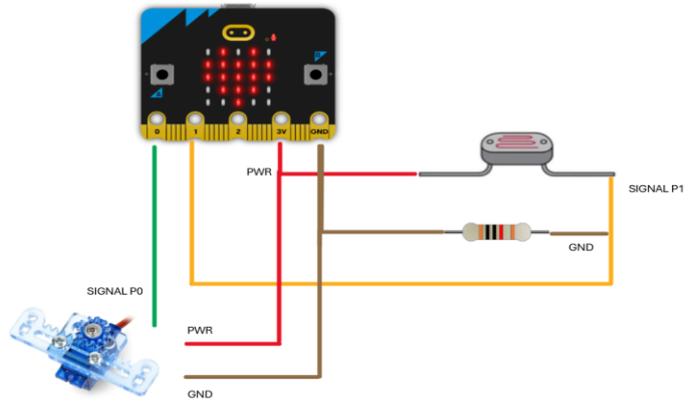
```
forever
  if digital read pin P1 ≠ 0 then
    set angle to 0
    repeat 180 times
      do
        change angle by 1
        set servo P0 angle to angle °
        pause (ms) 20
      end do
    end repeat
    repeat 180 times
      do
        change angle by -1
        set servo P0 angle to angle °
        pause (ms) 20
      end do
    end repeat
  end if
end forever
```

Step 3: Build the LDR Circuit

Connect the LDR circuit as shown in the diagram to the right.

Notice how the GND line is pulled down by a 10 kΩ resistor. This creates the voltage divider that lets the micro:bit measure light levels.

You can use crocodile clips to make the split connection between the LDR, the resistor, and Pin 1.



Step 4: Program the Light Sensor

Follow the code below which makes the flower open when the environment is bright, and close again when it becomes dark.

At the top, the micro:bit reads the voltage from the LDR circuit (Pin 1) and divides it to give a number between 0–20.

A threshold called lightlevel is set to 12 by default (in the on start block).

Adjusting the threshold:

Press Button A to increase the light level threshold.

Press Button B to decrease it.

This way, you can tune the sensitivity to match the actual lighting in your classroom. 🌻

How the servo moves:

If the light number is greater than the threshold *and* the flower is currently closed, the code:

- Starts the angle at 0.
- Slowly sweeps the servo from 0° → 180°, pausing 20 ms each step, to open the flower smoothly.
- Updates status to 1 (flower is open).

If the light number is less than or equal to the threshold *and* the flower is currently open, the code:

- Starts the angle at 180.
- Slowly sweeps the servo from 180° → 0°, pausing 20 ms each step, to close the flower smoothly.
- Updates status to 0 (flower is closed).

```
forever
  set reading to analog read pin analog pin P1
  set number to reading integer / 50
  if number > lightlevel and status = 0 then
    set angle to 0
    while angle < 181
      do
        set servo P0 angle to angle
        pause (ms) 20
        change angle by 1
      do
        set status to 1
    else if number <= lightlevel and status = 1 then
      set angle to 180
      while angle > -1
        do
          set servo P0 angle to angle
          pause (ms) 20
          change angle by -1
        do
          set status to 0

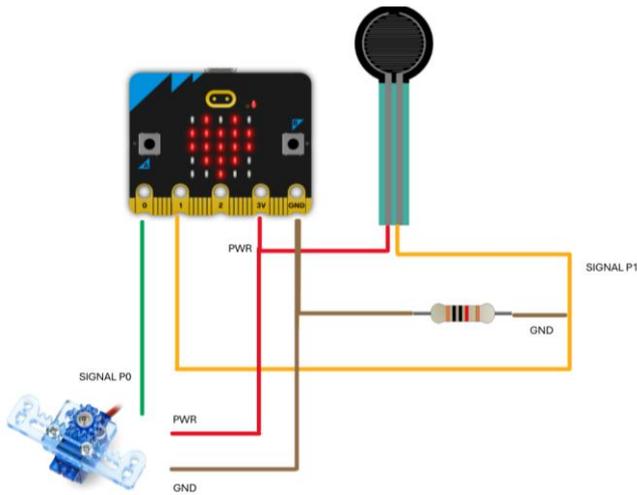
on start
  set lightlevel to 12

on button A pressed
  change lightlevel by 1

on button B pressed
  change lightlevel by -1
```

🔧 Step 5: Build the FSR Circuit

Connect the circuit as shown in the diagram below. It uses the same potential divider setup as the LDR, but this time you simply swap the LDR for the FSR. Just like with the LDR, you can use crocodile clips to make the split connection.



```
forever
  if (analog read pin P1 <= 50) then
    set status to 0
  else if (analog read pin P0 > 50 and analog read pin P1 <= 500) then
    if (status = 0) then
      set status to 1
    else if (status = 3) then
      set status to 2
    else
      set status to 3
```

🔧 Step 6: Program the Force Sensor

This code lets the flower respond differently depending on how hard you press the FSR. The FSR is read through the potential divider on Pin 1. The program compares the sensor's value and sets a variable called status to remember which "mode" the flower should be in.

🌱 How the above code works (deciding the mode)

If the reading on Pin 1 is $\leq 50 \rightarrow$ set status = 0

(very light or no pressure)

If the reading is between 50 and 500 \rightarrow check the current status:

- If closed \rightarrow set status = 1
- If fully open \rightarrow set status = 2
- Otherwise (stronger press) \rightarrow set status = 3

So, pressing harder changes the flower's behaviour! 🌸

🌱 How the right-hand side code works (moving the flower)

The forever loop checks status and runs different motions:

- status = 0 \rightarrow sweep the servo to 0° (fully closed)
- status = 1 \rightarrow sweep to 90° (half-open)
- status = 2 \rightarrow sweep back down to 90° (half-closed)
- status = 3 \rightarrow sweep to 180° (fully open)

The while loops slowly change the angle variable step by step, pausing 20 ms each time, so the petals open and close smoothly.

💡 Tip: You can adjust the threshold numbers (50, 500) to suit your sensor and pressing strength. If your flower reacts too quickly or not at all, tweak these values until it feels right.

```
forever
  if (status = 0) then
    while (angle <= 0)
      do
        change angle by -1
        set servo P0 angle to angle
        pause (ms) 20
  else if (status = 1) then
    while (angle <= 90)
      do
        change angle by 1
        set servo P0 angle to angle
        pause (ms) 20
  else if (status = 2) then
    while (angle <= 90)
      do
        change angle by -1
        set servo P0 angle to angle
        pause (ms) 20
  else if (status = 3) then
    while (angle <= 181)
      do
        change angle by 1
        set servo P0 angle to angle
        pause (ms) 20
```