

```

// This is the base program for the Plants domain. It involves using two
motors, and using an infrared sensor to control those motors.
// It also introduces controls for a 360-degree servo, but students can change
those commands if they're not using one.

//////// NAMING MULTIPLE SERVOS

#include <Servo.h>    // we want to use special commands made for controlling
servos.
Servo myservo;        // These are the names we are giving our servos.
Servo myservo2;       // Since we have two motors this time, we are naming the
second one "myservo2".

//////////////// VOID SETUP(): CODE THAT RUNS ONCE //////////////////

void setup() {

    Serial.begin(9600); // start up the Serial Monitor, which gets explained a
    little later.

    ////////// TELLING THE ARDUINO WHAT PINS OUR SERVOS ARE ON

    myservo.attach(9); // One servo is on pin 9,
    myservo2.attach(10); // and a second servo is on pin 10. Change these if you
    are using different pins.

}

//////////////// VOID LOOP(): CODE THAT RUNS OVER AND OVER //////////////////

void loop() {

    ////////// CHECKING THE READING FROM A SENSOR ON ANALOG PIN 0
    // A sensor reports what it sees by sending us a number. It is from 0 to 1023
    and it represents how much voltage the sensor is
    // sending back to the Arduino. The amount it will send back changes
    depending on what the sensor is "seeing".
    // We can see these numbers by using a tool called the Serial Monitor.
    // The infrared (IR) sensor should send a LOW number if it sees something in
    front of it and a HIGHER number if it doesn't.

    int sensorReading = analogRead(A0); // Read the values that the sensor on pin
    A0 is reporting,
    Serial.println(sensorReading);      // and show them in the Serial Monitor
    window so we can see them.

```

```
the Serial Monitor button // If you upload this code and click on
                             // (the magnifying glass) in the upper
right corner,
                             // you will see the values that the
sensor is reporting to you.
                             // Play with your sensor and see how the
numbers change under different conditions.
```

```
//////// IF STATEMENTS: USING OUR SENSOR READING TO MAKE SOMETHING HAPPEN
// An if statement lets us run some code only IF a condition is met.
// We can use this with our sensor data to make something happen IF our
infrared sensor detects something close to it.
// IF our condition in the ( ) gets met, then the code in the {
} will run.
// IF our condition in the ( ) DOESN'T get met, then the code in the {
} WON'T run.
```

```
if (sensorReading < 100) { // If the sensor is giving values less
than 100, we will do all the code between these { }.
```

```
    sweep(my servo, 180, 5); // First, we'll sweep our 180-degree
servo back and forth.
    delay(500);
    sweep(my servo, 0, 5);
    delay(500);
```

```
////////// 360-DEGREE SERVO CONTROL
```

```
// A 360-degree servo works a little differently from a 180-degree one. One
difference is that it can spin in a full circle.
```

```
// Also, we can't control precisely WHERE it moves, but we can control what
DIRECTION it spins in and the SPEED it spins at.
```

```
// We can use the command my servo2.write() plus a number to control its
direction and speed.
```

```
// CLOCKWISE DIRECTION: Use numbers from 0 to around 90, with 0 being the
fastest speed and 90 being pretty slow.
```

```
// COUNTER-CLOCKWISE: Use numbers from around 100 to 180, with 100 being
pretty slow and 180 being the fastest speed.
```

```
// (If you are not using a 360-degree servo in pin 10, you can replace these
commands with the sweep() commands that you already know.)
```

```
    my servo2.attach(10); // Activate the motor so it can spin.
    my servo2.write(40); // Spin the 360 servo clockwise pretty slowly
    delay(500); // for 1/4th of a second before doing anything
else,
```

```

        myservo2.write(150);    // then spin it in the other direction pretty
slowly
        delay(500);             // for 1/4th of a second before doing anything
else.
        myservo2.detach();      // Deactivate the motor so that it won't spin
forever.

    }                            // This is the end of our if statement!

}                                // And this is the end of our loop().

// Don't change the code under here or your program might break.
// This is some custom code to make the sweeping motion easier to use,
// so that we don't have to use more complicated concepts in our main program.

void sweep(Servo servoName, int targetAngle, int sweepSpeed) {
    int currentPos = servoName.read();

    if (currentPos < targetAngle) {
        for (int i = currentPos; i < targetAngle; i++) {
            servoName.write(i);
            delay(sweepSpeed);
        }
    }

    if (currentPos > targetAngle) {
        for (int i = currentPos; i > targetAngle; i--) {
            servoName.write(i);
            delay(sweepSpeed);
        }
    }
}

```