THE GARDUINO GARDEN CONTROLLER
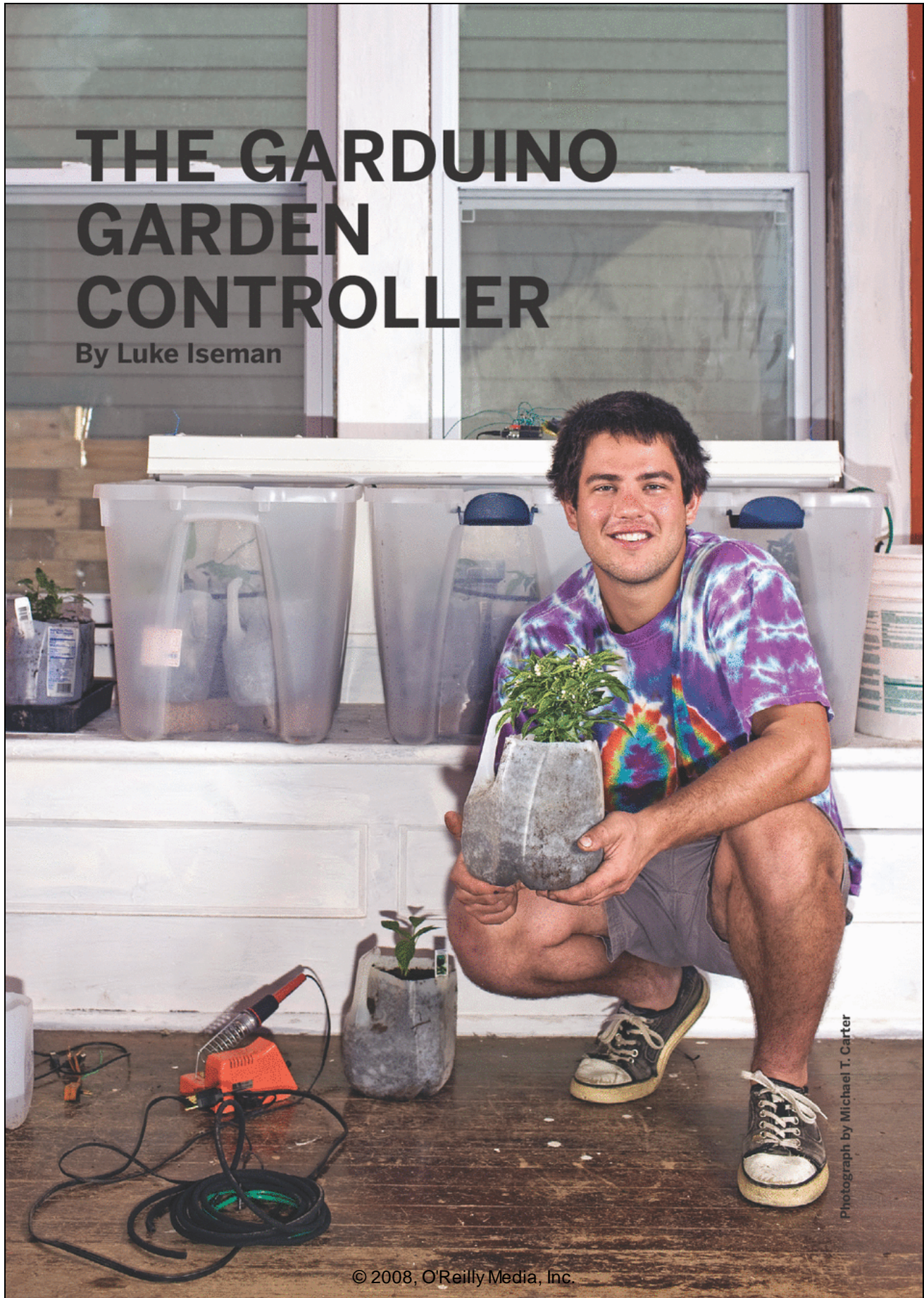
By Luke Iseman

Photograph by Michael T. Carter

GEEKED-OUT GARDENING

I wanted to start gardening, but I knew I wouldn't keep up the regular schedule of watering the plants and making sure they got enough light. So I recruited a microprocessor and a suite of sensors to help with these tasks.

My Garduino garden controller uses an Arduino micro-controller to run my indoor garden, watering the plants only when they're thirsty, turning on supplemental lights based on how much natural sunlight is received, and alerting me if the temperature drops below a plant-healthy level.

For sensors, the Garduino uses an inexpensive photocell (light), thermistor (temperature), and a pair of galvanized nails (moisture).

You can use a Garduino to experiment and learn what works best in your garden.

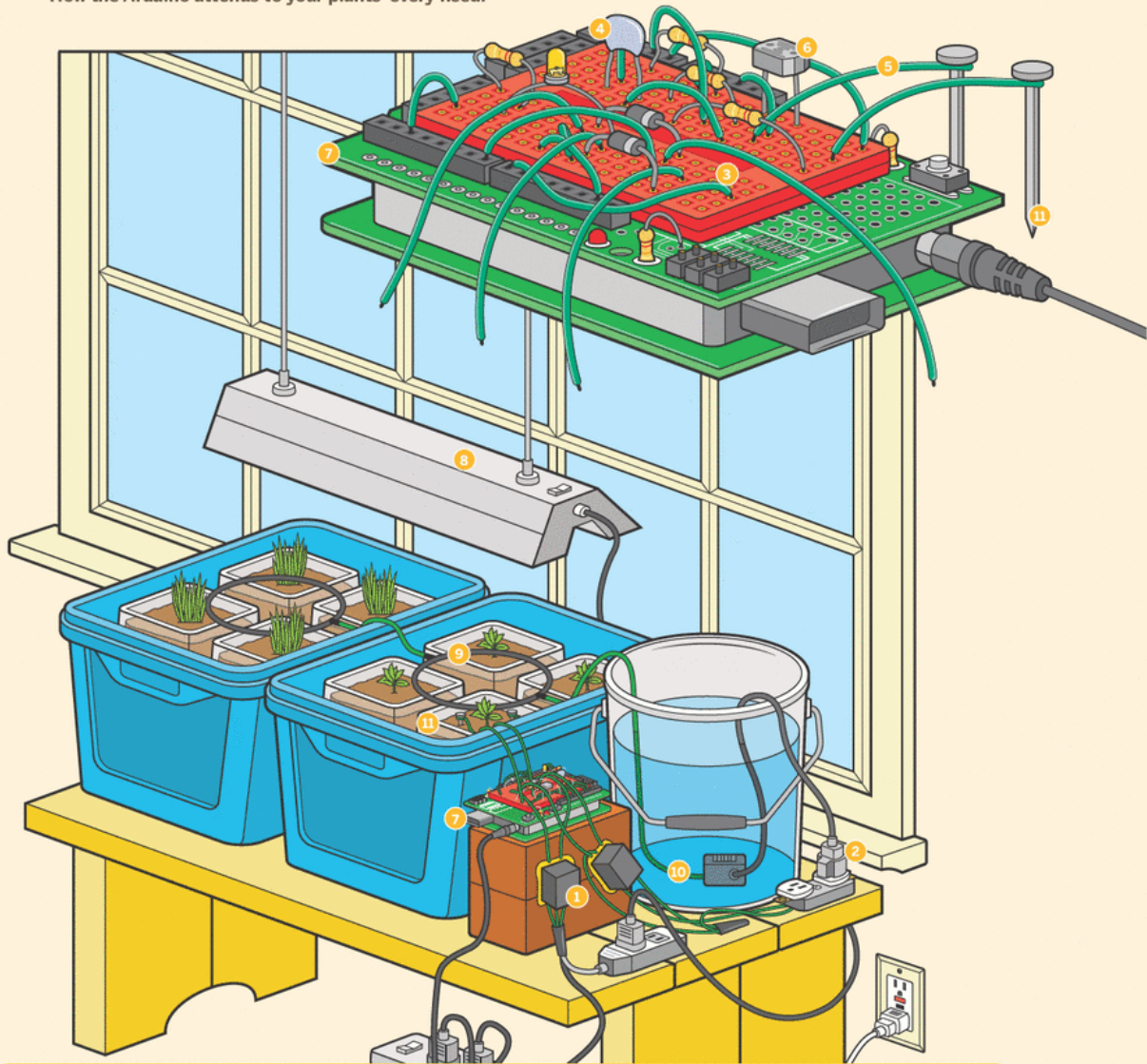**Set up: p.93   Make it: p.94   Use it: p.101**

Luke Iseman builds open source hardware, including pedicabs, pallet furniture, and chicken tractors, in Austin, Texas. His projects are available at dirtnail.com

Make: 91

MICROCONTROLLER-ASSISTED GARDENING

How the Arduino attends to your plants' every need.
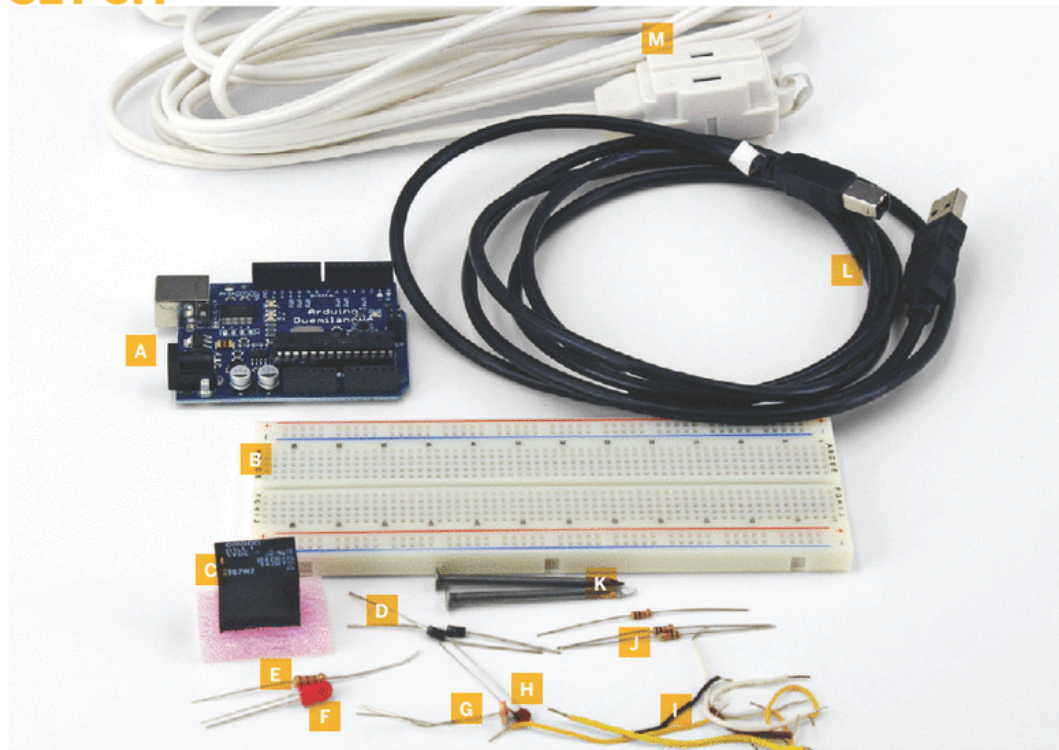

Illustration by Timmy Kucynda

INDOOR GARDEN

1 Relays let you control AC power (lights and pumps) with the Arduino.

2 Here we're controlling power to the pump with a relay wired to the Arduino.

3 The relays connect to our Arduino with diodes to prevent it from frying.

4 Thermistor: resistance changes with temperature.

5 These wires lead to nails stuck in the soil: our moisture sensor. Resistance between the nails, through the soil, changes with soil moisture.

6 Photocell: resistance changes with light intensity.

7 Garduino circuit board (Arduino + sensors).

8 The Garduino turns on the fluorescent light to make sure the plants receive the right amount of light (16 hours daily). The photocell plus Arduino measure how much natural sunlight is received and turn on the supplemental lights to make up any difference.

9 The black irrigation tubing has tiny holes that allow water to reach the plants when the pump is turned on.

10 The pump is submerged in a bucket of water.

11 Ordinary galvanized nails are used to sense soil moisture.

SET UP.

WARNING: This project involves using electricity near water. We recommend connecting to a GFCI outlet.

Photograph by Ed Troxell

MATERIALS

All electronics can be purchased at the Maker Shed (makershed.com) and Jameco (jameco.com), and everything else can be found at your local hardware store. See makezine.com/18/garduino for direct links to purchase the parts online. Total cost, including the Arduino, was about $150.

[A] Arduino microcontroller I used an Arduino Duemilanove, but any should work.

[B] Circuit board You can pack everything onto a ProtoShield (shown on previous page and available at the Maker Shed) but the following pages show a solderless breadboard for clarity.

[C] Omron G5LE-1 relays (2)

[D] 1N4004 diodes (2)

[E] 220Ω resistor for the LED

[F] Standard LED Any you'd use with an Arduino will do.

[G] Photocell

[H] 10kΩ thermistor

[I] 22-gauge wire, solid core and stranded, several feet

[J] 10kΩ resistors (3)

[K] Galvanized nails, 1"–4" long (2)

[L] USB cable

[M] AC extension cords (2)

[NOT SHOWN]

Plastic milk jugs for planting in. Use as many as you'd like. I used about 30.

Clear plastic storage containers, 28gal You'll need 1 for every 6 milk jugs.

Bricks or other spacers to raise the milk jugs at least 1" off the bottom of the bin. You'll need about 5 for each storage container.

Seeds preferably for things you'd like to eat. Swiss chard is an easy starter plant.

48" fluorescent light fixture

48" fluorescent tube "grow light" I used the OttLite, but any tube marketed for plant growth should be fine.

Soil mixture I used Mel's Mix, as recommended in Mel Bartholomew's *Square Foot Gardening*. It consists of $1/3$ peat moss, $1/3$ coarse vermiculite, and $1/3$ mixed compost, with the mixed compost coming from at least 6 different sources. You can use whatever works for your plants.

Clean-water pump A small, cheap one is fine; I used a mini submersible pump from Harbor Freight (item #45303, available online at www.harborfreight.com).

Micro soaker hose kit also from Harbor Freight (item #65015). Or you can use bike inner tubes and poke holes.
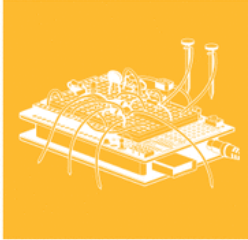
5gal bucket

Funnel I used a cut milk jug

Make: **93**

MAKE IT.

HOOK UP YOUR MICROCONTROLLER GARDEN

START ››

Time: A Weekend   **Complexity: Moderate**

This project is ambitious for a first Arduino undertaking. I recommend completing at least the first few lessons of an Arduino tutorial before attempting this. There's a great one at ladyada.net/learn/arduino.

1. PLANT YOUR GARDEN

Sprout your seeds before planting them, or buy started plants. I planted a variety of vegetables in milk jugs with the tops cut off, with holes in the bottom to allow drainage, and a surrounding plastic storage container to catch water as it drained out.

2. MAKE MOISTURE SENSORS

2a. Cut 2 pieces of wire, each 2' long, and strip ½" off the ends.

2b. Wrap 1 end of each wire around the head of each nail.

2c. Cover the wire-nail connection with a generous amount of solder.
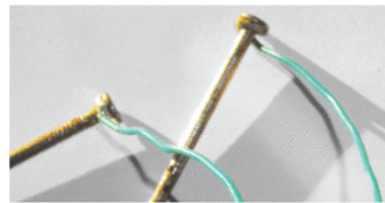
3. CONNECT MOISTURE SENSORS TO THE ARDUINO

You can tell when your soil needs water by measuring the resistance between the 2 nails stuck in the dirt. The more water in the soil, the more conductive it is.

3a. Connect a wire between ground on your Arduino and the ground (−) column on your breadboard. You'll use this column on the breadboard as ground for the rest of the circuit.

3b. Connect a wire between +5V on your Arduino and the positive (+) column on your breadboard. You'll use this column as the positive voltage connection for the rest of the circuit.

Photograph by Luke Iseman

3c. Connect one of the moisture sensors to +5V on the breadboard.

3d. Connect the other moisture sensor to a new row on the breadboard.

3e. Connect a 10kΩ resistor to the same row as the moisture sensor and also to a new row.

3f. Connect a wire from analog input 0 on your Arduino to the same row as the resistor and moisture sensor.

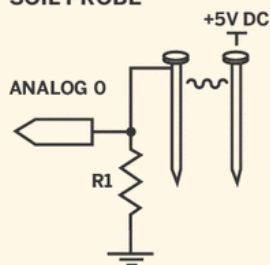**3g.** Connect the other end of the resistor (in the new row) to ground.
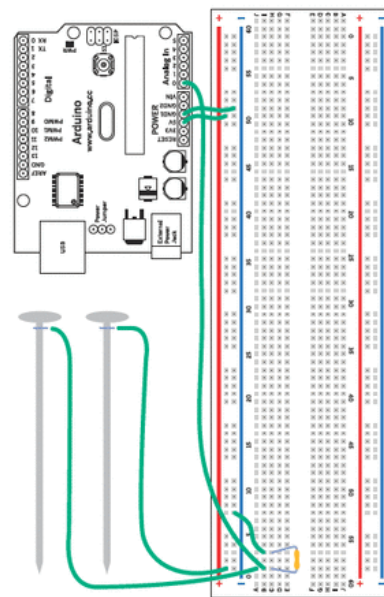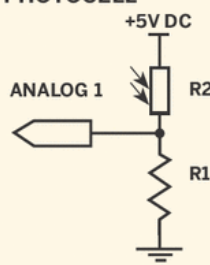
MICROCONTROLLER INPUTS AND OUTPUTS

Here's how the Arduino's I/O pins connect to the Garduino's sensors, relays, and LED.

KEY		R3	Thermistor
R1	10kΩ	R4	220Ω
R2	Photocell	D1	1N4004

SOIL PROBE

+5V DC

ANALOG 0

R1

PHOTOCELL

+5V DC

ANALOG 1   R2

R1

RELAYS

DIGITAL I/O 7, 8

120V PLUG

D1

EXTENSION CORD

2

1

RELAY

5

4

THERMISTOR

+5V DC

ANALOG 2   R3

R1

LED

DIGITAL I/O 2

R4
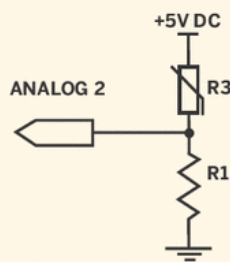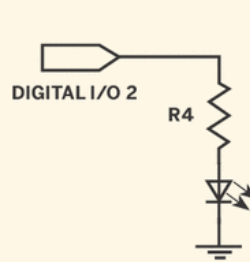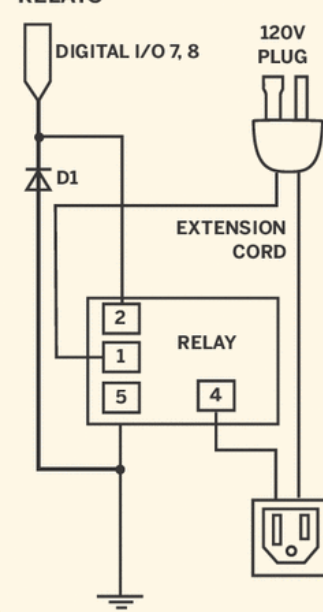
Illustrations by Gerry Arrington

Make: 95

4. LOAD THE SENSOR TEST CODE

Make sure that the moisture sensor works by connecting your Arduino to a computer and entering the sensor test code below (also at makezine.com/18/garduino). When you touch the nails together, the moisture value should read ~985; when they're not touching, the moisture value should be 0.

NOTE: If you haven't used your Arduino before, you need to connect it via its USB cable to your computer, then launch the Arduino development application (free download at arduino.cc), enter the code, and then upload it to the board. Select the serial monitor to see the output.

CODE:

```
        int moistureSensor = 0;
        int lightSensor = 1;
        int tempSensor = 2;
        int moisture_val;
        int light_val;
        int temp_val;

        void setup() {
        Serial.begin(9600); //open serial port
        }

void loop() {
        moisture_val = analogRead(moistureSensor); // read the value from the moisture-sensing probes
        Serial.print("moisture sensor reads ");
        Serial.println( moisture_val );
        delay(500);
        light_val = analogRead(lightSensor); // read the value from the photosensor
        Serial.print("light sensor reads ");
        Serial.println( light_val );
        delay(500);
        temp_val = analogRead(tempSensor); // read the value from the thermistor
        Serial.print("temp sensor reads ");
        Serial.println( temp_val );
        delay(1000);

        }
```

5. ADD THE LIGHT SENSOR

5a. First, connect the photocell to 2 new rows on the breadboard.

5b. Connect a wire between one row that the photocell touches and the positive column.

5c. Connect a 10kΩ resistor to the other row that the photocell touches and to a new row.

5d. Connect a wire between the photocell-resistor row and analog input 1 on your Arduino.

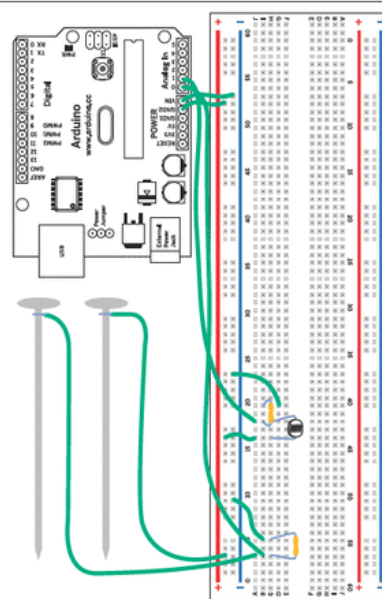**5e.** Connect the other end of the resistor to ground.

5f. Test your light sensor by connecting your Arduino to your computer and monitoring the serial output. I measured the following values:

　Indirect sun: 949
　Ambient indoor light at night: 658
　Ambient indoor light at night, with a
　　hand casting shadow over the sensor: 343

6. CONNECT THE TEMPERATURE SENSOR

6a. Connect the thermistor to 2 new rows on the breadboard.

6b. Connect a wire between one row that the thermistor touches and to the positive column.

6c. Connect the last of the 10kΩ resistors to the other row the thermistor touches and to a new row.

6d. Connect a wire between the thermistor-resistor row and analog input 2 on your Arduino.

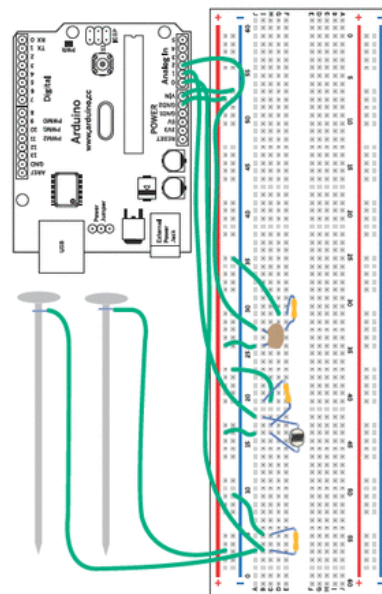**6e.** Connect the other end of the resistor to ground.

6f. Test your temperature sensor by connecting your Arduino to your computer and monitoring the serial output. I measured the following values:

　61°F = 901
　90°F = 949
　51°F = 877
　32°F = 796

Make: 97

PROJECTS: **GARDUINO**

7. SPLICE RELAYS INTO THE LIGHT/PUMP POWER CORDS

Now we work with the heavy lifters: our relay setups. These will turn the lights and pumps on and off, in response to hours of sunlight received and soil moisture.

7a. Cut four 10" lengths of 22-gauge wire and strip ½" off each end.

7b. Look at your extension cord's plug: one prong is larger than the other. Split the 2 wires of the cord apart, then cut the wire that runs to the smaller prong, and strip 1" off each side.

TIP: The correct wire is the one without ridges running along its length. Don't worry if you cut both wires; you can just splice the other wire back together.

**7c.** Solder a 22-gauge wire to each side of the split cord wire.

7d. Solder the 22-gauge wire that runs to the extension cord's receptacle to the lower right lead of the relay (it should be labeled "4" on the bottom of the relay).

7e. Solder the 22-gauge wire that runs to the extension cord's plug to the middle left lead of the relay (labeled "1").

7f. Connect a 22-gauge wire to each of the 2 other leads on the left side of the relay (labeled "2" and "5"). Optionally, you can cover the relay's bottom side with hot glue to strengthen all 4 connections.

7g. Wrap both connections to the extension cord in electrical tape or heat-shrink tubing. Congratulations, you've completed your relay-cord setup.

7h. Repeat Steps 7a–7g with another relay and extension cord to create the second relay-cord setup.
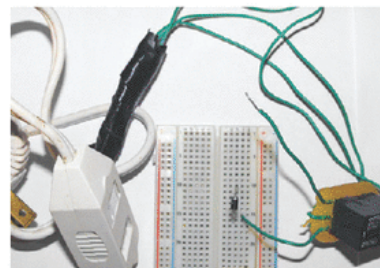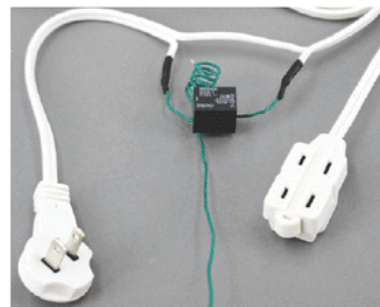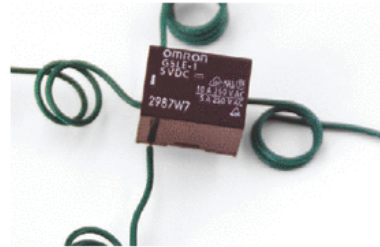
Photography by Ed Troxell; bottom photograph by Luke Iseman

98   **Make:** Volume 18

© 2008, O'Reilly Media, Inc.

8. CONNECT RELAYS AND AN LED TO THE ARDUINO

8a. Connect one of your diodes to 2 unused rows on the breadboard.

8b. Connect the bottom left lead (pin 2) of your relay (looking at it from the top, with the leads down) to the positive lead of your diode — the end that does not have a band on it.

8c. Connect the upper left lead (pin 5) of your relay to the negative lead of the diode — the end marked with a band.

8d. Connect a wire between ground on your Arduino and the ground column on your breadboard (if you're using the same breadboard for relays and sensors, just use the ground column you've already created).

8e. Connect the row containing the negative lead of your diode (the end with the band) and the upper left lead of your relay to digital input/output 7 on your Arduino.

8f. Connect the positive lead of the diode to your ground column. That's it for connecting the first relay.

8g. Now choose 2 new unused rows and repeat Steps 8a–8f to connect the second diode and relay as you did the first, except this one goes to digital input/output 8 on your Arduino.

8h. Connect the 220Ω resistor to 2 unused rows. Connect the LED's long leg (+) to either end of the resistor, and its short leg (−) to ground. Connect the other end of the resistor to digital input/output 2 on your Arduino.

to extension cord
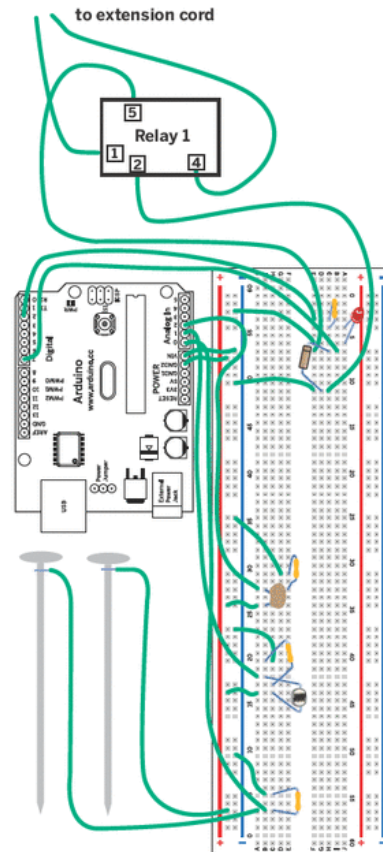
Relay 1

9. SET UP THE LIGHTING SYSTEM

An easy step here: after checking that your light fixture is working, plug it into the relay cord that will control it. I simply rested the fixture on top of the outer plastic bins, but feel free to get fancy. Optimum distance from these fluorescents for the light intensity you want is just a few inches, so make sure you get them up close, personal, and adjustable as the plants grow.

Photograph by Michael T. Carter

Make: 99

10. SET UP THE WATERING SYSTEM

You've got a wide variety of options here. Here's how mine works:

» A small pump is submerged in a 5gal bucket water source.

» A relay controlled by moisture sensors activates the pump to move water from the bucket to a milk-jug funnel.

» Gravity moves water from the funnel down to the soaker hoses, which drip into the plants.

I used a mini soaker hose kit from Harbor Freight Tools to assemble rings that drip into all the plant containers. If you make a setup like this, be sure to elevate the bucket on a crate or something else; moving water inches instead of feet vertically will greatly reduce the strain on your pump.

I initially tried using just a sprinkler valve mounted to the bottom of a bucket, without the pump. But gravity provided only enough pressure for the slightest trickle. I thought about connecting the sprinkler valve right into my plumbing, but I worried that the chlorine content of water straight from the faucet would be bad for my plants (chlorine evaporates from water within something like 24 hours).

A better version of this would be to use 2 buckets, with water coming from a sprinkler valve connected to house plumbing going into one bucket, being held there 24-plus hours, and then moving into the second, plant-feeding bucket.

11. PROGRAM YOUR GARDUINO

Lastly, you need to program your Garduino to run the garden. Because temperature and soil moisture are dealt with as constants (i.e., always turn on the LED if temperature is below a certain value, always turn on the water if moisture is below a certain value) they're simple to deal with.

Light is more complicated: you want to keep track of how much light your plants are getting, so that natural light plus supplemental light always equals optimum light time (in my case, I chose 14 hours daily). To do this, I used the *DateTime* Arduino library.

Photography by Michael T. Carter

FINISH

NOW GO USE IT »

USE IT.

SWITCHING TO ...
GARDENING COMPUTER

REAP YOUR HARVEST

Check your seed packets (you saved them, right?) to see how many weeks until your plants should be ready for harvest. But don't be surprised if they're ready sooner than that! If they seem to be growing too slowly, check your watering and lighting routines.

EVALUATE YOUR DATA

As currently implemented, Garduino needs to be hooked up to a computer that's monitoring serial output to obtain more meaningful data to share. With additional work, it's possible to store data on a USB drive. But for now, monitor the serial output in the Arduino environment to evaluate your Garduino's performance.
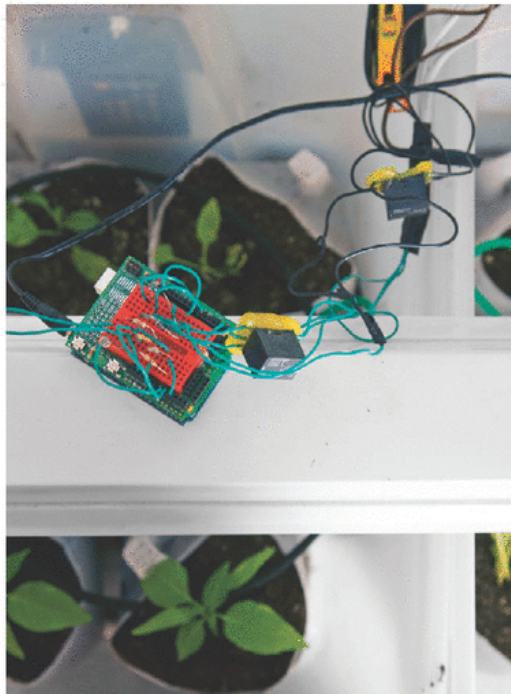
IMPROVE AWAY!

I don't expect this beta Garduino to get everyone gardening and save the world; that's an exercise for readers to solve with their improvements. But here are some initial ideas:

» Use pulsing red and blue LEDs for an ultra-efficient lighting system (see screwdecaf.cx/sept. html for Mikey Sklar's version).

» Figure out what times of night your utility charges lower rates for electricity, and turn the lights on during those times only.

» Build a pH probe and fine-tune your soil acidity for different plants.

» Add a relay-controlled heater to keep a greenhouse version above a minimum desired temperature.

» Add a battery and solar panel to take the whole system off-grid.

» Use an irrigation valve instead of a pump to water your larger, outdoor garden, and add some modified solar garden lights for additional lighting.

If many people start recording the efficiency and convenience of this automated approach to gardening, then maybe we can even grow more food of better quality with less energy. Happy Garduino-ing!

➕ You can find the complete code at makezine/18/garduino. I'll add links to better versions as readers create them.

Make: 101