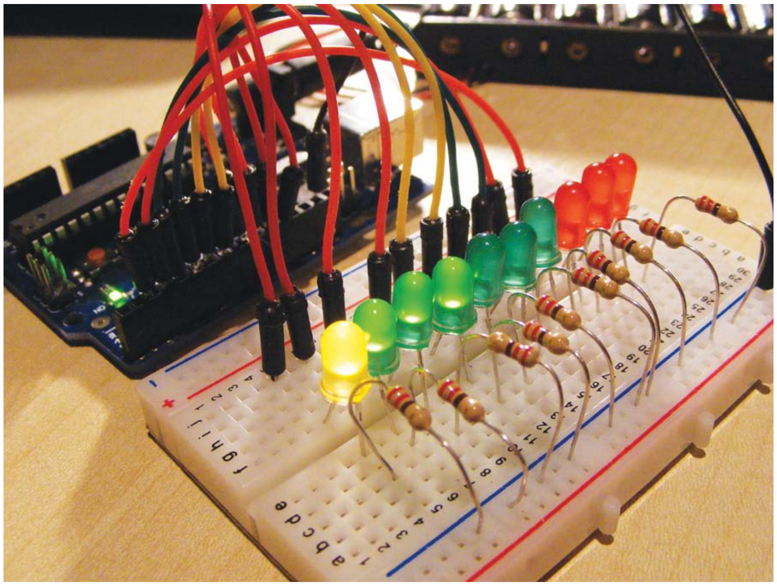
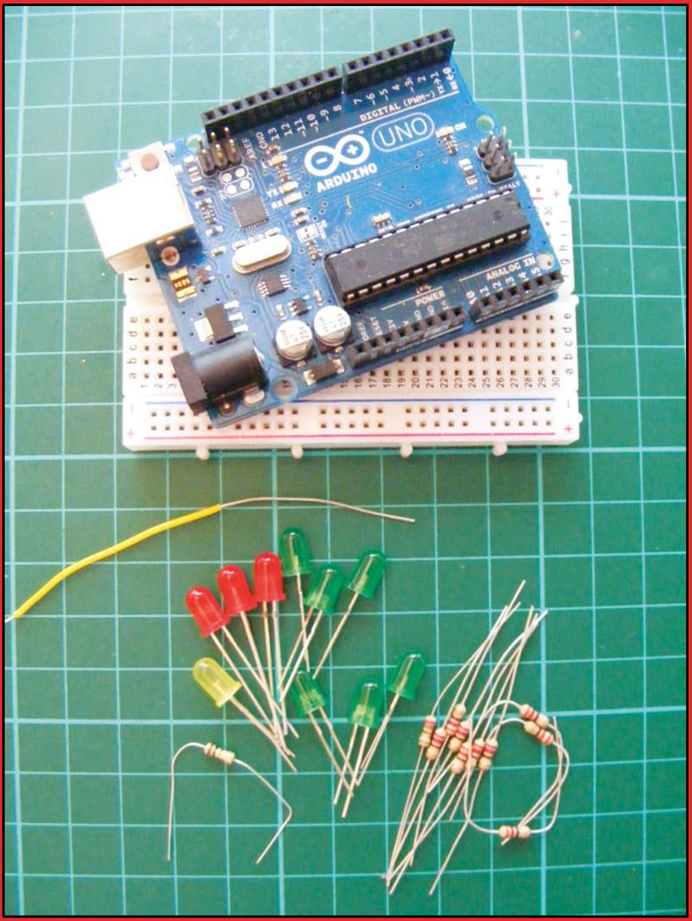
PROJECT 6: GHOST DETECTOR

WHO WOULDN'T WANT TO MAKE A GHOST DETECTOR? THIS IS A REALLY SIMPLE PROJECT THAT DOESN'T TAKE LONG TO PUT TOGETHER, SO YOU CAN START DETECTING GHOSTS RIGHT AWAY.





PARTS REQUIRED

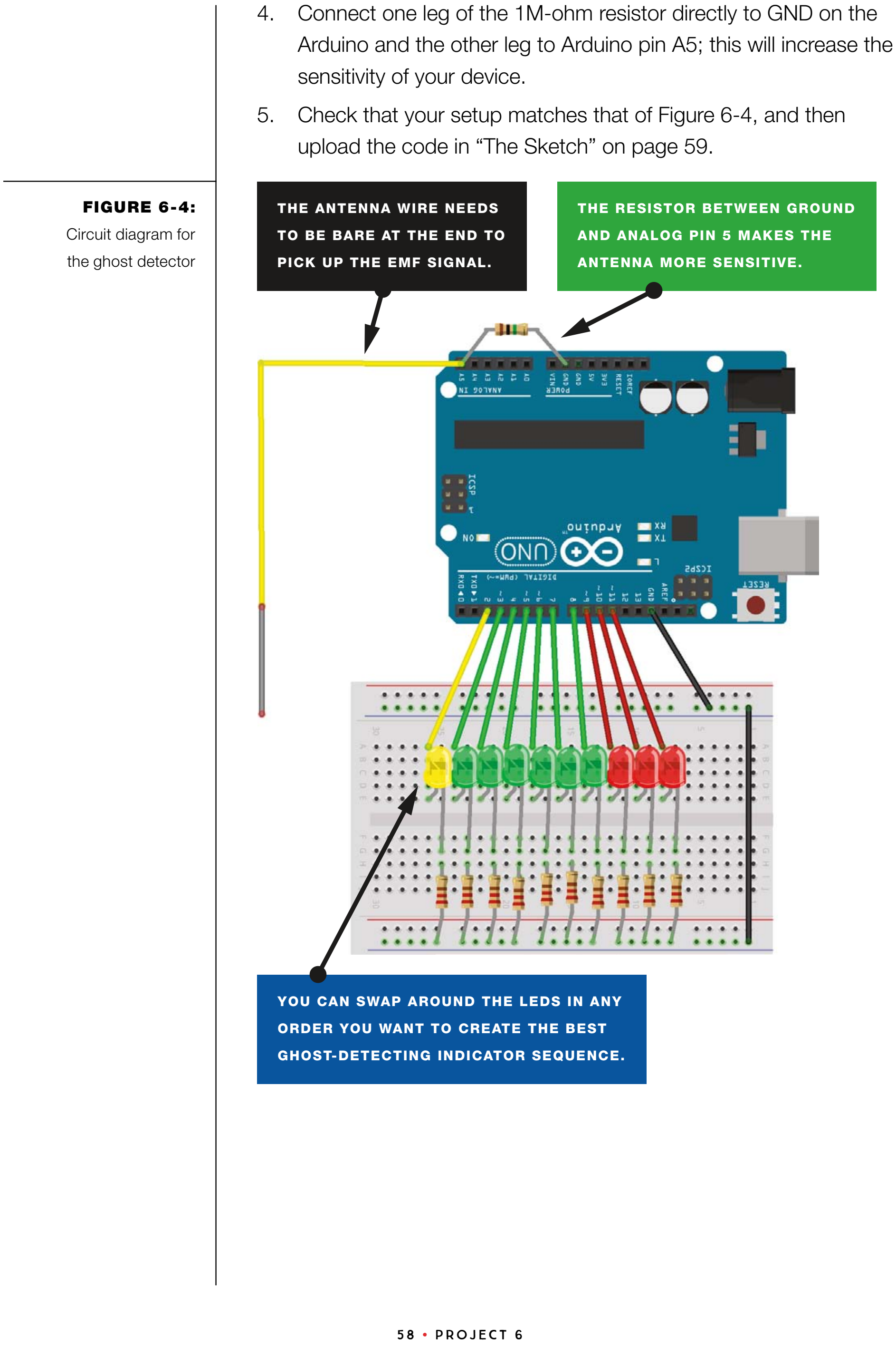
- Arduino board
- Breadboard
- Jumper wires
- 3 red LEDs
- 1 yellow LED
- 6 green LEDs
- 10 220-ohm resistors
- 20-centimeter length of single-core wire
- 1M-ohm resistor

1 Setup

First you will need to download, unzip, and install the Arduino Integrated Development Environment (IDE) from `https://www.arduino.cc/en/Main/Donate` (does not need admin privileges).

2 Building The Circuit

The schematic for the circuit you will be building is below.



4. Connect one leg of the 1M-ohm resistor directly to GND on the Arduino and the other leg to Arduino pin A5; this will increase the sensitivity of your device.
5. Check that your setup matches that of Figure 6-4, and then upload the code in “The Sketch” on page 59.

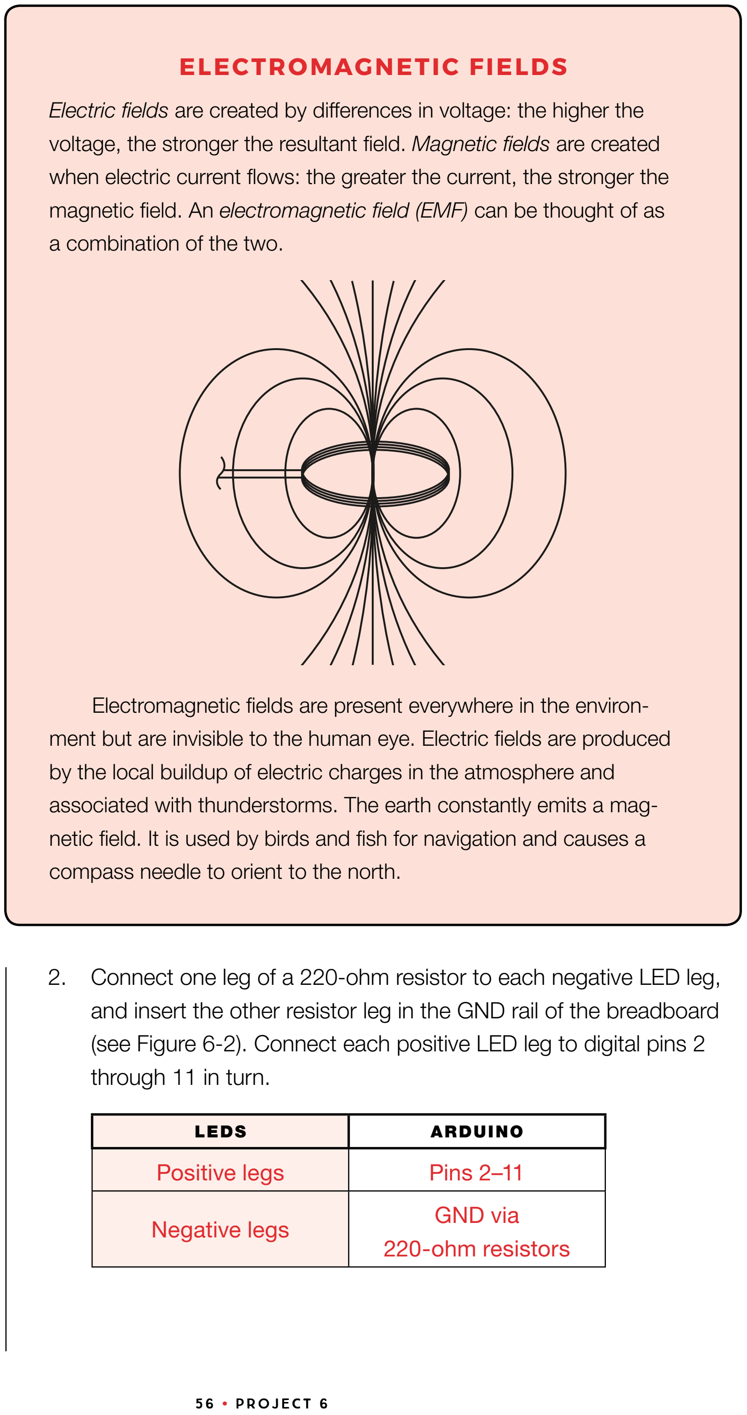
FIGURE 6-4: Circuit diagram for the ghost detector

THE ANTENNA WIRE NEEDS TO BE BARE AT THE END TO PICK UP THE EMF SIGNAL.

THE RESISTOR BETWEEN GROUND AND ANALOG PIN 5 MAKES THE ANTENNA MORE SENSITIVE.

YOU CAN SWAP AROUND THE LEDS IN ANY ORDER YOU WANT TO CREATE THE BEST GHOST-DETECTING INDICATOR SEQUENCE.

58 • PROJECT 6



ELECTROMAGNETIC FIELDS

Electric fields are created by differences in voltage: the higher the voltage, the stronger the resultant field. *Magnetic fields* are created when electric current flows: the greater the current, the stronger the magnetic field. An *electromagnetic field (EMF)* can be thought of as a combination of the two.

Electromagnetic fields are present everywhere in the environment but are invisible to the human eye. Electric fields are produced by the local buildup of electric charges in the atmosphere and associated with thunderstorms. The earth constantly emits a magnetic field. It is used by birds and fish for navigation and causes a compass needle to orient to the north.

2. Connect one leg of a 220-ohm resistor to each negative LED leg, and insert the other resistor leg in the GND rail of the breadboard (see Figure 6-2). Connect each positive LED leg to digital pins 2 through 11 in turn.

LEDS	ARDUINO
Positive legs	Pins 2–11
Negative legs	GND via 220-ohm resistors

PAUSE

What is a circuit? Why are resistors needed? Why can resistors be placed in any direction in a circuit, but LEDs only work if they are placed in a certain direction.

3 Programming the Arduino

The base code for programming the Arduino is provided. Using the Arduino IDE, open the `.ino` file.

`PAUSE`

The IDE allows you to do 4 things: edit the code, verify the code is correct (i.e. does not contain syntax errors), upload the code to the Arduino, and view the diagnostic output of things as they run on the Arduino.

The bare wire picks up the signal from the electromagnetic fields in the atmosphere and sends a value between 0 and 1023 to the Arduino (this is done by a special kind of circuit, an Analog-to-Digital Converter, or ADC for short). The base code evaluates the reading from the analog pin to determine how many LEDs should be on or off in sequence to indicate the strength of the electromagnetic signal. For example, 1023 would be the highest value, so all LEDs would be lit; a reading of 550 would light five LEDs. The sketch loops to continuously read the analog input, and the LED lights constantly move to show the reading. If you find that the EMF readings set off your LED sequence to the maxi- mum level every time, reduce the senseLimit value to compensate. The sketch takes an average of 25 number readings each time it loops through, and uses the average from those readings to mitigate big fluctuations that may cause the LEDs to light up too quickly.

`PAUSE`

Can you map each part of the description above to lines in the code?

4 Tuning The Code

You may notice that your detector is not that sensitive, or that it is TOO sensitive (i.e. all LEDs are always on). This can be fixed in the code by changing the thresholds that each LED turns on at. For example, if one additional LED turns on every time the reading goes up by 100, and that is not sensitive enough, you could make one additional LED turn on every time the reading goes up by 50 to make it more sensitive.

5 Extending The Code

Once you have the ghost detector working try adding some sounds via one or more buzzers that beep at increasing speeds or different patterns depending on the reading. The buzzers we have are called *piezo* buzzers. They are not like a regular speaker that you might think of. It uses a material that's piezoelectric, which means it changes shape when you apply electricity to it. By adhering a piezo-electric disc to a thin metal plate, and then applying electricity, we can bend the metal back and forth, which in turn creates noise. The faster you bend the material, the higher the pitch of the noise that's produced. This rate is called frequency. Again, the higher the frequency, the higher the pitch of the noise we hear. So basically, by shocking the plate over and over really fast, we can make noise.

You should following the same idea as the LED circuits you built:

1. Assign a new pin on the Arduino to your new buzzer (such as 12 or 13), and add this definition to the code.

2. The buzzers are similar to LEDs, in that they only work when plugged in a certain way, and they need a resistor between their negative end and ground.

3. Once you have the circuit built, you are ready to add the buzzer trigger into the code! The functions you will need are:

   `Tone(int pin, int frequency)`, which controls which pin gets the buzzer signal and what the frequency of the sound will be.

`noTone(int pin)`, which turns off the buzzing signal on the specified pin.

Can you figure out how to play a tone when the maximum sensor reading is observed (and all the LEDs are on) ? What about playing a different tone or pattern depending on what the sensor reading is (i.e. going up in pitch the more LEDs are on) ?