13

Web-Interacting Robots

One of the features that most of the robots in this book share is that you control them from a location right next to them. What if we built a robot that could be controlled from around the world? In this chapter, you learn about various possibilities. For instance, there are robots that interact with Twitter, either making posts or searching for ones containing a specific keyword. Other robots work almost as Internet-connected weather stations, sending data back to your computer to let you know what's going on.

In addition to covering some of these possibilities, I'll show you how to give the Arduino-equipped dart gun from Chapter 7, "Harnessing Infrared," the capability to send you an SMS (a text message) when it fires (see Figure 13.1).

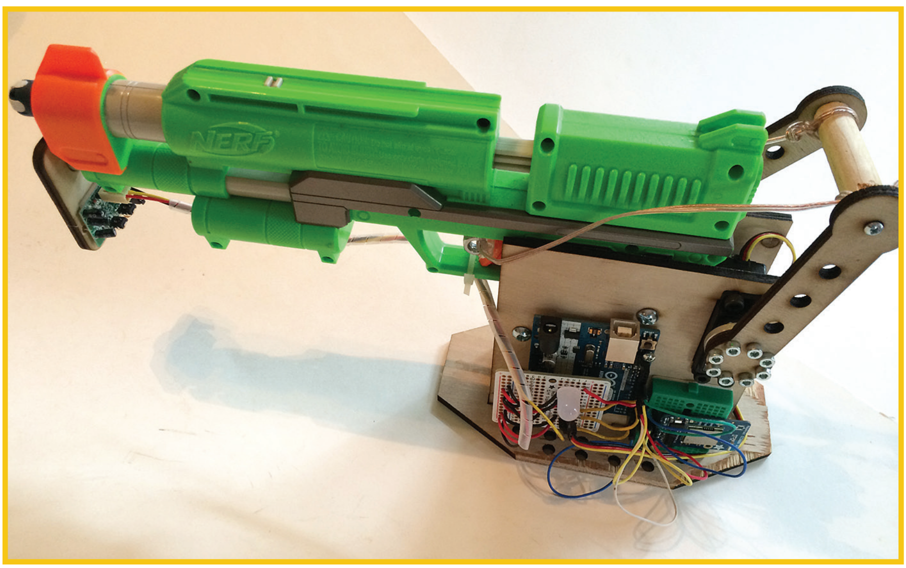

FIGURE 13.1 The Dart Texter improves on a classic with the help of a Wi-Fi card.

Types of Web-Interacting Robots

What exactly can you do with a web-connected robot? It turns out, quite a lot. In this section, I cover some of the possibilities.

Sniffers

Sniffer robots search the Internet for keywords and then do *something* with that information. For example, there are gumball machines that dispense only if a certain keyword appears on Twitter. Some organizations have a liquid-crystal display (LCD) screen scrolling with recent keyword mentions.

One advantage to this system is that it can be controlled remotely with no additional hardware needed—merely a Twitter account! The downside is the same, in that anyone can hijack your robot. Maybe that's a good thing!

Autotweeters

Autotweeters send out a Twitter message (tweet) when certain events take place. The classic example is a kegbot (kegbot.org), which monitors and controls a beer keg. It dispenses beer, measures its temperature, accepts payment, and sends out a tweet when it pours a pint.

Recently, Twitter changed the way it allows external devices to interact with it, and some old ways of sending out tweets no longer work. However, as long as Twitter is as popular as it is, there will be a way to interact with it.

Telepresence

As an introvert, I always dreamed of a robot that could go to family functions and school dances in my stead. I could sit back and direct the robot to roll around, with a camera and screen allowing me to interact with guests. It turns out that there is a category of robot that does that, called *telepresence bots*.

At its simplest, a telepresence robot is a rolling robot (like the Cruiser) that carries an iPad or other tablet around, with video conferencing running. The home base can include a desktop PC that controls the robot remotely. Some clever hacks have been done to piggyback the control functionality on top of the video conferencing feed; one bot creator used black-and-white cards to trigger a light sensor positioned at the corner of the screen.

Interactive Robots

Some robots, like the gumball machine I mentioned earlier in this section, are designed to be controlled remotely. There are robot arms that can be directed from websites, with web-cams positioned to show what's happening. A whole subcategory involves pet toys that let you play with your cat or dog from a remote location.

Some of them are games, like chess boards with players separated by miles, whereas others are much simpler, like the equivalent of robot soccer. Typically, interactive robots (like a lot of robots described in this section) are more for fun than to accomplish a serious task, but it doesn't have to be that way!

Home Automation

One category of web-connected robots that arguably is serious is that of home automation. Imagine the lure of being able to adjust your blinds with your smartphone or program a lighting sequence for when you're on vacation. Imagine a robot that can send you data on the status of the alarm system and let you know whether the furnace is on.

Home automation robots also include agricultural automation, such as greenhouse controllers, which turn on fans and trigger water valves, measure temperature and soil moisture, and turn on grow lights as needed.

Sensor Nets

One of the most intriguing aspects of all these sensors and microcontrollers connected across the Internet is the possibility that you can have them measure the same thing and publish the data to the web, giving you a real-time map of the area.

Sensor nets were established by amateurs during the Fukushima Daiichi nuclear accident in 2011. Government officials, these amateurs contend, were too slow to release radiation numbers for the area, so they formed their own network of Geiger counters (see Figure 13.2) and published up-to-date counts online. You can learn more about the organization at safecast.org.

FIGURE 13.2 This array of radio-linked Geiger counters upload data to the Internet.
Credit: Chris Wang

Hardware

It's very likely that you'll need additional hardware to connect your Arduino to the Internet. The following sections discuss some examples of add-on products that will help.

Arduino Ethernet Shield

Arduino's Ethernet Shield (seen in Figure 13.3) is one of the earliest products aimed at bringing Internet connectivity to the Arduino world. It is equipped with a standard RJ45 Ethernet jack and a secure digital (SD) card slot. The Internet functionality uses only four pins, so you're often able to stack another shield on top.

Because it's an established product, there are a lot of example sketches and tutorials around, including information on how to make a web server, how to publish sensor data online, and how to host a chat via Telnet—a way of sending raw text over the Internet.

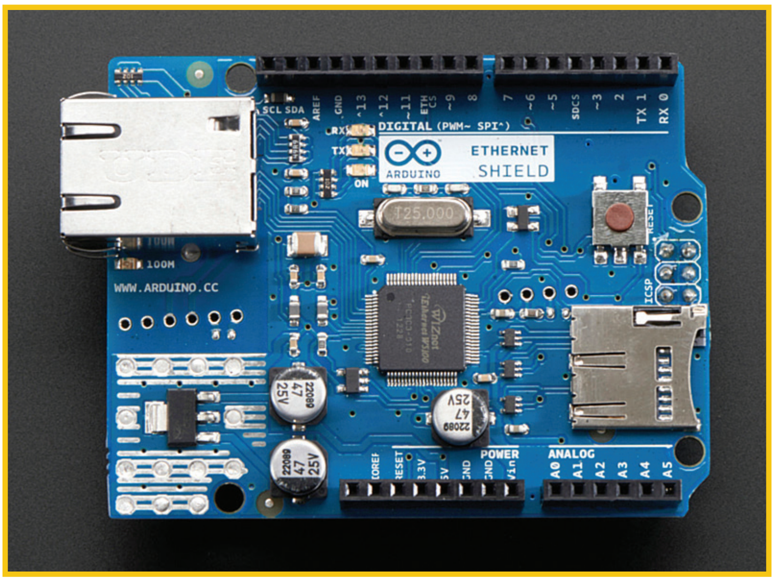

FIGURE 13.3 A lot of code has been written for the Arduino Ethernet Shield.
Credit: Adafruit Industries.

Adafruit CC3000 Breakout

The CC3000 is a relatively recent creation by Texas Instruments that drastically simplifies and economizes connecting to Wi-Fi via an Arduino. It supports 802.11b/g Wi-Fi protocols, as well as protected access security protocols, such as WPA and WEP.

Adafruit sells the CC3000 in a variety of configurations: on a breakout board with an onboard ceramic antenna (pictured in Figure 13.4), on the breakout board with a uFL connector for an external antenna, and a pair of shields—one with the ceramic antenna and one with the uFL lug. The shields have a bonus SD card slot and a prototyping area where other components can be soldered in. In addition, Adafruit has a robust library with plenty of examples demonstrating the chip's various functions.

You'll use the CC3000 breakout from Adafruit for this chapter's project: to modify the dart gun from Chapter 7 to add the capability to send text messages when the gun fires.

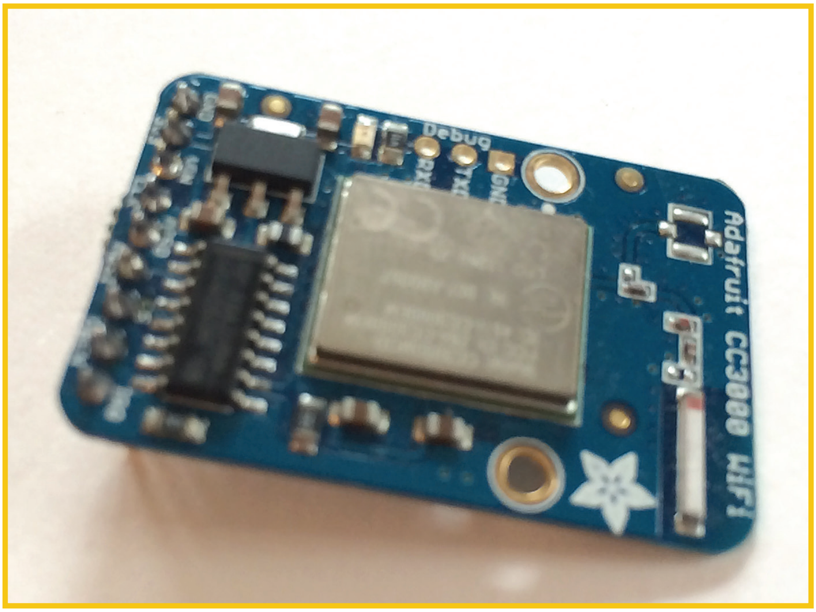

FIGURE 13.4 The CC3000 breakout is an easy way to connect your Arduino to the Internet.

Arduino Wi-Fi Shield

The Arduino Wi-Fi Shield (seen in Figure 13.5) is the inspiration for and competitor with Adafruit's CC3000 shield. Like its newer cousin, the Wi-Fi Shield can negotiate all standard Wi-Fi networks and offers encryption; and like Adafruit's shield, it has an SD card slot.

One advantage the Wi-Fi shield has is that it monopolizes only four pins, so another shield can fit on top; they include female headers for that purpose. Finally, one obvious difference is that the older Wi-Fi Shield lacks the prototyping area of the CC3000 shield.

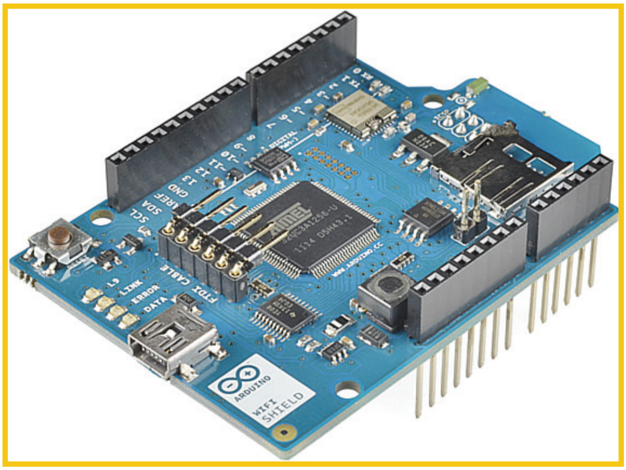

FIGURE 13.5 Arduino's Wi-Fi Shield offers many tempting features.

Roving Networks WiFly

The WiFly is a handy add-on board (Sparkfun P/N 10822) with the same footprint as the popular XBee wireless cards, making them easy to incorporate into electronics projects. Many breakout boards and shields accommodate this footprint.

Like the CC3000 boards by Adafruit, these Wi-Fi boards offer onboard whip antennas or external antennas via uFL and RP-SMA connectors. They need only four Arduino pins to get you on the Internet.

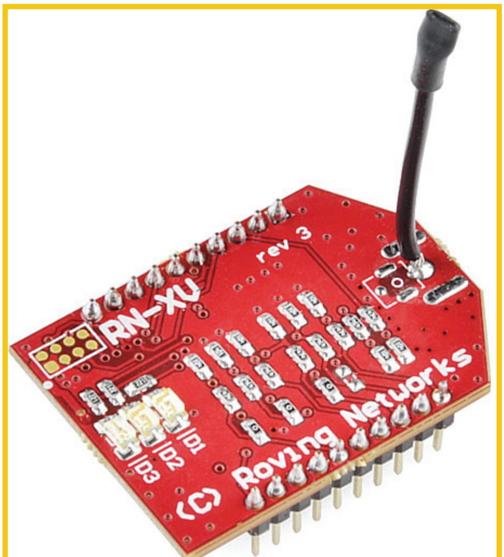

FIGURE 13.6 The WiFly has the same configuration as commonplace wireless cards.
Credit: Roving Networks

Microcomputers: Non-Arduino Solutions

Although robust, Arduinos are limited compared to a category called microcomputers, such as the Raspberry Pi pictured in Figure 13.7. These devices are characterized by featuring the Linux operating system stored on a chip. They're essentially a low-end PC in circuit board form, and frequently they can be plugged into a monitor and keyboard and work just like a computer; you can see a desktop with icons, and so on. They can be used to surf the Web, and they usually offer Wi-Fi or Ethernet connectivity built in. As such, many web-interacting projects will use a microcomputer instead of an Arduino.

However, this category doesn't fall within the realm of this book, so I'll cover some of the most popular microcomputers only briefly.

BeagleBone Black (beaglebone.org)

The BeagleBone Black high-performance Open Source microcomputer features a 1GHz processor, 512MB of RAM, an A8 processor from Texas Instruments, and the capability to connect to a TV via HDMI and the Internet via Ethernet.

Raspberry Pi (raspberrypi.org)

Another Open Source microcomputer, the Pi is the most popular entry in this category. You load the Linux operating system on an SD Card, including variants designed to be media centers, serving up movies and music. Because of its popularity, the Raspberry Pi has countless projects designed for it and an impressive knowledge base that can help you out with your own work.

pcDuino (pcduino.com)

The pcDuino runs like a PC using Linux or Android ICS, the version of Android meant for tablets and other interactive devices. As a nice bonus, it has the same female header pins as an Arduino so it can use the same shields.

Arduino Yún (arduino.cc/en/main/ArduinoBoardYun)

Arduino Yún is Arduino's response to the Raspberry Pi and other microcomputers. The Yún has an ATmega32u4 microcontroller running Arduino, but also packs an Atheros AR9331 processor that runs Linux. That and a built-in Wi-Fi chip make the Yún the latest and greatest in the Arduino world.

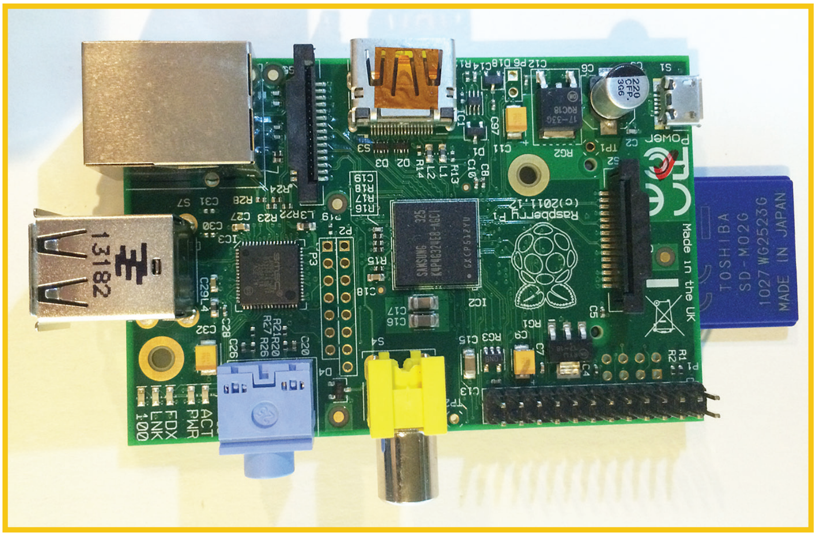

FIGURE 13.7 The Raspberry Pi is basically a tiny PC.

Dart Texter

The final project of the book updates the Dart Sentry from Chapter 7, shown in Figure 13.8. The Dart Sentry is a Nerf gun equipped with a passive infrared (PIR) sensor and a servo that yanked back on the trigger when an intruder was detected. We're going to add a Wi-Fi card so that when the dart is loosed, a text message is sent to your phone so you know what happened.

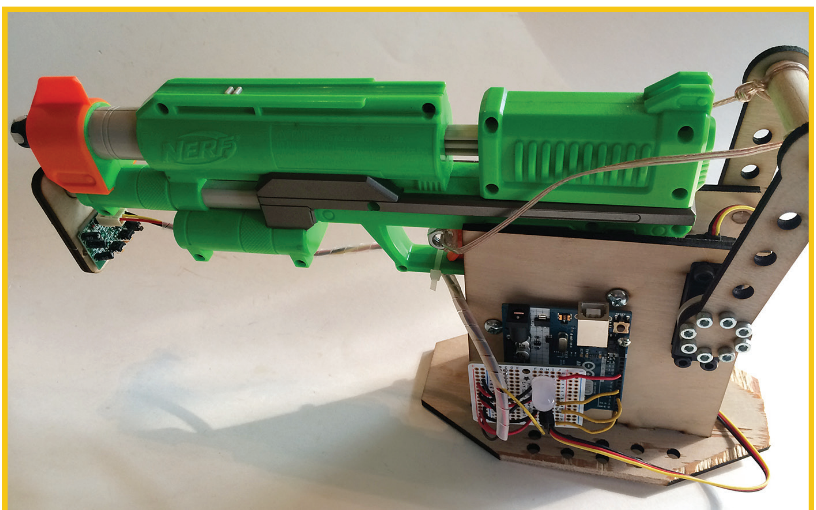

FIGURE 13.8 The Dart Texter lets you know when it shoots.

Parts List

Assuming you're beginning with a finished Dart Launcher, you'll need only a few parts for this rig:

- Adafruit CC3000 breakout board, P/N 1469
- Jumpers, such as Sparkfun P/N 11026
- Mini breadboard, Sparkfun P/N 12046
- Wire, a piece just a few inches long

Step-by-Steps

Follow along with these easy steps to get the hardware set up:

STEP 1 Attach the mini breadboard to the gun-holding rig, as shown in Figure 13.9. Most of these boards have adhesive backing, or you could use double-sided tape.

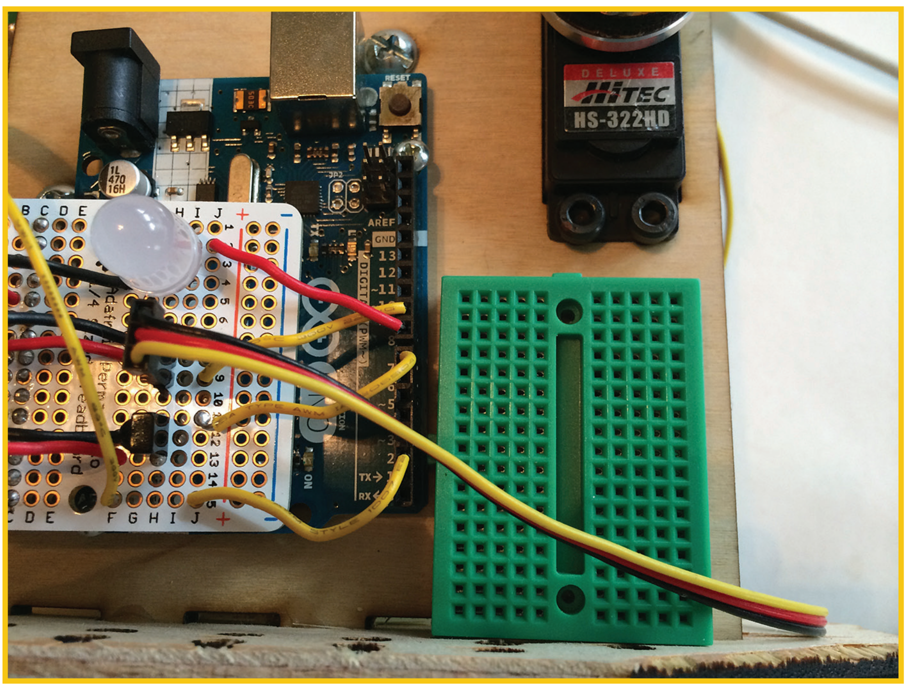

FIGURE 13.9 Attach the breadboard to the gun-holding rig.

STEP 2 Plug in the CC3000 with the supplied header pins. (If you haven't yet done so, you should solder the pins into the breakout board, allowing you to plug it into a breadboard.) Give yourself room to add jumpers; one row will do it, as you see in Figure 13.10.

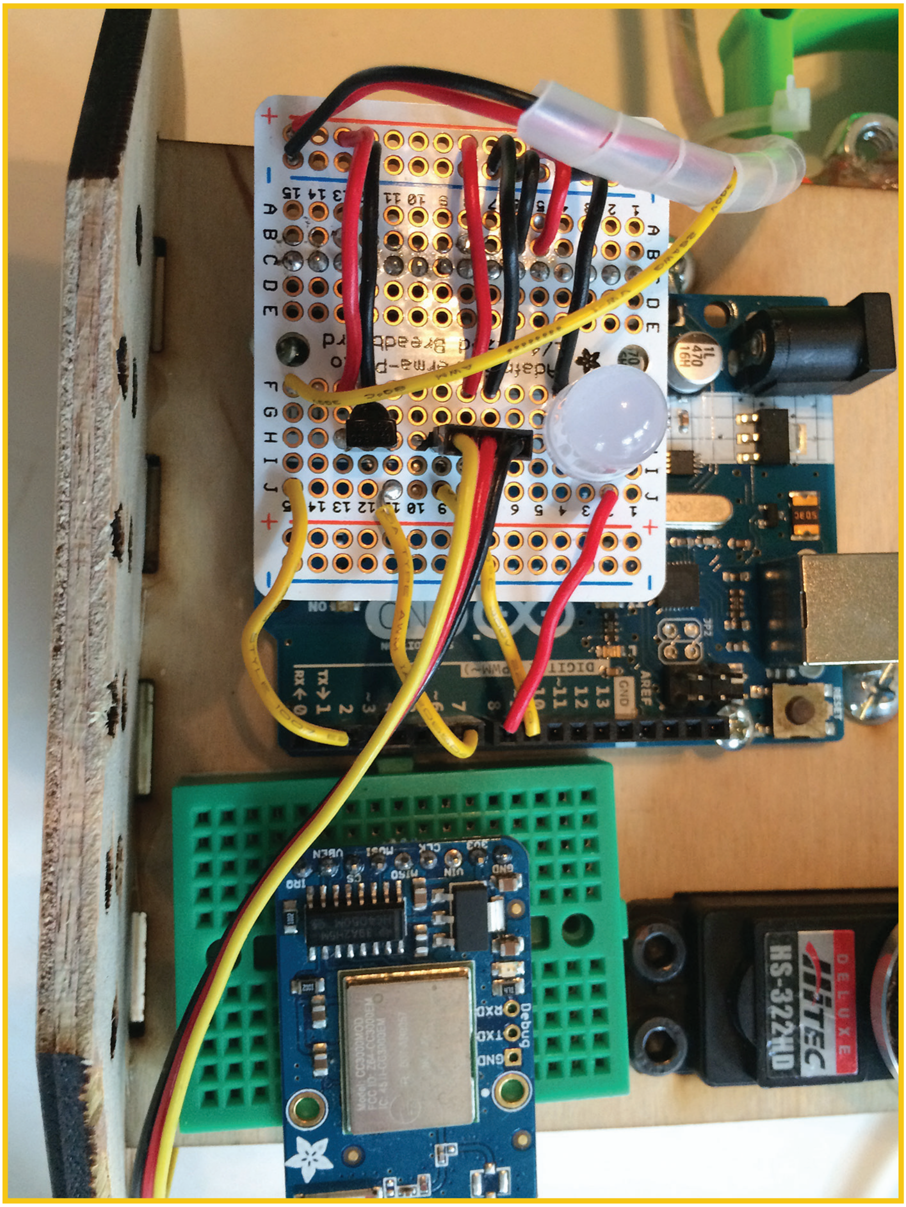

FIGURE 13.10 Plug the Wi-Fi card into the breadboard.

STEP 3 You'll need to adjust a couple of wires from where they were in the Dart Launcher. First, move the light-emitting diode (LED) wire from pin 13 to pin 8. Then, move the PIR's data wire from pin 3 to pin 2. This will make room for the CC3000's pins. You can see how it should look in Figure 13.11.

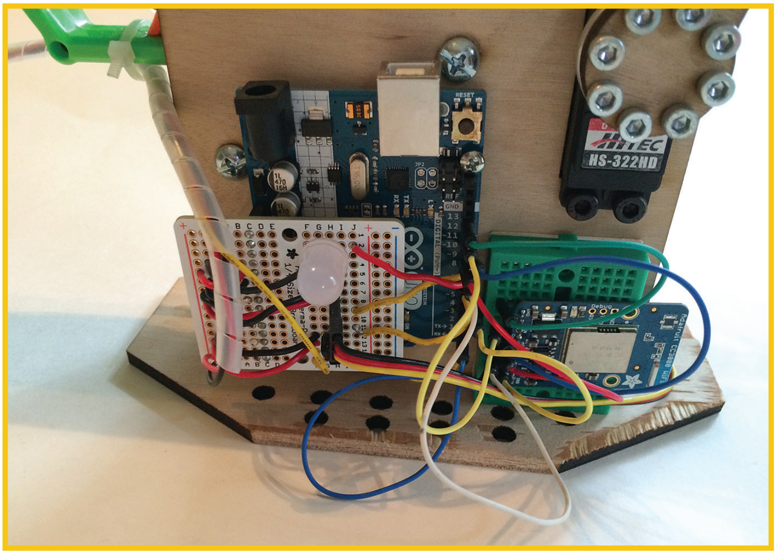

FIGURE 13.11
Move a couple of wires around to make room for the Wi-Fi card.

STEP 4 Wire up the individual pins of the CC3000: GND goes to GND on the Arduino, CLK plugs into pin 13, MISO into pin 12, MOSI into pin 11, CS into pin 10, VBEN into pin 5, and IRQ into pin 3 (see Figure 13.12).

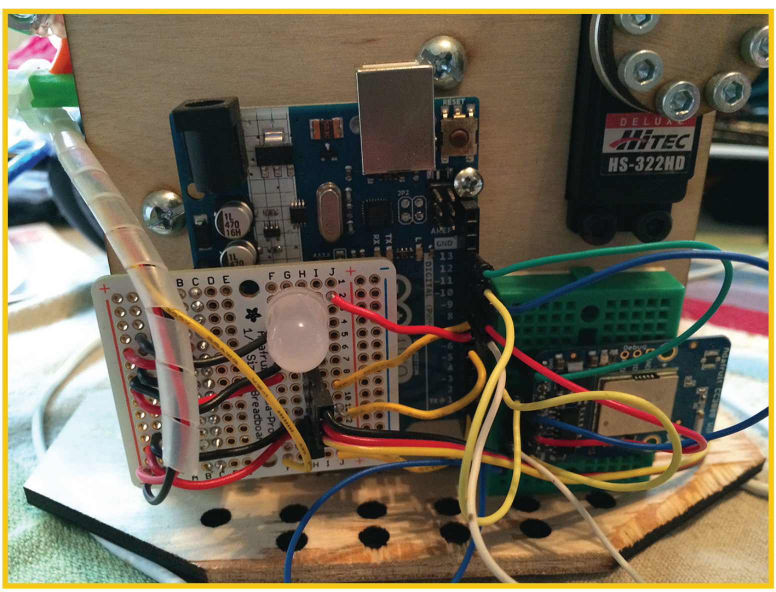

FIGURE 13.12
Connect the CC3000's pins to the Arduino.

STEP 5 Solder a wire to the power bus of the solderless breadboard, and then plug it into the pin of the CC3000 marked VIN. See the red wire in Figure 13.13 called out with an arrow.

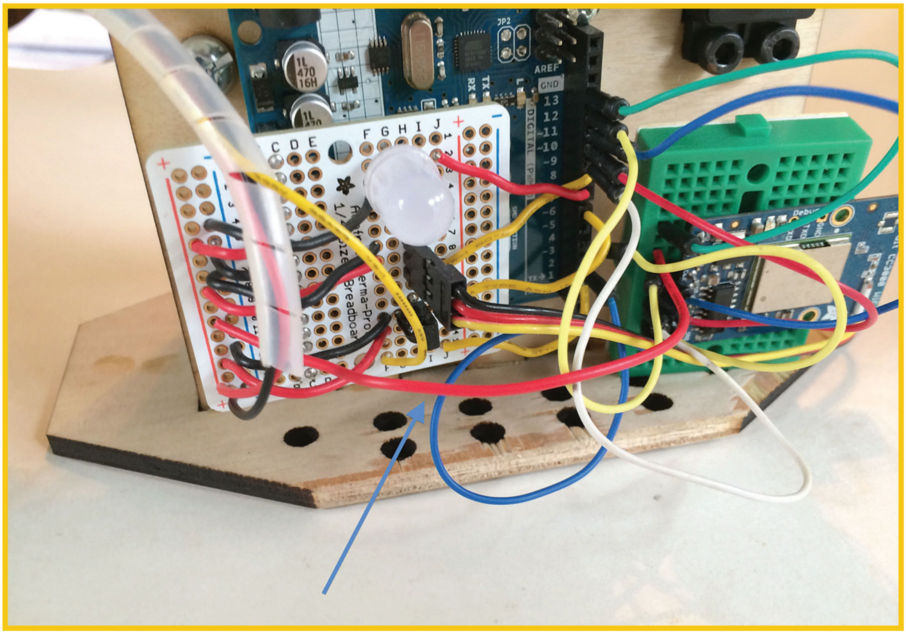

FIGURE 13.13 Grab 5V from the power bus of the breadboard.

Code

The code for the Dart Texter is exceedingly complicated—so much so that if I printed it here in its entirety it would add 10 pages to the book. Instead, I'm going to briefly cover the steps involved with programming the Texter:

1. Test out the CC3000 to ensure that it has been wired up correctly. First, download the CC3000 library from https://github.com/adafruit/Adafruit_CC3000_Library and install it. Launch one of the example sketches, buildtest, and add your Wi-Fi network name and password to the right place in the sketch (it should be obvious) and launch it. This will test out your Wi-Fi card and do things like ping websites and do a test connection to a domain name server.
2. Sign up with Amazon Web Services (AWS, aws.amazon.com), which has among its applications the Simple Notification Service (SNS) messaging service. Click the SNS icon and follow the directions to get signed up. You should end up with a couple of long strings of letters and numbers, access keys that will allow you to send a text

remotely.

3. While on AWS, you'll need to create a "topic" or specific set of procedures that results in the text being sent. Make sure you've set your region to East Coast (regardless of where you are—sending SMS messages requires that you choose that region, for some reason).
4. "Subscribe" to the topic and choose SMS for the delivery method, and then enter your phone number. The service should send you a confirmation text, which you reply to officially subscribe. Now all you have to do is trigger the snsPublish function in the sketch and your message will get sent.
5. Download the Dart Texter sketch from github.com/n1 and open it up. Change the network name and password as you did with the buildtest sketch. You'll also need to add the keys that you generated while signing up with AWS. You're done! If you get confused, AWS has really great help files and forums. If you have a question, you're sure to find the answer there.

Summary

In this chapter, you learned about some of the possibilities of adding web connectivity to your next Arduino project. Also, you're done with the book! Thanks for reading.