COOL IT: With some inexpensive electronics, you can switch things on and off in your house from afar—including the air-conditioning.
PHOTO: DAVID SCHNEIDER

A SMARTER, CHEAPER THERMOSTAT

Controlling home thermostats need not cost a bundle

Call me a wimp. Since moving from New York to North Carolina, I've grown completely dependent on one of engineering's greatest achievements—the air conditioner. For three seasons, the climate of the Southeast is lovely, but summers here are stultifying unless you pump the heat back outdoors.

Unfortunately, cooling the house takes a long time, so come summer, the AC is left on pretty much all the time. Better would be to switch the AC on, remotely, an hour or two before heading home. There's an easy way to do that nowadays, thanks to another great engineering achievement: the Internet. You might use Ecobee's Web-enabled thermostat, but that costs more than US $300. And what if you want to control the lights as well? Such systems exist, but they're even more expensive.

If you don't need a lot of bells and whistles, you can control your central air conditioner and lights without great expense. The system I cobbled together uses a single-board computer of growing popularity: the Arduino. Devoted *IEEE Spectrum* DIYers have seen this name several times in the past year, most notably in "Barbot, the Automated Bartender" (Geek Life, December 2009).

Arduinos come in many flavors. I bought an Arduino Duemilanove ($30). The nifty thing about this board is the way various accessory boards, or shields, can plug right in. Connecting my Arduino to the Internet was as easy as attaching an Ethernet shield ($46) and connecting it to my home router with an RJ-45 cable.

But how to control the air-conditioning? My thermostat, an ancient design, is basically a mercury switch that turns the AC or heat on and off when the bimetallic spiral it's attached to winds or unwinds with the change in temperature. So all I needed to do was to wire a second, Arduino-controlled switch in series with the mercury switch. Then the Arduino could either allow the thermostat to function normally (with the second switch closed) or prevent the air conditioner from being turned on (second switch open).

My home thermostat is located nowhere near my router, so to avoid having to string a cable between them, I decided to control the thermostat by radio, using X10 home-automation hardware. You may know X10 as a system that communicates using signals sent over power lines. But it turns out that some X10 devices can use the airwaves, and the price is right: Just $5 buys an X10 Firecracker radio transmitter. Another $9 gets you an X10 TM751 unit, which the Firecracker can command by radio to switch a plug-in appliance on or off.

By wiring the thermostat's mercury switch in series with a small 5-volt relay, I could enable or disable the air conditioner from another room, using the TM751 to power a wall wart attached to the relay's coil. Interfacing the Arduino with the X10 Firecracker proved easy, because others had posted online code they'd used for this very purpose. Now my Arduino had the means to shut off my air-conditioning system until I commanded it through the Internet to let normal temperature regulation resume.

So far, so good. But what if I didn't want the temperature in the house to climb quite so high? My little Arduino assistant could prevent that, but it needed some way to know how hot the house was. For that, I bought a Dallas Semiconductor 18B20 digital temperature sensor ($4), along with the relay I needed for the second thermostat switch ($2), a wall-wart AC adapter to power the Arduino ($6), and a 5-V wall wart to operate the relay ($6). That brought the total bill of goods to $108—pretty cheap as Internet-connected home-automation systems go.

Assembling the hardware was easy: The Ethernet shield just plugs into the Arduino, three wires connect the Firecracker, and two

more wires and a 4700-ohm resistor attach the temperature sensor. The only real work involved was writing some code so that I could command the Arduino through the Internet. Fortunately, Arduino's developers have done most of the heavy lifting and written an Ethernet-shield library.

In principle, I could have made my little Arduino board into a Web server, but I worried about using that approach. If the house lost power while I was on vacation, for example, my router might restart with a new IP address from my Internet service provider, rendering me unable to find my Arduino on the Internet. So I instead devised a system based on e-mail.

To control my thermostat, I send an e-mail message to an account I've set up just for this purpose. My ISP gives me several free accounts, so that's easy enough. Other mail servers would have been more difficult to use because they typically require Secure Shell, which the Arduino Ethernet library does not yet support. The Arduino board checks my account every 10 minutes. If it finds a message, it looks for lines that begin with a special sequence of characters, which serve as my secret password, followed by a one- or two-character instruction.

One instruction commands the Arduino to put the temperature regulation into "economy mode" (which sets the target temperature to some uncomfortably high value that won't make the AC run very much). Another commands it to return temperature regulation to its normal mode, controlled by the thermostat's faceplate setting.

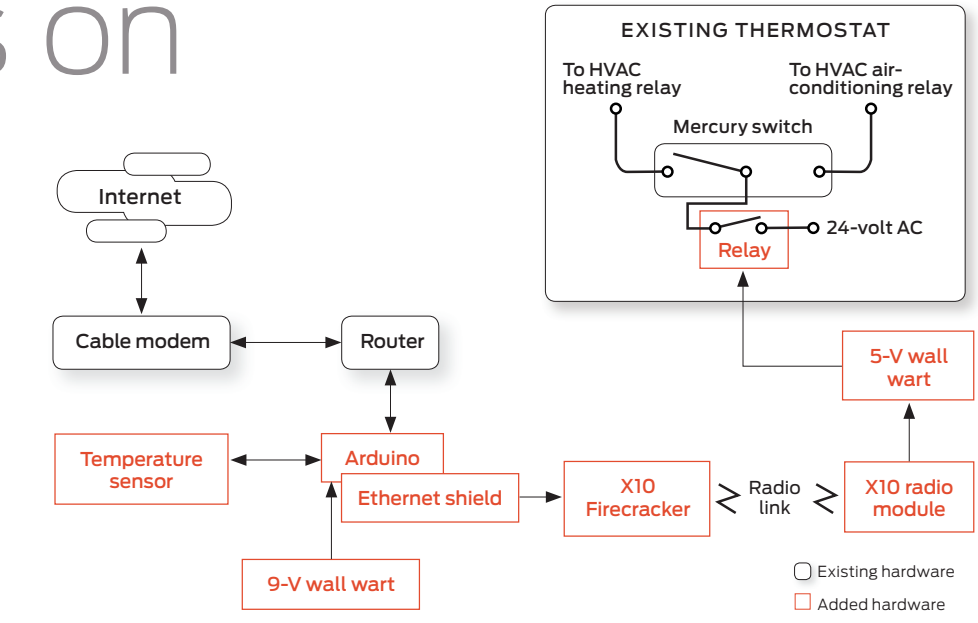
Other instructions allow the Arduino to turn on or off lamps that are plugged into additional TM751 units (at $9 each, I'm tempted to purchase a bunch). And lastly, I included an instruction for the Arduino to send me e-mail notification of the status of the house—the current temperature, whether it's in normal or economy mode, and whether any remotely operated lamps are switched to on. Now I can even control these things in my house using an e-mail–capable phone to send and receive short messages.

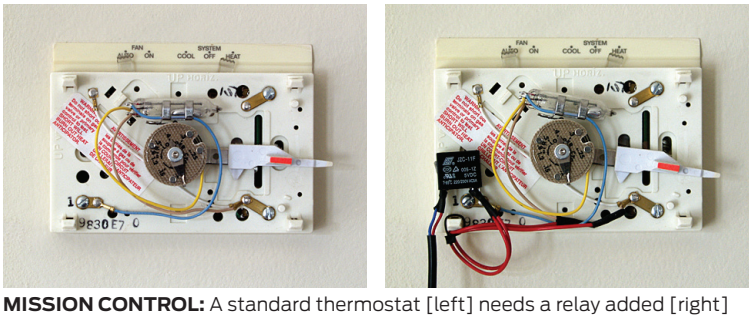
Although I've not taken pains to tune the feedback loop that controls the house when the Arduino is in economy mode, the air temperature seems just as stable as when the regular thermostat is doing the regulation. And the temperature remains stable even if I disconnect the Ethernet cable. So the little system I've assembled appears reasonably robust, although when my 11-year-old daughter saw the Internet-to-Arduino-to-Firecracker-to-TM751-to-relay control channel, she asked, "Who was that guy who drew pictures of those crazy contraptions?" That would be Rube Goldberg, dear.

If my contraption continues to work well all summer, I'll be tempted to rig it to regulate the heat in winter. But I won't do that until I've gained more confidence that no software or hardware gremlins are lurking. After all, I don't want to end up with a burst pipe if some rare failure condition locks up the Arduino board and leaves the heating completely disabled. Hmmm, maybe I could build another circuit to attach to the...

I can already hear Rube Goldberg chortling with glee.

—David Schneider





MISSION CONTROL: A standard thermostat [left] needs a relay added [right] to be controlled through the Internet [see diagram]. *PHOTOS: DAVID SCHNEIDER (2)*