11

THE HVAC GREEN HOUSE MODEL

Vern Graner

Introduction

In this chapter, we will explore a Propeller-powered HVAC (heating, ventilation, air conditioning) energy-saving “green” house model. This experimental platform takes advantage of the Propeller chip’s low cost and high versatility, allowing us to experiment with intelligently managing a scale model of a typical residential central heating and cooling system.

Through a combination of low-cost components and some powerful software objects, we can easily explore different methods to boost HVAC system efficiency, reduce operating costs, and ultimately make the environment more comfortable for the building’s inhabitants. During our exploration, we will be covering the following topics:

- Examining the makeup and drawbacks of central air HVAC systems
- Creating the HVAC green house model
- Designing the control electronics: Scalability and real-world considerations
- Software creation considerations: What gets processed where?
- Modular programming using Propeller objects
- Electronics design and implementation

We’ll start out by getting familiar with some inherent problems in a typical residential HVAC system and see how using the Propeller as a system management tool can be educational, interesting and fun!

Resources: Demo code and other resources for this chapter are available for free download from ftp.propeller-chip.com/PCMProp/Chapter_11.

Exploring the Problem

On April 23, 1886, an inventor named Albert Butz patented a furnace regulator he called the "damper flapper." When a room cooled below a predetermined temperature, a switch energized a motor to open a furnace's air damper so the fire would burn hotter. When the temperature rose above the preset level, the switch automatically signaled the motor to close the flapper, damping the fire so it burned cooler. This simple yet ingenious thermostat was the seed that eventually grew into a little company named "Honeywell."

Information: To learn more about Albert Butz and Honeywell, read the article referenced here at www51.honeywell.com/honeywell/about-us/our-history.html.

Though the "damper flapper"-inspired thermostat design is now more than 100 years old, the typical residential central air system controllers of today rely on essentially the same principles. Though the modern thermostat may contain a microprocessor, a real-time clock, timers, and factory-calibrated solid-state temperature sensors, the primary operating mode remains the same; when the air temperature nears a preset point, the thermostat simply turns an HVAC system off or on. Though this method of temperature management is widely used, it has observable issues regarding comfort and efficiency.

One problem with the "binary" nature of a thermostat is that the optimal temperature point is "chased," following a predictable sinusoidal path as it oscillates between "too cold" and "too hot." The reason for this cannot entirely be laid at the feet of the thermostat module itself, but instead may be attributed in part to the design of the heating and cooling portions of a typical central-air system. These systems are usually tuned to produce a *specific* amount of heated or cooled air (usually in cfm, or cubic feet per minute) when activated.

This precludes providing a *proportional* response to temperature needs, as fan speeds and heating/cooling element temperatures are typically set at a fixed value (though some newer HVAC fans do offer high-/low-speed options in a rudimentary attempt to provide some proportionality). As long as the thermostat is calling for heat or cool, most HVAC systems will blindly produce the hottest or coldest air that they can until the thermostat tells them to stop. To make matters worse, it is rare that a residential HVAC system has more than one thermostat, so a single sample is used to provide the standard for the entire building.

Without a way to monitor the temperature in each room, managing the distribution of conditioned air so that all rooms attain the same temperature requires someone to manually visit one room at a time, measure the room temperature, and then adjust the air register in that room in an attempt to "balance" it with the others.

TRYING TO STRIKE THAT BALANCE

This "balancing" process would seem to be relatively simple to perform and not require much maintenance after it has been completed. However, there is more to this process than you might imagine. When balancing airflow in a building, you enter the first room and either increase the opening of the air register to allow more airflow or decrease it to create the opposite effect. As pointed out earlier, in most central air systems, you have a set amount (in cubic feet per minute) of conditioned air being created. So when you reduce the amount of air allowed into one room, air is then diverted to the *other* rooms in the building.

This causes unwanted (and usually unpredictable) changes in temperature elsewhere in the building. The inverse is true as well: When you open a register more fully, you may "steal" air from rooms that were otherwise already comfortable. Determining the amount that an adjustment in one room may affect other rooms is further complicated by factors including air duct size, the distance from the air production source, and the size of the return air path for the rooms.

If this room-to-room "balancing" process is completed by a competent HVAC professional, this complex inter-relationship between air register settings can result in reasonably consistent air temperature between all rooms. However, the adjustments are based on the readings taken at a specific point in time and assume that the entire system of rooms is *static*. Unfortunately, in real-world HVAC environments, this is just not the case.

The reality is that a room's heating and cooling needs will change over time as many variables come into play. For example, sunlight shining on an exterior wall, through a window, or on the roof will likely increase the room temperature. The contents of a room can also have a considerable impact on the room temperature. Consumer electronic equipment such as stereo systems, televisions, and computers all create heat when operating. Other typical household devices, such as clothes dryers, refrigerators, stoves, and coffee makers, all release heat as well. Even if a room is relatively devoid of heat-producing equipment, it is extremely likely to have some type of lighting system, with low-voltage halogen lights and incandescent light bulbs that generate lots of heat as a by-product of creating light.

Of course, we can't overlook the inhabitants themselves. Humans give off a surprising amount of heat. It's generally accepted that a typical adult human being gives off about the same amount of heat as a 100-watt incandescent light bulb! Then, just when you thought it couldn't get more complex, think of what happens to room temperatures when you open doors between rooms or, worse yet, open exterior doors and windows!

With so many different variables affecting room temperature at any given time, the idea of attaining consistent, balanced temperatures among all the rooms in a building by the application of a one-time "balancing" of individual registers seems rather unlikely. Even if you could manage to get all the rooms to reach a specific temperature, the ultimate arbiter of temperature is not the thermostat or even a thermometer. It's the *inhabitants* of a given room that decide if the temperature is "right" or would need to be altered to make them more comfortable.

Personal perceptions of "warm" and "cold" vary widely, and it is rare that all of a building's inhabitants will be comfortable at the same room temperature. For example, a sedentary person may be comfortable in a room at 76 degrees, but an active person may feel the room is too warm. Even a sedentary person, when placed in a room with lots of electronic equipment, may require more cooling in order to be comfortable.

The sad reality is that, in most cases, a single thermostat in a multiple-room building just isn't up to the task of creating a balanced, comfortable temperature for everyone. If the thermostat in the building decides to make it cooler or warmer based on its single sample and that causes some rooms to be too hot or cold, that's just too bad. In some situations, it's the "worst case" room that controls the temperature for the whole building. A good example would be cranking up the AC to deal with one hot room in the house and wasting lots of energy "freezing out" other rooms.

So given such an extremely variable and dynamic environment to deal with, is it possible to design an upgrade to a standard HVAC system by adding some technology "smarts" to deal with all these situations? To find out, we built a scale-model HVAC green house on which we could experiment.

THE HVAC GREEN HOUSE MODEL

Two main considerations were taken into account when building the model house for this project. First, of course, we wanted to have a test bed to experiment with mechanical, electrical, and software designs to deal with the HVAC problems outlined previously. Second, we hoped to end up with an attractive and educational display piece we could use to showcase the concept of a dynamically managed HVAC system. We wanted to visually demonstrate that a small amount of technology could make a large difference in energy efficiency as well as the comfort level of the building inhabitants.

CHOOSING THE STRUCTURAL/MECHANICAL ELEMENTS

A number of smaller decisions needed to be made before the construction of the model house began. For example, we wanted all the electronic and mechanical devices to be visible, so using quarter-inch-thick clear plastic panels for the walls, ducts, and structural components was a must. In Fig. 11-1, you can see the preliminary sketch with clear panels.

The trade-off was that this type of material doesn't do a very god job of insulating the rooms from ambient temperatures external to the model or from adjacent rooms. To compensate for this, we did away with windows and doors in the rooms to make each one a "sealed" environment. Each room would only receive air from a single air register and would exhaust air through a single return air vent. We also "oversized" some components, in effect using "brute force" to overcome heat loss/gain from the construction materials. For example, at our scale size of approximately "1 inch = 1 foot," the air registers in each room are nearly one and a half feet wide by one foot tall, and the air circulation fan would be a whopping five feet tall!

The HVAC Itself The next consideration for this test system was exactly how we were going to cool or heat the air. We searched in vain for an actual Freon-based HVAC system with an associated external evaporator/condenser small enough to fit in our "dollhouse" scale building, but were unable to discover any inexpensive units this small. Luckily, we were able to find a small-scale solid-state heat pump that used a Peltier

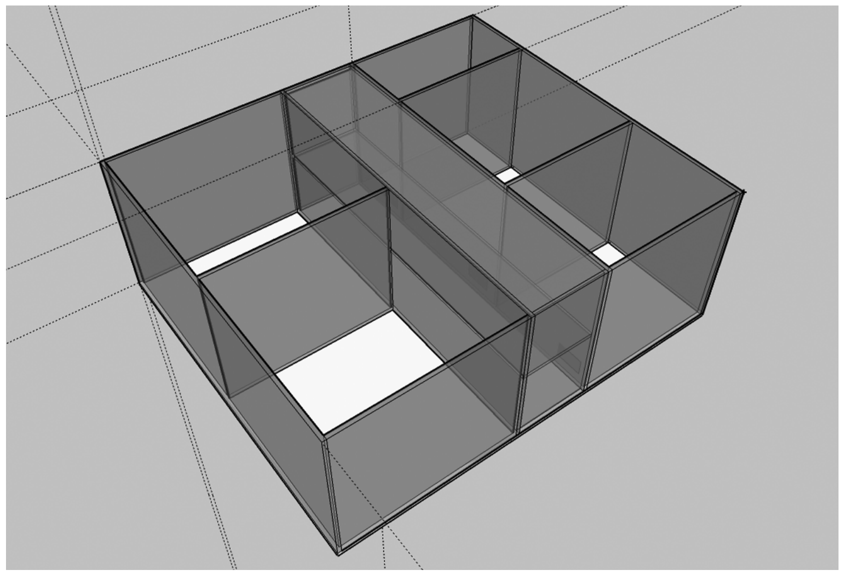

Figure 11-1 **Preliminary Google sketch of HVAC green house.**

device. Typically used in portable electric "iceless" coolers for RVs and cars, Peltier devices can act as both a cooling and heating device.

The Peltier device we used (see Fig. 11-2) consisted of a 40-mm-square thermoelectric heat pump. The device conveniently operates on 12 volts DC and draws 5 amps of current. On our test power supply running at approximately 12.3 volts, we measured the hot side at 183.6°F and the cold side at 56.2°F. In order to reverse the hot and cold sides, all you need to do is reverse polarity of the applied 12 V power.

The Cooling Tower We mounted the Peltier device in a clear polycarbonate "cooling tower" with 12 V high-speed fans on both the external "waste" heat sink and the internal "supply" heat sink that would serve to heat or cool the inside of our model house. This cooling tower (Fig. 11-3) was made as a separate enclosure, allowing us to remove it from the model for maintenance or cleaning and also for access to the send and return air ducts and the center wiring channel (aka "wire chase").

The Air Registers As we now had the capability to cool and warm air, we needed a way to remotely adjust the *amount* of air delivered to each room. This was a key concept in allowing us to experiment with balancing how much air was delivered to a given room. Though commercially available air registers used in residential construction consist of a series of adjustable louvers, we decided that a simple sliding panel would allow us to adjust the size of the entry port to the room while also making it clear to an observer exactly how much of the air vent was occluded. We fabricated brackets and used servo motors with linkages to adjust the sliding port cover as shown in Fig. 11-4.

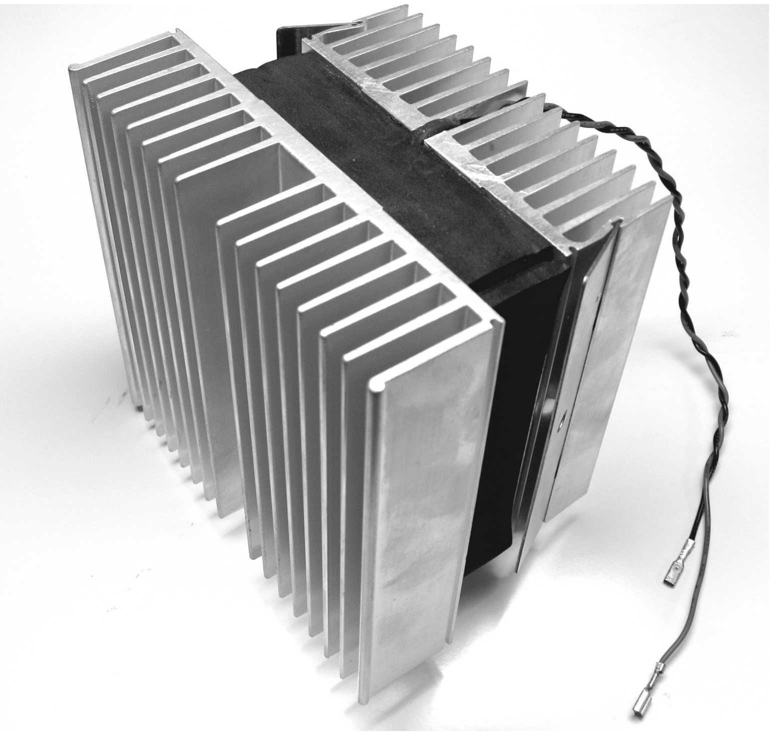

Figure 11-2 12 volt "Peltier" solid-state thermoelectric heat pump.

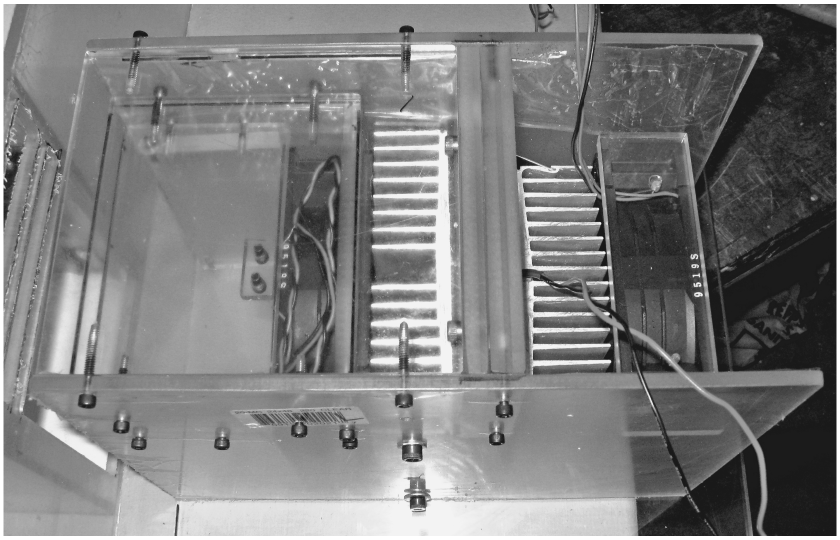

Figure 11-3 Clear polycarbonate "cooling tower" with external and internal blower fans mounted.

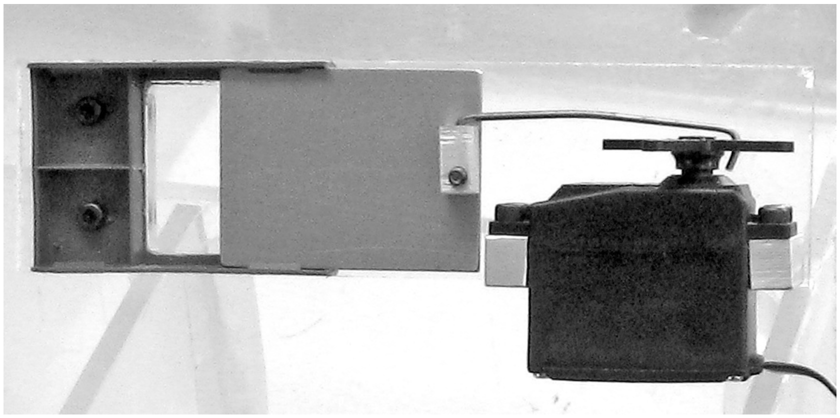

Figure 11-4 Servo-controlled air register prototype.

The Air Bypass System One of the last things to be implemented was an air bypass system (Fig. 11-5) that would allow us to either recirculate the existing air in the house for normal operation or draw air from outside the house, pass it through all rooms, and then exhaust it out the opposite end of the house for "fresh air" mode of operation.

The idea behind this feature was twofold: First, in the event that the external ambient air temperature was sufficient to cool or heat the inside of the house, engaging this

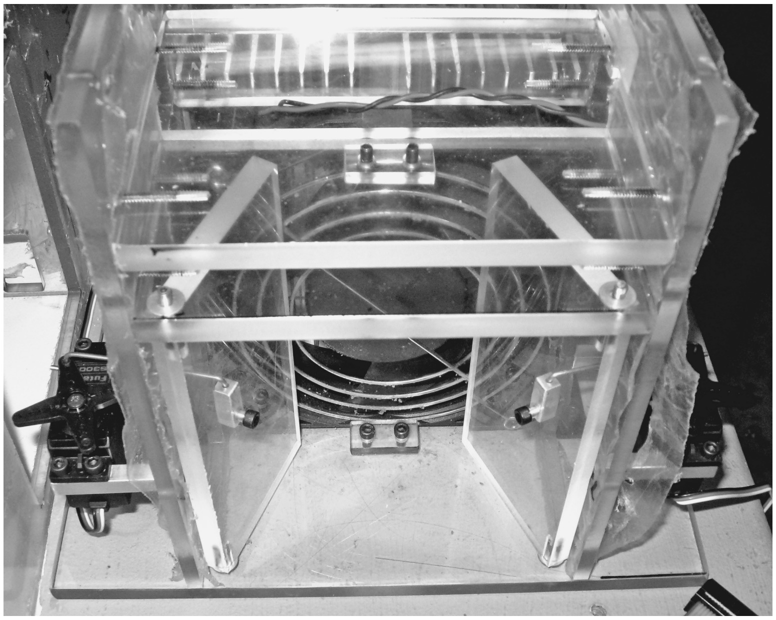

Figure 11-5 Photo of cooling tower servo-controlled "bypass" doors.

bypass and activating only the blower fan would allow the house temperature to be brought to a comfortable level without expending energy operating the heat pump. This part was a key piece of making our HVAC system more "green."

Second, in the event that the air inside the house was found to be unhealthy or dangerous (i.e., a natural gas leak or a buildup of carbon monoxide), the system could react by sounding an alarm, moving the bypass doors into "fresh air" position, moving every room air register to 100 percent open, and activating the blower fan, thereby purging unhealthy air from the entire house.

To implement this bypass system, two servo-controlled doors were added to the cooling tower that, when open, allow air to recirculate normally through the system (see Fig. 11-6).

When closed, they block the return-air plenum path to the fan while opening side vents in the cooling tower. These side vents allow external air to be drawn in to the fan from outside the cooling tower (a source outside the house), pushed through all the rooms in the house, and then exhausted via a vent flap at the opposite end of the house as shown in Fig. 11-7.

If you look closely at the CAD drawings of the house, you may notice that when the bypass doors are placed in the "fresh air" mode (i.e., the return air duct pathway is blocked), the air delivered to each of the rooms has no place to exit. To deal with this,

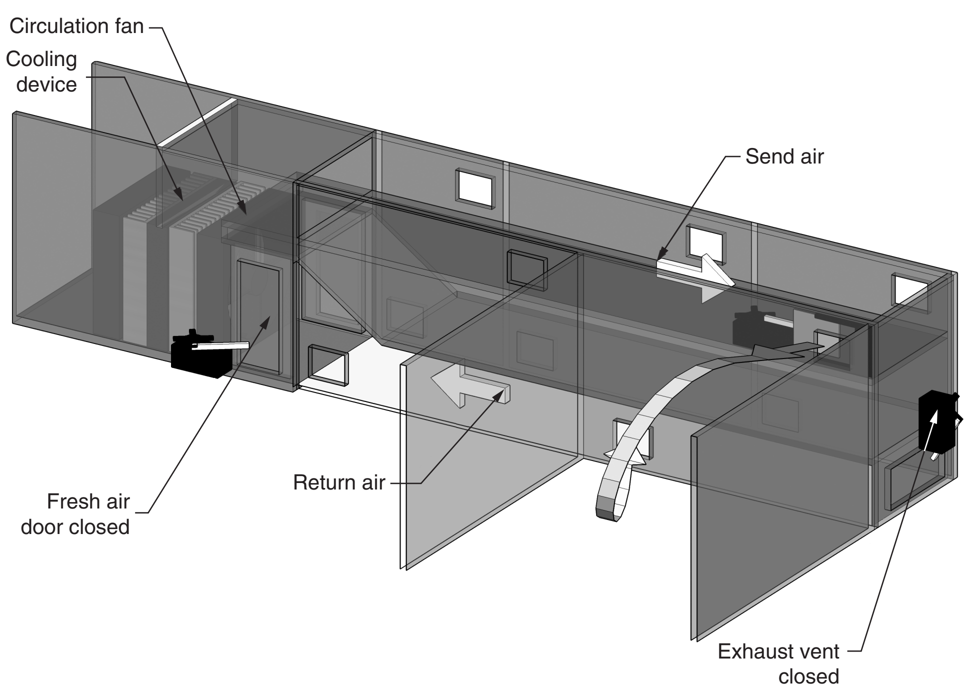

Figure 11-6 CAD "cutaway" drawing of cooling tower bypass doors in "recirculate" mode.

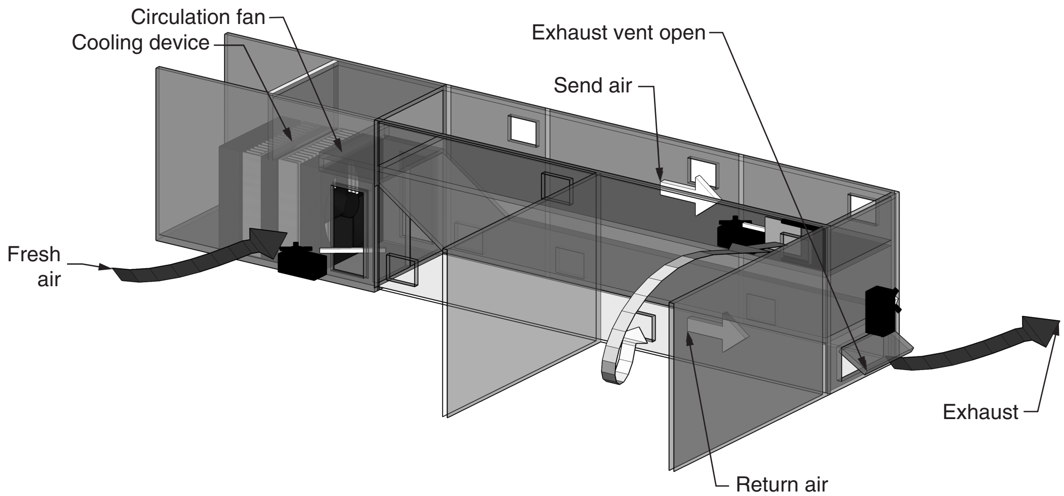

Figure 11-7 **CAD "cutaway" drawing of cooling tower bypass doors in "fresh air" mode.**

a servo-controlled exhaust vent "flap" (Fig. 11-8) was added to the end of the return air duct. When this flap is opened, the air exiting each room is vented out the opposite side of the house from the fresh air intakes on the cooling tower.

Besides being used for emergency house ventilation or for cooling/heating the house with external air, the cooling tower bypass could also be activated manually, creating a healthier atmosphere by bringing more fresh air into the house or even as a way to reduce odors from cooking or smoking.

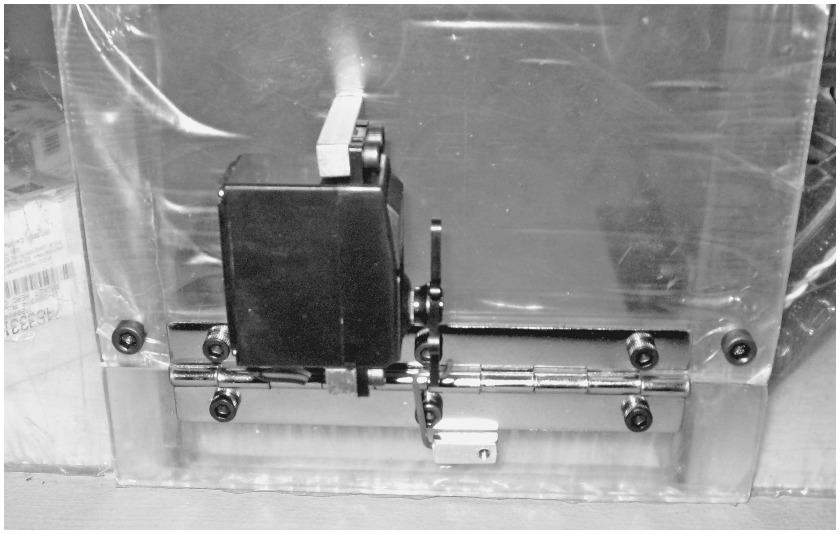

Figure 11-8 **Servo-controlled exhaust vent flap at the end of the return air duct (in closed position).**

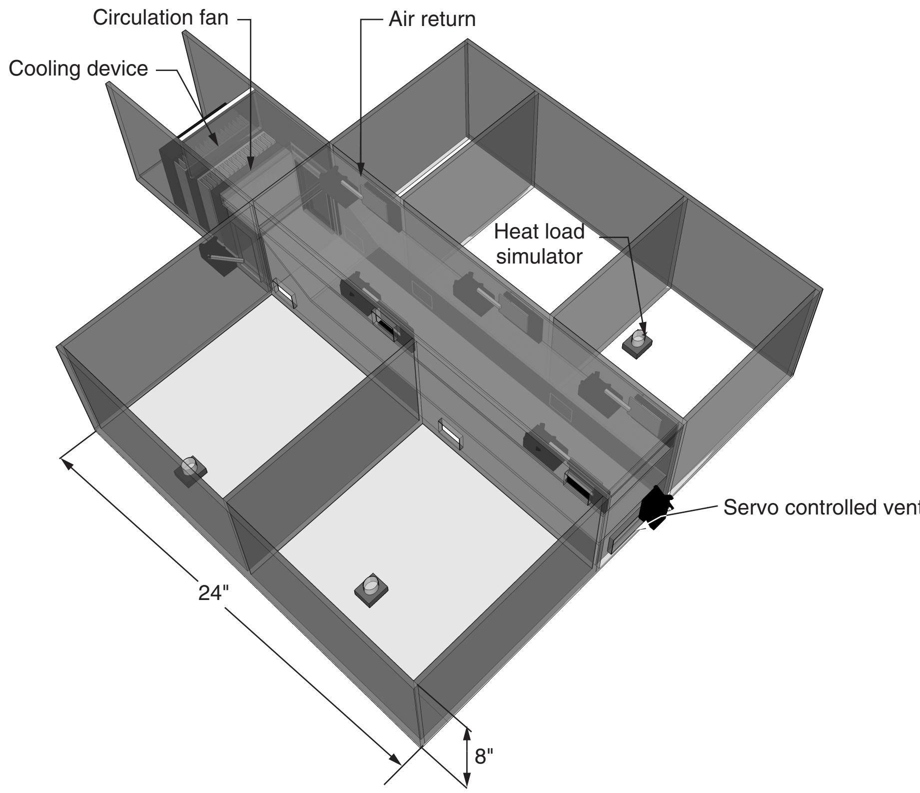

Figure 11-9 CAD drawing of final design configuration.

The Final Layout In the final design, we settled on five rooms, each one a different size to simulate different-sized rooms in a typical house as shown in Fig. 11-9.

A servo-controlled air register was created for each room and mounted in a central plenum that spanned the house from the cooling tower in front all the way to the back of the building. Beneath the "send" air plenum, we created a wire chase to allow us to run wire from each room back to a master control area.

The Hinged Roof Panels In addition to being able to control the amount of air delivered to each room, we needed a way to set a target temperature per room. We created hinged roof panels for each room and mounted pushbuttons that would allow an observer to alter the target temperature up or down simply by pressing one of the buttons. Above the buttons we added 2 × 16 LCD panels that would display both the current temperature and the target temperature to the observer as shown in Fig. 11-10.

Once all these mechanical components were mounted, it was time to design the electrical systems that would bring this creation to life.

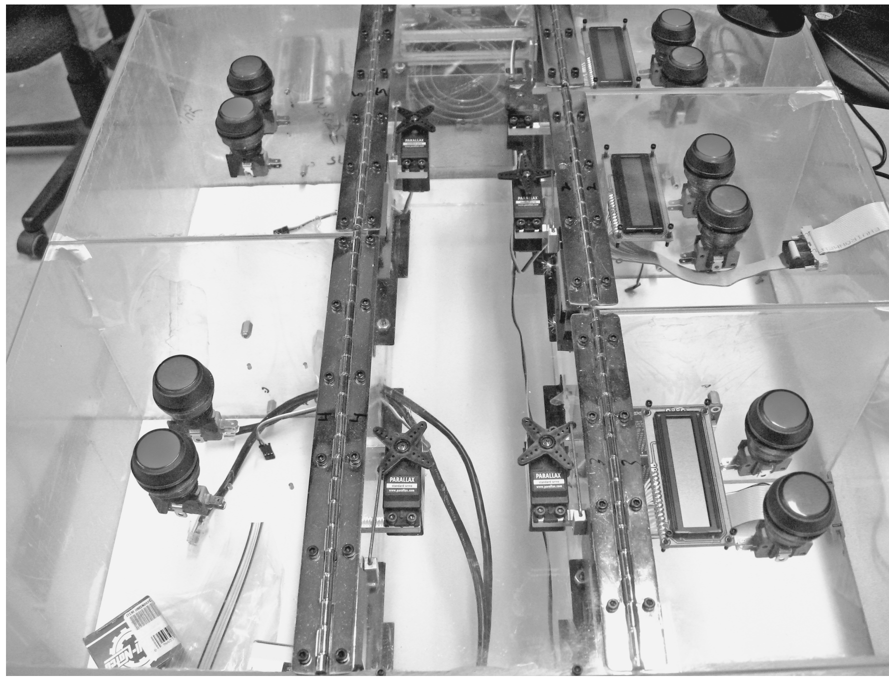

Figure 11-10 **Hinged top panels with RED/BLUE temperature control buttons and LCD displays.**

DESIGNING THE CONTROL ELECTRONICS: SCALABILITY AND REAL-WORLD CONSIDERATIONS

Though we are working with a model system, we don't want to preclude the possibility that this design may be implemented on a large scale, possibly even being used in an actual residential HVAC system. Since the Propeller chip has the ability to control multiple servos and to read multiple sensors, we sketched one of the first designs using a single Propeller to operate the entire system. This would require a "star" wiring design where each room would have a separate line running all the way back to a central wiring closet.

When planning an installation of technology such as this in a new home, or retrofitting an existing home for something similar, wiring requirements are a substantial issue, and the downside to a "star" design is the cost. There is both the cost of the wire itself and of the associated labor to run all the lines. Though the "star" wiring scheme was inherently expensive, it wasn't a deal killer in and of itself.

However, the second strike against this design had to do with the length of the cable runs from each room back to the wiring closet. Typically, the distance between a Propeller chip and devices such as the 5V transistor-transistor logic (TTL)-level serial LCD unit or pulse width modulation (PWM)-controlled servo motors would be on the order of a few inches to maybe a couple of feet. At real-world distances of 50 feet or more, there was a very good chance these devices would become unstable or completely fail to operate due to voltage loss over distance and/or wire capacitance “rounding” the edges of the square wave signals.

The final nail in the coffin for this design was that a single Propeller chip simply did not have sufficient pins to support many rooms, even when using low pin-count “intelligent” devices. For example, if each room had a 2 × 16 serial LCD display, a servo motor, two pushbutton switches, a temperature sensor, and two indicator lights, the minimum pin count per room would be as follows:

1. 5 V+ power
2. GND
3. Serial LCD signal
4. Servo motor control
5. Button 1-to-gnd
6. Button 2-to-gnd
7. “1-wire” temperature sensor
8. LED1
9. LED2

As the Propeller chip typically has ~~27~~ available pins, this design would only allow *three* rooms to be monitored before all the pins on the chip had been allocated! The one-Propeller design also called for more expensive components, such as serially controlled LCD displays and 1-wire temperature sensors, as well as additional expenses in cabling because readily available four-pair CAT-5 cable doesn’t contain enough wires. So, for implementation, either *two* CAT-5 cable runs would need to be run from each room back to the wiring closet or special-order wire containing at least nine leads would need to be purchased.

If One Propeller Is Good... As the single-Propeller design was clearly not feasible, we started to look for other design ideas. The next logical step was to see if we could place the sensors and controls closer to the processor. We started looking at placing one Propeller in *each room,* as this idea solved multiple problems and provided additional benefits. First, the wiring issue would become much less complex, as we need only supply power, ground, and TX/RX lines for digital communication. Second, we would be able to reduce costs by moving away from the more expensive “intelligent” peripherals and substituting a standard “parallel” LCD display for the serially controlled unit and replacing the 1-wire temperature sensor with an inexpensive DS-1620 chip.

Having the Propeller in the room placed it in close proximity to the RC servo motor, eliminating worries of signal degradation affecting performance. Also, driving the LCD from the Propeller eliminated concerns of long wire runs corrupting serial TTL signals to the LCD. As we mentioned earlier, because the Propeller has ~~27~~ available pins, we have the option of adding items such as the inexpensive HS-1101 humidity sensor, more indicator LEDs, and even a speaker for acoustic feedback and/or alarms.

The “one Propeller per room” design also lent itself to using a network protocol and having multiple units share a single cable. This eliminated the need for “star”-type wiring, allowing all the units to be “daisy chained” on a single cable from room to room. This reduces wiring cost in new construction and greatly simplifies installation in existing buildings. So, using our “one Propeller per room design,” our new cable pair consumption would simply be:

Pair 1: 5V+/GND

Pair 2: TX/RX

This new reduced wiring consumption also means less expensive two-pair wire (i.e., telephone cable) could be used to connect the rooms to the wiring closet. In our prototype, we decided to use four-pair CAT-5 cable to support the possible current draw that we may see from the servo motors. In addition, we decided to put +12 volts on one pair to drive 12 V indicator lights in each pushbutton switch.

Developing Control Board Prototypes At this point, we decided to build a test “room control” board (Fig. 11-11) to see how well it would function. The first one was

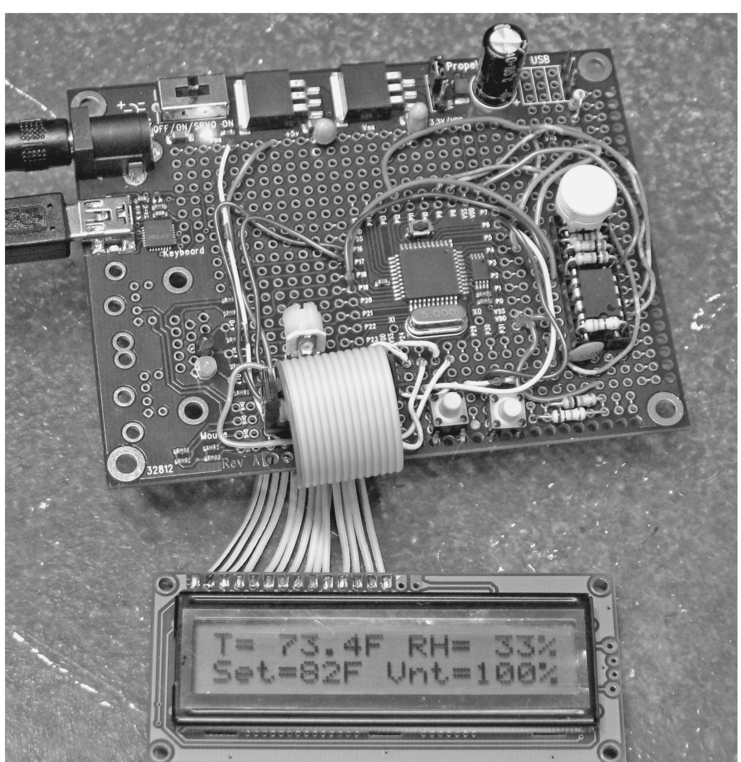

Figure 11-11 **The first prototype “room control” board with LCD display, temperature sensor, humidity sensor, servo driver, and pushbutton switches.**

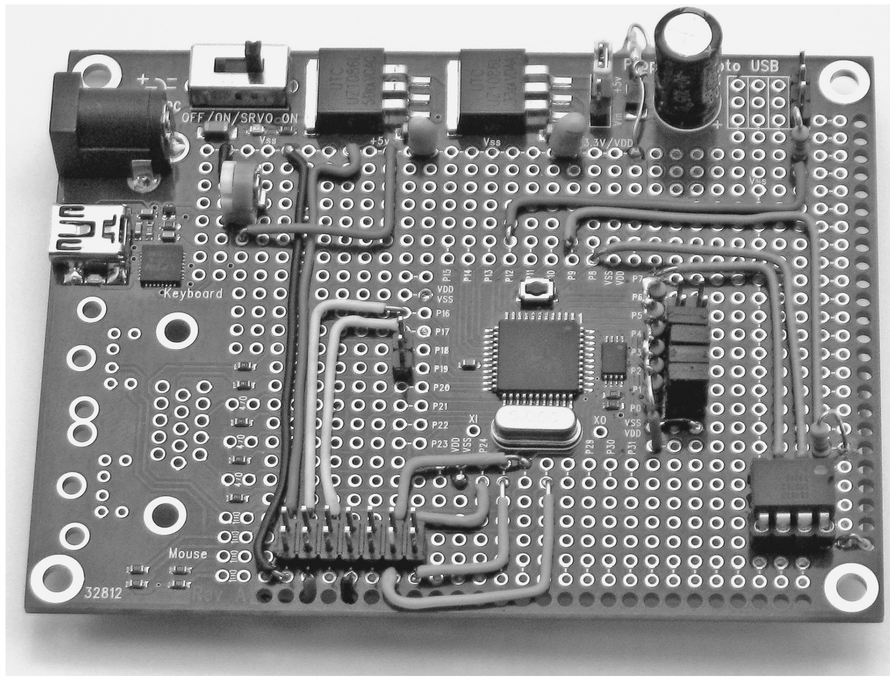

Figure 11-12 **The second prototype "room control" board.**

built using the low-cost Parallax Propeller Proto Board USB, and some rudimentary code was written to display current temperature and humidity, the "target" temperature for the room, and the position of the servo motor that would control the register.

Based on the success of the first board, a second board (Fig. 11-12) was carefully constructed by hand based on the layout of the first prototype.

As we expected these boards to communicate via CAT-5 cable, we needed to add RJ-45 jacks. The only PC-board-mount RJ-45–style connectors we could find did not have .100" pin spacing, making them rather difficult to install on the Proto Boards. Hand routing wires also led to a couple of mistakes, resulting in time being spent troubleshooting the second Proto Board. At this point, it was becoming clear that building each board by hand was going to be more time-consuming than we originally thought and more error-prone than we would like, especially considering how many boards would need to be built.

There would be five rooms in the model house requiring a room control board, and we wanted to have at least one board prebuilt and programmed as a spare to make it easy to repair the model if there were failures. In addition, we wanted to have some boards to use for testing and programming without having to be in possession of the entire model (which, at 3 feet × 4 feet in size, was decidedly nontrivial to move!). We hit upon the idea of creating "daughterboards" that would attach to the Propeller USB Proto Boards via female pins soldered around the surface-mounted Propeller chip, as shown in Fig. 11-13.

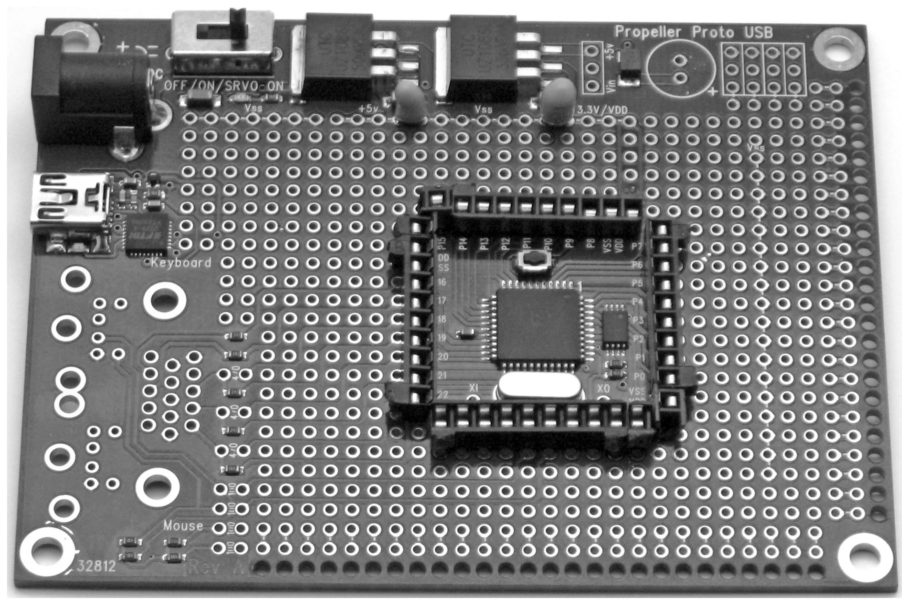

Figure 11-13 **Female .100" headers soldered onto the Propeller USB Proto Board.**

We then laid out a schematic for the room control board (Fig. 11-14) where all the components would be populated on the daughterboard and would connect to the underlying Propeller Proto Board through standard .100" spaced pins. Not only would this design allow us to create multiple room boards, it would also allow us to use the PC-board-mount RJ-45 jacks. This design had the added benefit of being "repairable." In the event of a failure of the surface-mount Propeller chip, we could simply "swap" the Propeller Proto Board with a new one, as none of the room board components were soldered to the Propeller Proto Board.

Using the daughterboard design had enough advantages that we wanted to forge ahead. However, we didn't want the associated "lag time" and expense of having a commercial board house etch the boards for us, so we went ahead and auto-routed the schematic to create a printed circuit board but we then converted to a "trace isolation" style board (Fig. 11-15) so it could be cut and drilled using the three-axis CNC system available in-house.

A Three-Board Solution As this method of creating boards worked rather well, we decided to use the same process to create daughterboards for the additional functions that would be required. For example, each room board would need to be able to send data back to a "master board." The master board collects room temperatures and calculates servo positions to control the air registers in each room. A new schematic was created for the master board, and it was then cut on the CNC system using the same board size and pin footprint as the "room board."

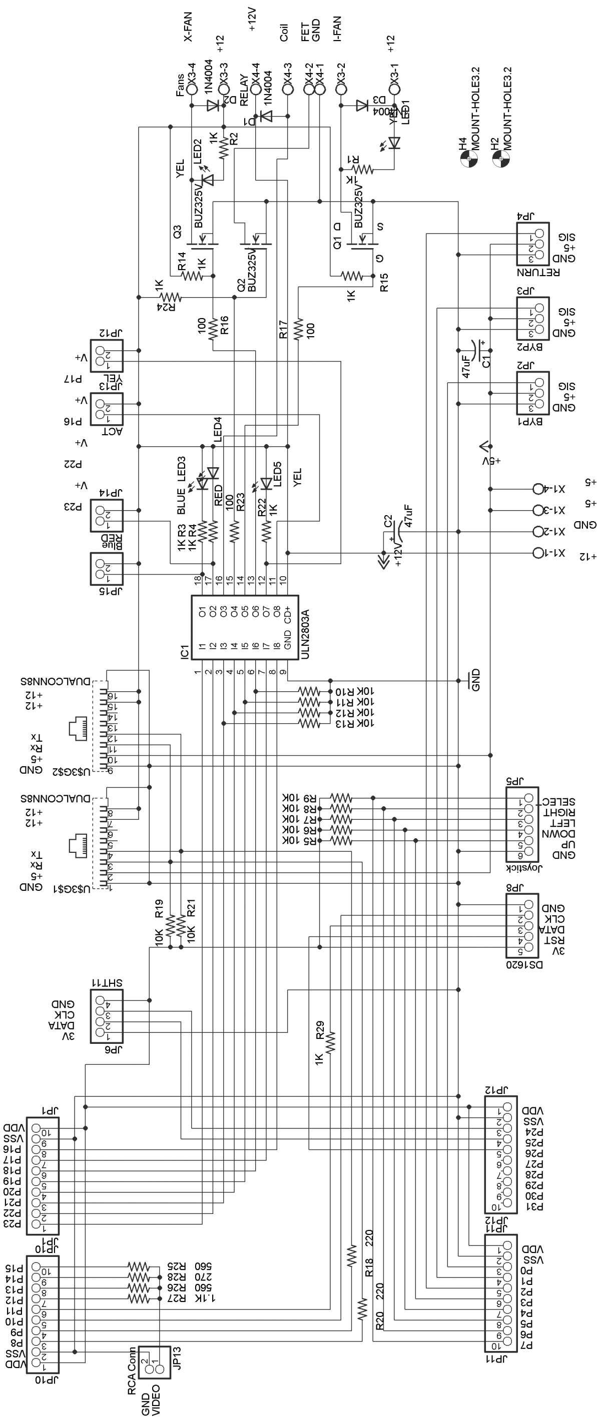

Figure 11-14 Preliminary schematic of the "room board" daughterboard.

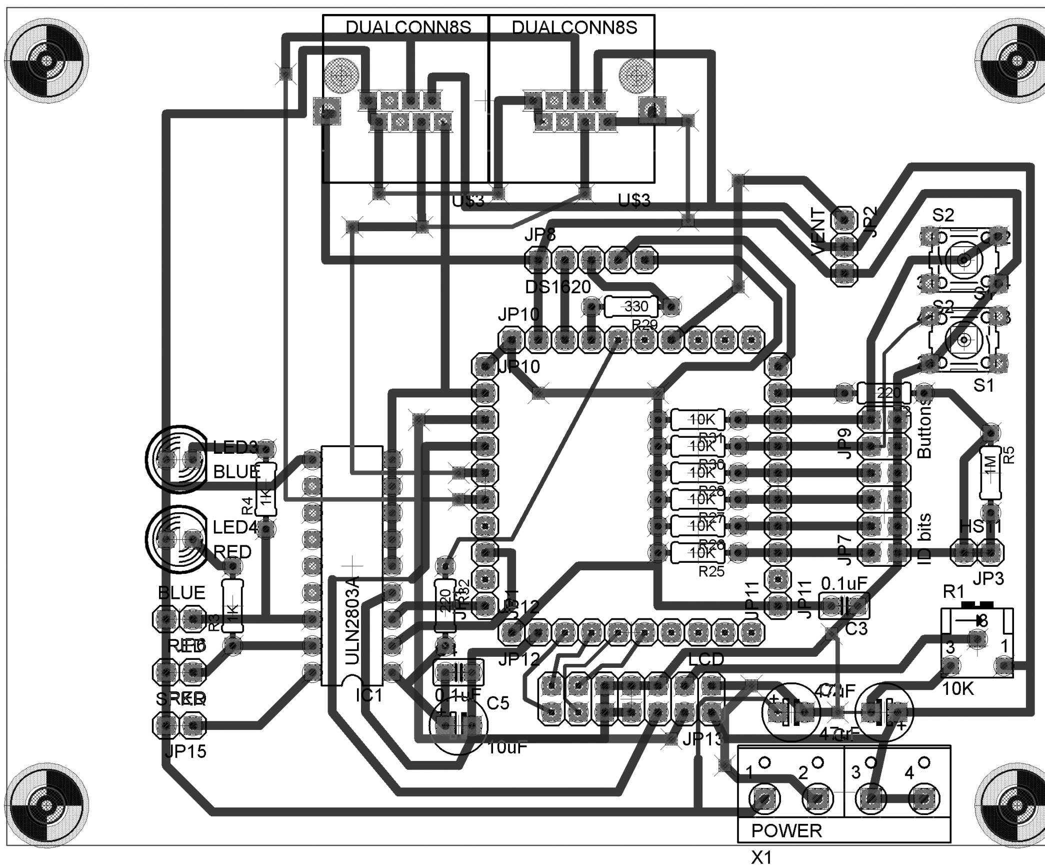

Figure 11-15 **Room board daughterboard design ~~using trace isolation~~.**

At first, this master board had both an NTSC output to be used for a GUI and the components to control the blower fans and heat pump. After thinking about "real-world" requirements, we decided that it would be more realistic to have the "control" components located outside the house or in the attic near the HVAC equipment. Subsequently, we decided that a third board would need to be created that would host all the high-amperage control components that would operate the heat pump and the blower fans. This board would also host a sensor that would read the *outdoor* ambient temperature and humidity to be used for determining if the outside air were suitable to use for indoor heating or cooling. As this board would be nearer the blower fans (in the cooling tower), we placed headers for servo motors to control the cooling tower bypass doors, as well as the return-air bypass vent flap. With its feature set complete, we dubbed this new board the "control board."

Another round of autorouting, and CNC cutting/drilling now left us with three board styles: "room," "master," and "control," all with the same physical size and pin configuration but with different component loads and roles. Figure 11-16 shows the results.

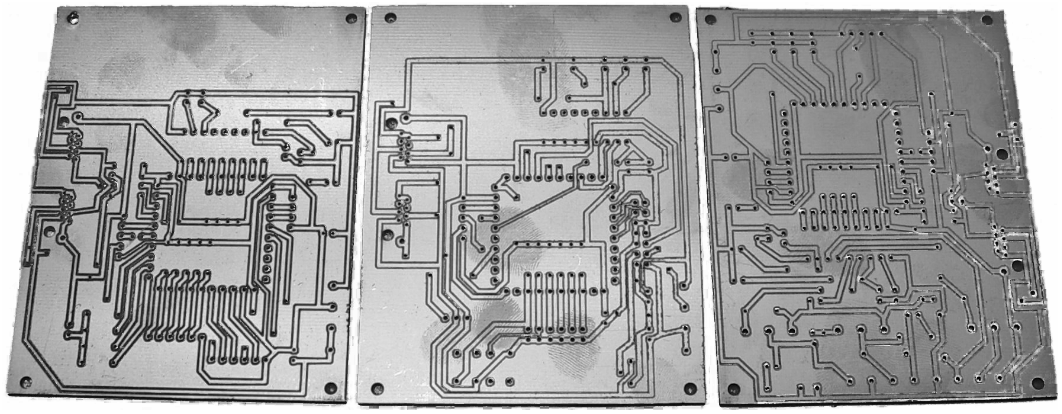

Figure 11-16 The "room," "master," and "control" daughterboards before being stuffed and soldered.

After stuffing, soldering, and carefully testing the daughterboards and the master board, we attached them to the Propeller Proto Boards and used aluminum standoffs on the four corners to stabilize the connections. The resulting board sets, shown in Figs. 11-17 and 11-18, were robust, compact, repairable, and reproducible.

So, now that we had the hardware designed for the house, it was time for some software design.

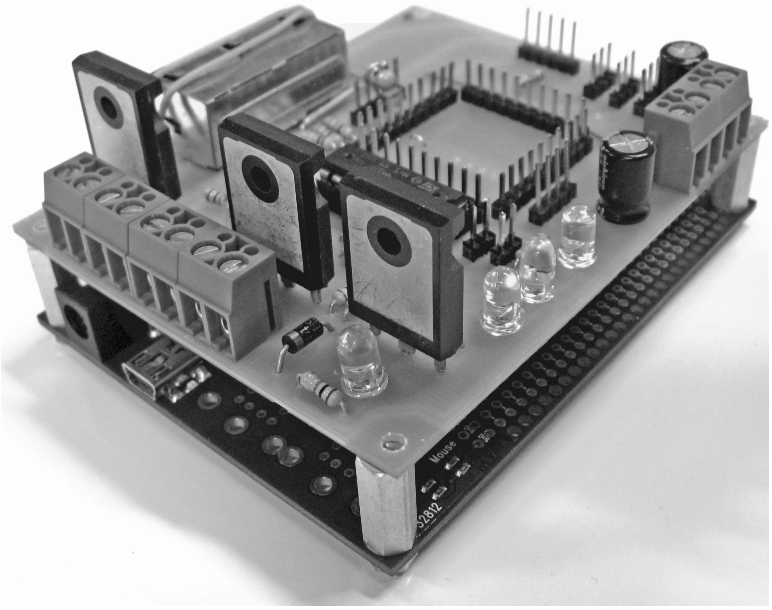

Figure 11-17 The "control" daughterboard mounted on the Propeller Proto Board.

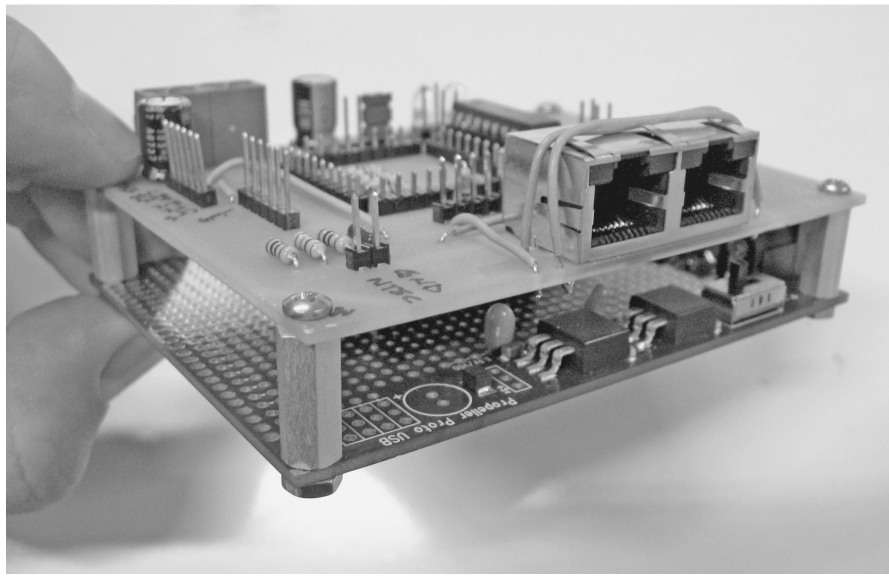

Figure 11-18 **The "master" daughterboard mounted on the Propeller Proto Board.**

SOFTWARE DESIGN CONSIDERATIONS: WHAT GETS PROCESSED WHERE?

Having processing power in each room made some functions intuitively obvious. Here is a listing of the functions that could easily be handled by the single Propeller chip in each room:

- Servo PWM generation
- Parallel LCD control
- Pushbutton detection/debouncing
- Temperature sensor polling
- Humidity sensor polling
- Status light control
- Target temperature display
- Communications (TX/RX)

Something to remember is that the design of the Propeller chip lends itself well to providing *parallel* functions and services. For example, servo motors require a steady stream of pulses in order to maintain their position. Therefore, it is possible to use an object to constantly supply these pulses. You need only tell the servo control object at what position the servo should be, and the object takes care of the rest. The same can be said for the parallel LCD. There are Propeller objects that make interacting and updating a parallel LCD as simple as interacting with the more expensive serial-controlled LCD units. A serial communication object can handle the complexity of sending and receiving data over a network with other devices. So, based on these capabilities, the room board would handle the servo motor, the LCD, the temperature sensors, the humidity sensor, user interface status lights, and network communications.

Since we knew there would be more than one room board, and we were hoping to reduce the wiring costs by using a daisy-chain wiring configuration, we decided to use an "open collector" communication approach and allow all the boards to share the transmit and receive pins. To make sure that only one board at a time would be transmitting, a polling method of communication was chosen where the master board would call each room board and only then would the room board send an answer to the master board. To differentiate the room boards, we placed a bank of four jumpers connected to Propeller pins P0-P3 to allow us to set an ID number, allowing a maximum of 16 room boards on our "network" at one time. The room boards would simply ignore any poll request that did not begin with their unique address.

Since the master board needs only two pins to perform its duties as the network master (i.e., just transmit and receive lines), it has plenty of pins (and processing power!) to allow for an advanced intuitive graphical user interface. By using the NTSC video output object, it is a straightforward process to display statistical information about system performance and present an advanced graphical user control interface through an embedded NTSC video monitor. For our experimental platform, we are currently implementing the NTSC video output to display current temperature, target temperature, and vent position for each room on a Parallax Mini LCD A/V color display (Fig. 11-19).

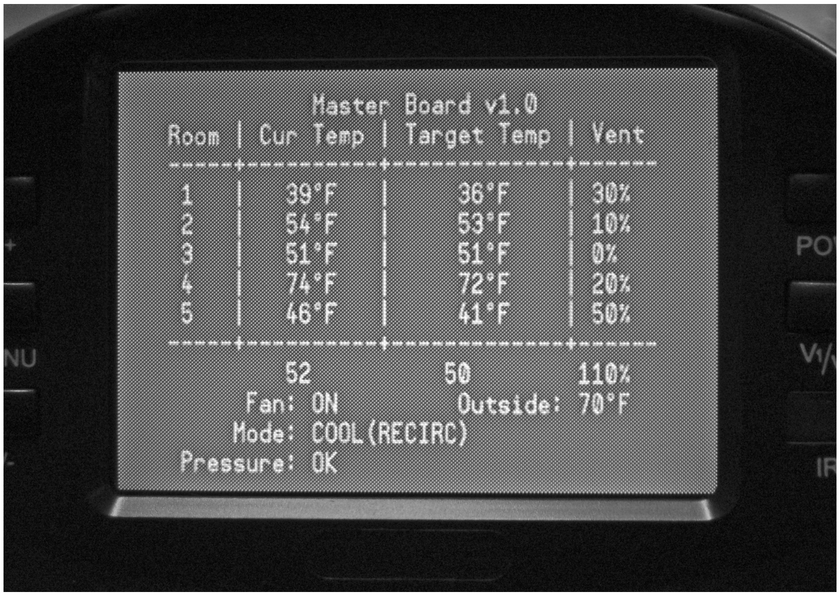

Figure 11-19 **The NTSC monitor with the preliminary system test data displayed.**

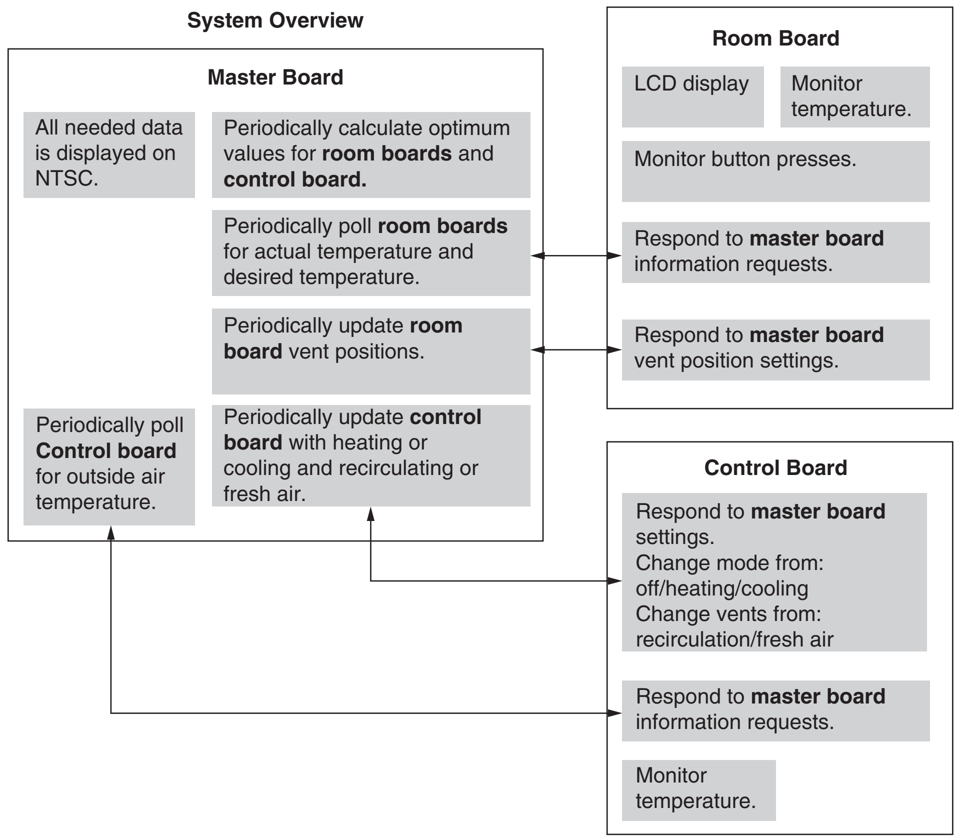

Figure 11-20 **System overview flow chart.**

The main job of the master board is to poll each of the room boards, retrieve their current temperature, retrieve the desired temperature, and then perform calculations that are used to direct the operations of the rest of the system. A quick recap of all the boards, their functions, and their interactions is shown in Fig. 11-20.

The point of building the HVAC green house was to create a test platform that would allow experimentation with different approaches to air system management and to observe how these approaches perform in this small-scale environment. Though full-scale environments may react differently, experimentation on a small scale can be quite useful in determining what approaches to use in a full-scale environment.

The flow chart in Fig. 11-21 is the first attempt to use an algorithm to balance and maintain the air temperature in the building.

As you can see in Fig. 11-21, we first calculate the average internal air temperature and then compare that with the outdoor air temperature to determine if the system should be in “heating” or “cooling” mode. The Spin code to accomplish that is listed here:

Note: Lines marked with the “←” symbol are intended to be on a single line and wrap here due to space constraints.

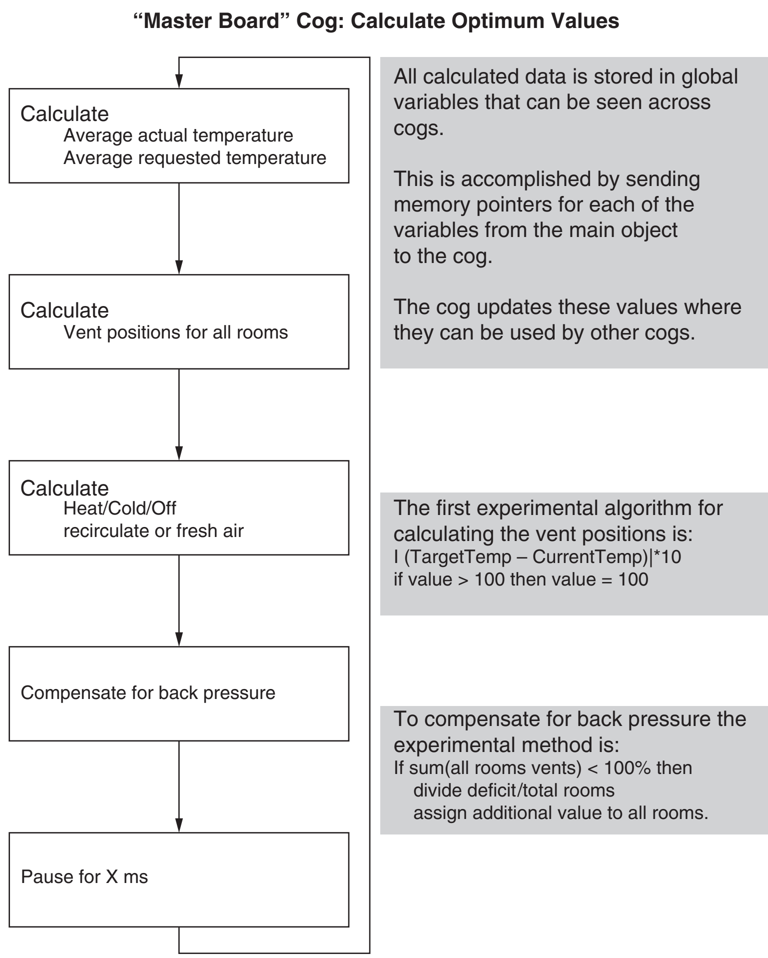

Figure 11-21 Master board cog flow chart.

```
' Calculate Room averages
  OverallAverageTemp10X:=0
  OverallAverageDesiredTemp10X:=0
  repeat count from 0 to 4
    OverallAverageTemp10X:=OverallAverageTemp10X+ ←
      LONG[RoomCurrentTemp10X+(count*4)] ←
    OverallAverageDesiredTemp10X:=OverallAverageDesiredTemp10X+ ←
      LONG[RoomTargetTemp10X+(count*4)] ←

  OverallAverageTemp10X:= OverallAverageTemp10X/5
  OverallAverageDesiredTemp10X:= OverallAverageDesiredTemp10X/5
```

Vent positions are then calculated for each room based on its "need" according to its temperature reading. Again, the Spin code to accomplish this is straightforward:

```
' Calculate Vent positions
  totalvent:=0
  repeat count from 0 to 4
    temp:= || (LONG[RoomTargetTemp10X+(count*4)]- ←
      LONG[RoomCurrentTemp10X+(count*4)]) ←
    if temp > 10
      temp:=10
    temp:=temp*10
    LONG[RoomCurrentVentPosPercent+(count*4)]:=temp
    totalvent:= totalvent + temp
```

Next, using the calculated average internal air temperature compared to the external air temperature allows us to automatically determine if the Peltier solid-state heat pump should be operating in a "cooling" or "heating" mode:

```
'''''''''''' HEATING ''''''''''''
  if OverallAverageDesiredTemp10X > OverallAverageTemp10X ' Heating
    if OverallAverageDesiredTemp10X < outsideTemp
      'Fresh
      LONG[BlowerFanPower] := POWER_ON
      LONG[PeltierPower] := POWER_OFF
      LONG[PeltierDirection] := PELTIER_HOT_INSIDE
      LONG[AirSupply] := FRESH_AIR
    if OverallAverageDesiredTemp10X > outsideTemp
      'Recirc
      LONG[BlowerFanPower] := POWER_ON
      LONG[PeltierPower] := POWER_ON
      LONG[PeltierDirection] := PELTIER_HOT_INSIDE
      LONG[AirSupply] := RECIRCULATE

  '''''''''''' COOLING ''''''''''''
  if OverallAverageDesiredTemp10X < OverallAverageTemp10X ' Cooling
    if OverallAverageDesiredTemp10X > outsideTemp
      'Fresh
      LONG[BlowerFanPower] := POWER_ON
      LONG[PeltierPower] := POWER_OFF
      LONG[PeltierDirection] := PELTIER_COOL_INSIDE
      LONG[AirSupply] := FRESH_AIR
    if OverallAverageDesiredTemp10X < outsideTemp
      'Recirc
      LONG[BlowerFanPower] := POWER_ON
      LONG[PeltierPower] := POWER_ON
      LONG[PeltierDirection] := PELTIER_COOL_INSIDE
      LONG[AirSupply] := RECIRCULATE
```

Once the mode of operation is determined, a comparison between the average desired temperature and the current outdoor temperature could be made. If bringing the outside air into the rooms would allow them to reach their target temperature without activating the heat pump, the control board may be instructed to open the cooling tower bypass doors, allowing outside air to be sent into the rooms.

This "fresh air" mode of operation could also be used in conjunction with various gas sensors to react to an unhealthy air situation. For example, a buildup of carbon monoxide due to leaving a fire burning, a propane leak from a stove/heater, or even a broken sewer line allowing methane into the building all could be dangerous, even life-threatening situations. It would be possible, using sensors currently available from Parallax, to detect these situations and not only alert the building occupants by sounding an alarm, but also activate the "fresh air" mode.

This would bring fresh air indoors and vent the dangerous gases out of the building. In the event that a fire is detected, every room vent could be instructed to close in order to reduce the spread of smoke through the building and keep from feeding "fresh air" to the fire.

Another real-world requirement has to do with *backpressure.* When the blower fan turns on, it generates a specific number of cubic feet per minute of air. In order to operate properly, a certain amount of air must be allowed to exit the ducting system. If the room air registers are closed, backpressure may develop, possibly causing the condenser coil to freeze up, the blower fan to overheat, and the compressor to be damaged. With such serious consequences, avoiding backpressure in the system is a major concern.

Though a Peltier solid-state heat pump does not suffer from backpressure issues the way a typical HVAC compressor-based system would, we decided to implement the backpressure detection correction system in anticipation of dealing with real-world issues.

To make the concept both easy to implement and to understand, we decided to use the equivalent value of "100 percent open" of one air register as the minimum outlet amount that must be available whenever the blower fan and heat pump were running. This makes it simple, as it can be expressed as four of the room's registers set to 0 percent and one of the room's registers set to 100 percent, or as each of the five rooms' air registers set to a position of 20 percent open. A simple backpressure routine was added to the master board software to manage distributing the "error" so backpressure would automatically be regulated by the system.

The Spin code to implement our backpressure detection is shown here:

```
' Check for Pressure Buildup
  pressure:= 0 ' Reset amount of Vent Openings
   ' We assume here that at least 1 vent must be at 100% or
   ' the total sum of all open vents should add to 100%
  repeat count from 0 to 4 ' Calculate Total Vent Openings
    pressure := pressure + RoomCurrentVentPosPercent[count]

  if pressure < 100   ' If this is true then there is not enough venting
    ' need to allocate the remainder
    totalvent:=0
    overpressure := 100-pressure
    splitpressure:= overpressure/5
```

```
    pressure:= overpressure
    repeat count from 0 to 4
      RoomCurrentVentPosPercent[count]:= ←
         RoomCurrentVentPosPercent[count]+ splitpressure ←
      totalvent:= totalvent + RoomCurrentVentPosPercent[count]
  else
    pressure:= 0
```

Summary

The Propeller-powered HVAC green house, shown in Fig. 11-22, allowed us to explore parallel processing and distributed computing. We delved into sensor sampling, network communications, servo motor control, video display, and data recording.

The simplicity of the Spin language, combined with the power of objects, makes it straightforward to convert theories into test cases. It also allows us to avoid *labor inertia*—that reluctance we all have to abandon a design after discovering a fatal flaw or a simpler/better way to accomplish the task. The Propeller chip, with its multiple processors and large library of preexisting software objects, allows us to change direction easily and reduce the amount of time required to see results.

The "Exercises" section lists a small sample of the things we have considered exploring with our HVAC green house research platform. As real-world design criteria were taken into account when building the system, it may well be possible to implement the system

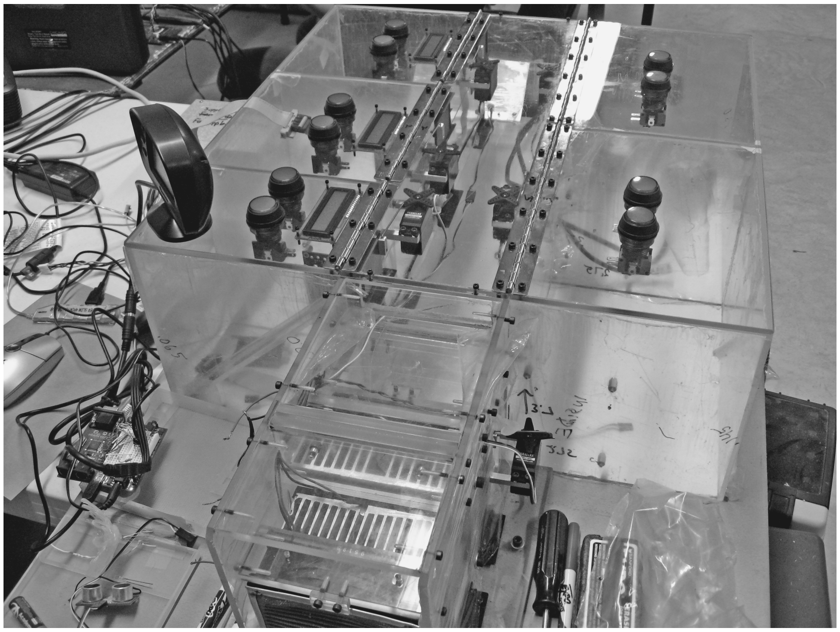

Figure 11-22 **The Propeller-powered HVAC green house hardware.**

in a new home or to retrofit an existing home. I would imagine it will only be a matter of time until someone implements a system such as this in an attempt to see if they can cut their heating and cooling costs, as well as make their home more comfortable.

IN CONCLUSION

At the time of this writing, we are continuing to develop software and test hardware for the model We have recently obtained the liquid propane, methane, and carbon monoxide detectors, and plan to add the emergency venting routines to the model, as shown in Fig. 11-23.

Note: If you would like to download the complete source code sets for all the boards in the HVAC green house model, as well as view high-resolution color pictures of the unit's construction and videos of the unit in action, please visit: ftp.propeller-chip.com/PCMProp/Chapter_11.

I would also like to take this opportunity to thank the team of amazing folks responsible for the creation of the HVAC green house model:

Rick Abbot—Machining and mechanical fabrication

Jake Ivey—Carpentry

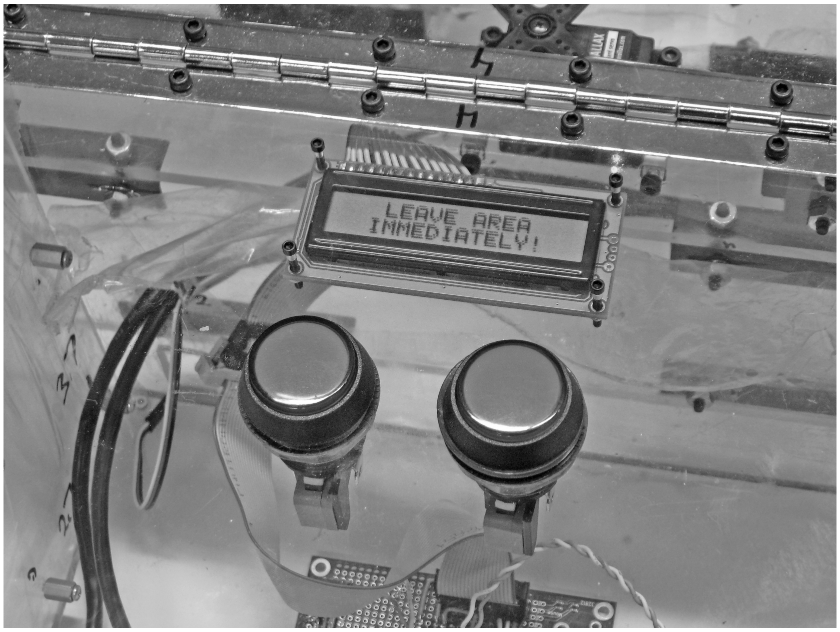

Figure 11-23 **The room board LCD display when unsafe air conditions are detected.**

Paul Atkinson—Schematic design, PCB fabrication and assembly

Gray Mack—Hardware prototyping, software design and implementation

James Delaney—Software design, 3-D CAD design, and system assembly/testing

Parallax Inc.—Component supplier

I would also like to thank André LaMothe and Ken Gracey for inviting me to participate in this book and Chip Gracey for making the Basic Stamp and the Propeller chip a reality!

Exercises

During the creation of the house and the various discussions that ensued, some ideas were tossed around but not implemented that I thought might be a good "jumping off" point for anyone wanting to explore a system similar to this one. Some of the idea would require little or no modification to the basic system as built.

1 Adding an IR (infrared) sensor to the master board. By adding an inexpensive IR receiver module and using the SIRC object to decode Sony IR signals, you could essentially use a handheld IR transmitter to control the system. In addition, the NTSC video output could be routed to a conveniently located TV monitor. This would allow the operator to review and change the system settings from their home entertainment system or to monitor the HVAC conditions using "picture in picture" technology available on many TV sets.
2 Adding an IR sensor to the room boards. The same IR sensor and SIRC object, when added to the room board, would allow the room occupant to control the temperature using a simple IR remote control (for example, use "channel up" and "channel down" signals to adjust the temperature set point).
3 Add speech synthesis to provide voice alerts. During an alarm condition (i.e., detected unhealthy air or fire), using verbal alerts would allow the room occupant to know to what the alarm issue relates without having to be close enough to the board to read the LCD screen. In addition, if an IR remote was being used to adjust the target temperature for the room, voice feedback could be used to "tell" the user what temperature they have selected
4 Select some rooms for "fresh" air and others for recirculated air. If some occupants preferred "fresh" air in their rooms, while others preferred recirculated air, it would be possible to meet both preferences by alternating the air-handling method between the rooms. To accomplish this, the air register in a selected room (or rooms) would be set to 100 percent open, while the air registers in the remaining rooms would be placed in the 100 percent closed position. Next, position the "bypass" doors to the "fresh air" positions, and start the blower fan. After a few moments, reverse the process for the rooms that have selected recirculated air.

5 Mix recirculated and fresh air. As the bypass doors are controlled by servomotors, they may be commanded to hold the doors in any position from 100 percent open to 0 percent open. So, it would be possible to set a "mixture" amount if you want to assure a certain amount of fresh air is moved through the building on a scheduled or regular basis to reduce odors and to remove "stale" air from the building.

6 Error detection and alerts. If a room fails to reach the desired temperature in a reasonable amount of time, the master board could show an alert or trouble condition. Another possible error-detection function could be if the room board has not been polled by the master board in a specific period. When this "timeout" has been reached, the board could show a "trouble" message on the LCD to alert the room occupant that the board needs diagnostics and repair.

7 Alert required preventive maintenance. The master board is able to track the "run time" for the blower fans so it would be possible to alert the user when filter replacement intake cover cleaning and other maintenance may be required.

8 Go wireless! As was shown earlier in this book, it is possible to forgo the use of CAT-5 cables for communication altogether. If the room boards, master board and control board were all equipped with ZigBee modules and powered by a small "wall-wart"–type power supply, it may be possible (depending on the distances involved) to retrofit an existing home HVAC system without the additional cost and disruption of running cable through the building.