

# EE459Lx/MKT446/FA402

Spring 2010

## Residential Thermostat

### Project Overview

This project is a device for controlling the heating and air-conditioning in home. Many older homes have simple thermostats that turn on the heating when the temperature falls below a set level, and turns the heat off once it rises above a set level. More advanced devices can be programmed to turn change the desired level based on the time of day. For example, the home is allowed to get colder at night and then warm up in the morning. Many of the thermostats available in home improvements stores have this capability. The goal of this semester's project is to improve on what is commercially available by utilizing state-of-the-art technology and sound design practices.

- Make it easier to operate by improving the user interface. Remove the excuses for wasting energy heating or cooling the house like "My husband/wife/son is the one who programmed it and I don't know how to adjust it so I just let it run."
- Make it more reliable. Thermostats are usually battery powered but can loose their settings if the main battery runs out and has to be changed. The internal clock is also affected by a loss of power.
- Add additional features that will appeal to consumers.

### Design Requirements

The requirements stated below for the basic project can be implemented in a variety of ways. It is up to the design team to decide how to build their project. In all projects involving microcontrollers there are ways to build the project that require more hardware and less software, and vice-versa. The design team must decide whether to implement the various parts of the design in hardware or software. Design trade-offs in areas such as reliability, manufacturability, ease of use, cost, etc., should also be analyzed to determine which is the best one to use.

In addition to the baseline requirements, students are expected to think of other features that would add value to the product and if possible add some of these features to the project. Still other features can be described in the final report as features that should be considered for inclusion if and when the device goes into production. Possible design options can be found by studying the features that other manufacturers have included in similar devices.

The thermostat must have the ability to have separate settings for each day of the week. For each of these days, the 24-hour day should have at least four times where the temperature can be changed to a new setting. The user has the option of programming a different temperature for each of these time periods. For example, they might program Tuesday as follows:

6:00 AM	68 degrees	(warm the house up for getting up in the morning)
7:00 AM	62 degrees	(let it get colder while at work or school)
5:00 PM	70 degrees	(warm it up for an evening at home)
11:00 PM	63 degrees	(let it get colder while in bed)

Since many people keep the same schedule Monday through Friday, it should not be necessary to program each weekday individually. Your project should have the ability to copy the settings for one weekday to all the other weekdays if the user so chooses.

In addition to the programmed settings, the user must have the ability to easily override the programmed settings for the remainder of the time period. For example, the user has to get up this morning an hour earlier so they want the heater to come on without having to reprogram the thermostat just for today's activities.

### **Inputs and Outputs**

The device must have the ability to control a standard heating and air conditioning system by switching 24VAC power. The 24VAC power supply is not a part of the project, but your design must incorporate something that allows it to control the application of the 24VAC power to the control inputs of the HVAC unit. This can be done in a variety of ways (relays, triacs, etc.) and your team will have to decide on a method. The thermostat must be able to control the cooling and heating, and also have the option of running the unit's fan for air circulation.

The user needs to have the ability to program the device, and it must have some form of output that the user can refer to when programming it. The input could be via buttons, keypad, IR remote control, Bluetooth, etc. The output could be 7-segment display, an LCD panel, etc. The choice of how the user interfaces with the controller can have a significant effect on both the ease of use, and the price of the device. For example, more buttons should make programming simpler, but the cost goes up.

### **Memory**

The device should have the ability to retain its internal information if power is lost. The user should not have to re-enter all the programming data if the device's power is lost for some reason. The settings can be stored in some sort of non-volatile memory such as an EEPROM or the device can be provided with long-term battery backup.

### **Timing**

There are several ways to implement the time-keeping ability of the controller. The team should examine the options and decide on the one that works best for the product. Many commercial thermostats do not continue to keep track of the time of the day if there is a power outage. This can lead to errant operation once the power is restored and your design should solve this problem. If there is a power outage, it is not required that the thermostat continue to operate normally during the outage, but when the power is restored the controller's internal clock should have the correct time and not require the user to reset the clock. It is also not required to continue to operate the controller's displays during the outage since these can consume a significant amount of backup power.

When testing the device it would be very useful to have the option to make it run much faster than normal. For example, the design might include a switch that makes the internal clock advance one minute for every second of actual time, or one day for every minute, etc.

### **Optional Features**

The design requirements stated above are for the basic controller and all product teams should try to design their product to at least comply with those requirements. However all teams are encouraged to go beyond the basic requirements in order to create a more marketable product. The choice of what other features to add to the design should be based on the market research as to what consumers want in the

product and how much they will be willing to pay. The additional features should add value to the project and hopefully will result in more people choosing to buy it. However the team must remember that each added feature will add to the product's cost so adding lots of interesting features may force the price above what many consumers would be willing to pay.

Teams must remember that they are responsible for coming up with a design that they can build during the time scheduled. Be careful to avoid "mission creep" where more and more features are added to the design until it becomes too bloated to be constructed. Do not allow optional features to get in the way of designing and constructing the baseline project.

If a team comes up with a new feature that is somewhat outside the product specifications and adding it may affect their ability to finish the baseline project, they can discuss the implications of adding this feature with the instructor. In some cases teams may be granted permission to pursue variations on the specified product in order to incorporate technologies they find interesting.

## Milestones

Each team is required to complete certain parts of the project and demonstrate that they have them working by a specified date. The purpose of this is to make sure no team gets too far along in the semester without having some very fundamental parts of the project working.

1. Power circuit: Install the connector for power and ground and confirm that you can hook your board to the lab power supply and apply power to it without shorting out the power source.
2. Timing circuit: Show that the logic board master clock is functioning and that any lower frequency clocks are being properly generated from the master clock. This can be shown using one of the oscilloscopes in the lab.
3. Microcontroller: Install the microcontroller and demonstrate that it is functioning and that you can program it. Write a small program that does something to confirm that the microcontroller is working. It is very important to get the micro working early in the project so it can then be used to test other parts of the project.
4. Display: Show that you can output something on your choice of display. Your display can be a very useful part of your debugging since once it is working you can use it to output information about what is happening with the rest of the system.
5. Inputs: Demonstrate that your input controls can be read by the microcontroller. For example, if you are using pushbuttons, show that each time the button is pressed something changes on the display
6. Valve control: Show that your microcontroller can cause a valve to turn on and off. This implies you have figured out how to make a digital signal control the 24VAC to the valve.

## Specifications

The following are some specifications as to how the project is to be constructed.

1. Most of the project circuitry must be built on a single prototyping board. These are provided in the class and are approximately 6 x 9 inches. If teams choose to build their project on some other board they must first get permission from the instructor. If the design a separate secondary device that communicates with the main unit it can built on a second board.
2. All signal and power connections to the project board must be made using connectors and wiring that make for a reliable connection. For example, using clip leads to attach to a pin of an IC socket should not be used for anything more than the very first test of the board. A person trying to use your board should be able to see immediately where the power and signals are connected. They shouldn't have to be told to "use a jumper to connect the +5 volts to this pin over here."
3. The design should be based on a Freescale MC68HC908JL16 microcontroller. Projects may use multiple microcontrollers if needed. The use of a different microcontroller must be approved by the instructor.
4. The microcontroller should be installed in a zero-insertion force (ZIF) socket in order to make it easy to remove it for reprogramming. When planning where to mount the various IC sockets on the board, make sure to leave extra room for the ZIF socket since it is longer and wider than a normal socket.