Thermostat Reference Design Using the MC9S08LL16

Devices Supported:
MC9S08LL16

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd. Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Chapter 1
Overview

Chapter 2
Hardware Description

Chapter 3
Firmware Description

Chapter 4
User's Guide

Appendix A
Schematics

Appendix B
Temperature Sensor Coeffecients

Chapter 1
Overview

1.1 Introduction

A thermostat regulates the temperature of a system, room, or building, keeping the temperature at a desired level. There are different ways to get the temperature measurement in an electronic system, for example the use of a thermistor, thermocouples, and in some cases integrated circuits.

To control the temperature a digital thermostat is typically connected to an HVAC unit. The thermostat provides voltage to the heater, venting, and air conditioning (HVAC) terminals indicating to the HVAC system, what should be turned on based on the current system temperature.

1.2 Freescale Solution

The thermostat reference design is an example of how a thermostat can be built taking advantage of the MC9S08LL16 features. The main features of this microcontroller that makes it ideal for low-power LCD applications are:

- Low power consumption with LCD configurations that support operation in low power microampere range, as low as 1.5 µA in the thermostat.
- 32 LCD pins, up to 196 segments with eight backplanes and 18 LCD-pins multiplexed with GPIO giving this module a lot of flexibility.
- Any LCD pin can be frontplane or backplane, simplifying the hardware routing and allowing the use of different LCDs with the same board.
- TOD module with quarter to second and second interrupt. It also has a comparator module that allows waking the MCU up to every 64 seconds.
- Low power blinking model enables an LCD blink without waking the controller that saves power consumption.
- 12 bit ADC that can be triggered by the TOD module.
- Wake from stop, based on time or keyboard event.

A thermostat application block diagram using the MC9S08LL16 is shown in Figure 1-1.
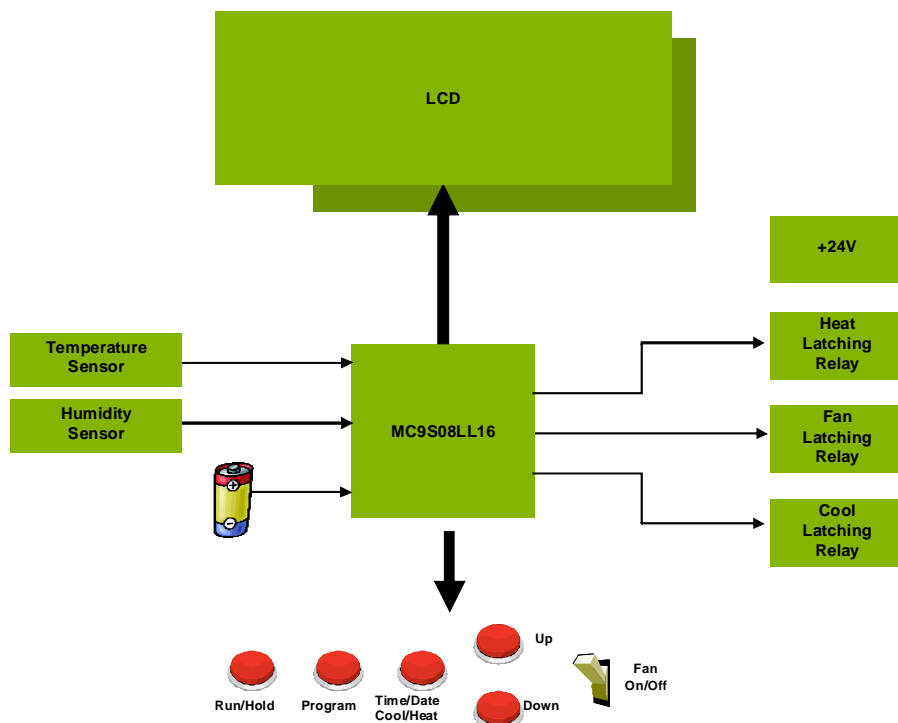
Figure 1-1. Thermostat block diagram

There are two versions of the thermostat reference design using the same hardware; one, with a 2 x 28 LCD (low-end thermostat) and the other with an 8 x 24 LCD (high-end thermostat). The two versions show the flexibility of driving small and large LCD panels and the power consumption trade-off that comes with it.
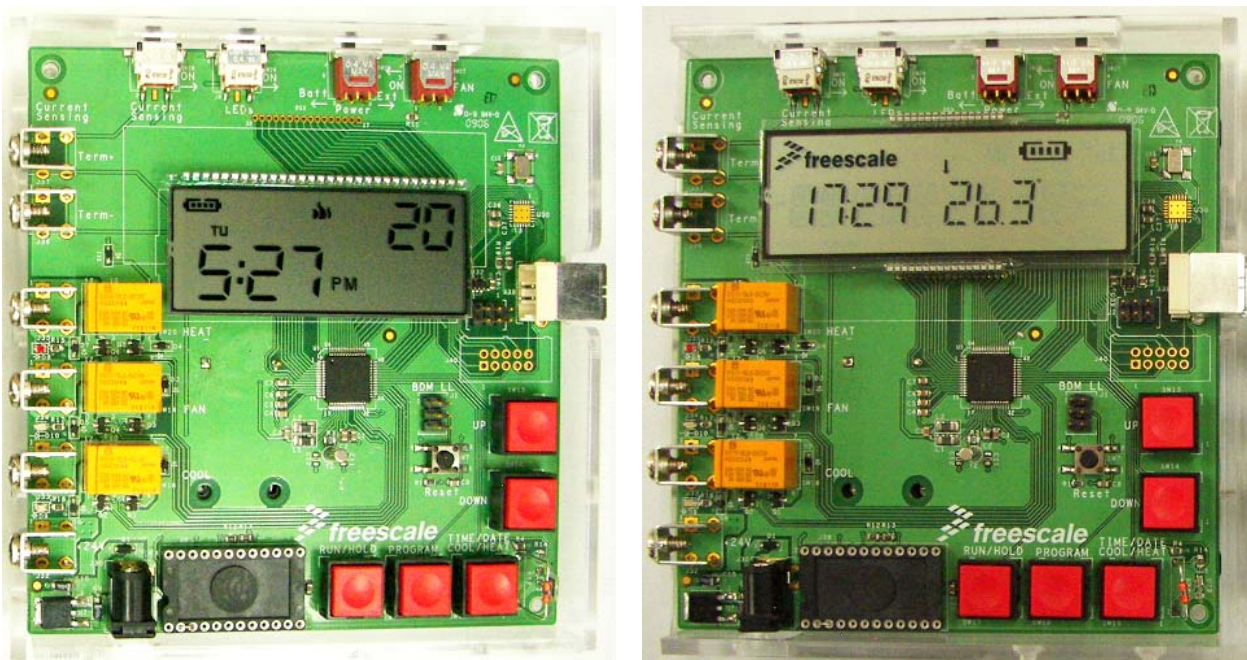
Figure 1-2. Low-end and high-end thermostat

Thermostat Reference Design Using the MPC9S08LL16. Rev. 1

1.3   Features and Advantages

The thermostat reference design has the following features:

- Accuracy of 0.1 °C
- Day-of-the-week, hours, and minutes calendar for the low-end version, and date, hours, and minutes for the high-end version
- Temperature display in °C or °F
- Programmable set points for heating and cooling
- Programmable run and hold mode
- Fan on and auto mode
- Standard HVAC connections
- Current sensing switch and screw-terminals to measure power consumption
- HVAC LED indicators
- Contrast control for the high-end version

Main features of the MC9S08LL16 family used in this reference design are:

- 2 x 28 (low-end) and 8 x 24 (high-end) LCD display to show the temperature, date, and time. Any LCD pin can be used either as frontplane or backplane simplifing the layout work.
- An LCD blink mode to display different messages without waking the MCU using a one second blink period.
- 12 bit ADC that gives more accuracy to the temperature measurement using a thermistor.
- TOD capable of waking the MCU from every quarter second up to every 64 seconds.
- Hardware triggered ADC to start the conversion after the TOD module wakes the MCU.
- Stop3 mode keeps the TOD and the LCD working using a typical 6 µA current consumption for the high-end and 1.5 µA for the low end.
- A very low power external oscillator used in stop2 or stop3 modes that provide an accurate clock source to the TOD. The external 32.768 KHz crystal does not need resistors or capacitors to work.

Chapter 2
Hardware Description

2.1 Hardware Implementation

The MC9S08LL16 is connected to the following hardware components:

- Temperature sensor
- Two different LCDs
- Five push buttons
- Four switches
- HVAC outputs

The reference design works with an external 32.768 KHz crystal. One of the advantages of the MC9S08LL16 family is; no external components are needed for this crystal configuration. The standard BDM connector is also available in the PCB to program any application.

2.2 Heating, Venting, and Air Conditioning-HVAC

This design uses a type of relay called "Latching Relay". Each relay has two coils-"set" and "reset". Each relay needs 180 mW nominal power to set or reset the coils. The current requirement of the coils is only for a few milliseconds. This type of relay has the advantage of keeping the current levels in control leaving the need to continuously drive the set and reset of the coils. This is ideal for battery operated systems where current consumption is critical for energy conservation.

This reference design uses a common +24 V AC circuit. It also has screw terminals for easy integration with HVAC equipment.

It is designed to operate a single transformer and work in the configuration found in Figure 2-1.
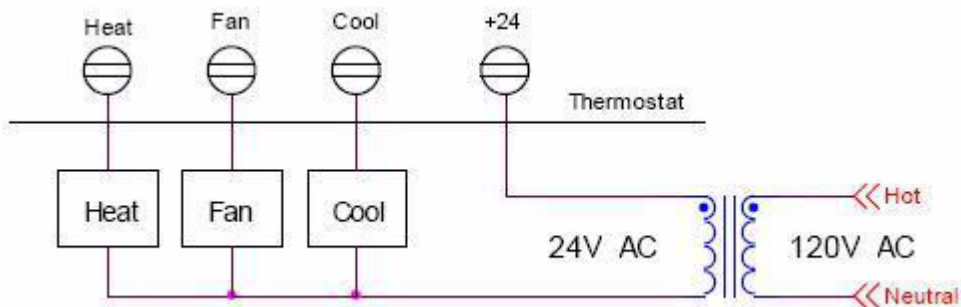
Figure 2-1. Configuration diagram

The heat, fan, cool, and +24 V connections described in Figure 2-1 can be found at the bottom left of the board. Screw terminals are used for the user's convenience.

A — Heat screw
B — Fan screw
C — Cool screw
D — +24 V screw
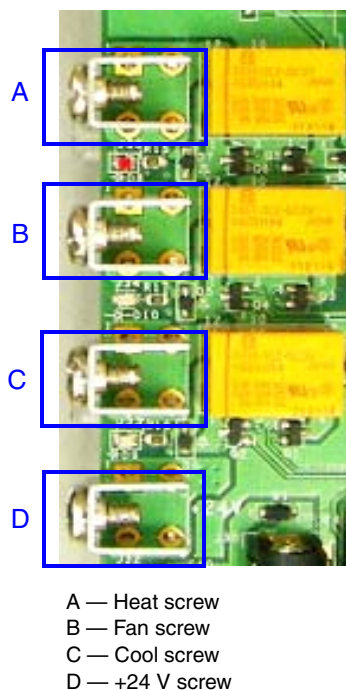
Figure 2-2. Physical location of the configuration

2.2.1   Temperature Sensor

This reference design includes a temperature sensor which is a variable temperature dependent resistor. A fixed 82 kΩ resistor is used to create a voltage divider. For example, this reference design uses a 100 kΩ temperature sensor. At 25 °C (77 °F) the temperature sensor coefficient is "1". Therefore, the voltage in the resistor divider network is 1.486 V.
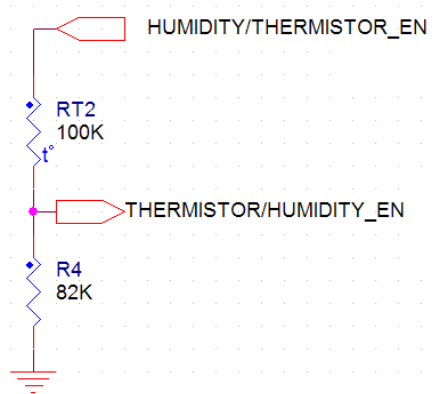
Figure 2-3. Temperature sensor resistor divider circuit

The temperature sensor manufacturer provides a list of values for the RT coefficient, it is possible to calculate the resistance and in this case the voltage present for each temperature. Please refer to Appendix B, "Temperature Sensor Coeffecients" for the complete range of temperature sensor coeffecients.

2.3 Thermostat Configuration and Controls

2.3.1 Buttons and Switches Identification

The thermostat is controlled with five buttons and four switches. Table 2-1 indicates where each button is connected to the microcontroller.

Table 2-1. Button and pin connection list

Button	MCU pin
Run/Hold	PTC7/$\overline{\text{IRQ}}$
Program	PTA0/KBIP0
Time/Date/Cool/Heat	PTA3/KBIP3
Up	PTA2/KBIP2
Down	PTA1/KBIP1

Table 2-2 indicates where each button is connected to the microcontroller and the functionality it serves.

Each button is connected to a pin capable of waking the MCU from stop mode to reduce power consumption. The buttons and switches functionality is described in detail in Chapter 3, "Firmware Description."

Figure 2-4 shows the location of each button.
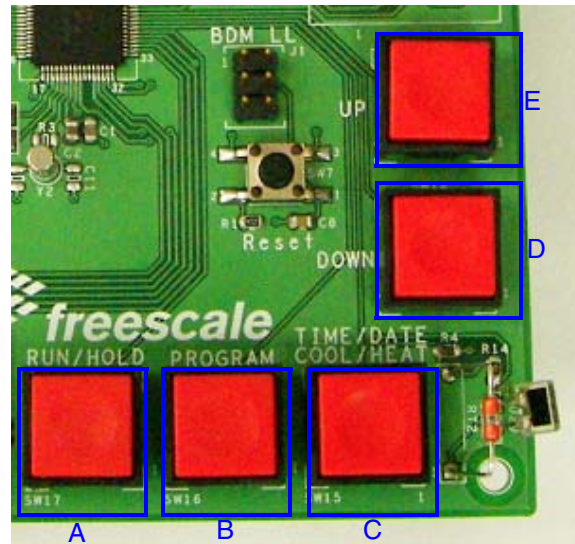
A — Run and hold
B — Program
C — Time/Date/ Cool/Heat
D — Down
E — Up

Figure 2-4. Thermostat configuration buttons

Most of the switches have only hardware functionality. The only switch connected to the MCU is the fan switch and its functionality is explained in Chapter 3, "Firmware Description." The other switches select the power supply source, turn on or off the indication LEDs, or activate the current measurement for the MCU. Table 2-2 indicates where each button is connected to the microcontroller and the functionality it serves, Figure 2-5 shows their position on the board.

Table 2-2. Switches and pin connection list

Switch function	Position / function	MCU pin
Current Sensing	Right Side / On	—
LEDs	Right Side / On	—
Power	Left / Battery – Right / External Power	—
Fan	Left Side / On	PTB4
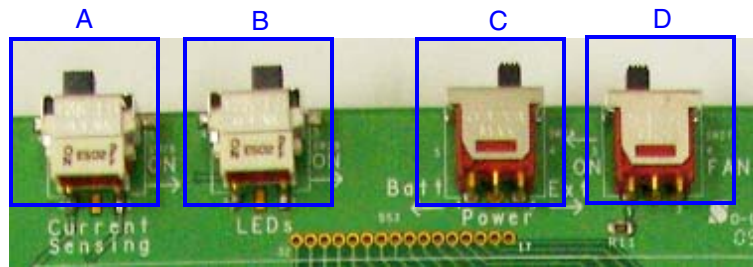
A — Current sensing
B — LEDs
C — Power source
D — Fan

Figure 2-5. Thermostat switches

2.4 Additional Hardware Features

2.4.1 Current Sensing

One of the key features of the MC9S08LL16 is its low power consumption. This reference design highlights this.

To measure the current, connect a multimeter to the TERM+ and TERM– screw terminals. After connecting the multimeter move the current sense switch shown in Figure 2-5 to the on position. This measures the current consumption for the digital section of the board (microcontroller, LCD, switches, buttons, and temperature sensor).

The location of the current sensing terminals can be found on the left side top part of the board, as shown in Figure 2-6.

A — Term+
B — Term-
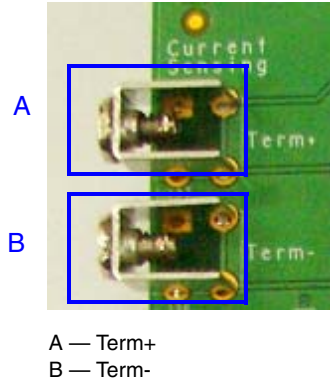
Figure 2-6. Current sensing terminals

2.4.2   Relays Status Indicator LEDs

LEDs in this reference design indicate if the output of the HVAC system is on or off. Unused LED's could be disconnected to prevent battery drainage. These can be turned on with a switch (see Figure 2-5–B). The location of the LEDs are shown in Figure 2-7.

A — Heat
B — Fan
C — Cool
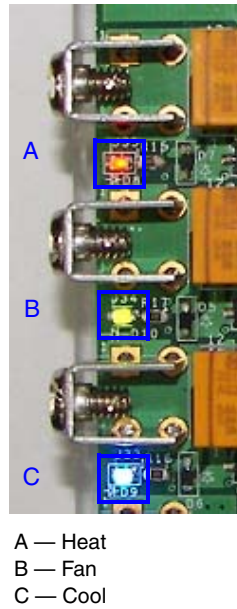
Figure 2-7. LEDs location

2.4.3 Dual Power Supply

The design includes a voltage regulator. The system can be used without batteries and with an external power supply. The voltage can go from 6 to 12 volts DC. The location of the external power supply jack can be found at the bottom left corner of the reference design.
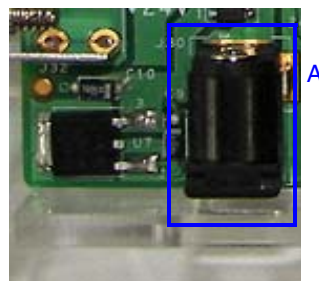
A — External power supply jack

Figure 2-8. External power supply jack

Chapter 3
Firmware Description

3.1   System Overview

The code for the thermostat is written in a way, that the software modules interacting with the MCU peripherals are independent of the modules that process the information. The modules used as an interface with the MCU peripherals are known as hardware abstraction layer (HAL) while the modules interacting with HAL and passing the information to main.c are known as a hardware independent layer (HIL).

The use of this architecture reduces dependency between the blocks and improves portability of the software drivers to a different application. Figure 3-1 gives a high-level overview of how the thermostat software works.

```
                    ┌─────────────┐
                    │    Main     │
                    └──────┬──────┘
                           │
                           ▼
            ┌──────────────────────────────┐
            │    Initialization routines:   │
            │     Clock and System          │
            │     Options TOD and           │
            │   Calendar Temperature        │
            │     Control and ADC           │
            │   Keyboard Temperature        │
            │    Control (HVAC) LCD         │
            └──────────────┬───────────────┘
                           │
                           ▼
            ┌──────────────────────────────┐
            │     StartTempMeasurment       │
            └──────────────┬───────────────┘
                           │
                           ▼
                     ◇ TempMeasurment
                       Finished?
                           │
                           ▼
            ┌──────────────────────────────┐
            │        Display Temp           │
            └──────────────┬───────────────┘
                           │
                           ▼
                     ◇ Calendar Match  ──────►  ┌─────────────────────┐
                       Event                     │ Start TempMeasurement│
                           │                     └─────────────────────┘
                           ▼
                     ◇ KeyPressed     ──────►  ┌─────────────────────┐
                                                │ CheckKey and Perform │
                           │                    │       Action         │
                           ▼                    └─────────────────────┘
            ┌──────────────────────────────┐
            │      Go to Low Power          │
            └───────────────────────────────┘
```
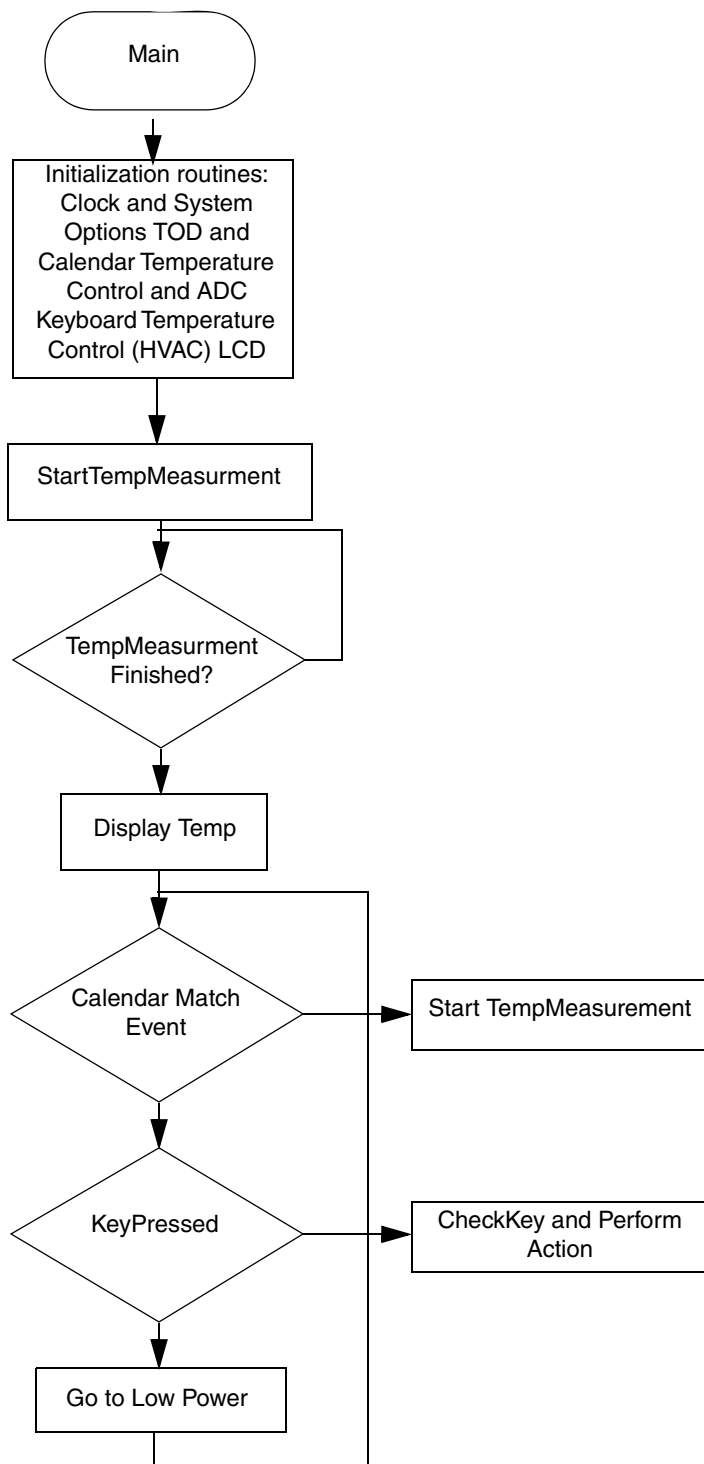
To exit low power it is necessary have either a TOD
match condition or a keyboard interrupt

Figure 3-1. High level thermostat workflow flow diagram

Figure 3-1 shows that the thermostat initializes by configuring all the peripherals and variables. It then measures and displays the initial temperature, after the thermostat enters the infinite loop that sends the microcontroller into stop mode where it spends most of the time. There are two scenarios where it can wake from stop:

- When the TOD module match condition occurs — This can be every 5, 10, 15, 30 or 60 seconds. When wake occurs, the time and date is updated and a temperature conversion and VDD level measurement starts. This finishes when the date and time have been updated in the LCD as well as the new temperature and voltage level. In case the temperature changes, the thermostat makes the decision to turn on or off the cooler, or the heater. After configuring HVAC outputs, it goes back to stop mode again.

- When a key is pressed — In this case the thermostat checks if the key is intended to do something based on the current status. If this is the case, then it might perform different actions. Depending on the key pressed it might go back to sleep immediately or perform prior actions.

For the low-end thermostat, the similar logic is applicable. The high-end version has to wakeup from stop mode every second to blink the time colon, and a different wakeup period when other sections of the LCD have to blink. Section 3.2.1, "Blink Mode" explains how the automatic blink mode works for the low-end thermostat.

3.2   LCD Driver
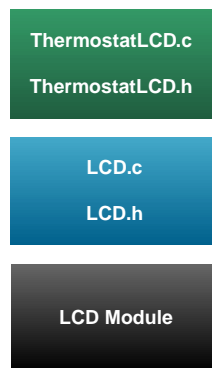
Figure 3-2. LCD SW modules

The LCD driver for both, the low-end and the high-end versions are found in application note *AN3796 — LCD Driver Specification*. This application note details how to migrate the software driver provided to a specific custom LCD.

The HAL driver for the LCD module provides the module initialization functions as well as routines to display information. For the low-end LCD, the HAL driver gives the option to write messages configuring the blink mode.

The HIL routines display relevant information for the thermostat application. They can write the date and time as well as the temperature among other features. These files take information from the HAL block to know the amount of characters that can be displayed for temperature and time as well as the position for used symbols: battery level, heating, and cooling symbols. By doing this, it is possible to have the same HIL file for both the LCDs, even when the hardware and the HAL implementations are different.

3.2.1 Blink Mode

The MC9S08LL16 LCD module includes a blink mode intended to alternate messages at a defined frequency. Write one message in the lower 4-bits of the corresponding waveform and the alternate message in the higher 4-bits of the same register. Depending upon the blink frequency selected, the LCD module automatically switches between the two messages. This feature works only for displays with four or less backplanes. This is why it is implemented only for the low-end thermostat.

To use the blink feature, the LCD driver always has this mode enabled with a one second blink period and writes the same contents for both messages, therefore displaying the same information even when the blink mode is active. When there is a need to configure a portion of the LCD to blink (for example, the time colon or any of the LCD data when configuring date, time, or temperature set points) the higher nibble is cleared. The LCD displays information during half of the period and the other half clears all the segments making a section of the LCD blink.

The main benefit of this feature is that the MCU does not have to wake to change the contents of the LCD. Because this is done automatically, the thermostat wakes only to make a new temperature conversion or when a button is pressed, otherwise it stays in low power mode increasing battery life by having a longer period working with a short current.

3.3 Temperature Measurement
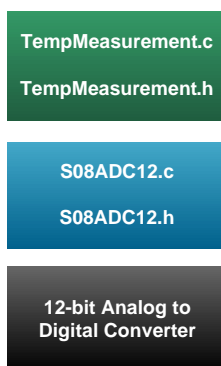
Figure 3-3. Temperature measurement modules

To measure temperature, the thermostat uses a thermistor and the ADC. The S08ADC12 is the HAL module designed to configure the ADC. It provides functions to start an ADC conversion for any channel, stops the ADC, declares the interrupt vector to read the last ADC conversion, and stores it in a buffer. The header file has definitions to declare the amount of channels used for analog and digital conversions.

The temperature measurement HIL block has two main functions:

- Read the ADC channel connected to the temperature sensor and convert this to a temperature value.
- Get the supply voltage after making an ADC conversion for the bandgap channel.

To measure temperature, the thermostat can use either celsius or fahrenheit. The process is to get the average of eight ADC conversions to reduce noise and then convert the ADC value to a temperature value by using a lookup table. Section 2.2.1, "Temperature Sensor" indicates the hardware connections. Appendix B, "Temperature Sensor Coeffecients" details the values used in the lookup table to get the environment temperature.

3.4 Thermostat Configuration and Menus
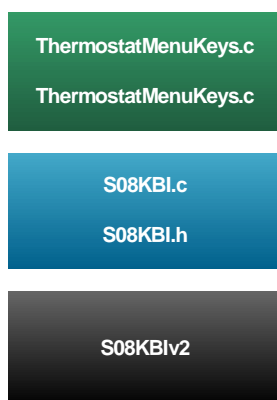
Figure 3-4. Thermostat configuration

The keys are intended to change and configure different information and parameters that can be shown or used in the thermostat. These parameters are:

- Cooling set point
- Heating set point
- Metric system
- Time
- Day (only in low-end thermostat)
- Date (only in high-end thermostat)
- Fan on, off, and auto

As shown in Figure 3-4, the hardware module used is the keyboard module S08KBI. This block detects when a key is pressed and determines which key is pressed. The ThermostatMenuKeys block indicates the exact key pressed (program, time, date, up, down, and so on). At this point, the main application determines how to respond with the key pressed. This depends on the thermostat status at that moment.

3.5 Time of Day and Calendar Functionality

Figure 3-5. Time of day module

The TOD module is used to create the calendar functionality. It also provides a match condition that can wake the MCU up to every 64 seconds. With this module, the thermostat performs the time and the date functionality. It displays the day-of-the-week and time for the low-end thermostat and the time in a 24-hour format in the high-end thermostat.
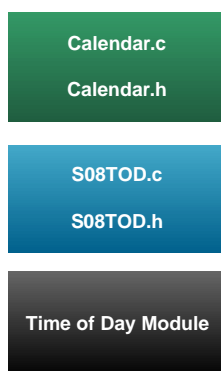
The calendar match condition also synchronizes the thermostat to start a new temperature conversion. By default it is configured to wake the MCU every five seconds, for the thermostat to show a rapid temperature change. However, in a home or building environment the temperature does not change quickly, therefore there is an option to configure this wakeup time to every 5, 10, 15, 30, or 60 seconds. By increasing the wakeup time, the power consumption is reduced and the batteries lifetime is increased.

Chapter 4, "User's Guide" details how to configure the wake-up time.

3.6 Other Firmware Functions

3.6.1 HVAC Control

This reference design controls standard HVAC outputs for heating, cooling, and air-condition. Section 2.2, "Heating, Venting, and Air Conditioning-HVAC" explains how latching relays work and their benefits in terms of current consumption. From the firmware perspective, the code has a set and a reset pin for each of the HVAC outputs.

To drive on an output, the MCU has to drive its set output while keeping the reset output disabled. The opposite configuration needs to be used to clear the output. As explained in Section 2.2, "Heating, Venting, and Air Conditioning-HVAC", the latching relays need only a few milliseconds to set or clear an output. The code drives the output for a short amount of time and then sends both outputs to their inactive state, reducing power-consumption, firmware, and hardware implementations.

3.6.2 Non-Volatile Backup

To preserve the configuration stored by the user, the thermostat uses the flash memory to store a non-volatile version of the heating and the cooling set points for each day-of-the-week. This is possible because the MC9S08 family of devices can write and erase the flash memory without the need of any external voltage.

Every time the parameters are changed, the MCU erases the previous stored parameters and writes the new values in the flash memory. By doing this, the parameters are preserved even if the thermostat is turned off.

NOTE

When storing the values in non-volatile memory, the MCU uses a higher IDD than a normal working mode. If there is an ammeter connected to measure the MCU current consumption during this operation, it is possible to cause a reset and the system looses the cooling and the heating set-points.

Thermostat Reference Design Using the MPC9S08LL16, Rev. 1

Chapter 4
User's Guide

To start the thermostat, use either two AA batteries or connect to an external power supply. The external power supply can go from 6 to 12 volts DC. The LCD shows the current temperature, day-of-the-week, hours and minutes, heating, cooling, and hold mode. The LCD distribution for both high-end and low-end thermostat versions are shown in Figure 4-1.
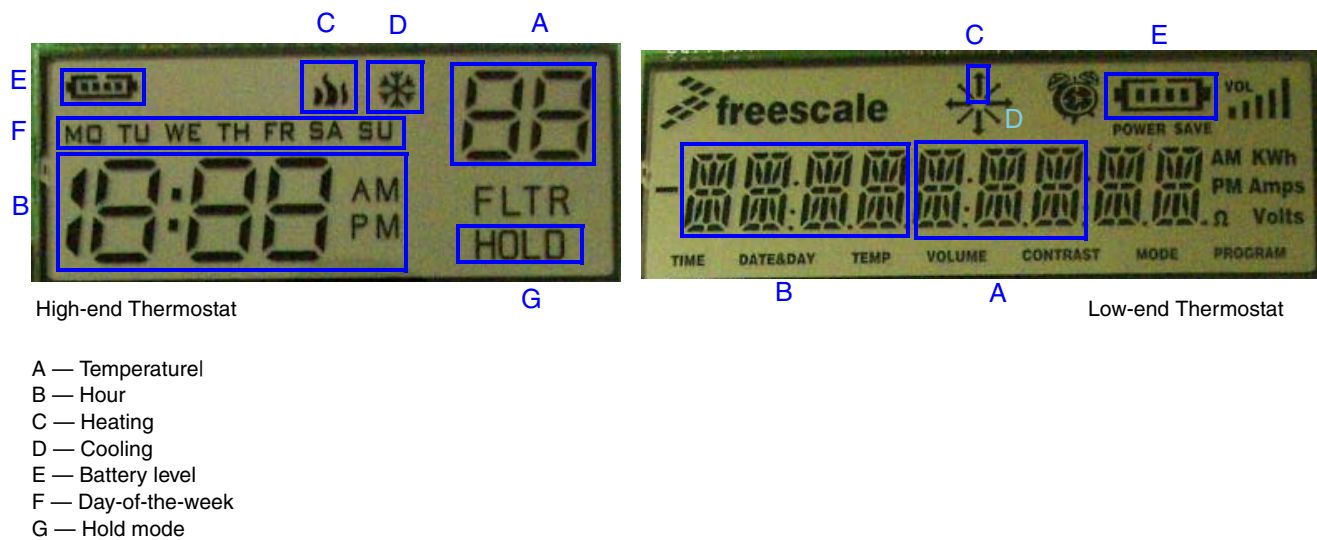

High-end Thermostat   Low-end Thermostat

A — Temperature
B — Hour
C — Heating
D — Cooling
E — Battery level
F — Day-of-the-week
G — Hold mode

Figure 4-1. Thermostat LCD details

4.1 Configuring the Thermostat

Thermostat configuration steps are similar for both versions. There are small differences because the LCD panels are different. For example, a 24 hour configuration for only the high-end thermostat or day-of-the-week for only the low-end thermostat.

4.1.1 Configuring Time and Day

Start by pressing the program button to configure the hour, minutes, and day-of-the-week. The hours start blinking in the LCD indicating the hours parameter will be configured. Use the up and down button to configure the hours. By pressing the program button again the thermostat starts blinking, the minutes blink, press again and it jumps to the day-of-the-week.
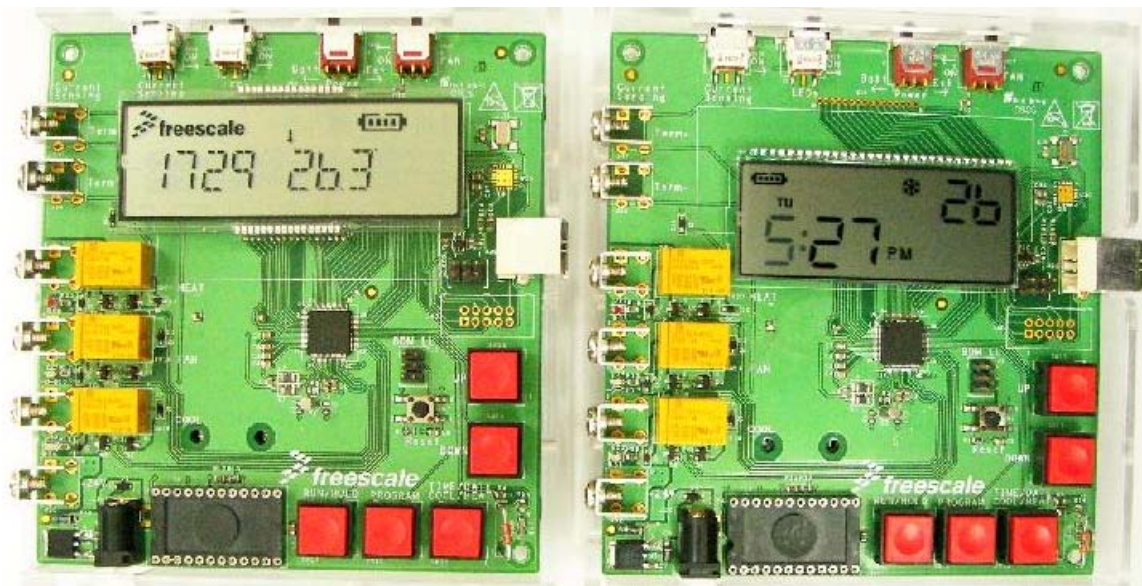
Figure 4-2. Configuring time and day

If the hours do not need to be re-configured press the time/date and cool/heat button. This directs you to the next step which is temperature configuration.

4.1.2   Configure the Temperature Set Points and Metric System

Access the temperature configuration menu after fully programming the hour or by pressing the time/date button at any time. By pressing the time/date button again while programming, the temperature set points and metric system exits the programming mode, and goes back to running mode showing the current thermostat operations.

The first configuration parameter for the temperature is the metric system. The thermostat can display all the parameters either in celsius (°C) or fahrenheit (°F). As shown in Figure 4-3, when changing the metric system, the thermostat selects the desired metric system in the minutes section of the LCD. The temperature display changes until the metric system is selected.
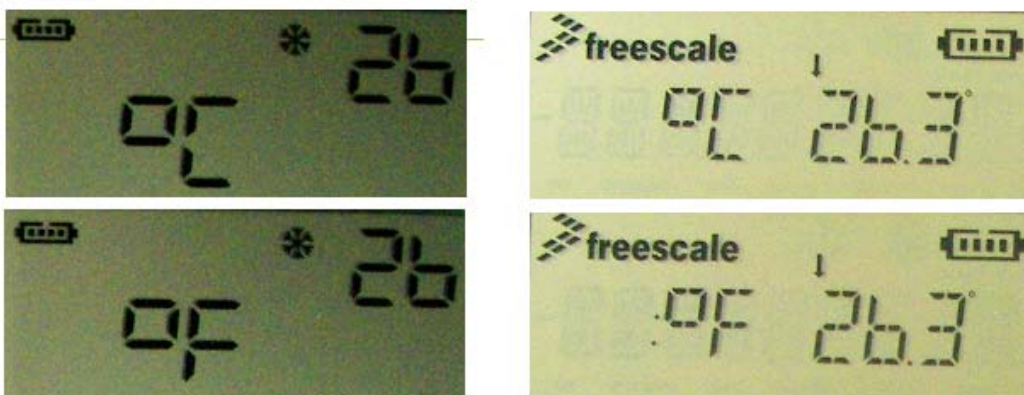

Figure 4-3. Configuring the metric system

Thermostat Reference Design Using the MPC9S08LL16, Rev. 1

By pressing the program button the thermostat jumps to the cooling set point configuration. The cooling indicator in the LCD starts blinking to show this is the parameter being configured at the time. When the thermostat is in a temperature above the cooling set point the cooling and fan switches turn on to drive an external fan and the air-conditioner. By pressing the programming button one more time the thermostat goes to select the heating set point. The heating logo starts blinking and the temperature display shows the heat temperature as shown in Figure 4-4. When the thermostat is at any temperature below the heating set point the heater and fan switches are activated.

Figure 4-4. Heating and cooling set points

4.1.3 Configure the Wakeup Time

The last parameter that can be configured is the wakeup time. This is normally a parameter that can not be configured in commercial thermostats. However, TOD module in the MC9S08LL16 allows automatic wakeup after several seconds. In the case of the thermostat, the available options are every 5, 10, 15, 30, and 60 seconds. After this period of time, the thermostat takes the temperature and VDD readings and updates the time, current temperature, and drives the HVAC outputs as needed. See Figure 4-5.
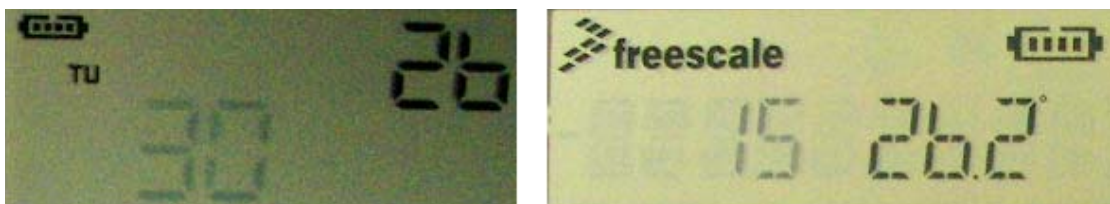
Figure 4-5. Configuring the wakeup time

The reason to add this feature is because in a home or building environment the temperature does not quickly change, waking every 15, 30, or 60 seconds could be enough to keep a stable environment. In the case of a demonstration, it is easier to show the temperature changes by waking up every 5 or 10 seconds to demonstrate the HVAC outputs changes and the temperature change.

4.2 Using the Thermostat

After configuring the thermostat it displays the day-of-the-week, hour, temperature, and the battery level.

Figure 4-6. Configured thermostat

4.2.1 Showing the Running Program

When the thermostat is running, pressing the time/date key shows the running program. The cooling logo indicates the coolin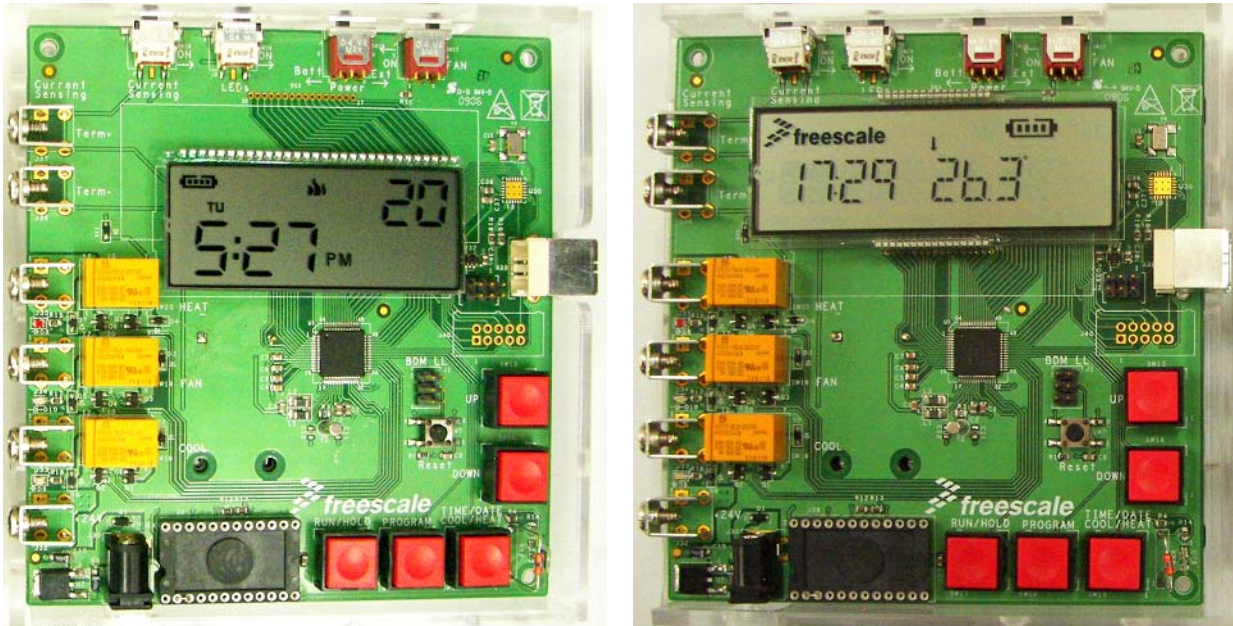g temperature setting and the heating logo indicates the temperature to turn on the heater. Each value is shown during a 5 second period and after showing both, the thermostat returns to normal operation mode.

4.2.2 Run/Hold Mode

By default the thermostat works in a normal running mode. After power on, the thermostat turns the heater on when reading temperatures below 20 °C (68 ° F) and starts the cooler when the temperature is above 23 °C (73 °F). While in the run mode, the thermostat automatically tries to keep the temperature within a range between the heating and cooling set points.

The thermostat switches to hold mode, if the run/hold button is pressed while in run mode. This means that it starts working and ensures the same temperature it had when the run/hold button was pressed. The heater and cooler are activated if the temperature goes either above or below the hold temperature at any moment. The LCD uses the hold indicator to show the user that it is currently bypassing the temperature program as shown in Figure 4-7.

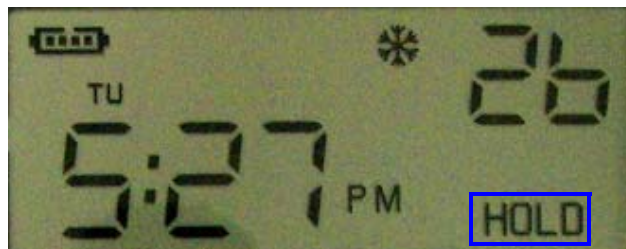**Figure 4-7. Thermostat in hold mode**

Thermostat Reference Design Using the MPC9S08LL16, Rev. 1

Pressing the up button increase the run/hold temperature by one degree. The opposite change occurs when pressing the down button. Pressing the hold button once more makes the thermostat enter normal run mode and the program stored for heating and cooling is used to control the temperature.

4.2.3   Fan Control

The fan switch intends to drive the fan output either as automatic or always on. When using auto mode, the thermostat only turns on the fan when heating or cooling. If the user selects the on position for the fan, the thermostat keeps the fan output active the whole time. This keeps the air flowing in a home or building, even when the heater or cooler are off.
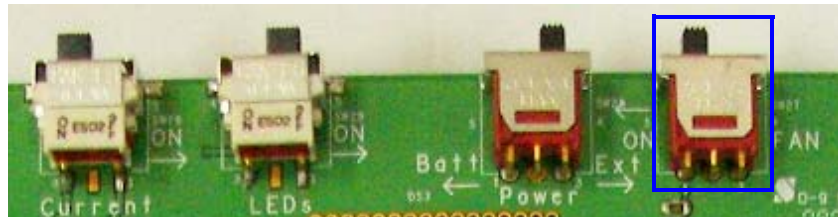

Figure 4-8. Fan On/Auto switch

NOTE

The fan switch is a three position switch. The user must select either on or off. Leaving the fan switch in the middle position results in a floating input that causes extra current consumption.

Thermostat Reference Design Using the MPC9S08LL16, Rev. 1

4.2.4   Changing Contrast

Only in the high-end thermostats after power on, the thermostat is at its lowest contrast level. By pressing the up and down keys while being idle the thermostat increases or decreases the contrast level accordingly. The best way to see the contrast change is to watch the display from a 90° angle. There is no visual indication or feature to indicate the current contrast level. It is also important to highlight that the increase in contrast also increases the current consumption.

4.2.5   Measuring Current

One of the major features of the MC9S08LL16 family is its low power consumption while driving an LCD and keeping alive the keyboard inputs and calendar functionality. To attain this thermostat reference design it comprises of a current sensing switch and two screw terminals to measure current.
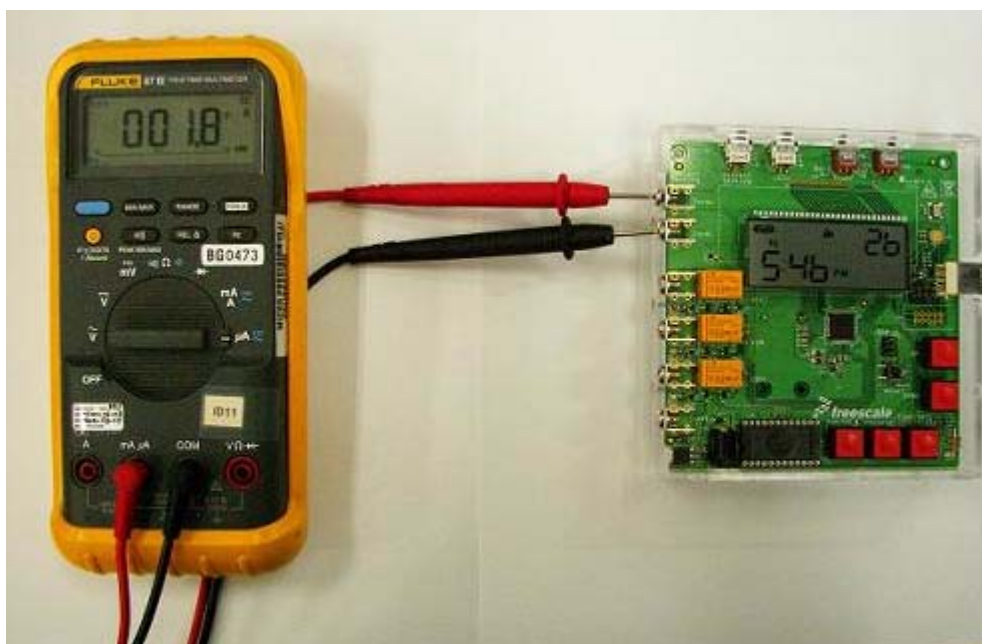

Figure 4-9. Current measurements

To measure the current, connect an ammeter to the current sensing screw terminals and then move the current sensing switch to the on position (Figure 4-9). The current shown is used to drive the LCD, monitor the inputs, keep the time and date functionality, and drive the HVAC outputs. Depending on the wakeup time configured, you will notice periodical increments in the current. These increments occur when the MCU wakes to make all the voltage and temperature readings and drive the HVAC outputs.

As explained earlier, the blink feature used for the low-end thermostat allows having an alternate message to avoid waking the MCU to blink a portion of the LCD. This is one of the major advantages of the MC9S08LL16 family of devices because the MCU does not need to go to run mode to blink the seconds colon or to blink a portion of the MCU when configuring the thermostat. This is a key feature to save energy and extend battery life. Table 4-1 shows the power consumption measurements for both low-end and high-end thermostat versions.

Table 4-1. MCU and system current consumption

State	Low End MCU Current Consumption	Low End Wakeup Period	Low End System Current Consumption	High End Current Consumption	High End Wakeup Period	High End System Current Consumption
Idle – Low Power	1.5 µA	—	15 µA	6 µA	N/A	20 µA
Seconds Wake-up	1.5 µA	—	15 µA	1.6 µA	17 µs	1.6 µA
Full-Wake up	1.5 µA	5 ms*	1.5 µA	1.6 µA	5 ms*	1.6 µA

*This value can from every 5 seconds up to every 60 seconds

Based on these current consumption values the estimated lifetime for different types of batteries are listed in Table 4-2:

Table 4-2. Batteries life time for different configurations

5 Second Wakeup

Capacity(mAH)	Battery Type	Low-End MCU Lifetime (years)	Low-End System Lifetime (years)	High-End MCU Lifetime (years)	High-End System Lifetime (years)
2000	AA	16.47284917	8.346536051	12.07831411	7.04740041
1250	AAA	10.29553073	5.216585032	7.548946316	4.404625256
830	CGR17500	6.836232404	3.463812461	5.012500354	2.92467117

10 Seconds Wakeup

Capacity(mAH)		Low-End MCU Lifetime (years)	Low-End System Lifetime (years)	High-End MCU Lifetime (years)	High-End System Lifetime (years)
2000	AA	18.00010669	8.718959028	12.89386639	7.315655702
1250	AAA	11.25006668	5.449349392	8.058666493	4.572284814
830	CGR17500	7.470044275	3.618367997	5.350954552	3.035997116

60 Seconds Wakeup

Capacity(mAH)		Low-End MCU Lifetime (years)	Low-End System Lifetime (years)	High-End MCU Lifetime (years)	High-End System Lifetime (years)
2000	AA	19.50727597	9.055682376	13.66265222	7.555316433
1250	AAA	12.19204748	5.659801485	8.53915764	4.722072771
830	CGR17500	8.095519527	3.758108186	5.670000673	3.13545632

Thermostat Reference Design Using the MPC9S08LL16, Rev. 1

Table 4-2 shows the batteries lifetime in two scenarios, the MCU and the whole system power consumption. It gives different examples based on the wakeup time configured in the thermostat.

The current consumption values shown in the low end and high end MCU columns measure the power consumption of the microcontroller and all the switches, buttons, temperature sensor, and the LCD. The current consumption values indicated in the low end and high end system columns are measured for the whole board, including transistors, relays, and so on. With this information, it is possible to get an estimate of the batteries lifetime for a similar application and also for an application that only includes LCDs and buttons (as in portable health care devices).
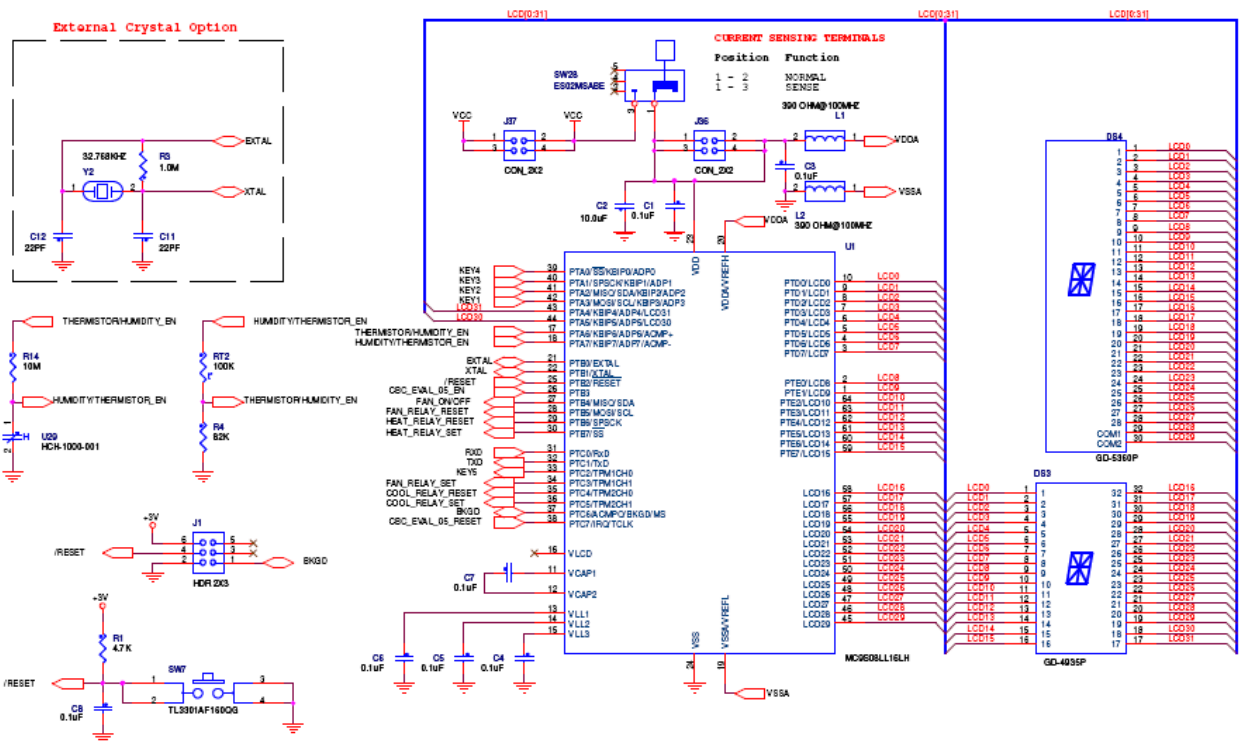
Appendix A
Schematics

Figure A-1. MC9S08LL16 connections
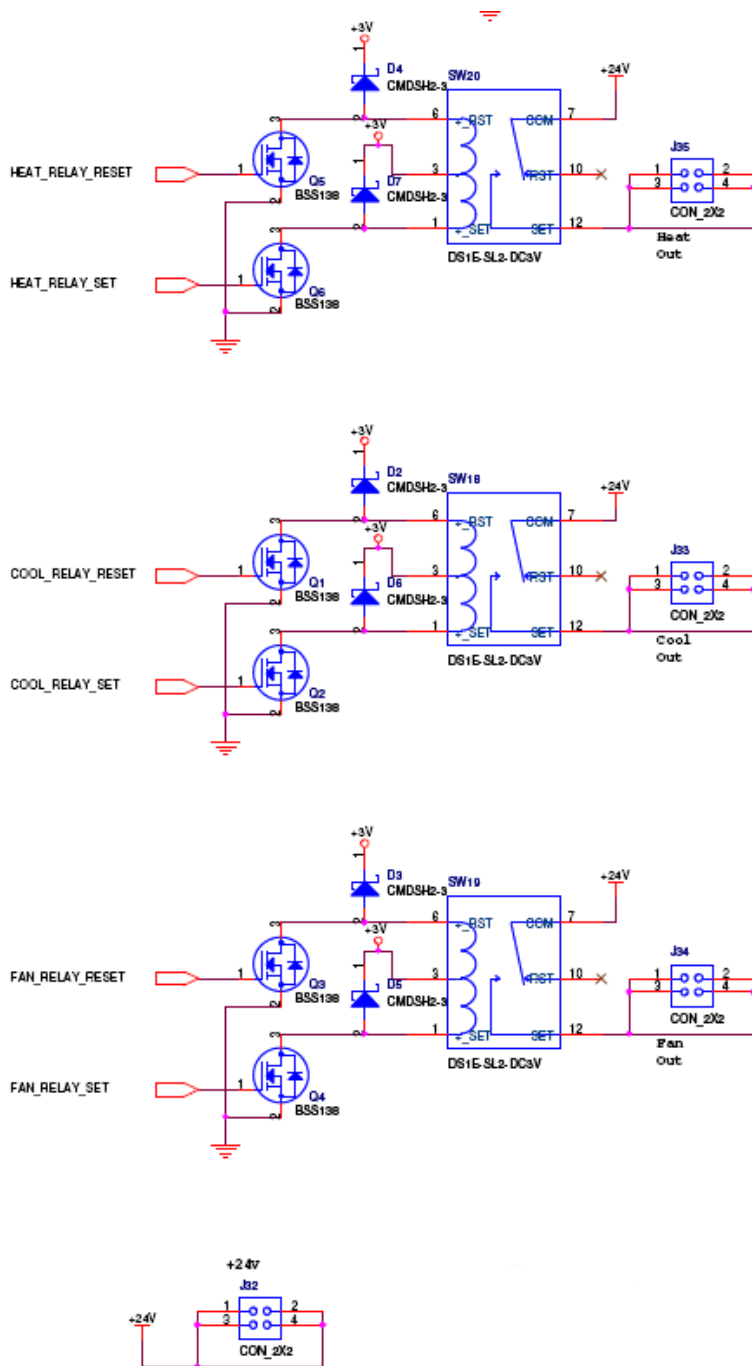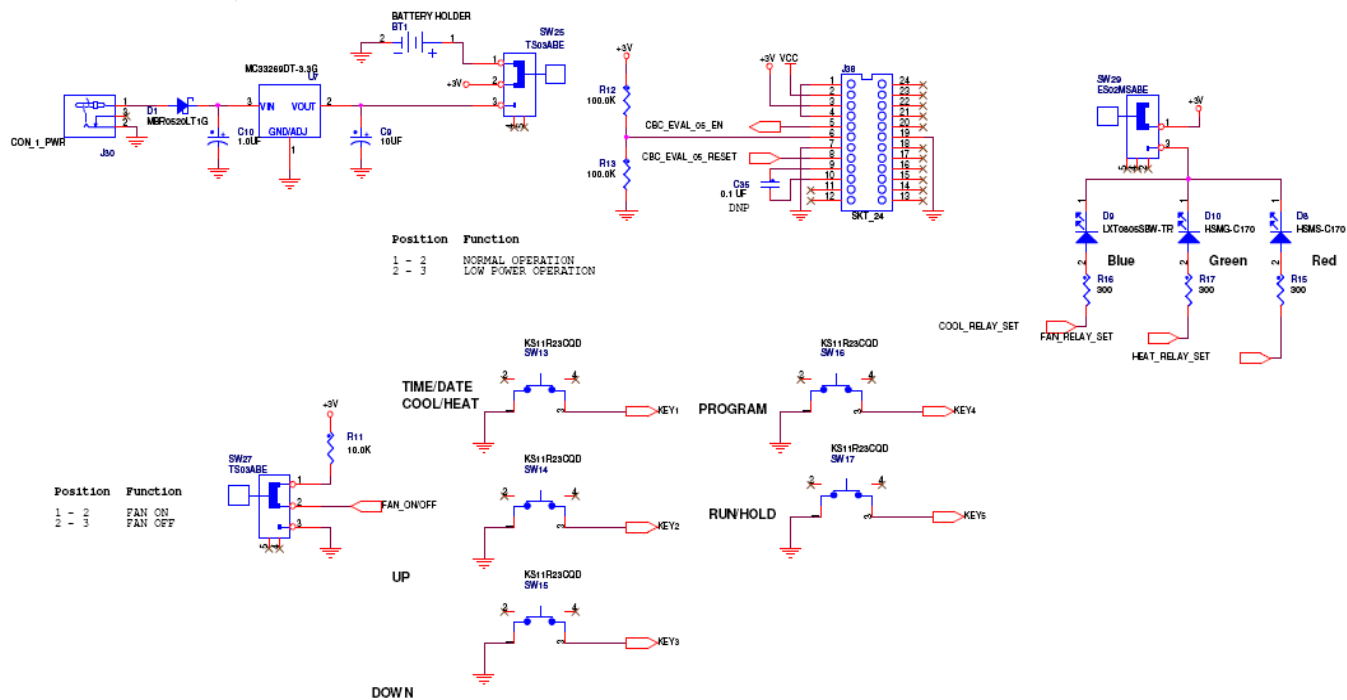
Figure A-2. HVAC connections

Figure A-3. Buttons, LED, and power supply
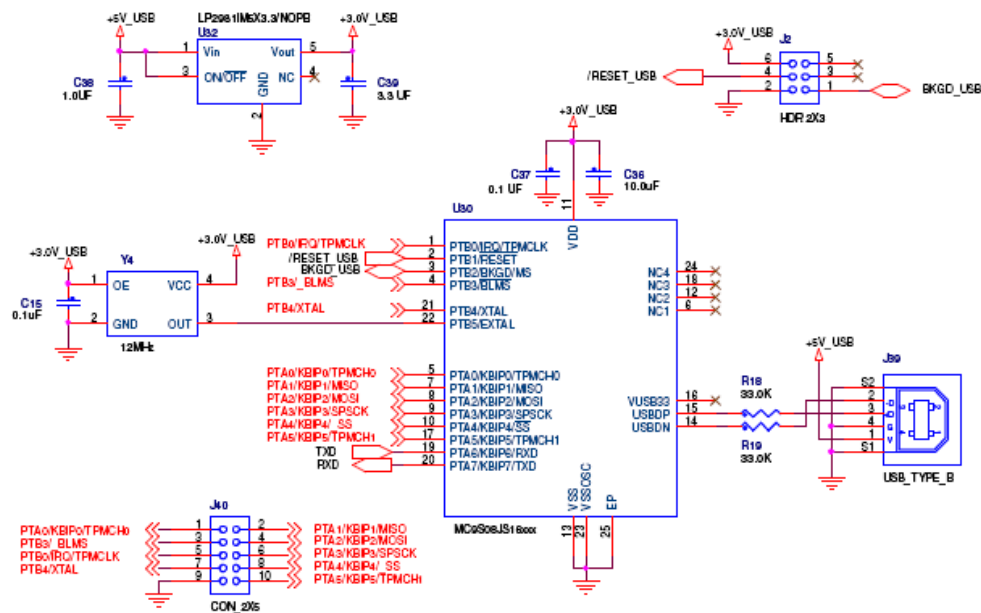
Figure A-4. Optional USB to SCI MC9S08JS16

Thermostat Reference Design Using the MPC9S08LL16, Rev. 1
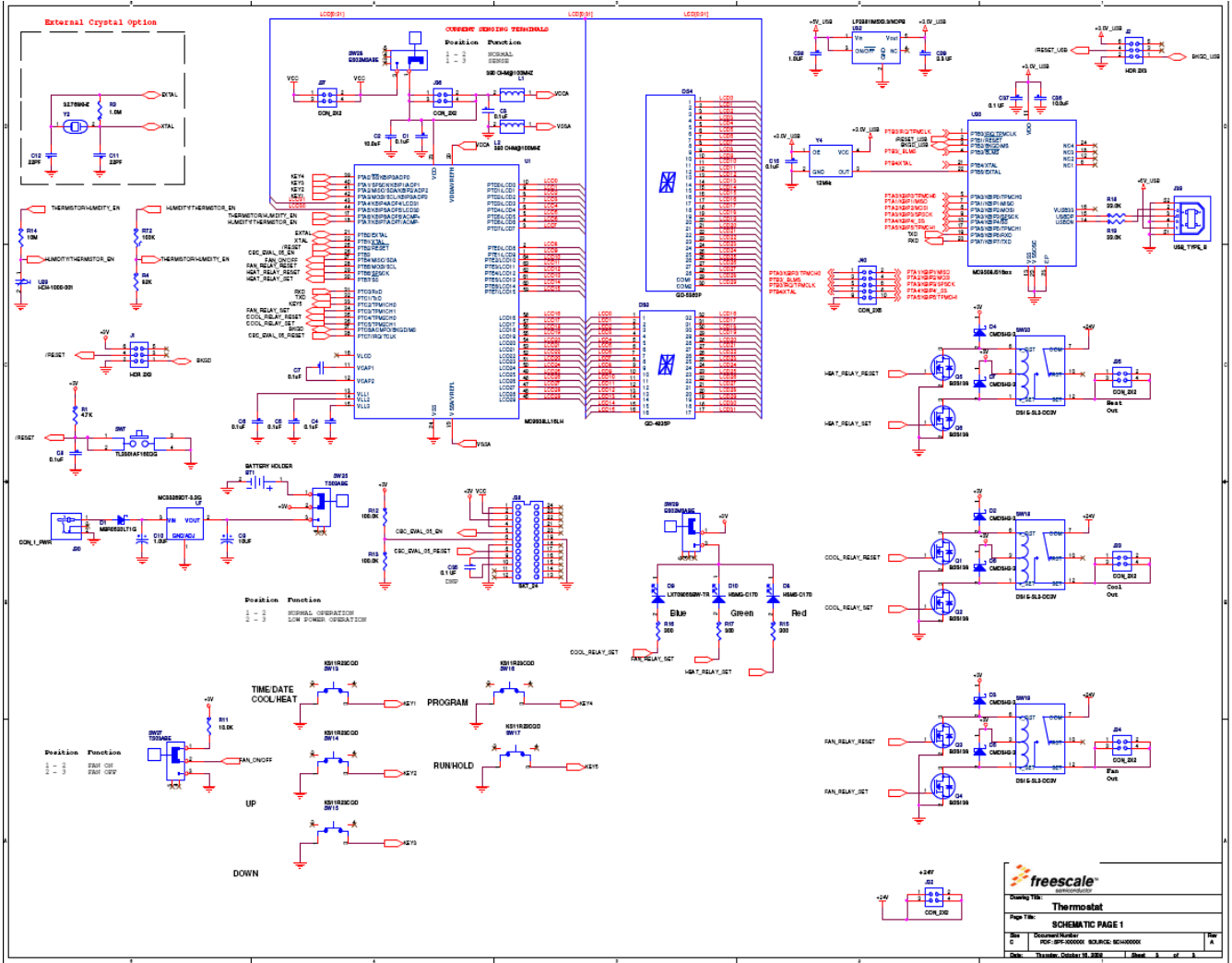
Figure A-5. Complete schematic

Appendix B
Temperature Sensor Coeffecients

Table 4-3. Temperature sensor coeffecients

°F	°C	RT Coeff	Resistance	Resistor-Divider Voltage (V)	ADC Values
50	10.0	1.9903	199030	0.963	1195
51	10.6	1.9387	193871.96	0.981	1217
52	11.1	1.8885	188847.6	0.999	1240
53	11.7	1.8395	183953.45	1.017	1263
54	12.2	1.7919	179186.14	1.036	1286
55	12.8	1.7454	174542.37	1.055	1309
56	13.3	1.7002	170018.96	1.074	1333
57	13.9	1.6561	165612.77	1.093	1356
58	14.4	1.6132	161320.77	1.112	1380
59	15.0	1.5714	157140	1.132	1404
60	15.6	1.5319	153185.57	1.151	1428
61	16.1	1.4933	149330.65	1.170	1452
62	16.7	1.4557	145572.75	1.189	1476
63	17.2	1.4191	141909.41	1.209	1500
64	17.8	1.3834	138338.26	1.228	1524
65	18.3	1.3486	134856.97	1.248	1549
66	18.9	1.3146	131463.3	1.268	1573
67	19.4	1.2816	128155.02	1.288	1598
68	20.0	1.2493	124930	1.308	1623
69	20.6	1.2188	121878.19	1.327	1647
70	21.1	1.1890	118900.93	1.347	1672
71	21.7	1.1600	115996.4	1.367	1696
72	22.2	1.1316	113162.82	1.387	1721
73	22.8	1.1040	110398.45	1.406	1746
74	23.3	1.0770	107701.62	1.426	1771
75	23.9	1.0507	105070.67	1.447	1795

Thermostat Reference Design Using the MPC9S08LL16, Rev. 1

Table 4-3. Temperature sensor coeffecients (continued)

76	24.4	1.0250	102503.98	1.467	1820
77	25.0	1	100000	1.487	1845
78	25.6	0.9763	97626.749	1.506	1870
79	26.1	0.9531	95309.822	1.526	1894
80	26.7	0.9305	93047.881	1.546	1919
81	27.2	0.9084	90839.621	1.566	1943
82	27.8	0.8868	88683.769	1.585	1968
83	28.3	0.8658	86579.081	1.605	1992
84	28.9	0.8452	84524.342	1.625	2017
85	29.4	0.8252	82518.368	1.645	2042
86	30.0	0.8056	80560	1.665	2066