

APPLICATION NOTE 3965

How to Use the MAXQ3212 Microcontroller as a Digital Thermostat

Abstract: This application note describes a digital thermostat based on the MAXQ3212 RISC microcontroller. This simple system senses the ambient temperature, compares it to a user-input set point, and switches an output relay as a result of this comparison. System components also include a 1-Wire digital thermometer and a 4-digit, 7-segment LED display driven by an 8-digit LED display driver. By choosing this microcontroller and the highly integrated display driver, a minimal component count system is achieved.

Introduction

There are situations where temperature regulation and control are desirable. When the ambient temperature is too cold, for example, a light bulb can be used as a heat source in a small enclosure to dry or cure paint or similar material. Temperature sensing, power switching, and a basic heat source thus provide a closed-loop temperature-control system for many situations.

This application note describes how to use a MAXQ microcontroller to implement this type of control system, also called a digital thermostat. The [MAXQ3212](#) is chosen as the system controller because of its functionality, ease of use, low cost, and the ready availability of an inexpensive development environment, the MAXQ3210 Evaluation Kit ([MAXQ3210-KIT](#)). The complete development environment for this example code is version 1.0 of the MAX-IDE and revision A of the MAXQ3210 kit board.

System Implementation Details

Overview

The system implementation for this digital thermostat is quite simple. **Figure 1** shows a schematic of the circuit. A [DS18B20](#) 1-Wire® Digital Thermometer measures ambient temperature. The system controller compares the measured ambient temperature to a user-input set point; and a relay is activated if the ambient temperature is below the set point. The system uses a 4-digit LED display to show the ambient temperature in normal operation and the set point when it is input by the user.

A MAXQ3212 serves as the system's controller. It performs the user-interface function (set point up and down) through two pushbutton switches connected to the processor through port pins configured as inputs. Another port pin is used as a single output dedicated to controlling the output relay.

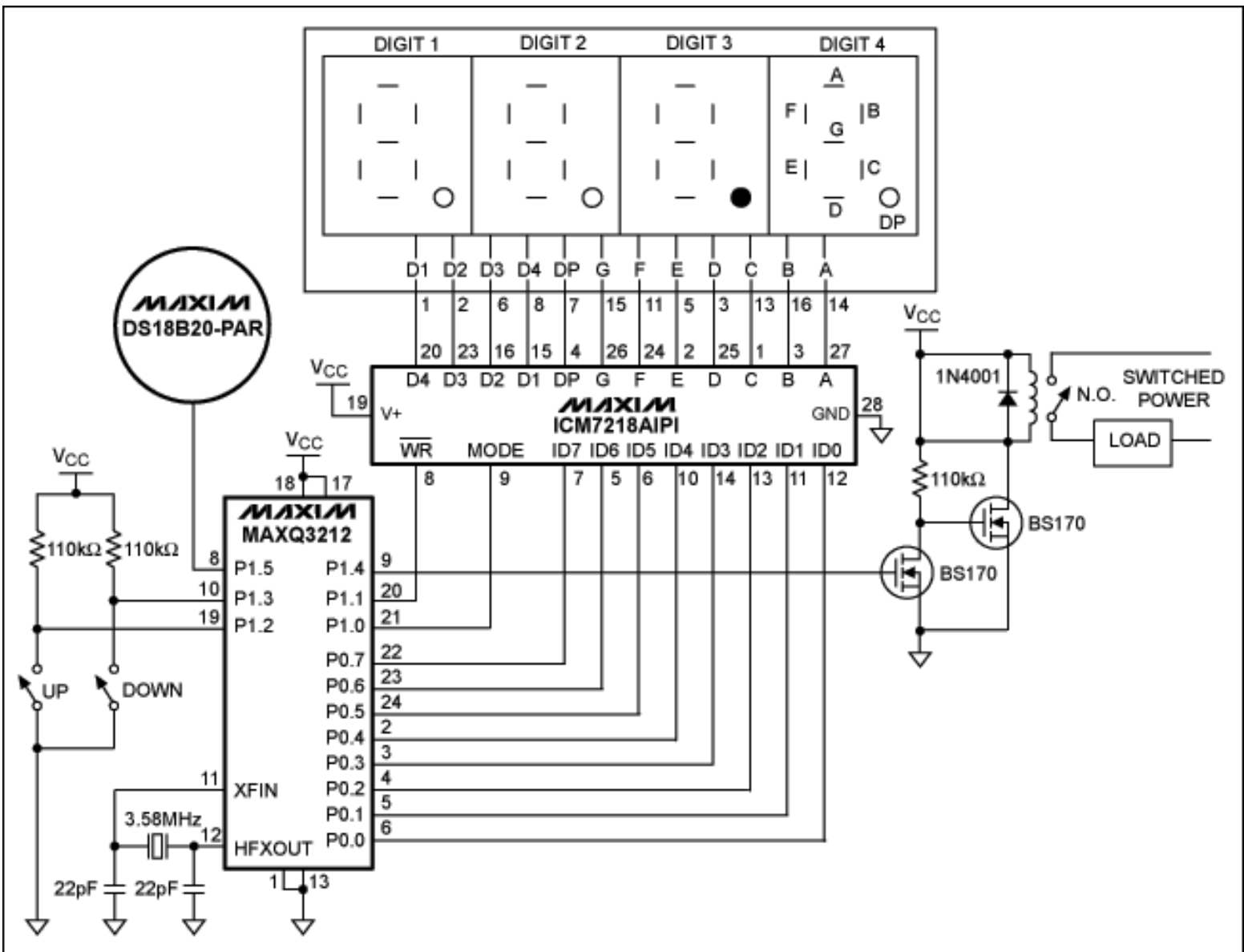


Figure 1. Thermostat circuit schematic.

For the system's display, a 4-digit, 7-segment LED was chosen so the thermostat would be usable in a variety of lighting conditions without additional electronics such as backlighting.

To simplify the processor's interface to the display and to provide the necessary drive current and multiplexing, a Maxim [ICM7218AIP1](#) 8-digit LED driver was selected. This device contains all the circuitry necessary to interface the microcontroller directly to the common-anode, LED display. The LED driver drives up to eight digits, although only four digits are used in this application. This device drives each of the 7 segments directly without external current-limiting resistors.

The system's output is switched AC power provided by a standard electromechanical relay which switches AC power to an arbitrary load and can be sized according to the load's power demand. This relay was chosen because it can switch 8 amps and its coil voltage is 5V, which is the system's supply voltage. The relay can, however, operate from any available power source that does not exceed the voltage or current capabilities of the final drive FET (BS170).

While it is certainly possible to do so, no attempt was made to model the precise behavior of the system and optimize the closed-loop performance. A simple comparison between the actual and desired temperatures is the only output-switching criterion here. The software's subroutine `Relay_Switch` can be modified, as desired, to achieve the desired system behavior.

A single DS18B20-PAR measures the ambient temperature. This device has an accuracy of $\pm 0.5^{\circ}\text{C}$ over a range of -10°C to $+85^{\circ}\text{C}$. In this application, the temperature is expected to range between $+12^{\circ}\text{C}$ to $+38^{\circ}\text{C}$ ($+55^{\circ}\text{F}$ to $+100^{\circ}\text{F}$). The resolution of this device is user-selectable from 9 to 12 bits; the resolution selected for this application is explained below. While it is not used in this application, each DS18B20 contains a unique, 64-bit identification number that supports multiple devices on the 1-Wire network.

1-Wire Interface

Maxim's 1-Wire bus is a unique serial interface that requires a single signal line and ground to perform communications. This interface enables a microcontroller to communicate with a 1-Wire device using a single port pin. This interface also provides power, called parasitic power, over the 1-Wire signal line, which allows the 1-Wire device to operate without a local supply. While not important for this application, this independent operation is required in some remote temperature-measuring situations.

Although this application requires only a single 1-Wire device, the 1-Wire bus is designed to operate with a single master and multiple slaves, i.e., it is multidrop. While this application uses a single DS18B20 device for temperature measurement, there are several devices from Maxim that perform this function. For instance, in applications sensing temperatures in multiple locations, the [DS28EA00](#) is designed specifically for multidrop applications, and it simplifies the detection of the physical sequence of 1-Wire devices in a 1-Wire network environment.

In this thermostat application, a single port pin is used to "bit-bang" and communicate over the 1-Wire bus. The software for this function comes directly from application note 3769, "[Building a 1-Wire Temperature Logger Using the MAXQ3210](#)." This code was written specifically for the MAXQ3210 microcontroller and the [DS1822](#) Econo 1-Wire Digital Thermometer. Only a minor modification was required to change the resolution from nine bits in the original software to 10 bits in this application. Application note 3769 also provides references for the 1-Wire bus and its applications.

MAXQ3212 I/O Pins

The MAXQ3212 has 15 port pins: 8 for port zero and 7 for port one. To interface with the ICM7218 display driver, a minimum of 10 signals (port pins) is required. Four other signals are required to interface with the temperature up switch, the temperature down switch, the relay driver, and the DS18B20. This makes the total requirement 14 port pins. In normal reset default operation, four of the MAXQ3212's 15 port pins are used for the JTAG interface (P1.0, P1.2, P1.3, and P1.4) and one for the reset input (P1.1) function. Although these five signals control debug functions, they are needed for this application. Consequently, some consideration was given to the overall debugging strategy.

The default debug functions of these five port pins can be disabled by setting the control bits TAP and RSTD to their appropriate levels (0 and 1, respectively). When disabled, the port pins allocated to these debug functions behave as ordinary port pins, and the debug operations are no longer available. Therefore, the final application software routines debugged were the display operations. Display operations were chosen because they require the largest number of I/O pins, and the functions are relatively uncomplicated and need little debug.

The assembly language code has a large delay (approximately 5 seconds) inserted before the instructions to disable these five debug control signals. This delay allows the debugger to gain control of the processor before the debug controls are disabled. In this way, the debugger can regain control of the part if necessary.

Display Selection

The primary requirement for the display options was that the thermostat be used indoors or outdoors, in bright light or darkness. It was decided that an LED display would provide the best performance under these widely varying ambient lighting conditions, except perhaps in very bright light such as a sunny day outdoors. When this device is used in such bright environments, a simple colored lens provides the necessary display contrast and visibility. Finally, the LED display avoids the need for additional circuitry such as a backlight source.

Four digits of display are required for this application and there are 7 segments per digit, without multiplexing. Therefore, 28 individual signals are required (4 digits x 7 segments/digit), not counting a decimal point. Clearly, the MAXQ3212 has an insufficient number of port pins to manage this interface without additional circuitry. The solution to the problem is the Maxim ICM7218, which provides multiplexing circuitry for up to 8 digits, and is capable of driving the LED segments directly without external current-limiting resistors. A simple 8-bit interface with two control signals (mode and write) interfaces this device to the processor.

User Input of the Set Point

Given the number of available port pins on the MAXQ3212, it is not practical to provide a parallel interface for the user to input the desired set point. Therefore, a simple up or down pushbutton interface allows the user to increment or decrement the value. When the system initially powers up, the firmware configures the default set point to +72.5°F, and stores this value in the MAXQ3212's EEPROM data memory. If the user increments or decrements the set point, the new value is stored in EEPROM. Since the EEPROM is nonvolatile, it is not erased by a power interruption.

When an up or down pushbutton closure is detected, the LED displays the current set point. To help identify that it is the set point being displayed, the microcontroller activates the decimal point of the last digit, which is normally off. If the pushbutton

remains pressed for a sufficient period, the set point automatically begins to increment or decrement by 0.5°F approximately every second. When the desired set point is reached, the user releases the pushbutton and the new set point takes effect. The display then returns to displaying the measured ambient temperature, and the last digit's decimal point is turned off.

Temperature Conversion

One of the more interesting aspects of the software for this application is the temperature conversion from degrees centigrade to degrees Fahrenheit. Upon completion of the DS18B20's "Convert Temperature" command, the 16-bit sign-extended results are stored in the device's temperature register in the format shown below.

Temperature Most Significant Byte

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Sign	Sign	Sign	Sign	Sign	2 ⁶	2 ⁵	2 ⁴

Temperature Least Significant Byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
2 ³	2 ²	2 ¹	2 ⁰	2 ⁻¹	2 ⁻²	2 ⁻³	2 ⁻⁴

The temperature sensor's resolution is user-configurable to 9, 10, 11, or 12 bits, which correspond to increments of 0.5°C, 0.25°C, 0.125°C, and 0.0625°C, respectively. The DS18B20's default resolution at power-up is 12 bits, but it is initialized to 10 bits for this application. When configured for 10-bit resolution, bits 1 and 0 (2⁻³, 2⁻⁴) of the results are undefined, and the minimum temperature resolution is 0.25 degrees centigrade.

In this application, the temperature is displayed in degrees Fahrenheit. The following equation performs this conversion.

$$\text{Degrees Fahrenheit} = (\text{Degrees Centigrade} \times 1.8) + 32 \quad (\text{Eq. 1})$$

Because the temperature is created as a fractional number and the conversion requires a multiplication by 1.8, calculations with binary fractions are required. In performing this multiplication of real numbers, minimizing any calculation errors without taxing the arithmetic resources of the processor is obviously desirable. Since the DS18B20 is initialized to produce the temperature value as a 10-bit number, bits 1 and 0 have no meaning. The remaining two bits (bits 3 and 2) of fractional temperature information provide a 0.25°C (0.45°F) resolution. As the MAXQ3212's accumulators are 8-bits wide, it is convenient to use a byte as the fractional portion of the calculated temperature data. Therefore, the multiplier can contain six bits of fractional information. The information below shows that six bits of fractional information will approximate the decimal value of 0.8 to 99.61 percent:

$$\begin{aligned} 2^{-1} &= 0.500000 = \Rightarrow 62.50\% \text{ of } 0.8 \\ 2^{-2} &= 0.250000 + 0.500000 = 0.750000 = \Rightarrow 93.7\% \text{ of } 0.8 \\ 2^{-5} &= 0.031250 + 0.0750000 = 0.781250 = \Rightarrow 97.65\% \text{ of } 0.8 \\ 2^{-6} &= 0.015625 + 0.781250 = 0.796875 = \Rightarrow 99.61\% \text{ of } 0.8 \end{aligned}$$

Converting to Fahrenheit with this degree of precision is more than sufficient for the resolution of the temperature sensor.

As a specific example of this conversion calculation, assume that the temperature measurement produces a value of +24.25₁₀ (018.1₁₆) degrees Centigrade. This is represented as follows:

Temperature MS Byte

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
0 (sign)	0 (sign)	0 (sign)	0 (sign)	0 (sign)	0	0	1

Temperature LS Byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1	0	0	0	0	1	x	x

Illustrated a different way, this is 00000001 1000.01xx₂. This value must be multiplied by 1.8₁₀. Converting 1.8₁₀ to a binary representation truncated to six digits of fractional information, this translates into a constant of 0001.110011₂. The conversion calculation follows:

$$\begin{array}{r}
0000s_0 \ 011000.01_2 = 0061_{16} \\
\times \ 00000000 \ 01.110011_2 = 0073_{16} \\
\hline
00101011 \ .10010011_2 = 2B93_{16} \\
+ \ 00100000 \ .00000000_2 = 32.0_{10} \\
\hline
01001011 \ .10010011_2 = 4B93_{16} = 75.574219_{10}
\end{array}$$

It can be seen that this results in a value of 75.574219 to the precision of the calculation. If this result is truncated to two bits of fractional information, the result is 75.5°F which is within 0.15°F of the value arrived at using a calculator and full floating-point arithmetic. This precision was deemed adequate for the accuracy of this application.

Dual-FET Coil Drive

In this application, two Field Effect Transistors (FETs) interface the processor's output port pin to the electromechanical relay. The reset default condition for the MAXQ3212's port pins is a weak pull-up, high state. Therefore, when the processor powers up, its port pins will be at a high state until the application software changes this. If a single n-channel FET (BS-170) is used to switch power to the relay's coil, the reset default polarity of the port pin would cause the relay to activate when the processor powers up. The port pin would remain in that state until the application software changes the pin's direction to an output and sets its state to zero. This state change is not desirable as it allows a transitional period where the relay is briefly activated and then deactivated. To correct this behavior, two FETs are used in series as shown in the schematic. By using two FETs, the reset default polarity of the port pin does not activate the relay.

Example Code

The software accompanying this application note is available for [download](#) (ZIP). This file contains the source code files: Thermostat.asm, 1-Wire.asm, BCD.asm, ThermDisp.asm, and the header file maxq3210.inc that contains the MAXQ3210/MAXQ3212's register address definitions. Also included in this .zip file is the MAX-IDE project file, Thermostat.prj, and the "loadable" hex file, Thermostat.hex. By extracting these files to a common directory, the software can be assembled and executed on the MAXQ3210 EV kit. On September 12, 2007, another .zip file (Temp3Gerber.zip) was added to the an3965_sw.zip file that contains the software. This addition contains all of the Gerber data, drill data, and parts list necessary to fabricate a two-sided printed circuit board for implementing this application.

Conclusion

The MAXQ3212 is a high-performance RISC processor suitable for a wide variety of applications. Because of its small size, low-cost, and high functionality, it is particularly well suited for cost-conscious high-volume applications. This application note describes a digital thermostat based on the MAXQ3212 RISC microcontroller. Using the MAXQ3212, this simple system senses the ambient temperature, compares it to a user-input set point, and switches an output relay after this comparison. The ambient temperature is sensed by a 1-Wire digital thermometer, the DS18B20. The temperature is displayed using a 4-digit, 7-segment LED display driven by an 8-digit LED display driver, the ICM7218 from Maxim Integrated Products. By choosing this microcontroller and the highly integrated display driver, a minimal component count and low-cost system is achieved.

1-Wire is a registered trademark of Maxim Integrated Products, Inc.

Application Note 3965: www.maxim-ic.com/an3965

More Information

For technical support: www.maxim-ic.com/support

For samples: www.maxim-ic.com/samples

Other questions and comments: www.maxim-ic.com/contact

Automatic Updates

Would you like to be automatically notified when new application notes are published in your areas of interest? [Sign up for EE-Mail™.](#)

Related Parts

DS18B20: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)
DS28EA00: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)
ICM7218: [QuickView](#) -- [Full \(PDF\) Data Sheet](#)
MAXQ3210: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)
MAXQ3212: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

AN3965, AN 3965, APP3965, Appnote3965, Appnote 3965
Copyright © by Maxim Integrated Products
Additional legal notices: www.maxim-ic.com/legal