Distributed temperature sensing using an ATmega8 and DS18B20 sensors

This circuit allows you to place digital temperature sensors in up to four different locations and display their readings in Celsius or Fahrenheit.

It is also equipped with an independent alarm system for each sensor. The unit is built around four DS18B20 digital temperature sensors, an Atmel AVR ATmega8A microcontroller and a 16x2 Alphanumeric LCD module.

The DS18B20 comes in a plastic TO-92 or waterproof package and provides a direct digital read-out of its own temperature, from -55°C to +125°C with ±0.5°C accuracy over the range of -10°C to +85°C.

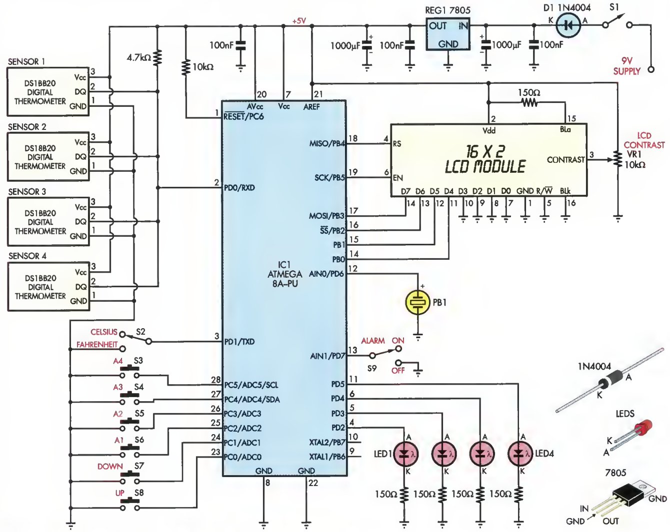
Each DS18B20 has a unique 64-bit serial code and its 1-wire interface requires only one port pin for communication. The micro identifies and addresses devices on the bus using each device's unique 64-bit code.

This allows multiple sensors to function on the same 1-wire bus. Thus, distributed temperature-sensing is simple as one micro can be used to control several DS18B20s distributed over a large area, using either a single cable or a number of cables wired in parallel.

As shown in the circuit, the data lines of all sensors are connected together and on to PD0 (pin 2) of microcontroller IC1. A 4.7kW pull-up resistor is connected between that pin and the 5V supply. All four sensors, plus microcontroller IC1 are powered from the same 5V rail.

IC1 reads out the temperature from each sensor and displays the readings across both lines of the LCD. The screen switches to showing the alarm temperature values if one of the alarm setting pushbuttons (S3-S6) is pressed and held.

To view the alarm temperature



DISTRIBUTED TEMPERATURE SENSING

sensor 1, press and hold A1 (switch S6). While holding S6, press the up or down buttons (S7 and S8) to change the alarm threshold.

When S6 is released, the LCD will return to displaying the temperature readings. The same procedure can be used to view or change the thresholds for the other sensors, holding switch S5, S4 or S3 down instead.

When the respective button is released, the alarm settings are saved into the EEPROM of the micro and will be retained even if power is lost.

Alarm LEDs1-4 are off when each sensor temperature is below the alarm threshold. When the temperature of a sensor rises above that threshold, the respective LED flashes and piezo sounder PB1 produces a tone since output PD6 (pin 12) is driven high.

To reset all the four alarm settings concurrently, to 0°C, press and hold both key A1 and key A2 (S5 & S6) simultaneously for one second. It is also possible to set all the alarms to 25°C or to 50°C by pressing and holding keys A2 and A3 (S4 & S5) or A3 and A4 (S3 & S4) together, respectively.

Switch S9 turns all the alarms on or off. It can be switched to the off position (pulling input PD7, pin 13, low) the first time the circuit is powered up since all the alarm thresholds are initialised to 0°C and otherwise they would immediately go off. Once the thresholds have been set, S9 can be switched to the "on" position.

The selector switch for Celsius and Fahrenheit, S2, is connected to input pin PD1 (pin 3) of the micro and pulls this pin low when set for Fahrenheit display.

Power comes from a 9V battery, via power switch S1 and reverse polarity protection diode D1. It is then regulated to 5V for the micro and temperature sensors using a standard 7805 linear regulator, REG1.

The software, named "distributed temperature sensing.bas", is written in BASCOM and the source code can be downloaded from the SILICON CHIP website.

It is compiled into a HEX file using the free BASCOM trial compiler, before being uploaded to the ATmega8 chip. The BASCOM compiler is available at the following URL:

www.mcselec.com/index.php?option=com_docman&task=doc_download&gid=139

At power up, the software identifies the unique code for all the sensors on the bus. It does this by using an initialization sequence that consists of a reset pulse from the micro, followed by a presence pulse from each sensor on the bus.

In the next stage, the micro issues a "skip ROM" command (CC hex) which allows the micro to access the memory of each sensor. To initiate a temperature measurement and A-to-D conversion, the micro must issue a "Convert T" (44 hex) command.

Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the sensor's scratchpad memory and the DS18B20 returns to its idle state.

In the main loop of the software, the micro issues five commands for each sensor in turn:

1. A reset pulse to start communications.
2. A "Match ROM" command (55 hex) to allow the micro to address a specific sensor on the bus.
3. A "Verify" command to verify if a sensor with the expected ID is available on the 1-wire bus.
4. A "Scratchpad" command (BE hex) to allow the micro to read the contents of the scratchpad (temperature information).
5. A "Read" command to read data from the 1-wire bus into a variable.

Each time the temperature of a sensor is successfully read, the LCD is updated to show its value and the code then compares the temperature reading against the alarm threshold in order to determine whether it needs to trigger the alarm.

Mahmood Alimohammadi, Tehran, Iran. ($50)