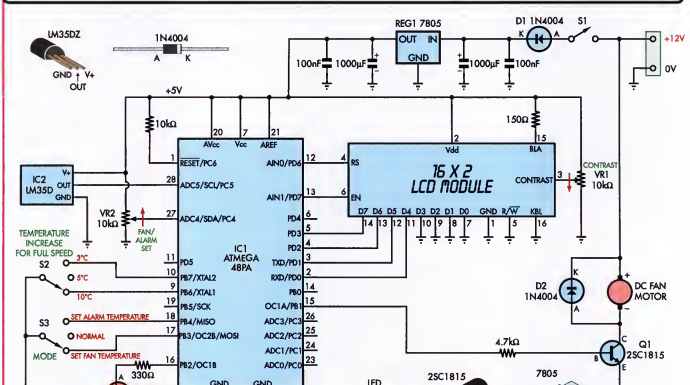
CIRCUIT NOTEBOOK

Interesting circuit ideas which we have checked but not built and tested. Contributions will be paid for at standard rates. All submissions should include full name, address & phone number.



PWM-based temperature-controlled fan

This circuit controls the speed of a DC fan, depending on the output of a temperature sensor, and it can switch the fan on and off and adjust its speed in steps as the temperature increases. The number of degrees increase required to run the fan at full speed is selectable at either 3°C, 5°C or 10°C. It can also sound an over-temperature alarm if the temperature rises above a critical value.

Potentiometer VR2, switch S3 and a 16x2 alphanumeric LCD are used to set the fan-on temperature and alarm temperature. During normal operation, this LCD shows the current temperature, fan cut-in temperature, alarm temperature and current fan speed.

The circuit uses an LM35 precision temperature sensor, IC2, which gives an output of 0V + 10mV/°C, ie, its output is 1V at 100°C. This voltage is fed to the ADC5 analog input (pin 28) of IC1, an ATmega8 microcontroller. Another analog input, ADC4 at pin 27, monitors the voltage at the wiper of potentiometer VR2.

Once the micro has measured the temperature, it decides what speed the fan needs to run at and produces an appropriate PWM signal from its OC1A output at pin 15. This drives the base of NPN transistor Q1 via a 4.7kΩ base current-limiting resistor and Q1 in turn switches pulses of current through the 12V DC fan. The higher the PWM duty cycle, the faster the fan runs.

The PWM output at pin 16 (OC1B) generates a separate PWM signal to drive alarm LED1 via a 330Ω current-limiting resistor. This LED is on when the unit is first powered up but fades out after a few seconds or flashes at full brightness in an over-temperature alarm condition.

Three-position switch S2 is used to select the temperature increase required for full fan speed. IC1 has individually-enabled internal pull-up current sources for each pin and the software switches these on for input pins 9 & 10. Hence, it can determine the position of S2 by reading the voltage at these inputs, as one or the other is pulled low (or neither), depending on the position of S2.

Note that the fan will continue to run for 60 seconds once the sensed temperature drops below the set threshold. During this time, a count-down is shown in the upper-right corner of the LCD screen. The fan then switches off. This prevents the fan from switching on and off if the temperature is hovering near the set-point.

IC1 reads the position of switch S3 via input pins 17 & 18 in the same manner as it monitors S2. S3 is used to set the fan-on temperature threshold and the alarm temperature threshold. During normal operation,

S3 is left in the middle position. To set either threshold, it is switched to one of the other positions and then VR2 is rotated until the desired value (in °C) is shown on the LCD. S3 can then be set back to its centre position and the new setting is stored in IC1's EEPROM.

The LCD interface uses the standard 4-bit configuration, with output pins PD0-PD3 (pins 2-5) used to send data and outputs PD7 (pin 13) and PD6 (pin 12) for control. The LCD backlight is powered via a 150Ω current-limiting resistor from the 5V rail while the contrast is set using 10kΩ potentiometer VR1.

On the LCD, the current temperature is shown after "T:", the alarm temperature threshold after "A:", fan temperature threshold after "F:" and current fan speed step after "Sp:". It also shows a blinking heart symbol as a "heartbeat" at 1Hz to indicate that the unit is operating.

Power comes from a 12V supply via power switch S1, reverse polarity protection diode D1 and 5V linear regulator REG1 which has a pair of input bypass capacitors and two output filter capacitors. Before switching the unit on for the first time, VR2 should be rotated fully clockwise and S3 set to the SET FAN TEMPERATURE position.

The software is written in BASIC and can be compiled into a HEX file to load into the Atmel processor using BASCOM. The source code is available for download from the SILICON CHIP website (*Software-pwm-based temperature-controlled fan.bas*).

Mahmood Alimohammadi, Tehran, Iran. ($60)