Programmable Refrigerator Watchdog

Inside a Raspberry Pi case

by **Jörg Trautmann**
(Germany)

Although only a small circuit, this smart alarm detects when your refrigerator door is open and lets you observe the internal temperature on a 7-segment display. Additional functionality indicates for how long the door is opened and sounds an audible alarm if this exceeds a predetermined time (or a predefined temperature value). The circuit uses only a few components, enabling it to fit inside a low-cost case, such as a Raspberry Pi box.

Features

- Displays current temperature inside refrigerator
- Monitors chill temperature with alarm function
- Monitors how long door is open, with alarm function
- Maximum acceptable chill temperature and door open time are programmable

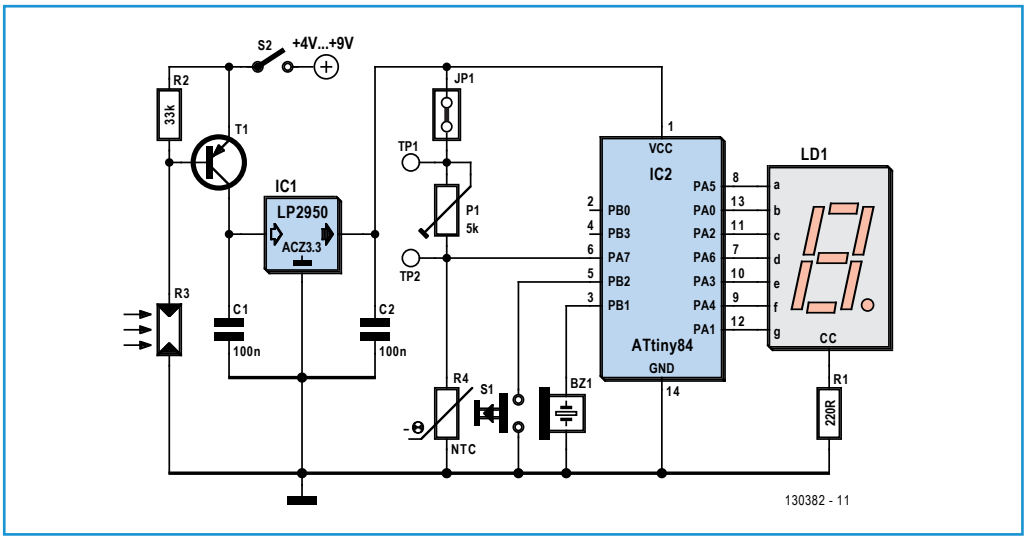


Figure 1.
Only a few components are used in this circuit, which is based around an ATtiny84 microcontroller.

The circuit presented here is based on an ATtiny84 microcontroller, garnished with a few other components. The microcontroller is provided with several 10-bit A/D converters, of which one is used to establish the current temperature inside the refrigerator.

Temperature measurement

For the temperature sensor we selected the widely available KTY81-210 type. In a normal domestic refrigerator the temperature range is generally between 2 and 8 °C (26 to 46 °F). To achieve the best possible measurement resolution we define 10 °C (50 °F) as the highest temperature to be measured. At this temperature the KTY81-210 has a resistance of 1772 Ω, whilst at 0 °C (32 °F) it amounts to 1630 Ω. According to the data sheet the degree of error in the region of 10 °C is ±1.67 °C. Reference measurements with 10 sensors of this type indicated, however, that the actual variation lies below ±1°C and the scatter does not range far beyond this figure.

As the internal reference voltage of the ATtiny84 microcontroller is set at 1.1 volt, the voltage under investigation needs to be connected via a voltage divider (P1/R4). The maximum value of 1100 mV should correspond to a temperature of 10 °C. Based on the known parameters, a value of 3544 Ω arises for resistor P1 (see schematic in **Figure 1**), enabling a theoretical resolution of 5.7 mV/°C to be achieved.

To avoid freak results a series of 10 measurements is taken. Following this the average value is calculated and passed to the display.

Display

For indicating the temperature detected we use a red 7-segment display. As we do not need to indicate two-digit values, a single display element is sufficient. To make the most efficient use of battery power, the 7-segment display is operated in multiplex mode, driven at digit level. The technique used was explained in detail in the Elektor article 'Frugal 7-Segment Display' (July & August 2012 edition [1]). In this way the entire circuit manages to operate with a maximum current draw of just 5 mA.

Press-button S1 is used for programming the maximum permissible temperature along with the permitted 'door open' duration. If this is exceeded, an intermittent alarm (for instance a siren) sounds and the display flashes.

Other components

If you have studied the schematic in **Figure 1** already, you will have spotted the LDR (R3) in the circuit. To save power the microcontroller should be switched on only while the refrigerator door is open. Once triggered, a timer runs to monitor the duration that the door remains open.

The power supply employs a low-drop-out voltage regulator of the type LP2950 CZ3.3, which produces the ideal voltage for the microcontroller (3.3 volts). A 9-volt battery will provide a very long period of use, since the quiescent current in standby mode amounts to a mere 5 µA and the circuit itself still works reliably at 4 volts.

Housing

Homebrew projects that are not intended for use in a laboratory environment deserve to use a case with a physically attractive appearance. Even when you have found something suitable, you frequently need to drill holes and cut out the larger openings, both of which tasks are taxing for many people. After a lengthy time spent searching, I came upon the Raspberry Pi Box. This struck me as heaven-sent as a suitable housing, because:

- In many cases drilled holes are already provided for switches, press-buttons, LDR, 7-segment display and piezo transducer.
- The design is very attractive.
- The average price of well below £5 or $10 is extremely reasonable.

Construction

The circuit is extremely simple to build and for convenience can be made using breadboard. Do consider programming the microcontroller before starting construction. As always the firmware is available for downloading gratis from the page dedicated to this project on the Elektor website [2]. The value of the trimpot resistor P1 is preset with a digital multimeter to be as close as possible to 3544 Ω (remove jumper JP1 during this operation).

Locations for the switch, push-button, LDR and 7-segment display can be determined from the photo of the finished project. The 7-segment display should stand directly above the base of the case. Its pins need to be soldered from the under-side of the circuit board in order to position the display perfectly in the case. The precise location will depend on the particular 7-segment display selected.

Commissioning

When the operating voltage is connected for the first time, after a few seconds an intermittent tone should be heard and a rapidly flashing H symbol should appear. If this is not the case you need to check your construction again carefully.

The next task is to set the maximum allowable temperature. Switch off the device and keeping push-button S1 depressed, re-apply the power. Continue to hold down the press-button for about 5 seconds until a flashing minus symbol is displayed. Now you can release the push-button. At this stage pressing the push-button a second time will set the desired maximum temperature. Do this for around 2 seconds. When the maximum value is reached, release the press-button and wait a few seconds. Once again an intermittent sound should be heard and a rapidly flashing H symbol seen. To check that the preset value has been successfully stored in memory, switch off the device and then power it up again. Now you should see the maximum temperature displayed, flashing rapidly, followed by the H symbol. Since the tempera-

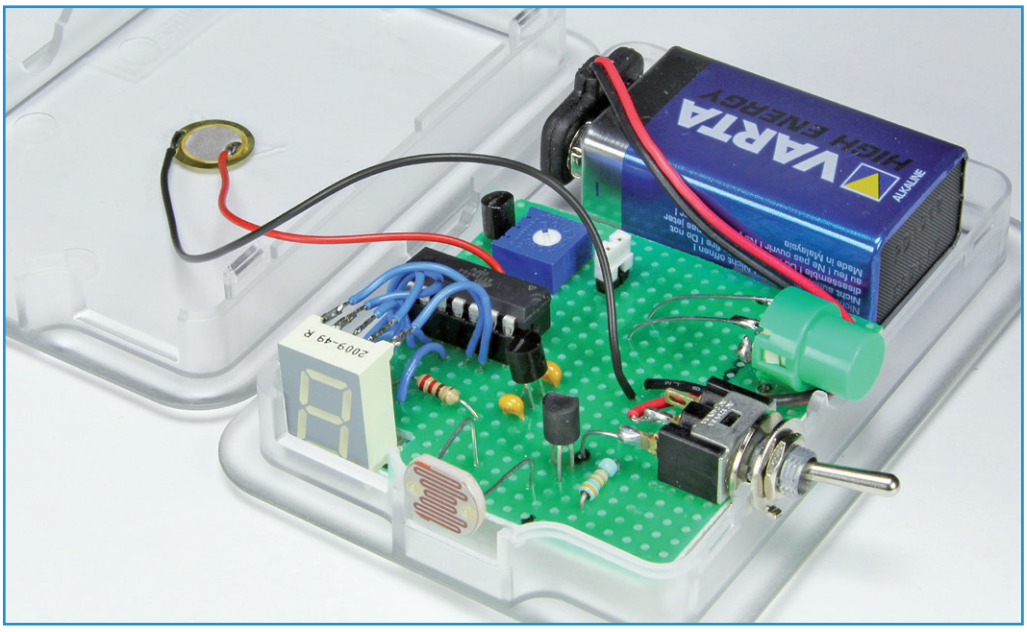


Figure 2.
When built on breadboard the circuit fits exactly inside a Raspberry Pi case.

ture currently exceeds the preset maximum, an intermittent tone is sounded again as an alarm warning.
Now we need to set the maximum time that we allow the door to be open. For this we need to power down the device, operate push-button S1 and power up the device again without releasing the push-button. After about 5 seconds a flashing minus sign appears and after a further 5 seconds this changes to two vertical bars. You can release the button now. Program the allowable 'door open' time desired by pressing the button repeatedly. On the display 1 stands for 10 seconds, 2 for 20 seconds and so on. In this case press the button down each time for about 2 seconds. Once the wanted duration has been released stop pressing the button and wait a few seconds until the H symbol appears and an interrupted tone is heard. The delay before the alarm sounds can be set in this way to a period of between 10 and 90 seconds. If no period is preset, the permitted time open is 60 seconds. To display the last value stored, press the button briefly and the figure will be appear as a flashing display.

In use

Now leave the device in the refrigerator for at least two hours to acclimatize. It is only after this duration it can be guaranteed that residual heat will not to lead to false interpretation of the temperature value displayed. When the refrigerator door is opened now the current temperature should be shown on the display. The device is now on the alert in two senses: if the door is opened the timer runs for the period it is open and the current temperature is re-checked continually.
Experiments with several KTY81-210 sensors have indicated an accuracy of around ±1.5 °C using the set-up procedure described above (and a good multimeter). If you wish to increase the accuracy of temperature measurement further, you can make use of a reference thermometer adjusting trimpot resistor P1 to perform an exact calibration.
Enjoy making this small but practical measurement and alarm device!

(130382)

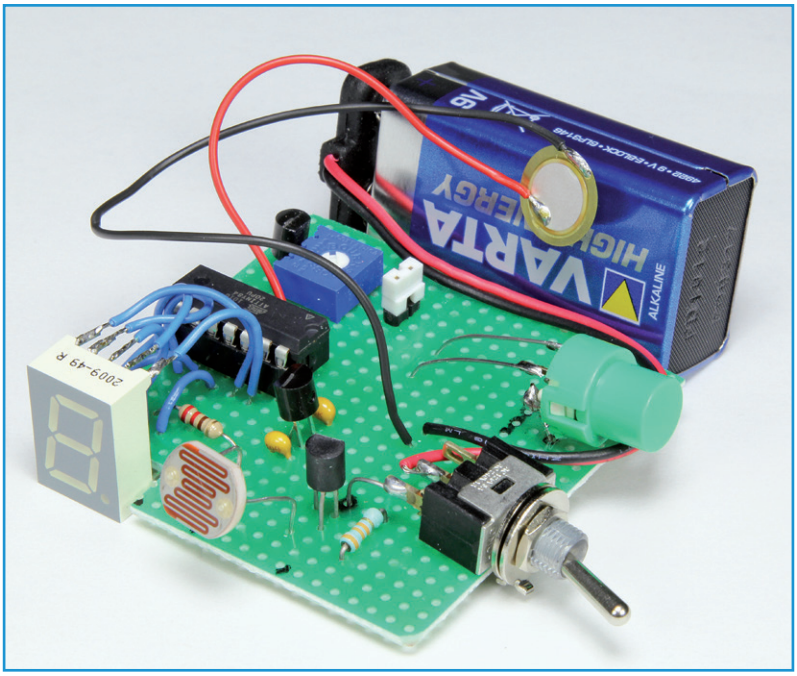
Internet Links

[1] www.elektor-magazine.com/120264

[2] www.elektor-magazine.com/130382

How the program works

Port pin PB1 is configured as an output for operating the piezo loudspeaker, with port pin PB2 contrastingly as an input with a pull-up resistor. The A/D converters are set using an internal voltage reference of 1.1 volt. Operating pushbutton S1 (pin PB2 = Low) carries out setting actions according to the duration of the Low period; following this the value entered is stored in the EEPROM.
A temperature measurement cycle embraces 10 measurements within a period of around 120 ms plus subsequent calculation of the average value. This procedure ensures sufficient accuracy of measurement and stability, as experiments have proved.
Because the program is written in BASCOM, it is relatively easy to understand on the basis of the commentary remarks.



COMPONENT LIST

Resistors
R1 = 220Ω
R2 = 33kΩ
R3 = LDR (1MΩ dark resistance, 5–10 kΩ @ 10 Lux)
R4 = KTY81-210
P1 = 5kΩ preset

Capacitors
C1,C2 = 100nF

Semiconductors
T1 = BC559
IC1 = LP2950 CZ3.3
IC2 = ATtiny84 (programmed, firmware download from [3])

Miscellaneous
LD1 = 7-segment display (common cathode)
BZ1 = Piezo transducer
S1 = pushbutton switch, ITT Schadow, round
JP1 = Jumper
S2 = Toggle switch single-pole, ø 6.2 mm
Raspberry Pi case
14-pin IC socket for IC2
9 volt battery with battery clip