

Programmierbarer Kühlschrankwächter Im Raspberry-Pi-Gehäuse

Von **Jörg Trautmann**
(Deutschland)

Diese kleine Schaltung detektiert das Öffnen der Kühlschranktür und zeigt die Temperatur dann auf einem 7-Segment-Display an. Überdies misst die Elektronik, wie lange die Tür geöffnet ist.

Nach dem Überschreiten einer vorher programmierten Zeit (oder eines programmierten Temperaturwerts) erklingt ein akustischer Alarm. Die Schaltung besteht nur aus einigen Komponenten und passt daher in günstige Gehäuse – wie zum Beispiel die Raspberry-Pi-Box.



Eigenschaften

- Anzeige der aktuellen Kühlschranktemperatur
- Überwachung der Kühltemperatur mit Alarmfunktion
- Öffnungszeitüberwachung mit Alarmfunktion
- Maximal zulässige Kühlschranktemperatur und Öffnungszeit programmierbar



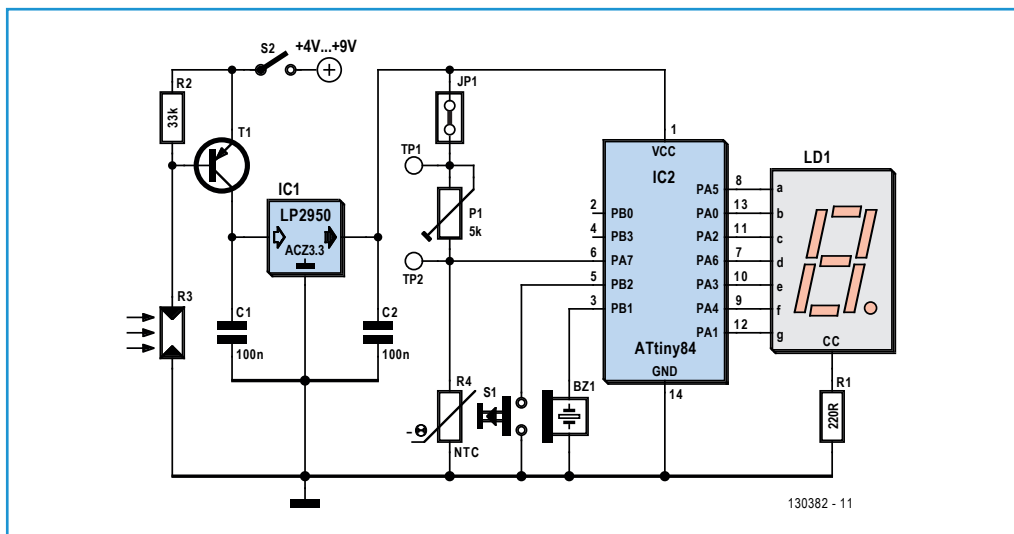


Bild 1. Die Schaltung besteht aus nur wenigen Komponenten, die rund um einen ATtiny84-Mikrocontroller angeordnet sind.

Die hier vorgestellte Schaltung basiert auf einem ATtiny84-Mikrocontroller, garniert mit einigen anderen Bauteilen. Der Mikrocontroller verfügt über mehrere 10-bit-A/D-Wandler, wovon einer dazu verwendet wird, die aktuelle Kühlschranktemperatur zu ermitteln.

Temperaturmessung

Als Temperatursensor kommt der handelsübliche Typ KTY81-210 zum Einsatz. In einem normalen Haushaltskühlschrank herrscht in der Regel eine Temperatur von 2 bis 8°C. Um eine möglichst hohe Messauflösung zu erreichen, wird als höchste zu messende Temperatur 10 °C angenommen. Bei dieser Temperatur hat der KTY81-210 einen Widerstand von 1772 Ω, bei 0 °C liegt der Widerstand bei 1630 Ω. Der Messfehler im Bereich von 10 °C beträgt laut Datenblatt ±1,67 °C. Referenzmessungen mit 10 Sensoren dieses Typs zeigten allerdings, dass die Abweichungen unter ±1°C lagen und die Streuungen sehr gering ausfielen. Da die interne Spannungsreferenz des Mikrocontrollers ATtiny84 auf 1,1 Volt festgelegt ist, muss die zu messende Spannung über einen Spannungsteiler (P1/R4) zugeführt werden. Der Höchstwert von 1100 mV soll bei einer Temperatur von 10 °C anliegen. Anhand der bekannten Parameter ergibt sich für den Widerstand P1 (siehe Schaltung in **Bild 1**) ein Wert von 3544 Ω, womit sich theoretisch eine Auflösung von 5,7 mV/°C realisieren lässt. Um ‚Ausreißer‘ zu vermeiden, umfasst ein Messzyklus 10 Messungen. Danach wird der Durchschnittswert errechnet und zur Anzeige gebracht.

Anzeige

Zur Darstellung der gemessenen Temperatur kommt eine rote 7-Segment-Anzeige zum Einsatz. Da wir keine zweistelligen Werte anzuzeigen brauchen, reicht ein Anzeigeelement aus. Um besonders effizient mit der Batteriespannung umzugehen, wird die 7-Segment-Anzeige im Multiplexbetrieb auf Digit-Ebene angesteuert. Das Verfahren hierzu wurde im Elektor-Artikel ‚Sparsame 7-Segment-Anzeige‘ (Heft 7/8-2012 [1]) ausführlich erläutert. Die gesamte Schaltung kommt so auf eine maximale Stromaufnahme von lediglich 5 mA. Mit Taster S1 kann die maximal erlaubte Temperatur sowie die erlaubte Türöffnungszeit einprogrammiert werden. Wird diese überschritten, ertönt ein intermittierender Ton bzw. eine Sirene und die Anzeige blinkt.

Übrige Komponenten

Wer bereits das Schaltbild in Bild 1 studiert hat, dem wird der LDR (R3) in der Schaltung aufgefallen sein. Um Strom zu sparen, soll der Mikrocontroller nur dann eingeschaltet werden, wenn dies nötig ist: Also bei geöffneter Kühlschranktür. Ab diesem Zeitpunkt läuft ein Timer zur Überwachung der Öffnungszeit. Die Spannungsversorgung erfolgt über einen Low-Drop-Spannungsregler des Typs LP2950 CZ3.3, welcher die für den Mikrocontroller optimale Spannung von 3,3 Volt erzeugt. Eine 9-Volt-Batterie kann sehr lange verwendet werden, da der Ruhestrom im Standby-Modus lediglich 5 µA beträgt und die Schaltung selbst bei 4 Volt noch zuverlässig arbeitet.

Die Verpackung

Selbstbauprojekte, die nicht nur im Labor zum Einsatz kommen, verlangen förmlich nach einem ansehnlichen Gehäuse. Ist dieses gefunden, müssen häufig entsprechende Ausparungen gefräst und Löcher gebohrt werden, was doch meistens als lästig empfunden wird. Nach längerer Suche bin ich auf die Raspberry-Pi-Box gestoßen, welche mir sofort als passendes Gehäuse prädestiniert schien:

- Die Bohrungen für Schalter, Taster, LDR, 7-Segment-Anzeige und Piezo-Schallwandler sind bereits in perfekter Anordnung vorhanden.
- Das Design ist sehr ansprechend.
- Der durchschnittliche Preis von 5 € ist unschlagbar günstig.

Aufbau

Der Aufbau ist recht einfach und kann auf einer Lochrasterplatine erfolgen. Denken Sie daran, den Mikrocontroller vor der Montage zu programmieren; die Firmware kann wie immer von der Elektor-Webseite zu diesem Artikel kostenlos heruntergeladen werden [2]. Der Wert des Trimmwiderstands P1 wird mit einem Digitalmultimeter so genau wie möglich auf 3544 Ω eingestellt (während der Messung Jumper JP1 entfernen).

Die Positionierung für Schalter, Taster, LDR, 7-Segment-Anzeige lassen sich dem Foto des fertigen Gerätes entnehmen. Die 7-Segment-Anzeige sollte direkt auf dem Gehäuseboden aufliegen. Die Pins müssen von unten

auf die Platine gelötet werden, damit die Anzeige perfekt im Gehäuse positioniert ist. Dies ist allerdings von der gewählten 7-Segment-Anzeige abhängig.

Inbetriebnahme

Wird nun zum ersten Mal die Betriebsspannung angelegt, sollte nach einigen Sekunden ein intermittierender Ton zu hören sein und ein schnell blinkendes H-Symbol erscheinen. Ist dies nicht der Fall, muss der Schaltungsaufbau nochmals genau geprüft werden.

Jetzt erfolgt die Einstellung der maximal erlaubten Temperatur. Dazu das Gerät ausschalten. Nun die Taste S1 gedrückt halten und gleichzeitig das Gerät einschalten. Danach hält man die Taste etwa 5 Sekunden gedrückt, bis ein blinkendes Minuszeichen angezeigt wird; jetzt kann man die Taste loslassen. Anschließend wird durch erneutes Drücken der Taste die gewünschte Maximaltemperatur eingestellt. Hierzu die Taste jeweils ca. 2 Sekunden gedrückt halten. Ist der Maximalwert erreicht, die Taste nicht mehr drücken und einige Sekunden warten. Es sollte jetzt wieder ein intermittierender Ton zu hören sein und ein schnell blinkendes H-Symbol erscheinen. Um zu prüfen, ob der eingestellte Wert im EEPROM gespeichert wurde, das Gerät aus- und wieder einschalten. Nun sollte die Maximaltemperatur schnell blinkend angezeigt werden und danach wieder das H-Symbol erscheinen. Da die eingestellte Temperatur jetzt überschritten ist, ertönt auch wieder ein

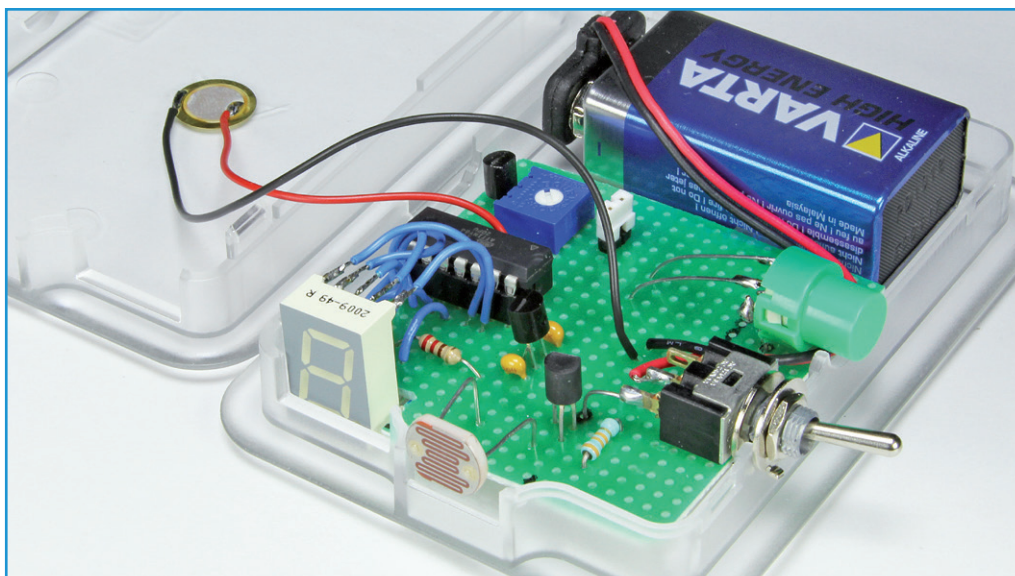


Bild 2. Die auf eine Lochrasterplatine aufgebaute Schaltung passt genau in eine Raspberry-Pi-Box.

intermittierender Ton als Alarmhinweis. Nun erfolgt die Einstellung der maximal erlaubten Öffnungszeit. Dazu das Gerät ausschalten. Dann die Taste S1 gedrückt halten und gleichzeitig das Gerät einschalten. Nach etwa 5 Sekunden erscheint ein blinkendes Minuszeichen, nach weiteren 5 Sekunden sollten zwei senkrechte Balken angezeigt werden. Nun die Taste loslassen. Jetzt kann durch erneutes Drücken der Taste die gewünschte Öffnungszeit eingestellt werden. 1 steht für 10 Sekunden, 2 für 20 Sekunden und so fort. Hierzu die Taste jeweils ca. 2 Sekunden gedrückt halten. Ist die gewünschte Öffnungszeit erreicht, die Taste nicht mehr drücken und einige Sekunden warten, bis das H-Symbol erscheint und wieder ein intermittierender Ton zu hören ist. Die Alarmzeit lässt sich auf diese Weise zwischen 10 und 90 Sekunden einstellen. Wird keine Einstellung vorgenommen, liegt die erlaubte Öffnungszeit bei 60 Sekunden. Um den zuletzt gespeicherten Wert anzuzeigen, drücken Sie kurz die Taste und die Zahl wird blinkend angezeigt.

Verwendung

Lassen Sie das Gerät nun im Kühlschrank mindestens 2 Stunden akklimatisieren. Erst nach dieser Zeit führt die gespeicherte Restwärme nicht mehr zu Fehlinterpretationen des angezeigten Temperaturwertes. Nach dem Öffnen der Kühlschranktür sollte die aktuelle Temperatur im Display angezeigt werden. Das Gerät befindet sich nun im doppelten Sinne in Alarmbereitschaft: Wird die Kühlschranktür geöffnet, läuft der Timer für die Öffnungszeit und die aktuelle Temperatur wird laufend überprüft. Experimente mit mehreren KTY81-210 Sensoren haben mit der oben beschriebenen Einstellmethode (und einem guten Multimeter) eine Genauigkeit von ca. ±1,5 °C ergeben. Wer die Genauigkeit der Temperaturmessung weiter erhöhen möchte, sollte ein Referenzthermometer zu Hilfe nehmen. Über den Trimmwiderstand P1 kann dann eine exakte Kalibrierung vorgenommen werden. Viel Spaß mit diesem kleinen, aber praktischen Mess- und Warngerät!

(130382)

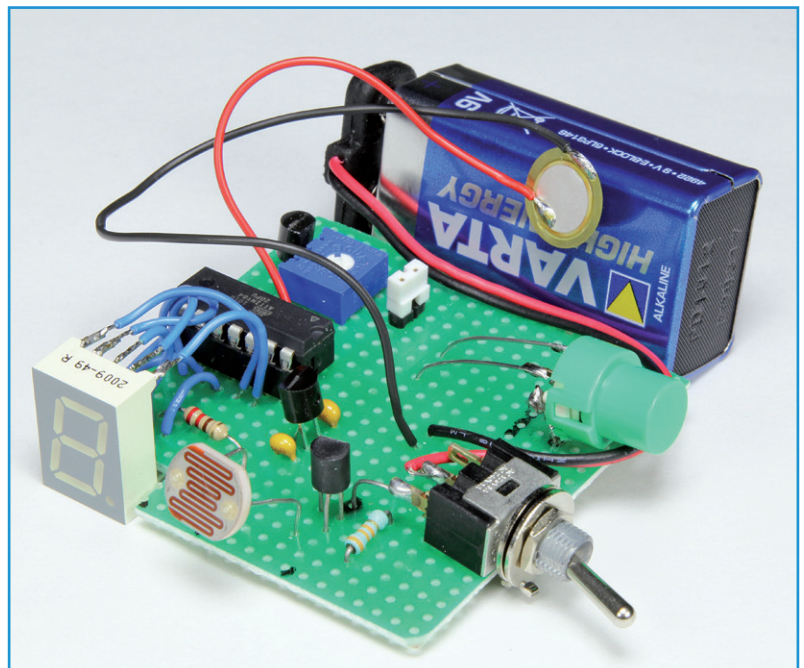
Weblinks

- [1] www.elektor-magazine.de/120264
- [2] www.elektor-magazine.de/130382

So arbeitet das Programm

Portpin PB1 wird als Ausgang zum Betrieb des Piezo-Lautsprechers konfiguriert, Portpin PB2 dagegen als Eingang mit Pullup-Widerstand. Die A/D-Wandler werden auf eine interne Spannungsreferenz von 1,1 Volt gesetzt. Bei geschlossenem Taster S1 (Pin PB2 = Low) werden in Abhängigkeit von der Dauer der Low-Zeit Einstell-Aktionen ausgelöst; anschließend wird der eingestellte Wert im EEPROM gespeichert.

Ein Temperatur-Messzyklus umfasst 10 Messungen innerhalb von ca. 120 ms und eine anschließende Berechnung des Durchschnittswertes. Dadurch ergibt sich eine ausreichende Messgenauigkeit und Stabilität, wie Experimente ergeben haben. Da das Programm in BASCOM geschrieben ist, sollte es relativ leicht anhand der Kommentare zu verstehen sein.



Stückliste

Widerstände:

- R1 = 220 Ω
- R2 = 33 k
- R3 = LDR (1 MΩ dunkel, 5...10 kΩ bei 10 Lux)
- R4 = KTY81-210
- P1 = Trimmer 5 k

Kondensatoren:

- C1,C2 = 100 n

Halbleiter:

- T1 = BC559
- IC1 = LP2950 CZ3.3
- IC2 = ATtiny84 (programmiert, Firmware-Download von [3])

Außerdem:

- LD1 = 7-Segment-Anzeige (gemeinsame Kathode)
- BZ1 = Piezo-Schallwandler
- S1 = Taster, ITT Schadow, rund
- JP1 = Jumper
- S2 = Kippschalter einpolig, Ø 6,2 mm
- Raspberry-Pi-Gehäuse
- 14-polige IC-Fassung für IC2
- 9-Volt-Batterie mit Batterieclip