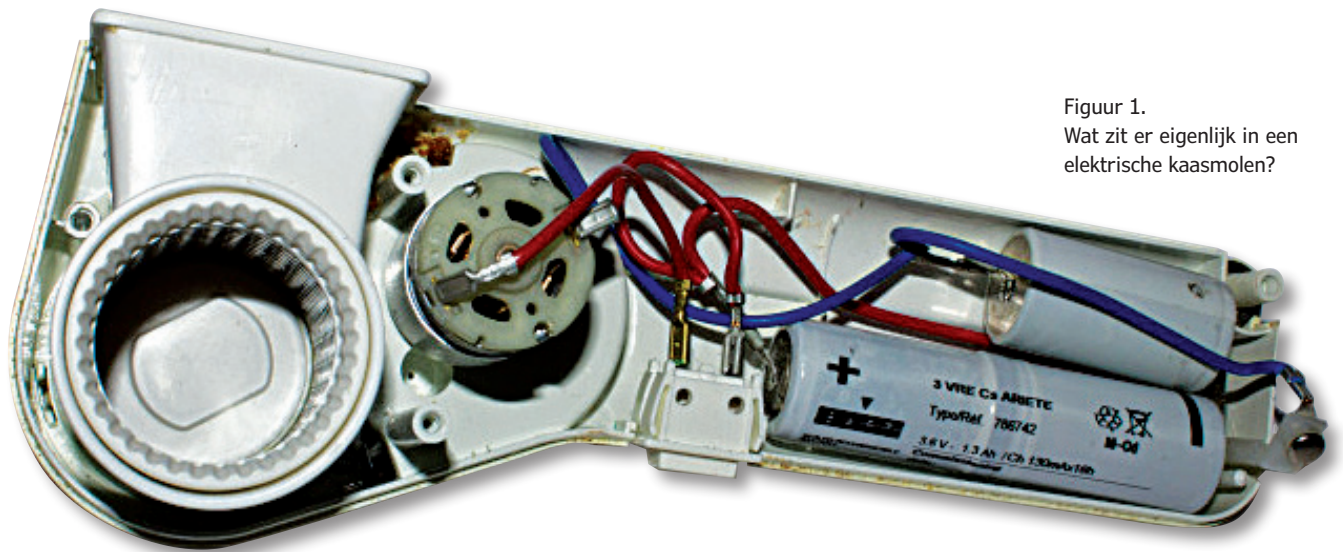


# Een lang leven vol spanning

## Levensduurverlenger voor accu's

Hoewel heel veel apparaten en consumentenelektronica werken op accu's, lijken de fabrikanten zich niet erg druk te maken over de levensduur van de accu. Elektrisch gezien bestaan de meeste apparaten uit niets meer dan een accupakket, een schakelaar en een motor. Daardoor worden de accu's soms volledig ontladen, wat niet goed is voor de levensduur. In dit .POST-project laten we zien hoe we daar iets aan kunnen doen met behulp van een ATtiny45V-microcontroller van Atmel, een vermogens-MOSFET en een paar discrete componenten. Onze 'patiënt' was een oude elektrische kaasmolen. Na de 'operatie' werkt hij veel beter en de levensduur van de accu's is aanzienlijk verbeterd. Natuurlijk is deze schakeling ook te gebruiken voor andere apparaten, bijvoorbeeld voor een kruimeldief.

**Volker Schmidt**  
(Duitsland)



Figuur 1.  
Wat zit er eigenlijk in een elektrische kaasmolen?

### Doel: een pastaschotel met geraspte Parmezaanse kaas

Toen we de behuizing van de kaasmolen open hadden gemaakt (zie **figuur 1**), was het meteen duidelijk wat er mis was. Het accupakket (drie NiCd-cellen van 1300 mAh) was kapot. De oorzaak was inherent aan het ontwerp: de accu was, via een schakelaar (zie **figuur 2**), rechtstreeks verbonden met de motor. Het was dus mogelijk om de accu's helemaal te ontladen, wat heel ongezond is voor NiCd-accu's.

Om te beginnen moesten er nieuwe accu's komen. Het originele accupakket was nog steeds leverbaar, maar er zijn tegenwoordig betere alternatieven te koop. De auteur koos voor 2400 mAh Sub-C NiMH-cellen, die prima passen in de behuizing in **figuur 1**. Ze zijn online te koop. Maar voor het Elektor-prototype zijn we gegaan voor een Li-Ion-cel die een nominale spanning van 3,6 V ( $V_{CC}$ ) levert. Alleen het vervangen van de accu's zou het probleem van te diep ontladen nooit hebben opgelost. Er moest een spanningsbe-

waking worden toegevoegd, maar een analoge oplossing zou, bij zo'n lage voedingsspanning, best lastig te realiseren zijn. Bovendien mocht de print niet te groot zijn, want we wilden hem graag in allerlei apparaten kunnen inbouwen. De keuze lag dus voor de hand: een microcontroller.

**Hardware**

Er is gekozen voor de ATtiny45V-microcontroller van Atmel [1] als besturingseenheid, vanwege zijn 10-bits A/D-converter (ADC), zijn kleine 8-pens behuizing en zijn grote voedingsspanningsbereik dat al begint bij 1,8 V. In de praktijk blijkt hij een minimumspanning van 2,1 V nodig te hebben om een goede A/D-conversie te kunnen doen (met een interne referentiespanning van 1,1 V).

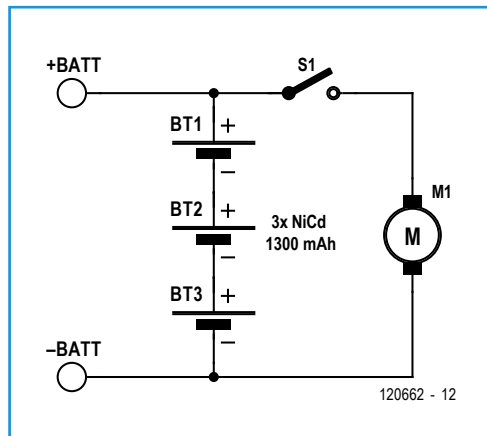
De ATtiny45V heeft ook een heel bescheiden stroomverbruik van ongeveer 0,3 mA en hij is leverbaar in een uitvoering met normale aansluitpennen, maar ook als SMD. De schakeling kan ook worden gebouwd met een ATtiny25V of met een ATtiny85V.

De  $\mu$ C wordt rechtstreeks gevoed door de Li-ion accu aan  $V_{CC}$ . Hij werkt met een klokfrequentie van 4 MHz die wordt opgewekt door de interne RC-oscillator. Er is dus geen extern kristal nodig.

Laten we eens kijken naar het schema in **figuur 3**. MOSFET T1 zit tussen de negatieve pool van de accu (GND) en de belasting (in dit geval een elektromotor). Hij wordt via R1 aangestuurd door poortpen PB3 van de  $\mu$ C. Spanningsdeler R2/R3 levert een spanning van  $V_{CC}/4$  op ADC-input PB4. De spanning moet worden gedeeld omdat de ADC werkt met een interne referentiespanning van 1,1 V. De microcontroller gebruikt dit spanningsniveau voor zijn bewakingstaak, zoals we verderop zullen bespreken. Condensator C2 staat parallel met de accu en voorkomt dat er stoorspanningen op de ATtiny terechtkomen. De reset-lijn wordt via R4 en D1 naar  $V_{CC}$  getrokken.

**Software**

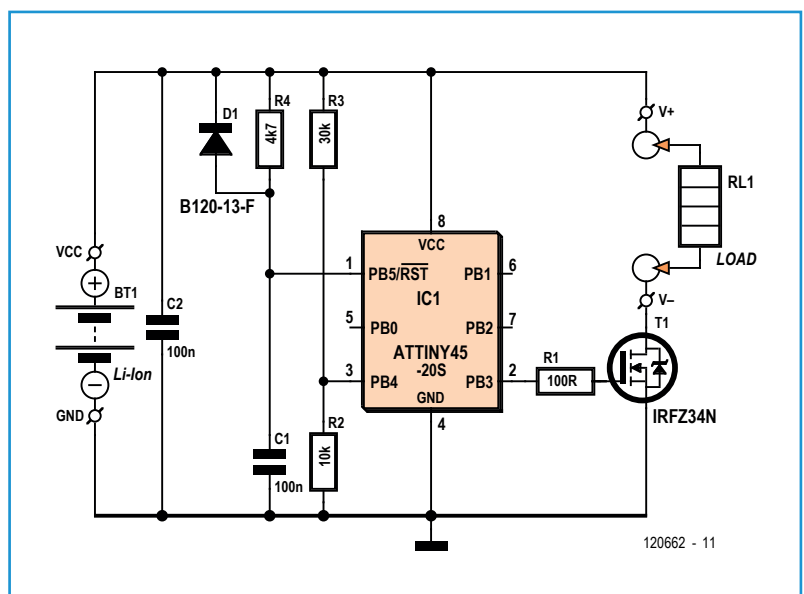
De werking van de firmware is gemakkelijk uit te leggen. De broncode is te vinden op [2]. Het komt er op neer dat in de Main()-functie een lus is geïmplementeerd, die bij elke her-



Figuur 2. Het originele schema van de kaasmolen stelt niet veel voor.

haling de spanning op PB4 inleest en opslaat in een array met 10 posities. Zoals besproken bij de hardware zijn deze waarden evenredig aan  $V_{CC}$  (de accuspanning). Met een eenvoudige berekening is daarmee de echte spanning van de accu in te schatten. Als het array vol is, vergelijkt het programma de waarden met de minimumspanning. Als alle waarden onder het minimum liggen, blijft output PB3 laag, waardoor de MOSFET is uitgeschakeld. De stroom naar de motor is dan onderbroken en zo wordt de accu beschermd tegen te ver ontladen. Zolang niet alle waarden onder het minimum liggen, wordt PB3 hoog en wordt de MOSFET ingeschakeld (waardoor de motor kan gaan draaien). De variabele 'MinVolt' in de code bewaart de minimumspanning in mV. Deze moet afhankelijk van de gebruikte accu worden ingesteld.

Figuur 3. Schema van de levensduurverlenger voor accu's.



Het programma is geschreven in C. Het is ruimschoots van commentaar voorzien en kan worden gedownload van de projectpagina op de Elektor.LABS-website [2]. Op de site zijn ook een project-file voor Atmel Studio 5 en een korte uitleg over het omzetten en vergelijken van de ADC-waarden te vinden. Wat betreft de beruchte instellingen van de fuses van de microcontroller: Let er op dat CKDIV8 op True gezet wordt (dat stelt de klokfrequentie in op 4 MHz).

**De praktijk**

De auteur heeft zijn prototype gebouwd op een klein stukje breadboard en met krimpkous geïsoleerd. Er was ruimte om het te installeren bij de bovenste Sub-C-accu in de kaasmolen (de originele opbouw van de fabrikant is te zien in figuur 1). Voor het gemak hebben we bij Elektor een printje voor de schakeling ontworpen. De layout is gratis te downloaden van de Elektor.LABS-pagina bij dit artikel [2], in de vorm van een PDF-file en als Gerbers- en Eagle-projectfiles. Wie nog niet vertrouwd is met Eagle, of er meer over wil lezen, kan aan de slag met het Elektor-boek *Eagle V6 Getting Started Guide — Learning to fly with Eagle* [3].

**Onderdelenlijst**

**Weerstanden**

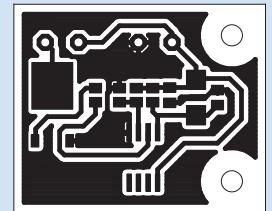
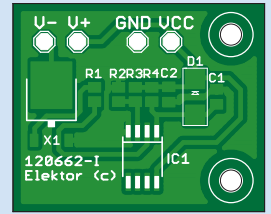
- R1 = 100 Ω, 0,25W, 5%
- R2 = 10 k, 0,25W, 1%
- R3 = 30 k, 0,25W, 1%
- R4 = 4k7, 0,25W, 1%

**Condensatoren**

- C1,C2 = 100 n/16 V min., keramisch

**Halfgeleiders**

- T1 = IRFZ34, N-kanaals MOSFET, 60 V
- D1 = B120-13-F, Schottky-diode, 1 A/20 V
- IC1 = ATtiny45V-10SU, SOIC (geprogrammeerd)



En nu kunnen we eindelijk al die langvergeten, defecte apparaten met accuvoeding weer tot leven gaan wekken. Een lekkere pastaschotel met geraspte Parmezaanse kaas hebben we wel verdiend...

(120662)

**Weblinks**

- [1] [www.atmel.com/devices/ATTINY45.aspx](http://www.atmel.com/devices/ATTINY45.aspx)
- [2] [www.elektor-labs.com/120662](http://www.elektor-labs.com/120662)
- [3] [www.elektor.nl/products/books/electronics/eagle-v6-getting-started-guide.2453009.lynkx](http://www.elektor.nl/products/books/electronics/eagle-v6-getting-started-guide.2453009.lynkx)

