

Vliegende letters

Draaiend LED-display met AVR-controller



Steffen Sorge

Het maken van een afstudeerproject was de aanleiding voor dit listige ontwerp dat het mogelijk maakt om met slechts acht blauwe LED's een 360 graden ronddraaiende lichtkrant op te wekken. Een zuiniger grootbeelddisplay kan men zich nauwelijks voorstellen!

Het idee voor de bouw van een dergelijk display ontstond tijdens het brainstormen over de bouw van een afstudeeropdracht. Hierbij gaat het om de realisering van een origineel object ter gelegenheid van het eindexamen. Een taak die op de toenmalige school van de auteur traditioneel door de technici voor hun rekening werd genomen.

We kwamen op het idee door klokken die volgens hetzelfde principe werken. Deze goedkope producten uit het Verre Oosten tonen meestal echter alleen de tijd en geen tekst.

Het eerste prototype was al na een dag als volgt gerealiseerd: op een plankje werden acht LED's uit de rommelbak, een schakelaar voor de synchronisatie van het begin van de teksten, een AT90S2313, een 9-V-blokbatterij en nog wat kleine onderdelen gemonteerd. Het plankje werd met de as van een gelijkstroommotorje verbonden en zo aan het draaien gezet. Na wat programmeren was op het display al snel het woord 'Hallo' te zien.

Dit eerste prototype (helemaal in de stijl van de Elektuur-workshops) had echter nog een paar nadelen:

- De LED's waren veel te donker.
- De levensduur van de schakelaar bleek begrensd te zijn (hij schakelde bij iedere motoromwenteling).
- de roterende batterij was ook niet alles...

Maar problemen zijn er nu eenmaal om opgelost te worden. Als eerste werden ultraheldere blauwe LED's met 3500 mCd aangeschaft. Vervolgens werd het schakelprobleem opgelost door een schakelaar met Hall-sensor uit een oude printer te gebruiken. De batterij werd door een voeding via sleep-ringen vervangen. En tenslotte moest ook de AT90S2313 zwichten voor een ATmega8.

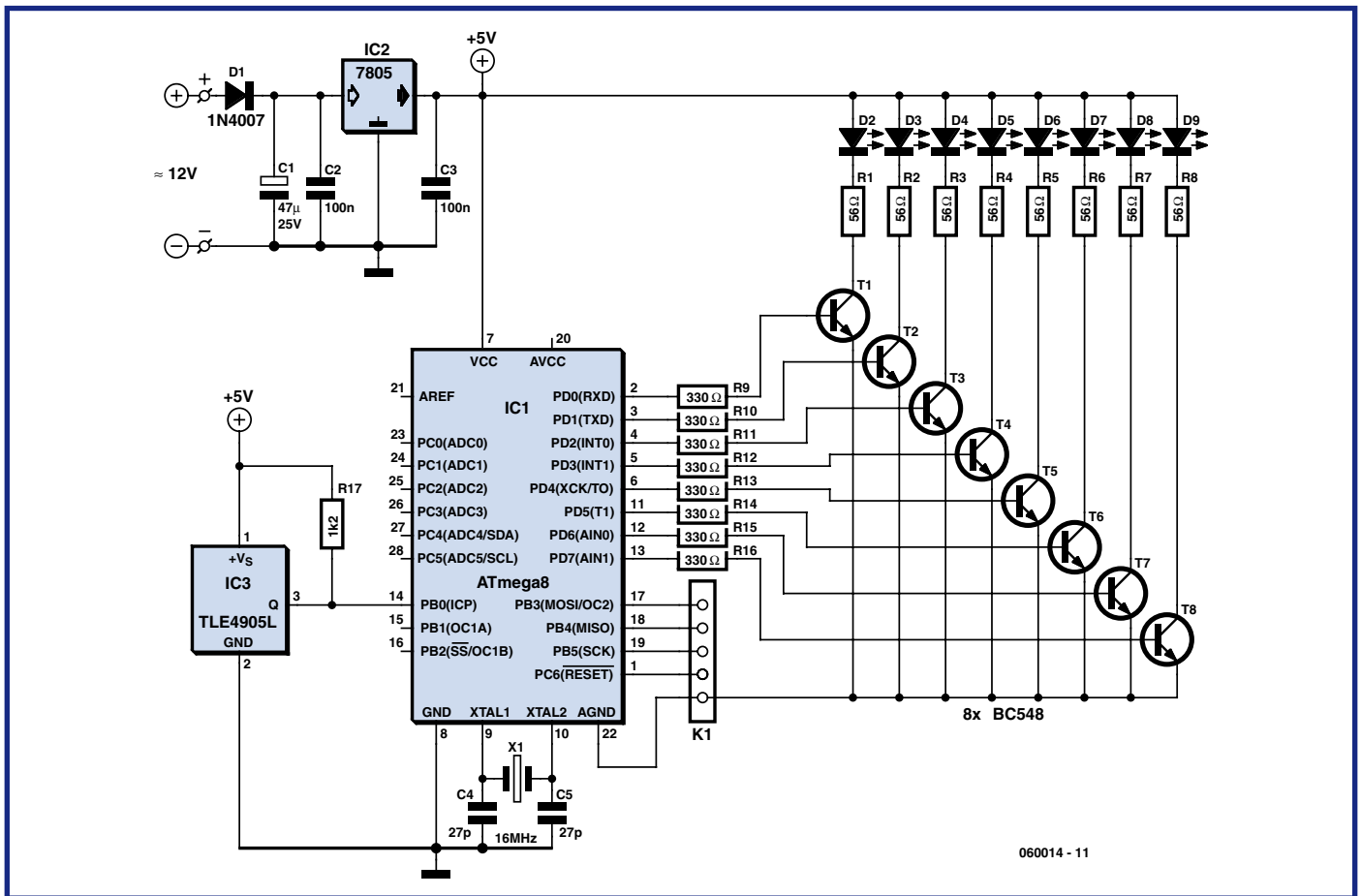
Een controller voor acht LED's

Zoals het schema (figuur 1) laat zien, bestaat de schakeling in feite alleen uit een microcontroller (IC1) die acht

LED's aanstuurt en met een Hall-sensor (IC3) verbonden is. De controller heeft aan externe componenten alleen een 16-MHz-kristal (X1) nodig met twee condensatoren (C4 en C5). Een betrouwbare 7805 (IC2) dient als spanningsregelaar.

Omdat de microcontroller niet voldoende stroom voor de sterk pulserende LED's kan leveren, wordt de besturing door de transistortrappen T1...T8 uitgevoerd.

Omdat de acht draaiende LED's met acht beeldregels een beeld 'schrijven' (weliswaar niet van links naar rechts, maar 360 graden ronddraaiend), moet de microcontroller met een 'beeldpuls' gesynchroniseerd worden. Deze taak neemt de Hall-schakelaar (IC3) voor zijn rekening, die zijn uitgangsniveau verandert zodra hij zich in een magnetisch veld bevindt (dus in de buurt van een magneet). Zonder magneetveld is zijn uitgang hoog; komt hij in de buurt van een magneet, dan wordt hij laag. Deze niveauwisseling start de beeldopbouw, die de software in de microcontroller verder verzorgt.



Figuur 1. De schakeling bestaat hoofdzakelijk uit een AVR-controller die acht LED's stuurt en met een Hall-sensor (IC3) verbonden is.

Software

De ATmega8 is een AVR-controller van Atmel van het type RISC (met gereduceerde commandoset). Het programma voor de lichtkrant werd echter niet in assembler geschreven, maar in het bekende BASCOM-AVR-Basic.

De software met uitvoerige (Duitstalige) toelichting kan men gratis downloaden van de Elektuur-website (bestand 060014-11.zip).

Het programma levert uit de in de controller opgeslagen tekst enkele beeldjes met tekstfragmenten. Deze tekstfragmenten worden omgerekend in kolommen en op een gedefinieerd tijdstip aan de LED's overgedragen. Door de rotatie van de motor ontstaat zo de indruk van tekst die op een denkbeeldige cilinder wordt geschreven.

De controller kan maximaal 255 tekens voor de lichtkrant bevatten. Deze tekens worden in het array 'instring' opgeslagen. Dit bevat drie elementen (drie tekstsegmenten) die na elkaar worden afgebeeld.

De output van tekens wordt door de al

vermelde Hall-schakelaar gesynchroniseerd. Deze levert ook de informatie of de motor draait (als de motor stilstaat, zijn alle LED's gedoofd). De lengte van de over het 'display' lopende tekst is afhankelijk van de cirkelstraal van de roterende LED's, dus van de lengte van de draaiarm. De op de foto's getoonde constructie werkt met een tekstlengte van 30 tekens. In het programma loopt een eindeloze lus die door de Hall-schakelaar gesynchroniseerd wordt en doorlopend 30 tekens aflevert.

Het programma kan de volgende tekens afgeven: hoofdletters van A tot Z, spatie, vraag- en uitroepetekens, punt, komma en koppelteken.

Voor ieder teken is een subroutine geprogrammeerd, men kan zo gemakkelijk zelf tekens toevoegen of veranderen. Een subroutine bestaat eenvoudig uit verschillende schrijfcommando's voor poort D, die de LED's aanstuurt. Op elk schrijfcommando volgt een korte pauze via een wachtlus.

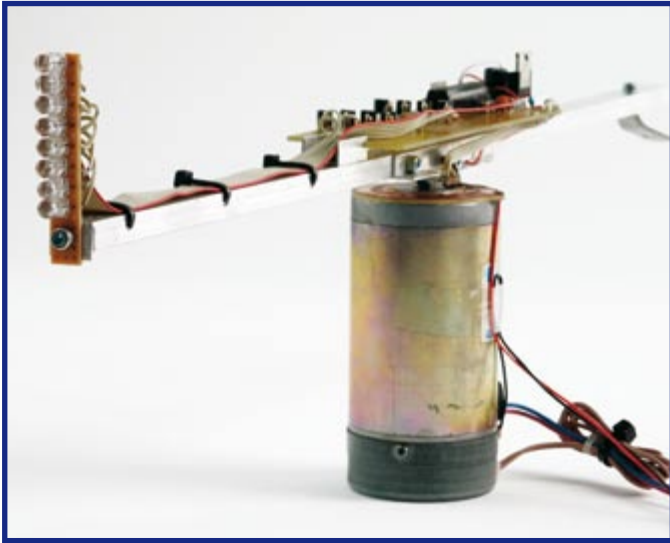
Ieder schrijfcommando produceert een kolom in het desbetreffende teken dat uit meerdere van zulke kolommen

van telkens acht beeldpunten bestaat. Men kan dus een teken heel eenvoudig op ruitjespapier 'ontwerpen' en de kolommen van links naar rechts in bytes omzetten. Als men de wachtlussen (subroutine 'Wa') in de broncode wegdenkt, kan men goed zien hoe een teken er uit ziet. Bijvoorbeeld:

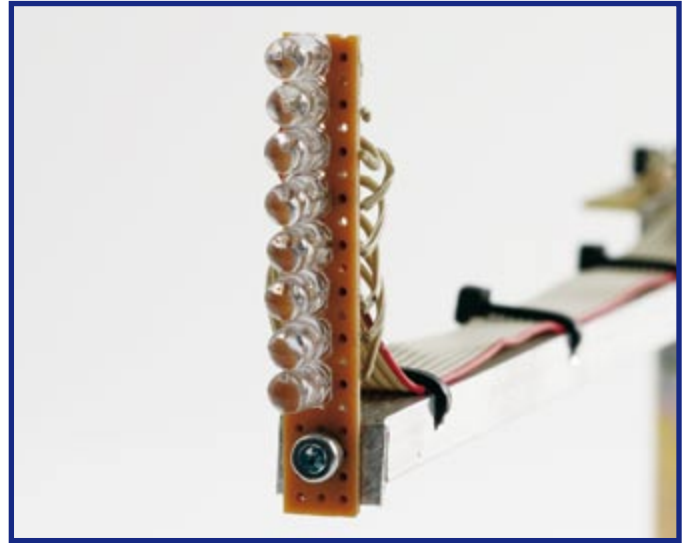
```
sub A
portd = &B11000000
portd = &B00110000
portd = &B00101110
portd = &B00100001
portd = &B00101110
portd = &B00110000
portd = &B11000000
portd = &B00000000
Wa
Wa
end sub
```

Het laatste schrijfcommando heeft altijd de waarde '0', om de LED's te doven. De wachttijd van subroutine 'Wa' moet op het motortoerental worden afgestemd; een optimum vindt men door verschillende waarden uit te proberen.

De weergave begint met de eerste 30



Figuur 2. Zicht op de draaiende arm met de controllerprint.



Figuur 3. Opstelling van de acht LED's aan het uiteinde van de arm.

tekens van het eerste stuk tekst in 'instring'. Na vier motoromwentelingen (bewaakt door de variabele 'Laufer1') wordt de tekst met één teken doorgeschoven (de beginpositie van de tekst wordt in 'Laufer' bewaakt). Als het einde van het eerste stuk tekst bereikt wordt, dan wordt het volgende stuk tekst geselecteerd (variabele 'Strz'). Aan het einde van het derde stuk tekst wordt weer met het eerste stuk tekst begonnen.

Als men parameters zoals toerental en omloopstraal van de ronddraaiende LED's wil aanpassen, dan hoeven slechts enkele kleine veranderingen in het programma gemaakt te worden.

Bij een andere straal (armlengte) is het (van 30 afwijkende) aantal van de op het 'display' zichtbare tekens op twee plaatsen aan te passen.

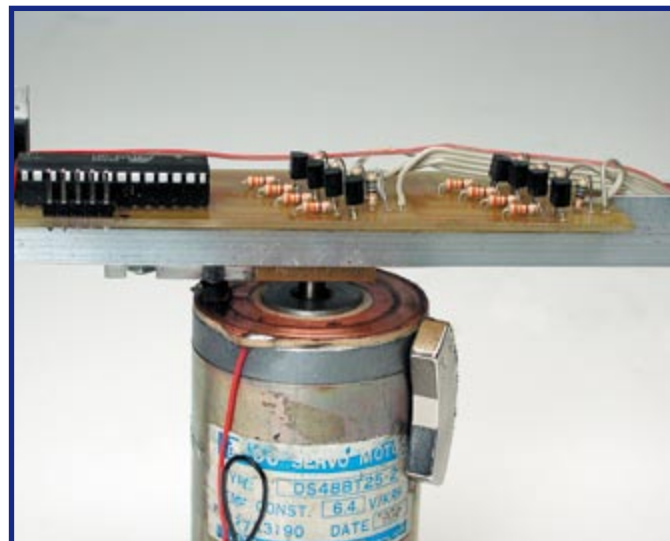
Bij toerentalveranderingen kan de pixellengte (subroutine 'Wa') aangepast worden.

De loopsnelheid van de tekst kan ook worden aangepast.

Het aantal teksten kan eveneens worden veranderd (vereist echter een controller met meer geheugen).

Schema

De schakeling kan met behulp van het schema gemakkelijk worden gebouwd.



Figuur 4. De voeding van de draaiende schakeling loopt via een sleefring.

Voor alle componenten zijn gewone versies gekozen. Ook voor de ATmega8 is een DIL-versie (ATmega8-16PU) toegepast. De weerstanden R9...R16 moeten aan de nominale stroom van de LED's worden aangepast. Men kan ze eenvoudig als serie-weerstand tussen de LED en de 5-V-voedingsspanning berekenen. Bij het prototype leverde dat een waarde van 56 Ω op.

Als hulp bij de bouw kan men de printlayout en de componentenopstelling van de auteur samen met de onderdelenlijst downloaden van de Elek-tuur website. Wel moet men er op letten dat de in de componentenopstelling gebruikte aansluitbenamingen in het schema niet vermeld zijn respec-

tievelijk afwijken. Elektrisch komt de print overeen met het schema.

Er werd gekozen voor een langwerpige, aan de afmetingen van de roterende arm aangepaste printvorm. De LED's zitten op een apart printplaatje vooraan de arm, in een rij boven elkaar gemonteerd (zie foto's). Om de 5-mm-LED's op een normaal 2,54-mm-printraster op een rij te kunnen plaatsen, moet de LED-behuizing aan twee tegenover elkaar gelegen zijden iets worden afgevlakt. De LED-print wordt via een 9-aderige flatcable met de andere print verbonden. De anoden van de LED's worden met elkaar en met +5 V verbonden, de kathoden worden

aangesloten op de bijbehorende serie-weerstanden (R9...R16). D2 (aan R1) is de bovenste LED op de print, D9 (aan R8) is de onderste LED. Als men de rij LED's verkeerd om aansluit (D9 boven, D2 onder), staat de tekst op zijn kop.

Programmering

Het pakket 060014-11.zip bevat een .bas-, een .hex- en een .bin-bestand. In het .bas-bestand staat de broncode voor 'Bascom'. Het .hex- of het .bin-bestand kan men direct in de ATmega8 flashen, die daarmee meteen aan het werk kan. Men krijgt dan de door de auteur geprogrammeerde demotekst als lichtkrant te zien.

Als men een eigen tekst wil programmeren, heeft men Bascom nodig. Daarbij opent men het bestand 'Schrift.bas' en verandert de inhoud van array 'instring'. Nu hoeft de nieuwe broncode alleen nog met F7 gecompileerd te worden.

Na het aansluiten van de voedingspanning (9...12 V) op de schakeling kan de controller worden geprogrammeerd. Daartoe verbindt men de programmeeraansluiting (K1 in het schema) via een geschikte programmer (bijv. PonyProg) met de PC en flasht het bestand Schrift.bin of Schrift.hex in de chip (fuse-bits op 16 MHz extern zetten!).

Als men voor de compilatie de gratis BASCOM-AVR demo 1.11.8.3 wil gebruiken, moet men de tekstlengte

de sleepring een kleine koolborstel, die elektrisch geïsoleerd moet zijn van het aluminiumprofiel van de arm. Aan het motorhuis wordt verder nog een magneet bevestigd die bij iedere omwenteling met zijn magneetveld rakelings langs de op de juiste plaats op de arm gemonteerde Hall-schakelaar gaat en deze daarmee activeert.

De print wordt nu zodanig op de arm gemonteerd dat er geen onbalans is. Voor de voeding verbindt men de massa-aansluiting van de print met de bus (op de motoras) en de positieve aansluiting met de koolborstel. Door uitbalanceren wordt de eventueel nog aanwezige onbalans weggewerkt.

Nu kunnen we de voedingsbron (netstekervoeding 9 tot 12 V) met de sleepring (plus) en het motorhuis (massa)



beperken, omdat de demoversie niet meer dan 4 KB code compileert.

Mechaniek

Als 'rotor' wordt een ca. 50 cm lange arm van aluminiumprofiel gebruikt. In het midden van de arm bevestigd men een busje dat elektrisch geleidend met de motoras verbonden moet zijn. De geleider naar massa wordt nu tegen het motorhuis (!) geklemd. Bovenop het motorhuis wordt een sleepring van printplaatmateriaal (zonder een laag fotolak) aangebracht en met de positieve voedingsspanning verbonden. Aan de rotorarm bevestigd men op

verbinden. Het stroomverbruik volgt in hoofdzaak uit de som van de LED-stromen; daarbij komt dan nog de stroomopname van de motor, als deze door dezelfde netvoeding gevoed wordt.

Zodra de spanning aangesloten is en de arm draait, moet de draaiende tekst op het 'display' zichtbaar worden – zoals op de foto aan het begin van dit artikel.

(060014)

PS. De auteur was niet alleen succesvol met zijn afstudeeropdracht, maar ook met het eindexamen. Hij studeert nu elektrotechniek/informatietechniek aan de Hogeschool in Heide.