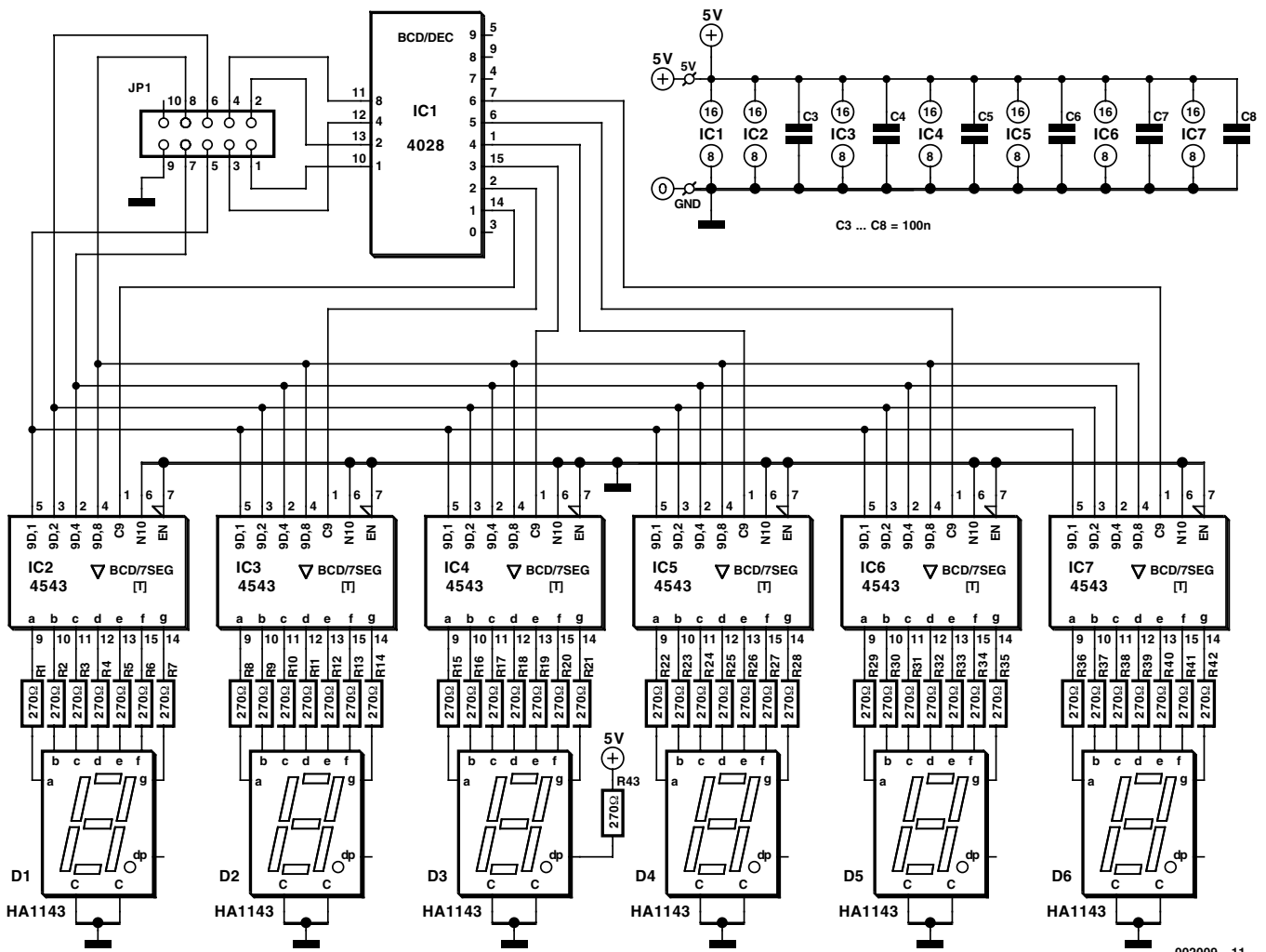


Microprocessors kunnen veel werk verzetten. Vaak gaat het om data vergaren, berekeningen uitvoeren en vervolgens de resultaten zichtbaar maken. Voor deze laatste, vaak erg belangrijke klus is een display nodig. De schakeling die we hier voorstellen laat zien dat met weinig componenten een goedkoop, contrastrijk en daarvoor goed afleesbaar display te maken is.

Ontwerp: Ingo Gerlach

# Flexibel LED-display

## Te combineren met veel microprocessorsystemen



Figuur 1. Het schema van het LED-display. Er is alleen gebruik gemaakt van standaardcomponenten. De gekozen opzet maakt het display zeer flexibel, zonder dat er veel processorbelasting nodig is.

Zonder een goed display zijn de meeste microprocessorsystemen moeilijk te gebruiken. Microprocessors zijn nu eenmaal zwarte doosjes die onzichtbaar voor de buitenwereld snel heel veel rekenwerk kunnen verrichten. Omdat het rekenen geen doel op zich is (de gebruiker wil heldere eindresultaten), is er bij veel projecten behoefte aan een display. Steeds vaker worden hiervoor LC-displays ingezet. De reden is evident:

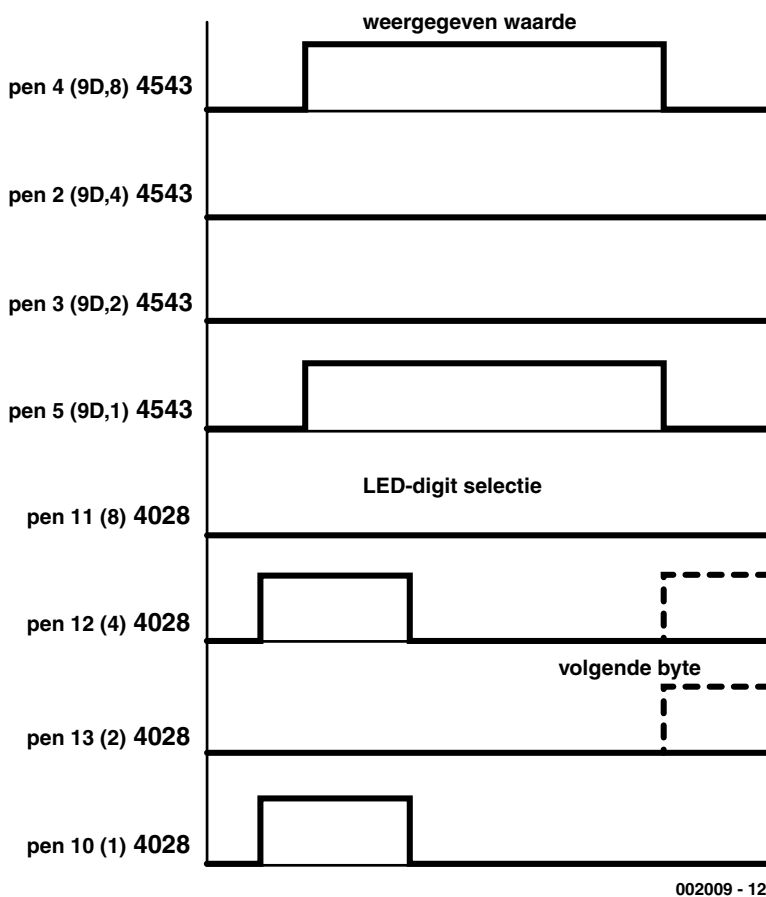
- ▶ op een LC-display kunnen zowel cijfers als alfanumerieke karakters worden getoond;
- ▶ een ingebouwde LCD-controller zorgt dat de aansturing eenvoudig is, en weinig processor-tijd vraagt;
- ▶ LC-displays gebruiken weinig energie;
- ▶ de displays zijn compact.

Naast alle voordelen hebben LC-displays natuurlijk ook nadelen, want:

- ▶ ze geven geen licht en hebben daarom bij slechte belichtingscondities een achtergrondverlichting nodig;
- ▶ ze hebben een beperkte afleeshoek.
- ▶ ze zijn maar in een beperkt temperatuurbereik inzetbaar;
- ▶ ze hebben een relatief laag contrast;
- ▶ ze zijn relatief duur;
- ▶ per type display is soms andere software nodig;
- ▶ de afmetingen van het display zijn beperkt;
- ▶ de levensduur is beperkt.

Samenvattend: soms is een LC-display de beste keuze, in andere gevallen zou een LED-display weer een betere optie zijn. Met dit artikel laten we zien dat voor weinig geld en met relatief lage kosten een LED-display te bouwen is, waarop maximaal zes cijfers getoond kunnen worden. Bij het uitwerken van dit concept stonden aan aantal uitgangspunten voorop. De schakeling moest:

- ▶ zuinig met de I/O-lijnen omspringen;
- ▶ goedkoop te bouwen zijn;
- ▶ weinig rekentijd van de processor vergen;
- ▶ flexibel en dus uitbreidbaar zijn;
- ▶ met vrijwel elke microcontroller te combineren zijn.

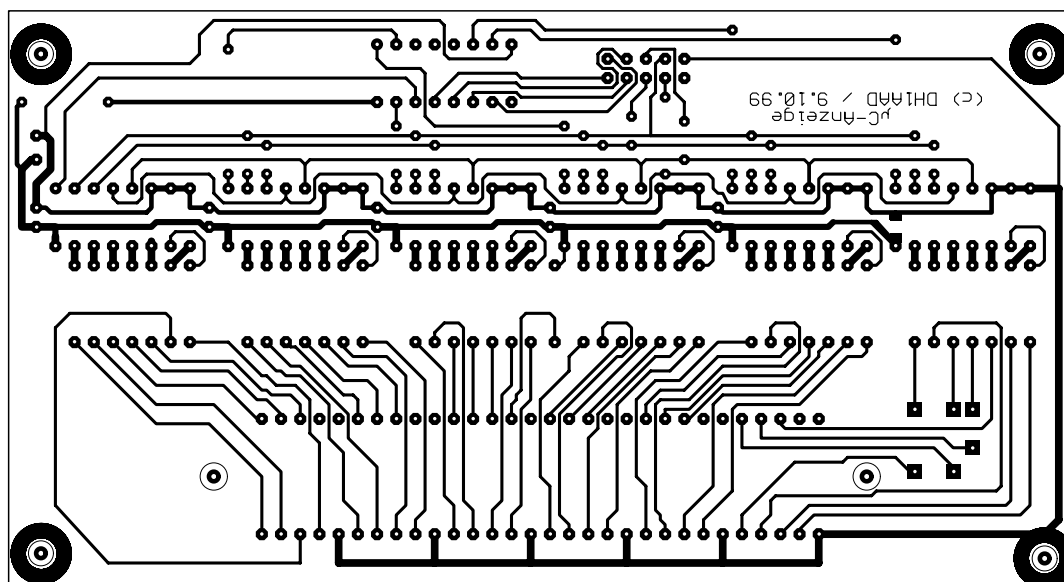
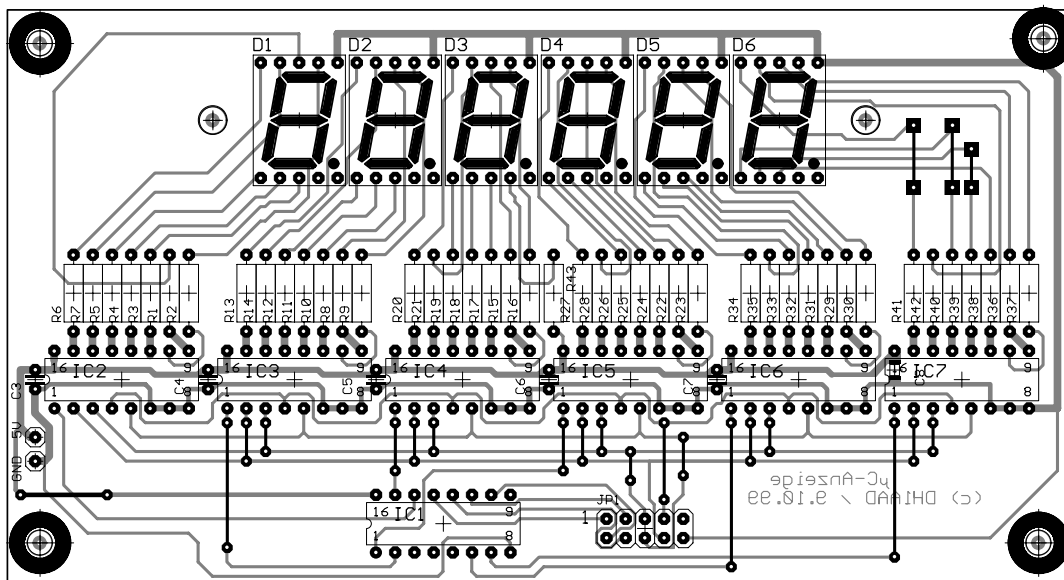


Figuur 2. Het tijdvolgordediagram van de aansturing. De vier bovenste signalen bepalen de code op het display (hier 9), de vier onderste het aangestuurde display (in dit geval 5).

## De hardware

In **figuur 1** is het schema van de schakeling te vinden. Er zijn bij dit project maar twee typen goed en goedkoop verkrijgbare IC's gebruikt. IC1 is een 4028, een eenvoudige demultiplexer en de overige zes IC's zijn van het type 4543, een BCD-decoder met ingebouwde latch en display-driver. De gebruikte displays zijn zogenaamde zevenssegment-displays met een gemeenschappelijke kathode (common cathode). Vandaar dat van de displays de aansluitingen CC met massa verbonden zijn. De overige aansluitingen (a, b...g) zijn direct met een overeenkomstige uitgang van de bijbehorende display-controller (IC2...IC7) verbonden. Tenslotte is er per display

nog een aansluitpunt te vinden: dp. Via deze pen wordt de decimale punt aangestuurd. Bij dit ontwerp is alleen van het derde display (LD3) de decimale punt geactiveerd. Daarom is aansluiting dp van LD3 via een weerstand (R43, 270  $\Omega$ ) met 5 V verbonden. Wordt de decimale punt op een andere plaats gewenst, dan is dat eenvoudig te veranderen. Het hele display is via een header met 2 x 5 aansluitingen, met het microprocessorsysteem verbonden. Van de 10 aansluitingen zijn er overigens maar 9 gebruikt. Aangezien er één pen met massa verbonden is, zijn van de processor slechts 8 I/O-lijnen nodig om het display aan te sturen. Op de pennen 1...4 staat een 4-bits signaal waarmee het gewenste display



Figuur 3. De koper-layout en componentenopstelling van de print waarop het LED-display opgebouwd kan worden.

wordt geselecteerd. De bitcombinatie op deze pennen bepaalt welke uitgang van IC1 hoog wordt. Combinatie 0000 is ongebruikt, op pen 3 (Q0) van IC1 is namelijk geen display aangesloten. De overige vier signaallijnen (5...8) worden gebruikt om de BCD-code op te wekken waarmee het display wordt aangestuurd. Zodra de juiste BCD-code op deze lijnen staat, wordt bij de controller van het gewenste display ingang LD (load, pen 1) hoog gemaakt. Op de overgang van hoog naar laag (neergaande flank) neemt de 4543 de code op de ingangen 1A...1D (pen 2...5) over. Volgens een in het IC geïntegreerde matrix wordt vervolgens bepaald welke uitgangen (A...G) hoog worden. In het display gaan de bijbehorende segmenten branden.

Het hele proces is in het tijdvolgorde diagram van **figuur 2** geschetst. De bovenste vier bits laten zien dat de BCD-code 1001 op de ingangen van de display-controllers wordt gezet. Vervolgens wordt via de 4028 IC5 (0101 = 5 decimaal) geactiveerd. De gekozen aanpak heeft voor de gebruiker twee belangrijke voordelen: Zolang de waarde op het display niet aangepast hoeft te worden, hoeven er geen nieuwe stuursignalen naar het display gestuurd te worden. Verder is het aanpassen van één enkel display mogelijk. Dit zijn twee voorzieningen die de processorbelasting tot een minimum beperken. Over de rest van de schakeling is verder weinig te zeggen. Er wordt gebruik gemaakt van een enkelvoudige voe-

dingsspanning van vijf volt. Per IC is een ontkoppelcondensator gebruikt. De serieweerstanden voor de display-segmenten zijn bij de gegeven voedingsspanning een goede keuze. Moeten de displays helderder oplichten, dan kan de waarde van deze weerstanden (R1...R43) verlaagd worden tot bijvoorbeeld 180 Ω. is juist een lagere helderheid gewenst (en daardoor ook een lager energieverbruik), dan kunnen de weerstandswaarden verhoogd worden.

## De print

Om de nabouw zo eenvoudig mogelijk te houden, heeft de auteur voor dit project een printontwerp gemaakt, dat in **figuur 3** is afgebeeld. De print is niet in de Elektuur Product Service opgeno-

men, dus u zult hem zelf moeten etsen. Dankzij de eenvoudige en enkelzijdige layout zal dat weinig problemen opleveren. Breng als eerste de draadbruggen aan. Daarna worden de overige componenten geplaatst. Gebruik bij voorkeur IC-voetjes en soldeer nauwkeurig. Bij IC2...IC7 is steeds de laatste weerstand zonder componentnummer afgedrukt. Dat is de serieweerstand die gebruikt wordt voor het aansturen van de decimale punt (R43 in het schema). Bepaal dus of er een decimale punt wordt gebruikt en vervolgens waar die

moet staan. Daarna is het aanbrengen van de serieweerstand (270 Ω) voldoende om die punt te activeren.

## Software-voorbeelden

Om het LED-display in combinatie met een microcontroller te kunnen gebruiken, is uiteraard software noodzakelijk. Vandaar dat we bij dit artikel een voorbeeldprogramma (in dit geval een machinetaalroutine voor de Atmel-processoren van het type AT90S1200...8515) hebben afgedrukt. Om toepas-

sing zo eenvoudig mogelijk te houden, is gekozen voor het gebruik van een aparte driver. Dit maakt een flexibele aanpak mogelijk en die driver kan in een applicatie worden aangeroepen. Hierdoor hoeft de besturing maar één keer geschreven te worden, om vervolgens in verschillende systemen toegepast te kunnen worden. **Listing 1** toont de opzet van de driver. De opzet is rechthoe-rechtaan en de werking is dan ook goed te volgen.

In **listing 2** is een kort demonstratieprogramma te vinden. Het programma zet

```

;*****
;* File Name      :LEDDisp.inc
;* Title          :Driver for LED Display
;* Date          :Ingo Gerlach / 10.10.99
;* Version       :1.0 / 11.10.99
;* Version       :
;* µC            :AT90S1200...8515
;*
;* Changes      :
;*
;*****
;
; Main program register variables
;-----
;.def temp      = r16
; Registers / LED
;-----
;.def cntr      = r20 ; counter
;.def dly       = r21 ; delay loop variable
;.def pos       = r23 ; position
;.def byte      = r24 ; byte

; Equates
;-----
;.equ LED_qty = 6 ; number of LEDs
;.equ LED_Del = 45 ; delay
;.equ OutPort = PortB

; Functions
; LED_Blank : switch display on
; LED_Null  : reset display 0 ( Null)
; LED_Show  : show bytes , transport byte (R24),
;            position (R23)

; **** Switch display off
;*****
LED_Blank: ldi cntr,LED_QTY
LedLoop1: ldi temp,192
          add temp,cntr
          out OutPort,temp
          Rcall Led_Delay
          dec cntr
          brne LedLoop1
          Ret

;***** Reset display
;*****
LED_Null: ldi cntr,LED_QTY ; load number of LEDs
LedLoop2: out OutPort,cntr
          Rcall Led_Delay
          dec cntr
          brne LedLoop2
          out OutPort,cntr
          Ret

;***** Show byte
;*****
LED_Show: mov temp,pos ; position in register
          out OutPort,temp; activate BCD-
          to-decimal decoder, LD 4543
          Rcall Led_Delay ; short delay
          mov temp,byte ; value in
          register
          swap temp ; value high
          nibble
          add temp,pos ; goto position
          out OutPort,temp; 4028 / 4543
          move value to display
          Rcall Led_Delay ; short delay
          sub temp,pos
          out OutPort,temp; LD signal
          off, store value
          Rcall Led_Delay
          Ret

;*****
; * Internal functions !!!

; *** Delay LED display
;*****
LED_Delay: ldi dly,LED_Del
LedLoop:  dec dly
          brne LEDLoop
          ret

```

Listing 1. De in machinetaal geschreven driver voor het LED-display, die in applicaties geïntegreerd kan worden.

het display eerst op nul en geeft vervolgens het getal 145675 weer. Wordt de decimale punt achter het derde segment gebruikt, dan is de afgelezen waarde dus 145.675.

Zelf experimenteren met dit flexibele

LED-display is nu eenvoudig mogelijk. Moet de code naar andere processor-families vertaald worden, dan is dat aan de hand van de informatie in de listings eenvoudig mogelijk.

Tekst: Hans Steeman

(002009-1)

```

;*****
;* File Name      :LED.asm
;* Title          :Test program for LED display
;* Date           :Ingo Gerlach / 10.10.99
;* Version        :1.0 / 10.10.99
;* Version        :
;* µC              :AT90S1200...8515
;*
;* Changes :
;*
;*****
;***** Directives
;
.device AT90S1200           ;device type
.NOLIST

.include "1200def.inc"

.list
.listmac

; Show data
; Structure of data
;
;      MSB              LSB
;      7 6 5 4          3 2 1 0
;      0 0 0 0          0 0 0 0
;
;      Select position 1 = 1. 2 = 2. etc
;      Number in BCD code
;
;e.g. 10010010b = 146d = pos. 2, value 9
;
; Main program register variables
;-----
.def    temp            = r16

; Registers / LED
;-----
.def    cntr            = r20 ; counter
.def    dly             = r21 ; delay loop variable
.def    pos             = r23 ; position
.def    byte            = r24 ; byte

; Equates
;-----
.equ    LED_qty = 6          ; number of LEDs
.equ    LED_Del = 40         ; delay
.equ    OutPort = PortB

;***** Interrupt vector table
reset:

;*****
;***** Functions
;*****
;***** Main *****
;
main:
; ldi temp, LOW(RAMEND) ; setup StackPointer for
> 90S1200
; out SPL, temp        ; initialize SPL
; ldi temp, HIGH(RAMEND)
; out SPH, temp        ; initialize SPH

ldi temp,255          ; temp = 255
out ddrb,temp         ; port B output

Rcall LED_Null        ; reset display

mainloop:
; Show 145.675

ldi Pos,1
ldi Byte,1
Rcall Led_Show

ldi Pos,2
ldi Byte,4
Rcall Led_Show

ldi Pos,3
ldi Byte,5
Rcall Led_Show

ldi Pos,4
ldi Byte,6
Rcall Led_Show

ldi Pos,5
ldi Byte,7
Rcall Led_Show

ldi Pos,6
ldi Byte,5
Rcall Led_Show

forever: rjmp forever

; ***** End of main program *****

; *** Include Files ***
.include "LEDDisp.inc"

```

Listing 2. Een voorbeeldprogramma dat laat zien hoe de driver in een toepassing geïntegreerd kan worden.