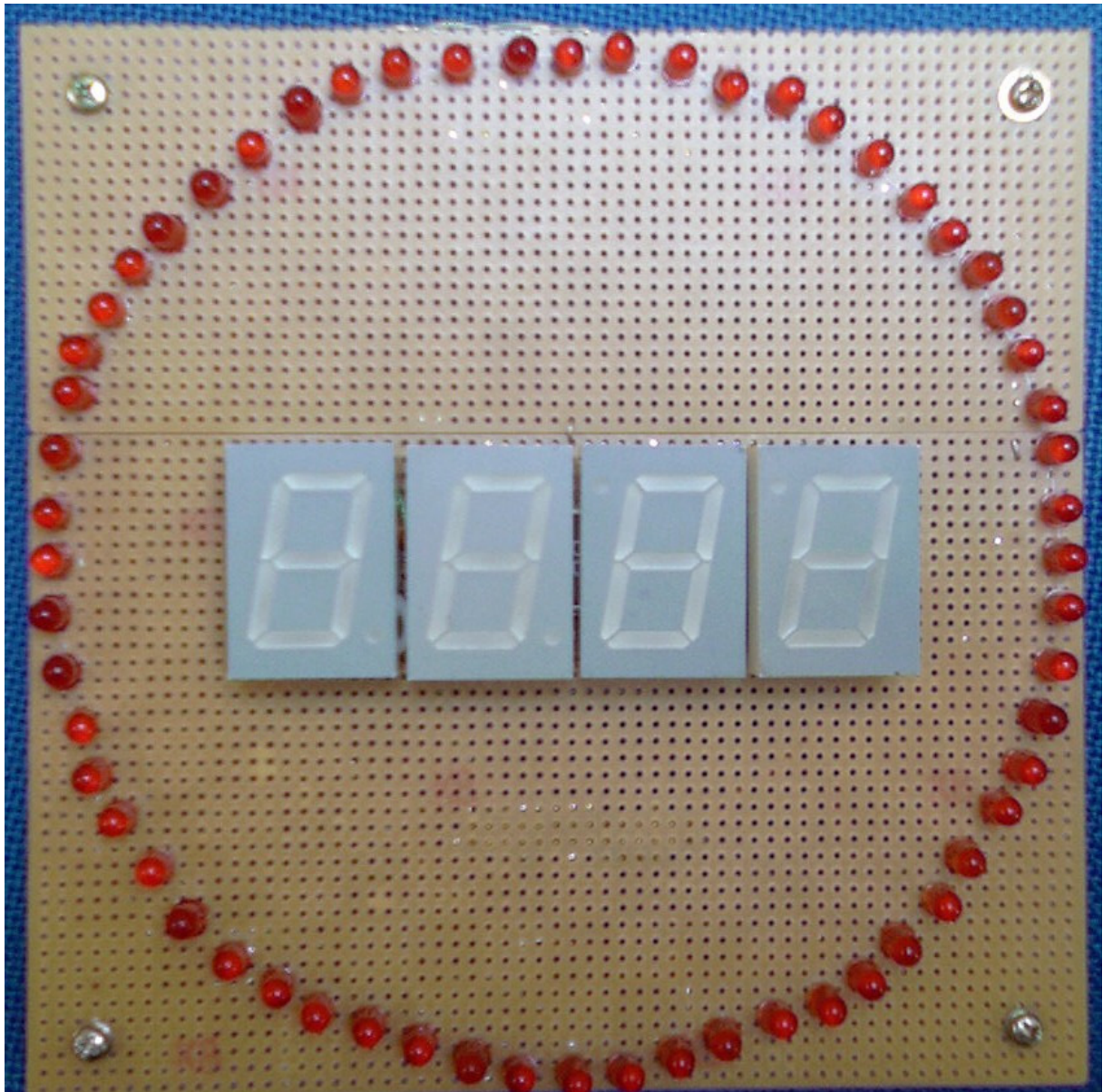


Project documentatie R8C/13 klok
of hoe kwam ik tot....



Auteur: R.J. Wiepkes

Inhoud

Inhoud.....	2
1. De historie	3
1.1. Het idee	3
1.2. De uitgangen	3
1.3. Het display.....	4
2. Het ontwerp van de stuurprint.....	5
2.1. De Processor R8C/13	5
2.2. De CPLD EPM3032	5
2.3. De DS1302	5
2.4. De 3,3V voeding	6
3. Het display.....	7
3.1. Led display met seconden in BCD of binaire weergave.	7
3.2. Led display met 60 seconden leds.....	7
3.3. Led Matrix 24 x 25 Leds.	7
3.4. Grafisch LCD-display	8

1. De historie

Hierover kan ik vrij kort zijn. Tot dusver heb ik mij beziggehouden met de Z80, Z8, recentelijk overgestapt naar de Z8encore. Al die tijd heb ik met belangstelling de artikelen gelezen, waarin gebruik werd gemaakt van de PIC controllers, ATtiny en ATmega etc. Echter hield ik het tot dusver bij de Z8 familie (en dit alles in assembler). Dit had als bezwaar dat alles van begin tot eind zelf moest uitzoeken (er is blijkbaar in de hobbysfeer weinig belangstelling voor deze processor), maar dit was tevens de uitdaging. Maar toen ik (in juli) las over jullie wedstrijd, heb ik meteen het plan opgevat hieraan mee te doen (met de R8C). Ik heb, nadat de plannen vorm had gekregen, de R8C/13 + experimenteerbord aangeschaft en ben eerst gaan kijken of ik de PC-software kon gebruiken om de R8C te programmeren. Ook deze hobbel was genomen. Tevens mijn eerste stappen op Embedded C gedaan, en heb daarbij de potentie van deze controller gezien. Het ontbreken van een I2C interface of SPI interface is wel jammer (de eerste zit wel in de R8C/17).

1.1. Het idee

Ik heb gekozen voor een ontwerp van een klok. Redelijk toegankelijk en in korte tijd opgezet, maar.... Hier zijn ook al veel varianten van ontworpen/beschreven etc., dus er moet iets bijzonders mee zijn. Hierdoor kreeg het ontwerp de volgende uitgangspunten:

- Flexibel
- Naar keuze DCF77 of RTC chip (als het ontvangst van dit signaal onvoldoende is, in Nederland b.v. de waddeneilanden).
- Voor hobbyist toegankelijke techniek (verkrijgbaarheid onderdelen resp. ontwikkelomgeving).

De centrale processor lag vast; de R8C. De verkrijgbaarheid van de R8C/13 is het eenvoudigst, dus ik ga van deze uit.

Je kunt een klok op verschillende manieren vormgeven. De eenvoudigste is het gebruik maken van een LCD-display gebaseerd op de HD64180, al dan niet met backlight. Deze kan rechtstreeks op het experimenteerbord geprikt worden, wat ik dus als eerste stap ook gedaan heb om de DCF77-code te ontwikkelen. Maar dit was voor mij minder interessant, want dit voegt niets toe. Voor andere weergaven heb je meer I/O nodig, met name uitgangen, dus dat werd de volgende stap. Er is gekozen voor een scheiding tussen de stuurprint en de displayprint; zodat de displayprint aan de uiteindelijke model en wensen kan worden aangepast, zonder een nieuwe stuurprint te hoeven ontwerpen. Achteraf blijkt dit meer lasten dan lusten te brengen.

1.2. De uitgangen

Hoe krijg je meer uitgangen. Hier zijn verschillende wegen te bewandelen. We gaan gebruik maken van een I/O expander. Maxim heeft er een heleboel mooie, de MAX7300 (20 of 28-bits poort), de MAX6956 (20- resp. 28-bits LEDdriver poort). Deze zijn echter voorzien van een I2C interface en die hebben we niet op de R8C/13. Okee, de MAX 7301 en de MAX6957 zijn SPI-varianten, en die zijn makkelijker aan te spreken, ook met de R8C en een paar pinnen als seriële interface, maar de verkrijgbaarheid blijft een probleem. Veel leveranciers (ook Conrad) heeft ze niet, en daar komt bij, dat ze alleen in SMD zijn te krijgen. Een variant die in dat opzicht wel interessant lijkt, is de ST M5450 (of Micrel MM5450), een 34-bits LEDdriver poort, die in DIP40 en in PLCC44 te verkrijgen is. Deze wordt serieel ingeklokt (klok en datalijn). Echter, dit IC heeft ook een niet te onderschatten nadeel. Het laden in de

driver register gebeurd op de 36^{ste} klokpuls, maar als (om een of andere reden) een klokpuls te veel of te weinig is voorbijgekomen, dan is de data uit sync, zonder dat de software dit kan corrigeren of herkend!

Er is gekozen voor flexibiliteit, verkrijgbaarheid en lage kostprijs. De processor stuurt een CPLD, de EPM3032 of EPM3064 van Altera aan, beide in PLCC44 behuizing. Dit is een IC, waarvan de interne logica geprogrammeerd kan worden via een JTAG interface. Hier kom ik later nog op terug. Deze wordt door de R8C aangestuurd via 6 aansluitingen (4 ingangen en 2 I/O poorten) en heeft de beschikking over 24 in-/uitgangen, waarvan de functie volledig programmeerbaar is. Tevens kunnen de poorten individueel 25mA opnemen of afgeven (sinken of sourcen).

Voor deze combinatie is een stuurprint ontworpen.

1.3. Het display

Door de gekozen opzet is het display helemaal vrij. Bij een handige aanpak van de programmering van de CPLD is het mogelijk het display model aan te passen, zonder dat je de software van de R8C hoeft aan te passen. Ik noem hier een paar voorbeelden:

- LED-display, bestaande uit 6x 7segments displays (of opgebouwde uit losse LEDS). Weergave uren/minuten/seconden resp. dag/maand/jaar. Aangestuurd via multiplexing.
Extra benodigde onderdelen, 6x BD140 (PNP). Gebruikte uitgangen: 6 (display select) + 8 (segmenten) = 14.
Nadeel: de processor moet het display steeds refreshen (en voldoende snel om knipperen tegen te gaan).
- LED-display, bestaande uit 6x 7segments displays (of opgebouwde uit losse LEDS). Weergave uren/minuten/seconden resp. dag/maand/jaar. Aangestuurd via CD4543's. Gebruikte uitgangen: 4 (data) + 6 (select lijnen) + 6 (de 6 display punten) = 16.
Extra onderdelen: 6 x CD4543.
Nadeel: Geen zelfgedefinieerde karakters op het display te tonen.
- **LED-display**, bestaande uit 4x 7segments displays (of opgebouwde uit losse LEDS). Weergave uren/minuten resp. dag/maand resp. jaar. Aangestuurd via multiplexing. Seconden in binaire of bcd-weergave. Deze LEDs worden ook in datumweergave gebruikt om de weekdag aan te duiden.
Extra onderdelen: 4x BD140 (PNP).
Gebruikte uitgangen: 4 (display select) + 8 (segmenten) + 7 (seconden) + 1 extra voor het vaag oplichten van de 7 secondeleds in datum weergave: 20.
Nadeel: de processor moet het display steeds refreshen (en voldoende snel om knipperen tegen te gaan).
- LED-display, bestaande uit 4x 7segments displays (of opgebouwde uit losse LEDS). Weergave uren/minuten resp. dag/maand resp. jaar. Aangestuurd via CD4543's. Seconden in binaire of bcd-weergave. Deze LEDs worden ook in datumweergave gebruikt om de weekdag aan te duiden.
Gebruikte uitgangen: 4 (data) + 6 (select lijnen) + 4 (de 4 display punten) + 7 (seconden) + 1 extra voor het vaag oplichten van de 7 secondeleds in datumw. = 22.
Extra onderdelen: 4 x CD4543.
Nadeel: Geen zelfgedefinieerde karakters op het display te tonen.
- **LED-display**, bestaande uit 4x 7segments displays (of opgebouwde uit losse LEDS). Weergave uren/minuten resp. dag/maand resp. jaar. Aangestuurd via 1 x M(M)5450. Seconden in 60-LED-weergave.

Gebruikte uitgangen: 2 (clk/data) + 12 (5seconden-leds) + 9 voor de overige 48 LEDs (CharliePlexing) en 1 extra voor het vaag oplichten van de 5seconden-leds = 24.

Extra onderdelen: Micrel MM5450 of ST M5450

Nadeel: De te tonen info kan uit sync raken, zonder dat de software er erg in heeft..

- **Matrix LED display.** Dit is natuurlijk volledig vrij. Hier noem ik als mogelijkheid 24x25 LEDs, opgebouwd met 5x8 matrix displays (of volledig discreet). Voor de aansturing wordt 8:1 multiplexing gebruikt. Extra onderdelen: UDN2981 (of 8x BD140 (PNP)) + 3x EPM3032 (PLCC44 behuizing)

Gebruikte uitgangen: 8 (mux uitgangen) + 6 (aansturing EPM3032s) = 14.

Extra onderdelen: UDN2981 (of 8x BD140 (PNP)) + 3x EPM3032

Nadeel: Multiplexing moet geheel door de processor worden verzorgd. Dit kost de processor snel de helft van de capaciteit. Nadeel van de toepassing van de UDN-chip is de grote spanningsval van ca. 2V. Dit zal met name in LV-circuits (3,3V) en batterijgevoede schakelingen een nadeel zijn.

- **Grafisch LCD-display.** hierbij geldt: mits het aantal benodigde aansluitingen kleiner is als 24, kan elk display gebruikt worden. Ik heb een 128x64 display liggen (YH12864) (met een touch screen) en zal deze als voorbeeld gebruiken.

Gebruikte uitgangen: 8 (data) + 5 (selectielijnen display) + 5 (touchscreen) = 18.

Van de voorbeelden, waarvan de eerste woorden vet zijn, zijn de schema's meegeleverd. Van de onderstreepte tevens de software.

2. Het ontwerp van de stuurprint.

Zoals eerder vermeld, zijn de hoofdonderdelen van de stuurprint: de processor (R8C/13 module), de CPLD (EPM3032), een RTC (MAX1302) en een 3,3V voeding. Deze print is ontworpen met gebruikmaking van een mix van SMD en gewone onderdelen. Zie schema R8C13-001.

2.1. De Processor R8C/13

De R8C/13 module is gekozen, omdat deze het gemakkelijkst te krijgen. In dit ontwerp wordt de processor op 3,3V gebruikt.

Ingang P4.5 is de ingang voor het DCF77 signaal. P1.0 – P1.7, P0.4 en P0.5 zijn apart naar buiten gevoerd. Deze aansluitingen kunnen vrij worden gebruikt.

P0.1 t/m P0.3 worden gebruikt om de RTC mee aan te sturen.

P0.6, P0.7 en P3.0 – P3.3 worden gebruikt om de CPLD mee aan te sturen. De functie van de lijnen worden aangepast aan het aan te sluiten display.

2.2. De CPLD EPM3032

Deze chip van Altera bevat 32 macrocellen, 4 ingangen en 26 programmeerbare in-/uitgangen en zit in een PLCC44 behuizing. Daarnaast kan de chip in-system geprogrammeerd worden via de JTAG-interface. Daar dit type een 3,3V type is, is een aangepaste JTAG interface nodig van de versie die in HG 2002 stond (ik zag later dat de aangepaste versie op de print van het FPGA project was opgenomen). Indien er meer macrocellen gewenst zijn, dan kan ook de EPM3064 gebruikt worden, welke pin-compatible is.

2.3. De DS1302

Een RTC, welke via een 3-draads seriële interface aan te sturen is. Er is voorzien in een backup systeem met naar keuze een GoldCap (0,1 Farad/5,5V) of een accu. De DS1302 is voorzien van een laadcircuit, zodat er geen voorzieningen nodig zijn buiten de RTC chip om.

Helaas kwam deze pas op 2 september binnen, dus is de aansturing hiervoor nog niet software opgenomen

2.4. De 3,3V voeding

Hier zit een oude vertrouwde regelaar in. Omdat 3,3V niet in de standaard reeks van de 78xx regulators zit, is hier gekozen voor een andere oude bekende, de LM317, waarvan ik laatst een schema zag in de halfgeleidergids van 1971. Deze kan voldoende stroom leveren en is instelbaar op 3,3V. Verder nog een gelijkrichter op de print, zodat ompoling geen probleem is. Verder is het een goed uitgangspunt om elke print te voorzien van een afvlakcondensator, dus die heb ik er ook op geplaatst.

3. Het display

Zoals eerder gemeld, ga ik vier verschillende modellen nader uitwerken voor het display. Deze vereisen helaas allemaal een andere programmering van de processor en de CPLD op de stuurprint.

3.1. Led display met seconden in BCD of binaire weergave.

De cijfers worden via 4:1-multiplexing aangestuurd. Er zijn 4 PNP-transistoren van het type BD140 of vergelijkbaar nodig, omdat de stroom voor een display te hoog kan oplopen voor een uitgang van de CPLD. Tevens is een hogere spanning nodig, omdat ik LED displays gebruik met een segment spanning van 4,2V, dus worden deze tevens toegepast als level-shifter. Overigens zijn de uitgangen van de CPLD 5-V tolerant, dus er hoeven geen maatregelen genomen te worden ter bescherming van de uitgangen. (zie ook schema Display -001 voor het schema en blokschema CPLD-001).

De seconden kunnen naar keuze (dit vereist alleen een aanpassing in de programmering van de CPLD) in BCD of binair worden weergegeven. Dit is via een poort op de CPLD en een jumper instelbaar gemaakt.

3.2. Led display met 60 seconden leds.

In dit geval is gekozen voor een niet-gemultiplexte opzet (i.v.m. minder storing). Dit is met name interessant als de DCF77 ontvanger gebruikt wordt in de randgebieden van het bereik van de zender in Frankfurt.

De cijfers worden nu via de M(M)5450 LED driver aangestuurd. Deze wordt serieel aangestuurd via een tweetal aansluitingen aan de CPLD (zie ook schema Display-002 en blokschema CPLD-002).

Aanvankelijk was het de bedoeling om charlieplexing toe te passen voor de LEDS. Hiervan zouden dan 48 LEDS met 9 lijnen kunnen worden aangestuurd en de 12 5-secondeLEDs, welke continue vaag moeten oplichten.. Deze leds worden dus direct aangestuurd. Overigens kan dit vaag oplichten uitgeschakeld worden.

Voor het iets makkelijker aansturen is gekozen voor een matrix van 8x6.

3.3. Led Matrix 24 x 25 Leds.

In dit model kan gekozen worden om alles als tekst weer te geven, maar men kan ook met symbolen en dergelijke gaan werken. Ook het aantal LEDS in de X- en Y-as kan aangepast worden. Dit ontwerp kan bijvoorbeeld ook gebruikt worden als basis voor een lichtkrant met één of meer regels. Echter dit kan niet onbepaald uitgebreid worden, want op een bepaald moment loop je tegen de snelheidsbeperking van de processor aan.

Ik heb gekozen voor dit ontwerp omdat ik een zandloper wil afbeelden op de klok, die elke minuut leegloopt en omgekeerd(!) wordt. Tevens zal tijd en datum op het display afgebeeld worden.

Om dit te realiseren is gekozen voor 8:1-multiplexing voor de rijen en drie EPM3032 voor het aansturen van de kolommen. Hiervoor zou ook de M(M)5450 voor gebruikt kunnen worden, al zal de stroom dan wat lager gehouden moeten worden(max. 15mA i.p.v. 25mA).

De schema's voor de print is Display-003 en het blokschema van de CPLD op de stuurprint CPLD-003. Het blokschema van de CPLD op de displayprint heeft CPLD-003-DP.

Het display kan ook aangesloten worden volgens Display-003b. Dit spaart een CPLD uit, maar kost een 74HC238 extra.... En qua prijs maakt dat niet veel uit.

3.4. Grafisch LCD-display

Het laatste voorbeeld van een display is er één waarbij geen gebruik van LEDs wordt gemaakt (behalve als backlight). Ik heb hiervoor een display gebruikt van 128x64 pixel (display maat ca. 6 x 5 cm) met een KS1080 controller en een touch screen (met een AD7843 als AD-converter). Dit betekent dus dat de processor niet hoeft zorg te dragen voor het refreshen van het display. Het display is uitgevoerd met 8 data ingangen en 5 controle signalen. Deze worden door de CPLD aangestuurd. Verder kan de backlight aan- en uitgezet worden. Het contrast wordt met een instelpotmeter ingesteld. Zoals tevens op het schema Display-004 is te zien, wordt de AD7843 ook via de CPLD aangesloten. In blokschema CPLD-004 is de uitwerking voor de CPLD opgenomen. Als er geen Touchpanel is, dan had dit display ook zonder CPLD kunnen worden aangesloten op de R8C/13.

P1.0 – P1.7 aan DB0 – DB7

4 van de 5 controle signalen op P3.0 – P3.3.

1 controlesignaal op P0.7.

Tevens kan dan de LM317-T ingesteld worden op 5V (alle onderdelen, m.u.v. de CPLD, kunnen op 5V werken).

4. Het project

Nadat de stuurprint (zie R8C-uC-board.pdf) besteld was bij The PCBshop, wilde ik toch alvast met de software beginnen. Ik heb toen de plannen uitgewerkt voor achtereenvolgens de 2x16 regelig LCD display met HD64180, een grafisch LCDscherm, een LED-display met 60 LEDs en een LED matrix display. Omdat ik nieuw was heb ik eerst programma's gezocht op basis van deze processor. Één hiervan was het project Elektor7_DCF77. Deze is uiteindelijk mijn uitgangspunt geweest. Ik heb er heftig in zitten wijzigen, maar één ding is vrijwel ongewijzigd, want ook in mijn omgeving, aan de rand van het ontvangstbereik van de DCF77-zender, is de ontvangst redelijk (mits de afstand zonder afscherming tussen ontvanger en processor groter is als 50cm).

Na aanvankelijk problemen te hebben met het kunnen programmeren van de processor. Dit kwam uiteindelijk door twee oorzaken: de uitgangsstroom van de usb-poorten van mijn notebook waren niet stabiel genoeg en doordat de prolific chip op het experimenteerbord zich steeds op een andere RS232-poort instelde (dan eens COM7, dan COM10 en op COM1 of 2.... Gaf de driver steeds een foutmelding.

Daarna een eigen usb-rs232 omzetter gekocht (nog geen 8 euro), ook met een prolific chip erin (kwam ik later achter), en deze pikte probleemloos COM2 op. Daarna ook het experimenteerbord met een externe (9V) voeding gevoed en de problemen deden niet meer voor, totdat..... de ID-check melding voorbij kwam. Met de notebook kreeg ik het probleem niet opgelost, dus naar mijn gewone PC, aangesloten via USB (zowel voeding als interface), proberen te flashen (met één van de niet aangepaste demo's) en bingo. Dit is meerdere keren voorgekomen en meestal als bij het debuggen iets grandioos mis ging.

De print was nog niet binnen, maar ik wilde toch verder met de software, dus toen maar het

4.1. Grafisch display

Zolang de stuurprint niet binnen is, een oplossing gezocht zodat ik met het grafisch display kon stoeien. Met behulp van 2 IC's en enkele losse onderdelen op experimenteerprint heb ik dit display op dezelfde aansluitingen als de standaard HD61480 display (4 datalijnen, Register Select en Enable). Zie het schema R8C13-Grafic1.pdf.

Dit display heeft een twee controllers van het type KS0108b aan boord, die samen met een KS0107 het display van 128x64 pixels refreshed. Details zijn te vinden in het bestand ks0108b.pdf. Bij dit display worden de 8 datalijnen, 2 chip select lijnen, een Register Select, een enable en een resetlijn aangestuurd.

Na enig stoeien kreeg ik de eerste pixels op het scherm. Omdat deze chip zelf geen karaktergenerator had, moest ik zelf karakters gaan definiëren. Ook de adressering was even lastig. Het display is nu in de volgende delen opgedeeld:

Bovenste en onderste 'regel' zijn gebruikt als versiering. (Een regel is steeds 8 pixels hoog). De zevende 'regel' bevat dag en datum.

In de linker helft van 'regel' 2 t/m 6 toont een geanimeerde zandloper en in de rechter helft wordt een analoge wijzerklok getoond. Hiervan is ook een foto bijgevoegd. De bijbehorende software is te vinden in R8C13-Grafic_4508.zip

In het Excel bestand Grafic_images.xls zijn de ontwerpen van de images opgenomen

De broncode is te vinden in de submap Display1

De stuurprint is er nog niet...., dus op naar de volgende uitdaging, een

4.2. LED-display met 60 seconden LEDs

Omdat het maken van een print en duur is en lang op zich laat wachten (het ontwerpen zelf is zo gebeurd), heb ik deze op een, of liever gezegd, twee experimenteer printen gemaakt met een totale oppervlakte van 16 x 16cm.

Hier zijn een aantal ontwerpen gepasseerd:

Eerst een ontwerp, waarbij op de displayprint alleen maar LEDS/Displays, weerstanden en (ter buffering) 2 ULN2803's en 4 PNP transistors. Dit ontwerp is opgenomen in R8C13-NonMux60-1.pdf

Het tweede ontwerp maakte dus gebruik van de MM5450.

Het was even puzzelen en rekenen, waar de 60 LEDs moesten worden geplaatst, maar (samen met 4 7-segment displays van 25mm hoog) lukte dit hierop. Uiteindelijk bleek dit echter vele malen lastiger te zijn dan ik van te voren kon bevroeden; ik was meerdere dagen bezig om het tot een geheel te maken. Tijdens dit opbouwen kwam ook de bestelde MM5450 binnen. Toen de boel opgebouwd was, had ik nog steeds geen stuurprint, dus....

4.3. LED matrix display

Gaan we verder met het matrix display.

Ten eerst maar eens enig rekenwerk doen en kijken of de R8C genoeg reserves heeft om dit display aan te sturen. Gelukkig bleek dit het geval, tenminste, zolang de resolutie niet te hoog wordt. Ik ben uitgegaan van 24x25 pixels (opgebouwd met matrix displays van 5x8 LEDs).

Eerst het ontwerp. Deze is te vinden in bestand R8C13-Matrix1.pdf. Om de multiplex verhouding niet te hoog te maken, is gekozen voor 8:1 multiplexing. Hierbij wordt gebruik gemaakt van 8 PNP-transistoren (BD139) voor de anodes van de LEDs, er wordt gebruik gemaakt van drie CPLDs van het type EPM3032 (PLCC44 behuizing) en er worden een partij weerstanden gebruikt. En daar is de stuurprint!!!

4.4. De stuurprint

Het is intussen half augustus, dus de tijd begint te dringen. Ik heb de print in elkaar gezet (zie foto R8C13-uC-board.jpg). De RTC-chip is nog niet binnen, maar voor de rest kan ik aan de slag. Het monteren van de PLCC44 chip is even een geduldwerkje, maar is prima te doen. Daarna alle weerstanden en condensatoren gemonteerd, de voet voor de R8C13 en de voeding rond de LM317. Testen, en de print is in orde bevonden. Nu eerst de Altera CPLD testen met de JTAG interface. Quartus II 6.0 SP1 opgestart en de programmer uitgeprobeerd.... En contact. Daarna met een schema een ontwerp gemaakt om in- en uitgangen te testen. Dit werkt. Daarna de R8C13 erop gezet.... En dat werkt NIET!. Het bleek, dat de op deze print gebruikte spanning van 3,3V te laag is voor de R8Ckey uit de HG 2006. Daarna heb ik eerst de R8C13 op jullie experimenteerboard gezet, en via 7 draadjes een verbinding gemaakt tussen beide printen. Dit is zonder meer mogelijk omdat de Altera CPLD 5V bestendige ingangen heeft, ook al wordt de chip uit 3,3V gevoed. Nu werkt het wel. Alleen dit is niet wat ik wou.

Toen bedacht ik dat ik nog een MAX3160 had liggen, een RS232/RS485 driver voor 3,3V met ingebouwde spanningsverdubbelaar, zodat de minimale spanningsniveau tenminste bereikt wordt. Alleen is dit een chip met een pootafstand van 0,65mm! Hier soldeer je niet even draadjes aan vast. Gelukkig had ik nog een print van mijn laatste Z8-project liggen, waar deze op werd gebruikt en zo gezegd, zo gedaan. Was het solderen al lastig van de CPLD, deze

is helemaal lastig. Maar 't is gelukt en nu werkt het wel. De stuurprint (m.u.v. de MAX3102) werkt.

4.5. Het grafisch scherm op de stuurprint

Het ontwerp werkte al met de twee MC14508 IC's, en nu wordt de grafische display (via een klein verloopprintje, voorzien van een 5V voeding, een transistor voor het aan en uitzetten van de backlight en een instelpot voor contrast) aangestuurd via de CPLD. Hierbij is gebruik gemaakt van de in bestand CPLD-Grafic2.pdf getoonde programmering. Hierbij is gekozen voor schema, omdat dit voor veel mensen beter herkenbaar is als Verilog of AHDL. Dit werkt.

De R8C-broncode is te vinden in Display2_cpld.

Over naar....

4.6. Het display met 60 secondenLEDs.

Zoals eerder beschreven, was het erg veel werk om het display op experimenteerprint op te bouwen. Ik ging ervan uit, dat alles goed was aangesloten en begon de aansturing hiervan in CPLD (zie CPLD-NonMux60.pdf) en R8C13 te programmeren. De signalen, die uit de CPLD komen, zijn dan precies volgens verwachting, echter de uitwerking op de displayprint is heel erg vreemd. Ik heb daarna nog drie aanpassingen gedaan in het ontwerp, maar dit heeft niet mogen baten. Het is mogelijk dat er LEDs omgedraaid zitten maar door de complexiteit kan ik dit er niet snel genoeg uitkrijgen. Het is intussen 1 september en alle documentatie moet nog rond gemaakt worden, dus is het hiermee afgelopen. Mocht ik tijd over hebben, dan ga ik hier nog mee verder. In de technische beschrijving kom ik hier dan nog op terug.

De R8C-broncode van dit project is dezelfde als van 4.1. Deze is te vinden in de submap Display1

5. Conclusie

Dit is een heel interessante controller, maar dit type (R8C/13) is erg beperkt. De eerste interessante opvolger is de R8C/17, die een I2C i.p.v. een 2^e UART aan boord heeft, of de R8C/1B, die zowel de 2^e UART als I2C aan boord heeft. De beschikbaarheid en tijd heeft mij doen besluiten mij voor dit project de R8C/13 te gebruiken.

De problemen rond ID-check, het feit dat je bij debugging alleen gebruik kan maken van de externe oscillator en de regelmatig voorkomende communicatiefouten hebben mij doen besluiten dit als een interessant project te zien, maar dat ik toch verderga met de Z8encore.

Deze processor heeft de genoemde beperkingen en problemen niet. Ik ben op dit moment bezig een E-Blocks uC-module hiervoor te ontwerpen (voor de Z8F08xx en de Z8F6423).