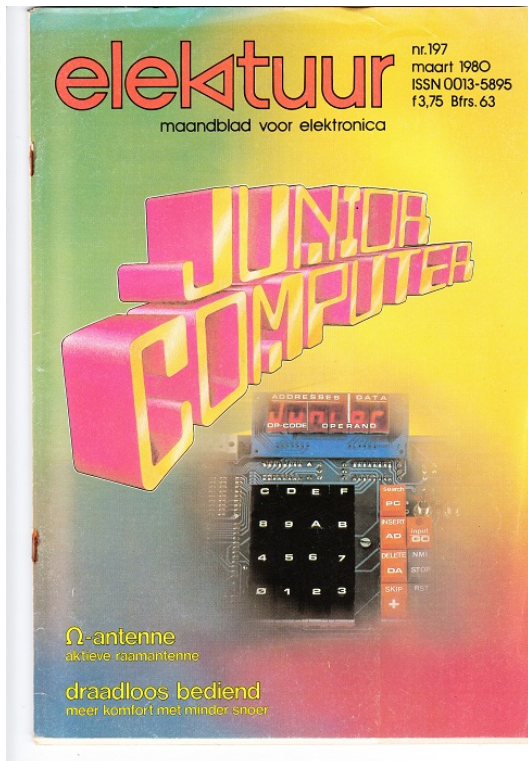


## Junior computer

### Junior computer.

Ooit in de begin tijd van de computers ca 1980, toen deze voor de hobbyisten beschikbaar kwamen, kwam het populaire maandblad elektuur met de "junior computer". (tegenwoordig Elektor). Ik was toen nog iets te jong, maar omstreeks 5 jaar later , medio 1985 , heb ik deze computer alsnog gebouwd. De computer was een kloon van de toentertijd populaire KIM-1. De computer was gebouwd rond de 6502 microprocessor familie. Later zou deze computer opgevolgd worden door Accorn Atom en de legendarische commodore 64, een computer die voor een groot publiek beschikbaar kwam. Op internet is hier veel over gepubliceerd. Graag verwijs ik dan ook nog even naar de site van [Hans Otten](#). Hans heeft ontzettend veel werk gedaan en gepubliceerd op zijn site. Veel retro elektronica en oude 'hobby' computers waar ik regelmatig dankbaar gebruik van maak!!



*Klik op het maandblad om het originele artikel te lezen.*

**Eigenschappen van deze computer:**

Microprocessor Rockwell: R6502 1MHz

RAM geheugen 1kb (2x 2114)

EPROM geheugen 1kb (2708)

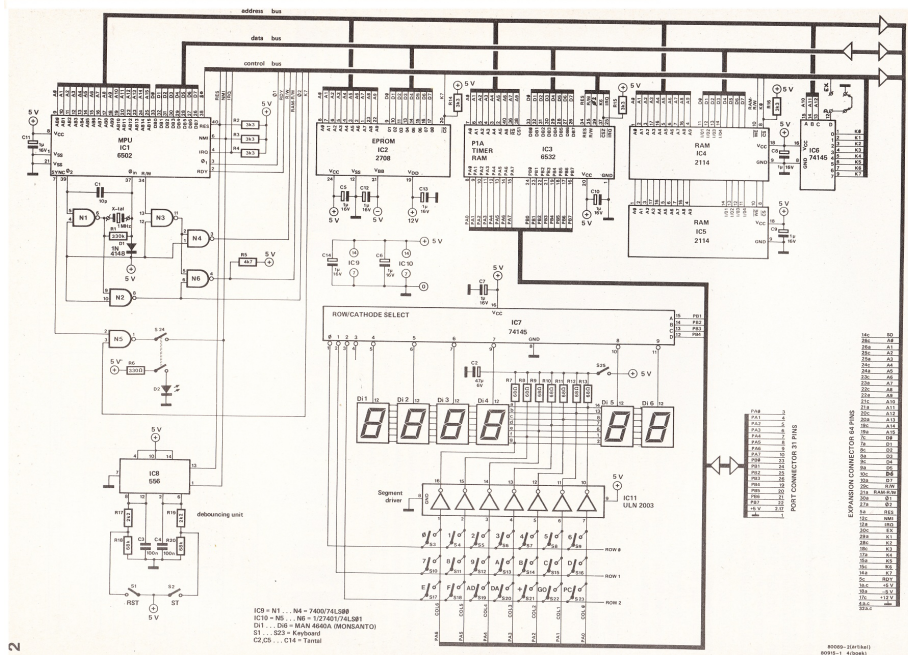
PIA Rockwell: R6532 (1x 8 bits timer 2x 8 bit bidirectional output's 128 bit static RAM)

display 6x 7-segments display.

( Tegenwoordig passen al deze eigenschappen ruim in de kleinste microcontroller )

Voor die tijd was dit een hele bijzondere computer waar ik vele uren mee geprogrammeerd heb. Later werd deze uitgebreid met veel extra geheugen (64kb), een VIA, RS232 interface, ADC, DAC en een tape recorder voor programma opslag. Het programmeren ging met de toetsen en zo werden byte voor byte de code ingevoerd. Dit moest dan van te voren wel eerst geschreven worden en uiteraard allemaal in assembler. Wel was er een soort editter aanwezig en hoefde de stapgroten bij sprongen niet zelf berekend te worden. En het mooie van dit systeem was dat het programma gedebuged kon worden on-board. Een eigenschap die ik bij de heden daagse microcontrollers soms erg mis. Vele uren heb ik geprogrammeerd en de eerste stappen van de computer geleerd. Bij de [AVR6440 computer](#) maak ik nog steeds gebruik van bepaalde routines, die ik toen geschreven heb.

Op een gegeven moment zat ik zo in deze programmeer taal dat ik de op-codes allemaal uit mijn hoofd wist en nu ken ik er zelfs nog een paar. '85 0A' LDA zeropage 0x0A, '8D 00 04' LDA adress 0400.



*Klik op het schema om deze te vergroten:*

Een paar jaar later heb ik deze computer afgeschreven en vervangen door een Accorn Atom computer. Wel met dezelfde processor, maar moderner en grafisch op een TV scherm. Een jaar later deed dan de eerste IBM compatible PC zijn intreden. een Schneider Amstrad. Eerst met 1 floppy diskdrive (180kb) en later uitgebreid met maar liefst een 20Mb harde schijf.



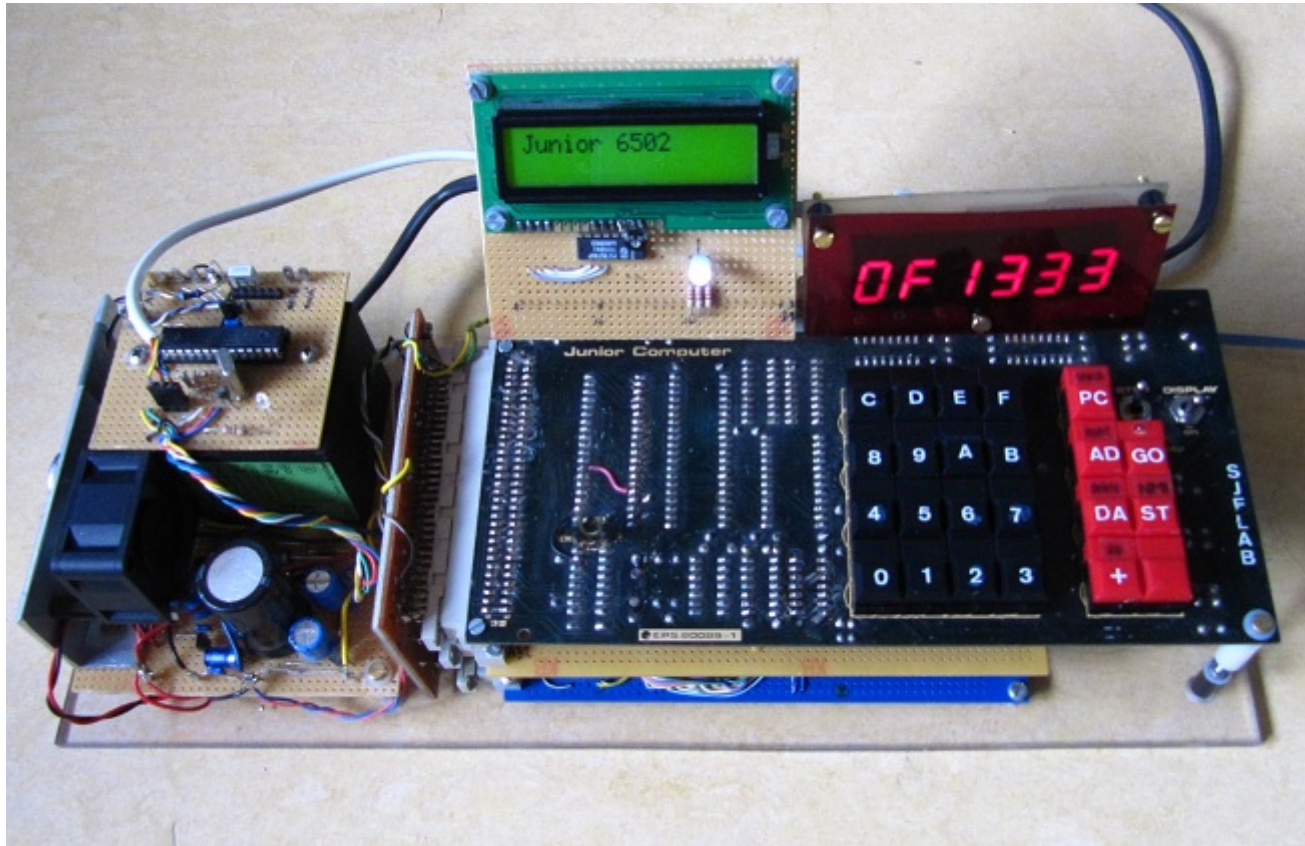
Ik weet niet hoe het bij de lezer is, maar ik kijk soms met heimwee terug naar die oude, soms zeer vernuftige, elektronica. Neem nou eens een schema van een oude radio en TV die slechts met een paar versterkers, buizen in die tijd, werd opgebouwd. Je kunt er soms uren ademloos naar zitten kijken hoe relatief simpel die apparaten in elkaar zaten, hoe begrijpelijk we het nu vinden, maar voor die tijd high tech. Nu ben ik echt niet van het type die beweert dat vroeguhhh alles beter was. Neem internet, wat heb ik vroeger een hoop moeite moeten doen om informatie te verkrijgen, elektuur, databoeken en de bibliotheek. Tegenwoordig is alles binnen een handomdraai op internet beschikbaar. Ben je niet thuis, dan wel via de smartphone. Hoe simpel is het nu voor de hobbyist om een schakeling op te bouwen met een microcontroller en niet eindeloze printen te moeten voorzien van TTL IC's en deze aan elkaar te koppelen. Ongekende mogelijkheden waarvan je vroeguhhh alleen maar van kon dromen.

Als je wat ouder wordt, de ambities bijstelt, is het soms erg leuk nog eens naar het oude terug te keren. Neem de Junior computer van mij. Deze heeft jaren, na enige jaren trouwe experimenteer en leer diensten, in een oude doos op zolder gelegen. Diep onder het stof, Afgeschreven, voorbij glorie tot het weer eens leuk is de computer aan te sluiten op een voedingsspanning. Zou hij het nog doen? Natuurlijk deed deze niets meer Enkele IC's waren verwijderd, blijkbaar uit nood om voor andere zeer nuttige taken te worden ingezet. Er waren wat modificaties aangebracht. Maar na een avondje puzzelen bleek ik de oude computer toch weer nieuw leven ingeblazen te hebben en de display's lichtten weer vertrouwd op. Diep gravend in mijn eigen geheugen kon toch weer een programmaatje worden geschreven. Daarna de computer voorzien van nieuwe druktoetsen, de interne contacten waren na 25 jaar erg geoxideerd en functioneerde niet goed meer. De

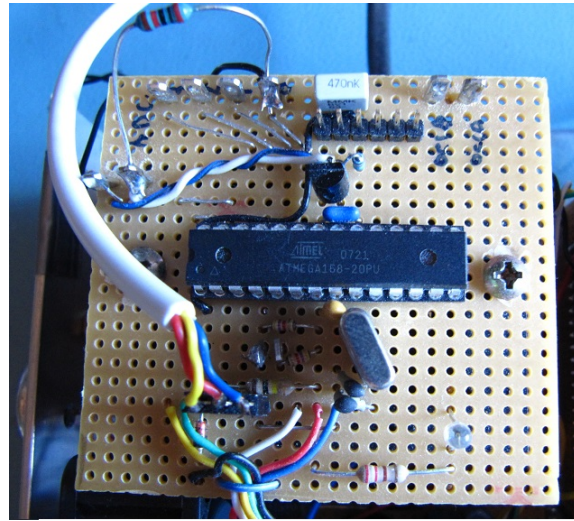
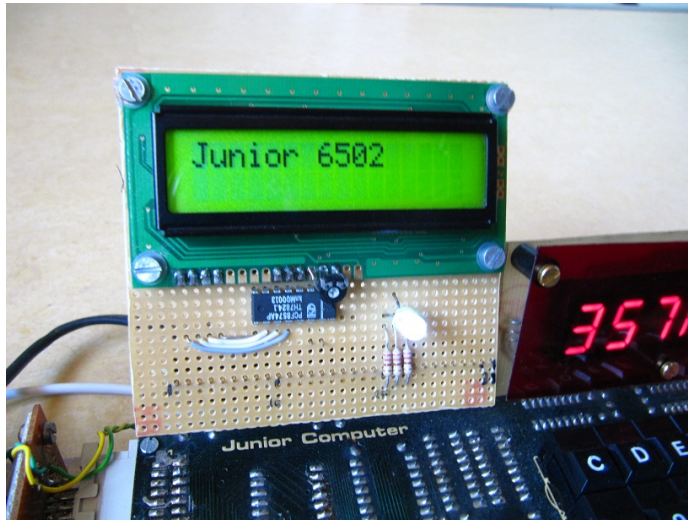
uitbreidingsconnectoren moesten ook worden vervangen en tot mijn verbazing nog gewoon leverbaar. De bejaarde computer mooi opgepoetst en een nieuw leven kon beginnen.

Goed de Junior-computer werkt dus weer, leuk! Maar ja kunnen we er nog iets mee? Allereerst toch maar de hulp ingeroepen van de moderne computer techniek. De basic versie van de junior heeft 1kb RAM geheugen, 1kb EPROM, daar beginnen we echt niets meer mee en het liefst wil ik de Junior nu zo origineel mogelijk houden. In het verleden zijn er talloze uitbreidingen geweest tot aan cassette interface's, toetsenborden en beeldschermen aan toe. In die tijd moest elk programma met de hand in getoetst worden en viel de spanning even weg, dan kon je weer van voren af aan beginnen. De latere uitbreiding met een cassette interface was een oplossing daarvoor. (beschreven volgens mij in juniorboek 3), maar ik heb geen cassette recorder meer. Om een programma in te voeren op de ouderwetse manier kan natuurlijk wel, maar dat ben je na één avond spelen wel zat. Ik heb daarom besloten om een 32k flash (AT28C256) en 32K RAM (62256) geheugen uitbreiding te gebruiken. Een 6551UART en een PCF8584 voor het aansturen van I2C. Omdat de Junior niet rechtstreeks een dergelijke flash kan programmeren, de adres en data lijnen moeten namelijk een tijdje stabiel zijn en dat is zondermeer niet mogelijk. Misschien dat ik daar later nog eens uit ga zoeken en dat het dan wel lukt, maar eigenlijk wil ik niet veel veranderen aan de nu originele Junior. Om de Flash toch te kunnen programmeren gebruik ik hiervoor een AVR om de Flash te schrijven, deze kan gewoon on-board blijven zitten. Alleen tijdens het schrijven kan de Junior de Flash niet uitlezen, maar dit is geen probleem. Voor uitbreiding naar analoge poorten, in en uitgangen timers etc gebruik ik een kleine AVR ATmega168 die via I2C kan worden aangestuurd. En natuurlijk kan een standaard 2x20 LCD display niet ontbreken. Die ook via I2C en een PCF8574 wordt aangestuurd.





**Compleet junior computer systeem nieuwe stijl.**



Display print

slave processor

De originele voedingsprint had ik niet meer en waarschijnlijk uit kosten overwegingen had ik de display print zelf geëetst. Echter de oude originele 5V stabilisator 'LM309K' vond ik nog in een bakje en heb ik maar weer ingezet. Ook kwam ik erachter dat de 6502 processor gerust op 2MHz kon lopen en dus het oude 1MHz kristal vervangen voor een 2MHz type.

Na wat experimentele programma's en heb opzetten van een programma op de PC om de Junior nieuwe stijl mee te programmeren, kwam achter een erg leuk idee. Ooit in een ver verleden (1976) is er een schaak programma '[microchess](#)' geschreven voor de [KIM-1](#). Zoals we gezien hebben een look a like van de Junior. Wat zou het leuk zijn om de junior computer eens te laten schaken.

Het programma is tamelijk populair en ook door andere programmeurs omgezet, o.a. voor de Arduino, zodat ook met een moderne AVR geschaakt kan worden. Om het voor de KIM-1 geschreven programma geschikt te maken voor de Junior zijn er maar een paar kleine wijzigingen nodig. De eerste kb RAM geheugen adressen 0000h tot 03FFh zijn hetzelfde. Niet zo verwonderlijk omdat hier het zeropage geheugen zit en de stack 1FFh .... 100h, uiteraard voor elke 6502 en equivalent hetzelfde. Omdat niet het gehele zeropage geheugen en stack nodig zijn, wordt dit gedeelte ook gebruikt voor het programma. Het eeprom geheugen met de vaste routines is wel anders dan de KIM en die aansturing kunnen we niet overnemen. Gelukkig heeft de Junior wel nagenoeg dezelfde routines die we kunnen gebruiken. Dit zijn de routine "SCAND" (zichtbaar maken van de inhoud van de F9h,FAh & FBh geheugen adressen op het display en detecteren van een ingedrukte toets) en "GETKEY" om de juiste toetswaarde op te halen. Verder zit er bij de KIM nog RAM geheugen in de 6530 I/O (1780h .... 17FFh) . De Junior beschikt over de PIA 6532 I/O TIMER RAM met ook 128b RAM geheugen op de plaatsen 1A00h ....1A7Fh. Geen probleem dus.

Er dienen dus een paar wijzigingen te worden door gevoerd:

Het volgende stukje programma wordt gebruikt om de inhoud van de DIS1(FBh), DIS2(FAh), DIS3(F9h) zichtbaar te maken en op een toetsenbord invoer te wachten.

### **OUT**

```
0008 20 1F 1F    JSR *OUT          DISPLAY AND
000B 20 6A 1F    JSR *GETKEY       GET INPUT
000E C5 F3      CMPZ .OLDKY        KEY IN ACC
0010 F0 F6      BEQ OUT            (DEBOUNCE)
0012 85 F3      STAZ .OLDKY
```

Dit stukje programma moeten we vervangen door het onderstaande het neemt 1 adres meer geheugen nodig, maar dit is geen probleem. Verder kan de variabele OLDKY op adres 00F3h komen te vervallen.

### **OUT:**

```
0008 20 8E 1D    JSR $1D8E  // SCAND: DISPLAY
000B D0 FB      BNE OUT    // Toets ingedrukt ?
```

### **OUT2:**

```
000D 20 8E 1D    JSR $1D8E  // SCAND: DISPLAY
0010 F0 FB      BEQ OUT2    // Toets ingedrukt ?
0012 20 F9 1D    JSR $1DF9  // GETKEY monitor programma
```

Hetzelfde komen we tegen op adres 039Fh. Ook hier bij de KIM een sprong naar JMP \*OU

### **RETP**

```
039F 4C 1F 1F    JMP *OUT          AND RTS
```

ook dit moet vervangen worden door:

```
039F 20 8E 1D    JSR $1D8E  // SCAND: DISPLAY
03A2 60          RTS
```



tot slot begint de routine "STRATGY" op adres [1780h](#) bij de KIM en moet dit [1A00h](#) worden bij de Junior.

En dat is alles. Dat moet lukken. Als we dit vervolgens in een editor programma zetten en vervolgens naar een compiler sturen, we krijgen een hex-file die we naar de RAM adressen van de Junior uploaden. Ik heb hier de [hex-file in Intel format](#). Het [geassembleerde programma](#) vind je hier. Mijn compiler vind niet alle label namen leuk die in het originele microchess programma worden gebruikt, zodat ik er een paar heb moeten aanpassen.

Ok! als we het programma hebben ge-upload gaan we naar het adres 0000h en druk op GO. Er lijkt even niets te gebeuren, maar als we op de C toets drukken verschijnen er 6 C's op het display. De computer is klaar voor een spelletje schaak. Drukken we vervolgens de PC toets in, dan wordt de eerste openingszet gedaan 0F1333 dat betekend pion voor de koning zet van veld 13 naar veld 33 anders gezegd voor de schakers: e2-e4.

Nadat ik een paar spelletjes gedaan heb, heb ik gezien dat het een simpel schaak programma is. Een fout detectie zit er niet in, verkeerde zetten b.v. gaat de computer gewoon mee door. Dus alles zelf goed in de gaten houden. Nadat je een zet ingevoerd hebt. b.v. 63 43 (e7--e5) niet vergeten op de 'F' toets drukken om de zet te bevestigen. Maar het wil niet zeggen dat je niet hoeft na te denken. De zetten die hij de computer doet zijn goed overwogen, maar echt veel zetten vooruit denken zit er niet in. Een beetje schaker kan makkelijk van de computer winnen. Maar wat wil je van een computer met 1kb geheugen, daar mag je nu eenmaal geen wonderen van verwachten. Veel lof voor de programmeurs in die tijd, want het is toch een alleraardigst schaakprogramma.



Bij vragen en opmerkingen kan je mij altijd emailen op het onderstaande email adres.

**e-mail:** [Sjaaks.eedoen](mailto:Sjaaks.eedoen@sjelab.nl) ( vervangt u AT voor @ )I

7 september 2014