

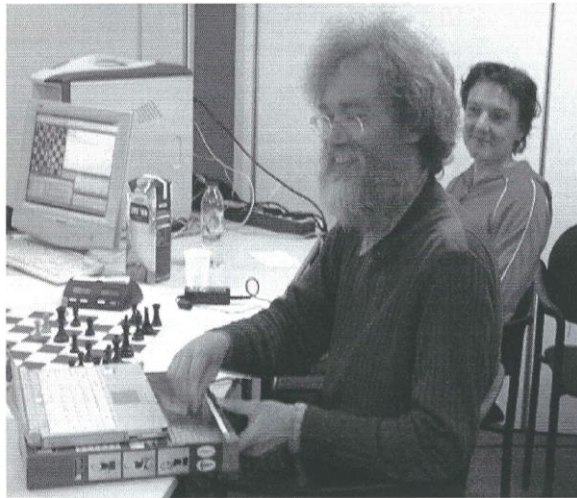
??-1977 [O-3901] Harm Geert Muller - **Usurpator** (Usurpator II)

Onlangs kreeg ik een kort berichtje van Luuk Hofman. Hij was op bezoek bij Hans Böhm en het gesprek ging al snel naar de geschiedenis van Harm Geert Muller's Usurpator. Op het internet staat alleen een 'tekening' van het befaamde luciferdoosje. Of ik meer gegevens had?

DE BELEVENISSEN VAN EEN HERINTREDER

USURPATOR op het **ONK** 2005

HARM GEERT
MULLER



Harm Geert Muller

In 1977, 2 jaar na mijn eerste kennismaking met computers als 1ejaars wiskundestudent, schreef ik mijn eerste echte schaakprogramma, de Usurpator I. Rond die tijd bereikten de 8-bit microprocessors de markt, d.w.z. de single-chip versies ervan zoals de 6800 en 8080. Daarvoor had de simpelste CPU ongeveer de afmeting van een hutkoffer, geheel volgestouwd met printplaten. En dat was dan alleen de CPU, voor het geheugen had je rekken zo groot als garderobekasten, gevuld met magnetische ringetjes. Maar in 1977 waren de integratiedichtheden zo toegenomen dat je met een paar dozijn chips een computer kon bouwen. Zonder beeldscherm en schijven, natuurlijk: programma's bewaarde je door ze op te nemen op een audiocassette, en de in- en uitvoer ging via een rekenmachine-achtig toetsenbordje/display. En als je bereid was het zaakje zelf in elkaar te solderen, dan werd het betaalbaar voor de amateur. De prijs lag weliswaar nog ver boven het budget van een student als ik, maar omdat een middelbare-schoolvriend van mij een echte baan had, en ik verstand van elektronica, konden we door onze krachten te bundelen zo'n computer in elkaar zetten. Dit vooral ten behoeve van de besturing van zijn modelspoorbaan.

Toen die computer, voor het oplossen van een hardwarestoring, een maandje bij mij thuis kwam te staan, schreef ik er een schaakprogramma voor, de Usurpator I. Dat moest als een reeks hexadecimale getallen met de hand in het geheugen worden ingevoerd, want assemblers en compilers hadden we niet. Maar omdat het hele geheugen sowieso maar 2 Kilobyte groot was (wat evengoed nog een gehele printplaat zo groot als een pocketboek in beslag nam) was ook dat leed te overzien. Rokeren en e.p. slaan kon hij niet, maar mat in één (eigenlijk: koning slaan in twee) zag hij nog net wel. Als gevolg van een 'piece-square' tabel die voor alle stukken gold vond hij dat een koning in de hoek heel slecht stond, en door ze tegenstander in die positie te manoeuvreren kwam het mat in een KQK eindspel vanzelf binnen beeld. Dat maakte dat er daadwerkelijk mensen waren waar hij vrij gemakkelijk van won!

Twee jaar later kocht ik zelf een computer, met de luxe van een assembler in ROM en 4Kbyte geheugen, gebaseerd op de (nieuwere) 6502 CPU (voor de kenners: een AIM 65). Ik gebruikte hem voornamelijk voor het verder ontwikkelen van dat schaakprogramma-

ma, dat ik eerst in 6502 assembly code vertaald had. Zo ontstond de Usurpator II. Grootste verbetering t.o.v. de Usurpator I was dat hij zich van een soort null-move bediende (ik noemde dat een pas-zet) om aan het eind van de zoekboom ook alle zetten van de partij die daar niet aan zet was te bekijken. Dit om te zien of de laatste zet van de tegenstander een dreiging inhield waarop gereageerd moest worden. Als zulks het geval was dan verdiepte hij de zoekboom selectief met een extra ply (nu zou je dat een 'extensie' noemen), voor die zetten waarvan hij meende dat ze wel eens tegen de dreiging konden helpen. (Aanvankelijk alleen het wegvlugten met het bedreigde stuk of de aanval blokkeren door er iets tussen te zetten).

Usurpator stond sowieso bol van de extensies, elke slagzet die op de vorige ply nog niet mogelijk was (zoals terugslaan, maar ook het afmaken van een aftrekdreiging of het afstraffen van een weggezet gepend stuk) leidde tot een extensie. Je moest eigenlijk wel, want een volledige zoekboom kwam niet dieper dan ongeveer 4 ply. Dus om nog in enige mate combinaties te kunnen doorzien moest je eigenlijk alle zetten die schaaktechnisch wel eens relevant zouden kunnen zijn niet voor een volle ply in de diepte meetellen. Bij 9 ply hield het trouwens sowieso op, omdat dan het einde van het geheugen bereikt werd.

Met Usurpator II heb ik, van 1981-1987, enige malen aan het NK meegedaan. Aanvankelijk ging dat niet eens zo slecht. Dat waren wel hectische tijden. Op het 1e NK moest je drie partijen tegelijk spelen, omdat de organisatoren op het onzalige idee gekomen waren de bedenktijd tot 4 min per speler per zet op te voeren. Men was bang uitgelachen te worden door serieuze schakers vanwege het vertoonde spelpeil, en leefde in de droom dat het ver-drievoudigen van de bedenktijd een wezenlijke verbetering zou opleveren. Omdat ik een nogal buitenissig model computer had, had ik grote moeite om aan 3 systemen te komen, maar uiteindelijk mochten we twee compatibele Siemens systemen van een computerzaak lenen (de 2 die ze op voorraad hadden).

Op het 2e NK werden nog steeds 3 partijen tegelijk gespeeld. Omdat ik toen helemaal niet aan andere computers kon komen, had ik een multi-tasking besturingssysteemje geschreven dat door het intoetsen van c1 A, B of C het toetsenbord en display aan de

Harm Geert Muller: De belevenissen van een herintreder... Usurpator op het ONK 2005

Bron: CSVN Computerschaak – februari 2006
(photo copyright © by <http://www.schaakcomputers.nl/>)

gewenste partij koppelde. Dat ging natuurlijk ernstig ten koste van de bedenktijd die voor elke partij beschikbaar was, en om het organisatie-committee daar zo hard mogelijk mee te confronteren had ik het geheel zo ingericht dat de 3 partijen verschillende prioriteit hadden, als A dacht werden B en C er volledig uitgedrukt. Door met de hand de zetten van de tegenpartij op het juiste moment in te voeren kon ik het zo regelen dat B dacht in de tijd dat A op de zet van de tegenstander wachtte, en partij C (tegen de sterkste tegenstander, natuurlijk) gaf ik zo zonder meer gewonnen omdat die eigenlijk nooit an de beurt kwam en na 5 zetten door zijn vlag ging. Dat werkte nogal competitie-ervalsend, en door bij storingen e.d., die bij alle deelnemers nog veelvuldig optraden veel stampij te maken wist ik er goed de aandacht op te vestigen. Als beloning kreeg ik na afloop nog een troostprijs ook, omdat men meende dat ik zoveel pech gehad had, en niet doorzag dat het boze opzet was! Maar hoe het ook zij, het was het laatste jaar van de simultaan toernooien, en ik hoop dat mijn wangedrag aan die beslissing heeft bijgedragen...

"Het luciferdoosje schaakte als brandhout!"

Na 1983 moest ik met mijn tijdroevende hobby van schaakprogrammeur stoppen, omdat je bij een promotie-onderzoek nu eenmaal niet zo kunt freewheelen als een student dat kan. In 1985 werd ik nog eens ter gelegenheid van het 5-jarig jubileum van het NK uitgenodigd om mee te doen. Hoewel ik het programma niet meer verder ontwikkelde, bleef ik ook daarna voor de gezelligheid meedoen. Maar omdat de 6502 computer waarop het zaakje draaide op een gegeven moment verloren ging, en de technologie inmiddels zo ver gevorderd was dat een computer met 4K RAM in essentie met twee chips gebouwd kon worden (een CPU en geheugen chip) besloot ik, speciaal om toch aan het NK mee te kunnen doen, nieuwe hardware te bouwen. De sport werd om het zo klein mogelijk te maken, en uiteindelijk kon ik het geheel in een luciferdoosje passes. De eerder genoemde twee chips, een paar horloge batterijtjes om het geheugen op spanning te houden zodat het programma erin bewaard bleef, nog twee support chips voor het opwekken van de klok en de uitvoer. Die laatste liep via een LED displayje van twee cijfers dat ik uit een rekenmachientje gesloopt had, en die afwisselend het begin en eindveld van de zet toonden. Een zet invoeren ging via een enkele tiptoets (twee nietjes die je met je vinger moest verbinden).

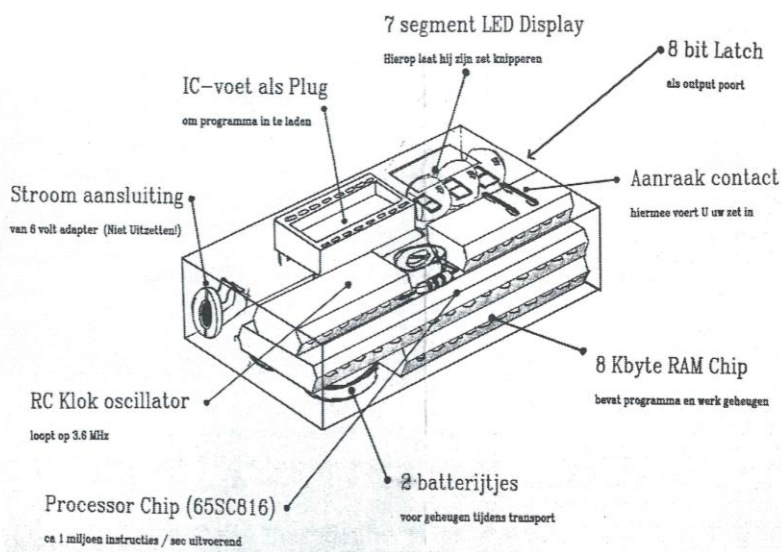
Dolle pret om met dat luciferdoosje op het NK'87 te verschijnen, waar de meeste mensen nog karrevrachten apparatuur moesten installeren. Ik haalde het 1 minuut voor de wedstrijd begon gewoon uit mijn vestjeszak, als de tegenstander zich al verbaasd begon af te vragen of er wel iemand zou komen opdagen. Alleen de tweede ronde verging het lachen mij: op mijn (vouw)fietsstocht van het station naar het CRI in Leiden moest ik een regenbui trotseren, en daardoor was het luciferdoosje enigzins vochtig geworden. De tiptoets was zo gevoelig dat hij dardoor meende dat hij voortdurend werd ingedrukt, hetgeen het reset-commando was dat stukken telkens in de beginstand terugzette. Pas nadat de wedstrijdleiding een föhn tevoorschijn had weten te toveren kon het luciferdoosje weer kurkdroog geblazen worden, en kon er gespeeld worden!

Wegens het daverend succes dat ik met deze show oogste (niet schaaktechnisch helaas...) werd ik, na twee jaar afwezigheid, op het jubileumtoernooi in 1990 nogmaals gevraagd. Omdat ik het schaakprogramma helemaal niet verder ontwikkelde, en de rest van de wereld natuurlijk niet stilstand, werd ik er kansloos afgehakt. Maar ja, wat kan je ook verwachten van een luciferdoosje? Het schaakte als brandhout.. September j.l. ontving ik een volkomen on-

Nou ja, meedoen is belangrijker dan het winnen, en het leek me leuk om weer eens te zien of het nog steeds zo gezellig toegang als vroeger.

Probleem was alleen dat ik het luciferdoosje niet meer kon vinden (hoewel ik het een paar jaar geleden nog een keer tegenkwam op mijn rommelzolder), en dat ik bovendien behoorlijk zeker wist dat ik het niet meer eenvoudig aan de praat zou kunnen krijgen. De batterijtjes zouden na 15 jaar heus wel leeg zijn, en van de zelfbouwcomputer via welk ik het programma weer in het doosje zou kunnen down-loaden wist ik dat de geheugenkaart kapot was. De broncode van Usurpator stond op een schijfje dat in geen enkele bestaande andere computer meer past (3 inch...) en bovendien niet-MS-DOS compatible geformateerd was. Een behoorlijk hopeloze zaak, en dat alles voor een computer van 4MHz met 8K RAM (zij het een erg ludieke).

Hoewel de amusementswaarde van het luciferdoosje natuurlijk niet te overtreffen was, besloot ik vanwege al deze problemen dat het toch gemakkelijker en lonender zou zijn om het programma te herschrijven voor moderne hardware, dan om te proberen de oude hardware te doen herleven. Dan had ik in elk geval het voordeel van 10.000 maal



verwacht telefoontje van de CSVN, waarmee ze mij uitnodigden om mee te doen aan het 25e (inmiddels open) NK. Toen ik omzichtig uit begon te leggen dat een programma, waar voor het laatst in 1983 iets aan veranderd was, misschien niet erg goed meer zou presteren, werd dit niet als een probleem gezien, omdat ook andere oud-deelnemers zouden worden uitgenodigd.

snellere hardware met 100.000 maal zoveel geheugen. Het algoritme van Usurpator stond me nog duidelijk genoeg voor de geest, en de details van de evaluatie-functie kon ik nog uit een boek halen waarin de broncode van Usurpator ooit was uitgegeven. In een hogere programmeertaal (C) moest dat een fluitje van een cent zijn vergeleken met dat gepriegel in assembler. Bovendien

Harm Geert Muller: De belevenissen van een herintreder... Usurpator op het ONK 2005

Bron: CSVN Computerschaak – februari 2006
(photo copyright © by <http://www.schaakcomputers.nl/>)

kon ik, door de opzet iets te veranderen (niet alleen een schaakbord bijhouden, maar ook een lijst met stukken zodat het programma snel kan uitvissen welk stuk waar staat), meteen een paar mij bekende tekortkomingen van het oude programma repareren. Met name had het mij altijd gestoord dat Usurpator vroeger nooit in de gaten had wat door wie gedekt werd, en aan het eind van zijn zoekboom onbezonnen stukken wegzetten die essentieel waren voor het dekken van iets anders, zonder dat hij een strafstelling van deze spellijn voorzag. Het leek allemaal wel te doen.

Nou, ik heb het geweten! Een maand is toch wel een heel korte tijd om een programma uit de grond te stampen. Zeker omdat ik het ook nog wilde voorzien van een hash-tabel, iets waar op een 4KB machine natuurlijk nooit ruimte voor geweest was. Je hebt die schier eindeloze hoeveelheid geheugen toch niet voor niets! Maar goed, ik had dus op basis van het oude Usurpator algoritme iets uit de grond gestampt wat schaakzetten produceerde, en hoopte dat het door de 10.000x hogere CPU snelheid ook wel een stuk beter zou schaken. Die snelheid was al gauw goed voor 4 extra ply's aan zoekdiepte. Getest was het nauwelijks, op een laag niveau een paar vluggerjes ertegen gespeeld om te checken of hij geen erg rare of onreglementaire zetten deed, en dat leek allemaal goed te gaan. Dus vol goede moed op weg naar het ONK, met in de trein nog net gelegenheid om de tijdmanagement in te programmeren, zodat hij in elk geval niet eindeloos over een zet zou kunnen blijven denken (zoals hij in 1981 nog wel eens wilde doen).

Nou, dat werd dus een ramp. Om te beginnen was ik de enige oud-deelnemer die daadwerkelijk de onbezonnenheid had gehad zich in te schrijven. Dus verder alleen maar tegenstanders van meester en grootmeesterniveau. Verder bleek het oude algoritme verbazingwekkend slecht te reageren op de verruimde bedenktijd (gemeten in CPU clocks). Er waren gewoon teveel schaaktechnische situaties waarin het programma zichzelf een extensie gaf, mede omdat ik er vanwege dat dekken nog een paar aan het repertoire had toegevoegd (zoals het slaan van een stuk waarvan de dekker op de voorgaande ply werd geslagen of weggezet). Bij geringe doordenkdiepte was dat nog geen probleem, maar bij de grotere diepte bereikbaar vanwege de supersnelle CPU bleek de fractie van takken in de boom die zichzelf eindeloos met extensies verlengde te exploderen. Vroeger, toen de diepte die je kon bereiken eigenlijk te klein was, moest je wel agressief extensies

gebruiken om nog wat combinaties te doorzien. Maar bij de snelheden van vandaag keek het programma daardoor veel te ver vooruit. Vooral als het onder druk gezet werd, zoals de tegenstanders natuurlijk genadeloos deden! Usurpator besteedde zoveel tijd aan het denken over wat er tussen ply 15 en 30 zou kunnen gebeuren, dat hij geen tijd meer overhield om grondig uit te zoeken wat er eigenlijk tussen ply 5 en 10 aan de hand was. Het pijnlijkt bleek dit in de openingswedstrijd tegen Chessmaster, toen op een gegeven moment de druk zo hoog werd dat de gehele bedenktijd nog niet genoeg was om het onderzoek van de null-move met een nominale diepte (zonder extensies) van 1 ply te voltooien, zodat in essentie een random zet werd afgegeven.

Daarnaast waren er natuurlijk de wat subtielere bugs, die je er niet gemakkelijk uit test en die er een maandje na de geboorte dus nog volop in zaten. Of die je er tijdens het toernooi nog even aan toevoegt door iets te 'verbeteren' i.v.m. in de vorige ronde gebleken fouten. Zo had ik bij de iteratieve verdieping ingeprogrammeerd dat er niet verdiept hoeft te worden als je maar 1 reglementaire zet hebt. Maar door een accoulade op de verkeerde plaatst gebeurde dat niet alleen op ply 1, maar ook dieper in de boom. En als je schaak gezet wordt, en maar op 1 manier weg kunt, dan is het toch wel erg handig als je weet wat er daarna gebeurt. Maar Usurpator was er dus blind voor, na uit het schaak gestapt te zijn hield de wereld voor hem op te bestaan. Een andere fout die ik er voor de 6e ronde nog even in bracht betrof het bepalen van de begindiepte voor het iteratief verdiepen. Omdat Usurpator met fractionele ply's werkt, en bij elke iteratie 1 hele ply verder vooruit denkt, moet hij bepalen met welke diepte hij moet beginnen om precies op het gewenste aantal ply's te eindigen. Door de verkeerde kant op af te ronden werd 0,5 'afgerond' naar 1,5, en gaf zichzelf er tot in het onelindige nog een onbedoelde extensie bij als het aantal resterende ply's toevallig precies op een half uitkwam.

Alsof dat alles nog niet erg genoeg was, werd ik ook nog getroffen door hardware problemen. Mijn laptop, die ik ook pas een paar maanden had, bleek met een kapotte ventilator te kampen. Als ik er gewoon teksten mee zat te verwerken werd hij al zo heet, maar als de CPU bijna voortdurend op volle sneheid moest werken raakte hij echt oververhit en schakelde zichzelf uit. De eerste dag kon ik alleen uitspelen door voortdurend in het ventilatiegat te blijven blazen. De volgende dag had ik hier een tot blaasbalg omgebouwde kartonnen doos voor

gefabriekt. Dat was een lapmiddel (maar ik kon de laptop met geen mogelijkheid open krijgen, en durfde het niet aan een alles-of-niets poging te wagen). Volgens mij veroorzaakte de oververhitting ondanks die koelpogingen nog steeds hardwarefouten, want in sommige partijen begon het programma gedrag te vertonen dat bij een herstart helemaal niet meer te reproduceren viel (het halstarrig weigeren om aangevallen stukken weg te zetten). In een andere partij merkte ik aan de output dat hij opeens was opgehouden de hash tabel te gebruiken (iets waarvoor ik normaal voor moet hercompileren).

Maar goed, meedoen was belangrijker als winnen, en ik heb met veel plezier meegeedaan. Alleen jammer dat door alle problemen het programma niet heeft kunnen brengen wat er met wat betere tuning en goed functionerende hardware mijns inziens toch wel in zat. Ik hoop daarom nog minstens 1 keer mee te doen met ditzelfde programma, maar dan zonder fouten, zodat het werkt zoals het bedoeld was. Dan verliest het waarschijnlijk ook wel alles, maar dan biedt tenminste een beetje tegenstand. In elk geval heb ik de smaak weer helemaal te pakken, en ik ben alvast begonnen aan het ontwerpen van een nieuwe engine, volgens principes die beter aansluiten bij de mogelijkheden die de huidige hardware biedt.

Een paar partijen van Usurpator op het NK:

The King - Usurpator [A36]
Leiden ONK 2005

```
1.c4 c5 2.Pc3 Pc6 3.g3 d6 4.Pf3 e5 5.d3
Lg4 6.Lg2 Lxf3 7.Lxf3 Pge7 8.0-0 h6
9.Lg2 Dd7 10.a3 Pd4 11.e3 Pe6 12.b4
a5 13.bxc5 Pxc5 14.d4 exd4 15.exd4
Pe6 16.Tb1 Tb8 17.Te1 g5 18.Pd5 Kd8
19.Lh3 Pg6 20.Ld2 Ta8 21.Pb6 Dc6
22.d5 Dxb6 23.Txb6 Pc5 24.Txb7 Pxb7
25.Da4 Pe5 26.Txe5 dxe5 27.Dd7# 1-0
```

Tao - Usurpator [B45]
Leiden ONK 2005

```
1.e4 e6 2.d4 c5 3.Pf3 cxd4 4.Pxd4 Pc6
5.Pc3 Da5 6.Pdb5 a6 7.Pd6+ Lxd6
8.Dxd6 Pf6 9.Ld3 h5 10.0-0 g5 11.Ld2
Tb8 12.Pd5 Dxd2 13.Tfd1 Dxd1+
14.Txd1 Ph7 15.c4 f6 16.Pb6 a5 17.e5
f5 18.Lxf5 exf5 19.Dg6+ Kd8 20.Dg7
Pf8 21.Dxh8 Ke7 22.Dg7+ Ke8 23.Pd5
Ph7 24.e6 dxe6 25.Pc7# 1-0 ●
```

Harm Geert Muller: De belevissen van een herintreder...

Usurpator op het ONK 2005

Bron: CSVN Computerschaak – februari 2006
(photo copyright © by <http://www.schaakcomputers.nl/>)

**“Ook
Usurpator
heeft aan een
maand niet
genoeg tijd
een
programma te
ontwikkelen”.**

A MATCHBOX CHESS COMPUTER

Harm Geert Muller

Amsterdam, The Netherlands

In 1985 I was invited by the Computer-Chess Association of the Netherlands (CSVN) to compete in their fifth national championship with my old chess programme "USURPATOR II". I had enlisted in that tournament some years before, but had long since lost the 6502-based computer (a Rockwell AIM '65) on which I had developed the programme. It would have been hard to find a similar system that had its operating speed boosted by a factor of two, like the one I had owned. Furthermore I had very unfavourable experiences with transferring computer programmes stored on audio tape to other computers with incompatible tape formats, and I was sure the only tape I had left containing the source programme was incompatible with anything else, because I had designed the tape format myself to circumvent the slow and error-prone cassette interface of the AIM '65.

I therefore decided to build a new computer for the occasion, one that would run as fast as I could make it. I did not want to spend too much time on this project, and so I wanted it to contain as few parts as possible. Since USURPATOR II consisted of only 4 Kbyte of code and used about 1K for data storage, a single 8K memory chip would be sufficient to hold both the programme and the data. I did not want to store the programme in EPROM, because in those days EPROMS were fairly slow memory devices. Therefore I decided that the new computer should consist of a fast 65C02 CPU chip, an 8Kbyte static memory (RAM) chip and whatever was needed to support those.

After some thinking it became clear that the volume of the components would hardly total more than the volume of an ordinary-size matchbox. It next became a challenge to squeeze the entire engine indeed into such a matchbox. Since in those days the technology of 'surface mounted devices' was not very developed yet, the use of chips in common 'dual in line' packages was mandatory. Especially the CPU chip wasted a lot of space because of its bulky plastic encapsulation. I had even to file a few millimeters off the ends to make it fit at all.

Cannibalizing a Calculator

Input and output is of course a problem in a computer the size of a matchbox. Of course I could have equipped the matchbox with a single LED that would flash the selected move in Morse code. Since I don't know Morse code at all, this idea seemed too cumbersome. The next best thing I could think of was a very tiny seven-segment LED display like those used in old pocket calculators. Liquid crystal displays used in watches would be smaller still, but after demolishing a watch I learned that every segment in such a display has to be driven separately, thus requiring a prohibitively large number of outputs on the computer. So I sacrificed my old pocket calculator to obtain a three-digit LED display.

This display required only seven output lines to drive the segments, while output line number eight drove a transistor that selected the digit of which these segments were on. The third digit was not used, and the move could be displayed by alternating the algebraic notation for the original and the target square in the two available digits. As is common in displays of this type the impression is created that the two digits needed to represent a square are lit simultaneously by switching 100 times a second between displaying the letter in the left position and the digit in the right position. The display could thus be controlled by an eight-bit output port, a function that could be fulfilled by a single chip. Displaying something requires the continuous attention of the computer, and has to be stopped while the chess program is thinking.

The input problem was much harder to solve than the output one. Even the smallest switch I could buy in my local electronics store would take up one quarter of the matchbox. Touch buttons could be made smaller than that, so I opted for those. The question was how many would be needed to operate the machine. Since it was clear that I did not have space to assign a key to each of the letters A to H and digits 1 to 8, some kind of sequence would be needed to enter the moves. If a sequence was needed anyway, multiple keys would only serve to confuse matters, so I decided to have just a single touch contact on the top surface of the matchbox. It consisted of two staples put into the cardboard, that could be connected electrically by putting one's finger on top of them.

Harm Geert Muller: A Matchbox Chess Computer

Source: ICCA Journal – Volume 13 December 1990
(photo copyright © by <http://www.schaakcomputers.nl/>)

What Happens to Stapled Input

To include a separate chip for just a single input line seemed wasteful, and therefore the touch contact was connected directly to the interrupt line of the CPU chip. This causes an interrupt each time the contact is touched, and software can determine how much time elapses between interrupts, and so decode the meaning of the sequence. It later turned out that the touch contact did not operate very well in a dry environment, and an extra transistor was added to amplify the signal. The drawback is that this made it perform poorly in a wet environment, so that one day after biking through the rain with the matchbox in my pocket I had to use a hairdryer before I could play. But I knew you couldn't win them all.

Eventually the design contained only five chips (CPU, memory, output, and two chips with gates used for miscellaneous functions), a display, two transistors, two tiny batteries, and a 16 pin IC socket. The socket can be used to plug in a cable connecting the matchbox to a personal computer, that then downloads the memory chip with software. Virtually no space is left inside.

Entry Just in Time

The entering of moves through the single contact seems very awkward at first, but after a dozen moves or so becomes quite natural. One simply taps on the touch contact a number of times in rapid succession, once for A or 1, twice for B or 2, etcetera. Between the letter and the digit one simply waits somewhat longer. To aid the operator, the character being entered is displayed, so after touching the contact once an A appears in the display, and if we touch again before the A disappears (after 300 ms) it changes into a B, and so on up to H after which it wraps around back to A. If the desired character is in the display we wait until it disappears, and start tapping again for the next character. Of course the computer now knows that we are entering a digit, so it counts 1, 2, 3, ..., up to 8. If we do not touch the contact for longer than a second without having entered four characters in this way, the move is void and we have to start from scratch. The computer then goes back to displaying whatever it was displaying before we started touching it. This seems a harsh way to treat the operator, but is important to prevent an accidental touch from erasing the move selected by the computer from the display.

If four characters are entered this way (requiring at most 31 touches, e.g. for h7-h8) the move is flashed in the display for three seconds, for final scrutiny on the part of the operator. If the move turns out to be in error after all, starting the entry of a new move within this period will overwrite the old one. After the three seconds have elapsed, the display is erased, the move is tested for legality, and depending on this either a prompt sign or an error symbol is displayed.

USURPATOR never starts thinking of its own accord. To have him play a move one has to press the contact for one steady second. The A in the display then changes into Ξ , and if we release the contact at that moment, USURPATOR starts thinking for the player whose turn it is to move. This procedure makes it possible to enter a sequence of moves to set up a game, play a game with either Black or White against the computer, or have the computer play itself, without the need for any other commands. One can even change sides halfway a game.

Don't Watch Us, We'll Beep You

After a few matches in the championship it became clear that there were two major shortcomings to this user interface. The first was that it required constant attention to know when USURPATOR made his move. One had to watch the tiny display like a hawk to avoid losing time. This was solved by including a beeper disk from a watch into the design. This very flat disk could be fitted between the drawer and the wrapping of the matchbox, at the bottom. It was connected to an unused address line of the CPU, so that the programme could change the voltage on the disks by jumping back and forth between two memory addresses for which this address bit differed. By doing this in rapid succession an audible beep is created.

The other problem was that it was very cumbersome to set up a previous position if an entry error had occurred. The rules of the tournament are such that if one of the operators made an entry error that went unnoticed for some time, the game should be restarted at the position just before the error was made. The only way to do this was to interrupt the power to reset the matchbox, and then enter all the moves played so far. To facilitate this, the matchbox now stores all the moves played so far. If the contact is touched extremely long, the display starts counting up from zero while the touch lasts. If the contact is released while the count reaches n , the game is restored to the situation after move number n . This also makes it easy to start a new game, by simply releasing the contact when the zero appears.

Harm Geert Muller: A Matchbox Chess Computer

Source: ICCA Journal – Volume 13 December 1990
(photo copyright © by <http://www.schaakcomputers.nl/>)

The least one can say about USURPATOR II, playing from its matchbox, is that the program is lightning fit. Not quite so fighting fit because it is not a very strong programme. The reason for this is that it started as an experiment, and this experiment was never really completed. USURPATOR relies very heavily on a Shannon-B strategy. To decide which moves are considered while searching the game tree, it distinguishes several levels of plausibility for moves. This plausibility does not in any way depend on an estimated value of the move, (which is determined by the tree search), but rather on how well the move logically follows from the previous moves.

For instance, if a piece moves to a certain square, it is logical to try and capture it on this new square. This does not mean that it is always good to capture it, but we can never know that before we try. Other logical continuations would be to try and capture something with a move that passes over the square just emptied (after all the piece might have been pinned down), or moves that save a piece that was attacked at the last ply.

The Penalties for Off-side

Moves of this type are always considered by USURPATOR, one could think of them as being part of an elaborate quiescence search. If they occur early in the tree they are not counted towards tree depth either. Only on the first two plies is a Shannon-A strategy used, because the programme is not supposed always to remember what went on before the current position was reached. The first two plies thus act as initiators of a plan, and after that only moves consistent with the plan are considered, on the assumption that all other moves are irrelevant to the point of nonsense.

In some situations USURPATOR's strategy works extremely well, but in others it fails miserably. The reason is that the strategy outlined above is not really implemented very consistently. For instance if Pawns are attacked this is considered such a minor threat that moves that could save them are not considered logical. The most severe shortcoming is that USURPATOR has only a very approximate notion of how to defend its pieces. It considers defending a valid way to fence off an attack from the previous ply, but if a move leaves certain pieces undefended this goes completely unnoticed. Furthermore it is easy to throw USURPATOR's analysis off track by an intervening exchange, after which it has forgotten all about its original plan.

Some of these problems could be fixed easily, but for solving others a drastic overhaul of the entire programme is desirable. I have therefore decided to write a new programme from scratch, and I might as well do this for another machine, since the matchbox computer is an environment that is not well adapted for programme development. I now use the matchbox only to entertain the spectators on special occasions, such as the tenth re-enactment of the Dutch National Computer-Chess Championship. Although the programme has not been changed in 7 years, sometimes it still surprises me by actually winning a game. Among its major achievements are victories over an 80386 and an 80286 machine in 1990, and a draw against a Gould computer in 1985.



Collection of Photos by Jos Uiterwijk



FIREPOWER FOR A FULL BOARD. For author Muller a matchbox packs enough kindling to set a match abl

Harm Geert Muller: A Matchbox Chess Computer

Source: ICCA Journal – Volume 13 December 1990
(photo copyright © by <http://www.schaakcomputers.nl/>)