MICROCHESS

A CHESS PLAYING PROGRAM FOR THE 8080 MICROCOMPUTER

Written By: P. Jennings & T. O'Brien

MICRO-WARE LTD.
MICROCHESS was originally conceived as a program which would play chess using only a minimum hobbyist microcomputer system. The program which was developed will run on any 8080 based microcomputer configured with at least 4K of contiguous RAM, and an ASCII input/output device.

Although MICROCHESS does not always play chess at the expert level, it will make a reasonable move under most circumstances. In addition to being great fun to play, it can provide a useful and tireless opponent for practising checkmates, learning openings, and sharpening general playing skills.

The MICROCHESS program is supplied on paper tape or on Tarbell(TM) cassette. The documentation provided includes complete player's instructions, a description of the program operation, and an appendix with details for modifying the I/O to suit the individual requirements of each user's personal computer system. If you should have any problems putting MICROCHESS up on your system, please send the details of your system and the exact problem to the address below. We will do our best to assist you in any way possible.

© This copy of the MICROCHESS program and documentation is provided for the personal use and enjoyment of the purchaser. Reproduction by any means is prohibited. Use of the MICROCHESS programs, or any part thereof, for the purpose of promotion or sale of microcomputer hardware or software, without the express written permission of the authors is prohibited. Address all communications to:

MICROCHESS,
MICRO-WARE LIMITED,
27 FIRSTBROOKE RD.,
TORONTO, Ont.,
M4E 2L2,
CANADA
# TABLE OF CONTENTS

## PLAYER'S MANUAL
- NOTATION ................................................. 3
- PROGRAM EXECUTION ................................. 4
- MICROCHESS COMMANDS .............................. 4
- THE DISPLAY COMMAND .............................. 5
- THE GO COMMAND ..................................... 5
- THE SPEED COMMAND ................................. 6
- THE RESIGN COMMAND ............................... 6
- THE EXCHANGE COMMAND ............................. 6
- ENTERING YOUR MOVE ................................ 7
- SPECIAL MOVES ....................................... 7
- CASTLING .............................................. 8
- PAWN PROMOTION ..................................... 8
- EN PASSANT ........................................... 8
- THE COMPUTER MOVE ................................. 9
- NOTES .................................................. 9

## APPENDICES
- APPENDIX A
  - THE PROGRAM ..................................... 11
  - THE CONTROL AND INPUT/OUTPUT .................. 11
  - MOVE GENERATION .................................. 11
  - DATA COLLECTION .................................. 12
  - STRATEGY ......................................... 12
- APPENDIX B
  - INPUT AND OUTPUT SUBROUTINES .................... 14
- APPENDIX C
  - DISPLAY OPTIONS ................................... 15
  - CRT DISPLAY ....................................... 15
  - CUSTOM BOARD DISPLAY ............................. 15
  - PIECE ADDRESSES FOR BOARD DISPLAY ............... 16
- APPENDIX D
  - RETURN TO YOUR OPERATING SYSTEM ................. 17
- APPENDIX E
  - HEX DUMP OF MICROCHESS ......................... 18
- APPENDIX F
  - TYPICAL OUTPUT FROM MICROCHESS ................. 22
NOTATION

MICROCHESS uses a special octal notation to identify the squares of the chess board. Each square is represented by a two-digit number. The first digit specifies the rank (0 to 7) from the computer's end of the board. The second digit specifies the file (0 to 7) from the computer's right (your left). A completely numbered board is shown below:

<table>
<thead>
<tr>
<th></th>
<th>00</th>
<th>01</th>
<th>02</th>
<th>03</th>
<th>04</th>
<th>05</th>
<th>06</th>
<th>07</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
<td>56</td>
<td>57</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
<td>64</td>
<td>65</td>
<td>66</td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>71</td>
<td>72</td>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
<td>77</td>
<td></td>
</tr>
</tbody>
</table>

CHALLENGER
PROGRAM EXECUTION

The MICROCHESS program is executed from address 0000. After printing the initial sign-on message, MICROCHESS will ask: "DO YOU WANT WHITE? (Y,N)". If you wish to play white, respond with 'Y'. If you wish to play black, respond with 'N'. If you wish MICROCHESS to decide which colour to play, respond with any other character. MICROCHESS will then display the board and prompt with a colon, indicating that the program is ready to receive any operating command.

MICROCHESS COMMANDS

MICROCHESS has seven special commands to which it will respond. Commands may be abbreviated to the first letter of the command word. All commands must be terminated with a carriage return. Typing errors may be corrected at any time by typing a control-X. This will clear the input buffer and allow you to retype the entire line.

COMMAND SUMMARY

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY</td>
<td>Display the board at the terminal.</td>
</tr>
<tr>
<td>GO</td>
<td>Make a move from the current position.</td>
</tr>
<tr>
<td>SPEED</td>
<td>Change the mode of the computer's play.</td>
</tr>
<tr>
<td>RESIGN</td>
<td>End the game.</td>
</tr>
<tr>
<td>EXCHANGE</td>
<td>Exchange sides.</td>
</tr>
<tr>
<td>AUTO DISPLAY</td>
<td>Display the board after each move.</td>
</tr>
<tr>
<td>NO DISPLAY</td>
<td>Do not display the board automatically.</td>
</tr>
</tbody>
</table>
THE DISPLAY COMMAND

The DISPLAY command instructs the computer to display the current position of the internal chess board at the terminal. MICROCHESS is always illustrated at the top of the display, and you are always at the bottom. Each piece is indicated by a two character mnemonic. The first character shows the colour of the piece. The second character shows the type of piece occupying that square. Black squares which are unoccupied are illustrated by ::. The sample display below shows the board set up to begin a game with MICROCHESS playing white.

```
+------- MICROCHESS -------+
| WR VN WB WK WQ WB WN WR |
| WP VP VP VP VP VP VP VP |
| :: :: :: :: :: :: :: |
| :: :: :: :: :: :: :: |
| :: :: :: :: :: :: :: |
| BP BP BP BP BP BP BP BP |
| BR BN BB BK BQ BB BN BR |
+------- CHALLENGER -------+
```

THE GO COMMAND

The GO command instructs MICROCHESS to examine the current position of the board, choose the best move available, make that move, and then print out the move that it has made. This command may be entered at any time. The computer will not check to see if you have made any moves since the last computer move, or if it is making the first move with the black men. MICROCHESS trusts you. You must referee the game.
THE SPEED COMMAND

MICROCHESS can play chess at three different levels. The best level is called the NORMAL speed, and requires from 60 to 300 seconds per move for analysis. By eliminating some time consuming portions of the strategic analysis, the speed can be increased. BLITZ mode requires only 20 seconds per move on the average, and SUPERBLITZ will make a move in about 10 seconds. In response to the SPEED command, MICROCHESS will ask: "WHICH MODE? (S,B,N)". Type one of the characters S,B, or N to choose the desired speed. This command may be entered at any time during the game.

### SPEED SUMMARY

<table>
<thead>
<tr>
<th>ENTER</th>
<th>SPEED</th>
<th>TIME PER MOVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>SUPERBLITZ</td>
<td>5 TO 10 SECONDS</td>
</tr>
<tr>
<td>B</td>
<td>BLITZ</td>
<td>10 TO 30 SECONDS</td>
</tr>
<tr>
<td>N</td>
<td>NORMAL</td>
<td>30 TO 300 SECONDS</td>
</tr>
</tbody>
</table>

THE RESIGN COMMAND

The RESIGN command may be entered at any time to end the game. MICROCHESS will display the final position of the board, and then ask if you wish to play again.

THE EXCHANGE COMMAND

The EXCHANGE command enables you to turn the board around at any point during the game. This forces MICROCHESS to play with your pieces in the position that you have left them. You must play with the computer's men. The relative positions of the pieces remain the same, but the numbering of the squares changes because the notation always has its origin at the computer's lower right.

It is possible to have MICROCHESS play a game against itself by entering the EXCHANGE command, then the GO command, then the EXCHANGE command, and so on. Remember that each move printed is being described from opposite ends of the board because of the intervening exchanges. It is best to display the board every two or three moves to be sure that you are following the game correctly.
ENTERING YOUR MOVE

Your move is described to MICROCHESS by specifying the square
the piece was moved from, and the square the piece was moved
to, using the octal notation described above. For example,
with the computer playing white, a KP to KP4 response would be
entered at the colon prompt as:

: 63-43

MICROCHESS will immediately move the appropriate piece
internally and begin to consider its response. The GO command
is assumed as soon as the move is entered. Note that
MICROCHESS carries out no legal validity check on your move.
The program will accept a move of any piece on the board to any
square on the board. If the square you move the piece to is
occupied, the occupying piece will be captured and removed from
the board. Therefore, it is very important when entering your
move, to take great care not to enter an incorrect square
number. As with the commands, typing errors may be corrected
by typing a control-x and retyping the entire line.

SPECIAL MOVES

Normally, MICROCHESS begins to consider its response as soon as
you have entered your move in the format shown above. If you
wish to inhibit this action, in order to make two consecutive
moves to set up a test position, or to make an en passant
capture as described below, enter an M after the move. For
example:

: 63-43M

MICROCHESS will move the appropriate piece on its internal
chess board, and then return to the command mode for further
commands or moves. Note once again, that you may move any
piece on the board in this manner. This includes the computer's
pieces, which you may wish to move in order to set up a
special position.
CASTLING

Castling is accomplished by entering 0-0 to castle on the king's side (short), and 0-0-0 to castle on the queen's side (long). The letter 0 is used, not the numeral 0.

:0-0

PAWN PROMOTION

If you move a pawn to the eighth rank (rank 0 in the octal notation of MICROCHESS), you may promote it to a piece. This may be done by following the move entry by an equal sign and the mnemonic of the piece you wish the pawn promoted to. For example, if you wish to promote the king pawn to a Queen, the following move would be entered:

: 13-03=Q

Because of the internal board representation of MICROCHESS, only one queen is allowed per side at any given time. If you already have a queen, it will be necessary to choose another piece which has already been lost.

EN PASSANT

En passant pawn capture may be accomplished by making two moves with the capturing pawn. The first move is a lateral move to capture the computer's pawn. The second move is forwards to the final square that you are moving your pawn to. For example, a capture of the computer's queen pawn which has just moved from 14 to 34 with your king pawn, now located at 33, is accomplished by first moving 33 to 34 to capture the pawn (using the M suffix to prevent MICROCHESS from moving), and then moving from 34 to 24 to move your pawn to the appropriate final square.

: 33-34M
: 34-24
THE COMPUTER MOVE

MICROCHESS indicates its move using the same notation that you use to enter your moves. To distinguish your moves from those of the computer when going over an old listing, the computer's moves are preceded by the notation: MC: , as shown in the example game illustrated in appendix F. En passant capture is not a part of the MICROCHESS move generation routines. Consequently, the computer will never capture en passant or recognize the danger of you capturing en passant when it formulates its optimum move.

NOTES

Some players may find that their level of play exceeds that of MICROCHESS. In order to make the game more challenging, these players may make the same sacrifice they might make to a weak human player. They can spot the computer a piece by removing it at the beginning of the game, or shortly after the opening play is concluded. This can easily be done by capturing it with one of your own pieces, then returning the piece to its own square. For example:

: 74-73M
: 73-74M

MICROCHESS has been designed for your enjoyment. Have fun! In addition, we are always open to suggestions, ideas, or criticisms. Please let us know if you feel that there is anything we can do to improve our products, or if there are any new products you would like to see us present.
THE PROGRAM

The program is divided into three functionally distinct sections: the control and input/output routines, the move generation and data collection routines, and the strategic analysis routines.

CONTROL AND INPUT/OUTPUT

This section of the program is responsible for all communications between the computer and the human player. The primary functions carried out are the board set up, and data table initialization sections. In addition to this, the various input commands are interpreted and subroutines are called which execute them. The most important subroutine called by the control section is the chess program itself. This is a complex set of routines which examine the current state of the chess board and return a move which has been evaluated as the best available.

MOVE GENERATION

The second major subsection of the program consists of a set of subroutines which generate legal moves from a given position. MICROCHESS, unlike most larger chess playing programs, evaluates its opportunities in a serial manner. That is, it generates an available move, and evaluates it completely before generating the next available move. The evaluation routines calculate a value for each move which is compared with the value of the best move found so far. If it is better, it becomes the best move for comparison with future moves generated. The move with the highest value will be selected by MICROCHESS.

To generate all the moves for a side MICROCHESS works through a table which contains the board position of each piece. This is the table shown in appendix E. First, a king pawn move is generated and evaluated. The evaluation includes the actual moving of the piece, and the generation of potential reply moves by the challenger. The sequence of trial moves of the computer's pieces and the challenger's pieces may extend as far as three moves for each side beyond the current position. At the end of this time, each move made will be taken back, until the board is returned to its original state. Then, the next available move will be made, and the replies tested. This continues until all the moves for each piece have been tested. MICROCHESS is capable of generating and evaluating about 10,000 moves per second. Thus, in a 300 second analysis 3,000,000 moves will be made and taken back in an attempt to evaluate the available moves.
DATA COLLECTION

For each test move available to the computer data are collected which will allow it to evaluate the resulting position. In the normal mode of operation MICROCHESS collects the following information for use by the strategy algorithms.

MOBILITY ($\mu$). This represents the number of legal moves that a side has available to it from a given position.

MAXIMUM CAPTURABLE PIECE ($\rho$). The value of the most valuable piece presently being attacked by a side.

TOTAL ATTACK ($\alpha$). The sum of the values of all the pieces under attack by a side.

CAPTURE ($\psi$). The value of any piece captured by the current move, or the maximum available capture in a future move which can be achieved by a series of captures (an exchange).

The mobility, maximum capturable piece, and the total attack are obtained for the current position, and the position after the test move has been made for both the computer and its opponent. Capture values are calculated to a depth of three moves per side beyond the current position provided the position examined can be achieved by a continuous sequence of piece captures. In addition, the value of the moving piece, and the squares it occupies before and after the move are used in the evaluation.

STRATEGY

After a test move has been generated, and the parameters above have been collected by the data collection routines, the strategic analysis algorithm assigns a value to the move. The basic algorithm is a linear combination of the various parameters. The basic value is then modified by factors such as the availability of a checkmate, or a positional bonus for motion to the center or out of the back rank.
VALUE = 4.00ψ₁ + 1.25ψ₂ + 0.75ρ₁ + 0.75α₁ + 0.25μ₁
      + 0.25ψ₃ - 3.00ρ₀ - 2.00α₀ - 1.25ψ₀ - 0.25μ₀
      - 0.25α₀ - 0.25ψ₂ - 0.25ψ₅ - 0.25μ₀

(') signifies the challenger's value.

(n) subscript signifies the position at time n.
   (time 0 is the current board position)

VALUE = VALUE + 02 if a piece is moved from the back rank.
VALUE = VALUE + 02 if a piece is moved to the centre.
VALUE = FF if the challenger is checkmated.

The algorithm used by MICROCHESS is a relatively simple one compared to major chess programs which can compete at an expert level of play. As a result, the computer must make the decision between positional development, or material advantage based upon the few factors outlined above. Good chess is considerably more complex, and requires that the player use algorithms which vary from time to time during the game. MICROCHESS has only a single algorithm which must be used at all stages during the game (except for a few opening moves which can be played from a limited book). This single algorithm is a compromise of the possible opening, middle game, end game, and special situation algorithms. It is because of this compromise that MICROCHESS sometimes makes moves which are not optimal.
INPUT AND OUTPUT SUBROUTINES

MICROCHESS is supplied with input and output subroutines for use with an ASR 33 or equivalent ASCII terminal. These routines are shown below in source format:

```
0 DE6
0 DE6
0 DE6
0 DE6
0 DE6
0 DE6
0 DE6 DB 00
0 DE8 E6 80
0 DEA CA E6 0D
0 DE9 78
0 DEE D3 01
0 DF0 C9
0 DF1
0 DF1
0 DF1 DB 00
0 DF3 E6 40
0 DFS CA F1 0D
0 DF8 DB 01
0 DFA E6 7F
0 DFC 47
0 DF0 C9
```

The conventions used by these routines are:

1- Status is on channel 0.
2- Data is on channel 1.
3- Data available is signalled by bit 6 (40H).
4- Transmit buffer empty is signalled by bit 7 (80H).

These routines are shown, so that you may modify them if necessary to suit the individual requirements of your system.

If you wish to use your own I/O routines replace the data at address 09DA (C3 F1 0D) with a JMP to your own input routine (C3 XX XX). Then, replace the data at address 09D7 (C3 E6 0D) with a JMP to your own output routine (C3 XX XX).

The data is passed in the B register. The parity bit may be 0 or 1 for an input instruction. Output from MICROCHESS has the parity bit set to 0. There is no requirement for saving any of the registers; however, the stack pointer must be preserved and the routines must end with a return instruction.
APPENDIX C

DISPLAY OPTIONS

Two display option commands are available at the MICROCHESS command prompt. These are AUTO DISPLAY and NO DISPLAY. Entering the AUTO DISPLAY command causes the program to display the board immediately after each move made by either side. Entering the NO DISPLAY command will turn off the automatic display feature. This is demonstrated in the sample game in Appendix F.

The default option in the copy of MICROCHESS you have received is NO DISPLAY. The user may change the default option to allow the program to display the board after each computer move, after each of the challenger's moves, or both. Replacing the three NOP instructions at address 0120 (00 00 00) with a call to the display subroutine (CD 42 02) will cause the board to be automatically displayed after each move made by MICROCHESS. If you wish to have the board displayed automatically after each of your moves as well, replace the three NOP instructions at address 00D4 with the same subroutine call (CD 42 02).

CRT DISPLAY

If you are using a CRT display with only 16 lines on the screen, you may wish to shorten the board display provided by MICROCHESS. This is easily accomplished by entering 3 NOP instructions (00 00 00) at address 0258. This replaces the CD DA 01 which appears in the original code.

CUSTOM BOARD DISPLAY

If you wish to design your own board display for use with a graphic terminal or just to gratify your own artistic ambitions, you may replace the MICROCHESS display routine by replacing the data at address 0242 (CD AC 09) with a JMP to your own display subroutine (C3 XX XX).
The data required to display the board is contained in a table at address $09ED$. This table contains the board location of each piece. The address and location of each piece as it would appear at the start of a game with MICROCHESS playing white is shown below.

### PIECE ADDRESSES FOR BOARD DISPLAY

<table>
<thead>
<tr>
<th>PIECE</th>
<th>MICROCHESS</th>
<th>CHALLENGER</th>
</tr>
</thead>
<tbody>
<tr>
<td>King</td>
<td>$09ED$ 03</td>
<td>$09FD$ 73</td>
</tr>
<tr>
<td>Queen</td>
<td>$09EE$ 04</td>
<td>$09FE$ 74</td>
</tr>
<tr>
<td>King Rook</td>
<td>$09EF$ 00</td>
<td>$09FF$ 70</td>
</tr>
<tr>
<td>Queen Rook</td>
<td>$09F0$ 07</td>
<td>$0A00$ 77</td>
</tr>
<tr>
<td>King Bishop</td>
<td>$09F1$ 02</td>
<td>$0A01$ 72</td>
</tr>
<tr>
<td>Queen Bishop</td>
<td>$09F2$ 05</td>
<td>$0A02$ 75</td>
</tr>
<tr>
<td>King Knight</td>
<td>$09F3$ 01</td>
<td>$0A03$ 71</td>
</tr>
<tr>
<td>Queen Knight</td>
<td>$09F4$ 06</td>
<td>$0A04$ 76</td>
</tr>
<tr>
<td>KR Pawn</td>
<td>$09F5$ 10</td>
<td>$0A05$ 60</td>
</tr>
<tr>
<td>QR Pawn</td>
<td>$09F6$ 17</td>
<td>$0A06$ 67</td>
</tr>
<tr>
<td>KN Pawn</td>
<td>$09F7$ 11</td>
<td>$0A07$ 61</td>
</tr>
<tr>
<td>QN Pawn</td>
<td>$09F8$ 16</td>
<td>$0A08$ 66</td>
</tr>
<tr>
<td>KB Pawn</td>
<td>$09F9$ 12</td>
<td>$0A09$ 62</td>
</tr>
<tr>
<td>QB Pawn</td>
<td>$09FA$ 15</td>
<td>$0A0A$ 65</td>
</tr>
<tr>
<td>Q Pawn</td>
<td>$09FB$ 14</td>
<td>$0A0B$ 64</td>
</tr>
<tr>
<td>K Pawn</td>
<td>$09FC$ 13</td>
<td>$0A0C$ 63</td>
</tr>
</tbody>
</table>
RETURNING TO YOUR OPERATING SYSTEM

If you wish to return directly to your operating system at the end of a game, this can be accomplished by replacing the HALT instruction at address $01D7$ with a JMP $xx$ $xx$ to your operating system entry point. Two NOPs have been included for your convenience in adding this patch.

Please note, that it is impossible to call MICROCHESS as a subroutine because the program manipulates the stack pointer several times during program execution. Thus, the original return address will not be at the top of the stack when the return instruction is executed.
DO YOU WANT WHITE? (Y,N) N

THE USER DECIDES TO PLAY BLACK.

+------- MICROCHESS -------+
| WR WN WB WK WQ WB WN WR |
| WP WP WP WP WP WP WP WP |
|   II   II   II   II   |
|   II   II   II   II   |
|   II   II   II   II   |
|   II   II   II   II   |
|   II   II   II   II   |
| BP BP BP BP BP BP BP BP |
| BR BN BB BK BQ BB BN BR |
+------- CHALLENGER -------+

THE BOARD IS AUTOMATICALLY DISPLAYED AT THE BEGINNING OF THE GAME.

SPEED

WHICH MODE? (S,B,N) S

GO

MC : 13-33
  63-43
  91-22
  76-55
  82-46
  71-52
  0-0
  AUTO
  52-33

THE USER WISHES TO SELECT THE SPEED OF PLAY.

HE SELECTS THE SUPERBLITZ MODE.

GO: CAUSES MICROCHESS TO MAKE A MOVE.

1 P-K4  P-K4
2 N-KB3  N-QB3
3 B-B4  N-B3
4 O-O

THE USER TURNS ON THE AUTOMATIC DISPLAY FEATURE.

... N X P

THE BOARD IS NOW DISPLAYED AFTER EVERY MOVE.

+------- MICROCHESS -------+
| WK WR II WQ WB WN WR |
| WP WP WP WP WP WP WP WP |
|   III   WN III   II   |
|   III   BN III   II   |
|   III   BP III   WB III |
|   III   III   BN III |
| BP BP BP II BP BP BP BP |
| BR BB BK BQ BB II BR |
+------- CHALLENGER -------+
THE USER TURNS OFF THE AUTOMATIC DISPLAY.

6  Q-K2  N-Q3
7  B X N

THE USER REQUESTS A BOARD DISPLAY.

THE USER EXCHANGES MEN WITH MICROCHESS.
DISPLAY

------- MICROCHESS -------
| BR || BB BQ BK || BR |
| BP BP BP BP BB BP BP BP |
| WB BN || || |
| BP || || |
| WP || || |
| WN || || |
| WP WP WP || WQ WP WP WP |
| WR WN WB || WR WK |
------- CHALLENGER -------

THE BOARD IS DISPLAYED WITH THE MEN EXchanged.
MICROCHESS IS NOW BLACK, AND THE CHALLENGER IS WHITE.

EXCHANGE BACK TO THE ORIGINAL POSITION.

TURN ON THE AUTO DISPLAY.

... P X B
--------- MICROCHESS ---------

WK WR ++ WB WN WR ++
WP WP WP WQ ++ WP WP WP ++
WN ++ ++ ++
++ ++ ++ ++
WP ++ ++
++ ++ BN BP ++
BP BP BP BB BP BP ++ BP ++
BR ++ BK BQ BB ++ BR ++
--------- CHALLENGER ---------

54-66

... N-N2
YOU RESIGNED - I WIN!!!

PLAY AGAIN? (Y,N) N

THANKS FOR THE GAME... MICROCHESS.
Micro-Ware Limited
27 Firstbrooke Road
Toronto Ontario
Canada M4E 2L2