```
*
* 68KTCM.SRC - assembly language source code.
* 68KTCM.LST - source & object code listing.
* 68kTCM.OBJ - S record object file for downloading.
*
* Sample MC68000 program for controlling the
* Microbot Teachmover Robot Arm with the
* MEX68KECB MC68000 Educational Computer Board.
*
* This program moves a block from one position
* to a second position and returns the block to
* the original position.
*
* Connect the Robot Arm left serial port (as viewed
* from the front of the Robot Arm) to the
* Host Port 2 serial port using a special RS232
* male to male cable.  Remember to press the TCM
* Reset button before running the program.  This
* starts the TCM at the zero position.
*
* Assembled by the CUG3:ASM68K Cross Assembler
* on MTS.
*
* written by Steven J. Dombrowski   6/06/86
*
CR      EQU   $0D
        ORG   $6000
*
* Teachmover position commands.
*
POS1    DC.B  '@STEP200,0,-246,0,0,0,600'
        DC.B  CR
POS2    DC.B  '@STEP200,0,0,-479,0,0,-479'
        DC.B  CR
POS3    DC.B  '@STEP200,-390,0,0,0,0,0'
        DC.B  CR
POS4    DC.B  '@STEP200,0,0,0,-89,89,0'
        DC.B  CR
POS5    DC.B  '@STEP200,0,260,0,0,0,0'
        DC.B  CR
POS6    DC.B  '@STEP200,0,0,130,0,0,159'
        DC.B  CR
POS7    DC.B  '@STEP200,0,0,0,0,0,-395'
        DC.B  CR
POS8    DC.B  '@STEP200,0,-643,-332,0,0,-332'
        DC.B  CR
POS9    DC.B  '@STEP200,810,0,0,0,0,0'
        DC.B  CR
POS10   DC.B  '@STEP200,0,0,0,187,-187,0'
        DC.B  CR
POS11   DC.B  '@STEP200,0,650,310,0,0,250'
        DC.B  CR
POS12   DC.B  '@STEP200,0,0,0,0,0,400'
        DC.B  CR
POS13   DC.B  '@STEP200,0,-524,-208,0,0,-208'
        DC.B  CR
POS14   DC.B  '@STEP200,-420,503,579,-98,98,5'
```

```
       DC.B  CR
POS15  DC.B  '@STEP200,420,-503,-579,98,-98,-5'
       DC.B  CR

POS16  DC.B  '@STEP200,0,524,208,0,0,208'
       DC.B  CR
POS17  DC.B  '@STEP200,0,0,0,0,0,-400'
       DC.B  CR
POS18  DC.B  '@STEP200,0,-650,-310,0,0,-250'
       DC.B  CR
POS19  DC.B  '@STEP200,0,0,0,-187,187,0'
       DC.B  CR
POS20  DC.B  '@STEP200,-810,0,0,0,0,0'
       DC.B  CR
POS21  DC.B  '@STEP200,0,643,332,0,0,332'
       DC.B  CR
POS22  DC.B  '@STEP200,0,0,0,0,0,395'
       DC.B  CR
POS23  DC.B  '@STEP200,0,0,-130,0,0,-159'
       DC.B  CR
POS24  DC.B  '@STEP200,0,-260,0,0,0,0'
       DC.B  CR
POS25  DC.B  '@STEP200,0,0,0,89,-89,0'
       DC.B  CR
POS26  DC.B  '@STEP200,390,0,0,0,0,0'
       DC.B  CR
POS27  DC.B  '@STEP200,0,0,479,0,0,479'
       DC.B  CR
POS28  DC.B  '@STEP200,0,246,0,0,0,-600'
       DC.B  CR
POS29  DC.B  $00
*
* Start of main program.
*
       ORG     $1000
       CLR.L   D2
       MOVE.L  #POS1,A5          1st command start addr.
       MOVE.L  #POS2,A6          End 1st string + 1.
       BSR     OUTPUT            Branch to output subroutine.
       MOVE.L  #POS2,A5          2nd command.
       MOVE.L  #POS3,A6
       BSR     OUTPUT
       MOVE.L  #POS3,A5          3rd command.
       MOVE.L  #POS4,A6
       BSR     OUTPUT
       MOVE.L  #POS4,A5          4th command.
       MOVE.L  #POS5,A6
       BSR     OUTPUT
       MOVE.L  #POS5,A5          5th command.
       MOVE.L  #POS6,A6
       BSR     OUTPUT
       MOVE.L  #POS6,A5          6th command.
       MOVE.L  #POS7,A6
       BSR     OUTPUT
       MOVE.L  #POS7,A5          7th command.
       MOVE.L  #POS8,A6
       BSR     OUTPUT
```

```
MOVE.L  #POS8,A5          8th command.
MOVE.L  #POS9,A6
BSR     OUTPUT
MOVE.L  #POS9,A5          9th command.
MOVE.L  #POS10,A6
BSR     OUTPUT
MOVE.L  #POS10,A5         10th command.

MOVE.L  #POS11,A6
BSR     OUTPUT
MOVE.L  #POS11,A5         11th command.
MOVE.L  #POS12,A6
BSR     OUTPUT
MOVE.L  #POS12,A5         12th command.
MOVE.L  #POS13,A6
BSR     OUTPUT
MOVE.L  #POS13,A5         13th command.
MOVE.L  #POS14,A6
BSR     OUTPUT
MOVE.L  #POS14,A5         14th command.
MOVE.L  #POS15,A6
BSR     OUTPUT
MOVE.L  #POS15,A5         15th command.
MOVE.L  #POS16,A6
BSR     OUTPUT
MOVE.L  #POS16,A5         16th command.
MOVE.L  #POS17,A6
BSR     OUTPUT
MOVE.L  #POS17,A5         17th command.
MOVE.L  #POS18,A6
BSR     OUTPUT
MOVE.L  #POS18,A5         18th command.
MOVE.L  #POS19,A6
BSR     OUTPUT
MOVE.L  #POS19,A5         19th command.
MOVE.L  #POS20,A6
BSR     OUTPUT
MOVE.L  #POS20,A5         20th command.
MOVE.L  #POS21,A6
BSR     OUTPUT
MOVE.L  #POS21,A5         21st command.
MOVE.L  #POS22,A6
BSR     OUTPUT
MOVE.L  #POS22,A5         22nd command.
MOVE.L  #POS23,A6
BSR     OUTPUT
MOVE.L  #POS23,A5         23rd command.
MOVE.L  #POS24,A6
BSR     OUTPUT
MOVE.L  #POS24,A5         24th command.
MOVE.L  #POS25,A6
BSR     OUTPUT
MOVE.L  #POS25,A5         25th command.
MOVE.L  #POS26,A6
BSR     OUTPUT
MOVE.L  #POS26,A5         26th command.
MOVE.L  #POS27,A6
```

```
        BSR     OUTPUT
        MOVE.L  #POS27,A5         27th command.
        MOVE.L  #POS28,A6
        BSR     OUTPUT
        MOVE.L  #POS28,A5         last command.
        MOVE.L  #POS29,A6
        BSR     OUTPUT
        MOVE.B  #228,D7           Return to TUTOR.
        TRAP    #14               Program finished.
*
* Subroutine to output string to Robot.

*
OUTPUT MOVE.B  #242,D7           Load OUTPUT21 routine.
        TRAP    #14               Output string to port 2.
OUT     MOVE.B  $10041,D2         Input ACIA2 status.
        CMP.B   #3,D2             Handshake frm TCM rcvd?
        BNE     OUT               Loop if not yet rcvd.
        CLR.L   D2                Clr for next handshake.
        RTS                       Return to main program.
        END
```