# The Least Common Robot Project

by Donald Meyers
and
A. Richard Miller
Miller Microcomputer Services
61 Lake Shore Road, Natick MA 01760
(617/653-6136)

June 13, 1987

## Abstract

This paper describes the first stages of a Forth User Group project to develop a Least Common Robot, a simple and affordable design easily interfaced with common microcomputers. Design trade-offs are aimed at providing easy availability of such robots for school and hobby use, for easy programming in standard Forth.

## Introduction

Miller Microcomputer Services is sponsoring a volunteer project within the joint MMSFORTH and FIG User Groups of Eastern Massachusetts. The project purpose is to develop and document a low-cost interface to the Mobile Armatron Robot (Radio Shack Cat. No. 60-2396, about $40.00 U.S.). The robot and interface are to be used to demonstrate how Forth simplifies the interfacing of a computer to control a device. The interface would allow a direct demonstration of the computer commands. A keyboard command would cause a definite movement of some portion of the robot.

This project was proposed during the "post-game analysis" of a first Forth User Groups' joint exhibit booth, at the Fall 1986 North East Computer Faire. It was felt that providing a stronger focus for the exhibit would entice and convince many more attendees to consider Forth, and an inexpensive and reproducible robot project was the most popular suggestion by far.

The first stage of the project was to make general design decisions. The Mobile Armatron Robot was selected for modification because of its clear advantages over all others in general availability and low cost, plus what appeared to be reasonable levels of mechanical accuracy, electromechanical adaptability, and probable reliability.

For the interface from the computer to the robot, we chose the standard Centronics printer interface which promises to be available on most computers. Serial interfaces and parallel interfaces using a PIA to interface to the computer bus were also considered, but were voted less attractive.

The second stage of the project was to get some hardware into operation. The third stage will be to provide feedback from the robot to determine the positions of the shoulder, wrist, hand, and fingers. The first and second stages have been completed, and a demonstration of the Least Common Robot, sans feedback, is planned for Saturday.

## Robot Description

The Mobile Armatron Robot has five motors. These are for control of:

1. Right Wheel
2. Left Wheel
3. Shoulder
4. Wrist
5. Hand/Fingers

In its original design, this robot houses four C batteries which are arranged to give positive and negative voltages to the drive motors. Currents in excess of 1 ampere were measured going to the motor drives at start-up. Operation is in the range of 100 to 300 milliamperes during normal operation.

A positive voltage applied to the drive motors causes the right and left wheels to go forward, the shoulder to raise, the wrist to raise, and the hand to turn. A negative voltage applied to the motors will cause the right and left wheels to reverse, the shoulder to lower, the wrist to lower, and the fingers to open and close. The shoulder and wrist have limited motion and will hit mechanical stops.

The shoulder can rotate approximately 135 degrees, and is counterbalanced with a heavy spring. Its DC motor has a heavy gear reduction of about 100:1. The drive mechanism uses a combination of idler, pinion, and worm gears; care must be exercised when disassembling the arm.

The wrist is much like the shoulder. The total movement of the wrist is 180 degrees. Its DC motor is combined with several gears; disassembly and assembly are moderately difficult.

Hand and finger operation are driven by a single motor, in an unusual manner. The hand will rotate continuously while a positive voltage is applied to the motor. A negative voltage applied to the same motor instead drives a cam, which causes the fingers to open and close continuously. There are no stops on either the hand rotation or the opening and closing of the fingers. This section has springs and cams. The motor assembly is in the upper portion of the hand. The shared operation of the hand rotation and of the fingers involves an ingenious combination of pawls and catches. Excessive force could cause breakage. This is the most difficult section of the robot to assemble.

Wheel action, in contrast, is very simple. Each of the two drive wheels can be driven in either direction. A third, castor wheel is not driven.

The electrical schematic of the unmodified robot is shown in Figure 1. A manual control pad, attached to the robot by a flat ribbon cable, is a clever arrangement of switches to control the robot's movements. The four C cells are arranged to give positive and negative voltage to each drive motor. In series with each of the leads is a variable resistor for speed control. The switches are contact closures, arranged to provide voltage of the appropriate polarity to the drive motors. In this project, the control pad is removed; instead, the cable is attached to the computer via a new, Centronics parallel interface.

## The Interface

A schematic of the hardware is shown in Figure 2.

The interface is the bridge between the computer hardware and the robot hardware. The Centronics printer interface is the connecting link to the computer. This printer interface has eight data lines and one strobe line. There are four output lines. It was decided to use four of the data lines and the strobe line to control the motor operation. The remaining four data lines are assigned for future use for feedback position signals. A 74154 integrated circuit was chosen for the decoder. This is a 4-line to 16-line decoder.

Relays were chosen for the motor drive, for ease of concept and for availability of parts. The relays are common 12-volt reed relays, from Radio Shack. There are ten relays, one for negative voltage and one for positive voltage for each of the five motors. A transistor is used to drive each of the relays, and a set of NAND gates is used to invert the signal from the 74154 and drive the transistors. Five of these transistors are contained in each CA3183 IC. CD4011 IC's provide four 2-input NAND gates per package. These IC's were chosen for immediate availability (in the hell box), rather than for function. As the design progresses, it should be possible to find a set of inverter drivers which will drive the relays directly.

A bias is needed for the transistors to operate, and is provided by adding a diode in series with the emitters of all the transistors. It was also found that pull-up resistors are desirable, so each line is connected with one.

Required power supplies are a 5 VDC supply for the one 74154 and the three CD4011's. 12 VDC is supplied for the operation of the relay coils. +3 VDC and -3 VDC are supplied from a Radio Shack transformer and two diodes. Although crude, this supply is inexpensive and saves replacing batteries.

The construction was done on perforated board. Sockets were used for the IC's and for some of the inputs to the board. Mostly wire wrap was used. I find this fast and cheap. Hand wrapping and unwrapping tools cost less than $5.00. I also find it far easier to unwrap a wire and put a new one on, compared to soldering.

The 74154 was chosen because it permits only one function at a time to be selected. If both positive and negative were applied to a motor at once, either the relays would weld or the transformer would smoke. A better design would be to install a one-shot in series with the *STB line. This would require the program to toggle the line to keep the motor going. A few times, the drive has gotten to the limit and the motor has kept running with the programmer frantically trying to countermand the last command!

## Software

Enough sample Forth source code is provided to start off any project. Block 215 is used to find the port address and to define a few constants. Block 216 is a temporary test program (with the printer attached), to determine whether Block 215 worked. Block 217 defines some shorthand constant wordnames for the fundamental movements. Block 218 and up are some examples of software that you can build upon this base. Have fun!

## Future Plans

MMS and the combined Forth User Groups plan to continue into the third stage of this project, adding feedback mechanisms during this Summer and Fall. The Least Common Robot is to be the main attraction at the Forth User Groups' exhibit booth, at the North East Computer Faire in Boston on October 15th through 17th, and one or more how-to articles are planned in popular magazines. It is hoped that these volunteer efforts will encourage a large number of Least Common Robot "laboratories" for hobby and educational use. That, in turn, can provide an interesting and convincing demonstration of Forth's flexibility and ease of use.

If you are in the Boston area, come on down and get involved. If you live elsewhere but want to contribute your ideas to this project, please send written information to Dick Miller at MMS, who will attempt to link appropriate volunteers.

## Credits

Project suggested by Jill Miller; documentation and coordination by Miller Microcomputer Services.
Electromechanical design and development by Donald Meyers, with suggestions by Doug Lurie.
Sample Forth programming by Jim Gerow, Don Meyers and David Lindbergh.

LEAST COMMON ROBOT PROJECT (DRAFT); 6/11/87 (22:13:16)

```
Block 210 [210 :0]

( 870505arm Modified ROBOT PROJECT routines, 1 of 8 )
: TASK ;
VARIABLE ROBOT-PORT

: SET-PORT ( printer# -> )    DUP +  1032 +  D@
  ROBOT-PORT ! ;

: SELECT ( func unit -> )     15 AND  SWAP  16* OR
  ROBOT-PORT @ PC! ;

: ON-OFF ( flag -> )          IF 13  ELSE 12  THEN
  ROBOT-PORT @ 2+ PC! ;

: FEEDBACK ( -> status-bits )
  ROBOT-PORT @ 1+ PC@ ;
                                                   -->
```

```
Block 211 [211 :0]

( 870505arm Modified ROBOT PROJECT routines, 2 of 8 )       -->

: TEST        ( not needed for final routines )
  0 SET-PORT ( select printer port )
             ( print ASCII characters, Space -> Delete )  8 2
  DO         ( outer loop is the function upper nibble ) 16 0
    DO       ( inner loop is the unit lower nibble )
      BEGIN  FEEDBACK  128 AND  128 =
      UNTIL                   ( wait until ready )
        J I SELECT            ( output the character )
        1 ON-OFF   0 ON-OFF   ( strobe )
    LOOP
  LOOP   PRINTER ON  CR CR  PRINTER OFF ;

                                                            -->
```

```
Block 212 [212 :0]

( 870505arm Modified ROBOT PROJECT routines, 3 of 8 )

0 CONSTANT RWF ( right wheel fwd )  1 CONSTANT RWR ( and revrs )
2 CONSTANT LWF ( left  wheel fwd )  3 CONSTANT LWR ( and revrs )
4 CONSTANT ARMUP                    5 CONSTANT ARMDN
6 CONSTANT WRUP ( wrist up )        7 CONSTANT WRDN ( and down )
8 CONSTANT WRTN ( wrist turn )      9 CONSTANT FINMV ( fingers )

: DELAY   50 0 DO LOOP ;

: GO ( rwf n -> )   SWAP  2 SWAP   SELECT
  1 ON-OFF  50 *  0 DO DELAY  LOOP  0 ON-OFF    0 ON-OFF ;

0 SET-PORT ( for printer#0 )
                                                               -->
```

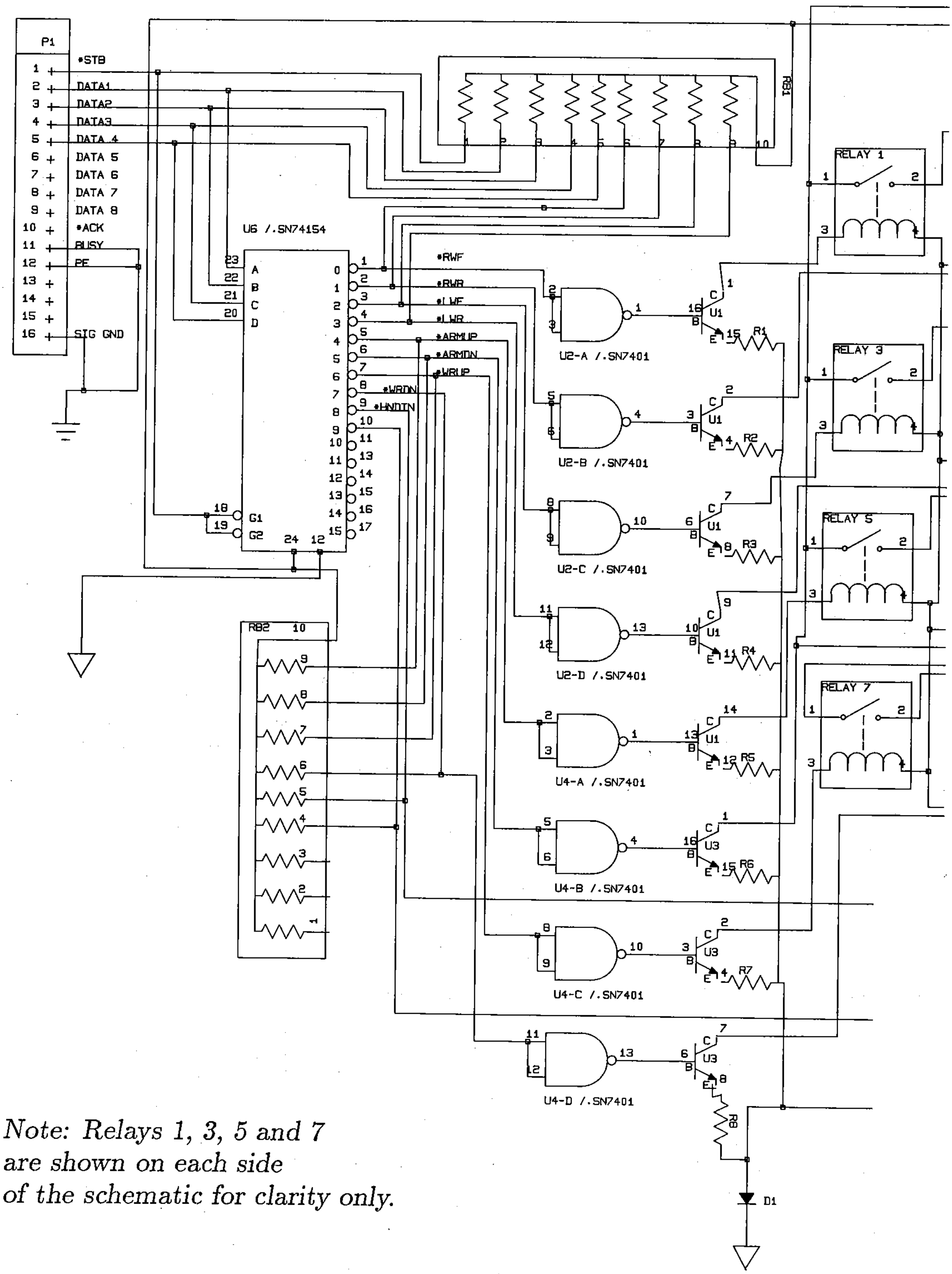
```
Block 213 [213 :0]

( 870510arm Modified ROBOT PROJECT routines, 4 of 8 )

: TEST1        10 0 DO     I 4 GO             LOOP ;

: FORWARD      9 0 DO  RWF 1 GO  LWF 1 GO  LOOP ;
: REVERSE      9 0 DO  RWR 1 GO  LWR 1 GO  LOOP ;
: LEFT-TURN    5 0 DO  RWF 1 GO  LWR 1 GO  LOOP ;
: RIGHT-TURN   5 0 DO  RWR 1 GO  LWF 1 GO  LOOP ;

: ARM-UP   5 0 DO  ARMUP  1 GO  LOOP ;
: ARM-DN   5 0 DO  ARMDN  1 GO  LOOP ;
: WR-UP    5 0 DO  WRUP   1 GO  LOOP ;
: WR-DN    5 0 DO  WRDN   1 GO  LOOP ;
: WR-TN    5 0 DO  WRTN   1 GO  LOOP ;
: FIN-MV   5 0 DO  FINMV  4 GO  LOOP ;
                                                               -->
```

```
Block 214 [214 :0]

( 870510arm Modified ROBOT PROJECT routines, 5 of 8 )
: FWD    RWF  1 GO  LWF  1 GO ;
: RIGHT  RWF  1 GO  LWR  1 GO ;
: REV    RWR  1 GO  LWR  1 GO ;
: LEFT   RWR  1 GO  LWF  1 GO ;

: NOP ;

: MOVE  ( -> b )  ." MOVE"
  BEGIN
     ?KEY DUP NCASE  168 171   173 170  0
              "       FWD RIGHT REV LEFT NOP
              OTHERWISE  EXIT
              CASEND     DROP   0
  UNTIL ;
                                                               -->
```

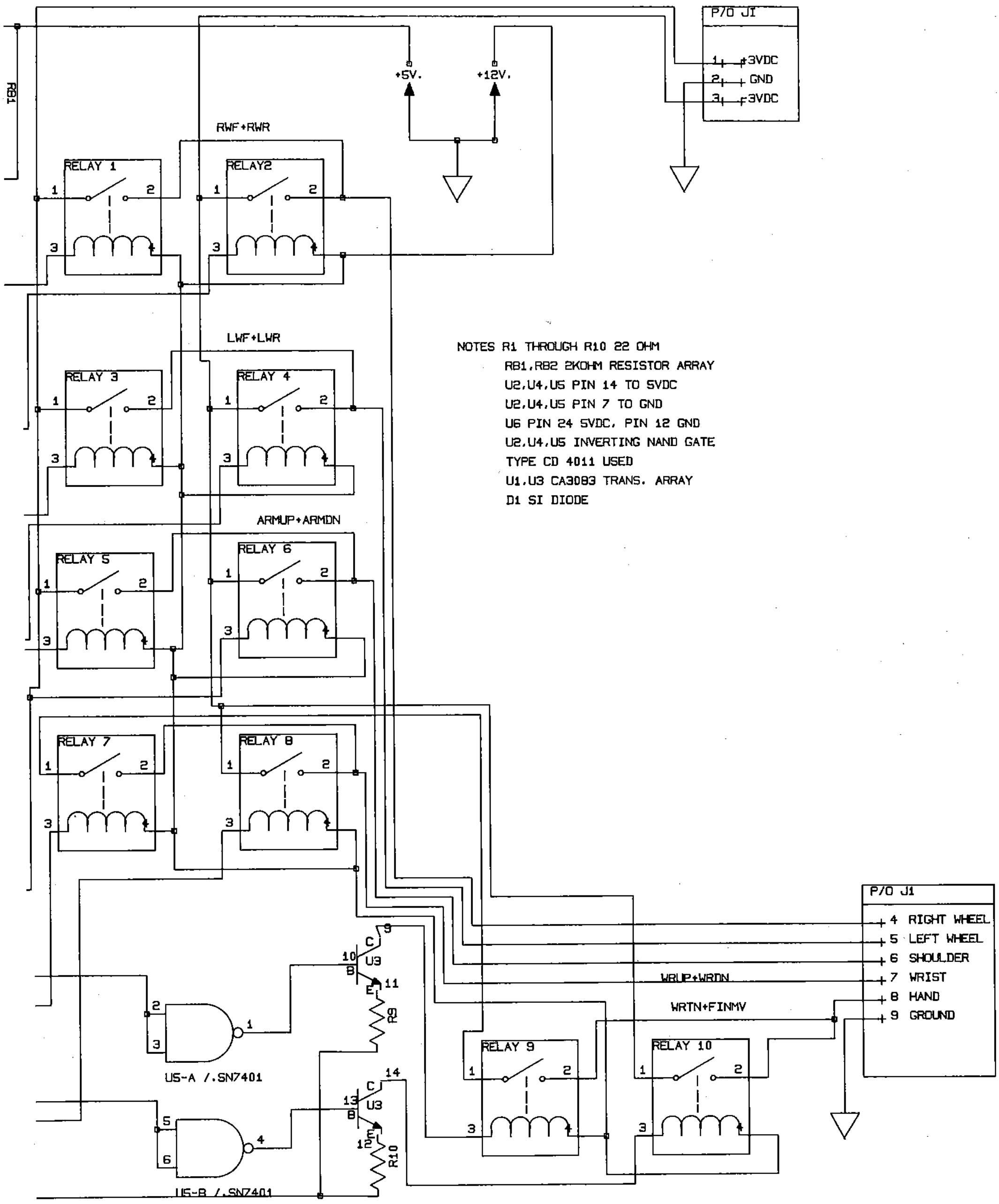
```
Block 215 [215 :0]

( 870510arm Modified ROBOT PROJECT routines, 6 of 8 )
: AU       ARMUP  1 GO ;
: AD       ARMDN  1 GO ;
: FINGER   FINMV  1 GO ;
: TURN     WRTN   1 GO ;

: ARM  ( -> b ) ." ARM"
  BEGIN
     ?KEY DUP NCASE 168 171    173 170  0
               "        AU  FINGER AD  TURN NOP
               OTHERWISE  EXIT
               CASEND     DROP  0
  UNTIL ;


                                                               -->
```

```
Block 216 [216 :0]

( 870510arm Modified ROBOT PROJECT routines, 7 of 8 )

: WU  WRUP  1 GO ;
: WD  WRDN  1 GO ;

: WRIST  ( -> b )  ." WRIST"
  BEGIN
     ?KEY DUP NCASE 168 171    173 170  0
              "    WU  FINGER WD  TURN NOP
              OTHERWISE  EXIT
              CASEND     DROP  0
  UNTIL ;



                                                               -->
```

```
Block 217 [217 :0]

( 870510arm Modified ROBOT PROJECT routines, 8 of 8 )

: ROBOT ( -> )  77 ( MOVE )
  BEGIN  ACASE AWM" ARM WRIST MOVE
         OTHERWISE EXIT  CASEND  CR  0 UNTIL ;

: ROBOT2
  BEGIN   ?KEY DUP
    NCASE 168 171   173 170  167 172 169 174 175  176    0
    "     FWD RIGHT REV LEFT AU  AD  WU  WD  TURN FINGER NOP
    OTHERWISE  EXIT
    CASEND  DROP  0
  UNTIL ;
```




*Note: Relays 1, 3, 5 and 7 are shown on each side of the schematic for clarity only.*


P/O J1
1 +3VDC
2 GND
3 +3VDC
+5V.
+12V.
RB1
RWF+RWR
RELAY 1
RELAY2
LWF+LWR
RELAY 3
RELAY 4
ARMUP+ARMDN
RELAY 5
RELAY 6
RELAY 7
RELAY 8
NOTES R1 THROUGH R10 22 OHM
RB1,RB2 2KOHM RESISTOR ARRAY
U2,U4,U5 PIN 14 TO 5VDC
U2,U4,U5 PIN 7 TO GND
U6 PIN 24 5VDC, PIN 12 GND
U2,U4,U5 INVERTING NAND GATE
TYPE CD 4011 USED
U1,U3 CA3083 TRANS. ARRAY
D1 SI DIODE
P/O J1
4 RIGHT WHEEL
5 LEFT WHEEL
6 SHOULDER
7 WRIST
8 HAND
9 GROUND
WRUP+WRDN
WRTN+FINMV
U3
R9
R10
U5-A /.SN7401
U5-B /.SN7401
RELAY 9
RELAY 10

ROBOT