S E C T I O N 1

A S Y S T E M E Q U A T E S

B S Y S T E M V A R I A B L E S

C S Y S T E M C O N S T A N T S

S

O

F

T

W

A

R

E.

ARM TRAINER MK2AL

DIRECT FULL STEP MOTOR CONTROL

FOR TRS8Ø MODEL 1, LEVEL 11

BY ANDREW LENNARD

*** July 1981 ***

4. SOFTWARE

4.1 Introduction

A machine code program, LEARN , to drive the ARMDROID has been specially written. It was designed for the Tandy TRS-80 Model 1 Level 11, and the loading instructions given here apply to that computer. But the program can be easily adapted to any Z80 microprocessor with the necessary port, and versions made available for the leading makes with variations of these instructions where appropriate. But of course users can write their own software in whatever language they choose.

4.2 Loading

When in Basic type SYSTEM, press ENTER, answer the '*' with LEARN and then press ENTER again. The cassette tape will take about 1½ minutes to load. Answer the next '*' with / 17408 and press ENTER.

4.3 General Description

LEARN is a menu-oriented program for teaching the ARMDROID a sequence of movements which it will then repeat either once or as many times as you like. The program is divided into four sections, one for learning the sequence and for fine-tuning it, one to save the sequence on tape and load it again , one for moving the arm without the learning function, and finally two exit commands.

We suggest that, if this is your first encounter with the program, you should read quickly through the commands without worrying too much about understanding all the details. Then go to Section 4.5 and follow the 'Sequence for Newcomers'. This will give you a good idea of what the program does. After that you can begin to discover some of the subtleties of planning and fine-tuning sequences of movements.

4.4 Explanation

L(EARN)

Stores a sequence of manual movements in memory. The arm is moved using the commands explained under M(ANUAL) . You can exit the command by pressing O (zero) , press G(O), and the arm will repeat the movement you have taught it.

On pressing L(EARN) you will be asked whether you want the S(TART) again or C(ONTINUE) from the current position. The first time press S(TART) . The arm is then free to be moved by hand without the motors' torque preventing you. Move it to a suitable starting position, then press the space bar. You will find that you cannot now move the arm by hand.

To add a sequence already in memory press C(ONTINUE) instead of S(TART).

Using the manual commands, move the arm to another position. As it goes the computer is adding up the steps each motor is making, either forward or back, and storing the data in memory. (holding the space bar down during manual control slows the movement)

Exit by pressing O (zero).

D (ISPLAY)

Displays the sequence stored in memory. The sequence can be edited with the E(DIT) command.

The six columns of figures correspond to the six motors, and the order is the same as that of the 1-6/Q-Y keys (see M(OVE). The first row (RELPOS) shows the current position. Each row represents a stage of the movement, and the actual figures are the number of steps each motor is to make, positive for forward, negative for reverse. The maximum number of steps stores in a row for one motor is +127 or -128, so if a movement consists of more than this number it is accomodated on several rows.

Movements of the arm can be fine-tuned by editing (see E(DIT)) the figures on display until the arm is positioned exactly.

Scrolling of the display can be halted by pressing O (zero). To continue scrolling, press any other key. To display the figures one after the other, keep pressing O.

E(DIT)

Allows the user to change the figures in the memorised sequence.

Truncate a sequence by pressing R(OW COUNT), then ENTER, then the number of the last row you want performed, and finally ENTER. This clears the memory from the next step pnwards, so you should only do this if you do not want the rest of the sequence kept in memory.

By pressing M(OTOR STEP), you can change any of the numbers in any row and column.

S(ET ARM)

Sets the current position of the arm as the 'zero' or starting position.

When pressed from the Menu, it simply zeroes the first row of the display.

S(ET ARM) has another function. During a L(EARN), pressing S(ET ARM) at any moment when the arm is at rest will ensure that the movements before and after are separated from each other instead of being merged. This is the way to make quite sure that the arm passes through a particular point during a sequence. Try the same two movements without pressing S(ET ARM), and note the difference in the display.

It is important to realise that, if a sequence has been memorised and S(ET ARM) is pressed from the Menu when the arm is not in its original starting position, pressing G(O) will take the arm through the sequence but from the new starting point. This can be useful for adjusting the whole of a sequence (perhaps slightly to right or left), but it can lead to the arm running into objects if the new starting point is not selected with care.

W(RITE)

Writes a memorised sequence to cassette tape.

R(EAD)

Reads a previously written sequence from cassette tape into memory.

C(HECK)

Compares a sequence written to cassette tape with the same sequence still in memory, to verify the tape.

G(O)

Moves the arm through a memorised sequence, either once or repeatedly.

It is important to make sure that the starting point in memory is the right one, or the sequence may try to take the arm into impossible positions. (see S(ET ARM)

T(O START)

Takes the arm back to the zero or starting position.

F(REE)

Removes the motors torque from the arm, thus allowing it to be moved by hand.

M(ANUAL)

Gives the user control of the movements of the arm direct from the keyboard. It is used (a) for practising manual control before L(EARN)ing, (b) for trying new combinations of separate movements, and (c) for moving the arm to a new starting position before pressing S(ET ARM). Holding the space bar down slows the movement by a factor o about 3.

The motors are controlled with the keys 1-6/Q-Y. The keys operate in pairs, each pair moving a motor forwards and backwards. Any combinati of the six motors may be moved together (or of course separately), but pressing both keys of a pair simply cancels any movement on that motor.

The geometry of the arm is designed to give the maximum flexibility combined with maximum practicality. A movement of one joint affects only that joint: with some designs one movement involuntarily produces movement in other joints.

It is a feature of the ARMDROID that it has a so-called 'parallelogra operation. Starting with the upper arm vertical, the forearm horizontal and the hand pointing directly downwards, the shoulder joint can be rotated in either direction and the forearm and hand retain their orientation. Equally the forearm can be raised and lowered while leaving the hand pointing downwards. Moving the arm outwards and down by rotating both the shoulder joints together still leaves the hand vertical. This is of vital importance for simplifying the picking and placing of objects.

The motors controlled by the keys are:

1/Q: Gripper
2/W: Wrist left
3/E: Wrist right
4/R: Forearm
5/T: Shoulder
6/Y: Base

B(OOT)

Returns the computer to the program start and clears the memories.

Q(UIT)

Returns the computer to TRS80 System level.

4.5 INTRODUCTORY DEMONSTRATION SEQUENCE

1. After loading the program, the screen shows the menu. Press L to enter L(EARN).

2. Screen: START AGAIN OR C(ONTINUE) FROM PRESENT POSITION, (.) TO EXIT. Press S

3. Screen: " ARM RESET
   ARM NOW FREE TO MOVE
   TYPE SPACE BAR WHEN READY, OR FULL STOP TO EXIT"
   Now move the arm so that both arm and forearm are vertical with the hand horizontal. For coarse movements grasp the forearm or upper arm and move it. For fine adjustments and for movements of the hand, it is better to use the large white gear wheels in the shoulder joint. Press the space bar and the arm will become rigidly fixed.

4. Screen: "*** TORQUE APPLIED ***"
   You can now move the arm using the 1-6/Q-Y keys as explained in the manual section. Try just one movement alone at first. Now press O (zero) to exit from L(EARN). The arm will return to the starting position, and the Menu appears on the screen.

5. Screen: Menu. Press D for D(ISPLAY).

6. Screen: Display and Menu. The numbers of steps you applied to each motor have been memorised by the computer, and these steps are now displayed see D(ISPLAY) section for explanation. Press G for G(O).

7. Screen: "DO (F) OREVER OR (O) NCE?. Press O (letter O), and the arm will repeat the movement it has learnt.

8. Screen: "SEQUENCE COMPLETE" and Menu. Press L.

9. Screen: as 2 above. This time press C. Now you can continue the movement from this position, using the 1-6/Q-Y keys as before. Now press O (zero). Again the arm returns to its original position.

10. Screen: Menu. Press D

11. Screen: Display and menu. Your new movement has been added to your first. Press G.

12. Screen: as 7 above. This time press F. Each time a sequence is started a full point is added to the row on the screen. To stop press full point.

This is a very simple demonstration of how complex movements can be built up, learnt as a sequence and then repeated endlessly and with great accuracy.

```
SYSTEM EQUATES


PORT    EQU    Ø 4      ;  ARM PORT NUMBER
FINAD   EQU    O2B2     ;  SYSTEM RESTART
PCHR    EQU    ØØ33H   ;  SYSTEM PRINT CHARACTER
GCHR    EQU    ØØ49H   ;  SYSTEM GET CHARACTER
KBD     EQU    ØØ2BH   ;  SCAN KEYBOARD
PUTSTR  EQU    28A7H    ;  SYSTEM PRINT STRING
CASON   EQU    Ø212H    ;  CASSETTE ON
CASOF   EQU    Ø1F8H    ;  CASSETTE OFF
RDHDR   EQU    Ø296H    ;  READ HEADER ON CASSETTE
READC   EQU    Ø235H    ;  READ CHARACTER FROM CASSETTE
WRLDR   EQU    Ø287H    ;  WRITE HEADER TO CASSETTE
WRBYA   EQU    Ø264H    ;  WRITE CHARACTER TO CASSETTE
MINUS   EQU    '-'      ;  ASCII MINUS
SPAC    EQU    '-'      ;  ASCII SPACE
NL      EQU    ØDH      ;  ASCII NEW LINE
NUMBA   EQU    3ØH      ;  ASCII NUMBER BASE
MAXLE   EQU    1Ø       ;  UPPER BOARD FOR ARST ROW COUNTER

;    ORG       174Ø 8  ;  = 44ØØ TRS8Ø HEX ADDRESS
                        ;  FOR START OF PROGRAM
```

```
VARIABLES USED

MIN             DEFB ØØ       ;  Has value of one if number input negative
MAN             DEFB ØØ       ;  If MAN = zero then steps are stored
STRFG           DEFB ØØ       ;  If STRFG non zero then store TBUF array
KEYP            DEFB ØØ       ;  Set if key pressed in KEYIN Routine
FORFG           DEFB ØØ       ;  Set if sequence to be done forever

COUNT           DEFB ØØØØ     ;  Number of motor slices stored
CUROW           DEFB ØØØØ     ;  Pointer to next free motor slice

ARRAYS

NUMAR           DEFS 1Ø       ;  Store used for Binary to ASCII Conversion
                              ;  Routine CTBAS

POSAR           DEFS 12       ;  Each two bytes of this six element array
                              ;  contain on value which is used to
                              ;  keep track of each motors motion,
                              ;  hence the array can be used to reset
                              ;  the arm, moving it into a defined
                              ;  start position.
                              ;  Each 16 bit value stores a motor
                              ;  steps in two's complement arithmetic.

CTPOS           DEFS  6       ;  6 Bytes, each relating to a motor.
                              ;  A number from 1-4 is stored in
                              ;  each bytes and this is used to
                              ;  index the FTABL (see constant definition)

TBUF            DEFS  6       ;  When learning a move sequence the
                              ;  six motors motions are stored in this
                              ;  six byte array.  Each byte relates
                              ;  to a motor and holds a motor step
                              ;  count in the range -128 to +127
                              ;  If the motor changes direction or a
                              ;  count exceeds the specified range then
                              ;  the whole TBUF array is stored in
                              ;  the ARST array and the TBUF array
                              ;  is cleared.
                              ;  TBUF means temporary buffer.

DRBUF           DEFS  6       ;  Each byte relates to the previous
                              ;  direction of a motor.

MOTBF           DEFS  6       ;  A six byte array used by DRAMT to
                              ;  tell which motors are being driven, and
                              ;  in which direction.
                              ;  Bit zero set if motor to be driven
                              ;  Bit one set if motor in reverse
                              ;  Byte zero if motor should not be driven.

ARST            DEFS N*6      ;  This array holds the sequence that
                              ;  the user teaches the system.  The array
                              ;  consists of N*6 bytes where N is
                              ;  the number of rows needed to store the
                              ;  sequence.
```

CONSTANTS USED

```
FTABL           DEFB 192    ;
                DEFB 144    ;
                DEFB  48    ;
                DEFB  96    ;


                ;  FTABL is a small table which defines the
                ;  order of the steps as they are sent out
                ;  to the arm.  To drive each motor the
                ;  DRAMT routine adds the motors offset
                ;  which is obtained from CTPOS and adds
                ;  this to the FTABL start address -1.  This
                ;  will now enable the DRAMT routine to
                ;  fetch the desired element from the FTABL
                ;  array, and this value is then sent to
                ;  the motor via the output port.
```

```
            CONSTANTS AND ARRAYS
            STRINGS


            SIGON     DEFM                          *** COLNE ROBOTICS ARM CONTROLLER
MK (AL2) ***'
                      DEFW        ØØØDH
            RELYQ     DEFB        ØDH
                      DEFM        'REALLY QUIT? (Y/N)'
                      DEFW        ØØ
            SIGOF     DEFW        ØDØDH
                      DEFM        'YOU ARE NOW AT TRS8Ø SYSTEM LEVEL'
                      DEFW        ØØ
            ECOMS     DEFM        'EDIT (M)OTOR STEP, OR (R) OW COUNT?'
                      DEFW        ØØØDH
            COUTS     DEFM        'NEW UPPER ROW BOUND IS?'
                      DEFB        ØØ
            EDSTR     DEFM        'ROW NUMBER?'
                      DEFB        ØØ
            BADMS     DEFM        '*** BAD INPUT VALUE ***'
                      DEFW        ØØØDH
            MOTNS     DEFM        'CHANGE STEPS ON WHICH MOTOR?'
                      DEFB        ØØ
            NVALS     DEFM        'REPLACEMENT STEP VALUE?'
                      DEFB        ØØ
            QUESS     DEFM        'LRN, READ, CHECK,WRITE, GO, DISP, BOOT, MAN,
                                  QUIT,  SETA, TOST, EDT, FREE
                      DEFW        ØØØDH
            RORNM     DEFM        'DO (F)OREVER OR (O)NCE?'
                      DEFB        ØØ
            CASRD     DEFM        'TYPE SPACE BAR WHEN READY, OF FULL STOP TO EXI
                      DEFB        ØØ
            QMESS     DEFM        'PARDON'
                      DEFW        ØØØDH
            BOOTS     DEFB        ØDH
                      DEFM        'WANT TO RE-START (Y/N)?'
                      DEFB        ØØ
            RELNS     DEFM        'START AGAIN OR (C)ONTINUE FROM CURRENT POSITION
                                  (.) TO EXIT
                      DEFW        ØØØDH
            DISPS     DEFB        ØDH
                      DEFM        ' *** MOVEMENT ARRAY DISPLAY *** '
                      DEFB        ØDH
                      DEFW        ØØØDH
            NODIS     DEFM        '*** NO SEQUENCE IN STORE ***'
                      DEFB        ØDH
                      DEFW        ØØØDH
            OVFMS     DEFM        'NO MORE ARM STORE LEFT, DELETE OR SAVE?'
                      DEFW        ØØØDH
            DONMS     DEFB        ØDH
                      DEFM        'SEQUENCE COMPLETE'
                      DEFW        ØØØDH
            RDMSG     DEFM        '*** READ ERROR ***'
                      DEFW        ØØØDH
            TAPOK     DEFM        '*** TAPE OK ***'
                      DEFW        ØØØDH
            STRST     DEFM        'ARM RESET'
                      DEFW        ØØØDH
            NOTOR     DEFM        'ARM NOW FREE TO MOVE'
```

```
        DEFB    ØØØDH
TORMS   DEFB    ØDH
        DEFM    '*** TORQUE APPLIED ***'
        DEFW    ØØØDH
POSST   DEFM    'RELPOS='
        DEFB    ØØ
```