```
LOCATION OBJECT CODE LINE     SOURCE LINE

  10AE EB                           EX   DE,HL                      ;Save momentarily
  10AF E1                           POP  HL                         ;Get back original timer
  10B0 CBA6                         RES  EOT,[HL]                   ;Reset previous last timer
  10B2 EB                           EX   DE,HL                      ;Get back current last timer
  10B3 D1                           POP  DE                         ;Restore DE register
  10B4 18A4                         JR   TIMER1                     ;Go back up and initialize counter for use

  10B6                  INIT_TIMER__EXIT:
  10B6 E1                           POP  HL
  10B7 CBAE                         RES  FREE,[HL]
  10B9 78                           LD   A,B             ;Put the offset into the Accumulator for the user of routine
  10BA C9                           RET


                        ;Procedure Test Signal
                        ;Acc has the Signal number to be tested
                        ;A value of True(1) or False(0) is returned in the Accumulator for the
                        ;                 Signal given.
                        ;           COMN
                        ;TEST_SIG_NUM:
                        ;           DEFS 1

                                    PROG
  10BB                  TEST_SIG_PARAM:
  10BB 00010001                     DEFW 1,1

  10BF                  TEST_SIGNALQ
  10BF 0110BB                       LD   BC,TEST_SIG_PARAM
  10C2 1173C2                       LD   DE,TEST_SIG_NUM
  10C5 CD0098                       CALL PARAM_
  10C8 3A73C2                       LD   A,[TEST_SIG_NUM]

  10CB                  TEST_SIGNAL_
  10CB 4F                           LD   C,A                        ;Put Signal Code into C register
  10CC 2A73D3                       LD   HL,[TIMER_TABLE_BASE]      ;Get Timer Base Address
  10CF 47                           LD   B,A                        ;Save Signal
  10D0 110003                       LD   DE,3                       ;Set up offset for next timer
  10D3 B7                           OR   A                          ;See if first timer is a match
  10D4 2808                         JR   Z,SIGNAL_MATCH             ;If so go check it

  10D6                  TEST1:                                      ;Loop to match timer table to desired timer
  10D6 CB66                         BIT  EOT,[HL]                   ;Check for end of table
  10D8 200C                         JR   NZ,SIGNAL_FALSE            ;If so then return a not done
  10DA 19                           ADD  HL,DE                      ;Now index to next timer
  10DB 0D                           DEC  C                          ;Decrement to the timer desired
  10DC 20F8                         JR   NZ,TEST1                   ;If not a timer desired timer then go back

  10DE                  SIGNAL_MATCH:                               ;Here with a timer match
  10DE CB6E                         BIT  FREE,[HL]
  10E0 2004                         JR   NZ,SIGNAL_FALSE
  10E2 CB7E                         BIT  DONE,[HL]                  ;Check for timer done
  10E4 2003                         JR   NZ,SIGNAL_TRUE             ;If so then go return a True
  10E6                  SIGNAL_FALSE:                               ;Here to return a false for either
                                                                    ;a not done or non_existent timer
  10E6 AF                           XOR  A                          ;Put a false in Acc
  10E7 180A                         JR   TEST_EXIT                  ;Go to the exit
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

   10E9             4480 SIGNAL_TRUE:                                    ;Here when timer is finished
   10E9 CB76        4481            BIT  REPEAT,[HL]                     ;Check for repeating timer
   10EB 2002        4482            JR   NZ,SIGNAL_TRUE1                 ;If so then just return True
   10ED CBEE        4483            SET  FREE,[HL]                       ;Free current timer since not repeating
   10EF             4484 SIGNAL_TRUE1:
                    4485                                                 ;***** Start add 4/30/82****
   10EF CBBE        4486            RES  DONE,[HL]                       ;Reset current timer to not done
                    4487                                                 ;***** End add 4/30/82****
   10F1 3E01        4488            LD   A,1                             ;Put a True in the Acc
   10F3             4489 TEST_EXIT:                                      ;Return
   10F3 B7          4490            OR   A
   10F4 C9          4491            RET
                    4492
                    4493            DATA
                    4494
   7303             4495 TIMER_TABLE_BASE:     DEFS 2
   7305             4496 NEXT_TIMER_DATA_BYTE: DEFS 2
                    4497
                    4498  PROG
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                    4500 ;CONTROLLER SOFTWARE
          <0000>    4501 FIRE              EQU            0          ;BITS IN STATUS WORD TO CHECK
          <0001>    4502 JOY               EQU            1          ;WHETHER DEVICE IS ACTIVE
          <0002>    4503 SPIN              EQU            2
          <0003>    4504 ARM               EQU            3
          <0004>    4505 KBD               EQU            4
          <0002>    4506 PLYR_0            EQU            2          ;OFFSET TO CONTROLLER MEMORY
          <0007>    4507 PLYR_1            EQU            7
          <0007>    4508 SEG_0             EQU            07H        ;BIT MASK IN PLAYER STATUS
          <0018>    4509 SEG_1             EQU            18H        ;FOR SEGMENT 0 OR 1 DEVICES
          <0000>    4510 FIRE_OLD          EQU            0          ;OFFSETS TO DEBOUNCE STATUS
          <0001>    4511 FIRE_STATE        EQU            1          ;BUFFER: (OLD,STATE)
          <0002>    4512 JOY_OLD           EQU            2
          <0003>    4513 JOY_STATE         EQU            3
          <0004>    4514 SPIN_OLD          EQU            4
          <0005>    4515 SPIN_STATE        EQU            5
          <0006>    4516 ARM_OLD           EQU            6
          <0007>    4517 ARM_STATE         EQU            7
          <0008>    4518 KBD_OLD           EQU            8
          <0009>    4519 KBD_STATE         EQU            9
          <000F>    4520 KBD_MASK          EQU            0FH       ;MASK FOR INPUT DATA BYTE
          <0040>    4521 FIRE_MASK         EQU            40H
          <0040>    4522 ARM_MASK          EQU            40H
          <000F>    4523 JOY_MASK          EQU            0FH
          <0080>    4524 SPNR_MASK         EQU            10110000B
          <0005>    4525 NUM_DEV           EQU            5         ;NUMBER OF POSSIBLE DEVICES
                    4526 ;STACK             EQU            73FFH
                    4527 ;MODE_0_PORT       EQU            0BEH       ;D8
                    4528 ;MODE_1_PORT       EQU            0BFH       ;D9
          <000F>    4529 KBD_NULL          EQU            0FH
          <00FF>    4530 CTRL_1_PORT       EQU            0FFH
          <00FC>    4531 CTRL_0_PORT       EQU            0FCH
          <00C0>    4532 STRB_RST_PORT     EQU            0C0H             ;STROBE RESET PORT
          <0080>    4533 STRB_SET_PORT     EQU            080H             ;STROBE SET PORT
          <0000>    4534 CONTROLLER_0      EQU            0
          <0001>    4535 CONTROLLER_1      EQU            1
          <0000>    4536 STROBE_RESET      EQU            0
          <0001>    4537 STROBE_SET        EQU            1
                    4538 ************************ MACRO *********************************
                    4539
                    4540
                    4541 DELAY_10    MACRO
                    4542                 CALL DELAY
                    4543                 MEND
                    4544
                    4545 ******************************DATA*****************************
                    4546 *
                    4547 *  DECODER TABLE FOR THE KEYBOARD
                    4548 *
10F5                4549 DEC_KBD_TBL
10F5 0F             4550                 DEFB KBD_NULL          ; NULL ENTRY
10F6 06             4551                 DEFB 6                 ; '6'
10F7 01             4552                 DEFB 1                 ; '1'
10F8 03             4553                 DEFB 3                 ; '3'
10F9 09             4554                 DEFB 9                 ; '9'
10FA 00             4555                 DEFB 0                 ; '0'
10FB 0A             4556                 DEFB 10                ; '*'
```

```
OCATION OBJECT CODE LINE     SOURCE LINE

  10FC 0F             4557              DEFB KBD_NULL         ; NULL ENTRY
  10FD 02             4558              DEFB 2                ; '2'
  10FE 08             4559              DEFB 11               ; RESET
  10FF 07             4560              DEFB 7                ; '7'
  1100 0F             4561              DEFB KBD_NULL         ; NULL ENTRY
  1101 05             4562              DEFB 5                ; '5'
  1102 04             4563              DEFB 4                ; '4'
  1103 08             4564              DEFB 8                ; '8'
  1104 0F             4565              DEFB KBD_NULL         ; NULL ENTRY
                      4566
                      4567 ******************************SUBROUTINES**********************
                      4568
                      4569                   GLB            CONTROLLER_INIT
  1105                4570 CONTROLLER_INIT                  ;INITIALIZE CONTROLLER TO STROBE RESET
  1105 D3C0           4571              OUT  [STRB_RST_PORT],A
  1107 AF             4572              XOR  A
  1108 DD2A8008       4573              LD   IX,[CONTROLLER_MAP]
  110C DD23           4574              INC  IX
  110E DD23           4575              INC  IX
  1110 FD2173D7       4576              LD   IY,DBNCE_BUFF
  1114 060A           4577              LD   B,NUM_DEV*2
                      4578 *  CLEAR CONTROLLER MEMORY AND DEBOUNCE STATUS BUFFER
  1116                4579 CINIT1
  1116 DD7700         4580              LD   [IX+0],A
  1119 DD23           4581              INC  IX
  111B FD7700         4582              LD   [IY+0],A
  111E FD23           4583              INC  IY
  1120 FD7700         4584              LD   [IY+0],A
  1123 FD23           4585              INC  IY
  1125 05             4586              DEC  B
  1126 20EE           4587              JR   NZ,CINIT1
                      4588 * CLEAR REMAINING VARIABLES
  1128 3273EB         4589              LD   [SPIN_SW0_CT],A
  112B 3273EC         4590              LD   [SPIN_SW1_CT],A
  112E 3273EE         4591              LD   [S0_C0],A
  1131 3273EF         4592              LD   [S0_C1],A
  1134 3273F0         4593              LD   [S1_C0],A
  1137 3273F1         4594              LD   [S1_C1],A
  113A C9             4595              RET
                      4596
                      4597
                      4598
  113B 00             4599 DELAY        NOP                        ;DELAY AFTER STROBE, BEFORE READ
  113C C9             4600              RET
                      4601 *
                      4602 *  CONTROLLER READ ROUTINE
                      4603 *                 INPUT:
                      4604 *                      H - CONTROLLER NUMBER
                      4605 *                 OUTPUT:
                      4606 *                      A - RAW DATA
                      4607 *
                      4608
  113D                4609 CONT_READ
  113D 7C             4610              LD   A,H
  113E FE00           4611              CP   CONTROLLER 0              ;IF CONTROLLER<>0
  1140 2004           4612              JR   NZ,CONT_READ1             ;THEN READ PLAYER 1
  1142 DBFC           4613              IN   A,[CTRL_0_PORT]           ;ELSE READ PLAYER 0
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

  1144 1802        4614                JR   CONT_READX
  1146             4615 CONT_READ1
  1146 DBFF        4616                IN   A,[CTRL_1_PORT]
  1148             4617 CONT_READX
  1148 2F          4618                CPL                              ;COMPLEMENT DATA SO 0=NO DATA
  1149 C9          4619                RET                                                      ;1= DATA
                   4620
                   4621
                   4622 *
                   4623 *  CONTROLLER SCANNER ROUTINE
                   4624 *
                   4625                GLB              CONT_SCAN
  114A             4626 CONT_SCAN
  114A DBFC        4627                IN   A,[CTRL_0_PORT]         ;READ SEGMENT 0, BOTH PLAYERS
  114C 2F          4628                CPL
  114D 3273EE      4629                LD   [S0_C0],A
  1150 DBFF        4630                IN   A,[CTRL_1_PORT]
  1152 2F          4631                CPL
  1153 3273EF      4632                LD   [S0_C1],A
  1156 D380        4633                OUT  [STRB_SET_PORT],A       ;STROBE SEGMENT 1
  1158             4634                DELAY_10                     ;WAIT 10 MICROSECS
  1158 CD1138         +                CALL DELAY
  115B DBFC        4635                IN   A,[CTRL_0_PORT]         ;READ SEGMENT 1, BOTH PLAYERS
  115D 2F          4636                CPL
  115E 3273F0      4637                LD   [S1_C0],A
  1161 DBFF        4638                IN   A,[CTRL_1_PORT]
  1163 2F          4639                CPL
  1164 3273F1      4640                LD   [S1_C1],A
  1167 D3C0        4641                OUT  [STRB_RST_PORT],A       ;RESET TO SEGMENT 0
  1169 C9          4642                RET
                   4643
                   4644
                   4645 *
                   4646 *              UPDATE SPINNER SWITCH ROUTINE                   *
                   4647 *
                   4648 *
                   4649
                   4650
                   4651
                   4652                GLB              UPDATE_SPINNER_
  116A             4653 UPDATE_SPINNER_
  116A DBFC        4654                IN   A,[CTRL_0_PORT]         ;GET DATA
  116C 2173EB      4655                LD   HL,SPIN_SW0_CT          ;ADDRESS OF SPINNER 0 COUNT
  116F CB67        4656                BIT  4,A                     ;IF INT BIT SET
  1171 2008        4657                JR   NZ,UPDATE_S1            ;THEN SPINNER 1
                   4658 *  CHECK DIRECTION                          ; ELSE SPINNER 0
  1173 CB6F        4659                BIT  5,A                     ; IF BIT 5 IS  SET
  1175 2003        4660                JR   NZ,UPDATE_R0            ; THEN GOING RIGHT
                   4661                                             ; ELSE LEFT
  1177 35          4662                DEC  [HL]                    ; DECREMENT SPINNER COUNTER
  1178 1801        4663                JR   UPDATE_S1               ; GO CHECK SPINNER 1
                   4664 ***  RIGHT SPINNER SWITCH
  117A             4665 UPDATE_R0
  117A 34          4666                INC  [HL]                    ;RIGHT, INCREMENT COUNTER
                   4667 * CHECK SPINNER 1
  117B DBFF        4668 UPDATE_S1   IN  A,[CTRL_1_PORT]         ;LOOK AT SPINNER 1 DATA
  117D CB67        4669                BIT  4,A                     ;IF INT BIT  SET
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

   117F 2009         4670             JR   NZ,UPDATE_SPINX         ;THEN NOT SPINNER 1
   1181 23           4671             INC  HL                      ;ELSE SPINNER 1, BUMP HL
                     4672 *  CHECK DIRECTION
   1182 CB6F         4673             BIT  5,A                     ; IF BIT 5 IS SET
   1184 2003         4674             JR   NZ,UPDATE_R1            ; THEN GOING RIGHT
                     4675                                          ; ELSE LEFT
   1186 35           4676             DEC  [HL]                    ; DECREMENT SPINNER COUNTER
   1187 1801         4677             JR   UPDATE_SPINX
                     4678 ***  RIGHT SPINNER SWITCH
   1189              4679 UPDATE_R1
   1189 34           4680             INC  [HL]                    ;RIGHT, INCREMENT COUNTER
   118A              4681 UPDATE_SPINX
   118A C9           4682             RET
                     4683
                     4684
                     4685
                     4686 **************  DECODER ROUTINE  ************************
                     4687 *  THIS ROUTINE RETURNS DECODED RAW, UNDEBOUNCED DATA              *
                     4688 *  AND MAY OR MAY NOT BE REQUIRED BY O/S                          *
                     4689 *
                     4690 *           INPUT:
                     4691 *                H - CONTROLLER NUMBER
                     4692 *                L - SEGMENT NUMBER
                     4693 *           OUTPUT:
                     4694 *                              SEGMENT 0      SEGMENT 1
                     4695 *                H - BYTE 1     FIRE            ARM
                     4696 *                L - BYTE 2     JOYSTK          KBD
                     4697 *                E - BYTE 3     SPINNER
                     4698 *
                     4699 *
                     4700                 GLB          DECODER_
   118B              4701 DECODER_
   118B 7D           4702             LD   A,L
   118C FE01         4703             CP   STROBE_SET              ;IF L=1 THEN DECODE SEGMENT 1
   118E 281A         4704             JR   Z,DEC_SEG1
                     4705 *
                     4706 *  SEGMENT 0 (FIRE BUTTON, JOYSTICK)
                     4707 *  RETURN H=FIRE BUTTON, L=JOYSTICK, E=SPINNER
                     4708 *
                     4709 * DO SPINNER FIRST
   1190 0173EB       4710             LD   BC,SPIN_SW0_CT
   1193 7C           4711             LD   A,H
   1194 FE00         4712             CP   CONTROLLER_0            ;IF PLAYER=0 THEN GO DECODE
   1196 2801         4713             JR   Z,DEC_PLYR
   1198 03           4714             INC  BC                      ;ELSE INCREMENT BC TO SPINNER 1
   1199 0A           4715 DEC_PLYR    LD   A,[BC]                  ;GET SPINNER SWITCH COUNT
   119A 5F           4716             LD   E,A                     ;RETURN IT IN E
   119B AF           4717             XOR  A
   119C 02           4718             LD   [BC],A                  ;CLEAR OUT SPINNER SWITCH COUNT
                     4719
   119D CD1130       4720             CALL CONT_READ               ;GET OTHER DEVICE DATA FOR PLAYER
   11A0 57           4721             LD   D,A                     ;SAVE IT
   11A1 E60F         4722             AND  JOY_MASK                ;MASK OUT JOYSTICK DATA
   11A3 6F           4723             LD   L,A                     ;RETURN IT IN L
                     4724
   11A4 7A           4725             LD   A,D                     ;RESTORE DATA
   11A5 E640         4726             AND  FIRE_MASK               ;MASK OUT FIRE BUTTON DATA
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

  11A7 67                       LD   H,A                      ;RETURN IT IN H
  11A8 1816                     JR   DECODERX
                       *
                       * SEGMENT 1 (ARM BUTTON, KEYBOARD)
                       *  RETURN H=ARM BUTTON, L=KEYBOARD
                       *
  11AA                 DEC_SEG1
  11AA D380                     OUT  [STRB_SET_PORT],A        ;STROBE SEGMENT 1
  11AC CD113D                   CALL CONT_READ                ;READ SEGMENT 1 PLAYER DATA
  11AF 57                       LD   D,A                      ;SAVE IT
  11B0 D3C0                     OUT  [STRB_RST_PORT],A        ;RESET BACK TO SEGMENT 0
  11B2 E60F                     AND  KBD_MASK                 ;MASK OUT KBD DATA
  11B4 2110F5                   LD   HL,DEC_KBD_TBL           ;GET DECODER TABLE ADDRESS
  11B7 0600                     LD   B,0
  11B9 4F                       LD   C,A
  11BA 09                       ADD  HL,BC                    ;COMPUTE OFFSET
  11BB 6E                       LD   L,[HL]                   ;RETURN KBD DATA IN L

  11BC 7A                       LD   A,D                      ;RESTORE DATA
  11BD E640                     AND  ARM_MASK                 ;MASK OUT ARM BUTTON DATA
  11BF 67                       LD   H,A                      ;RETURN IT IN H
  11C0                 DECODERX
  11C0 C9                       RET




                       *  POLLING ROUTINE FOR ALL DEVICES IN CONTROLLER *
                       *
                       *

                                GLB             POLLER_
  11C1                 POLLER_
  11C1 CD114A                   CALL CONT_SCAN                ;GO SCAN ALL THE DATA FIRST
  11C4 FD217307                 LD   IY,DBNCE_BUFF            ;DEBOUNCE BUFFER POINTER
  11C8 DD2A8008                 LD   IX,[CONTROLLER_MAP]      ;CONTROLLER MEMORY POINTER
  11CC DDE5                     PUSH IX
  11CE DD7E00                   LD   A,[IX+0]         ;GET PLAYER 0 STATUS
  11D1 CB7F                     BIT  7,A              ;IF PLAYER 0 NOT ACTIVE
  11D3 281E                     JR   Z,CHK_PLYR_1     ;THEN CHECK PLAYER 1
                       *  PLAYER 0 IS ACTIVE          ;ELSE
  11D5 47                       LD   B,A              ;SAVE STATUS
  11D6 110002                   LD   DE,PLYR_0        ;COMPUTE ADDRESS OF PLAYER_0
  11D9 DD19                     ADD  IX,DE            ;CONTROLLER_MEMORY
  11DB E607                     AND  SEG_0            ;IF SEGMENT 0 IS NOT ACTIVE
  11DD 2809                     JR   Z,CHK_SEG_01     ;THEN CHECK SEGMENT 1
                       *  SEGMENT 0 ACTIVE            ;ELSE
  11DF 3A73EE                   LD   A,[S0_C0]
  11E2 2173EB                   LD   HL,SPIN_SW0_CT
  11E5 CD1220                   CALL DECODE_0         ;DECODE DATA FOR SEGMENT 0
  11E8                 CHK_SEG_01
  11E8 78                       LD   A,B              ;RESTORE PLAYER 0 STATUS
  11E9 E618                     AND  SEG_1            ;IF SEGMENT 1 IS NOT ACTIVE
  11EB 2806                     JR   Z,CHK_PLYR_1     ;THEN CHECK PLAYER 1
                       *  SEGMENT 1 IS ACTIVE         ;ELSE
  11ED 3A73F0                   LD   A,[S1_C0]
  11F0 CD123F                   CALL DECODE_1         ;DECODE DATA FOR SEGMENT 1
```

```
CATION OBJECT CODE LINE     SOURCE LINE

 11F3               4784 CHK_PLYR_1
 11F3 DDE1          4785            POP  IX
 11F5 DD7E01        4786            LD   A,[IX+1]         ;GET PLAYER 1 STATUS
 11F8 CB7F          4787            BIT  7,A              ;IF PLAYER 1 IS NOT ACTIVE
 11FA 2823          4788            JR   Z,POLLER_X        ;THEN EXIT, ALL DONE
                    4789 *  PLAYER 1 IS ACTIVE
 11FC 47            4790            LD   B,A              ;SAVE PLAYER 1 STATUS
 11FD 11000A        4791            LD   DE,2*NUM_DEV     ;COMPUTE ADDRESS OF DEBOUNCE BUFFER
 1200 FD19          4792            ADD  IY,DE            ;FOR PLAYER 1
 1202 110007        4793            LD   DE,PLYR_1        ;COMPUTE ADDRESS OF CONTROLLER_MEMORY
 1205 DD19          4794            ADD  IX,DE            ;FOR PLAYER 1
 1207 E607          4795            AND  SEG_0            ;IF SEGMENT 0 IS NOT ACTIVE
 1209 2809          4796            JR   Z,CHK_SEG_11     ;THEN CHECK SEGMENT 1
                    4797 *  SEGMENT 0 IS ACTIVE          ;ELSE
 120B 3A73EF        4798            LD   A,[S0_C1]
 120E 2173EC        4799            LD   HL,SPIN_SW1_CT
 1211 CD1220        4800            CALL DECODE_0         ;DECODE DATA FOR SEGMENT 0
 1214               4801 CHK_SEG_11
 1214 78            4802            LD   A,B              ;RESTORE STATUS FOR PLAYER 1
 1215 E618          4803            AND  SEG_1            ;IF SEGMENT 1 IS NOT ACTIVE
 1217 2806          4804            JR   Z,POLLER_X        ;THEN EXIT, ALL DONE
                    4805 *  SEGMENT 1 IS ACTIVE          ;ELSE
 1219 3A73F1        4806            LD   A,[S1_C1]
 121C CD123F        4807            CALL DECODE_1         ;DECODE DATA FOR SEGMENT 1
 121F               4808 POLLER_X
 121F C9            4809            RET
                    4810
                    4811 *  DECODER ROUTINE FOR SEGMENT 0
                    4812 *
                    4813 *          INPUT:
                    4814 *               A - DATA
                    4815 *               B - DEVICE STATUS BYTE FOR CURRENT PLAYER
                    4816 *              HL - ADDRESS OF SPINNER DATA
                    4817 *              IX - POINTER TO CONTROLLER MEMORY
                    4818 *              IY - POINTER TO DEBOUNCE STATUS BUFFER
                    4819 *
 1220               4820 DECODE_0
 1220 4F            4821            LD   C,A              ;SAVE DATA
 1221 CB48          4822            BIT  JOY,B            ;IF JOYSTICK NOT ACTIVE
 1223 2804          4823            JR   Z,DEC_FIRE       ;THEN CHECK FIRE BUTTON
                    4824 *  JOYSTICK ACTIVE
 1225 CD12B9        4825            CALL JOY_DBNCE        ;DEBOUNCE JOYSTICK DATA
 1228 79            4826            LD   A,C
 1229               4827 DEC_FIRE
 1229 CB40          4828            BIT  FIRE,B           ;IF FIRE BUTTON NOT ACTIVE
 122B 2804          4829            JR   Z,DEC_SPNR       ;THEN CHECK SPINNER
                    4830 *  FIRE BUTTON ACTIVE           ;ELSE
 122D CD1289        4831            CALL FIRE_DBNCE       ;DEBOUNCE FIRE BUTTON
 1230 79            4832            LD   A,C
 1231               4833 DEC_SPNR
 1231 CB50          4834            BIT  SPIN,B           ;IF SPINNER NOT ACTIVE
 1233 2809          4835            JR   Z,DECODE_0X      ;THEN EXIT DECODER
                    4836 *  SPINNER ACTIVE
 1235 7E            4837            LD   A,[HL]           ;SAVE SPINNER COUNT
 1236 DD8602        4838            ADD  A,[IX+SPIN]
 1239 DD7702        4839            LD   [IX+SPIN],A      ;IN CONTROLLER MEMORY
 123C AF            4840            XOR  A
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

   123D 77           4841                LD   [HL],A           ;CLEAR COUNTER
   123E              4842 DECODE_0X
   123E C9           4843                RET
                     4844
                     4845
                     4846 *  DECODER ROUTINE FOR SEGMENT 1
                     4847 *
                     4848 *              INPUT:
                     4849 *                   A - DATA
                     4850 *                   B - DEVICE STATUS BYTE FOR CURRENT PLAYER
                     4851 *                  IX - POINTER TO CONTROLLER MEMORY
                     4852 *                  IY - POINTER TO DEBOUNCE STATUS BUFFER
                     4853 *
   123F              4854 DECODE_1
   123F 4F           4855                LD   C,A              ;SAVE DATA
   1240 CB58         4856                BIT  ARM,B            ;IF ARM BUTTON NOT ACTIVE
   1242 2804         4857                JR   Z,DEC_KBD        ;THEN CHECK KEYBOARD
                     4858 *  ARM BUTTON ACTIVE                 ;ELSE
   1244 CD12E9       4859                CALL ARM_DBNCE        ;DEBOUNCE ARM BUTTON
   1247 79           4860                LD   A,C
   1248              4861 DEC_KBD
   1248 CB60         4862                BIT  KBD,B            ;IF KEYBOARD NOT ACTIVE
   124A 2803         4863                JR   Z,DECODE_1X      ;THEN EXIT DECODER
                     4864 * KBD ACTIVE
   124C CD1250       4865                CALL KBD_DBNCE        ;DEBOUNCE KEYBOARD  -
   124F              4866 DECODE_1X
   124F C9           4867                RET
                     4868
                     4869
                     4870
                     4871 *  KEYBOARD DEBOUNCE ROUTINE  *
                     4872 *
                     4873 *              INPUT:
                     4874 *                   A - RAW DATA
                     4875 *                  IX - CONTROLLER MEMORY POINTER
                     4876 *                  IY - DEBOUNCE STATE BUFFER
                     4877 *
   1250              4878 KBD_DBNCE
   1250 C5           4879                PUSH BC
   1251 D5           4880                PUSH DE
   1252 E5           4881                PUSH HL
   1253 E60F         4882                AND  KBD_MASK         ;MASK OUT VALID DATA
   1255 5F           4883                LD   E,A              ;SAVE IT
   1256 FD4608       4884                LD   B,[IY+KBD_OLD]   ;GET OLD DATA AND CURREN STATE
   1259 FD7E09       4885                LD   A,[IY+KBD_STATE]
   125C FE00         4886                CP   0                ;IF STATE <> 0
   125E 201A         4887                JR   NZ,KBD_ST1       ;THEN MUST BE STATE 1
                     4888 *  STATE = 0                         ;ELSE
   1260 7B           4889                LD   A,E              ;GET CURRENT DATA
   1261 B8           4890                CP   B                ;IF OLD=NEW
   1262 2805         4891                JR   Z,KBD_REG        ;THEN SAW DATA TWICE IN SEQUENCE
   1264 FD7308       4892                LD   [IY+KBD_OLD],E   ;ELSE FIRST TIME, SAVE CURRENT DATA
   1267 181C         4893                JR   KBD_EXIT
                     4894 *  SAW DATA TWICE IN SEQUENCE
   1269              4895 KBD_REG
   1269 3E01         4896                LD   A,1              ;SET STATE=1
   126B FD7709       4897                LD   [IY+KBD_STATE],A
```

```
)CATION OBJECT CODE LINE     SOURCE LINE

                    4898 *  DECODE KEYBOARD DATA
 126E 2110F5        4899            LD   HL,DEC_KBD_TBL  ;DECODE TABLE ADDRESS
 1271 1600          4900            LD   D,0             ;D/E RAW DATA
 1273 19            4901            ADD  HL,DE           ;COMPUTE ADDRESS INTO TABLE
 1274 7E            4902            LD   A,[HL]          ;DO TABLE LOOKUP
 1275 DD7704        4903            LD   [IX+KBD],A      ;SAVE IN CONTROLLER MEMORY @KBD
 1278 1808          4904            JR   KBD_EXIT
                    4905 *  STATE = 1
 127A               4906 KBD_ST1
 127A 7B            4907            LD   A,E             ;GET CURRENT DATA
 127B B8            4908            CP   B               ;IF OLD=NEW
 127C 2807          4909            JR   Z,KBD_EXIT      ;NO CHANGE IN STATE
 127E FD7308        4910            LD   [IY+KBD_OLD],E  ;ELSE SAVE CURRENT DATA
 1281 AF            4911            XOR  A               ;SET STATE=0
 1282 FD7709        4912            LD   [IY+KBD_STATE],A
 1285               4913 KBD_EXIT
 1285 E1            4914            POP  HL
 1286 D1            4915            POP  DE
 1287 C1            4916            POP  BC
 1288 C9            4917            RET
                    4918
                    4919 *  FIRE BUTTON DEBOUNCE ROUTINE  *
                    4920 *
                    4921 *          INPUT:
                    4922 *              A - RAW DATA
                    4923 *             IX - CONTROLLER MEMORY POINTER
                    4924 *             IY - DEBOUNCE STATE BUFFER
                    4925 *
 1289               4926 FIRE_DBNCE
 1289 C5            4927            PUSH BC
 128A D5            4928            PUSH DE
 128B E640          4929            AND  FIRE_MASK        ;MASK OUT VALID DATA
 128D 5F            4930            LD   E,A             ;SAVE IT
 128E FD4600        4931            LD   B,[IY+FIRE_OLD]  ;GET OLD DATA AND CURRENT STATE
 1291 FD7E01        4932            LD   A,[IY+FIRE_STATE]
 1294 FE00          4933            CP   0               ;IF STATE <> 0
 1296 2013          4934            JR   NZ,FIRE_ST1      ;THEN MUST BE STATE 1
                    4935 *  STATE = 0                    ;ELSE
 1298 7B            4936            LD   A,E             ;GET CURRENT DATA
 1299 B8            4937            CP   B               ;IF OLD=NEW
 129A 2805          4938            JR   Z,FIRE_REG       ;THEN SAW DATA TWICE IN SEQUENCE
 129C FD7300        4939            LD   [IY+FIRE_OLD],E  ;ELSE FIRST TIME, SAVE CURRENT DATA
 129F 1815          4940            JR   FIRE_EXIT
                    4941 *  SAW DATA TWICE IN SEQUENCE
 12A1               4942 FIRE_REG
 12A1 3E01          4943            LD   A,1             ;SET STATE=1
 12A3 FD7701        4944            LD   [IY+FIRE_STATE],A
 12A6 DD7300        4945            LD   [IX+FIRE],E      ;SAVE IN CONTROLLER MEMORY @FIRE
 12A9 1808          4946            JR   FIRE_EXIT
                    4947 *  STATE = 1
 12AB               4948 FIRE_ST1
 12AB 7B            4949            LD   A,E             ;GET CURRENT DATA
 12AC B8            4950            CP   B               ;IF OLD=NEW
 12AD 2807          4951            JR   Z,FIRE EXIT      ;NO CHANGE IN STATE
 12AF FD7300        4952            LD   [IY+FIRE OLD],E  ;ELSE SAVE CURRENT DATA
 12B2                 53                 A                  1 ST.
 12B3 FD7701        4954            LD   [IY+FIRE STATE],A
```

```
LOCATION OBJECT CODE LINE      SOURCE LINE

   12B6               4955 FIRE_EXIT
   12B6 D1            4956            POP  DE
   12B7 C1            4957            POP  BC
   12B8 C9            4958            RET
                      4959
                      4960 *  JOYSTICK DEBOUNCE ROUTINE  *
                      4961 *
                      4962 *           INPUT:
                      4963 *                A - RAW DATA
                      4964 *               IX - CONTROLLER MEMORY POINTER
                      4965 *               IY - DEBOUNCE STATE BUFFER
                      4966 *
   12B9               4967 JOY_DBNCE
   12B9 C5            4968            PUSH BC
   12BA D5            4969            PUSH DE
   12BB E60F          4970            AND  JOY_MASK          ;MASK OUT VALID DATA
   12BD 5F            4971            LD   E,A               ;SAVE IT
   12BE FD4602        4972            LD   B,[IY+JOY_OLD]    ;GET OLD DATA AND CURRENT STATE
   12C1 FD7E03        4973            LD   A,[IY+JOY_STATE]
   12C4 FE00          4974            CP   0                 ;IF STATE <> 0
   12C6 2013          4975            JR   NZ,JOY_ST1        ;THEN MUST BE STATE 1
                      4976 *  STATE = 0                      ;ELSE
   12C8 7B            4977            LD   A,E               ;GET CURRENT DATA
   12C9 B8            4978            CP   B                 ;IF OLD=NEW
   12CA 2805          4979            JR   Z,JOY_REG         ;THEN SAW DATA TWICE IN SEQUENCE
   12CC FD7302        4980            LD   [IY+JOY_OLD],E    ;ELSE FIRST TIME, SAVE CURRENT DATA
   12CF 1815          4981            JR   JOY_EXIT
                      4982 *  SAW DATA TWICE IN SEQUENCE
   12D1               4983 JOY_REG
   12D1 3E01          4984            LD   A,1               ;SET STATE=1
   12D3 FD7703        4985            LD   [IY+JOY_STATE],A
   12D6 DD7301        4986            LD   [IX+JOY],E        ;SAVE IN CONTROLLER MEMORY @JOY
   12D9 1808          4987            JR   JOY_EXIT
                      4988 *  STATE = 1
   12DB               4989 JOY_ST1
   12DB 7B            4990            LD   A,E               ;GET CURRENT DATA
   12DC B8            4991            CP   B                 ;IF OLD=NEW
   12DD 2807          4992            JR   Z,JOY_EXIT        ;NO CHANGE IN STATE
   12DF FD7302        4993            LD   [IY+JOY_OLD],E    ;ELSE SAVE CURRENT DATA
   12E2 AF            4994            XOR  A                 ;SET STATE=0
   12E3 FD7703        4995            LD   [IY+JOY_STATE],A
   12E6               4996 JOY_EXIT
   12E6 D1            4997            POP  DE
   12E7 C1            4998            POP  BC
   12E8 C9            4999            RET
                      5000
                      5001 *  ARM BUTTON DEBOUNCE ROUTINE  *
                      5002 *
                      5003 *           INPUT:
                      5004 *                A - RAW DATA
                      5005 *               IX - CONTROLLER MEMORY POINTER
                      5006 *               IY - DEBOUNCE STATE BUFFER
                      5007 *
   12E9               5008 ARM_DBNCE
   12E9 C5            5009            PUSH BC
   12EA D5            5010            PUSH DE
   12EB E640          5011            AND  ARM_MASK          ;MASK OUT VALID DATA
```

```
OCATION OBJECT CODE LINE     SOURCE LINE

  12ED 5F           5012             LD   E,A             ;SAVE IT
  12EE FD4606       5013             LD   B,[IY+ARM_OLD]  ;GET OLD DATA AND CURRENT STATE
  12F1 FD7E07       5014             LD   A,[IY+ARM_STATE]
  12F4 FE00         5015             CP   0               ;IF STATE <> 0
  12F6 2013         5016             JR   NZ,ARM_ST1      ;THEN MUST BE STATE 1
                    5017 *  STATE = 0                     ;ELSE
  12F8 7B           5018             LD   A,E             ;GET CURRENT DATA
  12F9 B8           5019             CP   B               ;IF OLD=NEW
  12FA 2805         5020             JR   Z,ARM_REG       ;THEN SAW DATA TWICE IN SEQUENCE
  12FC FD7306       5021             LD   [IY+ARM_OLD],E  ;ELSE FIRST TIME, SAVE CURRENT DATA
  12FF 1815         5022             JR   ARM_EXIT
                    5023 *  SAW DATA TWICE IN SEQUENCE
  1301              5024 ARM_REG
  1301 3E01         5025             LD   A,1             ;SET STATE=1
  1303 FD7707       5026             LD   [IY+ARM_STATE],A
  1306 DD7303       5027             LD   [IX+ARM],E      ;SAVE IN CONTROLLER MEMORY @ARM
  1309 180B         5028             JR   ARM_EXIT
                    5029 *  STATE = 1
  130B              5030 ARM_ST1
  130B 7B           5031             LD   A,E             ;GET CURRENT DATA
  130C B8           5032             CP   B               ;IF OLD=NEW
  130D 2807         5033             JR   Z,ARM_EXIT      ;NO CHANGE IN STATE
  130F FD7306       5034             LD   [IY+ARM_OLD],E  ;ELSE SAVE CURRENT DATA
  1312 AF           5035             XOR  A               ;SET STATE=0
  1313 FD7707       5036             LD   [IY+ARM_STATE],A
  1316              5037 ARM_EXIT
  1316 D1           5038             POP  DE
  1317 C1           5039             POP  BC
  1318 C9           5040             RET
                    5041
                    5042
                    5043 ;                EXT             CONTROLLER_MAP
                    5044 * THIS IS AN EXTERNAL POINTER ( DEFINED IN THE CARTRIDGE) TO THE
                    5045 * CARTRIDGE PROGRAMMER'S CONTROLLER MAP AREA.
                    5046
                    5047 * THE CARTRIDGE PROGRAMMER IS RESPONSIBLE FOR MAINTAINING THIS AREA.
                    5048
                    5049                  DATA
  73D7              5050 DBNCE_BUFF       DEFS            NUM_DEV*4
  73EB              5051 SPIN_SW0_CT      DEFS            1
  73EC              5052 SPIN_SW1_CT      DEFS            1
  73ED              5053 STROBE_FLG       DEFS            1
  73EE              5054 S0_C0            DEFS            1
  73EF              5055 S0_C1            DEFS            1
  73F0              5056 S1_C0            DEFS            1
  73F1              5057 S1_C1            DEFS            1
                    5058  PROG
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

               5060
               5061 ************************ EXTERNAL SYMBOLS ***************************
               5062
               5063 * EXTERNAL ROUTINES FROM OS
               5064
               5065                  ;EXT INIT_TABLE
               5066                  ;EXT PUT_VRAM
               5067                  ;EXT GAME_NAME
               5068                  ;EXT WRITE_REGISTER
               5069                  ;EXT READ_REGISTER
               5070                  ;EXT WRITE_VRAM
               5071                  ;EXT START_GAME
               5072
               5073 ************************** DEFINITIONS ***************************
               5074
      <00BE>   5075 MODE_0_PORT      EQU 0BEH
      <00BF>   5076 MODE_1_PORT      EQU 0BFH
               5077
               5078 **************************** EXPORTS *****************************
               5079
               5080                  GLB ASCII_TBL                  ;POINTER TO UPPERCASE ASCII GENERATORS
               5081                  GLB NUMBER_TBL                 ;POINTER TO 0-9 GENERATORS
               5082                  GLB DISPLAY_LOGO               ;DISPLAY COLECOVISION LOGO
               5083                  GLB LOAD_ASCII_                ;LOAD PATTERN_GEN_TABLE WITH FULL ASCII SET
               5084                  GLB FILL_VRAM_                 ;FILL VRAM WITH A VALUE
               5085                  GLB MODE_1_                    ;SET UP MODE_1 GRAPHICS
               5086
               5087 *************************  DESCRIPTION *****************************
               5088 *
               5089 *  DISPLAY_LOGO  DISPLAYS THE COLECO LOGO SCREEN WITH COLECOVISION
               5090 *                ON A BLACK BACKGROUND.  THE GAME TITLE, MANUFACTURER,
               5091 *                AND COPYRIGHT YEAR ARE OBTAINED FROM THE CARTRIDGE
               5092 *                AND OVERLAYED ONTO THE LOGO SCREEN.  THE LOGO IS THEN
               5093 *                DISPLAYED FOR 10 SECONDS AFTER WHICH TIME A JUMP TO
               5094 *                THE GAME START ADDRESS IS EXECUTED.
               5095 *
               5096 *                IF NO CARTRIDGE IS PRESENT A DEFAULT MESSAGE IS
               5097 *                DISPLAYED, INSTRUCTING THE OPERATOR TO:
               5098 *
               5099 *                       "TURN GAME OFF"
               5100 *                       "BEFORE INSERTING CARTRIDGE"
               5101 *                       "OR EXPANSION MODULE."
               5102 *                       "[COPYRIGHT SYMBOL] 1982 COLECO"
               5103 *                THIS MESSAGE IS DISPLAYED FOR 60 SECONDS, THE SCREEN
               5104 *                IS THEN BLANKED AND FINALLY A SOFT HALT (JP $) IS
               5105 *                EXECUTED LOCKING UP THE PROGRAM UNTIL THE UNIT IS
               5106 *                RESET.
               5107 *
               5108 *                DISPLAY_LOGO EXITS WITH THE VDP IN MODE 1, THE SCREEN
               5109 *                BLANKED, AND THE ASCII CHARACTER SET IN VRAM.
               5110 *                THE MEMORY MAP IS AS FOLLOWS:
               5111 *
               5112 *                          VDP MEMORY MAP
               5113 *                3800H-3FFFH  SPRITE GENERATOR TABLE
               5114 *                2000H-37FFH  PATTERN COLOR TABLE
               5115 *                1B00H-1B7FH  SPRITE ATTRIBUTE TABLE
               5116 *                1800H-1AFFH  PATTERN NAME TABLE
```

```
LOCATION OBJECT CODE LINE      SOURCE LINE

               5117 *                   0000H-17FFH  PATTERN GENERATOR TABLE
               5118 *
               5119 ***********************************************************************
               5120
               5121 *************************** DISPLAY LOGO ****************************
               5122
               5123                     PROG
               5124
               5125 *  FILL VRAM WITH 0'S
 1319 210000   5126 DISPLAY_LOGO        LD HL,0
 131C 114000   5127                     LD DE,16384
 131F 3E00     5128                     LD A,0
 1321 CD18D4   5129                     CALL FILL_VRAM_
               5130
               5131 *  SET UP VDP WITH MODE_1
 1324 CD18E9   5132                     CALL MODE_1_
               5133
               5134 ****************** WRITE OUT PATTERN GEN TABLE *****************
               5135
               5136 *  WRITE OUT ASCII GENERATOR TABLES
 1327 CD1927   5137                     CALL LOAD_ASCII_
               5138
               5139 *  WRITE OUT GRAPHICS GENERATORS
 132A 2118A3   5140                     LD HL,OBJ_TABLE          ;POINT TO TABLE OF PTN GEN NUMBERS
 132D 110060   5141                     LD DE,60H                ;ITEM LOCATION IN VRAM PATTERN_GEN_TABLE
 1330 E5       5142 WRITE_LOOP          PUSH HL                  ;SAVE LOCATION OF CURRENT CONSTRUCTION
 1331 D5       5143                     PUSH DE                  ;SAVE VRAM ITEM #
               5144 *  CALCULATE GENERATOR LOCATION
 1332 7E       5145                     LD A,[HL]
 1333 FEFF     5146                     CP 0FFH                  ;HAVE WE PROCESSED ALL GENERATORS?
 1335 281B     5147                     JR Z,DONE_LOGO           ; YES.  WE'RE ALL DONE
 1337 47       5148                     LD B,A                   ; NO. B=NUMBER FROM OBJ_TABLES
 1338 04       5149                     INC B
 1339 2114C3   5150                     LD HL,LOGO_GEN           ;POINT TO ROM GENERATOR TABLE
 133C 110008   5151                     LD DE,8                  ;WE'RE GOING TO ADD 8 FOR EVERY
 133F 1015     5152 ADDR_ADJ            DJNZ ADD_8               ;GENERATOR INTO THE ROM GEN TBL
               5153
               5154 *  DONE ADJUSTING ROM GENERATOR ADDRESS
               5155
 1341 D1       5156                     POP DE                   ;RESTORE ITEM # IN VRAM
 1342 D5       5157                     PUSH DE                  ;SAVE IT
 1343 FD210001 5158                     LD IY,1                  ;NUMBER OF GENERATORS
 1347 3E03     5159                     LD A,3                   ;PATTERN_GEN_TABLE CODE
 1349 CD1FBE   5160                     CALL PUT_VRAM            ;HL=ROM ADDRESS
 134C D1       5161                     POP DE                   ;RESTORE ITEM # IN VRAM
 134D E1       5162                     POP HL                   ;RESTORE CONSTRUCTION ADDRESS
 134E 13       5163                     INC DE                   ;SET UP FOR NEXT ITEM
 134F 23       5164                     INC HL
 1350 180E     5165                     JR WRITE_LOOP            ;KEEP GOING UNTIL DONE
               5166
 1352 D1       5167 DONE_LOGO           POP DE                   ;GOT TO POP FOR EVERY PUSH
 1353 E1       5168                     POP HL
 1354 1803     5169                     JR WRITE_NAMES           ;HOP AROUND ADD_8
               5170
 1356 19       5171 ADD_8               ADD HL,DE                ;POINT TO NEXT GENERATOR
 1357 18E6     5172                     JR ADDR_ADJ
               5173
```

```
LOCATION OBJECT CODE LINE     SOURCE LINE

                     ****************** WRITE OUT PATTERN_NAME_TABLE ****************

                     *  WRITE OUT PATTERN_NAME_TABLE
  1359 21144D        WRITE_NAMES     LD HL,LOGO_NAMES          ;WRITE OUT TOP HALF
  135C 110085                        LD DE,133                 ;OF COLCOVISION
  135F FD210016                      LD IY,22
  1363 3E02                          LD A,2
  1365 CD1FBE                        CALL PUT_VRAM

  1368 211463                        LD HL,LOGO_NAMES+22       ;WRITE OUT BOTTOM HALF
  136B 1100A5                        LD DE,165                 ;OF COLECOVISION
  136E FD210016                      LD IY,22
  1372 3E02                          LD A,2
  1374 CD1FBE                        CALL PUT_VRAM


  1377 2114C1                        LD HL,TRADEMARK           ;WRITE OUT TM
  137A 11009B                        LD DE,155
  137D FD210002                      LD IY,2
  1381 3E02                          LD A,2
  1383 CD1FBE                        CALL PUT_VRAM

                     * SET UP DEFAULT COPYRIGHT MESSAGE
  1386 2114B4                        LD HL,LOGO_NAMES+103      ;WRITE OUT c 1982 COLECO
  1389 1102AA                        LD DE,682
  138C FD21000D                      LD IY,13
  1390 3E02                          LD A,2
  1392 CD1FBE                        CALL PUT_VRAM

                     ****************** WRITE OUT COLOR_NAME_TABLE ****************

  1395 21143B                        LD HL,LOGO_COLORS
  1398 110000                        LD DE,0
  139B 3E04                          LD A,4
  139D FD210012                      LD IY,18
  13A1 CD1FBE                        CALL PUT_VRAM

                     ************************** ENABLE DISPLAY *********************

                     * ENABLE DISPLAY
  13A4 0601                          LD B,1
  13A6 0EC0                          LD C,11000000B
  13A8 CD1FD9                        CALL WRITE_REGISTER

                     * CARTRIDGE TEST
  13AB 218000                        LD HL,8000H               ;IF A CARTRIDGE IS PRESENT
  13AE 7E                            LD A,[HL]                 ;LOCATION 8000H WILL = 0AAH
  13AF FEAA                          CP 0AAH                   ;AND 8001H WILL = 55H
  13B1 204C                          JR NZ,NO_CARTRIDGE        ;     NOT PRESENT
  13B3 23                            INC HL
  13B4 7E                            LD A,[HL]
  13B5 FE55                          CP 55H
  13B7 2046                          JR NZ,NO_CARTRIDGE
                                                               ;     NOT PRESENT
                     * CARTRIDGE PRESENT
                     * DISPLAY GAME TITLE
  13B9 218024                        LD HL,GAME_NAME
```

```
OCATION OBJECT CODE LINE     SOURCE LINE

  13BC CD1946     5231                CALL PARSE           ;GET LENGTH OF STRING
  13BF 118024     5232                LD DE,GAME_NAME      ;STARTING LOCATION OF 1ST STRING
  13C2 210201     5233                LD HL,513            ;LOCATION (0_767) TO START PRINTING
  13C5 CD1951     5234                CALL CENTER_PRT      ;PRINT IT
                  5235
                  5236 * DISPLAY COMPANY NAME
  13C8 218024     5237                LD HL,GAME_NAME      ;GET PAST 1ST STRING
  13CB CD1946     5238                CALL PARSE
  13CE 23         5239                INC HL
  13CF 54         5240                LD D,H               ;STARTING LOCATION OF 2ND STRING
  13D0 5D         5241                LD E,L               ;SAVED IN DE
  13D1 CD1946     5242                CALL PARSE           ;GET LENGTH OF STRING
  13D4 2101C1     5243                LD HL,449            ;LOCATION (0_767) TO START PRINTING
  13D7 CD1951     5244                CALL CENTER_PRT      ;PRINT IT
                  5245
                  5246 * CHANGE DATE
  13DA 218024     5247                LD HL,GAME_NAME      ;USE PARSE TO ADVANCE HL TO
  13DD CD1946     5248                CALL PARSE           ;COPYRIGHT YEAR
  13E0 23         5249                INC HL
  13E1 CD1946     5250                CALL PARSE
  13E4 23         5251                INC HL
  13E5 1102AC     5252                LD  DE,684           ;SCREEN LOCATION
  13E8 FD210004   5253                LD IY,4              ;# OF DIGITS
  13EC 3E02       5254                LD A,2
  13EE CD1FBE     5255                CALL PUT_VRAM
                  5256
                  5257 * DISPLAY 10 SECONDS
  13F1 CD1968     5258                CALL DELAY_10
                  5259
                  5260 * TURN OFF DISPLAY
  13F4 0601       5261                LD B,1
  13F6 0E80       5262                LD C,10000000B
  13F8 CD1FD9     5263                CALL WRITE_REGISTER
                  5264
                  5265 * EXIT LOGO
  13FB 2A800A     5266                LD HL,[START_GAME]         ;A CARTRIDGE WAS PRESENT
  13FE E9         5267                JP [HL]                    ;SO JUMP TO IT
                  5268
                  5269 *  WRITE OUT PATTERN_NAME_TABLE
  13FF 211479     5270 NO_CARTRIDGE   LD HL,LOGO_NAMES+44        ;NO CARTRIDGE PRESENT
  1402 1101AA     5271                LD DE,426                  ;DISPLAY DEFAULT
  1405 FD210000   5272                LD IY,13                   ;MESSAGE
  1409 3E02       5273                LD A,2
  140B CD1FBE     5274                CALL PUT_VRAM
                  5275
  140E 211486     5276                LD HL,LOGO_NAMES+57        ;"TURN GAME OFF"
  1411 1101E4     5277                LD DE,484                  ;"TO INSERT CARTRIDGE"
  1414 FD21001A   5278                LD IY,26                   ;"OR EXPANSION MODULE"
  1418 3E02       5279                LD A,2
  141A CD1FBE     5280                CALL PUT_VRAM
                  5281
  141D 2114A0     5282                LD HL,LOGO_NAMES+83
  1420 110227     5283                LD DE,551
  1423 FD210014   5284                LD IY,20
  1427 3E02       5285                LD A,2
  1429 CD1FBE     5286                CALL PUT_VRAM
                  5287
```