

```

%-----
% MORSE_1
% VER 1.0
%
% MORSE DECODER PROGRAM FOR PERSEUS-3 (CPU:MC6802)
%
% SPECIFICATION
%   THIS PROGRAM DECODES MORSE CODES INTO ASCII CODES.
%
%   MORSE SIGNAL INPUT: PARALLEL INTERFACE D0 ('0'= SIGNAL ON, '1'= SIGNAL OFF)
%   DECODED ASCII CHARACTER OUTPUT: SERIAL INTERFACE (RS-232C)
%
%   1. DETERMINATION DASH LENGTH THRESHOLD
%       SAMPLE 16 SET OF THE MIXED SIGNAL OF DOT and DASH,
%       AND DETERMINE 1/16 OF THE LENGTH AS THE DASH LENGTH THRESHOLD.
%       SAMPLING PERIOD = 10 MILLI SECOND
%
%   2. DETECT SIGNAL AND GENERATE INTERMEDIATE CODE
%       INTERMEDIATE CODE: (START BIT='1',DOT='0',DASH='1')
%
% HAND ASSEMBLED
%
% MITSURU YAMADA 10/MAY/2020
%-----
% COPYRIGHT (C) 2020 MITSURU YAMADA. ALL RIGHTS RESERVED.
%-----
% ADDRESS MAPPING:
%   VARIABLES          (ADDRESS $0000 - $0003)
%   STACK              (ADDRESS $0040 - $007F)
%   MAIN PROGRAM       (ADDRESS $1000 - $10C7)
%   CODE CONVERSION TABLE (ADDRESS $1200 - $127F)
%   I/O ROUTINE        (ADDRESS $F820 - $F99A)
%-----
% VARIABLES
%
%   SYMBOL          DATA
%   TSD             $0000          DASH LENGTH THRESHOLD
%   CODE_H          $0002          INTERMEDIATE CODE H BYTE
%   CODE_L          $0003          INTERMEDIATE CODE L BYTE
%
%   PORT           $4000          PARALLEL INTERFACE ADDRESS (R/W)
%   ACIA_STATUS    $A000          ACIA(6850) STATUS REGISTER
%   ACIA_DATA      $A001          ACIA(6850) DATA REGISTER
%-----
%
%
%               ADDRESS(HEX) DATA(HEX)
%-----
% PROGRAM START ADDRESS          MORSE_1          $1000
%
% RESET VECTOR
%   .ORIGIN $FFFE
%               FFFE          10 00
%-----
% MAIN PROGRAM
%   .ORIGIN $1000
%
% MORSE_1      LDS #$007F          1000      8E 00 7F          INITIALIZE STACK POINTER
%              LDX #$1200          1003      CE 12 00
%              STX CODE_H          1006      DF 02          INITIALIZE INTERMEDIATE CODE
%              JSR INIT_ACIA        1008      BD F8 20          INITIALIZE ACIA (SERIAL INTERFACE)
%              JSR DET_TSD          100B      BD 10 30          DETERMINATION TSD
% L12         JSR TIMER_10M         100E      BD 10 C0          TIMER 10 MILLI SECOND
%              LDAA PORT            1011      B6 40 00          INPUT SIGNAL BY PORT
%              STAA PORT            1014      B7 40 00
%              CMPA #$FF            1017      81 FF          IF D0='0' GOTO CONV. INT. CODE
%              BEQ L12              1019      27 F3          IF D0-'1' REPEAT NO SIGNAL LOOP
%              JSR DET_CODE          101B      BD 10 70          CONVERSION TO INT. CODE
%              LDX CODE_H           101E      DE 02          SET POINTER OFFSET BY INT. CODE
%              LDAB X,00            1020      E6 00          CONVERT TO ASCII CODE

```

```

JSR OUT_1CHA      1022 BD F9 90      OUTPUT 1 CHARACTER TO RS-232C
BRA L12          1025 20 E7      RETURN TO NO SIGNAL LOOP

```

%

```

-----
% DETERMINATION DASH LENGTH THRESHOLD (TSD)
  .ORIGIN $1030

```

%

```

DET_TSD      CLR TSD          1030 7F 00 00      CLEAR TSD ACCUMULATION VALUE
              LDAB #$10      1033 C6 10          SET NUMBER OF LOOP TO 16
L02          LDAA PORT      1035 B6 40 00      INPUT SIGNAL
              STAA PORT      1038 B7 40 00
              CMPA #$FF      103B 81 FF          D0='1'?
              BEQ L01        103D 27 12
L03          INC TSD        103F 7C 00 00      TSD VALUE + 1
              JSR TIMER_10MS 1042 BD 10 C0      TIMER 10 MILLI SECOND
              NOP            1045 01
              BRA L02        1046 20 ED          REPEAT ACCUMULATION

```

%

```

L14          LDAA TSD        1048 96 00          1/16 TSD ACCUMULATION VALUE
              LSRA          104A 44
              LSRA          104B 44
              LSRA          104C 44
              LSRA          104D 44
              STAA TSD      104E 97 00
              RTS           1050 39

```

%

```

L01          JSR TIMER_10MS 1051 BD 10 C0      TIMER 10 MILLI SECOND
              LDAA PORT      1054 B6 40 00      INPUT SIGNAL
              STAA PORT      1057 B7 40 00
              CMPA #$FE      105A 81 FE          D0='0'?
              BEQ L13        105C 27 02          SIGNAL DETECTED
              BRA L01        105E 20 F1          IF D0='1' REPEAT OFF SIGNAL DETECTION

```

%

```

L13          DECB          1060 5A              ONLY WHIN SIGNAL CANGES OFF TO ON,
              BNE L03        1061 26 DC          TSD - 1
              BRA L14        1063 20 E3

```

%

```

-----
% DETECT SIGNAL AND GENERATE INTERMEDIATE CODE
  .ORIGIN $1070

```

%

```

DET_CODE     LDAA #$01      1070 86 01          INIT. INTERMEDIATE CODE
              STAA CODE_L    1072 97 03
L10          CLR B          1074 5F              CLEAR LENGTH
L05          LDAA PORT      1075 B6 40 00      INPUT SIGNAL (SAMPLING)
              STAA PORT      1078 B7 40 00
              CMPA #$FF      107B 81 FF          D0='1'?
              BEQ L04        107D 27 0D          GO TO DETECTION OFF-SIGNAL
L15          INCB          107F 5C              ON-SIGNAL LENGTH + 1
              JSR TIMER_10MS 1080 BD 10 C0      TIMER 10 MILLI SECOND
              BRA L05        1083 20 F0          CONTINUE DETECTION ON-SIGNAL

```

%

```

  .ORIGIN $108C

```

%

```

L04          CMPB #$03      108C C1 03          IF LENGTH=<2 RETURN TO DET. ON-SIGNAL
              BCS L15        108E 25 EF
              CMPB TSD      1090 D1 00          COMPARE ON-SIGNAL-LENGTH AND TSD
              BCC L06        1092 24 05          ON-SIGNAL-LENGTH > TSD ?
              ASL CODE_L     1094 78 00 03      LEFT SHIFT INTERMEDIATE CODE
              BRA L07        1097 20 06
L06          ASL CODE_L     1099 78 00 03      LEFT SHIFT INTERMEDIATE CODE
              INC CODE_L     109C 7C 00 03      INTERMEDIATE CODE + 1
L07          CLR B          109F 5F              CLEAR LENGTH
L09          LDAA PORT      10A0 B6 40 00      INPUT SIGNAL (SAMPLING)
              STAA PORT      10A3 B7 40 00
              CMPA #$FE      10A6 81 FE          D0='0'?
              BEQ L10        10A8 27 CA          GO TO DET. ON-SIGNAL
              INCB          10AA 5C              IF OFF-SIGNAL, LENGTH + 1
              CMPB TSD      10AB D1 00
              BCC L08        10AD 24 05          IF OFF-SIGNAL-LENGTH > TSD, EXIT
              JSR TIMER_10MS 10AF BD 10 C0      TIMER 10 MILLI SECOND

```

```

L08          BRA L09          10B2  20 EC          CONTINUE DETECTION OFF-SIGNAL
            RTS              10B4  39
%
%-----

```

```

% TIMER 10 MILLI SECOND
    .ORIGIN $10C0
%

```

```

TIMER_10M    LDX #$03E7      10C0  CE 03 E7      TIMER VARIABLE
L11          DEX              10C3  09
            NOP              10C4  01
            BNE L11          10C5  26 FC
            RTS              10C7  39
%
%-----

```

```

% INITIALIZE ACIA (SERIAL INTERFACE)
    .ORIGIN $F820
%

```

```

INIT_ACIA    LDAA #$03        F820  86 03
            STAA ACIA_STATUS  F822  B7 A0 00
            LDAA #$15        F825  86 15      4800 BIT/S, 8 BIT, NO-PARITY
            STAA ACIA_STATUS  F827  B7 A0 00
            RTS              F82A  39
%
%-----

```

```

% OUTPUT 1 CHARACTER TO SERIAL INTERFACE (RS-232C)
    .ORIGIN $F990
%

```

```

OUT_1CHA     LDAA ACIA_STATUS  F990  B6 A0 00
            LSRA              F993  44
            LSRA              F994  44
            BCC OUT_1CHA      F995  24 F9
            STAB ACIA_DATA    F997  F7 A0 01
            RTS              F99A  39
%
%-----

```

```

% CODE CONVERSION TABLE
    .ORIGIN $1200
%

```

ADDRESS(HEX)	DATA	CHR.	MORSE	INTER.	(HEX)	ASCII
1200	5E	COR.	'.....'	'10000000'	00	5E
1201	3F					
1202	45	'E'	'...'	'10'	02	45
1203	54	'T'	'-..'	'11'	03	54
1204	49	'I'	'..-'	'100'	04	49
1205	41	'A'	'-.-'	'101'	05	41
1206	4E	'N'	'--.'	'110'	06	4E
1207	4D	'M'	'-.-'	'111'	07	4D
1208	53	'S'	'...'	'1000'	08	53
1209	55	'U'	'..-.'	'1001'	09	55
120A	52	'R'	'-.-.'	'1010'	0A	52
120B	57	'W'	'-.-.'	'1011'	0B	57
120C	44	'D'	'-..'	'1100'	0C	44
120D	4B	'K'	'-.-.'	'1101'	0D	4B
120E	47	'G'	'-.-.'	'1110'	0E	47
120F	4F	'O'	'---'	'1111'	0F	4F
1210	48	'H'	'....'	'10000'	10	48
1211	56	'V'	'...-'	'10001'	11	56
1212	46	'F'	'-..-'	'10010'	12	46
1213	3F					
1214	4C	'L'	'-.-.'	'10100'	14	4C
1215	3F					
1216	50	'P'	'-.-.'	'10110'	16	50
1217	4A	'J'	'-.-.'	'10111'	17	4A
1218	42	'B'	'-.-.'	'11000'	18	42
1219	58	'X'	'-.-.'	'11001'	19	58
121A	43	'C'	'-.-.'	'11010'	1A	43
121B	59	'Y'	'-.-.'	'11011'	1B	59
121C	5A	'Z'	'-.-.'	'11100'	1C	5A
121D	51	'Q'	'-.-.'	'11101'	1D	51
121E	3F					
121F	3F					
1220	35	'5'	'....'	'100000'	20	35
1221	34	'4'	'....'	'100001'	21	34

1222	21	ROGER	'..._.'	'100010'	22	21
1223	33	'3'	'...__'	'100011'	23	33
1224	3F					
1225	3F					
1226	3F					
1227	32	'2'	'..____'	'100111'	27	32
1228	26	WAIT	'._...'	'101000'	28	26
1229	3F					
122A	2B	'+'	'._._.'	'101010'	2A	2B
122B	3F					
122C	3F					
122D	3F					
122E	3F					
122F	31	'1'	'._....'	'101111'	2F	31
1230	36	'6'	'_....'	'110000'	30	36
1231	3D	START	'_....'	'110001'	31	3D
1232	2F	'/'	'_..._.'	'110010'	32	2F
1233	3F					
1234	3F					
1235	3F					
1236	28	'('	'_._...'	'110110'	36	28
1237	3F					
1238	37	'7'	'__...'	'111000'	38	37
1239	3F					
123A	3F					
123B	3F					
123C	38	'8'	'____.'	'111100'	3C	38
123D	3F					
123E	39	'9'	'_____.'	'111110'	3E	39
123F	30	'0'	'_____.'	'111111'	3F	30
1240	25	'^'	'.....'	'1000000'	40	25
1241						
1242						
1243						
1244						
1245						
1246						
1247						
1248						
1249						
124A						
124B						
124C	3F	'?'	'..__...'	'1001100'	4C	3F
124D						
124E						
124F						
1250						
1251						
1252	22	'”'	'._..._.'	'1010010'	52	22
1253						
1254						
1255	2E	'.'	'._._._.'	'1010101'	55	2E
1256						
1257						
1258						
1259						
125A	40	'@'	'_..._.'	'1011010'	5A	40
125B						
125C						
125D						
125E	27	'”'	'._....'	'1011110'	5E	27
125F						
1260						
1261	2D	'-'	'_.....'	'1100001'	61	2D
1262						
1263						
1264						
1265						
1266						
1267						
1268						
1269						

126A
126B
126C
126D 29 ‘)’ ‘_._._.’ ‘1101101’ 6D 29
126E
126F
1270
1271
1272
1273 2C ‘,’ ‘_._._.’ ‘1110011’ 73 2C
1274
1275
1276
1277
1278 3A ‘:’ ‘_._._.’ ‘1111000’ 78 3A
1279
127A
127B
127C
127D
127E
127F

%

%-----

% END OF PROGRAM