

300 Baud Cassette Interface for the MD690b

The following routine was written by Robert Alkire, the author of MONBUG II. It loads Motorola Format Kansas City Standard Cassette tapes through the 2400 baud cassette interface of the MD690b. It works so well, in fact, that the cassette interface doesn't even have to be recalibrated to change from 2400 baud to 300 and back again! It is even easier to **generate** 300 baud tapes through the cassette interface. To send a "one" bit at 300 baud, shift out the string "llllllll". This will generate 8 cycles of 2400 hz. To send a "zero" bit shift out the string "l0l0l0l0", which will generate 4 cycles of 1200 hz.

File: LOAD300 ASSM-6809

```

*****LOAD300*****
*WRITTEN BY R. ALKIRE AND S. ALCORN
*REV 1.1
*11/20/79
*
*GENERAL EQUATES
*
FC90 = EXEC EQU $FC90 MONBUG II ENTRY POINT
F400 = PIADA EQU $F400 PIA LOCATIONS
F401 = PIASA EQU PIADA+1
*
*
*SUPERVISORY ROUTINE
*
*
                ORG     $A100

A100 31 8D FF FD      LEAY  LBIT,PCR SET POINTER TO LBIT
A104 32 3F           LEAS  -1,Y   SET STACK POINTER TO STORAGE AREA
A106 6F A4           CLR   0,Y   CLEAR LBIT
A108 8D 68           0072 LD300 BSR  CIN300 GET A CHARACTER
A10A 81 53           CMPA  #'S'   FIND START OF RECORD
A10C 26 FA           0008 BNE  LD300 MARKER
A10E 8D 62           0072 BSR  CIN300 READ DIRECTIVE BYTE
A110 81 39           CMPA  #'9'
A112 27 29           003D BEQ  CEXIT  IF 9, TERMINATE LOAD
A114 81 31           CMPA  #'1'   VALID DATA MARKER?
A116 26 F0           0008 BNE  LD300 NO, KEEP LOOKING
A118 5F             CLRB             CHECKSUM
A119 8D 2D           0048 BSR  BYT300 GET DATA COUNT
A11B 80 02           SUBA  #2       ADJUST
A11D 34 02           PSHS A        SAVE ON STACK (6809!)
A11F 8D 27           0048 BSR  BYT300 GET ADDRESS
A121 34 06           PSHS A,B     SAVE
A123 8D 23           0048 BSR  BYT300 MORE ADDRESS
A125 A7 61           STA  1,S     SAVE AND TRANSFER TO
A127 35 10           PULS X      INDEX REGISTER
A129 8D 1D           0048 LD11 BSR  BYT300 GET DATA
A12B 6A E4           DEC  0,S    DECREMENT COUNTER

```

A12D 27 04	0033	BEQ	LD15	DONE?
A12F A7 80		STA	0,X+	SAVE DATA
A131 20 F6	0029	BRA	LD11	
A133 35 02	LD15	PULS	A	CHECKSUM
A135 5C		INCB		IF OK, CONTINUE
A136 27 D0	0008	BEQ	LD300	
A138 86 3F	CERR	LDA	#'?'	ELSE PRINT ?
A13A 10 3F		SWI2		OUTPUT CHARACTER TO SCREEN USING
A13C 02		BYTE	2	MONBUG II I/O CALL
A13D 7F F4 03	CEXIT	CLR	PIASA+2	RESTORE PIA TO
A140 86 35		LDA	#\$35	NORMAL OPERATION
A142 B7 F4 01		STA	PIASA	
A145 7E FC 90		JMP	EXEC	AND EXIT TO MONITOR
	*			
A148 8D 0F	0059	BYT300	BSR	INH300
A14A 48		ASLA		GET A HEX CHARACTER
A14B 48		ASLA		
A14C 48		ASLA		
A14D 48		ASLA		
A14E 34 02		PSHS	A	SAVE IT IN MSB ON STACK
A150 8D 07	0059	BSR	INH300	GET ANOTHER
A152 AB E0		ADDA	0,S+	COMBINE THEM
A154 34 02		PSHS	A	AND SAVE
A156 EB E0		ADDB	0,S+	ALSO UPDATE CHECKSUM
A158 39		RTS		
	*			
A159 8D 17	0072	INH300	BSR	CIN300
A15B 80 30		SUBA	#\$30	GET A CHARACTER
A15D 2B 0F	006E	BMI	CERR2	CONVERT TO HEX
A15F 81 09		CMPA	#9	NOT HEX?
A161 2F 0A	006D	BLE	INH1	DONE?
A163 81 11		CMPA	#\$11	
A165 2D 07	006E	BLT	CERR2	NOT HEX?
A167 81 16		CMPA	#\$16	
A169 2E 03	006E	BGT	CERR2	NOT HEX?
A16B 80 07		SUBA	#7	FINISH CONVERSION
A16D 39		INH1	RTS	AND RETURN
A16E 32 62		CERR2	LEAS	2,S
A170 20 C6	0038	BRA	CERR	ADJUST STACK POINTER
	*			AND PRINT "?"
A172 34 14		CIN300	PSHS	B,X
A174 C6 0C		LDB	#\$C	SAVE REGISTERS
A176 8D 3C	00B2	BSR	LD8	SET UP PIA
A178 C6 04	GSTRT	LDB	#4	COUNTER
A17A 8D 27	00A3	GSTP	BSR	GBIT
A17C 49		ROLA		GET A BIT
A17D 25 F9	0078	BCS	GSTRT	SHIFT IT
A17F 5A		DECB		
A180 26 F8	007A	BNE	GSTP	
A182 CC 00 08		LDD	#8	
A185 34 06		PSHS	A,B	
A187 8D 12	009B	NBIT	BSR	GBYT
A189 64 E4		LSR	0,S	
A18B AA E4		ORA	0,S	

A18D	A7	E4		STA	0,S	
A18F	6A	61		DEC	1,S	
A191	26	F4	0087	BNE	NBIT	
A193	8D	06	009B	BSR	GBYT	
A195	35	06		PULS	A,B	
A197	84	7F		ANDA	#\$7F	
A199	35	94		PULS	B,X,PC	
*						
A19B	C6	08		GBYT	LDB	#8
A19D	8D	04	00A3	GETBYT	BSR	GBIT GET BIT
A19F	5A			DECB		DECREMENT COUNTER
A1A0	26	FB	009D	BNE	GETBYT	8 BITS?
A1A2	39			RTS		
*						
A1A3	34	04		GBIT	PSHS	B SAVE B
A1A5	4F			CLRA		
A1A6	13			SYNC		WAIT FOR 2400 HZ CLOCK
A1A7	69	02		ROL	2,X	
A1A9	46			RORA		
A1AA	1F	89		TFR	A,B	
A1AC	A8	20		EORA	0,Y	
A1AE	88	80		EORA	#\$80	
A1B0	E7	A4		STB	0,Y	
A1B2	35	84		PULS	B,PC	
*						
A1B4	1A	10		LD8	ORCC	#\$10 MASK INTERRUPTS
A1B6	8E	F4	00		LDX	#\$PIADA INITIALIZE X TO PIA POINTER
A1B9	E7	03			STB	3,X SET UP PIA FOR 2400 HZ CLOCK TO IRQ
A1BB	E6	01			LDB	1,X
A1BD	C4	FC			ANDB	#\$FC
A1BF	E7	01			STB	1,X
A1C1	39				RTS	
*						
A1C2	00			RES	30	STACK AREA
A1E0	00			LBIT	RES	1 TEMPORARY STORAGE
*						
A1E1				END		

No Errors

Symbol Table:

BYT300	0048	C			
CERR	0038	C			
CERR2	006E	C		INH300	0059 C
CEXIT	003D	C		LBIT	0001 R
CIN300	0072	C		LD11	0029 C
EXEC	FC90	E		LD15	0033 C
GBIT	00A3	C		LD300	0008 C
GBYT	009B	C		LD8	00B4 C
GETBYT	009D	C		NBIT	0087 C
GSTP	007A	C		PIADA	F400 E
GSTRT	0078	C		PIASA	F401 E
INH	006D	C			

The following patches are required to use TSC 6809 software with the MicroDaSys MD-690b.

6809 BASIC

<u>Location</u>	<u>Data</u>	
0006	7E	Exit to MONBUG II
0007	FC	
0008	50	
0009	10	Input a character from keyboard
000A	3F	
000B	00	
000F	10	Output a character to video
0010	3F	
0011	01	
0012	7E	Tape Input
0013	FF	
0014	C3	
0015	7E	Tape Output
0016	FF	
0017	62	

6809 Assembler

<u>Location</u>	<u>Data</u>	
0068	FC	Exit to MONBUG II
0069	50	
006A	10	Output character to Video
006B	3F	
006C	01	
006E	FF	Tape Output Routine
006F	62	
0043	FF	Tape "On" Stream
0044	FF	
0045	FF	
0046	FF	

also see page 7 and appendix of TSC manual.

6809 Text Editor

<u>Location</u>	<u>Data</u>	
000E	7E	Exit to MONBUG II
000F	FC	
0010	50	
0006	10	Input a character from keyboard
0007	3F	
0008	00	

000A	10	Output a character to video
000B	3F	
000C	01	
0025	FF	Tape Input
0026	C3	
0023	FF	Tape Output
0024	62	

6809 Debug Package

<u>Location</u>	<u>Data</u>	
5514	7E	Exit to MONBUG II
5515	FC	
5516	50	
5510	10	Input a character from keyboard
5511	3F	
5512	00	
550C	10	Output a character to video
550D	3F	
550E	01	
5518		must branch to a routine which performs the following check for a character ready in the queue:

FE A3 FE	LDU SCR	GET SCRATCH POINTER
6D 5A	TST IQCNT,U	TEST FOR CHARACTERS IN QUEUE