

Tarbell Bootstrap Loader

The Tarbell Bootstrap Loader is a small program which will read any Tarbell format TCOS object file from audio cassette into memory and execute it. The Loader usually resides in two 8702 PROMs at addresses FA00 and FB00. It needs no RAM other than that needed to run the program being loaded. In particular, it needs no stack. The Loader always loads the first object file it finds on the tape, so it need not be specified by the user. While the program is loading, the high byte of the memory address being stored into is displayed in the PROGRAMMED OUTPUT LIGHTS. When the program has been completely read in, the Loader resets the lights and transfers control to the first byte, so the user program is started automatically.

The Loader uses port 03 for status input, with bit 2 going high to indicate data available on port 00.

To run the Loader, merely:

1. Start execution at FA00;
2. Insert the cassette and start the recorder; and
3. Stop the recorder when the program has been loaded. This may be determined by watching the PROGRAMMED OUTPUT LIGHTS.

THE UNIVERSITY OF CHICAGO
LIBRARY
540 EAST 57TH STREET
CHICAGO, ILL. 60637
TEL: 773-936-3000
WWW.CHICAGO.EDU

;18 JAN 77. BRH. CREATED.

```

;*****
;*****
;****
;****          TARBELL BOOTSTRAP LOADER          ****
;****          (C) COPYRIGHT 1977, IMSAI MFG CORP     ****
;****          SAN LEANDRO, CALIFORNIA                 ****
;*****
;*****

```

; THIS PROGRAM HAS SEVERAL SPECIAL CONSTRAINTS:

1. IT MUST BE CAPABLE OF RUNNING IN PROM.
2. IT MUST USE NO RAM OTHER THAN THAT INTO WHICH THE BOOTED PROGRAM IS LOADED; IN PARTICULAR, IT MUST NOT USE A STACK.
3. IT MUST RUN IN 512 BYTES, STARTING AT FA00.
4. IT MUST NOT INCLUDE BYTES 'FA' AND 'FB' AS ANYTHING BUT HI ADDRESS BYTES, SO IT MAY BE RELOCATED. THIS MEANS THAT THE INSTRUCTIONS 'JM' AND 'EI' MUST NOT BE USED, AND THAT 'FA' AND 'FB' MUST NOT OCCUR IN THE 2ND BYTE OF AN INSTRUCTION.

```

0000 =      CRI      EQU      0          ;CASSETTE PORT
0003 =      CRL      EQU      3          ;CONTROL PORT
0004 =      CRY      EQU      4          ;CASSETTE READY BIT
;
FA00          ORG      OFA00H
FA00 AF      START:  XRA      A          ;TURN ON LIGHTS...
FA01 D3FF    OUT      OFFH
FA03 3E60    MVI      A,60H          ;SET MIO TO READ BY BIT...
FA05 D303    OUT      CRL
FA07 DB03    SYNC0:  IN       CRL          ;WAIT TILL 8 BITS ARE READY...
FA09 E604    ANI      CRY
FA0B CA07FA  JZ       SYNC0
FA0E DB00    IN       CRI          ;READ THEM
FA10 FEE6    CPI      OE6H          ;ARE THEY A SYNC?
FA12 C207FA  JNZ      SYNC0          ;WAIT TILL A SYNC IS FOUND
FA15 3E20    MVI      A,20H          ;SET TO READ BY BYTE...
FA17 D303    OUT      CRL
FA19 061F    MVI      B,31          ;# OF OTHER SYNC'S AT START OF FILE
FA1B DB03    SYNC1:  IN       CRL          ;WAIT TILL A BYTE IS READY...
FA1D E604    ANI      CRY
FA1F CA1BFA  JZ       SYNC1
FA22 DB00    IN       CRI          ;READ THE BYTE
FA24 FEE6    CPI      OE6H          ;IS IT A SYNC?
FA26 C200FA  JNZ      START          ;START OVER IF NOT
FA29 05      DCR      B          ;COUNT SYNC'S
FA2A C21BFA  JNZ      SYNC1          ;LOOP UNTIL STRIP DONE
FA2D DB03    TYPE:  IN       CRL          ;READ RECORD TYPE...

```

FA2F E604		ANI	CRY	
FA31 CA2DFA		JZ	TYPE	
FA34 DB00		IN	CRI	
FA36 3D		DCR	A	;IS IT 1 (ABSOLUTE BINARY OBJECT)?
FA37 C200FA		JNZ	START	;START OVER IF NOT
FA3A DB03	COUNT:	IN	CRL	;READ RECORD COUNT...
FA3C E604		ANI	CRY	
FA3E CA3AFA		JZ	COUNT	
FA41 DB00		IN	CRI	
FA43 FE0B		CPI	11	;IS IT 11 (5-BYTE NAME + 3 2-BYTE ADDR'S)
FA45 C200FA		JNZ	START	;START OVER IF NOT
FA48 0605		MVI	B,5	;INIT COUNT OF BYTES IN NAME
FA4A 110000		LXI	D,0	;INIT CHECKSUM
FA4D DB03	NAME:	IN	CRL	;READ A BYTE...
FA4F E604		ANI	CRY	
FA51 CA4DFA		JZ	NAME	
FA54 DB00		IN	CRI	
FA56 83		ADD	E	;ADD TO CHECKSUM...
FA57 5F		MOV	E,A	
FA58 3E00		MVI	A,0	
FA5A 8A		ADC	D	
FA5B 57		MOV	D,A	
FA5C 05		DCR	B	;COUNT LETTERS
FA5D C24DFA		JNZ	NAME	;READ 5 OF THEM
FA60 DB03	ADDSO:	IN	CRL	;READ LO BYTE OF START ADDRESS...
FA62 E604		ANI	CRY	
FA64 CA60FA		JZ	ADDSO	
FA67 DB00		IN	CRI	
FA69 6F		MOV	L,A	;SAVE IN L
FA6A 83		ADD	E	;ADD TO CHECKSUM...
FA6B 5F		MOV	E,A	
FA6C 3E00		MVI	A,0	
FA6E 8A		ADC	D	
FA6F 57		MOV	D,A	
FA70 DB03	ADDS1:	IN	CRL	;READ HI BYTE OF START ADDRESS...
FA72 E604		ANI	CRY	
FA74 CA70FA		JZ	ADDS1	
FA77 DB00		IN	CRI	
FA79 67		MOV	H,A	;SAVE IN H
FA7A 83		ADD	E	;ADD TO CHECKSUM...
FA7B 5F		MOV	E,A	
FA7C 3E00		MVI	A,0	
FA7E 8A		ADC	D	
FA7F 57		MOV	D,A	
FA80 F9		SPLH		;SAVE START ADDRESS IN SP
FA81 DB03	ADDEO:	IN	CRL	;READ LO BYTE OF END ADDRESS...
FA83 E604		ANI	CRY	
FA85 CA81FA		JZ	ADDEO	
FA88 DB00		IN	CRI	
FA8A 6F		MOV	L,A	;SAVE IN L
FA8B 83		ADD	E	;ADD TO CHECKSUM...
FA8C 5F		MOV	E,A	
FA8D 3E00		MVI	A,0	
FA8F 8A		ADC	D	
FA90 57		MOV	D,A	
FA91 DB03	ADDE1:	IN	CRL	;READ HI BYTE OF END ADDRESS...
FA93 E604		ANI	CRY	
FA95 CA91FA		JZ	ADDE1	

FA98 DB00		IN	CRI	
FA9A 67		MOV	H,A	;SAVE IN H
FA9B 83		ADD	E	;ADD TO CHECKSUM...
FA9C 5F		MOV	E,A	
FA9D 3E00		MVI	A,0	
FA9F 8A		ADC	D	
FAA0 57		MOV	D,A	
FAA1 0602		MVI	B,2	;READ LAST 2 BYTES AND ADD TO CHECKSUM...
FAA3 DB03	ADDX:	IN	CRL	
FAA5 E604		ANI	CRY	
FAA7 CAA3FA		JZ	ADDX	
FAAA DB00		IN	CRI	
FAAC 83		ADD	E	
FAAD 5F		MOV	E,A	
FAAE 3E00		MVI	A,0	
FAB0 8A		ADC	D	
FAB1 57		MOV	D,A	
FAB2 05		DCR	B	
FAB3 C2A3FA		JNZ	ADDX	
FAB6 EB		XCHG		
FAB7 DB03	CHECB:	IN	CRL	;PUT CHECKSUM IN HL, END ADDRESS IN DE
FAB9 E604		ANI	CRY	;READ HI CHECKSUM BYTE INTO B...
FABB CAB7FA		JZ	CHECB	
FABE DB00		IN	CRI	
FAC0 47		MOV	B,A	
FAC1 DB03	CHECC:	IN	CRL	;READ LO CHECKSUM BYTE INTO C...
FAC3 E604		ANI	CRY	
FAC5 CAC1FA		JZ	CHECC	
FAC8 DB00		IN	CRI	
FACA 4F		MOV	C,A	
FACB 09		DAD	B	;ADD TO ACCUMULATED CHECKSUM
FACC 7C		MOV	A,H	;IS RESULT 0?...
FACD B5		ORA	L	
FACE C200FA		JNZ	START	;ABORT IF NOT
FAD1 39		DAD	SP	;GET START ADDRESS INTO HL
FAD2 7B		MOV	A,E	;SUBTRACT START FROM END ADDRESS AND
FAD3 95		SUB	L	; PUT RESULT IN HL...
FAD4 6F		MOV	L,A	
FAD5 7A		MOV	A,D	
FAD6 9C		SBB	H	
FAD7 67		MOV	H,A	
FAD8 29		DAD	H	;DOUBLE IT
FAD9 4C		MOV	C,H	;PUT RECORD COUNT IN C...
FADA 0C		INR	C	
FADB 210000		LXI	H,0	;PUT START ADDRESS IN HL...
FADE 39		DAD	SP	
FADF 7C	RECOR:	MOV	A,H	;PUT HIGH ADDRESS BYTE IN LIGHTS...
FAE0 2F		CMA		
FAE1 D3FF		OUT	OFFH	
FAE3 DB03	RTYPE:	IN	CRL	;READ RECORD TYPE...
FAE5 E604		ANI	CRY	
FAE7 CAE3FA		JZ	RTYPE	
FAEA DB00		IN	CRI	
FAEC FE81		CPI	81H	;IS IT AN ABSOLUTE BINARY OBJECT DATA RECF
FAEE C200FA		JNZ	START	;ABORT IF SO
FAF1 DB03	RCOUN:	IN	CRL	;READ BYTE COUNT...
FAF3 E604		ANI	CRY	
FAF5 CAF1FA		JZ	RCOUN	

```

FAF8 DB00          IN      CRI
FAFA 47           MOV     B,A      ;PUT IN B
FAFB 110000       LXI     D,0      ;INIT CHECKSUM
FAFE DB03        RDATA:  IN      CRL   ;READ A DATA BYTE...
FB00 E604         ANI     CRY
FB02 CAFEFA       JZ      RDATA
FB05 DB00         IN      CRI
FB07 77          MOV     M,A      ;STASH IT...
FB08 23          INX     H
FB09 83          ADD     E      ;ADD TO CHECKSUM...
FBOA 5F          MOV     E,A
FB0B 3E00        MVI     A,0
FB0D 8A          ADC     D
FB0E 57          MOV     D,A
FB0F 05          DCR     B      ;COUNT BYTES
FB10 C2FEFA      RDATA:  JNZ     RDATA  ;LOOP TILL ALL DATA IN
FB13 DB03        IN      CRL   ;READ HI CHECKSUM BYTE...
FB15 E604         ANI     CRY
FB17 CA13FB      JZ      RCHEK
FB1A DB00         IN      CRI
FB1C 47          MOV     B,A      ;SAVE IT IN B
FB1D DB03        RCHEL:  IN      CRL   ;READ LO CHECKSUM BYTE...
FB1F E604         ANI     CRY
FB21 CA1DFB      JZ      RCHEL
FB24 DB00         IN      CRI
FB26 83          ADD     E      ;ADD TO LO BYTE OF ACCUMULATED CHECKSUM
FB27 C200FA      JNZ     START  ;ABORT IF ERROR
FB2A 78          MOV     A,B      ;GET HI CHECKSUM BYTE BACK INTO A
FB2B 8A          ADC     D      ;ADD TO HI BYTE OF ACCUMULATED CHECKSUM
FB2C C200FA      JNZ     START  ;ABORT IF ERROR
FB2F 0D          DCR     C      ;COUNT RECORDS
FB30 C2DFFA      JNZ     RECOR  ;LOOP TILL ALL READ
FB33 AF          XRA     A      ;TURN LIGHTS ON (LIKE AFTER A RESET)...
FB34 D3FF        OUT     OFFH
FB36 210000      LXI     H,0      ;GET START ADDRESS INTO HL...
FB39 39          DAD     SP
FB3A E9          PCHL
;
FB3B            END

```