Getting Started

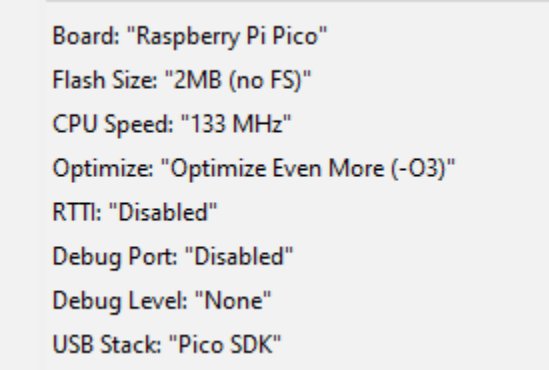
Simple-Circuit 2022

Installing sim680b.ino into your Pico board:

The Pico board requires that you install the Arduino IDE and Earle Philhower's Ardino-Pico port. Follow the instructions for "Installing via Arduino Boards Manager". These are found at:

https://github.com/earlephilhower/arduino-pico

Use the following settings:



PICOBUG Commands:

Monitor Prompt: >
Hex data byte: hh
Enter new hex data byte: **hh**
Hex address: hhhh
Enter new hex address: **hhhh**
<CR> Carriage Return

>M **hhhh** hh <CR> Examine Memory with no change
>M **hhhh** hh **hh** Examine Memory and change
hhhh+1 hh Continues to next address, enter new hex data value or <CR> to end

>N hhhh hh <CR> Examine Next Memory with no change
>N hhhh hh **hh** <CR> Examine Next Memory and change

>J **hhhh** Jump to address hhhh and start running code
>J **1000** Start FIG Forth

>D **hhhh hhhh** Hex dump of data address hhhh to hhhh

>L Load S-record

>P **hhhh hhhh** Punch S-record address hhhh to hhhh

>X **hhhh hhhh** Generate comma delimited 0xhh data dump
Useful for converting memory data to C array data

>B **hhhh** Insert SWI break (0x3F) at address hhhh
B saves the instruction at hhhh
Prints all processor registers when the SWI is executed

>R Restore instruction from B command and do a return from interrupt
Note: Do not use multiple B commands prior to R

Hooks: If you have software that works with the MITS ROM Monitor, address jump hooks have been placed for calling to the original ROM functions. The original ROM monitor source and MITS Basic S-record (paper tape contents) is available at:

https://deramp.com/swtpc.com/Altair/Altair_Basic.htm

The PICOBUG L command will load the Basic S-record into memory. Use the monitor command J 0000 to start Basic.

Picobug Code JSR Address Hook Entry Points:

It is recommended to use the hook addresses. The base addresses can change as the software is updated.

getch 0xFF00 – returns 7-bit ascii byte in acc B, calls prtch if 0x00F3 msb is clear (echo input)
getbyte 0xFF53 – returns one byte in acc B, exits to prompt if input is not 0-9 or A-F
getwrd 0xFF62 – returns one word in reg X and at address 0x00FA, exits to prompt if input is not hex
prtch 0xFF81 – prints one 7-bit ascii byte from acc B
prtbyt 0xFF2D – prints one byte from acc A, calls getch
reset 0xFFD8 – processor reset entry point, sets the stack pointer then goes to prompt

Addresses 0x00E0 through 0x00FF are used for the stack and system variables.

SP is initialized to 0X00F1 and it moves down with pushes

FIG Forth:

The Forth Interest Group source was transferred by hand from a PDF file. Some content such as comments may be omitted in parts. Also, the serial data code was modified to allow function with the Pico board. The Free License is presented in both the Arduino source and the assembly source file. Due to possible type errors, malfunction is possible.

Forth is preloaded on power up but it is not updated upon a reset. The PICOBUG ROM area is updated upon a reset. PICOBUG will always work after a reset but the RAM area for Forth could be corrupted prior to the reset.

If you send a program file from simpleCRT.exe to Forth, check the LF filter to remove line feeds. Else, the text from the file will generate errors when defining the new dictionary values.

Hardware I/O:

Serial data is transmitted through the USB interface at 9600 baud. You will need a serial terminal program to communicate with the Pico. A Windows Serial Terminal program written in Lazarus, simpleCRT.exe, is included in the file folder. It has a buffer of 256 lines and can plot space delimited decimal data. Set the Baud to 9600 from the drop down menu. Next, select your Pico Com number. Click on Open to start communication.

An 8-bit input port is present at address 0xf011. The output port is at address 0xf010. Check the sim680b.ino listing for details of which instructions access ports. I/O was limited to a few op codes.

The ADC is read by writing the input pin number 1, 2 or 3 to address 0xf020. The high byte is read at 0xf020 and the low byte at 0xf021. Data is 12-bits.

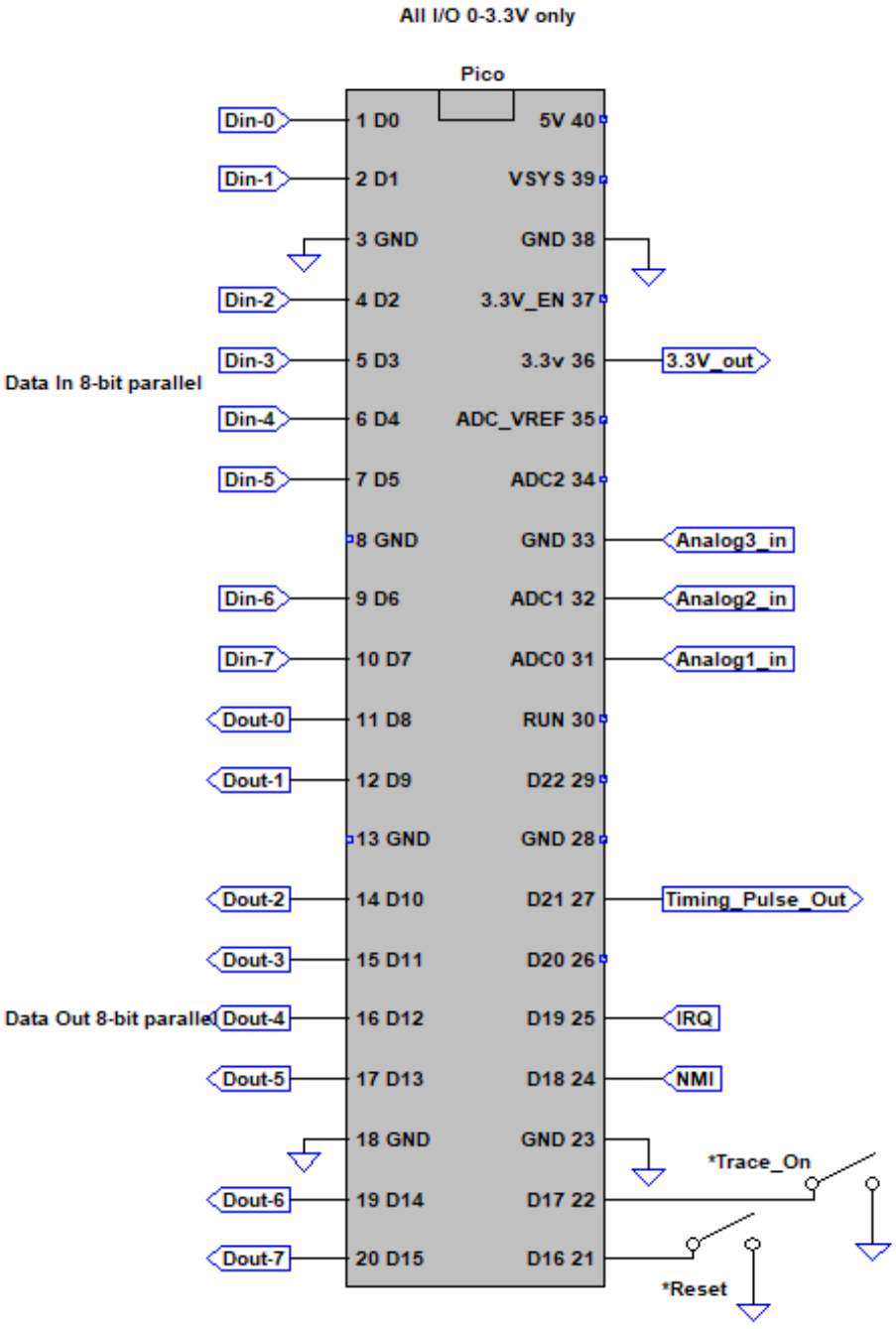
Example Forth I/O definitions:

```
HEX
: ADC 1 F020 C! F020 C@ 100 * F021 C@ + ;
: DIGOUT F010 C! ;
: DIGIN F011 C@ ;
: PLOTADC BEGIN ADC . CR ?TERMINAL UNTIL ;
DECIMAL
```

Example Basic ADC I/O:

```
10 POKE 61472,1
20 A = PEEK(61472)
30 B = PEEK(61473)
40 C = (A*256+B)*0.0008057
50 PRINT C
60 GOTO 10
```

Hardware Connections:

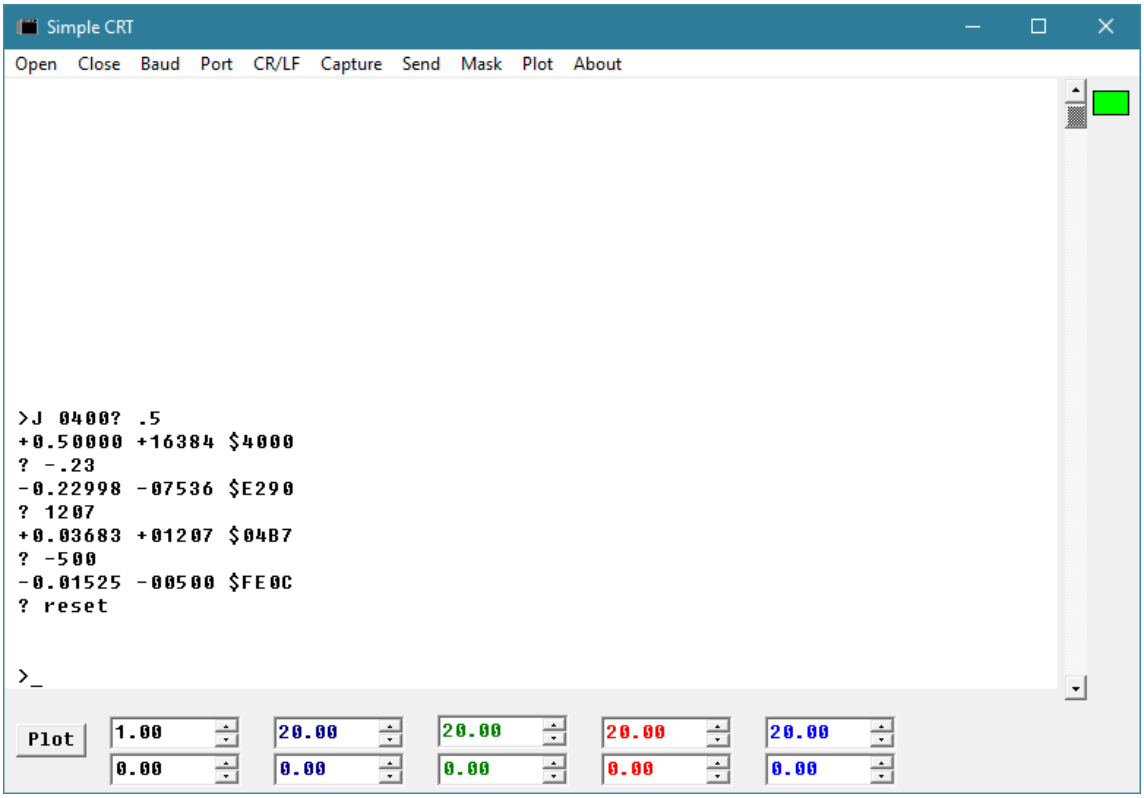


Example Assembly Language:

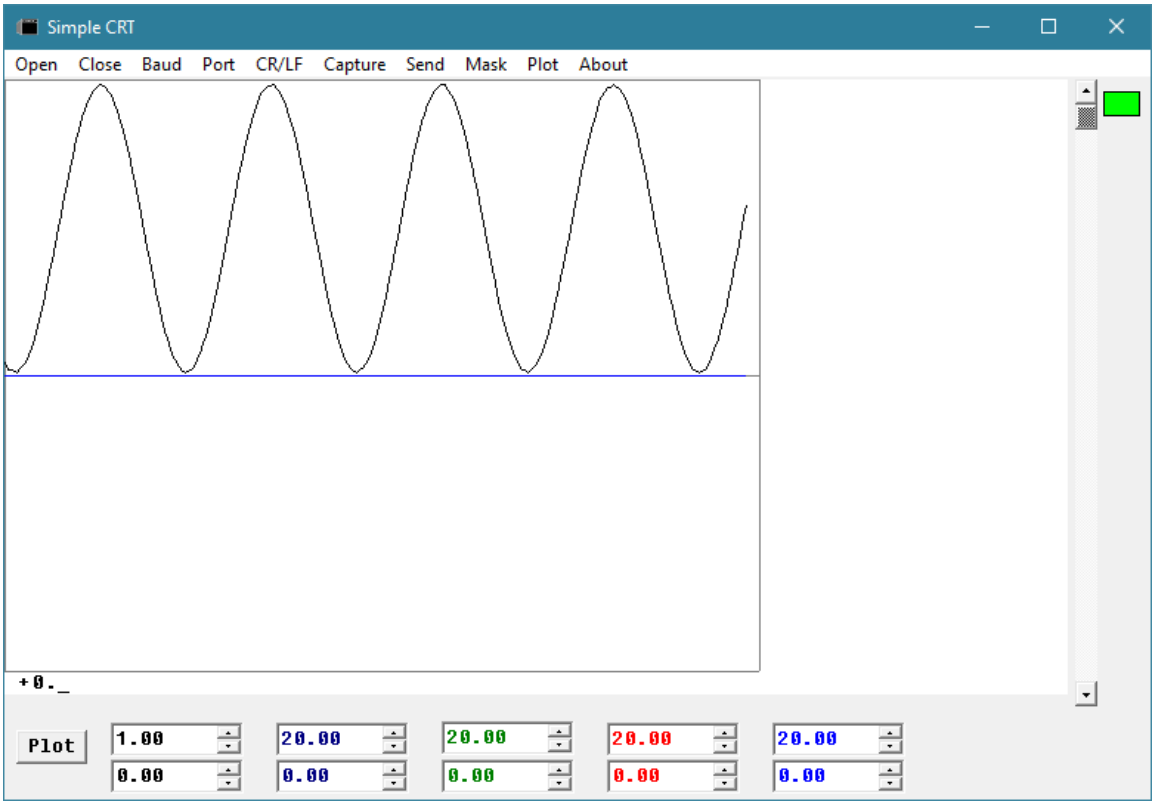
Copy the following S-record data and save it to a file. Then, use the L (load command) and file send to load the program. J 0400 will run a program the reads decimal integers or decimal fractions and converts them to binary values. The binary values are then printed as a decimal fraction, decimal integer and the binary hexadecimal value. Do a hardware reset to exit.

J 0500 will read the ADC channel 1 and print the value as a fraction. The value can be plotted in simpleCRT. Press any key to exit.

S1210400BDD00E9634D635BDD1B19634D635BDD1109634C624BDFF819634BDFF6D963E
S114041E35BDFF6DC60DBDFF81C60ABDFF817E0400CC
S1210500C601F7F020F6F02017F6F021584958495849BDD1B1C60DBDFF81C60ABDFF29
S10D051E81F6F0005424DB7EFFD8C0
S121D000E6002706BDFF810820F6393F2000CED00BBDD00086307F0038CE003AC60097
S121D01ED73AD73BD73CD73DD73ED73FBDFF00C12B2707C12D2606730038BDFF00C163
S121D03C2E2739C1302511C139220DE700088C003F2705BDFF0020EBCE003ABDD0E0D2
S121D05AD63596347D00382A064353CB0189009734D735C60DBDFF81C60ABDFF8139E2
S121D078BDFF00C130250EC139220AE700088C003E270220EBCE003ABDD09520C54F45
S121D0969734973597309731860 4E60058585858587900317900305879003179003 02C
S121D0B458790031790030587900317900300 84A26DAC60F96319B311997319630990A
S121D0D230199730790035790034 5A26E9394F97349735E6002726C40F088C003F2E42
S121D0F01ED730780035790034D635963458495849DB359934DB3089009734D7352020
S121D10ED6394D2A1A81802608C1002604CB0189004353CB018900D7319730C62D2023
S121D12C06D7319730C62BBDFF819630D6317F0039C01082272B09D73197307C003928
S121D14A20F1D639CB30BDFF819630D6317F0039C0E882032B09D73197307C003920E1
S121D168F1D639CB30BDFF819630D6317F0039C06482002B099730D7317C003920F179
S121D186D639CB30BDFF81D6317F0039C00A2B07D7317C003920F5D639CB30BDFF816C
S121D1A4D631CB30BDFF81C620BDFF8139D73197302B04C62B201CD6319630818026AF
S121D1C208C1002604CB0189004353CB0189009730D731C62DBDFF81C630BDFF81C620
S121D1E02EBDFF81780031790030CE00057F00327F00337800317900307900329632 75
S121D1FE9733D631963058497900335849790033DB319930D731973024037C003396CE
S115D21C329B338B3016BDFF810926C5C620BDFF81399E
S9



J 0400 convert program



J 0500 readadc program

M68HC11 Absolute Assembler Version 2.70C:CONVERT.ASC

```
 1 A              **EXAMPLE ASSEMBLY PROGRAMS SIMPLE-CIRCUIT 2022
 2 A
 3 A      0030             ORG      $0030             FREE MEMORY NOT USED BY ANY PROGRAMS
 4 P 0030 0002      TEMP     RMB      2
 5 P 0032 0002      TEMP2    RMB      2
 6 P 0034 0004      RSLT     RMB      4
 7 P 0038 0001      SGN      RMB      1
 8 P 0039 0001      COUNT    RMB      1
 9 P 003A 0006      STRG     RMB      6
10 A      FF81      PRTCH    EQU      $FF81
11 A      FF00      GETCH    EQU      $FF00
12 A      FF6D      PRT2H    EQU      $FF6D
13 A      FFD8      RESET    EQU      $FFD8
14 A      F020      ADC      EQU      $F020
15 A      F000      STATUS   EQU      $F000
16 A 0040
17 A              **READ AN INTEGER +/-DDDDD OR FRACTION +/-.DDDDD WHERE D=DECIMAL DIGIT
18 A              **THEN PRINT THE FRACTION AND INTEGER VALUES
19 A
20 A      0400             ORG      $0400
21 A 0400 BDD00E    START    JSR      READ              CONVERT DECIMAL NUMBER
22 A 0403 9634             LDAA     RSLT              LOAD BINARY RESULT
23 A 0405 D635             LDAB     RSLT+1
24 A 0407 BDD1B1           JSR      PRINT             PRINT DECIMAL FRACTION
25 A 040A 9634             LDAA     RSLT              LOAD BINARY RESULT
26 A 040C D635             LDAB     RSLT+1
27 A 040E BDD110           JSR      PRINTI            PRINT DECIMAL INTEGER
28 A 0411 9634             LDAA     RSLT              LOAD BINARY RESULT
29 A 0413 C624             LDAB     #'$
30 A 0415 BDFF81           JSR      PRTCH             PRINT $
31 A 0418 9634             LDAA     RSLT
32 A 041A BDFF6D           JSR      PRT2H             PRINT HEX VALUE
33 A 041D 9635             LDAA     RSLT+1
34 A 041F BDFF6D           JSR      PRT2H             PRINT HEX VALUE
35 A 0422 C60D             LDAB     #$0D
36 A 0424 BDFF81           JSR      PRTCH             PRINT <CR>
37 A 0427 C60A             LDAB     #$0A
38 A 0429 BDFF81           JSR      PRTCH             PRINT <LF>
39 A 042C 7E0400           JMP      START             LOOP BACK TO START, GROUND RESET PIN TO END
40 A 042F
```

```
41 A 042F
42 A              **READ THE ADC AND PRINT THE VALUE AS A FRACTION
43 A     0500             ORG      $0500
44 A 0500 C601      READADC   LDAB     #1
45 A 0502 F7F020            STAB     ADC              READ ADC CHANNEL 1
46 A 0505 F6F020            LDAB     ADC
47 A 0508 17               TBA
48 A 0509 F6F021            LDAB     ADC+1
49 A 050C 58               LSLB                      NORMALIZE FOR .0000 TO .9999
50 A 050D 49               ROLA
51 A 050E 58               LSLB
52 A 050F 49               ROLA
53 A 0510 58               LSLB
54 A 0511 49               ROLA
55 A 0512 BDD1B1            JSR      PRINT            PRINT FRACTION
56 A 0515 C60D             LDAB     #$0D
57 A 0517 BDFF81            JSR      PRTCH            <CR>
58 A 051A C60A             LDAB     #$0A
59 A 051C BDFF81            JSR      PRTCH            <LF>
60 A 051F F6F000            LDAB     $F000
61 A 0522 54               LSRB
62 A 0523 24DB             BCC      READADC            LOOP UNTIL KEY PRESSED
63 A 0525 7EFFD8            JMP      RESET            RETURN TO PROMPT
64 A 0528
65 A     D000             ORG      $D000
66 A              **PRINT A STRING POINTED TO BY X, END ON NULL
67 A D000 E600      PRNST     LDAB     0,X
68 A D002 2706             BEQ      PRNST2
69 A D004 BDFF81            JSR      PRTCH
70 A D007 08               INX
71 A D008 20F6             BRA      PRNST
72 A D00A 39       PRNST2    RTS
73 A D00B
74 A D00B 3F20      PROMPT    FCC      /? /
75 A D00D 00               FCB      0
76 A D00E
77 A              ** PARSE AND READ INTEGER OR FRACTION
78 A              ** [+-].NNNN OR [+-]NNNNN
79 A D00E CED00B    READ      LDX      #PROMPT
80 A D011 BDD000            JSR      PRNST
81 A D014 8630             LDAA     #'0
82 A D016 7F0038    READ1     CLR      SGN
83 A D019 CE003A            LDX      #STRG
84 A D01C C600             LDAB     #0               MARK END OF ASCII STRING
```

```
 85 A D01E D73A            STAB    STRG            ASCII DIGIT
 86 A D020 D73B            STAB    STRG+1          ASCII DIGIT
 87 A D022 D73C            STAB    STRG+2          ASCII DIGIT
 88 A D024 D73D            STAB    STRG+3          ASCII DIGIT
 89 A D026 D73E            STAB    STRG+4          ASCII DIGIT
 90 A D028 D73F            STAB    STRG+5          ASCII DIGIT
 91 A D02A BDFF00          JSR     GETCH
 92 A D02D C12B            CMPB    #'+             CHECK FOR SIGN CHARACTER
 93 A D02F 2707            BEQ     READP
 94 A D031 C12D            CMPB    #'-
 95 A D033 2606            BNE     READP1
 96 A D035 730038          COM     SGN             FLAG NEGATIVE SIGN
 97 A D038 BDFF00  READP    JSR     GETCH
 98 A D03B C12E    READP1   CMPB    #'.
 99 A D03D 2739            BEQ     READF           DO FRACTION IF DECIMAL POINT
100 A D03F C130    READI    CMPB    #'0
101 A D041 2511            BLO     READX           EXIT IF NOT 0-9 DIGIT OR 5 DIGITS MAX
102 A D043 C139            CMPB    #'9
103 A D045 220D            BHI     READX
104 A D047 E700            STAB    0,X
105 A D049 08              INX
106 A D04A 8C003F          CPX     #STRG+5
107 A D04D 2705            BEQ     READX
108 A D04F BDFF00          JSR     GETCH           GET NEXT DIGIT
109 A D052 20EB            BRA     READI
110 A D054 CE003A  READX   LDX     #STRG           POINT TO STRING START
111 A D057 BDD0E0          JSR     ITOB            CONVERT INTEGER TO BINARY
112 A D05A D635    READXF  LDAB    RSLT+1
113 A D05C 9634            LDAA    RSLT
114 A D05E 7D0038          TST     SGN
115 A D061 2A06            BPL     READXF1         IF NEGATIVE SIGN THEN NEGATE RESULT
116 A D063 43              COMA
117 A D064 53              COMB
118 A D065 CB01            ADDB    #1
119 A D067 8900            ADCA    #0
120 A D069 9734    READXF1 STAA    RSLT            SAVE ACC A AND ACC B TO RESULT
121 A D06B D735            STAB    RSLT+1
122 A D06D C60D            LDAB    #$0D            PRINT <CR><LF>
123 A D06F BDFF81          JSR     PRTCH
124 A D072 C60A            LDAB    #$0A
125 A D074 BDFF81          JSR     PRTCH
126 A D077 39              RTS                     RETURN FROM SUB
127 A D078 BDFF00  READF   JSR     GETCH
128 A D07B C130            CMPB    #'0             EXIT IF NOT 0-9 DIGIT
```

```
129 A D07D 250E            BLO     READF1
130 A D07F C139            CMPB    #'9
131 A D081 220A            BHI     READF1
132 A D083 E700            STAB    0,X
133 A D085 08              INX
134 A D086 8C003E          CPX     #STRG+4          READ 4 DIGITS MAX
135 A D089 2702            BEQ     READF1
136 A D08B 20EB            BRA     READF
137 A D08D CE003A  READF1  LDX     #STRG            POINT TO START OF STRING
138 A D090 BDD095          JSR     FTOB             CONVERT FRACTION TO BINARY
139 A D093 20C5            BRA     READXF
140 A D095
141 A D095
142 A             **CONVERT UNSIGNED FRACTION TO BINARY
143 A             **REGISTER X CONTAINS START ADDRESS OF DECIMAL STRING
144 A             **RSLT, RSLT+1 CONTAINS BINARY RESULT
145 A D095 4F      FTOB    CLRA
146 A D096 9734            STAA    RSLT
147 A D098 9735            STAA    RSLT+1
148 A D09A 9730            STAA    TEMP
149 A D09C 9731            STAA    TEMP+1
150 A D09E 8604            LDAA    #4
151 A D0A0 E600    FTOB1   LDAB    0,X              GET DIGIT
152 A D0A2 58              ASLB                     SHIFT LOWER NIBBLE TO HIGH NIBBLE
153 A D0A3 58              ASLB
154 A D0A4 58              ASLB
155 A D0A5 58              ASLB
156 A D0A6 58              ASLB                     SHIFT NIBBLE INTO 16-BIT TEMP AREA
157 A D0A7 790031          ROL     TEMP+1
158 A D0AA 790030          ROL     TEMP
159 A D0AD 58              ASLB
160 A D0AE 790031          ROL     TEMP+1
161 A D0B1 790030          ROL     TEMP
162 A D0B4 58              ASLB
163 A D0B5 790031          ROL     TEMP+1
164 A D0B8 790030          ROL     TEMP
165 A D0BB 58              ASLB
166 A D0BC 790031          ROL     TEMP+1
167 A D0BF 790030          ROL     TEMP
168 A D0C2 08              INX
169 A D0C3 4A              DECA                     GET ALL FOUR DIGITS INTO TEMP
170 A D0C4 26DA            BNE     FTOB1
171 A D0C6 C60F    FTOB2   LDAB    #15
172 A D0C8 9631    FTOB3   LDAA    TEMP+1           DECIMAL MULTIPLY TEMP BY 2
```

```
173 A D0CA 9B31          ADDA    TEMP+1
174 A D0CC 19            DAA
175 A D0CD 9731          STAA    TEMP+1
176 A D0CF 9630          LDAA    TEMP
177 A D0D1 9930          ADCA    TEMP
178 A D0D3 19            DAA
179 A D0D4 9730          STAA    TEMP
180 A D0D6 790035        ROL     RSLT+1          SHIFT CARRY INTO RESULT
181 A D0D9 790034        ROL     RSLT
182 A D0DC 5A            DECB                    REPEAT 15X
183 A D0DD 26E9          BNE     FTOB3
184 A D0DF 39            RTS
185 A D0E0
186 A           *CONVERT UNSIGNED INTEGER INPUT TO BINARY
187 A           **REGISTER X CONTAINS START ADDRESS OF DECIMAL STRING
188 A           **RSLT, RSLT+1 CONTAINS BINARY RESULT
189 A D0E0 4F       ITOB    CLRA
190 A D0E1 9734          STAA    RSLT
191 A D0E3 9735          STAA    RSLT+1
192 A D0E5 E600     INTBNX  LDAB    0,X             GET DECIMAL DIGIT
193 A D0E7 2726          BEQ     INTBRET
194 A D0E9 C40F          ANDB    #$0F            MASK FOR LOWER NIBBLE
195 A D0EB 08            INX
196 A D0EC 8C003F        CPX     #STRG+5
197 A D0EF 2E1E          BGT     INTBRET
198 A D0F1 D730          STAB    TEMP            MULTIPLY RESULT BY 10
199 A D0F3 780035        ASL     RSLT+1
200 A D0F6 790034        ROL     RSLT
201 A D0F9 D635          LDAB    RSLT+1
202 A D0FB 9634          LDAA    RSLT
203 A D0FD 58            ASLB
204 A D0FE 49            ROLA
205 A D0FF 58            ASLB
206 A D100 49            ROLA
207 A D101 DB35          ADDB    RSLT+1
208 A D103 9934          ADCA    RSLT
209 A D105 DB30          ADDB    TEMP            ADD NEXT DIGIT TO RESULT
210 A D107 8900          ADCA    #0
211 A D109 9734          STAA    RSLT
212 A D10B D735          STAB    RSLT+1
213 A D10D 20D6          BRA     INTBNX
214 A D10F 39       INTBRET RTS
215 A D110
216 A           *PRINT 2'S COMPLEMENT 16-BIT INTEGER IN ACC A ACC B
```

```
217 A D110
218 A D110 4D        PRINTI    TSTA
219 A D111 2A1A              BPL       PRINTI1             CHECK SIGN
220 A D113 8180              CMPA      #$80                FIX $8000 OVERFLOW
221 A D115 2608              BNE       PRINTI2
222 A D117 C100              CMPB      #$00
223 A D119 2604              BNE       PRINTI2
224 A D11B CB01              ADDB      #1
225 A D11D 8900              ADCA      #0
226 A D11F 43        PRINTI2   COMA                        NEGATE IF NEGATIVE
227 A D120 53                COMB
228 A D121 CB01              ADDB      #1
229 A D123 8900              ADCA      #0
230 A D125 D731              STAB      TEMP+1
231 A D127 9730              STAA      TEMP
232 A D129 C62D              LDAB      #'-                 PRINT MINUS
233 A D12B 2006              BRA       PRINTI3
234 A D12D D731      PRINTI1   STAB     TEMP+1
235 A D12F 9730              STAA      TEMP
236 A D131 C62B              LDAB      #'+                 PRINT PLUS
237 A D133 BDFF81    PRINTI3   JSR      PRTCH
238 A D136 9630              LDAA      TEMP
239 A D138 D631              LDAB      TEMP+1
240 A D13A 7F0039              CLR       COUNT
241 A D13D C010      PRINTI4   SUBB     #$10                DIV BY 10000 (USING SUBTRACT LOOP)
242 A D13F 8227              SBCA      #$27
243 A D141 2B09              BMI       PRINTI5
244 A D143 D731              STAB      TEMP+1
245 A D145 9730              STAA      TEMP
246 A D147 7C0039              INC       COUNT
247 A D14A 20F1              BRA       PRINTI4
248 A D14C D639      PRINTI5   LDAB      COUNT               PRINT 10K DIGIT
249 A D14E CB30              ADDB      #$30
250 A D150 BDFF81              JSR       PRTCH
251 A D153 9630              LDAA      TEMP
252 A D155 D631              LDAB      TEMP+1
253 A D157 7F0039              CLR       COUNT
254 A D15A C0E8      PRINTI6   SUBB     #$E8                DIV BY 1000
255 A D15C 8203              SBCA      #$03
256 A D15E 2B09              BMI       PRINTI7
257 A D160 D731              STAB      TEMP+1
258 A D162 9730              STAA      TEMP
259 A D164 7C0039              INC       COUNT
260 A D167 20F1              BRA       PRINTI6
```

```
261 A D169 D639      PRINTI7  LDAB     COUNT            PRINT 1K DIGIT
262 A D16B CB30              ADDB     #$30
263 A D16D BDFF81            JSR      PRTCH
264 A D170 9630              LDAA     TEMP
265 A D172 D631              LDAB     TEMP+1
266 A D174 7F0039            CLR      COUNT
267 A D177 C064      PRINTI8  SUBB     #$64             DIV BY 100
268 A D179 8200              SBCA     #0
269 A D17B 2B09              BMI      PRINTI9
270 A D17D 9730              STAA     TEMP
271 A D17F D731              STAB     TEMP+1
272 A D181 7C0039            INC      COUNT
273 A D184 20F1              BRA      PRINTI8
274 A D186 D639      PRINTI9  LDAB     COUNT            PRINT 100'S DIGIT
275 A D188 CB30              ADDB     #$30
276 A D18A BDFF81            JSR      PRTCH
277 A D18D D631              LDAB     TEMP+1
278 A D18F 7F0039            CLR      COUNT
279 A D192 C00A      PRINTIA  SUBB     #$0A             DIV BY 10
280 A D194 2B07              BMI      PRINTIB
281 A D196 D731              STAB     TEMP+1
282 A D198 7C0039            INC      COUNT
283 A D19B 20F5              BRA      PRINTIA
284 A D19D D639      PRINTIB  LDAB     COUNT            PRINT 10'S DIGIT
285 A D19F CB30              ADDB     #$30
286 A D1A1 BDFF81            JSR      PRTCH
287 A D1A4 D631              LDAB     TEMP+1           PRINT ONE'S DIGIT
288 A D1A6 CB30              ADDB     #$30
289 A D1A8 BDFF81            JSR      PRTCH
290 A D1AB C620      PRINTIC  LDAB     #$20             PRINT SPACE
291 A D1AD BDFF81            JSR      PRTCH
292 A D1B0 39              RTS
293 A D1B1
294 A              *PRINT 2'S COMPLEMENT 16-BIT FRACTION IN ACC A ACC B
295 A D1B1 D731      PRINT    STAB     TEMP+1
296 A D1B3 9730              STAA     TEMP
297 A D1B5 2B04              BMI      PRINT2           TEST SIGN
298 A D1B7 C62B              LDAB     #'+
299 A D1B9 201C              BRA      PRINT3           PRINT PLUS
300 A D1BB D631      PRINT2   LDAB     TEMP+1
301 A D1BD 9630              LDAA     TEMP
302 A D1BF 8180              CMPA     #$80             AJUST $8000 OVERFLOW
303 A D1C1 2608              BNE      PRINT6
304 A D1C3 C100              CMPB     #$00
```

```
305 A D1C5 2604            BNE      PRINT6
306 A D1C7 CB01            ADDB     #1
307 A D1C9 8900            ADCA     #0
308 A D1CB 43       PRINT6 COMA                NEGATE VALUE
309 A D1CC 53              COMB
310 A D1CD CB01            ADDB     #1
311 A D1CF 8900            ADCA     #0
312 A D1D1 9730            STAA     TEMP
313 A D1D3 D731            STAB     TEMP+1
314 A D1D5 C62D            LDAB     #'-
315 A D1D7 BDFF81   PRINT3 JSR      PRTCH          PRINT MINUS
316 A D1DA C630            LDAB     #'0
317 A D1DC BDFF81          JSR      PRTCH
318 A D1DF C62E            LDAB     #'.
319 A D1E1 BDFF81          JSR      PRTCH
320 A D1E4 780031          ASL      TEMP+1         REMOVE SIGN BIT
321 A D1E7 790030          ROL      TEMP
322 A D1EA CE0005          LDX      #5
323 A D1ED 7F0032   PRINT4 CLR      TEMP2          OVERFLOW
324 A D1F0 7F0033          CLR      TEMP2+1        RESULT
325 A D1F3 780031          ASL      TEMP+1         MULTIPLY BY 2
326 A D1F6 790030          ROL      TEMP
327 A D1F9 790032          ROL      TEMP2
328 A D1FC 9632            LDAA     TEMP2
329 A D1FE 9733            STAA     TEMP2+1
330 A D200 D631            LDAB     TEMP+1
331 A D202 9630            LDAA     TEMP
332 A D204 58              ASLB               MULTIPLY BY 8
333 A D205 49              ROLA
334 A D206 790033          ROL      TEMP2+1
335 A D209 58              ASLB
336 A D20A 49              ROLA
337 A D20B 790033          ROL      TEMP2+1
338 A D20E DB31            ADDB     TEMP+1         SUM FOR MULTIPLY BY 10
339 A D210 9930            ADCA     TEMP
340 A D212 D731            STAB     TEMP+1
341 A D214 9730            STAA     TEMP
342 A D216 2403            BCC      PRINT5
343 A D218 7C0033          INC      TEMP2+1
344 A D21B 9632     PRINT5 LDAA     TEMP2
345 A D21D 9B33            ADDA     TEMP2+1
346 A D21F 8B30            ADDA     #$30
347 A D221 16              TAB
348 A D222 BDFF81          JSR      PRTCH          PRINT DECIMAL DIGIT
```

```
 349 A D225 09              DEX
 350 A D226 26C5             BNE      PRINT4          LOOP UNTIL 5 DIGITS PRINTED
 351 A D228 C620             LDAB     #$20
 352 A D22A BDFF81            JSR      PRTCH
 353 A D22D 39              RTS
 354 A D22E
 355 A                   END

SYMBOL TABLE:  Total Entries=   53
ADC             F020    PRINTIB          D19D
COUNT            0039    PRINTIC          D1AB
FTOB            D095    PRNST           D000
FTOB1            D0A0    PRNST2           D00A
FTOB2            D0C6    PROMPT           D00B
FTOB3            D0C8    PRT2H            FF6D
GETCH            FF00    PRTCH            FF81
INTBNX           D0E5    READ             D00E
INTBRET          D10F    READ1            D016
ITOB            D0E0    READADC          0500
PRINT           D1B1    READF           D078
PRINT2           D1BB    READF1           D08D
PRINT3           D1D7    READI           D03F
PRINT4           D1ED    READP            D038
PRINT5           D21B    READP1           D03B
PRINT6           D1CB    READX            D054
PRINTI          D110    READXF           D05A
PRINTI1          D12D    READXF1          D069
PRINTI2          D11F    RESET           FFD8
PRINTI3          D133    RSLT            0034
PRINTI4          D13D    SGN             0038
PRINTI5          D14C    START            0400
PRINTI6          D15A    STATUS           F000
PRINTI7          D169    STRG            003A
PRINTI8          D177    TEMP             0030
PRINTI9          D186    TEMP2            0032
PRINTIA          D192
Total errors: 0
```