



The first in a series of tutorials for the beginner to intermediate machine language programmer

Machine Language Made BASIC

Part I: General Math

By William P. Nee

When I first started to learn machine language, setting up graphics seemed like a very complicated procedure. All those VDGs and SAMs were really discouraging. But then I realized that when you type in commands such as PMODE 4 or PCLS 2 or SCREEN 1,1, the computer must do something with those commands and numbers.

Why not find those same routines in ROM so you can just enter, for example, the PMODE numbers you want and then let the computer do all the work of setting the VDGs and SAMs? This series of articles is the result of several months of studying ROM routines to see where you can enter them, and it should help to make machine language programming a little more BASIC. These programs have been assembled with the *EDTASM+* cartridge.

In the next 13 articles, we'll use machine language for basic math functions: RND, PRINT, PRINT USING, PMODE, PSET, LINE, PLAY, etc. We'll make a "dump" program for a seven-dot printer, create music with six voices and wind up with a 3-D rotation program that includes perspective. Most of the machine language programs will have either an explanation or a BASIC program listing for comparison.

Many of these programs are written for ease of understanding rather than for maximum speed. You will find places where you can cut down on the number of commands and refine the program. Just be sure you understand how the whole program works before modifying it. Thanks to Adrian Kotik for debugging the programs and proofreading and assembling the articles.

Machine language is not a complicated language, just different. Commands are written in an abbreviated format such as LD for Load, ST for Store, SUB for Subtract, etc. Most numbers are written in Hex format, using the numerical Base 16; however, *EDTASM+* lets us use the more familiar Base 10, which we will be using in most of the programs. The \$ prefix will indicate an address or location in Hex; # \$ indicates a Hex number; # indicates a Base 10 number.

Bill Nee reversed the "snowbird" trend by retiring to Wisconsin from a banking career in Florida. He spends the long, cold winters writing programs for his CoCo.

All examples and methods are based on using *EDTASM+*. Be sure the computer is turned off before inserting the cartridge. Examples will usually originate (ORG) at Hex Location \$3000 for compatibility with either a 16K or 32K computer. Each example will end with SWI, although to avoid the End Error message, you may add an extra line by pressing TAB, typing END, pressing TAB again and typing START. If you do, the line after the ORG line should begin with something like START. Any listing you want to save on tape should have END START as the last line of the program.

The following examples will show how to add, subtract, multiply, and divide using machine language.

Example 1: Put the number 5 into Register A. Store the contents of Register A in Location \$D0. Increase the contents of Location \$D0 by 1. Load Register A with the contents of Location \$D0. To put the program into memory, type A/IM/A0 and let the program list to check for any errors; press Z to go to ZBUG, then type G3000. To check the result, press R to examine the registers. Register A now contains 6. The command INC can be used to add the number 1 to registers A, B or a memory location.

Example 1A: In this example, we load Register A with the number 5 and add to Register A (ADDA) the number 4. After running the program, examine the registers to see the result of 9 in Register A.

Example 2: This time the DEC command is used. The results show that DEC can be used to subtract the number 1 from registers A, B or a memory location. The commands INC and DEC are useful for the counting portion of your programs; however, most math problems are a little more complicated than 1 + 1.

Example 2A: The SUB command will subtract a number from Register A or Register B. In many cases, however, both registers A and B may contain numbers we need to use later.

The next six examples illustrate how to add, subtract, and multiply these two registers together.

Example 3: Load registers A and B with different numbers, save Register B in the S stack (PSHS B), add to Register A the contents of the S stack and increase the S stack pointer by 1 back to its original location (ADDA, S+). When we look at Register A we see that it contains F instead of 15. This

is the computer's way of displaying numbers in Base 16, or Hex. In Base 16, 10=A, 11=B, 12=C, 13=D, 14=E, 15=F and 16=10. So our answer of "F" is correct.

Example 4: The same procedure is used to subtract Register B from Register A.

Example 5: Another subtraction problem, but now when we look at Register A we see FB, not -5. The computer doesn't recognize negative numbers, it merely counts backwards from zero. The highest number that can be put into registers A or B is 255, or #FF. Counting backwards from zero in Base 16 you, get #FF, #FE, #FD, #FC, #FB, #FA, etc. The fifth number counting backwards from zero is #FB, so -5=#FB.

Wait a minute — isn't #FB also a positive number? Yes, in Base 10 it is $(F*16)+B$, or $(15*16)+11=251$. How does the computer — or operator — know which number is correct? In most math operations the computer will use "signed" numbers. If the left-most bit of an 8- or 16-bit number written in Base 2 is a 1, the number is considered to be negative; if it is a zero, the number is positive. So in registers A or B, numbers that appear to be greater than 127 may actually be negative numbers for math purposes.

In Example 5 our answer was #FB. Since #FB is greater than 127, we know the answer is actually a negative number. To use the "signed number check" you must convert #FB in Hex to 11111011 in Base 2. (An easy way to convert Hex numbers to the Base 2 is to convert one Hex digit at a time. #F equals 15, or 1111 in the Base 2; #B equals 11 or 1011 in the Base 2, so #FB=11111011.)

To prove that #FB as a "signed" number is the same as -5, use the "Two's Compliment" procedure. Write the number in Base 2, reverse all the 0s to 1s and all the 1s to 0s, then add 1. In doing so, 11111011 becomes 00000100, and adding 1 gives us a result of 00000101. This equals 5, so our answer is -5!

Example 6: This time we loaded registers A and B each with a number and used MUL to multiply them. Since the result could be greater than the space available in just one register, we must read registers A and B together all as one number. Registers A and B together are called Register D, although the D does not appear on the screen. Examine Register D (A and B) to find the answer \$32, which is $3*16+2$, or 50.

Example 7: In this example, we have to read all of Register D to find an answer of \$2710, which is $(2*16)**3 + (7*16)**2 + 1*16 + 0$, or 10000. (** is used here as a symbol for exponential.)

Example 8: Before we check the answer, an important point to remember is that MUL does not use "signed" numbers. So when we try to put in -100 or #9C, the computer uses #9C=156 and multiplies that by 100. The answer of Hex 3CF0 in Register D is 15600. Even larger numbers can be multiplied by using a ROM subroutine at Address \$9FB5.

Example 9: Load Register D with a Hex number and Register X with another Hex number. Jumping to the subroutine at \$9FB5 (JSR \$9FB5) will put the product of Register D times Register X in Register Y and Register U. If we read Register Y and Register U together, the product is \$06260060. You can check the result this time by converting the answer to the Base 10. Does this subroutine work with negative numbers?

So far, we've been working only with whole numbers, but that rarely occurs in math. The computer has two locations where it stores numbers up to nine digits long, including positive and negative numbers. These locations are called Floating Point Accumulator 1 and 2, or FPI and FP2. To

Example 1		Example 1A		Example 2	
ORG	\$3000	ORG	\$3000	ORG	\$3000
LDA	#5	LDA	#5	LDA	#5
STA	\$D0	ADDA	#4	STA	\$D0
INC	\$D0	SWI		DEC	\$D0
LDA	\$D0			LDA	\$D0
SWI				SWI	
Example 2A		Example 3		Example 4	
ORG	\$3000	ORG	\$3000	ORG	\$3000
LDA	#5	LDA	#10	LDA	#10
SUBA	#4	LDB	#5	LDB	#5
SWI		PSHS	B	PSHS	B
		ADDA	,S+	SUBA	,S+
		SWI		SWI	
Example 5		Example 6		Example 7	
ORG	\$3000	ORG	\$3000	ORG	\$3000
LDA	#5	LDA	#5	LDA	#100
LDB	#10	LDB	#10	LDB	#100
PSHS	B	MUL		MUL	
SUBA	,S+	SWI		SWI	
SWI					
Example 8			Example 9		
ORG	\$3000	ORG	\$3000		
LDA	#-100	LDD	##1234		
LDB	#100	LDX	##5678		
MUL		JSR	\$9FB5		
SWI		SWI			

get a "signed" number from Register D to FPI, use the ROM routine at Address \$B4F4; to get a number from FPI back to Register D, use the routine at \$B3ED. Remember, Register D can only hold a whole number.

Certain numbers already stored in the computer's memory that are used to perform internal calculations and their locations are:

Number	Location
-0.50	\$843C
0.25	\$BFC2
0.50	\$BEC0
1.00	\$BAC5
10.00	\$BB7D

Each number is five bytes long and is in floating point format.

To handle internal calculations, the computer has five ROM routines that add, subtract, multiply, or divide. The

symbol X means "the number at Location X." These routines are:

Function	Address
FP1=(X)+FP1	\$B9C2
FP1=(X)-FP1	\$B9B9
FP1=(X)*FP1	\$BACA
FP1=(X)/FP1	\$BB8F
FP1=FP2/(X)	\$BB88

Example 10: Load Register D with the number 6, put that in FP1 (JSR \$B4F4); load Register X with the location of the number 10 (LDX #\$BB7D), add the number at the location in Register X to FP1 (JSR \$B9C2). Transfer the result back to Register D (JSR \$B3ED). Examine the register to find Register D is Hex 10, or 16.

Example 11: Use the routine at \$B9B9 to subtract FP1 from the number at the location in Register X.

Example 12: The routine at \$BACA is used to multiply the number at the location in Register X by FP1. The answer of 3C in Register D is 60.

Example 13: The routine at \$BB8F divides the number at the location in Register X by FP1. Since Register D can only hold whole numbers, it shows a 1. The complete answer is in FP1, so we need a routine to display the entire answer instead of rounding it off to Register D. By using a combination of ROM routines, we can display the complete answer, including negative numbers, up to nine digits. (Putting a number back into Register D will lose the number in FP1.)

Example 13A: Type and run this example to see the entire

Example 10		Example 11		Example 12	
ORG	\$3000	ORG	\$3000	ORG	\$3000
LDD	#6	LDD	#6	LDD	#6
JSR	\$B4F4	JSR	\$B4F4	JSR	\$B4F4
LDX	#\$BB7D	LDX	#\$BB7D	LDX	#\$BB7D
JSR	\$B9C2	JSR	\$B9B9	JSR	\$BACA
JSR	\$B3ED	JSR	\$B3ED	JSR	\$B3ED
SWI		SWI		SWI	

Example 13		Example 13A	
ORG	\$3000	ORG	\$3000
LDD	#6	LDD	#6
JSR	\$B4F4	JSR	\$B4F4
LDX	#\$BB7D	LDX	#\$BB7D
JSR	\$BB8F	JSR	\$BB8F
JSR	\$B3ED	JSR	\$BDD9
SWI		LEAX	-1,X
		JSR	B99C
		JSR	\$B958
		SWI	



It's fun making your own Greeting Cards, Signs, and Banners.

Coco Graphics Designer Only \$29.95

The Coco Graphics Designer produces beautiful Greeting Cards, Banners, and Signs for holidays, birthdays and other occasions.

The program features picture, border, and character font editors, so that you can modify or expand the already built in libraries. Plus a special "grabber" utility is included to capture areas of high resolution screens for your picture library.

Requirements: a Coco I, II or III with at least 32K, one disk drive, BASIC 1.0/1.1, ADOS 1.0/1.1 or JDOS. Printers supported include: Epson RX/FX, Gemini 10X, SG10, NX10, DMP 100/105/110/130/430 CGP220, many Okidata (check with Zebra), Seikosha GP100/250, Gorilla Banana, Legend 808. Order #C323 Coco Graphics Designer

Picture Disk #1

This supplementary picture library diskette contains over one hundred additional pictures.

#C333 Picture Disk #1 \$14.95

Colored Paper Packs

150 sheets (50 each red, yellow, blue) with 60 matching envelopes. Perfect for making your productions outstanding.

#C274 Paper Pack \$19.95

Three New Picture Disks

We've hired freelance professional artists to expand the selection of pictures, and fonts available for our Coco Graphics Designer. We think you'll agree that the quality of their work is excellent. Each picture disk contains 128 pictures.

The selection of pictures has been guided by the requests we've received from our many Coco Graphics Designer customers. If we've missed drawing pictures for subjects that interest you, please submit your requests for our consideration.

Picture Disk #2 \$14.95

Special Occasions:

Party Hat, Cake, Gift Box, Champaigne, Juke Box, Saxophone, etc.

Sports: Baseball, Basketball, Tennis, Running, etc.

Office: Computer, File Cabinet, Memo Pad, Clip Board, etc.

American: Flag, Eagle, Astronaut, Indian, Liberty Bell, etc.

Picture Disk #3 \$14.95

Religion: Church, Cross, Candles, Menorah, Bible, Star, etc.
Animals: Dogs, Cats, Tiger, Cow, Giraffe, Birds, Elephant, Turtle, Pig, Horse, etc.

Nature: Flowers, Trees, Sunsets, Mountains, Lakes, etc.

Travel: Car, Bus, Airplane, Taxi, Gas Pump, Tickets, etc.

Picture Disk #4 \$14.95

Includes these holidays and others...

Christmas: Tree, Star, Wreath, etc.

Easter: Egg, Bunny, Lillies, etc.

New Years: Calendar, Fireworks

Chanukah: Menorah, Star, etc.

Halloween: Pumpkin, Witch, etc.

Independence Day: Liberty Bell, Independence Hall, Fireworks, etc.

Presidents Day: Lincoln, Washington, etc.

Ground Hog Day: Ground Hog, Etc.

Two New Font Disks

Font Disk A \$14.95

Contains 10 Fonts

Font Disk B \$14.95

Contains 10 Fonts

NOTE: Our WICO Trackballs and Coco Car Sign Designer are still available. See our ad in the previous issue of Rainbow.

Ordering Instructions: All orders add \$3.00 Shipping & Handling. UPS COD add \$3.00. VISA/MC Accepted. NY residents add sales tax.

Zebra Systems, Inc.
78-06 Jamaica Ave.
Woodhaven, NY 11421
(718) 296-2385

answer displayed with all nine digits (and in the Base 10). The following routines are used in this example:

Function	Address
Transfer FP1 to a buffer at \$3DA	\$BDD9
Decrease location for sign	LEAX -1,X
Print buffer contents	\$B99C
Print a carriage return	\$B958

The main FP routines are:

Function	Address
Transfer FP1 to FP2 (and FP1)	\$BC5F
Transfer FP2 TO FP1 (and FP2)	\$BC4A
Register B (-128 to +127) to FP1	\$BC7C
Register B (-128 to +127) + FP1	\$BD99
Register D (-32768 to +32767) to FP1	\$B4F4
(X) to FP1	\$BC14
(X) to FP2	\$BB2F
FP1 to (X)	\$BC35
10*FP1	\$BB6A
-1*FP1	\$BEE9
FP1/10	\$BB82
FP2/FP1	\$BB91

Some other numbers stored in the computer's memory are:

Number	Location
-32768	\$B3DF
PI/2	\$83AB
2*PI	\$BFBD or \$BFE1
SQR(2)/2	\$8432
SQR(2)	\$8437
LOG(2)	\$8441

The \$B4F4 routine is used in most function commands. Once a number is in FP1, you can jump to the routine for any of the following:

Function	Location
SIN	\$BF78
COS	\$8378
TAN	\$8381
SQR	\$8480
EXP	\$84F2
LOG	\$8446
RND	\$BF1F
INT	\$BCCE

You can print the results in FP1 or use the routine at \$B3ED to put the results back in Register D, but only as a whole number. For practice, develop a machine language program that will print the SIN of any angle you load into Register D. Remember, angles must be converted to radians, since trigonometric functions in the Color Computer are in radians. (Hint: one degree = 2*PI/360 radians.)

Now for our comparison programs. Both will perform a sort of numbers and graphics from screen locations &H400 to &H5FF. You may want to time each program. Is there a difference in the random portion? Is there much difference in the sorting portion? We will review these programs in the next article. Meanwhile, try making up your own programs using the ROM routines we've discussed.

(Questions or comments concerning this tutorial may be directed to the author at Route 2, Box 216C, Mason, WI 54856-9302. Please enclose an SASE when requesting a reply.) □

Listing 1: SORTBAS

```

11Ø CLS
12Ø FOR X=&H4ØØ TO &H5FF
13Ø POKE X,RND(255)
18Ø NEXT
2ØØ FLAG=1
22Ø FOR X=&H4ØØ TO &H5FE
23Ø A=PEEK(X) : B=PEEK(X+1)
25Ø IF A<=B THEN 3ØØ
27Ø TEMP=A:A=B:B=TEMP
28Ø POKE X,A:POKE X+1,B
29Ø FLAG=Ø
3ØØ NEXT
32Ø IF FLAG=Ø THEN 2ØØ
34Ø A$=INKEY$:IF A$="" THEN 34Ø
37Ø END
  
```

Introducing the FOCUS™ software system for OS-9™ Level II—serious business for your CoCo3!

FOCUS Applications Hub

A multi-tasking /multi user systems of I/O functions and utilities which features:

- Two level menu system with company code and access code protection.
- Unique filing, sorting, searching and record locking routines.
- Works with most printers, floppy/hard drives and terminals.
- Similar keys and screen formats for all FOCUS-MATE programs.
- Routines for file maintenance, data backup and setup.
- Context-sensitive help screens.

FOCUS-MATE Correspondence Module

An integrated Text Editor, Text Formatter and Mailing List Database:

- Import text or database files for mail merge facilities.
- Control all printer functions, change formats anywhere in text, save formats.
- Preview final text on screen.
- Print with left, right, full or centered justification, tabs, auto headers/footers, page numbering and dictionary lookup.
- Multiple text column capability.

FOCUS-MATE General Ledger Module

A sophisticated General Ledger package for small business.

- All features integrate with other FOCUS-MATE modules.
- Number of accounts and transactions limited only by disk space.
- Auto balance checking, flexible period and year-end procedures, profit/cost center consolidated ledger and batching capabilities.
- Reports: Balance Sheet, Trial Balance, P&L Statement, Transaction Journal, Transaction Aging.

OS-9 Programmers: FOCUS is a great tool-box for system developers!

with purchase of OS-9 Level II and FOCUS, boot file is free, multi-task on 128K! OS-9 Level II \$64.95

	Reg.	Now
FOCUS	\$65.95	\$55.95
Correspondence Module	49.95	39.95
General Ledger Module	49.95	39.95
FOCUS Technical Manual	19.95	15.95

Easy Street[™]
Data Systems

4500 19th St. #530
Boulder, CO 80302
(303) 440-4767

Add \$3.00 for shipping/handling
TERMS: COD, Money Order, Cashiers checks, Allow 2 weeks for personal check to clear. Sorry no MC/VISA
Hours: 10:00 to 6:00 MST, Tues.-Sat.

Min. Sys. Req.: CoCo3, OS-9 Level II, 360K disk drive, 80 col. display

OS-9 is a trademark of Microware and Motorola Inc.

Listing 2: SORTBIN

3000		00100	ORG	\$3000	
3000	BD	A928	00110	START	JSR \$A928
3003	108E	0400	00120		LDY # \$400
3007	GC	00FF	00130	LOOP1	LDD #255
300A	BD	B4F4	00140		JSR \$B4F4
300D	BD	BF1F	00150		JSR \$BF1F
3010	BD	B3ED	00160		JSR \$B3ED
3013	E7	A0	00170		STB ,Y+
3015	108C	05FF	00180		CMPY # \$5FF
3019	23	EC	00190		BLS LOOP1
301B	86	01	00200	SORT	LDA #1
301D	B7	3040	00210		STA FLAG
3020	8E	0400	00220		LDX # \$400
3023	EC	80	00230	LOOP2	LDD ,X+
3025	34	04	00240		PSHS B
3027	A1	E0	00250		CMPA ,S+
3029	23	07	00260		BLS CONT
302B	1E	89	00270		EXG A,B
302D	ED	1F	00280		STD -1,X
302F	7F	3040	00290		CLR FLAG
3032	8C	05FE	00300	CONT	CMPX # \$5FE
3035	23	EC	00310		BLS LOOP2
3037	7D	3040	00320		TST FLAG
303A	27	DF	00330		BEQ SORT
303C	BD	ADFB	00340		JSR \$ADFB
303F	3F		00350		SWI
3040			00360	FLAG	RMB 1
			00370		END START
	3000				110 CLS
					120 FOR X=&H400 TO &H5FF
					130 POKE X,RND(255)
					180 NEXT
					200 FLAG=1
					220 FOR X=&H400 TO &H5FE
					230 A=PEEK(X):B=PEEK(X+1)
					250 IF A<=B THEN 300
					270 TEMP=A:A=B:B=TEMP
					280 POKE X,A:POKE X+1,B
					290 FLAG=0
					300 NEXT
					320 IF FLAG=0 THEN 200
					340 A\$=INKEY\$:IF A\$="" THEN 340
					370 END

00000 TOTAL ERRORS

Hardware Special Communications Package

300/1200 baud Fully Hayes compatible

Modem - 1 Year Warranty

\$109.00

[Modem & Cable]

300/1200/2400 baud Fully Hayes Compatible Modem - CCITT 1 Year Warranty

\$179.00

[Modem & Cable]



THE OTHER GUYS CoCo

55 North Main Street
Suite 301-D
PO Box H
Logan Utah 84321

Software

'KEEP-TRAK' General Ledger Reg. \$69.95—Only \$39.95

"Double-Entry" General Ledger Accounting System for home or business: 16k, 32k, 64k. User-friendly, menu-driven. Program features: balance sheet, income & expense statement (current & 'YTD'), journal, ledger, 899 accounts [2350 entries on 32k & 64k (710 accounts & entries on 16k) [disk only]. Version 1.2 has screen printouts. Rainbow Review 1.1 - 9/84 : 1.2-4/85

'OMEGA FILE' Reg. \$69.95—ONLY \$24.95

Filing data base. File any information with Omega File. Records can have up to 16 fields with 255 characters per field (4080 characters/record). Sort, match & print any field. User friendly menu driven. Manual included (32k/64k disk only). Rainbow Review 3/85. Hot CoCo 10/85

BOB'S MAGIC GRAPHIC MACHINE

Can generate BASIC code to use in your programs. Easy drawing and manipulation of circles, ellipses, boxes, lines and ARCS. Single joystick operation with on line HELPS at all times. Allows text on the graphics screen & movement of objects on the screen. Can be used as a stand-alone graphics editor. Instruction Manual. **GRAPHICS EDITOR. REG. \$39.95—ONLY \$24.95** for disk or tape. 64k ECB.

Rainbow Review 7/85, Hot CoCo 9/85 "The graphics bargain of the year"

'KEEP-TRAK' Accounts Receivable.

Features: auto interest calculation, auto ageing of accounts, installment sales, total due sales, explanation space as long as you need, detailed statements, 'KEEP-TRAK' General Ledger tie in, account number checking, credit limit checking & more. User friendly/menu driven. Includes manual. **\$39.95** or **\$49.95** General Ledger & Accounts Receivables. [Disk Only].

'COCO WINDOWS'

With hi-res character display and window generator. Features an enhanced key board (clicks) and 10 programmable function keys. Allows the user to create multiple windows from basic. Includes menu driven printer setup and auto line numbering. Four function calculator, with memory. The above options can be called anytime while running or writing in BASIC. APPLE PULL YOUR DRAPES. YOU DON'T WANT TO SEE THIS. **\$24.95** (disk or tape) includes manual.



(801) 753-7620
(800) 942-9402

NEW FREE CATALOG

[Add \$3.00 for postage & handling]

C.O.D., Money Order, Check in U.S. Funds [Please specify if J&M controller]