

ENGINEERING NOTES  
on  
Radio Shack Color Computers

September 1984  
Vol. 1 No. 7

\$1.95

```
****      *  *      *  *      *      *  *      ****      ****
*  *      *  *      *  *      *  *      *  *      *  *      *  *
*  *      *  *      *  *      *  *      *  *      *  *      *  *
*  *      *  *      *  *      *  *      *  *      *  *      *  *
*  *      *  *      *  *      *  *      *  *      *  *      *  *
****      *  *      *  *      *  *      *  *      *  *      ****
```

```
***      ***      *      ***      ****
*  *      *  *      *  *      *  *      *  *      *  *
*  *      *  *      *  *      *  *      *  *      *  *
*  *      *  *      *  *      *  *      *  *      *  *
*  *      *  *      *  *      *  *      *  *      *  *
***      ***      *  *      ***      ****
```

```
*  *      *  *      *  *      *  *      *  *
*  *      *  *      *  *      *  *      *  *
** *      *  *      *  *      *  *      *  *
*  *      *  *      *  *      *  *      *  *
*  *      *  *      *  *      *  *      *  *
*  *      *  *      *  *      *  *      *  *
*  *      *  *      *  *      *  *      *  *
```

\* POWERFUL REMARKS  
(Part 3)

\* COMPUTER THEORY  
\* BASIC PROGRAMMING

\* MACHINE LANGUAGE PROGR.  
\* QUESTIONS & ANSWERS

DYNAMIC COLOR NEWS is published monthly by DYNAMIC ELECTRONICS, INC., P.O. Box 896, Hartselle, AL 35640, phone (205) 773-2758. Bill Chapple, President; Alene Chapple, Sec. & Treas.; John Pearson, Ph. D. Consultant; Bob Morgan, Ph. D., Consultant.

Entire Contents (c) by DYNAMIC ELECTRONICS INC., 1984. DYNAMIC COLOR NEWS is intended for the private use of our subscribers and purchasers. All rights reserved. Contents of this newsletter may not be copied in whole or in part without written permission from DYNAMIC ELECTRONICS INC. Subscriptions are \$15/yr for U.S.A. & Canada, \$30 other foreign.

The purpose of this newsletter is to provide instruction on Basic & Machine Language programming, Computer theory, operating techniques, computer expansion, plus provide answers to questions from our subscribers.

The submission of questions, operating hints, and solutions to problems to be published in this newsletter are encouraged. All submissions become the property of Dynamic Electronics if the material is used. We reserve the right to edit all material used and not to use material which we determine is unsuited for publication.

All paid subscribers are entitled to discounts of 10% on hardware, and 20% on software manufactured or produced by Dynamic Electronics Inc. plus "specials" mentioned in the newsletter. To receive these discounts use your DCN number which is at the right of your name on the address label. DCN subscribers may obtain a personalized reply to specific computer problems or advice on purchasing equipment. The charge for this service is \$10.

```

*****
*
*   DYNAMIC   COLOR   NEWS
*
*       September 1984
*
*   Editor and Publisher
*       Bill Chapple
*
*       Secretary
*       Helen Earls
*
*****

```

CONTENTS

Editor's Comments . . . .	3
MACHINE LANGUAGE PROG. . . .	4
Powerful Remarks . . . .	5
(Part 3)	
Questions & Answers . . . .	7
Product News & Specials . .	8

## EDITOR'S COMMENTS

I can feel the change of seasons as kids start back to school and football games appear on the television. Changes are what I have gotten accustomed to as an engineer. I remember when I was a technician I was a pretty good expert on troubleshooting vacuum tube circuits. Then transistors appeared. This really upset old timers who had become experts in troubleshooting vacuum tube circuits. Many retired but those who stayed had to learn how to handle equipment with transistor circuits.

My technician training was obtained at Southern Tech in Marietta, GA in the late 1950's. In the mid 60's I was teaching electronics at a junior college and learned transistor circuits on my own so I could teach my class. I obtained a good understanding of the semiconductor theory and since we had lab equipment I did a lot of experimentation. This really helped when I went through the engineering courses at the University of Alabama in Huntsville as I had a very good understanding of how semiconductors work.

Next came integrated circuits both linear and digital types. Change had crept in again and it was necessary to learn how to design circuits using these packages. An integrated circuit consists of many transistor circuits in a very small package. What was in the package was not important but how to get signals in and out of the packages was the name of the game. Systems using integrated circuits might require 10 to 50 separate chips to do a particular job.

Then I was hit with a blow that really stunned me for a while. Microprocessor chips were becoming available and could be made to do just about any task by programming one chip, the read only memory (ROM). Now all designs could be essentially the same eliminating hardware problems. What if you didn't know how to design the patterns for the ROMs? Back to the books.

So you see I am used to change. Sometimes it is painful but usually technological changes are for the better. Now we have microcomputers and they are becoming more and more powerful. They are used extensively for process controllers. The engineering on a new system is now reduced to designing the input and output circuitry to interface devices such as motors, controllers, and measuring instruments. Of course the microcomputer is a natural when it comes to recording data such as temperature and voltage.

Consider the field of communications. I have been in amateur radio since 1954 and my call letters are W4GQC. For sending or receiving Morse Code, I can let the microcomputer do the work. I can compose the messages on the microcomputer, and store or print received messages. If I want to use voice then I can let the computer talk with a voice adapter. Also I can transmit and receive pictures using slow scan television. It is pretty easy to keep notes of contests or keep records of people with whom I have communicated. The computer has become an important function in my ham radio shack.

Most people are familiar with the conventional uses for computer. With them you can play games, keep records, do your check book, write editorials, etc. The computers of the future will be even more powerful. Prices of disk drives have dropped over the past year and half size drives are now available. Also the price of 64K memory chips has decreased from what it was a couple of years ago. Static RAMS, bubble memories, and electrically erasable memories will be used more in computers as the prices drop.

All of these changes will mean more computer power for you. With

the expandable color computers any of the new features can easily be incorporated.

## PRODUCT INFORMATION

We have reduced the price of our 128K memory expanders by \$30 each. We are also running specials on them through October so if you are considering upgrading to 128K then you will want to take advantage of the specials.

Also we have developed an auxilliary cartridge port. This means that you can have two cartridges plugged into the computer at once and select the one you want with the included switch or use a cartridge with your disk drive. All of the signals are not available but enough are to run most cartridges. This is also on special through October.

We are adding accessories from other suppliers and will give discounts to DCN suppliers. A new catalog will be out soon and we will send all subscribers one.

## MACHINE LANGUAGE PROGRAMMING

### WRITING POSITION INDEPENDENT CODE (PIC)

Position independent code means that a machine language program will run anywhere within the memory map. There are some things to look out for when writing this type code and we want to discuss the problems and show how to write programs that will work in any memory location.

### SCRATCH PAD MEMORY (SPM)

Almost all machine language programs require some memory for holding results, vectors, or for doing calculations and operations on bytes. These areas are called scratch pad memories. What memory areas are available for scratch pad use? The lower 1535 bytes are reserved for the computer and the first graphics page and the upper 2K bytes or so should be reserved for the stack registers. We are assuming that the programs are being written on 32K computers. There are two ways to define the scratch pad memory area for machine language programs.

The first method is to designate fixed memory areas such as 4000 to 4100 for scratch pad use. This works fine as long as it is out of the way of any other program. Picking a memory area that will not conflict with other programs can be a problem. For example suppose you want both a basic and machine language program to be loaded at the same time. Then the scratch pad area should not be required by the basic program.

The second method is to link or index the scratch pad area to the beginning location of the machine language program. For example let's assume that the scratch pad area is 200 bytes below a machine language program. Then if the program is located at 20000 then the scratch pad area will be at 19800. If the program is at 12800 then the scratch pad memory starts at 12600. This is nice as you don't have to worry about interfering with other memory areas.

The method used depends upon the program. We have used both methods. For example our terminal program "DYTERM" uses a designated scratch pad memory. Our "MULTIPROGRAM MANAGER" software uses an indexed scratch pad memory area that is offset a given number of bytes from the beginning of basic vector plus a fixed scratch pad memory

area.

### MACHINE LANGUAGE PROGRAMS with FIXED SP MEMORIES

To use fixed memories the extended addressing mode will be required. This generally requires one more byte in the instruction. The major disadvantage will be if the SP memory area is to be later changed. Let's take an example of things we might want in the scratch pad memory area.

```
4000 MS of beginning vector
4001 LS of beginning vector
4002 MS of ending vector
4003 LS of ending vector
4004 First variable
4005 Second variable
4006 Third variable
4007 MS of answer
4008 LS of answer
```

The first statement of the program should load a register with the beginning of the memory area (4000). This has to be a 16 bit register. The 16 bit registers are X, Y, U, and S. X and Y are the registers available for the programmer. It takes less bytes to use the X register so let the first statement load X immediate with the value 4000. Now any of the other scratch pad memory areas can be obtained by pointing to the memory contained within X plus an offset of 0 to 8. So to load the A register with the first variable just enter LDA indexed to X with an offset of 4.

### MACHINE LANGUAGE PROGRAMMING with INDEXED SP MEMORIES

Now let's consider the same problem where we use the bytes preceeding the ML program for our SP memory. Let's reserve 10 bytes before the machine language program for this function. The only difference will be in the first assembly language instruction. We will still load the X register but it will be indexed to the program counter. The program counter points to the next instruction. So we will need to load X indexed to PC with an offset of -10 minus the number of bytes in the first instruction. If the first instruction is 3 bytes long then we need to load X offset from the PC by -13.

We are using decimal arithmetic for our discussions so if you have a HEX Assembler then you will need to enter the HEX values of course. Anytime you need to operate on any of the designated variables in the previous table then you can find the value by giving the correct offset from the X register as in the previous example.

Notice X is referenced to the program counter and the memory area is referenced to X. So using this technique the program will work in any memory area as long as there is usable RAM preceeding the machine language program.

## POWERFUL REMARKS PART 3

In the previous discussions we showed how to carry data in remark statements. The advantages are that you can go to any statement and remove the data. We showed how to find the memory locations using

Basic and a machine language subroutine.

This month we want to show how to carry machine language subroutines within the remark statements. You can have a very powerful programming tool if you can write Basic programs and call machine language subroutines when you need speed. It makes things much simpler if the machine language subroutine can reside within a remark statement and become a part of the Basic program. Let's look at the conventional method of carrying machine language programs within Basic programs. If they are carried within data statements they might appear as follows:

```
50 DATA C3,FA,05,89,CF,A1,F0,85,FC,D4,75
```

The data is obviously in HEX format and each data element takes 3 bytes, two for the HEX characters and one for the comma separating data elements. Let's look at carrying data in decimal rather than HEX.

```
60 DATA 138,25,208,125,3,100,75,211,53
```

Notice that this can take 4 bytes if the value of the data element is greater than 100. For either of the two examples given, the data has to be moved to some other location in order for the machine language subroutine to be executed.

What we like to do is to run the machine language subroutine from the remark statement without relocating it to another memory area. How can this be done? The first thing we need is to have a procedure for locating the information within the remark statements. In parts 1 and 2 of this series we showed how to do this by using basic and machine language subroutines. To simplify matters let's carry the machine language subroutines within remark statements near the beginning of the program. In fact let's put them as the first statements within the program. This will make it easier to find the information. If the subroutines do not contain any zeros then they can be run directly from the memory they occupy when the program is loaded. If they contain a zero then a different value such as 32 the ASCII value for a space can be put in that location. Then a routine needs to be written that will put a zero in the locations that require zeros. A word of warning is in order here. If a zero is poked into the remark statement then basic will think that is the beginning of a new basic statement. So if a zero is required in your machine language subroutine it is advisable to relocate the subroutine to another memory area.

Let's look at finding the beginning of the subroutines in the remark statements. For convenience label the subroutines U1, U2, U3, ...UN, etc. U1 will be the first data byte within the first remark statement. U2 will equal U1 + the number of bytes to skip. Let's take an example. Assume the subroutines occupy the number of bytes as shown.

```
U1 - 15  
U2 - 23  
U3 - 13  
U4 - 39  
U5 - 28
```

```
U1 = 256 * PEEK (25) + PEEK (26) + 6  
U2 = U1 + 15 : U3 = U2 + 23 : U4 = U3 + 13: U5 = U4 + 39.
```

**QUESTION:** My computer gets pretty hot. What can I do to reduce the heat?

**ANSWER:** Heat is directly related to power. When more accessories are added then more power is used causing more heat. There are several ways to reduce heat. One method is to install a fan inside the computer or let a fan blow air into the computer. A simple way to reduce heat is to drill a lot of holes in the plastic cover. This lets air come in and the heat escape. Plastic does not provide any radio frequency (RF) shielding so drilling holes does not increase noise or change any of your computer's operating characteristics. We eliminated the heat problem with one of our computers by drilling holes in the front above the keyboard and on each side to let additional air circulate.

**QUESTION:** After my computer is on for a few seconds strange characters appear on the screen. What could be the problem?

**ANSWER:** This could be a heat related problem. One of your chips could be going bad as it is heated up. Try operating the computer without the cover and see if it works ok. This would be a pretty good indication of a heat problem if it works without the cover. The suspected chips would be the 6809E, 6883, or one of the 6821/ 6822 chips. Change the chips one at a time until the problem disappears.

#### PRODUCT NEWS & SPECIALS

We have reduced the price of our 128K memory expanders by \$30 each and are offering special prices to DCN subscribers through October as follows. Our 128K expanders are compatible with all software and come in plug in modules. No circuit board to install.

ME-128-64 all 64 K Computers to 128K	\$145
ME-128F 285 and F computers to 128K	\$180
ME-128D D & E Computers to 128K	\$190

#### NEW PRODUCT SECOND CARTRIDGE PORT (CARTPORT)

Now you can add a second cartridge port to your computer for running programs in cartridges. The CARTPORT can be mounted to the right side of your computer. Drill 2 holes and file a notch in the case to allow the wires to come through. This is a plug in installation with one resistor to cut and two wires to crimp or solder to each side of the cut. Mount the included miniature toggle switch to select the CARTPORT or the normal expansion port. CARTPORT eliminates the need for a multipack or "Y" expander. CARTPORT will run most cartridge programs. List price \$44.95. DCN special through October \$29.95. Allow 2 to 3 weeks for delivery.

Add \$2 shipping US & Canada, \$3 other foreign.

**COLOR COMPUTER SOFTWARE**  
DCN SUBSCRIBERS TAKE 20% DISCOUNT

We save you money because our software works on both tape and disk systems. Extended Basic is not required except for DYFIN. Programs are supplied on tape and can be saved to a disk. Most software is also available in plug in cartridges.

DYTERM - An inexpensive terminal program that allows color computers to interface with modems, terminals, or other computers using the serial ASCII port. Excellent for bulletin boards. Features include

- \* User defined buffer for composing messages off line or storing received messages.
- \* 300 to 2400 baud rates
- \* 1 or 2 stop bits
- \* Low cost \$14.95 tape, \$24.95 cart.
- \* 7 or 8 bit words
- \* Variable parity

DISASM - 6809 Decimal Machine Language Assembler and Disassembler. Especially designed for the beginner, DISASM uses English mnemonics (Symbols for Instructions) instead of hard to remember symbols. DISASM is a real time assembler and stores the machine codes in memory after each instruction is entered. You don't have to remember formats as DISASM asks you for the information needed to complete an instruction if more information is required. The disassembler can be used to display the mnemonics in English and branch locations in decimal for any 6809 machine language program or subroutine. Tape \$19.95, Cart. \$29.95

MPM - Multiprogram manager allows up to 5 programs to be loaded into a 16K or 32K computer. When used with our 96KX you can load 5 programs into both 32K memory pages in 64K computers. Select the program you want to run from the menu. Add or delete programs, and save or load all programs at once as a block. Uses about 2K of memory. \$14.95

DYFIN - Finance Program. DYFIN is designed to aid in the Financial Planning of Loans, Savings, Annuities, Investments, and Retirements. Also included is a 500 year Calender plus a Checkbook Program. Was \$19.95. Now \$12.95 on tape.

UP-1 - This utility program does many of the every day computer operating requirements. Features include:

- \* Write data to a cassette
  - \* Read data from a cassette
  - \* Read values from memory
  - \* Store values in memory
  - \* ASCII to DECIMAL conversion
  - \* Data Relocation
  - \* Vector Combination & Reduction
  - \* Cassette Motor On or OFF with one key
  - \* Machine language subroutine for stacking 2 programs and switching programs with EXEC command
  - \* Allows text to be stored in memory and printed on a printer
- Cassette \$14.95



\*\*\*\*\*  
 \* Please sign me up for one year for the DYNAMIC COLOR NEWS SERVICE. I \*  
 \* understand I will receive a monthly news letter, Discounts on DYNAMIC \*  
 \* ELECTRONIC INC. Computer products plus the Individual Reply to my \*  
 \* Computer problems for a special of \$10 each. Also I understand that \*  
 \* there will be no charge for letters printed with answers in the \*  
 \* Newsletter. Cost \$15 USA & Canada, \$30 foreign. \*  
 \* \*  
 \* Name \_\_\_\_\_ Mail payment to \*  
 \* Address \_\_\_\_\_ Dynamic Electronics Inc \*  
 \* City \_\_\_\_\_ P. O. Box 896 \*  
 \* State & Zip \_\_\_\_\_ Hartselle, AL 35640 \*  
 \* Enclosed is a check \_\_\_ \*  
 \* charge to VISA \_\_\_ MC \_\_\_ Number \_\_\_\_\_ Exp. \_\_\_\_\_ \*  
 \* \*  
 \*\*\*\*\*

**DYNAMIC ELECTRONICS INC.**  
 P. O. Box 896 (205) 773-2758  
 Hartselle, AL 35640