

Here's March!

This month we're including a small questionnaire on what sort of people and equipment are out there in subscriberland. If you could take a moment, fill it out and return it to us, we will have a good idea of the things we can do to serve you better. Not that we're promising anything...



```

*****
*
* Level Title Turns Count
*
* CTR-41 CTR-80
*
* IIIIIIIIII
* II Birthday (cover) 18 & 263 10 & 155
* II Stock Page (part one) 75 & 305 44 & 180
* II Stock Page (part two) 144 & 357 85 & 210
* II YahtZ-80 206 & 406 121 & 238
* IIIIIIIIII
*
* IIIIIIIIII Birthday (cover) 12 & 53 6 & 31
* II II Stock Page (both parts) 92 & 186 54 & 109
* II II YahtZ-80 269 & 304 158 & 179
* II II Nerves 337 & 360 198 & 211
* IIIIIIIIII Level3 383 & 413 226 & 243
*
*****

```

"Birthday" is happy birthday to us, written by our editor of two summers back. Placing an AM radio next to the keyboard unit lets you listen to the somewhat tremulous voice of the computer singing along.

"Stock Page" is a rather effective tutorial on the various magic numbers that appear on the stock page of your newspaper. After walking through a "standard" entry explaining the various terms and how they're calculated, the program starts asking questions. It sets up six of the seven basic variables and asks you a question about the seventh. It took me a while to deduce that the missing number was the one represented by a question mark. Usually the question is which of three companies has the highest or lowest whatever, but occasionally it will set up one company and want an actual number. Warning: although the information to compute various factors is given beforehand, the information given might be missing a necessary piece of data that must itself be calculated. After the answer is given, the formula for finding it is given also.

"YahtZ-80" is essentially a poker game played with simulated dice. You are handed a "roll" and have two chances to improve it by identifying a die (by number) and pressing "R". If you're happy before your two rolls are up, press "R" without first selecting a die. You now have the choice of selecting which category to place your hand (twos, threes, full house, etc.). In the course of a game, each player must classify precisely thirteen hands into the thirteen different categories. The computer scores the totals, delivers the verdict, and locks up. To replay, press "BREAK" and type "RUN" again.

"Nerves" is a good exercise for those of you who are suffering from peace, happiness, quiet living and kind thoughts. The object of the game is to get your rat through the maze as quickly as possible, with the minimum number of commands. It is considered poor manners to poke your fist through the screen. As an example of skill, my highest score (cold sober) is 108.

"Level 3" is the prize this month, folks. It is a system file, so it needs memory reserved. To load it, turn on your computer, and enter 31100 for MEMORY SIZE. Then enter "SYSTEM" and "LEVEL3" and play the tape. When (if?) the SYSTEM prompt returns, enter a "/" <enter>. The instructions will slowly form on the screen. BASICALLY, the system traps out error messages and expands them to a more readable form, and also provides the ability to store strings of up to twenty bytes under each alphabet key (it is typed automatically when the key is typed shifted). After this file is loaded, you can proceed to use your computer to program in a normal fashion, keeping in mind that the upper 1667 bytes of memory are already spoken for.

As promised last month, I'd like to finish up on the subject of machine code CALLS with the introduction of interrupts.

We left off having explained the CALL <address> instruction and how it automatically saves its place in the program so it knows where to come back to when told to RETURN. If the machine language subroutine that is CALLED saves all of the registers in memory somewhere, and resets them to the way it found them before RETURNING, the main program will never know that a subroutine was called, and what it doesn't know won't hurt it.

Imagine the following situation: the TRS-80 (lets presume it is a disk system with a real time clock) is merrily computing along in the middle of a game, and the system clock hardware ticks (that's all it can do - tick, tick, tick). The actual clock is built in software - that's where the seconds, minutes, hours and so forth are kept updated. The user is not expected to always run programs which occasionally go out and look to see if the clock has ticked lately. The clock is set up to steal the time from the program that is currently being executed by - guess how? - "interrupting" the computer.

It turns out that there is a pin on the expansion interface connector (pin 21, INT*) that normally floats through life at a +5 volt level. When a piece of external hardware (such as the expansion interface) wants to interrupt the program in progress, it simply pulls the voltage at this pin down to ground (a pushbutton switch would work, too). Pulling this pin low has exactly the same effect as magically inserting a CALL 0038(Hex) instruction.

Address 0038(Hex) is in Read Only Memory, so it is not surprising to find a JP 4012(Hex) instruction there. O.K, let's jump to 4012(Hex) and there's yet another JP instruction, this time to the Real Time Clock subroutine. The Real Time Clock is a subroutine that saves all the registers that it uses and restores them prior to RETURNING, so it is an "invisible" interrupt. The main program doesn't ever feel it.

Let's step through the sequence one last time. The Z-80 chip is churning merrily along in the middle of some program, it doesn't matter what or where. Suddenly there is a Real Time Clock tick, and a hardware gate pulls the INT* line on the expansion connector to zero volts. The

Z-80 chip interprets this as a CALL instruction to address 0038(Hex). It pushes its current program address on the stack, and jumps to address 0038(Hex). There it finds a jump to address 4012(Hex), so it does that. Arriving at 4012(Hex), it finds yet even another jump, this time to the Real Time Clock subroutine. The Z-80 finally arrives at the RTC subroutine, increments the seconds count (or whatever...) and RETURNS to the main program by popping the address from the top of the stack.

Why all that jumping around? Well, the address 0038(Hex) was decreed by the designers of the Z-80 chip. In the TRS-80, that happens to be in the Read Only Memory, so the user cannot POKE a subroutine there. The codesmiths who wrote the level II BASIC decided to jump out to RAM (read-write memory), hence the JP 4012(Hex) instruction. These codesmiths reserved three bytes of RAM at 4012(Hex) so that a user could POKE a subroutine anywhere in RAM, and then POKE a jump to it in these three bytes. When the machine is first turned on, there is no interrupt subroutine anywhere in memory, so the three bytes at 4012(Hex) merely RETURN control back to whatever the main routine is.

One final word. Interrupts can be enabled or disabled with the EI or DI instructions, respectively. Executing a Disable Interrupts instruction will cause the Z-80 to defer looking at the INT* pin until the next Enable Interrupts instruction is executed (which could be never). When the Z-80 chip responds to an interrupt, it automatically disables the interrupts, so the last instruction (right before the RETURN instruction) in an interrupt subroutine should be an Enable Interrupts instruction (Opcode = FB(Hex)).

(over)

.....

Questionnaire: please fill out and return this form to CLOAD Magazine (note - if you object to tearing up your yellow sheets, just send us a note with the multiple guess answers - We'll be keying it all into a computer anyway).

- _____ Type of computer (level I or level II)
- _____ Amount of memory (in K's)
- _____ Do you have an expansion interface? (y/n)
- _____ Printer? (y/n)
- _____ Disk(s)? (y/n)
- _____ Do you do a lot of your own programming? (y/n)
- _____ What is your current main use of your computer? (B=business, E=education, F=fun & games, D=software development, O=other)
- _____ For what use did you purchase your computer? (B=business, E=education, F=fun & games, D=software development, O=other)
- _____ How many different people might use your computer in a typical week?

One hangup that a few of you expansion interface users may have been running into involves what happens when (1) someone has borrowed your printer or you don't own a printer, or your non-Centronics/Radio Shack style printer driver hasn't been patched during this session, and (2) a program you are using executes an LPRINT command. What BASIC will do is go out to the parallel printer port and wait until the port is ready, which takes forever if there's no printer plugged in. The only exit in such cases involves restarting the system. There are several solutions. One is to always have a printer plugged into your parallel port. Another is never to use the LPRINT statement in a program when there's no printer around. For those of you who find both these solutions difficult, there is a "dummy" parallel printer plug available from Fuller Electronics (7465 Hollister #232, Goleta CA 93017) that acts as a bit bucket. It's twelve bucks, which may seem a lot to do nothing but throw away bits, but if you're hung up on hangups, it's far cheaper than psychotherapy.

Also in our Miscellaneous Dinguses (dingii?) column this month, E.B. Garcia and Associates (203 N. Wabash, Rm. 2102 Chicago IL 60601) has come out with a cute little plunger thingie that extends the RESET button in the back of the computer out past the expansion interface cable connector. This replaces the curse - stand up - grab pencil - poke pencil in behind expansion plug sequence that sometimes occurs.

Next month!



Ralph McElroy, Publisher

.....

questionnaire (cont.)

- _____ How many times do you use your computer in a week?
- _____ Have you had trouble loading our cassettes in the last six months or so? (y/n)
- _____ Has the Postal Service lost, crunched, or otherwise done evil things to your cassettes? (y/n)
- _____ How long have you been "into computing"? (months)
- _____ How many people have been introduced to computing on your machine?
- _____ On a scale of 1=poor to 9=good, how would you rate CLOAD overall?
- _____ On the same scale, how would you rate your favorite software package? (besides us, that is)
- _____ Have you had any trouble with any of our advertisers? (if so, which one?) _____
- What has been your favorite program? _____
- What do like most about CLOAD Magazine? _____
- What do you like least about CLOAD Magazine? _____