

X10.COM  
CP/M Software for the  
CP-290 X-10 Interface

Copyright 1989, 1992 by Rick Swenton

User's Guide

---

**X10.COM - CP-290 Interface Software for CP/M**

Version 3.4 - January 24, 1992

**Users Guide**

Copyright 1989,1990,1991,1992 by  
Rick Swenton  
106 Melinda Lane  
Bristol, CT 06010-7176

---

This software and users guide is Copyright 1989,1990,1991,1992 by Rick Swenton and is released for private individual use only. It may not be sold or distributed for profit without the prior express written permission of the copyright holder. A small charge to cover the cost of a disk, printing or mailing may be made but the charge must not exceed actual costs. Public non-profit distribution over computer networks is specifically permitted. Conditions require that the copyright notices remain unaltered and visible and that any alterations you may make to the software must not be publically distributed. Although this program is released for general public distribution, it is not public domain software and all world-wide rights are reserved by the copyright holder.

## Acknowledgement

I would like to thank my friend, Al Hathway, for his inspiration and help in the initial development of this program. Al was my original beta test site and he put X10.COM through its paces on a regular basis.

I would also like to thank my friend, Biff Bueffel, who contributed a significant amount of quality code to X10 along with many innovative ideas and countless hours of uploading and downloading test versions on the BBS.

If it wasn't for their suggestions, X10 would not have as many nice features as it does now.

Rick Swenton  
106 Melinda Lane  
Bristol, CT 06010-7176  
(203) 585-9447 Voice (evenings)  
(203) 673-8752 BBS

## Introduction

The CP-290 Interface is a control unit which is part of the X-10 System for home power control. The X-10 system consists of remotely controlled power modules to operate lamps and appliances or anything you can think of. Among the available modules are plug-in lamp and appliance modules as well as replacement duplex outlets, wall switch units and set-back thermostats. These units accept a digital command sequence through the power lines from a control unit. The control unit may be a "Mini" or "Maxi" control unit as well as a clock timer unit or a radio controlled remote. Finally, there is the CP-290 interface which allows you to perform all of your home power control through your computer.

The formal name for the system is called "X10 Powerhouse". This is the same system sold under other names such as Radio Shack's "Plug 'n' Power". Compatible systems are sold by Sears, Stanley (Lightmaker), Heathkit and X10(USA) itself.

For your information, there are two basic types of modules - lamp modules and appliance modules. The important difference is that lamp modules are for incandescent loads only (light bulbs, no fluorescent lamps or inductive loads). Lamp modules are also dimmers. The brightness level can be controlled by the control unit. Appliance modules are more universal. They have a relay contact and can control inductive loads like TV sets, air conditioners, fluorescent lights, coffee pots, etc. Appliance modules do not respond to the "All Lights ON" command nor do they respond to the "DIM" command. Both units respond to the "ALL OFF" command.

Here are some of the features, in detail, of the X10 system:

All devices share some common features. The X10 system supports 16 separate unit codes in 16 separate house codes to yield a maximum support of 256 modules. You could have more than one module on the same address if you desired.

### Controllers:

There are several X10 control devices which send commands over the power lines. The smallest is the MC460 Mini-Controller. This device can control 8 modules (only the first 8, it can not address modules 9 -> 16). It has switches for "All Lights On", "All Off", "Dim" and "Bright". The SC503 Maxi- Controller has all these features too but supports all 16 addresses instead of only 8. The TR2700 Telephone Responder allows you to control your first 8 modules over the phone and the RC5000 controller allows you to control the first 8 modules by radio remote control.

For timer controllers, there is the MT522 which allows you to manually control 8 modules and timer-control 4 modules to go on or off at specific times up to twice a day. It also has a security mode to simulate random on/off operation while you are away from home. The CR512 has all of the features of the timer-controller built into a clock radio. Both have battery back-up.

Another device which falls into the category of controller is the BA284. X10 calls this unit a Burglar Alarm Interface. Radio Shack calls it a Universal Interface, and I like this name better. This unit is a small box which plugs into any outlet. You would use this unit to control X10 modules with an external voltage or contact closure. You can configure this unit to control a single module of any type. You can also configure it to transmit the "All Lights On" command in addition to a single selected unit code. Finally, you can configure it to flash all your lights on and off, such as would be the case if you interfaced it to your

alarm system. This interface is very versatile.

#### Modules:

The LM465 Lamp Module is a small box which plugs into any AC outlet and can control up to 300 watts of incandescent light. It can dim or brighten the lamp and responds to the "All Lights ON" and "All OFF" commands.

The LM486 (2-prong) and LM466 (3-prong) Appliance Modules are also small boxes which plug into the AC outlet. They control their loads with a relay so they can operate things like TV's and air conditioners, up to 15 Amperes. Appliance modules do not respond to the "Dim" or "All Lights On" commands but they do respond to the "All Units Off" command.

The SR227 is a duplex outlet which can control a 15 A load. This unit is a replacement for your existing wall outlet and unless you know what you are doing, you would be better off enlisting the help of an electrician. Like the appliance modules, the duplex outlet module does not respond to the "Dim" or "All Lights On" commands but it does respond to the "All Units Off" command.

The WS467 and WS4777 (3-way) are wall switch modules. They replace your existing wall switch and can control 500 watts of incandescent lamps. Consult with your electrician on this one too. If your wall switch controls wall receptacles in the room, keep in mind that these modules can only handle light bulbs and they may overheat or could even cause damage to anything other than a light plugged into the receptacle. Like the LM465 Lamp Module, the Wall Switch Modules can dim or brighten the lamp and respond to the "All Lights ON" and "All OFF" commands.

The HD243 (15 AMP) and HD245 (20 AMP) are heavy-duty plug-in modules to control high current 220 volt loads like air conditioners.

Finally, there is the TH2807 Thermostat Set-Back. You think I meant "Set- Back Thermostat" but I didn't! This unit is actually not a thermostat. It is simply a heater. You install it on the wall under your existing thermostat. When it is connected to an appliance module, you can use your control of the appliance module to operate the heater. When the heater heats the air around your thermostat, it "fools" it into thinking that the room is warmer than it really is, so it doesn't call for heat.

#### The Ultimate X10 Controller for Computer Junkies - The CP-290

The CP-290 is an intelligent interface to control your X10 modules with your computer. It connects to your serial I/O port and communicates in a manner very similar to an intelligent modem. Using the CP-290 you have the ability to turn units on or off directly from your computer keyboard as well as the ability to program the CP-290's event timer. There are 128 individual programmable events which can be set to turn things on or off at any predetermined time for selected days of the week. The CP-290 maintains an internal 24-hour 7-day clock. You do not need to keep your computer on or even connected to the interface. The interface will keep on transmitting the programmed event information over the power lines at the correct times. There is an internal 9-volt battery to maintain the programmed information during power failures.

The CP-290 comes with software. Only Apple-IIe, Macintosh or IBM-PC software is available from X10. Other computers are supported by private companies but none, to my knowledge, have any software available for use under CP/M. This is understandable since each CP/M computer accesses its serial I/O ports differently. At the time I purchased my CP-290, I only owned CP/M computers so the IBM or Macintosh software was of no value. It was obvious that if I wanted a comprehensive software package to run my CP-290 interface on CP/M systems, I would have to write one myself.

Recently I discovered a CP/M program for the CP-290 which was obtained through a CP/M Bulletin Board and originated from CPMSIG. This program was written several years ago and appears to have not been developed since. It appears to be functional but it lacks the presentation style of this X10 program. It was written in Small-C by Eric Schuyler in 1987.

## X10.COM - The Program

I am not a professional programmer so this was a rather ambitious project for me. I have been a long time fan of ZCPR3 and have developed a few system utilities in Z80 assembly language and modified my share of BIOS files. I thought this would be a good project to begin to learn how to use Rick Conn's SYSLIB. SYSLIB is a powerful library of utilities in Z80 assembler. It takes much of the grunt work out of creating programs in Z80 code. This speeded my development work considerably. Later on in the development cycle, we figured out how to be able to use the versatile Z3LIB and VLIB routines under plain CP/M. This opened the door to pleasing screen displays using efficient library routines for cursor positioning, reverse video and graphics while still having a program which worked well under ZCPR3 and CP/M.

Because X10.COM was written in Z80 code, a Z80 microprocessor is required to use the program. Nearly all of the CP/M systems in use today are Z80 compatible. This program will not run on an 8080 or 8085 microprocessor.

One of the reasons that it is difficult to write a CP/M application which will run on a wide variety of CP/M systems is that every system has a different hardware serial I/O configuration. In order to make it easy for X10.COM to run on many different computers, I have used the overlay scheme similar to the modem programs IMP and MEX. As a matter of fact, you could use a MEX or IMP overlay for your computer to assist you in creating an X10 overlay. The X10 overlay is actually less complicated than a MEX or IMP overlay. Sample overlays for the Heath H89 and Micromint SB-180 are included. If you need help creating the overlay for your particular computer, it is likely that there is a person in your local CP/M computer club who has experience with modem overlays for MEX or IMP. He should have little trouble creating an X10 overlay for you based on the included samples. More information on the overlay can be found later in this users guide.

## Design Philosophy

If you looked at the Programming Guide which came with the CP-290, you would quickly realize that it is no simple matter to write a program to control the interface. The microprocessor in the CP-290 is a relatively simple controller with only 8K of RAM. The firmware command instructions are not consistent. All have a different number of bytes and most bytes have bit- mapped definitions. This means that for a program to talk to the CP-290, not only must it know how many bytes to send, it must also know which particular bits within certain bytes must be on or off.

Being a longtime fan of 8-Bit CP/M compatible systems, I felt that this program needed two types of operations - Menu and Command Line. The Menu system is perfect for new users. You select a command from the list of menu options and the program prompts you for additional input. Command line operation is perfect for advanced users who may want to take advantage of other programs which can pass CP/M command lines. For ZCPR3 users, MENU and VMENU are very powerful ways to create custom application screens for your X10 system by having MENU or VMENU pass appropriate X10 command lines to the operating system.

While designing the X10 command line operation, I was careful to maintain as much consistency as possible between the commands. After you use the program for a short time, you will become comfortable with the command syntax and you will be able to at least make a very good guess at what the desired command should look like.

While designing the menu operation, I took advantage of the already built-in command line operation. In most cases when you select a menu option, the program will offer you a detailed picture of the required input in the same format that you would use if you were operating the program from the command line. With the menu system, a skeleton command line is displayed and you simply type your data directly under the skeleton. I felt that this approach was better than asking 200 questions - "Do you want this light to come on every Tuesday?", "How about every Wednesday?", "What about Thursday?", etc. Instead of this interrogation, the program simply types MTWTFSS. Right below it you type your input. Either type the letter to select the day, or a dash to skip the day, like this:

```
Computer types skeleton ->   MTWTFSS
You type data under it -->  MT-TF-S
```

Where you only selected Monday, Tuesday, Thursday, Friday and Sunday.

Try the commands from the menu display. Just invoke the program with no arguments - X10 <return> and the program will run in interactive mode and display the menu.

### A Word About X10.COM and ZCPR3

Originally, X10 was a CP/M program which had the terminal-specific screen codes contained in the X10 program overlay. As the program gradually became a ZCPR3 utility, it became apparent that it would be beneficial for X10 to be able to do advanced screen displays by obtaining terminal information from the ZCPR3 TCAP. Doing this would preclude using these advanced screen features by the plain CP/M users. Also, there was so much power in the Z3LIB and VLIB library routines which could only be used under a running ZCPR3 system.

In order to have the best of both worlds, I decided to adopt Biff Bueffel's recommendations that we maintain a "fake" ZCPR3 ENV within X10.COM. At address 300H, there is a complete Z3ENV and TCAP. When X10 is started, a test is made to see if ZCPR3x is running. If it is, X10 will obtain all necessary information from the currently running ZCPR3x ENV. This includes terminal codes, processor speed, quiet flag, and graphic characters. If X10 determines that ZCPR3x is not running, X10 will obtain that information from the internal "fake" ENV. A sample ENV file is provided. If your terminal does not support graphics, you still must define ASCII equivalents. For example, use the "-" character for a horizontal line. Use the "|" character for a vertical line. Use the "+" for intersects, the "#" for hashed, "" for solid, etc. You only have to set this up if you are running plain CP/M. For ZCPR3x users, you must be using an extended TCAP supporting graphics. A sample TCAP file is provided.

Using a "fake" Z3ENV within X10 allows the full and rich use of Z3LIB and VLIB library modules even if X10 is running under plain CP/M. Most of the plain CP/M users of X10 will likely be using a prepared version for the Heath H89 or Kaypro-II or 4, which will run "out of the box".

I know that this approach is along the lines of the now defunct ZCPR3 Type 2 programs but X10.COM is now only around 20K in size and the added memory usage by the 'fake' ENV and TCAP was a small tradeoff compared to the versatility of having X10.COM work under both ZCPR3 and CP/M.



## A Word About Skeletons and Command Syntax

Since some numerical data entries can consist of a variable number of digits, the command syntax can cope with leading zeros so that your data entry can match the skeleton. If you decide not to enter leading zeros, that's OK. Just remember that your command line will not align exactly to the skeleton. Here are two examples of valid command lines for the same EVENT command:

User did not enter leading zeros:

```
Skeleton:      X10 EVENT eee huu hh:mm mtwtfss mod fcn lvl [N]
User Command:  X10 EVENT 1 A1 9:23 MTWTFSS NOR DIM 6
```

User entered leading zeros:

```
Skeleton:      X10 EVENT eee huu hh:mm mtwtfss mod fcn lvl [N]
User Command:  X10 EVENT 001 A01 09:23 MTWTFSS NOR DIM 06
```

As you can see, leading zeros allow your command to align to the skeleton for easier entry and verification.

This is the X10 Menu Display:

```
+-----+
| \\\\\\\\\\\ X10.COM --> X-10 CP290 Computer Interface for CP/M \\\\\\\\\\\ |
| \\\\\\\\\\\ Copyright 1989,1992 by Rick Swenton \\\\\\\\\\\ |
+-----+
```

Current Drive/User: C10:

Version 3.4

```
+-----+
| Primary Commands Secondary Commands |
+-----+
| [N]ow - Immediate direct command [A]rea - Change drive/user |
| [E]vent - Program an event [B]ase - Change base housecode |
| [U]pload - Upload events from disk [D]ownload - Downloadeventsto disk |
| [R]ead - Read and display events [P]rint - Print Events on Printer |
| [C]lear - Clear selected Events [T]ime - Display the Time/Day |
| [F]iles - Display UPLOAD files [S]et - Set the Time/Day |
| [H]elp - Help with specific commands [I]nterface - Diagnostic self-test |
| [M]onitor - Display Activity |
+-----+
| [Q]uit - Exit the program (also X or ^C) |
+-----+
```

Enter selection:

---

NOW - Direct Command: Perform an X10 Direct Command

This allows you to send commands to any X10 module from your computer keyboard.

You will be asked to input the House/Unit code, the Function, and if the Function was "DIM", you will have to specify the Level.

Examples: [You pressed the [N]ow option from the menu]

Reading Menu Files from disk, if available. (More on menu files later.)

Enter House Code: [A -> P], [Q]uit G (You typed "G" <RETURN>)

(Typing "Q" instead of "G" would quit and return to the main menu.)

Enter Unit Code: [1 -> 16, ALL, Reset, Quit, Housecode Menu] ==> 10

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

| \_\_\_\_\_ These are the available unit codes \_\_\_\_\_ |  
| \_\_\_\_\_ Displayed at the bottom of the screen \_\_\_\_\_ |

(You typed "10" and <RETURN>). The "10" will light-up in reverse video (if supported by your terminal) in the unit code display at the bottom of the screen. If you would like to select additional units, just type the number and press <RETURN>. As you select additional units, the selected unit will light-up in reverse video. When you have finished selecting all the desired unit codes, press RETURN.

"Q" will quit and return to the main menu.

"H" will return you to the previous screen to re-enter a new house code.

"All" will specify that ALL unit codes on the requested house code will receive the command.

If you wish to clear the units already selected and start over, press "R" for "reset".

**Enter Function:** [OFF, ON, DIM, Quit, Unit Code Menu] ==> dim  
(You typed "dim", <RETURN>)

(Remember that appliance modules and wall receptacles to not respond to "DIM" commands.)

**Enter Level:** [1=Dim -> 16=Bright, Quit, Function Menu] ==> 12  
(You typed "12", <RETURN>)

The screen will clear and then display a confirmation of your command:

```
+-----+
| \\\\\\\\\\\ X10.COM --> X-10 CP290 Computer Interface for CP/M //\\\\\\\\\\\\ |
| //\\\\\\\\\\\\ Copyright 1989,1992 by Rick Swenton //\\\\\\\\\\\\ |
+-----+
```

House: G Units: 10

Function: DIM Level: 12 -----> Command Accepted.

### Menu Files:

A menu file is a user-created menu screen. This file is simply a document that you create which contains a list of the devices available to control in your house. You can use an ordinary text editor or word processor to create the menu screen. Wordstar, Magic Wand, VDE/ZDE, Pie, etc. are all capable of creating the menu screen.

## The House Menu - HOUSE.X10:

The first screen you should create is the file called HOUSE.X10. HOUSE.X10 will have a list of all available house codes. Even if you only use one house code, having that code displayed in the menu screen will assist unfamiliar users of the program and enable them to figure-out what they have to do to run the system. Here is a sample HOUSE.X10 display:

### My Very Own Computer Controlled House

#### House Code Menu

A - Main Floor	I -
B - Second Floor	J -
C - Basement	K -
D - Outdoors	L -
E -	M -
F -	N -
G -	O -
H -	P -

If you select the [N]ow command from the X10 Main Menu, the program will check to see if the file HOUSE.X10 exists on the default drive. If it does, it will display the contents of the file on the screen. This is a very convenient way of showing the user what house codes are available for selection. Remember to keep the size of the menu small enough to fit on a single screen or information will scroll off and disappear.

You can highlight any desired text in the menu (such as reverse video, if supported by your terminal and correctly defined in the TCAP) by using the control characters ^A (control-A) and ^B (control-B) to start/stop the highlighting. Use your word processor to directly enter the control characters. For example, under WordStar, a ^A would be entered by pressing "control-P" then "A". Only the ^A will appear on the screen. Here is an example of how it would look under WordStar:

^A This text is highlighted. ^B This text is not.

## The Unit Menus - UNITx.X10:

The next menu files you should create are the 16 Unit files. These are named UNITA.X10 through UNITP.X10. You don't have to create all 16 of them. Only create the ones you need. For example, if you only use House Codes B and H, then only create UNITB.X10 and UNITH.X10. Here is a sample UNITA.X10 menu file:

### House Code A Power Modules

1 - Living Room	9 - Dining Room
2 - Hallway	10 - Family Room
3 - Kitchen	11 - Den Overhead
4 -	12 - Den Table
5 - Kitchen Radio	13 -
6 -	14 -
7 -	15 -
8 -	16 -

Remember that these menus are just like ordinary documents or text files and that the program does not check to see if they are accurate, nor does the program extract any information from them. They are just a convenience display to the user.

### Upload Files

An Upload File is nothing more than a text file consisting of X10 Event Commands. An Upload File can be created either by using the X10 DOWNLOAD command which will write the file with the contents of the currently stored events or by using an ordinary text editor and creating it from scratch.

Upload Files have the extension .XUL (X10 UpLoad).

If you use a text editor to create the Upload File, you can enter several special commands which perform useful tasks during the upload.

- 1) If the first byte (the first character of the first line) of the upload file is the letter C (or CLEAR), then the program will cause ALL currently loaded events to be cleared before the UPLOAD file events are loaded. (Otherwise, the upload data will be overlaid on top of any currently loaded events.
- 2) Comments may be placed in the upload file. Lines beginning with a semi-colon (;) are treated as comments. Comments may also be appended to the end of event. The comment begins with a semi-colon.

- 3) Events may be cleared from within the upload file. Any line beginning with the ampersand (&) is treated as a clear command. The syntax is &sss eee where sss is the starting event number to clear and eee is the ending event number to clear. sss and eee have the range 001 to 128. (i.e. &010 020 will cause events 10 through 20 to be cleared.)
- 4) Each event is entered, one per line, using the same format and syntax as the X10 EVENT Command. See the description for the EVENT Command. Here is an example of a typical line in the Upload File:

21 E11 16:35 MT--FSS NOR ON

will program event 21 to activate the module  
set to House E, Unit 11 at 4:35 PM every day  
except for Wednesdays and Thursdays.

X10 Upload Files can be displayed with the [F]iles Command. Any file with the .XUL extension in the current drive/user area will be displayed. Use the [A]rea Command to change to a different drive/user area.

#### Interface Diagnostic Test

The CP-290 Diagnostic Test command will cause data stored in the CP-290 to be erased. Before starting the Interface Diagnostic Test, X10.COM will save the time, base housecode and all events. After the test is over, X10.COM will restore the base housecode and time. It will also re-load (UPLOAD) any events which were stored before the test. If the CP-290 fails the Diagnostic Test, it may have suffered hardware or firmware damage and may not function reliably.

CP/M Command Line Operation:

To run the NOW command from the CP/M command line:

X10 NOW huu fcn lvl

| | |  
| | +-- Dim Level - 1 (dim) --> 16 (bright)  
| +----- Function - ON, OFF or DIM  
+----- House/Unit Code - A ->P, 1 -> 16 or ALL

"huu" is the House/Unit Code. This code can range from A1 through P16 or from "AALL" through "PALL" where the first letter is the house code and the last three characters "ALL" means all units in that house code.

"fcn" is the Function Code. This can be ON, OFF or DIM. Remember that appliance modules do not respond to DIM commands.

"lvl" is the Level to Dim. It can range from 1 (dim) to 16 (bright) and only applies if you selected DIM for the Function Code.

Example: X10 NOW D7 DIM 10

will turn on the module set to House D, Unit 7  
and dim the lamp to level 10

As of X10 Version 3.1, extended NOW commands support more than one unit code on the command line.

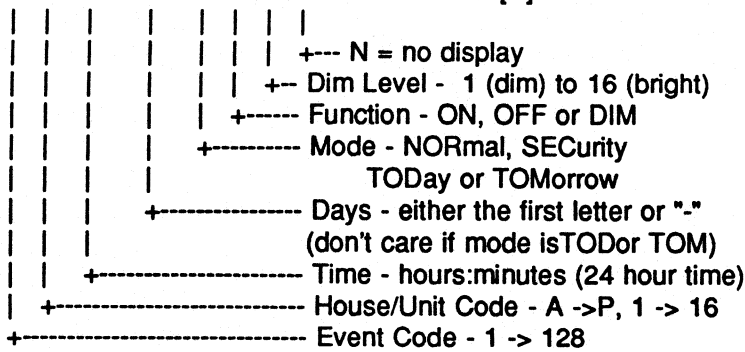
Example: X10 NOW D7 8 9 15 DIM 10

will turn on the modules set to House D, Units 7,8,9 and 15  
and dim the lamps to level 10

---

EVENT - Program an X10 event

X10 EVENT eee huu hh:mm mwtfss mod fcn lvl [N]



There are 128 events which can be programmed into the CP-290.

"eee" is the Event Code to program.

"huu" is the House/Unit Code and can range from A1 through P16.

"hh:mm" is the desired Time in 24 hour format. The colon is needed.

"mwtfss" is the desired days of the week. Upper or lower case is acceptable. Use the first letter of the day of the week to select that day, or use a dash "-" to skip that day. For example, to select every day, type "MTWTFSS". To select only weekends, type "-----SS". (Note: If using either TODAY or TOMorrow Modes below, the "mwtfss" data is ignored and if present is not relevant.)

"mod" is the Mode. There are four available modes.

Modes: NORmal mode - The unit will respond to the event on those days specified in the "Days" field.

SECurity Mode - The unit will respond to the event on those days specified in the "Days" field but the actual time will vary within the programmed hour to give the appearance of random activity, a "lived-in" look. The actual time of the event will vary from day to day.

TODay Mode - The unit will respond to the event today and then will be cleared from memory.

TOMorrow Mode - The unit will respond to the event tomorrow and then will be cleared from memory.

Type "NOR" "SEC" "TOD" or "TOM" for your mode selection.



"fcn" is the Function Code. This can be ON, OFF or DIM. Remember that appliance modules do not respond to DIM commands.

"lvl" is the Level to Dim. It can range from 1 (dim) to 16 (bright) and only applies if you selected DIM for the Function Code.

"N" is an optional control flag and is mainly useful when running X10 from the CP/M command line. If you type "N" at the end of the EVENT parameters, X10 will not display the events on the screen when it has finished programming the event. This would be desired if you wish to program several events at a time and do not want to take the time to display them until the last event has been entered.

EVENT Example:

```
X10 EVENT 21 E11 16:35 MT--FSS NOR ON
```

will program event 21 to activate the module set to House E, Unit 11 at 4:35 PM every day except for Wednesdays and Thursdays.

X10.COM will check to see that all the data is valid and is within proper limits. The program will complain about a missing colon in the time and any other invalid data.

As of X10 Version 3.1, extended EVENT commands now support more than one unit code on the command line.

EVENT Example:

```
X10 EVENT 21 E11 12 16 16:35 MT-FSS NOR ON
```

will program event 21 to activate the modules set to House E, Units 11, 12 and 16 at 4:35 PM every day except for Wednesdays and Thursdays.

---

TIME - Display the X10 internal Time, Day and Base Housecode  
X10 TIME

The CP-290 keeps track of the current time in 24 hour format and the current day of the week. The TIME command will show you the current time and day as well as the Base Housecode. The Base Housecode is the housecode which is controlled by the eight rocker switches on the front of the CP-290. You can use the "B" command from the main menu to select which housecode the front panel switches will control. Remember that if you change the base housecode, you will erase any programmed events. Save your events to disk before you change the base housecode so that you can easily re-load them.

---

**SET - Set the X10 internal Time and Day**

If your system has a hardware clock running under Datestamp or ZSDOS/ZDDOS, you can use the syntax X10 SET SYSTEM to use your system time/day.

**X10 SET hh:mm day**

    |  |  
    |  +---- Day of week (mon, tue, wed etc)  
  +----- Time - hours:minutes (24 hour time)

**Example:    X10 SET 18:45 TUE**

          will set the internal clock to 6:45 PM, Tuesday

X10.COM will check to see if the hours, minutes and day are valid entries. It will also complain if the colon is missing.

**Example:    X10 SET SYSTEM**

          will set the internal clock to the same time and day  
          currently in effect in your Datestamp or ZSDOS/ZDDOS  
          system.

---

**READ - Display the X10 programmed events**

**X10 READ**

This command will cause the CP-290 to transmit the contents of its Event Timer to the host for display on the screen. During many of the other X10.COM commands, the READ function will be requested. The program does not assume it knows what is in the Event Timers and always requests a read to be sure. It takes a few seconds and can become longer if all the events are used.

If you would like a hard copy of the Event Timer contents, use the PRINT command. If you would like the contents of the Event Timer saved to disk, use the DOWNLOAD command.

---

**PRINT - Print and Display the X10 programmed events**

**X10 PRINT**

This command performs the same function as the READ command except that a listing will be sent to your default printer. A FORMFEED will be sent after the listing is complete.

---

DOWNLOAD - Download X10 Events to a disk file

X10 DOWNLOAD [du:]filename[XUL] [N]

du: = optional drive/user area - can range from A0: through P15:

N = No, don't ask to overwrite existing file

This command will save the contents of the Event Timer to a disk file. The resulting file is a text file which can be typed or edited and reloaded with the UPLOAD command. Each event is saved as a line of data followed by a carriage return (CR) and Line Feed (LF). The very last line of the file has a single "#" character which X10.COM understands to mean the end of the data. After a DOWNLOAD, there will be nothing present after the "#". Any information you place there with an editor will be ignored during subsequent uploading with the UPLOAD command.

Example:           X10 DOWNLOAD B3:WINTER

will save the current contents of the event timer  
to Drive B:, User 3 under the filename WINTER.XUL

---

UPLOAD - Upload X10 events on disk to the interface

X10 UPLOAD [du:]filename[XUL]

du: = optional drive/user   .XUL is default file type

This command will cause a disk file of timer events to be uploaded to the CP-290 Event Timer. The events will overlay the existing contents. This means that existing events will be preserved only if there is not an event in the disk file which matches the event number of an existing event in memory.

For example, let's say that you currently have events 1,2,3 and 4 active in the CP-290 memory. You also have a disk file which will program events 9,10,11 and 15. When you upload the disk file, the CP-290 will have all eight events active.

Now, suppose you have the same starting condition with events 1,2,3 and 4 active and the disk file has data for events 3,4,5 and 6. After the upload, the CP-290 will have events 1,2,3,4 and 5 but the old events 3 and 4 were replaced with the data from the new events 3 and 4 from the disk file.

Example:           X10 UPLOAD A12:SUMMER

will add the contents of the file  
SUMMER.XUL on Drive A:, User 12 to the  
contents of the event timer

Upload files can contain comments. Comments are lines which begin with a semi-colon (;). Comments can also be appended to an event by starting the comment with a semi-colon. If the very first character of the Upload file contains a "C", all of the currently programmed events will be cleared before the upload is made. Any line beginning with an ampersand (&) will be interpreted as a 'clear range' command. The line &10 20 will cause events 10 through 20 to be cleared.

---

---

CLEAR - Clear X10 programmed events

X10 CLEAR sss eee [N]

| | |  
| | +--- (optional) N = don't ask for second chance  
| +----- ending event number (1 -> 128)  
+----- starting event number (1 -> 128)

This command will erase any events contained in the CP-290 within the desired range. You specify the starting and ending event to erase. To erase a single event, use the same value for the starting and ending event number.

Example: X10 CLEAR 4 17

will clear any active events contained  
in locations 4 through 17 of the event timer

---

BASE - Change the Base House Code

This function can only be used from the menu in interactive mode.

The program will save the current time and day. It will also save any currently loaded timer events to a disk file and restore everything after the new base housecode is loaded.

The base housecode is the housecode which will be controlled by the front panel rocker switches.

---

DISPLAY - Monitor Activity

This is a new function added in Version 3.3

DISPLAY will show all commands sent by the CP-290 into the power line. These include commands caused by pressing any of the CP-290 front panel rocker switches as well as commands generated from programmed timer events.

---

The X10 command syntax for operation on the CP/M command line is almost identical to the syntax when running from the menu system. The main difference is that under the menu system, the command name is not entered and you are shown a template of the command along with an explanation of the necessary data.

With all the X10 commands, only the first letter is required for the desired command. For example, "X10 R" is the same as "X10 READ".

---

**Interface Cable Definitions:**

The CP290 cable only uses three wires and a shield.

				----- Heath H89 Serial Ports -----				
CP290 Pins	Wire Color	Data Flow	Port	330 Pins	Port	320 Pins	Port	340 Pins
2	RED	<--	2	2	3	3	3	3
4	YEL	-->	3	3	2	2	2	2
3	BLUE	<-> (GND)	7	7	7	7	7	7
3	SHIELD	<-> (GND)	1	1	1	1	1	1

The shield can be either soldered to the connector shell or connected to pin 1 on the H89 connector.

The X10 interface cable which comes with the IBM-PC version of the CP-290 will directly connect to the Heath H89 Modem Port 330. The other H89 serial ports can be used but the gender of the connector needs to be changed from female to male and pins 2 and 3 need to be swapped.

## X10.COM Command Syntax Summary

(In all cases, only the first letter of the first argument need be entered)

---

NOW - Direct Command: Perform an X10 Direct Command

```
X10 NOW huu (uu uu ..) fcn lvl
      |      |      |      |
      |      |      |      +--- Dim Level - 1 (dim) to 16 (bright)
      |      |      +----- Function - ON, OFF or DIM
      |      +----- (more unit codes 1 -> 16)
      +----- House/Unit Code - A ->P, 1 -> 16 (or ALL)
```

---

TIME - Display the X10 internal Time, Day and Base Housecode

X10 TIME

---

SET - Set the X10 internal Time and Day

```
X10 SET hh:mm day (or X10 SET SYSTEM to use Datestamper or ZSDOS/ZDDOS time)
      |      |      |
      |      +----- Day of week (mon, tue, wed etc)
      +----- Time - hours:minutes (24 hour time)
```

---

READ/PRINT - Display or Print the X10 programmed events

X10 READ or X10 PRINT

---

EVENT - Program an X10 event

```
X10 EVENT eee huu (uu uu ..) hh:mm mtwtfss mod fcn lvl (N)
      |      |      |      |      |      |      |      |      |
      |      |      |      |      |      |      |      |      + - N = Don't Display
      |      |      |      |      |      |      |      |      +----- after loading
      |      |      |      |      |      |      |      +----- Dim Level -
      |      |      |      |      |      |      |      |      1 (dim) to
      |      |      |      |      |      |      |      |      16 (bright)
      |      |      |      |      |      |      |      +--- Function - ON, OFF or DIM
      |      |      |      |      |      |      +----- Mode - NORmal, SECurity
      |      |      |      |      |      |      |      |      TODAY or TOMorrow
      |      |      |      |      |      |      |      +----- Days - either the first letter or "--"
      |      |      |      |      |      |      |      |      (don't care if mode isTODor TOM)
      |      |      |      |      |      |      |      +----- Time - hours:minutes (24 hour time)
      |      |      |      |      |      |      |      |      (More Unit Codes)
      |      |      |      |      |      |      |      +----- House/Unit Code - A ->P, 1 -> 16 or ALL
      |      |      |      |      |      |      |      |      +----- Event Code - 1 -> 128
      +-----
```

---

CLEAR - Clear X10 programmed events

```
X10 CLEAR sss eee (N)
      |      |      |
      |      |      +----- No prompt - (optional) - don't ask "Are you sure?"
      |      +----- ending event number (1 -> 128)
      +----- starting event number (1 -> 128)
```

---

---

DOWNLOAD/UPLOAD - Upload events from a disk file or Download to a disk file

X10 DOWNLOAD (du:)filename(.XUL) (N) du: = optional Drive/User  
X10 UPLOAD (du:)filename(.XUL) XUL = Default file type  
N = Don't ask to overwrite existing file

---

DISPLAY - Monitor Activity

X10 DISPLAY

---

**Cross-Reference of X10 Products**

<u>Product</u>	<u>X10</u>	<u>HeathKit</u>	<u>Radio Shack</u>	<u>Crutchfield</u>	<u>DAK</u>
Lamp Module	LM465	BC-465	61-2683	009LM465	9779
Appliance 2 prong	LM486	AM-486	61-2681	009AM468	
Appliance 3 prong	LM466	BC-466	61-2684	009AM466	
Wall Switch	WS467	BC-467	61-2683	009WS467	9780
Wall Switch 3-way	WS4777	BC-4777	61-2686	009WS4777	
Wall Receptacle	SR227	BC-227	61-2685	009SR227	
220V 15A Appliance	HD243	HD-243			
220V 20A Appliance	HD245	BC-245			
Thermostat Set-Back	TH2807	BC-2807			
Mini Controller	MC460	MC-460	61-2677A		
Maxi Controller	SC503	BC-503	61-2690	009SC503	4622
Radio Controller	RC5000	BC-5000	61-2675	009RC5000	4712
Telephone Controller	TR2700	BC-2700			
Universal/Alarm	BA284	BC-284	61-2687		
(Interface external system inputs such as alarm systems)					
Universal Module		UM-506			5690
(Interface external system outputs such as sprinklers or drapery openers)					
Whole House Control		URC-5000			
(Infrared hand-held VCR-style remote control for X10 and TV/VCR/Stereo, etc)					
Command Center		URC-3000			
(The receiver portion needed with the hand-held URC-5000)					
Sensor Chime		SL-5321			
Timer	MT522	MT-522	61-2679	009MT522	4973
Timer/Radio	CR512				
Computer Interface	CP290P	BC-290P	905-2087	009CP290P	

The Computer interface CP290P comes with a serial cable which directly connects to the H89 modem port 330. The other H89 serial ports can be used but the gender of the connector needs to be changed from female to male and pins 2 and 3 need to be swapped.

You can order any product directly from X10 (USA) Inc. They accept major credit cards. You can also stop into your local Radio Shack store for most of the common products.

X10 (USA) INC.  
185A LeGrand Ave.  
Northvale, NJ 07647  
(201) 784-9700  
(800) 526-0027

DAK Industries  
8200 Remmet Ave.  
Canoga Park, CA 91304  
(800) 325-0800 - order  
(800) 888-9818 - tech info

Crutchfield  
1 Crutchfield Park  
Charlottesville, VA 22906  
(800) 446-1640

Heathkit  
Benton Harbor, MI 49022  
(800) 253-0570

Home Control Concepts  
P.O. Box 27983  
San Diego, CA 92198  
(800) 828-8537 Order Hotline (cards OK)  
(619) 484-0933 Info and Customer Service

HCC is a high volume distributor carrying a wide variety of standard and EXOTIC X10 products. They do have a \$100 minimum order but prices are low!



Publications:

Electronic House  
747 Church Road, G-11  
Elmhurst, IL 60126-1420  
(219) 256-2060  
\$14.95 per year MC/VISA/AMEX

Circuit Cellar INK  
4 Park Street Suite 20  
Vernon, CT 06066  
(203) 875-2751  
\$17.95 per year MC/VISA

Other:

There is a special user area in GENie dealing with Home Automation. GENie is an on-line subscription computer service. In the Home Automation area in GENie, you will find a roundtable -- a two-way dialog of people interested in home automation as well as computer programs available for downloading.

To sign-up for GENie, call client services at 1-800-636-9636 or write

GE Information Services  
401 North Washington Street  
Rockville, Maryland 20850

## Sample X10 Upload Files

---

```
;House Lighting Schedule  
;Revised 12/23/89
```

```
;  
001 F01 16:15 MTWTFSS NOR ON ; Living Room ON  
002 F01 00:00 MTWTFSS NOR OFF ; Living Room OFF  
003 F03 16:20 MTWTFSS NOR ON ; Foyer ON  
004 F03 00:30 MTWTFSS NOR OFF ; Foyer OFF  
024 F04 17:00 MTWTFSS NOR ON ; Porch ON  
025 F04 23:00 MTWTFSS NOR OFF ; Porch OFF  
&26 128; clear events 26 through 128  
# ; End-of-File Marker
```

Example 1: Events 1 through 4, 24 and 25 are programmed with comments. These events are overlaid on top of any other events currently in memory. Then events 26 through 128 are cleared.

---

```
C ; Master clear all events  
;House Lighting Schedule  
;Revised 02/01/90
```

```
;  
001 F01 16:15 MTWTFSS NOR ON ; Living Room ON  
002 F01 00:00 MTWTFSS NOR OFF ; Living Room OFF  
003 F03 16:20 MTWTFSS NOR ON ; Foyer ON  
004 F03 00:30 MTWTFSS NOR OFF ; Foyer OFF  
024 F04 17:00 MTWTFSS NOR ON ; Porch ON  
025 F04 23:00 MTWTFSS NOR OFF ; Porch OFF  
# ; End-of-File Marker
```

Example 2: The character "C" as the very first character of the upload file causes a master clear of any currently programmed events. Then events 1 through 4, 24 and 25 are programmed with comments.

---

---

```

Clear
;This is the Late Spring upload file
;
;A1=Front Porch light
;A2=Living Room table lamp
;A3=Entry Way cabinet light
;A4=Lamp over stereo
;A7=Family Room Table Lamp
;A8=Dining Room cabinet
;A10=South wall outside light
;
;Downstairs light
001 A01 02 03 10 05:45 MTWTF-- NOR DIM 10;On early school days
002 A01 02 03 10 06:30 MTWTFSS NOR ON
003 A01 02 03 10 07:30 MTWTFSS NOR OFF
004 A07          08:30 MTWTF-- NOR OFF
005 A07          18:15 MTWTFSS NOR ON
;
006 A01 02 03 08 10      20:30 MTWTFSS NOR ON
007 A01 02 03 04 07 08 10 23:00 MTWT--S NOR OFF
008 A01 02 03 04 07 08 10 00:00 ----FS- NOR OFF;On later on weekends
;
;B1=Upper hallway lights
;B5=Gamerom lights
;B6=Gamerom sewing table lamp
;B9=Elaine's table lamp
;B10=Master bedroom closet
;Upstairs lights
030 B01          06:00 MTWTF-- NOR DIM 06;Getting up for school
031 B01          06:30 MTWTF-- NOR DIM 16
032 B01 05 06 09 10 08:30 MTWTFSS NOR OFF
034 B10          12:00 MTWTFSS NOR OFF
035 B01 05 06 10    18:30 MTWTFSS NOR ON
036 B01 05 06      23:00 MTWT--S NOR OFF
037 B01 05 06      00:01 ----FS- NOR OFF
038 B01 05 06      02:00 MTWTFSS NOR OFF;Did they turn them on again?
#

```

Example 3: Here is Biff Bueffel's "late spring" upload file. Notice the liberal use of comments as well as the efficient use of padding blank spaces to align the commands in a vertical format. The program correctly processes any blank space between command arguments.

---

## X10.COM Computer/CRT Hardware Interface Overlay

Here is the standard Heath H8/H89 overlay for X10.COM. It is patterned after common overlays found in the CP/M public domain world of modem programs. You can use this one to create a custom version for a different computer or CRT display. Older overlays used with X10 before Version 3 Release 07 are not compatible with the newer ones.

To use a new or modified overlay, you need to first assemble it into a HEX file. The syntax to do this using the SLR assembler is -

```
SLR180 XOH89/H
```

Next, you need to overlay the new HEX file into X10.COM with MLOAD -

```
MLOAD NEWX10.COM=X10.COM,XO-H89.HEX
```

This will create NEWX10.COM which will have the new overlay inserted.

Hardware overlays begin at 110H and may not grow beyond 2FFH.

'Fake' ZCPR3 ENV begins at 300H and ends at 37FH.

'Fake' ZCPR3 TCAP begins at 380H and ends at 3FFH.

You can create Hardware-only Overlays or ENV-only, TCAP-only or ENV+TCAP-only Overlays provided that they begin at the correct address and fall within the size restrictions.

```
=====
;
;   This is the Overlay for X10.COM
;   by Rick Swenton
;
;   This overlay is for the Heath 89
;
;   Version 3.0  02/26/90
;   Version 3.1  03/17/90  Added graphics to TCAP
;   Version 3.2  07/03/90  Modified TCAP to support new NZCOM
;                           TCAP and VLIB standards.
;
;
;   Overlay starts here at 110H
;
=====

org      110h

;       OVR1 through OVR9 must be in this location, starting at 110H
;       The code after OVR9 is free form but must not grow beyond 27FH.

ovr1:   defw      name           ; Overlay Name
```

```

ovr2:    jp      in$cpport      ; IN control port
ovr3:    jp      out$dport      ; OUT data port
ovr4:    jp      in$dport      ; IN data port
ovr5:    jp      ani$rcvb       ; RX ready bit
ovr6:    jp      cpi$rcvr      ; RX ready value
ovr7:    jp      ani$sndb      ; TX ready bit
ovr8:    jp      cpi$snldr     ; TX ready value
ovr9:    jp      initsio       ; Initialize UART

;
; Overlay name can occupy 50 chars here. Needs trailing 0
;
name:    defb    'Overlay for Heath 89 with 8250 I/O'
defb    'PORT>'
port:    defb    0D8H          ; H89 8250 Base Port

MODRCVB EQU      1            ; Bit to test for received data
MODRCVR EQU      1            ; Modem receive ready
MODSNDB EQU     20H          ; Bit to test for ready to send
MODSNDR EQU     20H          ; Modem send ready bit
BD600 EQU      00C0H        ; 8250 Baud Rate Divisor

IN$CPORT:
PUSH    BC
LD      A,(port)           ; Get the base port address
ADD     A,5                ; Offset to status port
LD      C,A               ; Save in C
IN      A,(C)              ; Read the modem control port
POP     BC
RET

OUT$DPORT:
PUSH    BC
PUSH    AF
LD      A,(port)           ; Get the base port address
LD      C,A               ; Save in C
POP     AF
OUT     (C),A              ; Send byte to modem data port
POP     BC
RET

IN$DPORT:
PUSH    BC
LD      A,(port)           ; Get the base port
LD      C,A               ; Save in C
IN      A,(C)              ; Read the modem data port
POP     BC
RET

ANI$RCVB:
AND     MODRCVB           ;
RET                                           ; Bit to test for receive ready

CPI$RCVR:

```

```

CP      MODRCVR      ;
RET                                ; Value of rcv. bit when ready

ANI$SNDB:
AND     MODSNDB     ;
RET                                ; Bit to test for send ready

CPI$SNDR:
CP      MODSNDR     ;
RET                                ; Value of send bit when ready

```

; The following is used to initialize the H89 8250 I/O port to 600 baud

```

INITSIO:
DI      ;
LD      A,(port)    ; Turn off interrupts for init.
INC     A           ; get the base port
LD      C,A         ; offset to interrupt enable register
XOR     A
OUT     (C),A       ; Interrupt enable register
INC     C
INC     C           ; Offset to Line Control Register
LD      A,80H       ; Insure out of mode to set baud rate
OUT     (C),A       ; Line control register
OUT     (C),A

LD      HL,BD600
DEC     C
DEC     C
DEC     C           ; back to base port
LD      A,L         ; Get least significant baud rate byte
OUT     (C),A       ; Register for 'LSP' speed if enabled

LD      A,H         ; Get most significant baud rate byte
INC     C           ;
OUT     (C),A       ; Register for 'MSP' speed if enabled

LD      A,03H       ; 8-level, 1 stop bit, no parity
INC     C
INC     C
OUT     (C),A       ; Line control register
INC     C
IN      A,(C)       ; Set 'DTR' for proper modem output level
OR      00000001B
OUT     (C),A       ; Modem control register
EI      ; Restore interrupts to normal
RET

```

```

if $ gt 2ffh
'Hardware overlay is greater than 2ffh'
endif

```

; Hardware overlay code must end before 02FFh

ORG 0300H ; Do not change

; A complete Z3ENV + TCAP is included here so that  
; we can use Z3LIB and VLIB routines when the program is running  
; under plain CP/M. If running under ZCPR3, all ENV and TCAP data  
; is derived from the real Z3ENV. When running under plain CP/M,  
; data is taken from the user provided Z3ENV and TCAP below.  
; Most of the entries are unused. A required entry is the  
; Processor Speed (for X10 and VLIB timing). The Quiet Flag  
; is respected under plain CP/M.

Z3ENV: ; fake z3env  
db 0C3H ; Jump instruction  
dw 0 ; CBIOS address  
db 'Z3ENV' ; Environment ID  
db 0 ; Environment class  
dw 0 ; external path address  
db 0 ; number of 2-byte elements in path  
dw 0 ; RCP address  
db 0 ; number of 128-byte blocks in RCP  
dw 0 ; IOP address  
db 0 ; number of 128-byte blocks in IOP  
dw 0 ; FCP address  
db 0 ; number of 128-byte blocks in FCP  
dw 0 ; NDR address  
db 0 ; number of 18-byte entries in NDR  
dw 0 ; ZCPR3 Command Line  
db 0 ; number of bytes in Command Line  
dw z3env ; ZCPR3 Environment Descriptor  
db 2 ; number of 128-byte blocks in Descriptor  
dw 0 ; Shell Stack address  
db 0 ; number of shsize-byte entires in Shell Stack  
db 0 ; size of a Shell Stack entry  
dw 0 ; ZCPR3 Message buffer  
dw 0 ; ZCPR3 External FCB  
dw 0 ; ZCPR3 External Stack  
db 0 ; quiet flag (1=quiet, 0=not quiet) <== X10  
dw 0 ; address of Wheel Byte  
db 4 ; Processor Speed in MHz <== X10  
db 0 ; maximum drive  
db 0 ; maximum user  
db 0 ; 1=OK to accept DU, 0=not OK  
db 0 ; CRT selection (MUST BE LEFT AS 0000!)  
db 0 ; Printer selection (MUST BE LEFT AS 0000!)  
db 80 ; width of CRT  
db 24 ; number of lines on CRT <== X10  
db 24 ; number of lines of text on CRT <== X10  
dw 0 ; drive vector (see p. 44 in manual)  
db 0 ; (fills in space from Z3 SYSENV)  
db 0 ; width of printer

```

db      0          ; number of lines on printer
db      0          ; number of lines of text on printer
db      0          ; form feed flag (0=can't formfeed, 1=can)
db      0          ; (fills in space from Z3 SYSENV)
db      0          ; (fills in space from Z3 SYSENV)
db      0          ; (fills in space from Z3 SYSENV)
db      0          ; (fills in space from Z3 SYSENV)
dw      0          ; address of CCP
db      0          ; CCP size
dw      0          ; address of DOS
db      0          ; DOS size
dw      0          ; address of BIOS (NZCOM's - not CBIOS)
db      'SH      '  ; shell variable filename
db      'VAR'    '  ; shell variable filetype
db      '      '  ; filename 1
db      '      '  ; filetype 1
db      '      '  ; filename 2
db      '      '  ; filetype 2
db      '      '  ; filename 3
db      '      '  ; filetype 3
db      '      '  ; filename 4
db      '      '  ; filetype 4
db      0          ; drive location of public files (p. 46)
db      0          ; user location of public files (p. 46)

; NZTCAP: H19.Z80
; Author: Joe Wright
; Date: 12 October 87
; Version: 1.0

; VERSION: 2.2 3 February 90
;
; Moved the CD (clear to end of screen) string back after LI string.
; Define B13 as offset from NZTCAP to GOELD.

; VERSION: 2.1 19 January 90

; Added Read Cursor and Read Line functions.
; Graphics characters now in a table instead of individual strings.

; VERSION: 2.0 12 December 89

; Modified for new VLIB4 graphics and for new SETATT and SETATR routines.
; The SA: macro string is modified for the routines and a new string,
; AT: is added to contain the four main attributes for this terminal:
; Normal, Blink, Reverse, Underscore

; Extended Termcap Data for NZ

; This is the proposed TermCap Data for the New Z-System.
; It more fully describes the terminal and its capabilities.

ESC     EQU      27          ; ASCII escape character

```



```

; I have adopted the convention that a terminal name is terminated
; with a space character, therefore no spaces within the name.
; Also that the terminal name is unique in the first eight characters.

NZTCAP: DB      'Heath-H19      ' ; Name of terminal (13 bytes)

; The Graphics section is no longer fixed so we must provide an
; offset to it. One byte is sufficient for a two-record TCAP.

B13:      DB      GOELD-NZTCAP      ; Offset to GOELD

; Bit 7 of B14 indicates the new Extended TCAP. Bits 6-0 are undefined.

B14:      DB      10000000B      ; Extended TCAP

; It is often desirable to differentiate terminals with other than
; their commands. For example TeleVideo 955 has the same command
; set as Wyse 60 but takes less time to initialize (reset).

; 16 bits are now reserved for indicating terminal characteristics
; which cannot be known from the strings. I have defined five
; of these bits for my own purposes.

; B15 b0 Standout          0 = Half-Intensity, 1 = Reverse Video
; B15 b1 Power Up Delay    0 = None, 1 = 10-second delay
; B15 b2 No Wrap           0 = Line Wrap, 1 = No Wrap
; B15 b3 No Scroll         0 = Scroll, 1 = No Scroll
; B15 b4 ANSI              0 = ASCII, 1 = ANSI

B15:      DB      00000101B      ; Power Up Delay

;      Single character Arrow Keys or WordStar Diamond

DB      'E'-'@'      ; Cursor up
DB      'X'-'@'      ; Cursor down
DB      'D'-'@'      ; Cursor right
DB      'S'-'@'      ; Cursor left
DB      00           ; Cl delay
DB      00           ; Cm delay
DB      00           ; Ce delay

; Strings start here.

CL:      DB      ESC,'E',0      ; Cl string - Home Cursor and Clear Screen      0
DB      ESC,'Y'+ %+ ',0      ; Cm string - Cursor motion macro      1
DB      ESC,'K',0      ; Ce string - Erase from cursor to EOL      2
DB      ESC,'p',0      ; So string - Start standout      3
DB      ESC,'q',0      ; Se string - End standout      4
DB      0      ; Ti string - Terminal Initialization      5
DB      0      ; Te string - Terminal Deinitilaization      6

; Extensions to Standard Z3TCAP

LD:      DB      ESC,'M',0      ; Line Delete      7

```

```

LI:      DB      ESC,'L',0      ; Line Insert      8
CD:      DB      ESC,'J',0      ; Clear to end of screen  9

; Set Attribute strings once again included.

SA:      DB      0                ; Set Attributes      10
AT:      DB      0                ; Attributes          11

; These two allow reading the Terminal's screen.

RC:      DB      0                ; Read current cursor position 12
RL:      DB      0                ; Read line until cursor  13

; Graphics start here.  Wyse-60 goes graphic with bit 7 set.

GOELD:   DB      0                ; On/Off Delay        0

; Graphics strings offset from Delay value.

DB      ESC,'F',0      ; Graphics On          1
DB      ESC,'G',0      ; Graphics Off         2
DB      ESC,'x5',0     ; Cursor Off           3
DB      ESC,'y5',0     ; Cursor On            4

; Graphics Characters

DB      'f'            ; Upper Left Corner   5
DB      'c'            ; Upper Right Corner  6
DB      'e'            ; Lower Left Corner   7
DB      'd'            ; Lower Right Corner  8
DB      'a'            ; Horizontal Line     9
DB      ''            ; Vertical Line       10
DB      'i'            ; Full Block          11
DB      'w'            ; Hashed Block        12
DB      's'            ; Upper Intersect     13
DB      'u'            ; Lower Intersect     14
DB      'b'            ; Mid Intersect       15
DB      't'            ; Right Intersect     16
DB      'v'            ; Left Intersect      17

; Fill unused space with nulls

REPT    128-($-NZTCAP)
DB      0
ENDM

; End of NZTCAPD

if      $ gt 4ffh
'Overlay is larger than 500h bytes'
endif

;      the overlay must end before 500H
;=====

```

end

35General Purpose Overlay35

```

test equ 0
;=====
;
; This is the Overlay for X10.COM
;   by Rick Swenton and Biff Bueffel
;
; Version 1.0  03/03/90
;
; Version 3.2  07/03/90      Modified TCAP to support new NZCOM
;                             TCAP and VLIB standards
;
; This is a General Purpose overlay to be used to adapt your computer
; to use X10.COM. The overlay can be easily modified for various serial
; I/O types such as the UART, 8251 USART, 8250 ACE, 2661 EPCI, etc.
;
; You MUST supply a minimum of information for this overlay to work.
; That which must be supplied is shown with an asterisk border. You
; must give the values for PORT, MODRCVB, MODRCVR, MODSNDB, and MODSNDR.
; You must provide the routines IN$CPORT, OUT$DPORT and IN$DPORT.
;
; If you can not figure out how to code INITSIO, you could use your
; CP/M Configure program to set the modem port for 600 baud before
; running X10.COM. The default for this port is usually an 8 bit word
; with 1 stop bit and no parity (this is what the CP-290 requires).
;
; Unless you have a very unusual computer all the information you
; will need to make an overlay can be found in an overlay for a modem
; program such a MEX114, MDM7xx or IMP (in the latter MODRCVB=MDRCV and
; MODRCVR=MDRCV, etc).
;
;
; If you are NOT using ZCPR3 or NZCOM (you know if you are!), you will
; also need to supply some data in the ZCPR INFORMATION area. Read
; carefully the asterisk highlighted sections for information you
; must provide.
;
; Overlay starts here at 110H
;=====

org 110h

; OVR1 through OVR9 must be in this location, starting at 110H
; The code after OVR9 is free form but must not grow beyond 27FH.

ovr1:  defw  name           ; Overlay Name
ovr2:  jp   in$control     ; IN control port
ovr3:  jp   out$dport      ; OUT data port
ovr4:  jp   in$dport       ; IN data port
ovr5:  jp   ani$rcvb       ; RX ready bit
ovr6:  jp   cpi$rcvr      ; RX ready value
ovr7:  jp   ani$sndb       ; TX ready bit

```

```

ovr8:    jp      cpi$sndr      ; TX ready value
ovr9:    jp      initsio      ; Initialize UART

BD600    EQU     00C0H        ; 8250 Baud Rate Divisor

;
; Overlay name can occupy 50 chars here. Needs trailing 0
;
name:    defb    'Overlay for My Computer with my I/O',0
defb    'PORT>'

;*****
;*      IMPORTANT!! You MUST supply the following values for      *
;*      the modem port or other serial port you will use AND      *
;*      the test plus ready bits.                                  *
;*****

port:    defb    0000H        ; Base Modem Port

MODRCVB  EQU     0            ; Bit to test for received data
MODRCVR  EQU     0            ; Modem receive ready
MODSNDB  EQU     0H          ; Bit to test for ready to send
MODSNDR  EQU     0H          ; Modem send ready bit

;*****

;*****
;*      The following three routines MUST be supplied.          *
;*****
IN$CPRT:
    RET

OUT$DPRT:
    RET

IN$DPRT:
    RET

;*****

;      DO NOT CHANGE THE NEXT FOUR ROUTINES

ANI$RCVB:
    AND     MODRCVB          ;
    RET                                ; Bit to test for receive ready

CPI$RCVR:
    CP      MODRCVR          ;
    RET                                ; Value of rcv. bit when ready

ANI$SNDB:

```

```

AND     MODSNDB           ;
RET                                           ; Bit to test for send ready

```

```

CPI$SNDR:
CP      MODSNDR           ;
RET                                           ; Value of send bit when ready

```

```

;           ^                               ^
;           | DO NOT CHANGE THE FOUR ROUTINES ABOVE! |

```

```

; The following is used to initialize the I/O port to 600 baud

```

```

INITSIO:           ;
RET

```

```

IF      $ gt 2ffh
'ERROR -- Hardware overlay is TOO large! '
ENDIF

```

```

; ZCPR3 INFORMATION

```

```

ORG     300H

```

```

; At this time, a complete Z3ENV + TCAP is included here so that
; we can use Z3LIB and VLIB routines when the program is running
; under plain CP/M. If running under ZCPR3, all ENV and TCAP data
; is derived from the real Z3ENV. When running under plain CP/M,
; data is taken from the user provided Z3ENV and TCAP below.
; Most of the entries are unused. A required entry is the
; Processor Speed (for X10 and VLIB timing). The Quiet Flag
; is respected under plain CP/M.

```

```

Z3ENV:           ; fake z3env
db         0C3H           ; Jump instruction
dw         0              ; CBIOS address
db         'Z3ENV'       ; Environment ID
db         0              ; Environment class
dw         0              ; external path address
db         0              ; number of 2-byte elements in path
dw         0              ; RCP address
db         0              ; number of 128-byte blocks in RCP
dw         0              ; IOP address
db         0              ; number of 128-byte blocks in IOP
dw         0              ; FCP address
db         0              ; number of 128-byte blocks in FCP
dw         0              ; NDR address
db         0              ; number of 18-byte entries in NDR
dw         0              ; ZCPR3 Command Line
db         0              ; number of bytes in Command Line
dw         z3env          ; ZCPR3 Environment Descriptor
db         2              ; number of 128-byte blocks in Descriptor
dw         0              ; Shell Stack address

```

```

db      0      ; number of shsize-byte entires in Shell Stack
db      0      ; size of a Shell Stack entry
dw      0      ; ZCPR3 Message buffer
dw      0      ; ZCPR3 External FCB
dw      0      ; ZCPR3 External Stack
db      0      ; quiet flag (1=quiet, 0=not quiet) <== X10
dw      0      ; address of Wheel Byte
;*****
;*          You MUST supply your correct Processor Speed          *
;*****
db      4      ; Processor Speed in MHz          <== X10
db      0      ; maximum drive
db      0      ; maximum user
db      0      ; 1=OK to accept DU, 0=not OK
db      0      ; CRT selection (MUST BE LEFT AS 0000!)
db      0      ; Printer selection (MUST BE LEFT AS 0000!)
db      80     ; width of CRT          <== X10
db      24     ; number of lines on CRT      <== X10
db      24     ; number of lines of text on CRT <== X10
dw      0      ; drive vector (see p. 44 in manual)
db      0      ; (fills in space from Z3 SYSENV)
db      0      ; width of printer
db      0      ; number of lines on printer
db      0      ; number of lines of text on printer
db      0      ; form feed flag (0=can't formfeed, 1=can)
db      0      ; (fills in space from Z3 SYSENV)
db      0      ; (fills in space from Z3 SYSENV)
db      0      ; (fills in space from Z3 SYSENV)
db      0      ; (fills in space from Z3 SYSENV)
dw      0      ; address of CCP
db      0      ; CCP size
dw      0      ; address of DOS
db      0      ; DOS size
dw      0      ; address of BIOS (NZCOM's - not CBIOS)
db      'SH'   ; shell variable filename
db      'VAR'  ; shell variable filetype
db      ' '    ; filename 1
db      ' '    ; filetype 1
db      ' '    ; filename 2
db      ' '    ; filetype 2
db      ' '    ; filename 3
db      ' '    ; filetype 3
db      ' '    ; filename 4
db      ' '    ; filetype 4
db      0      ; drive location of public files (p. 46)
db      0      ; user location of public files (p. 46)
;*****
; The Z3TCAP contains the definition for many terminal functions such
; as clearing the screen and placing the cursor at the top left hand
; position on the CRT.
;
; You should supply the TCAP information here UNLESS you are using
; ZCPR3 or NZCOM.
;

```

```

; The necessary data may be contained in the documentation for your
; computer or it may be obtained from many of the TCAP.LBRs that may
; be found on computer bulletin boards which still have CPM files.
;*****
; NZTCAP: MYTERM.Z80
; Author: Joe Wright
; Date: 12 October 87
; Version: 1.0

; VERSION: 2.2 3 February 90

; Moved the CD (clear to end of screen) string back after LI string.
; Define B13 as offset from NZTCAP to GOELD.

; VERSION: 2.1 19 January 90

; Added Read Cursor and Read Line functions.
; Graphics characters now in a table instead of individual strings.

; VERSION: 2.0 12 December 89

; Modified for new VLIB4 graphics and for new SETATT and SETATR routines.
; The SA: macro string is modified for the routines and a new string,
; AT: is added to contain the four main attributes for this terminal:
; Normal, Blink, Reverse, Underscore

; Extended Termcap Data for NZ

; This is the proposed TermCap Data for the New Z-System.
; It more fully describes the terminal and its capabilities.

ESC EQU 27 ; ASCII escape character

; I have adopted the convention that a terminal name is terminated
; with a space character, therefore no spaces within the name.
; Also that the terminal name is unique in the first eight characters.

NZTCAP: DB 'Myterm' ; Name of terminal (13 bytes)

; The Graphics section is no longer fixed so we must provide an
; offset to it. One byte is sufficient for a two-record TCAP.

B13: DB GOELD-NZTCAP ; Offset to GOELD

; Bit 7 of B14 indicates the new Extended TCAP. Bits 6-0 are undefined.

B14: DB 10000000B ; Extended TCAP

; It is often desirable to differentiate terminals with other than
; their commands. For example TeleVideo 955 has the same command
; set as Wyse 60 but takes less time to initialize (reset).

; 16 bits are now reserved for indicating terminal characteristics
; which cannot be known from the strings. I have defined five

```



; of these bits for my own purposes.

; B15 b0 Standout 0 = Half-Intensity, 1 = Reverse Video  
; B15 b1 Power Up Delay 0 = None, 1 = 10-second delay  
; B15 b2 No Wrap 0 = Line Wrap, 1 = No Wrap  
; B15 b3 No Scroll 0 = Scroll, 1 = No Scroll  
; B15 b4 ANSI 0 = ASCII, 1 = ANSI

B15: DB 00000000B ; Power Up Delay

; Single character arrow keys or WordStar diamond

DB 'E'-'@' ; Cursor up  
DB 'X'-'@' ; Cursor down  
DB 'D'-'@' ; Cursor right  
DB 'S'-'@' ; Cursor left  
DB 00 ; Cl delay  
DB 00 ; Cm delay  
DB 00 ; Ce delay

; Strings start here.

```
*****  
;* Please enter your terminal value. Example: *  
;* CL: DB ESC,'E',0 *  
;* The ",0" is MANDATORY! *  
*****  
CL: DB 0 ; Home cursor and clear screen 0  
*****  
DB ESC,'=%+ %+ ',0 ; Cm string - Cursor motion macro 1  
DB 'X'-'@',0 ; Ce string - Erase from cursor to EOL 2  
*****  
;* Please enter your terminal values. Example: *  
;* SO: DB ESC,'p',0 *  
;* SE: DB ESC,'q',0 *  
*****  
SO: DB 0 ; Start standout mode 3  
SE: DB 0 ; End standout mode 4  
*****  
DB 0 ; Ti string - Terminal initialization 5  
DB 0 ; Te string - Terminal deinitialization 6  
*****
```

; Extensions to Standard Z3TCAP

LD: DB 0 ; Line Delete 7  
LI: DB 0 ; Line Insert 8  
CD: DB 0 ; Erase to end of screen 9

; Set Attribute strings once again included.

SA: DB 0 ; Set Attributes 10  
AT: DB 0 ; Attributes 11

; These two allow reading the Terminal's screen.

```

RC:      DB      0          ; Read current cursor position 12
RL:      DB      0          ; Read line until cursor      13

```

```

; Graphics start here.  Wyse-60 goes graphic with bit 7 set.

```

```

GOELD:   DB      0          ; On/Off Delay                                0

```

```

; Graphics strings offset from Delay value.

```

```

DB      0          ; Graphics On                                1
DB      0          ; Graphics Off                               2
DB      0          ; Cursor Off                               3
DB      0          ; Cursor On                                4

```

```

; Graphics Characters

```

```

DB      '+'        ; Upper Left Corner                        5
DB      '+'        ; Upper Right Corner                       6
DB      '+'        ; Lower Left Corner                        7
DB      '+'        ; Lower Right Corner                       8
DB      '-'        ; Horizontal Line                          9
DB      '|'        ; Vertical Line                               10
DB      '*'        ; Full Block                               11
DB      '#'        ; Hashed Block                               12
DB      '+'        ; Upper Intersect                               13
DB      '+'        ; Lower Intersect                               14
DB      '+'        ; Mid Intersect                               15
DB      '+'        ; Right Intersect                               16
DB      '+'        ; Left Intersect                               17

```

```

; Fill unused space with nulls

```

```

REPT    128-($-NZTCAP)
DB      0
ENDM

```

```

; End of NZTCAPD

```

```

if      $ gt 4ffh
'Overlay is larger than 500h bytes'
endif

```

```

; the overlay must end before 500H
=====

```

```

end

```