



The

MCU-500

SiteMan – MCU500 user management program

Introduction

Version 1.10

Release Date: 12/4/04

Doc: SM_UserMan.doc

Welcome to the user's introduction to the SiteMan program.

SiteMan was created as an interface for users to manage MCU's in a distributed network. It also includes display capabilities. The display capabilities are tailored for smaller systems, not quite on a scale as large as something like WonderWare (Tm), or the equivalent, but sufficient for most applications. And, it's a bit less pricey.

SiteMan works in conjunction with another background program called NetCom. It's NetCom's job to interface with the MCU's directly, and act as both a router, and as an authenticator for the Users of SiteMan. Only valid users are allowed access into an MCU network; both SiteMan and NetCom enforce a rigid access policy, based on user names, and passwords. Users are granted access by NetCom. NetCom, in turn, is authenticated by each MCU on the network. If NetCom has the proper access codes, it's allowed to have access to a particular MCU.

In finer detail, Users are granted access according to a priority level associated with each user. Depending on the level granted, you, as a logged in SiteMan user, may read channel data, change setpoints or alarm levels, turn on or off channels, adjust and reset system parameters, upload and activate databases, change switch states and digital out parameters, and so on.

SiteMan screens are in general self explanatory. About the only area that requires a bit of study is the SiteView setup. More on that later. Let's see what SiteMan looks like:



Figure 1.0 SiteMan Main entrance screen.

Pretty much self explanatory, right? OK, so perhaps not. But, in a few moments, you may begin to appreciate that setting up a control system, much less a control network, may just take a bit of planning.

Let's start with the basic interface: the configuration menu. This is the screen that will contain your logon information (sorry, you have to remember your own password.), the information identifying the NetCom address you want to log into, and so forth.

Configuration Screen	
Help	
Remote Host:	<input type="text" value="COS TEST SITE 1"/> <input type="button" value="addresses"/>
Login Name:	<input type="text" value="bill"/>
My Host ID #:	<input type="text" value="56"/>
My Alarm ID #:	<input type="text" value="57"/>
MCU ID #:	<input type="text" value="140"/>
Display Activity:	<input type="text" value="Off"/>
Clear Window:	<input type="text" value="On"/>
Command Retry Max:	<input type="text" value="3"/>
Command Timeout:	<input type="text" value="10"/>
Connection Timeout:	<input type="text" value="10"/>
Port Offset:	<input type="text" value="3400"/>
Alarm Notify:	<input type="text" value="On"/>
<input type="button" value="OK"/> <input type="button" value="CANCEL"/>	

Figure 1.2 – SiteMan configuration main screen

The “**Remote Host**” window is where you select the NetCom that you wish to connect to. Each NetCom is defined in the address setup screen. You have a choice . You can use the network name of the desired NetCom, if you have a name server running, or you may select it on the basis of a network address. The network name includes world wide web addresses, such as www.cos-X01.com (a test site of ours, which you will be able to access later). Yes, you can log into a Netcom from anywhere in the world!

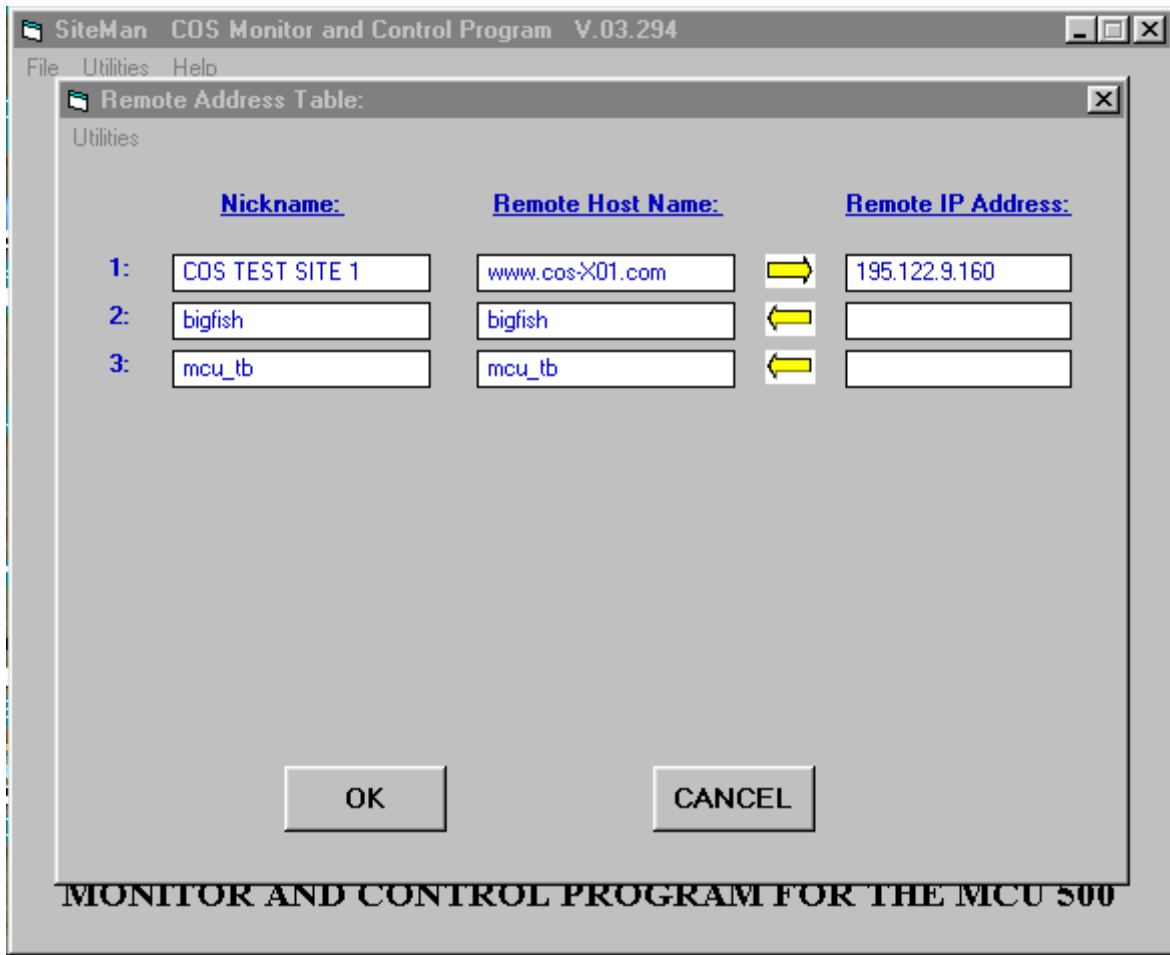


Figure 1.3 – Setting up address relationships.

You can set up to 10 such addresses. The yellow arrow selects between either the name field, or a pre-defined fixed address. Tap the arrow to change it's direction and selection. Assign a unique name for each address setup in the nickname field. This is the name which will appear on the main configuration "Remote Host" drop-down field.

The **Logon Name** is the name you must use to log into the NetCom. It must be pre-setup by the system administrator who has permission to setup such things in that NetCom. It's unique. As is the password that is associated with it, which you'll be asked to supply when you attempt to log in. No, SiteMan does not keep passwords.. That you'll have to remember..

My_Host_ID and **My_Alarm_ID** are addresses, unique to this SiteMan instance. (An "instance" is this particular copy of SiteMan running on a particular PC – Yes, there can be more than one copy of SiteMan running on a particular PC). This gets a bit involved – just suffice it to say that NetCom supports up to 199 MCU addresses on it's MCU side, and up to 199 addresses on it's SiteMan side. They don't overlap (i.e. address 150 on the MCU side is not the same as SiteMan address 150). More, when you take the course.

The **MCU address** is optional – it's a default that used to pre-configure such fields on other pages. You can change it at will.

Display Activity, when turned on, sends additional messages, mostly of a diagnostic nature, to the activity screen. Usually left off.

Clear Window. If you turn this on, all messages in the activity window are cleared when they roll off the screen. If off, all messages are saved in a buffer, and you can scroll through them. Again, a diagnostic tool.

Command Retry max. Set this to 3. If it can't get through the first time, this is the number of re-try's it will make.

Command Timeout and connection timeouts. Enter the time, in seconds, to wait before re-issuing the command (a command failure) or re connecting to NetCom. Although real command failures are very rare, it is possible that an MCU has been "locked" by another process (as in downloading a database to it); in which case, you might not get a response before the timeout takes place.

Port Offset is the offset that you will use to communicate with NetCom. This is going to change. It's actually a NetCom specific parameter, which needs to be set up as a unique offset for each address. Later.

Alarm Notify: If turned on, any alarm coming into NetCom will be forwarded to this SiteMan. Normally, on.

So Much for the Configuration screen. And back to the **main screen**



Figure 1.4 – Main setup screen, again

The **Window** button will take you to a special screen where you'll be able to see the actual communication going on with the NetCom program. It's only used for debug purposes, which you may be asked to invoke if doing an on-line troubleshooting session with one of our technicians.

Help screen: self explanatory. But you may also get help on most topics by simply moving the mouse cursor over the title on an entry. If the question mark appears, click on the title for an in-depth discussion on same.

Order Wire: This is a special screen that will allow you to send messages to an MCU. But, you must know the specific address of the display, printer, or whatever it is you wish to send the message to. You'll learn more on this in the training class.

MCU Control: This button takes you to a screen that looks like this:

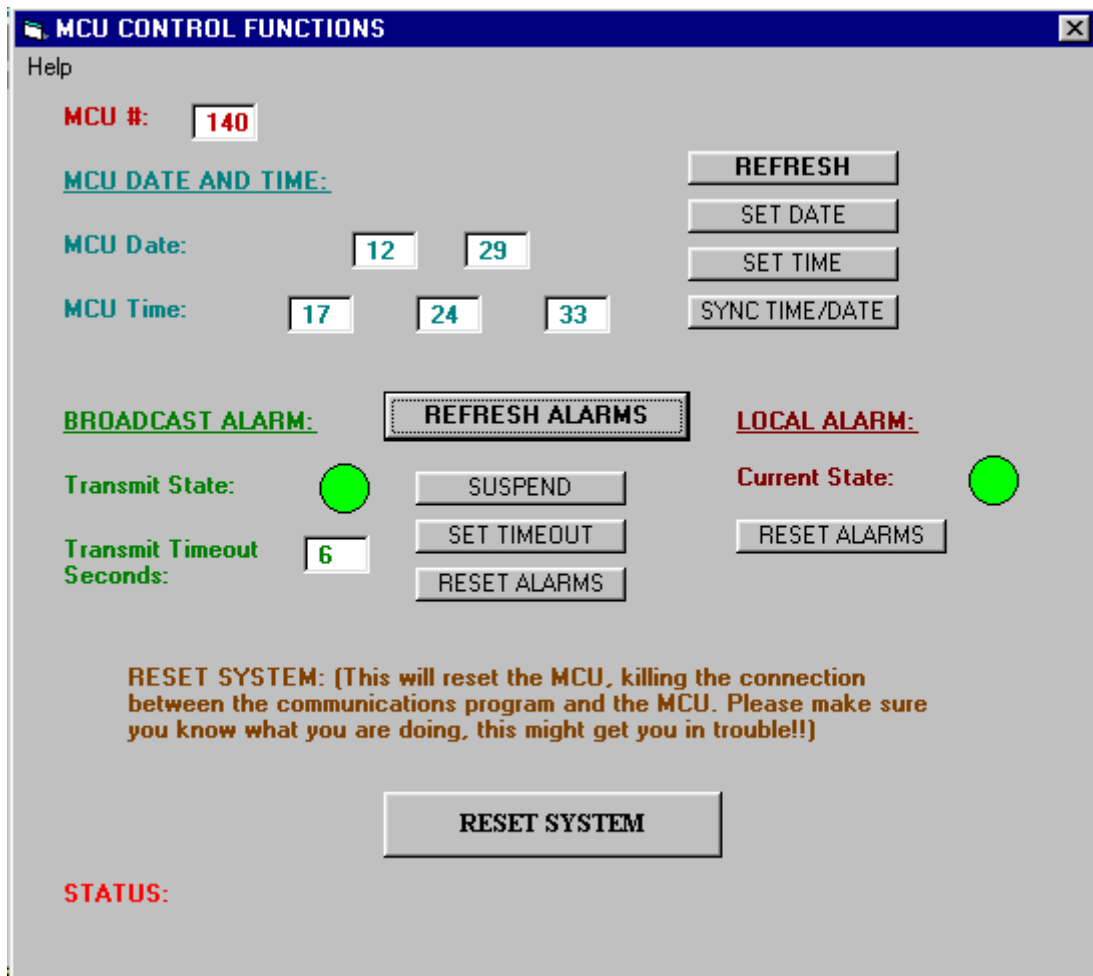


Figure 1.5 – MCU Control screen

The MCU Control screen gives you the tools to view and set the MCU's clocks, set the Broadcast alarm timeouts, see if any alarms are hanging, and view the local alarm status. The local status shows you if any alarms are being reported out through the MCU's local screen and if the local bell, buzzer, Klaxon or whatever is bothering the neighbors. And, you can clear the alarms. So, when the neighbors call at three AM, you can turn off the alarm bells at the site, remotely.

You can also force the MCU to reset. Don't do this unless you really know what you're doing.

COMM Opts refers to Communication Options. When you open this screen, you are presented with the current status of the various communication ports on the MCU (you specify the MCU's address). You can see how many messages are currently queued to be sent out, and you can turn on or off specific ports. You're not allowed to turn off the local printer port, if it's used.

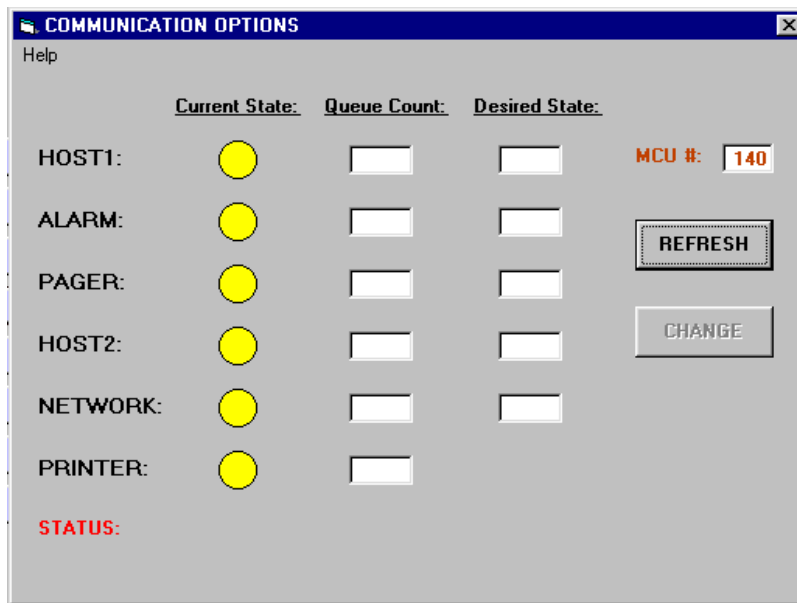


Figure 1.6 - Setting the MCU's communication options

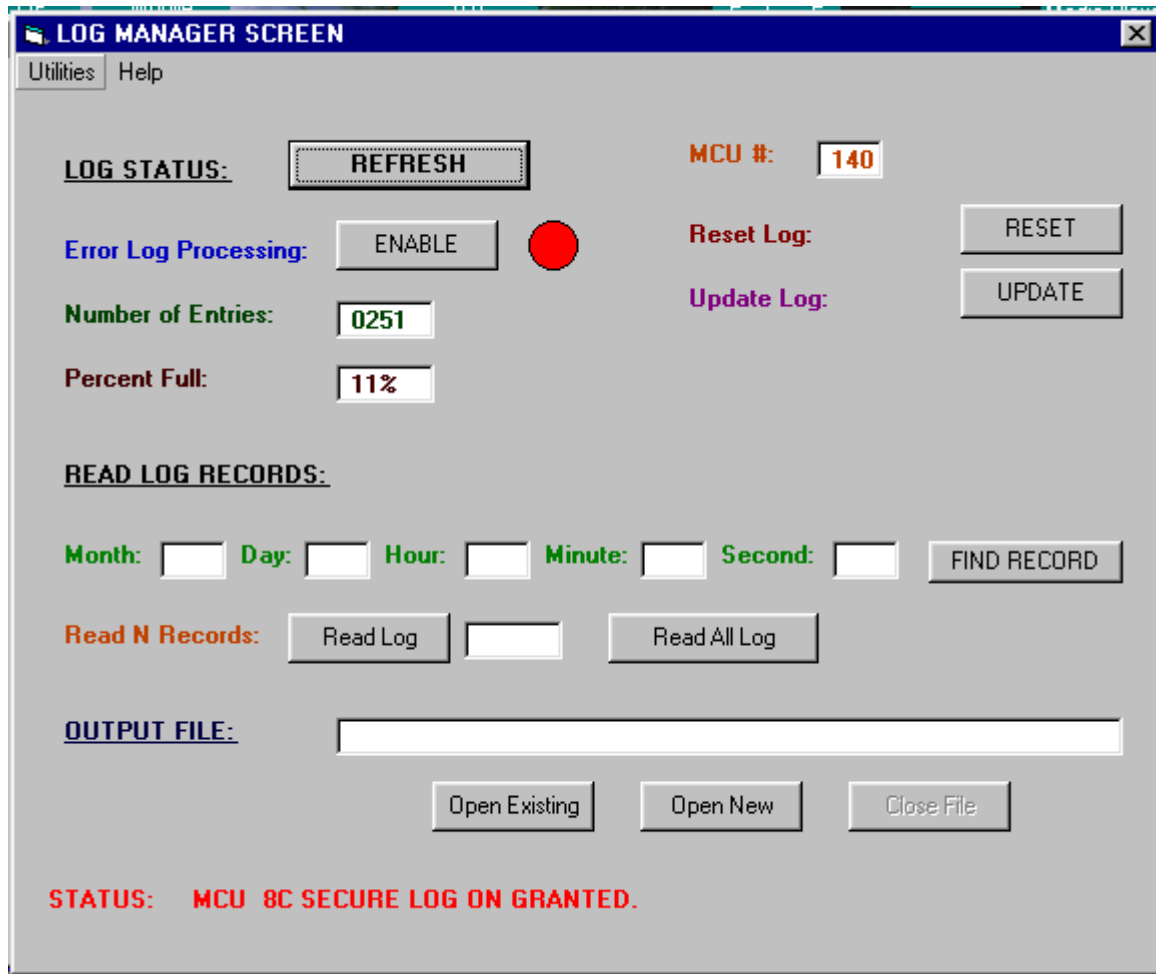


Figure 1.7 – The Log Management Screen

The MCU Log Control screen lets you manage the MCU's local logging function. All MCU's have the built-in capability to store data (you set up the format in the database). This is the primary logging function. On a periodic basis, NetCom will retrieve the data from the MCU, emptying the MCU's log file. Normally, MCU's can store up to about two to four days of data (assuming an average of 15 channels, sampled every 15 minutes) before the local memory is filled up. NetCom is normally configured to dump the logs long before this happens. NetCom maintains daily records on every MCU in its database, including daily log files containing the complete record of all log data retrieved from each MCU.

You can also search the MCU's log for records recorded after a specific time, and retrieve a portion of the log, if you wish. Very handy for checking an MCU for alarms, or other events.

The SWITCHES screen lets you view the current status of all switch outputs on a particular MCU. You can also alter any switch, if you have the right access permission.

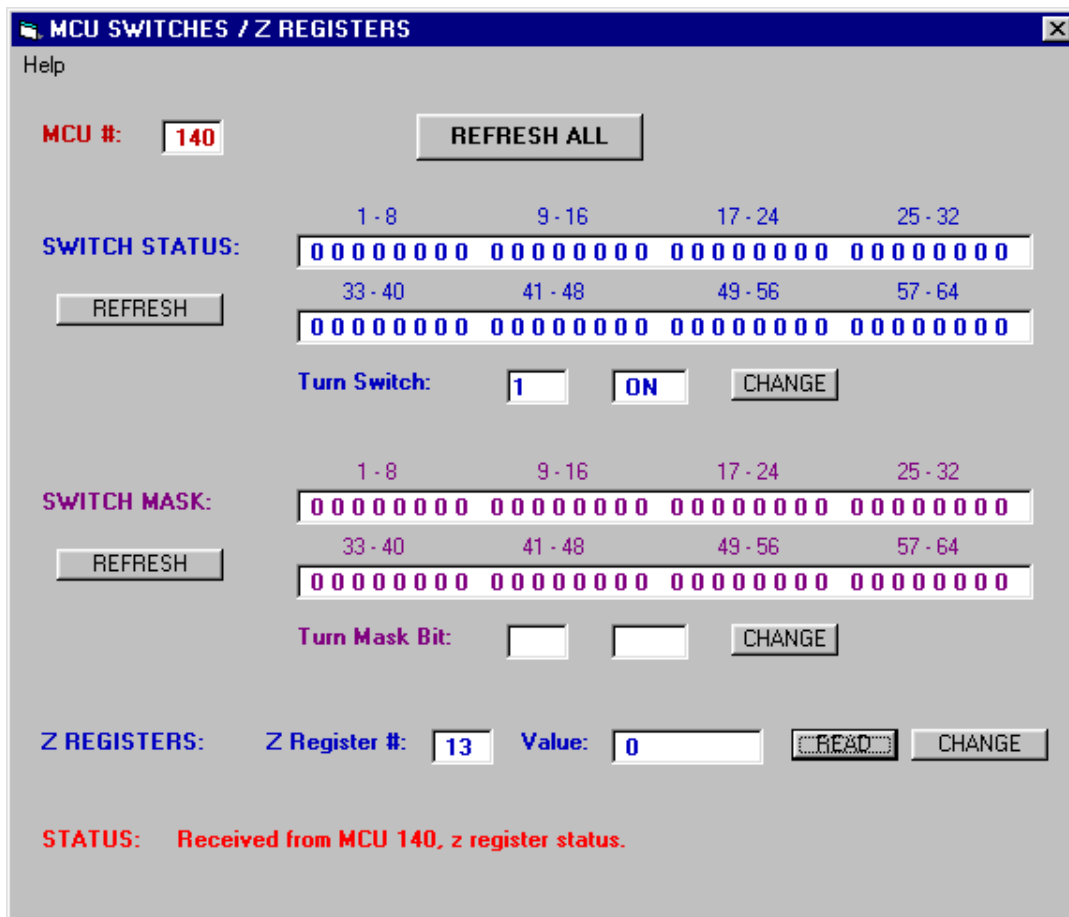


Figure 1.8 – The MCU SWITCH setup screen

You can also set the switch mask. If the mask is ON (set to a 1), then if there's any difference detected between the switch setting and the status reported back on that switch, an alarm will be generated.

Each MCU also has 64 Z registers, which you can do anything with via the database commands. For instance, a Z register, say reg 10, might hold a calibration linearization scale factor, or it might be the data that's input to a particular script function. You can read and/or alter that value remotely through this screen. It's a very simple, but powerful capability.

The Digital Screen lets you read and change digital inputs and outputs, much like the Switch's screen. Here you can read all the input bit status, and the transition registers. Transition registers are implemented on the first 16 digital bits (bits 1 through 16). Normally, the digital bits are sampled by the MCU's acquisition program on a timed interval. Although very quick, it is possible that the MCU would miss a fast responding transition on a bit. The Transition feature is implemented in hardware, and never misses. This is very useful for digital inputs that come and go very quickly. Transition bits are treated just like other digital input bits, and may be used by any channel in the data base.

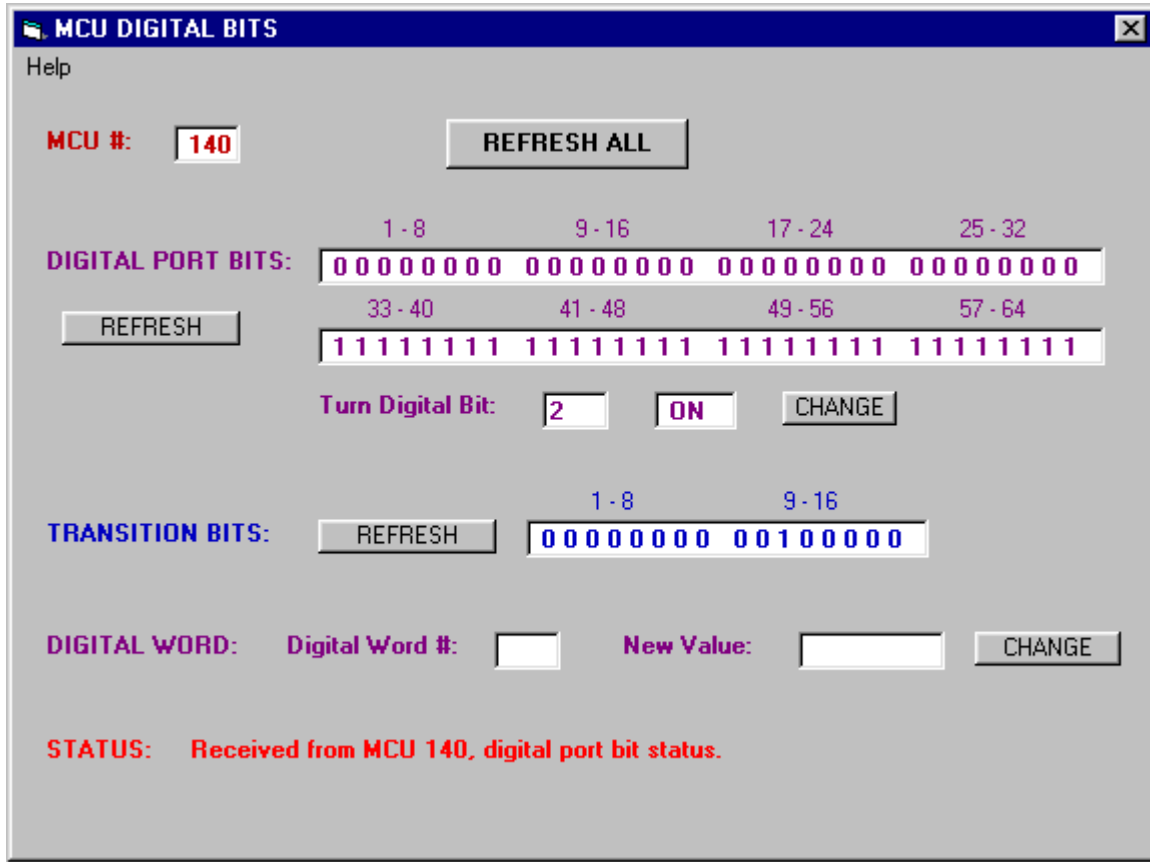


Figure 1.9 – The DIGITAL bit screen

You can also read and change digital words. Words are made up out of concatenated strings of digital bits. Words are defined in the database by you. You specify the start bit position, and the word width. Each MCU supports up to 64 digital input bits. Let's say you wanted to instrument a tank that had three float levels on it, one for empty, one for low (to turn on a fill pump), and one for full. To instrument this, you need 3 bits, one for each switch. So, pick three digital bits. You can start the word at any point in the sequence (i.e. bit 4), and define how many bits make up the word (i.e., 3 bits). Then, bit 4 would go to the empty switch, bit 5 to the low switch, and bit 6 to the full switch. Now, make up a calibration table for the 7 states. I'll let you think about this for now, but I will tell you the answer's in the database programmers manual.

Again, this function is a very simple and straight forward one, but has some very powerful capabilities.

Siteman also gives you the capability to read and change channel information, and to read and modify the database in a particular MCU. For instance, the CHANNELS button takes you to a screen that looks like this:

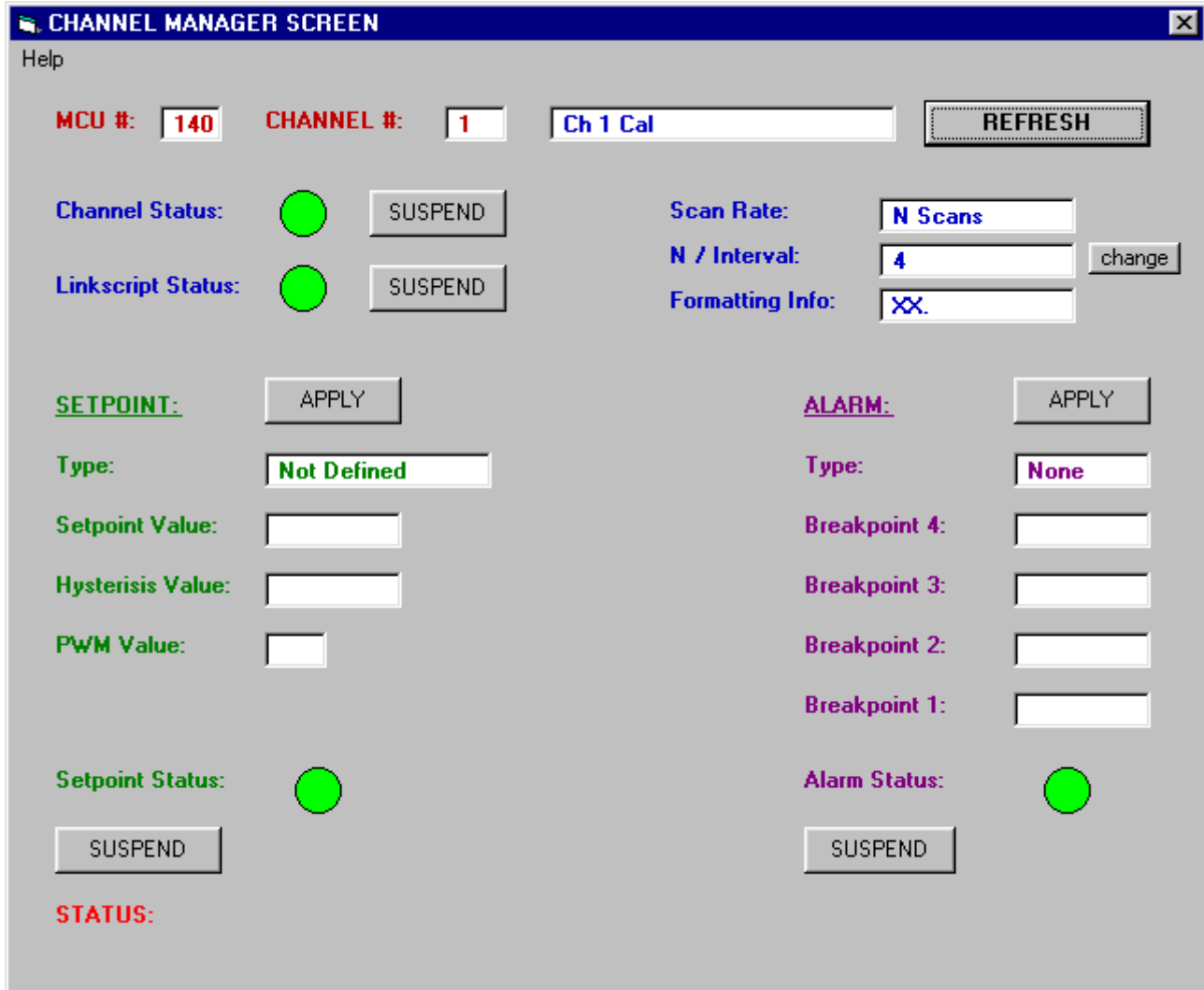


Figure 1.10 – The CHANNELS maintenance screen

By entering an MCU’s address and a channel number (there’s up to 61 user defined channels in each MCU – part of the database setup), and pushing the refresh button, you can get an up to the second display of the setup for that channel. You can modify the scan type, scan rate, and formatting information (how the data’s displayed). You can turn the basic channel on and off, turn on or off the link script (see the database manual for this function. Taken out of context, it can get very complicated real quick...). You can also change setpoint values, and alarm values. Total management of the channel.. well, almost. For that, you’ll have to read the database manual.

Databases are downloaded into MCU’s using the SiteMan program (and NetCom). That’s the only way to do it. But it’s easy. Let’s take a look at the database management screen:

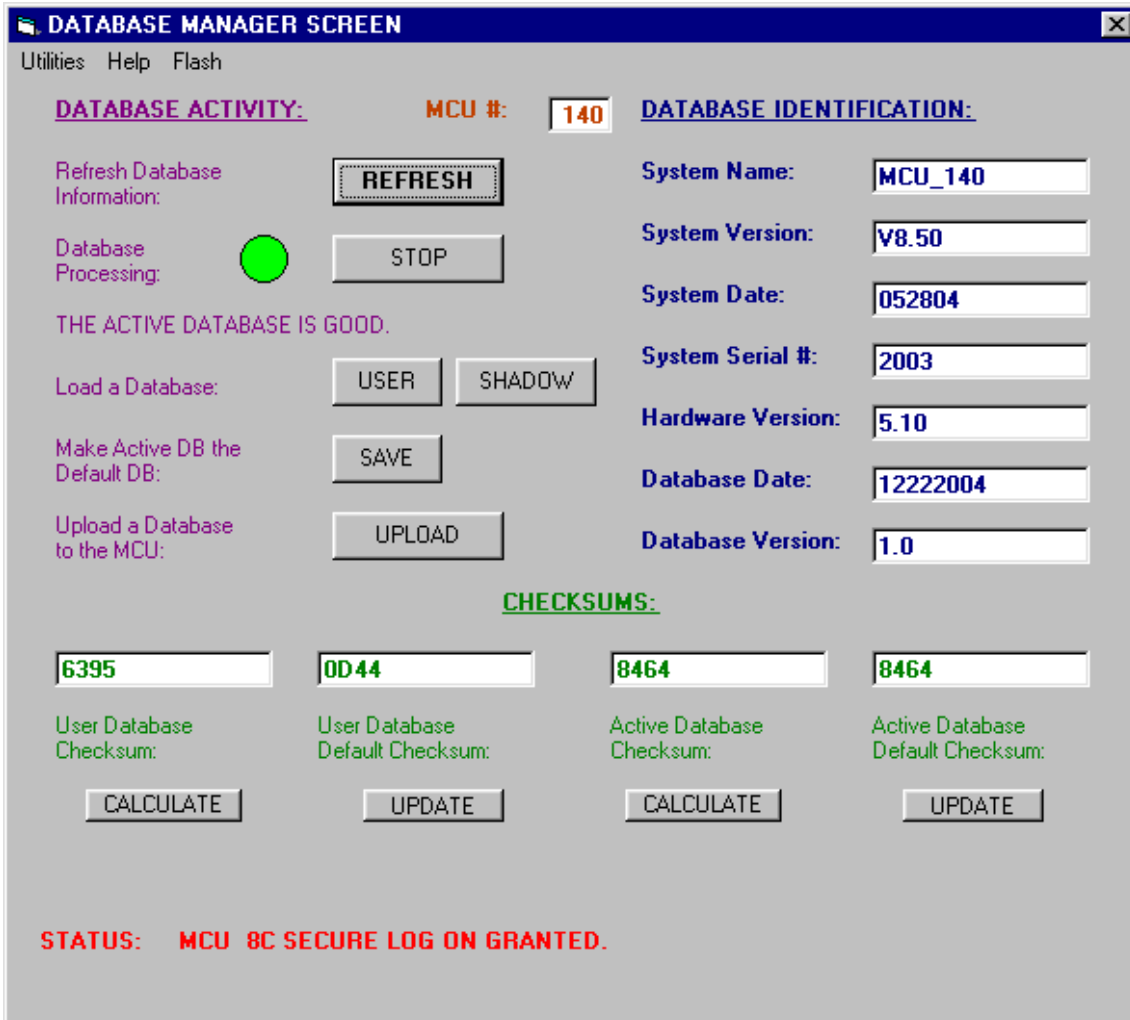


Figure 1.11 – the Database management screen

Again, to use it, simply enter the MCU's address and push the refresh button. All the configuration data on that MCU's database will be displayed. To load a new database, simply push the upload button. This will take you to a file management menu where you select the directory and file you want to load. It has to be a valid database file, created by the database manager, or DbMan. After you've set up the database, using DbMan, you compile it (using the built-in DbMan compiler function). If there's no errors, you have a "valid" database. That is, it will load correctly; but.. will it work? That all depends on your skills in setting it up..

After you've loaded the database to the MCU (you'll get a window showing you the load progress, and when it's complete), you should do a checksum calculation to insure there were no loader errors. This is very unlikely, but could happen. If the calculated checksums match the ones stored as part of the database, you've got a good database. To use it, push the USER button on the "load a database" line. That's it. Well, almost. Yes, you have a valid database loaded, and it will run.. But, if anything happens (power loss, catastrophic failures, etc.,) it's possible the RAM memory where it's stored will get corrupted. So, if you want to REALLY save it, push the save button. This saves it out to permanent memory. And yes, it DOES erase the previous version of the database. But, a word of caution...

If this IS a brand new database, and you're not too sure about it, simple push the run button, and let 'er rip! Don't use the SAVE button.. not yet! It is possible to put the MCU into some unknown state, and it might even be possible to lock it up where only a reset will bring it back. The MCU compiler attempts to catch most errors, such as recursive loops and so on, but it also allows you, the user, a great deal of latitude in what you may do. Hence, it's power. So, if it does lock up, or the database causes unwanted actions, what do you do?

Simple. Just hit the SHADOW database load button. This will re-load the prior database. Now, a word of caution. Don't do this if you can't get to the MCU, physically. In other words, don't use the remote loading function unless you've got a proven database! Since the database contains ALL the setup information for the MCU, INCLUDING it's comm port settings, it's just possible you can completely loose control of the MCU by re-loading the database with the wrong comm settings (normally, for the comm settings to take effect, you have to reset the MCU – but, if something else caused the MCU to reset, and the Comm settings were wrong...)

And now we come to the meat of the SiteMan program. So far, we've talked about it's management capabilities. But what about it's display capabilities?

One of the nicer feature'tets that SiteMan has is it's simple display interface, called MONITOR. It's really nice to be able to take a laptop out to the field where the sensors are located and be able to SEE the output of the sensor, in real time, as you're calibrating it! Here's a screenshot of an example:

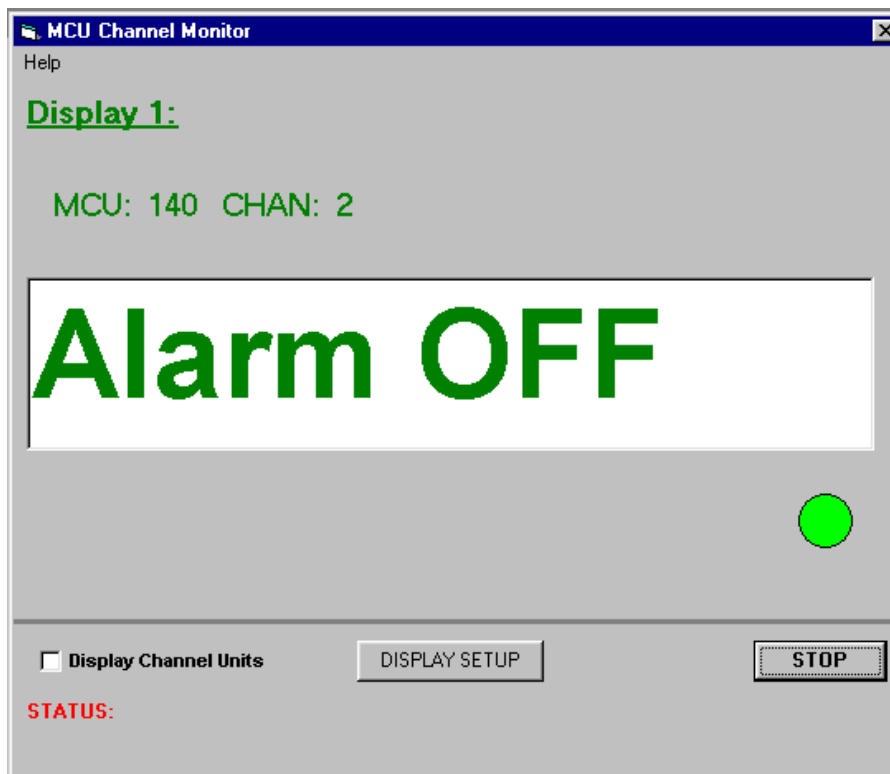


Figure 1.12 – the SiteMan MONITOR screen

In this case, were looking at a screen showing the alarm state of a channel. It could just as easily be the pH reading of a probe, or a temperature reading, etc. You pick the MCU and channel to display, and the refresh rate. How does it work? Simple. The laptop is connected to the local LAN via a standard Ethernet wireless link, which connects it to the PC where NetCom is running, which is connected to the particular MCU (by Ethernet or other means). The MCU is monitoring the sensor. SiteMan makes the request for the data, and the MCU answers. And the knee bone is connected ..

This is a very nice tool for doing routine maintenance. It can display up to 4 channels of information, like so:

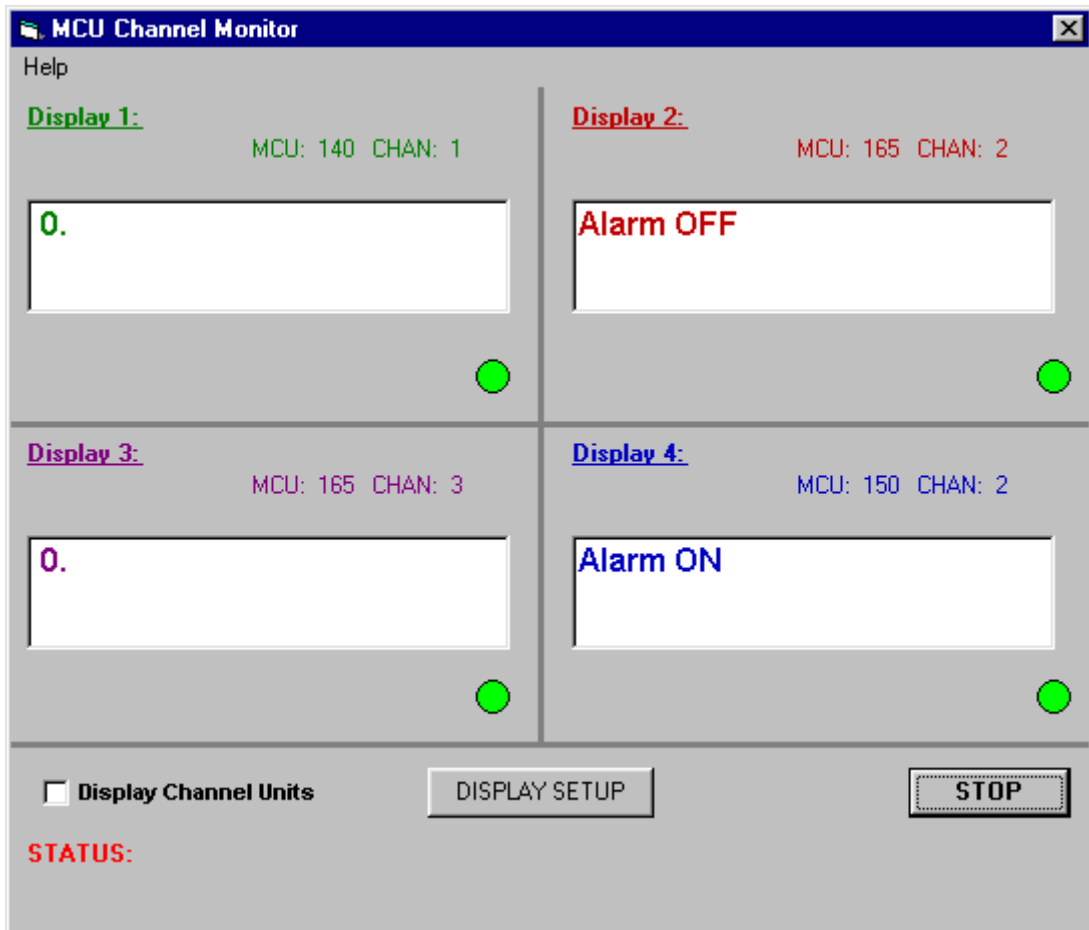


Figure 1.13 – MONITOR with 4 channels displayed

An excellent tool for quick monitoring or status checking on an MCU/channel. But not for trend logging, etc.

So, how about this?

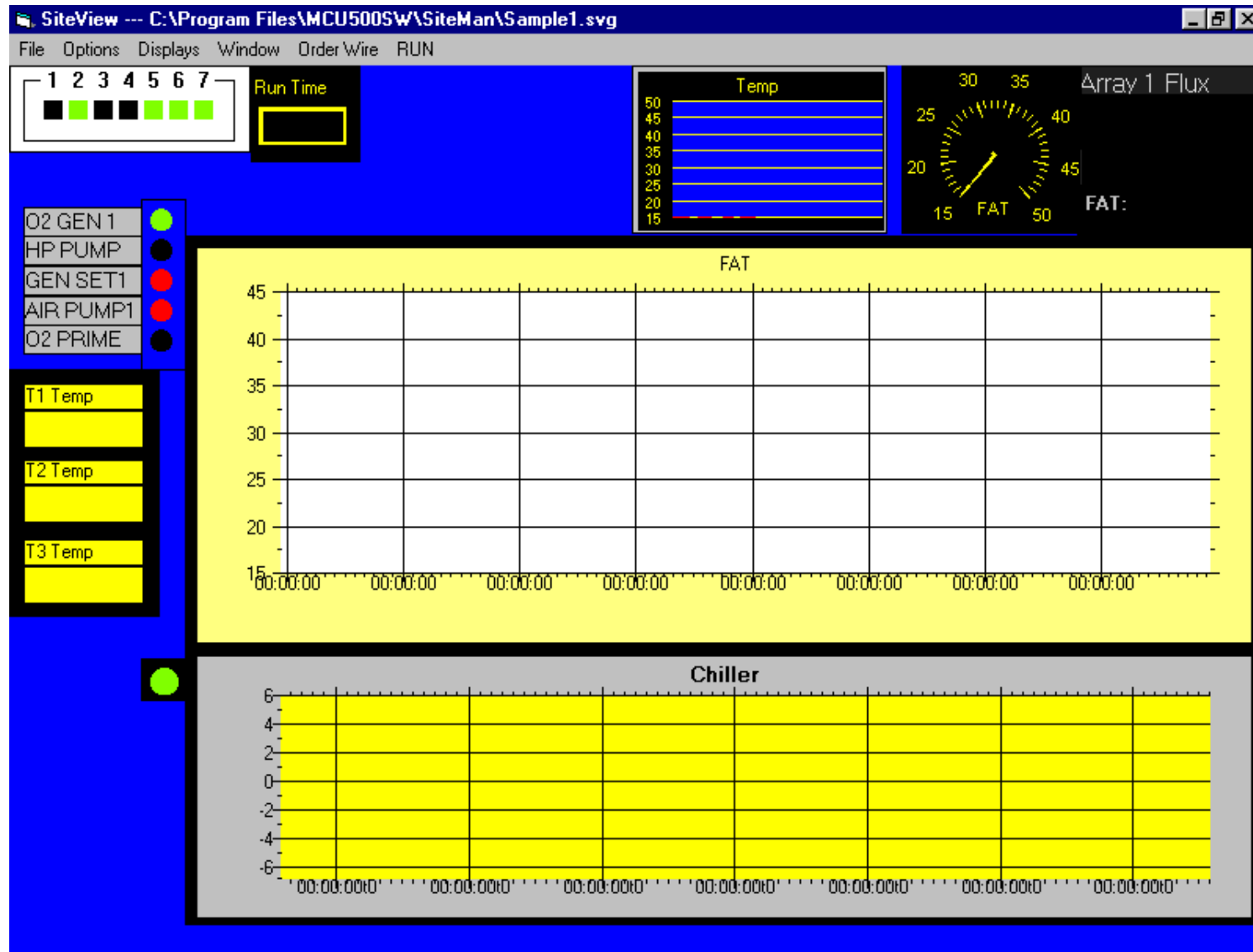


Figure 2.1 – SiteView, A sample of the bar graphs, charts, bit displays, instrument displays and other functions

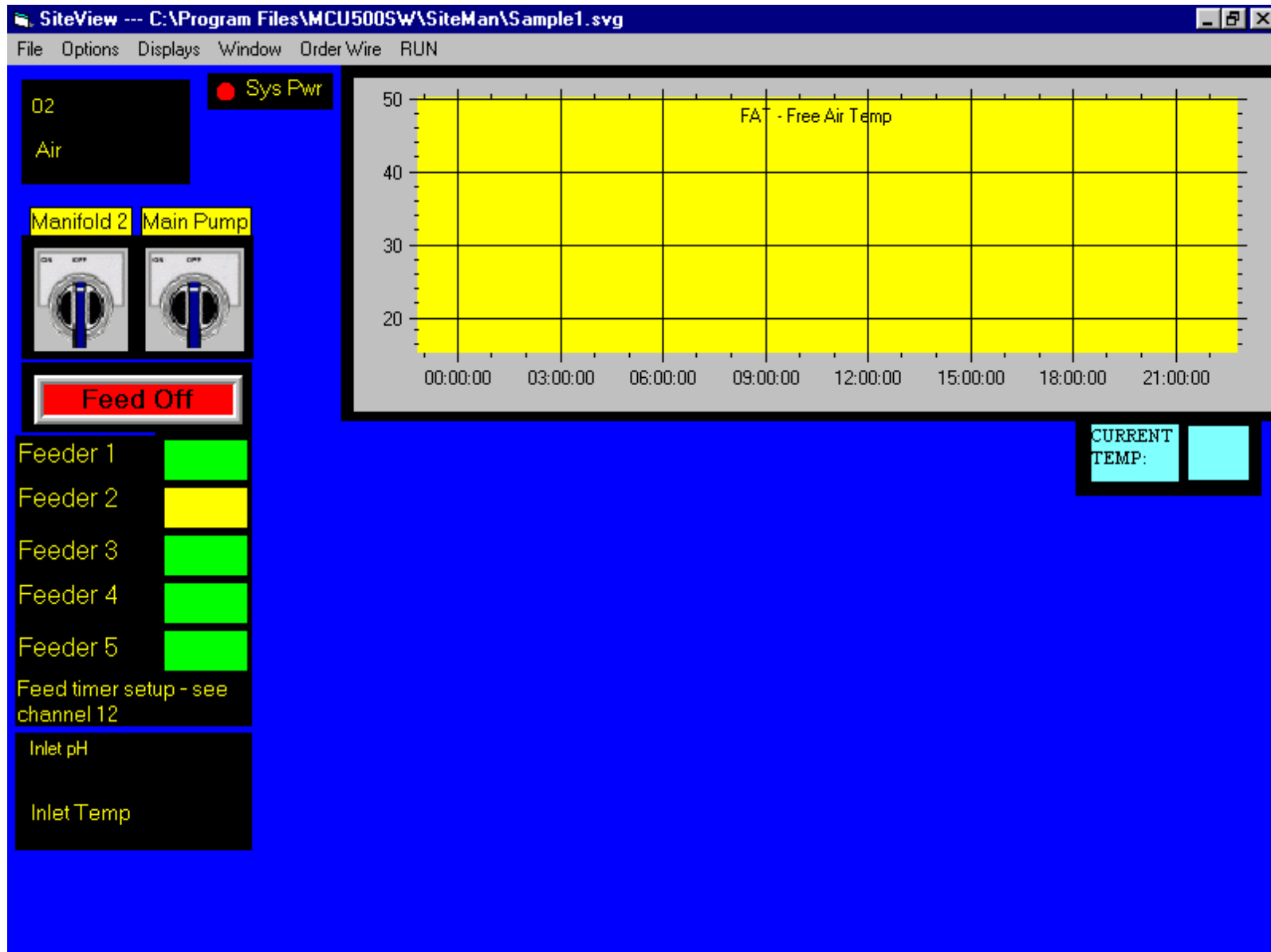


Figure 2.2 - SiteView, an example of interactive switches and controls – several out of many possibilities

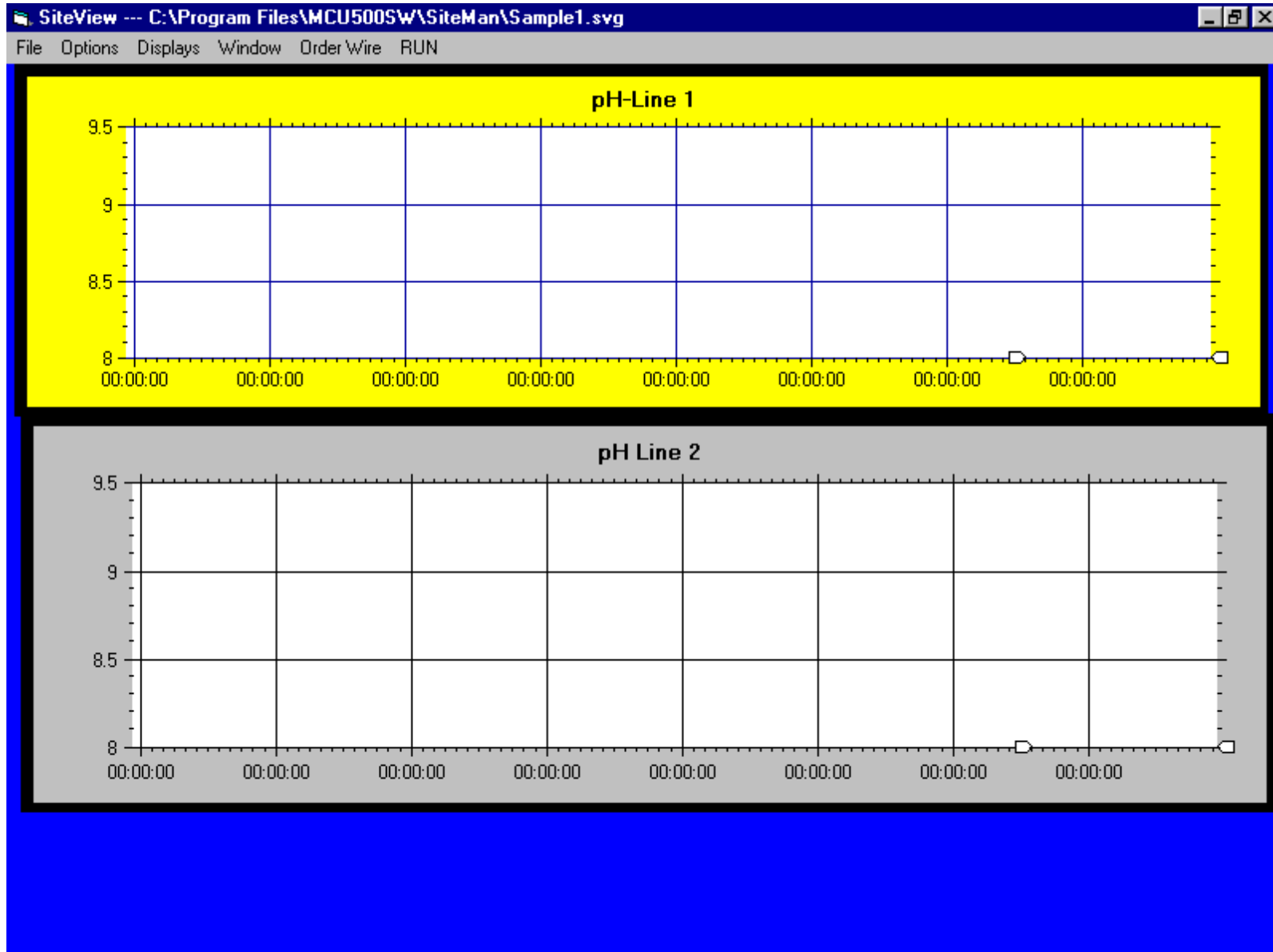


Figure 2.3 - SiteView an example of long term trend logging over many days..

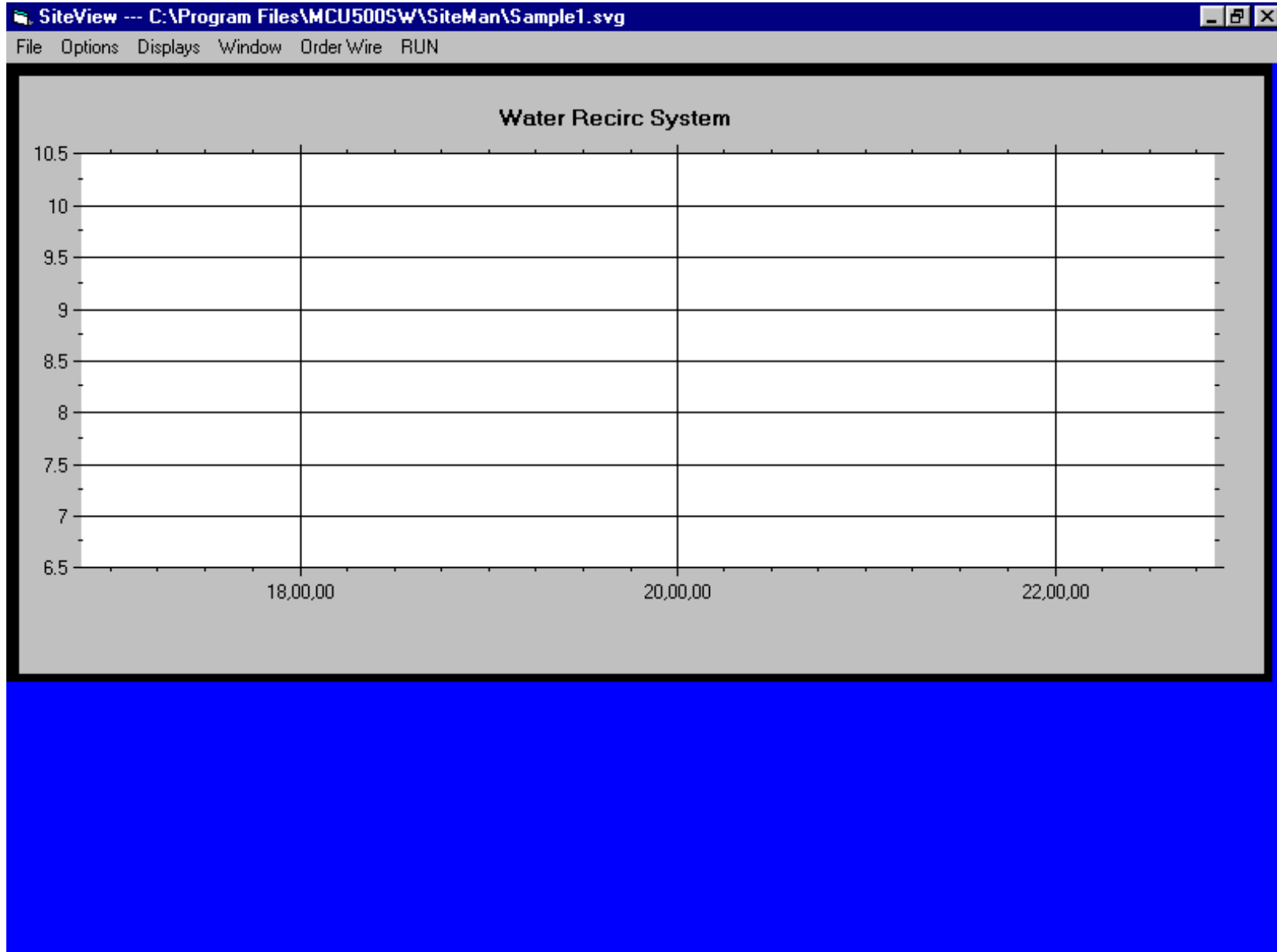


Figure 2.4 SiteView - BIG charts, easy to read even across the room.

SiteView is the main display function for SiteMan. SiteView lets you be truly creative. From simple charts, to interactive switches and controls, you set up your environment just the way you want it. The setup menus let you pick the MCU and Channel sources, digital bits, or words, switches, etc, and map them to SiteView channels. You specify the data acquisition rate. SiteView automatically scans these parameters, according to your setup and transfers the data to the respective SiteView channels, which you then use, and re-use as necessary to set up the displays and controls.

Need a strip chart? Simple. From the pull down DISPLAY menu, select strip chart. Place the control where you want it, and size it. Then open the control and set it up. You specify what SiteView channels get displayed, and in what order. Set up the log options, then open the chart setup control, and have a ball. Select display modes, colors, points and lines, etc. But a word of caution: you can easily get carried away. Strip charts are by far the most detailed and time consuming to set up, just because they tell so much. You'll want to get the format just right, the display sized just right, and so on. To help, turn on the grid locks, and set your grid scale. This lets you position controls and displays precisely where you want.

LED displays, bar charts, dial displays, switches, and other displays and controls are not as complicated as strip chart controls, so they're even easier to master. Soon you'll be setting up all kinds of displays.

Not only does the SiteView record the data it's currently displaying for each channel, but each display also has it's own data logging function. It records data by day, as well as into a master log, as you need. Need to take a look at what happened last year? No problem, just stop the display, save the current data, and re-load the data from last year. Or, create a second chart, side-by-side for comparison.

SiteView gives you the flexibility to monitor and control many parameters just as you need. Couple this with the capability to log into a remote NetCom from anywhere in the world, and you have a very powerful site management tool.

In some cases, it's not practical to run SiteMan/SiteView as a separate logging tool, simply because it's not on-line continuously. But the SiteMan at the main plant or facility usually is. It's on-line around the clock. So, we suggest an alternative: Use the excellent tool called GoToMyPC, to provide a secure, totally on-line connection to the PC running SiteMan. This, again, lets you log into that PC as though you were sitting at the keyboard of the machine. From anywhere., across the Internet. You can view the charts, and perform any of the functions allowed by SiteMan, just as if you were sitting there.

For more information, or to get a demo, give us a call.