SECTION 1. FUNCTIONAL MODES

1.1 DATALOGGER PROGRAMS - * 1, * 2, * 3, AND * 4 MODES

Data acquisition and processing functions are controlled by user-entered instructions contained in program tables. Programming can be separated into 2 tables, each having its own user-entered execution interval. A third table is available for programming subroutines which may be called by instructions in Tables 1 or 2 or by special interrupts. The * 1 and * 2 Modes are used to access Tables 1 and 2. The * 3 Mode is used to access Subroutine Table 3.

The * 4 Mode Table is a table of values used in the program that someone can change while the rest of the program is protected. These values may be used for sensor calibrations or to select optional sensors. The * 4 Table is only available when a special program created by EDLOG is loaded in the CR23X.

When a program table is first entered, the display shows the mode (table) number on the first line and 0000 on the second line. Keying an "A" will advance the editor to the scan interval. If there is an existing program in the table, keying an instruction location number prior to "A" will advance directly to the instruction (e.g., 5 will advance to the fifth instruction in the table).

1.1.1 SCAN (EXECUTION) INTERVAL

The scan interval is entered in units of seconds as follows:

1/100 .... 1 second, in multiples of 1/100 (.01)

1 .... 6553.5 seconds, in multiples of 1/10 (0.1)

Execution of the table is repeated at the rate determined by this entry. The table will not be executed if 0 is entered.

The sample rate for a CR23X measurement is the rate at which the measurement instruction can be executed (i.e., the measurement made, scaled with the instruction's multiplier and offset, and the result placed in Input Storage). Additional processing requires extra time. The throughput rate is the rate at which a measurement can be made and the resulting value stored in Final Storage. The maximum throughput rate for fast single-ended measurements, other than burst measurements, is 600 measurements per second (24 measurements repeated 25 times per second with the settling time set at 100 µs with Instruction P132).

If the specified execution interval for a table is less than the time required to process that table, the CR23X finishes processing the table and waits for the next occurrence of the execution interval before again initiating the table (i.e., when the execution interval has elapsed and the table is still executing, that execution is skipped). Since no advantage is gained in the rate of execution with this situation, it should be avoided by specifying an execution interval adequate for the table processing time.

> **NOTE:** Whenever the processing time of the user's program exceeds a table's execution interval, an error is logged in memory. The number of overrun errors can be displayed and reset in the B mode (Section 1.6) or using the Telecommunications A command (Section 5.1). An overrun will also cause "$^{T}{}_{o}$" to appear in the lower right corner of the display. "$^{T}{}_{o}$" will appear on the first table overruns and continue to be displayed until table overruns stop and * 0 or another * mode command is entered.

In some cases, the processing time may exceed the execution interval only when the Output Flag is set and extra time is consumed by final Output Processing. This may be acceptable. For example, suppose it is desired to sample some phenomena every 0.1 seconds and output processed data every 10 minutes. The processing time of the table which does this is less than 0.1 seconds except when output occurs (every 10 minutes). With final output the processing time is 1 second. With the execution interval set at 0.1 seconds, and a one second lag between samples once every 10 minutes, 9 measurements out of 5000 (.18%) are missed: an acceptable statistical error for most populations.

1.1.2 SUBROUTINES

Table 3 is used to enter subroutines which may be called with Program Control Instructions in Tables 1 and 2 or other subroutines. The group of instructions which form a subroutine must begin with Instruction 85, Label Subroutine, and end with Instruction 95, End (Section 12).

Subroutines 95, 96, 97, and 98 have the unique capability of being executed when a port goes high (ports 5, 6, 7, and 8 respectively). Any of these subroutines will interrupt Tables 1 and 2 (Section 1.1.3) when the appropriate port goes high. When the port goes high, the processor awakes within a few microseconds. The port triggers on the rising edge (i.e., when it goes from low to high). If the port stays high, the subroutine is not called again.

1.1.3 TABLE PRIORITY/INTERRUPTS

Table 1 execution has priority over Table 2. If Table 2 is being executed when it is time to execute Table 1, Table 2 will be interrupted. After Table 1 processing is completed, Table 2 processing resumes at the interruption point. If the execution interval of Table 2 coincides with Table 1, Table 1 is executed first, then Table 2.

Interrupts by Table 1 are not allowed in the middle of an instruction or while output to Final Storage is in process (flag 0 is set high). The interrupt occurs as soon as the instruction is completed or flag 0 is set low.

Special subroutines 95, 96, 97, and 98, initiated by a port going high (Section 1.1.2), can interrupt either Table 1 or 2 or can occur when neither is being executed. These subroutines can interrupt a table while the Output Flag is set. When the port goes high during the execution of a table, the instruction being executed is completed before the subroutine is run (i.e., as if the subroutine was called by the next instruction). For more information, refer to Section 12 (P85 Label Subroutine).

1.1.4 * 4 PARAMETER ENTRY TABLE

The CR23X * 4 mode is a table with up to one hundred values. Each value corresponds to an instruction parameter in the datalogger program. When the datalogger compiles the program, values in the * 4 table are transferred to the corresponding instruction parameter. The datalogger program must be created using EDLOG which allows instruction parameters to be assigned to the * 4 table.

In a network of datalogger stations, the * 4 table can be used to simplify site customization and the procedure of information entry. Once a generalized program is developed, application specific details, e.g., sensor calibration, can be entered without accessing the * 1 and * 2 program tables or the * 3 subroutine table.

ASSIGNING PARAMETERS TO * 4 - EDLOG

The only way to implement the * 4 mode is through EDLOG. The datalogger program is generated in EDLOG in the normal way.

To assign a parameter to a * 4 location, position the cursor on the desired parameter and press the "@" key. EDLOG then prompts for the location number in the * 4 table to be assigned to the associated parameter. After a valid number is entered, EDLOG marks the parameter with "@@nn" to the right of the parameter description, where "nn" is the * 4 location number.

Any program parameter or execution interval can be marked for inclusion in the table, as illustrated below.

PROGRAM

```
*       Table 1 Program
  01:        0.0        Execution Interval
                        (seconds) @@0

01:     Volts (SE) (P1)
        1:   1          Reps
        2:   1          ±2.5 mV Slow Range
        3:   1          SE Channel
        4:   1          Loc [ _________ ]
        5:   1          Mult @@1
        6:   0          Offset @@2
```

In the above example, * 4 location 0 is assigned to the program table execution interval, and locations 1 and 2 to the multiplier and offset of the measurement instruction. Note that a default execution interval of zero means the program will not execute until an alternative interval is entered in location 00 of the * 4 mode. A default multiplier and offset of 1 and 0 means that the measurement value is in units of millivolts. A different multiplier and offset can be entered in * 4 locations 1 and 2, respectively.

A * 4 location can be used in only one program parameter. For example, * 4 locations 0, 1, and 2 used in the example cannot be reused in another instruction in the same program.

If the * 4 feature is enabled in EDLOG when printing a program to a printer or disk file, the * 4 list is printed at the end of the file.

Once the EDLOG created program has been sent to the CR23X, it can be saved in the Flash memory program storage area using the * D Mode (Section 1.8).

CHANGING VALUES IN * 4 TABLE

Enter the * 4 Mode by keying "* 4"; "04:00" is then displayed. At this point it is possible to jump to any valid * 4 location by keying the desired location number and pressing the A key. For example, when the display shows 04:00 and the desired location is 80, key in the number 80, press the A key and the display will show "80:XXXXX." where XXXXX. is the value stored in location 80. Pressing the "A" key advances to the next * 4 location, and the "B" key backs up to the previous location. If a * 4 location is not assigned in the datalogger program, it can not be displayed in the * 4 mode.

To enter a value in a * 4 location, advance to the desired location, key in the number and enter it by pressing the "A" key. The value is not entered if the "A" key is not pressed.

Entering a new value causes the datalogger to stop logging. Logging resumes when the program is compiled. Upon compiling (* 0 or * 6), all current * 4 values are incorporated into the program. For this reason, whenever changes are made in the * 4 mode, make sure that all * 4 values are correct before exiting the * 4 mode.

Removing or adding an instruction to a program residing in the datalogger disables the * 4 mode. An instruction parameter may be edited without any adverse affect. If the * 4 mode is disabled, it may be reactivated by downloading the program to the datalogger or, if the program was saved to Flash storage, retrieving the program from the stored program area.

The * C mode (Section 1.7) may be used to secure the datalogger program and the * 4 mode entries. The lowest level of security prevents access to the * 1, * 2, and * 3 modes. Higher levels of security block * 4.

The CR23X will not respond to the * 4 command if any of the following conditions exist.

- the program that was downloaded does not contain any * 4 assignments.
- a program that was downloaded has since been hand edited.
- Security is blocking access to * 4.

1.1.5 COMPILING A PROGRAM

When a program is first loaded, or if any changes are made in the * 1, * 2, * 3, * 4, * A, or * C Modes, the program must be compiled before it starts running. The compile function checks for programming errors and optimizes program information for use during program execution. If errors are detected, the appropriate error codes are indicated on the display (Section 3.10). The compile function is executed when the * 0, * 6, or * B Modes are entered and prior to saving a program listing in the * D Mode. The compile function is only executed after a program change has been made and any subsequent use of any of these modes will return to the mode without recompiling.

When the * 0 or * B Mode is used to compile, all output ports and flags are set low, the timer is reset, and data values contained in Input and Intermediate Storage are reset to zero.

When * 0 is used, one of the following status lines will be displayed: Running Table 1, Running Table 2, No Active Program, Running Table 1, 2, Tables Not Running.

When the * 6 Mode is used to compile data values contained in Input Storage, the state of flags, control ports, and the timer (Instruction 26) are unaltered. Compiling always zeros Intermediate Storage.

1.2 SETTING AND DISPLAYING THE CLOCK - * 5 MODE

The * 5 Mode is used to display or set time. When "*5" is entered, time is displayed. It is updated approximately once a second or longer depending on the rate and degree of data collection and processing taking place. The sequence of time parameters displayed in the * 5 Mode is given in Table 1.2-1.

To set the year, day, or hours and minutes, enter the * 5 Mode and advance to display the appropriate value. Key in the desired number and enter the value by keying "A". When a new value for hours and minutes is entered, the seconds are set to zero and current time is again displayed. To exit the * 5 Mode, key "*" and the mode you wish to enter.

When the time is changed, a partial recompile is done automatically to synchronize the program with real time.

Changing time affects the output and execution intervals in which time is changed. Because time can only be set with a 1 second resolution, execution intervals of 1 second or less remain constant. Averaged values will still be accurate, though the interval may have a different number of samples than normal. Totalized values will reflect the different number of samples. The pulse count instruction will use the previous interval's value if an option has been selected to discard odd intervals, otherwise it will use the count accumulated in the interval.

TABLE 1.2-1. Sequence of Time Parameters in * 5 Mode

Key	Display ID:DATA	Description
* 5	HH:MM:SS	Display current time
A	Year XXXX	Display/enter year
A	Day of Year XXXX or MMDD XXXX	Display/enter day of year 1-365(366) or press D and enter as month and day, such as 1012 for October 12. D toggles back to Day of Year.
A	Time HHMM	Display/enter hours:minutes
A	Seconds SS	Display/enter seconds

1.3 DISPLAYING/ALTERING INPUT MEMORY, FLAGS, AND PORTS - * 6 MODE

The * 6 Mode is used to display and/or change Input Storage values and to toggle and display user flags and ports. If the * 6 Mode is entered immediately following any changes in program tables, the program will be compiled and run.

NOTE: Input Storage data and the state of flags, control ports, and the timer (Instruction 26) are UNALTERED whenever program tables are altered and recompiled with the * 6 Mode. Compiling always zeros Intermediate Storage.

TABLE 1.3-1. * 6 Mode Commands

Key	Action
* 6	Enter * 6 mode
A	Advance to next input location or enter new value
B	Back-up to previous location
C	Change value in first input location on display (followed by keyed in value, then "A")
D	Display/alter user flags 1 through 8
1	Display/alter user flags 11 through 18
0	Display/alter ports
#	Display current location and allow a location number to be keyed in, followed by "A" to jump to that location

1.3.1 DISPLAYING AND ALTERING INPUT STORAGE

When "*6" is entered, the keyboard/display will read "Mode 06 Enter Loc". One can advance to view the value stored in input location 1 by keying "A". To go directly to a specific location, key in the location number before keying "A". For example, to view the value contained in Input Storage location 20, key in "*6 20 A". The left portion of the display shows the location number and the 9-character label assigned to that location in the programming portion (EDLOG) of Campbell Scientific's PC208W datalogger support software. If the value stored in the location being monitored is the result of a program instruction, the value on the display will be the result of the most recent scan and will be updated each time the instruction is executed. When using the * 6 Mode from a remote terminal, a number (any number) must be sent before the value shown will be updated.

Input locations can be used to store parameters for use in computations. To store a value in a location, or change the current value, key "C" while monitoring the location, followed by the desired number and "A".

If an algorithm requires parameters to be manually modified during execution of the Program without interruption of the Table execution process, the * 6 Mode can be used. (If parameters will not need modification, it is better to load them from the program using Instruction 30.) If initial parameter values are required to be in place before program execution commences, use Instruction 91 at the beginning of the program table to prevent the execution until a flag is set (see the next section). Initial parameter values can be entered into input locations using the * 6 Mode C command. The flag can then be set to enable the table(s).

If the program is altered and compiled with * 0 Mode, all values previously entered via * 6 C will be set to zero. To preserve * 6 C entered values, compile with * 6 after changing the program.

1.3.2 DISPLAYING AND TOGGLING USER FLAGS

If D is keyed (for Flags 1 to 8), or 1 is keyed (for Flags 11 to 18) while the CR23X is displaying a location value, the current status of the user flags will be displayed in the following format: "0001.0010". The characters represent the flags, the left-most digit is Flag 1 (or 11) and right most is Flag 8 (or 18). A "0" indicates the flag is low and a "1" indicates the flag is high. In the above example, Flags 4 (or 14) and 7 (or 17) are set. To toggle a flag, simply press the corresponding number. To return to displaying the input location, press "A".

Entering appropriate flag tests into the program allows manual control of program execution. For example, to manually start the execution of Table 2: enter Instruction 91 as the first instruction in Table 2. The first parameter is 25 (do if Flag 5 is low), the second parameter is 0, go to end of program table. If Flag 5 is low, all subsequent instructions in Table 2 will be skipped. Flag 5 can be toggled from the * 6 Mode, effectively starting and stopping the execution of Table 2.

1.3.3 DISPLAYING AND TOGGLING PORTS

> **NOTE:** The switched 12 V port is displayed as “control port 9.” Other port options are not available on the switched 12 volt channel.

The status of the CR23X ports can be displayed by hitting "0" while looking at an input location (e.g., * 6 A 0). Ports are displayed left to right as SW12, C8, C7, ... , C1 (opposite to the flags). A port configured as output can be toggled by hitting its number while in the port display mode. There is no effect on ports configured as inputs.

On power up all ports are configured as inputs. Instruction 20 is used to configure a port as an output. Ports are also configured as outputs by any program control commands which uses the port as an output (pulse, set high, set low, toggle).

1.4 COMPILING AND LOGGING DATA - * 0 MODE

When the * 0 Mode is entered after programming the CR23X, the program is compiled and the display shows "Running Table" followed by the active program table numbers. The display is not updated after entering * 0.

> **NOTE:** All output ports are set low, the timer is reset, and data values in Input and Intermediate Storage are RESET TO ZERO whenever the program tables are altered and the Program is recompiled with the * 0 Mode. The same is true when the programs are compiled with * B or * D.

To minimize current drain, the CR23X should be left in the * 0 Mode when logging data, and by turning off the display by pressing #.

1.5 MEMORY ALLOCATION - *

1.5.1 INTERNAL MEMORY

When powered up, “Hello” is displayed while a self check is performed. The total system memory is then displayed in K bytes. The size of memory can be displayed in the * 0 mode. A "--" after the number displayed means that the memory test was aborted. The number shown indicates how far the test progressed before aborted.

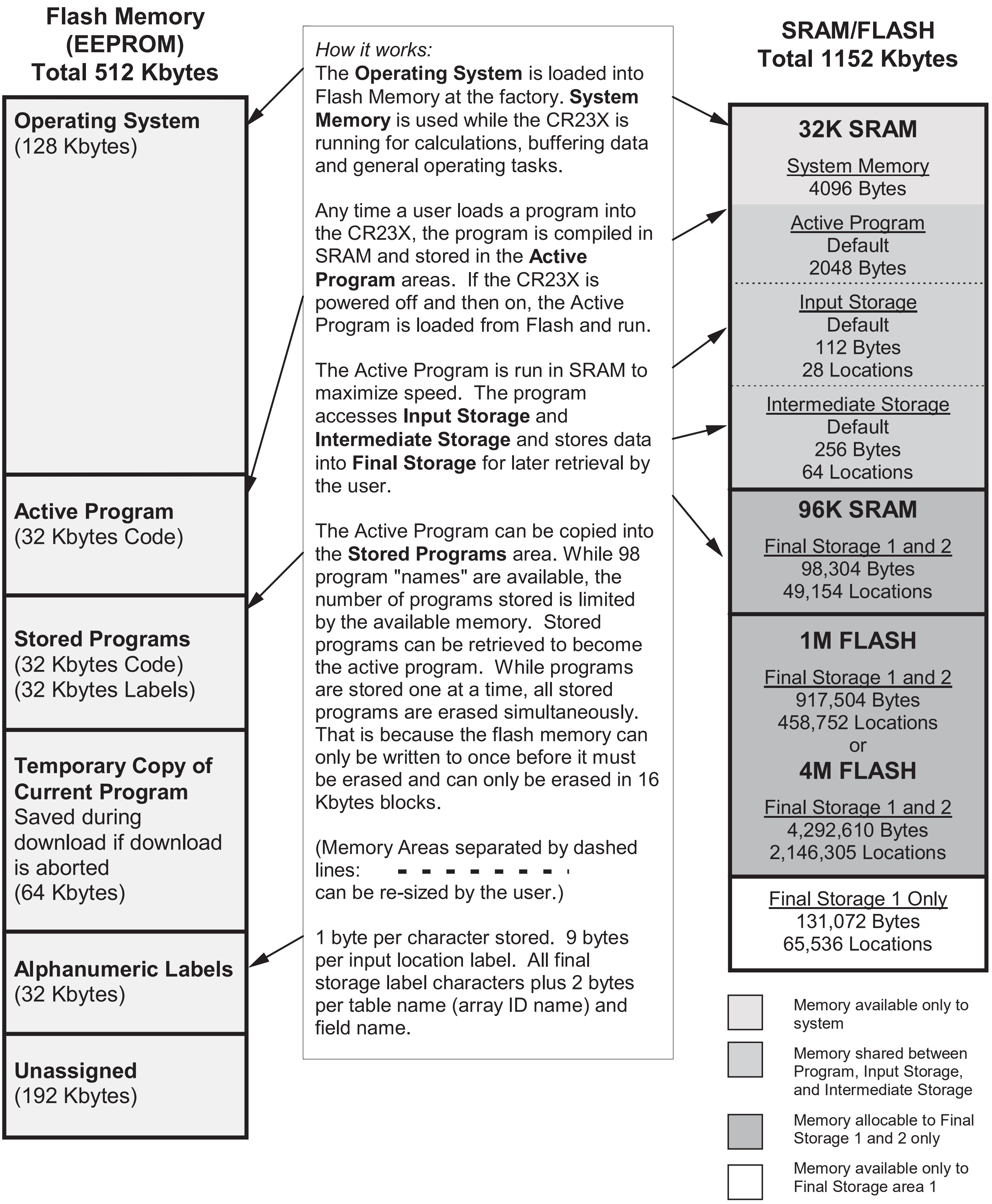
Input Storage is used to store the results of Input/Output and Processing Instructions. The values stored in input locations may be displayed using the * 6 Mode (Section 1.3).

Intermediate Storage is a scratch pad for Output Processing Instructions. It is used to store the results of intermediate calculations necessary for averages, standard deviations, histograms, etc. Intermediate Storage is not accessible by the user.

Final Storage holds stored data for a permanent record. Output Instructions store data in Final Storage when the Output Flag is set (Section 3.7). The data in Final Storage can be monitored using the * 7 Mode (Section 2.3).

Each Input or Intermediate Storage location requires 4 bytes of memory. Each Final Storage location requires 2 bytes of memory. Low resolution data points require 1 Final Storage location and high resolution data points require 2. Section 2 describes Final Storage and data retrieval in detail.

Figure 1.5-1 lists the basic memory functions and the amount of memory allotted to them.



How it works:
The **Operating System** is loaded into Flash Memory at the factory. **System Memory** is used while the CR23X is running for calculations, buffering data and general operating tasks.

Any time a user loads a program into the CR23X, the program is compiled in SRAM and stored in the **Active Program** areas. If the CR23X is powered off and then on, the Active Program is loaded from Flash and run.

The Active Program is run in SRAM to maximize speed. The program accesses **Input Storage** and **Intermediate Storage** and stores data into **Final Storage** for later retrieval by the user.

The Active Program can be copied into the **Stored Programs** area. While 98 program "names" are available, the number of programs stored is limited by the available memory. Stored programs can be retrieved to become the active program. While programs are stored one at a time, all stored programs are erased simultaneously. That is because the flash memory can only be written to once before it must be erased and can only be erased in 16 Kbytes blocks.

(Memory Areas separated by dashed lines: - - - - - - - - can be re-sized by the user.)

1 byte per character stored. 9 bytes per input location label. All final storage label characters plus 2 bytes per table name (array ID name) and field name.

░ Memory available only to system

▒ Memory shared between Program, Input Storage, and Intermediate Storage

▓ Memory allocable to Final Storage 1 and 2 only

□ Memory available only to Final Storage area 1

FIGURE 1.5-1. CR23X Memory

1.5.2 * A MODE

CAUTION: Reallocating memory will result in all data being lost.

The * A Mode is used to 1) determine the number of locations allocated to Input Storage, Intermediate Storage, Final Storage Area 2, Final Storage Area 1, and Program Memory; 2) repartition this memory; 3) check the number of bytes remaining in Program memory; 4) erase Final Storage; and 5) to completely reset the datalogger.

A second Final Storage area (Storage Area 2) can be allocated in the * A Mode. The default number of locations allocated for Storage Area 2 is 0. Final Storage Area 1 is the source from which memory is taken when Final Storage Area 2 is increased. When Final Storage Area 2 is reduced, Final Storage Area 1 memory is increased.

When * A is entered, the first number displayed is the number of memory locations allocated to Input Storage. The "A" key is used to advance through the next 6 windows. Table 1.5-2 describes what the values in the * A Mode represent.

Memory allocation defaults at reset to the values in Table 1.5-1.

The sizes of Input, Intermediate, Final Storage Area 2, and Program Memory may be altered by keying in the desired value and entering it by keying "A".

The maximum size of Input and Intermediate Storage and the minimum size of Final Storage are determined by the memory installed (Table 1.5-1). A minimum 64 Input location and 65,536 Final Storage Area 1 locations will ALWAYS be retained. The size of Intermediate Storage may be reduced to 0.

TABLE 1.5-2. Description of * A Mode Data

Keyboard Entry	Display ID: Data	Description of Data
* A	01: Input Locations XXXX	**Input Storage Locations.** Default 28, minimum of 1, maximum of 7138. This value can be changed by keying in the desired number.
A	02: Intermediate Locs XXXX	**Intermediate Storage Locations.** Default 64, maximum of 7137. This value can be changed by keying in the desired number. **Enter 0 then recompile, and the CR23X will assign the exact number needed**. Entering 0 may also result in the CR23X erasing all data whenever the program is changed and compiled.
A	03: Final Storage 2 XXXXX	**Final Storage Area 2 Locations** (CR23X-4M). Default 0, minimum of 0, maximum of 507,905 (2,080,769). Valid inputs are 0…32,769 or 49,153 and 32,768*N, where N is an integer. Changing this number automatically reallocates Final Storage Area 1.
A	04: Final Storage 1 XXXXX	**Final Storage Area 1 Locations** (CR23X-4M). Default 573,441 (2,146,305), minimum of 65,536, maximum of 573,441 (2,146,305). This number is automatically altered when the memory allocation for Final Storage Area 2 is changed.
A	05: Alloc. Program Bytes +XXXXX	**Bytes allocated for user program.** Default 2048, minimum 116, maximum 28,552. The number of bytes to assign to program memory can be keyed in to change the size of program memory. Changing the size of program memory results in all data being erased. **Enter 0 and the CR23X will assign the exact number needed above 116**. Entering 0 will also result in the CR23X erasing all data whenever the program is changed and compiled. **Key in 98765 to completely reset datalogger.**

A	06: Prog. Bytes Unused +XXXXX	**Bytes free in program memory.** The user cannot change this window. It is a function of window 5 and the program.
A	07: Prog. Bytes Available +XXXXX	The user cannot change this window. It is a function of Window 5 and total available memory.
A	08: Label Bytes Used +XXXXX	The user cannot change this window. It is a function of the program.
A	09: Label Bytes Free +XXXXX	The user cannot change this window. It is a function of Window 8 and the program.

Input Storage, Intermediate Storage, and Final Storage are erased when memory is repartitioned. This feature may be used to clear memory without altering programming. The number of locations does not actually need to be changed; the same value can be keyed in and entered.

If Intermediate Storage size is too small to accommodate the programs or instructions entered, the "E:04" ERROR CODE will be displayed in the * 0, * 6, and * B Modes. The user may remove this error code by entering a larger value for Intermediate Storage size. Intermediate Storage and Program Memory can be automatically allocated by entering 0 for their size. When automatic allocation is used, all data are erased any time the program is exchanged and recompiled. Final Storage size is maximized by limiting Intermediate Storage and Program Memory to the minimum necessary. The size of Final Storage and the rate at which data are stored determines how long it will take for Final Storage to fill, at which point new data will write over old.

After repartitioning memory, the program must be recompiled. Compiling erases Intermediate Storage. Compiling with * 0 erases Input Storage; compiling with * 6 leaves Input Storage unaltered.

ENTERING 98765 for the number of bytes to allocate for program memory COMPLETELY RESETS THE CR23X. All memory is erased including any stored programs and memory is checked. Memory allocation returns to the default. The reset operation requires approximately 5 minutes for a CR23X. Memory reset can be aborted by pressing any key on the keypad, or raising the ring line high.

1.6 MEMORY TESTING AND SYSTEM STATUS - * B

The * B Mode is used to check the status of the program's operating system, memory, and lithium battery. Table 1.6-1 describes what the values seen in the * B Mode represent.

A signature is a number which is a function of the data and the sequence of data in memory. It is derived using an algorithm which assures a 99.998% probability that if either the data or its sequence changes, the signature changes. The algorithm used to calculate the signature is described in Appendix C.

The signature of the program memory is used to determine if the program tables have been altered. The program signature is calculated only at compile time. In the background FLASH memory of the program is periodically checked against RAM memory of the program. If a byte is different, an E08 watchdog error is flagged.

During the self check on reset, the signature computed for the OS is compared with a stored signature to determine if a failure has occurred. The operating system (OS) signature is calculated in the background of 8 bytes per second and is updated at least once every three days. It is also done when memory is reset or a new operating system is downloaded.

The contents of windows 6 and 7, Operating System (OS) version and version revision, are helpful in determining what OS is in the datalogger. As different versions are released, there may be operational differences. When calling Campbell Scientific for datalogger assistance, please have these numbers available.

Window 13 is a real time display (updated every 0.1 seconds) of the "Program Time", the time it takes Table 1 to execute. The resolution is 0.407 µs, and the range is 6.826 seconds. To read this time as part of a datalogger program, see the description for Instruction P130.

TABLE 1.6-1. Description of * B Mode Data

Keyboard Entry	Display ID: Data	Description of Data
* B	01: +XXXXX	Program memory Signature. The value is dependent upon the programming entered and memory allotment. If the program has not been previously compiled, it will be compiled and run.
A	02: +XXXXX	Operating System (OS) Signature
A	03: XXXXX	Memory Size, Kbytes (Flash + SRAM). "--" indicates that a full memory reset was aborted.
A	04: XX	Number of E08 occurrences (Key in 88 to reset)
A	05: XX	Number of table overrun occurrences (Key in 88 to reset)
A	06: +X.XXXX	Operating System version number
A	07: XXXX.	Version revision number
A	08: +X.XXXX	Lithium battery voltage (measured daily)
A	09: XX	Low 12 V battery detect counter (Key in 88 to reset)
A	10: XX	Extended memory error counter (Key in 88 to reset)
A	11: +X.XXXX	Extended Memory time of erase, seconds. If >5 at room temperature, flash memory may be wearing out. Contact CSI for replacement information.
A	12: XX	Low 5 V counter (Key 88 to reset)
A	13: +X.XXXX	Program Time (0.407 µsec resolution, range is 6.826 seconds, above 6.826, time = 6.826 + displayed value)
A	14: +X.XXXX	Panel Temperature (updated at least every 1 to 2.8 minutes)
A	15: XX	Coprocessor Revision
A	16: XX	Coprocessor Status
A	17: XX	CPLD Revision

TABLE 1.7-1. * C Mode Entries

SECURITY DISABLED

Keyboard Entry	**Display ID: Data**	**Description**
* C	01: XXXX	Non-zero password blocks entry to * 1, * 2, * 3, * A, and * D Modes, telecommunication S command.
A	02: XXXX	Non-zero password blocks * 4, * 5, and * 6 except for display.
A	03: XXXX	Non-zero password blocks * 5, * 6, * 7, * 8, * 9, * B, and all telecommunications commands except A, L, N, and E.

SECURITY ENABLED

Keyboard Entry	**Display ID: Data**	**Description**
* C	12: 0000	Enter password. If correct, security is temporarily unlocked through that level.
A	01: XX	Level to which security has been disabled. 0 -- Password 1 entered (everything unlocked) 1 -- Password 2 entered 2 -- Password 3 entered

1.7 * C MODE -- SECURITY

The * C Mode is used to block access to the user's program information and certain CR23X functions. There are 3 levels of security, each with its own 4 digit password. Setting a password to a non-zero value "locks" the functions secured at that level. The password must subsequently be entered to temporarily unlock security through that level. Passwords are part of the program. If security is enabled in the active program, it is enabled as soon as the program is run when the CR23X is powered up.

When security is disabled, * C will advance directly to the window containing the first password. A non-zero password must be entered in order to advance to the next window. Leaving a password 0, or entering 0 for the password disables that and subsequent levels of security.

Security may be temporarily disabled by entering a password in the * C Mode or using the telecommunications L command (Section 5.1). The password entered determines what operations are unlocked (e.g., entering password 2 unlocks the functions secured by passwords 2 and 3). Password 1 (everything unlocked) must be entered before any passwords can be altered.

When security is temporarily disabled in the * C Mode, entering * 0 will automatically re-enable security to the level determined by the passwords entered.

The telecommunications L command temporarily changes the security level. After hanging up, security is reset.

1.8 * D MODE -- SAVE OR LOAD PROGRAM

The * D Mode is used to save or load CR23X programs, to set the degree to which memory is cleared on powerup, to set the datalogger ID, to set communication to full or half duplex, and to set the display's contrast level.

Programs (* 1, * 2, * 3, * 4, * A, * C, and * D Mode data) may be stored to and from computers, internal flash memory, and Storage Modules. Several programs can be stored in the CR23X Flash Memory and later recalled and run using the * D Mode or Instruction 111.

PC208W automatically makes use of the * D Mode to upload and download programs from a computer. Appendix C gives some additional information on Commands 1 and 2 that are used for these operations.

When "*D" is keyed in, the CR23X will display "13: Enter Command". A command (Table 1.8-1) is entered by keying the command number and "A".

TABLE 1.8-1. * D Mode Commands

Command	Description
1	Send (Print) ASCII Program
2	Load ASCII Program, * 0 Compile
2--	Load ASCII Program, * 6 Compile (canceled by * D 1 0 mode)
6	Store Program in Flash
7	Load Program from Flash
7N	Save/Load/Clear Program from Storage Module N
8	Set Datalogger ID
9	Set Full/Half Duplex
10	Set Powerup Options
11	Set Display Contrast Level
12	Set Initial Baud/Set RS232 Power
13	Set Compile Option

If the CR23X program has not been compiled when the command to save a program is entered, it will be compiled before the program is saved. When a program is loaded, it is immediately compiled and run. When a command is complete, "13:0000" is displayed; * D must be entered again before another command can be given.

If a program download is aborted, the CR23X will reload the program in its flash into RAM, compile it, and run it.

TABLE 1.8-2. Program Load Error Codes

E 94	Program Storage Area full
E 95	Program does not exist in flash
E 96	Storage Module not connected or wrong address
E 97	Data not encountered within 30 sec.
E 98	Uncorrectable errors detected
E 99	Wrong type of file or Editor Error

1.8.1 INTERNAL FLASH PROGRAM STORAGE

Several programs can be stored in the CR23X Flash Memory and later recalled and run using the * D Mode. The Flash Electrically Erasable Programmable Read Only Memory is non-volatile memory that can only be erased in 16K blocks. The CR23X has 512K of Flash EEPROM memory, one 16K block is reserved for storing extra programs.

When a program is loaded and compiled, it is saved as the active program. The active program will be automatically loaded and run when the CR23X is powered up. Automatic loading of the program can be aborted by pressing any key while “Hello” is showing on the CR23X display; the display will show “Program Aborted.” (If a Storage Module with a program 8 is connected when the CR23X powers-up, the Storage Module program 8 will be loaded into the CR23X and become the active program.)

The active program can be stored in internal flash memory program storage with * D command 6 (Table 1.8-3). Programs can be retrieved with * D command 7 (Table 1.8-4).

TABLE 1.8-3. Storing Program in Internal Flash

Key entry	Display
* D	13: Enter Command 00
6 A	06: Program ID 00

You may now enter one of the following options:

xx A	Save active program as number xx, xx may be 1-98.
A B	Scroll forward and backward through saved program numbers. The numbers are displayed in the order saved.
9 9 A 9 9 A	Clear all saved programs.
0 A	Display number of bytes free in saved program area.

TABLE 1.8-4. Retrieving a Program from Internal Flash

Key entry	Display
* D	13: Enter Command 00
7 A	07: Program ID 00

You may now enter one of the following options:

xx A	Retrieve program number xx (the most recent xx saved). To have the program compile like * 6 (no resetting of input locations, flags, or ports) press C (xx--) before A.
0 A	Erase active program (i.e., load a blank program; memory allocation and Final Storage are reset).
A B	Scroll forward and backward through saved program numbers.

Scrolling through the program names begins with the oldest program. "A" advances to the next newer program, "B" backs up to the next older program. While scrolling, at any time typing in a number (xxA) will cause a save or a retrieve operation.

Each program saved takes up the memory required for the program + 6 bytes.

Flash memory can only be written to once before being erased. Because it can only be erased in 16K blocks, if one stored program is to be erased, all must be erased. To allow revising a program and storing it with the same number (name) as an earlier version, the same number can be used by more than one saved program. When retrieving a program, the programs are searched beginning with the last program saved; the most recently saved version will be retrieved. An older program with a duplicate name cannot be retrieved. When the flash program memory is full, all programs must be erased before any more can be added (error 94 will be displayed).

1.8.2 PROGRAM TRANSFER WITH STORAGE MODULE

Storage Modules can store up to eight separate programs. The Storage Module and Keyboard/Display or Modem/Terminal must both be connected to the CR23X. After keying * D, the command 7N, is entered (N is the Storage Module address 1-8, Section 4.4.1). Address 1 will work with any Storage Module address; the CR23X will search for the lowest address Storage Module that is connected. The command to save, load, or clear a program and the program number (Table 1.8-5) is entered. After the operation is finished "13:0000" is displayed. Error 96 indicates that the Storage Module is not connected or the wrong address was given.

TABLE 1.8-5 Transferring a Program using a Storage Module

Key entry	Display
* D	13: Enter Command 00
7 N A	7N: Save, Load, Clr 00

(N is Storage Module address 1-8)

You may now enter one of the following options:

1 x	Save Program x to Storage Module (x = 1-8)
2 x	Load Program x from Storage Module (x = 1-8)
3 x	Erase Program x in Storage Module (x = 1-8)

The datalogger can be programmed on power-up using a Storage Module. If a program is stored as program number 8, and the Storage Module is connected to the datalogger I/O at power-up, program number 8 is automatically loaded into the active program area of the datalogger and run.

1.8.3 SET DATALOGGER ID

Command 8 is used to set the datalogger ID. The ID can be moved to an input location with Instruction 117 and can then be sampled as part of the data.

TABLE 1.8-6 Setting Datalogger ID

Key Entry	Display
* D	13: Enter Command 00
8 A	08: Datalogger ID 0XXX

Where XXX are 0s or the current ID. You may now key in the ID (1-12, 14-254).

1.8.4 FULL/HALF DUPLEX

The * D Mode can also be used to set communications to full or half duplex. The default is full duplex, which works best in most situations.

TABLE 1.8-7. Setting Duplex

Key entry	Display
* D	13: Enter Command 00
9 A	09: Comm Duplex 0x

If x=0 the CR23X is set for full duplex.
If x=1 the CR23X is set for half duplex.

You may now change the option:

0 A	Set full duplex
1 A	Set half duplex

1.8.5 SETTING POWERUP OPTIONS

Setting options for the Program on Powerup allows the user to specify what information to retain from when the datalogger was last on. This allows Flag/Port status, the User Timer, and the Input/Intermediate Storage to be cleared or not cleared.

Table 1.8-8. Setting Powerup Options

Key entry	Display
* D	13: Enter Command 00
1 0 A	10: Power Up Option 0X

Where X is the powerup option currently selected. You may now change the option:

0 A	Clears input locations, ports, flags, user timer, and intermediate storage locations.
1 A	Clears intermediate storage only (leaves Input Storage, Flags/Ports, and User Timer as is).
2 A	Doesn’t clear anything.

1.8.6 SETTING DISPLAY CONTRAST

The CR23X automatically adjusts the LCD display contrast for temperature within two seconds after power-up. If necessary, the user can fine tune the default contrast in the * D mode. The user entered adjustment is valid only for the specific temperature range wherein the adjustment was made. If the CR23X temperature moves out of that range, the default setting for the next range controls the contrast. See “Telecommunications” R command information on changing default settings in each temperature range.

TABLE 1.8-9.

Key Entry	Display	Comments
* D	13: Enter Command 00	
1 1 A	11: Program Stopped 11: Dsply Contrast xxxx	xxxx is the current setting. Key in new setting followed by an A or . . .
A	11: D Dark C Light xxxx	Press A to darken, B to lighten
* 0	LOG	Save setting, restart program

1.8.7 SET INITIAL BAUD / SET RS232 POWER

Table 1.8-11 shows the option codes available for setting the initial baud rate. Setting the initial baud rate forces the CR23X to try the selected baud rate first when connecting with a device. By indexing the option, the “Computer RS232” port can be powered up. Power up of the RS232 port puts 9 volts on pins 1 (DTR) and 8 (RTS), and 8 volts on pin 2 (TX).

TABLE 1.8-10. Set Initial Baud Rate / Set RS232 Power

Key Entry	Display	Comments
* D	13:Enter Command 00	
1 2 A	12: Connect Baud Rate 00	
X C A	12: Connect Baud Rate 0X--	Enter Baud Rate Code X (Table 1.8-11). Index (--) is optional.

TABLE 1.8-11. Baud Rate Codes

X = 0	300 Baud
X = 1	1200 Baud
X = 4	2400 Baud
X = 5	4800 Baud
X = 2	9600 Baud
X = 6	19.2 K Baud
X = 7	38.4 K Baud
X = 3	76.8 K Baud
X-- =	RS232 Power On

1.8.8 SET PROGRAM COMPILE OPTION

Table 1.8-13 shows the option codes available for setting the program compile option. This setting will affect the program compile when the program is downloaded from the PC or a SM192/716 Storage Module. It also affects compiling with * B and the arcane * D 7 command. Keyboard or Remote Keyboard compiling with * 6 and * 0 is not affected by this setting. If a .DLD file has this setting, it will affect the compile operation AFTER the .DLD file is downloaded.

TABLE 1.8-12. Set Program Compile Option

Key Entry	Display	Comments
* D	13:Enter Command 00	
1 3 A	13: Compile Option 00	
1 A	13: Compile Option 01	Sets Compile like * 6

TABLE 1.8-13. Compile Option Codes

0	Compile like * 0 (See Section 1.4)
1	Compile like * 6 (See Section 1.3)
2	Do not clear intermediate storage