

CORE

REFERENCE MANUAL

©1987 CL9

The above notice applies to all copyrighted materials included with this product, including but not limited to this manual and the software embodied within the product.

Warranty

For warranty information see the card enclosed in the CORE package.

Written by Ron Karr, Robert Retzlaff, Tom Johnson, Chuck Van Dusen & Meg Beeler

Published in U.S.A.

CL9 Customer Service: (800) 544-4259

Introduction

What CORE is

Congratulations on your purchase of CORE, the Controller of Remote Equipment. CORE is a **universal programmable master remote controller with a built-in timer**.

1. CORE is a **master controller** because it allows you to operate **all** of your infrared remote-controlled equipment with one device.
2. CORE is **universal** because it can learn, or "capture", the signals from virtually every infrared remote control device on the market today. If you look on the upper back of your CORE, you'll notice a ribbed section. This Master Module houses one of the most powerful infrared transmitters in the industry. It was designed to be removable because no one knows for certain what new technology tomorrow will bring. Our engineers are committed to keeping your CORE from becoming obsolete. We are constantly checking CORE with new equipment as it comes on the marketplace to ensure optimum compatibility.
3. CORE is **programmable** because it is actually a high-powered microcomputer! CORE is powered by two custom microprocessors and has 36 kilobytes of memory. This programmability allows you to create powerful sequences of commands which you can activate with the touch of a key.
4. In addition, CORE has a **built-in timer** which allows you to operate CORE automatically any time of day, even when you're not around.

What You Can Do with CORE

If you're wondering what types of things you can do with your CORE, here are some ideas:

1. You have two or more pieces of electronic equipment which have infrared remote-control units. You enjoy the convenience of remote control, but the problem is you have too many remotes! It's cumbersome to have all these controllers lying around, and confusing to figure out which one to pick up. With CORE, you only need one controller!
2. There are certain things that you do over and over again which require several **button**-presses on your controllers. For example, you turn on the power to your TV set and also your VCR; then you set the volume to a certain level. With CORE, you can do all of that with a single keypress!
3. You're tired of the sound of your alarm clock, and you'd like to have your TV set wake you up instead. You want the volume to come on at a low level, then increase gradually every 2 minutes in case you have trouble getting out of bed. No problem with CORE!
4. You want to record several shows on different channels while you're on vacation. The problem is: you have a timer on your VCR which will record at different times, but you can't change the channel without using the cable box, and the cable box doesn't have a timer. With CORE, it's an easy job--you can have timed control over all your remote equipment.

You have great flexibility in using CORE with your remote equipment. You can use the basic commands only, or you can use as much of CORE's powerful programming language as you like.

No other infrared controller on the market today has the flexibility or the expandability of CORE. We hope you enjoy exploring the many possible applications of CORE and would like to hear about any ideas you come up with. We also welcome any suggestions you might have. Be creative and have fun!

How to Use this Manual

"**Getting Started with CORE**" gets you up and running. It covers the following basic functions:

- Setting the CORE clock
- Learning infrared commands
- Programming sequences of commands
- Unattended operation

"**CORE Reference**" is the main body of the manual. It covers all of the features of CORE in detail. You can find anything you need to know about CORE in this section. Use the table of contents and/or the index to find the section you're looking for.

1. **Programming Basics:** the fundamentals of CORE programming, including learning remote commands and organizing your commands.
2. **Further Programming Features: Sequences:** advanced features of CORE programming.
3. **Timed Operation:** how to use the CORE clock and program CORE for unattended operation.
4. **Editing:** how to review and change what you've programmed.
5. **Modifying Captured Codes:** how to fine-tune CORE for optimum performance.
6. **Maintenance and Troubleshooting:** how to care for CORE, and what to do if you have problems.

Appendices:

- A: **CORE Display Symbols:** shows the relationship between keypresses and symbols which appear in the display.
- B: **Syntax Reference Guide:** a summary of the CORE programming language, showing the required keypresses for all programming functions.
- C: **Glossary:** contains definitions of the terms used throughout this manual. Each term appears in **boldface** the first time it appears in the manual.

"**CORE Parts**" is a fold-out illustrated guide at the back of this manual, labeling all the parts you will use in programming and operating CORE. Refer to it as you follow the instructions throughout the manual.

Other Resources

"**CORE Journal**" is a separate book which comes with your CORE. You will use it to make a written record of the commands, sequences and timed operations you teach each CORE key.

"**Factory Presets**" is an insert sheet describing the remote controller commands that come pre-programmed with CORE.

CL 9 Customer Service

If you have any problems or questions, call CL 9 Customer Service at (800) 544-4259.

Chapter 4: Editing	55
Editing a Program	55
Modifying a Program	57
Built-In Verification	58
Reviewing the Event Queue	59

Chapter 5: Modifying Capture Codes	61
Altering the Number of Sends	61
Changing the Output Frequency of a Code	63
Altering the Sequence Send Rate	65
Problems With Repeaters	66

Chapter 6: Maintenance and Troubleshooting	67
Core Maintenance	67
Batteries	67
To Store CORE for an Extended Period	69
To Clean CORE	69
Resetting CORE	70
Troubleshooting	72
CORE Messages and Tones	73

Appendix A: CORE Display Symbols	74
Appendix B: Syntax Guide	77
Appendix C: Glossary	83

Index	87
--------------	-----------

Getting Started with CORE

Examining CORE




First, fold out the illustrated guide "CORE Parts" at the back of this book so that you can refer to it as you go through this chapter.

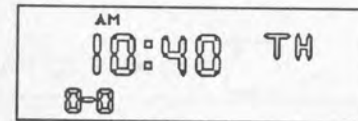
The diagram shows CORE divided into three major sections:

1. The **LCD display** at the top.
2. The 20 **operating keys**. These include the 16 **location keys**, plus the (PAGE) KEY, plus the 3 **direct-access PAGE keys**.
3. The **programming keys**, which are hidden beneath the programming key cover. Whenever you need to access these keys, grasp the finger grips on both sides near the top edge, and slide the cover down until it locks open.



Waking Up the Display

When you removed CORE from the box, most likely the LCD display was off (not displaying any characters). The display automatically goes off after a short period of non-use to conserve the battery (display **time-out**). To turn on the display again (**wake-up**):

1. Press the  key (you can press any key, but pressing  is a good way to insure that CORE doesn't send any commands when you just want the display to come on). You should see the time displayed at the top (the time may or may not be correct), and the **P** symbol at the bottom.
2. Press the  key again, and you'll see one or two characters on the bottom line (referred to as the **status line**). These characters represent what is called the **current location** (explained in the next section).






In case the display didn't come on, you may have to press the **reset** switch. This could happen if your CORE got jostled during shipping. It might also happen if you dropped your CORE.

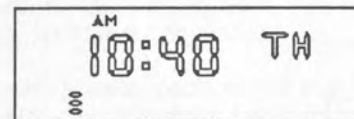
1. Open the door to the programming keys.
2. Take a paper clip and insert it into the reset hole between the  (KEY) and  (DO) keys. The display should come on and you're ready to continue. See "Resetting CORE" in Chapter 6 for further information.

Setting the Time

Even if the time on your CORE is correct, go through the following exercise to learn how to adjust the time.

1. Make sure the programming key door is open.
2. Make sure the display is on.
3. Press the  SET key.


When you press  SET, the symbol, , appears on the status line:

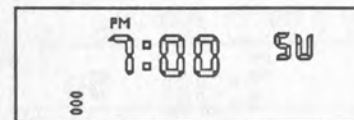



4. Change the time using either the **direct-digit** or **up-down (scanning)** method.

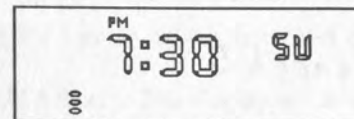
Direct-digit method:


Enter the first digit of the time. For example, if the time is 7:32 PM,

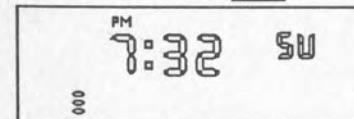
press . You will see 7:00 in the display:



Then press the next digit, in this case .




Then press the last digit, in this case .





Note: If the time is a four-digit number (10:00 thru 12:59), you will have to enter 4 digits.


Up-down (scanning) method:

Change the hour and minute using the paired rocker keys:



 = hour down



 = hour up

 = minute down

 = minute up

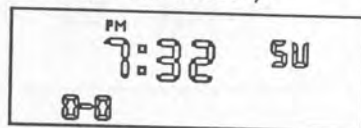
5. Set the day and AM/PM.

Press the  or  key if you need to change the AM/PM settings.

Use the   key (day down; day up) to change the day.

6. Press  when the **time display** is accurate.

The status line reverts to the current location-- the last **PAGE** and **KEY** pressed (more on this in the next section).



seconds cleared?

Operating the CORE

The next step is to store and execute commands from your remote controllers in CORE.



Using the Pre-stored Commands

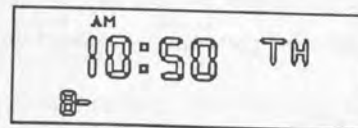
When you bought your CORE, it actually came with pre-stored commands from a few remote controllers.





The insert sheets explain which controllers they are. If you own any equipment which works with these controllers, you may not need to enter the commands yourself. Even if you don't, you will see from the following examples how CORE executes commands.

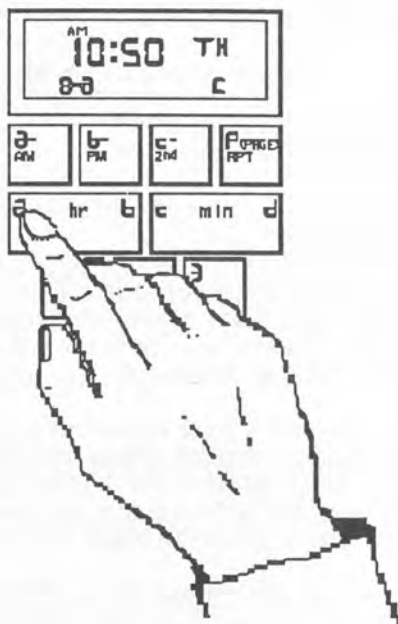
If you look at the insert sheets, you'll notice that the diagrams are labeled PAGE 1, PAGE 2, etc... For example, PAGE 8 is labeled TV. On the left-hand side there is a diagram which looks like the CORE keyboard, and on the right-hand side is a list of commands labeled **a** through **F** and **0** through **9**. These sheets resemble the pages in the CORE Journal.


In order to execute one of the commands on PAGE 8, you would do the following:

1. Press the  key. A **P** appears in the display. Press the  key. The display will read **8-**. The CORE is now set up to execute commands on PAGE 8.



2. In order to send TV Power, which is stored on key , press the  key and hold it down. The display will read **8-a**. Also, there will be a  on the right-hand side of the display which will remain until you take your finger off the key. The  means that an infrared signal is being transmitted. If you own this TV, your power should go on (see illustration, next page).




Similarly, to send Channel Up, you would simply press key .

Now suppose you want to operate your VCR. The VCR commands are stored on PAGE 9. Simply press , then , and then the key corresponding to whatever VCR command you want to execute.

Location: PAGE and KEY

You can see from these diagrams that the word PAGE refers to a group of 16 commands from one (or more) of your remote controllers. Each of the 16 PAGES on CORE is available to store up to 16 separate **programs**. All pre-stored programs contain single IR commands. As you will see later, a program is not restricted to a single IR command.



All programs stored in CORE are referred to by their **location**, which consists of a PAGE number and a KEY number. You can change pages by using the  key, followed by a key from 0 to 9 or a to F. When you change PAGES, each of the **operating keys** has a new function.

Choosing the PAGE-KEY Location

The first thing to do is to decide which PAGE you are going to put the commands from your controller on. We'll refer to your controller as the **source remote controller**.


Note: If your controller has more than 16 buttons you want to use, you'll have to use more than 1 PAGE. For now, we'll assume you are going to use a single PAGE.


Let's assume you're going to use PAGE 0 for now, since this PAGE is not pre-programmed with any commands. In order to choose PAGE 0, you do the following:


1. Press . You'll see a **P** symbol in the display.
2. Press . You'll see **0 -** in the display.

Let's start with the Power command on your controller. (If the controller you're working with doesn't have a Power button, pick any other button for this example.) You need to decide which key on PAGE 0 will be used to store the Power command from your controller. For this example, you will use KEY a on the CORE. Later, you may decide to put it somewhere else. That's perfectly OK.

In order to choose key a:


3. Press the  key. The display will show **0 - 0**.

The **0** is the symbol for the  key.

4. Press the  key. The display will show **0 - a**.

Now you are ready to store a command in PAGE 0, KEY a.

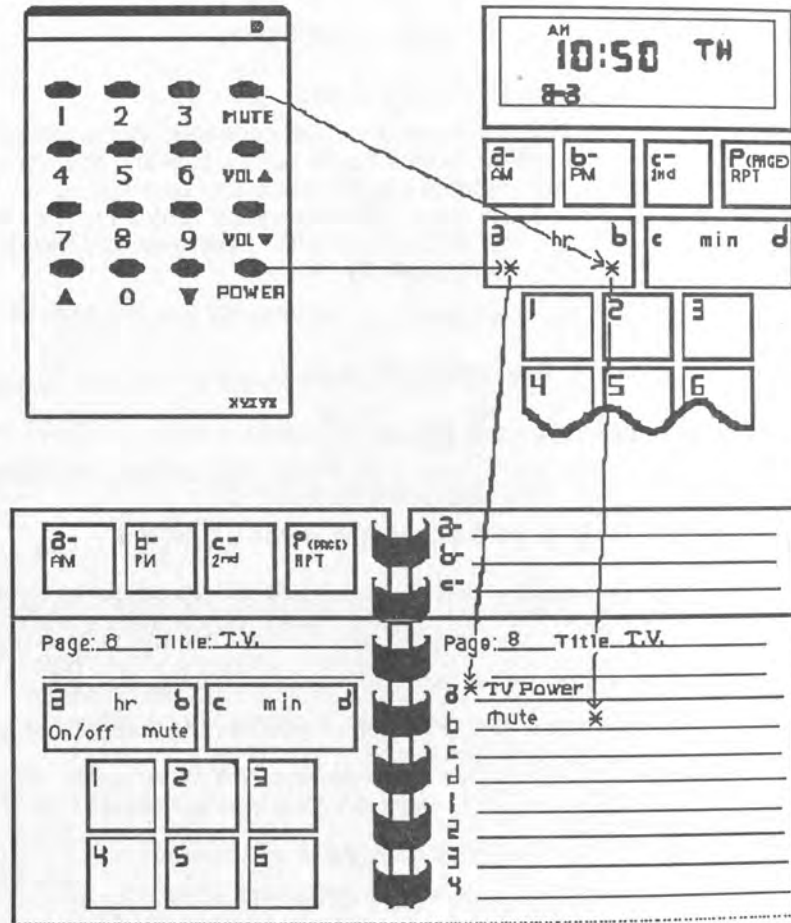
Note: If you had skipped step 3, you would still be set up to store a program in PAGE 0, KEY a. However, any program already stored at

that location would have been **executed**. Using the  key prevents the command from executing. If you don't care about that, you can save time by omitting step 3.

Storing Your Commands in CORE

The next step is for you to store the commands from your own remote controllers in CORE (assuming that you have one or more remote controllers that are not pre-programmed into CORE). First, you'll learn how to **teach** CORE a single infrared command from your controller. From there it will be a simple matter to teach CORE ALL the commands from ALL your controllers.

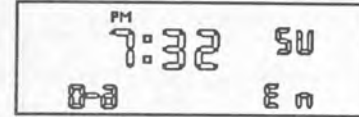
The illustration below shows the process of storing commands in CORE. You start with a command on your remote controller, store that command at a CORE **PAGE-KEY** location, then record the command in your Journal.



Storing Your Commands in CORE

Learning the Infrared (IR) Command

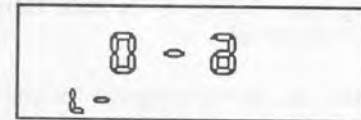
1. Press the **ENABLE** key. You should see **En** displayed at the extreme right of the status line.



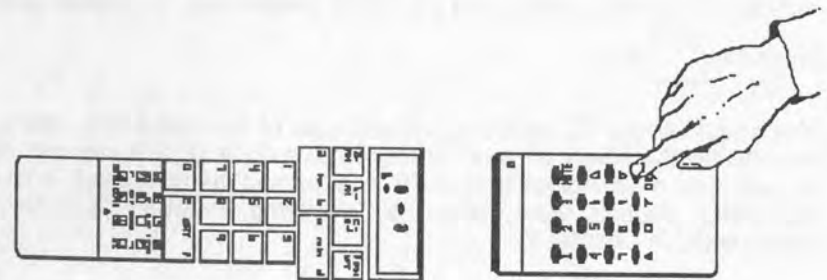
Note what happens if you press **ENABLE** again: the **En** disappears.

When the **En** is showing, you can teach CORE new commands. When it is not showing, no teaching is allowed.

2. Press **DO** to prepare CORE to learn the remote controller command. (If you haven't enabled this teaching session, you'll hear a **reject tone** (a single short low-frequency buzz). Repeat the previous step on Enabling.) When you press **DO**, the PAGE-KEY location replaces the time display and you see the **L** symbol followed by an unblinking cursor on the next entry position.




CORE is now ready to accept a signal from your remote controller. Align your CORE and the other remote controller head to head. You may need to adjust the distance between the two controllers (up to six or so inches apart).
(see illustration, next page)




Storing Your Commands in CORE 9

3. Press the Power button (or whichever button you've chosen) on your controller. Hold it down until you hear a tone. One of three things will happen:



- a. You see the **Capture** symbol  and hear the CORE alert tone (a single high-frequency chime).





This means that CORE has accepted the signal from your controller. You should release the button and proceed with step 4.


- b. You hear the CORE reject tone (a low-frequency buzz) and don't see the  on your display. This means CORE hasn't learned the command. Release the button for a moment; then press it again. If you continue to get the reject tone, try varying the distance and/or angle between the CORE and your controller.

- c. You hear no tone at all. In this case:
--Make sure the source controller is working by seeing if it controls your equipment. The batteries could be low.
--Try different angles and distances between the two controllers
--See "Teaching your CORE a Remote Controller Function" in Chapter 1 for further details.

4. Press  to signify the end of the program for this particular key. (As you'll see later, it's possible to have more than a single IR command on one CORE key.) You will hear another alert tone, and the display will go back to displaying the current time and  on the status line. Now the Power command (or whichever command you chose) from your remote controller will be stored on PAGE 0, KEY a.

5. The next step is to make sure CORE has in fact learned the command. In order to test this, simply point the CORE toward your equipment and press the  key.

You should see a  at the right-hand side of the status line. More importantly, the power on your TV should go on! (Or off, if it was on). If it does, you have verified that CORE has learned the command. If the command doesn't work, refer to "Verifying Remote Controller Commands" in Chapter 2.

Note: At any time during a teaching session,  can be pressed to stop the teaching session and return CORE to the display with the time and current location. Whatever was previously stored at the current location will be preserved.

6. Record the name of the command at the appropriate place in the CORE Journal.

Creating Sequences

You've learned how to transfer all the individual commands from your remote controllers to individual PAGE-KEY locations in CORE. An additional, powerful feature of CORE allows you to combine commands into useful **sequences**.

In the normal course of operating your remote equipment, there are certain combinations of commands that you will use frequently. For example, if you want to view a videotape, you would press Power on your TV controller, Power on your VCR controller, and Play on your VCR controller. Every time you wanted to play a tape, you'd use the exact same series of commands. With CORE, you can combine all 3 commands into a single keypress! Once you've programmed this sequence, all you need do is press a single key (or possibly a PAGE-KEY combination) and CORE will do the rest.

Note: This description is a sample of what you might do if you have a TV and a VCR. If you don't have this equipment, substitute other equipment in a similar type of sequence. Keep your program sequences simple at first until you have a better understanding of how they work.

Suppose you have the following commands:

VCR Power: Stored at location **E-a**.
TV Power: at location **8-a**.
and VCR Play: at location **9-b**.

What you are going to do is create a program which:

- a. performs all three of the above actions.
- b. will be activated when you press a single key.


The steps for programming a sequence are:

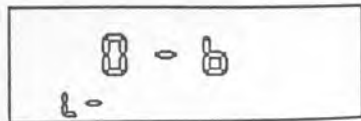
1. Decide which location (PAGE and KEY) to store the program in. Let's assume you're going to use PAGE **0**, KEY **b**. To set up this location,

--if you're not in PAGE **0** already, press , .

--Press , .

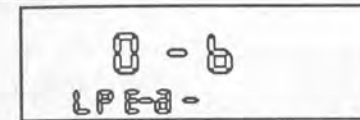
2. If the **En** isn't showing on your display, press Enable.

3. Press . You'll now see the following display:



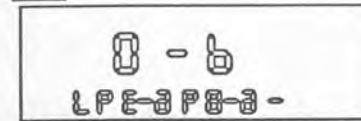
Note: This is the same procedure you followed for capturing infrared commands. The difference is that this time we don't need to use any other controllers.

4. Press , , .



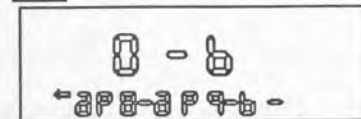
This tells CORE to activate the program stored at PAGE **E**, Key **a**, which is VCR Power.

5. Press , , .








This sequence tells CORE to activate the program stored at PAGE **8**, Key **a**, which happens to be TV Power.




6. Press , , .



This tells CORE to activate the program stored at PAGE **9**, KEY **b**, which is VCR Play. You'll notice that after you hit the **P**, the **L** disappeared off the left-hand side of the display, and an arrow appeared. The arrow lets you know that the program is too long to fit on the display.

Optional: Press . You'll see that the last character you pressed, the **b**, disappears. You can use the  key when you make a mistake and want to erase a character. Press  again to restore the **b**.

7. Press . You'll see the OK symbol () briefly, and you'll hear an alert tone. Then the display will revert to its previous state.

8. Test the sequence to make sure it works as it should. First, make sure the power on your TV and VCR are OFF. Point your CORE in the direction of the equipment and press , , . 0-6 will be displayed on the status line. The power to the VCR should go on, as well as the power to the TV, and then the VCR should play.

If part or all of the program did not work:



- First review your program to make sure you entered it correctly. To do this, see Chapter 4 on Editing.
- If it's correct and still doesn't work, refer to Chapter 2 of the Reference for more information on sequences.

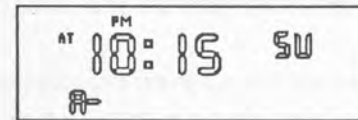
Programming CORE for Unattended Operation


You've just seen that you can create a program to perform one or more actions, and then activate that program by pressing a key. With CORE's **built-in timer**, you have an additional option: you can have CORE activate the program for you **automatically** at a time and day of your choosing.


For this example, you can use the sequence you created which played a tape on your VCR (or any sequence you have already created--just substitute your program location for the one in this example). You could have it wake you up by telling CORE to execute the sequence at 7 AM tomorrow morning. The problem is, you'd have to wait until then to see if it worked. So for illustration's sake, you can tell CORE to execute the program **3 minutes** from now.



Here's how you do it:




- Make sure Enable is on.
- Press the  key. This begins programming of what is called an **event timer**. You'll see the symbol  on the status line, and AT to the left of the time.

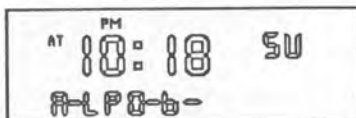


- As you did in the section on setting the clock, adjust the time to the time that you want the program to happen. For this example, press the minute-up key (the  key) three times. As you can see, the time will advance three minutes.

Note: Be careful if the minutes "wrapped around". For example if the time had been 1:58 and you increased it by 3, the time would now say 1:01. What you really want would be 2:01, so you have to press the hour-up () key to increase the hours.

- Press the  Key. You'll now see the symbol  displayed on the bottom line with a flashing cursor next to it.

5. Press , , . This represents the sequence that we did before which executed TV Power, VCR Power and VCR Play.





6. Press .

To sum up what you just did: You told CORE to execute the program at PAGE 0, KEY b (which you had already programmed); at the time you put in the display (in this case, three minutes from now).

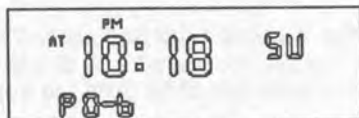
Reviewing the Event Queue

Suppose you'd like to check to make sure you set the event timer correctly. Or you may have programmed several timers, and you want to see which one is due to execute next. To do this, you can review the **event queue**. The event queue is a list CORE keeps of all the timers you've set which haven't executed yet. Here's the procedure:

1. Press .
2. Press .


If you hear a reject tone, it means there's **nothing** in the event queue. You'll have to repeat the above steps to program the timer.



If you programmed the timer correctly, you'll see a display like the following one:



The display shows the time of the event and the location to be executed at that time (in this example PAGE 0 KEY b). You'll see AT in the upper left-hand corner. If the AT is flashing, it means Enable is on. If it is solid, Enable is off.

3. If the display is correct, proceed to step 4. If there's an error in the time or location, you can **delete** the timer from the queue. Turn Enable on,

signified by a flashing AT. Now press . Since you only had one event in the queue, CORE will buzz once and then return you back to the main display. Now repeat the procedure for setting a timer.


4. Now to exit the event queue you can either press  or .

The next step is to verify that the timer executes on schedule. Make sure that the power on both your TV and VCR is off. Point CORE in the general direction of your TV and VCR and wait for the timer to execute.

Note: The timer may not execute exactly on the minute. For example, if you set the timer for 10:18 PM, it will execute somewhere between 10:18 and 10:19. For instructions on how to control the seconds count, see Chapter 3 of the Reference.

When the timer goes off, you will hear an **alarm** consisting of three alert tones. CORE will then activate your equipment exactly as when you pressed the key yourself.

Note: If at any time while you're reviewing the event queue you hear two alert tones, this means it's time for your event to happen. However, CORE will not execute the program while you are in the middle of reviewing the event queue. As soon as you leave the queue, you will hear the three alert tones and the timer will execute.

If the display timed out, you can wake it up by pressing the  key twice to verify that the time on the CORE is the time that you had set the event for. Also, you'll see a flashing "bell" which will indicate that an alarm had gone off.


See Chapter 3 of the Reference for more details on timed operations.

Chapter 1: Programming Basics


What is a CORE Program?



A CORE program is a sequence of actions assigned to a PAGE and KEY location. A program can contain remote controller commands, event timers or **intervals**, other program definitions, and valid operating key sequences. No program sequence can contain more than 64 keystrokes.


How CORE Storage Is Organized

- There are 16 PAGE locations. Physically, a new PAGE is accessed by using the  key, followed by one of the characters 0 through 9 and a through F.




- Each PAGE location subdivides into 16 KEY locations, also identified by the keys 0 through 9 and a through F. Each CORE program is identified by its PAGE-KEY location. Therefore there are 16 x 16 = 256 separate locations to store CORE programs.


- In addition, there are three special **direct-access PAGE keys**,  ,

 , and  . These keys can be used:

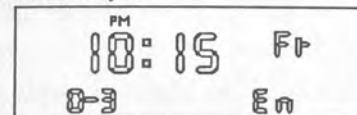
1. To provide quicker access to 3 of the PAGES. For example, to access PAGE a, you can simply press  . You do not have to

use the  key.

2. To store another program, as a "preamble" to entering the PAGE. Normally programs require both a PAGE and a KEY location. These special cases,  ,  and  , can have their own programs.

Therefore, pressing the  key can *execute* a program as well as setting up CORE for use on PAGE a.

- Hence you can store up to 259 programs—one at each location. The total of 259 is a combination of 16 PAGE locations times 16 KEY locations plus 3 direct PAGE access keys.



- The status line of the display shows the current PAGE and last KEY accessed. For example, in the above display, the current PAGE is 0 and the last key that was accessed was KEY 3.

Suggestions for Organizing Your Commands

There are many schemes you could conceivably use to organize your remote commands into the 259 locations on your CORE. Here are some suggestions for starting out:

1. First, you will generally want to reserve a separate PAGE or two for each of your controllers. If your controller has more than 16 buttons, you will need to use two PAGES. (If your controller has certain commands that you will use infrequently or not at all, you may want to save space by not storing them in CORE.)
2. It's a good idea to put similar functions on the same KEY location in each PAGE. This will help you in remembering where things are. For example, if you look at the "Factory Preset Programs" sheets, you'll notice that Channel Down and Channel Up commands are stored on KEYS E and F on several different PAGES; Volume Down and Volume Up are on KEYS c and d. Of course, it's logical to put numbered commands like 0 through 9 on the corresponding CORE KEYS.

Also, notice that the **key pairs a-b, c-d, E-F** should be used for **paired** commands like Volume Up/Down, Channel Up/Down or other commands which seem to go together.

3. You can reserve a few PAGES for special programs, such as those containing sequences and timers.
4. Keep in mind that PAGES a, b and c are special, in that they can be accessed quickly with the direct-access $\boxed{\text{a-AM}}$, $\boxed{\text{b-PM}}$, and $\boxed{\text{2nd}}$ keys. In addition, each of these keys can store its own program. Therefore it makes sense to use these for some kind of **configuration** program, one which turns power on several units, selects a particular station and/or sets the volume to a predetermined level. That way you can press, say, the $\boxed{\text{b-PM}}$ key, which executes the configuration program and sets you up to **operate** commands on PAGE b.

You also might want to put commands that you will use frequently on one of the direct-access PAGES. Commands such as Master Volume, Channel Up/Down and Mute, which work with several of your components, can then be accessed quickly no matter what PAGE you're currently in.

5. It may take some experimentation to determine the best way to use CORE's memory for your own convenience.

Choosing a Location for Programming

1. To choose a new PAGE: press the $\boxed{\text{PAGE/RPT}}$ key. You will see the CORE symbol **P** displayed in the lower left of the status line. Then press one of the keys **0** through **9** or **a** through **F**. The display will show the PAGE location with a trailing dash; the KEY location will be blank.

Note: If the display shows that the current PAGE is the one you want to choose, you can skip this step.

2. To choose one of the direct-access PAGES, you can use a shortcut: press the $\boxed{\text{a-AM}}$, $\boxed{\text{b-PM}}$, or $\boxed{\text{2nd}}$ key.

Note: Choosing a direct PAGE this way will also **execute** the program stored at that location, if any. If you want to select the direct PAGE location for programming without executing it, use method #1.




3. To choose a KEY within the current PAGE, press the $\boxed{\text{(KEY)}}$ key. You will see the CORE symbol **a** in the second position of the status line. Then press one of the keys between **0** through **9** or **a** through **F**.

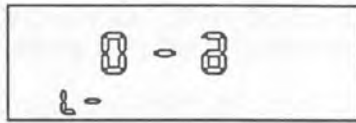
Note: To save time, you can skip pressing the $\boxed{\text{(KEY)}}$ key. You must be careful, however, because if there already is a program stored at that location, it will execute. This is particularly unwise if the stored program contains an event timer.

Once you have selected the location, you are ready to create a program that will be stored at that location. The next section covers the most fundamental type of program: capturing a single infrared command from one of your controllers.

Teaching CORE a Remote Controller Function

In this example you will be storing a command at location 0-a.

1. Press  to begin the teaching session. An ϵn will appear at the lower right of the status line. If the ϵn is already showing, you do not need to press .
2. Choose a location for the program to be stored in, as described in the section above.
3. Press . This is the beginning of any CORE program. Notice that the PAGE-KEY location replaces the time display and you see the \mathcal{L} (DO) symbol followed by a cursor in the next entry position:




If CORE is not Enabled, you will hear a reject tone (a single short low-frequency buzz) and return to the main display (see Step 1).



4. Align your CORE and the other remote controller head to head, about 3 to 6 inches apart. (Distances may vary more with some units.)
5. Press the source remote controller button you want to teach until you see the Capture symbol \mathcal{C} and hear the CORE alert tone. Then release the source remote controller button. A successful capture appears as below.



• If you heard a reject tone, you may need to experiment with the way the CORE and the other remote controller heads are facing each other. Some remote controllers issue infrared commands from the center, some from one or the other side, and some at an angle. Experiment with distance and angle of the CORE and the other remote controller until you find an alignment that works.

• **A special case—no tone:** (First, make sure that your source remote controller is functioning). If you don't hear *any* tone, try pressing the  key to complete the capture. If the capture works, you'll see a

reversed capture symbol \mathcal{J} on the status line, which is the symbol for a "one-shot." This is a special type of IR code which does not repeat when the key is held down. See the section below on "Repeating and Non-Repeating Commands" for further details on one-shots.

6. Press  after the command is captured. All CORE programs end with . Now the remote controller function from the source remote is stored at the PAGE-KEY location displayed on the status line.



7. Record the remote controller function in the *CORE Journal* at the appropriate location. (This example would go on PAGE 0 KEY a.)
8. Repeat steps 2 through 7 for the all the remote commands you want to store in CORE. You may find it easier to organize and record the remainder of the source remote controller commands in the *CORE Journal* before capturing them. Refer to the section "Suggestions for Organizing Your Commands" above and also to the insert "Factory Presets".

Verifying Remote Controller Commands

It is a good idea to **verify**—just after you teach—that the CORE keys have accurately learned the commands you've programmed them to perform.

1. Point CORE towards the component (TV, VCR, etc.).
2. Press the key corresponding to the program you've just stored. For example, if you stored the program at PAGE 0, KEY a, press the a key to activate the command. (In this case, you don't need to change the PAGE since PAGE 0 is still the current PAGE.)

When the key is pressed, CORE will transmit the remote command and display the location you've selected. A **C** will appear on the right of the status line on the display.

If the **C** does not appear, you've done something incorrectly. Try recapturing the code. If the **C** appears but your equipment does not respond as you expect:

- try it again. Hold the key down until the equipment responds.
- make sure the component is turned on (if you are verifying a command other than Power) and that it responds to the source remote controller.
- make sure there is a clear path between CORE and your component.
- teach the command to CORE again (sometimes CORE doesn't get a complete signal from your remote controller). Try varying the distance and/or angle between CORE and the remote you are trying to capture.

If you still can't get your unit to respond, you may need to change the output frequency of the signal. See Chapter 5 for information.

Repeating and Non-Repeating Commands

As you've seen, when you press a key to send an infrared command, the **C** symbol displays. In most cases, the **C** symbol will stay on for as long as you hold your finger on the key. There are some exceptions to this, however.

In general, the CORE does whatever the source controller command does. When you press a button such as Channel Up on your source controller, the channel continues to increase as long as the button is pressed. If you

capture this command on CORE, you should get the same result. The **C** will stay on for as long as the code transmits.

Note: If the **C** stays on but the signal does *not* repeat, you may have to use one of the modification procedures in Chapter 5. Refer to the section on "Problems with Repeaters."

Some commands, such as Power, will continue to send with the key held down, but the receiving equipment will only respond one time. In this case the **C** will stay on even though the receiver doesn't appear to be doing anything.

Some commands send only once ("one-shots") or twice ("two-shots") no matter how long you press the key. One-shots and two-shots are captured with the **EDIT** key, as explained above, and display the reversed capture symbol **]** during learning. If the command from the source controller appears to be a repeater (such as a Volume-Up command) but CORE captures it with the reversed capture symbol **]**, try this alternate method of capturing:

When you press the button on your controller the first time, you get neither an alert tone nor a reject tone. Instead of pressing the **EDIT** key, try pressing the source controller button one or two more times until you hear a tone. If you hear an alert tone, you will see the normal capture symbol **C** and you should have a successful capture. If you hear a reject tone, try again.

Capturing Multiple Commands

It's possible to capture more than one IR command in a single PAGE-KEY location. After the first IR, wait for the **C** and the alert tone; then repeat the process with additional commands on the same or other controllers. Each command will show another **C** symbol in the display. When you're done,


press **OK**. When you execute the program by pressing the appropriate key, the IR commands will be sent in sequence as captured.

The maximum number of IR's in a program is usually around 6 (it depends on the amount of memory each code requires). When you hear two reject tones, it means there's no more room left for any IR's in that program.

While it's possible to create sequences in this manner, we do not recommend this as a general practice, for the following reasons:


1. In case **any** of the commands turn out to have been captured incorrectly, you'll have to learn the entire program from scratch.
2. There is another, more flexible way to string IR commands together by means of program sequences. This will be covered in Chapter 2.

Canceling Execution of a Program

Any time you want to stop the execution of a program before it has completed, press the  **CANCEL** key. You will hear a reject tone and the program will stop.

Deleting a Program

There are times when you may wish to completely erase the program stored at a particular PAGE-KEY location. For example, you may decide that you have no need for some of the pre-stored commands that come with your CORE. Or you may have some programs that you decide you never want to use again, and want to insure that you don't inadvertently access these programs.

To delete a program, simply set the location, turn Enable on, and press  **DO OK** ! This has the effect of storing an empty or **null program** at that location.

Of course, you can avoid this by simply teaching a new command to the location, but in the (extremely unlikely) event that you ever fill up CORE's memory, you may need to use the above procedure to free up some memory space for new programs.

Chapter 2: Further Programming Features: Sequences

This chapter shows how to go beyond capturing a single infrared (IR) command at a PAGE-KEY location. You will learn to create sequences which combine two or more remote controller commands, and other useful programming features. Some examples of the ways these features can be used will be shown.

"Getting Started" showed a basic example of a sequence. This chapter explains in detail all the elements which can be part of a program, except for those relating to timed events. Timed events will be covered in Chapter 3.

Elements of a CORE Program

As discussed in Chapter 1, all programs begin with \square_{DO} and end with \square_{OK} . Enable must be on to begin a program.

When you press \square_{DO} , you see the \mathcal{L} symbol, followed by a non-blinking cursor. This cursor means that the program will accept either a keypress or an IR command at this point. If the cursor is blinking, the program will only accept a keypress. A CORE program can accept a maximum of 62 keypresses.

A CORE program can contain the following elements (*italics denote in which chapters the elements are discussed*):

(Chapter 1)

- An IR command

(Chapter 2)

- Indirect access to another CORE location
- Sequences: access to multiple locations
- Multiple IR commands
- A repeat loop
- Re-defining a location

(Chapter 3)

- A time-set command
- An event timer or interval

Indirect Access to Another CORE Location

The simplest type of program is one which executes a program stored at another location, simply by naming the location. For example, suppose

you've captured an IR code at PAGE 7, KEY 3. If you press the \square_3 key when in PAGE 7, CORE sends the IR command. Now suppose you create another program on PAGE d, KEY F using the following keystrokes:

\square_{DO} \square_{PAGE} \square_{RPT} \square_7 \square_3 \square_{OK} . If you press KEY F of PAGE d, CORE will send the identical IR code. In programming terms, we say that location d-F provides **indirect access** to the program at location 7-3. (If you had decided to put this program on the same PAGE as the original signal (PAGE 7), you could have defined the program without specifying the

PAGE: \square_{DO} \square_3 \square_{OK} would be sufficient.)

Notes on Indirect Access:


Although the two locations now contain the same IR code, there are some differences in the way they will act when you hold the key down and keep it down. The rule is this:








*If the program is defined by a single \mathcal{C} , the command will repeat when the key is held down. If it is defined by anything else, the command will be sent a finite number of times (called the **number of sends**), then stop, even if the key is held down.*



In other words, when you press KEY F of PAGE d, it executes a program defined as $\mathcal{P} 7 - 3$; therefore it will not repeat. Since the code does not repeat, it is very important to verify that this command actually controls your equipment. If it does not (and the original captured code on $\mathcal{P} 7 - 3$ does), you will need to modify the number of sends. Refer to "Altering the Number of Sends" in Chapter 5.

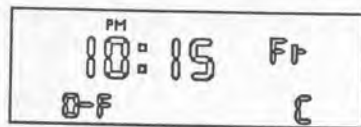
Sequences

Suppose you want to have a program which does two or more things: for example, send Power to your TV and Power to your VCR. First you need to make sure both commands have been captured at individual PAGE-KEY locations.

Assume TV Power is on PAGE 8 KEY a, and VCR Power is on PAGE E KEY a. Let's say you want to put the sequence at PAGE 0 KEY F. Simply create a program starting with  which contains the following keystrokes:

1.  . This changes the current PAGE from whatever it was to PAGE 8. Any KEY locations named subsequent to this will be assumed to be on PAGE 8.
2. . This tells CORE to execute the program stored at KEY a of the current PAGE (in this case, PAGE 8). This has the effect of sending TV Power.
3.  . This changes the current PAGE from 8 to E.
4. . This tells CORE to execute the program at PAGE E, KEY a: VCR Power.
5.  to end the program.




When you execute this program (by pressing the F key when in PAGE 0), the codes for TV Power and VCR Power will be transmitted. The  symbol will appear while the two codes are sent, and then stop. The status line will show .



Note: As discussed in the section on "Indirect Access" above, it is important to verify that the individual elements of this program work correctly, and modify the number of sends if necessary. For long programs, see the section on "Single-Step Verification" at the end of this chapter.

Changing the Current PAGE In a Program














Even though the program executed instructions on PAGE 8 and PAGE E, the current PAGE returned to PAGE 0, which shows in the display. It's possible to change the current PAGE only if the *very last command* in the program is a PAGE-change command. For example, after turning on TV Power and VCR power, it might be convenient to change the current PAGE to the PAGE your TV commands are stored on, which in this example is

PAGE 8. You would add the instructions   between steps 4 and 5 above: the program would send TV power, then VCR power, then change the current PAGE to 8. The display would read  and you would be set up to execute programs on PAGE 8.



There are times when operating CORE when you want to move frequently between two or more PAGES. This occurs when:

1. There are more than 16 commands on a particular controller, and you are forced to split them up into 2 PAGES.
2. You use 2 or more pieces of equipment simultaneously, such as the TV and VCR.


When a set of programs for a particular device is stored on two different PAGES, you can program one of the paired keys to function as a toggle between the two PAGES. For example, to go back and forth between

PAGES 0 and 9 with , you could program location   with    , and you could program location   with    .

Entering Long Programs

When you enter a program that has more than 6 characters (in addition to the initial ) , each new character appears in the 7th position of the display, the cursor appears in the 8th position, and all previous characters shift one position to the left. The  symbol appears on the left side of the display to indicate that there are more characters to the left which are not shown.

The Delete Key








If you make a mistake or change your mind during the creation of a program, you can backspace by using the  key. Each time you press this, the character to the left of the cursor is erased, and the cursor moves


to the left. You can only delete back to (but not including) the leftmost character displayed in the status line.

Recursion

As you can see, CORE is much more than an infrared controller. It's really a computer, with its own built-in programming language. As with all computers, it gives you power to do a great many things. It also gives you the responsibility to make sure you program it carefully to avoid unwanted results.

When programming sequences, you should be careful to avoid doing something like putting the following program on PAGE 1, KEY 1:








 . The problem with this program is that KEY 1 tries to execute KEY 1. This is a circular definition, like trying to define a word by using the same word. (In programming jargon, this is called **recursion**.) This will cause CORE to go into an (almost) infinite loop. Eventually CORE will get tired of this and will sound two reject tones. Any program which includes a call to KEY 1, such as       , will also have the same result.

Any time you press a key and CORE seems to "go off into space", you may be executing a recursive program. Try pressing the  key: you will hear a reject tone and the program will stop executing.

Repeat Loops

The CORE programming language contains a feature called a **repeat loop**. This is a shortcut way of sending the same command several times in a row. Any time you want to send the same command two or more times (such as a series of Volume-Up commands), you should use the repeat loop feature.

The repeat loop works as follows: Suppose the command that you wish to repeat is stored at PAGE 8, KEY d. You want to repeat this command 15 times. The part of the program that does the loop is as follows:

1.  : this represents the beginning of a loop.
 2.   : this represents the number of times the command will be repeated. It may be a one-or two digit number from 1 to 99.
 3.    : this is the location of the program to be repeated.
- Note:**
- a. You **must** specify both the PAGE and the KEY, even if the PAGE you want is the same as the current PAGE.
 - b. You cannot specify one of the direct access PAGES, **a**, **b** or **c**.
4.  : this represents the end of the loop.

When you execute the program at PAGE 0, KEY a, it will cause whatever is stored at PAGE 8, KEY d to be repeated 15 consecutive times.

The next section describes a practical example of using repeat loops.

Automatic Volume Adjustment

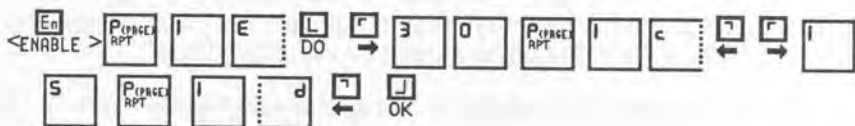
You may have a particular volume level that's comfortable for you, and you'd like to be able to set that level automatically, with a single keypress, no matter what level was previously set. For example, your teenagers turn the volume up to full blast and you want to turn it down to a tolerable level (or vice versa).

When using remote control to adjust volume, there is no single command to set a particular level—you must continuously send the Volume Up or Volume Down command until the volume is at the desired level. The idea behind automatic volume control is that you create a sequence which:

- turns the volume all the way down to its lowest level, no matter what level it is currently at.
- turns it up to the level you've predetermined.

The sequence you program to accomplish this will consist of two repeat loops. In the first one, CORE will send Volume Down the number of times needed to get the volume to zero; in the second one, CORE will send Volume Up the number of times needed to get it to your desired level.

For the following program, assume that Volume Up is stored at PAGE 1, KEY d, and Volume Down is on PAGE 1, KEY c. The program has a repeat loop which sends Volume Down 30 times, then another repeat loop which sends Volume Up 15 times. Assume you are storing this program at PAGE 1, KEY E. The following keystrokes are necessary to enter the program.



Note: The number of times you'd repeat the Volume Up and Volume Down commands in a loop of your own depend, of course, on your remote controller. Your receiving equipment may not respond fast enough to catch all the infrared commands you send within a loop. If you notice that there's not a one-to-one correspondence between the loop count and the action the loop is performing, experiment until you determine exactly how many times the loop should be performed.

An option to consider is to add a Mute command at the beginning and the end of this sequence. This way you would not hear the volume go down and up; the volume would simply come on at the correct level.

Re-Defining a Location

One of the interesting features of CORE is the ability to dynamically change the definition of a program during operation. Why would you want to do this? The following example shows how this feature can be used.

Splitting the Functions of an Alternate-Action Key

Some commands on your controllers perform one function when sent once, then another (opposite) function when sent a second time. Examples of this are Power On/Power Off, Pause/Still, and Video/TV. These are called **alternate-action** or **toggle** functions.

If you're in a room with your equipment, you can tell at a glance whether the stereo or television set is on, and can therefore decide whether or not to press the CORE key that represents the Power On/Off for that equipment. If, however, you are programming CORE to record a program on the VCR when you know you'll be away from home, you must be sure that CORE sends the On command only if the equipment is off.

With CORE, it's possible to *split* the functions into two separate keys, for example On and Off. With this method, if the On key is activated when the equipment is already on, nothing will happen.

The example that follows splits the power switch required to watch a television. You can use this example for your own alternate action key splits, substituting the appropriate PAGE-KEY locations for the ones listed.

1. First, capture the actual infrared code (say, the On/Off button) from the source controller. In this example, assume Television Power On/Off is stored on PAGE 4 KEY E.
2. Choose two CORE locations to be the separate On and Off keys. In this example, PAGE a, KEY a will be defined as Power On, and PAGE a, KEY b will be Power Off.
3. Reserve two CORE locations as intermediate keys, which are needed to make this programming sequence work. In this example, these will be PAGE 3, KEY 7 and PAGE 3, KEY 8. You should never access these keys directly, or you will interfere with the correct operation of the alternate-action sequence.

4. Program PAGE a, KEY a (Power On) as:

. This means "execute the program at P 3 - 7 ."

Program PAGE a, KEY b (Power Off) as:

. This means "execute the program at P 3 - 8 ."

5. Define the intermediate key on P 3 - 7 as:

: start the program.

: send the basic Power command.

: set up PAGE a, KEY a for programming. Note that the key is required here to mean that KEY a is not to be executed but rather **redefined**.

: this is what PAGE a, KEY a is being programmed to do, namely nothing. This is referred to as a null program.

: now set up PAGE a, KEY b for programming.

: program it to execute the program at PAGE 3, KEY 8.

: the final OK to end the program.

6. Define the intermediate key at P 3 - 8 as:

: start of program.

: execute the Power command.

: program P 3 - b to do nothing.

: program P 3 - a to execute P 3 - 7 .

: end of program.

The gist of what's happening here is:






The very first time you execute P 3 - a , it will activate the program at P 3 - 7 . This in turn sends Power (our equipment should be off, so this

will have the effect of being a Power On key) and then **redefines** P 3 - a so that the next time it is accessed, it will do nothing! Also P 3 - b is redefined as P 3 - 8 , which will send Power (in this case, Off), and then redefine both keys back to the way they were originally. In other words, when P 3 - a is pressed, the Power command is sent, turning the equipment on; then the Power command is transferred over to P 3 - b , which when pressed will turn the equipment off; and so on.


Note: The risk of programming alternate action keys is that someone may manually operate a device or use its original remote controller and interfere with the synchronization you have established. For example, if you set up the programs described above, and pressed the Power On key, then manually turned off the TV, the functions of the two switches would be reversed! To re-synchronize the device with your alternate-action sequences, you can either manually adjust the equipment, or use the source controller, or use the CORE location containing the original captured command (which in the above example would be PAGE 4, KEY E).


Single-Step Verification of Long Programs


Suppose you've created a program which is supposed to perform several actions. When you execute it, you discover that it doesn't do what you think it should do. You want to determine which part of the program is incorrect. CORE gives you the ability to **single-step** through the program. That is, you can execute the program one step at a time to see what happens.




1. Press  to enter Single-Step mode.
You'll see  and  at the ends of the status line.
2. Select the location to be verified, the same way you would normally choose a location. You'll see the first 8 characters of the program on the status line. The  on the right-hand side of the display will flash.
3. Press  to execute the first step, or **element**, of the program.




Note: A program element is any part of a program which can stand alone by itself. The following are examples of program elements:

 : a call to KEY 1.

 : a change of PAGE.

 : an event timer (see Chapter 3 for a discussion of event timers).

- When you press , the step will execute **in its entirety**. For example, a program element such as  calls KEY 1 of the current PAGE. If KEY 1 contains a long, complex program, the entire program on KEY 1 will be executed. If you wanted to verify the program on KEY 1, you would have to single-step through that program instead.
- Any element containing an event timer will execute immediately, instead of putting the timer into the event queue for later execution. For example, if the program contains a 10-minute interval, it would be annoying to have to wait ten minutes to see if it executed. Instead, after pressing , you will hear the alarm signal and the event will execute.
- Repeat loops will be executed in their entirety as a single step.
- After the step has executed, the display will scroll to the left so that the next step is shown at the left of the status line.

4. Press  to verify (execute) each step until all steps of the program have been executed (verified). This way, you should be able to determine which part(s) of the program aren't working correctly and take action to correct it.
5. Pressing  at any time will complete the remainder of the program without stopping and return CORE to normal use. Pressing  will abort the program.



Chapter 3: Timed Operation

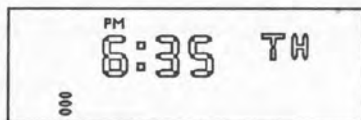
In this chapter you will learn to:

- Set the CORE clock to the correct time.
- Program CORE for unattended operation.
- Use the full power of CORE's built-in timers.

Setting the Clock

The built-in quartz crystal real-time clock is a special feature of CORE. The first step in using CORE for timed control of your remote equipment is setting the clock to the current time.

1. Press  and the display symbol  appears on the status line. The current time, day, and AM or PM is showing.

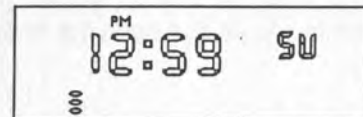
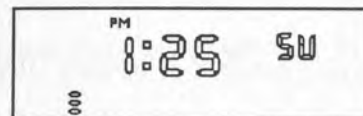
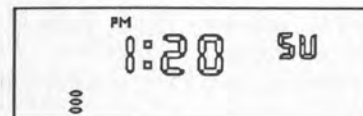
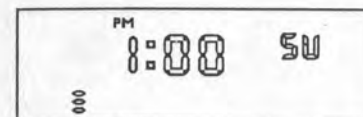


2. Adjust the time by using either the direct-digit or scanning methods.


- 2a. **Direct-digit entry:** Set the time by pressing the digit keys. Start with the hour, then do the minutes.

The first digit pressed clears the time and appears in the single hours position. The second and third digits pressed appear in the minutes positions. If a valid fourth digit is pressed, the first three digits entered shift to the left.

If the hour you are setting is 10, 11, or 12 (two digits), the full time won't appear until you press the fourth digit. If you are setting the time to 12:59, the successive display looks like this:



If you try to enter a nonexistent time, such as 25:89, CORE sounds a reject tone to let you know it will not accept the time.






You may press  at any time to start over.

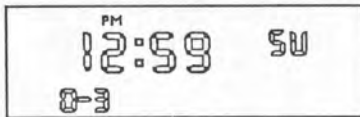
- 2b. **Scanning (up-down) method:** In any order, press the hour or minute key-pairs. Press the left side of the key-pair to scan to earlier times. Press the right side of the key-pair to scan to later times.

Hour key-pair down <  > up.

Minute key-pair down <  > up

Note: It is possible to use a combination of direct-digit and scanning keys.

3. Set the day by pressing  for an earlier day of the week,  for a later day of the week.
4. Press  or  if you need to change to AM or PM.
5. Press  when the time display is accurate. The status line reverts to the current location, with the new time now showing.



Seconds count: The seconds count (which is not displayed) is adjusted according to the following rules:

- if you have used *only* up-down keys in setting the time, the seconds count will be preserved when you press . For example, if the original time was 6:35:27 PM, when you press , the new time will be 12:59:27 PM.
- if you have used *any* direct-digit keys, the seconds count will be reset to zero when you press . In this case, the new time will be 12:59:00 PM.

Setting the Time Within a Program

In addition to setting the time directly, it is also possible to create a program which, if executed, will change the time for you. This is not a common occurrence: the main reasons for doing this would be to be able to set the clock ahead or back one hour when changing to or from Daylight Savings Time, or when crossing one or more time zones in travel.

To create a program on PAGE 0, KEY 2 which sets the clock ahead one hour, first make sure Enable is on and that shows in the display.

Then enter the following keys: .

Note that the key in this context means "hour up." The result is that when you execute this program (by pressing KEY 2 of PAGE 0), the time will increase by an hour. Try it!

You can create a corresponding program to set the clock down by one hour; simply substitute the key for the key. See the section on "Programming Daylight Savings Time" below for an example of how CORE can be programmed to do this time change in advance.

Event Timers

You use event timers when you want CORE to operate automatically at a particular time or times.

The CORE can be used to replace the timers of your other remote-controlled equipment, such as your VCR. CORE lets you program an almost unlimited number of timed operations, rather than the limited number that most devices offer. CORE also provides timed control of remote-controlled equipment that does not have its own timer, such as a cable converter.

There are two ways to program CORE for timed operation:

1. **Absolute event timers** are programmed using the key. They are used to have CORE operate at an exact time and day, such as "7:30 PM Friday."
2. **Relative event timers (intervals)** are programmed using the key. They are used to have CORE operate at a time relative to the present, such as "3 minutes from now" or "4 hours from the time this program executes."

Absolute Event Timers

To set an absolute event timer, you need to specify two things:

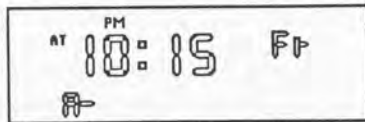
- When you want the event to happen (i.e. time and day).
- Which stored program you wish to execute at the specified time.

Specifying the Time and Day

Specifying the time and day of a timed event is very similar to the process used for setting the time, with a few exceptions. The process goes as follows:

1. If ϵn isn't displayed, press $\boxed{E_n}$ **ENABLE**.
Note: Enabling is required here, unlike time-setting.
2. Press \boxed{R} **AT** to initiate event timer setting.

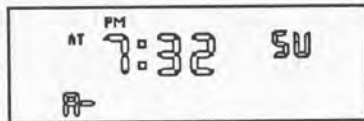
When you press \boxed{R} **AT**, the ϵn symbol appears on the status line. ϵn disappears and the **AT** symbol appears to the left of the time display. ϵn reappears when you complete the timer sequence.



3. Set the desired event time using the direct-digit or scanning methods, as explained above in the section on time-setting. (**Note:** this has no effect on the current time, which will be displayed again as soon as you finish setting the event timer.)

Seconds count: The display shows the day, hour and minute that the event will take place. The seconds are determined as follows:

- If you use the direct-digit method (i.e. if you press any of the digits 0 to 9 to set the event time), the event will occur right on the minute, i.e. at :00 seconds.
- If you use **only** the up-down keys, the seconds count will be preserved. For example, if you set an event for 7:32 PM, and the time when you finished setting the event was 10:15:30 PM, then the event will occur at 7:32:30.



4. **Timer options:** The following optional features can be used with absolute event timers:

Monday through Friday (MF):

When you use the day key-pairs to set event timers, you'll notice that **MF** (Monday...Friday) is inserted between Sunday and Monday on the display. **MF** lets you program an event timer to execute at the same time Monday through Friday of the current week. For example, if the event time is **7:32 PM MF**, and the time right now is 9 PM Tuesday, this event will first execute at 7:32 PM Wednesday (since 7:32 PM Tuesday has passed) and will continue at 7:32 PM on Thursday and Friday of this week only. (If the RPT option were in effect, it would continue executing EVERY Monday thru Friday at 7:32 PM.)

Second Week (2ND):

You can set the 2ND option with the $\boxed{2nd}$ toggle key. This means, "execute this event at the second occurrence of this day and time." For example, if the event is set for **7:32 PM TU 2ND**, it will happen a week from next Tuesday at 7:32 PM.

Repeating events (RPT):

You can set the RPT option with the $\boxed{P\text{SPACE}RPT}$ toggle key. This means "repeat this event indefinitely." If the event is **7:32 PM TU RPT**, it will occur EVERY Tuesday at 7:32 PM.

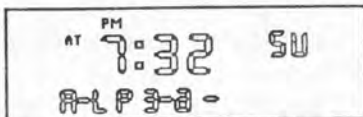
Combinations:


You can use the options in combination, as follows:

- **MF RPT:** execute every weekday, indefinitely.
- **MF 2ND:** execute Monday through Friday of the second week. For example, if today is Wednesday, it would start next Monday; but if today is Saturday, it would start a week from next Monday.
- **MF 2ND RPT:** start executing as in the above example, and execute every other week Monday through Friday.


Specifying the program to be executed

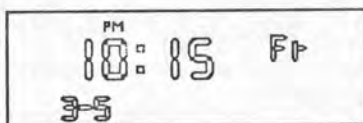
5. When the time and day for your timed event are correct, press \boxed{L} **DO**.
6. Specify the location of the program you want CORE to execute when the event timer goes off. Press $\boxed{P\text{SPACE}RPT}$, then the PAGE location, then the KEY location.



Note: If the PAGE of the program is the same as the current page you are in, you can specify the KEY only. If you make a mistake in entering the location, you cannot use the  key. You will have to re-enter the timer.

Note: You cannot teach CORE a command from another remote controller within an event timer sequence. You should already have captured the command at a particular location.



7. Press . The display will return to showing the current time and last key pressed.



The event timer is now stored in the event queue. CORE stores all event timers in the event queue in chronological order. You can review the event queue to remind yourself of what events are due to execute; you can also cancel an event that you have already put into the queue. See "Reviewing the Event Queue" in Chapter 4 for details.

Execution of the Event Timer

CORE will execute the program stored at the specified location at the specified time and day(s). In this example, the program stored at PAGE 3, KEY 5 will execute at 7:32 PM on Thursday.

When time for an event arrives, you will hear an alarm (three alert tones); then the specified program executes. If the display is on at the time the event happens, you will see the  symbol flashing. If the display is off, then the next time you wake up CORE with a keypress, the  symbol will be flashing to let you know an event occurred while the display was off. You can review the event queue (see Chapter 4) if you need to find out which alarm executed.

Note: You must make sure that your CORE is pointing at your equipment in preparation for any event timers which will be executing.

Including Event Timers in Program Sequences

In addition to setting **stand-alone event timers**, you may also include event timers within programs. Why would you want to do this? Here's an example. Suppose you want to set an alarm to wake yourself up in the morning by having your TV set go on at 7 AM. But you don't necessarily want to have the alarm go off every day; maybe you don't have a regular schedule, and some days you will sleep late.

By creating a program containing an event timer, you can accomplish two things:

- You don't have to go through the whole process every time you want to set the alarm. You can simply press a single key, or PAGE-KEY combination, which will cause the event timer to be set.
- You can decide the night before whether you want to set the event timer. If you don't want it set, you don't press the key to execute it.

When you include event timers in program sequences you follow the same general steps as when you set stand-alone event timers. Refer to "Syntax Guide" if necessary.

When event timers are included in program sequences, the keys you press to set the time are displayed as part of the program on the status line (rather than affecting the time display). For example, suppose your wake-up (TV on) program is stored on PAGE 8, KEY 1, and want to put your "alarm" program on PAGE 0, KEY 0.

First make sure Enable is on. Then select PAGE 0, KEY 0. Then press the following keys:

1. : This begins the program, as usual.
2. : This is the start of an absolute event timer.
3. : This means "7 o'clock", using the direct digit method.
4. : This means "day up" in the context of timer-setting. You do this since you want the timer to execute the **next day**. When you execute the program, CORE will automatically figure out what day the next day is. (Make sure you execute this program before midnight, or it will skip a day!)
5. : This means AM. It is represented in the display by the symbol **u**.
6. : This begins specifying which program the event should execute
7. : The location to execute (PAGE 8, KEY 1).
8. . This ends the event-program specification.
9. . This ends the entire program.

Note: if you are confused by the fact that there are two consecutive 's, think of it this way: every needs a corresponding . The at #1 corresponds to the at #9--they are the outermost bounds of the entire program. The at #6 corresponds to the at #8: these are the bounds of the program to be executed when the timer happens.

Note: keys 2-10 are the exact same keys you would have pressed if you had set a stand-alone alarm for 7 AM.

The program will look like this in the display: **Ⓛ 7 F u Ⓛ P 8 1 J J**

Hint: Use through rather than the key-pairs when you are setting an event timer within a program so you can see the time as you are setting it on the status line.

Relative Event Timers (Intervals)

Relative event timers, or intervals, are different from absolute event timers in that you don't specify the exact time something will happen, but how many hours, minutes or seconds from NOW it will happen. For example, after having been awakened by your TV set, you decide you'd rather sleep for another 5 minutes. You can activate a program (which you've cleverly programmed ahead of time) which turns the TV off, waits 5 minutes, and then turns it on again. And, of course, you can repeat your 5-minute snooze as many times as you like.

Intervals automatically operate a command or sequence of commands at a time relative to the present. For example, a five minute interval will be activated five minutes from when it is set. Intervals are also used to sound alarms and to insert pauses between commands within programs.

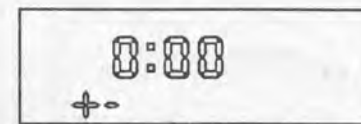
You have the option of setting **long** or **short intervals**. Short intervals wait from one second to 9 minutes and 59 seconds, using one second increments. Long intervals wait from 1 minute to 9 hours and 59 minutes, using one minute increments.

As with absolute timers, you can have a relative timer that is a stand-alone, or included within a program.

To set a stand-alone Interval of under ten minutes (0:00:01 to 0:09:59)

1. If **En** isn't displayed, press .
2. Press the key to initiate interval programming.

When you press , **En** disappears and the **+** symbol appears on the status line, as shown below. (**En** reappears when you complete the timer sequence.)



You'll notice that the the numbers 0:00 are displayed where the time normally is.

3. Select the minutes from through .
Press zero here if the interval you're setting is less than 60 seconds.
4. Select the tens of seconds from through .

Press zero here if the interval you're setting is less than 10 seconds.

5. Select the ones of seconds from through .

6. Press key.



This diagram illustrates an interval of 15 seconds. The flashing cursor (shown as solid here) indicates that only keystrokes (not capture codes) are acceptable.

7. Specify the location of the program you want CORE to execute at the end of the interval. (To sound an alarm tone only, select a location that contains no program.)

Note: If you make a mistake while setting an event timer, press to start over.

8. Press . This stand-alone interval is now stored in the event queue (where you can review it). The 15-second countdown starts from the time you press .

To set a stand-alone long interval of under ten hours (0:01 to 9:59):

Use the same procedure as for short intervals above, with one addition. After step 2, when the 0:00 is displayed, press either side of the

paired key. You'll see an **h** on the status line following the **+**:



Now continue as before with steps 3 through 8, keeping in mind that you are using hours and minutes instead of minutes and seconds. For example, 0:15 would mean fifteen minutes; 9:59 would mean nine hours and fifty-nine minutes.

Intervals Within Programs

Chapter 2 explained how to program CORE to transmit long sequences of IR commands with a single keypress, a very useful feature. The problem is that sometimes you don't want all your commands to execute in sequence; rather, you would like to wait some amount of time between commands.

The following examples will show you some useful things you can do with intervals, while describing how to program them. You can create your own programs using the same principles.

Record a TV program for 1 hour; stop for 1/2 hour; change to channel 4; record for 2 hours; and shut off all equipment.

Assume that your TV and VCR are already on and set to the channel you wish to record first, and that you have the following commands at these locations (taken from the CORE pre-stored commands):

- Location 9-1: VCR Record
- 9-2: VCR Stop
- E-a: VCR Power
- E-4: VCR Channel 4
- 7-a: TV Power
- 7-b: TV Mute

The following keypresses are needed to create this program (making sure Enable is on first):

1. : start of program.
2. : send VCR Record to begin recording.
3. : wait 1 hour, then send VCR Stop. Remember that the sent after means "long intervals."
4. : wait 30 minutes, then send VCR Channel 4 (assuming you are controlling the channel from the VCR).
5. : wait 3 seconds, then send VCR Record.

Note: This is an example of a delay which is sometimes necessary when sending two commands to the same equipment. If the second signal is sent too quickly, sometimes the equipment may not respond.

6. WAIT 2 DO P (PAGE) RPT E a OK : wait 2 hours, then send VCR Power (to turn off the VCR).
7. WAIT DO P (PAGE) RPT 1 a OK : wait zero seconds, then send TV Power (to turn off the TV). **Note:** When there are no numbers after the WAIT, the default interval is zero seconds.
8. OK : end of program.

Some Fine Points

In step 7, you might wonder why a zero second interval is required. Why couldn't you simply press P (PAGE) RPT 1 a and forget about the interval? To understand this, you need to understand what happens when a program containing an interval is executed.

Let's assume the above program was stored at location 0-7. When you press KEY 7 of PAGE 0, first step 2 is executed, sending a VCR Record command. All subsequent commands, which are in the form of intervals, are not executed at once but are put into the event queue to execute at the appropriate later times. If you have a command which is *not* an interval (such as P (PAGE) RPT 1 a), it does not go into the queue but is executed *immediately*. This is not what you want.

In summary, whenever there is any kind of event timer inside a program, all subsequent actions must be preceded by WAIT or RT. If they are not, they will execute immediately.

Suppose you had programmed the following on PAGE 1, KEY 1:

WAIT 0 0 1 DO 1 OK. This means "In 1 second, execute KEY 1". In 1 second, CORE will sound the 3-tone alarm & execute KEY 1, which puts another 1-second event into the queue, which executes in 1 second, and so on. The upshot is that you will get a constant stream of alarms, which will never cease. The only recourse is to press the Reset switch. After you've done that, make sure to delete the definition of location 1-1 so that it will not repeat the same process. You should also go through the event queue and delete all timers which may have inadvertently been set. (In an extreme case, you can press R and Reset simultaneously, which deletes all timers.)

Send a Mute command to your TV (when a commercial comes on); wait one minute and send Mute again so the sound comes back on.

- DO : start of program.
- P (PAGE) RPT 1 b : send TV mute.
- WAIT 1 DO b OK : wait one minute, then send mute again.
- OK : end of program.

Programming Daylight Savings Time

Now, suppose it's the end of May. You're going to be out of town for the next two weeks. You have CORE programmed to record several TV or radio shows at different times throughout the two-week period. The problem is that this coming Sunday, while you're gone, there's going to be a change from Standard Time to Daylight Time, which means the clock should be set forward an hour, otherwise all your event times will be off by an hour. But you won't be home, so you must tell CORE to do the job for you. Here's how you do it:

You already set up a program on PAGE 0, KEY 2 to set the time ahead one hour. The next step is to create an event timer to execute that program. Let's suppose today is Wednesday, and that the time change is this coming Sunday.

You would need to press the following keys:



Chapter 4: Editing

In this chapter you will learn to:




- Review your programs to see what you've programmed at any location.
- Change the contents of your programs.
- Copy a program from one location to another.
- Review the contents of the event queue to see which timed events are due to be executed, and delete any if desired.

Editing a Program




Editing is useful in the following cases:

- You've entered a fairly long sequence, or one with some event timers in it, and you want to make sure it came out correctly. If it didn't, you would like to change it.
- A program doesn't do what you think it's supposed to do. You want to check it to find out why.
- You'd like to make a minor change to a long program, such as changing an interval from 30 minutes to 1 hour, but leaving everything else the same.



The following steps are required when reviewing or making changes to one of your programs:



1. Press . The symbol  appears in the status line.
2. Select the location to be edited. This means , followed by the PAGE location, followed by the KEY location. Or you can specify only the KEY location if the current PAGE is the same as the program you want to edit.



Once you've entered the location, you'll see that the PAGE-KEY location replaces the time at the top of the display. The program stored at that location is displayed on the status line. The cursor (a dash which alternates with a character) will be located at the leftmost character.


Note: The  and  at the beginning and end of the program are not displayed. If the program contains more than 8 characters, you'll see the  symbol displayed to the right of the 8th character. This means that there are more characters to the right that won't fit on the display.

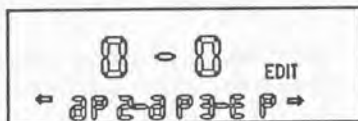
3. You'll see the **EDIT** symbol on the right-hand side of the display. If the **EDIT** is solid, it means Enable is off: you may only review programs without changing them. If **EDIT** is flashing, it means Enable is on: you will be able to modify your programs.





4. Use the right () and left () arrow keys to move through the program.

- Press  to move the cursor right one character. The cursor will move with each keypress until it reaches the end of the program or the eighth character. Then, if the  is displayed (indicating more program information), the display will scroll with the next keypress.

- Press  to move the cursor left one character. The cursor will move with each keypress until it reaches the second character of the program. Then, if the  is displayed (indicating more program information), the display will scroll with the next keypress. If the cursor

is at the first character of the program, pressing  again will cause a reject tone, unless Enable is on, in which case the cursor will move to the top of the display (see Copying a Program, step #8 below).






- When you are reviewing long programs (with Enable Off) you can use the   key-pair to move the cursor half a window at a time. In this **Quick Cursor Move** one keypress is equivalent to 4 presses of the arrow keys. The  key moves the cursor 4 characters to the right; the  key moves 4 characters to the left.

5. If you have finished reviewing the program, and you do not wish to make any changes, skip to step #9. If you want to change the program in any way, proceed with the following section.

Modifying a Program









To make any modifications in the program, Enable must be on (indicated by a flashing **EDIT**). You will hear a single reject tone if you try to modify a program with Enable off.

6. *To delete a character:*



Press  and the character to the LEFT of the cursor will disappear. If the  is displayed, the cursor will stay at the same position and all the characters to the left of the cursor will shift one position to the right. If the  is not displayed, the cursor will shift one character to the left.

7. *To insert one or more characters into the program:*

Press any key, and the corresponding character will be inserted to the left of the character that the cursor is on. The characters under and to the right of the cursor will shift one position to the right.

Note: The following keys perform editing functions and cannot be used to insert characters: , , , , , . The only problem this creates is that you can't use the  or  keys to insert a **repeat loop** into a program. If you wish to use repeat loops you have to enter the program from scratch.

8. *To copy the program to another location without altering the original:*

Press  until the cursor is blinking at the KEY location at the top of the display. Then enter the new KEY location, if any. Press  again and the cursor will blink alternately with the PAGE location. Now enter the new PAGE location, if any.

This results in the program being stored at the new location shown. This type of procedure is useful in the following situations:

- You want to move an IR code from one location to another, but you don't have your original remote handy, or you don't want to use it. You simply Edit the location with the IR code in it, change the location at the top of the display, and store the program. (You can then delete the program at the original location if desired.)
- You want to create a new program which is similar to one which you already have stored. Rather than enter the new program from scratch, you can Edit the old one, then store it at the new location. This is particularly helpful for very long programs.

This example shows the steps involved in duplicating the contents of location 2-0 at location 3-3:

[E]n [←]ENABLE [→]EDIT [P]PAGE [RPT] 2 0 [←] 3 [←] 3 [→]EDIT

When you press the second [→]EDIT, the complete program from location 2-0 is also stored at location 3-3.

- You may press [C]ANCEL at any time to exit Edit and leave the original program intact. When you are done reviewing or changing the program, press [→]EDIT. If Enable is off, the program will remain unchanged. If Enable is on, CORE will attempt to store the new, revised program in place of the old one.

Built-In Verification

After you've made your changes to the program, and you press [→]EDIT to store the new program, CORE checks to see whether the **syntax** of your program is correct; in other words, whether your program makes sense. If it is correct, you will hear an alert tone and the new version of the program will be stored. If you followed step #8 above, the program will be copied to the new location and the original program will be preserved. Otherwise the original program will be replaced by the new one.

If the program is incorrect, CORE sounds a reject tone when you press [→]EDIT. A blinking cursor appears at the position in the program which contains the first illegal character. If the cursor appears at the end of the program, it means that the program is not complete: more characters are required to have a valid program.

Recall that when you enter a program with the [L]DO key, if you enter an invalid key by mistake, you hear a reject tone. When in Edit mode, things work a little differently. You are allowed to insert and delete keys in any sequence. CORE doesn't check for illegal syntax until you are finished.

Also, CORE waits until you finish editing to convert **context-sensitive symbols**. Some keys display different symbols depending on the context of the program. For example, if [a-]AM is used within an event or interval timer to indicate AM, it appears as **u**. If it is used to represent a location, it appears as **♯**. When editing, if you insert [a-]AM, it will always appear as **♯** until you save the program. Then if you review it again, you will see the correct symbol. (The double function CORE keys [←]a-AM,

[b-]PM, [e-]2nd, [P]PAGE [RPT]—are represented by context-sensitive symbols). CORE also waits to insert a dash between the PAGE and KEY portions of a location.

Reviewing the Event Queue

As discussed in Chapter 3, the event queue contains a list of all event timers that have been programmed, either directly by you, or as a result of executing a program containing an event timer. With the following procedure, you can look through all the events in the queue. You cannot change any events, but you can delete events if Enable is on. If you wish to change an event, you will have to re-program it.

- Press [→]EDIT.
- Press [A]T to display the event queue. If you hear a reject tone, it means there are no events in the queue. Otherwise you will see a display like this:

```

  AT  PM
      2:00  SU
      0-3
  
```

If the **[(A)]** symbol is flashing, it means that the first event shown is the most recent event that executed. The time and location displayed correspond to the event timer which executed. Otherwise, the next event to execute will be first in the queue. The display shows the time and location of the event. For example, this display indicates that, at 2:00 PM on Sunday, the program at location 0-3 will execute. If the **[(A)]** were flashing, it would mean that the most recent past event that happened was Sunday at 2 PM.

```

  AT  PM
      2:00  (A) SU
      0-3
  
```

- Press [E]n [ENABLE] to turn Enable on or off. A blinking **AT** on the display means that Enable is on, and you will be allowed to delete timers. A solid **AT** means that Enable is off and only reviewing is allowed.
- Use the right arrow key ([→]) to move through the queue. The events will be displayed in their chronological order of execution. A reject tone will sound when you reach the end of the queue. Press the left arrow key ([←]) to move backward toward the beginning of the queue.

Some notes about the way timers are shown in the queue:

If **2nd** is illuminated, the event on the status line executes when that time and day occur (during the second week from the present.).

If **2nd** is flashing, the event on the status line executes at the next occurrence of that time and day, although the event was originally set as a "2nd-week" event.

If **MF** is illuminated, the event will execute at the time shown, every weekday for the current week.

If **RPT** is illuminated, the event will execute, then go back into the queue for the following week.

5. Press **DELETE** if you want to remove the event displayed. The display will then revert to the first event in the queue.
6. Press **OK** or **CANCEL** to exit the event queue.

Chapter 5: Modifying Captured Codes

This chapter contains instructions on procedures to modify various characteristics of the infrared codes that you have captured in CORE. Refer here if your commands don't seem to be operating as you expect. Refer to the Troubleshooting section in Chapter 6 for guidance in where to look.

Altering the Number of Sends

Most infrared-controlled equipment requires that commands be sent more than once in order to operate. In other words, if you briefly tap a button on your controller, it will send a signal but won't operate the equipment. You have to hold your finger on the button long enough for the code to be sent two or more times. The exact number of times required can vary.

Normally you don't have to worry about this--you simply press the button until the equipment responds. CORE, however, must be able to work in indirect or **unattended operation**--in effect pressing the keys itself. For example, suppose you capture an IR command on one of your CORE keys, say **PAGE 0, KEY E**. If you hold the **E** key down, the code will keep sending (and the **C** will stay on). You simply have to hold it down long enough for the equipment to respond.

However, suppose you have another program at **PAGE 0, KEY d**, which calls the program on **PAGE 0, KEY E**. If you press the **d** key, it will send the infrared signal a specified number of times (called the **number of sends**), then stop, even if you keep your finger on the key. (See "Notes on Indirect Access" in Chapter 2.)


When you capture an IR, CORE sets the number of sends automatically to 2. This is the number that works accurately with most equipment. However:

- some equipment (Mitsubishi, JVC, for example) may require the codes to be sent 3 or more times in order to work.
- some equipment responds after a single send. This may mean that when you press a key, it will perform the action twice, such as sending two Channel Up's, or turning the power on, then off. In this case the number of sends should be set to 1.


The following describes the method you can use to change the number of sends from 2 to any other number from 1 to 8.

1. Press **EDIT** to begin.
2. Select the PAGE-KEY combination to be edited.

The **EDIT** symbol on the display should be solid. If it's blinking, press

 **ENABLE** to turn Enable off.


3. Use the arrow keys to move the cursor to the right of the **C** to be altered.

4. Press .


C will be replaced by **2**, the number of times CORE normally sends the captured code.

5. Press a key from  to  to replace the **2**.

(When you need to increase the number of sends, try 3 first; then if that doesn't work, try 4, etc.)

6. Press  **ENABLE**.

Turning Enable back on is the only way to store the change you have made to the code. (If you are altering more than one code in a session, be sure to turn Enable off and back on again for each code replacement.)

7. Press  **EDIT** to finish the session. You are then ready to send the code to test whether your change has worked.



Changing the Output Frequency of a Code


If CORE doesn't work over at least the same range, distance, or angle as the source remote controller, you can try changing the output frequency of the code. Fine-tuning CORE's output frequency lets you set it up to send codes at least as well as the source remote controller.

CORE lets you alter the number of microseconds in each frequency cycle in the range of 17 to 40 microseconds. When you use this technique, you have to rely on trial and error: change the frequency by one or two microseconds, try the code to see if it works better, change the frequency by one or two more microseconds, try the code again, and so on.

Each code usually has two or three settings that work best. For optimum results, find the *middle* setting in the range of effective control of the receiving equipment, and set CORE at the middle number in that range.

The method for changing the output frequency is as follows:




1. Press  **EDIT** to begin.
2. Select the code location to be edited. The **EDIT** symbol on the display should be solid. If it's blinking, press  **ENABLE** to turn Enable off.
3. Use the arrow keys to move the cursor to the right of the **C** to be altered.

4. Press .


The time display is replaced by **P**: followed by a number between 17 and 40 that represents the current **period**, or number of microseconds in each code frequency cycle. The period is the inverse of frequency: the larger the period, the lower the frequency, and vice versa.

For example, you might see **P:24**.

Note: If you see **P:F**, you can't modify the frequency of the code you have chosen. In this case, your only recourse will be to try capturing the signal again.


5. Use the  key-pair to alter the number of microseconds. Press  to lower the period (raise the frequency); press  to raise the period (lower the frequency).

You'll need to use trial and error. In general, only alter the code by one or two microseconds at a time. Then proceed to steps 6 and 7, and try the altered code out.

6. Press  ENABLE .

Turning Enable back on is the only way to store the change you have made to the code.

If you are altering more than one code in a session, be sure to turn Enable off and back on again for each code replacement.




7. Press  EDIT to finish the session.

Altering the Sequence Send Rate


One of the nice features of CORE is the ability to send several IR codes in sequence, with a single keypress. When CORE sends two consecutive signals to *the same equipment*, it's possible that the second signal will not be recognized because there is not a long enough pause between signals.


CORE normally uses a 2/10 second delay between each code sent. This works for most equipment. However, in some cases a longer delay is required. The following method is used to increase the delay for a given PAGE-KEY location to 1 second.


Note: When you change the send rate to slow for a given location, CORE automatically inserts delays of 1 second between all IR codes sent when the location is executed. The send rate does not affect repeat loops, however.

1. Press  EDIT to begin.
2. Select the code location to be edited.
The EDIT symbol on the display should be solid. If it's blinking, press  ENABLE to turn Enable off.
3. Press  WAIT .

An **SL** appears above EDIT on the display. The **SL** means that each successive command in the sequence will be delayed one second when you operate the sequence.

Each press of  WAIT turns **SL** alternately on or off.

4. Press  EDIT to complete the change.

Note: You do not have to press  ENABLE to save the change.

In the event that a one-second delay between codes is still not enough to get the equipment to respond, you can add short intervals within the program. Refer to Chapter 3 for information on using intervals inside a program.

Problems with Repeaters

Suppose you captured a command which is a repeater on your source controller: that is, it keeps sending as long as you keep your finger on the button. An example would be a Channel Up command which continues to increase the channel. You've captured this command at a particular PAGE-KEY location in CORE. When you press the key, the equipment responds once or twice, then stops, but the **C** stays on the display.

First, be sure the definition of the location is a single **C** (refer to Chapter 4 on Editing if you forget how to review a program). If the definition is anything but a single **C**, the command will not repeat. If it is a single **C**, try re-capturing the command.

If it still doesn't work, your controller may fall into a category which requires the following adjustment:

1. Make sure Enable is OFF.
2. Press **EDIT**, followed by the location of the command. You should see a flashing **C** in the edit line.
3. Move the cursor one space to the right of the **C**.
4. Press the **a** key. The **C** is replaced with **u**.
5. Press the **b-PM** key. The **u** is replaced by **n**.
6. Press **ENABLE**. EDIT starts to flash.
7. Press **EDIT**. You'll hear an alert tone, meaning the program has been stored.
8. Try executing the command now; it should repeat.

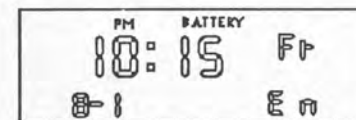
Chapter 6: Maintenance and Troubleshooting

Core Maintenance

Batteries

CORE uses four AAA batteries. For best results, you should use alkaline batteries only.

When the batteries get low, the **BATTERY** indicator shows at the top of the display. When this occurs, you should replace the batteries as soon as possible, preferably within a week.




CORE has a Lithium backup battery which protects the memory when batteries are low or removed. Still, when changing batteries you should never leave the batteries out for more than a few minutes if possible. **You should not remove the batteries for extended periods of time. Above all, when storing or shipping, always make sure that CORE contains batteries.** (This is different than storing other devices that use batteries.) Follow the instructions below for storing CORE when you know you won't be using it for a time.


The battery cover is on the back side of the CORE, opposite the programming key cover. See fold-out drawing at the back of the manual.

To Change Batteries:

1. Turn CORE so you're looking at the back side.
2. Push down on the lower ribbed area and slide the battery cover off.
3. Remove the old batteries. Insert new ones as soon as possible.
4. Replace the battery cover by sliding it back on until the center tab clicks into place.

5. Press the  key twice to wake up CORE. If the display does not come on, refer to the section below on "Resetting CORE".

To Store CORE for an Extended Period:

1. Turn CORE so you're looking at the back side.
2. Install fresh batteries.
3. Find the "Master Module" label near the top.
4. Pull up on the ribbed area to remove the module.
5. Press  and let the display time out.


To Clean CORE:

Wipe CORE with a damp sponge. Use a mild dish soap if necessary, but do not immerse.


Resetting CORE



If your CORE keyboard does not respond to any keypresses, or you have any kind of failure, follow these steps to test CORE:

1. Wait for the display to time-out.


2. Press the  key twice. The display should come on, showing the time and current location. If the display does not come on, proceed to step 4.


3. Press some keys to test them. If they respond correctly, return to what you were doing. If not, continue with step 4.


4. Push the end of a paper clip into the reset hole between the  and


 keys. Pressing Reset restores the current location on the status line to , and displays the current time. (Occasionally you may have to adjust the time by using the clock-setting procedures.)

If some other display appears, go on to step 6.


5. Press .


- If you do not hear any tones, your CORE should be OK and ready to use. All of your programs and alarms should be intact.
- If you hear six low tones sound, even though the  appears on your display, you will have perform a reset that clears the event queue but retains your key definitions.

6. Press on the reset hole while holding down the  key.

This resets CORE and clears the event queue. The display should show . The contents of the event queue will be lost. If you hear no tones, all is well and you can return to using CORE. If you hear 6 low tones, proceed with step 7.

7. As a last resort, press on the reset hole while holding down the

 key.

All captured remote controller codes, programmed keys and alarms will be lost, but the CORE should now be functional. The display should show  and no tones should sound.

8. If CORE still does not appear to be responding properly, call CL 9 Customer Service at (800) 544-4259.


Troubleshooting

The following are some common problems that may arise while using CORE. They refer you to the section in the Reference Guide to look for help.

Problem	Chapter: Section
Can't capture IR signal: get reject tone or no tone	1: Teaching your CORE a Remote Controller Function
Captures but won't operate equipment; or won't operate consistently	5: Changing the Output Frequency of a Code
Command sends but won't repeat when key is held down	1: Repeating & Non-Repeating Commands 5: Problems with Repeaters
Display won't come on, or Keys won't respond	6: Resetting CORE
Unit won't respond when IR is accessed indirectly; or CORE sends signal twice	5: Altering the Number of Sends
Timers didn't happen when I expected	4: Reviewing the Event Queue
Signals being sent too quickly for equipment to respond	3: Intervals Within Programs 5: Altering the Sequence Send Rate
Part of sequence worked OK; part didn't	2: Single-Step Verification of Long Programs
I want to stop an executing program	1: Canceling Execution of a Program
The BATTERY indicator is on	6: Batteries
CORE buzzes 6 times	6: CORE Messages and Tones

CORE Messages and Tones

CORE sounds a number of different tones to communicate with you:





- **A single alert tone (high frequency chime)** signals CORE's acceptance of an infrared signal, a program, or an editing session.
- **Two alert tones** sound when a timer is pending (i.e. it's time for the timer to execute, but something else is happening).
- **Three alert tones** indicate that a timer is executing.
- **A single reject tone (short, low frequency buzz)** sounds when there is an unacceptable keypress, if a remote controller code hasn't been successfully captured, or if a program is aborted by pressing the  **CANCEL** key.
- **Two reject tones** indicate an out-of-memory condition. It occurs if:
 1. You try to put too many characters in a program.
 2. You try to capture too many infrared codes at a single location (six is usually the maximum).
 3. You fill up CORE's memory.
 4. You execute a recursive program (see Chapter 2 for details).
- **Six reject tones** indicate something is wrong with the area of CORE memory that stores your programs and timers. Refer to the section on "Resetting CORE" in this chapter.

Appendix A: CORE DISPLAY SYMBOLS

This chart shows the symbols that can appear on the CORE display, along with the keypress(es) which produce them, and a brief description of their function. Keep in mind that many of these symbols are context-sensitive, i.e. what is displayed depends upon what you're doing with CORE at the time.

Symbol	Keypress	Description
u		AM
a		direct access PAGE a-
n		PM
b		direct access PAGE b-
		second week event
c		direct access PAGE c-
z		repeating event
p		PAGE
h		hour down; hour up
a, b		locations a, b
		minute down; minute up
c, d		locations c, d

Symbol	Keypress	Description
MF, SU, MD, etc.		day down; day up
E, F		locations E, F
0 - 9		locations 0 - 9
=		set time
⌘		event timer
~		edit
none		cancel
↵		end loop
none		move cursor to left
←	none	more data to the left
+		relative event timer
SL		slow send
none		delete
↻		begin loop
none		move cursor to right
→	none	more data to the right
En		enable

Symbol	Keypress	Description
o	 (KEY)	key
L	 DO	do (learn)
J	 OK	OK
((A))	none	event timer alarm
[source controller	capture (repeating key)
]	source controller	capture (one-shot)
← →	 ←	single-step verification







Appendix B: Syntax Guide

This section is a quick guide to the CORE programming language. The proper syntax (rules) for all CORE programming functions is described. For each function, the guide shows which keys must be pressed and which are optional.


* denotes keys which are optional. This means that the key or keys specified may or may not be pressed. Which key you press depends on exactly what you want to do. Keys without the * must be pressed to form a valid syntax statement.

Location- In the syntax structures that follow, a reference to "Location" can refer to one or more of the following ways to represent a location. The number(s) in brackets tell which representations are legal in the given context.

For example: "Location (1,2)" means that either #1 or #2 can be used. In the following list, "PAGE location" and "KEY location" are represented by one of the location keys 0 to 9 and a to F.

-  , PAGE location, KEY location.
- KEY location.
-  , KEY location.
- One of the three Direct Page Keys;  ,  , or  .
-  , PAGE location.

Please refer to the chapters of this manual for complete descriptions of the keys and their functions.

Note:  is allowed at any point in any of the functions.

Moving to a location without executing its contents:

- *1. Location (5)
2. Location (3)

Programming a PAGE-KEY location

1. Location (1,2,3,4)
2. - on
- 3.
- *4. A Capture, C
- *5. Location (1,2,3,4,5)
- *6. - see Redefining a Location
- *7. - see Repeat Loop
- *8. - see Absolute Event Timers
- *9. - see Relative Event Timers
- *10. - see Setting the Time
- 11.

Redefining a Location (within a program)

- *1. Location (5)
2. Location (3,4)
- 3.
4. Any legal program
- 5.

Repeat Loop: (within a program)

- 1.
2. Any number from 1 to 99
3. Location (1)
- 4.

Single step verification of programs:

- 1.
2. Location (1,2,4)
- *3.
- *4.
- *5.

- Setting the time (direct or within a program)

Scanning method:

- 1.
- *2.

a	HR	b	,	c	MIN	d	,	E	DAY	F	,	a-	AM	,	b-	PM
---	----	---	---	---	-----	---	---	---	-----	---	---	----	----	---	----	----
- 3.

Direct digit method:

- 1.
- *2.

1	-	9
---	---	---

 1st digit
- *3.

0	-	9
---	---	---

 2nd digit
- *4.

0	-	9
---	---	---

 3rd digit
- *5.

0	-	9
---	---	---

 4th digit
- *6.

E	DAY	F	,	a-	AM	,	b-	PM
---	-----	---	---	----	----	---	----	----
- 7.

Note: The methods can be combined.

Absolute event timer (direct or within a program)

Scanning method:

1. RT
- *2. hr , min , DAY F , AM , PM , PAGE RPT , Snd
3. DO
4. Location (1,2,4)
5. OK

Direct digit method:

1. RT
- *2. - 1st digit
- *3. - 2nd digit
- *4. - 3rd digit
- *5. - 4th digit
- *6. DAY F , AM , PM , PAGE RPT , Snd
7. DO
8. Location (1,2,4)
9. OK

Relative event timer (interval) (direct or within a program)

1. WRT
- *2. hr b
- *3. - 1st digit
- *4. - 2nd digit
- *5. - 3rd digit
6. DO
7. Location (1,2,4)
8. OK

Reviewing the event queue:

1. EDIT
2. RT
- *3. ,
4. DELETE (only if ENABLE on)
5. OK

Editing a Program








Edit (modify) a program:

1. EDIT
2. Location (1,2,3,4)
- *3. ,
- *4. DELETE (only if ENABLE on)
5. Any other key (only if ENABLE on)
6. EDIT











Duplicate a program to another key:

1. EDIT
2. Location (1,2,3,4)
- *3. ENABLE - on
- *4.
- *5. - , : F - to change KEY
- *6.
- *7. - , : F , AM , PM , Snd - to change PAGE
8. EDIT





Changing the number of times an IR code is sent:

1.  EDIT
2. Location (1,2,3,4)
- *3.  ENABLE - off
4. ,  (to the immediate right of the captured code, C)
5. 
6. Any number from 1 to 8
7.  ENABLE - on
8.  EDIT

Changing the frequency of a captured code:

1.  EDIT
2. Location (1,2,3,4)
- *3.  ENABLE - off
4. ,  - to immediate right of captured code, C
5. 
6.  C  MM  D
- *7.  ENABLE - on
8.  EDIT

Changing the sequence send rate to slow:

1.  EDIT
2. Location (1,2,3,4)
- *3.  ENABLE - off
4.  MRT
5.  EDIT

Appendix C: Glossary





alarm three alert tones indicating a timed event is about to occur.

alert tone A high-frequency chime. Refer to Chapter 6 for details about when you will hear this.

alternate action key A toggle key that switches back and forth between two states with each press; such as Power On/Off.

button A button or key on a source controller (i.e. any controller *except* CORE).




capture To teach CORE a command code from a source remote controller.

context-sensitive symbol In Edit mode, a display character representing a double function key (such as , , , ) that may be interpreted and displayed differently after completion than when entered.

current location The last PAGE-KEY location executed.

cursor A pointer represented by a hyphen which indicates program position.

direct-digit method Method of setting the clock using the numeric keys. Resets the seconds count to :00.

direct-access PAGE keys The , , and  keys.





display The liquid crystal window which shows the time and status .

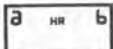
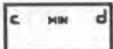
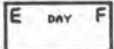
event queue Chronological listing of event timers awaiting execution.

event timer Alarm that automatically operates a stored CORE command or sequence of commands at the specified time(s).

interval Relative event timer setting; can be used to insert pauses within programs.

key Any of the physical keys on the CORE keyboard.

(location) KEY Any of the 16 keys,  through  and  through , used to represent the two-part storage location address.

key pairs The rocker keys , , and .

location Storage space in the CORE programmable memory. Each location is referenced by PAGE and KEY.

Master Module The removable head on the upper back part of CORE that contains the infrared transmitter and associated circuitry.

null program An empty program; a location without any contents.

number of sends the number of times CORE sends an infrared signal when a key is pressed.

one-shot An infrared code that transmits once, then stops.

one-step access A single keypress used to operate programs stored on the direct access PAGE keys.

operate To execute a program stored in CORE.

operating key Any of 20 keys used to operate the CORE. See *CORE Parts: An illustrated guide*.

PAGE-KEY location The place in CORE memory where a program is stored.

period The number of microseconds (between 17 and 40 in CORE) in each remote controller code cycle.




program A command or sequence of actions assigned to a specific location on your CORE. Programs can include remote controller codes, timed events, loops and other program definitions. Each program contains a maximum of 62 keystrokes.


programming keys Any one of 12 keys used to program and edit the CORE. See *CORE Parts: An illustrated guide*.


recursion When a program calls itself putting the processor in an infinite loop.

reject tone A short low-frequency buzz. See Chapter 6 for details on when this occurs.

repeat loop A section of program that is executed repeatedly a specified number of times.

reset To restart the microprocessor by pressing the switch located in the hole between  and . Reset  clears the event queue. Reset

 clears the event queue and key definitions.

reverse capture The display symbol  indicating that the source remote controller only sends its command one or two times per keypress.

scanning method Method of setting the clock using the key-pairs. Preserves the current seconds count.

sequence A program that combines two or more operations at a single location.

sequence send rate The amount of time between each code send. CORE normally uses a .2 second delay, but this can be changed to a 1.0 second delay indicated by **SL** on the display.

single-step verification Procedure for checking out program sequences by executing each program element in succession.

source remote controller One of your remote controllers used to teach CORE the infrared command code required to operate your component(s).


stand-alone event timer An event time that is not part of a program.

status line Eight-digit display at the bottom half of the LCD window.

storage location One of 259 CORE memory spaces where programs are kept.




syntax Rules for programming. See "CORE Syntax Guide".

teach To learn a sequence of actions or a remote controller command.

three-step access A three keypress sequence:  + PAGE location + KEY location; used to operate a program stored at that location.

time display Six character line in the upper half of the display window.

time-out When the CORE display goes off after a few minutes of non-use to conserve the battery. Any programming operation in progress will be canceled.

two-step access Operating a program stored on the  ,  , or  PAGES by pressing the corresponding direct PAGE access key and the KEY location.

unattended operation Automatic execution of CORE through the use of timed events.

verification Ensuring that remote controller commands captured on CORE perform the intended operation on your component.

wake-up To turn on the CORE display after a time-out by pressing any key.

Index

A

a- key *see* direct-access PAGE keys
alarm 17,38,47,49-50,53
alert tone 10,13,17,22,25,47,58,60,73
alternate-action key 35-37
AM /PM 4,41,58
arrow keys 33,38-39,56,59
AT key 15,16,43,44,48,53,59
AT symbol 15-17,44,59

B

b- key *see* direct-access PAGE keys
batteries 67-69
bell symbol 17,47,59

C

c- key 62 *see also* direct-access PAGE keys
CANCEL key 11,17,26,32,39,41,50,57,
58,60,69,73
capture *see also* infrared commands
reversed 22,23,25
symbol 5,10,22,24-25,61-66
Channel Up/Down command 6,20
clock-setting 3,40-42
context-sensitive symbols 58
copying a program 57
CORE Parts back page
cursor
blinking 28,31,50,55-58
quick move 56
unblinking 9,15,22,28
Customer Service iv,71

D

day, changing the 4,41,45,48
Daylight Savings Time change 54
deleting
a character 31,57
a program 27
an event timer 16,53,59
DELETE key 13,16,31,57,60
direct-access PAGE keys 1,18,20,21
direct-digit method 3,40,44

display (LCD) 1
arrow symbols 38,55,56
symbols 74
time-out 1,17,69
wake-up 1,17,47,68
DO key 9,12,16,22,28,45,48
duplicate a program 57

E

EDIT key 16,22,55-58,59,61-66
editing a program 55-58
copying a program 57
modifying 57-58
reviewing 55-56
editing keys 57
editing event queue *see* event queue
EDIT symbol 56
ENABLE key *see* Enable
Enable 9,15,22,44,47,56,59,62-66
event queue 50
delete from 16,53
review 16,59-60
event timers
absolute 15,43-48
execution of 38,47
relative (intervals) 43,49-54
stand-alone 15,47
within program 47,51-54
examining CORE 1
executing a program 5,8

F

failure 70
frequency *see* output frequency

G

Glossary 83

H

hours, changing 41,50

I

indirect access 29
infrared (IR) commands 7,9
capturing 9-11,22-26
modifying 61-66
multiple per key 25
inserting a key (editing) 57-58
intervals *see* event timers, relative

J

Journal 5,7,23

K

(KEY) key 6,12,21,36
KEY location 4,18
key pairs 20

L

LEFT ARROW key *see* arrow keys
location *see* PAGE-KEY location
long interval 49-50
loop *see* repeat loop

M

maintenance 67-69
 cleaning CORE 69
Master Module 69
messages & tones 73
MF (Monday thru Friday) 45,60
microseconds 63,64
minutes, changing 15,41,49,50
modifying captured codes
 see infrared commands
modifying a program *see* editing
Mute commands 20,34,51,53

N

non-repeating commands 23-24,29
null program 27
number of sends 27,61
 altering 61

O

OK key 4,10,13,17,23,28,41,46,48,60
one-shot commands 23,25
operating CORE 5
operating keys 2,6
organization of CORE storage 19
organizing your commands 20
output frequency, changing 63

P

(PAGE) key 1,5,6,18,21,36,63
PAGE-KEY location 6-8,18,21

changing 6,24,30,31
choosing a 8,21
current 1,4,19
PAGE location 4-6,18
 in a program 31
period *see* output frequency
Play command 12
Power commands 5,6,10,12,13,
 30,35,51-52
pre-stored commands 5,20
program
 definition 18
 deleting a 27
 elements of 18,28,38
 long 31
 sequences 12
programming keys 1,2

Q

quick cursor move 56

R

Record command 51
recursion 32
re-defining a location 35-37
reject tone 10,16,22,58
repeat loops 33-34,38,57,65
repeating commands 24,25,29,66
resetting CORE 2,70-71
reverse capture symbol *see* capture
RIGHT ARROW key *see* arrow keys
RPT (repeating event) 45,60

S

scanning method 3,4,41
2ND (second-week event) 45,60
seconds count
 time-setting 42
 event timer 44,49-50
sequences 12-14,25,28-30,47
sequence send rate, altering 65
SET key 3,40-42
short interval 49-50
single-step verification 38-39
SL (slow send) 65
source remote controller 8-10
status line 1,18
Stop commands 51
stop program execution *See* CANCEL key
storing CORE 69
storing commands 7-11
Syntax Guide 77

syntax verification 58

T

teaching commands to CORE 7,9,22
television (TV) 5,12,30,35,47,49,51-53
time-out *see* display
time-setting
 setting the clock 3,40-42
 within a program 42
timer *see* event timers
toggle key *see* alternate-action
tones 73
troubleshooting 73

U

unattended operation 15,61
up-down method *see* scanning

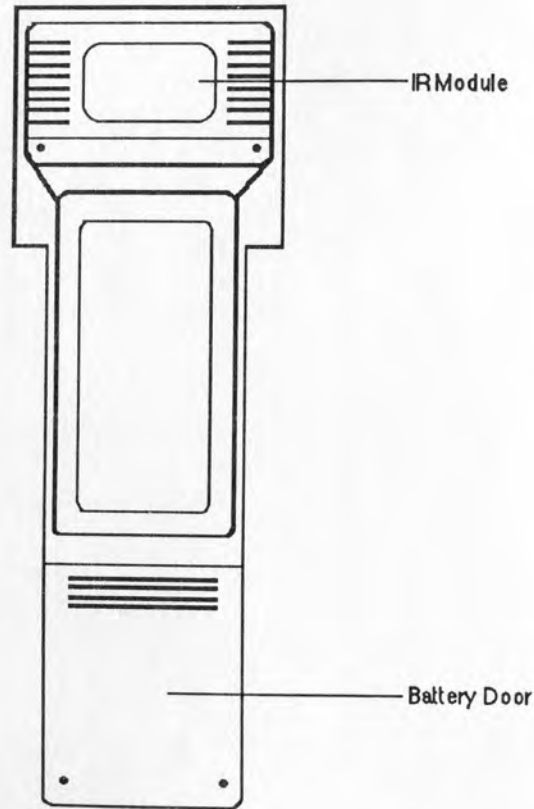
V

VCR 6,12,30,51-53
verify
 command 14,24
 single-step 38-39
volume adjustment example 34
Volume Up/Down commands 20,34

W

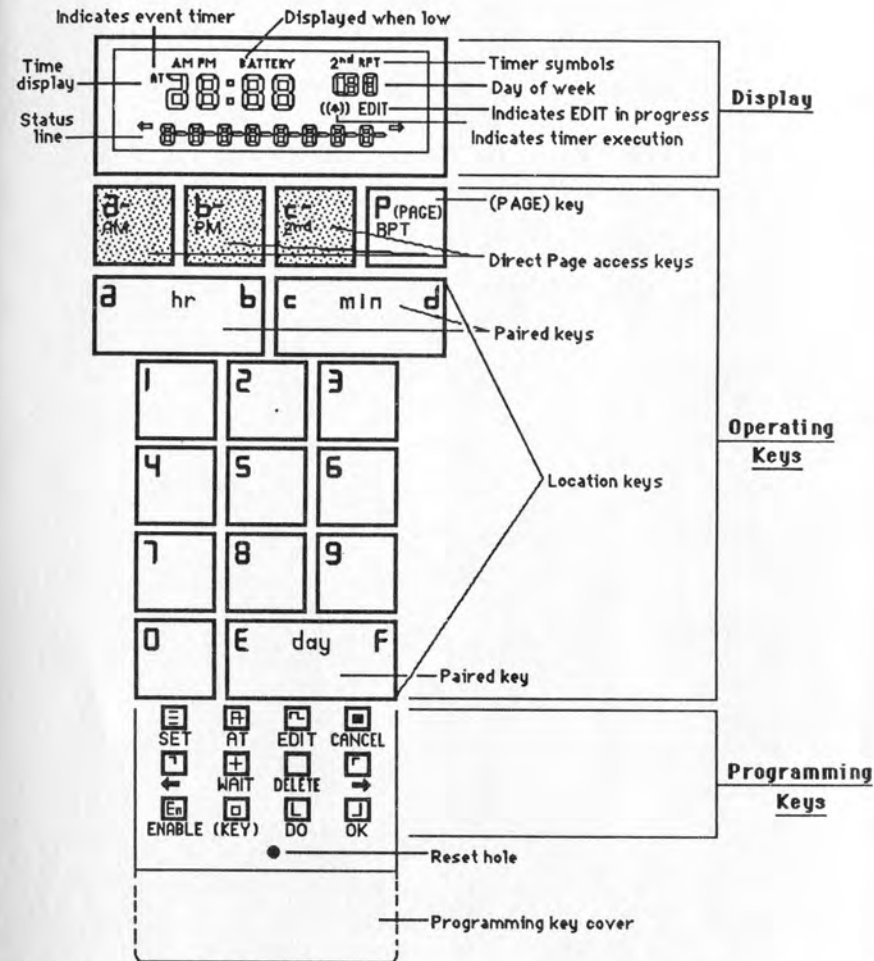
WAIT key 43,49-53,65
wake-up *see* display

Rear View of CORE



CAUTION: You should not remove the batteries for extended periods of time. When storing CORE or when shipping, always make sure that CORE has batteries. Please see "CORE Maintenance" in chapter 6.

CORE Parts: An Illustrated Guide



The CORE **display** shows the current time and, depending on CORE's state of operation, may also show a program on the status line, and illumine symbols for the day of week, timers, EDIT, ENABLE, or BATTERY low warning.

The **operating keys** let you select and operate any commands or command sequences stored at a PAGE-KEY location.

The **programming keys** let you program CORE, review and modify programs, set the clock, program unattended operation or event timers, and review and cancel timers.