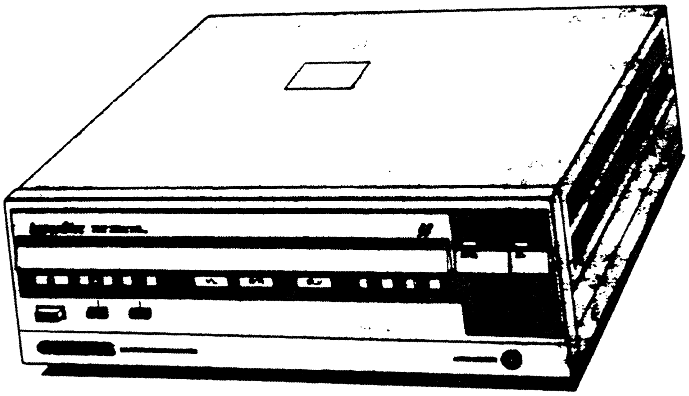
PIONEER

PIONEER VIDEO, INC.

LASERDISC PLAYER
LD-V6000 VERSION 2.0
USER'S MANUAL

(PRELIMINARY)



April 1984

TP109 Version 2.0

NOTE TO USERS:

The technical information contained in this manual is based on information available at the time of publication, and is subject to change. Although every reasonable effort has been made to include accurate information, the statements in this document are not warranties. Pioneer Video, Inc. makes no warranty or claims as to the accuracy, completeness, or fitness of any particular purpose of the technical information provided.

Throughout this manual, NOTES often reflect details of this version of the player which may not be available, or which might be different on future versions. They are included to aid understanding, but should not be depended upon in designing applications.

TABLE OF CONTENTS

TABLE OF CONTENTS
(continued)

TABLE OF CONTENTS
(continued)

TABLE OF CONTENTS
(continued)

CHAPTER I.

INTRODUCTION

1. SCOPE

This manual provides information to assist programmers in the design and development of application programs for the Pioneer Optical Laser Videodisc Player, Model LD-V6000, marketed by Pioneer Video, Inc., Montvale, New Jersey, and referred to in this manual as the "videodisc player."

The videodisc player can be used as a standalone system which, when connected to a television set or television monitor, makes a self-contained interactive video retrieval and display system. The videodisc player may also be connected to and controlled by a remote computer. Chapter III presents general interface and control information for users who may wish to attach the player to a computer. More detailed information, and assistance with specific computer-controlled applications, may be obtained from Pioneer Video, Inc.

2. VIDEODISC PLAYER OPERATION

The videodisc player is a highly flexible, user-programmable playback system that employs a laser to read television program material from a rotating videodisc. The videodisc player employs a low-power laser to read the data on the videodisc. Playback is achieved by directing a highly focussed laser beam onto the disc, recovering the reflected optical signal, and supplying it to a photodiode. The photodiode converts the optical data to electrical signals, which are further processed and ultimately converted to NTSC compatible video and VHF signals. These signals can drive either a television monitor or a conventional television receiver. Two channels of audio are separately available for connection to an optional user-supplied stereophonic sound system. The combined audio signals are also available as part of the RF modulated video signal, for use with a standard NTSC television receiver.

An internal microcomputer controls all phases of videodisc player operation, processing both external and internally stored command functions and internally-generated control and status signals. The integration of a microcomputer into the videodisc player design makes possible the many play, search, and display functions of which the player is capable.

Because the videodisc player is programmable, the exact sequence and display of information presented to the viewer can be predetermined by the program designer. Audio-visual messages can be developed for a wide variety of applications and tailored under program control. In addition, programs can be constructed which allow viewer interaction with the displayed material. By entering numerical responses from a Remote Control Unit (RCU), the viewer may select particular topics for display, respond to questions and receive information appropriate to his response. In general, the controlling program may shape the presentation of the audio-visual messages to the user's unique requirements. Properly formatted, the program-controlled videodisc brings outstanding flexibility to the presentation of audio-visual messages in industry, business, education, and entertainment.

3. VIDEODISC

The videodisc is the storage medium for the video, audio, and digital information processed by the videodisc player. It is a molded plastic disc with a 30 centimeter diameter, and with a capacity of up to 54,000 tracks per side. The tracks are arranged on the disc in a continuous spiral, with information beginning at the inner circumference of the disc and proceeding to the outer.

Data on the videodisc exists in the form of microscopic depressions ("pits") in the inner surface of a disc side. This surface is coated with a thin aluminum layer which, in turn, is covered with a thin protective coating. The reflective surfaces of two disc sides are bonded together, back to back, to form a two-sided disc with the recorded information sandwiched inside. The videodisc player's laser beam passes through the transparent outer plastic and reads the recorded information from the raised reflective impressions ("bumps") within. The outer plastic layer protects the reflective layer from mechanical damage.

Videodiscs may be recorded in either of two formats - Constant Angular Velocity (CAV) or Constant Linear Velocity (CLV).

A. CAV Format

A videodisc formatted for CAV is designed to rotate at a constant 1800 rpm. This format provides up to 30 minutes of television program material per side. Each of the 54,000 tracks contains one complete television frame. A series of frames can be "played" in forward or reverse direction, at slow, normal, or fast speed. Individual frames may also be displayed step by step in either direction, in freeze-frame mode.

Each frame is logically identified by a unique frame number. Individual frames can thus be accessed by the player's internal logic and read by the laser. This capability makes possible many of the user-programmable features of the player, such as search for a specified frame and play of selected segments.

B. CLV Format - Extended Play Disc

A CLV formatted disc is designed to be read at a linear rate that is constant over the entire disc. To achieve this, the angular rotation rate of the disc is varied during playback from 1800 rpm at the innermost track to 600 rpm at the outermost. The number of frames per track varies, increasing with circumference. Although this format provides up to 60 minutes of television program material per side, discrete frames cannot be displayed individually. As a consequence, the CAV-type program control features are not available with a CLV disc on this version of the player. Thus, when the player is playing a CLV disc, only the following commands can be executed by the player:

REJECT
PLAY
SCAN FWD
SCAN REV
AUDIO1
AUDIO2
Axx
CX
DISPLAY
VON
VOFF

Enable or Disable Character Generator
Enable or Disable Binary Output
Transmit Frame Number (Time Code)
Transmit Status

Note: At the current time, external sync lock is not guaranteed with CLV play discs.

For the remainder of this manual, all references to videodiscs assume that the discs are in the CAV format, unless specified otherwise.

4. INTERACTIVE PROGRAMS

The player's microcomputers, in addition to exercising direct control of the videodisc player's operations, provide the user with one kilobyte (1024 bytes) of programmable Random Access Memory (RAM). This memory space is available for the storage of user-designed programs to control player operation. Execution of programs stored in RAM is done by an interpreter resident in the Erasable Programmable Read Only Memory (EPROM). There is no provision to access the microcomputer instruction set directly.

User programs may be loaded into RAM in one of three ways: from programs coded on the videodisc, manually from a Remote Control Unit, or under the control of an attached host computer.

A. Program Dumps

In addition to audio and visual information, a videodisc may be encoded with one or more program "dumps." This is a computer program of 1022 bytes that is written on Audio Channel 2 during the disc mastering process. These program dumps may be loaded into the player's RAM from the videodisc. There can be multiple program dumps on the same disc. Under user program control, each may overwrite 1022 bytes of the 1024 byte RAM.

B. Remote Control Unit

Computer programs may be written into RAM through use of the optional Remote Control Unit (RCU). The RCU duplicates the PLAY button on the front panel of the player, and has many additional keys for controlling the player and entering program instructions and data. The RCU permits a programmer to enter and debug programs prior to their becoming finalized as program dumps on the videodisc. Users may also modify existing programs and tailor them to their unique requirements.

C. Attached Host Computer

The External Computer Interface (ECI) is an RS232 connector on the rear of the videodisc player, which permits interconnection of the player and a host computer. All aspects of videodisc player operation, including the loading of user programs into microcomputer RAM, may be performed via the ECI. (Interface information is contained in Chapter III of this manual.)

5. OPERATING MODES

The videodisc player may be operated in any of three modes: Manual, Automatic, or Programming. Each mode is briefly described below.

A. Manual Mode

Manual Mode is initiated by inserting a disc, closing the drawer, and giving the PLAY command.* Assuming that the videodisc does not contain a program dump (or that the Audio 2 channel is OFF, which instructs the player to bypass looking for the initial program dump), the disc may be played from beginning to end (or up to the first Picture Stop, if any exist on the disc) without any further action by the viewer. However, the presentation of the material may be changed at any time through the use of the ECI or RCU to direct the player. Host computer control frequently uses this mode to control the player. (Refer to the LD-V6000 Operatoring Instructions for a description of Manual Mode operation.)

Manual Mode terminates under the following conditions:

- o Videodisc Reject. The REJECT command causes videodisc rotation to slow to a stop, and the player to return to the PARK position.
- o Automatic Mode Select. Upon receiving the RUN command, the player enters Automatic Mode and begins executing the application program stored in RAM.
- o Automatic load of frame 1 Program Dump. If Audio channel 2 is "ON", and a disc containing one or more program dumps "spins up", the player loads the initial program dump into RAM. When the load is completed, the player begins executing the program.
- o Programming Mode Select. If the player receives the PROGRAM command, any play in progress stops in a freeze-frame and the player enters Programming Mode.

* The videodisc player is equipped with a switch-selectable "Auto-Play" function that automatically begins play of the videodisc after the power is turned on when the drawer is closed.

B. Automatic Mode

In Automatic Mode, operation of the videodisc player is controlled by the application program stored in RAM. This mode is initiated by the successful loading of a program dump, or with the RUN command. If the user wishes to begin program execution at a specific memory location, he enters the location number and then the RUN command. If no location is specified, program execution begins at location 0000.

Input during Automatic Mode is limited to the following:

- Audio Control. Both Audio channel 1 and Audio 2 outputs are normally turned on when the player enters Automatic Mode. The viewer may toggle Audio 1 and Audio 2 ON or OFF, undetected by the application program. The fifteen Axx commands can be used to set the state of either or both audio output channels.

  A command to toggle CX* ON/OFF is also available. However, if the disc is encoded with automatic CX control codes, they take priority over the commands.

- Frame Display. The display of frame numbers is turned off when the player enters Automatic Mode. The viewer may toggle this display ON and OFF with the DISP command, undetected by the application program.

- Viewer Response. If the application program has been written to allow viewer interaction, it will normally request this interaction by presenting a statement or question, followed by multiple choices. The viewer may, at that time, make the numeric entry that corresponds to his choice (0 ... 9). The subsequent path of program control is altered by that choice.

- Mode Termination. The user may terminate Automatic Mode by issuing either the HALT or the REJECT command. A HALT command returns the player to Manual Mode; however, the player may continue in motion until the STOP button is pressed, depending on the program command which was being executed at the time of the HALT. A REJECT operates as discussed previously for Manual Mode.

* CX is a trademark of CBS, Inc.

- o Transmit Commands. Frame number, status and program addresses may be read. The three Transmit Memory commands may be used to read memory, and the Transmit Pointer may be incremented or decremented, but not set.

- o Video. May be set ON or OFF with the Video ON/OFF commands.

- o Flags. May be set ON/OFF with the E0 ... E7 commands.

C. Programming Mode

Programming Mode provides direct access to the "addressable" RAM memory used for Program dumps, which includes the 512 user "registers."

Programming Mode is initiated with the PROGRAM command. The user may then enter commands and data to construct a program that will subsequently control the operation of the player in Automatic Mode. During Programming Mode, the television set displays numerical values and command mnemonics as they are entered. In addition, the programmer may examine any selected memory location by preceding the PROGRAM command with the location number. While in Programming mode, an additional PROGRAM command will advance the display by one location, allowing the user to view the RAM controls without making any changes.

Programming Mode is terminated with the END command, which returns the player to Manual Mode.

Note: The user may terminate Programming Mode with REJECT; this action will not cause the REJECT command to be stored in the program.

CHAPTER II.

COMMAND INFORMATION

1. GENERAL

This chapter provides the user with the information necessary to develop application programs for use with the videodisc player. It contains a brief description of the programmable memory and registers, describes the essential characteristics of a user program, and presents a comprehensive discussion of commands for use in programs and for external control of the player.

Throughout the remainder of this manual, extensive reference is made to hexadecimal values. Hex values are denoted by single quotes ('F2') or by a leading asterisk (*F2). Although not standard notation, this method has been adopted because the video display employs the asterisk convention to display many of the commands that are entered into RAM during Programming Mode.

2. RANDOM ACCESS MEMORY

The videodisc player includes a Z80 microcomputer, an EPROM that contains the basic operating system of the player, and eight kilobytes of Random Access Memory (RAM), of which 1K (1024 bytes) is available to the user. The RAM performs a dual function, serving as both addressable program command memory and user registers. This coexistence requires the RAM to be used both for program storage and as a group of 512 registers that the program can reference. Because memory and registers may overlay each other, the programmer must understand how each is structured and addressed.

A. Addressable Memory

Each RAM memory location is one eight-bit byte. Memory locations begin at address 0 and continue to address 1023. Locations 1022 and 1023 are not normally used for program storage, but are reserved for use as Register 0.

Program data, i.e., arguments and commands, are stored in coded format. Each argument digit and each command require one byte, and thus occupy one memory location. For example, the value 1536 stored within a program as a command argument occupies four bytes and has the hexadecimal format *0F*AF*4F*6F.

B. User Registers

Each user register occupies two bytes of RAM (two memory locations). There are 512 possible registers, addressed 0 through 511. Register numbers begin at the high end of memory and proceed downward: Register 0 occupies memory locations 1022 and 1023, while Register 511 occupies memory locations 0 and 1.

Note: The two memory locations corresponding to a particular register number can be computed as follows:

> Mem Loc = 1022 - (2 x Reg. No.)
> The register will occupy Mem Loc and Mem Loc+1, with the high order byte stored in Mem Loc and the low order byte in Mem Loc+1.

Register data are stored in a 16-bit binary format. For example, the value 1536, can be stored in a two-byte register, and has the hexadecimal value *0600.

Since registers are 16 bits long, they can contain values of 0 through 65535. There are no negative numbers, and larger positive numbers are taken modulo 65536.

Of the 512 available registers, only one is "active" at any given time. A register must be active before its contents can be used or modified by the program.

Register 0 is set active at Power-ON and by REJECT. Any register can be designated as active by specifying it as the argument of a RECALL command. In addition, successive registers are automatically activated whenever one of the following commands is executed, even if the command does not use the register: AUTOSTOP, SEARCH, STORE, and RECALL.

3. USER PROGRAM

A user program for the videodisc player consists of a series of commands that, when stored in RAM and interpreted by the microcomputer, cause the player to operate in a predefined way.

The following is a brief description of arguments and commands, and gives an overview of a program structure. Detailed descriptions of arguments and commands are contained in later sections of this chapter.

A. Arguments (Digit Commands)

Arguments specify the numerical parameters required by their associated commands. An argument may specify an integer data value, a frame number, a memory location, or a register number. These arguments may denote time delays or the acceptable range of numeric values to be entered by the viewer in response to an INPUT command.

Some commands cannot take arguments; others require arguments; for still others, an argument is optional. In the case of optional arguments, there is usually a default value, or an implied argument taken from the active register.

Each numeric digit of an argument is internally represented as a two digit hexadecimal value. When stored in memory, each of these two digit hex values occupies one memory location. When the programmer enters digit commands, the video display shows the numeric value of the digit, not the command hexadecimal representation.

Any number of digits may be entered immediately prior to a command, but only the low-order five digits are interpreted as the argument. The numerical value of these low-order digits is taken as modulo 65536.

B. Commands

Commands represent the functions available for development of an application program. Many of the commands are direct counterparts of buttons on the RCU (e.g., AUDIO1, DISP, etc.). They cause corresponding operations to be performed by the player. Other commands are used for controlling program interpretation, directing the path of execution, managing registers, etc.

Each command is internally represented by a two digit hex value. (Refer to Appendix A.) Many of these commands can be executed directly by the player or entered into RAM from the RCU as a single button press; others cannot be executed from the RCU, but can be entered into RAM with three button presses, following the procedure described in Chapter IV. When stored in memory, each command occupies one location (one byte). When the programmer enters commands from the RCU, the video display shows some of the commands as hex values, and others as command mnemonics.

Almost all commands can be executed or input into RAM with one 8-bit input via the ECI RS232 port.

C. Program Structure

A user program, when stored in memory, resembles a continuous string of characters. The string is processed by a program interpreter contained in the player's EPROM. Beginning at the location specified or implied by the RUN command, each element of the string is inspected for a command code. When the command is located, an internal command processor performs the required function. Characters preceding the command code are treated as the command argument. Some commands have implied, default, or no arguments; in such cases there will not be any preceding argument in the character string.

As an example, the commands 1000 SEARCH 2000 AUTOSTOP are internally represented as *0F*3F*3F*3F*F7*8F*3F*3F*3F*F3. Starting with the first byte, *0F, the argument is accumulated while the character string is being scanned for a command code. In this example, the SEARCH command code, *F7, is detected. The SEARCH command processor, using the currently accumulated argument (*0F*3F*3F*3F), instructs the player to position the laser at frame 1000.

When in Automatic Mode, succeeding commands in RAM are not processed until the function specified by the "current" command has been completed. The PLAY command is the only exception. A PLAY command instructs the videodisc player to begin playing audio-video material, and to continue until terminated by the program. Program execution continues while the play is in progress.

Note: Except for SEARCH, AUTOSTOP, WAIT, etc., each command, with up to 15 preceeding digits, is executed in one cycle (1/60 sec).

4. COMMAND DESCRIPTIONS

This section provides a detailed description of each command in the videodisc player's instruction set.

The program commands are grouped by function:

Mode Control
Program Load Control
Audio Control
Video Control
Register Management
Input Processing
Program Execution Control
Flag Setting
Transmit Functions

Within each function group, individual commands are presented in accordance with the standard format shown below:

#. **COMMAND NAME** - Command Title

	CODING	HEX	EQUIVALENT	DSPLY
ARGUMENT	MNM	VAL	RCU BUTTON	MNM

The first line of the description contains the command name and the command title, e.g. AUTOSTOP - Automatic Stop. For clarity, the COMMAND NAME is used throughout the description and examples.

The second line contains a tabulation of command programming information:

ARGUMENT — If an argument is required for a particular command, the table indicates the argument type, i.e., frame number, register number or integer. If the argument is optional, it is enclosed in parentheses.

CODING MNM — Following the argument is a command mnemonic which the programmer should use when coding a program to be submitted to Pioneer Video, Inc.

HEX VAL — The hexadecimal value that follows the mnemonic identifies the command to the program interpreter. During Programming Mode, the programmer may enter this value into memory by following the hex-entry procedure described in Chapter IV.

EQUIVALENT RCU BUTTON — For many commands, the hexadecimal value of the command may be entered directly as a single button-press. This entry indicates what button, if any, corresponds to the command being described.

DSPLY MNM — Commands entered during Programming Mode are displayed on the video screen at the time of entry. Some commands are represented by one or two-letter mnemonics; others are shown as hex values. The table indicates the displayed characters for the command under discussion.

Following the table is a description of the command, specifying the function it accomplishes, how it relates to other commands and, where appropriate, how it affects the videodisc player when the command is executed from RAM during Automatic Mode. Some commands cause different operations when executed in Manual Mode, or when received by the player while it is in Automatic Mode. These differences are described when appropriate.

"Notes" are sometimes included, to reflect details of the command or its operation that are preculiar to this version of the player, but may be different in future versions. They are included for understanding, but these details should not be relied upon in designing applications.

The final paragraph provides an example of use of the command in a program (except for those few commands whose function is obvious from the description). Most of the examples necessarily include commands other than the one under discussion. There is a brief description of each command as used in the example.

A. Mode Control Commands

Mode Control commands are those the user enters to initiate and terminate Programming Mode and Automatic Mode.

The commands described in this section are:

1. PROGRAM
2. END
3. RUN
4. HALT

The commands PROGRAM and END should not be included in a user program, even though it is possible to enter them. The PROGRAM, END, and RUN commands are ignored if the player receives them while it is in Automatic Mode.

Note: If the player executes a "PROGRAM" or an "END" command from a stored program, these commands will be functionally ignored, but it will use one execution cycle (16.7 ms) .

1) **PROGRAM** - Enter Programming Mode

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
(Address)	None	DF	PROGRAM	*DF

While in Manual Mode, the PROGRAM command is entered to initiate Programming Mode. Following entry of this command, new data can be entered into RAM, or existing data can be reviewed. A specific memory location may be accessed by entering the desired address, followed the PROGRAM. If no address is specified, location 0000 is assumed. Argument values greater than 1023 are taken modulo 1024.

Program entries are stored in successive memory locations. Each instruction and each digit of an argument occupy one memory location. The video screen displays the current memory address and the contents of that address. A subsequent PROGRAM command advances the address by one location.

Previously stored information can be edited or updated during a program review by entering a new command or data byte to replace the one being examined. The user should take care not to inadvertently overwrite necessary commands or data.

2) **END** - End Programming Mode

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	None	EF	END	*EF

The END command is used to terminate Programming Mode and to return the player to Manual Mode.

Note: A REJECT command will also terminate Programming Mode; the videodisc will stop rotating and the player will return to the PARK position. REJECT is not entered into RAM. Frame Display, Audio ON/OFF, etc. are not affected by Programming Mode.

3) **RUN** - Run Program

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
(Address)	None	CF	RUN	None

Execution of programs stored in RAM may be initiated with the RUN command or in conjunction with the LOAD command. (Refer to Section B, "Program Load Control Command".)

Use of RUN requires that the player be in Manual Mode. The command causes the microcomputer to begin executing the program at the specified address (location), or at location 0000 if no argument is supplied. RUN turns on Video and Audio 1 and 2, turns off CX and the frame-number display, and transfers the player from Manual Mode to Automatic Mode.

Note: The active register pointer is set to 1.

Note: If the player is in Programming Mode, the RUN command becomes the BRANCH command. (Refer to Section G "Program Execution Control Commands.)

4) **HALT** - Halt Program

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	H	BF	CLR/HALT	H

The HALT command terminates program control of the player, and returns the player to Manual Mode. A HALT from RAM, the RCU, or ECI will take the player out of Automatic Mode and place it in Manual Mode. Any partial argument, left over from the previus entry of digits or partial execution of a program, is cleared. (Programs need not contain a HALT instruction if they are designed to execute continuously.)

Note: If a programmed HALT is executed in RAM while a PLAY is in progress, program control terminates, but the videodisc continues to "play".

B. Program Load Control Command

1) **LOAD** - Load Program From Disc

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	L	CC		*CC

The LOAD command is used to load a 1022 byte program dump from the disc into the player's RAM. Only 1022 bytes are loaded, not 1024. The remaining two bytes of RAM (addresses 1023 and 1024 which correspond to Register 0), are not contained in the program dump. Register 0 is therefore not overlayed or changed by a LOAD, and may be used for passing data between different programs. The active register pointer is not affected. There are no prohibited byte values in a dump. Any of the hex codes ('00' through 'FF') may be loaded into RAM via a program dump.

Program dumps are physically located on Audio channel 2. When the LOAD command is executed, valid program leader must be detected from Audio channel 2 for the load to be successful. It is not necessary for Audio 2 output to be turned on except in the case where the dump is to be loaded automatically, immediately following the initial PLAY command. If the load is successful, program execution begins at location 0000.

Load data bits are recorded at 5000 per second and span approximately 50 frames of Audio Channel 2. The combined leader tone and program data may occupy 60 or more frames of Audio Channel 2. Information regarding the recommended positioning and spacing of multiple program dumps may be obtained from Pioneer Video, Inc.

CAUTION: This version of the player does not squelch the video during the LOAD.

Note: If no progam leader signal is detected, or the load cannot be successfully accomplished, the player may freeze-frame at the point where the load was supposed to begin, and then revert to Manual Mode. Also, the player may not properly process commands received while it is in the process of loading a program dump from disc. The LOAD command is ignored unless executed from RAM.

Example: The dump leader tone signal starts before frame 450, and the program dump starts after frame 450. The associated RAM code would be:

```
450 SEARCH
    LOAD
```

C. Audio Control Commands

The following commands provide program control over the output of audio information contained on a videodisc:

1) CX
2) AUDIO1 or AUDIO2
3) Axx (where X can be I, F, N, or T)

All of these commands (without arguments) can be executed by the player when received from the ECI (A1 and A2 from the RCU also). This is true in both Manual and Automatic modes. The Audio and CX commands only take an argument when executed from RAM.

1) CX - CX Control

	CODING	HEX	EQUIVALENT	DSPLY
ARGUMENT	MNM	VAL	RCU BUTTON	MNM
(Integer)	CX	EE		CX

This command turns the CX decoder ON and OFF. If it doesn't have an argument, this command will toggle the CX function. If the argument is even, it will turn the CX decoder OFF and if the argument is odd, it will turn the CX decoder ON.

Note: If the disc being played has automatic CX control codes, the disc's control codes will override the CX commands.

2) **AUDIO 1/2** - Audio Output Control

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
(Integer)	A1	F4	AUDIO 1/L	A1
(Integer)	A2	FC	AUDIO 2/R	A2

AUDIO1 controls output from Audio Channel 1 (Left), and AUDIO2 controls Audio Channel 2 (Right). When the command is preceded by an even argument (i.e., 0, 2, 4 ...), audio from that particular channel is turned "OFF"; an odd argument (i.e., 1, 3, 5 ...) turns the audio channel "ON". If there is no preceding argument, the commands act as a toggle switch to turn audio on or off, depending on its previous state.

When both audio channels are ON, each signal is available at its own audio output jack, and the combined signal is put into the VHF output. When only one channel is on, it's signal appears at both audio outputs.

Example: Play from frame 1000 to frame 3000 with Audio output 1 off and Audio output 2 on; from frame 3000 to frame 5000 with Audio output 1 on and Audio output 2 off.

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0013	0 AUDIO1	Turn off Audio 1.
0015	1 AUDIO2	Turn on Audio 2.
0017	1000 SEARCH	Search to frame 1000.
0022	3000 AUTOSTOP	Play to frame 3000.
0027	AUDIO1	Toggle Audio 1 (on)*.
0028	AUDIO2	Toggle Audio 2 (off)*.
0029	5000 AUTOSTOP	Play to frame 5000 and freeze frame.

* This would actually be poor programming practice if it was really desired to force AUDIO1 on and AUDIO2 off. The viewer might have toggled either audio channel during the AUTOSTOP.

3) **Axx** - Set Audio Status

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	AFF	A0		*A0
	AFN	A1		*A1
	ANF	A2		*A2
	ANN	A3		*A3
	AFT	A4		*A4
	AFI	A5		*A5
	ANT	A6		*A6
	ANI	A7		*A7
	ATF	A8		*A8
	ATN	A9		*A9
	AIF	AA		*AA
	AIN	AB		*AB
	ATT	AC		*AC
	ATI	AD		*AD
	AIT	AE		*AE

The audio status commands, coded *A0 through *AE, allow the programmer to set Audio Channel 1 and Audio Channel 2 to any one of four possible combinations through use of a single-byte command. Outputs from the audio channels may be turned on, turned off, toggled, or ignored (left in the same state). The following table indicates the function performed by each command. Note the use of T for toggle, F for off, N for on, and I for ignore (do nothing).

CMD	EFFECT ON AUDIO CHANNELS
AFF	Turn off 2, Turn off 1.
AFN	Turn off 2, Turn on 1.
ANF	Turn on 2, Turn off 1.
ANN	Turn on 2, Turn on 1.
AFT	Turn off 2, Toggle 1.
AFI	Turn off 2, Ignore 1.
ANT	Turn on 2, Toggle 1.
ANI	Turn on 2, Ignore 1.
ATF	Toggle 2, Turn off 1.
ATN	Toggle 2, Turn on 1.
AIF	Ignore 2, Turn off 1.
AIN	Ignore 2, Turn on 1.
ATT	Toggle 2, Toggle 1.
ATI	Toggle 2, Ignore 1.
AIT	Ignore 2, Toggle 1.

D. Video Control Commands

The following commands control the selection, sequencing, and display of video information contained on the videodisc:

1) AUTOSTOP
2) DISPLAY
3) SLOW SET
4) FAST SET
5) PLAY
6) SEARCH
7) MUTLI FWD/REV
8) STEP FWD/REV
9) STOP
10) WAIT
11) VOFF
12) VON
13) SCAN FWD/REV

1) **AUTOSTOP** - Play With Automatic Stop

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
(Frame Number)	AS	F3	AUTOSTOP	AS

The AUTOSTOP command begins normal-speed play of a videodisc, starting at the current frame number and ending in freeze-frame at the frame specified by the argument. If no argument is specified, an argument is taken from the active register.

The AUTOSTOP command always causes the current active register to be deactivated and the next register to be activated, whether or not an argument is specified.

In Manual Mode, AUTOSTOP can be interrupted by another motion control command (STOP, STEP FWD, PLAY, SEARCH, etc.).

Note: If the frame number associated with the AUTOSTOP command is less than the current frame number, the player will not play in reverse; it will search to the target frame and may, in the process, display other frames.

Example: Play 1000 frames of a sequence, delay five seconds and play the next 1500 frames. Assume:

R50 = 2000, R51 = 8000, R52 = 4500

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0010	50 RECALL	Activate Register 50.
0013	SEARCH	Search to frame 2000 and activate Register 51.
0014	3000 AUTOSTOP	Play to frame 3000; activate Register 52.
0019	50 WAIT	Display frame 3000 for five seconds.
0022	AUTOSTOP	Play to frame 4500 and freeze frame

2) **DISPLAY** - Frame Display

ARGUMENT	CODE MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
(Integer)	DI	F1	FRM DSP	DI

The DISPLAY command controls the ON/OFF status for the display of the five-digit frame number. An even argument (i.e., 0, 2, 4...) turns the frame number display off; an odd argument (i.e., 1, 3, 5...) turns it on. If used without an argument, DISPLAY acts as a toggle switch to turn the frame number on or off, depending on its previous state. The command takes an argument only when in RAM.

When a CLV disc is playing, a Time Code is displayed which may include "seconds" if the disc has them encoded. Otherwise, the Time Code shows only hours and minutes.

Note: The RUN and REJECT commands turn the frame display off. During Automatic Mode, the user may toggle the frame display at any time, undetected by the program. Also, the DISABLE CHARACTER GENERATOR command (refer to Section 5.A.) turns off all display from the character generator, regardless of the DISPLAY ON/OFF status.

Example: Ensure that frame numbers will be displayed as the first video sequence begins playing:

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0049	EF0	Turn on the Character Generator.
0050	1000 SEARCH	Search to frame 1000
0051	1 DISPLAY	Turn ON the frame display.
0053	2000 AUTOSTOP	Play to frame 2000 with the display on.
0054	DISPLAY	Toggle the frame display.
0055	3000 AUTOSTOP	Play to frame 3000 with the frame display off.

3) **SLOW SPEED SET** - Play Slow Speed Control

ARGUMENT	CODE MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
Integer	SS	ED	Multi-speed Set (Slow)	SS

This "SLOW SPEED SET" command defines the speed that is to be used by the MULTI FWD/REV mode. The argument is a speed parameter, which should be from 0 through 127.

Note: If an argument over 127 is used, it will be taken modulo 128.

Argument	Play Speed
0	Still
1	Normal
2	Normal x 1/2
N	Normal x 1/N
127	Normal x 1/127

4) **FAST SPEED SET** - Play Fast Speed Control

ARGUMENT	CODE MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
Integer	FS	EC	Multi-speed Set (Fast)	FS

This "FAST SPEED SET" command defines the speed that is to be used by the MULTI FWD/REV mode. The argument is the speed parameter, which should be 0 through 3.

Note: Three times normal is the current maximum. If an argument over 3 is used, the player will operate at three times normal play.

Argument	Play Speed
0	Still
1	Normal
2	Normal x 2
3	Normal x 3

5) **PLAY** - Play a Sequence at Normal Speed

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	P	FD		P

The PLAY command instructs the videodisc player to begin playing the disc at normal speed, with audio output unsquelched, starting at the current frame and continuing for an indeterminate time. When the PLAY command is entered into memory, it must be entered as PLAY, MULTI FWD, SEARCH. The PLAY command itself, in Programming Mode, conditions the player to accept the next two button presses as a byte value in hexadecimal format. (Refer to Chapter IV.)

The PLAY command permits program execution to continue while the disc is being played, and thus permits motion sequences to be displayed on a time basis rather than a frame basis. Play is terminated when one of the following commands is executed:

AUTOSTOP
REJECT
SEARCH
MULTI FWD
MULTI REV
STEP FWD
STEP REV
STOP (WAIT)

The HALT command ends program execution, but does not terminate PLAY.

Picture Stops are read by the player, and will cause a freeze frame if the player is in Manual Mode.

Note: At leadout, the player continues playing through leadout for about 10 seconds; then the player searches to the beginning of the disc, and continues playing from there.

6) **SEARCH** - Search to a Frame

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
(Frame Number)	SC	F7	SEARCH	SC

The SEARCH command instructs the player to position the laser at the frame number given in the argument. If no frame number is provided, the frame number in the active register is used. Video is "black" during the SEARCH; at the end of the SEARCH the target frame image is displayed in freeze-frame mode. SEARCH activates the next register, even if there is an argument.

If the player is unable to locate the target frame, it makes two retries at the next two higher frame numbers. If these retries are unsuccessful, the SEARCH ends on a nearby, but indeterminate frame, and the next program instruction is executed.

Note: During a manual mode search, if the player receives a mode-control command (for example, PLAY, AUTOSTOP, STOP, STEP F/R, SCAN F/R, MULTI F/R, PROGRAM, RUN, or REJECT) the player immediately executes the new command. Generally, other commands are executed as well. They do steal some time from the search, however the search continues.

Example: The command 1500 SEARCH directs the player to search to frame 1500, stop, and display the image in freeze-frame.

7) **MULTI FWD/REV** - Multi-Speed Forward or Reverse

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
(Frame Number)	MF	F2	Multi Fwd	MF
(Frame Number)	MR	FA	Multi Rev	MR

Using the speed which is currently defined (by SS, FS, or default), the player will "play" forward or reverse until the specified frame number is reached or passed. At normal play speeds or slower, the player will freeze on the frame specified. However at faster than normal speeds, the player may stop after the frame is passed. If the player is instructed to play in the wrong direction to reach the target frame, the player will search to the indicated frame number and stop. In this playback mode, audio output is squelched.

This command is an expansion of the previous SLOW FWD/REV command. With MULTI F/R, the speed is set by command rather than with the manual slide control of previous players, and 2x and 3x speeds are also available.

In Manual Mode, the speeds may be changed while in MULTI-SPEED play, with the most recent speed setting, whether FAST or SLOW, overriding the previous setting. The default speed, set by Power-on and REJECT, is one-quarter speed, as would be set by the command "4 SS."

Picture stops are read by the player, and will cause a freeze frame if the player is in Manual Mode and not playing faster than normal speed.

Note: When the command is in RAM it must have an argument or it will be ignored. When not in RAM, it does not use an argument.

8) **STEP FWD/REV** - Step Single Frame Forward or Reverse

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	SF	F6	STEP FWD	SF
	SR	FE	STEP REV	SR

The STEP FWD/STEP REV commands cause the videodisc player to "freeze frame" and to step forward or backward one frame for each execution of the command. These commands can be used in a programmed loop to display a series of still frames, or to provide programmed slow-motion effects.

Note: These commands are "executed" in one cycle, but the motion that is begun by the command is not completed in that cycle.

Example: Beginning at the current frame, display the next ten frames as a series of 8-second stills:

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0000	20 RECALL	Activate Register 20.
0003	10 STORE	Place the value 10 in Register 20.
0006	STEP FWD	Step to the next frame for 8 seconds.
0007	80 WAIT	Display the frame for 8 seconds.
0010	20 DEC REG	Decrement Register 20. If R20 is still not 0, go to location 00013. If R20 is now 0, skip one BRANCH to go to location 0015.
0013	6 BRANCH	Go to location 0006.
0015	HALT	Terminate the program.

9) **STOP** - Freeze Frame

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	W	FB	STOP	W

The STOP command causes the player to enter freeze-frame mode. Audio output is squelched.

10) **WAIT** - Freeze Frame and Delay

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
Integer	W	FB	STOP	W

The WAIT command causes a freeze-frame for a period of time defined by the integer argument. The integer specifies the time in tenths of a second.

The WAIT command will first cause the motion to stop in a freeze frame; then it will wait for the designated time before executing the next program command. The WAIT command must be in RAM.

Note: The maximum freeze-frame delay available is 1000 seconds (10,000 tenth seconds or 10000 WAIT).

Example: A programmed freeze frame for 5.2 seconds is coded as:

```
52 WAIT
```

11) **VOFF** - Video Off

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	VFF	1C		*1C

The VOFF command turns the video image "black" leaving a "blank" screen. The video images coming off the disc will become visible again after execution of a VON or REJECT command. Although the screen is "black", the player continues to produce proper NTSC signals, including sync.

12) **VON** - Video On

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	VON	1B		*1B

The VON command turns on the video image, which may have been previously "blacked out" by VOFF.

Example: Flash the video image off and on.

MEMORY LOCATION	ARGUMENT AND COMMAND		DESCRIPTION
0000	1000	SEARCH	Search to frame 1000.
0005		VOFF	Turn video off.
0006	10	WAIT	Delay 1 second.
0009		VON	Turn video on.
0010	10	WAIT	Delay 1 second.
0013	5	BRANCH	Loop again.

13) **SCAN FWD/REV** - Scan Forward and Reverse

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
		F0	SCAN (FWD)	
		F8	SCAN (RWD)	

SCAN FWD and SCAN REV can be executed only while the videodisc player is operating in Manual Mode. These commands cause the player to scan rapidly in a forward or reverse direction until the RCU button is released. If, however, these command codes are encountered in Automatic Mode, they will be interpreted as DEC REG (*F0) and INPUT (*F8).

Scan across the entire disc takes about 15 seconds.

E. Register Management Commands

The following commands affect the user registers and their contents:

1) RECALL
2) DEC REG
3) STORE

1) **RECALL** - Recall and Activate Register

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
(Register Number)	RC	7F	RECALL	RC

The RECALL command sets the active register pointer to the register number specified in the argument. This "activates" the register specified by the argument. In Manual Mode, the register number and its contents are also displayed on the TV screen by the character generator. If no register number is specified, one of the following occurs:

- o If the active register pointer has not been incremented since the last RECALL command, RECALL activates the next sequential register (increments the active register pointer).

- o If the active register pointer has been incremented since the last RECALL command, RECALL "reactivates" the already-active register, - i.e., the active register pointer does not change.

Note: The following commands increment the active register pointer: SEARCH, AUTOSTOP, and STORE. However, HALT and CLEAR do not change the active register pointer. RUN sets the pointer to 1.

Example:

ARGUMENT AND COMMAND	DESCRIPTION
10 RECALL	Activates R10
RECALL	Activates R11
300 SEARCH	Activates R12
RECALL	Activates R12
RECALL	Activates R13

2) **DEC REG** - Decrement Register

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
Register Number	DR	F0	DEC REG	DR

The DEC REG command subtracts one from the contents of the register specified in the argument. If the result is greater than 0, command execution proceeds with the command immediately following the DEC REG. If the result is equal to 0, all succeeding commands are skipped until one BRANCH is passed; then command execution resumes with the next command immediately following the passed BRANCH. There are no negative numbers -- a zero decremented remains a zero. An argument is required. A DEC REG command does not change the active register pointer. This command can only be executed from RAM.

Note: If the command doesn't have an argument, the command will be ignored.

Example: Assuming that Registers 21 through 30 contain different frame numbers, play five scenes and then quit:

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0050	20 RECALL	Activate Register 20
0503	5 STORE	Set the loop counter to 5; activate R21
0505	SEARCH	Search to the frame in the active register, then activate the next register.
0506	AUTOSTOP	Play to the frame in the active register, then activate the next register.
0507	20 DEC REG	Decrement R20 - the loop counter. If the contents = 0, go to location 0514, otherwise continue at location 510.
0510	505 BRANCH	Transfer to location 505.

3) **STORE** - Store Number in Active Register

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
(Integer)	ST	F5	STORE	ST

The STORE command places a value in the active register. The value stored is the number contained in the argument. If there is no argument, the current frame number is stored. Then, the STORE command increments the active register pointer which "activates" the next register.

Note: Argument values greater than 65535 are taken as Modulo 65536.

Example: Place the current frame number in Register 20 and the value 5000 in Register 21:

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0010	20 RECALL	Activate Register 20.
0013	STORE	Store the current frame number in Register 20 and activate Register 21.
0014	5000 STORE	Store 5000 in Register 21 and activate Register 22.

F. Input Processing Command

The command described in this section is used to process a viewer input entered from the RCU or host while a program is running. The subsequent path of program execution is changed, depending upon the viewer's input, thereby permitting a wide range of interactivity between the viewer and the videodisc player.

1) **INPUT** - Input Viewer Data

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
Integer	IN	F8	INPUT	IN

The INPUT command allows the processing of a numeric entry that is input to the player by the user. The input is usually made in response to a question or "menu" displayed on the video screen. The entry is a single digit in the range of 0 through 9. INPUT can only be executed in RAM.

The INPUT command's argument is normally a single digit N. High-order digits, if present in the argument, are ignored. The argument N establishes the range of "expected" viewer input, digits 0 through N-1. Any digit outside the expected range is treated in one single category, "other." For example, if N = 3, one of the three numeric responses (0, 1 and 2) are expected; any other numeric entry (3 through 9) falls into the category "other."

When coding an INPUT for three expected entries, there should be three "groups" of commands following the INPUT statement, each group ending with a BRANCH command. If the viewer responds to the Input with a digit that is within the expected range, that digit specifies the number of "groups" of commands to be skipped before program execution resumes. If the viewer's response is in the "other" range, N "groups" are skipped.

NOTE: An argument value of 0 is treated the same as an argument value of 9. If compatibility with PR7820-I is desired, the byte immediately after the INPUT should be coded as 'FF'.

The following table shows the relationship between argument values, "expected" input and "other" input.

ARGUMENT VALUE (N)	RANGE OF "EXPECTED" NUMERIC INPUT	RANGE OF "OTHER" NUMERIC INPUT
1	0 only	1 through 9
2	0 or 1	2 through 9
3	0 through 2	3 through 9
4	0 through 3	4 through 9
5	0 through 4	5 through 9
6	0 through 5	6 through 9
7	0 through 6	7 through 9
8	0 through 7	8 and 9
9	0 through 8	9 only

Example: The following sequence of commands will cause a search to frame 1000 and will accept viewer input for two expected entries and one "other."

MEMORY LOCATION	ARGUMENT AND COMMAND	DESCRIPTION
0100	1000 SEARCH	Search to frame 1000.
0105	2 INPUT	Allow the viewer input of 0, 1, or "other":
0107	1500 SEARCH	Search to frame 1500 (in response to a "0").
0112	130 BRANCH	Go to location 0130.
0116	2000 SEARCH	Search to frame 2000 (in response to a "1").
0121	130 BRANCH	Go to location 0130.
0125	2500 SEARCH	Search to frame 2500 (in response to a "2" through "9").
0130	55 WAIT	Show the frame 2500 for 5.5 seconds.
0133	....	Next instruction.

G. Program Command Sequence Control

The following command is used to control and alter the order in which program commands are executed:

1) BRANCH

These commands have special uses in a program:

2) NE
3) REJECT

1) **BRANCH** - Alter Execution Sequence.

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
(Address)	BR	CF	BRANCH	BR

The BRANCH command alters the normal sequential flow of command execution by directing the program to continue execution at the memory location specified by the integer argument. If no argument is present, location 0000 is assumed. The BRANCH command does not alter the existing status of Audio outputs 1 and 2, and has no effect on the frame number display or the active register. The BRANCH command can only be executed from RAM.

BRANCHes are generally used under the following conditions:

o For conditional branching or loop control following a DEC REG.

o For path selection, where a BRANCH command ends each of the command "groups" that follow an INPUT command.

o An unconditional branch to cause immediate transfer of control to another location.

Examples:

	CODE	EFFECT
o	BRANCH	Transfer control to location 0000.
o	256 BRANCH	Transfer control to location 00256.

2) **NE** - No Entry

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	NE	FF		NE

NE instructs the computer to "do nothing", and then execute the next command. It may take some time for it's "execution," and a NE occupies one memory location. NE's might be used to "erase" unwanted commands which are part of an already loaded program, or put into new programs to reserve space for future program modifications.

Note: Leading (non-significant) zeros in command arguments might be used in a similar manner. Execution of NE requires one program cycle.

3) **REJECT** - Reject Videodisc

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	RJ	F9		RJ

The REJECT command terminates program execution, stops disc rotation, and returns the player to the PARK position. It causes a 1 to be stored in Register 0. REJECT is a legitimate programming instruction. It may be entered into the player's RAM only in the 3-command format (see Chapter IV). When executed, it will cause the program to abruptly terminate by rejecting the videodisc.

In addition, the following effects occur as a result of a REJECT:

o Video is turned ON
o Frame Display and CX are turned OFF
o Player enters Manual Mode
o AUDIO 1 and AUDIO 2 are turned ON
o The Character Generator is ENABLED
o Multi-Speed is set to 1/4 normal speed
o The active register pointer is set to zero
o Register zero is set to one.

H. Flag Set Commands

The following paragraphs describe some commands that facilitate control of the videodisc player by an attached external computer. These commands may also be included in a program dump to help the player control an external device, printer, terminal, computer, etc.

1) Enable or Disable Character Generator

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	EF0	E0		*E0
	EF1	E1		*E1
	EF2	E2		*E2
	EF3	E3		*E3

These commands Enable and Disable the Character Generator. These commands are not ignored if received by the player while it is in Automatic mode; they are executed. Power-on and REJECT both enable the character generator. There are four commands for compatibility with the PR7820-III.

CMD	Character Generator
EF0	ENABLED
EF1	DISABLED
EF2	ENABLED
EF3	DISABLED

2) Enable or Disable Binary Output

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	EF4	E4		*E4
	EF5	E5		*E5
	EF6	E6		*E6
	EF7	E7		*E7

Two flags are set or reset by commands EF4 through EF7: the Binary Output Mode flag, and the Unique Input flag. Reject does not affect these flags. Power-on sets the binary output flag OFF, and the Unique input flag ON.

1) Binary Output Flag: A transmission from the player is one byte or two bytes of data. The one (or two) byte transmissions are usually two (or four) ASCII hex characters (0 ... 9, A ... F), followed by a carriage return ('0D') and a line feed ('0A'). The high order hex character is sent first. The ASCII characters have the parity bit held at Ø. Of course, the parity settings of the RS232 interface may change the parity bit. In 8-bit data, no parity mode, the ASCII "A" is sent as a hex *41.

   This data format may prove inefficient for systems that can accept the eight bit data bytes directly. Such systems may turn on the Binary Output Flag to inhibit the reformatting of each data byte into two ASCII characters, and inhibit the carriage return and line feed.

2) Unique Input Flag: Normally, the player executes all commands that are input. However, if the Unique Input Flag is set OFF by the user, the player will consider each RS232 port entry to be significant only if it differs from the previous port entry. It is recommend that this be avoided, and that only the commands 'E5' or 'E7' be used. Future versions of the LD-V6000 may not support Unique Input OFF.

These commands are not ignored if entered while the player is in Automatic Mode; they are executed.

CMD	Hex	Binary Output Flag	Unique Input Flag
EF4	*E4	OFF	OFF (not recommended)
EF5	*E5	OFF	ON
EF6	*E6	ON	OFF (not recommended)
EF7	*E7	ON	ON

I. Transmit Commands

Nine commands are provided to support the communication of information from the player to an external device. The operation of most of these is affected by the status of the Binary Output flag. All two-byte data is sent high order byte first.

Note: All the Transmit commands, except STP, are executed by the player when received by the External Computer Interface, even when a program is running or the player is in PARK mode. This allows for monitoring of the player even while it is being controlled by a program dump.

1) Transmit Frame Number

	CODING	HEX	EQUIVALENT	DSPLY
ARGUMENT	MNM	VAL	RCU BUTTON	MNM
	TFN	D3		*D3

The 16-bit binary number representing the most recent frame number is two binary bytes of data. This data is normally transmitted as four ASCII hex characters, high order byte first followed by CR LF. A byte is sent as two ASCII characters, with the high order nibble first. Frame number 1034 is high byte '04' and low byte '0A', so the characters "040A" are transmitted.

If the Binary Output Flag is ON, the frame number is transmitted as two 8-bit bytes, with the high order byte first, and no CR or LF is added. This allows faster access to the frame number.

When a CLV disc is playing, this command will send a "Time Code", in seconds. For example, one hour and 40 minutes = 6000 seconds, so "1770" would be transmitted. When a CLV disc has the codes for "seconds", the time code will be displayed as hours, minutes, and seconds. The Transmitted Time Code will increment by one each second, instead of incrementing by 60 at the end of each minute.

Note: During Leadin, Leadout, and PARK, special "frame numbers" will be sent.

2) Transmit Program Address Pointer

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	TPA	DB		*DB

The current value of the Program Address pointer is transmitted. The two bytes of data (0000 through 03FF) are sent high order byte first, like the frame number. The address is that of the next byte to be executed, or the next byte to be fetched.

3) Transmit Status

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	TS	D4		*D4

One data byte is sent, either in ASCII-hex format, or as one binary byte. The player's modes are:

4C	SCAN EXECUTING
50/D0	SEARCH EXECUTING
55/D5	AUTOSTOP
E0	INPUT WHILE PLAYING
E1	INPUT AT FREEZE FRAME
64/E4	PLAY NORMAL FORWARD
65/E5	FREEZE FRAME and STEP FWD/REV
66/E6	PLAY MULTI FWD
67/E7	PLAY MULTI REV
68	SPIN UP
78	PARK or DRAWER OPEN or LOADING
7C	REJECT or UNLOADING

The byte format is:

Bit 7 (high order): Automatic Mode Flag
Bits 2 through 6: Current Player Mode
Bits 0 and 1 (low order): Effects

00	PLAY
01	FREEZE FRAME
10	MULTI FWD
11	MUTLI REV

4) **Set Transmit Pointer**

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
Integer	STP	E8		*E8

A pointer is used to determine which byte is to be output for three of the fifteen TRANSMIT commands. This pointer should be set to point to one of the 1024 user program bytes (0 through 1023).

The pointer value is taken modulo 1024. The integer operand is used to set the value of that pointer. Two of the TRANSMIT commands modify the pointer before a byte is transmitted. Power-on and REJECT initialize the pointer to 0.

Note: This command will not be accepted as input by the player during Automatic mode.

Example: Monitor a program's use of the bytes in Register 3 (program locations 1016 and 1017).

ARGUMENT AND COMMAND	DESCRIPTION
1016 STP	Initialize pointer to program location 1016
ITM	Increment to low order byte (location 1017) and transmit
DTM	Decrement to high order byte and transmit

(Repeat ITM and DTM as required for monitoring.)

5) **Increment or Decrement Transmit Pointer**

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	ITP	D9		*D9
	DTP	DA		*DA

The transmit data pointer may be incremented (by ITP) or decremented (by DTP), allowing sequential data locations to be skipped rather than transmitted.

Example: Transmit the low order part of Register 511 (program location 1) in ASCII hex, followed by the low order part of Register 510 (program location 3).

ARGUMENT	COMMAND	DESCRIPTION
1	STP	Initialize pointer to program location 1
	TM	Transmit contents of program location 1
	ITP	Increment pointer to program location 2
	ITM	Increment pointer to program location 3 and transmit.

6) **Transmit Memory** (Contents of RAM)

ARGUMENT	CODING MNM	HEX VAL	EQUIVALENT RCU BUTTON	DSPLY MNM
	TM	DC		*DC
	ITM	DD		*DD
	DTM	DE		*DE

One byte is transmitted from RAM, from the address pointed to by the transmit-data pointer (taken modulo 1024). The transmit-data pointer is incremented by ITM (or decremented by DTM) before the byte is selected for transmission. Thus, any section of RAM memory may be transmitted or monitored.

The user program in RAM can contain any of the TRANSMIT commands, causing the player to alert or control external devices. (A clock chip and a UART could provide an external eight-bit register to hold the most recently transmitted data. Those data lines might then be used to control other devices.)

Example: Transmit an ASCII "A" if the user presses the numeric zero in response to an INPUT. Otherwise transmit a "B". In this example, the first executable command is at location 15.

MEMORY LOCATION	ARGUMENT	COMMAND	DESCRIPTION
0013		*41	ASCII "A"
0014		*42	ASCII "B"
0015	13	STP	
0018	1	INPUT	
0020		TM	
0021	25	BRANCH	
0024		ITM	
0025		...	

CHAPTER III

THE EXTERNAL COMPUTER INTERFACE

1. GENERAL

External control of the LD-V6000 player is available using the External Computer Interface (ECI) on the rear of the player. This external interface is an RS232C serial port. This chapter discusses the use of this RS232 port.

2. USING THE EXTERNAL COMPUTER INTERFACE

Commands entered via the External Computer Interface (ECI) are treated essentially the same as commands which are input from the RCU. However, the RCU can send only 32 commands, while the ECI port can receive 256 commands (*00 through *FF, see Appendix C).

Via the ECI, two-way transfer of codes from '00' to 'FF' is possible. However, as input, 34 of these codes are unconditionally interpreted as "ASCII-hex" input. Four other codes are ignored by the player: Carriage Return ('0D' and '8D') and Line Feed ('0A' and '8A').

Each input code is examined, ignoring the high order bit of the byte, for the ASCII "Ø" through "9" (*30 - *39) and ASCII "@", "A", ..., "F" (*40 - *46). The first "ASCII-hex" character received is considered to represent the upper four bits of a command code, and the next "ASCII-hex" character received is considered to represent the lower four bits of the command code, even if there are other intervening characters.

Thus, all commands can be input in the 2-character ASCII-hex format. Most commands can also be input as a single 8-bit byte.

3. COMMAND AND DATA FORMATS

Commands sent into the player through the ECI may be formatted in either of two formats:

A. Eight-Bit Binary Format

In this format, the 8-bit pattern must be equivalent to that represented by the hex value for that command.

Example:

```
   PLAY:  *FD = 11111101
Digit 2:  *8F = 10001111
   LOAD:  *CC = 11001100
```

Note: The 8-bit binary codes for carriage return ('0D' and '8D') and those for line feed ('0A' and '8A') are ignored by the player. The 8-bit binary code '00' is translated to 'FF'. Then, 'FF' (and thus the '00') is, for most practical purposes, ignored.

B. Two ASCII Characters Format

Commands may be formatted as two ASCII "hex" characters (Øthrough 9 and A through F). When it is more convenient to transmit only ASCII (7 bits with or without parity), two ASCII characters are transmitted to represent each command. The first hex character is the ASCII character representing the high-order 4-bit nibble of the command byte. The second ASCII character represents the low-order 4-bit nibble. For example: A RECALL (*7F) may be entered as an ASCII 7 followed by an ASCII F.

To accomplish this, 38 "8-bit codes" are reserved for "ASCII-hex" input. The 8-bit codes for the ASCII characters are shown in the following list. If the RS232 port receives one of these 8-bit "command codes," the code is intercepted and two are used to manufacture one player command. Carriage Return and Line Feed are ignored by the player.

In the event that an ASCII "hex" character (a nibble) is somehow lost in the process of data communication between the player and the attached computer, the at-sign (@) can be used to clean out any previously saved nibble, such that the next ASCII hex character is considered to be the high-order nibble.

ASCII CHARACTER	EQUIVALENT "COMMAND CODE"		BINARY
	P=0	P=1	
0	30	B0	P0110000
1	31	B1	P0110001
2	32	B2	P0110010
3	33	B3	P0110011
4	34	B4	P0110100
5	35	B5	P0110101
6	36	B6	P0110110
7	37	B7	P0110111
8	38	B8	P0111000
9	39	B9	P0111001
A	41	C1	P1000001
B	42	C2	P1000010
C	43	C3	P1000011
D	44	C4	P1000100
E	45	C5	P1000101
F	46	C6	P1000110
@	40	C0	P1000000
CR	OD	8D	P0001101
LF	OA	8A	P0001010

C. Three-Command Format

In Programming Mode there is a third format available for use. Data (or data that represents commands) can be put into the RAM memory using a sequence of three "normal" RCU commands. With this method the "PLAY" command is followed by two commands from the table below. The commands in this table are used to represent the two hex digits of the data byte that is to go into RAM. Each of the three commands in the sequence can be in the 8-bit binary format or the 2-character ASCII format.

Command	Hex Digit Represented
0 DIGIT	'0'
1 DIGIT	'1'
...	...
9 DIGIT	'9'
AUTOSTOP	'A'
STEP REV	'B'
STEP FWD	'C'
SEARCH	'D'
MULT REV	'E'
MULT FWD	'F'

For example, the data byte 'F9' (which is also the REJECT command) can be entered into RAM with the "PLAY", "MULT FWD", "9 DIGIT" command sequence. The data bytes below can only be entered into RAM using the 3-command sequence.

'FD'	(PLAY)
'F9'	(REJECT)
'DF'	(PGM)
'EF'	(END)

The following data bytes can be entered as two ASCII-hex characters, or with the 3-command sequence described above.

'FF'	(NO ENTRY)
'00'	
'30' through '39'	"0" through "9" ASCII
'B0' through 'B9'	"0" through "9" ASCII
'40' through '46'	"At-sign", A ... F
'C0' through 'C6'	"At-sign", A ... F
'0A' and '8A'	Line feed
'0C' and '8C'	Carriage return

4. RS232C INTERFACE SIGNALS

When the player is powered on and the initialization routine has ended, DSR and DCD are set active (high) by the player. Within 20 ms of RTS and DTR being detected by the player as "high," CTS is set high, indicating that the player is ready to accept input on the RS232 port.

The LD-V6000 can read repeatedly during one field-time, so that arguments (the digits 0 to 9) can be input, along with their associated command, in 1/60 of a second. Thus if you select a high enough baud rate, you can send the command 10000 SEARCH during one field-time.

Immediately after receiving any RS232 input, CTS is brought low until the input buffer is empty. When CTS is high, three inputs can be buffered, but a fourth input, arriving two soon, may cause an overrun error.

Although commands from an external computer should use the CTS "hand-shake" to avoid overruns, data transmitted from the player (FRAME NUMBER, STATUS, etc.) is output without regard to any of the control lines. The player will send the whole data stream (may be six bytes) as fast as the baud rate will allow.

Note: When an input error occurs, this version of the player may ignore the input. In future versions, the error handling may be different. For example, if a parity or framing error is detected, the player might substitute a "1-digit" command ('0F') for the input, to aid in identifying the error.

Pin No.	Symbol	Signal Name	IN/OUT of LD-V6000	Level (True)
1	FG	Frame GND		
2	TxD	Transmitted Data	In	L
3	RxD	Received Data	Out	L
4	RTS	Request to Send	In	H
5	CTS	Clear to Send	Out	H
6	DSR	Data Set Ready	Out	H
7	GND	Signal GND		
8	DCD	Data Carrier Detector	Out	H
20	DTR	Data Terminal Ready	In	H

Figure 1: LD-V6000 RS232C Port Signals

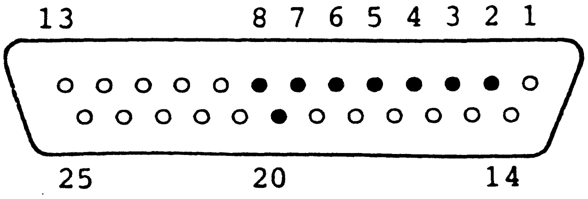


Figure 2: RS232 Connector

5. COMMUNICATION AND REAR SWITCH OPTIONS

A. OUTPUT OPTION: Two modes are available for output from the player, depending upon the setting of the Binary Output Flag.

OFF : ASCII hex mode: Any data byte is translated into two ASCII (hex) characters before output. CR (*ØD') and LF (*'ØA') are added after each transmission. This is the Power-on setting.

ON : Binary mode: Frame Number, Status, etc. will be output as 8-bit binary bytes of data (one or two bytes - see individual commands). There is no carriage return or line feed added.

B. REAR SWITCH OPTIONS: There are ten switches numbered 1 through 10, from left to right. Switch up (open) means "ON" or "1", and down is "OFF" or "0".

1) RS232C Baud Rate:

	S1	S2	S3
300	0	0	0
600	1	0	0
1200	0	1	0
2400	1	1	0
4800	0	0	1
9600	1	0	1

2) RS232C Data Format:

- o Stop Bits Transmitted, and Required with Input.
  S4 = 0 : 1 stop bit
  S4 = 1 : 2 stop bits

- o Parity Enable/Disable
  S5 = 0 : Parity Disable
  S5 = 1 : Parity Enable

- o Parity Odd/Even
  S6 = 0 : Even
  S6 = 1 : Odd

- o Data Length
  S7 = 0 : 7 bit
  S7 = 1 : 8 bit

3) Auto Play Startup:

This function is selected with rear switch number 9. When the player is powered on and this switch is "ON", the player will start playing automatically, without waiting for the "PLAY" command. If the drawer is open at Power-on, Auto-Play mode will begin when the drawer is closed.

4) Test Mode:

When the player is powered on and switch number 10 is "ON", the player will have the "TEST MODE" enabled. The five tests are:

Test 1 = Display the EPROM version number

Test 2 = Display current settings of Rear Switches

Test 3 = X3 MULTI FWD motion

Test 4 = X3 MULTI REV motion

Test 5 = Clear RAM to all '00'

After disc spin up, the individual test is selected with an argument (1 to 5) preceding the PROGRAM command. The player then executes the specified test. Upon receiving the END command, the player exits the specified test. An END command is always required when going from one test to another.

Note: The Test mode is provided for player service and troubleshooting, and may not be available in future players.

4800 Baud, 8 bit data,
No Parity, 1 Stop Bit.

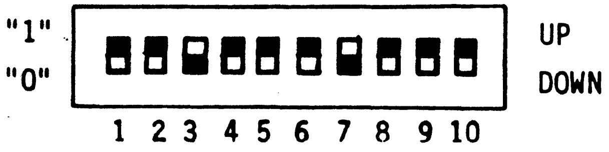


FIGURE 3 : Rear Switches Example

CHAPTER IV

RCU DATA ENTRY AND DISPLAY

1. GENERAL

All user programs may be manually entered into the RAM with the Remote Control Unit.* When the RCU (or ECI)is used, the result of entering each instruction may be viewed on the television display. Appendix A lists each mnemonic displayed in Programming Mode when a command is entered.

This section describes the nature of the character display, and use of the RCU for data or command entry to RAM.

2. CHARACTER DISPLAY

The videodisc player's character generator displays two lines of information on the television screen. The top line is used in non-programming modes to display frame numbers, elapsed play time, or user register numbers. In Programming Mode, this line displays the address of the "changeable" memory location.

In Manual Mode, the second line of the character generator displays the contents of user registers. In Programming Mode it displays stored instructions. The three right-most positions contain mnemonics representing the instruction in the "changeable" memory location. Consecutive RAM-resident program instructions may be viewed in the second line, as they are scrolled from right to left by repeated pressing of the PROGRAM button.

Alternatively, the contents of the "changeable" location, may be altered by entering an argument digit (one "numeric" button push), a command (one "standard command" button push), or hexadecimal data (by pressing PLAY followed by any two keys in the 0 to 9 or A to F range).

When any one of these complete entries is made to the "changeable" memory location, the completed entry shifts left in line two. The address in line one is automatically incremented, and indicates the next address. The current value of the next instruction is displayed in the right-most three positions of line 2.

* Programs may also be loaded from disc dumps (Chapter I) and via the External Computer Interface (Chapter III).

There are two display formats for a complete LD-V6000 program instruction. One format consists of an argument followed by either a mnemonic or hex representation of a command; the second format may be a mnemonic or a two-character hex command without a preceding argument. For both cases, complete instructions are delineated from each other in line 2 of the character display by the use of the comma (,) symbol. All mnemonics are displayed as one or two alphabetic characters.

In Programming Mode, the hex equivalent of a standard command may be entered (using the PLAY key followed by two entries 0 to 9 or A to F). The displayed representation in line 2 will be the two character hex code until entry is finished. Then it will become the standard mnemonic as it is shifted left.

3. RCU DATA AND COMMAND ENTRY

The RCU uses a Pioneer standard remote control I.C. chip, which has "button-press" codes that are quite different from the command codes of the LD-V6000. However, there is a functional equivalence achieved through an internal code conversion. Appendix B shows the RCU button-press codes.

In Figure 4, standard RCU button function definitions are shown labeled above each button. Most of these functions or numerics may be entered into RAM after the PROGRAM key has been pushed (i.e., when in Programming Mode).

Alternatively, in Programming Mode, the "Hex-Entry" method can be used, with the PLAY button and the alternate characters shown below the RCU buttons in Figure 4. Using this hex representation any data or command may be entered into RAM.

Example: The procedure to enter the VON command (*1B) into location 50 is as follows:

1. Press 5 0 PROGRAM — Begins Programming Mode, with the screen displaying the contents of location 50.

2. Press PLAY — Initiates "hex-entry" mode, and "*00" is displayed.

3. Press 1 — Generates a "1" - the first "hex" character.

4. Press STEP REV — Generates a "B" following the "1", thereby completing the VON instruction entry. The screen address indicator (line 1) is incremented to display location 51. The instruction *1B is displayed and shifted left, out of the three right-most positions of the second display line.

5. Press END — Returns to Manual Mode.

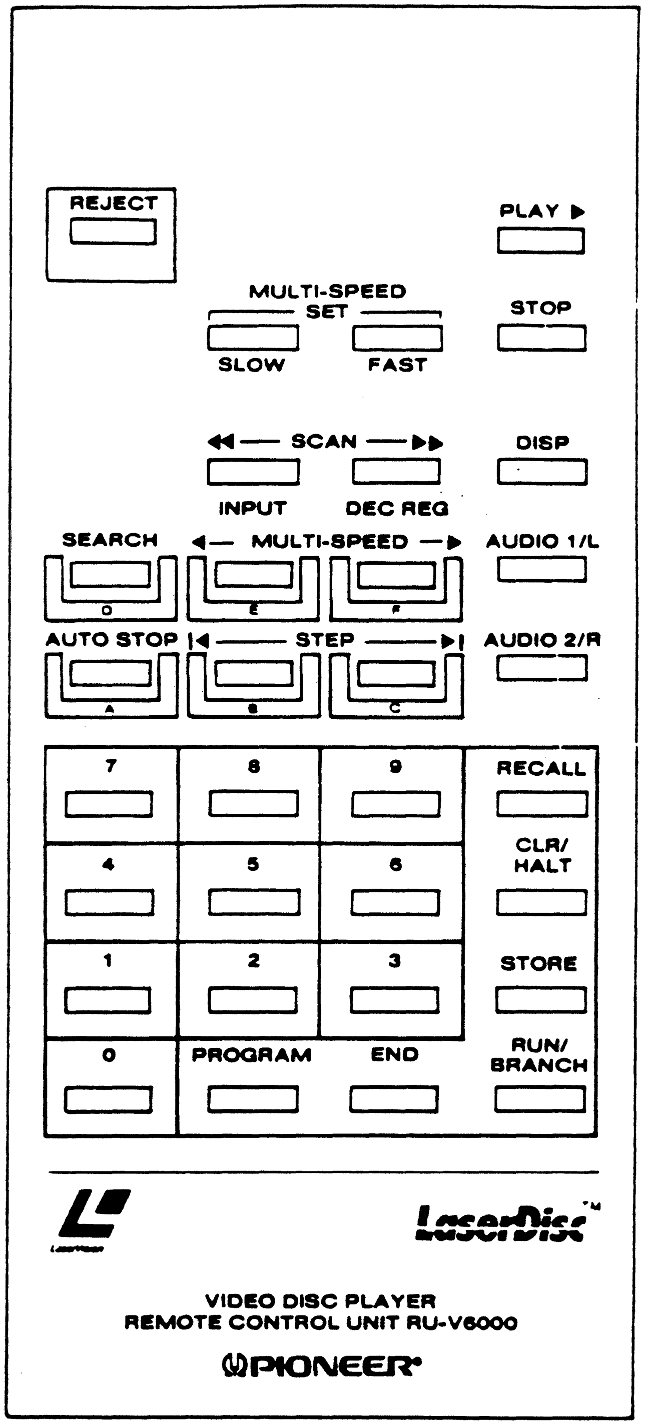


Figure 4. RCU Buttons

CHAPTER V.

REMOTE CONTROL INTERFACE

1. INTRODUCTION

This section provides information for LD-V6000 videodisc player users who wish to control the player using the 24-pin Amphenol (remote control) connector. This interface requires that the controller or host emulate the signal which is normally produced by the player's remote control unit (RCU). However, this port gives no feedback or output, so it is rarely used.

Actual hardware and software specifics are left to the system integrater and programmer. All intended uses should be thoroughly tested by the customer.

2. MECHANICAL CONSIDERATIONS

The LD-V6000 remote control port is an Amphenol 57-40240 jack located on the rear of the player. The mating connector (for example, 57-30240) must be supplied by the user. Pin usage is as follows:

AMPHENOL SERIES 57 CONNECTOR - 24 PIN

Pin #	Signal Name
1-3	NO CONNECTION (reserved for future use)
4	+11 VOLTS OUT, UNREGULATED, 50MA
5	VIDEO OUT, UNCLIPPED, NOT DROPOUT COMPENSATED
6	INTERNAL / EXTERNAL CONTROL, ... TO PLAYER
7	COMMAND DATA (LOW ACTIVE), ... TO PLAYER
8	NO CONNECTION (reserved for future use)
9	DISC BROADBAND F.M. SIGNAL OUT
10-12	NO CONNECTION (reserved for future use)
13-24	GROUND

When INTERNAL/EXTERNAL is low, the Remote Control Unit's infrared input is disabled. However, the front panel controls and the RS-232C port are still functional. It is recommended that shielded cables be used for connections.

3. ELECTRICAL CONSIDERATIONS

The Player's input port is TTL, with input pull-ups as shown, to a 74LS04. This input port will not accept the 40KHz-chopped levels that the RCU produces.

Figure 8 shows a schematic of the input circuitry. These circuits are located inside the player and the schematic is included to provide signal interfacing and loading information.

4. COMMAND DATA INPUT SIGNAL FORMAT

The input control signal must be structured to closely follow the format of a special transmitter chip (NEC uPD6102G) used in the player's Remote Control Unit. The player must receive a group of pulses which represents a player COMMAND WORD. When a valid COMMAND WORD is received, a COMMAND is executed by the player's microprocessor.

COMMAND WORD FORMAT ... Refer to Figure 5. Each Command Word is a series of thirty-four pulses, configured to contain one leader pulse and then 32 data bits. The first pulse (leader) is used for "start framing". The next 33 pulses bracket the 32 data bits. The pulse spacing determines whether the bit is a "Ø" or a "1". The last pulse is transmitted to close out the timing of the final data bit. A list of player RCU command codes is provided in Appendix B.

BIT TIMING ... The "Ø" bit and "1' bit are differentiated within a given command word by the pulse spacing (refer to Figure 5). A pulse repetition time of about 1 msec represents a "0" bit and a pulse timing of about 2 msec is a "1" bit. The pulse (T1) is about 500 microseconds. The timing shown in Figure 6 is very close to that produced by the Remote Control Unit, where the timing values are all derived from a 500 KHz crystal. The decoding algorithm in the player is reasonably tolerant and usually accepts a range of values wider than those indicated. Table 3 details the RCU signal timing and indicates a range of timing values that could be used when constructing a controller.

The first two bytes of the command data are always 'A8' and '57', with the least significant bit sent first (00010101 11101010). The next eight bits are the command code, from Appendix B, least significant bit first. The last eight bits are the complement of the command code, least significant bit first. For example, the REJECT command ('9E' hex) would be 'A8' '57' '9E' '61'. Thus, the 32 data bits for REJECT are (00010101 11101010 01111001 10000110).

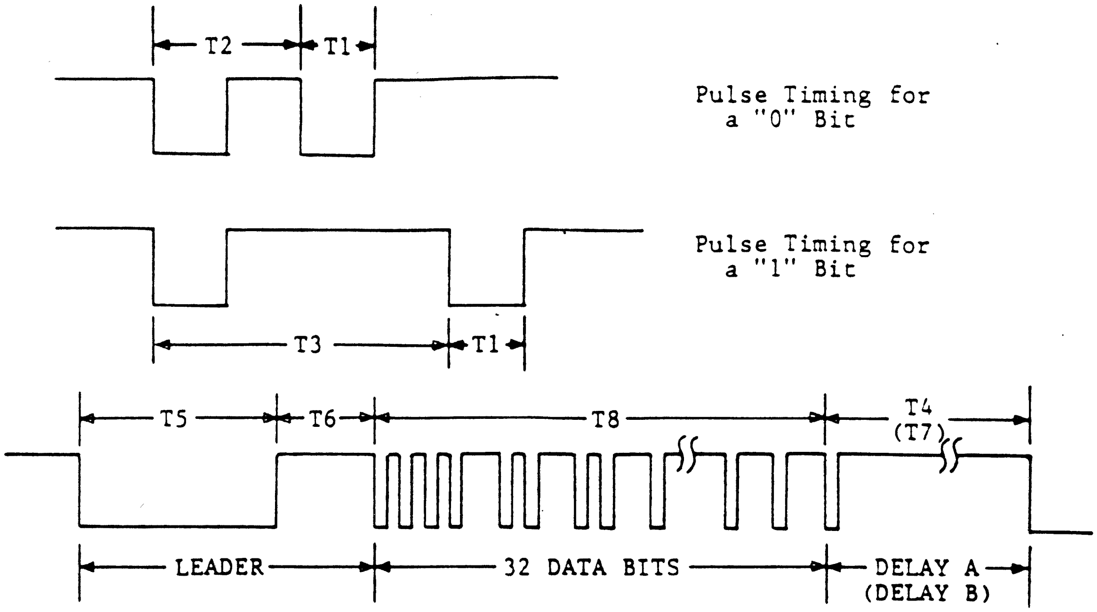


Figure 5: Command Word Timing

	MIN	RCU	MAX	EXAMPLE
T1: Pulse Width	0.48	0.51	0.54	0.50
T2: "0" Bit Time	0.97	1.02	1.07	1.00
T3: "1" Bit Time	1.95	2.05	2.15	2.00
T4: Delay A	19.0	30.0	40.0	20.0
T5: Leader Down	7.80	8.20	8.60	8.00
T6: Leader Up	3.90	4.10	4.30	4.00
T7: Delay B	60.0	--	--	60.0
T8: Data Time	46.7	49.1	51.5	48.0

Command Word Time = 16*(T2+T3)+T5+T6

Figure 6: Command Word Timing Values (Milliseconds)

COMMAND FORMATS ... Before a valid command is recognized, at least one valid command word must be received. The command word's data time is about 48 msec, depending upon the timing selected for the leader and the "1" and "0" data bits. The time does not depend upon the command code, since the data always contains 16 "1" bits and 16 "0" bits. A leader (about 12 msec) and a "Delay A" (about 20 msec) must be added between commands. Thus, the timing from the start of one command word to the beginning of the next is about 80 msec. A "command word" is the leader plus the 32-bit command code. A single "command" is typicaly one or more identical sequential command words, separated by "Delay A"'s.

When the player receives a string of command words (one command word every 80 to 100 msec) which are identical, the entire string is treated as a single command or one entry. This corresponds to a user pressing and holding down a key on the Remote Control Unit. In order to punctuate or separate sequential identical commands, a "Delay B" of about 60 msec is necessary between command words. This delay allows the receiving algorithm enough time to reset.

INSTRUCTION TIMING ... Figure 7 shows examples of three methods of entering multiple instructions into the player. In these examples the command word timing is 60 msec. A "Delay A" of 20 msec (T4) or a "Delay B" of 60 mec (T7) is used between command words.

In Figure 7A two strings of command words are input to the player. In this example each word contains the data bits that represent "STEP FORWARD". Since the delay between command words is only 20 msec, the player will treat each group of command words as a single command entry and cause the player's servo mechanisms to advance two frames.

Figure 7B shows a more efficient method of making individual entries of the same instruction. In this example each command word is followed by the 60 msec "Delay B." Thus, the player will treat each of the command words as a separate valid instruction and advance four frames. This method allows for about eight instructions per second (one every 120 msec).

Figure 7C shows that instructions may be crowded together, one command word (and a "Delay A") right after another, if the command words are different. It also shows that the "Ø"'s in "1200" must be separated with a "Delay B" in order to be distinguished from "120". This is the fastest command entry mode, but requires a bit more intelligence in the controller. Thus, the fastest instruction rate is about twelve per second (80 msec. per command word).

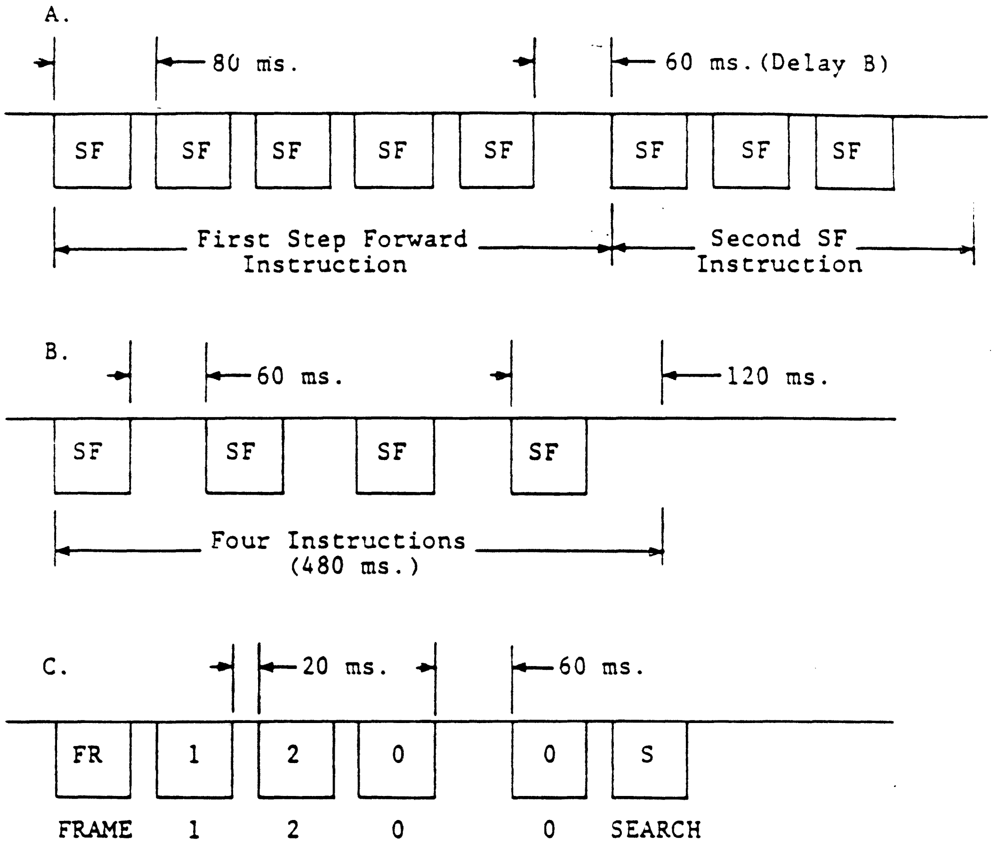


Figure 7: Instruction Formats and Timing

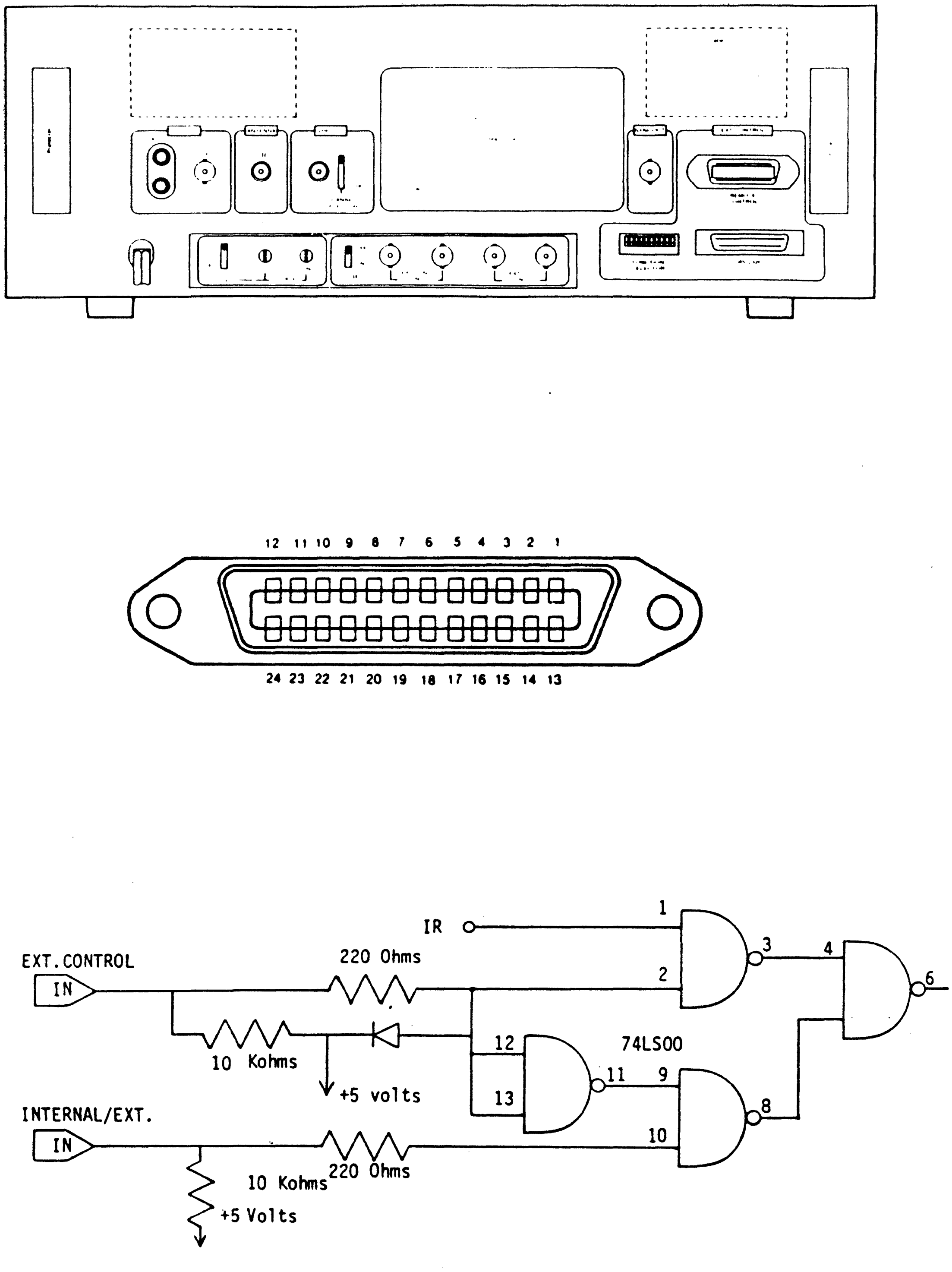


Figure 8: Remote Interface

APPENDIX A

SUMMARY OF PLAYER COMMANDS

		COMMAND					TEXT
COMMAND[1]	CODING	CODE		EQUIVALENT	DISPLAYED		PAGE
NAME	MNEMONIC	HEX	DEC	RCU BUTTON	MNEMONIC	ARGUMENT[2]	NO.
AUDIO1	A1	F4	244	AUDIO 1/L	A1	(Integer)	18
AUDIO2	A2	FC	252	AUDIO 2/R	A2	(Integer)	18
AUTOSTOP	AS	F3	243	AUTOSTOP	AS	(Frame No.)	21
BRANCH	BR	CF	207	BRANCH	BR	(Address)	36
.CX	CX	EE	238		CX	(Integer)	17
DEC REG	DR	F0	240	DEC REG	DR	Register No.	32
DISPLAY	DI	F1	241	FRM DSP	DI	(Integer)	22
END (3)	-	EF	239	END	-		14
.FS	FS	EC	236	SPEED SET(F)	FS	Fast Speed	23
HALT	H	BF	191	HALT	H		15
INPUT	IN	F8	248	INPUT	IN	Integer	34
LOAD	L	CC	204		L		16
.MULTI FWD	MF	F2	242	MULTI-FWD	MF	Frame No.	26
.MULTI REV	MR	FA	250	MULTI-REV	MR	Frame No.	26
NE	NE	FF	255		NE		37
PLAY	P	FD	253		P		24
PROGRAM(3)	-	DF	223	PROGRAM	-	(Address)	14
RECALL	RC	7F	127	RECALL	RC	(Register No.)	31
REJECT	RJ	F9	249		RJ		37
RUN(3)	-	CF	207	RUN	-	(Address)	15
SCAN FWD(3)	-	F0	240	SCAN FWD	-		30
SCAN REV(3)	-	F8	248	SCAN REV	-		30
SEARCH	SC	F7	247	SEARCH	SC	(Frame No.)	25
.SS	SS	ED	237	SPEED SET(S)	SS	Slow Speed	23
STEP FWD	SF	F6	246	STEP FWD	SF		27
STEP REV	SR	FE	254	STEP REV	SR		27
STOP	W	FB	251	STOP	W		28
STORE	ST	F5	245	STORE	ST	(Integer)	33
VOFF	VFF	1C	28		*1C		29
VON	VON	1B	27		*1B		29
WAIT	W	FB	251	STOP	W	Integer	28
AFF	AFF	A0	160		*A0		19
AFI	AFI	A5	165		*A5		19
AFN	AFN	A1	161		*A1		19
AFT	AFT	A4	164		*A4		19
AIF	AIF	AA	170		*AA		19
AIN	AIN	AB	171		*AB		19
AIT	AIT	AE	174		*AE		19
ANF	ANF	A2	162		*A2		19
ANI	ANI	A7	167		*A7		19
ANN	ANN	A3	163		*A3		19
ANT	ANT	A6	166		*A6		19
ATF	ATF	A8	168		*A8		19
ATI	ATI	AD	173		*AD		19
ATN	ATN	A9	169		*A9		19
ATT	ATT	AC	172		*AC		19

APPENDIX A

SUMMARY OF PLAYER COMMANDS
(Concluded)

		COMMAND					TEXT
COMMAND[1]	CODING	CODE		EQUIVALENT	DISPLAYED		PAGE
ABBREVIATION	MNEMONIC	HEX	DEC	RCU BUTTON	MNEMONIC	ARGUMENT[2]	NO.
	EF4	E4	228		*E4		38
ASCII OUT	EF5	E5	229		*E5		38
	EF6	E6	230		*E6		38
BIN OUT	EF7	E7	231		*E7		38
CG ON	EF0	E0	224		*E0		38
CG OFF	EF1	E1	225		*E1		38
	EF2	E2	226		*E2		38
	EF3	E3	227		*E3		38
DEC TP	DTP	DA	218		*DA		43
INC TP	ITP	D9	217		*D9		43
SET TP	STP	E8	232		*E8	Integer	42
TRAN MEM	TM	DC	220		*DC		44
INC & TR	ITM	DD	221		*DD		44
DEC & TR	DTM	DE	222		*DE		44
TRAN F#	TFN	D3	211		*D3		40
TRAN PA	TPA	DB	219		*DB		40
TRAN ST	TS	D4	212		*D4		41
0	0	3F	63	0	0		
1	1	0F	15	1	1		
2	2	8F	143	2	2		
3	3	4F	79	3	3		
4	4	2F	47	4	4		
5	5	AF	175	5	5		
6	6	6F	111	6	6		
7	7	1F	31	7	7		
8	8	9F	159	8	8		
9	9	5F	95	9	9		

Notes:

(1) The commands prefaced by a dot are unique to the LD-V6000 player.

(2) Arguments in parentheses are optional.

(3) Not a programmable function.

APPENDIX B

RCU COMMAND CODES

	'8X'	'9X'	'AX'	'BX'	'CX'	'DX'
'X0'	"0"	DISP			PROGRAM	
'X1'	"1"	SCAN REV			END	
'X2'	"2"	SCAN FWD			RUN	
'X3'	"3"				STORE	
'X4'	"4"	SLOW SET			CLEAR	
'X5'	"5"	FAST SET			RECALL	
'X6'	"6"	AUDIO 1				
'X7'	"7"	AUDIO 2				
'X8'	"8"					
'X9'	"9"					
'XA'	AUTOSTOP	CX				
'XB'	STEP REV					
'XC'	STEP FWD	STOP				
'XD'	SEARCH	PLAY				
'XE'	MULTI REV	REJECT				
'XF'	MULTI FWD					