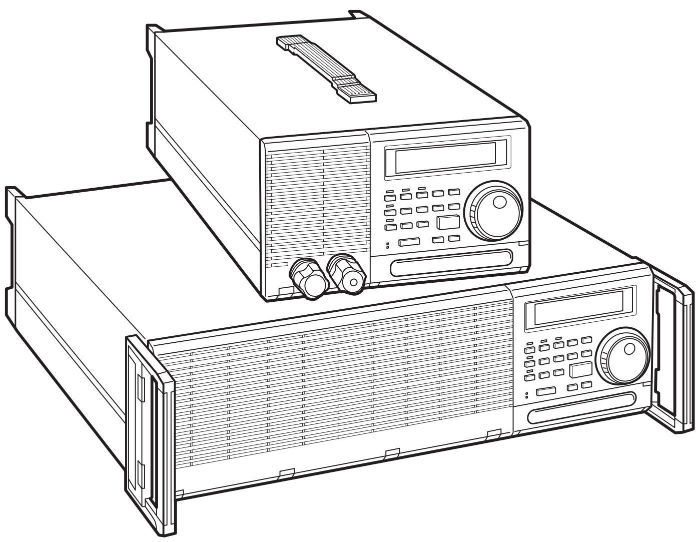
Part No. Z1-002-512, IB003022
Sep. 2004

SEQUENCE OPERATION GUIDEBOOK

ELECTRONIC LOAD

PLZ-3W Series
PLZ-3WH Series



KIKUSUI

Use of Guidebook

If you find any incorrectly arranged or missing pages in this guidebook, they will be replaced. If the guidebook it gets lost or soiled, a new copy can be provided for a fee. In either case, please contact Kikusui distributor/agent, and provide the “Kikusui Part No.” given on the cover.

This guidebook has been prepared with the utmost care; however, if you have any questions, or note any errors or omissions, please contact Kikusui distributor/agent.

Introduction

This book has been written to aid you understanding the Sequence Operation of PLZ-3W/PLZ-3WH Electronic Load. Unless you are well versed in computers, programming may seem to you to be very difficult and trouble-some. Programming for Sequence Operation of PLZ-3W/PLZ-3WH is not so difficult. This book introduces examples of programming for training. As you practice these, you will become familiar with programming of Sequence Operation. Let's try and practice, and you will find that the programming is not so difficult as you expected.

For PLZ-3WH Series User

As this guidebook describes the sequence function taking the PLZ-3W Series an example, real indication of the PLZ-3WH Series may be different from example of display such as decimal point position or number of digit. And also, current value which is greater than the rated current of the PLZ-3WH Series may be set. In this case, set the value for your PLZ-3WH. You can read "PLZ-3W" as "PLZ-3WH" in this guide book.PLZ-3WH

Chapters

This book is composed of the following chapters:

TABLE OF CONTENTS

Chapter 1

SEQUENCE FUNCTION

This chapter briefly explains the Sequence Function. When you use the Sequence Function, you may refer to this chapter as well as to the operation manual.

Contents Page

The term "Sequence Function" as used here menas a function that the PLZ-3W automatically exhibits a certain pattern of load effect by drawing a certain pattern of input current by executing sequentially the programmed steps of operation.

There are two major modes of Sequence Operation, namely, Normal Speed Sequence and Fast Speed Sequence, whose typical Sequence Patterns are shown in Figures 1 and 2, respectively.

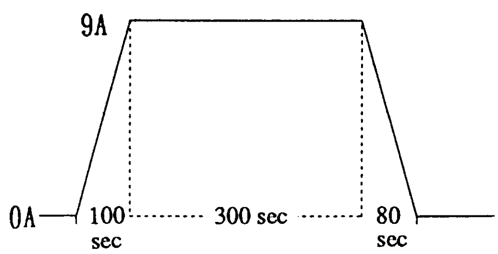


Figure 1. Normal Speed Sequence

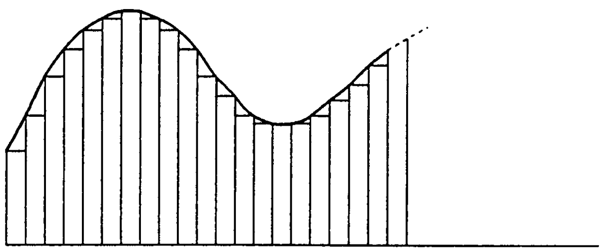


Figure 2. Fast Speed Sequence

The *Normal Speed Sequence* is to simulate a slowly-changing longer-period pattern. It is especially effective to simulate a pattern with slowly-changing linear slopes. A typical simple pattern is shown in Figure 1, where the current slowly increases from 0A to 9A in 100 seconds, remains at 9A for 300 seconds (for various evaluations to be done within this period, for example), and slowly decreases from 9A to 0A in 80 seconds.

The *Fast Speed Sequence* is to simulate a rapidly-changing pattern, by using up to 1024 bars of uniform durations of down to 100 μ sec, as shown in Figure 2.

Individual modes of Sequence Operation are described in more detail on subsequent pages.

1.1 Normal Speed Sequence

Assume that we need an overall Sequence Pattern as shown in Figure 3.

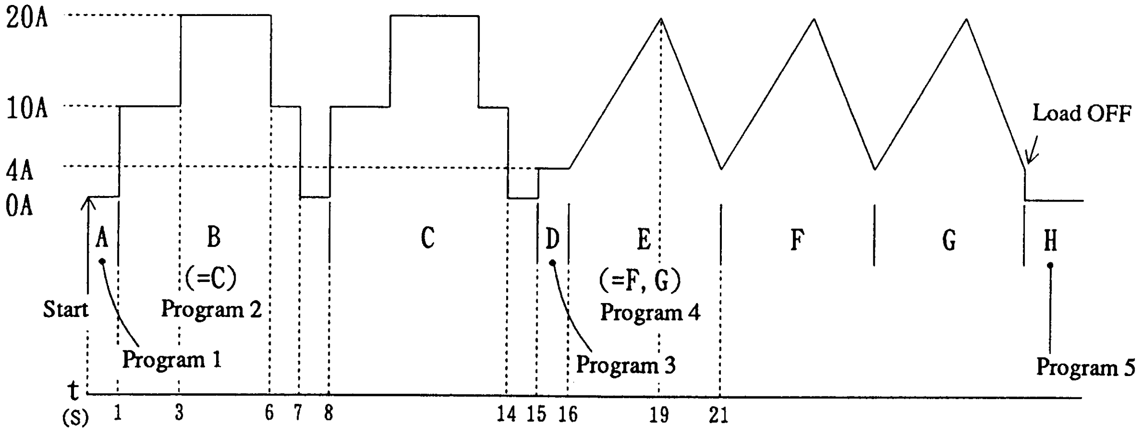


Figure 3. An Example of Sequence Pattern

1.1.1 Analyzing a Sequence Pattern for Programming

To analyze the overall Sequence Pattern shown in Figure 3, we may divide it into patterns A through H. We find that patterns B and C are identical and patterns E, F, and G also are identical. We find also that the overall Sequence Pattern involves only five patterns, namely, patterns A, B, D, E, and H. For programming, we denote these patterns with ***program*** numbers as follows:

Pattern A : Program 1
Pattern B : Program 2
Pattern C : Program 3
Pattern D : Program 4
Pattern E : Program 5

Now, let us analyze individual patterns. For the Normal Speed Sequence, a current value and its duration can be programmed as a pair of data items. For example, Program 2 (Pattern B) can be analyzed into four pairs of data items as follows:

1 : 10A for 2 seconds
2 : 20A for 3 seconds
3 : 10A for 1 second
4 : 0A for 1 second

Each pair of data items is referred to as "***Step.***" Thus, we can say "Program 2 (Pattern B) consists of four Steps."

Program 4 (Pattern E) is for a ramp change waveform (refer to the "column"). It consists of two Steps in which the current changes linearly as follows:

Step 1 : Increases from 4A to 20A, in 3 sec.

Step 2 : Decreases from 20A to 4A, in 2 sec.

Each of Programs 1, 3, and 5 (Patterns A, D, and H) consists of a single Step as follows:

Program 1 : Step 1 (0A for 1 sec.)

Program 3 : Step 1 (4A for 1 sec.)

Program 5 : Step 1 (0A for load off)

Here, let us impart you with a secret for effective use of the Sequence Function. The role of smaller programs such as Programs 1, 3, and 5 is to set initial values at the beginnings and ends of larger programs such as Programs 2 and 4. The Electronic Load involves many other functions than the Sequence Function, and settings of these functions are intricately related. Due to this, unless the programs are accurately written, the resultant Sequence Operation may quite differ from what you have expected. Be sure to set accurately the initial values, especially before you become fully familiar with and well versed in programming for Sequence Operation.

<<Column>> A new way of programming a ramp waveform

To program a ramp waveform with a conventional Sequence Function, you will have to write a large number of programs simulating individual steps--such as 4A for 0.375 sec., 6A for 0.375 sec., and so forth. With the new sequence function offered here, you can simulate a ramp change with a single step of Program, for a stepwise waveform with a resolution of an order of one millisecond.

For example, even Pattern E can be written as a single Step of Program. Pattern E is for current increase from 4A to 20A in 3 seconds — which is to be expressed with 3000 stepwise changes — with 5.4mA increase per step (although the number of Steps actually is less than this because the resolution of the input current is not that fine.)

1.1.2 Sequence and Chain

The "*Sequence*" function is to specify a particular flow of Programs. For example, an input current pattern as shown in Figure 3 cannot be obtained by executing Programs 1 through 5 straightforward one by one. To simulate the pattern, Program 2 must be repeated twice consecutively, and Program 4 three times.

For a Sequence Operation, you must assign a Sequence to each of the Programs. You must specify the starting Program number, the number of repetitions of the Program, and the next Program to be executed. For the sequence pattern shown in Figure 3 for example, you should specify the following:

Sequence 1 : Run Program 1 once, and invoke Sequence 2

Sequence 2 : Run Program 2 twice, and invoke Sequence 3

Sequence 3 : Run Program 3 once, and invoke Sequence 4

Sequence 4 : Run Program 4 thrice, and invoke Program 5

The "***Chain***" function is to specify the next Sequence to be invoked when the repetitions of the current Program is over. In the above example, if you specify 2 for the chained Sequence item in Sequence 1, Sequence 2 will be invoked upon completion of Sequence 1.

However, the situation differs a little for Sequence 4. If the above situation were applied directly, it would be written as "Run Program 4 thrice, and invoke Sequence 5." Instead, it is written as "Run Program 4 thrice, and invoke Program 5." The reason for this will be discussed in the next section.

1.1.3 End Program

Assume that you want to halt, for any reason, the sequence operation in progress. In this occasion, you will usually want to reduce the input current to zero and to turn off the input for the sake of safety (although in some particular cases you may not want to reduce the current to zero).

Of the example shown in Figure 3, Program 5 which is to be executed last is for "current 0A, and load off." You may utilize this Program to halt the Sequence Operation in progress. Thus, this Program is referred to as "***End Program***" and can be used in a particular manner as shown in Figure 4.

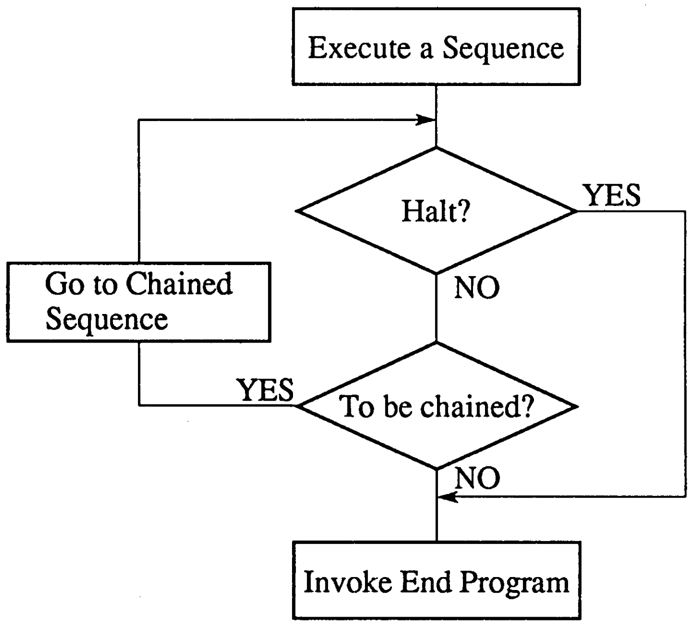


Figure 4. Relationship Between Chained Sequence and End Program

1.2 Fast Speed Sequence

The Fast Speed Sequence allows you to simulate a waveform by rapidly changing the setup values, as shown in Figure 2 for example.

Assume here that you want to simulate a 50Hz sine wave with the Fast Speed Sequence. One period of the 50Hz sine wave is 20 milliseconds. Dividing one period into 100 points of uniform intervals, each interval will be 200 microseconds. This means that, with the Fast Speed Sequence, you can simulate a sine wave with a stepwise waveform of 100 steps (set values) per one period, with a duration of 200 microseconds per step as shown in Figure 5.

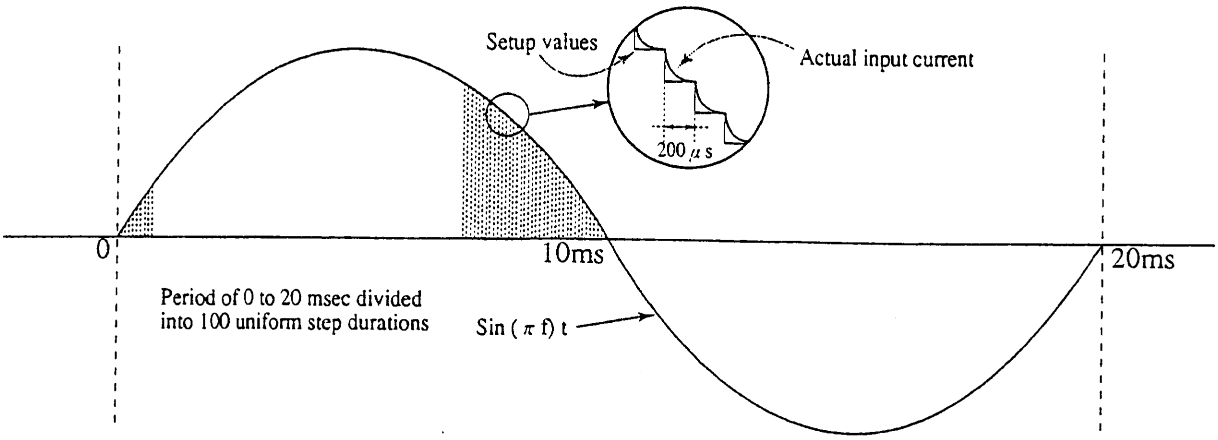


Figure 5. Approximation by Fast Speed Sequence

The Fast Speed Sequence offers the maximum resolution of 100 μsec. You may suppose that "'100 μsec' sounds to be rather slow." As a matter of fact, however, this is quite fast in the field of instruments which deal with large currents,

The programming method for the Fast Speed Sequence basically is identical with that for the Normal Speed Sequence except the following: In the Fast Speed Sequence, all step durations are uniform within the sequence. That is, you can specify only one duration — you cannot specify different durations for individual steps. You can specify one duration when editing the sequence to control the Program flow. Another difference is that no ramp change function is available in the Fast Speed Sequence.

Programming can be done by entering data from the front panel of the instrument. More efficiently, however, you may calculate data with a personal computer and transfer the calculated data to the instrument through GPIB or RS-232C.

1.3 Structure of a Sequence

So far we used terms ***Step***, ***Program***, and ***Sequence*** without defining their meanings as used in this book. To grasp their meanings, please refer to Figure 8 "Structure of a Sequence File."

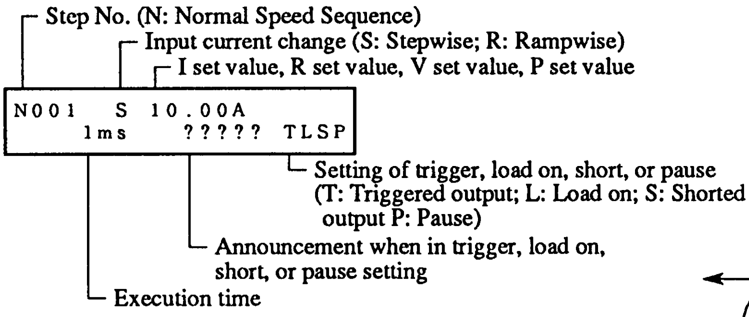


Figure 6. Display of ***Step*** Edit for Normal Speed Sequence

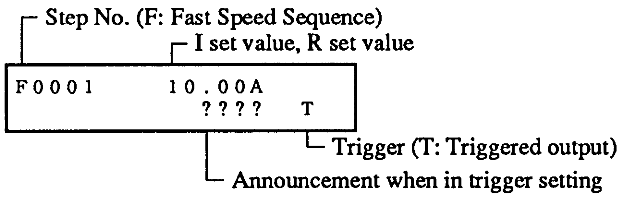


Figure 7. Display of ***Step*** Edit for Fast Speed Sequence

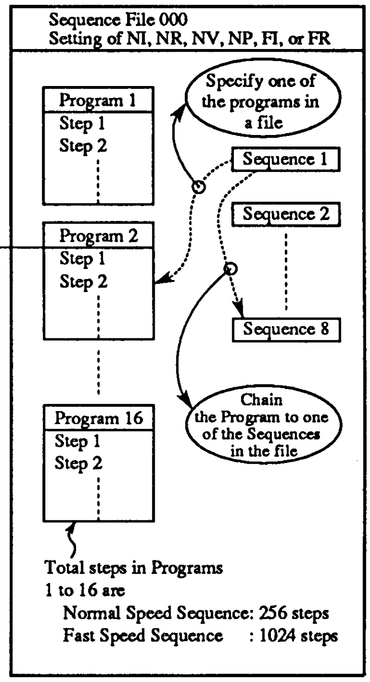


Figure 8. Structure of a Sequence File

1.3.1 Sequence File

The "NI, NR, NV, NP, FI, and FR" in the Sequence File column in Figure 8 denote Sequence Modes. N stands for Normal Speed Sequence and F for Fast Speed Sequence of mode. I stands for current, R for resistance, V for voltage, and P for power for the variable item.

Note that, with the PLZ-3W Instrument, you can specify only one of these variables. For example, assume that you want to select the NI mode. In this case, the I value alone is variable. You must set the V value and P value at certain fixed values before starting the sequence operation.

You must specify these modes for respective sequence files when creating the files. In the case of the above example, you must specify a "sequence file of NI mode."

A sequence file is composed of ***Steps***, ***Programs*** and ***Sequences***.

1.3.2 Step

Each of the minimal elements of a sequence pattern is called a "*Step*." See the examples of Program Editing Displays of Normal and Fast Speed Sequences shown in Figures 6 and 7. For the Normal Speed Sequence, you can specify the execution time for each Step. Thus, the section where the input current remains constant or changes linearly can be handled as one Step. For the Fast Speed Sequence, however, you cannot specify durations for individual Steps. For the Fast Speed Sequence, the durations of all Steps must be uniform.

For the Normal Speed Sequence, the maximum available number of Steps is 256; that for the Fast Speed Sequence is 1024.

1.3.3 Program

When analyzing a Sequence Pattern, you may regard a section consisting of a certain number of repetitions of the same waveform as a section for one ***Program***. Each Program can be a cluster of ***Steps***. Each Sequence File can accommodate up to sixteen Programs.

1.3.4 Sequence

To attain the objective current waveform, you must specify the order of execution of ***Programs*** and how may times a Program is to be repeated. What specify these is what we call here "***Sequence***." Each Sequence file can accommodate up to eight Sequences.

The above have been explained with examples earlier in this book. Please review them once more.

1.3.5 Edit Memory

When you edit ***Steps***, ***Programs*** and ***Sequences***, you are in access to the ***Edit Memory*** as shown in the conceptual diagram of Figure 9. The ***Edit Memory*** is backed up by a battery and data stored in it is not lost even when the instrument power is turned off. Due to this, the instrument can run as if it had one file within itself. Even if you have turned off the instrument power after a program is completed or halted, you can run the instrument again without requiring to load a file.

The contents of the Edit Memory remain unaltered unless you modify it by re-edit.

NOTE

*Note that the **Edit Memory** may be referred to as "Sequence Execution Memory" elsewhere in this book and in the operation manual.*

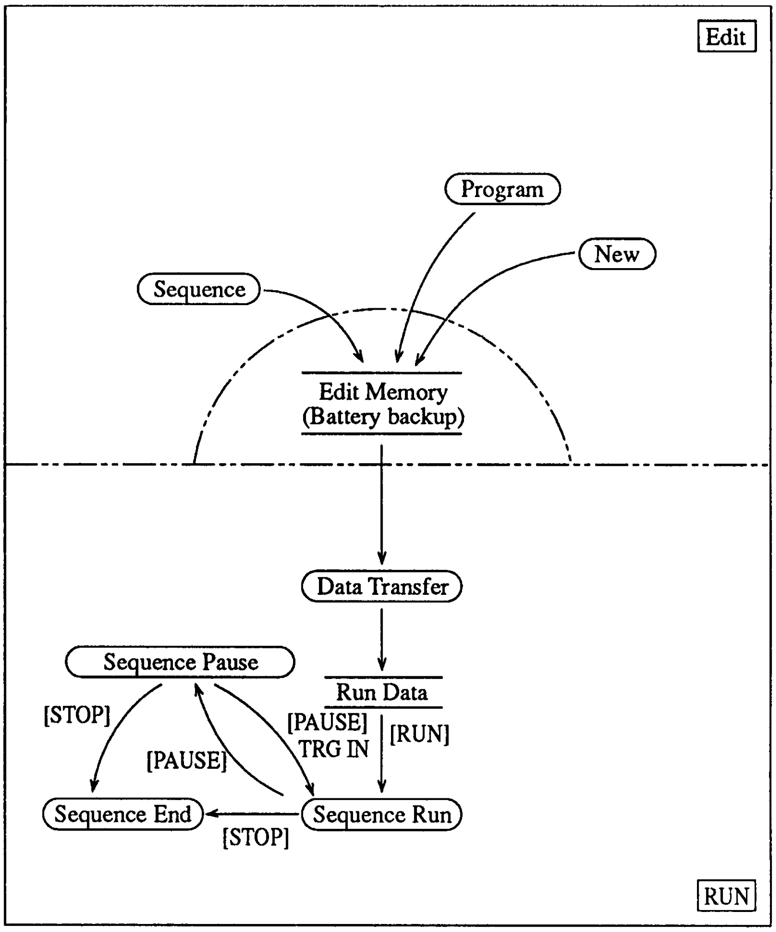


Figure 9. Conceptual Diagram of Sequence Operation

Chapter 2

PROCEDURE FOR CREATING A SEQUENCE FILE

This chapter covers the matters you should confirm before start creating a Sequence File, and introduces a flowchart which clearly indicates the Sequence File creation procedure.

Contents Page

2.1 Before Creating a Sequence File

- *Root State*

```
0.00A   0.00V    0.0W
--ISET  0.00A H --
```

The above is an example of root display that appears when the instrument is in the ***root state*** — which the instrument assumes immediately after you have turned its power on. The top row indicates the measured values of current, voltage and power — and they may differ from those shown here. The bottom row also may differ from those shown here.

The instrument assumes this state immediately after you have turned its power on. It assumes this mode before entering a sequence mode. You can reset the instrument to this mode by pressing the [ESC] key for a required number of times.

- Check the Contents of Edit Memory!
  Check the contents of Edit Memory before erasing them.

2.2 Flow of Procedure for Creating a Sequence File

Here we introduce a Sequence File creating flowchart (Figure 11) and a sequence function menu hierarchy chart (Figure 10). These charts provide you with useful information on how to create a Sequence File.

In the subsequent chapters where you will learn how to create Sequences by examples we offer, you will be often referred to these charts. Moreover, these charts will prove to be useful references when you will be creating your own Sequences for actual use.

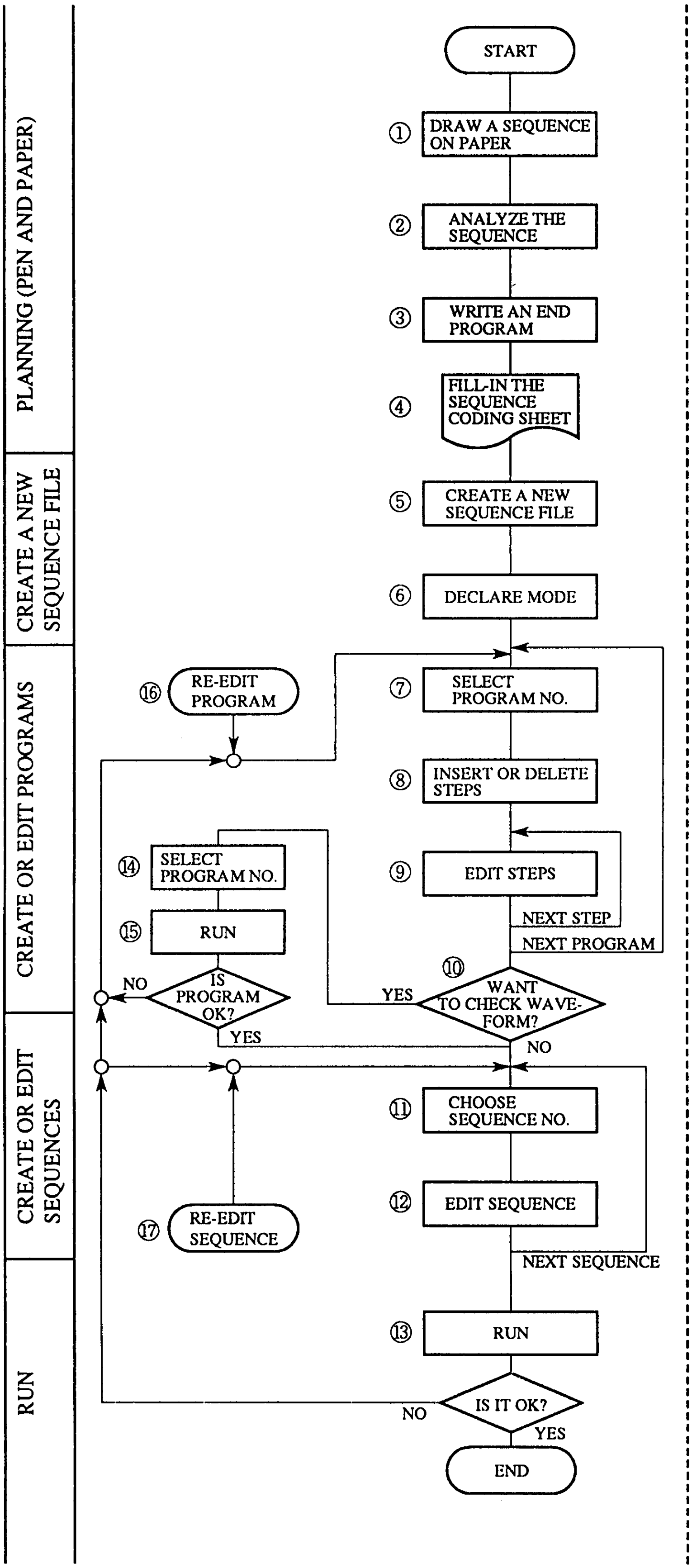
SEQUENCE MENU

- 1 : Edit [Edit Menu]
  - 1 : Edit Program [Modify, insert, or delete steps in edit memory]
    - 1 : Modify [Modify steps in edit memory]
    - 2 : Insert [Insert steps in edit memory]
    - 3 : Delete [Delete steps in edit memory]
  - 2 : Edit Sequence [Create or edit sequence parameters in edit memory]
  - 3 : New [Initialize the edit memory]
- 2 : Run[Run Menu] [Run the sequence in edit memory]

Figure 10. Hierarchy and Functions of Sequence Operation Menus

<<Column>> Selecting a Menu Item

A prompt ">" appears at the left end of one of the selectable menu items. You can move the prompt with the 【◁▷】 keys or JOG dial. By pressing the 【ENTER】 key, you can select the item indicated by the prompt.



① Draw the waveform you need, with pen and paper. Select a Normal or Speed Sequence mode (NI, NR, NV, NP, FI, or FR mode).

② Analyze the waveform with Steps and Programs, and specify a Sequence. If you have selected a Normal Speed Sequence mode, specify a time unit (msec., sec., minute, or hour) also.

③ Specify the states the Electronic Load should assume when the Sequence is over or halted.

④ Specify respective values for the waveform.

⑤ If there already is a Sequence in memory, be sure that you do not need it any longer and it may be deleted. Then, initialize memory by selecting "3:New" under the Edit Menu.

⑥ With the JOG dial, specify the mode you selected in Step ①. If you have selected a Normal Speed Sequence mode, specify a time also.

⑦ Choose "1:Edit Program" under the Edit Menu, and a Program display will appear. With the JOG dial, select a Program Number for the Program you want to edit. Press the 【ENTER】 key to call out the program.

⑧ If the program is a one to be newly created, since no program data has been entered yet, "end of step" alone will be displayed. Press the 【ENTER】 key, choose "2:Insert" under the Program Edit Menu, and then insert the required Steps. To delete a Step or Steps to re-edit a Program, choose "3:Delete." If you need no step insertion or deletion, select "1:Modify" and proceed to Step ⑨.

⑨ Enter the Step data values as you filled in the Sequence Coding sheet. To re-edit the program, change the Step valued. Advance to the next Step with the 【SHIFT】 + 【◁▷】 keys and change its values. If you want to create or edit the next program, press the 【ESC】 key to call out the Program Display and then repeat Steps ⑦ through ⑨.

⑩ You may run the program for test.

⑪ Return to the Edit Menu with the 【ESC】 key and choose "2:Edit Sequence." Choose a sequence number for the Sequence to be edited, with the JOG key.

⑫ Pressing the 【ENTER】 key, edit the Sequence. Be sure to provide an end program at the end of each Sequence.

⑬ Press the RUN key and choose a Sequence Number. As you press the RUN key again, the Sequence will run.

⑭ From the program Display, choose the Program Number of the Program you want to check the JOG dial.

⑮ Each time as you press the RUN key, the Program will run once.

⑯ Press the 【EDIT】 key to call out the Edit Menu. Choose "1:Edit Program."

⑰ Press the 【EDIT】 key to call out the Edit Menu. Choose "2:Edit Sequence."

Figure 11. Flowchart for Creating a Sequence File

Chapter 3

A PETIT PROGRAM FOR A SIMPLE PATTERN

You can create a petit program for a simple pattern and run it for trial, without requiring to complete any full-scale Sequence File.

Contents Page

In this chapter you will learn how to create a ***Program*** for a very simple Sequence pattern and how to run it. It will be of the Normal Speed Sequence mode only. Thus, you will acquire the basics of programming that will help you when you will be leaning more advanced programming in the subsequent chapters.

3.1 Analyzing the Petit Pattern

Assume a simple Sequence Pattern as shown in Figure 12.

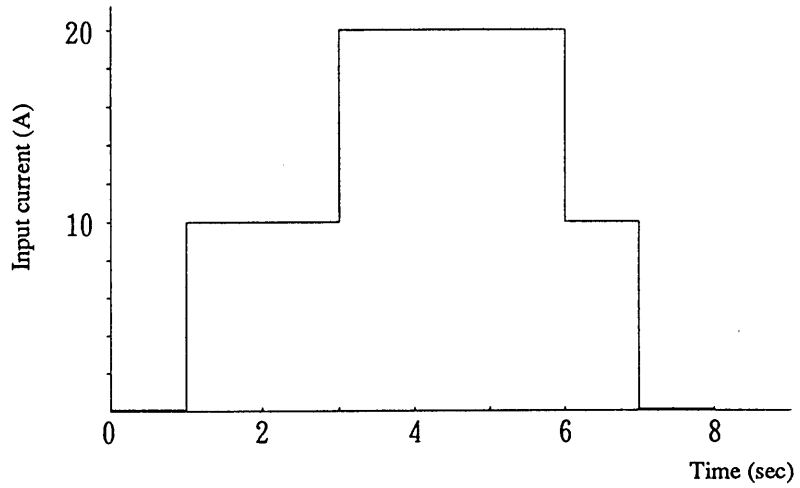


Figure 12. A Simple Sequence Pattern

The pattern shown in Figure 12 may be analyzed as shown in Figures 13 and 14.

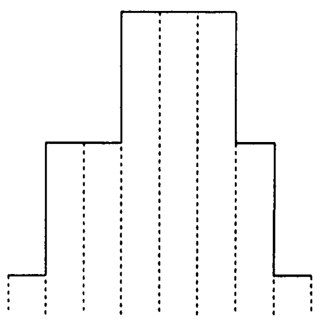


Figure 13. Analysis by Fixed Intervals (1 sec)

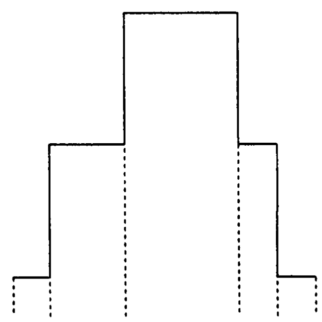


Figure 14. Analysis by Variable Duration

In Figure 13, the Sequence Pattern is analyzed by dividing it into one-second intervals. There are eight uniform intervals. Each interval is represented by a step. Thus, the pattern can be expressed with an 8-step program as shown in Table 1. The same concept — uniform time intervals — applies also to the Fast Speed Sequence mode of operation, although the 1-second intervals are too slow for a Fast Speed Sequence.

Table 1

Execution time of each step: 1 second

Step No.	Current
001	0A
002	10A
003	10A
004	20A
005	20A
006	20A
007	10A
008	0A

In Figure 14, the sequence pattern is analyzed being based on the input current change events. Each step consists of a current value and its duration. Thus, the Sequence Pattern consists of a lesser number of steps — namely, five steps in this example.

Table 2

Step No.	Current and Duration
001	0A for 1 sec
002	10A for 2 sec (stepwise change)
003	20A for 3 sec (stepwise change)
004	10A for 1 sec (stepwise change)
005	0A for 1 sec (stepwise change)

For a drill, let's fill in the program columns of Sequence Coding Sheet. The sheet will be as shown in Table 3. Since the pattern has no rampwise changes, enter S (stepwise change) in the S/R column.

Table 3. An Example of Sequence Coding Sheet

Program No. 01

Step No.	S/R	Specified Value	Time	Trig	Load	Pause	Short	Comment
001	S	0A	1S	0	1	0	0	My Training
002	S	10A	2S	0	1	0	0	
003	S	20A	3S	0	1	0	0	
004	S	10A	1S	0	1	0	0	
005	S	0A	1S	0	1	0	0	

3.2 A Petit Normal Speed Sequence

3.2.1 Creating a New Sequence File

Now, let's start programming a petit pattern. First of all, we should create a sequence file. Press the [ESC] key many times until the Root Display appears. Then, proceed as follows:

1) Press the [SEQ] key to invoke the Sequence Menu.

```
>1: Edit
 2: Run
```

Select "1: Edit" with the [◁▷] keys and press the [ENTER] key to invoke the Edit Menu.

```
>1: Edit Program
 2: Edit Sequence
```

2) Select "3: New" (initialization of edit memory) by scrolling the display with the [◁▷] keys.

```
 2: Edit Sequence
>3: New
```

As you press the [ENTER] key, the following message will appear prompting you to acknowledge the execution of the command since the existing contents of the memory will be lost if the command is executed.

```
Create New Swquence
        Sure?
```

3) To acknowledge, press the [ENTER] key.

```
Mode-> NI
Unit :msec
```

4) Next, choose one of the Sequence Modes (NI, NR, NV, NP, FI and FP) with the JOG dial. As you turn the dial, the modes will scroll NI → NR → and so forth. Select "NI" here in this example. Do not press the 【ENTER】 key yet.

```
Mode-> NI
Unit : msec
```

5) Move the prompt (->) to the next item (time unit for a step) with the 【◁▷】 keys. As you turn the JOG dial, the time units will scroll as follows:

```
Mode : NI
Unit-> msec
```

msec
↓
sec
↓
minute second
↓
hour minute

Select "msec" here for this example. Let "msec" displayed and press the 【ENTER】 key.

6) The following message will appear for 2 seconds and then the Edit Menu will resume.

```
Completed
```

3.2.2 Creating a Petit Program

You have created a new Sequence File by the above procedure. What you should do now is to create a new program. Assume here a program comprised of five steps as shown in Table 2 and to call it "Program 01."

The Sequence File has just been initialized. The number of Steps with which Program 01 consists of is not specified yet. When in this state, you cannot write even a single step in Program 01. Before writing Steps, you must secure a memory area for a required number of Steps (five steps in this example). To secure a memory area, proceed as follows:

Securing a Step Area

```
>1: Edit Program
 2: Edit Sequence
```

1) Choose "1: Edit Program," and the following display will appear.

```
Program:01        :N1
      000 Step [T000]
```

Number "01" of the top row denotes the Program Number. Number "000" of the bottom row denotes the number of Steps currently assigned to Program 01 — it is 000 because the program area currently is zero. Item "[T000]" denotes the total number of Steps of the entire file.

To view the setting of each Step, press the 【ENTER】 key.

```
N001
   END of Step
```

2) As Program 01 has no Steps at all yet, the only message displayed is "END of Step." This message means that "this is the last Steps and there are no other Steps at all." Here, you have to secure a program area for the required number of Steps as follows:

Press the [ENTER] key to invoke the Program Edit Menu.

```
N001
    1:Modify
```

```

   >2:Insert
```

Select "2: Insert" with the [◁▷] keys and press the [ENTER] key.

3) You will be prompted to enter the number of Steps to be inserted.

```
Insert:001
 How Many steps?   1
```

Scroll the number to 5 with the JOG dial and press the [ENTER] key.

```

 Completed
```

An area for five Steps will be secured and the Edit Menu will resume.

```
N001 [S] 0.00A
    >2: Insert
```

Thus, a program area for 5 Steps has been secured.

3.2.3 Editing Steps

1) Invoke the Edit Menu.

```
N001  S  0.00A
   >1:Modify
```

Select "1: Modify" with the [◁▷] keys and press the [ENTER] key.

2) The following display will appear, indicating that you can modify the contents of Steps.

```
N 0 0 1  [S]  0 . 0 0 A
        1 m s              · L · ·
```

You can modify the item indicated by the cursor. You can move the cursor with the [◁▷] keys. Thus, you can edit the seven items.

3) Number "N001" of the above display is a Step Number. To change it to another Step Number, use the [SHIFT] + [◁▷] keys. Try it. You will see that the numbers scroll as N001, N001, ... N006. At N006, the following display will appear.

```
N 0 0 6
    E N D   o f   S t e p
```

This means that there are five Steps (N001 through N005) and there is no meaningful N006 Step.

4) Return the Step Number to N001 with the [SHIFT] + [◁▷] keys. The meanings of individual items of the display are as described below.

```
N 0 0 1   S     0 . 0 0 A
        1 m s              · L · ·
```

N001 : "N" stands for Normal Speed Sequence. "001" denotes the Step Number of the Step to be executed next when Program 01 runs. This number will change in the due order as Steps advances.

S (or R) : "S" stands for Stepwise current change (and "R" for Rampwise).

1ms : This item denotes the duration of each Step — 1 millisecond per Step in this example. You can change the duration with the JOG dial.

· L · · : The final four items mean the following, as indicated by cursor □.

```
        Trig  [·]L · ·
```

```
        Load  ·[L]· ·
```

```
        Short  · L[·]·
```

```
        Pause  · L ·[·]
```

Trig : When this Step is executed, a trigger signal is delivered via the Trigger Output terminal.

"T" ... A trigger signal is delivered.
" · " ... No trigger signal is delivered.

Load : When this Step is executed, the instrument is driven into the Load ON state or Load OFF state.

"L" ... Load ON
" · " ... Load OFF

Short : When this Step is executed, the instrument's Short function is brought into effect.

"S" ... Shorted
" · " ... Not shorted

Pause : Halts the execution of the current Step, holding the input current at the value that existed immediately before entering this Step.

"P" ... Paused
" · " ... Not paused

For further details of the above, refer to the operation manual.

5) Enter data for the five Steps as shown below. To advance Steps, use the [◁▷] keys. For settings of S/R, Trig, Short and Pause, use the JOG dial. To advance to the next Step, use the [SHIFT] + [◁▷] keys.

Step 001

```
N001 S   0.00A
.1000ms          ·L··
```

Step 002

```
N002 S  10.00A
 2000ms          ·L··
```

Step 003

```
N003 S  20.00A
 3000ms          ·L··
```

Step 004

```
N004 S  10.00A
 1000ms          ·L··
```

Step 005

```
N005 S   0.00A
 1000ms          ·L··
```

After the data entry is over, press the [ESC] key to return to the program display as shown blow. This display indicates that five Steps have been inserted.

```
Program:01        :N1
    005 Step   [T005]
```

Program 01 is complete by the above procedure. Now, you'd better check whether you have entered data correctly or not. For trial, press the [RUN] key. By pressing the [RUN] key when a program as shown in the above is displayed during Edit Program selected from the Edit Menu, you can run the program you are developing. Remember this convenient feature. Be sure to connect a constant-voltage power supply or other appropriate source before running the program for trial.

In this chapter, you have learned programming of a simple waveform pattern. In the subsequent chapters, you will learn programming of more sophisticated waveforms.

Chapter 4

NORMAL SPEED SEQUENCE FOR A COMPLEX PATTERN (PRACTICE)

This chapter introduces programming for a complex Sequence Pattern in the Normal Speed Sequence mode.

Contents Page

In this chapter you will learn how to program a more complex sequence pattern than that you learned in Chapter 3, to be run in the Normal Speed Sequence mode. Assume a Sequence Pattern as shown Figure 15.

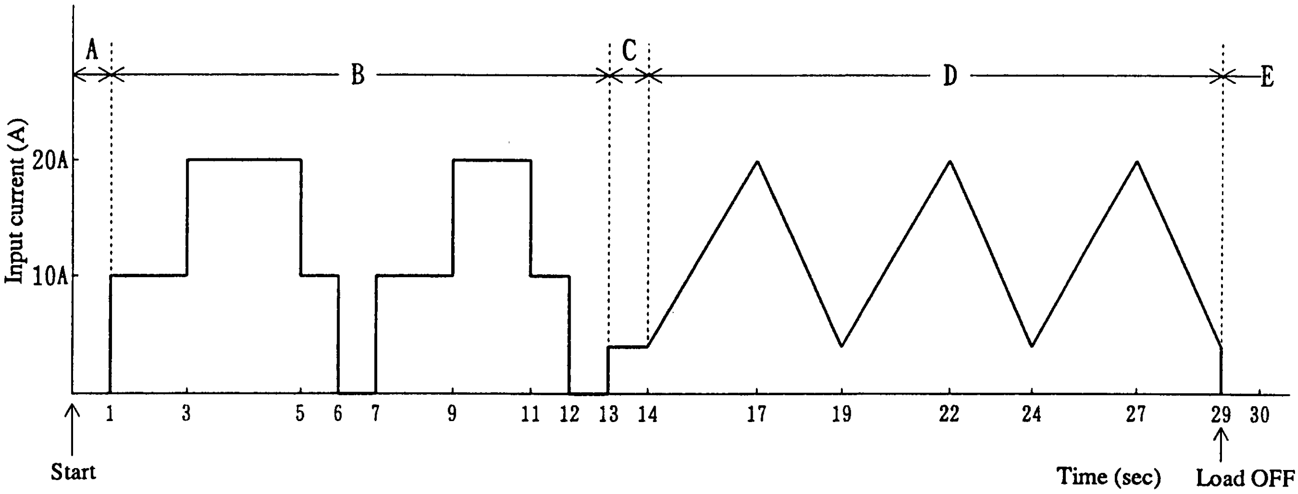


Figure 15. Analyzing a Complex Pattern

4.1 Analyzing the Pattern

The first thing you should do is to analyze the Sequence Pattern. There is no fixed way of pattern analysis. You may do it at you discretion. The basics of analysis, however, is to analyze it into Start, Loop (repetition), Chain, and End sections. In this example, let's divide the pattern into five sections (A through E) as shown in Figure 15 and assign ***Program*** numbers (01 through 05) to them. Based on these, let's analyze the pattern as shown in Table 4.

Table 4

A	Step 001: First, 0A for 1 sec	Program 01	Sequence 1
B	Step 001 10A for 2 sec (step change) Step 002 20A for 3 sec (step change) Step 003 10A for 1 sec (step change) Step 004 0A for 1 sec (step change) (Repeat Steps 001 through 004 twice.)	Program 02	Sequence 2
C	Step 001 4A for 1 sec (step change)	Program 03	Sequence 3
D	Step 001 Ramp change from 4A to 20A in 3 sec Step 002 Ramp change from 20A to 4A in 2 sec (Repeat Steps 001 and 002 three times.)	Program 04	Sequence 4
E	Step 001 0A and load OFF	Program 05	End Program

As can be seen in Table 4, there are nine steps in total. Normally, you should fill in a Sequence Coding Sheet with data as shown in Table 4.

4.2 Creating a New Sequence File

In this chapter you will create another new Sequence File. As you do this, the program you created in the preceding chapter will be deleted. If you need it, you can save it. The saving procedure also is described in this chapter.

Creating a New File

1) Choose "3: NEW" from the Edit Menu.

```
  2:  Edit  Sequence
 >3:  New
```

2) Here you create a file in a similar manner as you did in Chapter 3. The file is for a Sequence of "NI mode, sec unit."

```
 Create  New  Sequence
     Sure  ?
```

Press the [ENTER] key. Select mode "NI" and time unit "sec."

```
 Mode  :  NI
 Unit  ->  sec
```

For you information, an example of Edit Display of Steps of Normal Speed Sequence is shown in Figure 16 and that of Sequence in Figure 17.

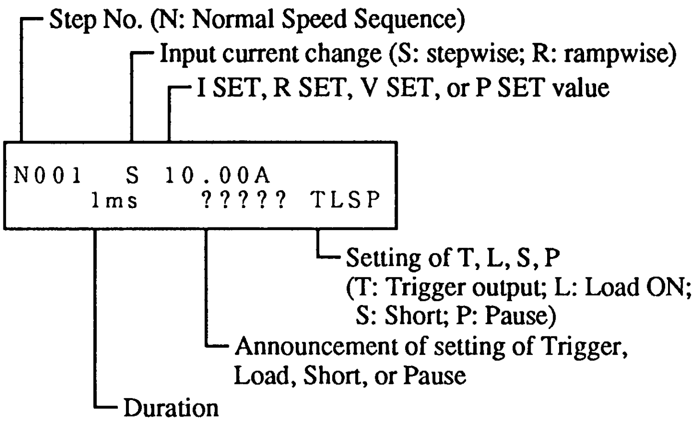


Figure 16. An Example of Step Edit Display

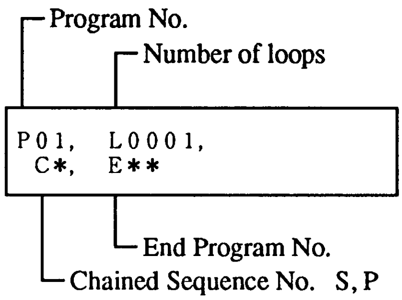


Figure 17. An Example of Sequence Edit Display

4.3 Creating a Program

4.3.1 Securing a Step Area for Program 01

After declaring a new file, create the first Program shown in Table 4. As shown in Table 4, Program 01 consists of a single Step. So, secure an area for one Step. This procedure is identical with that you practiced in 3.2.2, and is omitted to be described here.

1) Choose "Edit Program" from the Edit Menu.

```
Program:              :NI
     000  Step  [T000]
```

Press the [ENTER] key, and the following message will appear.

```
N001
  End  of  Step
```

2) Since no Steps at all are inserted yet, Secure a step area.

```
N001
    1:Modify
```

```

  >2:Insert
```

```
Insert:001
 How  many  steps?   1
```

Press the [ENTER] key to secure an area for one Step.

4.3.2 Editing a Step

In the above you inserted an area for a single Step in Program 01. Here, you edit the Step. For the example given in Chapter 3, you edited five Steps. It shouldn't be very difficult for you to edit a single Step. First, invoke the Step Edit Menu.

```
N001  S  0.00A
   >1:Modify
```

Select "1: Modify" with the 【◁▷】 keys. Press the 【ENTER】 key and enter data as follows:

Step 1

```
N001  S  0.00A
     1.0s
```

If you have entered a wrong data and have to correct it later, follow the re-edit procedures introduced in the next section. If you don't need re-edit, just skim the next section.

4.3.3 Re-editing a Program

Here we introduce the procedures to modify, insert or delete Steps of an existing Program.

Return to the Root State by pressing the 【ESC】 key the required number of times. Choose "1: Edit Program"from the Edit Menu.

```
Program:01
     000 Step  [T001]
```

Remember that you can select other Program Number with the 【◁▷】 keys and check the number of Steps the Program has.

Item "001" of the above display means that the Program has only one Step in this example. (In other cases the number may be different.)

Press the [ENTER] key, and the Step Edit Menu will appear.

```
N001  S  0.00A
   >1:Modify
```

This menu offers the following three items:

1 : Modify	To modify steps
2 : Insert	To insert Steps
3 : Delete	To delete Steps

1) Modifying Steps

Select "1: Modify" and press the [ENTER] key, and the Step Edit Menu will appear. Select a step number with the [SHIFT] + [◁▷] keys and modify the Step following the procedure of 3.2.3 "Editing Steps."

2) Inserting Steps

Select "2: Insert" and press the [ENTER] key, and the Step Edit Menu will appear. With the [SHIFT] + [◁▷] keys, Select the Step Number next to the one following which Steps are to be inserted. (For example, if you want to insert Steps following Step 4, select Step Number 4; if you want to add Steps following the final Step, select the Step Number of "End of Step.") Specify the number of Steps you want to insert with the JOG dial, and press the [ENTER] key.

3) Deleting Steps

Select "3: Delete" and press the [ENTER] key, and the Step Edit Menu will appear. Select the Step Number of the top one of the Steps to be deleted (for example, No. 4 if you want to delete Steps Nos. 4 and 5) with the [SHIFT] + [◁▷] keys and select the number of Steps you want to delete (2 in this example) with the JOG dial, and then press the [ENTER] key.

4.4 Creating a Sequence

Now, you will create the ***Sequence*** that specifies how Program 01 is to be executed.

The procedures used in this chapter so far are similar to those introduced in Chapter 3. However, the procedures introduced hereafter differ from them. These procedures are for the Sequence that specifies in what manner the Program is to be executed. If you want to make sophisticated Sequence Files, it is essential that you make yourself fully familiar with these procedures.

To create a Sequence, return once to the Root State. Then proceed as follows:

1) Choose "Edit:" to invoke the Edit Menu.

```
>1: Edit Program
 2: Edit Sequence
```

2) Choose "2: Edit Sequence" by scrolling the menu with the 【◁▷】 keys. A Sequence monitor display as follows will appear.

```
Sequence:1          :NI
 P01, L0000, C*,   E**
```

3) You can select a Sequence to be edited, with the JOG dial. Turn the JOG dial for trial.

4) Select "Sequence: 1" and press the 【ENTER】 key. The Edit Display with a cursor will appear.

4.4.1 Setting Sequence Items

The Display shows four data items of the Sequence. The item indicated by the cursor (□) is the one that is selected. You can move the cursor among the four items with the 【◁▷】 keys. The meanings of the individual items are as described below.

```
P 0[1],    L 0 0 0 0,
  C *,     E * *
```

1) "P" specifies a program number of the program by which this Sequence should start. In this example, it is "P01" specifying that "This Sequence starts by Program 01."

2) "L" specifies the number of repetitions of the Program (the number of loops). Number "0000" is for the initial value only. The Sequence does not run with loop zero — an error will result. You must set it at 1 or more, with the JOG dial. Number "9999" (the actual indication is with symbol ∞) denotes infinitive and endless repetitions and the program is repeated until you press the 【STOP】 key.

3) "C" specifies the Sequence number of the next Sequence to which the current Sequence is to be chained when it is over (when the program has been repeated for the number of times indicated by the "P" item.) You can select the next sequence number with the JOG dial. The asterisk means that there is no chain — in which case the current Sequence is chained to the End Program explained below. If the Sequence number of own Sequence is specified for the C item, the Sequence will run endless.

4) "E" specifies the Program Number of the Program to be executed when the 【STOP】 key is pressed to halt the execution of the Sequence. This Program is called "End Program." It specifies the output state to be assumed (for example, output to be turned off) when the Sequence is halted. Because it is for the end state, it needs only one Step. Even when it has two or more Steps, the initial one alone of them is executed.

   Note that what is specified here is not a Sequence Number but is a ***Program*** Number. Also note that, if "*" is specified in the C (chain) item, the Sequence jumps to the End Program.

   If "*" is specified for the C item and "**" (no End Program) for the E item, the end state will be as follows: When the Sequence is interrupted, it will maintain the input current that existed at the instant of interruption. When the loops are normally completed, the Sequence will end by holding the input current that has been set by the final Step of the Program specified by P.

   It is recommended to specify an End Program for each Sequence in order to make certain the state the instrument should assume when the Sequence is halted.

   The relationship between Chain and End Program is shown in Figure 18.

Set Sequence 1 as shown below, for example. To move among menu items, use the 【◁▷】 keys1; to select a numeric value, use the JOG dial.

```
P01,   L0001
 C2,   E**
```

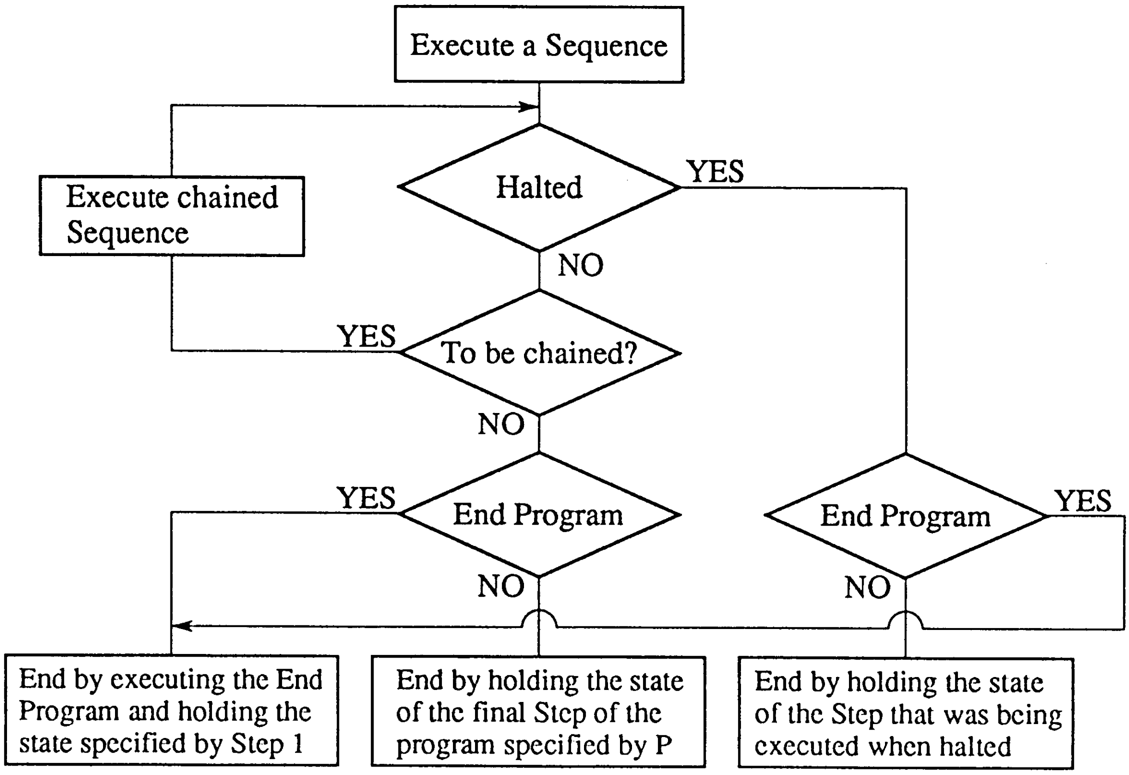


Figure 18. Relationship Between Chain and End Program

4.4.2 Re-editing a Sequence

If you want to re-edit a Sequence, press the 【ESC】 key several times to return to the Root State and then choose "Edit Sequence" from the Edit Menu. For re-edit, follow a procedure identical with that for creating Programs.

4.5 Creating Remaining Programs and Sequences

4.5.1 Creating Program 2

With similar procedures as you learned in the above, create Program 02 shown in Table 2 and register it as the *Start Program* of Sequence 2. According to Table 4, Program 2 will have four Steps.

```
1) Program:02        :NI
      000 Step   [T001]
```

1) First, secure an area for Step and edit the Program.

Step 001: 10A for 2 sec (stepwise change), load ON

```
N001   S  10.00A
  2.0s              ·L··
```

Step 002: 20A for 3 sec (stepwise change), load ON

```
N002   S  20.00A
  3.0s              ·L··
```

Step 003: 10A for 1 sec (stepwise change), load ON

```
N003   S  10.00A
  1.0s              ·L··
```

Step 004: 0A for 1 sec (stepwise change), load ON

```
N004   S   0.00A
  1.0s              ·L··
```

Note that Step numbers are locally allocated within each Program. That is, Program 01 has its Step 1 and Program 02 also has its Step 1.

2) Register this program as Program 2, Sequence 2.

Sequence 2: (Start Program 2, Loop 2, Chain 3, End 5)

```
P02,  L002,
 C3,  E05
```

4.5.2 Editing Programs 03 Through 05

Now, edit Programs 03, 04, and 05.

1) Program 03

Program 03 has one Step. It is registered as the Start Program of Sequence 3.

Step 001: 4A for 1 sec (stepwise change), load ON

```
N001   S   4.00A
   1.0s              ·L··
```

Sequence 3: (Start Program 3, Loop 2, Chain 4, End 5)

```
P03,  L002
 C4,  E05
```

2) Program 04

Program 04 has two Steps. It is registered as the Start Program of Sequence 4.

Step 001: To 20A in 3 sec (rampwise change), load ON

```
N001    S  20.00A
  3.0s              ·L··
```

Step 002: To 4A in 2 sec (rampwise change), load ON

```
N002    S   4.00A
  2.0s              ·L··
```

Sequence 4: (Start Program 4, Loop 3, Chain none, End 5)

```
P04,  L0003
 C*,  E05
```

Program 04 is for rampwise change of input current. So, you have to choose "R" (rampwise change) instead of "S" on the Step Edit Display. You may do this with the JOG dial. If you specify "R," the input current will change rampwise from the value that existed immediately before entering this Step to the value specified for this Step in the period specified for this Step. (See Figure 19.)

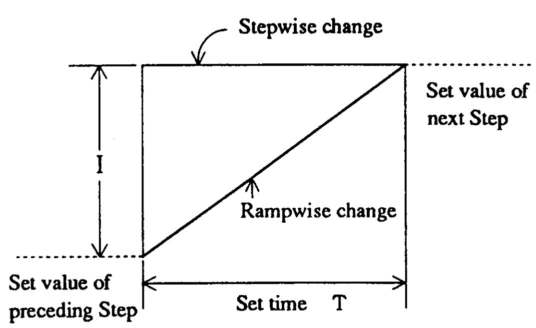


Figure 19. Stepwise Change and Rampwise Change

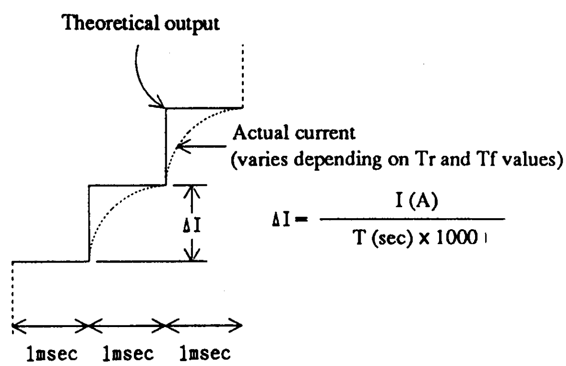


Figure 20. Enlarged View of Rampwise Change

Rampwise changes actually are through approximation by fine stepwise changes as shown in Figure 20.

NOTE

With a Step which specifies a rampwise change, the current should change rampwise from the value that existed immediately before entering this Step to the value specified for this Step in the period specified for this Step. Therefore, if the initial Step of a Sequence specifies a rampwise change, be sure to check that the setting of the current before entering this Step is correct.

3) Editing Program 05

In this example, Program 05 is used as an End Program. So, only a single Step is enough for this Program. Note that, unlike the other Programs, Program 05 is not required to be registered to Sequence 5.

The End Program is invoked when the [STOP] key is pressed while the Sequence is in progress or when the Sequence is over. There cannot be such Sequence that begins with an End Program.

To create an End Program, proceed as follows:

Step 001: 0A (stepwise change), load OFF

```
N001   S   0.00A
  0.1s             ....
```

The setting of execution time (duration) in this case is meaningless and you may leave it at 0.1s. By the above procedure, edition of Programs 03 through 05 is complete.

Now you have created a Sequence File for the actions specified in Table 4. However, the contents of Sequence 2 and thereafter (Program 02 and thereafter) are still on the volatile Edit Memory.

4.6 Running a Sequence

Let's run the Sequence you have just created. First, return the instrument to the Root State by pressing the [ESC] key for the required number of times (or by turning ON the instrument power after turning it OFF once). To the Electronic Load, connect a power supply of a level of 5V and 25A or thereabout.

To Run a Sequence

1) Press the [RUN] key.

```
Sequence:1
 P01, L0001,  C2,  E05
```

With the JOG dial or [◁▷] keys, select a Sequence Number to be executed at first. In this example here, select Sequence 1. Press the [RUN] key.

```
 0.00A -0.00V    0. 0W
-- S2, P02, L0002, 0004
```

The currently running Sequence number, Program number, the number of Loops, and the Step Number will be displayed on real time. When steps are short, they may change too fast for you to read them. But you may know that they are running.

If you have correctly programmed the Sequence, the input current pattern should be as shown in Figure 15. If the pattern is incorrect, it means that there are programming errors. Repeat programming until a correct current pattern is realized. (Practice it until you become thoroughly familiar with programming.)

For trial, return to the above procedure (2), select Sequence Number 4, and press the [RUN] key. Thus, it also is possible for you to run the Sequence starting by its midway.

In the above you have learned how to create a Normal Speed Sequence.If this was the first time for you to program a Sequence, you might have consumed quite a time to create this Sequence. When you have fully understood the structure of the Sequence File (Figure 8) and become fully familiar with the procedures of creating Files, you will be able to create a Sequence of this level within five minutes or so.

Chapter 5

FAST SPEED SEQUENCE FOR A RAPIDLY CHANGING PATTERN (PRACTICE)

This chapter introduces programming for a rapidly changing Sequence Pattern in the Fast Speed Sequence mode.

Contents Page

In this chapter you will learn how to program a Fast Speed Sequence for a rapidly changing pattern. The programming procedures basically are identical with those for the Normal Speed Sequence. Rather, programming for the former is simpler than that for the latter, because the former involves less items to be programmed.

Assume that we need a Fast Speed Sequence Pattern as shown in Figure 21.

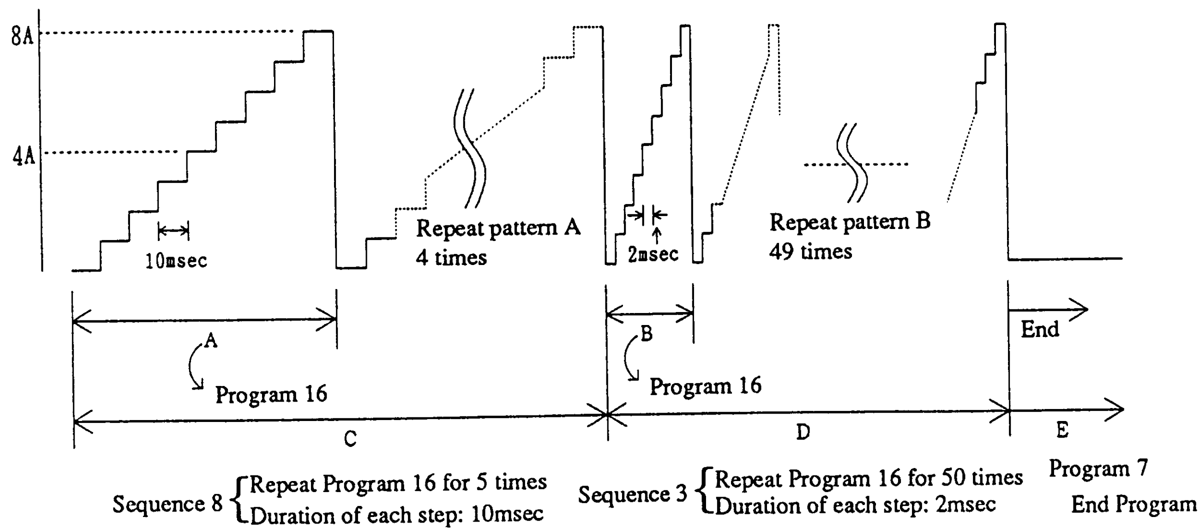


Figure 21. An Example of Fast Speed Sequence

5.1 Analyzing the Pattern

First, let's analyze the patterns of sections A and B in Figure 21. Although the duration per Step differs, the two patterns are identical in that the current changes from 0A to 8A, by 1A per Step. (As described in Chapter 1, with programming for a Fast Speed Sequence, the peak current values alone are specified. The durations of Steps are uniform and are specified for each Sequence.) Thus, overall pattern of Figure 21 consists of repetitions of identical element patterns. Let's call these element patterns as "Program 16" as shown in Figure 21.

Section C consists of five repetition of Program 16, with duration 10 milliseconds per Step. Let's call Section C as "Sequence 8" as shown in Figure 21. Section D consists of 50 repetition of Program 16, with duration 2 milliseconds per Step. Let's call Section D as "Sequence 3" as shown in Figure 21.

Section E is of the End Program. It specified the state the instrument should assume at the end of the Sequence. When the final Step of Sequence 3 is finished, the current will remain at its set value (8A). The End Program (Program 7) reduces it to 0A.

The Programs and Sequences are summarized in the following tables:

Programs

Program 16	Increase from 0A to 8A, with 9 Steps (1A per Step)
Program 07	End Program for 0A, with 1 Step

Sequences

Sequence 8	Repeat Program 16 for 5 times, with duration 10 milliseconds per Step, and then chain to Sequence 3.
Sequence 3	Repeat Program 16 for 50 times, with duration 2 milliseconds per Step, and then execute the End Program.

It may sound to you that we have assigned the Program Numbers and Sequence numbers quite at random. It is not important to assign the numbers in a neat order. What we want you to understand is that you can attain various current patterns through various combinations of 8 Sequences and 16 Programs. This applies to the Normal Speed Sequences also.

5.2 Creating a New Sequence File

You have to create a new Sequence File for Program 16 and Program 07(End Program). You have already leaned this procedure in Chapter 3 and subsequent chapters, and this procedure will not be described here again in detail.

What you are going to learn here is a Fast Speed Sequence. So. create a File of FI mode. (You can select the mode with the JOG dial.)

```
Mode:FI
```

The duration per Step must be specified for each Sequence when editing the Sequence (refer to Section 5.4).

Examples of Edit Displays of Fast Speed Sequences are shown in Figures 22 and 23.

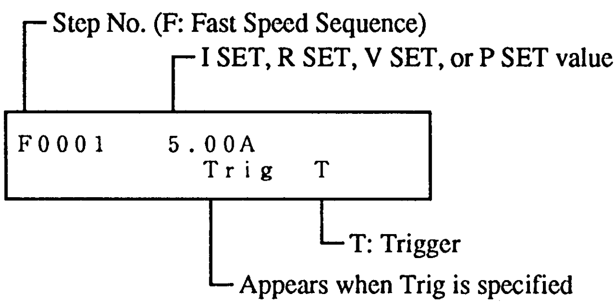


Figure 22. An Example of Step Edit Display

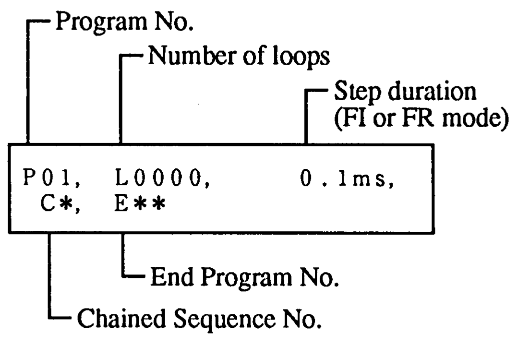


Figure 23. An Example of Sequence Edit Display

5.3 Creating a Program

5.3.1 Creating Program 16

Let's create Program 16. Program 16 is comprised of nine Steps. Secure a memory area for these Steps.

1) Invoke the Sequence Menu with the [SEQ] key. From "2: Edit," choose ": Edit Program." To create Program 16, select "Program: 16" with the [◁▷] keys.

```
Program:16          :F1
     000 Step  [T0000]
```

2) Press the [ENTER] key twice to invoke the Program Edit Menu. Choose "2: Insert."

```
F0001
  End of Step
```

```

 >2:Insert
```

3) As nine Steps are to be inserted, choose 9 with the JOG dial.

```
Insert:0001
 How many Steps?     9
```

Press the [ENTER] key to secure an area.

5.3.2 Editing Steps

Now, let's edit the nine Steps for which you secured an area in the above. The Step edit procedures basically are identical with those for the Normal Speed Sequence. Rather, the former are simpler as they eliminated some items (Pause, Load, and Short).

You cannot use the Pause function when in the Fast Speed Sequence mode. The matter of whether the input current flows or not depends on the Load ON/OFF setting of the instrument that has been done before starting the Sequence. Due to this, you have to set the instrument to the Load ON state before start running the Sequence. Furthermore, different from that when in the Normal Speed Sequence mode, you cannot assign different durations to individual Steps. You can specify only a single duration value that is uniformly applied to all Step durations throughout each Sequence.

Now, let's enter values for the nine Steps as shown below. First, select "1:Modify" and then proceed as follows:

Step 0001

```
F0001    0.00A
                   .
```

Step 0002

```
F0002    1.00A
                   .
```

Step 0003

```
F0003    2.00A
                   .
```

Step 0004

```
F0004    3.00A
                   .
```

Step 0005

```
F0005    4.00A
                   .
```

Step 0006

```
F0006    5.00A
                   .
```

Step 0007

```
F0007    6.00A
                   .
```

Step 0008

```
F0008     7.00A
                  .
```

Step 0009

```
F0009     8.00A
                  .
```

Program 16 you have created here will be registered in double — as Start Program of Sequence 8 and that of Sequence 3. We will do this registration work (setting of Sequences) later.

5.3.3 Making an End Program

Let's make an End Program (Program 7) here. For this Program, secure a single Step and set it as follows:

Step 0001

```
F0001     0.00A
                  .
```

The current to be set in this Step is 0A. Immediately after you have secured the area, the default value is 0A. Normally, you will not need to change the value.

5.4 Creating Sequences

Here, let's register Program 16 to two different Sequences, with two different Step durations for respective Sequences. (For the Normal Speed Sequence, you specify the durations of individual Steps. For the Fast Speed Sequence, you specify a duration that is uniformly applied to all Steps of a Sequence.)

The edit procedures of Fast Speed Sequences are identical with those of Normal Speed Sequences. Let's edit Sequence 8 and Sequence 3 as follows:

Sequence 8

```
P16,  L0005,     10.0ms
 C3,  E07
```

Sequence 3

```
P16,  L0050,      2.0ms
 C*,  E07
```

Now you have completed a Fast Speed Sequence File.

5.5 Running a Sequence

Now let's run the Sequence you have just created. First, return to the Root State by pressing the 【ESC】 key for the required number of times. To the Electronic Load, connect a Power Supply of a level of 5V and 10A or thereabout.

To Run the Sequence

1) Press the 【RUN】 key.

The Sequence to be executed at first is Sequence 8. Be sure to select Sequence 8 with the 【◁ ▷】 keys.

```
Sequence:8            :FI
 P16,  L0000,    10.0ms
```

Be sure to set the instrument to Load ON. The Fast Speed Sequence mode of operation has no item to specify Load ON/OFF. No input current flows unless you have set the instrument to Load ON.

Press the 【LOAD】 key to turn ON the input current. Press the 【RUN】 key.

```
 3.00A   5.00V   15.0W
--  S8, P16,L0003, 0004
```

As the Sequence runs, the displayed items will change rapidly on real time. When the Sequence is over, the display will become stationary and, since 0A is specified by the End Program, the input current will be reduced to zero.

(NOTE) · Note for Running a Fast Speed Sequence

This chapter describes an example of the chained sequence for the Fast Speed Sequence. The Fast Speed Sequence has priority in the high-speed operation steps. Thus, care must be taken when using the Fast Speed Sequence since the final step execution time of a program differs from the set value if sequences are chained.
In the example of Figure 21, the pattern A final step execution time differs from the set value when sequences C and D are chained whereas the pattern B final step execution time differs from the set value when sequences D and E are chained.

INDEX

Alphabetic order

ELECTRONIC LOAD PLZ-3W/3WH Sequence Operation Guidebook