

μ PD17104

APPLICATION FOR ELECTRIC FAN

NEC

APPLICATION NOTE

μ PD17104

APPLICATION FOR ELECTRIC FAN

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Corporation. NEC Corporation assumes no responsibility for any errors which may appear in this document.

NEC Corporation does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from use of a device described herein or any other liability arising from use of such device. No license, either express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Corporation or of others.

CONTENTS

CHAPTER 1	OUTLINE	1
CHAPTER 2	FUNCTIONS OF MICROCOMPUTER-CONTROLLED ELECTRIC FAN	2
2.1	FUNCTIONAL DESCRIPTION	2
2.2	INPUT KEYS AND OPERATION	2
2.3	DISPLAY	3
CHAPTER 3	HARDWARE AND SOFTWARE SPECIFICATIONS	4
3.1	BLOCK DIAGRAM	4
3.2	CIRCUIT DIAGRAM	5
3.3	PORT ASSIGNMENT	6
3.4	STATUS TRANSITION DIAGRAM AND STATUS TRANSITION TABLE	8
3.5	DESCRIPTION OF EACH MODE	9
3.6	KEYS	12
3.7	ZERO-CROSS PULSE DETECTION	13
3.8	DISPLAY	14
3.9	FAN MOTOR CONTROL	16
CHAPTER 4	CONTROL PROGRAM	18
4.1	ALGORITHM	18
4.2	RAM MAP (Description of Symbols and Flags)	19
4.3	MEMORY MAP	24
4.4	GENERAL FLOW CHART	25
4.5	DETAILED FLOW CHART	26
4.5.1	30 ms Delay	26
4.5.2	50 Hz/60 Hz Decision	27
4.5.3	Initial Setting	28
4.5.4	LED Scan	30
4.5.5	Key Scan	34
4.5.6	Buzzer Output and 2 ms Adjustment Process	39
4.5.7	4 sec Count	41
4.5.8	Timer, Sleeping Timer Remaining Time Check	42
4.5.9	Sleeping Timer ON/OFF Control	43
4.5.10	Random Pattern Count	45
4.5.11	Input Key Process	46
4.5.12	Motor Control	53
4.6	SOURCE PROGRAM LIST	57

CHAPTER 1 OUTLINE

Most electric fans available in recent years employ microcomputers to provide various functions as value added, such as random blow simulating natural breeze, and employment of the Sleeping Timer that gradually reduce the air flow with time or gradually lengthens the intervals at which intermittent fan operation is carried out.

This Application Note introduces programs for the μ PD17104 used to control the fan motor, head swing motor, and timers, for an electric fan.

The μ PD17104 is a low-price "tiny microcontroller", which consists of 1K-byte ROM, 16-word RAM, and 16 I/O ports. The μ PD17104 is suitable for electrical appliances, in which relatively simple control action is required, such as electric fans.

CHAPTER 2 FUNCTIONS OF MICROCOMPUTER-CONTROLLED ELECTRIC FAN

2.1 FUNCTIONAL DESCRIPTION

This electric fan has the following functions:

- Wind speed selectable from Low, Medium, and High
- Random air flow changing wind velocity (Low random air flow, Medium random air flow, High random air flow)
- Electronics timer for 30 minute, 1 hour, or 2 hour operation
- Sleeping Timer for 1 hour, 2 hour, or 4 hour operation (Sleeping Timer: repeats ON → OFF → ON, and gradually lengthens intervals at which intermittent fan operation is carried out.)
- Motor specifically provided for head swing operation

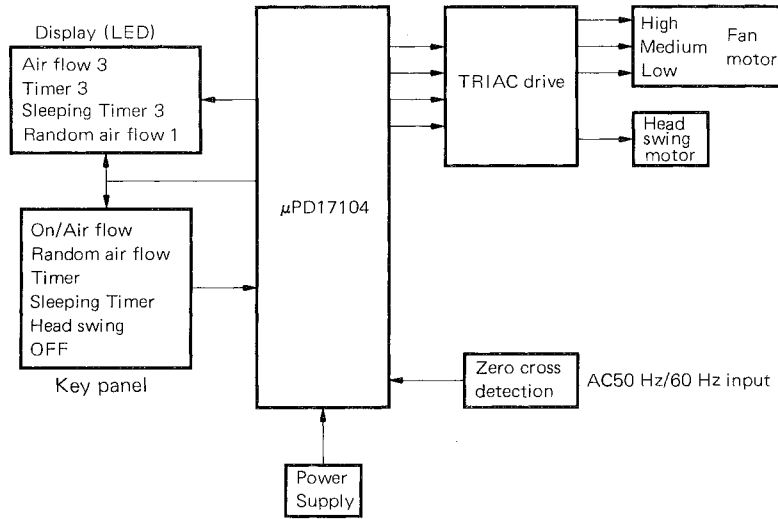
2.2 INPUT KEYS AND OPERATION

- On/Air flow key Air velocity changes from Low, to Medium, to High, then to Low by repeatedly pressing the key. Initially low is selected. The LED indication corresponds to the air velocity.
- Timer key Timer length changes from 30 minutes to 1 hour, to 2 hours, to Timer-OFF, then to 30 minutes by repeatedly pressing the key. The LED indication corresponds to the timer length.
- Sleeping Timer key Sleeping Timer length changes from 1 hour, to 2 hours, to 4 hours, to Sleeping Timer-OFF, then to 1 hour by repeatedly pressing the key. The LED indication corresponds to the Sleeping Timer length.
ON → OFF → ON → is repeated in the pattern shown in the following page.

CHAPTER 3 HARDWARE AND SOFTWARE SPECIFICATIONS

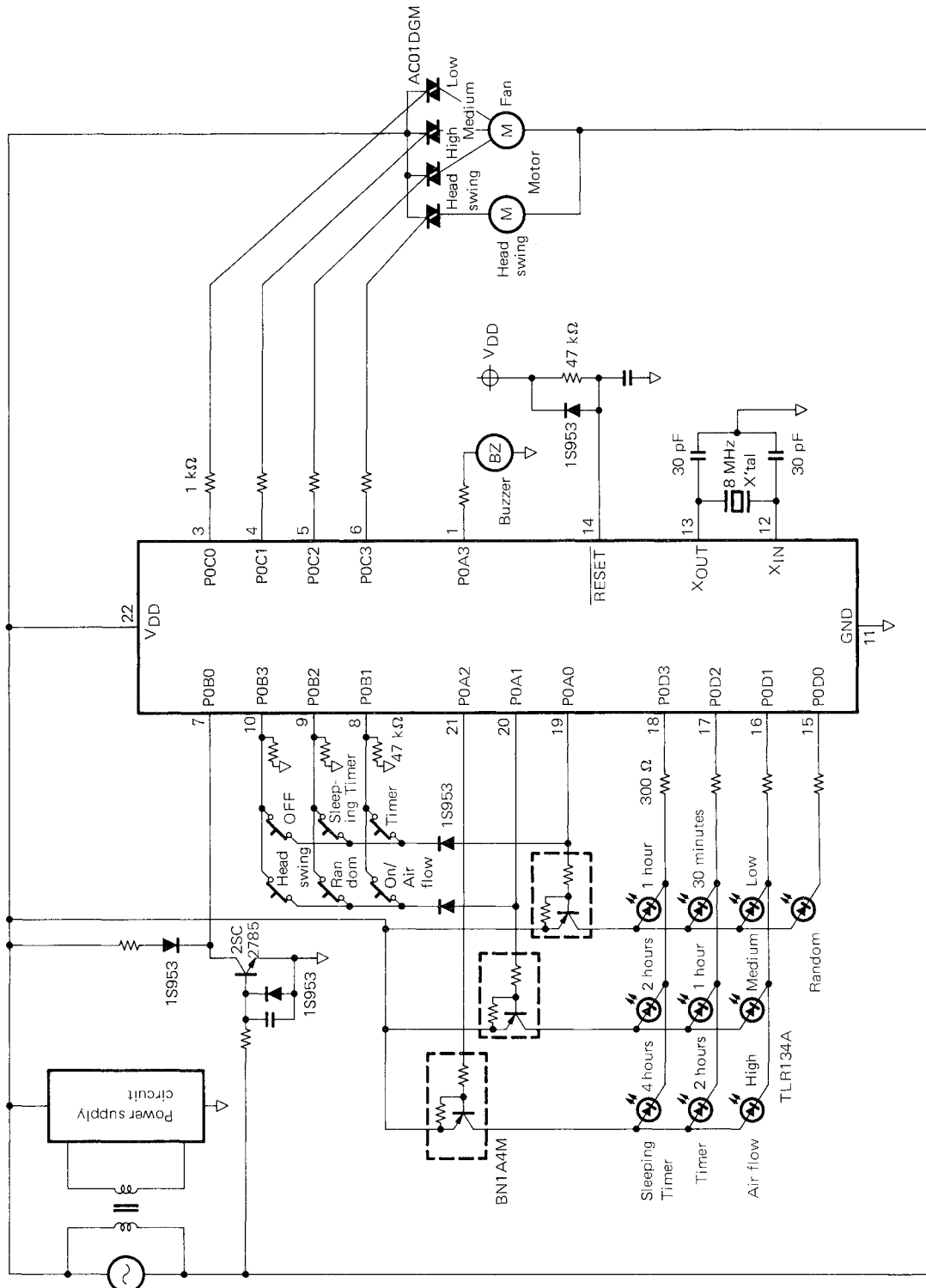
3.1 BLOCK DIAGRAM

The figure below shows a block diagram of this system.



3.2 CIRCUIT DIAGRAM

The figure below shows a circuit diagram for the electric fan used in this system (constants are for reference purposes only).



3.3 PORT ASSIGNMENT

Pin No.	Port name	Input/Output	Signal name	Active level	Condition at Power on Reset (See Note 1)	Condition during standby mode		Function etc.
						STOP mode	HALT mode	
1	POA3	O	Buzzer output	H/L	L	L	—	
2	NC	—	Unused	—	—	—	—	
3	POC0	O	Low air flow TRIAC output	L	H	H	—	
4	POC1	O	Medium air flow TRIAC output	L	H	H	—	
5	POC2	O	High air flow TRIAC output	L	H	H	—	
6	POC3	O	Head swing TRIAC output	L	H	H	—	
7	POB0	I	Zero cross detection	H/L	—	—	—	
8	POB1	I	Key input	H	—	—	—	Standby mode release
9	POB2	I	Key input	H	—	—	—	
10	POB3	I	Key input	H	—	—	—	
11	GND	—	—	—	—	—	—	GND potential pin
12	X _{IN}	—	System clock oscillator	—	—	—	—	8 MHz ceramic resonator
13	X _{OUT}	—	System clock oscillator	—	—	—	—	8 MHz ceramic resonator

Note 1: Before initial setting

(continued)

Pin No.	Port name	Input/Output	Signal name	Active level	Condition at Power on Reset (See Note 1)	Condition during standby mode		Function etc.
						STOP mode	HALT mode	
14	$\overline{\text{RESET}}$	I	Reset	L	—	—	—	CPU reset pin
15	P0D0	O	LED segment signal output	L	H	H	—	
16	P0D1	O	LED segment signal output	L	H	H	—	
17	P0D2	O	LED segment signal output	L	H	H	—	
18	P0D3	O	LED segment signal output	L	H	H	—	
19	P0A0	O	LED digit signal output/key scan 1 signal output	Note 2	L	L	—	
20	P0A1	O	LED digit signal output/key scan 2 signal output	Note 2	L	H	—	
21	P0A2	O	LED digit signal output	L	H	H	—	
22	V _{DD}	—	—	—	—	—	—	Positive power supply pin

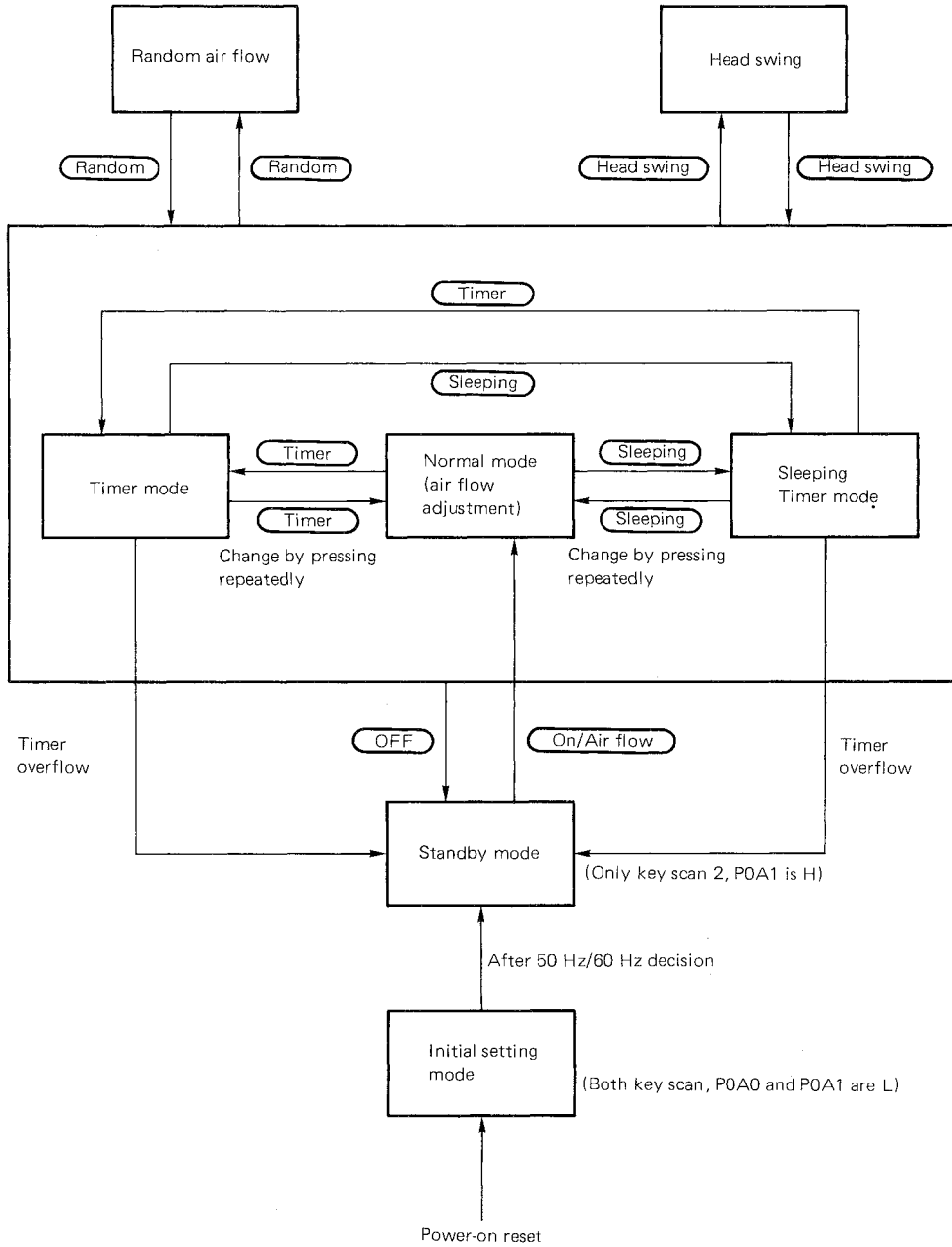
Note 1: Before initial setting

2: Pins 19 and 20 Active low for LED digit
Active high for key scan

3.4 STATUS TRANSITION DIAGRAM AND STATUS TRANSITION TABLE

The figure below shows status transition of this system for key input.

(1) Status transition diagram



○ : Key input

(2) Status transition table

Key input Status	On/Air flow	Timer	Sleeping timer	Random	Head swing	Off	Time overflow	After 60 Hz/ 50 Hz decision
Normal mode	○	Timer mode	Sleeping Timer mode	○	○	Standby mode	—	—
Timer mode	○	Timer length change. Moving to normal mode is possible	Sleeping Timer mode	○	○	Standby mode	Standby mode	—
Sleeping Timer mode	○	Timer mode	Timer length change. Moving to normal mode is possible	○	○	Standby mode	Standby mode	—
Standby mode	Normal mode	—	—	—	—	—	—	—
Initial setting		—	—	—	—	—	—	Standby mode

○: Mode is not changed, but processing is carried out in the respective mode.

—: Input is invalid.

3.5 DESCRIPTION OF EACH MODE

(1) Initial setting mode

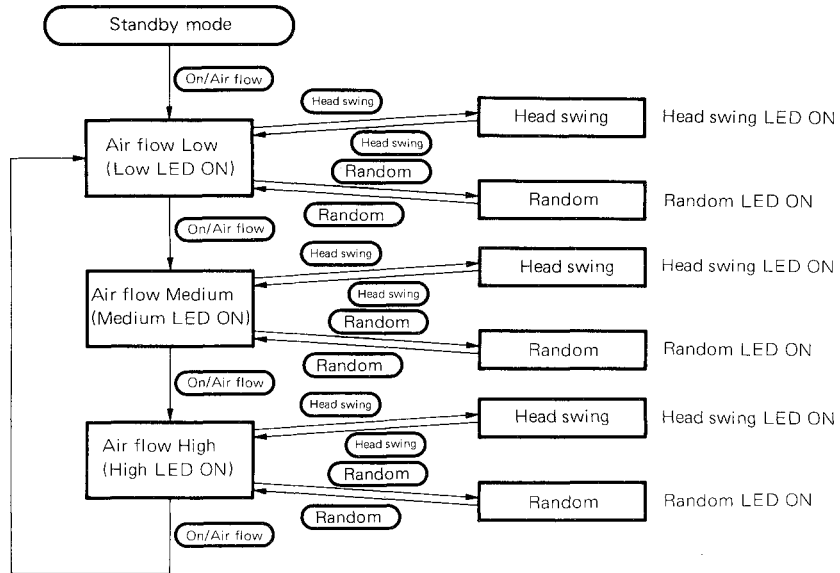
After power-on reset, checks the frequency of the AC power supply pulse (reference pulse of various control timings) input to the POBO pin to determine whether the frequency is 50 Hz or 60 Hz, then enters the standby mode.

(2) Standby mode

All operations are stopped and the system enters the standby mode (STOP mode). The standby mode is released only when the On/Air flow key is pressed.

(3) Normal mode

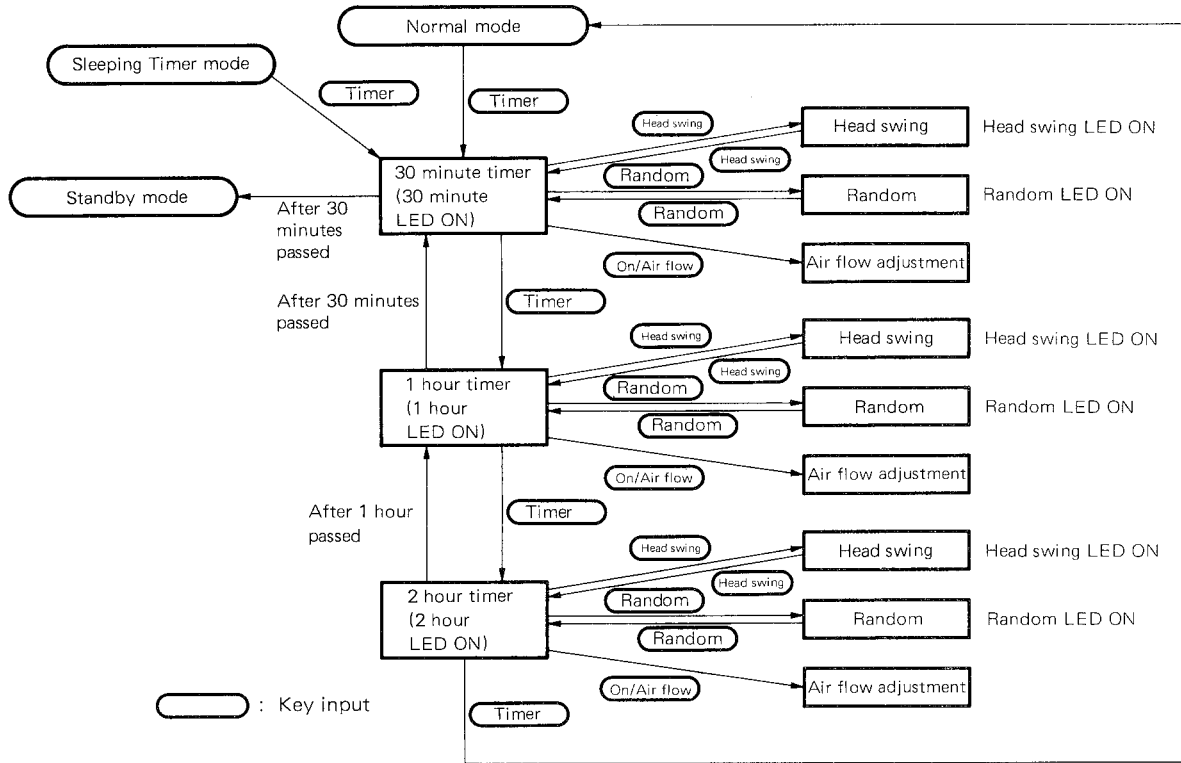
When the On/Air flow key is pressed in the standby mode, the system enters the Normal mode, and the fan starts to operate at Low speed without head swing. Then, status change, as shown in the figure below, becomes possible. Air flow, head swing, and random air flow can be independently set.



* Air flow indication LED is always ON
 ○ : Key input

(4) Timer mode

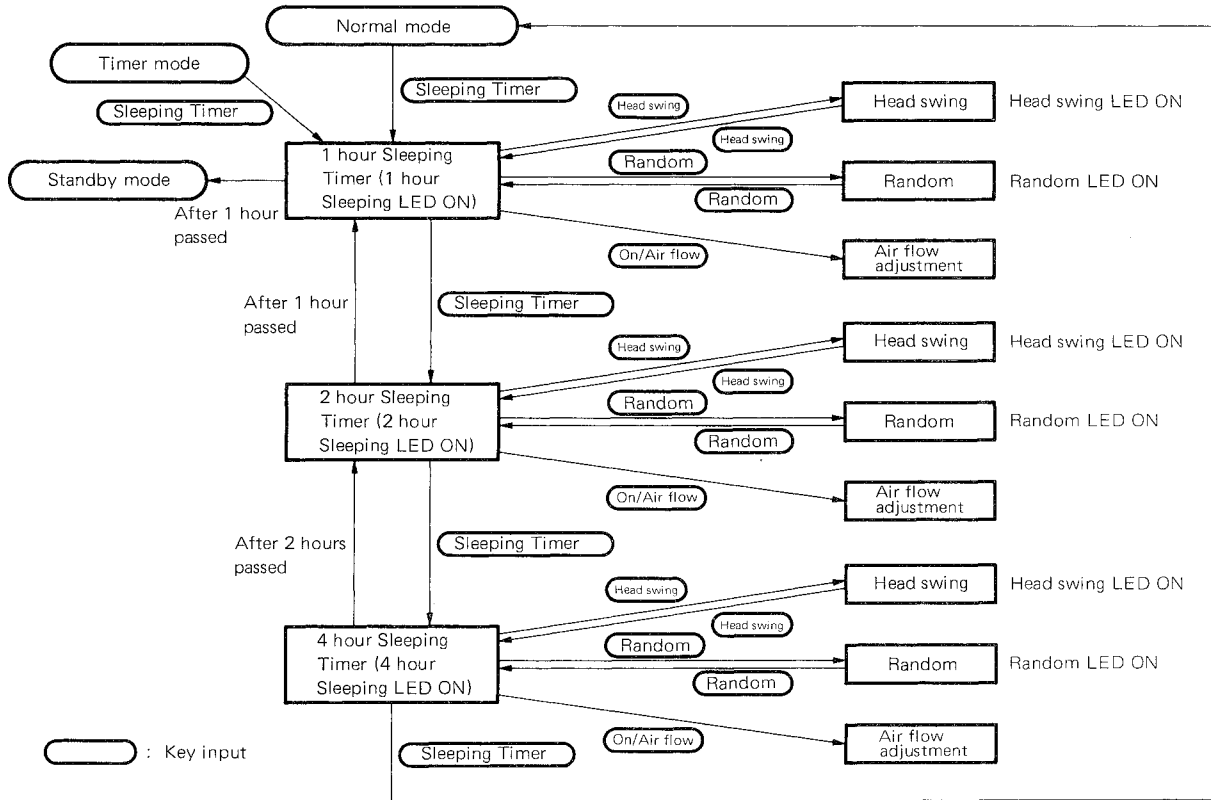
When the timer key is pressed in the Normal mode or the Sleeping Timer mode, the system enters this mode, and the 30 minutes timer is set. Then, status change, as shown in the figure below, becomes possible. LED indication changes as time passes, and the system enters the standby mode after the specified time passed (timer overflow).



(5) Sleeping Timer mode

When the Sleeping Timer key is pressed in the Normal mode or the Timer mode, the system enters this mode, and the 1 hour sleeping timer is set. Then, status change, as shown in the figure below, becomes possible. The Sleeping Timer length, air flow, head swing, and random air flow can be independently set.

LED indication changes as time passes, and the system enters the standby mode after the specified time passed.



3.6 KEYS

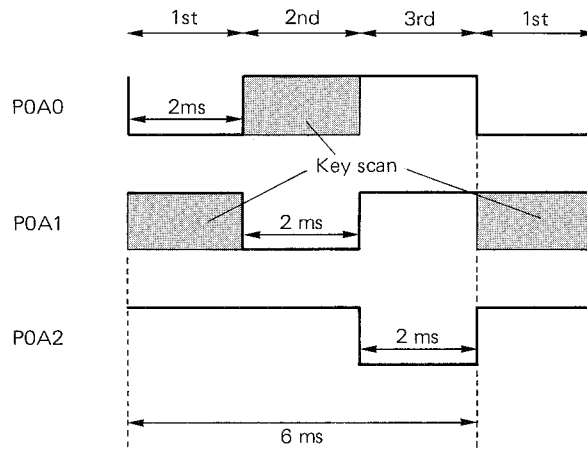
(1) Key and port assignment diagram

POA0	POA1	
Timer	On/ Air flow	P0B1/RLS _{STOP}
Sleeping Timer	Random	P0B2
OFF	Head swing	P0B3

When a key is pressed, 4 kHz buzzer sounds for 50 ms.

(2) Key scan

The LED display digit signal output pins also serve as key scan signal output pins. Their interval is 6 ms.



The display digit signals are active low, and the key scan signals are active high.

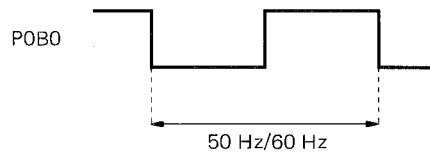
(3) Key valid/invalid table

○: Valid
 -: Invalid

Key name / Mode	ON/ Air flow	Timer	Sleeping Timer	Random	Head swing	OFF
Normal mode	○	○	○	○	○	○
Timer mode	○	○	○	○	○	○
Sleeping Timer mode	○	○	○	○	○	○
Standby mode	○	-	-	-	-	-

3.7 ZERO-CROSS PULSE DETECTION

50 Hz/60 Hz AC signal used as the reference pulse for time control for Timer, Sleeping Timer, and random air flow, is detected by software. The duty ratio for the input pulse must be at least 40 %.



3.8 DISPLAY

Display is 1/3 duty dynamic drive LED. Timing is as shown in (3) in 3.6. The table shown below describes the port assignment.

(1) Port assignment table

Segment Digit	P0D0	P0D1	P0D2	P0D3
POA0	Random	Air flow (High)	Timer 2 hours	Sleeping Timer 4 hours
POA1		Air flow (Medium)	Timer 1 hour	Sleeping Timer 2 hours
POA2		Air flow (Low)	Timer 30 minutes	Sleeping Timer 1 hour

Both segments and digits are active low.

(2) Key processing

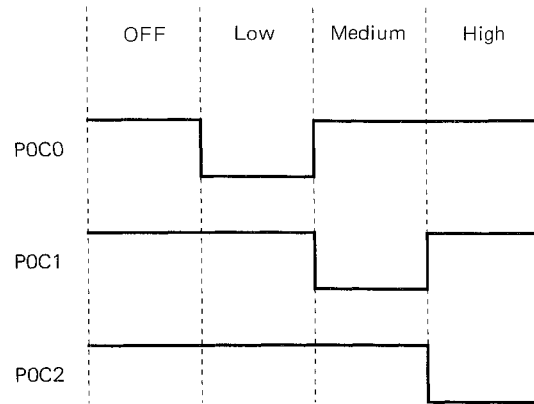
Key name	Condition	Processing		
		When pressed once	When held down	Multiple pressing
On/Air flow	During standby	Enters the Normal mode, and operates with Low air flow.	No change unless Key is released.	Invalid
	During Normal mode	Air flow can be changed from Low to Medium, to High, then to Low by pressing repeatedly.		
Random	Other than standby mode	When pressed once, random operation, in which the air flow is randomly changed, is carried out. When pressed again, the random operation is stopped and the normal operation will be resumed.	Same as above	Invalid
Head swing	Same as above	Swings the head to left and right. When pressed again, head swing operation will be stopped.	Same as above	Invalid
Timer	Same as above	When pressed once, the timer is set to 30 minutes. When repeatedly pressed, setting will be changed to 1 hour, to 2 hours, to timer off, then to 30 minutes.	Same as above	Invalid
Sleeping Timer	Same as above	ON → OFF → ON will be repeated. The OFF period will be gradually increased. Initially, the total time is set to 1 hour. When repeatedly pressed, setting can be changed to 2 hours, to 4 hours, then to OFF.	Same as above	Invalid
OFF	Same as above	Stops all operations, and enters the standby mode	Same as above	Invalid

Chattering is protected up to 30 ms for both ON and OFF.

3.9 FAN MOTOR CONTROL

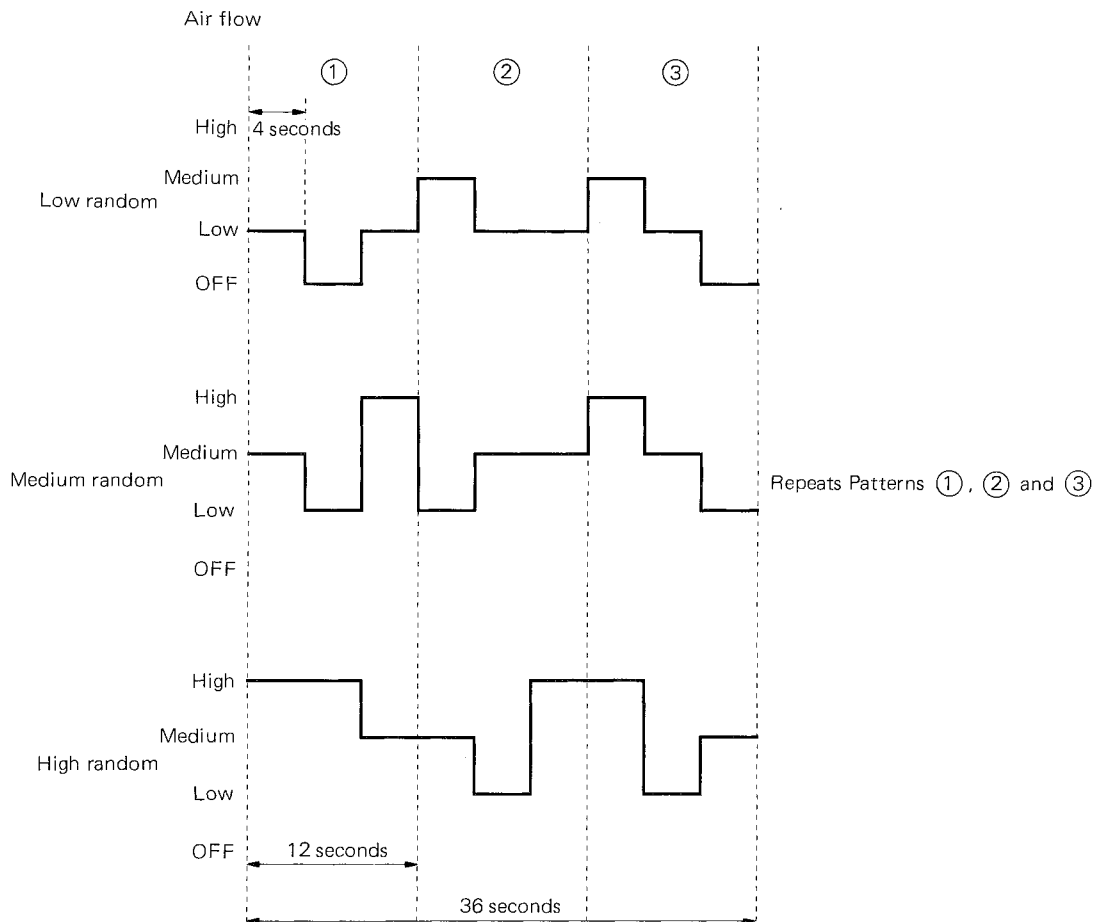
(1) Port output condition for each air flow

The following figure shows the output condition of each port in OFF, Low, Medium, and High air flow. TRIAC, which drives the motor, is turned ON at low level.



(2) Air flow variation pattern for random air flow

Random air flow has three modes; Low random air flow, Medium random air flow, and High random air flow. The figure below shows the variation pattern for the random air flow. The air flow changes in units of 4 seconds, and one cycle is 36 seconds.



(3) Combining Sleeping Timer and random air flow

When random air flow is used in conjunction with Sleeping Timer, stop period is periodically inserted into Patterns ①, ② and ③ described in (2). The timing becomes as follows:

Up to 30 minutes

After two Patterns among ①, ②, and ③ have been activated, stops for 12 seconds.

① → ② → 12 sec stop → ③ → ① → 12 sec stop → ② → ③ →

30 minutes to 1 hour

After each of Patterns ①, ②, and ③ has been activated, stops for 12 seconds.

① → 12 sec stop → ② → 12 sec stop → ③ → 12 sec stop → ① →

1 to 1.5 hours

After each of Patterns ①, ②, and ③ has been activated, stops for 24 seconds.

① → 24 sec stop → ② → 24 sec stop → ③ → 24 sec stop → ① →

1.5 to 2 hours

After each of Patterns ①, ②, and ③ has been activated, stops for 36 seconds.

① → 36 sec stop → ② → 36 sec stop → ③ → 36 sec stop → ① →

2 to 3 hours

After each of Patterns ①, ②, and ③ has been activated, stops for 48 seconds.

① → 48 sec stop → ② → 48 sec stop → ③ → 48 sec stop → ① →

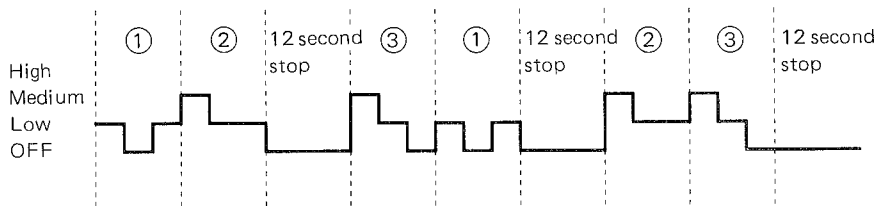
3 to 4 hours

After each of Patterns ①, ②, and ③ has been activated, stops for 60 seconds.

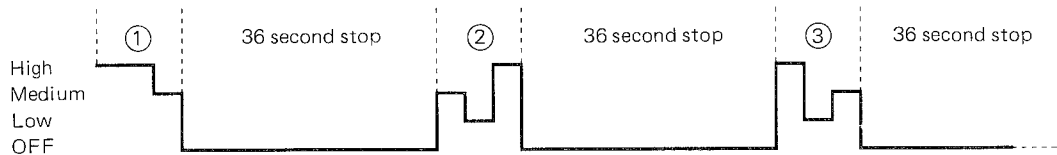
① → 60 sec stop → ② → 60 sec stop → ③ → 60 sec stop → ① →

The following shows examples of air flow change, when random air flow is used in conjunction with the Sleeping Timer:

Example 1: Low random air flow up to 30 minutes



Example 2: High random air flow during 1.5 to 2 hours



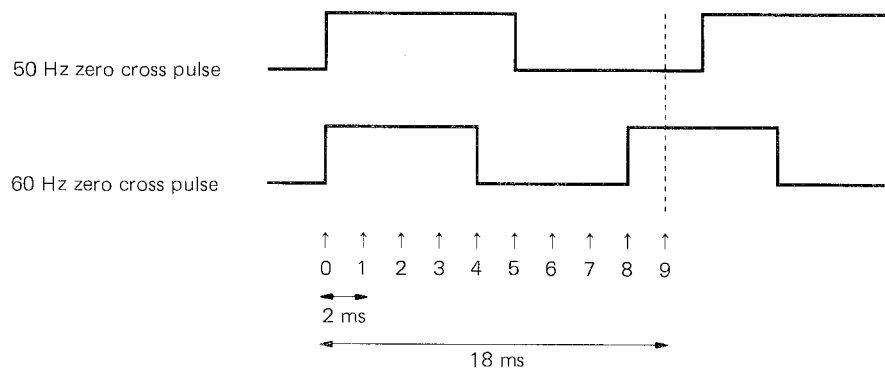
CHAPTER 4 CONTROL PROGRAM

4.1 ALGORITHM

The steps in this program are adjusted in such a manner that the main routine is repeated in 2 ms. In each cycle, key scan signal and LED digit signal outputs are switched.

Zero cross pulses from the POB0 pin are counted to generate the period of Timer, Sleeping Timer, and the random air flow patterns. If the line frequency is 50 Hz, 200 counts are equivalent to 4 seconds. 240 counts are required to generate 4 seconds, when the line frequency is 60 Hz.

50 Hz/60 Hz decision is made in the following manner during initial setting.

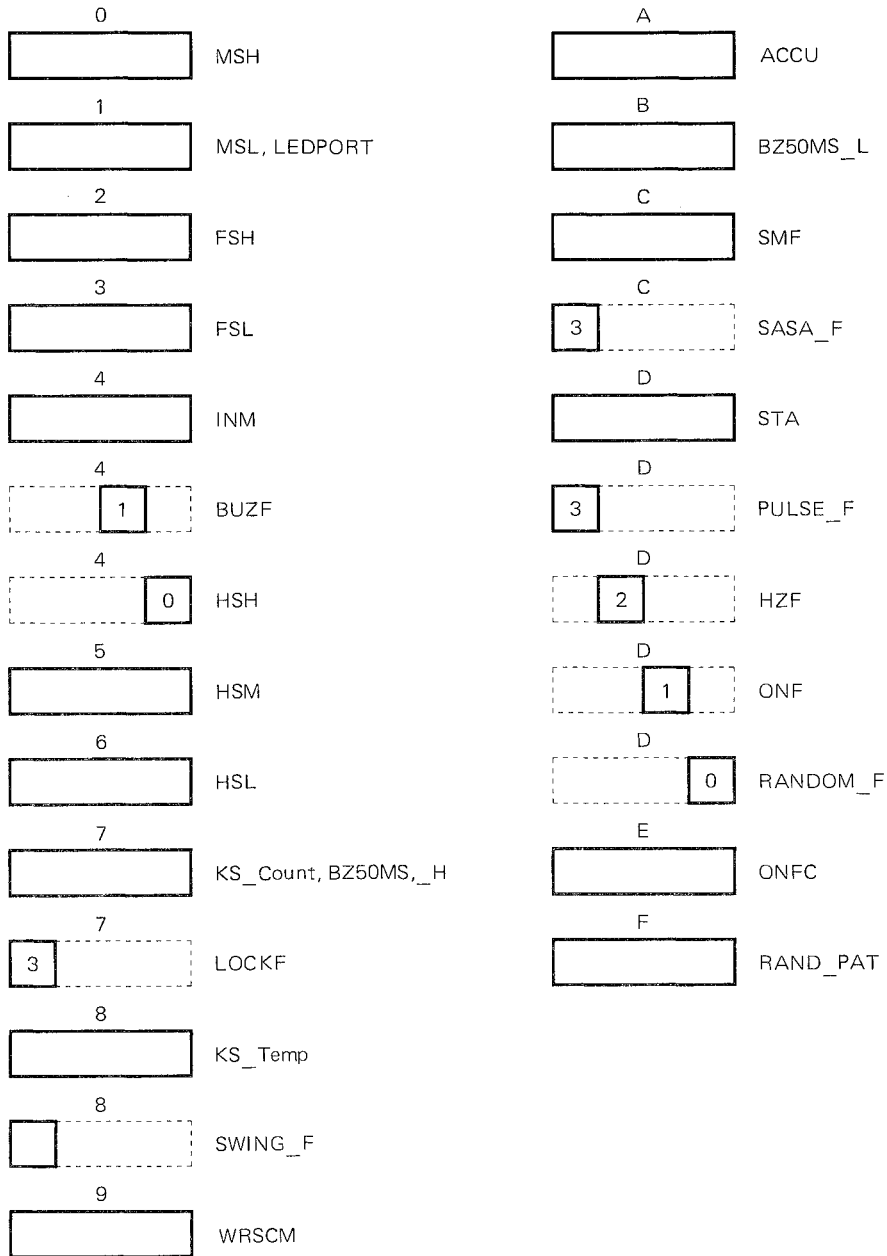


The POB0 pin level is checked every 2 ms after the zero cross pulse changes from "L" to "H". If the pin level is "L", when the 9th check is made (after 18 ms), the frequency is 50 Hz. If it is "H", the frequency is determined to be 60 Hz.

The TRIAC, which drives the motor, is not controlled in synchronization with the AC frequency, but is controlled with DC level.

4.2 RAM MAP (Description of Symbols and Flags)

Address 0X



Address 7X

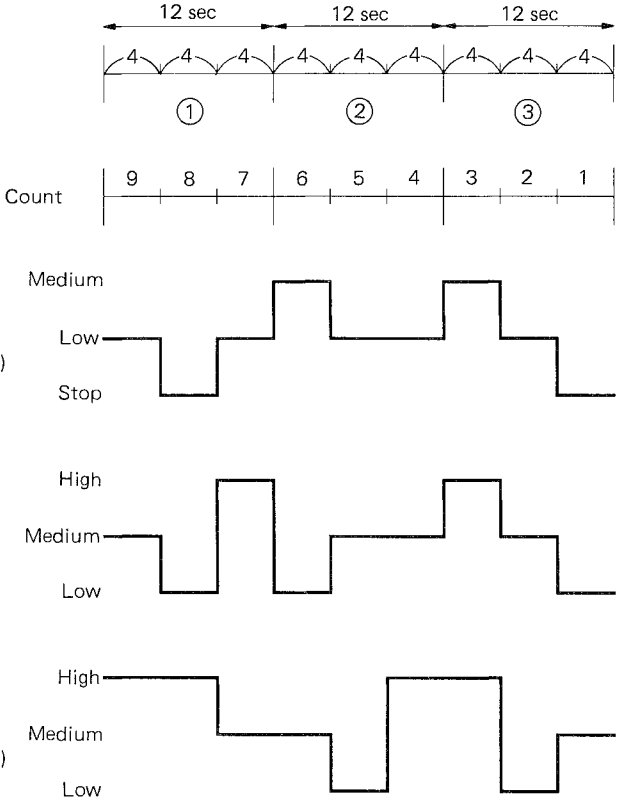


Description of symbols and flags

Symbol and flag	Name	Description																														
BUZF	Buzzer flag	The flag is set when initial setting is made or when a key input is determined to be valid. The flag is reset after 50 ms buzzer output.																														
LOCKF	Lock flag	The flag is set after 36 ms (chattering protect time) are counted or when multiple keys are depressed. The flag is reset when the key is released.																														
SWING_F	Head swing flag	This flag is set during head swing operation.																														
SASA_F	Sleeping Timer flag	This flag is set during the Sleeping Timer mode.																														
PULSE_F	Pulse flag	This flag is set when zero cross pulse is "H", and reset when it is "L".																														
HZF	HZ flag	This flag is set when zero cross pulse is 60 Hz.																														
ONF	ON/OFF flag	This flag is set during the Normal mode, Timer mode, or Sleeping Timer mode operation. This flag is reset during stop periods for the Sleeping Timer mode.																														
RANDOM_F	Random flag	This flag is set during the random air flow mode.																														
MSH	2 ms counter (upper)	These are used as a temporary counter, in order to adjust the main routine time to 2 ms, and to count 124 μ s of buzzer output. Additionally, these are used to count 30 ms before initiating determination of 50 Hz/60 Hz after power-on setting, and to count 18 ms to determine whether the line frequency is 50 Hz or 60 Hz. In this case, 1 count is made every 5 instruction executions (10 μ s).																														
MSL	2 ms counter (lower)																															
LEDPORT	LED signal output register	This is a temporary register, which temporarily stores the LED segment data.																														
FSH	4 sec. counter (upper)	This counter counts 4 seconds to control time for the Timer, Sleeping Timer, and random air flow. 1 count is made every zero cross pulse. Additionally, these are used to count 30 ms before initiating determination of 50 Hz/60 Hz after power-on setting, and to count 18 ms to determine whether the line frequency is 50 Hz or 60 Hz. In this case, 1 count is made every 2 ms (after counting MSH, MSL).																														
FSL	4 sec. counter (lower)																															
INM	Sleeping Timer set period register	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>3</td> <td>2</td> <td>x</td> <td>x</td> <td></td> </tr> <tr> <td>:</td> <td>:</td> <td>:</td> <td>:</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>x</td> <td>x</td> <td>Unused</td> </tr> <tr> <td>0</td> <td>1</td> <td>x</td> <td>x</td> <td>Sleeping Timer set to 1 hr</td> </tr> <tr> <td>1</td> <td>0</td> <td>x</td> <td>x</td> <td>Sleeping Timer set to 2 hrs</td> </tr> <tr> <td>1</td> <td>1</td> <td>x</td> <td>x</td> <td>Sleeping Timer set to 4 hrs</td> </tr> </table>	3	2	x	x		:	:	:	:		0	0	x	x	Unused	0	1	x	x	Sleeping Timer set to 1 hr	1	0	x	x	Sleeping Timer set to 2 hrs	1	1	x	x	Sleeping Timer set to 4 hrs
3	2	x	x																													
:	:	:	:																													
0	0	x	x	Unused																												
0	1	x	x	Sleeping Timer set to 1 hr																												
1	0	x	x	Sleeping Timer set to 2 hrs																												
1	1	x	x	Sleeping Timer set to 4 hrs																												

Symbol and flag	Name	Description						
HSH	30 min. counter (upper)	Timer, Sleeping Timer time control counter. Counts one every 4 seconds (after counting FSL, FSH)						
HSM	30 min. counter (middle)							
HSL	30 min. counter (lower)							
KS_Count	Key scan counter	<p>Chattering time counter. Lower 3 bits of address 07H are used. Port is checked every 6 ms. When "H" is counted six time, the key input is recognized as valid input, then the lock flag is set. This counter is cleared when the key is released.</p> <p>The diagram shows a horizontal timeline with six 2ms intervals. Each interval contains two downward-pointing pulses: 'Key scan 1' and 'Key scan 2'. Below the timeline, a double-headed arrow indicates a 6ms interval, with 'First' and 'Second' labels above it.</p>						
BZ50MS_H	Buzzer 50 ms counter (upper)	Bit 0 of address 07H is used.	Counts buzzer output time 50 ms. When the buzzer flag is set, counts one every 2 ms.					
BZ50MS_L	Buzzer 50 ms counter (lower)							
KS_Temp	Key input status register	<p>This register indicates which key is pressed.</p> <table border="1" style="margin-left: 20px;"> <tr> <td>x</td> <td>2</td> <td>1</td> <td>0</td> <td></td> </tr> </table> <ul style="list-style-type: none"> x 0 0 0 No key input x 0 0 1 Air flow key x 0 1 0 Random key x 0 1 1 Head swing key x 1 0 0 Multiple key depressing x 1 0 1 Timer key x 1 1 0 Sleeping Timer key x 1 1 1 OFF key 		x	2	1	0	
x	2	1	0					

Symbol and flag	Name	Description																																																																																															
WRSCM	Air flow status register LED scan mode register	<p>The upper 2 bits indicate the current air flow status, and the lower 2 bits indicate the status of the LED scan mode.</p> <table border="1"> <tr> <td>3</td><td>2</td><td>x</td><td>x</td> <td>x</td><td>x</td><td>1</td><td>0</td> </tr> <tr> <td>0</td><td>1</td><td>x</td><td>x</td> <td>Low</td> <td>x</td><td>x</td><td>0</td><td>1</td> <td>1st scan</td> </tr> <tr> <td>1</td><td>0</td><td>x</td><td>x</td> <td>Medium</td> <td>x</td><td>x</td><td>1</td><td>0</td> <td>2nd scan</td> </tr> <tr> <td>1</td><td>1</td><td>x</td><td>x</td> <td>High</td> <td>x</td><td>x</td><td>1</td><td>1</td> <td>3rd scan</td> </tr> </table>	3	2	x	x	x	x	1	0	0	1	x	x	Low	x	x	0	1	1st scan	1	0	x	x	Medium	x	x	1	0	2nd scan	1	1	x	x	High	x	x	1	1	3rd scan																																																									
3	2	x	x	x	x	1	0																																																																																										
0	1	x	x	Low	x	x	0	1	1st scan																																																																																								
1	0	x	x	Medium	x	x	1	0	2nd scan																																																																																								
1	1	x	x	High	x	x	1	1	3rd scan																																																																																								
ACCU	General register	This register temporarily stores data.																																																																																															
SMF	Timer/ Sleeping Timer status register	<p>This register indicates the remaining time of the Timer and Sleeping Timer. When other than in the Normal mode, the contents are decremented (-1) every 30 minutes, and enters the standby mode when the specified time passed.</p> <table border="1"> <tr> <td>3</td><td>2</td><td>1</td><td>0</td> <td></td> <td></td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>0</td> <td>.....</td> <td>Timer</td> <td>0.5 h</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>1</td> <td>.....</td> <td rowspan="2">↓</td> <td>1 h</td> </tr> <tr> <td>0</td><td>0</td><td>1</td><td>0</td> <td>.....</td> <td>1.5 h</td> </tr> <tr> <td>0</td><td>0</td><td>1</td><td>1</td> <td>.....</td> <td>2 h</td> <td></td> </tr> <tr> <td>0</td><td>1</td><td>0</td><td>0</td> <td>.....</td> <td>Normal mode</td> <td></td> </tr> <tr> <td>1</td><td>0</td><td>0</td><td>0</td> <td>.....</td> <td>Sleeping Timer</td> <td>0.5 h</td> </tr> <tr> <td>1</td><td>0</td><td>0</td><td>1</td> <td>.....</td> <td rowspan="2">↓</td> <td>1 h</td> </tr> <tr> <td>1</td><td>0</td><td>1</td><td>0</td> <td>.....</td> <td>1.5 h</td> </tr> <tr> <td>1</td><td>0</td><td>1</td><td>1</td> <td>.....</td> <td>2 h</td> <td></td> </tr> <tr> <td>1</td><td>1</td><td>0</td><td>0</td> <td>.....</td> <td>2.5 h</td> <td></td> </tr> <tr> <td>1</td><td>1</td><td>0</td><td>1</td> <td>.....</td> <td>3 h</td> <td></td> </tr> <tr> <td>1</td><td>1</td><td>1</td><td>0</td> <td>.....</td> <td>3.5 h</td> <td></td> </tr> <tr> <td>1</td><td>1</td><td>1</td><td>1</td> <td>.....</td> <td>4 h</td> <td></td> </tr> </table>	3	2	1	0			0	0	0	0	Timer	0.5 h	0	0	0	1	↓	1 h	0	0	1	0	1.5 h	0	0	1	1	2 h		0	1	0	0	Normal mode		1	0	0	0	Sleeping Timer	0.5 h	1	0	0	1	↓	1 h	1	0	1	0	1.5 h	1	0	1	1	2 h		1	1	0	0	2.5 h		1	1	0	1	3 h		1	1	1	0	3.5 h		1	1	1	1	4 h	
3	2	1	0																																																																																														
0	0	0	0	Timer	0.5 h																																																																																											
0	0	0	1	↓	1 h																																																																																											
0	0	1	0		1.5 h																																																																																											
0	0	1	1	2 h																																																																																												
0	1	0	0	Normal mode																																																																																												
1	0	0	0	Sleeping Timer	0.5 h																																																																																											
1	0	0	1	↓	1 h																																																																																											
1	0	1	0		1.5 h																																																																																											
1	0	1	1	2 h																																																																																												
1	1	0	0	2.5 h																																																																																												
1	1	0	1	3 h																																																																																												
1	1	1	0	3.5 h																																																																																												
1	1	1	1	4 h																																																																																												
STA	Status flag register	<table border="1"> <tr> <td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td></td><td></td><td></td><td>→ RANDOM_F</td> </tr> <tr> <td></td><td></td><td></td><td>→ ONF</td> </tr> <tr> <td></td><td></td><td></td><td>→ HZF</td> </tr> <tr> <td></td><td></td><td></td><td>→ PULSE_F</td> </tr> </table>	3	2	1	0				→ RANDOM_F				→ ONF				→ HZF				→ PULSE_F																																																																											
3	2	1	0																																																																																														
			→ RANDOM_F																																																																																														
			→ ONF																																																																																														
			→ HZF																																																																																														
			→ PULSE_F																																																																																														

Symbol and flag	Name	Description				
ONFC	Sleeping Timer pattern counter	<p>Used for counting the operation time and stop time in the Sleeping Timer. The contents are decremented (-1) every 4 seconds, and ON/OFF is inverted when the contents become 0.</p> <table border="1" data-bbox="662 422 850 474"> <tr> <td>3</td> <td>2</td> <td>1</td> <td>0</td> </tr> </table> <p>0 0 1 1 12 seconds 0 1 1 0 24 seconds 1 0 0 1 36 seconds 1 1 0 0 48 seconds 1 1 1 1 60 seconds</p>	3	2	1	0
3	2	1	0			
RAND_PAT	Random pattern counter	<p>This counter exchanges random patterns. Initial value is 9, and the contents are decremented (-1) every 4 seconds. The value returns to 9 when the contents become 0.</p>  <p>Count: 9 8 7 6 5 4 3 2 1</p> <p>(Low random) Low, Stop, Medium</p> <p>(Medium random) High, Low</p> <p>(High random) High, Medium, Low</p>				

4.3 MEMORY MAP

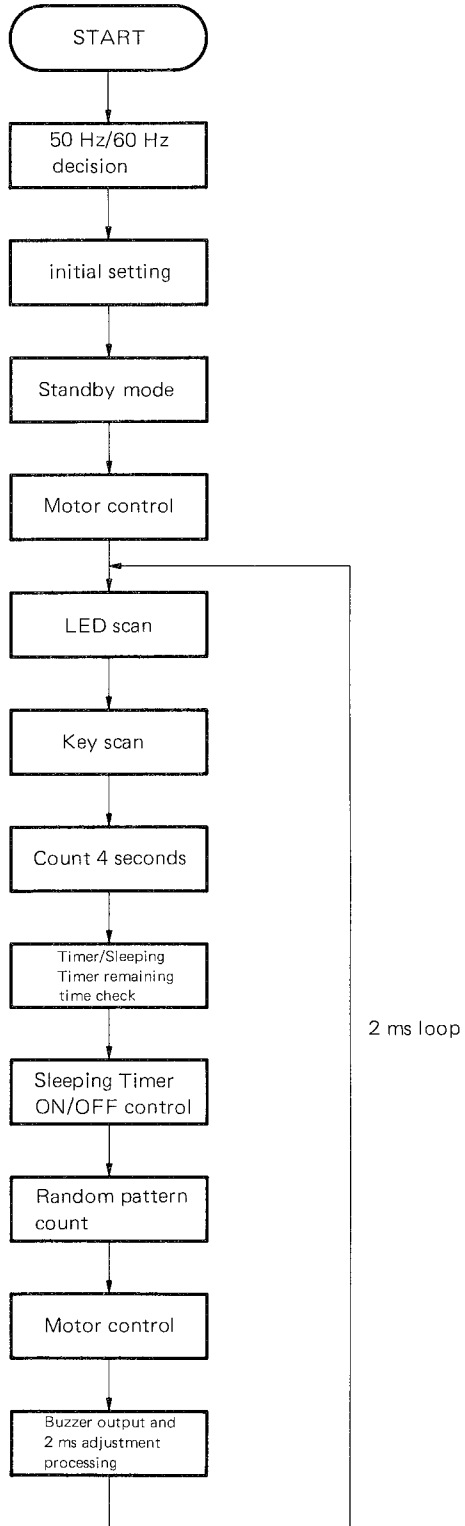
		Memoires used
0	30 ms delay	20
15	50 Hz/60 Hz decision	17
26	Initial setting	22
3C	LED scan and key input chattering time count processing	73
85	4 second count processing	22
9B	30 minute count and remaining time check	24
B3	Sleeping Timer ON/OFF inversion processing	32
D3	Random pattern count processing	11
DE	Input key processing	68
122	Motor control processing	43
14D	LED turning on processing	46
17B	Key input port check	21
190	Zero cross pulse count	10
19A	Buzzer output processing	23
1B1	2 ms adjustment and counter clear processing	21
1C6	(Unused)	
1FFH		

Total 453 words

4.4 GENERAL FLOW CHART

The figure below shows the general flow for this program.

Time adjustment subroutines or NOP instructions are inserted for this program, in various locations of each branch destination, so that the execution time for the main routine becomes 2 ms.



4.5 DETAILED FLOW CHART

4.5.1 30 ms Delay

(1)

Input variable	Flow chart	Processing, remarks	Output variable
MSL MSH	<pre> graph TD START([START]) --> B1[MSH ← 00H MSL ← 00H] B1 --> B2[FSH ← 00H FSL ← 00H] B2 --> D1{MSH = 0CH?} D1 -- Y --> D2{HSL = 08H?} D1 -- N --> B3[MSH ← 00H MSL ← 00H] D2 -- Y --> B3 D2 -- N --> B4["(MSH,MSL) ← (MSH,MSL)+1"] B4 --> D1 B3 --> B5[FSL ← FSL + 1] B5 --> D3{FSL = 0FH?} D3 -- Y --> B6[P0A ← 0100B P0C ← 1111B P0D ← 1111B] D3 -- N --> D1 B6 --> B7[MSH ← 00H MSL ← 00H] B7 --> B8[FSH ← 00H FSL ← 00H] B8 --> END1{ } END1 --> END2{ } </pre>	<p>Clears 2 ms counter</p> <p>Clears 4 ms counter</p> <p>10 ms (5 instructions) x 200 = 2 ms count</p> <p>Clears 2 ms counter</p>	MSH MSL MSH MSL MSH MSL
FSL	<p>FSL ← FSL + 1</p> <p>FSL = 0FH?</p> <p>P0A ← 0100B P0C ← 1111B P0D ← 1111B</p> <p>MSH ← 00H MSL ← 00H</p> <p>FSH ← 00H FSL ← 00H</p> <p>(2)</p> <p>1</p>	<p>2 ms x 15 counts? Yes: 30 ms has passed</p> <p>Sets output port to reset state</p> <p>Clears 2 ms counter</p>	FSL P0A P0C P0D MSH MSL FSH FSL

4.5.2 50 Hz/60 Hz Decision

(2)

Input variable	Flow chart	Processing, remarks	Output variable
<p>MSH MSL</p> <p>FSL</p>	<pre> graph TD 1((1)) --> D1{P0B0=L?} D1 --> D2{P0B0=H?} D2 --> D3{MSH=0CH?} D3 -- N --> I1[MSH,MSL ← (MSH,MSL) + 1] D3 -- Y --> D4{MSL=08H?} D4 -- N --> I1 D4 -- Y --> R1[MSH ← 00H MSL ← 00H] R1 --> I2[FSL ← FSL + 1] I2 --> D5{FSL=09H?} D5 -- N --> D3 D5 -- Y --> D6{P0B0=H?} D6 -- N --> I3[HZF ← 0] D6 -- Y --> I4[HZF ← 1] I3 --> I4 I4 --> 3[/3/] </pre>	<p>Waits until zero cross pulse becomes "L"</p> <p>Waits until zero cross pulse becomes "H" (Checks the point at which "L" changes to "H")</p> <p>Counts 2 ms</p> <p>2 ms x 9 counts? Yes: 18 ms has passed</p> <p>Sets HZF (60 Hz), if the level of the zero cross pulse is "H" when 18 ms has passed after it changed from "L" to "H", and reset HZF (50 Hz), if it is "L".</p>	<p>MSH</p> <p>MSH MSL</p> <p>MSH MSL</p> <p>FSL</p> <p>HZF</p>

4.5.3 Initial Setting

(3)

Input variable	Flow chart	Processing, remarks	Output variable
	<pre> graph TD Start((1)) --> Standby[P0A=0110B P0C=1111B P0D←1111B] Standby --> Stop[STOP0001B] Stop --> BUZF[BUZF←1] BUZF --> KS_Count[KS_Count ←1000B] KS_Count --> BZ50MS_L[BZ50MS_L ←00H] BZ50MS_L --> Pulse_Reset((2)) Pulse_Reset --> Normal_Mode_In[PULSE_F←0 RANDOM_F←0] Normal_Mode_In --> KS_Temp[KS_Temp ←1000B] KS_Temp --> WRSCM[WRSCM ←0101B] WRSCM --> FSH_FSL[FSH←00H FSL←00H] FSH_FSL --> HSH_HSM_HSL[HSH←××0B HSM←00H HSL←00H] HSH_HSM_HSL --> SMF[SMF←0100B] SMF --> End{4 1} </pre>	<p>Sets output port to the standby state</p> <p>Releases the standby mode at the rising edge of POB1 (On/ Air flow key input)</p> <p>Sets the buzzer flag</p> <p>Sets the lock flag and clears the key scan counter</p> <p>Clears the buzzer 50 ms counter</p> <p>Resets the pulse flag and random flag</p> <p>Sets head swing to OFF, and no input key</p> <p>Sets to Low air flow and 1st scan mode</p> <p>Clears 4 second counter</p> <p>Clears 30 minute counter</p> <p>Sets to the normal mode</p>	<p>P0A P0C P0D</p> <p>BUZF</p> <p>KS_Count</p> <p>BZ50MS_L</p> <p>PULSE_F RANDOM_F</p> <p>KS_Temp</p> <p>WRSCM</p> <p>FSH FSL</p> <p>HSH HSM HSL</p> <p>SMF</p>

(4)

Input variable	Flow chart	Processing, remarks	Output variable
	<pre> graph TD Start((1)) --> ONF1[COMMON : ONF ← 1] Input((2)) --> RAN_IN[RAN_IN : RAND_PAT ← 09H] ONF1 --> ONF1D{ONF ← 1} ONF1D -- Y --> FAN_ACTION[FAN ACTION] ONF1D -- N --> RAN_PAT_0AH[RAND_PAT ← 0AH] RAN_PAT_0AH --> FAN_ACTION FAN_ACTION --> Stop{5} </pre>	<p>Sets ON/OFF flag (sets to operation mode)</p> <p>Sets the random pattern count value to 9</p> <p>Operation mode?</p> <p>If it is the stop mode, resets the random pattern count value</p> <p>Motor control subroutine</p>	<p>ONF</p> <p>RAND_PAT</p> <p>RAND_PAT</p>

4.5.4 LED Scan

(5)

Input variable	Flow chart	Processing, remarks	Output variable
<p>LEDPORT</p> <p>WRSCM</p>	<pre> graph TD Start((1)) --> SCAN[SCAN: LEDPORT ← 1111B] SCAN --> P0D[P0D ← LEDPORT] P0D --> ACCU1[ACCU ← WRSCM] ACCU1 --> ACCU2[ACCU ← × × 00B] ACCU2 --> D1{WRSCM = × × × 1B?} D1 -- N --> C7((7)) D1 -- Y --> D2{WRSCM = × × 0 × B?} D2 -- N --> C8((8)) D2 -- Y --> LEDSCAN1[LEDSCAN1: P0A ← 0110B] LEDSCAN1 --> D3{BUZF=1?} D3 -- Y --> P0A1[P0A ← 1110B] D3 -- N --> C6((6)) P0A1 --> C6 C7 --> C6 C8 --> C6 </pre>	<p>Sets all LED segment outputs to "H" (turns all LEDs off)</p> <p>Stores the air flow condition to the general register</p> <p>1st or 3rd scan mode?</p> <p>1st scan mode?</p> <p>Sets to 1st scan mode</p> <p>Is buzzer flag set?</p> <p>Sets buzzer output port to "H"</p>	<p>LEDPORT</p> <p>P0D</p> <p>ACCU</p> <p>ACCU</p> <p>P0A</p> <p>P0A</p>

(6)

Input variable	Flow chart	Processing, remarks	Output variable
LEDPORT	<pre> graph TD Start((1)) --> D1{RANDOM_F = 1?} D1 -- Y --> P1[LEDPORT ← ×××0B] D1 -- N --> D2{ACCU = 1100B?} D2 -- Y --> P2[LEDPORT ← ××0×B] D2 -- N --> D3{SASA_F = 1?} D3 -- Y --> D4{SMF = 11×××B?} D4 -- Y --> P3[LEDPORT ← 0××××B] D4 -- N --> D5{SMF = 0010B or 0011B?} D5 -- Y --> P4[LEDPORT ← ×0×××B] D5 -- N --> P5[P0D ← LEDPORT] P5 --> P6[WRSCM ← ××10B] P6 --> D6{BUZF = 1?} D6 -- Y --> End1((10) 1) D6 -- N --> End2((9) 1) </pre>	<p>(1st LED scan)</p> <p>Random mode?</p> <p>Prepares random LED to turn on</p> <p>High air flow mode?</p> <p>Prepares High air flow LED to turn on</p> <p>Sleeping Timer mode?</p> <p>Is the remaining time of Timer 1 to 2 hrs?</p> <p>Is the remaining time of Sleeping Timer 2 to 4 hrs?</p> <p>Prepares 2 hr timer LED to turn on</p> <p>Prepares 4 hr Sleeping Timer LED to turn on</p> <p>Turn LED on</p> <p>Sets to 2nd LED scan mode</p> <p>Is the buzzer flag set?</p>	<p>LEDPORT</p> <p>LEDPORT</p> <p>LEDPORT</p> <p>LEDPORT</p> <p>P0D</p> <p>WRSCM</p>

(7)

Input variable	Flow chart	Processing, remarks	Output variable
LEDPORT	<pre> graph TD Start((1)) --> P0A1[P0A ← 0101B] P0A1 --> BUZF1{BUZF = 1?} BUZF1 -- Y --> P0A2[P0A ← 1101B] BUZF1 -- N --> ACCU{LED2: ACCU = 1000B?} ACCU -- Y --> LEDPORT1[LEDPORT ← ××0×B] ACCU -- N --> SASA{SASA F = 1?} SASA -- Y --> SMF1{SMF = 101×B?} SMF1 -- Y --> LEDPORT2[LEDPORT ← 0×××B] SMF1 -- N --> SMF2{SMF = 0001B?} SMF2 -- Y --> LEDPORT3[LEDPORT ← ×0××B] SMF2 -- N --> P0D[P0D ← LEDPORT] P0D --> WRSCM[WRSCM ← ××11B] WRSCM --> BUZF2{BUZF = 1?} BUZF2 -- Y --> End1{14 1} BUZF2 -- N --> End2{10 1} </pre>	<p>(2nd LED scan)</p> <p>Sets to 2nd scan mode?</p> <p>Is the buzzer flag set?</p> <p>Sets the buzzer output port to "H"</p> <p>Medium air flow mode?</p> <p>Prepares Medium air flow LED to turn on</p> <p>Sleeping Timer mode?</p> <p>Is the remaining time of Timer 30 min. to 1 hrs?</p> <p>Is the remaining time of Sleeping Timer 1 to 2 hrs?</p> <p>Prepares 1 hr timer LED to turn on</p> <p>Prepares 2 hr Sleeping Timer LED to turn on</p> <p>Turn LED on</p> <p>Sets to 3rd LED scan mode</p> <p>Is the buzzer flag set?</p>	<p>P0A</p> <p>P0A</p> <p>LEDPORT</p> <p>LEDPORT</p> <p>LEDPORT</p> <p>LEDPORT</p> <p>P0D</p> <p>WRSCM</p>

(8)

Input variable	Flow chart	Processing, remarks	Output variable
LEDPORT	<pre> graph TD Start((1)) --> LEDSCAN3[P0A ← 0011B] LEDSCAN3 --> BUZF{BUZF = 1?} BUZF -- Y --> P0A_H[P0A ← 1011B] BUZF -- N --> LED3{LED3: ACCU = 0100B?} LED3 -- Y --> LEDPORT_1[LEDPORT ← ××0×B] LED3 -- N --> SASA_F{SASA F = 1?} SASA_F -- Y --> SMF_100{SMF = 100×B?} SMF_100 -- Y --> LEDPORT_0[LEDPORT ← 0×××B] SMF_100 -- N --> SMF_0000{SMF = 0000B?} SMF_0000 -- Y --> LEDPORT_2[LEDPORT ← ×0××B] SMF_0000 -- N --> P0D[P0D ← LEDPORT] LEDPORT_1 --> P0D LEDPORT_0 --> P0D LEDPORT_2 --> P0D P0D --> WRSCM[WRSCM ← ××01B] WRSCM --> End{14 1} </pre>	<p>(3rd LED scan)</p> <p>Sets to 3rd scan mode?</p> <p>Is the buzzer flag set?</p> <p>Sets the buzzer output port to "H"</p> <p>Low air flow mode?</p> <p>Prepares Low air flow LED to turn on</p> <p>Sleeping timer mode?</p> <p>Is the remaining time of Timer 30 min. or less?</p> <p>Is the remaining time of Sleeping Timer 1 hr or less?</p> <p>Prepares 30 minute timer LED to turn on</p> <p>Prepares 1hr Sleeping Timer LED to turn on</p> <p>Turn LED on</p> <p>Sets to 1st scan mode</p>	<p>P0A</p> <p>P0A</p> <p>LEDPORT</p> <p>LEDPORT</p> <p>LEDPORT</p> <p>LEDPORT</p> <p>P0D</p> <p>WRSCM</p>

4.5.5 Key Scan

(9)

Input variable	Flow chart	Processing, remarks	Output variable
P0B	<pre> graph TD Start((I)) --> Process1[KEYSCAN1: KS_Temp ← ×000B] Process1 --> Process2[ACCU ← P0B] Process2 --> Process3[ACCU ← ×××0B] Process3 --> Decision1{ACCU = 0000B?} Decision1 -- Y --> Conn1((14) 1) Decision1 -- N --> Decision2{ACCU = 0010B?} Decision2 -- Y --> Process4[KS_Temp ← ×××1B] Process4 --> Conn2((14) 1) Decision2 -- N --> Decision3{ACCU = 0100B?} Decision3 -- Y --> Process5[KS_Temp ← ×××1×B] Process5 --> Conn3((14) 1) Decision3 -- N --> Decision4{ACCU = 1000B?} Decision4 -- Y --> Process6[KS_Temp ← ×××11B] Process6 --> Conn4((14) 1) Decision4 -- N --> Process7[LOCKF ← 1] Process7 --> Process8[KS_Temp ← ×××1××B] Process8 --> Conn5((14) 1) </pre>	<p>1st scan</p> <p>Sets to no key input state</p> <p>Checks key input port</p> <p>Resets bit 0 only (checks key input only)</p> <p>No key input?</p> <p>Air flow key input?</p> <p>Temporarily stores air flow key input information</p> <p>Random key input?</p> <p>Temporarily stores random key input information</p> <p>Head swing key input?</p> <p>Temporarily stores head swing key input information</p> <p>Regards multiple keyings, and sets the lock flag</p> <p>Sets to multiple keying state</p>	<p>KS_Temp</p> <p>ACCU</p> <p>ACCU</p> <p>KS_Temp</p> <p>KS_Temp</p> <p>KS_Temp</p> <p>KS_Temp</p> <p>LOCKF</p> <p>KS_Temp</p>

(11)

Input variable	Flow chart	Processing, remarks	Output variable
P0B	<p>Multi_key_CHK :</p> <pre> graph TD Start((1)) --> ACCU_P0B[ACCU ← P0B] ACCU_P0B --> ACCU_mask[ACCU ← × × × 0] ACCU_mask --> ACCU_zero{ACCU = 0000B?} ACCU_zero -- Y --> Exit13[/ (13) 1 /] ACCU_zero -- N --> LOCKF[LOCKF ← 1] LOCKF --> Exit14[/ (14) 1 /] </pre>	<p>Multiple keying check</p> <p>Key input port check</p> <p>Resets bit 0 only (checks key input only)</p> <p>No key input?</p> <p>Regards multiple Keyings, and sets the lock flag</p>	<p>ACCU</p> <p>ACCU</p> <p>LOCKF</p>

(12)

Input variable	Flow chart	Processing, remarks	Output variable
	<p>KEY_RELEASE :</p> <pre> graph TD Start((1)) --> Decision{KS_Temp = X000B?} Decision --> Process[KS_Count = 0000B] Decision --> Process Process --> End[/ (4) 1 /] </pre>	<p>No key input?</p> <p>Clears the key scan counter ('H' level check counter for key input port) Also resets the lock flag simultaneously</p>	<p>KS_Count</p>

(13)

Input variable	Flow chart	Processing, remarks	Output variable
KS_Count	<p>KEY_EFFECT_CHK :</p> <pre> graph TD Start((1)) --> D1{LOCKF = 1?} D1 -- Y --> Exit1[/ (14) /] D1 -- N --> P1[KS_Count ← KS_Count + 1] P1 --> D2{KS_Count = 06H?} D2 -- Y --> P2[LOCKF ← 1] P2 --> P3[BUZF ← 1] P3 --> Exit2[/ (14) /] D2 -- N --> Exit3[/ (14) /] </pre>	<p>Is lock flag set? (Is the key input already recognized?)</p> <p>Key scan counter + 1</p> <p>Is the key scan count value 6?</p> <p>Recognizes the key input and sets the lock flag</p> <p>Sets the buzzer flag</p>	<p>LOCKF</p> <p>KS_Count</p> <p>LOCKF</p> <p>BUZF</p>

4.5.6 Buzzer Output and 2 ms Adjustment Process

(14)

Input variable	Flow chart	Processing, remarks	Output variable
<p>P0A3</p> <p>MSH</p> <p>MSL</p>	<pre> graph TD 1((1)) --> BUZF1{BUZ_F = 1?} BUZF1 -- Y --> P0A3_0[P0A3 ← 0] BUZF1 -- N --> SMF{SMF = 0100B?} SMF -- Y --> RAND_F{RANDOM_F = 1?} RAND_F -- Y --> 10{10} RAND_F -- N --> 2((2)) 2 --> MSL_0[MSL ← 00H MSH ← 00H] MSL_0 --> BUZF2{BUZ_F = 1?} BUZF2 -- Y --> P0A3_inv[P0A3 ← ~P0A3] BUZF2 -- N --> MSH_inc[MSH ← MSH + 1] P0A3_inv --> MSH_inc MSH_inc --> MSL_inc[MSL ← MSL + 1] MSL_inc --> MSL_0E{MSL = 0EH?} MSL_0E -- Y --> MSL_00[MSL ← 00H] MSL_0E -- N --> BUZF2 MSL_00 --> MSH_0E{MSH = 0EH?} MSH_0E -- Y --> 15{15} MSH_0E -- N --> BUZF2 </pre>	<p>Is buzzer flag set?</p> <p>Sets buzzer output port to "L" level</p> <p>Normal mode?</p> <p>Random mode?</p> <p>Clears 2 ms counter</p> <p>Is buzzer flag set?</p> <p>Inverts buzzer output port level</p> <p>If buzzer is ON, outputs a 248 μs frequency pulse If buzzer is OFF, adjusts time</p>	<p>P0A3</p> <p>MSL MSH</p> <p>P0A3</p> <p>MSH</p> <p>MSL</p> <p>MSL</p>

(15)

Input variable	Flow chart	Processing, remarks	Output variable
<p>BZ50MS_L BZ50MS_H</p>	<pre> graph TD Start((1)) --> BUZF{BUZF = 1?} BUZF -- N --> C5_1((5)) BUZF -- Y --> Inc["(BZ50MS_L, BZ50MS_H) ← (BZ50MS_L, BZ50MS_H) + 1"] Inc --> BH{BZ50MS_H = ××0BH?} BH -- Y --> C5_2((5)) BH -- N --> BL{BZ50MS_L = 09H?} BL -- Y --> ClearH["BZ50MS_H ← ××0B"] ClearH --> ResetBUZF["BUZF ← 0"] ResetBUZF --> ClearL["BZ50MS_L ← 00H"] ClearL --> C2_1((2)) BL -- N --> C5_3((5)) C5_1 --> C5_2 C5_2 --> C5_3 C5_3 --> C2_1 </pre>	<p>Is buzzer flag set?</p> <p>Increments (+1) the buzzer 50 ms counter</p> <p>2 ms x 25 counts? Yes: 50 ms has passed</p> <p>Clears the upper 50 ms counter</p> <p>Resets buzzer flag</p> <p>Clears the lower 50 ms counter</p>	<p>BZ50MS_L BZ50MS_H</p> <p>BZ50MS_H</p> <p>BUZF</p> <p>BZ50MS_L</p>

4.5.7 4 sec Count

(16)

Input variable	Flow chart	Processing, remarks	Output variable
FSH FSL	<pre> graph TD Start((1)) --> D1{P0B0 = 1?} D1 -- Y --> P1[PULSE_F ← 1] D1 -- N --> P2[PULSE_F ← 0] P1 --> D2{PULSE_F = 1?} D2 -- Y --> C14_2_1{ (14) 2 } D2 -- N --> P3[PULSE_F ← 1] P3 --> P4["(FSH, FSL) ← (FSH, FSL) + 1"] P4 --> D3{HZF = 1?} D3 -- Y --> D4{FSH = 0FH, FSL = 00H?} D3 -- N --> D5{FSH = 0CH, FSL = 08H?} D4 -- Y --> P5["FSH ← 00H, FSL ← 00H"] D4 -- N --> C14_2_2{ (14) 2 } D5 -- Y --> P5 D5 -- N --> C14_2_3{ (14) 2 } P5 --> D6{SMF = 0100B?} D6 -- Y --> C18_1_1{ (18) 1 } D6 -- N --> C17_1_1{ (17) 1 } </pre>	<p>Is zero cross pulse "H"?</p> <p>If "L", resets the pulse flag</p> <p>The pulse flag already set?</p> <p>Sets the pulse flag</p> <p>Increments (+1) the 4 second counter</p> <p>Is HZF set? (60 Hz)</p> <p>Have 4 seconds passed? (50 Hz)</p> <p>Have 4 seconds passed? (60 Hz)</p> <p>Clears 4 second counter</p> <p>Normal/random mode?</p>	<p>PULSE_F</p> <p>PULSE_F</p> <p>FSH FSL</p> <p>FSH FSL</p>

4.5.8 Timer, Sleeping Timer Remaining Time Check

(17)

Input variable	Flow chart	Processing, remarks	Output variable
<p>HSH HSM HSL</p>	<pre> graph TD Start((1)) --> Process1["(HSH, HSM, HSL) ← (HSH, HSM, HSL) + 1"] Process1 --> Dec1{"HSH = 0B?"} Dec1 -- Y --> Conn1["(18)"] Dec1 -- N --> Dec2{"HSM = 0CH?"} Dec2 -- Y --> Dec3{"HSL = 02H?"} Dec2 -- N --> Conn1 Dec3 -- Y --> Process2["HSH ← 0B HSM ← 00H HSL ← 00H"] Dec3 -- N --> Conn1 Process2 --> Process3["SMF ← SMF - 1"] Process3 --> Dec4{"SMF = 11B?"} Dec4 -- Y --> Conn2["(3)"] Dec4 -- N --> Conn1 Conn1 --> Conn1 Conn2 --> Conn2 </pre>	<p>Increments (+1) 30 minute counter</p> <p>Have 30 minutes passed?</p> <p>Clears 30 minute counter</p>	<p>HSH HSM HSL</p>
<p>SMF</p>	<pre> graph TD Process3 --> Dec4{"SMF = 11B?"} Dec4 -- Y --> Conn2["(3)"] Dec4 -- N --> Conn1["(18)"] Conn1 --> Conn1 Conn2 --> Conn2 </pre>	<p>Decrements (-1) the Timer and Sleeping Timer remaining time counter</p> <p>No time remaining?</p>	<p>SMF</p>

4.5.9 Sleeping Timer ON/OFF Control

(18)

Input variable	Flow chart	Processing, remarks	Output variable
ONFC	<pre> graph TD Start((1)) --> D1{SASA_F = 1?} D1 -- N --> J1((20)) D1 -- Y --> P1[ONFC ← ONFC - 1] </pre>	<p>Sleeping Timer mode?</p> <p>Decrement (-1) ON/OFF counter</p>	ONFC
ONF	<pre> graph TD P1 --> D2{ONFC = 00H?} D2 -- N --> J2((20)) D2 -- Y --> P2[ONF ← ONF] </pre>	<p>ON/OFF switching required?</p> <p>Inverts ON/OFF flag</p>	ONF
SMF	<pre> graph TD P2 --> P3[ACCU ← SMF] </pre>	<p>Stores the Sleeping Timer remaining time into the general register</p>	ACCU
ACCU	<pre> graph TD P3 --> P4[ACCU ← ACCU] </pre>	<p>Inverts all bits of the general register</p>	ACCU
ACCU	<pre> graph TD P4 --> D3{INM = 1x1xB?} D3 -- Y --> P5[ACCU ← ACCU - 0100B] D3 -- N --> D4{INM = 1x1xB?} </pre>	<p>1 hr or 4 hr Sleeping Timer?</p> <p>Checks the time passed from the 2 h Sleeping Timer was started</p>	ACCU
ACCU	<pre> graph TD P5 --> D4{INM = 1x1xB?} D4 -- Y --> P6[ACCU ← ACCU - 0110B] D4 -- N --> D5{ONF = 1?} </pre>	<p>4 hr Sleeping Timer?</p> <p>Checks the time passed from the 4 h Sleeping Timer was started</p>	ACCU
ONFC	<pre> graph TD D5 -- N --> J3((19)) D5 -- Y --> P7[ONFC ← 03H] </pre>	<p>Operation mode?</p> <p>Sets the ON/OFF counter value to 3 (12 second operation)</p>	ONFC
ONFC	<pre> graph TD P7 --> D6{ACCU = 0000B?} D6 -- Y --> P8[ONFC ← 06H] D6 -- N --> J4((20)) </pre>	<p>Within 30 minutes from the start of operation?</p> <p>Sets the ON/OFF counter value to 6 (24 second operation)</p>	ONFC

(19)

Input variable	Flow chart	Processing, remarks	Output variable
	<pre> graph TD Start((1)) --> D1{ACCU = 0000B?} D1 -- Y --> P1[ONFC ← 03H] D1 -- N --> D2{ACCU = 0001B?} D2 -- Y --> P1 D2 -- N --> D3{ACCU = 0010B?} D3 -- Y --> P2[ONFC ← 06H] D3 -- N --> D4{ACCU = 0011B?} D4 -- Y --> P3[ONFC ← 09H] D4 -- N --> D5{ACCU = 0100B?} D5 -- Y --> P4[ONFC ← 0CH] D5 -- N --> D6{ACCU = 0101B?} D6 -- Y --> P4 D6 -- N --> D7{ACCU = 0110B?} D7 -- Y --> P5[ONFC ← 0FH] D7 -- N --> D8{ACCU = 0111B?} D8 -- Y --> P5 D8 -- N --> End[/20/] End --> End2[/2/] </pre>	<p>Sleeping Timer stop mode?</p> <p>Within 30 minutes from the start of operation?</p> <p>30 min. to 1 hr?</p> <p>Sets the ON/OFF counter value to 3 (12 second stop)</p> <p>1 to 1.5 hrs?</p> <p>Sets the ON/OFF counter value to 6 (24 second stop)</p> <p>1.5 to 2 hrs?</p> <p>Sets the ON/OFF counter value to 9 (36 second stop)</p> <p>2 to 2.5 hrs?</p> <p>2.5 to 3 hrs?</p> <p>Sets the ON/OFF counter value to 12 (48 second stop)</p> <p>3 to 3.5 hrs?</p> <p>3.5 to 4 hrs?</p> <p>Sets the ON/OFF counter value to 15 (60 second stop)</p>	<p>ONFC</p> <p>ONFC</p> <p>ONFC</p> <p>ONFC</p> <p>ONFC</p> <p>ONFC</p>

4.5.10 Random Pattern Count

(20)

Input variable	Flow chart	Processing, remarks	Output variable
<p>RAND_PAT</p>	<pre> graph TD Start((1)) --> SetONF[NON_SASA: ONF ← 1] SetONF --> Join(()) Circle2((2)) --> Join Join --> IsRandom{RANDOM: RANDOM_F = 1?} IsRandom -- Y --> IsONF{ONF = 1?} IsONF -- Y --> DecRand[RAND PAT ← RAND_PAT - 1] DecRand --> IsZero{RAND_PAT = 00H?} IsZero -- Y --> Set9[RAND_PAT ← 09H] IsZero -- N --> FanAction[FAN ACTION] IsONF -- N --> FanAction IsRandom -- N --> FanAction Set9 --> FanAction FanAction --> End{14 2} </pre>	<p>Sets ON/OFF flag (sets to operation mode)</p> <p>Random mode?</p> <p>Is ON/OFF flag set?</p> <p>Decrements (-1) the random pattern counter</p> <p>Is random pattern counter 0?</p> <p>Returns the random pattern counter to 9</p> <p>Motor control subroutine</p>	<p>ONF</p> <p>RAND_PAT</p> <p>RAND_PAT</p>

4.5.11 Input Key Process

(21)

Input variable	Flow chart	Processing, remarks	Output variable
KS_Temp	<pre> graph TD Start((1)) --> Process1[ACCU ← KS_Temp] Process1 --> Process2[ACCU ← 0x x B] Process2 --> Decision1{ACCU = 0110B?} Decision1 -- Y --> Exit22[22 1] Decision1 -- N --> Decision2{ACCU = 0111B?} Decision2 -- Y --> Exit3[3 1] Decision2 -- N --> Decision3{ACCU = 0001B?} Decision3 -- Y --> Exit24[24 1] Decision3 -- N --> Decision4{ACCU = 0010B?} Decision4 -- Y --> Exit25[25 1] Decision4 -- N --> Decision5{ACCU = 0011B?} Decision5 -- Y --> Exit26[26 1] Decision5 -- N --> Decision6{ACCU = 0000B?} Decision6 -- Y --> Exit3_2[3 2] Decision6 -- N --> Exit27[27 1] </pre>	<p>Stores only key input information into the general register</p> <p>Sleeping Timer key input?</p> <p>Off key input?</p> <p>Air flow key input?</p> <p>Random key input?</p> <p>Head swing key input?</p> <p>Initial operation?</p>	<p>ACCU</p> <p>ACCU</p>

(22)

Input variable	Flow chart	Processing, remarks	Output variable
	<pre> graph TD 1((1)) --> SAPR[SAPR: INM ← 00 × B] SAPR --> SASA{SASA_F = 1?} SASA -- Y --> 2((2)) SASA -- N --> INM2[INM ← 01 × B] INM2 --> SMF[SMF ← 1001B] SMF --> ONFC[INITSASA: ONFC ← 06H] ONFC --> INITTIMER[INITTIMER: FSH ← 00H FSL ← 00H] INITTIMER --> HSH[HSH ← × × 0B HSM ← 00H HSL ← 00H] HSH --> 4{4} 2 --> ONFC 3((3)) --> INITTIMER </pre>	<p>Resets the Sleeping Timer set period register</p> <p>Sleeping Timer mode?</p> <p>Sets 1 hr sleeping Timer</p> <p>Sets 1 hr as the remaining time</p> <p>Sets the ON/OFF counter value to 6 (24 second operation)</p> <p>Clears 4 second counter</p> <p>Clears 30 minute counter</p>	<p>INM</p> <p>INM</p> <p>SMF</p> <p>ONFC</p> <p>FSH FSL</p> <p>HSH HSM HSL</p>

(23)

Input variable	Flow chart	Processing, remarks	Output variable
	<pre> graph TD Start((1)) --> D1{SMF = 1000B?} D1 -- Y --> P1[INM ← 10 × B] D1 -- N --> D2{SMF = 1001B?} D2 -- Y --> P2[SMF ← 1011B] D2 -- N --> P3[SALD2H: INM ← 10 × B] P3 --> P4[SMF ← 1011B] P4 --> D3{SMF = 1010B?} D3 -- Y --> P5[INM ← 11 × B] D3 -- N --> D4{SMF = 1011B?} D4 -- Y --> P6[SMF ← 1111B] D4 -- N --> P7[SMF ← 0100B] P5 --> Exit2((2)) P6 --> Exit2 P7 --> Exit4((4)) D3 --> Exit1((1)) </pre>	<p>Is the remaining time within 30 minutes?</p> <p>Is the remaining time within 30 minutes to 1 hr?</p> <p>Sets 2 hrs Sleeping Timer</p> <p>Sets 2 hrs as the remaining time</p> <p>Is the remaining time within 1 hr to 1.5 hr?</p> <p>Is the remaining time within 1.5 hr to 2 hrs?</p> <p>Sets 4 hr Sleeping Timer</p> <p>Sets 4 hrs as the remaining time</p> <p>Sets the Normal mode</p>	<p>INM</p> <p>SMF</p> <p>INM</p> <p>SMF</p> <p>SMF</p>

(24)

Input variable	Flow chart	Processing, remarks	Output variable
WRSCM	<pre> graph TD Start((1)) --> Decision{WRSCM = 11 x B?} Decision -- Y --> Process1[WRSCM ← 01 x B] Decision -- N --> Process2[WRSCM ← WRSCM + 0100B] Process1 --> Process3[FAN ACTION] Process2 --> Process3 Process3 --> Connector((5)) Connector --> Terminator{1} </pre>	<p>High air flow mode?</p> <p>Sets the low air mode</p> <p>Low → Medium Medium → High</p> <p>Motor control subroutine</p>	<p>WRSCM</p> <p>WRSCM</p>

(25)

Input variable	Flow chart	Processing, remarks	Output variable
RANDOM_F	<pre> graph TD Start((1)) --> Process[RANDOM_F ← RANDOM_F] Process --> End{ (4) 2 } </pre>	Inverts the random flag	RANDOM_F

(26)

Input variable	Flow chart	Processing, remarks	Output variable
SWING_F	<pre> graph TD Start((1)) --> Process1[SWINGPR : SWING_F ←SWING_F] Process1 --> Process2[FAN ACTION] Process2 --> End{5 1} </pre>	<p>Inverts the head swing flag</p> <p>Motor control subroutine</p>	SWING_F

(27)

Input variable	Flow chart	Processing, remarks	Output variable
	<pre> graph TD Start((1)) --> D1{SMF = 0100B?} D1 -- Y --> P1[SMF ← 0000B] D1 -- N --> D2{SASA_F = 1?} D2 -- Y --> P1 D2 -- N --> D3{SMF = 0000B?} D3 -- Y --> P2[SMF ← 0001B] D3 -- N --> D4{SMF = 0001B?} D4 -- Y --> P3[SMF ← 0011B] D4 -- N --> P4[SMF ← 0100B] P1 --> E2((2)) P2 --> E3((3)) P3 --> E3 P4 --> E4((4)) E1((1)) </pre>	<p>Timer key input processing</p> <p>Normal mode?</p> <p>Sleeping Timer mode?</p> <p>Sets Timer mode to 30 minutes</p> <p>30 minutes Timer mode?</p> <p>Sets Timer mode to 1 hr</p> <p>1 hr timer mode?</p> <p>Sets Timer mode to 2 hr</p> <p>Sets to Normal mode</p>	<p>SMF</p> <p>SMF</p> <p>SMF</p> <p>SMF</p>

4.5.12 Motor Control

(28)

Input variable	Flow chart	Processing, remarks	Output variable
	<pre> graph TD Start([FAN ACTION]) --> ONF{ONF=1?} ONF -- Y --> RAND_F{RANDOM_F=1?} ONF -- N --> C31_1[/31 1/] RAND_F -- Y --> C31_1 RAND_F -- N --> RAND_PAT_09[RAND_PAT ← 09H] RAND_PAT_09 --> WRSCM_1{WRSCM = 1 × × B?} WRSCM_1 -- Y --> C29_1[/29 1/] WRSCM_1 -- N --> WRSCM_0{WRSCM = 0 × × B?} WRSCM_0 -- Y --> ACCU_0110[ACCU ← 0110B] WRSCM_0 -- N --> C30_1[/30 1/] ACCU_0110 --> RAND_PAT_08{RAND_PAT = 08H?} RAND_PAT_08 -- Y --> C31_1 RAND_PAT_08 -- N --> RAND_PAT_01{RAND_PAT = 01H?} RAND_PAT_01 -- Y --> ACCU_0111[ACCU ← 0111B] RAND_PAT_01 -- N --> RAND_PAT_06{RAND_PAT = 06H?} RAND_PAT_06 -- Y --> C31_1 RAND_PAT_06 -- N --> RAND_PAT_03{RAND_PAT = 03H?} RAND_PAT_03 -- Y --> ACCU_0101[ACCU ← 0101B] RAND_PAT_03 -- N --> C31_2[/31 2/] </pre>	<p>Operation mode?</p> <p>Random mode?</p> <p>If not random mode, sets 9 as the pattern count value</p> <p>Low air flow or High air flow mode?</p> <p>Low air flow mode?</p> <p>Stores Low air flow pattern into the general register</p> <p>Is the random pattern count value 8?</p> <p>Is the random pattern count value 1?</p> <p>Stores the stop pattern into the general register</p> <p>Is the random pattern count value 6?</p> <p>Is the random pattern count value 3?</p> <p>Stores the Medium air flow pattern into the general register</p> <p>* Low air flow pattern when the random pattern count value is 2, 4, 5, 7, or 9</p>	<p>RAND_PAT</p> <p>ACCU</p> <p>ACCU</p> <p>ACCU</p>

(29)

Input variable	Flow chart	Processing, remarks	Output variable
	<pre> graph TD Start((1)) --> P1[ACCU ← 0101B] P1 --> D1{RAND_PAT = 07H?} D1 -- Y --> P2[ACCU ← 0011B] D1 -- N --> D2{RAND_PAT = 03H?} D2 -- Y --> P2 D2 -- N --> D3{RAND_PAT = 08H?} D3 -- Y --> P3[ACCU ← 0110B] D3 -- N --> D4{RAND_PAT = 06H?} D4 -- Y --> P3 D4 -- N --> D5{RAND_PAT = 01H?} D5 -- Y --> P4[ACCU ← 0110B] D5 -- N --> P5[31] P2 --> P5 P3 --> P5 P4 --> P5 P5 --> End{2} </pre>	<p>(Medium, Medium random air flow output)</p> <p>Stores Medium air flow pattern into the general register</p> <p>Is the random pattern count value 7?</p> <p>Is the random pattern count value 3?</p> <p>Stores the High air flow pattern into the general register</p> <p>Is the random pattern count value 8?</p> <p>Is the random pattern count value 6?</p> <p>Stores the Low air flow pattern into the general register</p> <p>Is the random pattern count value 1?</p> <p>Stores the Low air flow pattern into the general register</p> <p>* Medium air flow pattern when the random pattern count value is 2, 4, 5, or 9</p>	<p>ACCU</p> <p>ACCU</p> <p>ACCU</p> <p>ACCU</p>

(30)

Input variable	Flow chart	Processing, remarks	Output variable
	<pre> graph TD Start((1)) --> P1[ACCU ← 0011B] P1 --> D1{RAND_PAT = 05H?} D1 -- Y --> P2[ACCU ← 0110B] D1 -- N --> D2{RAND_PAT = 02H?} D2 -- Y --> P2 D2 -- N --> D3{RAND_PAT = 07H?} D3 -- Y --> P3[ACCU ← 0101B] D3 -- N --> D4{RAND_PAT = 06H?} D4 -- Y --> P3 D4 -- N --> D5{RAND_PAT = 01H?} D5 -- Y --> P4[ACCU ← 0101B] D5 -- N --> End{31 2} </pre>	<p>(High, High random air flow output)</p> <p>Stores High air flow pattern into the general register</p> <p>Is the random pattern count value 5?</p> <p>Is the random pattern count value 2?</p> <p>Stores the Low air flow pattern into the general register</p> <p>Is the random pattern count value 7?</p> <p>Is the random pattern count value 6?</p> <p>Stores the Medium air flow pattern into the general register</p> <p>Is the random pattern count value 1?</p> <p>Stores the Medium air flow pattern into the general register</p> <p>* High air flow pattern when the random pattern count value is 3, 4, 8, or 9</p>	<p>ACCU</p> <p>ACCU</p> <p>ACCU</p> <p>ACCU</p>

(31)

Input variable	Flow chart	Processing, remarks	Output variable
<p>ACCU</p>	<pre> graph TD 1((1)) --> S1[STPOP: ACCU ← 0111B] S1 --> D1{SWING_F = 0?} 2((2)) --> D1 D1 -- N --> S2[ACCU ← 1 × × × B] D1 -- Y --> S3[P0C ← ACCU] S2 --> S3 S3 --> D2{ONF = 0?} D2 -- Y --> S4[NOP × 8] D2 -- N --> S5([RET]) S4 --> S5 </pre>	<p>Stores the stop pattern into the general register</p> <p>Head swing ON?</p> <p>Sets to head swing OFF mode</p> <p>Outputs the general register contents to the motor control port</p> <p>Stop mode?</p> <p>Counts $2 \mu\text{s} \times 8 = 16 \mu\text{s}$ (time adjustment)</p> <p>Return</p>	<p>ACCU</p> <p>ACCU</p> <p>P0C</p>

4.6 SOURCE PROGRAM LIST

```

80      ;
81      :*****
82      :                                     *
83      :      ASSEMBLER DEF                                     *
84      :                                     *
85      :*****
86      0070      POA      MEM 0. 70H      :PORT A
87      0071      POB      MEM 0. 71H      :PORT B
88      0072      POC      MEM 0. 72H      :PORT C
89      0073      POD      MEM 0. 73H      :PORT D
90      ;
91      0001      LEDPORT MEM 0. 01H      :LED OUTPUT TEMPORARY REG.
92      0000      MSH      MEM 0. 00H      :2ms_Counter_High_Nibble
93      0001      MSL      MEM 0. 01H      :2ms_Counter_Low_Nibble
94      0002      FSH      MEM 0. 02H      :4Sec_Counter_High_Nibble
95      0003      FSL      MEM 0. 03H      :4Sec_Counter_Low_Nibble
96      0004      INM      MEM 0. 04H
97      0004      HSH      MEM 0. 04H      :30Minutes_Counter_High_Nibble
98      0005      HSM      MEM 0. 05H      :30Minutes_Counter_Mid_Nibble
99      0006      HSL      MEM 0. 06H      :30Minutes_Counter_Low_Nibble
100     0007      KS_Count MEM 0. 07H      :KeyScan_Counter_REG.
101     0007      BZ50MS_H MEM 0. 07H      :BUZZ_50ms_Counter_HIGH_NIBBLE
102     0008      KS_Temp  MEM 0. 08H      :KeyScan_Temporary_REG.
103     0009      WRSCM   MEM 0. 09H      :Wind_Status & KeyScan_Mode REG.
104     000A      ACCU    MEM 0. 0AH      :ACCUMULATOR
105     000B      BZ50MS_L MEM 0. 0BH      :BUZZ_50ms_Counter_LOW_NIBBLE
106     000C      SMF     MEM 0. 0CH      :Status_Mode
107     000D      STA     MEM 0. 0DH      :Status_Flag_REG.
108     000E      ONFC    MEM 0. 0EH      :SASATIMER_ON_OFF_PATTERN_REG.
109     000F      RAND_PAT MEM 0. 0FH      :RANDOM_PATTERN_REG.
110     ;
111     0088      SWING_F  FLG  KS_Temp. 3      :SWING_FLAG
112     0042      BUZF    FLG  HSH. 1          :BUZZ_ON_FLAG
113     0078      LOCKF   FLG  KS_Count. 3     :LOCK_MODE_FLAG
114     ;
115     00C8      SASA_F  FLG  SMF. 3          :SASA_FLAG
116     ;
117     00D8      PULSE_F  FLG  STA. 3         :PULSE_FLAG
118     00D4      HZF     FLG  STA. 2         :50/60Hz_MODE_FLAG
119     00D2      ONF     FLG  STA. 1         :ON_OFF_MODE_FLAG
120     00D1      RANDOM_F FLG  STA. 0        :RANDOM_MODE_FLAG
121     ;
122     :*****
123     :                                     *
124     :      4 Sec & 30 Minutes Adjustment                                     *
125     :                                     *
126     :*****
127     :      50HZ MODE
128     :-----0
129     : SET "HZ50_4secADJ" VALUE ==> : 199 (C7H) <---- 200 (C8H) ----> 201 (C9H) ;
130     : : : ^ ^ ;
131     : 4Sec_Counter SPEED ==> : FAST Standard SLOW ;
132     :-----0
133     ;
134     :      60HZ MODE
135     :-----0
136     : SET "HZ60_4secADJ" VALUE ==> : 239 (EFH) <---- 240 (F0H) ----> 241 (F1H) ;
137     : : : ^ ^ ;
138     : 4Sec_Counter SPEED ==> : FAST Standard SLOW ;
139     :-----0
140     ;

```


CHAPTER 4 CONTROL PROGRAM

```

141          :      30 Minutes ADJ
142          :
143          :-----o
144          : SET "Minute_30ADJ" VALUE ==> : 449 (1C1H) <--- 450 (1C2) ---> 451 (1C3) :
145          :                               :                               ^                               :
146          :      4Sec_Counter SPEED ==> :      FAST          Standard          SLOW          :
147          :-----o
148          :
149 000C      HZ50_4secADJH DAT 0CH :50 Hz MODE : 4sec Counter ADJ HIGH
150 0008      HZ50_4secADJL DAT 08H :50 Hz MODE : 4sec Counter ADJ LOW
151 000F      HZ60_4secADJH DAT 0FH :60 Hz MODE : 4sec Counter ADJ HIGH
152 0000      HZ60_4secADJL DAT 00H :60 Hz MODE : 4sec Counter ADJ LOW
153 0002      Minute_30ADJ DAT 02H :30 Minutes Counter ADJ
154          :
155 0003      HI DAT 0011B :PORTC HIGH OUTPUT VALUE
156 0005      MID DAT 0101B :PORTC MID OUTPUT VALUE
157 0006      LOW DAT 0110B :PORTC LOW OUTPUT VALUE
158 0007      STP DAT 0111B :PORTC STOP OUTPUT VALUE
159          :
160          :
161          :*****
162          :                               *
163          :      FAN MAIN PROGRAM                               *
164          :                               *
165          :*****
166          :
167          :
168          :*****
169          :      POWER ON DELAY                               *
170          :*****
171 0000 074F0      NOP                               :POWER ON DELAY
172 0001 074F0      NOP                               :DELAY WAITING FOR STABLE STATUS
173 0002 074F0      NOP
174 0003 074F0      NOP
175 0004 074F0      NOP
176 0005 074F0      NOP
177 0006 074F0      NOP
178 0007 074F0      NOP
179 0008 074F0      NOP
180 0009 074F0      NOP
181 000A 074F0      NOP
182 000B 1C1BC      CALL MS_H_L_CLR :INIT MS REG.
183 000C 1C1BF      CALL FS_H_L_CLR :INIT FS REG.
184          :
185          POWER_DELAY:
186 000D 1C1B1      CALL WAIT_2MS :DELAY 2MS
187 000E 1C1BC      CALL MS_H_L_CLR :RESET MS REG.
188 000F 10031      ADD FSL,#01H :INC 1
189 0010 0903F      SKE FSL,#0FH :2ms * 15 = 30ms ENDING ?
190 0011 0C00D      BR POWER_DELAY
191          :*****
192          :      POWER ON RESET                               *
193          :*****
194 0012 1D704      MOV POA,#0100B :POWER ON RESET
195 0013 1D72F      MOV POC,#1111B
196 0014 1D73F      MOV POD,#1111B
197          :*****
198          :      50/60 Hz DECISION                               *
199          :*****
200          :
201 0015 1C1BC      CALL MS_H_L_CLR :CLR 2ms_COUNTER_REG
202 0016 1C1BF      CALL FS_H_L_CLR :INIT FS REG.
203          HZ_INIT:
204          1      SKF1 POB0 :TEST ZEROCROSS BEGIN WITH "LOW" STATUS
+ 1 0017 1F711 1      SKF .MF,POB0 SHR 4.#.DF,POB0 AND 0FH
205 0018 0C017      BR HZ_INIT :"HIGH" STATUS RETEST AGAIN
206          :

```

CHAPTER 4 CONTROL PROGRAM

```

207          HZIN:
208          1          SKT1  POB0          :READ Zerocrossing PULSE 1st "HIGH"
+ 1 0019 1E711 1      SKT   .MF.POB0 SHR 4.#.DF.POB0 AND 0FH
209 001A 0C019        BR    HZIN          :REPEAT READ
210 001B 1D030        MOV    FSL.#00H     :INITIAL COUNTER_9 VALUE=0
211          ;
212          HZ_WAIT:          :DELAY 18ms
213 001C 1C1B1        CALL   WAIT_2MS       :DELAY 2ms FOR COUNT
214 001D 1C1BC        CALL   MS_H_L_CLR     :CLR 2ms_Counter_REG
215 001E 10031        ADD    FSL.#01H       :COUNT 1 PER 2ms
216 001F 09039        SKE   FSL.#09H       :COUNTER_9= 9 ? (2ms * 9 = 18ms )
217 0020 0C01C        BR    HZ_WAIT       :REPEAT WAIT 2ms
218          ;
219          1          SKT1  POB0          :CHK 9th PULSE
+ 1 0021 1E711 1      SKT   .MF.POB0 SHR 4.#.DF.POB0 AND 0FH
220 0022 0C025        BR    HZ50          :9th Pulse =" LOW ".SET 50Hz FLAG
221          1          SET1  HZF          :9th Pulse ="HIGH ".SET 50Hz FLAG
+ 1 0023 160D4 1      OR    .MF.HZF SHR 4.#.DF.HZF AND 0FH
222 0024 0C026        BR    STANDBY       :BRANCH STANDBY MODE
223          HZ50:
224          1          CLR1  HZF          :HZF=0 IN 50Hz
+ 1 0025 140DB 1      AND   .MF.HZF SHR 4.#.DF.(NOT HZF AND 0FH)
225          ;
226          :*****
227          :   STANDBY MODE          *
228          :*****
229          STANDBY:          :STANDBY MODE INITIAL OUTPUT PORT
230 0026 1D706        MOV    POA.#0110B
231 0027 1D72F        MOV    POC.#1111B
232 0028 1D73F        MOV    POD.#1111B
233 0029 072F1        STOP   0001B
234          :*****
235          :   INITIAL          *
236          :*****
237          1          SET1  BUZF          :WHEN "ON" KEY PRESSED.SET BUZZ_ON
+ 1 002A 16042 1      OR    .MF.BUZF SHR 4.#.DF.BUZF AND 0FH
238 002B 1D078        MOV    KS_Count.#1000B :SET "LOCK" FLAG & CLR KEYSKAN_COUNTER
239 002C 1D0B0        MOV    BZ50MS_L.#00H   :INIT BUZZ_50ms_COUNTER
240          ;
241          NORMAL_MODE_IN:      :1st "ON" KEY RETURN HERE
242          1          CLR2  PULSE_F_RANDOM_F :CLR PULSE_FLAG & RANDOM_FLAG
+ 1 002D 140D6 1      AND   .MF.PULSE_F SHR 4.#.DF.(NOT (PULSE_F OR RANDOM_F) AND 0FH)
243 002E 1D088        MOV    KS_Temp.#1000B  :RESET SWING_FLAG & KEYSKAN_STATUS_REG.
244 002F 1D095        MOV    WRSCM.#0101B   :INIT "LOW" Wind & "1st" SCAN MODE
245 0030 1C1BF        CALL   FS_H_L_CLR     :INIT 4Sec_Counter
246 0031 1C1C2        CALL   HS_H_M_L_CLR   :INIT 30Minutes_Counter
247          ;-----
248          :   KEYPROCESS RETURN POINT -
249          ;-----
250          NORMAL_INT:          :"NORMAL MODE" KEYPROCESS RETURN HERE
251 0032 1D0C4        MOV    SMF.#0100B     :RESET "STATUS_MODE_REG" =NORMAL MODE
252 0033 0C037        BR    COMMON         :SKIP NEXT LINES RESET INSTRUCTION
253          ;
254          INITSASA:          :"SASA_TIMER" KEYPROCESS RETURN HERE
255 0034 1D0E6        MOV    ONFC.#06H     :RESET ON_OFF_PATTERN_COUNTER=9
256          ;
257          INITTIMER:          :"TIMER" KEYPROCESS RETURN HERE
258 0035 1C1BF        CALL   FS_H_L_CLR     :RESET 4Sec_COUNTER
259 0036 1C1C2        CALL   HS_H_M_L_CLR   :RESET 30 Minutes_Counter
260          ;
261          COMMON:          :COMMON RESET ITEM
262          1          SET1  ONF          :SET ON_OFF_FLAG = ON
+ 1 0037 160D2 1      OR    .MF.ONF SHR 4.#.DF.ONF AND 0FH
263          ;
264          RAND_IN:          :"RANDOM" KEYPROCESS RETURN HERE
265 0038 1D0F9        MOV    RAND_PAT.#09H  :RESET RANDOM_PATTERN = 9
266          1          SKT1  ONF          :CHK OFF MODE

```

CHAPTER 4 CONTROL PROGRAM

```

+ 1 0039 1E0D2 1          SKT      . MF. ONF SHR 4. #. DF. ONF AND OFH
267 003A 1D0FA          MOV      RAND_PAT. #0AH          ;RESET RANDOM_PATTERN = A
268                      ;
269                      KEY_INT:                          ;WIND & SWING KEYPROCESS RETURN HERE
270                      :*****
271                      :   INIT ACTION                    *
272                      :*****
273 003B 1C122          CALL     ACTION          ;INIT ACTION
274                      ;;
275                      :*****
276                      :   #
277                      :   SCAN                          #
278                      :   #
279                      :*****
280                      SCAN:                                ;PROGRAM LOOP START HERE
281 003C 1D01F          MOV      LEDPORT. #1111B          ;INIT LEDPORT
282 003D 18731          ST       P0D. LEDPORT          ;RESET LED OUTPUT PORT_D
283                      ;
284 003E 0809A          LD       ACCU. WRSCM          ;LOAD WIND REG. FOR CHK
285 003F 140AC          AND     ACCU. #1100B          ;ACCU=11XX (WR)
286                      ;
287                      ;SCAN MODE DECISION
288 0040 1E091          SKT     WRSCM. #0001B          ;TEST IF 2nd SCAN_MODE (XXX1=XX01. XX11)
289 0041 0C054          BR     LEDSCAN2          ;BRANCH 2nd SCAN_MODE (XX10)
290 0042 1F092          SKF    WRSCM. #0010B          ;TEST IF 3rd SCAN_MODE (XX0X=XX00. XX01)
291 0043 0C07E          BR     LEDSCAN3          ;BRANCH 3rd SCAN_MODE (XX11)
292                      ;OTHER 1st SCAN_MODE (XX01)
293                      :*****
294                      :   LEDSCAN1  KEYSKAN1            *
295                      :*****
296                      ;LEDSCAN1:
297                      ;
298 0044 1D706          MOV     P0A. #0110B          ;SET 1st LEDSCAN OUTPUT PORT_A ACTION
299                      SKF1   BUZF          ;CHK IF BUZZ ON MODE
+ 1 0045 1F042 1          SKF    . MF. BUZF SHR 4. #. DF. BUZF AND OFH
300 0046 1D70E          MOV     P0A. #1110B          ;SET BUZZER PORT ON "HIGH"
301                      ;
302 0047 1C14D          CALL   LED1          ;CALL LEDSCAN1 SUBROUTINE
303                      ;
304 0048 1409C          AND     WRSCM. #1100B          ;SET 2nd SCAN_MODE ( WRSCM=XX10 )
305 0049 16092          OR     WRSCM. #0010B
306                      ;-----
307                      SKF1   BUZF          ;CHK IF BUZZ MODE
+ 1 004A 1F042 1          SKF    . MF. BUZF SHR 4. #. DF. BUZF AND OFH
308 004B 0C052          BR     COUNT1_1          ;BUZZ MODE. KEYSKAN SKIP
309                      ;KEYSCAN1:
310 004C 14088          AND     KS_Temp. #1000B          ;CLR KEYSKAN_STATUS =000
311                      ;
312 004D 1C17B          CALL   KEYSKAN          ;CALL KEYSKAN SUBROUTINE
313                      ;
314                      ;RETURN
315 004E 074F0          NOP          ;RETURN HERE
316                      ;RETURN SKIP
317 004F 1D017          MOV     MSL. #07H
318                      COUNT_ADJ:
319 0050 1C1B8          CALL   MSADJ          ;TIME DELAY ADJ
320 0051 0C085          BR     COUNT
321                      ;
322                      COUNT1_1:
323 0052 1D01C          MOV     MSL. #0CH
324 0053 0C050          BR     COUNT_ADJ
325                      :*****
326                      :   LEDSCAN2  KEYSKAN2            *
327                      :*****
328                      ;
329                      LEDSCAN2:
330 0054 1D705          MOV     P0A. #0101B          ;SET 2st LEDSCAN OUTPUT PORT_A ACTION

```

CHAPTER 4 CONTROL PROGRAM

```

331          1          SKF1 BUZF          :CHK IF BUZZ ON MODE
+ 1 0055 1F042 1      SKF      .MF. BUZF SHR 4. #. DF. BUZF AND OFH
332 0056 1D70D          MOV      POA. #1101B          :SET BUZZER PORT ON "HIGH"
333          ;
334 0057 1C15A          CALL     LED2          :CALL LEDSCAN2 SUBROUTINE
335          ;
336 0058 1409C          AND      WRSCM. #1100B          :SET 3rd SCAN_MODE ( WRSCM=XX11 )
337 0059 16093          OR       WRSCM. #0011B
338          ;-----
339          1          SKF1 BUZF          :CHK IF BUZZ MODE
+ 1 005A 1F042 1      SKF      .MF. BUZF SHR 4. #. DF. BUZF AND OFH
340 005B 0C052          BR       COUNT1_1          :BUZZ MODE. KEYSKAN SKIP
341          ;-----
342          :KEYSCAN2:
343 005C 1F087          SKF      KS_Temp. #0111B          :1ST KeyScan_Temporary=000 ?
344 005D 0C067          BR       Multi_Key_CHK          :1ST KeyScan_Temporary=NOT ZERO
345          ;
346 005E 1C17B          CALL     KEYSKAN          :CALL 2nd KEYSKAN SUBROUTINE
347          :RETURN
348 005F 0C062          BR       KEY_RELEASE          :1st & 2nd BOTH NO KEY PRESS==> RELEASE
349          :RETURN SKIP
350 0060 10084          ADD     KS_Temp. #0100B          :ADJ 2ND KeyScan_Temporary=1XX
351 0061 0C071          BR       KEY_EFFECT_CHK          :2nd KEYSKAN TAKE EFFECT
352          ;-----
353          KEY_RELEASE:
354 0062 1F087          SKF      KS_Temp. #0111B          :KEYSCAN_TEMP EMPTY CHK
355 0063 0C065          BR       KYRL          :NOT EMPTY
356 0064 1D070          MOV     KS_Count. #0000B          :NON KEY PRESS | RELEASE |
357          ;
358 0065 1D016          KYRL:  MOV     MSL. #06H          ;
359 0066 0C050          BR       COUNT_ADJ
360          ;-----
361          Multi_Key_CHK:
362 0067 0871A          LD       ACCU. POB          :LOAD 2nd KEYSKAN STATUS
363 0068 140AE          AND     ACCU. #1110B          :MASK 1110
364 0069 090A0          SKE     ACCU. #0000B          :CHK 2nd KEYSKAN=0 ?
365 006A 0C06E          BR       ERRKY          :2nd KEYSKAN <> 0 ==> Multi KEY PERSS
366          ; (ERROR KEY)
367 006B 1D013          MOV     MSL. #03H          :2ms COUNTER ADJ
368 006C 1C1B8          CALL     MSADJ
369 006D 0C071          BR       KEY_EFFECT_CHK          :1st KEYSKAN TAKE EFFECT
370          ERRKY:
371          1          SETI     LOCKF          :ERROR KEY LOCK
+ 1 006E 16078 1      OR       .MF. LOCKF SHR 4. #. DF. LOCKF AND OFH
372 006F 1D019          MOV     MSL. #09H
373 0070 0C050          BR       COUNT_ADJ
374          ;*****
375          : KEY_EFFECT_CHK          *
376          ;*****
377          KEY_EFFECT_CHK:
378          1          SKF1     LOCKF          :CHK IF KEY LOCK MODE ?
+ 1 0071 1F078 1      SKF      .MF. LOCKF SHR 4. #. DF. LOCKF AND OFH
379 0072 0C07A          BR       COUNT2_1          :KEY LOCK. BRANCH OUT |
380          ;
381 0073 10071          ADD     KS_Count. #01H          :KEY_SCAN_COUNTER + 1
382          ;
383 0074 09076          SKE     KS_Count. #06H          :KEY_SCAN_COUNTER = 6 ?
384 0075 0C07C          BR       COUNT2_2          :KEY_SCAN_COUNTER < 6. BRANCH OUT
385          ;
386          1          SETI     LOCKF          :LOCK KEY COUNT
+ 1 0076 16078 1      OR       .MF. LOCKF SHR 4. #. DF. LOCKF AND OFH
387          1          SETI     BUZF          :SET BUZZ_ON MODE
+ 1 0077 16042 1      OR       .MF. BUZF SHR 4. #. DF. BUZF AND OFH
388          ;
389 0078 1D013          MOV     MSL. #03H          :2ms ADJ
390 0079 0C050          BR       COUNT_ADJ
391          ;

```

CHAPTER 4 CONTROL PROGRAM

```

392          COUNT2_1:
393 007A 1D015          MOV     MSL.#05H          :2ms ADJ
394 007B 0C050          BR      COUNT_ADJ
395
396          COUNT2_2:
397 007C 1D014          MOV     MSL.#04H
398 007D 0C050          BR      COUNT_ADJ
399
400          :*****
401          :      LEDSCAN3          *
402          :*****
403
404          LEDSCAN3:
405 007E 1D703          MOV     POA.#0011B          :SET 3rd LEDSCAN OUTPUT PORT_A ACTION
406          :      SKF1     BUZF          :CHK IF BUZZ ON MODE
+ 1 007F 1F042 1          SKF     .MF.BUZF SHR 4.#.DF.BUZF AND OFH
407 0080 1D70B          MOV     POA.#1011B          :SET BUZZER PORT ON "HIGH"
408
409 0081 1C166          CALL    LED3          :CALL 3rd LEDSCAN SUBROUTINE
410          :-----
411 0082 1409C          AND     WRSCM.#1100B          :SET NEXT SCAN_MODE TO 1st
412 0083 16091          OR      WRSCM.#0001B          : (XX01)
413
414 0084 0C052          BR      COUNT1_1
415          :*****
416          :      COUNT          *
417          :*****
418          COUNT:          :COUNT 4 Sec ONLY
419
420          :      SKF1     BUZF          :CHK BUZZ_ON MODE ?
+ 1 0085 1F042 1          SKF     .MF.BUZF SHR 4.#.DF.BUZF AND OFH
421          :      CLR1     POA3          :BUZZ_ON MODE MIDDLE POINT SET
+ 1 0086 14707 1          AND     .MF.POA3 SHR 4.#.DF.(NOT POA3 AND OFH)
422          :          :SET BUZZ="LOW" (2ND)
423
424 0087 0B0C4          SKNE    SMF.#0100B          :NORMAL MODE ?
425          :      SKF1     RANDOM_F          :NORMAL RANDOM MODE ?
+ 1 0088 1F0D1 1          SKF     .MF.RANDOM_F SHR 4.#.DF.RANDOM_F AND OFH
426 0089 0C08B          BR      PULSE          :TIMER COUNTER MODE
427 008A 0C0AA          BR      COUNT_WAITADJ1          :NORMAL MODE ONLY.SKIP TO COUNT WAIT
428
429          :      PULSE_COUNT
430          :-----
431          PULSE:          :READ 50/60 Hz Pulse
432 008B 1C190          CALL    PULSE_COUNT
433
434          :RETURN
435 008C 0C0AC          BR      COUNT_WAITADJ2          :NON HZ PULSE SKIP TO COUNT WAIT
436          :RETURN SKIP
437 008D 10031          ADD     FSL.#01H          :4SEC+1
438 008E 12020          ADDC    FSH.#00H
439
440          :      FS 4sec Ending CHK
441          :-----
442          :      SKT1     HZF          :CHK 50/60 Hz MODE
+ 1 008F 1E0D4 1          SKT     .MF.HZF SHR 4.#.DF.HZF AND OFH
443 0090 0C095          BR      HZ50CHK          :HZF=0 BRANCH 50Hz CHK
444          :60HzCHK
445 0091 0B02F          SKNE    FSH.#HZ60_4secADJH          :CHK 60 pulse * 4 = 240 ?
446 0092 09030          SKE     FSL.#HZ60_4secADJL
447 0093 0C0AE          BR      COUNT_WAITADJ3          :< 4Sec NOTHING TO DO
448 0094 0C099          BR      FS_INIT          := 4Sec.BRANCH TO TIMER CHK
449
450          HZ50CHK:
451 0095 0B02C          SKNE    FSH.#HZ50_4secADJH          :CHK 50 pulse * 4 = 200 ?
452 0096 09038          SKE     FSL.#HZ50_4secADJL
453 0097 0C0AE          BR      COUNT_WAITADJ3          :< 4 Sec NOTHING TO DO

```

CHAPTER 4 CONTROL PROGRAM

```

454 0098 074F0          NOP                      ;2ms_Counter ADJ
455                    ;-----
456                    ;      4ms Ending
457                    ;-----
458                    FS_INIT:
459 0099 1C1BF          CALL   FS_H_L_CLR          ;4 Sec Ending RESET 4 Sec_Counter=0
460 009A 074F0          NOP                      ;2ms_Counter ADJ
461                    ;
462                    ;*****
463                    ;      TIMER                      *
464                    ;*****
465 009B 0B0C4          SKNE   SMF, #0100B          ;NORMAL RANDOM MODE?
466 009C 0C0B1          BR     SASACHKADJ1          ;NORMAL RANDOM MODE
467                    ;
468                    ;                      ;4Sec * 450 =1800sec
469 009D 100E1          ADD    HSL, #01H          ;30 MIN+1 (TIMER, SASA TIMER MODE)
470 009E 12050          ADDC   HSM, #00H
471 009F 12040          ADDC   HSH, #00H
472                    ;-----
473                    ;      HS 30 minutes Ending CHK
474                    ;-----
475                    ;                      ;1800Sec/4=450. (1C2H)=450
476 00A0 1F041          SKF    HSH, #01H          ;CHK 30 Minutes HIGH BIT (XXX0)
477 00A1 0905C          SKE   HSM, #0CH
478 00A2 0C0A4          BR     H2
479 00A3 09062          SKE   HSL, #Minute_30ADJ
480 00A4 0C0B2          H2:   BR     SASACHKADJ2          ;NOT YET 30 minutes SKIP TO SASACHK
481                    ;-----
482                    ;      30_Minutes Ending
483                    ;-----
484                    ;                      ;30 Minutes ENDING
485 00A5 1C1C2          CALL   HS_H_M_L_CLR          ;30 minutes REG (HS) RESEET
486                    ;-----
487                    ;      STATUS ENDING CHK
488                    ;-----
489                    ;                      ;CHECK SMF=END ?
490 00A6 110C1          SUB    SMF, #01H          ;SMF=SMF-1
491 00A7 1E0C7          SKT   SMF, #0111B          ;CHK ENDING
492 00A8 0C0B3          BR     SASACHK          ;SMF NOT yet = 0
493 00A9 0C026          BR     STANDBY          ;SMF=0
494                    ;-----
495                    ;      COUNT WAIT ADJ
496                    ;-----
497                    ;                      < 4sec COME HERE . NOTHING TO DO
498                    COUNT_WAITADJ1:          ;2ms_Cunter ADJ
499 00AA 1D01F          MOV    MSL, #0FH
500 00AB 0C0AF          BR     WAIT_ADJ
501                    COUNT_WAITADJ2:
502 00AC 1D01E          MOV    MSL, #0EH
503 00AD 0C0AF          BR     WAIT_ADJ
504                    COUNT_WAITADJ3:
505 00AE 1D01B          MOV    MSL, #0BH
506                    WAIT_ADJ:
507 00AF 1C1B8          CALL   MSADJ
508 00B0 0C0DB          BR     COUNT_WAIT
509                    ;-----
510                    ;      SASACHK WAIT ADJ
511                    ;-----
512                    ;                      ;TIMER COUNTER MODE , <30 minutes
513                    SASACHKADJ1:          ;2ms_Cunter ADJ
514 00B1 1C11C          CALL   ADJ6
515                    SASACHKADJ2:
516 00B2 0C0B3          BR     SASACHK
517                    ;;;
518                    ;*****
519                    ;      SASA CHK                      *
520                    ;*****

```

CHAPTER 4 CONTROL PROGRAM

```

521          SASACHK:
522          SKT1  SASA_F          :SASA MODE ?
+ 1 00B3 1E0C8 1          SKT  .MF.SASA_F SHR 4. #. DF. SASA_F AND 0FH
523 00B4 0C0D3          BR  NON_SASA          :NOT SASA MODE BRANCH TO RANDOM CHK
524          ;
525 00B5 110E1          SUB  ONFC.#01H          :SASA_ON_OFF_PATTERN_COUNTER - 1
526          ;
527 00B6 090E0          SKE  ONFC.#00H          :SASA_ON_OFF_PATTERN_COUNTER = 0 ?
528 00B7 0C0D4          BR  RANDOM          :SASA_ON_OFF_PATTERN_COUNTER <>0
529          ;BRANCH RANDOM CHK
530          ;-----
531          SASA_PATTERN_ENDING          :RESET SASA_PATTERN_COUNTER
532          ;-----
533          NOT1  ONF          :INVERT SASA_ON_OFF_FLAG
+ 1 00B8 150D2 1          XOR  .MF.ONF SHR 4. #. DF. ONF AND 0FH
534          ;-----
535          PATTERN_OFFSET
536          ;-----
537 00B9 080CA          LD  ACCU.SMF          :XOR SMF TO ACCU FOR SASA OFFSET
538 00BA 150AF          XOR  ACCU.#1111B
539          ;
540          ;OFFSET 4H. 2H. 1H
541          :SASA OFFSET DECISION
542 00BB 1E044          SKT  INM.#0100B          :X1XX=01XX(1H). 11XX(4H) SKIP
543          :OFFSET2H:
544 00BC 110A4          SUB  ACCU.#0100B          :OFFSET SASA 2H VALUE
545 00BD 1E048          SKT  INM.#1000B          :1XXX=10XX(2H). 11XX(4H) SKIP
546          :OFFSET1H:
547 00BE 110A6          SUB  ACCU.#0110B          :OFFSET SASA 1H VALUE
548          :OFFSET4H:
549          ;NON OFFSET
550          ;-----
551          :ON_OFF_CHK:
+ 1 00BF 1E0D2 1          SKT1  ONF          :SASA ON MODE ?
          SKT  .MF.ONF SHR 4. #. DF. ONF AND 0FH
552 00C0 0C0C5          BR  OFFACT          :SASA OFF MODE
553          ;
554          ;-----
555          SASA_ON_PATTERN_RESET
556          ;-----
557          :ONACT:
558 00C1 1D0E3          MOV  ONFC.#03H          :LOAD SASA ON 0.5H ----> 4H = 12 Sec
559 00C2 0B0A0          SKNE  ACCU.#0000B          :SASA ON 0.5H MODE ?
560 00C3 1D0E6          MOV  ONFC.#06H          :LOAD SASA ON 0.5H = 24 Sec
561 00C4 0C0D4          BR  RANDOM
562          ;
563          ;-----
564          SASA_OFF_PATTERN_RESET
565          ;-----
566          OFFACT:
567 00C5 090A0          SKE  ACCU.#0000B          : 0H --- 1H MODE ?
568 00C6 0B0A1          SKNE  ACCU.#0001B
569          :LD3:
570 00C7 1D0E3          MOV  ONFC.#03H          :LOAD 12 SEC
571          ;
572 00C8 0B0A2          SKNE  ACCU.#0010B          : 1H --- 1.5H MODE ?
573          :LD6:
574 00C9 1D0E6          MOV  ONFC.#06H          :SASA_TIMER_ON_OFF_PATTERN_COUNTER=6
575 00CA 0B0A3          SKNE  ACCU.#0011B          :1.5H --- 2H MODE ?
576          :LD9:
577 00CB 1D0E9          MOV  ONFC.#09H          :LOAD 36 SEC
578 00CC 090A4          SKE  ACCU.#0100B          :SASA_TIMER_ON_OFF_PATTERN_COUNTER=9
579 00CD 0B0A5          SKNE  ACCU.#0101B          : 2H --- 3H MODE ?
580          :LD12:
581 00CE 1D0EC          MOV  ONFC.#0CH          :LOAD 48 SEC
582 00CF 090A6          SKE  ACCU.#0110B          :SASA_TIMER_ON_OFF_PATTERN_COUNTER=12
583 00D0 0B0A7          SKNE  ACCU.#0111B          : 3H --- 4H MODE ?
584          :LD15:
          :LOAD 60 SEC

```

CHAPTER 4 CONTROL PROGRAM

```

585 00D1 1D0EF          MOV    ONFC, #0FH          :SASA_TIMER_ON_OFF_PATTERN_COUNTER=15
586 00D2 0C0D4          BR     RANDOM
587                      ;
588                      ;*****
589                      ;    RANDOM CHK          *
590                      ;*****
591                      NON_SASA:
592                      |          SETI   ONF          :EVERY NON_SASA MODE, SET ONF=1
+ 1 00D3 160D2 |          OR     .MF, ONF SHR 4, #.DF, ONF AND 0FH
593                      |          RANDOM:
594                      |          SKF1   RANDOM_F      :CHK RANDOM MODE ?
+ 1 00D4 1F0D1 |          SKF    .MF, RANDOM_F SHR 4, #.DF, RANDOM_F AND 0FH
595                      |          SKT1   ONF          :CHK SASA_OFF MODE ?
+ 1 00D5 1E0D2 |          SKT    .MF, ONF SHR 4, #.DF, ONF AND 0FH
596 00D6 0C0DA          BR     END_RANDOM      :NON_RANDOM & SASA_OFF
597                      :          :BRANCH OUT RANDOM_MODE
598                      ;-----
599                      ;          RANDOM MODE PROCESS
600                      ;-----
601 00D7 110F1          SUB    RAND_PAT, #01H    :RANDOM_PATTERN - 1 (RAND_PAT-1)
602                      ;
603 00D8 0B0F0          SKNE   RAND_PAT, #00H    :RANDOM_PATTERN ENDING ? (RAND_PAT=0 ?)
604 00D9 1D0F9          MOV    RAND_PAT, #09H   : ENDING! RESET RANDOM_PATTERN = 9
605                      :          : NOT ENDING BRANCH TO NEXT ACTION
606                      END_RANDOM:
607                      ;*****
608                      ;    OUTPUT ACTION          *
609                      ;*****
610                      :OUTPUT_ACTION:
611 00DA 1C122          CALL   ACTION          :MAKE ACTION AFTER EVERY 4 SEC ENDING
612                      ;*****
613                      ;    COUNT WAIT TO 2ms          *
614                      ;*****
615                      COUNT_WAIT:          :EVERY CHK ACTION OK, WAITING TO 2ms
616 00DB 1C19A          CALL   BUZZ_WAIT_2MS   :[1] :DO 126us BUZZ WORKING
617 00DC 0C03C          BR     SCAN          :BUZZER_ON_MODE < 50ms REPEAT SCAN
618 00DD 0C0DE          BR     KEYPROCESS     :BUZZER_ON_MODE = 50ms GOTO KEYPROCESS
619                      ;;;
620                      ;*****
621                      ;    KEYPROCESS          *
622                      ;*****
623                      KEYPROCESS:
624 00DE 0B08A          LD     ACCU, KS_Temp    :LOAD KEYSKAN TEMPORARY VALUE
625 00DF 140A7          AND    ACCU, #011B     :MASK
626                      ;-----
627                      ;    KEY STATUS DECISION
628                      ;-----
629 00E0 0B0A6          SKNE   ACCU, #0110B    : SASA KEY ?
630 00E1 0C0FA          BR     SAPR
631                      ;
632 00E2 0B0A7          SKNE   ACCU, #0111B    : CUT KEY ?
633 00E3 0C10D          BR     CUTPR
634                      ;
635 00E4 0B0A1          SKNE   ACCU, #0001B    : WIND KEY ?
636 00E5 0C10E          BR     WINPR
637                      ;
638 00E6 0B0A2          SKNE   ACCU, #0010B    : RANDOM KEY ?
639 00E7 0C115          BR     RANPR
640                      ;
641 00E8 0B0A3          SKNE   ACCU, #0011B    : SWING KEY ?
642 00E9 0C117          BR     SWINGPR
643                      ;
644 00EA 0B0A0          SKNE   ACCU, #0000B    : STNDBY KEY ?
645 00EB 0C119          BR     STANDPR
646                      ;-----
647                      ;    TIMER KEY PROCESS
648                      ;-----

```


CHAPTER 4 CONTROL PROGRAM

```

649          :TIMPR:
650 00EC 090C4      SKE   SMF.#0100B      :NORMAL MODE ?
651          SKFI   SASA_F      :SASA MODE ?
+ 1 00ED 1F0C8 1    SKF   .MF.SASA_F SHR 4. #. DF. SASA_F AND 0FH
652 00EE 0C0F8      BR    TIMADJ3      :NORMAL/SASA ==> 0.5H
653          :TIMADJ:
654 00EF 0B0C0      SKNE  SMF.#0000B      :0.5H MODE?
655 00F0 0C0F4      BR    TIMADJ1      :0.5H ==> 1H
656          ;
657 00F1 0B0C1      SKNE  SMF.#0001B      :1H MODE ?
658 00F2 0C0F6      BR    TIMADJ2      :1H ==> 2H
659          ;-----
660          :TIMADJ0:
661 00F3 0C032      BR    NORMAL_INT      :02H ==> NORMAL
662          ;
663          TIMADJ1:
664 00F4 1D0C1      MOV   SMF.#0001B      :0.5H ==> 01H
665 00F5 0C035      BR    INITTIMER
666          ;
667          TIMADJ2:
668 00F6 1D0C3      MOV   SMF.#0011B      :01H ==> 02H
669 00F7 0C035      BR    INITTIMER
670          ;
671          TIMADJ3:
672 00F8 1D0C0      MOV   SMF.#0000B      :NORMAL/SASA ==> 0.5H
673 00F9 0C035      BR    INITTIMER
674          ;
675          ;-----
676          :      SASA   KEY PROCESS
677          ;-----
678          SAPR:
679 00FA 14043      AND   INM.#0011B      :INIT INM REG
680          SKTI   SASA_F      :SASA MODE ?
+ 1 00FB 1E0C8 1    SKT   .MF.SASA_F SHR 4. #. DF. SASA_F AND 0FH
681 00FC 0C10A      BR    SAADJ0      :NORMAL/TIMER ==> 01H
682          ;
683          :SASA MODE ADJ
684 00FD 090C8      SKE   SMF.#1000B      :01H ==> 02H
685 00FE 0B0C9      SKNE  SMF.#1001B
686 00FF 0C104      BR    SALD2H
687          ;
688 0100 090CA      SKE   SMF.#1010B      :02H ==> 04H
689 0101 0B0CB      SKNE  SMF.#1011B
690 0102 0C107      BR    SALD4H
691          ;-----
692 0103 0C032      BR    NORMAL_INT      :04H ==> NORMAL MODE
693          ;
694          SALD2H:
695 0104 16048      OR    INM.#1000B
696 0105 1D0CB      MOV   SMF.#1011B      :01H ==> 02H
697 0106 0C034      BR    INITSASA
698          ;
699          SALD4H:
700 0107 1604C      OR    INM.#1100B
701 0108 1D0CF      MOV   SMF.#1111B      :02H ==> 04H
702 0109 0C034      BR    INITSASA
703          ;
704          SAADJ0:
705 010A 16044      OR    INM.#0100B
706 010B 1D0C9      MOV   SMF.#1001B      :NORMAL/TIMER ==> 01H
707 010C 0C034      BR    INITSASA
708          ;
709          ;-----
710          :      CUT    KEY PROCESS
711          ;-----
712          CUTPR:
713 010D 0C026      BR    STANDBY

```

CHAPTER 4 CONTROL PROGRAM

```

714 ;
715 ;-----
716 ; WIND KEY PROCESS
717 ;-----
718 WINPR:
719 010E 1E09C SKT WRSCM #1100B ;WR=11 ?
720 010F 0C113 BR WRADJ
721 0110 14093 AND WRSCM #0011B ;WR=11 ==> 01
722 0111 16094 OR WRSCM #0100B
723 0112 0C03B BR KEY_INT
724 WRADJ:
725 0113 10094 ADD WRSCM #0100B
726 0114 0C03B BR KEY_INT
727 ;
728 ;-----
729 ; RANDOM KEY PROCESS
730 ;-----
731 RANPR:
732 ; NOT1 RANDOM_F ;INVERT RANDOM_F ?
+ 1 0115 150D1 1 XOR .MF. RANDOM_F SHR 4. #. DF. RANDOM_F AND OFH
733 0116 0C03B BR RAN_IN
734 ;-----
735 ; SWING KEY PROCESS
736 ;-----
737 SWINGPR:
738 ; NOT1 SWING_F ;INVERT SWING FLAG
+ 1 0117 15088 1 XOR .MF. SWING_F SHR 4. #. DF. SWING_F AND OFH
739 0118 0C03B BR KEY_INT
740 ;-----
741 ; STANDBY KEY PROCESS
742 ;-----
743 STANDPR:
744 0119 0C02D BR NORMAL_MODE_IN ;1st "ON" KEY PROCESS
745 ;
746 ;*****
747 ; PROGRAM ADJ *
748 ;*****
749 ;
750 011A 074F0 ADJB: NOP ; [8] ;2ms_Cunter ADJ
751 011B 074F0 NOP ; [7]
752 011C 074F0 ADJB: NOP ; [6]
753 011D 074F0 NOP ; [5]
754 011E 074F0 NOP ; [4]
755 011F 074F0 NOP ; [3]
756 0120 074F0 NOP ; [2]
757 0121 070E0 RET ; [1]
758 ;;;
759 ;*****
760 ; CALL SUBROUTINE HERE *
761 ;*****
762 ;*****
763 ; ACTION SUBROUTINE [+17] *
764 ;*****
765 ACTION:
766 ;
767 ; SKT1 ONF ;ON ACTION ?
+ 1 0122 1E0D2 1 SKT .MF. ONF SHR 4. #. DF. ONF AND OFH
768 0123 0C146 BR STPOP ;OFF ACTION
769 ;
770 ; SKT1 RANDOM_F ;RANDOM ?
+ 1 0124 1E0D1 1 SKT .MF. RANDOM_F SHR 4. #. DF. RANDOM_F AND OFH
771 0125 1D0F9 MOV RAND_PAT.#09H ;NORMAL WIND MODE SET
772 ;-----
773 ; RANDOM WIND CHK
774 ;-----
775 ;RANDOM_WIND_MODE DECISION
776 0126 1E094 SKT WRSCM #0100B ;X1XX=01XX. 11XX SKIP

```

CHAPTER 4 CONTROL PROGRAM

```

777 0127 0C132          BR      MIDWIND          :10XX  MID_WIND MODE
778 0128 1F088          SKF      WRSCM.#1000B      :0XXX=01XX,00XX SKIP
779 0129 0C13C          BR      HIWIND           :11XX  HI_WIND_MODE
780                      ;----- RANDOM LOW WIND MODE -----
781                      ;LOWWIND:
782 012A 1D0A6          MOV      ACCU.#LOW          :INIT OUTPUT PORT=X110
783                      ;
784 012B 090F6          SKE      RAND_PAT.#08H      :RAND_PAT=8
785 012C 0B0F1          SKNE     RAND_PAT.#01H      :RAND_PAT=1
786 012D 1D0A7          MOV      ACCU.#STP          :SET OUTPUT PORT=X111
787                      ;
788                      ;
789 012E 090F6          SKE      RAND_PAT.#06H      :RAND_PAT=6
790 012F 0B0F3          SKNE     RAND_PAT.#03H      :RAND_PAT=3
791 0130 1D0A5          MOV      ACCU.#MID          :SET OUTPUT PORT=X101
792 0131 0C147          BR      OUTPUT             :RAND_PAT=2, 4, 5, 7, 9
793                      ;
794                      ;----- RANDOM MID WIND MODE -----
795                      ;MIDWIND:
796 0132 1D0A5          MOV      ACCU.#MID          :INIT OUTPUT PORT=X101
797                      ;
798 0133 090F7          SKE      RAND_PAT.#07H      :RAND_PAT=7
799 0134 0B0F3          SKNE     RAND_PAT.#03H      :RAND_PAT=3
800 0135 1D0A3          MOV      ACCU.#HI           :SET OUTPUT PORT=X011
801                      ;
802 0136 090F8          SKE      RAND_PAT.#08H      :RAND_PAT=8
803 0137 0B0F6          SKNE     RAND_PAT.#06H      :RAND_PAT=6
804 0138 1D0A6          MOV      ACCU.#LOW          :SET OUTPUT PORT=X110
805                      ;
806 0139 0B0F1          SKNE     RAND_PAT.#01H      :RAND_PAT=1
807 013A 1D0A6          MOV      ACCU.#LOW          :SET OUTPUT PORT=X110
808 013B 0C147          BR      OUTPUT             :RAND_PAT=2, 4, 5, 9
809                      ;----- RANDOM HI WIND MODE -----
810                      ;HIWIND:
811 013C 1D0A3          MOV      ACCU.#HI           :SET OUTPUT PORT=X011
812                      ;
813 013D 090F5          SKE      RAND_PAT.#05H      :RAND_PAT=5
814 013E 0B0F2          SKNE     RAND_PAT.#02H      :RAND_PAT=2
815 013F 1D0A6          MOV      ACCU.#LOW          :SET OUTPUT PORT=X110
816                      ;
817 0140 090F7          SKE      RAND_PAT.#07H      :RAND_PAT=7
818 0141 0B0F6          SKNE     RAND_PAT.#06H      :RAND_PAT=6
819 0142 1D0A5          MOV      ACCU.#MID          :SET OUTPUT PORT=X101
820                      ;
821 0143 0B0F1          SKNE     RAND_PAT.#01H      :RAND_PAT=1
822 0144 1D0A5          MOV      ACCU.#MID          :SET OUTPUT PORT=X101
823 0145 0C147          BR      OUTPUT             :RAND_PAT=3, 4, 8, 9
824                      ;-----
825                      ;STPOP:
826 0146 1D0A7          MOV      ACCU.#STP          :SET OUTPUT PORT=X111
827                      ;
828                      ;OUTPUT:
829                      ;
+ 1 0147 1F088 1          SKF1     SWING_F            :CHK SWING MODE ?
830 0148 160A8          SKF      .MF.SWING_F SHR 4.#.DF.SWING_F AND OFH
831 0149 1872A          OR      ACCU.#1000B          :SET SWING OFF
832                      ;
832                      ;
+ 1 014A 1F0D2 1          ST      POC.ACCU            :OUTPUT TO ACTION PORT (POC)
833 014B 070E0          SKF1     ONF                :OFF ACTION ?
834 014C 0C11A          RET
835                      ;
835                      ;*****
836                      ; LEDSCAN 1st *
837                      ;*****
838                      ;LED1:
839                      ;
840                      ;
+ 1 014D 1F0D1 1          SKF1     RANDOM_F            :CHK RANDOM MODE ?
+ 1 014D 1F0D1 1          SKF      .MF.RANDOM_F SHR 4.#.DF.RANDOM_F AND OFH

```

CHAPTER 4 CONTROL PROGRAM

```

841 014E 1401E      AND    LEDPORT, #1110B      :RANDOM LED ON
842                ;
843 014F 0B0AC      SKNE   ACCU, #1100B        :CHK HI_WIND MODE
844 0150 1401D      AND    LEDPORT, #1101B      :SCAN1 HI_WIND LED ON
845                ;
846                1          SKFI   SASA_F          :CHK SASA MODE
+ 1 0151 1F0C8 1    SKF    .MF. SASA_F SHR 4. #. DF. SASA_F AND 0FH
847 0152 0C157      BR     X2                    :BR SASA CHK
848                ;
849 0153 090C2      SKE    SMF, #0010B         :CHK TIMER 2H MODE ?
850 0154 0B0C3      SKNE   SMF, #0011B         :
851 0155 0C172      BR     TL                    :SCAN1 TIMER 2H LED ON
852 0156 0C174      BR     TIMOUT              :NO TIMER .RETURN
853                ;
854                X2:
855 0157 1E0CC      SKT    SMF, #1100B         :CHK SASA 4H MODE (11XX)
856 0158 0C17A      BR     SDJ1                 :NO SASA, RETURN
857 0159 0C170      BR     TH                    :SCAN1 SASA 4H LED ON
858                ;*****
859                ;    LEDSCAN 2nd          *
860                ;*****
861                LED2:
862 015A 0B0A8      SKNE   ACCU, #1000B         :CHK MID_WIND MODE
863 015B 1401D      AND    LEDPORT, #1101B      :SCAN1 MID_WIND LED ON
864                ;TIMLED2:
865                1          SKTI   SASA_F          :CHK SASA MODE
+ 1 015C 1E0C8 1    SKT    .MF. SASA_F SHR 4. #. DF. SASA_F AND 0FH
866 015D 0C162      BR     Y2                    :BR TIMER MODE
867                ;
868 015E 090CA      SKE    SMF, #1010B         :CHK SASA 2H MODE
869 015F 0B0CB      SKNE   SMF, #1011B         :
870 0160 0C170      BR     TH                    :SCAN2 SASA 2H LED ON
871 0161 0C176      BR     SDJ5                 :NO SASA, RETURN
872                ;
873                Y2:
874 0162 090C1      SKE    SMF, #0001B         :CHK TIMER 1H MODE
875 0163 0C176      BR     SDJ5                 :NO TIMER .RETURN
876 0164 0C172      BR     TL                    :SCAN2 TIMER 1H LED ON
877                ;*****
878                ;    LEDSCAN 3rd          *
879                ;*****
880                LED3:
881 0165 0B0A4      SKNE   ACCU, #0100B         :CHK LOW_WIND MODE
882 0166 1401D      AND    LEDPORT, #1101B      :SCAN1 LOW_WIND LED ON
883                ;TIMLED3:
884                1          SKTI   SASA_F          :CHK SASA MODE ?
+ 1 0167 1E0C8 1    SKT    .MF. SASA_F SHR 4. #. DF. SASA_F AND 0FH
885 0168 0C16D      BR     Z2                    :BR TIMER MODE
886                ;
887 0169 090C8      SKE    SMF, #1000B         :CHK SASA 1H MODE ?
888 016A 0B0C9      SKNE   SMF, #1001B         :
889 016B 0C170      BR     TH                    :SCAN2 SASA 1H LED ON
890 016C 0C176      BR     SDJ5                 :NO SASA . RETURN
891                ;
892                Z2:
893 016D 090C0      SKE    SMF, #0000B         :CHK TIMER 0.5H MODE
894 016E 0C176      BR     SDJ5                 :NO TIMER, RETURN
895 016F 0C172      BR     TL                    :SCAN3 TIMER 0.5H LED ON
896                ;-----
897                TH:
898 0170 14017      AND    LEDPORT, #0111B      :LED TIME OUTPUT HI BIT
899 0171 0C174      BR     TIMOUT              :
900                TL:
901 0172 1401B      AND    LEDPORT, #1011B      :LED TIME OUTPUT LOW BIT
902 0173 0C174      BR     TIMOUT              :
903                TIMOUT:
904 0174 18731      ST     POD, LEDPORT        :LED OUTPUT TO P0C

```

CHAPTER 4 CONTROL PROGRAM

```

905 0175 070E0          RET
906                    :-----
907                    :   SDJ
908                    :-----
909 0176 074F0      SDJ5:  NOP
910 0177 074F0          NOP
911 0178 074F0          NOP
912 0179 074F0          NOP
913 017A 0C174      SDJ1:  BR    TIMEOUT
914                    :;;
915                    :-----
916                    :   KEYSKAN          [+9]  -
917                    :-----
918                    :KEYSCAN:
919 017B 0871A          LD    ACCU.P0B          :LOAD KEY PORT (P0B)
920 017C 140AE          AND   ACCU.#1110B          :MASK KEY=XXX0 (SKIP ZERO CROSSING)
921                    :
922 017D 0B0A0          SKNE  ACCU.#0000B          :KEY=000X ?
923 017E 0C188          BR    NON_KEY
924 017F 0B0A2          SKNE  ACCU.#0010B          :KEY=001X ?
925 0180 0C18A          BR    WIND_TIMER_KEY
926 0181 0B0A4          SKNE  ACCU.#0100B          :KEY=010X ?
927 0182 0C18C          BR    RANDOM_SASA_KEY
928 0183 0B0A8          SKNE  ACCU.#1000B          :KEY=100X ?
929 0184 0C18E          BR    SWING_CUT_KEY
930                    :-----
931                    :   MULKY
932                    :-----
933                    :MULKY:
934                    |   SETI  LOCKF          :MULTI PRESS ERROR KEY(LOCK)
+ 1 0185 16078 |   OR    .MF.LOCKF SHR 4.#.DF.LOCKF AND OFH
935 0186 16084          OR    KS_Temp.#0100B
936 0187 070E0          RET
937                    :-----
938                    :   NON_KEY
939                    :-----
940                    :NON_KEY:
941 0188 074F0          NOP          :RELEASE KEY
942 0189 070E0          RET
943                    :-----
944                    :   WIND_TIMER_KEY
945                    :-----
946                    :WIND_TIMER_KEY:
947 018A 16081          OR    KS_Temp.#0001B          :KeyScan_Temporary=X001 WIND/TIMER KEY
948 018B 0C18D          BR    SKDJ2
949                    :
950                    :-----
951                    :   RANDOM_SASA_KEY
952                    :-----
953                    :RANDOM_SASA_KEY:
954 018C 16082          OR    KS_Temp.#0010B          :KeyScan_Temporary=X010 RANDOM/SASA KEY
955 018D 0C18F      SKDJ2: BR    SKDJ1
956                    :
957                    :-----
958                    :   SWING_CUT_KEY
959                    :-----
960                    :SWING_CUT_KEY:
961 018E 16083          OR    KS_Temp.#0011B          :KeyScan_Temporary=X011 SWING/CUT KEY
962 018F 071E0      SKDJ1: RETSK          :[9]
963                    :
964                    :-----
965                    :   PULSE COUNT          [+4]  -
966                    :-----
967                    :PULSE_COUNT:
968                    |   SKTI  P0B0          :CHK PULSE=HIGH ?
+ 1 0190 1E711 |   SKT   .MF.P0B0 SHR 4.#.DF.P0B0 AND OFH
969 0191 0C196          BR    PFCLR          :   PULSE=LOW

```

CHAPTER 4 CONTROL PROGRAM

```

970      ;-----
971      :PULSE_HIGH -
972      ;-----
973      SKF1 PULSE_F          ;PULSE_F EVER SET ?
+ 1 0192 1F0D8 1      SKF .MF.PULSE_F SHR 4.#.DF.PULSE_F AND OFH
974 0193 0C198      BR PRET          ;PULSE_F SET ALREADY
975      SET1 PULSE_F          ;1st High Pulse SET PULSE_F=1
+ 1 0194 160D8 1      OR .MF.PULSE_F SHR 4.#.DF.PULSE_F AND OFH
976 0195 071E0      RETSK
977      ;-----
978      :PULSE_LOW -
979      ;-----
980      PFCLR:                ;PULSE_LOW STATE
981 0196 074F0      NOP
982      CLR1 PULSE_F          ;CLR PULSE_F=0
+ 1 0197 140D7 1      AND .MF.PULSE_F SHR 4.#.DF.(NOT PULSE_F AND OFH)
983 0198 074F0      PRET: NOP
984 0199 070E0      RET
985      ;-----
986      ; BUZZ WAIT 2MS [ +5 ] -
987      ;-----
988      BUZZ_WAIT_2MS:
989      ;
990 019A 1D010      MOV MSL.#00H :[2] :INITIAL 124us_COUNTER MAX=[7]
991 019B 1D000      MOV MSH.#00H :[3] :INITIAL 2ms_COUNTER MAX=[14(0EH)]
992      ;
993      LOP0: SKF1 BUZF :[1]
+ 1 019C 1F042 1      SKF .MF.BUZF SHR 4.#.DF.BUZF AND OFH
994      NOT1 POA3 :[2] :INVERT BUZZER "H/L"
+ 1 019D 15708 1      XOR .MF.POA3 SHR 4.#.DF.POA3 AND OFH
995 019E 10001      ADD MSH.#01H :[3] :2ms Counter + 1
996      ; : DELAY 112us
997 019F 074F0      LOP1: NOP :[1] [4]
998 01A0 10011      ADD MSL.#01H :[2] V
999 01A1 0901E      SKE MSL.#0EH :[3] V 2us * 4 * 14(0EH) = 112 us
1000 01A2 0C19F      BR LOP1 :[4] [59]
1001      ;-----
1002      ; 124 us Ending
1003      ;-----
1004 01A3 1D010      MOV MSL.#00H :[1] [60] CLR 124us Counter (MSL)
1005 01A4 0900E      SKE MSH.#0EH :[2] [61] 2ms Ending ?
1006 01A5 0C19C      BR LOP0 :[3] [62] 2us * 62 = 124 us
1007      ;
1008      ;-----
1009      ; 2ms Ending
1010      ;-----
1011      SKT1 BUZF : BUZZER MODE ?
+ 1 01A6 1E042 1      SKT .MF.BUZF SHR 4.#.DF.BUZF AND OFH
1012 01A7 070E0      RET : BUZZER OFF MODE
1013      ;
1014 01A8 100B1      ADD BZ50MS_L.#01H :50ms_COUNTER +1
1015 01A9 12070      ADDC BZ50MS_H.#00H
1016      ; : (19H) = (25) 2ms*25=50ms
1017 01AA 1F071      SKF BZ50MS_H.#01H :CHK >=50ms ?
1018 01AB 090B9      SKE BZ50MS_L.#09H ;
1019 01AC 070E0      RET : < 50ms REPEAT SCAN
1020      ;
1021      ;-----
1022      ; 50ms Ending
1023      ;-----
1024      ;
1025      ; = 50ms GOTO KEYPRESS ROUTINE
1026 01AD 1407E      AND BZ50MS_H.#1110B :CLR KeyScan_Counter=XXX0
1027      CLR1 BUZF :RESET TO BUZZ_OFF_MODE
+ 1 01AE 1404D 1      AND .MF.BUZF SHR 4.#.DF.(NOT BUZF AND OFH)
1028 01AF 140B0      AND BZ50MS_L.#00H :CLR BUZZ_50ms_COUNTER
1029 01B0 071E0      RETSK :BRANCH TO KEYPROCESS

```

CHAPTER 4 CONTROL PROGRAM

```

1030      ;-----
1031      :      WAIT 2MS          [+5]  -
1032      ;-----
1033      WAIT_2MS:                :2000us/10us=200 (C8H)
1034 01B1 0C1B4      BR      AZ0          :[2]  :INITIAL JUMP
1035      ;
1036 01B2 10011      AZ1:  ADD      MSL.#01H      :[1]  :MS+1
1037 01B3 12000      ADDC     MSH.#00H      :[2]
1038 01B4 0B00C      AZ0:  SKNE     MSH.#0CH      :[3]  :CHK MS=C8H ?
1039 01B5 09018      SKE      MSL.#08H      :[4]
1040 01B6 0C1B2      BR      AZ1          :[5]
1041 01B7 070E0      RET                :MS=C8H (2ms)
1042      ;-----
1043      :      2ms  ADJ          -
1044      ;-----
1045      MSADJ:                    :2ms Counter ADJ
1046 01B8 11011      SUB      MSL.#01H      :[1]  :ADJ VALUE - 1
1047 01B9 09010      SKE      MSL.#00H      :[2]  :ADJ VALUE = 0 ?
1048 01BA 0C1B8      BR      MSADJ         :[3]  :ADJ VALUE <0 , REPEAT SUB
1049 01BB 070E0      RET                :ADJ VALUE = 0 , RETURN
1050      ;-----
1051      :      2ms  REG (MS) RESET [+3] -
1052      ;-----
1053      MS_H_L_CLR:
1054 01BC 1D000      MOV      MSH.#00H
1055 01BD 1D010      MOV      MSL.#00H
1056 01BE 070E0      RET
1057      ;-----
1058      :      4ms_Counter REG (FS) RESET [+3] -
1059      ;-----
1060      FS_H_L_CLR:
1061 01BF 1D020      MOV      FSH.#00H
1062 01C0 1D030      MOV      FSL.#00H
1063 01C1 070E0      RET
1064      ;-----
1065      :      HALF_HOUR REG (HS) RESET [+4] -
1066      ;-----
1067      HS_H_M_L_CLR:
1068 01C2 1404E      AND      HSH.#1110B
1069 01C3 1D050      MOV      HSM.#00H
1070 01C4 1D060      MOV      HSL.#00H
1071 01C5 070E0      RET
1072      ;
1073      OPTION
1074      OPTPOB OPEN. OPEN. OPEN. OPEN
+ 14      0000      |
1075      OPTRES RESPLUP
+ 5       0001      |
1076      ENDOP
1077      END

```

TOTAL ERRORS = 0
TOTAL WARNINGS = 0

END OF LIST

AS17K V1.02 02 << D17104 XREF LIST >> 12:49:58 12/17/89 PAGE 01-001

PROG = FAN27 TEST

SOURCE = FAN27.ASM

SYMBOL TYPE A VALUE /REF (#DEF)

ACCU	MEM L	0.0A /# 104	284	285	362	363	364	537	538	544	547	
			559	567	568	572	575	578	579	582	583	624

CHAPTER 4 CONTROL PROGRAM

			625	629	632	635	638	641	644	782	786	791
			796	800	804	807	811	815	819	822	826	830
			831	843	862	881	919	920	922	924	926	928
ACTION	LAB L	122 /	273	611	# 765							
ADJ6	LAB L	11C /	514	# 752								
ADJ8	LAB L	11A /#	750	834								
AZO	LAB L	1B4 /	1034	#1038								
AZ1	LAB L	1B2 /#	1036	1040								
BUZF	FLG L	0.04.1 /#	112	237	237-1	237-1	299	299-1	299-1	307	307-1	307-1
			331	331-1	331-1	339	339-1	339-1	387	387-1	387-1	406
			406-1	406-1	420	420-1	420-1	993	993-1	993-1	1011	1011-1
			1011-1	1027	1027-1	1027-1						
BUZZ_WAIT_2MS	LAB L	19A /	616	# 988								
BZ50MS_H	MEM L	0.07 /#	101	1015	1017	1026						
BZ50MS_L	MEM L	0.0B /#	105	239	1014	1018	1028					
COMMON	LAB L	37 /	252	# 261								
COUNT	LAB L	85 /	320	# 418								
COUNT1_1	LAB L	52 /	308	# 322	340	414						
COUNT2_1	LAB L	7A /	379	# 392								
COUNT2_2	LAB L	7C /	384	# 396								
COUNT_ADJ	LAB L	50 /#	318	324	359	373	390	394	398			
COUNT_WAIT	LAB L	DB /	508	# 615								
COUNT_WAITADJ1	LAB L	AA /	427	# 498								
COUNT_WAITADJ2	LAB L	AC /	435	# 501								
COUNT_WAITADJ3	LAB L	AE /	447	453	# 504							
CUTPR	LAB L	10D /	633	# 712								
END_RANDOM	LAB L	DA /	596	# 606								
ERRKY	LAB L	6E /	365	# 370								
FS_H_L_CLR	LAB L	1BF /	183	202	245	258	459	#1060				
FS_INIT	LAB L	99 /	448	# 458								
FSH	MEM L	0.02 /#	94	438	445	451	1061					
FSL	MEM L	0.03 /#	95	188	189	210	215	216	437	446	452	1062
H2	LAB L	A4 /	478	# 480								
H1	DAT L	3 /#	155	800	811							
HIWIND	LAB L	13C /	779	# 810								
HS_H_M_L_CLR	LAB L	1C2 /	246	259	485	#1067						
HSH	MEM L	0.04 /#	97	112	471	476	1068					
HSL	MEM L	0.06 /#	99	469	479	1070						
HSM	MEM L	0.05 /#	98	470	477	1069						
HZ50	LAB L	25 /	220	# 223								
HZ50_4secADJH	DAT L	C /#	149	451								
HZ50_4secADJL	DAT L	8 /#	150	452								
HZ50CHK	LAB L	95 /	443	# 450								
HZ60_4secADJH	DAT L	F /#	151	445								
HZ60_4secADJL	DAT L	0 /#	152	446								
HZ_INIT	LAB L	17 /#	203	205								
HZ_WAIT	LAB L	1C /#	212	217								
HZF	FLG L	0.0D.2 /#	118	221	221-1	221-1	224	224-1	224-1	442	442-1	442-1
HZIN	LAB L	19 /#	207	209								
INITSASA	LAB L	34 /#	254	697	702	707						
INITTIMER	LAB L	35 /#	257	665	669	673						
INM	MEM L	0.04 /#	96	542	545	679	695	700	705			
KEY_EFFECT_CHK	LAB L	71 /	351	369	# 377							
KEY_INT	LAB L	3B /#	269	723	726	739						
KEY_RELEASE	LAB L	62 /	348	# 353								
KEYPROCESS	LAB L	DE /	618	# 623								
KEYSCAN	LAB L	17B /	312	346	# 918							
KS_Count	MEM L	0.07 /#	100	113	238	356	381	383				
KS_Temp	MEM L	0.08 /#	102	111	243	310	343	350	354	624	935	947
			954	961								
KYRL	LAB L	65 /	355	# 358								
LED1	LAB L	14D /	302	# 838								
LED2	LAB L	15A /	334	# 861								
LED3	LAB L	166 /	409	# 880								
LEDPOR	MEM L	0.01 /#	91	281	282	841	844	863	882	898	901	904
LEDSCAN2	LAB L	54 /	289	# 329								
LEDSCAN3	LAB L	7E /	291	# 404								

CHAPTER 4 CONTROL PROGRAM

LOCKF	FLG	L	0.07.3	/#	113	.	371	.	371-1	.	371-1	.	378	.	378-1	.	378-1	.	386	.	386-1	.	386-1
							934	.	934-1	.	934-1												
LOPO	LAB	L	19C	/#	993	.	1006																
LOP1	LAB	L	19F	/#	997	.	1000																
LOW	DAT	L	6	/#	157	.	782	.	804	.	807	.	815										
MID	DAT	L	5	/#	156	.	791	.	796	.	819	.	822										
MIDWIND	LAB	L	132	/	777	.	#	795															
Minute_30ADJ	DAT	L	2	/#	153	.	479																
MS_H_L_CLR	LAB	L	18C	/	182	.	187	.	201	.	214	.	#1053										
MSADJ	LAB	L	188	/	319	.	368	.	507	.	#1045	.	1048										
MSH	MEM	L	0.00	/#	92	.	991	.	995	.	1005	.	1037	.	1038	.	1054						
MSL	MEM	L	0.01	/#	93	.	317	.	323	.	358	.	367	.	372	.	389	.	393	.	397	.	499
							502	.	505	.	990	.	998	.	999	.	1004	.	1036	.	1039	.	1046
							1055																
Multi_Key_CHK	LAB	L	67	/	344	.	#	361															
NON_KEY	LAB	L	188	/	923	.	#	940															
NON_SASA	LAB	L	D3	/	523	.	#	591															
NORMAL_INT	LAB	L	32	/#	250	.	661	.	692														
NORMAL_MODE_IN	LAB	L	2D	/#	241	.	744																
OFFACT	LAB	L	C5	/	552	.	#	566															
ONF	FLG	L	0.0D.1	/#	119	.	262	.	262-1	.	262-1	.	266	.	266-1	.	266-1	.	533	.	533-1	.	533-1
							551	.	551-1	.	551-1	.	592	.	592-1	.	592-1	.	595	.	595-1	.	595-1
							767-1	.	767-1	.	832	.	832-1	.	832-1								
ONFC	MEM	L	0.0E	/#	108	.	255	.	525	.	527	.	558	.	560	.	570	.	574	.	577	.	581
							585																
OPTPOB	MAC	L	*****	/	1074																		
OPTRES	MAC	L	*****	/	1075																		
OUTPUT	LAB	L	147	/	792	.	808	.	823	.	#	828											
POA	MEM	L	0.70	/#	86	.	194	.	230	.	298	.	300	.	330	.	332	.	405	.	407		
POB	MEM	L	0.71	/#	87	.	362	.	919														
POC	MEM	L	0.72	/#	88	.	195	.	231	.	831												
POD	MEM	L	0.73	/#	89	.	196	.	232	.	282	.	904										
PFCLR	LAB	L	196	/	969	.	#	980															
POWER_DELAY	LAB	L	D	/#	185	.	190																
PRET	LAB	L	198	/	974	.	#	983															
PULSE	LAB	L	8B	/	426	.	#	431															
PULSE_COUNT	LAB	L	190	/	432	.	#	967															
PULSE_F	FLG	L	0.0D.3	/#	117	.	242	.	242-1	.	242-1	.	973	.	973-1	.	973-1	.	975	.	975-1	.	975-1
							982	.	982-1	.	982-1												
RAN_IN	LAB	L	38	/#	264	.	733																
RAND_PAT	MEM	L	0.0F	/#	109	.	265	.	267	.	601	.	603	.	604	.	771	.	784	.	785	.	789
							790	.	798	.	799	.	802	.	803	.	806	.	813	.	814	.	817
							821																
RANDOM	LAB	L	D4	/	528	.	561	.	586	.	#	593											
RANDOM_F	FLG	L	0.0D.0	/#	120	.	242	.	242-1	.	425	.	425-1	.	425-1	.	594	.	594-1	.	594-1	.	732
							732-1	.	732-1	.	770	.	770-1	.	770-1	.	840	.	840-1	.	840-1		
RANDOM_SASA_KEY	LAB	L	18C	/	927	.	#	953															
RANPR	LAB	L	115	/	639	.	#	731															
SAADJO	LAB	L	10A	/	681	.	#	704															
SALD2H	LAB	L	104	/	686	.	#	694															
SALD4H	LAB	L	107	/	690	.	#	699															
SAPR	LAB	L	FA	/	630	.	#	678															
SASA_F	FLG	L	0.0C.3	/#	115	.	522	.	522-1	.	522-1	.	651	.	651-1	.	651-1	.	680	.	680-1	.	680-1
							846	.	846-1	.	846-1	.	865	.	865-1	.	865-1	.	834	.	884-1	.	884-1
SASACHK	LAB	L	B3	/	492	.	516	.	#	521													
SASACHKADJ1	LAB	L	B1	/	466	.	#	513															
SASACHKADJ2	LAB	L	B2	/	480	.	#	515															
SCAN	LAB	L	3C	/#	280	.	617																
SDJ1	LAB	L	17A	/	856	.	#	913															
SDJ5	LAB	L	176	/	871	.	875	.	890	.	894	.	#	909									
SKDJ1	LAB	L	18F	/	955	.	#	962															
SKDJ2	LAB	L	18D	/	948	.	#	955															
SMF	MEM	L	0.0C	/#	106	.	115	.	251	.	424	.	465	.	490	.	491	.	537	.	650	.	654
							657	.	664	.	668	.	672	.	684	.	685	.	638	.	689	.	696
							706	.	849	.	850	.	855	.	868	.	869	.	874	.	887	.	888
STA	MEM	L	0.0D	/#	107	.	117	.	118	.	119	.	120										
STANDBY	LAB	L	26	/	222	.	#	229	.	493	.	713											

CHAPTER 4 CONTROL PROGRAM

```

STANDPR      LAB L 119 / 645 .# 743
STP          DAT L 7 /# 158 . 786 . 826
STPOP       LAB L 146 / 768 .# 825
SWING_CUT_KEY LAB L 18E / 929 .# 960
SWING_F     FLG L 0.08.3 /# 111 . 738 . 738-1 . 738-1 . 829 . 829-1 . 829-1
SWINGPR     LAB L 117 / 642 .# 737
TH          LAB L 170 / 857 . 870 . 889 .# 897
TIMADJ1     LAB L F4 / 655 .# 663
TIMADJ2     LAB L F6 / 658 .# 667
TIMADJ3     LAB L F8 / 652 .# 671
TIMOUT      LAB L 174 / 852 . 899 . 902 .# 903 . 913
TL          LAB L 172 / 851 . 876 . 895 .# 900
WAIT_2MS    LAB L 1B1 / 186 . 213 .#1033
WAIT_ADJ    LAB L AF / 500 . 503 .# 506
WIND_TIMER_KEY LAB L 18A / 925 .# 946
WINPR       LAB L 10E / 636 .# 718
WRADJ       LAB L 113 / 720 .# 724
WRSCM       MEM L 0.09 /# 103 . 244 . 284 . 288 . 290 . 304 . 305 . 336 . 337 . 411
           412 . 719 . 721 . 722 . 725 . 776 . 778
X2          LAB L 157 / 847 .# 854
Y2          LAB L 162 / 866 .# 873
Z2          LAB L 16D / 885 .# 892

```

TOTAL SYMBOLS = 136

END OF XREF LIST

