To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

RENESAS

To all our customers

Regarding the change of names mentioned in the document, such as Hitachi Electric and Hitachi XX, to Renesas Technology Corp.

The semiconductor operations of Mitsubishi Electric and Hitachi were transferred to Renesas Technology Corporation on April 1st 2003. These operations include microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.) Accordingly, although Hitachi, Hitachi, Ltd., Hitachi Semiconductors, and other Hitachi brand names are mentioned in the document, these names have in fact all been changed to Renesas Technology Corp. Thank you for your understanding. Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.

Renesas Technology Home Page: http://www.renesas.com

Renesas Technology Corp.
Customer Support Dept.
April 1, 2003

Renesas Technology Corp.

Cautions

Application Note

RENESAS

HMCS400 Series

Application Note

Renesas Electronics
www.renesas.com

Rev.1.0 1999.03

Cautions

Preface

The HMCS43XX family is a family of 4-bit microcomputers built around the HMCS400 CPU and including standard peripheral functions such as A/D converters with a variety of A/D input channels, serial interfaces, and multifunction timers. The architecture of the powerful HMCS400 CPU core is known for its ease of programming.

The peripheral functions of the HMCS43XX have been developed as standalone modules and a modular architecture employed in which the respective modules are connected via a standardized interface.

The HD404889 Series are also 4-bit microcomputers also built around the powerful HMCS400 CPU core with its excellent ease of programming, and including various peripheral functions such as LCD circuit, A/D converters, and multifunction timers. The microcomputers in this series are ideally suited to display panel control and system control in a wide range of applications, primarily audio-visual equipment such as radiocassette sets with built-in CD players, as well as home appliances such as electronic jars, and telephones and pagers.

The peripheral functions of the HM404889 Series have been developed as standalone modules and a modular architecture employed in which the respective modules are connected via a standardized interface.

The “Applications” volume of the HMCS400 Series Application Notes is a collection of examples of combinations of the built-in peripherals in the HMCS400 Series of microcomputers. This collection is intended as a reference for software and hardware designers.

The operation of the programs and circuits, etc., described in these application notes has been checked. However, please be sure to confirm their operation before actually using them in any application.

Contents

Section 1 HMCS400 Series Application Notes—How to Use the Applications Volume

The Application Notes are, as shown in figure 1, divided into two sections.

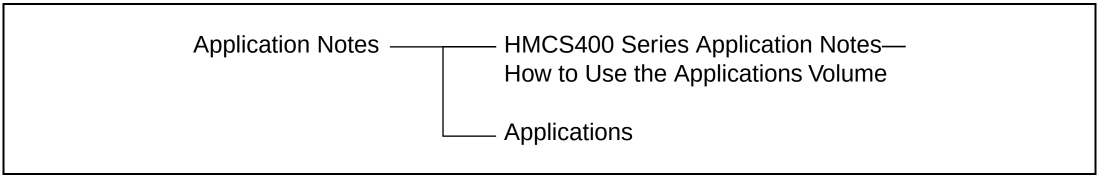


Figure 1 Structure of Application Notes

HMCS400 Series Application Notes—How to Use the Applications Volume

This section describes how to use the HMCS400 Series Application Notes—Applications Volume.

Applications

This section uses simple example tasks to describe how various combinations of the built-in peripherals (timers, serial interface, A/D converters, I/O ports, interrupts, low-power modes, etc.) of the MHCS400 Series of microcomputers are used.

1.1 Structure of Applications Section

As shown in figure 2, the Applications section describes how to use the built-in functions of the HMCS400 Series.

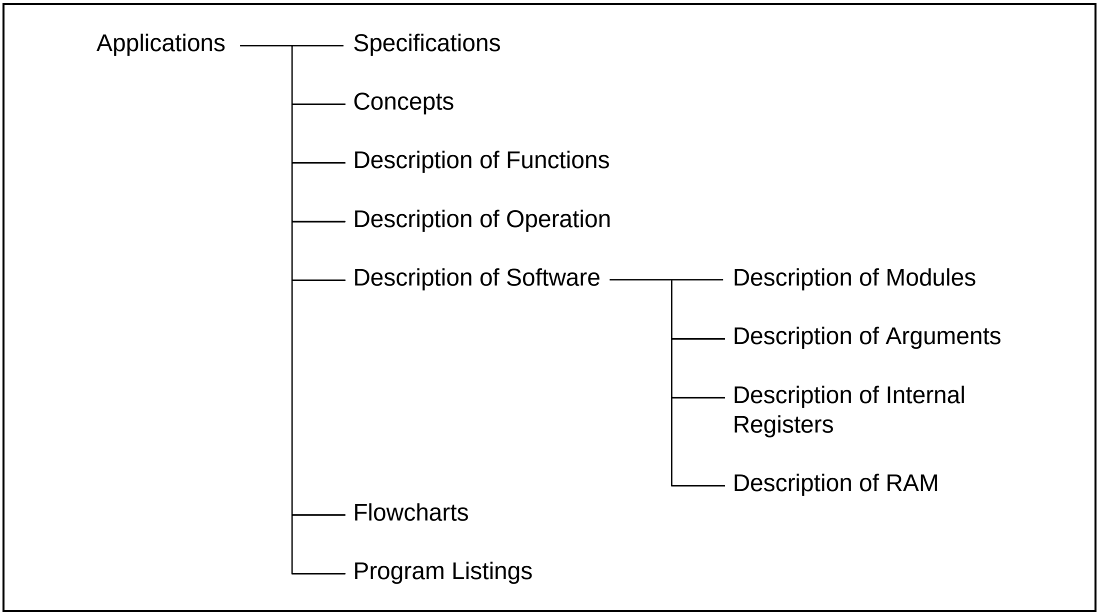


Figure 2 Structure of Applications Section

Specifications

This section describes the system specifications for the example tasks.

Concepts

This section describes the methods employed to realize the systems in the example tasks.

Description of Functions

This section describes the features and distribution of the peripheral functions employed in the example tasks.

Description of Operation

This section uses timing charts to describe the operation of the example tasks.

Description of Software

1. Description of Modules

   This section describes the software module operating in the example tasks.

2. Description of Arguments

   This section describes the input arguments required for execution of the respective modules, and the arguments output on completion of module execution.

3. Description of Internal Registers

   This section describes the internal registers (timer control register and serial mode register, etc.) set by the module.

4. Description of RAM

   This section describes the RAM labels used by the modules and their functions.

Flowcharts

This section provides flowcharts of the software run in the example tasks.

Program Listings

This section provides listings of the software run in the example tasks.

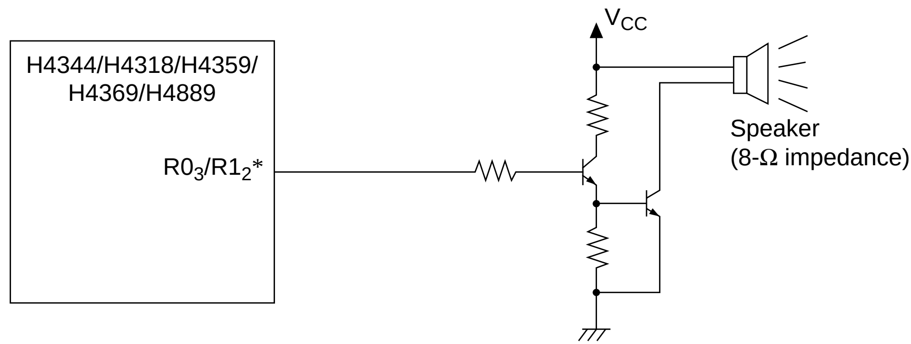
Section 2 Applications

2.1 Musical Performance

Musical Performance	MCU: H4344/H4318/H4359/ H4369/H4889	Functions Used: R0/R1 Port, Timer B, and Timer C

Specifications

1. As shown in figure 1, the Bach minuet is played repeatedly in the H4344/H4318/H4359/ H4369 Series by connecting a speaker with an 8-Ω resistance to the $R0_3$ port. In the H4889, this is achieved by connecting to the $R1_2$ port.



Note: * $R0_3$: Employed in the H4344/H4318/H4359/H4369 Series
$R1_2$: Employed in the H4889 Series

Figure 1 Speaker Connection in Musical Performance

Concepts

1. The “minuet” is played by outputting a pulse with a frequency corresponding to the notes from the $R0_3/R1_2$ port.

2. The frequencies corresponding to the notes are set using the timer B reload timer function. The data table is referenced using the pattern command and the referenced data set in the timer-counter B reload value to achieve the frequency corresponding to the desired note. Moreover, the High/Low output from the $R0_3/R1_2$ port is controlled during timer B interrupt processing.

3. The duration of each note is set using the timer C reload timer function. The data table is referenced using the pattern command and the referenced data set in the timer counter C reload value to set the output duration of each note.

4. Figure 2 shows the settings for the frequencies for the respective notes and the duration of each note.

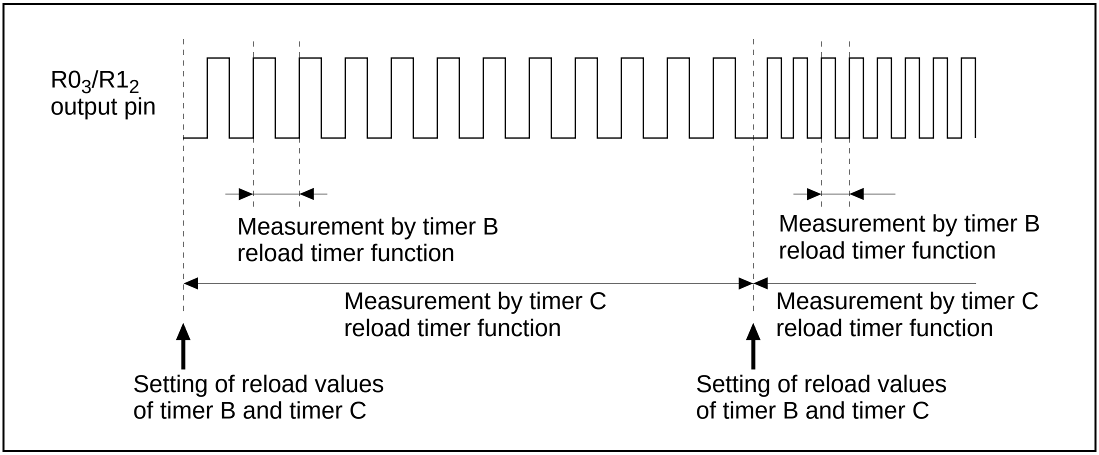


Figure 2 Method of Setting Frequencies for Notes and Output Durations

Description of Functions

1. In this example task, the R0/R1 port, timer B, and timer C functions are used to play a Bach minuet. Figure 3 is a functional block diagram of this task.

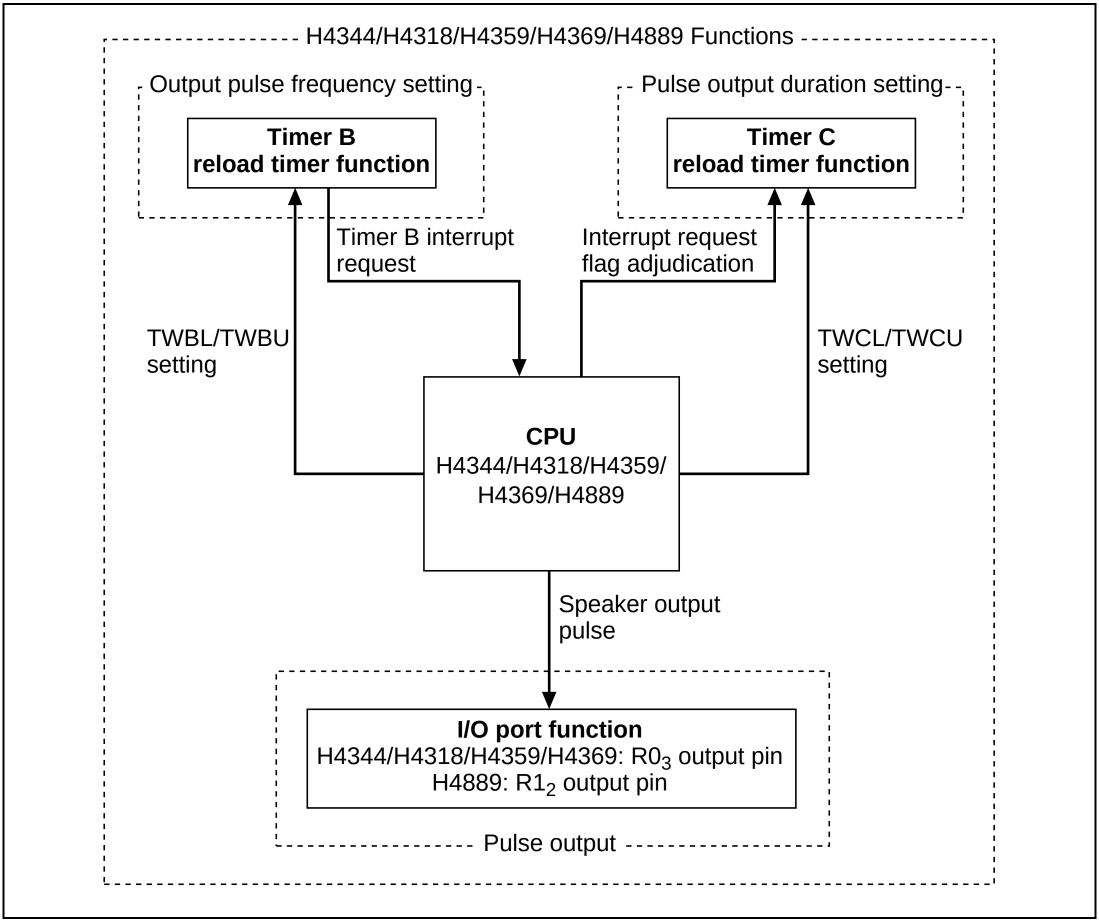


Figure 3 Functional Block Diagram

- Timer B reload timer function

  This function sets the output pulse frequency. The frequency of the output pulse is set by the timer counter B reload value. The reload value to be set is referenced from the data table.

- Timer C reload timer function

  This function sets the duration for which a frequency pulse is output. The duration of pulse output is set by the timer counter C reload value. The reload value to be set is referenced from the data table.

- I/O port function

  This is the output for the pulse to be output to the speaker.

2. The timer B, timer C, and I/O port functions are described below.

   a. Figure 4 is a block diagram of the timer B function.

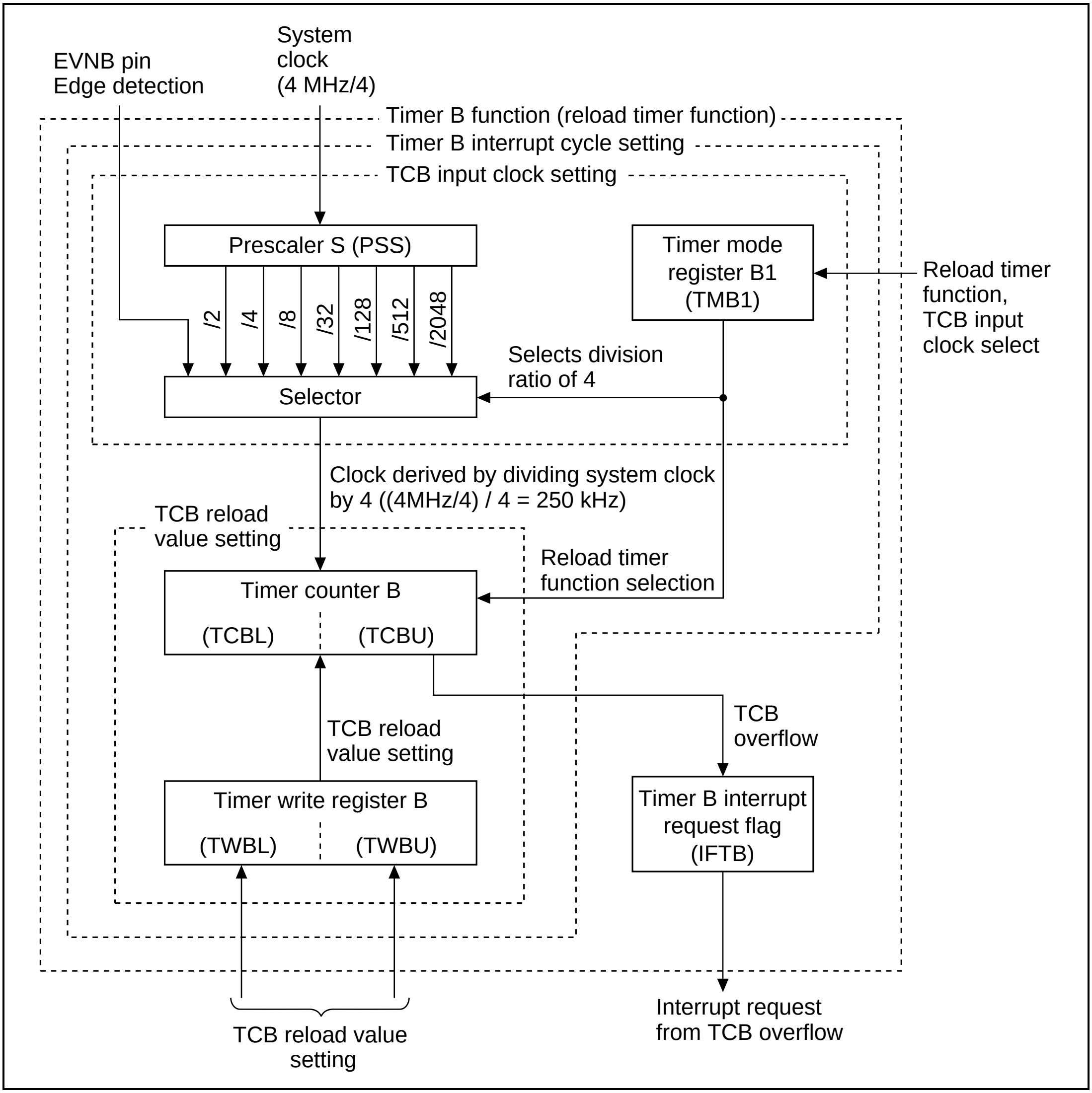


Figure 4   Timer B Function Block Diagram

b. Timer B is an 8-bit multifunction timer (free running/event counter/reload timer/input capture [*1]). In this example task, timer B is used as a reload timer. Table 1 describes the timer B functions.

Table 1 Timer B Functions

Timer Mode Register B1 (TMB1)	
Function	TMB1 is a 4-bit write-only register. It selects the timer B function (free-running/reload timer) and operating clock. TMB1 is initialized to $0 when reset and in stop mode.
Timer Write Register BL, U (TWBL, TWBU)	
Function	TWBL and TWBU form an 8-bit write-only register, which is made up of the lower digit (TWBL) and upper digit (TWBU). TWBL and TWBU are used for the initial TCB setting (the reload setting when operation as a reload timer).
Timer Counter B (TCB)	
Function	TCB is an 8-bit up-counter, which is incremented by the input internal clock. The TCB input clock is selected using bits TMB12 to TMB10 of TMB1. The value written to TWBL and TWBU is also written to TCB. When TCB overflows, the timer B interrupt request flag (IFTB) is set to “1”. If, at this point, timer B is set as a reload timer, the value of TWBL and TWBU is written to this counter and the count starts from this value. TCB is initialized to $00 when reset and in stop mode.
Prescaler S (PSS)	
Function	PSS is an 11-bit counter to which the system clock is input when in active mode and standby mode, and the subsystem clock is input when in subactive mode[*2] PSS is initialized to $000 at a reset, and division of the system clock starts when the reset is canceled. PSS operation is halted when reset, in stop mode, and in watch mode[*2]. However, it runs in other operating modes. The PSS output is shared by the internal peripheral modules, the division ratio being set independently for each of the internal peripheral modules.
Timer B Interrupt Request Flag (IFTB)	
Function	IFTB reflects the existence of the timer B interrupt request. When timer B overflows, IFTB is set to “1”. IFTB can only be read/written to (only “0” can be written) using bit operation commands. Note that IFTB is not automatically cleared even when the interrupt is received, and must be cleared by writing “0” using software. IFTB is cleared at a reset and in stop mode.
Timer B Interrupt Mask (IMTB)	
Function	IMTB is the bit that masks IFTB. When IFTB is set to “1” and, additionally, IMTB is “0”, a timer B interrupt request is sent to the CPU (when IE = “1”). If IFTB is set to “1” but IMTB is “1”, no interrupt request is sent to the CPU and the timer B interrupt is held. IMTB can only be read or written to using bit operation commands. It is set to “1” at a reset and in stop mode.

Notes: 1. Applies to H4318/H4359/H4369 Series only. In the H4344/H4889 Series, timer B has no input capture function.

2. Applies only to H4369/H4889 Series.

c. Figure 5 is a block diagram of the timer C function in the H4344/H4318/H4359/H4369 Series.

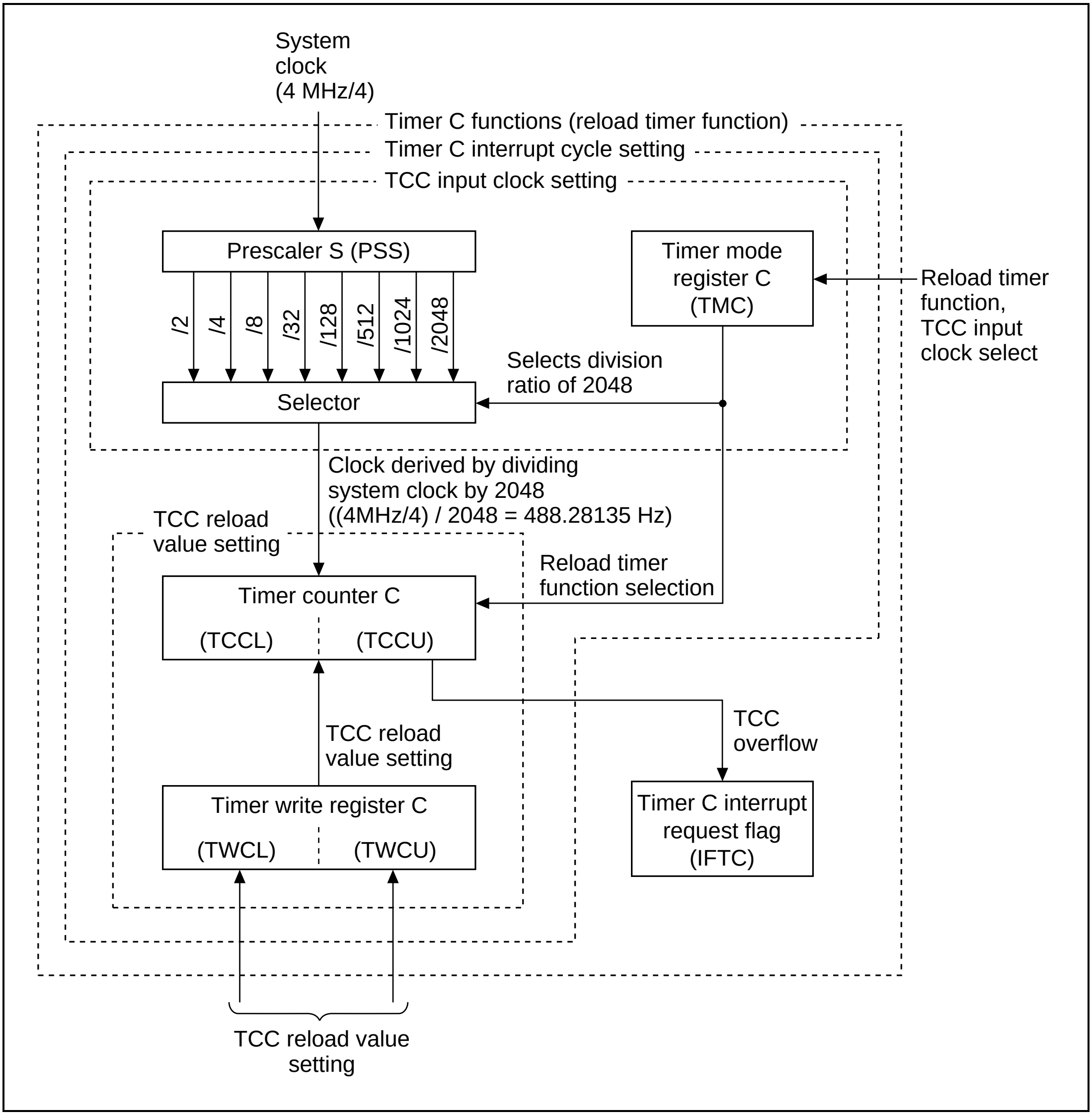


Figure 5 H4344/H4318/H4359/H4369 Series Timer C Function Block Diagram

d. Figure 6 is a block diagram of the timer C function in the H4889 Series.

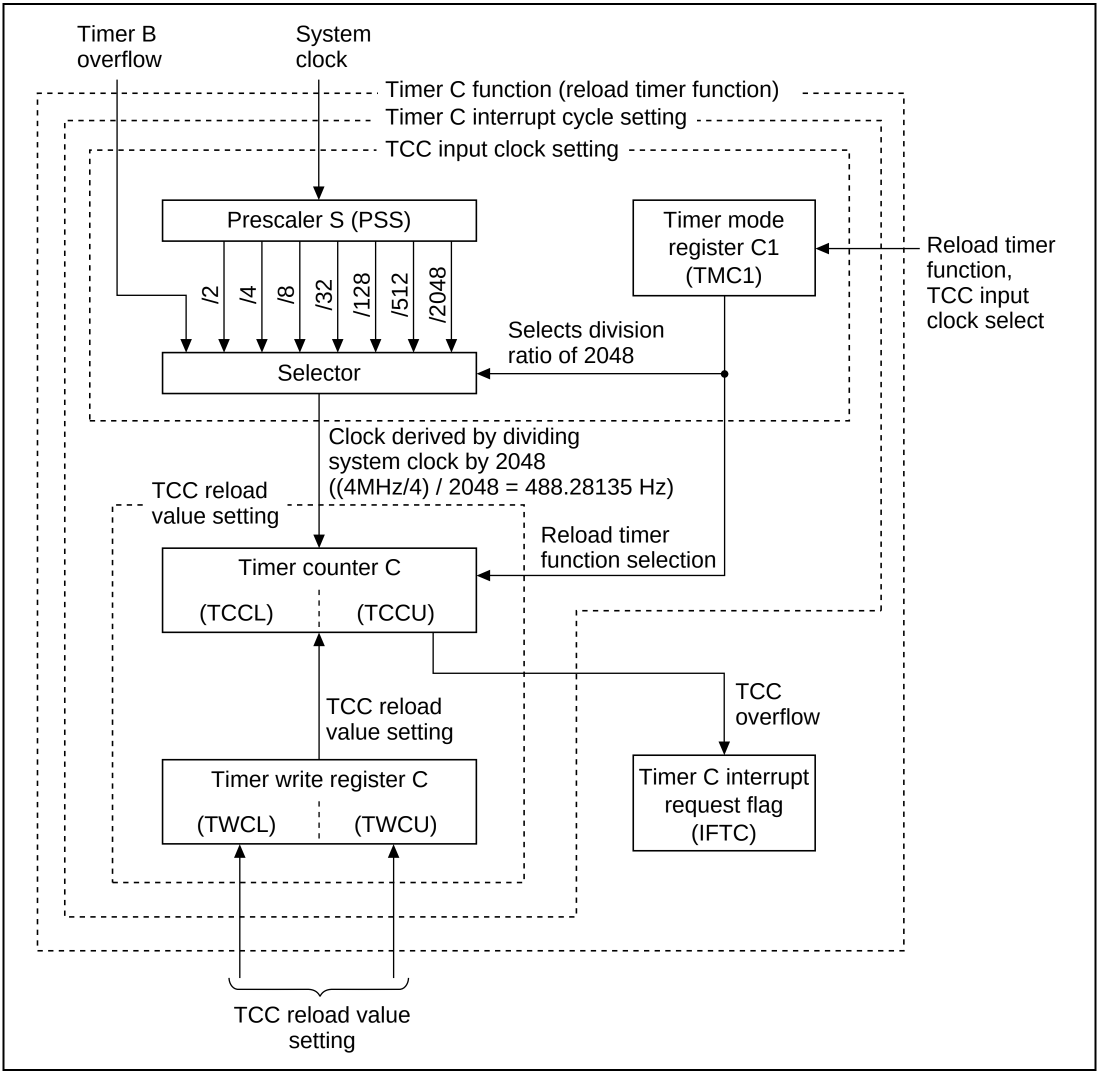


Figure 6 H4889 Series Timer C Function Block Diagram

e. Timer C is an 8-bit multifunction timer (free running/reload timer). In the H4889 Series, timer B overflow can be selected as the clock source, allowing timer B and timer C to be used as a 16-bit counter. In this example task, timer C is used as a reload timer. Table 2 describes the timer C functions.

Table 2 Timer C Functions

Timer Mode Register C (TMC)	Note: Applies to H4344/H4318/H4359/H4369 Series
Function	TMC is a 4-bit write-only register. It selects the timer C function (free-running/reload timer) and operating clock. TMC is initialized to $0 when reset and in stop mode.
Timer Mode Register C1 (TMC1)	Note: Applies to H4889 Series
Function	TMC1 is a 4-bit write-only register. It selects the timer C function (free-running/reload timer) and operating clock. TMC1 is initialized to $0 when reset and in stop mode.
Timer Write Register CL, U (TWCL, TWCU)	
Function	TWCL and TWCU form an 8-bit write-only register, which is made up of the lower digit (TWCL) and upper digit (TWCU). TWCL and TWCU are used for the initial TCC setting (the reload setting when operation as a reload timer).
Timer Counter C (TCC)	
Function	TCC is an 8-bit up-counter, which is incremented by the input internal clock. The TCC input clock is selected using bits TMC12 to TMC10 of TMC1. The value written to TWCL and TWCU is also written to TCC. When TCC overflows, the timer C interrupt request flag (IFTC) is set to “1”. If, at this point, timer C is set as a reload timer, the value of TWCL and TWCU is written to this counter and the count starts from this value. TCC is initialized to $00 when reset and in stop mode.
Timer C Interrupt Request Flag (IFTC)	
Function	IFTC reflects the existence of the timer C interrupt request. When timer C overflows, IFTC is set to “1”. IFTC can only be read/written to (only “0” can be written) using bit operation commands. Note that IFTC is not automatically cleared even when the interrupt is received, and must be cleared by writing “0” using software. IFTC is cleared at a reset and in stop mode.
Timer C Interrupt Mask (IMTC)	
Function	IMTC is the bit that masks IFTC. When IFTC is set to “1” and, additionally, IMTC is “0”, a timer C interrupt request is sent to the CPU (when IE = “1”). If IFTC is set to “1” but IMTC is “1”, no interrupt request is sent to the CPU and the timer C interrupt is held. IMTC can only be read or written to using bit operation commands. It is set to “1” at a reset and in stop mode.

f. Figure 7 is a block diagram of the R0 port functions in the H4344/H4318/H4359/H4369 Series. Figure 8 is a block diagram of the R1 port functions in the H4889 Series.

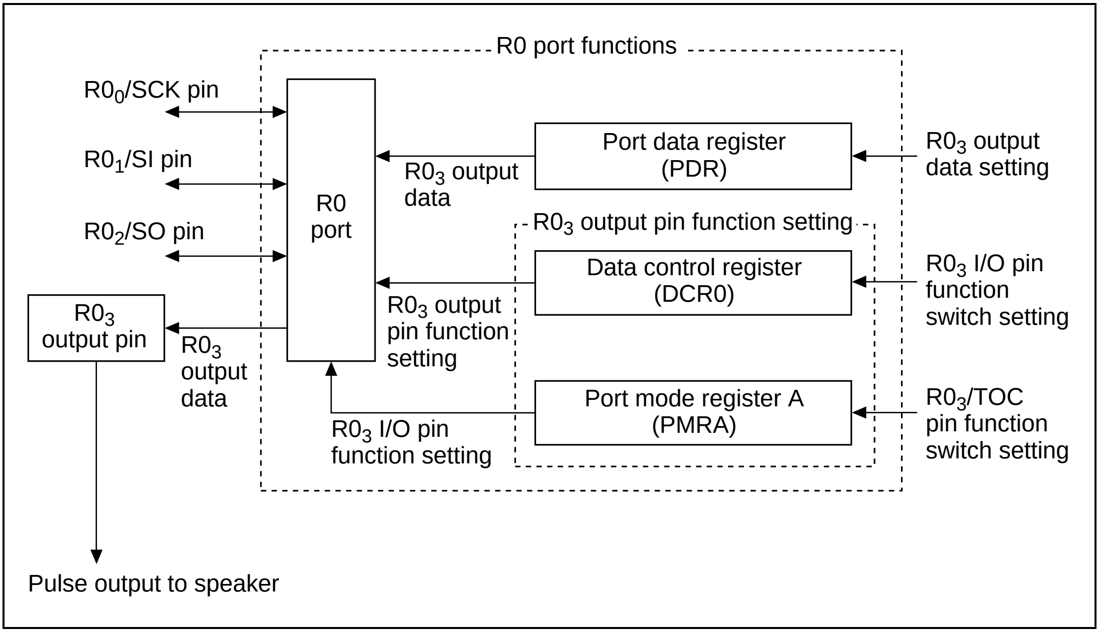


Figure 7 Function Block Diagram of R0 Port in H4344/H4318/H4359/H4369 Series

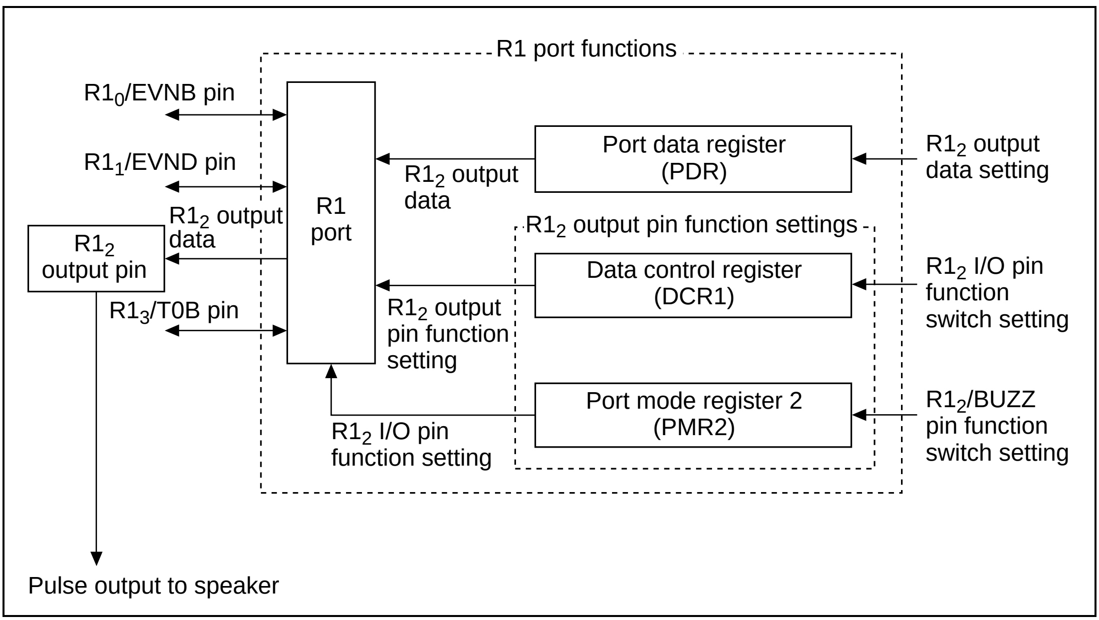


Figure 8 Function Block Diagram of R1 Port in H4889 Series

g. The R0 port in the H4344/H4318/H4359/H4369 Series, and the R1 port in the H4889 Series are 4-bit I/O ports. The LAR and LBR commands are used for 4-bit input, and the LRA and LRB commands for 4-bit output of both ports. The output data is stored in the PDR of the respective pin. In this example task, the $R0_3$ pin in the H4344/H4318/H4359/H4369 Series and the $R1_2$ pin the H4889 Series are set for output, and output a pulse to a speaker. Table 3 describes the functions of the R0 port in the H4344/H4318/H4359/H4369 Series and the R1 port in the H4889 Series.

Table 3 Description of R0 Port Functions in H4344/H4318/H4359/H4369 Series and R1 Port Functions in H4889 Series

Data Control Register R0 (DCR0)	Note: Applies to H4344/H4318/H4359/H4369 Series
Function	DCR0 switches the I/O pin function of the R0 port. When any bit of DCR0 is cleared to “0”, the output buffer (CMOS) of the corresponding pin is turned OFF and the output is set to high impedance. When the respective bit of DCR0 is set to “1”, the output buffer of the corresponding pin is set ON and the corresponding PDR value is output.
Data Control Register R1 (DCR1)	Note: Applies to H4889 Series
Function	DCR1 switches the I/O pin function of the R1 port. When any bit of DCR1 is cleared to “0”, the output buffer (CMOS) of the corresponding pin is turned OFF and the output is set to high impedance. When the respective bit of DCR1 is set to “1”, the output buffer of the corresponding pin is set ON and the corresponding PDR value is output.
Port Mode Register A (PMRA)	Note: Applies to H4344/H4318/H4359/H4369 Series
Function	PMRA is a 4-bit write-only register. Bits PMRA2 to PMRA0 switch the dual-function R0 port pins.
Port Mode Register 2 (PMR2)	Note: Applies to H4889 Series
Function	PMR2 is a 4-bit write-only register. Bits PMR23 to PMR20 switch the dual-function R1 port pins.
Port Data Register (PDR)	
Function	The I/O pins of the R ports have built-in PDRs to store the output data. When the LRA and LRB commands are executed, the contents of the accumulator (A) and B register (B) are transferred to the PDR of the specified R port. When the corresponding bit of the DCR of the R port is “1”, the output buffer of the appropriate pin is set ON and the value in the PDR is output via that pin. The PDR is initialized to $F at a reset.

3. Table 4 shows the allocation of functions in the example task.

Table 4 Function Allocation

Function	Function Allocation
System clock	The system clock is obtained by dividing the clock output from the system clock oscillator by 4. It is used for operating the CPU and internal peripheral modules. In this example task, a 4 MHz system clock oscillator is used, so the clock supplied to the CPU and internal peripheral modules is 1 MHz. The clock used by timer B and timer C is obtained by dividing the 1 MHz clock at PSS.
PSS	The clock input to timer B and timer C is obtained by dividing the system clock. The clock supplied to timer B is obtained by dividing the system clock by 4. The clock supplied to timer C is obtained by dividing the system clock by 2048.
TCB	This is an 8-bit up-counter. The count starts from the value set in TWBL and TWBU. When an overflow occurs, IFTB is set to “1”. After an overflow, the reload value set in TWBL and TWBU is set in TCB.
TWBL, TWBU	The TCB reload value is set in TWBL and TWBU. The reload value is determined from the pulse frequency to be output to the speaker. The frequencies corresponding to the various notes are stored in the data table.
TMB1	TMB1 selects the reload timer function for timer B and a clock obtained by dividing the system clock by 4 as the TCB input clock.
IFTB	IFTB reflects the existence of a timer B interrupt request. The pulse output pin output level is set in the timer B interrupt processing.
IMTB	Enables/disables timer B interrupt requests.
TCC	This is an 8-bit up-counter. The count starts from the value set in TWCL and TWCU. When an overflow occurs, IFTC is set to “1”. After an overflow, the reload value set in TWCL and TWCU is set in TCC.
TWCL, TWCU	The TCC reload value is set in TWCL and TWCU. The reload value is determined from the duration of pulse output to the speaker. The output duration for each note is stored in the data table.
TMC (H4344/H4318/H4359/H4369) TMC1 (H4889)	TMC (or TMC1) selects the reload timer function for timer C and a clock obtained by dividing the system clock by 2048 as the TCC input clock.
IFTC	Reflects the existence of a timer C interrupt request.
IMTC	Enables/disables timer C interrupt requests.
DCR0 (H4344/H4318/H4359/H4369) DCR1 (H4889)	Sets the $R0_3$ pins (H4344/H4318/H4359/H4369 Series) and $R1_2$ pins (H4889 Series) as output pins.

Table 4 Function Allocation (cont)

Function	Function Allocation
PMRA (H4344/H4318/ H4359/H4369) PMR2 (H4889)	Sets the $R0_3$/TOC dual-function pin (H4344/H4318/H4359/H4369 Series) as an $R0_3$ I/O pin and the $R1_2$/BUZZ dual-function pin (H4889 Series) as an $R1_2$ pin.
PDR	Stores the output data for the $R0_3$/$R1_2$ pin.
$R0_3$ pin	Output pin for the pulse in the H4344/H4318/H4359/H4369 Series.
$R1_2$ pin	Output pin for the pulse in the H4889 Series.

Description of Operation

1. Figure 9 shows the operating principles of the timer B, timer C, and $R0_3/R1_2$ output pins.

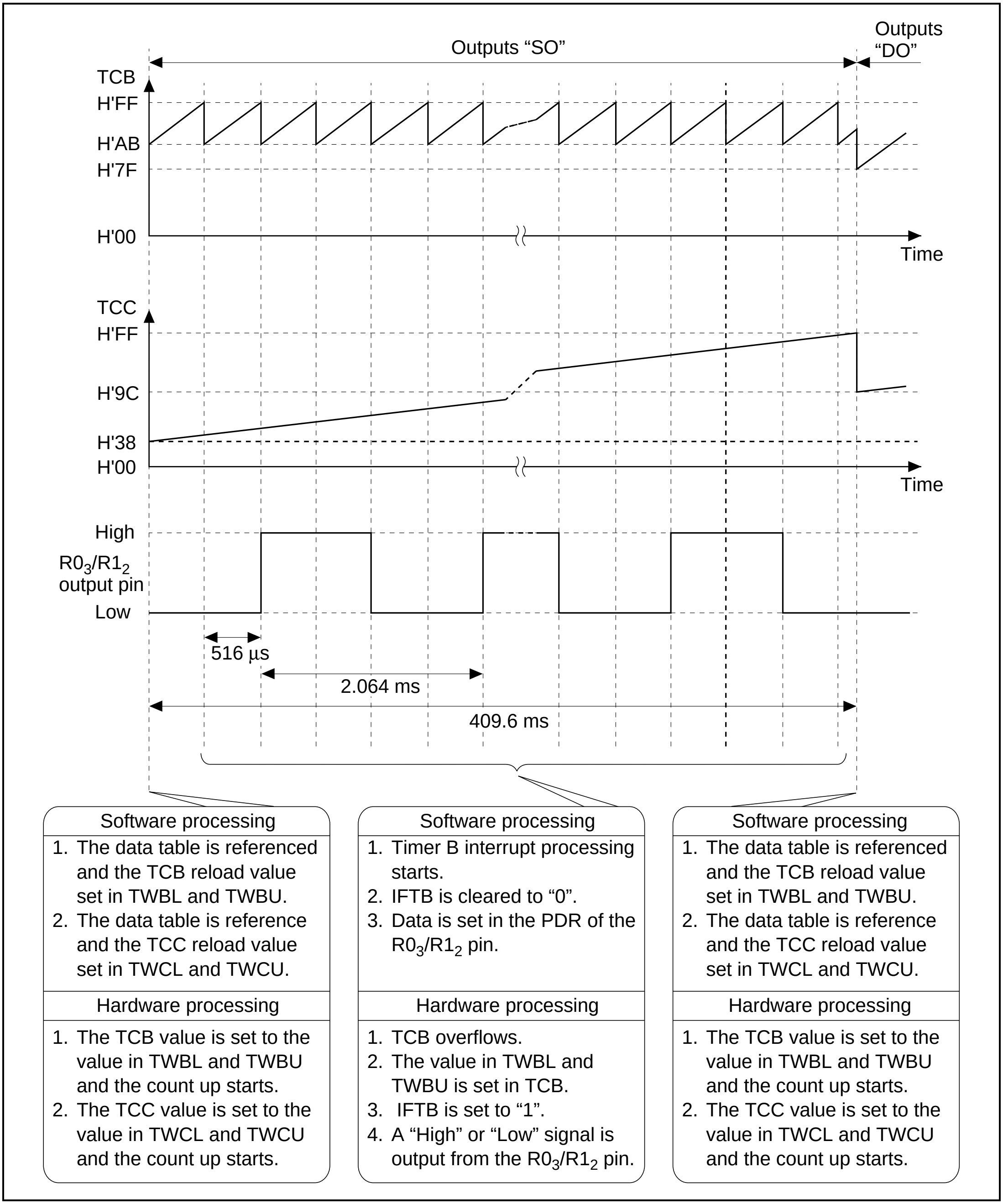


Figure 9 Operating Principles of Timer C, Timer B, and $R0_3/R1_2$ Output Pins

2. Figure 10 shows the operating principles for the pulse output from the $R0_3/R1_2$ pin.

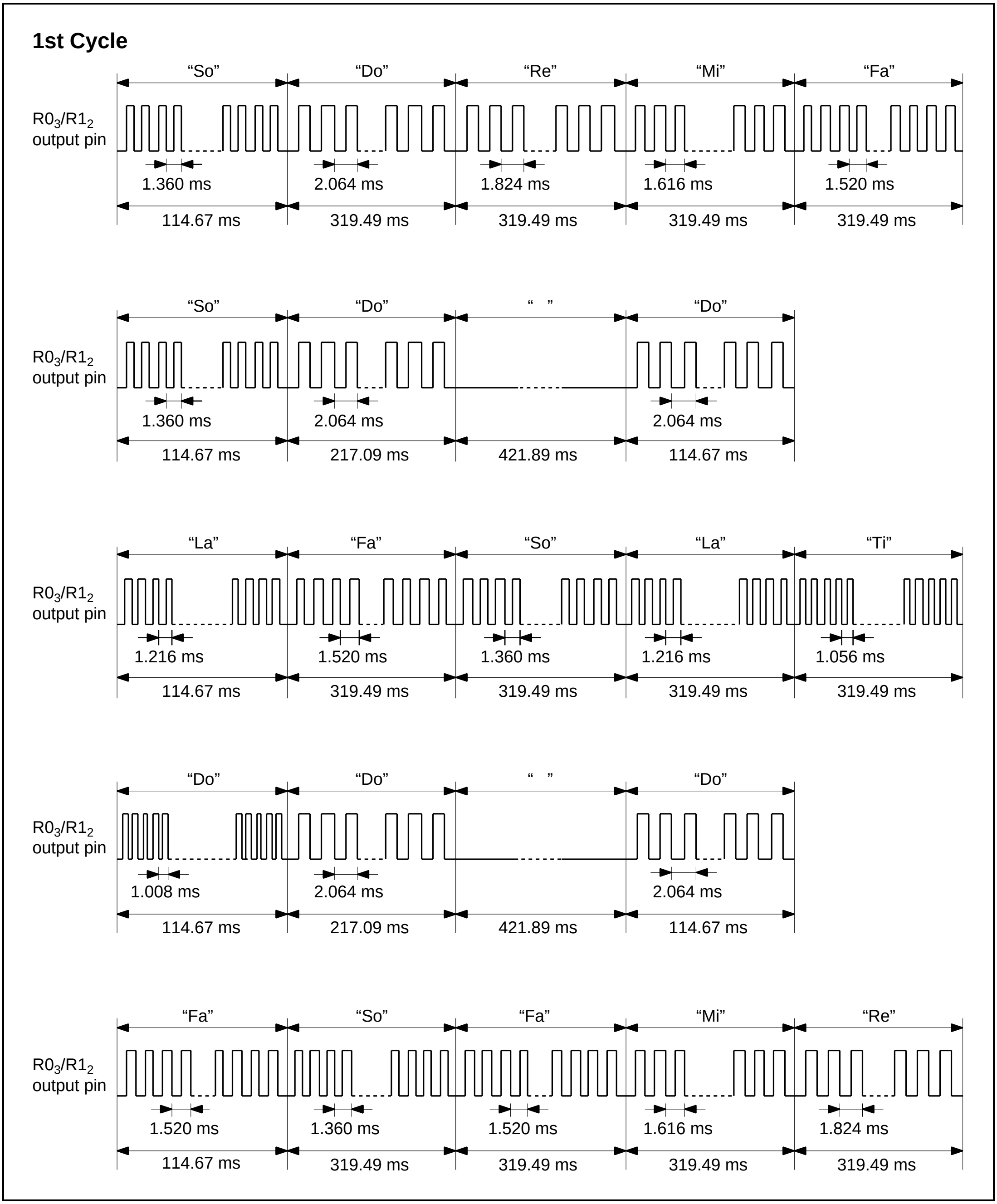


Figure 10 Operating Principles of Output Pulse

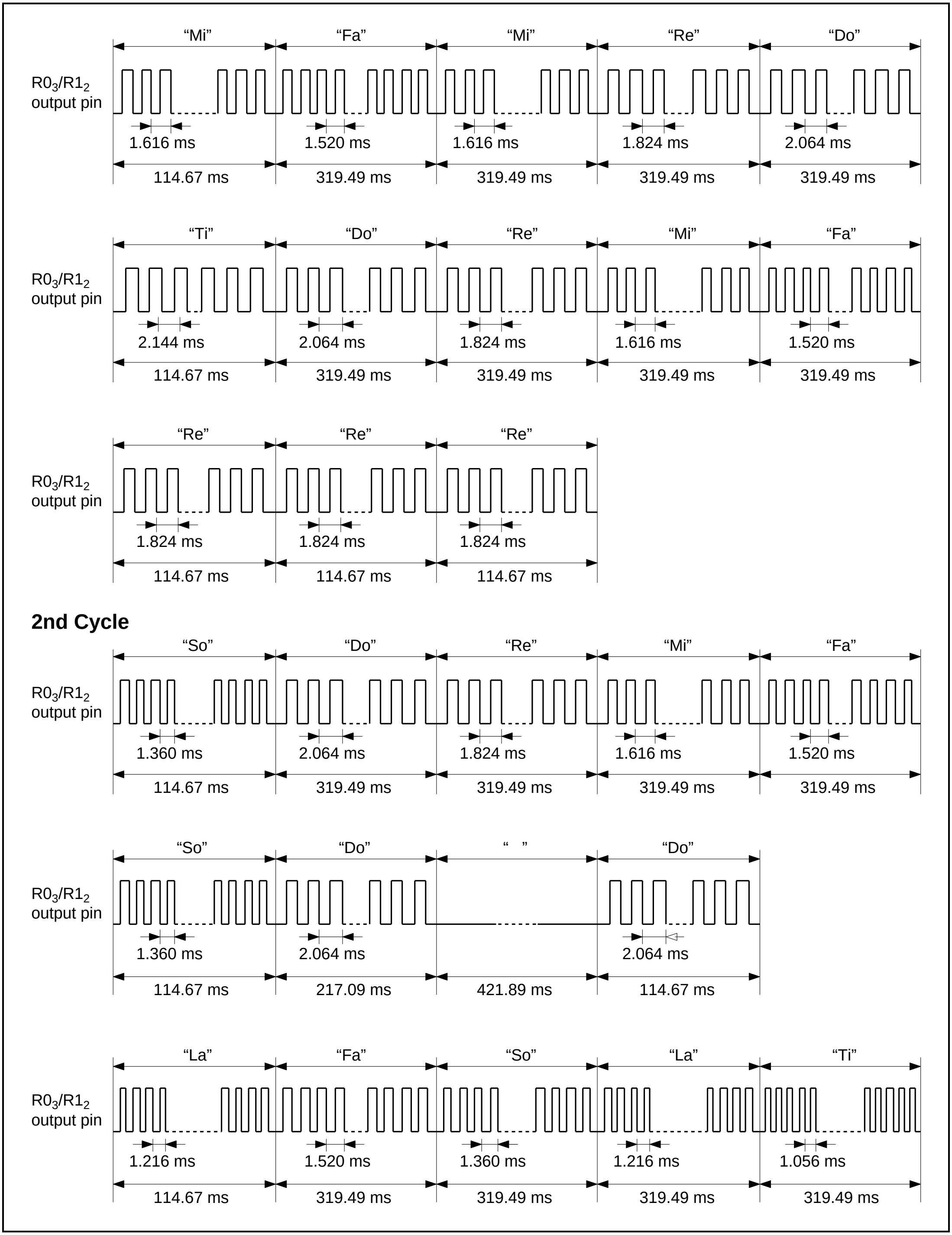


Figure 10 Operating Principles of Output Pulse (cont)

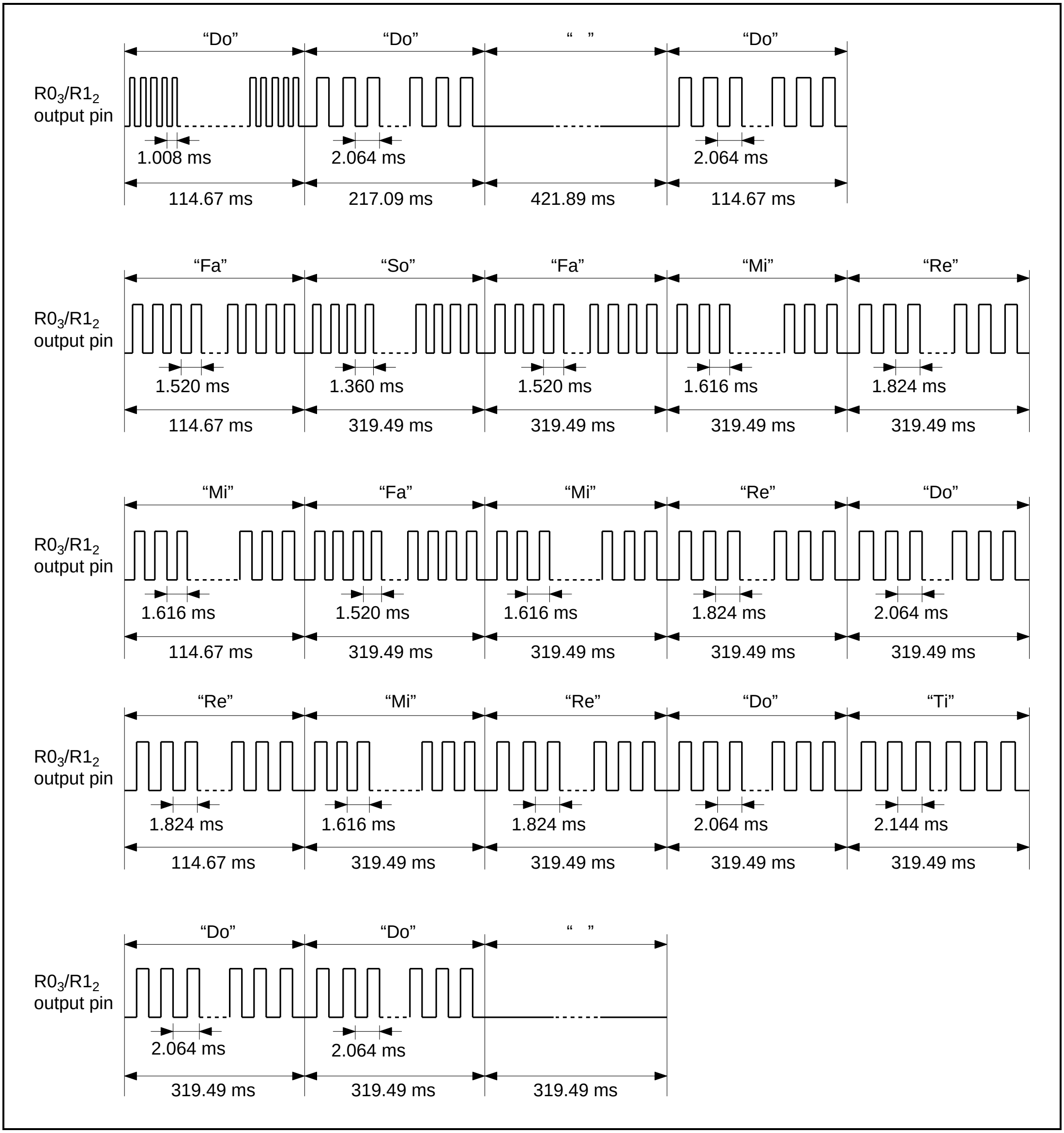
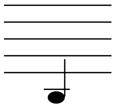
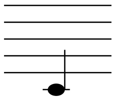
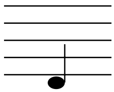
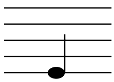
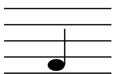
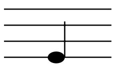
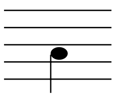
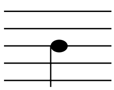
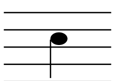


Figure 10 Operating Principles of Output Pulse (cont)

3. Table 5 shows the output pulse frequencies for the respective notes.

Table 5 Output Pulse Frequencies for Respective Notes

Note	Score	TCB Reload Value	TCB Overflow Cycle	Speaker Pulse Output Frequency
Ti		$7A	536 µs	1 / (536 µs × 4) ≈ 466.42 Hz
Do		$7F	516 µs	1 / (516 µs × 4) ≈ 484.50 Hz
Re		$8E	456 µs	1 / (456 µs × 4) ≈ 548.25 Hz
Mi		$9B	404 µs	1 / (404 µs × 4) ≈ 618.81 Hz
Fa		$A1	380 µs	1 / (380 µs × 4) ≈ 657.89 Hz
So		$AB	340 µs	1 / (340 µs × 4) ≈ 735.29 Hz
La		$B4	304 µs	1 / (304 µs × 4) ≈ 822.37 Hz
Ti		$BD	264 µs	1 / (264 µs × 4) ≈ 946.97 Hz
Do		$C1	252 µs	1 / (252 µs × 4) ≈ 992.06 Hz

Description of Functions

1. Description of Modules

   Table 6 describes the modules used in the example task.

Table 6 Description of Modules

Module	Label	Function
Main routine	SPLMN	This routine makes the initial stack pointer, timer B, timer C, and I/O port settings, enables the interrupts, and initializes the RAM to be used. It also sets the output pulse cycle created by the timer C overflow and the output duration, and initializes and sets those again each time another play starts.
Timer B interrupt processing routine	SPLINT	Controls the “High/Low” setting of the output pulse.

2. Description of Arguments

   No arguments are used in this example task.

3. Description of Internal Registers

a. Table 7 shows the internal registers of the H4344/H4318/H4359/H4369 used in this example.

Table 7 Internal Registers of H4344/H4318/H4359/H4369 Used in Example

Register	Description	RAM Address	Setting
IE	Interrupt Enable Flag This flag controls reception of all interrupts by the CPU. • When IE = “0”, CPU reception of all interrupts is disabled. • When IE = “1”, CPU reception is enabled.	0, $000	1
RSP	Reset Stack Pointer Clearing RSP to “0” initializes the stack pointer.	1, $000	0
IFTB	Timer B Interrupt Request Flag Reflects the existence of a timer B interrupt request. • When IFTB = “0”, no timer B interrupt is requested. • When IFTB = “1”, a timer B interrupt is requested.	0, $002	0
IMTB	Timer B Interrupt Mask This bit masks IFTB. • When IMTB = “0”, IFTB is enabled. • When IMTB = “1”, IFTB is masked.	1, $002	0
IFTC	Timer C Interrupt Request Flag Reflects the existence of a timer C interrupt request. • When IFTC = “0”, no timer C interrupt is requested. • When IFTC = “1”, a timer C interrupt is requested.	2, $002	0
IMTC	Timer C Interrupt Mask This bit masks IFTC. • When IMTC = “0”, IFTC is enabled. • When IMTC = “1”, IFTC is masked.	3, $002	1
PMRA	Port Mode Register A Bit 0 switches $R0_2$, SO pin functions, bit 1 switches $R0_1$/SI pin functions, bit 2 switches $R0_3$/TOC pin functions, and bit 3 switches D_3/BUZZ pin functions. • When PMRA = $0, pins $R0_2$, $R0_1$, $R0_3$ and D_3 are selected. Note: PMRA bit 3 cannot be used in the H4344 Series.	$004	$0

Table 7 Internal Registers of H4344/H4318/H4359/H4369 Used in Example (cont)

Register	Description	RAM Address	Setting
TMB1	Timer Mode Register B1 TMB13 selects timer B functions, TMB12 to TMB10 select the operating clock • When TMB13 = “1”, TMB12 = “1”, TMB11 = “1”, and TMB10 = “1”, timer B is set for reload timer functions and the operating clock is set for the system clock divided by 4.	$009	$D
TWBL	Timer Write Register BL Sets the lower digit of the TCB reload value.	$00A	$0
TWBU	Timer Write Register BU Sets the upper digit of the TCB reload value.	$00B	$0
TMC	Timer Mode Register C Selects timer C functions and operating clock. • When TMC3 = “1”, TMC2 = “0”, TMC1 = “0”, and TMC0 = “0”, timer C is set for reload timer functions and the operating clock is set for the system clock divided by 2048.	$00D	$8
TWCL	Timer Write Register CL Sets the lower digit of the TCC reload value.	$00E	$0
TWCU	Timer Write Register CU Sets the upper digit of the TCC reload value.	$00F	$0
TMB2	Timer Mode Register B2 Sets the input capture function and selects the detection edge of the EVNB pin input. • When TMB22 = “0”, free-running/reload timer B functions are selected. • When TMB22 = “1”, input capture timer B functions are selected. Note: The TMB22 bit cannot be used in the H4344. • When TMB21 = “0” and TMB20 = “0”, there is no edge detection of the EVNB pin input.	$026	$0

Table 7 Internal Registers of H4344/H4318/H4359/H4369 Used in Example (cont)

Register	Description	RAM Address	Setting
SSR1	System Clock Selection Register 1 Selects the system clock oscillation frequency, subsystem clock frequency division, and, in stop mode, the subsystem clock oscillation. • When SSR11 = "0", the system clock oscillation frequency is set to 0.4 to 1 MHz. • When SSR11 = "1", the system clock oscillation frequency is set to 1.6 to 5 MHz. Note: Applicable only to H4369	$027	$2
DCR0	Data Control Register R0 Controls the ON/OFF state of the R0 port output buffer. • When DCR03 = "0", the $R0_3$ pin output buffer is set OFF and output set to high impedance. • When DCR03 = "1", the $R0_3$ pin output buffer is set ON and the value in the corresponding PDR is output.	$030	$8

b. Table 8 describes the internal registers used in the H4889.

Table 8 Internal Registers Used in H4889

Register	Function	RAM Address	Setting
IE	Interrupt Enable Flag Controls whether interrupts are received by the CPU. • When IE = "0", CPU interrupt reception is disabled. • When IE = "1", CPU interrupt reception is enabled.	0, $000	1
RSP	Reset Stack Pointer Clearing RSP to "0" initializes the stack pointer.	1, $000	0
IFTB	Timer B Interrupt Request Flag Reflects the existence of a timer B interrupt request. • When IFTB = "0", no timer B interrupt is requested. • When IFTB = "1", a timer B interrupt is requested.	2, $002	0
IMTB	Timer B Interrupt Mask This bit masks IFTB. • When IMTB = "0", IFTB is enabled. • When IMTB = "1", IFTB is masked.	3, $002	0
IFTC	Timer C Interrupt Request Flag Reflects the existence of a timer C interrupt request. • When IFTC = "0", no timer C interrupt is requested. • When IFTC = "1", a timer C interrupt is requested.	0, $003	0
IMTC	Timer C Interrupt Mask This bit masks IFTC. • When IMTC = "0", IFTC is enabled. • When IMTC = "1", IFTC is masked.	1, $003	1
SSR	System Clock Selection Register Selects the system clock oscillation frequency, subsystem clock frequency division, the subsystem clock oscillation, and the system clock division ratio for stop mode. • When SSR1 = "0", the system clock oscillation frequency is set to 0.4 to 1.0 MHz. • When SSR1 = "1", the system clock oscillation frequency is set to 1.6 to 4.5 MHz.	$004	$2

Table 8 Internal Registers Used in H4889 (cont)

Register	Function	RAM Address	Setting
PMR2	Port Mode Register 2 Bit 0 switches $R1_0$/EVNB pin functions, bit 1 switches $R1_1$/EVND pin functions, bit 2 switches $R1_2$/BUZZ pin functions, and bit 3 switches $R1_3$/TOB pin functions. • When PMR2 = $0, $R1_0$, $R1_1$, $R1_2$ and $R1_3$ pins are selected.	$00A	$0
TMB1	Timer Mode Register B1 TMB13 selects timer B functions, TMB12 to TMB10 select the operating clock. • When TMB13 = “1”, TMB12 = “1”, TMB11 = “0” and TMB10 = “1”, timer B has reload timer functions, and the operating clock is set to the system clock divided by 4.	$010	$D
TMB2	Timer Mode Register B2 Selects timer B output mode and EVNB pin input detection edge. • When TMB22 = “0”, timer B output is set to a toggle waveform. • When TMB22 = “1”, timer B output is set for PWM output. • When TMB21 = “0” and TMB20 = “0”, there is no EVNB pin input edge detection.	$011	$0
TWBL	Timer Write Register BL Sets the lower digit of the TCB reload value.	$012	$0
TWBU	Timer Write Register BU Sets the upper digit of the TCB reload value.	$013	$0
TMC1	Timer Mode Register C1 Selects timer C functions and operating clock. • When TMC13 = “1”, TMC12 = “0”, TMC11 = “0”, and TMC10 = “0”, timer C functions as a reload timer, and the operating clock is set to the system clock divided by 2048.	$014	$8
TWCL	Timer Write Register CL Sets the lower digit of the TCC reload value.	$016	$0
TWCU	Timer Write Register CU Sets the upper digit of the TCC reload value.	$017	$0

Table 8 Internal Registers Used in H4889 (cont)

Register	Function	RAM Address	Setting
DCR1	Data Control Register R1 Controls the ON/OFF state of the R1 port output buffer. • When DCR12 = “0”, the $R1_2$ pin output buffer is set OFF and the output is set to high impedance. • When DCR12 = “1”, the $R1_2$ pin output buffer is set ON and the value of the corresponding PDR is output.	$035	$4

4. Description of RAM

Table 9 shows the RAM used in this example task.

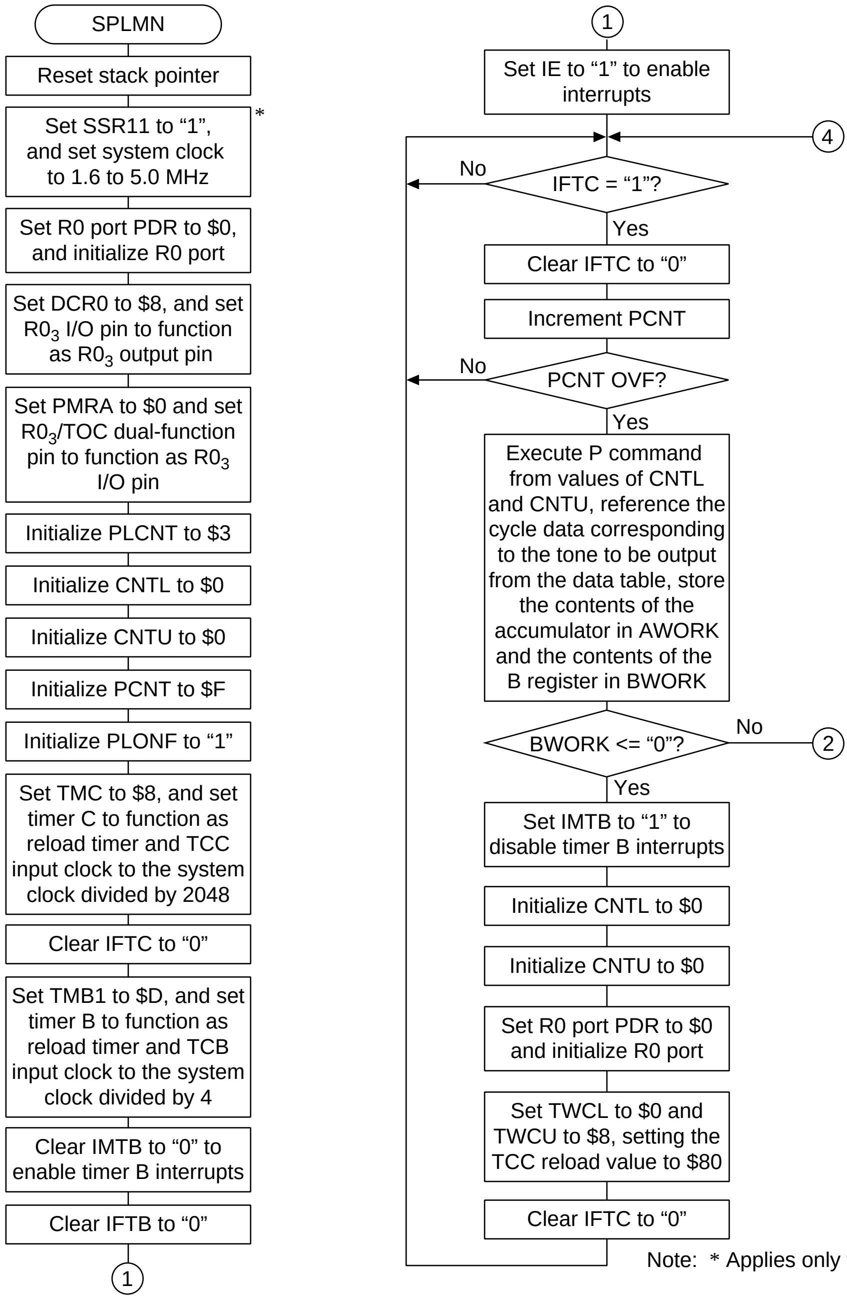
Table 9 Used RAM

Label	Function	RAM Address	Module
AESC	Stores the contents of the accumulator during timer B interrupt processing.	$040	SPLINT
BESC	Stores the contents of the B register during timer B interrupt processing.	$041	SPLINT
PLCNT	This counter controls the High/Low output when outputting a pulse from the $R0_3$/$R1_2$ pin.	$090	SPLMN, SPLINT
PLONF	This flag controls the enabling/disabling of pulse output.	0, $091	SPLMN, SPLINT
CNTL	Stores the contents of the accumulator used for specifying the address when executing a pattern command.	$093	SPLMN
CNTU	Stores the contents of the B register used for specifying the address when executing a pattern command.	$094	SPLMN
PCNT	This counter controls the number of times a pulse of a given frequency is output.	$095	SPLMN
AWORK	Temporarily stores the contents of the accumulator during processing of the main routine.	$097	SPLMN
BWORK	Temporarily stores the contents of the B register during processing of the main routine.	$098	SPLMN

Flow charts

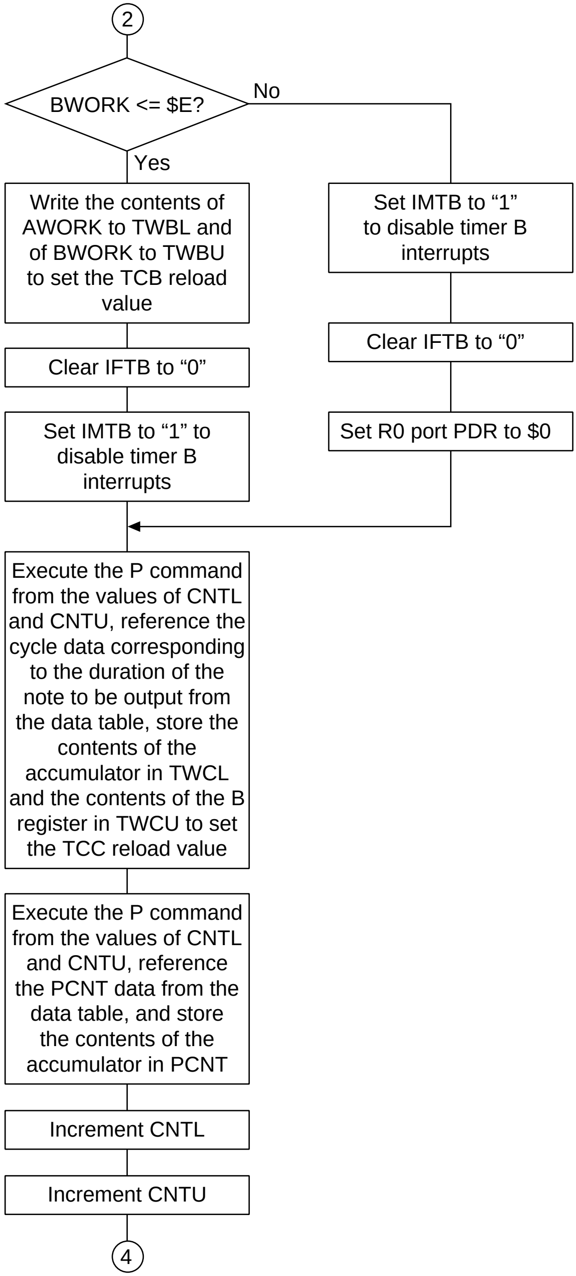
1. H4344/H4318/H4359/H4369

a. Main Routine

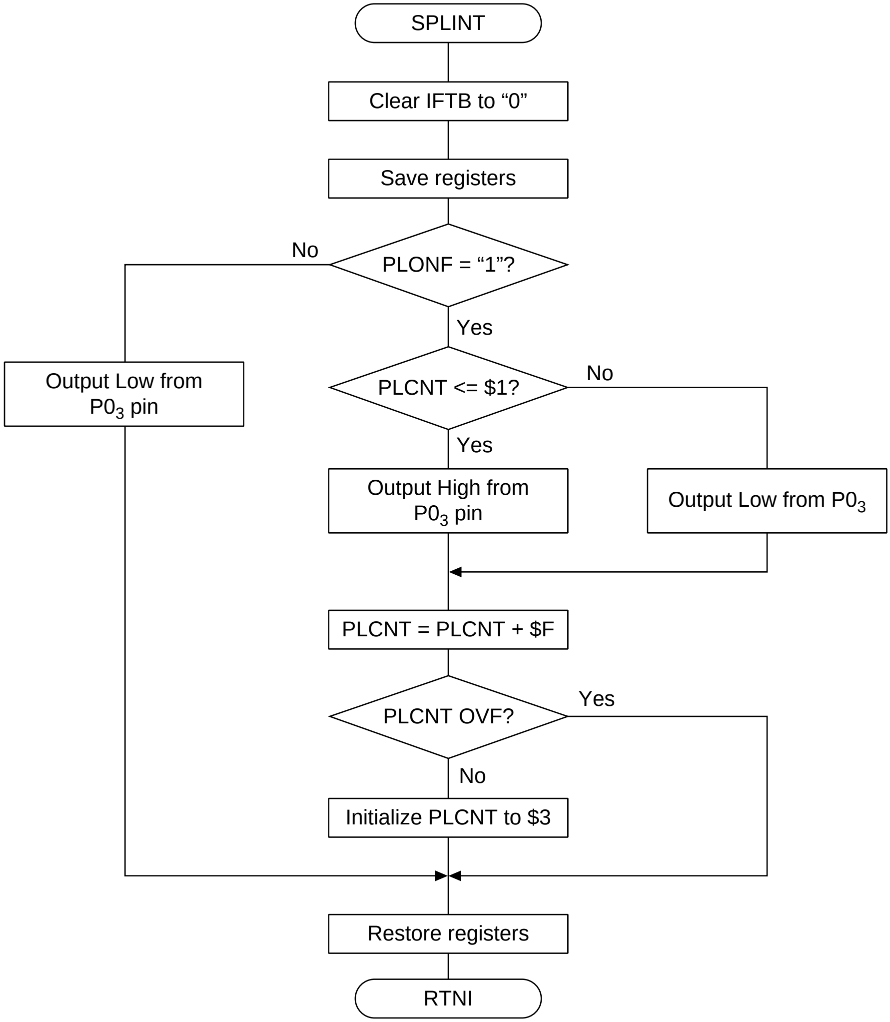


Note: * Applies only to H4369.

RENESAS

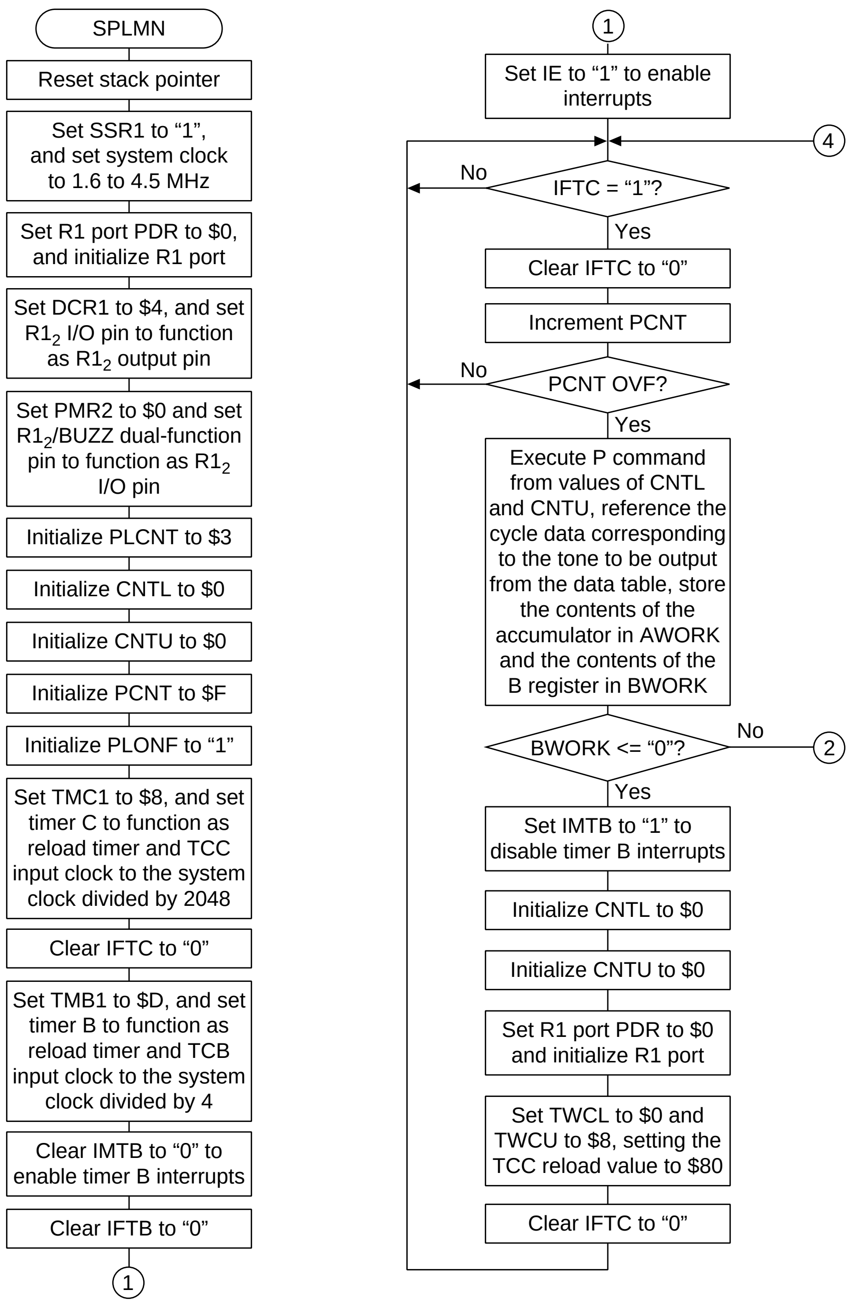

2
BWORK <= $E?
No
Yes
Write the contents of AWORK to TWBL and of BWORK to TWBU to set the TCB reload value
Clear IFTB to “0”
Set IMTB to “1” to disable timer B interrupts
Set IMTB to “1” to disable timer B interrupts
Clear IFTB to “0”
Set R0 port PDR to $0
Execute the P command from the values of CNTL and CNTU, reference the cycle data corresponding to the duration of the note to be output from the data table, store the contents of the accumulator in TWCL and the contents of the B register in TWCU to set the TCC reload value
Execute the P command from the values of CNTL and CNTU, reference the PCNT data from the data table, and store the contents of the accumulator in PCNT
Increment CNTL
Increment CNTU
4

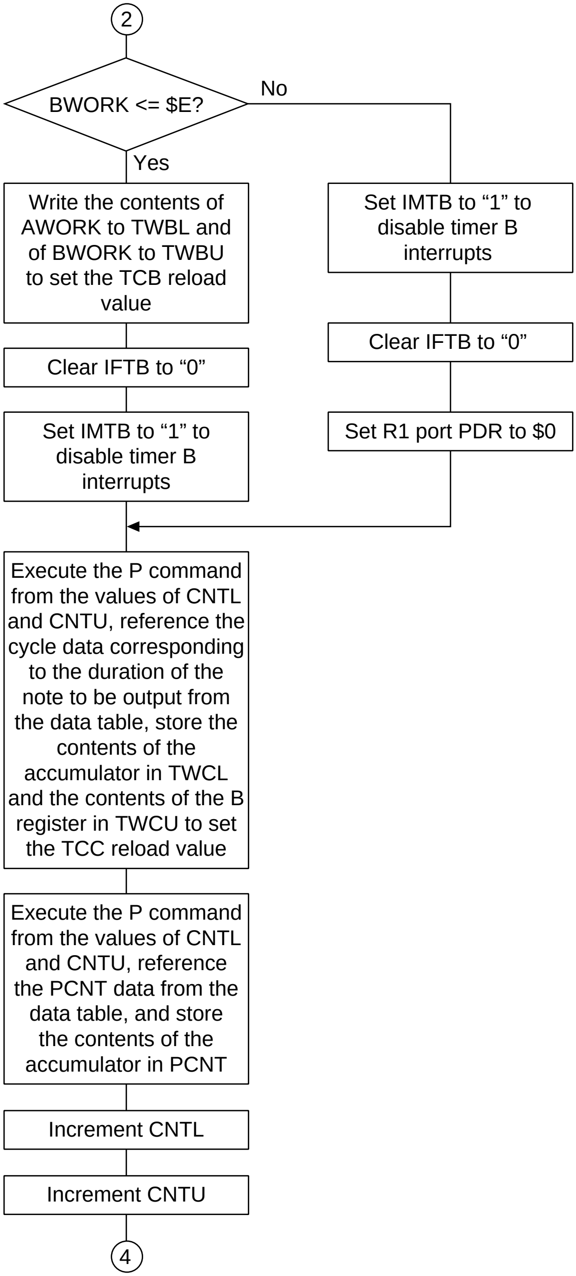
b. Timer B Interrupt Processing Routine



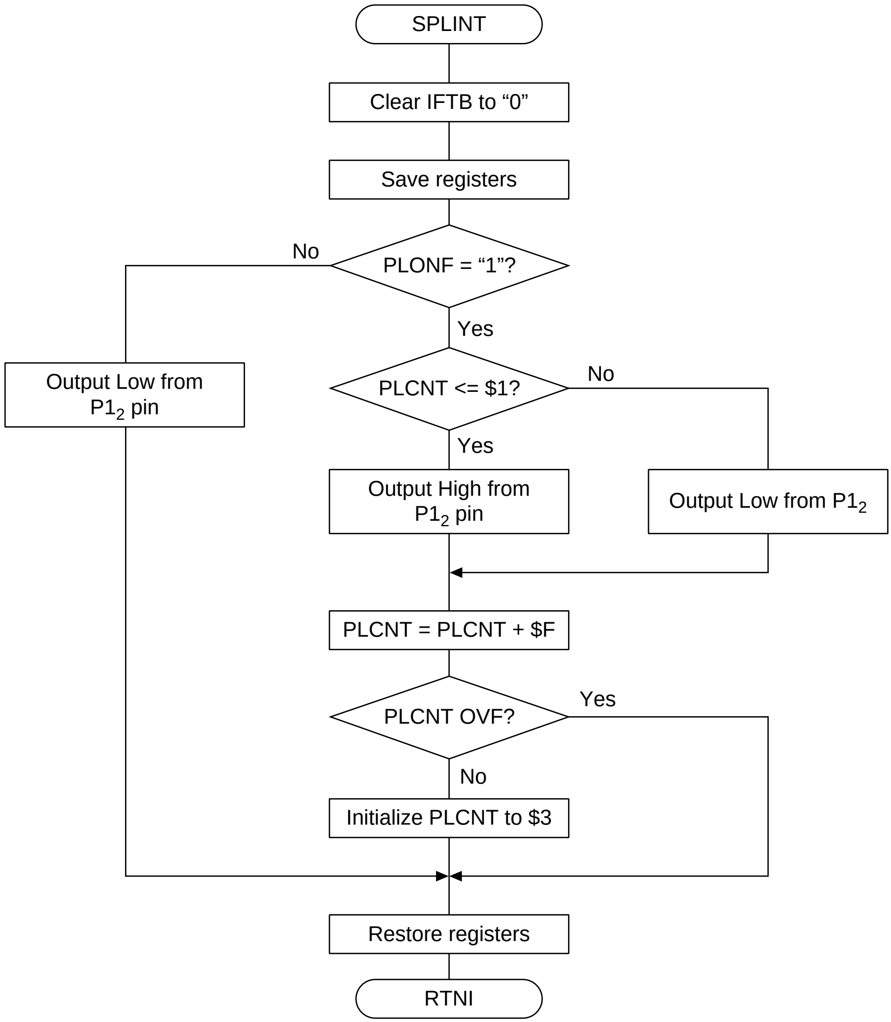
2. H4889

a. Main Routine




2
BWORK <= $E?
No
Yes
Write the contents of AWORK to TWBL and of BWORK to TWBU to set the TCB reload value
Clear IFTB to “0”
Set IMTB to “1” to disable timer B interrupts
Set IMTB to “1” to disable timer B interrupts
Clear IFTB to “0”
Set R1 port PDR to $0
Execute the P command from the values of CNTL and CNTU, reference the cycle data corresponding to the duration of the note to be output from the data table, store the contents of the accumulator in TWCL and the contents of the B register in TWCU to set the TCC reload value
Execute the P command from the values of CNTL and CNTU, reference the PCNT data from the data table, and store the contents of the accumulator in PCNT
Increment CNTL
Increment CNTU
4

b. Timer B Interrupt Processing Routine



RENESAS

Program Listing

1. H4344

```
************************************************************
*
*         HMCS400 Series Application Note
*
*         'Sound Play
*         - Minuet : J.S.Bach'
*
*         MCU : H4344
*
*         External Clock : 4MHz
*         Internal Clock : 1MHz
*         Sub Clock      : 32.768kHz
*
************************************************************
*
************************************************************
*         Symbol Definition
************************************************************
*
IE        equ     0,$000    Interrupt Enable Flag
RSP       equ     1,$000    Reset Stack Pointer
IF0       equ     2,$000    INT0 Interrupt Request Flag
IM0       equ     3,$000    INT0 Interrupt Mask
*
IFTB      equ     0,$002    Timer B Interrupt Request Flag
IMTB      equ     1,$002    Timer B Interrupt Mask
IFTC      equ     2,$002    Timer C Interrupt Request Flag
IMTC      equ     3,$002    Timer C Interrupt Mask
*
IFAD      equ     0,$003    A/D Converter Interrupt Request Flag
IMAD      equ     1,$003    A/D Converter Interrupt Mask
IFS       equ     2,$003    Serial Interrupt Request Flag
IMS       equ     3,$003    Serial Interrupt Mask
*
PMRA      equ     $004      Port Mode Register
SMR       equ     $005      Serial Mode Register
SRL       equ     $006      Serial Data Register L
SRU       equ     $007      Serial Data Register U
*
TMB1      equ     $009      Timer Mode Register B1
TRBL      equ     $00A      Timer Read Register BL
TWBL      equ     $00A      Timer Write Register BL
TRBU      equ     $00B      Timer Read Register BU
TWBU      equ     $00B      Timer Write Register BU
MIS       equ     $00C      Miscellaneous Register
TMC       equ     $00D      Timer Mode Register C
TRCL      equ     $00E      Timer Read Register CL
```

```
TWCL     equ     $00E       Timer Write Register CL
TRCU     equ     $00F       Timer Read Register CU
TWCU     equ     $00F       Timer Write Register CU
ACR      equ     $016       A/D Channel Register
ADRL     equ     $017       A/D Data Register L
ADRU     equ     $018       A/D Data Register U
AMR1     equ     $019       A/D Mode Register 1
AMR2     equ     $01A       A/D Mode Register 2
*
WDON     equ     1,$020     Watchdog on Flag
ADSF     equ     2,$020     A/D Start Flag
*
IAOF     equ     2,$021     IAD off Flag
RAME     equ     3,$021     RAM Enable Flag
*
PMRB     equ     $024       Port Mode Register B
PMRC     equ     $025       Port Mode Register C
TMB2     equ     $026       Timer Mode Register B2
*
DCD0     equ     $02C       D Port Data Control Register 0
DCD1     equ     $02D       D Port Data Control Register 1
*
DCR0     equ     $030       R Port Data Control Register 0
DCR1     equ     $031       R Port Data Control Register 1
DCR2     equ     $032       R Port Data Control Register 2
DCR3     equ     $033       R Port Data Control Register 3
*
************************************************************
*        RAM ALLOCATION
************************************************************
*
AESC     equ     $040       Accumulator Escape
BESC     equ     $041       B Register Escape
*
PLCNT    equ     $090       Pulse Counter
*
PLFLG1   equ     $091       Pulse Flag 1
PLONF    equ     0,PLFLG1   Pulse Output Enable Flag
*
CNTL     equ     $093       Lower Counter
CNTU     equ     $094       Upper Counter
PCNT     equ     $095       Period Counter
AWORK    equ     $096       Accumulator Work RAM Area
BWORK    equ     $097       B Register Work RAM Area
*
************************************************************
*        Vector Address
************************************************************
*
         org     $0000
*
```

```
         JMPL    SPLMN      Reset Interrupt
         JMPL    SPLMN      INT0 Interrupt
*
         org     $0008
*
         JMPL    SPLINT     Timer B Interrupt
         JMPL    SPLMN      Timer C Interrupt
         JMPL    SPLMN      A/D Converter Interrupt
         JMPL    SPLMN      Serial Interrupt
*
************************************************************
*        Main Program
************************************************************
*
         org     $1000
*
SPLMN    REMD    RSP        Reset Stack Pointer
*
         LAI     0
         LRA     0          Initialize R0 Port PDR
         LMID    8,DCR0     Initialize R03 Output Terminal Function
         LMID    0,PMRA     Initialize R03 Input/Output Terminal Function
*
         LMID    3,PLCNT    Initialize Pulse Counter
         LMID    0,CNTL     Initialize Lower Counter
         LMID    0,CNTU     Initialize Upper Counter
         LMID    $F,PCNT    Initialize Period Counter
         SEMD    PLONF      Initialize Pulse Output Enable Flag
*
         LMID    8,TMC      Initialize Timer C Function & Input Clock Period
         REMD    IFTC       Clear IFTC to 0
*
         LMID    $D,TMB1    Initialize Timer B Function & Input Clock Period
         LMID    0,TMB2     Initialize Timer B Function
*
         SEMD    IMTB       Timer B Interrupt Disable
         REMD    IFTB       Clear IFTB to 0
*
         SEMD    IE         All Interrupt Enable
*
SPLMN00  TMD     IFTC       IFTC = "1" ?
         BRS     SPLMN10    Yes. Branch to SPLMN10
         BRS     SPLMN00    No. Branch to SPLMN00
*
         REMD    IFTC       Clear IFTC to 0
         LMID    9,TWCL     Set TCC Reload Value Lower
         LMID    0,TWCU     Set TCC Reload Value Upper
*
SPLMN05  TMD     IFTC       IFTC = "1" ?
         BRS     SPLMN10    Yes. Branch to SPLMN10
         SEMD    IMTB       Timer B Interrupt Disable
```

```
          REMD    IFTB      Clear IFTB to 0
          LAI     0
          LRA     0         Sound off
          BRS     SPLMN05   Branch to SPLMN05
*
SPLMN10   REMD    IFTC      Clear IFTC to 0
          LAMD    PCNT      Load PCNT
          AI      1         Increment PCNT
          BRS     SPLMN40   PCNT Overflow ? Yes. Branch to SPLMN40
          LMAD    PCNT      No. Save PCNT
          BRS     SPLMN00   Branch to SPLMN00
*
SPLMN40   LAMD    CNTU      Load CNTL
          LBA
          LAMD    CNTL      Load CNTU
          P       2         Scale Data Pattern Generation
          LMAD    AWORK     Save Scale Lower Data
          LAB
          LMAD    BWORK     Save Scale Upper Data
          ALEI    0         Scale Upper Data <= $0 ? End Sound Play ?
          BRS     SPLMN90   Yes. Branch to SPLMN90
          ALEI    $E        Scale Upper Data <= $E ?
          BRS     SPLMN20   Yes. Branch to SPLMN20
          SEMD    IMTB      Timer B Interrupt Disable
          REMD    IFTB      Clear IFTB to 0
          LAI     0
          LRA     0         R03 Output Terminal is "Low" Output
          BRS     SPLMN30   Branch to SPLMN30
*
SPLMN20   LAMD    AWORK     Load AWORK
          LMAD    TWBL      Set TCB Reload Value Lower to Scale Data
          LAMD    BWORK     Load BWORK
          LMAD    TWBU      Set TCB Reload Value Upper to Scale Data
          REMD    IFTB      Clear IFTB to 0
          REMD    IMTB      Timer B Interrupt Enable
*
SPLMN30   LAMD    CNTU      Load Upper Counter
          LBA
          LAMD    CNTL      Load Lower Counter
          P       3         Time Period Data Pattern Generation
          LMAD    TWCL      Set TCC Reload Value Lower to Time Period Lower Data
          LAB
          LMAD    TWCU      Set TCC Reload Value Upper to Time Period Upper Data
*
          LAMD    CNTU      Load Upper Counter
          LBA
          LAMD    CNTL      Load Lower Counter
          P       4         Time Counter Data Pattern Generation
          LMAD    PCNT      Set PCNT to Time Counter Data
*
          SEC               Set Carry Flag at 1
```

```
         LAI     0
         AMCD    CNTL       Increment Lower Counter
         LMAD    CNTL       Save Lower Counter
         LAI     0
         AMCD    CNTU       If CNTL Overflow, Increment Upper Counter
         LMAD    CNTU       Save Upper Counter
         BR      *+1
         BRS     SPLMN00    Branch to SPLMN00
*
SPLMN90  SEMD    IMTB       Timer B Interrupt Disable
         LMID    0,CNTL     Initialize Lower Counter
         LMID    0,CNTU     Initialize Upper Counter
         LAI     0
         LRA     0          Initialize R0 Port PDR
         LMID    0,TWCL     Initialize TCC Reload Value Lower
         LMID    8,TWCU     Initialize TCC Reload Value Upper
         REMD    IFTC       Clear IFTC to 0
         BRS     SPLMN00    Branch to SPLMN00
*
************************************************************
*        Timer B Interrupt Process
*        ---Pulse Output Routine---
************************************************************
*
SPLINT   REMD    IFTB       Clear IFTB to 0
*
         LMAD    AESC       Store Accumulator
         LAB
         LMAD    BESC       Store B Register
*
         TMD     PLONF      Pulse Enable Flag = "1" ?
         BRS     PLI00      Yes. Branch to PLI00
         LAI     0          No. Stop Output Pulse
         LRA     0          R03 Output Terminal is "Low" Output
         BRS     PLI99      Branch to PLI99
*
PLI00    LAMD    PLCNT      Load Pulse Counter Value
         LBA                Store Pulse Counter Value
         ALEI    1          PLCNT <= $1 ? Is R03 Output Terminal "High" Output ?
         BRS     PLI10      Yes. Branch to PLI10
         LAI     0          No. R03 Output Terminal is "Low" Output
         BRS     PLI20      Branch to PLI20
PLI10    LAI     8
PLI20    LRA     0          R03 Output Terminal is "High" Output
         LAB                Restore Pulse Counter Value
         AI      $F         PLCNT + $F <= $F ? Pulse Counter = $0 ?
         BRS     PLI30      No. Branch to PLI30
         LAI     3          Yes. Pulse Counter Initialize
PLI30    LMAD    PLCNT      Save Pulse Counter Value
*
PLI99    LAMD    BESC       Restore B Register
```

```
          LBA
          LAMD    AESC       Restore Accumulator
*
          RTNI               Return from Interrupt
*
************************************************************
*         Scale Data
************************************************************
*
          ORG     $0200
*
*** 1st Cycle
*
          dc      $1AB       'G' Data. TCB Reload Value = $AB
          dc      $17F       'C' Data. TCB Reload Value = $7F
          dc      $18E       'D' Data. TCB Reload Value = $8E
          dc      $19B       'E' Data. TCB Reload Value = $9B
          dc      $1A1       'F' Data. TCB Reload Value = $A1
*
          dc      $1AB       'G' Data. TCB Reload Value = $AB
          dc      $17F       'C' Data. TCB Reload Value = $7F
          dc      $1FF       ' ' Data. TCB Reload Value = $FF
          dc      $17F       'C' Data. TCB Reload Value = $7F
*
          dc      $1B4       'A' Data. TCB Reload Value = $B4
          dc      $1A1       'F' Data. TCB Reload Value = $A1
          dc      $1AB       'G' Data. TCB Reload Value = $AB
          dc      $1B4       'A' Data. TCB Reload Value = $B4
          dc      $1BD       'B' Data. TCB Reload Value = $BD
*
          dc      $1C1       'C' Data. TCB Reload Value = $C1
          dc      $17F       'C' Data. TCB Reload Value = $7F
          dc      $1FF       ' ' Data. TCB Reload Value = $FF
          dc      $17F       'C' Data. TCB Reload Value = $7F
*
          dc      $1A1       'F' Data. TCB Reload Value = $A1
          dc      $1AB       'G' Data. TCB Reload Value = $AB
          dc      $1A1       'F' Data. TCB Reload Value = $A1
          dc      $19B       'E' Data. TCB Reload Value = $9B
          dc      $18E       'D' Data. TCB Reload Value = $8E
*
          dc      $19B       'E' Data. TCB Reload Value = $9B
          dc      $1A1       'F' Data. TCB Reload Value = $A1
          dc      $19B       'E' Data. TCB Reload Value = $9B
          dc      $18E       'D' Data. TCB Reload Value = $8E
          dc      $17F       'C' Data. TCB Reload Value = $7F
*
          dc      $17A       'B' Data. TCB Reload Value = $7A
          dc      $17F       'C' Data. TCB Reload Value = $7F
          dc      $18E       'D' Data. TCB Reload Value = $8E
          dc      $19B       'E' Data. TCB Reload Value = $9B
```

```
        dc      $17F        'C' Data. TCB Reload Value = $7F
*
        dc      $18E        'D' Data. TCB Reload Value = $8E
*
*** 2nd Cycle
*
        dc      $1AB        'G' Data. TCB Reload Value = $AB
        dc      $17F        'C' Data. TCB Reload Value = $7F
        dc      $18E        'D' Data. TCB Reload Value = $8E
        dc      $19B        'E' Data. TCB Reload Value = $9B
        dc      $1A1        'F' Data. TCB Reload Value = $A1
*
        dc      $1AB        'G' Data. TCB Reload Value = $AB
        dc      $17F        'C' Data. TCB Reload Value = $7F
        dc      $1FF        ' ' Data. TCB Reload Value = $FF
        dc      $17F        'C' Data. TCB Reload Value = $7F
*
        dc      $1B4        'A' Data. TCB Reload Value = $B4
        dc      $1A1        'F' Data. TCB Reload Value = $A1
        dc      $1AB        'G' Data. TCB Reload Value = $AB
        dc      $1B4        'A' Data. TCB Reload Value = $B4
        dc      $1BD        'B' Data. TCB Reload Value = $BD
*
        dc      $1C1        'C' Data. TCB Reload Value = $C1
        dc      $17F        'C' Data. TCB Reload Value = $7F
        dc      $1FF        ' ' Data. TCB Reload Value = $FF
        dc      $17F        'C' Data. TCB Reload Value = $7F
*
        dc      $1A1        'F' Data. TCB Reload Value = $A1
        dc      $1AB        'G' Data. TCB Reload Value = $AB
        dc      $1A1        'F' Data. TCB Reload Value = $A1
        dc      $19B        'E' Data. TCB Reload Value = $9B
        dc      $18E        'D' Data. TCB Reload Value = $8E
*
        dc      $19B        'E' Data. TCB Reload Value = $9B
        dc      $1A1        'F' Data. TCB Reload Value = $A1
        dc      $19B        'E' Data. TCB Reload Value = $9B
        dc      $18E        'D' Data. TCB Reload Value = $8E
        dc      $17F        'C' Data. TCB Reload Value = $7F
*
        dc      $18E        'D' Data. TCB Reload Value = $8E
        dc      $19B        'E' Data. TCB Reload Value = $9B
        dc      $18E        'D' Data. TCB Reload Value = $8E
        dc      $17F        'C' Data. TCB Reload Value = $7F
        dc      $17A        'B' Data. TCB Reload Value = $7A
*
        dc      $17F        'C' Data. TCB Reload Value = $7F
        dc      $1FF        ' ' Data. TCB Reload Value = $FF
*
        dc      $100        ' ' Data. TCB Reload Value = $00
*
```

```
************************************************************
*        Time Period Data
************************************************************
*
         org     $0300
*
*** 1st Cycle
*
         dc      $138        'G' Time Period Data. TCC Reload Value = $38
         dc      $19C        'C' Time Period Data. TCC Reload Value = $9C
         dc      $19C        'D' Time Period Data. TCC Reload Value = $9C
         dc      $19C        'E' Time Period Data. TCC Reload Value = $9C
         dc      $19C        'F' Time Period Data. TCC Reload Value = $9C
*
         dc      $138        'G' Time Period Data. TCC Reload Value = $38
         dc      $16A        'C' Time Period Data. TCC Reload Value = $6A
         dc      $1CE        ' ' Time Period Data. TCC Reload Value = $CE
         dc      $138        'C' Time Period Data. TCC Reload Value = $38
*
         dc      $138        'A' Time Period Data. TCC Reload Value = $38
         dc      $19C        'F' Time Period Data. TCC Reload Value = $9C
         dc      $19C        'G' Time Period Data. TCC Reload Value = $9C
         dc      $19C        'A' Time Period Data. TCC Reload Value = $9C
         dc      $19C        'B' Time Period Data. TCC Reload Value = $9C
*
         dc      $138        'C' Time Period Data. TCC Reload Value = $38
         dc      $16A        'C' Time Period Data. TCC Reload Value = $6A
         dc      $1CE        ' ' Time Period Data. TCC Reload Value = $CE
         dc      $138        'C' Time Period Data. TCC Reload Value = $38
*
         dc      $138        'F' Time Period Data. TCC Reload Value = $38
         dc      $19C        'G' Time Period Data. TCC Reload Value = $9C
         dc      $19C        'F' Time Period Data. TCC Reload Value = $9C
         dc      $19C        'E' Time Period Data. TCC Reload Value = $9C
         dc      $19C        'D' Time Period Data. TCC Reload Value = $9C
*
         dc      $138        'E' Time Period Data. TCC Reload Value = $38
         dc      $19C        'F' Time Period Data. TCC Reload Value = $9C
         dc      $19C        'E' Time Period Data. TCC Reload Value = $9C
         dc      $19C        'D' Time Period Data. TCC Reload Value = $9C
         dc      $19C        'C' Time Period Data. TCC Reload Value = $9C
*
         dc      $138        'B' Time Period Data. TCC Reload Value = $38
         dc      $19C        'C' Time Period Data. TCC Reload Value = $9C
         dc      $19C        'D' Time Period Data. TCC Reload Value = $9C
         dc      $19C        'E' Time Period Data. TCC Reload Value = $9C
         dc      $19C        'C' Time Period Data. TCC Reload Value = $9C
*
         dc      $138        'D' Time Period Data. TCC Reload Value = $38
*
*** 2nd Cycle
```

```
*
        dc      $138        'G' Time Period Data. TCC Reload Value = $38
        dc      $19C        'C' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'D' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'E' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'F' Time Period Data. TCC Reload Value = $9C
*
        dc      $138        'G' Time Period Data. TCC Reload Value = $38
        dc      $16A        'C' Time Period Data. TCC Reload Value = $6A
        dc      $1CE        ' ' Time Period Data. TCC Reload Value = $CE
        dc      $138        'C' Time Period Data. TCC Reload Value = $38
*
        dc      $138        'A' Time Period Data. TCC Reload Value = $38
        dc      $19C        'F' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'G' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'A' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'B' Time Period Data. TCC Reload Value = $9C
*
        dc      $138        'C' Time Period Data. TCC Reload Value = $38
        dc      $16A        'C' Time Period Data. TCC Reload Value = $6A
        dc      $1CE        ' ' Time Period Data. TCC Reload Value = $CE
        dc      $138        'C' Time Period Data. TCC Reload Value = $38
*
        dc      $138        'F' Time Period Data. TCC Reload Value = $38
        dc      $19C        'G' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'F' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'E' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'D' Time Period Data. TCC Reload Value = $9C
*
        dc      $138        'E' Time Period Data. TCC Reload Value = $38
        dc      $19C        'F' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'E' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'D' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'C' Time Period Data. TCC Reload Value = $9C
*
        dc      $138        'D' Time Period Data. TCC Reload Value = $38
        dc      $19C        'E' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'D' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'C' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'B' Time Period Data. TCC Reload Value = $9C
*
        dc      $138        'C' Time Period Data. TCC Reload Value = $38
        dc      $138        ' ' Time Period Data. TCC Reload Value = $38
*
************************************************************
*       Times Counter Data
************************************************************
*
        org     $0400
*
*** 1st Cycle
```

```
*
         dc      $10F       'G' Time Counter Data. PCNT = $F
         dc      $10F       'C' Time Counter Data. PCNT = $F
         dc      $10F       'D' Time Counter Data. PCNT = $F
         dc      $10F       'E' Time Counter Data. PCNT = $F
         dc      $10F       'F' Time Counter Data. PCNT = $F
*
         dc      $10F       'G' Time Counter Data. PCNT = $F
         dc      $10F       'C' Time Counter Data. PCNT = $F
         dc      $10F       ' ' Time Counter Data. PCNT = $F
         dc      $10F       'C' Time Counter Data. PCNT = $F
*
         dc      $10F       'A' Time Counter Data. PCNT = $F
         dc      $10F       'F' Time Counter Data. PCNT = $F
         dc      $10F       'G' Time Counter Data. PCNT = $F
         dc      $10F       'A' Time Counter Data. PCNT = $F
         dc      $10F       'B' Time Counter Data. PCNT = $F
*
         dc      $10F       'C' Time Counter Data. PCNT = $F
         dc      $10F       'C' Time Counter Data. PCNT = $F
         dc      $10F       ' ' Time Counter Data. PCNT = $F
         dc      $10F       'C' Time Counter Data. PCNT = $F
*
         dc      $10F       'F' Time Counter Data. PCNT = $F
         dc      $10F       'G' Time Counter Data. PCNT = $F
         dc      $10F       'F' Time Counter Data. PCNT = $F
         dc      $10F       'E' Time Counter Data. PCNT = $F
         dc      $10F       'D' Time Counter Data. PCNT = $F
*
         dc      $10F       'E' Time Counter Data. PCNT = $F
         dc      $10F       'F' Time Counter Data. PCNT = $F
         dc      $10F       'E' Time Counter Data. PCNT = $F
         dc      $10F       'D' Time Counter Data. PCNT = $F
         dc      $10F       'C' Time Counter Data. PCNT = $F
*
         dc      $10F       'B' Time Counter Data. PCNT = $F
         dc      $10F       'C' Time Counter Data. PCNT = $F
         dc      $10F       'D' Time Counter Data. PCNT = $F
         dc      $10F       'E' Time Counter Data. PCNT = $F
         dc      $10F       'C' Time Counter Data. PCNT = $F
*
         dc      $10D       'D' Time Counter Data. PCNT = $D
*
*** 2nd Cycle
*
         dc      $10F       'G' Time Counter Data. PCNT = $F
         dc      $10F       'C' Time Counter Data. PCNT = $F
         dc      $10F       'D' Time Counter Data. PCNT = $F
         dc      $10F       'E' Time Counter Data. PCNT = $F
         dc      $10F       'F' Time Counter Data. PCNT = $F
*
```

```
        dc      $10F        'G' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
        dc      $10F        ' ' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
*
        dc      $10F        'A' Time Counter Data. PCNT = $F
        dc      $10F        'F' Time Counter Data. PCNT = $F
        dc      $10F        'G' Time Counter Data. PCNT = $F
        dc      $10F        'A' Time Counter Data. PCNT = $F
        dc      $10F        'B' Time Counter Data. PCNT = $F
*
        dc      $10F        'C' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
        dc      $10F        ' ' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
*
        dc      $10F        'F' Time Counter Data. PCNT = $F
        dc      $10F        'G' Time Counter Data. PCNT = $F
        dc      $10F        'F' Time Counter Data. PCNT = $F
        dc      $10F        'E' Time Counter Data. PCNT = $F
        dc      $10F        'D' Time Counter Data. PCNT = $F
*
        dc      $10F        'E' Time Counter Data. PCNT = $F
        dc      $10F        'F' Time Counter Data. PCNT = $F
        dc      $10F        'E' Time Counter Data. PCNT = $F
        dc      $10F        'D' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
*
        dc      $10F        'D' Time Counter Data. PCNT = $F
        dc      $10F        'E' Time Counter Data. PCNT = $F
        dc      $10F        'D' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
        dc      $10F        'B' Time Counter Data. PCNT = $F
*
        dc      $10F        'F' Time Counter Data. PCNT = $F
        dc      $10F        'E' Time Counter Data. PCNT = $F
        dc      $10F        'D' Time Counter Data. PCNT = $F
*
        dc      $10F        'E' Time Counter Data. PCNT = $F
        dc      $10F        'F' Time Counter Data. PCNT = $F
        dc      $10F        'E' Time Counter Data. PCNT = $F
        dc      $10F        'D' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
*
        dc      $10F        'B' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
        dc      $10F        'D' Time Counter Data. PCNT = $F
        dc      $10F        'E' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
*
        dc      $10D        'D' Time Counter Data. PCNT = $D
```

```
*** 2nd Cycle
        dc      $10F        'G' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
        dc      $10F        'D' Time Counter Data. PCNT = $F
        dc      $10F        'E' Time Counter Data. PCNT = $F
        dc      $10F        'F' Time Counter Data. PCNT = $F
*
        dc      $10F        'G' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
        dc      $10F        ' ' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
*
        dc      $10F        'A' Time Counter Data. PCNT = $F
        dc      $10F        'F' Time Counter Data. PCNT = $F
        dc      $10F        'G' Time Counter Data. PCNT = $F
        dc      $10F        'A' Time Counter Data. PCNT = $F
        dc      $10F        'B' Time Counter Data. PCNT = $F
*
        dc      $10F        'C' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
        dc      $10F        ' ' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
*
        dc      $10F        'F' Time Counter Data. PCNT = $F
        dc      $10F        'G' Time Counter Data. PCNT = $F
        dc      $10F        'F' Time Counter Data. PCNT = $F
        dc      $10F        'E' Time Counter Data. PCNT = $F
        dc      $10F        'D' Time Counter Data. PCNT = $F
*
        dc      $10F        'E' Time Counter Data. PCNT = $F
        dc      $10F        'F' Time Counter Data. PCNT = $F
        dc      $10F        'E' Time Counter Data. PCNT = $F
        dc      $10F        'D' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
*
        dc      $10F        'D' Time Counter Data. PCNT = $F
        dc      $10F        'E' Time Counter Data. PCNT = $F
        dc      $10F        'D' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
        dc      $10F        'B' Time Counter Data. PCNT = $F
*

        dc      $10E        'C' Time Counter Data. PCNT = $E
        dc      $10F        ' ' Time Counter Data. PCNT = $F
        end
```

2. H4318/H4359

```
**************************************************************
*
*         HMCS400 Series Application Note
*
*         'Sound Play
*         - Minuet : J.S.Bach'
*
*         MCU : H4318/H4359
*
*         External Clock : 4MHz
*         Internal Clock : 1MHz
*         Sub Clock      : 32.768kHz
*
**************************************************************
*
**************************************************************
*         Symbol Definition
**************************************************************
*
IE        equ     0,$000     Interrupt Enable Flag
RSP       equ     1,$000     Reset Stack Pointer
IF0       equ     2,$000     INT0 Interrupt Request Flag
IM0       equ     3,$000     INT0 Interrupt Mask
*
IF1       equ     0,$001     INT1 Interrupt Request Flag
IM1       equ     1,$001     INT1 Interrupt Mask
IFTA      equ     2,$001     Timer A Interrupt Request Flag
IMTA      equ     3,$001     Timer A Interrupt Mask
*
IFTB      equ     0,$002     Timer B Interrupt Request Flag
IMTB      equ     1,$002     Timer B Interrupt Mask
IFTC      equ     2,$002     Timer C Interrupt Request Flag
IMTC      equ     3,$002     Timer C Interrupt Mask
*
IFAD      equ     0,$003     A/D Converter Interrupt Request Flag
IMAD      equ     1,$003     A/D Converter Interrupt Mask
IFS       equ     2,$003     Serial Interrupt Request Flag
IMS       equ     3,$003     Serial Interrupt Mask
*
PMRA      equ     $004       Port Mode Register
SMR       equ     $005       Serial Mode Register
SRL       equ     $006       Serial Data Register L
SRU       equ     $007       Serial Data Register U
TMA       equ     $008       Timer Mode Register A
TMB1      equ     $009       Timer Mode Register B1
TRBL      equ     $00A       Timer Read Register BL
TWBL      equ     $00A       Timer Write Register BL
TRBU      equ     $00B       Timer Read Register BU
TWBU      equ     $00B       Timer Write Register BU
```

```
MIS      equ     $00C       Miscellaneous Register
TMC      equ     $00D       Timer Mode Register C
TRCL     equ     $00E       Timer Read Register CL
TWCL     equ     $00E       Timer Write Register CL
TRCU     equ     $00F       Timer Read Register CU
TWCU     equ     $00F       Timer Write Register CU
ACR      equ     $016       A/D Channel Register
ADRL     equ     $017       A/D Data Register L
ADRU     equ     $018       A/D Data Register U
AMR1     equ     $019       A/D Mode Register 1
AMR2     equ     $01A       A/D Mode Register 2
*
WDON     equ     1,$020     Watchdog on Flag
ADSF     equ     2,$020     A/D Start Flag
*
ICSF     equ     0,$021     Input Capture Status Flag
ICEF     equ     1,$021     Input Capture Error Flag
IAOF     equ     2,$021     IAD off Flag
RAME     equ     3,$021     RAM Enable Flag
*
PMRB     equ     $024       Port Mode Register B
PMRC     equ     $025       Port Mode Register C
TMB2     equ     $026       Timer Mode Register B2
DCD0     equ     $02C       D Port Data Control Register 0
DCD1     equ     $02D       D Port Data Control Register 1
DCD2     equ     $02E       D Port Data Control Register 2
DCR0     equ     $030       R Port Data Control Register 0
DCR1     equ     $031       R Port Data Control Register 1
DCR2     equ     $032       R Port Data Control Register 2
DCR3     equ     $033       R Port Data Control Register 3
DCR4     equ     $034       R Port Data Control Register 4
DCR8     equ     $038       R Port Data Control Register 8
*
*************************************************************
*        RAM ALLOCATION
*************************************************************
*
AESC     equ     $040       Accumulator Escape
BESC     equ     $041       B Register Escape
*
PLCNT    equ     $090       Pulse Counter
*
PLFLG1   equ     $091       Pulse Flag 1
PLONF    equ     0,PLFLG1   Pulse Output Enable Flag
*
CNTL     equ     $093       Lower Counter
CNTU     equ     $094       Upper Counter
PCNT     equ     $095       Period Counter
AWORK    equ     $096       Accumulator Work RAM Area
BWORK    equ     $097       B Register Work RAM Area
*
```

```
***********************************************************
*        Vector Address
***********************************************************
*
         org     $0000
*
         JMPL    SPLMN      Reset Interrupt
         JMPL    SPLMN      INT0 Interrupt
         JMPL    SPLMN      INT1 Interrupt
         JMPL    SPLMN      Timer A Interrupt
         JMPL    SPLINT     Timer B Interrupt
         JMPL    SPLMN      Timer C Interrupt
         JMPL    SPLMN      A/D Converter Interrupt
         JMPL    SPLMN      Serial Interrupt
*
***********************************************************
*        Main Program
***********************************************************
*
         org     $1000
*
SPLMN    REMD    RSP        Reset Stack Pointer
*
         LAI     0
         LRA     0          Initialize R0 Port PDR
         LMID    8,DCR0     Initialize R03 Output Terminal Function
         LMID    0,PMRA     Initialize R03 Input/Output Terminal Function
*
         LMID    3,PLCNT    Initialize Pulse Counter
         LMID    0,CNTL     Initialize Lower Counter
         LMID    0,CNTU     Initialize Upper Counter
         LMID    $F,PCNT    Initialize Period Counter
         SEMD    PLONF      Initialize Pulse Output Enable Flag
*
         LMID    8,TMC      Initialize Timer C Function & Input Clock Period
         REMD    IFTC       Clear IFTC to 0
*
         LMID    $D,TMB1    Initialize Timer B Function & Input Clock Period
         LMID    0,TMB2     Initialize Timer B Function
*
         SEMD    IMTB       Timer B Interrupt Disable
         REMD    IFTB       Clear IFTB to 0
*
         SEMD    IE         All Interrupt Enable
*
SPLMN00  TMD     IFTC       IFTC = "1" ?
         BRS     SPLMN10    Yes. Branch to SPLMN10
         BRS     SPLMN00    No. Branch to SPLMN00
*
         REMD    IFTC       Clear IFTC to 0
         LMID    9,TWCL     Set TCC Reload Value Lower
```

```
         LMID    0,TWCU     Set TCC Reload Value Upper
*
SPLMN05  TMD     IFTC       IFTC = "1" ?
         BRS     SPLMN10    Yes. Branch to SPLMN10
         SEMD    IMTB       Timer B Interrupt Disable
         REMD    IFTB       Clear IFTB to 0
         LAI     0
         LRA     0          Sound off
         BRS     SPLMN05    Branch to SPLMN05
*
SPLMN10  REMD    IFTC       Clear IFTC to 0
         LAMD    PCNT       Load PCNT
         AI      1          Increment PCNT
         BRS     SPLMN40    PCNT Overflow ? Yes. Branch to SPLMN40
         LMAD    PCNT       No. Save PCNT
         BRS     SPLMN00    Branch to SPLMN00
*
SPLMN40  LAMD    CNTU       Load CNTL
         LBA
         LAMD    CNTL       Load CNTU
         P       2          Scale Data Pattern Generation
         LMAD    AWORK      Save Scale Lower Data
         LAB
         LMAD    BWORK      Save Scale Upper Data
         ALEI    0          Scale Upper Data <= $0 ? End Sound Play ?
         BRS     SPLMN90    Yes. Branch to SPLMN90
         ALEI    $E         Scale Upper Data <= $E ?
         BRS     SPLMN20    Yes. Branch to SPLMN20
         SEMD    IMTB       Timer B Interrupt Disable
         REMD    IFTB       Clear IFTB to 0
         LAI     0
         LRA     0          R03 Output Terminal is "Low" Output
         BRS     SPLMN30    Branch to SPLMN30
*
SPLMN20  LAMD    AWORK      Load AWORK
         LMAD    TWBL       Set TCB Reload Value Lower to Scale Data
         LAMD    BWORK      Load BWORK
         LMAD    TWBU       Set TCB Reload Value Upper to Scale Data
         REMD    IFTB       Clear IFTB to 0
         REMD    IMTB       Timer B Interrupt Enable
*
SPLMN30  LAMD    CNTU       Load Upper Counter
         LBA
         LAMD    CNTL       Load Lower Counter
         P       3          Time Period Data Pattern Generation
         LMAD    TWCL       Set TCC Reload Value Lower to Time Period Lower Data
         LAB
         LMAD    TWCU       Set TCC Reload Value Upper to Time Period Upper Data
*
         LAMD    CNTU       Load Upper Counter
         LBA
```

```
          LAMD    CNTL       Load Lower Counter
          P       4          Time Counter Data Pattern Generation
          LMAD    PCNT       Set PCNT to Time Counter Data
*
          SEC                Set Carry Flag at 1
          LAI     0
          AMCD    CNTL       Increment Lower Counter
          LMAD    CNTL       Save Lower Counter
          LAI     0
          AMCD    CNTU       If CNTL Overflow, Increment Upper Counter
          LMAD    CNTU       Save Upper Counter
          BR      *+1
          BRS     SPLMN00    Branch to SPLMN00
*
SPLMN90   SEMD    IMTB       Timer B Interrupt Disable
          LMID    0,CNTL     Initialize Lower Counter
          LMID    0,CNTU     Initialize Upper Counter
          LAI     0
          LRA     0          Initialize R0 Port PDR
          LMID    0,TWCL     Initialize TCC Reload Value Lower
          LMID    8,TWCU     Initialize TCC Reload Value Upper
          REMD    IFTC       Clear IFTC to 0
          BRS     SPLMN00    Branch to SPLMN00
*
************************************************************
*         Timer B Interrupt Process
*         ---Pulse Output Routine---
************************************************************
*
SPLINT    REMD    IFTB       Clear IFTB to 0
*
          LMAD    AESC       Store Accumulator
          LAB
          LMAD    BESC       Store B Register
*
          TMD     PLONF      Pulse Enable Flag = "1" ?
          BRS     PLI00      Yes. Branch to PLI00
          LAI     0          No. Stop Output Pulse
          LRA     0          R03 Output Terminal is "Low" Output
          BRS     PLI99      Branch to PLI99
*
PLI00     LAMD    PLCNT      Load Pulse Counter Value
          LBA                Store Pulse Counter Value
          ALEI    1          PLCNT <= $1 ? Is R03 Output Terminal "High" Output ?
          BRS     PLI10      Yes. Branch to PLI10
          LAI     0          No. R03 Output Terminal is "Low" Output
          BRS     PLI20      Branch to PLI20
PLI10     LAI     8
PLI20     LRA     0          R03 Output Terminal is "High" Output
          LAB                Restore Pulse Counter Value
          AI      $F         PLCNT + $F <= $F ? Pulse Counter = $0 ?
```

```
        BRS     PLI30       No. Branch to PLI30
        LAI     3           Yes. Pulse Counter Initialize
PLI30   LMAD    PLCNT       Save Pulse Counter Value
*
PLI99   LAMD    BESC        Restore B Register
        LBA
        LAMD    AESC        Restore Accumulator
*
        RTNI                Return from Interrupt
*
************************************************************
*       Scale Data
************************************************************
*
        ORG     $0200
*
*** 1st Cycle
*
        dc      $1AB        'G' Data. TCB Reload Value = $AB
        dc      $17F        'C' Data. TCB Reload Value = $7F
        dc      $18E        'D' Data. TCB Reload Value = $8E
        dc      $19B        'E' Data. TCB Reload Value = $9B
        dc      $1A1        'F' Data. TCB Reload Value = $A1
*
        dc      $1AB        'G' Data. TCB Reload Value = $AB
        dc      $17F        'C' Data. TCB Reload Value = $7F
        dc      $1FF        ' ' Data. TCB Reload Value = $FF
        dc      $17F        'C' Data. TCB Reload Value = $7F
*
        dc      $1B4        'A' Data. TCB Reload Value = $B4
        dc      $1A1        'F' Data. TCB Reload Value = $A1
        dc      $1AB        'G' Data. TCB Reload Value = $AB
        dc      $1B4        'A' Data. TCB Reload Value = $B4
        dc      $1BD        'B' Data. TCB Reload Value = $BD
*
        dc      $1C1        'C' Data. TCB Reload Value = $C1
        dc      $17F        'C' Data. TCB Reload Value = $7F
        dc      $1FF        ' ' Data. TCB Reload Value = $FF
        dc      $17F        'C' Data. TCB Reload Value = $7F
*
        dc      $1A1        'F' Data. TCB Reload Value = $A1
        dc      $1AB        'G' Data. TCB Reload Value = $AB
        dc      $1A1        'F' Data. TCB Reload Value = $A1
        dc      $19B        'E' Data. TCB Reload Value = $9B
        dc      $18E        'D' Data. TCB Reload Value = $8E
*
        dc      $19B        'E' Data. TCB Reload Value = $9B
        dc      $1A1        'F' Data. TCB Reload Value = $A1
        dc      $19B        'E' Data. TCB Reload Value = $9B
        dc      $18E        'D' Data. TCB Reload Value = $8E
        dc      $17F        'C' Data. TCB Reload Value = $7F
```

```
*
        dc      $17A      'B' Data. TCB Reload Value = $7A
        dc      $17F      'C' Data. TCB Reload Value = $7F
        dc      $18E      'D' Data. TCB Reload Value = $8E
        dc      $19B      'E' Data. TCB Reload Value = $9B
        dc      $17F      'C' Data. TCB Reload Value = $7F
*
        dc      $18E      'D' Data. TCB Reload Value = $8E
*
*** 2nd Cycle
*
        dc      $1AB      'G' Data. TCB Reload Value = $AB
        dc      $17F      'C' Data. TCB Reload Value = $7F
        dc      $18E      'D' Data. TCB Reload Value = $8E
        dc      $19B      'E' Data. TCB Reload Value = $9B
        dc      $1A1      'F' Data. TCB Reload Value = $A1
*
        dc      $1AB      'G' Data. TCB Reload Value = $AB
        dc      $17F      'C' Data. TCB Reload Value = $7F
        dc      $1FF      ' ' Data. TCB Reload Value = $FF
        dc      $17F      'C' Data. TCB Reload Value = $7F
*
        dc      $1B4      'A' Data. TCB Reload Value = $B4
        dc      $1A1      'F' Data. TCB Reload Value = $A1
        dc      $1AB      'G' Data. TCB Reload Value = $AB
        dc      $1B4      'A' Data. TCB Reload Value = $B4
        dc      $1BD      'B' Data. TCB Reload Value = $BD
*
        dc      $1C1      'C' Data. TCB Reload Value = $C1
        dc      $17F      'C' Data. TCB Reload Value = $7F
        dc      $1FF      ' ' Data. TCB Reload Value = $FF
        dc      $17F      'C' Data. TCB Reload Value = $7F
*
        dc      $1A1      'F' Data. TCB Reload Value = $A1
        dc      $1AB      'G' Data. TCB Reload Value = $AB
        dc      $1A1      'F' Data. TCB Reload Value = $A1
        dc      $19B      'E' Data. TCB Reload Value = $9B
        dc      $18E      'D' Data. TCB Reload Value = $8E
*
        dc      $19B      'E' Data. TCB Reload Value = $9B
        dc      $1A1      'F' Data. TCB Reload Value = $A1
        dc      $19B      'E' Data. TCB Reload Value = $9B
        dc      $18E      'D' Data. TCB Reload Value = $8E
        dc      $17F      'C' Data. TCB Reload Value = $7F
*
        dc      $18E      'D' Data. TCB Reload Value = $8E
        dc      $19B      'E' Data. TCB Reload Value = $9B
        dc      $18E      'D' Data. TCB Reload Value = $8E
        dc      $17F      'C' Data. TCB Reload Value = $7F
        dc      $17A      'B' Data. TCB Reload Value = $7A
*
```

```
        dc      $17F        'C' Data. TCB Reload Value = $7F
        dc      $1FF        ' ' Data. TCB Reload Value = $FF
*
        dc      $100        ' ' Data. TCB Reload Value = $00
*
************************************************************
*       Time Period Data
************************************************************
*
        org     $0300
*
*** 1st Cycle
*
        dc      $138        'G' Time Period Data. TCC Reload Value = $38
        dc      $19C        'C' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'D' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'E' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'F' Time Period Data. TCC Reload Value = $9C
*
        dc      $138        'G' Time Period Data. TCC Reload Value = $38
        dc      $16A        'C' Time Period Data. TCC Reload Value = $6A
        dc      $1CE        ' ' Time Period Data. TCC Reload Value = $CE
        dc      $138        'C' Time Period Data. TCC Reload Value = $38
*
        dc      $138        'A' Time Period Data. TCC Reload Value = $38
        dc      $19C        'F' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'G' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'A' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'B' Time Period Data. TCC Reload Value = $9C
*
        dc      $138        'C' Time Period Data. TCC Reload Value = $38
        dc      $16A        'C' Time Period Data. TCC Reload Value = $6A
        dc      $1CE        ' ' Time Period Data. TCC Reload Value = $CE
        dc      $138        'C' Time Period Data. TCC Reload Value = $38
*
        dc      $138        'F' Time Period Data. TCC Reload Value = $38
        dc      $19C        'G' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'F' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'E' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'D' Time Period Data. TCC Reload Value = $9C
*
        dc      $138        'E' Time Period Data. TCC Reload Value = $38
        dc      $19C        'F' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'E' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'D' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'C' Time Period Data. TCC Reload Value = $9C
*
        dc      $138        'B' Time Period Data. TCC Reload Value = $38
        dc      $19C        'C' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'D' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'E' Time Period Data. TCC Reload Value = $9C
```

```
        dc      $19C      'C' Time Period Data. TCC Reload Value = $9C
*
        dc      $138      'D' Time Period Data. TCC Reload Value = $38
*
*** 2nd Cycle
*
        dc      $138      'G' Time Period Data. TCC Reload Value = $38
        dc      $19C      'C' Time Period Data. TCC Reload Value = $9C
        dc      $19C      'D' Time Period Data. TCC Reload Value = $9C
        dc      $19C      'E' Time Period Data. TCC Reload Value = $9C
        dc      $19C      'F' Time Period Data. TCC Reload Value = $9C
*
        dc      $138      'G' Time Period Data. TCC Reload Value = $38
        dc      $16A      'C' Time Period Data. TCC Reload Value = $6A
        dc      $1CE      ' ' Time Period Data. TCC Reload Value = $CE
        dc      $138      'C' Time Period Data. TCC Reload Value = $38
*
        dc      $138      'A' Time Period Data. TCC Reload Value = $38
        dc      $19C      'F' Time Period Data. TCC Reload Value = $9C
        dc      $19C      'G' Time Period Data. TCC Reload Value = $9C
        dc      $19C      'A' Time Period Data. TCC Reload Value = $9C
        dc      $19C      'B' Time Period Data. TCC Reload Value = $9C
*
        dc      $138      'C' Time Period Data. TCC Reload Value = $38
        dc      $16A      'C' Time Period Data. TCC Reload Value = $6A
        dc      $1CE      ' ' Time Period Data. TCC Reload Value = $CE
        dc      $138      'C' Time Period Data. TCC Reload Value = $38
*
        dc      $138      'F' Time Period Data. TCC Reload Value = $38
        dc      $19C      'G' Time Period Data. TCC Reload Value = $9C
        dc      $19C      'F' Time Period Data. TCC Reload Value = $9C
        dc      $19C      'E' Time Period Data. TCC Reload Value = $9C
        dc      $19C      'D' Time Period Data. TCC Reload Value = $9C
*
        dc      $138      'E' Time Period Data. TCC Reload Value = $38
        dc      $19C      'F' Time Period Data. TCC Reload Value = $9C
        dc      $19C      'E' Time Period Data. TCC Reload Value = $9C
        dc      $19C      'D' Time Period Data. TCC Reload Value = $9C
        dc      $19C      'C' Time Period Data. TCC Reload Value = $9C
*
        dc      $138      'D' Time Period Data. TCC Reload Value = $38
        dc      $19C      'E' Time Period Data. TCC Reload Value = $9C
        dc      $19C      'D' Time Period Data. TCC Reload Value = $9C
        dc      $19C      'C' Time Period Data. TCC Reload Value = $9C
        dc      $19C      'B' Time Period Data. TCC Reload Value = $9C
*
        dc      $138      'C' Time Period Data. TCC Reload Value = $38
        dc      $138      ' ' Time Period Data. TCC Reload Value = $38
*
************************************************************
*       Time Counter Data
```

```
************************************************************
*
        org     $0400
*
*** 1st Cycle
*
        dc      $10F      'G' Time Counter Data. PCNT = $F
        dc      $10F      'C' Time Counter Data. PCNT = $F
        dc      $10F      'D' Time Counter Data. PCNT = $F
        dc      $10F      'E' Time Counter Data. PCNT = $F
        dc      $10F      'F' Time Counter Data. PCNT = $F
*
        dc      $10F      'G' Time Counter Data. PCNT = $F
        dc      $10F      'C' Time Counter Data. PCNT = $F
        dc      $10F      ' ' Time Counter Data. PCNT = $F
        dc      $10F      'C' Time Counter Data. PCNT = $F
*
        dc      $10F      'A' Time Counter Data. PCNT = $F
        dc      $10F      'F' Time Counter Data. PCNT = $F
        dc      $10F      'G' Time Counter Data. PCNT = $F
        dc      $10F      'A' Time Counter Data. PCNT = $F
        dc      $10F      'B' Time Counter Data. PCNT = $F
*
        dc      $10F      'C' Time Counter Data. PCNT = $F
        dc      $10F      'C' Time Counter Data. PCNT = $F
        dc      $10F      ' ' Time Counter Data. PCNT = $F
        dc      $10F      'C' Time Counter Data. PCNT = $F
*
        dc      $10F      'F' Time Counter Data. PCNT = $F
        dc      $10F      'G' Time Counter Data. PCNT = $F
        dc      $10F      'F' Time Counter Data. PCNT = $F
        dc      $10F      'E' Time Counter Data. PCNT = $F
        dc      $10F      'D' Time Counter Data. PCNT = $F
*
        dc      $10F      'E' Time Counter Data. PCNT = $F
        dc      $10F      'F' Time Counter Data. PCNT = $F
        dc      $10F      'E' Time Counter Data. PCNT = $F
        dc      $10F      'D' Time Counter Data. PCNT = $F
        dc      $10F      'C' Time Counter Data. PCNT = $F
*
        dc      $10F      'B' Time Counter Data. PCNT = $F
        dc      $10F      'C' Time Counter Data. PCNT = $F
        dc      $10F      'D' Time Counter Data. PCNT = $F
        dc      $10F      'E' Time Counter Data. PCNT = $F
        dc      $10F      'C' Time Counter Data. PCNT = $F
*
        dc      $10D      'D' Time Counter Data. PCNT = $D
*
*** 2nd Cycle
*
        dc      $10F      'G' Time Counter Data. PCNT = $F
```

```
        dc      $10F        'C' Time Counter Data. PCNT = $F
        dc      $10F        'D' Time Counter Data. PCNT = $F
        dc      $10F        'E' Time Counter Data. PCNT = $F
        dc      $10F        'F' Time Counter Data. PCNT = $F
*
        dc      $10F        'G' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
        dc      $10F        ' ' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
*
        dc      $10F        'A' Time Counter Data. PCNT = $F
        dc      $10F        'F' Time Counter Data. PCNT = $F
        dc      $10F        'G' Time Counter Data. PCNT = $F
        dc      $10F        'A' Time Counter Data. PCNT = $F
        dc      $10F        'B' Time Counter Data. PCNT = $F
*
        dc      $10F        'C' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
        dc      $10F        ' ' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
*
        dc      $10F        'F' Time Counter Data. PCNT = $F
        dc      $10F        'G' Time Counter Data. PCNT = $F
        dc      $10F        'F' Time Counter Data. PCNT = $F
        dc      $10F        'E' Time Counter Data. PCNT = $F
        dc      $10F        'D' Time Counter Data. PCNT = $F
*
        dc      $10F        'E' Time Counter Data. PCNT = $F
        dc      $10F        'F' Time Counter Data. PCNT = $F
        dc      $10F        'E' Time Counter Data. PCNT = $F
        dc      $10F        'D' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
*
        dc      $10F        'D' Time Counter Data. PCNT = $F
        dc      $10F        'E' Time Counter Data. PCNT = $F
        dc      $10F        'D' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
        dc      $10F        'B' Time Counter Data. PCNT = $F
*
        dc      $10E        'C' Time Counter Data. PCNT = $E
        dc      $10F        ' ' Time Counter Data. PCNT = $F
*
        end
```

3. H4369

```
**************************************************************
*
*         HMCS400 Series Application Note
*
*         'Sound Play
*         - Minuet : J.S.Bach'
*
*         MCU : H4369
*
*         External Clock : 4MHz
*         Internal Clock : 1MHz
*         Sub Clock      : 32.768kHz
*
**************************************************************
*
**************************************************************
*         Symbol Definition
**************************************************************
*
IE        equ     0,$000     Interrupt Enable Flag
RSP       equ     1,$000     Reset Stack Pointer
IF0       equ     2,$000     INT0 Interrupt Request Flag
IM0       equ     3,$000     INT0 Interrupt Mask
*
IF1       equ     0,$001     INT1 Interrupt Request Flag
IM1       equ     1,$001     INT1 Interrupt Mask
IFTA      equ     2,$001     Timer A Interrupt Request Flag
IMTA      equ     3,$001     Timer A Interrupt Mask
*
IFTB      equ     0,$002     Timer B Interrupt Request Flag
IMTB      equ     1,$002     Timer B Interrupt Mask
IFTC      equ     2,$002     Timer C Interrupt Request Flag
IMTC      equ     3,$002     Timer C Interrupt Mask
*
IFAD      equ     0,$003     A/D Converter Interrupt Request Flag
IMAD      equ     1,$003     A/D Converter Interrupt Mask
IFS       equ     2,$003     Serial Interrupt Request Flag
IMS       equ     3,$003     Serial Interrupt Mask
*
PMRA      equ     $004       Port Mode Register
SMR       equ     $005       Serial Mode Register
SRL       equ     $006       Serial Data Register L
SRU       equ     $007       Serial Data Register U
TMA       equ     $008       Timer Mode Register A
TMB1      equ     $009       Timer Mode Register B1
TRBL      equ     $00A       Timer Read Register BL
TWBL      equ     $00A       Timer Write Register BL
TRBU      equ     $00B       Timer Read Register BU
TWBU      equ     $00B       Timer Write Register BU
```

```
MIS      equ     $00C       Miscellaneous Register
TMC      equ     $00D       Timer Mode Register C
TRCL     equ     $00E       Timer Read Register CL
TWCL     equ     $00E       Timer Write Register CL
TRCU     equ     $00F       Timer Read Register CU
TWCU     equ     $00F       Timer Write Register CU
*
ACR      equ     $016       A/D Channel Register
ADRL     equ     $017       A/D Data Register L
ADRU     equ     $018       A/D Data Register U
AMR1     equ     $019       A/D Mode Register 1
AMR2     equ     $01A       A/D Mode Register 2
*
LSON     equ     0,$020     Low Speed on Flag
WDON     equ     1,$020     Watchdog on Flag
ADSF     equ     2,$020     A/D Start Flag
DTON     equ     3,$020     Direct Transfer on Flag
*
ICSF     equ     0,$021     Input Capture Status Flag
ICEF     equ     1,$021     Input Capture Error Flag
IAOF     equ     2,$021     IAD off Flag
RAME     equ     3,$021     RAM Enable Flag
*
PMRB     equ     $024       Port Mode Register B
PMRC     equ     $025       Port Mode Register C
TMB2     equ     $026       Timer Mode Register B2
SSR1     equ     $027       System Clock Selection Register 1
SSR2     equ     $028       System Clock Selection Register 2
*
DCD0     equ     $02C       D Port Data Control Register 0
DCD1     equ     $02D       D Port Data Control Register 1
DCD2     equ     $02E       D Port Data Control Register 2
DCD3     equ     $02F       D Port Data Control Register 3
DCR0     equ     $030       R Port Data Control Register 0
DCR1     equ     $031       R Port Data Control Register 1
DCR2     equ     $032       R Port Data Control Register 2
DCR3     equ     $033       R Port Data Control Register 3
DCR4     equ     $034       R Port Data Control Register 4
DCR5     equ     $035       R Port Data Control Register 5
DCR6     equ     $036       R Port Data Control Register 6
SCR7     equ     $037       R Port Data Control Register 7
DCR8     equ     $038       R Port Data Control Register 8
*
************************************************************
*        RAM ALLOCATION
************************************************************
*
AESC     equ     $040       Accumulator Escape
BESC     equ     $041       B Register Escape
*
PLCNT    equ     $090       Pulse Counter
```

```
*
PLFLG1   equ    $091       Pulse Flag 1
PLONF    equ    0,PLFLG1   Pulse Output Enable Flag
*
CNTL     equ    $093       Lower Counter
CNTU     equ    $094       Upper Counter
PCNT     equ    $095       Period Counter
AWORK    equ    $096       Accumulator Work RAM Area
BWORK    equ    $097       B Register Work RAM Area
*
************************************************************
*        Vector Address
************************************************************
*
         org    $0000
*
         JMPL   SPLMN      Reset Interrupt
         JMPL   SPLMN      INT0 Interrupt
         JMPL   SPLMN      INT1 Interrupt
         JMPL   SPLMN      Timer A Interrupt
         JMPL   SPLINT     Timer B Interrupt
         JMPL   SPLMN      Timer C Interrupt
         JMPL   SPLMN      A/D Converter Interrupt
         JMPL   SPLMN      Serial Interrupt
*
************************************************************
*        Main Program
************************************************************
*
         org    $1000
*
SPLMN    REMD   RSP        Reset Stack Pointer
         LMID   $2,SSR1    Set System Clock to 1.6 - 5.0MHz
*
         LAI    0
         LRA    0          Initialize R0 Port PDR
         LMID   8,DCR0     Initialize R03 Output Terminal Function
         LMID   0,PMRA     Initialize R03 Input/Output Terminal Function
*
         LMID   3,PLCNT    Initialize Pulse Counter
         LMID   0,CNTL     Initialize Lower Counter
         LMID   0,CNTU     Initialize Upper Counter
         LMID   $F,PCNT    Initialize Period Counter
         SEMD   PLONF      Initialize Pulse Output Enable Flag
*
         LMID   8,TMC      Initialize Timer C Function & Input Clock Period
         REMD   IFTC       Clear IFTC to 0
*
         LMID   $D,TMB1    Initialize Timer B Function & Input Clock Period
         LMID   0,TMB2     Initialize Timer B Function
*
```

```
         SEMD    IMTB       Timer B Interrupt Disable
         REMD    IFTB       Clear IFTB to 0
*
         SEMD    IE         All Interrupt Enable
*
SPLMN00  TMD     IFTC       IFTC = "1" ?
         BRS     SPLMN10    Yes. Branch to SPLMN10
         BRS     SPLMN00    No. Branch to SPLMN00
*
         REMD    IFTC       Clear IFTC to 0
         LMID    9,TWCL     Set TCC Reload Value Lower
         LMID    0,TWCU     Set TCC Reload Value Upper
*
SPLMN05  TMD     IFTC       IFTC = "1" ?
         BRS     SPLMN10    Yes. Branch to SPLMN10
         SEMD    IMTB       Timer B Interrupt Disable
         REMD    IFTB       Clear IFTB to 0
         LAI     0
         LRA     0          Sound off
         BRS     SPLMN05    Branch to SPLMN05
*
SPLMN10  REMD    IFTC       Clear IFTC to 0
         LAMD    PCNT       Load PCNT
         AI      1          Increment PCNT
         BRS     SPLMN40    PCNT Overflow ? Yes. Branch to SPLMN40
         LMAD    PCNT       No. Save PCNT
         BRS     SPLMN00    Branch to SPLMN00
*
SPLMN40  LAMD    CNTU       Load CNTL
         LBA
         LAMD    CNTL       Load CNTU
         P       2          Scale Data Pattern Generation
         LMAD    AWORK      Save Scale Lower Data
         LAB
         LMAD    BWORK      Save Scale Upper Data
         ALEI    0          Scale Upper Data <= $0 ? End Sound Play ?
         BRS     SPLMN90    Yes. Branch to SPLMN90
         ALEI    $E         Scale Upper Data <= $E ?
         BRS     SPLMN20    Yes. Branch to SPLMN20
         SEMD    IMTB       Timer B Interrupt Disable
         REMD    IFTB       Clear IFTB to 0
         LAI     0
         LRA     0          R03 Output Terminal is "Low" Output
         BRS     SPLMN30    Branch to SPLMN30
*
SPLMN20  LAMD    AWORK      Load AWORK
         LMAD    TWBL       Set TCB Reload Value Lower to Scale Data
         LAMD    BWORK      Load BWORK
         LMAD    TWBU       Set TCB Reload Value Upper to Scale Data
         REMD    IFTB       Clear IFTB to 0
         REMD    IMTB       Timer B Interrupt Enable
```


RENESAS

```
*
SPLMN30  LAMD    CNTU       Load Upper Counter
         LBA
         LAMD    CNTL       Load Lower Counter
         P       3          Time Period Data Pattern Generation
         LMAD    TWCL       Set TCC Reload Value Lower to Time Period Lower Data
         LAB
         LMAD    TWCU       Set TCC Reload Value Upper to Time Period Upper Data
*
         LAMD    CNTU       Load Upper Counter
         LBA
         LAMD    CNTL       Load Lower Counter
         P       4          Time Counter Data Pattern Generation
         LMAD    PCNT       Set PCNT to Time Counter Data
*
         SEC                Set Carry Flag at 1
         LAI     0
         AMCD    CNTL       Increment Lower Counter
         LMAD    CNTL       Save Lower Counter
         LAI     0
         AMCD    CNTU       If CNTL Overflow, Increment Upper Counter
         LMAD    CNTU       Save Upper Counter
         BR      *+1
         BRS     SPLMN00    Branch to SPLMN00
*
SPLMN90  SEMD    IMTB       Timer B Interrupt Disable
         LMID    0,CNTL     Initialize Lower Counter
         LMID    0,CNTU     Initialize Upper Counter
         LAI     0
         LRA     0          Initialize R0 Port PDR
         LMID    0,TWCL     Initialize TCC Reload Value Lower
         LMID    8,TWCU     Initialize TCC Reload Value Upper
         REMD    IFTC       Clear IFTC to 0
         BRS     SPLMN00    Branch to SPLMN00
*
************************************************************
*        Timer B Interrupt Process
*        ---Pulse Output Routine---
************************************************************
*
SPLINT   REMD    IFTB       Clear IFTB to 0
*
         LMAD    AESC       Store Accumulator
         LAB
         LMAD    BESC       Store B Register
*
         TMD     PLONF      Pulse Enable Flag = "1" ?
         BRS     PLI00      Yes. Branch to PLI00
         LAI     0          No. Stop Output Pulse
         LRA     0          R03 Output Terminal is "Low" Output
         BRS     PLI99      Branch to PLI99
```

```
*
PLI00    LAMD    PLCNT      Load Pulse Counter Value
         LBA                Store Pulse Counter Value
         ALEI    1          PLCNT <= $1 ? Is R03 Output Terminal "High" Output ?
         BRS     PLI10      Yes. Branch to PLI10
         LAI     0          No. R03 Output Terminal is "Low" Output
         BRS     PLI20      Branch to PLI20
PLI10    LAI     8
PLI20    LRA     0          R03 Output Terminal is "High" Output
         LAB                Restore Pulse Counter Value
         AI      $F         PLCNT + $F <= $F ? Pulse Counter = $0 ?
         BRS     PLI30      No. Branch to PLI30
         LAI     3          Yes. Pulse Counter Initialize
PLI30    LMAD    PLCNT      Save Pulse Counter Value
*
PLI99    LAMD    BESC       Restore B Register
         LBA
         LAMD    AESC       Restore Accumulator
*
         RTNI               Return from Interrupt
*
************************************************************
*        Scale Data
************************************************************
*
         ORG     $0200
*
*** 1st Cycle
*
         dc      $1AB       'G' Data. TCB Reload Value = $AB
         dc      $17F       'C' Data. TCB Reload Value = $7F
         dc      $18E       'D' Data. TCB Reload Value = $8E
         dc      $19B       'E' Data. TCB Reload Value = $9B
         dc      $1A1       'F' Data. TCB Reload Value = $A1
*
         dc      $1AB       'G' Data. TCB Reload Value = $AB
         dc      $17F       'C' Data. TCB Reload Value = $7F
         dc      $1FF       ' ' Data. TCB Reload Value = $FF
         dc      $17F       'C' Data. TCB Reload Value = $7F
*
         dc      $1B4       'A' Data. TCB Reload Value = $B4
         dc      $1A1       'F' Data. TCB Reload Value = $A1
         dc      $1AB       'G' Data. TCB Reload Value = $AB
         dc      $1B4       'A' Data. TCB Reload Value = $B4
         dc      $1BD       'B' Data. TCB Reload Value = $BD
*
         dc      $1C1       'C' Data. TCB Reload Value = $C1
         dc      $17F       'C' Data. TCB Reload Value = $7F
         dc      $1FF       ' ' Data. TCB Reload Value = $FF
         dc      $17F       'C' Data. TCB Reload Value = $7F
*
```

```
          dc      $1A1        'F' Data. TCB Reload Value = $A1
          dc      $1AB        'G' Data. TCB Reload Value = $AB
          dc      $1A1        'F' Data. TCB Reload Value = $A1
          dc      $19B        'E' Data. TCB Reload Value = $9B
          dc      $18E        'D' Data. TCB Reload Value = $8E
*
          dc      $19B        'E' Data. TCB Reload Value = $9B
          dc      $1A1        'F' Data. TCB Reload Value = $A1
          dc      $19B        'E' Data. TCB Reload Value = $9B
          dc      $18E        'D' Data. TCB Reload Value = $8E
          dc      $17F        'C' Data. TCB Reload Value = $7F
*
          dc      $17A        'B' Data. TCB Reload Value = $7A
          dc      $17F        'C' Data. TCB Reload Value = $7F
          dc      $18E        'D' Data. TCB Reload Value = $8E
          dc      $19B        'E' Data. TCB Reload Value = $9B
          dc      $17F        'C' Data. TCB Reload Value = $7F
*
          dc      $18E        'D' Data. TCB Reload Value = $8E
*
*** 2nd Cycle
*
          dc      $1AB        'G' Data. TCB Reload Value = $AB
          dc      $17F        'C' Data. TCB Reload Value = $7F
          dc      $18E        'D' Data. TCB Reload Value = $8E
          dc      $19B        'E' Data. TCB Reload Value = $9B
          dc      $1A1        'F' Data. TCB Reload Value = $A1
*
          dc      $1AB        'G' Data. TCB Reload Value = $AB
          dc      $17F        'C' Data. TCB Reload Value = $7F
          dc      $1FF        ' ' Data. TCB Reload Value = $FF
          dc      $17F        'C' Data. TCB Reload Value = $7F
*
          dc      $1B4        'A' Data. TCB Reload Value = $B4
          dc      $1A1        'F' Data. TCB Reload Value = $A1
          dc      $1AB        'G' Data. TCB Reload Value = $AB
          dc      $1B4        'A' Data. TCB Reload Value = $B4
          dc      $1BD        'B' Data. TCB Reload Value = $BD
*
          dc      $1C1        'C' Data. TCB Reload Value = $C1
          dc      $17F        'C' Data. TCB Reload Value = $7F
          dc      $1FF        ' ' Data. TCB Reload Value = $FF
          dc      $17F        'C' Data. TCB Reload Value = $7F
*
          dc      $1A1        'F' Data. TCB Reload Value = $A1
          dc      $1AB        'G' Data. TCB Reload Value = $AB
          dc      $1A1        'F' Data. TCB Reload Value = $A1
          dc      $19B        'E' Data. TCB Reload Value = $9B
          dc      $18E        'D' Data. TCB Reload Value = $8E
*
          dc      $19B        'E' Data. TCB Reload Value = $9B
```

```
        dc      $1A1       'F' Data. TCB Reload Value = $A1
        dc      $19B       'E' Data. TCB Reload Value = $9B
        dc      $18E       'D' Data. TCB Reload Value = $8E
        dc      $17F       'C' Data. TCB Reload Value = $7F
*
        dc      $18E       'D' Data. TCB Reload Value = $8E
        dc      $19B       'E' Data. TCB Reload Value = $9B
        dc      $18E       'D' Data. TCB Reload Value = $8E
        dc      $17F       'C' Data. TCB Reload Value = $7F
        dc      $17A       'B' Data. TCB Reload Value = $7A
*
        dc      $17F       'C' Data. TCB Reload Value = $7F
        dc      $1FF       ' ' Data. TCB Reload Value = $FF
*
        dc      $100       ' ' Data. TCB Reload Value = $00
*
************************************************************
*       Time Period Data
************************************************************
*
        org     $0300
*
*** 1st Cycle
*
        dc      $138       'G' Time Period Data. TCC Reload Value = $38
        dc      $19C       'C' Time Period Data. TCC Reload Value = $9C
        dc      $19C       'D' Time Period Data. TCC Reload Value = $9C
        dc      $19C       'E' Time Period Data. TCC Reload Value = $9C
        dc      $19C       'F' Time Period Data. TCC Reload Value = $9C
*
        dc      $138       'G' Time Period Data. TCC Reload Value = $38
        dc      $16A       'C' Time Period Data. TCC Reload Value = $6A
        dc      $1CE       ' ' Time Period Data. TCC Reload Value = $CE
        dc      $138       'C' Time Period Data. TCC Reload Value = $38
*
        dc      $138       'A' Time Period Data. TCC Reload Value = $38
        dc      $19C       'F' Time Period Data. TCC Reload Value = $9C
        dc      $19C       'G' Time Period Data. TCC Reload Value = $9C
        dc      $19C       'A' Time Period Data. TCC Reload Value = $9C
        dc      $19C       'B' Time Period Data. TCC Reload Value = $9C
*
        dc      $138       'C' Time Period Data. TCC Reload Value = $38
        dc      $16A       'C' Time Period Data. TCC Reload Value = $6A
        dc      $1CE       ' ' Time Period Data. TCC Reload Value = $CE
        dc      $138       'C' Time Period Data. TCC Reload Value = $38
*
        dc      $138       'F' Time Period Data. TCC Reload Value = $38
        dc      $19C       'G' Time Period Data. TCC Reload Value = $9C
        dc      $19C       'F' Time Period Data. TCC Reload Value = $9C
        dc      $19C       'E' Time Period Data. TCC Reload Value = $9C
        dc      $19C       'D' Time Period Data. TCC Reload Value = $9C
```

```
*
        dc      $138        'E' Time Period Data. TCC Reload Value = $38
        dc      $19C        'F' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'E' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'D' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'C' Time Period Data. TCC Reload Value = $9C
*
        dc      $138        'B' Time Period Data. TCC Reload Value = $38
        dc      $19C        'C' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'D' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'E' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'C' Time Period Data. TCC Reload Value = $9C
*
        dc      $138        'D' Time Period Data. TCC Reload Value = $38
*
*** 2nd Cycle
*
        dc      $138        'G' Time Period Data. TCC Reload Value = $38
        dc      $19C        'C' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'D' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'E' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'F' Time Period Data. TCC Reload Value = $9C
*
        dc      $138        'G' Time Period Data. TCC Reload Value = $38
        dc      $16A        'C' Time Period Data. TCC Reload Value = $6A
        dc      $1CE        ' ' Time Period Data. TCC Reload Value = $CE
        dc      $138        'C' Time Period Data. TCC Reload Value = $38
*
        dc      $138        'A' Time Period Data. TCC Reload Value = $38
        dc      $19C        'F' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'G' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'A' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'B' Time Period Data. TCC Reload Value = $9C
*
        dc      $138        'C' Time Period Data. TCC Reload Value = $38
        dc      $16A        'C' Time Period Data. TCC Reload Value = $6A
        dc      $1CE        ' ' Time Period Data. TCC Reload Value = $CE
        dc      $138        'C' Time Period Data. TCC Reload Value = $38
*
        dc      $138        'F' Time Period Data. TCC Reload Value = $38
        dc      $19C        'G' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'F' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'E' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'D' Time Period Data. TCC Reload Value = $9C
*
        dc      $138        'E' Time Period Data. TCC Reload Value = $38
        dc      $19C        'F' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'E' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'D' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'C' Time Period Data. TCC Reload Value = $9C
*
```

```
         dc      $138      'D' Time Period Data. TCC Reload Value = $38
         dc      $19C      'E' Time Period Data. TCC Reload Value = $9C
         dc      $19C      'D' Time Period Data. TCC Reload Value = $9C
         dc      $19C      'C' Time Period Data. TCC Reload Value = $9C
         dc      $19C      'B' Time Period Data. TCC Reload Value = $9C
*
         dc      $138      'C' Time Period Data. TCC Reload Value = $38
         dc      $138      ' ' Time Period Data. TCC Reload Value = $38
*
************************************************************
*        Time Counter Data
************************************************************
*
         org     $0400
*
*** 1st Cycle
*
         dc      $10F      'G' Time Counter Data. PCNT = $F
         dc      $10F      'C' Time Counter Data. PCNT = $F
         dc      $10F      'D' Time Counter Data. PCNT = $F
         dc      $10F      'E' Time Counter Data. PCNT = $F
         dc      $10F      'F' Time Counter Data. PCNT = $F
*
         dc      $10F      'G' Time Counter Data. PCNT = $F
         dc      $10F      'C' Time Counter Data. PCNT = $F
         dc      $10F      ' ' Time Counter Data. PCNT = $F
         dc      $10F      'C' Time Counter Data. PCNT = $F
*
         dc      $10F      'A' Time Counter Data. PCNT = $F
         dc      $10F      'F' Time Counter Data. PCNT = $F
         dc      $10F      'G' Time Counter Data. PCNT = $F
         dc      $10F      'A' Time Counter Data. PCNT = $F
         dc      $10F      'B' Time Counter Data. PCNT = $F
*
         dc      $10F      'C' Time Counter Data. PCNT = $F
         dc      $10F      'C' Time Counter Data. PCNT = $F
         dc      $10F      ' ' Time Counter Data. PCNT = $F
         dc      $10F      'C' Time Counter Data. PCNT = $F
*
         dc      $10F      'F' Time Counter Data. PCNT = $F
         dc      $10F      'G' Time Counter Data. PCNT = $F
         dc      $10F      'F' Time Counter Data. PCNT = $F
         dc      $10F      'E' Time Counter Data. PCNT = $F
         dc      $10F      'D' Time Counter Data. PCNT = $F
*
         dc      $10F      'E' Time Counter Data. PCNT = $F
         dc      $10F      'F' Time Counter Data. PCNT = $F
         dc      $10F      'E' Time Counter Data. PCNT = $F
         dc      $10F      'D' Time Counter Data. PCNT = $F
         dc      $10F      'C' Time Counter Data. PCNT = $F
*
```

```
         dc      $10F       'B' Time Counter Data. PCNT = $F
         dc      $10F       'C' Time Counter Data. PCNT = $F
         dc      $10F       'D' Time Counter Data. PCNT = $F
         dc      $10F       'E' Time Counter Data. PCNT = $F
         dc      $10F       'C' Time Counter Data. PCNT = $F
*
         dc      $10D       'D' Time Counter Data. PCNT = $D
*
*** 2nd Cycle
*
         dc      $10F       'G' Time Counter Data. PCNT = $F
         dc      $10F       'C' Time Counter Data. PCNT = $F
         dc      $10F       'D' Time Counter Data. PCNT = $F
         dc      $10F       'E' Time Counter Data. PCNT = $F
         dc      $10F       'F' Time Counter Data. PCNT = $F
*
         dc      $10F       'G' Time Counter Data. PCNT = $F
         dc      $10F       'C' Time Counter Data. PCNT = $F
         dc      $10F       ' ' Time Counter Data. PCNT = $F
         dc      $10F       'C' Time Counter Data. PCNT = $F
*
         dc      $10F       'A' Time Counter Data. PCNT = $F
         dc      $10F       'F' Time Counter Data. PCNT = $F
         dc      $10F       'G' Time Counter Data. PCNT = $F
         dc      $10F       'A' Time Counter Data. PCNT = $F
         dc      $10F       'B' Time Counter Data. PCNT = $F
*
         dc      $10F       'C' Time Counter Data. PCNT = $F
         dc      $10F       'C' Time Counter Data. PCNT = $F
         dc      $10F       ' ' Time Counter Data. PCNT = $F
         dc      $10F       'C' Time Counter Data. PCNT = $F
*
         dc      $10F       'F' Time Counter Data. PCNT = $F
         dc      $10F       'G' Time Counter Data. PCNT = $F
         dc      $10F       'F' Time Counter Data. PCNT = $F
         dc      $10F       'E' Time Counter Data. PCNT = $F
         dc      $10F       'D' Time Counter Data. PCNT = $F
*
         dc      $10F       'E' Time Counter Data. PCNT = $F
         dc      $10F       'F' Time Counter Data. PCNT = $F
         dc      $10F       'E' Time Counter Data. PCNT = $F
         dc      $10F       'D' Time Counter Data. PCNT = $F
         dc      $10F       'C' Time Counter Data. PCNT = $F
*
         dc      $10F       'D' Time Counter Data. PCNT = $F
         dc      $10F       'E' Time Counter Data. PCNT = $F
         dc      $10F       'D' Time Counter Data. PCNT = $F
         dc      $10F       'C' Time Counter Data. PCNT = $F
         dc      $10F       'B' Time Counter Data. PCNT = $F
*
         dc      $10E       'C' Time Counter Data. PCNT = $E
```

```
          dc      $10F       ' ' Time Counter Data. PCNT = $F
*
          end
```

4. H4889

```
************************************************************
*
*        HMCS400 Series Application Note
*
*        'Sound Play
*        - Minuet : J.S.Bach'
*
*        MCU : H4889
*
*        External Clock : 4MHz
*        Internal Clock : 1MHz
*        Sub Clock      : 32.768kHz
*
************************************************************
*
************************************************************
*        Symbol Definition
************************************************************
*
IE       equ     0,$000    Interrupt Enable Flag
RSP      equ     1,$000    Reset Stack Pointer
IFWU     equ     2,$000    WU0-WU3 Interrupt Request Flag
IMWU     equ     3,$000    WU0-WU3 Interrupt Mask
*
IF0      equ     0,$001    INT0 Interrupt Request Flag
IM0      equ     1,$001    INT0 Interrupt Mask
IF1      equ     2,$001    INT1 Interrupt Request Flag
IM1      equ     3,$001    INT1 Interrupt Mask
*
IFTA     equ     0,$002    Timer A Interrupt Request Flag
IMTA     equ     1,$002    Timer A Interrupt Mask
IFTB     equ     2,$002    Timer B Interrupt Request Flag
IMTB     equ     3,$002    Timer B Interrupt Mask
*
IFTC     equ     0,$003    Timer C Interrupt Request Flag
IMTC     equ     1,$003    Timer C Interrupt Mask
IFAD     equ     2,$003    A/D Converter Interrupt Request Flag
IMAD     equ     3,$003    A/D Converter Interrupt Mask
*
SSR      equ     $004      System Clock selection Register
MIS      equ     $005      Miscellaneous Register
ESR      equ     $006      Edge Detect Selection Register
*
PMR0     equ     $008      Port Mode Register 0
PMR1     equ     $009      Port Mode Register 1
PMR2     equ     $00A      Port Mode Register 2
PMR3     equ     $00B      Port Mode Register 3
PMR4     equ     $00C      Port Mode Register 4
MSR1     equ     $00D      Module Standby Register 1
```

```
MSR2     equ     $00E       Module Standby register 2
TMA      equ     $00F       Timer Mode Register A
TMB1     equ     $010       Timer Mode Register B1
TMB2     equ     $011       Timer Mode register B2
TRBL     equ     $012       Timer Read Register BL
TWBL     equ     $012       Timer Write Register BL
TRBU     equ     $013       Timer Read Register BU
TWBU     equ     $013       Timer Write Register BU
TMC1     equ     $014       Timer Mode Register C1
TMC2     equ     $015       Timer Mode Register C2
TRCL     equ     $016       Timer Read Register CL
TWCL     equ     $016       Timer Write Register CL
TRCU     equ     $017       Timer Read Register CU
TWCU     equ     $017       Timer Write Register CU
TMD1     equ     $018       Timer Mode Register D1
TMD2     eqU     $019       Timer Mode Register D2
TRDL     equ     $01A       Timer Read Register DL
TWDL     equ     $01A       Timer Write Register DL
TRDU     equ     $01B       Timer Read Register DU
TWDU     equ     $01B       Timer Write Register DU
*
LSON     equ     0,$020     Low Speed on Flag
WDON     equ     1,$020     Watchdog on Flag
ADSF     equ     2,$020     A/D Start Flag
DTON     equ     3,$020     Direct Transfer on Flag
*
ICSF     equ     0,$021     Input Capture Status Flag
ICEF     equ     1,$021     Input Capture Error Flag
GEF      equ     3,$021     Gear Enable Flag
*
IFTD     equ     2,$022     Timer D Interrupt Request Flag
IMTD     equ     3,$022     Timer D Interrupt Mask
*
IFS      equ     2,$023     Serial Interrupt Request Flag
IMS      equ     3,$023     Serial Interrupt Mask
*
SMR1     equ     $024       Serial Mode Register 1
SMR2     equ     $025       Serial Mode Register 2
SRL      equ     $026       Serial Data Register L
SRU      equ     $027       Serial Data Register U
AMR      equ     $028       A/D Mode Register
*
ADRL     equ     $02A       A/D Data Register L
ADRU     equ     $02B       A/D Data Register U
LCR      equ     $02C       LCD Control Register
LMR      equ     $02D       LCD Mode Register
BMR      equ     $02E       Buzzer Mode Register
*
DCD0     equ     $030       D Port Data Control Register 0
DCD1     equ     $031       D Port Data Control Register 1
DCD2     equ     $032       D Port Data Control Register 2
```

```
*
DCR0     equ    $034      R Port Data Control Register 0
DCR1     equ    $035      R Port Data Control Register 1
DCR2     equ    $036      R Port Data Control Register 2
DCR3     equ    $037      R Port Data Control Register 3
DCR4     equ    $038      R Port Data Control Register 4
DCR5     equ    $039      R Port Data Control Register 5
DCR6     equ    $03A      R Port Data Control Register 6
DCR7     equ    $03B      R Port Data Control Register 7
DCR8     equ    $03C      R port Data Control Register 8
*
V        equ    $03F      Bank Register
*
************************************************************
*        RAM ALLOCATION
************************************************************
*
AESC     equ    $040      Accumulator Escape
BESC     equ    $041      B Register Escape
*
PLCNT    equ    $090      Pulse Counter
*
PLFLG1   equ    $091      Pulse Flag 1
PLONF    equ    0,PLFLG1  Pulse Output Enable Flag
*
CNTL     equ    $093      Lower Counter
CNTU     equ    $094      Upper Counter
PCNT     equ    $095      Period Counter
AWORK    equ    $096      Accumulator Work RAM Area
BWORK    equ    $097      B Register Work RAM Area
*
************************************************************
*        Vector Address
************************************************************
*
         org    $0000
*
         JMPL   SPLMN     Reset Interrupt
         JMPL   SPLMN     WU0-WU3 Interrupt
         JMPL   SPLMN     INT0 Interrupt
         JMPL   SPLMN     INT1 Interrupt
         JMPL   SPLMN     Timer A Interrupt
         JMPL   SPLINT    Timer B/D Interrupt
         JMPL   SPLMN     Timer C Interrupt
         JMPL   SPLMN     A/D Converter/Serial Interrupt
*
************************************************************
*        Main Program
************************************************************
*
         org    $1000
```

```
*
SPLMN     REMD    RSP       Reset Stack Pointer
          LMID    $2,SSR    Set System Clock to 1.6 - 4.5MHz
*
          LAI     $0
          LRA     $1        Initialize R1 Port PDR
          LMID    $4,DCR1   Initialize R12 Output Terminal Function
          LMID    $0,PMR2   Initialize R12 Input/Output Terminal Function
*
          LMID    $3,PLCNT  Initialize Pulse Counter
          LMID    $0,CNTL   Initialize Lower Counter
          LMID    $0,CNTU   Initialize Upper Counter
          LMID    $F,PCNT   Initialize Period Counter
          SEMD    PLONF     Initialize Pulse Output Enable Flag
*
          LMID    $8,TMC1   Initialize Timer C Function & Input Clock Period
          REMD    IFTC      Clear IFTC to 0
*
          LMID    $D,TMB1   Initialize Timer B Function & Input Clock Period
          LMID    $0,TMB2   Initialize Timer B Function
*
          SEMD    IMTB      Timer B Interrupt Disable
          REMD    IFTB      Clear IFTB to 0
*
          SEMD    IE        All Interrupt Enable
*
SPLMN00   TMD     IFTC      IFTC = "1" ?
          BRS     SPLMN10   Yes. Branch to SPLMN10
          BRS     SPLMN00   No. Branch to SPLMN00
*
          REMD    IFTC      Clear IFTC to 0
          LMID    $9,TWCL   Set TCC Reload Value Lower
          LMID    $0,TWCU   Set TCC Reload Value Upper
*
SPLMN05   TMD     IFTC      IFTC = "1" ?
          BRS     SPLMN10   Yes. Branch to SPLMN10
          SEMD    IMTB      Timer B Interrupt Disable
          REMD    IFTB      Clear IFTB to 0
          LAI     $0
          LRA     $1        Sound off
          BRS     SPLMN05   Branch to SPLMN05
*
SPLMN10   REMD    IFTC      Clear IFTC to 0
          LAMD    PCNT      Load PCNT
          AI      $1        Increment PCNT
          BRS     SPLMN40   PCNT Overflow ? Yes. Branch to SPLMN40
          LMAD    PCNT      No. Save PCNT
          BRS     SPLMN00   Branch to SPLMN00
*
SPLMN40   LAMD    CNTU      Load CNTL
          LBA
```

```
         LAMD    CNTL       Load CNTU
         P       $2         Scale Data Pattern Generation
         LMAD    AWORK      Save Scale Lower Data
         LAB
         LMAD    BWORK      Save Scale Upper Data
         ALEI    $0         Scale Upper Data <= $0 ? End Sound Play ?
         BRS     SPLMN90    Yes. Branch to SPLMN90
         ALEI    $E         Scale Upper Data <= $E ?
         BRS     SPLMN20    Yes. Branch to SPLMN20
         SEMD    IMTB       Timer B Interrupt Disable
         REMD    IFTB       Clear IFTB to 0
         LAI     $0
         LRA     $1         R12 Output Terminal is "Low" Output
         BRS     SPLMN30    Branch to SPLMN30
*
SPLMN20  LAMD    AWORK      Load AWORK
         LMAD    TWBL       Set TCB Reload Value Lower to Scale Data
         LAMD    BWORK      Load BWORK
         LMAD    TWBU       Set TCB Reload Value Upper to Scale Data
         REMD    IFTB       Clear IFTB to 0
         REMD    IMTB       Timer B Interrupt Enable
*
SPLMN30  LAMD    CNTU       Load Upper Counter
         LBA
         LAMD    CNTL       Load Lower Counter
         P       $3         Time Period Data Pattern Generation
         LMAD    TWCL       Set TCC Reload Value Lower to Time Period Lower Data
         LAB
         LMAD    TWCU       Set TCC Reload Value Upper to Time Period Upper Data
*
         LAMD    CNTU       Load Upper Counter
         LBA
         LAMD    CNTL       Load Lower Counter
         P       $4         Time Counter Data Pattern Generation
         LMAD    PCNT       Set PCNT to Time Counter Data
*
         SEC                Set Carry Flag at 1
         LAI     $0
         AMCD    CNTL       Increment Lower Counter
         LMAD    CNTL       Save Lower Counter
         LAI     $0
         AMCD    CNTU       If CNTL Overflow, Increment Upper Counter
         LMAD    CNTU       Save Upper Counter
         BR      *+1
         BRS     SPLMN00    Branch to SPLMN00
*
SPLMN90  SEMD    IMTB       Timer B Interrupt Disable
         LMID    $0,CNTL    Initialize Lower Counter
         LMID    $0,CNTU    Initialize Upper Counter
         LAI     $0
         LRA     $1         Initialize R1 Port PDR
```

```
         LMID    $0,TWCL    Initialize TCC Reload Value Lower
         LMID    $8,TWCU    Initialize TCC Reload Value Upper
         REMD    IFTC       Clear IFTC to 0
         BRS     SPLMN00    Branch to SPLMN00
*
************************************************************
*        Timer B Interrupt Process
*        ---Pulse Output Routine---
************************************************************
*
SPLINT   REMD    IFTB       Clear IFTB to 0
*
         LMAD    AESC       Store Accumulator
         LAB
         LMAD    BESC       Store B Register
*
         TMD     PLONF      Pulse Enable Flag = "1" ?
         BRS     PLI00      Yes. Branch to PLI00
         LAI     $0         No. Stop Output Pulse
         LRA     $1         R12 Output Terminal is "Low" Output
         BRS     PLI99      Branch to PLI99
*
PLI00    LAMD    PLCNT      Load Pulse Counter Value
         LBA                Store Pulse Counter Value
         ALEI    $1         PLCNT <= $1 ? Is R12 Output Terminal "High" Output ?
         BRS     PLI10      Yes. Branch to PLI10
         LAI     $0         No. R12 Output Terminal is "Low" Output
         BRS     PLI20      Branch to PLI20
PLI10    LAI     $4
PLI20    LRA     $1         R03 Output Terminal is "High" Output
         LAB                Restore Pulse Counter Value
         AI      $F         PLCNT + $F <= $F ? Pulse Counter = $0 ?
         BRS     PLI30      No. Branch to PLI30
         LAI     $3         Yes. Pulse Counter Initialize
PLI30    LMAD    PLCNT      Save Pulse Counter Value
*
PLI99    LAMD    BESC       Restore B Register
         LBA
         LAMD    AESC       Restore Accumulator
*
         RTNI               Return from Interrupt
*
************************************************************
*        Scale Data
************************************************************
*
         ORG     $0200
*
*** 1st Cycle
*
         dc      $1AB       'G' Data. TCB Reload Value = $AB
```

```
          dc      $17F       'C' Data. TCB Reload Value = $7F
          dc      $18E       'D' Data. TCB Reload Value = $8E
          dc      $19B       'E' Data. TCB Reload Value = $9B
          dc      $1A1       'F' Data. TCB Reload Value = $A1
*
          dc      $1AB       'G' Data. TCB Reload Value = $AB
          dc      $17F       'C' Data. TCB Reload Value = $7F
          dc      $1FF       ' ' Data. TCB Reload Value = $FF
          dc      $17F       'C' Data. TCB Reload Value = $7F
*
          dc      $1B4       'A' Data. TCB Reload Value = $B4
          dc      $1A1       'F' Data. TCB Reload Value = $A1
          dc      $1AB       'G' Data. TCB Reload Value = $AB
          dc      $1B4       'A' Data. TCB Reload Value = $B4
          dc      $1BD       'B' Data. TCB Reload Value = $BD
*
          dc      $1C1       'C' Data. TCB Reload Value = $C1
          dc      $17F       'C' Data. TCB Reload Value = $7F
          dc      $1FF       ' ' Data. TCB Reload Value = $FF
          dc      $17F       'C' Data. TCB Reload Value = $7F
*
          dc      $1A1       'F' Data. TCB Reload Value = $A1
          dc      $1AB       'G' Data. TCB Reload Value = $AB
          dc      $1A1       'F' Data. TCB Reload Value = $A1
          dc      $19B       'E' Data. TCB Reload Value = $9B
          dc      $18E       'D' Data. TCB Reload Value = $8E
*
          dc      $19B       'E' Data. TCB Reload Value = $9B
          dc      $1A1       'F' Data. TCB Reload Value = $A1
          dc      $19B       'E' Data. TCB Reload Value = $9B
          dc      $18E       'D' Data. TCB Reload Value = $8E
          dc      $17F       'C' Data. TCB Reload Value = $7F
*
          dc      $17A       'B' Data. TCB Reload Value = $7A
          dc      $17F       'C' Data. TCB Reload Value = $7F
          dc      $18E       'D' Data. TCB Reload Value = $8E
          dc      $19B       'E' Data. TCB Reload Value = $9B
          dc      $17F       'C' Data. TCB Reload Value = $7F
*
          dc      $18E       'D' Data. TCB Reload Value = $8E
*
*** 2nd Cycle
*
          dc      $1AB       'G' Data. TCB Reload Value = $AB
          dc      $17F       'C' Data. TCB Reload Value = $7F
          dc      $18E       'D' Data. TCB Reload Value = $8E
          dc      $19B       'E' Data. TCB Reload Value = $9B
          dc      $1A1       'F' Data. TCB Reload Value = $A1
*
          dc      $1AB       'G' Data. TCB Reload Value = $AB
          dc      $17F       'C' Data. TCB Reload Value = $7F
```

```
        dc      $1FF       ' ' Data. TCB Reload Value = $FF
        dc      $17F       'C' Data. TCB Reload Value = $7F
*
        dc      $1B4       'A' Data. TCB Reload Value = $B4
        dc      $1A1       'F' Data. TCB Reload Value = $A1
        dc      $1AB       'G' Data. TCB Reload Value = $AB
        dc      $1B4       'A' Data. TCB Reload Value = $B4
        dc      $1BD       'B' Data. TCB Reload Value = $BD
*
        dc      $1C1       'C' Data. TCB Reload Value = $C1
        dc      $17F       'C' Data. TCB Reload Value = $7F
        dc      $1FF       ' ' Data. TCB Reload Value = $FF
        dc      $17F       'C' Data. TCB Reload Value = $7F
*
        dc      $1A1       'F' Data. TCB Reload Value = $A1
        dc      $1AB       'G' Data. TCB Reload Value = $AB
        dc      $1A1       'F' Data. TCB Reload Value = $A1
        dc      $19B       'E' Data. TCB Reload Value = $9B
        dc      $18E       'D' Data. TCB Reload Value = $8E
*
        dc      $19B       'E' Data. TCB Reload Value = $9B
        dc      $1A1       'F' Data. TCB Reload Value = $A1
        dc      $19B       'E' Data. TCB Reload Value = $9B
        dc      $18E       'D' Data. TCB Reload Value = $8E
        dc      $17F       'C' Data. TCB Reload Value = $7F
*
        dc      $18E       'D' Data. TCB Reload Value = $8E
        dc      $19B       'E' Data. TCB Reload Value = $9B
        dc      $18E       'D' Data. TCB Reload Value = $8E
        dc      $17F       'C' Data. TCB Reload Value = $7F
        dc      $17A       'B' Data. TCB Reload Value = $7A
*
        dc      $17F       'C' Data. TCB Reload Value = $7F
        dc      $1FF       ' ' Data. TCB Reload Value = $FF
*
        dc      $100       '  ' Data. TCB Reload Value = $00
*
************************************************************
*       Time Period Data
************************************************************
*
        org     $0300
*
*** 1st Cycle
*
        dc      $138       'G' Time Period Data. TCC Reload Value = $38
        dc      $19C       'C' Time Period Data. TCC Reload Value = $9C
        dc      $19C       'D' Time Period Data. TCC Reload Value = $9C
        dc      $19C       'E' Time Period Data. TCC Reload Value = $9C
        dc      $19C       'F' Time Period Data. TCC Reload Value = $9C
*
```

```
         dc      $138       'G' Time Period Data. TCC Reload Value = $38
         dc      $16A       'C' Time Period Data. TCC Reload Value = $6A
         dc      $1CE       ' ' Time Period Data. TCC Reload Value = $CE
         dc      $138       'C' Time Period Data. TCC Reload Value = $38
*
         dc      $138       'A' Time Period Data. TCC Reload Value = $38
         dc      $19C       'F' Time Period Data. TCC Reload Value = $9C
         dc      $19C       'G' Time Period Data. TCC Reload Value = $9C
         dc      $19C       'A' Time Period Data. TCC Reload Value = $9C
         dc      $19C       'B' Time Period Data. TCC Reload Value = $9C
*
         dc      $138       'C' Time Period Data. TCC Reload Value = $38
         dc      $16A       'C' Time Period Data. TCC Reload Value = $6A
         dc      $1CE       ' ' Time Period Data. TCC Reload Value = $CE
         dc      $138       'C' Time Period Data. TCC Reload Value = $38
*
         dc      $138       'F' Time Period Data. TCC Reload Value = $38
         dc      $19C       'G' Time Period Data. TCC Reload Value = $9C
         dc      $19C       'F' Time Period Data. TCC Reload Value = $9C
         dc      $19C       'E' Time Period Data. TCC Reload Value = $9C
         dc      $19C       'D' Time Period Data. TCC Reload Value = $9C
*
         dc      $138       'E' Time Period Data. TCC Reload Value = $38
         dc      $19C       'F' Time Period Data. TCC Reload Value = $9C
         dc      $19C       'E' Time Period Data. TCC Reload Value = $9C
         dc      $19C       'D' Time Period Data. TCC Reload Value = $9C
         dc      $19C       'C' Time Period Data. TCC Reload Value = $9C
*
         dc      $138       'B' Time Period Data. TCC Reload Value = $38
         dc      $19C       'C' Time Period Data. TCC Reload Value = $9C
         dc      $19C       'D' Time Period Data. TCC Reload Value = $9C
         dc      $19C       'E' Time Period Data. TCC Reload Value = $9C
         dc      $19C       'C' Time Period Data. TCC Reload Value = $9C
*
         dc      $138       'D' Time Period Data. TCC Reload Value = $38
*
*** 2nd Cycle
*
         dc      $138       'G' Time Period Data. TCC Reload Value = $38
         dc      $19C       'C' Time Period Data. TCC Reload Value = $9C
         dc      $19C       'D' Time Period Data. TCC Reload Value = $9C
         dc      $19C       'E' Time Period Data. TCC Reload Value = $9C
         dc      $19C       'F' Time Period Data. TCC Reload Value = $9C
*
         dc      $138       'G' Time Period Data. TCC Reload Value = $38
         dc      $16A       'C' Time Period Data. TCC Reload Value = $6A
         dc      $1CE       ' ' Time Period Data. TCC Reload Value = $CE
         dc      $138       'C' Time Period Data. TCC Reload Value = $38
*
         dc      $138       'A' Time Period Data. TCC Reload Value = $38
         dc      $19C       'F' Time Period Data. TCC Reload Value = $9C
```

```
        dc      $19C        'G' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'A' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'B' Time Period Data. TCC Reload Value = $9C
*
        dc      $138        'C' Time Period Data. TCC Reload Value = $38
        dc      $16A        'C' Time Period Data. TCC Reload Value = $6A
        dc      $1CE        ' ' Time Period Data. TCC Reload Value = $CE
        dc      $138        'C' Time Period Data. TCC Reload Value = $38
*
        dc      $138        'F' Time Period Data. TCC Reload Value = $38
        dc      $19C        'G' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'F' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'E' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'D' Time Period Data. TCC Reload Value = $9C
*
        dc      $138        'E' Time Period Data. TCC Reload Value = $38
        dc      $19C        'F' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'E' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'D' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'C' Time Period Data. TCC Reload Value = $9C
*
        dc      $138        'D' Time Period Data. TCC Reload Value = $38
        dc      $19C        'E' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'D' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'C' Time Period Data. TCC Reload Value = $9C
        dc      $19C        'B' Time Period Data. TCC Reload Value = $9C
*
        dc      $138        'C' Time Period Data. TCC Reload Value = $38
        dc      $138        ' ' Time Period Data. TCC Reload Value = $38
*
************************************************************
*       Time Counter Data
************************************************************
*
        org     $0400
*
*** 1st Cycle
*
        dc      $10F        'G' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
        dc      $10F        'D' Time Counter Data. PCNT = $F
        dc      $10F        'E' Time Counter Data. PCNT = $F
        dc      $10F        'F' Time Counter Data. PCNT = $F
*
        dc      $10F        'G' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
        dc      $10F        ' ' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
*
        dc      $10F        'A' Time Counter Data. PCNT = $F
        dc      $10F        'F' Time Counter Data. PCNT = $F
```

```
        dc      $10F      'G' Time Counter Data. PCNT = $F
        dc      $10F      'A' Time Counter Data. PCNT = $F
        dc      $10F      'B' Time Counter Data. PCNT = $F
*
        dc      $10F      'C' Time Counter Data. PCNT = $F
        dc      $10F      'C' Time Counter Data. PCNT = $F
        dc      $10F      ' ' Time Counter Data. PCNT = $F
        dc      $10F      'C' Time Counter Data. PCNT = $F
*
        dc      $10F      'F' Time Counter Data. PCNT = $F
        dc      $10F      'G' Time Counter Data. PCNT = $F
        dc      $10F      'F' Time Counter Data. PCNT = $F
        dc      $10F      'E' Time Counter Data. PCNT = $F
        dc      $10F      'D' Time Counter Data. PCNT = $F
*
        dc      $10F      'E' Time Counter Data. PCNT = $F
        dc      $10F      'F' Time Counter Data. PCNT = $F
        dc      $10F      'E' Time Counter Data. PCNT = $F
        dc      $10F      'D' Time Counter Data. PCNT = $F
        dc      $10F      'C' Time Counter Data. PCNT = $F
*
        dc      $10F      'B' Time Counter Data. PCNT = $F
        dc      $10F      'C' Time Counter Data. PCNT = $F
        dc      $10F      'D' Time Counter Data. PCNT = $F
        dc      $10F      'E' Time Counter Data. PCNT = $F
        dc      $10F      'C' Time Counter Data. PCNT = $F
*
        dc      $10D      'D' Time Counter Data. PCNT = $D
*
*** 2nd Cycle
*
        dc      $10F      'G' Time Counter Data. PCNT = $F
        dc      $10F      'C' Time Counter Data. PCNT = $F
        dc      $10F      'D' Time Counter Data. PCNT = $F
        dc      $10F      'E' Time Counter Data. PCNT = $F
        dc      $10F      'F' Time Counter Data. PCNT = $F
*
        dc      $10F      'G' Time Counter Data. PCNT = $F
        dc      $10F      'C' Time Counter Data. PCNT = $F
        dc      $10F      ' ' Time Counter Data. PCNT = $F
        dc      $10F      'C' Time Counter Data. PCNT = $F
*
        dc      $10F      'A' Time Counter Data. PCNT = $F
        dc      $10F      'F' Time Counter Data. PCNT = $F
        dc      $10F      'G' Time Counter Data. PCNT = $F
        dc      $10F      'A' Time Counter Data. PCNT = $F
        dc      $10F      'B' Time Counter Data. PCNT = $F
*
        dc      $10F      'C' Time Counter Data. PCNT = $F
        dc      $10F      'C' Time Counter Data. PCNT = $F
        dc      $10F      ' ' Time Counter Data. PCNT = $F
```

```
        dc      $10F        'C' Time Counter Data. PCNT = $F
*
        dc      $10F        'F' Time Counter Data. PCNT = $F
        dc      $10F        'G' Time Counter Data. PCNT = $F
        dc      $10F        'F' Time Counter Data. PCNT = $F
        dc      $10F        'E' Time Counter Data. PCNT = $F
        dc      $10F        'D' Time Counter Data. PCNT = $F
*
        dc      $10F        'E' Time Counter Data. PCNT = $F
        dc      $10F        'F' Time Counter Data. PCNT = $F
        dc      $10F        'E' Time Counter Data. PCNT = $F
        dc      $10F        'D' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
*
        dc      $10F        'D' Time Counter Data. PCNT = $F
        dc      $10F        'E' Time Counter Data. PCNT = $F
        dc      $10F        'D' Time Counter Data. PCNT = $F
        dc      $10F        'C' Time Counter Data. PCNT = $F
        dc      $10F        'B' Time Counter Data. PCNT = $F
*
        dc      $10E        'C' Time Counter Data. PCNT = $E
        dc      $10F        ' ' Time Counter Data. PCNT = $F
*
        end
```

2.2 Stepping Motor Control

Stepping Motor Control	MCU: H4344/H4318/H4359/ H4369/H4889	Functions Used: R2/R8 Port and Timer B

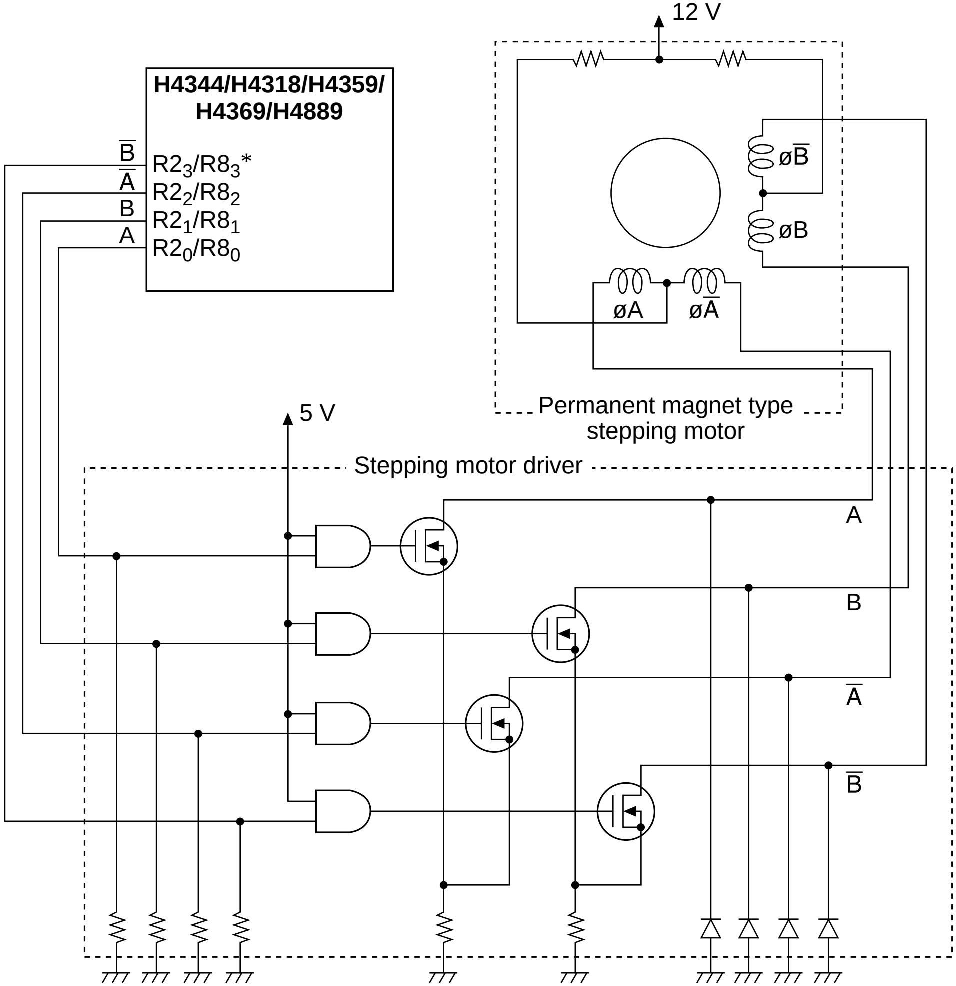
Specifications

1. The H4344/H4318/H4359/H4369/H4889 Series are used for stepping motor control. The stepping motor is of the permanent magnet type.

2. The stepping motor repeats a forward run, stop, and reverse run cycle.

3. The stepping motor is run for approximately 4 seconds in forward, in stop, and in reverse.

4. Figure 1 shows the example stepping motor control circuit used in this example task.



Note: * $R2_0$ to $R2_3$ pins are used in the H4344/H4889 Series.
$R8_0$ to $R8_3$ pins are used in the H4318/H4359/H4369 Series.

Figure 1 Example Stepping Motor Control Circuit

Concepts

1. Stepping motors have superb starting, stopping, and positional control characteristics.

2. Stepping motors are synchronous motors that run in sync with the pulse signals output from a pulse oscillator. The motor is run in sync with the pulse signals, so, because there is no speed fluctuation due to load fluctuations, it will stop exactly at the intended position.

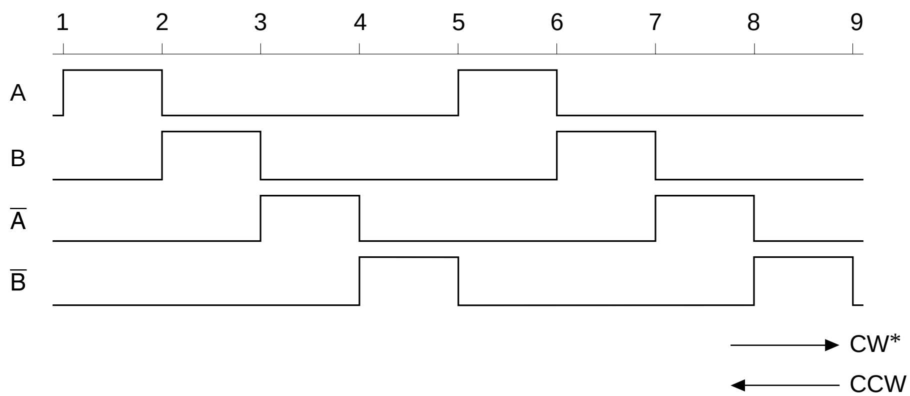
3. Stepping motors have the following features:

   a. The rotational angle of stepping motors is proportional to the number of input pulses.
   b. There is minimal angle error per step, and no cumulative error.
   c. Stepping motors have superb starting and stopping response.
   d. By direct connection to the motor shaft, it's possible to achieve synchronous rotation at extremely low speeds.
   e. The self holding capacity of stepping motors means that the stop position can be maintained.
   f. Superb control characteristics can be achieved using open-loop control.

4. The following shows the method of excitation (the method of turning on the current to the windings of the stepping motor in order) of a 2-phase stepping motor.

   - Single-phase excitation

     In this method, only one phase excitation is performed at all times. This results in lower power dissipation, but because of the small amount of damping, vibration is more likely. Figure 2 shows the excitation sequence in single-phase excitation.



Note: * CW: Clockwise (as seen from axis)
CCW: Counterclockwise

Figure 2 Excitation Sequence in Single-Phase Excitation

- Two-phase excitation

  In this method, two-phase excitation is performed at all times. Twice the input is required over single-phase excitation, but the output torque is greater and damping is superior. Figure 3 shows the excitation sequence in two-phase excitation.

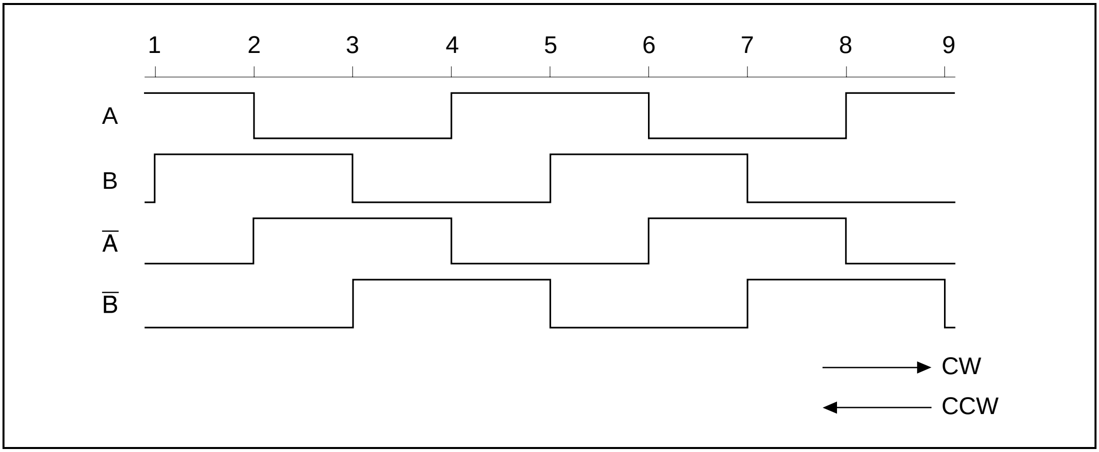


Figure 3 Excitation Sequence in Two-Phase Excitation

- 1-2-phase excitation

  In this method, single-phase excitation and two-phase excitation are alternated. When a stepping motor is driven using this method, the motor's step angle is halved. Because the step angle is halved, running is smoother and there is minimal vibration. Figure 4 shows the excitation sequence in 1-2-phase excitation.

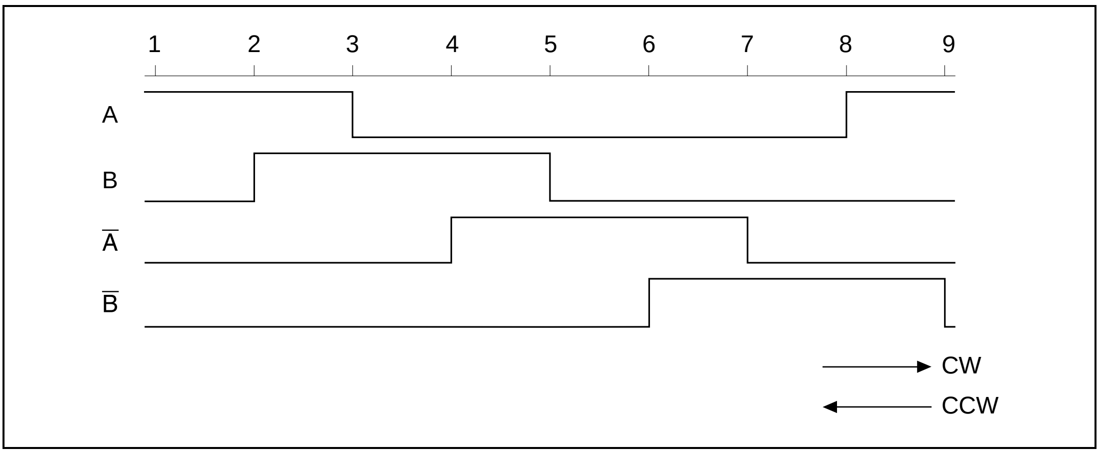


Figure 4 Excitation Sequence in 1-2-Phase Excitation

5. In this example task, the stepping motor is controlled using two-phase excitation. Figure 5 shows an example of port output when the motor is run in the forward direction in this task.

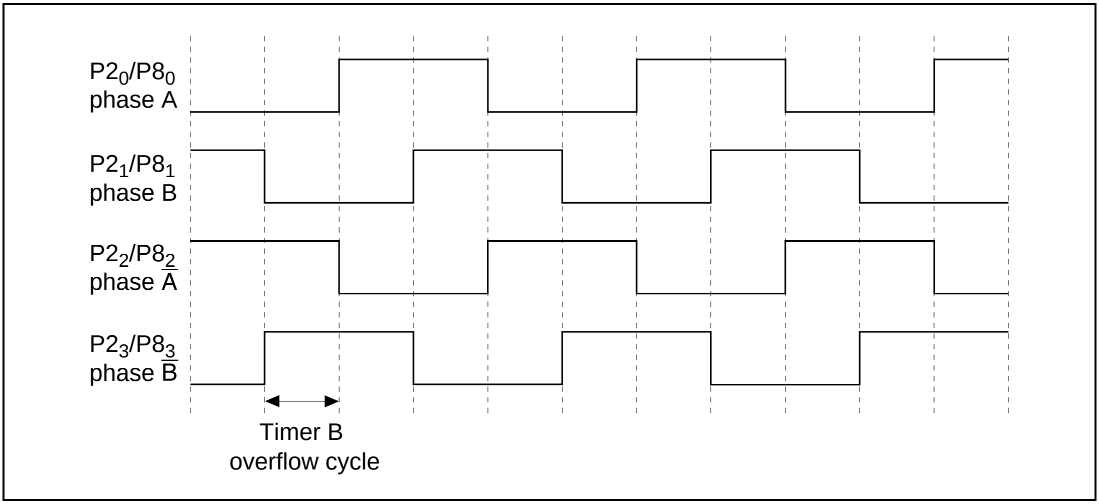


Figure 5 Example Port Output With Motor Running in Forward Direction

Description of Functions

1. In this example task, we use a permanent magnet type of stepping motor (KP6P8-701 from Japan Servo Co., Ltd.) Table 1 shows the standard specifications of the KP6P8-701. Figure 6 shows the wiring.

Table 1 KP6P8-701 Standard Specifications

Item	Unit	Value
Model No.	—	KP6P8-701
No. of phases	—	2
Step angle	deg./step	7.5
Voltage	V	12
Current	A/PHASE	0.33
Winding resistance	Ω/PHASE	36
Inductance	mH/PHASE	28
Maximum holding torque	gf·cm	800
Détente torque	gf·cm	160
Rotor inertia	gf·cm^2	23.7
Weight	kg	0.25
Insulation class	—	Type E or equivalent
Insulation resistance	—	500 V DC 100 MΩ 1 min.
Dielectric strength	—	500 V AC 50 Hz 1 min.
Operating temperature range	°C	–10 to +45
Temperature rise	deg	70
Lead specifications	—	AWG #22 UL3266

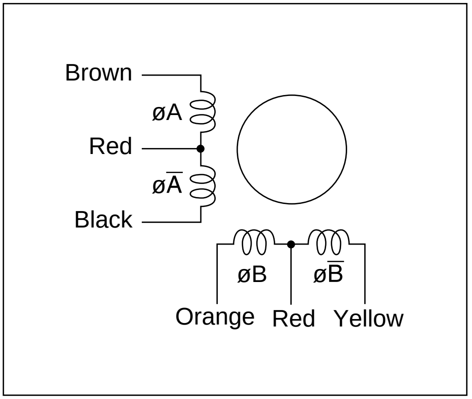


Figure 6 KP6P8-701 Wiring

2. This section describes the functions of the H4344/H4318/H4359/H4369/H4889 used in stepping motor control. Figure 7 is a block diagram of the functions used in this example task.

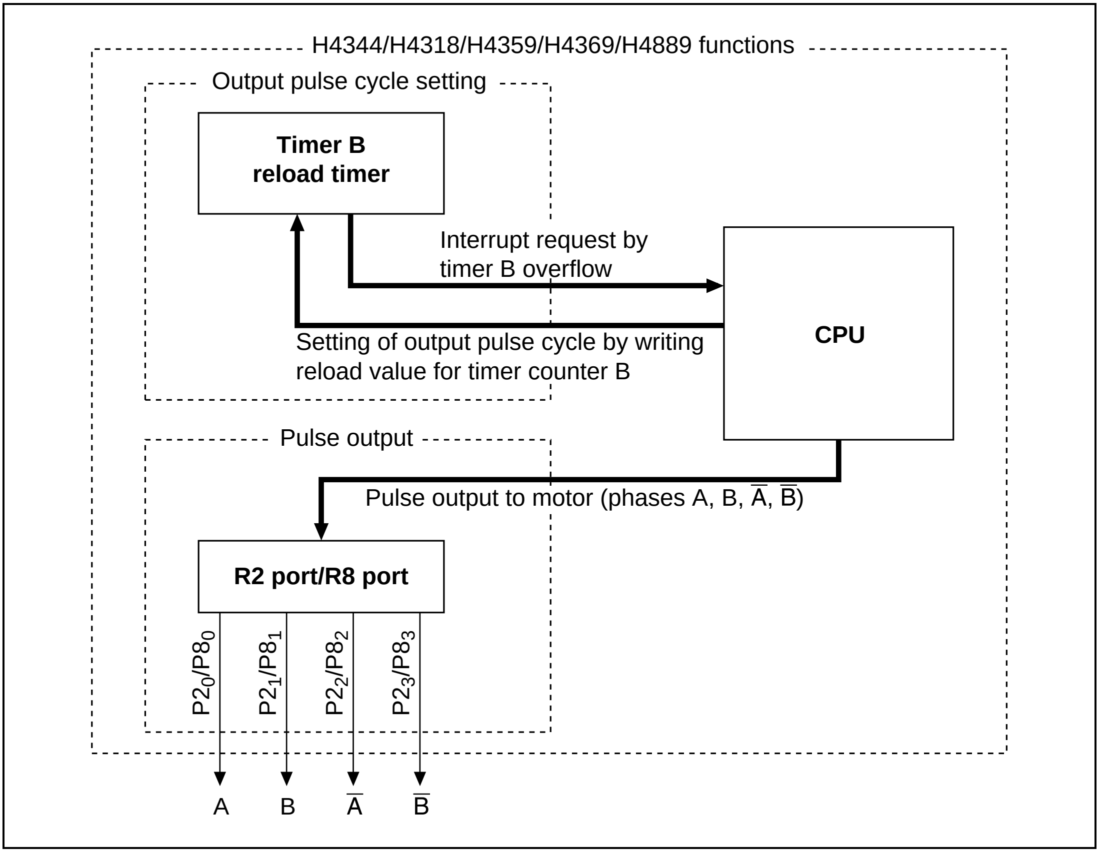


Figure 7 Block Diagram of Functions H4344/H4318/H4359/H4369/H4889 Used in Stepping Motor Control

- Timer B reload timer functions

  Sets the output pulse cycle. The output pulse cycle is determined from the timer counter B reload value. The reload value to be set is referenced from the data table.

- R2 port/R8 port functions

  These are the output pins for the pulses (phases A, B, $\overline{A}$, $\overline{B}$) output to the stepping motor.

3. The following describes the functions of timer B and the R2/R8 port.

a. Figure 8 is a block diagram of timer B functions.

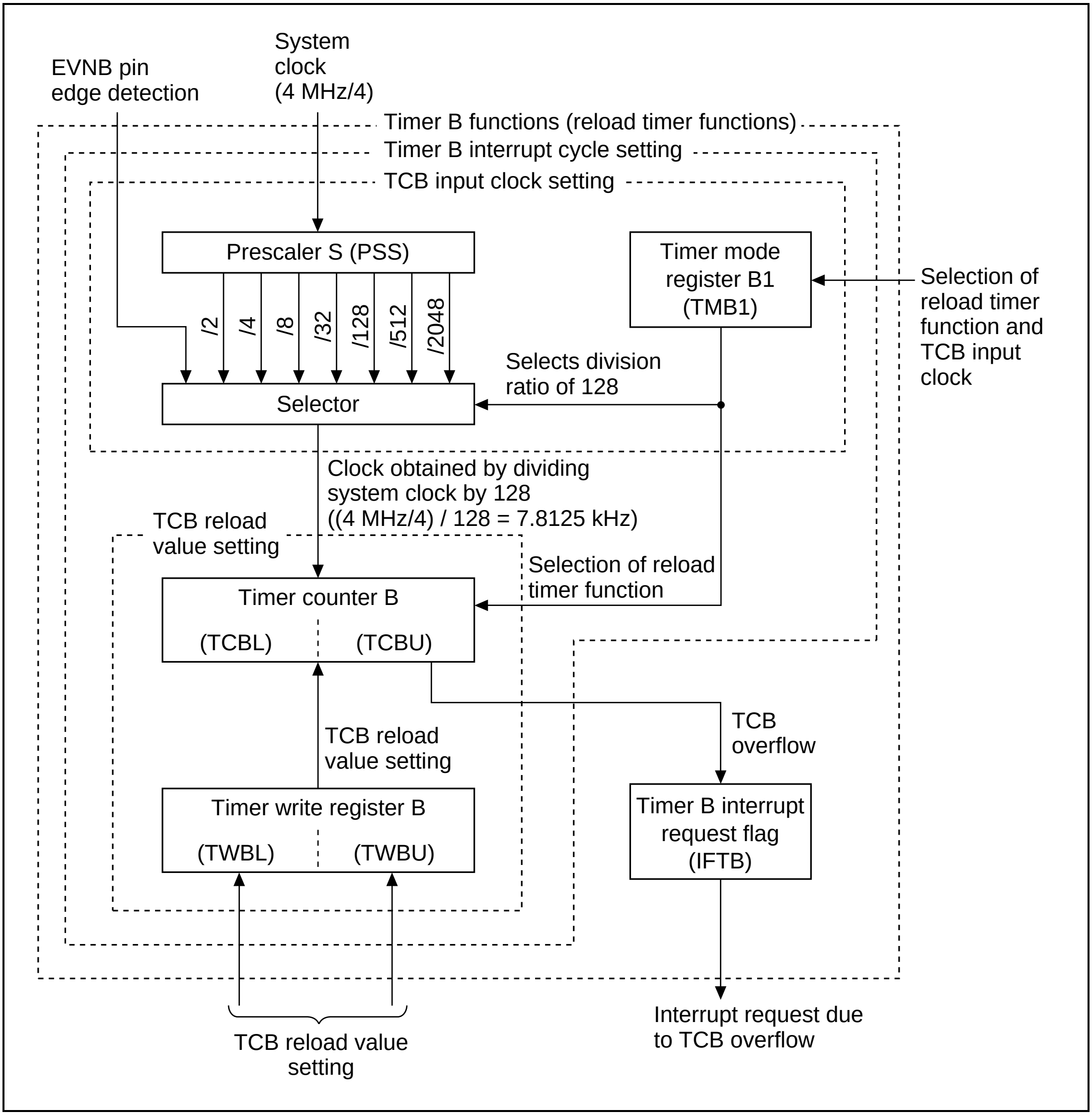


Figure 8 Block Diagram of Timer B Functions

b. Timer B is an 8-bit multifunction timer (free running/event counter/reload timer/input capture[*1]). In this example task, timer B is used as a reload timer. Table 2 describes the timer B functions.

Table 2 Timer B Functions

Timer Mode Register B1 (TMB1)	
Function	TMB1 is a 4-bit write-only register. It selects the timer B function (free-running/reload timer) and operating clock. TMB1 is initialized to $0 when reset and in stop mode.
Timer Write Register BL, U (TWBL, TWBU)	
Function	TWBL and TWBU form an 8-bit write-only register, which is made up of the lower digit (TWBL) and upper digit (TWBU). TWBL and TWBU are used for the initial TCB setting (the reload setting when operation as a reload timer).
Timer Counter B (TCB)	
Function	TCB is an 8-bit up-counter, which is incremented by the input internal clock. The TCB input clock is selected using bits TMB12 to TMB10 of TMB1. The value written to TWBL and TWBU is also written to TCB. When TCB overflows, the timer B interrupt request flag (IFTB) is set to “1”. If, at this point, timer B is set as a reload timer, the value of TWBL and TWBU is written to this counter and the count starts from this value. TCB is initialized to $00 when reset and in stop mode.
Prescaler S (PSS)	
Function	PSS is an 11-bit counter to which the system clock is input when in active mode and standby mode, and the subsystem clock is input when in subactive mode[*2]. PSS is initialized to $000 at a reset, and starts to count the system clock when the reset is canceled. PSS operation is halted when reset, in stop mode, and in watch mode[*2]. However, it runs in other operating modes. The PSS output is shared by the internal peripheral modules, the division ratio being set independently for each of the internal peripheral modules.
Timer B Interrupt Request Flag (IFTB)	
Function	IFTB reflects the existence of the timer B interrupt request. When timer B overflows, IFTB is set to “1”. IFTB can only be read/written to (only “0” can be written) using bit operation commands. Note that IFTB is not automatically cleared even when the interrupt is received, and must be cleared by writing “0” using software. IFTB is cleared at a reset and in stop mode.
Timer B Interrupt Mask (IMTB)	
Function	IMTB is the bit that masks IFTB. When IFTB is set to “1” and, additionally, IMTB is “0”, a timer B interrupt request is sent to the CPU (when IE = “1”). If IFTB is set to “1” but IMTB is “1”, no interrupt request is sent to the CPU and the timer B interrupt is held. IMTB can only be read or written to using bit operation commands. It is set to “1” at a reset and in stop mode.

Notes: 1. Applies to H4318/H4359/H4369 Series only. In the H4344/H4889 Series, timer B has no input capture function.
2. Applies only to H4369/H4889 Series.

c. Figure 9 is a block diagram of the R2 port functions used in the H4344; figure 10 is a block diagram of the R2 port functions used in the H4889; figure 11 is a block diagram of the R8 port functions used in the H4318/H4359/H4369.

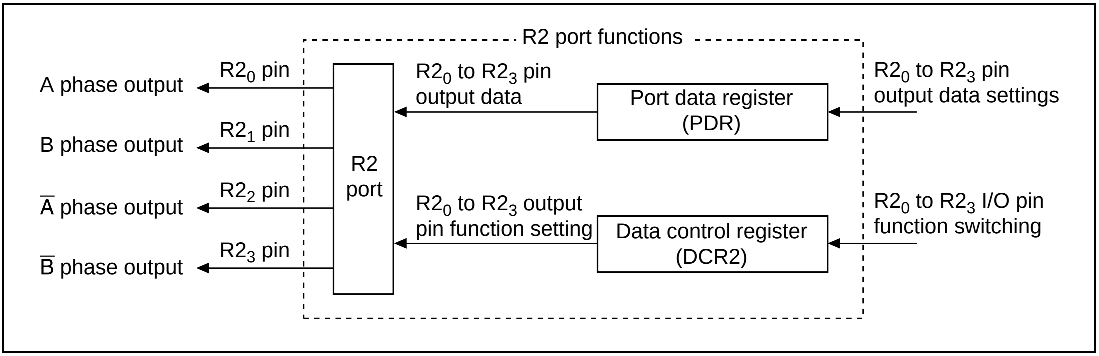


Figure 9 Block Diagram of Port R2 Functions in H4344 Series

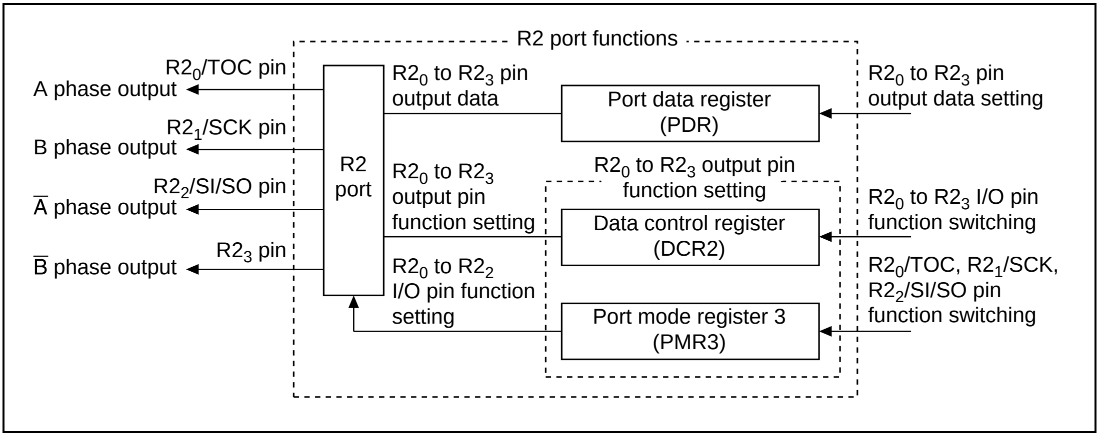


Figure 10 Block Diagram of Port R2 Functions in H4889 Series

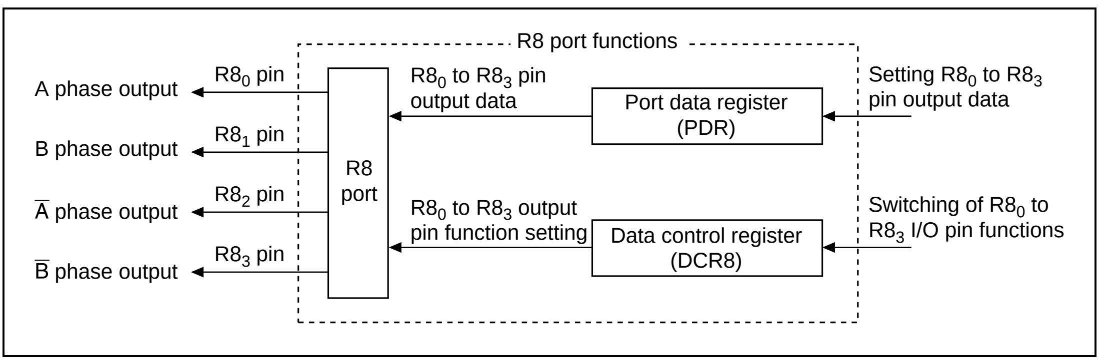


Figure 11 Block Diagram of Port R8 Functions in H4318/H4359/H4369 Series

d. The R2 port in the H4344/H4889 and the R8 port in the H4318/H4359/H4369 are 4-bit I/O ports. Both R2 and R8 are capable of 4-bit input using the LAR and LBR commands, and 4-bit output using the LRA and LRB commands. The output data is stored in the PDR of the respective pins.

   In this example task, the $R2_0$ to $R2_3$ pins in the H4344/H4889 Series and the $R8_0$ to $R8_3$ pins in the H4318/H4359/H4369 Series are set for output and used to output pulses to the stepping motor.

   Table 3 describes the functions of the R2 port in the H4344/H4889 and the R8 port in the H4318/H4359/H4369.

Table 3 R2 Port Functions in H4344/H4889 and R8 Port Functions in H4318/H4359/H4369

Data Control Register R2 (DCR2)	Note: Applies to H4344/H4889 Series
Function	DCR2 switches the I/O pin function of the R2 port. When any bit of DCR2 is cleared to "0", the output buffer (CMOS) of the corresponding pin is turned OFF and the output is set to high impedance. When the respective bit of DCR2 is set to "1", the output buffer of the corresponding pin is set ON and the corresponding PDR value is output.
Data Control Register R8 (DCR8)	Note: Applies to H4318/H4359/H4369 Series
Function	DCR8 switches the I/O pin function of the R8 port. When any bit of DCR8 is cleared to "0", the output buffer (CMOS) of the corresponding pin is turned OFF and the output is set to high impedance. When the respective bit of DCR8 is set to "1", the output buffer of the corresponding pin is set ON and the corresponding PDR value is output.
Port Mode Register 3 (PMR3)	Note: Applies to H4889 Series
Function	PMR3 is a 4-bit write-only register. Bits PMR33 to PMR30 switch the functions of the R2 port's dual-function pins.
Port Data Register (PDR)	
Function	The I/O pins of the R ports have built-in PDRs to store the output data. When the LRA and LRB commands are executed, the contents of the accumulator (A) and B register (B) are transferred to the PDR of the specified R port. When the corresponding bit of the DCR of the R port is "1", the output buffer of the appropriate pin is set ON and the value in the PDR is output via that pin. The PDR is initialized to $F at a reset.

3. Table 4 shows the allocation of functions in this example task.

Table 4 Function Allocation

Function	Function Allocation
System clock	The system clock is obtained by dividing the clock output from the system clock oscillator by 4. It is used for operating the CPU and internal peripheral modules. In this example task, a 4 MHz system clock oscillator is used, so the clock supplied to the CPU and internal peripheral modules is 1 MHz. The clock used by timer B is obtained by dividing the 1 MHz clock at PSS.
PSS	The clock input to timer B is obtained by dividing the system clock. The clock supplied to timer B is obtained by dividing the system clock by 128.
TCB	This is an 8-bit up-counter. The count starts from the value set in TWBL and TWBU. When an overflow occurs, IFTB is set to “1”. After an overflow, the reload value set in TWBL and TWBU is set in TCB.
TWBL, TWBU	The TCB reload value is set in TWBL and TWBU. The reload value is determined from the pulse cycle to be output to the stepping motor.
TMB1	TMB1 selects the reload timer function for timer B and a clock obtained by dividing the system clock by 128 as the TCB input clock.
IFTB	IFTB reflects the existence of a timer B interrupt request. The pulse output pin output level is set in the timer B interrupt processing.
IMTB	Enables/disables timer B interrupt requests.
DCR2 (H4344/H4889) DCR8 (H4318/H4359/ H4369)	Sets the $R2_0$ to $R2_3$ pins (H4344/H4889) and $R8_0$ to $R8_3$ pins (H4318/H4359/H4369 Series) as output pins.
PMR3 (H4889)	Sets the $R2_0$/TOC dual-function pin as an $R2_0$ I/O pin and the $R2_1$/SCK dual-function pin as an $R2_1$ pin.
PDR	Stores the output data for the $R2_0$ to $R2_3$/$R8_0$ to $R8_3$ pins.
$R2_0$ to $R2_3$ pins	Output pins for the pulse output to the stepping motor in the H4344/H4889 Series.
$R8_0$ to $R8_3$ pins	Output pins for the pulse output to the stepping motor in H4318/H4359/ H4369 Series.

Description of Operation

1. Figure 12 is a flowchart of the stepping motor control.

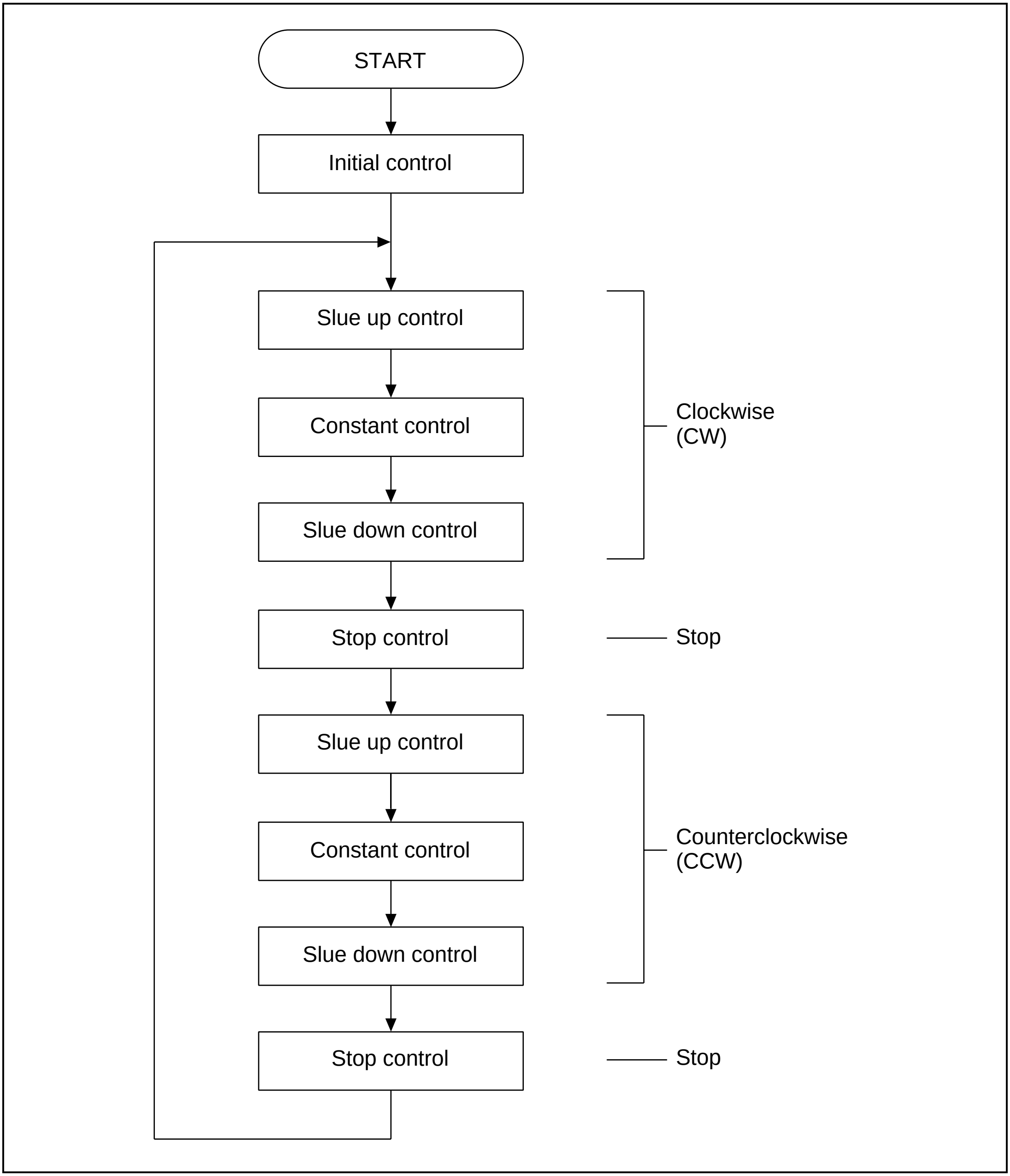


Figure 12 Flowchart of Stepping Motor Control

2. Figure 13 shows the operating principle of initial control.

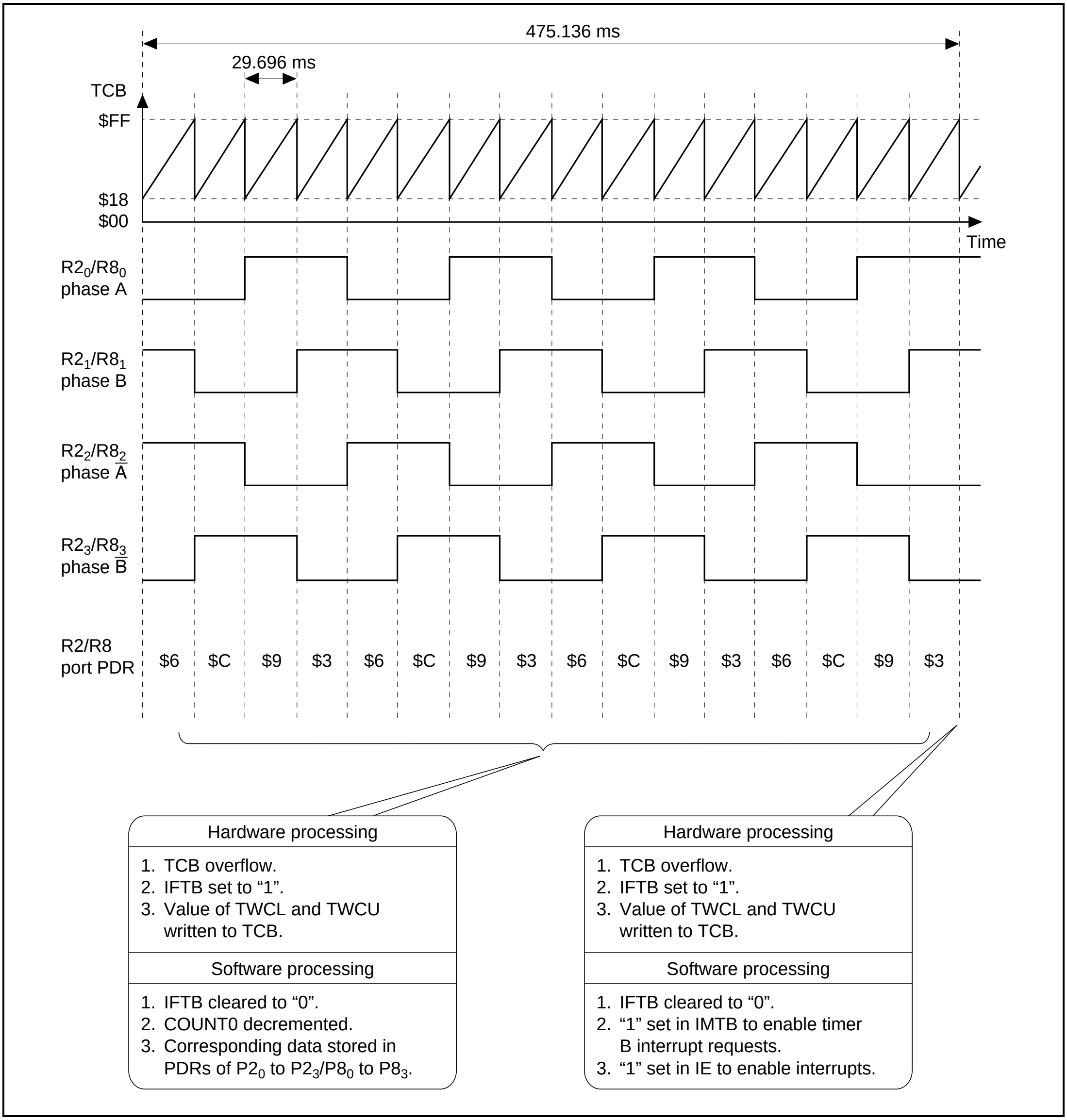


Figure 13 Operating Principle of Initial Control

3. Figure 14 shows the operating principle of Slue up control when the motor is running clockwise.

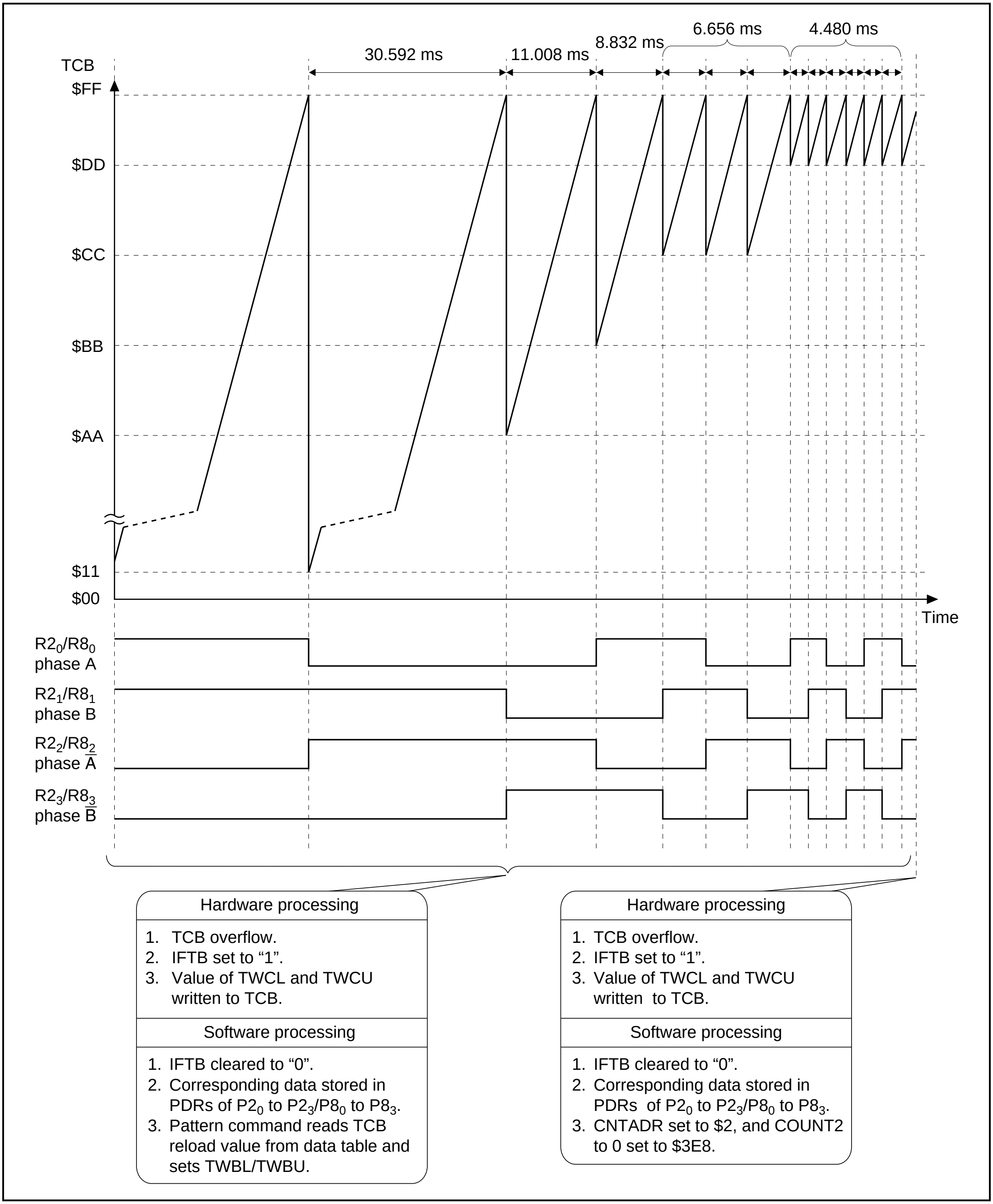


Figure 14 Operating Principle of Slue up Control When Motor Running Clockwise

4. Figure 15 shows the operating principle of Constant control when the motor is running clockwise.

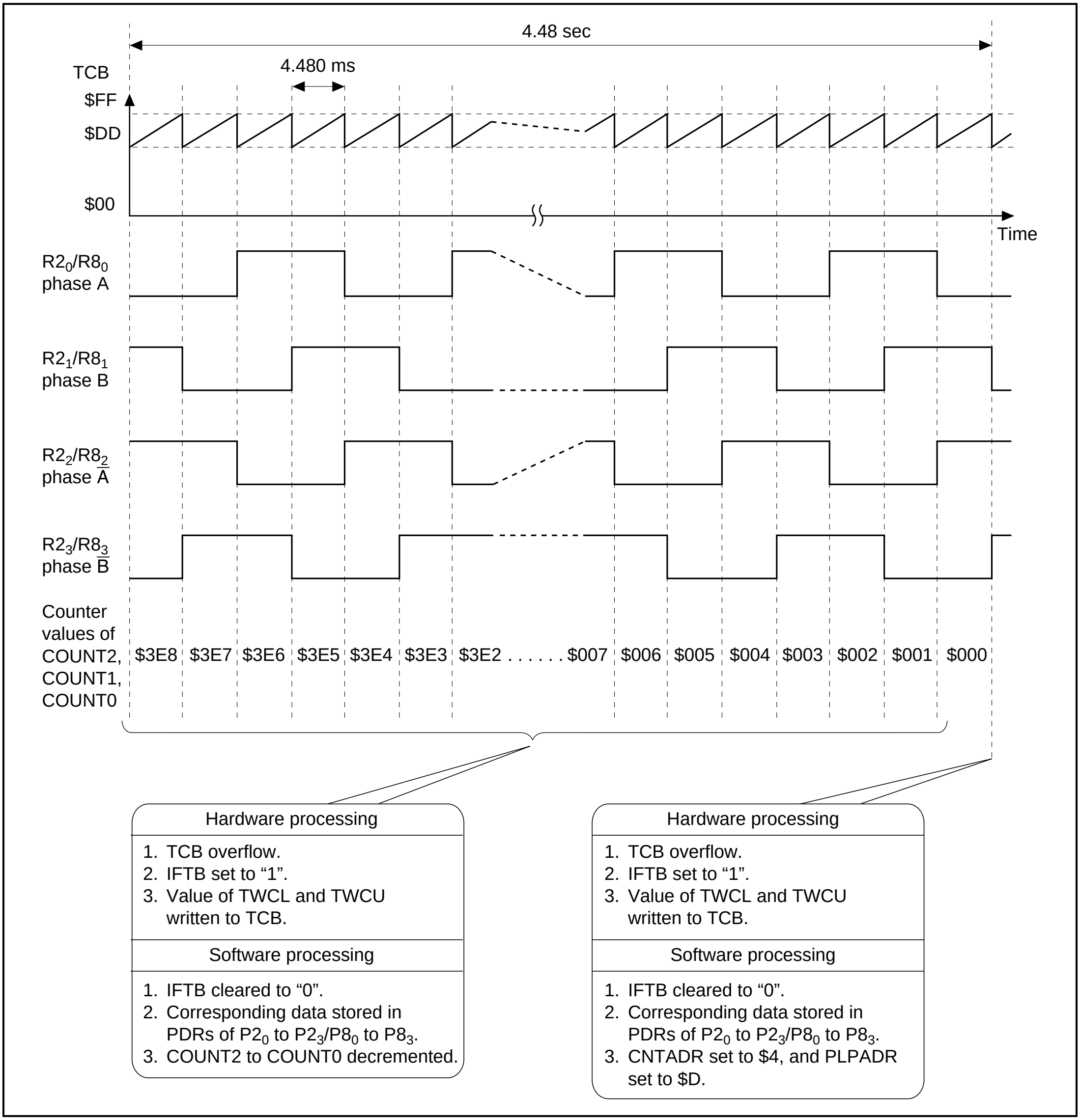


Figure 15 Operating Principle of Constant Control When Motor Running Clockwise

5. Figure 16 shows the operating principle of Slue down control when the motor is running clockwise.

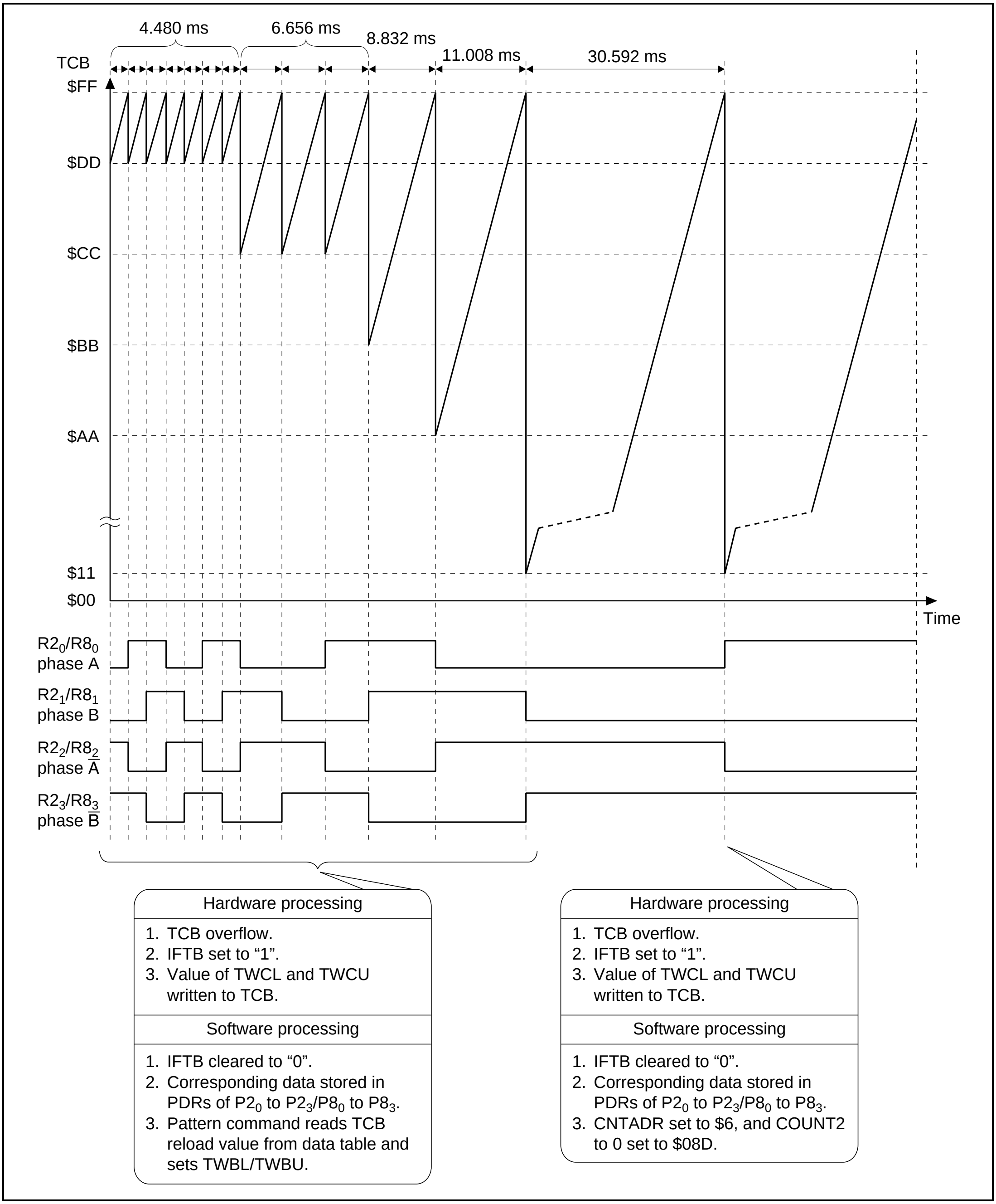


Figure 16 Operating Principle of Slue down Control When Motor Running Clockwise

6. Figure 17 shows the operating principle of Stop control when the motor is running clockwise.

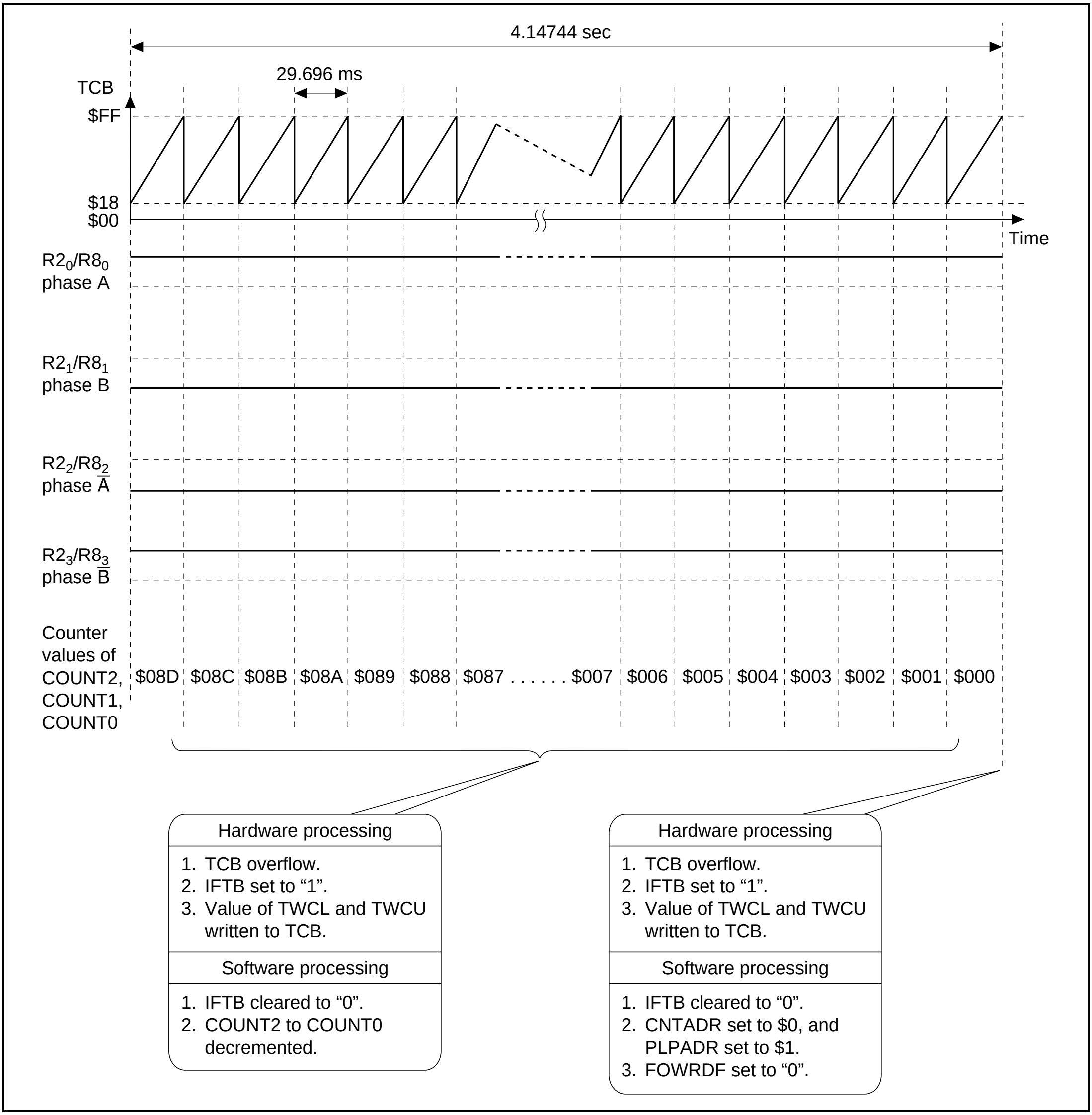


Figure 17 Operating Principle of Stop Control When Motor Running Clockwise

7. Figure 18 shows the operating principle of Slue up control when the motor is running counterclockwise.

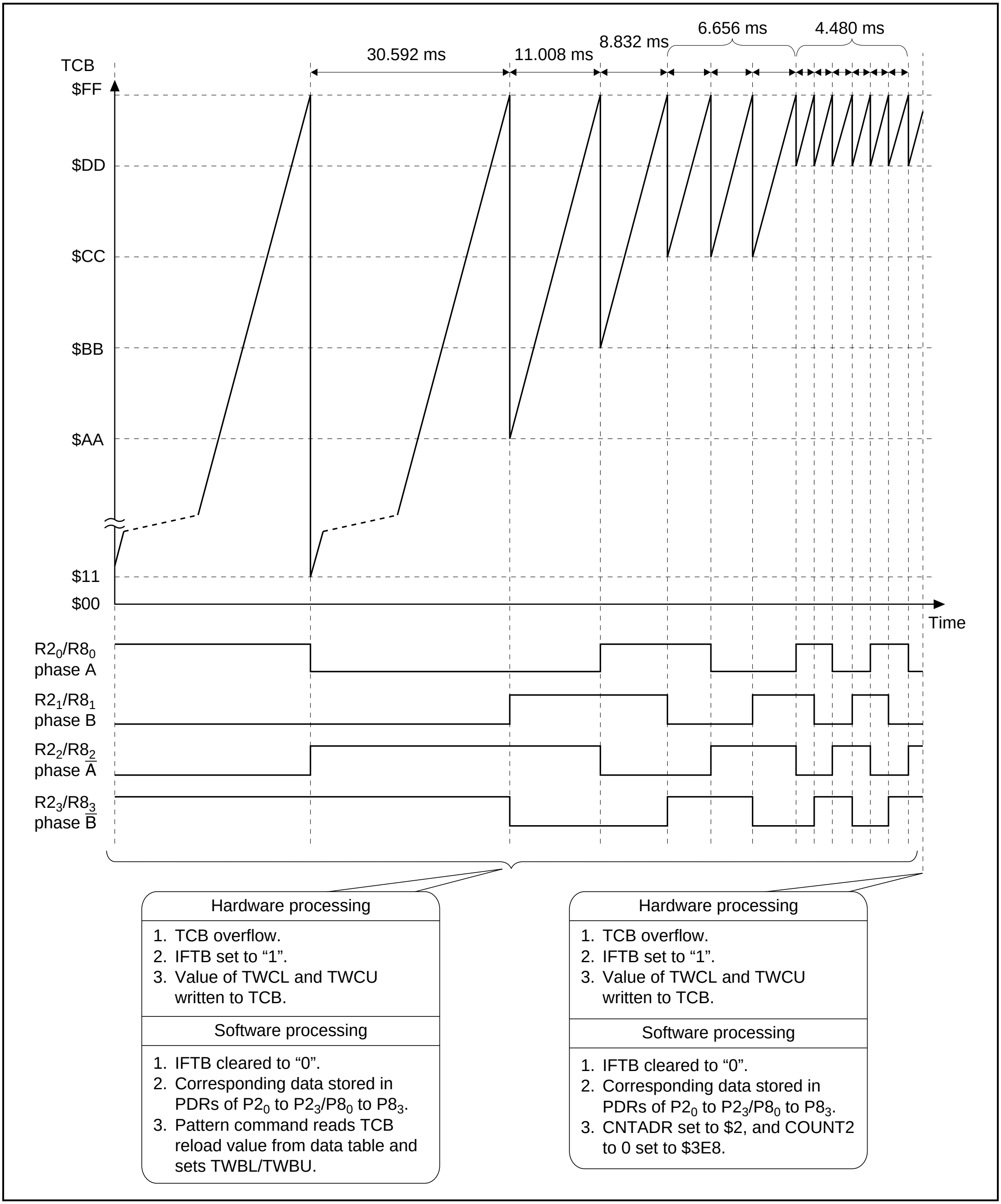


Figure 18 Operating Principle of Slue up Control When Motor Running Counterclockwise

8. Figure 19 shows the operating principle of Constant control when the motor is running counterclockwise.

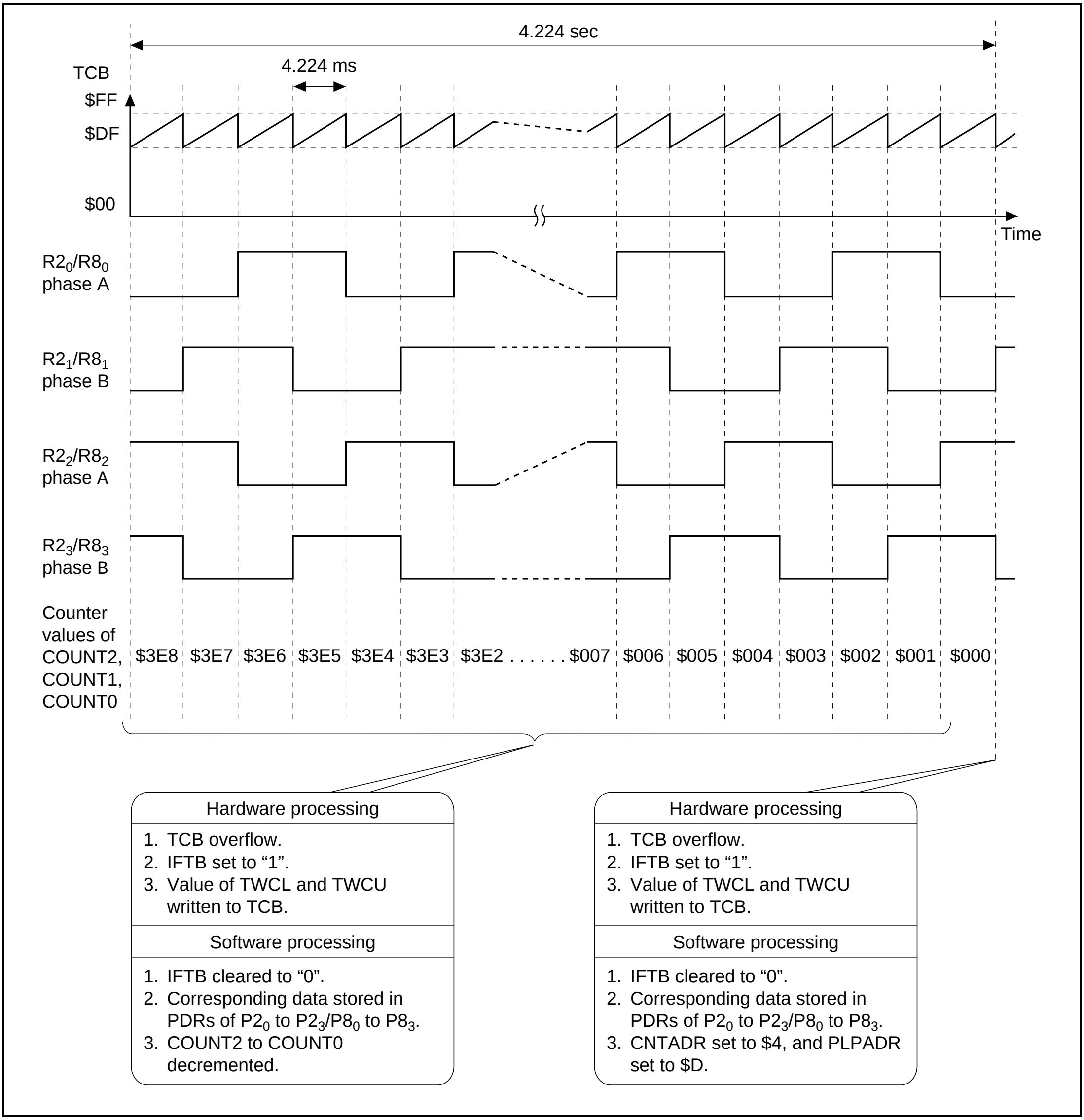


Figure 19 Operating Principle of Constant Control When Motor Running Counterclockwise

9. Figure 20 shows the operating principle of Slue down control when the motor is running counterclockwise.

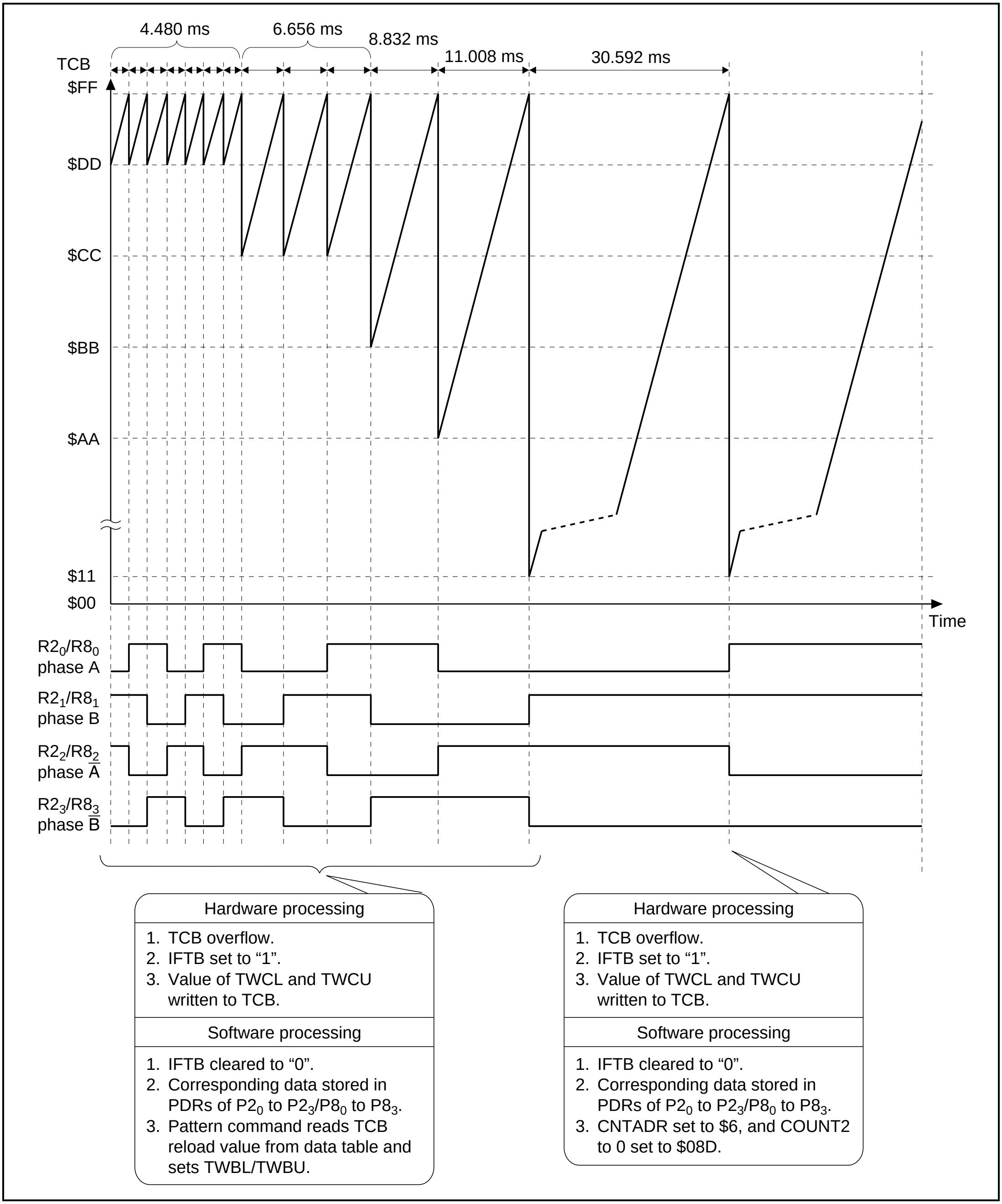


Figure 20 Operating Principle of Slue down Control When Motor Running Counterclockwise

10. Figure 21 shows the operating principle of Stop control when the motor is running counterclockwise.

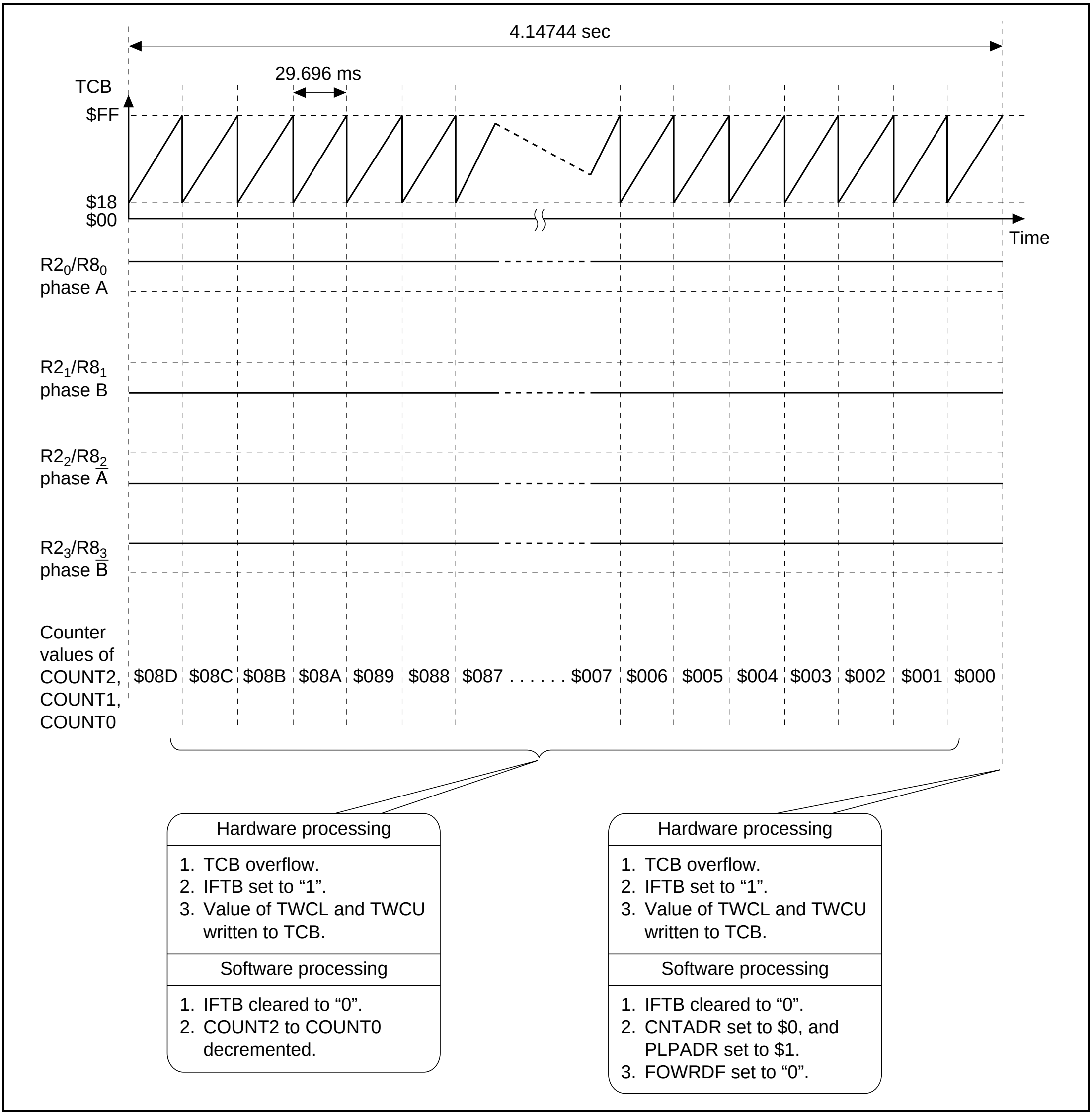


Figure 21 Operating Principle of Stop Control When Motor Running Counterclockwise

Description of Functions

1. Description of Modules

Table 5 describes the modules used in this example task.

Table 5 Description of Modules

Module	Label	Function
Main routine	STEPMN	This routine makes the initial stack pointer, RAM, I/O port, and timer B settings, performs initial stepping motor control, and enables interrupts.
Timer B interrupt processing routine	STEPCNT	This routine saves the contents of the registers, makes table branch settings, and restores the registers.
Pulse output subroutine table	PLOUT	Branches to the subroutine for pulse output (POUT).
Decrement subroutine table	DEC	Branches to the subroutine for decrementing the count (DECR).
Pulse output	POUT	Rotates the output pulse data and stores it in the PDR of the output port.
Decrement	DECR	Decrements the 12-bit counter made up of COUNT2, COUNT1, and COUNT0.
Slue up control	SLUP	Reads the timer B reload value data using the pattern command and, by increasing the output pulse cycle, performs Slue up control of the stepping motor.
Constant control	CNST	Performs Constant control of the stepping motor by maintaining the output pulse cycle at a constant rate.
Slue down control	SLDOWN	Reads the timer B reload value data using the pattern command and, by decreasing the output pulse cycle, performs Slue down control of the stepping motor.
Stop control	STOP	Performs Stop control by stopping the pulse output.

2. Description of Arguments

No arguments are used in this example task.

3. Description of Internal Registers

a. Table 6 describes the internal registers used in the H4344.

Table 6 Internal Registers Used in H4344/H4318/H4359/H4369

Register	Description	RAM Address	Setting
IE	Interrupt Enable Flag This flag controls reception of all interrupts by the CPU. • When IE = "0", CPU reception of all interrupts is disabled. • When IE = "1", CPU reception is enabled.	0, $000	1
RSP	Reset Stack Pointer Clearing RSP to "0" initializes the stack pointer.	1, $000	0
IFTB	Timer B Interrupt Request Flag Reflects the existence of a timer B interrupt request. • When IFTB = "0", no timer B interrupt is requested. • When IFTB = "1", a timer B interrupt is requested.	0, $002	0
IMTB	Timer B Interrupt Mask This bit masks IFTB. • When IMTB = "0", IFTB is enabled. • When IMTB = "1", IFTB is masked.	1, $001	1
TMB1	Timer Mode Register B1 TMB13 selects the timer B functions. TMB12 to TMB10 select the operating clock. • When TMB13 = "1", TMB12 = "0", TMB11 = "1", and TMB10 = "0", timer B functions as a reload timer, and the operating clock for timer B is set to the system clock divided by 128.	$009	$A
TWBL	Timer Write Register BL Sets the lower digit of the TCB reload value.	$00A	$8
TWBU	Timer Write Register BU Sets the upper digit of the TCB reload value.	$00B	$1

Table 6 Internal Registers Used in H4344/H4318/H4359/H4369 (cont)

Register	Description	RAM Address	Setting
TMB2	Timer Mode Register B2 Sets the input capture function and selects the detection edge for EVNB pin input. • When TMB22 = "0", timer B functions as a free-running/reload timer. • When TMB22 = "1", timer B functions as an input capture timer. Note: TMB22 is not used in the H4344. • When TMB21 = "0" and TMB20 = "0", no edge detection is performed on EVNB pin input.	$026	$0
SSR1	System Clock Selection Register 1 Sets the oscillation frequency of the system clock, sets the division for the subsystem clock frequency, and sets the subsystem clock oscillation in stop mode. • When SSR11 = "0", the system clock oscillation frequency is set to 0.4 to 1MHz. • When SSR11 = "1", the system clock oscillation frequency is set to 1.6 to 5 MHz. Note: Applies only to H4369.	$027	$2
DCR2	Data Control Register R2 Controls the ON/OFF switching of the R2 port output buffer. • When DCR23 to DCR20 = "0", the output buffers of $R2_3$ to $R2_0$ are OFF and $R2_3$ to $R2_0$ pins are set to high impedance. • When DCR23 to DCR20 = "1", the output buffers of $R2_3$ to $R2_0$ are ON and the values of the corresponding PDRs are output. Note: Applies to H4344 only.	$032	$F
DCR8	Data Control Register R8 Controls the ON/OFF switching of the R8 port output buffer. • When DCR83 to DCR80 = "0", the output buffers of $R8_3$ to $R8_0$ are OFF, and $R8_3$ to $R8_0$ pins are set to high impedance. • When DCR83 to DCR80 = "1", the output buffers of $R8_3$ to $R8_0$ are ON and the values of the corresponding PDRs are output. Note: Applies to H4318/H4359/H4369 only.	$038	$F

b. Table 7 describes the internal registers used in the H4889.

Table 7 Internal Registers Used in H4889

Register	Description	RAM Address	Setting
IE	Interrupt Enable Flag This flag controls reception of all interrupts by the CPU. • When IE = "0", CPU reception of all interrupts is disabled. • When IE = "1", CPU reception is enabled.	0, $000	1
RSP	Reset Stack Pointer Clearing RSP to "0" initializes the stack pointer.	1, $000	0
IFTB	Timer B Interrupt Request Flag Reflects the existence of a timer B interrupt request. • When IFTB = "0", no timer B interrupt is requested. • When IFTB = "1", a timer B interrupt is requested.	2, $002	0
IMTB	Timer B Interrupt Mask This bit masks IFTB. • When IMTB = "0", IFTB is enabled. • When IMTB = "1", IFTB is masked.	3, $002	1
SSR	System Clock Selection Register Selects the system clock oscillation frequency, subsystem clock frequency division, and, in stop mode, the subsystem clock oscillation and system clock frequency division ratio. • When SSR1 = "0", the system clock oscillation frequency is set to 0.4 to 1 MHz. • When SSR1 = "1", the system clock oscillation frequency is set to 1.6 to 4.5 MHz.	$004	$2
PMR3	Port Mode Register 3 Sets the functions of the $R2_2$/SI/SO pin, $R2_1$/SCK pin, and $R2_0$/TOC pin. • When PMR33 = "0", PMR32 = "0", PRM31 = "0" and PRM30 = "0", the $R2_2$/SI/SO pin functions as $R2_2$ I/O pin, $R2_1$/SCK functions as $R2_1$ I/O pin, and $R2_0$/TOC functions as $R2_0$ I/O pin.	$00B	$0

Table 7 Internal Registers Used in H4889 (cont)

Register	Description	RAM Address	Setting
TMB1	Timer Mode Register B1 TMB13 selects the timer B function, TMB12 to TMB10 select the operating clock. • When TMB13 = “1”, TMB12 = “0”, TMB11 = “1”, and TMB10 = “0”, timer B functions as a reload timer, and the timer B clock is the system clock divided by 128.	$010	$A
TMB2	Timer Mode Register B2 Sets the timer B output mode and selects the detection edge for EVNB pin input. • When TMB22 = “0”, timer B output is set to toggle waveform. • When TMB22 = “1”, timer B output is set to PWM. • When TMB21 = “0” and TMB20 = “0”, no edge detection is performed on EVNB pin input.	$011	$0
TWBL	Timer Write Register BL Sets the lower digit of the TCB reload value.	$012	$8
TWBU	Timer Write Register BU Sets the upper digit of the TCB reload value.	$013	$1
DCR8	Data Control Register 8 Controls the ON/OFF switching of the R8 port output buffer. • When DCR83 to DCR80 = “0”, the output buffers of $R8_3$ to $R8_0$ are OFF, and the output pins are set to high impedance. • When DCR83 to DCR80 = “1”, the output buffers of $R8_3$ to $R8_0$ are ON and the values of the corresponding PDRs are output.	$03C	$F

4. Description of RAM

Table 8 describes the RAM used in this example task.

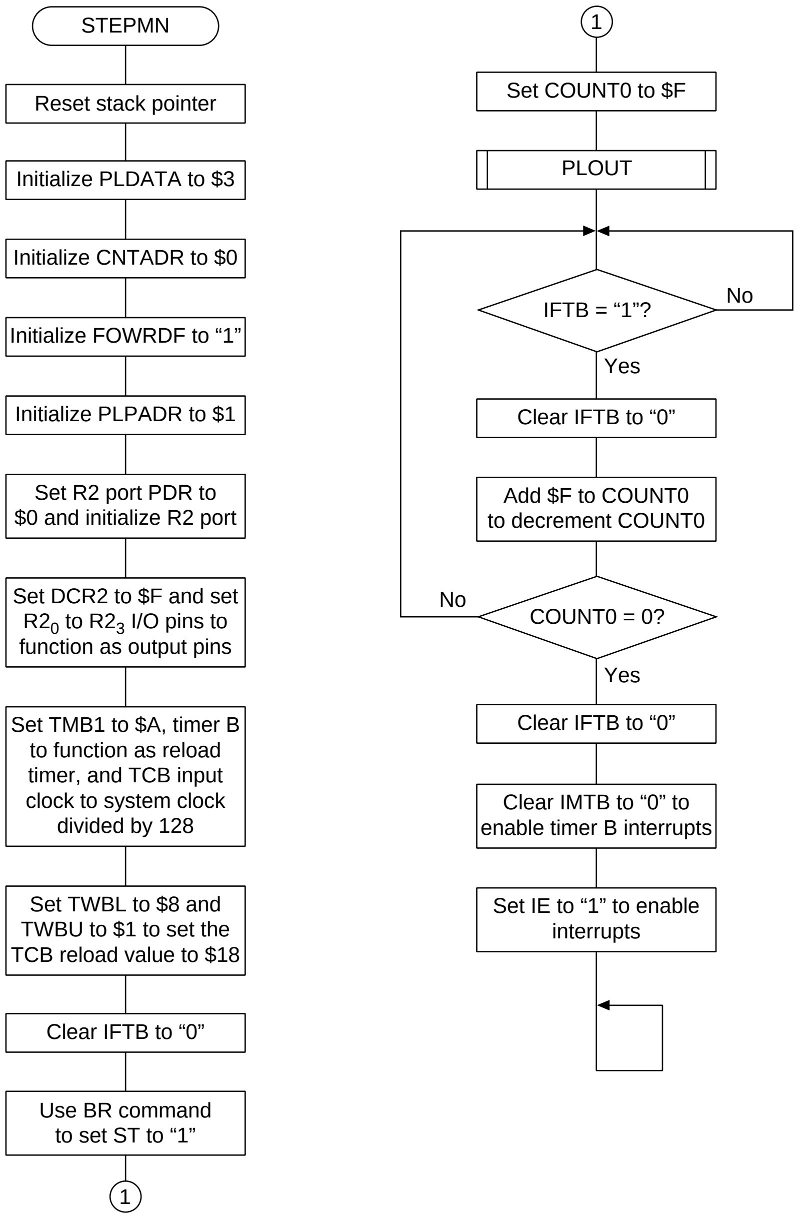
Table 8 RAM

Label	Function	RAM Address	Module
AESC	Stores the contents of the accumulator during timer B interrupt processing.	$040	STEPCNT
BESC	Stores the contents of the B register during timer B interrupt processing.	$041	STEPCNT
XESC	Stores the contents of the X register during timer B interrupt processing.	$043	STEPCNT
YESC	Stores the contents of the Y register during timer B interrupt processing.	$044	STEPCNT
PLDATA	Stores the data for the pulse output from the port.	$050	STEPMN, POUT
CNTADR	Stores the contents of the accumulator used for specifying the address when executing a table branch command.	$051	STEPMN, SLUP, CNST, SLDOWN, STOP, STEPCNT
PLPADR	Stores the contents of the accumulator used for specifying the address when executing a pattern command.	$052	STEPMN, SLUP, CNST, SLDOWN, STOP
COUNT0	Stores bits 3 to 0 of the 12-bit counter made up of COUNT0, COUNT1, COUNT2.	$053	STEPMN, SLUP, SLDOWN, DECR
COUNT1	Stores bits 7 to 4 of the 12-bit counter made up of COUNT0, COUNT1, COUNT2.	$054	SLUP, SLDOWN, DECR
COUNT2	Stores bits 11 to 8 of the 12-bit counter made up of COUNT0, COUNT1, COUNT2.	$055	SLUP, SLDOWN, DECR
FOWRDF	This flag is used to identify the direction (CW or CCW) of the stepping motor.	0, $05F	STEPMN, STOP, POUT

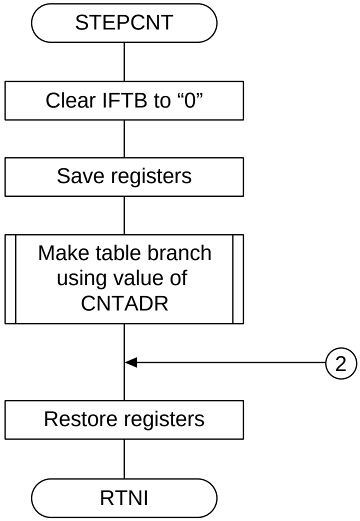
Flow charts

1. H4344

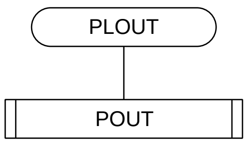
   a. Main Routine



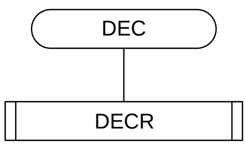
b. Timer B Interrupt Processing Routine



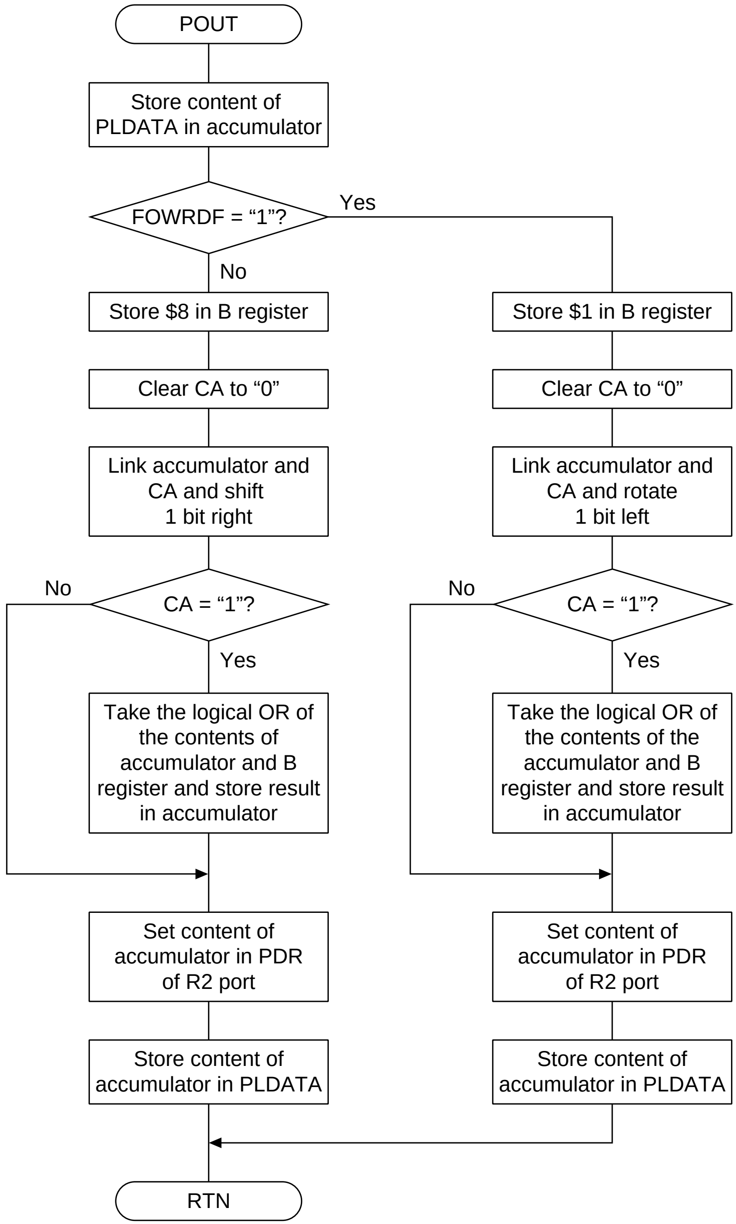
c. Pulse Output



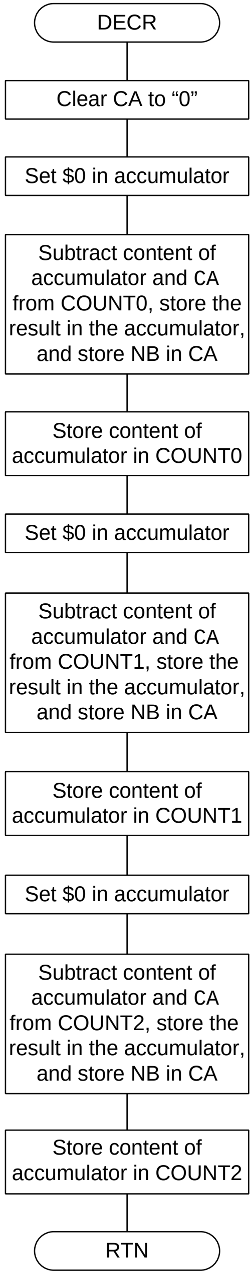
d. Decrement



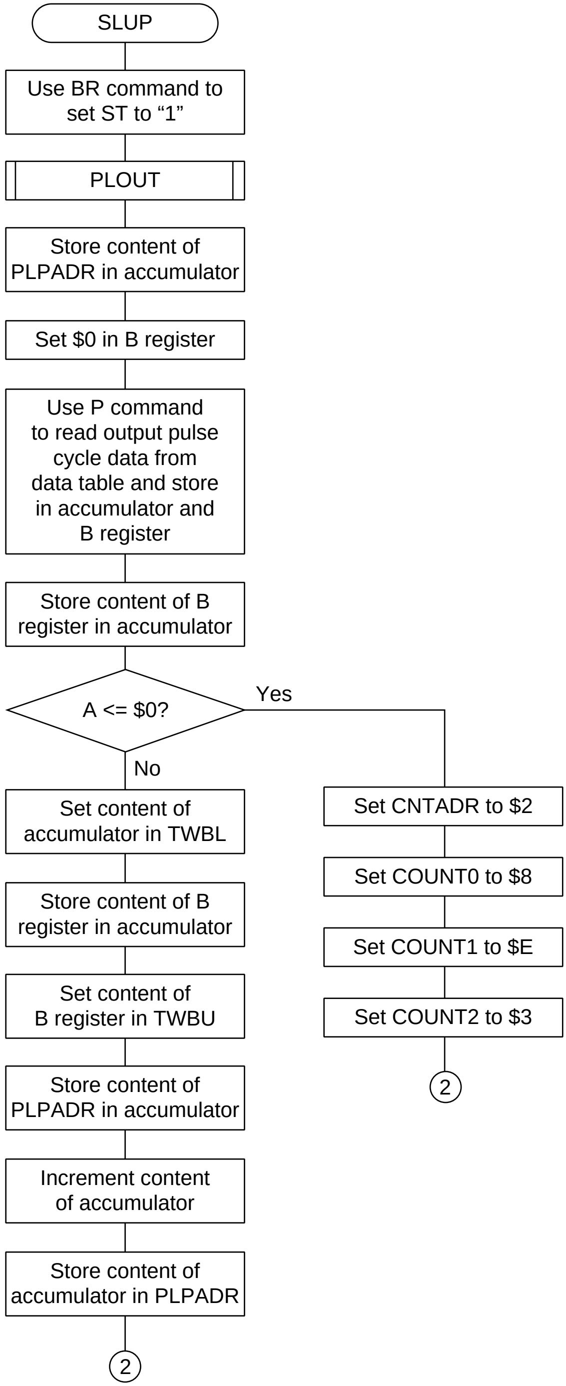
e. Pulse Output



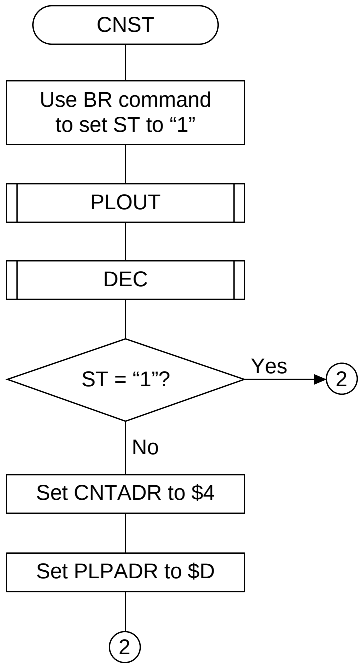
f. Decrement


DECR
Clear CA to “0”
Set $0 in accumulator
Subtract content of accumulator and CA from COUNT0, store the result in the accumulator, and store NB in CA
Store content of accumulator in COUNT0
Set $0 in accumulator
Subtract content of accumulator and CA from COUNT1, store the result in the accumulator, and store NB in CA
Store content of accumulator in COUNT1
Set $0 in accumulator
Subtract content of accumulator and CA from COUNT2, store the result in the accumulator, and store NB in CA
Store content of accumulator in COUNT2
RTN

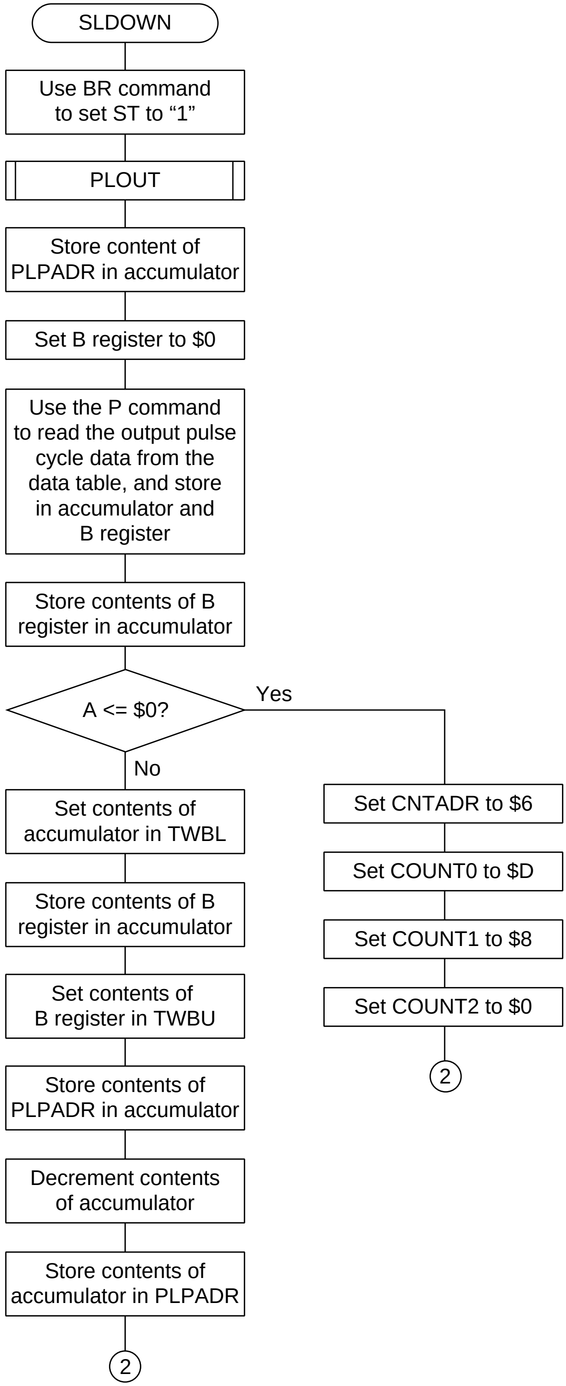
g. Slue Up Control



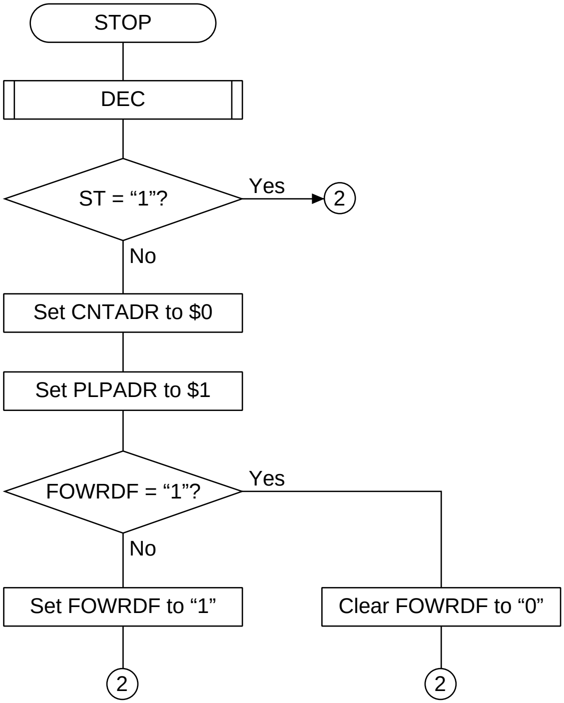
h. Constant Control



i. Slow Down Control

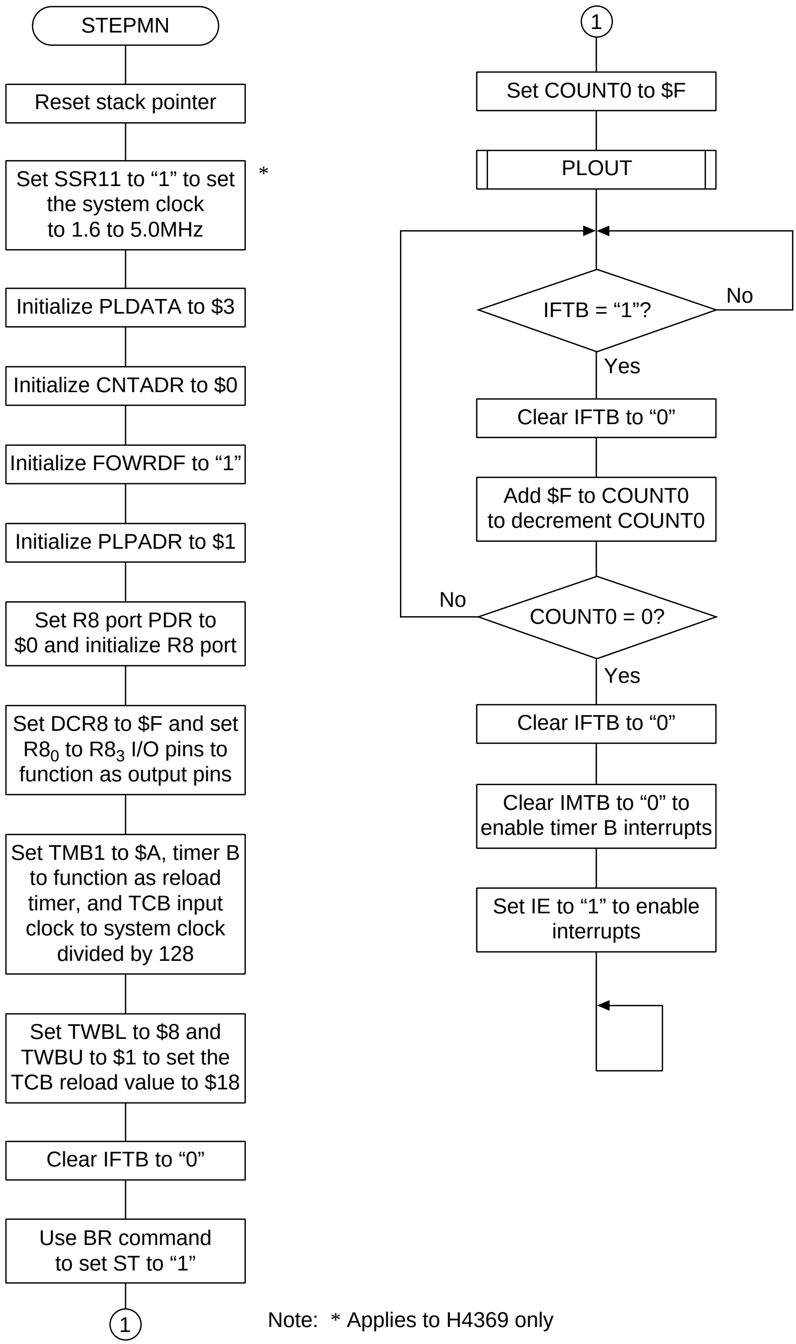


j. Stop Control



2. H4318/H4359/H4369

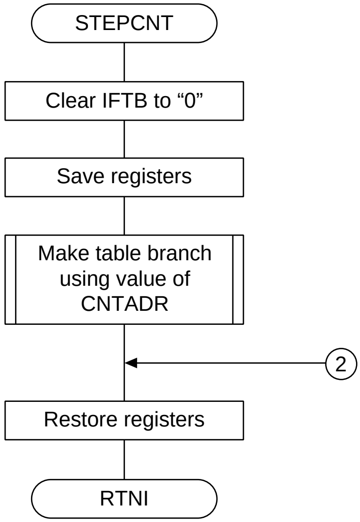
a. Main Routine



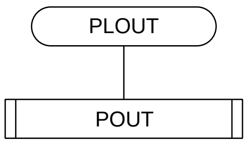
Note: * Applies to H4369 only

RENESAS

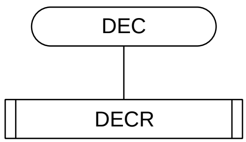
b. Timer B Interrupt Processing Routine



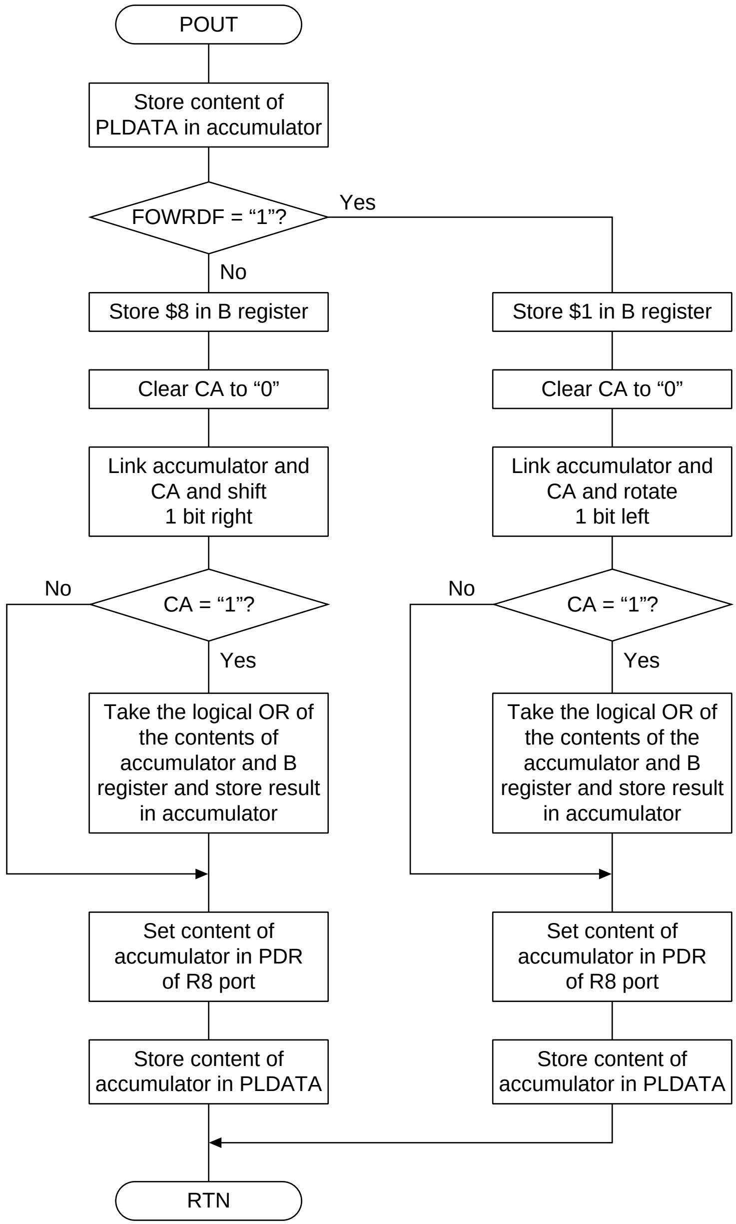
c. Pulse Output



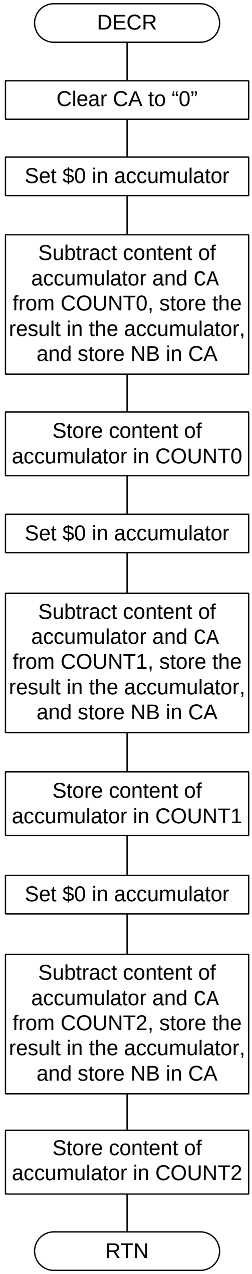
d. Decrement



e. Pulse Output

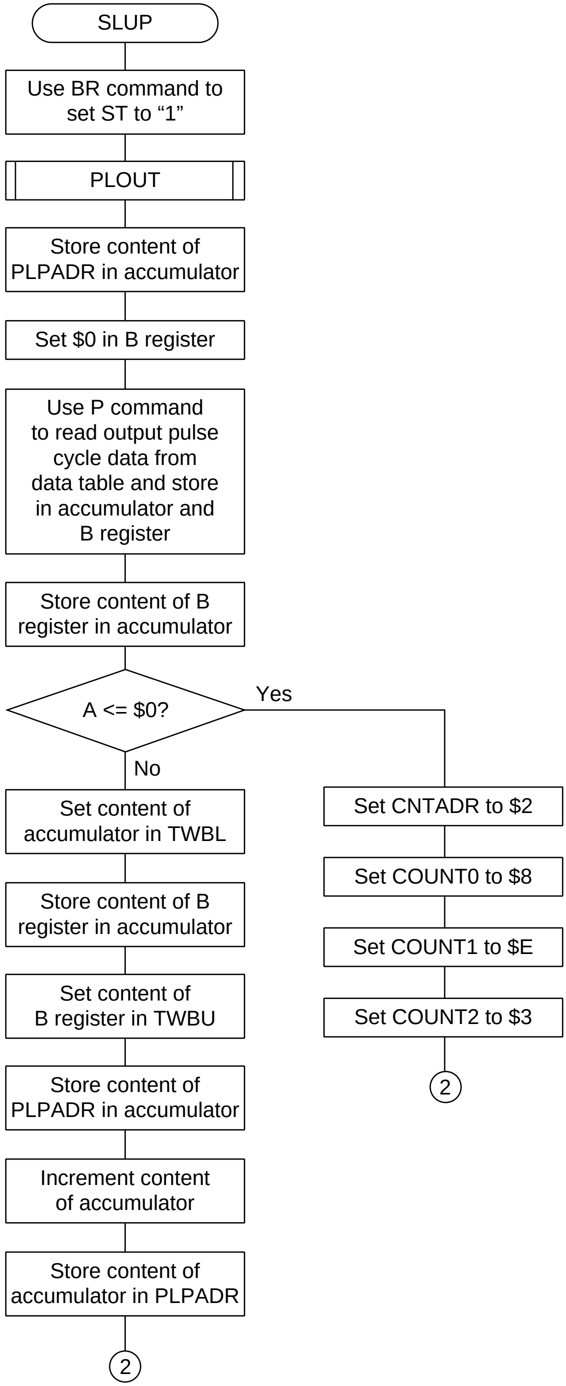


f. Decrement

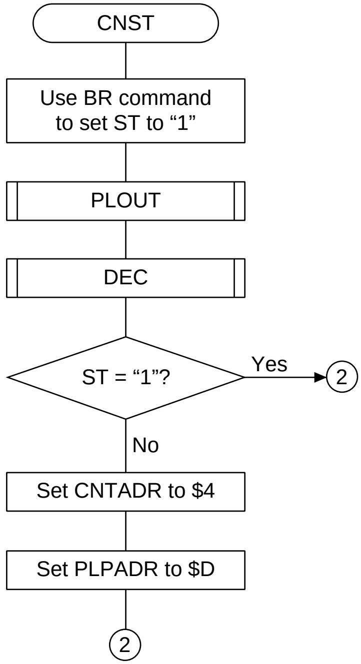


DECR
Clear CA to “0”
Set $0 in accumulator
Subtract content of accumulator and CA from COUNT0, store the result in the accumulator, and store NB in CA
Store content of accumulator in COUNT0
Set $0 in accumulator
Subtract content of accumulator and CA from COUNT1, store the result in the accumulator, and store NB in CA
Store content of accumulator in COUNT1
Set $0 in accumulator
Subtract content of accumulator and CA from COUNT2, store the result in the accumulator, and store NB in CA
Store content of accumulator in COUNT2
RTN

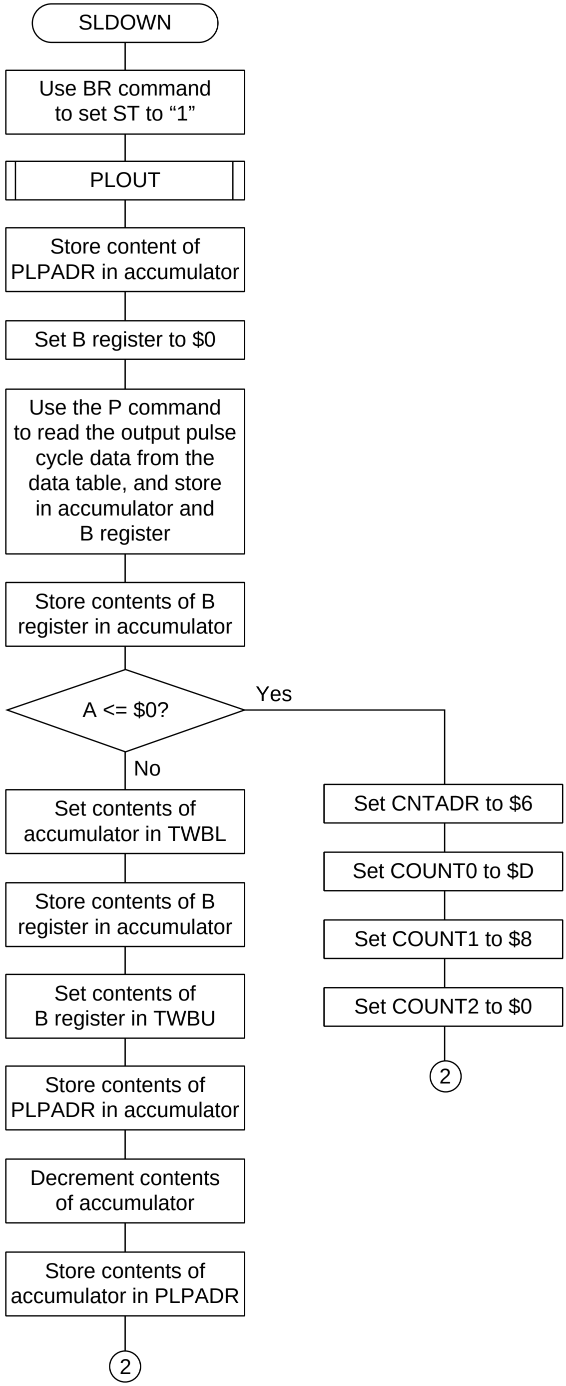
g. Slue Up Control



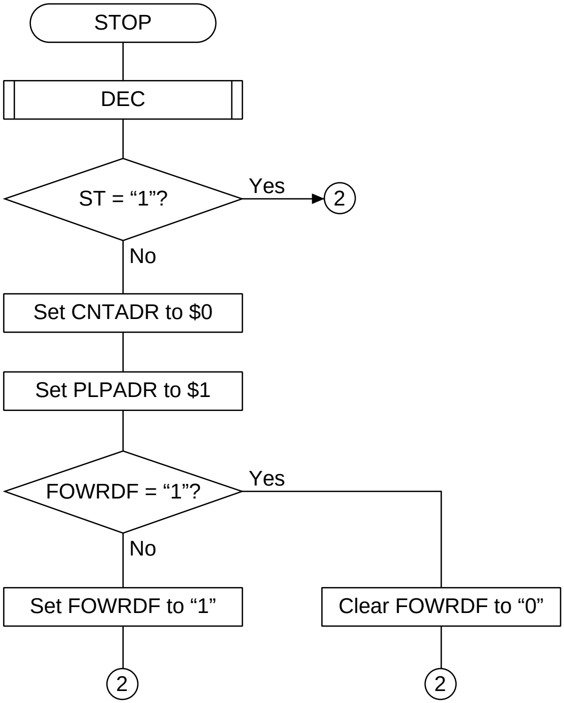
h. Constant Control



i. Slue Down Control

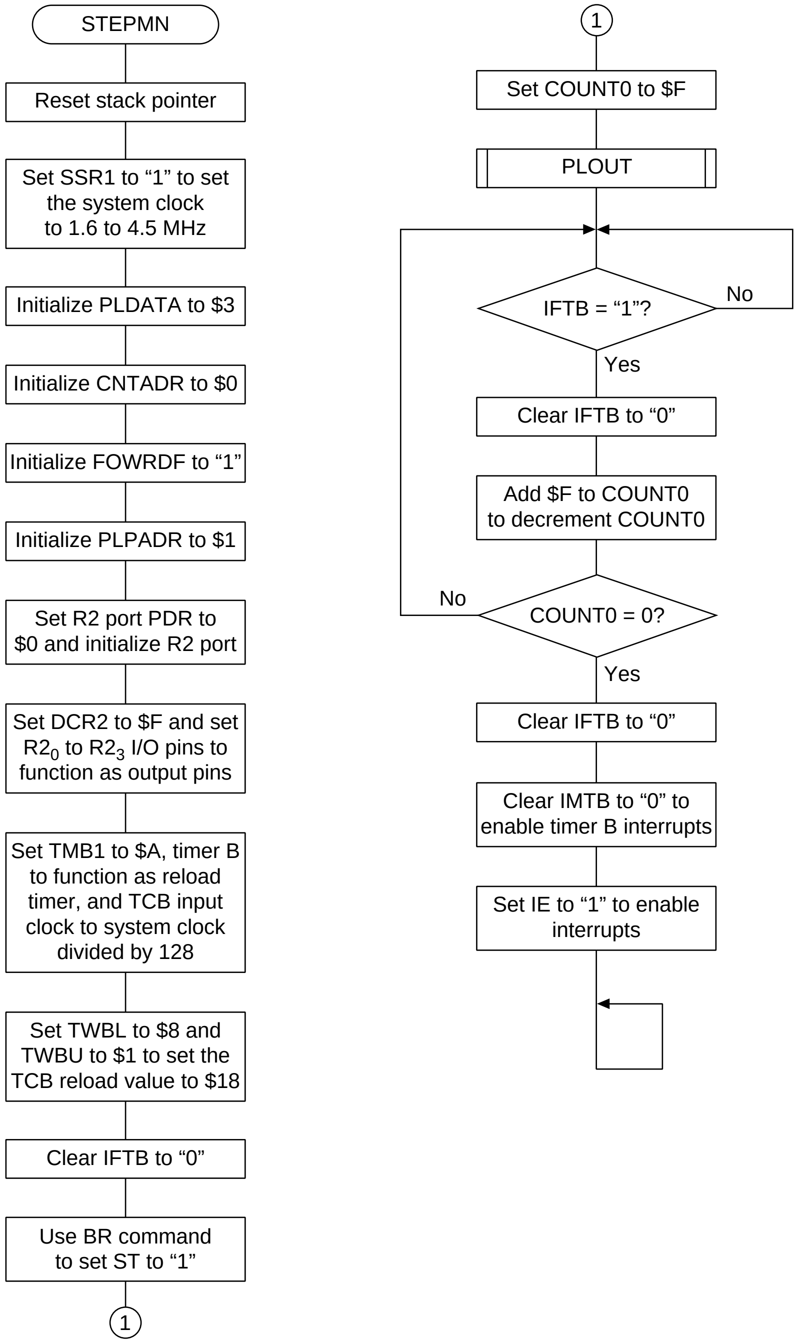


j. Stop Control

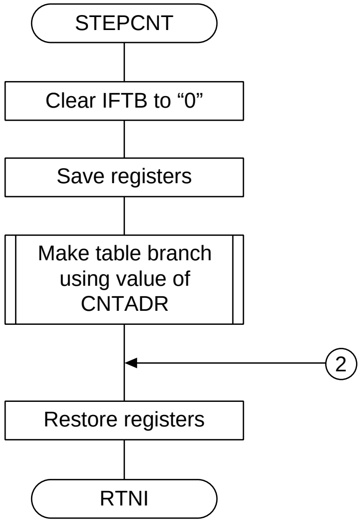


3. H4889

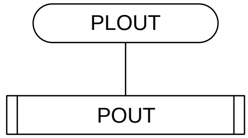
a. Main Routine



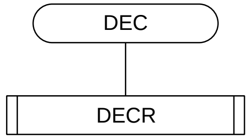
b. Timer B Interrupt Processing Routine



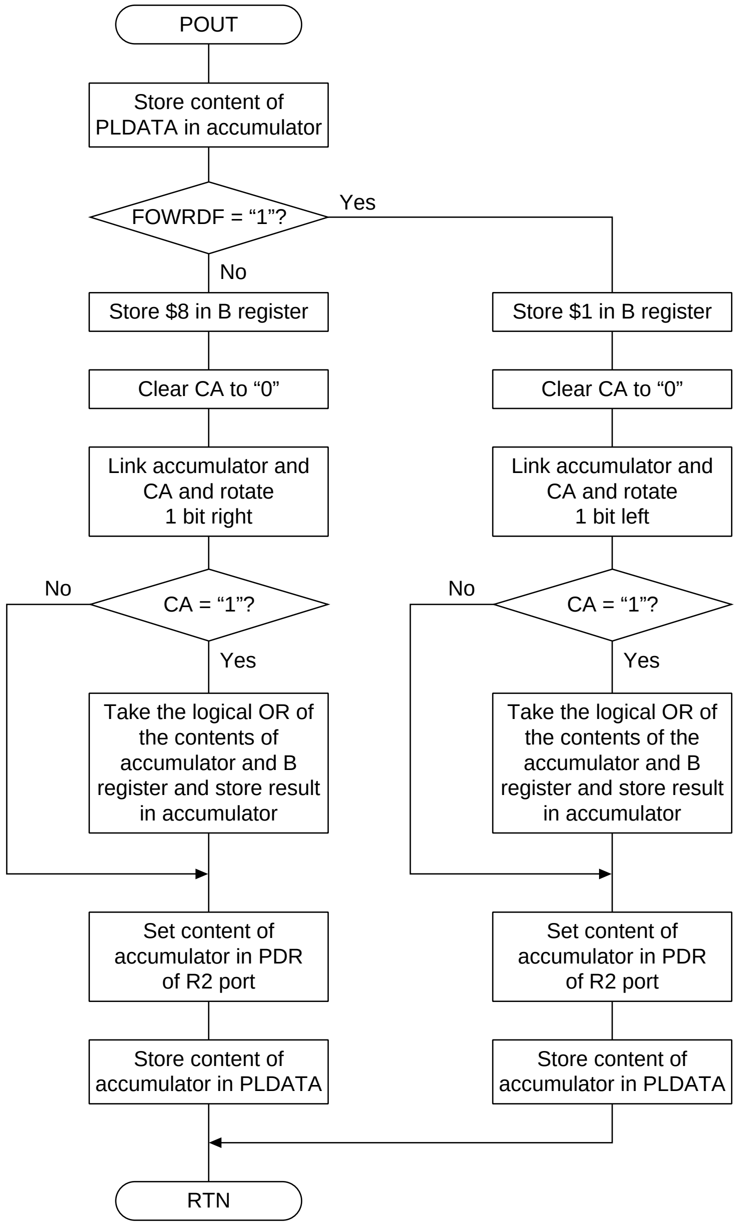
c. Pulse Output



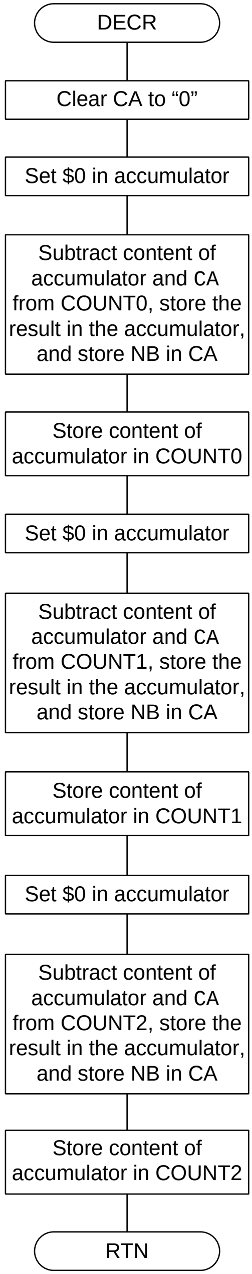
d. Decrement



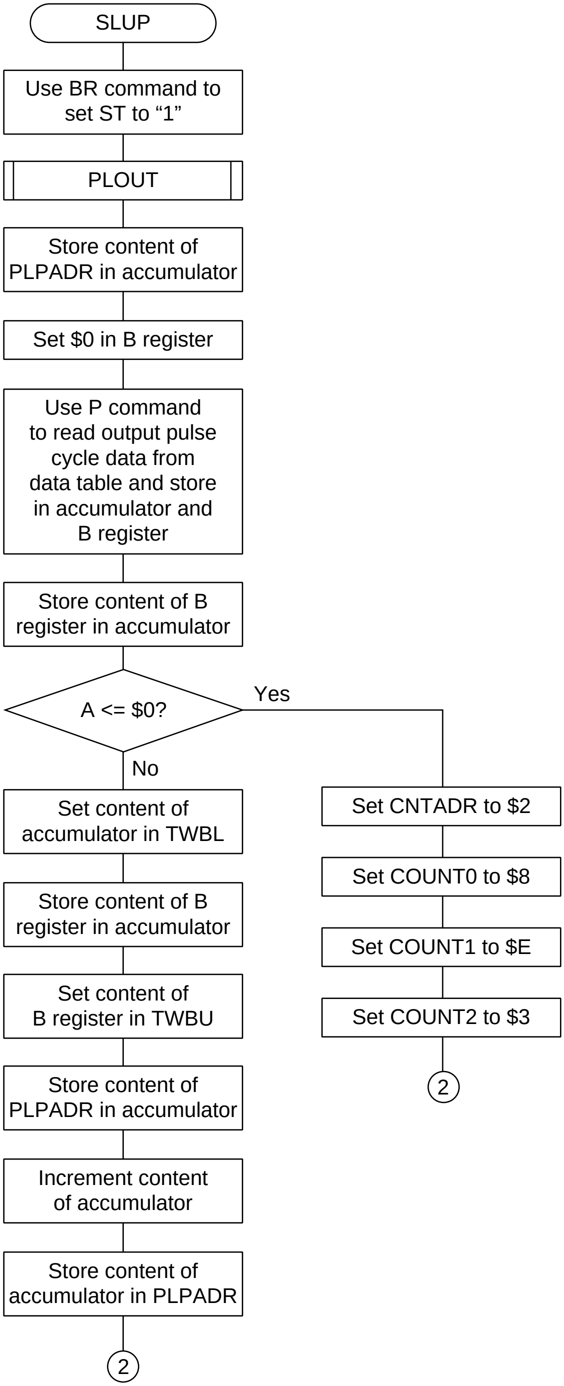
e. Pulse Output



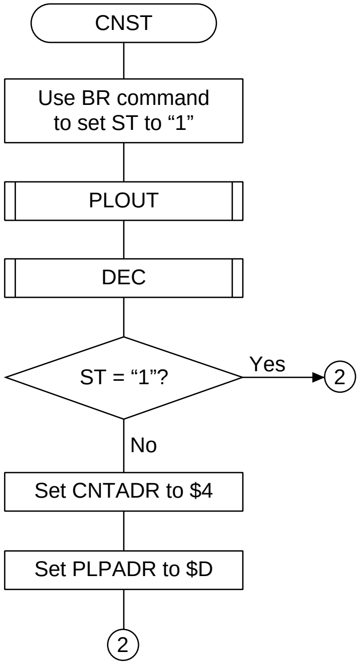
f. Decrement


DECR
Clear CA to “0”
Set $0 in accumulator
Subtract content of accumulator and CA from COUNT0, store the result in the accumulator, and store NB in CA
Store content of accumulator in COUNT0
Set $0 in accumulator
Subtract content of accumulator and CA from COUNT1, store the result in the accumulator, and store NB in CA
Store content of accumulator in COUNT1
Set $0 in accumulator
Subtract content of accumulator and CA from COUNT2, store the result in the accumulator, and store NB in CA
Store content of accumulator in COUNT2
RTN

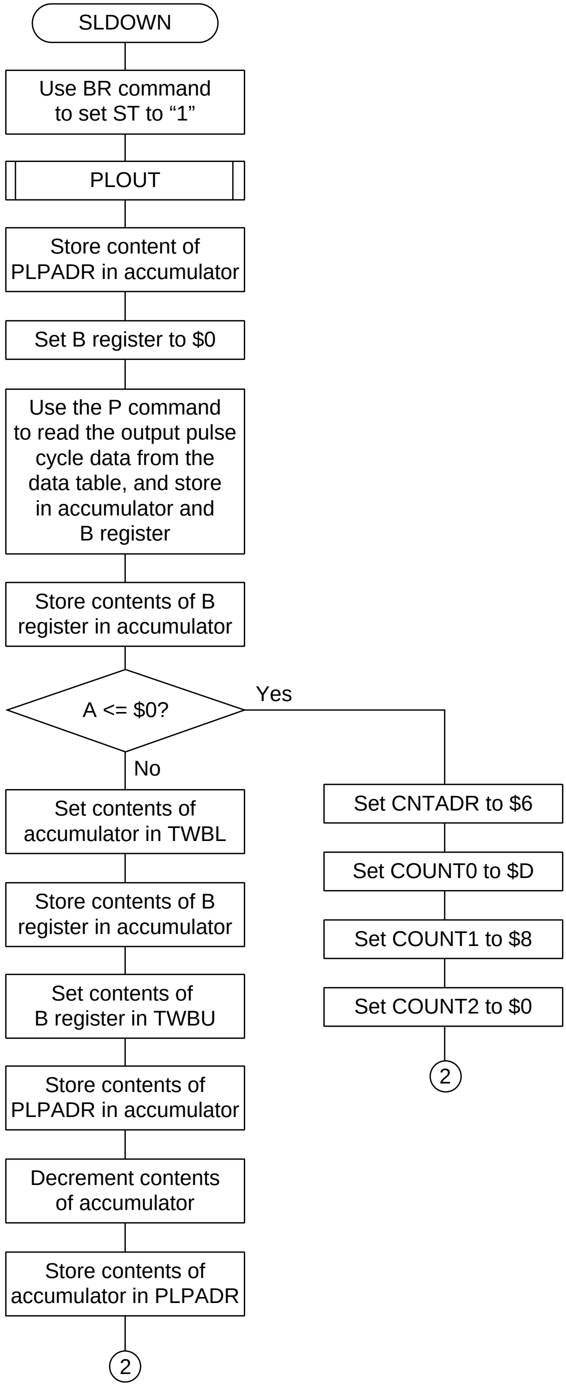
g. Slue Up Control



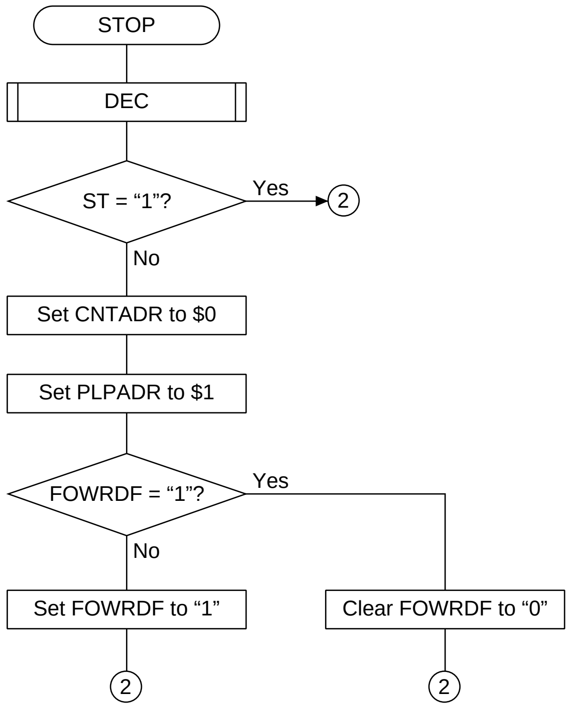
h. Constant Control



i. Slow Down Control



j. Stop Control



Program Listing

1. H4344

```
************************************************************
*
*         H400 Series Application Note
*         - Application Chapter -
*
*         'Stepping Motor Control'
*
*         Function
*         : Timer B Reload Timer
*         : I/O Port
*
*         MCU : H4344
*
*         External Clock : 4MHz
*         Internal Clock : 1MHz
*
************************************************************
*
************************************************************
*         Symbol Definition
************************************************************
*
IE        equ     0,$000     Interrupt Enable Flag
RSP       equ     1,$000     Reset Stack Pointer
IF0       equ     2,$000     INT0 Interrupt Request Flag
IM0       equ     3,$000     INT0 Interrupt Mask
*
IFTB      equ     0,$002     TIMER B Interrupt Request Flag
IMTB      equ     1,$002     TIMER B Interrupt Mask
IFTC      equ     2,$002     TIMER C Interrupt Request Flag
IMTC      equ     3,$002     TIMER C Interrupt Mask
*
IFAD      equ     0,$003     A/D Interrupt Request Flag
IMAD      equ     1,$003     A/D Interrupt Mask
IFS       equ     2,$003     SCI Interrupt Request Flag
IMS       equ     3,$003     SCI Interrupt Mask
*
PMRA      equ     $004       Port Mode Register A
SMR       equ     $005       Serial Mode Register
SRL       equ     $006       Serial Data Register L
SRU       equ     $007       Serial Data Register U
*
TMB1      equ     $009       Timer Mode Register B1
TRBL      equ     $00A       Timer Read Register BL
TWBL      equ     $00A       Timer Write register BL
TRBU      equ     $00B       Timer Read Register BU
TWBU      equ     $00B       Timer Write Register BU
```

```
MIS       equ     $00C        Miscellaneous Register
TMC       equ     $00D        Timer Mode Register C
TRCL      equ     $00E        Timer Read Register CL
TWCL      equ     $00E        Timer Write Register CL
TRCU      equ     $00F        Timer Read Register CU
TWCU      equ     $00F        Timer Write Register CU
*
ACR       equ     $016        A/D Control Register
ADRL      equ     $017        A/D Data Register L
ADRU      equ     $018        A/D Data Register U
AMR1      equ     $019        A/D Mode Register 1
AMR2      equ     $01A        A/D Mode Register 2
*
WDON      equ     1,$020      Watchdog on Flag
ADSF      equ     2,$020      A/D Start Flag
*
IAOF      equ     2,$021      IAD off Flag
RAME      equ     3,$021      RAM Enable Flag
*
PMRB      equ     $024        Port Mode register B
PMRC      equ     $025        Port Mode Register C
TMB2      equ     $026        Timer Mode Register B2
DCD0      equ     $02C        Data Control Register D0
DCD1      equ     $02D        Data Control Register D1
*
DCR0      equ     $030        Data Control Register R0
DCR1      equ     $031        Data Control Register R1
DCR2      equ     $032        Data Control Register R2
DCR3      equ     $033        Data Control Register R3
*
************************************************************
*         RAM Allocation
************************************************************
*
AESC      equ     $040        A Escape RAM Area
BESC      equ     $041        B Escape RAM Area
WESC      equ     $042        W Escape RAM Area
XESC      equ     $043        X Escape RAM Area
YESC      equ     $044        Y Escape RAM Area
SXESC     equ     $045        SPX Escape RAM Area
SYESC     equ     $046        SPY Escape RAM Area
*
PLDATA    equ     $050        Pulse Data ($3 <-> $6 <-> $C <-> $9)
CNTADR    equ     $051        Control Process Address
PLPADR    equ     $052        Pulse Period Address
COUNT0    equ     $053        Counter 0
COUNT1    equ     $054        Counter 1
COUNT2    equ     $055        Counter 2
*
FOWRDF    equ     0,$05F      Forward Flag( 1:Forward, 0:Reverce )
*
```

```
***********************************************************
*        Vector Address
***********************************************************
*
         org     $0000
*
         JMPL    STEPMN      Reset Interrupt
         JMPL    STEPMN      INT0 Interrupt
*
         org     $0008
*
         JMPL    STEPCNT     Timer B Interrupt
         JMPL    STEPMN      Timer C Interrupt
         JMPL    STEPMN      A/D Interrupt
         JMPL    STEPMN      SCI Interrupt
*
***********************************************************
*        STEPMN : Main Program
***********************************************************
*
         org     $1000
*
STEPMN   REMD    RSP         Stack Pointer Reset
*
         LMID    $3,PLDATA   Initialize Pulse Data
         LMID    $0,CNTADR   Initialize Control Process Address
         SEMD    FOWRDF      Set Forward Mode
         LMID    $1,PLPADR   Initialize Pulse Period Address
*
         LAI     $0          Initialize R2 Port Data Register
         LRA     $2
         LMID    $F,DCR2     Initialize R2 Port Output Terminal Function
*
         LMID    $A,TMB1     Initialize Timer B (128tcyc,Reload Timer ON)
         LMID    $8,TWBL     Initialize Period 29.696ms
         LMID    $1,TWBU
         REMD    IFTB        Clear Timer B Interrupt Request Flag
*
         BR      *+1         Set Status Flag
         LMID    $F,COUNT0   Initialize Counter 0
STMN10   CAL     PLOUT       Subroutine Call 'PLOUT'
*
STMN11   TMD     IFTB        29.696ms Pass ?
         BRS     STMN12      Yes. Branch to STMN12
         BRS     STMN11      No. Branch to STMN11
*
STMN12   REMD    IFTB        Clear IFTB
         LAMD    COUNT0      Load Counter 0
         AI      $F          Decrement Counter 0
         LMAD    COUNT0      Save Counter 0
         BRS     STMN10      16 times End ? No. Branch to STMN10
```

```
*
         REMD    IFTB        Clear Timer B Interrupt Request Flag
         REMD    IMTB        Timer B Interrupt Enable
         SEMD    IE          Interrupt Enable
*
STMN99   BRS     STMN99      Infinite Loop
*
************************************************************
*        STEPCNT : Stepping Motor Control
************************************************************
*
STEPCNT  REMD    IFTB        Clear Timer B Interrupt Request Flag
*
         LMAD    AESC        Store Accumulator
         LAB
         LMAD    BESC        Store B Register
         XSPX
         LASPX
         LMAD    XESC        Store X Register
         XSPX
         LAY
         LMAD    YESC        Store Y Register
*
         LAMD    CNTADR      Load Control Process Address
         LBI     $0
         TBR     $2          Table Branch (CNTADR=0:SLUP, 2:CNST, 4:SLDOWN, 6:STOP)
*
STCNT90  LAMD    YESC        Restore Y Register
         LYA
         LAMD    XESC        Restore X Register
         LXA
         LAMD    BESC        Restore B Register
         LBA
         LAMD    AESC        Restore Accumulator
*
         RTNI                Return from Interrupt
*
************************************************************
*        SLUP : Slue up Control
************************************************************
*
SLUP     BR      *+1         Set Status Flag
         CAL     PLOUT       Subroutine Jump to 'PLOUT'
         LAMD    PLPADR      Load Pulse Period Address
         LBI     $0
         P       $1          Pattern Generation 'Output Pulse Period'
         LAB
         ALEI    $0          A <= $0 ?
         BRS     SLUP10      Yes. Branch to SLUP10
         LMAD    TWBL        Set Pulse Period Data Lower
         LAB
```

```
        LMAD    TWBU        Set Pulse Period Data Upper
        LAMD    PLPADR      Load Pulse Period Address
        AI      $1          Increment Pulse Period Address
        LMAD    PLPADR      Save Pulse Period Address
        BR      *+1
        BRS     STCNT90     Branch to STCNT90
*
SLUP10  LMID    $2,CNTADR   Set Constant Control Mode
        LMID    $8,COUNT0   Initialize Counter0 to COUNT0
        LMID    $E,COUNT1   12-bit Counter = H'3E8 = D'1000
        LMID    $3,COUNT2
        BRS     STCNT90     Branch to STCNT90
*
************************************************************
*       CNST : Constant Control
************************************************************
*
CNST    BR      *+1         Set Status Flag
        CAL     PLOUT       Subroutine Jump to 'PLOUT'
*
        CAL     DEC         Subroutine Jump to 'DEC'
        BRS     STCNT90     Counter = H'000 ? Yes. Branch to STCNT90
*
        LMID    $4,CNTADR   Set Slue down Control Mode
        LMID    $D,PLPADR   Set Pulse Period Address
        BRS     STCNT90     Branch to STCNT90
*
************************************************************
*       STDOWN : Slue down Control
************************************************************
*
SLDOWN  BR      *+1         Set Status Flag
        CAL     PLOUT       Subroutine Jump to 'PLOUT'
        LAMD    PLPADR      Load Pulse Period Address
        LBI     $0
        P       $1          Pattern Generation 'Output Pulse Period'
        LAB
        ALEI    $0          A <= $0 ?
        BRS     SLDW10      Yes. Branch to SLDW10
        LMAD    TWBL        Set Pulse Period Data Lower
        LAB
        LMAD    TWBU        Set Pulse Period Data Upper
        LAMD    PLPADR      Load Pulse Period Address
        AI      $F          Decrement Pulse Period Address
        LMAD    PLPADR      Save Pulse Period Address
        BR      *+1
        BRS     STCNT90     Branch to STCNT90
*
SLDW10  LMID    $6,CNTADR   Set Stop Control Mode
        LMID    $D,COUNT0   Initialize COUNT0 to COUNT2
        LMID    $8,COUNT1   12-bit Counter = H'08D = D'141
```

```
          LMID    $0,COUNT2
          BRS     STCNT90     Branch to STCNT90
*
************************************************************
*         STOP : Stop Control
************************************************************
*
STOP      CAL     DEC         Subroutine Jump to 'DEC'
          BRS     STCNT90     ST = "1"? Yes. Branch to STCNT9
*
          LMID    $0,CNTADR   No. Set Slue up Control Mode
          LMID    $1,PLPADR   Set Pulse period Address
          TMD     FOWRDF      FOWRDF = "1" ?
          BRS     STOP10      Yes. Branch to STOP10
          SEMD    FOWRDF      Set FOWRDF
          BRS     STCNT90     Branch to STCNT90
STOP10    REMD    FOWRDF      Reset FOWRDF
          BRS     STCNT90     Branch to STCNT90
*
************************************************************
*         POUT : Output Pulse
************************************************************
*
POUT      LAMD    PLDATA      Load Pulse Data
          TMD     FOWRDF      FOWRDF = 1 ?
          BRS     POT20       Yes. Branch to POT20
*
          LBI     $8          No.
          REC                 Reset CA
          ROTR                Rotate Right A with Carry
          TC                  CA = 1 ?
          BRS     POT10       Yes. Branch to POT10
          BRS     POT11       No. Branch to POT11
POT10     OR                  A = A OR B
POT11     LRA     $2          Set R2 Port PDR
          LMAD    PLDATA      Save Pulse Data
          BRS     POT99       Branch to POT99
*
POT20     LBI     $1
          REC                 Reset CA
          ROTL                Rotate Left A with Carry
          TC                  CA = 1 ?
          BRS     POT21       Yes. Branch to POT21
          BRS     POT22       No. Branch to POT22
POT21     OR                  A = A OR B
POT22     LRA     $2          Set R2 Port PDR
          LMAD    PLDATA      Save Pulse Data
*
POT99     RTN                 Return from Subroutine
*
************************************************************
```

```
*         DECR : Decrement Counter
************************************************************
*
DECR      REC                   Reset Carry
          LAI     $0
          SMCD    COUNT0        A = COUNT0 - A - _CA
          LMAD    COUNT0        Save COUNT0
          LAI     $0
          SMCD    COUNT1        A = COUNT1 - A - _CA
          LMAD    COUNT1        Save COUNT1
          LAI     $0
          SMCD    COUNT2        A = COUNT2 - A - _CA
          LMAD    COUNT2        Save COUNT2
          RTN                   Return from Subroutine
*
************************************************************
*         Subroutine Table
************************************************************
*
          org     $20
*
PLOUT     BRL     POUT          Branch to POUT
DEC       BRL     DECR          Branch to DECR
*
************************************************************
*         Timer Period Data Table
************************************************************
*
          org     $100
*
          dc      $100          Start
          dc      $111          30.592ms
          dc      $1AA          11.008ms
          dc      $1BB          8.832ms
          dc      $1CC          6.656ms
          dc      $1CC          6.656ms
          dc      $1CC          6.656ms
          dc      $1DD          4.480ms
          dc      $1DD          4.480ms
          dc      $1DD          4.480ms
          dc      $1DD          4.480ms
          dc      $1DD          4.480ms
          dc      $1DD          4.480ms
          dc      $1DD          4.480ms
          dc      $100          End
*
************************************************************
*         Pulse Motor Control Address
************************************************************
*
          org     $200
```

```
*
        JMPL    SLUP        Jump to Slue up Control Routine
        JMPL    CNST        Jump to Constant Control Routine
        JMPL    SLDOWN      Jump to Slue down Control Routine
        JMPL    STOP        Jump to Stop Control Routine
*
        end
```

```
************************************************************
*
*         H400 Series Application Note
*         - Application Chapter -
*
*         'Stepping Motor Control'
*
*         Function
*         : Timer B Reload Timer
*         : I/O Port
*
*         MCU : H4318/H4359
*
*         External Clock : 4MHz
*         Internal Clock : 1MHz
*
************************************************************
*
************************************************************
*         Symbol Definition
************************************************************
*
IE        equ     0,$000      Interrupt Enable Flag
RSP       equ     1,$000      Reset Stack Pointer
IF0       equ     2,$000      INT0 Interrupt Request Flag
IM0       equ     3,$000      INT0 Interrupt Mask
*
IF1       equ     0,$001      INT1 Interrupt Request Flag
IM1       equ     1,$001      INT1 Interrupt Mask
IFTA      equ     2,$001      TIMER A Interrupt Request Flag
IMTA      equ     3,$001      TIMER A Interrupt Mask
*
IFTB      equ     0,$002      TIMER B Interrupt Request Flag
IMTB      equ     1,$002      TIMER B Interrupt Mask
IFTC      equ     2,$002      TIMER C Interrupt Request Flag
IMTC      equ     3,$002      TIMER C Interrupt Mask
*
IFAD      equ     0,$003      A/D Interrupt Request Flag
IMAD      equ     1,$003      A/D Interrupt Mask
IFS       equ     2,$003      SCI Interrupt Request Flag
IMS       equ     3,$003      SCI Interrupt Mask
*
PMRA      equ     $004        Port Mode Register A
SMR       equ     $005        Serial Mode Register
SRL       equ     $006        Serial Data Register L
SRU       equ     $007        Serial Data Register U
TMA       equ     $008        Timer Mode Register A
TMB1      equ     $009        Timer Mode Register B1
TRBL      equ     $00A        Timer Read Register BL
```

```
TWBL      equ     $00A          Timer Write register BL
TRBU      equ     $00B          Timer Read Register BU
TWBU      equ     $00B          Timer Write Register BU
MIS       equ     $00C          Miscellaneous Register
TMC       equ     $00D          Timer Mode Register C
TRCL      equ     $00E          Timer Read Register CL
TWCL      equ     $00E          Timer Write Register CL
TRCU      equ     $00F          Timer Read Register CU
TWCU      equ     $00F          Timer Write Register CU
ACR       equ     $016          A/D Control Register
ADRL      equ     $017          A/D Data Register L
ADRU      equ     $018          A/D Data Register U
AMR1      equ     $019          A/D Mode Register 1
AMR2      equ     $01A          A/D Mode Register 2
*
WDON      equ     1,$020        Watchdog on Flag
ADSF      equ     2,$020        A/D Start Flag
*
ICSF      equ     0,$021        Input Capture Status Flag
ICEF      equ     1,$021        Input Capture Error Flag
IAOF      equ     2,$021        IAD off Flag
RAME      equ     3,$021        RAM Enable Flag
*
PMRB      equ     $024          Port Mode register B
PMRC      equ     $025          Port Mode Register C
TMB2      equ     $026          Timer Mode Register B2
DCD0      equ     $02C          Data Control Register D0
DCD1      equ     $02D          Data Control Register D1
DCD2      equ     $02E          Data Control Register D2
DCR0      equ     $030          Data Control Register R0
DCR1      equ     $031          Data Control Register R1
DCR2      equ     $032          Data Control Register R2
DCR3      equ     $033          Data Control Register R3
DCR4      equ     $034          Data Control Register R4
DCR8      equ     $038          Data Control Register R8
*
************************************************************
*         RAM Allocation
************************************************************
*
AESC      equ     $040          A Escape RAM Area
BESC      equ     $041          B Escape RAM Area
WESC      equ     $042          W Escape RAM Area
XESC      equ     $043          X Escape RAM Area
YESC      equ     $044          Y Escape RAM Area
SXESC     equ     $045          SPX Escape RAM Area
SYESC     equ     $046          SPY Escape RAM Area
*
PLDATA    equ     $050          Pulse Data ($3 <-> $6 <-> $C <-> $9)
CNTADR    equ     $051          Control Process Address
PLPADR    equ     $052          Pulse Period Address
```

```
COUNT0   equ     $053        Counter 0
COUNT1   equ     $054        Counter 1
COUNT2   equ     $055        Counter 2
*
FOWRDF   equ     0,$05F      Forward Flag( 1:Forward, 0:Reverce )
*
************************************************************
*        Vector Address
************************************************************
*
         org     $0000
*
         JMPL    STEPMN      Reset Interrupt
         JMPL    STEPMN      INT0 Interrupt
         JMPL    STEPMN      INT1 Interrupt
         JMPL    STEPMN      Timer A Interrupt
         JMPL    STEPCNT     Timer B Interrupt
         JMPL    STEPMN      Timer C Interrupt
         JMPL    STEPMN      A/D Interrupt
         JMPL    STEPMN      SCI Interrupt
*
************************************************************
*        STEPMN : Main Program
************************************************************
*
         org     $1000
*
STEPMN   REMD    RSP         Stack Pointer Reset
*
         LMID    $3,PLDATA   Initialize Pulse Data
         LMID    $0,CNTADR   Initialize Control Process Address
         SEMD    FOWRDF      Set Forward Mode
         LMID    $1,PLPADR   Initialize Pulse Period Address
*
         LAI     $0          Initialize R8 Port Data Register
         LRA     $8
         LMID    $F,DCR8     Initialize R8 Port Output Terminal Function
*
         LMID    $A,TMB1     Initialize Timer B (128tcyc,Reload Timer ON)
         LMID    $8,TWBL     Initialize Period 29.696ms
         LMID    $1,TWBU
         REMD    IFTB        Clear Timer B Interrupt Request Flag
*
         BR      *+1         Set Status Flag
         LMID    $F,COUNT0   Initialize Counter 0
STMN10   CAL     PLOUT       Subroutine Call 'PLOUT'
*
STMN11   TMD     IFTB        29.696ms Pass ?
         BRS     STMN12      Yes. Branch to STMN12
         BRS     STMN11      No. Branch to STMN11
*
```

```
STMN12   REMD    IFTB        Clear IFTB
         LAMD    COUNT0      Load Counter 0
         AI      $F          Decrement Counter 0
         LMAD    COUNT0      Save Counter 0
         BRS     STMN10      16 times End ? No. Branch to STMN10
*
         REMD    IFTB        Clear Timer B Interrupt Request Flag
         REMD    IMTB        Timer B Interrupt Enable
         SEMD    IE          Interrupt Enable
*
STMN99   BRS     STMN99      Infinite Loop
*
************************************************************
*        STEPCNT : Stepping Motor Control
************************************************************
*
STEPCNT  REMD    IFTB        Clear Timer B Interrupt Request Flag
*
         LMAD    AESC        Store Accumulator
         LAB
         LMAD    BESC        Store B Register
         XSPX
         LASPX
         LMAD    XESC        Store X Register
         XSPX
         LAY
         LMAD    YESC        Store Y Register
*
         LAMD    CNTADR      Load Control Process Address
         LBI     $0
         TBR     $2          Table Branch (CNTADR=0:SLUP, 2:CNST, 4:SLDOWN, 6:STOP)
*
STCNT90  LAMD    YESC        Restore Y Register
         LYA
         LAMD    XESC        Restore X Register
         LXA
         LAMD    BESC        Restore B Register
         LBA
         LAMD    AESC        Restore Accumulator
*
         RTNI                Return from Interrupt
*
************************************************************
*        SLUP : Slue up Control
************************************************************
*
SLUP     BR      *+1         Set Status Flag
         CAL     PLOUT       Subroutine Jump to 'PLOUT'
         LAMD    PLPADR      Load Pulse Period Address
         LBI     $0
         P       $1          Paturn Generation 'Output Pulse Period'
```

```
         LAB
         ALEI    $0          A <= $0 ?
         BRS     SLUP10      Yes. Branch to SLUP10
         LMAD    TWBL        Set Pulse Period Data Lower
         LAB
         LMAD    TWBU        Set Pulse Period Data Upper
         LAMD    PLPADR      Load Pulse Period Address
         AI      $1          Increment Pulse Period Address
         LMAD    PLPADR      Save Pulse Period Address
         BR      *+1
         BRS     STCNT90     Branch to STCNT90
*
SLUP10   LMID    $2,CNTADR   Set Constant Control Mode
         LMID    $8,COUNT0   Initialize Counter0 to COUNT0
         LMID    $E,COUNT1   12-bit Counter = H'3E8 = D'1000
         LMID    $3,COUNT2
         BRS     STCNT90     Branch to STCNT90
*
************************************************************
*        CNST : Constant Control
************************************************************
*
CNST     BR      *+1         Set Status Flag
         CAL     PLOUT       Subroutine Jump to 'PLOUT'
*
         CAL     DEC         Subroutine Jump to 'DEC'
         BRS     STCNT90     Counter = H'000 ? Yes. Branch to STCNT90
*
         LMID    $4,CNTADR   Set Slue down Control Mode
         LMID    $D,PLPADR   Set Pulse Period Address
         BRS     STCNT90     Branch to STCNT90
*
************************************************************
*        STDOWN : Slue down Control
************************************************************
*
SLDOWN   BR      *+1         Set Status Flag
         CAL     PLOUT       Subroutine Jump to 'PLOUT'
         LAMD    PLPADR      Load Pulse Period Address
         LBI     $0
         P       $1          Pattern Generation 'Output Pulse Period'
         LAB
         ALEI    $0          A <= $0 ?
         BRS     SLDW10      Yes. Branch to SLDW10
         LMAD    TWBL        Set Pulse Period Data Lower
         LAB
         LMAD    TWBU        Set Pulse Period Data Upper
         LAMD    PLPADR      Load Pulse Period Address
         AI      $F          Decrement Pulse Period Address
         LMAD    PLPADR      Save Pulse Period Address
         BR      *+1
```

```
        BRS     STCNT90      Branch to STCNT90
*
SLDW10  LMID    $6,CNTADR    Set Stop Control Mode
        LMID    $D,COUNT0    Initialize COUNT0 to COUNT2
        LMID    $8,COUNT1    12-bit Counter = H'08D = D'141
        LMID    $0,COUNT2
        BRS     STCNT90      Branch to STCNT90
*
************************************************************
*       STOP : Stop Control
************************************************************
*
STOP    CAL     DEC          Subroutine Jump to 'DEC'
        BRS     STCNT90      ST = "1"? Yes. Branch to STCNT9
*
        LMID    $0,CNTADR    No. Set Slue up Control Mode
        LMID    $1,PLPADR    Set Pulse period Address
        TMD     FOWRDF       FOWRDF = "1" ?
        BRS     STOP10       Yes. Branch to STOP10
        SEMD    FOWRDF       Set FOWRDF
        BRS     STCNT90      Branch to STCNT90
STOP10  REMD    FOWRDF       Reset FOWRDF
        BRS     STCNT90      Branch to STCNT90
*
************************************************************
*       POUT : Output Pulse
************************************************************
*
POUT    LAMD    PLDATA       Load Pulse Data
        TMD     FOWRDF       FOWRDF = 1 ?
        BRS     POT20        Yes. Branch to POT20
*
        LBI     $8           No.
        REC                  Reset CA
        ROTR                 Rotate Right A with Carry
        TC                   CA = 1 ?
        BRS     POT10        Yes. Branch to POT10
        BRS     POT11        No. Branch to POT11
POT10   OR                   A = A OR B
POT11   LRA     $8           Set R8 Port PDR
        LMAD    PLDATA       Save Pulse Data
        BRS     POT99        Branch to POT99
*
POT20   LBI     $1
        REC                  Reset CA
        ROTL                 Rotate Left A with Carry
        TC                   CA = 1 ?
        BRS     POT21        Yes. Branch to POT21
        BRS     POT22        No. Branch to POT22
POT21   OR                   A = A OR B
POT22   LRA     $8           Set R8 Port PDR
```

```
          LMAD    PLDATA      Save Pulse Data
*
POT99     RTN                 Return from Subroutine
*
************************************************************
*         DECR : Decrement Counter
************************************************************
*
DECR      REC                 Reset Carry
          LAI     $0
          SMCD    COUNT0      A = COUNT0 - A - _CA
          LMAD    COUNT0      Save COUNT0
          LAI     $0
          SMCD    COUNT1      A = COUNT1 - A - _CA
          LMAD    COUNT1      Save COUNT1
          LAI     $0
          SMCD    COUNT2      A = COUNT2 - A - _CA
          LMAD    COUNT2      Save COUNT2
          RTN                 Return from Subroutine
*
************************************************************
*         Subroutine Table
************************************************************
*
          org     $20
*
PLOUT     BRL     POUT        Branch to POUT
DEC       BRL     DECR        Branch to DECR
*
************************************************************
*         Timer Period Data Table
************************************************************
*
          org     $100
*
          dc      $100        Start
          dc      $111        30.592ms
          dc      $1AA        11.008ms
          dc      $1BB        8.832ms
          dc      $1CC        6.656ms
          dc      $1CC        6.656ms
          dc      $1CC        6.656ms
          dc      $1DD        4.480ms
          dc      $1DD        4.480ms
          dc      $1DD        4.480ms
          dc      $1DD        4.480ms
          dc      $1DD        4.480ms
          dc      $1DD        4.480ms
          dc      $1DD        4.480ms
          dc      $100        End
*
```

```
************************************************************
*        Pulse Motor Control Address
************************************************************
*
         org     $200
*
         JMPL    SLUP        Jump to Slue up Control Routine
         JMPL    CNST        Jump to Constant Control Routine
         JMPL    SLDOWN      Jump to Slue down Control Routine
         JMPL    STOP        Jump to Stop Control Routine
*
         end
```

3. H4369

```
************************************************************
*
*         H400 Series Application Note
*         - Application Chapter -
*
*         'Stepping Motor Control'
*
*         Function
*         : Timer B Reload Timer
*         : I/O Port
*
*         MCU : H4369
*
*         External Clock : 4MHz
*         Internal Clock : 1MHz
*         Sub Clock      : 32.768kHz
*
************************************************************
*
************************************************************
*         Symbol Definition
************************************************************
*
IE        equ     0,$000      Interrupt Enable Flag
RSP       equ     1,$000      Reset Stack Pointer
IF0       equ     2,$000      INT0 Interrupt Request Flag
IM0       equ     3,$000      INT0 Interrupt Mask
*
IF1       equ     0,$001      INT1 Interrupt Request Flag
IM1       equ     1,$001      INT1 Interrupt Mask
IFTA      equ     2,$001      TIMER A Interrupt Request Flag
IMTA      equ     3,$001      TIMER A Interrupt Mask
*
IFTB      equ     0,$002      TIMER B Interrupt Request Flag
IMTB      equ     1,$002      TIMER B Interrupt Mask
IFTC      equ     2,$002      TIMER C Interrupt Request Flag
IMTC      equ     3,$002      TIMER C Interrupt Mask
*
IFAD      equ     0,$003      A/D Interrupt Request Flag
IMAD      equ     1,$003      A/D Interrupt Mask
IFS       equ     2,$003      SCI Interrupt Request Flag
IMS       equ     3,$003      SCI Interrupt Mask
*
PMRA      equ     $004        Port Mode Register A
SMR       equ     $005        Serial Mode Register
SRL       equ     $006        Serial Data Register L
SRU       equ     $007        Serial Data Register U
TMA       equ     $008        Timer Mode Register A
TMB1      equ     $009        Timer Mode Register B1
```

```
TRBL      equ     $00A         Timer Read Register BL
TWBL      equ     $00A         Timer Write register BL
TRBU      equ     $00B         Timer Read Register BU
TWBU      equ     $00B         Timer Write Register BU
MIS       equ     $00C         Miscellaneous Register
TMC       equ     $00D         Timer Mode Register C
TRCL      equ     $00E         Timer Read Register CL
TWCL      equ     $00E         Timer Write Register CL
TRCU      equ     $00F         Timer Read Register CU
TWCU      equ     $00F         Timer Write Register CU
*
ACR       equ     $016         A/D Control Register
ADRL      equ     $017         A/D Data Register L
ADRU      equ     $018         A/D Data Register U
AMR1      equ     $019         A/D Mode Register 1
AMR2      equ     $01A         A/D Mode Register 2
*
LSON      equ     0,$020       LSON Flag
WDON      equ     1,$020       Watchdog on Flag
ADSF      equ     2,$020       A/D Start Flag
DTON      equ     3,$020       DTON Flag
*
ICSF      equ     0,$021       Input Capture Status Flag
ICEF      equ     1,$021       Input Capture Error Flag
IAOF      equ     2,$021       IAD off Flag
RAME      equ     3,$021       RAM Enable Flag
*
PMRB      equ     $024         Port Mode register B
PMRC      equ     $025         Port Mode Register C
TMB2      equ     $026         Timer Mode Register B2
SSR1      equ     $027         System Clock Selection Register 1
SSR2      equ     $028         System Clock Selection Register 2
*
DCD0      equ     $02C         Data Control Register D0
DCD1      equ     $02D         Data Control Register D1
DCD2      equ     $02E         Data Control Register D2
DCD3      equ     $02F         Data Control Register D3
DCR0      equ     $030         Data Control Register R0
DCR1      equ     $031         Data Control Register R1
DCR2      equ     $032         Data Control Register R2
DCR3      equ     $033         Data Control Register R3
DCR4      equ     $034         Data Control Register R4
DCR5      equ     $035         Data Control Register R5
DCR6      equ     $036         Data Control Register R6
DCR7      equ     $037         Data Control Register R7
DCR8      equ     $038         Data Control Register R8
DCR9      equ     $039         Data Control Register R9
*
************************************************************
*         RAM Allocation
************************************************************
```

```
*
AESC     equ     $040        A Escape RAM Area
BESC     equ     $041        B Escape RAM Area
WESC     equ     $042        W Escape RAM Area
XESC     equ     $043        X Escape RAM Area
YESC     equ     $044        Y Escape RAM Area
SXESC    equ     $045        SPX Escape RAM Area
SYESC    equ     $046        SPY Escape RAM Area
*
PLDATA   equ     $050        Pulse Data ($3 <-> $6 <-> $C <-> $9)
CNTADR   equ     $051        Control Process Address
PLPADR   equ     $052        Pulse Period Address
COUNT0   equ     $053        Counter 0
COUNT1   equ     $054        Counter 1
COUNT2   equ     $055        Counter 2
*
FOWRDF   equ     0,$05F      Forward Flag( 1:Forward, 0:Reverce )
*
************************************************************
*        Vector Address
************************************************************
*
         org     $0000
*
         JMPL    STEPMN      Reset Interrupt
         JMPL    STEPMN      INT0 Interrupt
         JMPL    STEPMN      INT1 Interrupt
         JMPL    STEPMN      Timer A Interrupt
         JMPL    STEPCNT     Timer B Interrupt
         JMPL    STEPMN      Timer C Interrupt
         JMPL    STEPMN      A/D Interrupt
         JMPL    STEPMN      SCI Interrupt
*
************************************************************
*        STEPMN : Main Program
************************************************************
*
         org     $1000
*
STEPMN   REMD    RSP         Stack Pointer Reset
         LMID    $2,SSR1     Initialize System Clock
*
         LMID    $3,PLDATA   Initialize Pulse Data
         LMID    $0,CNTADR   Initialize Control Process Address
         SEMD    FOWRDF      Set Forward Mode
         LMID    $1,PLPADR   Initialize Pulse Period Address
*
         LAI     $0          Initialize R8 Port Data Register
         LRA     $8
         LMID    $F,DCR8     Initialize R8 Port Output Terminal Function
*
```

```
        LMID    $A,TMB1     Initialize Timer B (128tcyc,Reload Timer ON)
        LMID    $8,TWBL     Initialize Period 29.696ms
        LMID    $1,TWBU
        REMD    IFTB        Clear Timer B Interrupt Request Flag
*
        BR      *+1         Set Status Flag
        LMID    $F,COUNT0   Initialize Counter 0
STMN10  CAL     PLOUT       Subroutine Call 'PLOUT'
*
STMN11  TMD     IFTB        29.696ms Pass ?
        BRS     STMN12      Yes. Branch to STMN12
        BRS     STMN11      No. Branch to STMN11
*
STMN12  REMD    IFTB        Clear IFTB
        LAMD    COUNT0      Load Counter 0
        AI      $F          Decrement Counter 0
        LMAD    COUNT0      Save Counter 0
        BRS     STMN10      16 times End ? No. Branch to STMN10
*
        REMD    IFTB        Clear Timer B Interrupt Request Flag
        REMD    IMTB        Timer B Interrupt Enable
        SEMD    IE          Interrupt Enable
*
STMN99  BRS     STMN99      Infinite Loop
*
************************************************************
*       STEPCNT : Stepping Motor Control
************************************************************
*
STEPCNT REMD    IFTB        Clear Timer B Interrupt Request Flag
*
        LMAD    AESC        Store Accumulator
        LAB
        LMAD    BESC        Store B Register
        XSPX
        LASPX
        LMAD    XESC        Store X Register
        XSPX
        LAY
        LMAD    YESC        Store Y Register
*
        LAMD    CNTADR      Load Control Process Address
        LBI     $0
        TBR     $2          Table Branch (CNTADR=0:SLUP, 2:CNST, 4:SLDOWN, 6:STOP)
*
STCNT90 LAMD    YESC        Restore Y Register
        LYA
        LAMD    XESC        Restore X Register
        LXA
        LAMD    BESC        Restore B Register
        LBA
```

```
          LAMD    AESC         Restore Accumulator
*
          RTNI                 Return from Interrupt
*
************************************************************
*         SLUP : Slue up Control
************************************************************
*
SLUP      BR      *+1          Set Status Flag
          CAL     PLOUT        Subroutine Jump to 'PLOUT'
          LAMD    PLPADR       Load Pulse Period Address
          LBI     $0
          P       $1           Pattern Generation 'Output Pulse Period'
          LAB
          ALEI    $0           A <= $0 ?
          BRS     SLUP10       Yes. Branch to SLUP10
          LMAD    TWBL         Set Pulse Period Data Lower
          LAB
          LMAD    TWBU         Set Pulse Period Data Upper
          LAMD    PLPADR       Load Pulse Period Address
          AI      $1           Increment Pulse Period Address
          LMAD    PLPADR       Save Pulse Period Address
          BR      *+1
          BRS     STCNT90      Branch to STCNT90
*
SLUP10    LMID    $2,CNTADR    Set Constant Control Mode
          LMID    $8,COUNT0    Initialize Counter0 to COUNT0
          LMID    $E,COUNT1    12-bit Counter = H'3E8 = D'1000
          LMID    $3,COUNT2
          BRS     STCNT90      Branch to STCNT90
*
************************************************************
*         CNST : Constant Control
************************************************************
*
CNST      BR      *+1          Set Status Flag
          CAL     PLOUT        Subroutine Jump to 'PLOUT'
*
          CAL     DEC          Subroutine Jump to 'DEC'
          BRS     STCNT90      Counter = H'000 ? Yes. Branch to STCNT90
*
          LMID    $4,CNTADR    Set Slue down Control Mode
          LMID    $D,PLPADR    Set Pulse Period Address
          BRS     STCNT90      Branch to STCNT90
*
************************************************************
*         STDOWN : Slue down Control
************************************************************
*
SLDOWN    BR      *+1          Set Status Flag
          CAL     PLOUT        Subroutine Jump to 'PLOUT'
```

```
          LAMD    PLPADR      Load Pulse Period Address
          LBI     $0
          P       $1          Pattern Generation 'Output Pulse Period'
          LAB
          ALEI    $0          A <= $0 ?
          BRS     SLDW10      Yes. Branch to SLDW10
          LMAD    TWBL        Set Pulse Period Data Lower
          LAB
          LMAD    TWBU        Set Pulse Period Data Upper
          LAMD    PLPADR      Load Pulse Period Address
          AI      $F          Decrement Pulse Period Address
          LMAD    PLPADR      Save Pulse Period Address
          BR      *+1
          BRS     STCNT90     Branch to STCNT90
*
SLDW10    LMID    $6,CNTADR   Set Stop Control Mode
          LMID    $D,COUNT0   Initialize COUNT0 to COUNT2
          LMID    $8,COUNT1   12-bit Counter = H'08D = D'141
          LMID    $0,COUNT2
          BRS     STCNT90     Branch to STCNT90
*
************************************************************
*         STOP : Stop Control
************************************************************
*
STOP      CAL     DEC         Subroutine Jump to 'DEC'
          BRS     STCNT90     ST = "1"? Yes. Branch to STCNT9
*
          LMID    $0,CNTADR   No. Set Slue up Control Mode
          LMID    $1,PLPADR   Set Pulse period Address
          TMD     FOWRDF      FOWRDF = "1" ?
          BRS     STOP10      Yes. Branch to STOP10
          SEMD    FOWRDF      Set FOWRDF
          BRS     STCNT90     Branch to STCNT90
STOP10    REMD    FOWRDF      Reset FOWRDF
          BRS     STCNT90     Branch to STCNT90
*
************************************************************
*         POUT : Output Pulse
************************************************************
*
POUT      LAMD    PLDATA      Load Pulse Data
          TMD     FOWRDF      FOWRDF = 1 ?
          BRS     POT20       Yes. Branch to POT20
*
          LBI     $8          No.
          REC                 Reset CA
          ROTR                Rotate Right A with Carry
          TC                  CA = 1 ?
          BRS     POT10       Yes. Branch to POT10
          BRS     POT11       No. Branch to POT11
```

```
POT10    OR                 A = A OR B
POT11    LRA     $8         Set R8 Port PDR
         LMAD    PLDATA     Save Pulse Data
         BRS     POT99      Branch to POT99
*
POT20    LBI     $1
         REC                Reset CA
         ROTL               Rotate Left A with Carry
         TC                 CA = 1 ?
         BRS     POT21      Yes. Branch to POT21
         BRS     POT22      No. Branch to POT22
POT21    OR                 A = A OR B
POT22    LRA     $8         Set R8 Port PDR
         LMAD    PLDATA     Save Pulse Data
*
POT99    RTN                Return from Subroutine
*
************************************************************
*        DECR : Decrement Counter
************************************************************
*
DECR     REC                Reset Carry
         LAI     $0
         SMCD    COUNT0     A = COUNT0 - A - _CA
         LMAD    COUNT0     Save COUNT0
         LAI     $0
         SMCD    COUNT1     A = COUNT1 - A - _CA
         LMAD    COUNT1     Save COUNT1
         LAI     $0
         SMCD    COUNT2     A = COUNT2 - A - _CA
         LMAD    COUNT2     Save COUNT2
         RTN                Return from Subroutine
*
************************************************************
*        Subroutine Table
************************************************************
*
         org     $20
*
PLOUT    BRL     POUT       Branch to POUT
DEC      BRL     DECR       Branch to DECR
*
************************************************************
*        Timer Period Data Table
************************************************************
*
         org     $100
*
         dc      $100       Start
         dc      $111       30.592ms
         dc      $1AA       11.008ms
```


RENESAS

```
         dc      $1BB        8.832ms
         dc      $1CC        6.656ms
         dc      $1CC        6.656ms
         dc      $1CC        6.656ms
         dc      $1DD        4.480ms
         dc      $1DD        4.480ms
         dc      $1DD        4.480ms
         dc      $1DD        4.480ms
         dc      $1DD        4.480ms
         dc      $1DD        4.480ms
         dc      $1DD        4.480ms
         dc      $100        End
*
************************************************************
*        Pulse Motor Control Address
************************************************************
*
         org     $200
*
         JMPL    SLUP        Jump to Slue up Control Routine
         JMPL    CNST        Jump to Constant Control Routine
         JMPL    SLDOWN      Jump to Slue down Control Routine
         JMPL    STOP        Jump to Stop Control Routine
*
         end
```

4. H4889

```
*************************************************************
*
*        H400 Series Application Note
*        - Application Chapter -
*
*        'Stepping Motor Control'
*
*        Function
*        : Timer B Reload Timer
*        : I/O Port
*
*        MCU : H4889
*
*        External Clock : 4MHz
*        Internal Clock : 1MHz
*        Sub Clock      : 32.768kHz
*
*************************************************************
*
*************************************************************
*        Symbol Definition
*************************************************************
*
IE       equ     0,$000      Interrupt Enable Flag
RSP      equ     1,$000      Reset Stack Pointer
IFWU     equ     2,$000      _WU0-_WU3 Interrupt Request Flag
IMWU     equ     3,$000      _WU0-_WU3 Interrupt Mask
*
IF0      equ     0,$001      _INT0 Interrupt Request Flag
IM0      equ     1,$001      _INT0 Interrupt Mask
IF1      equ     2,$001      _INT1 Interrupt request Flag
IM1      equ     3,$001      _INT1 Interrupt Mask
*
IFTA     equ     0,$002      Timer A Interrupt Request Flag
IMTA     equ     1,$002      Timer A Interrupt Mask
IFTB     equ     2,$002      Timer B Interrupt Request Flag
IMTB     equ     3,$002      Timer B Interrupt Mask
*
IFTC     equ     0,$003      Timer C Interrupt Request Flag
IMTC     equ     1,$003      Timer C Interrupt Mask
IFAD     equ     2,$003      A/D Converter Interrupt Request Flag
IMAD     equ     3,$003      A/D Converter Interrupt Mask
*
SSR      equ     $004        System Clock Selection Register
MIS      equ     $005        Miscellaneous Register
ESR      equ     $006        Edge Detection Selection Register
*
PMR0     equ     $008        Port Mode Register 0
PMR1     equ     $009        Port Mode Register 1
```

```
PMR2    equ   $00A     Port Mode Register 2
PMR3    equ   $00B     Port Mode Register 3
PMR4    equ   $00C     Port Mode Register 4
MSR1    equ   $00D     Module Standby Register 1
MSR2    equ   $00E     Module Standby Register 2
TMA     equ   $00F     Timer Mode Register A
TMB1    equ   $010     Timer Mode Register B1
TMB2    equ   $011     Timer Mode Register B2
TRBL    equ   $012     Timer Read Register BL
TWBL    equ   $012     Timer Write Register BL
TRBU    equ   $013     Timer Read Register BU
TWBU    equ   $013     Timer Write Register BU
TMC1    equ   $014     Timer Mode Register C1
TMC2    equ   $015     Timer Mode Register C2
TRCL    equ   $016     Timer Read Register CL
TWCL    equ   $016     Timer Write Register CL
TRCU    equ   $017     Timer Read Register CU
TWCU    equ   $017     Timer Write Register CU
TMD1    equ   $018     Timer Mode Register D1
TMD2    equ   $019     Timer Mode Register D2
TRDL    equ   $01A     Timer Read Register DL
TWDL    equ   $01A     Timer Write Register DL
TRDU    equ   $01B     Timer read Register DU
TWDU    equ   $01B     Timer Write register DU
*
LSON    equ   0,$020   Low Speed on Flag
WDON    equ   1,$020   Watchdog on Flag
ADSF    equ   2,$020   A/D Start Flag
DTON    equ   3,$020   DTON Flag
*
ICSF    equ   0,$021   Input Capture Status Flag
ICEF    equ   1,$021   Input Capture Error Flag
GEF     equ   3,$021   Gear Enable Flag
*
IFTD    equ   2,$022   Timer D Interrupt Request Flag
IMTD    equ   3,$022   Timer D Interrupt Mask
*
IFS     equ   2,$023   Serial Interrupt Request Flag
IMS     equ   3,$023   Serial Interrupt Mask
*
SMR1    equ   $024     Serial Mode Register 1
SMR2    equ   $025     Serial Mode Register 2
SRL     equ   $026     Serial Data Register L
SRU     equ   $027     Serial Data Register U
AMR     equ   $028     A/D Mode Register
*
ADRL    equ   $02A     A/D Data Register L
ADRU    equ   $02B     A/D Data Register U
LCR     equ   $02C     LCD Control Register
LMR     equ   $02D     LCD Mode Register
BMR     equ   $02E     Buzzer Mode Register
```

```
*
DCD0     equ     $030        Data Control Register D0
DCD1     equ     $031        Data Control Register D1
DCD2     equ     $032        Data Control Register D2
*
DCR0     equ     $034        Data Control Register R0
DCR1     equ     $035        Data Control Register R1
DCR2     equ     $036        Data Control Register R2
DCR3     equ     $037        Data Control Register R3
DCR4     equ     $038        Data Control Register R4
DCR5     equ     $039        Data Control Register R5
DCR6     equ     $03A        Data Control Register R6
DCR7     equ     $03B        Data Control Register R7
DCR8     equ     $03C        Data Control Register R8
*
V        equ     $03F        Bank Register
*
************************************************************
*        RAM Allocation
************************************************************
*
AESC     equ     $040        A Escape RAM Area
BESC     equ     $041        B Escape RAM Area
WESC     equ     $042        W Escape RAM Area
XESC     equ     $043        X Escape RAM Area
YESC     equ     $044        Y Escape RAM Area
SXESC    equ     $045        SPX Escape RAM Area
SYESC    equ     $046        SPY Escape RAM Area
*
PLDATA   equ     $090        Pulse Data ($3 <-> $6 <-> $C <-> $9)
CNTADR   equ     $091        Control Process Address
PLPADR   equ     $092        Pulse Period Address
COUNT0   equ     $093        Counter 0
COUNT1   equ     $094        Counter 1
COUNT2   equ     $095        Counter 2
*
FOWRDF   equ     0,$09F      Forward Flag( 1:Forward, 0:Reverce )
*
************************************************************
*        Vector Address
************************************************************
*
         org     $0000
*
         JMPL    STEPMN      Reset Interrupt
         JMPL    STEPMN      _WU0-_WU3 Interrupt
         JMPL    STEPMN      _INT0 Interrupt
         JMPL    STEPMN      _INT1 Interrupt
         JMPL    STEPMN      Timer A Interrupt
         JMPL    STEPCNT     Timer B/D Interrupt
         JMPL    STEPMN      Timer C Interrupt
```

```
        JMPL    STEPMN      A/D Converter / Serial Interrupt
*
************************************************************
*       STEPMN : Main Program
************************************************************
*
        org     $1000
*
STEPMN  REMD    RSP         Stack Pointer Reset
        LMID    $2,SSR      Initialize System Clock
*
        LMID    $3,PLDATA   Initialize Pulse Data
        LMID    $0,CNTADR   Initialize Control Process Address
        SEMD    FOWRDF      Set Forward Mode
        LMID    $1,PLPADR   Initialize Pulse Period Address
*
        LAI     $0          Initialize R2 Port Data Register
        LRA     $2
        LMID    $F,DCR2     Initialize R2 Port Output Terminal Function
*
        LMID    $A,TMB1     Initialize Timer B (128tcyc,Reload Timer ON)
        LMID    $8,TWBL     Initialize Period 29.696ms
        LMID    $1,TWBU
        REMD    IFTB        Clear Timer B Interrupt Request Flag
*
        BR      *+1         Set Status Flag
        LMID    $F,COUNT0   Initialize Counter 0
STMN10  CAL     PLOUT       Subroutine Call 'PLOUT'
*
STMN11  TMD     IFTB        29.696ms Pass ?
        BRS     STMN12      Yes. Branch to STMN12
        BRS     STMN11      No. Branch to STMN11
*
STMN12  REMD    IFTB        Clear IFTB
        LAMD    COUNT0      Load Counter 0
        AI      $F          Decrement Counter 0
        LMAD    COUNT0      Save Counter 0
        BRS     STMN10      16 times End ? No. Branch to STMN10
*
        REMD    IFTB        Clear Timer B Interrupt Request Flag
        REMD    IMTB        Timer B Interrupt Enable
        SEMD    IE          Interrupt Enable
*
STMN99  BRS     STMN99      Infinite Loop
*
************************************************************
*       STEPCNT : Stepping Motor Control
************************************************************
*
STEPCNT REMD    IFTB        Clear Timer B Interrupt Request Flag
*
```

```
         LMAD    AESC        Store Accumulator
         LAB
         LMAD    BESC        Store B Register
         XSPX
         LASPX
         LMAD    XESC        Store X Register
         XSPX
         LAY
         LMAD    YESC        Store Y Register
*
         LAMD    CNTADR      Load Control Process Address
         LBI     $0
         TBR     $2          Table Branch (CNTADR=0:SLUP, 2:CNST, 4:SLDOWN, 6:STOP)
*
STCNT90  LAMD    YESC        Restore Y Register
         LYA
         LAMD    XESC        Restore X Register
         LXA
         LAMD    BESC        Restore B Register
         LBA
         LAMD    AESC        Restore Accumulator
*
         RTNI                Return from Interrupt
*
************************************************************
*        SLUP : Slue up Control
************************************************************
*
SLUP     BR      *+1         Set Status Flag
         CAL     PLOUT       Subroutine Jump to 'PLOUT'
         LAMD    PLPADR      Load Pulse Period Address
         LBI     $0
         P       $1          Pattern Generation 'Output Pulse Period'
         LAB
         ALEI    $0          A <= $0 ?
         BRS     SLUP10      Yes. Branch to SLUP10
         LMAD    TWBL        Set Pulse Period Data Lower
         LAB
         LMAD    TWBU        Set Pulse Period Data Upper
         LAMD    PLPADR      Load Pulse Period Address
         AI      $1          Increment Pulse Period Address
         LMAD    PLPADR      Save Pulse Period Address
         BR      *+1
         BRS     STCNT90     Branch to STCNT90
*
SLUP10   LMID    $2,CNTADR   Set Constant Control Mode
         LMID    $8,COUNT0   Initialize Counter0 to COUNT0
         LMID    $E,COUNT1   12-bit Counter = H'3E8 = D'1000
         LMID    $3,COUNT2
         BRS     STCNT90     Branch to STCNT90
*
```

```
**********************************************************
*        CNST : Constant Control
**********************************************************
*
CNST     BR      *+1          Set Status Flag
         CAL     PLOUT        Subroutine Jump to 'PLOUT'
*
         CAL     DEC          Subroutine Jump to 'DEC'
         BRS     STCNT90      Counter = H'000 ? Yes. Branch to STCNT90
*
         LMID    $4,CNTADR    Set Slue down Control Mode
         LMID    $D,PLPADR    Set Pulse Period Address
         BRS     STCNT90      Branch to STCNT90
*
**********************************************************
*        STDOWN : Slue down Control
**********************************************************
*
SLDOWN   BR      *+1          Set Status Flag
         CAL     PLOUT        Subroutine Jump to 'PLOUT'
         LAMD    PLPADR       Load Pulse Period Address
         LBI     $0
         P       $1           Pattern Generation 'Output Pulse Period'
         LAB
         ALEI    $0           A <= $0 ?
         BRS     SLDW10       Yes. Branch to SLDW10
         LMAD    TWBL         Set Pulse Period Data Lower
         LAB
         LMAD    TWBU         Set Pulse Period Data Upper
         LAMD    PLPADR       Load Pulse Period Address
         AI      $F           Decrement Pulse Period Address
         LMAD    PLPADR       Save Pulse Period Address
         BR      *+1
         BRS     STCNT90      Branch to STCNT90
*
SLDW10   LMID    $6,CNTADR    Set Stop Control Mode
         LMID    $D,COUNT0    Initialize COUNT0 to COUNT2
         LMID    $8,COUNT1    12-bit Counter = H'08D = D'141
         LMID    $0,COUNT2
         BRS     STCNT90      Branch to STCNT90
*
**********************************************************
*        STOP : Stop Control
**********************************************************
*
STOP     CAL     DEC          Subroutine Jump to 'DEC'
         BRS     STCNT90      ST = "1"? Yes. Branch to STCNT9
*
         LMID    $0,CNTADR    No. Set Slue up Control Mode
         LMID    $1,PLPADR    Set Pulse period Address
         TMD     FOWRDF       FOWRDF = "1" ?
```

```
         BRS     STOP10      Yes. Branch to STOP10
         SEMD    FOWRDF      Set FOWRDF
         BRS     STCNT90     Branch to STCNT90
STOP10   REMD    FOWRDF      Reset FOWRDF
         BRS     STCNT90     Branch to STCNT90
*
************************************************************
*        POUT : Output Pulse
************************************************************
*
POUT     LAMD    PLDATA      Load Pulse Data
         TMD     FOWRDF      FOWRDF = 1 ?
         BRS     POT20       Yes. Branch to POT20
*
         LBI     $8          No.
         REC                 Reset CA
         ROTR                Rotate Right A with Carry
         TC                  CA = 1 ?
         BRS     POT10       Yes. Branch to POT10
         BRS     POT11       No. Branch to POT11
POT10    OR                  A = A OR B
POT11    LRA     $2          Set R2 Port PDR
         LMAD    PLDATA      Save Pulse Data
         BRS     POT99       Branch to POT99
*
POT20    LBI     $1
         REC                 Reset CA
         ROTL                Rotate Left A with Carry
         TC                  CA = 1 ?
         BRS     POT21       Yes. Branch to POT21
         BRS     POT22       No. Branch to POT22
POT21    OR                  A = A OR B
POT22    LRA     $2          Set R2 Port PDR
         LMAD    PLDATA      Save Pulse Data
*
POT99    RTN                 Return from Subroutine
*
************************************************************
*        DECR : Decrement Counter
************************************************************
*
DECR     REC                 Reset Carry
         LAI     $0
         SMCD    COUNT0      A = COUNT0 - A - _CA
         LMAD    COUNT0      Save COUNT0
         LAI     $0
         SMCD    COUNT1      A = COUNT1 - A - _CA
         LMAD    COUNT1      Save COUNT1
         LAI     $0
         SMCD    COUNT2      A = COUNT2 - A - _CA
         LMAD    COUNT2      Save COUNT2
```

```
         RTN                  Return from Subroutine
*
************************************************************
*        Subroutine Table
************************************************************
*
         org      $20
*
PLOUT    BRL      POUT        Branch to POUT
DEC      BRL      DECR        Branch to DECR
*
************************************************************
*        Timer Period Data Table
************************************************************
*
         org      $100
*
         dc       $100        Start
         dc       $111        30.592ms
         dc       $1AA        11.008ms
         dc       $1BB        8.832ms
         dc       $1CC        6.656ms
         dc       $1CC        6.656ms
         dc       $1CC        6.656ms
         dc       $1DD        4.480ms
         dc       $1DD        4.480ms
         dc       $1DD        4.480ms
         dc       $1DD        4.480ms
         dc       $1DD        4.480ms
         dc       $1DD        4.480ms
         dc       $1DD        4.480ms
         dc       $100        End
*
************************************************************
*        Pulse Motor Control Address
************************************************************
*
         org      $200
*
         JMPL     SLUP        Jump to Slue up Control Routine
         JMPL     CNST        Jump to Constant Control Routine
         JMPL     SLDOWN      Jump to Slue down Control Routine
         JMPL     STOP        Jump to Stop Control Routine
*
         end
```

2.3 Key Scan and 7-Segment LED Display

Key Scan and 7-Segment LED Display	MCU: H4318/H4359/H4369	Functions Used: R1, R2, R3 Ports, D Port, and Timer A

Specifications

1. As shown in figure 1, the H4318/H4359/H4369 Series are used to construct a key matrix key-scanning function and 7-segment LED display.
2. Numbers 0 to F are assigned to the 16 keys and the number (0 to F of the pressed key) is displayed on LED0, which is one of four 7-segment LEDs (LED0 to LED3).
3. Two keys cannot be pressed simultaneously (the result is invalid).
4. Key chattering is suppressed in software.
5. Figure 2 shows the key configuration and number allocation. Figure 3 shows an example 7-segment LED display.
6. Figure 4 shows the connection of H4318/H4359/H4369 and keys and 7-segment LED.

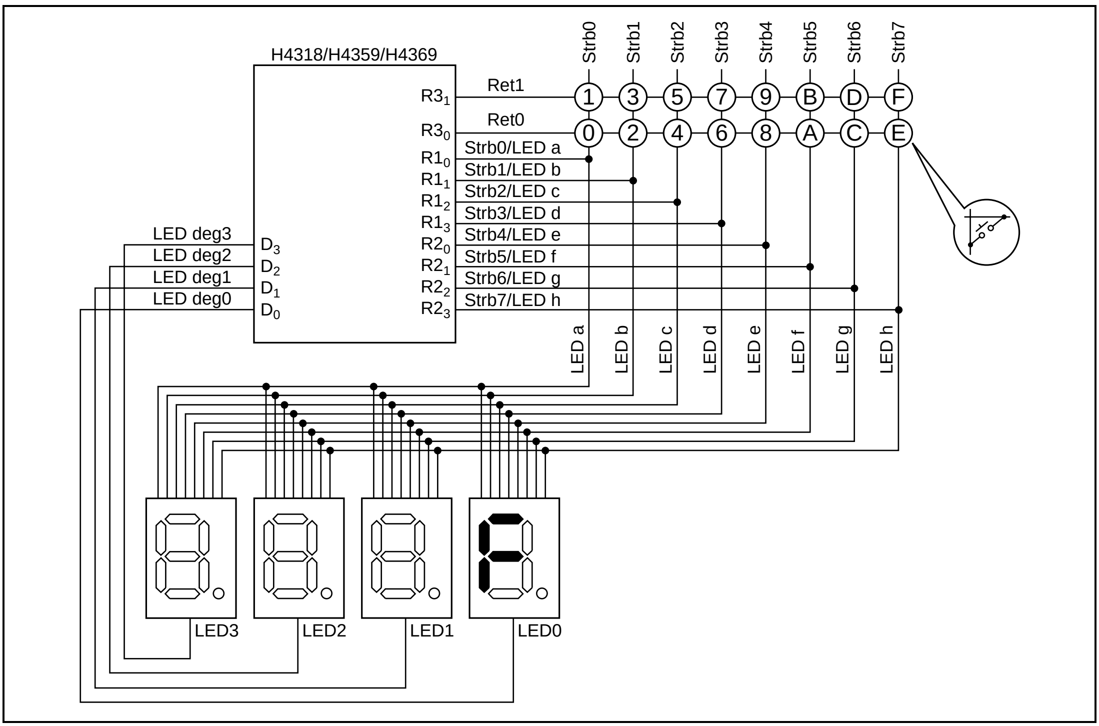


Figure 1 Key Scan and 7-Segment LED Display

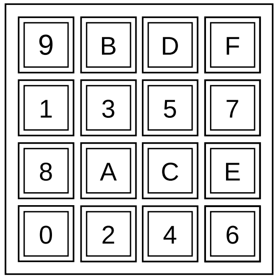


Figure 2   Key Configuration and Assigned Numbers

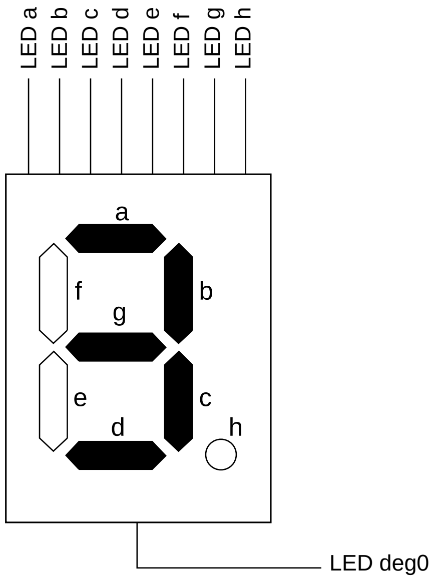


Figure 3   Example 7-Segment LED Display

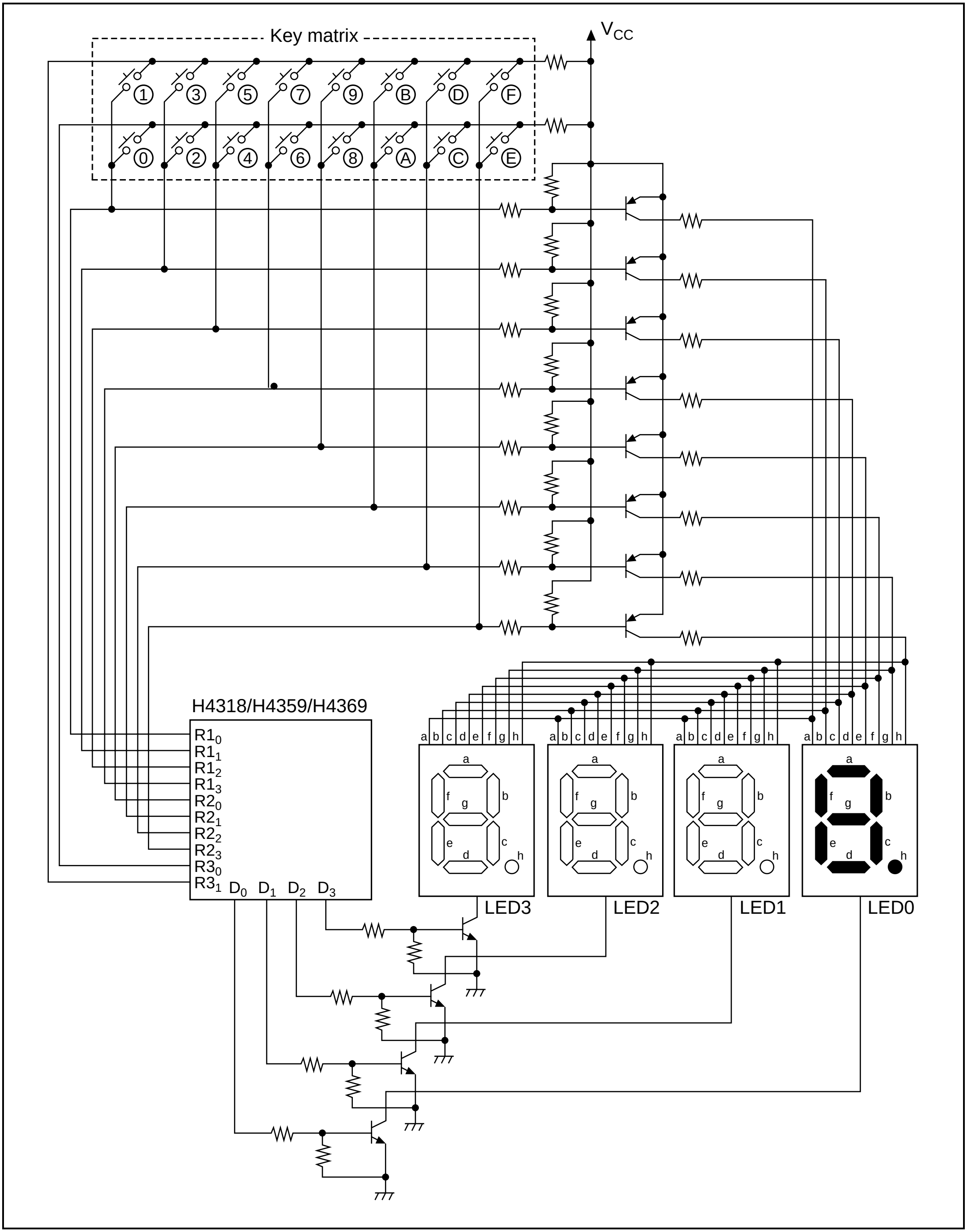


Figure 4   Connection of H4318/H4359/H4369 and keys and 7-Segment LED

Concepts

1. Key Scan

Figure 5 is a timing chart of the key scanning operation.

a. A "Low" signal is output from $P1_0$ pin, and a strobe signal output to the STrb0 row of the key matrix.

b. The key data for the Ret0 row of the key matrix is read from $R3_0$ and $R3_1$ pins when the strobe signal is output.

c. Key depression is detected from the data read from the pins.

d. The strobe signal is shifted and steps a to c repeated.

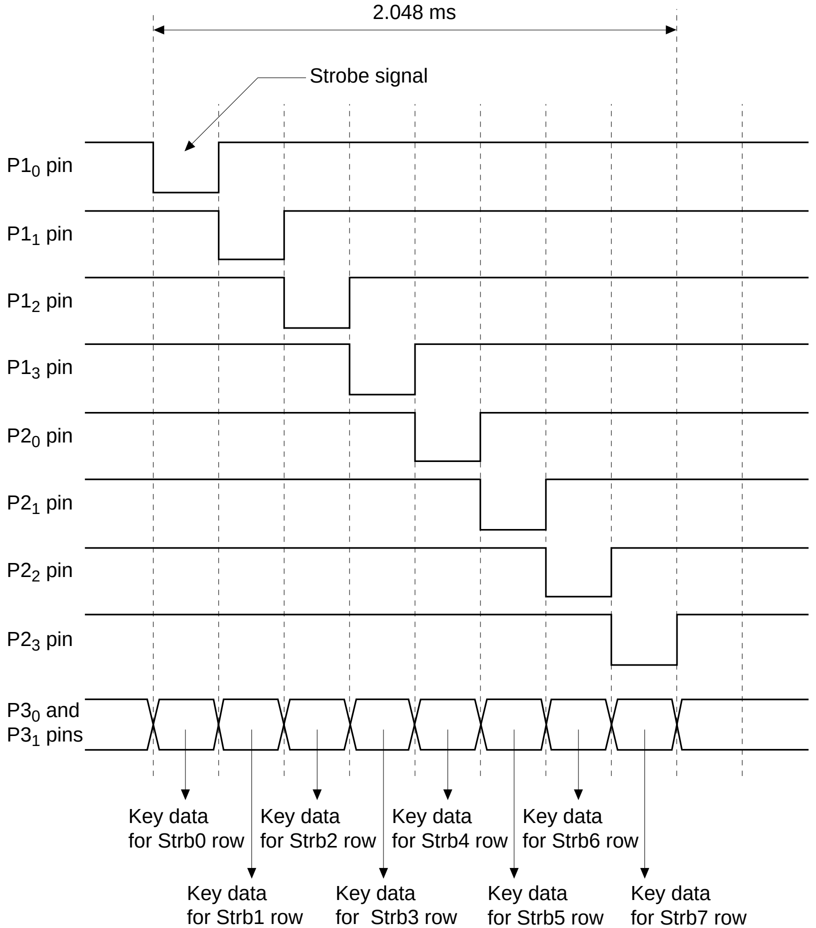


Figure 5 Timing Chart for Key Scanning Operation

2. Chattering Suppression

Figure 6 is a timing chart for chattering suppression.

a. Key data is sampled at intervals of 10.24 ms.
b. The data is checked to see if the same data is obtained 3 times in succession.
c. If the key data is not the same for all 3 times, key depression is ignored.
d. If the key data is the same for all 3 times, the key is assumed to be depressed and the key data is taken to be valid.

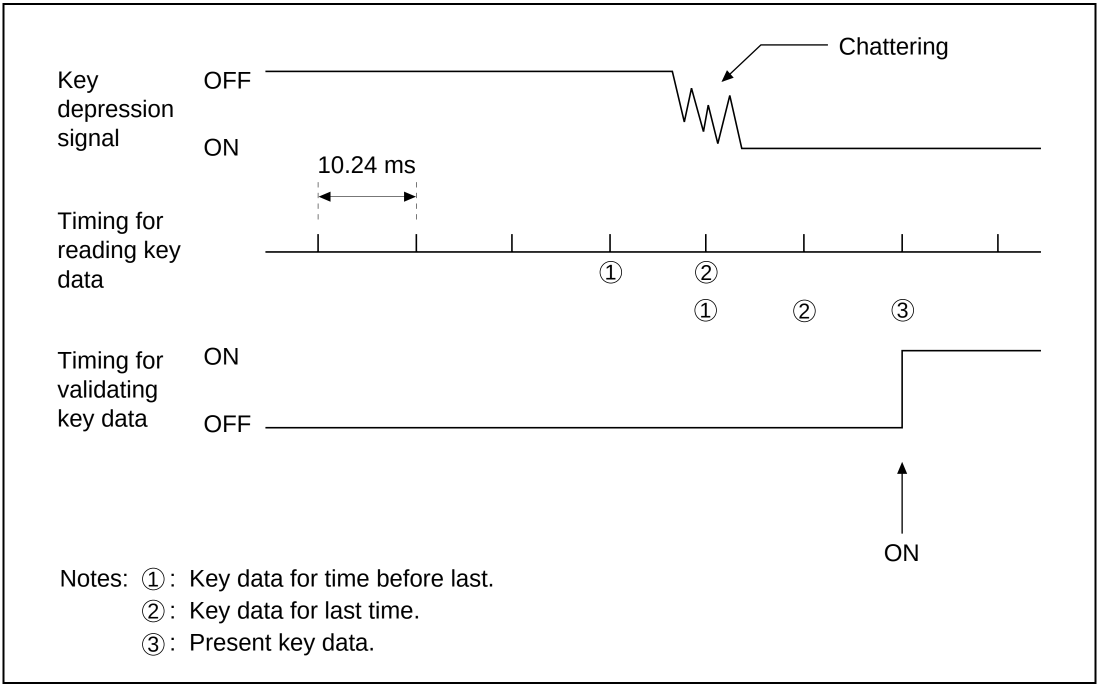


Figure 6 Timing Chart for Chattering Suppression

2. LED Display

Table 1 shows the relationship between 7-segment LED display, port output and segment data.

Table 1 Relationship Between LED Display, Port Output, and Segment Data

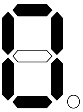
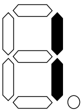
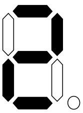
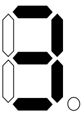
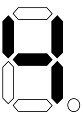
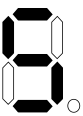
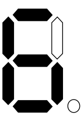
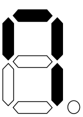
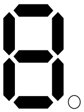
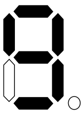
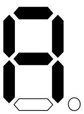
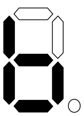
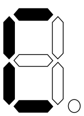
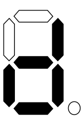
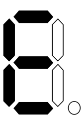
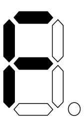
	R1 and R2 Port Output								Segment Data	
LED Display	$R1_0$	$R1_1$	$R1_2$	$R1_3$	$R2_0$	$R2_1$	$R2_2$	$R2_3$	SEG0L	SEG0U
	0	0	0	0	0	0	1	1	$0	$C
	1	0	0	1	1	1	1	1	$9	$F
	0	0	1	0	0	1	0	1	$4	$A
	0	0	0	0	1	1	0	1	$0	$B
	1	0	0	1	1	0	0	1	$9	$9
	0	1	0	0	1	0	0	1	$2	$9
	0	1	0	0	0	0	0	1	$2	$8
	0	0	0	1	1	0	1	1	$8	$D

Table 1 Relationship Between LED Display, Port Output, and Segment Data (cont)

LED Display	R1 and R2 Port Output								Segment Data	
	$R1_0$	$R1_1$	$R1_2$	$R1_3$	$R2_0$	$R2_1$	$R2_2$	$R2_3$	SEG0L	SEG0U
	0	0	0	0	0	0	0	1	$0	$8
	0	0	0	0	1	0	0	1	$0	$9
	0	0	0	1	0	0	0	1	$8	$8
	1	1	0	0	0	0	0	1	$3	$8
	0	1	1	0	0	0	1	1	$6	$C
	1	0	0	0	0	1	0	1	$1	$A
	0	1	1	0	0	0	0	1	$6	$8
	0	1	1	1	0	0	0	1	$E	$8

Description of Functions

1. This section describes the functions of the H4318/H4359/H4369 used in key scanning and 7-segment LED display. Figure 7 is a block diagram of the functions used in this example task.

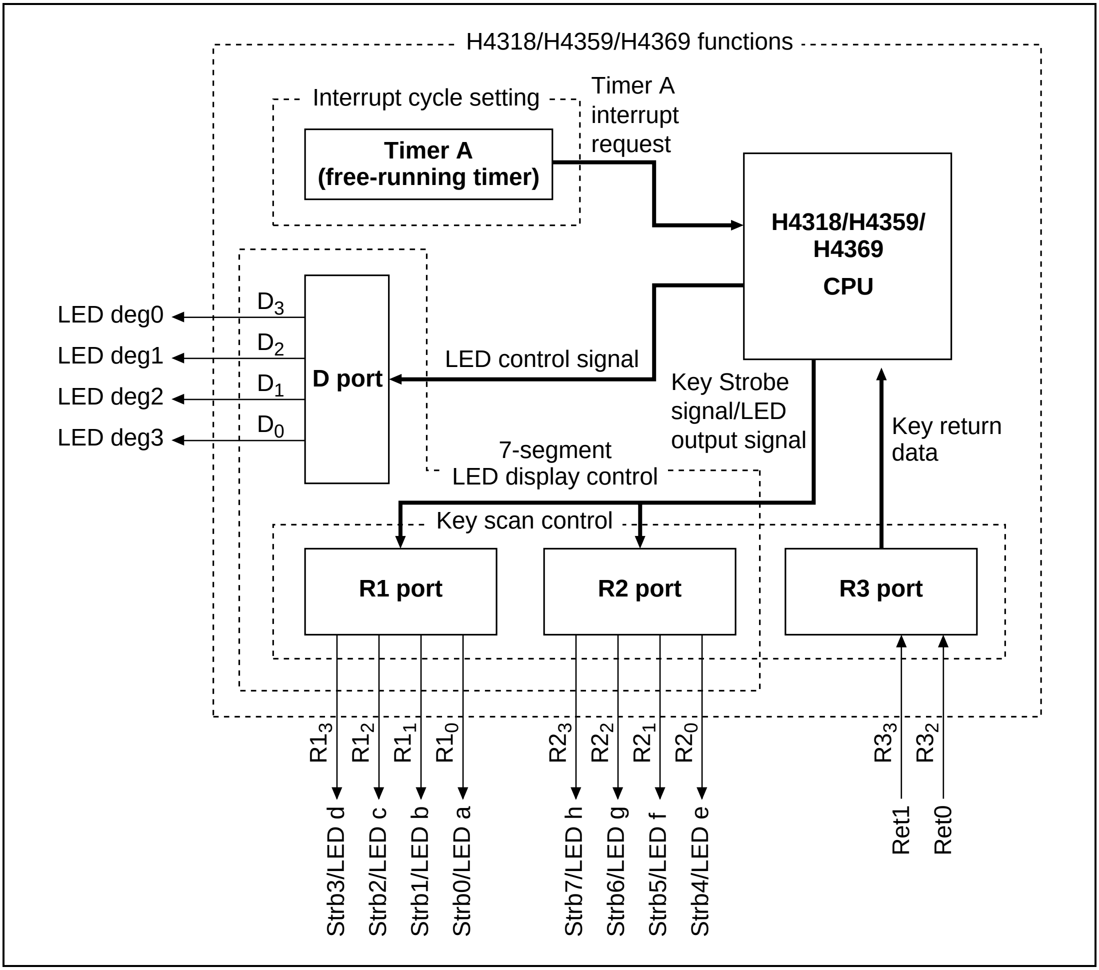


Figure 7 Block Diagram of H4318/H4359/H4369 Functions Used in Key Scanning and 7-Segment LED Display

2. Descriptions of Functions of Timer A, R1/R2/R3 Ports, and D Port.

a. Figure 8 is a block diagram of the timer A functions.

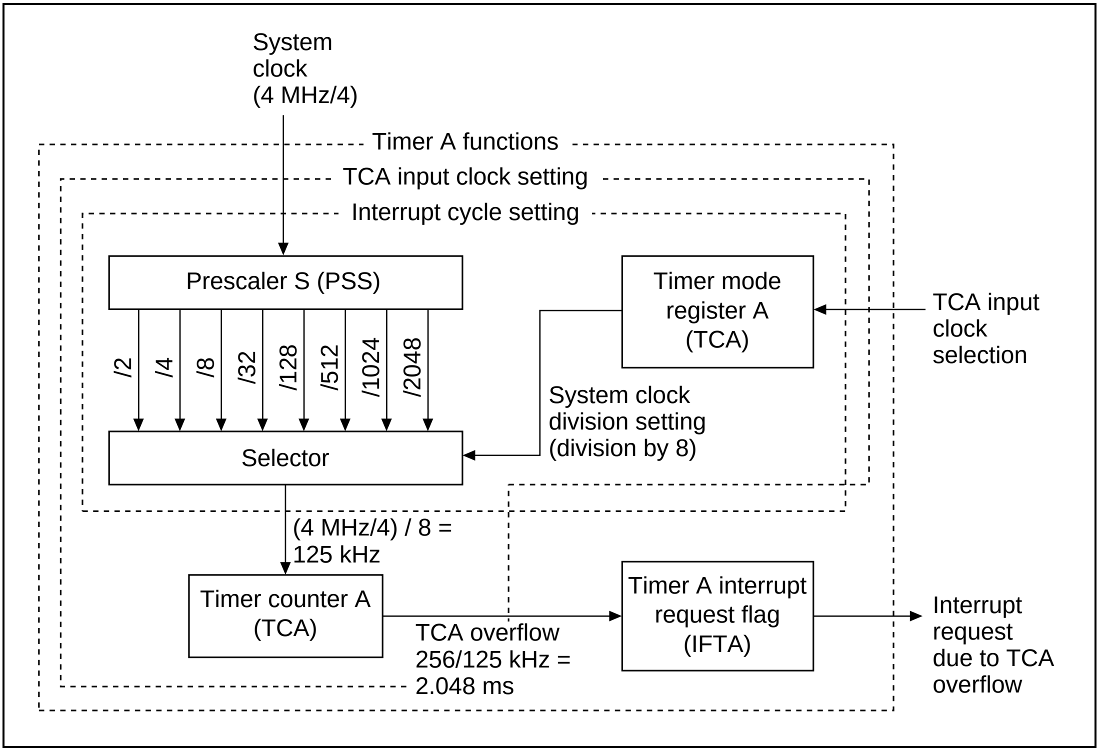


Figure 8 Block Diagram of Timer A Functions

b. Timer A is an 8-bit free-running timer. However, in the H4369, it can also be used as a time based real-time clock using the system clock oscillator (32.768 kHz). Table 2 describes the timer A functions.

Table 2 Timer A Functions

Timer Mode Register A (TMA)	
Function	TMA is a 4-bit write-only register. It selects the division ratio of the prescaler S, which is the clock source for timer A. TMA is initialized to $0 when reset and in stop mode. In the H4318/H4359 TMA bit 3 (TMA3) cannot be used. In the H4369, TMA3 selects the prescaler (PSS or PSW), which is the clock source of timer A.
Timer Counter A (TCA)	
Function	TCA is an 8-bit up-counter, which is incremented by the input internal clock. The TCA input clock is selected by TMA. TCA cannot be read or written to. When TCA overflows, the timer A interrupt request flag (IFTA) is set to “1”. TCA is initialized to $00 when reset and in stop mode.
Prescaler S (PSS)	
Function	PSS is an 11-bit counter to which the system clock is in put when in active mode and standby mode, and the subsystem clock is input when in subactive mode*. PSS is initialized to $000 at a reset, and the system clock count starts when the reset is canceled. PSS operation is halted when reset, in stop mode, and in watch mode*. However, it runs in other operating modes. The PSS output is shared by the internal peripheral modules, the division ratio being set independently for each of the internal peripheral modules.
Timer A Interrupt Request Flag (IFTA)	
Function	IFTA reflects the existence of the timer A interrupt request. When timer A overflows, IFTA is set to “1”. IFTA can only be read/written to (only “0” can be written) using bit operation commands. Note that IFTA is not automatically cleared even when the interrupt is received, and must be cleared by writing “0” using software. IFTA is cleared at a reset and in stop mode.
Timer A Interrupt Mask (IMTA)	
Function	IMTA is the bit that masks IFTA. When IFTA is set to “1” and, additionally, IMTA is “0”, a timer A interrupt request is sent to the CPU (when IE = “1”). If IFTA is set to “1” but IMTA is “1”, no interrupt request is sent to the CPU and the timer A interrupt is held. IMTA can only be read or written to using bit operation commands. It is set to “1” at a reset and in stop mode.

Note: * Applies to H4369 only.

c. Ports R1, R2, and R3 are 4-bit I/O ports, accessed in units of 4 bits. Each of ports R1 to R3 are accessed in 4-bit units using the output commands (LRA and LRB) to control the output level High/Low. The output data is stored in the port data registers (PDR) of the respective pins. The input commands (LAR and LBR) are used for 4-bit access to read the pin levels.

d. Table 3 describes the functions of the R1, R2, and R3 ports.

Table 3 Functions of R1, R2, and R3 Ports

Data Control Register R1 (DCR1)	
Function	DCR1 switches the I/O pin function of the R1 port. When any bit of DCR1 is cleared to “0”, the output buffer (CMOS) of the corresponding pin is turned OFF and the output is set to high impedance. When the respective bit of DCR1 is set to “1”, the output buffer of the corresponding pin is set ON and the corresponding PDR value is output.
Data Control Register R2 (DCR2)	
Function	DCR2 switches the I/O pin function of the R2 port. When any bit of DCR2 is cleared to “0”, the output buffer (CMOS) of the corresponding pin is turned OFF and the output is set to high impedance. When the respective bit of DCR2 is set to “1”, the output buffer of the corresponding pin is set ON and the corresponding PDR value is output.
Data Control Register R3 (DCR3)	
Function	DCR3 switches the I/O pin function of the R3 port. When any bit of DCR3 is cleared to “0”, the output buffer (CMOS) of the corresponding pin is turned OFF and the output is set to high impedance. When the respective bit of DCR3 is set to “1”, the output buffer of the corresponding pin is set ON and the corresponding PDR value is output.
Port Data Register (PDR)	
Function	The I/O pins of the R ports have built-in PDRs to store the output data. When the LRA and LRB commands are executed, the contents of the accumulator (A) and B register (B) are transferred to the PDR of the specified R port. When the corresponding DCR of the R port is “1”, the output buffer of the appropriate pin is set ON and the value in the PDR is output via that pin. The PDR is initialized to $F at a reset.
A/D Mode Register 1 (AMR1)	
Function	AMR1 is a 4-bit write-only register. Bits AMR13 to AMR10 switch the functions of the ports dual-function pins.

e. The D ports are 1-bit input output ports, accessed in 1-bit units. Pins D_0 to D_3 are accessed in 1-bit units using the output commands (SED, SEDD, RED, and REDD) to control the High/Low output level. The output data is stored in the PDR of the respective pin. Pins D_0 to D_3 are also accessed in 1-bit units using the input commands (TD and TDD) to test the pin level.

f. Table 4 describes the functions of the D ports.

Table 4 D Port Functions

Data Control Register D0 (DCD0)	
Function	DCD0 switches the I/O pin function of pins D_0 to D_3. When any bit of DCD0 is cleared to “0”, the output buffer (CMOS) of the corresponding pin is turned OFF and the output is set to high impedance. When the respective bit of DCD0 is set to “1”, the output buffer of the corresponding pin is set ON and the corresponding PDR value is output.
Port Data Register (PDR)	
Function	The I/O pins D_0 to D_8 have built-in PDRs to store the output data. When the SED or SEDD commands are executed for pins D_0 to D_8, the corresponding PDR is set to “1”. When the RED or REDD commands are executed, the corresponding PDR is cleared to “0”. When a corresponding bit to DCD0 to DCD2 is “1”, the output buffer of that pin is turned ON and the value in the PDR is output via that pin. The PDR is set to “1” at a reset an in stop mode.
Port Mode Register A (PMRA)	
Function	PMRA is a 4-bit write-only register. PMRA3 switches the function of the D_3/BUZZ pin.
Port Mode Register B (PMRB)	
Function	PMRB is a 4-bit write-only register. PMRB0 switches the functions of the $D_0/\overline{INT}_0$ pin, PMRB1 switches $D_1/\overline{INT}_1$, and PMRB2 switches D_2/EVNB.

3. Table 5 describes the allocation of functions in this example task.

Table 5 Allocation of Functions

Function	Function Allocation
System clock	The system clock is obtained by dividing the clock output from the system clock oscillator by 4. It is used for operating the CPU and internal peripheral modules. In this example task, a 4 MHz system clock oscillator is used, so the clock supplied to the CPU and internal peripheral modules is 1 MHz. The clock used by timer B and timer C is obtained by dividing the 1 MHz clock at PSS.
PSS	Generates the clock input to timer A by dividing the system clock. The clock input to timer A is obtained by dividing the system clock by 8.
TCA	This 8-bit up-counter counts on the input internal clock ((4 MHz/4) / 8 = 125 kHz). IFTA is set to “1” when TCA overflows.
TMA	Selects the system clock divided by 8 as the TCA input clock.
IFTA	Reflects the existence of timer A interrupt requests.
IMTA	Enables/disables timer A interrupt requests.
DCD0	Sets D_0 to D_3 to function as output pins.
DCR1	Sets $R1_0$ to $R1_3$ of the R1 port to function as output pins.
DCR2	Sets $R2_0$ to $R2_3$ of the R2 port to function as output pins.
DCR3	Sets $R3_0$ and $R3_1$ of the R3 port to function as output pins.
PDR	Stores the data output from the respective pins.
AMR1	Sets the $R3_0/AN_0$ to $R3_3/AN_3$ pins to function as $R3_0$ to $R3_3$ I/O pins.
PMRA	Sets the D_3/BUZZ pin to function as a D_3 I/O pin.
PMRB	Sets the $D_0/\overline{INT}_0$ pin to function as D_0 I/O pin, $D_1/\overline{INT}_1$ pin to function as D_1 I/O pin, and D_2/EVNB pin to function as D_2 I/O pin.
Pins $R1_0$ to $R1_3$	Output pins for Strb0/LED a to Strb3/LED d signals.
Pins $R2_0$ to $R2_3$	Output pins for Strb4/LED e to Strb7/LED h signals.
Pins $R3_0$ and $R3_1$	Input pins for Ret0 and Ret1 signals.
Pins D_0 to D_3	Output pins for LED deg1 to LED deg3 signals.

Description of Operation

1. Figure 9 shows the operating principles of key scanning and the 7-segment LED display.

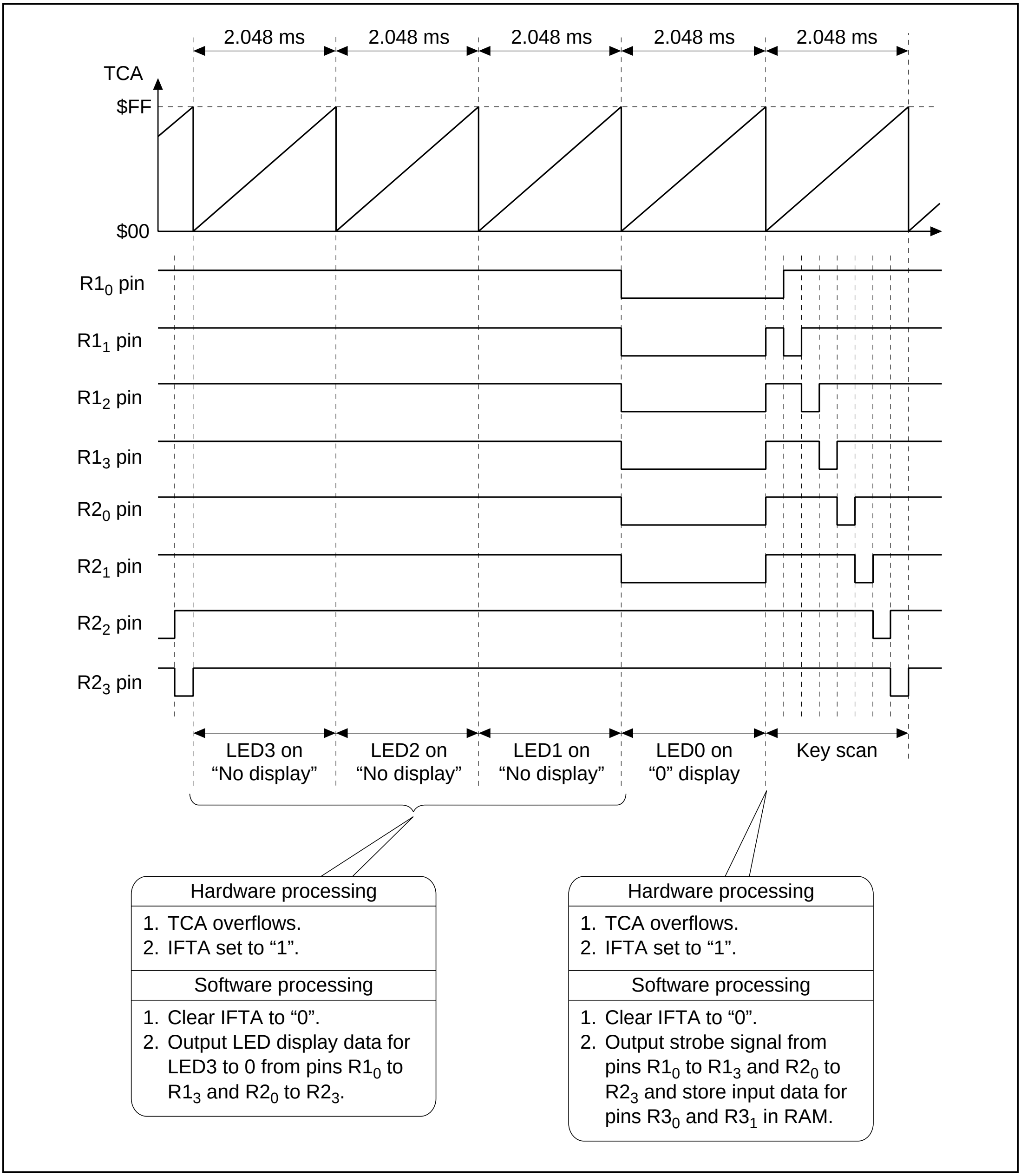


Figure 9 Operating Principles of Key Scanning and 7-Segment LED Display

Description of Software

1. Description of Modules

Table 6 shows the modules used in this example task.

Table 6 Modules

Module	Label	Functions
Main routine	LDKYMN	This routine makes the initial stack pointer, RAM, I/O port, and timer A settings, enables interrupts, and calls the KEYDEC and LEDDSP subroutines.
Key decoder	KEYDEC	Converts to key data on completion of key scanning, judges if the key data is the same three times in succession, and suppresses chattering.
LED display	LEDDSP	When the key data is the same three times in succession and is therefore taken to be valid, this routine converts the key data to LED display data and displays the number of the depressed key on the 7-segment LED.
Timer A interrupt processing routine	LDKYINT	Saves the registers, calls the LED and KEY subroutines, and restores registers.
LED control	LED	Controls the lighting of LED0 to LED3.
Key scan	KEY	Performs key scanning by outputting a strobe signal and storing the data input to pins $R3_0$ and $R3_1$ in RAM.

2. Description of Arguments

No arguments are used in this example task.

3. Description of Internal Registers

Table 7 describes the internal registers used in this example task.

Table 7 Internal Registers Used in Example Task

Register	Description	RAM Address	Setting
IE	Interrupt Enable Flag Controls the CPU can receive any interrupts. • When IE = “0”, CPU reception of all interrupts is disabled. • When IE = “1”, CPU reception is enabled.	0, $000	1
RSP	Reset Stack Pointer The stack pointer is initialized by clearing RSP to “0”.	1, $000	0
IFTA	Timer A Interrupt Request Flag Reflects the existence of a timer A interrupt request. • When IFTA = “0”, there is no timer A interrupt request. • When IFTA = “1”, there is a timer A interrupt request.	2, $001	0
IMTA	Timer A Interrupt Mask This bit masks IFTA. • When IMTA = “0”, IFTA is enabled. • When IMTA = “1”, IFTA is masked.	3, $001	0
PMRA	Port Mode Register A Switches the function of the D_3/BUZZ pin. • When PMRA = “0”, the D_3/BUZZ pin functions as the D_3 I/O pin. • When PMRA = “1”, the D_3/BUZZ pin functions as the BUZZ output pin.	$004	$0
TMA	Timer Mode Register A Selects the timer A clock source and the input clock cycle. • When TMA3 = “0”, the timer A clock source is the PSS. However, this applies only to the H4369. In the H4318/H4359, TMA3 cannot be used. • When TMA2 = “1” and TMA1 = “0”, and TMA0 = “1”, the timer A input clock cycle is set to 8 µs.	$008	$5

Table 7 Internal Registers Used in Example Task (cont)

Register	Description	RAM Address	Setting
AMR1	A/D mode register (AMR1)	$019	$0
	Switches the function of the R3 port’s dual-function pins.		
	• When AMR13 = “0”, the $R3_3/AN_3$ pin functions as the $R3_3$ I/O pin.		
	• When AMR13 = “1”, the $R3_3/AN_3$ pin functions as the AN_3 input pin.		
	• When AMR12 = “0”, the $R3_2/AN_2$ pin functions as the $R3_2$ I/O pin.		
	• When AMR12 = “1”, the $R3_2/AN_2$ pin functions as the AN_2 input pin.		
	• When AMR11 = “0”, the $R3_1/AN_1$ pin functions as the $R3_1$ I/O pin.		
	• When AMR11 = “1”, the $R3_1/AN_1$ pin functions as the AN_1 input pin.		
	• When AMR10 = “0”, the $R3_0/AN_0$ pin functions as the $R3_0$ I/O pin.		
	• When AMR10 = “1”, the $R3_0/AN_0$ pin functions as the AN_0 input pin.		
PMRB	Port Mode Register B (PMRB)	$024	$0
	Switches the function of the D port’s dual-function pins.		
	• When PMRB2 = “0”, the D_2/EVNB pin functions as the D_2 I/O pin.		
	• When PMRB2 = “1”, the D_2/EVNB pin functions as the EVNB input pin.		
	• When PMRB1 = “0”, the $D_1/\overline{INT_1}$ pin functions as the D_1 I/O pin.		
	• When PMRB1 = “1”, the $D_1/\overline{INT}_1$ pin functions as the $\overline{INT}_1$ input pin.		
	• When PMRB0 = “0”, the $D_0/\overline{INT}_0$ pin functions as the D_0 I/O pin.		
	• When PMRB0 = “1”, the $D_0/\overline{INT}_0$ pin functions as the $\overline{INT}_0$ input pin.		

Table 7 Internal Registers Used in Example Task (cont)

Register	Description	RAM Address	Setting
SSR1	System Clock Selection Register 1 Selects the system clock oscillation frequency, subsystem clock frequency division, and, in stop mode, the subsystem clock oscillation. • When SSR11 = “0”, the system clock oscillation frequency is set to 0.4 to 1 MHz. • When SSR11 = “1”, the system clock oscillation frequency is set to 1.6 to 5 MHz. Note: Applicable only to H4369.	$027	$2
DCD0	Data Control Register D0 Controls the ON/OFF state of the D port output buffer. • When DCD03 to DCD00 = “0”, the output buffers of the D_3 to D_0 pins are OFF and output set to high impedance. • When DCD03 to DCD00 = “1”, the output buffers of the D_3 to D_0 pins are ON and the values of the corresponding PDRs are output.	$02C	$F
DCR1	Data Control Register R1 Controls the ON/OFF state of the R1 port output buffer. • When DCR13 to DCR10 = “0”, the output buffers of the $R1_3$ to $R1_0$ pins are OFF and output set to high impedance. • When DCR13 to DCR10 = “1”, the output buffers of the $R1_3$ to $R1_0$ pins are ON and the values of the corresponding PDRs are output.	$030	$F
DCR2	Data Control Register R2 DCR2 switches the output buffer of the R2 port ON/OFF. • When DCR23 to DCR20 = “0”, the output buffers of pins $R2_3$ to $R2_0$ are OFF and the pins are in the high impedance state. • When DCR23 to DCR20 = “1”, The output buffers of pins $R2_3$ to $R2_0$ are ON and the values in the corresponding PDRs are output.	$031	$F

Table 7 Internal Registers Used in Example Task (cont)

Register	Description	RAM Address	Setting
DCR3	Data Control Register R3 DCR3 switches the output buffer of the R3 port ON/OFF. • When DCR33 to DCR30 = "0", the output buffers of pins $R3_3$ to $R3_0$ are OFF and the pins are in the high impedance state. • When DCR33 to DCR30 = "1", The output buffers of pins $R3_3$ to $R3_0$ are ON and the values in the corresponding PDRs are output.	$032	$0

4. Description of RAM

Table 8 describes the RAM used in this example task.

Table 8 RAM

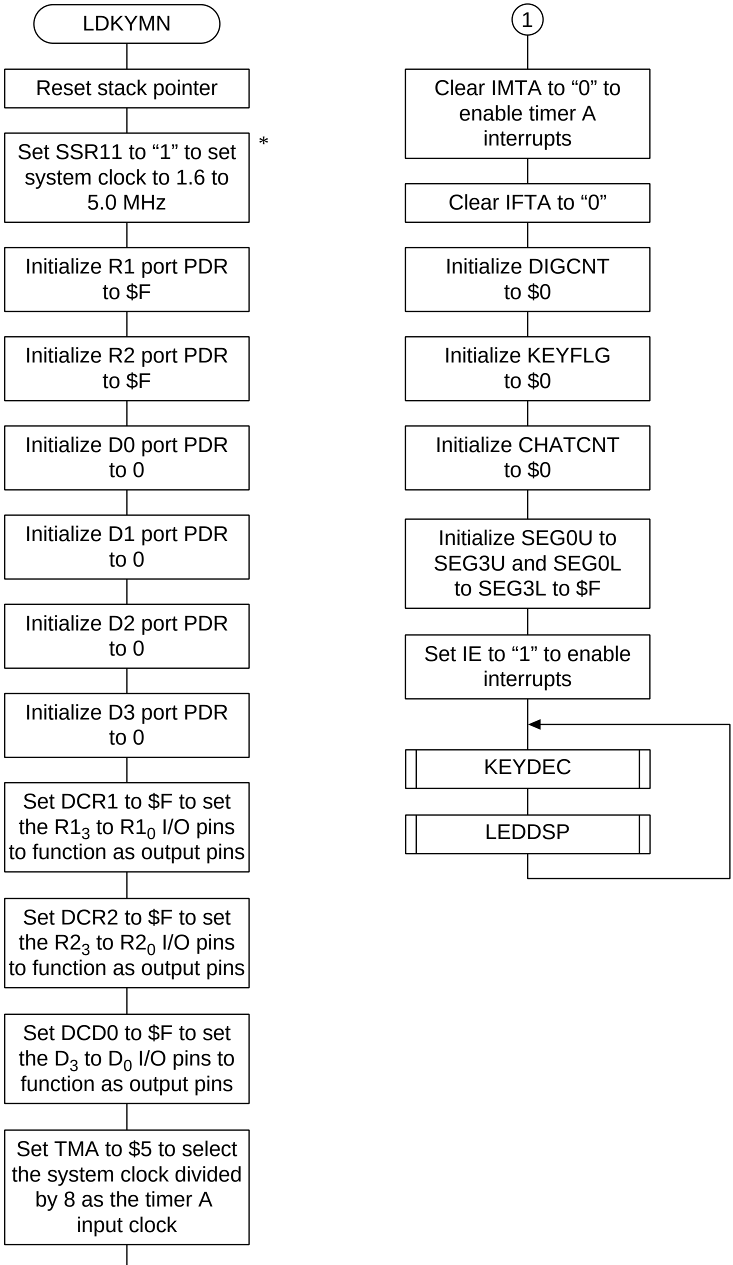
Label	Description	RAM Address	Module
AESC	Stores content of accumulator when processing timer A interrupt.	$040	LDKYINT
BESC	Stores content of B register when processing timer A interrupt.	$041	LDKYINT
XESC	Stores content of X register when processing timer A interrupt.	$043	LDKYINT
YESC	Stores content of Y register when processing timer A interrupt.	$044	LDKYINT
DIGCNT	Counter to control output to 7-segment LED.	$054	LDKYMN, LED, KEY
SEG0L	Stores lower 4 bits of display data output to LED0.	$050	LDKYMN, LEDDSP, LED
SEG1L	Stores lower 4 bits of display data output to LED1.	$051	LDKYMN, LEDDSP, LED
SEG2L	Stores lower 4 bits of display data output to LED2.	$052	LDKYMN, LEDDSP, LED
SEG3L	Stores lower 4 bits of display data output to LED3.	$053	LDKYMN, LEDDSP, LED
SEG0U	Stores upper 4 bits of display data output to LED0.	$060	LDKYMN, LEDDSP, LED

Table 8 RAM (cont)

Label	Description	RAM Address	Module
SEG1U	Stores upper 4 bits of display data output to LED1.	$061	LDKYMN, LEDDSP, LED
SEG2U	Stores upper 4 bits of display data output to LED2.	$062	LDKYMN, LEDDSP, LED
SEG3U	Stores upper 4 bits of display data output to LED3.	$063	LDKYMN, LEDDSP, LED
KEYFLG	Stores KEYONF and KDECONF.	$080	—
KEYONF	Flag showing end of key scanning.	0, $080	LDKYMN, KEYDEC, KEY
KDECONF	Flag confirming key data.	1, $080	LDKYMN, KEYDEC, LEDDSP
KONNEW	Stores new key data	$07F	KEYDEC, LEDDSP
KONOLD	Stores old key data	$07E	KEYDEC
CHATCNT	Counter to suppress chattering	$07D	LDKYMN, KEYDEC
STRBU	Stores upper 4 bits of output strobe signal	$079	KEY
STRBL	Stores lower 4 bits of output strobe signal	$078	KEY

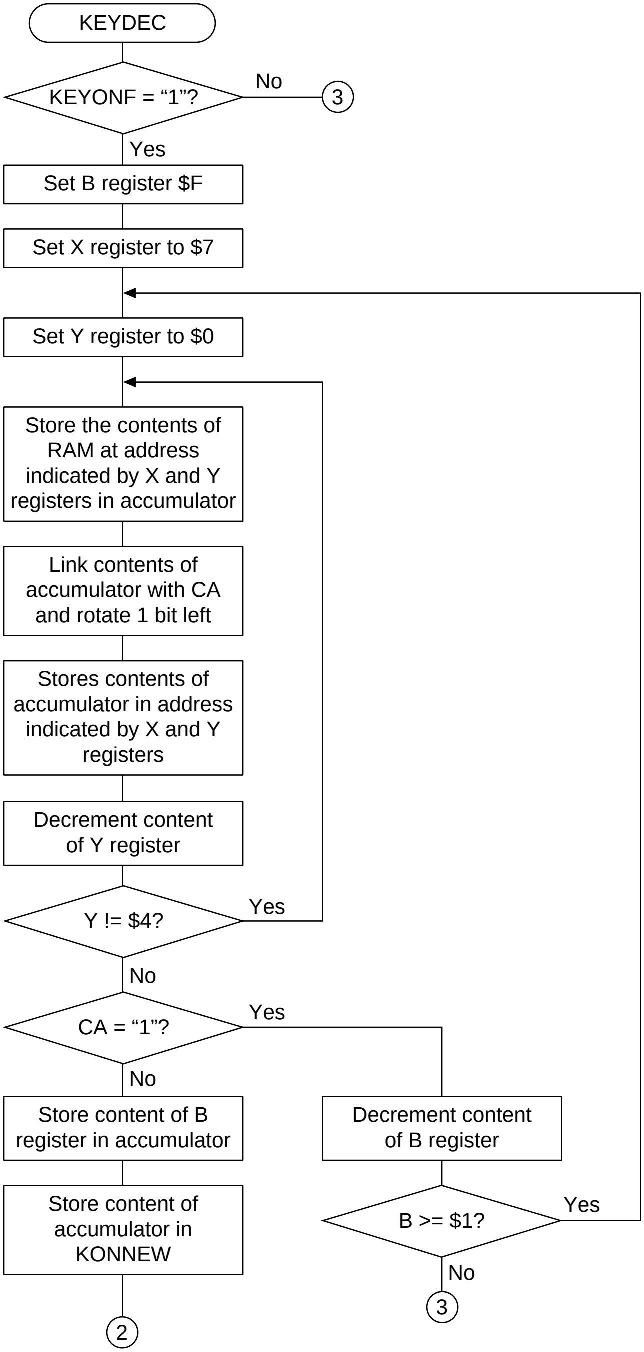
Flowcharts

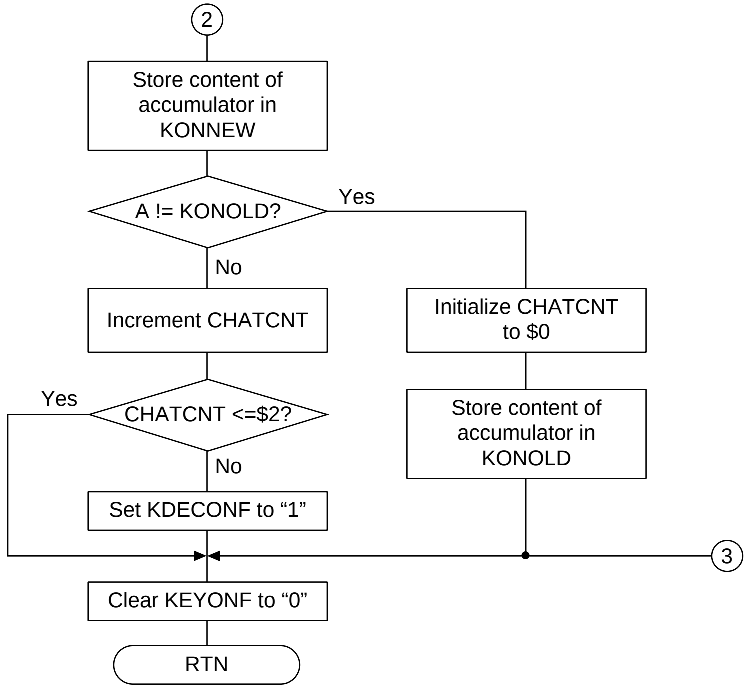
1. Main Routine

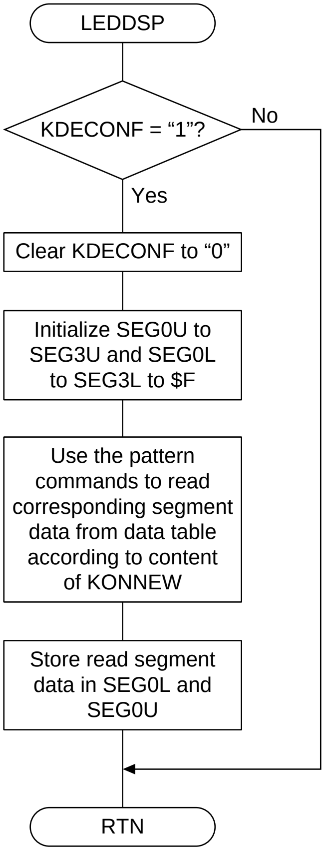


Note: * Applies to H4369 only

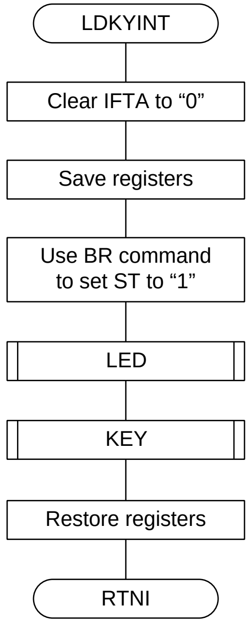
2. Key Decoder



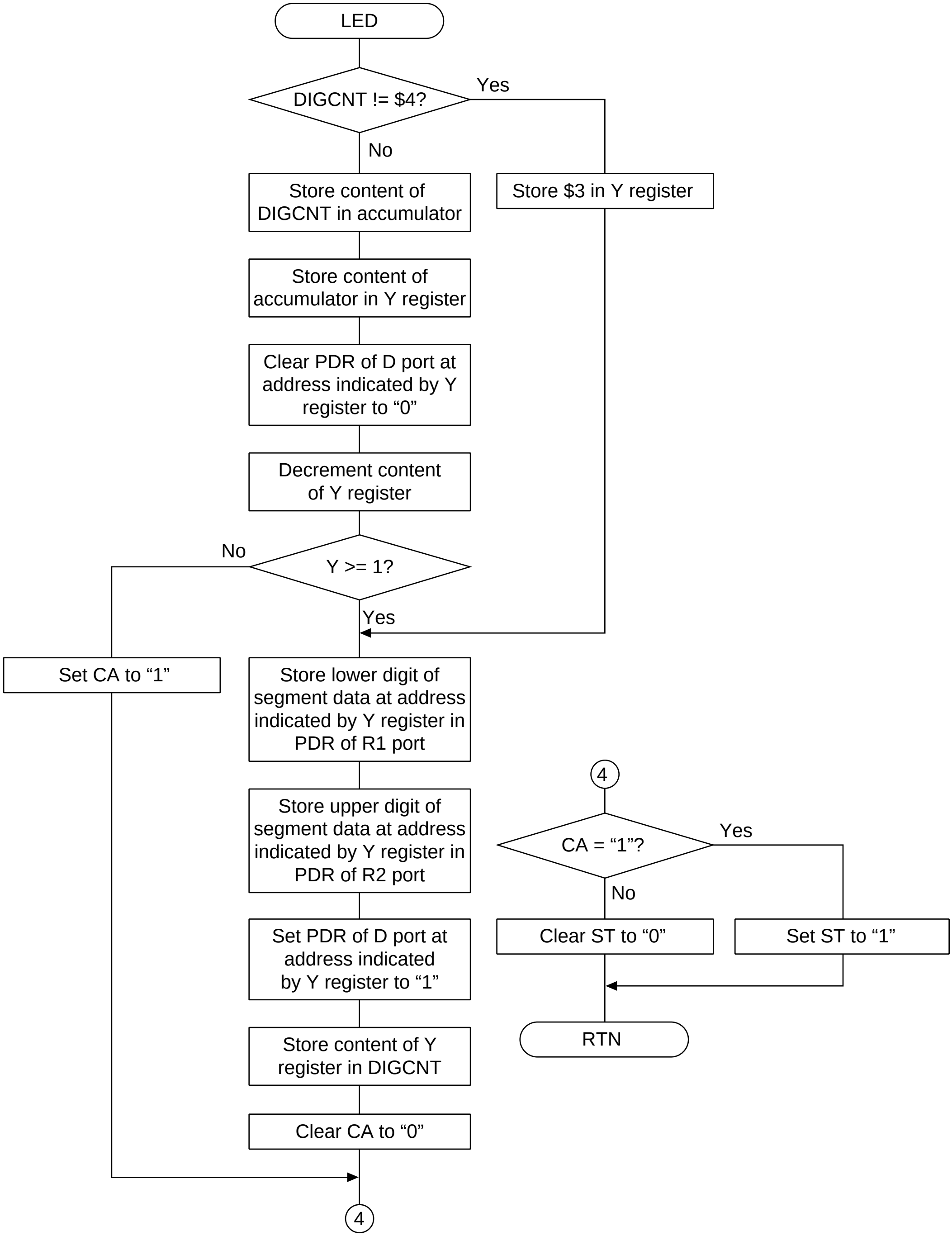

2
Store content of accumulator in KONNEW
A != KONOLD?
Yes
No
Increment CHATCNT
Initialize CHATCNT to $0
Yes
CHATCNT <=$2?
No
Store content of accumulator in KONOLD
Set KDECONF to “1”
3
Clear KEYONF to “0”
RTN


LEDDSP
KDECONF = “1”?
No
Yes
Clear KDECONF to “0”
Initialize SEG0U to SEG3U and SEG0L to SEG3L to $F
Use the pattern commands to read corresponding segment data from data table according to content of KONNEW
Store read segment data in SEG0L and SEG0U
RTN

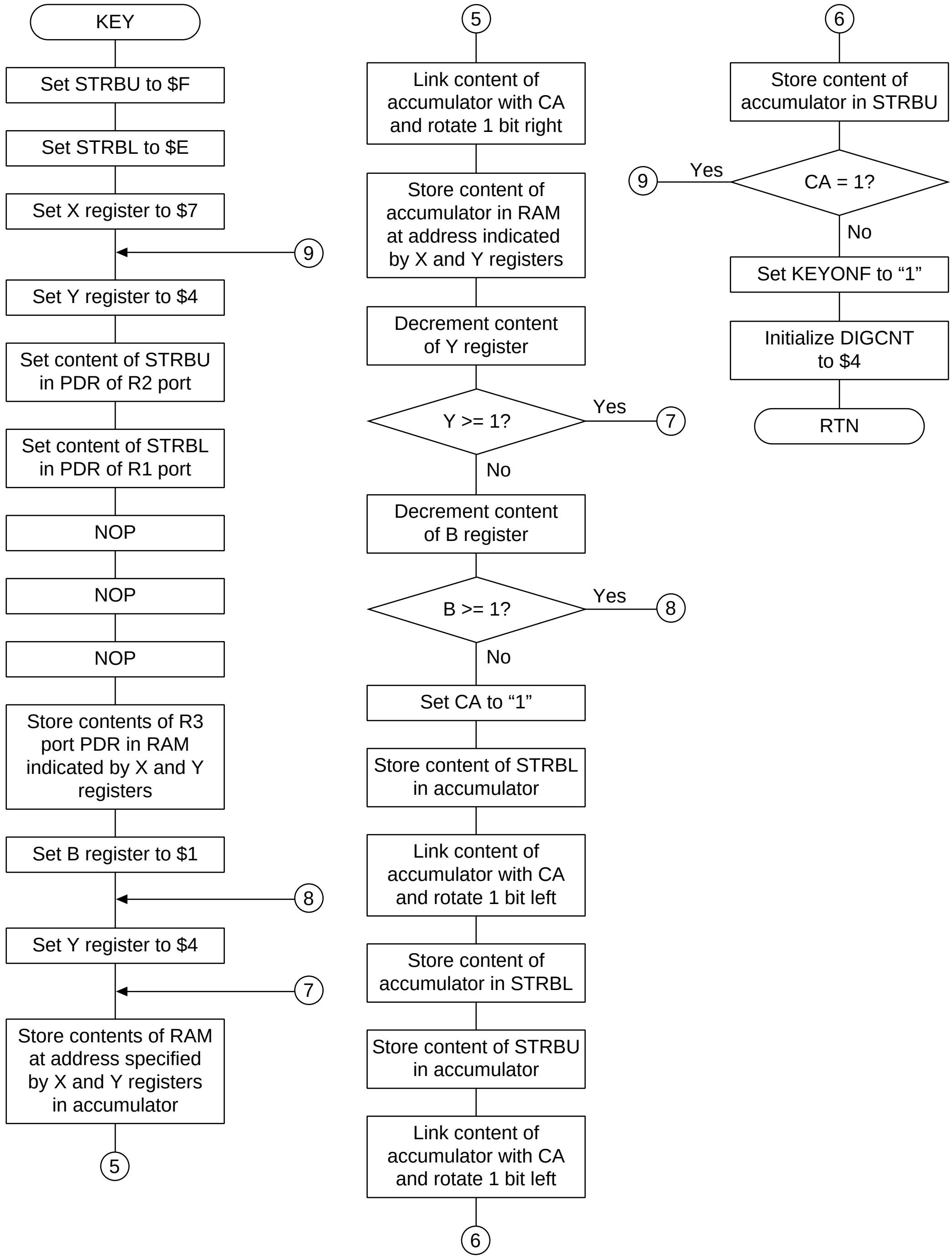
4. Timer A Interrupt Processing Routine



5. LED Control



6. Key Scan



RENESAS

Program Listing

1. H4318/H4359

```
***********************************************************
*
*          H400 Series Application Note
*          - Application Chapter -
*
*          'Keyscan & 8-segment LED Display'
*
*          Function
*          : Timer A Free Running Timer
*          : I/O Port (D0-D3, R1-R3 Port)
*
*          MCU : H4318/H4359
*
*          External Clock : 4MHz
*          Internal Clock : 1MHz
*
***********************************************************
*
***********************************************************
*          Symbol Definition
***********************************************************
*
IE         equ     0,$000      Interrupt Request Flag
RSP        equ     1,$000      Reset Stack Pointer
IF0        equ     2,$000      _INT0 Interrupt Request Flag
IM0        equ     3,$000      _INT0 Interrupt Mask
*
IF1        equ     0,$001      _INT1 Interrupt Request Flag
IM1        equ     1,$001      _INT1 Interrupt Mask
IFTA       equ     2,$001      Timer A Interrupt Request Flag
IMTA       equ     3,$001      Timer A Interrupt Mask
*
IFTB       equ     0,$002      Timer B Interrupt Request Flag
IMTB       equ     1,$002      Timer B Interrupt Mask
IFTC       equ     2,$002      Timer C Interrupt Request Flag
IMTC       equ     3,$002      Timer C Interrupt Mask
*
IFAD       equ     0,$003      A/D Converter Interrupt Request Flag
IMAD       equ     1,$003      A/D Converter Interrupt Mask
IFS        equ     2,$003      Serial Interrupt Request Flag
IMS        equ     3,$003      Serial Interrupt Mask
*
PMRA       equ     $004        Port Mode Register A
SMR        equ     $005        Serial Mode Register
SRL        equ     $006        Serial Data Register L
SRU        equ     $007        Serial Data Register U
TMA        equ     $008        Timer Mode Register A
```

```
TMB1     equ     $009        Timer Mode Register B1
TRBL     equ     $00A        Timer Read Register BL
TWBL     equ     $00A        Timer Write Register BL
TRBU     equ     $00B        Timer Read Register BU
TWBU     equ     $00B        Timer Write Register BU
MIS      equ     $00C        Miscellaneous register
TMC      equ     $00D        Timer Mode Register C
TRCL     equ     $00E        Timer Read Register CL
TWCL     equ     $00E        Timer Write Register CL
TRCU     equ     $00F        Timer Read Register CU
TWCU     equ     $00F        Timer Write Register CU
ACR      equ     $016        A/D Control Register
ADRL     equ     $017        A/D Data Register L
ADRU     equ     $018        A/D Data Register U
AMR1     equ     $019        A/D Mode Register 1
AMR2     equ     $01A        A/D Mode Register 2
*
WDON     equ     1,$020      Watchdog on Flag
ADSF     equ     2,$020      A/D Start Flag
*
ICSF     equ     0,$021      Input Capture Status Flag
ICEF     equ     1,$021      Input Capture Error Flag
IAOF     equ     2,$021      I_AD off Flag
RAME     equ     3,$021      RAM Enable Flag
*
PMRB     equ     $024        Port Mode Register B
PMRC     equ     $025        Port Mode Register C
TMB2     equ     $026        Timer Mode Register B2
DCD0     equ     $02C        Data Control Register D0
DCD1     equ     $02D        Data Control Register D1
DCD2     equ     $02E        Data Control Register D2
DCR0     equ     $030        Data Control Register R0
DCR1     equ     $031        Data Control Register R1
DCR2     equ     $032        Data Control register R2
DCR3     equ     $033        Data Control Register R3
DCR4     equ     $034        Data Control Register R4
DCR8     equ     $038        Data Control Register R8
*
************************************************************
*        Ram Allocation
************************************************************
*
AESC     equ     $040        A Escape RAM Area
BESC     equ     $041        B Escape RAM Area
WESC     equ     $042        W Escape RAM Area
XESC     equ     $043        X Escape RAM Area
YESC     equ     $044        Y Escape RAM Area
SXESC    equ     $045        SPX Escape RAM Area
SYESC    equ     $046        SPY Escape RAM Area
*
DIGCNT   equ     $054        LED Digit Counter
```

```
*
SEG0U    equ    $060        LED Display Data 0 Upper
SEG1U    equ    $061        LED Display Data 1 Upper
SEG2U    equ    $062        LED Display Data 2 Upper
SEG3U    equ    $063        LED Display Data 3 Upper
*
SEG0L    equ    $050        LED Display Data 0 Lower
SEG1L    equ    $051        LED Display Data 1 Lower
SEG2L    equ    $052        LED Display Data 2 Lower
SEG3L    equ    $053        LED Display Data 3 Lower
*
KEYFLG   equ    $080        Key Flag Area
KEYONF   equ    0,KEYFLG    Key on Flag
KDECONF  equ    1,KEYFLG    Key Decode on Flag
*
KONNEW   equ    $07F        Key New Data
KONOLD   equ    $07E        Key Old Data
CHATCNT  equ    $07D        Chattering Counter
KEYYADR  equ    $07A
STRBU    equ    $079        Key Strobe Data UpperPPER
STRBL    equ    $078        Key Strobe Data Lower
*
************************************************************
*        Vector Address
************************************************************
*
         org    $0000
*
         JMPL   LDKYMN      Reset Interrupt
         JMPL   LDKYMN      _INT0 Interrupt
         JMPL   LDKYMN      _INT1 Interrupt
         JMPL   LDKYINT     Timer A Interrupt
         JMPL   LDKYMN      Timer B Interrupt
         JMPL   LDKYMN      Timer C Interrupt
         JMPL   LDKYMN      A/D Converter Interrupt
         JMPL   LDKYMN      SCI Interrupt
*
************************************************************
*        LDKYMN : Main Program
************************************************************
*
         org    $1000
*
LDKYMN   REMD   RSP         Reset Stack Pointer
*
         LAI    $F
         LRA    $1          Initialize R1 Port PDR
         LRA    $2          Initialize R2 Port PDR
         REDD   $0          Initialize D0 Port PDR
         REDD   $1          Initialize D1 Port PDR
         REDD   $2          Initialize D2 Port PDR
```

```
          REDD    $3          Initialize D3 Port PDR
          LMID    $F,DCR1     Initialize R1 Port Terminal Function
          LMID    $F,DCR2     Initialize R2 Port Terminal Function
          LMID    $F,DCD0     Initialize D0-D3 Port Terminal Function
*
          LMID    $5,TMA      Initialize Timer A Input Clock Period
          REMD    IMTA        Timer A Interrupt Enable
          REMD    IFTA        Clear IFTA
*
          LMID    $0,DIGCNT   Initialize LED Digit Counter
          LMID    $0,KEYFLG   Initialize Key on Flag & Key Decode on Flag
          LMID    $0,CHATCNT  Initialize Chattering Counter
*
          LXI     $6          Initialize LED Display Data
          XSPX
          LXI     $5
          LYI     $3
          LAI     $F
LKMN10    LMAX
          LMADYX
          BRS     LKMN10
*
          SEMD    IE          Interrupt Enable
*
LKMN90    CALL    KEYDEC      Subroutine Jump to KEYDEC
          CALL    LEDDSP      Subroutine Jump to LEDDSP
          BRS     LKMN90      Branch to LKMN90
*
************************************************************
*         KEYDEC : Key Decoder
************************************************************
*
KEYDEC    TMD     KEYONF      KEYONF = "1" ? Keyscan End ?
          BRS     KD10        Yes. Branch to KD10
          BRS     KD90        NO. Branch to KD90
*
KD10      LBI     $F
          LXI     $7
KD12      LYI     $0
KD15      LAM                 Load Key Data
          ROTL                Rotate Left with Carry
          LMAIY               Save Key Data. Increment Y Register
          YNEI    $4          Y+1 != 0
          BRS     KD15        Yes. Branch to KD15
          TC                  No. CA = 1 ?
          BRS     KD16        Yes. Branch to KD16
          LAB
          LMAD    KONNEW      Save Key on New Data
          BRS     KD20
KD16      DB                  Decrement B Register. B >= 1 ?
          BRS     KD12        Yes. Branch to KD12
```

```
        BRS     KD90         No. Branch to KD90
*
KD20    LMAD    KONNEW       Load Key on New Data
        ANEMD   KONOLD       Key on New Data != Key on Old Data ?
        BRS     KD21         Yes. Branch to KD21
        LAMD    CHATCNT      Load Chattering Counter
        AI      $1           Increment Chattering Counter
        LMAD    CHATCNT      Save Chattering Counter
        ALEI    $2           A <= $2 ?
        BRS     KD90         Yes. Branch to KD90
        SEMD    KDECONF      Set Key Decode on Flag
        BRS     KD90         Branch to KD90
KD21    LMID    $0,CHATCNT   Initialize Chattering Counter
        LMAD    KONOLD       Save Key on Old Data
*
KD90    REMD    KEYONF       Clear Key on Flag
        RTN                  Return
*
************************************************************
*       LEDDSP : LED Display
************************************************************
*
LEDDSP  TMD     KDECONF      KDECONF = 1 ?
        BRS     LDSP10       Yes. Branch to LDSP10
        BRS     LDSP90       No. Branch to LDSP90
*
LDSP10  REMD    KDECONF      Clear Key Decode on Flag
*
        LXI     $6           Initialize LED Display Data
        XSPX
        LXI     $5
        LYI     $3
        LAI     $F
LDSP20  LMAX
        LMADYX               End ?
        BRS     LDSP20       No. Branch to LDSP20
*
        LAMD    KONNEW       Load Key Data
        LBI     $0
        P       $F           Pattern Generation
        LMAD    SEG0L        Save LED Display Data Lower
        LAB
        LMAD    SEG0U        Save LED Display Data Upper
*
LDSP90  RTN                  Return from Subroutine
*
************************************************************
*       LDKYINT : Timer A Interrupt Routine
************************************************************
*
LDKYINT REMD    IFTA         Clear Timer A Interrupt Request Flag
```

```
*
          LMAD    AESC        Store Accumulator
          LAB
          LMAD    BESC        Store B Register
          XSPX
          LASPX
          LMAD    XESC        Store X Register
          XSPX
          LAY
          LMAD    YESC        Store Y Register
*
          BR      *+1         Set Status Flag
          CALL    LED         Subroutine Jump to 'LED'
          CALL    KEY         Subroutine Jump to 'KEY'
*
          LAMD    YESC        Restore Y Register
          LYA
          LAMD    XESC        Restore X Register
          LXA
          LAMD    BESC        Restore B Register
          LBA
          LAMD    AESC        Restore Accumulator
*
          RTNI                Return from Interrupt
*
************************************************************
*         LED : LED Control
************************************************************
*
LED       INEMD   $4,DIGCNT   DIGCNT = 4 ?
          BRS     LED00       No. Branch to LED00
          LYI     $3          Yes. Initialize DIGCNT
          BRS     LED10       Branch to LED10
LED00     LAMD    DIGCNT      Load DIGCNT
          LYA
          RED                 Turn off Now Digit
          DY                  D >= 1 ?
          BRS     LED10       Yes. Branch to LED10
          SEC                 No. Set CA
          BRS     LED20       Branch to LED20
LED10     LXI     $5          Load LED Display Data
          LAM
          LRA     $1          Output Display Data Lower to R1 Port
          LXI     $6
          LAM
          LRA     $2          Output Display Data Upper to R2 Port
          SED                 Turn on Next Digit
          LAY
          LMAD    DIGCNT      Save DIGCNT
          REC                 Reset CA
LED20     TC                  Test CA
```

```
*
        RTN                     Return from Subroutine
*
************************************************************
*       KEY : Key Scan
************************************************************
*
KEY     LMID    $F,STRBU        Initialize Strobe Data Upper
        LMID    $E,STRBL        Initialize Strobe Data Lower
        LXI     $7
*
KEY00   LYI     $4
        LAMD    STRBU           Load Strobe Data Upper
        LRA     $2              Output Strobe Data Upper to R2 Port
        LAMD    STRBL           Load Strobe Data Lower
        LRA     $1              Output Strobe Data Lower to R1 Port
        NOP                     Wait
        NOP                     Wait
        NOP                     Wait
        LAR     $3              Input Key Return Data
        LMA                     Save Key Return Data
*
        LBI     $1
KEY10   LYI     $4
KEY20   LAM                     Load Key Return Data
        ROTR                    Rotate Right with Carry
        LMADY                   Save Key Return Data. Y >= 1 ?
        BRS     KEY20           Yes. Branch to KEY20
        DB                      Decrement B Register. B >= 1 ?
        BRS     KEY10           Yes. Branch to KEY10
*
        SEC                     Set CA
        LAMD    STRBL           Load Strobe Data
        ROTL                    Rotate Left with Carry
        LMAD    STRBL           Save Strobe Data
        LAMD    STRBU           Load Strobe Data
        ROTL                    Rotate Left with Carry
        LMAD    STRBU           Save Strobe Data
        TC                      CA = 1 ?
        BRS     KEY00           Yes. Branch to KEY20
*
        SEMD    KEYONF          Set KEYONF
*
KEY90   LMID    $4,DIGCNT       Initialize DIGCNT
*
        RTN
*
************************************************************
*       LED Display Data Table
************************************************************
*
```

```
          org     $F00
*
          dc      $1C0        LED Display Data "0"
          dc      $1F9        LED Display Data "1"
          dc      $1A4        LED Display Data "2"
          dc      $1B0        LED Display Data "3"
          dc      $199        LED Display Data "4"
          dc      $192        LED Display Data "5"
          dc      $182        LED Display Data "6"
          dc      $1D8        LED Display Data "7"
          dc      $180        LED Display Data "8"
          dc      $190        LED Display Data "9"
          dc      $188        LED Display Data "A"
          dc      $183        LED Display Data "B"
          dc      $1C6        LED Display Data "C"
          dc      $1A1        LED Display Data "D"
          dc      $186        LED Display Data "E"
          dc      $18E        LED Display Data "F"
*
          end
```

2. H4369

```
**************************************************************
*
*         H400 Series Application Note
*         - Application Chapter -
*
*         'Keyscan & 8-segment LED Display'
*
*         Function
*         : Timer A Free Running Timer
*         : I/O Port (D0-D3, R1-R3 Port)
*
*         MCU : H4369
*
*         External Clock : 4MHz
*         Internal Clock : 1MHz
*         Sub Clock      : 32.768kHz
*
**************************************************************
*
**************************************************************
*         Symbol Definition
**************************************************************
*
IE        equ     0,$000      Interrupt Request Flag
RSP       equ     1,$000      Reset Stack Pointer
IF0       equ     2,$000      _INT0 Interrupt Request Flag
IM0       equ     3,$000      _INT0 Interrupt Mask
*
IF1       equ     0,$001      _INT1 Interrupt Request Flag
IM1       equ     1,$001      _INT1 Interrupt Mask
IFTA      equ     2,$001      Timer A Interrupt Request Flag
IMTA      equ     3,$001      Timer A Interrupt Mask
*
IFTB      equ     0,$002      Timer B Interrupt Request Flag
IMTB      equ     1,$002      Timer B Interrupt Mask
IFTC      equ     2,$002      Timer C Interrupt Request Flag
IMTC      equ     3,$002      Timer C Interrupt Mask
*
IFAD      equ     0,$003      A/D Converter Interrupt Request Flag
IMAD      equ     1,$003      A/D Converter Interrupt Mask
IFS       equ     2,$003      Serial Interrupt Request Flag
IMS       equ     3,$003      Serial Interrupt Mask
*
PMRA      equ     $004        Port Mode Register A
SMR       equ     $005        Serial Mode Register
SRL       equ     $006        Serial Data Register L
SRU       equ     $007        Serial Data Register U
TMA       equ     $008        Timer Mode Register A
TMB1      equ     $009        Timer Mode Register B1
```

```
TRBL      equ     $00A        Timer Read Register BL
TWBL      equ     $00A        Timer Write Register BL
TRBU      equ     $00B        Timer Read Register BU
TWBU      equ     $00B        Timer Write Register BU
MIS       equ     $00C        Miscellaneous register
TMC       equ     $00D        Timer Mode Register C
TRCL      equ     $00E        Timer Read Register CL
TWCL      equ     $00E        Timer Write Register CL
TRCU      equ     $00F        Timer Read Register CU
TWCU      equ     $00F        Timer Write Register CU
ACR       equ     $016        A/D Control Register
ADRL      equ     $017        A/D Data Register L
ADRU      equ     $018        A/D Data Register U
AMR1      equ     $019        A/D Mode Register 1
AMR2      equ     $01A        A/D Mode Register 2
*
LSON      equ     0,$020      LSON Flag
WDON      equ     1,$020      Watchdog on Flag
ADSF      equ     2,$020      A/D Start Flag
DTON      equ     3,$020      DTON Flag
*
ICSF      equ     0,$021      Input Capture Status Flag
ICEF      equ     1,$021      Input Capture Error Flag
IAOF      equ     2,$021      I_AD off Flag
RAME      equ     3,$021      RAM Enable Flag
*
PMRB      equ     $024        Port Mode Register B
PMRC      equ     $025        Port Mode Register C
TMB2      equ     $026        Timer Mode Register B2
SSR1      equ     $027        System Clock Selection Register 1
SSR2      equ     $028        System Clock Selection register 2
*
DCD0      equ     $02C        Data Control Register D0
DCD1      equ     $02D        Data Control Register D1
DCD2      equ     $02E        Data Control Register D2
DCD3      equ     $02F        Data Control Register D3
*
DCR0      equ     $030        Data Control Register R0
DCR1      equ     $031        Data Control Register R1
DCR2      equ     $032        Data Control register R2
DCR3      equ     $033        Data Control Register R3
DCR4      equ     $034        Data Control Register R4
DCR5      equ     $035        Data Control Register R5
DCR6      equ     $036        Data Control Register R6
DCR7      equ     $037        Data Control Register R7
DCR8      equ     $038        Data Control Register R8
DCR9      equ     $039        Data Control Register R9
*
*************************************************************
*         Ram Allocation
*************************************************************
```

```
*
AESC     equ    $040       A Escape RAM Area
BESC     equ    $041       B Escape RAM Area
WESC     equ    $042       W Escape RAM Area
XESC     equ    $043       X Escape RAM Area
YESC     equ    $044       Y Escape RAM Area
SXESC    equ    $045       SPX Escape RAM Area
SYESC    equ    $046       SPY Escape RAM Area
*
DIGCNT   equ    $054       LED Digit Counter
*
SEG0U    equ    $060       LED Display Data 0 Upper
SEG1U    equ    $061       LED Display Data 1 Upper
SEG2U    equ    $062       LED Display Data 2 Upper
SEG3U    equ    $063       LED Display Data 3 Upper
*
SEG0L    equ    $050       LED Display Data 0 Lower
SEG1L    equ    $051       LED Display Data 1 Lower
SEG2L    equ    $052       LED Display Data 2 Lower
SEG3L    equ    $053       LED Display Data 3 Lower
*
KEYFLG   equ    $080       Key Flag Area
KEYONF   equ    0,KEYFLG   Key on Flag
KDECONF  equ    1,KEYFLG   Key Decode on Flag
*
KONNEW   equ    $07F       Key New Data
KONOLD   equ    $07E       Key Old Data
CHATCNT  equ    $07D       Chattering Counter
KEYYADR  equ    $07A
STRBU    equ    $079       Key Strobe Data UpperPPER
STRBL    equ    $078       Key Strobe Data Lower
*
************************************************************
*        Vector Address
************************************************************
*
         org    $0000
*
         JMPL   LDKYMN     Reset Interrupt
         JMPL   LDKYMN     _INT0 Interrupt
         JMPL   LDKYMN     _INT1 Interrupt
         JMPL   LDKYINT    Timer A Interrupt
         JMPL   LDKYMN     Timer B Interrupt
         JMPL   LDKYMN     Timer C Interrupt
         JMPL   LDKYMN     A/D Converter Interrupt
         JMPL   LDKYMN     SCI Interrupt
*
************************************************************
*        LDKYMN : Main Program
************************************************************
*
```

```
         org     $1000
*
LDKYMN   REMD    RSP          Reset Stack Pointer
         LMID    $2,SSR1      Initialize System Clock
*
         LAI     $F
         LRA     $1           Initialize R1 Port PDR
         LRA     $2           Initialize R2 Port PDR
         REDD    $0           Initialize D0 Port PDR
         REDD    $1           Initialize D1 Port PDR
         REDD    $2           Initialize D2 Port PDR
         REDD    $3           Initialize D3 Port PDR
         LMID    $F,DCR1      Initialize R1 Port Terminal Function
         LMID    $F,DCR2      Initialize R2 Port Terminal Function
         LMID    $F,DCD0      Initialize D0-D3 Port Terminal Function
*
         LMID    $5,TMA       Initialize Timer A Input Clock Period
         REMD    IMTA         Timer A Interrupt Enable
         REMD    IFTA         Clear IFTA
*
         LMID    $0,DIGCNT    Initialize LED Digit Counter
         LMID    $0,KEYFLG    Initialize Key on Flag & Key Decode on Flag
         LMID    $0,CHATCNT   Initialize Chattering Counter
*
         LXI     $6           Initialize LED Display Data
         XSPX
         LXI     $5
         LYI     $3
         LAI     $F
LKMN10   LMAX
         LMADYX
         BRS     LKMN10
*
         SEMD    IE           Interrupt Enable
*
LKMN90   CALL    KEYDEC       Subroutine Jump to KEYDEC
         CALL    LEDDSP       Subroutine Jump to LEDDSP
         BRS     LKMN90       Branch to LKMN90
*
************************************************************
*        KEYDEC : Key Decoder
************************************************************
*
KEYDEC   TMD     KEYONF       KEYONF = "1" ? Keyscan End ?
         BRS     KD10         Yes. Branch to KD10
         BRS     KD90         NO. Branch to KD90
*
KD10     LBI     $F
         LXI     $7
KD12     LYI     $0
KD15     LAM                  Load Key Data
```

```
        ROTL                    Rotate Left with Carry
        LMAIY                   Save Key Data. Increment Y Register
        YNEI    $4              Y+1 != 0
        BRS     KD15            Yes. Branch to KD15
        TC                      No. CA = 1 ?
        BRS     KD16            Yes. Branch to KD16
        LAB
        LMAD    KONNEW          Save Key on New Data
        BRS     KD20
KD16    DB                      Decrement B Register. B >= 1 ?
        BRS     KD12            Yes. Branch to KD12
        BRS     KD90            No. Branch to KD90
*
KD20    LMAD    KONNEW          Load Key on New Data
        ANEMD   KONOLD          Key on New Data != Key on Old Data ?
        BRS     KD21            Yes. Branch to KD21
        LAMD    CHATCNT         Load Chattering Counter
        AI      $1              Increment Chattering Counter
        LMAD    CHATCNT         Save Chattering Counter
        ALEI    $2              A <= $2 ?
        BRS     KD90            Yes. Branch to KD90
        SEMD    KDECONF         Set Key Decode on Flag
        BRS     KD90            Branch to KD90
KD21    LMID    $0,CHATCNT      Initialize Chattering Counter
        LMAD    KONOLD          Save Key on Old Data
*
KD90    REMD    KEYONF          Clear Key on Flag
        RTN                     Return
*
************************************************************
*       LEDDSP : LED Display
************************************************************
*
LEDDSP  TMD     KDECONF         KDECONF = 1 ?
        BRS     LDSP10          Yes. Branch to LDSP10
        BRS     LDSP90          No. Branch to LDSP90
*
LDSP10  REMD    KDECONF         Clear Key Decode on Flag
*
        LXI     $6              Initialize LED Display Data
        XSPX
        LXI     $5
        LYI     $3
        LAI     $F
LDSP20  LMAX
        LMADYX                  End ?
        BRS     LDSP20          No. Branch to LDSP20
*
        LAMD    KONNEW          Load Key Data
        LBI     $0
        P       $F              Pattern Generation
```

```
         LMAD    SEG0L       Save LED Display Data Lower
         LAB
         LMAD    SEG0U       Save LED Display Data Upper
*
LDSP90   RTN                 Return from Subroutine
*
************************************************************
*        LDKYINT : Timer A Interrupt Routine
************************************************************
*
LDKYINT  REMD    IFTA        Clear Timer A Interrupt Request Flag
*
         LMAD    AESC        Store Accumulator
         LAB
         LMAD    BESC        Store B Register
         XSPX
         LASPX
         LMAD    XESC        Store X Register
         XSPX
         LAY
         LMAD    YESC        Store Y Register
*
         BR      *+1         Set Status Flag
         CALL    LED         Subroutine Jump to 'LED'
         CALL    KEY         Subroutine Jump to 'KEY'
*
         LAMD    YESC        Restore Y Register
         LYA
         LAMD    XESC        Restore X Register
         LXA
         LAMD    BESC        Restore B Register
         LBA
         LAMD    AESC        Restore Accumulator
*
         RTNI                Return from Interrupt
*
************************************************************
*        LED : LED Control
************************************************************
*
LED      INEMD   $4,DIGCNT   DIGCNT = 4 ?
         BRS     LED00       No. Branch to LED00
         LYI     $3          Yes. Initialize DIGCNT
         BRS     LED10       Branch to LED10
LED00    LAMD    DIGCNT      Load DIGCNT
         LYA
         RED                 Turn off Now Digit
         DY                  D >= 1 ?
         BRS     LED10       Yes. Branch to LED10
         SEC                 No. Set CA
         BRS     LED20       Branch to LED20
```

```
LED10   LXI     $5          Load LED Display Data
        LAM
        LRA     $1          Output Display Data Lower to R1 Port
        LXI     $6
        LAM
        LRA     $2          Output Display Data Upper to R2 Port
        SED                 Turn on Next Digit
        LAY
        LMAD    DIGCNT      Save DIGCNT
        REC                 Reset CA
LED20   TC                  Test CA
*
        RTN                 Return from Subroutine
*
************************************************************
*       KEY : Key Scan
************************************************************
*
KEY     LMID    $F,STRBU    Initialize Strobe Data Upper
        LMID    $E,STRBL    Initialize Strobe Data Lower
        LXI     $7
*
KEY00   LYI     $4
        LAMD    STRBU       Load Strobe Data Upper
        LRA     $2          Output Strobe Data Upper to R2 Port
        LAMD    STRBL       Load Strobe Data Lower
        LRA     $1          Output Strobe Data Lower to R1 Port
        NOP                 Wait
        NOP                 Wait
        NOP                 Wait
        LAR     $3          Input Key Return Data
        LMA                 Save Key Return Data
*
        LBI     $1
KEY10   LYI     $4
KEY20   LAM                 Load Key Return Data
        ROTR                Rotate Right with Carry
        LMADY               Save Key Return Data. Y >= 1 ?
        BRS     KEY20       Yes. Branch to KEY20
        DB                  Decrement B Register. B >= 1 ?
        BRS     KEY10       Yes. Branch to KEY10
*
        SEC                 Set CA
        LAMD    STRBL       Load Strobe Data
        ROTL                Rotate Left with Carry
        LMAD    STRBL       Save Strobe Data
        LAMD    STRBU       Load Strobe Data
        ROTL                Rotate Left with Carry
        LMAD    STRBU       Save Strobe Data
        TC                  CA = 1 ?
        BRS     KEY00       Yes. Branch to KEY20
```

```
*
         SEMD    KEYONF      Set KEYONF
*
KEY90    LMID    $4,DIGCNT   Initialize DIGCNT
*
         RTN
*
************************************************************
*        LED Display Data Table
************************************************************
*
         org     $F00
*
         dc      $1C0        LED Display Data "0"
         dc      $1F9        LED Display Data "1"
         dc      $1A4        LED Display Data "2"
         dc      $1B0        LED Display Data "3"
         dc      $199        LED Display Data "4"
         dc      $192        LED Display Data "5"
         dc      $182        LED Display Data "6"
         dc      $1D8        LED Display Data "7"
         dc      $180        LED Display Data "8"
         dc      $190        LED Display Data "9"
         dc      $188        LED Display Data "A"
         dc      $183        LED Display Data "B"
         dc      $1C6        LED Display Data "C"
         dc      $1A1        LED Display Data "D"
         dc      $186        LED Display Data "E"
         dc      $18E        LED Display Data "F"
*
         end
```

HMCS400 Series Application Note — Software —

Publication Date: 1st Edition, March1999
Published by: Electronic Devices Sales & Marketing Group
Semiconductor & Integrated Circuits Group
Hitachi, Ltd.
Edited by: Technical Documentation Group
UL Media Co., Ltd.