TinyPower™ I/O Flash 8-Bit MCU with LCD & EEPROM

HT69F30A/HT69F40A/HT69F50A

Revision: V1.00 Date: March 19, 2013

www.holtek.com

Table of Contents

Features

CPU Features

- Operating voltage:
  - f_{SYS}=4MHz: 2.2V~5.5V
  - f_{SYS}=8MHz: 2.4V~5.5V
  - f_{SYS}=12MHz: 2.7V~5.5V
  - f_{SYS}=20MHz: 4.5V~5.5V
- Up to 0.2μs instruction cycle with 20MHz system clock at V_{DD}=5V
- Power down and wake-up functions to reduce power consumption
- Oscillator types:
  - External Crystal – HXT
  - External 32.765kHz Crystal – LXT
  - External RC – ERC
  - External Clock – EC
  - Internal RC – HIRC
  - Internal 32kHz RC – LIRC
- Multi-mode operation: NORMAL, SLOW, IDLE and SLEEP
- Fully integrated internal 4MHz, 8MHz and 12MHz oscillator requires no external components
- All instructions executed in one or two instruction cycles
- Table read instructions
- 63 powerful instructions
- Up to 8-level subroutine nesting
- Bit manipulation instruction

Peripheral Features

- Flash Program Memory: 2K×16~8K×16
- Data Memory: 128×8~384×8
- EEPROM Memory: 64×8~128×8
- Watchdog Timer function
- Up to 52 bidirectional I/O lines
- LCD driver function
- Multiple pin-shared external interrupts
- Multiple Timer Module for time measure, input capture, compare match output, PWM output or single pulse output function
- Dual Time-Base functions for generation of fixed time interrupt signals
- Low voltage reset function
- Low voltage detect function
- Wide range of available package types

General Description

The HT69Fx0A series of devices are Flash Memory LCD type 8-bit high performance RISC architecture microcontrollers, designed for applications which require an LCD interface. Offering users the convenience of Flash Memory multi-programming features, these devices also include a wide range of functions and features. Other memory includes an area of RAM Data Memory.

Analog features include multiple and extremely flexible Timer Modules provide timing, pulse generation and PWM generation functions. Protective features such as an internal Watchdog Timer, Low Voltage Reset and Low Voltage Detector coupled with excellent noise immunity and ESD protection ensure that reliable operation is maintained in hostile electrical environments.

A full choice of internal and external oscillator functions are provided including fully integrated low and high speed system oscillators which requires no external components for their implementation. The ability to operate and switch dynamically between a range of operating modes using different clock sources gives users the ability to optimise microcontroller operation and minimise power consumption.

The inclusion of flexible I/O programming features, Time-Base functions along with many other features ensure that the devices will find excellent use in applications such as consumer products, handheld instruments, household appliances, electronically controlled tools and motor driving in addition to many others.

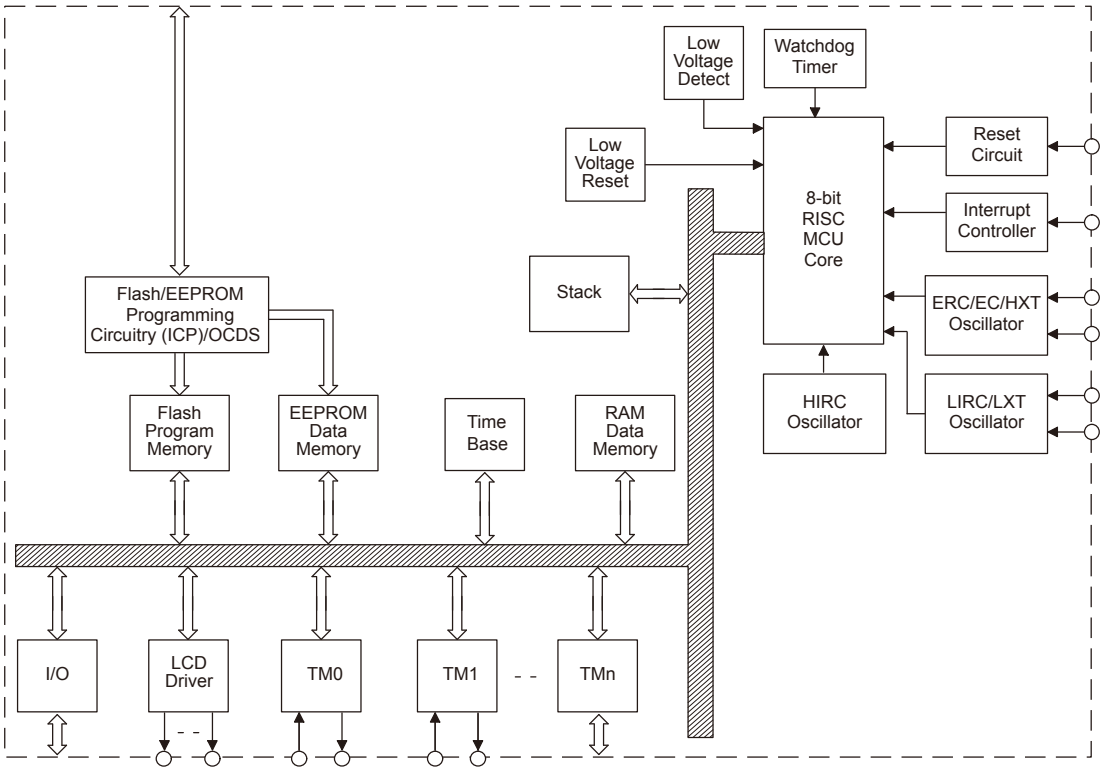
Selection Table

Most features are common to all devices. The main features distinguishing them are Memory capacity, I/O count, TM features, stack capacity, LCD driver and package types. The following table summarises the main features of each device.

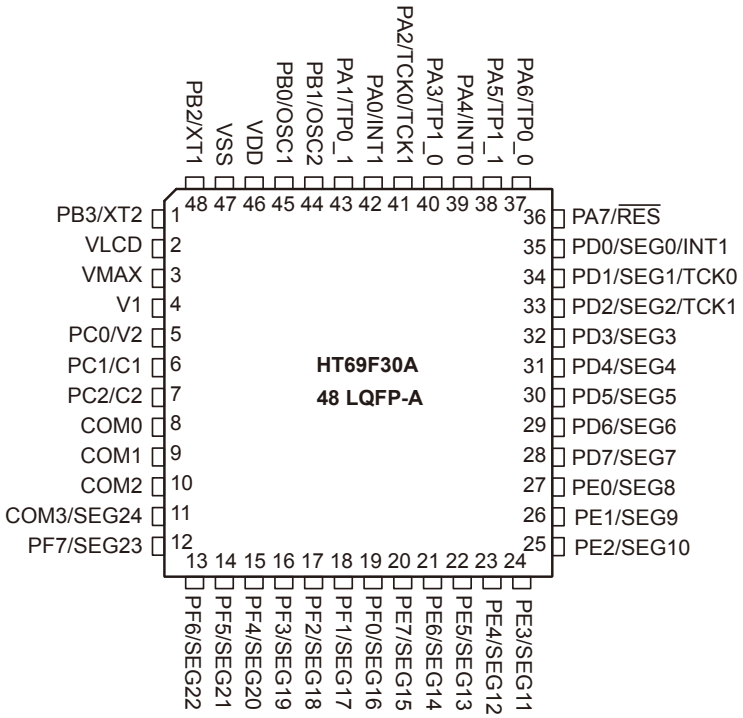
Part No.	Program Memory	Data Memory	Data EEPROM	I/O	External Interrupt	LCD Driver	Timer Module	Time Base	Stacks	Package
HT69F30A	2k×16	128×8	64×8	39	2	24×4 25×3	10-bit CTM×1 10-bit STM×1	2	4	48LQFP
HT69F40A	4k×16	256×8	128×8	51	2	36×4 37×3	10-bit CTM×1 10-bit STM×1 10-bit ETM×1	2	8	48/64LQFP
HT69F50A	8k×16	384×8	128×8	63	2	48×4 49×3	10-bit CTM×1 16-bit STM×1 10-bit ETM×1	2	8	48/64/80LQFP

Note: As devices exist in more than one package format, the table reflects the situation for the package with the most pins.

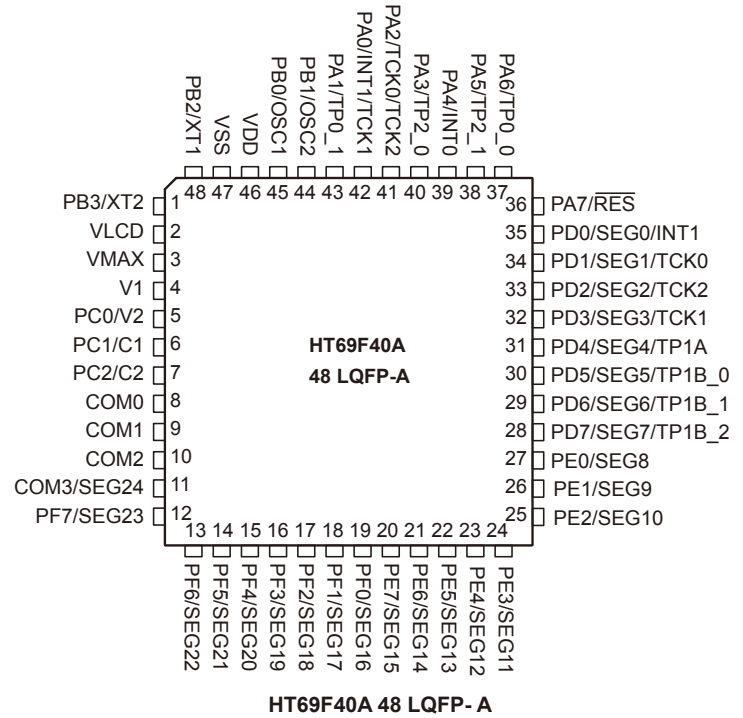
Block Diagram



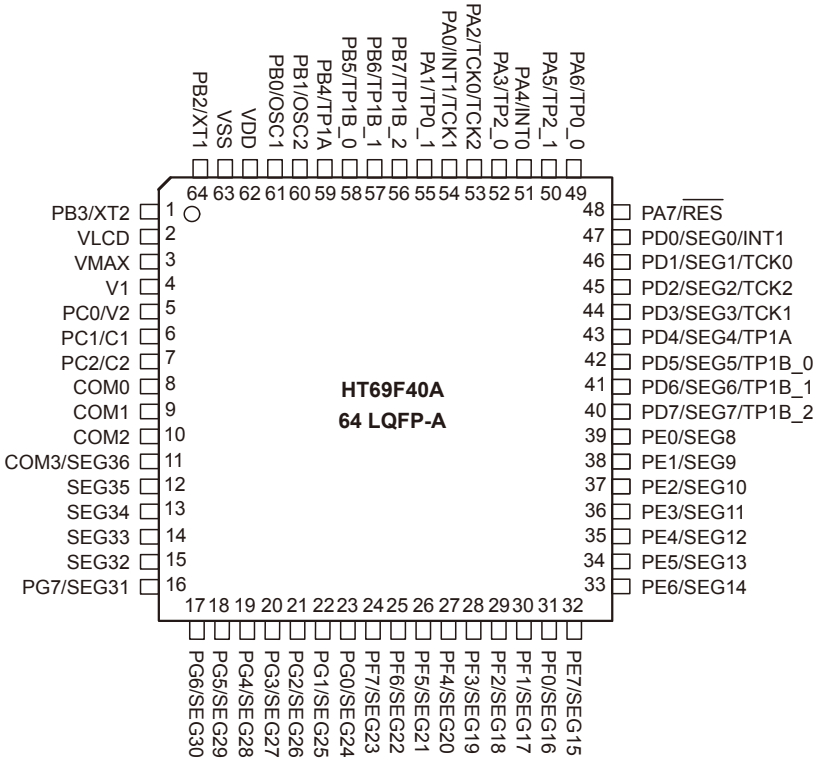
Pin Assignment



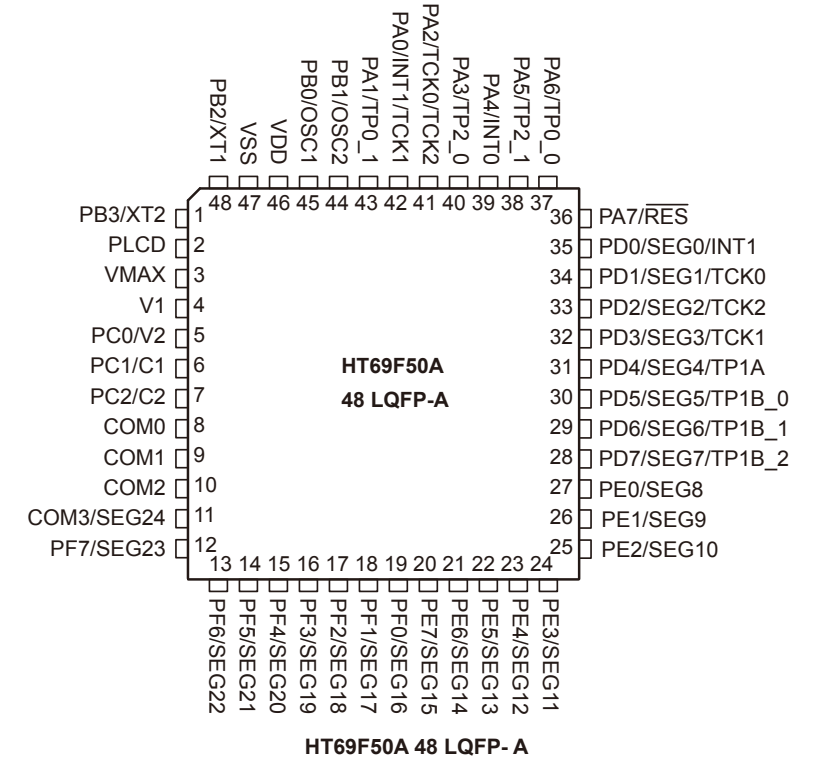
HT69F30A 48 LQFP- A



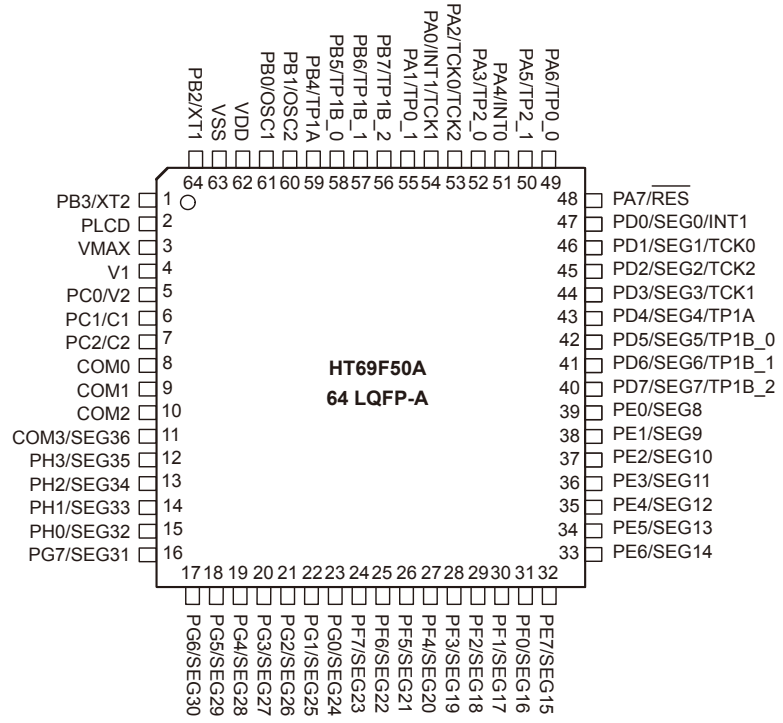
HT69F40A 48 LQFP- A



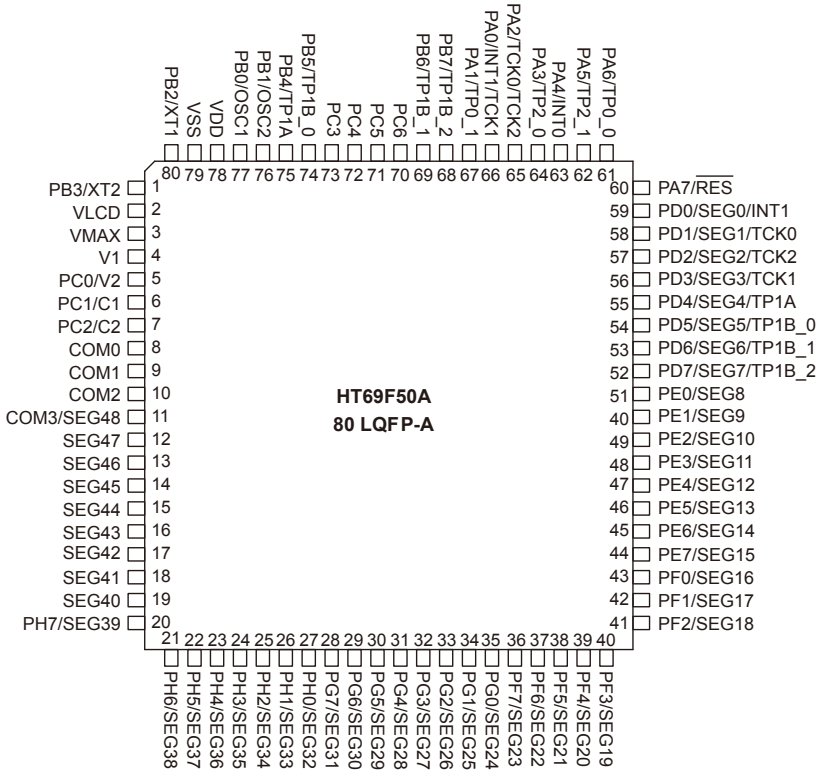
HT69F40A 64 LQFP- A



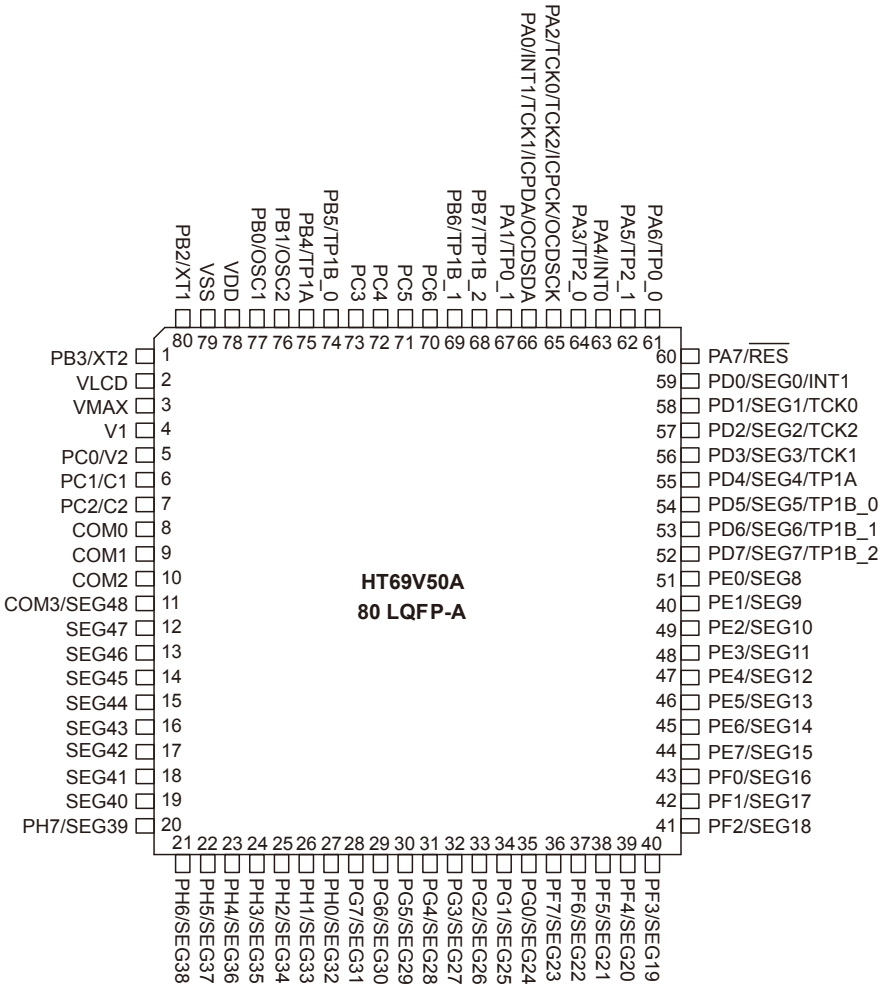
HT69F50A 48 LQFP- A



HT69F50A 64 LQFP- A



HT69F50A 80 LQFP- A



HT69V50A 80 LQFP- A

Note: 1. If the pin-shared pin functions have multiple outputs simultaneously, the output function is determined by the corresponding software control bits except the functions determined by the configuration options.

2. The HT69V50A device is the EV chip of the HT69Fx0A series of devices. It supports the "On-Chip Debug" function for debugging during development using the OCDSDA and OCDSCK pins connected to the Holtek HT-IDE development tools.

Pin Description

HT69F30A

Pad Name	Function	OPT	I/T	O/T	Description
PA0/INT1/ICPDA /OCDSDA	PA0	PAWU PAPU PAFS	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	INT1	INTEG INTC0 SFS	ST	—	External Interrupt 1
	ICPDA	—	ST	CMOS	ICP Data/Address
	OCDSDA	—	ST	CMOS	OCDS Data/Address
PA1/TP0_1	PA1	PAWU PAPU PAFS	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	TP0_1	PAFS	ST	CMOS	TM0 I/O pin
PA2/TCK0/TCK1 /ICPCK/OCDSCK	PA2	PAWU PAPU PAFS	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	TCK0	SFS	ST	—	TM0 input
	TCK1	SFS	ST	—	TM1 input
	ICPCK	—	ST	CMOS	ICP Clock pin
	OCDSCK	—	ST	—	OCDS Clock pin
PA3/TP1_0	PA3	PAWU PAPU PAFS	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	TP1_0	PAFS	ST	CMOS	TM1 I/O pin
PA4/INT0	PA4	PAPU PAWU	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	INT0	INTEG INTC0	ST	—	External Interrupt 0
PA5/TP1_1	PA5	PAWU PAPU PAFS	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	TP1_1	PAFS	ST	CMOS	TM1 I/O pin
PA6/TP0_0	PA6	PAWU PAPU PAFS	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	TP0_0	PAFS	ST	CMOS	TM0 I/O pin
PA7/$\overline{\text{RES}}$	PA7	PAWU PAPU PAFS	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	$\overline{\text{RES}}$	CO	ST	—	Reset pin
PB0/OSC1	PB0	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up
	OSC1	CO	HXT	—	HXT/ERC oscillator pin & EC mode input pin
PB1/OSC2	PB1	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up
	OSC2	CO	—	HXT	HXT oscillator pin
PB2/XT1	PB2	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up
	XT1	CO	LXT	—	LXT oscillator pin
PB3/XT2	PB3	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up
	XT2	CO	—	LXT	LXT oscillator pin

Pad Name	Function	OPT	I/T	O/T	Description
PC0/V2	PC0	PCPU PCFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	V2	PCFS	—	AO	LCD voltage pump
PC1/C1	PC1	PCPU PCFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	C1	PCFS	—	AO	LCD voltage pump
PC2/C2	PC2	PCPU PCFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	C2	PCFS	—	AO	LCD voltage pump
PD0/SEG0/INT1	PD0	PDPU PDFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG0	PDFS	—	AO	LCD segment output
	INT1	INTEG INTC0 SFS	ST	—	External Interrupt 1
PD1/SEG1/TCK0	PD1	PDPU PDFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG1	PDFS	—	AO	LCD segment output
	TCK0	SFS	ST	—	TM0 input
PD2/SEG2/TCK1	PD2	PDPU PDFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG2	PDFS	—	AO	LCD segment output
	TCK1	SFS	ST	—	TM1 input
PD3/SEG3	PD3	PDPU PDFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG3	PDFS	—	AO	LCD segment output
PD4/SEG4	PD4	PDPU PDFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG4	PDFS	—	AO	LCD segment output
PD5/SEG5	PD5	PDPU PDFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG5	PDFS	—	AO	LCD segment output
PD6/SEG6	PD6	PDPU PDFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG6	PDFS	—	AO	LCD segment output
PD7/SEG7	PD7	PDPU PDFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG7	PDFS	—	AO	LCD segment output
PE0/SEG8	PE0	PEPU PEFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG8	PEFS	—	AO	LCD segment output
PE1/SEG9	PE1	PEPU PEFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG9	PEFS	—	AO	LCD segment output
PE2/SEG10	PE2	PEPU PEFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG10	PEFS	—	AO	LCD segment output
PE3/SEG11	PE3	PEPU PEFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG11	PEFS	—	AO	LCD segment output

Pad Name	Function	OPT	I/T	O/T	Description
PE4/SEG12	PE4	PEPU PEFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG12	PEFS	—	AO	LCD segment output
PE5/SEG13	PE5	PEPU PEFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG13	PEFS	—	AO	LCD segment output
PE6/SEG14	PE6	PEPU PEFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG14	PEFS	—	AO	LCD segment output
PE7/SEG15	PE7	PEPU PEFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG15	PEFS	—	AO	LCD segment output
PF0/SEG16	PF0	PFPU PFFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG16	PFFS	—	AO	LCD segment output
PF1/SEG17	PF1	PFPU PFFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG17	PFFS	—	AO	LCD segment output
PF2/SEG18	PF2	PFPU PFFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG18	PFFS	—	AO	LCD segment output
PF3/SEG19	PF3	PFPU PFFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG19	PFFS	—	AO	LCD segment output
PF4/SEG20	PF4	PFPU PFFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG20	PFFS	—	AO	LCD segment output
PF5/SEG21	PF5	PFPU PFFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG21	PFFS	—	AO	LCD segment output
PF6/SEG22	PF6	PFPU PFFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG22	PFFS	—	AO	LCD segment output
PF7/SEG23	PF7	PFPU PFFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG23	PFFS	—	AO	LCD segment output
COM0~COM2	COMn	—	—	AO	LCD common outputs
COM3/SEG24	COM3	—	—	AO	LCD common output
	SEG24	LCDC	—	AO	LCD segment output
V1	V1	—	—	AO	LCD voltage pump
VLCD	VLCD	—	PWR	—	LCD power supply
VMAX	VMAX	—	PWR	—	IC maximum voltage, connected to VDD, VLCD or V1
VDD	VDD	—	PWR	—	Positive Power supply
VSS	VSS	—	PWR	—	Negative Power supply. Ground

Note: I/T: Input type; O/T: Output type
OPT: Optional by configuration option (CO) or register option
PWR: Power; CO: Configuration option
ST: Schmitt Trigger input; AO: Analog output
CMOS: CMOS output; NMOS: NMOS output
HXT: High frequency crystal oscillator
LXT: High frequency crystal oscillator

HT69F40A

Pad Name	Function	OPT	I/T	O/T	Description
PA0/INT1 /TCK1/ICPDA /OCDSDA	PA0	PAWU PAPU PAFS	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	INT1	INTEG INTC0 SFS	ST	—	External Interrupt 1
	TCK1	SFS	ST	—	TM1 input
	ICPDA	—	ST	CMOS	ICP Data/Address
	OCDSDA	—	ST	CMOS	OCDS Data/Address
PA1/TP0_1	PA1	PAWU PAPU PAFS	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	TP0_1	PAFS	ST	CMOS	TM0 I/O pin
PA2/TCK0 /TCK2/ICPCK /OCDSCK	PA2	PAWU PAPU PAFS	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	TCK0	SFS	ST	—	TM0 input
	TCK2	SFS	ST	—	TM2 input
	ICPCK	—	ST	CMOS	ICP Clock pin
	OCDSCK	—	ST	—	OCDS Clock pin
PA3/TP2_0	PA3	PAWU PAPU PAFS	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	TP2_0	PAFS	ST	CMOS	TM2 I/O pin
PA4/INT0	PA4	PAPU PAWU	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	INT0	INTEG INTC0	ST	—	External Interrupt 0
PA5/TP2_1	PA5	PAWU PAPU PAFS	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	TP2_1	PAFS	ST	CMOS	TM2 I/O pin
PA6/TP0_0	PA6	PAWU PAPU PAFS	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	TP0_0	PAFS	ST	CMOS	TM0 I/O pin
PA7/$\overline{\text{RES}}$	PA7	PAWU PAPU PAFS	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	$\overline{\text{RES}}$	CO	ST	—	Reset pin
PB0/OSC1	PB0	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up
	OSC1	CO	HXT	—	HXT/ERC oscillator pin & EC mode input pin
PB1/OSC2	PB1	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up
	OSC2	CO	—	HXT	HXT oscillator pin
PB2/XT1	PB2	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up
	XT1	CO	LXT	—	LXT oscillator pin
PB3/XT2	PB3	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up
	XT2	CO	—	LXT	LXT oscillator pin
PB4/TP1A	PB1	PBPU PBFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	TP1A	PBFS	ST	CMOS	TM1 I/O pin

Pad Name	Function	OPT	I/T	O/T	Description
PB5/TP1B_0	PB5	PBPU PBFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	TP1B_0	PBFS	ST	CMOS	TM1 I/O pin
PB6/TP1B_1	PB6	PBPU PBFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	TP1B_1	PBFS	ST	CMOS	TM1 I/O pin
PB7/TP1B_2	PB7	PBPU PBFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	TP1B_2	PBFS	ST	CMOS	TM1 I/O pin
PC0/V2	PC0	PCPU PCFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	V2	PCFS	—	AO	LCD voltage pump
PC1/C1	PC1	PCPU PCFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	C1	PCFS	—	AO	LCD voltage pump
PC2/C2	PC2	PCPU PCFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	C2	PCFS	—	AO	LCD voltage pump
PD0/SEG0/INT1	PD0	PDPU PDFS	ST	CMOS	General purpose I/O. Register enabled pull-up.
	SEG0	PDFS	—	AO	LCD segment output
	INT1	INTEG INTC0 SFS	ST	—	External Interrupt 1
PD1/SEG1/TCK0	PD1	PDPU PDFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG1	PDFS	—	AO	LCD segment output
	TCK0	SFS	ST	—	TM0 input
PD2/SEG2/TCK2	PD2	PDPU PDFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG2	PDFS	—	AO	LCD segment output
	TCK2	SFS	ST	—	TM2 input
PD3/SEG3/TCK1	PD3	PDPU PDFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG3	PDFS	—	AO	LCD segment output
	TCK1	SFS	ST	—	TM1 input
PD4/SEG4/TP1A	PD4	PDPU PDFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG4	PDFS SFS	—	AO	LCD segment output
	TP1A	PDFS SFS	ST	CMOS	TM1 I/O pin
PD5/SEG5 /TP1B_0	PD5	PDPU PDFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG5	PDFS SFS	—	AO	LCD segment output
	TP1B_0	PDFS SFS	ST	CMOS	TM1 I/O pin

Pad Name	Function	OPT	I/T	O/T	Description
PD6/SEG6 /TP1B_1	PD6	PDPU PDFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG6	PDFS SFS	—	AO	LCD segment output
	TP1B_1	PDFS SFS	ST	CMOS	TM1 I/O pin
PD7/SEG7 /TP1B_2	PD7	PDPU PDFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG7	PDFS SFS	—	AO	LCD segment output
	TP1B_2	PDFS SFS	ST	CMOS	TM1 I/O pin
PE0/SEG8	PE0	PEPU PEFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG8	PEFS	—	AO	LCD segment output
PE1/SEG9	PE1	PEPU PEFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG9	PEFS	—	AO	LCD segment output
PE2/SEG10	PE2	PEPU PEFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG10	PEFS	—	AO	LCD segment output
PE3/SEG11	PE3	PEPU PEFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG11	PEFS	—	AO	LCD segment output
PE4/SEG12	PE4	PEPU PEFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG12	PEFS	—	AO	LCD segment output
PE5/SEG13	PE5	PEPU PEFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG13	PEFS	—	AO	LCD segment output
PE6/SEG14	PE6	PEPU PEFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG14	PEFS	—	AO	LCD segment output
PE7/SEG15	PE7	PEPU PEFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG15	PEFS	—	AO	LCD segment output
PF0/SEG16	PF0	PFPU PFFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG16	PFFS	—	AO	LCD segment output
PF1/SEG17	PF1	PFPU PFFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG17	PFFS	—	AO	LCD segment output
PF2/SEG18	PF2	PFPU PFFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG18	PFFS	—	AO	LCD segment output
PF3/SEG19	PF3	PFPU PFFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG19	PFFS	—	AO	LCD segment output
PF4/SEG20	PF4	PFPU PFFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG20	PFFS	—	AO	LCD segment output

Pad Name	Function	OPT	I/T	O/T	Description
PF5/SEG21	PF5	PFPU PFFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG21	PFFS	—	AO	LCD segment output
PF6/SEG22	PF6	PFPU PFFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG22	PFFS	—	AO	LCD segment output
PF7/SEG23	PF7	PFPU PFFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG23	PFFS	—	AO	LCD segment output
PG0/SEG24	PG0	PGPU PGFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG24	PGFS	—	AO	LCD segment output
PG1/SEG25	PG1	PGPU PGFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG25	PGFS	—	AO	LCD segment output
PG2/SEG26	PG2	PGPU PGFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG26	PGFS	—	AO	LCD segment output
PG3/SEG27	PG3	PGPU PGFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG27	PGFS	—	AO	LCD segment output
PG4/SEG28	PG4	PGPU PGFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG28	PGFS	—	AO	LCD segment output
PG5/SEG29	PG5	PGPU PGFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG29	PGFS	—	AO	LCD segment output
PG6/SEG30	PG6	PGPU PGFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG30	PGFS	—	AO	LCD segment output
PG7/SEG31	PG7	PGPU PGFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG31	PGFS	—	AO	LCD segment output
SEG32~SEG35	SEGn	—	—	AO	LCD segment outputs
COM0~COM2	COMn	—	—	AO	LCD common outputs
COM3/SEG36	COM3	—	—	AO	LCD common output
	SEG36	LCDC	—	AO	LCD segment output
V1	V1	—	—	AO	LCD voltage pump
VLCD	VLCD	—	PWR	—	LCD power supply
VMAX	VMAX	—	PWR	—	IC maximum voltage, connected to VDD, VLCD or V1
VDD	VDD	—	PWR	—	Positive Power supply
VSS	VSS	—	PWR	—	Negative Power supply. Ground

Note: I/T: Input type; O/T: Output type
OPT: Optional by configuration option (CO) or register option
PWR: Power; CO: Configuration option
ST: Schmitt Trigger input; AO: Analog output
CMOS: CMOS output; NMOS: NMOS output
HXT: High frequency crystal oscillator
LXT: High frequency crystal oscillator

HT69F50A

Pad Name	Function	OPT	I/T	O/T	Description
PA0/INT1 /TCK1/ICPDA /OCDSDA	PA0	PAWU PAPU PAFS	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	INT1	INTEG INTC0 SFS	ST	—	External Interrupt 1
	TCK1	SFS	ST	—	TM1 input
	ICPDA	—	ST	CMOS	ICP Data/Address
	OCDSDA	—	ST	CMOS	OCDS Data/Address
PA1/TP0_1	PA1	PAWU PAPU PAFS	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	TP0_1	PAFS	ST	CMOS	TM0 I/O pin
PA2/TCK0 /TCK2/ICPCK /OCDSCK	PA2	PAWU PAPU PAFS	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	TCK0	SFS	ST	—	TM0 input
	TCK2	SFS	ST	—	TM2 input
	ICPCK	—	ST	CMOS	ICP Clock pin
	OCDSCK	—	ST	—	OCDS Clock pin
PA3/TP2_0	PA3	PAWU PAPU PAFS	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	TP2_0	PAFS	ST	CMOS	TM2 I/O pin
PA4/INT0	PA4	PAPU PAWU	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	INT0	INTEG INTC0	ST	—	External Interrupt 0
PA5/TP2_1	PA5	PAWU PAPU PAFS	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	TP2_1	PAFS	ST	CMOS	TM2 I/O pin
PA6/TP0_0	PA6	PAWU PAPU PAFS	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	TP0_0	PAFS	ST	CMOS	TM0 I/O pin
PA7/$\overline{\text{RES}}$	PA7	PAWU PAPU PAFS	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	$\overline{\text{RES}}$	CO	ST	—	Reset pin
PB0/OSC1	PB0	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up
	OSC1	CO	HXT	—	HXT/ERC oscillator pin & EC mode input pin
PB1/OSC2	PB1	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up
	OSC2	CO	—	HXT	HXT oscillator pin
PB2/XT1	PB2	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up
	XT1	CO	LXT	—	LXT oscillator pin
PB3/XT2	PB3	PBPU	ST	CMOS	General purpose I/O. Register enabled pull-up
	XT2	CO	—	LXT	LXT oscillator pin
PB4/TP1A	PB1	PBPU PBFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	TP1A	PBFS	ST	CMOS	TM1 I/O pin

Pad Name	Function	OPT	I/T	O/T	Description
PB5/TP1B_0	PB5	PBPU PBFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	TP1B_0	PBFS	ST	CMOS	TM1 I/O pin
PB6/TP1B_1	PB6	PBPU PBFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	TP1B_1	PBFS	ST	CMOS	TM1 I/O pin
PB7/TP1B_2	PB7	PBPU PBFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	TP1B_2	PBFS	ST	CMOS	TM1 I/O pin
PC0/V2	PC0	PCPU PCFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	V2	PCFS	—	AO	LCD voltage pump
PC1/C1	PC1	PCPU PCFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	C1	PCFS	—	AO	LCD voltage pump
PC2/C2	PC2	PCPU PCFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	C2	PCFS	—	AO	LCD voltage pump
PC3~PC6	PCn	PCPU	ST	CMOS	General purpose I/O. Register enabled pull-up
PD0/SEG0/INT1	PD0	PDPU PDFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG0	PDFS	—	AO	LCD segment output
	INT1	INTEG INTC0 SFS	ST	—	External Interrupt 1
PD1/SEG1/TCK0	PD1	PDPU PDFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG1	PDFS	—	AO	LCD segment output
	TCK0	SFS	ST	—	TM0 input
PD2/SEG2/TCK2	PD2	PDPU PDFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG2	PDFS	—	AO	LCD segment output
	TCK2	SFS	ST	—	TM2 input
PD3/SEG3/TCK1	PD3	PDPU PDFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG3	PDFS	—	AO	LCD segment output
	TCK1	SFS	ST	—	TM1 input
PD4/SEG4/TP1A	PD4	PDPU PDFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG4	PDFS SFS	—	AO	LCD segment output
	TP1A	PDFS SFS	ST	CMOS	TM1 I/O pin
PD5/SEG5 /TP1B_0	PD5	PDPU PDFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG5	PDFS SFS	—	AO	LCD segment output
	TP1B_0	PDFS SFS	ST	CMOS	TM1 I/O pin

Pad Name	Function	OPT	I/T	O/T	Description
PD6/SEG6 /TP1B_1	PD6	PDPU PDFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG6	PDFS SFS	—	AO	LCD segment output
	TP1B_1	PDFS SFS	ST	CMOS	TM1 I/O pin
PD7/SEG7 /TP1B_2	PD7	PDPU PDFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG7	PDFS SFS	—	AO	LCD segment output
	TP1B_2	PDFS SFS	ST	CMOS	TM1 I/O pin
PE0/SEG8	PE0	PEPU PEFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG8	PEFS	—	AO	LCD segment output
PE1/SEG9	PE1	PEPU PEFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG9	PEFS	—	AO	LCD segment output
PE2/SEG10	PE2	PEPU PEFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG10	PEFS	—	AO	LCD segment output
PE3/SEG11	PE3	PEPU PEFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG11	PEFS	—	AO	LCD segment output
PE4/SEG12	PE4	PEPU PEFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG12	PEFS	—	AO	LCD segment output
PE5/SEG13	PE5	PEPU PEFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG13	PEFS	—	AO	LCD segment output
PE6/SEG14	PE6	PEPU PEFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG14	PEFS	—	AO	LCD segment output
PE7/SEG15	PE7	PEPU PEFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG15	PEFS	—	AO	LCD segment output
PF0/SEG16	PF0	PFPU PFFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG16	PFFS	—	AO	LCD segment output
PF1/SEG17	PF1	PFPU PFFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG17	PFFS	—	AO	LCD segment output
PF2/SEG18	PF2	PFPU PFFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG18	PFFS	—	AO	LCD segment output
PF3/SEG19	PF3	PFPU PFFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG19	PFFS	—	AO	LCD segment output
PF4/SEG20	PF4	PFPU PFFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG20	PFFS	—	AO	LCD segment output

Pad Name	Function	OPT	I/T	O/T	Description
PF5/SEG21	PF5	PFPU PFFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG21	PFFS	—	AO	LCD segment output
PF6/SEG22	PF6	PFPU PFFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG22	PFFS	—	AO	LCD segment output
PF7/SEG23	PF7	PFPU PFFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG23	PFFS	—	AO	LCD segment output
PG0/SEG24	PG0	PGPU PGFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG24	PGFS	—	AO	LCD segment output
PG1/SEG25	PG1	PGPU PGFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG25	PGFS	—	AO	LCD segment output
PG2/SEG26	PG2	PGPU PGFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG26	PGFS	—	AO	LCD segment output
PG3/SEG27	PG3	PGPU PGFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG27	PGFS	—	AO	LCD segment output
PG4/SEG28	PG4	PGPU PGFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG28	PGFS	—	AO	LCD segment output
PG5/SEG29	PG5	PGPU PGFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG29	PGFS	—	AO	LCD segment output
PG6/SEG30	PG6	PGPU PGFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG30	PGFS	—	AO	LCD segment output
PG7/SEG31	PG7	PGPU PGFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG31	PGFS	—	AO	LCD segment output
PH0/SEG32	PH0	PHPU PHFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG32	PHFS	—	AO	LCD segment output
PH1/SEG33	PH1	PHPU PHFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG33	PHFS	—	AO	LCD segment output
PH2/SEG34	PH2	PHPU PHFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG34	PHFS	—	AO	LCD segment output
PH3/SEG35	PH3	PHPU PHFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG35	PHFS	—	AO	LCD segment output
PH4/SEG36	PH4	PHPU PHFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG36	PHFS	—	AO	LCD segment output
PH5/SEG37	PH5	PHPU PHFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG37	PHFS	—	AO	LCD segment output

Pad Name	Function	OPT	I/T	O/T	Description
PH6/SEG38	PH6	PHPU PHFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG38	PHFS	—	AO	LCD segment output
PH7/SEG39	PH7	PHPU PHFS	ST	CMOS	General purpose I/O. Register enabled pull-up
	SEG39	PHFS	—	AO	LCD segment output
SEG40~SEG47	SEGn	—	—	AO	LCD segment outputs
COM0~COM2	COMn	—	—	AO	LCD common outputs
COM3/SEG48	COM3	—	—	AO	LCD common output
	SEG48	LCDC	—	AO	LCD segment output
V1	V1	—	—	AO	LCD voltage pump
VLCD	VLCD	—	PWR	—	LCD power supply
VMAX	VMAX	—	PWR	—	IC maximum voltage, connected to VDD, VLCD or V1
VDD	VDD	—	PWR	—	Positive Power supply
VSS	VSS	—	PWR	—	Negative Power supply. Ground

Note: I/T: Input type; O/T: Output type
OPT: Optional by configuration option (CO) or register option
PWR: Power; CO: Configuration option
ST: Schmitt Trigger input; AO: Analog output
CMOS: CMOS output; NMOS: NMOS output
HXT: High frequency crystal oscillator
LXT: High frequency crystal oscillator

Absolute Maximum Ratings

Supply Voltage ..... V_{SS}-0.3V to V_{SS}+6.0V
Input Voltage ..... V_{SS}-0.3V to V_{DD}+0.3V
I_{OL} Total 80mA
Total Power Dissipation 500mW
Storage Temperature ..... -50°C to 125°C
Operating Temperature ..... -40°C to 85°C
I_{OH} Total ..... -80mA

Note: These are stress ratings only. Stresses exceeding the range specified under "Absolute Maximum Ratings" may cause substantial damage to the device. Functional operation of this device at other conditions beyond those listed in the specification is not implied and prolonged exposure to extreme conditions may affect device reliability.

D.C. Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD1}	Operating Voltage (HXT)	—	f_{SYS}=4MHz	2.2	—	5.5	V
			f_{SYS}=8MHz	2.2	—	5.5	V
			f_{SYS}=12MHz	2.7	—	5.5	V
			f_{SYS}=16MHz	4.5	—	5.5	V
V_{DD2}	Operating Voltage (ERC)	—	f_{SYS}=4MHz	2.2	—	5.5	V
			f_{SYS}=8MHz	2.4	—	5.5	V
			f_{SYS}=12MHz	2.7	—	5.5	V
			f_{SYS}=16MHz	4.5	—	5.5	V
			f_{SYS}=20MHz	4.5	—	5.5	V
V_{DD3}	Operating Voltage (HIRC)	—	f_{SYS}=4MHz	2.2	—	5.5	V
			f_{SYS}=8MHz	2.2	—	5.5	V
			f_{SYS}=12MHz	2.7	—	5.5	V
I_{DD1}	Operating Current (HXT, $f_{SYS}=f_H$, $f_S=f_{SUB}=f_{RTC}$ or f_{LIRC})	3V	No load, f_H=455kHz, WDT enable	—	100	150	μA
		5V		—	280	450	μA
		3V	No load, f_H=1MHz, WDT enable	—	250	400	μA
		5V		—	500	1000	μA
		3V	No load, f_H=4MHz, WDT enable	—	450	700	μA
		5V		—	1000	1500	μA
		3V	No load, f_H=8MHz, WDT enable	—	0.8	1.5	mA
		5V		—	1.5	3.0	mA
		3V	No load, f_H=12MHz, WDT enable	—	1.5	2.5	mA
		5V		—	3.0	5.0	mA
		5V	No load, f_H=16MHz, WDT enable	—	4.0	6.0	mA
I_{DD2}	Operating Current (ERC, $f_{SYS}=f_H$, $f_S=f_{SUB}=f_{RTC}$ or f_{LIRC})	3V	No load, f_H=455kHz, WDT enable	—	66	110	μA
		5V		—	200	300	μA
		3V	No load, f_H=1MHz, WDT enable	—	250	400	μA
		5V		—	500	1000	μA
		3V	No load, f_H=4MHz, WDT enable	—	450	700	μA
		5V		—	1000	1500	μA
		3V	No load, f_H=8MHz, WDT enable	—	0.8	1.5	mA
		5V		—	1.5	3.0	mA
		3V	No load, f_H=12MHz, WDT enable	—	1.5	2.5	mA
		5V		—	3.0	5.0	mA
		5V	No load, f_H=16MHz, WDT enable	—	4.0	6.0	mA
I_{DD3}	Operating Current (HIRC OSC, $f_{SYS}=f_H$, $f_S=f_{SUB}=f_{RTC}$ or f_{LIRC})	3V	No load, f_H=4MHz, WDT enable	—	420	630	μA
		5V		—	700	1000	μA
		3V	No load, f_H=8MHz, WDT enable	—	0.8	1.5	mA
		5V		—	1.5	3.0	mA
		3V	No load, f_H=12MHz, WDT enable	—	1.5	2.5	mA
		5V		—	3.0	5.0	mA
I_{DD4}	Operating Current (EC, $f_{SYS}=f_H$, $f_S=f_{SUB}=f_{RTC}$ or f_{LIRC})	3V	No load, f_H=4MHz, WDT enable	—	330	500	μA
		5V		—	550	820	μA

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
I_{DD5}	Operating Current (HXT, $f_{SYS}=f_L$, $f_S=f_{SUB}=f_{RTC}$ or f_{LIRC})	3V	No load, f_H=8MHz, $f_L=f_H/2$, WDT enable	—	500	750	μA
		5V		—	800	1200	μA
		3V	No load, f_H=8MHz, $f_L=f_H/4$, WDT enable	—	420	630	μA
		5V		—	700	1000	μA
		3V	No load, f_H=8MHz, $f_L=f_H/8$, WDT enable	—	400	600	μA
		5V		—	600	800	μA
		3V	No load, f_H=8MHz, $f_L=f_H/16$, WDT enable	—	360	540	μA
		5V		—	560	700	μA
		3V	No load, f_H=8MHz, $f_L=f_H/32$, WDT enable	—	320	480	μA
		5V		—	520	650	μA
		3V	No load, f_H=8MHz, $f_L=f_H/64$, WDT enable	—	280	420	μA
		5V		—	440	600	μA
I_{DD6}	Operating Current (HXT, $f_{SYS}=f_L$, $f_S=f_{SUB}=f_{RTC}$ or f_{LIRC})	3V	No load, f_H=4MHz, $f_L=f_H/2$, WDT enable	—	270	400	μA
		5V	No load, f_H=4MHz, $f_L=f_H/2$, WDT enable	—	560	840	μA
		3V	No load, f_H=4MHz, $f_L=f_H/4$, WDT enable	—	180	270	μA
		5V	No load, f_H=4MHz, $f_L=f_H/4$, WDT enable	—	400	600	μA
I_{DD7}	Operating Current (HXT, $f_{SYS}=f_L$, $f_S=f_{SUB}=f_{RTC}$ or f_{LIRC})	3V	No load, f_H=12MHz, $f_L=f_H/2$, WDT enable	—	0.9	1.5	mA
		5V	No load, f_H=12MHz, $f_L=f_H/2$, WDT enable	—	1.8	2.7	mA
		3V	No load, f_H=12MHz, $f_L=f_H/4$, WDT enable	—	0.6	1.0	mA
		5V	No load, f_H=12MHz, $f_L=f_H/4$, WDT enable	—	0.9	1.4	mA
I_{DD8}	Operating Current (LXT, $f_{SYS}=f_L=f_{RTC}$, $f_S=f_{SUB}=f_{RTC}$) (HT69F30A, HT69F40A)	3V	No load, WDT enable, QOSC=0	—	10	20	μA
		5V		—	20	35	μA
		3V	No load, WDT enable, QOSC=1	—	10	20	μA
		5V		—	20	35	μA
I_{DD8a}	Operating Current (LXT, $f_{SYS}=f_L=f_{RTC}$, $f_S=f_{SUB}=f_{RTC}$) (HT69F50A)	3V	No load, WDT enable, QOSC=0	—	10	20	μA
		5V		—	30	50	μA
		3V	No load, WDT enable, QOSC=1	—	10	20	μA
		5V		—	30	50	μA
I_{DD9}	Operating Current (LIRC, $f_{SYS}=f_L=f_{LIRC}$, $f_S=f_{SUB}=f_{LIRC}$) (HT69F30A, HT69F40A)	3V	No load, WDT enable	—	10	20	μA
		5V		—	20	35	μA
I_{DD9a}	Operating Current (LIRC, $f_{SYS}=f_L=f_{LIRC}$, $f_S=f_{SUB}=f_{LIRC}$) (HT69F50A)	3V	No load, WDT enable	—	10	20	μA
		5V		—	30	50	μA
I_{DD10}	Operating Current (LXT or LIRC, $f_{SYS}=f_L=f_{SUB}$) (HT69F30A, HT69F40A)	3V	No load, WDT enable, QOSC=0	—	10	20	μA
		5V		—	20	35	μA
I_{DD10a}	Operating Current (LXT or LIRC, $f_{SYS}=f_L=f_{SUB}$) (HT69F50A)	3V	No load, WDT enable, QOSC=0	—	10	20	μA
		5V		—	30	50	μA
I_{STB1}	Standby Current (Idle1) (HXT, $f_{SYS}=f_H$, $f_S=f_{SUB}=f_{RTC}$ or f_{LIRC})	3V	No load, system HALT, WDT enable, f_{SYS}=8MHz, FSYSON=1	—	0.3	0.5	μA
		5V		—	0.5	0.8	μA
I_{STB2}	Standby Current (Idle0) (HXT, f_{SYS}=off, $f_S=f_{SUB}=f_{RTC}$ or f_{LIRC})	3V	No load, system HALT, WDT enable, f_{SYS}=8MHz, LXTLP=1	—	1.5	3.0	μA
		5V		—	2.5	5.0	μA
I_{STB3}	Standby Current (Idle0) (HIRC, f_{SYS}=off, $f_S=f_{SUB}=f_{LIRC}$)	3V	No load, system HALT, WDT enable, f_{SYS}=8MHz	—	1.5	3.0	μA
		5V		—	2.5	5.0	μA

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
I_{STB4}	Standby Current (Idle0.) (HIRC, f_{SYS}=off, f_S=f_{SUB}=f_{LIRC})	3V	No load, system HALT, WDT enable, f_{SYS}=32768Hz, LXTLP=1	—	1.5	3.0	μA
		5V		—	2.5	5.0	μA
I_{STB5}	Standby Current (Idle0) (LXT, f_{SYS}=off)	3V	No load, system HALT, WDT enable, f_{SYS}=32768Hz	—	1.5	3.0	μA
		5V		—	2.5	5.0	μA
I_{STB6}	Standby Current (Idle0) (f_{SYS}=f_L=f_{RTC}, f_S=f_{SUB}=f_{RTC})	3V	No load, system HALT, WDT enable, f_{SYS}=32768Hz, LXTLP=1	—	1.9	4.0	μA
		5V		—	3.3	7.0	μA
I_{STB7}	Standby Current (Idle) (LIRC, f_{SYS}=off, f_S=f_{SUB}=f_{LIRC})	3V	No load, system HALT, WDT enable, f_{SYS}=32kHz	—	1.5	3.0	μA
		5V		—	2.5	5.0	μA
I_{STB8}	Standby Current (Sleep0) (HXT, f_{SYS}=off, f_S=f_{SUB}=f_{RTC} or f_{LIRC})	3V	No load, system HALT, WDT disable, f_{SYS}=12MHz, LXTLP=1	—	0.1	1	μA
		5V		—	0.3	2	μA
I_{STB9}	Standby Current (Sleep1) (HXT, f_{SYS}=off, f_S=f_{SUB}=f_{RTC})	3V	No load, system HALT, WDT enable, f_{SYS}=12MHz	—	1.5	3.0	μA
		5V		—	2.5	5.0	μA
I_{STB10}	Standby Current (Sleep1) (HXT, f_{SYS}=off, f_S=f_{SUB}=f_{LIRC})	3V	No load, system HALT, WDT enable, f_{SYS}=12MHz	—	1.5	3.0	μA
		5V		—	2.5	5.0	μA
I_{STB11}	Standby Current (Sleep0) (LXT, f_{SYS}=off, f_S=f_{SUB}=f_{RTC})	3V	No load, system HALT, WDT disable, f_{SYS}=32768Hz	—	0.1	1	μA
		5V		—	0.3	2	μA
I_{STB12}	Standby Current (Sleep1) (LXT, f_{SYS}=off, f_S=f_{SUB}=f_{RTC})	3V	No load, system HALT, WDT enable, f_{SYS}=32768Hz, LXTLP=1	—	1.5	3.0	μA
		5V		—	2.5	5.0	μA
V_{IL1}	Input Low Voltage for PA, PB, PC, PD, PE, PF, TCKn and INTn	5	—	0V	—	1.5V	V
		—		0V	—	$0.2V_{DD}$	V
V_{IH1}	Input High Voltage for PA, PB, PC, PD, PE, PF, TCKn and INTn	5	—	3.5V	—	5V	V
		—		$0.8V_{DD}$	—	V_{DD}	V
V_{IL2}	Input Low Voltage ($\overline{RES}$)	—	—	0	—	$0.4V_{DD}$	V
V_{IH2}	Input High Voltage ($\overline{RES}$)	—	—	$0.9V_{DD}$	—	V_{DD}	V
I_{OL}	I/O Port Sink Current	3V	V_{OL}=$0.1V_{DD}$	4	8	—	mA
		5V	V_{OL}=$0.1V_{DD}$	10	20	—	mA
I_{OH}	I/O Port Source Current	3V	V_{OH}=$0.9V_{DD}$	-2	-4	—	mA
		5V	V_{OH}=$0.9V_{DD}$	-5	-10	—	mA
R_{PH}	Pull-high Resistance of I/O Ports	3V	—	20	60	100	kΩ
		5V	—	10	30	50	kΩ

A.C. Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Condition				
f_{CPU}	Operating Clock	—	2.2~5.5V	DC	—	4	MHz
			2.4~5.5V	DC	—	8	MHz
			2.7~5.5V	DC	—	12	MHz
			4.5~5.5V	DC	—	16	MHz
f_{SYS}	System clock (HXT)	2.2V~5.5V	—	0.4	—	4	MHz
		2.2V~5.5V		0.4	—	8	MHz
		2.7V~5.5V		0.4	—	12	MHz
		4.5V~5.5V		0.4	—	16	MHz
f_{HIRC}	System clock (HIRC)	3V/5V	Ta=25°C	-2%	4	+2%	MHz
		3V/5V	Ta=25°C	-2%	8	+2%	MHz
		5V	Ta=25°C	-2%	12	+2%	MHz
		2.2V~3.6V	Ta=-40°C~85°C	-8%	4	+8%	MHz
		3.0V~5.5V	Ta=-40°C~85°C	-8%	4	+8%	MHz
		2.2V~3.6V	Ta=-40°C~85°C	-8%	8	+8%	MHz
		3.0V~5.5V	Ta=-40°C~85°C	-8%	8	+8%	MHz
		3.0V~5.5V	Ta=-40°C~85°C	-8%	12	+8%	MHz
f_{ERC}	System clock (ERC)	5V	Ta=25°C External R_{ERC}=150kΩ	-2%	4	+2%	MHz
		5V	Ta=0°C~70°C External R_{ERC}=150kΩ	-5%	4	+5%	MHz
		5V	Ta=-40°C~85°C External R_{ERC}=150kΩ	-7%	4	+7%	MHz
		3.0V~5.5V	Ta=-40°C~85°C External R_{ERC}=150kΩ	-9%	4	+9%	MHz
		2.2V~5.5V	Ta=-40°C~85°C External R_{ERC}=150kΩ	-12%	4	+12%	MHz
f_{LXT}	System Clock (LXT)	—	—	—	32768	—	Hz
f_{LIRC}	System Clock (LIRC)	5V	Ta=25°C	-10%	32	+10%	kHz
		2.2V~5.5V	Ta=-40°C~85°C	-50%	32	+60%	kHz
t_{TIMER}	TCKn and timer capture Input Pulse Width	—	—	0.3	—	—	μs
t_{RES}	External Reset Low Pulse Width	—	—	10	—	—	μs
t_{INT}	Interrupt Pulse Width	—	—	10	—	—	μs
t_{SST}	System Start-up Timer Period (Wake-up from HALT, f_{SYS} off at HALT state, Slow Mode→Normal Mode, Normal Mode→Slow Mode)	—	f_{SYS}=HXT or LXT (Slow Mode→Normal Mode(HXT), Normal Mode→Slow Mode(LXT))	1024	—	—	t_{SYS}
			f_{SYS}=HXT or LXT (Wake-up from HALT, f_{SYS} off at HALT state)	1024	—	—	t_{SYS}
		—	f_{SYS}=ERC or HIRC	16	—	—	t_{SYS}
		—	f_{SYS}=LIRC or EC	2	—	—	t_{SYS}
	System Start-up Timer Period (Wake-up from HALT, f_{SYS} on at HALT state)	—	—	2	—	—	t_{SYS}

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Condition				
t_{RSTD}	System Reset Delay Time (Power On Reset, LVR reset, LVR S/W reset (LVRC), WDT S/W reset (WDTC))	—	—	25	50	100	ms
	System Reset Delay Time ($\overline{RES}$ reset, WDT normal reset)	—	—	8.3	16.7	33.3	ms
t_{BGS}	V_{BG} turn on stable time	—	—	10	—	—	ms
t_{EERD}	EEPROM Read Time	—	—	1	2	4	t_{SYS}
t_{EEWR}	EEPROM Write Timet	—	—	1	2	4	ms

Note: $t_{SYS}=1/f_{SYS}$

LVD & LVR Electrical Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{LVR1}	Low Voltage Reset Voltage	—	LVR Enable, 2.1V option	-5%	2.1	+5%	V
V_{LVR2}			LVR Enable, 2.55V option		2.55		
V_{LVR3}			LVR Enable, 3.15V option		3.15		
V_{LVR4}			LVR Enable, 3.8V option		3.8		
V_{LVD1}	Low Voltage Detector Voltage	—	LVDEN=1, V_{LVD}=2.0V	-5%	2.0	+5%	V
V_{LVD2}			LVDEN=1, V_{LVD}=2.2V		2.2		V
V_{LVD3}			LVDEN=1, V_{LVD}=2.4V		2.4		V
V_{LVD4}			LVDEN=1, V_{LVD}=2.7V		2.7		V
V_{LVD5}			LVDEN=1, V_{LVD}=3.0V		3.0		V
V_{LVD6}			LVDEN=1, V_{LVD}=3.3V		3.3		V
V_{LVD7}			LVDEN=1, V_{LVD}=3.6V		3.6		V
V_{LVD8}			LVDEN=1, V_{LVD}=4.0V		4.0		V
V_{BG}	Bandgap reference with buffer voltage	—	—	-3%	1.25	+3%	V
I_{BG}	Additional Power Consumption if bandgap reference with buffer is used	—	—	—	200	300	μA
I_{LVR}	Additional Power Consumption if LVR is used	3V	LVR disable→LVR enable	—	10	20	μA
		5V		—	15	30	μA
I_{LVD}	Additional Power Consumption if LVD is used	3V	LVD disable→LVD enable (LVR disable)	—	10	20	μA
		5V		—	15	30	μA
		3V	LVD disable→LVD enable (LVR enable)	—	1	2	μA
		5V		—	2	4	μA
t_{LVR}	Low Voltage Width to Reset	—	—	120	240	480	μs
t_{LVD}	Low Voltage Width to Interrupt	—	—	20	45	90	t_{LIRC}
t_{LVDS}	LVDO stable time	—	For LVR enable, LVD off→on	15	—	—	μs
		—	For LVR disable, LVD off→on	15	—	—	μs
t_{SRESET}	Software Reset Width to Reset	—	—	20	45	90	t_{LIRC}

LCD D.C. Characteristics

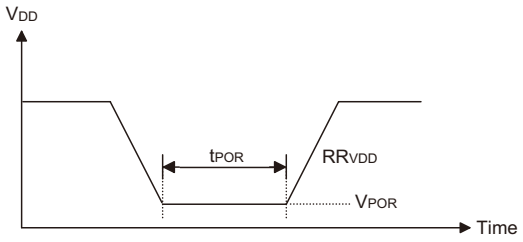
Ta=25°C

Symbol	Parameter	Test Condition		Min.	Typ.	Max.	Unit
		V_{DD}	Condition				
I_{STB1}	Standby Current (Sleep) (f_{SYS}, f_{WDT} off, f_S=f_{SUB}=32768 or 32K RC OSC)	3V	No load, system HALT, LCD on, WDT off, C type, V_{LCD}=V_{DD}, 1/2 Bias,	—	2.0	4.0	μA
		5V		—	3.0	5.0	μA
I_{STB2}	Standby Current (Sleep) (f_{SYS}, f_{WDT} off, f_S=f_{SUB}=32768 or 32K RC OSC)	3V	No load, system HALT, LCD on, WDT off, C type, V_{LCD}=V_{DD}, 1/3 Bias,	—	2.0	4.0	μA
		5V		—	3.0	5.0	μA
I_{STB3}	Standby Current (Idle) (f_{SYS}, f_{WDT} off, f_S=f_{SUB}=32768 or 32K RC OSC)	3V	No load, system HALT, LCD on, WDT off, R type, V_{LCD}=V_{DD}, 1/2 bias (I_{BIAS}=7.5μA)	—	13.5	20.0	μA
		5V		—	22.5	40.0	μA
I_{STB4}	Standby Current (Idle) (f_{SYS}, f_{WDT} off, f_S=f_{SUB}=32768 or 32K RC OSC)	3V	No load, system HALT, LCD on, WDT off, R type, V_{LCD}=V_{DD}, 1/2 bias (I_{BIAS}=15μA)	—	21	40	μA
		5V		—	35	60	μA
I_{STB5}	Standby Current (Idle) (f_{SYS}, f_{WDT} off, f_S=f_{SUB}=32768 or 32K RC OSC)	3V	No load, system HALT, LCD on, WDT off, R type, V_{LCD}=V_{DD}, 1/2 bias (I_{BIAS}=45μA)	—	51	80	μA
		5V		—	85	160	μA
I_{STB6}	Standby Current (Idle) (f_{SYS}, f_{WDT} off, f_S=f_{SUB}=32768 or 32K RC OSC)	3V	No load, system HALT, LCD on, WDT off, R type, V_{LCD}=V_{DD}, 1/2 bias (I_{BIAS}=90μA)	—	96	160	μA
		5V		—	160	320	μA
I_{STB7}	Standby Current (Idle) (f_{SYS}, f_{WDT} off, f_S=f_{SUB}=32768 or 32K RC OSC)	3V	No load, system HALT, LCD on, WDT off, R type, V_{LCD}=V_{DD}, 1/3 bias (I_{BIAS}=7.5μA)	—	11	20	μA
		5V		—	18.3	40	μA
I_{STB8}	Standby Current (Idle) (f_{SYS}, f_{WDT} off, f_S=f_{SUB}=32768 or 32K RC OSC)	3V	No load, system HALT, LCD on, WDT off, R type, V_{LCD}=V_{DD}, 1/3 bias (I_{BIAS}=15μA)	—	16	25	μA
		5V		—	26.6	50	μA
I_{STB9}	Standby Current (Idle) (f_{SYS}, f_{WDT} off, f_S=f_{SUB}=32768 or 32K RC OSC)	3V	No load, system HALT, LCD on, WDT off, R type, V_{LCD}=V_{DD}, 1/3 bias (I_{BIAS}=45μA)	—	36	50	μA
		5V		—	60	100	μA
I_{STB10}	Standby Current (Idle) (f_{SYS}, f_{WDT} off, f_S=f_{SUB}=32768 or 32K RC OSC)	3V	No load, system HALT, LCD on, WDT off, R type, V_{LCD}=V_{DD}, 1/3 bias (I_{BIAS}=90μA)	—	66	100	μA
		5V		—	110	200	μA
I_{OL2}	LCD Common and Segment Sink Current	3V	V_{OL}=0.1V_{LCD}	210	420	—	μA
		5V		350	700	—	μA
I_{OH2}	LCD Common and Segment Source Current	3V	V_{OH}=0.9V_{LCD}	-80	-160	—	μA
		5V		-180	-360	—	μA

Power-on Reset Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{POR}	V_{DD} Start Voltage to Ensure Power-on Reset	—	—	—	—	100	mV
RR_{VDD}	V_{DD} Raising Rate to Ensure Power-on Reset	—	—	0.035	—	—	V/ms
t_{POR}	Minimum Time for V_{DD} Stays at V_{POR} to Ensure Power-on Reset	—	—	1	—	—	ms



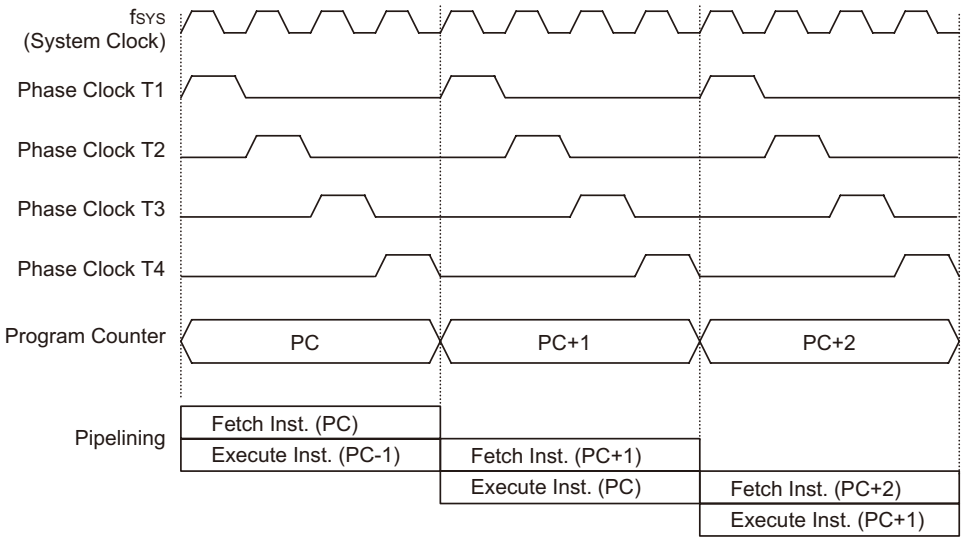
System Architecture

A key factor in the high-performance features of the Holtek range of microcontrollers is attributed to their internal system architecture. The range of devices take advantage of the usual features found within RISC microcontrollers providing increased speed of operation and enhanced performance. The pipelining scheme is implemented in such a way that instruction fetching and instruction execution are overlapped, hence instructions are effectively executed in one cycle, with the exception of branch or call instructions. An 8-bit wide ALU is used in practically all instruction set operations, which carries out arithmetic operations, logic operations, rotation, increment, decrement, branch decisions, etc. The internal data path is simplified by moving data through the Accumulator and the ALU. Certain internal registers are implemented in the Data Memory and can be directly or indirectly addressed. The simple addressing methods of these registers along with additional architectural features ensure that a minimum of external components is required to provide a functional I/O control system with maximum reliability and flexibility. This makes the device suitable for low-cost, high-volume production for controller applications.

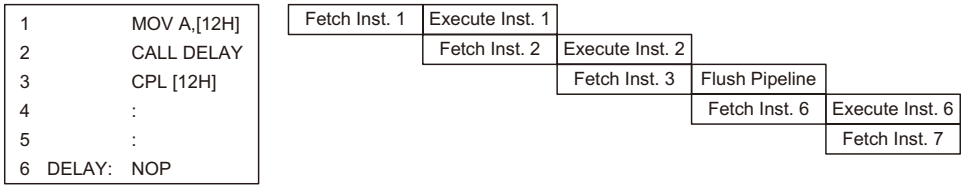
Clocking and Pipelining

The main system clock, derived from either a HXT, LXT, HIRC, LIRC, EC or ERC oscillator is subdivided into four internally generated non-overlapping clocks, T1~T4. The Program Counter is incremented at the beginning of the T1 clock during which time a new instruction is fetched. The remaining T2~T4 clocks carry out the decoding and execution functions. In this way, one T1~T4 clock cycle forms one instruction cycle. Although the fetching and execution of instructions takes place in consecutive instruction cycles, the pipelining structure of the microcontroller ensures that instructions are effectively executed in one instruction cycle. The exception to this are instructions where the contents of the Program Counter are changed, such as subroutine calls or jumps, in which case the instruction will take one more instruction cycle to execute.

For instructions involving branches, such as jump or call instructions, two machine cycles are required to complete instruction execution. An extra cycle is required as the program takes one cycle to first obtain the actual jump or call address and then another cycle to actually execute the branch. The requirement for this extra cycle should be taken into account by programmers in timing sensitive applications.



System Clocking and Pipelining



Instruction Fetching

Program Counter

During program execution, the Program Counter is used to keep track of the address of the next instruction to be executed. It is automatically incremented by one each time an instruction is executed except for instructions, such as "JMP" or "CALL" that demand a jump to a non-consecutive Program Memory address. Only the lower 8 bits, known as the Program Counter Low Register, are directly addressable by the application program.

When executing instructions requiring jumps to non-consecutive addresses such as a jump instruction, a subroutine call, interrupt or reset, etc., the microcontroller manages program control by loading the required address into the Program Counter. For conditional skip instructions, once the condition has been met, the next instruction, which has already been fetched during the present instruction execution, is discarded and a dummy cycle takes its place while the correct instruction is obtained.

Device	Program Counter	
	Porgram Counter High Byte	Porgram Counter Low Byte
HT69F30A	PC10~PC8	PCL7~PCL0
HT69F40A	PC11~PC8	
HT69F50A	PC12~PC8	

Program Counter

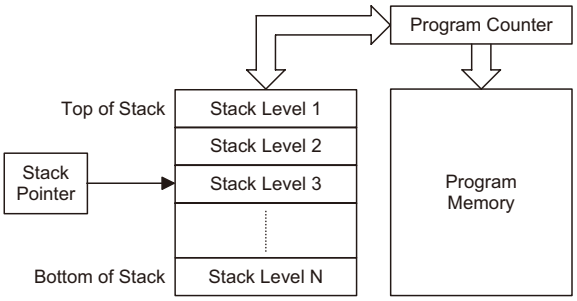
The lower byte of the Program Counter, known as the Program Counter Low register or PCL, is available for program control and is a readable and writeable register. By transferring data directly into this register, a short program jump can be executed directly, however, as only this low byte is available for manipulation, the jumps are limited to the present page of memory, that is 256 locations. When such program jumps are executed it should also be noted that a dummy cycle will be inserted. Manipulating the PCL register may cause program branching, so an extra cycle is needed to pre-fetch.

Stack

This is a special part of the memory which is used to save the contents of the Program Counter only. The stack has multiple levels depending upon the device and is neither part of the data nor part of the program space, and is neither readable nor writeable. The activated level is indexed by the Stack Pointer, and is neither readable nor writeable. At a subroutine call or interrupt acknowledge signal, the contents of the Program Counter are pushed onto the stack. At the end of a subroutine or an interrupt routine, signaled by a return instruction, RET or RETI, the Program Counter is restored to its previous value from the stack. After a device reset, the Stack Pointer will point to the top of the stack.

If the stack is full and an enabled interrupt takes place, the interrupt request flag will be recorded but the acknowledge signal will be inhibited. When the Stack Pointer is decremented, by RET or RETI, the interrupt will be serviced. This feature prevents stack overflow allowing the programmer to use the structure more easily. However, when the stack is full, a CALL subroutine instruction can still be executed which will result in a stack overflow. Precautions should be taken to avoid such cases which might cause unpredictable program branching.

If the stack is overflow, the first Program Counter save in the stack will be lost.



Device	Stack Levels
HT69F30A	4
HT69F40A	8
HT69F50A	8

Arithmetic and Logic Unit – ALU

The arithmetic-logic unit or ALU is a critical area of the microcontroller that carries out arithmetic and logic operations of the instruction set. Connected to the main microcontroller data bus, the ALU receives related instruction codes and performs the required arithmetic or logical operations after which the result will be placed in the specified register. As these ALU calculation or operations may result in carry, borrow or other status changes, the status register will be correspondingly updated to reflect these changes. The ALU supports the following functions:

- Arithmetic operations: ADD, ADDM, ADC, ADCM, SUB, SUBM, SBC, SBCM, DAA
- Logic operations: AND, OR, XOR, ANDM, ORM, XORM, CPL, CPLA
- Rotation RRA, RR, RRCA, RRC, RLA, RL, RLCA, RLC
- Increment and Decrement INCA, INC, DECA, DEC
- Branch decision, JMP, SZ, SZA, SNZ, SIZ, SDZ, SIZA, SDZA, CALL, RET, RETI

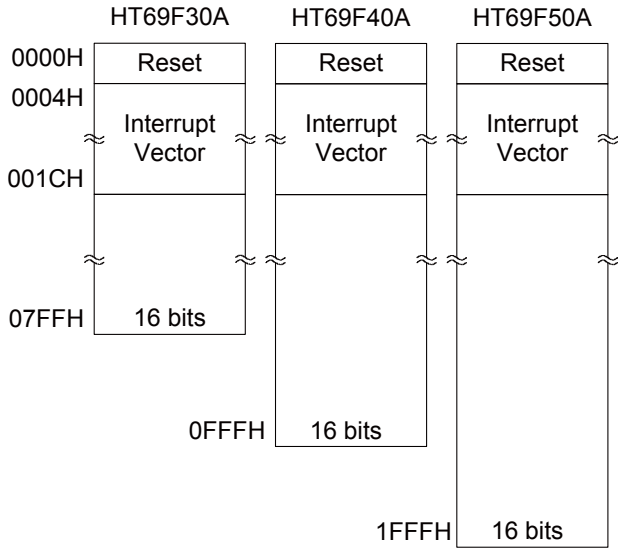
Flash Program Memory

The Program Memory is the location where the user code or program is stored. For this device series the Program Memory is Flash type, which means it can be programmed and re-programmed a large number of times, allowing the user the convenience of code modification on the same device. By using the appropriate programming tools, these Flash devices offer users the flexibility to conveniently debug and develop their applications while also offering a means of field programming and updating.

Structure

The Program Memory has a capacity of up to 8k×16 bits. The Program Memory is addressed by the Program Counter and also contains data, table information and interrupt entries information. Table data, which can be setup in any location within the Program Memory, is addressed by separate table pointer registers.

Device	Capacity
HT69F30A	2K×16
HT69F40A	4K×16
HT69F50A	8K×16



Program Memory Structure

Special Vectors

Within the Program Memory, certain locations are reserved for the reset and interrupts. The location 000H is reserved for use by the device reset for program initialisation. After a device reset is initiated, the program will jump to this location and begin execution.

Look-up Table

Any location within the Program Memory can be defined as a look-up table where programmers can store fixed data. To use the look-up table, the table pointer must first be setup by placing the address of the look up data to be retrieved in the table pointer register, TBLP and TBHP. These registers define the total address of the look-up table.

After setting up the table pointer, the table data can be retrieved from the Program Memory using the "TABRDC [m]" or "TABRDL [m]" instructions, respectively. When the instruction is executed, the lower order table byte from the Program Memory will be transferred to the user defined Data Memory register [m] as specified in the instruction. The higher order table data byte from the Program Memory will be transferred to the TBLH special register. Any unused bits in this transferred higher order byte will be read as 0.

The accompanying diagram illustrates the addressing data flow of the look-up table.

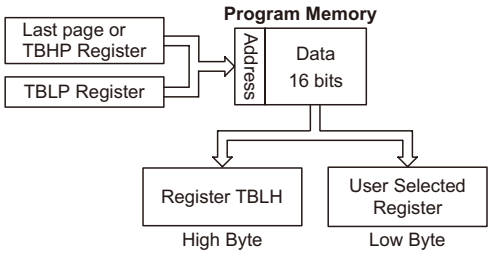


Table Program Example

The accompanying example shows how the table pointer and table data is defined and retrieved from the device. This example uses raw table data located in the last page which is stored there using the ORG statement. The value at this ORG statement is "700H" which refers to the start address of the last page within the 2K Program Memory of the HT69F30A device. The table pointer is setup here to have an initial value of "06H". This will ensure that the first data read from the data table will be at the Program Memory address "706H" or 6 locations after the start of the last page. Note that the value for the table pointer is referenced to the first address of the present page if the "TABRDC [m]" instruction is being used. The high byte of the table data which in this case is equal to zero will be transferred to the TBLH register automatically when the "TABRDC [m]" instruction is executed.

Because the TBLH register is a read-only register and cannot be restored, care should be taken to ensure its protection if both the main routine and Interrupt Service Routine use table read instructions. If using the table read instructions, the Interrupt Service Routines may change the value of the TBLH and subsequently cause errors if used again by the main routine. As a rule it is recommended that simultaneous use of the table read instructions should be avoided. However, in situations where simultaneous use cannot be avoided, the interrupts should be disabled prior to the execution of any main routine table-read instructions. Note that all table related instructions require two instruction cycles to complete their operation.

Table Read Program Example

```
tempreg1 db ?              ; temporary register #1
tempreg2 db ?              ; temporary register #2
        :
        :
mov  a,06h                 ; initialise table pointer - note that this address
                           ; is referenced to the last page or present page
mov  tblp, a
        :
        :
tabrdl   tempreg1          ; transfers value in table referenced by table pointer
                           ; to tempreg1
                           ; data at program memory address "706H" transferred to
                           ; to tempreg1 and TBLH
dec  tblp                  ; reduce value of table pointer by one
tabrdl   tempreg2          ; transfers value in table referenced by table pointer
                           ; to tempreg2
                           ; data at program memory address "705H" transferred to
                           ; tempreg2 and TBLH
                           ; in this example the data "1AH" is transferred to
                           ; tempreg1 and data "0FH" to register tempreg2
                           ; the value "00H" will be transferred to the high byte
                           ; register TBLH
        :
        :
org 700h                   ; sets initial address of last page
dc 00Ah, 00Bh, 00Ch, 00Dh, 00Eh, 00Fh, 01Ah, 01Bh
        :
        :
```

In Circuit Programming – ICP

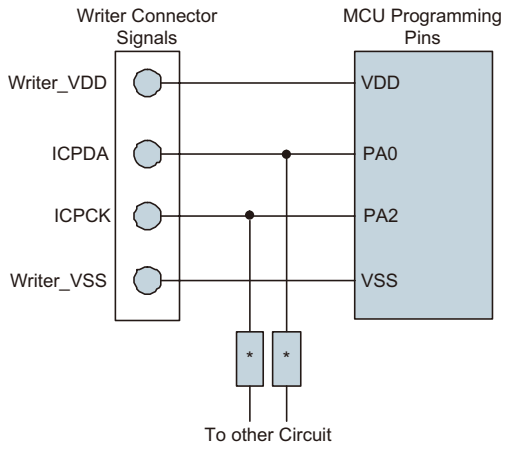
The provision of Flash type Program Memory provides the user with a means of convenient and easy upgrades and modifications to their programs on the same device.

As an additional convenience, Holtek has provided a means of programming the microcontroller in-circuit using a 4-pin interface. This provides manufacturers with the possibility of manufacturing their circuit boards complete with a programmed or un-programmed microcontroller, and then programming or upgrading the program at a later stage. This enables product manufacturers to easily keep their manufactured products supplied with the latest program releases without removal and re-insertion of the device.

Holtek Writer Pins	MCU Programming Pins	Pin Description
ICPDA	PA0	Programming Serial Data
ICPCK	PA2	Programming Clock
VDD	VDD	Power Supply
VSS	VSS	Ground

The Program Memory can be programmed serially in-circuit using this 4-wire interface. Data is downloaded and uploaded serially on a single pin with an additional line for the clock. Two additional lines are required for the power supply and one line for the reset. The technical details regarding the in-circuit programming of the devices are beyond the scope of this document and will be supplied in supplementary literature.

During the programming process, taking control of the ICPDA and ICPCK pins for data and clock programming purposes. The user must there take care to ensure that no other outputs are connected to these two pins.



Note: * may be resistor or capacitor. The resistance of * must be greater than 1k or the capacitance of * must be less than 1nF.

On-Chip Debug Support – OCDS

There is an EV chip named HT69V0A which is used to emulate the HT69Fx0A series of devices. The HT69V50A device also provides the "On-Chip Debug" function to debug the HT69Fx0A series of devices during development process. The devices, HT69Fx0A and HT69V50A, are almost functional compatible except the "On-Chip Debug" function and package types. Users can use the HT69V50A device to emulate the HT69Fx0A series of devices behaviors by connecting the OCDSDA and OCDSCK pins to the Holtek HT-IDE development tools. The OCDSDA pin is the OCDS Data/Address input/output pin while the OCDSCK pin is the OCDS clock input pin. When users use the HT69V50A EV chip for debugging, the corresponding pin functions shared with the OCDSDA and OCDSCK pins in the HT69Fx0A series of devices will have no effect in the HT69V50A EV chip. However, the two OCDS pins which are pin-shared with the ICP programming pins are still used as the Flash Memory programming pins for ICP. For more detailed OCDS information, refer to the corresponding document named "Holtek e-Link for 8-bit MCU OCDS User's Guide".

Holtek e-Link Pins	EV Chip Pins	Pin Description
OCDSDA	OCDSDA	On-Chip Debug Support Data/Address input/output
OCDSCK	OCDSCK	On-Chip Debug Support Clock input
VDD	VDD	Power Supply
GND	VSS	Ground

Data Memory

The Data Memory is an 8-bit wide RAM internal memory and is the location where temporary information is stored.

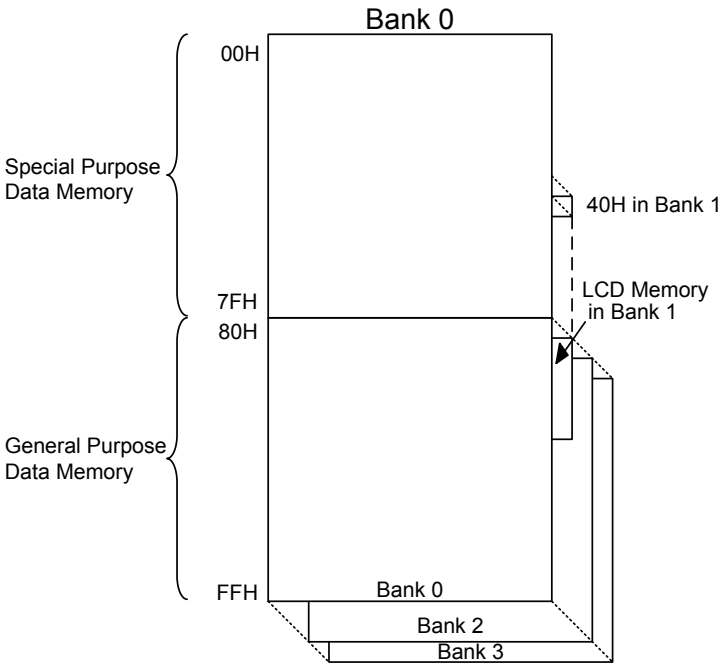
Divided into three sections, the first of these is an area of RAM where special function registers are located. These registers have fixed locations and are necessary for correct operation of the device. Many of these registers can be read from and written to directly under program control, however, some remain protected from user manipulation. The second area of Data Memory is reserved for general purpose use. All locations within this area are read and write accessible under program control. The third area is reserved for the LCD Memory. This special area of Data Memory is mapped directly to the LCD display so data written into this memory area will directly affect the displayed data. Switching between the different Data Memory banks is achieved by setting the Bank Pointer to the correct value.

Structure

The Data Memory is subdivided into several banks, all of which are implemented in 8-bit wide RAM. The Data Memory located in Bank 0 is subdivided into two sections, the Special Purpose Data Memory and the General Purpose Data Memory.

The start address of the Data Memory for all devices is the address 00H. Registers which are common to all microcontrollers, such as ACC, PCL, etc., have the same Data Memory address. The LCD Memory is mapped into Bank 1. The Banks 2 to 3 contain only General Purpose Data Memory for those devices with larger Data Memory capacities. As the Special Purpose Data Memory registers are mapped into all bank areas, they can subsequently be accessed from any bank location.

Device	Capacity	Banks
HT69F30A	General Purpose: 128×8 LCD Memory: 25	0: 80H~FFH 1: 80H~98H
HT69F40A	General Purpose: 256×8 LCD Memory: 37	0: 80H~FFH 1: 80H~A4H 2: 80H~FFH
HT69F50A	General Purpose: 384×8 LCD Memory: 49	0: 80H~FFH 1: 80H~B0H 2: 80H~FFH 3: 80H~FFH



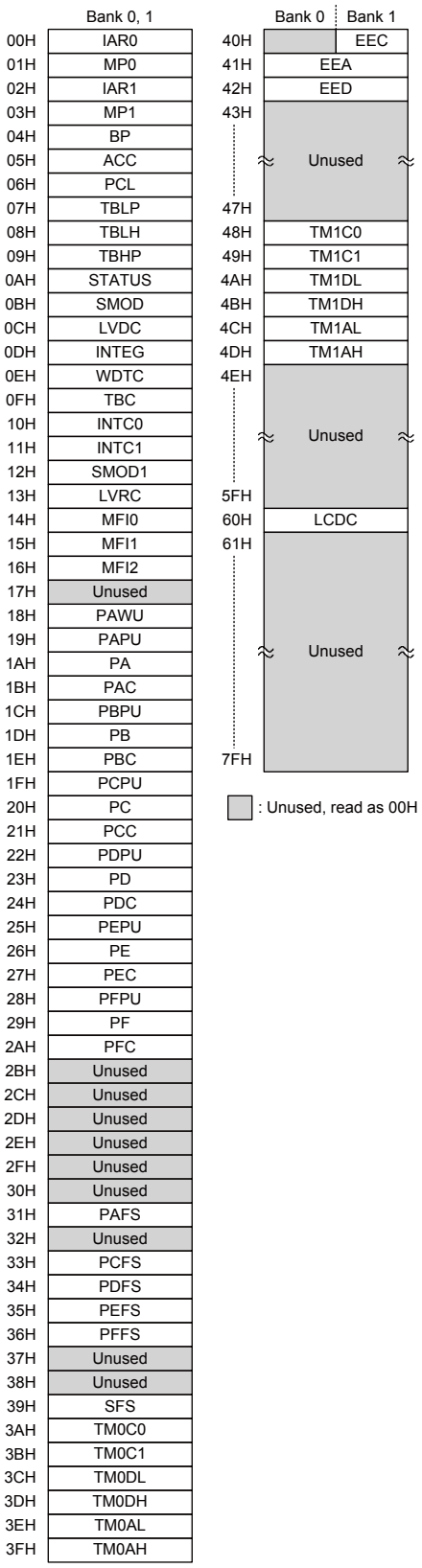
Data Memory Structure

General Purpose Data Memory

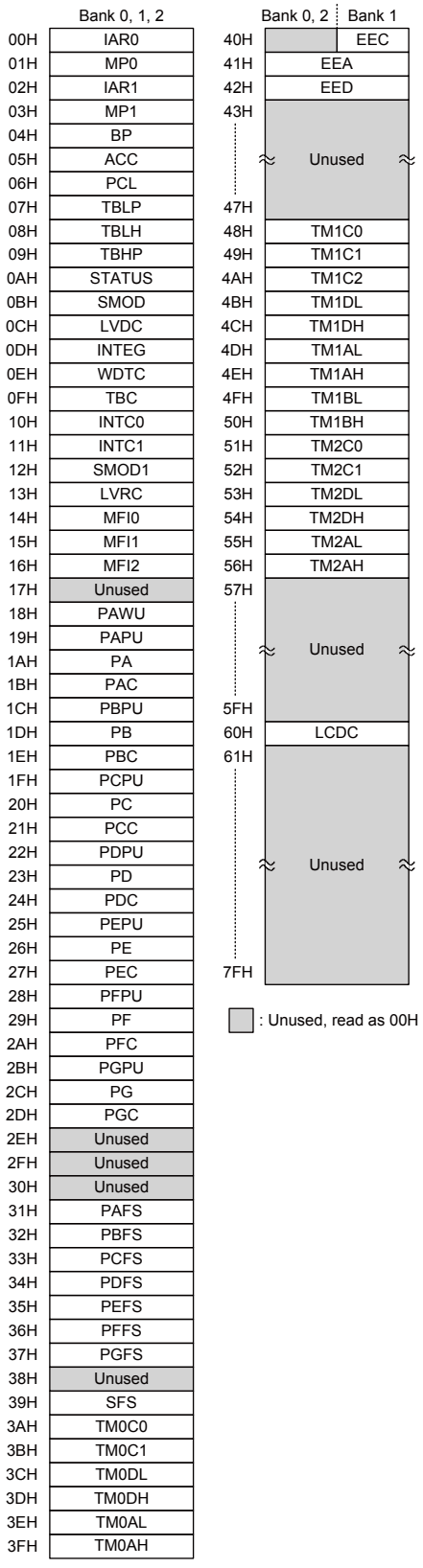
All microcontroller programs require an area of read/write memory where temporary data can be stored and retrieved for use later. It is this area of RAM memory that is known as General Purpose Data Memory. This area of Data Memory is fully accessible by the user programing for both reading and writing operations. By using the "SET [m].i" and "CLR [m].i" instructions individual bits can be set or reset under program control giving the user a large range of flexibility for bit manipulation in the Data Memory.

Special Purpose Data Memory

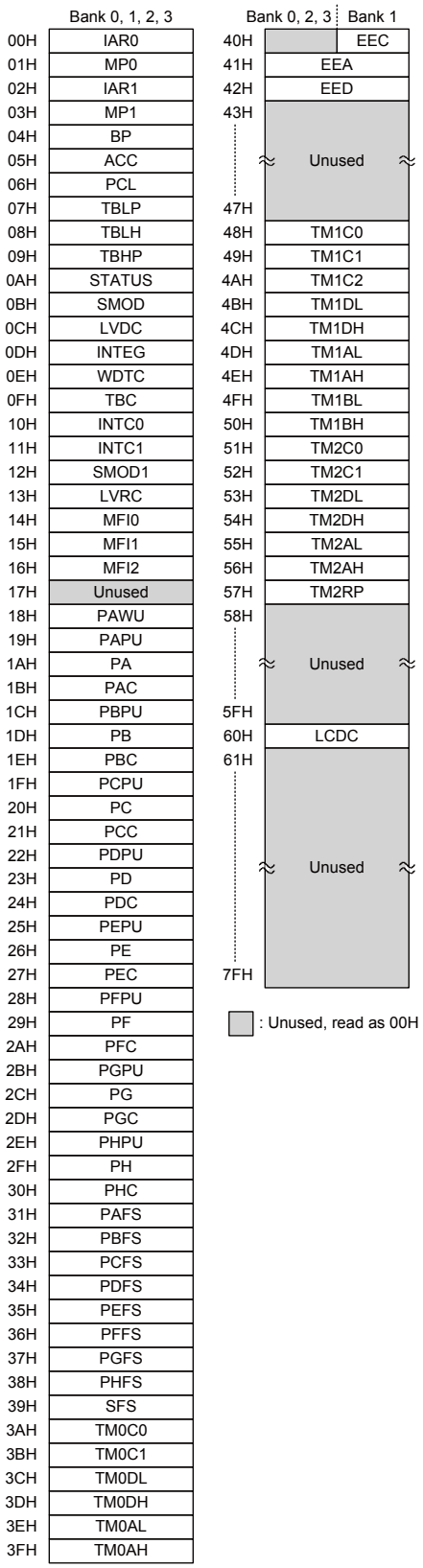
This area of Data Memory is where registers, necessary for the correct operation of the microcontroller, are stored. Most of the registers are both readable and writeable but some are protected and are readable only, the details of which are located under the relevant Special Function Register section. Note that for locations that are unused, any read instruction to these addresses will return the value "00H".



HT69F30A Special Purpose Data Memory



HT69F40A Special Purpose Data Memory



HT69F50A Special Purpose Data Memory

Special Function Register Description

Most of the Special Function Register details will be described in the relevant functional section; however several registers require a separate description in this section.

Indirect Addressing Registers – IAR0, IAR1

The Indirect Addressing Registers, IAR0 and IAR1, although having their locations in normal RAM register space, do not actually physically exist as normal registers. The method of indirect addressing for RAM data manipulation uses these Indirect Addressing Registers and Memory Pointers, in contrast to direct memory addressing, where the actual memory address is specified. Actions on the IAR0 and IAR1 registers will result in no actual read or write operation to these registers but rather to the memory location specified by their corresponding Memory Pointers, MP0 or MP1. Acting as a pair, IAR0 and MP0 can together access data from Bank 0 while the IAR1 and MP1 register pair can access data from any bank. As the Indirect Addressing Registers are not physically implemented, reading the Indirect Addressing Registers indirectly will return a result of "00H" and writing to the registers indirectly will result in no operation.

Memory Pointers – MP0, MP1

Two Memory Pointers, known as MP0 and MP1 are provided. These Memory Pointers are physically implemented in the Data Memory and can be manipulated in the same way as normal registers providing a convenient way with which to address and track data. When any operation to the relevant Indirect Addressing Registers is carried out, the actual address that the microcontroller is directed to, is the address specified by the related Memory Pointer. MP0, together with Indirect Addressing Register, IAR0, are used to access data from Bank 0, while MP1 and IAR1 are used to access data from all banks according to BP register. Direct Addressing can only be used with Bank 0, all other Banks must be addressed indirectly using MP1 and IAR1.

Indirect Addressing Program Example

```
data .section data
adres1   db ?
adres2   db ?
adres3   db ?
adres4   db ?
block    db ?
code .section at 0 code
org 00h

start:
     mov a,04h               ; setup size of block
     mov block,a
     mov a,offset adres1     ; Accumulator loaded with first RAM address
     mov mp0,a               ; setup memory pointer with first RAM address

loop:
     clr IAR0                ; clear the data at address defined by MP0
     inc mp0                 ; increment memory pointer
     sdz block               ; check if last memory location has been cleared
     jmp loop
     continue:
```

The important point to note here is that in the example shown above, no reference is made to specific RAM addresses.

Bank Pointer – BP

Depending upon which device is used, the Data Memory is divided into several banks. Selecting the required Data Memory area is achieved using the Bank Pointer. Bits 0~1 of the Bank Pointer are used to select Data Memory Banks 0~3.

The Data Memory is initialised to Bank 0 after a reset, except for a WDT time-out reset in the Power Down Mode, in which case, the Data Memory bank remains unaffected. It should be noted that the Special Function Data Memory is not affected by the bank selection, which means that the Special Function Registers can be accessed from within any bank. Directly addressing the Data Memory will always result in Bank 0 being accessed irrespective of the value of the Bank Pointer. Accessing data from banks other than Bank 0 must be implemented using Indirect addressing.

As both the Program Memory and Data Memory share the same Bank Pointer Register, care must be taken during programming.

Device	Bit							
	7	6	5	4	3	2	1	0
HT69F30A	—	—	—	—	—	—	—	DMBP0
HT69F40A	—	—	—	—	—	—	DMBP1	DMBP0
HT69F50A	—	—	—	—	—	—	DMBP1	DMBP0

BP Register List

BP Register

- **HT69F30A**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	—	DMBP0
R/W	—	—	—	—	—	—	—	R/W
POR	—	—	—	—	—	—	—	0

Bit 7~1 Unimplemented, read as "0"

Bit 0 **DMBP0:** Data memory bank point
0: Bank 0
1: Bank 1

- **HT69F40A**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	DMBP1	DMBP0
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 Unimplemented, read as "0"

Bit 1~0 **DMBP1, DMBP0:** Data memory bank point
00: Bank 0
01: Bank 1
10: Bank 2
11: Undefined

• **HT69F50A**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	DMBP1	DMBP0
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 Unimplemented, read as "0"

Bit 1~0 **DMBP1, DMBP0:** Data memory bank point
00: Bank 0
01: Bank 1
10: Bank 2
11: Bank 3

Accumulator – ACC

The Accumulator is central to the operation of any microcontroller and is closely related with operations carried out by the ALU. The Accumulator is the place where all intermediate results from the ALU are stored. Without the Accumulator it would be necessary to write the result of each calculation or logical operation such as addition, subtraction, shift, etc., to the Data Memory resulting in higher programming and timing overheads. Data transfer operations usually involve the temporary storage function of the Accumulator; for example, when transferring data between one user defined register and another, it is necessary to do this by passing the data through the Accumulator as no direct transfer between two registers is permitted.

Program Counter Low Register – PCL

To provide additional program control functions, the low byte of the Program Counter is made accessible to programmers by locating it within the Special Purpose area of the Data Memory. By manipulating this register, direct jumps to other program locations are easily implemented. Loading a value directly into this PCL register will cause a jump to the specified Program Memory location, however, as the register is only 8-bit wide, only jumps within the current Program Memory page are permitted. When such operations are used, note that a dummy cycle will be inserted.

Look-up Table Registers – TBLP, TBHP, TBLH

These three special function registers are used to control operation of the look-up table which is stored in the Program Memory. TBLP and TBHP are the table pointer and indicates the location where the table data is located. Their value must be setup before any table read commands are executed. Their value can be changed, for example using the “INC” or “DEC” instructions, allowing for easy table data pointing and reading. TBLH is the location where the high order byte of the table data is stored after a table read data instruction has been executed. Note that the lower order table data byte is transferred to a user defined location.

Status Register – STATUS

This 8-bit register contains the zero flag (Z), carry flag (C), auxiliary carry flag (AC), overflow flag (OV), power down flag (PDF), and watchdog time-out flag (TO). These arithmetic/logical operation and system management flags are used to record the status and operation of the microcontroller.

With the exception of the TO and PDF flags, bits in the status register can be altered by instructions like most other registers. Any data written into the status register will not change the TO or PDF flag. In addition, operations related to the status register may give different results due to the different instruction operations. The TO flag can be affected only by a system power-up, a WDT time-out or by executing the "CLR WDT" or "HALT" instruction. The PDF flag is affected only by executing the "HALT" or "CLR WDT" instruction or during a system power-up.

The Z, OV, AC and C flags generally reflect the status of the latest operations.

- C is set if an operation results in a carry during an addition operation or if a borrow does not take place during a subtraction operation; otherwise C is cleared. C is also affected by a rotate through carry instruction.
- AC is set if an operation results in a carry out of the low nibbles in addition, or no borrow from the high nibble into the low nibble in subtraction; otherwise AC is cleared.
- Z is set if the result of an arithmetic or logical operation is zero; otherwise Z is cleared.
- OV is set if an operation results in a carry into the highest-order bit but not a carry out of the highest-order bit, or vice versa; otherwise OV is cleared.
- PDF is cleared by a system power-up or executing the "CLR WDT" instruction. PDF is set by executing the "HALT" instruction.
- TO is cleared by a system power-up or executing the "CLR WDT" or "HALT" instruction. TO is set by a WDT time-out.

In addition, on entering an interrupt sequence or executing a subroutine call, the status register will not be pushed onto the stack automatically. If the contents of the status registers are important and if the subroutine can corrupt the status register, precautions must be taken to correctly save it.

STATUS Register

Bit	7	6	5	4	3	2	1	0
Name	—	—	TO	PDF	OV	Z	AC	C
R/W	—	—	R	R	R/W	R/W	R/W	R/W
POR	—	—	0	0	x	x	x	x

"x" unknown

Bit 7, 6 Unimplemented, read as "0"

Bit 5 **TO:** Watchdog Time-Out flag
0: After power up or executing the "CLR WDT" or "HALT" instruction
1: A watchdog time-out occurred

Bit 4 **PDF:** Power down flag
0: After power up or executing the "CLR WDT" instruction
1: By executing the "HALT" instruction

Bit 3 **OV:** Overflow flag
0: No overflow
1: An operation results in a carry into the highest-order bit but not a carry out of the highest-order bit or vice versa

Bit 2 **Z:** Zero flag
0: The result of an arithmetic or logical operation is not zero
1: The result of an arithmetic or logical operation is zero

Bit 1 **AC:** Auxiliary flag
0: No auxiliary carry
1: An operation results in a carry out of the low nibbles in addition, or no borrow from the high nibble into the low nibble in subtraction

Bit 0 **C:** Carry flag
0: No carry-out
1: An operation results in a carry during an addition operation or if a borrow does not take place during a subtraction operation

C is also affected by a rotate through carry instruction.

EEPROM Data Memory

The EEPROM Data Memory capacity is up to 128×8 bits for this series of devices. Unlike the Program Memory and RAM Data Memory, the EEPROM Data Memory is not directly mapped into memory space and is therefore not directly addressable in the same way as the other types of memory. Read andWrite operations to the EEPROM are carried out in single byte operations using an address and data register in Bank 0 and a single control register in Bank 1.

Device	Capacity	Address
HT69F30A	64×8	00H~3FH
HT69F40A	128×8	00H~7FH
HT69F50A	128×8	00H~7FH

EEPROM Registers

Three registers control the overall operation of the internal EEPROM Data Memory. These are the address register, EEA, the data register, EED and a single control register, EEC. As both the EEA and EED registers are located in Bank 0, they can be directly accessed in the same was as any other Special Function Register. The EEC register however, being located in Bank1, cannot be addressed directly and can only be read from or written to indirectly using the MP1 Memory Pointer and Indirect Addressing Register, IAR1. Because the EEC control register is located at address 40H in Bank 1, the MP1 Memory Pointer must first be set to the value 40H and the Bank Pointer register, BP, set to the value, 01H, before any operations on the EEC register are executed.

EEPROM Register List

- **HT69F30A**

Register Name	Bit							
	7	6	5	4	3	2	1	0
EEA	—	—	EEA5	EEA4	EEA3	EEA2	EEA1	EEA0
EED	EED7	EED6	EED5	EED4	EED3	EED2	EED1	EED0
EEC	—	—	—	—	WREN	WR	RDEN	RD

- **HT69F40A/HT69F50A**

Register Name	Bit							
	7	6	5	4	3	2	1	0
EEA	—	EEA6	EEA5	EEA4	EEA3	EEA2	EEA1	EEA0
EED	EED7	EED6	EED5	EED4	EED3	EED2	EED1	EED0
EEC	—	—	—	—	WREN	WR	RDEN	RD

EEA Register

- **HT69F30A**

Bit	7	6	5	4	3	2	1	0
Name	—	—	EEA5	EEA4	EEA3	EEA2	EEA1	EEA0
R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	—	x	x	x	x	x	x

"x": unknown

Bit 7~6 Unimplemented, read as "0"

Bit 5~0 **EEA5~EEA0:** Data EEPROM address bit 5~bit 0

• **HT69F40A/HT69F50A**

Bit	7	6	5	4	3	2	1	0
Name	—	EEA6	EEA5	EEA4	EEA3	EEA2	EEA1	EEA0
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	x	x	x	x	x	x	x

"x": unknown

Bit 7 Unimplemented, read as "0"

Bit 6~0 **EEA6~EEA0:** Data EEPROM address bit 6~bit 0

EED Register

Bit	7	6	5	4	3	2	1	0
Name	EED5	EED4	EED5	EED4	EED3	EED2	EED1	EED0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	x	x	x	x	x	x

"x": unknown

Bit 7~0 **EED7~EED0:** Data EEPROM data bit 7~bit 0

EEC Register

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	WREN	WR	RDEN	RD
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

Bit 7~4 Unimplemented, read as "0"

Bit 3 **WREN:** Data EEPROM write operation enable
0: Disable
1: Enable
This is the Data EEPROM Write Operation Enable bit which must be set high before Datat EEPROM write operations are carried out. Clearing this bit to zero will inhibit Data EEPROM write operations.

Bit 2 **WR:** Data EEPROM write control
0: Write cycle has finished
1: Activate a write cycle
This is the Data EEPROM Write Control bit and when set high by the application program will activate a write cycle. This bit will be automatically reset to zero by the hardware after the write cycle has finished. Setting this bit high will have no effect if the WREN bit has not first been set high.

Bit 1 **RDEN:** Data EEPROM read operation enable
0: Disable
1: Enable
This is the Data EEPROM Read Operation Enable bit which must be set high before Datat EEPROM read operations are carried out. Clearing this bit to zero will inhibit Data EEPROM read operations.

Bit 0 **RD:** Data EEPROM read control
0: Read cycle has finished
1: Activate a read cycle
This is the Data EEPROM Read Control bit and when set high by the application program will activate a read cycle. This bit will be automatically reset to zero by the hardware after the read cycle has finished. Setting this bit high will have no effect if the RDEN bit has not first been set high.

Note: The WREN, WR, RDEN and RD bits can not be set to "1" at the same time in one instruction.
The WR and RD bits can not be set to "1" at the same time.

Reading Data from the EEPROM

To read data from the EEPROM, the read enable bit, RDEN, in the EEC register must first be set high to enable the read function. The EEPROM address of the data to be read must then be placed in the EEAregister. If the RD bit in the EEC register is now set high, a read cycle will be initiated. Setting the RD bit high will not initiate a read operation if the RDEN bit has not been set. When the read cycle terminates, the RD bit will be automatically cleared to zero, after which the data can be read from the EED register. The data will remain in the EED register until another read or write operation is executed. The application program can poll the RD bit to determine when the data is valid for reading.

Writing Data to the EEPROM

The EEPROM address of the data to be written must first be placed in the EEA register and the data placed in the EED register. To write data to the EEPROM, the write enable bit, WREN, in the EEC register must first be set high to enable the write function. After this, the WR bit in the EEC register must be immediately set high to initiate a write cycle. These two instructions must be executed consecutively. The global interrupt bit EMI should also first be cleared before implementing any write operations, and then set again after the write cycle has started. Note that setting the WR bit high will not initiate a write cycle if the WREN bit has not been set. As the EEPROM write cycle is controlled using an internal timer whose operation is asynchronous to microcontroller system clock, a certain time will elapse before the data will have been written into the EEPROM. Detecting when the write cycle has finished can be implemented either by polling the WR bit in the EEC register or by using the EEPROM interrupt. When the write cycle terminates, the WR bit will be automatically cleared to zero by the microcontroller, informing the user that the data has been written to the EEPROM. The application program can therefore poll the WR bit to determine when the write cycle has ended.

Write Protection

Protection against inadvertent write operation is provided in several ways. After the device is powered-on the Write Enable bit in the control register will be cleared preventing any write operations. Also at power-on the Bank Pointer, BP, will be reset to zero, which means that Data Memory Bank 0 will be selected. As the EEPROM control register is located in Bank 1, this adds a further measure of protection against spurious write operations. During normal program operation, ensuring that the Write Enable bit in the control register is cleared will safeguard against incorrect write operations.

EEPROM Interrupt

The EEPROM write interrupt is generated when an EEPROM write cycle has ended. The EEPROM interrupt must first be enabled by setting the DEE bit in the relevant interrupt register. However as the EEPROM is contained within a Multi-function Interrupt, the associated multi-function interrupt enable bit must also be set. When an EEPROM write cycle ends, the DEF request flag and its associated multi-function interrupt request flag will both be set. If the global, EEPROM and Multi-function interrupts are enabled and the stack is not full, a jump to the associated Multi-function Interrupt vector will take place. When the interrupt is serviced only the Multi-function interrupt flag will be automatically reset, the EEPROM interrupt flag must be manually reset by the application program. More details can be obtained in the Interrupt section.

Programming Considerations

Care must be taken that data is not inadvertently written to the EEPROM. Protection can be enhanced by ensuring that the Write Enable bit is normally cleared to zero when not writing. Also the Bank Pointer could be normally cleared to zero as this would inhibit access to Bank 1 where the EEPROM control register exist. Although certainly not necessary, consideration might be given in the application program to the checking of the validity of new write data by a simple read back process.

When writing data the WR bit must be set high immediately after the WREN bit has been set high, to ensure the write cycle executes correctly. The global interrupt bit EMI should also be cleared before a write cycle is executed and then re-enabled after the write cycle starts.

Programming Examples

Reading Data from the EEPROM – Polling Mothod

```
MOV  A, EEPROM_ADRES          ; user defined address
MOV  EEA, A
MOV  A, 040H                  ; setup memory pointer MP1
MOV  MP1, A                   ; MP1 points to EEC register
MOV  A, 01H                   ; setup Bank Pointer
MOV  BP, A
SET  IAR1.1                   ; set RDEN bit, enable read operations
SET  IAR1.0                   ; start Read Cycle - set RD bit
BACK:
SZ   IAR1.0                   ; check for read cycle end
JMP  BACK
CLR  IAR1                     ; disable EEPROM read/write
CLR  BP
MOV  A, EED                   ; move read data to register
MOV  READ_DATA, A
```

Writing Data to the EEPROM – Polling Mothod

```
CLR  EMI
MOV  A, EEPROM_ADRES          ; user defined address
MOV  EEA, A
MOV  A, EEPROM_DATA           ; user defined data
MOV  EED, A
MOV  A, 040H                  ; setup memory pointer MP1
MOV  MP1, A                   ; MP1 points to EEC register
MOV  A, 01H                   ; setup Bank Pointer
MOV  BP, A
SET  IAR1.3                   ; set WREN bit, enable write operations
SET  IAR1.2                   ; Start Write Cycle - set WR bit - executed immediately
                              ; after set WREN bit
SET  EMI
BACK:
SZ   IAR1.2                   ; check for write cycle end
JMP  BACK
CLR  IAR1                     ; disable EEPROM read/write
CLR  BP
```

Oscillator

Various oscillator options offer the user a wide range of functions according to their various application requirements. The flexible features of the oscillator functions ensure that the best optimisation can be achieved in terms of speed and power saving. Oscillator selections and operation are selected through a combination of configuration options and registers.

Oscillator Overview

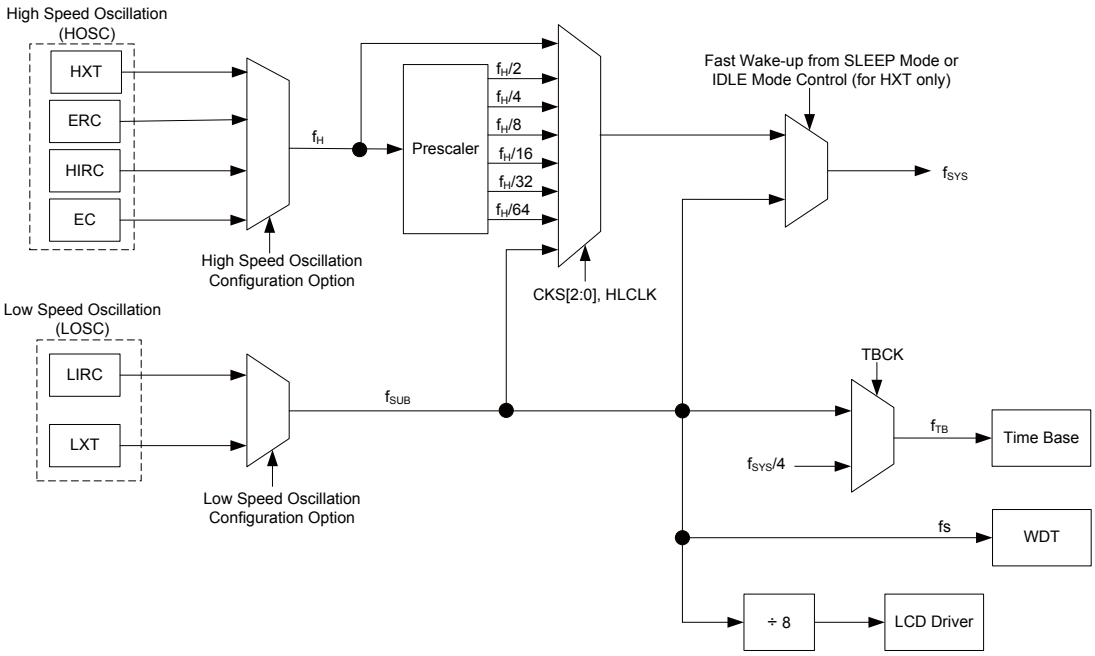
In addition to being the source of the main system clock the oscillators also provide clock sources for the Watchdog Timer and Time Base Interrupts. External oscillators requiring some external components as well as fully integrated internal oscillators, requiring no external components, are provided to form a wide range of both fast and slow system oscillators. All oscillator options are selected through the configuration options. The higher frequency oscillators provide higher performance but carry with it the disadvantage of higher power requirements, while the opposite is of course true for the lower frequency oscillators. With the capability of dynamically switching between fast and slow system clock, the device has the flexibility to optimize the performance/power ratio, a feature especially important in power sensitive portable applications.

Type	Name	Freq.	Pins
External Crystal	HXT	400kHz~16MHz	OSC1/OSC2
External RC	ERC	8MHz	OSC1
External Clock	EC	400kHz~20MHz	OSC1
Internal High Speed RC	HIRC	4, 8 or 12MHz	—
External Low Speed Crystal	LXT	32.768kHz	XT1/XT2
Internal Low Speed RC	LIRC	32kHz	—

Oscillator Types

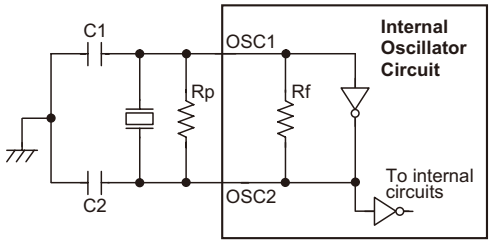
System Clock Configurations

There are five methods of generating the system clock, three high speed oscillators and two low speed oscillators. The high speed oscillators are is the external crystal/ceramic oscillator, external RC network oscillator and the internal 4MHz, 8MHz or 12MHz RC oscillator. The two low speed oscillators are the internal 32kHz RC oscillator and the external 32.768kHz crystal oscillator. Selecting whether the low or high speed oscillator is used as the system oscillator is implemented using the HLCLK bit and CKS2~CKS0 bits in the SMOD register and as the system clock can be dynamically selected.



External Crystal/Ceramic Oscillator – HXT

The External Crystal/Ceramic System Oscillator is one of the high frequency oscillator choices, which is selected via configuration option. For most crystal oscillator configurations, the simple connection of a crystal across OSC1 and OSC2 will create the necessary phase shift and feedback for oscillation, without requiring external capacitors. However, for some crystal types and frequencies, to ensure oscillation, it may be necessary to add two small value capacitors, C1 and C2. Using a ceramic resonator will usually require two small value capacitors, C1 and C2, to be connected as shown for oscillation to occur. The values of C1 and C2 should be selected in consultation with the crystal or resonator manufacturer's specification. An additional configuration option must be setup to configure the device according to whether the oscillator frequency is high, defined as equal to or above 1MHz, or low, which is defined as below 1MHz.



Note: 1. Rp is normally not required. C1 and C2 are required.
2. Although not shown OSC1/OSC2 pins have a parasitic capacitance of around 7pF.

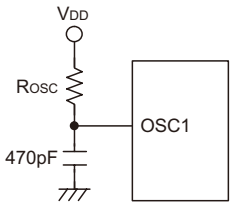
Crystal/Resonator Oscillator – HXT

Crystal Oscillator C1 and C2 Values		
Crystal Frequency	C1	C2
12MHz	0pF	0pF
8MHz	0pF	0pF
4MHz	0pF	0pF
1MHz	100pF	100pF
455kHz (see Note2)	100pF	100pF
Note: 1. C1 and C2 values are for guidance only. 2. XTAL mode configuration option: 455kHz.		

Crystal Recommended Capacitor Values

External RC Oscillator – ERC

Using the ERC oscillator only requires that a resistor, with a value between 56kΩ and 2.4MΩ, is connected between OSC1 and VDD, and a capacitor is connected between OSC1 and ground, providing a low cost oscillator configuration. It is only the external resistor that determines the oscillation frequency; the external capacitor has no influence over the frequency and is connected for stability purposes only. Device trimming during the manufacturing process and the inclusion of internal frequency compensation circuits are used to ensure that the influence of the power supply voltage, temperature and process variations on the oscillation frequency are minimised. As a resistance/frequency reference point, it can be noted that with an external 150kΩ resistor connected and with a 5V voltage power supply and temperature of 25°C degrees, the oscillator will have a frequency of 4MHz within a tolerance of 2%. Here only the OSC1 pin is used, which is shared with I/O pin PB0, leaving pin PB1 free for use as a normal I/O pin.



External RC Oscillator — ERC

Internal RC Oscillator – HIRC

The internal RC oscillator is a fully integrated system oscillator requiring no external components. The internal RC oscillator has three fixed frequencies of either 4MHz, 8MHz or 12MHz. Device trimming during the manufacturing process and the inclusion of internal frequency compensation circuits are used to ensure that the influence of the power supply voltage, temperature and process variations on the oscillation frequency are minimised. As a result, at a power supply of either 3V or 5V and at a temperature of 25°C degrees, the fixed oscillation frequency of 4MHz, 8MHz or 12MHz will have a tolerance within 2%. Note that if this internal system clock option is selected, as it requires no external pins for its operation, I/O pins PB0 and PB1 are free for use as normal I/O pins.

External 32.768kHz Crystal Oscillator – LXT

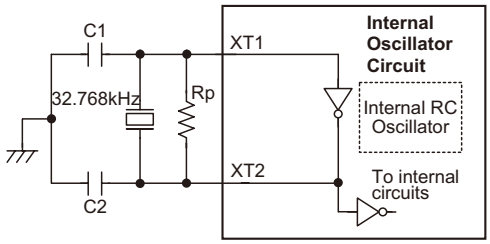
The External 32.768kHz Crystal System Oscillator is one of the low frequency oscillator choices, which is selected via configuration option. This clock source has a fixed frequency of 32.768kHz and requires a 32.768kHz crystal to be connected between pins XT1 and XT2. The external resistor and capacitor components connected to the 32.768kHz crystal are necessary to provide oscillation. For applications where precise frequencies are essential, these components may be required to provide frequency compensation due to different crystal manufacturing tolerances. During power-up there is a time delay associated with the LXT oscillator waiting for it to start-up.

When the microcontroller enters the SLEEP or IDLE Mode, the system clock is switched off to stop microcontroller activity and to conserve power. However, in many microcontroller applications it may be necessary to keep the internal timers operational even when the microcontroller is in the SLEEP or IDLE Mode. To do this, another clock, independent of the system clock, must be provided.

However, for some crystals, to ensure oscillation and accurate frequency generation, it is necessary to add two small value external capacitors, C1 and C2. The exact values of C1 and C2 should be selected in consultation with the crystal or resonator manufacturer's specification. The external parallel feedback resistor, Rp, is required.

Some configuration options determine if the XT1/XT2 pins are used for the LXT oscillator or as I/O pins.

- If the LXT oscillator is not used for any clock source, the XT1/XT2 pins can be used as normal I/O pins.
- If the LXT oscillator is used for any clock source, the 32.768kHz crystal should be connected to the XT1/XT2 pins.



Note: 1. Rp, C1 and C2 are required.
2. Although not shown pins have a parasitic capacitance of around 7pF.

External LXT Oscillator

LXT Oscillator C1 and C2 Values		
Crystal Frequency	**C1**	**C2**
32.768kHz	10pF	10pF
Note:1. C1 and C2 values are for guidance only. 2. R_P=5M~10MΩ is recommended.		

32.768kHz Crystal Recommended Capacitor Values

LXT Oscillator Low Power Function

The LXT oscillator can function in one of two modes, the Quick Start Mode and the Low Power Mode. The mode selection is executed using the LXTLP bit in the TBC register.

LXTLP Bit	LXT Mode
0	Quick Start
1	Low-power

After power on the LXTLP bit will be automatically cleared to zero ensuring that the LXT oscillator is in the Quick Start operating mode. In the Quick Start Mode the LXT oscillator will power up and stabilise quickly. However, after the LXT oscillator has fully powered up it can be placed into the Low-power mode by setting the LXTLP bit high. The oscillator will continue to run but with reduced current consumption, as the higher current consumption is only required during the LXT oscillator start-up. In power sensitive applications, such as battery applications, where power consumption must be kept to a minimum, it is therefore recommended that the application program sets the LXTLP bit high about 2 seconds after power-on.

It should be noted that, no matter what condition the LXTLP bit is set to, the LXT oscillator will always function normally, the only difference is that it will take more time to start up if in the Low-power mode.

Internal 32kHz Oscillator – LIRC

The Internal 32kHz System Oscillator is one of the low frequency oscillator choices, which is selected via configuration option. It is a fully integrated RC oscillator with a typical frequency of 32kHz at 5V, requiring no external components for its implementation. Device trimming during the manufacturing process and the inclusion of internal frequency compensation circuits are used to ensure that the influence of the power supply voltage, temperature and process variations on the oscillation frequency are minimised. As a result, at a power supply of 5V and at a temperature of 25°C degrees, the fixed oscillation frequency of 32kHz will have a tolerance within 10%.

External Clock – EC

The system clock can also be supplied by an externally supplied clock giving users a method of synchronizing their external hardware to the microcontroller operation. This is selected using a configuration option and supplying the clock on pin OSC1. Pin OSC2 should be left floating if the external oscillator is used. The internal oscillator circuit contains a filter circuit to reduce the possibility of erratic operation due to noise on the oscillator pin, however as the filter circuit consumes a certain amount of power, a oscillator configuration option exists to turn this filter off. Not using the internal filter should be considered in power sensitive applications and where the externally supplied clock is of a high integrity and supplied by a low impedance source.

Supplementary Oscillators

The low speed oscillators, in addition to providing a system clock source are also used to provide a clock source to two other device functions. These are the Watchdog Timer, the LCD driver and the Time Base Interrupts.

Operating Modes and System Clocks

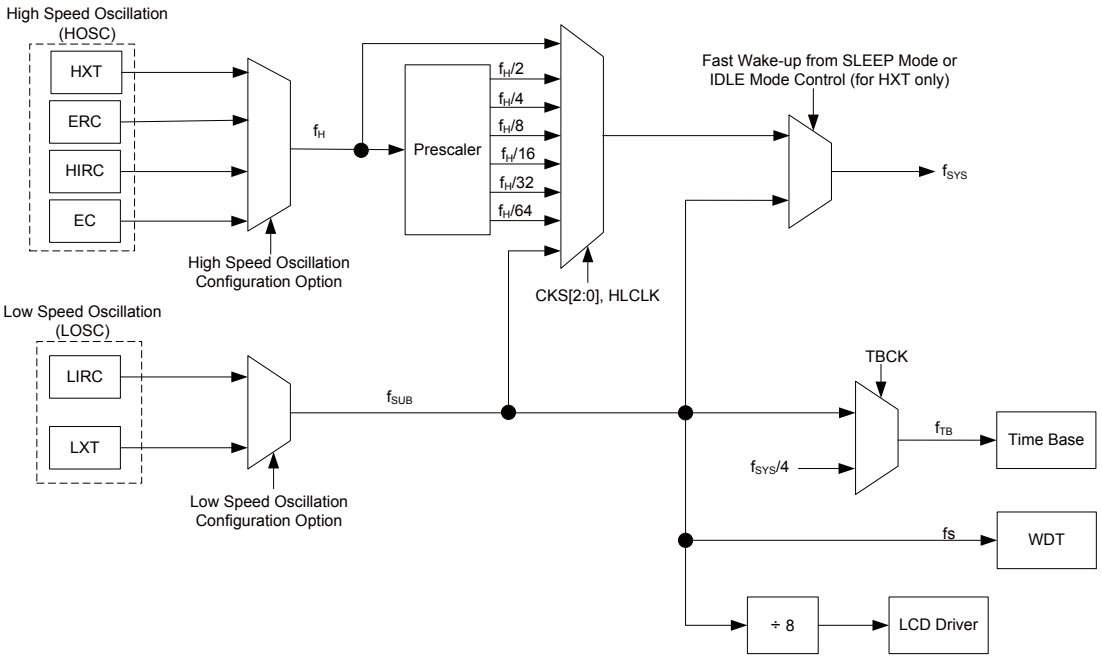
Present day applications require that their microcontrollers have high performance but often still demand that they consume as little power as possible, conflicting requirements that are especially true in battery powered portable applications. The fast clocks required for high performance will by their nature increase current consumption and of course vice-versa, lower speed clocks reduce current consumption. As Holtek has provided these devices with both high and low speed clock sources and the means to switch between them dynamically, the user can optimise the operation of their microcontroller to achieve the best performance/power ratio.

System Clock

The device has many different clock sources for both the CPU and peripheral function operation. By providing the user with a wide range of clock options using configuration options and register programming, a clock system can be configured to obtain maximum application performance.

The main system clock, can come from either a high frequency, f_H, or low frequency, f_{SUB}, source, and is selected using the HLCLK bit and CKS2~CKS0 bits in the SMOD register. The high speed system clock can be sourced from an HXT, ERC or HIRC oscillator, selected via a configuration option. The low speed system clock source can be sourced from the clock, f_{SUB}. If f_{SUB} is selected then it can be sourced by either the LXT or LIRC oscillators, selected via a configuration option. The other choice, which is a divided version of the high speed system oscillator has a range of $f_H/2$~$f_H/64$.

The f_{SUB} clock is used to provide a substitute clock for the microcontroller just after a wake-up has occurred to enable faster wake-up times.



System Clock Configurations

Note: When the system clock source f_{SYS} is switched to f_{SUB} from f_H, the high speed oscillation will stop to conserve the power. Thus there is no f_H~$f_H/64$ for peripheral circuit to use.

System Operation Modes

There are six different modes of operation for the microcontroller, each one with its own special characteristics and which can be chosen according to the specific performance and power requirements of the application. There are two modes allowing normal operation of the microcontroller, the NORMAL Mode and SLOW Mode. The remaining four modes, the SLEEP0, SLEEP1, IDLE0 and IDLE1 Mode are used when the microcontroller CPU is switched off to conserve power.

Operating Mode	Description			
	CPU	f_{SYS}	f_{SUB}	f_S
NORMAL Mode	On	f_H~f_H/64	On	On
SLOW Mode	On	f_{SUB}	On	On
IDLE0 Mode	Off	Off	On	On
IDLE1 Mode	Off	On	On	On
SLEEP0 Mode	Off	Off	Off	Off
SLEEP1 Mode	Off	Off	On	On

NORMAL Mode

As the name suggests this is one of the main operating modes where the microcontroller has all of its functions operational and where the system clock is provided by one of the high speed oscillators. This mode operates allowing the microcontroller to operate normally with a clock source will come from one of the high speed oscillators, either the HXT, ERC, EC or HIRC oscillators. The high speed oscillator will however first be divided by a ratio ranging from 1 to 64, the actual ratio being selected by the CKS2~CKS0 and HLCLK bits in the SMOD register. Although a high speed oscillator is used, running the microcontroller at a divided clock ratio reduces the operating current.

SLOW Mode

This is also a mode where the microcontroller operates normally although now with a slower speed clock source. The clock source used will be from one of the low speed oscillators, either the LXT or the LIRC. Running the microcontroller in this mode allows it to run with much lower operating currents. In the SLOW Mode, the f_H is off.

SLEEP0 Mode

The SLEEP Mode is entered when an HALT instruction is executed and when the IDLEN bit in the SMOD register is low. In the SLEEP0 mode the CPU will be stopped, and the f_{SUB} and f_S clocks will be stopped too, and the Watchdog Timer function is disabled. In this mode, the LVDEN is must set to "0". If the LVDEN is set to "1", it won't enter the SLEEP0 Mode.

SLEEP1 Mode

The SLEEP1 Mode is entered when an HALT instruction is executed and when the IDLEN bit in the SMOD register is low. In the SLEEP1 mode the CPU will be stopped. However the f_S will continue to operate if the LVDEN is “1” or the Watchdog Timer function is enabled as its clock source is from the f_{SUB}.

IDLE0 Mode

The IDLE0 Mode is entered when a HALT instruction is executed and when the IDLEN bit in the SMOD register is high and the FSYSON bit in the SMOD1 register is low. In the IDLE0 Mode the system oscillator will be inhibited from driving the CPU but some peripheral functions will remain operational such as the Watchdog Timer, TMs and LCD driver. In the IDLE0 Mode, the system oscillator will be stopped while the Watchdog Timer clock, f_S, will be on.

IDLE1 Mode

The IDLE1 Mode is entered when an HALT instruction is executed and when the IDLEN bit in the SMOD register is high and the FSYSON bit in the SMOD1 register is high. In the IDLE1 Mode the system oscillator will be inhibited from driving the CPU but may continue to provide a clock source to keep some peripheral functions operational such as the Watchdog Timer, TMs and SIM. In the IDLE1 Mode, the system oscillator will continue to run, and this system oscillator may be high speed or low speed system oscillator. In the IDLE1 Mode the Watchdog Timer clock, f_S, will also be on.

Control Register

A register pair, SMOD and SMOD1, is used for overall control of the internal clocks within the device.

SMOD Register

Bit	7	6	5	4	3	2	1	0
Name	CKS2	CKS1	CKS0	FSTEN	LTO	HTO	IDLEN	HLCLK
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W
POR	0	0	0	0	0	0	1	1

Bit 7~5 **CKS2~CKS0:** The system clock selection when HLCLK is "0"
000: f_{SUB} (f_{LXT} or f_{LIRC})
001: f_{SUB} (f_{LXT} or f_{LIRC})
010: $f_H/64$
011: $f_H/32$
100: $f_H/16$
101: $f_H/8$
110: $f_H/4$
111: $f_H/2$

These three bits are used to select which clock is used as the system clock source. In addition to the system clock source the LIRC, a divided version of the high speed system oscillator can also be chosen as the system clock source.

Bit 4 **FSTEN:** Fast Wake-up Control (only for HXT)
0: Disable
1: Enable

This is the Fast Wake-up Control bit which determines if the f_{SUB} clock source is initially used after the device wakes up. When the bit is high, the f_{SUB} clock source can be used as a temporary system clock to provide a faster wake up time as the f_{SUB} clock is available.

Bit 3 **LTO:** Low speed system oscillator ready flag
0: Not ready
1: Ready

This is the low speed system oscillator ready flag which indicates when the low speed system oscillator is stable after power on reset or a wake-up has occurred. The flag will be low when in the SLEEP0 Mode but after a wake-up has occurred, the flag will change to a high level after 1~2 clock cycles if the LIRC oscillator is used.

Bit 2 **HTO:** High speed system oscillator ready flag
0: Not ready
1: Ready

This is the high speed system oscillator ready flag which indicates when the high speed system oscillator is stable. This flag is cleared to "0" by hardware when the device is powered on and then changes to a high level after the high speed system oscillator is stable. Therefore this flag will always be read as "1" by the application program after device power-on. The flag will be low when in the SLEEP or IDLE0 Mode but after a wake-up has occurred, the flag will change to a high level after 1024 clock cycles if the HXT oscillator is used.

bit 1 **IDLEN:** IDLE Mode control
0: Disable
1: Enable

This is the IDLE Mode Control bit and determines what happens when the HALT instruction is executed. If this bit is high, when a HALT instruction is executed the device will enter the IDLE Mode. In the IDLE1 Mode the CPU will stop running but the system clock will continue to keep the peripheral functions operational, if FSYSON bit is high. If FSYSON bit is low, the CPU and the system clock will all stop in IDLE0 mode. If the bit is low the device will enter the SLEEP Mode when a HALT instruction is executed.

bit 0 **HLCLK:** System clock selection
0: $f_H/2$~$f_H/64$ or f_{SUB}
1: f_H

This bit is used to select if the f_H clock or the $f_H/2$~$f_H/64$ or f_{SUB} clock is used as the system clock. When the bit is high the f_H clock will be selected and if low the $f_H/2$~$f_H/64$ or f_{SUB} clock will be selected. When system clock switches from the f_H clock to the f_{SUB} clock and the f_H clock will be automatically switched off to conserve power.

SMOD1 Register

Bit	7	6	5	4	3	2	1	0
Name	FSYSON	—	—	—	—	LVRF	LRF	WRF
R/W	R/W	—	—	—	—	R/W	R/W	R/W
POR	0	—	—	—	—	x	0	0

"x" unknown

Bit 7 **FSYSON:** f_{SYS} Control in IDLE Mode
0: Disable
1: Enable

Bit 6~3 Unimplemented, read as "0"

Bit 2 **LVRF:** LVR function reset flag
0: Not occurred
1: Occurred

This bit is set to 1 when a specific Low Voltage Reset situation occurs. This bit can only be cleared to 0 by the application program.

Bit 1 **LRF:** LVR Control register software reset flag
0: Not occurred
1: Occurred

This bit is set to 1 if the LVRC register contains any non defined LVR voltage register values. This in effect acts like a software-reset function. This bit can only be cleared to 0 by the application program.

bit 0 **WRF:** WDT Control register software reset flag
0: Not occurred
1: Occurred

This bit is set to 1 by the WDT Control register software reset and cleared by the application program. Note that this bit can only be cleared to 0 by the application program.

Fast Wake-up

To minimise power consumption the device can enter the SLEEP or IDLE0 Mode, where the system clock source to the device will be stopped. However when the device is woken up again, it can take a considerable time for the original system oscillator to restart, stabilize and allow normal operation to resume. To ensure the device is up and running as fast as possible a Fast Wake-up function is provided, which allows f_{SUB}, namely the LIRC oscillator, to act as a temporary clock to first drive the system until the original system oscillator has stabilised. As the clock source for the Fast Wake-up function is f_{SUB}, the Fast Wake-up function is only available in the SLEEP1 and IDLE0 modes. When the device is woken up from the SLEEP0 mode, the FastWake-up function has no effect because the f_{SUB} clock is stopped. The FastWake-up enable/disable function is controlled using the FSTEN bit in the SMOD register.

If the HXT oscillator is selected as the NORMAL Mode system clock and if the Fast Wake-up function is enabled, then it will take one to two t_{SUB} clock cycles of the LIRC oscillator for the system to wake-up. The system will then initially run under the f_{SUB} clock source until 1024 HXT clock cycles have elapsed, at which point the HTO flag will switch high and the system will switch over to operating from the HXT oscillator.

If the ERC/HIRC or EC/LIRC oscillator is used as the system oscillator, then it will take15~16 clock cycles of the ERC/HIRC oscillator or 1~2 clock cycles of the LIRC osrillator respectively to wake up the system from the SLEEP or IDLE0 Mode. The Fast Wake-up bit, FSTEN will have no effect in these cases.

System Oscillator	FSTEN Bit	Wake-up Time (SLEEP0 Mode)	Wake-up Time (SLEEP1 Mode)	Wake-up Time (IDLE0 Mode)	Wake-up Time (IDLE1 Mode)
HXT	0	1024 HXT cycles	1024 HXT cycles		1~2 HXT cycles
	1	1024 HXT cycles	1~2 f_{SUB} cycles (System runs with f_{SUB} first for 1024 HXT cycles and then switches over to run with the HXT clock)		1~2 HXT cycles
ERC	x	15~16 ERC cycles	15~16 ERC cycles		1~2 ERC cycles
HIRC	x	15~16 HIRC cycles	15~16 HIRC cycles		1~2 HIRC cycles
EC	x	1~2 EC cycles	1~2 EC cycles		1~2 EC cycles
LIRC	x	1~2 LIRC cycles	1~2 LIRC cycles		1~2 LIRC cycles
LXT	x	1024 LXT cycles	1024 LXT cycles		1~2 LXT cycles

Wake-up Times

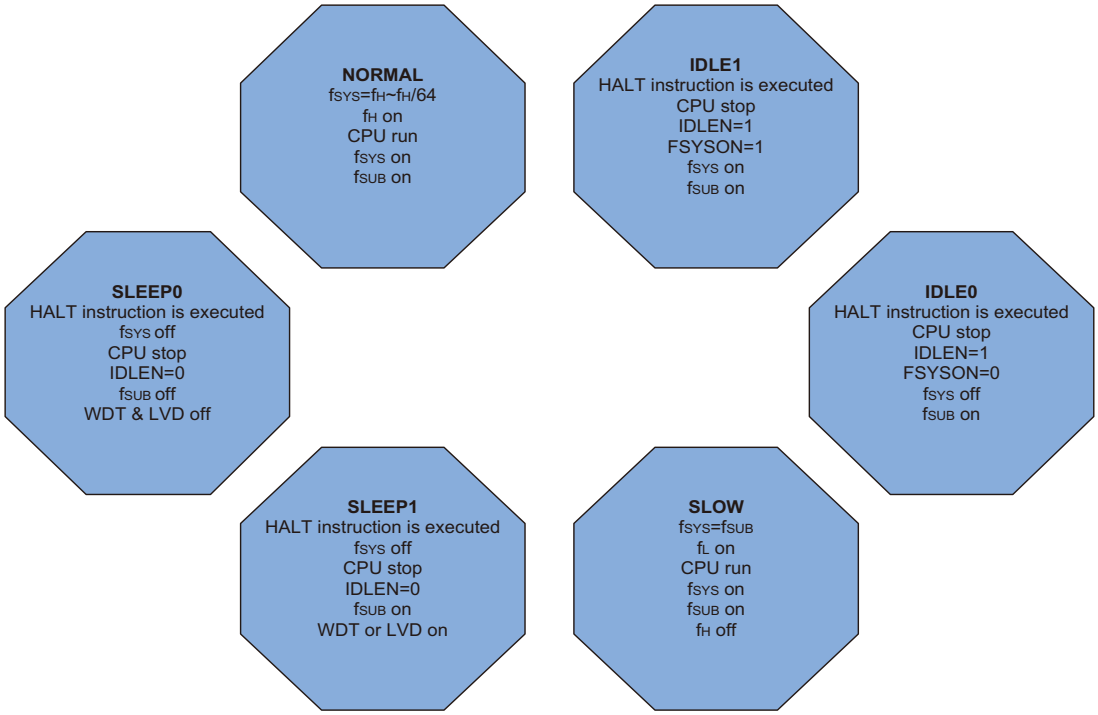
Note that if the Watchdog Timer is disabled, which means that the f_{SUB} clock drived from the LXT or LIRC oscillator is off, then there will be no Fast Wake-up function available when the device wakes-up from the SLEEP0 Mode.

Operating Mode Switching

The device can switch between operating modes dynamically allowing the user to select the best performance/power ratio for the present task in hand. In this way microcontroller operations that do not require high performance can be executed using slower clocks thus requiring less operating current and prolonging battery life in portable applications.

In simple terms, Mode Switching between the NORMAL Mode and SLOW Mode is executed using the HLCLK bit and CKS2~CKS0 bits in the SMOD register while Mode Switching from the NORMAL/SLOW Modes to the SLEEP/IDLE Modes is executed via the HALT instruction. When a HALT instruction is executed, whether the device enters the IDLE Mode or the SLEEP Mode is determined by the condition of the IDLEN bit in the SMOD register and FSYSON in the SMOD1 register.

When the HLCLK bit switches to a low level, which implies that clock source is switched from the high speed clock source, f_H, to the clock source, $f_H/2$~$f_H/64$ or f_{SUB}. If the clock is from the f_{SUB}, the high speed clock source will stop running to conserve power. When this happens it must be noted that the $f_H/16$ and $f_H/64$ internal clock sources will also stop running, which may affect the operation of other internal functions such as the TMs and the LCD driver. The accompanying flowchart shows what happens when the device moves between the various operating modes.



Operating Mode Switching and Wake-up

The device can switch between operating modes dynamically allowing the user to select the best performance/power ratio for the present task in hand. In this way microcontroller operations that do not require high performance can be executed using slower clocks thus requiring less operating current and prolonging battery life in portable applications.

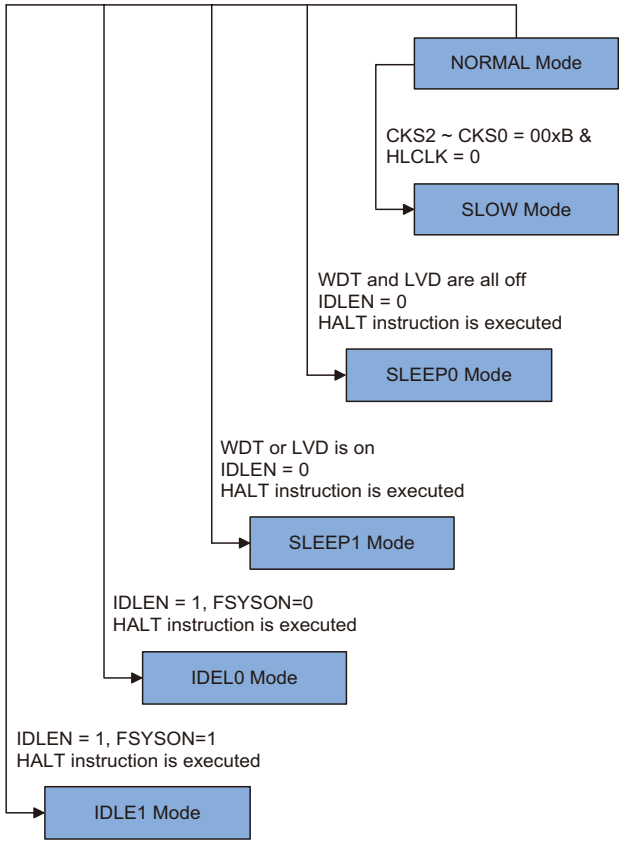
In simple terms, Mode Switching between the NORMAL Mode and SLOW Mode is executed using the HLCLK bit and CKS2~CKS0 bits in the SMOD register while Mode Switching from the NORMAL/SLOW Modes to the SLEEP/IDLE Modes is executed via the HALT instruction. When a HALT instruction is executed, whether the device enters the IDLE Mode or the SLEEP Mode is determined by the condition of the IDLEN bit in the SMOD register and FSYSON in the WDTC register.

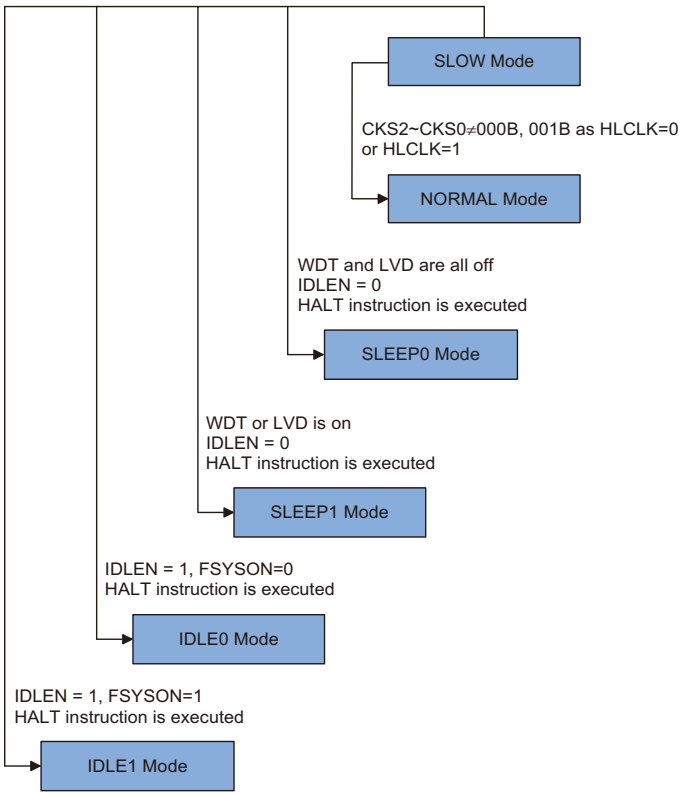
When the HLCLK bit switches to a low level, which implies that clock source is switched from the high speed clock source, f_H, to the clock source, $f_H/2$~$f_H/64$ or f_L. If the clock is from the f_L, the high speed clock source will stop running to conserve power. When this happens it must be noted that the $f_H/16$ and $f_H/64$ internal clock sources will also stop running, which may affect the operation of other internal functions such as the TMs and the SIM. The accompanying flowchart shows what happens when the device moves between the various operating modes.

NORMAL Mode to SLOW Mode Switching

When running in the NORMAL Mode, which uses the high speed system oscillator, and therefore consumes more power, the system clock can switch to run in the SLOW Mode by set the HLCLK bit to "0" and set the CKS2~CKS0 bits to "000" or "001" in the SMOD register. This will then use the low speed system oscillator which will consume less power. Users may decide to do this for certain operations which do not require high performance and can subsequently reduce power consumption.

The SLOW Mode is sourced from the LXT or the LIRC oscillators and therefore requires these oscillators to be stable before full mode switching occurs. This is monitored using the LTO bit in the SMOD register.





SLOW Mode to NORMAL Mode Switching

In SLOW Mode the system uses either the LXT or LIRC low speed system oscillator. To switch back to the NORMAL Mode, where the high speed system oscillator is used, the HLCLK bit should be set to "1" or HLCLK bit is "0", but CKS2~CKS0 is set to "010", "011", "100", "101", "110" or "111". As a certain amount of time will be required for the high frequency clock to stabilise, the status of the HTO bit is checked. The amount of time required for high speed system oscillator stabilization depends upon which high speed system oscillator type is used.

Entering the SLEEP0 Mode

There is only one way for the device to enter the SLEEP0 Mode and that is to execute the "HALT" instruction in the application program with the IDLEN bit in SMOD register equal to "0" and the WDT and LVD both off. When this instruction is executed under the conditions described above, the following will occur:

- The system clock and the f_{SUB} clock will be stopped and the application program will stop at the "HALT" instruction.
- The Data Memory contents and registers will maintain their present condition.
- The WDT will be cleared and stopped as the WDT is disabled.
- The I/O ports will maintain their present conditions.
- In the status register, the Power Down flag, PDF, will be set and the Watchdog time-out flag, TO, will be cleared.

Entering the SLEEP1 Mode

There is only one way for the device to enter the SLEEP1 Mode and that is to execute the "HALT" instruction in the application program with the IDLEN bit in SMOD register equal to "0" and the WDT or LVD on. When this instruction is executed under the conditions described above, the following will occur:

- The system clock will be stopped and the application program will stop at the "HALT" instruction, but the WDT or LVD will remain with the clock source coming from the f_{SUB} clock.
- The Data Memory contents and registers will maintain their present condition.
- The WDT will be cleared and resume counting as the WDT is enabled and its clock source is selected to come from the f_{SUB} clock.
- The I/O ports will maintain their present conditions.
- In the status register, the Power Down flag, PDF, will be set and the Watchdog time-out flag, TO, will be cleared.

Entering the IDLE0 Mode

There is only one way for the device to enter the IDLE0 Mode and that is to execute the "HALT" instruction in the application program with the IDLEN bit in SMOD register equal to "1" and the FSYSON bit in SMOD1 register equal to "0". When this instruction is executed under the conditions described above, the following will occur:

- The system clock will be stopped and the application program will stop at the "HALT" instruction, but the f_{SUB} clock will be on.
- The Data Memory contents and registers will maintain their present condition.
- The WDT will be cleared and resume counting as the WDT clock source is derived from the f_{SUB} clock.
- The I/O ports will maintain their present conditions.
- In the status register, the Power Down flag, PDF, will be set and the Watchdog time-out flag, TO, will be cleared.

Entering the IDLE1 Mode

There is only one way for the device to enter the IDLE1 Mode and that is to execute the "HALT" instruction in the application program with the IDLEN bit in SMOD register equal to "1" and the FSYSON bit in SMOD1 register equal to "1". When this instruction is executed under the conditions described above, the following will occur:

- The system clock and f_{SUB} clock will be on and the application program will stop at the "HALT" instruction.
- The Data Memory contents and registers will maintain their present condition.
- The WDT will be cleared and resume counting as the WDT clock source is derived from the f_{SUB} clock.
- The I/O ports will maintain their present conditions.
- In the status register, the Power Down flag, PDF, will be set and the Watchdog time-out flag, TO, will be cleared.

Standby Current Considerations

As the main reason for entering the SLEEP or IDLE Mode is to keep the current consumption of the device to as low a value as possible, perhaps only in the order of several micro-amps except in the IDLE1 Mode, there are other considerations which must also be taken into account by the circuit designer if the power consumption is to be minimised. Special attention must be made to the I/O pins on the device. All high-impedance input pins must be connected to either a fixed high or low level as any floating input pins could create internal oscillations and result in increased current consumption. This also applies to devices which have different package types, as there may be unbonbed pins. These must either be setup as outputs or if setup as inputs must have pull-high resistors connected.

Care must also be taken with the loads, which are connected to I/O pins, which are setup as outputs. These should be placed in a condition in which minimum current is drawn or connected only to external circuits that do not draw current, such as other CMOS inputs. Also note that additional standby current will also be required if the configuration options have enabled the LXT or LIRC oscillator.

In the IDLE1 Mode the system oscillator is on, if the system oscillator is from the high speed system oscillator, the additional standby current will also be perhaps in the order of several hundred micro-amps.

Wake-up

After the system enters the SLEEP or IDLE Mode, it can be woken up from one of various sources listed as follows:

- An external reset
- An external falling edge on Port A
- A system interrupt
- A WDT overflow

If the system is woken up by an external reset, the device will experience a full system reset, however, if the device is woken up by a WDT overflow, a Watchdog Timer reset will be initiated. Although both of these wake-up methods will initiate a reset operation, the actual source of the wake-up can be determined by examining the TO and PDF flags. The PDF flag is cleared by a system power-up or executing the clear Watchdog Timer instructions and is set when executing the "HALT" instruction. The TO flag is set if a WDT time-out occurs, and causes a wake-up that only resets the Program Counter and Stack Pointer, the other flags remain in their original status.

Each pin on Port A can be setup using the PAWU register to permit a negative transition on the pin to wake-up the system. When a Port A pin wake-up occurs, the program will resume execution at the instruction following the "HALT" instruction. If the system is woken up by an interrupt, then two possible situations may occur. The first is where the related interrupt is disabled or the interrupt is enabled but the stack is full, in which case the program will resume execution at the instruction following the "HALT" instruction. In this situation, the interrupt which woke-up the device will not be immediately serviced, but will rather be serviced later when the related interrupt is finally enabled or when a stack level becomes free. The other situation is where the related interrupt is enabled and the stack is not full, in which case the regular interrupt response takes place. If an interrupt request flag is set high before entering the SLEEP or IDLE Mode, the wake-up function of the related interrupt will be disabled.

Programming Considerations

The HXT and LXT oscillators both use the same SST counter. For example, if the system is woken up from the SLEEP0 Mode and both the HXT and LXT oscillators need to start-up from an off state. The LXT oscillator uses the SST counter after HXT oscillator has finished its SST period.

- If the device is woken up from the SLEEP0 Mode to the NORMAL Mode, the high speed system oscillator needs an SST period. The device will execute first instruction after HTO is "1". At this time, the LXT oscillator may not be stable if f_{SUB} is from LXT oscillator. The same situation occurs in the power-on state. The LXT oscillator is not ready yet when the first instruction is executed.
- If the device is woken up from the SLEEP1 Mode to NORMAL Mode, and the system clock source is from HXT oscillator and FSTEN is “1”, the system clock can be switched to the LXT or LIRC oscillator after wake up.
- There are peripheral functions, such as WDT, TMs and LCD driver, for which the f_{SYS} is used. If the system clock source is switched from f_H to f_{SUB}, the clock source to the peripheral functions mentioned above will change accordingly.
- The on/off condition of f_{SUB} and f_S depends upon whether the WDT is enabled or disabled as the WDT clock source is selected from f_{SUB}.

Watchdog Timer

The Watchdog Timer is provided to prevent program malfunctions or sequences from jumping to unknown locations, due to certain uncontrollable external events such as electrical noise.

Watchdog Timer Clock Source

The Watchdog Timer clock source is provided by the internal clock f_S, which is in turn supplied by the f_{SUB} clock. The f_{SUB} clock can be sourced from either the LXT or LIRC oscillator selected by a configuration option. The LIRC internal oscillator has an approximate frequency of 32kHz and this specified internal clock period can vary with V_{DD}, temperature and process variations. The LXT oscillator is supplied by an external 32.768kHz crystal. The Watchdog Timer source clock is then subdivided by a ratio of 2^8 to 2^{18} to give longer timeouts, the actual value being chosen using the WS2~WS0 bits in the WDTC register.

Watchdog Timer Control Register

A single register, WDTC, controls the required time-out period as well as the enable/disable operation. This register controls the overall operation of the Watchdog Timer.

WDTC Register

Bit	7	6	5	4	3	2	1	0
Name	WE4	WE3	WE2	WE1	WE0	WS2	WS1	WS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	1	0	1	0	0	1	1

Bit 7~3 **WE4~WE0:** WDT function enable control
If the WDT configuration option is "always enable":
10101 or 01010: Enabled
Other Values: Reset MCU
If the WDT configuration option is "application program enable":
10101: Disabled
01010: Enabled
Other Values: Reset MCU
If these bits are changed due to adverse environmental conditions, the microcontroller will be reset. The reset operation will be activated after 2~3 LIRC clock cycles and the WRF bit in the CTRL register will be set to 1.

Bit 2~0 **WS2~WS0:** Select WDT Timeout Period
000: $2^8/f_S$
001: $2^{10}/f_S$
010: $2^{12}/f_S$
011: $2^{14}/f_S$
100: $2^{15}/f_S$
101: $2^{16}/f_S$
110: $2^{17}/f_S$
111: $2^{18}/f_S$
These three bits determine the division ratio of the Watchdog Timer source clock, which in turn determines the timeout period.

SMOD1 Register

Bit	7	6	5	4	3	2	1	0
Name	FSYSON	—	—	—	—	LVRF	LRF	WRF
R/W	R/W	—	—	—	—	R/W	R/W	R/W
POR	0	—	—	—	—	x	0	0

"x" unknown

Bit 7 **FSYSON:** f_{SYS} Control in IDLE Mode
Described elsewhere.

Bit 6~3 Unimplemented, read as "0"

Bit 2 **LVRF:** LVR function reset flag
Described elsewhere.

Bit 1 **LRF:** LVR Control register software reset flag
Described elsewhere.

bit 0 **WRF:** WDT Control register software reset flag
0: Not occurred
1: Occurred
This bit is set to 1 by the WDT Control register software reset and cleared by the application program. Note that this bit can only be cleared to 0 by the application program.

Watchdog Timer Operation

The Watchdog Timer operates by providing a device reset when its timer overflows. This means that in the application program and during normal operation the user has to strategically clear the Watchdog Timer before it overflows to prevent the Watchdog Timer from executing a reset. This is done using the clear watchdog instructions. If the program malfunctions for whatever reason, jumps to an unknown location, or enters an endless loop, these clear instructions will not be executed in the correct manner, in which case the Watchdog Timer will overflow and reset the device. The Watchdog Timer function is determined using a configuration option. With regard to the Watchdog Timer enable/disable function, there are also five bits, WE4~WE0, in the WDTC register to offer the additional enable/disable control and reset control of the Watchdog Timer. If the WDT function configuration option is determined that the WDT function is always enabled, the WE4~WE0 bits still have effects on the WDT function. When the WE4~WE0 bits value is equal to 01010B or 10101B, the WDT function is enabled. However, if the WE4~WE0 bits are changed to any other values except 01010B and 10101B, which is caused by the environmental noise, it will reset the microcontroller after 2~3 f_{SUB} clock cycles. If the WDT function configuration option is determined that the WDT function is controlled by the WDTC control register using the application program, the WE4~WE0 values can determine which mode the WDT operates in. The WDT function will be disabled when the WE4~WE0 bits are set to a value of 10101B while the WDT function will be enabled if the WE4~WE0 bits are equal to 01010B. If the WE4~WE0 bits are set to any other values, other than 01010B and 10101B, it will reset the device after 2~3 f_{SUB} clock cycles. After power on these bits will have a value of 01010B.

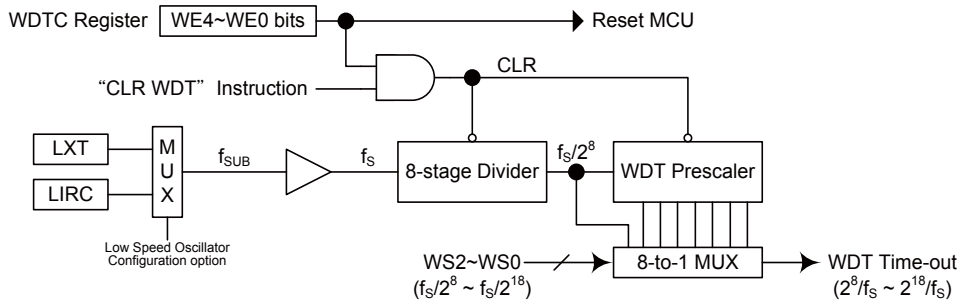
WDT Function Control	WE4~WE0 Bits	WDT Function
Application Program Enabled	10101B	Disable
	01010B	Enable
	Any other value	Reset MCU
Always Enabled	01010B or 10101B	Enable
	Any other value	Reset MCU

Watchdog Timer Enable/Disable Control

Under normal program operation, a Watchdog Timer time-out will initialise a device reset and set the status bit TO. However, if the system is in the SLEEP or IDLE Mode, when a Watchdog Timer time-out occurs, the TO bit in the status register will be set and only the Program Counter and Stack Pointer will be reset. Three methods can be adopted to clear the contents of the Watchdog Timer. The first is a WDT reset, which means a certain value except 01010B and 10101B written into the WE4~WE0 field, the second is using the Watchdog Timer software clear instruction and the third is via a HALT instruction.

There is only one method of using software instruction to clear the Watchdog Timer. That is to use the single "CLR WDT" instruction to clear the WDT contents.

The maximum time out period is when the 2^{18} division ratio is selected. As an example, with a 32kHz LIRC oscillator as its source clock, this will give a maximum watchdog period of around 8 second for the 2^{18} division ratio, and a minimum timeout of 7.8ms for the 2^8 division ration.



Watchdog Timer

Reset and Initialisation

A reset function is a fundamental part of any microcontroller ensuring that the device can be set to some predetermined condition irrespective of outside parameters. The most important reset condition is after power is first applied to the microcontroller. In this case, internal circuitry will ensure that the microcontroller, after a short delay, will be in a well defined state and ready to execute the first program instruction. After this power-on reset, certain important internal registers will be set to defined states before the program commences. One of these registers is the Program Counter, which will be reset to zero forcing the microcontroller to begin program execution from the lowest Program Memory address.

In addition to the power-on reset, situations may arise where it is necessary to forcefully apply a reset condition when the is running. One example of this is where after power has been applied and the is already running, the $\overline{\text{RES}}$ line is forcefully pulled low. In such a case, known as a normal operation reset, some of the registers remain unchanged allowing the to proceed with normal operation after the reset line is allowed to return high.

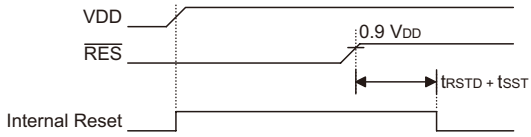
Another type of reset is when the Watchdog Timer overflows and resets the . All types of reset operations result in different register conditions being setup. Another reset exists in the form of a Low Voltage Reset, LVR, where a full reset, similar to the $\overline{\text{RES}}$ reset is implemented in situations where the power supply voltage falls below a certain threshold.

Reset Functions

There are five ways in which a reset can occur, through events occurring both internally and externally:

Power-on Reset

The most fundamental and unavoidable reset is the one that occurs after power is first applied to the. As well as ensuring that the Program Memory begins execution from the first memory address, a power-on reset also ensures that certain other registers are preset to known conditions. All the I/O port and port control registers will power up in a high condition ensuring that all pins will be first set to inputs.



Power-On Reset Timing Chart

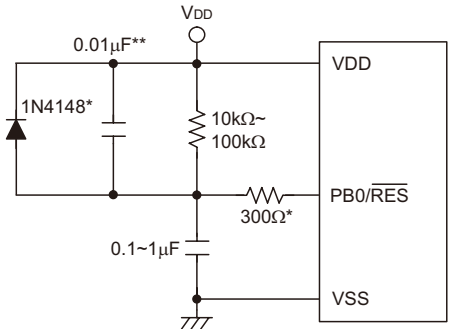
Note: t_{RSTD} is power-on delay, typical time=50ms

RES Pin

As the reset pin is shared with an I/O pin, the reset function must be selected using a configuration option. Although the has an internal RC reset function, if the V_{DD} power supply rise time is not fast enough or does not stabilise quickly at power-on, the internal reset function may be incapable of providing proper reset operation. For this reason it is recommended that an external RC network is connected to the $\overline{RES}$ pin, whose additional time delay will ensure that the $\overline{RES}$ pin remains low for an extended period to allow the power supply to stabilise. During this time delay, normal operation of the will be inhibited. After the $\overline{RES}$ line reaches a certain voltage value, the reset delay time t_{RSTD} is invoked to provide an extra delay time after which the will begin normal operation. The abbreviation SST in the figures stands for System Start-up Timer.

For most applications a resistor connected between VDD and the $\overline{RES}$ pin and a capacitor connected between V_{SS} and the $\overline{RES}$ pin will provide a suitable external reset circuit. Any wiring connected to the $\overline{RES}$ pin should be kept as short as possible to minimise any stray noise interference.

For applications that operate within an environment where more noise is present the Enhanced Reset Circuit shown is recommended.



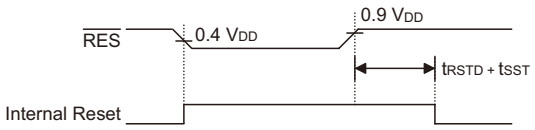
External $\overline{RES}$ Circuit

Note: * It is recommended that this component is added for added ESD protection.

** It is recommended that this component is added in environments where power line noise is significant.

More information regarding external reset circuits is located in Application Note HA0075E on the Holtek website.

Pulling the $\overline{RES}$ Pin low using external hardware will also execute a device reset. In this case, as in the case of other resets, the Program Counter will reset to zero and program execution initiated from this point.

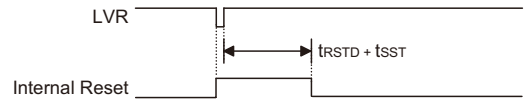


$\overline{RES}$ Reset Timing Chart

Note: t_{RSTD} is power-on delay, typical time=16.7ms

Low Voltage Reset – LVR

The microcontroller contains a low voltage reset circuit in order to monitor the supply voltage of the device. The LVR function is always enabled with a specific LVR voltage, V_{LVR}. If the supply voltage of the device drops to within a range of 0.9V~V_{LVR} such as might occur when changing the battery, the LVR will automatically reset the device internally and the LVRF bit in the SMOD1 register will also be set to 1. For a valid LVR signal, a low supply voltage, i.e., a voltage in the range between 0.9V~V_{LVR} must exist for a time greater than that specified by t_{LVR} in the A.C. characteristics. If the low supply voltage state does not exceed this value, the LVR will ignore the low supply voltage and will not perform a reset function. The actual V_{LVR} value can be selected by the LVS bits in the LVRC register. If the LVS7~LVS0 bits have any other value, which may perhaps occur due to adverse environmental conditions such as noise, the LVR will reset the device after 2~3 f_{SUB} clock cycles. When this happens, the LRF bit in the SMOD1 register will be set to 1. After power on the register will have the value of 01010101B. Note that the LVR function will be automatically disabled when the device enters the power down mode.



Note: t_{RSTD} is power-on delay, typical time=50ms

Low Voltage Reset Timing Chart

• **LVRC Register**

Bit	7	6	5	4	3	2	1	0
Name	LVS7	LVS6	LVS5	LVS4	LVS3	LVS2	LVS1	LVS0
R/W	R/W	R/W	R/W	R/W	R/w	R/w	R/W	R/W
POR	0	1	0	1	0	1	0	1

Bit 7~0 **LVS7~LVS0:** LVR voltage select
01010101: 2.1V
00110011: 2.55V
10011001: 3.15V
10101010: 3.8V
Any other values: Generates MCU reset – register is reset to POR value

When an actual low voltage condition occurs, as specified by one of the four defined LVR voltage values above, an MCU reset will be generated. The reset operation will be activated after 2~3 f_{SUB} clock cycles. In this situation the register contents will remain the same after such a reset occurs.

Any register value, other than the four defined register values above, will also result in the generation of an MCU reset. The reset operation will be activated after 2~3 f_{SUB} clock cycles. However in this situation the register contents will be reset to the POR value.

• **SMOD1 Register**

Bit	7	6	5	4	3	2	1	0
Name	FSYSON	—	—	—	—	LVRF	LRF	WRF
R/W	R/W	—	—	—	—	R/W	R/W	R/W
POR	0	—	—	—	—	x	0	0

"x" unknown

Bit 7 **FSYSON:** f_{SYS} Control in IDLE Mode
Described elsewhere.

Bit 6~3 Unimplemented, read as "0"

Bit 2 **LVRF:** LVR function reset flag
0: Not occurred
1: Occurred

This bit is set to 1 when a specific Low Voltage Reset situation condition occurs. This bit can only be cleared to 0 by the application program.

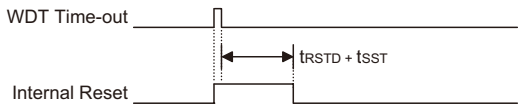
Bit 1 **LRF:** LVR Control register software reset flag
0: Not occurred
1: Occurred

This bit is set to 1 if the LVRC register contains any non defined LVR voltage register values. This in effect acts like a software-reset function. This bit can only be cleared to 0 by the application program.

bit 0 **WRF:** WDT Control register software reset flag
Described elsewhere.

Watchdog Time-out Reset during Normal Operation

The Watchdog time-out Reset during normal operation is the same as a hardware $\overline{RES}$ pin reset except that the Watchdog time-out flag TO will be set to "1".

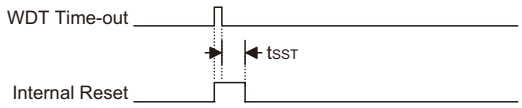


WDT Time-out Reset during Normal Operation Timing Chart

Note: t_{RSTD} is power-on delay, typical time=16.7ms

Watchdog Time-out Reset during SLEEP or IDLE Mode

The Watchdog time-out Reset during SLEEP or IDLE Mode is a little different from other kinds of reset. Most of the conditions remain unchanged except that the Program Counter and the Stack Pointer will be cleared to "0" and the TO flag will be set to "1". Refer to the A.C. Characteristics for t_{SST} details.



WDT Time-out Reset during SLEEP or IDLE Timing Chart

Note: The t_{SST} is 15~16 clock cycles if the system clock source is provided by ERC or HIRC. The t_{SST} is 1024 clock for HXT or LXT. The t_{SST} is 1~2 clock for LIRC.

Reset Initial Conditions

The different types of reset described affect the reset flags in different ways. These flags, known as PDF and TO are located in the status register and are controlled by various microcontroller operations, such as the SLEEP or IDLE Mode function or Watchdog Timer. The reset flags are shown in the table:

TO	PDF	RESET Conditions
0	0	Power-on reset
u	u	$\overline{RES}$ or LVR reset during NORMAL or SLOW Mode operation
1	u	WDT time-out reset during NORMAL or SLOW Mode operation
1	1	WDT time-out reset during IDLE or SLEEP Mode operation

"u" stands for unchanged

The following table indicates the way in which the various components of the microcontroller are affected after a power-on reset occurs.

Item	Condition After RESET
Program Counter	Reset to zero
Interrupts	All interrupts will be disabled
WDT, Time Base	Clear after reset, WDT begins counting
Timer Modules	Timer Modules will be turned off
Input/Output Ports	I/O ports will be setup as inputs
Stack Pointer	Stack Pointer will point to the top of the stack

The different kinds of resets all affect the internal registers of the microcontroller in different ways. To ensure reliable continuation of normal program execution after a reset occurs, it is important to know what condition the microcontroller is in after a particular reset occurs. The following table describes how each type of reset affects the microcontroller internal registers.

Register Reset Status Table

Register	HT69F30A	HT69F40A	HT69F50A	Power-on Reset	$\overline{RES}$ or LVR Reset	WDT Time-out (Normal Operation)	WDT Time-out (HALT)
IAR0	●	●	●	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
MP0	●	●	●	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
IAR1	●	●	●	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
MP1	●	●	●	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
BP	●			---- ---0	---- ---0	---- ---0	---- ---u
BP		●	●	---- --00	---- --00	---- --00	---- --uu
ACC	●	●	●	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
PCL	●	●	●	0000 0000	0000 0000	0000 0000	0000 0000
TBLP	●	●	●	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLH	●	●	●	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBHP	●			---- -xxx	---- -uuu	---- -uuu	---- -uuu
TBHP		●		---- xxxx	---- uuuu	---- uuuu	---- uuuu
TBHP			●	---x xxxx	---u uuuu	---u uuuu	---u uuuu
STATUS	●	●	●	--00 xxxx	--uu uuuu	--1u uuuu	--11 uuuu
SMOD	●	●	●	0000 0011	0000 0011	0000 0011	uuuu uuuu
LVDC	●	●	●	--00 -000	--00 -000	--00 -000	--uu -uuu
INTEG	●	●	●	--00 0000	--00 0000	--00 0000	--uu uuuu
WDTC	●	●	●	0101 0011	0101 0011	0101 0011	uuuu uuuu
TBC	●	●	●	0011 0111	0011 0111	0011 0111	uuuu uuuu
INTC0	●	●	●	-000 0000	-000 0000	-000 0000	-uuu uuuu
INTC1	●	●	●	0000 0000	0000 0000	0000 0000	uuuu uuuu
SMOD1	●	●	●	0--- -x00	0--- -1uu	0--- -uuu	u--- -uuu
LVRC	●	●	●	0101 0101	uuuu uuuu	0101 0101	uuuu uuuu
MFI0	●			--00 --00	--00 --00	--00 --00	--uu --uu
MFI0		●	●	0000 0000	0000 0000	0000 0000	uuuu uuuu
MFI1	●	●	●	-000 -000	-000 -000	-000 -000	-uuu -uuu
MFI2	●	●	●	--00 --00	--00 --00	--00 --00	--uu --uu
PAWU	●	●	●	0000 0000	0000 0000	0000 0000	uuuu uuuu
PAPU	●	●	●	0000 0000	0000 0000	0000 0000	uuuu uuuu
PA	●	●	●	1111 1111	1111 1111	1111 1111	uuuu uuuu
PAC	●	●	●	1111 1111	1111 1111	1111 1111	uuuu uuuu
PBPU	●			---- 0000	---- 0000	---- 0000	---- uuuu
PBPU		●	●	0000 0000	0000 0000	0000 0000	uuuu uuuu
PB	●			---- 1111	---- 1111	---- 1111	---- uuuu
PB		●	●	1111 1111	1111 1111	1111 1111	uuuu uuuu
PBC	●			---- 1111	---- 1111	---- 1111	---- uuuu
PBC		●	●	1111 1111	1111 1111	1111 1111	uuuu uuuu
PCPU	●	●		---- -000	---- -000	---- -000	---- -uuu
PCPU			●	-000 0000	-000 0000	-000 0000	-uuu uuuu

Register	HT69F30A	HT69F40A	HT69F50A	Power-on Reset	$\overline{RES}$ or LVR Reset	WDT Time-out (Normal Operation)	WDT Time-out (HALT)
PC	●	●		---- -111	---- -111	---- -111	---- -uuu
PC			●	-111 1111	-111 1111	-111 1111	-uuu uuuu
PCC	●	●		---- -111	---- -111	---- -111	---- -uuu
PCC			●	-111 1111	-111 1111	-111 1111	-uuu uuuu
PDPU	●	●	●	0000 0000	0000 0000	0000 0000	uuuu uuuu
PD	●	●	●	1111 1111	1111 1111	1111 1111	uuuu uuuu
PDC	●	●	●	1111 1111	1111 1111	1111 1111	uuuu uuuu
PEPU	●	●	●	0000 0000	0000 0000	0000 0000	uuuu uuuu
PE	●	●	●	1111 1111	1111 1111	1111 1111	uuuu uuuu
PEC	●	●	●	1111 1111	1111 1111	1111 1111	uuuu uuuu
PFPU	●	●	●	0000 0000	0000 0000	0000 0000	uuuu uuuu
PF	●	●	●	1111 1111	1111 1111	1111 1111	uuuu uuuu
PFC	●	●	●	1111 1111	1111 1111	1111 1111	uuuu uuuu
PGPU		●	●	0000 0000	0000 0000	0000 0000	uuuu uuuu
PG		●	●	1111 1111	1111 1111	1111 1111	uuuu uuuu
PGC		●	●	1111 1111	1111 1111	1111 1111	uuuu uuuu
PHPU			●	0000 0000	0000 0000	0000 0000	uuuu uuuu
PH			●	1111 1111	1111 1111	1111 1111	uuuu uuuu
PHC			●	1111 1111	1111 1111	1111 1111	uuuu uuuu
PAFS	●	●	●	-00- 0-0-	-00- 0-0-	-00- 0-0-	-uu- u-u-
PBFS		●	●	0000 ----	0000 ----	0000 ----	uuuu ----
PCFS	●	●	●	---- -000	---- -000	---- -000	---- -uuu
PDFS	●	●	●	1111 1111	1111 1111	1111 1111	uuuu uuuu
PEFS	●	●	●	1111 1111	1111 1111	1111 1111	uuuu uuuu
PFFS	●	●	●	1111 1111	1111 1111	1111 1111	uuuu uuuu
PGFS		●	●	1111 1111	1111 1111	1111 1111	uuuu uuuu
PHFS			●	1111 1111	1111 1111	1111 1111	uuuu uuuu
SFS	●			-000 ----	-000 ----	-000 ----	-uuu ----
SFS		●	●	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM0C0	●	●	●	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM0C1	●	●	●	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM0DL	●	●	●	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM0DH	●	●	●	---- --00	---- --00	---- --00	---- --uu
TM0AL	●	●	●	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM0AH	●	●	●	---- --00	---- --00	---- --00	---- --uu
EEC	●	●	●	---- 0000	---- 0000	---- 0000	---- uuuu
EEA	●			--00 0000	--00 0000	--00 0000	--uu uuuu
EEA		●		-000 0000	-000 0000	-000 0000	-uuu uuuu
EED	●	●	●	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1C0	●			0000 0000	0000 0000	0000 0000	uuuu uuuu

Register	HT69F30A	HT69F40A	HT69F50A	Power-on Reset	$\overline{RES}$ or LVR Reset	WDT Time-out (Normal Operation)	WDT Time-out (HALT)
TM1C1	●			0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1DL	●			0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1DH	●			---- --00	---- --00	---- --00	---- --uu
TM1AL	●			0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1AH	●			---- --00	---- --00	---- --00	---- --uu
TM1C0		●	●	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1C1		●	●	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1C2		●	●	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1DL		●	●	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1DH		●	●	---- --00	---- --00	---- --00	---- --uu
TM1AL		●	●	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1AH		●	●	---- --00	---- --00	---- --00	---- --uu
TM1BL		●	●	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM1BH		●	●	---- --00	---- --00	---- --00	---- --uu
TM2C0		●		0000 0000	0000 0000	0000 0000	uuuu uuuu
TM2C1		●		0000 0000	0000 0000	0000 0000	uuuu uuuu
TM2DL		●		0000 0000	0000 0000	0000 0000	uuuu uuuu
TM2DH		●		---- --00	---- --00	---- --00	---- --uu
TM2AL		●		0000 0000	0000 0000	0000 0000	uuuu uuuu
TM2AH		●		---- --00	---- --00	---- --00	---- --uu
TM2C0			●	0000 0---	0000 0---	0000 0---	uuuu u---
TM2C1			●	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM2DL			●	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM2DH			●	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM2AL			●	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM2AH			●	0000 0000	0000 0000	0000 0000	uuuu uuuu
TM2RP			●	0000 0000	0000 0000	0000 0000	uuuu uuuu
LCDC	●	●	●	0-0- 0000	0-0- 0000	0-0- 0000	u-u- uuuu

Note: "-" not implement
"u" means "unchanged"
"x" means "unknown"

Input/Output Ports

Holtek microcontrollers offer considerable flexibility on their I/O ports. With the input or output designation of every pin fully under user program control, pull-high selections for all ports and wake-up selections on certain pins, the user is provided with an I/O structure to meet the needs of a wide range of application possibilities.

The device provides bidirectional input/output lines labeled with port names PA~PH. These I/O ports are mapped to the RAM Data Memory with specific addresses as shown in the Special Purpose Data Memory table. All of these I/O ports can be used for input and output operations. For input operation, these ports are non-latching, which means the inputs must be ready at the T2 rising edge of instruction "MOV A, [m]", where m denotes the port address. For output operation, all the data is latched and remains unchanged until the output latch is rewritten.

I/O Port Register List

HT69F30A

Register Name	Bit							
	7	6	5	4	3	2	1	0
PAWU	PAWU7	PAWU6	PAWU5	PAWU4	PAWU3	PAWU2	PAWU1	PAWU0
PAPU	PAPU7	PAPU6	PAPU5	PAPU4	PAPU3	PAPU2	PAPU1	PAPU0
PA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
PAC	PAC7	PAC6	PAC5	PAC4	PAC3	PAC2	PAC1	PAC0
PBPU	—	—	—	—	PBPU3	PBPU2	PBPU1	PBPU0
PB	—	—	—	—	PB3	PB2	PB1	PB0
PBC	—	—	—	—	PBC3	PBC2	PBC1	PBC0
PCPU	—	—	—	—	—	PCPU2	PCPU1	PCPU0
PC	—	—	—	—	—	PC2	PC1	PC0
PCC	—	—	—	—	—	PCC2	PCC1	PCC0
PDPU	PDPU7	PDPU6	PDPU5	PDPU4	PDPU3	PDPU2	PDPU1	PDPU0
PD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
PDC	PDC7	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0
PEPU	PEPU7	PEPU6	PEPU5	PEPU4	PEPU3	PEPU2	PEPU1	PEPU0
PE	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
PEC	PEC7	PEC6	PEC5	PEC4	PEC3	PEC2	PEC1	PEC0
PFPU	PFPU7	PFPU6	PFPU5	PFPU4	PFPU3	PFPU2	PFPU1	PFPU0
PF	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
PFC	PFC7	PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0

HT69F40A

Register Name	Bit							
	7	6	5	4	3	2	1	0
PAWU	PAWU7	PAWU6	PAWU5	PAWU4	PAWU3	PAWU2	PAWU1	PAWU0
PAPU	PAPU7	PAPU6	PAPU5	PAPU4	PAPU3	PAPU2	PAPU1	PAPU0
PA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
PAC	PAC7	PAC6	PAC5	PAC4	PAC3	PAC2	PAC1	PAC0
PBPU	PBPU7	PBPU6	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0
PB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
PBC	PBC7	PBC6	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
PCPU	—	PCPU6	PCPU5	PCPU4	PCPU3	PCPU2	PCPU1	PCPU0
PC	—	PC6	PC5	PC4	PC3	PC2	PC1	PC0
PCC	—	PCC6	PCC4	PCC4	PCC3	PCC2	PCC1	PCC0
PDPU	PDPU7	PDPU6	PDPU5	PDPU4	PDPU3	PDPU2	PDPU1	PDPU0
PD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
PDC	PDC7	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0
PEPU	PEPU7	PEPU6	PEPU5	PEPU4	PEPU3	PEPU2	PEPU1	PEPU0
PE	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
PEC	PEC7	PEC6	PEC5	PEC4	PEC3	PEC2	PEC1	PEC0
PFPU	PFPU7	PFPU6	PFPU5	PFPU4	PFPU3	PFPU2	PFPU1	PFPU0
PF	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
PFC	PFC7	PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0
PGPU	PGPU7	PGPU6	PGPU5	PGPU4	PGPU3	PGPU2	PGPU1	PGPU0
PG	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
PGC	PGC7	PGC6	PGC5	PGC4	PGC3	PGC2	PGC1	PGC0

HT69F50A

Register Name	Bit							
	7	6	5	4	3	2	1	0
PAWU	PAWU7	PAWU6	PAWU5	PAWU4	PAWU3	PAWU2	PAWU1	PAWU0
PAPU	PAPU7	PAPU6	PAPU5	PAPU4	PAPU3	PAPU2	PAPU1	PAPU0
PA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
PAC	PAC7	PAC6	PAC5	PAC4	PAC3	PAC2	PAC1	PAC0
PBPU	PBPU7	PBPU6	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0
PB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
PBC	PBC7	PBC6	PBC5	PBC4	PBC3	PBC2	PBC1	PBC0
PCPU	—	PCPU6	PCPU5	PCPU4	PCPU3	PCPU2	PCPU1	PCPU0
PC	—	PC6	PC5	PC4	PC3	PC2	PC1	PC0
PCC	—	PCC6	PCC4	PCC4	PCC3	PCC2	PCC1	PCC0
PDPU	PDPU7	PDPU6	PDPU5	PDPU4	PDPU3	PDPU2	PDPU1	PDPU0
PD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
PDC	PDC7	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0
PEPU	PEPU7	PEPU6	PEPU5	PEPU4	PEPU3	PEPU2	PEPU1	PEPU0
PE	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
PEC	PEC7	PEC6	PEC5	PEC4	PEC3	PEC2	PEC1	PEC0
PFPU	PFPU7	PFPU6	PFPU5	PFPU4	PFPU3	PFPU2	PFPU1	PFPU0
PF	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
PFC	PFC7	PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0
PGPU	PGPU7	PGPU6	PGPU5	PGPU4	PGPU3	PGPU2	PGPU1	PGPU0
PG	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
PGC	PGC7	PGC6	PGC5	PGC4	PGC3	PGC2	PGC1	PGC0
PHPU	PHPU7	PHPU6	PHPU5	PHPU4	PHPU3	PHPU2	PHPU1	PHPU0
PH	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
PHC	PHC7	PHC6	PHC5	PHC4	PHC3	PHC2	PHC1	PHC0

"—" Unimplemented, read as "0"

PAWUn: PA wake-up function control
0: Disable
1: Enable

PAn/PBn/PCn/PDn/PEn/PFn/PGn/PHn: I/O Data bit
0: Data 0
1: Data 1

PACn/PBCn/PCCn/PDCn/PECn/PFCn/PGCn/PHCn: I/O type selection
0: Output
1: Input

PAPUn/PBPUn/PCPUn/PDPUn/PEPUn/PFPUn/PGPUn/PHPUn: Pull-high function control
0: Disable
1: Enable

Pull-high Resistors

Many product applications require pull-high resistors for their switch inputs usually requiring the use of an external resistor. To eliminate the need for these external resistors, all I/O pins, when configured as an input have the capability of being connected to an internal pull-high resistor. These pull-high resistors are selected using registers PAPU~PHPU, and are implemented using weak PMOS transistors.

Port A Wake-up

The HALT instruction forces the microcontroller into the SLEEP or IDLE Mode which preserves power, a feature that is important for battery and other low-power applications. Various methods exist to wake-up the microcontroller, one of which is to change the logic condition on one of the Port A pins from high to low. This function is especially suitable for applications that can be woken up via external switches. Each pin on Port A can be selected individually to have this wake-up feature using the PAWU register.

I/O Port Control Registers

Each Port has its own control register, known as PAC~PHC, which controls the input/output configuration. With this control register, each I/O pin with or without pull-high resistors can be reconfigured dynamically under software control. For the I/O pin to function as an input, the corresponding bit of the control register must be written as a "1". This will then allow the logic state of the input pin to be directly read by instructions. When the corresponding bit of the control register is written as a "0", the I/O pin will be setup as a CMOS output. If the pin is currently setup as an output, instructions can still be used to read the output register.

However, it should be noted that the program will in fact only read the status of the output data latch and not the actual logic status of the output pin.

Pin-shared Functions

The flexibility of the microcontroller range is greatly enhanced by the use of pins that have more than one function. Limited numbers of pins can force serious design constraints on designers but by supplying pins with multi-functions, many of these difficulties can be overcome. For these pins, the chosen function of the multi-function I/O pins is selected by a series of regiters via the application program control.

Pin-shared Function Selection Register List

- **HT69F30A**

Register Name	Bit							
	7	6	5	4	3	2	1	0
PAFS	—	PAFS6	PAFS5	—	PAFS3	—	PAFS1	—
PCFS	—	—	—	—	—	PCFS2	PCFS1	PCFS0
PDFS	PDFS7	PDFS6	PDFS5	PDFS4	PDFS3	PDFS2	PDFS1	PDFS0
PEFS	PEFS7	PEFS6	PEFS5	PEFS4	PEFS3	PEFS2	PEFS1	PEFS0
PFFS	PFFS7	PFFS6	PFFS5	PFFS4	PFFS3	PFFS2	PFFS1	PFFS0
SFS	—	SFS6	SFS5	SFS4	—	—	—	—

- **HT69F40A**

Register Name	Bit							
	7	6	5	4	3	2	1	0
PAFS	—	PAFS6	PAFS5	—	PAFS3	—	PAFS1	—
PBFS	PBFS7	PBFS6	PBFS5	PBFS4	—	—	—	—
PCFS	—	—	—	—	—	PCFS2	PCFS1	PCFS0
PDFS	PDFS7	PDFS6	PDFS5	PDFS4	PDFS3	PDFS2	PDFS1	PDFS0
PEFS	PEFS7	PEFS6	PEFS5	PEFS4	PEFS3	PEFS2	PEFS1	PEFS0
PFFS	PFFS7	PFFS6	PFFS5	PFFS4	PFFS3	PFFS2	PFFS1	PFFS0
PGFS	PGS7	PFGFS6	PGFS5	PGFS4	PGFS3	PGFS2	PGFS1	PGFS0
SFS	SFS7	SFS6	SFS5	SFS4	SFS3	SFS2	SFS1	SFS0

- **HT69F50A**

Register Name	Bit							
	7	6	5	4	3	2	1	0
PAFS	—	PAFS6	PAFS5	—	PAFS3	—	PAFS1	—
PBFS	PBFS7	PBFS6	PBFS5	PBFS4	—	—	—	—
PCFS	—	—	—	—	—	PCFS2	PCFS1	PCFS0
PDFS	PDFS7	PDFS6	PDFS5	PDFS4	PDFS3	PDFS2	PDFS1	PDFS0
PEFS	PEFS7	PEFS6	PEFS5	PEFS4	PEFS3	PEFS2	PEFS1	PEFS0
PFFS	PFFS7	PFFS6	PFFS5	PFFS4	PFFS3	PFFS2	PFFS1	PFFS0
PGFS	PGS7	PGFS6	PGFS5	PGFS4	PGFS3	PGFS2	PGFS1	PGFS0
PHFS	PHS7	PHFS6	PHFS5	PHFS4	PHFS3	PHFS2	PHFS1	PHFS0
SFS	SFS7	SFS6	SFS5	SFS4	SFS3	SFS2	SFS1	SFS0

PAFS Register – HT69F30A

Bit	7	6	5	4	3	2	1	0
Name	—	PAFS6	PAFS5	—	PAFS3	—	PAFS1	—
R/W	—	R/W	R/W	—	R/W	—	R/W	—
POR	—	0	0	—	0	—	0	—

Bit 7 Unimplemented, read as "0"

Bit 6 **PAFS6:** Port A 6 Function Selection
0: I/O
1: TP0_0

Bit 5 **PAFS5:** Port A 5 Function Selection
0: I/O
1: TP1_1

Bit 4 Unimplemented, read as "0"

Bit 3 **PAFS3:** Port A 3 Function Selection
0: I/O
1: TP1_0

Bit 2 Unimplemented, read as "0"

Bit 1 **PAFS1:** Port A 1 Function Selection
0: I/O
1: TP0_1

Bit 0 Unimplemented, read as "0"

PAFS Register – HT69F40A/HT69F50A

Bit	7	6	5	4	3	2	1	0
Name	—	PAFS6	PAFS5	—	PAFS3	—	PAFS1	—
R/W	—	R/W	R/W	—	R/W	—	R/W	—
POR	—	0	0	—	0	—	0	—

Bit 7 Unimplemented, read as "0"

Bit 6 **PAFS6:** Port A 6 Function Selection
0: I/O
1: TP0_0

Bit 5 **PAFS5:** Port A 5 Function Selection
0: I/O
1: TP2_1

Bit 4 Unimplemented, read as "0"

Bit 3 **PAFS3:** Port A 3 Function Selection
0: I/O
1: TP2_0

Bit 2 Unimplemented, read as "0"

Bit 1 **PAFS1:** Port A 1 Function Selection
0: I/O
1: TP0_1

Bit 0 Unimplemented, read as "0"

PBFS Register – HT69F40A/HT69F50A

Bit	7	6	5	4	3	2	1	0
Name	PBFS7	PBFS6	PBFS5	PBFS4	—	—	—	—
R/W	R/W	R/W	R/W	R/W	—	—	—	—
POR	0	0	0	0	—	—	—	—

Bit 7 **PBFS7:** Port B 7 Function Selection
0: I/O
1: TP1B_2

Bit 6 **PBFS6:** Port B 6 Function Selection
0: I/O
1: TP1B_1

Bit 5 **PBFS5:** Port B 5 Function Selection
0: I/O
1: TP1B_0

Bit 4 **PBFS4:** Port B 4 Function Selection
0: I/O
1: TP1A

Bit 3~0 Unimplemented, read as "0"

PCFS Register – HT69F30A/HT69F40A/HT69F50A

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	PCFS2	PCFS1	PCFS0
R/W	—	—	—	—	—	R/W	R/W	R/W
POR	—	—	—	—	—	0	0	0

Bit 7~3 Unimplemented, read as "0"

Bit 2 **PCFS2:** Port C 2 Function Selection
0: I/O
1: C2

Bit 1 **PCFS1:** Port C 1 Function Selection
0: I/O
1: C1

Bit 0 **PCFS0:** Port C 0 Function Selection
0: I/O
1: V2

PDFS Register – HT69F30A

Bit	7	6	5	4	3	2	1	0
Name	PDFS7	PDFS6	PDFS5	PDFS4	PDFS3	PDFS2	PDFS1	PDFS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

Bit 7 **PDFS7:** Port D 7 Function Selection
0: I/O
1: SEG7

Bit 6 **PDFS6:** Port D 6 Function Selection
0: I/O
1: SEG6

Bit 5 **PDFS5:** Port D 5 Function Selection
0: I/O
1: SEG5

Bit 4 **PDFS4:** Port D 4 Function Selection
0: I/O
1: SEG4

Bit 3 **PDFS3:** Port D 3 Function Selection
0: I/O
1: SEG3

Bit 2 **PDFS2:** Port D 2 Function Selection
0: I/O
1: SEG2

Bit 1 **PDFS1:** Port D 1 Function Selection
0: I/O
1: SEG1

Bit 0 **PDFS0:** Port D 0 Function Selection
0: I/O
1: SEG0

PDFS Register – HT69F40A/HT69F50A

Bit	7	6	5	4	3	2	1	0
Name	PDFS7	PDFS6	PDFS5	PDFS4	PDFS3	PDFS2	PDFS1	PDFS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

Bit 7 **PDFS7:** Port D 7 Function Selection
0: I/O
1: SEG7 or TP1B_2

Bit 6 **PDFS6:** Port D 6 Function Selection
0: I/O
1: SEG6 or TP1B_1

Bit 5 **PDFS5:** Port D 5 Function Selection
0: I/O
1: SEG5 or TP1B_0

Bit 4 **PDFS4:** Port D 4 Function Selection
0: I/O
1: SEG4 or TP1A

Bit 3 **PDFS3:** Port D 3 Function Selection
0: I/O
1: SEG3

Bit 2 **PDFS2:** Port D 2 Function Selection
0: I/O
1: SEG2

Bit 1 **PDFS1:** Port D 1 Function Selection
0: I/O
1: SEG1

Bit 0 **PDFS0:** Port D 0 Function Selection
0: I/O
1: SEG0

PEFS Register – HT69F30A/HT69F40A/HT69F50A

Bit	7	6	5	4	3	2	1	0
Name	PEFS7	PEFS6	PEFS5	PEFS4	PEFS3	PEFS2	PEFS1	PEFS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

Bit 7 **PEFS7:** Port E 7 Function Selection
0: I/O
1: SEG15

Bit 6 **PEFS6:** Port E 6 Function Selection
0: I/O
1: SEG14

Bit 5 **PEFS5:** Port E 5 Function Selection
0: I/O
1: SEG13

Bit 4 **PEFS4:** Port E 4 Function Selection
0: I/O
1: SEG12

Bit 3 **PEFS3:** Port E 3 Function Selection
0: I/O
1: SEG11

Bit 2 **PEFS2:** Port E 2 Function Selection
0: I/O
1: SEG10

Bit 1 **PEFS1:** Port E 1 Function Selection
0: I/O
1: SEG9

Bit 0 **PEFS0:** Port E 0 Function Selection
0: I/O
1: SEG8

PFFS Register – HT69F30A/HT69F40A/HT69F50A

Bit	7	6	5	4	3	2	1	0
Name	PFFS7	PFFS6	PFFS5	PFFS4	PFFS3	PFFS2	PFFS1	PFFS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

Bit 7 **PFFS7:** Port F 7 Function Selection
0: I/O
1: SEG23

Bit 6 **PFFS6:** Port F 6 Function Selection
0: I/O
1: SEG22

Bit 5 **PFFS5:** Port F 5 Function Selection
0: I/O
1: SEG21

Bit 4 **PFFS4:** Port F 4 Function Selection
0: I/O
1: SEG20

Bit 3 **PFFS3:** Port F 3 Function Selection
0: I/O
1: SEG19

Bit 2 **PFFS2:** Port F 2 Function Selection
0: I/O
1: SEG18

Bit 1 **PFFS1:** Port F 1 Function Selection
0: I/O
1: SEG17

Bit 0 **PFFS0:** Port F 0 Function Selection
0: I/O
1: SEG16

PGFS Register – HT69F40A/HT69F50A

Bit	7	6	5	4	3	2	1	0
Name	PGFS7	PGFS6	PGFS5	PGFS4	PGFS3	PGFS2	PGFS1	PGFS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

Bit 7 **PGFS7:** Port G 7 Function Selection
0: I/O
1: SEG31

Bit 6 **PGFS6:** Port G 6 Function Selection
0: I/O
1: SEG30

Bit 5 **PGFS5:** Port G 5 Function Selection
0: I/O
1: SEG29

Bit 4 **PGFS4:** Port G 4 Function Selection
0: I/O
1: SEG28

Bit 3 **PGFS3:** Port G 3 Function Selection
0: I/O
1: SEG27

Bit 2 **PGFS2:** Port G 2 Function Selection
0: I/O
1: SEG26

Bit 1 **PGFS1:** Port G 1 Function Selection
0: I/O
1: SEG25

Bit 0 **PGFS0:** Port G 0 Function Selection
0: I/O
1: SEG24

PHFS Register – HT69F50A

Bit	7	6	5	4	3	2	1	0
Name	PHFS7	PHFS6	PHFS5	PHFS4	PHFS3	PHFS2	PHFS1	PHFS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

Bit 7 **PHFS7:** Port H 7 Function Selection
0: I/O
1: SEG39

Bit 6 **PHFS6:** Port H 6 Function Selection
0: I/O
1: SEG38

Bit 5 **PHFS5:** Port H 5 Function Selection
0: I/O
1: SEG37

Bit 4 **PHFS4:** Port H 4 Function Selection
0: I/O
1: SEG36

Bit 3 **PHFS3:** Port H 3 Function Selection
0: I/O
1: SEG35

Bit 2 **PHFS2:** Port H 2 Function Selection
0: I/O
1: SEG34

Bit 1 **PHFS1:** Port H 1 Function Selection
0: I/O
1: SEG33

Bit 0 **PHFS0:** Port H 0 Function Selection
0: I/O
1: SEG32

SFS Register – HT69F30A

Bit	7	6	5	4	3	2	1	0
Name	—	SFS6	SFS5	SFS4	—	—	—	—
R/W	—	R/W	R/W	R/W	—	—	—	—
POR	—	0	0	0	—	—	—	—

Bit 7 Unimplemented, read as "0"

Bit 6 **SFS6:** TCK1 Source Selection
0: PA2
1: PD2

Bit 5 **SFS5:** TCK0 Source Selection
0: PA2
1: PD1

Bit 4 **SFS4:** INT1 Source Selection
0: PA0
1: PD0

Bit 3~0 Unimplemented, read as "0"

SFS Register – HT69F40A/HT69F50A

Bit	7	6	5	4	3	2	1	0
Name	SFS7	SFS6	SFS5	SFS4	SFS3	SFS2	SFS1	SFS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **SFS7:** TCK2 Source Selection
0: PA2
1: PD2

Bit 6 **SFS6:** TCK1 Source Selection
0: PA0
1: PD3

Bit 5 **SFS5:** TCK0 Source Selection
0: PA2
1: PD1

Bit 4 **SFS4:** INT1 Source Selection
0: PA0
1: PD0

Bit 3 **SFS3:** PD7 Special Function Selection
0: SEG7
1: TP1B_2

Bit 2 **SFS2:** PD6 Special Function Selection
0: SEG6
1: TP1B_1

Bit 1 **SFS1:** PD5 Special Function Selection
0: SEG5
1: TP1B_0

Bit 0 **SFS0:** PD4 Special Function Selection
0: SEG4
1: TP1A

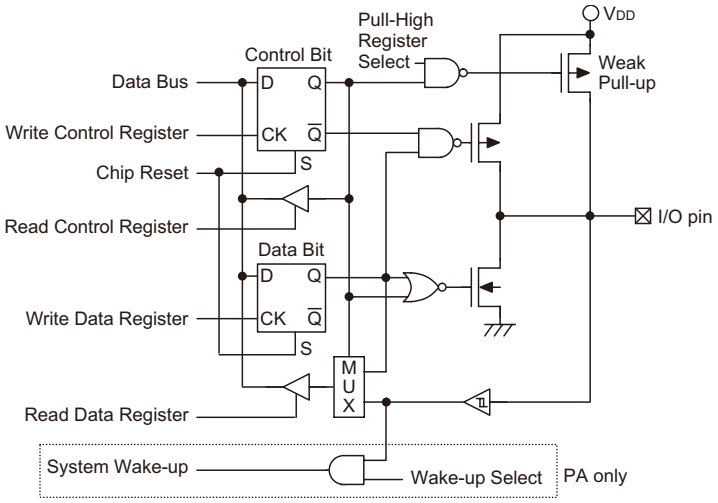
I/O Pin Structures

The accompanying diagrams illustrate the internal structures of some generic I/O pin types. As the exact logical construction of the I/O pin will differ from these drawings, they are supplied as a guide only to assist with the functional understanding of the I/O pins. The wide range of pin-shared structures does not permit all types to be shown.

Programming Considerations

Within the user program, one of the things first to consider is port initialisation. After a reset, all of the I/O data and port control registers will be set to high. This means that all I/O pins will be defaulted to an input state, the level of which depends on the other connected circuitry and whether pull-high selections have been chosen. If the port control registers are then programmed to setup some pins as outputs, these output pins will have an initial high output value unless the associated port data registers are first programmed. Selecting which pins are inputs and which are outputs can be achieved byte-wide by loading the correct values into the appropriate port control register or by programming individual bits in the port control register using the "SET [m].i" and "CLR [m].i" instructions. Note that when using these bit control instructions, a read-modify-write operation takes place. The microcontroller must first read in the data on the entire port, modify it to the required new bit values and then rewrite this data back to the output ports.

Port A has the additional capability of providing wake-up functions. When the device is in the SLEEP or IDLE Mode, various methods are available to wake the device up. One of these is a high to low transition of any of the Port A pins. Single or multiple pins on Port A can be setup to have this function.



Generic Input/Output Structure

Timer Modules – TM

One of the most fundamental functions in any microcontroller device is the ability to control and measure time. To implement time related functions each device includes several Timer Modules, abbreviated to the name TM. The TMs are multi-purpose timing units and serve to provide operations such as Timer/Counter, Input Capture, Compare Match Output and Single Pulse Output as well as being the functional unit for the generation of PWM signals. Each of the TMs has either two or three individual interrupts. The addition of input and output pins for each TM ensures that users are provided with timing units with a wide and flexible range of features.

The common features of the different TM types are described here with more detailed information provided in the individual Compact, Standard and Enhanced TM sections.

Introduction

The devices contain from two to three TMs depending upon which device is selected with each TM having a reference name of TM0, TM1, and TM2. Each individual TM can be categorised as a certain type, namely Compact Type TM, Standard Type TM or Enhanced Type TM. Although similar in nature, the different TM types vary in their feature complexity. The common features to all of the Compact, Standard and Enhanced TMs will be described in this section, the detailed operation regarding each of the TM types will be described in separate sections. The main features and differences between the three types of TMs are summarised in the accompanying table.

TM Function	CTM	STM	ETM
Timer/Counter	√	√	√
I/P Capture	—	√	√
Compare Match Output	√	√	√
PWM Channels	1	1	2
Single Pulse Output	—	1	2
PWM Alignment	Edge	Edge	Edge & Centre
PWM Adjustment Period & Duty	Duty or Period	Duty or Period	Duty or Period

TM Function Summary

Each device in the series contains a specific number of either Compact Type, Standard Type and Enhanced Type TM unit which are shown in the table together with their individual reference name, TM0~TM2.

Device	TM0	TM1	TM2
HT69F30A	10-bit CTM	10-bit STM	—
HT69F40A	10-bit CTM	10-bit ETM	10-bit STM
HT69F50A	10-bit CTM	10-bit ETM	16-bit STM

TM Name/Type Reference

TM Operation

The three different types of TM offer a diverse range of functions, from simple timing operations to PWM signal generation. The key to understanding how the TM operates is to see it in terms of a free running counter whose value is then compared with the value of pre-programmed internal comparators. When the free running counter has the same value as the pre-programmed comparator, known as a compare match situation, a TM interrupt signal will be generated which can clear the counter and perhaps also change the condition of the TM output pin. The internal TM counter is driven by a user selectable clock source, which can be an internal clock or an external pin.

TM Clock Source

The clock source which drives the main counter in each TM can originate from various sources. The selection of the required clock source is implemented using the TnCK2~TnCK0 bits in the TMn control registers. The clock source can be a ratio of either the system clock f_{SYS} or the internal high clock f_H, the f_{SUB} clock source or the external TCKn pin. Note that setting these bits to the value 101 will select a reserved clock input, in effect disconnecting the TM clock source. The TCKn pin clock source is used to allow an external signal to drive the TM as an external clock source or for event counting.

TM Interrupts

The Compact and Standard type TMs each have two internal interrupts, one for each of the internal comparator A or comparator P, which generate a TM interrupt when a compare match condition occurs. As the Enhanced type TM has three internal comparators and comparator A or comparator B or comparator P compare match functions, it consequently has three internal interrupts. When a TM interrupt is generated it can be used to clear the counter and also to change the state of the TM output pin.

TM External Pins

Each of the TMs, irrespective of what type, has one TM input pin, with the label TCKn. The TM input pin, is essentially a clock source for the TM and is selected using the TnCK2~TnCK0 bits in the TMnC0 register. This external TM input pin allows an external clock source to drive the internal TM. This external TM input pin is shared with other functions but will be connected to the internal TM if selected using the TnCK2~TnCK0 bits. The TM input pin can be chosen to have either a rising or falling active edge.

The TMs each have one or more output pins with the label TPn. When the TM is in the Compare Match Output Mode, these pins can be controlled by the TM to switch to a high or low level or to toggle when a compare match situation occurs. The external TPn output pin is also the pin where the TM generates the PWM output waveform. As the TM output pins are pin-shared with other function, the TM output function must first be setup using registers. A single bit in one of the registers determines if its associated pin is to be used as an external TM output pin or if it is to have another function. The number of output pins for each TM type and device is different, the details are provided in the accompanying table.

All TM output pin names have a "_n" suffix. Pin names that include a "_1" or "_2" suffix indicate that they are from a TM with multiple output pins. This allows the TM to generate a complimentary output pair, selected using the I/O register data bits.

Device	CTM	STM	ETM	Registers
HT69F30A	TP0_0,TP0_1	TP1_0,TP1_1	—	PAFS
HT69F40A	TP0_0,TP0_1	TP2_0,TP2_1	TP1A, TP1B_0, TP1B_1, TP1B_2	PAFS, PBFS, PDFS
HT69F50A	TP0_0,TP0_1	TP2_0,TP2_1	TP1A, TP1B_0, TP1B_1, TP1B_2	PAFS, PBFS, PDFS

TM Output Pins

TM Input/Output Pin Control Registers

Selecting to have a TM input/output or whether to retain its other shared function, is implemented using one or two registers, with a single bit in each register corresponding to a TM input/output pin. Setting the bit high will setup the corresponding pin as a TM input/output, if reset to zero the pin will retain its original other function.

TM Input/Output Pin Control Register List

- **HT69F30A**

Register Name	Bit							
	7	6	5	4	3	2	1	0
PAFS	—	PAFS6	PAFS5	—	PAFS3	—	PAFS1	—
SFS	—	SFS6	SFS5	SFS4	—	—	—	—

- **HT69F40A/HT69F50A**

Register Name	Bit							
	7	6	5	4	3	2	1	0
PAFS	—	PAFS6	PAFS5	—	PAFS3	—	PAFS1	—
PBFS	PBFS7	PBFS6	PBFS5	PBFS4	—	—	—	—
PDFS	PDFS7	PDFS6	PDFS5	PDFS4	PDFS3	PDFS2	PDFS1	PDFS0
SFS	SFS7	SFS6	SFS5	SFS4	SFS3	SFS2	SFS1	SFS0

PAFS Register – HT69F30A

Bit	7	6	5	4	3	2	1	0
Name	—	PAFS6	PAFS5	—	PAFS3	—	PAFS1	—
R/W	—	R/W	R/W	—	R/W	—	R/W	—
POR	—	0	0	—	0	—	0	—

Bit 7 Unimplemented, read as "0"

Bit 6 **PAFS6:** Port A 6 Function Selection
0: I/O
1: TP0_0

Bit 5 **PAFS5:** Port A 5 Function Selection
0: I/O
1: TP1_1

Bit 4 Unimplemented, read as "0"

Bit 3 **PAFS3:** Port A 3 Function Selection
0: I/O
1: TP1_0

Bit 2 Unimplemented, read as "0"

Bit 1 **PAFS1:** Port A 1 Function Selection
0: I/O
1: TP0_1

Bit 0 Unimplemented, read as "0"

PAFS Register – HT69F40A/HT69F50A

Bit	7	6	5	4	3	2	1	0
Name	—	PAFS6	PAFS5	—	PAFS3	—	PAFS1	—
R/W	—	R/W	R/W	—	R/W	—	R/W	—
POR	—	0	0	—	0	—	0	—

Bit 7 Unimplemented, read as "0"

Bit 6 **PAFS6:** Port A 6 Function Selection
0: I/O
1: TP0_0

Bit 5 **PAFS5:** Port A 5 Function Selection
0: I/O
1: TP2_1

Bit 4 Unimplemented, read as "0"

Bit 3 **PAFS3:** Port A 3 Function Selection
0: I/O
1: TP2_0

Bit 2 Unimplemented, read as "0"

Bit 1 **PAFS1:** Port A 1 Function Selection
0: I/O
1: TP0_1

Bit 0 Unimplemented, read as "0"

PBFS Register – HT69F40A/HT69F50A

Bit	7	6	5	4	3	2	1	0
Name	PBFS7	PBFS6	PBFS5	PBFS4	—	—	—	—
R/W	R/W	R/W	R/W	R/W	—	—	—	—
POR	0	0	0	0	—	—	—	—

Bit 7 **PBFS7:** Port B 7 Function Selection
0: I/O
1: TP1B_2

Bit 6 **PBFS6:** Port B 6 Function Selection
0: I/O
1: TP1B_1

Bit 5 **PBFS5:** Port B 5 Function Selection
0: I/O
1: TP1B_0

Bit 4 **PBFS4:** Port B 4 Function Selection
0: I/O
1: TP1A

Bit 3~0 Unimplemented, read as "0"

PDFS Register – HT69F40A/HT69F50A

Bit	7	6	5	4	3	2	1	0
Name	PDFS7	PDFS6	PDFS5	PDFS4	PDFS3	PDFS2	PDFS1	PDFS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

Bit 7 **PDFS7:** Port D 7 Function Selection
0: I/O
1: SEG7 or TP1B_2

Bit 6 **PDFS6:** Port D 6 Function Selection
0: I/O
1: SEG6 or TP1B_1

Bit 5 **PDFS5:** Port D 5 Function Selection
0: I/O
1: SEG5 or TP1B_0

Bit 4 **PDFS4:** Port D 4 Function Selection
0: I/O
1: SEG4 or TP1A

Bit 3~0 **PDFS3~PDFS0:** Port D 3~0 Function Selection
Described elsewhere.

SFS Register – HT69F30A

Bit	7	6	5	4	3	2	1	0
Name	—	SFS6	SFS5	SFS4	—	—	—	—
R/W	—	R/W	R/W	R/W	—	—	—	—
POR	—	0	0	0	—	—	—	—

Bit 7 Unimplemented, read as "0"

Bit 6 **SFS6:** TCK1 Source Selection
0: PA2
1: PD2

Bit 5 **SFS5:** TCK0 Source Selection
0: PA2
1: PD1

Bit 4 **SFS4:** INT1 Source Selection
Described elsewhere.

Bit 3~0 Unimplemented, read as "0"

SFS Register – HT69F40A/HT69F50A

Bit	7	6	5	4	3	2	1	0
Name	SFS7	SFS6	SFS5	SFS4	SFS3	SFS2	SFS1	SFS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **SFS7:** TCK2 Source Selection
0: PA2
1: PD2

Bit 6 **SFS6:** TCK1 Source Selection
0: PA0
1: PD3

Bit 5 **SFS5:** TCK0 Source Selection
0: PA2
1: PD1

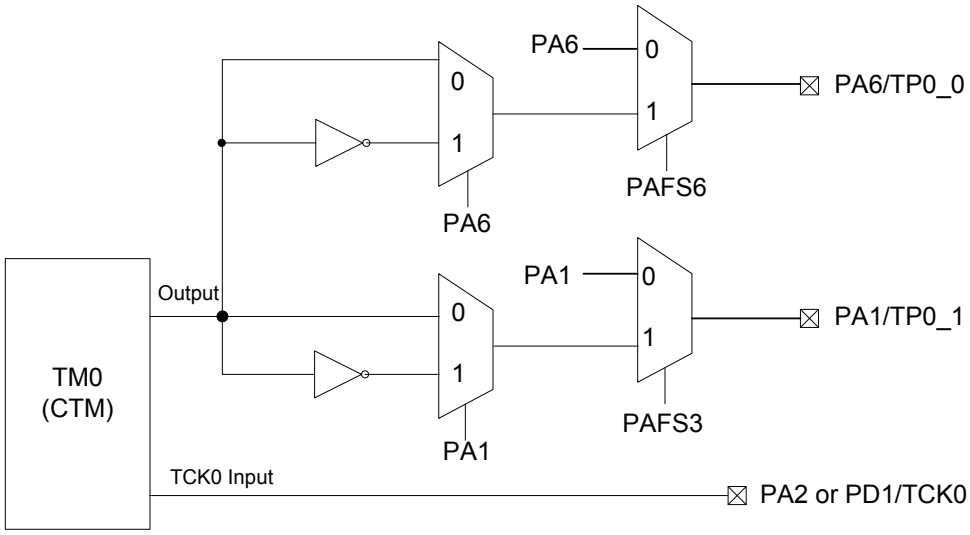
Bit 4 **SFS4:** INT1 Source Selection
Described elsewhere.

Bit 3 **SFS3:** PD7 Special Function Selection
0: SEG7
1: TP1B_2

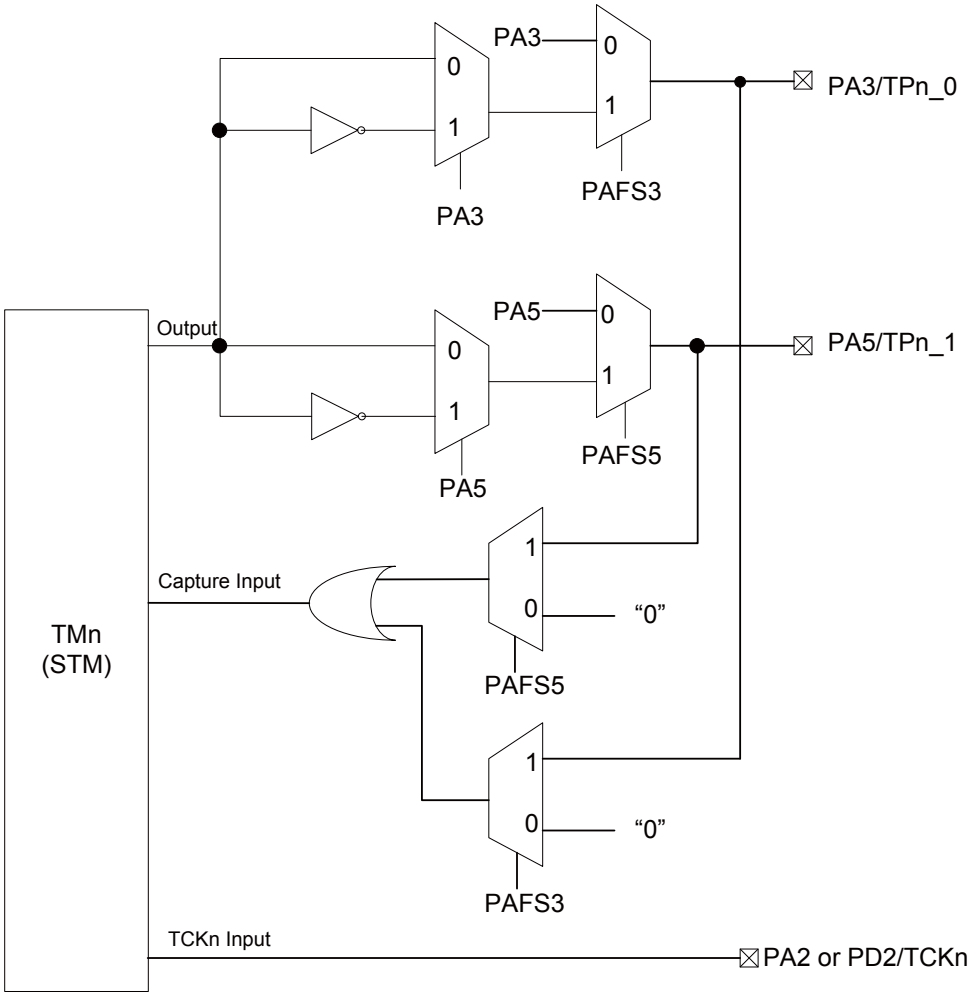
Bit 2 **SFS2:** PD6 Special Function Selection
0: SEG6
1: TP1B_1

Bit 1 **SFS1:** PD5 Special Function Selection
0: SEG5
1: TP1B_0

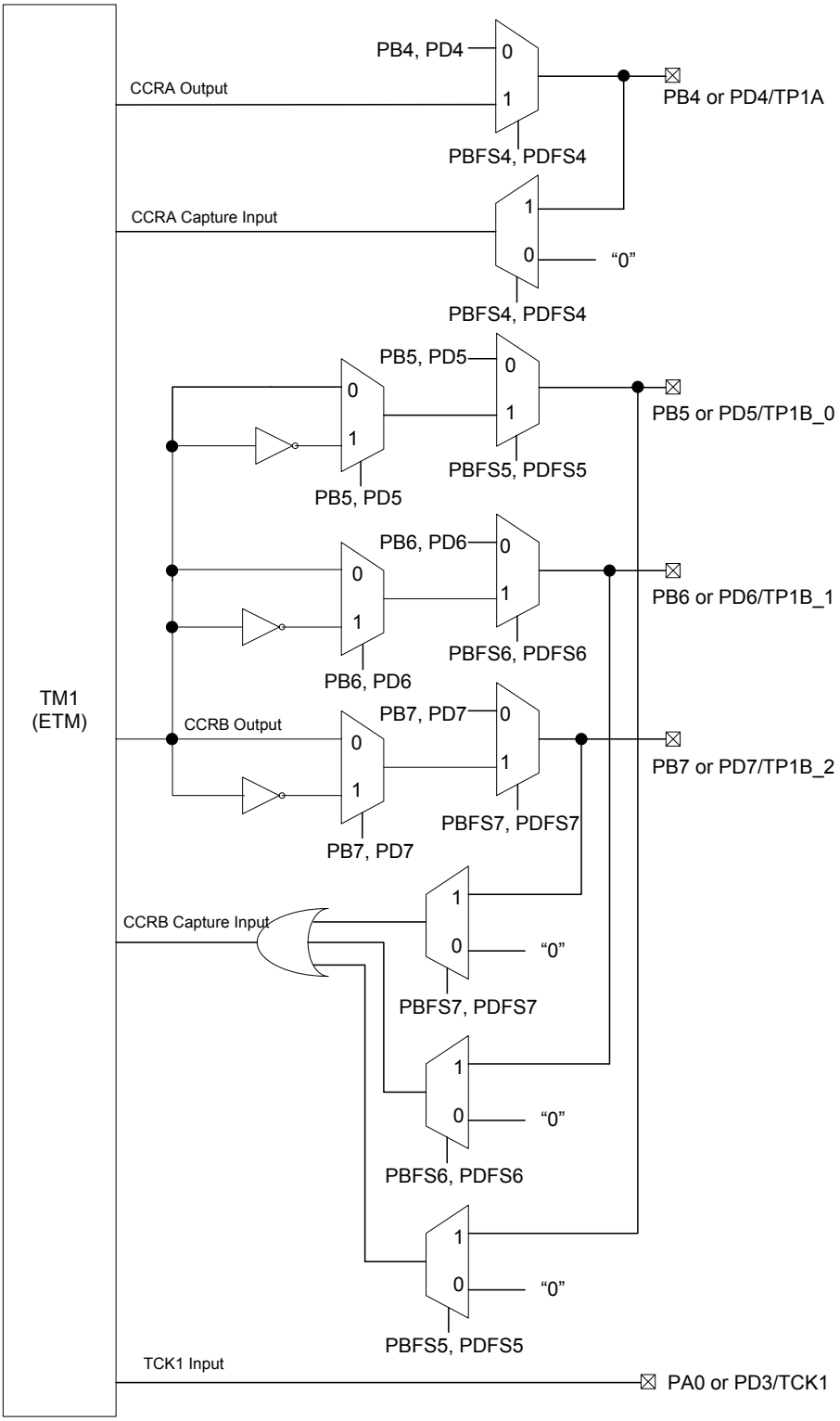
Bit 0 **SFS0:** PD4 Special Function Selection
0: SEG4
1: TP1A



CTM Function Pin Control Block Diagram – HT69F30A/HT69F40A/HT69F50A



STM Function Pin Control Block Diagram – HT69F30A (n=1); HT69F40A/HT69F50A (n=2)



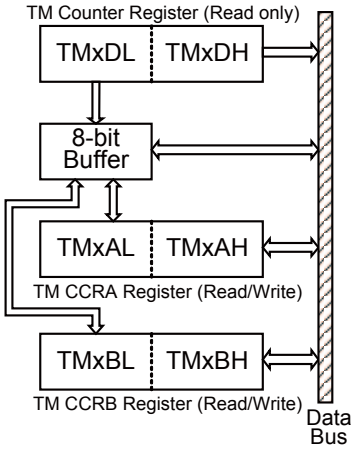
ETM Function Pin Control Block Diagram – HT69F40A/HT69F50A

Note: The I/O register data bits shown are used for TM output inversion control. In the Capture Input Mode, the TM pin control register must never enable more than one TM input.

Programming Considerations

The TM Counter Registers and the Capture/Compare CCRA and CCRB registers, being either 10-bit or 16-bit, all have a low and high byte structure. The high bytes can be directly accessed, but as the low bytes can only be accessed via an internal 8-bit buffer, reading or writing to these register pairs must be carried out in a specific way. The important point to note is that data transfer to and from the 8-bit buffer and its related low byte only takes place when a write or read operation to its corresponding high byte is executed.

As the CCRA and CCRB registers are implemented in the way shown in the following diagram and accessing these register pairs is carried out in a specific way described above, it is recommended to use the "MOV" instruction to access the CCRA and CCRB low byte registers, named TMxAL and TMxBL, using the following access procedures. Accessing the CCRA or CCRB low byte registers without following these access procedures will result in unpredictable values.



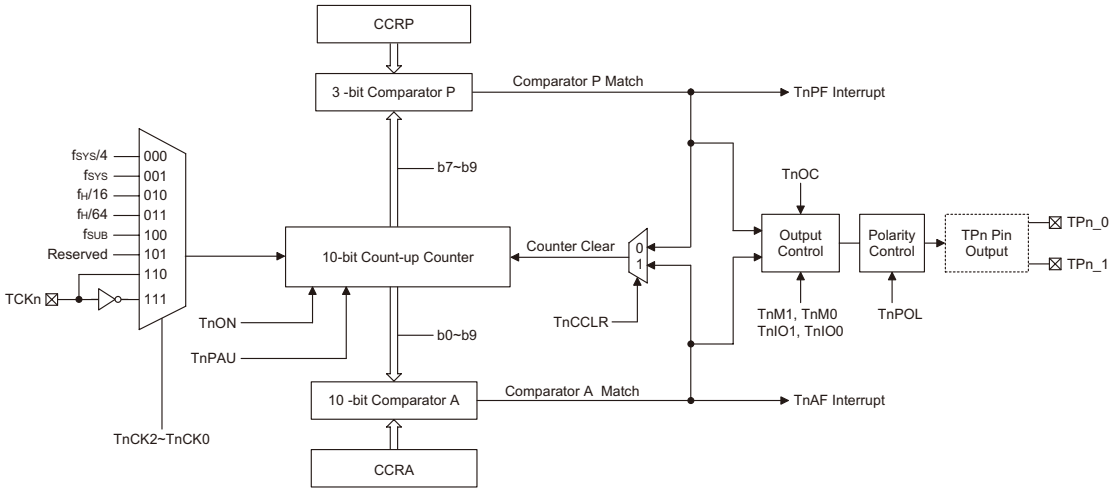
The following steps show the read and write procedures:

- Writing Data to CCRB or CCRA
  - Step 1. Write data to Low Byte TMxAL or TMxBL
    - Note that here data is only written to the 8-bit buffer.
  - Step 2. Write data to High Byte TMxAH or TMxBH
    - Here data is written directly to the high byte registers and simultaneously data is latched from the 8-bit buffer to the Low Byte registers.
- Reading Data from the Counter Registers and CCRB or CCRA
  - Step 1. Read data from the High Byte TMxDH, TMxAH or TMxBH
    - Here data is read directly from the High Byte registers and simultaneously data is latched from the Low Byte register into the 8-bit buffer.
  - Step 2. Read data from the Low Byte TMxDL, TMxAL or TMxBL
    - This step reads data from the 8-bit buffer.

Compact Type TM – CTM

Although the simplest form of the three TM types, the Compact TM type still contains three operating modes, which are Compare Match Output, Timer/Event Counter and PWM Output modes. The Compact TM can also be controlled with an external input pin and can drive one or two external output pins. These two external output pins can be the same signal or the inverse signal.

Device	TM Type	TM Name.	TM Input Pin	TM Output Pin
HT69F30A HT69F40A HT69F50A	10-bit CTM	TM0	TCK0	TP0_0, TP0_1



Compact Type TM Block Diagram (n=0)

Compact TM Operation

At its core is a 10-bit count-up counter which is driven by a user selectable internal or external clock source. There are also two internal comparators with the names, Comparator A and Comparator P. These comparators will compare the value in the counter with CCRP and CCRA registers. The CCRP is three bits wide whose value is compared with the highest three bits in the counter while the CCRA is the ten bits and therefore compares with all counter bits.

The only way of changing the value of the 10-bit counter using the application program, is to clear the counter by changing the TnON bit from low to high. The counter will also be cleared automatically by a counter overflow or a compare match with one of its associated comparators. When these conditions occur, a TM interrupt signal will also usually be generated. The Compact Type TM can operate in a number of different operational modes, can be driven by different clock sources including an input pin and can also control an output pin. All operating setup conditions are selected using relevant internal registers.

Compact Type TM Register Description

Overall operation of the Compact TM is controlled using six registers. A read only register pair exists to store the internal counter 10-bit value, while a read/write register pair exists to store the internal 10-bit CCRA value. The remaining two registers are control registers which setup the different operating and control modes as well as the three CCRP bits.

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMnC0	TnPAU	TnCK2	TnCK1	TnCK0	TnON	TnRP2	TnRP1	TnRP0
TMnC1	TnM1	TnM0	TnIO1	TnIO0	TnOC	TnPOL	TnDPX	TnCCLR
TMnDL	D7	D6	D5	D4	D3	D2	D1	D0
TMnDH	—	—	—	—	—	—	D9	D8
TMnAL	D7	D6	D5	D4	D3	D2	D1	D0
TMnAH	—	—	—	—	—	—	D9	D8

Compact TM Register List (n=0)

TMnDL Register

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~0 **TMnDL**: TMn Counter Low Byte Register bit 7 ~ bit 0
TMn 10-bit Counter bit 7 ~ bit 0

TMnDH Register

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R	R
POR	—	—	—	—	—	—	0	0

Bit 7~2 Unimplemented, read as "0"

Bit 1~0 **TMnDH**: TMn Counter High Byte Register bit 1 ~ bit 0
TMn 10-bit Counter bit 9 ~ bit 8

TMnAL Register

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **TMnAL**: TMn CCRA Low Byte Register bit 7 ~ bit 0
TMn 10-bit CCRA bit 7 ~ bit 0

TMnAH Register

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 Unimplemented, read as "0"

Bit 1~0 **TMnAH**: TMn CCRA High Byte Register bit 1 ~ bit 0
TMn 10-bit CCRA bit 9 ~ bit 8

TMnC0 Register

Bit	7	6	5	4	3	2	1	0
Name	TnPAU	TnCK2	TnCK1	TnCK0	TnON	TnRP2	TnRP1	TnRP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **TnPAU:** TMn Counter Pause Control
0: Run
1: Pause
The counter can be paused by setting this bit high. Clearing the bit to zero restores normal counter operation. When in a Pause condition the TM will remain powered up and continue to consume power. The counter will retain its residual value when this bit changes from low to high and resume counting from this value when the bit changes to a low value again.

Bit 6~4 **TnCK2~TnCK0:** Select TMn Counter clock
000: $f_{SYS}/4$
001: f_{SYS}
010: $f_H/16$
011: $f_H/64$
100: f_{SUB}
101: Reserved
110: TCKn rising edge clock
111: TCKn falling edge clock
These three bits are used to select the clock source for the TM. Selecting the Reserved clock input will effectively disable the internal counter. The external pin clock source can be chosen to be active on the rising or falling edge. The clock source f_{SYS} is the system clock, while f_H and f_{SUB} are other internal clocks, the details of which can be found in the oscillator section.

Bit 3 **TnON:** TMn Counter On/Off Control
0: Off
1: On
This bit controls the overall on/off function of the TM. Setting the bit high enables the counter to run, clearing the bit disables the TM. Clearing this bit to zero will stop the counter from counting and turn off the TM which will reduce its power consumption. When the bit changes state from low to high the internal counter value will be reset to zero, however when the bit changes from high to low, the internal counter will retain its residual value. If the TM is in the Compare Match Output Mode then the TM output pin will be reset to its initial condition, as specified by the TnOC bit, when the TnON bit changes from low to high.

Bit 2~0 **TnRP2~TnRP0:** TMn CCRP 3-bit register, compared with the TMn Counter bit 9~bit 7 Comparator P Match Period
000: 1024 TMn clocks
001: 128 TMn clocks
010: 256 TMn clocks
011: 384 TMn clocks
100: 512 TMn clocks
101: 640 TMn clocks
110: 768 TMn clocks
111: 896 TMn clocks

These three bits are used to setup the value on the internal CCRP 3-bit register, which are then compared with the internal counter's highest three bits. The result of this comparison can be selected to clear the internal counter if the TnCCLR bit is set to zero. Setting the TnCCLR bit to zero ensures that a compare match with the CCRP values will reset the internal counter. As the CCRP bits are only compared with the highest three counter bits, the compare values exist in 128 clock cycle multiples. Clearing all three bits to zero is in effect allowing the counter to overflow at its maximum value.

TMnC1 Register

Bit	7	6	5	4	3	2	1	0
Name	TnM1	TnM0	TnIO1	TnIO0	TnOC	TnPOL	TnDPX	TnCCLR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **TnM1~TnM0**: Select TMn Operating Mode
00: Compare Match Output Mode
01: Undefined
10: PWM Mode
11: Timer/Counter Mode

These bits setup the required operating mode for the TM. To ensure reliable operation the TM should be switched off before any changes are made to the TnM1 and TnM0 bits. In the Timer/Counter Mode, the TM output pin control must be disabled.

Bit 5~4 **TnIO1~TnIO0**: Select TPn_0, TPn_1 output function
Compare Match Output Mode
00: No change
01: Output low
10: Output high
11: Toggle output
PWM Mode
00: PWM Output inactive state
01: PWM Output active state
10: PWM output
11: Undefined
Timer/counter Mode
unused

These two bits are used to determine how the TM output pin changes state when a certain condition is reached. The function that these bits select depends upon in which mode the TM is running.

In the Compare Match Output Mode, the TnIO1 and TnIO0 bits determine how the TM output pin changes state when a compare match occurs from the Comparator A. The TM output pin can be setup to switch high, switch low or to toggle its present state when a compare match occurs from the Comparator A. When the bits are both zero, then no change will take place on the output. The initial value of the TM output pin should be setup using the TnOC bit in the TMnC1 register. Note that the output level requested by the TnIO1 and TnIO0 bits must be different from the initial value setup using the TnOC bit otherwise no change will occur on the TM output pin when a compare match occurs. After the TM output pin changes state it can be reset to its initial level by changing the level of the TnON bit from low to high.

In the PWM Mode, the TnIO1 and TnIO0 bits determine how the TM output pin changes state when a certain compare match condition occurs. The PWM output function is modified by changing these two bits. It is necessary to only change the values of the TnIO1 and TnIO0 bits only after the TMn has been switched off. Unpredictable PWM outputs will occur if the TnIO1 and TnIO0 bits are changed when the TM is running.

Bit 3 **TnOC**: TPn_0, TPn_1 Output control bit

Compare Match Output Mode
0: Initial low
1: Initial high

PWM Mode
0: Active low
1: Active high

This is the output control bit for the TM output pin. Its operation depends upon whether TM is being used in the Compare Match Output Mode or in the PWM Mode. It has no effect if the TM is in the Timer/Counter Mode. In the Compare Match Output Mode it determines the logic level of the TM output pin before a compare match occurs. In the PWM Mode it determines if the PWM signal is active high or active low.

Bit 2 **TnPOL**: TPn_0, TPn_1 Output polarity Control
0: Non-invert
1: Invert

This bit controls the polarity of the TPn_0 or TPn_1 output pin. When the bit is set high the TM output pin will be inverted and not inverted when the bit is zero. It has no effect if the TM is in the Timer/Counter Mode.

Bit 1 **TnDPX**: TMn PWM period/duty Control
0: CCRP - period; CCRA - duty
1: CCRP - duty; CCRA - period

This bit, determines which of the CCRA and CCRP registers are used for period and duty control of the PWM waveform.

Bit 0 **TnCCLR**: Select TMn Counter clear condition
0: TMn Comparatror P match
1: TMn Comparatror A match

This bit is used to select the method which clears the counter. Remember that the Compact TM contains two comparators, Comparator A and Comparator P, either of which can be selected to clear the internal counter. With the TnCCLR bit set high, the counter will be cleared when a compare match occurs from the Comparator A. When the bit is low, the counter will be cleared when a compare match occurs from the Comparator P or with a counter overflow. A counter overflow clearing method can only be implemented if the CCRP bits are all cleared to zero. The TnCCLR bit is not used in the PWM Mode.

Compact Type TM Operating Modes

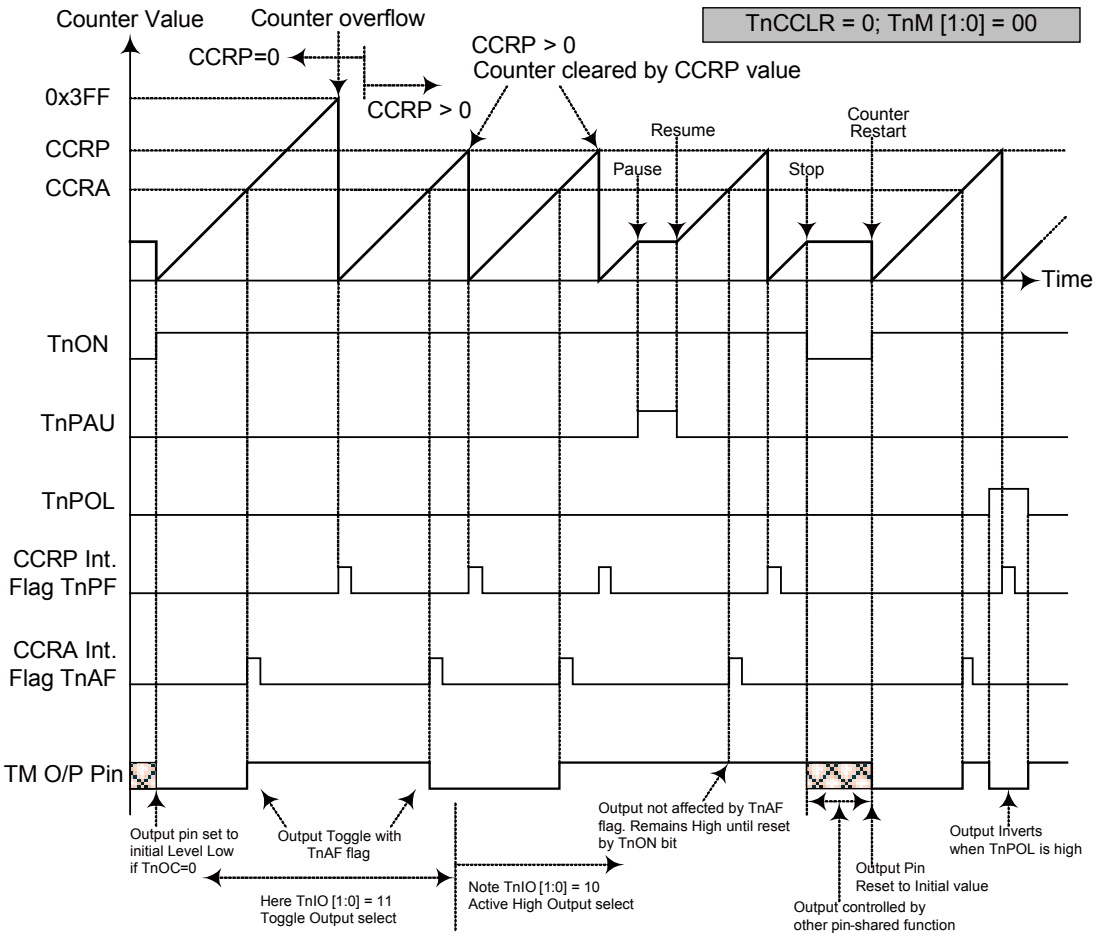
The Compact Type TM can operate in one of three operating modes, Compare Match Output Mode, PWM Mode or Timer/Counter Mode. The operating mode is selected using the TnM1 and TnM0 bits in the TMnC1 register.

Compare Match Output Mode

To select this mode, bits TnM1 and TnM0 in the TMnC1 register, should be set to 00 respectively. In this mode once the counter is enabled and running it can be cleared by three methods. These are a counter overflow, a compare match from Comparator A and a compare match from Comparator P. When the TnCCLR bit is low, there are two ways in which the counter can be cleared. One is when a compare match occurs from Comparator P, the other is when the CCRP bits are all zero which allows the counter to overflow. Here both TnAF and TnPF interrupt request flags for the Comparator A and Comparator P respectively, will both be generated.

If the TnCCLR bit in the TMnC1 register is high then the counter will be cleared when a compare match occurs from Comparator A. However, here only the TnAF interrupt request flag will be generated even if the value of the CCRP bits is less than that of the CCRA registers. Therefore when TnCCLR is high no TnPF interrupt request flag will be generated. If the CCRA bits are all zero, the counter will overflow when its reaches its maximum 10-bit, 3FF Hex, value, however here the TnAF interrupt request flag will not be generated.

As the name of the mode suggests, after a comparison is made, the TM output pin will change state. The TM output pin condition however only changes state when an TnAF interrupt request flag is generated after a compare match occurs from Comparator A. The TnPF interrupt request flag, generated from a compare match occurs from Comparator P, will have no effect on the TM output pin. The way in which the TM output pin changes state are determined by the condition of the TnIO1 and TnIO0 bits in the TMnC1 register. The TM output pin can be selected using the TnIO1 and TnIO0 bits to go high, to go low or to toggle from its present condition when a compare match occurs from Comparator A. The initial condition of the TM output pin, which is setup after the TnON bit changes from low to high, is setup using the TnOC bit. Note that if the TnIO1 and TnIO0 bits are zero then no pin change will take place.



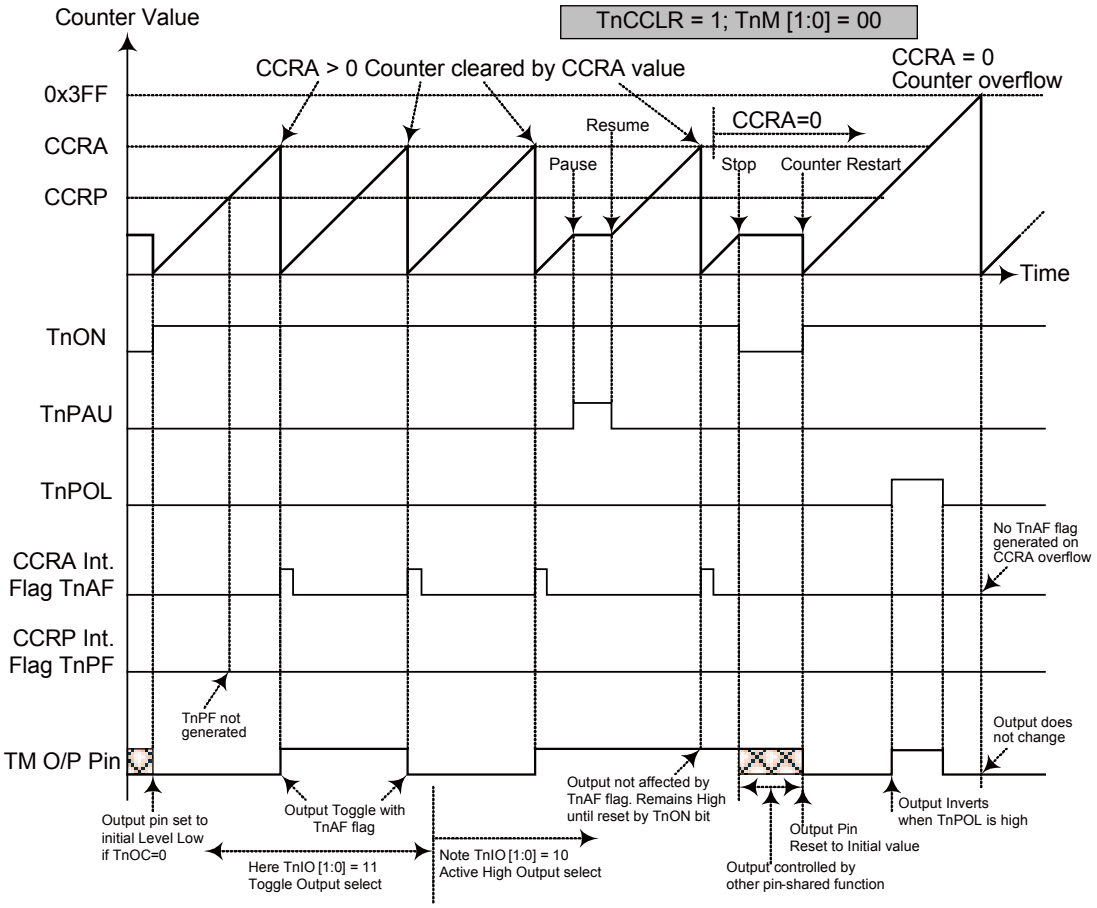
Compare Match Output Mode – TnCCLR=0

Note: 1. With TnCCLR=0, a Comparator P match will clear the counter

2. The TM output pin is controlled only by the TnAF flag

3. The output pin is reset to its initial state by a TnON bit rising edge

4. n=0



Compare Match Output Mode – TnCCLR=1

Note: 1. With TnCCLR=1, a Comparator A match will clear the counter

2. The TM output pin is controlled only by the TnAF flag
3. The output pin is reset to its initial state by a TnON bit rising edge
4. The TnPF flag is not generated when TnCCLR=1
5. n=0

Timer/Counter Mode

To select this mode, bits TnM1 and TnM0 in the TMnC1 register should be set to 11 respectively. The Timer/Counter Mode operates in an identical way to the Compare Match Output Mode generating the same interrupt flags. The exception is that in the Timer/Counter Mode the TM output pin is not used. Therefore the above description and Timing Diagrams for the Compare Match Output Mode can be used to understand its function. As the TM output pin is not used in this mode, the pin can be used as a normal I/O pin or other pin-shared function.

PWM Output Mode

To select this mode, bits TnM1 and TnM0 in the TMnC1 register should be set to 10 respectively. The PWM function within the TM is useful for applications which require functions such as motor control, heating control, illumination control etc. By providing a signal of fixed frequency but of varying duty cycle on the TM output pin, a square wave AC waveform can be generated with varying equivalent DC RMS values.

As both the period and duty cycle of the PWM waveform can be controlled, the choice of generated waveform is extremely flexible. In the PWM mode, the TnCCLR bit has no effect on the PWM operation. Both of the CCRA and CCRP registers are used to generate the PWM waveform, one register is used to clear the internal counter and thus control the PWM waveform frequency, while the other one is used to control the duty cycle. Which register is used to control either frequency or duty cycle is determined using the TnDPX bit in the TMnC1 register. The PWM waveform frequency and duty cycle can therefore be controlled by the values in the CCRA and CCRP registers.

An interrupt flag, one for each of the CCRA and CCRP, will be generated when a compare match occurs from either Comparator A or Comparator P. The TnOC bit in the TMnC1 register is used to select the required polarity of the PWM waveform while the two TnIO1 and TnIO0 bits are used to enable the PWM output or to force the TM output pin to a fixed high or low level. The TnPOL bit is used to reverse the polarity of the PWM output waveform.

CTM, PWM Mode, Edge-aligned Mode, T0DPX=0

CCRP	001b	010b	011b	100b	101b	110b	111b	000b
Period	128	256	384	512	640	768	896	1024
Duty	CCRA							

If f_{SYS}=16MHz, TM clock source is $f_{SYS}/4$, CCRP=100b and CCRA=128,

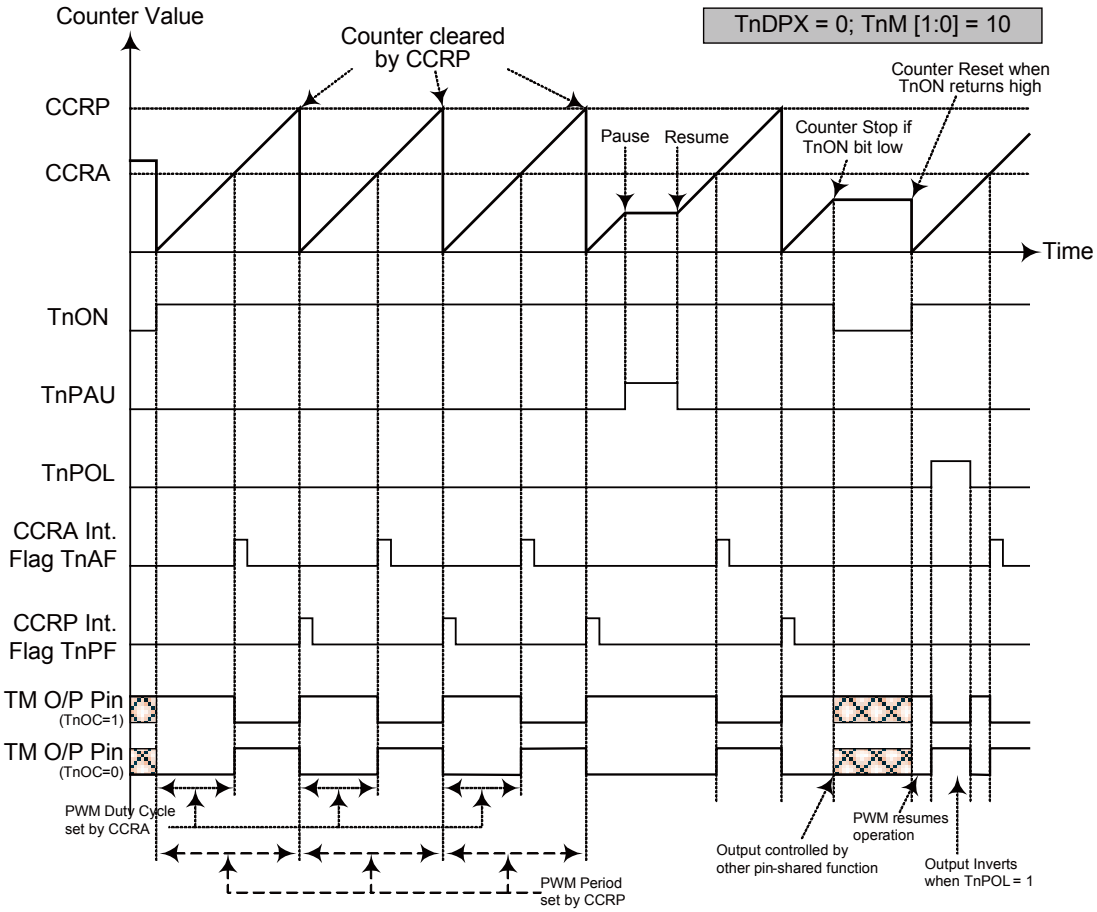
The CTM PWM output frequency=($f_{SYS}/4$)/512=$f_{SYS}/2048$=7.8125kHz, duty=128/512=25%.

If the Duty value defined by the CCRA register is equal to or greater than the Period value, then the PWM output duty is 100%.

CTM, PWM Mode, Edge-aligned Mode, T0DPX=1

CCRP	001b	010b	011b	100b	101b	110b	111b	000b
Period	CCRA							
Duty	128	256	384	512	640	768	896	1024

The PWM output period is determined by the CCRA register value together with the TM clock while the PWM duty cycle is defined by the CCRP register value.



PWM Mode – TnDPX=0

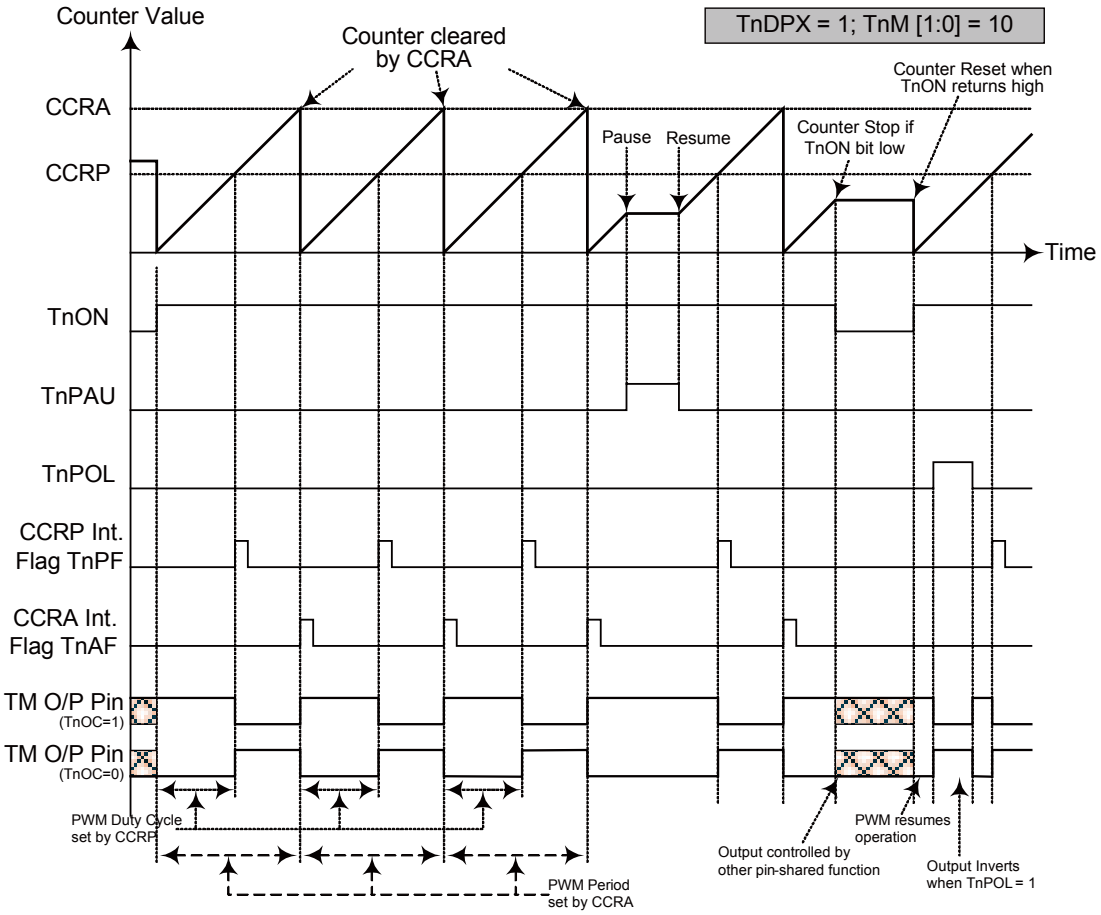
Note: 1. Here TnDPX=0 -- Counter cleared by CCRP

2. A counter clear sets the PWM Period

3. The internal PWM function continues running even when TnIO [1:0]=00 or 01

4. The TnCCLR bit has no influence on PWM operation

5. n=0



PWM Mode – TnDPX=1

Note: 1. Here TnDPX=1 – Counter cleared by CCRA

2. A counter clear sets the PWM Period

3. The internal PWM function continues running even when TnIO [1:0]=00 or 01

4. The TnCCLR bit has no influence on PWM operation

5. n=0

Standard Type TM – STM

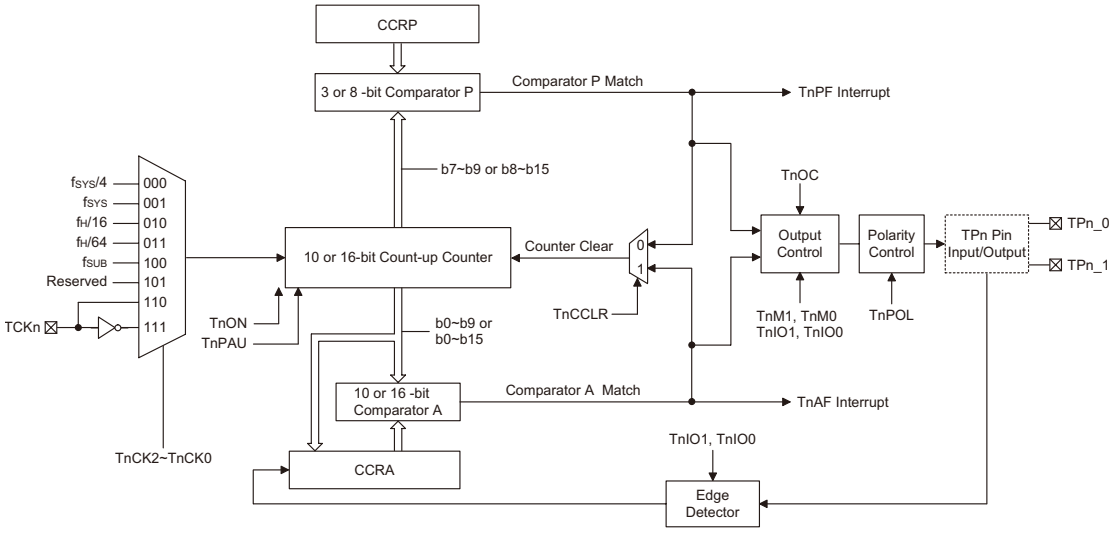
The Standard Type TM contains five operating modes, which are Compare Match Output, Timer/Event Counter, Capture Input, Single Pulse Output and PWM Output modes. The Standard TM can also be controlled with an external input pin and can drive one or two external output pins.

Device	TM Type	TM Name.	TM Input Pin	TM Output Pin
HT69F30A	10-bit STM	TM1	TCK1	TP1_0, TP1_1
HT69F40A	10-bit STM	TM2	TCK2	TP2_0, TP2_1
HT69F50A	16-bit STM	TM2	TCK2	TP2_0, TP2_1

Standard TM Operation

There are two sizes of Standard TMs, one is 10-bits wide and the other is 16-bits wide. At the core is a 10 or 16-bit count-up counter which is driven by a user selectable internal or external clock source. There are also two internal comparators with the names, Comparator A and Comparator P. These comparators will compare the value in the counter with CCRP and CCRA registers. The CCRP comparator is 3 or 8-bits wide whose value is compared the with highest 3 or 8 bits in the counter while the CCRA is the ten or sixteen bits and therefore compares all counter bits.

The only way of changing the value of the 10 or 16-bit counter using the application program, is to clear the counter by changing the TnON bit from low to high. The counter will also be cleared automatically by a counter overflow or a compare match with one of its associated comparators. When these conditions occur, a TM interrupt signal will also usually be generated. The Standard Type TM can operate in a number of different operational modes, can be driven by different clock sources including an input pin and can also control an output pin. All operating setup conditions are selected using relevant internal registers.



Standard Type TM Block Diagram (n=1 or 2)

Standard Type TM Register Description

Overall operation of the Standard TM is controlled using a series of registers. A read only register pair exists to store the internal counter 10 or 16-bit value, while a read/write register pair exists to store the internal 10 or 16-bit CCRA value. The remaining two registers are control registers which setup the different operating and control modes as well as the three or eight CCRP bits.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1C0	T1PAU	T1CK2	T1CK1	T1CK0	T1ON	T1RP2	T1RP1	T1RP0
TM1C1	T1M1	T1M0	T1IO1	T1IO0	T1OC	T1POL	T1DPX	T1CCLR
TM1DL	D7	D6	D5	D4	D3	D2	D1	D0
TM1DH	—	—	—	—	—	—	D9	D8
TM1AL	D7	D6	D5	D4	D3	D2	D1	D0
TM1AH	—	—	—	—	—	—	D9	D8

10-bit Standard TM Register List – HT69F30A

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM2C0	T2PAU	T2CK2	T2CK1	T2CK0	T2ON	T2RP2	T2RP1	T2RP0
TM2C1	T2M1	T2M0	T2IO1	T2IO0	T2OC	T2POL	T2DPX	T2CCLR
TM2DL	D7	D6	D5	D4	D3	D2	D1	D0
TM2DH	—	—	—	—	—	—	D9	D8
TM2AL	D7	D6	D5	D4	D3	D2	D1	D0
TM2AH	—	—	—	—	—	—	D9	D8

10-bit Standard TM Register List – HT69F40A

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM2C0	T2PAU	T2CK2	T2CK1	T2CK0	T2ON	—	—	—
TM2C1	T2M1	T2M0	T2IO1	T2IO0	T2OC	T2POL	T2DPX	T2CCLR
TM2DL	D7	D6	D5	D4	D3	D2	D1	D0
TM2DH	D15	D14	D13	D12	D11	D10	D9	D8
TM2AL	D7	D6	D5	D4	D3	D2	D1	D0
TM2AH	D15	D14	D13	D12	D11	D10	D9	D8
TM2RP	D7	D6	D5	D4	D3	D2	D1	D0

16-bit Standard TM Register List – HT69F50A

10-bit STM Register Definitions – n=1 for HT69F30A and n=2 for HT69F40A

- **TMnC0 Register**

Bit	7	6	5	4	3	2	1	0
Name	TnPAU	TnCK2	TnCK1	TnCK0	TnON	TnRP2	TnRP1	TnRP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **TnPAU:** TMn Counter Pause Control
0: Run
1: Pause
The counter can be paused by setting this bit high. Clearing the bit to zero restores normal counter operation. When in a Pause condition the TM will remain powered up and continue to consume power. The counter will retain its residual value when this bit changes from low to high and resume counting from this value when the bit changes to a low value again.

Bit 6~4 **TnCK2, TnCK1, TnCK0:** Select TMn Counter clock
000: $f_{SYS}/4$
001: f_{SYS}
010: $f_H/16$
011: $f_H/64$
100: f_{SUB}
101: Reserved
110: TCKn rising edge clock
111: TCKn falling edge clock
These three bits are used to select the clock source for the TMn. Selecting the Reserved clock input will effectively disable the internal counter. The external pin clock source can be chosen to be active on the rising or falling edge. The clock source f_{SYS} is the system clock, while f_H and f_{SUB} are other internal clocks, the details of which can be found in the oscillator section.

Bit 3 **TnON:** TMn Counter On/Off Control
0: Off
1: On
This bit controls the overall on/off function of the TM. Setting the bit high enables the counter to run, clearing the bit disables the TM. Clearing this bit to zero will stop the counter from counting and turn off the TM which will reduce its power consumption. When the bit changes state from low to high the internal counter value will be reset to zero, however when the bit changes from high to low, the internal counter will retain its residual value until the bit returns high again. If the TM is in the Compare Match Output Mode then the TM output pin will be reset to its initial condition, as specified by the TnOC bit, when the TnON bit changes from low to high.

Bit 2~0 **TnRP2~TnRP0:** TMn CCRP 3-bit register, compared with the TMn Counter bit 9~bit 7 Comparator P Match Period
000: 1024 TMn clocks
001: 128 TMn clocks
010: 256 TMn clocks
011: 384 TMn clocks
100: 512 TMn clocks
101: 640 TMn clocks
110: 768 TMn clocks
111: 896 TMn clocks
These three bits are used to setup the value on the internal CCRP 3-bit register, which are then compared with the internal counter's highest three bits. The result of this comparison can be selected to clear the internal counter if the TnCCLR bit is set to zero. Setting the TnCCLR bit to zero ensures that a compare match with the CCRP values will reset the internal counter. As the CCRP bits are only compared with the highest three counter bits, the compare values exist in 128 clock cycle multiples. Clearing all three bits to zero is in effect allowing the counter to overflow at its maximum value.

• **TMnC1 Register – 10-bit STM**

Bit	7	6	5	4	3	2	1	0
Name	TnM1	TnM0	TnIO1	TnIO0	TnOC	TnPOL	TnDPX	TnCCLR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **TnM1~TnM0:** Select TMn Operating Mode
00: Compare Match Output Mode
01: Capture Input Mode
10: PWM Mode or Single Pulse Output Mode
11: Timer/Counter Mode

These bits setup the required operating mode for the TM. To ensure reliable operation the TM should be switched off before any changes are made to the TnM1 and TnM0 bits. In the Timer/Counter Mode, the TM output pin control must be disabled.

Bit 5~4 **TnIO1~TnIO0:** Select TPn_0, TPn_1 output function
Compare Match Output Mode
00: No change
01: Output low
10: Output high
11: Toggle output
PWM Mode/Single Pulse Output Mode
00: PWM Output inactive state
01: PWM Output active state
10: PWM output
11: Single pulse output
Capture Input Mode
00: Input capture at rising edge of TPn_0, TPn_1
01: Input capture at falling edge of TPn_0, TPn_1
10: Input capture at falling/rising edge of TPn_0, TPn_1
11: Input capture disabled
Timer/counter Mode
Unused

These two bits are used to determine how the TM output pin changes state when a certain condition is reached. The function that these bits select depends upon in which mode the TM is running.

In the Compare Match Output Mode, the TnIO1 and TnIO0 bits determine how the TM output pin changes state when a compare match occurs from the Comparator A. The TM output pin can be setup to switch high, switch low or to toggle its present state when a compare match occurs from the Comparator A. When the bits are both zero, then no change will take place on the output. The initial value of the TM output pin should be setup using the TnOC bit in the TMnC1 register. Note that the output level requested by the TnIO1 and TnIO0 bits must be different from the initial value setup using the TnOC bit otherwise no change will occur on the TM output pin when a compare match occurs. After the TM output pin changes state it can be reset to its initial level by changing the level of the TnON bit from low to high.

In the PWM Mode, the TnIO1 and TnIO0 bits determine how the TM output pin changes state when a certain compare match condition occurs. The PWM output function is modified by changing these two bits. It is necessary to only change the values of the TnIO1 and TnIO0 bits only after the TM has been switched off. Unpredictable PWM outputs will occur if the TnIO1 and TnIO0 bits are changed when the TM is running.

Bit 3 **TnOC:** TPn_0, TPn_1 Output control bit
Compare Match Output Mode
0: Initial low
1: Initial high
PWM Mode/Single Pulse Output Mode
0: Active low
1: Active high
This is the output control bit for the TM output pin. Its operation depends upon whether TM is being used in the Compare Match Output Mode or in the PWM Mode/ Single Pulse Output Mode. It has no effect if the TM is in the Timer/Counter Mode. In the Compare Match Output Mode it determines the logic level of the TM output pin before a compare match occurs. In the PWM Mode it determines if the PWM signal is active high or active low.

Bit 2 **TnPOL:** TPn_0, TPn_1 Output polarity Control
0: Non-invert
1: Invert
This bit controls the polarity of the TPn_0 or TPn_1 output pin. When the bit is set high the TM output pin will be inverted and not inverted when the bit is zero. It has no effect if the TM is in the Timer/Counter Mode.

Bit 1 **TnDPX:** TMn PWM period/duty Control
0: CCRP - period; CCRA - duty
1: CCRP - duty; CCRA - period
This bit, determines which of the CCRA and CCRP registers are used for period and duty control of the PWM waveform.

Bit 0 **TnCCLR:** Select TMn Counter clear condition
0: TMn Comparator P match
1: TMn Comparator A match
This bit is used to select the method which clears the counter. Remember that the Standard TM contains two comparators, Comparator A and Comparator P, either of which can be selected to clear the internal counter. With the TnCCLR bit set high, the counter will be cleared when a compare match occurs from the Comparator A. When the bit is low, the counter will be cleared when a compare match occurs from the Comparator P or with a counter overflow. A counter overflow clearing method can only be implemented if the CCRP bits are all cleared to zero. The TnCCLR bit is not used in the PWM, Single Pulse or Input Capture Mode.

- **TMnDL Register – 10-bit STM**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~0 **TMnDL:** TMn Counter Low Byte Register bit 7~bit 0
TMn 10-bit Counter bit 7~bit 0

- **TMnDH Register – 10-bit STM**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R	R
POR	—	—	—	—	—	—	0	0

Bit 7~2 Unimplemented, read as "0"

Bit 1~0 **TMnDH:** TMn Counter High Byte Register bit 1~bit 0
TMn 10-bit Counter bit 9~bit 8

- **TMnAL Register – 10-bit STM**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **TMnAL:** TMn CCRA Low Byte Register bit 7~bit 0
TMn 10-bit CCRA bit 7~bit 0

- **TMnAH Register – 10-bit STM**

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 Unimplemented, read as "0"

Bit 1~0 **TMnAH:** TMn CCRA High Byte Register bit 1~bit 0
TM1 10-bit CCRA bit 9~bit 8

16-bit STM Register Definitions – HT69F50A

- **TM2C0 Register**

Bit	7	6	5	4	3	2	1	0
Name	T2PAU	T2CK2	T2CK1	T2CK0	T2ON	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	—	—	—
POR	0	0	0	0	0	—	—	—

Bit 7 **T2PAU:** TM2 Counter Pause Control
0: Run
1: Pause
The counter can be paused by setting this bit high. Clearing the bit to zero restores normal counter operation. When in a Pause condition the TM will remain powered up and continue to consume power. The counter will retain its residual value when this bit changes from low to high and resume counting from this value when the bit changes to a low value again.

Bit 6~4 **T2CK2, T2CK1, T2CK0:** Select TM2 Counter clock
000: $f_{SYS}/4$
001: f_{SYS}
010: $f_H/16$
011: $f_H/64$
100: f_{SUB}
101: Reserved
110: TCK2 rising edge clock
111: TCK2 falling edge clock
These three bits are used to select the clock source for the TM2. Selecting the Reserved clock input will effectively disable the internal counter. The external pin clock source can be chosen to be active on the rising or falling edge. The clock source f_{SYS} is the system clock, while f_H and f_{SUB} are other internal clocks, the details of which can be found in the oscillator section.

Bit 3 **T2ON:** TM2 Counter On/Off Control
0: Off
1: On
This bit controls the overall on/off function of the TM. Setting the bit high enables the counter to run, clearing the bit disables the TM. Clearing this bit to zero will stop the counter from counting and turn off the TM which will reduce its power consumption. When the bit changes state from low to high the internal counter value will be reset to zero, however when the bit changes from high to low, the internal counter will retain its residual value until the bit returns high again. If the TM is in the Compare Match Output Mode then the TM output pin will be reset to its initial condition, as specified by the T2OC bit, when the T2ON bit changes from low to high.

Bit 2~0 Unimplemented, read as "0"

- **TM2C1 Register – 16-bit STM**

Bit	7	6	5	4	3	2	1	0
Name	T2M1	T2M0	T2IO1	T2IO0	T2OC	T2POL	T2DPX	T2CCLR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **T2M1~T2M0:** Select TM2 Operating Mode
00: Compare Match Output Mode
01: Capture Input Mode
10: PWM Mode or Single Pulse Output Mode
11: Timer/Counter Mode

These bits setup the required operating mode for the TM. To ensure reliable operation the TM should be switched off before any changes are made to the T2M1 and T2M0 bits. In the Timer/Counter Mode, the TM output pin control must be disabled.

Bit 5~4 **T2IO1~T2IO0:** Select TP2_0, TP2_1 output function
Compare Match Output Mode
00: No change
01: Output low
10: Output high
11: Toggle output
PWM Mode/Single Pulse Output Mode
00: PWM output inactive state
01: PWM output active state
10: PWM output
11: Single pulse output
Capture Input Mode
00: Input capture at rising edge of TP2_0, TP2_1
01: Input capture at falling edge of TP2_0, TP2_1
10: Input capture at falling/rising edge of TP2_0, TP2_1
11: Input capture disabled
Timer/counter Mode
Unused

These two bits are used to determine how the TM output pin changes state when a certain condition is reached. The function that these bits select depends upon in which mode the TM is running.

In the Compare Match Output Mode, the T2IO1 and T2IO0 bits determine how the TM output pin changes state when a compare match occurs from the Comparator A. The TM output pin can be setup to switch high, switch low or to toggle its present state when a compare match occurs from the Comparator A. When the bits are both zero, then no change will take place on the output. The initial value of the TM output pin should be setup using the T2OC bit in the TM2C1 register. Note that the output level requested by the T2IO1 and T2IO0 bits must be different from the initial value setup using the T2OC bit otherwise no change will occur on the TM output pin when a compare match occurs. After the TM output pin changes state it can be reset to its initial level by changing the level of the T2ON bit from low to high.

In the PWM Mode, the T2IO1 and T2IO0 bits determine how the TM output pin changes state when a certain compare match condition occurs. The PWM output function is modified by changing these two bits. It is necessary to only change the values of the T2IO1 and T2IO0 bits only after the TM has been switched off. Unpredictable PWM outputs will occur if the T2IO1 and T2IO0 bits are changed when the TM is running.

Bit 3 **T2OC:** TP2_0, TP2_1 Output control bit
Compare Match Output Mode
0: Initial low
1: Initial high
PWM Mode/Single Pulse Output Mode
0: Active low
1: Active high
This is the output control bit for the TM output pin. Its operation depends upon whether TM is being used in the Compare Match Output Mode or in the PWM Mode/ Single Pulse Output Mode. It has no effect if the TM is in the Timer/Counter Mode. In the Compare Match Output Mode it determines the logic level of the TM output pin before a compare match occurs. In the PWM Mode it determines if the PWM signal is active high or active low.

Bit 2 **T2POL:** TP2_0, TP2_1 Output polarity Control
0: Non-invert
1: Invert
This bit controls the polarity of the TP2_0 or TP2_1 output pin. When the bit is set high the TM output pin will be inverted and not inverted when the bit is zero. It has no effect if the TM is in the Timer/Counter Mode.

Bit 1 **T2DPX:** TM2 PWM period/duty Control
0: CCRP - period; CCRA - duty
1: CCRP - duty; CCRA - period
This bit, determines which of the CCRA and CCRP registers are used for period and duty control of the PWM waveform.

Bit 0 **T2CCLR:** Select TM2 Counter clear condition
0: TM2 Comparator P match
1: TM2 Comparator A match
This bit is used to select the method which clears the counter. Remember that the Standard TM contains two comparators, Comparator A and Comparator P, either of which can be selected to clear the internal counter. With the T2CCLR bit set high, the counter will be cleared when a compare match occurs from the Comparator A. When the bit is low, the counter will be cleared when a compare match occurs from the Comparator P or with a counter overflow. A counter overflow clearing method can only be implemented if the CCRP bits are all cleared to zero. The T1CCLR bit is not used in the PWM, Single Pulse or Input Capture Mode.

- **TM2DL Register – 16-bit STM**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~0 **TM2DL:** TM2 Counter Low Byte Register bit 7~bit 0
TM2 16-bit Counter bit 7~bit 0

- **TM2DH Register – 16-bit STM**

Bit	7	6	5	4	3	2	1	0
Name	D15	D14	D13	D12	D11	D10	D9	D8
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~0 **TM2DH:** TM2 Counter High Byte Register bit 7~bit 0
TM2 16-bit Counter bit 15~bit 8

- **TM2AL Register – 16-bit STM**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **TM2AL:** TM2 CCRA Low Byte Register bit 7~bit 0
TM2 16-bit CCRA bit 7~bit 0

- **TM2AH Register – 16-bit STM**

Bit	7	6	5	4	3	2	1	0
Name	D15	D14	D13	D12	D11	D10	D9	D8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **TM2AH:** TM2 CCRA High Byte Register bit 7~bit 0
TM2 16-bit CCRA bit 15~bit 8

- **TM2RP Register – 16-bit STM**

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **TM2RP:** TM2 CCRP Register bit 7~bit 0
TM2 CCRP 8-bit register, compared with the TM2 Counter bit 15~bit 8.
Comparator P Match Period
0: 65536 TM2 clocks
1~255: 256×(1~255) TM2 clocks

These eight bits are used to setup the value on the internal CCRP 8-bit register, which are then compared with the internal counter's highest eight bits. The result of this comparison can be selected to clear the internal counter if the T2CCLR bit is set to zero. Setting the T2CCLR bit to zero ensures that a compare match with the CCRP values will reset the internal counter. As the CCRP bits are only compared with the highest eight counter bits, the compare values exist in 256 clock cycle multiples. Clearing all eight bits to zero is in effect allowing the counter to overflow at its maximum value.

Standard Type TM Operating Modes

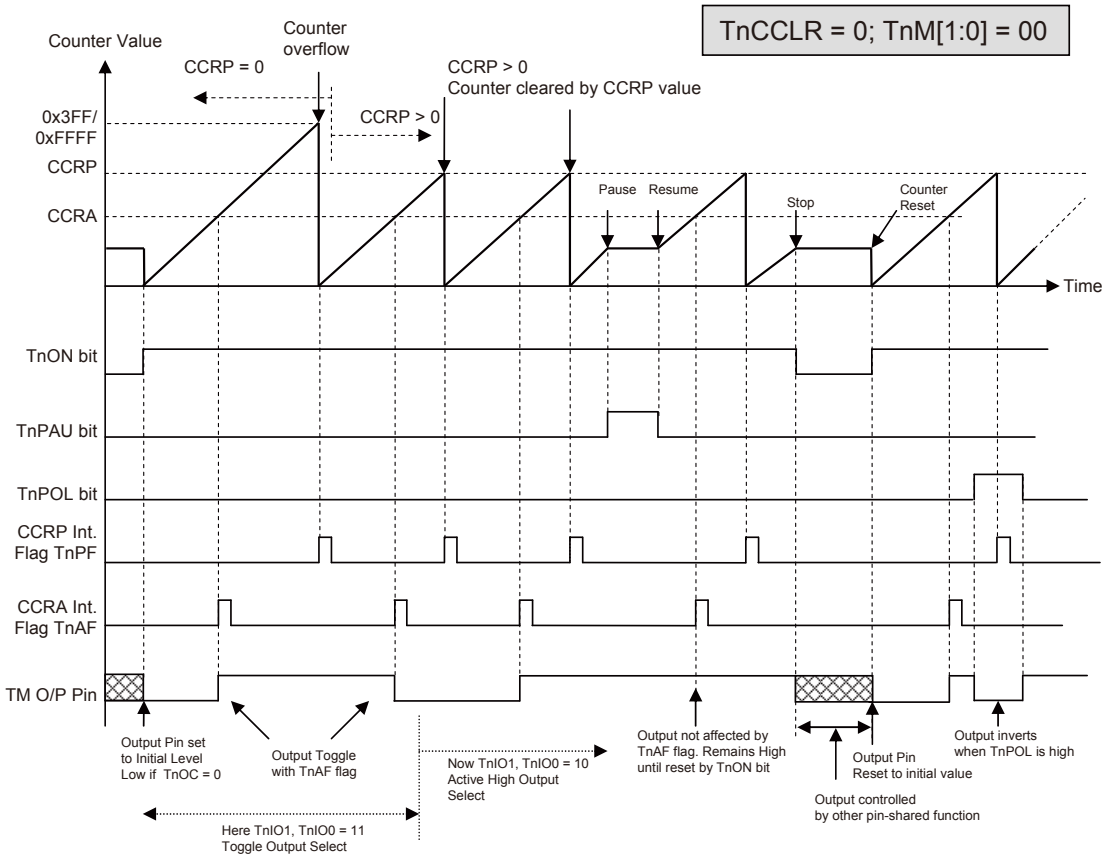
The Standard Type TM can operate in one of five operating modes, Compare Match Output Mode, PWM Output Mode, Single Pulse Output Mode, Capture Input Mode or Timer/Counter Mode. The operating mode is selected using the TnM1 and TnM0 bits in the TMnC1 register.

Compare Output Mode

To select this mode, bits TnM1 and TnM0 in the TMnC1 register, should be set to 00 respectively. In this mode once the counter is enabled and running it can be cleared by three methods. These are a counter overflow, a compare match from Comparator A and a compare match from Comparator P. When the TnCCLR bit is low, there are two ways in which the counter can be cleared. One is when a compare match from Comparator P, the other is when the CCRP bits are all zero which allows the counter to overflow. Here both TnAF and TnPF interrupt request flags for Comparator A and Comparator P respectively, will both be generated.

If the TnCCLR bit in the TMnC1 register is high then the counter will be cleared when a compare match occurs from Comparator A. However, here only the TnAF interrupt request flag will be generated even if the value of the CCRP bits is less than that of the CCRA registers. Therefore when TnCCLR is high no TnPF interrupt request flag will be generated. In the Compare Match Output Mode, the CCRA can not be set to "0".

As the name of the mode suggests, after a comparison is made, the TM output pin, will change state. The TM output pin condition however only changes state when an TnAF interrupt request flag is generated after a compare match occurs from Comparator A. The TnPF interrupt request flag, generated from a compare match occurs from Comparator P, will have no effect on the TM output pin. The way in which the TM output pin changes state are determined by the condition of the TnIO1 and TnIO0 bits in the TMnC1 register. The TM output pin can be selected using the TnIO1 and TnIO0 bits to go high, to go low or to toggle from its present condition when a compare match occurs from Comparator A. The initial condition of the TM output pin, which is setup after the TnON bit changes from low to high, is setup using the TnOC bit. Note that if the TnIO1 and TnIO0 bits are zero then no pin change will take place.



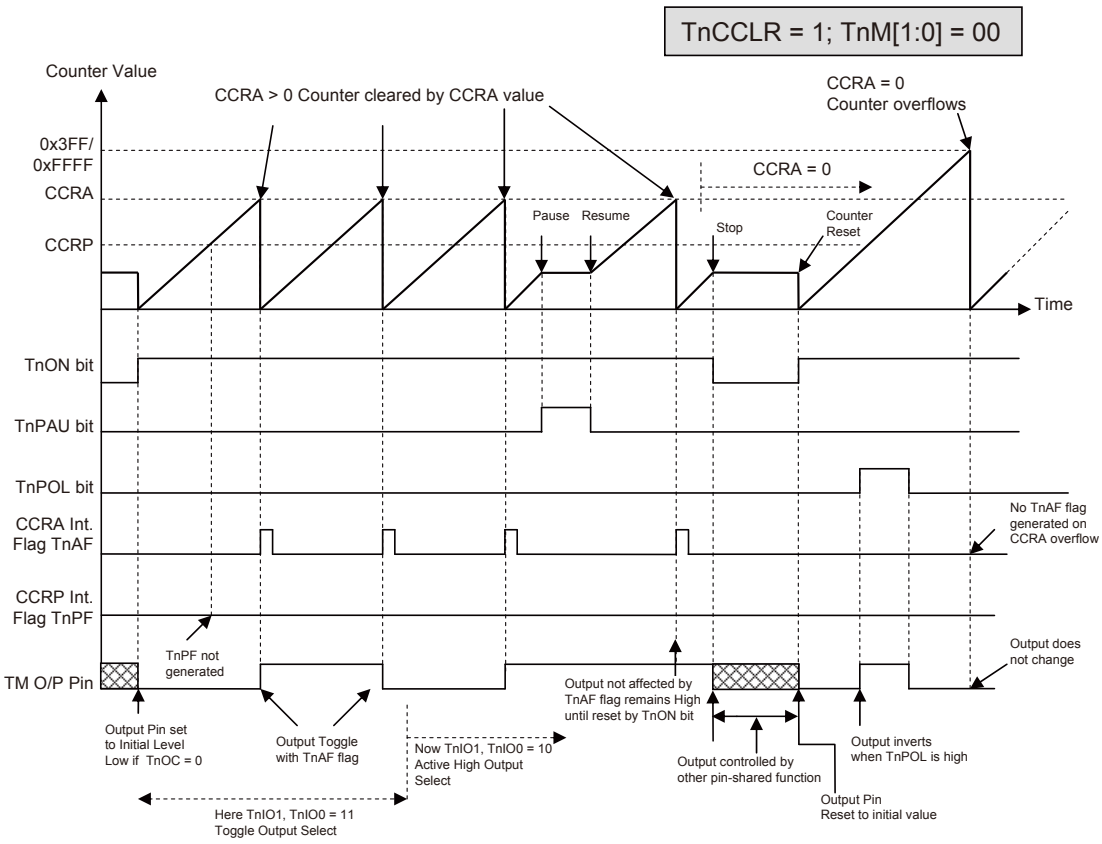
Compare Match Output Mode – TnCCLR=0

Note: 1. With TnCCLR=0, a Comparator P match will clear the counter

2. The TM output pin is controlled only by the TnAF flag

3. The output pin is reset to its initial state by a TnON bit rising edge

4. n=1 for HT69F30A; n=2 for HT69F40A/HT69F50A



Compare Match Output Mode – TnCCLR=1

Note: 1. With TnCCLR=1, a Comparator A match will clear the counter

2. The TM output pin is controlled only by the TnAF flag
3. The output pin is reset to its initial state by a TnON bit rising edge
4. A TnPF flag is not generated when TnCCLR=1
5. n=1 for HT69F30A; n=2 for HT69F40A/HT69F50A

Timer/Counter Mode

To select this mode, bits TnM1 and TnM0 in the TMnC1 register should be set to 11 respectively. The Timer/Counter Mode operates in an identical way to the Compare Match Output Mode generating the same interrupt flags. The exception is that in the Timer/Counter Mode the TM output pin is not used. Therefore the above description and Timing Diagrams for the Compare Match Output Mode can be used to understand its function. As the TM output pin is not used in this mode, the pin can be used as a normal I/O pin or other pin-shared function.

PWM Output Mode

To select this mode, bits TnM1 and TnM0 in the TMnC1 register should be set to 10 respectively and also the TnIO1 and TnIO0 bits should be set to 10 respectively. The PWM function within the TM is useful for applications which require functions such as motor control, heating control, illumination control etc. By providing a signal of fixed frequency but of varying duty cycle on the TM output pin, a square wave AC waveform can be generated with varying equivalent DC RMS values.

As both the period and duty cycle of the PWM waveform can be controlled, the choice of generated waveform is extremely flexible. In the PWM mode, the TnCCLR bit has no effect as the PWM period. Both of the CCRA and CCRP registers are used to generate the PWM waveform, one register is used to clear the internal counter and thus control the PWM waveform frequency, while the other one is used to control the duty cycle. Which register is used to control either frequency or duty cycle is determined using the TnDPX bit in the TMnC1 register. The PWM waveform frequency and duty cycle can therefore be controlled by the values in the CCRA and CCRP registers.

An interrupt flag, one for each of the CCRA and CCRP, will be generated when a compare match occurs from either Comparator A or Comparator P. The TnOC bit in the TMnC1 register is used to select the required polarity of the PWM waveform while the two TnIO1 and TnIO0 bits are used to enable the PWM output or to force the TM output pin to a fixed high or low level. The TnPOL bit is used to reverse the polarity of the PWM output waveform.

10-bit STM, PWM Mode, Edge-aligned Mode, TnDPX=0

CCRP	001b	010b	011b	100b	101b	110b	111b	000b
Period	128	256	384	512	640	768	896	1024
Duty	CCRA							

If f_{SYS}=16MHz, TM clock source select f_{SYS}/4, CCRP=100b and CCRA=128,

The STM PWM output frequency=(f_{SYS}/4)/512=f_{SYS}/2048=7.8125kHz, duty=128/512=25%

If the Duty value defined by the CCRA register is equal to or greater than the Period value, then the PWM output duty is 100%.

10-bit STM, PWM Mode, Edge-aligned Mode, TnDPX=1

CCRP	001b	010b	011b	100b	101b	110b	111b	000b
Period	CCRA							
Duty	128	256	384	512	640	768	896	1024

The PWM output period is determined by the CCRA register value together with the TM clock while the PWM duty cycle is defined by the CCRP register value.

16-bit STM, PWM Mode, Edge-aligned Mode, TnDPX=0

CCRP	1~255	000b
Period	CCRP×256	65536
Duty	CCRA	

If f_{SYS}=16MHz, TM clock source select f_{SYS}/4, CCRP=2 and CCRA=128,

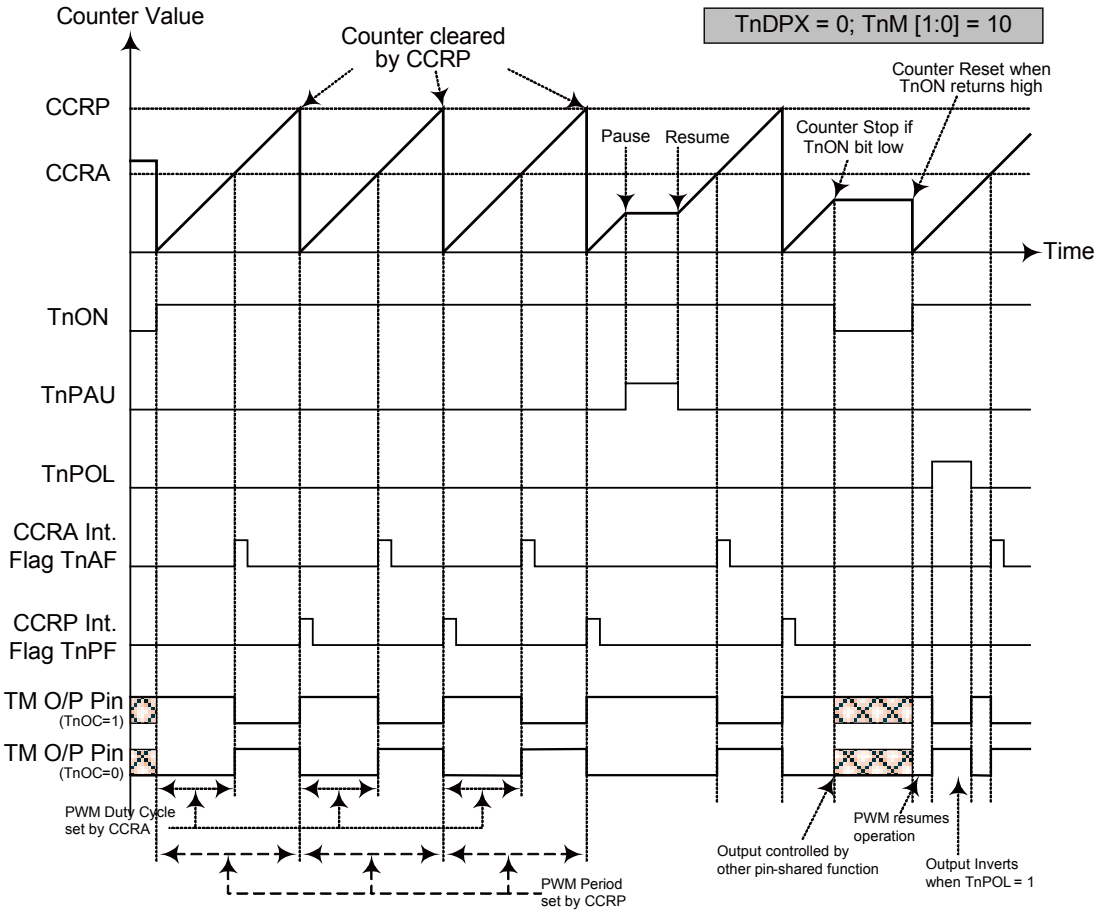
The STM PWM output frequency=(f_{SYS}/4)/(2×256)=f_{SYS}/2048=7.8125kHz, duty=128/(2×256)=25%.

If the Duty value defined by the CCRA register is equal to or greater than the Period value, then the PWM output duty is 100%.

16-bit STM, PWM Mode, Edge-aligned Mode, TnDPX=1

CCRP	1~255	000b
Period	CCRA	
Duty	CCRP×256	65536

The PWM output period is determined by the CCRA register value together with the TM clock while the PWM duty cycle is defined by the (CCRP×256) except when the CCRP value is equal to 000b.



PWM Mode – TnDPX=0

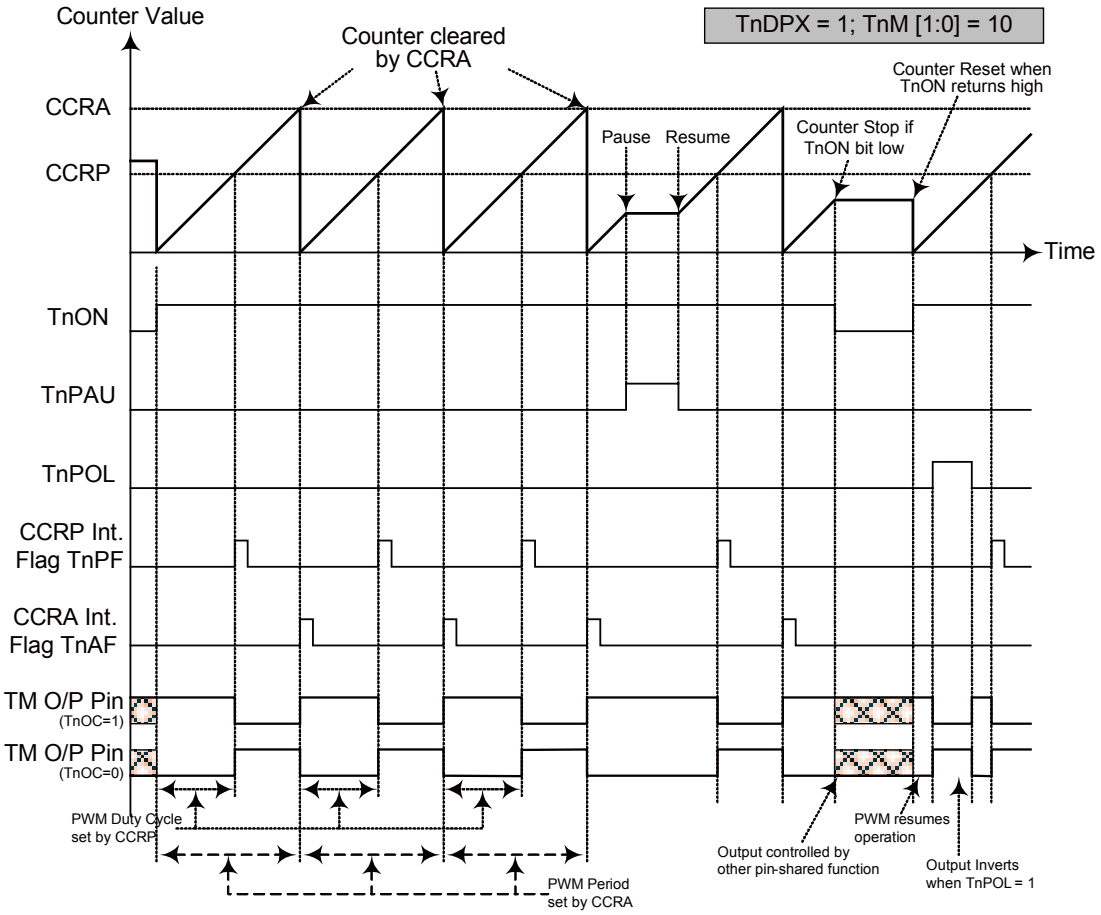
Note: 1. Here TnDPX=0, Counter cleared by CCRP

2. A counter clear sets the PWM Period

3. The internal PWM function continues running even when TnIO [1:0]=00 or 01

4. The TnCCLR bit has no influence on PWM operation

5. n=1 for HT69F30A; n=2 for HT69F40A/HT69F50A



PWM Mode – TnDPX=1

Note: 1. Here TnDPX=1 -- Counter cleared by CCRA

2. A counter clear sets the PWM Period

3. The internal PWM function continues running even when TnIO [1:0]=00 or 01

4. The TnCCLR bit has no influence on PWM operation

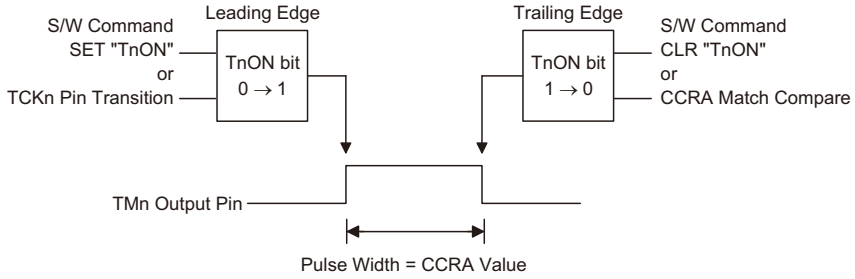
5. n=1 for HT69F30A; n=2 for HT69F40A/HT69F50A

Single Pulse Mode

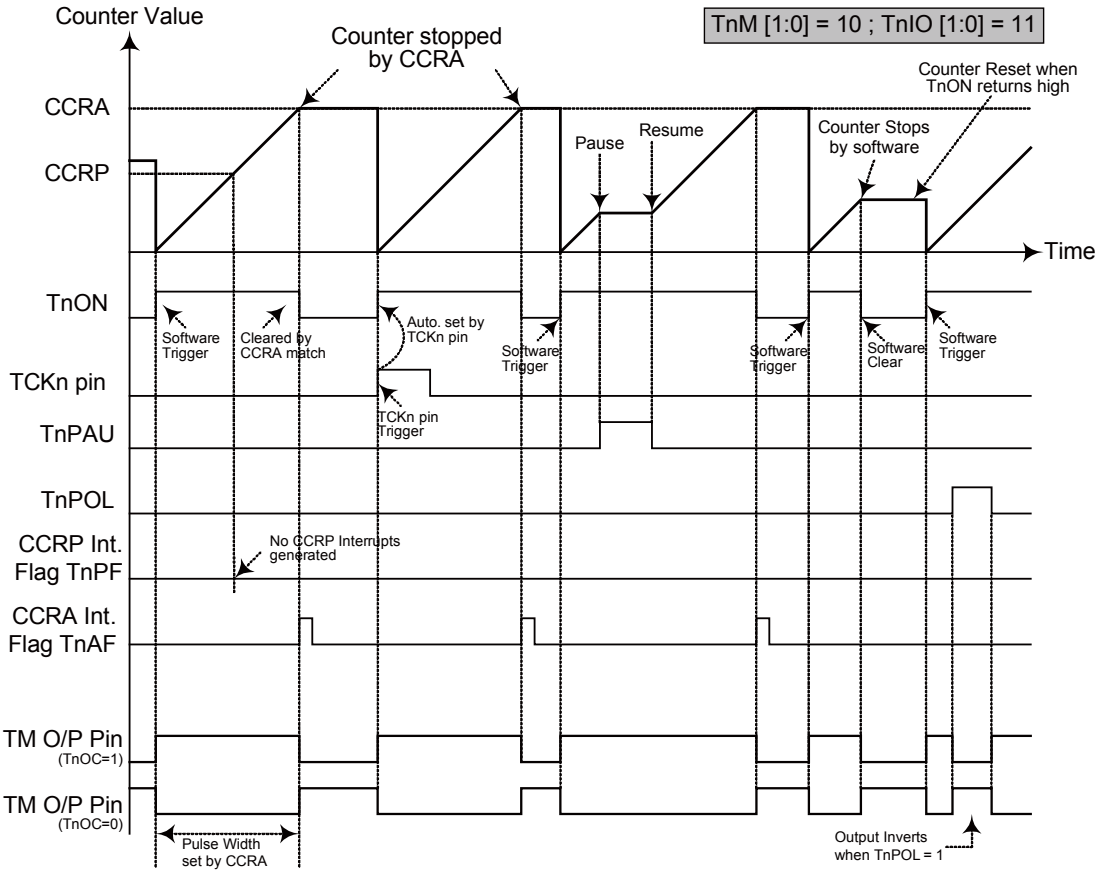
To select this mode, bits TnM1 and TnM0 in the TMnC1 register should be set to 10 respectively and also the TnIO1 and TnIO0 bits should be set to 11 respectively. The Single Pulse Output Mode, as the name suggests, will generate a single shot pulse on the TM output pin.

The trigger for the pulse output leading edge is a low to high transition of the TnON bit, which can be implemented using the application program. However in the Single Pulse Mode, the TnON bit can also be made to automatically change from low to high using the external TCKn pin, which will in turn initiate the Single Pulse output. When the TnON bit transitions to a high level, the counter will start running and the pulse leading edge will be generated. The TnON bit should remain high when the pulse is in its active state. The generated pulse trailing edge will be generated when the TnON bit is cleared to zero, which can be implemented using the application program or when a compare match occurs from Comparator A.

However a compare match from Comparator A will also automatically clear the TnON bit and thus generate the Single Pulse output trailing edge. In this way the CCRA value can be used to control the pulse width. A compare match from Comparator A will also generate a TM interrupt. The counter can only be reset back to zero when the TnON bit changes from low to high when the counter restarts. In the Single Pulse Mode CCRP is not used. The TnCCLR and TnDPX bits are not used in this Mode.



Single Pulse Generation



Single Pulse Mode

Note: 1. Counter stopped by CCRA

2. CCRP is not used

3. The pulse triggered by the TCKn pin or by setting the TnON bit high

4. A TCKn pin active edge will automatically set the TnON bit high

5. In the Single Pulse Mode, TnIO [1:0] must be set to “11” and can not be changed

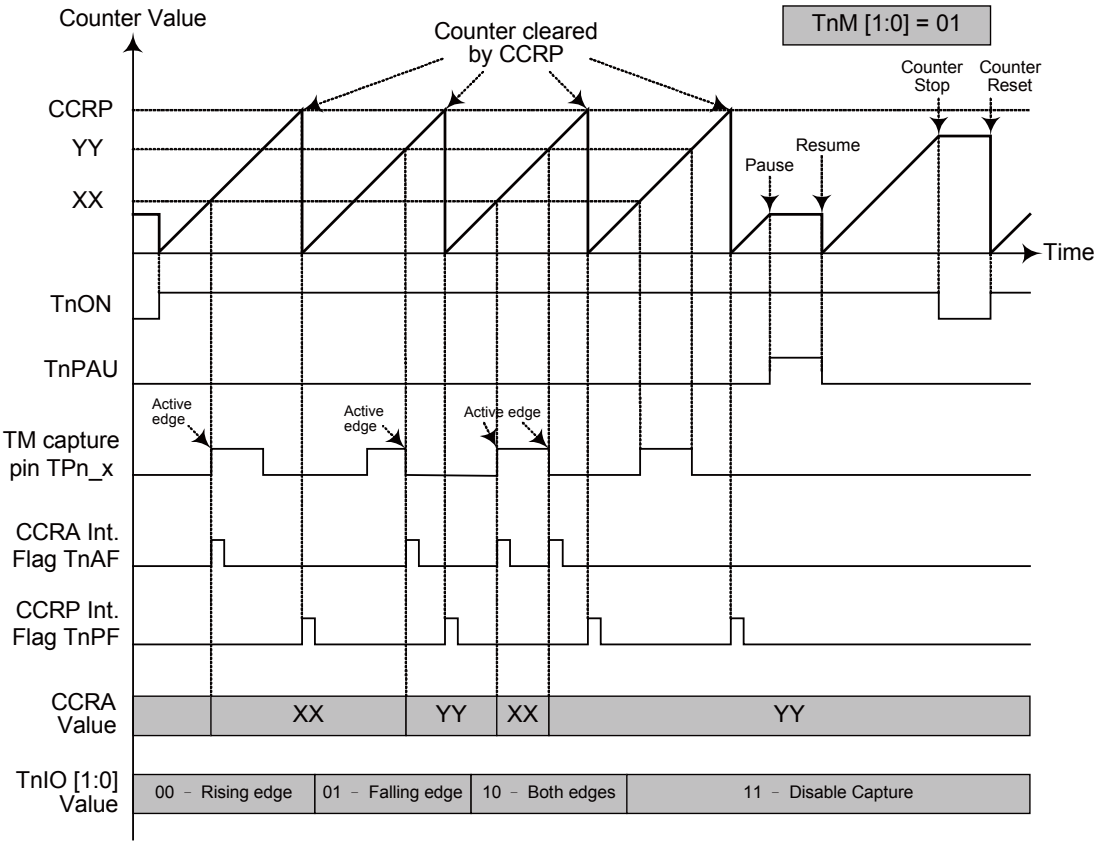
6. n=1 for HT69F30A; n=2 for HT69F40A/HT69F50A

Capture Input Mode

To select this mode bits TnM1 and TnM0 in the TMnC1 register should be set to 01 respectively. This mode enables external signals to capture and store the present value of the internal counter and can therefore be used for applications such as pulse width measurements. The external signal is supplied on the TPn_0 or TPn_1 pin, whose active edge can be either a rising edge, a falling edge or both rising and falling edges; the active edge transition type is selected using the TnIO1 and TnIO0 bits in the TMnC1 register. The counter is started when the TnON bit changes from low to high which is initiated using the application program.

When the required edge transition appears on the TPn_0 or TPn_1 pin the present value in the counter will be latched into the CCRA registers and a TM interrupt generated. Irrespective of what events occur on the TPn_0 or TPn_1 pin the counter will continue to free run until the TnON bit changes from high to low. When a CCRP compare match occurs the counter will reset back to zero; in this way the CCRP value can be used to control the maximum counter value. When a CCRP compare match occurs from Comparator P, a TM interrupt will also be generated. Counting the number of overflow interrupt signals from the CCRP can be a useful method in measuring long pulse widths. The TnIO1 and TnIO0 bits can select the active trigger edge on the TPn_0 or TPn_1 pin to be a rising edge, falling edge or both edge types. If the TnIO1 and TnIO0 bits are both set high, then no capture operation will take place irrespective of what happens on the TPn_0 or TPn_1 pin, however it must be noted that the counter will continue to run.

As the TPn_0 or TPn_1 pin is pin shared with other functions, care must be taken if the TM is in the Input Capture Mode. This is because if the pin is setup as an output, then any transitions on this pin may cause an input capture operation to be executed. The TnCCLR and TnDPX bits are not used in this Mode.



Capture Input Mode

Note: 1. TnM [1:0]=01 and active edge set by the TnIO [1:0] bits
2. A TM Capture input pin active edge transfers the counter value to CCRA
3. TnCCLR bit not used
4. No output function -- TnOC and TnPOL bits are not used
5. CCRP determines the counter value and the counter has a maximum count value when CCRP is equal to zero
6. n=1 for HT69F30A; n=2 for HT69F40A/HT69F50A

Enhanced Type TM – ETM

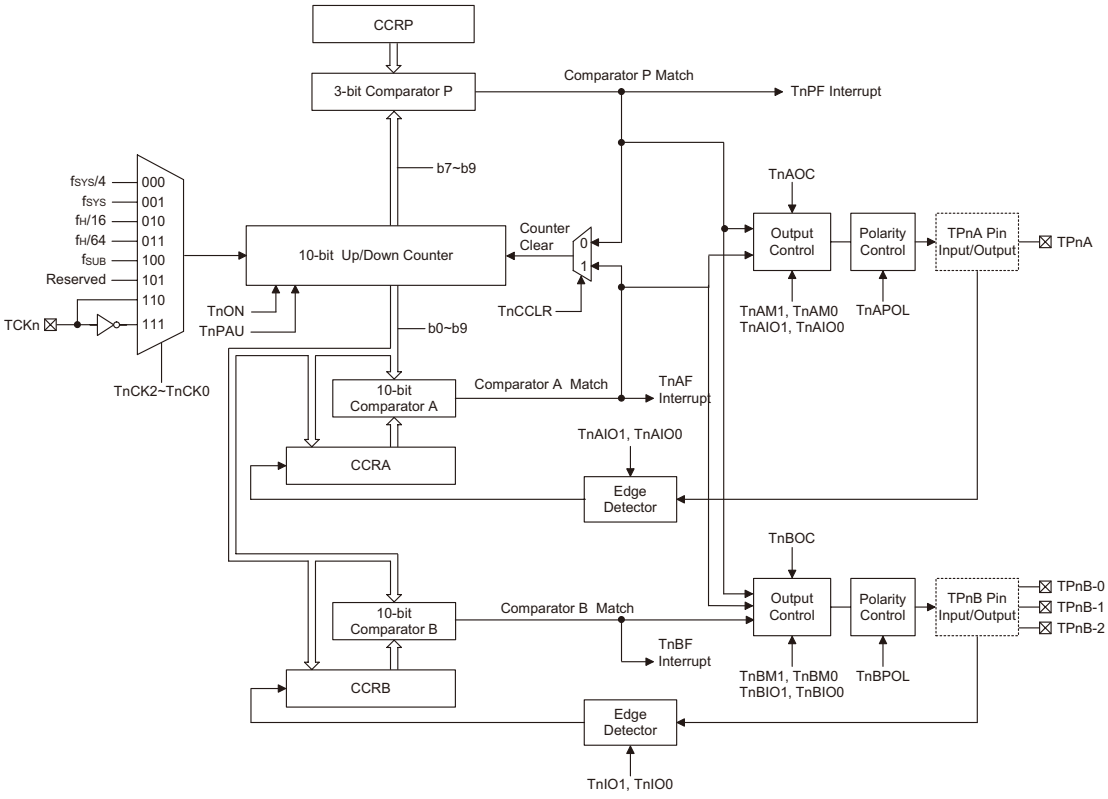
The Enhanced Type TM contains five operating modes, which are Compare Match Output, Timer/Event Counter, Capture Input, Single Pulse Output and PWM Output modes. The Enhanced TM can also be controlled with an external input pin and can drive three or four external output pins.

Device	TM Type	TM Name.	TM Input Pin	TM Output Pin
HT69F30A	—	—	—	—
HT69F40A HT69F50A	10-bit ETM	TM1	TCK1	TP1A; TP1B_0, TP1B_1, TP1B_2,

Enhanced TM Operation

At its core is a 10-bit count-up/count-down counter which is driven by a user selectable internal or external clock source. There are three internal comparators with the names, Comparator A, Comparator B and Comparator P. These comparators will compare the value in the counter with the CCRA, CCRB and CCRP registers. The CCRP comparator is 3-bits wide whose value is compared with the highest 3-bits in the counter while CCRA and CCRB are 10-bits wide and therefore compared with all counter bits.

The only way of changing the value of the 10-bit counter using the application program, is to clear the counter by changing the TnON bit from low to high. The counter will also be cleared automatically by a counter overflow or a compare match with one of its associated comparators. When these conditions occur, a TM interrupt signal will also usually be generated. The Enhanced Type TM can operate in a number of different operational modes, can be driven by different clock sources including an input pin and can also control output pins. All operating setup conditions are selected using relevant internal registers.



Enhanced Type TM Block Diagram (n=1)

Enhanced Type TM Register Description

Overall operation of the Enhanced TM is controlled using a series of registers. A read only register pair exists to store the internal counter 10-bit value, while two read/write register pairs exist to store the internal 10-bit CCRA and CCRB value. The remaining three registers are control registers which setup the different operating and control modes as well as the three CCRP bits.

Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TM1C0	T1PAU	T1CK2	T1CK1	T1CK0	T1ON	T1RP2	T1RP1	T1RP0
TM1C1	T1AM1	T1AM0	T1AIO1	T1AIO0	T1AOC	T1APOL	T1CDN	T1CCLR
TM1C2	T1BM1	T1BM0	T1BIO1	T1BIO0	T1BOC	T1BPOL	T1PWM1	T1PWM0
TM1DL	D7	D6	D5	D4	D3	D2	D1	D0
TM1DH	—	—	—	—	—	—	D9	D8
TM1AL	D7	D6	D5	D4	D3	D2	D1	D0
TM1AH	—	—	—	—	—	—	D9	D8
TM1BL	D7	D6	D5	D4	D3	D2	D1	D0
TM1BH	—	—	—	—	—	—	D9	D8

10-bit Enhanced TM Register List

TM1C0 Register – 10-bit ETM

Bit	7	6	5	4	3	2	1	0
Name	T1PAU	T1CK2	T1CK1	T1CK0	T1ON	T1RP2	T1RP1	T1RP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **T1PAU:** TM1 Counter Pause Control
0: Run
1: Pause

The counter can be paused by setting this bit high. Clearing the bit to zero restores normal counter operation. When in a Pause condition the TM will remain powered up and continue to consume power. The counter will retain its residual value when this bit changes from low to high and resume counting from this value when the bit changes to a low value again.

Bit 6~4 **T1CK2~T1CK0:** Select TM1 Counter clock
000: $f_{SYS}/4$
001: f_{SYS}
010: $f_H/16$
011: $f_H/64$
100: f_{SUB}
101: Reserved
110: TCK1 rising edge clock
111: TCK1 falling edge clock

These three bits are used to select the clock source for the TM. Selecting the Reserved clock input will effectively disable the internal counter. The external pin clock source can be chosen to be active on the rising or falling edge. The clock source f_{SYS} is the system clock, while f_H and f_{SUB} are other internal clocks, the details of which can be found in the oscillator section.

Bit 3 **T1ON:** TM1 Counter On/Off Control
0: Off
1: On

This bit controls the overall on/off function of the TM. Setting the bit high enables the counter to run, clearing the bit disables the TM. Clearing this bit to zero will stop the counter from counting and turn off the TM which will reduce its power consumption. When the bit changes state from low to high the internal counter value will be reset to zero, however when the bit changes from high to low, the internal counter will retain its residual value until the bit returns high again.

If the TM is in the Compare Match Output Mode then the TM output pin will be reset to its initial condition, as specified by the T1OC bit, when the T1ON bit changes from low to high.

Bit 2~0 **T1RP2~T1RP0:** TM1 CCRP 3-bit register, compared with the TM1 Counter bit 9~bit 7 Comparator P Match Period
000: 1024 TM1 clocks
001: 128 TM1 clocks
010: 256 TM1 clocks
011: 384 TM1 clocks
100: 512 TM1 clocks
101: 640 TM1 clocks
110: 768 TM1 clocks
111: 896 TM1 clocks

These three bits are used to setup the value on the internal CCRP 3-bit register, which are then compared with the internal counter highest three bits. The result of this comparison can be selected to clear the internal counter if the T1CCLR bit is set to zero. Setting the T1CCLR bit to zero ensures that a compare match with the CCRP values will reset the internal counter. As the CCRP bits are only compared with the highest three counter bits, the compare values exist in 128 clock cycle multiples. Clearing all three bits to zero is in effect allowing the counter to overflow at its maximum value.

TM1C1 Register – 10-bit ETM

Bit	7	6	5	4	3	2	1	0
Name	T1AM1	T1AM0	T1AIO1	T1AIO0	T1AOC	T1APOL	T1CDN	T1CCLR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **T1AM1~T1AM0:** Select TM1 CCRA Operating Mode
00: Compare Match Output Mode
01: Capture Input Mode
10: PWM Mode or Single Pulse Output Mode
11: Timer/Counter Mode

These bits setup the required operating mode for the TM. To ensure reliable operation the TM should be switched off before any changes are made to the T1AM1 and T1AM0 bits. In the Timer/Counter Mode, the TM output pin control must be disabled.

Bit 5~4 **T1AIO1~T1AIO0:** Select TP1A output function
Compare Match Output Mode
00: No change
01: Output low
10: Output high
11: Toggle output
PWM Mode/Single Pulse Output Mode
00: PWM output inactive state
01: PWM output active state
10: PWM output
11: Single pulse output
Capture Input Mode
00: Input capture at rising edge of TP1A
01: Input capture at falling edge of TP1A
10: Input capture at falling/rising edge of TP1A
11: Input capture disabled
Timer/counter Mode
Unused

These two bits are used to determine how the TM output pin changes state when a certain condition is reached. The function that these bits select depends upon in which mode the TM is running.

In the Compare Match Output Mode, the T1AIO1 and T1AIO0 bits determine how the TM output pin changes state when a compare match occurs from the Comparator A. The TM output pin can be setup to switch high, switch low or to toggle its present state when a compare match occurs from the Comparator A. When the bits are both zero, then no change will take place on the output. The initial value of the TM output pin should be setup using the T1AOC bit in the TM1C1 register. Note that the output level requested by the T1AIO1 and T1AIO0 bits must be different from the initial value setup using the T1AOC bit otherwise no change will occur on the TM output pin when a compare match occurs. After the TM output pin changes state it can be reset to its initial level by changing the level of the T1ON bit from low to high.

In the PWM Mode, the T1AIO1 and T1AIO0 bits determine how the TM output pin changes state when a certain compare match condition occurs. The PWM output function is modified by changing these two bits. It is necessary to change the values of the T1AIO1 and T1AIO0 bits only after the TM has been switched off. Unpredictable PWM outputs will occur if the T1AIO1 and T1AIO0 bits are changed when the TM is running.

Bit 3 **T1AOC:** TP1A Output control bit
Compare Match Output Mode
0: Initial low
1: Initial high
PWM Mode/Single Pulse Output Mode
0: Active low
1: Active high
This is the output control bit for the TM output pin. Its operation depends upon whether TM is being used in the Compare Match Output Mode or in the PWM Mode/ Single Pulse Output Mode. It has no effect if the TM is in the Timer/Counter Mode. In the Compare Match Output Mode it determines the logic level of the TM output pin before a compare match occurs. In the PWM Mode it determines if the PWM signal is active high or active low.

Bit 2 **T1APOL:** TP1A Output polarity Control
0: Non-invert
1: Invert
This bit controls the polarity of the TP1A output pin. When the bit is set high the TM output pin will be inverted and not inverted when the bit is zero. It has no effect if the TM is in the Timer/Counter Mode.

Bit 1 **T1CDN:** TM1 Counter count up or down flag
0: Count up
1: Count down

Bit 0 **T1CCLR:** Select TM1 Counter clear condition
0: TM1 Comparator P match
1: TM1 Comparator A match
This bit is used to select the method which clears the counter. Remember that the Enhanced TM contains three comparators, Comparator A, Comparator B and Comparator P, but only Comparator A or Comparator P can be selected to clear the internal counter. With the T1CCLR bit set high, the counter will be cleared when a compare match occurs from the Comparator A. When the bit is low, the counter will be cleared when a compare match occurs from the Comparator P or with a counter overflow. A counter overflow clearing method can only be implemented if the CCRP bits are all cleared to zero. The T1CCLR bit is not used in the Single Pulse or Input Capture Mode.

TM1C2 Register – 10-bit ETM

Bit	7	6	5	4	3	2	1	0
Name	T1BM1	T1BM0	T1BIO1	T1BIO0	T1BOC	T1BPOL	T1PWM1	T1PWM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 **T1BM1~T1BM0:** Select TM1 CCRB Operating Mode
00: Compare Match Output Mode
01: Capture Input Mode
10: PWM Mode or Single Pulse Output Mode
11: Timer/Counter mode

These bits setup the required operating mode for the TM. To ensure reliable operation the TM should be switched off before any changes are made to the T1BM1 and T1BM0 bits. In the Timer/Counter Mode, the TM output pin control must be disabled.

Bit 5~4 **T1BIO1~T1BIO0:** Select TP1B_0, TP1B_1, TP1B_2 output function
Compare Match Output Mode
00: No change
01: Output low
10: Output high
11: Toggle output
PWM Mode/Single Pulse Output Mode
00: PWM Output inactive state
01: PWM Output active state
10: PWM output
11: Single pulse output
Capture Input Mode
00: Input capture at rising edge of TP1B_0, TP1B_1, TP1B_2
01: Input capture at falling edge of TP1B_0, TP1B_1, TP1B_2
10: Input capture at falling/rising edge of TP1B_0, TP1B_1, TP1B_2
11: Input capture disabled
Timer/counter Mode
Unused

These two bits are used to determine how the TM output pin changes state when a certain condition is reached. The function that these bits select depends upon in which mode the TM is running.

In the Compare Match Output Mode, the T1BIO1 and T1BIO0 bits determine how the TM output pin changes state when a compare match occurs from the Comparator A. The TM output pin can be setup to switch high, switch low or to toggle its present state when a compare match occurs from the Comparator A. When the bits are both zero, then no change will take place on the output. The initial value of the TM output pin should be setup using the T1BOC bit in the TM1C2 register. Note that the output level requested by the T1BIO1 and T1BIO0 bits must be different from the initial value setup using the T1BOC bit otherwise no change will occur on the TM output pin when a compare match occurs. After the TM output pin changes state it can be reset to its initial level by changing the level of the T1ON bit from low to high.

In the PWM Mode, the T1BIO1 and T1BIO0 bits determine how the TM output pin changes state when a certain compare match condition occurs. The PWM output function is modified by changing these two bits. It is necessary to only change the value of the T1BIO1 and T1BIO0 bits only after the TM has been switched off. Unpredictable PWM outputs will occur if the T1BIO1 and T1BIO0 bits are changed when the TM is running.

Bit 3 **T1BOC:** TP1B_0, TP1B_1, TP1B_2 Output control bit
Compare Match Output Mode
0: Initial low
1: Initial high
PWM Mode/Single Pulse Output Mode
0: Active low
1: Active high
This is the output control bit for the TM output pin. Its operation depends upon whether TM is being used in the Compare Match Output Mode or in the PWM Mode/ Single Pulse Output Mode. It has no effect if the TM is in the Timer/Counter Mode. In the Compare Match Output Mode it determines the logic level of the TM output pin before a compare match occurs. In the PWM Mode it determines if the PWM signal is active high or active low.

Bit 2 **T1BPOL:** TP1B_0, TP1B_1, TB1B_2 Output polarity Control
0: Non-invert
1: Invert
This bit controls the polarity of the TP1B_0, TP1B_1, TP1B_2 output pin. When the bit is set high the TM output pin will be inverted and not inverted when the bit is zero. It has no effect if the TM is in the Timer/Counter Mode.

Bit 1~0 **T1PWM1~T1PWM0:** Select PWM Mode
00: Edge aligned
01: Centre aligned, compare match on count up
10: Centre aligned, compare match on count down
11: Centre aligned, compare match on count up or down

TM1DL Register – 10-bit ETM

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~0 **TM1DL:** TM1 Counter Low Byte Register bit 7~bit 0
TM1 10-bit Counter bit 7~bit 0

TM1DH Register – 10-bit ETM

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R	R
POR	—	—	—	—	—	—	0	0

Bit 7~2 Unimplemented, read as "0"

Bit 1~0 **TM1DH:** TM1 Counter High Byte Register bit 1~bit 0
TM1 10-bit Counter bit 9~bit 8

TM1AL Register – 10-bit ETM

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **TM1AL:** TM1 CCRA Low Byte Register bit 7~bit 0
TM1 10-bit CCRA bit 7~bit 0

TM1AH Register – 10-bit ETM

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 Unimplemented, read as "0"

Bit 1~0 **TM1AH:** TM1 CCRA High Byte Register bit 1~bit 0
TM1 10-bit CCRA bit 9~bit 8

TM1BL Register – 10-bit ETM

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~0 **TM1BL:** TM1 CCRB Low Byte Register bit 7~bit 0
TM1 10-bit CCRB bit 7~bit 0

TM1BH Register – 10-bit ETM

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D9	D8
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	0	0

Bit 7~2 Unimplemented, read as "0"

Bit 1~0 **TM1BH:** TM1 CCRB High Byte Register bit 1~bit 0
TM1 10-bit CCRB bit 9~bit 8

Enhanced Type TM Operating Modes

The Enhanced Type TM can operate in one of five operating modes, Compare Match Output Mode, PWM Output Mode, Single Pulse Output Mode, Capture Input Mode or Timer/Counter Mode. The operating mode is selected using the TnAM1 and TnAM0 bits in the TMnC1, and the TnBM1 and TnBM0 bits in the TMnC2 register.

ETM Operating Mode	CCRA Compare Match Output Mode	CCRA Timer/ Counter Mode	CCRA PWM Output Mode	CCRA Single Pulse Output Mode	CCRA Input Capture Mode
CCRB Compare Match Output Mode	√	—	—	—	—
CCRB Timer/Counter Mode	—	√	—	—	—
CCRB PWM Output Mode	—	—	√	—	—
CCRB Single Pulse Output Mode	—	—	—	√	—
CCRB Input Capture Mode	—	—	—	—	√

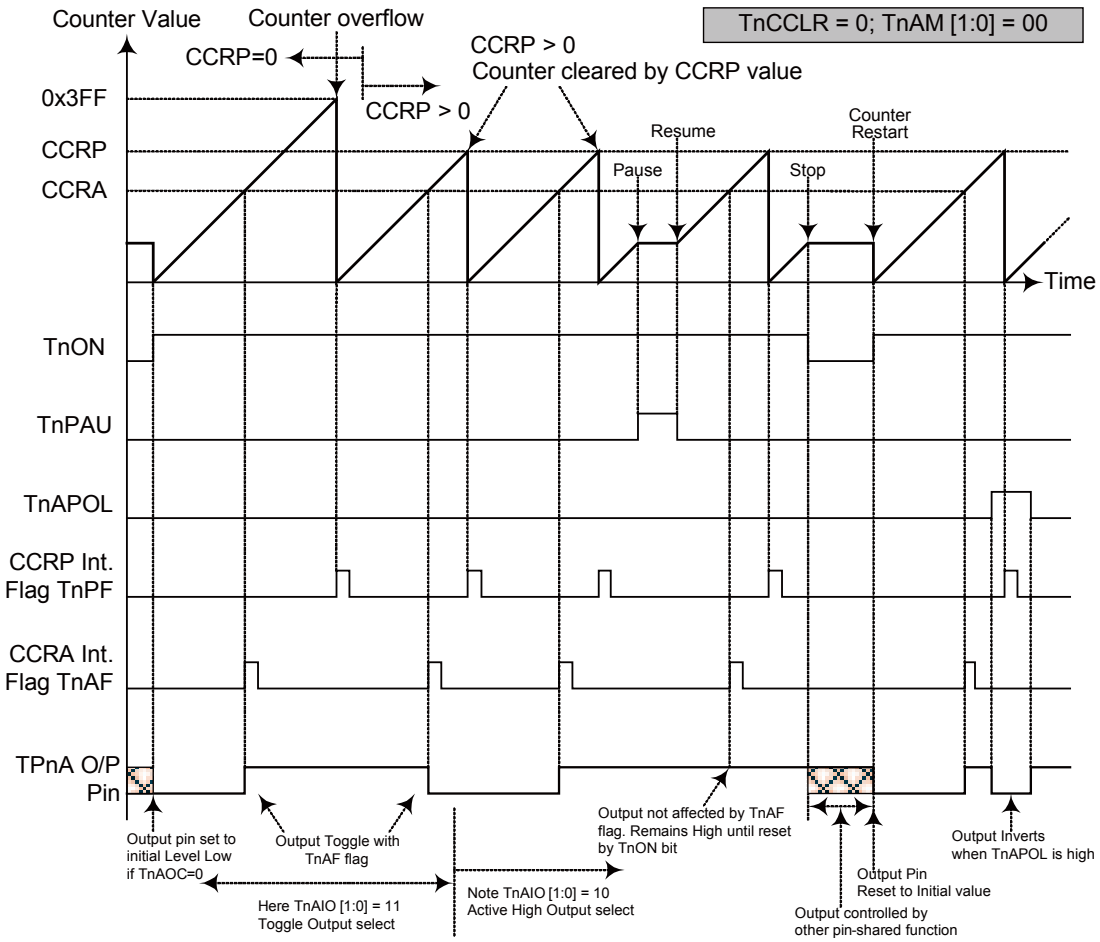
"√": permitted; "—": not permitt

Compare Output Mode

To select this mode, bits TnAM1, TnAM0 and TnBM1, TnBM0 in the TMnC1/TMnC2 registers should be all cleared to zero. In this mode once the counter is enabled and running it can be cleared by three methods. These are a counter overflow, a compare match from Comparator A and a compare match from Comparator P. When the TnCCLR bit is low, there are two ways in which the counter can be cleared. One is when a compare match occurs from Comparator P, the other is when the CCRP bits are all zero which allows the counter to overflow. Here both the TnAF and TnPF interrupt request flags for Comparator A and Comparator P respectively, will both be generated.

If the TnCCLR bit in the TMnC1 register is high then the counter will be cleared when a compare match occurs from Comparator A. However, here only the TnAF interrupt request flag will be generated even if the value of the CCRP bits is less than that of the CCRA registers. Therefore when TnCCLR is high no TnPF interrupt request flag will be generated.

As the name of the mode suggests, after a comparison is made, the TM output pin, will change state. The TM output pin condition however only changes state when an TnAF or TnBF interrupt request flag is generated after a compare match occurs from Comparator A or Comparator B. The TnPF interrupt request flag, generated from a compare match from Comparator P, will have no effect on the TM output pin. The way in which the TM output pin changes state is determined by the condition of the TnAIO1 and TnAIO0 bits in the TMnC1 register for ETM CCRA, and the TnBIO1 and TnBIO0 bits in the TMnC2 register for ETM CCRB. The TM output pin can be selected using the TnAIO1, TnAIO0 bits (for the TPnA pin) and TnBIO1, TnBIO0 bits (for the TPnB_0, TPnB_1 or TPnB_2 pins) to go high, to go low or to toggle from its present condition when a compare match occurs from Comparator A or a compare match occurs from Comparator B. The initial condition of the TM output pin, which is setup after the TnON bit changes from low to high, is setup using the TnAOC or TnBOC bit for TPnA or TPnB_0, TPnB_1, TPnB_2 output pins. Note that if the TnAIO1,TnAIO0 and TnBIO1, TnBIO0 bits are zero then no pin change will take place.



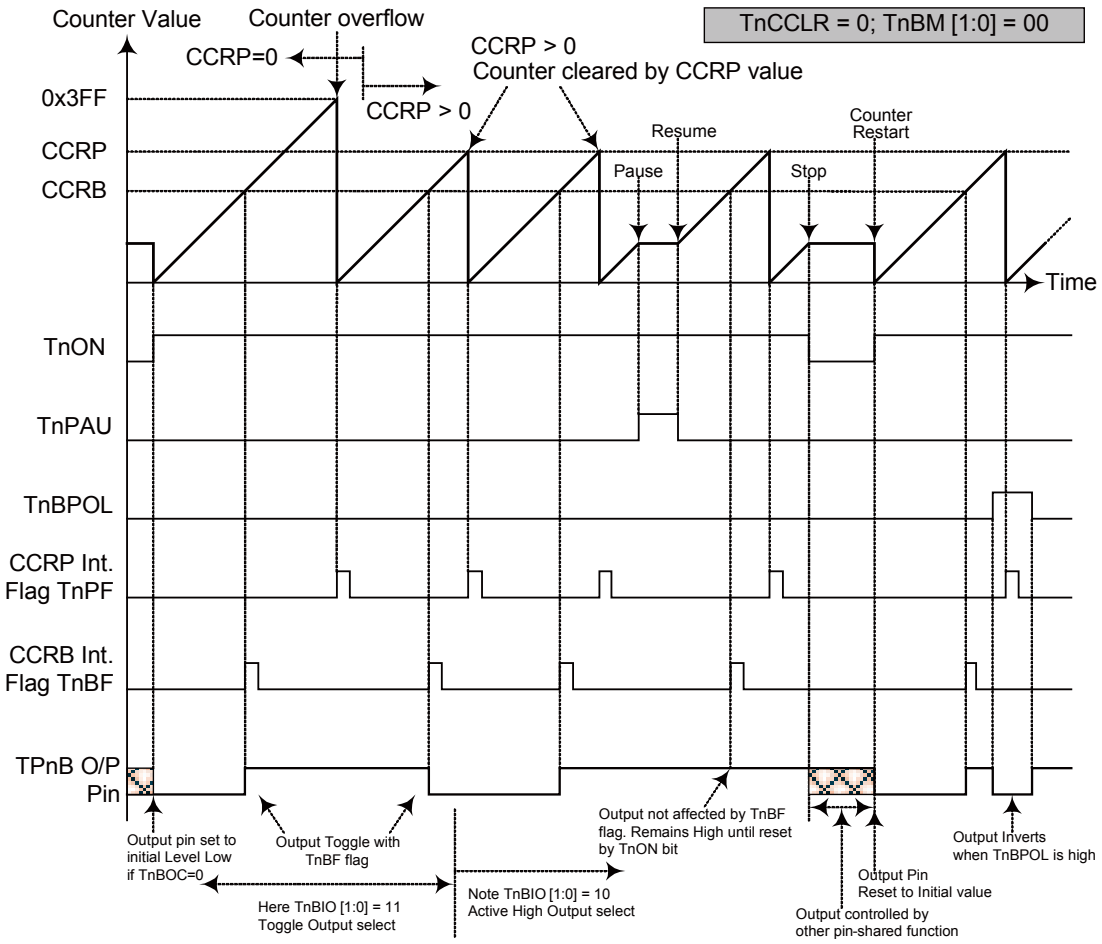
ETM CCRA Compare Match Output Mode – TnCCLR=0

Note: 1. With TnCCLR=0, a Comparator P match will clear the counter

2. The TPnA output pin is controlled only by the TnAF flag

3. The output pin is reset to its initial state by a TnON bit rising edge

4. n=1 for HT69F40A/HT69F50A



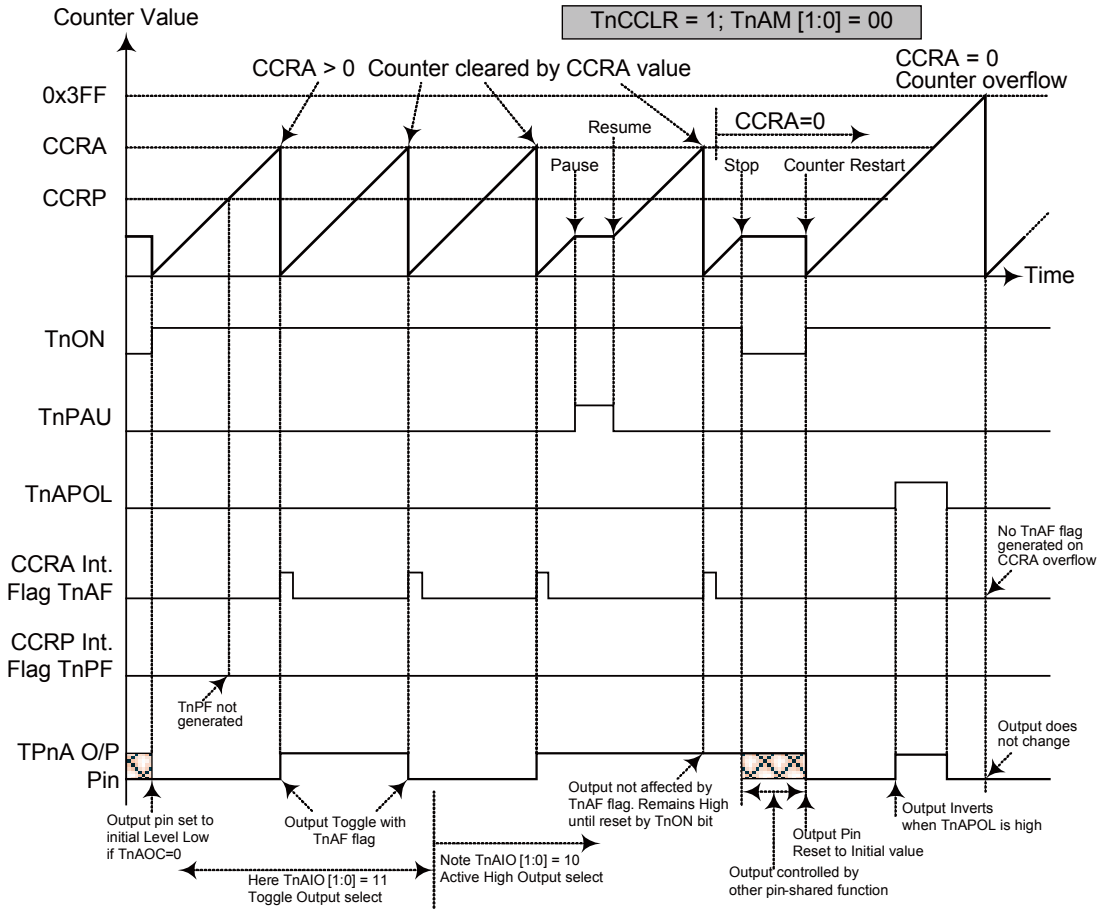
ETM CCRB Compare Match Output Mode – TnCCLR=0

Note: 1. With TnCCLR=0, a Comparator P match will clear the counter

2. The TPnB output pin is controlled only by the TnBF flag

3. The output pin is reset to its initial state by a TnON bit rising edge

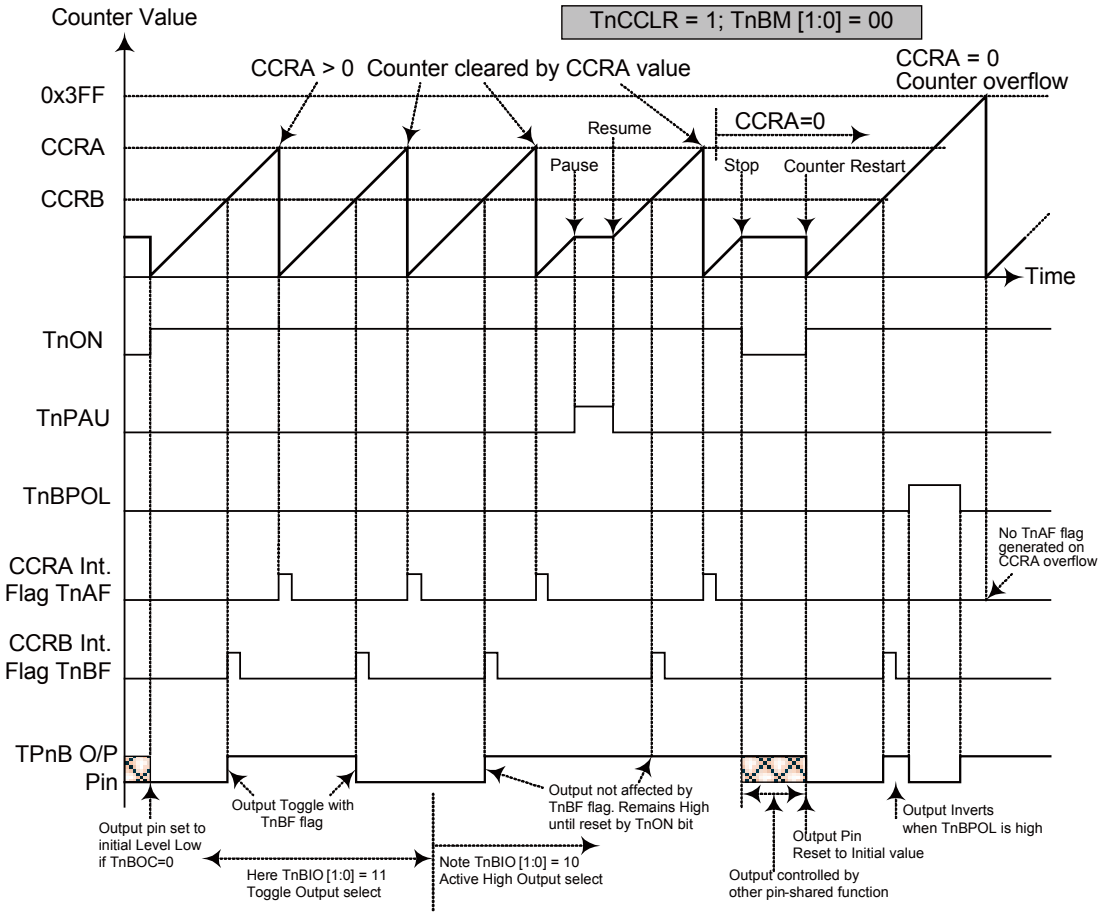
4. n=1 for HT69F40A/HT69F50A



ETM CCRA Compare Match Output Mode – TnCCLR=1

Note: 1. With TnCCLR=1, a Comparator A match will clear the counter

2. The TPnA output pin is controlled only by the TnAF flag
3. The TPnA output pin is reset to its initial state by a TnON bit rising edge
4. The TnPF flag is not generated when TnCCLR=1
5. n=1 for HT69F40A/HT69F50A



ETM CCRB Compare Match Output Mode – TnCCLR=1

Note: 1. With TnCCLR=1, a Comparator A match will clear the counter
2. The TPnB output pin is controlled only by the TnBF flag
3. The TPnB output pin is reset to its initial state by a TnON bit rising edge
4. The TnPF flag is not generated when TnCCLR=1
5. n=1 for HT69F40A/HT69F50A

Timer/Counter Mode

To select this mode, bits TnAM1, TnAM0 and TnBM1, TnBM0 in the TMnC1 and TMnC2 registers should all be set high. The Timer/Counter Mode operates in an identical way to the Compare Match Output Mode generating the same interrupt flags. The exception is that in the Timer/Counter Mode the TM output pin is not used. Therefore the above description and Timing Diagrams for the Compare Match Output Mode can be used to understand its function. As the TM output pin is not used in this mode, the pin can be used as a normal I/O pin or other pin-shared functions.

PWM Output Mode

To select this mode, the required bit pairs, TnAM1, TnAM0 and TnBM1, TnBM0 in the TMnC1 and TMnC2 registers should be set to 10 respectively and also the TnAIO1, TnAIO0 and TnBIO1, TnBIO0 bits should be set to 10 respectively. The PWM function within the TM is useful for applications which require functions such as motor control, heating control, illumination control etc. By providing a signal of fixed frequency but of varying duty cycle on the TM output pin, a square wave AC waveform can be generated with varying equivalent DC RMS values.

As both the period and duty cycle of the PWM waveform can be controlled, the choice of generated waveform is extremely flexible. In the PWM mode, the TnCCLR bit is used to determine in which way the PWM period is controlled. With the TnCCLR bit set high, the PWM period can be finely controlled using the CCRA registers. In this case the CCRB registers are used to set the PWM duty value (for TPnB_x output pins). The CCRP bits are not used and TPnA output pin is not used. The PWM output can only be generated on the TPnB_x output pins. With the TnCCLR bit cleared to zero, the PWM period is set using one of the eight values of the three CCRP bits, in multiples of 128. Now both CCRA and CCRB registers can be used to setup different duty cycle values to provide dual PWM outputs on their relative TPnA and TPnB_x pins.

The TnPWM1 and TnPWM0 bits determine the PWM alignment type, which can be either edge or centre type. In edge alignment, the leading edge of the PWM signals will all be generated concurrently when the counter is reset to zero. With all power currents switching on at the same time, this may give rise to problems in higher power applications. In centre alignment the centre of the PWM active signals will occur sequentially, thus reducing the level of simultaneous power switching currents.

Interrupt flags, one for each of the CCRA, CCRB and CCRP, will be generated when a compare match occurs from the Comparator A, Comparator B or Comparator P. The TnAOC and TnBOC bits in the TMnC1 and TMnC2 register are used to select the required polarity of the PWM waveform while the two TnAIO1, TnAIO0 and TnBIO1, TnBIO0 bits pairs are used to enable the PWM output or to force the TM output pin to a fixed high or low level. The TnAPOL and TnBPOL bit are used to reverse the polarity of the PWM output waveform.

ETM, PWM Mode, Edge-aligned Mode, TnCCLR=0

CCRP	001b	010b	011b	100b	101b	110b	111b	000b
Period	128	256	384	512	640	768	896	1024
A Duty	CCRA							
B Duty	CCRB							

If f_{SYS}=12MHz, TM clock source select f_{SYS}/4, CCRP=100b, CCRA=128 and CCRB=256,

The TPnA PWM output frequency=(f_{SYS}/4)/512=f_{SYS}/2048=5.8594kHz, duty=128/512=25%.

The TPnB_x PWM output frequency=(f_{SYS}/4)/512=f_{SYS}/2048=5.8594kHz, duty=256/512=50%.

If the Duty value defined by CCRA or CCRB register is equal to or greater than the Period value, then the PWM output duty is 100%.

ETM, PWM Mode, Edge-aligned Mode, TnCCLR=1

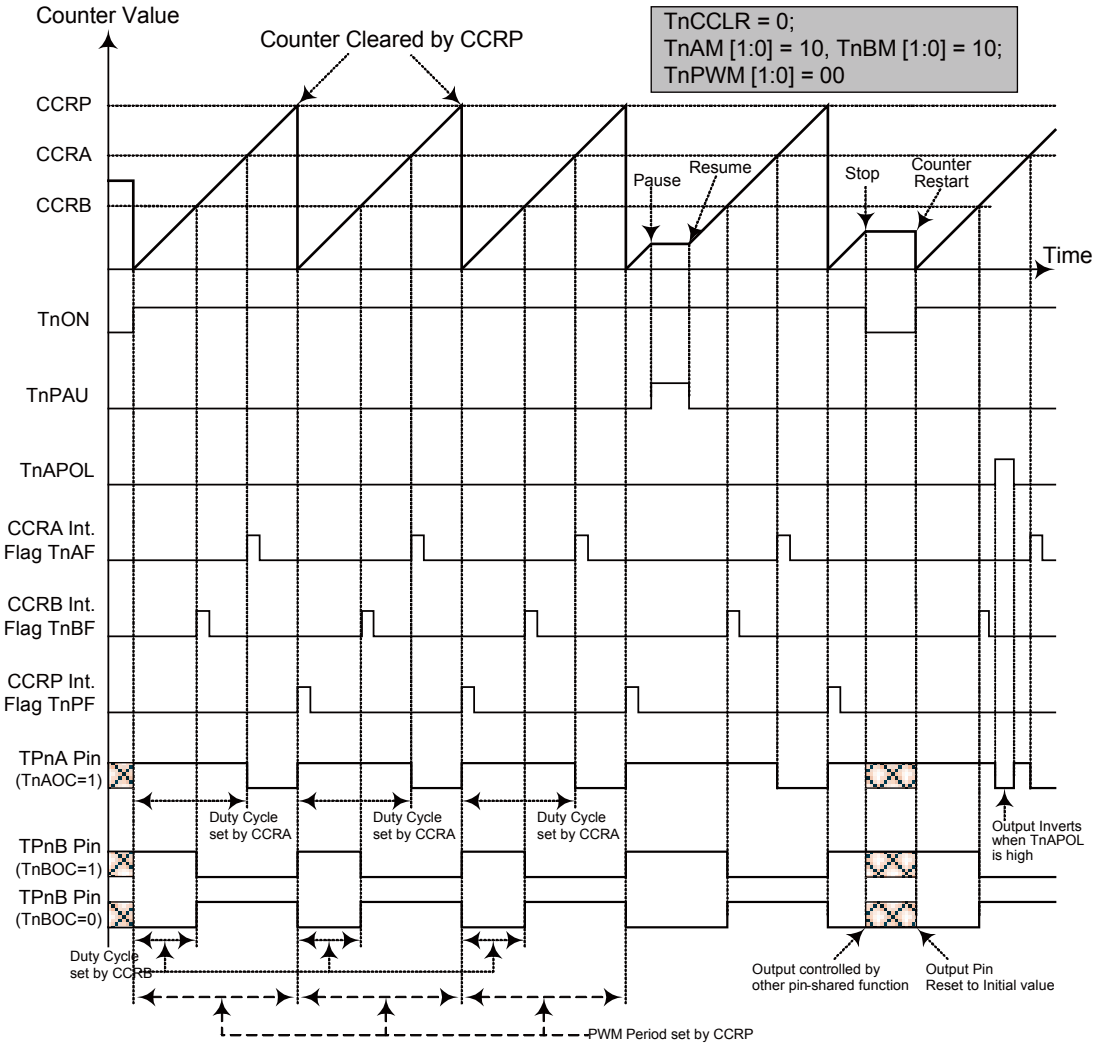
CCRA	1	2	3	511	512	1021	1022	1023
Period	1	2	3	511	512	1021	1022	1023
B Duty	CCRB							

ETM, PWM Mode, Center-aligned Mode, TnCCLR=0

CCRA	001b	010b	011b	100b	101b	110b	111b	000b
Period	256	512	768	1024	1280	1536	1792	2046
A Duty	(CCRA×2)-1							
B Duty	(CCRB×2)-1							

ETM, PWM Mode, Center-aligned Mode, TnCCLR=1

CCRA	1	2	3	511	512	1021	1022	1023
Period	2	4	6	1022	1024	2042	2044	2046
B Duty	(CCRB×2)-1							



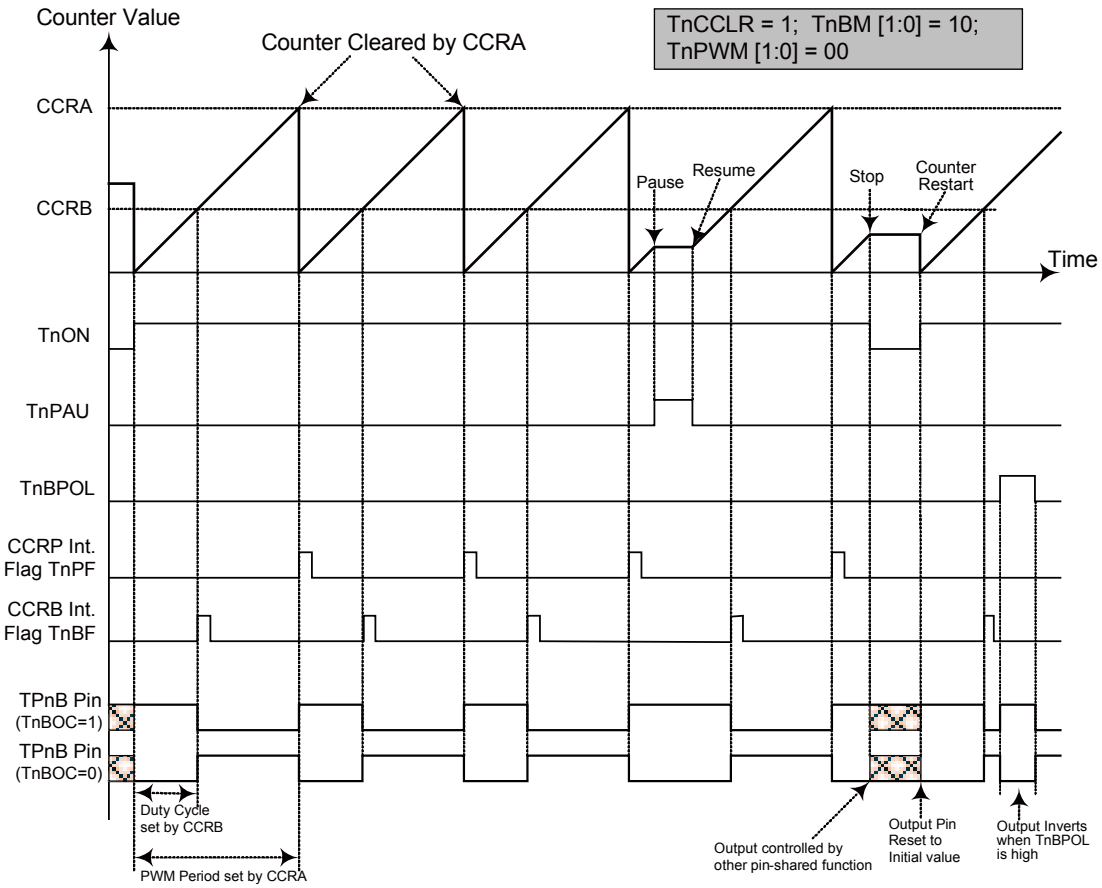
ETM PWM Mode – Edge Aligned

Note: 1. Here TnCCLR=0 therefore CCRP clears the counter and determines the PWM period

2. The internal PWM function continues running even when TnAIO [1:0] (or TnBIO [1:0])=00 or 01

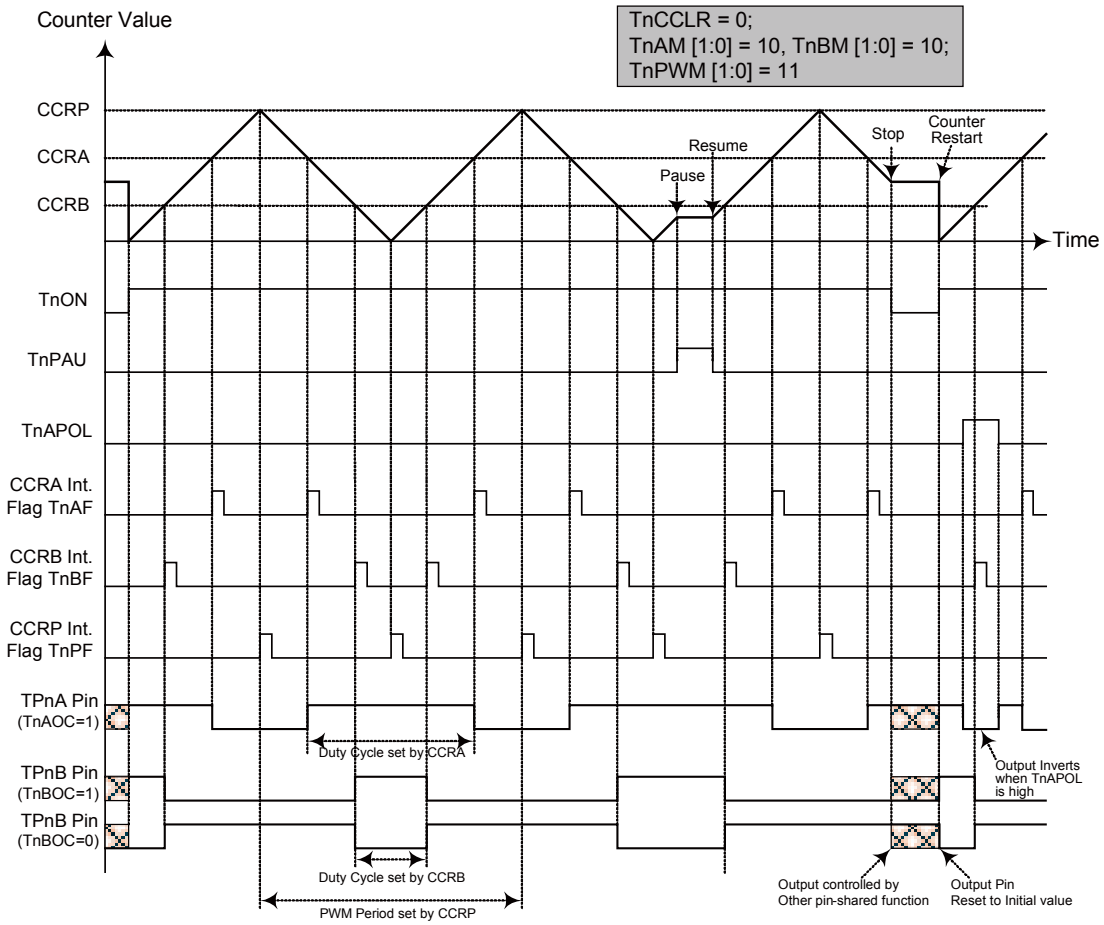
3. CCRA controls the TPnA PWM duty and CCRB controls the TPnB PWM duty

4. n=1 for HT69F40A/HT69F50A



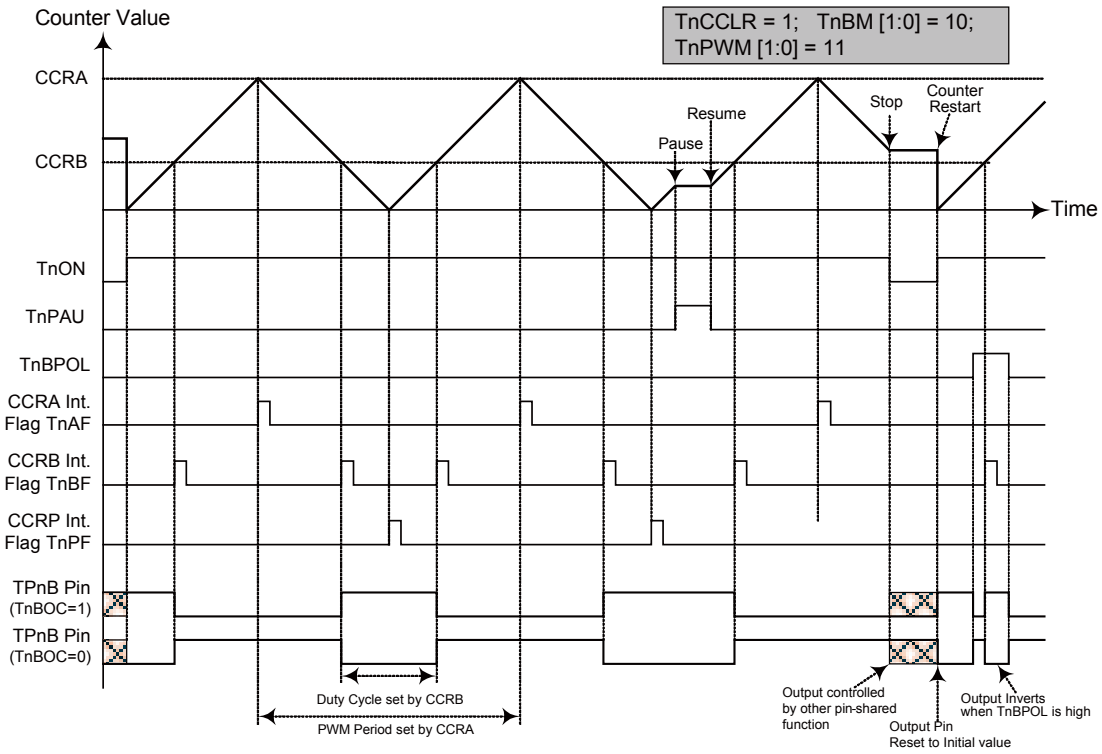
ETM PWM Mode – Edge Aligned

Note: 1. Here TnCCLR=1 therefore CCRA clears the counter and determines the PWM period
2. The internal PWM function continues running even when TnBIO [1:0]=00 or 01
3. The CCRA controls the TPnB PWM period and CCRB controls the TPnB PWM duty
4. Here the TM pin control register should not enable the TPnA pin as a TM output pin.
5. n=1 for HT69F40A/HT69F50A



ETM PWM Mode – Centre Aligned

Note: 1. Here TnCCLR=0 therefore CCRP clears the counter and determines the PWM period
2. TnPWM [1:0]=11 therefore the PWM is centre aligned
3. The internal PWM function continues running even when TnAIO [1:0] (or TnBIO [1:0])=00 or 01
4. CCRA controls the TPnA PWM duty and CCRB controls the TPnB PWM duty
5. CCRP will generate an interrupt request when the counter decrements to its zero value
6. n=1 for HT69F40A/HT69F50A



ETM PWM Mode – Centre Aligned

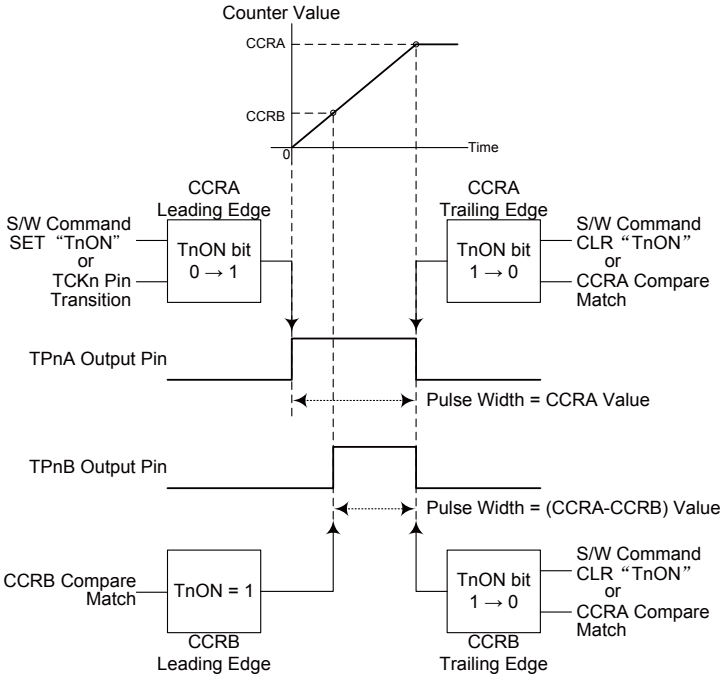
Note: 1. Here TnCCLR=1 therefore CCRA clears the counter and determines the PWM period
2. TnPWM [1:0]=11 therefore the PWM is centre aligned
3. The internal PWM function continues running even when TnBIO [1:0]=00 or 01
4. CCRA controls the TPnB PWM period and CCRB controls the TPnB PWM duty
5. CCRP will generate an interrupt request when the counter decrements to its zero value
6. n=1 for HT69F40A/HT69F50A

Single Pulse Output Mode

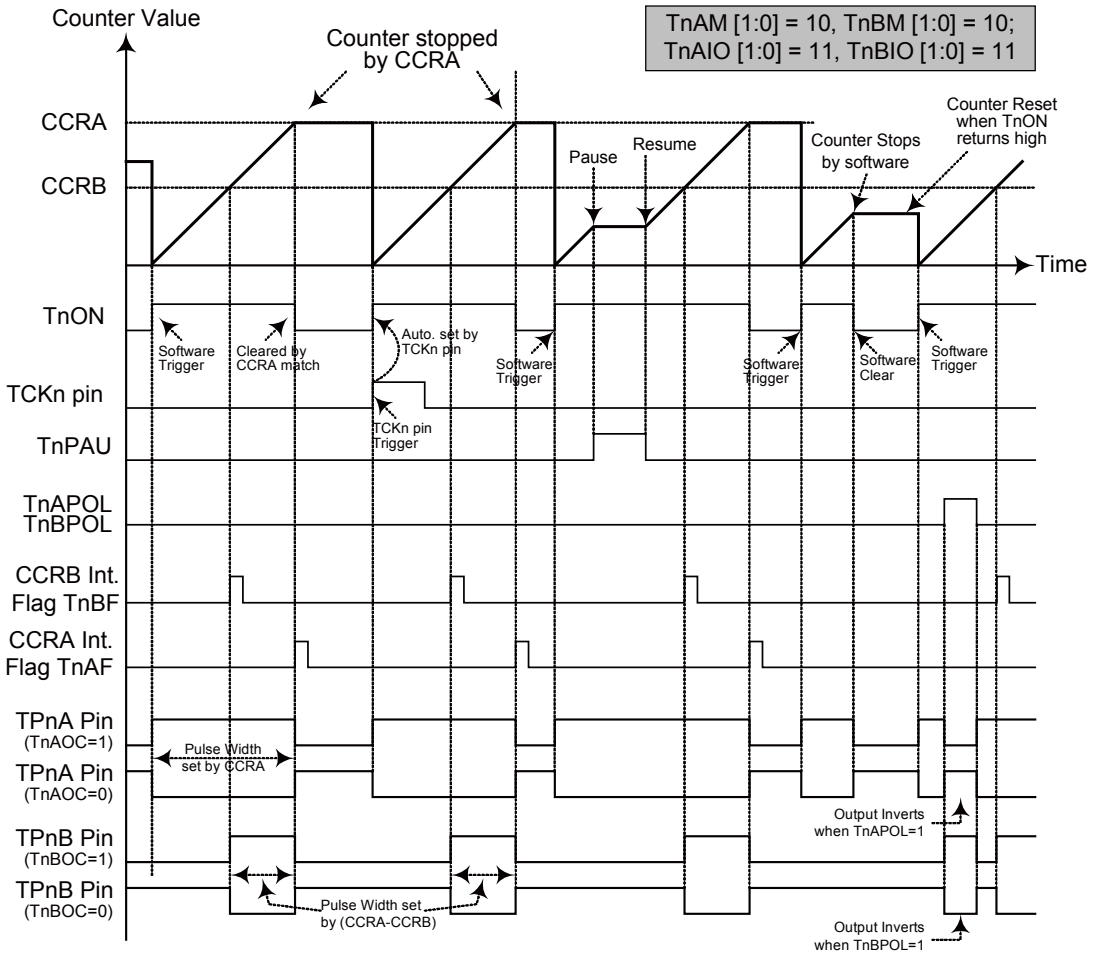
To select this mode, the required bit pairs, TnAM1, TnAM0 and TnBM1, TnBM0 should be set to 10 respectively and also the corresponding TnAIO1, TnAIO0 and TnBIO1, TnBIO0 bits should be set to 11 respectively. The Single Pulse Output Mode, as the name suggests, will generate a single shot pulse on the TM output pin.

The trigger for the pulse TPnA output leading edge is a low to high transition of the TnON bit, which can be implemented using the application program. The trigger for the pulse TPnB_x output leading edge is a compare match from Comparator B, which can be implemented using the application program. However in the Single Pulse Mode, the TnON bit can also be made to automatically change from low to high using the external TCKn pin, which will in turn initiate the Single Pulse output of TPnA. When the TnON bit transitions to a high level, the counter will start running and the pulse leading edge of TPnA will be generated. The TnON bit should remain high when the pulse is in its active state. The generated pulse trailing edge of TPnA and TPnB_x will be generated when the TnON bit is cleared to zero, which can be implemented using the application program or when a compare match occurs from Comparator A.

However a compare match from Comparator A will also automatically clear the TnON bit and thus generate the Single Pulse output trailing edge of TPnA and TPnB_x. In this way the CCRA value can be used to control the pulse width of TPnA. The (CCRA-CCRB) value can be used to control the pulse width of TPnB_x. A compare match from Comparator A and Comparator B will also generate TM interrupts. The counter can only be reset back to zero when the TnON bit changes from low to high when the counter restarts. In the Single Pulse Mode CCRP is not used. The TnCCLR bit is also not used.



Single Pulse Generation



ETM – Single Pulse Mode

Note: 1. Counter stopped by CCRA

2. CCRP is not used

3. The pulse triggered by the TCKn pin or by setting the TnON bit high

4. A TCKn pin active edge will automatically set the TnON bit high

5. In the Single Pulse Mode, TnAIO [1:0] and TnBIO [1:0] must be set to "11" and can not be changed

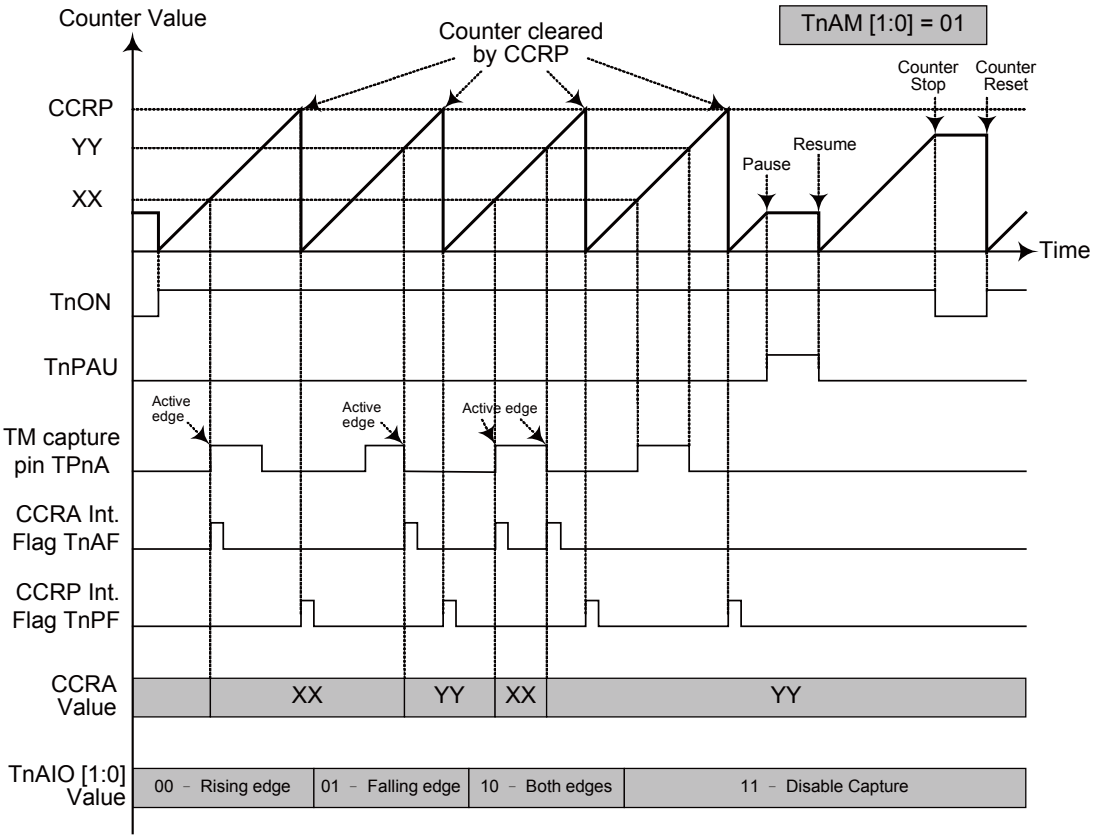
6. n=1 for HT69F40A/HT69F50A

Capture Input Mode

To select this mode bits TnAM1, TnAM0 and TnBM1, TnBM0 in the TMnC1 and TMnC2 registers should be set to 01 respectively. This mode enables external signals to capture and store the present value of the internal counter and can therefore be used for applications such as pulse width measurements. The external signal is supplied on the TPnA and TPnB_0, TPnB_1, TPnB_2 pins, whose active edge can be either a rising edge, a falling edge or both rising and falling edges; the active edge transition type is selected using the TnAIO1, TnAIO0 and TnBIO1, TnBIO0 bits in the TMnC1 and TMnC2 registers. The counter is started when the TnON bit changes from low to high which is initiated using the application program.

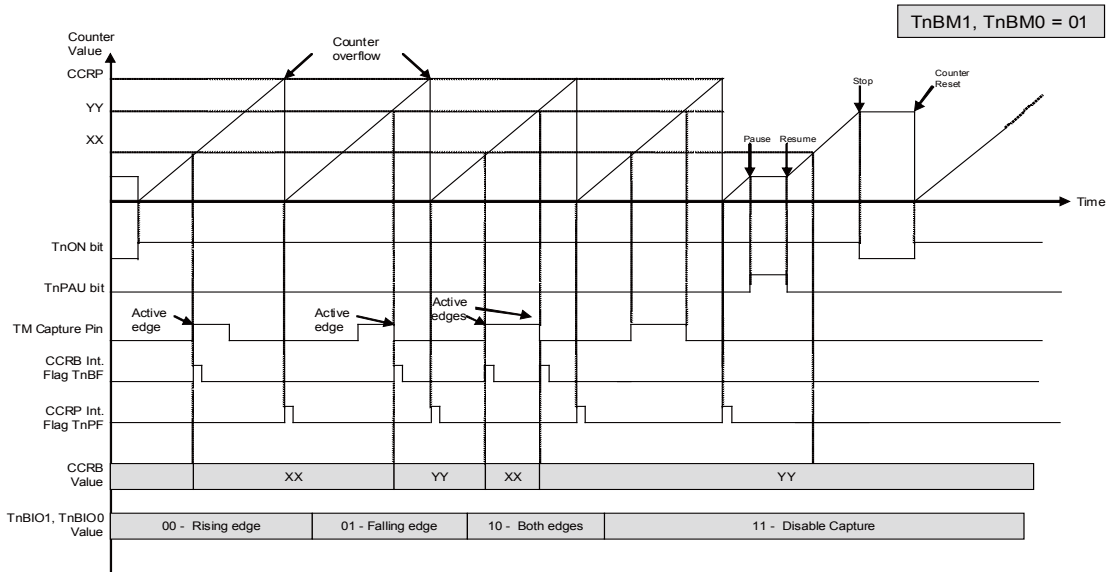
When the required edge transition appears on the TPnA and TPnB_0, TPnB_1, TPnB_2 pins the present value in the counter will be latched into the CCRA and CCRB registers and a TM interrupt generated. Irrespective of what events occur on the TPnA and TPnB_0, TPnB_1, TPnB_2 pins the counter will continue to free run until the TnON bit changes from high to low. When a CCRP compare match occurs the counter will reset back to zero; in this way the CCRP value can be used to control the maximum counter value. When a CCRP compare match occurs from Comparator P, a TM interrupt will also be generated. Counting the number of overflow interrupt signals from the CCRP can be a useful method in measuring long pulse widths. The TnAIO1, TnAIO0 and TnBIO1, TnBIO0 bits can select the active trigger edge on the TPnA and TPnB_0, TPnB_1, TPnB_2 pins to be a rising edge, falling edge or both edge types. If the TnAIO1, TnAIO0 and TnBIO1, TnBIO0 bits are both set high, then no capture operation will take place irrespective of what happens on the TPnA and TPnB_0, TPnB_1, TPnB_2 pins, however it must be noted that the counter will continue to run.

As the TPnA and TPnB_0, TPnB_1, TPnB_2 pins are pin shared with other functions, care must be taken if the TM is in the Capture Input Mode. This is because if the pin is setup as an output, then any transitions on this pin may cause an input capture operation to be executed. The TnCCLR, TnAOC, TnBOC, TnAPOL and TnBPOL bits are not used in this mode.



ETM CCRA Capture Input Mode

Note: 1. TnAM [1:0]=01 and active edge set by the TnAIO [1:0] bits
2. The TM Capture input pin active edge transfers the counter value to CCRA
3. TnCCLR bit not used
4. No output function -- TnAOC and TnAPOL bits not used
5. CCRP determines the counter value and the counter has a maximum count value when CCRP is equal to zero
6. n=1 for HT69F40A/HT69F50A



ETM CCRB Capture Input Mode

Note: 1. TnBM [1:0]=01 and active edge set by the TnBIO [1:0] bits
2. The TM Capture input pin active edge transfers the counter value to CCRB
3. TnCCLR bit not used
4. No output function -- TnBOC and TnBPOL bits not used
5. CCRP determines the counter value and the counter has a maximum count value when CCRP is equal to zero
6. n=1 for HT69F40A/HT69F50A

Interrupts

Interrupts are an important part of any microcontroller system. When an external event or an internal function such as a Timer Module requires microcontroller attention, their corresponding interrupt will enforce a temporary suspension of the main program allowing the microcontroller to direct attention to their respective needs. The device contains several external interrupt and internal interrupts functions. The external interrupts are generated by the action of the external INT0~INT1 pins, while the internal interrupts are generated by various internal functions such as the TMs, Time Base and LVD.

Interrupt Registers

Overall interrupt control, which basically means the setting of request flags when certain microcontroller conditions occur and the setting of interrupt enable bits by the application program, is controlled by a series of registers, located in the Special Purpose Data Memory, as shown in the accompanying table. The number of registers depends upon the device chosen but fall into three categories. The first is the INTC0~INTC1 registers which setup the primary interrupts, the second is the MFI0~MFI2 registers which setup the Multi-function interrupts. Finally there is an INTEG register to setup the external interrupt trigger edge type.

Each register contains a number of enable bits to enable or disable individual registers as well as interrupt flags to indicate the presence of an interrupt request. The naming convention of these follows a specific pattern. First is listed an abbreviated interrupt type, then the (optional) number of that interrupt followed by either an "E" for enable/disable bit or "F" for request flag.

Function	Enable Bit	Request Flag	Notes
Global	EMI	—	—
INTn Pin	INTnE	INTnF	n=0~1
Time Base	TBnE	TBnF	n=0~1
Multi-function	MFnE	MFnF	n=0~2
LVD	LVE	LVF	—
EEPROM	DEE	DEF	—
TM	TnPE	TnPF	n=0~2
	TnAE	TnAF	n=0~2
	TnBE	TnBF	n=1

Interrupt Register Bit Naming Conventions

Interrupt Register Contents

- **HT69F30A**

Name	Bit							
	7	6	5	4	3	2	1	0
INTEG	—	—	—	—	INT1S1	INT1S0	INT0S1	INT0S0
INTC0	—	TB0F	INT1F	INT0F	TB0E	INT1E	INT0E	EMI
INTC1	MF2F	MF1F	MF0F	TB1F	MF2E	MF1E	MF0E	TB1E
MFI0	—	—	T0AF	T0PF	—	—	T0AE	T0PE
MFI1	—	—	T1AF	T1PF	—	—	T1AE	T1PE
MFI2	—	—	DEF	LVF	—	—	DEE	LVE

- **HT69F40A/HT69F50A**

Name	Bit							
	7	6	5	4	3	2	1	0
INTEG	—	—	—	—	INT1S1	INT1S0	INT0S1	INT0S0
INTC0	—	TB0F	INT1F	INT0F	TB0E	INT1E	INT0E	EMI
INTC1	MF2F	MF1F	MF0F	TB1F	MF2E	MF1E	MF0E	TB1E
MFI0	T2AF	T2PF	T0AF	T0PF	T2AE	T2PE	T0AE	T0PE
MFI1	—	T1BF	T1AF	T1PF	—	T1BE	T1AE	T1PE
MFI2	—	—	DEF	LVF	—	—	DEE	LVE

INTEG Register

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	INT1S1	INT1S0	INT0S1	INT0S0
R/W	—	—	—	—	R/W	R/W	R/W	R/W
POR	—	—	—	—	0	0	0	0

Bit 7~4 Unimplemented, read as "0"

Bit 3~2 **INT1S1, INT1S0:** interrupt edge control for INT1 pin
00: Disable
01: Rising edge
10: Falling edge
11: Rising and falling edges

Bit 1~0 **INT0S1, INT0S0:** interrupt edge control for INT0 pin
00: Disable
01: Rising edge
10: Falling edge
11: Rising and falling edge

INTC0 Register

Bit	7	6	5	4	3	2	1	0
Name	—	TB0F	INT1F	INT0F	TB0E	INT1E	INT0E	EMI
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	0	0	0	0	0	0	0

Bit 7 Unimplemented, read as "0"

Bit 6 **TB0F:** Time Base 0 interrupt request flag
0: No request
1: Interrupt request

Bit 5 **INT1F:** INT1 interrupt request flag
0: No request
1: Interrupt request

Bit 4 **INT0F:** INT0 interrupt request flag
0: No request
1: Interrupt request

Bit 3 **TB0E:** Time Base 0 interrupt control
0: Disable
1: Enable

Bit 2 **INT1E:** INT1 interrupt control
0: Disable
1: Enable

Bit 1 **INT0E:** INT0 interrupt control
0: Disable
1: Enable

Bit 0 **EMI:** Global interrupt control
0: Disable
1: Enable

INTC1 Register

Bit	7	6	5	4	3	2	1	0
Name	MF2F	MF1F	MF0F	TB1F	MF2E	MF1E	MF0E	TB1E
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **MF2F:** Multi-function interrupt 2 request flag
0: No request
1: Interrupt request

Bit 6 **MF1F:** Multi-function interrupt 1 request flag
0: No request
1: Interrupt request

Bit 5 **MF0F:** Multi-function interrupt 0 request flag
0: No request
1: Interrupt request

Bit 4 **TB1F:** Time Base 1 interrupt request flag
0: No request
1: Interrupt request

Bit 3 **MF2E:** Multi-function interrupt 2 control
0: Disable
1: Enable

Bit 2 **MF1E:** Multi-function interrupt 1 control
0: Disable
1: Enable

Bit 1 **MF0E:** Multi-function interrupt 0 control
0: Disable
1: Enable

Bit 0 **TB1E:** Time Base 1 interrupt control
0: Disable
1: Enable

MFI0 Register – HT69F30A

Bit	7	6	5	4	3	2	1	0
Name	—	—	T0AF	T0PF	—	—	T0AE	T0PE
R/W	—	—	R/W	R/W	—	—	R/W	R/W
POR	—	—	0	0	—	—	0	0

Bit 7~6 Unimplemented, read as "0"

Bit 5 **T0AF:** TM0 Comparator A match interrupt request flag
0: No request
1: Interrupt request

Bit 4 **T0PF:** TM0 Comparator P match interrupt request flag
0: No request
1: Interrupt request

Bit 3~2 Unimplemented, read as "0"

Bit 1 **T0AE:** TM0 Comparator A match interrupt control
0: Disable
1: Enable

Bit 0 **T0PE:** TM0 Comparator P match interrupt control
0: Disable
1: Enable

MFI0 Register – HT69F40A/HT69F50A

Bit	7	6	5	4	3	2	1	0
Name	T2AF	T2PF	T0AF	T0PF	T2AE	T2PE	T0AE	T0PE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **T2AF:** TM2 Comparator A match interrupt request flag
0: No request
1: Interrupt request

Bit 6 **T2PF:** TM2 Comparator P match interrupt request flag
0: No request
1: Interrupt request

Bit 5 **T0AF:** TM0 Comparator A match interrupt request flag
0: No request
1: Interrupt request

Bit 4 **T0PF:** TM0 Comparator P match interrupt request flag
0: No request
1: Interrupt request

Bit 3 **T2AE:** TM2 Comparator A match interrupt control
0: Disable
1: Enable

Bit 2 **T2PE:** TM2 Comparator P match interrupt control
0: Disable
1: Enable

Bit 1 **T0AE:** TM0 Comparator A match interrupt control
0: Disable
1: Enable

Bit 0 **T0PE:** TM0 Comparator P match interrupt control
0: Disable
1: Enable

MFI1 Register – HT69F30A

Bit	7	6	5	4	3	2	1	0
Name	—	—	T1AF	T1PF	—	—	T1AE	T1PE
R/W	—	—	R/W	R/W	—	—	R/W	R/W
POR	—	—	0	0	—	—	0	0

Bit 7~6 Unimplemented, read as "0"

Bit 5 **T1AF:** TM1 Comparator A match interrupt request flag
0: No request
1: Interrupt request

Bit 4 **T1PF:** TM1 Comparator P match interrupt request flag
0: No request
1: Interrupt request

Bit 3~2 Unimplemented, read as "0"

Bit 1 **T1AE:** TM1 Comparator A match interrupt control
0: Disable
1: Enable

Bit 0 **T1PE:** TM1 Comparator P match interrupt control
0: Disable
1: Enable

MFI1 Register – HT69F40A/HT69F50A

Bit	7	6	5	4	3	2	1	0
Name	—	T1BF	T1AF	T1PF	—	T1BE	T1AE	T1PE
R/W	—	R/W	R/W	R/W	—	R/W	R/W	R/W
POR	—	0	0	0	—	0	0	0

Bit 7 Unimplemented, read as "0"

Bit 6 **T1BF:** TM1 Comparator B match interrupt request flag
0: No request
1: Interrupt request

Bit 5 **T1AF:** TM1 Comparator A match interrupt request flag
0: No request
1: Interrupt request

Bit 4 **T1PF:** TM1 Comparator P match interrupt request flag
0: No request
1: Interrupt request

Bit 3 Unimplemented, read as "0"

Bit 2 **T1BE:** TM1 Comparator B match interrupt control
0: Disable
1: Enable

Bit 1 **T1AE:** TM1 Comparator A match interrupt control
0: Disable
1: Enable

Bit 0 **T1PE:** TM1 Comparator P match interrupt control
0: Disable
1: Enable

MFI2 Register

Bit	7	6	5	4	3	2	1	0
Name	—	—	DEF	LVF	—	—	DEE	LVE
R/W	—	—	R/W	R/W	—	—	R/W	R/W
POR	—	—	0	0	—	—	0	0

Bit 7~6 Unimplemented, read as "0"

Bit 5 **DEF:** Data EEPROM interrupt request flag
0: No request
1: Interrupt request

Bit 4 **LVF:** LVD interrupt request flag
0: No request
1: Interrupt request

Bit 3~2 Unimplemented, read as "0"

Bit 1 **DEE:** Data EEPROM Interrupt Control
0: Disable
1: Enable

Bit 0 **LVE:** LVD Interrupt Control
0: Disable
1: Enable

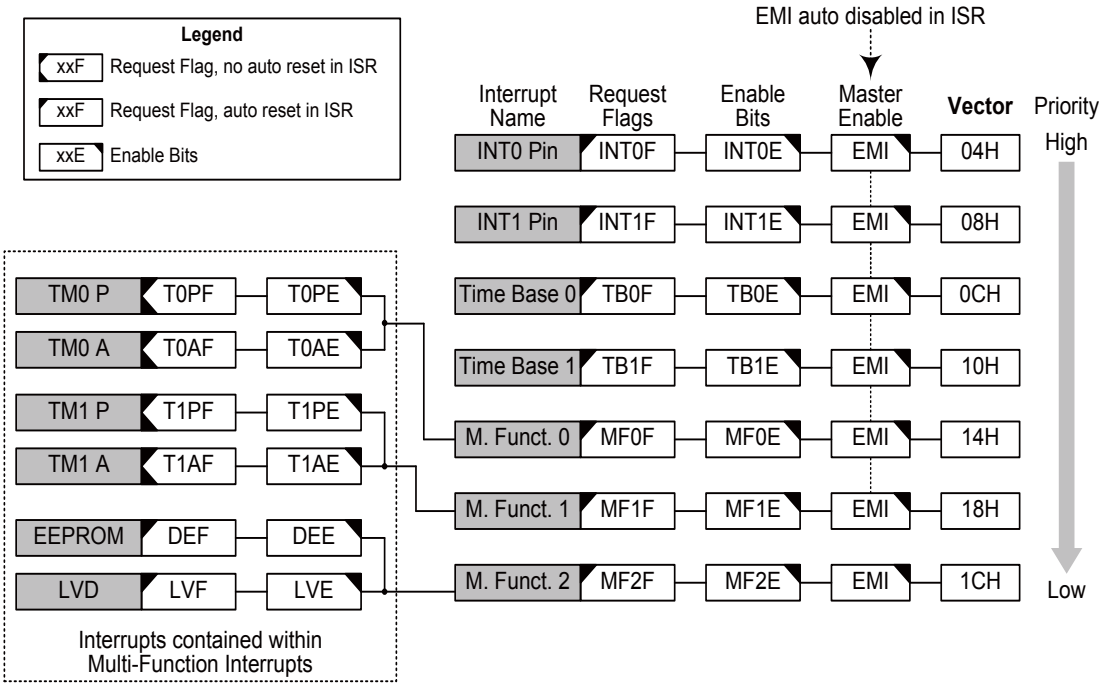
Interrupt Operation

When the conditions for an interrupt event occur, such as a TM Comparator P, Comparator A or Comparator B match etc, the relevant interrupt request flag will be set. Whether the request flag actually generates a program jump to the relevant interrupt vector is determined by the condition of the interrupt enable bit. If the enable bit is set high then the program will jump to its relevant vector; if the enable bit is zero then although the interrupt request flag is set an actual interrupt will not be generated and the program will not jump to the relevant interrupt vector. The global interrupt enable bit, if cleared to zero, will disable all interrupts.

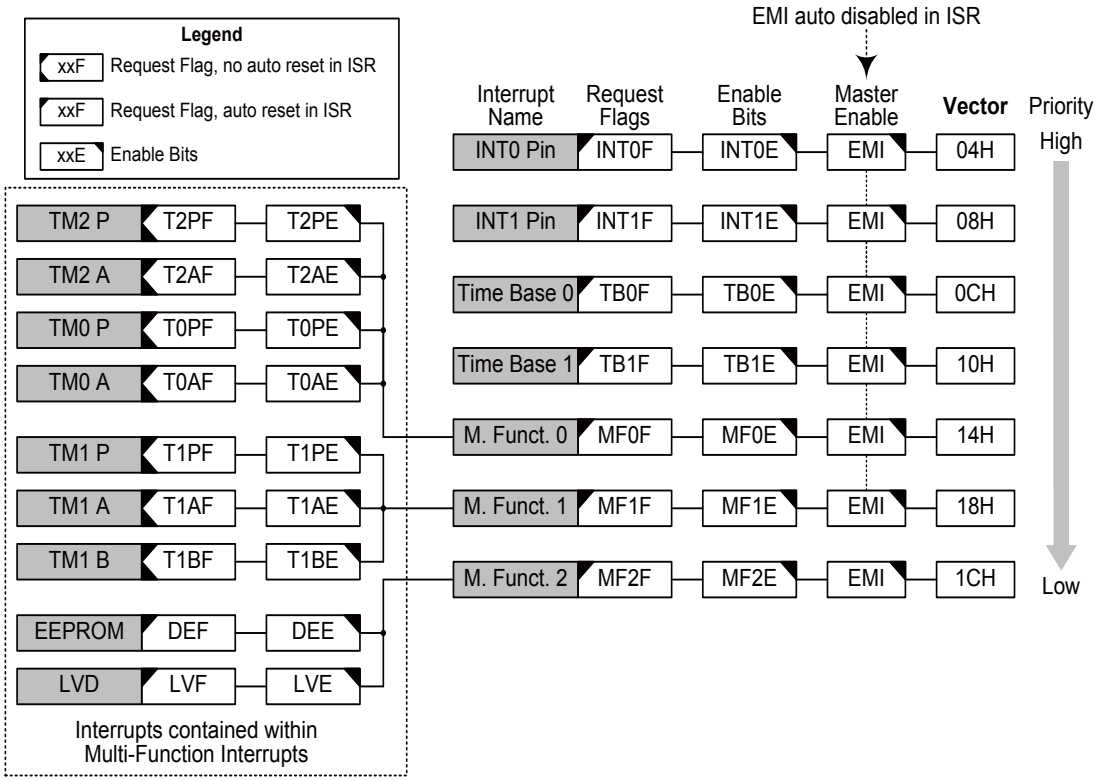
When an interrupt is generated, the Program Counter, which stores the address of the next instruction to be executed, will be transferred onto the stack. The Program Counter will then be loaded with a new address which will be the value of the corresponding interrupt vector. The microcontroller will then fetch its next instruction from this interrupt vector. The instruction at this vector will usually be a “JMP” which will jump to another section of program which is known as the interrupt service routine. Here is located the code to control the appropriate interrupt. The interrupt service routine must be terminated with a "RETI" , which retrieves the original Program Counter address from the stack and allows the microcontroller to continue with normal execution at the point where the interrupt occurred.

The various interrupt enable bits, together with their associated request flags, are shown in the accompanying diagrams with their order of priority. Some interrupt sources have their own individual vector while others share the same multi-function interrupt vector. Once an interrupt subroutine is serviced, all the other interrupts will be blocked, as the global interrupt enable bit, EMI bit will be cleared automatically. This will prevent any further interrupt nesting from occurring. However, if other interrupt requests occur during this interval, although the interrupt will not be immediately serviced, the request flag will still be recorded.

If an interrupt requires immediate servicing while the program is already in another interrupt service routine, the EMI bit should be set after entering the routine, to allow interrupt nesting. If the stack is full, the interrupt request will not be acknowledged, even if the related interrupt is enabled, until the Stack Pointer is decremented. If immediate service is desired, the stack must be prevented from becoming full. In case of simultaneous requests, the accompanying diagram shows the priority that is applied. All of the interrupt request flags when set will wake-up the device if it is in SLEEP or IDLE Mode, however to prevent a wake-up from occurring the corresponding flag should be set before the device is in SLEEP or IDLE Mode.



Interrupt Structure – HT69F30A



Interrupt Structure – HT69F40A/HT69F50A

External Interrupt

The external interrupts are controlled by signal transitions on the pins INT0~INT1. An external interrupt request will take place when the external interrupt request flags, INT0F~INT1F, are set, which will occur when a transition, whose type is chosen by the edge select bits, appears on the external interrupt pins. To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EMI, and respective external interrupt enable bit, INT0E~INT1E, must first be set. Additionally the correct interrupt edge type must be selected using the INTEG register to enable the external interrupt function and to choose the trigger edge type. As the external interrupt pins are pin-shared with I/O pins, they can only be configured as external interrupt pins if their external interrupt enable bit in the corresponding interrupt register has been set. The pin must also be setup as an input by setting the corresponding bit in the port control register. When the interrupt is enabled, the stack is not full and the correct transition type appears on the external interrupt pin, a subroutine call to the external interrupt vector, will take place. When the interrupt is serviced, the external interrupt request flags, INT0F~INT1F, will be automatically reset and the EMI bit will be automatically cleared to disable other interrupts. Note that any pull-high resistor selections on the external interrupt pins will remain valid even if the pin is used as an external interrupt input.

The INTEG register is used to select the type of active edge that will trigger the external interrupt. A choice of either rising or falling or both edge types can be chosen to trigger an external interrupt. Note that the INTEG register can also be used to disable the external interrupt function.

Multi-function Interrupt

Within this device there are up to three Multi-function interrupts. Unlike the other independent interrupts, these interrupts have no independent source, but rather are formed from other existing interrupt sources, namely the TM Interrupts, EEPROM Interrupt and LVD interrupt.

A Multi-function interrupt request will take place when any of the Multi-function interrupt request flags, MFnF, are set. The Multi-function interrupt flags will be set when any of their included functions generate an interrupt request flag. To allow the program to branch to its respective interrupt vector address, when the Multi-function interrupt is enabled and the stack is not full, and either one of the interrupts contained within each of Multi-function interrupt occurs, a subroutine call to one of the Multi-function interrupt vectors will take place. When the interrupt is serviced, the related Multi-Function request flag, will be automatically reset and the EMI bit will be automatically cleared to disable other interrupts.

However, it must be noted that, although the Multi-function Interrupt flags will be automatically reset when the interrupt is serviced, the request flags from the original source of the Multi-function interrupts, namely the TM Interrupts, EEPROM Interrupt and LVD interrupt will not be automatically reset and must be manually reset by the application program.

Time Base Interrupt

The function of the Time Base Interrupt is to provide regular time signal in the form of an internal interrupt. It is controlled by the overflow signal from its internal timer. When this happens its interrupt request flag, TBnF, will be set. To allow the program to branch to its respective interrupt vector addresses, the global interrupt enable bit, EMI and Time Base enable bit, TBnE, must first be set. When the interrupt is enabled, the stack is not full and the Time Base overflows, a subroutine call to its respective vector location will take place. When the interrupt is serviced, the interrupt request flag, TBnF, will be automatically reset and the EMI bit will be cleared to disable other interrupts.

The purpose of the Time Base Interrupt is to provide an interrupt signal at fixed time periods. Its clock source, f_{TB}, originates from the internal clock source f_{SUB} or $f_{SYS}/4$. And then passes through a divider, the division ratio of which is selected by programming the appropriate bits in the TBC register to obtain longer interrupt periods whose value ranges. The clock source which in turn controls the Time Base interrupt period is selected using a bit in the TBC register.

TBC Register

Bit	7	6	5	4	3	2	1	0
Name	TBON	TBCK	TB11	TB10	LXTLP	TB02	TB01	TB00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	1	1	0	1	1	1

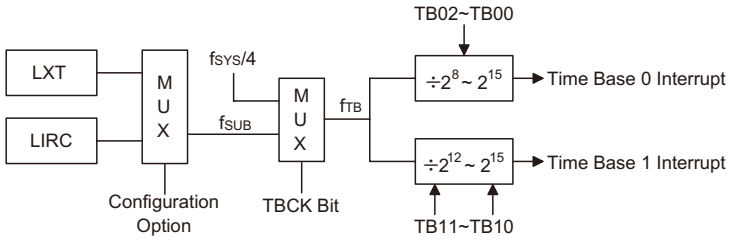
Bit 7 **TBON:** Time Base 0 and Time Base 1 Enable/Disable
0: Disable
1: Enable

Bit 6 **TBCK:** TB Clock f_{TB} Select
0: f_{SUB}
1: $f_{SYS}/4$

Bit 5~4 **TB11~TB10:** Time Base 1 Time-out Period Selection
00: $2^{12}/f_{TB}$
01: $2^{13}/f_{TB}$
10: $2^{14}/f_{TB}$
11: $2^{15}/f_{TB}$

Bit 3 **LXTLP:** LXT Low Power Control
0: Disable (LXT quick start-up)
1: Enable (LXT low power start-up)

Bit 2~0 **TB02~TB00:** Time Base 0 Time-out Period
000: $2^{8}/f_{TB}$
001: $2^{9}/f_{TB}$
010: $2^{10}/f_{TB}$
011: $2^{11}/f_{TB}$
100: $2^{12}/f_{TB}$
101: $2^{13}/f_{TB}$
110: $2^{14}/f_{TB}$
111: $2^{15}/f_{TB}$



Time Base Interrupt

EEPROM Interrupt

The EEPROM interrupt is contained within the Multi-function Interrupt. An EEPROM Interrupt request will take place when the EEPROM Interrupt request flag, DEF, is set, which occurs when an EEPROM write operation ends. To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EMI, and the EEPROM Interrupt enable bit, DEE, and Muti-function interrupt enable bits, must first be set. When the interrupt is enabled, the stack is not full and an EEPROM write operation ends, a subroutine call to the respective Multi-function Interrupt vector, will take place.When the EEPROM Interrupt is serviced, the EMI bit will be automatically cleared to disable other interrupts, however only the Multi-function interrupt request flag will be also automatically cleared. As the DEF flag will not be automatically cleared, it has to be cleared by the application program.

LVD Interrupt

The Low Voltage Detector Interrupt is contained within the Multi-function Interrupt. An LVD Interrupt request will take place when the LVD Interrupt request flag, LVF, is set, which occurs when the Low Voltage Detector function detects a low power supply voltage. To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EMI, Low Voltage Interrupt enable bit, LVE, and associated Multi-function interrupt enable bit, must first be set. When the interrupt is enabled, the stack is not full and a low voltage condition occurs, a subroutine call to the Multi-function Interrupt vector, will take place. When the Low Voltage Interrupt is serviced, the EMI bit will be automatically cleared to disable other interrupts, however only the Multi-function interrupt request flag will be also automatically cleared. As the LVF flag will not be automatically cleared, it has to be cleared by the application program.

TM Interrupts

The Compact and Standard Type TMs have two interrupts each, while the Enhanced Type TM has three interrupts. All of the TM interrupts are contained within the Multi-function Interrupts. For each of the Compact and Standard Type TMs there are two interrupt request flags TnPF and TnAF and two enable bits TnPE and TnAE. For the Enhanced Type TM there are three interrupt request flags TnPF, TnAF and TnBF and three enable bits TnPE, TnAE and TnBE. A TM interrupt request will take place when any of the TM request flags are set, a situation which occurs when a TM comparator P, A or B match situation happens.

To allow the program to branch to its respective interrupt vector address, the global interrupt enable bit, EMI, respective TM Interrupt enable bit, and relevant Multi-function Interrupt enable bit, MFnE, must first be set. When the interrupt is enabled, the stack is not full and a TM comparator match situation occurs, a subroutine call to the relevant Multi-function Interrupt vector locations, will take place. When the TM interrupt is serviced, the EMI bit will be automatically cleared to disable other interrupts, however only the related MFnF flag will be automatically cleared. As the TM interrupt request flags will not be automatically cleared, they have to be cleared by the application program.

Interrupt Wake-up Function

Each of the interrupt functions has the capability of waking up the microcontroller when in the SLEEP or IDLE Mode. A wake-up is generated when an interrupt request flag changes from low to high and is independent of whether the interrupt is enabled or not. Therefore, even though the device is in the SLEEP or IDLE Mode and its system oscillator stopped, situations such as external edge transitions on the external interrupt pins, a low power supply voltage or comparator input change may cause their respective interrupt flag to be set high and consequently generate an interrupt. Care must therefore be taken if spurious wake-up situations are to be avoided. If an interrupt wake-up function is to be disabled then the corresponding interrupt request flag should be set high before the device enters the SLEEP or IDLE Mode. The interrupt enable bits have no effect on the interrupt wake-up function.

Programming Considerations

By disabling the relevant interrupt enable bits, a requested interrupt can be prevented from being serviced, however, once an interrupt request flag is set, it will remain in this condition in the interrupt register until the corresponding interrupt is serviced or until the request flag is cleared by the application program.

Where a certain interrupt is contained within a Multi-function interrupt, then when the interrupt service routine is executed, as only the Multi-function interrupt request flags, MF0F~MF2F, will be automatically cleared, the individual request flag for the function needs to be cleared by the application program.

It is recommended that programs do not use the "CALL" instruction within the interrupt service subroutine. Interrupts often occur in an unpredictable manner or need to be serviced immediately. If only one stack is left and the interrupt is not well controlled, the original control sequence will be damaged once a CALL subroutine is executed in the interrupt subroutine.

Every interrupt has the capability of waking up the microcontroller when it is in SLEEP or IDLE Mode, the wake up being generated when the interrupt request flag changes from low to high. If it is required to prevent a certain interrupt from waking up the microcontroller then its respective request flag should be first set high before enter SLEEP or IDLE Mode.

As only the Program Counter is pushed onto the stack, then when the interrupt is serviced, if the contents of the accumulator, status register or other registers are altered by the interrupt service program, their contents should be saved to the memory at the beginning of the interrupt service routine.

To return from an interrupt subroutine, either a RET or RETI instruction may be executed. The RETI instruction in addition to executing a return to the main program also automatically sets the EMI bit high to allow further interrupts. The RET instruction however only executes a return to the main program leaving the EMI bit in its present zero state and therefore disabling the execution of further interrupts.

Low Voltage Detector – LVD

Each device has a Low Voltage Detector function, also known as LVD. This enabled the device to monitor the power supply voltage, V_{DD}, and provide a warning signal should it fall below a certain level. This function may be especially useful in battery applications where the supply voltage will gradually reduce as the battery ages, as it allows an early warning battery low signal to be generated. The Low Voltage Detector also has the capability of generating an interrupt signal.

LVD Register

The Low Voltage Detector function is controlled using a single register with the name LVDC. Three bits in this register, VLVD2~VLVD0, are used to select one of eight fixed voltages below which a low voltage condition will be detemined. A low voltage condition is indicated when the LVDO bit is set. If the LVDO bit is low, this indicates that the V_{DD} voltage is above the preset low voltage value. The LVDEN bit is used to control the overall on/off function of the low voltage detector. Setting the bit high will enable the low voltage detector. Clearing the bit to zero will switch off the internal low voltage detector circuits. As the low voltage detector will consume a certain amount of power, it may be desirable to switch off the circuit when not in use, an important consideration in power sensitive battery powered applications.

LVDC Register

Bit	7	6	5	4	3	2	1	0
Name	—	—	LVDO	LVDEN	—	VLVD2	VLVD1	VLVD0
R/W	—	—	R	R/W	—	R/W	R/W	R/W
POR	—	—	0	0	—	0	0	0

Bit 7~6 Unimplemented, read as "0"

Bit 5 **LVDO:** LVD Output Flag
0: No Low Voltage Detected
1: Low Voltage Detected

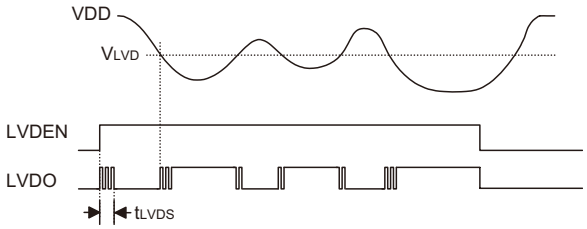
Bit 4 **LVDEN:** Low Voltage Detector Enable/Disable
0: Disable
1: Enable

Bit 3 Unimplemented, read as "0"

Bit 2~0 **VLVD2~VLVD0:** Select LVD Voltage
000: 2.0V
001: 2.2V
010: 2.4V
011: 2.7V
100: 3.0V
101: 3.3V
110: 3.6V
111: 4.0V

LVD Operation

The Low Voltage Detector function operates by comparing the power supply voltage, V_{DD}, with a pre-specified voltage level stored in the LVDC register. This has a range of between 2.0V and 4.0V. When the power supply voltage, V_{DD}, falls below this pre-determined value, the LVDO bit will be set high indicating a low power supply voltage condition. The Low Voltage Detector function is supplied by a reference voltage which will be automatically enabled. When the device is powered down the low voltage detector will remain active if the LVDEN bit is high. After enabling the Low Voltage Detector, a time delay t_{LVDS} should be allowed for the circuitry to stabilise before reading the LVDO bit. Note also that as the V_{DD} voltage may rise and fall rather slowly, at the voltage nears that of V_{LVD}, there may be multiple bit LVDO transitions.



LVD Operation

The Low Voltage Detector also has its own interrupt which is contained within one of the Multi-function interrupts, providing an alternative means of low voltage detection, in addition to polling the LVDO bit. The interrupt will only be generated after a delay of t_{LVD} after the LVDO bit has been set high by a low voltage condition. When the device is powered down the Low Voltage Detector will remain active if the LVDEN bit is high. In this case, the LVF interrupt request flag will be set, causing an interrupt to be generated if V_{DD} falls below the preset LVD voltage. This will cause the device to wake-up from the SLEEP or IDLE Mode, however if the Low Voltage Detector wake up function is not required then the LVF flag should be first set high before the device enters the SLEEP or IDLE Mode.

When LVD function is enabled, it is recommended to clear LVD flag first, and then enables interrupt function to avoid mistake action.

LCD Driver

For large volume applications, which incorporate an LCD in their design, the use of a custom display rather than a more expensive character based display reduces costs significantly. However, the corresponding COM and SEG signals required, which vary in both amplitude and time, to drive such a custom display require many special considerations for proper LCD operation to occur. These devices all contain an LCD Driver function, which with their internal LCD signal generating circuitry and various options, will automatically generate these time and amplitude varying signals to provide a means of direct driving and easy interfacing to a range of custom LCDs.

All devices include a wide range of options to enable LCD displays of various types to be driven. The table shows the range of options available across the device range.

Device	Duty	Bias	Bias Type	Wave Type
HT69F30A	1/3 or 1/4	1/2 or 1/3	R or C	A or B
HT69F40A				
HT69F50A				

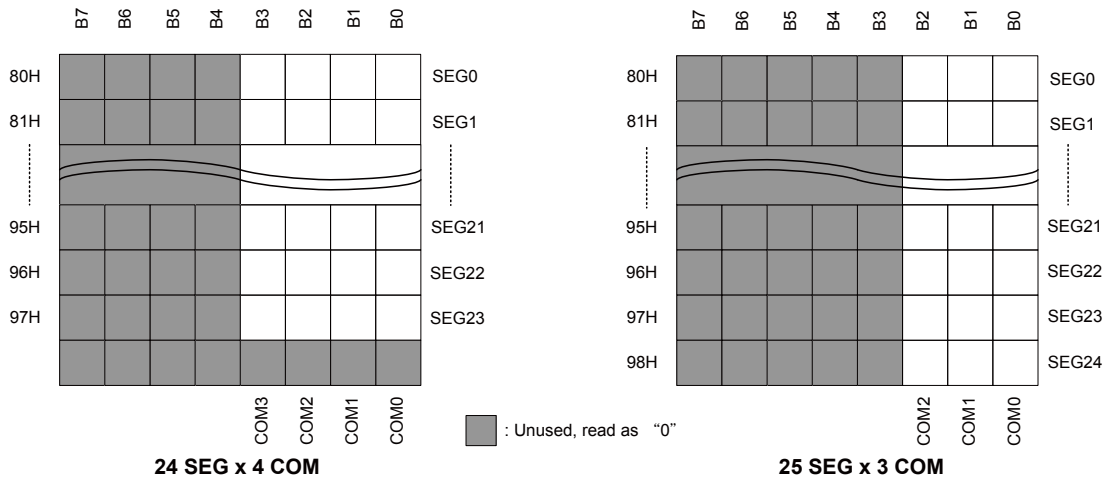
LCD Selections

LCD Memory

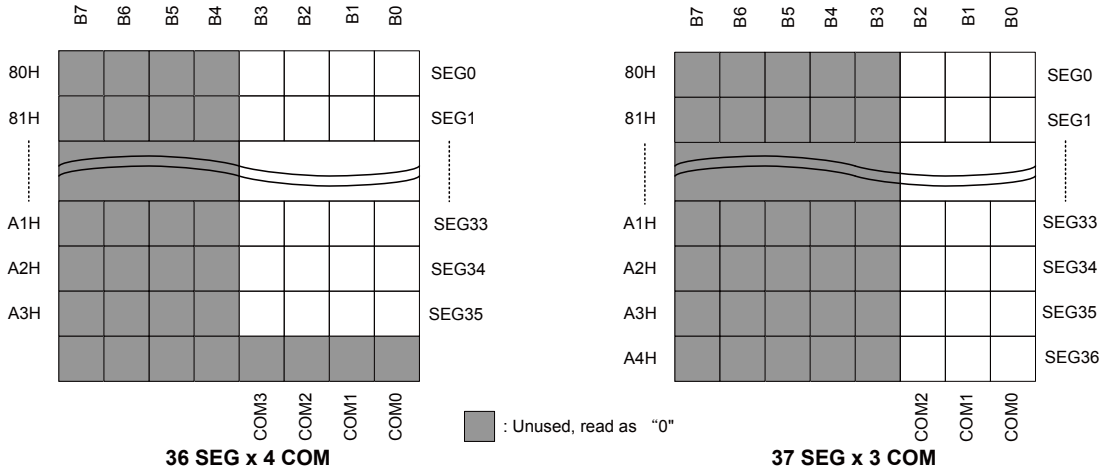
An area of Data Memory is especially reserved for use for the LCD display data. This data area is known as the LCD Memory. Any data written here will be automatically read by the internal display driver circuits, which will in turn automatically generate the necessary LCD driving signals. Therefore any data written into this Memory will be immediately reflected into the actual display connected to the microcontroller.

As the LCD Memory addresses overlap those of the General Purpose Data Memory, it s stored in its own independent Bank 1 area. The Data Memory Bank to be used is chosen by using the Bank Pointer, which is a special function register in the Data Memory, with the name, BP. To access the LCD Memory therefore requires first that Bank 1 is selected by writing a value of 01H to the BP register. After this, the memory can then be accessed by using indirect addressing through the use of Memory Pointer MP1. With Bank 1 selected, then using MP1 to read or write to the memory area, starting with address “80H” for all the devices, will result in operations to the LCD Memory. Directly addressing the Display Memory is not applicable and will result in a data access to the Bank 0 General Purpose Data Memory.

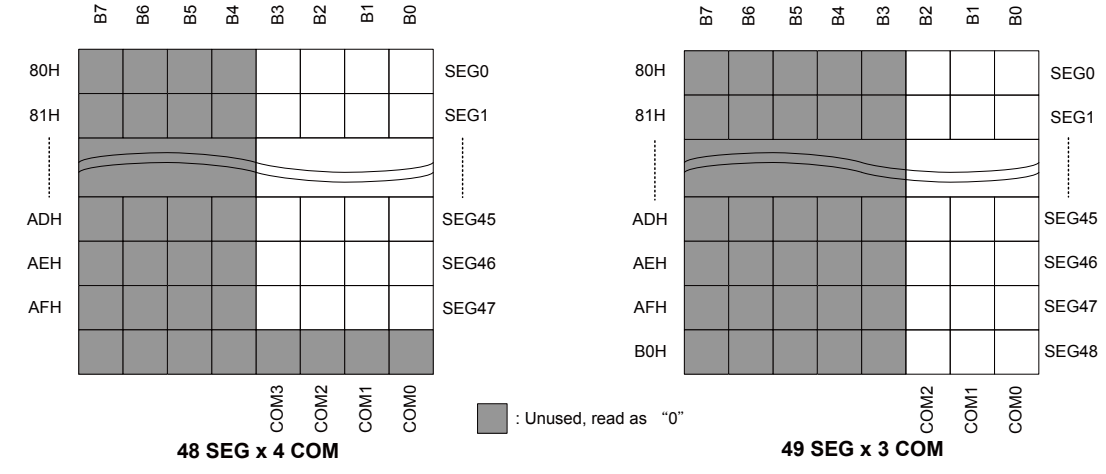
The accompanying LCD Memory Map diagrams shows how the internal LCD Memory is mapped to the Segments and Commons of the display for the devices. LCD Memory Maps for devices with smaller memory capacities can be extrapolated from these diagrams.



HT69F30A LCD Memory Map



HT69F40A LCD Memory Map



HT69F50A LCD Memory Map

LCD Register

There is one control register, named as LCDC, in the Data Memory used to control the various setup features of the LCD Driver.

Various bits in this registers control functions such as LCD wave type, duty type, bias type, bias resistor selection as well as overall LCD enable and disable. The LCDEN bit in the LCDC register, which provides the overall LCD enable/disable function, will only be effective when the device is in the NOAMRL, SLOW or IDLE Mode. If the device is in the SLEEP Mode then the display will always be disabled. Bits, RSEL0 and RSEL1, in the LCDC register select the internal bias resistors to supply the LCD panel with the correct bias voltages. A choice to best match the LCD panel used in the application can be selected also to minimise bias current. The TYPE bit in the same register is used to select whether Type A or Type B LCD control signals are used.

LCDC Register

Bit	7	6	5	4	3	2	1	0
Name	TYPE	—	DTYC	—	BIAS	RSEL1	RSEL0	LCDEN
R/W	R/W	—	R/W	—	R/W	R/W	R/W	R/W
POR	0	—	0	—	0	0	0	0

Bit 7 **TYPE:** LCD Wave Type Control
0: Type A
1: Type B

Bit 6 Unimplemented, read as "0"

Bit 5 **DTYC:** LCD Duty Control
0: 1/3 duty
1: 1/4 duty

Bit 4 Unimplemented, read as "0"

Bit 3 **BIAS:** LCD Bias Control
0: 1/2 bias
1: 1/3 bias

Bit 2~1 **RSEL1~RSEL0:** LCD Bias Resistor Selection
1/3 Bias
00: 600kΩ
01: 300kΩ
10: 100kΩ
11: 50kΩ
1/2 Bias
00: 400kΩ
01: 200kΩ
10: 67kΩ
11: 34kΩ

Bit 0 **LCDEN:** LCD Enable Control
0: Disable
1: Enable
In the NORMAL, SLOW or IDLE mode, the LCD on/off function can be controlled by this bit. in the SLEEP mode, the LCD function is always off.

LCD Reset Function

The LCD has an internal reset function that is an OR function of the inverted LCDEN bit in the LCDC register and the Sleep function. When the LCDEN bit is set to 1 to enable the LCD driver function before the device enters the SLEEP mode, the LCD function will be reset after the device enters the SLEEP mode. Clearing the LCDEN bit to zero will also reset the LCD function.

LCDEN	SLEEP Mode	Reset LCD
0	Off	√
0	On	√
1	Off	x
1	On	√

LCD Reset Function

Clock Source

The LCD clock source is the internal clock signal, f_{SUB}, divided by 8, using an internal divider circuit. The f_{SUB} internal clock is supplied by either the LIRC or LXT oscillator, the choice of which is determined by a configuration option. For proper LCD operation, this arrangement is provided to generate an ideal LCD clock source frequency of 4kHz.

f_{SUB} Clock Source	LCD Clock Frequency
LIRC	4kHz
LXT	4kHz

LCD Clock Source

LCD Driver Output

The number of COM and SEG outputs supplied by the LCD driver, as well as its biasing and duty selections, are dependent upon how the LCD control bits are programmed. The Bias Type, whether C or R type is selected via a configuration option.

The nature of Liquid Crystal Displays require that only AC voltages can be applied to their pixels as the application of DC voltages to LCD pixels may cause permanent damage. For this reason the relative contrast of an LCD display is controlled by the actual RMS voltage applied to each pixel, which is equal to the RMS value of the voltage on the COM pin minus the voltage applied to the SEG pin. This differential RMS voltage must be greater than the LCD saturation voltage for the pixel to be on and less than the threshold voltage for the pixel to be off.

The requirement to limit the DC voltage to zero and to control as many pixels as possible with a minimum number of connections, requires that both a time and amplitude signal is generated and applied to the application LCD. These time and amplitude varying signals are automatically generated by the LCD driver circuits in the microcontroller. What is known as the duty determines the number of common lines used, which are also known as backplanes or COMs. The duty, which is chosen by a control bit to have a value of 1/3 or 1/4 and which equates to a COM number of 3 or 4 respectively, therefore defines the number of time divisions within each LCD signal frame. Two types of signal generation are also provided, known as Type A and Type B, the required type is selected via the TYPE bit in the LCDC register. Type B offers lower frequency signals, however lower frequencies may introduce flickering and influence display clarity.

LCD Voltage Source Biasing

The time and amplitude varying signals generated by the LCD Driver function require the generation of several voltage levels for their operation. The number of voltage levels used by the signal depends upon the value of the BIAS bit in the LCDC register. The device can have either R type or C type biasing selected via a configuration option. Selecting the C type biasing will enable an internal charge pump whose multiplier ratio can be selected using an additional configuration option.

For R type biasing an external LCD voltage source must be supplied on pin VLCD to generate the internal biasing voltages. This could be the microcontroller power supply or some other voltage source. For the R type 1/2 bias selection, three voltage levels V_{SS}, V_A and V_B are utilised. The voltage V_A is equal to the externally supplied voltage source applied to pin VLCD. The voltage V_B is generated internally by the microcontroller and will have a value equal to $V_{LCD}/2$. For the R type 1/3 bias selection, four voltage levels V_{SS}, V_A, V_B and V_C are utilised. The voltage V_A is equal to V_{LCD}. The voltage V_B is equal to $V_{LCD}\times2/3$ while the voltage V_C is equal to $V_{LCD}\times1/3$. In addition to selecting 1/2 or 1/3 bias, several values of bias resistor can be chosen using bits in the LCDC register.

Different values of internal bias resistors can be selected using the RSEL0 and RESEL1 bits in the LCDC register. This along with the voltage on pin VLCD will determine the bias current. The connection to the VMAX pin depends upon the voltage that is applied to the VLCD pin. If the VDD voltage is greater than the voltage applied to the VLCD pin then the VMAX pin should be connected to VDD, otherwise the VMAX pin should be connected to pin VLCD. Note that no external capacitors or resistors are required to be connected if R type biasing is used.

Condition	VMAX connection
$V_{DD} > V_{LCD}$	Connect VMAX to VDD
Otherwise	Connect VMAX to VLCD

R Type Bias VMAX Pin Connection

For C type biasing an external LCD voltage source must also be supplied on pin VLCD to generate the internal biasing voltages. The C type biasing scheme uses an internal charge pump circuit, which in the case of the 1/3 bias selection, can generate voltages higher than what is supplied on VLCD. This feature is useful in applications where the microcontroller supply voltage is less than the supply voltage required by the LCD. An additional charge pump capacitor must also be connected between pins C1 and C2 to generate the necessary voltage levels.

For the C type 1/2 bias selection, three voltage levels V_{SS}, V_A and V_B are utilised. The voltage V_A is generated internally and has a value of V_{LCD}. The voltage V_B will have a value equal to $V_A\times1/2$. For the C type 1/2 bias configuration V_C is not used.

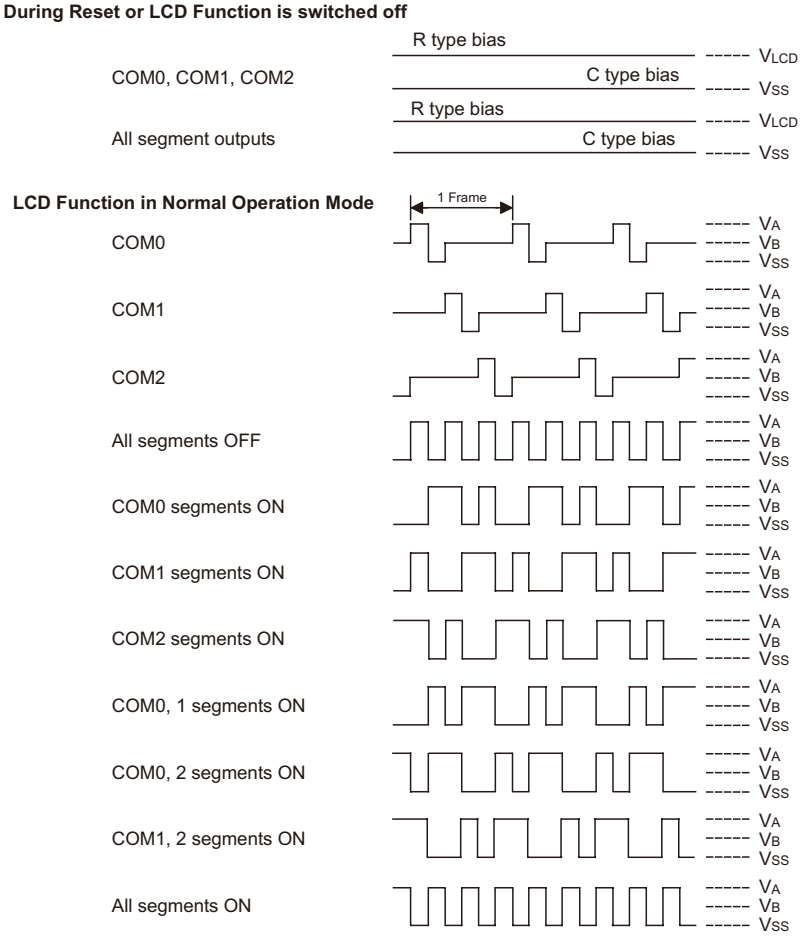
For the C type 1/3 bias selection, four voltage levels V_{SS}, V_A, V_B and V_C are utilised. The voltage V_A is generated internally and has a value of $V_{LCD}\times3/2$. The voltage V_B will have a value equal to $V_A\times2/3$ and V_C will have a value equal to $V_A\times1/3$. The connection to the VMAX pin depends upon the bias and the voltage that is applied to VLCD. It is extremely important to ensure that these charge pump generated internal voltages do not exceed the maximum V_{DD} voltage of 5.5V.

Biasing Type		VMAX Connection
1/3 Bias	$V_{DD}>V_{LCD}\times1.5$	Connect VMAX to VDD
	Otherwise	Connect VMAX to V1
1/2 Bias	$V_{DD}>V_{LCD}$	Connect VMAX to VDD
	Otherwise	Connect VMAX to VLCD

C Type Bias VMAX Pin Connection

LCD Waveform Timing Diagram

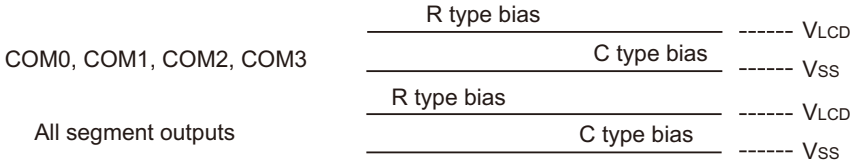
The accompanying timing diagrams depict the display driver signals generated by the microcontroller for various values of duty and bias. The huge range of various permutations only permits a few types to be displayed here.



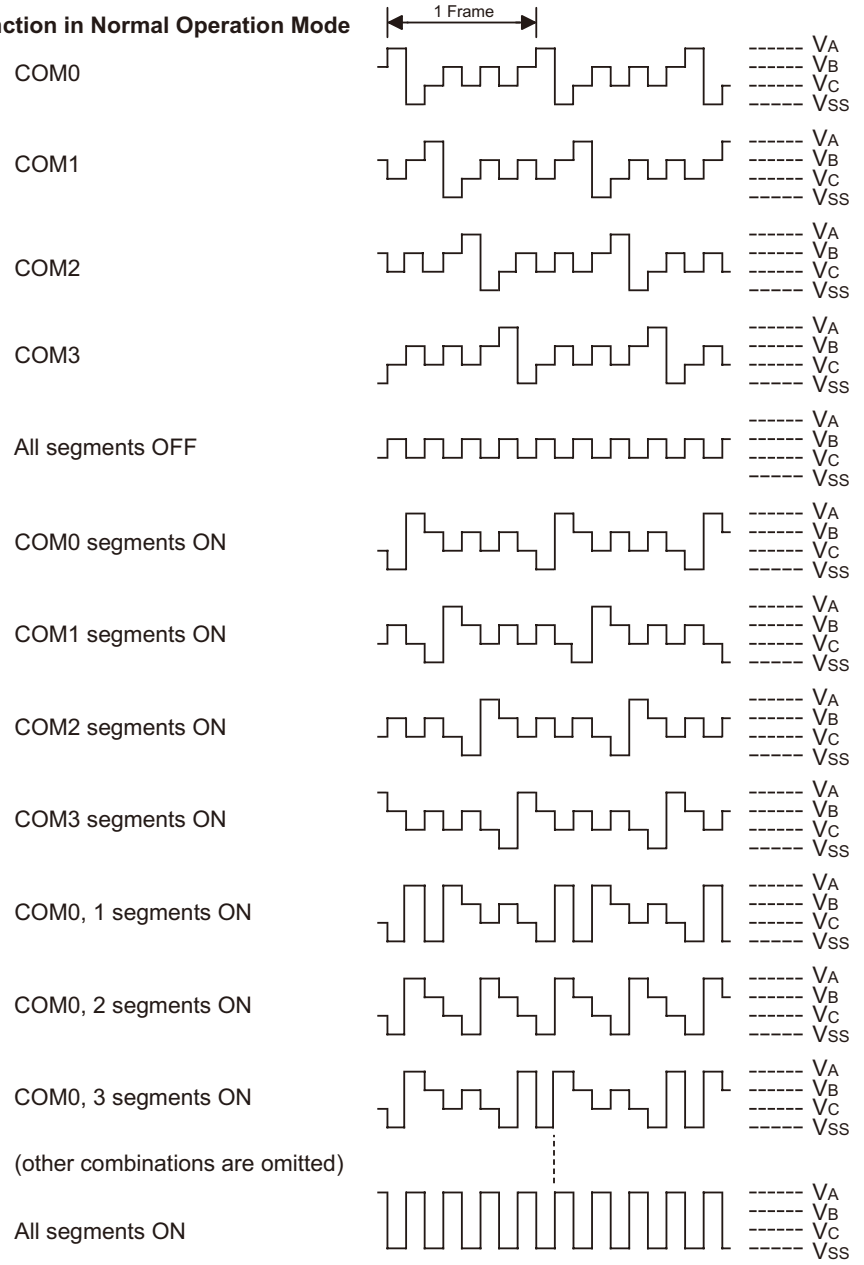
LCD Driver Output – Type A- 1/3 Duty, 1/2 Bias

Note: For 1/2 Bias, the $V_A=V_{LCD}$, $V_B=V_{LCD}\times1/2$ for both R and C type.

During Reset or LCD Function is switched off



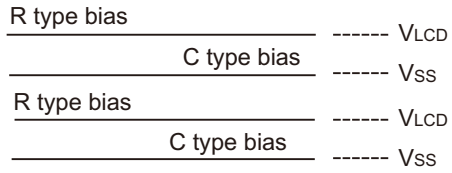
LCD Function in Normal Operation Mode



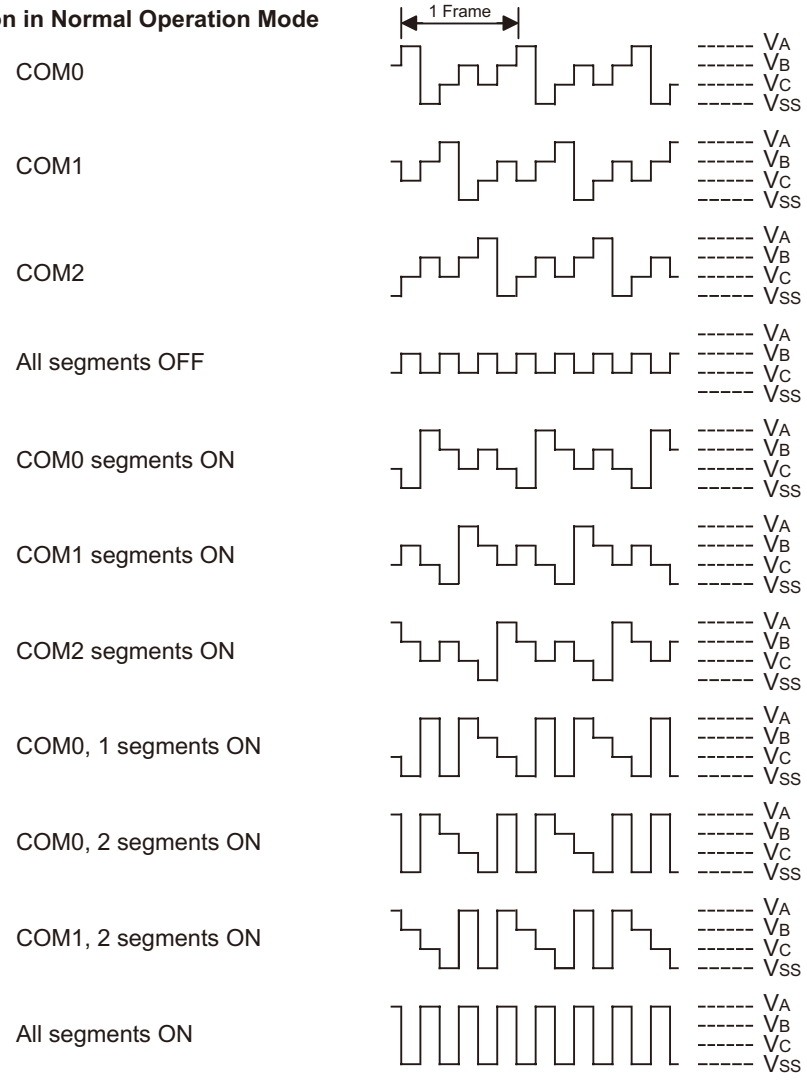
LCD Driver Output – Type A - 1/4 Duty, 1/3 Bias

Note: For 1/3 R type bias, the $V_A=V_{LCD}$, $V_B=V_{LCD}\times 2/3$ and $V_C=V_{LCD}\times 1/3$.
For 1/3 C type bias, the $V_A=V_{LCD}\times 1.5$, $V_B=V_{LCD}$ and $V_C=V_{LCD}\times 1/2$.

During Reset or LCD Function is switched off

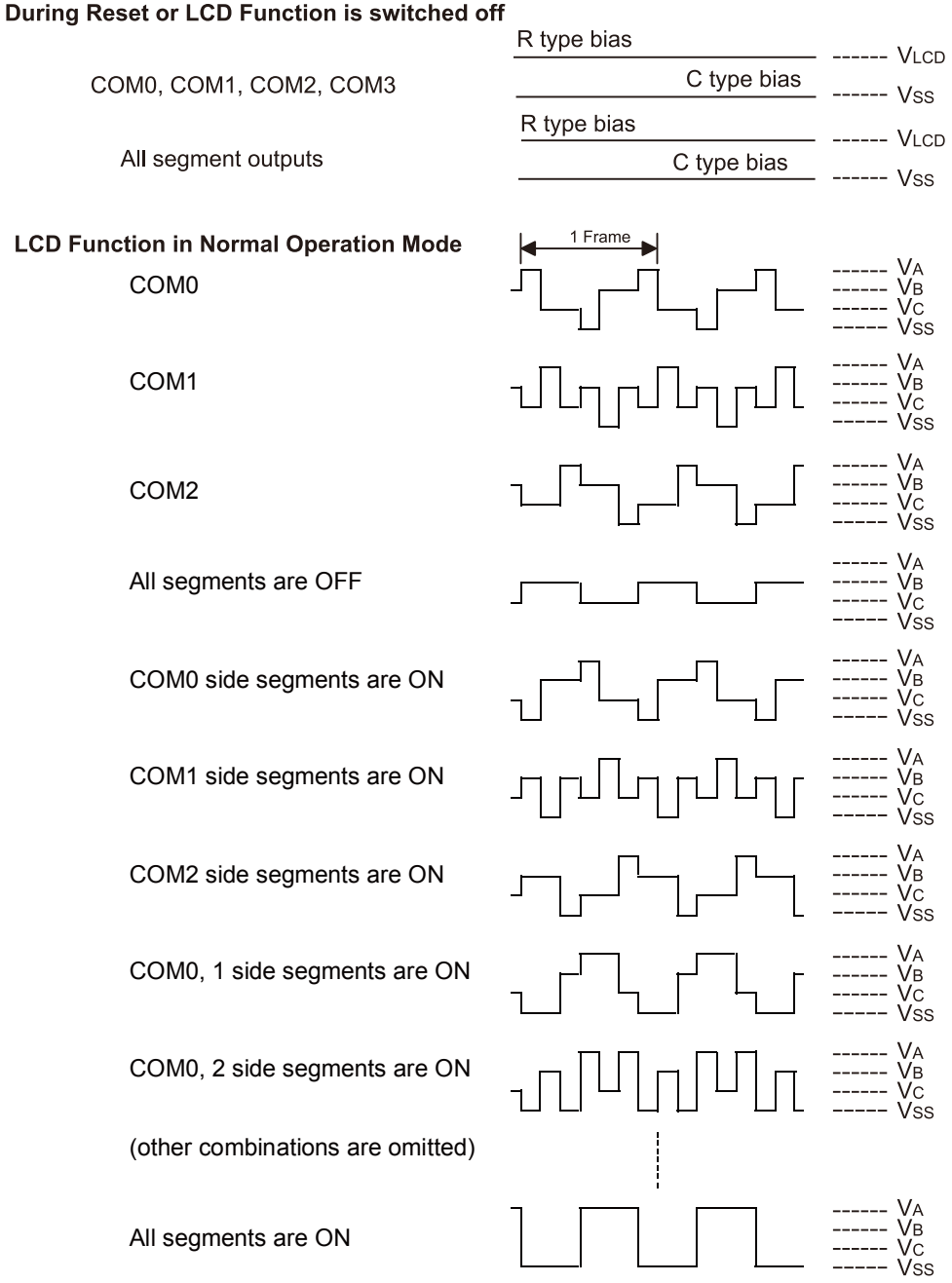


LCD Function in Normal Operation Mode



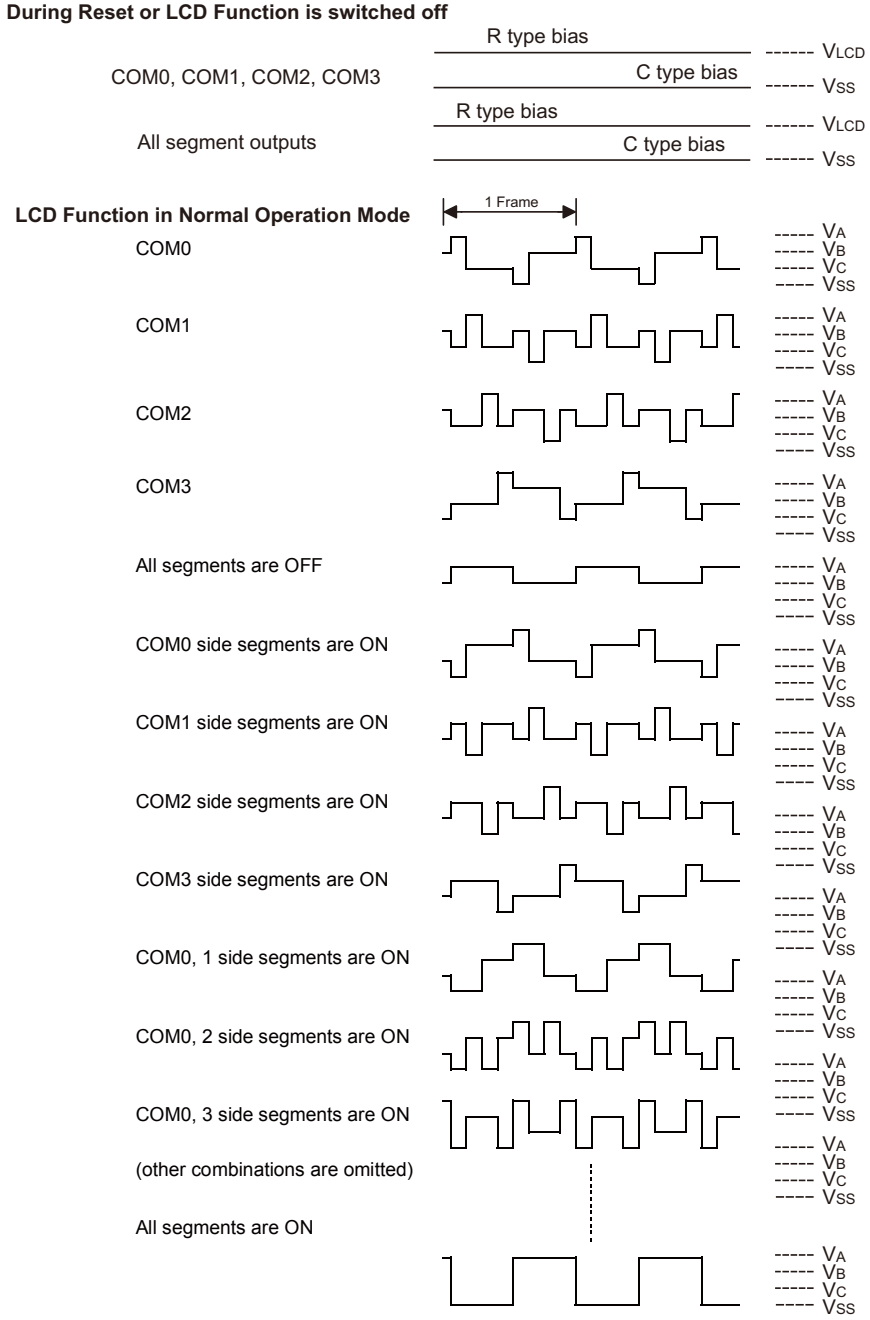
LCD Driver Output – Type A- 1/3 Duty, 1/3 Bias

Note: For 1/3 R type bias, the $V_A=V_{LCD}$, $V_B=V_{LCD}\times2/3$ and $V_C=V_{LCD}\times1/3$.
For 1/3 C type bias, $V_A=V_{LCD}\times1.5$, $V_B=V_{LCD}$ and $V_C=V_{LCD}\times1/2$.



LCD Driver Output – Type B- 1/3 Duty, 1/3 Bias

Note: For 1/3 R type bias, the $V_A=V_{LCD}$, $V_B=V_{LCD}\times2/3$ and $V_C=V_{LCD}\times1/3$.
For 1/3 C type bias, $V_A=V_{LCD}\times3/2$, $V_B=V_{LCD}$ and $V_C=V_{LCD}\times1/2$.



LCD Driver Output – Type B- 1/4 Duty, 1/3 Bias

Note: For 1/3 R type bias, the $V_A=V_{LCD}$, $V_B=V_{LCD}\times2/3$ and $V_C=V_{LCD}\times1/3$.
For 1/3 C type bias, $V_A=V_{LCD}\times3/2$, $V_B=V_{LCD}$ and $V_C=V_{LCD}\times1/2$.

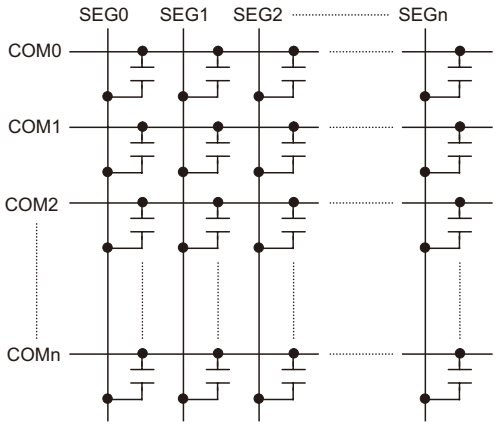
Programming Considerations

Certain precautions must be taken when programming the LCD. One of these is to ensure that the LCD Memory is properly initialised after the microcontroller is powered on. Like the General Purpose Data Memory, the contents of the LCD Memory are in an unknown condition after power-on. As the contents of the LCD Memory will be mapped into the actual display, it is important to initialise this memory area into a known condition soon after applying power to obtain a proper display pattern.

Consideration must also be given to the capacitive load of the actual LCD used in the application. As the load presented to the microcontroller by LCD pixels can be generally modeled as mainly capacitive in nature, it is important that this is not excessive, a point that is particularly true in the case of the COM lines which may be connected to many LCD pixels. The accompanying diagram depicts the equivalent circuit of the LCD.

One additional consideration that must be taken into account is what happens when the microcontroller enters the Idle or Slow Mode. The LCDEN control bit in the LCDC register permits the display to be powered off to reduce power consumption. If this bit is zero, the driving signals to the display will cease, producing a blank display pattern but reducing any power consumption associated with the LCD.

After Power-on, note that as the LCDEN bit will be cleared to zero, the display function will be disabled.



LCD Panel Equivalent Circuit

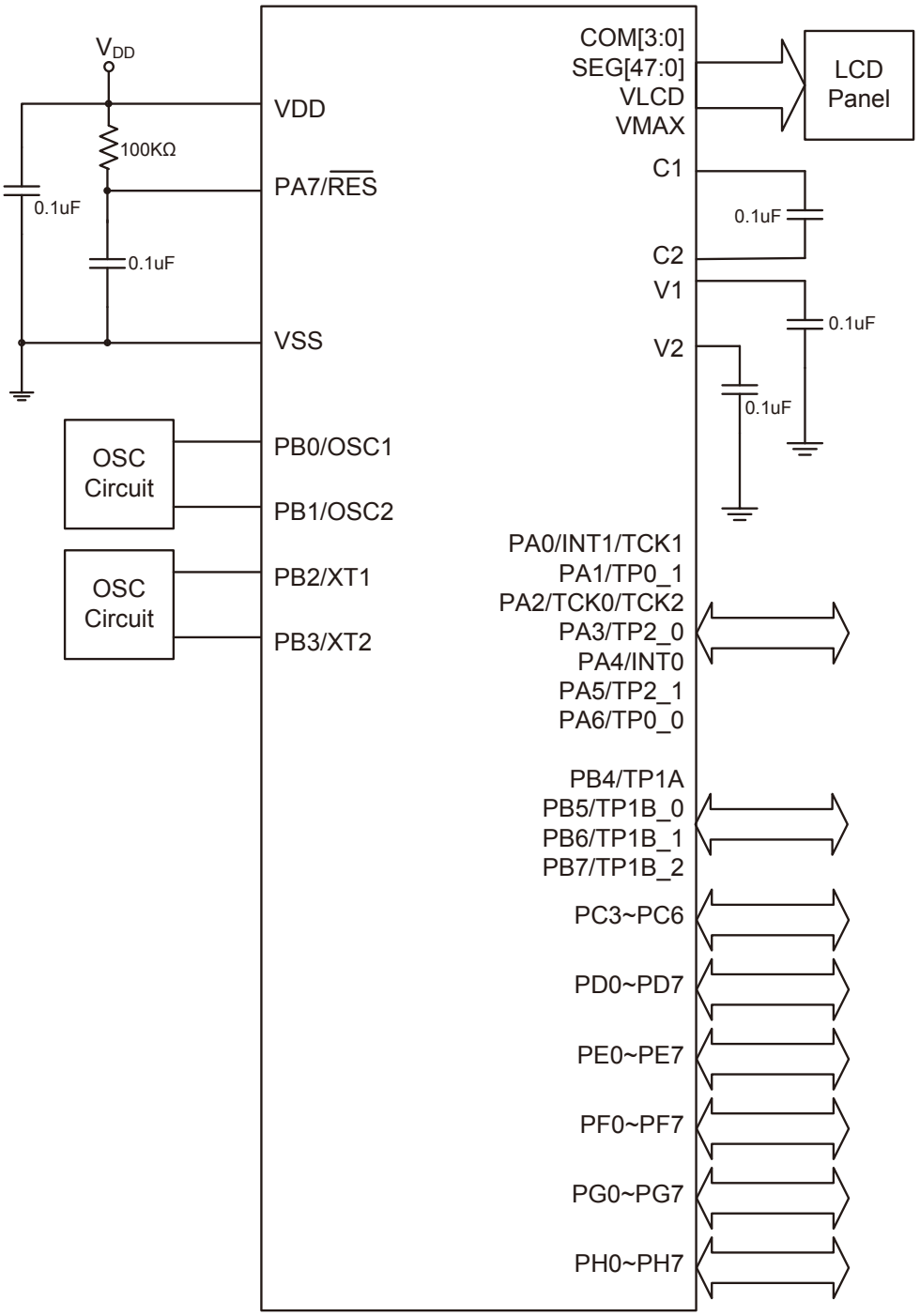
Configuration Options

Configuration options refer to certain options within the MCU that are programmed into the device during the programming process. During the development process, these options are selected using the HT-IDE software development tools. As these options are programmed into the device using the hardware programming tools, once they are selected they cannot be changed later using the application program. All options must be defined for proper system function, the details of which are shown in the table.

No.	Options
1	High Speed System Oscillator Selection – f_H HXT, ERC, EC or HIRC
2	HXT Mode Selection 1MHz~12MHz or 455kHz
3	Low Speed System Oscillator Selection – f_L LXT or LIRC
4	HIRC Frequency Selection 4MHz, 8MHz or 12MHz
5	LCD Bias Type Selection R type or C type
6	LCD Voltage Selection V_{LCD} voltage is 3.0V, 4.5V or 1.5V
7	TMR/INT Pin Input Filter Function Enable or Disable
8	I/O or Reset pin selection Reset pin or I/O pin
9	Watchdog Timer Function Always enable or Application program enable

Note: The f_{SUB} clock source is derived from LXT or LIRC selected by the f_L configuration option.

Application Circuits



Instruction Set

Introduction

Central to the successful operation of any microcontroller is its instruction set, which is a set of program instruction codes that directs the microcontroller to perform certain operations. In the case of Holtek microcontrollers, a comprehensive and flexible set of over 60 instructions is provided to enable programmers to implement their application with the minimum of programming overheads.

For easier understanding of the various instruction codes, they have been subdivided into several functional groupings.

Instruction Timing

Most instructions are implemented within one instruction cycle. The exceptions to this are branch, call, or table read instructions where two instruction cycles are required. One instruction cycle is equal to 4 system clock cycles, therefore in the case of an 8MHz system oscillator, most instructions would be implemented within 0.5μs and branch or call instructions would be implemented within 1μs. Although instructions which require one more cycle to implement are generally limited to the JMP, CALL, RET, RETI and table read instructions, it is important to realize that any other instructions which involve manipulation of the Program Counter Low register or PCL will also take one more cycle to implement. As instructions which change the contents of the PCL will imply a direct jump to that new address, one more cycle will be required. Examples of such instructions would be ″CLR PCL″ or ″MOV PCL, A″. For the case of skip instructions, it must be noted that if the result of the comparison involves a skip operation then this will also take one more cycle, if no skip is involved then only one cycle is required.

Moving and Transferring Data

The transfer of data within the microcontroller program is one of the most frequently used operations. Making use of three kinds of MOV instructions, data can be transferred from registers to the Accumulator and vice-versa as well as being able to move specific immediate data directly into the Accumulator. One of the most important data transfer applications is to receive data from the input ports and transfer data to the output ports.

Arithmetic Operations

The ability to perform certain arithmetic operations and data manipulation is a necessary feature of most microcontroller applications. Within the Holtek microcontroller instruction set are a range of add and subtract instruction mnemonics to enable the necessary arithmetic to be carried out. Care must be taken to ensure correct handling of carry and borrow data when results exceed 255 for addition and less than 0 for subtraction. The increment and decrement instructions INC, INCA, DEC and DECA provide a simple means of increasing or decreasing by a value of one of the values in the destination specified.

Logical and Rotate Operations

The standard logical operations such as AND, OR, XOR and CPL all have their own instruction within the Holtek microcontroller instruction set. As with the case of most instructions involving data manipulation, data must pass through the Accumulator which may involve additional programming steps. In all logical data operations, the zero flag may be set if the result of the operation is zero. Another form of logical data manipulation comes from the rotate instructions such as RR, RL, RRC and RLC which provide a simple means of rotating one bit right or left. Different rotate instructions exist depending on program requirements. Rotate instructions are useful for serial port programming applications where data can be rotated from an internal register into the Carry bit from where it can be examined and the necessary serial bit set high or low. Another application where rotate data operations are used is to implement multiplication and division calculations.

Branches and Control Transfer

Program branching takes the form of either jumps to specified locations using the JMP instruction or to a subroutine using the CALL instruction. They differ in the sense that in the case of a subroutine call, the program must return to the instruction immediately when the subroutine has been carried out. This is done by placing a return instruction RET in the subroutine which will cause the program to jump back to the address right after the CALL instruction. In the case of a JMP instruction, the program simply jumps to the desired location. There is no requirement to jump back to the original jumping off point as in the case of the CALL instruction. One special and extremely useful set of branch instructions are the conditional branches. Here a decision is first made regarding the condition of a certain data memory or individual bits. Depending upon the conditions, the program will continue with the next instruction or skip over it and jump to the following instruction. These instructions are the key to decision making and branching within the program perhaps determined by the condition of certain input switches or by the condition of internal data bits.

Bit Operations

The ability to provide single bit operations on Data Memory is an extremely flexible feature of all Holtek microcontrollers. This feature is especially useful for output port bit programming where individual bits or port pins can be directly set high or low using either the "SET [m].i" or "CLR [m].i" instructions respectively. The feature removes the need for programmers to first read the 8-bit output port, manipulate the input data to ensure that other bits are not changed and then output the port with the correct new data. This read-modify-write process is taken care of automatically when these bit operation instructions are used.

Table Read Operations

Data storage is normally implemented by using registers. However, when working with large amounts of fixed data, the volume involved often makes it inconvenient to store the fixed data in the Data Memory. To overcome this problem, Holtek microcontrollers allow an area of Program Memory to be setup as a table where data can be directly stored. A set of easy to use instructions provides the means by which this fixed data can be referenced and retrieved from the Program Memory.

Other Operations

In addition to the above functional instructions, a range of other instructions also exist such as the "HALT" instruction for Power-down operations and instructions to control the operation of the Watchdog Timer for reliable program operations under extreme electric or electromagnetic environments. For their relevant operations, refer to the functional related sections.

Instruction Set Summary

The following table depicts a summary of the instruction set categorised according to function and can be consulted as a basic instruction reference using the following listed conventions.

Table conventions

x: Bits immediate data
m: Data Memory address
A: Accumulator
i: 0~7 number of bits
addr: Program memory address

Mnemonic	Description	Cycles	Flag Affected
Arithmetic			
ADD A,[m]	Add Data Memory to ACC	1	Z, C, AC, OV
ADDM A,[m]	Add ACC to Data Memory	1Note	Z, C, AC, OV
ADD A,x	Add immediate data to ACC	1	Z, C, AC, OV
ADC A,[m]	Add Data Memory to ACC with Carry	1	Z, C, AC, OV
ADCM A,[m]	Add ACC to Data memory with Carry	1Note	Z, C, AC, OV
SUB A,x	Subtract immediate data from the ACC	1	Z, C, AC, OV
SUB A,[m]	Subtract Data Memory from ACC	1	Z, C, AC, OV
SUBM A,[m]	Subtract Data Memory from ACC with result in Data Memory	1Note	Z, C, AC, OV
SBC A,[m]	Subtract Data Memory from ACC with Carry	1	Z, C, AC, OV
SBCM A,[m]	Subtract Data Memory from ACC with Carry, result in Data Memory	1Note	Z, C, AC, OV
DAA [m]	Decimal adjust ACC for Addition with result in Data Memory	1Note	C
Logic Operation			
AND A,[m]	Logical AND Data Memory to ACC	1	Z
OR A,[m]	Logical OR Data Memory to ACC	1	Z
XOR A,[m]	Logical XOR Data Memory to ACC	1	Z
ANDM A,[m]	Logical AND ACC to Data Memory	1Note	Z
ORM A,[m]	Logical OR ACC to Data Memory	1Note	Z
XORM A,[m]	Logical XOR ACC to Data Memory	1Note	Z
AND A,x	Logical AND immediate Data to ACC	1	Z
OR A,x	Logical OR immediate Data to ACC	1	Z
XOR A,x	Logical XOR immediate Data to ACC	1	Z
CPL [m]	Complement Data Memory	1Note	Z
CPLA [m]	Complement Data Memory with result in ACC	1	Z
Increment & Decrement			
INCA [m]	Increment Data Memory with result in ACC	1	Z
INC [m]	Increment Data Memory	1Note	Z
DECA [m]	Decrement Data Memory with result in ACC	1	Z
DEC [m]	Decrement Data Memory	1Note	Z
Rotate			
RRA [m]	Rotate Data Memory right with result in ACC	1	None
RR [m]	Rotate Data Memory right	1Note	None
RRCA [m]	Rotate Data Memory right through Carry with result in ACC	1	C
RRC [m]	Rotate Data Memory right through Carry	1Note	C
RLA [m]	Rotate Data Memory left with result in ACC	1	None
RL [m]	Rotate Data Memory left	1Note	None
RLCA [m]	Rotate Data Memory left through Carry with result in ACC	1	C
RLC [m]	Rotate Data Memory left through Carry	1Note	C

Mnemonic	Description	Cycles	Flag Affected
Data Move			
MOV A,[m]	Move Data Memory to ACC	1	None
MOV [m],A	Move ACC to Data Memory	1Note	None
MOV A,x	Move immediate data to ACC	1	None
Bit Operation			
CLR [m].i	Clear bit of Data Memory	1Note	None
SET [m].i	Set bit of Data Memory	1Note	None
Branch			
JMP addr	Jump unconditionally	2	None
SZ [m]	Skip if Data Memory is zero	1Note	None
SZA [m]	Skip if Data Memory is zero with data movement to ACC	1Note	None
SZ [m].i	Skip if bit i of Data Memory is zero	1Note	None
SNZ [m].i	Skip if bit i of Data Memory is not zero	1Note	None
SIZ [m]	Skip if increment Data Memory is zero	1Note	None
SDZ [m]	Skip if decrement Data Memory is zero	1Note	None
SIZA [m]	Skip if increment Data Memory is zero with result in ACC	1Note	None
SDZA [m]	Skip if decrement Data Memory is zero with result in ACC	1Note	None
CALL addr	Subroutine call	2	None
RET	Return from subroutine	2	None
RET A,x	Return from subroutine and load immediate data to ACC	2	None
RETI	Return from interrupt	2	None
Table Read			
TABRDC [m]	Read table (current page) to TBLH and Data Memory	2Note	None
TABRDL [m]	Read table (last page) to TBLH and Data Memory	2Note	None
Miscellaneous			
NOP	No operation	1	None
CLR [m]	Clear Data Memory	1Note	None
SET [m]	Set Data Memory	1Note	None
CLR WDT	Clear Watchdog Timer	1	TO, PDF
CLR WDT1	Pre-clear Watchdog Timer	1	TO, PDF
CLR WDT2	Pre-clear Watchdog Timer	1	TO, PDF
SWAP [m]	Swap nibbles of Data Memory	1Note	None
SWAPA [m]	Swap nibbles of Data Memory with result in ACC	1	None
HALT	Enter power down mode	1	TO, PDF

Note: 1. For skip instructions, if the result of the comparison involves a skip then two cycles are required, if no skip takes place only one cycle is required.

2. Any instruction which changes the contents of the PCL will also require 2 cycles for execution.

3. For the "CLR WDT1" and "CLR WDT2" instructions the TO and PDF flags may be affected by the execution status. The TO and PDF flags are cleared after both "CLR WDT1" and "CLR WDT2" instructions are consecutively executed. Otherwise the TO and PDF flags remain unchanged.

Instruction Definition

ADC A,[m] Add Data Memory to ACC with Carry

Description: The contents of the specified Data Memory, Accumulator and the carry flag are added. The result is stored in the Accumulator.

Operation: ACC ← ACC + [m] + C

Affected flag(s): OV, Z, AC, C

ADCM A,[m] Add ACC to Data Memory with Carry

Description: The contents of the specified Data Memory, Accumulator and the carry flag are added. The result is stored in the specified Data Memory.

Operation: [m] ← ACC + [m] + C

Affected flag(s): OV, Z, AC, C

ADD A,[m] Add Data Memory to ACC

Description: The contents of the specified Data Memory and the Accumulator are added. The result is stored in the Accumulator.

Operation: ACC ← ACC + [m]

Affected flag(s): OV, Z, AC, C

ADD A,x Add immediate data to ACC

Description: The contents of the Accumulator and the specified immediate data are added. The result is stored in the Accumulator.

Operation: ACC ← ACC + x

Affected flag(s): OV, Z, AC, C

ADDM A,[m] Add ACC to Data Memory

Description: The contents of the specified Data Memory and the Accumulator are added. The result is stored in the specified Data Memory.

Operation: [m] ← ACC + [m]

Affected flag(s): OV, Z, AC, C

AND A,[m] Logical AND Data Memory to ACC

Description: Data in the Accumulator and the specified Data Memory perform a bitwise logical AND operation. The result is stored in the Accumulator.

Operation: ACC ← ACC "AND" [m]

Affected flag(s): Z

AND A,x Logical AND immediate data to ACC

Description: Data in the Accumulator and the specified immediate data perform a bit wise logical AND operation. The result is stored in the Accumulator.

Operation: ACC ← ACC "AND" x

Affected flag(s): Z

ANDM A,[m] Logical AND ACC to Data Memory

Description: Data in the specified Data Memory and the Accumulator perform a bitwise logical AND operation. The result is stored in the Data Memory.

Operation: [m] ← ACC "AND" [m]

Affected flag(s): Z

CALL addr Subroutine call

Description Unconditionally calls a subroutine at the specified address. The Program Counter then increments by 1 to obtain the address of the next instruction which is then pushed onto the stack. The specified address is then loaded and the program continues execution from this new address. As this instruction requires an additional operation, it is a two cycle instruction.

Operation Stack ← Program Counter + 1
Program Counter ← addr

Affected flag(s) None

CLR [m] Clear Data Memory

Description Each bit of the specified Data Memory is cleared to 0.

Operation [m] ← 00H

Affected flag(s) None

CLR [m].i Clear bit of Data Memory

Description Bit i of the specified Data Memory is cleared to 0.

Operation [m].i ← 0

Affected flag(s) None

CLR WDT Clear Watchdog Timer

Description The TO, PDF flags and the WDT are all cleared.

Operation WDT cleared
TO ← 0
PDF ← 0

Affected flag(s) TO, PDF

CLR WDT1 Pre-clear Watchdog Timer

Description The TO, PDF flags and the WDT are all cleared. Note that this instruction works in conjunction with CLR WDT2 and must be executed alternately with CLR WDT2 to have effect. Repetitively executing this instruction without alternately executing CLR WDT2 will have no effect.

Operation WDT cleared
TO ← 0
PDF ← 0

Affected flag(s) TO, PDF

CLR WDT2 Pre-clear Watchdog Timer

Description The TO, PDF flags and the WDT are all cleared. Note that this instruction works in conjunction with CLR WDT1 and must be executed alternately with CLR WDT1 to have effect. Repetitively executing this instruction without alternately executing CLR WDT1 will have no effect.

Operation WDT cleared
TO ← 0
PDF ← 0

Affected flag(s) TO, PDF

CPL [m] Complement Data Memory

Description Each bit of the specified Data Memory is logically complemented (1′s complement). Bits which previously contained a 1 are changed to 0 and vice versa.

Operation [m] ← $\overline{[m]}$

Affected flag(s) Z

CPLA [m] Complement Data Memory with result in ACC

Description Each bit of the specified Data Memory is logically complemented (1′s complement). Bits which previously contained a 1 are changed to 0 and vice versa. The complemented result is stored in the Accumulator and the contents of the Data Memory remain unchanged.

Operation ACC ← $\overline{[m]}$

Affected flag(s) Z

DAA [m] Decimal-Adjust ACC for addition with result in Data Memory

Description Convert the contents of the Accumulator value to a BCD (Binary Coded Decimal) value resulting from the previous addition of two BCD variables. If the low nibble is greater than 9 or if AC flag is set, then a value of 6 will be added to the low nibble. Otherwise the low nibble remains unchanged. If the high nibble is greater than 9 or if the C flag is set, then a value of 6 will be added to the high nibble. Essentially, the decimal conversion is performed by adding 00H, 06H, 60H or 66H depending on the Accumulator and flag conditions. Only the C flag may be affected by this instruction which indicates that if the original BCD sum is greater than 100, it allows multiple precision decimal addition.

Operation [m] ← ACC + 00H or
[m] ← ACC + 06H or
[m] ← ACC + 60H or
[m] ← ACC + 66H

Affected flag(s) C

DEC [m] Decrement Data Memory

Description Data in the specified Data Memory is decremented by 1.

Operation [m] ← [m] − 1

Affected flag(s) Z

DECA [m] Decrement Data Memory with result in ACC

Description Data in the specified Data Memory is decremented by 1. The result is stored in the Accumulator. The contents of the Data Memory remain unchanged.

Operation ACC ← [m] − 1

Affected flag(s) Z

HALT Enter power down mode

Description This instruction stops the program execution and turns off the system clock. The contents of the Data Memory and registers are retained. The WDT and prescaler are cleared. The power down flag PDF is set and the WDT time-out flag TO is cleared.

Operation TO ← 0
PDF ← 1

Affected flag(s) TO, PDF

INC [m] Increment Data Memory

Description Data in the specified Data Memory is incremented by 1.

Operation [m] ← [m] + 1

Affected flag(s) Z

INCA [m] Increment Data Memory with result in ACC

Description Data in the specified Data Memory is incremented by 1. The result is stored in the Accumulator. The contents of the Data Memory remain unchanged.

Operation ACC ← [m] + 1

Affected flag(s) Z

JMP addr Jump unconditionally

Description The contents of the Program Counter are replaced with the specified address. Program execution then continues from this new address. As this requires the insertion of a dummy instruction while the new address is loaded, it is a two cycle instruction.

Operation Program Counter ← addr

Affected flag(s) None

MOV A,[m] Move Data Memory to ACC

Description The contents of the specified Data Memory are copied to the Accumulator.

Operation ACC ← [m]

Affected flag(s) None

MOV A,x Move immediate data to ACC

Description The immediate data specified is loaded into the Accumulator.

Operation ACC ← x

Affected flag(s) None

MOV [m],A Move ACC to Data Memory

Description The contents of the Accumulator are copied to the specified Data Memory.

Operation [m] ← ACC

Affected flag(s) None

NOP No operation

Description No operation is performed. Execution continues with the next instruction.

Operation No operation

Affected flag(s) None

OR A,[m] Logical OR Data Memory to ACC

Description Data in the Accumulator and the specified Data Memory perform a bitwise logical OR operation. The result is stored in the Accumulator.

Operation ACC ← ACC ″OR″ [m]

Affected flag(s) Z

OR A,x Logical OR immediate data to ACC

Description Data in the Accumulator and the specified immediate data perform a bitwise logical OR operation. The result is stored in the Accumulator.

Operation ACC ← ACC ″OR″ x

Affected flag(s) Z

ORM A,[m] Logical OR ACC to Data Memory

Description Data in the specified Data Memory and the Accumulator perform a bitwise logical OR operation. The result is stored in the Data Memory.

Operation [m] ← ACC ″OR″ [m]

Affected flag(s) Z

RET Return from subroutine

Description The Program Counter is restored from the stack. Program execution continues at the restored address.

Operation Program Counter ← Stack

Affected flag(s) None

RET A,x Return from subroutine and load immediate data to ACC

Description The Program Counter is restored from the stack and the Accumulator loaded with the specified immediate data. Program execution continues at the restored address.

Operation Program Counter ← Stack
ACC ← x

Affected flag(s) None

RETI Return from interrupt

Description The Program Counter is restored from the stack and the interrupts are re-enabled by setting the EMI bit. EMI is the master interrupt global enable bit. If an interrupt was pending when the RETI instruction is executed, the pending Interrupt routine will be processed before returning to the main program.

Operation Program Counter ← Stack
EMI ← 1

Affected flag(s) None

RL [m] Rotate Data Memory left

Description The contents of the specified Data Memory are rotated left by 1 bit with bit 7 rotated into bit 0.

Operation [m].(i+1) ← [m].i; (i=0~6)
[m].0 ← [m].7

Affected flag(s) None

RLA [m] Rotate Data Memory left with result in ACC

Description The contents of the specified Data Memory are rotated left by 1 bit with bit 7 rotated into bit 0. The rotated result is stored in the Accumulator and the contents of the Data Memory remain unchanged.

Operation ACC.(i+1) ← [m].i; (i=0~6)
ACC.0 ← [m].7

Affected flag(s) None

RLC [m] Rotate Data Memory left through Carry

Description The contents of the specified Data Memory and the carry flag are rotated left by 1 bit. Bit 7 replaces the Carry bit and the original carry flag is rotated into bit 0.

Operation [m].(i+1) ← [m].i; (i=0~6)
[m].0 ← C
C ← [m].7

Affected flag(s) C

RLCA [m] Rotate Data Memory left through Carry with result in ACC

Description Data in the specified Data Memory and the carry flag are rotated left by 1 bit. Bit 7 replaces the Carry bit and the original carry flag is rotated into the bit 0. The rotated result is stored in the Accumulator and the contents of the Data Memory remain unchanged.

Operation ACC.(i+1) ← [m].i; (i=0~6)
ACC.0 ← C
C ← [m].7

Affected flag(s) C

RR [m] Rotate Data Memory right

Description The contents of the specified Data Memory are rotated right by 1 bit with bit 0 rotated into bit 7.

Operation [m].i ← [m].(i+1); (i=0~6)
[m].7 ← [m].0

Affected flag(s) None

RRA [m] Rotate Data Memory right with result in ACC

Description: Data in the specified Data Memory and the carry flag are rotated right by 1 bit with bit 0 rotated into bit 7. The rotated result is stored in the Accumulator and the contents of the Data Memory remain unchanged.

Operation:
ACC.i ← [m].(i+1); (i=0~6)
ACC.7 ← [m].0

Affected flag(s): None

RRC [m] Rotate Data Memory right through Carry

Description: The contents of the specified Data Memory and the carry flag are rotated right by 1 bit. Bit 0 replaces the Carry bit and the original carry flag is rotated into bit 7.

Operation:
[m].i ← [m].(i+1); (i=0~6)
[m].7 ← C
C ← [m].0

Affected flag(s): C

RRCA [m] Rotate Data Memory right through Carry with result in ACC

Description: Data in the specified Data Memory and the carry flag are rotated right by 1 bit. Bit 0 replaces the Carry bit and the original carry flag is rotated into bit 7. The rotated result is stored in the Accumulator and the contents of the Data Memory remain unchanged.

Operation:
ACC.i ← [m].(i+1); (i=0~6)
ACC.7 ← C
C ← [m].0

Affected flag(s): C

SBC A,[m] Subtract Data Memory from ACC with Carry

Description: The contents of the specified Data Memory and the complement of the carry flag are subtracted from the Accumulator. The result is stored in the Accumulator. Note that if the result of subtraction is negative, the C flag will be cleared to 0, otherwise if the result is positive or zero, the C flag will be set to 1.

Operation: ACC ← ACC − [m] − C

Affected flag(s): OV, Z, AC, C

SBCM A,[m] Subtract Data Memory from ACC with Carry and result in Data Memory

Description: The contents of the specified Data Memory and the complement of the carry flag are subtracted from the Accumulator. The result is stored in the Data Memory. Note that if the result of subtraction is negative, the C flag will be cleared to 0, otherwise if the result is positive or zero, the C flag will be set to 1.

Operation: [m] ← ACC − [m] − C

Affected flag(s): OV, Z, AC, C

SDZ [m] Skip if decrement Data Memory is 0

Description: The contents of the specified Data Memory are first decremented by 1. If the result is 0 the following instruction is skipped. As this requires the insertion of a dummy instruction while the next instruction is fetched, it is a two cycle instruction. If the result is not 0 the program proceeds with the following instruction.

Operation:
[m] ← [m] − 1
Skip if [m]=0

Affected flag(s): None

SDZA [m] Skip if decrement Data Memory is zero with result in ACC

Description The contents of the specified Data Memory are first decremented by 1. If the result is 0, the following instruction is skipped. The result is stored in the Accumulator but the specified Data Memory contents remain unchanged. As this requires the insertion of a dummy instruction while the next instruction is fetched, it is a two cycle instruction. If the result is not 0, the program proceeds with the following instruction.

Operation ACC ← [m] − 1
Skip if ACC=0

Affected flag(s) None

SET [m] Set Data Memory

Description Each bit of the specified Data Memory is set to 1.

Operation [m] ← FFH

Affected flag(s) None

SET [m].i Set bit of Data Memory

Description Bit i of the specified Data Memory is set to 1.

Operation [m].i ← 1

Affected flag(s) None

SIZ [m] Skip if increment Data Memory is 0

Description The contents of the specified Data Memory are first incremented by 1. If the result is 0, the following instruction is skipped. As this requires the insertion of a dummy instruction while the next instruction is fetched, it is a two cycle instruction. If the result is not 0 the program proceeds with the following instruction.

Operation [m] ← [m] + 1
Skip if [m]=0

Affected flag(s) None

SIZA [m] Skip if increment Data Memory is zero with result in ACC

Description The contents of the specified Data Memory are first incremented by 1. If the result is 0, the following instruction is skipped. The result is stored in the Accumulator but the specified Data Memory contents remain unchanged. As this requires the insertion of a dummy instruction while the next instruction is fetched, it is a two cycle instruction. If the result is not 0 the program proceeds with the following instruction.

Operation ACC ← [m] + 1
Skip if ACC=0

Affected flag(s) None

SNZ [m].i Skip if bit i of Data Memory is not 0

Description If bit i of the specified Data Memory is not 0, the following instruction is skipped. As this requires the insertion of a dummy instruction while the next instruction is fetched, it is a two cycle instruction. If the result is 0 the program proceeds with the following instruction.

Operation Skip if [m].i ≠ 0

Affected flag(s) None

SUB A,[m] Subtract Data Memory from ACC

Description The specified Data Memory is subtracted from the contents of the Accumulator. The result is stored in the Accumulator. Note that if the result of subtraction is negative, the C flag will be cleared to 0, otherwise if the result is positive or zero, the C flag will be set to 1.

Operation ACC ← ACC − [m]

Affected flag(s) OV, Z, AC, C

SUBM A,[m] Subtract Data Memory from ACC with result in Data Memory

Description The specified Data Memory is subtracted from the contents of the Accumulator. The result is stored in the Data Memory. Note that if the result of subtraction is negative, the C flag will be cleared to 0, otherwise if the result is positive or zero, the C flag will be set to 1.

Operation [m] ← ACC − [m]

Affected flag(s) OV, Z, AC, C

SUB A,x Subtract immediate data from ACC

Description The immediate data specified by the code is subtracted from the contents of the Accumulator. The result is stored in the Accumulator. Note that if the result of subtraction is negative, the C flag will be cleared to 0, otherwise if the result is positive or zero, the C flag will be set to 1.

Operation ACC ← ACC − x

Affected flag(s) OV, Z, AC, C

SWAP [m] Swap nibbles of Data Memory

Description The low-order and high-order nibbles of the specified Data Memory are interchanged.

Operation [m].3~[m].0 ↔ [m].7~[m].4

Affected flag(s) None

SWAPA [m] Swap nibbles of Data Memory with result in ACC

Description The low-order and high-order nibbles of the specified Data Memory are interchanged. The result is stored in the Accumulator. The contents of the Data Memory remain unchanged.

Operation ACC.3~ACC.0 ← [m].7~[m].4
ACC.7~ACC.4 ← [m].3~[m].0

Affected flag(s) None

SZ [m] Skip if Data Memory is 0

Description If the contents of the specified Data Memory is 0, the following instruction is skipped. As this requires the insertion of a dummy instruction while the next instruction is fetched, it is a two cycle instruction. If the result is not 0 the program proceeds with the following instruction.

Operation Skip if [m]=0

Affected flag(s) None

SZA [m] Skip if Data Memory is 0 with data movement to ACC

Description The contents of the specified Data Memory are copied to the Accumulator. If the value is zero, the following instruction is skipped. As this requires the insertion of a dummy instruction while the next instruction is fetched, it is a two cycle instruction. If the result is not 0 the program proceeds with the following instruction.

Operation ACC ← [m]
Skip if [m]=0

Affected flag(s) None

SZ [m].i Skip if bit i of Data Memory is 0

Description If bit i of the specified Data Memory is 0, the following instruction is skipped. As this requires the insertion of a dummy instruction while the next instruction is fetched, it is a two cycle instruction. If the result is not 0, the program proceeds with the following instruction.

Operation Skip if [m].i=0

Affected flag(s) None

TABRDC [m] Read table (current page) to TBLH and Data Memory

Description The low byte of the program code (current page) addressed by the table pointer (TBLP) is moved to the specified Data Memory and the high byte moved to TBLH.

Operation [m] ← program code (low byte)
TBLH ← program code (high byte)

Affected flag(s) None

TABRDL [m] Read table (last page) to TBLH and Data Memory

Description The low byte of the program code (last page) addressed by the table pointer (TBLP) is moved to the specified Data Memory and the high byte moved to TBLH.

Operation [m] ← program code (low byte)
TBLH ← program code (high byte)

Affected flag(s) None

XOR A,[m] Logical XOR Data Memory to ACC

Description Data in the Accumulator and the specified Data Memory perform a bitwise logical XOR operation. The result is stored in the Accumulator.

Operation ACC ← ACC ″XOR″ [m]

Affected flag(s) Z

XORM A,[m] Logical XOR ACC to Data Memory

Description Data in the specified Data Memory and the Accumulator perform a bitwise logical XOR operation. The result is stored in the Data Memory.

Operation [m] ← ACC ″XOR″ [m]

Affected flag(s) Z

XOR A,x Logical XOR immediate data to ACC

Description Data in the Accumulator and the specified immediate data perform a bitwise logical XOR operation. The result is stored in the Accumulator.

Operation ACC ← ACC ″XOR″ x

Affected flag(s) Z

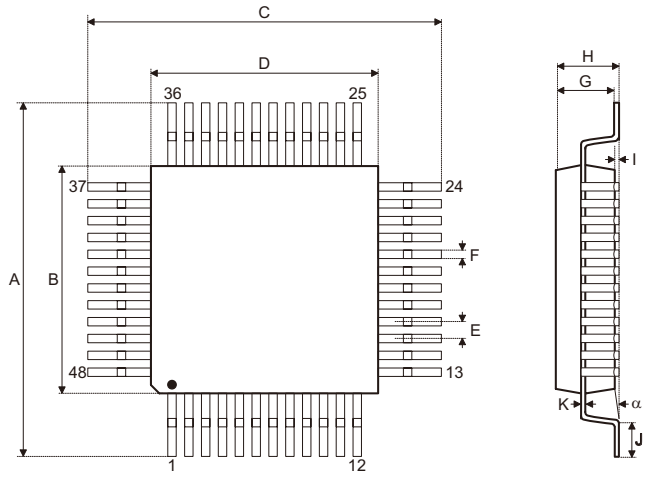
Package Information

Note that the package information provided here is for consultation purposes only. As this information may be updated at regular intervals users are reminded to consult the Holtek website for the latest version of the package information.

Additional supplementary information with regard to packaging is listed below. Click on the relevant section to be transferred to the relevant website page.

- Further Package Information (include Outline Dimensions, Product Tape and Reel Specifications)
- Packing Meterials Information
- Carton information
- PB FREE Products
- Green Packages Products

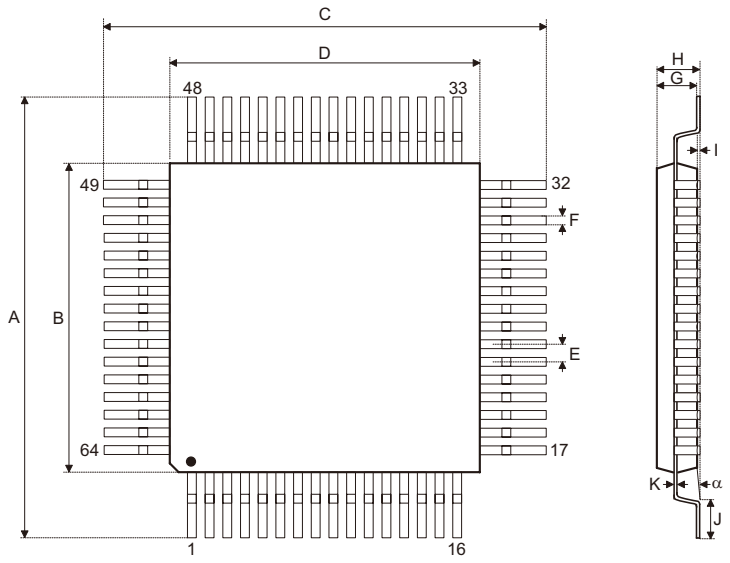
48-pin LQFP (7mm×7mm) Outline Dimensions



Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.350	—	0.358
B	0.272	—	0.280
C	0.350	—	0.358
D	0.272	—	0.280
E	—	0.020	—
F	—	0.008	—
G	0.053	—	0.057
H	—	—	0.063
I	—	0.004	—
J	0.018	—	0.030
K	0.004	—	0.008
α	0°	—	7°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	8.90	—	9.10
B	6.90	—	7.10
C	8.90	—	9.10
D	6.90	—	7.10
E	—	0.50	—
F	—	0.20	—
G	1.35	—	1.45
H	—	—	1.60
I	—	0.10	—
J	0.45	—	0.75
K	0.10	—	0.20
α	0°	—	7°

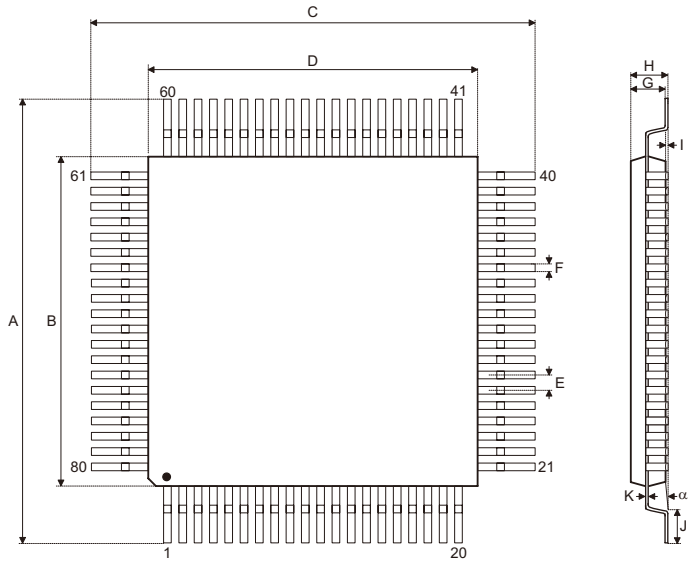
64-pin LQFP (7mm×7mm) Outline Dimensions



Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.350	—	0.358
B	0.272	—	0.280
C	0.350	—	0.358
D	0.272	—	0.280
E	—	0.016	—
F	0.005	—	0.009
G	0.053	—	0.057
H	—	—	0.063
I	0.002	—	0.006
J	0.018	—	0.030
K	0.004	—	0.008
α	0°	—	7°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	8.90	—	9.10
B	6.90	—	7.10
C	8.90	—	9.10
D	6.90	—	7.10
E	—	0.40	—
F	0.13	—	0.23
G	1.35	—	1.45
H	—	—	1.60
I	0.05	—	0.15
J	0.45	—	0.75
K	0.09	—	0.20
α	0°	—	7°

80-pin LQFP (10mm×10mm) Outline Dimensions



Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.469	—	0.476
B	0.390	—	0.398
C	0.469	—	0.476
D	0.390	—	0.398
E	—	0.016	—
F	—	0.006	—
G	0.053	—	0.057
H	—	—	0.063
I	—	0.004	—
J	0.018	—	0.030
K	0.004	—	0.008
α	0°	—	7°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	11.90	—	12.10
B	9.90	—	10.10
C	11.90	—	12.10
D	9.90	—	10.10
E	—	0.40	—
F	—	0.16	—
G	1.35	—	1.45
H	—	—	1.60
I	—	0.10	—
J	0.45	—	0.75
K	0.10	—	0.20
α	0°	—	7°

Holtek Semiconductor Inc. (Headquarters)
No.3, Creation Rd. II, Science Park, Hsinchu, Taiwan
Tel: 886-3-563-1999
Fax: 886-3-563-1189
http://www.holtek.com.tw

Holtek Semiconductor Inc. (Taipei Sales Office)
4F-2, No. 3-2, YuanQu St., Nankang Software Park, Taipei 115, Taiwan
Tel: 886-2-2655-7070
Fax: 886-2-2655-7373
Fax: 886-2-2655-7383 (International sales hotline)

Holtek Semiconductor (China) Inc. (Dongguan Sales Office)
Building No.10, Xinzhu Court, (No.1 Headquarters), 4 Cuizhu Road, Songshan Lake, Dongguan, China 523808
Tel: 86-769-2626-1300
Fax: 86-769-2626-1311, 86-769-2626-1322

Holtek Semiconductor (USA), Inc. (North America Sales Office)
46729 Fremont Blvd., Fremont, CA 94538, USA
Tel: 1-510-252-9880
Fax: 1-510-252-9885
http://www.holtek.com