HOLTEK

2-Wire Communication Flash MCU

HT45F2002

Revision: V1.00 Date: July 14, 2016

www.holtek.com

Features

MCU CPU Features

- Operating voltage
  - f_{SYS}=8MHz: 2.2V~5.5V
  - f_{SYS}=12MHz: 2.7V~5.5V
  - f_{SYS}=16MHz: 3.3V~5.5V
- Up to 0.25μs instruction cycle with 16MHz system clock at V_{DD}=5V
- Power down and wake-up functions to reduce power consumption
- Two oscillators
  - Internal RC − HIRC
  - Internal 32kHz RC − LIRC
- Fully intergrated internal 8/12/16MHz oscillator requires no external components
- Multi-mode operation: NORMAL, SLOW, IDLE and SLEEP
- All instructions executed in one or two instruction cycles
- Table read instructions
- 63 powerful instructions
- Up to 6-level subroutine nesting
- Bit manipulation instruction

MCU Peripheral Features

- Flash Program Memory: 2K×15
- RAM Data Memory: 96×8
- EEPROM Memory: 32×8
- Watchdog Timer function
- Up to 11 bidirectional I/O lines
- Software controlled SCOM lines LCD driver with 1/2 bias
- Multiple Timer Modules for time measure, compare match output, capture input, PWM output, single pulse output functions
- Dual Time-Base functions for generation of fixed time interrupt signals
- 5-channel 12-bit resolution A/D converter
- Serial Interface Module − SPI or I^2C
- Programmable I/O port source current for LED driving applications
- Low voltage reset function
- Flash program memory can be re-programmed up to 100,000 times
- Flash program memory data retention > 10 years
- EEPROM data memory can be re-programmed up to 1,000,000 times
- EEPROM data memory data retention > 10 years
- Package type: 20-pin SSOP

Two Line Type Power Line Data Transmitter Features

- Complete data transmission on power line functions

- High maximum input voltage: 42V
- Build-in 5V LDO output
- Integrated low dropout voltage regulator with soft-start and short circuit protection
- Integrated voltage detector for power supply monitoring
- Integrated comparator
- Open drain NMOS driver for flexible interfacing
- Power and reset protection features
- Minimal external component requirements

General Description

This device is Flash Memory type 8-bit high performance RISC architecture microcontroller. Offering users the convenience of Flash Memory multi-programming features, this device also includes a wide range of functions and features. Other memory includes an area of RAM Data Memory as well as an area of EEPROM memory for storage of non-volatile data such as serial numbers, calibration data etc.

Analog features include a multi-channel 12-bit A/D converter function. Multiple and extremely flexible Timer Modules provide timing, pulse generation, capture input, compare match output, single pulse output and PWM generation functions. Communication with the outside world is managed by including fully integrated SPI and I^{2}C interface functions, two popular interfaces which provide designers with a means of easy communication with external peripheral hardware. Protective features such as an internal Watchdog Timer and Low Voltage Reset coupled with excellent noise immunity and ESD protection ensure that reliable operation is maintained in hostile electrical environments.

This device also contains a two line type power line data transceiver. In systems where a master controller controls a number of individual interconnected subsystems such as found in smoke detector systems, water metering systems, solar energy system, etc., the cost of the extensive interconnecting cabling can be a major factor. By sending data along the power supply lines, the interconnecting cables can be reduced to a simple two line type, thus greatly reducing both cable and installation costs.

With the addition of a few external components, this power line data transceiver device contains all the internal components required to provide users with a system for power line data transmission and reception. Data is modulated onto the power line by the simple reduction of the power line voltage for a specific period of time. Power supply voltage changes can be initiated by the master controller for data reception or initiated by the power line data transceiver for data transmission. An internal voltage regulator with Soft-Start and short circuit protection function within the device ensures that a constant voltage power supply is provided to the interconnected subsystem units while an internal voltage detector monitors the power line voltage level. An internal comparator is used to translate the differential signal into a logic signal for the MCU.

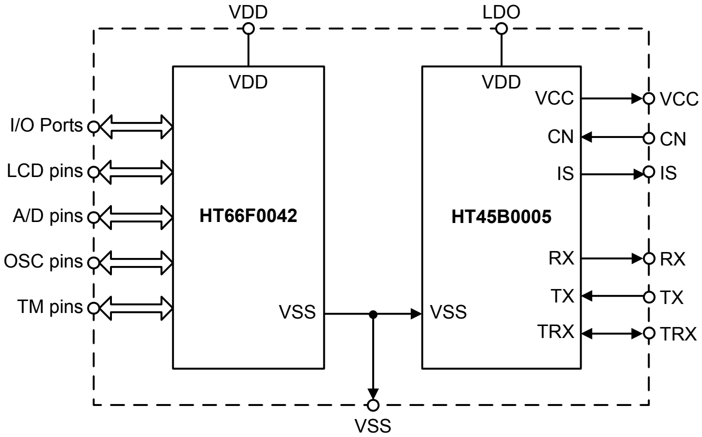
A full choice of internal low and high speed, oscillator functions are provided including fully integrated system oscillators which require no external components for their implementation. The ability to operate and switch dynamically between a range of operating modes using different clock sources gives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

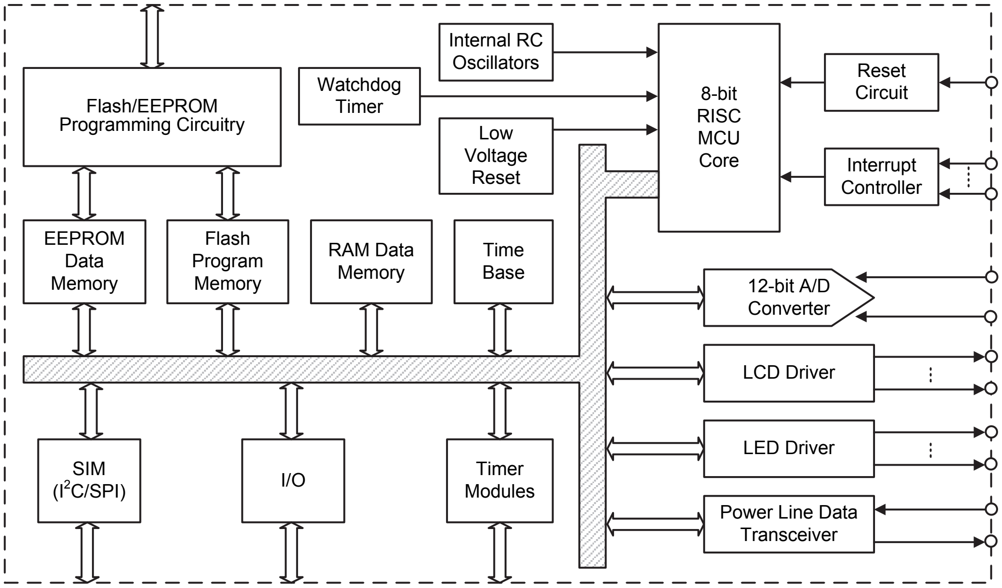
The SPI and I^{2}C interfaces offer possibilities for data communication networks between microcontrollers, low-cost data links between PCs and peripheral devices, portable and battery operated device communication, etc.

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

Block Diagram

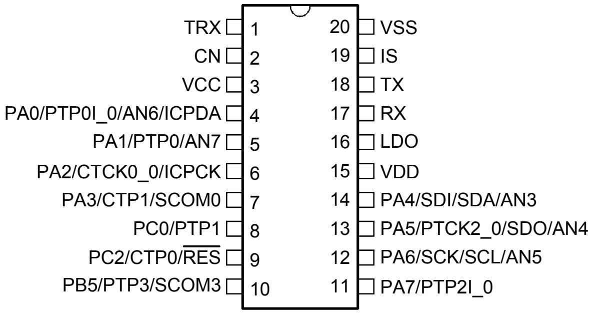
The following block diagram illustrates the dual-chip structure of the device, where an individual MCU and a two-line type power line data transceiver chip are combined into a single package.





Note: The Power Line Data Transceicer is the IC of the dual-chip in one device.

Pin Assignment



HT45F2002
20 SSOP-A

Note: The PB0~PB4, PB5, PC, PC3~PC6 and their pin-shared functions are not connected to external package pins, care must therefore be taken to manage them properly with the application program.

Pin Descriptions

With the exception of the power pins and some relevant power line data transceiver pins, all pins on this device can be referenced by its Port name, e.g. PA.0, PA.1 etc, which refer to the digital I/O function of the pins. However these Port pins are also shared with other function such as the Analog to Digital Converter, Timer Module pins etc. The function of each pin is listed in the following table, however the details behind how each pin is configured is contained in other sections of the datasheet.

Pin Name	Function	OPT	I/T	O/T	Description
PA0/PTP0I_0/AN6/ICPDA	PA0	PAWU PAPU PAS0	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	PTP0I_0	PAS0 PTM0C0	ST	—	PTM0 input
	AN6	PAS0	AN	—	A/D Converter input channel 6
	ICPDA	—	ST	CMOS	ICP Data Line
PA1/PTP0/AN7	PA1	PAWU PAPU PAS0	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	PTP0	PAS0	—	CMOS	PTM0 output
	AN7	PAS0	AN	—	A/D Converter input channel 7
PA2/CTCK0_0/ICPCK	PA2	PAWU PAPU	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	CTCK0_0	CTM0C0	ST	—	CTM0 clock input
	ICPCK	—	ST	—	ICP Clock Line
PA3/CTP1/SCOM0	PA3	PAWU PAPU PAS0	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	CTP1	PAS0	—	CMOS	CTM1output
	SCOM0	PAS0	—	AN	LCD driver output for LCD panel common

Pin Name	Function	OPT	I/T	O/T	Description
PA4/SDI/SDA/AN3	PA4	PAWU PAPU PAS1	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	SDI	PAS1 SIMC0	ST	—	SPI serial data input
	SDA	PAS1 SIMC0	ST	NMOS	I^{2}C data line
	AN3	PAS1	AN	—	A/D Converter input channel 3
PA5/PTCK2_0/ SDO/AN4	PA5	PAWU PAPU PAS1	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	PTCK2_0	PAS1 PTM2C0 IFS	ST	—	PTM2 clock input
	SDO	PAS1	—	CMOS	SPI serial data output
	AN4	PAS1	AN	—	A/D Converter input channel 4
PA6/SCK/SCL/AN5	PA6	PAWU PAPU PAS1	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	SCK	PAS1 SIMC0	ST	CMOS	SPI serial clock
	SCL	PAS1 SIMC0	ST	NMOS	I^{2}C clock line
	AN5	PAS1	AN	—	A/D Converter input channel 5
PA7/PTP2I_0	PA7	PAWU PAPU	ST	CMOS	General purpose I/O. Register enabled pull-up and wake-up
	PTP2I_0	PTM2C0 IFS	ST	—	PTM2 iutput
PB5/PTP3/SCOM3	PB5	PBPU PBS1	ST	CMOS	General purpose I/O. Register enabled pull-up
	PTP3	PBS1	—	CMOS	PTM3 output
	SCOM3	PBS1	—	AN	LCD driver output for LCD panel common
PC0/PTP1	PC0	PCPU PCS0	ST	CMOS	General purpose I/O. Register enabled pull-up
	PTP1	PCS0	—	CMOS	PTM1 output
PC2/CTP0/$\overline{\text{RES}}$	PC2	PCPU PCS0 RSTC	ST	CMOS	General purpose I/O. Register enabled pull-up
	CTP0	PCS0 RSTC	—	CMOS	CTM0 output
	$\overline{\text{RES}}$	RSTC	ST	—	External reset input
CN	CN	—	ST	—	Comparator Negative Input
TRX	TRX	—	ST	CMOS	Transceiver signal detect/modulate
IS	IS	—	—	CMOS	Source terminal of constant current NMOS driver
TX	TX	—	ST	—	Input pin for constant current modulate
RX	RX	—	—	CMOS	Comparator output, transmitter signal detect output
LDO	LDO	—	—	CMOS	LDO Voltage output

Pin Name	Function	OPT	I/T	O/T	Description
VCC	VCC	—	PWR	—	Input voltage
VDD	VDD	—	PWR	—	Digital positive power supply. Could connect to LDO pin externally
VSS	VSS	—	PWR	—	Digital negative power supply. Ground pin - VSS

Legend: I/T: Input type
O/T: Output type
OPT: Optional by register option
PWR: Power
ST: Schmitt Trigger input
AN: Analog signal
CMOS: CMOS output
NMOS: NMOS output
Note: The PB0~PB4, PB5, PC1, PC3~PC6 and their pin-shared functions are not connected to external package pins.

Internally Connected Pins

Except the pins mentioned in the table above several pins are not connected to external package pins. These pins are interconnection pins between the MCU and the power line data transceiver chips and are listed in the following table. The description is provided from the power line data transceiver chip standpoint.

Power Line Data Transceiver Pin Name	Type	Description
VSS	PWR	Ground Pin. Connected to MCU negative power supply, VSS

Absolute Maximum Ratings

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

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

MCU D.C. Characteristics

Ta = 25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	Operating Voltage (HIRC)	—	f_{SYS}=f_{HIRC}=8MHz	2.2	—	5.5	V
			f_{SYS}=f_{HIRC}=12MHz	2.7	—	5.5	V
			f_{SYS}=f_{HIRC}=16MHz	3.3	—	5.5	V
	Operating Voltage (LIRC)	—	f_{SYS}=f_{LIRC}=32kHz	2.2	—	5.5	V
I_{DD}	Operating Current (HIRC)	3V	No load, f_{SYS}=f_{HIRC}=8MHz, ADC off, WDT enable, LVR enable	—	1.0	2.0	mA
		5V		—	2.5	4.0	mA
		3V	No load, f_{SYS}=f_{HIRC}=12MHz, ADC off, WDT enable, LVR enable	—	1.5	2.5	mA
		5V		—	3.5	5.5	mA
		3V	No load, f_{SYS}=f_{HIRC}=16MHz, ADC off, WDT enable, LVR enable	—	2.0	3.0	mA
		5V		—	4.5	7.0	mA
	Operating Current (LIRC)	3V	No load, f_{SYS}=f_{LICRC}=32kHz, ADC off, WDT enable, LVR enable	—	15	30	μA
		5V		—	30	60	μA
	Operating Current, Normal Mode, f_H=16MHz (HIRC)	3V	No load, f_{SYS}=f_H/2, ADC off, WDT enable, LVR enable	—	2.0	3.0	mA
		5V		—	3.0	4.0	mA
		3V	No load, f_{SYS}=f_H/64, ADC off, WDT enable, LVR enable	—	1.0	1.6	mA
		5V		—	1.6	2.2	mA
	Operating Current, Normal Mode, f_H=12MHz (HIRC)	3V	No load, f_{SYS}=f_H/2, ADC off, WDT enable, LVR enable	—	1.5	2.2	mA
		5V		—	2.2	3.0	mA
		3V	No load, f_{SYS}=f_H/64, ADC off, WDT enable, LVR enable	—	0.7	1.2	mA
		5V		—	1.2	1.6	mA
	Operating Current, Normal Mode, f_H=8MHz (HIRC)	3V	No load, f_{SYS}=f_H/2, ADC off, WDT enable, LVR enable	—	1.0	1.5	mA
		5V		—	1.5	2.0	mA
		3V	No load, f_{SYS}=f_H/64, ADC off, WDT enable, LVR enable	—	0.5	0.8	mA
		5V		—	0.8	1.1	mA
I_{STB}	SLEEP0 Mode Standby Current (LIRC Off)	3V	No load, all peripherals off, WDT off	—	0.2	0.8	μA
		5V		—	0.5	1.0	μA
	SLEEP1 Mode Standby Current (LIRC On)	3V	No load, ADC off, WDT enable, LVR disable	—	1.3	3.0	μA
		5V		—	5.0	10	μA
	IDLE0 Mode Standby Current (LIRC On)	3V	No load, ADC off, WDT enable, LVR disable	—	1.3	3.0	μA
		5V		—	5.0	10	μA
	IDLE1 Mode Standby Current (HIRC On)	3V	No load, ADC off, WDT enable, f_{SYS}=f_{HIRC}=8MHz on	—	0.8	1.6	mA
		5V		—	1.0	2.0	mA
		3V	No load, ADC off, WDT enable, f_{SYS}=f_{HIRC}=12MHz on	—	1.2	2.4	mA
		5V		—	1.5	3.0	mA
		3V	No load, ADC off, WDT enable, f_{SYS}=f_{HIRC}=16MHz on	—	1.6	3.2	mA
		5V		—	2.0	4.0	mA
V_{IL}	Input Low Voltage for I/O Ports or Input Pins except $\overline{RES}$ Pin	5V	—	0	—	1.5	V
		—	—	0	—	$0.2V_{DD}$	V
	Input Low Voltage for $\overline{RES}$ Pin	—	—	0	—	$0.4V_{DD}$	V
V_{IH}	Input High Voltage for I/O Ports or Input Pins except $\overline{RES}$ Pin	5V	—	3.5	—	5.0	V
		—	—	$0.8V_{DD}$	—	V_{DD}	V
	Input High Voltage for $\overline{RES}$ Pin	—	—	$0.9V_{DD}$	—	V_{DD}	V

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
I_{OL}	I/O Port Sink Current	3V	$V_{OL}=0.1V_{DD}$	18	36	—	mA
		5V	$V_{OL}=0.1V_{DD}$	40	80	—	mA
I_{OH}	I/O Port, Source Current	3V	$V_{OH}=0.9V_{DD}$, PxPS=00	-1.0	-2.0	—	mA
		5V	$V_{OH}=0.9V_{DD}$, PxPS=00	-2.0	-4.0	—	mA
		3V	$V_{OH}=0.9V_{DD}$, PxPS=01	-1.75	-3.5	—	mA
		5V	$V_{OH}=0.9V_{DD}$, PxPS=01	-3.5	-7.0	—	mA
		3V	$V_{OH}=0.9V_{DD}$, PxPS=10	-2.5	-5.0	—	mA
		5V	$V_{OH}=0.9V_{DD}$, PxPS=10	-5.0	-10	—	mA
		3V	$V_{OH}=0.9V_{DD}$, PxPS=11	-5.5	-11	—	mA
		5V	$V_{OH}=0.9V_{DD}$, PxPS=11	-11	-22	—	mA
R_{PH}	Pull-high Resistance for I/O Ports	3V	—	20	60	100	kΩ
		5V	—	10	30	50	kΩ

MCU A.C. Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Condition				
f_{SYS}	System Clock (HIRC)	2.2V~5.5V	$f_{SYS}=f_{HIRC}=8MHz$	—	8	—	MHz
		2.7V~5.5V	$f_{SYS}=f_{HIRC}=12MHz$	—	12	—	MHz
		3.3V~5.5V	$f_{SYS}=f_{HIRC}=16MHz$	—	16	—	MHz
	System Clock (LIRC)	2.2V~5.5V	$f_{SYS}=f_{LIRC}=32kHz$	—	32	—	kHz
f_{HIRC}	High Speed Internal RC Oscillator (HIRC Trim @ V_{DD}=3V/5V, Ta=25°C)	5V	Ta= -20°C to 50°C	-2%	8	+2%	MHz
				-2%	12	+2%	MHz
				-2%	16	+2%	MHz
		2.2V~5.5V	Ta= -40°C to 85°C	-3%	8	+3%	MHz
		2.7V~5.5V		-3%	12	+3%	MHz
		3.3V~5.5V		-3%	16	+3%	MHz
f_{LIRC}	System Clock (32kHz RC Oscillator)	2.2V~5.5V	Ta= -40°C to 85°C	8	32	50	kHz
t_{TCK}	xTCKn, PTPnI Input Pulse Width	—	—	0.3	—	—	μs
t_{RES}	External Reset Low Pulse Width	—	—	10	—	—	μs
t_{INT}	Interrupt Pulse Width	—	—	0.3	—	—	μs
t_{EERD}	EEPROM Read Time	—	—	—	2	4	t_{SYS}
t_{EEWR}	EEPROM Write Time	—	—	—	2	6.5	ms

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Condition				
t_{SST}	System Start-up Timer Period (Wake-up from HALT, f_{SYS} Off at HALT State, Reset Pin Reset)	—	f_{SYS}=HIRC	16	—	—	t_{SYS}
			f_{SYS}=LIRC	2	—	—	
	System Start-up Timer Period (Wake-up from HALT, f_{SYS} on at HALT State)	—	—	2	—	—	t_{SYS}
t_{RSTD}	System Reset Delay Time (Power On Reset, LVR Reset, WDT S/W Reset (WDTC)	—	—	6.25	50	125	ms
	System Reset Delay Time ($\overline{\text{RES}}$ Pin Reset, WDT Normal Reset)	—	—	2.08	16.7	47.9	ms
f_{I2C}	I^2C Standard Mode (100kHz) f_{SYS} Frequency	—	No clock debounce	2	—	—	MHz
		—	2 system clock debounce	4	—	—	MHz
		—	4 system clock debounce	8	—	—	MHz
	I^2C Fast Mode (400kHz) f_{SYS} Frequency	—	No clock debounce	5	—	—	MHz
		—	2 system clock debounce	10	—	—	MHz
		—	4 system clock debounce	20	—	—	MHz

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

2. To maintain the accuracy of the internal HIRC oscillator frequency, a 0.1µF decoupling capacitor should be connected between VDD and VSS and located as close to the device as possible.

ADC Electrical Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
AV_{DD}	A/D Converter Operating Voltage	—	—	2.7	—	5.5	V
V_{ADI}	A/D Converter Input Voltage	—	—	0	—	AV_{DD} /V_{REF}	V
V_{REF}	A/D Converter Reference Voltage	3V	—	2	—	AV_{DD}	V
		5V					
DNL	Differential Non-linearity	2.7V	V_{REF}=AV_{DD}=V_{DD} t_{ADCK} =0.5μs	-3	—	+3	LSB
		3V					
		5V					
INL	Integral Non-linearity	2.7V	V_{REF}=AV_{DD}=V_{DD} t_{ADCK} =0.5μs	-4	—	+4	LSB
		3V					
		5V					
I_{ADC}	Additional Power Consumption if A/D Converter is used	3V	No load (t_{ADCK} =0.5μs)	—	1.0	2.0	mA
		5V	No load (t_{ADCK} =0.5μs)	—	1.5	3.0	mA
t_{ADCK}	A/D Converter Clock Period	2.7V~ 5.5V	—	0.5	—	10	μs
t_{ADC}	A/D Conversion Time (Include Sample and Hold Time)	2.7~ 5.5V	12-bit ADC	16	—	20	t_{ADCK}
t_{ADS}	A/D Converter Sampling Time	2.7V~ 5.5V	—	—	4	—	t_{ADCK}
t_{ON2ST}	A/D Converter On-to-Start Time	2.7V~ 5.5V	—	4	—	—	μs

LVR Electrical Characteristics

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	Operating Voltage	—	—	1.9	—	5.5	V
V_{LVR}	Low Voltage Reset Voltage	—	LVR Enable, 2.1V option	-5%	2.1	+5%	V
BUFO	Reference Output with Buffer	—	T_J = +25°C@3.15V	-5%	1.04	+5%	V
t_{LVR}	Low Voltage Width to Reset	—	—	160	320	640	μs

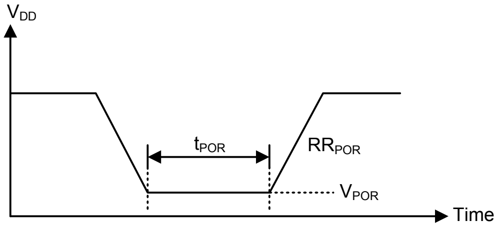
LCD Electrical Characteristics

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
I_{BIAS}	V_{DD}/2 Bias Current for LCD	5V	ISEL[1:0]=00	17.5	25.0	32.5	μA
			ISEL[1:0]=01	35	50	65	μA
			ISEL[1:0]=10	70	100	130	μA
			ISEL[1:0]=11	140	200	260	μA
V_{SCOM}	V_{DD}/2 Voltage for LCD COM Port	2.2V~5.5V	No load	0.475	0.500	0.525	V_{DD}

Power on Reset Electrical 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_{POR}	V_{DD} Rising 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



Two Line Type Power Line Data Transmitter D.C. Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{CC}	Operating Voltage	—	—	7	—	42	V
I_{CC}	Operating Current	—	V_{CC}=42V, V_{DD} No Load,	—	20	40	μA
I_{OFF}	Offline Current	—	V_{cc}=42V, V_{DD} No Load, TRX=0V	—	10	20	μA
V_{OFF}	TRX Offline Voltage	—	—	—	—	0.5	V
V_{ON}	TRX Online Voltage	—	—	4	—	—	V
V_T	Threshold Voltage	—	—	—	V_{MARK}-5.6	—	V
I_{MC}	Modulate Current	—	R_S=100Ω	—	15	—	mA
		—	R_S=47Ω	—	32	—	mA
V_{IL}	Input low voltage for TX pin	5V	—	0	—	1.5	V
V_{IH}	Input high voltage for TX pin	5V	—	3.5	—	5	V
I_{OL}	Sink current for RX pin	5V	V_{OL}=0.1V_{DD}	10	20	—	mA
I_{OH}	Source current for RX pin	5V	V_{OH}=0.9V_{DD}	-5	-10	—	mA
R_{PH}	Pull-high Resistance for TX	—	—	-30%	50	30%	kΩ

LDO Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{OUT}	Output Voltage	5V	V_{CC}=7V, I_{LOAD}=10mA	4.85	5	5.15	V
I_{OUT}	Output Current	—	V_{CC}=10V, ΔV_{OUT}=-3%	60	—	—	mA
			V_{CC}=7V, ΔV_{OUT}=-3%	30	—	—	mA
ΔV_{LINE}	Line Regulation	—	7V ≤ V_{IN} ≤ 42V, I_{LOAD}=1mA	—	—	0.2	%/V
TC	Temperature Coefficient	5V	Ta=-40°C ~ 85°C, V_{CC}=7V, I_{LOAD}=10mA	—	±0.75	±1.5	mV/°C
ΔV_{OUT_RIPPLE}	Output Voltage Ripple	—	V_{CC}=7V, I_{LOAD}=10mA	—	—	40	mV
t_{START}	LDO Startup Time	5V	V_{CC}=7V, I_{LOAD}=1mA, V_{OUT}=5V ± 3%	—	—	10	ms
I_{OL}	Sink current for VDD	—	V_{CC}=5V, V_{OL}=0.5V	0.8	—	—	mA

LVD Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{LVD}	Low voltage detection voltage	—	—	5.7	6.0	6.3	V
V_{HYS}	Hysteresis Voltage	—	—	—	0.5	—	V
TC	Temperature coefficient ($\Delta V_{LVD}/\Delta Ta$)	—	Ta=-40°C ~ 85°C	—	±0.9	—	mV/°C

Comparator Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
A_{OL}	Open loop gain	—	—	60	80	—	dB
V_{HYS}	Hysteresis	—	—	—	0.15	—	V
t_{RP}	Response time	—	—	—	—	5	μs

Constant Current Modulator Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
t_{RP}	Response Time	—	No Load	—	—	5	μs

Functional Description

As this device package contains multiple internal chips, for a detailed functional description, users must refer to the relevant individual datasheets for both the MCU and the power line data transceiver chip. The following table shows what devices are inside for this package.

Device	MCU	Power Line Data Transceiver
HT45F2002	HT66F0042	HT45B0005

Multi-chip Internal Device

Although most of the functional description material will be located in the individual datasheets, there are some special considerations which need to be taken into account when using multi-chip device. These points will be mentioned in the following sections

Multi-chip Hardware Considerations

As the single-package-multi-chip device is composed of an individual MCU and the power line data transceiver chip, using them together requires the user to take care of some special points.

Absolute Maximum Rating

The Absolute Maximum Ratings for the two individual chips must be checked for discrepancies and the necessary care taken in device handling and usage.

Power Supply and Consumption

Examination of the block diagram will reveal that the power line data transceiver chip Ground pin, VSS, has internal connection to the MCU Ground pin, VSS. For this reason these two pins do not need to be connected externally. The single-package-multi-chip device LDO pin is the power line data transceiver chip LDO output pin, VDD, while the single-package-multi-chip device VDD pin is the MCU power supply VDD pin.

To calculate the power consumption for the device, the total operating current is the sum of the operating current for the MCU specified in the MCU datasheet and the operating current for the power line data transceiver chip listed in its datasheet. Similarly, the standby current is the sum of the two individual chip standby currents.

Power Down and Wake up

The MCU power-down and wake-up functions are covered in the relevant MCU datasheet. However, the power line data transceiver chip is always operating after a power-up. Therefore, no power-down issue for the power line data transceiver chip is controlled by the relevant MCU device.

Unbonded MCU Pins

Examination of the relevant MCU datasheet will reveal that not all of the MCU I/O port lines are bonded out to external pins. As a result special attention regarding initialisation procedures should be paid to these port lines. Users should ensure that these pins are setup in input states with pull high resistors or in output states with either a high or low levels to avoid additional power consumption resulting from floating input pins.

Multi-chip Programming Considerations

To use the power line data transceiver function, several important steps must be implemented to ensure that the power line data transceiver chip operates normally:

The MCU power line, VSS, is internally connected to the power line data transceiver VSS pin. For this reason, these two pins no need to be connected externally.

Other corresponding programming considerations may be covered in the MCU and power line data transceiver datasheets respectively. Refer to all information in the individual datasheet for desired applications.

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 microcontroller, 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 several 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 such as 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 Operation

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 which 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	1^{Note}	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	1^{Note}	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	1^{Note}	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	1^{Note}	Z, C, AC, OV
DAA [m]	Decimal adjust ACC for Addition with result in Data Memory	1^{Note}	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	1^{Note}	Z
ORM A,[m]	Logical OR ACC to Data Memory	1^{Note}	Z
XORM A,[m]	Logical XOR ACC to Data Memory	1^{Note}	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	1^{Note}	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	1^{Note}	Z
DECA [m]	Decrement Data Memory with result in ACC	1	Z
DEC [m]	Decrement Data Memory	1^{Note}	Z
Rotate			
RRA [m]	Rotate Data Memory right with result in ACC	1	None
RR [m]	Rotate Data Memory right	1^{Note}	None
RRCA [m]	Rotate Data Memory right through Carry with result in ACC	1	C
RRC [m]	Rotate Data Memory right through Carry	1^{Note}	C
RLA [m]	Rotate Data Memory left with result in ACC	1	None
RL [m]	Rotate Data Memory left	1^{Note}	None
RLCA [m]	Rotate Data Memory left through Carry with result in ACC	1	C
RLC [m]	Rotate Data Memory left through Carry	1^{Note}	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	1^{Note}	None
MOV A,x	Move immediate data to ACC	1	None
Bit Operation			
CLR [m].i	Clear bit of Data Memory	1^{Note}	None
SET [m].i	Set bit of Data Memory	1^{Note}	None
Branch Operation			
JMP addr	Jump unconditionally	2	None
SZ [m]	Skip if Data Memory is zero	1^{Note}	None
SZA [m]	Skip if Data Memory is zero with data movement to ACC	1^{Note}	None
SZ [m].i	Skip if bit i of Data Memory is zero	1^{Note}	None
SNZ [m].i	Skip if bit i of Data Memory is not zero	1^{Note}	None
SIZ [m]	Skip if increment Data Memory is zero	1^{Note}	None
SDZ [m]	Skip if decrement Data Memory is zero	1^{Note}	None
SIZA [m]	Skip if increment Data Memory is zero with result in ACC	1^{Note}	None
SDZA [m]	Skip if decrement Data Memory is zero with result in ACC	1^{Note}	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 Operation			
TABRD [m]	Read table (specific page) to TBLH and Data Memory	2^{Note}	None
TABRDC [m]	Read table (current page) to TBLH and Data Memory	2^{Note}	None
TABRDL [m]	Read table (last page) to TBLH and Data Memory	2^{Note}	None
Miscellaneous			
NOP	No operation	1	None
CLR [m]	Clear Data Memory	1^{Note}	None
SET [m]	Set Data Memory	1^{Note}	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	1^{Note}	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	$[m] \leftarrow \overline{[m]}$
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 is 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

TABRD [m]	Read table (specific page) to TBLH and Data Memory
Description	The low byte of the program code (specific page) addressed by the table pointer pair (TBHP and 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

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/Carton Information.

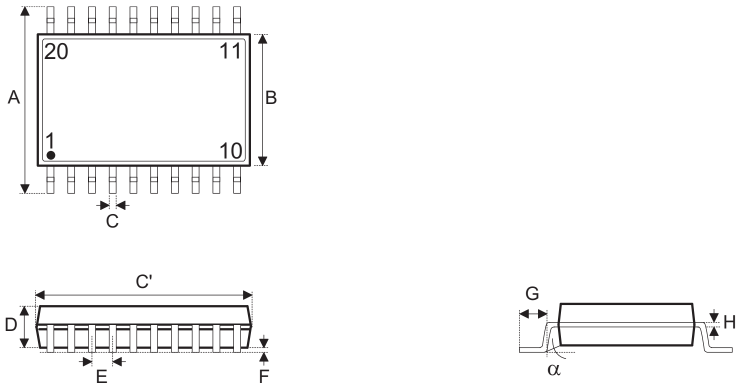
Additional supplementary information with regard to packaging is listed below. Click on the relevant section to be transferred to the relevant website page.

- Package Information (include Outline Dimensions, Product Tape and Reel Specifications)

- The Operation Instruction of Packing Materials

- Carton information

20-pin SSOP (150mil) Outline Dimensions



Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	—	0.236 BSC	—
B	—	0.154 BSC	—
C	0.008	—	0.012
C’	—	0.341 BSC	—
D	—	—	0.069
E	—	0.025 BSC	—
F	0.004	—	0.0098
G	0.016	—	0.05
H	0.004	—	0.01
α	0°	—	8°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	—	6.000 BSC	—
B	—	3.900 BSC	—
C	0.20	—	0.30
C’	—	8.660 BSC	—
D	—	—	1.75
E	—	0.635 BSC	—
F	0.10	—	0.25
G	0.41	—	1.27
H	0.10	—	0.25
α	0°	—	8°