Features

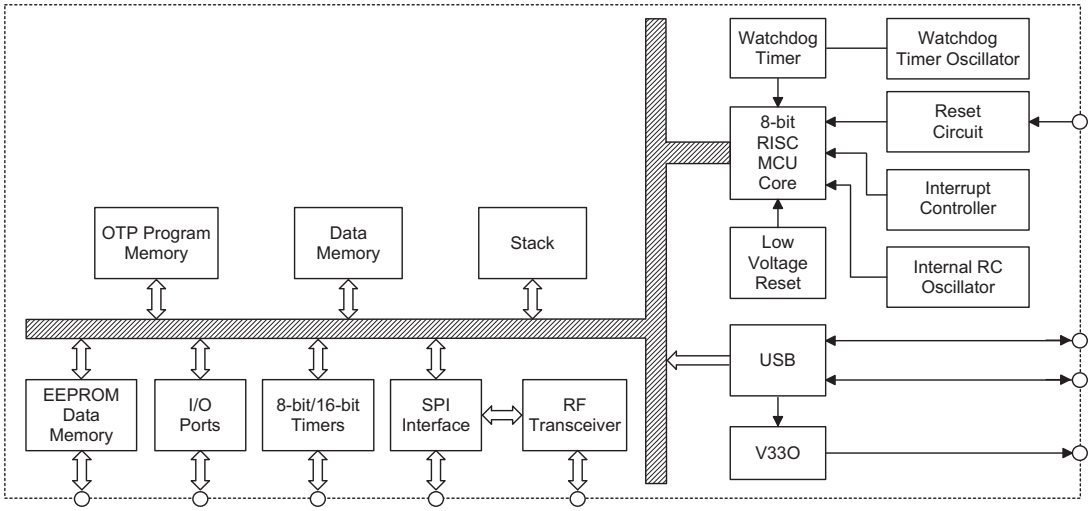
- Operating voltage:
  f_{SYS}=6M/12MHz: 3.3V~5.5V
- 4096×15 program memory
- 160×8 data memory RAM
- 128×8 EEPROM data memory
- USB 2.0 low speed function
- 3 endpoints supported - endpoint 0 included
- PS2 and USB modes supported
- Low voltage reset function
- 9 bidirectional I/O lines (max.)
- 8-bit programmable timer/event counter with overflow interrupt
- 16-bit programmable timer/event counter and overflow interrupts
- Watchdog Timer
- RF Transceiver with 2.4GHz RF frequency
- Integrated 1.5kΩ resistor between V33O and D- pins for USB applications
- Fully integrated 6MHz or 12MHz oscillator
- All I/O pins have wake-up functions
- Power-down function and wake-up feature reduce power consumption
- 8-level subroutine nesting
- Up to 0.33μs instruction cycle with 12MHz system clock at V_{DD}=5V
- Bit manipulation instruction
- 15-bit table read instruction
- 63 powerful instructions
- All instructions in one or two machine cycles
- 40-pin QFN package

General Description

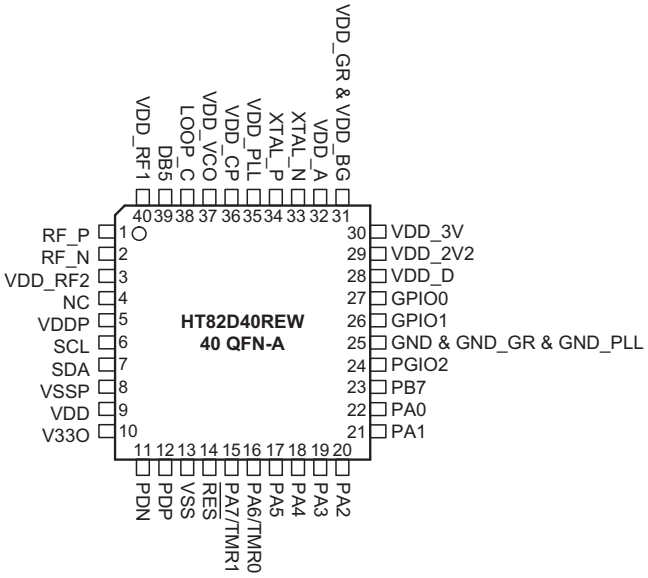
The HT82D40REW is 8-bit high performance, RISC architecture microcontroller device specifically designed for multiple I/O control product applications.

The advantages of low power consumption, I/O flexibility, timer functions, integrated USB interface, Power Down and wake-up functions, Watchdog timer etc, make the devices extremely suitable for use in computer peripheral product applications as well as many other applications such as industrial control, consumer products, subsystem controllers, etc.

Block Diagram



Pin Assignment



Pin Description

Pin Name	I/O	Options	Description
PA0~PA5 PA6/TMR0 PA7/TMR1	I/O	Pull-high Wake-up NMOS/CMOS/PMOS	Bidirectional 8-bit input/output port. Each pin can be configured as a wake-up input by a configuration option. Software instructions determine if the pin is an output or Schmitt Trigger input. Configuration options determine if the output structures are CMOS, NMOS or PMOS types. Each pin can be configured as an input with or without a pull-high resistor by a configuration option. TMR0 and TMR1 are pin-shared with PA6 and PA7, respectively.
PB0/$\overline{\text{INT}}$ PB1/$\overline{\text{RST}}$ PB2/WAKE PB3/SDI PB4/SDO PB5/$\overline{\text{SCK}}$ PB6/$\overline{\text{SCS}}$ PB7	I/O	Pull-high Wake-up	Bidirectional 8-bit input/output port. Each pin can be configured as a wake-up input by a configuration option. Software instructions determine if the pin is a CMOS output or Schmitt Trigger input. Each pin can be configured as an input with or without a pull-high resistor by a configuration option. PB3~PB6 are pin-shared with the SPI interface pins named SDI, SDO, SCK and $\overline{\text{SCS}}$ respectively. These pins are not bonded out to the external pins and internally connected to the relevant RF Transceiver lines. PB0~PB2 are also not bonded out to the external pins and internally connected to the corresponding RF Transceiver lines named $\overline{\text{INT}}$, $\overline{\text{RST}}$ and WAKE.
PDP/CLK	I/O	—	USB D+ line. USB function is controlled by software control register. PDP pin is also pin-shared with CLK for PS2.
PDN/DATA	I/O	—	USB D- line. USB function is controlled by software control register. PDN pin is also pin-shared with DATA for PS2.
SCL	I	—	Serial Clock Input for EEPROM memory
SDA	I/O	—	Serial Data Input/Output for EEPROM memory
VDDP	—	—	Positive power supply for EEPROM memory (Note 1)
VSSP	—	—	Negative power supply for EEPROM memory, ground (Note 1)
VDD_RF2	—	—	RF transceiver power supply (Note 2)
RF_P	I/O	—	Differential RF input/output (+)
RF_N	I/O	—	Differential RF input/output (-)
VDD_RF1	—	—	RF transceiver power supply (Note 2)
GPIO0, GPIO1	I/O	—	General Purpose digital I/O It is also used as an external TX/RX switch control.
GPIO2	I/O	—	General Purpose digital I/O It is also used as an external Power Amplifier (P.A.) enable control.
VDD_D	—	—	RF transceiver digital circuit power supply (+)
GND & GND_GR & GND_PLL	—	—	GND is the RF transceiver digital circuit power supply (-). GND_GR is the RF transceiver Guard-Ring ground. GND_PLL is the RF transceiver PLL negative power supply (-), ground. These pins are internally bonded together.
VDD_2V2	O	—	RF transceiver DC-DC output voltage It can not be used.
VDD_3V	I	—	RF transceiver 3V input for the DC-DC regulator
VDD_GR & VDD_BG	—	—	VDD_GR is the RF transceiver Guard-Ring power supply (Note 2) VDD_BG is the RF transceiver Band-gap power supply (Note 2)
VDD_A	—	—	RF transceiver power supply for analog circuit (Note 2)
VDD_PLL	—	—	RF transceiver positive power supply for PLL circuit (Note 2)
VDD_CP	—	—	RF transceiver Charge pump power supply (Note 2)

Pin Name	I/O	Options	Description
VDD_VCO		—	RF transceiver Voltage-Controlled Oscillator pump power supply (Note 2)
LOOP_C	I/O	—	RF transceiver PLL loop filter external capacitor It should be connected to the external capacitor.
XTAL_P	I	—	RF transceiver 32MHz Crystal input (+)
XTAL_N	I	—	RF transceiver 32MHz Crystal input (-)
DB5	I	—	Test pin. It should be connected to ground.
$\overline{RES}$	I	—	Schmitt trigger reset input. Active low
VSS	—	—	Digital negative power supply, ground
VDD	—	—	Digital positive power supply
V33O	O	—	3.3V regulator output

Note: (1) VDDP and VSSP should be externally connected to the MCU power supply named VDD and VSS respectively.

(2) Connecting bypass capacitor(s) as close to the pin as possible.

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 150mA
Total Power Dissipation 500mW

Storage Temperature ..... –50°C to 125°C
Operating Temperature ..... –40°C to 85°C
I_{OH} Total ..... –100mA

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_{DD}	Operating Voltage (Integrated oscillator)	—	f_{SYS}=6MHz or 12MHz	3.3	—	5.5	V
I_{DD}	Operating Current	5V	No load, f_{SYS}=6MHz	—	6.5	12	mA
			No load, f_{SYS}=12MHz	—	7.5	16	mA
I_{STB}	Standby Current (WDT Enabled)	5V	NO load, system USB suspend	—	—	250	μA
	Standby Current (WDT Disabled)		NO load, system USB suspend	—	—	230	μA
	Standby Current (WDT Enabled)		No load, system HALT, input/output mode, set SUSPEND2 [1CH]	—	—	15	μA
V_{IL}	Input low voltage for I/O ports	5V	—	0	—	0.8	V
	Input Low Voltage for $\overline{RES}$ pin			0	—	$0.4V_{DD}$	V
V_{IH}	Input High Voltage for PA, PC	5V	—	2	—	5	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				
V_{LVR}	Low voltage detecting voltage	5V	—	2.0	2.6	3.2	V
V_{V33O}	3.3V Regulator Output for USB SIE	5V	I_{V33O}=70mA	3.0	3.3	3.6	V
I_{OL1}	Output sink current for PA, PB7	5V	V_{OL}=0.4V	2	—	—	mA
I_{OH1}	Output source current for PA, PB7	5V	V_{OH}=3.4V	–2	—	—	mA
I_{OL2}	Output sink current for PB0~PB6	5V	V_{OL}=0.4V	6	—	—	mA
I_{OH2}	Output source current for PB0~PB6	5V	V_{OH}=3.4V	–6	—	—	mA
R_{PH1}	Pull-high resistor for CLK, DATA	5V	—	—	4.7	—	kΩ
R_{PH2}	Pull-high resistor for PA, PB7			25	50	80	kΩ
R_{PH3}	Pull-high resistor for PB0~PB6			10	30	50	kΩ

EEPROM Memory D.C. Characteristics

Ta=–40°C~85°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DDP}	Conditions				
V_{CC}	Operating Voltage	—	—	2.2	—	5.5	V
I_{CC1}*	Operating Current	5V	Read at 100kHz	—	—	2	mA
I_{CC2}*	Operating Current	5V	Write at 100kHz	—	—	5	mA
I_{STB1}*	Standby Current	5V	V_{IN}=0 or V_{DDP}	—	—	4	μA
I_{STB2}*	Standby Current	2.4V	V_{IN}=0 or V_{DDP}	—	—	3	μA
V_{IL}	Input Low Voltage	—	—	–1	—	0.3V_{DDP}	V
V_{IH}	Input High Voltage	—	—	0.7V_{DDP}	—	V_{DDP}+0.5	V
V_{OL}	Output Low Voltage	2.4V	I_{OL}=2.1mA	—	—	0.4	V
I_{LI}	Input Leakage Current	5V	V_{IN}=0 or V_{DDP}	—	—	1	μA
I_{LO}	Output Leakage Current	5V	V_{OUT}=0 or V_{DDP}	—	—	1	μA

Note: *: The operating current I_{CC1} and I_{CC2} listed here are the additional currents consumed when the EEPROM Memory operates in Read Operation and Write Operation respectively. If the EEPROM is operating, the I_{CC1} or I_{CC2} should be added to calculate the relevant operating current of the device for different conditions. To calculate the standby current for the whole device, the standby current shown above should also be taken into account.

A.C. Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
f_{SYS}	System Clock	5V	—	—	6	—	MHz
				—	12	—	MHz
f_{RCSYS}	RC Clock with 8-bit Prescaler Register	—	—	—	32	—	kHz
$t_{configure}$	Watchdog Time-out Period	—	—	1024	—	—	$1/f_{RCSYS}$
	USBD+, USBD- Rising & Falling Time	—	—	75	—	300	ns
t_{OST}	Oscillation Start-up Timer Period	—	—	—	1024	—	$1/f_{SYS}$

Note: Power_on period = $t_{configure}$ + t_{OST}

WDT Time_out in Normal Mode = 1/ f_{RCSYS} × 256 × WDTS + $t_{configure}$

WDT Time_out in Power Down Mode = 1/ f_{RCSYS} × 256 × WDTS + t_{OST}

EEPROM Memory A.C. Characteristics

Ta=–40°C~85°C

Symbol	Parameter	Remark	Standard Mode*		V_{DDP}=5V±10%		Unit
			Min.	Max.	Min.	Max.	
f_{SK}	SCL Clock Frequency	—	—	100	—	400	kHz
t_{HIGH}	Clock High Time	—	4000	—	600	—	ns
t_{LOW}	Clock Low Time	—	4700	—	1200	—	ns
t_r	SDA and SCL Rise Time	Note	—	1000	—	300	ns
t_f	SDA and SCL Fall Time	Note	—	300	—	300	ns
$t_{HD:STA}$	START Condition Hold Time	After this period the first clock pulse is generated	4000	—	600	—	ns
$t_{SU:STA}$	START Condition Setup Time	Only relevant for repeated START condition	4000	—	600	—	ns
$t_{HD:DAT}$	Data Input Hold Time	—	0	—	0	—	ns
$t_{SU:DAT}$	Data Input Setup Time	—	200	—	100	—	ns
$t_{SU:STO}$	STOP Condition Setup Time	—	4000	—	600	—	ns
t_{AA}	Output Valid from Clock	—	—	3500	—	900	ns
t_{BUF}	Bus Free Time	Time in which the bus must be free before a new transmission can start	4700	—	1200	—	ns
t_{SP}	Input Filter Time Constant (SDA and SCL Pins)	Noise suppression time	—	100	—	50	ns
t_{WR}	Write Cycle Time	—	—	5	—	5	ms

Note: * The standard mode means V_{DDP}=2.2V to 5.5V

For relative timing, refer to the timing diagrams in EEPROM Data Memory section.

System Architecture

A key factor in the high-performance features of the Holtek range of microcontrollers is attributed to the 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 operations of the instruction set. It 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 and A/D control system with maximum reliability and flexibility.

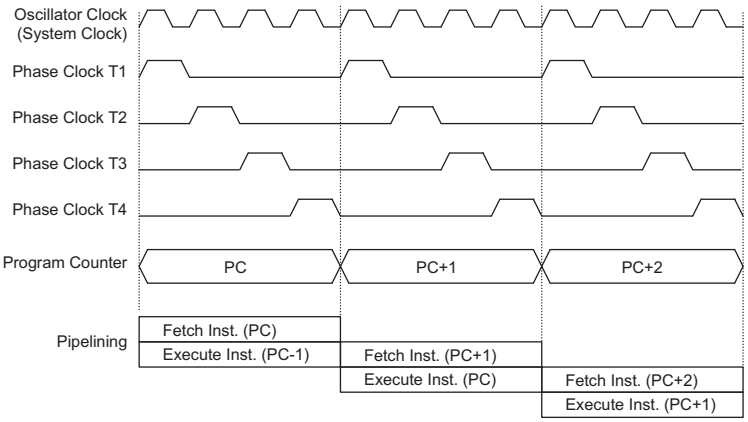
Clocking and Pipelining

The system clock is derived from an internal oscillator and 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.

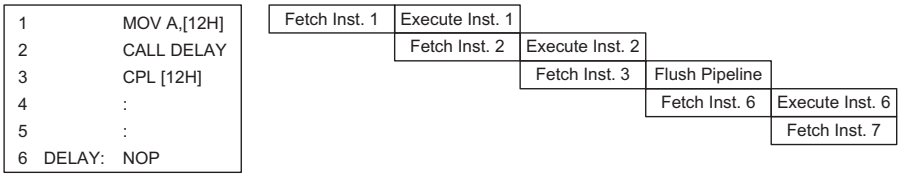
For instructions involving branches, such as jump or call instructions, two machine cycles are required to complete instruction execution. An extra cycle is required as the program takes one cycle to first obtain the actual jump or call address and then another cycle to actually execute the branch. The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

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. It must be noted that only the lower 8 bits, known as the Program Counter Low Register, are directly addressable by user.



System Clocking and Pipelining



Instruction Fetching

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.

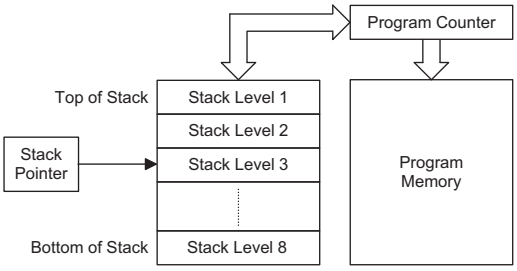
The lower byte of the Program Counter, known as the Program Counter Low register or PCL, is available for program control and is a readable and writeable register. By transferring data directly into this register, a short program jump can be executed directly, however, as only this low byte is available 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.

The lower byte of the Program Counter is fully accessible under program control. Manipulating the PCL might cause program branching, so an extra cycle is needed to pre-fetch. Further information on the PCL register can be found in the Special Function Register section.

Stack

This is a special part of the memory which is used to save the contents of the Program Counter only. The stack has 8 levels 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, SP, 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.



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

Mode	Program Counter Bits											
	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Initial Reset	0	0	0	0	0	0	0	0	0	0	0	0
USB Interrupt	0	0	0	0	0	0	0	0	0	1	0	0
Timer/Event Counter 0 Overflow	0	0	0	0	0	0	0	0	1	0	0	0
Timer/Event Counter 1 Overflow	0	0	0	0	0	0	0	0	1	1	0	0
SPI Interrupt	0	0	0	0	0	0	0	1	0	0	0	0
Skip	Program Counter + 2											
Loading PCL	PC11	PC10	PC9	PC8	@7	@6	@5	@4	@3	@2	@1	@0
Jump, Call Branch	#11	#10	#9	#8	#7	#6	#5	#4	#3	#2	#1	#0
Return from Subroutine	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0

Program Counter

Note: PC11~PC8: Current Program Counter bits
#11~#0: Instruction code address bits
@7~@0: PCL bits
S11~S0: Stack register bits

- 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

Program Memory

The Program Memory is the location where the user code or program is stored. The HT82D40REW is a One-Time Programmable, OTP, memory type device where users can program their application code into the device. By using the appropriate programming tools, OTP devices offer users the flexibility to freely develop their applications which may be useful during debug or for products requiring frequent upgrades or program changes. OTP devices are also applicable for use in applications that require low or medium volume production runs.

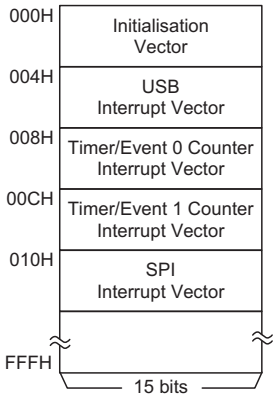
Structure

The Program Memory has a capacity of 4K by 15 bits. The Program Memory is addressed by the Program Counter and also contains data, table information and interrupt entries. Table data, which can be setup in any location within the Program Memory, is addressed by separate table pointer registers.

Special Vectors

Within the Program Memory, certain locations are reserved for special usage such as reset and interrupts.

- Location 000H
  This area is reserved for program initialization. After chip reset, the program always begins execution at location 000H.
- Location 004H
  This area is reserved for the USB interrupt service program. If the USB interrupt is activated, the interrupt is enabled and the stack is not full, the program jumps to this location and begins execution.
- Location 008H
  This area is reserved for the Timer/Event Counter 0 interrupt service program. If a timer interrupt results from a Timer/Event Counter 0 overflow, and if the interrupt is enabled and the stack is not full, the program jumps to this location and begins execution.



Program Memory Structure

- Location 00CH
  This area is reserved for the Timer/Event Counter 1 interrupt service program. If a timer interrupt results from a Timer/Event Counter 1 overflow, and the interrupt is enabled and the stack is not full, the program jumps to this location and begins execution.
- Location 010H
  This vector is used by serial interface. When 8-bits of data have been received or transmitted successfully from serial interface. The program will jump to this location and begin execution if the interrupt is enabled and the stack is not full.
- Table location
  Any location in the program memory can be used as look-up tables. There are three methods to read the Program Memory data using two table read instructions: "TABRDC" and "TABRDL", transfer the contents of the lower-order byte to the specified data memory, and the higher-order byte to TBLH (08H).
  The three methods are shown as follows:
  - Using the instruction "TABRDC [m]" for the current Program Memory page, where one page= 256words, where the table location is defined by TBLP in the current page. This is where the configuration option has disabled the TBHP register.
  - Using the instruction "TABRDC [m]", where the table location is defined by registers TBLP and TBHP. Here the configuration option has enabled the TBHP register.
  - Using the instruction "TABRDL [m]", where the table location is defined by registers TBLP in the last page which has the address range 0F00H~0FFFH.

Instruction	Table Location Bits											
	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TABRDC [m]	PC11	PC10	PC9	PC8	@7	@6	@5	@4	@3	@2	@1	@0
TABRDL [m]	1	1	1	1	@7	@6	@5	@4	@3	@2	@1	@0

Table Location

Note: PC11~PC8: Current Program Counter bits when TBHP register is disabled.
TBHP register Bit 3 ~ Bit 0 when TBHP is enabled.
@7~@0: Table Pointer TBLP bits

Only the destination of the lower-order byte in the table is well-defined, the other bits of the table word are transferred to the lower portion of TBLH, and the remaining 1-bit words are read as ″0″. The Table Higher-order byte register (TBLH) is read only. The table pointers, TBLP and TBHP, are read/write registers, which indicate the table location. Before accessing the the table, the locations must be placed in the TBLP and TBHP registers (if the configuration option has disabled TBHP then the value in TBHP has no effect). TBLH is read only and cannot be restored. If the main routine and the ISR (Interrupt Service Routine) both employ the table read instruction, the contents of the TBLH in the main routine are likely to be changed by the table read instruction used in the ISR and errors can occur. Using the table read instruction in the main routine and the ISR simultaneously should be avoided. However, if the table read instruction has to be applied in both the main routine and the ISR, the interrupt should be disabled prior to the table read instruction. It will not be enabled until the TBLH has been backed up. All table related instructions require two cycles to complete the operation. These areas may function as normal program memory depending on the requirements.

Once TBHP is enabled, the instruction ″TABRDC [m]″ reads the Program Memory data as defined by the TBLP and TBHP values. If the Program Memory code option has disabled TBHP, the instruction ″TABRDC [m]″ reads the Program Memory data as defined by TBLP only in the current Program Memory page.

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 lower order address of the look up data to be retrieved in the TBLP register and the higher order address in the TBHP register. These two registers define the full address of the look-up table. Using the TBHP must be selected by configuration option, if not used table data can still be accessed but only the lower byte address in the current page or last page can be defined.

After setting up the table pointers, the table data can be retrieved from the current Program Memory page or last Program Memory page using the ″TABRDC[m]″ or ″TABRDL [m]″ instructions, respectively. When these instructions are 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″.

Table Program Example

The following example shows how the table pointer and table data is defined and retrieved from the microcontroller. 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 ″F00H″ which refers to the start address of the last page within the 4K Program Memory of 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 ″F06H″ 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 ″TABRDL [m]″ instruction is executed.

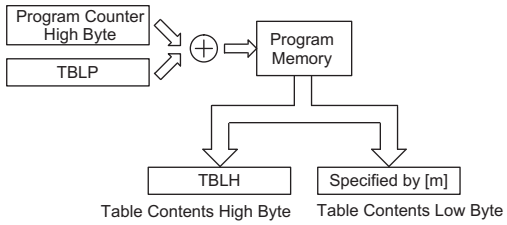


Table Read – TBLP only

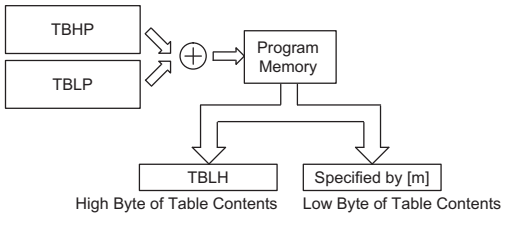


Table Read – TBLP/TBHP

Table High Byte Pointer for Current Table Read TBHP (Address 0X1F)

Register	Bits	Read/Write	Functions
TBHP (1FH)	3~0	R/W	Store current table location bit11~bit8 value

```
tempreg1    db      ?   ; temporary register #1
tempreg2    db      ?   ; temporary register #2
            :
            :

mov         a,06h       ; initialise table pointer - note that this address is referenced

mov         tblp,a      ; to the last page or present page
            :
            :

tabrdl      tempreg1    ; transfers value in table referenced by table pointer to tempregl
                        ; data at prog. memory address "F06H" transferred 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 prog.memory address "F05H" 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         F00h        ; sets initial address of last page

dc          00Ah, 00Bh, 00Ch, 00Dh, 00Eh, 00Fh, 01Ah, 01Bh
            :
            :
```

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 the table read instructions. If using the table read instructions, the Interrupt Service Routines may change the value of 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.

Data Memory

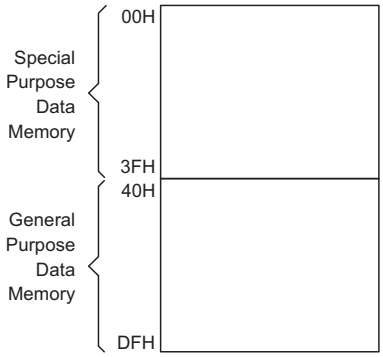
The Data Memory is a volatile area of 8-bit wide RAM internal memory and is the location where temporary information is stored. Divided into two sections, the first of these is an area of RAM where special function registers are located. These registers have fixed locations and are necessary for correct 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.

Structure

The two sections of Data Memory, the Special Purpose and General Purpose Data Memory are located at consecutive locations. All are implemented in RAM and are 8 bits wide. 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.

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 program for both read and write 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.



Data Memory Structure

Note: Most of the Data Memory bits can be directly manipulated using the "SET [m].i" and "CLR [m].i" with the exception of a few dedicated bits. The Data Memory can also be accessed through the memory pointer register MP.

Special Purpose Data Memory

This area of Data Memory is where registers, necessary for the correct operation of the microcontroller, are stored. It is divided into two banks, Bank 0 and Bank1. 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″.

The Special Purpose Registers for the USB interface are stored in Bank 1 which can only be accessed by first setting the Bank Pointer to a value of 01H and then using Indirect Addressing Register IAR1 and Memory Pointer MP1. Bank 1 can only be accessed indirectly using the MP1 Memory Pointer, direct addressing is not possible.

	Bank0		Bank1
00H	IAR0	40H	USB_STAT
01H	MP0	41H	PIPE_CTRL
02H	IAR1	42H	AWR
03H	MP1	43H	STALL
04H	BP	44H	PIPE
05H	ACC	45H	SIES
06H	PCL	46H	MISC
07H	TBLP	47H	ENDPT_EN
08H	TBLH	48H	FIFO0
09H	WDTS	49H	FIFO1
0AH	STATUS	4AH	FIFO2
0BH	INTC		
0CH			
0DH	TMR0		
0EH	TMR0C		
0FH	TMR1H		
10H	TMR1L		
11H	TMR1C		
12H	PA		
13H	PAC		
14H	PB		
15H	PBC		
16H			
17H			
18H			
19H			
1AH			
1BH			
1CH	SPIR		
1DH			
1EH	INTC1		
1FH	TBHP		
20H	USC		
21H	USR		
22H	SCC		
23H	SBCR		
24H	SBDR		
25H ... 3FH			

: Unused read as "0"

Special Purpose Data Memory

Special Function Registers

To ensure successful operation of the microcontroller, certain internal registers are implemented in the Data Memory area. These registers ensure correct operation of internal functions such as timers, interrupts, etc., as well as external functions such as I/O data control. The location of these registers within the Data Memory begins at the address 00H. Any unused Data Memory locations between these special function registers and the point where the General Purpose Memory begins is reserved and attempting to read data from these locations will return a value of 00H.

Indirect Addressing Register – 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 Pointer, MP0 or MP1. Acting as a pair, IAR0 and MP0 can together only access data from Bank 0, while the IAR1 and MP1 register pair can access data from both Bank 0 and Bank 1. 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 Pointer – MP0, MP1

For all devices, 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 can only access data in Bank 0 while MP1 can access both banks.

```
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 Data Memory addresses.

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, TBLH, TBHP

These two 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 pointers and indicate the location where the table data is located. Their value must be setup before any table read commands are executed. Their values 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.

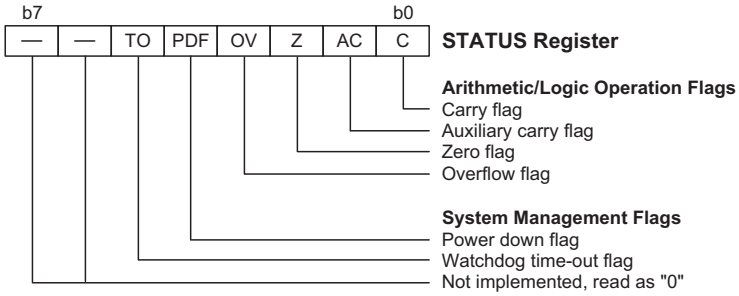
Watchdog Timer Register – WDTS

The Watchdog feature of the microcontroller provides an automatic reset function giving the microcontroller a means of protection against spurious jumps to incorrect Program Memory addresses. To implement this, a timer is provided within the microcontroller which will issue a reset command when its value overflows. To provide variable Watchdog Timer reset times, the Watchdog Timer clock source can be divided by various division ratios, the value of which is set using the WDTS register. By writing directly to this register, the appropriate division ratio for the Watchdog Timer clock source can be setup. Note that only the lower 3 bits are used to set division ratios between 1 and 128.

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.



Status Register

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 interrupt routine can change the status register, precautions must be taken to correctly save it.

Interrupt Control Registers – INTC, INTC1

The microcontrollers provide two internal timer/event counter overflow interrupts, one USB interrupt and one SPI interrupt. By setting various bits within this register using standard bit manipulation instructions, the enable/disable function of each interrupt can be independently controlled. A master interrupt bit within this register, the EMI bit, acts like a global enable/disable and is used to set all of the interrupt enable bits on or off. This bit is cleared when an interrupt routine is entered to disable further interrupt and is set by executing the ″RETI″ instruction.

Timer/Event Counter Registers – TMR0, TMR0C, TMR1H, TMR1L, TMR1C

Both devices possess a single internal 8-bit count-up timer. An associated register known as TMR0 is the location where the timers 8-bit value is located. This register can also be preloaded with fixed data to allow different time intervals to be setup. An associated control register, known as TMR0C, contains the setup information for this timer, which determines in what mode the timer is to be used as well as containing the timer on/off control function.

All devices possess one internal 16-bit count-up timer. An associated register pair known as TMR1L/TMR1H is the location where the timer 16-bit value is located. This register can also be preloaded with fixed data to allow different time intervals to be setup. An associated control register, known as TMR1C, contains the setup information for this timer, which determines in what mode the timer is to be used as well as containing the timer on/off control function.

Input/Output Ports and Control Registers

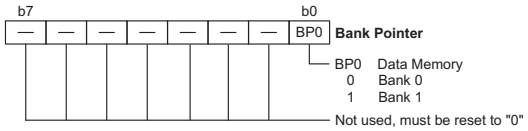
Within the area of Special Function Registers, the I/O registers and their associated control registers play a prominent role. All I/O ports have a designated register correspondingly labeled as PA and PB. These labeled I/O registers are mapped to specific addresses within the Data Memory as shown in the Data Memory table, which are used to transfer the appropriate output or input data on that port. With each I/O port there is an associated control register labeled PAC and PBC also mapped to specific addresses with the Data Memory.

The control register specifies which pins of that port are set as inputs and which are set as outputs. To setup a pin as an input, the corresponding bit of the control register must be set high, for an output it must be set low. During program initialisation, it is important to first setup the control registers to specify which pins are outputs and which are inputs before reading data from or writing data to the I/O ports. One flexible feature of these registers is the ability to directly program single bits using the ″SET [m].i″ and ″CLR [m].i″ instructions. The ability to

change I/O pins from output to input and vice versa by manipulating specific bits of the I/O control registers during normal program operation is a useful feature of these devices.

Bank Pointer – BP

The Special Purpose Data Memory is divided into two Banks, Bank 0 and Bank 1. The USB control registers are located in Bank 1, while all other registers are located in Bank 0. The Bank Pointer selects which bank data is to be accessed from. If Bank 0 is to be accessed, then BP must be set to a value of 00H, while if Bank 1 is to be accessed, then BP must be set to a value of 01H.



Bank Pointer

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 and wake-up options for all ports, the user is provided with an I/O structure to meet the needs of a wide range of application possibilities.

The microcontroller provides up to 9 bidirectional input/output lines labeled with port names PA and PB7.

These I/O ports are mapped to the Data Memory with addresses as shown in the Special Purpose Data Memory table. 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.

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, I/O pins, when configured as an input have the capability of being connected to an internal pull-high resistor. The pull-high resistors are selectable via configuration options and are implemented using weak PMOS transistors. A configuration option on each I/O port pin can be selected to select pull-high resistor.

Port A CMOS/NMOS/PMOS Structure

The pins on Port A can be setup via configuration option to be either CMOS, NMOS or PMOS types.

Port Pin Wake-up

If the HALT instruction is executed, the device will enter the Power Down Mode, where the system clock will stop resulting in power being conserved, 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 pins from high to low. After a HALT instruction forces the microcontroller into entering the Power Down Mode, the processor will remain in a low-power state until the logic condition of the selected wake-up pin on the port pin changes from high to low. This function is especially suitable for applications that can be woken up via external switches. Each pin on Port A and PB7 has the capability to wake-up the device on an external falling edge. Note that some pins can only be setup nibble wide whereas other can be bit selected to have a wake-up function.

I/O Port Control Registers

Each I/O port has its own control register named PAC and PBC to control the input/output configuration. With this control register, each CMOS output or input with or without pull-high resistor structures can be reconfigured dynamically under software control. Each of the I/O ports is directly mapped to a bit in its associated port control register. Note that several pins can be setup to have NMOS or PMOS outputs using configuration options.

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 an 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 some pins, the chosen function of the multi-function I/O pins is set by configuration options while for others the function is set by application program control.

- External Timer0 Clock Input
  The external timer pin TMR0 is pin-shared with the I/O pin PA6. To configure this pin to operate as timer input, the corresponding control bits in the timer control register must be correctly set. For applications that do not require an external timer input, this pin can be used as a normal I/O pin. Note that if used as a normal I/O pin

the timer mode control bits in the timer control register must select the timer mode, which has an internal clock source, to prevent the input pin from interfering with the timer operation.

- SPI interface pins
  The SPI interface pins known as SDI, SDO, SCK and SCS are pin-shared with the I/O lines PB3~PB6 respectively. To configure these pins to operate as input or output, the corresponding control bit in the SPI control register must be correctly set. However, these pins are not bonded out to external pins and used as the master SPI pins to be internally connect to the RF Transceiver slave SPI interface to control the overall RF Transceiver functions.
- External Timer1 Clock Input
  The external timer pin TMR1 is pin-shared with the I/O pin PA7. To configure this pin to operate as timer input, the corresponding control bits in the timer control register must be correctly set. For applications that do not require an external timer input, this pin can be used as a normal I/O pin. Note that if used as a normal I/O pin the timer mode control bits in the timer control register must select the timer mode, which has an internal clock source, to prevent the input pin from interfering with the timer operation

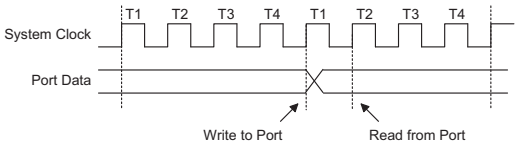
I/O Pin Structures

The diagram illustrates a generic I/O pin internal structures. As the exact logical construction of the I/O pin will differ and as the pin-shared structures are not illustrated this diagram is supplied as a guide only to assist with the functional understanding of the I/O pins.

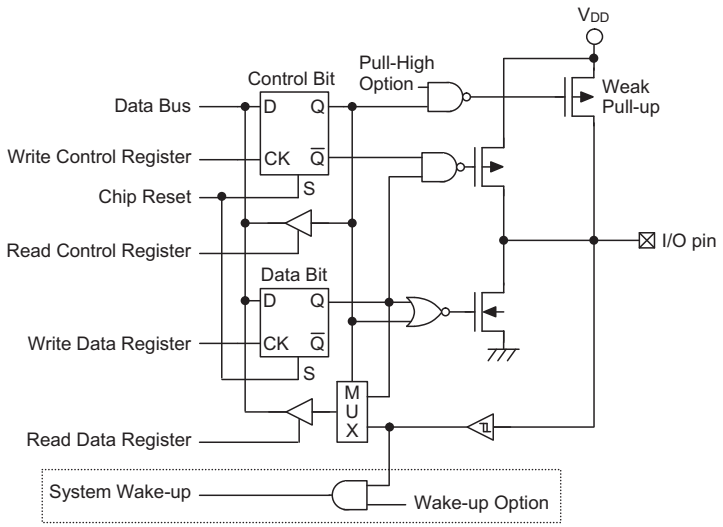
Programming Considerations

Within the user program, one of the first things to consider is port initialisation. After a reset, all of the data and port control register will be set high. This means that all I/O pins will default to an input state, the level of which depends on the other connected circuitry and whether pull-high options have been selected. If the PAC and PBC 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 PA and PB port data registers are first programmed. Selecting which pins are inputs and which are outputs can be achieved byte-wide by loading the correct value into the 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.

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



Read/Write Timing



Input/Output Ports

Timer/Event Counters

The provision of timers form an important part of any microcontroller, giving the designer a means of carrying out time related functions. This device contains two count-up timers of 8-bit and 16-bit capacities respectively. As each timer has three different operating modes, they can be configured to operate as a general timer, an external event counter or as a pulse width measurement device.

There are two types of registers related to the Timer/Event Counters. The first is the register that contains the actual value of the Timer/Event Counter and into which an initial value can be preloaded, and is known as TMR0, TMR1H or TMR1L. Reading from this register retrieves the contents of the Timer/Event Counter. The second type of associated register is the Timer Control Register, which defines the timer options and determines how the Timer/Event Counter is to be used, and has the name TMR0C or TMR1C. This device can have the timer clocks configured to come from the internal clock sources. In addition, the timer clock source can also be configured to come from the external timer pins.

The external clock source is used when the Timer/Event Counter is in the event counting mode, the clock source being provided on the external timer pin. The pin has the name TMR0 or TMR1 and is pin-shared with an I/O pin. Depending upon the condition of the T0E or T1E bit in the Timer Control Register, each high to low, or low to high transition on the external timer input pin will increment the Timer/Event Counter by one.

Configuring the Timer/Event Counter Input Clock Source

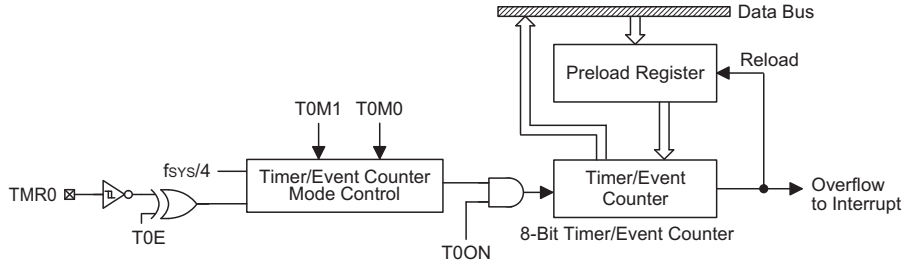
The Timer/Event Counter's clock can originate from various sources. The instruction clock source (system clock source divided by 4) is used when the Timer/Event Counter 0 or Timer/Event Counter 1 is in the timer mode or in the pulse width measurement mode. The external clock source is used when the Timer/Event Counter is in the event counting mode, the clock source being provided on the external timer pin, TMR0 or TMR1. Depending upon the condition of the T0E or T1E bit, each high to low, or low to high transition on the external timer pin will increment the counter by one.

Timer Register – TMR0, TMR1L/TMR1H

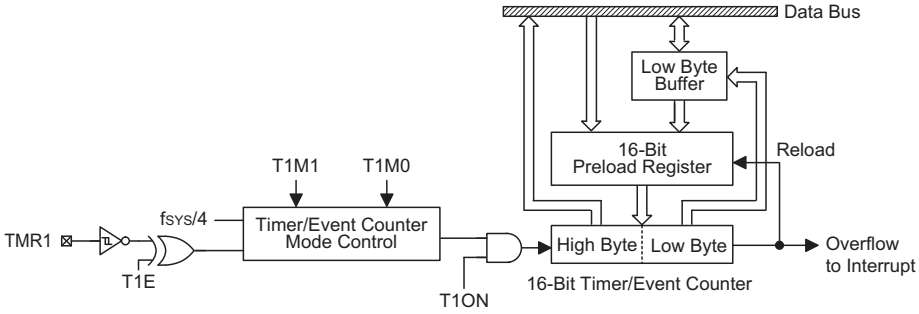
The timer registers are special function registers located in the Special Purpose RAM Data Memory and are the places where the actual timer values are stored. For 8-bit Timer/Event Counter 0, this register is known as TMR0. For 16-bit Timer/Event Counter 1, the timer registers are known as TMR1L and TMR1H. The value in the timer registers increases by one each time an internal clock pulse is received or an external transition occurs on the external timer pin. The timer will count from the initial value loaded by the preload register to the full count of FFH for the 8-bit timer or FFFFH for the 16-bit timer at which point the timer overflows and an internal interrupt signal is generated. The timer value will then be reset with the initial preload register value and continue counting.

To achieve a maximum full range count of FFH for the 8-bit timer or FFFFH for the 16-bit timer, the preload registers must first be cleared to all zeros. It should be noted that after power-on, the preload register will be in an unknown condition. Note that if the Timer/Event Counter is switched off and data is written to its preload registers, this data will be immediately written into the actual timer registers. However, if the Timer/Event Counter is enabled and counting, any new data written into the preload data registers during this period will remain in the preload registers and will only be written into the timer registers the next time an overflow occurs.

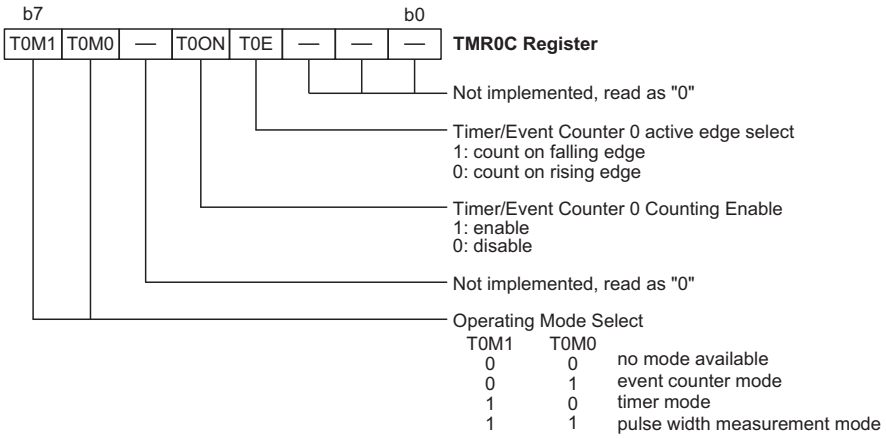
For the 16-bit Timer/Event Counter which has both low byte and high byte timer registers, accessing these registers is carried out in a specific way. It must be note when using instructions to preload data into the low byte timer register, namely TMR1L, the data will only be placed in a low byte buffer and not directly into the low byte timer register. The actual transfer of the data into the low byte timer register is only carried out when a write to its associated high byte timer register, namely TMR1H, is executed. On the other hand, using instructions to preload data into the high byte timer register will result in the data being directly written to the high byte timer register. At the same time the data in the low byte buffer will be transferred into its associated low byte timer register. For this reason, the low byte timer register should be written first when preloading data into the 16-bit timer registers. It must also be noted that to read the contents of the low byte timer register, a read to the high byte timer register must be executed first to latch the contents of the low byte timer register into its associated low byte buffer. After this has been done, the low byte timer register can be read in the normal way. Note that reading the low byte timer register will result in reading the previously latched contents of the low byte buffer and not the actual contents of the low byte timer register.



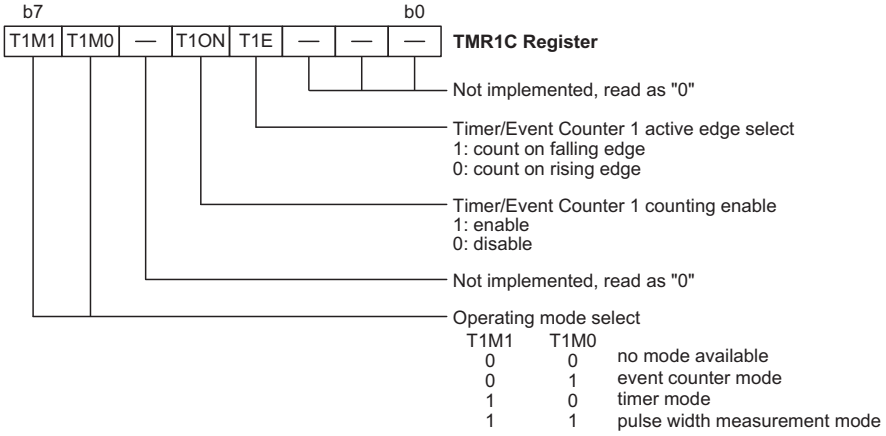
8-bit Timer/Event Counter 0 Structure



16-bit Timer/Event Counter 1 Structure



Timer/Event Counter 0 Control Register



Timer/Event Counter 1 Control Register

Timer Control Register – TMR0C/TMR1C

The flexible features of the Holtek microcontroller Timer/Event Counters enable them to operate in three different modes, the options of which are determined by the contents of their respective control register. For devices are two timer control registers known as TMR0C, TMR1C . It is the timer control register together with its corresponding timer registers that control the full operation of the Timer/Event Counters. Before the timers can be used, it is essential that the appropriate timer control register is fully programmed with the right data to ensure its correct operation, a process that is normally carried out during program initialization.

To choose which of the three modes the timer is to operate in, either in the timer mode, the event counting mode or the pulse width measurement mode, bits 7 and 6 of the Timer Control Register, which are known as the bit pair T0M1/T0M0 or T1M1/T1M0 respectively, depending upon which timer is used, must be set to the required logic levels. The timer-on bit, which is bit 4 of the Timer Control Register and known as T0ON or T1ON, depending upon which timer is used, provides the basic on/off control of the respective timer. Setting the bit high allows the counter to run, clearing the bit stops the counter. If the timer is in the event count or pulse width measurement mode, the active transition edge level type is selected by the logic level of bit 3 of the Timer Control Register which is known as T0E or T1E, depending upon which timer is used.

Configuring the Timer Mode

In this mode, the Timer/Event Counter can be utilised to measure fixed time intervals, providing an internal interrupt signal each time the Timer/Event Counter overflows. To operate in this mode, the Operating Mode Select bit pair, T0M1/T0M0 or T1M1/T1M0, in the Timer Control Register must be set to the correct value as shown.

Control Register Operating Mode Select Bits for the Timer Mode	Bit7	Bit6
	1	0

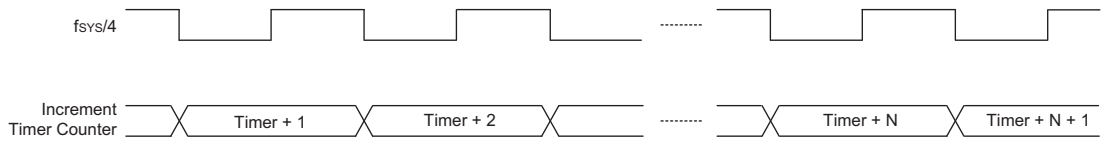
In this mode the internal clock, $f_{SYS}/4$ is used as the internal clock for the Timer/Event Counters. After the other bits in the Timer Control Register have been setup, the enable bit T0ON or T1ON, which is bit 4 of the Timer Control Register, can be set high to enable the Timer/Event Counter to run.Each time an internal clock cycle occurs, the Timer/Event Counter increments by one. When it is full and overflows, an interrupt signal is generated and the Timer/Event Counter will reload the value already loaded into the preload register and continue counting. The interrupt can be disabled by ensuring that the Timer/Event Counter Interrupt Enable bit in the Interrupt Control Register, INTC, is reset to zero.

Configuring the Event Counter Mode

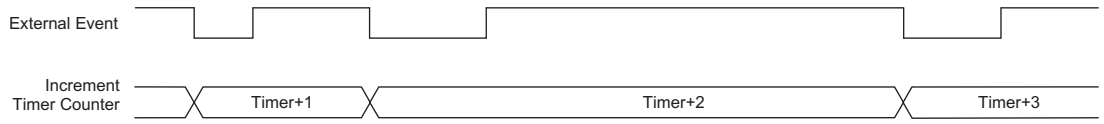
In this mode, a number of externally changing logic events, occurring on the external timer pin, can be recorded by the Timer/Event Counter. To operate in this mode, the Operating Mode Select bit pair, T0M1/T0M0 or T1M1/T1M0, in the Timer Control Register must be set to the correct value as shown.

Control Register Operating Mode Select Bits for the Event Counter Mode	Bit7	Bit6
	0	1

In this mode, the external timer pin, TMR0 or TMR1, is used as the Timer/Event Counter clock source, however it is not divided by the internal prescaler. After the other bits in the Timer Control Register have been setup, the enable bit T0ON or T1ON, which is bit 4 of the Timer Control Register, can be set high to enable the Timer/Event Counter to run. If the Active Edge Select bit T0E or T1E, which is bit 3 of the Timer Control Register, is low, the Timer/Event Counter will increment each time the external timer pin receives a low to high transition. If the Active Edge Select bit is high, the counter will increment each time the external timer pin receives a high to low transition. When it is full and overflows, an interrupt



Timer Mode Timing Chart



Event Counter Mode Timing Chart

signal is generated and the Timer/Event Counter will reload the value already loaded into the preload register and continue counting. The interrupt can be disabled by ensuring that the Timer/Event Counter Interrupt Enable bit in the Interrupt Control Register, INTC, is reset to zero.

As the external timer pin is shared with an I/O pin, to ensure that the pin is configured to operate as an event counter input pin, two things have to happen. The first is to ensure that the Operating Mode Select bits in the Timer Control Register place the Timer/Event Counter in the Event Counting Mode, the second is to ensure that the port control register configures the pin as an input. It should be noted that in the event counting mode, even if the microcontroller is in the Power Down Mode, the Timer/Event Counter will continue to record externally changing logic events on the timer input pin. As a result when the timer overflows it will generate a timer interrupt and corresponding wake-up source.

Configuring the Pulse Width Measurement Mode

In this mode, the Timer/Event Counter can be utilised to measure the width of external pulses applied to the external timer pin. To operate in this mode, the Operating Mode Select bit pair, T0M1/T0M0 or T1M1/T1M0, in the Timer Control Register must be set to the correct valueas shown.

Control Register Operating Mode Select Bits for the Pulse Width Measurement Mode

Bit7	Bit6
1	1

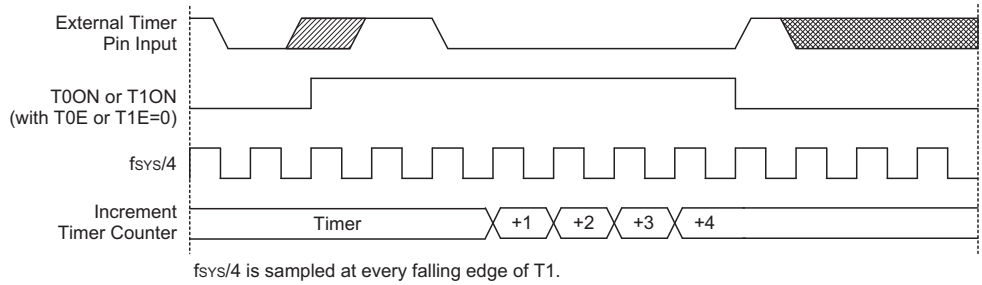
In this mode the internal clock, $f_{SYS}/4$ is used as the internal clock for the Timer/Event Counters. After the other bits in the Timer Control Register have been setup, the enable bit T0ON or T1ON, which is bit 4 of the Timer Control Register, can be set high to enable the Timer/Event Counter, however it will not actually start counting until an active edge is received on the external timer pin.

If the Active Edge Select bit T0E or T1E, which is bit 3 of the Timer Control Register, is low, once a high to low transition has been received on the external timer pin, TMR0 or TMR1, the Timer/Event Counter will start counting until the external timer pin returns to its original high level. At this point the enable bit will be automatically reset to zero and the Timer/Event Counter will stop counting. If the Active Edge Select bit is high, the Timer/Event Counter will begin counting once a low to high transition has been received on the external timer pin and stop counting when the external timer pin returns to its original low level. As before, the enable bit will be automatically reset to zero and the Timer/Event Counter will stop counting. It is important to note that in the Pulse Width Measurement Mode, the enable bit is automatically reset to zero when the external control signal on the external timer pin returns to its original level, whereas in the other two modes the enable bit can only be reset to zero under program control.

The residual value in the Timer/Event Counter, which can now be read by the program, therefore represents the length of the pulse received on the external timer pin. As the enable bit has now been reset, any further transitions on the external timer pin will be ignored. Not until the enable bit is again set high by the program can the timer begin further pulse width measurements. In this way, single shot pulse measurements can be easily made.

It should be noted that in this mode the Timer/Event Counter is controlled by logical transitions on the external timer pin and not by the logic level. When the Timer/Event Counter is full and overflows, an interrupt signal is generated and the Timer/Event Counter will reload the value already loaded into the preload register and continue counting. The interrupt can be disabled by ensuring that the Timer/Event Counter Interrupt Enable bit in the Interrupt Control Register, INTC, is reset to zero.

As the external timer pin is shared with an I/O pin, to ensure that the pin is configured to operate as a pulse width measurement pin, two things have to happen. The first is to ensure that the Operating Mode Select bits in the Timer Control Register place the Timer/Event Counter in the Pulse Width Measurement Mode and the second is to ensure that the port control register configures the pin as an input.



fSYS/4 is sampled at every falling edge of T1.

Pulse Width Measure Mode Timing Chart

I/O Interfacing

The Timer/Event Counter, when configured to run in the event counter or pulse width measurement mode, require the use of the external TMR0 and TMR1 pins for correct operation. As these pins are shared pins they must be configured correctly to ensure they are setup for use as Timer/Event Counter inputs and not as a normal I/O pins. This is implemented by ensuring that the mode select bits in the Timer/Event Counter control register, select either the event counter or pulse width measurement mode. Additionally the Port Control Register bits for these pins must be set high to ensure that the pin is setup as an input. Any pull-high resistor configuration option on these pins will remain valid even if the pin is used as a Timer/Event Counter input.

Programming Considerations

When configured to run in the timer mode, the internal instruction clock is used as the timer clock source and is therefore synchronised with the overall operation of the microcontroller. In this mode when the appropriate timer counter is full, the microcontroller will generate an internal interrupt signal directing the program flow to the respective internal interrupt vector. For the pulse width measurement mode, the internal instruction clock is also used as the timer clock source but the timer will only run when the correct logic condition appears on the external timer input pin. As this is an external event and not synchronised with the internal timer clock, the microcontroller will only see this external event when the next timer clock pulse arrives. As a result, there may be small differences in measured values requiring programmers to take this into account during programming. The same applies if the timer is configured to be in the event counting mode, which again is an external event and not synchronised with the internal system or timer clock.

When the Timer/Event Counter is read, or if data is written to the preload register, the clock is inhibited to avoid errors, however as this may result in a counting error, this should be taken into account by the programmer. Care must be taken to ensure that the timers are properly initialised before using them for the first time. The associated timer interrupt enable bits in the interrupt control register must be properly set otherwise the internal interrupt associated with the timer will remain inactive. The edge select, timer mode and clock source control bits in timer control register must also be correctly set to ensure the timer is properly configured for the required application. It is also important to ensure that an initial value is first loaded into the timer registers before the timer is switched on; this is because after power-on the initial values of the timer registers are unknown. After the timer has been initialised the timer can be turned on and off by controlling the enable bit in the timer control register. Note that setting the timer enable bit high to turn the timer on, should only be executed after the timer mode bits have been properly setup. Setting the timer enable bit high together with a mode bit modification, may lead to improper timer operation if executed as a single timer control register byte write instruction.

When the Timer/Event counter overflows, its corresponding interrupt request flag in the interrupt control register will be set. If the timer interrupt is enabled this will in turn generate an interrupt signal. However irrespective of whether the interrupts are enabled or not, a Timer/Event counter overflow will also generate a wake-up signal if the device is in a Power-down condition. This situation may occur if the Timer/Event Counter is in the Event Counting Mode and if the external signal continues to change state. In such a case, the Timer/Event Counter will continue to count these external events and if an overflow occurs the device will be woken up from its Power-down condition. To prevent such a wake-up signal from occurring, the timer interrupt request flag should first be set high before issuing the ″HALT″ instruction to enter the Power Down Mode.

Timer Program Example

This program example shows how the Timer/Event Counter registers are setup, along with how the interrupts are enabled and managed. Note how the Timer/Event Counter is turned on, by setting bit 4 of the Timer Control Register. The Timer/Event Counter can be turned off in a similar way by clearing the same bit. This example program sets the Timer/Event Counter to be in the timer mode, which uses the internal system clock as the clock source.

```
org 04h              ; USB interrupt vector
reti
org 08h              ; Timer/Event Counter interrupt vector
jmp tmr0int          ; jump here when Timer0 overflows
:
org 20h              ; main program
;internal Timer/Event Counter 0 interrupt routine
Tmr0int:
:
; Timer/Event Counter 0 main program placed here
:
reti
:
:
begin:
;setup Timer registers
mov a,09bh           ; setup Timer preload value
mov tmr0,a;
mov a,080h           ; setup Timer control register
mov tmr0c,a          ; timer mode
; setup interrupt register
mov a,005h           ; enable master interrupt and timer interrupt
mov intc,a
set tmr0c.4          ; start Timer/Event Counter - note mode bits must be previously setup
```

Interrupts

Interrupts are an important part of any microcontroller system. When an internal function such as a Timer/Event Counter overflow or a USB interrupt occur, 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 interrupts generated by the Timer/Event Counters overflow, USB interrupt and SPI interrupt.

Interrupt Register – INTC, INTC1

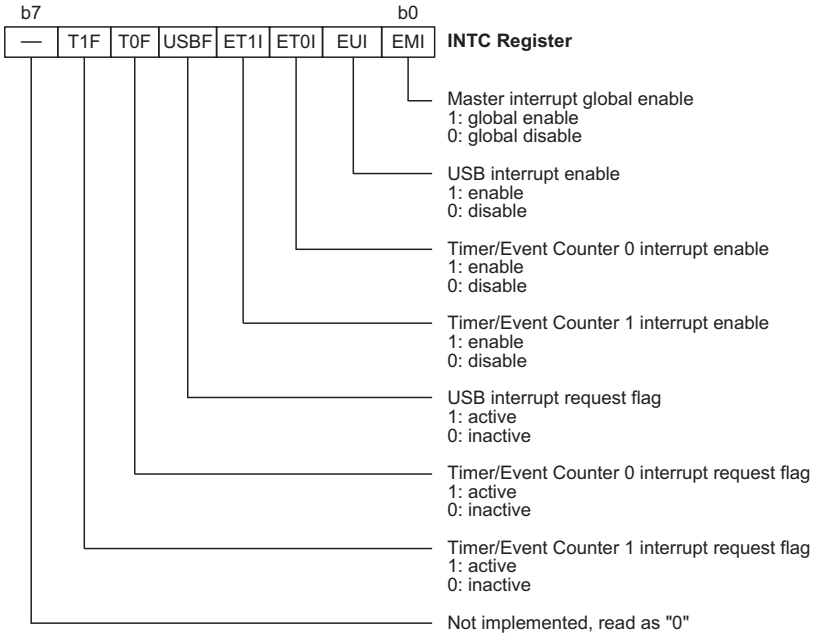
Overall interrupt control, which means interrupt enabling and request flag setting, is controlled by the interrupt control registers named INTC and INTC1. By controlling the appropriate enable bits in these registers each individual interrupt can be enabled or disabled. Also when an interrupt occurs, the corresponding request flag will be set by the microcontroller. The global enable flag if cleared to zero will disable all interrupts.

Interrupt Operation

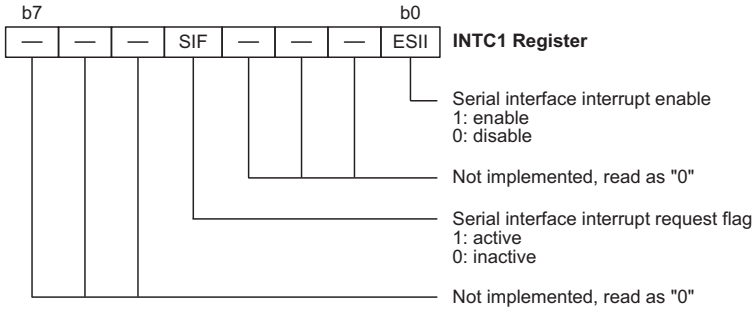
When a USB or SPI interrupt occurs or one of the Timer/Event Counters overflow, if their appropriate interrupt enable bit is set, the Program Counter, which stores the address of the next instruction to be executed, will be transferred onto the stack. The Program Counter will then be loaded with a new 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 statement 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 statement, 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 diagram with their order of priority.

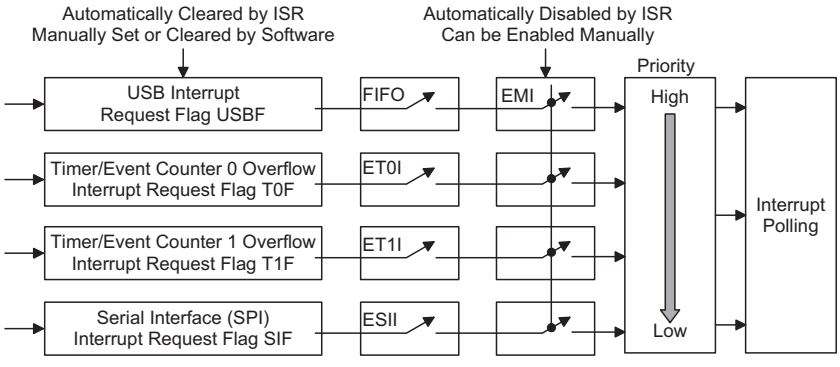
Once an interrupt subroutine is serviced, all the other interrupts will be blocked, as the EMI bit will be cleared automatically. This will prevent any further interrupt nesting from occurring. However, if other interrupt requests occur during this interval, although the interrupt will not be immediately 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.



INTC Register



INTC1 Register



Interrupt Structure

Interrupt Priority

Interrupts, occurring in the interval between the rising edges of two consecutive T2 pulses, will be serviced on the latter of the two T2 pulses, if the corresponding interrupts are enabled. In case of simultaneous requests, the following table shows the priority that is applied. These can be masked by resetting the EMI bit.

Interrupt Source	Priority	Vector
USB Interrupt	1	004H
Timer/Event Counter 0 Overflow Interrupt	2	008H
Timer/Event Counter 1 Overflow Interrupt	3	00CH
Serial Interface (SPI) Interrupt	4	010H

In cases where the USB interrupt, Timer/Event Counters overflow interrupts and the SPI interrupt are enabled and where these interrupts occur simultaneously, the USB interrupt will always have priority and will therefore be serviced first. Suitable masking of the individual interrupts using the interrupt registers can prevent simultaneous occurrences.

Timer/Event Counter Interrupt

For a Timer/Event Counter interrupt to occur, the global interrupt enable bit, EMI, and the corresponding timer interrupt enable bit, ET0I/ET1I, must first be set. An actual Timer/Event Counter interrupt will take place when the Timer/Event Counter interrupt request flag, T0F/T1F, is set, a situation that will occur when the Timer/Event Counter overflows. When the interrupt is enabled, the stack is not full and a Timer/Event Counter overflow occurs, a subroutine call to the timer interrupt vector at location 08H/0CH, will take place. When the interrupt is serviced, the timer interrupt request flag, T0F/T1F, will be automatically reset and the EMI bit will be automatically cleared to disable other interrupts.

USB Interrupt

The USB interrupts are triggered by the following USB events causing the related interrupt request flag, USBF, to be set.

- Access of the corresponding USB FIFO from PC
- A USB suspend signal from the PC
- A USB resume signal from the PC
- A USB Reset signal

When the interrupt is enabled, the stack is not full and the USB interrupt is active, a subroutine call to location 04H will occur. The interrupt request flag, USBF, and the EMI bit will be cleared to disable other interrupts.

When the PC Host accesses the FIFO of the device, the corresponding request bit, USR, is set, and a USB interrupt is triggered. So the user can easy determine which FIFO has been accessed. When the interrupt has been served, the corresponding bit should be cleared by firmware. When the device receive a USB Suspend signal from Host PC, the suspend line (bit0 of USC) is set and a USB interrupt is also triggered.

Also when device receive a Resume signal from Host PC, the resume line (bit3 of USC) is set and a USB interrupt is triggered.

Serial Interface (SPI) Interrupt

For a Serial Interface (SPI) interrupt to occur the global interrupt enable bit EMI and the corresponding interrupt enable bit, ESII, must first be set. An actual SPI interrupt will take place when the SPI interrupt request flag SIF is set, a situation that will occur when the SPI interrupt event occurs. When the interrupt is enabled, the stack is not full and a SPI interrupt event occurs a subroutine call to SPI vector will take place. When the interrupt is serviced, the SPI interrupt flag SIF will be automatically reset and the EMI bit will be automatically cleared to disable other interrupts.

Programming Considerations

By disabling the 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 control register until the corresponding interrupt is serviced or until the request flag is cleared by a software instruction.

It is recommended that programs do not use the ″CALL subroutine″ instruction within the interrupt subroutine. Interrupts often occur in an unpredictable manner or need to be serviced immediately in some applications. 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.

All of these interrupts have the capability of waking up the processor when in the Power Down Mode.

Only the Program Counter is pushed onto the stack. If the contents of the accumulator or status register are altered by the interrupt service program, which may corrupt the desired control sequence, then the contents should be saved in advance.

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.

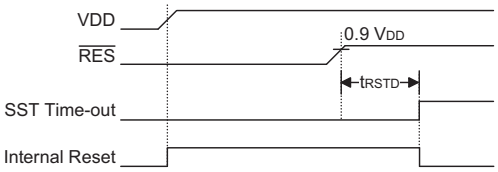
In addition to the power-on reset, situations may arise where it is necessary to forcefully apply a reset condition when the microcontroller is running. One example of this is where after power has been applied and the microcontroller is already running, the $\overline{RES}$ line is forcefully pulled low. In such a case, known as a normal operation reset, some of the microcontroller registers remain unchanged allowing the microcontroller 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 microcontroller. 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 $\overline{RES}$ reset is implemented in situations where the power supply voltage falls below a certain threshold.

Reset Functions

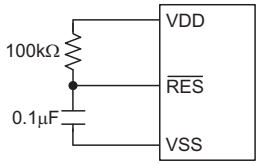
There are five ways in which a microcontroller reset can occur, through events occurring both internally and externally:

- Power-on Reset
  The most fundamental and unavoidable reset is the one that occurs after power is first applied to the microcontroller. 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.
  Although the microcontroller has an internal RC reset function, if the VDD 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 a proper reset operation. In such cases 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 microcontroller 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 microcontroller will begin normal operation. The abbreviation SST in the figures stands for System Start-up Timer.



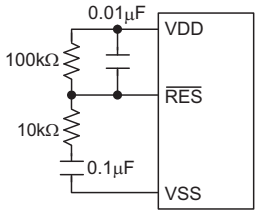
Power-On Reset Timing Chart

For most applications a resistor connected between VDD and the $\overline{RES}$ pin and a capacitor connected between VSS 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.



Basic Reset Circuit

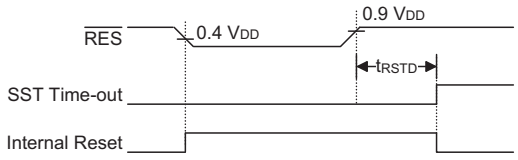
For applications that operate within an environment where more noise is present the Enhanced Reset Circuit shown is recommended.



Enhanced Reset Circuit

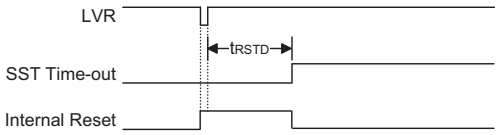
More information regarding external reset circuits is located in Application Note HA0075E on the Holtek website.

- $\overline{RES}$ Pin Reset
  This type of reset occurs when the microcontroller is already running and the $\overline{RES}$ pin is forcefully pulled low by external hardware such as an external switch. In this case as in the case of other reset, the Program Counter will reset to zero and program execution initiated from this point.



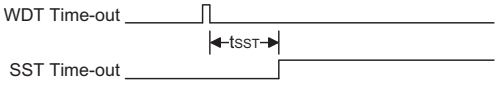
$\overline{RES}$ Reset Timing Chart

- 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 selected via a configuration option. 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. 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 via configuration options.



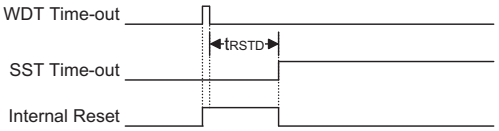
Low Voltage Reset Timing Chart

- 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 Power Down Timing Chart

- Watchdog Time-out Reset during Power Down
  The Watchdog time-out Reset during Power Down 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 Normal Operation Timing Chart

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 Power Down function or Watchdog Timer. The reset flags are shown in the table:

TO	PDF	RESET Conditions
0	0	$\overline{RES}$ reset during power-on
0	0	$\overline{RES}$ wake-up during Power Down
0	0	$\overline{RES}$ or LVR reset during normal operation
1	u	WDT time-out reset during normal operation
1	1	WDT time-out reset during Power Down

Note: "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	Clear after reset, WDT begins counting
Timer/Event Counter	Timer Counter 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 (Power-on)	WDT Time-out (Normal Operation)	$\overline{RES}$ Reset (Normal Operation)	$\overline{RES}$ Reset (HALT)	WDT Time-out (HALT)*	USB Reset (Normal)	USB Reset (HALT)
MP0	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
MP1	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
BP	---- ---0	---- ---0	---- ---0	---- ---0	---- ---u	---- ---0	---- ---0
ACC	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
PCL	0000H	0000H	0000H	0000H	0000H	0000H	0000H
TBLP	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBHP	---- xxxx	---- uuuu	---- uuuu	---- uuuu	---- uuuu	---- uuuu	---- uuuu
TBLH	-xxx xxxx	-uuu uuuu	-uuu uuuu	-uuu uuuu	-uuu uuuu	-uuu uuuu	-uuu uuuu
WDTS	1000 0111	1000 0111	1000 0111	1000 0111	uuuu uuuu	1000 0111	1000 0111
STATUS	--00 xxxx	--1u uuuu	--00 uuuu	--00 uuuu	--11 uuuu	--uu uuuu	--01 uuuu
INTC	-000 0000	-000 0000	-000 0000	-000 0000	-uuu uuuu	-000 0000	-000 0000
TMR0	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
TMR0C	00-0 1---	00-0 1---	00-0 1---	00-0 1---	uu-u u---	uu-u u---	uu-u u---
TMR1H	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
TMR1L	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu
TMR1C	00-0 1---	00-0 1---	00-0 1---	00-0 1---	uu-u u---	uu-u u---	uu-u u---
PA	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
PAC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
PB	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111
PBC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu	1111 1111	1111 1111

Register	Reset (Power-on)	WDT Time-out (Normal Operation)	$\overline{RES}$ Reset (Normal Operation)	$\overline{RES}$ Reset (HALT)	WDT Time-out (HALT)*	USB Reset (Normal)	USB Reset (HALT)
SPIR	0000 0000	0000 0000	0000 0000	0000 0000	0000 uuuu	0000 0000	0000 0000
INTC1	---0 ---0	---0 ---0	---0 ---0	---0 ---0	---u ---u	---0 ---0	---0 ---0
USC	11xx 0000	uuxx uuuu	11xx 0000	11xx 0000	uuxx uuuu	uu00 0u00	uu00 0u00
USR	0000 0000	uuuu uuuu	0000 0000	0000 0000	uuuu uuuu	u1uu 0000	u1uu 0000
SCC	0000 0000	uuuu uuuu	0000 0000	0000 0000	uuuu uuuu	0uu0 u000	0uu0 u000
SBCR	0110 0000	0110 0000	0110 0000	0110 0000	uuuu uuuu	0110 0000	0110 0000
SBDR	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu	xxxx xxxx	xxxx xxxx
USB_STAT	---x xxxx	---x xxxx	---x xxxx	---x xxxx	---x xxxx	---x xxxx	---x xxxx
PIPE_CTRL	0000 0110	0000 0uuu	0000 0110	0000 0110	0000 0uuu	0000 0110	0000 0110
AWR	0000 0000	uuuu uuuu	0000 0000	0000 0000	uuuu uuuu	0000 0000	0000 0000
STALL	0000 0110	0000 0uuu	0000 0110	0000 0110	0000 0uuu	0000 0110	0000 0110
PIPE	0000 0000	xxxx xxxx	0000 0000	0000 0000	xxxx xxxx	0000 0000	0000 0000
SIES	0100 0000	uxux xuuu	0100 0000	0100 0000	uxux xuuu	0100 0000	0100 0000
MISC	0x00 0000	uxuu uuuu	0x00 0000	0x00 0000	uxuu uuuu	0x00 0000	0x00 0000
ENDPT_EN	0000 0111	0000 0uuu	0000 0111	0000 0111	0000 0uuu	0000 0111	0000 0111
FIFO0	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	0000 0000	0000 0000
FIFO1	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	0000 0000	0000 0000
FIFO2	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu	0000 0000	0000 0000

Note: "*" means "warm reset"
"-" not implemented
"u" means "unchanged"
"x" means "unknown"

Oscillator

The clock source for these devices is provided by an integrated oscillator requiring no external components. This oscillator has two fixed frequencies of either 6MHz, or 12MHz, the selection of which is made by the SCLKSEL bit in the SCC register.

Watchdog Timer Oscillator

The WDT oscillator is a fully self-contained free running on-chip RC oscillator with a typical period of 31.2μs at 5V requiring no external components. When the device enters the Power Down Mode, the system clock will stop running but the WDT oscillator continues to free-run and to keep the watchdog active. However, to preserve power in certain applications the WDT oscillator can be disabled via a configuration option.

Power Down Mode and Wake-up

Power Down Mode

All of the Holtek microcontrollers have the ability to enter a Power Down Mode. When the device enters this mode, the normal operating current, will be reduced to an extremely low standby current level. This occurs because when the device enters the Power Down Mode, the system oscillator is stopped which reduces the power consumption to extremely low levels, however, as the device maintains its present internal condition, it can be woken up at a later stage and continue running, without requiring a full reset. This feature is extremely important in application areas where the microcontroller must have its power supply constantly maintained to keep the device in a known condition but where the power supply capacity is limited such as in battery applications.

Entering the Power Down Mode

There is only one way for the device to enter the Power Down Mode and that is to execute the "HALT" instruction in the application program. When this instruction is executed, the following will occur:

- The system oscillator will stop running 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 if the WDT clock source is selected to come from the WDT oscillator. The WDT will stop if its clock source originates from the system clock/4.
- The I/O ports will maintain their present condition.
- 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 Power Down Mode is to keep the current consumption of the microcontroller to as low a value as possible, perhaps only in the order of several micro-amps, 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. 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.

If the configuration options have enabled the Watchdog Timer internal oscillator then this will continue to run when in the Power Down Mode and will thus consume some power. For power sensitive applications it may be therefore preferable to use the system clock source for the Watchdog Timer.

Wake-up

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

- An external reset
- An external falling edge on each port pin
- 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 the I/O ports can be setup via an individual configuration option to permit a negative transition on the pin to wake-up the system. When a port 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 to "1" before entering the Power Down Mode, the wake-up function of the related interrupt will be disabled.

No matter what the source of the wake-up event is, once a wake-up situation occurs, a time period equal to 1024 system clock periods will be required before normal system operation resumes. However, if the wake-up has originated due to an interrupt, the actual interrupt subroutine execution will be delayed by an additional one or more cycles. If the wake-up results in the execution of the next instruction following the "HALT" instruction, this will be executed immediately after the 1024 system clock period delay has ended.

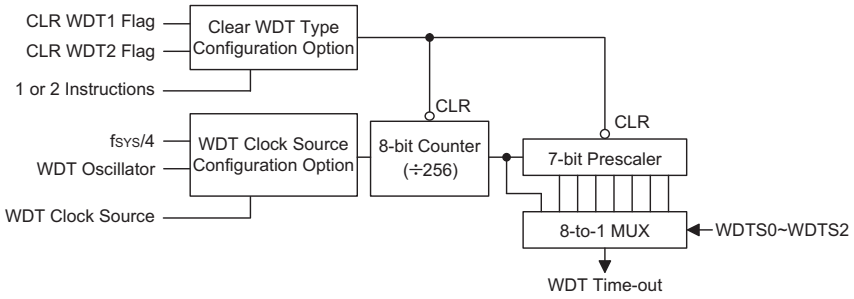
Watchdog Timer

The WDT clock source is implemented by a dedicated RC oscillator (WDT oscillator) or instruction clock (system clock divided by 4), enabled using a configuration option. This timer is designed to prevent a software malfunction or sequence jumping to an unknown location with unpredictable results. If the Watchdog Timer is disabled, all the executions related to the WDT results in no operation.

Once the internal WDT oscillator (RC oscillator normally with a period of 31.2μs) is selected, it is first divided by 256 (8-stages) to get the nominal time-out period of approximately 8ms. This time-out period may vary with temperature, VDD and process variations. By using the WDT prescaler, longer time-out periods can be realized. Writing data to WDTS2, WDTS1, WDTS0 (bit 2, 1, 0 of the WDTS) can give different time-out periods. If WDTS2, WDTS1, WDTS0 are all equal to "1", the division ratio is up to 1:128, and the maximum time-out period is 1s.

If the WDT oscillator is disabled, the WDT clock source may still come from the instruction clock and operate in the same manner except that in the Power down Mode state the WDT may stop counting and lose its protecting purpose. In this situation the WDT logic can be restarted by external logic. The high nibble and bit 3 of the WDTS are reserved for user defined flags, which can be used to indicate some specified status.

If the device operates in a noisy environment, using the on-chip RC oscillator (WDT OSC) is strongly recommended, since the HALT will stop the system clock.



Watchdog Timer

Bit No.	Label	Function
0 1 2	WDTS0 WDTS1 WDTS2	Watchdog Timer division ratio selection bits Bit 2,1,0 = 000, division ratio = 1:1 Bit 2,1,0 = 001, division ratio = 1:2 Bit 2,1,0 = 010, division ratio = 1:4 Bit 2,1,0 = 011, division ratio = 1:8 Bit 2,1,0 = 100, division ratio = 1:16 Bit 2,1,0 = 101, division ratio = 1:32 Bit 2,1,0 = 110, division ratio = 1:64 Bit 2,1,0 = 111, division ratio = 1:128
3	WDTS3	Bit3=1, PDP, and PDN connected to 300kΩ pull-high resistor Bit3=0, No pull-high - default at MCU reset
4~6	—	Not used
7	WDTS7	Bit7=1, USB reset signal can reset MCU and set URST_FLAG (bit 2 of USC register) (default state after a MCU reset) Bit7=0, USB reset signal cannot reset MCU

WDTS Register

Suspend Wake-Up and Remote Wake-Up

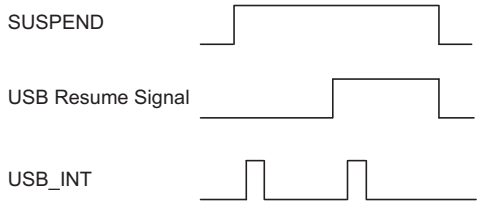
If there is no signal on the USB bus for over 3ms, the device will go into a suspend mode. The Suspend line (bit 0 of the USC register) will be set to ″1″ and a USB interrupt is triggered to indicate that the devices should jump to the suspend state to meet the 500μA USB suspend current spec.

In order to meet the 500μA suspend current, the firmware should disable the USB clock by clearing the USBCKEN bit which is bit3 of the SCC register to ″0″. The suspend current is 400μA.

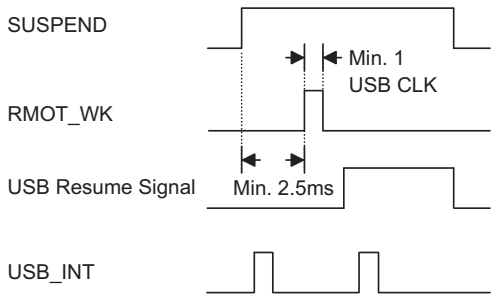
The user can further decrease the suspend current to 250μA by setting the BGOFF bit which is bit4 of the SCC register. If this bit is necessary to be set when the USB entering the suspend mode, the LVR function must be disabled by a configuration option.

When the resume signal is sent out by the host, the devices will wake up the MCU with a USB interrupt and the Resume line (bit 3 of the USC register) is set. In order to make the device function properly, the firmware must set the USBCKEN (bit 3 of the SCC register) to 1 and clear the BGOFF (bit4 of the SCC register). Since the Resume signal will be cleared before the Idle signal is sent out by the host, the Suspend line (bit 0 of the USC register) will be set to ″0″. So when the MCU is detecting the Suspend line (bit0 of USC register), the Resume line condition should be noted and taken into consideration.

After finishing the resume signal, the suspend line will go inactive and a USB interrupt will be triggered. The following is the timing diagram.



As the device has a remote wake up function it can wake-up the USB Host by sending a wake-up pulse through RMWK (bit 1 of the USC register). Once the USB Host receives a wake-up signal from the devices, it will send a Resume signal to the device. The timing is as follows:



To Configure as PS2 Device

The devices can also be configured as a USB interface or PS2 interface device, by configuring the mode control bits named MODE_CTRL1 and MODE_CTRL0 bits (bit 5~4 of the USR register). If the MODE_CTRL1 and MODE_CTRL0 bits are equal to ″1″ and ″0″ respectively, the device will be configured as a PS2 interface, pin USB PDN is configured as a PS2 DATA pin and USB PDP is configured as a PS2 CLK pin. The user can read or write to the PS2 DATA or PS2 CLK pin by accessing the corresponding bit PS2_DAI (bit 4 of the USC register), PS2_CKI (bit 5 of the USC register), PS2_DAO (bit 6 of the USC register) and PS2_CKO (bit 7 of the USC register) respectively.

The user should make sure that in order to read the data properly, the corresponding output bit must be set to ″1″. For example, if it is desired to read the PS2 Data by reading PS2_DAI, then PS2_DAO should set to ″1″. Otherwise it is always read as ″0″.

If the MODE_CTRL1 and MODE_CTRL0 bits are equal to ″0″ and ″1″ respectively, the device is configured as a USB interface. Both the USB PDN and USB PDP are driven by the SIE. The user can only write or read the USB data through the corresponding FIFO. Both the default value of the MODE_CTRL1 and MODE_CTRL0 bits is ″00B″.

USB Interface

There are eleven registers used for the USB function. The AWR register contains the current address and a remote wake up function control bit. The initial value of AWR is ″00H″. The address value extracted from the USB command is not to be loaded into this register until the SETUP stage is completed.

Bit No.	Label	R/W	Function
0	WKEN	W	Remote wake-up enable/disable
7~1	AD6~AD0	W	USB device address

AWR (42H) Register

Bit No.	Label	R/W	Function
0	SUSPEND	R	Read only, USB suspend indication. When this bit is set to ″1″ (set by the SIE), it indicates that the USB bus has entered the suspend mode. The USB interrupt is also triggered on any change of this bit.
1	RMOT_ WK	W	USB remote wake up command. Set by the MCU to force the USB host to leave the suspend mode. When this bit is set to ″1″, a 2μs delay for clearing this bit to ″0″ is needed to insure the RMWK command is accepted by SIE.
2	URST_ FLAG	R/W	USB reset indication. This bit is set/cleared by the USB SIE. This bit is used to detect which bus (PS2 or USB) is attached. When the URST is set to ″1″, this indicates that a USB reset has occurred (the attached bus is USB) and a USB interrupt will be initialised.
3	RESUME_O	R	USB resume indication. When the USB leaves the suspend mode, this bit is set to ″1″ (set by the SIE). This bit will appear for 20ms waiting for the MCU to detect. When the RESUME is set by the SIE, an interrupt will be generated to wake-up the MCU. In order to detect the suspend state, the MCU should set the USBCKEN and clear SUSP2 (in the SCC register) to enable the SIE detect function. The RESUME will be cleared while SUSP is going to ″0″. When the MCU is detecting the SUSP, the condition of RESUME (which wakes-up the MCU) should be noted and taken into consideration.
4	PS2_DAI	R	Read only, PDN/DATA input
5	PS2_CKI	R	Read only, PDP/CLK input
6	PS2_DAO	W	Data for driving the PDN/DATA pin when working under 3D PS2 mouse function. (Default=″1″)
7	PS2_CKO	W	Data for driving the PDP/CLK pin when working under 3D PS2 mouse function. (Default=″1″)

USC (20H) Register

The USR (USB endpoint interrupt status register) register is used to indicate which endpoint is accessed and to select the serial bus, PS2 or USB. If an endpoint is accessed, the related endpoint request flag will be set to ″1″ and the USB interrupt will occur, if the USB interrupt is enabled and the stack is not full. When the active endpoint request flag is served, the endpoint request flag has to be cleared to ″0″.

Bit No.	Label	R/W	Function
0	EP0_INT	R/W	When this bit is set to ″1″ (set by the SIE), it indicates that endpoint 0 is accessed and a USB interrupt will occur. When the interrupt has been served, this bit should be cleared by firmware.
1	EP1_INT	R/W	When this bit is set to ″1″ (set by the SIE), it indicates that endpoint 1 is accessed and a USB interrupt will occur. When the interrupt has been served, this bit should be cleared by firmware.
2	EP2_INT	R/W	When this bit is set to ″1″ (set by the SIE), it indicates that endpoint 2 is accessed and a USB interrupt will occur. When the interrupt has been served, this bit should be cleared by firmware.
3, 6	—	—	Reserved
4 5	MODE_CTRL0 MODE_CTRL1	R/W	00: Non-USB mode, turn-off V33O, both PDP and PDN can be read and write - default 01: Non-USB mode, has 200 ohm between VDD and V33O, both PDP and PDN can be read and write 10: USB mode, 1.5kΩ between PDN and V33O, V33O output 3.3V, both PDP and PDN are read only 11: Non-USB mode, V33O output 3.3V, both PDP and PDN can be read and write
7	USB_flag	R/W	This flag is used to indicate that the MCU is in the USB mode - Bit=1 This bit is R/W by FW and will be cleared to ″0″ after power-on reset - Default=″0″

USR (21H) Register

There is a system clock control register implemented to select the clock used in the MCU. This register consists of the USB clock control bit, USBCKEN, second suspend mode control bit, SUSP2, and a system clock selection bit, SYSCLK. The PS2 mode indicate bit, PS2_flag, and a system clock adjust control bit, CLK_adj.

Bit No.	Label	R/W	Function
0, 1, 2	—	—	Reserved bit set ″0″
3	USBCKEN	R/W	USB clock control bit. When this bit is set to ″1″, it indicates that the USB clock is enabled. Otherwise, the USB clock is turned-off. (Default=″0″)
4	SUSP2	—	When set to 1, turn-off Band-gap circuit. Default value is 0. In the Power-down Mode this bit should be set high to reduce power consumption.The LVR has no function. In the USB mode this bit cannot be set high.
5	PS2_flag	R/W	This flag is used to indicate that the MCU is in the PS2 mode. (Bit=1) This bit is R/W by FW and will be cleared to ″0″ after power-on reset. (Default=″0″)
6	SYSCLK	R/W	This bit is used to specify the system oscillator frequency used by the MCU. If an Integrated 6MHz oscillator is used, this bit should be set to ″1″. If an Integrated 12MHz oscillator is used, this bit should be cleared to ″0″. (default).
7	CLK_adj	R/W	This bit is used to adjust the system clock for the USB mode for temperature changes. In the Power-down Mode this bit should be set high to reduce power consumption. 0: enable (default) 1: disable

SCC (22H) Register

STALL and PIPE, PIPE_CTRL, ENDPT_EN Registers

The PIPE register represents whether the corresponding endpoint is accessed by the host or not. After an ACT_EN signal has been sent out, the MCU can check which endpoint had been accessed. This register is set only after the a time when the host is accessing the corresponding endpoint.

The STALL register shows whether the corresponding endpoint works or not. As soon as the endpoint works improperly, the corresponding bit must be set.

The PIPE_CTRL Register is used for configuring the IN (Bit=1) or OUT (Bit=0) Pipe. The default is define IN pipe. Bit0 (DATA0) of the PIPE_CTRL Register is used to set the data toggle of any endpoint (except endpoint 0) using data toggles to the value DATA0. Once the user wants any endpoint (except endpoint 0) using data toggles to the value DATA0. the user can output a LOW pulse to this bit. The LOW pulse period must at least 10 instruction cycles.

The Endpt_EN Register is used to enable or disable the corresponding endpoint (except endpoint 0) Enable Endpoint (Bit=1) or disable Endpoint (Bit=0)

The bitmaps are list are shown in the following table:

Register Name	R/W	Bit7~Bit3 Reserved	Bit 2	Bit 1	Bit 0	Default Value
PIPE_CTRL	R/W	—	SETIO2	SETIO1	DATA0	00000111
STALL	R/W	—	STL2	STL1	STL0	00000111
PIPE	R	—	Pipe2	Pipe1	Pipe0	00000000
ENDPT_EN	R/W	—	EP2EN	EP1EN	EP0EN	00000111

PIPE_CTRL (41H), STALL (43H), PIPE (44H) and ENDPT_EN (47H) Registers

The USB_STAT Register (40H) is used to indicate the present USB signal state.

Bit No.	Label	R/W	Function
0	EOP	R/W	This bit is used to indicate the SIE has detected a EOP USB signal in the USB Bus. This bit is set by SIE and cleared by F/W.
1	J_state	R/W	This bit is used to indicate the SIE has detected a J_state USB signal in the USB Bus. This bit is set by SIE and cleared by F/W.
2	K_state	R/W	This bit is used to indicate the SIE has detected a K_state USB signal in the USB Bus. This bit is set by SIE and cleared by F/W.
3	SE0	R/W	This bit is used to indicate the SIE has detected a SE0 noise in the USB Bus. This bit is set by SIE and cleared by F/W.
4	SE1	R/W	This bit is used to indicate the SIE has detected a SE1 noise in the USB Bus. This bit is set by SIE and cleared by F/W.
5~7	—	—	Unused bit, read as ″0″

USB_STAT (40H) Register

The SIES Register is used to indicate the present signal state in which the SIE receives and also defines whether the SIE has to change the device address automatically.

Bit No.	Label	R/W	Function
0	ADR_SET	R/W	This bit is used to configure the SIE to automatically change the device address with the value of the Address+Remote_WakeUp Register. When this bit is set to ″1″ by F/W, the SIE will update the device address with the value of the Address+Remote_WakeUp Register after the PC Host has successfully read the data from the device by the IN operation. The SIE will clear the bit after updating the device address. Otherwise, when this bit is cleared to ″0″, the SIE will update the device address immediately after an address is written to the Address+Remote_WakeUp Register. Default 0.
1	F0_ERR	R/W	This bit is used to indicate that some errors have occurred when accessing the FIFO0. This bit is set by SIE and cleared by F/W. Default 0
2	OUT	R/W	This bit is used to indicate there are OUT token (except the OUT zero length) token has been received. The F/W clear the bit after the OUT data has been read. Also, this bit will be clear by SIE after the next valid SETUP token is received.
3	IN	R	This bit is used to indicate the current USB receiving signal from PC Host is IN token.
4	NAK	R	This bit is used to indicate the SIE is transmitted NAK signal to Host in response to PC Host IN or OUT token.
5	CRC_ERR	R/W	This bit indicated there are CRC error, PID error, Bit stuffing error (bit=1). Firmware must to do something to save device keep alive. This bit is set by SIE and clear by F/W.
6	EOT	R	End of transaction flag, normal status is ″1″. If Suspend=″1″ line & EOT=″0″ indicated something wrong in USB Interface. Firmware must to do something to save device keep alive.
7	NMI	R/W	This bit is used to control whether the USB interrupt is output to the MCU in a NAK response to the PC Host IN or OUT token. Only for Endpoint0 1: has only USB interrupt, data is transmitted to the PC host or data is received from the PC Host 0: always has USB interrupt if the USB accesses FIFO0 Default 0

SIES (45H) Register

The MISC register combines a command and status to control desired endpoint FIFO action and to show the status of the desired endpoint FIFO. The MISC will be cleared by the USB reset signal.

Bit No.	Label	R/W	Function
0	REQ	R/W	After setting the other status of the desired one in the MISC, endpoint FIFO can be requested by setting this bit to ″1″. After the task is completed, this bit must be cleared to ″0″.
1	TX	R/W	This bit defines the direction of data transferring between the MCU and endpoint FIFO. When the TX is set to ″1″, this means that the MCU wants to write data to the endpoint FIFO. After the task is completed, this bit must be cleared to ″0″ before terminating the request to represent the end of transferring. For a read action, this bit has to be cleared to ″0″ to represent that MCU wants to read data from the endpoint FIFO and has to be set to ″1″ after completion.
2	CLEAR	R/W	Clear the requested endpoint FIFO, even if the endpoint FIFO is not ready.
4 3	SELP1 SELP0	R/W	Defines which endpoint FIFO is selected, SELP1,SELP0: 00: endpoint FIFO0 01: endpoint FIFO1 10: endpoint FIFO2 11: reserved
5	SCMD	R/W	Used to show that the data in the endpoint FIFO is a SETUP command. This bit has to be cleared by firmware. That is to say, even if the MCU is busy, the device will not miss any SETUP commands from the host.
6	READY	R	Read only status bit, this bit is used to indicate that the desired endpoint FIFO is ready for operation.
7	LEN0	R/W	Used to indicate that a 0-sized packet has been sent from a host to the MCU. This bit should be cleared by firmware.

MISC (46H) Register

The MCU can communicate with the endpoint FIFO by setting the corresponding registers, of which the address is listed in the following table. After reading the current data, the next data will show after 2μs, this is used to check the endpoint FIFO status and response to the MISC register, if the read/write action is still going on.

Registers	R/W	Bank	Address	Bit7~Bit0
FIFO0	R/W	1	48H	Data7~Data0
FIFO1	R/W	1	49H	Data7~Data0
FIFO2	R/W	1	4AH	Data7~Data0

There are some timing constrains and usages illustrated here. By setting the MISC register, the MCU can perform reading, writing and clearing actions. There are some examples shown in the following table for endpoint FIFO reading, writing and clearing.

Actions	MISC Setting Flow and Status
Read FIFO0 sequence	00H→01H→delay 2μs, check 41H→read* from FIFO0 register and check not ready (01H)→03H→02H
Write FIFO1 sequence	0AH→0BH→delay 2μs, check 4BH→write* to FIFO1 register and check not ready (0BH)→09H→08H
Check whether FIFO0 can be read or not	00H→01H→delay 2μs, check 41H (ready) or 01H (not ready)→00H
Check whether FIFO1 can be written or not	0AH→0BH→delay 2μs, check 4BH (ready) or 0BH (not ready)→0AH
Read 0-sized packet sequence form FIFO0	00H→01H→delay 2μs, check 81H→read once (01H)→03H→02H
Write 0-sized packet sequence to FIFO1	0AH→0BH→delay 2μs, check 4BH→09H→08H

Note: *: There is a 2μs time between 2 read actions or between 2 write actions.

SPI Serial Interface

The device includes one SPI Serial Interfaces. The SPI interface is a full duplex serial data link, originally designed by Motorola, which allows multiple devices connected to the same SPI bus to communicate with each other. The devices communicate using a master/slave technique where only the single master device can initiate a data transfer. A simple four line signal bus is used for all communication.

SPI Interface Communication

Four lines are used for each function. These are, SDI - Serial Data Input, SDO - Serial Data Output, SCK - Serial Clock and $\overline{SCS}$ - Slave Select. Note that the condition of the Slave Select line is conditioned by the CSEN bit in the SBCR control register. If the CSEN bit is high then the $\overline{SCS}$ line is active while if the bit is low then the $\overline{SCS}$ line will be I/O mode. The accompanying timing diagram depicts the basic timing protocol of the SPI bus.

SPI Registers

There are three registers for control of the SPI Interface. These are the SBCR register which is the control register and the SBDR which is the data register and SPIR register which is the SPI mode control register. The SBCR register is used to setup the required setup parameters for the SPI bus and also used to store associated operating flags, while the SBDR register is used for data storage.

The SPIR register is used to select SPI mode, clock polarity edge selection and SPI enable or disable selection.

After Power on, the contents of the SBDR register will be in an unknown condition while the SBCR register will default to the condition below:

CKS	M1	M0	SBEN	MLS	CSEN	WCOL	TRF
0	1	1	0	0	0	0	0

Note that data written to the SBDR register will only be written to the TXRX buffer, whereas data read from the SBDR register will actual be read from the register.

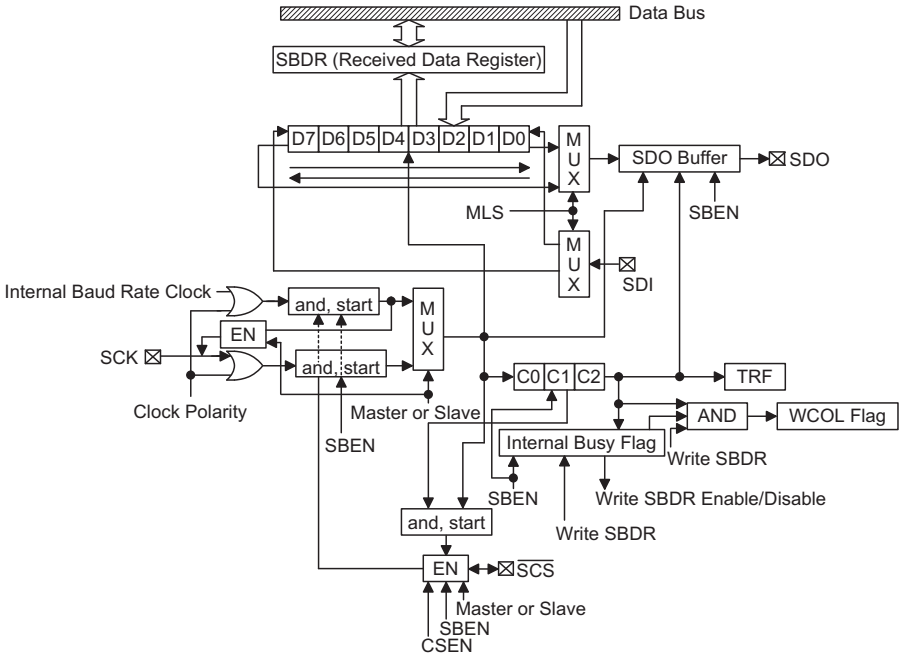
SPI Bus Enable/Disable

To enable the bus, the SBEN bit should be set high, then wait for data to be written to the SBDR (TXRX buffer) register. For the Master Mode, after data has been written to the SBDR (TXRX buffer) register then transmission or reception will start automatically. When all the data has been transferred, the TRF bit should be set. For the Slave Mode, when clock pulses are received on SCK, data in the TXRX buffer will be shifted out or data on SDI will be shifted in.

To Disable the SPI bus SCK, SDI, SDO, $\overline{SCS}$ should be I/O mode.

Bit No.	Label	R/W	Function
0	SPI_CPOL	R/W	0: clock polarity falling (default falling) 1: clock polarity rising
1	SPI_MODE	R/W	0: SPI output the data in the rising edge(polarity=1) or falling edge (polarity=0); SPI read data in the in the falling edge(polarity=1) or rising edge (polarity=0); (default) 1: SPI first output the data immediately after the SPI is enable. And SPI output the data in the falling edge(polarity=1) or rising edge (polarity=0); SPI read data in the in the rising edge(polarity=1) or falling edge (polarity=0)
2	SPI_CSEN	R/W	0: SPI_CSEN disable, $\overline{SCS}$ define as GPIO (default disable) 1: SPI_CSEN Enable , this bit is used to enable/disable software CSEN function
3	SPI_EN	R/W	This bit control the shared PIN ($\overline{SCS}$, SDI, SDO and SCK) is SPI or GPIO mode 0: I/O mode (default) 1: SPI mode
7~4	Reserved bit	R/W	Always 0

SPIR Register



SPI Block Diagram

Note: WCOL: set by SPI cleared by users

CSEN: enable/disable chip selection function pin
master mode: 1/0 = with/without $\overline{SCS}$ output function
Slave mode: 1/0 = with/without $\overline{SCS}$ input control function

SBEN: enable/disable serial bus (0: initialise all status flags)
when SBEN=0, all status flags should be initialised
when SBEN=1, all SPI related function pins should stay at floating state

TRF: 1 = data transmitted or received, 0= data is transmitting or still not received

CPOL: I/O = clock polarity rising/falling edge: For SPIR Register.

If clock polarity set to rising edge (SPI_CPOL=1), serial clock timing follow SCK, otherwise (SPI_CPOL=0) SCK is the serial clock timing.

SPI Operation

All communication is carried out using the 4-line interface for both Master or Slave Mode. The timing diagram shows the basic operation of the bus.

The CSEN bit in the SBCR register controls the $\overline{SCS}$ line of the SPI interface. Setting this bit high, will enable the SPI interface by allowing the $\overline{SCS}$ line to be active, which can then be used to control the SPI inteface. If the CSEN bit is low, the $\overline{SCS}$ line will be in a floating condition and can therefore not be used for control of the SPI interface. The SBEN bit in the SBCR register must also be high which will place the SDI line in a floating condition and the SDO line high. If in the Master Mode the SCK line will be either high or low depending upon the clock polarity control bit in SPIR register. If in the Slave Mode the SCK line will be in a floating condition. If SBEN is low then the bus will be disabled and $\overline{SCS}$, SDI, SDO and SCK will all be I/O mode.

In the Master Mode, the Master will always generate the clock signal. The clock and data transmission will be initiated after data has been written to the SBDR register. In the Slave Mode, the clock signal will be received from an external master device for both data transmission or reception. The following sequences show the order to be followed for data transfer in both Master and Slave Mode:

- Master Mode
  - Step 1. Select the clock source using the CKS bit in the SBCR control register
  - Step 2. Setup the M0 and M1 bits in the SBCR control register to select the Master Mode and the required Baud rate. Values of 00, 01 or 10 can be selected.
  - Step 3. Setup the CSEN bit and setup the MLS bit to choose if the data is MSB or LSB first, this must be same as the Slave device.
  - Step 4. Setup the SBEN bit in the SBCR control register to enable the SPI interface.
  - Step 5. For write operations: write the data to the SBDR register, which will actually place the data into the TXRX buffer. Then use the SCK and $\overline{SCS}$ lines to output the data. Goto to step 6.For read operations: the data transferred in on the SDI line will be stored in the TXRX buffer until all the data has been received at which point it will be latched into the SBDR register.
  - Step 6. Check the WCOL bit, if set high then a collision error has occurred so return to step5. If equal to zero then go to the following step.
  - Step 7. Check the TRF bit or wait for an SPI serial bus interrupt.
  - Step 8. Read data from the SBDR register.
  - Step 9. Clear TRF.
  - Step10. Goto step 5.
- Slave Mode:
  - Step 1. The CKS bit has a don't care value in the slave mode.
  - Step 2. Setup the M0 and M1 bits to 11 to select the Slave Mode. The CKS bit is don't care.
  - Step 3. Setup the CSEN bit and setup the MLS bit to choose if the data is MSB or LSB first, this must be same as the Master device.
  - Step 4. Setup the SBEN bit in the SBCR control register to enable the SPI interface.
  - Step 5. For write operations: write data to the SBDR register, which will actually place the data into the TXRX register, then wait for the master clock and $\overline{SCS}$ signal. After this goto Step 6.
    For read operations: the data transferred in on the SDI line will be stored in the TXRX buffer until all the data has been received at which point it will be latched into the SBDR register.
  - Step 6. Check the WCOL bit, if set high then a collision error has occurred so return to step5. If equal to zero then goto the following step.
  - Step 7. Check the TRF bit or wait for an SPI serial bus interrupt.
  - Step 8. Read data from the SBDR register.
  - Step 9. Clear TRF
  - Step10. step 5

SPI Configuration Options and Status Control

One option is to enable the operation of the WCOL, write collision bit, in the SBCR register. Some control in SPIR register. The SPI_CPOL select the clock polarity of the SCK line . The SPI_MODE select SPI data output mode.

SPI include four pins , can share I/O mode status . The status control combine with four bits for SPIR and SBCR register. Include SPI_CSEN , SPI_EN for SPIR register and CSEN ,SBEN for SBCR register.

SPIR(22H)		SBCR(23H)		I/O Status		Note
SPI_EN	SPI_CSEN	SBEN	CSEN	SPI	$\overline{SCS}$	
0	x	x	x	I/O mode	I/O mode	
1	x	0	x	I/O mode	I/O mode	
1	0	1	x	SPI mode	I/O mode	$\overline{SCS}$ not Floating
1	1	1	0	SPI mode	I/O mode	$\overline{SCS}$ not Floating
1	1	1	1	SPI mode	$\overline{SCS}$ mode	The SPI enable, $\overline{SCS}$, SDI, SDO, SCK the internal Pull-high function is invalid.

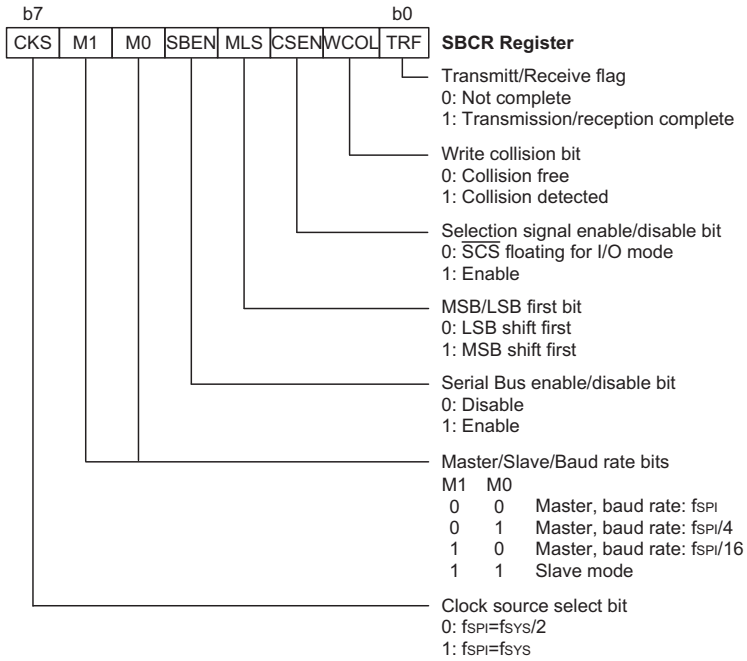
Note: X: don't care

Error Detection

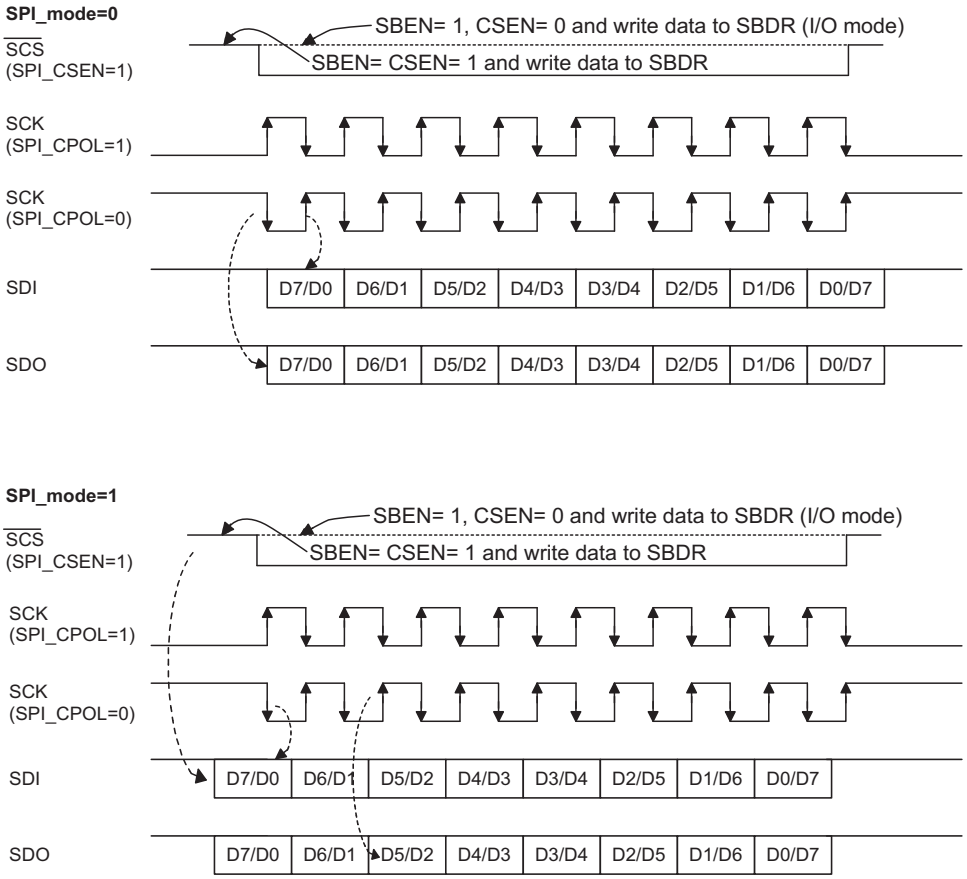
The WCOL bit in the SBCR register is provided to indicate errors during data transfer. The bit is set by the Serial Interface but must be cleared by the application program. This bit indicates a data collision has occurred which happens if a write to the SBDR register takes place during a data transfer operation and will prevent the write operation from continuing. The bit will be set high by the Serial Interface but has to be cleared by the user application program. The overall function of the WCOL bit can be disabled or enabled by a configuration option.

Programming Considerations

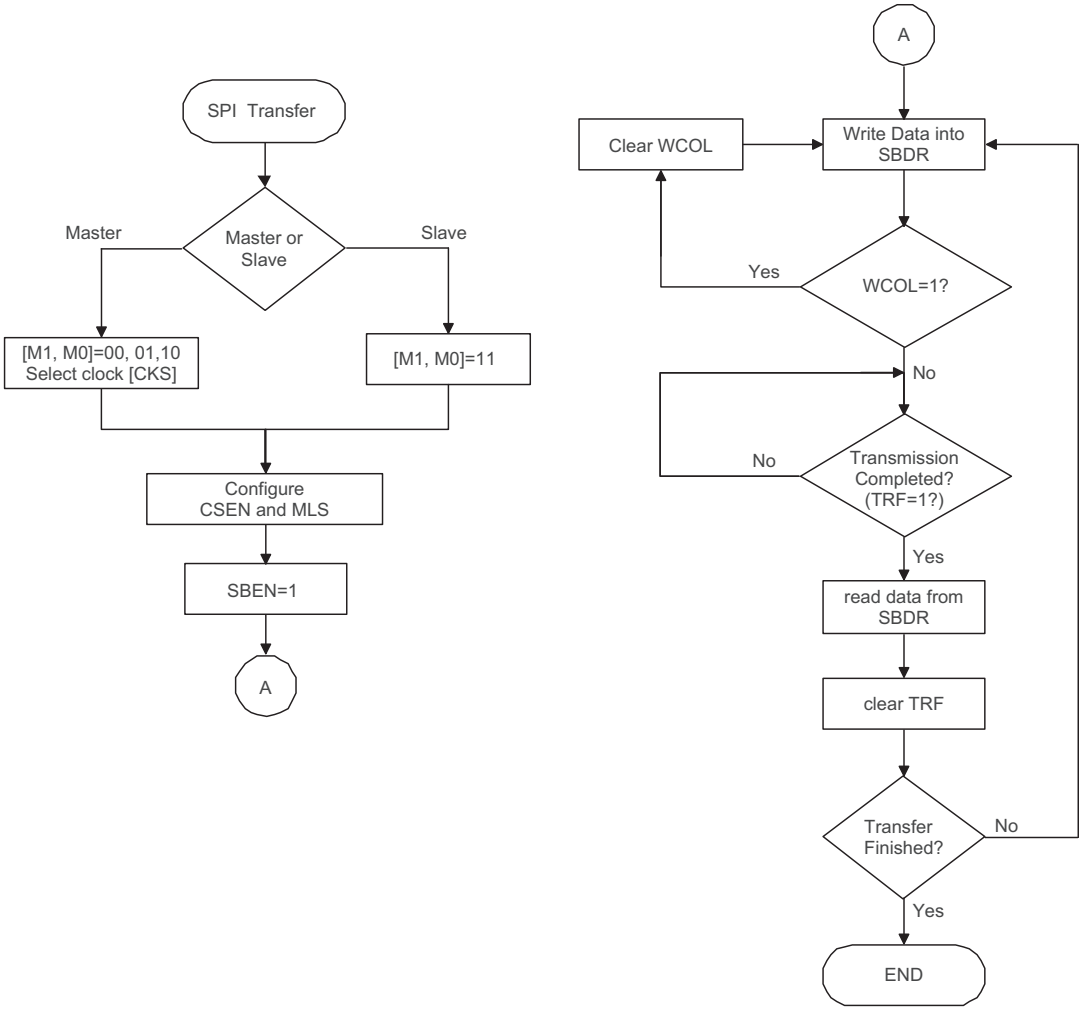
When the device is placed into the Power Down Mode note that data reception and transmission will continue. The TRF bit is used to generate an interrupt when the data has been transferred or received.



SPI Interface Control Register



SPI Bus Timing



SPI Transfer Control Flowchart

Configuration Options

No.	Options
1	PA0~PA7 Pull-high by bit (default Pull-high)
2	PB0~PB7 Pull-high by bit (default Pull-high)
3	PA output mode (CMOS/NMOS/PMOS) by bit (default CMOS)
4	PA0~PA7 wake-up by bit (default enable)
5	PB0~PB7 wake-up by bit (default disable)
6	LVR enable/disable (default enable)
7	WDT function : enable, disable for normal mode (default enable)
8	WDT clock source: 32K RC OSC; T1 (default T1)
9	CLRWDT instruction is by 1 or 2 (default 1 CLRWDT instruction)
10	TBHP enable /disable (default disable)
11	SPI_WCOL enable/disable (default disable)

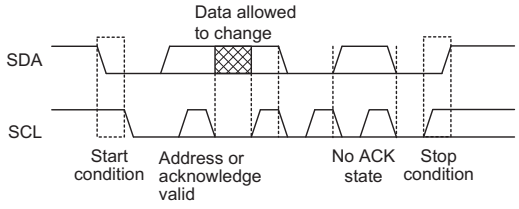
EEPROM Data Memory

The embedded EEPROM Data Memory is an I^2C type device and therefore operates using a two wire serial bus. It has a capacity is 1K organized into a structure of 128 8-bit words and contains the information or data important for user.

EEPROM Memory Interface

The two I^2C lines are the Serial Clock line, SCL, and the Serial Data line SDA. The SDA and SCL lines are bonded to external pins used to control the overall Read and Write operations on the EEPROM Data Memory.

- Serial data - SDA
  The SDA line is the bidirectional EEPROM serial data line which is controlled by the external device such as other host MCU. The host MCU should configure its relative pin connected to the SDA lines as input or output dynamically opposite to the data direction of the EEPROM.
- Serial data - SCL
  The SCL line is the EEPROM serial clock input line which is controlled by the external device such as other host MCU. The host MCU should configure its relative pin connected to the SCL line as output pin. The SCL input clocks data into the EEPROM on its positive edge and clocks data out of the EEPROM on its negative edge.
- Clock and data transition
  Data transfer may be initiated only when the bus is not busy. During data transfer, the data line must remain stable whenever the clock line is high. Changes in the data line while the clock line is high will be interpreted as a START or STOP condition.
- Start condition
  A high-to-low transition of SDA with SCL high will be interpreted as a start condition which must precede any other command - refer to the Start and Stop Definition Timing diagram.



Start and Stop Definition Timing Diagram

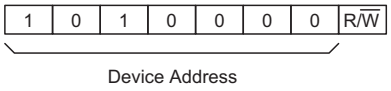
- Stop condition
  A low-to-high transition of SDA with SCL high will be interpreted as a stop condition. After a read sequence the stop command will place the EEPROM in a standby power mode - refer to Start and Stop Definition Timing Diagram.
- Acknowledge
  All addresses and data words are serially transmitted to and from the EEPROM in 8-bit words. The EEPROM sends a zero to acknowledge that it has received each word. This happens during the ninth clock cycle.

EEPROM Memory Addressing

The EEPROM memory requires an 8-bit device address word following a start condition to enable the EEPROM for read or write operations. The device address word consist of a mandatory one, zero sequence for the first four most significant bits. Refer to the diagram showing the Device Address. This is common to all the EEPROM devices. The next three bits are all zero bits.

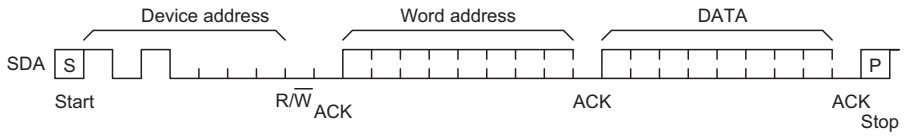
The 8th bit of device address is the read/write operation select bit. A read operation is initiated if this bit is high and a write operation is initiated if this bit is low.

If the comparison of the device address is successful then the EEPROM will output a zero as an ACK bit. If not, the EEPROM will return to a standby state.



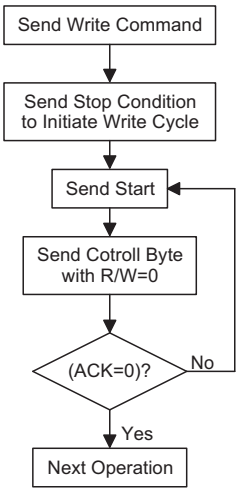
EEPROM Memory Operations

- Byte write
  A write operation requires an 8-bit data word address following the device address word and acknowledgment. Upon receipt of this address, the EEPROM will again respond with a zero and then clock in the first 8-bit data word. After receiving the 8-bit data word, the EEPROM will output a zero and the addressing device must terminate the write sequence with a stop condition. At this time the EEPROM enters an internally-timed write cycle to the non-volatile memory. All inputs are disabled during this write cycle and EEPROM will not respond until the write cycle is completed.



Byte Write Timing

- Acknowledge polling
  To maximize bus throughput, one technique is to allow the master to poll for an acknowledge signal after the start condition and the control byte for a write command have been sent. If the device is still busy implementing its write cycle, then no ACK will be returned. The master can send the next read/write command when the ACK signal has finally been received.



Acknowledge Polling Flow

- Read operations
  The data EEPROM supports three read operations, namely, current address read, random address read and sequential read. During read operation execution, the read/write select bit should be set to ″1″.

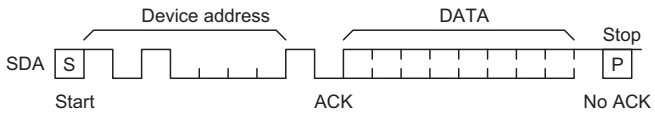
- Current address read
  The internal data word address counter maintains the last address accessed during the last read or write operation, incremented by one. This address stays valid between operations as long as the EEPROM power is maintained. The address will roll over during a read from the last byte of the last memory page to the first byte of the first page. Once the device address with the read/write select bit set to one is clocked in and acknowledged by the EEPROM, the current address data word is serially clocked out. The microcontroller should respond a No ACK - High - signal and a following stop condition.

- Random read
  A random read requires a dummy byte write sequence to load in the data word address which is then clocked in and acknowledged by the EEPROM. The microcontroller must then generate another start condition. The microcontroller now initiates a current address read by sending a device address with the read/write select bit high. The EEPROM acknowledges the device address and serially clocks out the data word. The microcontroller should respond with a No ACK signal - high - followed by a stop condition.

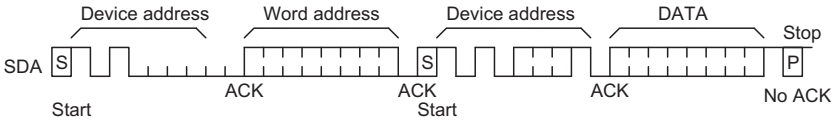
- Sequential read
  Sequential reads are initiated by either a current address read or a random address read. After the microcontroller receives a data word, it responds with an acknowledgment. As long as the EEPROM receives an acknowledgment, it will continue to increment the data word address and serially clock out sequential data words. When the memory address limit is reached, the data word address will roll over and the sequential read continues. The sequential read operation is terminated when the microcontroller responds with a No ACK signal - high - followed by a stop condition.

EEPROM Memory Power-down Considerations

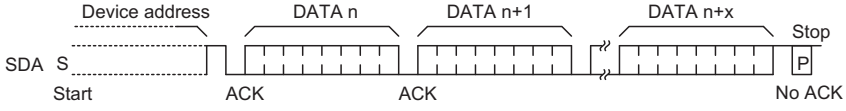
The MCU and EEPROM Memory are powered down independently of each other. The method of powering down the MCU is covered in the previous MCU section of the datasheet. The MCU must be powered down after the read and write operations of the EEPROM Memory have been completed. The method of the read or write operation of the EEPROM Memory is mentioned in the previous EEPROM Memory Operations section of this datasheet.



Current Read Timing

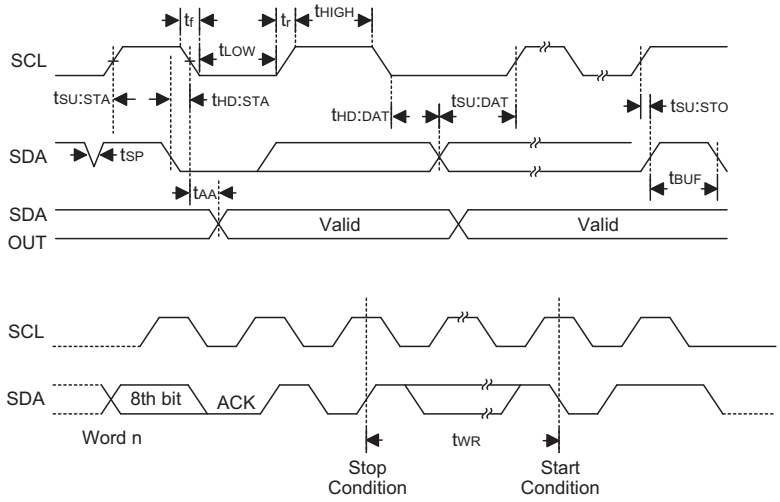


Random Read Timing



Sequential Read Timing

Data EEPROM Timing Diagram



Note: The write cycle time t_{WR} is the time from a valid stop condition of a write sequence to the end of the valid start condition of sequential command.

RF Transceiver

RF Transceiver Features

RF/Analog Circuit Features

- ISM band 2.400GHz~2.495GHz operation
- -90dBm/-80dBm sensitivity @ 250k/1M bps (Packet error rate under 0.1%)
- 3dBm maximum input level
- -3dBm~0dBm typical output power
- Differential RF input/output and integrated TX/RX switch
- Integrated low phase noise VCO, frequency synthesizer and PLL loop filter
- Integrated 32MHz oscillator drive
- Digital VCO and filter calibration
- 18mA in RX and 15mA in TX mode
- 2.4μA deep sleep mode, 0.1μA power down
- 1M bps turbo mode supported
- PLL lock-on time less than 130μs

MAC/Baseband Features

- Automatic ACK response and FCS check
- 62-byte TX FIFO
- Dual 64-byte RX FIFOs
- Various power saving modes
- Simple four-wire SPI interface

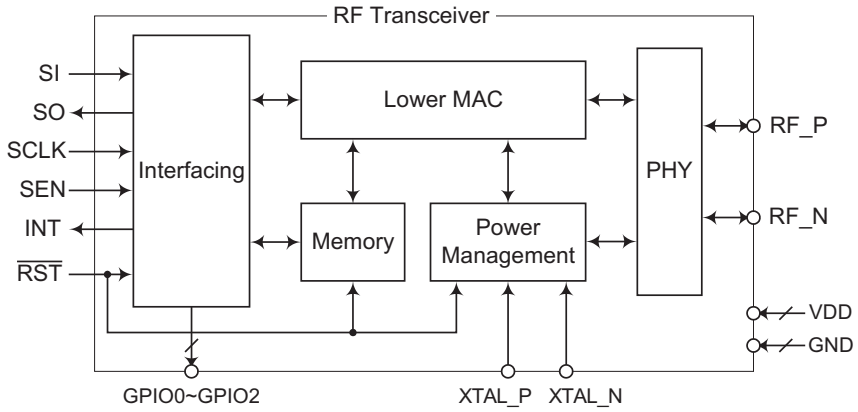
RF Transceiver Applications

- Home/Building/Factory Automation
- PC Peripheral
- RF Remote Controller
- Consumer Electronics
- 2-way Medium-Data-Rate Applications

RF Transceiver Overview

The device contains a 2.4GHz RF transceiver with a Baseband/MAC block. The RF transceiver can be controlled by the MCU for low data rate applications such as consumer electronics, PC peripherals, toys, industrial automations, etc. For medium data rate applications like wireless voice and image transmission, the RF transceiver provides 1M bps turbo mode.

RF Transceiver Block Diagram



RF Transceiver D.C. Characteristics

V_{DD}=3V, Ta=25°C

Symbol	Test Conditions		Min.	Typ.	Max.	Unit
I_{TX}	RF Transceiver TX Active. At 0 dBm output power	DC-DC Off*	—	21	30	mA
I_{RX}	RF Transceiver RX Active in Normal Mode (250 Kbps)	DC-DC Off*	—	19	28	mA
	RF Transceiver RX Active in Turbo Mode (1M bps)	DC-DC Off*	—	21	30	mA
I_{STB}	RF Transceiver in STANDBY mode. Partial 32MHz clock and Sleep clock remains active. RF/MAC/BB, system clock shutdown.		—	60	80	μA
I_{DS}	RF Transceiver in DEEP_SLEEP mode. Power to digital circuit remains active to retain Registers and FIFOs. All the other power is shutdown.		—	3.2	10.0	μA
I_{PD}	RF Transceiver in POWER_DOWN mode. Minimum wake-up circuit remains active. All power is shutdown. Register and FIFO data are not retained.		—	0.6	2.0	μA

Note: "*" The operating current I_{TX} or I_{RX} listed here is the additional current consumed when the RF Transceiver operates in Active TX mode or Active RX mode. If the RF Transceiver is active, either I_{TX} or I_{RX} should be added to calculate the relevant operating current of the device for different operating mode. To calculate the standby current for the whole device, the standby current shown above including I_{STB}, I_{DS} and I_{PD} should be taken into account for different Power Saving Mode.

RF Transceiver A.C. Characteristics

V_{DD}=3V, Ta=25°C, LO frequency=2.445GHz, DC-DC Off

Receiver

Parameters	Test Conditions		Min.	Typ.	Max.	Unit
RF Input Frequency	—		2.400	—	2.495	GHz
RF Sensitivity	At antenna input with O-QPSK signal, PER ≤ 0.1%	250 Kbps	—	-90	—	dBm
		1 Mbps	—	-80	—	dBm
Maximum RF Input	—		—	5	—	dBm
Adjacent Channel Rejection	@ ±5MHz, 250 Kbps (-82 dBm + 20 dB = -62 dBm)		—	20 -62	—	dBc dBm
Alternative Channel Rejection	@ ±10 MHz, 250Kbps (-82 dBm + 40 dB = -42 dBm)		—	40 -42	—	dBc dBm
LO Leakage	Measured at the balun matching the network with the input frequency at 2.4~2.5GHz		—	-60	—	dBm
Noise figure (Including matching)	—		—	8	—	dB

Transmitter

V_{DD}=3V, Ta=25°C, LO frequency=2.445GHz, 250 Kbps, DC-DC Off

Parameters	Test Conditions	Min.	Typ.	Max.	Unit
RF carrier frequency	—	2.400	—	2.495	GHz
Maximum RF output power	At 0 dBm output power setting	-3	0	—	dBm
RF output power Accuracy	—	—	—	±4	dBm
RF output power control range	—	—	36	—	dB
TX gain control resolution	—	0.1	—	0.5	dB
Carrier suppression	—	—	-30	—	dBc
TX spectrum mask for O-QPSK signal	Offset frequency > 3.5MHz At 0 dBm output power	—	—	-30	dBm
		—	—	-20	dBc
TX EVM	—	—	30	—	%

Synthesizer

V_{DD}=3V, Ta=25°C, LO frequency=2.445GHz, 250 Kbps, DC-DC Off

Parameters	Test Conditions	Min.	Typ.	Max.	Unit
PLL Stable Time	—	—	130	—	μs
PLL Programming resolution	—	—	1	—	MHz

RF Transceiver Power-on and Reset Characteristics

The RF Transceiver has built-in power-on reset (POR) circuit which automatically resets all digital registers when the power is turned on. The 32MHz oscillator circuit starts to lock frequency of the right clock after power-on. The whole process takes 3ms for a clock circuit to become stable and completes the power-on reset. It is highly recommended that the user waits at least 3ms before starting to access the RF Transceiver.

The RF Transceiver hardware reset signal (warm start) named $\overline{RST}$ is controlled by MCU I/O pin and internally pulled high with 33kΩ resistor connected to VCC within the RF Transceiver. The RF Transceiver will hold in reset state around 20μs after $\overline{RST}$ signal is released from the low state.

RF Transceiver Crystal Parameter Specifications

The RF Transceiver utilizes external 32MHz crystal to generate the oscillation for RF Transceiver input clock. The associated pins are XTAL_P and XTAL_N. The table below lists the parameters of the crystal oscillator used in the RF Transceiver. To operate the RF Transceiver properly, user has to select the crystal which meets the following requirements.

Parameters	Min.	Typ.	Max.	Unit
Crystal Frequency		32	—	MHz
Frequency Offset	-40	—	40	ppm
Load Capacitance	—	—	10	pF
Recovery Time	—	—	180	μs

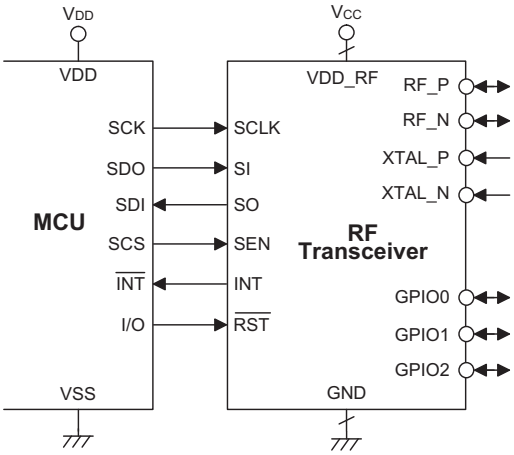
32MHz crystal oscillator recovery time highly depends on the shunt capacitance of 32MHz crystal. The lower shunt capacitance value makes the recovery time shorter. This recovery time 180μs is measured with 32MHz crystal by NDK NX3225SA.

RF Transceiver Functional Description

The RF transceiver integrates receiver, transmitter, voltage-controlled oscillator (VCO), and phase-locked loop (PLL). It uses advanced radio architecture to minimize the external component count and the power consumption. The Baseband/MAC block provides the hardware architecture for both MAC and PHY layers. It mainly consists of TX/RX control and digital signal processing module. Interconnection between the MCU and the RF Transceiver is implemented by internally connecting the MCU Master SPI interface to the RF Transceiver Slave SPI interface. All data transmissions and receptions between MCU and RF Transceiver including RF Transceiver commands are conducted along this interconnected SPI interface. The RF Transceiver function control is executed by the MCU using its SPI Master serial interface. The RF Transceiver contains its own independent interrupt which can be used to indicate when a wake-up event occurs, an available packet reception occurs or when a packet transmission has successfully terminated or retransmission is timed out.

RF Transceiver Internal Connection

In addition to the RF Transceiver external pins described above there are other MCU to RF Transceiver interconnecting lines that are described in the above RF Transceiver Pin Description table. Note that these lines are internal to the device and are not bonded to external pins.



MCU to RF Transceiver Internal Connection

MCU to RF Transceiver Internal Line Descriptions

Pin Name	Type	Description
SCLK	I	Internal RF Transceiver Slave SPI Serial Clock Input Signal. Internally connected to the MCU Master SPI SCK output signal.
SI	I	Internal RF Transceiver Slave SPI Serial Data Input Signal. Internally connected to the MCU Master SPI SDO output signal.
SO	O	Internal RF Transceiver Slave SPI Serial Data Output Signal. Internally connected to the MCU Master SPI SDI input signal.
SEN	I	Internal RF Transceiver Slave SPI Serial interface Enable Input Signal. Internally connected to the MCU Master SPI $\overline{SCS}$ output signal.
$\overline{INT}$	I	Internal RF Transceiver Interrupt Output Signal. Internally connected to the MCU $\overline{INT}$ input signal.
$\overline{RST}$	I	Internal RF Transceiver global hardware reset input signal, active low. Internally connected to the MCU I/O pin configured as output type.
WAKE	I	Internal RF Transceiver Wake-up trigger input signal. Internally connected to the MCU I/O pin configured as output type.

Notes: (1) The pin descriptions for all external pins except the RF Transceiver pins listed in the above table are described in the preceding MCU section.

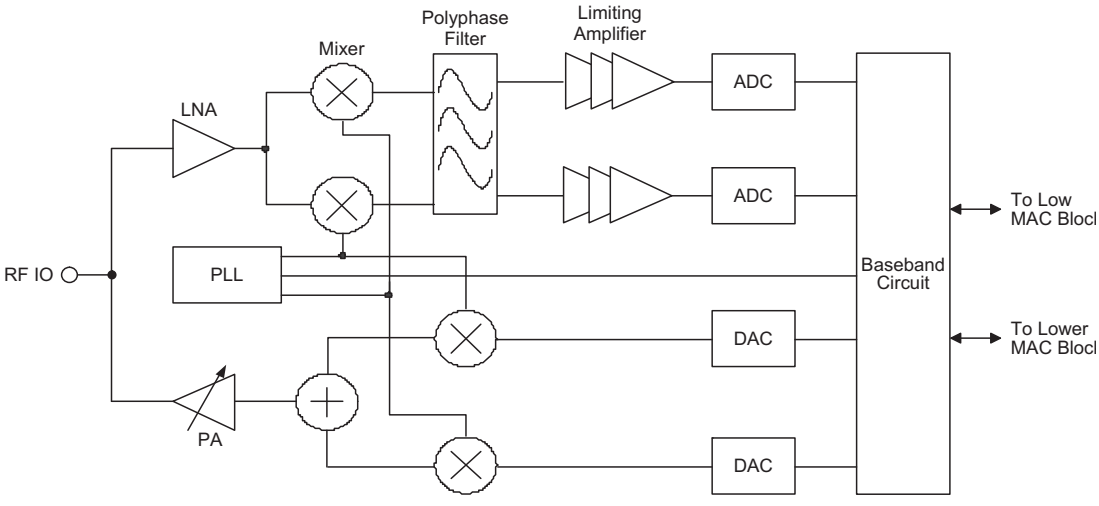
(2) The INT, RSTB and WAKE lines are internally connected to the MCU I/O pins PB0, PB1 and PB2 respectively while the SO, SI, SCLK and SEN lines are internal connected to the MCU I/O pins PB3, PB4, PB5 and PB6 respectively.

The RF Transceiver is composed of several functional blocks named Interfacing block, Lower MAC block, Memory block, Power Management block and PHY block. The detailed functions of the functional blocks are described in the following sections.

RF Transceiver PHY Block

The key features and the block diagram of the PHY layer in RF Transceiver are listed as below.

- Operating frequency range is from 2400MHz to 2495MHz.
- It uses Offset QPSK (OQPSK) modulation to transmit data at 250k/1M bps.
- Direct Sequence Spreading Spectrum (DSSS) is used in baseband algorithm to increase the SNR.

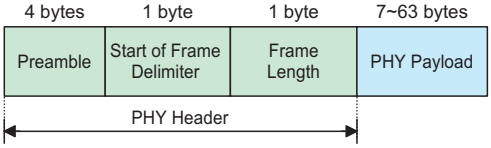


PHY Block Architecture

The RF Transceiver uses a fractional-N Phase-Locked Loop (PLL) as frequency synthesizer. Therefore, 1MHz channel spacing is supported and any integer carrier frequency between 2400MHz to 2495MHz can be used. The loop filters of PLL are integrated into the RF Transceiver except one external capacitor which should be connected between the PLL loop filter external pin and the ground. In order to keep the PLL stable, the board layout around the PLL loop filter external capacitor pin should be carefully designed to avoid EMI. The recommended value of this external capacitor is 100pF.

Under 1M bps turbo mode, user can use the same program settings of MAC and all MAC functions are remained the same. Compare with 250k bps mode, in 1M bps mode, signal bandwidth is extended to 8MHz.

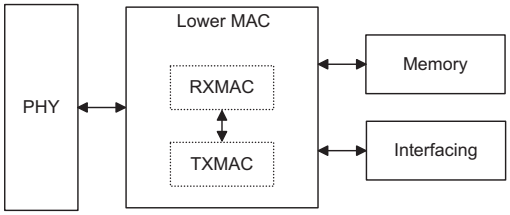
The packet includes a 6 bytes PHY header and a 7~63 bytes PHY payload. The 6 bytes PHY header includes 4 bytes of preamble, 1 byte of start-of-frame delimiter (SFD) and 1 byte of payload length. Preamble and SFD are used for receiver packet detection and synchronization. The Frame Length field specifies the length of the PHY payload field. The valid length can be from 7 to 63 bytes. The frame format is shown as below:



PHY Layer Frame Format

RF Transceiver Low MAC Block

The RF Transceiver MAC provides plenty of hardware-assisted features to relieve the host MCU power requirement. Besides providing reliable wireless packet transactions between two nodes, it also handles data and command transfer between the network and the physical layers PHY.



Lower MAC Block Diagram

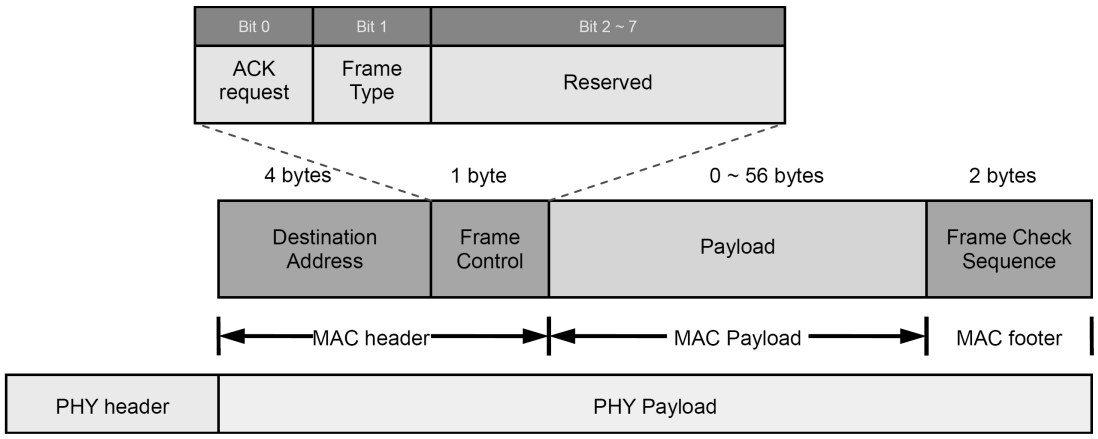
MAC Frame Format

- Data Frame
  The address field contains the broadcast address (0xFFFF-FFFFH) or destination address. The bit 0 of frame control (FC) field is used for Ack-Request which specifies whether an acknowledgement is required from the recipient device. If the bit is ″1″, the recipient device shall send an acknowledgement frame back after determining that the received frame is valid. The bit 1 of FC field is ″0″ for data frame. The length of payload field is variable from 0 to 56 bytes. The frame check sequence (FCS) is calculated over the address field, FC field and the payload. The polynomial is degree 16:

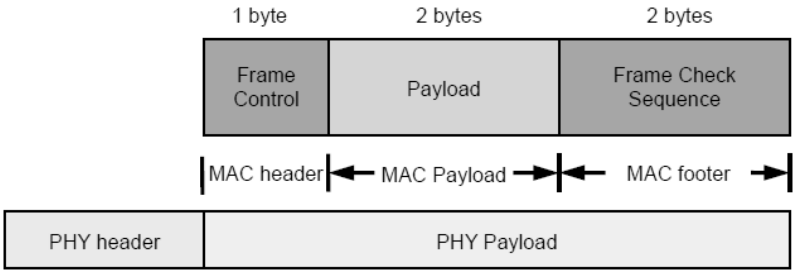
$$G_{16}(x) = x^{16} + x^{12} + x^5 + 1$$

- Acknowledgement Frame
  The length of acknowledgement frame is always 5 bytes. Bit 1 of FC field is 1 for ACK frame. The payload field, containing user information of acknowledgement frame, can be configured by SREG0x03 and SREG0x04. The FCS is calculated over the FCS of the received packet, FC field and the payload field. The polynomial is degree 16:

$$G_{16}(x) = x^{16} + x^{12} + x^5 + 1$$



Data Frame



Acknowledgement Frame

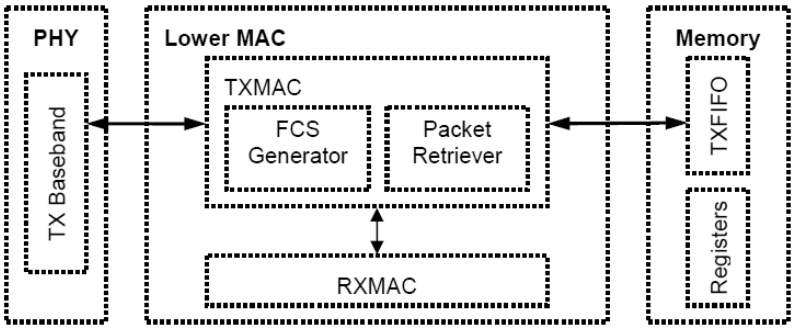
TXMAC

When the TXFIFO is triggered, the TXMAC gets the data from TXFIFO to generate a 16-bit FCS and sends the packet to the PHY layer of the TX immediately. If necessary, TXMAC handles the retransmission, when the acknowledgement packet is not received. The block diagram of a TXMAC is shown below.

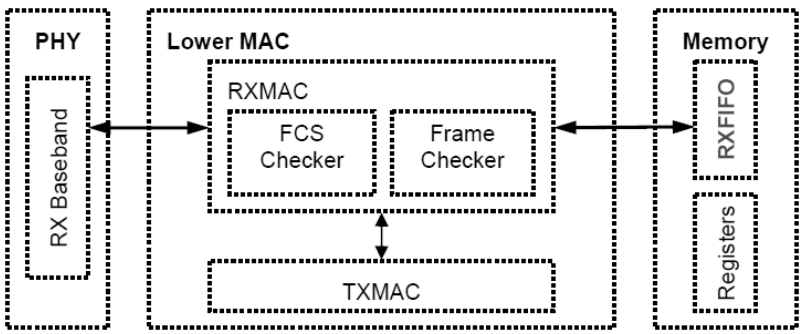
RXMAC

The RX PHY of the RF Transceiver filters signals and tracks the synchronization symbols. If a packet passes the filtering, RXMAC performs frame type parsing, address recognition and FCS checking. If the destination address is broadcast address or matches its own identity, configured by SREG0x05 to SREG0x08, and the FCS check is passed, an interrupt is issued at SREG0x31 [3] to indicate a valid packet is received. Meanwhile, the frame length field of PHY header and PHY payload will be stored in RXFIFO. Unqualified packets are skipped.

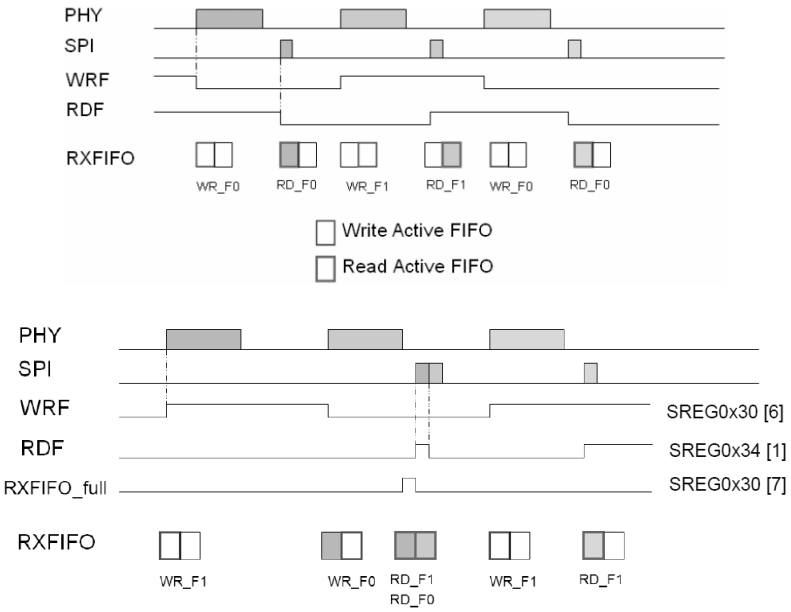
RXFIFO0 and RXFIFO1 are mapped into the 64-byte memory space from 0x300H to 0x33FH as Ping-Pong FIFOs. If Ping-Pong RX mode is enabled by SREG0x34 [0], RXMAC automatically switches between RXFIFO0 and RXFIFO1 to store incoming frame whenever a new packet comes. When the MCU host reads the long address memory 0x300H, the RXMAC will change the flag of SREG0x34 [1] automatically. For manually controlled RX operation, if the value of the flag SREG0x34 [1] is ″0″, the RXFIFO0 shall be read. Otherwise, the RXFIFO1 shall be read.



TXMAC Block Diagram



RXMAC Block Diagram



In the above diagram, the current status of each frame is represented in SREG0x30. SREG0x30 [7] means ″RXFIFO full″ indicating the two RXFIFOs are occupied. If the MCU host cannot read the RXFIFO in time, the value of SREG0x30 [7] will be set to ″1″. Once the MCU host read the RXFIFO, the value of the SREG0x30 [7] will be set to ″0″ automatically.

The contents of the RXFIFO can be flushed only by the following three ways: (1) the MCU host reads length field of RXFIFO and the last byte of the packet, (2) the host issues an RX flush, and (3) the software reset by SREG0x2A [0]. Note that RXFIFO is ready to receive next packet and all the data in RFIFO will be overwritten after RXFIFO flushed.

Auto Acknowledgement

The RXMAC supports automatically acknowledgement. If and only if the packet is successfully received and an Ack-Request bit, Bit 0, in the FC field of the received packet is set, RXMAC informs TXMAC to send an acknowledgement packet automatically. User should write the FC field correctly into the TX FIFO.

If an acknowledgement is requested and the replied ACK frame is not received, the transmitter automatically resends the packet until the maximum retransmission times, specified in SREG0x1B [7:4], are reached. To utilize the function properly, the corresponding registers of both transmitting and receiving sides need to be set correctly.

- Auto-retransmission on TX Side
  To automatically retransmit a packet when an ACK is not received, SREG0x1B [2] is required to be set to ″1″.
- Auto-acknowledgement on RX Side
  To automatically reply an ACK packet when Ack-Request bit is set to ″1″, SREG0x00 [5] should be set to ″0″.

RF Transceiver Memory Block

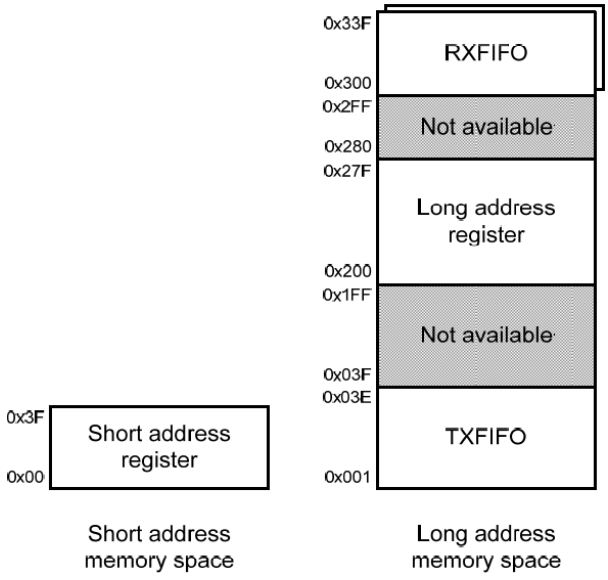
The Memory Block of the RF Transceiver is implemented by the SRAM. As the following Memory Block diagram shown, the RF Transceiver Memory is composed of registers and FIFOs, which can be accessed by the SPI interface. They are categorized into two kinds of address spaces. One is the short address space; the other is the long address space.

Registers

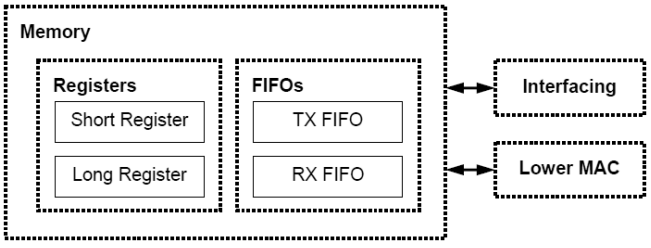
Registers provide control bits and status flags for the RF Transceiver operations, including transmission, reception, interrupt control, MAC/baseband/RF parameter settings, etc. The registers are divided into two types according to addressing mode as listed below.

- Short address register (6-bit short addressing mode register, total 64 registers)
- Long address register (10-bit long addressing mode register, total 128 registers)

Short address registers are accessed by short addressing mode with valid addresses ranging from 0x00H to 0x3FH. Long address registers are accessed by long addressing mode with valid addresses ranging from 0x200H to 0x27FH. Short registers are accessed faster than long registers. Please refer to the following SPI Interface section for detailed addressing rules via SPI interface.



Memory Space Diagram



Memory Block Diagram

Short Address Registers Legend: r=reserved

Addr.	File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR
0x00	RXMCR	r	r	NOACKRSP	r	r	r	r	r	0000 0000
0x03	AUINFL	AUINF7	AUINF6	AUINF5	AUINF4	AUINF3	AUINF2	AUINF1	AUINF0	0000 0000
0x04	AUINFH	AUINF15	AUINF14	AUINF13	AUINF12	AUINF11	AUINF10	AUINF9	AUINF8	0000 0000
0x05	DADR_0	DADR[7:0]								0000 0000
0x06	DADR_1	DADR[15:8]								0000 0000
0x07	DADR_2	DADR[23:16]								0000 0000
0x08	DADR_3	DADR[31:24]								0000 0000
0x0D	RXFLUSH	r	WAKEPOL	WAKEPAD	r	r	PTX	r	RXFLUSH	0110 0000
0x12	ACKTO	r	MATOP6	MATOP5	MATOP4	MATOP3	MATOP2	MATOP1	MATOP0	0011 1001
0x17	PACON	r	r	r	PAONTS3	PAONTS2	PAONTS1	PAONTS0	r	0000 0010
0x18	TXCON	r	r	TXONTS3	TXONTS2	TXONTS1	TXONTS0	r	r	1000 1000
0x1B	TXTRIG	TXRTYN3	TXRTYN2	TXRTYN1	TXRTYN0	r	TXACKREQ	r	TXTRIG	0011 0000
0x22	WAKECTL	IMMWAKE	REGWAKE	r	r	r	r	r	r	0100 0000
0x24	TXSR	TXRETRY3	TXRETRY2	TXRETRY1	TXRETRY0	r	r	r	TXNS	0000 0000
0x26	GATECLK	r	r	SPISYNC	r	r	ENTXM	r	r	0000 0000
0x2A	SOFTRST	r	r	r	r	r	r	RSTBB	RSTMAC	0000 0000
0x2E	TXPEMISP	TXPET3	TXPET2	TXPET1	TXPET0	r	r	r	r	0111 0101
0x30	RXSR	RXFFFULL	WRFF1	r	RXFFOVFL	RXCRCERR	r	r	r	0000 0000
0x31	ISRSTS	r	WAKEIF	r	r	RXIF	r	r	TXNIF	0000 0000
0x32	INTMSK	r	WAKEMSK	r	r	RXMSK	r	r	TXNMSK	1111 1111
0x34	BATRXF	r	r	BATIND	r	r	r	RDFF1	RXFIFO2	0000 0000
0x35	SLPACK	SLPACK	WAKECNT6	WAKECNT5	WAKECNT4	WAKECNT3	WAKECNT2	WAKECNT1	WAKECNT0	0000 0000
0x36	RFCTL	r	r	r	WAKECNT8	WAKECNT7	RFRST	r	r	0000 0000
0x38	BBREG0	r	r	r	r	r	r	r	TURBO	1000 0001

Short Address Registers List

Long Address Registers Legend: r=reserved

Addr.	File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR
0x200	RFCTRL0	CHANNEL3	CHANNEL2	CHANNEL1	CHANNEL0	r	r	r	r	0000 0001
0x201	RFCTRL1	r	r	r	r	r	r	VCORX1	VCORX0	0000 0001
0x202	RFCTRL2	r	RXFC0-1	RXFC0-0	r	r	r	r	r	1000 0100
0x203	RFCTRL3	TXGB4	TXGB3	TXGB2	TXGB1	TXGB0	r	r	r	0000 0000
0x204	RFCTRL4	r	r	r	r	r	RXFCO	RXD2CO1	RXD2CO0	0000 0000
0x205	RFCTRL5	BATTH3	BATTH2	BATTH1	BATTH0	r	r	r	r	0000 0000
0x206	RFCTRL6	TXFBW1	TXFBW0	32MXCO1	32MXCO0	BATEN	r	r	r	1111 0000
0x207	RFCTRL7	r	r	r	RXFC2	r	r	r	r	0000 0000
0x208	RFCTRL8	r	TXD2CO0	r	r	r	r	r	r	0000 1100
0x209	SLPCAL_0	SLPCAL7	SLPCAL6	SLPCAL5	SLPCAL4	SLPCAL3	SLPCAL2	SLPCAL1	SLPCAL0	0000 0000
0x20A	SLPCAL_1	SLPCAL15	SLPCAL14	SLPCAL13	SLPCAL12	SLPCAL11	SLPCAL10	SLPCAL9	SLPCAL8	0000 0000
0x20B	SLPCAL_2	SLPCALRDY	r	r	SLPCALEN	SLPCAL19	SLPCAL18	SLPCAL17	SLPCAL16	0000 0000
0x211	IRQCTRL	r	r	r	r	r	r	IRQCTRL	r	0000 0000
0x22F	TESTMODE	MPSPI	r	r	r	r	TESTMODE2	TESTMODE1	TESTMODE0	0010 1000
0x23D	GPIODIR	r	r	GDIRCTRL2	GDIRCTRL1	GDIRCTRL0	GPIO2DIR	GPIO1DIR	GPIO0DIR	0011 1111
0x23E	GPIO	r	r	r	r	r	GPIO2	GPIO1	GPIO0	0000 0000
0x250	RFCTRL50	r	r	r	DCPOC	DCOPC3	DCOPC2	DCOPC1	DCOPC0	0000 0000
0x251	RFCTRL51	DCOPC5	DCOPC4	r	r	r	r	r	r	0000 0000
0x252	RFCTRL52	SLCTRL6	SLCTRL5	SLCTRL4	SLCTRL3	SLCTRL2	SLCTRL1	SLCTRL0	32MXCTRL	1111 1111
0x253	RFCTRL53	r	FIFOPS	DIGITALPS	P32MXE	PACEN2	PACTRL2-2	PACTRL2-1	PACTRL2-0	0000 0000
0x254	RFCTRL54	1MCSEN	1MFRCH6	1MFRCH5	1MFRCH4	1MFRCH3	1MFRCH2	1MFRCH1	1MFRCH0	0000 0000
0x259	RFCTRL59	r	r	r	r	r	r	r	PLLOPT3	0000 0001
0x273	RFCTRL73	VCOTXOPT1	VCOTXOPT0	r	r	PLLOPT2	PLLOPT1	PLLOPT0	r	0000 0000
0x274	RFCTRL74	PACEN0	PACTRL0-2	PACTRL0-1	PACTRL0-0	PACEN1	PACTRL1-2	PACTRL1-1	PACTRL1-0	1100 1010
0x275	RFCTRL75	r	r	r	r	SCLKOPT3	SCLKOPT2	SCLKOPT1	SCLKOPT0	0001 0101
0x276	RFCTRL76	r	r	r	r	r	SCLKOPT6	SCLKOPT5	SCLKOPT4	0000 0001
0x277	RFCTRL77	r	r	SLPSEL1	SLPSEL0	SLPVCTRL1	SLPVCTRL0	SLPVSEL1	SLPVSEL0	0000 1000

FIFOs

FIFOs serve as the temporary data buffers for data transmission and reception. Each FIFO holds only one packet at a time. TXFIFO, the transmission FIFO, is composed of 62-byte FIFO. RX FIFO, the receiving FIFO, is composed of two 64-byte FIFOs.

- TX FIFO - (62 bytes)
  The TXMAC gets the to-be transmitted data from the 62-byte TXFIFO. The memory space of TXFIFO is from ″0x001″ to ″0x03E″ and contains a FL field, address field, FC field and payload field. The FL field indicates the length of the address field, FC field and the payload field. The valid value of frame length is from 5 to 61 bytes.
- RX FIFO - RXFIFO0 (64 bytes) and RXFIFO1 (64 bytes)
  A RXFIFO is composed of two 64-byte FIFOs (RXFIFO0 and RXFIFO1) to store the incoming packet. Each of them is designed to store one packet at a time. RXFIFO contains a FL field, address field, FC field, payload field and FCS field. The memory space of RXFIFO is from ″0x300″ to ″0x33F″. The FL field, which is extracted from the PHY header, indicates the length of the address field, FC field, the payload field and FCS field. The valid value of frame length is from 7 to 63 bytes. The value of the FL field of PHY header is calculated by adding 2, the length of FCS field of MAC frame, and the above mentioned value up.

RF Transceiver Power Management Block

Almost all wireless sensor network applications require low-power consumption to lengthen battery life. Typical battery-powered device is required to be operated over years without replacing its battery. The RF Transceiver achieves low active current consumption of both the digital and the RF/analog circuits by controlling the supply voltage and using low-power architecture.

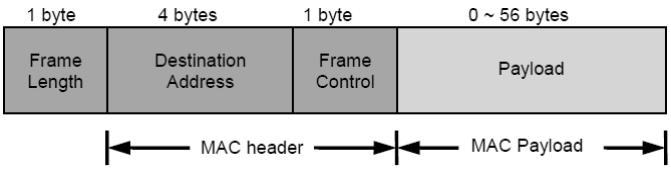
The RF Transceiver has four power saving modes that will be further described in Power Saving Modes Section. For ultra low-power operation, Power-down mode is available which consumes around 0.1μA while the RF Transceiver is powered down. All data stored in registers and FIFOs will be lost under Power-down mode. In this mode, The RF Transceiver is able to wake up by a wake-up input signal. Except Power down mode, the data in the registers/FIFOs are retained during the other power saving modes.

Power Supply Scheme

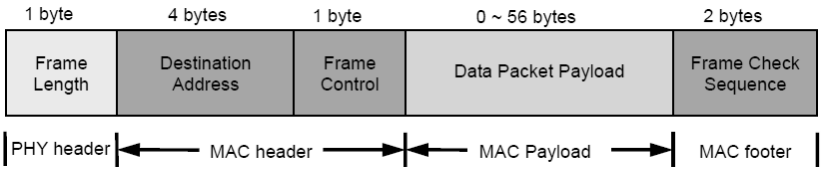
The table below lists the recommended values of the external bypass capacitors for each power pin of the RF Transceiver. For the power pins VDD_RF1 and VDD_3V, an extra bypass capacitor is needed for the decoupling purpose while the rest of the power pins require only one bypass capacitor. The path length between the bypass capacitors to each pin should be made as short as possible.

Pin Name	Bypass Capacitor 1	Bypass Capacitor 2
VDD_RF1	47pF	10nF
VDD_RF2	47pF	
VDD_D	10nF	
VDD_3V	10μF	10nF
VDD_A	47pF	
VDD_PLL	47pF	
VDD_CP	10nF	

Recommended External Bypass Capacitors



TXFIFO Format



RXFIFO Format

DC-DC Converter

There are two ways to supply power to the RF Transceiver. One is through the on-chip DC-DC converter and the other is without the DC-DC converter. With DC-DC converter, the RF Transceiver consumes lower current. With the DC-DC converter, power pins including VDD_RF1, VDD_RF2, VDD_D, VDD_A, VDD_PLL and VDD_CP should be hardwired to the DC-DC converter output, pin VDD_2V2 of the RF Transceiver. Without DC-DC converter, all the power pins should be directly hardwired to the external supplied voltage.

For this device the on-chip DC-DC converter is not used. User can set LREG0x250 [4] to ′0′ and LREG0x273 to ″0x4E″ to bypass the DC-DC converter. When the DC-DC converter is bypassed, pins VDD_2V2 and VDD_3V are shorted internally.

Battery Monitor

The RF Transceiver provides a function to monitor the RF Transceiver supplied voltage. A 4-bit voltage threshold can be configured so that when the supplied voltage is lower than the threshold, the system will be notified. For battery monitor function, please refer to the Section named Battery Monitor Operations.

Power Saving Modes

The RF Transceiver power modes are classified into the following four modes:

- IDLE: RF circuit off. The regulator, oscillator, and digital circuits are on.
- STANDBY: RF/MAC/BB shutdown with sleep and 32MHz clocks remain active
- DEEP_SLEEP: All power is shutdown except the power to the digital circuits and registers and FIFOs data are retained.
- POWER_DOWN: All power is shutdown. Registers and FIFOs data are not retained, a wake-up input signal can wake up the RF Transceiver.

IDLE mode is rarely used because the device should at least always turns on its RX circuit to capture the on-air RF signals. The only difference between STANDBY mode and DEEP_SLEEP mode is the power status of the sleep clock. To wake the RF Transceiver up, the MCU host has to control the time of sleep process.

The power management control is used for the low power operation of MAC and baseband modules. It manages to turn on and off the 32MHz clock when the RF Transceiver goes into power saving mode. By turning off the 32MHz clock, the MAC and baseband circuits become inactive regardless whether their power supplies exist or not. All the digital modules are clock-gated automatically. That means only when a module is functioning, its clock would then be turned on. This approach efficiently decreases certain amount of the current consumption.

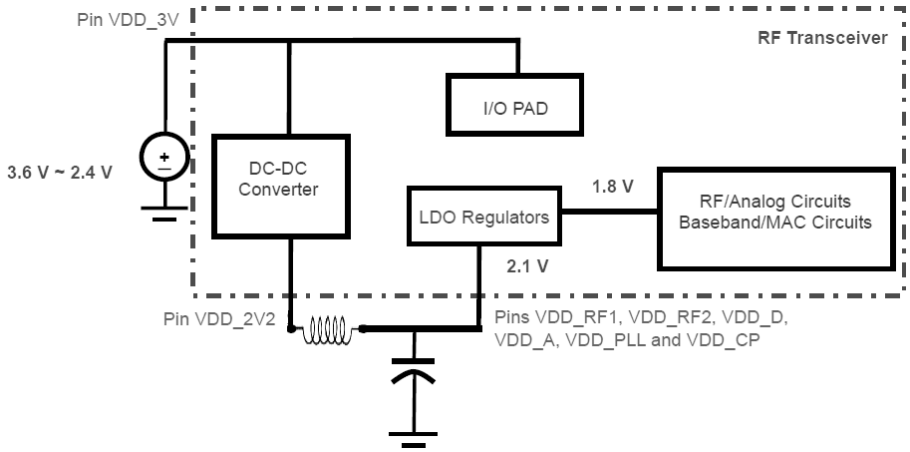
RF Transceiver Interfacing Block

The interfacing block mainly includes three parts named SPI interface, GPIO and Interrupt signal. Each of them is described as followings.

SPI Interface

The MCU communicates with the RF Transceiver via an internal SPI interface to read/write the control registers and FIFOs. The SPI interface connected to the MCU SPI master in the RF Transceiver has the following features:

- A 4-line slave SPI interface composed of: SEN (SPI enable), SCLK (SPI Clock), SI (Serial Data Input) and SO (Serial Data Output).
- Most significant bit (MSB) of all addresses and data transfers on the SPI interface is done first.



Block Diagram of Voltage Regulators with DC-DC Converter On/Bypass

- SPI Addressing Format

MSB of addressing frame indicates the addressing mode of the packet. The length of address field is 6 or 10 bits for short and long addressing mode respectively. Bit 0 is a one-bit read/write indicator.

Short Addressing Format

Bit 7	Bit 6 ~ 1	Bit 0
0	0x00 ~ 0x3F	Read: 0 Write: 1

Long Addressing Format

Bit 11	Bit 10 ~ 1	Bit 0
1	0x000 ~ 0x33F	Read: 0 Write: 1

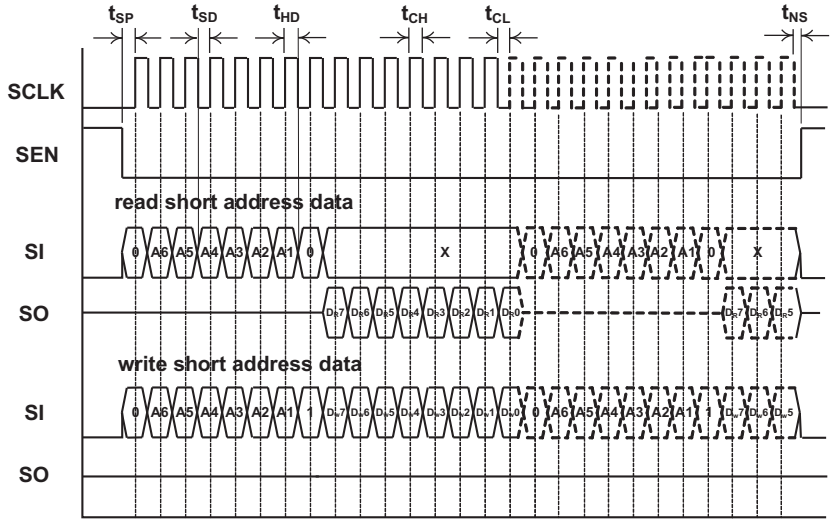
SPI Addressing Format

- SPI Characteristics

Parameter	Symbol	Min.	Max.	Unit	Conditions
SCLK, clock frequency	f_{SCLK}	—	5	MHz	
SCLK low pulse duration	t_{CL}	100	—	ns	The minimum time SCLK must be low.
SCLK high pulse duration	t_{CH}	100	—	ns	The minimum time SCLK must be high.
SEN setup time	t_{SP}	100	—	ns	The minimum time SEN must be low before the first positive edge of SCLK.
SEN hold time	t_{NS}	100	—	ns	The minimum time SEN must be held low after the last negative edge of SCLK.
SI setup	t_{SD}	25	—	ns	The minimum time data must be ready at SI, before the positive edge of SCLK
SI hold time	t_{HD}	25	—	ns	The minimum time data must be held at SI, after the positive edge of SCLK.
Rise time	t_{RISE}	—	25	ns	The maximum rise time for SCLK and SEN.
Fall time	t_{FALL}	—	25	ns	The maximum fall time for SCLK and SEN.

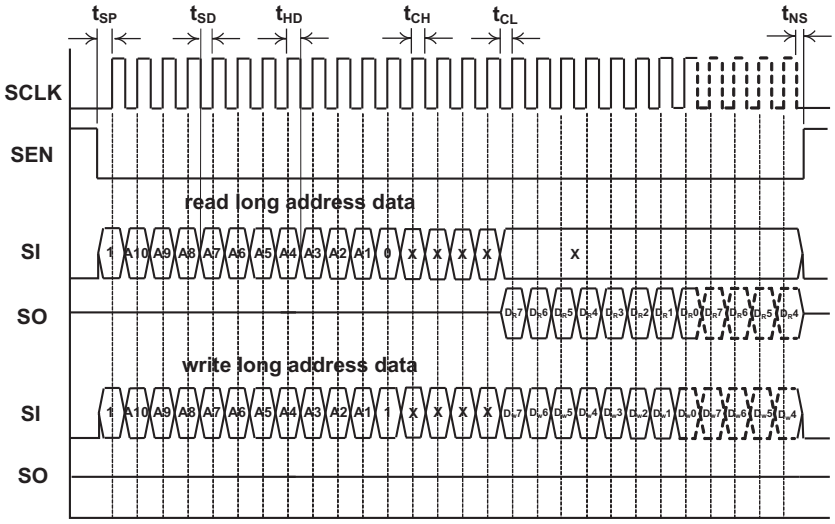
- SPI Timing Diagram

The following figures show the timing diagrams for the short and long addressing mode respectively. The MCU SPI master will initiate a read or write operation by asserting the interface enable signal SEN to low, toggling SCLK and sent the address field by SI. The interface enable signal SEN should be high when a transaction is completed.



Timing Diagram of Short Addressing Mode

The SPI burst mode is provided for the access of long address memory space on a continuous basis. If SEN does not go high after the 8-bit write data and the SCLK continuously toggles, the followed 8-bit write data is written to the next address field. Same for the read access, the data of the next address will be read. The SPI burst mode is only available for the long-address mode.



Timing Diagram of Long Addressing Mode

GPIO

The RF Transceiver has 3 digital GPIO pins. Each GPIO pins can be configured as input or output by LREG0x23D respectively. When being configured as an output pad, the driving capability is 4mA for GPIO0 and 1mA for GPIO1 and GPIO2. The status of these pins can be configured or read by LREG0x23E.

To benefit wide rang applications, GPIO0, GPIO1 and GPIO2 can be configured to control the external Power Amplifier (P.A.) and RF switch according to the current RF state automatically. Please refer to the following section named ″External Power Amplifier Configuration″ for details.

Interrupt Signal

The RF Transceiver provides an interrupt output pin named INT and the polarity of the interrupt signal is selectable. The RF Transceiver issues interrupts to the MCU host on three possible events. If one of the three events happens, the RF Transceiver sets the corresponding status bit in SREG0x31. If the corresponding interrupt mask in SREG0x32 is clear (i.e. equals 0), an interrupt will be issued on the interrupt output pin INT. If the corresponding interrupt mask is set to 1 (masked), no interrupt will be issued, but the status is still present. Whenever the SREG0x31 register is read, the interrupt and the status are cleared. The three interrupt events are described as below:

- Wake-up Alert Interrupt (WAKEIF): Each time a wake-up event happens the RF Transceiver issues the interrupt event.
- Packet Received Interrupt (RXIF): This interrupt is issued when an available packet is received in the RXFIFO. An available packet means that it passes a RXMAC filter, which includes frame type identifying, address filtering and FCS check.
- TX FIFO Release Interrupt (TXNIF): This interrupt can be issued in two possible conditions. The two conditions are when a packet in TXFIFO is triggered and sent successfully, or when a packet is triggered and the retransmission is timed out.

RF Transceiver Application Guide

Some typical applications are described in this section to help user gains more understanding of the operation of the RF Transceiver.

RF Transceiver Hardware Connection

A typical application connection is shown in Application Circuit section. The MCU host serves as a master role, and the RF Transceiver serves as a slave role. For more information, refer to the Application Circuit section.

RF Transceiver Initialization

After the RF Transceiver is powered up, some registers need to be configured before the data transmission or reception. The procedure is described as below.

- Procedure List

Parameter	Symbol	Min.	Max.	Unit	Conditions
SREG	0x26	GATECLK	Enable SPI sync function	20	
SREG	0x17	PACON1	Increase PAON time	08	
SREG	0x18	FIFOEN	Increase TXON time	94	
SREG	0x2E	TXPEMISP	VCO calibration period	95	
LREG	0x200	RFCTL0	RF optimized control	01	
LREG	0x201	RFCTL1	RF optimized control	02	
LREG	0x202	RFCTL2	RF optimized control	E0	
LREG	0x204	RFCTL4	RF optimized control	06	
LREG	0x206	RFCTL6	RF optimized control	C0	1M bps
LREG	0x207	RFCTL7	RF optimized control	F0	1M bps
LREG	0x208	RFCTL8	RF optimized control	8C	
LREG	0x23D	GPIODIR	For Setting GPIO to Output	00	
LREG	0x250	RFCTL50	RF optimized control	07	DC-DC OFF
LREG	0x251	RFCTL51	RF optimized control	C0	
LREG	0x252	RFCTL52	RF optimized control	01	
LREG	0x259	RFCTL59	RF optimized control	00	
LREG	0x273	RFCTL73	RF optimized control	40	
LREG	0x274	RFCTL74	RF optimized control	C6	DC-DC OFF
LREG	0x275	RFCTL75	RF optimized control	13	
LREG	0x276	RFCTL76	RF optimized control	07	
SREG	0x32	INTMSK	Enable all interrupt	00	
SREG	0x2A	SOFTRST	Baseband Reset	02	
SREG	0x36	RFCTL	RF Reset	04	Reset RF State Machine
SREG	0x36	RFCTL	RF Reset	00	Release RF State Machine

- Registers associated with Initialization

Addr.	File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR
0x17	PACON	r	r	r	PAONTS3	PAONTS2	PAONTS1	PAONTS0	r	0000 0010
0x18	TXCON	r	r	TXONTS3	TXONTS2	TXONTS1	TXONTS0	r	r	1000 1000
0x26	GATECLK	r	r	SPISYNC	r	r	ENTXM	r	r	0000 0000
0x2A	SOFTRST	r	r	r	r	r	r	RSTBB	RSTMAC	0000 0000
0x2E	TXPEMISP	TXPET3	TXPET2	TXPET1	TXPET0	r	r	r	r	0111 0101
0x32	INTMSK	r	WAKEMSK	r	r	RXMSK	r	r	TXNMSK	1111 1111
0x36	RFCTL	r	r	r	WAKECNT8	WAKECNT7	RFRST	r	r	0000 0000
0x200	RFCTRL0	CHANNEL3	CHANNEL2	CHANNEL1	CHANNEL0	r	r	r	r	0000 0001
0x201	RFCTRL1	r	r	r	r	r	r	VCORX1	VCORX0	0000 0001
0x202	RFCTRL2	r	RXFC0-1	RXFC0-0	r	r	r	r	r	1000 0100
0x204	RFCTRL4	r	r	r	r	r	RXFCO	RXD2CO1	RXD2CO0	0000 0000
0x206	RFCTRL6	TXFBW1	TXFBW0	32MXCO1	32MXCO0	BATEN	r	r	r	1111 0000
0x207	RFCTRL7	r	r	r	RXFC2	r	r	r	r	0000 0000
0x208	RFCTRL8	r	TXD2CO0	r	r	r	r	r	r	0000 1100
0x23D	GPIODIR	r	r	GDIRCTRL2	GDIRCTRL1	GDIRCTRL0	GPIO2DIR	GPIO1DIR	GPIO0DIR	0011 1111
0x250	RFCTRL50	r	r	r	DCPOC	DCOPC3	DCOPC2	DCOPC1	DCOPC0	0000 0000
0x251	RFCTRL51	DCOPC5	DCOPC4	r	r	r	r	r	r	0000 0000
0x252	RFCTRL52	SLCTRL6	SLCTRL5	SLCTRL4	SLCTRL3	SLCTRL2	SLCTRL1	SLCTRL0	32MXCTRL	1111 1111
0x259	RFCTRL59	r	r	r	r	r	r	r	PLLOPT3	0000 0001
0x273	RFCTRL73	VCOTXOPT1	VCOTXOPT0	r	r	PLLOPT2	PLLOPT1	PLLOPT0	r	0000 0000
0x274	RFCTRL74	PACEN0	PACTRL0-2	PACTRL0-1	PACTRL0-0	PACEN1	PACTRL1-2	PACTRL1-1	PACTRL1-0	1100 1010
0x275	RFCTRL75	r	r	r	r	SCLKOPT3	SCLKOPT2	SCLKOPT1	SCLKOPT0	0001 0101
0x276	RFCTRL76	r	r	r	r	r	SCLKOPT6	SCLKOPT5	SCLKOPT4	0000 0001

Change RF Channel Procedure

The RF Transceiver operates in 2.4GHz ISM band. The operating frequency is divided into 16 channels. The procedure to change the channels is described as below.

- Set the RF channel. Users can select one of the channels by configuring either LREG0x200 or LREG0x254.
- Turn on the TX MAC gated clock by setting SREG0x26 [2] to 1. To avoid an incomplete acknowledgment frame transmission happen during RF state machine reset period.
- Reset RF Transceiver state machine by setting SREG0x36 [2] to 1 and then set SREG0x36 [2] back to 0.
- After RF Transceiver reset, delay for a while to ensure the acknowledgment frame, if any, is successfully transmitted.
  250 kbps mode: delay 550μs
  1M bps mode: delay 300μs
- To disable the TX MAC gated clock by setting SREG26 [2] to 0.

Registers associated with Change Channel Procedure.

Addr.	File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR
0x26	GATECLK	r	r	SPISYNC	r	r	ENTXM	r	r	0000 0000
0x36	RFCTL	r	r	r	WAKECNT8	WAKECNT7	RFRST	r	r	0000 0000
0x200	RFCTRL0	CHANNEL3	CHANNEL2	CHANNEL1	CHANNEL0	r	r	r	r	0000 0001
0x254	RFCTRL54	1MCSEN	1MFRCH6	1MFRCH5	1MFRCH4	1MFRCH3	1MFRCH2	1MFRCH1	1MFRCH0	0000 0000

RF Transceiver Interrupt Configuration

The RF Transceiver issues a hardware interrupt at the internally connected interrupt signal line named INT to the MCU host. There are two related registers that need to be set correctly. All the interrupts are masked (disabled) by default. The interrupt mask should be removed by setting SREG0x32 in advance. The interrupt is by default sent to the MCU host as a falling edge signal after mask removed. The polarity can be configured by LREG0x211. The interrupt status can be read from SREG0x31 when it is triggered.

Registers associated with Interrupt Configuration

Addr.	File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR
0x31	ISRSTS	r	WAKEIF	r	r	RXIF	r	r	TXNIF	0000 0000
0x32	INTMSK	r	WAKEMSK	r	r	RXMSK	r	r	TXNMSK	1111 1111
0x211	IRQCTRL	r	r	r	r	r	r	IRQPOL	r	0000 0000

RF Transceiver External Power Amplifier Configuration

To enable the Power Amplifier (P.A.), users can set LREG0x22F [2:0] value to 0x001B. This register setting integrates the P.A. enable and the RF Switch Control (TX branch, RX branch) by utilizing GPIO0, GPIO1 and GPIO2. If the RF Transceiver is in TX mode, the GPIO0 (external P.A. enable) and GPIO1 (TX branch enable) will be pulled HIGH, and GPIO2 (RX branch enable) will be pulled LOW. If the RF Transceiver is in RX mode, the GPIO0 and GPIO1 will be pulled LOW, and GPIO2 will be pulled HIGH. The status of GPIO pins are automatically changed corresponding to TX/RX mode of the RF Transceiver.

- TX mode: [GPIO0, GPIO1, GPIO2] = [HIGH, HIGH, LOW]
- RX mode: [GPIO0, GPIO1, GPIO2] = [LOW, HIGH, HIGH]

Registers associated with External Power Amplifier Configuration

Addr.	File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR
0x22F	TESTMODE	MSPI	r	r	r	r	TESTMODE2	TESTMODE1	TESTMODE0	0010 1000

RF Transceiver Turbo Mode Configuration

The RF Transceiver provides 1M bps Turbo mode to transmit and receive data at a higher data rate. Turbo mode provides an added capability for applications which require more bandwidth. The application circuits need not any modification for Turbo mode.

To use the RF Transceiver in 250k and 1M bps, the following registers need to be configured as below.

Address Mode	Addr.	Register Name	Descriptions	Value (hex)	
				250k	1M
LREG	0x206	RFCTL6	RF optimized control	0x00	0xC0
LREG	0x207	RFCTL7	RF optimized control	0xE0	0xF0
SREG	0x38	BBREG0	Enable Normal/Turbo mode	0x80	0x81
SREG	0x2A	SOFTRST	Baseband Reset	0x02	

Registers associated with External Power Amplifier Configuration

Addr.	File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR
0x2A	SOFTRST	r	r	r	r	r	r	RSTBB	RSTMAC	0000 0000
0x38	BBREG0	r	r	r	r	r	r	r	TURBO	1000 0001
0x206	RFCTRL6	TXFBW1	TXFBW0	32MXCO1	32MXCO0	BATEN	r	r	r	1111 0000
0x207	RFCTRL7	r	r	r	RXFC2	r	r	r	r	0000 0000

Typical RF Transceiver TX Operation

The TXMAC inside the RF Transceiver will automatically generate the preamble, Start-of-Frame Delimiter and the FCS when transmitting. The MCU host must write all other frame fields into TXFIFO for TX operation. To send a packet in TX FIFO, there are several steps to follow:

Fill necessary data in TXFIFO. The format of TXFIFO is as follows:

- TXFIFO Address

0x001			0x0+N
1 Byte	4 Bytes	1 Byte	N Bytes
Frame Length	Destination Address	Frame Control	Payload

- Set Ackreq by SREG0x1B [2], if an acknowledgement / retransmission is required. The RF Transceiver automatically retransmits the packet till the number of the Max trial times specified in SREG1B [7:4] is reached, if there is no acknowledgement received.
- By triggering SREG0x1B [0], the TXMAC will send the packet immediately. This bit will be automatically cleared.
- Wait for the interrupt status shown in SREG0x31 [0]. If retransmission is not required, SREG0x31 [0] indicates the packet is successfully transmitted.
- Check SREG0x24 [0] to see if transmission is successful. If SREG0x24 [0] is equal to 0, it means that the transmission is successful and the ACK was received. The number of times of the retransmission can be read at SREG0x24 [7:4]. If SREG0x24 [0] is equal to 1, it means that the transmission failed and ACK was not received.

Registers associated with Typical TX Operation

Addr.	File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR
0x1B	TXTRIG	TXRTYN3	TXRTYN2	TXRTYN1	TXRTYN0	r	TXACKREQ	r	TXTRIG	0011 0000
0x24	TXSR	TXRETRY3	TXRETRY2	TXRETRY1	TXRETRY0	r	r	r	TXNS	0000 0000
0x31	ISRSTS	r	WAKEIF	r	r	RXIF	r	r	TXNIF	0000 0000

Typical RF Transceiver RX Operation

When a valid packet is received, an interrupt is issued at SREG0x31 [3]. The MCU host can read the whole packet inside the RXFIFO. The RXFIFO is flushed when the frame length field and the last byte of RXFIFO are read, or when the MCU host triggers a RX flush by SREG0x0D [0]. The format of RXFIFO is as follows:

- RXFIFO Address

0x300				0x307+N
1 Byte	4 Bytes	1 Byte	N Byte	1 Bytes
Frame Length	Destination Address	Frame Control	Payload	Frame Control

- Registers associated with Typical RX Operation

Addr.	File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR
0x0D	RXFLUSH	r	r	r	r	r	PTX	r	RXFLUSH	0110 0000
0x31	ISRSTS	r	WAKEIF	r	r	RXIF	r	r	TXNIF	0000 0000

RF Transceiver Power Saving Operation

Standby, Deep-Sleep and Power-Down Modes are designed for the RF Transceiver. It is only allowed to switch between power saving modes and active mode. The following settings are effective in active mode only.

Standby Mode

Shutdown RF/MAC/BB, while the voltage regulator, partial 32MHz clock and sleep clock remains active.

- Set LREG0x277 [5:4] to ″00″ to select for STANDBY Mode.
- Set LREG0x277 [3:2] to ″10″ for enable sleep voltage automatically controlled by internal circuit.
- Set LREG0x253 [4] to ″1″ to enable partial 32MHz clock.

Deep_Sleep Mode

All power is shutdown except the power to the digital circuits and sleep clock. Registers and FIFOs are retained.

- Set LREG0x277 [5:4] to ″00″ to select for DEEP_SLEEP Mode.
- Set LREG0x277 [3:2] to ″10″ for enable sleep voltage automatically controlled by internal circuit.

Power Down Mode

All power is shutdown. Registers and FIFOs data are not retained. Initialization is needed after the RF Transceiver back to active mode. Only the internal connected interrupt line named WAKE can wake the RF Transceiver up.

- Set LREG0x277 [5:4] to ″11″ to select for POWER DOWN Mode.
- Set LREG0x277 [3:2] to ″10″ for enable sleep voltage automatically controlled by internal circuit.
- Set LREG0x253 [6:5] to ″11″ to connect the FIFO power and digital circuit power to ground.

If the internal connected interrupt line named WAKE is going to be used to wake the RF Transceiver up, the configuration for WAKE line should be included. Refer to the following WAKE Line Wake-up Section for details. The on-chip DC-DC converter is not used for this device and then bypasses it by setting the DCPOC bit in LREG0x250 register to 0.

After the necessary settings mentioned above are configured, user can execute the following procedures to disable SPISYNC and place the RF Transceiver to the desired power saving mode.

- Set SREG0x26 [5] to ″0″ to disable SPISYNC.
- Set SREG0x35 [7] to ″1″ to place the RF Transceiver to power saving mode.

Registers associated with Power Saving Operation:

Addr.	File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR
0x26	GATECLK	r	r	SPISYNC	r	r	ENTXM	r	r	0000 0000
0x35	SLPACK	SLPACK	WAKECNT6	WAKECNT5	WAKECNT4	WAKECNT3	WAKECNT2	WAKECNT1	WAKECNT0	0000 0000
0x250	RFCTRL50	r	r	r	DCPOC	DCOPC3	DCOPC2	DCOPC1	DCOPC0	0000 0000
0x253	RFCTRL53	r	FIFOPS	DIGITALPS	P32MXE	PACEN2	PACTRL2-2	PACTRL2-1	PACTRL2-0	0000 0000
0x277	RFCTRL77	r	r	SLPSEL1	SLPSEL0	SLPVCTRL1	SLPVCTRL0	SLPVSEL1	SLPVSEL0	0000 1000

RF Transceiver Wake-up Operation

After entering into Power Saving Mode, the RF Transceiver could be waked up by the internal register trigger. One and only one method should be used for wake-up operation.

- Configure clock recovery time
  WAKECNT, used to calculate for recovery time of 32MHz clock of the RF Transceiver, should be set in advance. User shall follow the following two steps to configure WAKECNT.
  - Calculate the period of sleep clock
    Set LREG0x20B [4] to 1 and then keep polling LREG0x20B [7] until the value becomes 1. After the value of LREG0x20B [7] becomes 1, LREG0x20B [3:0], LREG0x20A, LREG0x209 form a 20-bit value C. Then the period of the sleep clock ($P_{sleepclock}$) can be calculated by the following equation:

    $$P_{sleepclock} = \frac{62.5xC}{16}(ns)$$

    If the sleep clock frequency is higher than the expected value, user can configure LREG0x220 [4:0] to slow down the clock rate. The new clock period $P_{sleepclock\_new}$ is obtained by the following equation:

    $P_{sleepclock\_new} = P_{sleepclock\_ori} \times 2^{LREG0x220[4:0]}$ (ns)
  - Configure WAKECNT to set the recovery time of 32MHz clock to 180μs
    Set WAKECNT, i.e. SREG0x36 [4:3] and SREG0x35 [6:0], to (1000*180) / $P_{sleepclock}$. For example, the period of the sleep clock, $P_{sleepclock}$, is 10000ns. Set SREG0x36 [4:3] and SREG0x35 [6:0]} to 0x12.

Register Trigger Wake-up

User can wake the RF Transceiver up from STANDBY and DEEP_SLEEP modes by simply setting SREG0x22 [7:6] to ″11″.

When the RF Transceiver is woken up by Register trigger, the following steps shall be executed to complete the operation:

- Wait the RF Transceiver issues a wake-up interrupt. The related wake-up interrupt flag is stored in SREG0x31 [6].
- Turn on SPISYNC function by setting SREG0x26 [5] to 1.
- Setting the LREG0x250 [4] to 1 to turn off the on-chip DC-DC converter.

Registers associated with Power Saving Operation:

Addr.	File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR
0x0D	RXFLUSH	r	r	r	r	r	PTX	r	RXFLUSH	0110 0000
0x22	WAKECTL	IMMWAKE	REGWAKE	r	r	r	r	r	r	0100 0000
0x26	GATECLK	r	r	SPISYNC	r	r	ENTXM	r	r	0000 0000
0x31	ISRSTS	r	WAKEIF	r	r	RXIF	r	r	TXNIF	0000 0000
0x35	SLPACK	SLPACK	WAKECNT6	WAKECNT5	WAKECNT4	WAKECNT3	WAKECNT2	WAKECNT1	WAKECNT0	0000 0000
0x36	RFCTL	r	r	r	WAKECNT8	WAKECNT7	RFRST	r	r	0000 0000
0x209	SLPCAL_0	SLPCAL7	SLPCAL6	SLPCAL5	SLPCAL4	SLPCAL3	SLPCAL2	SLPCAL1	SLPCAL0	0000 0000
0x20A	SLPCAL_1	SLPCAL15	SLPCAL14	SLPCAL13	SLPCAL12	SLPCAL11	SLPCAL10	SLPCAL9	SLPCAL8	0000 0000
0x20B	SLPCAL_2	SLPCALRDY	r	r	SLPCALEN	SLPCAL19	SLPCAL18	SLPCAL17	SLPCAL16	0000 0000
0x250	RFCTRL50	r	r	r	DCPOC	DCOPC3	DCOPC2	DCOPC1	DCOPC0	-0000 0000

Primary RF Transceiver TX Operation

Users activate the primary TX mode by setting SREG0x0D [2] to 1. After changing the SREG0x0D [2] value, users have to reset the RF and let RF state machine go to primary TX mode correctly. If primary TX mode is enabled, the RF Transceiver will enter power waving mode after ant packet transmits. If primary TX mode is not enabled, the RF Transceiver will switch to RX mode after any packet transmits. If ACK response is needed and primary TX mode is enabled, the RF Transceiver will enter the Power Saving Mode after ACK frame received. If no ACK frame received, the RF Transceiver will not enter the Power Saving Mode until the max time to wait for an acknowledgement frame by setting SREG0x12 [6:0].

Registers associated with Power Saving Operation:

Addr.	File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR
0x0D	RXFLUSH	r	r	r	r	r	PTX	r	RXFLUSH	0110 0000
0x12	ACKTO	r	MATOP6	MATOP5	MATOP4	MATOP3	MATOP2	MATOP1	MATOP0	0011 1001

RF Transceiver Battery Monitor Operation

The RF Transceiver has Battery Monitor function and the procedure to enable the Battery Monitor function is described as below.

- Set the battery monitor threshold value at LREG0x205 [7:4].
- Enable the battery monitor by setting the LREG0x206 [3] to the value 1.
- Read the battery-low indicator at SREG0x34 [5]. If this bit is set, it means that the supply voltage is lower than the battery monitor threshold specified by LREG0x205 [7:4].

Registers associated with Power Saving Operation:

Addr.	File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR
0x34	BATRXF	r	r	BATIND	r	r	r	RDFF1	RXFIFO2	0000 0000
0x205	RFCTRL5	BATTH3	BATTH2	BATTH1	BATTH0	r	r	r	r	0000 0000
0x206	RFCTRL6	TXFBW1	TXFBW0	32MXCO1	32MXCO0	BATEN	r	r	r	1111 0000

RF Transceiver Register Definitions

The Memory of the RF Transceiver is categorized into two kinds of addressing mode, known as Short Addressing Registers and Long Addressing Registers. Each of the Register definition is described in the following sections.

Legends of RF Transceiver Register Types

Register Type	Description
R/W	Read/Write register
WT	Write 1 to trigger register, automatically cleared by hardware
RC	Read to clear register
R	Read-only register
R/W1C	Read/Write ″1″ to clear register

RF Transceiver Short Addressing Registers (SREG0x00~SREG0x3F)

0x00	RXMCR	0x12	ACKTO	0x22	WAKECTL	0x30	RXSR
0x03	AUINFL	0x17	PACON	0x24	TXSR	0x31	ISRSTS
0x04	AUINFH	0x18	TXCON	0x26	GATECLK	0x32	INTMSK
0x05	DADR_0	0x1B	TXTRIG	0x2A	SOFTRST	0x34	BATRXF
0x06	DADR_1		—	0x2E	TXPEMISP	0x35	SLPACK
0x07	DADR_2		—		—	0x36	RFCTL
0x08	DADR_3		—		—	0x38	BBREG0
0x0D	RXFLUSH		—		—		—

- SREG0x00 - RXMCR: Receive MAC Control Register

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

Bit 7~6 Reserved: maintain as ″0b00″

Bit 5 **NOACKRSP**: Automatic Acknowledgement Response
0: (default) enables automatic acknowledgement response
1: disables automatic acknowledgement response

Bit 4~0 Reserved: maintain as ″0b00000″

- SREG0x03 - AUINFL: Acknowledgement User Information Low Byte

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	AUINF7	AUINF6	AUINF5	AUINF4	AUINF3	AUINF2	AUINF1	AUINF0
Type	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 **AUINF [7:0]**: 16-bit User Information of Acknowledgement frame Low Byte.

- SREG0x04 - AUINFH: Acknowledgement User Information High Byte

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	AUINF15	AUINF14	AUINF13	AUINF12	AUINF11	AUINF10	AUINF9	AUINF8
Type	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 **AUINF [15:8]**: 16-bit User Information of Acknowledgement frame High Byte.

- SREG0x05 - DADR_0: Device Address 0

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	DADR7	DADR6	DADR5	DADR4	DADR3	DADR2	DADR1	DADR0
Type	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 **DADR [7:0]**: 32-bit Address of the RF Transceiver.

- SREG0x06 - DADR_1: Device Address 1

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	DADR15	DADR14	DADR13	DADR12	DADR11	DADR10	DADR9	DADR8
Type	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 **DADR [15:8]**: 32-bit Address of the RF Transceiver

- SREG0x07 - DADR_2: Device Address 2

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	DADR23	DADR22	DADR21	DADR20	DADR19	DADR18	DADR17	DADR16
Type	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 **DADR [23:16]**: 32-bit Address of the RF Transceiver.

- SREG0x08 - DADR_3: Device Address 3

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	DADR31	DADR30	DADR29	DADR28	DADR27	DADR26	DADR25	DADR24
Type	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 **DADR [31:24]**: 32-bit Address of the RF Transceiver

- SREG0x0D - RXFLUSH: Receive FIFO Flush

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	—	—	—	—	PTX	—	RXFLUSH
Type	R	R	R	R	R	R/W	R	WT
POR	0	1	1	0	0	0	0	0

Bit 7 Reserved: maintain as ″0b0″

Bit 6-5 Reserved: maintain as ″0b11″

Bit 4-3 Reserved: maintain as ″0b00″

Bit 2 **PTX**: Primary TX mode enable (1)
1: primary TX mode
0: primary RX mode (default)
Note: RF reset, SREG0x36 [2], is needed after switching between PTX and PRX modes

Bit 1 Reserved: maintain as ″0b0″

Bit 0 **RXFLUSH**: Flush the RX FIFO
1: Flush RX FIFO. RX FIFO data is not modified. If Ping-pong FIFO is enabled (SREG0x34 [0] =1), both FIFOs are flushed at the same time. Bit is automatically cleared to ″0″ by hardware.

- SREG0x12 - ACKTO: Acknowledgement Timeout Period

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	MATOP6	MATOP5	MATOP4	MATOP3	MATOP2	MATOP1	MATOP0
Type	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	1	1	1	0	0	1

Bit 7 Reserved: maintain as "0b0"

Bit 6~0 **MATOP [6:0]**: Maximum Acknowledgement Timeout Period
0000000: 0 (default)
0000001: 1
0000010: 2
:
0111001: 57
:
1111111: 127

- SREG0x17 - PACON: Power Amplifier Control Register

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	—	—	PAONTS3	PAONTS2	PAONTS1	PAONTS0	—
Type	R	R	R	R/W	R/W	R/W	R/W	R
POR	0	0	0	0	0	0	1	0

Bit 7~5 Reserved: maintain as "0b000"

Bit 4~1 **PAONTS [3:0]**: Power Amplifier Settling Time to begin packet transmission.
0001: (default)
0100: (optimized - do not change)

Bit 0 Reserved: maintain as "0b0"

- SREG0x18 - TXCON: Transmitter Control Register

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	—	TXONTS3	TXONTS2	TXONTS1	TXONTS0	—	—
Type	R	R	R/W	R/W	R/W	R/W	R	R
POR	0	0	0	0	1	0	0	0

Bit 7~6 Reserved: maintain as "0b00"

Bit 5~2 **TXONTS [3:0]**: Transmitter Settling Time to begin packet transmission
0010: (default)
0101: (optimized - do not change)

Bit 1~0 Reserved: maintain as "0b00"

- SREG0x1B - TXTRIG: Transmit FIFO Control Register

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	TXRTYN3	TXRTYN2	TXRTYN1	TXRTYN0	—	TXACKREQ	—	TXTRIG
Type	R/W	R/W	R/W	R/W	R	R/W	R	WT
POR	0	0	1	1	0	0	0	0

Bit 7-4 **TXRTYN [3:0]**: Maximum TX Retry Times
0000: 0
:
0011: 3 (default)
:
0101: 15

Bit 3 Reserved: maintain as ″0b0″

Bit 2 **TXACKREQ**: TX FIFO Acknowledge Request bit
1: acknowledgement packet requested
0: no acknowledgement packet requested (default)
Transmit a packet with Acknowledgement request. If Acknowledgement is not received, the RF Transceiver retransmits up to xx times.

Bit 1 Reserved: maintain as ″0b0″

Bit 0 **TXTRIG**: Transmit Trigger bit
1: Transmit Frame in TX FIFO. Bit is automatically cleared to ″0″ by hardware.

- SREG0x22 - WAKECTL: Wake-up Control Register

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	IMMWAKE	REGWAKE	—	—	—	—	—	—
Type	R/W	WT	R	R	R	R	R	R
POR	0	1	0	0	0	0	0	0

Bit 7 **IMMWAKE**: Immediate Wake-up Mode Enable bit
1: enable immediate Wake-up Mode
0: disable immediate Wake-up Mode (default)

Bit 6 **REGWAKE**: Register Triggered Wake-up bit
1: To wake the RF Transceiver up. Bit is automatically to ″0″ by hardware.

Bit 5~0 Reserved: maintain as ″0b000000″

- SREG0x24 - TXSR: TX Status Register

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	TXRETRY3	TXRETRY2	TXRETRY1	TXRETRY0	—	—	—	TXNX
Type	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7-4 **TXRETRY [3:0]**: TXFIFO Retry Times
0000: 0 (default)
:
0101: 15
TXRETRY indicates the maximum number of retries of the most recent TXFIFO transmission.

Bit 3-1 Reserved: maintain as ″0b000″

Bit 0 **TXNX**: TXFIFO Normal Release Status
1: Fail, retry count exceed
0: Succeeded (default)

- SREG0x26 - GATECLK: Gated Clock control Register

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	—	SPISYNC	—	—	ENTRM	—	—
Type	R	R	R/W	R	R	R/W	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~6 Reserved: maintain as ″0b00″

Bit 5 **SPISYNC**: SPI Interface Synchronization
1: enable (optimized - do not change)
0: disable (default)

Bit 4~3 Reserved: maintain as ″0b00″

Bit 2 **ENTRM**: TX MAC Clock Enable Control
1: enable
0: disable (default)

Bit 1~0 Reserved: maintain as ″0b00″

- SREG0x2A - SOFTRST: Software Reset control Register

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	—	—	—	—	—	RSTBB	RSTMAC
Type	R	R	R	R	R	R	WT	WT
POR	0	0	0	0	0	0	0	0

Bit 7-2 Reserved: Maintain as ″0b000000″

Bit 1 **RSTBB**: Baseband Reset
1: reset baseband circuitry. Initialization is not needed after RSTBB reset. Bit is automatically cleared to 0 by hardware.

Bit 0 **RSTMAC**: MAC and Short/Long Addressing Registers Reset.
1: Reset MAC circuitry and Short/Long Addressing Registers. Initialization is needed after RSTMAC reset. Bit is automatically cleared to ″0″ by hardware.

- SREG0x2E - TXPEMISP: Transmit Parameters Register

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	TXPET3	TXPET2	TXPET1	TXPET0	—	—	—	—
Type	R/W	R/W	R/W	R/W	R	R	R	R
POR	0	1	1	1	0	1	0	1

Bit 7~4 **TXPET [3:0]**: TXFIFO Retry Times.
0111: (default)
1001: (optimized - do not change)

Bit 3~0 Reserved: maintain as ″0b0101″

- SREG0x30 - RXSR: RX MAC Status Register

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	RXFFFULL	WRFF1	—	RXFFOVFL	RXCRCERR	—	—	—
Type	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7 **RXFFFULL**: RX FIFO Full
1: RX FIFO is not available for data receiving
0: RX FIFO is available for data receiving (default)

Bit 6 **WRFF1**: RX FIFO Status
1: Packet is ready in RX FIFO 1
0: Packet is ready in RX FIFO 0 (default)

Bit 5 Reserved: maintain as ″0b0″

Bit 4 **RXFFOVFL**: RX FIFO Overflow
1: RX FIFO overflows
0: (default) RX FIFO not overflow

Bit 3 **RXCRCERR**: RX CRC Error
1: RX CRC error
0: RX CRC is correct (default)

Bit 2~0 Reserved: maintain as ″0b000″

- SREG0x31 - ISRSTS: Interrupt Status Register

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	WAKEIF	—	—	RXIF	—	—	TXNIF
Type	R	RC	R	R	RC	R	R	RC
POR	0	0	0	0	0	0	0	0

Bit 7 Reserved: maintain as ″0b0″

Bit 6 **WAKEIF**: Wake-up Alert Interrupt
1: A wake-up interrupt occurred
0: No wake-up alert interrupt occurred (default)
This bit is cleared to 0 when the register is read.

Bit 5-4 Reserved: maintain as ″0b00″

Bit 3 **RXIF**: RX FIFO Reception Interrupt
1: A RX FIFO reception interrupt occurred
0: No RX FIFO reception interrupt occurred (default)
This bit is cleared to 0 when the register is read.

Bit 2-1 Reserved: maintain as ″0b00″

Bit 0 **TXNIF**: TX FIFO Normal Transmission Interrupt
1: TX FIFO normal transmission interrupt occurred
0: No TX FIFO normal transmission interrupt occurred (default)
This bit is cleared to ″0″ when the register is read.

- SREG0x32 - INTMSK: Interrupt Mask control Register

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	WAKEMSK	—	—	RXMSK	—	—	TXNMSK
Type	R	R/W	R	R	R/W	R	R	R/W
POR	1	1	1	1	1	1	1	1

Bit 7 Reserved: maintain as "0b1"

Bit 6 **WAKEMSK**: Wake-up Alert Interrupt Mask
1: disable the wake-up interrupt (default)
0: enable the wake-up alert interrupt

Bit 5~4 Reserved: maintain as "0b11"

Bit 3 **RXMSK**: RX FIFO Reception Interrupt Mask
1: disable the RX FIFO reception interrupt (default)
0: enable the RX FIFO reception interrupt

Bit 2~1 Reserved: maintain as "0b11"

Bit 0 **TXNMSK**: TX FIFO Normal Transmission Interrupt Mask
1: disable the TX FIFO Normal Transmission interrupt (default)
0: enable the TX FIFO Normal Transmission interrupt

- SREG0x34 - BATRXF: Battery and RX FIFO control Register

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	—	BATIND	—	—	—	RDFF1	RXFIFO2
Type	R	R	R	R	R	R	R	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~6 Reserved: Maintain as "0b00"

Bit 5 **BATIND**: Battery Low Indicator
1: battery voltage is lower than the threshold voltage specified by the LREG0x205 [7:4].
0: battery voltage is higher than the threshold voltage specified by the LREG0x205 [7:4] (default)

Bit 4~2 Reserved: Maintain as "0b000"

Bit 1 **RDFF1**: RX FIFO Selected to Read
1: read data from RX FIFO 1
0: read data from RX FIFO 0 (default)

Bit 0 **RXFIFO2**: RX Ping-Pong FIFO Enable Control
1: enable the RX Ping-Pong FIFOs
0: disable the RX Ping-Pong FIFOs (default)

- SREG0x35 - SLPACK: Sleep Acknowledgement and Wake-up Counter Register

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	SLPACK	WAKECNT6	WAKECNT5	WAKECNT4	WAKECNT3	WAKECNT2	WAKECNT1	WAKECNT0
Type	WT	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 **SLPACK**: Sleep Acknowledgement
Place the RF Transceiver to Power Saving Mode. bit is automatically cleared to 0 by hardware.

Bit 6~0 **WAKECNT [6:0]**: System Clock Recovery Time
0000000: (default).
WAKECNT is a 9-bit value. The WAKECNT [8:7] bits are located in SREG0x36 [4:3].

- SREG0x36 - RFCTL: RF Mode Control Register

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	—	—	WAKECNT8	WAKECNT7	RFRST	—	—
Type	R	R	R	R/W	R/W	R/W	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~5 Reserved: Maintain as ″0b000″

Bit 4~3 **WAKECNT [8:7]**: System Clock Recovery Time
00: (default).
WAKECNT is a 9-bit value. The WAKECNT [6:0] bits are located in SREG0x35 [6:0].

Bit 2 **RFRST**: RF State Machine Reset.
1: Hold RF state machine in Reset state
0: Normal operation of RF state machine (default)
Perform RF reset by setting RFRST to ″1″ and then setting RFRST to ″0″. Delay at least 192μs after performing to allow RF circuitry to calibrate.

Bit 1~0 Reserved: Maintain as ″0b00″

- SREG0x38 - BBREG0: Baseband Register

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

Bit 7~1 Reserved: Maintain as ″0b1000000″

Bit 0 **TURBO**: Turbo Mode Select
1: 1M bps Turbo Mode (default)
0: 250k bps Normal Mode

RF Transceiver Long Addressing Registers (LREG0x200~LREG0x27F)

0x200	RFCTRL0	0x211	IRQCTL	0x250	RFCTRL50	0x273	RFCTRL73
0x201	RFCTRL1	0x22F	TESTMODE	0x251	RFCTRL51	0x274	RFCTRL74
0x202	RFCTRL2	0x23C	—	0x252	RFCTRL52	0x275	RFCTRL75
0x203	_RFCTRL3	0x23D	GPIODIR	0x253	RFCTRL53	0x276	RFCTRL76
0x204	RFCTRL4	0x23E	GPIO	0x254	RFCTRL54	0x277	RFCTRL77
0x205	RFCTRL5		—	0x259	RFCTRL59		—
0x206	RFCTRL6		—		—		—
0x207	RFCTRL7		—		—		—
0x208	RFCTRL8		—		—		—
0x209	SLPCAL_0		—		—		—
0x20A	SLPCAL_1		—		—		—
0x20B	SLPCAL_2		—		—		—

- LREG0x200 - RFCTRL0: RF Control Register 0

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

Bit 7~4 **CHANNEL [3:0]**: Channel Number. IEEE 802.15.4 2.4GHz band channels (11~26)
0000: channel 11, 2405MHz (default)
0001: channel 12, 2410MHz
0010: channel 13, 2415MHz
0011: channel 14, 2420MHz
0100: channel 15, 2425MHz
0101: channel 16, 2430MHz
0110: channel 17, 2435MHz
0111: channel 18, 2440MHz
1000: channel 19, 2445MHz
1001: channel 20, 2450MHz
1010: channel 21, 2455MHz
1011: channel 22, 2460MHz
1100: channel 23, 2465MHz
1101: channel 24, 2470MHz
1110: channel 25, 2475MHz
1111: channel 26, 2480MHz

Bit 3~0 Reserved: Maintain as "0b0001"

- LREG0x201 - RFCTRL1: RF Control Register 1

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	—	—	—	—	—	VCORX1	VCORX0
Type	R	R	R	R	R	R	R/W	R/W
POR	0	0	0	0	0	0	0	1

Bit 7~2 Reserved: Maintain as "0b000000"

Bit 1~0 **VCORX [1:0]**: RX VC
01: (default)
10: (optimized - do not change)

- LREG0x202 - RFCTRL2: RF Control Register 2

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	RXFC0-1	RXFC0-0	—	—	—	—	—
Type	R	R/W	R/W	R	R	R	R	R
POR	1	0	0	0	0	0	0	0

Bit 7 Reserved: Maintain as "0b1"

Bit 6~5 **RXFC0 [1:0]**: RX Filter Control 0.
00: (default)
11: (optimized - do not change)

Bit 4~0 Reserved: Maintain as "0b00000"

- LREG0x203 - RFCTRL3: RF Control Register 3

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	TXGB4	TXGB3	TXGB2	TXGB1	TXGB0	—	—	—
Type	R/W	R/W	R/W	R/W	R/W	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7-3 **TXGB [4:0]**: TX Gain Control in dB. Gain step is monotonic and nonlinear with gain resolution of 0.1 ~ 0.5 dB. Gain Resolution is variable between chips.

00000: 0 dBm (default)
00001: -0.1 dBm
00010: -0.3 dBm
00011: -0.6 dBm
00100: -0.9 dBm
00101: -1.1 dBm
00110: -1.2 dBm
00111: -1.3 dBm
01000: -1.4 dBm
01001: -1.5 dBm
01010: -1.7 dBm
01011: -2.0 dBm
01100: -2.2 dBm
01101: -2.4 dBm
01110: -2.6 dBm
01111: -2.8 dBm
10000: -3.1 dBm
10001: -3.3 dBm
10010: -3.6 dBm
10011: -3.8 dBm
10100: -4.2 dBm
10101: -4.4 dBm
10110: -4.7 dBm
10111: -5.0 dBm
11000: -5.3 dBm
11001: -5.7 dBm
11010: -6.2 dBm
11011: -6.5 dBm
11100: -6.9 dBm
11101: -7.4 dBm
11110: -7.9 dBm
11111: -8.3 dBm

TX Output Power Configuration Summary table:

TX Output Power Register Control			
LREG0x253 [3:0]	**LREG0x274 [7:0]**	**LREG0x203 [7:3]**	**TX Output Power**
00	C6 for DC-DC OFF	00000	0 dBm
		00001	-0.1 dBm
		00010	-0.3 dBm
		00011	-0.6 dBm
		00100	-0.9 dBm
		00101	-1.1 dBm
		00110	-1.2 dBm
		00111	-1.3 dBm
		01000	-1.4 dBm
		01001	-1.5 dBm
		01010	-1.7 dBm
		01011	-2.0 dBm
		01100	-2.2 dBm
		01101	-2.4 dBm
		01110	-2.6 dBm
		01111	-2.8 dBm
		10000	-3.1 dBm
		10001	-3.3 dBm
		10010	-3.6 dBm
		10011	-3.8 dBm
		10100	-4.2 dBm
		10101	-4.4 dBm
		10110	-4.7 dBm
		10111	-5.0 dBm

TX Output Power Register Control			
LREG0x253 [3:0]	LREG0x274 [7:0]	LREG0x203 [7:3]	TX Output Power
00	C6 for DC-DC OFF	11000	-5.3 dBm
		11001	-5.7 dBm
		11010	-6.2 dBm
		11011	-6.5 dBm
		11100	-6.9 dBm
		11101	-7.4 dBm
		11110	-7.9 dBm
		11111	-8.3 dBm
0C	81	11111	-16 dBm
0C	09	11111	-24 dBm
09	01	11111	-32 dBm
08	01	11111	-40 dBm

Bit 2~0 Reserved: Maintain as ″0b000″

- LREG0x204 - RFCTRL4: RF Control Register 4

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	—	—	—	—	RXFCO	RXD2O1	RXD2O0
Type	R	R	R	R	R	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~3 Reserved: Maintain as ″0b00000″

Bit 2 **RXFCO**: RX Filter Calibration output
1: (optimized - do not change)
0: (default)

Bit 1~0 **RXD2O [1:0]**: RX Divide-by-2 option
00: (default)
10: (optimized - do not change)

- LREG0x205 - RFCTRL5: RF Control Register 5

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	BATTH3	BATTH2	BATTH1	BATTH0	—	—	—	—
Type	R/W	R/W	R/W	R/W	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~4 **BATTH [3:0]**: Battery Monitor Threshold.

0000: 1.8V (default)	1000: 2.6V
0001: 1.9V	1001: 2.7V
0010: 2.0V	1010: 2.8V
0011: 2.1V	1011: 2.9V
0100: 2.2V	1100: 3.0V
0101: 2.3V	1101: 3.3V
0110: 2.4V	1110: 3.4V
0111: 2.5V	1111: 3.6V

Bit 3~0 Reserved: Maintain as ″0b0000″

• LREG0x206 - RFCTRL6: RF Control Register 6

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	TXFBW1	TXFBW0	32MXCO1	32MXCO0	BATEN	—	—	—
Type	R/W	R/W	R/W	R/W	R/W	R	R	R
POR	1	1	1	1	0	0	0	0

Bit 7~6 **TXFBW [1:0]**: TX Filter
00: Optimized for 250k bps Normal Mode
11: Optimized for 1M bps Turbo Mode

Bit 5~4 **32MXCO [1:0]**: 32MHz Crystal Oscillator
00: (Optimized - do not change)
11: (default)

Bit 3 **BATEN**: Battery Monitor Enable
1: Enable
0: Disable (default)

Bit 2~0 Reserved: Maintain as ″0b000″

• LREG0x207 - RFCTRL7: RF Control Register 7

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	—	—	RXFC2	—	—	—	—
Type	R	R	R	R/W	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~5 Reserved: Maintain as ″0b000″

Bit 4 **RXFC2**: RX Filter Control 2
1: For 1M bps Turbo Mode
0: For 250k bps Normal Mode (default)

Bit 3~0 Reserved: Maintain as ″0b0000″

• LREG0x208 - RFCTRL8: RF Control Register 8

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	TXD2CO1	TXD2CO0	—	—	—	—	—	—
Type	R/W	R/W	R	R	R	R	R	R
POR	0	0	0	0	1	1	0	0

Bit 7~6 **TXD2CO [1:0]**: TX Divide-by-2 Option
00: (default)
10: (Optimized - do not change)

Bit 5~0 Reserved: Maintain as ″0b001100″

• LREG0x209 - SLPCAL_0: Sleep Clock Calibration 0

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	SLPCAL7	SLPCAL6	SLPCAL5	SLPCAL4	SLPCAL3	SLPCAL2	SLPCAL1	SLPCAL0
Type	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 **SLPCAL [7:0]**: Sleep Clock Calibration Counter bit 7~0
A 20-bit calibration counter which calibrates the sleep clock. SLPCAL [19:0] indicates the time period of 16 sleep clock cycles. The unit is 62.5ns, counted by the 16MHz.

- LREG0x20A - SLPCAL_1: Sleep Clock Calibration 1

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	SLPCAL15	SLPCAL14	SLPCAL13	SLPCAL12	SLPCAL11	SLPCAL10	SLPCAL9	SLPCAL8
Type	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 **SLPCAL [15:8]**: Sleep Clock Calibration Counter bit 15~8
A 20-bit calibration counter which calibrates the sleep clock. SLPCAL [19:0] indicates the time period of 16 sleep clock cycles. The unit is 62.5ns, counted by the 16MHz.

- LREG0x20B - SLPCAL_2: Sleep Clock Calibration 2

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	SLPCALRDY	—	—	SLPCALEN	SLPCAL19	SLPCAL18	SLPCAL17	SLPCAL16
Type	R	R	R	WT	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **SLPCALRDY**: Sleep Clock Calibration Ready
1: Sleep Clock Calibration counter is ready to be read.
0: Not Ready (default)

Bit 6-5 Reserved: Maintain as ″0b00″

Bit 4 **SLPCALEN**: Sleep Clock Calibration Enable
1: Starts the Sleep Clock Calibration counter. Bit is automatically cleared to ″0″ by hardware

Bit 3-0 **SLPCAL [19:16]**: Sleep Clock Calibration Counter bit 19~16
A 20-bit calibration counter which calibrates the sleep clock. SLPCAL [19:0] indicates the time period of 16 sleep clock cycles. The unit is 62.5ns, counted by the 16MHz.

- LREG0x211 - IRQCTRL: Interrupt Control Register

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

Bit 7~2 Reserved: Maintain as ″0b000000″

Bit 1 **IRQPOL**: Interrupt Edge Polarity
1: Rising Edge
0: Falling Edge (default)

Bit 0 Reserved: Maintain as ″0b0″

- LREG0x22F - TESTMODE: Test Mode Register

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	MSPI	—	—	—	—	TESTMODE2	TESTMODE1	TESTMODE0
Type	R/W	R	R	R	R	R/W	R/W	R/W
POR	0	0	1	0	1	0	0	0

Bit 7 **MSPI**: Multiple SPI Operation
1: Enable multiple SPI Operation, SO will be High-Z state when SPI inactive
0: Single SPI Operation, SO will be low when SPI inactive (default)

Bit 6-3 Reserved: Maintain as ″0b0101″

Bit 2-0 **TESTMODE [2:0]**: Special Operation
000: (default) Normal Operation
001: GPIO0, GPIO1 and GPIO2 are configured to control the external P.A., LNA and RF switch
101: Single-Tone test mode
Others: Undefined.

- LREG0x23D - GPIODIR: GPIO Pin Direction

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	—	GDIRCTRL2	GDIRCTRL1	GDIRCTRL0	GPIO2DIR	GPIO1DIR	GPIO0DIR
Type	R	R	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	1	1	1	1	1	1

Bit 7-6 Reserved: Maintain as ″0b00″

Bit 5-3 **GDIRCTRL [2:0]**: GPIO Direction Control
000: (Optimized - do not change)
111: (default)

Bit 2 **GPIO2DIR**: General Purpose I/O GPIO2 Direction
1: Input (default)
0: Output

Bit 1 **GPIO1DIR**: General Purpose I/O GPIO1 Direction
1: Input (default)
0: Output

Bit 0 **GPIO0DIR**: General Purpose I/O GPIO0 Direction
1: Input (default)
0: Output

- LREG0x23E - GPIO: GPIO

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	—	—	—	—	GPIO2	GPIO1	GPIO0
Type	R	R	R	R	R	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~3 Reserved: Maintain as ″0b00000″

Bit 2 **GPIO2**: Setting for output/Status for input of General Purpose I/O Pin GPIO2
0: (default)

Bit 1 **GPIO1**: Setting for output/Status for input of General Purpose I/O Pin GPIO1
0: (default)

Bit 0 **GPIO0**: Setting for output/Status for input of General Purpose I/O Pin GPIO0
0: (default)

- LREG0x250 - RFCTRL50: RF Control Register 50

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	—	—	DCPOC	DCOPC3	DCOPC2	DCOPC1	DCOPC0
Type	R	R	R	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7~5 Reserved: Maintain as ″0b000″

Bit 4 **DCPOC**: DC-DC Converter Power Control
1: Enable
0: Bypass (default)

Bit 3~0 **DCOPC [3:0]**: DC-DC Converter Optimization Control
0000: (default)
0111: (Optimized - do not change)

- LREG0x251 - RFCTRL51: RF Control Register 51

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	DCOPC5	DCOPC4	—	—	—	—	—	—
Type	R/W	R/W	R/W	R/W	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~6 **DCOPC [5:4]**: DC-DC Converter Optimization Control
00: (default)
11: (Optimized - do not change)

Bit 5~0 Reserved: Maintain as 0b000000

- LREG0x252 - RFCTRL52: RF Control Register 52

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	SLCTRL6	SLCTRL5	SLCTRL4	SLCTRL3	SLCTRL2	SLCTRL1	SLCTRL0	32MXCTRL
Type	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 7~1 **SLCTRL [6:0]**: Sleep Clock Control
0000000: (Optimized - do not change)
1111111: (default)

Bit 0 **32MXCTRL**: Start-up Circuit in 32MHz Crystal Oscillator Control
1: Enable (default)
0: Disable

- LREG0x253 - RFCTRL53: RF Control Register 53

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	FIFOPS	DIGITALPS	P32MXE	PACEN2	PACTRL2-2	PACTRL2-1	PACTRL2-0
Type	R/W	R/W	R/W	R/W	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7 Reserved: Maintain as ″0b0″

Bit 6 **FIFOPS**: FIFO Power while the RF Transceiver is in Power Saving Mode
1: GND
0: VDD (default)

Bit 5 **DIGITALPS**: Digital Power while Sleep
1: GND
0: VDD (default)

Bit 4 **P32MXE**: Partial 32MHz Clock Enable
1: Enable
0: Disable (default)

Bit 3 **PACEN2**: Power Amplifier Control 2 Enable
1: Enable
0: Disable (default)

Bit 2~0 **PACTRL2-[2:0]**: Power Amplifier Control 2
000: (default)
PACTRL2 [2:0] is for 1st stage Power Amplifier current fine tuning. Please follow the TX Output Power Configuration Summary Table in LREG0x203 Register definition.

- LREG0x254 - RFCTRL54: RF Control Register 54

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	1MCSEN	1MCSCH6	1MCSCH5	1MCSCH4	1MCSCH3	1MCSCH2	1MCSCH1	1MCSCH0
Type	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 **1MCSEN**: 1 MHz Channel Spacing Enable
1: Enable. When LREG0x254 [7] = 1, RF channel can only be selected by LREG0x254 [6:0] and the setting of LREG0x200 [7:4] will not change the channel number at all.
0: Disable (default)

Bit 6~0 **1MCSCH [6:0]**: 1 MHz Channel Spacing Channel Number
LREG0x254 [6:0] only works when LREG0x254 [7] = 1.

0000000: 2400 MHz	0100001: 2433 MHz	1000010: 2466 MHz
0000001: 2401 MHz	0100010: 2434 MHz	1000011: 2467 MHz
0000010: 2402 MHz	0100011: 2435 MHz	1000100: 2468 MHz
0000011: 2403 MHz	0100100: 2436 MHz	1000101: 2469 MHz
0000100: 2404 MHz	0100101: 2437 MHz	1000110: 2470 MHz
0000101: 2405 MHz	0100110: 2438 MHz	1000111: 2471 MHz
0000110: 2406 MHz	0100111: 2439 MHz	1001000: 2472 MHz
0000111: 2407 MHz	0101000: 2440 MHz	1001001: 2473 MHz
0001000: 2408 MHz	0101001: 2441 MHz	1001010: 2474 MHz
0001001: 2409 MHz	0101010: 2442 MHz	1001011: 2475 MHz
0001010: 2410 MHz	0101011: 2443 MHz	1001100: 2476 MHz
0001011: 2411 MHz	0101100: 2444 MHz	1001101: 2477 MHz
0001100: 2412 MHz	0101101: 2445 MHz	1001110: 2478 MHz
0001101: 2413 MHz	0101110: 2446 MHz	1001111: 2479 MHz
0001110: 2414 MHz	0101111: 2447 MHz	1010000: 2480 MHz
0001111: 2415 MHz	0110000: 2448 MHz	1010001: 2481 MHz
0010000: 2416 MHz	0110001: 2449 MHz	1010010: 2482 MHz
0010001: 2417 MHz	0110010: 2450 MHz	1010011: 2483 MHz
0010010: 2418 MHz	0110011: 2451 MHz	1010100: 2484 MHz
0010011: 2419 MHz	0110100: 2452 MHz	1010101: 2485 MHz
0010100: 2420 MHz	0110101: 2453 MHz	1010110: 2486 MHz
0010101: 2421 MHz	0110110: 2454 MHz	1010111: 2487 MHz
0010110: 2422 MHz	0110111: 2455 MHz	1011000: 2488 MHz
0010111: 2423 MHz	0111000: 2456 MHz	1011001: 2489 MHz
0011000: 2424 MHz	0111001: 2457 MHz	1011010: 2490 MHz
0011001: 2425 MHz	0111010: 2458 MHz	1011011: 2491 MHz
0011010: 2426 MHz	0111011: 2459 MHz	1011100: 2492 MHz
0011011: 2427 MHz	0111100: 2460 MHz	1011101: 2493 MHz
0011100: 2428 MHz	0111101: 2461 MHz	1011110: 2494 MHz
0011101: 2429 MHz	0111110: 2462 MHz	1011111: 2495 MHz
0011110: 2430 MHz	0111111: 2463 MHz	1100000: Undefined
0011111: 2431 MHz	1000000: 2464 MHz	:
0100000: 2432 MHz	1000001: 2465 MHz	1111111: Undefined

• LREG0x259 - RFCTRL59: RF Control Register 59

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	—	—	—	—	—	—	PLLOPT3
Type	R	R	R	R	R	R	R	R
POR	0	0	0	0	0	0	0	0

Bit 7~1 Reserved: Maintain as "0b0000000"

Bit 0 **PLLOPT3**: PLL Performance Optimization
1: (default)
0: (Optimized - do not change)

• LREG0x273 - RFCTRL73: RF Control Register 73

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	VCOTXOPT1	VCOTXOPT0	—	—	PLLOPT2	PLLOPT1	PLLOPT0	—
Type	R/W	R/W	R	R	R/W	R/W	R/W	R
POR	0	0	0	0	0	0	0	0

Bit 7~6 **VCOTXOPT [1:0]**: VCO for TX Optimization
00: (default)
01: (Optimized - do not change)

Bit 5~4 Reserved: Maintain as "0b00"

Bit 3~1 **PLLOPT [2:0]**: PLL Performance Optimization
000: (default)
111: Optimized for DC-DC Converter Bypass

Bit 0 Reserved: Maintain as "0b0"

• LREG0x274 - RFCTRL74: RF Control Register 74

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	PAC0EN	PACTRL0-2	PACTRL0-1	PACTRL0-0	PAC1EN	PACTRL1-2	PACTRL1-1	PACTRL1-0
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	0	0	1	0	1	0

Bit 7 **PAC0EN**: Power Amplifier Control 0 Enable
1: Enable (default)
0: Disable

Bit 6~4 **PACTRL0 [2:0]**: Power Amplifier Control 0
100: (default)
PACTRL0 [2:0] is for 1st stage Power Amplifier current large scale control.

Bit 3 **PAC1EN**: Power Amplifier Control 1 Enable
1: Enable (default)
0: Disable

Bit 2-0 **PACTRL1 [2:0]**: Power Amplifier Control 1
100: Optimized for DC-DC on
110: Optimized for DC-DC off
010: (default)
PACTRL1 [2:0] is for 2nd stage Power Amplifier current control. Please follow the TX Output Power Configuration Summary Table in LREG0x203 Register definition.

- LREG0x275 - RFCTRL75: RF Control Register 75

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	—	—	—	SCLKOPT3	SCLKOPT2	SCLKOPT1	SCLKOPT0
Type	R	R	R	R	R/W	R/W	R/W	R/W
POR	0	0	0	1	0	1	0	1

Bit 7~4 Reserved: Maintain as ″0b0001″

Bit 3~0 **SCLKOPT [3:0]**: Sleep Clock Optimization
0011: (Optimized - do not change)
0101: (default)

- LREG0x276 - RFCTRL76: RF Control Register 76

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	—	—	—	—	SCLKOPT6	SCLKOPT5	SCLKOPT4
Type	R	R	R	R	R	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	1

Bit 7~3 Reserved: Maintain as ″0b00000″

Bit 2~0 **SCLKOPT [6:4]**: Sleep Clock Optimization
111: (Optimized - do not change)
001: (default)

- LREG0x277 - RFCTRL77: RF Control Register 77

Bit	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Name	—	—	SLPSEL1	SLPSEL0	SLPVCTRL1	SLPVCTRL0	SLPVSEL1	SLPVSEL0
Type	R	R	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	1	0	0	0

Bit 7~6 Reserved: Maintain as ″0b00″

Bit 5~4 **SLPSEL [1:0]**: Power Saving Mode Selection
00: (default) Standby / Deep Sleep Mode
01: Undefined
10: Undefined
11: Power down Mode

Bit 3~2 **SLPVCTRL [1:0]**: Sleep Voltage Control
00: Undefined
01: Controlled by LREG0x27 [1:0]
10: (default) automatically controlled by internal circuit
11: Undefined

Bit 1~0 **SLPVSEL [1:0]**: Sleep Voltage Selection
00: (default - no not change)

RF Transceiver Power-down and Wake-up

The MCU and RF Transceiver are powered down independently of each other. The method of powering down the MCU is covered in the previous MCU section of the datasheet. The RF Transceiver must be powered down before the MCU is powered down. The method of powering down the RF Transceiver is mentioned in the previous Power Saving Mode section of this datasheet.

For a RF Transceiver interrupt to occur, in addition to the bits for the related enable and interrupt polarity control described in RF Transceiver Interrupt Configuration section being set, the global interrupt enable control and the related interrupt enable control bits in host MCU must also be set. If the bits related to the interrupt function are configured properly, the RF Transceiver will generate an interrupt signal on INT pin connected to the MCU I/O pin to get the attentions from MCU and then the interrupt subroutine will be serviced. If the related interrupt control bits in host MCU are not set properly, then the interrupt signal on RF Transceiver INT line will be a wake-up signal and no interrupt will be serviced.

Using the RF Transceiver Function

To use the RF Transceiver function, several important steps must be implemented to ensure that the RF Transceiver operates normally:

- The host MCU must be configured as the Master SPI. Therefore, the MCU SPI mode selection bits [M1, M0] in SPI Interface Control Register can not be set to [1, 1] as slave mode.
- Although the SPI mode selection bits [M1, M0] can be set to [0, 0], [0, 1] and [1, 0], along with the SPI clock source selection bit CKS, to force the host MCU SPI interface to operate as Master SPI with different baud rate, there are some limitations on the maximum SPI clock speed that can be selected to be suitable for the RF Transceiver slave SPI clock speed. As the maximum RF Transceiver slave SPI clock frequency is 5MHz, care must be taken for the combinations of the SPI clock source selection CKS and mode selection [M1, M0] when the system clock frequency is greater than 5MHz. For example, if the system clock operates at a frequency of 6MHz, the SPI mode selection [M1, M0] and clock source selection CKS should not be set to [0, 0] and 0. Doing so will obtain a Master SPI baud rate of 6MHz that is greater than the maximum clock frequency of the slave SPI which may result in SPI interface malfunction.
  - SPI Master/Slave/Baud rate selection bits in SBCR Register

Bit	Bit 6	Bit 5
Name	M1	M0
Value	00, 01, 10	

  00: SPI master mode; baud rate is f_{SPI}

  01: SPI master mode; baud rate is $f_{SPI}/4$

  10: SPI master mode; baud rate is $f_{SPI}/16$

  11: SPI slave mode → can not be used

  - SPI Clock source selection bit in SBCR Register

Bit	Bit 7
Name	CKS
Value	0, 1

  0: $f_{SPI} = f_{SYS}/2$

  1: $f_{SPI} = f_{SYS}$

- Since the MSB is first shifted in on SI line and shifted out on SO line for the RF Transceiver slave SPI read/write operations, the MSB/LSB selection bit MLS in MCU Master SPI SBCR register should be set to 1 for MSB shift first on SDI/SDO lines.
  - SPI MSB/LSB first selection bit in SBCR Register

Bit	Bit 5
Name	MLS
Setting value	1

- As the RF Transceiver slave SPI timing diagram shows, the SPI data output mode selection SPI_MODE and the clock polarity selection SPI_CPOL of the master SPI should be correctly set to fit the slave SPI protocol requirement. To successfully communicate with the RF Transceiver slave SPI, the SPI_MODE and SPI_CPOL of the MCU master SPI should be set to [1, 1].
  - SPI_MODE and SPI_CPOL setup in SPIR Register

Bit	Bit 1	Bit 0
Name	SPI_MODE	SPI_CPOL
Setting value	1	1

  The relevant timing diagram for the above setting is shown in the preceding SPI Bus Timing diagram in SPI Serial Interface section.

- For the MCU master SPI to completely control the slave SPI SEN line, the MCU master SPI uses a general purpose I/O line via application program instead of the SPI $\overline{SCS}$ line with hardware mechanism. To achieve this requirement, the software CSEN enable control bit SPI_CSEN of the Master SPI should be set to 0, then the master SPI $\overline{SCS}$ line will lose the $\overline{SCS}$ line characteristics and be configured as a general purpose I/O line.
  - SPI_CSEN software CSEN enable control bit in SPIR Register

Bit	Bit 2
Name	SPI_CSEN
Setting value	0

  0: the software CSEN function is disabled and the $\overline{SCS}$ line is configured as an I/O line

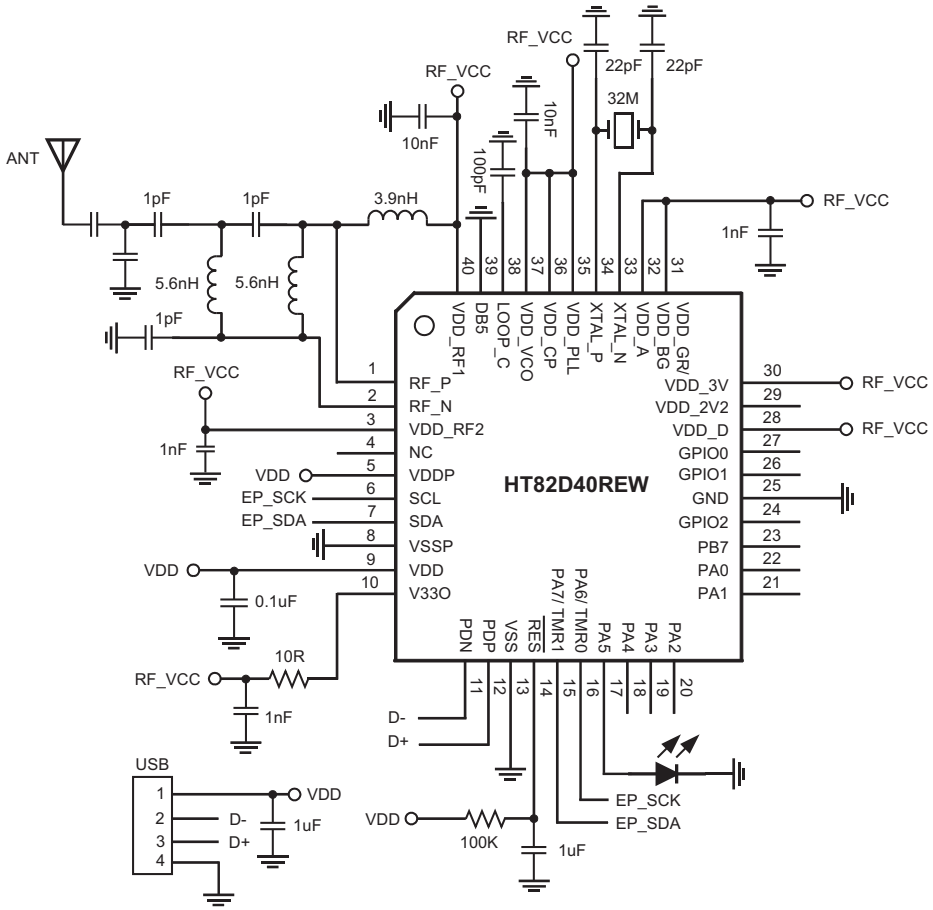
- Finally set the SPI_EN bit to 1 to ensure that the pin-shared function for other three SPI lines known as SCK, SDI and SDO are surely selected.
  - SPI_EN software SPI interface lines enable control bit in SPIR Register

Bit	Bit 2
Name	SPI_EN
Setting value	1

  1: the pin-shared function of the SPI interface lines is enabled.

  After the above setup conditions have been implemented, the MCU can enable the SPI interface by setting the SBEN bit high. The MCU can then begin communication with the RF Transceiver using the SPI interface. The detailed MCU Master SPI functional description is provided within the SPI Serial Interface section of the MCU datasheet.

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

Mnemonic	Description	Cycles	Flag Affected
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
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](4)	Read ROM code (locate by TBLP and TBHP) to data memory and TBLH	2Note	None
TABRDC [m](5)	Read ROM code (current page) to data memory and TBLH	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.

4. Configuration option ″TBHP option″ is enabled

5. Configuration option ″TBHP option″ is disabled

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 bitwise 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 $ACC \leftarrow \overline{[m]}$

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 $[m] \leftarrow ACC + 00H$ or
$[m] \leftarrow ACC + 06H$ or
$[m] \leftarrow ACC + 60H$ or
$[m] \leftarrow ACC + 66H$

Affected flag(s) C

DEC [m] Decrement Data Memory

Description Data in the specified Data Memory is decremented by 1.

Operation $[m] \leftarrow [m] - 1$

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 $ACC \leftarrow [m] - 1$

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 $TO \leftarrow 0$
$PDF \leftarrow 1$

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 $ACC \leftarrow ACC - [m] - \overline{C}$

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 $[m] \leftarrow ACC - [m] - \overline{C}$

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 $[m] \leftarrow [m] - 1$
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 $ACC \leftarrow [m] - 1$
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 $[m] \leftarrow FFH$

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 $[m].i \leftarrow 1$

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 $\neq$ 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

TABRDC [m]	Move the ROM code (locate by TBLP and TBHP) to TBLH and data memory (ROM code TBHP is enabled)
Description	The low byte of ROM code addressed by the table pointers (TBLP and TBHP) is moved to the specified data memory and the high byte transferred to TBLH directly.
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

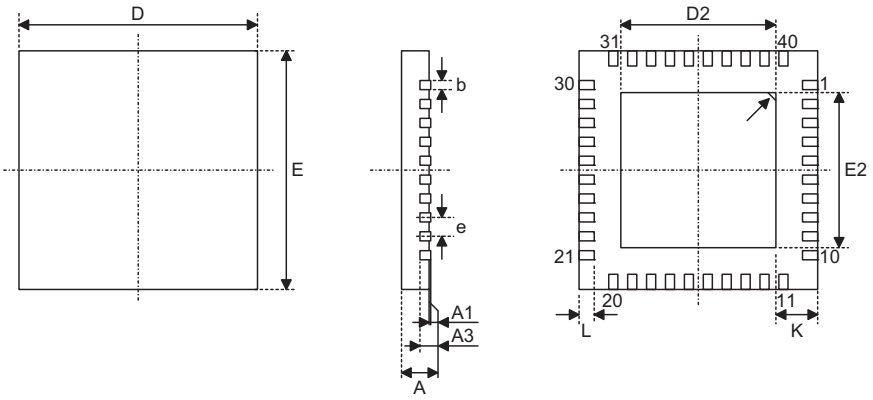
XOR A,[m]	Logical XOR Data Memory to ACC
Description	Data in the Accumulator and the specified Data Memory perform a bitwise logical XOR operation. The result is stored in the Accumulator.
Operation	ACC ← ACC ″XOR″ [m]
Affected flag(s)	Z

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

SAW Type 40-pin (6mm×6mm for 0.75mm) QFN Outline Dimensions



• GTK

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.028	0.030	0.031
A1	0.000	0.001	0.002
A3	—	0.008	—
b	0.007	0.010	0.012
D	—	0.236	—
E	—	0.236	—
e	—	0.020	—
D2	0.173	0.177	0.179
E2	0.173	0.177	0.179
L	0.014	0.016	0.018
K	0.008	—	—

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	0.70	0.75	0.80
A1	0.00	0.02	0.05
A3	—	0.20	—
b	0.18	0.25	0.30
D	—	6.00	—
E	—	6.00	—
e	—	0.50	—
D2	4.40	4.50	4.55
E2	4.40	4.50	4.55
L	0.35	0.40	0.45
K	0.20	—	—

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 Inc. (Shenzhen Sales Office)
5F, Unit A, Productivity Building, No.5 Gaoxin M 2nd Road, Nanshan District, Shenzhen, China 518057
Tel: 86-755-8616-9908, 86-755-8616-9308
Fax: 86-755-8616-9722

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