HT45B0F
SPI-to-UART Bridge

Features

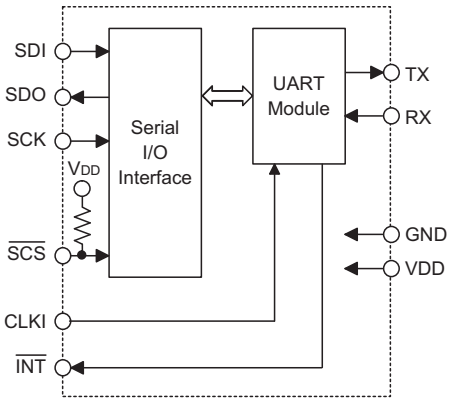
- Operating Voltage
  - f_{CLKI}= 12MHz: 2.0V~5.5V
  - f_{CLKI}= 16MHz: 2.7V~5.5V
  - f_{CLKI}= 20MHz: 4.5V~5.5V
- Glueless slave SPI interface to Holtek's MCU
- Full-duplex, Universal Asynchronous Receiver and Transmitter (UART) communication
  - 8 or 9 bit character length
  - Even, odd or no parity options
  - One or two stop bits
  - Baud rate generator with 8-bit prescaler
  - Parity, framing, noise and overrun error detection
  - Support for interrupt on address detect
  - Address Detect Interrupt - last character bit=1
  - Transmitter and receiver enabled independently
  - 4-byte deep FIFO receiver data buffer
  - Transmit and Receive Multiple Interrupt Generation Sources:
    - Transmitter Empty
    - Transmitter Idle
    - Receiver Full
    - Receiver Overrun
    - Address Mode Detect
  - TX pin is high impedance when the UART transmit module is disabled
  - RX pin is high impedance when the UART receive module is disabled
- CMOS clock input, CLKI, up to 20MHz at 5V operating voltage
- 16-pin NSOP package

General Description

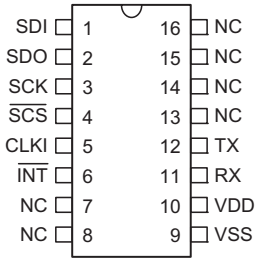
The HT45B0F is microcontroller peripheral device to implement SPI-to-UART data conversions. The UART module can operate in full-duplex mode with an interrupt output capability to the host microcontroller.

Possible application could include data communication networks between microcontrollers, low-cost data links between PCs and peripheral devices, portable and battery operated devices, factory automation and process control to name but a few.

Block Diagram



Pin Assignment



HT45B0F
16 NSOP-A

Pin Description

Pin Name	I/O	Description
SDI	I	Serial I/O data input SDI is high impedance when $\overline{SCS}$ = HIGH
SDO	O	Serial I/O data output. SDO is high impedance when $\overline{SCS}$ = HIGH
SCK	I	Serial I/O clock input SCK is high impedance when $\overline{SCS}$ = HIGH
$\overline{SCS}$	I	Serial I/O Chip select input, active low $\overline{SCS}$ has an internal pull-high resistor.
CLKI	I	External clock input
$\overline{INT}$	O	Interrupt output (CMOS output structure). Connected to the MCU's external interrupt input
RX	I	UART RX Pin If UARTEN = 1 and RXEN = 1, then RX is the UART serial data input If UARTEN = 0 or RXEN = 0, then RX is high impedance
TX	O	UART TX Pin If UARTEN = 1 and TXEN = 1, then TX is the UART serial data output If UARTEN = 0 or TXEN = 0, then TX is high impedance
VDD	—	Positive power supply
VSS	—	Negative power supply, ground
NC	—	No connection

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 35mA

Total Power Dissipation 135mW

Storage Temperature ..... –50°C to +150°C

Operating Temperature ..... –40°C to +85°C

I_{OH} Total ..... –35mA

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	—	f_{CLKI}= 12MHz	2.0	—	5.5	V
			f_{CLKI}= 16MHz	2.7	—	5.5	V
			f_{CLKI}= 20MHz	4.5	—	5.5	V
I_{DD1}	Operating Current (SPI Enabled, UART disabled)	2.0V	f_{CLKI}=12MHz, SCK=f_{CLKI}/4, Output no load	—	—	0.8	mA
		2.7V	f_{CLKI}=16MHz, SCK=f_{CLKI}/4, Output no load	—	—	1.0	mA
		3.3V		—	—	1.5	mA
		5.0V		—	—	2.0	mA
i_{DD2}	Operating Current (SPI enabled, UART enabled)	2.0V	f_{CLKI}=6MHz, SCK=f_{CLKI}/4, Output no load	—	—	4.0	mA
		2.7V		—	—	4.5	mA
		3.3V	f_{CLKI}=12MHz, SCK=f_{CLKI}/4, Output no load	—	—	5.0	mA
		5.0V		—	—	6.0	mA
I_{STB}	Standby Current (SPI disabled, UART disabled)	2.0V	f_{CLKI}=12MHz, SCK=f_{CLKI}/4, $\overline{SCS}$=V_{DD}, UARTEN=0, TXEN=1, RXEN=1, SDI=H, RX=H, Output no load	—	—	0.6	μA
		2.7V	f_{CLKI}=16MHz, SCK=f_{CLKI}/4, $\overline{SCS}$=V_{DD}, UARTEN=0, TXEN=1, RXEN=1, SDI=H, RX=H, Output no load	—	—	0.6	μA
		3.3V		—	—	0.6	μA
		5.0V		—	—	0.6	μA
V_{IL1}	Input Low Voltage for I/O Ports	—	—	0	—	0.3V_{DD}	V
V_{IH1}	Input High Voltage for I/O Ports	—	—	0.7V_{DD}	—	V_{DD}	V
V_{IL2}	Input Low Voltage for CLKI	—	—	0	—	0.3V_{DD}	V
V_{IH2}	Input High Voltage for CLKI	—	—	0.7V_{DD}	—	V_{DD}	V
I_{OL}	I/O Port Sink Current	2.0V	V_O=0.1V_{DD}	0.5	1.0	—	mA
		2.7V		1.0	2.0	—	mA
		3.3V		4.0	10.0	—	mA
		5.0V		10.0	25.0	—	mA
I_{OH}	I/O Port Source Current	2.0V	V_O=0.9V_{DD}	–0.4	–0.8	—	mA
		2.7V		–1.0	–2.0	—	mA
		3.3V		–4.0	–5.0	—	mA
		5.0V		–5.0	–8.0	—	mA
R_{PH}	Pull-high Resistance for $\overline{SCS}$ only	2.0V	—	30	90	150	kΩ
		2.7V		20	60	100	kΩ
		3.3V		15	45	75	kΩ
		5.0V		10	30	50	kΩ

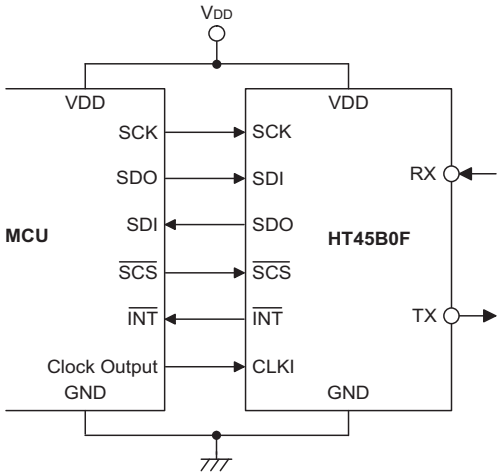
A.C. Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
	V_{DD} Slew Rate for POR	—	—	0.0034	—	—	V/ms
f_{CLKI}	System Clock (External Clock Input)	2.0V	—	400	—	12000	kHz
		2.7V	—	400	—	16000	kHz
		5.0V	—	400	—	20000	kHz
t_{CP}	SCK Period (t_{CH} + t_{CL})	2.0V	—	83.3	—	—	ns
		2.7V	—	62.5	—	—	ns
		3.0V	—	62.5	—	—	ns
		5.0V	—	50.0	—	—	ns
t_{CH}	SCK High Time	2.0V	—	37	—	—	ns
		2.7V	—	28	—	—	ns
		3.0V	—	28	—	—	ns
		5.0V	—	22	—	—	ns
t_{CL}	SCK Low Time	2.0V	—	37	—	—	ns
		2.7V	—	28	—	—	ns
		3.0V	—	28	—	—	ns
		5.0V	—	22	—	—	ns
t_{CSW}	$\overline{SCS}$ High Pulse Width	2.0V	—	667	—	—	ns
		2.7V	—	500	—	—	ns
		3.0V	—	500	—	—	ns
		5.0V	—	400	—	—	ns
t_{CSS}	$\overline{SCS}$ to SCK Setup Time	—	—	100	—	—	ns
t_{CSH}	$\overline{SCS}$ to SCK Hold Time	—	—	0	—	—	ns
t_{SDS}	SDI to SCK Setup Time	—	—	100	—	—	ns
t_{SDH}	SDI to SCK Hold Time	—	—	0	—	—	ns
t_R	SPI Output Rise Time	—	—	—	10	—	ns
t_F	SPI Output Fall Time	—	—	—	10	—	ns
t_W	SPI Data Output Delay Time	—	—	0	—	—	ns
$t_{\overline{INT}}$	$\overline{INT}$ Pin Low Pulse Signal Output Time	—	—	1	—	—	t_{CLKI}

Functional Description

The HT45B0F is full-duplex asynchronous serial communications UART interface that enables communication with external devices that contain a serial interface. The UART function has many features and can transmit and receive data serially by transferring a frame of data with eight or nine data bits per transmission as well as being able to detect errors when the data is overwritten or incorrectly framed. All data transmissions and receptions between MCU and HT45B0F including UART commands are conducted along this interconnected SPI interface. The UART function control is executed by the MCU using its SPI Master serial interface. The device contains its own independent interrupt which can be used to indicate when a data reception occurs or when a data transmission has terminated.



For Example Connection

SPI Interface

The MCU communicates with the device via an internal SPI interface. The SPI interface on this device is comprised of four signals: SCS (SPI Chip Select), SCK (SPI Clock), SDI (Serial Data Input) and SDO (Serial Data Output). The SPI master, which is the MCU, asserts SCS by pulling it low to start the data transaction cycle. When the first 8 bits of data are transmitted, SCS should not return to a high level. Instead, SCS must remain at a low level until the whole 16-bit data transaction is completed. If SCS is de-asserted, that is returned to a high level before the 16-bit data transaction is completed, all data bits will be discarded by the device SPI slave.

SPI Timing

Both read and write operations are conducted along the SPI common interface with the following format:

- Write Type Format: 8-bit command input + 8-bit data input
- Read Type Format: 8-bit command input + 8-bit data output

To initiate a data transaction, the MCU master SPI needs to pull SCS to a low level first and then also pull SCK low. The input data bit on SDI should be stable before the next SCK rising edge, as the device will latch the SDI status on the next SCK rising edge. Regarding the SDO line, the output data bit will be updated on the SCK falling edge. The master needs to obtain the line status before the next SCK falling edge.

There are 16 bits of data transmitted and/or received by the SPI interface for each transaction. Each transaction consists of a command phase and a data phase. When SCS is high, the SPI interface is disabled and SDO will be set to a high impedance state.

After a complete transaction has been implemented, which requires 16 SCK clock cycles, the master needs to set SCS to a high level in preparation for the next data transaction.

For write operations, the device will begin to execute the command only after it receives a 16-bit serial data sequence and when the SCS has been set high again by the master.

For read operations, the device will begin to execute the command only after it receives an 8-bit read command after which it will be ready to output data. If necessary, the master can de-assert the SCS pin to abort the transaction at any time which will cause any data transactions to be abandoned.

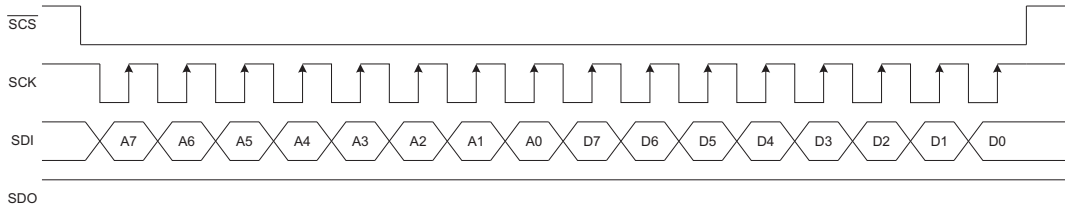
Pin Interfacing

The UART has two pins known as TX and RX. The TX pin is the UART transmitter serial data output pin if the corresponding control bits named UARTEN in UCR1 register and TXEN in UCR2 register are set to 1. If the control bit UARTEN or TXEN is equal to zero, the TX pin is in the state of high impedance. Similarly, the RX pin is the UART receiver serial data input pin if the corresponding control bits named UARTEN and RXEN in UCR1 and UCR2 registers are set to 1. If the control bit UARTEN or RXEN is equal to zero, the RX pin is in the state of high impedance.

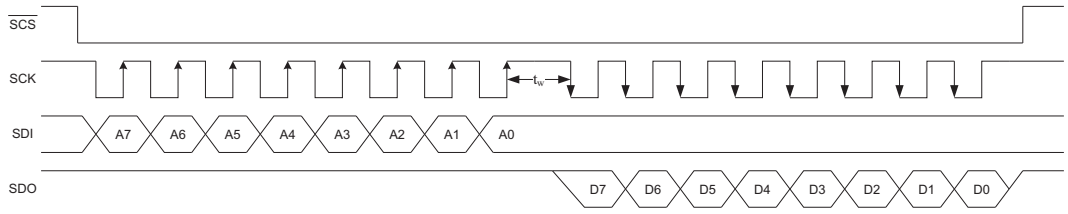
UART Data Transfer Scheme

The following block diagram shows the overall data transfer structure arrangement for the UART. The actual data to be transmitted from the MCU is first transferred to the TXR register by the application program. The data will then be transferred to the Transmitter Shift Register named TSR from where it will be shifted out, LSB first, onto the TX pin at a rate controlled by the Baud Rate Generator. Only the TXR register is accessible to the application program, the Transmitter Shift Register is not mapped into the Data Memory area and is inaccessible to the application program.

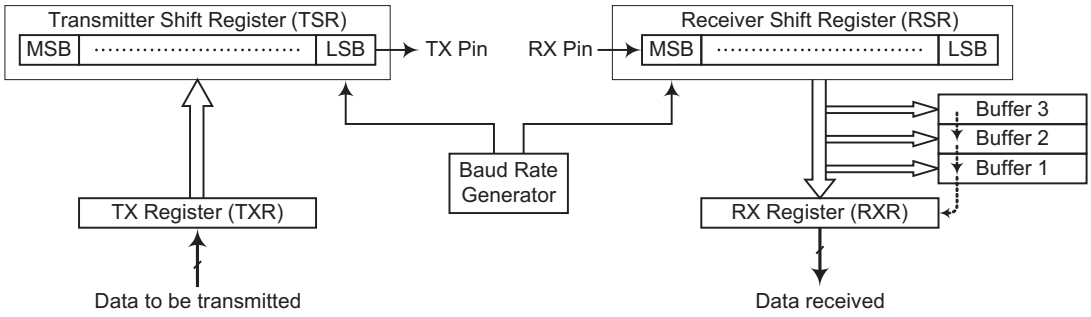
Data to be received by the UART is accepted on the RX pin, from where it is shifted in, LSB first, to the Receiver Shift Register named RSR at a rate controlled by the Baud Rate Generator. When the shift register is full, the data will then be transferred from the shift register to the internal RXR register, where it is buffered and can be manipulated by the application program. Only the RXR register is accessible to the application program, the Receiver Shift Register is not mapped into the Data Memory area and is inaccessible to the application program. It should be noted that the actual register for data transmission and reception, although referred to in the text, and in application programs, as separate TXR and RXR registers, only exists as a single shared register physically. This shared register known as the TXR/RXR register is used for both data transmission and data reception.



Writing Type Format: 8-bit Command Input + 8-bit Data Input



Reading Type Format: 8-bit Command Input + 8-bit Data Output



UART Data Transfer Scheme

UART Commands

There are both read and write commands. For reading and writing to registers both command and address information is contained within a single byte. The format for reading and writing is shown in the following table.

Command Type	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Read FIFO	0	0	0	0	0	X	X	X
Read Register	0	0	0	1	0	A2	A1	A0
Write FIFO	0	0	0	0	1	X	X	X
Write Register	0	0	0	1	1	A2	A1	A0

Note: ″X″ here stands for ″don't care″

UART Status and Control Registers

There are six registers associated with the UART function. The USR, UCR1, UCR2 and UCR3 registers control the overall function, while the BRG register controls the Baud rate. The actual data to be transmitted and received on the serial interface is managed through the TXR/RXR data register.

A[2:0]	Name	Reset	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
00H	USR	0000 1011	PERR	NF	FERR	OERR	RIDLE	RXIF	TIDLE	TXIF
01H	UCR1	0000 0X00	UARTEN	BNO	PREN	PRT	STOPS	TXBRK	RX8	TX8
02H	UCR2	0000 0000	TXEN	RXEN	BRGH	ADDEN	WAKE	RIE	TIIE	TEIE
03H	BRG	XXXX XXXX	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0
04H	UCR3	0––– –––––	URST	—	—	—	—	—	—	—
05H~ 07H	Unused	––––– –––––	Reserved							

UART Register Summary

- USR Register

The USR register is the status register for the UART, which can be read by the application program to determine the present status of the UART. All flags within the USR register are read only. Further explanation on each of the flags is given below:

Bit	7	6	5	4	3	2	1	0
Name	PERR	NF	FERR	OERR	RIDLE	RXIF	TIDLE	TXIF
R/W	R	R	R	R	R	R	R	R
POR	0	0	0	0	1	0	1	1

Bit 7 **PERR**: Parity error flag
0: no parity error is detected
1: parity error is detected
The PERR flag is the parity error flag. When this read only flag is ″0″, it indicates a parity error has not been detected. When the flag is ″1″, it indicates that the parity of the received word is incorrect. This error flag is applicable only if Parity mode (odd or even) is selected. The flag can also be cleared by a software sequence which involves a read to the status register USR followed by an access to the RXR data register.

Bit 6 **NF**: Noise flag
0: no noise is detected
1: noise is detected
The NR flag is the noise flag. When this read only flag is ″0″, it indicates no noise condition. When the flag is ″1″, it indicates that the UART has detected noise on the receiver input.
The NF flag is set during the same cycle as the RXIF flag but will not be set in the case of as overrun. The NF flag can be cleared by a software sequence which will involve a read to the status register USR followed by an access to the RXR data register.

Bit 5 **FERR**: Framing error flag
0: no framing error is detected
1: framing error is detected
The FERR flag is the framing error flag. When this read only flag is ″0″, it indicates that there is no framing error. When the flag is ″1″, it indicates that a framing error has been detected for the current character. The flag can also be cleared by a software sequence which will involve a read to the status register USR followed by an access to the RXR data register.

Bit 4 **OERR**: Overrun error flag
0: no overrun error is detected
1: overrun error is detected
The OERR flag is the overrun error flag which indicates when the receiver buffer has overflowed. When this read only flag is ″0″, it indicates that there is no overrun error. When the flag is ″1″, it indicates that an overrun error occurs which will inhibit further transfers to the RXR receive data register. The flag is cleared by a software sequence, which is a read to the status register USR followed by an access to the RXR data register.

Bit 3 **RIDLE**: Receiver status
0: data reception is in progress (data being received)
1: no data reception is in progress (receiver is idle)
The RIDLE flag is the receiver status flag. When this read only flag is ″0″, it indicates that the receiver is between the initial detection of the start bit and the completion of the stop bit. When the flag is ″1″, it indicates that the receiver is idle. Between the completion of the stop bit and the detection of the next start bit, the RIDLE bit is ″1″ indicating that the UART receiver is idle and the RX pin stays in logic high condition.

Bit 2 **RXIF**: Receive RXR data register status
0: RXR data register is empty
1: RXR data register has available data
The RXIF flag is the receive data register status flag. When this read only flag is ″0″, it indicates that the RXR read data register is empty. When the flag is ″1″, it indicates that the RXR read data register contains new data. When the contents of the shift register are transferred to the RXR register, an interrupt is generated if RIE=1 in the UCR2 register. If one or more errors are detected in the received word, the appropriate receive-related flags NF, FERR, and/or PERR are set within the same clock cycle. The RXIF flag is cleared when the USR register is read with RXIF set, followed by a read from the RXR register, and if the RXR register has no data available.

Bit 1 **TIDLE**: Transmission idle
0: data transmission is in progress (data being transmitted)
1: no data transmission is in progress (transmitter is idle)
The TIDLE flag is known as the transmission complete flag. When this read only flag is ″0″, it indicates that a transmission is in progress. This flag will be set to ″1″ when the TXIF flag is ″1″ and when there is no transmit data or break character being transmitted. When TIDLE is equal to ″1″, the TX pin becomes idle with the pin state in logic high condition. The TIDLE flag is cleared by reading the USR register with TIDLE set and then writing to the TXR register. The flag is not generated when a data character or a break is queued and ready to be sent.

Bit 0 **TXIF**: Transmit TXR data register status
0: character is not transferred to the transmit shift register
1: character has transferred to the transmit shift register (TXR data register is empty)
The TXIF flag is the transmit data register empty flag. When this read only flag is ″0″, it indicates that the character is not transferred to the transmitter shift register. When the flag is ″1″, it indicates that the transmitter shift register has received a character from the TXR data register. The TXIF flag is cleared by reading the UART status register (USR) with TXIF set and then writing to the TXR data register. Note that when the TXEN bit is set, the TXIF flag bit will also be set since the transmit data register is not yet full.

- UCR1 register

The UCR1 register together with the UCR2 register are the two UART control registers that are used to set the various options for the UART function such as overall on/off control, parity control, data transfer bit length, etc. Further explanation on each of the bits is given below:

Bit	7	6	5	4	3	2	1	0
Name	UARTEN	BNO	PREN	PRT	STOPS	TXBRK	RX8	TX8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	W
POR	0	0	0	0	0	0	X	0

″x″ unknown

Bit 7 **UARTEN**: UART function enable control
0: disable UART. TX and RX pins are in the state of high impedance
1: enable UART. TX and RX pins function as UART pins
The UARTEN bit is the UART enable bit. When this bit is equal to ″0″, the UART will be disabled and the RX pin as well as the TX pin will be in the state of high impedance. When the bit is equal to ″1″, the UART will be enabled and the TX and RX pins will function as defined by the TXEN and RXEN enable control bits. When the UART is disabled, it will empty the buffer so any character remaining in the buffer will be discarded. In addition, the value of the baud rate counter will be reset. If the UART is disabled, all error and status flags will be reset. Also the TXEN, RXEN, TXBRK, RXIF, OERR, FERR, PERR and NF bits will be cleared, while the TIDLE, TXIF and RIDLE bits will be set. Other control bits in UCR1, UCR2 and BRG registers will remain unaffected. If the UART is active and the UARTEN bit is cleared, all pending transmissions and receptions will be terminated and the module will be reset as defined above. When the UART is re-enabled, it will restart in the same configuration.

Bit 6 **BNO**: Number of data transfer bits selection
0: 8-bit data transfer
1: 9-bit data transfer
This bit is used to select the data length format, which can have a choice of either 8-bit or 9-bit format. When this bit is equal to ″1″, a 9-bit data length format will be selected. If the bit is equal to ″0″, then an 8-bit data length format will be selected. If 9-bit data length format is selected, then bits RX8 and TX8 will be used to store the 9th bit of the received and transmitted data respectively.

Bit 5 **PREN**: Parity function enable control
0: parity function is disabled
1: parity function is enabled
This is the parity enable bit. When this bit is equal to ″1″, the parity function will be enabled. If the bit is equal to ″0″, then the parity function will be disabled.

Bit 4 **PRT**: Parity type selection bit
0: even parity for parity generator
1: odd parity for parity generator
This bit is the parity type selection bit. When this bit is equal to ″1″, odd parity type will be selected. If the bit is equal to ″0″, then even parity type will be selected.

Bit 3 **STOPS**: Number of Stop bits selection
0: one stop bit format is used
1: two stop bits format is used
This bit determines if one or two stop bits are to be used. When this bit is equal to ″1″, two stop bits are used. If this bit is equal to ″0″, then only one stop bit is used.

Bit 2 **TXBRK**: Transmit break character
0: no break character is transmitted
1: break characters transmit
The TXBRK bit is the Transmit Break Character bit. When this bit is ″0″, there are no break characters and the TX pin operates normally. When the bit is ″1″, there are transmit break characters and the transmitter will send logic zeros. When this bit is equal to ″1″, after the buffered data has been transmitted, the transmitter output is held low for a minimum of a 13-bit length and until the TXBRK bit is reset.

Bit 1 **RX8**: Receive data bit 8 for 9-bit data transfer format (read only)
This bit is only used if 9-bit data transfers are used, in which case this bit location will store the 9th bit of the received data known as RX8. The BNO bit is used to determine whether data transfers are in 8-bit or 9-bit format.

Bit 0 **TX8**: Transmit data bit 8 for 9-bit data transfer format (write only)
This bit is only used if 9-bit data transfers are used, in which case this bit location will store the 9th bit of the transmitted data known as TX8. The BNO bit is used to determine whether data transfers are in 8-bit or 9-bit format.

- UCR2 register

The UCR2 register is the second of the UART control registers and serves several purposes. One of its main functions is to control the basic enable/disable operation if the UART Transmitter and Receiver as well as enabling the various UART interrupt sources. The register also serves to control the baud rate speed, receiver wake-up function enable and the address detect function enable. Further explanation on each of the bits is given below:

Bit	7	6	5	4	3	2	1	0
Name	TXEN	RXEN	BRGH	ADDEN	WAKE	RIE	TIIE	TEIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	W
POR	0	0	0	0	1	0	1	1

Bit 7 **TXEN**: UART Transmitter enable control
0: UART transmitter is disabled
1: UART transmitter is enabled
The bit named TXEN is the Transmitter Enable Bit. When this bit is equal to ″0″, the transmitter will be disabled with any pending data transmissions being aborted. In addition the buffers will be reset. In this situation the TX pin will be in the state of high impedance. If the TXEN bit is equal to ″1″ and the UARTEN bit is also equal to ″1″, the transmitter will be enabled and the TX pin will be controlled by the UART. Clearing the TXEN bit during a transmission will cause the data transmission to be aborted and will reset the transmitter. If this situation occurs, the TX pin will be in the state of high impedance.

Bit 6 **RXEN**: UART Receiver enable control
0: UART receiver is disabled
1: UART receiver is enabled
The bit named RXEN is the Receiver Enable Bit. When this bit is equal to ″0″, the receiver will be disabled with any pending data receptions being aborted. In addition the receive buffers will be reset. In this situation the RX pin will be in the state of high impedance. If the RXEN bit is equal to ″1″ and the UARTEN bit is also equal to ″1″, the receiver will be enabled and the RX pin will be controlled by the UART. Clearing the RXEN bit during a reception will cause the data reception to be aborted and will reset the receiver. If this situation occurs, the RX pin will be in the state of high impedance.

Bit 5 **BRGH**: Baud Rate speed selection
0: low speed baud rate
1: high speed baud rate
The bit named BRGH selects the high or low speed mode of the Baud Rate Generator. This bit, together with the value placed in the baud rate register BRG, controls the Baud Rate of the UART. If this bit is equal to ″1″, the high speed mode is selected. If the bit is equal to ″0″, the low speed mode is selected.

Bit 4 **ADDEN**: Address detect function enable control
0: address detect function is disabled
1: address detect function is enabled
The bit named ADDEN is the address detect function enable control bit. When this bit is equal to ″1″, the address detect function is enabled. When it occurs, if the 8th bit, which corresponds to RX7 if BNO=0 or the 9th bit, which corresponds to RX8 if BNO=1, has a value of ″1″, then the received word will be identified as an address, rather than data. If the corresponding interrupt is enabled, an interrupt request will be generated each time the received word has the address bit set, which is the 8th or 9th bit depending on the value of BNO. If the address bit known as the 8th or 9th bit of the received word is ″0″ with the address detect function being enabled, an interrupt will not be generated and the received data will be discarded.

Bit 3 **WAKE**: RX pin falling edge wake-up function enable control
0: RX pin wake-up function is disabled
1: RX pin wake-up function is enabled
This bit enables or disables the receiver wake-up function. If this bit is equal to ″1″ and the MCU is in IDLE or SLEEP mode, a falling edge on the RX input pin will through $\overline{INT}$ pin wake-up the device. If this bit is equal to ″0″ and the MCU is in IDLE or SLEEP mode, any edge transitions on the RX pin will not wake-up the device.

Bit 2 **RIE**: Receiver interrupt enable control
0: receiver related interrupt is disabled
1: receiver related interrupt is enabled
This bit enables or disables the receiver interrupt. If this bit is equal to ″1″ and when the receiver overrun flag OERR or receive data available flag RXIF is set, the UART interrupt request flag will be set. If this bit is equal to ″0″, the UART interrupt request flag will not be influenced by the condition of the OERR or RXIF flags.

Bit 1 **TIIE**: Transmitter Idle interrupt enable control
0: transmitter idle interrupt is disabled
1: transmitter idle interrupt is enabled
This bit enables or disables the transmitter idle interrupt. If this bit is equal to ″1″ and when the transmitter idle flag TIDLE is set, due to a transmitter idle condition, the UART interrupt request flag will be set. If this bit is equal to ″0″, the UART interrupt request flag will not be influenced by the condition of the TIDLE flag.

Bit 0 **TEIE**: Transmitter Empty interrupt enable control
0: transmitter empty interrupt is disabled
1: transmitter empty interrupt is enabled
This bit enables or disables the transmitter empty interrupt. If this bit is equal to ″1″ and when the transmitter empty flag TXIF is set, due to a transmitter empty condition, the UART interrupt request flag will be set. If this bit is equal to ″0″, the UART interrupt request flag will not be influenced by the condition of the TXIF flag.

- UCR3 register

The UCR3 register is the last of the UART control registers and controls the software reset operation. The only one available bit named URST in the UART control register UCR3 is the UART software reset control bit. When this bit is equal to ″0″, the UART operates normally. If this bit is equal to ″1″, the whole HT45B0F will be reset. When this situation occurs, the transmitter and receiver will be reset. The UART registers including the status register and control registers will keep the POR states shown in the above UART registers table after the reset condition occurs.

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

Bit 7 **URST**: UART software reset
0: no action
1: UART reset occurs

Bit 6~0 unimplemented, read as ″0″

Baud Rate Generator

To setup the speed of the serial data communication, the UART function contains its own dedicated baud rate generator. The baud rate is controlled by its own internal free running 8-bit timer, the period of which is determined by two factors. The first of these is the value placed in the baud rate register BRG and the second is the value of the BRGH bit with the control register UCR2. The BRGH bit decides if the baud rate generator is to be used in a high speed mode or low speed mode, which in turn determines the formula that is used to calculate the baud rate. The value N in the BRG register which is used in the following baud rate calculation formula determines the division factor. Note that N is the decimal value placed in the BRG register and has a range of between 0 and 255.

UCR2 BRGH Bit	0	1
Baud Rate (BR)	$\frac{f_{CLKI}}{[64\ (N+1)]}$	$\frac{f_{CLKI}}{[16\ (N+1)]}$

By programming the BRGH bit which allows selection of the related formula and programming the required value in the BRG register, the required baud rate can be setup. Note that because the actual baud rate is determined using a discrete value, N, placed in the BRG register, there will be an error associated between the actual and requested value. The following example shows how the BRG register value N and the error value can be calculated.

- Calculating the baud rate and error values
  For a clock frequency of 4MHz, and with BRGH set to ″0″ determine the BRG register value N, the actual baud rate and the error value for a desired baud rate of 4800.
  From the above table the desired baud rate BR = $\frac{f_{CLKI}}{[64\ (N+1)]}$

Re-arranging this equation gives $N = \frac{f_{CLKI}}{(BRx64)} - 1$

Giving a value for $N = \frac{4000000}{(4800x\,64)} - 1 = 12.0208$

To obtain the closest value, a decimal value of 12 should be placed into the BRG register. This gives an actual or calculated baud rate value of $BR = \frac{4000000}{[64(12+1)]} = 4808$

Therefore the error is equal to $\frac{4808 - 4800}{4800} = 0.16\%$

The following tables show the actual values of baud rate and error values for the two value of BRGH.

Baud Rate K/BPS	Baud Rates for BRGH=0								
	f_{CLKI}=4MHz			f_{CLKI}=3.579545MHz			f_{CLKI}=7.159MHz		
	BRG	Kbaud	Error (%)	BRG	Kbaud	Error (%)	BRG	Kbaud	Error (%)
0.3	207	0.300	0.16	185	0.300	0.00	—	—	—
1.2	51	1.202	0.16	46	1.190	–0.83	92	1.203	0.23
2.4	25	2.404	0.16	22	2.432	1.32	46	2.380	–0.83
4.8	12	4.808	0.16	11	4.661	–2.90	22	4.863	1.32
9.6	6	8.929	–6.99	5	9.321	–2.90	11	9.322	–2.90
19.2	2	20.833	8.51	2	18.643	–2.90	5	18.643	–2.90
38.4	—	—	—	—	—	—	2	32.286	–2.90
57.6	0	62.500	8.51	0	55.930	–2.90	1	55.930	–2.90
115.2	—	—	—	—	—	—	0	111.859	–2.90

Baud Rates and Error Values for BRGH = 0

Baud Rate K/BPS	Baud Rates for BRGH=1								
	f_{CLKI}=4MHz			f_{CLKI}=3.579545MHz			f_{CLKI}=7.159MHz		
	BRG	Kbaud	Error (%)	BRG	Kbaud	Error (%)	BRG	Kbaud	Error (%)
0.3	—	—	—	—	—	—	—	—	—
1.2	207	1.202	0.16	185	1.203	0.23	—	—	—
2.4	103	2.404	0.16	92	2.406	0.23	185	2.406	0.23
4.8	51	4.808	0.16	46	4.76	–0.83	92	4.811	0.23
9.6	25	9.615	0.16	22	9.727	1.32	46	9.520	–0.83
19.2	12	19.231	0.16	11	18.643	–2.90	22	19.454	1.32
38.4	6	35.714	–6.99	5	37.286	–2.90	11	37.286	–2.90
57.6	3	62.5	8.51	3	55.930	–2.90	7	55.930	–2.90
115.2	1	125	8.51	1	111.86	–2.90	3	111.86	–2.90
250	0	250	0	—	—	—	—	—	—

Baud Rates and Error Values for BRGH = 1

- BRG Register

Bit	7	6	5	4	3	2	1	0
Name	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	x	x	x	x	x	x

"x": unknown

Bit 7~0 BRG7~BRG0: Baud Rate values
By programming the BRGH bit in UCR2 Register which allows selection of the related formula described above and programming the required value in the BRG register, the required baud rate can be setup.

UART Setup and Control

For data transfer, the UART function utilizes a non-return-to-zero, more commonly known as NRZ, format. This is composed of one start bit, eight or nine data bits and one or two stop bits. Parity is supported by the UART hardware and can be setup to be even, odd or no parity. For the most common data format, 8 data bits along with no parity and one stop bit, denoted as 8, N, 1, is used as the default setting, which is the setting at power-on. The number of data bits and stop bits, along with the parity, are setup by programming the corresponding BNO, PRT, PREN and STOPS bits in the UCR1 register. The baud rate used to transmit and receive data is setup using the internal 8-bit baud rate generator, while the data is transmitted and received LSB first. Although the transmitter and receiver of the UART are functionally independent, they both use the same data format and baud rate. In all cases stop bits will be used for data transmission.

- Enabling/Disabling the UART

  The basic on/off function of the internal UART function is controlled using the UARTEN bit in the UCR1 register. If the UARTEN, TXEN and RXEN bits are set, then these two UART pins will act as normal TX output pin and RX input pin respectively. If no data is being transmitted on the TX pin, then it will default to a logic high value.

  Clearing the UARTEN bit will disable the TX and RX pins and these two pins will be in the state of high impedance. When the UART function is disabled, the buffer will be reset to an empty condition, at the same time discarding any remaining residual data. Disabling the UART will also reset the enable control, the error and status flags with bits TXEN, RXEN, TXBRK, RXIF, OERR, FERR, PERR and NF being cleared while bits TIDLE, TXIF and RIDLE will be set. The remaining control bits in the UCR1, UCR2 and BRG registers will remain unaffected. If the UARTEN bit in the UCR1 register is cleared while the UART is active, then all pending transmissions and receptions will be immediately suspended and the UART will be reset to a condition as defined above. If the UART is then subsequently re-enabled, it will restart again in the same configuration.

- Data, parity and stop bit selection

  The format of the data to be transferred is composed of various factors such as data bit length, parity on/off, parity type, address bits and the number of stop bits. These factors are determined by the setup of various bits within the UCR1 register. The BNO bit controls the number of data bits which can be set to either 8 or 9. The PRT bit controls the choice if odd or even parity. The PREN bit controls the parity on/off function. The STOPS bit decides whether one or two stop bits are to be used. The following table shows various formats for data transmission. The address detect mode control bit identifies the frame as an address character. The number of stop bits, which can be either one or two, is independent of the data length.

Start Bit	Data Bits	Address Bits	Parity Bits	Stop Bit
Example of 8-bit Data Formats				
1	8	0	0	1
1	7	0	1	1
1	7	1	0	1
Example of 9-bit Data Formats				
1	9	0	0	1
1	8	0	1	1
1	8	1	0	1

Transmitter Receiver Data Format

The following diagram shows the transmit and receive waveforms for both 8-bit and 9-bit data formats.

- UART transmitter

  Data word lengths of either 8 or 9 bits can be selected by programming the BNO bit in the UCR1 register. When BNO bit is set, the word length will be set to 9 bits. In this case the 9th bit, which is the MSB, needs to be stored in the TX8 bit in the UCR1 register. At the transmitter core lies the Transmitter Shift Register, more commonly known as the TSR, whose data is obtained from the transmit data register, which is known as the TXR register. The data to be transmitted is loaded into this TXR register by the application program. The TSR register is not written to with new data until the stop bit from the previous transmission has been sent out. As soon as this stop bit has been transmitted, the TSR can then be loaded with new data from the TXR register, if it is available. It should be noted that the TSR register, unlike many other registers, is not directly mapped into the Data Memory area and as such is not available to the application program for direct read/write operations. An actual transmission of data will normally be enabled when the TXEN bit is set, but the data will not be transmitted until the TXR register has been loaded with data and the baud rate generator has defined a shift clock source. However, the transmission can also be initiated by first loading data into the TXR register, after which the TXEN bit can be set. When a transmission of data begins, the TSR is normally empty, in which case a transfer to the TXR register will result in an immediate transfer to the TSR. If during a transmission the TXEN bit is cleared, the transmission will immediately cease and the transmitter will be reset. The TX output pin will then return to the high impedance state.

- Transmitting data

  When the UART is transmitting data, the data is shifted on the TX pin from the shift register, with the least significant bit LSB first. In the transmit mode, the TXR register forms a buffer between the internal bus and the transmitter shift register. It should be noted that if 9-bit data format has been selected, then the MSB will be taken from the TX8 bit in the UCR1 register. The steps to initiate a data transfer can be summarized as follows:

  - Make the correct selection of the BNO, PRT, PREN and STOPS bits to define the required word length, parity type and number of stop bits.
  - Setup the BRG register to select the desired baud rate.
  - Set the TXEN bit to ensure that the UART transmitter is enabled and the TX pin is used as a UART transmitter pin.
  - Access the USR register and write the data that is to be transmitted into the TXR register. Note that this step will clear the TXIF bit.

  This sequence of events can now be repeated to send additional data.

  It should be noted that when TXIF=0, data will be inhibited from being written to the TXR register. Clearing the TXIF flag is always achieved using the following software sequence:

  1. A USR register access
  2. A TXR register write execution

  The read-only TXIF flag is set by the UART hardware and if set indicates that the TXR register is empty and that other data can now be written into the TXR register without overwriting the previous data. If the TEIE bit is set, then the TXIF flag will generate an interrupt.

  During a data transmission, a write instruction to the TXR register will place the data into the TXR register, which will be copied to the shift register at the end of the present transmission. When there is no data transmission in progress, a write instruction to the TXR register will place the data directly into the shift register, resulting in the commencement of data transmission, and the TXIF bit being immediately set. When a frame transmission is complete, which happens after stop bits are sent or after the break frame, the TIDLE bit will be set. To clear the TIDLE bit the following software sequence is used:

  1. A USR register access
  2. A TXR register write execution

  Note that both the TXIF and TIDLE bits are cleared by the same software sequence.

- Transmitting break

  If the TXBRK bit is set, then the break characters will be sent on the next transmission. Break character transmission consists of a start bit, followed by 13×N ″0″ bits, where N=1, 2, etc. if a break character is to be transmitted, then the TXBRK bit must be first set by the application program and then cleared to generate the stop bits. Transmitting a break character will not generate a transmit interrupt. Note that a break condition length is at least 13 bits long. If the TXBRK bit is continually kept at a logic high level, then the transmitter circuitry will transmit continuous break characters. After the application program has cleared the TXBRK bit, the transmitter will finish transmitting the last break character and subsequently send out one or two stop bits. The automatic logic high at the end of the last break character will ensure that the start bit of the next frame is recognized.

- UART receiver

  The UART is capable of receiving word lengths of either 8 or 9 bits can be selected by programming the BNO bit in the UCR1 register. When BNO bit is set, the word length will be set to 9 bits. In this case the 9th bit, which is the MSB, will be stored in the RX8 bit in the UCR1 register. At the receiver core lies the Receiver Shift Register more commonly known as the RSR. The data which is received on the RX external input pin is sent to the data recovery block. The data recovery block operating speed is 16 times that of the baud rate, while the main receive serial shifter operates at the baud rate. After the RX pin is sampled for the stop bit, the received data in RSR is transferred to the receive data register, if the register is empty. The data which is received on the external RX input pin is sampled three times by a majority detect circuit to determine the logic level that has been placed onto the RX pin. It should be noted that the RSR register, unlike many other registers, is not directly mapped into the Data Memory area and as such is not available to the application program for direct read/write operations.

- Receiving data

  When the UART receiver is receiving data, the data is serially shifted in on the external RX input pin to the shift register, with the least significant bit LSB first. The RXR register is a four byte deep FIFO data buffer, where four bytes can be held in the FIFO while the 5th byte can continue to be received. Note that the application program must ensure that the data is read from RXR before the 5th byte has been completely shifted in, otherwise the 5th byte will be discarded and an overrun error OERR will be subsequently indicated. The steps to initiate a data transfer can be summarized as follows:

  - Make the correct selection of the BNO, PRT, PREN and STOPS bits to define the required word length, parity type and number of stop bits.
  - Setup the BRG register to select the desired baud rate.
  - Set the RXEN bit to ensure that the UART receiver is enabled and the RX pin is used as a UART receiver pin.

  At this point the receiver will be enabled which will begin to look for a start bit.

  When a character is received, the following sequence of events will occur:

  - The RXIF bit in the USR register will be set then RXR register has data available, at least three more character can be read.
  - When the contents of the shift register have been transferred to the RXR register and if the RIE bit is set, then an interrupt will be generated.
  - If during reception, a frame error, noise error, parity error or an overrun error has been detected, then the error flags can be set.

  The RXIF bit can be cleared using the following software sequence:

  1. A USR register access
  2. A RXR register read execution

- Receiving break

  Any break character received by the UART will be managed as a framing error. The receiver will count and expect a certain number of bit times as specified by the values programmed into the BNO and STOPS bits. If the break is much longer than 13 bit times, the reception will be considered as complete after the number of bit times specified by BNO and STOPS. The RXIF bit is set, FERR is set, zeros are loaded into the receive data register, interrupts are generated if appropriate and the RIDLE bit is set. If a long break signal has been detected and the receiver has received a start bit, the data bits and the invalid stop bit, which sets the FERR flag, the receiver must wait for a valid stop bit before looking for the next start bit. The receiver will not make the assumption that the break condition on the line is the next start bit. A break is regarded as a character that contains only zeros with the FERR flag set. The break character will be loaded into the buffer and no further data will be received until stop bits are received. It should be noted that the RIDLE read only flag will go high when the stop bits have not yet been received. The reception of a break character on the UART registers will result in the following:

  - The framing error flag, FERR, will be set.
  - The receive data register, RXR, will be cleared.
  - The OERR, NF, PERR, RIDLE or RXIF flags will possibly be set.

- Idle status

  When the receiver is reading data, which means it will be in between the detection of a start bit and the reading of a stop bit, the receiver status flag in the USR register, otherwise known as the RIDLE flag, will have a zero value. In between the reception of a stop bit and the detection of the next start bit, the RIDLE flag will have a high value, which indicates the receiver is in an idle condition.

- Receiver interrupt

  The read only receive interrupt flag RXIF in the USR register is set by an edge generated by the receiver. An interrupt is generated if RIE=1, when a word is transferred from the Receive Shift Register, RSR, to the Receive Data Register, RXR. An overrun error can also generate an interrupt if RIE=1.

Managing Receiver Errors

Several types of reception errors can occur within the UART module, the following section describes the various types and how they are managed by the UART.

- Overrun Error – OERR flag
  The RXR register is composed of a four byte deep FIFO data buffer, where four bytes can be held in the FIFO register, while a 5th byte can continue to be received. Before the 5th byte has been entirely shifted in, the data should be read from the RXR register. If this is not done, the overrun error flag OERR will be consequently indicated.
  In the event of an overrun error occurring, the following will happen:
  - The OERR flag in the USR register will be set.
  - The RXR contents will not be lost.
  - The shift register will be overwritten.
  - An interrupt will be generated if the RIE bit is set.

  The OERR flag can be cleared by an access to the USR register followed by a read to the RXR register.

- Noise Error – NF flag
  Over-sampling is used for data recovery to identify valid incoming data and noise. If noise is detected within a frame, the following will occur:
  - The read only noise flag, NF, in the USR register will be set on the rising edge of the RXIF bit.
  - Data will be transferred from the shift register to the RXR register.
  - No interrupt will be generated. However this bit rises at the same time as the RXIF bit which itself generates an interrupt.

  Note that the NF flag is reset by a USR register read operation followed by an RXR register read operation.

- Framing Error – FERR flag
  The read only framing error flag, FERR, in the USR register, is set if a zero is detected instead of stop bits. If two stop bits are selected, both stop bits must be high. Otherwise the FERR flag will be set. The FERR flag is buffered along with the received data and is cleared in any reset.

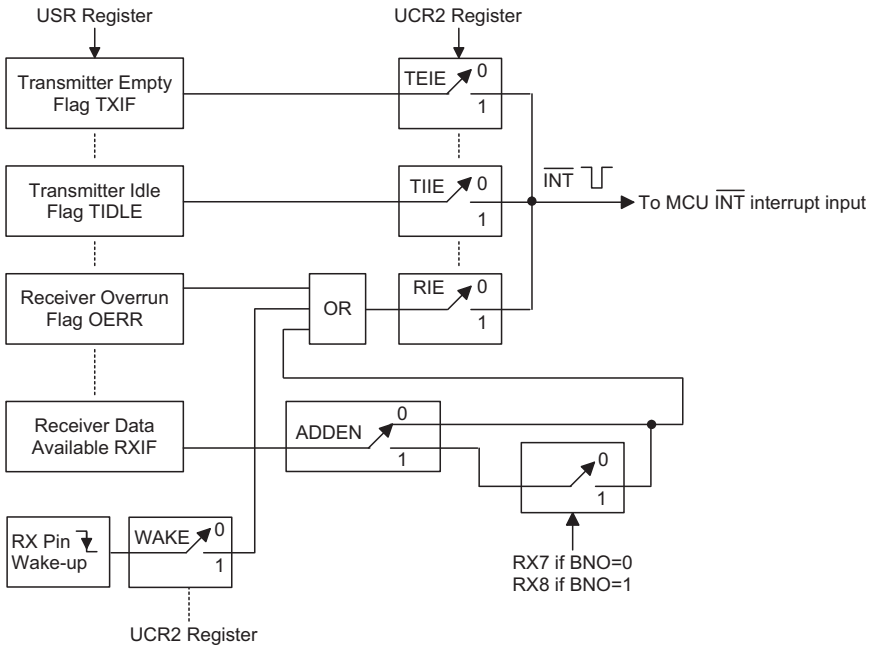
- Parity Error – PERR flag
  The read only parity error flag, PERR, in the USR register, is set if the parity of the received word is incorrect. This error flag is only applicable if the parity function is enabled, PREN=1, and if the parity type, odd or even, is selected. The read only PERR flag is buffered along with the received data bytes. It is cleared on any reset, it should be noted that the FERR and PERR flags are buffered along with the corresponding word and should be read before reading the data word.

UART Interrupt Structure

Several individual UART conditions can generate a UART interrupt. When these conditions exist, a low pulse will be generated on the $\overline{INT}$ line to get the attention of the microcontroller. These conditions are a transmitter data register empty, transmitter idle, receiver data available, receiver overrun, address detect and an RX pin wake-up. When any of these conditions are created, if its corresponding interrupt control is enabled and the stack is not full, the program will jump to its corresponding interrupt vector where it can be serviced before returning to the main program. Four of these conditions have the corresponding USR register flags which will generate a UART interrupt if its associated interrupt enable control bit in the UCR2 register is set. The two transmitter interrupt conditions have their own corresponding enable control bits, while the two receiver interrupt conditions have a shared enable control bit. These enable bits can be used to mask out individual UART interrupt sources.

The address detect condition, which is also a UART interrupt source, does not have an associated flag, but will generate a UART interrupt when an address detect condition occurs if its function is enabled by setting the ADDEN bit in the UCR2 register. An RX pin wake-up, which is also a UART interrupt source, does not have an associated flag, but will generate a UART interrupt if the microcontroller is woken up by a falling edge on the RX pin, if the WAKE and RIE bits in the UCR2 register are set. Note that in the event of an RX wake-up interrupt occurring, there will be a certain period of delay, commonly known as the System Start-up Time, for the oscillator to restart and stabilize before the system resumes normal operation.

Note that the USR register flags are read only and cannot be cleared or set by the application program, neither will they be cleared when the program jumps to the corresponding interrupt servicing routine, as is the case for some of the other interrupts. The flags will be cleared automatically when certain actions are taken by the UART, the details of which are given in the UART register section. The overall UART interrupt can be disabled or enabled by the related interrupt enable control bits in the interrupt control registers of the microcontroller to decide whether the interrupt requested by the UART module is masked out or allowed.



Interrupt Structure

- Address detect mode

  Setting the Address Detect function enable control bit, ADDEN, in the UCR2 register, enables this special function. If this bit is set to "1", then an additional qualifier will be placed on the generation of a Receiver Data Available interrupt, which is requested by the RXIF flag. If the ADDEN bit is equal to "1", then when the data is available, an interrupt will only be generated, if the highest received bit has a high value. Note that the related interrupt enable control bit and the EMI bit of the microcontroller must also be enabled for correct interrupt generation. The highest address bit is the 9th bit if the bit BNO=1 or the 8th bit if the bit BNO=0. If the highest bit is high, then the received word will be defined as an address rather than data. A Data Available interrupt will be generated every time the last bit of the received word is set. If the ADDEN bit is equal to "0", then a Receive Data Available interrupt will be generated each time the RXIF flag is set, irrespective of the data last but status. The address detect and parity functions are mutually exclusive functions. Therefore if the address detect function is enabled, then to ensure correct operation, the parity function should be disabled by resetting the parity function enable bit PREN to zero.

ADDEN	Bit 9 if BNO=1, Bit 8 if BNO=0	UART Interrupt Generated
0	0	√
	1	√
1	0	X
	1	√

ADDEN Bit Function

Power-down and Wake-up

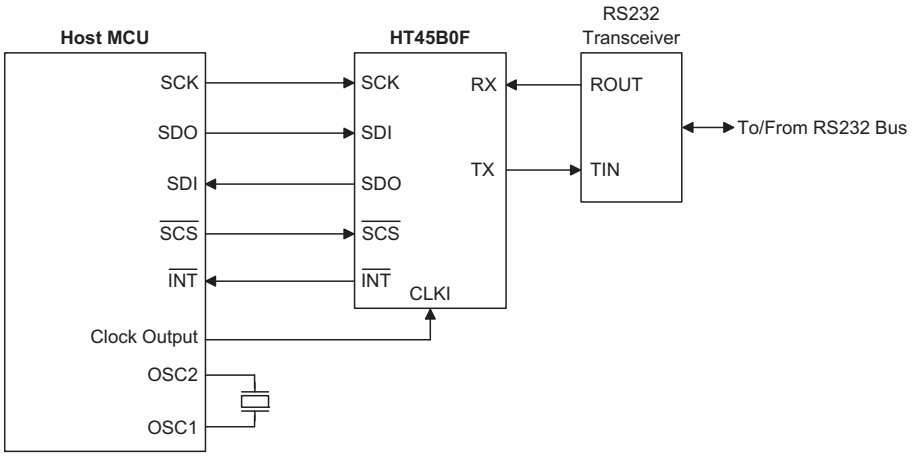
The MCU and device are powered down independently of each other. The method of powering down the MCU is covered in the previous MCU section of the datasheet. The device must be powered down before the MCU is powered down. This is implemented by first clearing the UARTEN bit in the UCR1 register to disable the UART Module circuitry after which the $\overline{SCS}$ internal line can be set high to disable the SPI interface circuits. When the UART and SPI interfaces are powered down, the SCK and CLKI clock sources to the device will be disabled. The device can be powered up by the MCU by first clearing the $\overline{SCS}$ line to zero and then setting the UARTEN bit. If the UART circuits is powered down while a transmission is still in progress, then the transmission will be terminated and the external TX transmit pin will be forced to a logic high level. In a similar way, if the UART circuits is powered down while receiving data, then the reception of data will likewise be terminated. When the UART circuits is powered down, note that the USR, UCR1, UCR2, UCR3, transmit and receive registers, as well as the BRG register will not be affected.

The device contains a receiver RX pin wake-up function, which is enabled or disabled by the WAKE bit in the UCR2 register. If this bit, along with the UART enable bit named UARTEN, the receiver enable bit named RXEN and the receiver interrupt enable bit named RIE, are all set before the MCU and device are powered down, then a falling edge on the RX pin will wake up the MCU from its power down condition. Note that as it takes a certain period of time known as the System Start-up Time for oscillator to restart and stabilize after a wake-up, any data received during this time on the RX pin will be ig-

nored.

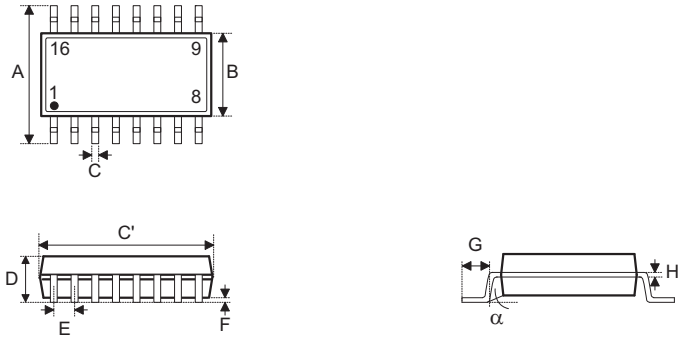
For a UART wake-up interrupt to occur, in addition to the bits for the wake-up enable control and Receive interrupt enable control being set, the global interrupt enable control and the related interrupt enable control bits must also be set. If these two bits are not set, then only a wake-up event will occur and no interrupt will be serviced. Note also that as it takes a period of delay after a wake-up before normal microcontroller resumes, the relevant UART interrupt will not be serviced until this period of delay time has elapsed.

Application Circuit



Package Information

16-pin NSOP (150mil) Outline Dimensions



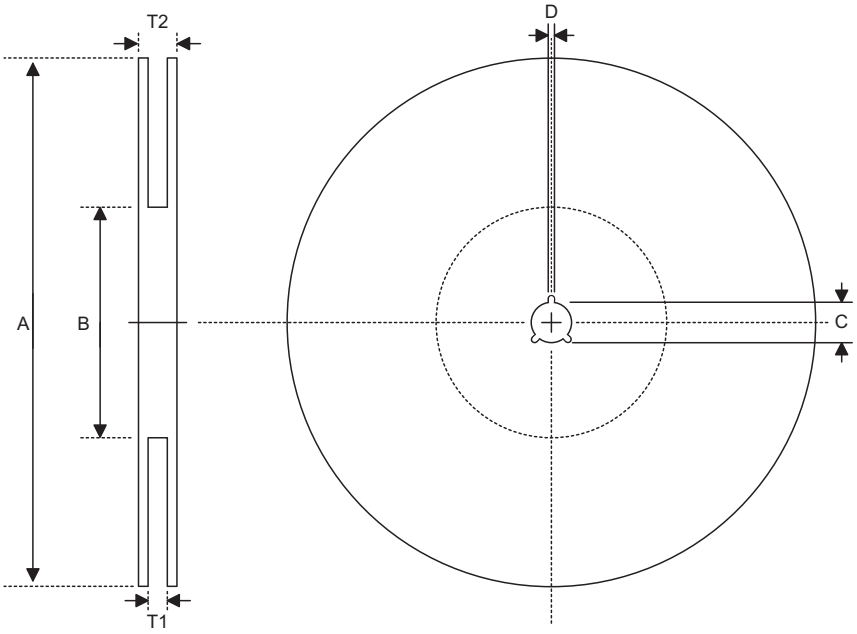
- MS-012

Symbol	Dimensions in inch		
	Min.	Nom.	Max.
A	0.228	—	0.244
B	0.150	—	0.157
C	0.012	—	0.020
C′	0.386	—	0.402
D	—	—	0.069
E	—	0.050	—
F	0.004	—	0.010
G	0.016	—	0.050
H	0.007	—	0.010
α	0°	—	8°

Symbol	Dimensions in mm		
	Min.	Nom.	Max.
A	5.79	—	6.20
B	3.81	—	3.99
C	0.30	—	0.51
C′	9.80	—	10.21
D	—	—	1.75
E	—	1.27	—
F	0.10	—	0.25
G	0.41	—	1.27
H	0.18	—	0.25
α	0°	—	8°

Product Tape and Reel Specifications

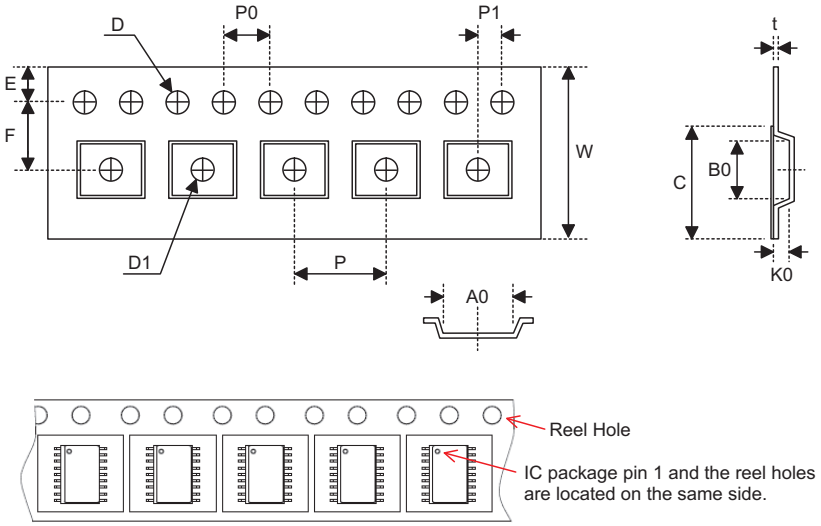
Reel Dimensions



16-pin NSOP (150mil)

Symbol	Description	Dimensions in mm
A	Reel Outer Diameter	330.0±1.0
B	Reel Inner Diameter	100.0±1.5
C	Spindle Hole Diameter	13.0 $^{+0.5/-0.2}$
D	Key Slit Width	2.0±0.5
T1	Space Between Flange	16.8 $^{+0.3/-0.2}$
T2	Reel Thickness	22.2±0.2

Carrier Tape Dimensions



16-pin NSOP (150mil)

Symbol	Description	Dimensions in mm
W	Carrier Tape Width	16.0±0.3
P	Cavity Pitch	8.0±0.1
E	Perforation Position	1.75±0.1
F	Cavity to Perforation (Width Direction)	7.5±0.1
D	Perforation Diameter	1.55 +0.10/-0.00
D1	Cavity Hole Diameter	1.50 +0.25/-0.00
P0	Perforation Pitch	4.0±0.1
P1	Cavity to Perforation (Length Direction)	2.0±0.1
A0	Cavity Length	6.5±0.1
B0	Cavity Width	10.3±0.1
K0	Cavity Depth	2.1±0.1
t	Carrier Tape Thickness	0.30±0.05
C	Cover Tape Width	13.3±0.1

Copyright © 2012 by HOLTEK SEMICONDUCTOR INC.

The information appearing in this Data Sheet is believed to be accurate at the time of publication. However, Holtek assumes no responsibility arising from the use of the specifications described. The applications mentioned herein are used solely for the purpose of illustration and Holtek makes no warranty or representation that such applications will be suitable without further modification, nor recommends the use of its products for application that may present a risk to human life due to malfunction or otherwise. Holtek's products are not authorized for use as critical components in life support devices or systems. Holtek reserves the right to alter its products without prior notification. For the most up-to-date information, please visit our web site at http://www.holtek.com.tw.