HT8 MCU UART Application Note

D/N: AN0687EN

Introduction

UART, which stands for Universal Asynchronous Receiver/Transmitter, is a common serial communication protocol. The UART interface supports bidirectional communication, which can implement full-duplex or half-duplex transmission and reception. The UART interface is widely used for communication between the MCU and external peripheral hardware, such as other MCUs and sensors.

This document will introduce a UART functional description and application considerations, which can assist users to get started quickly in using the HT8 MCU UART function and accelerate product development.

Functional Description

The HT8 MCU provides two UART function interface types: the independent UART and the USIM. The main differences between the two interfaces are shown in the table below.

Differences	Independent UART	USIM
Supported communication protocol	UART	SPI, I^2C, UART
UART Mode Selection	Not necessary to select the UART mode - some MCUs support the single wire mode, which is selected using the SWM bit.	Select the UART mode using the UMD bit in the SIMC0 register - some MCUs support the single wire mode, which is selected using the USWM bit.

Table 1

The communication protocols and selection methods supported by the two UART interfaces may be different, but their principles are similar in applications.

UART Interface and Features

Interface Description

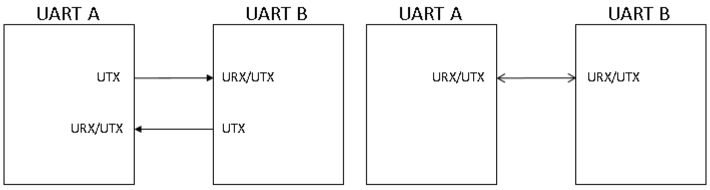
The HT8 MCU UART interface generally uses two communication pins, namely the transmitter pin, TX or UTX, and the receiver pin, RX or URX. Some MCUs support the single wire mode and can implement the half-duplex data transmission and reception using only one communication pin (RX/TX or URX/UTX). A summary of this is shown in the table below:

Pin Function	Independent UART Single Wire Mode not Supported	Independent UART Single Wire Mode Supported	USIM Single Wire Mode not Supported	USIM Single Wire Mode Supported
Transmitter pin	TX	TX or RX/TX	UTX	UTX or URX/UTX
Receiver pin	RX	RX/TX	URX	URX/UTX

Table 2. HT8 MCU UART Transmitter / Receiver pin

Note: Refer to the corresponding MCU datasheet for the specific pin names.

Taking the USIM that supports the single wire UART mode as an example, there are two common connection methods for UART communication:



a. Full-duplex Mode b. Half-duplex mode

Figure 1. UART Connection Method

Feature Introduction

The HT8 MCU UART has the following features, some of which vary depending upon different MCUs:

Item	Description
Communication Method	Full-duplex supported; some MCUs support half-duplex (single-wire) mode
Baud Rate Generator	Due to the differences in baud rate generator specifications, the HT8 MCU UART can be divided into general and high speed types: 1. General type: 8-bit prescaler supported 2. High speed type: 16-bit prescaler supported Different UART types correspond to different baud rate setting methods and calculation formulas. Refer to the UART Baud Rate Configuration section below for details
Transfer Format Length	8 or 9 bits data length selectable
Parity Type	All types support even, odd or no parity; Some MCUs support mark parity and space parity
Number of Stop Bits	Distinguished by the UART type: 1. General type: Supports selectable 1 or 2 stop bits 2. High speed type: (1) Transmission: Fixed 2 stop bits, not selectable (2) Reception: selectable 1 or 2 stop bits
Error Detection	Parity, framing, noise and overrun error detection
Transmit Data Buffer	1-byte FIFO
Receive Data Buffer	Distinguished by the UART type: 1. General: 2-byte FIFO supported 2. High speed type: 4-byte FIFO supported – a 4-byte FIFO interrupt trigger level can be set by a register
Interrupt trigger conditions	1. Transmitter empty 2. Transmitter idle 3. Receiving completed or receiver reaching the FIFO trigger level 4. Receiver overrun 5. Address mode detect 6. Receiver pin wake-up
Other	Independent transmission and reception enable

Table 3. HT8 MCU UART Feature List

UART Communication Description

For data transfer, the UART function utilises 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 automatically performed by the hardware.

The format of the data to be transferred is determined by the data bit length, parity on/off, parity type, address bits and the number of stop bits. The following table shows various formats for data transmission.

Start Bit	Data Bits	Address Bit	Parity Bit	Stop Bit
Example of 8-bit Data Formats				
1	8	0	0	1 or 2
1	7	0	1	1 or 2
1	7	1	0	1 or 2
Example of 9-bit Data Formats				
1	9	0	0	1 or 2
1	8	0	1	1 or 2
1	8	1	0	1 or 2

Table 4. UART Data Transmission Format

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

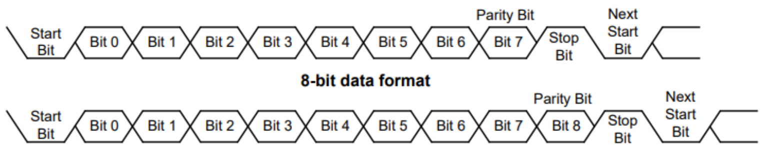


Figure 2. 8-bit Data and 9-bit Data Transmit Waveform

UART Register Description

General type UART - 8-bit baud rate timer: Taking the HT66L2530A as an example, here its UART (USIM structure) related registers are shown in the following table. Refer to the MCU datasheet for a more detailed description.

Register Name	Function
SIMC0	Select the UART mode by setting the UMD bit
UUSR	UART status register, which can be read by the program to determine the present status of the UART.
UUCR1	First UART control register - used to set the UART data structure.
UUCR2	Second UART control register - used to control the basic enable/disable operation of the UART Transmitter and Receiver as well as enabling the various UART interrupt sources.
UUCR3	Third UART control register - used to enable the UART Single Wire Mode.
UTXR_RXR	Data register - used to store the data to be transmitted on the UTX pin or being received from the URX/UTX pin.
UBRG	Used to set the baud rate.

Table 5. General Type UART Related Registers – Take HT66L2530A as an Example

High speed type UART - 16-bit baud rate timer: Taking the HT66F2030 as an example, here its UART (independent UART) related registers are shown in the following table. Refer to the MCU datasheet for a more detailed description.

Register Name	Function
USR	UART status register - can be read by the program to determine the present status of the UART.
UCR1	First UART control register - used to set the UART data structure.
UCR2	Second UART control register - used to control the basic enable/disable operation of the UART Transmitter and Receiver as well as enabling the various UART interrupt sources.
UCR3	Third UART control register - used to enable the UART Single Wire Mode.
TXR_RXR	Data register - used to store the data to be transmitted on the TX pin or being received from the RX/TX pin.
BRDH	Used to set the baud rate divider high byte.
BRDL	Used to set the baud rate divider low byte.
UFCR	FIFO control register - used for UART modulation control, BRD range selection and trigger level selection for RXIF and interrupt.
RxCNT	Used to record the number of received data bytes in the Receiver FIFO which have not been read by the MCU.

Table 6. High Speed UART Related Registers – Take HT66F2030 as an Example

UART Configuration Description

The configuration methods of the HT8 MCU UART function are introduced below, mainly including the UART pin configuration, baud rate configuration, single wire mode and interrupt configuration methods.

UART Pin Configuration

The UART communication pins are generally pin-shared with I/O ports or other functions. Before or after using the UART function, the relevant pins are required to be set according to the following instructions.

- Enable the UART communication pin function: This requires the corresponding pin-shared control registers to be configured properly before configuring the corresponding peripheral functions. Taking the HT66L2530A USIM structure UART as an example, UTX is pin-shared with PC2 and URX/UTX is pin-shared with PC1.
  1. Set PCS05~PCS04=11 and PCS03~PCS02=11
  2. Set UREN=1, UTXEN=1 and URXEN=1
- Disable the UART communication pin function: This requires the peripheral function to be disabled before configuring the corresponding pin-shared control registers to select another function. Taking the HT66L2530A USIM structure UART as an example, UTX is pin-shared with PC2 and URX/UTX is pin-shared with PC1.
  1. Set UREN=0, UTXEN=0 and URXEN=0
  2. Set PCS05~PCS04≠11 and PCS03~PCS02≠11
  3. Set PCC1=1, PCC2=1, PCPU1=1 and PCPU2=1

If only step 1 is implemented, the UART pin will be in a floating state. If the UTX or URX/UTX pins are not in use, it is recommended to set them as inputs with pull-up resistors enabled, namely step 3.

UART Baud Rate Configuration

To setup the speed of the serial data communication, i.e. the baud rate, the HT8 MCU UART function contains its own dedicated baud rate generator. According to the number of the baud rate timer bits, the baud rate generator is divided into two different specifications, 8-bit and 16-bit. The configuration methods and error calculations for these two baud rate specifications are described below.

Note that in practical applications, the baud rate maximum error should not exceed ±2%.

8-bit Timer

Taking the HT66L2530A as an example, the baud rate is controlled by the UBRG register and the UBRGH bit in the UUCR2 register. The UBRGH bit determines the formula that is used to calculate the baud rate, as shown in the following table. The value N in the UBRG 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 UBRG register and has a range of between 0 and 255.

UUCR2 UBRGH Bit	0	1
Baud Rate (BR)	$BR = f_H/[64(N+1)]$	$BR = f_H/[16(N+1)]$

By programming the UBRGH bit which allows selection of the related formula and programming the required value in the UBRG 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 UBRG register, there will be an error associated between the actual and desired value.

The following example shows how the UBRG register value N and the error value can be calculated.

- For a clock frequency of 4MHz, and with UBRGH cleared to zero, determine the UBRG 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 $\mathrm{BR} = f_H/[64(N+1)]$
- Re-arranging this equation gives $\mathrm{N} = [f_H/(BR \times 64)] - 1$, giving a value for $\mathrm{N} = [4000000/(4800 \times 64)] - 1 = 12.0208$
- To obtain the nearest value, a decimal value of 12 should be placed into the UBRG register. This gives an actual or calculated baud rate value of BR=4000000/[64×(12+1)]=4808

  Therefore the error is equal to (4808-4800)/4800=0.16%

16-bit Timer

Taking the HT66F2030 as an example, the baud rate is controlled by the BRDH/BRDL register and the UART modulation control bits UMOD2~UMOD0. The range of the BRD (BRDH/BRDL) value is determined by the BRDS bit as shown below:

$$0:\ \mathrm{BRD} = 16 \sim 65535$$

$$1:\ \mathrm{BRD} = 8 \sim 65535$$

If a baud rate BR is required with UART clock f_H:

$$f_H/BR = \text{Integer Part} + \text{Fractional Part}$$

The integer part is loaded into BRD (BRDH/BRDL). The fractional part is multiplied by 8, rounded and then loaded into the UMOD bit field below:

$$BRD = TRUNC(f_H/BR)$$

$$UMOD = ROUND[MOD(f_H/BR) \times 8]$$

Therefore, the actual baud rate is calculated as follows:

$$\text{Actual Baud Rate} = f_H/[BRD + (UMOD/8)]$$

The following example demonstrates how to calculate the BRDH/BRDL value, actual baud rate and error value.

For a clock frequency of 4MHz and a desired baud rate of 230400:

- From the above formula, $BRD = TRUNC(f_H/BR) = TRUNC(17.3611) = 17$
- $UMOD = ROUND[MOD(f_H/BR) \times 8] = ROUND[0.3611 \times 8] = ROUND(2.88888) = 3$
- $\text{The actual baud rate} = f_H/[BRD + (UMOD/8)] = 230215.83$
- Therefore the error is equal to $(230215.83 - 230400)/230400 = -0.08\%$

UART Single Wire Mode

Refer to the corresponding datasheet section for the general steps to setup the UART to transmit and receive data. Here the HT66L2530A setup method to transmit and then receive is used as an example to describe the UART single wire mode configuration steps.

- Step 1: Ensure the correct selection of the UBNO, UPRT, UPREN and USTOPS bits to define the required data length, parity type and number of stop bits
- Step 2: Setup the UBRG register to select the desired baud rate
- Step 3: Set the USWM to "1" to enable the single wire mode
- Step 4: Set UTXEN=1 and URXEN=0, to ensure that the URX/UTX pin is used as a UART transmitter pin
- Step 5: Access the UUSR register and write the data that is to be transmitted into the UTXR_RXR register
- Step 6: After the data has been transmitted, set UTXEN=0 and URXEN=1, to ensure that the URX/UTX pin is used as a receiver pin
- Step 7: Set URIE=1 to enable the receiver interrupt
- Step 8: When UOERR or URXIF are set, the USIM interrupt request flag USIMF will be set
- Step 9: Access the UUSR register to check if there are errors
- Step 10: Access the UTXR_RXR register to obtain the data

UART Interrupt

There are 6 conditions that can trigger the HT8 MCU UART interrupt. Here the HT66L2530A is taken as an example to introduce the USIM interrupt configuration method.

- Transmitter data register empty
  Set UTEIE=1 using the program. When the transmitter register is empty, UTXIF will be set by the hardware. At the same time the USIM interrupt request flag will be set.
- Transmitter idle
  Set UTIIE=1 using the program. When the transmitter is idle, UTIDLE will be set by the hardware. At the same time the USIM interrupt request flag will be set.
- Receiving completed
  Set URIE=1 using the program. When the receiving has completed and the URXIF is set, the USIM interrupt request flag will be set.
- Receiver overrun
  When a receiver overrun occurs, UOERR=1 and URIE=1, the USIM interrupt request flag will be set.
- Address mode detect
  Set UADDEN=1 using the program. When an address is detected (when the highest data bit is 1), the USIM interrupt request flag will be set.
- Receiver pin wake-up
  Set UAXEN=1, UWAKE=1 and URIE=1 using the program. If the UART clock (f_H) source is switched off and a falling edge on the URX/UTX pin occurs, the USIM interrupt request flag USIMF will be set.

Application Considerations

General Considerations

1. The data transmission format of both sides in UART communication requires consistency in its setup. This consistency includes baud rate, data length, etc. The baud rate error is directly related to the UART prescaler and system clock error. The maximum allowable error between both sides of the communication is related to the selected transmission format. For example, for the common 8-N-1 format, it has a total of 10-bits of data combined with the start bit. Assuming that each bit level sampling position is at the center point, the maximum allowable error cannot exceed ±50%/10=±5%. For the 8-E-2 or 8-O-2 format with a total of 12-bits of data, the allowable error cannot exceed ±50%/12=±4.1%. Therefore, it is recommended that the maximum error of a single MCU baud rate should not exceed ±2%.
2. When the voltage between both sides of the UART communication is inconsistent, the communication can only be carried out after level shifting. Some of the Holtek MCUs have a VDDIO pin which can provide a voltage to the UART communication pins that is different from the MCU VDD voltage.
3. When both sides of the UART communication are in an idle state, TX will output a high level. When one side of the UART is in a low-power state which uses a power-down mode, note that

a high level on the other side's TX pin may create unexpected leakage or a running state. The other side's RX pin may be in a floating state which may receive unexpected data.

4. The error flag bits (UOERR/UFERR/UPERR) in the UUSR register are read-only bits that can be cleared by software. The specific steps are to read the UUSR register first and then the UTXR_RXR register. (Note)
5. It is recommended not to set both the URXEN and UTXEN bits high in the single wire mode. If both the URXEN and UTXEN bits are set high, the URXEN bit will have a priority and the UART will act as a receiver. (Note)

Note: The register or bit names mentioned in points 4 and 5 of the above general considerations are all taken from the HT66L2530A datasheet. Refer to the corresponding MCU datasheet for practical applications.

Special Considerations

When using the high speed UART function, it is recommended to follow the following considerations:

1. When the UART is required to continuously receive multiple data frames, it is recommended that the receiver uses two stop bits to avoid a receiving error caused by the accumulated error of the receiver baud rate frequency.
2. When EMI is turned on and a subroutine is called that has an execution time longer than the time for the UART to receive five data bytes, then if the UART received data cannot be read in time during the subroutine execution, clear the RXEN bit to zero in advance to suspend data reception. When EMI is turned off, if the subroutine cannot respond to the UART interrupt in time to handle overflow errors, it is necessary to ensure that RXEN is turned off when executing the subroutine. After the subroutine has completed execution, turn RXEN on again and continue to receive UART data. (Note)
3. When the receiver overflow flag OERR is set to "1", this indicates that the 4-byte FIFO is full and an additional received data byte has been stored in the shift register. Therefore, it is necessary to read five data bytes immediately to avoid unexpected errors, such as the UART being unable to receive further data. If such an error occurs, clear the RXEN bit to "0" then set it to "1" again to continue data reception. It is recommended that users configure the UART receiver FIFO to trigger an interrupt when it contains 2 bytes (setting RxFTR1~ RxFTR0=10), avoiding an overrun state that cannot be processed by the program in time when more than 4 data bytes are received. (Note)

Note: The register or bit names mentioned in points 2 and 3 of the above special considerations are all taken from the HT66F2030 datasheet. Refer to the corresponding MCU datasheet for practical applications.

Conclusion

This application note has provided a detailed description for the HT8 MCU UART specifications and functions and also listed some application considerations. Using this guide, users should have a more comprehensive understanding of the HT8 MCU UART interface which will improve their in-depth use of the UART function.

Reference Material

Reference files: HT66L2530A and HT66F2030 Datasheet.

For more information, refer to the Holtek's official website: www.holtek.com.

Versions and Modification Information

Date	Author	Issue	Modification
2024.5.24	鄭家鑽	V1.00	First Version

Disclaimer

All information, trademarks, logos, graphics, videos, audio clips, links and other items appearing on this website ('Information') are for reference only and is subject to change at any time without prior notice and at the discretion of Holtek Semiconductor Inc. and its related companies (hereinafter 'Holtek', 'the company', 'us', 'we' or 'our'). Whilst Holtek endeavors to ensure the accuracy of the Information on this website, no express or implied warranty is given by Holtek to the accuracy of the Information. Holtek shall bear no responsibility for any incorrectness or leakage.

Holtek shall not be liable for any damages (including but not limited to computer virus, system problems or data loss) whatsoever arising in using or in connection with the use of this website by any party. There may be links in this area, which allow you to visit the websites of other companies. These websites are not controlled by Holtek. Holtek will bear no responsibility and no guarantee to whatsoever Information displayed at such sites. Hyperlinks to other websites are at your own risk.

Limitation of Liability

In no event shall Holtek Limited be liable to any other party for any loss or damage whatsoever or howsoever caused directly or indirectly in connection with your access to or use of this website, the content thereon or any goods, materials or services.

Governing Law

The Disclaimer contained in the website shall be governed by and interpreted in accordance with the laws of the Republic of China. Users will submit to the non-exclusive jurisdiction of the Republic of China courts.

Update of Disclaimer

Holtek reserves the right to update the Disclaimer at any time with or without prior notice, all changes are effective immediately upon posting to the website.