UART and SPI Data Transmission Functions using Time Sharing in the HT66FU50

D/N : AN0231E

Introduction

The HT66FU50 includes an internal UART function which is implemented through its internal SPI interface which connects to a UART Bridge IC. The following content describes the way to use pin remapping to implement the SPI and UART functions using time sharing.

UART Features

- Connected to the Holtek MCU via an SPI interface
- Full duplex asynchronous receiver/transmitter
  - 8 or 9-bit data length
  - Odd, even or no parity bit selectable
  - One or two stop bit
  - Divide by 8 serial transmission ratio generator
  - Parity, error and erroneous noise detections
  - Address detect interrupt
  - 4-bit receiver buffer
- Transmit Receiver Complex Interrupt
  - Transmitter is empty
  - Transmitter is idle
  - Receiver buffer is full
  - Receiver buffer overflows
  - Address detection mode
- CMOS clock input with a maximum value of 20MHz at 5V.
Operating Principles

UART Module

The HT66FU50 implements a UART function by connecting a UART Bridge IC via its SPI interface. The structure is shown below.

Read the register value of the UART Bridge IC through the SPI interface to implement UART data transmission using the following operation. Write: 8-bit command input + 8-bit data input. Read: 8-bit command input + 8-bit data transmission.

The following table is the command format to read the UART register of which the read/write commands and the address are contained in one single byte.

<table>
<thead>
<tr>
<th>Tone</th>
<th>Frequency</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision Dial Tone</td>
<td>350Hz+440Hz</td>
<td>Continuous high</td>
</tr>
<tr>
<td>Old Dial Tone</td>
<td>120Hz (or 133Hz, …) +600Hz</td>
<td>Continuous high</td>
</tr>
<tr>
<td>Precision Busy Tone</td>
<td>480Hz+620Hz</td>
<td>0.5sec high and 0.5sec low</td>
</tr>
<tr>
<td>Old Busy Tone</td>
<td>120Hz+600Hz</td>
<td>0.5sec high and 0.5sec low</td>
</tr>
<tr>
<td>Precision Reorder Tone</td>
<td>480Hz+620Hz</td>
<td>0.3sec high and 0.2sec low</td>
</tr>
<tr>
<td>Old Reorder Tone</td>
<td>120Hz+600Hz</td>
<td>0.2sec high and 0.3sec low or 0.25sec high and 0.25sec low</td>
</tr>
<tr>
<td>Precision Ring-back Tone</td>
<td>440Hz+480Hz</td>
<td>2sec high and 4sec low</td>
</tr>
<tr>
<td>Old Ring-back Tone</td>
<td>40Hz (or the others) +420Hz</td>
<td>2sec high and 4sec low</td>
</tr>
</tbody>
</table>
The three low bits represent the address of the register to be operated upon. Bit3~bit4 are the command bits. When writing data to the UCR1 register (addressing 01H), send 19H via the SPI and then send another one byte of data.

<table>
<thead>
<tr>
<th>Command Type</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read FIFO</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Read Register</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>A2</td>
<td>A1</td>
<td>A0</td>
</tr>
<tr>
<td>Write FIFO</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Write Register</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>A2</td>
<td>A1</td>
<td>A0</td>
</tr>
</tbody>
</table>

Note: “X” here stands for “don’t care”

The UART related registers are listed in the following table. The UCR1, UCR2 and UCR3 registers control the UART module function. The BRG register controls the serial transmission ratio. The USR register controls the UART status.

### UART Register Summary

<table>
<thead>
<tr>
<th>A[2:0]</th>
<th>Name</th>
<th>Reset</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>00H</td>
<td>USR</td>
<td>0000 1011</td>
<td>PERR</td>
<td>NF</td>
<td>FERR</td>
<td>OERR</td>
<td>RIDLE</td>
<td>RXIF</td>
<td>TIDLE</td>
<td>TXIF</td>
</tr>
<tr>
<td>01H</td>
<td>UCR1</td>
<td>0000 0X00</td>
<td>UARTEN</td>
<td>BNO</td>
<td>PREN</td>
<td>PRT</td>
<td>STOPS</td>
<td>TXBRK</td>
<td>RX8</td>
<td>TX8</td>
</tr>
<tr>
<td>02H</td>
<td>UCR2</td>
<td>0000 0000</td>
<td>TXEN</td>
<td>RXEN</td>
<td>BRGH</td>
<td>ADDEN</td>
<td>WAKE</td>
<td>RIE</td>
<td>TIE</td>
<td>TEIE</td>
</tr>
<tr>
<td>03H</td>
<td>BRG</td>
<td>XXXXXXEX</td>
<td>BRG7</td>
<td>BRG6</td>
<td>BRG5</td>
<td>BRG4</td>
<td>BRG3</td>
<td>BRG2</td>
<td>BRG1</td>
<td>BRG0</td>
</tr>
<tr>
<td>04H</td>
<td>UCR3</td>
<td>0</td>
<td>URST</td>
<td>___</td>
<td>___</td>
<td>___</td>
<td>___</td>
<td>___</td>
<td>___</td>
<td>___</td>
</tr>
<tr>
<td>05H~07H</td>
<td>Unused</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Using UART Function

Before using the UART function, the SPI pin remapping function should be correctly configured by connecting the SPI to the internal UART using the PRM0 register. Detailed configuration is shown in the table below. After the setups, set the SIMEN to 1 to enable the UART function via the SPI interface.

<table>
<thead>
<tr>
<th>BIT</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>SIMPS1</td>
<td>SIMPS0</td>
<td>PCKS</td>
</tr>
<tr>
<td>Setting value</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Data transmission steps are as follows:
1. Define the word length, the parity type and number of stop bits.
2. Setup the baud rate.
3. Set UARTEN and TXEN to 1 to enable UART transmission.
4. Read the USR register and write the data to be transmitted into TXR.

The data reception steps are as follows:
1. Define the word length, the parity type and number of stop bits.
2. Setup the baud rate.
3. Set UARTEN and RXEN as 1 to enable UART transmission.
When data is received, the following events will occur:

1. The RXIF bit will be set to 1 if the data is effective.
2. Data will be written into RXR from the shift register. If RIE is set to 1, an interrupt will be generated.
3. In the receiving process, if an error, noise error, erroneous parity or receive overflow is detected, the corresponding error flag will be set to 1.

**Pin Remapping Function**

The HT66FU50 provides a Pin Remapping function which allows different functions to share the same pin. This brings more flexibility to users by selecting some pin functions through the PRM0, PRM1 and PRM2 registers. For example, there are three sets of SPI pins in the HT66FU50 of which the pin shared with the I/O ports can be selected using the SIMPS1 and SIMPS0 bit in the PRM0 register.

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>-</td>
<td>C1XPS</td>
<td>-</td>
<td>C0XPS</td>
<td>1</td>
<td>SPRM</td>
<td>SIMPS</td>
<td>SIMPS</td>
</tr>
<tr>
<td>R/W</td>
<td>-</td>
<td>R/W</td>
<td>-</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
<td>R/W</td>
</tr>
<tr>
<td>POR</td>
<td>-</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

SIMPS1.SIMPS0 : 01 SDO ON PD3; SDI ON PD2; SCK ON PD1; SCS ON PD0
SIMPS1.SIMPS0 : 10 SDO ON PB7; SDI ON PD6; SCK ON PD6; SCS ON PD7

**H/W Function Block Diagram**

The HT66U50 firstly writes data to the Flash memory through the SPI interface and then reads it back. The pin remapping function will configure the SPI pin connected to the internal SPI of the UART module to transmit the received data to the computer through the UART and implement the SPI and UART function by time sharing.
Application Circuits

Note: there is another SPI pin set available to connect with an external device using the SPI interface.
S/W Function Block Flowchart

Start

Initialize

Erase Flash Memory

Write data to Flash Memory

Read Data to Flash Memory

Pin Remapping

Initialise UART

Send Data to PC
File Description

- **Initial.C**
  The file contents is the initialisation operation for the registers and I/O ports
- **Flash.C**
  The file contents is the subroutine for the Flash Memory read/write operation
- **UART.C**
  The file contents are the relevant functions and variable definition for UART control
- **HT66FU50UART.C**
  The file contents is the main program
- **Command.h**
  The header file defines all the constants and macros used in the program operation

Conclusion

This application has provided an introduction as to how to use the pin-remapping and UART functions. Using the pin-remapping function allows the implementation of time-sharing SPI and UART use.