SCI

### Introduction

The serial communication interface (SCI) module. SCI is a two−wire asynchronous serial port, commonly known as a UART. The SCI modules support digital communications between the CPU and other asynchronous peripherals that use the standard non-returnto-zero (NRZ) format. The SCI receiver and transmitter each have a 16-level deep FIFO for reducing servicing overhead, and each has its own separate enable and interrupt bits.

### Usage

**Initialization of SCI**

This part describe what do function scia\_echoback\_init() and scia\_fifo\_init() in “Serial.h” do.

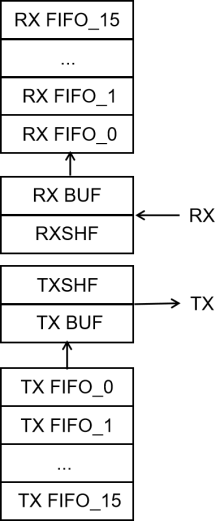
First, we list the registers used and the meaning of their bits : (The initial value of our setting is in brackets or in the second line of each register.)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Register** | **Bit7** | **Bit6** | **Bit5** | **Bit4** | **Bit3** | **Bit2** | **Bit1** | **Bit0** |
| SCICCR | STOPBITS | PARITY | PARITYENA | LOOPBKENA | ADDRIDLE\_MODE | SCICHAR | | |
| 0x0007 | 1 stop bit; No loop-back; No parity; 8 char bits; async mode; idle-line protocol | | | | | | | |
| SCICTL1 | RESERVED | RXERRINTENA | SWRESET | RESERVED | TXWAKE | SLEEP | TXENA | RXENA |
| 0x0003 | Enable TX, RX, internal SCICLK; Disable RX ERR, SLEEP, TXWAKE | | | | | | | |
| SCICTL2 | TXRDY | TXEMPTY | RESERVED | | | | RXBKINTENA | TXINTENA |
| 0x0003 | Set transmitter flag: SCITXBUF full, SCITXBUF/TXSHF loaded with data ; Enable RXRDY/BRKDT, TXRDY interrupt | | | | | | | |
| SCIHBAUD | BAUD (0x0002) | | | | | | | |
| SCILBAUD | BAUD (0x008A) | | | | | | | |
| SCI Asynchronous Baud = LSPCLK / ((BRR + 1) \*8) ; BRR = LSPCLK / (SCI Asynchronous Baud \* 8) - 1 | | | | | | | | |
| SCIFFTX | **Bit15** | **Bit14** | **Bit13** | **Bit12** | **Bit11** | **Bit10** | **Bit9** | **Bit8** |
| SCIRST | SCIFFENA | TXFIFORESET | TXFFST | | | | |
| **Bit7** | **Bit6** | **Bit5** | **Bit4** | **Bit3** | **Bit2** | **Bit1** | **Bit0** |
| TXFFINT | TXFFINTCLR | TXFFIENA | TXFFIL | | | | |
| 0xE040 | SCI Reset; SCI FIFO enable; Re-enable transmit FIFO operation; Transmit FIFO is empty; TXFIFO interrupt has not occurred, read-only bit; Write 1 to clear TXFFINT flag in bit 7;TX FIFO interrupt based on TXFFIL match (less than or equal to) is disabled | | | | | | | |
| SCIFFRX | **Bit15** | **Bit14** | **Bit13** | **Bit12** | **Bit11** | **Bit10** | **Bit9** | **Bit8** |
| RXFFOVF | RXFFOVRCLR | RXFIFORESET | RXFFST | | | | |
| **Bit7** | **Bit6** | **Bit5** | **Bit4** | **Bit3** | **Bit2** | **Bit1** | **Bit0** |
| RXFFINT | RXFFINTCLR | RXFFIENA | RXFFIL | | | | |
| 0x2044 | Receive FIFO has not overflowed, read-only bit; Write 0 has no effect on RXFFOVF flag bit, Bit reads back a zero; Re-enable receive FIFO operation; Receive FIFO is empty; RXFIFO interrupt has not occurred, read-only bit; Write 1 to clear RXFFINT flag in bit 7; RX FIFO interrupt based on RXFFIL match (greater than or equal to) will be disabled; Generates an interrupt whenever the FIFO status bits (RXFFST4-0) are greater than or equal to the FIFO level bits | | | | | | | |
| SCIFFCT | **Bit15** | **Bit14** | **Bit13** | **Bit12** | **Bit11** | **Bit10** | **Bit9** | **Bit8** |
| ABD | ABDCLR | CDC | RESERVED | | | | |
| **Bit7** | **Bit6** | **Bit5** | **Bit4** | **Bit3** | **Bit2** | **Bit1** | **Bit0** |
| FFTXDLY | | | | | | | |
| 0x0 | Auto-baud detection is not complete; Write 0 has no effect on ABD flag bit. Bit reads back a Zero; Disables auto-baud alignment; FIFO transfer delay = 0 | | | | | | | |

### SCI communication method

**Software:** Functions defined in “Serial.h” include: serialIn(), serialOut(), serialOutArray(), serialOutCharArray(), serialCheck(), serialEcho(). They are designed as basic SCI communication method for further use. Since they are self-explanatory, we will skip it.

**Hardware:** There are two external pins: SCITXD (SCI transmit-output pin), SCIRXD (SCI receive-input pin).



### Registers

### *SCI Base Address Table*

|  |  |  |  |
| --- | --- | --- | --- |
| **Device Registers** | **Register Name** | **Start Address** | **End Address** |
| SciaRegs | SCI\_REGS | 0x0000\_7200 | 0x0000\_720F |
| ScibRegs | SCI\_REGS | 0x0000\_7210 | 0x0000\_721F |
| ScicRegs | SCI\_REGS | 0x0000\_7220 | 0x0000\_722F |
| ScidRegs | SCI\_REGS | 0x0000\_7230 | 0x0000\_730F |

### *SCI\_REGS Registers*

|  |  |  |
| --- | --- | --- |
| **Offset** | **Acronym** | **Register Name** |
| 0h | SCICCR | Communications control register |
| 1h | SCICTL1 | Control register 1 |
| 2h | SCIHBAUD | Baud rate (high) register |
| 3h | SCILBAUD | Baud rate (low) register |
| 4h | SCICTL2 | Control register 2 |
| 5h | SCIRXST | Receive status register |
| 6h | SCIRXEMU | Receive emulation buffer register |
| 7h | SCIRXBUF | Receive data buffer |
| 9h | SCITXBUF | Transmit data buffer |
| Ah | SCIFFTX | FIFO transmit register |
| Bh | SCIFFRX | FIFO receive register |
| Ch | SCIFFCT | FIFO control register |
| Fh | SCIPRI | SCI Priority control |