CMPE 443
Principles of Embedded System Design
fall’11

Problem Session
Overview

- Asynchronous Serial Communication Interface (SCI)
- Sample Code
Serial Communication Interface

- SCI is a full duplex, asynchronous, serial interface
- Can send and receive 8- or 9-bit data
  - Use same baud rate for transmitter and receiver
- If SCI is enabled
  - Port S bit 0 is assigned to received-data (RXD)
  - Port S bit 1 is assigned to transmitted-data (TXD)
Serial data read from and written to the SCIDR register

- **SCIDRH**
  - Address: Base+$00CE

<table>
<thead>
<tr>
<th>Read</th>
<th>Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>R8</td>
<td>T8</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- **SCIDRL**
  - Address: Base+$00CF

<table>
<thead>
<tr>
<th>Read</th>
<th>Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>R7</td>
<td>R6</td>
</tr>
<tr>
<td>R5</td>
<td>R4</td>
</tr>
<tr>
<td>R3</td>
<td>R2</td>
</tr>
<tr>
<td>R1</td>
<td>R0</td>
</tr>
<tr>
<td>T7</td>
<td>T6</td>
</tr>
<tr>
<td>T5</td>
<td>T4</td>
</tr>
<tr>
<td>T3</td>
<td>T2</td>
</tr>
<tr>
<td>T1</td>
<td>T0</td>
</tr>
</tbody>
</table>

- In 8-bit data format, only SCIDRL has to be written and read.
Receiver and Transmitter Enable

- SCI Control Register 2 (SCICR2)
  - Address: Base+$00CB

<table>
<thead>
<tr>
<th>Bit 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>TIE</td>
<td>TCIE</td>
<td>RIE</td>
<td>ILIE</td>
<td>TE</td>
<td>RE</td>
<td>RWU</td>
<td>SBK</td>
</tr>
</tbody>
</table>

  - **TE**: Transmitter Enable
    - 0: SCI transmitter is disabled (default)
    - 1: SCI transmitter is enabled

  - **RE**: Receiver Enable
    - 0: SCI receiver is disabled (default)
    - 1: SCI receiver is enabled
SCI Mode Control

- SCI Control Register 1 (SCICR1)
  - Address: Base+$00CA

<table>
<thead>
<tr>
<th>Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

- SCIWAI: SCI stop in Wait Mode
  - 0: SCI enabled in wait mode (default)
  - 1: SCI disabled in wait mode

- M: Data Format Select
  - 0: One start, eight data, one stop bit (default)
  - 1: One start, eight data, plus ninth data, one stop bit

- PE: Parity Enable (Parity bit inserted to the most significant bit position)
  - 0: Parity is disabled (default)
  - 1: Parity is enabled

- PT: Parity Type
  - 0: Even parity (default)
  - 1: Odd parity
The rate at which serial data bits are sent is called the baud rate. "SCI Baud rate is selected by SBR12:SBR0"

- **SCI Baud Rate Register High (SCIBDH)**
  - Address: Base+$00C8

- **SCI Baud Rate Register Low (SCIBDL)**
  - Address: Base+$00C9
SCI Status Flags

- SCI Status Register 1 (SCISR1)
  - Address: Base+$00CC

<table>
<thead>
<tr>
<th>Bit 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>TDRE</td>
<td>TC</td>
<td>RDRF</td>
<td>IDLE</td>
<td>OR</td>
<td>NF</td>
<td>FE</td>
<td>PF</td>
</tr>
</tbody>
</table>

- **TDRE**: Transmitter Data Register Empty Flag (Cleared by writing to SCIDRL)
  - 0: No byte transferred to the transmit shift registers
  - 1: Byte has been transferred to the transmit shift register and transmit data register is empty (default)

- **TC**: Transmit Complete Flag (Cleared by writing to SCIDRL)
  - 0: Transmission in progress
  - 1: No transmission in progress (default)

- **RDRF**: Receiver Data Register Full Flag (Cleared by reading from SCIDRL)
  - 0: Data not available in the SCI data register (default)
  - 1: Receiver data is available

- **PF**: Parity Error Flag
  - 0: Parity on the last received data is correct (Default)
  - 1: Parity incorrect
Enable SCI Interrupts

- SCI Control Register 2 (SCICR2)
  - Address: Base + $00CB

<table>
<thead>
<tr>
<th>Bit 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>TIE</td>
<td>TCIE</td>
<td>RIE</td>
<td>ILIE</td>
<td>TE</td>
<td>RE</td>
<td>RWU</td>
<td>SBK</td>
</tr>
</tbody>
</table>

- **TIE**: Transmitter Interrupt Enable
  - 0: TDRE flag interrupt requests disabled (default)
  - 1: TDRE flag interrupt requests enabled

- **TCIE**: Transmission Complete Interrupt Enable
  - 0: TC flag interrupt requests disabled (default)
  - 1: TC flag interrupt requests enabled

- **RIE**: Receiver Interrupt Enable
  - 0: RDRF flag interrupt requests disabled (default)
  - 1: RDRF flag interrupt requests enabled
Sample Code

```c
#include <hidef.h>    /* common defines and macros */
#include <mc9s12dg128.h>    /* derivative information */
#include <string.h>
#pragma LINK_INFO DERIVATIVE "mc9s12dt128b"
#pragma inline

void initSci(void) {
    SCI0CR1_M=SCI0CR1_PE=0;
    SCI0CR2_TE=SCI0CR2_RE=1;
    SCI0BDL=26;
}

char k;

void main(void) {
    char c; k=0;
    initSci();
    EnableInterrupts;
    printChar(13);
    printString("Please enter a number between 0 and 255 in decimal");
    c=readChar();
    printChar(c);
    while(c!=13) {
        k=k*10+c-'0';
        c=readChar();
        printChar(c);
    }
    for(;;) {} /* wait forever */
}

void printChar(char c) {
    while(!SCI0SR1_TDRE);
    SCI0DRL=c;
}

void printString(char s[]) {
    int i,len=strlen(s);
    for(i=0;i<len;i++)
        printChar(s[i]);
}

char readChar(void) {
    char c;
    while(!SCI0SR1_RDRF);
    c=SCI0DRL;
    return c;
}
```