Freescale Semiconductor
Application Note

AN3035
Rev. 0, 09/2005

Using the HC08 SCI Module

By Jorge Zambada Tinoco
Oscar Luna González
RTAC Americas
Mexico 2005

Overview

This document is intended to serve as a quick reference for an embedded engineer to get the serial communications interface (SCI) module up and running for any HC08 MCU. Basic knowledge about the functional description and configuration options will give the user a better understanding on how the SCI module works.

This application note provides an example illustrating one use of the SCI module within the HC08 Family of microcontrollers. The example mentioned is intended to be modified to suit the specific needs of any application.

SCI Module

The SCI allows full duplex, asynchronous, NRZ serial communication among the MCU and remote devices, including other microcontrollers. The transmitter and receiver of the SCI operate independently, although they use the same baud rate generator. During normal operation, the CPU monitors the status of the SCI, writes the data to be transmitted, and processes received data.

freescale™ semiconductor

Features

Some features of this module are:

- Full duplex operation
- Standard mark/space non-return-to-zero (NRZ) format
- 32 programmable baud rates
- Programmable 8-bit or 9-bit character length
- Two receiver wakeup methods:
  - Idle line wakeup
  - Address mark wakeup
- Interrupt driven operations with eight interrupt flags:
  - Transmitter empty
  - Transmission complete
  - Receiver full
  - Idle receiver input
  - Receiver overrun
  - Noise error
  - Framing error
  - Parity error
- 1/16 bit time noise detection
- Two different SCI source clock selection:
  - Internal data bus clock
  - External oscillator clock

Register Description

The main registers found within the SCI module are:

1. The Configuration Register 2 (CONFIG2) which:
   - Selects the clock source used for the standard SCI module (bit 0, SCIDBSRC)
2. The SCI Control Register 1 (SCC1) which:
   - Enables loop mode operation (bit 7, LOOP)
   - Enables the SCI (bit 6, ENSCI)
   - Controls output polarity (bit 5, TXINV)
   - Controls character length (bit 4, M)
   - Controls SCI wake up method (bit 3, WAKE)
   - Controls idle character detection (bit 2, ILTY)
   - Enables parity function (bit 1, PEN)
   - Control parity type (bit 0, PTY)
3. The SCI Control Register 2 (SCC2) which:
   - Enables SCI interrupt requests (bit 7:4, SCTIE, TCIE, SCRIE, ILIE)
   - Enables the transmitter and receiver (bit 3:2, TE, RE)
   - Enables SCI wakeup (bit 1, RWU)
   - Transmits SCI break characters (bit 0, SBK)

4. The SCI Control Register 3 (SCC3) which:
   – Stores the ninth SCI data bit received and the ninth SCI data bit to be transmitted (bit 7:6, R8, T8)
   – Enables interrupts for receiver overrun, noise, framing, and parity errors (bit 3:0, ORIE, NEIE, FEIE, PEIE)
5. The SCI Status Register 1 (SCS1) which:
   – Flags a transfer of the data register to the transmit shift data register (bit 7, SCTE)
   – Flags a complete transmission (bit 6, TC)
   – Flags a transfer of the receive shift data register to the data register (bit 5, SCRF)
   – Flags a receiver input idle condition (bit 4, IDLE)
   – Flags the receiver error conditions for overrun, noisy data, framing error, and parity error (bit 3:0, OR, NF, FE, PE)
6. The SCI Status Register 2 (SCS2) which:
   – Flags a break character detection (bit 1, BKF)
   – Flags an incoming data condition (bit 0, RPF)
7. The SCI Data Register (SCDR) which:
   – When read, contains the last data received (bit 7:0, R7:R0 or T7:T0)
   – When written, contains the next data to be transmitted (bit 7:0, R7:R0 or T7:T0)
8. The SCI Baud Rate Register (SCBR) which:
   – Selects the baud rate for both the receiver and the transmitter (bit 5:4, SCP1:SCP0 and bit 2:0, SCR2:SCR0)

Code Example and Explanation

The example shown in this application note consists of:

- Getting the received data
- Adding an integer value of 1
- Sending back the resultant byte

The configuration used for the HyperTerminal is described in the Considerations section of this application note.

Following these steps, the user will be able to use the SCI for this example:

1. Configure the SCI clock source

```
CONFIG2 = 0x01;  /* Internal data bus clock source used as clock source for SCI */
```

2. Configure the microcontroller's pins for SCI communications

```
DDRE &= ~(0x02); /* Configure Rx pin as input for reception */
PTE |= 0x01;     /* Set Tx pin to have an idle state */
DDRE |= 0x01;    /* Configure Tx pin as output for transmission */
```

3. Configure SCI control register 1, 2, and 3

```
SCC1 = 0x00;  /* Loop mode disabled, disable SCI, Tx output not inverted,
                 8-bit characters, idle line wakeup, disable parity bit */
SCC2 = 0x20;  /* Enable SCI receive interrupts, Disable transmitter and
                 receiver */
SCC3 = 0x00;  /* Disable all error interrupts */
```

4. Configure the SCI baud rate register

```
*****************************************************************
*                    Fbus = XTAL/4       2.4576 MHz             *
*    Baud Rate = -------------------- = ---------- = 9600 bps    *
*                64 x SCP1:0 x SCR2:0    64 x 1 x 4              *
*****************************************************************/

SCBR = 0x02; /* Select a baud rate of 9600 bps with Fbus = 2.4576 MHz */
```

NOTE

Refer to the Serial Communications Interface section of a specific microcontroller data sheet for a detailed description of the baud rate calculation.

For detailed information about proper calculation of the SCI baud rate register (SCBR), refer to the "SCI Baud Rate Selection Examples" table found in the device data sheet's Serial Communications Interface section. This table provides the user with references to obtain the necessary SCBR (SCR2:0, SCP0:1) values for different baud rates.

5. Enable receiver, transmitter, and SCI module

```
SCC1 |= 0x40;  /* Enable SCI Module */
SCC2 |= 0x0C;  /* Enable Transmitter and Receiver */
```

6. Declare SCI interrupt function

```
void interrupt 13 SCIIsr (void) /* Declare SCI vector address interrupt */
                                / *   SCI Vector Address = 13 */
```

Since an interrupt based algorithm is being implemented, the global interrupt enable mask has to be cleared as follows:

```
EnableInterrupts;   /* __asm CLI; */
```

From this point on, the code execution is performed inside the SCI receive interrupt service routine. The code inside does the following:

1. Clears the SCI receiver full interrupt flag.

```
SCS1 &= ~(0x20);       /* Clear SCI Receiver Full Flag */
```

2. Reads the received data in a global variable called ReceivedByte and increments it.

```
ReceivedByte = SCDR;   /* Load received data into a global variable */
ReceivedByte += 1;     /* Increment received data by 1 */
```

3. Waits for the transmitter to be empty, so that we can queue a new transmission.

```
while ((SCS1 & 0x80) == 0);   /* Wait for the transmitter to be empty */
```

4. Stores the new computed byte in the SCI data register.

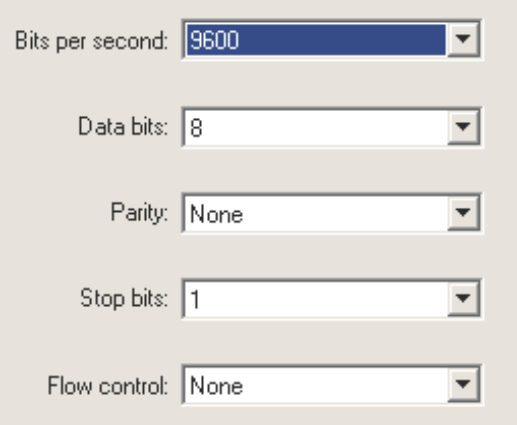
```
SCDR = ReceivedByte; /* Store new data to be transmitted */
```

Considerations

The considerations listed here must be taken.

1. This example code was developed using Metrowerks CodeWarrior IDE version 3.1 for HC08 and was expressly made for the MC68HC908GR8 in a DIP package. There may be changes needed in the code for it to be used for other HC08 Family members.

   The hardware used for the example is shown in the Schematic section. On the PC side of this example, the Windows HyperTerminal was used with the configuration shown in the following figure.



2. Values presented in “SCI Baud Rate Selection Examples” table under Serial Communications Interface in the device data sheet was calculated with f_{BUS} = 4.9152 MHz.
3. SCI I/O lines are implemented by sharing parallel I/O port pins. The full name of an SCI input or output reflects the name of the shared port pin. Table 1 shows the full names and the generic names of the SCI I/O pins. The generic pin names appear in the text of this section.
4. Another consideration that the user must take is the module availability in the HC08 Family of microcontrollers. Table 1 summarizes this availability, Table 1 also shows which microcontrollers have a different SCI module, such enhanced SCI module (ESCI) or infrared SCI module (IRSCI). These SCI module variants are not covered in this application note.

Table 1. SCI Selection Guide

HC08 Families	SCI Available			SCI Clock Source Selection Bit	
	SCI	ESCI	IRSCI	SCIBDSRC	ESCIBSRC
AB16/32	√	χ	χ	χ	χ
AP8/16/32/64	√	χ	√	√	χ
AS32/60	√	χ	χ	χ	χ
AZ32/60	√	χ	χ	χ	χ
EY16	χ	√	χ	χ	√
GP32	√	χ	χ	√	χ
GR4/8	√	χ	χ	√	χ
GR16/32/48/60	χ	√	χ	√	χ
GT8/16	χ	√	χ	χ	χ
GZ8/16/32/48/60	χ	√	χ	√	χ
JB12/16	√	χ	χ	χ	χ
JG16	√	χ	χ	χ	χ
JK8	√	χ	χ	χ	χ
JL8	√	χ	χ	χ	χ
KX2/8	√	χ	χ	√	χ
LJ12/24	χ	χ	√	χ	χ
LK24	χ	χ	√	χ	χ
MR8/16/32	√	χ	χ	χ	χ
QB4/8	χ	√	χ	√	χ
SR12	√	χ	χ	√	χ

LEGEND:

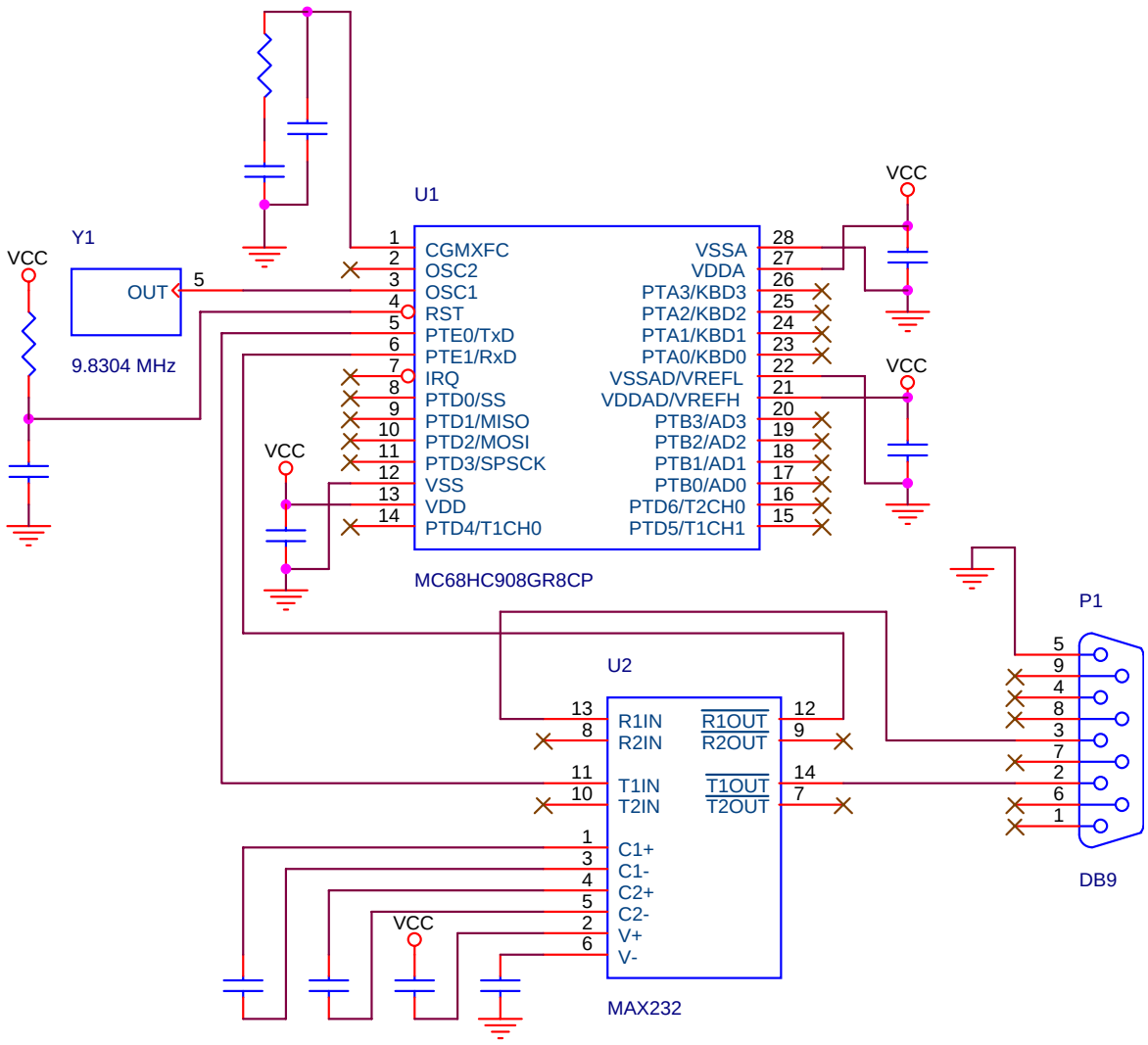
√ **= Module available on this microcontroller family**

χ = Module NOT available on this microcontroller family

NOTE:

Information gathered from the Microcontrollers Selector Guide Quarter 3, 2005 (Freescale order number SG1006). This document can be found on the World Wide Web at: http://freescale.com

Schematic



References

Refer to the following documents for more information on subjects in this application note.

- *MC68HC908GR8 Data Sheet*
- AN1818 — *Software SCI Routines with the 16-Bit Timer Module*
- AN2502 — *Using Two Channels of the HC08 TIM to Achieve a Full Duplex Software SCI*

These documents can be found on the World Wide Web at:
http://freescale.com

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

AN3035
Rev. 0, 09/2005

freescale™
semiconductor