# Microprocessor Lab
# Asynchronous Serial I/O

Serial communication is the transfer of data performed by sending one bit at a time over a communication channel. Serial communication can be either synchronous or asynchronous. Synchronous communication uses a clock signal, which is common to the receiver and transmitter, to indicate when each bit is being transmitted. Asynchronous communication does not use a common clock. The 68HC12 microcontroller contains on chip support for both synchronous and asynchronous communications. In this course, we will only be using the standard asynchronous mode (e.g., what is used when you connect a serial port to a printer).

Serial I/O is performed by Serial Communication Interface (SCI) chip contained in the 68HC12. The SCI is controlled, through software, by registers. The registers are special memory locations that transfer data to hardware. The registers on the 68HC12 are mapped into memory locations internally within the microcontroller. Typically, these addresses cannot be changed (but may differ among models of a given microprocessor since different address decoding functions can be used within the IC for different models). The three general types of register are (i) control registers "controlling" the operation of the peripheral, (ii) status registers where you can find the current state of the peripheral, and (iii) data registers through which you can send and receive data.


Figure 1

Figure 1 illustrates the signals corresponding to the data being sent across a serial RS232 connection. When no data is being transmitted on a data line, it is held in the high state. When data is transferred, a start bit must precede the data, to insure that the receiver can find the beginning of the data. The start bit is low, so that it differs from the idle data line. After the data, a stop bit must be sent to return the data line to the high state, so the next character (i.e., next start bit) can be detected. The UART generates the start and stop bits, when it transmits the data.

There are two distinct serial communication ports, SC0 and SC1. You are already using a serial port when you run the D-Bug12 monitor from the desktop computer so the serial port to the computer is already set up (baud rate, etc.). You do not need to reset that port. You can, of course, but then will need to go to the desktop computer and change the port's baud rate to match the new number you used.

The baud rate is obtained from the clock rate (P clock of the HC12) by dividing by an integer (13-bit integer). This divisor requires two 8-bit words to handle to total of 13 bits. The addresses for the two registers (i.e. SC0BDH/SC1BDH and SC0BDL/SC0BDL) in the SCI are given on page 369 of your textbook ($00C1 for the lower part and $00C1 for the upper part). Note that the top three bits of the upper register are not used. The baud rate is then equal to the P rate divided by [16 x (13-bit number in these registers)].

**PROJECTS (From Chapter 9 of HC12 Textbook)**

Complete the following examples in your textbook (pp. 378 - 380):

- Example 9.6
- Example 9.7
- **Lab Exercises L9.1 .**

**Notes for Example L9.1**

You can look Example 9.8 (page 380) for some clues, and include all necessary subroutines.

**Please copy following codes as the head of your program.**

```
CR          equ  $0D                ; carriage return
LF          equ  $0A                ; line feed
SC0SR       EQU $00CC   ;serial communication status register
SC0DR       EQU $00CF   ;serial communication data register
            org $6000
age         rmb 3       ;store user's two digit age input, e.g.
19, plus the NULL char
height      rmb 5       ;store floating point height, e.g. 5.72,
plus the NULL char
weight      rmb 4       ;store three digit weight, e.g. 155, plus
the NULL char
prompt1     fcc  "Please enter your age"
            db  CR,LF,0   ;a prompt string is terminated by NULL
after a CR and LF
prompt2     fcc  "Please enter your height:"
            db  CR,LF,0
prompt3     fcc  "Please enter your weight in lbs:"
            db  CR,LF,0
out1        fcc "You are "
            db  0
out2        fcc  " years old;"
            db  CR,LF,0
out3        fcc  " inches tall;"
```

```
            db  CR,LF,0
out4        fcc  "You weigh "
            db  0
out5        fcc  " lbs."
            db  CR,LF,0
```

**Your main program should start from here ……**

## APPENDIX A: Addresses of peripherals for 9S12DP256B Version of HC12. Note that these differ from those in your textbook.

| | | |
|---|---|---|
| $0000 - $0017 | CORE (Ports A, B, E, Modes, Inits, Test) | 24 |
| $0018 - $0019 | Reserved | 2 |
| $001A - $001B | Device ID register (PARTID) | 2 |
| $001C - $001F | CORE (MEMSIZ, IRQ, HPRIO) | 4 |
| $0020 - $0027 | Reserved | 8 |
| $0028 - $002F | CORE (Background Debug Mode) | 8 |
| $0030 - $0033 | CORE (PPAGE, Port K) | 4 |
| $0034 - $003F | Clock and Reset Generator (PLL, RTI, COP) | 12 |
| $0040 - $007F | Enhanced Capture Timer 16-bit 8 channels | 64 |
| $0080 - $009F | Analog to Digital Converter 10-bit 8 channels (ATD0) | 32 |
| $00A0 - $00C7 | Pulse Width Modulator 8-bit 8 channels (PWM) | 40 |
| $00C8 - $00CF | Serial Communications Interface 0 (SCI0) | 8 |
| $00D0 - $00D7 | Serial Communications Interface 0 (SCI1) | 8 |
| $00D8 - $00DF | Serial Peripheral Interface (SPI0) | 8 |
| $00E0 - $00E7 | Inter IC Bus | 8 |
| $00E8 - $00EF | Byte Data Link Controller (BDLC) | 8 |
| $00F0 - $00F7 | Serial Peripheral Interface (SPI1) | 8 |
| $00F8 - $00FF | Serial Peripheral Interface (SPI2) | 8 |
| $0100- $010F | Flash Control Register | 16 |
| $0110 - $011B | EEPROM Control Register | 12 |
| $011C - $011F | Reserved | 4 |
| $0120 - $013F | Analog to Digital Converter 10-bit 8 channels (ATD1) | 32 |
| $0140 - $017F | Motorola Scalable Can (CAN0) | 64 |
| $0180 - $01BF | Motorola Scalable Can (CAN1) | 64 |
| $01C0 - $01FF | Motorola Scalable Can (CAN2) | 64 |
| $0200 - $023F | Motorola Scalable Can (CAN3) | 64 |
| $0240 - $027F | Port Integration Module (PIM) | 64 |
| $0280 - $02BF | Motorola Scalable Can (CAN4) | 64 |
| $02C0 - $03FF | Reserved | 320 |
| $0000 - $0FFF | EEPROM array | 4096 |
| $1000 - $3FFF | RAM array | 12288 |
| $4000 - $7FFF | Fixed Flash EEPROM array | |

| | | |
|---|---|---|
| | incl. 0.5K, 1K, 2K or 4K Protected Sector at start | 16384 |
| $8000 - $BFFF | Flash EEPROM Page Window | 16384 |

## APPENDIX B: Details of serial port register addresses for 9S12DP256B Version of HC12. Note that these differ from those in your textbook.

### TABLE 1: MEMORY $00C8 - $00CF: SCI0 SERIAL INTERFACE

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $00C8 | SCIOBDH | R<br>W | 0 | 0 | 0 | SBR12<br>SBR12 | SBR11<br>SBR11 | SBR10<br>SBR10 | SBR9<br>SBR9 | SBR8 |
| $00C9 | SCIOBDL | R/W | SBR7 | SBR6 | SBR5 | SBR4 | SBR3 | SBR2 | SBR1 | SBBR0 |
| $00CA | SCIOCR1 | R/W | LOOPS | SCISWAI | RSRC | M | WAKE | ILT | PE | PT |
| $00CB | SCIOCR2 | R/W | TIE | TCIE | RIE | ILIE | TE | RE | RWU | SBK |
| **$00CC** | SCIOSR1 | R | TDRE | TC | RDRF | IDLE | OR | NF | FE | PF |
| **$00CD** | SCIOSR2 | R<br>W | 0 | 0 | 0 | 0 | 0 | BRK13<br>BRK13 | TXDIR<br>TXDIR | RAF<br>RAF |
| $00CE | SCIODRH | R<br>W | R8 | T8<br>T8 | 0 | 0 | 0 | 0 | 0 | 0 |
| **$00CF** | SCIODRL | R<br>W | R7<br>T7 | R6<br>T6 | R5<br>T5 | R4<br>T4 | R3<br>T3 | R2<br>T2 | R1<br>T1 | R0<br>T0 |

### TABLE 2: MEMORY $00D8 - $00DF: SCI1 SERIAL INTERFACE

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $00C8 | SCIOBDH | R<br>W | 0 | 0 | 0 | SBR12<br>SBR12 | SBR11<br>SBR11 | SBR10<br>SBR10 | SBR9<br>SBR9 | SBR8 |
| $00C9 | SCIOBDH | R/W | SBR7 | SBR6 | SBR5 | SBR4 | SBR3 | SBR2 | SBR1 | SBBR0 |
| $00DA | SCI1CR1 | R/W | LOOPS | SCISWAI | RSRC | M | WAKE | ILT | PE | PT |
| $00DB | SCI1CR2 | R/W | TIE | TCIE | RIE | ILIE | TE | RE | RWU | SBK |
| $00DC | SCI1SR1 | R | TDRE | TC | RDRF | IDLE | OR | NF | FE | PF |
| $00DD | SCI1SR2 | R<br>W | 0 | 0 | 0 | 0 | 0 | BRK13<br>BRK13 | TXDIR<br>TXDIR | RAF<br>RAF |
| $00DE | SCI1DRH | R<br>W | R8 | T8<br>T8 | 0 | 0 | 0 | 0 | 0 | 0 |
| $00DF | SCI1DRL | R<br>W | R7<br>T7 | R6<br>T6 | R5<br>T5 | R4<br>T4 | R3<br>T3 | R2<br>T2 | R1<br>T1 | R0<br>T0 |