Overview

- Parallel vs. Serial Communication
- Synchronous vs. Asynchronous communication
- Asynchronous Serial Transmission
- SCI Registers
Serial vs. Parallel

Serial

- Cheap
- Slow
- Uses one line to transmit and one to receive
  - Modems
  - Printers
  - HC11
- Longer transmission distance

Parallel

- Expensive
- Fast
- Requires multiple lines in order to simultaneously transmit data bits
- Short transmission distance

*Ref (3)
SCI Setup

Tx = Transmit
Rx = Receive

Synchronous vs. Asynchronous

Parallel

Serial

Synchronous
Asynchronous
Synchronous Communication

• Constant transmission of data

• Both clocks of the receiver and transmitter must be synchronized

• No safe-guards against error and/or noise

Asynchronous Communication

• Relatively slow

• Transmission through data “words”

• Continuous data transfer is unnecessary

• Built in safe-guards against noise and/or errors
Data Words

• Information Packets used in asynchronous serial communication

• Composed of:
  – Start bit
  – Data bits
  – Parity bit
  – Stop bits

Start and Stop bits

• Start Bit
  – Used to indicate the beginning of data transmission
  – Marked by a low(0) bit following high bits.

• Stop Bits
  – Signify the end of the data word
  – Noted by two consecutive high(1) bits
Data Bits

- The “Meat” of the data word
- Either 8 or 9 consecutive bits depending on transmission settings
- Transmission starts with LSB and ends with MSB

Parity Bit

- Used to detect transmission errors
- For even parity, bit value is:
  - Low(0) if sum of data bits is even
  - High(1) if sum of data bits is odd
- For odd parity the high and low values are reversed
- Transmitter determines value based on outgoing data
- Receiver compares value to incoming data
- Parity bit cannot detect even number of transmission errors
Data transfer example (Even parity)

Data Byte = #$74 = #01110100

Baud Rate vs. Bit Rate

- Baud rate includes all of the bits in a data word (including start, stop and parity bits)
- Bit rate reveals the actual number of data bits transferred in a given period of time
- Baud rate > Bit rate
Bit/Baud rate Example

Problem: How long would it take to transmit 800 Bytes of data given a continuous transmission with a Baud rate of 9600 and total of 8 data bits per data word?

\[
Bit Rate = \frac{8 \text{ Data bits}}{1 \text{ Start bit} + 8 \text{ Data bits} + 1 \text{ Parity bit} + 2 \text{ Stop bits}} \text{ Baud Rate}
\]

\[
Bit Rate = \frac{2}{3} \text{ Baud Rate} = 6400
\]

\[
\frac{800 \text{ Bytes}}{1 \text{ Bytes}} \cdot \frac{8 \text{ Bits}}{6400 \text{ Bits}} = 1 \text{ sec}
\]

Transmission Errors and Noise

- Typically, a higher baud rate means the transmission is more susceptible to errors and noise
- Parity Bit can be used to detect a faulty transmission
- NF flag in SCI status register helps to detect a noisy signal
  - Each data bit being transmitted is sampled three times around the midpoint of the period
  - If all three samples are not identical, NF flag is set
SCI Registers

- Five Registers
  - Baud-Rate Control Register
  - SCI Control Register 1
  - SCI Control Register 2
  - SCI Status Register
  - SCI Data Register

Baud-Rate Control Register

"How fast can you deliver?"

- Allows user to set a maximum baud rate by picking from a pre-select table.

- Baud register is written to once during initialization of data transmission.

- Both Transmitter and Receiver use the same baud rate.
Baud-Rate Control Register

$102B

RESET: 0 0 0 0 0 U U U

TCLR and RCKB are For test modes only

The Bus Frequency divided by the Division Factor is 16x the highest baud rate:

2E6/(16 X 13) = 9600 Baud

Taken From M68HC11 Reference Manual

SCI Control Registers

"Your Data or Mine?"

The SCI Registers contain details of the way data is to be transferred.

- Set 8 or 9 bit data word
- Set data detection (wakeup) settings
- Set Receive/Transmit enable
### SCI Control Register 1

#### Bit 0 to Bit 7

<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>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>R8</td>
<td>T8</td>
<td>0</td>
<td>M</td>
<td>WAKE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- **R8** – (Read Only) Stores ninth data bit of data received when data word is set to nine bit mode.
- **T8** – Stores ninth data bit of data sent when data word is set to nine bit mode.
- **M** – Controls the character length for transmitter and receiver.
  - 0 = eight data bits
  - 1 = nine data bits
- **Wake** – Determines length of idle time before receiver is ready to receive data.

### SCI Control Register 2

#### Bit 0 to Bit 7

<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>Bit 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, TCIE, RIE, ILIE** bits can be set to request interrupts when the SCI Status register’s TDRE, TC, RDRF, and IDLE bits are set, respectively.
- **TE** enables transmitter
- **RE** enables receiver
- **RWU** enables wakeup feature
- **SBK** enables transmitter to send Synchronous break characters
SCI Status Register
"Flagging down your data"

$102E

<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>Bit 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>0</td>
</tr>
</tbody>
</table>

- TDRE and TC are transmitter flags. TDRE goes high when there is room in the SCDR to send a new character. TC goes high when the transmitter has finished sending and reached an idle state.
- RDRF set high when a character has been received.
- IDLE set high when no data word received for a given time.
- OR set when a data overrun has occurred.
- NF set when noise has been detected reading data word.
- FE set when a logic zero is detected where the stop bit was expected.

SCI Data Register
"The Data Customs House"

$102F

Read: Write:

<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>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>R7</td>
<td>T6</td>
<td>R5</td>
<td>T4</td>
<td>R3</td>
<td>R2</td>
<td>R1</td>
<td>R0</td>
</tr>
<tr>
<td>T7</td>
<td>T6</td>
<td>T5</td>
<td>T4</td>
<td>T3</td>
<td>T2</td>
<td>T1</td>
<td>T0</td>
</tr>
</tbody>
</table>

- All data sent moves through this register one data word at a time
- R0 through R7: Received data (read only)
- T0 through T7: Transmitted data
Transmitting and Receiving with the HC11

- Receiver uses pin 0 of port D
  - When SCI receiver is enabled DDRD0 is set to zero to disable output buffer.

- Transmitter uses pin 1 of port D
  - When SCI transmitter is enabled DDRD1 is set to one to disable input buffer.

- The original state of DDRD is restored once transmitting/receiving has ended.

Transmitting Steps

1. Set Baud rate to equal receiver
2. Set TE (SCCR2) high to enable
3. Set Wake Up mode (SCCR1)
4. TE sends idle character to wake receiver
5. Receiver determines if message is intended for it
6. Load character into SCI Data Register (SCDR)
7. Character placed in shift register and shifted out
8. When TDRE (SCSR) sets back to 1, load another character (both polling and interrupts can be used).
9. Transmission complete (TC in SCSR)
10. Idle line rests at logic 1, RWU goes to 0
Receiving Steps

1. Set Baud rate in Baud register ($102B)
2. Set bit 4 in SCCR1 ($102C) to select 8 or 9 bit characters; set bit 3 to select wake up mode
3. Set bit 2 in SCCR2 ($102D) to enable receiver; set bit 4 to enable interrupt on idle; set bit 5 to enable interrupt when character received or overrun occurs.
4. Read status of receive from SCSR ($102E) Bit 5 will be set when data is received; framing error sets bit 1; noise sets bit 2; overrun sets bit 3; idle sets bit 4
5. Read data received from SCDR ($102F)
6. If 9 bit data format is used, the ninth bit of data will be located in bit 7 of SCCR1 ($102C)

References

www.lion.odu.edu/~wstanley/02-12-02-1.pdf
http://fargo.itp.tsoa.nyu.edu/~dano/physical/comm.html
http://www.quatech.com/support/comm-over-parallel.php (3)

David G. Alciatore and Michael B. Histand,
Introduction to Mechatronics and Measurement Systems

M68HC11 Reference Manual