Si8900 Automatic Baud Rate Detection

1. Introduction

Universal Asynchronous Receiver Transmitters (UARTs) are a decades-old technology used at relatively low baud rates for serial data communication in many applications. The Silicon Labs Si890x Monitoring ADC family includes the Si8900, which has a simple Rx/Tx UART interface. The Si8900 does not have dedicated baud rate selection pins, but instead uses an adaptive baud rate detection algorithm to synchronize the baud rate between the Si8900 and the master controller. The Si8900 automatically detects and adapts to the host's baud rate to a maximum of 520 kbps. This application note explains how the adaptive baud rate detection is implemented and includes proven firmware routines for the user's host to establish initial communication.

2. Si8900 Auto Baud Process

2.1. Operation

The synchronization process (Figure 1) begins with the master processor transmitting (at its own standard or non-standard baud rate) a series of hex bytes to the Si8900, each with a value of 0xAA. Upon receiving the first byte, the Si8900 defaults to its highest baud rate (520 kbps) and compares the received byte to 0xAA. A received byte other than 0xAA indicates that the Si8900 baud rate setting is different from that of the master, causing the Si8900 to incrementally reduce its baud rate, then wait for the next 0xAA host byte. This process continues until a valid 0xAA byte is received, indicating the upper boundary of the correct baud rate. The same process is used to determine the lower boundary of the host's baud rate. With the upper and lower baud rate boundaries known, the Si8900 then places its baud rate setting in the middle of the high and low boundaries. The Si8900 then sends two continuous 0x55 bytes back to the host to terminate the master/slave baud rate synchronization process, and communication between the host and the Si8900 is established. Note that this iterative process also has the benefit of finding the optimal baud rate, thereby minimizing bit error rate.

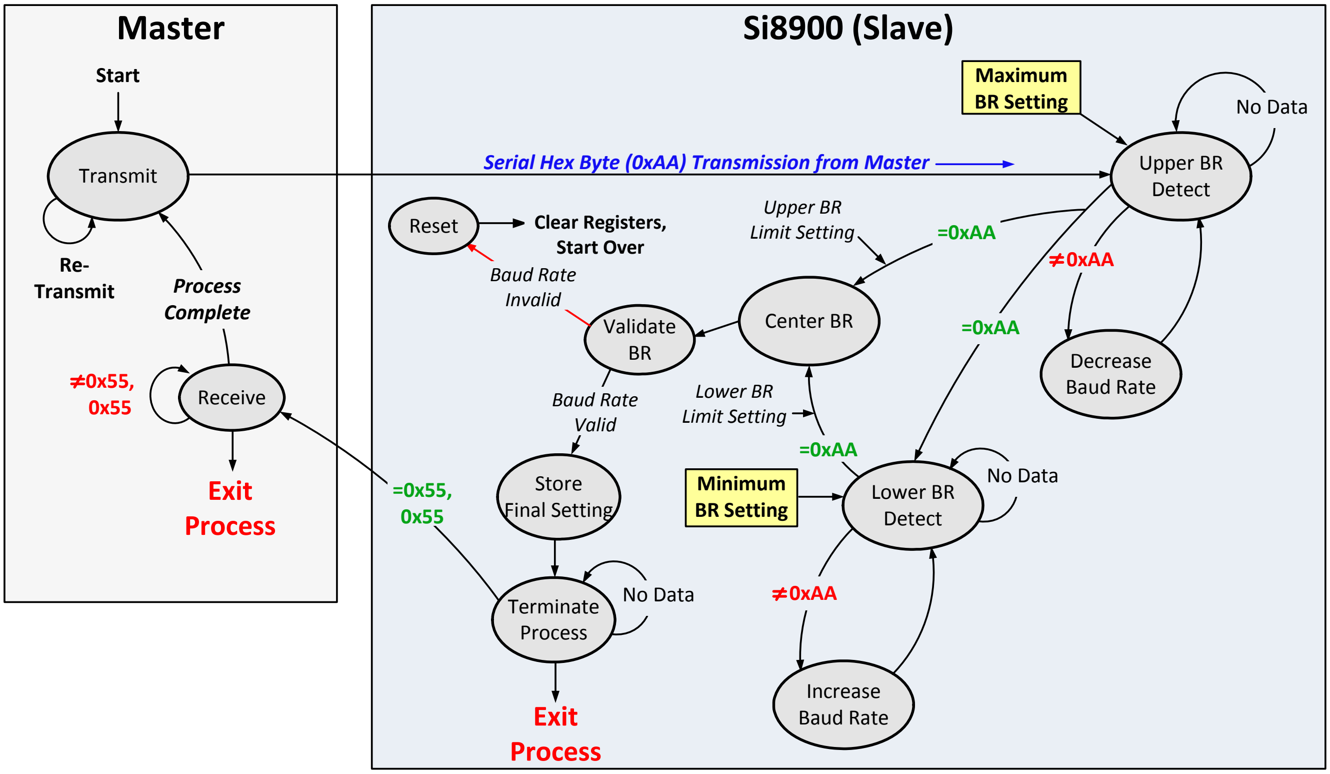


Figure 1. State Diagram for Auto Baud Rate Detection and Setting

2.2. Auto Baud State Machine

The Si8900 internal auto baud state machine is composed of six key processes:

- Upper baud rate limit detection
- Lower baud rate limit detection
- Baud rate calculation
- Baud rate verification
- Baud rate setting
- Baud rate reset

The upper and lower baud rate limits are discovered through the iterative process described in “2.1. Operation”. Once these limits are known, the baud rate is calculated (i.e. centered between the upper and lower limits), and the result verified to ensure the upper and lower baud rate limits are valid. A valid baud rate is then stored in the internal baud rate setting register, at which time two 0x55 bytes are transmitted to the master indicating successful measurement of both limits, and halting the process. An invalid baud rate is discarded, and processing moves to the RESET state, which clears both limit settings and restarts the process.

3. Programming the Master Controller

Auto baud rate detection significantly reduces the bit error rate caused by clock frequency mismatch between the master and the Si8900. Master/slave communication reliability is further increased by virtue of the Si8900 serial protocol, in which the Si8900 returns each command received back to the master for verification enabling the master to re-send a command that was not properly received by the Si8900.

3.1. Master's Auto Baud Program

The auto baud rate program must be instantiated in the master controller to provide the necessary timing reference information for the Si8900 to synchronize to the master. Figure 2 shows the C-language auto baud code that must be programmed into the user's master controller. Note this code must be used as-is and without modifications to ensure proper communication with the Si8900.

```
SBUF = 0xAA;                               // Send the first timing sample to UART
                                           //      transmission buffer SBUF
Code_receive = 0;                          // Correct code receive = 0
Code_confirm = 0;                          // Confirm code receiving = 0
while ((Code receive == 0)||( Code_confirm == 0))
   {                                       // Establish correct communication and confirm
      while ((SCON & 0x01) != 0x01);       // Response received?
      SCON &= 0xFE;                        // Clear receiving indicator
      if ( SBUF == 0x55 )                  // Check two continuous "55" receiving
        {
          if (Code_receive == 1)
            {
              Code_confirm = 1;            // Confirm communication established
            }
          Code_receive = 1;                // Initial communication established
        }
     else
        {
          Code_receive = 0;                //  If receive one in-correctly, clear both
          Code_confirm = 0;                //  Code_receive & Code_confirm
        }
      SBUF = 0xAA;                         //  If not right, keep trying
   }
```

Figure 2. Master Controller Auto Baud Rate Detection and Setting Routine

3.2. Command Program for the Master

Figure 3 shows a C-Language code example for master controller command transmissions to the Si8900.

Note: For more detailed information about the register set and command protocol, see the Si890x data sheet available for download at www.silabs.com/isolation.

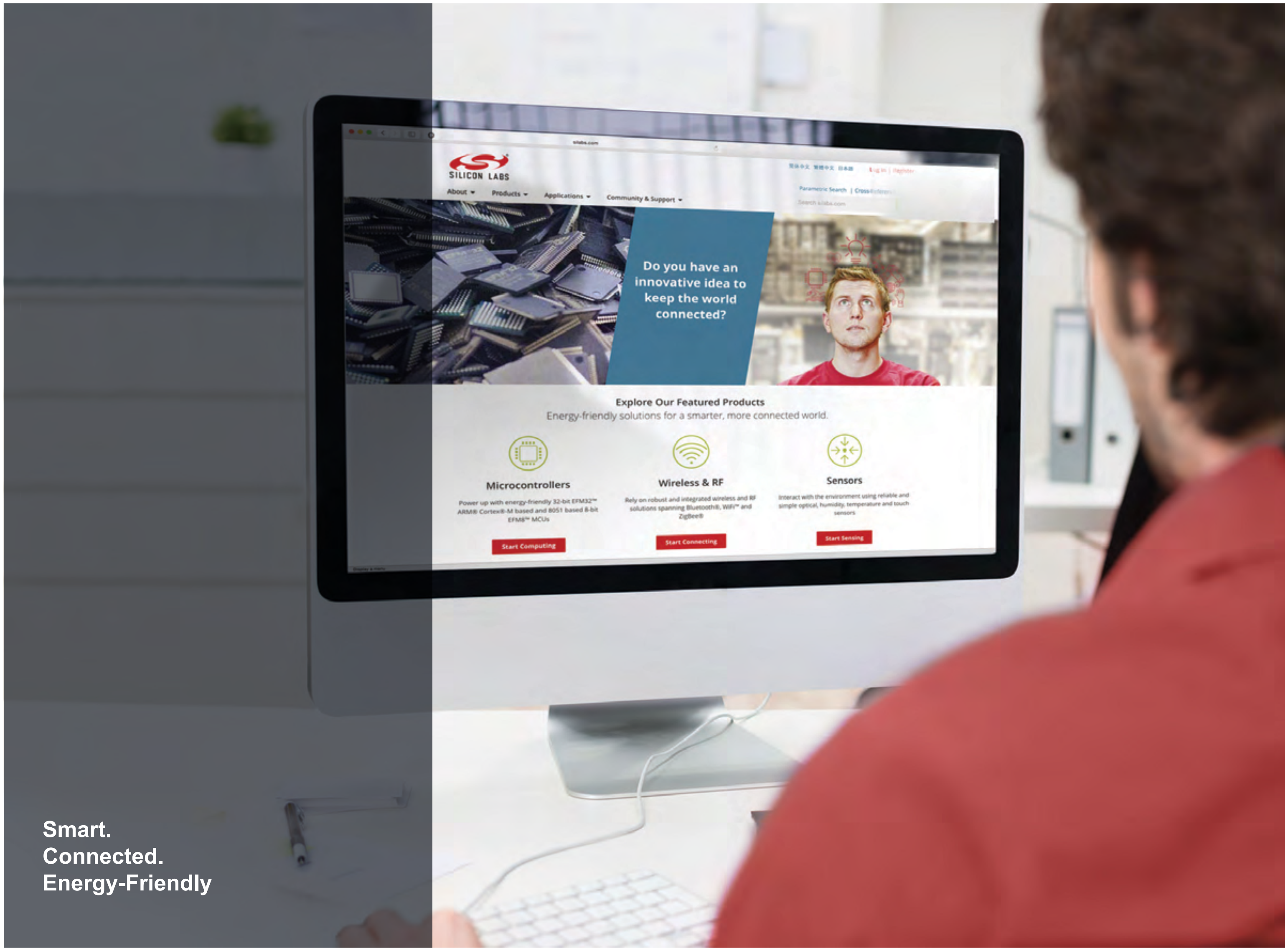
```
CNFG = 0xCB;                          // Command code = 0xCB
SBUF = CNFG;                          // Send command
while ((SCON & 0x01) != 0x01);        // Receive the command back
SCON &= 0xFE;                         // RI1 = 0;
While (SBUF != CNFG)
{
   SBUF = CNFG;                       // Continue sending command until received correctly
   while ((SCON & 0x01) != 0x01);
   SCON &= 0xFE;                      // RI1 = 0;
}
```

Figure 3. Firmware Example for Master Controller Commands to the Si8900

In the example of Figure 3, the configuration register (CNFG) value of “0xCB” specifies a Demand Mode read of ADC input channel 0, using VDD as the reference voltage and a unity gain PGA setting. The master processor issues an ADC read command by transmitting CNFG to the Si8900 receive buffer, SBUF, where serial control register SCON provides communication status information (e.g. byte successfully received). In response, the Si8900 returns a copy of each received command back to the master for verification. If the master verifies the received command is valid, the process continues; if not, the master must reissue the command. (For more information about Si8900 protocol and register set, see the Si890x data sheet.)

4. Related Documents

- Silicon Laboratories, Inc., Si8900/1/2 data sheet
- AN637: Configuring the Si890x Master Controller
- AN638: Isolated ADC Facilitates Low Noise, Precision Measurements



Products
www.silabs.com/products

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com