AN12726

Using SCT to Simulate UART

Rev. 0 — February 2020

Application Note

Contents

1 Introduction

1.1 Background

The LPC800 series of microcontrollers offers a range of low-power, space efficient, low-pin-count options for basic microcontroller applications. Unique to low-end devices, the LPC800 series MCUs include differentiated product features, such as patent-approved State Configurable Timer (SCTimer/PWM), giving embedded developers unprecedented design flexibility.

Due to the low cost, some LPC800 series products have only a limited number of UART (Universal Asynchronous Receiver Transmitter). In some applications, all UARTs need to be used to communicate with external components/modules. In this case, there is no UART left to be used for other applications such as debugging.

SCT is very flexible and it can be used to simulate a UART for solving this limitation.

NOTE

Although this solution is using SCT feature, it increases the loading on the Arm core relative to the real hardware UART.

1.2 SCT

The SCTimer/PWM is a peripheral that is unique to NXP Semiconductors. It can operate like most traditional timers, but also adds a state machine to give it a higher degree of configurability and control. This allows the SCT to be configured as multiple PWMs, a PWM with dead-time control, and a PWM with reset capability, in addition to many other configurations that cannot be duplicated with traditional timers. After the SCTimer/PWM is configured, it can run autonomously from the microcontroller core, unless the SCTimer/PWM interrupt is enabled, which requires the core to service the interrupt.

The state variable is the main feature that distinguishes the SCTimer/PWM from other counter/timer/PWM blocks. Events can be made to occur only in certain states and if certain conditions are satisfied. Events, in turn, can perform some actions. The mechanism is shown in Figure 1.

Figure 1. SCT

For more details about SCT, see AN11538.

1.3 UART

UART is an asynchronous serial communication protocol. The data is not synchronized with a clock, instead the data bits are transmitted with pre-defined baud rate and also appended with synchronization start and stop bits. It uses two-wire interface (TX and RX) to communicate between devices.

The simple UART packet is shown in Figure 2.

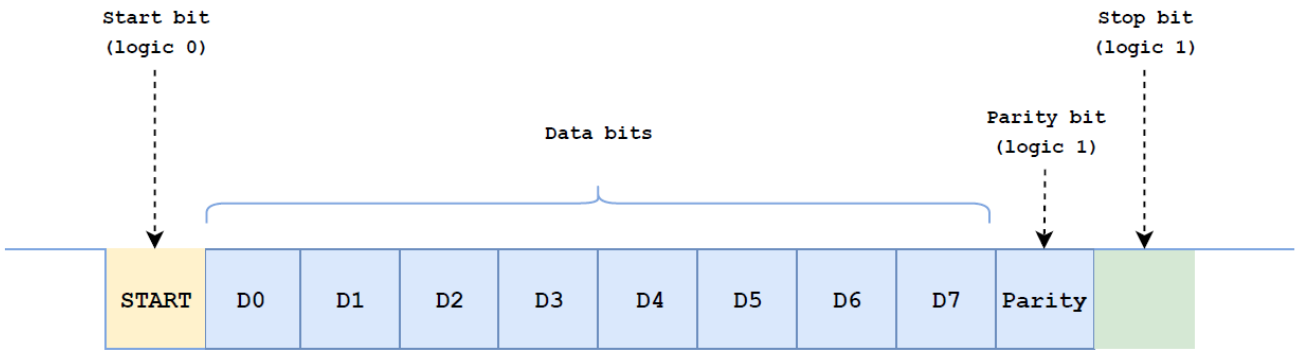

Figure 2. Simple UART packet format

2 Implementation

This solution relies on a single SCT and its ability to toggle a dedicated pin, when needed, while monitoring another pin at the same time.

Suggested software SCT UART supports transfer with format of 1 start bit, 8 data, and 1 stop bit, with no parity bits.

The state machine diagram for SCT UART is as follows. The 32-bit SCT counter is configured as two 16 bit counters named L and H. L counter is used for receiving, H counter is used for sending.

As shown in Figure 3, the event 0 is used for sending. It can be triggered in all states so that the receiving and sending can be at same time. The events 1,2, and 3 are used for receiving.

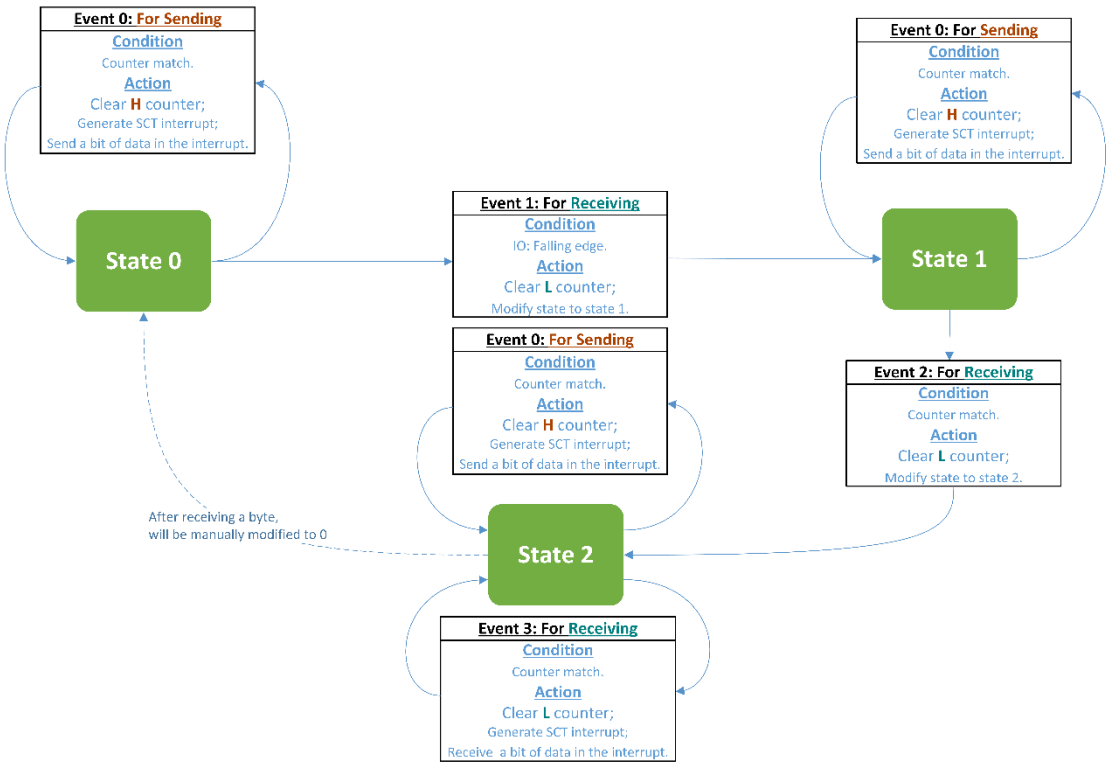

Figure 3. State machine diagram for SCT UART

2.1 UART Send

SCT UART sending is simple. Event 0 generates an SCT period interrupt. The period is based on UART bit rate. The data is sent in the SCT interrupt bit by bit, as shown in Figure 4.

Based on this mechanism, SCT outputs the complete UART data flow.

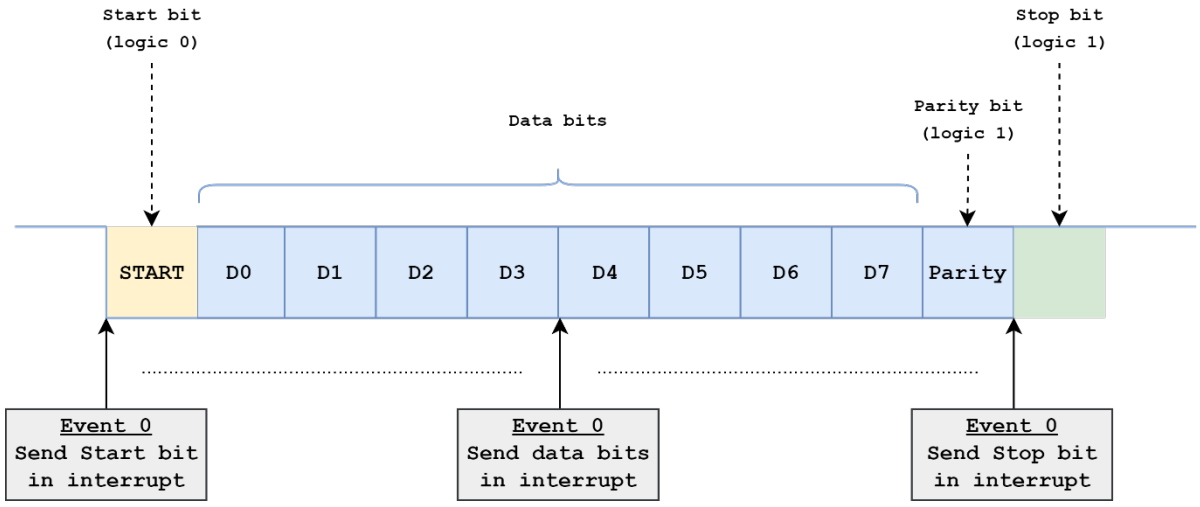

Figure 4. SCT UART sending

2.2 UART Receive

UART data flow starts from the falling edge generated by the start bit. Therefore, to receive UART data, it is necessary to capture the start bit and its falling edge. The Event 1 is used for implementing this function. When the falling edge is captured, use Event 2 to wait for a while to ensure that the signal level is stable. The SCT interrupt is generated periodically by Event 3. In the interrupt, the level of the signal is monitored to decode the data.

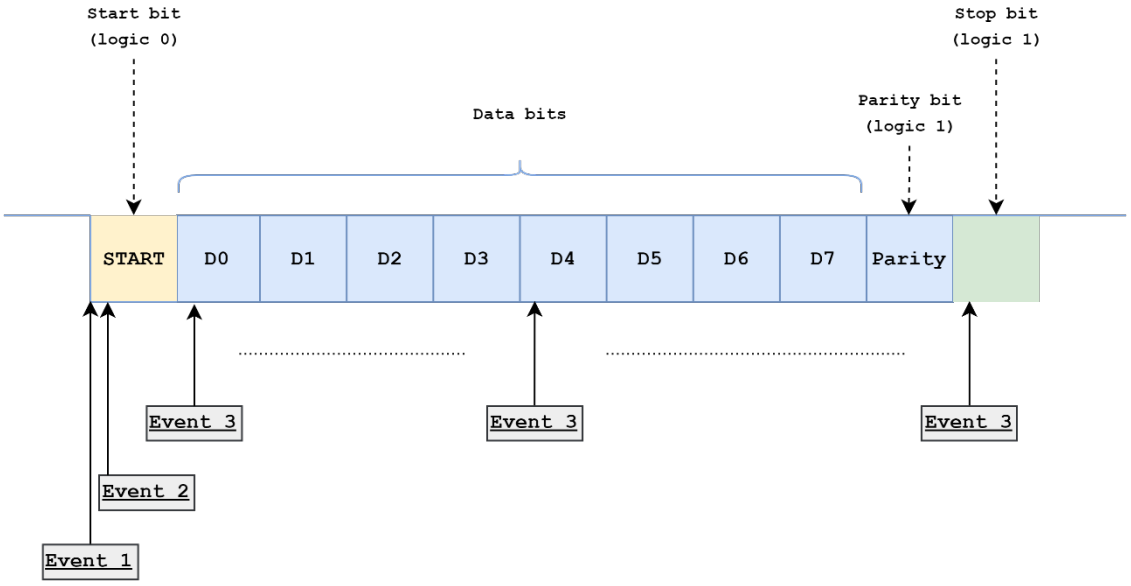

Figure 5. SCT UART receiving

3 Example

3.1 Environment

3.1.1 Hardware environment

- Board
  - LPC824 Xpresso (OM13071)
- Debugger
  - Integrated CMSIS-DAP debugger on the board
- PC software
  - PuTTY or similar UART terminal
- Miscellaneous
  - 1 Micro USB cable
  - USB to UART module
  - PC
- Board Setup
  - Connect the J2 D0(SCT UART TX) to USB to UART module's RX; Connect J2 D1(SCT UART RX) to USB to UART module's TX. Then connect the USB to UART module to PC.
  - Connect the micro USB cable between PC and J3 on the board for loading and running a demo.

3.1.2 Software environment

- Tool chain
  - Keil MDK
- Software package
  - lpc824_sct_uart.zip

3.2 Steps and result

This example demonstrates SCT UART.

The basic steps are as follows:

1. **Compile and download**

   Ensure that the hardware environment is set up as mentioned earlier. Compile the project located in lpc824_sct_uart \Keil_Project and download into the board.

   Open the terminal and configure communication protocol as 9600+8+N+1.

2. **Run**

   Reset the board to run, by pressing the SW3 (reset) button on the board. The data can be entered on PC, the SCT UART echoes the input data, as shown in Figure 6.

Figure 6. Results printed on terminal

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer’s applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: February 2020

Document identifier: AN12726

arm