Freescale Semiconductor
Application Note

AN4427
Rev. 2.0, 02/2012

MM912_634 - LIN Tx Dominant Timeout Implementation

1 Introduction

This document describes the software implementation of an example of Tx LIN Timeout on the MM912_634 product family, for the requirement of a TxD-dominant timeout function for LIN.

Contents

2 LIN Tx-Dominant Timeout Implementation Example

The LIN Tx-Dominant Timeout is not directly implemented into hardware.

To monitor the activities of the Tx function for the embedded LIN physical layer, the user needs to implement various hardware / software functions when designing the MM912_634 products into an application.

The MM912_634 has LIN/SCI/GPIOs capabilities that allow connection of the internal SCI to the LIN physical layer and monitoring through the GPIOs LIN Tx/Rx as described below.

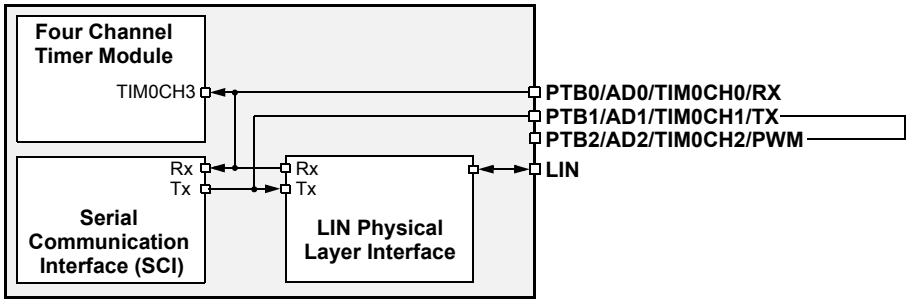

Figure 1. MM912_634 GPIO Mode 3 (observe)

Configuration of the Tx internal signal to PTB1 is done by setting the SERMOD[1..0] bits to 11 in the PTBC2 register.

The PTB1 pin has to be connected to PTB2 on the PCB.

PTB2 should be configured to input capture on Timer Channel 2. It should then trigger a high to low transition (capture time of the first Tx high to low transition - t_{START}).

In Timer Channel 3 in output compare mode, store the value of (t_{START} + a count corresponding to nine bit times at the communication baud rate). Note that "nine bit times" is only a suggestion and depends on the application requirements. When the output compare event occurs, monitor the Tx state through the PTB2 input/output data read capability. PTB2 should report a high level, unless Tx is driven low for abnormal reasons.

If this is the case, the LIN line can be released into a recessive state by disabling the LIN module, through setting the LINEN bit of the LINR register to 0.

When reactivating the LIN after a timeout has occurred, LIN is enabled by setting the LINEN bit of the LINR register to 1. When enabling the LIN, the LIN module is required to be configured to match the usage needed by the application.

3 Sample C Code for LIN Tx Dominant Timeout

This code assumes the LIN is running at 20 kbps and the MM912_634 D2D clock is running at 8.0 MHz.

The D2D clock is divided down according to the prescaler PR[2:0] settings below to give the timer update rate.

PR2	PR1	PR0	Timer Clock
0	0	0	D2D Clock/1
0	0	1	D2D Clock/2
0	1	0	D2D Clock/4
0	1	1	D2D Clock/8
1	0	0	D2D Clock/16
1	0	1	D2D Clock/32
1	1	0	D2D Clock/64
1	1	1	D2D Clock/128

```
//**************  Initialize the LIN bit counter
 int LINBITCOUNT = 29;                    // 62500 Hz --> 16 usec per bit; 9 bit times @ 20kbps -> 29 counts
//**************  Initialize the timer channels
B_PTBC2 = 0x03;                           // Initialize LIN mode 3 - Tx appears on PTB1
B_TIOS = 0x08;                            // set up input capture on TC2, output compare on TC3
B_TCTL2  = 0x20;                          // Set up input capture on PTB2/TC2, trigger on falling edge
B_TCTL1 = 0x0;                            // timers disconnected from output pin logic (default state)
B_TSCR2 = 0x07;                           // set timer prescaler to 128 ( with 8MHz clock, --> 62500 Hz count rate):
B_TFLG1 = 0x0C;                           // clear interrupt flags on timer channels 2 & 3 (C2I)
B_TIE = 0x04;                             // enable interrupt on TC2
EnableInterrupts;                         // standard routine, equivalent to _ASM CLI
B_TSCR1 = 0x80;                           // enable timer
```

A timer interrupt activates the D2D interrupt line and causes a jump to the D2D interrupt handler.

```
// ***************** D2D interrupt routine
 if (B_ISR & 0x0008)                      // check for first input capture event on TC2 (Tx going low)
 {
B_TC3 = B_TC2 + LINBITCOUNT;              // set up output compare on TC3 for 9 bit times ahead
B_TIE = 0x08;                             // enable interrupt on TC3 (C3F of TFLG1) and disable further TC2 interrupts
B_TFLG1 = 0x0C;                           // clear TC2 & TC3 interrupt flags
 }
 else if (B_ISR & 0x0010)                 // if not input capture, check output compare interrupt on TC3
 {
   if(!B_PTB_PTB2)  B_LINR = B_LINR & 0xFB;  // check for PTB2 low: if still low, disable LIN
 }
```

In addition to the code above, the normal LIN message handling software should include some code that will re-enable LIN if the LIN bus fault has cleared, thus allowing further LIN transmissions:

```
 if(B_PTB_PTB2)                           // check for LIN bus high (recessive)
 {
B_LINR_LINEN = 1;                         // if high, re-activate LIN
 }
```

When enabling the LIN, the LIN module is required to be configured to match the usage needed by the application.

4 Revision History

REVISION	DATE	DESCRIPTION OF CHANGES
1.0	01/2012	Initial release.
2.0	02/2012	Added code example.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

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)
www.freescale.com/support

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 China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
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 +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

AN4427
Rev. 2.0
02/2012