

# Start-of-Frame Example Code

## 1 Introduction

This example code demonstrates how to program the MPC8272 to successfully generate Start-of-Frames (SOFs) for a USB host controller. The code includes proper CRC checking, as well as a byte ordering mechanism. The main purpose of the code is to show an example of the minimum programming requirements necessary to generate the SOFs. A few caveats are illustrated which are important to realize when designing a USB host interface with the MPC8272.

### Contents

1. Introduction . . . . .	1
2. Development Environment . . . . .	2
3. Details . . . . .	2
4. File Structure . . . . .	3
5. Verification . . . . .	3

## 2 Development Environment

The following equipment and settings were used to build the project:

- CodeWarrior™ for PPC version 8.6
- CodeWarrior Ethernet TAP
- Windows XP™
- MPC8272ADS
- MPC8272 processor
- Input clock -> 100MHz
- JP8 -> Host mode
- SW5 -> b10100100
- SW2 -> b0010
- HP1662A Logic Analyzer
- USB 1.1 probe from Future Plus Systems
- MPC850FADS

## 3 Details

The MPC8272 is a CPM-based product, which uses microcode to aid in the execution of various protocols. The USB controller also uses microcode to implement the protocol. In this example, the microcode is responsible for forming the Start-of-Frame packet and loading it into the FIFO for transmission. The user is responsible for programming the device to provide a hardware timer which will generate the 1ms time base that the host controller uses for packet transmission. The user is also responsible for generating the correct frame number and its corresponding CRC value, then placing the correct values in the FRAME\_N register.

In this example, the SOF packets were prepared using the USBER[SOF] event register as a software trigger. This particular bit was used because it indicates that a SOF was just received by the controller. The next frame value should be prepared at the beginning of the previous frame reception to provide ample time. At this point, it's important to point out why there's a frame being received in the host controller. The FRAME\_N register is used in both host and device mode with different functionality. Every time the host controller transmits a frame, the SOF packet is immediately received back on the receive line and placed in the FRAME\_N register. If the CRC is valid for the particular frame, the valid bit will be set. After valid frame transmission, the FRAME\_N register will contain only the frame number.

Another caveat about setting up the controller in host mode is that the MPC8272 is designed such that Endpoint 0 emulates the host's communication pipes. All host transmit and receive data happens over Endpoint 0. Therefore, in order to initialize the host controller, Endpoint 0 has to be initialized and at least one TxBD programmed. For this example, EP0's TxBD was minimally initialized to be the only BD in the ring and have a data length of zero.

The USB parameter RAM requires a certain amount of programming. Besides FRAME\_N, the RSTATE value has to be cleared. Endpoint 0's PRAM also has to be minimally initialized. Clear the TSTATE bits, program the upper 16 bits of the IMMR value into HIMMR and set up TBASE are the main points.

## 4 File Structure

Main files:

mpc8272.h	Updated header file for MPC8272
main.c	Complete file for generating SOFs & CRCs
PQ27e_init.cfg	Generic Metrowerks configuration file

## 5 Verification

The code was verified using the HP1662A logic analyzer. The MPC8272ADS was connected upstream to the probe and the probe's Mictors were connected to the logic analyzer. On the downstream side of the probe, an MPC850FADS was used to provide the host with at least one device on the bus. This removes the reset signaling from the analyzer capture and provides a cleaner output. Numerous tests show the SOFs incrementing correctly and that the correct CRC value is being decoded by the analyzer.

### **How to Reach Us:**

#### **Home Page:**

www.freescale.com

#### **email:**

support@freescale.com

#### **USA/Europe or Locations Not Listed:**

Freescale Semiconductor  
Technical Information Center, CH370  
1300 N. Alma School Road  
Chandler, Arizona 85224  
(800) 521-6274  
480-768-2130  
support@freescale.com

#### **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)  
support@freescale.com

#### **Japan:**

Freescale Semiconductor Japan Ltd.  
Headquarters  
ARCO Tower 15F  
1-8-1, Shimo-Meguro, Meguro-ku  
Tokyo 153-0064, Japan  
0120 191014  
+81 2666 8080  
support.japan@freescale.com

#### **Asia/Pacific:**

Freescale Semiconductor Hong Kong Ltd.  
Technical Information Center  
2 Dai King Street  
Tai Po Industrial Estate,  
Tai Po, N.T., Hong Kong  
+800 2666 8080  
support.asia@freescale.com

#### **For Literature Requests Only:**

Freescale Semiconductor  
Literature Distribution Center  
P.O. Box 5405  
Denver, Colorado 80217  
(800) 441-2447  
303-675-2140  
Fax: 303-675-2150  
LDCForFreescaleSemiconductor  
@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor 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 which may be provided in Freescale Semiconductor 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. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™, the Freescale logo and CodeWarrior™ are trademarks of Freescale Semiconductor, Inc.

Windows XP™ is a trademark of Microsoft Corporation.

All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc., 2004, 2005.

Document Number: AN2928

Rev. 0

10/2005