NXP Semiconductors
Application Note

Document Number: AN5293
Rev. 0, 06/2016

Pretended Networking on MPC5748G

by: Maoxin Tian

Contents

1 Overview

The MPC574xG is the first device targeted at Automotive Body Applications in 55-nm technology. It represents a step forward in technology compared to its predecessor, the 90-nm based MPC5646C. The MPC574xG positions itself as the natural choice for MPC5646C based products with an evolved performance.

This document presents an overview of the networking in vehicle. After introducing MPC5748G's operating modes with Pretended Networking feature, software examples are also provided.

2 Introduction

The MPC5748G is a 32-bit Power Architecture® based microcontroller unit (MCU) for automotive applications and builds on the successful legacy of the MPC56xx family of microcontrollers.

There are three application processor cores on this device, as shown in Table 1.

The MPC5748G uses e200 cores based on Power Architecture technology developed for automotive gateway and high-end, centralized body controller module applications. The device contains two 160 MHz e200z4 cores and an 80 MHz e200z2

core to offer a flexible power-performance solution. The MCU's salient features include 6 MB of embedded non-volatile flash memory and 768 KB of embedded SRAM, in addition to support for revolutionary new low power modes. The feature set includes an e200z0-based hardware security module (HSM) exceeding the requirements of the secure hardware extension (SHE) of the Hersteller Initiative Software (HIS) standard, and a selection of communications, analog and timer modules. The device is a SafeAssure solution. It has been developed in accordance with the Automotive Functional Safety Standard (ISO 26262) and is targeted at specific safety functions of at least an Automotive Safety Integrity Level ASIL-B rating.

Table 1. Cores in MPC5748G

Computational Shell		
Core 0	Main Core 0	e200z420
Core 1	Main Core 1	e200z420
Core 2	Peripheral Core 2	e200z210

3 Target applications

The family of devices are designed to address a wide variety of automotive applications including but not limited to the door modules, seat modules, central body, vehicle body controllers, smart junction box, front module applications, high end gateway or combined body controller and gateway applications.

4 Networking in vehicle

Modern vehicles have a large number of ECUs that deliver many functions. Those functions may be distributed among several ECUs with the majority being networked nodes that are connected to one or more system buses.

These ECUs control a range of functions, such as lighting, air conditioning, seats and the engine or transmission. The various bus systems that connect them such as Controller Area Network (CAN), Local Interconnect Network (LIN) and FlexRay form a distributed network within the vehicle.

Vehicle network architectures consist of highly integrated domain controllers, which are interconnected via higher-speed bus systems. The industry trend indicates that Ethernet will be the protocol of choice forming the "backbone" of the domain network and replacing CAN, however, there are some instances of FlexRay. Sub-buses with CAN, FlexRay and LIN will provide connectivity to intelligent nodes within a vehicle sub- domain. Powerful domain controllers will be required to support this highly interconnected architecture.

The table below shows the level of communications interfaces supported by MPC5748G.

Table 2. Communications interfaces supported by MPC5748G

Communications	Bit rate	Description
FlexCAN	CAN2.0	CAN2.0B compliance
	1 MB/s	Mailbox support
	CAN FD	FIFO support
	8 MB/s	Active and passive CAN_FD compliance
		Low-power pretended networking filtering on one node
LINFlex	20 Kbps	LIN protocol version 1.3, 2.0 and 2.1
		1x master/slave, 17x master supporting LIN

Table continues on the next page...

Table 2. Communications interfaces supported by MPC5748G (continued)

Communications	Bit rate	Description
Ethernet	100 MB/s	Supporting (RMII, MII + 1588)
FlexRay	10 MB/s	Supporting FlexRay 2.1
		128 MB/s
SDIO(full speed)	25 MHz	Secure digital input output
SDIO(high speed)	40 MHz	
USB	480 MB/s	1 x on-the-go
		1 x host controller
		ULPI interface
MediaLB	2048 fs ~ 98 MB/s	3-pin and 6-pin interface
		Speed grade up to 2048 fs
SPI	40 MHz	Serial peripheral interface
		Up to four with features for SPI controlled LED drivers

5 Power modes of MPC5748G

MPC5748G has several working modes. Each mode is configurable and can define a policy for energy and processing power management to fit particular system requirements. An application can easily switch from one mode to another depending on the current needs of the system. The modes, such as RESET, DRUN and SAFE, aim to ease the configuration and monitoring of the system. The modes, such as RUN0:3, HALT0, STOP0, LPU_RUN, LPU_STOP and STANDBY, can be configured to meet the application requirements in terms of energy management and available processing power.

Compared with its predecessor, the 90-nm based MPC5646C, MPC5748G allows the application developer to choose some new operation modes, as well as traditional operating modes:

1. RUN
   - Full support for maximum speed/Idd mode
   - All modules/flash powered, clocking optional
2. STOP
   - Main peripherals' state retained
   - LPU peripherals' states retained
   - Cores(e200z2 and e200z4) powered, state retained but clock gated
3. LPU_RUN, LPU_STOP small micro system
   - CAN0, LIN0, SPI0, 10-bit ADC, timer, etc.
   - Reduced frequency execution mode
   - Main cores/platform/flash, phase-locked loop (PLL), etc. All power gated off.
   - Large parts of the SoC inactive
4. STANDBY
   - Required to support 8 kB RAM up to 256 kB of system RAM
   - Also supports wakeup logic, API/RTC, 3kHz SXOSC, 8~40 MHz FXOSC
   - Analog comparator

An overview of resource control possibilities for each mode is shown in the table below. A '√' indicates that a given resource is configurable for a given mode.

Table 3. Working modes and resources

Resource	Mode					
	RESET	SAFE	DRUN	RUN0..3	STOP0	STANDBY0
IRC	on	on	on	on	√	√
					on	on
FXOSC	off	off	√	√	√	√
			off	off	off	off
PLL	off	off	√	√	off	off
			off	off		
SIRC	off	off	√	√	√	√
				off	off	off
SXOSC	off	off	√	√	√	√
			off	off	off	off
FLASH	normal	normal	√	√	√	power-down
			normal	normal	power-down	
MVREG	on	on	on	on	on	off
PDO	off	√	off	off	√	on
		on			off	

6 Pretended Networking

Pretended Networking can be used in networks where the MCU is designed to switch into a low power mode. It could return to normal operation through received CAN frames or signals with/without designed condition, like ID, payload. This feature enables transparent behavior in a given network. MCU can take action at any time and with a very low response time on incoming defined events. It could transparently implement such ECU into an existing network without affecting other ECU in the network.

For MPC5748G, Pretended Networking mode is a special low power mode used to receive wake up messages with low power consumption and supports on FlexCAN0 only. This mode can be selected to operate together with Stop mode. Before entering in one of these low power modes, the PNET_EN bit in CAN_MCR Register must be asserted. Once in low power mode, CHI(Controller Host Interface) sub-block is shut down and CAN_PE sub-block is kept active, so that the Rx receive process is still active to filter incoming messages as defined by the configuration registers (see Pretended Networking Control 1 Register (CAN_CTRL1_PN) in the device Reference Manual). Upon detecting a wake up event, a Wake Up interrupt is issued to the system.

To enter in Pretended Networking mode, FlexCAN must be in normal mode (neither in Freeze, nor in Disable mode). When Stop mode is requested, FlexCAN performs the following steps:

- Waits to be in idle state, or else waits for the third bit of Intermission, and then checks it to be recessive.
- Sets the LPM_ACK bit in CAN_MCR Register.
- Requests the shutdown of the CHI sub-module clock, while keeping the PE sub-module clock active.

Under Pretended Networking mode, FlexCAN keeps itself synchronized with the CAN BUS in Stop mode. FlexCAN can exit Pretended Networking mode by the following ways:

- The CPU removing the Stop Mode request.
- FlexCAN will wait until Bus Idle or third bit of Intermission state to negate CAN_MCR[LPM_ACK] bit.

The above exit ways can be triggered either by the FlexCAN action upon detecting a wake up event and issuing the respective interrupt, or by the CPU itself upon being waked up by others. In consequence, FlexCAN will wait until the Bus Idle state or until the third bit of Intermission state to negate CAN_MCR[LPM_ACK] bit and resume to the Normal mode. This procedure ensures that FlexCAN will be synchronized to the CAN bus after exiting the Pretended Networking mode. The CPU must wait for the CAN_MCR[LPM_ACK] bit to be negated before performing any access to FlexCAN. When PNET_EN bit in CAN_MCR is asserted, the CPU must disable the Self Wake Up feature by configuring SLF_WAK=0 in CAN_MCR register.

7 Software implementation on MPC5748G

This section describes the software implementation on the device. Several functions have been implemented in this application. Development environment is with NXP S32 Design Studio for Power v1.0 and P&E USB MULTILINK. EVBs are standard MPC5748GMB with X-MPC574XG-256DS.

The demo software includes the following functions.

- PLL initialization for clock setting
- LEDs initialization for software flow hints
- FlexCAN initialized to be 100 Kbps with Pretended Networking setting
- Change MCU's working modes to test Pretended Networking

For Pretended Networking test, user could change the code in flexcan.c to achieve the following different functions.

- Pretended Networking wakeup through ID matched only
- Pretended Networking wakeup after N times matched of ID
- Pretended Networking wakeup with ID and payload

In the sample code, it could be waken up through ID matched.

Before MPC5748G EVB is powered-on, one CAN analyzer should connected with P15 on EVB which is for CAN_H/ CAN_L. Download the compiled project and power-off the board.

Power-on MPC5748G EVB again and all the LEDs will be lighted for a second, then all of them are off. After that, the CAN analyzer should receive one frame with ID = 0x007 and data = {"LPU_RUN "}. After LED of DS2 is lighted, MPC5748G enters STOP mode. MPC5748G will be waken up through one CAN frame which contains the ID as defined.

Current of MPC5748G under different working mode can be measured through J18, J19 and J21 if MPC5784G is powered through 5 V on X-MPC574XG-256DS, or, J20 and J22 if MCP5748G is powered through 3.3 V on X-MPC574XG-256DS.

8 Conclusion

More and more ECUs are used in vehicle. Low power strategy for them has raised their use where the battery power is limited, especially in pure electric cars. Pretended Networking meets the trends and such ECU can be transparently implemented into an existing network without affecting other ECUs in the network. It offers less risk and higher flexibility of implementation. MPC5748G is the newest generation of NXP's microcontroller families and could offer Pretended Networking which achieve the most efficient energy management on CAN.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number AN5293
Revision 0, 06/2016