Freescale Semiconductor
Application Note

Document Number: AN3265
Rev. 1, 05/2007

Incorporating an MFR4310 Node in an MFR4200 Network

by: Jaromir Jasik
System Application Engineer, Roznov, Czech Republic

and: David Paterson
Application Engineer, EKB, Scotland

1 Introduction

Freescale Semiconductor offers a range of standalone FlexRay communication controllers (MFR4xxx) that can be interfaced with 16-bit and 32-bit microcontrollers (MCU). This application note describes the software configuration requirements necessary to incorporate an MFR4310 node in an MFR4200 network.

2 Objective

The aim of the document is to demonstrate the simplicity of the software configuration between the MFR4310 and the MFR4200 and to provide an application example of the software used to configure both controllers for successful inter-operation. The information contained can help you quickly incorporate a fully functional MFR4310 FlexRay node into an existing MFR4200 network.

Evaluation boards with software are available from Freescale, to assist in the development of FlexRay applications. (See http://www.freescale.com/flexray.)

Contents

1 Introduction 1
2 Objective 1
3 FlexRay Initialization Sequence 2
3.1 MFR4200 Initialization. 2
3.2 MFR4310 Initialization. 3
4 Cluster Configuration. 3
4.1 System and Application Data 3
4.2 MFR4200 Configuration 6
4.3 MFR4310 Protocol Configuration 7
4.4 MFR4310 Configuration 9
5 Conclusions 10
6 References 10

3 FlexRay Initialization Sequence

The initialization sequences for both the MFR4200 and the MFR4310 are slighty different as both FlexRay communication controllers were developed at different times. See Section 3.1, "MFR4200 Initialization" and Section 3.2, "MFR4310 Initialization" for details of the initialization sequences.

3.1 MFR4200 Initialization

1. Wait for the controller to initialize
   a) The MNR contains 0x0000 while the controller is initializing after leaving the hard reset state.
   b) Only after this initialization is completed, does it contain the value 0x0815.
   c) After the controller leaves the hard reset state, the host must wait until the initialization is completed before reading or writing to the controller.
   d) The initialization takes 1025 cycles (CC_CLK) after de-assertion of the hard reset.
2. Configure FlexRay Block when the CC is in the CONFIG state. The Module Configuration Registers (MCR0 and MCR1) can be set.
3. Enable the FlexRay Block
   a) Set the CONFIG bit in the Module Configuration Register 0 (MCR0)
   b) The FlexRay Block enters the configuration state.
4. Configure the Protocol Engine
   a) Write to the appropriate registers to set all protocol parameters.
5. Configure all Message Buffers and FIFOs as follows
   a) After the CC has entered the configuration state, the host configures the message buffers.
   b) The host configures the BUFCSnR register (CFG, IENA, CHA, CHB, BT, CCFE, and TT bits)
   c) The host configures the CCFnR register of the message buffer.
   d) The host locks a message buffer (in accordance with the MFR4200 data sheet).
   e) After the message buffer is locked, the host configures the remaining configuration fields of the message buffer in accordance with the configuration principles.
   f) The host unlocks the message buffer.
6. The host requests the CC to leave CONFIG state by clearing the CONFIG bit in MCR0 register.
7. Configuration becomes active
   a) After the CC leaves the configuration state.
   b) The CC goes to the normal mode of operation

After this sequence, the FlexRay Block is configured as a FlexRay node and is ready to be integrated into the FlexRay cluster.

3.2 MFR4310 Initialization

The following is an initialization sequence applicable to the FlexRay Block after a hard reset.

1. Configure FlexRay Block
   a) Set the control bits in the Module Configuration Register (MCR)
2. Enable the FlexRay Block
   a) Set the MEN bit in the Module Configuration Register (MCR)
   b) The FlexRay Block enters the Normal Mode
3. Configure the Protocol Engine
   a) Write the CONFIG command into the POCCMD field of the Protocol Operation Control Register (POCR)
   b) After each command written into PCR0 you should "hand-shake" for the required state to occur by observing the POCR.
   c) Write to the PCR[0:30] registers to set all protocol parameters.
4. Configure the Message Buffers and FIFOs
   a) Set the number of message buffers used and the message buffer segmentation in the Message Buffer Segment Size and Utilization Register (MBSSUTR)
   b) Define the message buffer data size in the Message Buffer Data Size Register (MBDSR)
   c) Configure each message buffer by setting the configuration values in the Message Buffer Configuration, Control, Status Registers (MBCCSRn), Message Buffer Cycle Counter Filter Registers (MBCCFRn), Message Buffer Frame ID Registers (MBFIDRn), and Message Buffer Index Registers (MBIDXRn). The message buffer header area must also be configured.
   d) Configure the receive FIFOs if required by the application.
5. Start the FlexRay Block as a FlexRay node
   a) Write the READY command into the POCCMD field of the Protocol Operation Control Register (POCR)
   b) Wait for the READY state by checking PSR0.

After this sequence, the FlexRay Block is configured as a FlexRay node and is ready to be integrated into the FlexRay cluster.

4 Cluster Configuration

This section details the protocol configuration parameters and register configuration parameters for both the MFR4200 and the MFR4310. This is based on an application example and is used as an example to show how to incorporate an MFR4310 node into an MFR4200 network.

4.1 System and Application Data

Table 1 and Table 2 show protocol related and relevant parameters and communication system specific data common to both the MFR4200 and the MFR4310 controllers. This data is then then taken into Section 4.2, “MFR4200 Configuration” where they are implemented into specific MFR4200 registers

Table 1. System and Application Data

Name	Value[dec]
gBitRate[Mbit/s]	10.00
pControllerFrequency[MHz]	40
gdCycle[ms]	5.00
gNumberOfStaticSlots[-]	16
gPayloadLengthStatic[uint16]	8
pPayloadLengthDynMax[uint16]	8
number of nodes[-]	4
pDynamicSegmentEnable	YES
Symbol Window	YES

Table 2. Communication System Specific Data

Name	Value[dec]
gdMacrotickNom[µs]	1.00
gMacroPerCycle[MT]	5000
pMicroPerCycle[µT]	200000
pMicroPerMacroNom[µT/MT]	40
pdMicrotick[ns]	25.00
gdSampleClockPeriod[ns]	12.50
gdStaticSlot[MT]	34
gdActionPointOffset[MT]	3
gdDynamicSegment[MT]	3942
gNumberOfMinislots[-]	657
gdMinislot[MT]	6
pLatestTx[Minislot]	651
pLatestTx[MT]	4452
gdMinislotActionPointOffset[MT]	3
vdActionPointDifference[MT]	0
gdDynamicSlotIdlePhase[Minislots]	1
gdSymbolWindow[MT]	13
gdCASRxLowMax[gdBit]	83
gdNIT[MT]	501
gOffsetCorrectionStart[MT]	4600
pdMaxDrift[µT]	601
gMacroInitialOffset_A[MT]	5
gMacroInitialOffset_B[MT]	5
pMicroInitialOffset_A[µT]	23
pMicroInitialOffset_B[µT]	23

Table 2. Communication System Specific Data (continued)

Name	Value[dec]
pdListenTimeout[µT]	401202
gListenNoise[-]	3
gColdStartAttempts[-]	10
gdMaxInitializationError[µs]	3.00
pdAcceptedStartupRange[µT]	250
pOffsetCorrectionOut[µT]	55
pRateCorrectionOut[µT]	240
pClusterDriftDamping[µT]	1
gSyncNodeMax[-]	4
pExternOffsetCorrection[µT]	0
pExternRateCorrection[µT]	0
gdBit[ns]	100
Bit duration [µT]	4
pDecodingCorrection[µT]	56
gdBitMax[ns]	100.15
gdBitMin[ns]	99.85
pdTSSReceiver[gdBit]	12
gdTSSTransmitter[gdBit]	11
gMaxWithoutClockCorrectionPassive	3
gMaxWithoutClockCorrectionFatal	6
gdWakeupSymbolRxIdle[gdBit]	57
gdWakeupSymbolRxLow[gdBit]	50
gdWakeupSymbolRxWindow[gdBit]	301
gdWakeupSymbolTxIdle[gdBit]	180
gdWakeupSymbolTxLow[gdBit]	60
gNetworkManagementVectorLength[uint8]	8
pAllowHaltDueToClock[boolean]	FALSE
pAllowPassiveToActive[cyclepairs]	20
pDelayCompensation[A][µT]	1
pDelayCompensation[B][µT]	1
pWakeupChannel[-]	Application dependant
pWakeupPattern[-]	16

4.2 MFR4200 Configuration

The values defined in Section 4.1, “System and Application Data” are now converted into specific MFR4200 registers, as shown in Table 3.

Table 3. MFR4200 Configuration

Register	Name	Value[hex]
NSSR	Number of Static Slots Register	0x0010
SPLR	Static Payload Length Register	0x0008
MPLDR	Maximum Payload Length Dynamic Register	0x0008
BDR	Bit Duration Register	0x0004
IDLR	Idle Detection Length Register	0x0001
NMLR	Nominal Macrotick Length Register	0x0028
SSLR	Static Slot Length Register	0x0022
CLR	Cycle Length Register	0x1388
MPCLR	Microticks per Cycle Low Register	0xF9B8
MPCHR	Microticks per Cycle High Register	0x0002
MCLDAR	Maximum Cycle Length Deviation Register	0x0259
TSSLR	Transmit Start Sequence Length Register	0x0B0C
SWCR	Symbol Window Configuration Register	0x1185
NITCR	Network Idle Time Configuration Register	0x1192
CSMR	Cold Start Maximum Register	0x000A
MSFR	Maximum Sync Frames Register	0x0004
LDTSR	Latest Dynamic Transmission Start Register	0x1164
MSLR	Minislot Length Register	0x0006
MSAPOR	Minislot Action Point Offset Register	0x0003
SSAPOR	Static Slot Action Point Offset Register	0x0003
MOCWCFR	Maximum Odd Cycles Without Clock Correction Fatal Register	0x0006
DCAR	Delay Compensation Channel A Register	0x0001
DCBR	Delay Compensation Channel B Register	0x0001
LNLR	Listen timeout with Noise Length Register	0x0003
MOCWCPR	Maximum Odd Cycles Without Clock Correction Passive Register	0x0003
MOCR	Maximum Offset Correction Register	0x0037
MRCR	Maximum Rate Correction Register	0x00F0
CDDR	Cluster Drift Damping Register	0x0001
SOCCTR	Start of Offset Correction Cycle Time Register	0x11F8
WUSTXIR	Wakeup Symbol TX Idle Register	0x00B4
WUSTXLR	Wakeup Symbol TX Low Register	0x003C
EOCR	External Offset Correction Register	0x0000
ERCR	External Rate Correction Register	0x0000

4.3 MFR4310 Protocol Configuration

This section shows all protocol configuration paramaters and specific values for each. These are then taken into Section 4.4, “MFR4310 Configuration” where they are implemented into specific MFR4310 registers.

Table 4. MFR4310 Protocol Configuration Register Fields (Sheet 1 of 2)

Protocol Configuration Register Fields Name	Value[dec]
coldstart_attempts	10
action_point_offset	2
cas_rx_low_max	82
dynamic_slot_idle_phase	1
minislot_action_point_offset	2
minislot_after_action_point	2
static_slot_length	34
static_slot_after_action_point	30
symbol_window_exists	TRUE
symbol_window_after_action_point	9
tss_transmitter	11
wakeup_symbol_rx_idle	57
wakeup_symbol_rx_low	50
wakeup_symbol_rx_window	301
wakeup_symbol_tx_idle	180
wakeup_symbol_tx_low	60
noise_listen_timeout	1203605
macro_initial_offset_a	5
macro_initial_offset_b	5
macro_per_cycle	5000
macro_after_first_static_slot	4966
macro_after_offset_correction	400
max_without_clock_correction_fatal	6
max_without_clock_correction_passive	3
minislot_exists	TRUE
minislots_max	656
number_of_static_slots	16
offset_correction_start	4600
payload_length_static	8
max_payload_length_dynamic	8
first_minislot_action_point_offset	2
allow_halt_due_to_clock	FALSE
allow_passive_to_active	20

Table 4. MFR4310 Protocol Configuration Register Fields (Sheet 2 of 2)

Protocol Configuration Register Fields Name	Value[dec]
cluster_drift_damping	1
comp_accepted_startup_range_a	249
comp_accepted_startup_range_b	249
listen_timeout	401201
key_slot_id	Application dependant
key_slot_used_for_startup	Application dependant
key_slot_used_for_sync	Application dependant
latest_tx	6
sync_node_max	4
micro_initial_offset_a	23
micro_initial_offset_b	23
micro_per_cycle	200000
micro_per_cycle_min	199399
micro_per_cycle_max	200601
micro_per_macro_nom_half	20
offset_correction_out	55
rate_correction_out	240
single_slot_enabled	FALSE
wakeup_channel	Application dependant
wakeup_pattern	16
decoding_correction_a	59
decoding_correction_b	59
filter_bypass	Application dependant
key_slot_header_crc	Application dependant
extern_offset_correction	0
extern_rate_correction	0

4.4 MFR4310 Configuration

The values defined in Section 4.3, “MFR4310 Protocol Configuration” are now converted into specific MFR4310 registers, as shown in Table 5.

Table 5. MFR4310 Configuration

Register	Name	Value[hex]
NMVLR	Network Management Vector Length Register	0x0008
PCR0	Protocol Configuration Register 0	0x0822
PCR1	Protocol Configuration Register 1	0x1366
PCR2	Protocol Configuration Register 2	0x0810
PCR3	Protocol Configuration Register 3	0xC84A
PCR4	Protocol Configuration Register 4	0xA52D
PCR5	Protocol Configuration Register 5	0xBF39
PCR6	Protocol Configuration Register 6	0x0485
PCR7	Protocol Configuration Register 7	0x1D94
PCR8	Protocol Configuration Register 8	0x63B4
PCR9	Protocol Configuration Register 9	0xC037
PCR10	Protocol Configuration Register 10	0x1388
PCR11	Protocol Configuration Register 11	0xD1F8
PCR12	Protocol Configuration Register 12	0xA2F1
PCR13	Protocol Configuration Register 13	0x081E
PCR14	Protocol Configuration Register 14	0x1E06
PCR15	Protocol Configuration Register 15	0x1F31
PCR16	Protocol Configuration Register 16	0x0A12
PCR17	Protocol Configuration Register 17	0x5D95
PCR18	Protocol Configuration Register 18	0x4003
PCR19	Protocol Configuration Register 19	0x1D88
PCR20	Protocol Configuration Register 20	0x1717
PCR21	Protocol Configuration Register 21	0x0006
PCR22	Protocol Configuration Register 22	0x0F93
PCR23	Protocol Configuration Register 23	0x0D40
PCR24	Protocol Configuration Register 24	0x0883
PCR25	Protocol Configuration Register 25	0x0AE7
PCR26	Protocol Configuration Register 26	0x0F93
PCR27	Protocol Configuration Register 27	0x0F99
PCR28	Protocol Configuration Register 28	0x4190
PCR29	Protocol Configuration Register 29	0x0290
PCR30	Protocol Configuration Register 30	0x0004

5 Conclusions

An MFR4310 node can be successfully incorporated into an MFR4200 network and both FlexRay communication controllers can operate together.

The application note is based on an application example to show the simplicity of incorporating an MFR4310 node into an MFR4200 network.

6 References

1. MFR4200 FlexRay Communication Controller Data Sheet (MFR4200)
2. MFR4310 FlexRay Communication Controller Reference Manual (MFR4310RM)
3. FlexRay Communications System Protocol Specification Version 1.1
4. FlexRay Communications System Protocol Specification Version 2.1a

These documents are available on the Freescale Semiconductor web site at http://www.freescale.com or at http://www.flexray.com.

More information on Freescale’s FlexRay products can be found at http://www.freescale.com/flexray.

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
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 or +81 3 5437 9125
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
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Document Number: AN3265
Rev. 1
05/2007

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 that 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.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-compliant and/or non-Pb-free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to http://www.freescale.com/epp.

freescale™
semiconductor