

MNM1221
100 Mbps Communication ASIC
For RTEX

Example Code Descriptions
for Master

Panasonic Industry Co., Ltd.

Revision History

Revision	Date	Change Description
0.1	2004/5/6	Initial Release (Preliminary)
1	2012/2/10	P1 Changed title and company name. P4 Clarified the address-unit. P6 Corrected the notes. P11 Changed CPU example from SH7065 to SH7216. P14 Renamed from “Servo drive” to “Generic slave”. Renamed from “Input module” to “Input slave”. Renamed from “Output module” to “Output slave”.
2	2023/3/31	P1 Changed the company name.

Table of Contents

INTRODUCTION.....	4
OVERVIEW.....	4
PROGRAMMING LANGUAGE.....	4
FILES	4
PRECAUTION	4
SYSTEM STRUCTURE	5
HARDWARE COMPOSITION EXAMPLE.....	5
MNM1221 BLOCK DIAGRAM.....	5
TX AND RX MEMORY IN MNM1221	6
MEMORY MAP OF MNM1221	7
ASSIGNMENT ON TX AND RX MEMORY.....	8
STATE TRANSITION OF MNM1221	9
CONDITIONS ON USING THE EXAMPLE CODE	10
PRECONDITION	10
CONNECTION OF TIMING SIGNALS	11
DESCRIPTIONS FOR THE EXAMPLE CODE	12
DATA	12
<i>Transmitted and Received Data</i>	13
<i>Slave Information Table</i>	14
EXAMPLE OF REARRANGING THE DATA.....	15
FLOW CHART.....	16
<i>Receiving Interrupt</i>	16
<i>Timer Interrupt</i>	17

Introduction

Overview

This example code contains basic process to enable MNM1221 to operate as Master mode. The main body is the startup routine from releasing the reset to cyclic transmitting, and the data exchange routine between the buffer on CPU RAM and TX, RX memory in MNM1221.

The code for the application is separately necessary because this example code does not include it.

Programming Language

C language based on ANSI

Files

File Name	Contents
mnmm1221_m.h	Header for registers definition
mnmm1221_m.c	Source code

Precaution

The modification of this code is required in order to meet respective user's system. First of all, the following portion of "mnmm1221_m.h" must be configured.

- The assigned address of MNM1221. (The address is byte-unit.)
- The data bus size of either 16-bit-wide or 32-bit-wide.

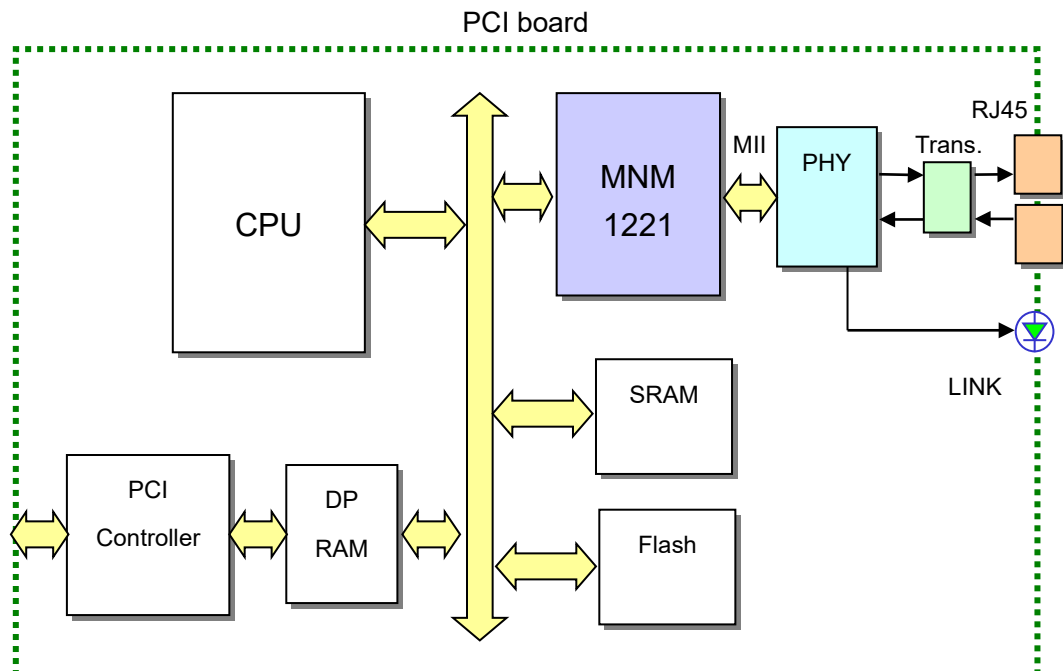
These details and other points to be changed are described in the source code file as a comment. Please refer them carefully.

Attention:

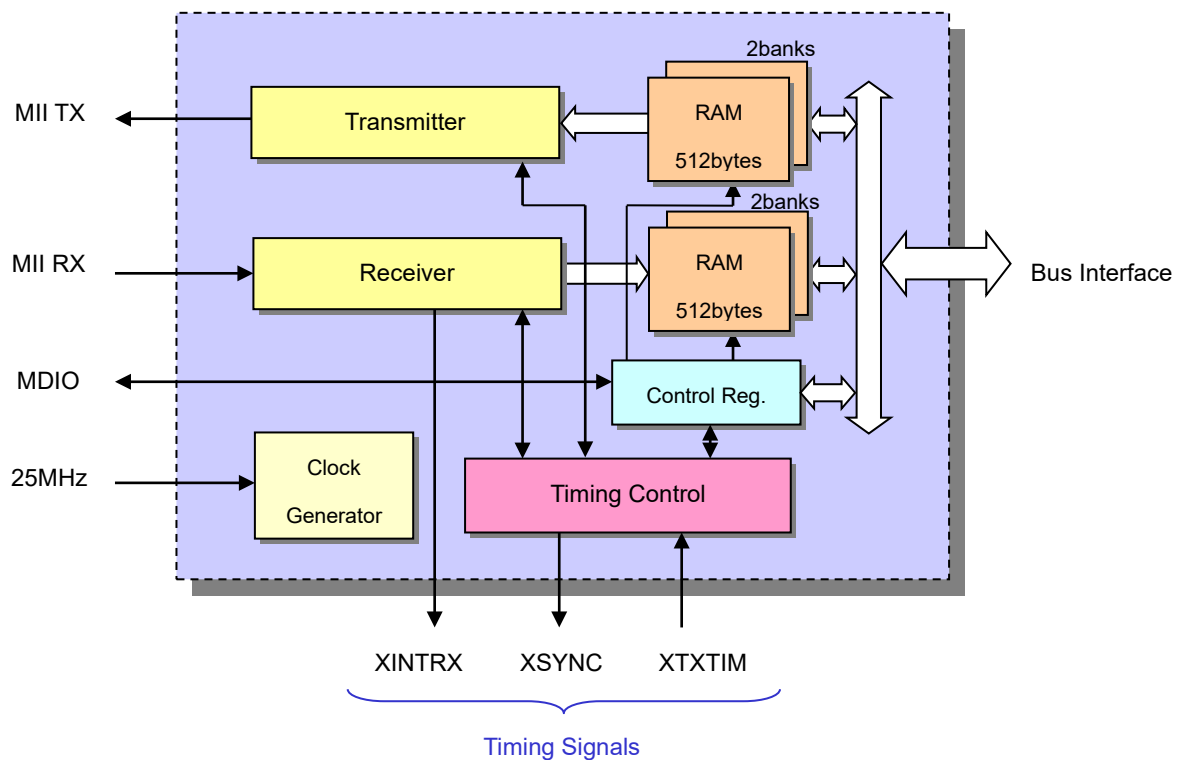
This example source code helps the understanding of MNM1221, however, it is impossible to assure the proper operation under the user systems. We cannot have the responsibility when damage appears by the problem that may be contained in the code.

System Structure

Hardware Composition Example

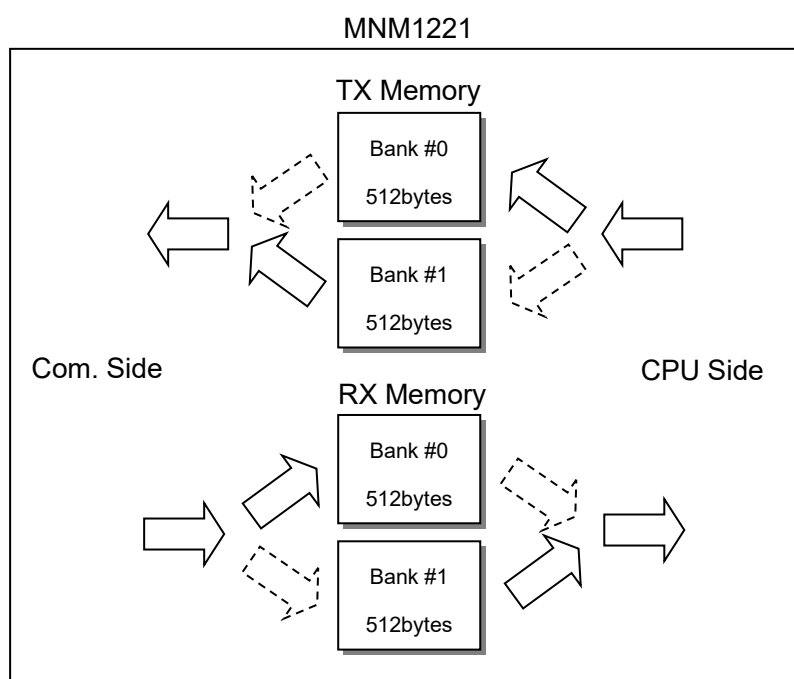


MNM1221 Block Diagram



TX and RX Memory in MNM1221

A MNM1221 transmission buffer memory (TX memory) and reception buffer memory (RX memory) are respectively composed of 2 banks. To avoid conflict of data access, one bank is dedicated to the external CPU and another bank is dedicated to the internal communication module. And such assignment of the 2 banks is switched alternately.



Switching of banks:

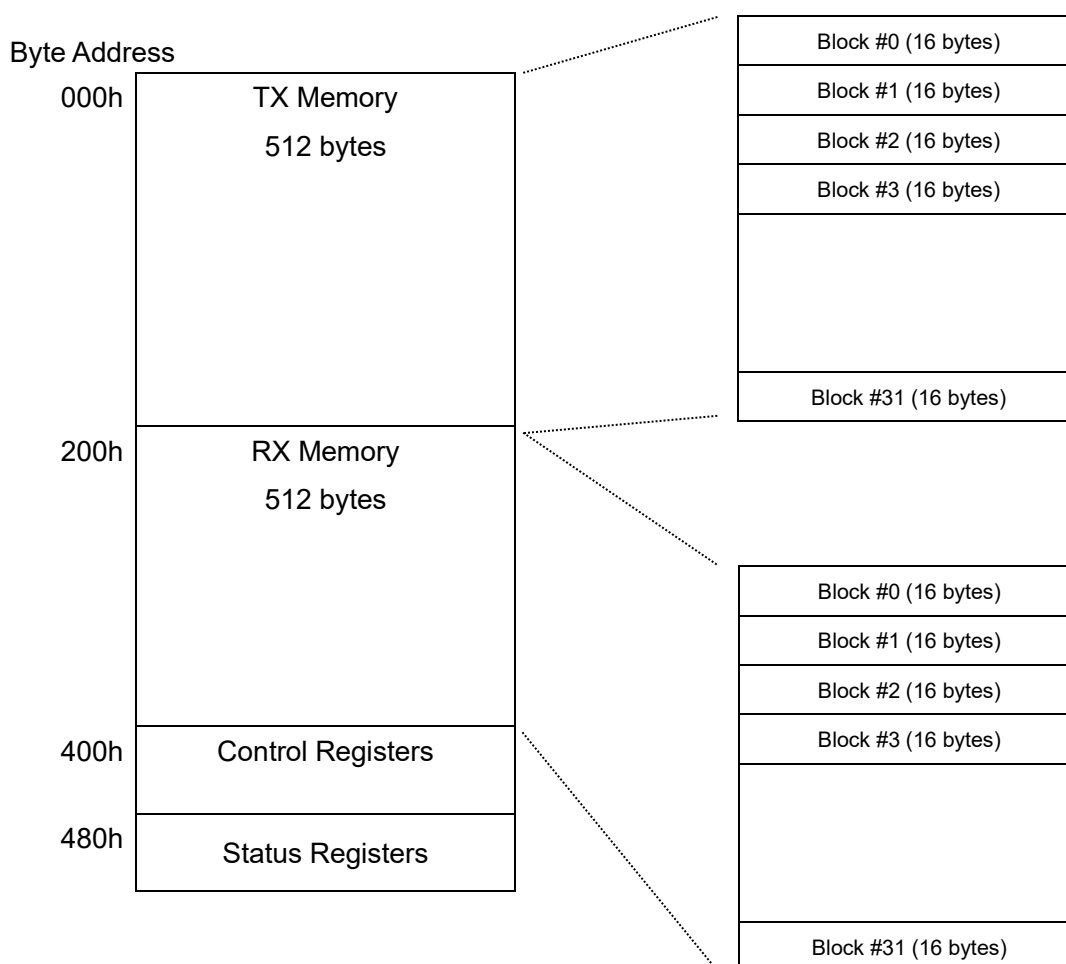
	The trigger of switching	Deferred switching
TX memory	M_TXMEM_SW = 1 (This register is self cleared)	Being transmitted
RX memory	Receiving completion	M_RXMEM_HOLD = 1

Notes:

- For transmitting, M_TXMEM_SW must be set to 1 after writing the data.
 - At least once before reading the received data, M_RXMEM_HOLD must be set to 0 to allow switching the bank.
- In the example code, the following receiving procedure has done in order to get non-error data only in the minimum delay under noisy environment.
- Plural communication cycle per one NC calculation cycle.
 - Normally, M_RXMEM_HOLD = 1(Deferred switching).
 - If non-error data is confirmed at receiving, M_RXMEM_HOLD = 0(Released and switched).
 - The RX memory is read in the NC calculation routine.

Memory Map of MNM1221

Regardless of the bank switching, TX and RX memories are accessed with the same address. Each memory is divided into 32 blocks and one block consists of 16 bytes. Normally, one block corresponds to one Slave node.



Byte order:

“Little Endian” i.e. Lower data is in lower address.

Byte Address	4n + 3		4n + 2		4n + 1		4n + 0	
Data	31	24	23	16	15	8	7	0

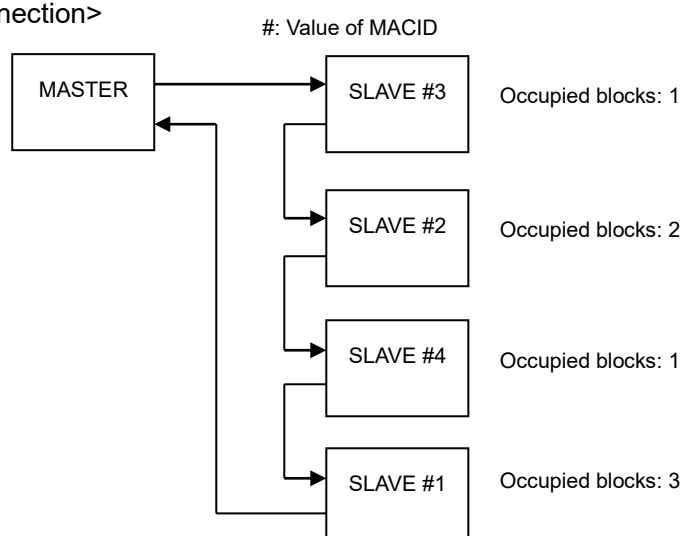
(n: 0, 1, 2, 3, ...)

Assignment on TX and RX Memory

The data of Slave nodes are arranged to on TX and RX memory in order of the cable connection. Since the arranging is from the lower block on the memory, the upper block area becomes empty as long as the sum of used blocks is less than 32.

The below figures show an example, there are Slave nodes that occupies plural blocks. But, normally, Slave occupies only one block.

<Structure of Connection>



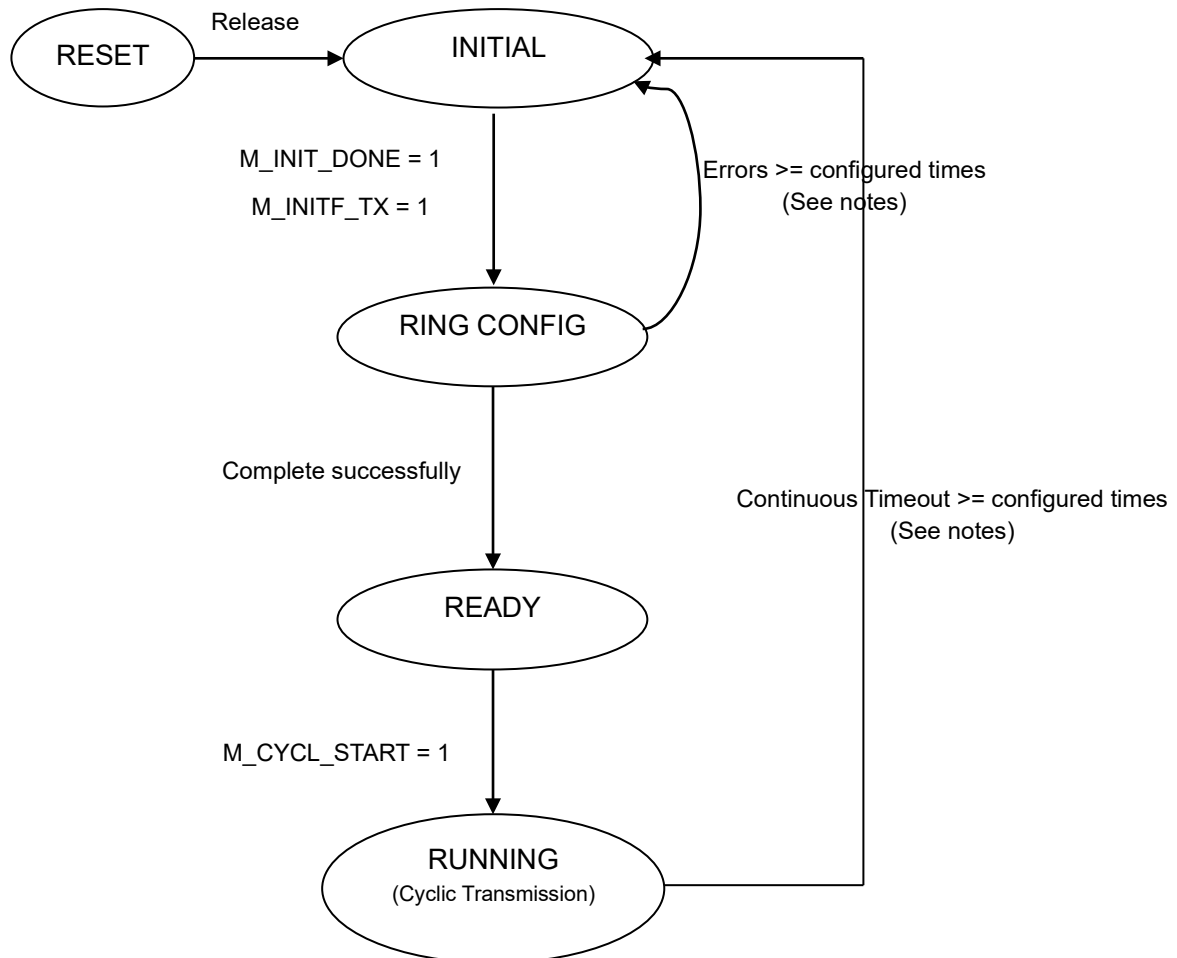
<Assignment on Memory>

Block #0	SLAVE #3
Block #1	SLAVE #2
Block #2	SLAVE #2
Block #3	SLAVE #4
Block #4	SLAVE #1
Block #5	SLAVE #1
Block #6	SLAVE #1
Block #7 to Block #31	Not used

State Transition of MNM1221

Before reaching RUNNING state that performs cyclic transmission, it goes through the process showed the following figure.

“M_...”: control registers



Notes:

- In RING CONFIG or RUNNING, M_ERR_COUNT configures the conditions changing to INITIAL.
- If M_RESET is set to 1 in any state, the state becomes INITIAL.

In the example code, the transition to INITIAL is not occurred because M_ERR_COUNT is set to 0. Instead of that, the firmware detects the timeout and makes MNM1221 transit to INITIAL using M_RESET.

Conditions on Using the Example Code

Precondition

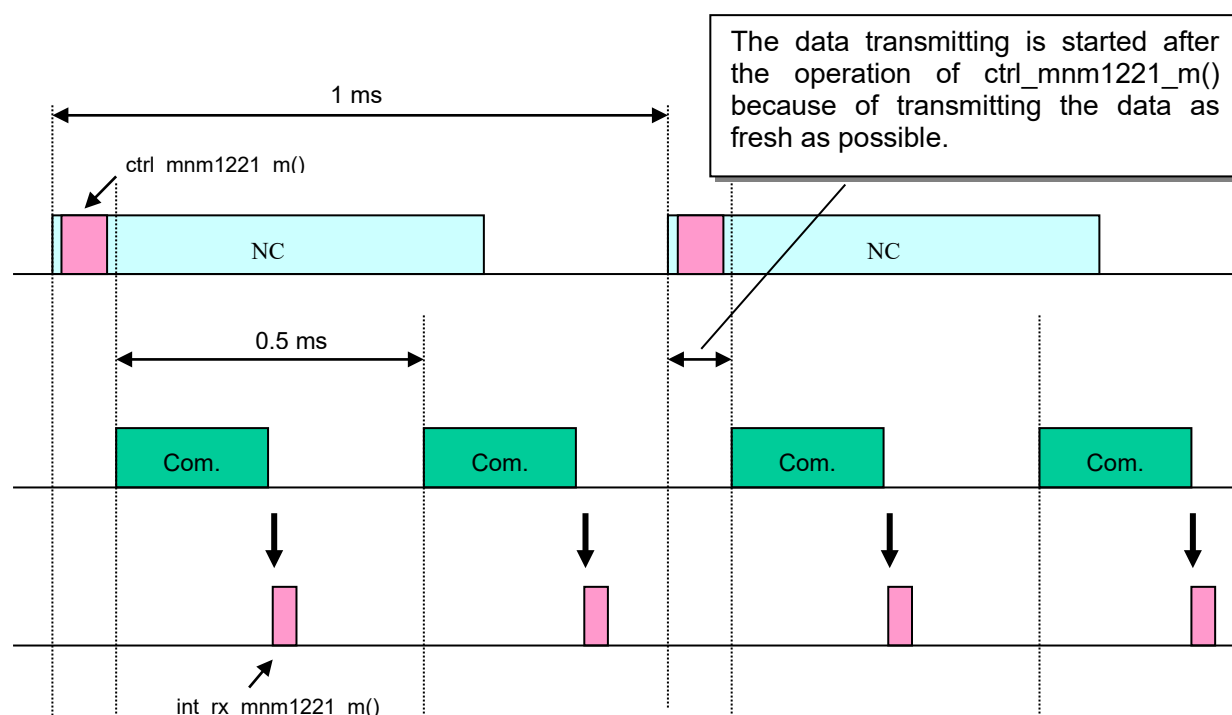
This example code is designed based upon using the following interrupts.

#	Trigger signal	Contents	Period	Priority
1	Timer (e.g. CPU built-in timer)	Interrupt for NC calculation	#2 x n (n: 1, 2, 3, ...)	Lower than #2 (multiplex interrupt)
2	XINTRX of MNM1221	Interrupt of receiving	-	High

The following functions are provided in the above premise.

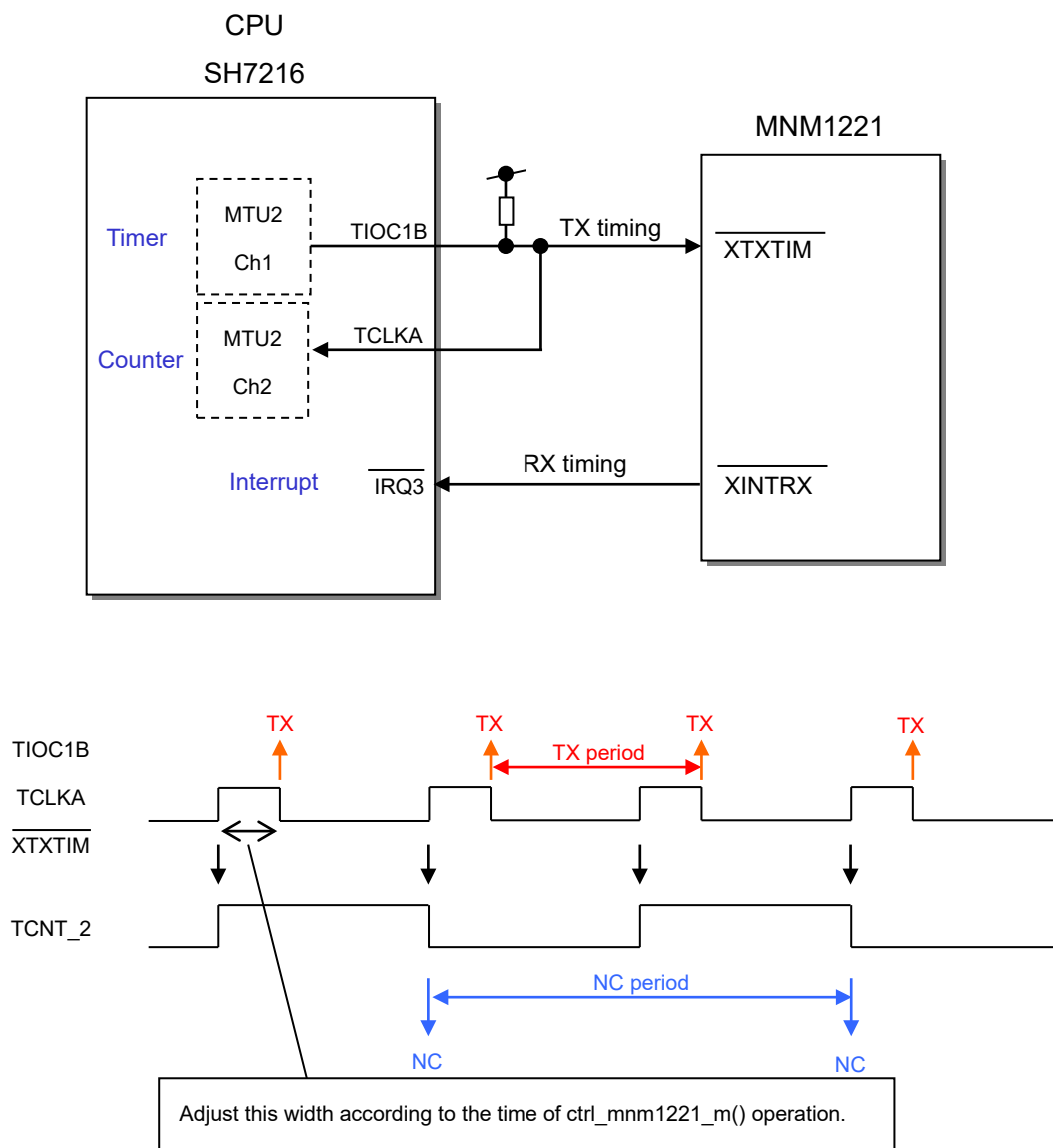
Function	Installed Location	Features
ctrl_mnm1221_m()	The head in the interrupt for NC calculation	The operation control of MNM1221. Communication data exchange.
int_rx_mnm1221_m()	Interrupt of receiving	Communication status confirmation.

e.g.) The below figure shows a case that NC calculation period is 1 ms and communication period is 0.5 ms.



Connection of Timing Signals

The following figure is a connection example of timing signals to construct the time chart in the former clause. It is an example when SH7216 by Renesas as the CPU is used.



- MTU2-Ch1 generates 0.5 ms TX timing.
- MTU2-Ch2 divides this signal, and generates 1 ms for NC calculating interrupt.
- IRQ3 causes RX interrupt.

Descriptions for the Example Code

Data

Main data used in this example code are shown in the table below.

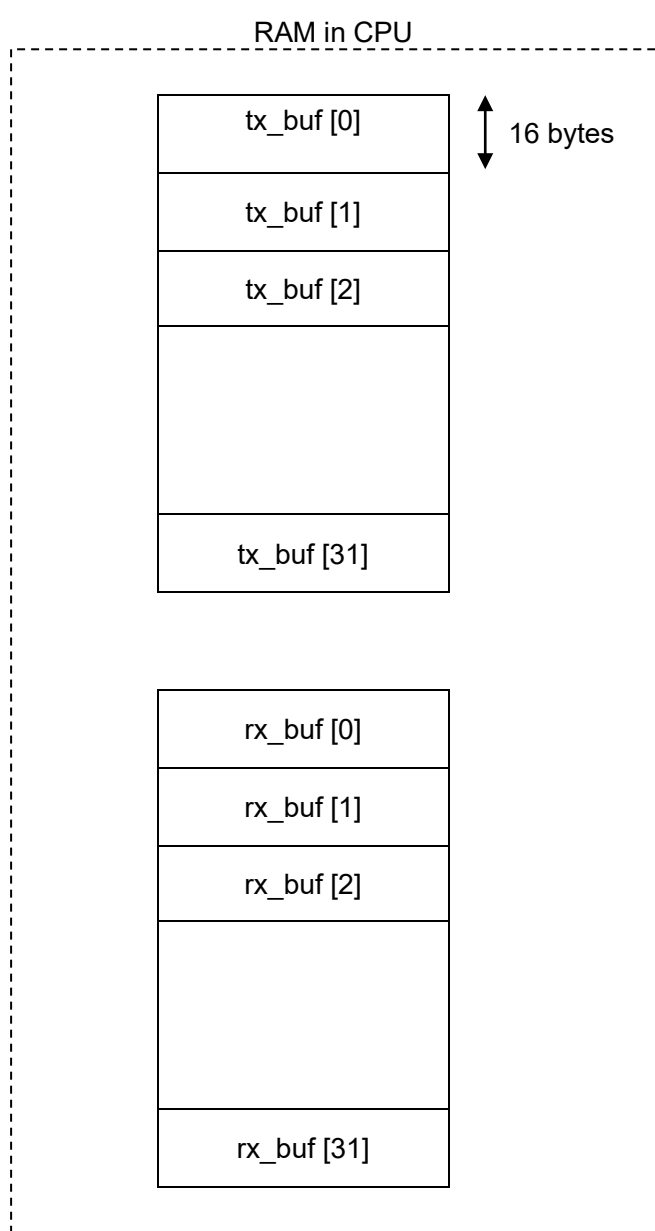
The `tx_buf[]` and `rx_buf[]` are buffer for the transmitting and receiving data. They corresponds with the slave information table `slave_inf_base[]` that is set up in advance before starting the operation. The order of `tx_buf[]` and `rx_buf[]` is free, but the data of Slaves are assigned to respective blocks on TX and RX memory in MNM1221 with the order of the ring connection. Therefore the sort of blocks is required using `order_ref_tbl[]` made with `slave_inf_actual[]` at the MNM1221 initialization.

Data	Contents	Note
<code>tx_buf[]</code>	Transmit data structure	One element is for one block (16 bytes).
<code>rx_buf[]</code>	Receive data structure	One element is for one block (16 bytes).
<code>slave_inf_base[]</code>	Base Slave information table	Corresponds with <code>tx_buf[]</code> and <code>rx_buf[]</code> . One element stands for one node. The sample code has the fixed values in this table. Therefore, you have to modify in the real application.
<code>slave_inf_actual[]</code>	Actual Slave information table	Corresponds with TX and RX memory in MNM1221. Copied from M_SINF registers. One element stands for one node.
<code>order_ref_tbl[]</code>	Order reference table	For the replacement of the data order. One element is for one block (16 bytes).

Transmitted and Received Data

The transmitted and received data should be accessed through `tx_buf[]` and `rx_buf[]` in the NC calculation processing.

It is composed of the structure arrangement, and one element is one block (16 bytes). The information of the data (e.g. MAC-ID) is given `slave_inf_base[]`. The order may be free regardless of MAC-ID, but invalid data between valid data is not allowed. Empty area must be assigned on upper element number as long as the sum of used blocks is less than 32.



Slave Information Table

Bit fields of slave_inf_base[] and slave_inf_actual[] is the same as the M_SINF registers in MNM1221, and one element is for one Slave node. The slave_inf_base[] must be set up in advance before starting. The slave_inf_actual[] is the copy of M_SINF registers, and it's element must correspond with the element of slave_inf_base[] in proper quantities. When it is not consistent, it means that there is either mistakes in the setup or problems of the actual connected devices.

Bit fields:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACT	MODE[1:0]	MAC-ID [4:0]				-	-	Number of Blocks [5:0]							

Bit 15: ACT

Indicates presence of the Slave node.

	Description
0	Not presence or not detected
1	Presence (The slave is active.)

Bit [14:13]: MODE[1:0]

Indicates a kind of the Slave node.

MODE1	MODE0	Description
0	0	None
0	1	Generic slave
1	0	Input slave
1	1	Output slave

Bit [12:8]: MAC-ID

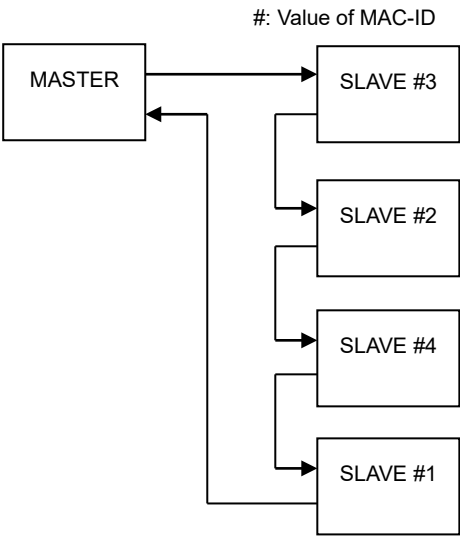
Indicates MAC-ID (0 to 31) of the Slave node.

Bit [5:0]: Number of Blocks

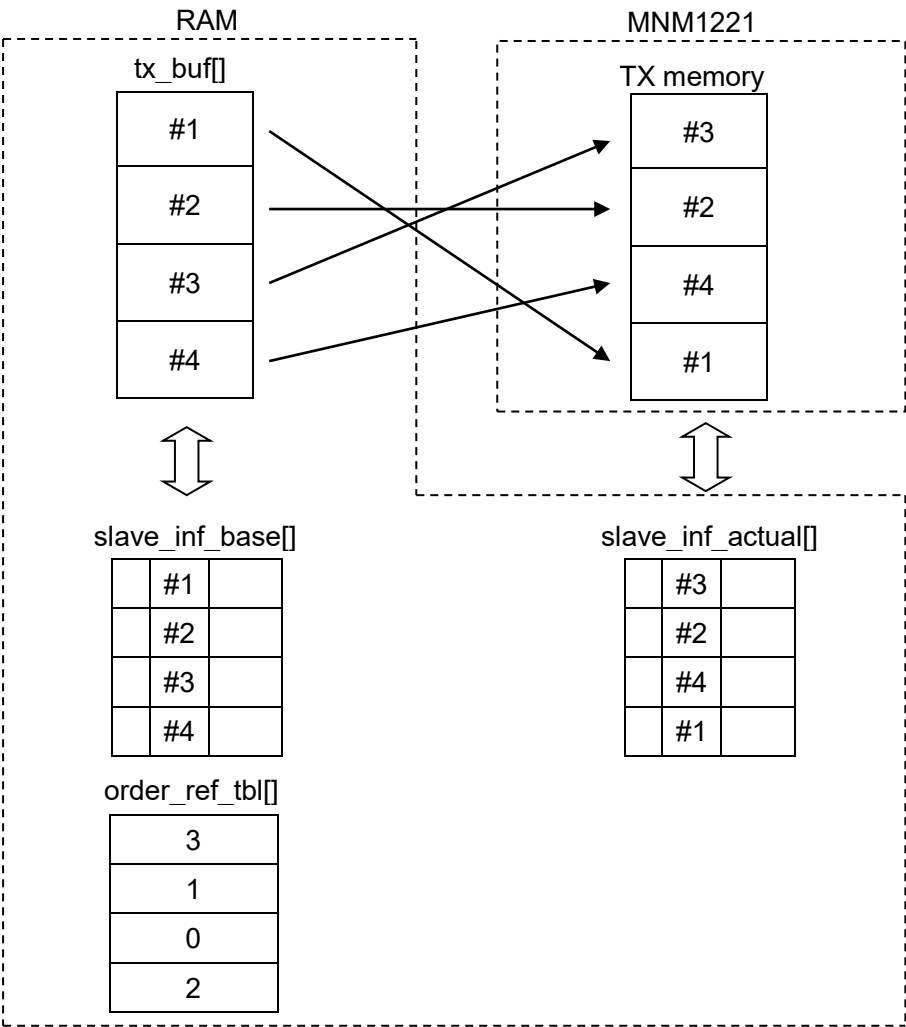
Indicates the number of data blocks the Slave node occupies.

Example of Rearranging the Data

<Connection>



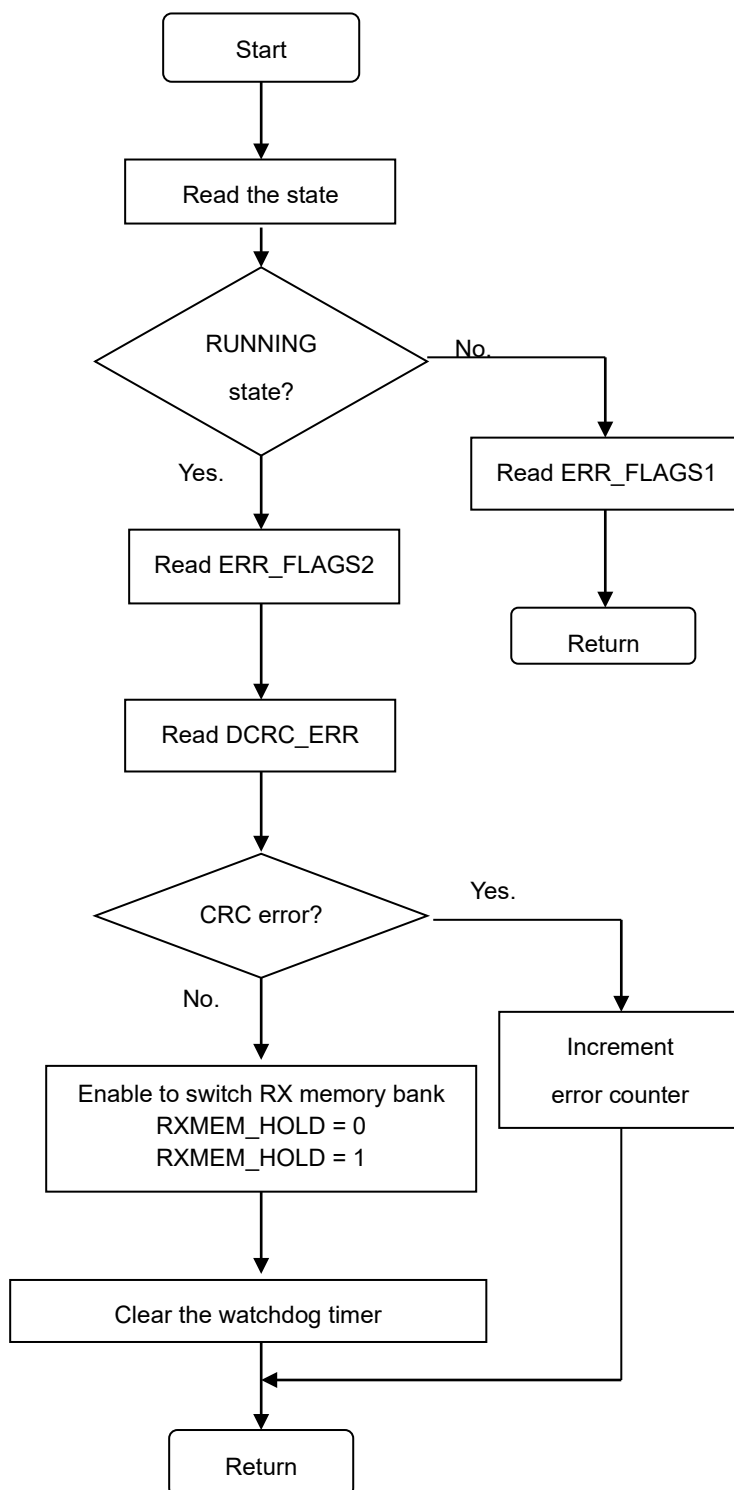
<Rearrangement at TX>



Flow Chart

Receiving Interrupt

Function Name	Contents
int_rx_mnm1221_m()	The receiving interrupt triggered by XINTRX of MNM1221.



Timer Interrupt

Function Name	Contents
ctrl_mnm1221_m()	Operated at the head of the timer interrupt for the NC calculation.

