

MNM1221

RTEX 用 100 Mbps 通信制御 ASIC

マスタ用サンプルコード

パナソニック インダストリー株式会社

変更履歴

Revision	日付	変更内容
0.1	2004/6/1	初版（暫定）
1	2012/2/10	P1 タイトルと社名を変更 P4 アドレスの単位を明記 P11 CPU 例を SH7065 から SH7216 に変更 P14 「サーボドライブ」を「汎用スレーブ」に、 「IN モジュール」を「IN スレーブ」に、 「OUT モジュール」を「OUT スレーブ」に変更
2	2023/3/31	P1 社名を変更 P11 図の CPU 品番の誤記を訂正

目次

はじめに.....	4
概要.....	4
プログラミング言語.....	4
ファイル構成.....	4
注意事項.....	4
システム構成.....	5
ハードウェア構成例.....	5
MNM1221 ブロック図.....	5
MNM1221 の送受信メモリ.....	6
MNM1221 のメモリマップ.....	7
送受信メモリのデータ配置.....	8
MNM1221 の状態遷移.....	9
サンプルコードの使用条件.....	10
前提条件.....	10
タイミング信号の接続.....	11
サンプルコードの解説.....	12
データ.....	12
送受信データ.....	13
スレーブ情報テーブル.....	14
データ並べ替えの例.....	15
フローチャート.....	16
受信割り込み.....	16
タイマ割り込み.....	17

はじめに

概要

本サンプルコードは MNM1221 をマスタとして動作させるための基本的な処理を収めたものです。主に、リセット解除後からサイクリック通信を行うまでの MNM1221 の起動処理と、RAM 上に配置した通信データバッファと MNM1221 内部の送受信メモリ間のデータ交換処理を収めています。

なお、アプリケーション処理は含んでいませんので、別途作成が必要となります。

プログラミング言語

C 言語(ANSI 規格準拠)

ファイル構成

ファイル名	内容
mnmm1221_m.h	レジスタ定義ヘッダファイル
mnmm1221_m.c	ソースファイル

注意事項

本サンプルコードは使用するシステム構成に合わせて変更を加える必要があります。必ず、最初に mnmm1221_m.h 内の次の箇所を設定してください。

- ・ MNM1221 を配置したアドレス（バイト単位のアドレス）
- ・ データバス幅（32 bits or 16 bits）

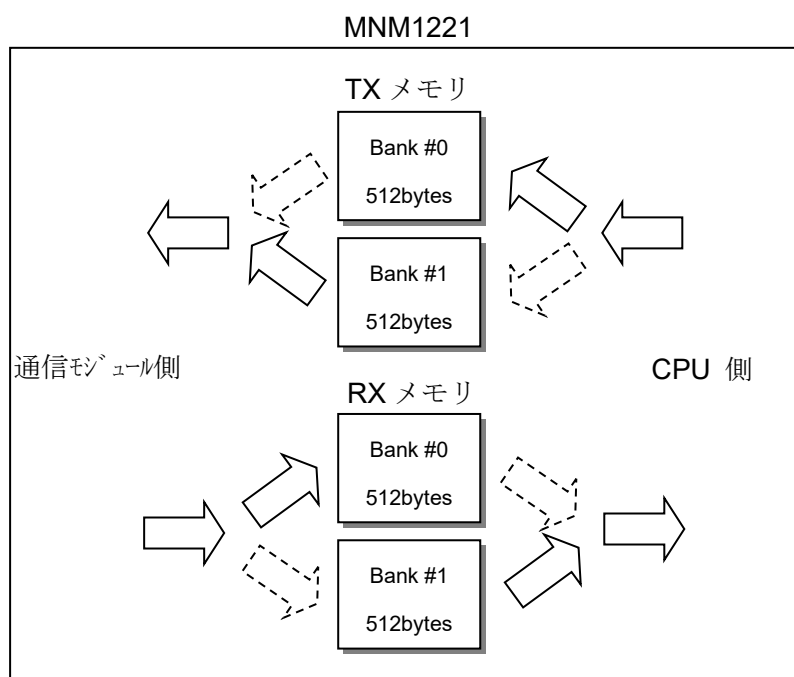
この詳細や他の変更すべき箇所についてはソースファイル中のコメントを参照してください。

重要：

本サンプルコードは MNM1221 の理解を助け、マスタ装置の開発を支援することを目的とするものであり、正しい動作を保証するものではありません。コード中に含まれているかも知れない不具合により被害が発生した場合、その責任を当社が負うことはできませんので、ご了承ください。

MNM1221 の送受信メモリ

MNM1221 内部の送信バッファメモリと受信バッファメモリはそれぞれ2バンク構成になっています。アクセスの競合を避けるため、2つのバンクの内、一方が外部の CPU からのアクセス側、他方が内部の通信モジュール側となり、交互に切り替えて使用します。



バンク切り替え:

	切り替えのトリガ	切り替えの保留条件
送信メモリ	M_TXMEM_SW = 1 (レジスタは自動的にクリアされます)	送信中
受信メモリ	受信完了	M_RXMEM_HOLD = 1

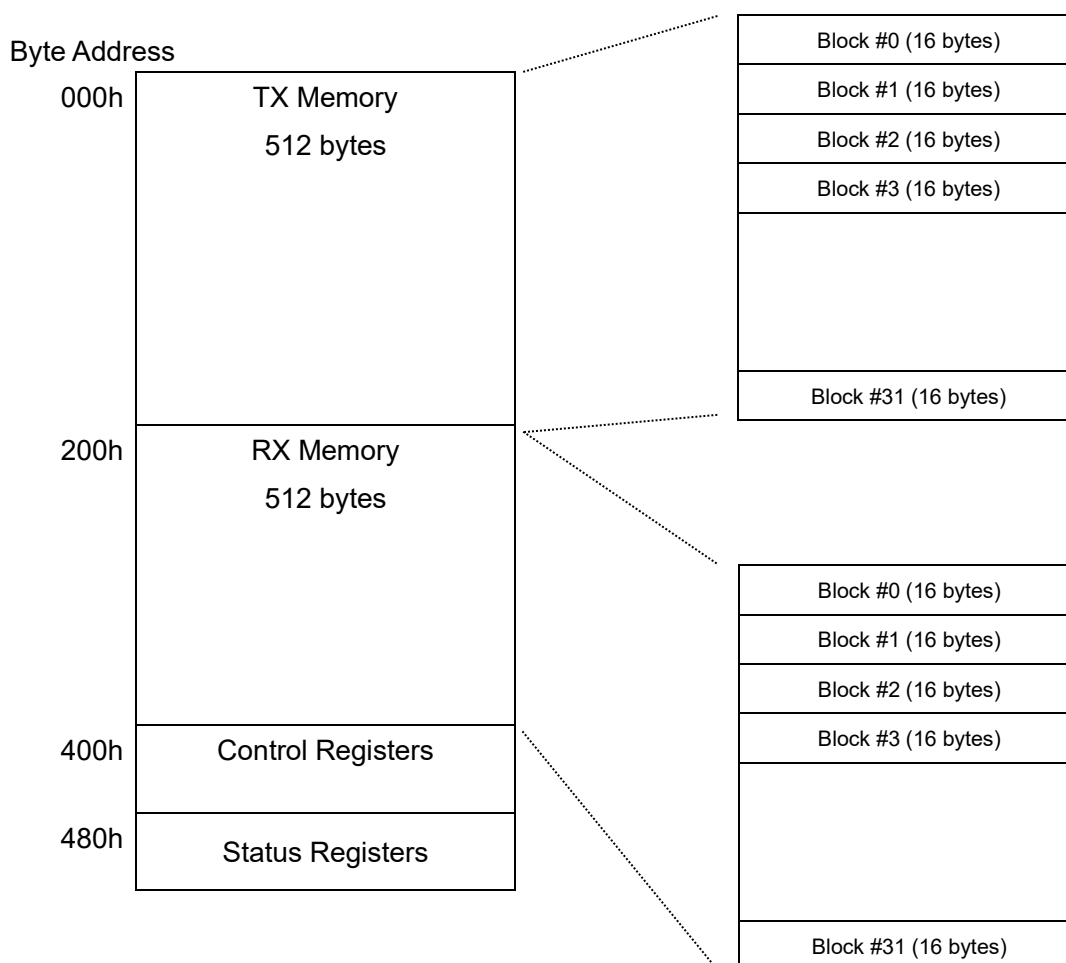
注記:

- 送信メモリへの書き込み後は、**M_TXMEM_SW** を1にしてバンクを切り替えてください。
- 受信データを読み出す前には、**M_RXMEM_HOLD** を0にして保留されているバンク切り替えを実行させてください。
- ノイズの多い動作環境においても、正常なデータのみを最小の遅れで取り込むことを目的として、本サンプルコードでは、次のような受信処理を行っています。
 - ・NC 演算 1 回あたりに複数回の通信を行う。
 - ・通常は受信メモリの切り替えを保留状態（切り替え禁止状態）にしておく。
 - ・受信毎に正常受信時のみ、切り替え保留を解除して切り替えを行う。
 - ・NC 演算処理で RX メモリを読み出す。

MNM1221 のメモリマップ

送受信メモリはバンクの切り替えに関わらず、同じエリアでアクセスします。

各メモリの内部は32個のブロックに分割されており（1ブロックは16バイト）、通常は1つのブロックが1スレーブノードに対応します（1スレーブが複数ブロックを占有する場合もあります）。



バイト順序:

データは「リトルエンディアン」、すなわち、下位データが下位アドレスになります。

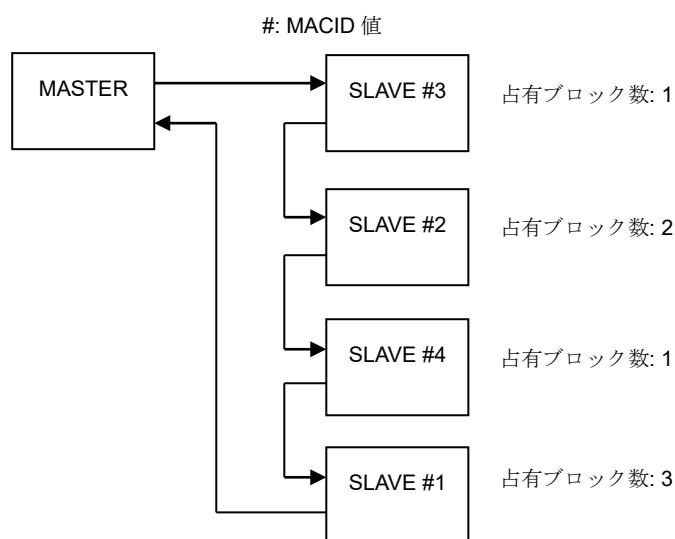
Byte Address	4n + 3		4n + 2		4n + 1		4n + 0	
Data	31	24	23	16	15	8	7	0

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

送受信メモリのデータ配置

スレーブのデータはケーブル接続順で送受信メモリ上にデータが配置されます。メモリの下位ブロックから順に配置されるので、32ブロック全てを使用しない限り上位側ブロックは空きとなります。下図は配置例を示すもので、複数のブロックを占有するスレーブが存在する場合を表しています。但し、通常は、1スレーブは1ブロックのみを占有します。

<接続構成>



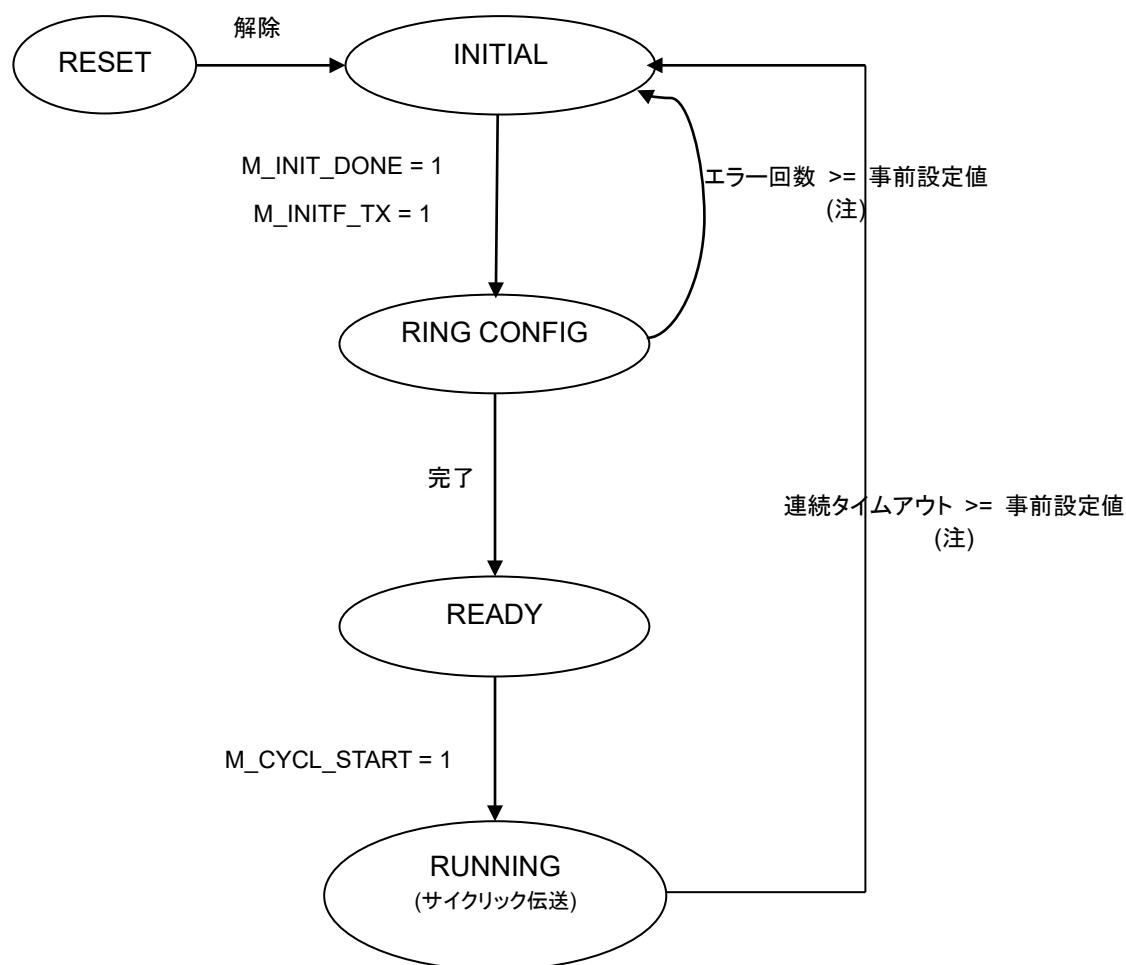
<メモリ配置>

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	未使用

MNM1221 の状態遷移

サイクリック転送を行なう **RUNNING** 状態になるまでの状態遷移を下図に示します。

“M_...”: 制御レジスタ



注記:

- RING CONFIG や RUNNING 状態では、M_ERR_COUNT レジスタで INITIAL 状態に抜ける条件を設定します。但し、本サンプルコードではこれを INITIAL 状態には遷移しない設定にしています。MNM1221 によるハードウェア処理に代えてファームウェア処理でタイムアウトを検出し、M_RESET レジスタの操作で INITIAL 状態に遷移させる構成にしています。

- M_RESET レジスタを 1 にすれば、どの状態からでも INITIAL 状態に抜けることができます。

サンプルコードの使用条件

前提条件

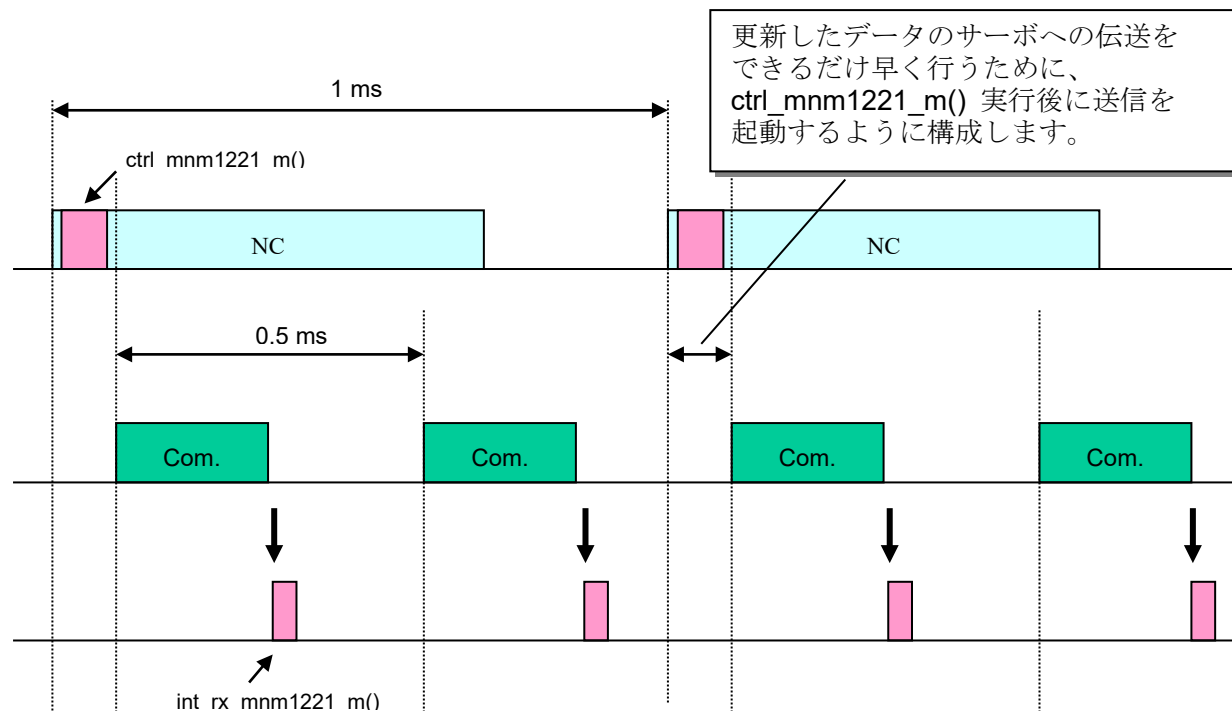
本サンプルコードは、次の割り込み処理が行われることを前提にしています。

#	起動信号	内容	周期	優先順位
1	タイマ (e.g. CPU 内蔵タイマ)	NC 演算割り込み	#2 x n (n: 1, 2, 3, ...)	#2 より低 (多重割り込み)
2	MNM1221 の XINTRX	受信割り込み	-	高

上記前提に立ち、次の関数を用意しています。

関数	実行箇所	機能
ctrl_mnm1221_m()	NC 演算割り込みの先頭	MNM1221 の動作制御 通信データ交換
int_rx_mnm1221_m()	受信割り込み内	通信ステータスの確認

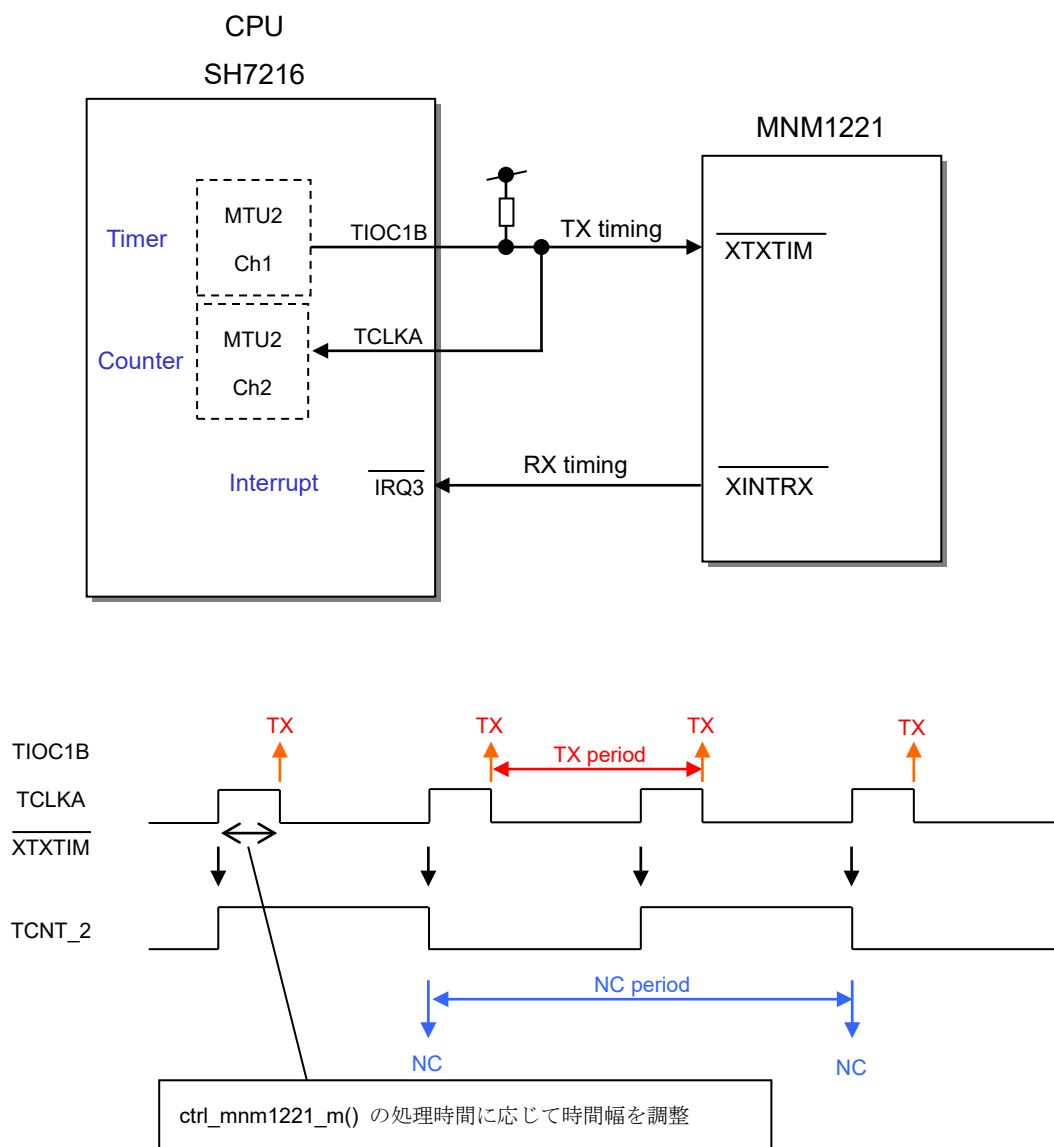
例) 下図は NC 演算周期が 1 ms、通信周期が 0.5 ms の場合を示します。



タイミング信号の接続

前項のタイムチャートを構成するための、タイミング信号接続例を以下に示します。

CPU が SH7216 (ルネサス・テクノロジー) の場合の例です。



- MTU2-Ch1で周期0.5 msの送信起動信号を生成
- この0.5 ms周期信号をMTU2-Ch2で分周し、周期1 msのNC演算起動信号を生成
- 受信割り込みとして IRQ3 を使用

サンプルコードの解説

データ

本サンプルコードで使用する主なデータを下表に示します。

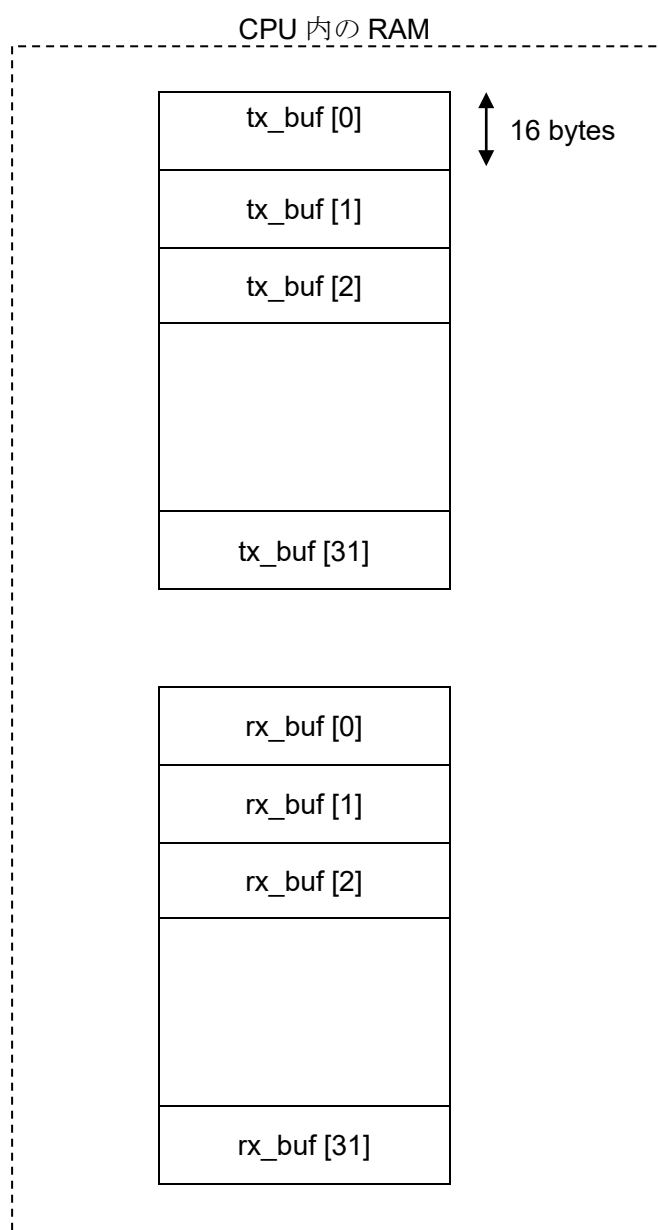
`tx_buf[]`, `rx_buf[]`は送受信データを納めるバッファであり、動作開始前に、事前に設定しておくスレーブ情報テーブル `slave_inf_base[]`と対応しています。配列の順序は任意ですが、これに対して MNM1221 内部のメモリはノードの接続順に配置されているので、データ転送時に並べ替えを行います。並べ替えは MNM1221 初期化時に `slave_inf_actual[]`に基づいて作成する `order_ref_tbl[]`を用いて行います。

データ	内容	備考
<code>tx_buf[]</code>	送信データ構造体配列	1 配列要素は 1 ブロック分 (16 バイト)。
<code>rx_buf[]</code>	受信データ構造体配列	1 配列要素は 1 ブロック分 (16 バイト)。
<code>slave_inf_base[]</code>	基準スレーブ情報テーブル	1 配列要素は 1 ノード分。 <code>tx_buf[]</code> , <code>rx_buf[]</code> と対応。 サンプルコードではテーブルを固定値にしていますが、実際のアプリケーションでは設定変更できるように構成してください。
<code>slave_inf_actual[]</code>	実際のスレーブ情報テーブル	1 配列要素は 1 ノード分。 MNM1221 内の送受信メモリと対応。 M_SINF レジスタのコピー。
<code>order_ref_tbl[]</code>	順序参照テーブル	1 配列要素は 1 ブロック分 (16 バイト)。 データ順序の並べ替えに使用。

送受信データ

送受信データは NC 演算処理の中で、`tx_buf[]`, `rx_buf[]`を介してアクセスしてください。

構造体配列で構成しており、1 配列要素は1 ブロック分（16 バイト）です。このデータに関する情報（MACID 等）は `slave_inf_base[]`で与えます。MNM1221 とのデータ交換時に順序の並べ替えを行うため、順序は自由に配置できます。MACID 順でなくても構いません。但し、有効なデータの間に無効なデータを配置することはできません。必ず小さな配列番号から順に詰めて配置してください。使用するブロック総数が32でない限り、配列番号の大きな要素は空きとなります。



スレーブ情報テーブル

slave_inf_base[] と slave_inf_actual[] は 1 配列要素が 1 ノードに相当するもので、ビット構成は MNM1221 の M_SINF レジスタと同じです。この内、slave_inf_base[] は基準となるものであり、動作開始前に、事前に設定しておく必要があります。また、もう一方の slave_inf_actual[] は M_SINF レジスタのコピーであり、slave_inf_base[] の配列要素と過不足なく対応しなければなりません。整合しない場合は、事前の設定情報、もしくは実際の接続デバイスのいずれかに誤りがあることを意味します。

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

スレーブノードの有無を示します。

	説明
0	無し (このスレーブ情報は無効)
1	在り (このスレーブ情報は有効)

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

スレーブの種類を示します。

MODE1	MODE0	説明
0	0	無効
0	1	汎用スレーブ
1	0	IN スレーブ
1	1	OUT スレーブ

Bit [12:8]: MAC-ID

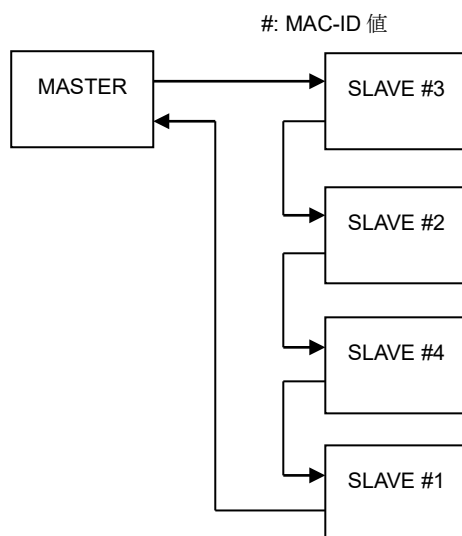
MAC-ID (0 to 31) を示します。

Bit [5:0]: Number of Blocks

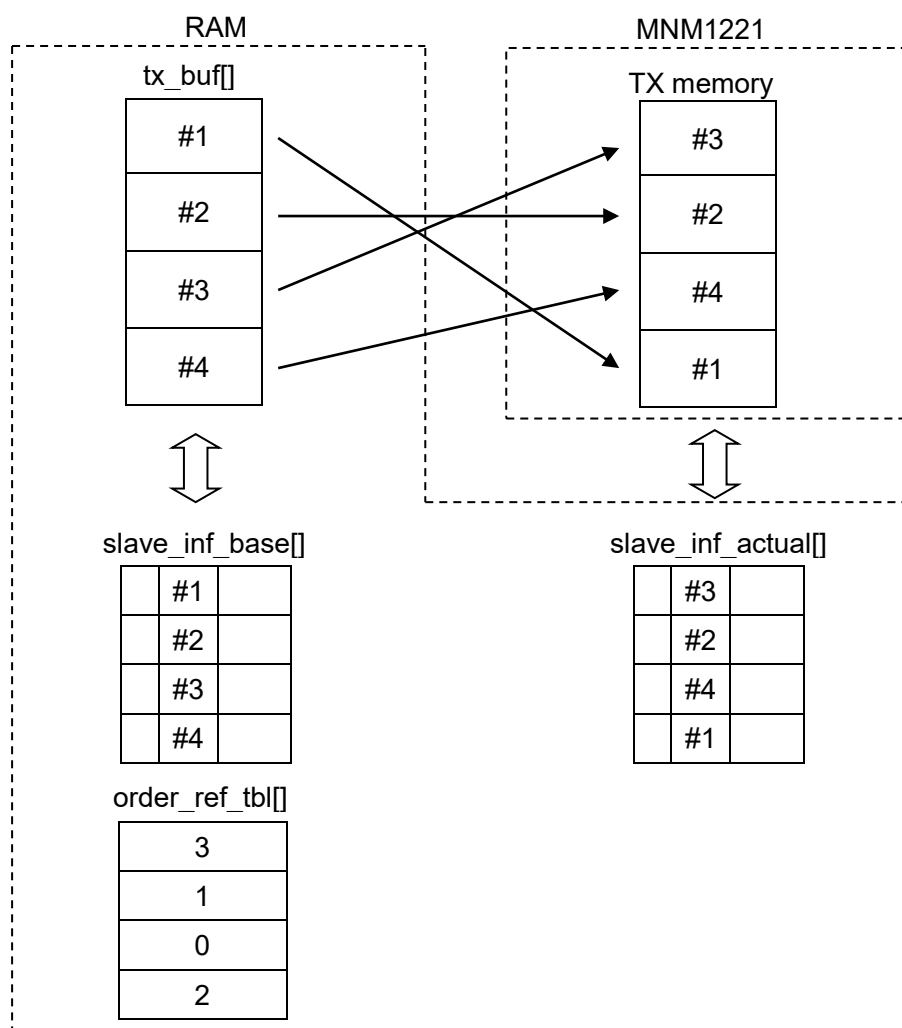
占有するデータブロック数を示します。

データ並べ替えの例

<接続構成>



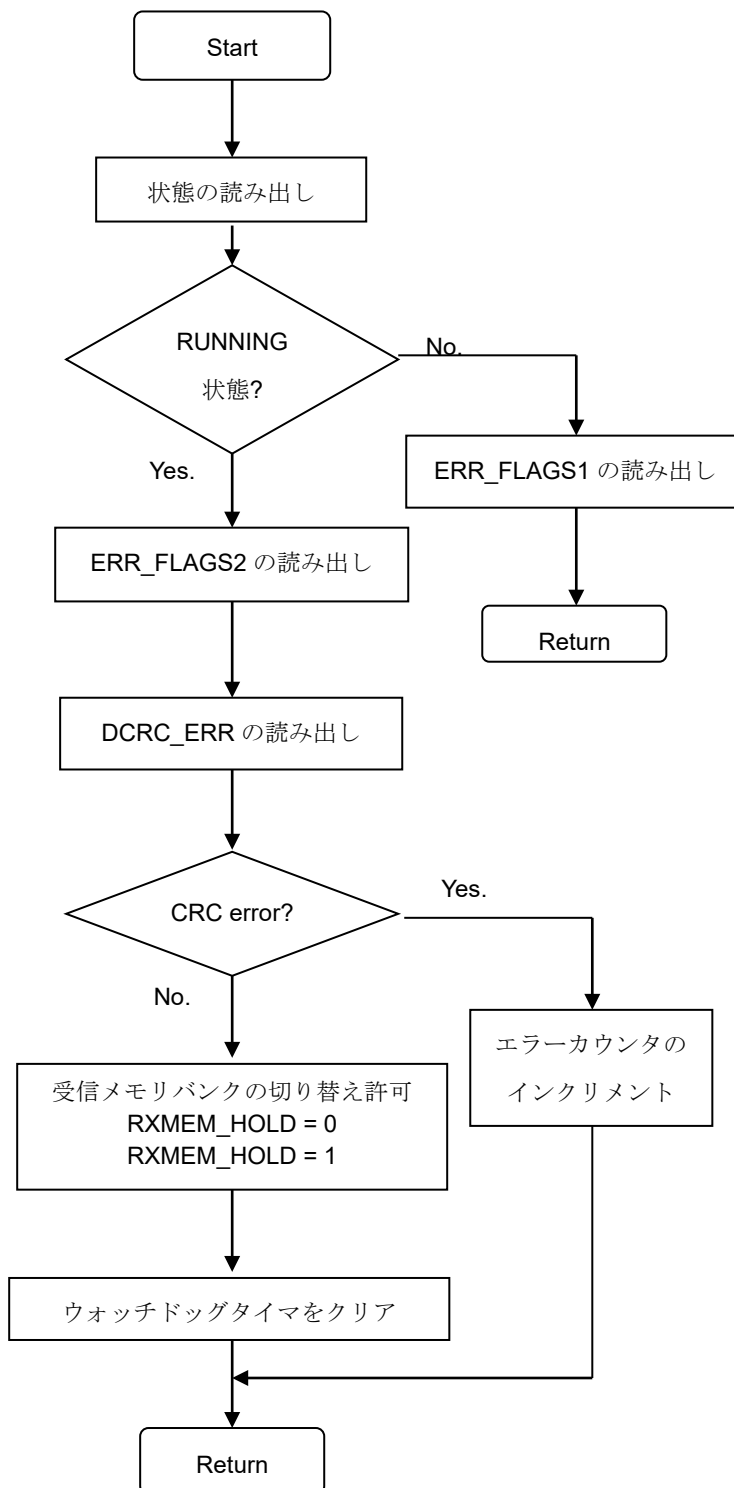
<送信時の並べ替え（受信も同様）>



フローチャート

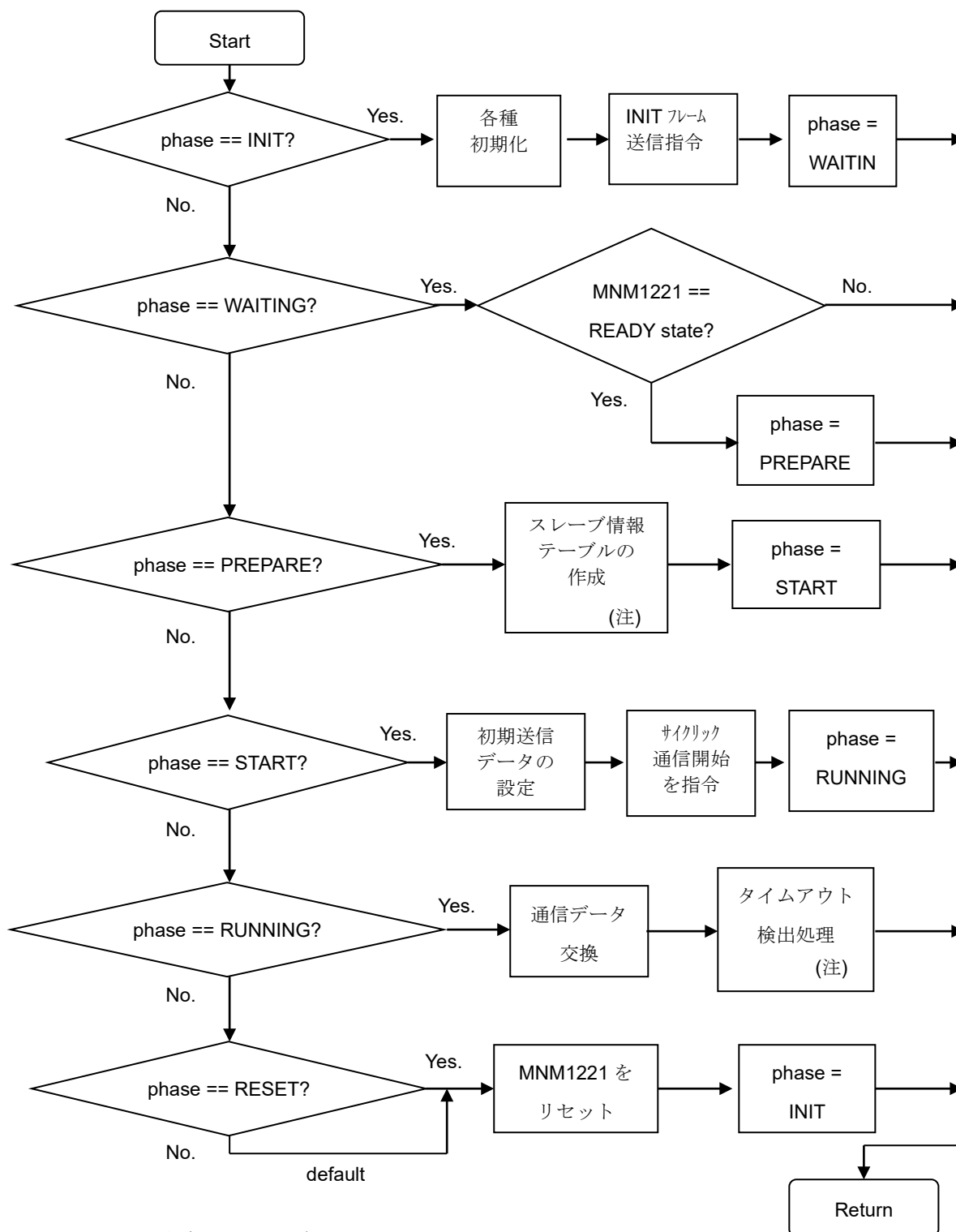
受信割り込み

関数名	内容
int_rx_mnm1221_m()	MNM1221 の XINTRX で起動する受信割り込み



タイマ割り込み

関数名	内容
ctrl_mnm1221_m()	NC 演算用タイマ割り込みの先頭で実行



注記: エラー発生時の処理が別途必要です。