

Modbus プロトコル概説書

目 次

1、はじめに	4
2、Modbus プロトコル	4
2、1 Modbus プロトコル概要	4
2、2 Modbus メッセージ・フレーム	5
(1) ASCII モード時のフレーム	5
(2) RTU モード時のフレーム	5
(3) アドレス (Address)・フィールド	5
(4) ファンクション・フィールド	5
(5) データ・フィールド	5
(6) エラーチェック・フィールド	6
(7) LRC チェック	6
(8) CRC チェック	6
2、3 Modbus ファンクション・フォーマット	7
(1) データ・アドレス	7
(2) コイル (Coil)	7
(3) 入力ステータス (Input Status)	7
(4) 入力レジスタ (Input Register)	7
(5) 保持レジスタ (Holding Register)	7
2、4 Modbus メッセージ構成	8
3、ファンクション・コード (Modbus ファンクション)	9
3、1 Read Coil Status (01)	9
3、2 Read Input Status (02)	10
3、3 Read Holding Register (03)	11
3、4 Read Input Register (04)	12
3、5 Force Single Coil (05)	13
3、6 Preset Single Register (06)	14
3、7 Diagnostics (08)	15
3、8 Fetch Communication Event Counter (11, 0XB)	16
3、9 Fetch Communication Event Log (12, 0XC)	17
3、10 Force Multiple Coils (15, 0X0F)	18
3、11 Preset Multiple Registers (16, 0X10)	19
3、12 Report Slave ID (17, 0X11)	20

4、診断 (0X08).....	21
4、 1 Return Query Data (00).....	21
4、 2 Restart Communications Option (01).....	21
4、 3 Return Diagnostics Register (02).....	21
4、 4 Force Listen Only Mode (04).....	22
4、 5 Clear Counters and Diagnostic Register (10 , 0X0A).....	22
4、 6 Return Bus Message Count (11 , 0X0B).....	22
4、 7 Return Bus Communication Error Count (12 , 0X0C).....	22
4、 8 Return Bus Exception Error Count (13 , 0X0D).....	22
4、 9 Return Slave Message Count (14 , 0X0E).....	22
4、 10 Return Slave No Response Count (15 , 0X0F).....	23
4、 11 Return Slave Busy Count (17 , 0X11).....	23
4、 12 Return Bus Character Overrun Count (18 , 0X12).....	23
5、例外レスポンス (Exception Response).....	24
補 足 1 Modbus 通信機能	26
1、R1M シリーズの Modbus 通信機能	26
1、 1 ファンクション・コード	26
1、 2 データアドレス.....	26
1、 3 コイル (DO) データ詳細	27
(1) DO (1 ~ 32).....	27
(2) 冷接点温度補償 SW (33 ~ 48).....	27
1、 4 入力ステータス (DI) データ詳細	27
(1) DI (10001 ~ 10032).....	27
(2) ADC レンジオーバー (10033 ~ 10048).....	27
1、 5 入力レジスタデータ詳細	27
(1) アナログ入力 % 値 (30001 ~ 30016).....	27
(2) アナログ入力工業単位値 (30017 ~ 30048).....	27
(3) 冷接点温度 (30049 から 30050).....	27
(4) チャンネル・ステータス (30081 ~ 30096).....	28
(5) システム・ステータス (30513).....	28
(6) 形式 (30514 ~ 30521).....	28
(7) 機番 (30522 ~ 30529).....	28
(8) ハードウェア・バージョン番号 (30530 ~ 30537).....	28
(9) ファームウェア・バージョン番号 (30538 ~ 30545).....	28
1、 6 保持レジスタ・データ詳細	29
(1) アナログ出力 % 値 (40001 ~ 40016).....	29
(2) アナログ出力工業単位値 (40017 ~ 40048).....	29
(3) 入出力タイプ番号 (40145 ~ 40160).....	29
(4) パーンアウトタイプ (40161 ~ 40176).....	30

2、R2M シリーズの Modbus 通信機能	30
2、1 通信パラメータ	30
2、2 ファンクション・コード	30
2、3 データアドレス	31
2、4 コイル (DO) データ詳細	31
(1) DO (1 ~ 32)	31
(2) 冷接点温度補償 SW (33 ~ 40)	31
2、5 入力ステータス (DI) データ詳細	31
(1) DI (10001 ~ 10032)	31
(2) ADC レンジオーバー (10033 ~ 10040)	31
2、6 入力レジスタデータ詳細	32
(1) アナログ入力工業単位値 (30017 ~ 30032)	32
(2) 冷接点温度値 (30049 ~ 30050)	32
(3) チャンネル・ステータス (30081 ~ 30088)	32
(4) システム・ステータス (30513)	32
(5) 形式 (30514 ~ 30521)	33
(6) 機番 (30522 ~ 30529)	33
(7) ハードウェア・バージョン番号 (30530 ~ 30537)	33
(8) ソフトウェア・バージョン番号 (30538 ~ 30545)	33
2、7 保持レジスタ・データ詳細	33
(1) 入力フィルタリング時定数 (40049 ~ 40050)	33
(2) 入出力タイプ番号 (40145 ~ 40152)	33
(3) バーンアウトタイプ (40514)	33
補 足 2 Modbus / TCP プロトコル	34
1、はじめに	34
2、プロトコルのレイアウト	34
3、動作の例	35
4、注意事項	35

1、はじめに

Modbus プロトコルは、Modicon Inc. (AEG Schneider Automation International S.A.S.) が PLC 用に開発した通信プロトコルで、プロトコル仕様書 (PI-MBUS-300 Rev.J) に記載されています。Modbus プロトコルの詳細な仕様に関しては当仕様書をご参照下さい。Modbus プロトコルで定義されているのは通信プロトコルのみで、通信媒体などの物理レイヤは規定されていません。Modbus プロトコルをサポートしている機器には、R1M、R2M シリーズ、60M および M2XUM があります。これら機器がサポートしている通信パラメータは、各仕様書を参照して下さい。

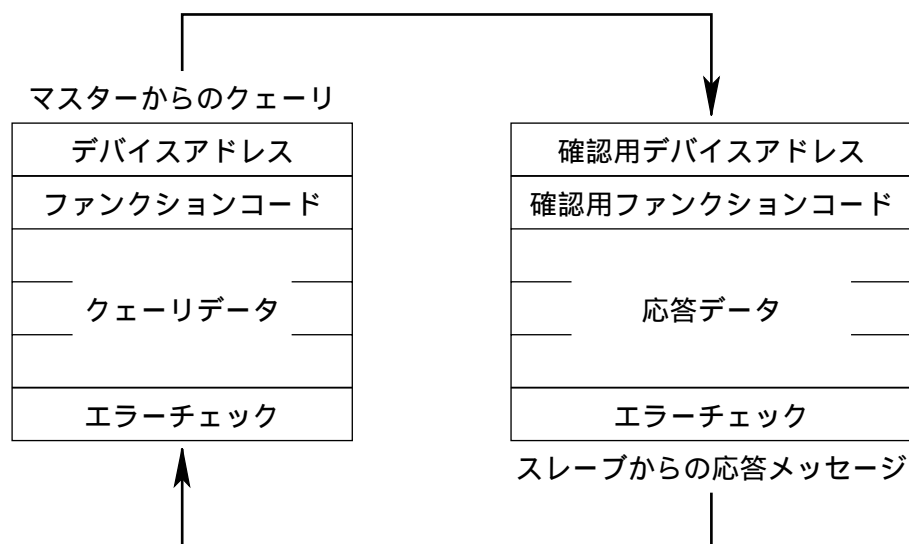
2、Modbus プロトコル

2、1 Modbus プロトコル概要

Modbusの通信方式は、シングルマスター/マルチスレーブ方式です。マスターだけがクエリ(通信の開始)を発行することができます。スレーブは、このクエリを見て、指定された機能を実行し、応答メッセージを返します。マスターは、指定のスレーブに対するクエリ、または全てのスレーブに対するブロードキャストクエリのいずれかを発行することができます。ブロードキャストクエリの場合には、スレーブは指定の機能を実行するのみで、応答メッセージは返しません。スレーブは、自分に対するクエリの際にだけ応答メッセージを返します。

クエリの伝送フォーマットは、スレーブのアドレス(またはブロードキャスト) 要求内容を定義するファンクションコード、データおよびエラーチェックフィールドから構成されています。また、応答メッセージの伝送フォーマットは、要求内容の確認フィールド、応答データおよびエラーチェックフィールドから構成されています。

クエリと応答メッセージの伝送フォーマットを下图に示します。



シリアル伝送モードには ASCII (American Standard Code for Information Interchange) モードと RTU (Remote Terminal Unit) モードの 2 種類があり、選択することができます。ただし、1つのネットワーク上では、全てのデバイスが同一モードでなくてはなりません。ASCII モードでは、1バイト(8ビット)データを2文字のASCIIコードに変換して伝送します。RTU モードでは、1バイト(8ビット)データをそのまま伝送します。従って ASCII モードより伝送効率が良いと言えます。

エラーチェックフィールドのチェックアルゴリズムは伝送モードによって異なります。ASCII モードの場合には、LRC (Longitudinal Redundancy Check) 法が、また RTU モードの場合には、CRC (Cyclical Redundancy Check) 法が採用されています。

2、2 Modbus メッセージ・フレーム

Modbus メッセージ・フレームは、2つの伝送モードによって異なります。

(1) ASCII モード時のフレーム

ASCII モードの場合には、メッセージは“ : ”(0X3A)のASCII文字列で始まり、“ CR / LF ”(Carriage return line feed)(0X0D, 0X0A)で終わります。その間に許されるデータは、“ 0 ” - “ 9 ”、“ A ” - “ F ” からなる文字列のみです。

Ethernet 通信の場合、当メッセージ・フレームをそのままデータフレームとし、TCP / IP プロトコルの1データフレームとします。メッセージ・フレームを分割してはなりません。

ASCII モード時のメッセージ・フレーム構成 (表1)

Start	Address	Function	Data	LRC Check	End
1- 文字 :	2- 文字	2- 文字	n - 文字	2- 文字	2- 文字 CR / LF

(2) RTU モード時のフレーム

RTU モードの場合、少なくとも3.5文字分のサイレントインターバルで始まり、3.5文字分のサイレントインターバルで終わります。これを(表2)ではT1-T2-T3-T4で表します。その他のフィールドは、8ビットのデータそのものです。

RTU モード時のメッセージ・フレーム構成 (表2)

Start	Address	Function	Data	CRC Check	End
T1-T2-T3-T4	8- ビット	8- ビット	n*8- ビット	16- ビット	T1-T2-T3-T4

(3) アドレス (Address) ・フィールド

アドレス・フィールドは、0から247(10進数)が許されます。スレーブ・アドレスは1から247です。マスターがスレーブにクエリを発する場合には、このアドレスフィールドにスレーブのアドレスをセットします。スレーブがマスターに回答メッセージを返す場合には、スレーブのアドレスをセットします。これによって、マスターはどのスレーブからの回答であるかを知ることができます。アドレス0はブロードキャストクエリに用います。

(4) ファンクション・フィールド

設定可能なファンクション・コードは1から255(10進数)です。ファンクション・コードに従って、スレーブは指定された機能を実行します。実行後、回答メッセージを返す場合、正常回答メッセージには同じファンクション・コードを設定し、例外回答メッセージにはファンクション・コードのMSBを1にセットします。これで、マスターはどのファンクション・コードに関する回答メッセージであるかを知ることができます。設定したファンクション・コードが有効か否かは、スレーブデバイスに依存します。従って、各スレーブの仕様書をご参照下さい。

(5) データ・フィールド

ファンクション・コードに関連した、データを送信する場合に用います。フィールド長は可変で、データ・フィールドなしも許されます。データ・フィールドの構成と意味は各スレーブの仕様書をご参照下さい。

(6) エラーチェック・フィールド

伝送モードによって、エラーチェック・フィールドの内容は異なります。

ASCII モード

ASCII モードの場合には、エラーチェック・フィールドは2文字から構成されます。Start “ : ” 文字と End “ CR / LF ” を除く文字の和 (LRC : Longitudinal Redundancy Check) の結果を2文字で表します。

RTU モード

RTU モードの場合には、エラーチェック・フィールドは16ビットのデータを2つの8ビットバイトデータで表します。エラーチェックの結果はCRC (Cyclical Redundancy Check calculation) と言われる計算方法で計算されます。

(7) LRC チェック

ASCII モードにおいて、メッセージにはLRC方式に基づいたエラーチェックフィールドが含まれています。LRC フィールドは、最初のコロンと最後のCRLFを除いたメッセージの中身をチェックします。これは、メッセージ中の個別の文字(キャラクタ)のパリティチェック方式とは関連せずに行われます。

LRC フィールドは1バイト構成で、8ビットのバイナリ値で構成されています。LRC値は、LRCをメッセージに付加する送信側が計算します。受信側は、メッセージ受信中にLRCを計算して、その計算結果とLRCフィールドに受信した実際の値と比較します。もし、この2つの値が一致しなければ、結果はエラーとなります。

LRCを計算するときは、メッセージの中の連続した8ビットのバイトからキャリーを除いて足し算し、結果を2の補数にします。これはASCIIメッセージフィールドの中身について行われますが、メッセージの最初のコロン記号と最後のCRLFは除きます。

(8) CRC チェック

RTUモードにおいて、メッセージにはCRC方式に基づいたエラーチェックフィールドが含まれています。CRCフィールドの場合は、メッセージ全体の内容をチェックします。また、メッセージ中の個別の文字(キャラクタ)のパリティチェック方式と関連せず行われます。

CRCフィールドは2バイト構成で、16ビットのバイナリ値で構成されています。CRC値は、CRCをメッセージに付加する送信側が計算します。受信側は、メッセージ受信中にCRCを再計算して、その計算結果とCRCフィールドに受信した実際の値と比較します。もし、この2つの値が一致しなければ、結果はエラーとなります。

CRC計算ではまず、すべて1の16ビットのレジスタをプレロードします。そして次に、メッセージの中の連続した8ビットのバイトを、現在のレジスタの中身に適用していきます。CRCを生成するときには、各キャラクタのうち8ビットだけを使います。スタートおよびストップ、パリティビットはCRCには適用されません。

CRCを生成する途中、各8ビットキャラクタはレジスタの中身とエクスクルーシブORされます。さらにその結果を最下位桁の方向にシフトし、最上位桁にはゼロを入れます。最下位桁を取出して検査します。もし、最下位が1の場合はさらに、レジスタは既設の固定値(0XA001)でエクスクルーシブORされます。もし、最下位が0の場合、エクスクルーシブORは起こりません。

この過程を8回シフトするまで繰り返します。最後(8回目)のシフトの後、次の8ビットのバイトについて、レジスタの現在値でエクスクルーシブORします。そして前述のように、この過程をさらに8回繰り返します。メッセージの全てのバイトについて適用した後、レジスタの最後の中身がCRC値となります。

メッセージにCRCを付加するときには、下位バイトが先に付加され、その後に上位バイトが続きます。

2、3 Modbus ファンクション・フォーマット

(1) データ・アドレス

Modbus 通信で、データの参照や変更を行う場合、データ・アドレスを用いて行います。データの種類には、コイル、入力ステータス、入力レジスタ、保持レジスタの4種類があります。

(2) コイル (Coil)

コイルは、フィールドへの ON / OFF 出力である DO (Discrete Output) やスレーブデバイスの状態やモードを変更する為にスイッチとして用いられます。参照・変更が可能な2値のデータで、可能アドレス範囲は1 から 9999 です。

(3) 入力ステータス (Input Status)

入力ステータスは、フィールドからの ON / OFF 入力である DI (Discrete Input) やスレーブデバイスの状態入力として用いられます。参照のみで変更はできない2値のデータで、可能アドレス範囲は10001 から 19999 です。

(4) 入力レジスタ (Input Register)

入力レジスタは、フィールドからの AI (Analog Input) やスレーブデバイス内の情報として用いられます。16 ビット長のデータで、参照するのみで変更はできません。可能アドレス範囲は30001 から 39999 です。複数の連続したアドレスを割当てることにより、単精度実数、倍精度実数などのデータを扱うこともできます。

(5) 保持レジスタ (Holding Register)

保持レジスタは、フィールドからの AO (Analog Output) やスレーブデバイス内の設定情報として用いられます。16 ビット長のデータで、参照・変更ができます。可能アドレス範囲は40001 から 49999 です。複数の連続したアドレスを割当てることによって、単精度実数、倍精度実数などのデータを扱うこともできます。

2、4 Modbus メッセージ構成

Modbus 伝送メッセージ上での、データアドレスの指定方法は、各データの先頭アドレスを0番地とします。例えば、入力レジスタアドレス 30156 (10 進数) を指定するには、メッセージ上でのアドレス指定は 155 (10 進数) となります。データの種類の、ファンクション・コードで決まります。

(表3) に Modbus クェリメッセージの例を示します。(表3) の例は、スレーブアドレス6 に対して保持レジスタの読出し (ファンクション・コード3) のクェリです。メッセージでは、40108 から 40110 までの3個の保持レジスタの内容を読出すリクエストメッセージです。開始アドレスの値が 107 (0X6B) になっていることに注意して下さい。

マスターからのクェリメッセージ構成例 (表3)

フィールド名	例 (Hex)	ASCII モード 文字列	RTU モード 8-ビット (Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X06	“ 0 ”, “ 6 ”	0X06
ファンクションコード	0X03	“ 0 ”, “ 3 ”	0X03
開始アドレス (上位)	0X00	“ 0 ”, “ 0 ”	0X00
開始アドレス (下位)	0X6B	“ 6 ”, “ B ”	0X6B
レジスタの数 (上位)	0X00	“ 0 ”, “ 0 ”	0X00
レジスタの数 (下位)	0X03	“ 0 ”, “ 3 ”	0X03
エラーチェック		LRC (2 文字)	CRC (16 ビット)
トレーラー		CR / LF	なし
	合計バイト数	17	8

(表4) に (表3) に対するスレーブからの応答メッセージ例を示します。スレーブアドレスとファンクションコードはそのままエコーバックします。データ数がバイト単位でいくつあるかを示します。(伝送上のバイトデータカウントではないことにご注意下さい。例では、16 ビットデータが3個ありますのでデータバイト数は、ASCII モードの場合でも RTU モードの場合でも同じ6 となります。)

スレーブからの応答メッセージ構成例 (表4)

フィールド名	例 (Hex)	ASCII モード 文字列	RTU モード 8-ビット (Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X06	“ 0 ”, “ 6 ”	0X06
ファンクションコード	0X03	“ 0 ”, “ 3 ”	0X03
データバイト数	0X06	“ 0 ”, “ 6 ”	0X06
データ1 (上位)	0X03	“ 0 ”, “ 3 ”	0X03
データ1 (下位)	0XE8	“ E ”, “ 8 ”	0XE8
データ2 (上位)	0X01	“ 0 ”, “ 1 ”	0X01
データ2 (下位)	0XF4	“ F ”, “ 4 ”	0XF4
データ3 (上位)	0X00	“ 0 ”, “ 0 ”	0X00
データ3 (下位)	0X0A	“ 0 ”, “ A ”	0X0A
エラーチェック		LRC (2 文字)	CRC (16 ビット)
トレーラー		CR / LF	なし
	合計バイト数	23	11

3、ファンクション・コード (Modbus ファンクション)

3、 1 Read Coil Status (01)

機能

スレーブの DO (Discrete Output) の ON / OFF 状態を読み出します。ブロードキャストはありません。アドレスとその内容については、各スレーブデバイスの仕様書をご参照下さい。

キューリ

キューリメッセージでは、コイルの開始アドレスとコイルの数を指定します。(表5)に、スレーブデバイス3に対してコイル20番地から56番地までの、37個のコイルのステータスを読む例を示します。開始アドレスの値が、1少ない19 (0X13) になっていることに注意して下さい。

・ Read Coil Status のキューリメッセージ例 (表5)

フィールド名	例 (Hex)	ASCII モード 文字列	RTU モード 8-ビット (Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X03	“ 0 ”, “ 3 ”	0X03
ファンクションコード	0X01	“ 0 ”, “ 1 ”	0X01
開始アドレス (上位)	0X00	“ 0 ”, “ 0 ”	0X00
開始アドレス (下位)	0X13	“ 1 ”, “ 3 ”	0X13
レジスターの数 (上位)	0X00	“ 0 ”, “ 0 ”	0X00
レジスターの数 (下位)	0X25	“ 2 ”, “ 5 ”	0X25
エラーチェック		LRC (2文字)	CRC (16ビット)
トレーラー		CR / LF	なし
	合計バイト数	17	8

レスポンス

レスポンスメッセージにおけるコイルのステータスはデータフィールドのデータに、1コイル当り1ビットのバックされます。ビットの意味は1 = ON、0 = OFFで、最初のデータのLSBが先頭アドレスのコイルのステータスになります。コイル20から27のステータスがON - ON - OFF - OFF - ON - OFF - ON - OFFとすると2進数で01010011 (0X53) となります。1バイトで8コイル分のステータスを表します。最後のデータでコイル数が8個に満たない場合には、残りには0を詰めます。(表5)のキューリに対するレスポンス例を(表6)に示します。

・ スレーブからの応答メッセージ例 (表6)

フィールド名	例 (Hex)	ASCII モード 文字列	RTU モード 8-ビット (Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X03	“ 0 ”, “ 3 ”	0X03
ファンクションコード	0X01	“ 0 ”, “ 1 ”	0X01
データバイト数	0X05	“ 0 ”, “ 5 ”	0X05
データ1	0X53	“ 5 ”, “ 3 ”	0X53
データ2	0X6B	“ 6 ”, “ B ”	0X6B
データ3	0X01	“ 0 ”, “ 1 ”	0X01
データ4	0XF4	“ F ”, “ 4 ”	0XF4
データ5	0X1B	“ 1 ”, “ B ”	0X1B
エラーチェック		LRC (2文字)	CRC (16ビット)
トレーラー		CR / LF	なし
	合計バイト数	21	10

3、2 Read Input Status (02)

機能

スレーブのDI (Discrete Input) の ON / OFF 状態を読出します。ブロードキャストはありません。アドレスとその内容については、各スレーブデバイスの仕様書をご参照下さい。

クエーリ

クエーリメッセージでは、入力ステータスの開始アドレスと入力ステータスの数を指定します。スレーブデバイス3 に対して入力ステータス 10101 番地から 10120 番地までの 20 個の入力ステータスのステータスを読む例を示します。開始アドレスの値が、10001 少ない 100 (0X64) になっていることに注意して下さい。

・ Read Coil Status のクエーリメッセージ例 (表 7)

フィールド名	例 (Hex)	ASCII モード 文字列	RTU モード 8- ビット (Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X03	“ 0 ”, “ 3 ”	0X03
ファンクションコード	0X02	“ 0 ”, “ 2 ”	0X02
開始アドレス (上位)	0X00	“ 0 ”, “ 0 ”	0X00
開始アドレス (下位)	0X64	“ 6 ”, “ 4 ”	0X64
レジスターの数 (上位)	0X00	“ 0 ”, “ 0 ”	0X00
レジスターの数 (下位)	0X14	“ 1 ”, “ 4 ”	0X14
エラーチェック		LRC (2 文字)	CRC (16 ビット)
トレーラー		CR / LF	なし
	合計バイト数	17	8

レスポンス

レスポンスメッセージにおけるデータの構成、意味は Read Coil Status (01) と同じです。(表 7) のクエーリに対するレスポンス例を (表 8) に示します。

・ スレーブからの応答メッセージ例 (表 8)

フィールド名	例 (Hex)	ASCII モード 文字列	RTU モード 8- ビット (Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X03	“ 0 ”, “ 3 ”	0X03
ファンクションコード	0X02	“ 0 ”, “ 2 ”	0X02
データバイト数	0X03	“ 0 ”, “ 3 ”	0X03
データ 1	0X53	“ 5 ”, “ 3 ”	0X53
データ 2	0X6B	“ 6 ”, “ B ”	0X6B
データ 3	0X01	“ 0 ”, “ 1 ”	0X01
エラーチェック		LRC (2 文字)	CRC (16 ビット)
トレーラー		CR / LF	なし
	合計バイト数	17	8

3、3 Read Holding Register (03)

機能

スレーブの保持レジスタの内容を読み出します。ブロードキャストはありません。
アドレスとその内容については、各スレーブデバイスの仕様書をご参照下さい。

クエーリ

クエーリメッセージでは、保持レジスタの開始アドレスと保持レジスタの数を指定します。(表9)に、スレーブデバイス7に対して保持レジスタ40201番地から40203番地までの3個の保持レジスタの内容を読み出す例を示します。開始アドレスの値が、40001少ない200(0XC8)になっていることに注意して下さい。

・ Read Holding Register のクエーリメッセージ例 (表9)

フィールド名	例 (Hex)	ASCIIモード 文字列	RTUモード 8-ビット(Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X07	“ 0 ”, “ 7 ”	0X07
ファンクションコード	0X03	“ 0 ”, “ 3 ”	0X03
開始アドレス(上位)	0X00	“ 0 ”, “ 0 ”	0X00
開始アドレス(下位)	0XC8	“ C ”, “ 8 ”	0XC8
レジスタの数(上位)	0X00	“ 0 ”, “ 0 ”	0X00
レジスタの数(下位)	0X03	“ 0 ”, “ 3 ”	0X03
エラーチェック		LRC (2文字)	CRC (16ビット)
トレーラー		CR / LF	なし
	合計バイト数	17	8

レスポンス

レスポンスメッセージにおけるデータは、1保持アドレス当り16ビットのデータです。(表9)のクエーリに対するレスポンス例を(表10)に示します。

・ スレーブからの応答メッセージ例 (表10)

フィールド名	例 (Hex)	ASCIIモード 文字列	RTUモード 8-ビット(Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X07	“ 0 ”, “ 7 ”	0X07
ファンクションコード	0X03	“ 0 ”, “ 3 ”	0X03
データバイト数	0X06	“ 0 ”, “ 6 ”	0X06
データ1(上位)	0X03	“ 0 ”, “ 3 ”	0X03
データ1(下位)	0XE8	“ E ”, “ 8 ”	0XE8
データ2(上位)	0X01	“ 0 ”, “ 1 ”	0X01
データ2(下位)	0XF4	“ F ”, “ 4 ”	0XF4
データ3(上位)	0X00	“ 0 ”, “ 0 ”	0X00
データ3(下位)	0X0A	“ 0 ”, “ A ”	0X0A
エラーチェック		LRC (2文字)	CRC (16ビット)
トレーラー		CR / LF	なし
	合計バイト数	23	11

3、4 Read Input Register (04)

機能

スレーブの入力レジスタの内容を読出します。ブロードキャストはありません。
アドレスとその内容については、各スレーブデバイスの仕様書をご参照下さい。

クエーリ

クエーリメッセージでは、入力レジスタの開始アドレスと入力レジスタの数を指定します。(表11)にスレーブデバイス7に対して入力レジスタ 30301 番地から 30303 番地までの3個の入力レジスタの内容を読む例を示します。開始アドレスの値が、30001 少ない300 (0X12C) になっていることに注意して下さい。

・ Read Input Register のクエーリメッセージ例 (表 11)

フィールド名	例 (Hex)	ASCII モード 文字列	RTU モード 8-ビット (Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X07	“ 0 ”, “ 7 ”	0X07
ファンクションコード	0X04	“ 0 ”, “ 4 ”	0X04
開始アドレス (上位)	0X01	“ 0 ”, “ 1 ”	0X01
開始アドレス (下位)	0X2C	“ 2 ”, “ C ”	0X2C
レジスタの数 (上位)	0X00	“ 0 ”, “ 0 ”	0X00
レジスタの数 (下位)	0X03	“ 0 ”, “ 3 ”	0X03
エラーチェック		LRC (2 文字)	CRC (16 ビット)
トレーラー		CR / LF	なし
	合計バイト数	17	8

レスポンス

レスポンスメッセージにおけるデータは、1入力アドレス当り16ビットのデータです。(表11)のクエーリに対するレスポンス例を (表 12) に示します。

・ スレーブからの応答メッセージ例 (表 12)

フィールド名	例 (Hex)	ASCII モード 文字列	RTU モード 8-ビット (Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X07	“ 0 ”, “ 7 ”	0X07
ファンクションコード	0X04	“ 0 ”, “ 4 ”	0X04
データバイト数	0X06	“ 0 ”, “ 6 ”	0X06
データ1 (上位)	0X03	“ 0 ”, “ 3 ”	0X03
データ1 (下位)	0XE8	“ E ”, “ 8 ”	0XE8
データ2 (上位)	0X01	“ 0 ”, “ 1 ”	0X01
データ2 (下位)	0XF4	“ F ”, “ 4 ”	0XF4
データ3 (上位)	0X00	“ 0 ”, “ 0 ”	0X00
データ3 (下位)	0X0A	“ 0 ”, “ A ”	0X0A
エラーチェック		LRC (2 文字)	CRC (16 ビット)
トレーラー		CR / LF	なし
	合計バイト数	23	11

3、5 Force Single Coil (05)

機能

スレーブの DO (Discrete Output) の状態を ON / OFF のいずれかに変更 (書込み) します。ブロードキャストの場合には、全スレーブの同じアドレスのコイルを書換えます。

アドレスとその内容については、各スレーブデバイスの仕様書をご参照下さい。

クエーリ

クエーリメッセージでは、コイルのアドレスと変更したいステータスを指定します。変更したいステータス (ON / OFF) は、クエーリのデータエリアで指定します。0xFF、0x00 は ON のリクエストで、0x00、0x00 は OFF のリクエストです。それ以外のデータは不正データで、変更動作は行われません。(表 13) に、スレーブデバイス 3 のコイル 150 番地を ON にする例を示します。開始アドレスの値が、1 少ない 149 (0x95) になっていることに注意して下さい。

・ Force Single Coil のクエーリメッセージ例 (表 13)

フィールド名	例 (Hex)	ASCII モード 文字列	RTU モード 8- ビット (Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X03	“ 0 ”, “ 3 ”	0X03
ファンクションコード	0X05	“ 0 ”, “ 5 ”	0X05
開始アドレス (上位)	0X00	“ 0 ”, “ 0 ”	0X00
開始アドレス (下位)	0X95	“ 9 ”, “ 5 ”	0X95
変更データ (上位)	0XFF	“ F ”, “ F ”	0XFF
変更データ (下位)	0X00	“ 0 ”, “ 0 ”	0X00
エラーチェック トレーラー		LRC (2 文字) CR / LF	CRC (16 ビット) なし
	合計バイト数	17	8

レスポンス

正常に変更された場合の正常メッセージは、クエーリと同じになります。(表 14) に、(表 13) に対する正常応答メッセージを示します。

・スレーブからの正常応答メッセージ例 (表 14)

フィールド名	例 (Hex)	ASCII モード 文字列	RTU モード 8- ビット (Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X03	“ 0 ”, “ 3 ”	0X03
ファンクションコード	0X05	“ 0 ”, “ 5 ”	0X05
アドレス (上位)	0X00	“ 0 ”, “ 0 ”	0X00
アドレス (下位)	0X95	“ 9 ”, “ 5 ”	0X95
変更データ (上位)	0XFF	“ F ”, “ F ”	0XFF
変更データ (下位)	0X00	“ 0 ”, “ 0 ”	0X00
エラーチェック トレーラー		LRC (2 文字) CR / LF	CRC (16 ビット) なし
	合計バイト数	17	8

3、6 Preset Single Register (06)

機能

スレーブの保持レジスタの内容を変更（書込み）します。ブロードキャストの場合には、全スレーブの同じアドレスの保持レジスタの内容が変更されます。

保持レジスタのアドレスとその内容については、各スレーブデバイスの仕様書をご参照下さい。

クエーリ

クエーリメッセージでは、保持レジスタのアドレスと変更したいデータを指定します。変更したいデータは、クエーリのデータエリアで16ビットのデータとして指定します。（表15）に、スレーブデバイス3の保持レジスタ150番地を1000にする例を示します。アドレスの値が、1少ない149（0X95）になっていることに注意して下さい。

・Preset Single Registerのクエーリメッセージ例（表15）

フィールド名	例 (Hex)	ASCIIモード 文字列	RTUモード 8-ビット(Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X03	“ 0 ”, “ 3 ”	0X03
ファンクションコード	0X06	“ 0 ”, “ 6 ”	0X06
アドレス(上位)	0X00	“ 0 ”, “ 0 ”	0X00
アドレス(下位)	0X95	“ 9 ”, “ 5 ”	0X95
変更データ(上位)	0X03	“ 0 ”, “ 3 ”	0X03
変更データ(下位)	0XE8	“ E ”, “ 8 ”	0XE8
エラーチェック		LRC (2文字)	CRC (16ビット)
トレーラー		CR / LF	なし
	合計バイト数	17	8

レスポンス

正常に変更された場合の正常メッセージは、クエーリと同じになります。（表16）に、（表15）に対する正常応答メッセージを示します。

・スレーブからの正常応答メッセージ例（表16）

フィールド名	例 (Hex)	ASCIIモード 文字列	RTUモード 8-ビット(Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X03	“ 0 ”, “ 3 ”	0X03
ファンクションコード	0X06	“ 0 ”, “ 6 ”	0X06
アドレス(上位)	0X00	“ 0 ”, “ 0 ”	0X00
アドレス(下位)	0X95	“ 9 ”, “ 5 ”	0X95
変更データ(上位)	0X03	“ 0 ”, “ 3 ”	0X03
変更データ(下位)	0XE8	“ E ”, “ 8 ”	0XE8
エラーチェック		LRC (2文字)	CRC (16ビット)
トレーラー		CR / LF	なし
	合計バイト数	17	8

3、7 Diagnostics (08)

機能

マスターとスレーブ間の通信の診断やスレーブの機器の診断ファンクションです。ブロードキャストは、ありません。診断のタイプを定義する為に、クエリの中に2バイトの診断サブコードフィールドがあります。スレーブからの正常なレスポンスには、ファンクションコードと共にこのサブコードもエコーバックします。また、クエリにはスレーブのコントロールや診断の為にデータを渡す為に、2バイトのデータがあります。

診断サブファンクションとその内容については、「4、診断 (0X08)」の項をご参照下さい。

クエリ

(表17)は、スレーブデバイス5とのクエリのリードバック診断の例です。リードバック診断のサブコードは、0 (0X0000) です。

・Diagnostics のクエリメッセージ例 (表 17)

フィールド名	例 (Hex)	ASCII モード 文字列	RTU モード 8-ビット (Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X05	“ 0 ”, ’5 ”	0X05
ファンクションコード	0X08	“ 0 ”, ’8 ”	0X08
診断サブコード (上位)	0X00	“ 0 ”, ’0 ”	0X00
診断サブコード (下位)	0X00	“ 0 ”, ’0 ”	0X00
データ (上位)	0X03	“ 0 ”, ’3 ”	0X03
データ (下位)	0XE8	“ E ”, ’8 ”	0XE8
エラーチェック トレーラー		LRC (2 文字) CR / LF	CRC (16 ビット) なし
	合計バイト数	17	8

レスポンス

(表 17) のリードバック診断の正常レスポンスは、クエリと同じになります。

・スレーブからの正常応答メッセージ例 (表 18)

フィールド名	例 (Hex)	ASCII モード 文字列	RTU モード 8-ビット (Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X05	“ 0 ”, ’5 ”	0X05
ファンクションコード	0X08	“ 0 ”, ’8 ”	0X08
診断サブコード (上位)	0X00	“ 0 ”, ’0 ”	0X00
診断サブコード (下位)	0X00	“ 0 ”, ’0 ”	0X00
データ (上位)	0X03	“ 0 ”, ’3 ”	0X03
データ (下位)	0XE8	“ E ”, ’8 ”	0XE8
エラーチェック トレーラー		LRC (2 文字) CR / LF	CRC (16 ビット) なし
	合計バイト数	17	8

3、 8 Fetch Communication Event Counter (11 , 0XB)

機能

スレーブの通信イベントカウンタからステータスワードとイベントカウンタを読出します。メッセージ通信の前後でこのカウンタを読むことで、マスタはスレーブが正しくメッセージを処理しているかが分かります。ブロードキャストはありません。

コントローラはメッセージを正しく処理した場合、イベントをカウントアップします。例外レスポンス、ポールコマンドおよび本コマンドの場合にはカウントアップしません。イベントカウンタは診断ファンクションのサブコード Restart Communication Option (コード 0X0001) と、Clear Counters and Diagnostic Register (0X000A) でリセットされます。

クエーリ

(表 19) は、スレーブデバイス 5 への Fetch Communication Event Counter の例です。

・ Fetch Communication Event Counter のクエーリ (表 19)

フィールド名	例 (Hex)	ASCII モード 文字列	RTU モード 8-ビット (Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X05	“ 0 ”, “ 5 ”	0X05
ファンクションコード	0X0B	“ 0 ”, “ B ”	0X0B
エラーチェック		LRC (2 文字)	CRC (16 ビット)
トレーラー		CR / LF	なし
	合計バイト数	9	4

レスポンス

Fetch Communication Event Counter の正常レスポンスは、2 バイトのステータスワードと 2 バイトのイベントカウンタが含まれます。(表 20) に、(表 19) に対する応答メッセージの例を示します。

・ スレーブからの正常応答メッセージ例 (表 20)

フィールド名	例 (Hex)	ASCII モード 文字列	RTU モード 8-ビット (Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X05	“ 0 ”, “ 5 ”	0X05
ファンクションコード	0X0B	“ 0 ”, “ B ”	0X0B
ステータス (上位)	0X00	“ 0 ”, “ 0 ”	0X00
ステータス (下位)	0X00	“ 0 ”, “ 0 ”	0X00
イベントカウンタ (上位)	0X03	“ 0 ”, “ 3 ”	0X03
イベントカウンタ (下位)	0XE8	“ E ”, “ 8 ”	0XE8
エラーチェック		LRC (2 文字)	CRC (16 ビット)
トレーラー		CR / LF	なし
	合計バイト数	17	8

3、9 Fetch Communication Event Log (12 , 0XC)

機能

スレーブの通信イベントログ(ステータスワード、イベントカウンタ、メッセージカウンタおよびイベント)を読み出します。ブロードキャストはありません。

メッセージカウンタはパワーアップ、リスタート、カウンタリセット以降、スレーブが発行したメッセージをカウントアップします。

メッセージカウンタは診断ファクションのサブコード Restart Communication Option (コード 0X0001) と、Clear Counters and Diagnostic Register (0X000A) でリセットされます。

クエーリ

(表21) は、スレーブデバイス5へのFetch Communication Event Logの例です。

・Fetch Communication Event Logのクエーリ(表21)

フィールド名	例 (Hex)	ASCIIモード 文字列	RTUモード 8-ビット(Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X05	“ 0 ”, “ 5 ”	0X05
ファンクションコード	0X0C	“ 0 ”, “ C ”	0X0C
エラーチェック		LRC (2文字)	CRC (16ビット)
トレーラー		CR / LF	なし
	合計バイト数	9	4

レスポンス

Fetch Communication Event Logの正常レスポンスは、2バイトのステータスワード、2バイトのイベントカウンタ、2バイトのメッセージカウンタおよび0から64バイトのイベントからなります。可変長のデータなのでデータ数を示すバイトカウントが含まれます。(表22)に、(表21)に対する応答メッセージの例を示します。

・スレーブからの正常応答メッセージ例(表22)

フィールド名	例 (Hex)	ASCIIモード 文字列	RTUモード 8-ビット(Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X05	“ 0 ”, “ 5 ”	0X05
ファンクションコード	0X0C	“ 0 ”, “ C ”	0X0C
バイトカウント	0X08	“ 0 ”, “ 8 ”	0X08
ステータス(上位)	0X00	“ 0 ”, “ 0 ”	0X00
ステータス(下位)	0X00	“ 0 ”, “ 0 ”	0X00
イベントカウンタ(上位)	0X01	“ 0 ”, “ 1 ”	0X01
イベントカウンタ(下位)	0XE8	“ E ”, “ 8 ”	0XE8
メッセージカウンタ(上位)	0X01	“ 0 ”, “ 1 ”	0X01
メッセージカウンタ(下位)	0XF6	“ F ”, “ 6 ”	0XF6
イベント0	0X20	“ 2 ”, “ 0 ”	0X20
イベント1	0X00	“ 0 ”, “ 0 ”	0X00
エラーチェック		LRC (2文字)	CRC (16ビット)
トレーラー		CR / LF	なし
	合計バイト数	25	12

3、10 Force Multiple Coils (15 , 0X0F)

機能

スレーブの連続した複数の DO (Discrete Output) の状態を、ON / OFF のいずれかに変更 (書込み) します。ブロードキャストの場合には、全スレーブの同じアドレスのコイルを書換えます。

アドレスとその内容については、各スレーブデバイスの仕様書をご参照下さい。

キューリ

キューリメッセージでは、コイルの先頭アドレスと変更したいステータスを指定します。変更したいステータス (ON / OFF) は、キューリのデータエリアで指定します。コイルの ON / OFF とデータエリアのデータ構成は Read Coil Status をご参照下さい。(表 23) ではスレーブデバイス 5 のコイル 20 番地から 30 番地を、以下のようにセットする場合の例を示します。

```
ビット  1  1  0  1  0  0  0  1  0  0  0  0  0  1  0  1
コイル  27 26 25 24 23 22 21 20  . . . . . 30 29 28
```

開始アドレスの値が、1 少ない 19 (0X13) になっていることに注意して下さい。

・ Force Multiple Coils のキューリメッセージ例 (表 23)

フィールド名	例 (Hex)	ASCII モード 文字列	RTU モード 8- ビット (Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X05	“ 0 ”, ‘5’	0X05
ファンクションコード	0X0F	“ 0 ”, ‘F’	0X0F
開始アドレス (上位)	0X00	“ 0 ”, ‘0’	0X00
開始アドレス (下位)	0X13	“ 1 ”, ‘3’	0X13
レジスターの数 (上位)	0X00	“ 0 ”, ‘0’	0X00
レジスターの数 (下位)	0X0B	“ 0 ”, ‘B’	0X0B
バイト数	0X02	“ 0 ”, ‘2’	0X02
変更データ (上位)	0XD1	“ D ”, ‘1’	0XD1
変更データ (下位)	0X05	“ 0 ”, ‘5’	0X05
エラーチェック		LRC (2 文字)	CRC (16 ビット)
トレーラー		CR / LF	なし
	合計バイト数	23	11

レスポンス

正常に変更された場合の正常メッセージは、キューリの中のバイト数とデータエリアを除いた部分のコピーをレスポンスします。(表 24) に、(表 23) に対する正常応答メッセージを示します。

・ スレーブからの正常応答メッセージ例 (表 24)

フィールド名	例 (Hex)	ASCII モード 文字列	RTU モード 8- ビット (Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X05	“ 0 ”, ‘5’	0X05
ファンクションコード	0X0F	“ 0 ”, ‘F’	0X0F
開始アドレス (上位)	0X00	“ 0 ”, ‘0’	0X00
開始アドレス (下位)	0X13	“ 1 ”, ‘3’	0X13
レジスターの数 (上位)	0X00	“ 0 ”, ‘0’	0X00
レジスターの数 (下位)	0X0B	“ 0 ”, ‘B’	0X0B
エラーチェック		LRC (2 文字)	CRC (16 ビット)
トレーラー		CR / LF	なし
	合計バイト数	17	8

3、11 Preset Multiple Registers (16 , 0X10)

機能

スレーブの連続した複数の保持レジスタの内容を変更（書込み）します。ブロードキャストの場合には、全スレーブの同じアドレスの保持レジスタの内容を書換えます。

アドレスとその内容については、各スレーブデバイスの仕様書をご参照下さい。

クエーリ

クエーリメッセージでは、保持レジスタの内容の先頭アドレスと変更したいデータを指定します。変更したいデータは、クエーリのデータエリアで指定します。(表25)ではスレーブデバイス5の保持レジスタ 40020 番地から 40022 番地を以下のようにセットする場合の例を示します。

40020 番地 データ 0X0164

40021 番地 データ 0X0165

40022 番地 データ 0X0166

開始アドレスの値が、40001 少ない 19 (0X13) になっていることに注意して下さい。

・Preset Multiple Registers のクエーリメッセージ例 (表 25)

フィールド名	例 (Hex)	ASCII モード 文字列	RTU モード 8-ビット (Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X05	“ 0 ”, “ 5 ”	0X05
ファンクションコード	0X10	“ 1 ”, “ 0 ”	0X10
開始アドレス (上位)	0X00	“ 0 ”, “ 0 ”	0X00
開始アドレス (下位)	0X13	“ 1 ”, “ 3 ”	0X13
レジスタの数 (上位)	0X00	“ 0 ”, “ 0 ”	0X00
レジスタの数 (下位)	0X03	“ 0 ”, “ 3 ”	0X03
バイト数	0X06	“ 0 ”, “ 6 ”	0X06
変更データ 1 (上位)	0X01	“ 0 ”, “ 1 ”	0X01
変更データ 1 (下位)	0X64	“ 6 ”, “ 4 ”	0X64
変更データ 2 (上位)	0X01	“ 0 ”, “ 1 ”	0X01
変更データ 2 (下位)	0X65	“ 6 ”, “ 5 ”	0X65
変更データ 3 (上位)	0X01	“ 0 ”, “ 1 ”	0X01
変更データ 3 (下位)	0X66	“ 6 ”, “ 6 ”	0X66
エラーチェック		LRC (2 文字)	CRC (16 ビット)
トレーラー		CR / LF	なし
	合計バイト数	31	15

レスポンス

正常に変更された場合の正常メッセージは、クエリの中のバイト数とデータエリアを除いた部分のコピーをレスポンスします。(表26)に、(表25)に対する正常応答メッセージを示します。

・スレーブからの正常応答メッセージ例 (表26)

フィールド名	例 (Hex)	ASCIIモード 文字列	RTUモード 8-ビット(Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X05	“ 0 ”, “ 5 ”	0X05
ファンクションコード	0X10	“ 1 ”, “ 0 ”	0X10
開始アドレス(上位)	0X00	“ 0 ”, “ 0 ”	0X00
開始アドレス(下位)	0X13	“ 1 ”, “ 3 ”	0X13
レジスターの数(上位)	0X00	“ 0 ”, “ 0 ”	0X00
レジスターの数(下位)	0X03	“ 0 ”, “ 3 ”	0X03
エラーチェック		LRC (2文字)	CRC (16ビット)
トレーラー		CR / LF	なし
	合計バイト数	17	8

3、12 Report Slave ID (17, 0X11)

機能

スレーブのコントローラタイプや動作モードなどスレーブの情報を読み出します。ブロードキャストはありません。

レスポンスメッセージの構成はデバイスによって異なります。

クエリ

(表27)は、スレーブデバイス5へのReport Slave IDの例です。

・Report Slave IDのクエリメッセージ例 (表27)

フィールド名	例 (Hex)	ASCIIモード 文字列	RTUモード 8-ビット(Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X05	“ 0 ”, “ 5 ”	0X03
ファンクションコード	0X11	“ 1 ”, “ 1 ”	0X11
エラーチェック		LRC (2文字)	CRC (16ビット)
トレーラー		CR / LF	なし
	合計バイト数	9	4

レスポンス

Report Slave ID の一般的なレスポンスは、スレーブ ID、RUN インディケータおよびデバイス依存の付加情報から構成されます。(表 28) に、応答メッセージの一般的な構成例を示します。

・スレーブからの正常応答メッセージ構成例 (表 28)

フィールド名	例 (Hex)	ASCII モード 文字列	RTU モード 8-ビット (Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X05	“ 0 ”, “ 5 ”	0X05
ファンクションコード	0X11	“ 1 ”, “ 1 ”	0X11
バイトカウント	デバイス依存	デバイス依存	デバイス依存
スレーブ ID	デバイス依存	デバイス依存	デバイス依存
RUN インディケータ	0XFF	“ F ”, “ F ”	0XFF
付加情報 1	デバイス依存	デバイス依存	デバイス依存
.....	デバイス依存	デバイス依存	デバイス依存
エラーチェック		LRC (2 文字)	CRC (16 ビット)
トレーラー		CR / LF	なし
	合計バイト数	デバイス依存	デバイス依存

4、診断 (0X08)

診断サブコードと診断内容は以下の通りです。

4、1 Return Query Data (00)

診断内容	クエリのデータフィールドで渡されたデータをレスポンスする
サブコード	0X00, 0X00
クエリのデータフィールド	任意の 16 ビットデータ
レスポンスのデータフィールド	クエリデータのエコー

4、2 Restart Communications Option (01)

診断内容	通信ポートを初期化し、コミュニケーションイベントカウンタをクリアする。レスポンスは初期化の前に行う。Listen Only Mode のときも動作するが、レスポンスは返さない。	
サブコード	0X00, 0X01	
クエリのデータフィールド	0X00, 0X00	イベントログは残す
	0XFF, 0X00	イベントログもクリアする
レスポンスのデータフィールド	クエリデータのエコー	

4、3 Return Diagnostics Register (02)

診断内容	16 ビットの診断レジスタの内容を返す。
サブコード	0X00, 0X02
クエリのデータフィールド	0X00, 0X00
レスポンスのデータフィールド	診断レジスタの内容

4、 4 Force Listen Only Mode (04)

診断内容	スレーブを受信オンリーモードにする。 全てのメッセージを無視し、アクションやレスポンスを行わない。 ただし、診断サブコード1のみは、受付け、通信を初期化し再スタートし、受信オンリーモードを解除する。
サブコード	0X00 , 0X04
クエリのデータフィールド	0X00 , 0X00
レスポンスのデータフィールド	レスポンスを返さない。

4、 5 Clear Counters and Diagnostic Register (10 , 0X0A)

診断内容	全てのカウンタと診断レジスタをクリアします。
サブコード	0X00 , 0X0A
クエリのデータフィールド	0X00 , 0X00
レスポンスのデータフィールド	クエリのエコーを返す。

4、 6 Return Bus Message Count (11 , 0X0B)

診断内容	スレーブが検知したメッセージの合計を讀出す。
サブコード	0X00 , 0X0B
クエリのデータフィールド	0X00 , 0X00
レスポンスのデータフィールド	メッセージの合計

4、 7 Return Bus Communication Error Count (12 , 0X0C)

診断内容	スレーブが検知したCRCエラーの合計を讀出す。
サブコード	0X00 , 0X0C
クエリのデータフィールド	0X00 , 0X00
レスポンスのデータフィールド	CRCエラーの合計

4、 8 Return Bus Exception Error Count (13 , 0X0D)

診断内容	スレーブが発行した例外レスポンスの合計を讀出す。
サブコード	0X00 , 0X0D
クエリのデータフィールド	0X00 , 0X00
レスポンスのデータフィールド	例外レスポンスの合計

4、 9 Return Slave Message Count (14 , 0X0E)

診断内容	当該スレーブ向けの受信メッセージの合計を讀出す。
サブコード	0X00 , 0X0E
クエリのデータフィールド	0X00 , 0X00
レスポンスのデータフィールド	スレーブ・受信メッセージの合計

4、 10 Return Slave No Response Count (15 , 0X0F)

診断内容	当該スレーブ向けのメッセージで、レスポンスメッセージを返さなかったメッセージの合計を讀出す。
サブコード	0X00 , 0X0F
クエリのデータフィールド	0X00 , 0X00
レスポンスのデータフィールド	無応答メッセージの合計

4、 11 Return Slave Busy Count (17 , 0X11)

診断内容	スレーブが発行したスレーブ・ビジー・例外レスポンスの合計を讀出す。
サブコード	0X00 , 0X11
クエリのデータフィールド	0X00 , 0X00
レスポンスのデータフィールド	スレーブ・ビジー・例外レスポンスの合計

4、 12 Return Bus Character Overrun Count (18 , 0X12)

診断内容	当該スレーブ向けのメッセージで、キャラクタ・オーバーランのエラーを検出した、キャラクタ・オーバーラン・カウンタ値を讀出す。
サブコード	0X00 , 0X12
クエリのデータフィールド	0X00 , 0X00
レスポンスのデータフィールド	キャラクタ・オーバーラン・カウンタ値

5、例外レスポンス (Exception Response)

ブロードキャスト・メッセージ以外のクエリの場合、マスターは正常なレスポンスを期待してクエリを発行します。それに対して、スレーブはレスポンスを返さなければなりません。正常な場合には、正常レスポンスを返しますが、以下のような異常が発生した場合、例外レスポンス (Exception Response) を返します。

クエリに対するスレーブの応答の方法には、以下の4種類があります。

- 正常にクエリを受信し、正常に処理を行い、正常レスポンスを返します。
- 通信エラー等により、スレーブはクエリを受信することができなく、無応答になります。マスターはタイムアウトエラーになります。
- スレーブはクエリを受信することができたが、パリティ、CRCおよびLRCエラーを検出し、正しいクエリではない。この場合にはスレーブは、無応答になります。従って、マスター側はタイムアウトになります。
- スレーブは、エラーもなく正しいクエリを受信できたが、何らかの理由 (例えば、当該レジスタは存在しないなど) で、処理できない。この場合には、例外の内容を示す例外コード (Exception Code) をつけて例外レスポンスを返します。

例外レスポンスは、スレーブアドレス、ファンクションコードおよびデータ・フィールドより構成されています。スレーブアドレス・フィールドには、正常レスポンスと同じように、スレーブのアドレスをセットします。ファンクションコード・フィールドには、クエリのファンクションコードをセットし更に、そのMSBを1にします。これによりマスターは、正常レスポンスではなく例外レスポンスであることを検知できます。データフィールドには、例外の内容を示す例外コードがセットされます。(表29)にクエリと対応する例外レスポンスの例を示します。

Read Input Register のクエリメッセージ例 (表29)

フィールド名	例 (Hex)	ASCII モード 文字列	RTU モード 8-ビット (Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X07	“ 0 ”, “ 7 ”	0X07
ファンクションコード	0X04	“ 0 ”, “ 4 ”	0X04
開始アドレス (上位)	0X01	“ 0 ”, “ 1 ”	0X01
開始アドレス (下位)	0X2C	“ 2 ”, “ C ”	0X2C
レジスタの数 (上位)	0X00	“ 0 ”, “ 0 ”	0X00
レジスタの数 (下位)	0X03	“ 0 ”, “ 3 ”	0X03
エラーチェック トレーラー		LRC (2文字) CR / LF	CRC (16ビット) なし
	合計バイト数	17	8

入力レジスタ 30301 が存在しない場合、(表30)の例外レスポンスが返されます。

スレーブからの例外レスポンス例 (表30)

フィールド名	例 (Hex)	ASCII モード 文字列	RTU モード 8-ビット (Hex)
ヘッダー		“ : ”	なし
スレーブアドレス	0X07	“ 0 ”, “ 7 ”	0X07
ファンクションコード	0X84	“ 8 ”, “ 4 ”	0X84
例外コード	0X02	“ 0 ”, “ 2 ”	0X02
エラーチェック トレーラー		LRC (2文字) CR / LF	CRC (16ビット) なし
	合計バイト数	11	5

例外コードとその発生要因を以下の表に示します。

例外コード	名 称	意 味
01	不正ファンクション	スレーブは当該ファンクションをサポートしていない。
02	不正データアドレス	指定されたデータアドレスは、スレーブには存在しない。
03	不正データ	指定されたデータは、許されない。

補 足 1 Modbus 通信機能

以下に、機器の Modbus 通信機能を抜粋して掲載します。掲載していないその他の機器については、各機器の仕様書をご参照下さい。

1、R1M シリーズの Modbus 通信機能

R1M シリーズでは、伝送モードとして RTU のみをサポートしており、ASCII モードはサポートしていません。

1、 1 ファンクション・コード

(表 31) に R1M シリーズでサポートするファンクション・コードを示します。

ファンクションコード一覧表 (表 31)

コード	ファンクション名 (仕様書 PI-MBUS-300 より)		機 能
01 (0X01)	Read Coil Status		コイル、DO の読出し
02 (0X02)	Read Input Status		入力ステータス、DI の読出し
03 (0X03)	Read Holding Register		保持レジスタの読出し
04 (0X04)	Read Input Register		入力レジスタの読出し
05 (0X05)	Force Single Coil		コイル、DO への 1 点書込み
06 (0X06)	Preset Single Register		保持レジスタへの書込み
15 (0X0F)	Force Multiple Coils		複数コイル、DO への一括書込み
16 (0X10)	Force Multiple Registers		複数保持レジスタへの一括書込み

1、 2 データアドレス

	アドレス	タイプ				データ 形式	データ名称
		GH2	J3	A1	D1		
コイル (0X)	1 - 32						DO
	33 - 48						冷接点温度補償 SW
入力 ステータス (1X)	1 - 32						DI
	33 - 48						ADC レンジオーバー
入力 レジスタ (3X)	1 - 16					I	アナログ入力 % 値
	17 - 48					F	チャンネル毎アナログ入力値 (工業単位値)
	49 - 50					F	冷接点温度値
	81 - 96					I	チャンネル・ステータス
	513					I	システム・ステータス
	514 - 521					B16	形式
	522 - 529					B16	機番
	530 - 537					B16	ハードウェア・バージョン番号
538 - 545					B16	ファームウェア・バージョン番号	
保持 レジスタ (4X)	1 - 16					I	(予備：アナログ出力 % 値)
	17 - 48					F	(予備：アナログ出力工業単位値)
	145 - 160					I	入出力タイプ番号
	161 - 176					I	バーンアウトタイプ

(注) I : 16 ビット長整数、F : 32 ビット長実数、B16 : 16 バイト長文字列

1、3 コイル (DO) データ詳細

(1) DO (1 ~ 32)

32 点の DO (Discrete Output) です。D1 タイプのみ有効です。

(2) 冷接点温度補償 SW (33 ~ 48)

冷接点温度補償を行うか否かの指示ビットです。1 の場合温度補償動作を行います。また、入力タイプが熱電対の場合のみ有効です。

1、4 入力ステータス (DI) データ詳細

(1) DI (10001 ~ 10032)

32 点の DI (Discrete Input) です。A1、GH2 および J3 のとき有効です。GH2、J3 の場合アドレス 10001 のみが有効で、内容はトリガー SW 入力です。

(2) ADC レンジオーバー (10033 ~ 10048)

アナログ入力が ADC のレンジオーバー (0X0000 または 0XFFFF) になっていることを示します。GH2、J3 のとき有効です。

1、5 入力レジスタデータ詳細

(1) アナログ入力 % 値 (30001 ~ 30016)

各入力チャンネル (1 ~ 16) のアナログ入力値です。

入力の種類	対応する A / D 値(10 進)
± 20 V	± 20000
± 5 V	± 5000
± 1 V	± 10000
± 0.8 V	± 8000
± 0.2 V	± 20000
± 50 mV	± 5000
± 10 mV	± 10000
熱電対入力	検出温度の 10 倍の値

(2) アナログ入力工業単位値 (30017 ~ 30048)

各入力チャンネル (1 ~ 16) のアナログ入力値です。値は工業単位量でその単位は入力のタイプで決定されます。温度の場合には、電圧の場合には V、およびポテンシオメータの場合には % 値になります。データ形式は 32 ビット長の実数で表現されます。従って、入力レジスタは各モジュールに対して連続した 2 個を使用します。

(3) 冷接点温度 (30049 から 30050)

冷接点の現在の温度 () を示します。GH2 のみ有効です。データ形式は 32 ビット長実数です。

(4) チャンネル・ステータス (30081 ~ 30096)

アナログ入力チャンネル (GH2 の場合 1 ~ 16、J3 の場合 1 ~ 8) の状態を示します。各ビットの構成と意味は以下の通りです。

ビット番号	内 容	詳 細
6	入力レンジオーバー	入力がレンジオーバーしているか否かを示します。 以下の状態を検知するとレンジオーバーとなります。 ・ADC 入力値が 0X0000 または 0XFFFF ・熱電対の場合バーンアウトを検出 ・熱電対および測温抵抗値の場合 温度変換テーブルの範囲外になっている。 ・ADC 異常 0 : 正常 1 : レンジオーバー
7	ADC エラー	ADC の状態を示します。 0 : 正常 1 : 異常
12	冷接点温度補償 SW	熱電対の場合の冷接点温度補償状態を示します。 0 : 補償なし 1 : 補償あり
上記以外	予備	システム使用

(5) システム・ステータス (30513)

R1M シリーズのステータスを、16 ビット長整数で示します。各ビットの構成と意味は以下の通りです。

ビット番号	内 容	詳 細
0 ~ 3	予備	システム使用
4	予備	システム使用
5	予備	システム使用
6	E ² PROM 診断	モジュールの E ² PROM サムチェック診断結果を示します。 0 : 正常 1 : サムチェックエラーです。
7	ADC エラー	ADC の状態を示します。 0 : 正常 1 : 異常
8 ~ 15	予備	システム使用

(6) 形式 (30514 ~ 30521)

R1M シリーズの形式を、16 バイト長の文字列で示します。

(7) 機番 (30522 ~ 30529)

R1M シリーズの機番 (シリアル番号) を、16 バイト長の文字列で示します。

(8) ハードウェア・バージョン番号 (30530 ~ 30537)

R1M シリーズのハードウェア・バージョン番号を、16 バイト長の文字列で示します。

(9) ファームウェア・バージョン番号 (30538 ~ 30545)

R1M シリーズのファームウェア・バージョン番号を、16 バイト長の文字列で示します。

1、6 保持レジスタ・データ詳細

(1) アナログ出力 % 値 (40001 ~ 40016)

各出力チャンネル (1 ~ 16) のアナログ出力 % 値です。将来の予備です。

(2) アナログ出力工業単位値 (40017 ~ 40048)

各出力チャンネル (1 ~ 16) のアナログ出力値です。値は工業単位量でその単位は出力のタイプで決定されます。データ形式は32ビット長の実数で表現されます。従って、保持レジスタは各チャンネルに対して連続した2個を使用します。将来の予備です。

(3) 入出力タイプ番号 (40145 ~ 40160)

各入力チャンネル (1 ~ 16) の入出力タイプ番号です。入出力タイプ番号を16ビット整数で示します。

形 式	入出力タイプ	入出力タイプ番号	入力レンジ名称	測定可能レンジ	備 考
R1M - GH2	DC 入力	0X00	-20 to 20 V	-22.7 to 22.7 V	ATT SW ON
		0X01	-5 to 5 V	-5.6 to 5.6 V	ATT SW ON
		0X02	-1 to 1 V	-1.4 to 1.4 V	ATT SW ON
		0X03	-800 to 800 mV	-860 to 860 mV	
		0X04	-200 to 200 mV	-215 to 215 mV	
		0X05	-50 to 50 mV	-53 to 53 mV	
		0X06	-10 to 10 mV	-13.4 to 13.4 mV	
	熱電対入力	0X10	(PR)	0 to 1760	
		0X11	K (CA)	-270 to 1370	
		0X12	E (CRC)	-270 to 1000	
		0X13	J (IC)	-210 to 1200	
		0X14	T (CC)	-270 to 400	
		0X15	B (RH)	200 to 1820	
		0X16	R	-50 to 1760	
		0X17	S	-50 to 1760	
		0X18	C (WRe 5-26)	0 to 2315	
		0X19	N	-270 to 1300	
		0X1A	U	-200 to 400	
		0X1B	L	-200 to 900	
		0X1C	P (PlatineI2)	0 to 1395	
R1M - J3	測温抵抗体入力	0X30	JPt 100 (JIS 89)	-200 to 500	
		0X31	Pt 100 (JIS 89)	-200 to 850	
		0X32	Pt 100 (JIS 97)	-200 to 850	
		0X33	Pt 50 (JIS 81)	-200 to 649	
		0X34	Ni 508.4	-50 to 200	
		0X35	Pt 1000	-200 to 850	
	ポテンショメータ	0X40	0 to 100 ohm	0 to 100 %	
		0X41	0 to 500 ohm	0 to 100 %	
		0X42	0 to 1 Kohm	0 to 100 %	
		0X43	0 to 10 Kohm	0 to 100 %	
	R1M - D1	DI	0X60		
	R1M - A1	DO	0X70		

(4) バーンアウトタイプ (40161 ~ 40176)

バーンアウト検出時のバーンアウト動作タイプを指定します。熱電対入力のみ有効です。

0 : 無指定

1 : Up Scale

2 : Down Scale

2、R2M シリーズの Modbus 通信機能

2、1 通信パラメータ

(表 32) に R2M シリーズがサポートする通信パラメータを示します。

通信パラメータ (表 32)

通信パラメータ	仕様	工場出荷時設定	設定方法
伝送モード	RTU	RTU	変更不可
伝送速度	9600 / 19200 / 38400 bps	38400 bps	R2CON によるソフト設定
パリティ	None / ODD / EVEN	ODD	R2CON によるソフト設定
ビット長	8	8	変更不可
ストップビット	1	1	変更不可
ノードアドレス	1 - 247	1	1 - 15 まではハードウェアのロータリ SW。16 - 247 までは R2CON によるソフト設定(このときロータリ SW は 0 にする)
浮動小数点データの順序	Normal / Swapped	Normal	R2CON によるソフト設定
接続	RS-232-C	RS-232-C	変更不可

2、2 ファンクション・コード

(表 33) に R2M シリーズがサポートするファンクション・コードを示します。

ファンクションコード一覧表 (表 33)

コード	ファンクション名	機能
	(仕様書 PI-MBUS-300 より)	
01 (0X01)	Read Coil Status	コイル、DO の読出し
02 (0X02)	Read Input Status	入力ステータス、DI の読出し
03 (0X03)	Read Holding Register	保持レジスタの読出し
04 (0X04)	Read Input Register	入力レジスタの読出し
05 (0X05)	Force Single Coil	コイル、DO への 1 点書込み
06 (0X06)	Preset Single Register	保持レジスタへの書込み
15 (0X0F)	Force Multiple Coils	複数コイル、DO への一括書込み
16 (0X10)	Force Multiple Registers	複数保持レジスタへの一括書込み

2、3 データアドレス

	アドレス	データ形式	データ名称
コイル (0X)	1 - 32		DO
	33 - 40		冷接点温度補償 SW
入力 ステータス(1X)	1 - 32		DI
	33 - 40		ADC レンジオーバー
入力 レジスタ (3X)	17 - 32	F	アナログ入力値 (工業単位値)
	49 - 50	F	冷接点温度値
	81 - 96	I	アナログ入力チャンネル・ステータス
	513	I	システム・ステータス
	514 - 521	B16	形式
	522 - 529	I	機番
	530 - 537	I	ハードウェア・バージョン番号
	538 - 545	B16	ソフトウェア・バージョン番号
保持 レジスタ (4X)	49 - 50	F	入力フィルタリング時定数
	145 - 152	I	入力タイプ番号
	514	I	バーンアウトタイプ

(注) I : 16 ビット長整数、F : 32 ビット長浮動小数点データ、Bn : n バイト長文字列

2、4 コイル (DO) データ詳細

(1) DO (1 ~ 32)

32 点の DO (Discrete Output) です。アドレス 1 はアラーム出力で、その他は予備です。

(2) 冷接点温度補償 SW (33 ~ 40)

各入力チャンネル (1 ~ 8) の冷接点温度補償を行うか否かのスイッチで、熱電対入力時の場合のみ有効です。ON (1) の場合、冷接点温度補償動作を行います。

2、5 入力ステータス (DI) データ詳細

(1) DI (10001 ~ 10032)

32 点の DI (Discrete Input) 入力用です。アドレス 10001 はトリガー入力で、その他は予備です。

(2) ADC レンジオーバー (10033 ~ 10040)

この値が ON の場合には、各入力チャンネル (1 ~ 8) の ADC 入力値がレンジオーバー (0X0000 または 0XFFFF) になっていることを示します。

2、6 入力レジスタデータ詳細

(1) アナログ入力工業単位値 (30017 ~ 30032)

各入力チャンネル(1 ~ 8)のアナログ入力工業単位値です。データ形式は32ビット長の浮動小数点です。従って、連続した2個の入力レジスタが使用されます。値の単位は、入力のタイプで決定されます。入力タイプが熱電対の場合には温度 です。DC入力の場合は電圧 V です。

(2) 冷接点温度値 (30049 ~ 30050)

冷接点の温度です。データ形式は32ビット長の浮動小数点です。値の単位は温度 です。入力タイプが熱電対のときのみ有効です。

(3) チャンネル・ステータス (30081 ~ 30088)

各アナログ入力チャンネル(1 ~ 8)の状態を示します。各ビットの構成と内容は以下の通りです。

ビット番号	内 容	詳 細
6	入力レンジオーバー	入力がレンジオーバーしているか否かを示します。 以下の状態を検知するとレンジオーバーとなります。 ・ADC 入力値が 0X0000 または 0XFFFF ・熱電対の場合バーンアウトを検出 ・熱電対および測温抵抗値の場合 温度変換テーブルの範囲外になっている。 ・ADC 異常 0 : 正常 1 : レンジオーバー
7	ADC エラー	ADC の状態を示します。 0 : 正常 1 : 異常
12	冷接点温度補償 SW	熱電対の場合の冷接点温度補償状態を示します。 0 : 補償なし 1 : 補償あり
上記以外	予備	システム使用

(4) システム・ステータス (30513)

R2M シリーズのステータスを、16 ビット長整数で示します。各ビットの構成と意味は以下の通りです。

ビット番号	内 容	詳 細
0 ~ 3	予備	システム使用
4	予備	システム使用
5	予備	システム使用
6	E ² PROM 診断	モジュールの E ² PROM サムチェック診断結果を示します。 0 : 正常 1 : サムチェックエラーです。
7	ADC エラー	ADC の状態を示します。 0 : 正常 1 : 異常
8 ~ 15	予備	システム使用

(5) 形式 (30514 ~ 30521)

R2M シリーズの形式を、16 バイト長の文字列で示します。

(6) 機番 (30522 ~ 30529)

R2M シリーズの機番 (シリアル番号) を、16 バイト長の文字列で示します。

(7) ハードウェア・バージョン番号 (30530 ~ 30537)

R2M シリーズのハードウェア・バージョン番号を、16 バイト長の文字列で示します。

(8) ソフトウェア・バージョン番号 (30538 ~ 30545)

R2M シリーズのソフトウェア・バージョン番号を、16 バイト長の文字列で示します。

2、7 保持レジスタ・データ詳細

(1) 入力フィルタリング時定数 (40049 ~ 40050)

入力信号のフィルタリング¹です。データ形式は32ビット長の浮動小数点です。値の単位は秒です。入力信号にノイズが大きい場合に有効です。0秒の場合には、フィルタリング動作を行いません。全て入力チャンネル(1~8)に対して共通です。

(2) 入出力タイプ番号 (40145 ~ 40152)

各入力チャンネル(1~8)の入力タイプ番号です。入力タイプ番号を16ビット整数で示します。入力タイプ番号と入力タイプを(表34)に示します。表中、測定可能レンジは参考のため記載したもので、仕様を示したものではありません。仕様については各仕様書を参照して下さい。

入力タイプと入力タイプ番号 (表 34)

形 式	入出力タイプ	入出力 タイプ 番 号	入力レンジ 名 称	測定可能レンジ	備 考
R2M - 2G3	DC 入力	0	-10 to 10 V	-10 to 10 V	
R2M - 2H3	熱電対入力	16	(PR)	0 to 1760	
		17	K (CA)	-270 to 1370	
		18	E (CRC)	-270 to 1000	
		19	J (IC)	-210 to 1200	
		20	T (CC)	-270 to 400	
		21	B (RH)	200 to 1820	
		22	R	-50 to 1760	
		23	S	-50 to 1760	
		24	C (WRe 5-26)	0 to 2315	
		25	N	-270 to 1300	
		26	U	-200 to 400	
		27	L	-200 to 900	
		28	P (Platinel 2)	0 to 1395	

(3) バーンアウトタイプ (40514)

信号入力のバーンアウト検出時のバーンアウト動作タイプを指定します。

0 : バーンアウト検出なし

1 : Up Scale

2 : Down Scale

補 足 2 Modbus / TCP プロトコル

以下に、Modbus / TCP プロトコル仕様の概要を掲載します。Modbus / TCP の仕様は、Web 上で「OPEN Modbus / TCP SPECIFICATION」として公表されています。詳細は原文を参照下さい。

1、はじめに

Modbus / TCP は、業界標準となっている Modbus プロトコルを TCP / IP に拡張したプロトコルです。TCP / IP を使用していますのでインターネット環境でもメッセージのやり取りが可能になります。Modbus / TCP では、クライアント・サーバモデルで通信を行なうことになります。クライアントは従来の Modbus プロトコルでのマスターに相当し、サーバはスレーブに対応します。TCP / IP プロトコルを基本にしていますので、複数クライアント、複数サーバのサポートが可能になります。即ちマルチマスター、マルチスレーブのシステム構成が可能になります。

TCP 経由で Modbus / TCP 通信を行なう場合には、登録されたポート番号 502 を用います。

2、プロトコルのレイアウト

以下に Modbus リクエストまたはレスポンスが Modbus / TCP ネットワーク上で通信されるときのカプセル化の一般的な形式を記載します。Modbus / TCP 通信データ上のファンクションコードからデータ部分の終わりまでのリクエストとレスポンス本体の構造が、以下の Modbus 体系で、完全に同一レイアウトと同一の意味を持ちます。

Modbus serial port - ASCII encoding

Modbus serial port - RTU (binary) encoding

アドレスの解釈については相違がありますので注意が必要です。

Modbus のスレーブアドレス・フィールドは、1 バイト長のユニット識別子に置換えられます。ユニット識別子はブリッジやゲートウェイ (例えば、72EM) のようなデバイスと通信する場合に用いられ、同一 IP アドレス上の複数の独立した終端ユニットを指定するのに使います。単一ユニットの場合には、IP アドレスがユニットを指定しますのでスレーブアドレス・フィールドは意味を持ちません。

Modbus / TCP プロトコルのリクエストとレスポンスには、次のような 6 バイト情報を前に付けます。

byte 0 : トランザクション識別子 - サーバはコピーするのみ - 通常 0

byte 1 : トランザクション識別子 - サーバはコピーするのみ - 通常 0

byte 2 : プロトコル識別子 = 0

byte 3 : プロトコル識別子 = 0

byte 4 : フィールド長 (上位バイト) = 0 (なぜなら全てのメッセージは 256 以下だから)

byte 5 : フィールド長 (下位バイト) = 以下に続くバイト列の数

byte 6 : ユニット識別子 (スレーブ・アドレスと言っていたもの)

byte 7 : Modbus ファンクションコード

byte 8 以降 : 必要なデータ列 (Modbus プロトコル参照)

トランザクション識別子は、サーバ側はコピーを返すのみで、クライアント側でのメッセージのトランザクション管理に使います。0 固定でも構いません。

Modbus / TCP と Modbus とのメッセージのデータ構造の関係を図式化すると以下のようになります。

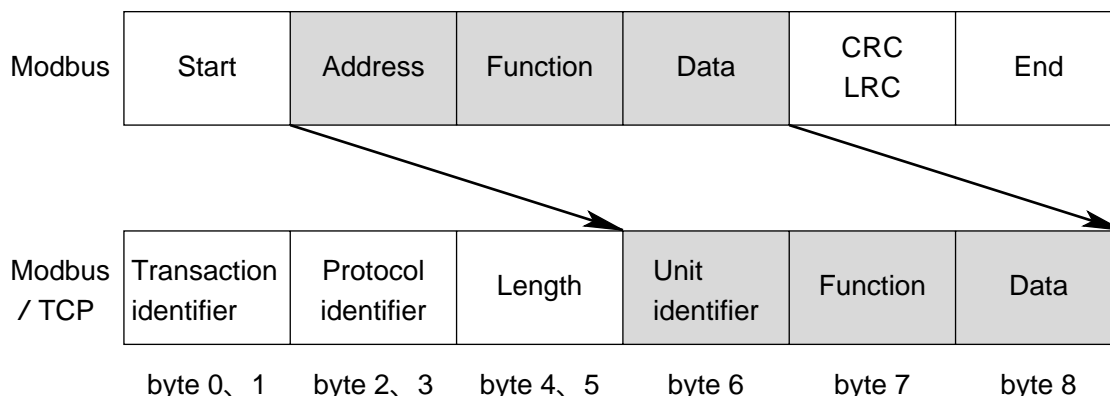


図 Modbus と Modbus / TCP メッセージのデータ構成の関係

3、動作の例

ユニット識別子9のデバイスからオフセット4のレジスタを読むと、値5が返ってくるトランザクションの例は以下のようになります。

リクエスト : 00 00 00 00 00 06 09 03 00 04 00 01

応 答 : 00 00 00 00 00 05 09 03 02 00 05

4、注意事項

Modbus / TCP では LRC または CRC-16 チェック・フィールドが不要な点に注意しなければなりません。パケットの正確な伝送のチェックには、TCP / IP やリンク層 (例えば Ethernet) のチェックメカニズムが使われます。