

Modbus 通信プロトコル

KN0008-00／2023 年 12 月 10 日／こがねさん(著)

<https://www.kumikomist.com/>

■目次

1. はじめに	2	(0Ah)	17
1.1. 用語	2	3.8.7. Return Bus Message Count (0Bh)	17
2. 通信プロトコル	3	3.8.8. Return Bus Communication Error Count	
2.1. ASCII モード	3	(0Ch)	17
2.1.1. 通信設定	4	3.8.9. Return Bus Exception Error Count (0Dh)	
2.1.2. LRC	4		18
2.2. RTU モード	5	3.8.10. Return Server Message Count (0Eh) .	18
2.2.1. 通信設定	5	3.8.11. Return Server No Response Count (0Fh)	
2.2.2. CRC	6		18
2.3. 共通項目	6	3.8.12. Return Server NAK Count (10h)	18
2.3.1. Address	6	3.8.13. Return Server Busy Count (11h)	18
2.3.2. Function	6	3.8.14. Return Bus Character Overrun Count	
2.3.3. Data	7	(12h)	18
3. ファンクションコード	8	3.8.15. Clear Overrun Counter and Flag (14h)	
3.1. Read Coils (01h)	9		19
3.2. Read Discrete Inputs (02h)	9	3.9. Get Com Event Counter (0Bh)	20
3.3. Read Holding Registers (03h)	10	3.10. Get Com Event Log (0Ch)	21
3.4. Read Input Registers (04h)	10	3.10.1. イベントについて	22
3.5. Write Single Coil (05h)	11	3.11. Write Multiple Coils (0Fh)	23
3.6. Write Single Register (06h)	12	3.12. Write Multiple registers (10h)	24
3.7. Read Exception Status (07h)	13	3.13. Report Server ID (11h)	25
3.8. Diagnostics (08h)	14	3.14. Read File Record (14h)	25
3.8.1. Return Query Data (00h)	14	3.15. Write File Record (15h)	25
3.8.2. Restart Communications Option (01h) .	15	3.16. Mask Write Register (16h)	25
3.8.3. Return Diagnostic Register (02h)	15	3.17. Read/Write Multiple registers (17h)	25
3.8.4. Change ASCII Input Delimiter (03h)	16	3.18. Read FIFO Queue (18h)	26
3.8.5. Force Listen Only Mode (04h)	16	3.19. 例外コード	26
3.8.6. Clear Counters and Diagnostic Register			

■文書内の記号

	取り扱いにおける禁止事項（してはいけないこと）を示しています。		取り扱いにおける注記事項を示しています。
	取扱における指示事項（必ずしなければならないこと）を示しています。		取り扱いにおけるポイントを示しています。

1. はじめに

Modbus（モドバス）は、Modicon 社が同社の PLC（ピーエルシー：Programmable Logic Controller）向けに策定した通信プロトコルです。仕様が公開されていてかつ無料利用できることから、産業向け機器で幅広く使われるようになりました。特に RS-485 上の通信プロトコルとしてよく利用されています。

Modbus には RS-485 などのシリアル通信向けと、TCP/IP 向けがあります。本書ではシリアル通信向けについてのみ紹介します。

	● Modbus は規格として組織的な運用をされていません。このため Modbus 対応と謳われている機器同士を接続しても、すんなりとは動かない恐れがあります。世の中のほとんどの機器は Modbus 準拠なのです。 ● そんなこともあり Modbus 仕様の詳細を掴みきれておらず、本書内の一部項目は空欄となっています。
---	--

1.1. 用語

用語	内容
クライアント	メッセージを送信する機器です。
サーバー	クライアントからのメッセージを受け、対応する処理を行う機器です。
メッセージ	クライアントからサーバーに、サーバーからクライアントに返信する通信データです。
コイル	On/Off 制御により、状態やモードを切り替えるスイッチのようなものを表します。
入力レジスター	読み出し専用のレジスターです。
保持レジスター	読み書き可能なレジスターです。
FIFO	先入れ先出し（First In, First Out）を表します。キューの動作原理を表す言葉で、キューにデータを入れることをエンキュー、データを取り出すことをデキューといいます。
受信専用モード	Listen Only Mode です。受信のみを行い、クライアントへの返信を行いません。

2. 通信プロトコル

Modbus にはシリアル通信向けの Modbus RTU、Modbus ASCII と、TCP/IP 向けの Modbus TCP があります。

モード	内容
Modbus RTU	RTU は Remote Telemetry Unit の略で、訳すと遠隔監視制御装置などになります。RTU モードと呼ぶのはいまいっパツとしない、むしろバイナリーモードというべき Modbus の基本的な通信モードです。 1 バイトのバイナリーデータを、そのまま送受信します。シリアル通信向けとして幅広く使用されています。
Modbus ASCII	バイナリー通信が困難な場合に用いられる通信モードで、1 バイトのデータを上位ビットと下位ビットに分け、2 文字の ASCII として送受信します。 RTU モードと比較してデータ量が 2 倍になるのがデメリットですが、通信データを確認しやすいことや、他の通信プロトコルと競合を起こしにくいというメリットはあります。
Modbus TCP/IP	TCP/IP 上で Modbus 通信を行うためのプロトコルです。本書ではこれ以上紹介しません。

2.1. ASCII モード

ASCII モードの送信データは次のとおりです。

Start	Address	Function	Data	LRC	End
':'				チェックサム	<CR><LF>
1 バイト	2 バイト	2 バイト	2×N バイト※1	2 バイト	2 バイト

※1 N の最大は 252 です。

ASCII モードでは「:(3Ah)」で始まり、「<CR><LF> (0D0Ah)」で終わります。その間に許されるのは「'0'～'9' (30h～39h)」と「'A'～'F' (41h～46h)」のみです。

2.1.1. 通信設定

ASCII モードでの通信設定は下記が基本です。

- (1) データ長：7 ビット
- (2) パリティ：偶数パリティ
- (3) ストップビット長：1 ビット

ただし、すべての Modbus 対応機器がこの通信設定であるとは限りません。このため下記の範囲から選択できるようにしておくべきです。

- (1) データ長：7 ビット / 8 ビット
- (2) パリティ：なし / 偶数 / 奇数
- (3) ストップビット長：1 ビット / 2 ビット

	● データ長 7 ビット、パリティなしの場合、ストップビット長は 2 ビットにすべきです。 ● データ長 8 ビットまたはパリティありの場合、ストップビット長は 1 ビットにすべきです。
---	---

2.1.2. LRC

LRC (Longitudinal Redundancy Check) は水平パリティチェックと呼ばれるもので、メッセージが正しくやりとりされているかを簡易的に確認するために使用します。Address から Data までのデータを 1 バイトずつ加算していき、最後に 2 の補数を取った値となります。

例) 「010203040506」の場合

合計は「01h + 02h + 03h + 04h + 05h + 06h = 15h」であり、この 2 の補数として LRC は「(FFh - 15h) + 1 = EBh」です。結果、送信メッセージは「:010203040506EB<CR><LF>」となります。

2.2. RTU モード

RTU モードの送信データは次のとおりです。

無通信時間	Address	Function	Data	CRC	無通信時間
3.5 バイト	1 バイト	1 バイト	N バイト※1	2 バイト	3.5 バイト

※1 N の最大は 252 です。

ASCII モードと違い、開始文字や終了文字はありません。一定時間（3.5 バイト分）無通信状態が続くと、そこでメッセージの終わりを表します。また一定時間無通信状態が続いたあとに受けた 1 バイト目のデータが、Address となります。なお無通信時間のことを「T1-T2-T3-T4」と表すこともよくありますが、この意味は調べてもわかりませんでした。

また RTU モードの全データは、連続して一気に送信しきる必要があります。バイト間に 1.5 バイト分の無通信状態が続くと、そこで受信側は異常を検知してそれまで受信したデータを破棄します。

例) 19200bps の場合の無通信時間

$$\begin{aligned} & (\text{スタートビット} + \text{データ長} + \text{パリティ} + \text{ストップビット長}) / \text{伝送速度} \times 3.5 \\ & = (1+8+1+1)/19200 \times 3.5 = 2\text{ms} \end{aligned}$$

2.2.1. 通信設定

RTU モードでの通信設定は下記が基本です。

- (1) データ長：8 ビット
- (2) パリティ：偶数パリティ
- (3) ストップビット長：1 ビット

ただし、すべての Modbus 対応機器がこの通信設定であるとは限りません。このため下記の範囲から選択できるようにしておくべきです。

- (1) パリティ：なし / 偶数 / 奇数
- (2) ストップビット長：1 ビット / 2 ビット

	● パリティなしの場合は、ストップビット長を 2 ビットにすべきです。 ● パリティありの場合は、ストップビット長を 1 ビットにすべきです。
---	---

2.2.2. CRC

巡回冗長検査（じゅんかいじょうちょうけんさ、Cyclic Redundancy Check：CRC）は、メッセージが正しくやりとりされているかを確認するために使用します。確認精度は LRC より強固で、USB や Bluetooth などでも用いられています。

計算条件は次のとおりです。

- (1) 多項式：8005h
- (2) 初期値：ffffh
- (3) 出力 XOR：0000h
- (4) シフト方向：左
- (5) 入力反転：あり
- (6) 出力反転：あり

例)「010203040506h」の場合

CRC は DDBAh です。CRC は下位バイトから送信するため、送信メッセージは「01h 02h 03h 04h 05h 06h BAh DDh」となります。

2.3. 共通項目

2.3.1. Address

通信を行うサーバー機器のアドレスを 1～247 で指定します。つまり Modbus の通信網に接続できるサーバー台数は、247 台が上限であることを表しています。

なおアドレス 0 はブロードキャストの用途に使用します。ブロードキャストとは通信網に接続している、すべてのサーバーに対してメッセージを送信するものです。この場合、サーバー側はメッセージを受けて処理を実行するのみで、クライアントに対して返信は行いません。したがって、サーバーからの返信を必要とする読み出し系のファンクションには、ブロードキャスト通信は使えません。

2.3.2. Function

サーバー機器に対して行う指令（ファンクションコード）です。クライアントからサーバーに送信するファンクションコードは 1～127（7Fh）です。MSB（80h）はサーバーからの例外返信に使用されます。詳細は「3. ファンクションコード」を参照ください。

2.3.3. Data

ファンクションコードに対応したデータを送信します。データは可変長で、データがないということもあります。なおデータについてはビッグエンディアンで送信します。

しかしデータフォーマットについても、メーカーによるところがあります。特に問題になりがちなのは 32 ビット以上のデータの取り扱いです。Modbus では特に規定がないことから、メーカー毎に仕様が異なる可能性がとて高いものとなります。このためもし Modbus 機器を開発する側に立つときは、図 1、図 2 のようにフォーマットを図で示すと良いでしょう。

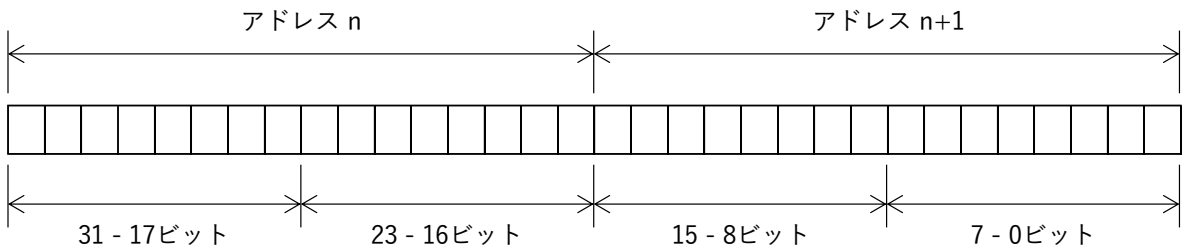


図 1 32 ビット整数

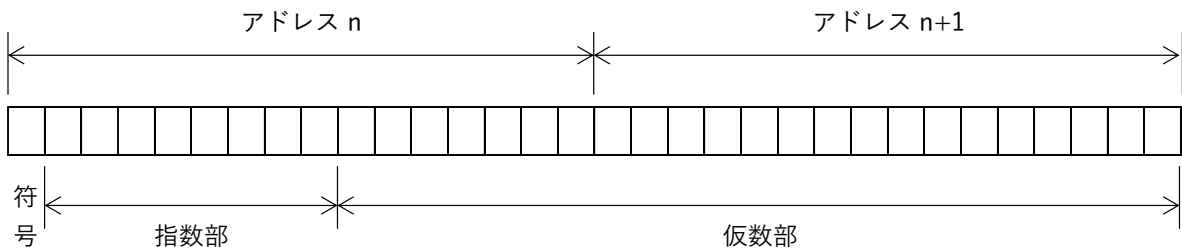


図 2 32 ビット浮動小数点数

- 図 1、図 2 は上位バイトから順に並んでいますが、下位バイトから並んでいる可能性も考慮すべきです。

3. ファンクションコード

表1 ファンクションリスト

ファンクションタイプ			ファンクション名	コード (Hex)
データ アクセス	ビット アクセス	デジタル信号入力	Read Discrete Inputs	02
		内部ビット または コイル	Read Coils	01
			Write Single Coil	05
			Write Multiple Coils	0F
	16 ビット アクセス	入力レジスター	Read Input Register	04
		内部レジスター または 出力レジスター	Read Holding Registers	03
			Write Single Register	06
			Write Multiple Registers	10
			Read/Write Multiple Registers	17
			Mask Write Register	16
			Read FIFO queue	18
			ファイルレコードアクセス	Read File record
	Write File record	15		
診断			Read Exception status	07
			Diagnostic	08
			Get Com event counter	0B
			Get Com Event Log	0C
			Report Server ID	11
			Read device Identification	2B
その他			Encapsulated Interface Transport	2B

	● Modbus で扱えるのは 16 ビットデータまでです。32 ビット以上のデータを扱うためには、2 つ以上のレジスター枠を用いる必要があります。 ● メッセージでやり取りする開始アドレスは、実際のアドレス-1 となります。
---	---

3.1. Read Coils (01h)

サーバーの連続したデジタル出力（On/Off）情報を読み出します。

返信のコイル情報は LSB から順に並びます。アドレス 0 から 10 ビット分のコイル情報を取得するとします。
このときコイル情報が順に「0 1 1 1 0 0 1 0 1 1」とすると、返信データは「4Eh 03h」となります。

表 2 送信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	01h
読み出し開始アドレス	2 バイト	0～65535
読み出し数	2 バイト	1～2000

表 3 返信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	01h
データ数	1 バイト	N※1
コイル情報	N バイト	ビット毎：1=On / 0=Off

※1 N=読み出し数/8。余りが 0 でないとき、N=N+1。

表 4 エラー応答

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	81h
例外コード	1 バイト	1 or 2 or 3 or 4

表 5 通信例

送信		返信	
フィールド名	(Hex)	フィールド名	(Hex)
ファンクションコード	01	ファンクションコード	01
読み出し開始アドレス (H)	00	データ数	03
読み出し開始アドレス (L)	13	コイル情報 [No.27 - 20]	CD
読み出し数 (H)	00	コイル情報 [No.35 - 28]	6B
読み出し数 (L)	13	コイル情報 [No.38 - 36]	05

3.2. Read Discrete Inputs (02h)

サーバーのデジタル入力（On/Off）情報を読み出します。使用方法は「3.1. Read Coils (01h)」と同じです。

3.3. Read Holding Registers (03h)

サーバーの保持レジスタの値を読み出します。

表 6 送信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	03h
読み出し開始アドレス	2 バイト	0～65535
読み出し数	2 バイト	1～125

表 7 返信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	03h
データ数	1 バイト	2 × N ^{※1}
レジスタ値	2 × N バイト	0000h～FFFFh

※1 N=読み出し数。

表 8 エラー応答

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	83h
例外コード	1 バイト	1 or 2 or 3 or 4

表 9 通信例

送信		返信	
フィールド名	(Hex)	フィールド名	(Hex)
ファンクションコード	03	ファンクションコード	03
読み出し開始アドレス (H)	00	データ数	06
読み出し開始アドレス (L)	6B	レジスタ値 [No.108 (H)]	02
読み出し数 (H)	00	レジスタ値 [No.108 (L)]	2B
読み出し数 (L)	03	レジスタ値 [No.109 (H)]	00
		レジスタ値 [No.109 (L)]	00
		レジスタ値 [No.110 (H)]	00
		レジスタ値 [No.110 (L)]	64

3.4. Read Input Registers (04h)

サーバーの入力レジスタの値を読み出します。使用方法は「3.3. Read Holding Registers (03h)」と同じです。

3.5. Write Single Coil (05h)

サーバーのデジタル出力（On/Off）情報に書き込みます。

表 10 送信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	05h
書き込み開始アドレス	2 バイト	0～65535
コイル情報	2 バイト	FF00h=On / 0000h=Off

表 11 返信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	05h
書き込み開始アドレス	2 バイト	受けた値
コイル情報	2 バイト	受けた値

表 12 エラー応答

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	85h
例外コード	1 バイト	1 or 2 or 3 or 4

表 13 通信例

送信		返信	
フィールド名	(Hex)	フィールド名	(Hex)
ファンクションコード	05	ファンクションコード	05
書き込み開始アドレス (H)	00	書き込み開始アドレス (H)	00
書き込み開始アドレス (L)	AC	書き込み開始アドレス (L)	AC
コイル情報 (H)	FF	コイル情報 (H)	FF
コイル情報 (L)	00	コイル情報 (L)	00

3.6. Write Single Register (06h)

サーバーの保持レジスターに書き込みます。

表 14 送信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	06h
書き込み開始アドレス	2 バイト	0～65535
レジスター値	2 バイト	0000h～FFFFh

表 15 返信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	06h
書き込み開始アドレス	2 バイト	受けた値
レジスター値	2 バイト	受けた値

表 16 エラー応答

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	86h
例外コード	1 バイト	1 or 2 or 3 or 4

表 17 通信例

送信		返信	
フィールド名	(Hex)	フィールド名	(Hex)
ファンクションコード	06	ファンクションコード	06
書き込み開始アドレス (H)	00	書き込み開始アドレス (H)	00
書き込み開始アドレス (L)	01	書き込み開始アドレス (L)	01
レジスター値 (H)	00	レジスター値 (H)	00
レジスター値 (L)	03	レジスター値 (L)	03

3.7. Read Exception Status (07h)

8 個の例外ステータスの内容を読み出します。詳細な用途は不明です。

表 18 送信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	07h

表 19 返信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	07h
例外ステータス	1 バイト	00h~FFh

表 20 エラー応答

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	87h
例外コード	1 バイト	1 or 4

表 21 通信例

送信		返信	
フィールド名	(Hex)	フィールド名	(Hex)
ファンクションコード	07	ファンクションコード	07
		例外ステータス	6D

3.8. Diagnostics (08h)

クライアントとサーバー間の通信状態や、サーバーのエラー状態を確認します。本ファンクションにはさらにサブファンクションがあり、実行するテストの内容を指定します（表 22 参照）。

表 22 サブファンクションリスト

コード (Hex)	サブファンクション名	内容
00	Return Query Data	通信テストを行います。
01	Restart Communications Option	再起動します。
02	Return Diagnostic Register	診断レジスターを読み出します。
03	Change ASCII Input Delimiter	メッセージ最後の区切り文字を変更します。
04	Force Listen Only Mode	受信専用モードに切り替えます。
0A	Clear Counters and Diagnostic Register	カウンタと診断レジスターをクリアします。
0B	Return Bus Message Count	メッセージカウントを読み出します。
0C	Return Bus Communication Error Count	CRC エラーカウントを読み出します。
0D	Return Bus Exception Error Count	例外カウントを読み出します。
0E	Return Server Message Count	当該サーバー向けのメッセージカウントを読み出します。
0F	Return Server No Response Count	返信しなかったメッセージカウントを読み出します。
10	Return Server NAK Count	否定応答の例外カウントを読み出します。
11	Return Server Busy Count	ビジー応答の例外カウントを読み出します。
12	Return Bus Character Overrun Count	オーバーランエラーカウントを読み出します。
14	Clear Overrun Counter and Flag	オーバーランエラーカウントをクリアします。

3.8.1. Return Query Data (00h)

サーバーから返信があるかを確認することで、通信に問題がないことを確認します。

表 23 送信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	08h
サブファンクションコード	2 バイト	0000h
データ	2 バイト	0000h~FFFFh

表 24 返信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	08h
サブファンクションコード	2 バイト	0000h
データ	2 バイト	受けた値

3.8.2. Restart Communications Option (01h)

サーバーのシリアルポートを初期化するため再起動します。受信専用モードの場合は返信しません。
本サブファンクションは受信専用モードを解除する唯一の方法です。

表 25 送信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	08h
サブファンクションコード	2 バイト	0001h
データ	2 バイト	FF00h=ログをクリアーする 0000h=ログをクリアーしない

表 26 返信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	08h
サブファンクションコード	2 バイト	0001h
データ	2 バイト	受けた値

3.8.3. Return Diagnostic Register (02h)

サーバーのもつ診断レジスターを読み出します。

表 27 送信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	08h
サブファンクションコード	2 バイト	0002h
データ	2 バイト	0000h

表 28 返信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	08h
サブファンクションコード	2 バイト	0002h
データ	2 バイト	診断レジスター値

3.8.4. Change ASCII Input Delimiter (03h)

ASCII モードにおけるメッセージ最後の区切り文字「<LF>」を別の文字に置き換えます。

表 29 送信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	08h
サブファンクションコード	2 バイト	0003h
データ	2 バイト	CHAR※1 00h

※1 CHAR には 0～127 の文字データを格納します。

表 30 返信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	08h
サブコード	2 バイト	0003h
データ	2 バイト	受けた値

3.8.5. Force Listen Only Mode (04h)

サーバーを受信専用モードに切り替えます。本サブファンクションにサーバーは返信しません。またこれ以降、すべてのメッセージに対してサーバーは返信しなくなります。受信専用モードを解除するには「3.8.2. Restart Communications Option (01h)」を実行してください。

表 31 送信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	08h
サブファンクションコード	2 バイト	0004h
データ	2 バイト	0000h

3.8.6. Clear Counters and Diagnostic Register (0Ah)

カウンターと診断レジスターをクリアします。なおカウンターは電源投入時もクリアします。

表 32 送信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	08h
サブファンクションコード	2 バイト	000Ah
データ	2 バイト	0000h

表 33 返信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	08h
サブファンクションコード	2 バイト	000Ah
データ	2 バイト	0000h

3.8.7. Return Bus Message Count (0Bh)

サーバーが検出したメッセージカウントを読み出します。本カウントは電源投入、再起動、カウンタークリアによってリセットされます。

表 34 送信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	08h
サブファンクションコード	2 バイト	000Bh
データ	2 バイト	0000h

表 35 返信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	08h
サブファンクションコード	2 バイト	000Bh
データ	2 バイト	カウント値

3.8.8. Return Bus Communication Error Count (0Ch)

サーバーが検出した CRC エラーカウントを読み出します。本カウントは電源投入、再起動、カウンタークリアによってリセットされます。使用方法は「3.8.7. Return Bus Message Count (0Bh)」と同じです。

3.8.9. Return Bus Exception Error Count (0Dh)

サーバーが検出した例外カウントを読み出します。本カウントは電源投入、再起動、カウンタークリアーによってリセットされます。使用方法是「3.8.7. Return Bus Message Count (0Bh)」と同じです。

3.8.10. Return Server Message Count (0Eh)

サーバーが検出した、当該サーバー向けのメッセージカウントを読み出します。本カウントは電源投入、再起動、カウンタークリアーによってリセットされます。使用方法是「3.8.7. Return Bus Message Count (0Bh)」と同じです。

3.8.11. Return Server No Response Count (0Fh)

サーバーが検出した、当該サーバー向けのメッセージのうち、返信しなかったメッセージカウントを読み出します。本カウントは電源投入、再起動、カウンタークリアーによってリセットされます。使用方法是「3.8.7. Return Bus Message Count (0Bh)」と同じです。

3.8.12. Return Server NAK Count (10h)

サーバーが検出した否定応答の例外カウントを読み出します。本カウントは電源投入、再起動、カウンタークリアーによってリセットされます。使用方法是「3.8.7. Return Bus Message Count (0Bh)」と同じです。

3.8.13. Return Server Busy Count (11h)

サーバーが検出したビジー応答の例外カウントを読み出します。本カウントは電源投入、再起動、カウンタークリアーによってリセットされます。使用方法是「3.8.7. Return Bus Message Count (0Bh)」と同じです。

3.8.14. Return Bus Character Overrun Count (12h)

サーバーが検出したオーバーランエラーカウントを読み出します。本カウントは電源投入、再起動、カウンタークリアーによってリセットされます。使用方法是「3.8.7. Return Bus Message Count (0Bh)」と同じです。

3.8.15. Clear Overrun Counter and Flag (14h)

オーバーランエラーカウントをクリアし、エラーフラグをリセットします。

表 36 送信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	08h
サブファンクションコード	2 バイト	0014h
データ	2 バイト	0000h

表 37 返信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	08h
サブファンクションコード	2 バイト	0014h
データ	2 バイト	0000h

3.9. Get Com Event Counter (0Bh)

サーバーからステータスとイベントカウンタを読み出します。

一連のメッセージの前後で本カウンタを読み出すことで、サーバーが正常にメッセージを処理したかを確認できます。本カウンタは「3.8.2. Restart Communications Option (01h)」や「3.8.6. Clear Counters and Diagnostic Register (0Ah)」でリセットできます。

表 38 送信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	0Bh

表 39 返信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	0Bh
ステータス	2 バイト	0000h~FFFFh
イベントカウンタ	2 バイト	0~65535

表 40 エラー応答

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	8Bh
例外コード	1 バイト	1 or 4

表 41 通信例

送信		返信	
フィールド名	(Hex)	フィールド名	(Hex)
ファンクションコード	0B	ファンクションコード	0B
		ステータス (H)	FF
		ステータス (L)	FF
		イベントカウンタ (H)	01
		イベントカウンタ (L)	08

3.10. Get Com Event Log (0Ch)

サーバーからステータス、イベントカウント、メッセージカウントおよびイベントを読み出します。ステータスとイベントカウントは「3.9. Get Com Event Counter (0Bh)」と同じ値です。メッセージカウントは「3.8.7. Return Bus Message Count (0Bh)」と同じ値です。

イベントは 0～64 バイトで時系列順に格納しており、イベント 0 が最新となります。詳細は「3.10.1. イベントについて」を参照ください。

表 42 送信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	0Ch

表 43 返信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	0Ch
データ数	1 バイト	N※1
ステータス	2 バイト	0000h～FFFFh
イベントカウント	2 バイト	0～65535
メッセージカウント	2 バイト	0～65535
イベント	(N-6)バイト	

※1 N=イベント数+3×2。

表 44 エラー応答

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	8Ch
例外コード	1 バイト	1 or 4

表 45 通信例

送信		返信	
フィールド名	(Hex)	フィールド名	(Hex)
ファンクションコード	0C	ファンクションコード	0C
		データ数	08
		ステータス (H)	00
		ステータス (L)	00
		イベントカウント (H)	01
		イベントカウント (L)	08
		メッセージカウント (H)	01
		メッセージカウント (L)	21
		イベント 0	20
		イベント 1	00

3.10.1. イベントについて

イベントには2種類あります。イベントはビットフィールドとなっており、7ビットの状態によって分かります。また今後、6ビットの状態によってさらに分岐していく可能性があります。

表 46 受信イベント (7 ビット = 1)

ビット	内容
0	
1	通信エラー
2	
3	
4	オーバーランエラー
5	受信専用モードへ切り替え
6	ブロードキャストを受信
7	1

表 47 送信イベント (7 ビット = 0)

ビット	内容
0	読み取り例外を送信 (例外コード 1~3)
1	障害例外を送信 (例外コード 4)
2	ビジー例外を送信 (例外コード 5~6)
3	否定応答例外を送信 (例外コード 7)
4	書き込みタイムアウト
5	受信専用モード中
6	1
7	0

	● 例外コードは「3.19 例外コード」を参照ください。
---	------------------------------

3.11. Write Multiple Coils (0Fh)

サーバーの連続したデジタル出力（On/Off）情報に書き込みます。

書き込むコイル情報は LSB から順に並びます。アドレス 0 から 10 ビット分のコイル情報を書き込みするとします。このときコイル情報が順に「0111001011」とすると、「4Eh 03h」となります。

表 48 送信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	0Fh
書き込み開始アドレス	2 バイト	0～65535
書き込み数	2 バイト	1～1968
データ数	1 バイト	N ^{※1}
コイル情報	N バイト	ビット毎：1=On / 0=Off

※1 N=書き込み数/8。余りが 0 でないとき、N=N+1。

表 49 返信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	0Fh
書き込み開始アドレス	2 バイト	受けた値
書き込み数	2 バイト	受けた値

表 50 エラー応答

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	8Fh
例外コード	1 バイト	1 or 2 or 3 or 4

表 51 通信例

送信		返信	
フィールド名	(Hex)	フィールド名	(Hex)
ファンクションコード	0F	ファンクションコード	05
書き込み開始アドレス (H)	00	書き込み開始アドレス (H)	00
書き込み開始アドレス (L)	13	書き込み開始アドレス (L)	13
書き込み数 (H)	00	書き込み数 (H)	00
書き込み数 (L)	0A	書き込み数 (L)	0A
データ数	02		
コイル情報 [27 - 20]	CD		
コイル情報 [29 - 28]	01		

3.12. Write Multiple registers (10h)

サーバーの連続したレジスターに書き込みます。

表 52 送信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	10h
書き込み開始アドレス	2 バイト	0～65535
書き込み数	2 バイト	1～123
データ数	1 バイト	2×N ^{※1}
レジスター値	2×N バイト	0000h～FFFFh

※2 N=書き込み数。

表 53 返信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	10h
書き込み開始アドレス	2 バイト	受けた値
書き込み数	2 バイト	受けた値

表 54 エラー応答

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	90h
例外コード	1 バイト	1 or 2 or 3 or 4

表 55 通信例

送信		返信	
フィールド名	(Hex)	フィールド名	(Hex)
ファンクションコード	10	ファンクションコード	10
書き込み開始アドレス (H)	00	書き込み開始アドレス (H)	00
書き込み開始アドレス (L)	01	書き込み開始アドレス (L)	01
書き込み数 (H)	00	書き込み数 (H)	00
書き込み数 (L)	02	書き込み数 (L)	02
データ数	04		
レジスター値 0 (H)	00		
レジスター値 0 (L)	0A		
レジスター値 1 (H)	01		
時レスター値 1 (L)	02		

3.13. Report Server ID (11h)

3.14. Read File Record (14h)

3.15. Write File Record (15h)

3.16. Mask Write Register (16h)

3.17. Read/Write Multiple registers (17h)

サーバーから保持レジスタの読み出しと書き込みを同時に行います。書き込み処理は読み出し処理の前に実行します。

表 56 送信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	17h
読み出し開始アドレス	2 バイト	0～65535
読み出し数	2 バイト	1～125
書き込み開始アドレス	2 バイト	0～65535
書き込み数	2 バイト	1～121
データ数	1 バイト	$2 \times N^{\ast 1}$
レジスタ値	$2 \times N$ バイト	0000h～FFFFh

※1 N=書き込み数。

表 57 返信データ

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	17h
データ数	2 バイト	$2 \times n^{\ast 1}$
レジスタ値	$2 \times n$ バイト	0000h～FFFFh

※1 n=読み出し数。

表 58 エラー応答

フィールド名	サイズ	範囲
ファンクションコード	1 バイト	97h
例外コード	1 バイト	1 or 2 or 3 or 4

表 59 通信例

送信		返信	
フィールド名	(Hex)	フィールド名	(Hex)
ファンクションコード	17	ファンクションコード	17
読み出し開始アドレス (H)	00	データ数	0C
読み出し開始アドレス (L)	03	レジスター値 0 (H)	00
読み出し数 (H)	00	レジスター値 0 (L)	FE
読み出し数 (L)	06	レジスター値 1 (H)	0A
書き込み開始アドレス (H)	00	レジスター値 1 (L)	CD
書き込み開始アドレス (L)	0E	レジスター値 2 (H)	00
書き込み数 (H)	00	レジスター値 2 (L)	01
書き込み数 (L)	03	レジスター値 3 (H)	00
データ数	06	レジスター値 3 (L)	03
レジスター値 0 (H)	00	レジスター値 4 (H)	00
レジスター値 0 (L)	FF	レジスター値 4 (L)	0D
レジスター値 1 (H)	00	レジスター値 5 (H)	00
レジスター値 1 (L)	FF	レジスター値 5 (L)	FF
レジスター値 2 (H)	00		
レジスター値 2 (L)	FF		

3.18. Read FIFO Queue (18h)

3.19. 例外コード

コード	内容	詳細
1	不正な機能	サポートしていないファンクションコードです。
2	不正なアドレス	アドレスが範囲外です。
3	不正な値	データが範囲外です。
4	障害	読み出しや書き込みに失敗しました。
5	肯定応答	処理中です。
6	ビジー	処理中です。
7	否定応答	処理を実行できません。診断機能を使って状態を確認してください。
8	パリティエラー	メモリー内でパリティエラーを検出しました。